Modern Fruits Web Store with Personalized Recommender System

By

LIM JUN PENG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

JANUARY 2023

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: __Modern Fruits Web Store with Personalized Recommender System ___

Academic Session: __January__2023_______

I ___________LIM JUN PENG____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

Lim

SunTeikHeng

(Author's signature) (Supervisor's signature)

Address:

___12, Jalan Cenderawasih Indah,

Taman Cenderawasih Indah, 14300,

Nibong Tebal, Pulau Pinang

_Ts Sun Teik Heng _

Supervisor's name

Date: ____28/4/2023_________________

Date: ______28/4/2023____

Universiti Tunku Abdul Rahman			
Form Title : **Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE**	Page No.: **1 of 1**

FACULTY/INSTITUTE* OF _ UNIVERSITI TUNKU ABDUL RAHMAN__

Date: ___28/4/2023_______________

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ______***Lim Jun Peng***___________________________ (ID No: _***19ACB03930***____) has completed this final year project/ dissertation/ thesis* entitled "____________*Modern Fruits Web Store with Personalized Recommender System* ____ _" under the supervision of _____________ Ts Sun Teik Heng ________________ (Supervisor) from the Department of _______ Information Systems ___________________, Faculty/Institute* of ______ Information and Communication Technology ___ , and _____________________ (Co-Supervisor)* from the Department of ______________, Faculty/Institute* of ___________________________.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

[illegible]

____

(*Lim Jun Peng*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Modern Fruits Web Store with Personalized Recommender System**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature	:	Lim
Name	:	LIM JUN PENG
Date	:	28/4/2023

ACKNOWLEDGEMENTS

I would like to sincerely express my thanks and appreciation to my supervisors, Ts Sun Teik Heng who has given me this good opportunity to take part in this project. This act as my first step to full-stack application development integrating with technologies and API. A million thanks to you.

First of all, I would like to my family for always supporting me and continuously encourage me throughout the whole course. I also would like to thank my friends to stand by my side as we are study together and been through the tough moments.

ABSTRACT

E-commerce is emerged in the last few years. With the current demanding trend and existing technology, people can start selling and buying anything online virtually. This project will develop an online fruit and fruit juice web application. Besides, there will be a discussion about the strength and weakness of the current existing system. The system architecture, framework is also being reviewed before building the web application which will provide support on understanding the concept, methodology and also the design of full-stack web application. After the literature review, the research tells that some existing systems do not actually implement the recommender systems to provide product recommendations. So, the main goal and objective of this project is to develop a well-developed online fruit and fruit juice web application with the recommender system. The techniques being chosen for the recommender system testing are the non-personalized popularity-based and personalized collaborative-based filtering covering user-based and item-based. ASP.NET Core MVC is the main programming framework to build this e-commerce project and deploy using Azure services. The proposed system is a full stack web ap plication which can involve the communication between the front-end framework and the back-end framework.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AWS	Amazon Web Services
CRUD	Create,Read,Update,Delete
SSMS	SQL Server Management Studio

CHAPTER 1

Introduction

E-commerce, which is the short abbreviation for electronic commerce, can be defined as the buying and selling activities of products, goods and services via the Internet [1]. With the current existing Internet, people are able to start selling and buying anything online virtually. It can be in terms of physical goods like electronics devices, apparel, food or virtual goods like services, software. E-commerce provides a convenient platform for buyers and sellers to purchase or sell things they intended, which also promotes active business transaction between two aspects which is facilitated through the powerful Internet. Furthermore, business transactions can be carried out in several main models of e-commerce. Some of the popular models are Business-to-Consumer (B2C), Business-to-Business (B2B). Before this, the business model for commerce is more on the physical interaction between the retail seller and buyers. To go more into clearer definition and explanation, the kind of commerce business form is called brick-and-mortar which is totally in contrast to E-commerce [2]. Brick-and-mortar businesses often require customers to purchase any products and services via face-to-face interaction in a specific physical location, namely a retail supermarket, or a store selling the products. However, due to the enormous changes in evolution and technology, it is expeditiously starting to be taken over the place by E-commerce. Many companies and retail sellers are starting to transform and put in the effort in order to embrace the trend of E-commerce by promoting their business and selling the products or services online.

In this day and age, E-commerce is substantially popular and not only limited to those big retail companies. Generally, e-commerce businesses are available to everyone including start-up person, small business enterprise and large business enterprise. E-commerce introduces many advantages over conventional types of retail selling. In recent days, as mentioned the online business which is categorised under E-commerce is easy to start up for anyone. Some of the popular norms nowadays is that sellers will start to deploy their own websites using some simple web design tools or hire some web designers to help them to develop their websites after they provide the business requirements. After that, they can start to sell their products and operate their online business easily. In addition, e-commerce can help the sellers to increase the visibility and availability of their business because it can be accessed and purchased worldwide as long as the delivery service is provided, and the website is always active. The other noteworthy benefits for both aspects include reducing the travelling cost of customers, gaining new customers faster and providing better personalization in terms of goods and services to every customer. In sum, we can clearly understand that e-commerce is playing a very important role and dominant position in the present market.

Although it seems to be easy-peasy to start up the online e-commerce business, it requires external efforts on understanding the customer behaviours in order to provide better user and shopping experience on the website. In order to maintain the competitiveness in the e-commerce market, recommender systems exist to help the seller to filter the products and services to particular target groups of customers/audience according to their needs and preferences. A very clear example can be shown is that Amazon, one of the biggest giants in the e-commerce industry, launched the recommender system approximately 2 decades ago just to provide better recommendations to their millions of customers to find the product they may need. According to a report made by Microsoft Research, approximately 30% of the Amazon.com website browsing is indeed actually contributed by the recommendation system. By gathering as much information as possible on customers, e-commerce recommender systems will create customer models that can represent their shopping' characteristics and activities [3]. In the e-commerce website, the recommender system plays the role of virtual salesman to customise each customer's website by only recommending and displaying the products or services which they are more likely to view. Unquestionably, new incomers or visitors will be interested to consume the goods which in turn boost the sales as the product result being displayed matched to their needs. The next time when they revisit the website, the recommender system which already studied the customer shopping behaviours will personalise their shopping experience by showing the recommended list of products which are highly related to the previous orders or their interests. There are also other alternatives in which a recommender system can study the customer buying behaviours commonly through the products rating, transaction, and their actions. For example, the number of clicks into a specific product, or the duration of browsing the particular category of product [4].

Other than that, a recommender system can increase their customer loyalty and make the website to be the preference choice of customers predominantly when they need to purchase any products because the website can quickly understand and satisfy their demand. This is because customers in the present days apparently like to visit the website which they are familiar with and make the purchase.

This project mainly serves to present an e-commerce website which is titled as "Modern fruit store with personalised juice purchase and delivery" with the recommender system which can ameliorate the customer shopping experience. There are some existing similar web applications that can be found in the market currently and there will be some study and discussion in the next chapter. The target audience in my proposed system will be those people or individuals who like to include fruits in their daily meals and strive for a balanced diet, usually the group of women and elderly.

1.1 Problem Statement and Motivation

Recently, despite the fact that people start to sell their products through their websites and the trend is still growing, the fruit and fruit juice industry is not yet taking a formidable step and not ready to move to the e-commerce mode**. Buying fruits or fruit juices online and delivery are still not a popular trend and high demand in this generation** in many areas as compared to other products such as apparels, food or furniture. In common sense, people always understand that fresh fruits are actually essential and being necessities to our human bodies. Nevertheless, it is not a usual or common activity for everyone to purchase fresh fruit from the fruit stalls, supermarket daily or weekly, especially for those people who keep busting for work. In special scenarios, there are certain groups of customers who are having difficulties purchasing fruits constantly from the fruit selling. Additionally, people are afraid to buy the fruit from crowded areas, especially in some serious pandemic areas or lockdown area like China even though they realise that fruits are way vital to maintain their immune system. This indicates that people are struggling to get the fruits physically from the store. **In certain existing fruit or fruit juice selling companies, they only create websites to show what fruits or products are available for selling in their physical**

store but not open for order online and delivery. In this case, customers have no choice but to spend their time and travel to their destination to get their goods.

Next, some existing fruit or fruit juice ordering systems are operating business well and providing delivery services**. Despite some of the existing websites indeed having really impressive front-end design, they are still practising the approach in which they are showing the static long list of products on their websites arranged in some categories.** Anyway, this method is no longer effective for business as dashboard management and dynamic shopping experience are preferable for both business owners and consumers. This statement can be proved when the website keeps showing the same list of products at the page even though different users have logged in to their system and business owners not able to manage their business online easily. As the e-commerce era keeps growing, this way may not be able to tailor to every customer's unique needs, particularly when the number of products on the website keeps increasing. In this situation, customers may find it troublesome to find certain products from big bundles of products. In certain existing systems, there is also a lack of basic filters which can display the products ascendingly with the product rating in which this may influence their shopping experience and lower the first impression**. Some of the existing systems or websites also do not implement certain recommender systems or algorithms to personalise their customer's shopping experience.** Without these features, the management personnel of the existing systems may struggle to get insight into every customer preference and their shopping behaviours. Consequently, it is hard for the existing fruit and fruit juice ordering system to automate the customization of their websites and provide more recommendations to each user according to their preferences and needs. Meanwhile, they need to pay for external work and attempt to build their customer loyalty.

As a result of the growth of e-commerce and technology becoming more advanced, this gives me the motivation for building this project. In fact, one of the major reasons is because the rising usage of mobile devices uplift the growth of e-commerce. Online transactions are carried out anytime and anywhere by people simply using a mobile device. E-commerce transactions mainly rely on the advantages brought by those online transactions where the customers can easily purchase the product from their system using the integrated payment method they provided. In today's society, people tend to purchase products virtually rather than physically. Thus, this also provides me with the encouragement and the aim of this project to propose a fruit and fruit juice web application with the current expert technology to ease the customers to purchase the goods from the online system. Moreover, as mentioned in the previous section, some of the existing systems do not actually try to understand the customer preferences in order to provide relevant product recommendations for their purchases. So, the proposed solution shall be able to personalise the customer shopping experiences by understanding the customer shopping behaviors through several aspects which will be discussed in later sections.

1.2 Research Objectives

The objectives of the project are:

- To analyse different algorithm or technique that can be used as the recommender system.
- To design a recommender system that provides product recommendation to customers.
- To develop an online fruit and fruit juice web application which should have all the required features using ASP.NET framework.

1.3 Project Scope and Direction

This section basically covers the future work that needs to be done. It will focus more on the features that need to be developed and included in the proposed system.

Product module for displaying purpose

All the products should be added to the website with their detailed information. The important product information such as the product price, the number of available stocks, quantity and its ratings. When customers clicked the particular product, they should be able to gain all the product information. Other than that, product description can also be included if it is necessary to let customers understand more about the products.

Product Category

All the products should be labelled to their respective categories or types. This is a must for every e-commerce website because the genre of items will keep increasing when the business continues to expand. This also can speed up the process for a customer to easily target the product by just checking the product category. This process should also be implemented dynamically instead of manually manage the category of each product one by one.

Search Engine

This is another tool that should be included in the e-commerce system in order to provide customers the ability to just search for a specific target of a product. For example, customers can use the keyword like “juice” to filter all the results matched. This can also be useful in collecting the customer preference which can help websites to improve their shopping experience.

Login Module

Login module is a prerequisite feature before customers make orders. Firstly, it is useful for the company to track every customer and ease the communication with them. Besides, it is useful for the recommender system to provide different recommendations to every user with their own taste and needs. In some cases, it is an authentication module to protect some privacy and also security because e-commerce accounts conventionally will involve the money transactions.

Shopping Cart

This is a place where customers can add the products into which will store only the products customers are interested in placing orders. After customers make the decision, they can just proceed to check out their cart. If customers are not interested in purchasing any products on the website, they can just clear the cart.

Wish list

A wish list is more like a temporary store to allow users to allocate their interested products which they have not yet decided to buy. Every time they revisit the website, they are still able to view the wish list. If once they have made the decision, they can directly just add the product into cart from the wish list.

Customer Profile

Customer is able to manage their profile on the website. For instance, their shipping information and also their personal information. This will fasten the process as they do not need to repeat the tedious process of filling up long list of information before they can submit their order.

Recommender systems

A recommender system that implements the popularity-based and collaborative-based filtering which will provide some recommendations. This will read the customer preference and recommend the similar products which customers may like or give a high rating. The recommender system will implement a machine learning model to predict the products which may be suitable to the customers before recommending product for each of the customers. As a result, there will be some work that needs to be done before building the recommender systems like collect and analyse the dataset, data pre-processing, train and test the model. Lastly, the recommenders will need to be deployed in Flask and run on the website to provide recommendations.

Newsletter

When most of the customers first come to the website, they can subscribe to the newsletter even though they do not have registered an account on the website. The business owner will have a list of mailing list and will send the promotions, news, or releases of new products on their website to customers on the list. This will increase the website exposure to customers through Gmail with interesting mail content and increase the two-way interactions.

Management Dashboard

This will be mainly implemented to assist the business administrator in managing the business. For instance, they may need products, categories, and order dashboard to have a glance into each and every transaction and manage all the products like updating their product information and price.

Payment

A payment module is required for the customers to purchase the products. In this system, there will be integrated payment option using Stripe and allow customers to pay using their credit and debit card after filling up the required information.

1.4 Contributions

This proposed system can encourage more people to start up their own e-commerce business. This is because e-commerce online business will never limit to the boundary of large enterprises only. Other than that, this is a small step which can encourage more fruit and fruit juice industries to transform or upgrade their business model to an e-commerce model and start operating their online business. The predominant targets are those small and medium companies, fruit stores or grocery shops. Moreover, it can help business companies and owners to understand more about their customer shopping behaviours through the recommender systems. With the recommender systems, undeniably it can improve the company sales. In the future, they can continuously improve the model and predict any potential unforeseeable trend or demand. Thirdly, this proposed web application makes it more convenient for customers to get their wanted products online. This will greatly reduce the cost of customers in terms of petrol expenses and time to acquire the fruits and fruit juice. By simply just a few clicks, they can receive the products in their home with the delivery services. As a matter of fact, there are certain customers who prefer to purchase the fruits and fruit juice face-to-face because they think that the fruits and juice ordered online may not be as fresh as the fruit from the physical store which they can touch and feel. They are worried that the quality of fruit ordered online may be compromised. Therefore, if the online fruit and fruit juice store business owner can ensure the freshness of their products, the proposed system can act as a middleman which provides a platform for the customers to place orders and dispel the customers' doubts and worries.

1.5 Report Organization

The details of this research are shown in the following chapters. In Chapter 2, there will be some literature being reviewed and discussion on the related project tools, framework, and similar systems. After that, there will be a preliminary study of the required proposed method and approach presented in Chapter 3 with the title of system methodology and approach. To be further in detail, design specifications, system requirements and some diagrams like the system architecture diagram will be covered in this section. Next, Chapter 4 will report on the system design and Chapter 5 will be covering the system implementation like mentioning all the required hardware and software tools, setting and configuration. In Chapter 6, there will be system evaluation and discussion on the result after performing testing on the system. All the output obtained according to the proposed method in Chapter 3 and 4 will be shown in this part and mention any project challenge and issues that had been met. Lastly, Chapter 7 wraps up the project and summarizes the project including the problem, motivation, and proposed solutions. There will be some future enhancements and recommendations to be proposed on the end of this project development.

CHAPTER 2

Literature Review

2.1 Overview

This chapter will be mainly discussing the current existing similar systems including comparison of the strength and weakness for each of existing similar systems. After reviewing these systems, there will be clear and concise direction on how the proposed system can improve and do better and try to avoid those limitations that have been discovered in future work. Furthermore, there will also be study and discussion on the algorithm which will be used for the recommender systems. Literature study on the system architecture and the framework will provide me with a blueprint and plan to build the structure of the proposed system and understand which one will be the good option.

2.2 Recommender System Algorithm

In this section, there will be discussion of several algorithms and techniques which are normally implemented in the recommender systems to personalise the customers shopping experiences by providing suggestions or recommendations to customers. To further go deeper into the recommendations, it can be actually categorised into several types which can be either personalised or non-personalized, item-to-item correlation, attribute based or people-to-people correlation [4]. For instance, personalised recommendations do not require manual input or just simply involve minimal work and normally will be based on the customer's preferences. On the contrary, for the case of non-personalized recommendations, commonly they are generated according to the actual product ratings which are being reviewed by other customers. In the current technology era, there are plenty of algorithms and alternatives which can be implemented in the recommender system. For example, popularity-based, content-based filtering and collaborative filtering, association rules are some of the popular recommendation techniques and this section will be focusing on the first three techniques.

2.2.1 Popularity-based filtering

Popularity-based filtering is the approach which considered as the most direct one in product recommender system. This is because it is indeed not a personalized recommendation algorithm for everyone though [19]. In normal recommender system, it always looks into the special attributes and indicators like ratings, purchase history and user preference in order to suggest the most suitable recommendation to customers. For popularity-based filtering, it will take into accounts of all the rating count or purchase count from the database or dataset. In some cases, the number can also be further converted into score. The usual scenario is that the product with the greatest number of purchase and rate or highest score will be considered as the most popular product in the dataset and suggest it to the end customers. At the end, it will just need the effort to sort the products in a ranking. Same list of product recommendations will be generated and displayed to the customers with top N popular recommendations without sense of personalization [19]. This is one of the limitations but on the other hand, it is a handful strategy for most of the online business to cold start their the recommender system and attracting new customers with just populating a list of most popular products on their website [19].

2.2.2 Content-based filtering

Content-based filtering is a technique which mainly focuses on retrieving information, further performing analysis and starting to filter it before generating any predictions. The good point for this filtering is that it may not need to refer to any order history which must be taken from the database while it focuses on the product profile. It is regularly implemented in the situations that the content can be analysed, and the recommendation will be generated by extracting the features from the content which had been seen or evaluated by the customers before typically like categories, types, brands and product description. There is a general example here, when the particular customers had rated a product with higher rating, potentially the products which are closely related to high rating products will usually be recommended to the same customer [5]. In this case, it can be stated that content-based filtering systems are able to recommend the products by checking the history of products which have been liked or given a high rating by the customers [6]. Hence it is possible to create a user's content-based profile by just collecting the product information from the customers referring to their past liked products, rating, and other data which can describe the customer behaviour. The next favourable characteristic of content-based filtering is that the recommendations given are not dependent on other customer profiles. Thus, it is possible that the time required to make adjustments on the customer profile in order to provide more accurate recommendations can be in a very short span of period. Besides, it is a common technique to be implemented in a lot of start-up and small enterprise companies which has rich variety of products.

Nevertheless, this algorithm may still contain some limitations or weaknesses. First of all, the main challenge for this technique is that content description in the profile requires much understanding and knowledge. In this case, the content of product information and user profile have to be well-organised in order to make sure the recommender systems can understand the meaning and provide good recommendations. For examples, missing in product-related information which has been rated by customers will make the recommender system to generate suitable recommendation. Even though content-based filtering can generate recommendations if there are less or no ratings or any liked items in the customer profile, overspecialization may be a serious problem. This is because there is not sufficient information that can be analysed from the customer profile. Due to the limited information, the recommender system is only able to generate recommendations of a small range of similar items that may be rated by the customers in the past. In such situations, the recommender system is not very effective and may not provide the recommendations which are outside the ratings [6].

2.2.3 Collaborative-based filtering

Collaborative-based filtering is another type of technique which is totally the opposite of content-based filtering. The reason is because the recommendations cannot be simply generated through a personal user profile or metadata. The algorithm must go through the actual whole database in order to discover knowledge about the preferred products for each group of customers. After that, the recommender system which implemented the collaborative-based filtering will begin to make recommendations by firstly analysing all the customers who do have the highly identical preferences and highly related interests. This group of customers can be formed as a group and it is called neighbours [5].

The two main type of Collaborative-based filtering are user-based and item-based. For user-based collaborative filtering, it will firstly find the group of customers who share the similar product ratings and compare their similarity between two users. Pearson correlation or cosine similarity are the usual techniques used to measure the similarity. The products with higher similarity strength will be recommended to the customer even though they did not rate the products before [20]. Next, for the item-based collaborative, it will find every pair of the products in which one of them or both of them has rated by the customer and measure the similarity of the products' rating across every customer who have rated both of them via computation using Pearson correlation or cosine similarity. The next step will then just sort the product according to similarity strength and the products which have higher correlation/ similarity strength with the product which has been rated by customers will be displayed to customers [21].

One of the apparent strengths in this technique is that certain recommendations can be generated to the customers even though they never rated these relevant products before. This is primarily because the group of neighbours the customers belong to might have already given some high rating in the past. The major advantage of this technique is that it possesses the ability to provide generations despite there is lack of content in the customer's profile. Moreover, the technique can outperform when it is hard to analyse the content description in the user profile. In fact, collaborative-based filtering is famous for being widely implemented in a lot of well-known companies like Amazon, Facebook, and LinkedIn and indeed it performs well.

On the other hand, the wide usage of collaborative-based filtering uncovers some limitations and problems. One of the famous problems is the data sparsity problems and this is commonly happening in giant e-commerce websites due to the inactive number of purchases [4]. The primary cause of this problem is the lack of information provided in the database. Without enough information, the performance of the recommender systems will degrade and not be able to group the customers to the perfect neighbours. Consequently, it is possible that customers may receive some recommendations which are fit to their needs. The next limitation is the cold-start problem which usually happens in the small business. Similarly, this can be explained in the situation that the recommender system is not able to make any thesis if it does not collect sufficient information. The common scenario is that when new users sign up for an account, logically their profile is totally empty and does not have any behaviours done in the past. The recommender system totally has no knowledge on how to match them to the neighbours and will not recommend any products to them. In sum, collaborative-based filtering has a harsh requirement on the adequate amount of information about the customers and products.

2.3 System Architecture

2.3.1 Web and Business Tiers

In this technology era, there are many ways to start deploying web applications and people are beginning to move to cloud services. Cloud computing has become the new trend and advanced technology since several years ago as they bring much convenience to the business world, especially in terms of virtualization and resource utilisation. In the cloud computing industry, there are many cloud service providers. Among them, Microsoft Azure, Amazon Web Services and Google Cloud are giants in providing cloud services. On top of that, there are mostly 3 types of cloud computing which are the – Infrastructure as a Service (IaaS), Software as a Service (SaaS) and Platform as a Service (PaaS) [7].

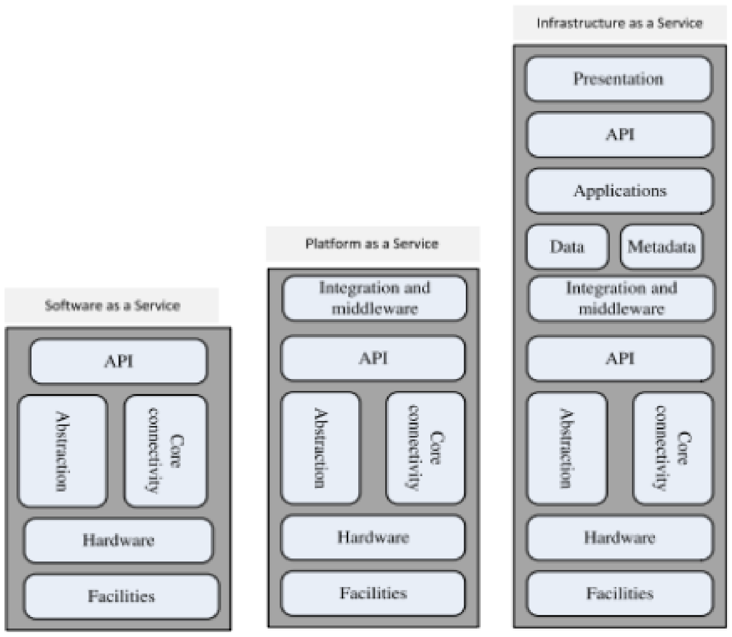


Figure 2.3.1.1 The infrastructure difference between 3 types of cloud computing

Amazon Web Services

Amazon Web Services was launched by Amazon in 2006. The cloud computing services are still improving, and it is the dominant position cloud service provider in the current market. This is because it supports a lot of online businesses in e-commerce especially selling products like food, furniture, books and others. The AWS was actually an IaaS-based model in which they provide low-cost, multi-variety of services and easy to use computing infrastructure [7]. Besides, AWS provides the popular web service - Elastic Compute Cloud (EC2) which allows end users to run their applications which can support different operating systems like Linux, Microsoft, and FreeBSD via a simple interface. AWS also supports the usage of different programming languages as they are a flexible and open platform [8]. Talking about database storage, AWS has provided multiple databases including relational database and non-relational database. For instance, Amazon SimpleDB, AmazonRDS and DynamoDB. However, it happens to be hard to learn and use by novice users because it provides overload features and options. Besides, customers also need to make good decisions before starting the subscription because the price varies in a complex way. The price is usually charged per hour.

Microsoft Azure

Microsoft Azure often called Azure, is another superb cloud service provided by Microsoft. It mainly provides the flexibility for the customers or users to develop their application because it is also an open platform which supports development by using any existing programming language like ASP.NET and PHP, framework and also the tools. It is actually IaaS, PaaS, and SaaS based cloud computing services which provides different services for their cloud application. Inside Azure, Compute, Fabric Controller and Storage are constructed as the three main core elements [7]. To further explain this, Compute can build up the environment to do the computation while Fabric Controller will be responsible for the management of the cloud application. Meanwhile, it will ensure the rapid connections between switches and also the servers which are normally called as nodes. Other than that, it also provided database storage and support like CosmosDB, Azure database for PostgreSQL and database for MySQL. One of the limitations for Microsoft Azure is the complicated pricing and it is charger per minute. Another common problem is the complexity of using Microsoft Azure. Sometimes, it requires training for novice users to gain start-up knowledge and experts to provide support in using the services. However, people are still willing to use Azure because they provide some Microsoft Azure Certification training and Microsoft has comprehensive documentation and guidance in their official website.

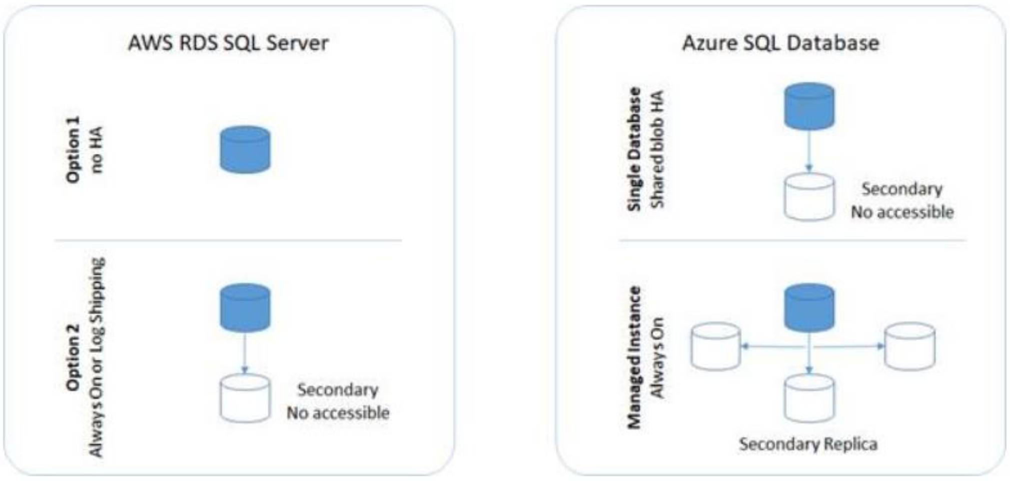


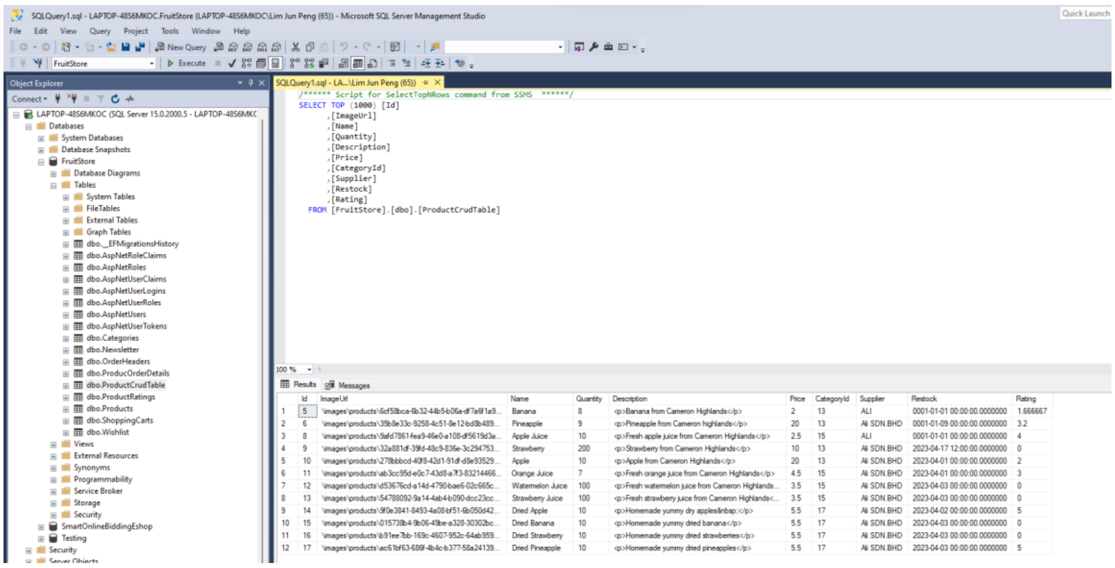
Figure 2.3.1.2 The difference between AWS RDS SQL Server and Azure SQL Database

For this development project, Azure will be chosen as the approach to deploy the web application once the end of development. Inside Azure, the main services will be used is the resource group. The other resources which are also required to store the data of web application are the SQL server and SQL database for database technology, Azure Monitor, Azure Key Vault. Other than that, the app service plan is used to allocate all the front-end and back-end resources of the project development. The reason to choose Azure SQL database over AWS RDS SQL Server is because of its higher availability and secondary read replica. Azure SQL database can support up to 99.99% uptime while AWS RDS can support up to 99.95% availability [22]. The noteworthy point about Azure SQL Database service is that it allows the deployed web applications to access the read-only version of the database so that it would not affect the primary read/write to the database by providing a secondary read replica for customer who subscribe the premium availability tier. This feature is still not supportable in AWS RDS SQL server; thus, it needs the connection to the primary read/write to the database instance every time.

2.3.2 Data Tiers

Relational database

Traditional relational databases work very well in storing structured data in the table format with rows and columns and each of the tables will have the unique attributes in the tables to differentiate between rows of records. It has been working very well until now and it is still implemented. The relational database systems such as SQL is based on the ACID properties which are the Atomicity, Consistency, Isolation and Durability [9]. The main advantage is because that relational database reduces the data redundancies and can promote data integrity. Secondly, users are able to retrieve data from multiple tables by using complex queries. As mentioned, this is because the tables are interlinking to each other using keys. Nevertheless, the main drawback is that the structure and relationship between tables are way too defined and rigid. This problem causes the database manipulation to be complex when more data comes in. The common SQL databases are Azure SQL database, MySQL, Oracle SQL and PostgreSQL.


SQLQuery1.sql - LAPTOP-48S6MKOC.FruitStore (LAPTOP-48S6MKOC\Lim Jun Peng (65)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
Execute
FruitStore
Object Explorer
SQLQuery1.sql - LA...\Lim Jun Peng (65))
100 %
Results
Messages

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Id]
      ,[ImageUrl]
      ,[Name]
      ,[Quantity]
      ,[Description]
      ,[Price]
      ,[CategoryId]
      ,[Supplier]
      ,[Restock]
      ,[Rating]
  FROM [FruitStore].[dbo].[ProductCrudTable]
```

	Id	ImageUrl	Name	Quantity	Description	Price	CategoryId	Supplier	Restock	Rating
1	5	\images\products\6cf58bca-6b32-44b5-b06a-df7a6f1a9...	Banana	8	<p>Banana from Cameron Highlands</p>	2	13	ALI	0001-01-01 00:00:00.0000000	1.666667
2	6	\images\products\35b8e33c-9258-4c51-8e12-bd8b489...	Pineapple	9	<p>Pineapple from Cameron highlands</p>	20	13	Ali SDN.BHD	0001-01-09 00:00:00.0000000	3.2
3	8	\images\products\9afd7861-fea9-46e0-a108-df5619d3a...	Apple Juice	10	<p>Fresh apple juice from Cameron Highlands</p>	2.5	15	ALI	0001-01-01 00:00:00.0000000	4
4	9	\images\products\32a881df-39fd-48c9-836e-3c294753...	Strawberry	200	<p>Strawberry from Cameron Highlands</p>	10	13	Ali SDN.BHD	2023-04-17 12:00:00.0000000	0
5	10	\images\products\278bbbcd-40f8-42d1-91df-d8e93529...	Apple	10	<p>Apple from Cameron Highlands</p>	20	13	Ali SDN.BHD	2023-04-01 00:00:00.0000000	2
6	11	\images\products\ab3cc95d-e0c7-43d8-a7f3-83214466...	Orange Juice	7	<p>Fresh orange juice from Cameron Highlands</p>	4.5	15	Ali SDN.BHD	2023-04-01 00:00:00.0000000	3
7	12	\images\products\d53676cd-a14d-4790-bae6-02c665c...	Watermelon Juice	100	<p>Fresh watermelon juice from Cameron Highlands...	3.5	15	Ali SDN.BHD	2023-04-03 00:00:00.0000000	0
8	13	\images\products\54788092-9a14-4ab4-b090-dcc23cc...	Strawberry Juice	100	<p>Fresh strawberry juice from Cameron Highlands<...	3.5	15	Ali SDN.BHD	2023-04-03 00:00:00.0000000	0
9	14	\images\products\9f0e3841-8493-4a08-bf51-6b050d42...	Dried Apple	10	<p>Homemade yummy dry apples&nbsp;</p>	5.5	17	Ali SDN.BHD	2023-04-02 00:00:00.0000000	5
10	15	\images\products\015738b4-9b06-49be-a328-30302bc...	Dried Banana	10	<p>Homemade yummy dried banana</p>	5.5	17	Ali SDN.BHD	2023-04-03 00:00:00.0000000	0
11	16	\images\products\b91ee7bb-169c-4607-952c-64ab959...	Dried Strawberry	10	<p>Homemade yummy dried strawberries</p>	5.5	17	Ali SDN.BHD	2023-04-03 00:00:00.0000000	0
12	17	\images\products\ac61bf63-686f-4b4c-b377-58a24139...	Dried Pineapple	10	<p>Homemade yummy dried pineapples</p>	5.5	17	Ali SDN.BHD	2023-04-03 00:00:00.0000000	5

Figure 2.3.2.1 The Microsoft SQL Server Management Studio

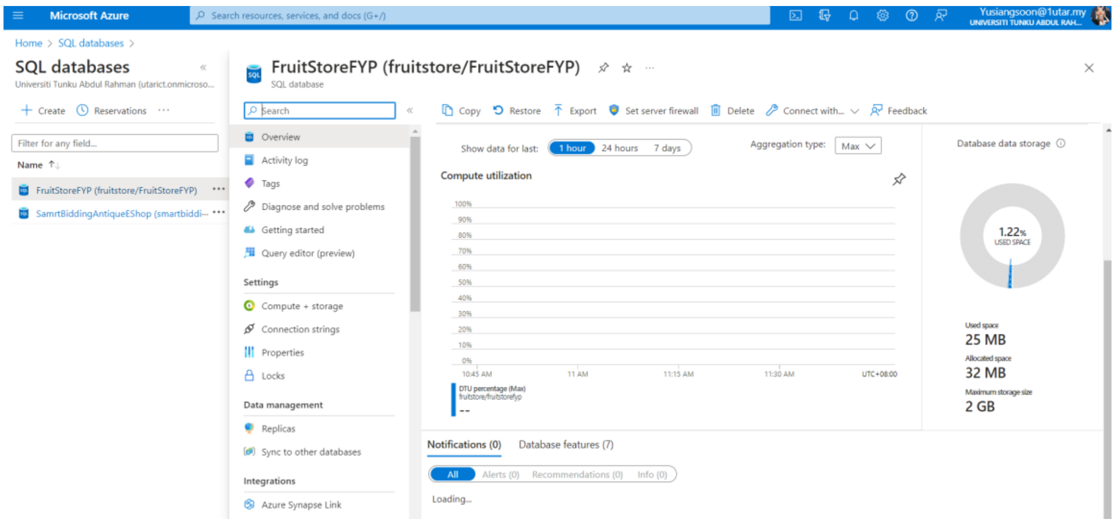


Figure 2.3.2.2 The Azure SQL database in the Azure services

For this development project, the main SQL tools being used are Microsoft SQL Server in Microsoft SQL Server Management Studio (SSMS) and the Azure SQL database in the Azure services. SSMS will be used because it is easy to use as desktop itself can serve as the SQL server and used for testing purpose along the project development through localhost. It supports both GUI and query for database operation which is beginner friendly. The web application is connected to Microsoft SQL Server can also perform the database CRUD operation through entity-framework using migration. The database administrator also provided with the option to view all the data in a just a simple click and able to edit the data from the SSMS directly. At the end of development, the SQL server can connect to Azure SQL database instance and administrators can manage all the tables in the SSMS as well with proper authentication. Azure SQL database plays the role to make sure the web application is deployed to the Internet and saving the data globally. The web application will need to connect to the Azure SQL database through connection string in the App Service plan before deployed. The process is deemed quick for the setting and configuration of the data storage in Azure SQL database and easy for a web application to go online using the Azure services.

Non-Relational database

The non-relational databases are getting popular to be used in the current market in different scenarios. The database structure is different with the relational database. There is a common term called NoSQL which can represent the non-relational database because it does not make use of the SQL query language to retrieve the data. The fundamental reason is because the database is not formed by the rows and columns which are in the tabular form. The non-relational database can consist of 4 types which are the Key-Value database, Column Family database, Document database and Graph database [10]. The notable strength of Non-relational databases is that the database is able to retrieve a large amount of unstructured data that can be in any form by using the key or document ID. It can work well with large scales of data as compared to relational databases. This can be explained why a lot of e-commerce companies implemented this option for their database design. In terms of performance, a NOSQL database can always have a shorter response time when querying the information. Cosmos DB is one clear example of a non-relational database because it can support different kinds of API when building the applications. The others can be DynamoSQL, MongoDB. One of the limitations can be lack of standardisation because they are not suitable and do not aim to store structured data.

2.4 Frameworks

Web development has become increasingly difficult and demanding due to the widespread usage of the internet in modern society. Much grateful that the most recent technology has solved these problems. Nowadays, it seems like everything can go online and this pushes forward the growth of web application development. There are many frameworks which can help people to start-up their business and build their web application. For instance, the front-end framework can be like AngularJS, Bootstrap, and React which are widely used in the client-side framework. Besides, the examples of trendy back-end frameworks like Django, ASP.Net, Laravel and many more. In order to encourage full stack web development in e-commerce, this section will discuss two full stack frameworks which are extensively being used in order to build up the client side and the server-side module.

2.4.1 MEAN stack framework

An emerging tool stack for web-application development which has been very popular in the last few years is called the MEAN Stack or just MEAN. There is an interesting thing to take note of in this framework is originally online developed for the purpose of client-side programming at first but it does evolved very fast start to build up the functionalities for the server-side programming [11]. It is mainly a combination of technologies which is made up of 4 powerful tools which are the Express.js, Angular.js, Node.js and MongoDB. Each of the components are performing their own function in order to greatly maximise the interaction between the client side and server side in the web application. Firstly, Angular.js is an open-source framework which serves as the client framework. Node.js is the server-side application framework which is built by the Javascript that is mainly utilised in order to create a lightweight and well-performing web server environment [10]. The next component is the Express.js. It is a middle person which is used to connect client and server via routing. It is a very important component especially dealing with the Model-View-Controller (MVC) web application as it behaves like the controller. The last element is the MongoDB which serves as the data storage and management system of the MEAN stack web application. It is a NoSQL database which can accept JSON passing format data that is pretty fit to the JavaScript environments. One of the notable strengths of MEAN stack is that it is a dynamic programming language that can maintain good communication between the front-end and back-end and it is widely used in reality as it is comparatively easy to learn.

2.4.2 ASP.NET full-stack framework

ASP.NET is an open-source web programming framework which is created by the famous company called Microsoft to produce dynamic web pages. It is a popular full-stack framework which is mainly used for building modern web applications and services with .NET. ASP.NET which is the extension of .NET platform is built up with tools and libraries specifically for building web apps [14]. There are different development models which can be supported by ASP.NET itself. For instance, ASP.NET Web Forms, ASP.NET MVC for the Model-View-Controller patterns, and ASP.NET Web Pages. In addition, ASP.net can support with several type of programming language which are actually object-oriented based, namely Visual C++.net and Visual C# [12]. It also supports the data structure and algorithms. A few years ago in 2016, ASP.NET Core introduced the open-source version of ASP.NET which can run on other OS such as macOS, Linux, and Windows. According to the Microsoft documentation [15], ASP.NET Core has improved the performance in terms of their speed of running the request as compared to other frameworks as shown in the figure 2.4.1 below. The good thing about ASP.net is that it can perform outstandingly when published and working with Microsoft Azure platform and it is also chosen as one of the popular tools and framework to build the web application in the e-commerce industry [13]. However, when compared to other open source, it may be costly due to the Microsoft licences for example like Microsoft Visual Studio licences and certain frameworks like ASP.NET MVC is a tough subject for naive learners to learn and explore but it also can be an advantage if the person is experienced.

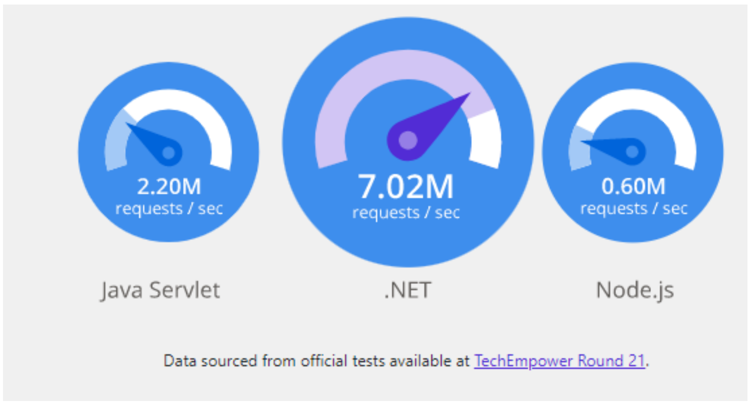


Figure 2.4.1 The performance benchmarking of .NET with other framework

In this project development, ASP.NET MVC is the chosen design pattern for building the web application. According to the official .NET documentation[23], it is a design pattern that introduced for achieving the clear concern of segregation. In further explanation, it is mainly used to separate the data(model), user-interface(view) and application logic (controller) that form the MVC (model-view-controller). It provides the approach for developers to create dynamic websites in MVC pattern. At first developer needs to define and scaffold the Model, then migrate it to the database. When the web requests are passed to the Controller which working with the application logic such as performs data retrieving, adding and updating, they need to work with the

Model first. After that, Views play the role to display the data of the model passing from the controller and lastly display it to the user. With that having said, it provides developers the full control on the web application [24]. It will also ease the process of testing as the complexity has been divided accordingly to model, view, and controller. It helps the .NET developer to easily detects the bugs and solve it using MVC framework.

2.5 Previous work or Literature study on existing system

2.5.1 MBG Online Fruit Shop

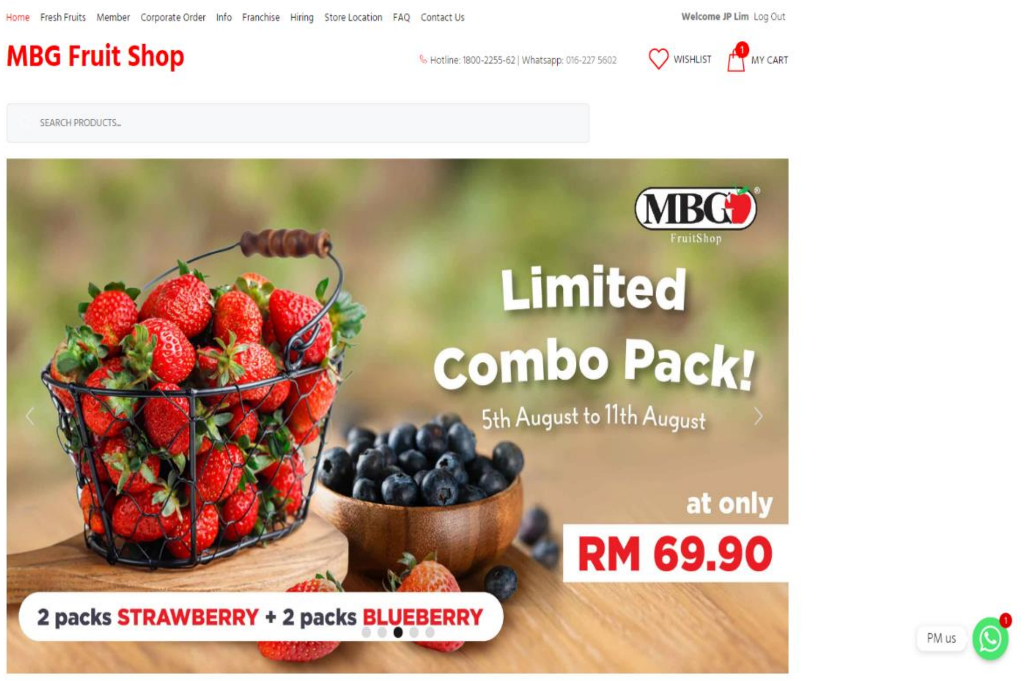


Figure 2.5.1 MBG Fruit Shop

MBG Fruit Shop is a giant in the Malaysian network of fruit shops that places a premium on their fruit quality. MBG has approximately 34 outlets currently opening in Kuala Lumpur and they serve over millions of customers annually. Multi-varieties of fresh fruits including local and imported fruits from other countries are mainly served by MBG. Besides, they also sell other products like fruit baskets, fresh fruit juices and cut fruits.

MBG Fruit Shop provides some basic features in their web application. First and foremost, they have well organised their product diversity in different categories such as tropical and exotic fruits, stone fruits, citrus fruits, grapes. This feature is good because the website will not look very messy and compact to their customers because the products have already been labelled properly. After clicking into the particular product, it clearly shows the detailed information of the product. For example, the figure 2.5.2 below shows the quantity, price, product type of the grapes. From the figure below, customers are also able to check the stock availability.

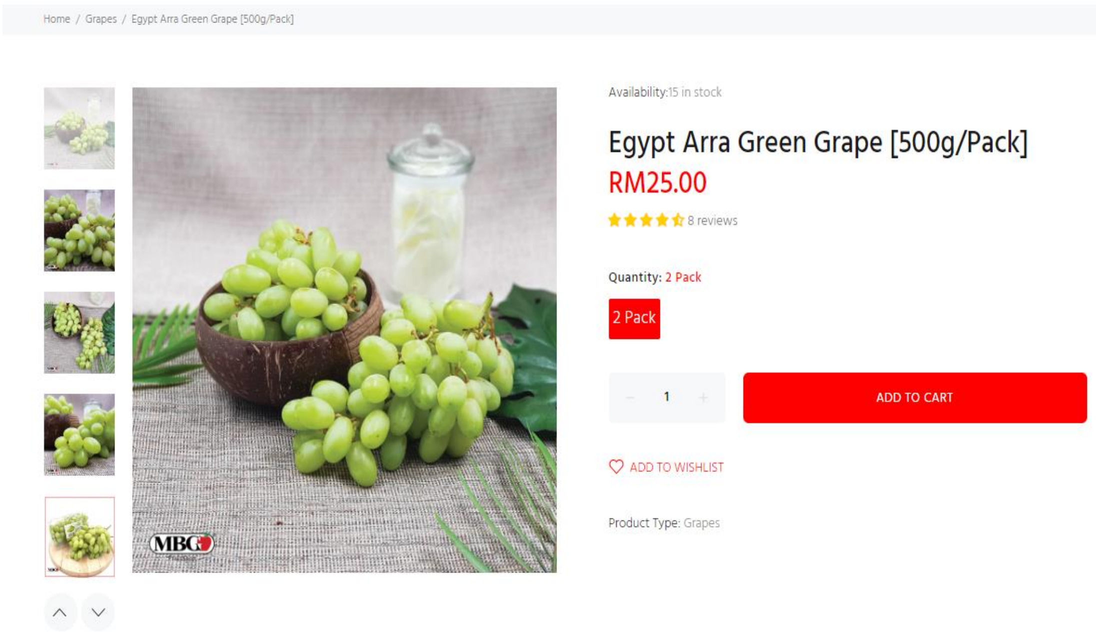


Figure 2.5.2 Detailed information of grapes

Once customers are interested with the fruits or products, they can add the items into the cart. They can view and manage their cart item by just clicking the cart icon. The action can be add or delete the items inside the cart section. Once customers decide to place an order with the items in the cart, they can proceed to check out. Prior to that, MBG fruit store web application will ask users to sign up or login which indicates that they have implemented a signup and login module which allows every customer to have their own user accounts. For new visitors, they can choose either to sign up as a user in the database or either sign in using their Facebook or Gmail account. After that, they can move ahead to check out and make their payments. Move on to the payment section, MBG has integrated extensive payment methods to provide customers more options to make the payment. For instance, credit or debit card, e-wallet. Paypal and online banking. This can ensure that all online customers are able to make the transaction with the methods they are comfortable with. Additionally, MBG also provides a small WhatsApp icon to redirect their customers to their customer service if they have any queries or question about the products and order.

Furthermore, MBG also provides several northworthy features that can improve customer loyalty and enhance their shopping experience. The first notable feature is that MBG develops a wish list feature which allows customers to insert the items which they are interested to buy later or in future. They can still view the wish list anytime or order the products easily from the wish list when they revisit the website. From figure 2.5.2 above, customers can just click the "ADD TO WISHLIST" to perform the action. This feature is quite familiar to see in a lot of shopping apps nowadays, so it is quite useful to exist in the online fruit store. Other than that, they also introduced a personalised membership feature in the website. With this handful feature, customers can earn points every time when they make any purchases or refer to other friends or customers. They can also have extra points to earn when they are visiting the physical store to purchase products, sign up to their mailing list, follow their official social media or even can gain points during the customer's birthday. It is also simple to join as a member because as long as customers create an account on the website, they are automatically enrolled in the membership program. This will in turn increase the stickiness of customers to their website and build up a good relationship with customers. Besides, they also upload some FAQs, useful posts and external info about their products to inform the customers about the benefits of consuming their products. In their website, there is also a unique feature which is that customers are able to translate the website into language they are more accustomed with which in turn gives a good impression to their customers. This can be shown in figure 2.5.3.

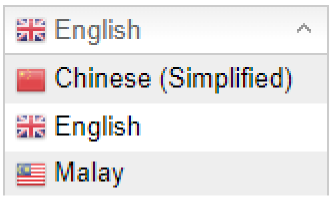
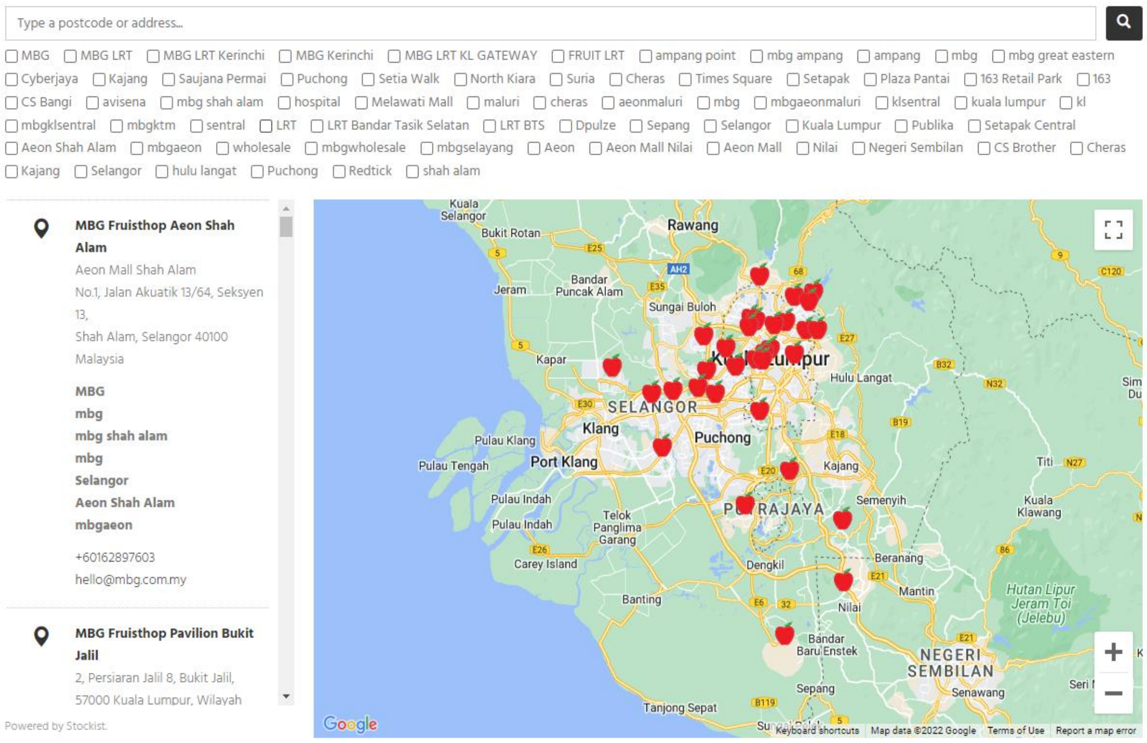


Figure 2.5.3 Translation feature in the MBG website

Last but not least, MBG has made use of the Google Map API shown in figure 2.5.4 to show all the actual location of their outlet in Google Map to ease the customers who wish to visit the MBG Fruit Store which is nearby to their housing address.



MBG Fruistshop Aeon Shah Alam
Aeon Mall Shah Alam
No.1, Jalan Akuatik 13/64, Seksyen 13,
Shah Alam, Selangor 40100
Malaysia

MBG
mbg
mbg shah alam
mbg
Selangor
Aeon Shah Alam
mbgaeon

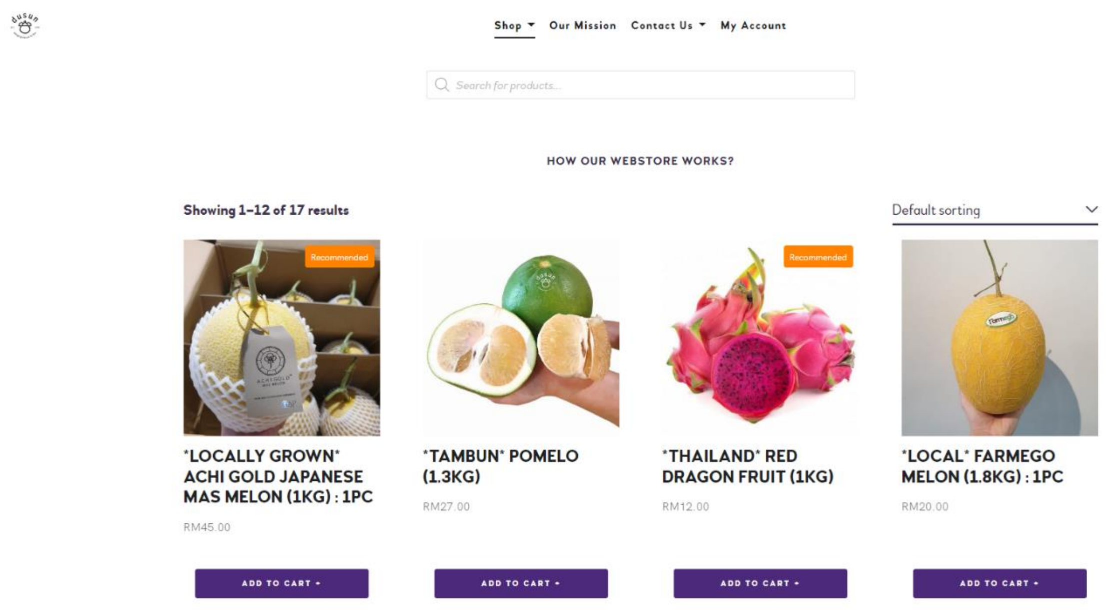
+60162897603
hello@mbg.com.my

MBG Fruistshop Pavilion Bukit Jalil
2, Persiaran Jalil 8, Bukit Jalil,
57000 Kuala Lumpur, Wilayah

Figure 2.5.4 Google Map to show all the nearby MBG physical fruit store

The overall experience of browsing the website of MBG Fruit Store is quite good and it is informative. In contrast, there are certain limitations in the system which can be improved. First of all, the overload features are provided to novice users which may require them to spend more time on testing all the features. Next, after experiencing all the provided features, it can be assumed that they are not implementing any recommender systems in the website. The website will still show the same list of products in every page whenever using two different accounts to login, search and revisit the website. The website also does not provide any recommendation or collect any information from any fields to take note of customer preferences. Some filters of the products also may not be as useful when showing all the products. For instance, the "best sellings" filter do not display the products according to the number of user reviews, product rating and also they are not displaying the the number of sold which may not convey the true meaning of the filters. After checking the FAQs, there is also another limitation which is customers are not able to directly check their order history from the website whereas they have to track their orders via email tracking link or contact their helpline or live chat.

2.5.2 Dusun Fruits, Vegetables and Cold Pressed Juice Online Store



Showing 1–12 of 17 results

"LOCALLY GROWN" ACHI GOLD JAPANESE MAS MELON (1KG) : 1PC
RM45.00

"TAMBUN" POMELO (1.3KG)
RM27.00

"THAILAND" RED DRAGON FRUIT (1KG)
RM12.00

"LOCAL" FARMEGO MELON (1.8KG) : 1PC
RM20.00

Figure 2.5.5 Dusun Fruits,Vegetables and Cold Pressed Juice Online Store

Dusun is a local online fruit, vegetables, fruit juice selling website which is run by a small business owned by a team of independent women. The business scale is comparatively smaller to the first existing system in the upper section but this does not prohibit them from striving to provide fresh and high quality and healthy products to our customers. They aim to provide the fruits, vegetables that are taken from local farmers which are fresh and a better choice for their customers' health.

This website still provides the essential features which should be existing in an online e-commerce website. Firstly, as usual, customers can view all the products selling on the website like the figure shown in 2.5.5. The business owner has categorised the products in very concise categories as shown in the figure 2.5.6 below to easily target different types of customers. For example, fruit lover, greens (vegetable lover), and juice are examples of categories.

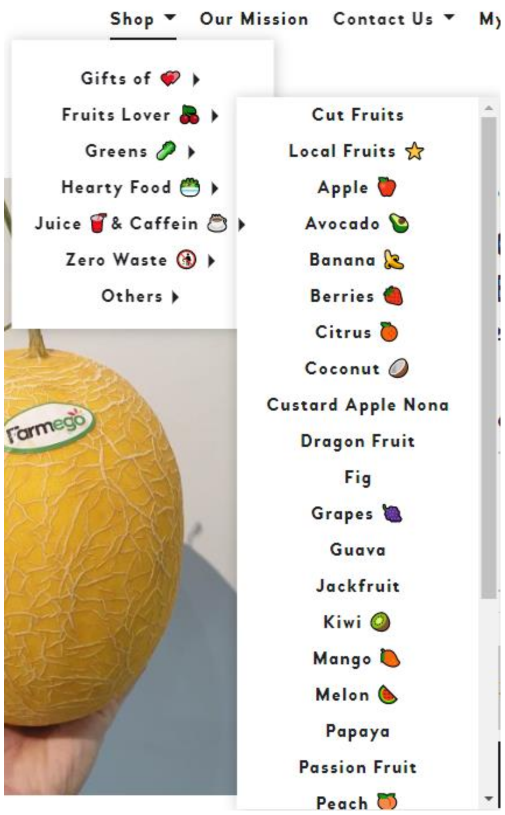


Figure 2.5.6 The available product categories

From the figure 2.5.7 below, customers are able to read all the information for the framego fruits. The special thing in this page is that they provide a textarea which can let users fill up their special requests. This special request information will be attached with the products if customers add the products into cart. This would be helpful and considerable to some people who may have special requests on the particular type of product like less sugar on the juice. In this website, customers are also able to add the products and place orders. For the user sign up and login module, they only provide a method for their newcomers to sign up which is through registering a new account on their website. As the product diversified into different types, they have implemented a search engine to ease the customers to browse and find a particular product via matching the keywords.

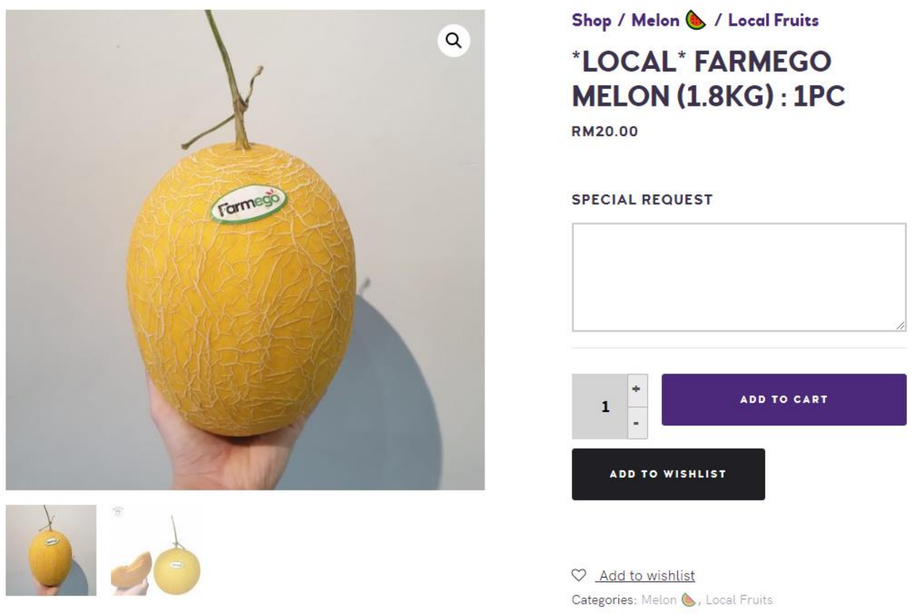

Shop / Melon / Local Fruits
LOCAL FARMEGO MELON (1.8KG) : 1PC
RM20.00
SPECIAL REQUEST
1
ADD TO CART
ADD TO WISHLIST
Add to wishlist
Categories: Melon, Local Fruits

Figure 2.5.7 The detailed information of farmego melon

Due to the fact that this is a smaller scale online fruit store, it only provides the important features basically just to let customers make orders on their websites. The only notable special feature is the wish list which allows customers to save their interested products into a list for themselves. Overall, it looks more like a minimalist online fruit store website in which the business owners do not want to overload their customers. The user interface appearance looks clean and simple which really attracts the people who do not like complex websites with bundles of features which are seldom used.

In contrast, Dusun may have some limitations or weaknesses which can be further improved in future. They do not implement any features which can personalise every customer account. For instance, they also do not include any recommender systems which can customise their customer needs. As a result, each customer will not have their own personalised shopping experience since the system will not recommend any products which they are more interested in. In short, they do not have a function which can strengthen the relationship with customers. They also do not introduce any membership module which can encourage more customers to purchase the products to gain the points to redeem any discounts. They require some features which can boost the two-way relationship in order to generate sales. Moreover, customers are not able to make payment via the website once they submit the order. This is because the order request needs to be confirmed by the business team in order to make sure the stock availability of the ordered products and customers only can make the transaction through the payment link via email. Prior to placing an order, the website also does not show the current stock available to inform the customers. Therefore, the shopping flow may need to be improved because it will be troublesome if the Dusun Online Fruit and Vegetable store does not have the sufficient inventory and they will need to reject the customer order request. This may affect customer shopping experience in their website and it is possible to lose their customers. This existing system also does not have the translation feature. Although translation is more like an optional feature, it would be recommended to include because it can increase the acceptability of this website to more customers.

2.5.3 PUR Cold Pressed Juice Online Store

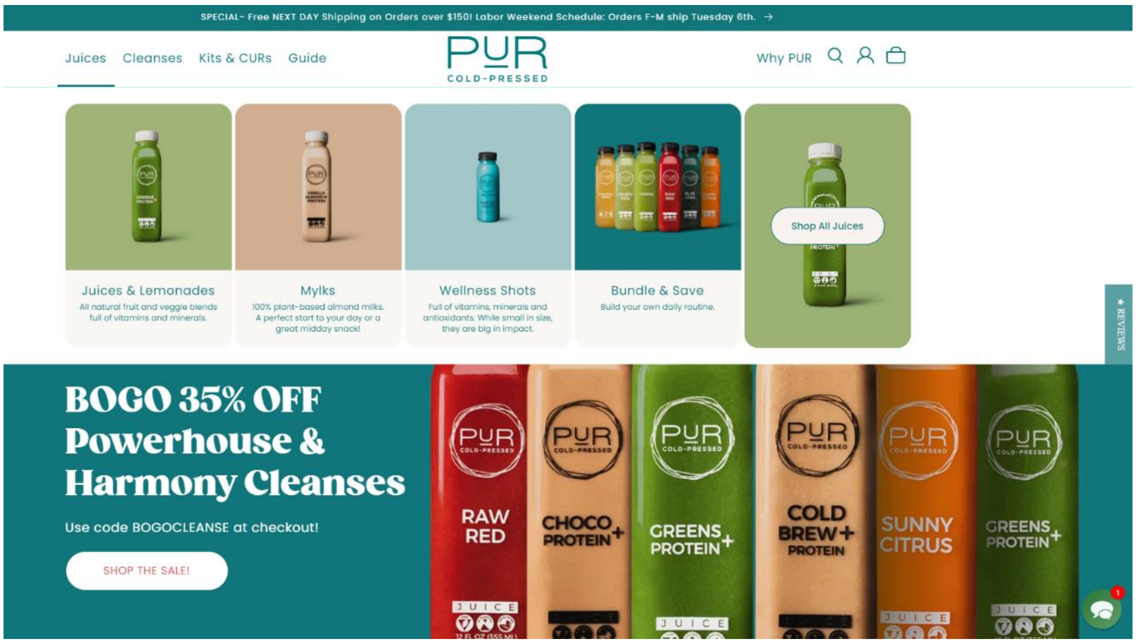
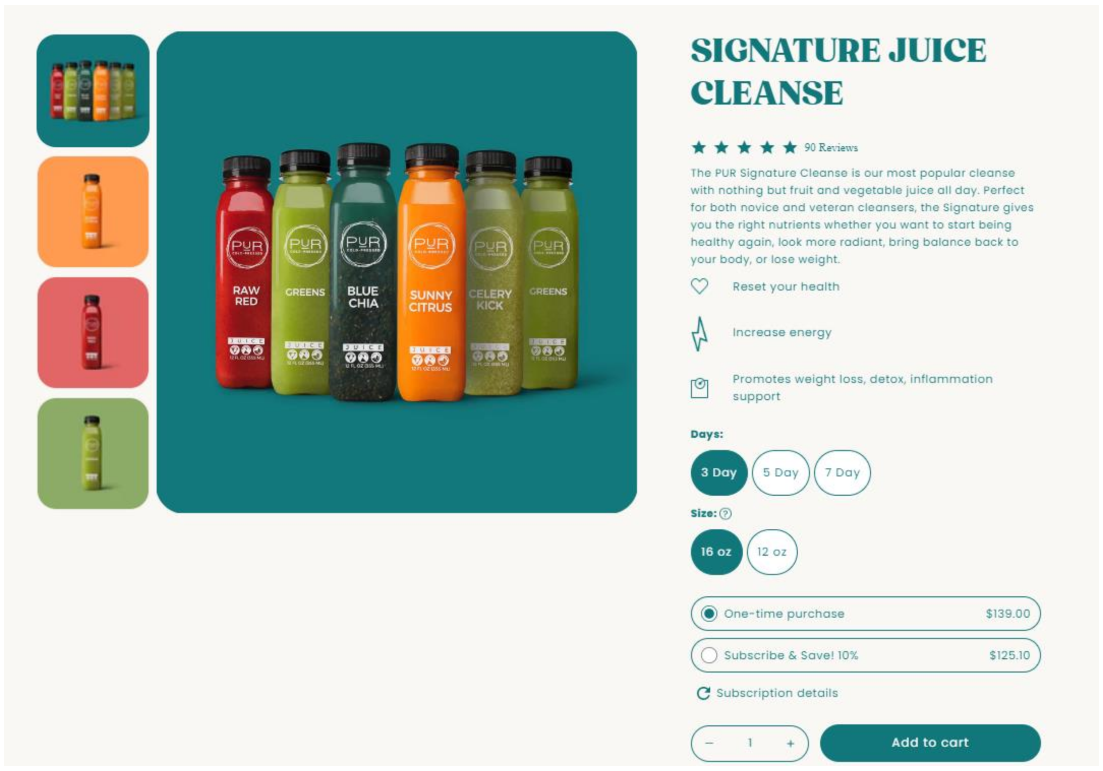

SPECIAL- Free NEXT DAY Shipping on Orders over $150! Labor Weekend Schedule: Orders F-M ship Tuesday 6th. →
Juices
Cleanses
Kits & CURs
Guide
PUR COLD-PRESSED
Why PUR
Juices & Lemonades
All natural fruit and veggie blends full of vitamins and minerals.
Mylks
100% plant-based almond milks. A perfect start to your day or a great midday snack!
Wellness Shots
Full of vitamins, minerals and antioxidants. While small in size, they are big in impact.
Bundle & Save
Build your own daily routine.
Shop All Juices
REVIEWS
BOGO 35% OFF Powerhouse & Harmony Cleanses
Use code BOGOCLEANSE at checkout!
SHOP THE SALE!
RAW RED
CHOCO PROTEIN+
GREENS PROTEIN+
COLD BREW+ PROTEIN
SUNNY CITRUS
GREENS PROTEIN+

Figure 2.5.8 PUR Cold Pressed Juice Online Store

PUR Cold Pressed Juice Online Store is an online business which focuses on selling cold pressed and bottled fruits and vegetables juices. This is a very modernist concept of fruit juices proposed by PUR. They preserve nutrients by using the cold pressing techniques. It is not the usual type of juices and the cold press is mainly to prevent the juices from being oxidised. However, the taste of the juices will still remain and the freshness is maintained even after a certain period without worry about the risk of losing the nutritional values.

First of all, this website has a minimalist and sleek interface design which will attract the first impression when customers first visit the website as shown in figure 2.5.8. Moving on to the feature part, PUR also does provide the basic features which are all the elements which must be obtained in a normal e-commerce website. For instance, the order module, signup and login module and product module. Customers can easily sign up as an account and login to kickstart their shopping. Besides, the customers are also able to use the search engine being provided by the system to find the interested products. As shown in the figure 2.5.8, the juices selling are being well-organised in their own label.


SIGNATURE JUICE CLEANSE

★★★★★ 90 Reviews

The PUR Signature Cleanse is our most popular cleanse with nothing but fruit and vegetable juice all day. Perfect for both novice and veteran cleansers, the Signature gives you the right nutrients whether you want to start being healthy again, look more radiant, bring balance back to your body, or lose weight.

Reset your health

Increase energy

Promotes weight loss, detox, inflammation support

Days: [✓ 3 Day] 5 Day 7 Day

Size: [✓ 16 oz] 12 oz

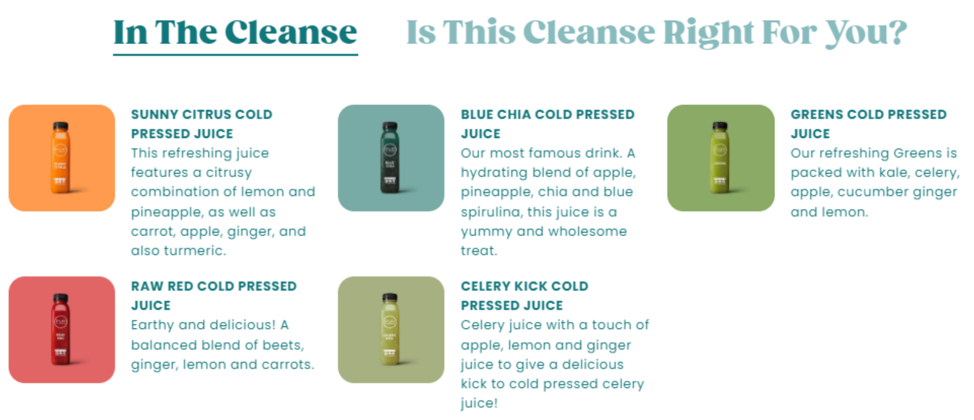
● One-time purchase $139.00

○ Subscribe & Save! 10% $125.10

Subscription details

− 1 + Add to cart

Figure 2.5.9 Detailed information of the signature juice cleanse


In The Cleanse   Is This Cleanse Right For You?

SUNNY CITRUS COLD PRESSED JUICE
This refreshing juice features a citrusy combination of lemon and pineapple, as well as carrot, apple, ginger, and also turmeric.

BLUE CHIA COLD PRESSED JUICE
Our most famous drink. A hydrating blend of apple, pineapple, chia and blue spirulina, this juice is a yummy and wholesome treat.

GREENS COLD PRESSED JUICE
Our refreshing Greens is packed with kale, celery, apple, cucumber ginger and lemon.

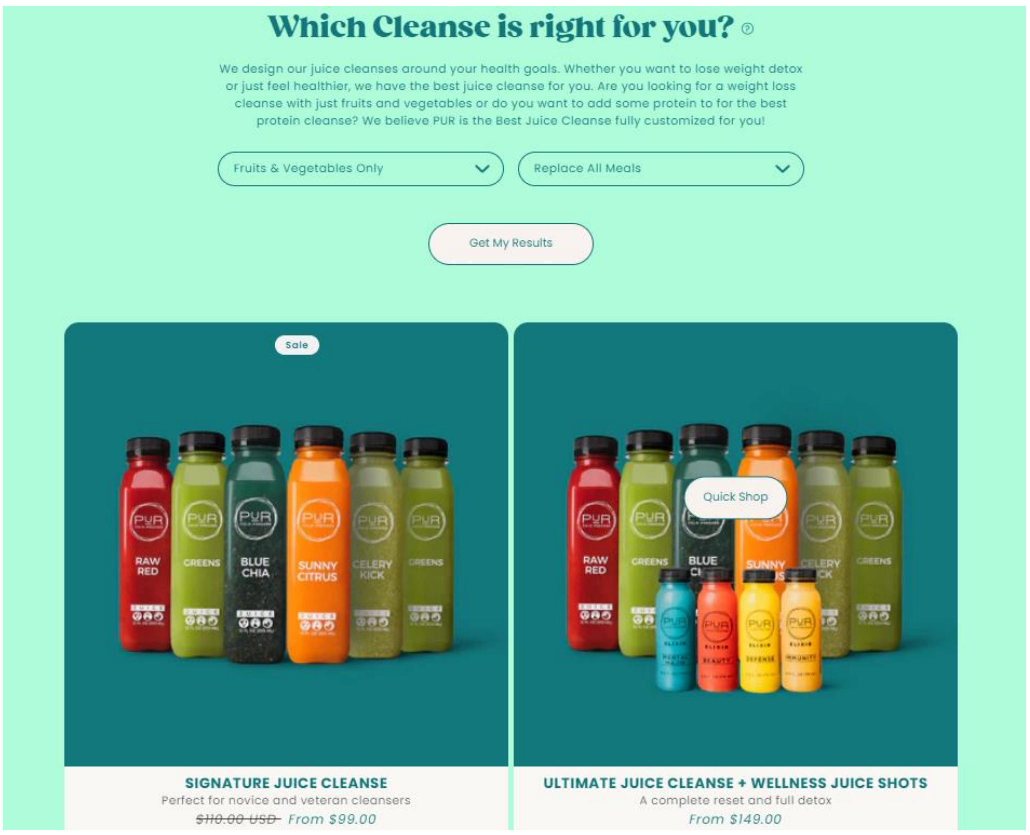
RAW RED COLD PRESSED JUICE
Earthy and delicious! A balanced blend of beets, ginger, lemon and carrots.

CELERY KICK COLD PRESSED JUICE
Celery juice with a touch of apple, lemon and ginger juice to give a delicious kick to cold pressed celery juice!

Figure 2.5.10 The description of all the juices in the package

Another notable strength of this system is that the business team are very concerned and considerate of their customers. The evidence is that all the products are written with their thorough description provided with the benefits that the human body gains after consuming the juices. All the nutritional content is listed properly for each bottle of juices including the main ingredients. This will greatly help the target customers and novice customers to easily get the products which best suit them. Similar to other existing systems, customers are able to add the items to their cart and proceed to check out their orders.

There are actually some special features which are noteworthy to be mentioned in this existing system. Predominantly, customers can earn extra points and some exclusive rewards every time they purchase something from the online shop once they register an account as a new member or during their birthday. PUR introduced the membership module which can attract new customers to join as a member and start to claim the referral rewards by inviting their family and friends. This is because the points can directly convert to cash discount in their next order which is really a captivating event to increase the customer engagement that will in turn boost their sales.



Which Cleanse is right for you?

We design our juice cleanses around your health goals. Whether you want to lose weight detox or just feel healthier, we have the best juice cleanse for you. Are you looking for a weight loss cleanse with just fruits and vegetables or do you want to add some protein to for the best protein cleanse? We believe PUR is the Best Juice Cleanse fully customized for you!

Fruits & Vegetables Only

Replace All Meals

Get My Results

Sale

SIGNATURE JUICE CLEANSE
Perfect for novice and veteran cleansers
~~$110.00 USD~~ From $99.00

Quick Shop

ULTIMATE JUICE CLEANSE + WELLNESS JUICE SHOTS
A complete reset and full detox
From $149.00

Figure 2.5.11 Recommendation for the suitable cleanse to customer

In addition, the system implements some sort of recommender system which will take in the customer preference and generate some results for the customers. Basically, these are the recommendations given by the existing system to the customers who want to get the right cleanse for themselves. It will generate the list of products which are customised according to the type of ingredients and type of cleanse which customers prefer. If the customers are satisfied with the products recommended, there will be a high possibility that these customers will revisit the website. Additionally, this existing system also provides a live chatbot which can increase the chance of communication with customers. This also helps the business to understand more on the customer behaviours and foster the customer loyalty.

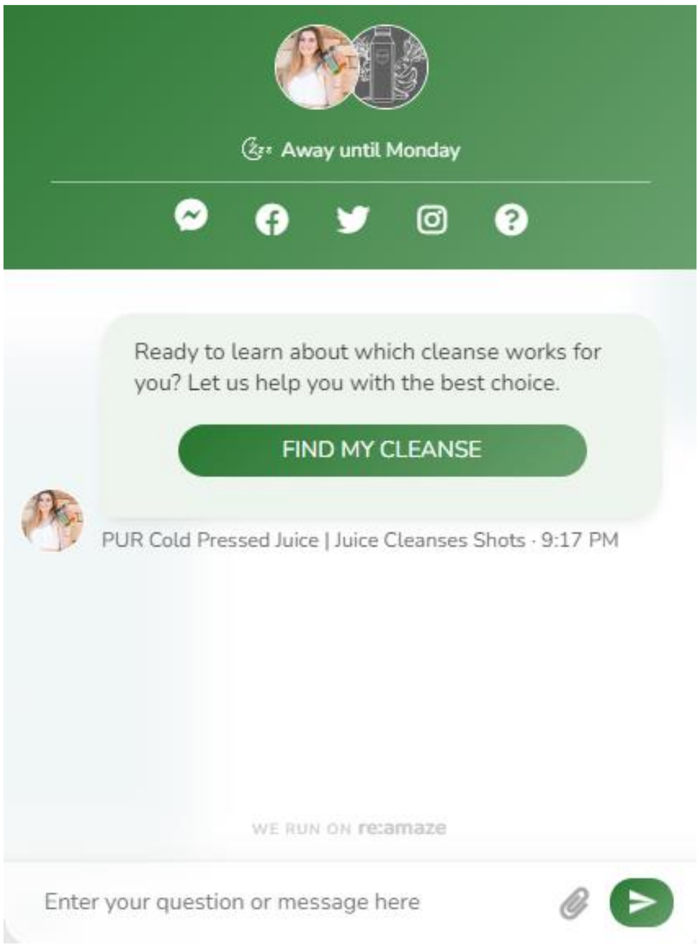

Away until Monday
Ready to learn about which cleanse works for you? Let us help you with the best choice.
FIND MY CLEANSE
PUR Cold Pressed Juice | Juice Cleanses Shots · 9:17 PM
WE RUN ON re:amaze
Enter your question or message here

Figure 2.5.12 Live chatbot

Conversely, there are still some minor limitations after reviewing this existing system. When customers are checking the products, the existing system does not have many useful filters like best rating, popularity, or most number of reviews. Without these handful filters, customers may not know which products are the hot and trendy products. Sometimes, people prefer to buy products which are more popular in terms of reviews and rating. On top of that, currently the online business only supports delivery inside the US which means local customers may not have the opportunity to enjoy the services and products. The existing systems also do not support multiple language translators since they do not target the customers from overseas.

2.6 Critical remarks

2.6.1 Comparison between algorithms used for recommender systems

Table 2.6.1 Table of comparison between algorithms used for recommender systems

	Content-based filtering	**Collaborative -based filtering**
Advantages	● Do not need to refer database order history. Only require product profile ● Provide recommendations based on products which are closely to the products which liked or or highly rated before ● Able to make adjustments in a short period of time ● Do not need to refer other profiles	● Able to provide recommendations even though they never rated these relevant products before ● Provide recommendations based on the group of neighbours of customer or products belongs to ● Perform well when hard to analyse content in user profile
Disadvantages	● Require much understanding and knowledge	● May suffer in data sparsity problems ● May suffer in cold-start problem

	● Content and user profiles need to be structured well-organised ● May suffer in overspecialization problem	● Require sufficient information about customers and products

2.6.2 Comparison between architecture used

Web and Business Tiers

Table 2.6.2 Table of comparison between Web and Business Tiers

	Amazon Web Services	**Microsoft Azure**
Advantages	● Dominant position cloud service provider ● Support multiple databases including relational database and non-relational database ● Supports the usage of different programming languages ● IaaS-based ● Popular web service - Elastic Compute Cloud (EC2)	● Another giant cloud service provider ● Support multiple databases including relational database and non-relational database ● Supports the usage of different programming languages ● PaaS, and SaaS based

	● Cheaper when size get increases ● AWS RDS can support up to 99.95% availability	● More price option for short term subscriptions plan ● Comprehensive microsoft documentation ● Azure SQL database can support up to 99.99% uptime ● Allows the deployed web applications to access the read-only version of the database
Disadvantages	● Hard to learn and use by novice users ● Overload features and options ● Complicated pricing (Charged per hour) ● Require the connection to the primary read/write to the database instance every time.	● Complex features ● Require training and learning or external support ● Complicated pricing (Charged per minute)

Data Tiers

Table 2.6.3 Table of comparison between Data Tiers

	Relational Database	**Non-Relational database**
Advantages	● Able to store structured data in the table format ● Able to retrieve data from multiple tables ● Reduces the data redundancies ● Promote data integrity ● Examples: MySQL, Oracle SQL and PostgreSQL.	● Does not need complex SQL query language to retrieve the data ● Use key or document ID to retrieve data ● Work well with large amount of data ● Shorter response time ● Examples: DynamoSQL, MongoDB
Disadvantages	● Structure and relationship between tables are way too defined and rigid ● Becomes complex when more data comes in	● Not suitable to store structured data ● Lack of standardisation

2.6.3 Comparison between full-stack frameworks used

Table 2.6.4 Table of comparison between between full-stack frameworks used

	MEAN stack	**ASP.NET full-stack**
Advantages	● Popular framework ● Dynamic programming language ● Easy to learn as compared to other framework ● Support NoSQL database ● Open source framework	● Popular e-commerce framework ● Produce dynamic pages ● Capable of running on other OS ● Better performance ● Support the data structure and algorithms ● Open source framework ● Work well with Microsoft Azure platform
Disadvantages	● Not the best choice to deal with large scale of data ● Do not have proper JavaScript documentation [16]	● Difficult to learn ● Costly

2.6.4 Comparison between similar existing systems

Table 2.6.5 Table of comparison between between similar existing systems

Functionalities	**MBG Online Fruit Shop**	**Dusun Fruits,Vegetables and Cold Pressed Juice Online Store**	**PUR Cold Pressed Juice Online Store**
Make order	✓	✓	✓
Check product availability	✓		
Wishlist	✓	✓	
Add items to cart	✓	✓	✓
Register and login	✓	✓	✓
Make payment	✓	✓	✓
Search engine	✓	✓	✓
Membership feature	✓		✓
Newsletter sign up	✓		✓
Translation	✓		
View Order History	✓	✓	✓

Chatbot			✓
Product filters	✓	✓	✓
*Recommender system			✓

2.7 Proposed Solutions

This project aims to propose an online fruit and fruit juice web application which will equip all the basic features in the online e-commerce website. On top of that, the proposed system will also implement a special recommender system based on the various filtering techniques like collaborative-based filtering to study the user preference. It requires effort to train and test model. After deploying the model, the proposed system or web application will be able to provide some recommendations to them.

The database being used for the e-commerce online store will be a relational-database like SQL to store structured data which is powerful in Azure and able to support a lot of API. Main SQL tools will be used are Microsoft SQL Server in Microsoft SQL Server Management Studio (SSMS) and the Azure SQL database. The advantages:

- Easy to use as desktop itself can serve as the SQL server
- Supports both GUI and query for database operation
- Support database CRUD operation through entity-framework using migration
- Make sure the web application is deployed to the Internet and saving the data globally
- Quick process for the setting and configuration of the data storage in Azure SQL database and
- Easy for a web application to go online using the Azure services.

After the review and some consideration, the proposed system will be built using the ASP.NET MVC full-stack framework and deployed in Microsoft Azure at the end of development because ASP.NET MVC framework provides these benefits:

- Achieving the clear concern of segregation
- It provides the approach for developers to create dynamic websites
- It provides developers the full control on the web application
- It will also ease the process of testing as the complexity has been divided accordingly to model, view, and controller
- It helps the .NET developer to easily detects the bugs and solve it using MVC framework

CHAPTER 3

System Methodology/Approach

In this chapter, there will be explanations on system design specifications, system design flow diagrams and the system architecture diagram. Besides, system deployment diagram and project planning (Gantt chart) are included as well.

3.1 System Design Specification Diagram

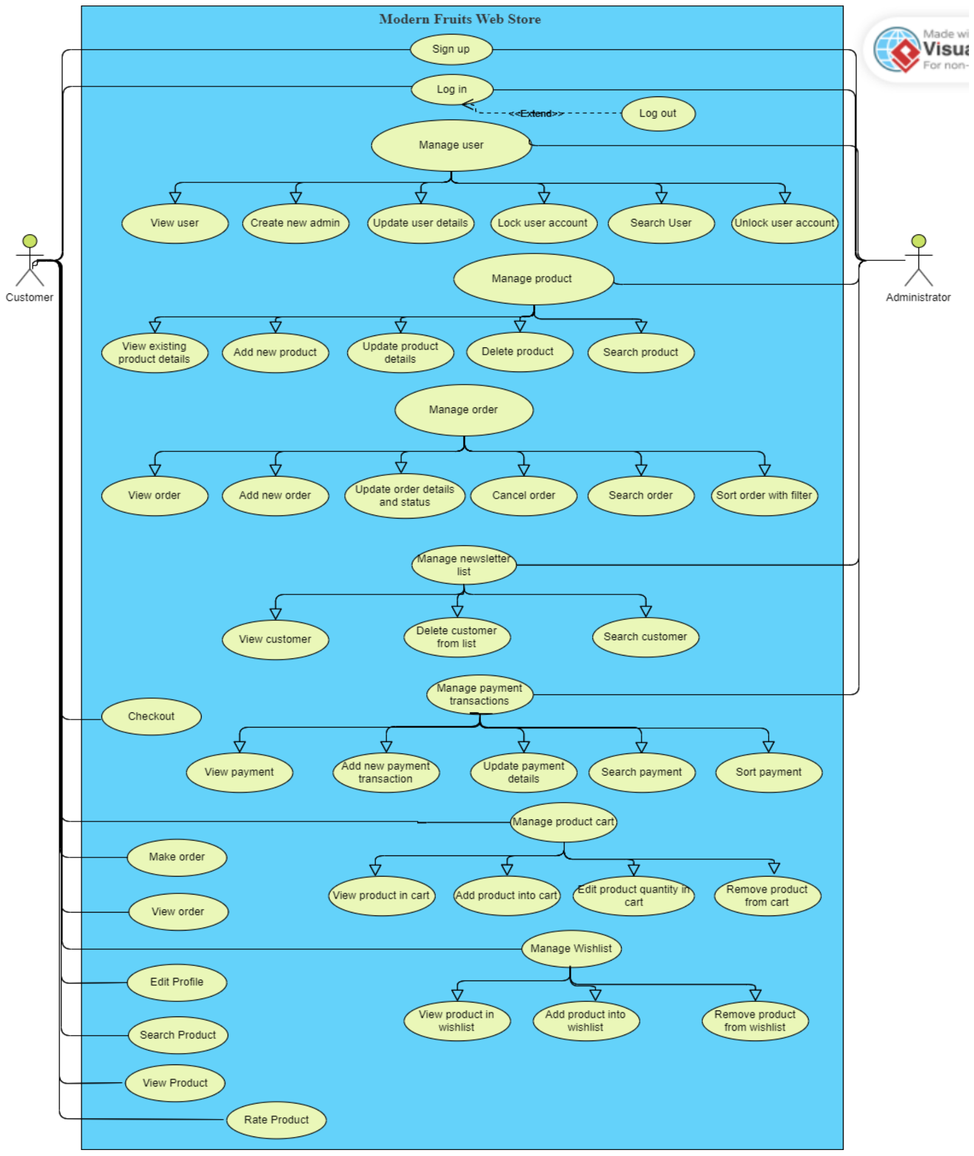


Figure 3.1.1 Use case diagram of Modern Fruit Web Store

According to Figure 3.1.1 above, the modern fruit web store is mainly proposed to server the customers and administrators. There are certain differences between them in terms of what action they are authorized to do. Firstly, customer need to sign up or register prior to start to perform any action in the website. After the system verify the account, customers are successfully registered and able to log in to the web application with their own profile. Then, they can proceed to check their own profile or try to edit their personal account information. For instance, they can complete and fill up their account information like address, phone number and personal details. Besides, they can also be able to change their password any time after login successfully. Next, they can have a view on the website especially all the products. They can also search the product name or find product based on product categories or filter. If customers are interested to buy the product, they can add the product into shopping cart. In the shopping cart, add product, edit product details, and remove product action are applicable. They can also add the product into a noteworthy feature called wish list which only stores the product they may be interested to review again next time when they return to this website. It will ease them to directly approach the product in the wish list rather than searching the product. They can remove product from the wish list if they wish. After customer makes decision to buy, they can check out the cart and place order. When the order is made successfully, they can proceed to review their order details and status.

Moving on to the administrator section, they are the user who own the top-level privilege to gain to all data typically all records generated in respective table like customer, order and product. At first, administrators need to log in the system with their login credentials which already credited with the highest role. With the given role, they can perform business management and monitoring which will give the administrators an overview of the business. Administrators can manage the users, orders, products, customers, newsletter list and payments. In further detail, they can perform view, add, edit, and delete operation on these modules. For instance, administrators can add a new product and display on the website for selling. They can update the product description or other associated details anytime. Apart from that, they can search and sort the records accordingly to filter and find records. For the user table, administrator can also perform and lock and unlock operation for active and inactive customer account. If the account is under locked status, this indicates that the account is being suspended and customer will not be able to log in using the account anymore unless it is being unlocked.

3.2 System Design Flow Diagram

3.2.1 Normal user/Customer system design

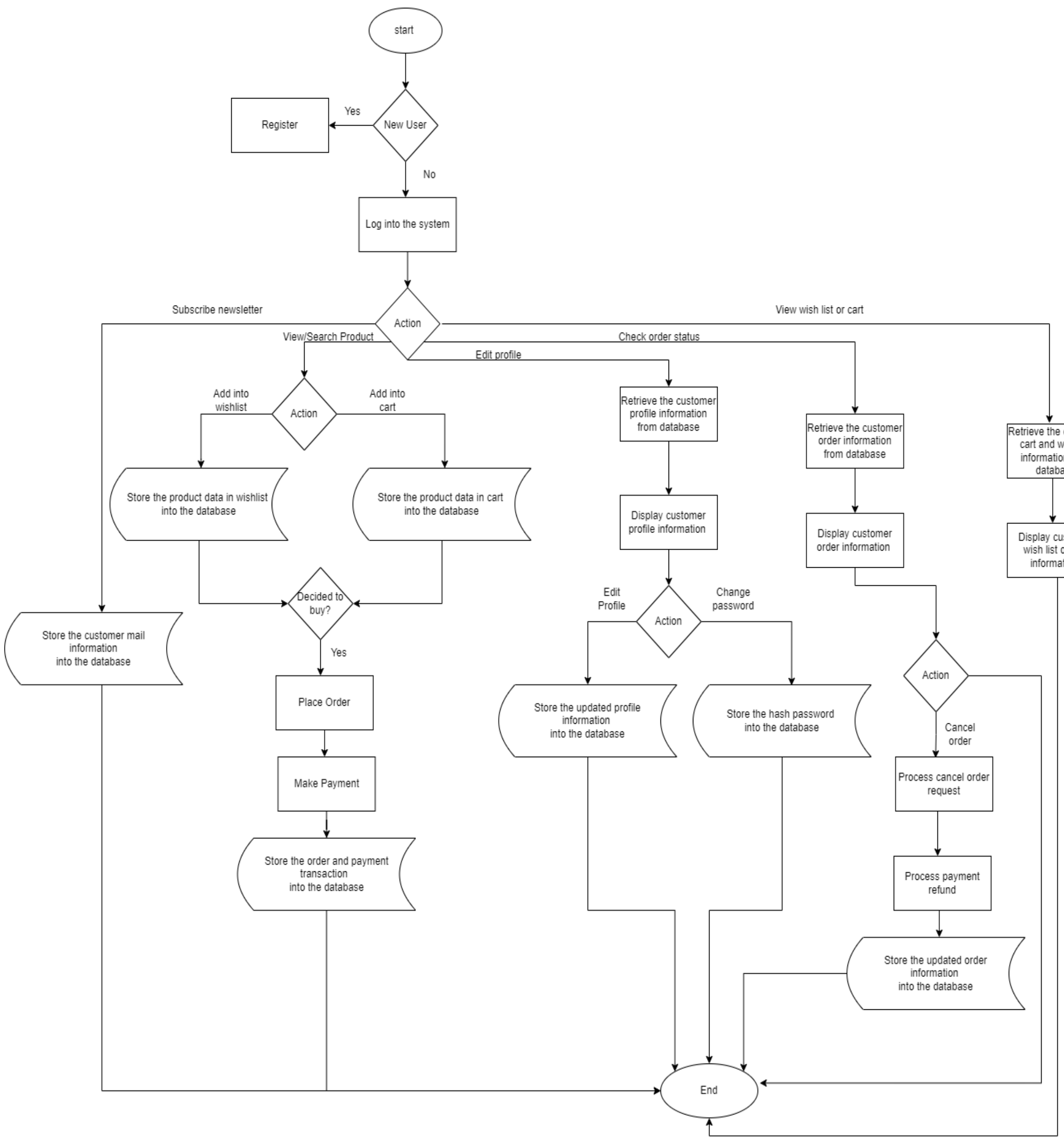

start
New User
Yes
Register
No
Log into the system
Action
Subscribe newsletter
View/Search Product
Edit profile
Check order status
View wish list or cart
Add into wishlist
Action
Add into cart
Store the product data in wishlist into the database
Store the product data in cart into the database
Decided to buy?
Yes
Place Order
Make Payment
Store the order and payment transaction into the database
Store the customer mail information into the database
Retrieve the customer profile information from database
Display customer profile information
Edit Profile
Action
Change password
Store the updated profile information into the database
Store the hash password into the database
Retrieve the customer order information from database
Display customer order information
Action
Cancel order
Process cancel order request
Process payment refund
Store the updated order information into the database
Retrieve the customer cart and wish list information from database
Display customer wish list or cart information
End

Figure 3.2.1 Normal user/ customer system design flowchart

In this section, there are overall system design flowchart to briefly explain about the flow of using the web application. According to Figure 3.2.1, user or customer who comes to this website can have the option to register or log in before place any orders. For newcomers, they need to register a new account by filling up the required information. After register, they can use the account credentials and login to the website.

The following action can be either choose to view and search products or edit profile. For the first option, in the product page, customers can choose to add product that they are interested to buy into the shopping cart or wish list as they wish. Once the action is performed, the product information will be stored so that the product information in the cart and wish list can be restored for display whenever customer or user login to the website. Once they decided to purchase the products, they can check out the shopping cart and place order. The last step before order is made is that customers must make payment via payment integration.

After that, the order information and payment transaction will be carried and stored into the database. On the other hand, customers can check and review their profile. System will render and fetch the user information like username, contact number, address in the profile page. They can perform update or edit operation in their profile or change password if they opt for another password. Lastly, system will store the updated profile information and new hash password into database.

Besides, customers can also view their wish list or cart in which they had added some products before. The web application will retrieve the cart and wish list information from database for each customer session. In other case when customer had placed any order, they can check their order status whether it is processing or shipped. Before the product is being shipped, customer still own the option to cancel order on any acceptable reason. The system will process the cancel order request and refund the payment amount to customer.

3.2.2 Administrator system design

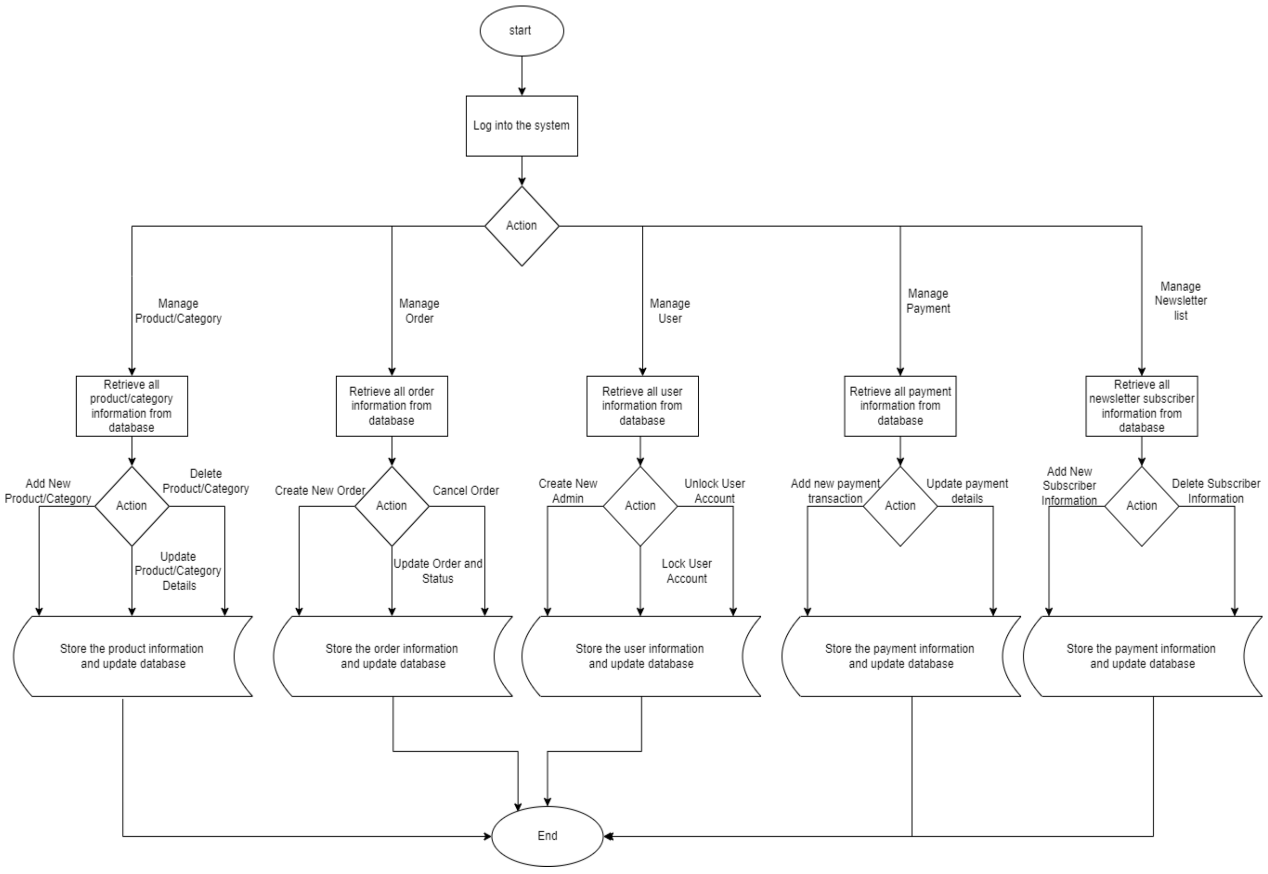

start
Log into the system
Action
Manage Product/Category
Manage Order
Manage User
Manage Payment
Manage Newsletter list
Retrieve all product/category information from database
Retrieve all order information from database
Retrieve all user information from database
Retrieve all payment information from database
Retrieve all newsletter subscriber information from database
Add New Product/Category
Delete Product/Category
Update Product/Category Details
Create New Order
Cancel Order
Update Order and Status
Create New Admin
Unlock User Account
Lock User Account
Add new payment transaction
Update payment details
Add New Subscriber Information
Delete Subscriber Information
Store the product information and update database
Store the order information and update database
Store the user information and update database
Store the payment information and update database
Store the payment information and update database
End

Figure 3.2.2 Administrator system design flowchart

Referring to the Figure 3.2.2, administrator may need to login to the system before performing any operation. The system flow is clear and concise after they logged in. There are several options for action which can be done. This is based on what table the administrator intends to manage. They can choose either to manage the products that are currently selling on the website or manage orders, users, and payments transactions. The system is responsible to retrieve and display the requested table information. For product part, administrator is authorized to add new product, edit product information or delete the product from the database. The administrator is also capable to create new order manually, update order status or details and cancel order manually. Next, the administrator can also manage all the users in the table in the operation of create new administrator, lock and unlock the user account if there are special circumstances. Usually, only administrator can view the user database though. Moreover, the administrator can always monitor the payment transaction conducted every day. It is also possible to have scenarios in which administrator may need to add or update any payment like the status and he or she is able to make it. Last but not least, the administrator can also have the option to manage the newsletter list which consists of all the users who have subscribe the newsletter. They are able to add or delete the subscriber from the newsletter.

3.2.3 Recommender system design

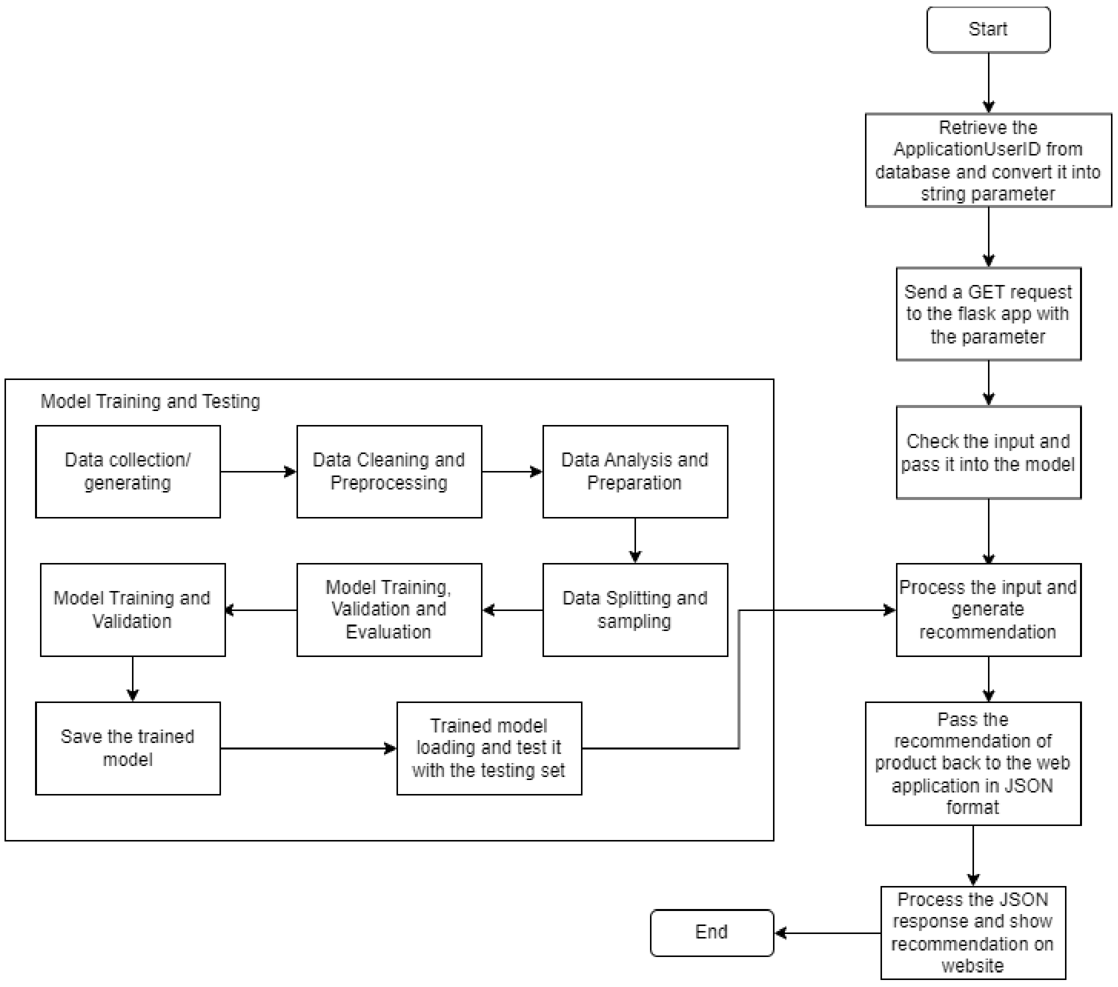

Start
Retrieve the ApplicationUserID from database and convert it into string parameter
Send a GET request to the flask app with the parameter
Check the input and pass it into the model
Process the input and generate recommendation
Pass the recommendation of product back to the web application in JSON format
Process the JSON response and show recommendation on website
End
Model Training and Testing
Data collection/ generating
Data Cleaning and Preprocessing
Data Analysis and Preparation
Data Splitting and sampling
Model Training, Validation and Evaluation
Model Training and Validation
Save the trained model
Trained model loading and test it with the testing set

Figure 3.2.3 User-based collaborative filtering system design flowchart

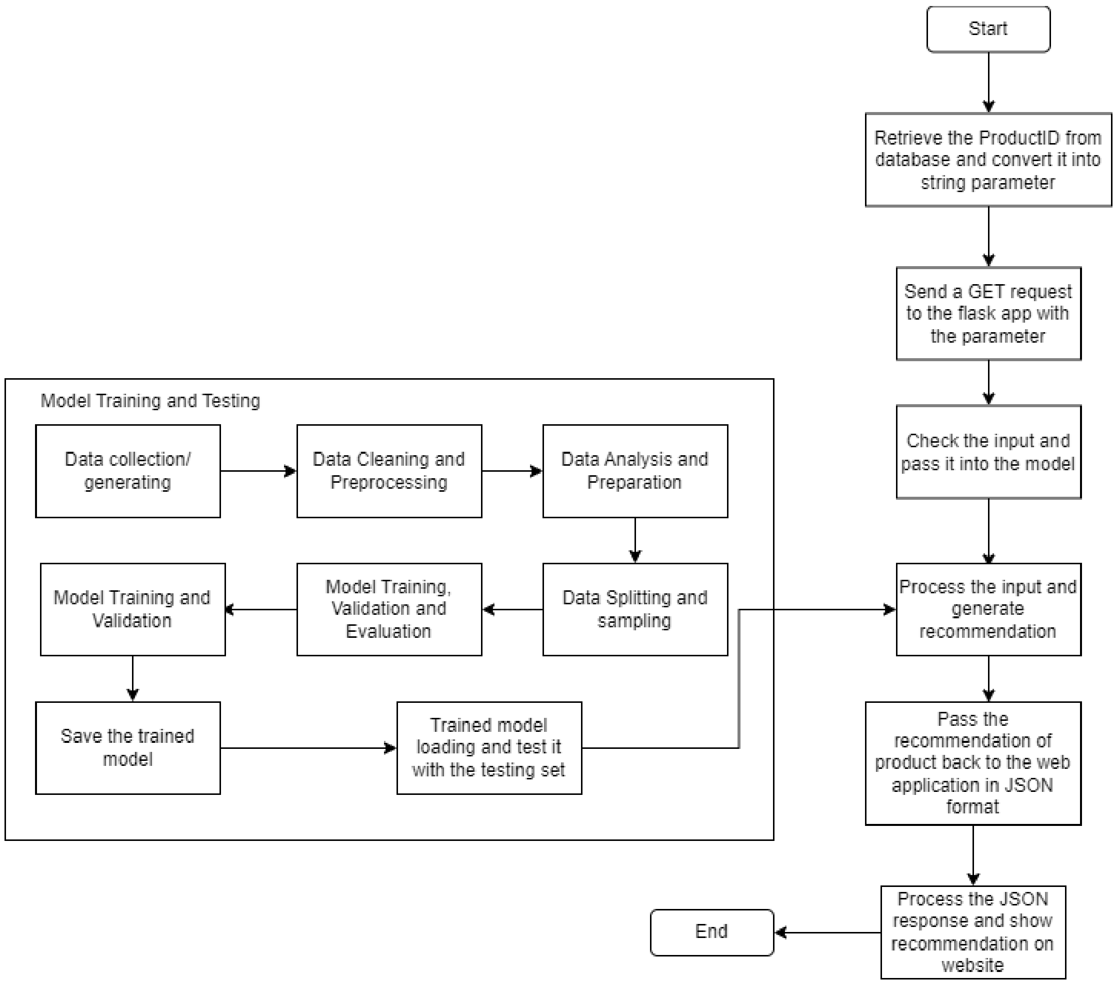

Start
Retrieve the ProductID from database and convert it into string parameter
Send a GET request to the flask app with the parameter
Check the input and pass it into the model
Process the input and generate recommendation
Pass the recommendation of product back to the web application in JSON format
Process the JSON response and show recommendation on website
End
Model Training and Testing
Data collection/ generating
Data Cleaning and Preprocessing
Data Analysis and Preparation
Data Splitting and sampling
Model Training, Validation and Evaluation
Model Training and Validation
Save the trained model
Trained model loading and test it with the testing set

Figure 3.2.4 Item-based collaborative system design flowchart

The Figure 3.2.3 shows the overall recommender system design in terms of design flow. The left-hand side shows how the user-based collaborative filtering model is being trained and tested. The model will then be deployed in the Flask application and wait for the request input. Initially, the web application will retrieve the ApplicationUserID from the database and convert it into string parameter. The noteworthy point is that the ApplicationUserID is only retrieved when the customer start rating the products. The MVC web application will send a GET request to the flask app with the parameter(‘ApplicationUserID’). The input will be checked in terms of the type and format before passing to the model. The model will perform the prediction and generate recommendations. Once the prediction is done, the product recommendations will be passed back to the web application in JSON format. Once the web application received the response from the flask application successfully, the JSON response will be parsed and the recommended product information will be retrieved from the database and displayed on customer's website. The difference in terms of the design flow between item-based and user-based collaborative filtering is the input. For item-based, the ProductId which is the ID of a product that had been rated by customer will be passed as the string parameter as shown in Figure 3.2.4.

3.3 System Architecture Diagram

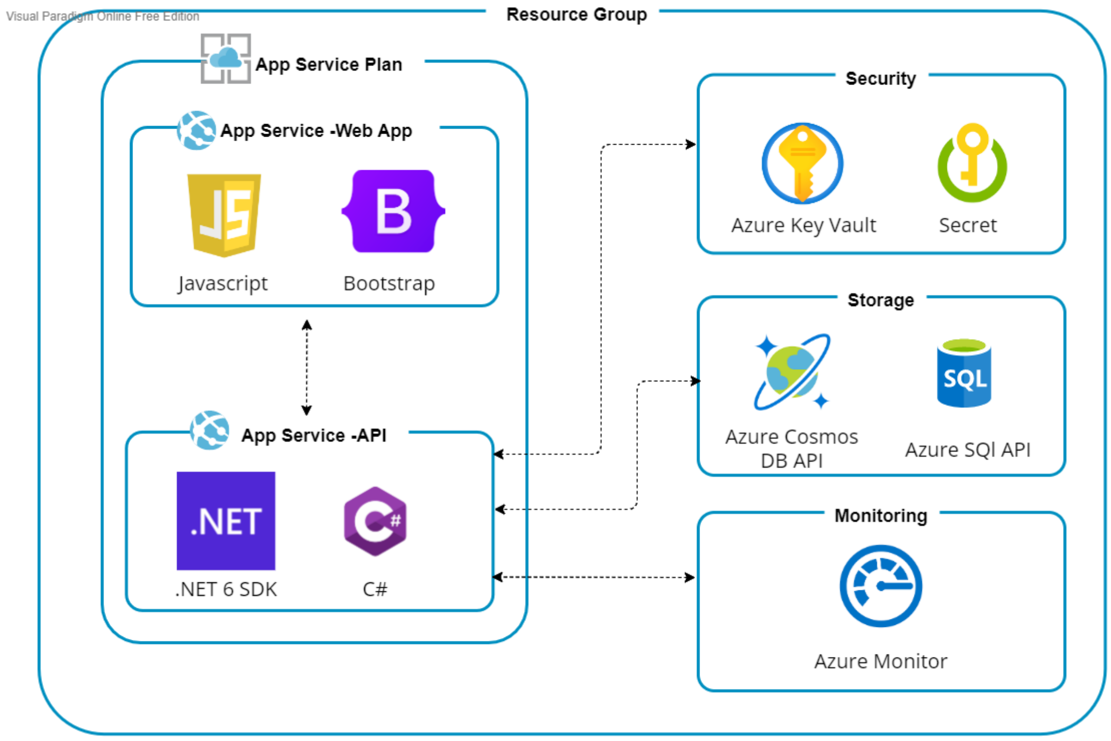


Figure 3.3.1 Architectural Diagram for Modern Fruit Web Store

In the section, there is a discussion on system architecture diagram. As show in Figure 3.3.1, it shows the overall system architecture and all the required components when deploying an e-commerce web application in Azure. Firstly, it is necessary to group our resources like storage, web application, security and API components into an organized resource group for better management. Prior to that, a subscription should be made. Next, app service plan is required to host our web application or web app in the Azure Service environment. In this case, only free and shared service plans will be implemented mainly to serve the purpose of development and testing purposes. Inside our app service, there are client-side and server-side components and API to bunch up the whole full-stack application. For instance, C# for .NET SDK 6, JavaScript. Next, our back-end API will connect with storage, security and monitor components when hosting on the Internet. In terms of security, Azure Key Vault plays the role to securely store and manage the configuration secrets and encryption keys for the server apps. It is usually not accessible by client-side [17]. Moving to the storage part, database API like SQL and Cosmos DB API are being implemented to store different type of data, namely structured data and unstructured data. Moreover, the web application after publishing to Internet will be monitored by Azure Monitor. Azure Monitor is there built to monitor the health, availability and performance of the application [18]. With the monitoring by Azure Monitor, factors that affecting the performance and resources can be significantly identified.

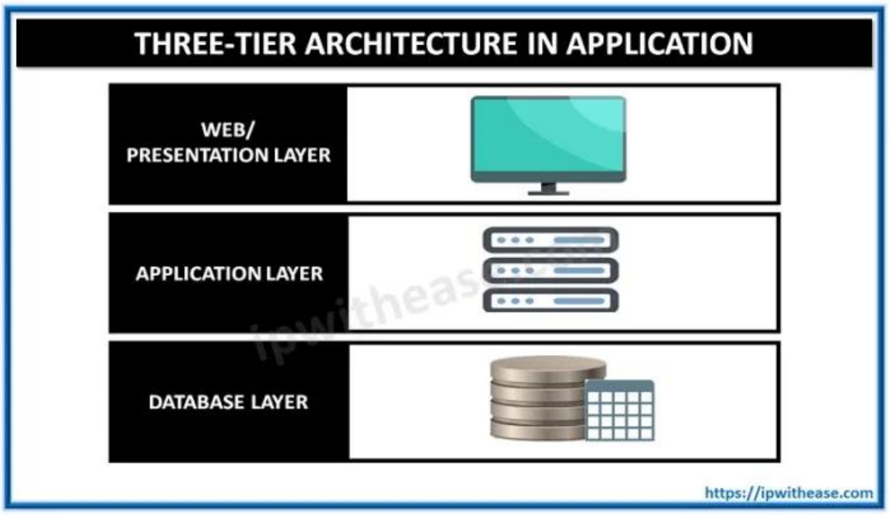


Figure 3.3.2 Three-tier architecture of web application

Generally, the architecture can be divided into 3 main tiers which are the presentation layer, application layer and database layer just like the Figure 3.3.2. Firstly, the customer web interface belongs to the presentation layer that they can interact with the application. In the proposed system, the first tier will be running on the customer's browser. Customers can browse to the website using their browser and perform some action. For instance, they can view their account profile or add the product into cart via the static and dynamic content shown. The second layer is the application layer where the web app will be deployed at. All the communication directs this middle layer as the presentation layer cannot directly reach the database layer. To relate with the e-commerce website, the application server which is the Azure Web App will host the web application and perform any CRUD function to retrieve any data from the data layer. This is also the part where ASP.NET (C#) will be used for the application development. For instance, the order details of a particular customer can be retrieved from the database in order to display in the customer site. The last layer is the data layer. It is the database server which mainly handles the data processing and data storage. It is a compulsory layer to provide web applications a place to store the product, customer and order information. In the proposed system, the option chosen for the database are the Cosmos DB which is the non-relational database and SQL database which is relational database.

3.4 System Deployment Diagram

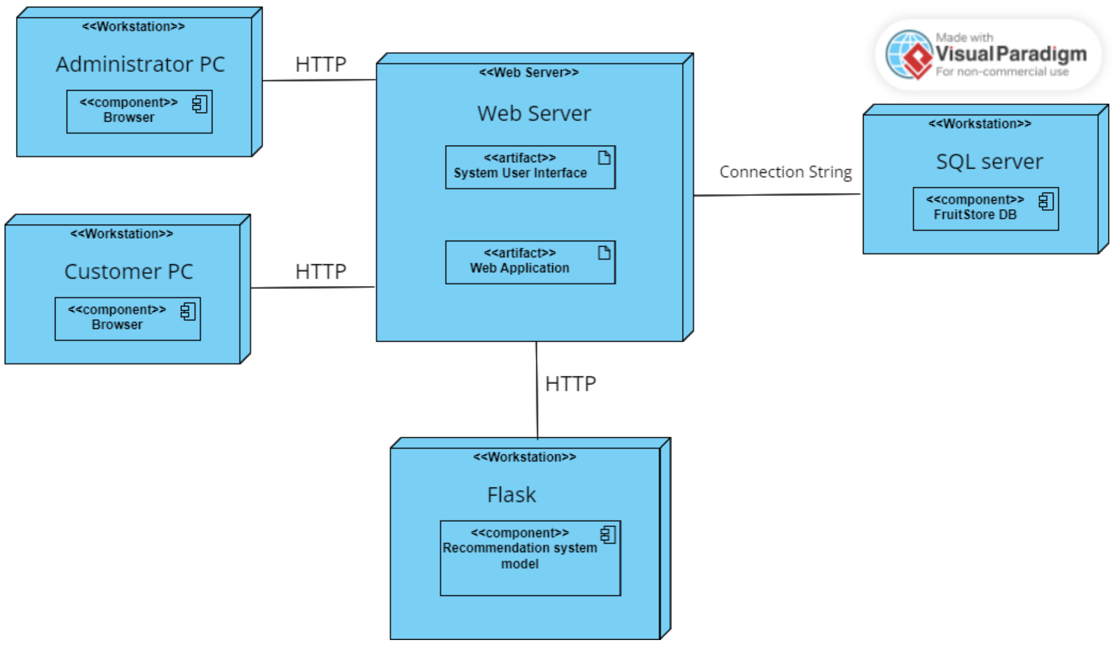


Figure 3.4.1 System Deployment Diagram

The current system will implement a client-server architecture and operate mainly around the client and server module. The administrators and customers will be main clients of the system. Customers and administrators can use their browser to reach the web server through HTTP request. The clients of system will need a database server to store the respective data information and perform any data operation like data retrieving. SQL server is the option being chosen as the data storage. All the application logic will also be being processed and handled in the web server. It acts as the middleman to handle all the traffic and communications between the clients, SQL server and Flask application. Flask will be the actual location to deploy and run the recommender system model. The JSON input will be passing from the web server through HTTP GET request and return the JSON response (recommendations) to the web server for further processing.

3.5 Project Planning

3.5.1 FYP1 Gantt chart timeline

Table 3.5.1 FYP1 Gantt chart timeline

Activities/Task	**Week**						
	1	**2**	**3**	**4**	**5**	**6**	**7**
Brainstorm on project ideas and understanding Azure	**X**						
Understand, install, configure the required software tools and framework for developing the web application	**X**						
Self-learning on Asp.Net core 6 MVC framework and C# language	**X**	**X**					
Finalising the required workload and output of the system for this trimester		**X**					
Self-learning on Entity Framework 6 Code First Approach to deal with back-end database and servers-side components in MVC		**X**	**X**				
Register Microsoft Azure Fundamentals Certification Training and self-learning on Microsoft Azure learning-path as requested by organizers			**X**	**X**			

Azure Exam Certification preparation and revision			**X**	**X**			
Program first product CRUD module using Migration, self-learning on ASP.NET Core Identity				**X**			
Attend online Microsoft Certification exam and program the authentication module					**X**		
Program dynamic product single page, role-based authorization					**X**		
Testing the system					**X**	**X**	
Finalising contents and writing the full project report						**X**	
Presenting the contents of the FYP1 report							**X**

3.5.2 FYP2 Gantt chart timeline

Table 3.5.2 FYP2 Gantt chart timeline

Activities/Task	**Week**													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Finalising the required workload and output of the system for this trimester	**X**													
Program email verification and forgot password module and customer profile	**X**													
Program user management CRUD module		**X**												
Program product category and shopping cart module			**X**											
Program product wish list module			**X**											
Program search engine module				**X**										
Program newsletter module				**X**										

Program make order, cancel order module					**X**									
Program order CRUD module					**X**	**X**								
Program payment module						**X**								
Understand and familiarize with different type of recommender systems							**X**							
Compare the pros and cons of recommender systems implemented with different techniques							**X**							
Design and program the recommender system							**X**	**X**						
Deploy the recommender system into Flask and testing the overall web application								**X**	**X**					
Improve and wrap up the web application										**X**				
Publish the web application and database to Azure with appropriate configuration										**X**				
Finalising contents and writing the full project report											**X**	**X**		
Presenting the contents of the FYP2 report													**X**	

CHAPTER 4

System Design

In this chapter, there will be some diagrams and explanations to illustrate the system design. For instance, system block diagram in the first part, system components specifications that focus on the components in the proposed web application. Besides, there are system components interaction operations which illustrate the interaction between the components. Other than that, the database ERD diagram is given to illustrate the relationship between the tables exist in the database.

4.1 System Block Diagram

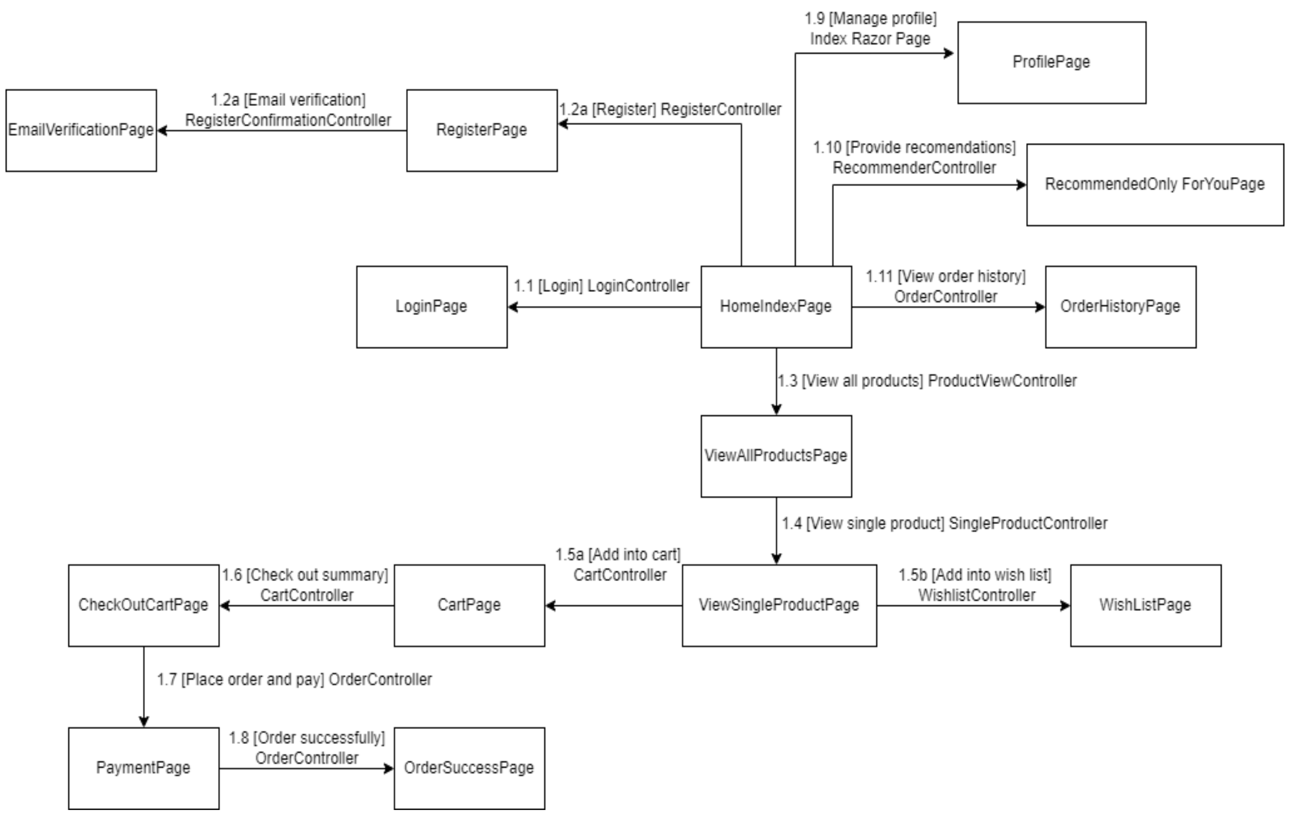

1.9 [Manage profile] Index Razor Page
ProfilePage
1.2a [Email verification] RegisterConfirmationController
EmailVerificationPage
RegisterPage
1.2a [Register] RegisterController
1.10 [Provide recomendations] RecommenderController
RecommendedOnly ForYouPage
LoginPage
1.1 [Login] LoginController
HomeIndexPage
1.11 [View order history] OrderController
OrderHistoryPage
1.3 [View all products] ProductViewController
ViewAllProductsPage
1.4 [View single product] SingleProductController
CheckOutCartPage
1.6 [Check out summary] CartController
CartPage
1.5a [Add into cart] CartController
ViewSingleProductPage
1.5b [Add into wish list] WishlistController
WishListPage
1.7 [Place order and pay] OrderController
PaymentPage
1.8 [Order successfully] OrderController
OrderSuccessPage

Figure 4.1.1 Customer module block diagram

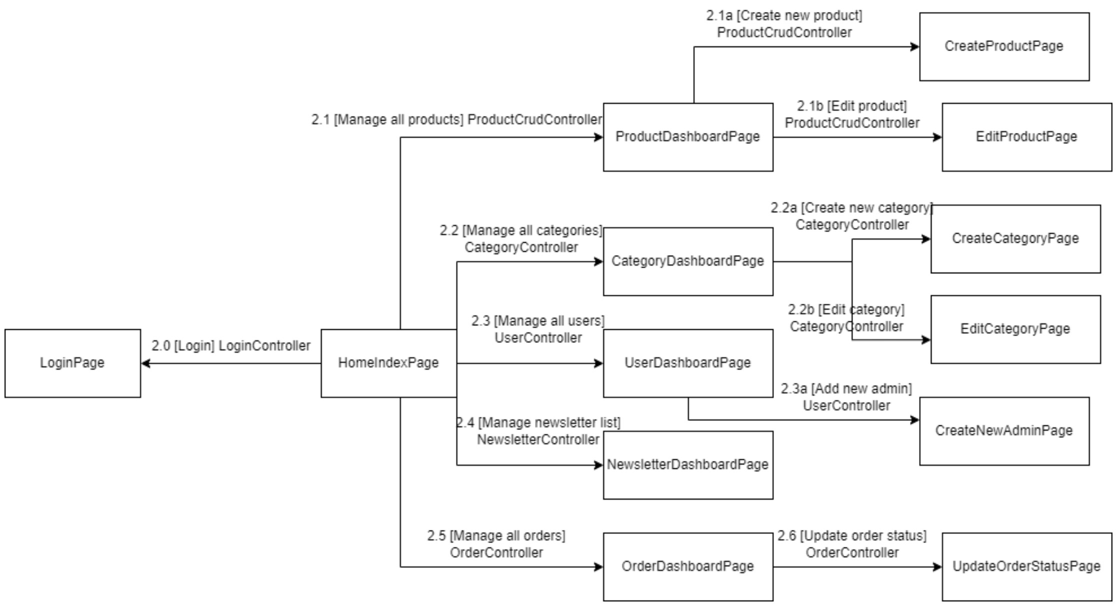

2.1a [Create new product] ProductCrudController
CreateProductPage
2.1 [Manage all products] ProductCrudController
ProductDashboardPage
2.1b [Edit product] ProductCrudController
EditProductPage
2.2a [Create new category] CategoryController
CreateCategoryPage
2.2 [Manage all categories] CategoryController
CategoryDashboardPage
2.2b [Edit category] CategoryController
EditCategoryPage
LoginPage
2.0 [Login] LoginController
HomeIndexPage
2.3 [Manage all users] UserController
UserDashboardPage
2.3a [Add new admin] UserController
CreateNewAdminPage
2.4 [Manage newsletter list] NewsletterController
NewsletterDashboardPage
2.5 [Manage all orders] OrderController
OrderDashboardPage
2.6 [Update order status] OrderController
UpdateOrderStatusPage

Figure 4.1.2 Administrator module block diagram

4.2 System Components Specifications

4.2.1 Customer Module

HomeIndexPage

This is the main home page for the web application when user first reach to the website.

LoginPage

This is the login page which allows customers to login to the web application using their account credentials. The system will retrieve customer data and check the credentials. It also redirects users to a home page or once successfully login or login failed when the email or password is incorrect.

RegisterPage

This page is responsible for new customers who have yet to register their user account in the web application. Once users have registered, their account information will be stored in the database so that they can login using the same credentials in the future.

EmailVerificationPage

This page is to request customers to verify their email account before they can successfully register the account. They can also request email verification resend if the timeout is reached.

ViewAllProductsPage

This page will display all the products which are currently selling on the web application. Customers can filter the products according to category or directly search for the products using keyword search.

ViewSingleProductPage

This page will display the single product which customers have clicked into it. Customers can rate the product, add the product to their wish list or the cart. There will be some recommendations suggested to customers through item-based collaborative filtering model if they have rated the particular product before.

CartPage

This page will display all the products which customers have interested in and added into cart. Inside the cart, customers can either add or reduce the quantity of product or they can directly remove it from the cart. If they have decided to place an order, they can click the checkout button to proceed to the checkout summary.

WishListPage

This page will display all the products which customers have interested in and added into the wish list. Inside the wish list, customers can either add or remove any product. If they wish to view the single product detail, they can click the product and redirect to the single product page.

CheckOutCartPage

This page will display the order summary and require customers to fill in the personal and shipping information before placing the order. The system will calculate the total amount to pay and display the subtotal price for each product. After everything is filled up, customers can proceed to click the place order button and make the payment.

PaymentPage

This page requires customers to fill up their email, credit/debit card number with its associated information to make payment.

OrderSuccessPage

This page will display order successfully once customers successfully made the payment. It will display the order ID and customers can check the order status later. An email will be sent to the customer's phone to inform them that the order has been successfully placed.

ProfilePage

This page will display all the customers' personal information and customers are able to edit all these details and save them into database. The system will retrieve it and automatically display the customer detail on the field without customer input next time before they want to place order. They also can change their password, add 2FA authenticator and some settings in the customer profile.

RecommendedOnlyForYouPage

This page will generate and display some product recommendations to customers through user-based collaborative filtering model if they have rated some product before.

OrderHistoryPage

This page will display all the order details for a particular customer after he/she has placed the order.

4.2.2 Administrator Module

HomeIndexPage

This is the main home page for the web application when user first reach to the website.

LoginPage

This is the login page which allows administrators to login to the web application using their account credentials. The system will retrieve administrator data and check the credentials. It also redirects the administrator to a home page or once successfully login or login failed when the email or password is incorrect.

ProductDashboardPage

This is the dashboard page which displays all the products which are currently selling on the web application. This is only accessible and managed by administrators. On this page, administrators can delete products which may no longer be selling or exist. They also can click the create new product and edit product buttons and redirect to those pages.

CreateProductPage

This is the page which allows administrators to create a new product. They need to fill in all the information and upload an image for the product. They can click the create button and save it into the database.

EditProductPage

This is the page which allows administrators to edit the existing product information. They can amend all the information or upload new images for the products. They can click the update button and save the changes into the database.

CategoryDashboardPage

This is the dashboard page which displays all the categories associated for products which are currently selling on the web application. This is only accessible and managed by administrators. On this page, administrators can delete categories which may no longer be selling or exist. They also can click the create new category and edit category buttons and redirect to those pages.

CreateCategoryPage

This is the page which allows administrators to create a category product. They need to fill up the name of the category. They can click the create button and save it into the database.

EditCategoryPage

This is the page which allows administrators to edit the existing category information. They can amend the category name. They can click the update button and save the changes into the database.

UserDashboardPage

This is the dashboard page which displays all the user accounts on this web application. This is only accessible and managed by administrators. On this page, administrators can click the lock button to suspend accounts which may no longer exist. They also can click the unlock button and reactivate the user accounts.

CreateNewAdminPage

This is the page which allows administrators to create a new admin account. They need to fill in all the required personal information of administrators. They can click the sign-up button to create the new administrator account and save it into the database.

NewsletterDashboardPage

This is the dashboard page which displays a mailing list which consists of all the users who subscribe to the newsletter. This is only accessible and managed by administrators. On this page, administrators have the permission to delete the user in special circumstances.

OrderDashboardPage

This is the dashboard page which displays all the orders which have been placed by customers on the web application. This is only accessible and managed by administrators. On this page, administrators can view and filter the records according to the status of order. They also can click the edit order button to update the status.

UpdateOrderStatusPage

This is the page which allows administrators to update the status of a particular order. They need to fill in all the required courier, tracking number and shipping information. There are several types of buttons like process order, ship order button to update the progress based on the current order status. The changes will then be committed and saved into the database.

4.3 System Components Interaction Operations

4.3.1 Customer Module

1.1 [Login] LoginController

This is a controller which will redirect customers to the login page when they want to perform any further action like adding products to cart or wish list.

1.2a [Register] RegisterController

If the customer is a newcomer who never signed up for an account on the web application, he/she can register a new account by filling in their personal information in the register form. They must provide their email and password to set up an account.

1.2a [Email verification] RegisterConfirmationController

The controller will generate an email verification template, set up an email service, send the email to the customer email account and customers need to verify their account in the email to successfully register.

1.3 [View all products] ProductViewController

This controller will retrieve all the products associated with its information from the ProductCrud table and store them in a list. This list will be passed to the ViewAllProductsPage for display purposes.

1.4 [View single product] SingleProductController

When the customers click a particular product, this controller will retrieve the single product information like the price, category from the ProductCrud table. This model information is passed to the ViewSingleProductPage for display purposes. Besides, the controller will call the flask application to generate item-based collaborative filtering recommendation and the product information will also be passed to the same page. The pre-condition is that this product has been rated by customers before.

1.5a [Add into cart] CartController

When the customers wish to add a product into cart, this controller will store this product detail into the ShoppingCarts table with the ApplicationUserID of this user. When customers click the cart icon, this controller will retrieve the all the cart products of this customer from the database and pass to the CartPage.

1.5b [Add into wish list] WishlistController

When the customers wish to add a product to wish list, this controller will store this product detail into the Wishlist table with the ApplicationUserID of this user. When customers click the wish list icon, this controller will retrieve the all the wish list products of this customer from the database and pass to the WishListPage.

1.6 [Check out summary] CartController

When the customers click the checkout button and wish to check out the cart with the products they are interested in buying, the controller will process the calculation of subtotal of each product and the total amount need to pay by customer to place order. All the order summary information will be passed to the CheckOutCartPage for display purposes.

1.7 [Place order and pay] OrderController

When the customers click the pay now button and wish to place order, the controller will initiate and configure the stripe payment service for payment. The total payment information will be passed to the PaymentPage and proceed for customer payment.

1.8 [Order successfully] OrderController

After the customers place the order successfully, the OrderController will store all the products, customer, and shipping information of the order into the OrderHeaders and ProductOrderDetails table. The order ID will be passed to OrderSuccessPage for display purpose.

1.9 [Manage profile] Index Razor Page

The Index Razor Page will retrieve customer information like username, phone number, address from the model binding. All the information will be passed to the ProfilePage for display and editing purposes.

1.10 [Provide recomendations] RecommenderController

This controller will call the flask application to generate user-based collaborative filtering recommendation and the product information will also be passed to the RecommendedOnlyForYouPage page. The pre-condition is that this customer had rated some products before.

1.11 [View order history] OrderController

This controller will retrieve the order information of all orders being placed by customers before and store them into a list. This list will be passed to the OrderHistoryPage for display purposes.

4.3.2 Administrator Module

2.0 [Login] LoginController

This is a controller which will redirect administratos to the login page when they want to perform any further action like managing products and orders.

2.1 [Manage all products] ProductCrudController

This controller will retrieve all the products associated with its information from the ProductCrud table and store them in a list. This list will be passed to the ProductDashboardPage for management purposes.

2.1a [Create new product] ProductCrudController

The ProductCrudController will redirect the administrator to CreateProductPage to fill up the new product information. This controller will store the new product information into the ProductCrud table when the administrator clicks the create button.

2.1b [Edit product] ProductCrudController

The ProductCrudController will redirect the administrator to EditProductPage to edit the existing product information. This controller will store the new product information into the ProductCrud table after the administrator clicks the update button.

2.2 [Manage all categories] CategoryController

This controller will retrieve all the categories associated with its information from the Categories table and store them in a list. This list will be passed to the CategoryDashboardPage for management purposes.

2.2a [Create new category] CategoryController

The CategoryController will redirect the administrator to CreateCategoryPage to fill up the new category information. This controller will store the new category information into the Categories table when the administrator clicks the create button.

2.2b [Edit category] CategoryController

The CategoryController will redirect the administrator to EditCategoryPage to edit the existing category information. This controller will store the new category information into the Categories table after the administrator clicks the update button.

2.3 [Manage all users] UserController

This controller will retrieve all the user account information who have registered an account on the web application. The controller retrieves the data from the AspNetUsers table and stores it in a list. This list will be passed to the UserDashboardPage for management purposes.

2.3a [Add new admin] UserController

The UserController will redirect the administrator to CreateNewAdminPage to fill up the new administrator personal information. This controller will store the new administrator information into the AspNetUsers table when the administrator clicks the sign-up button.

2.4 [Manage newsletter list] NewsletterController

This controller will retrieve all the users who have subscribed the newsletter on the web application. The controller retrieves the data from the Newsletter table and stores it in a list. This list will be passed to the NewsletterDashboardPage for management purposes.

2.5 [Manage all orders] OrderController

This controller will retrieve all the orders being placed by customers on the web application. The controller retrieves the data from the OrderHeaders table and stores them in a list. This list will be passed to the OrderDashboardPage for management purposes.

2.6 [Update order status] OrderController

The OrderController will redirect the administrator to UpdateOrderStatusPage to update the status of a particular order. The controller retrieves the data from the ProductOrderdetails table. This controller will store the new order information (shipping information, tracking number) into the ProductOrderdetails table after the administrator clicks the update button.

4.4 ERD Diagram

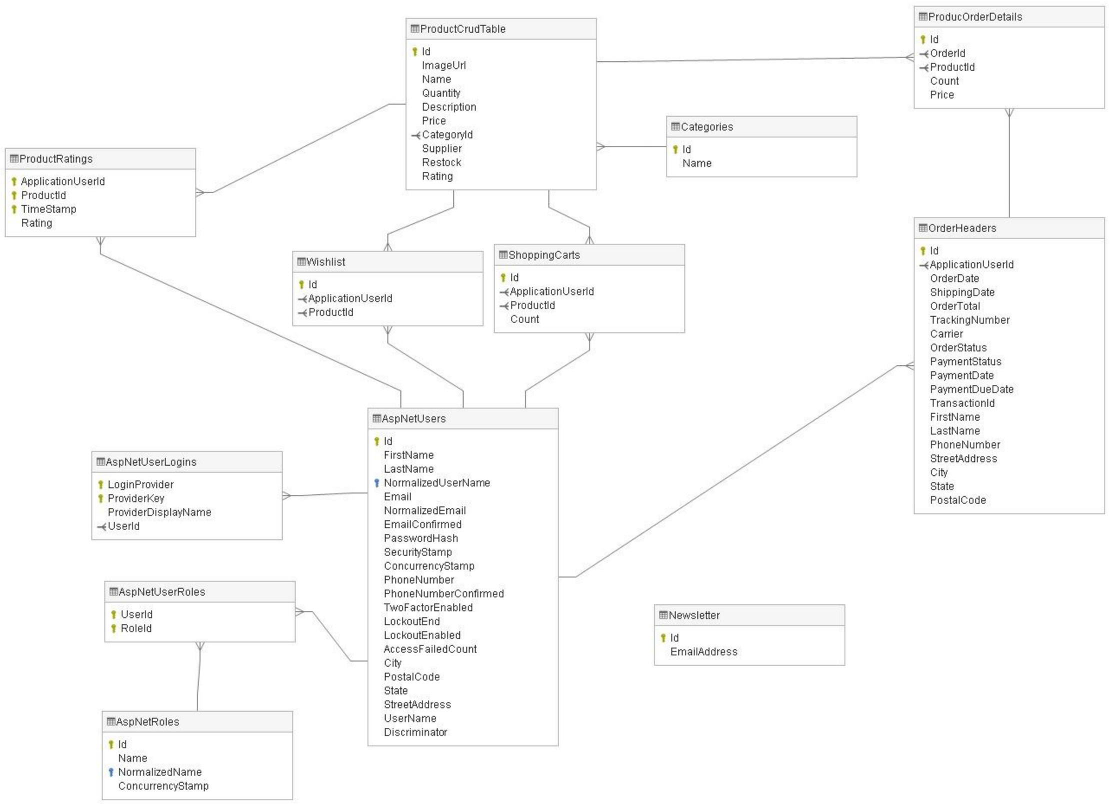


Figure 4.4 ERD diagram for proposed web application

CHAPTER 5

System Implementation

In this Chapter 5, there are discussion on the hardware and software setup had been done to build the proposed web application. There are code snippets to show the settings and configuration. Moreover, there are also some screen captures of the UI to simulate the system operation of web application and these can be act as some sort of user manual as well. Next, there will be a short section discussing about the implementation issues and challenges being met.

5.1 Hardware Setup

The hardware involved in this project is a generic computer. It is used to develop the web application with following specifications:

Table 5.1.1 Hardware used for web application development

Description	**Specifications**
Brand Model	ACER Aspire 5
Operating system	Microsoft Windows 10 Home (64-bit)
RAM memory	12 GB RAM DDR4
Disk space storage	1000GB HDD
Screen Resolution	15.60-inch, 1920 x 1080 pixel

Processor	Intel Core i5-7200U 2 x 2.5 - 3.1 GHz
Graphic	NVIDIA GeForce MX150 with 2GB VRAM

5.2 Software Setup

The software used to develop the web application:

Table 5.2.1 Software used for web application development

Description	**Tools**
IDE tools	Microsoft Visual Studio 2022
Third-party framework and plug-in	Azure web app services and API
Database	SQL database, Microsoft SQL Server Management Studio, Azure SQL server and database
Framework used	ASP.NET Core 6, Bootstrap, MVC architecture, Flask
Languages used	C#, Python, HTML, CSS, Javascript

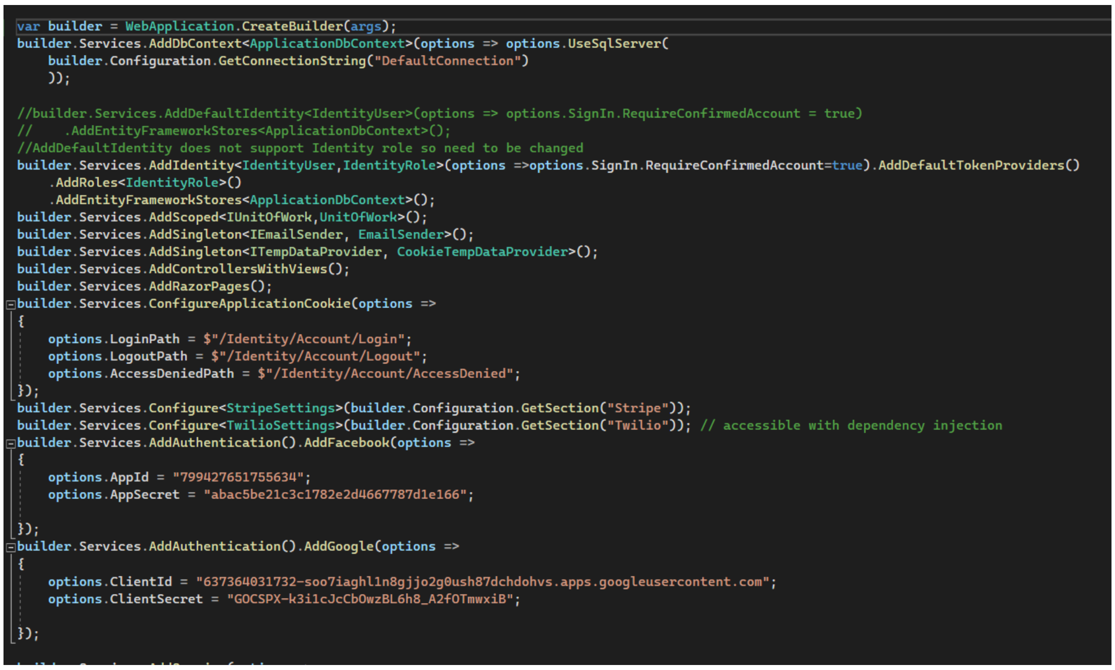
5.2.1 Software

These are the software and packages being installed and downloaded in my laptop for web application development:

1. Visual Studio 2022 Community Edition, SDK
2. Microsoft SQL Server Management Studio 18
3. Microsoft.AspNetCore.Identity.EntityFrameworkCore
4. Microsoft.AspNetCore.Identity.UI
5. Microsoft.EntityFrameworkCore
6. Microsoft.EntityFrameworkCore.SqlServer
7. Microsoft.EntityFrameworkCore.Tools
8. Microsoft.VisualStudio.Web.CodeGeneration.Design
9. Stripe.net
10. Twilio
11. Microsoft.AspNetCore.Authentication.Facebook
12. Microsoft.AspNetCore.Authentication.Google
13. Azure.Extensions.AspnetCore.Configuration.Secrets

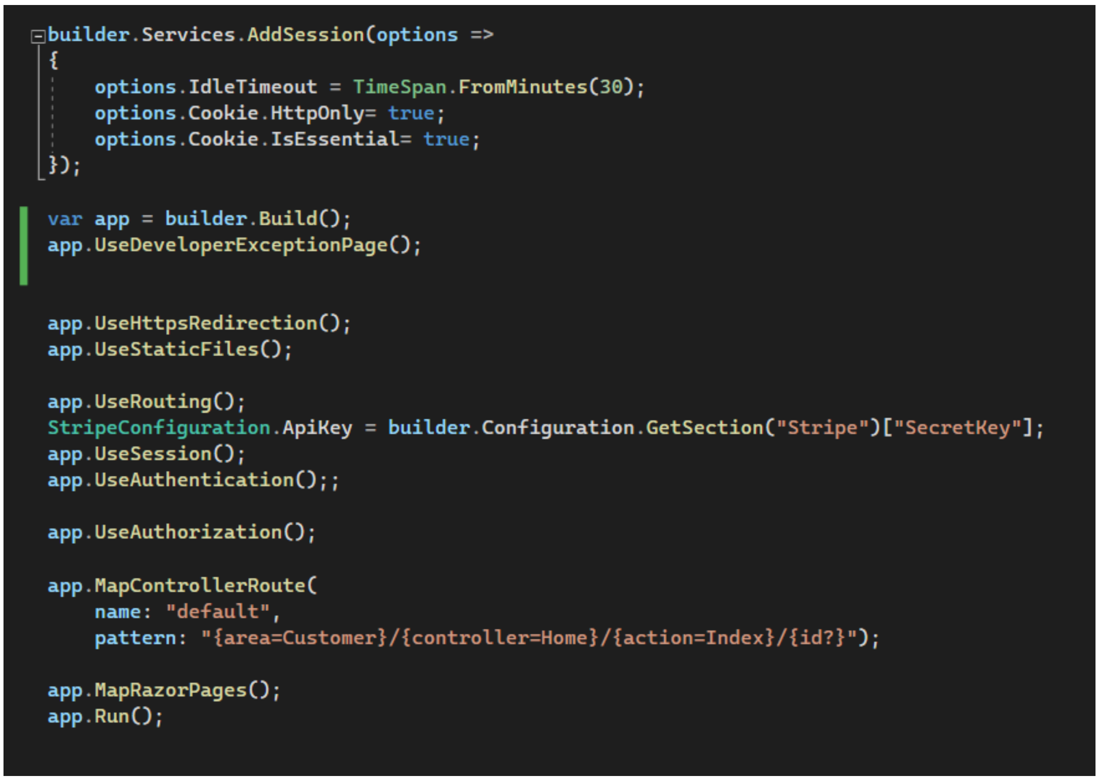
5.3 Setting and Configuration

5.3.1 Web application setup



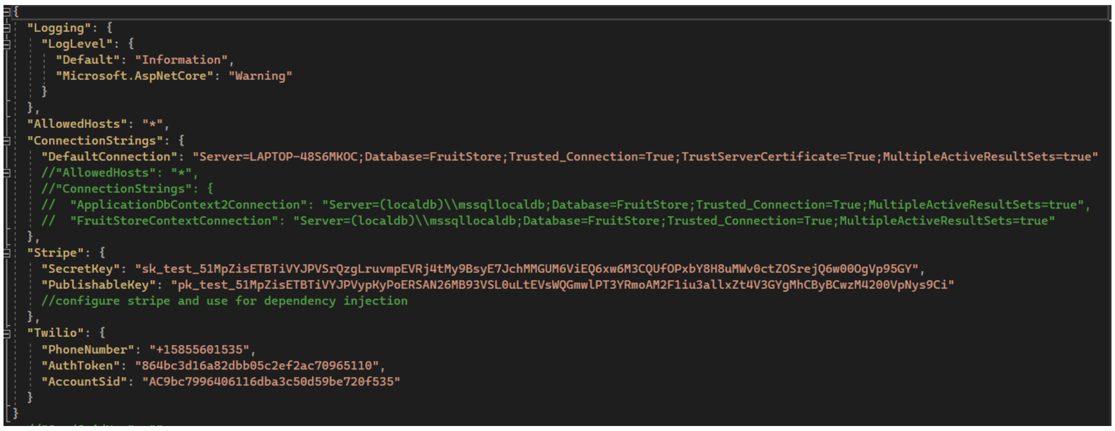
```
var builder = WebApplication.CreateBuilder(args);
builder.Services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer(
    builder.Configuration.GetConnectionString("DefaultConnection")
    ));

//builder.Services.AddDefaultIdentity<IdentityUser>(options => options.SignIn.RequireConfirmedAccount = true)
//    .AddEntityFrameworkStores<ApplicationDbContext>();
//AddDefaultIdentity does not support Identity role so need to be changed
builder.Services.AddIdentity<IdentityUser,IdentityRole>(options =>options.SignIn.RequireConfirmedAccount=true).AddDefaultTokenProviders()
    .AddRoles<IdentityRole>()
    .AddEntityFrameworkStores<ApplicationDbContext>();
builder.Services.AddScoped<IUnitOfWork,UnitOfWork>();
builder.Services.AddSingleton<IEmailSender, EmailSender>();
builder.Services.AddSingleton<ITempDataProvider, CookieTempDataProvider>();
builder.Services.AddControllersWithViews();
builder.Services.AddRazorPages();
builder.Services.ConfigureApplicationCookie(options =>
{
    options.LoginPath = $"/Identity/Account/Login";
    options.LogoutPath = $"/Identity/Account/Logout";
    options.AccessDeniedPath = $"/Identity/Account/AccessDenied";
});
builder.Services.Configure<StripeSettings>(builder.Configuration.GetSection("Stripe"));
builder.Services.Configure<TwilioSettings>(builder.Configuration.GetSection("Twilio")); // accessible with dependency injection
builder.Services.AddAuthentication().AddFacebook(options =>
{
    options.AppId = "799427651755634";
    options.AppSecret = "abac5be21c3c1782e2d4667787d1e166";

});
builder.Services.AddAuthentication().AddGoogle(options =>
{
    options.ClientId = "637364031732-soo7iaghl1n8gjjo2g0ush87dchdohvs.apps.googleusercontent.com";
    options.ClientSecret = "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";

});
```

Figure 5.3.1 Program.cs configuration of web application part 1



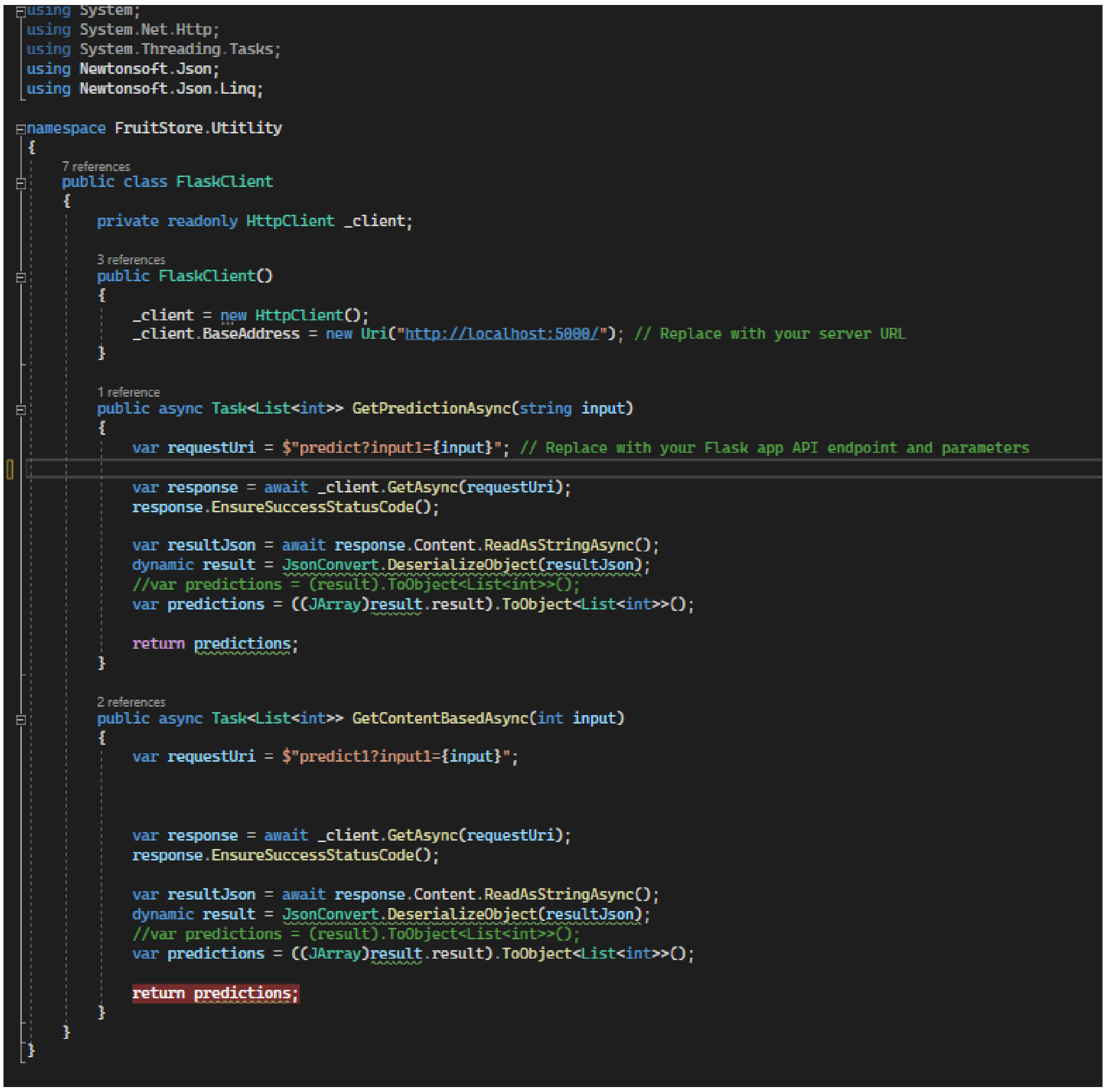
```
builder.Services.AddSession(options =>
{
    options.IdleTimeout = TimeSpan.FromMinutes(30);
    options.Cookie.HttpOnly= true;
    options.Cookie.IsEssential= true;
});

var app = builder.Build();
app.UseDeveloperExceptionPage();


app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();
StripeConfiguration.ApiKey = builder.Configuration.GetSection("Stripe")["SecretKey"];
app.UseSession();
app.UseAuthentication();;

app.UseAuthorization();

app.MapControllerRoute(
    name: "default",
    pattern: "{area=Customer}/{controller=Home}/{action=Index}/{id?}");

app.MapRazorPages();
app.Run();
```

Figure 5.3.2 Program.cs configuration of web application part 2



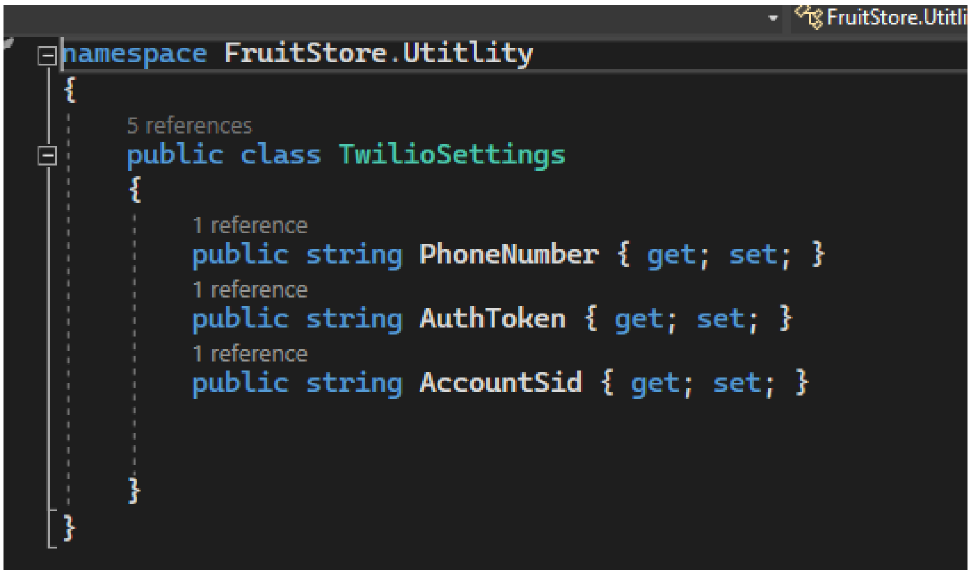
```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Server=LAPTOP-48S6MKOC;Database=FruitStore;Trusted_Connection=True;TrustServerCertificate=True;MultipleActiveResultSets=true"
    //"AllowedHosts": "*",
    //"ConnectionStrings": {
    //  "ApplicationDbContext2Connection": "Server=(localdb)\\mssqllocaldb;Database=FruitStore;Trusted_Connection=True;MultipleActiveResultSets=true",
    //  "FruitStoreContextConnection": "Server=(localdb)\\mssqllocaldb;Database=FruitStore;Trusted_Connection=True;MultipleActiveResultSets=true"
  },
  "Stripe": {
    "SecretKey": "sk_test_51MpZisETBTiVYJPVSrQzgLruvmpEVRj4tMy9BsyE7JchMMGUM6ViEQ6xw6M3CQUfOPxbY8H8uMWv0ctZOSrejQ6w00OgVp95GY",
    "PublishableKey": "pk_test_51MpZisETBTiVYJPVypKyPoERSAN26MB93VSL0uLtEVsWQGmwlPT3YRmoAM2F1iu3allxZt4V3GYgMhCByBCwzM4200VpNys9Ci"
    //configure stripe and use for dependency injection
  },
  "Twilio": {
    "PhoneNumber": "+15855601535",
    "AuthToken": "864bc3d16a82dbb05c2ef2ac70965110",
    "AccountSid": "AC9bc7996406116dba3c50d59be720f535"
  }
}
```

Figure 5.3.3 appsettings.json configuration of web application



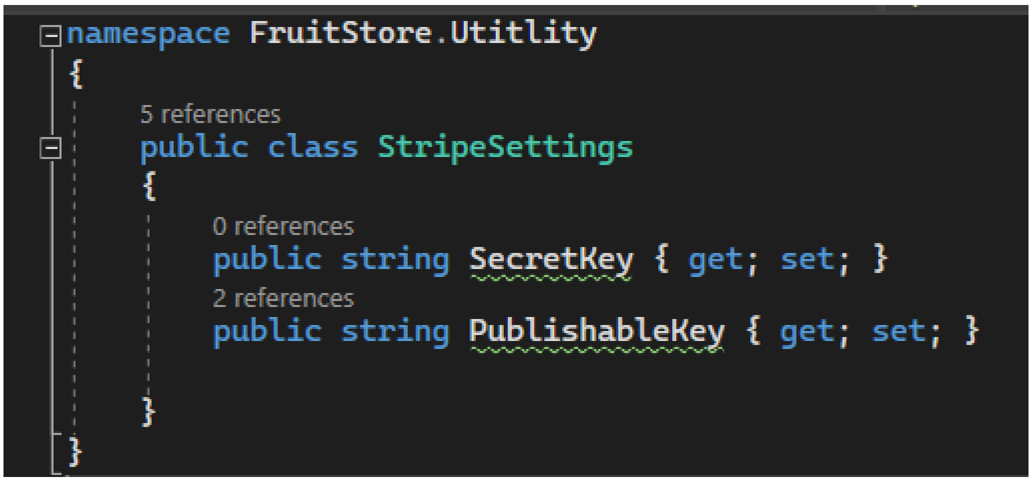
```
using System;
using System.Net.Http;
using System.Threading.Tasks;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;

namespace FruitStore.Utitlity
{
    7 references
    public class FlaskClient
    {
        private readonly HttpClient _client;

        3 references
        public FlaskClient()
        {
            _client = new HttpClient();
            _client.BaseAddress = new Uri("http://localhost:5000/"); // Replace with your server URL
        }

        1 reference
        public async Task<List<int>> GetPredictionAsync(string input)
        {
            var requestUri = $"predict?input1={input}"; // Replace with your Flask app API endpoint and parameters

            var response = await _client.GetAsync(requestUri);
            response.EnsureSuccessStatusCode();

            var resultJson = await response.Content.ReadAsStringAsync();
            dynamic result = JsonConvert.DeserializeObject(resultJson);
            //var predictions = (result).ToObject<List<int>>();
            var predictions = ((JArray)result.result).ToObject<List<int>>();

            return predictions;
        }

        2 references
        public async Task<List<int>> GetContentBasedAsync(int input)
        {
            var requestUri = $"predict1?input1={input}";


            var response = await _client.GetAsync(requestUri);
            response.EnsureSuccessStatusCode();

            var resultJson = await response.Content.ReadAsStringAsync();
            dynamic result = JsonConvert.DeserializeObject(resultJson);
            //var predictions = (result).ToObject<List<int>>();
            var predictions = ((JArray)result.result).ToObject<List<int>>();

            return predictions;
        }
    }
}
```

Figure 5.3.4 FlaskClient configuration to redirect request and response to Flask



```
namespace FruitStore.Utitlity
{
    5 references
    public class TwilioSettings
    {
        1 reference
        public string PhoneNumber { get; set; }
        1 reference
        public string AuthToken { get; set; }
        1 reference
        public string AccountSid { get; set; }


    }
}
```

Figure 5.3.5 TwilioSettings configuration to send customer the order success message via phone number



```
namespace FruitStore.Utitlity
{
    5 references
    public class StripeSettings
    {
        0 references
        public string SecretKey { get; set; }
        2 references
        public string PublishableKey { get; set; }

    }
}
```

Figure 5.3.6 StripeSettings configuration for payment option

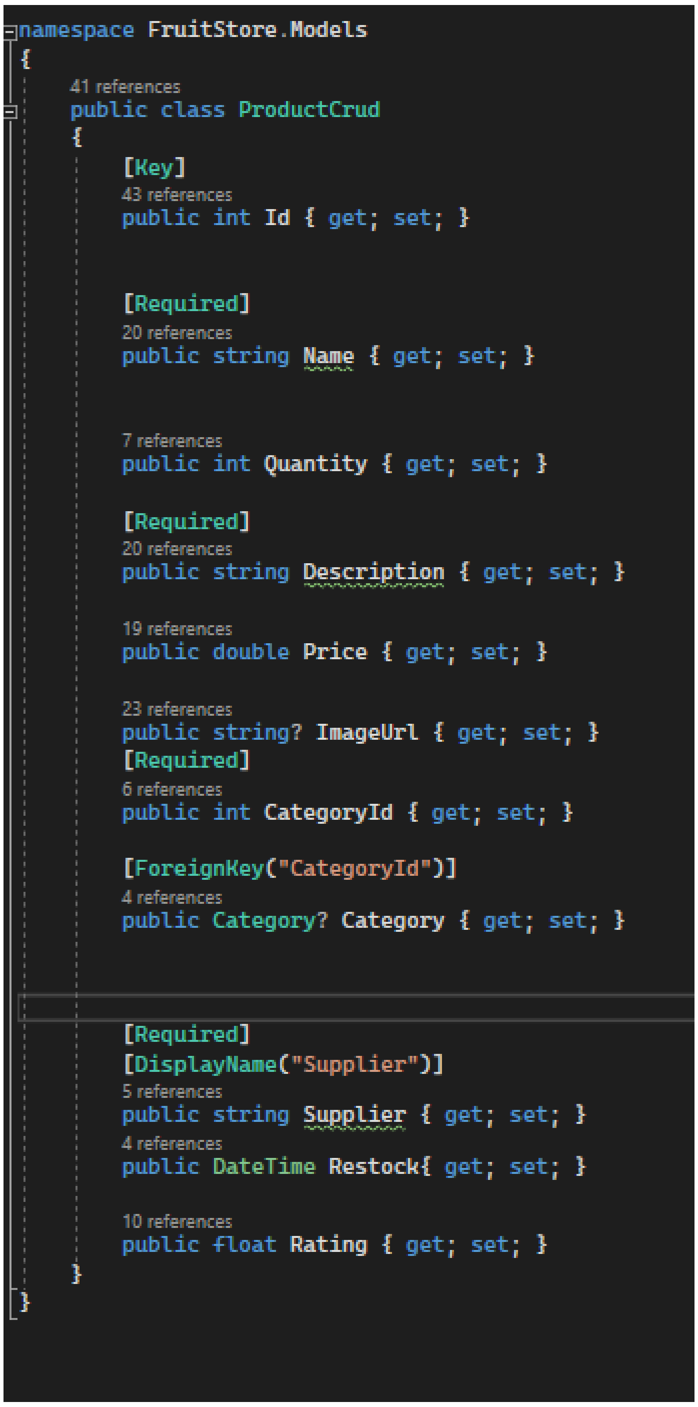
5.3.2 Program Code Snippet and Implementation flow

CRUD module using MVC Architecture and ASP.NET Entity Framework Core

In order to allow administrator to perform management on the web application and display to client-side dynamically, we need to create a product module using ASP.NET MVC architecture with the notable ASP.NET Entity Framework Core Code First approach as an example. This approach is applicable to every module. Precisely speaking, Code First approach allows developer to create the entity model in coding which will then provide the ability to create the database from the model without knowing any SQL queries. After creating this module, administrator is able to perform create, read, update and delete action on the modules like product module, order module, category module. For example, administrator inputs the required product information and create them. After that, all the products which created successfully and stored in the database will dynamically display in the product page of the web application without the effort of hardcoding all the products and its information.

The following are the steps of creating a CRUD module (example using product CRUD module):

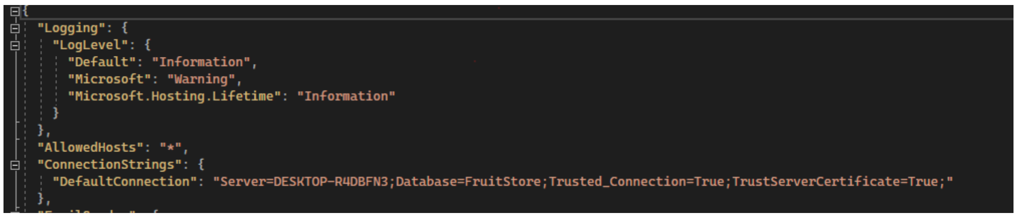
i. Create the product model and define the field member for this class.



```
namespace FruitStore.Models
{
    41 references
    public class ProductCrud
    {
        [Key]
        43 references
        public int Id { get; set; }


        [Required]
        20 references
        public string Name { get; set; }


        7 references
        public int Quantity { get; set; }

        [Required]
        20 references
        public string Description { get; set; }

        19 references
        public double Price { get; set; }

        23 references
        public string? ImageUrl { get; set; }
        [Required]
        6 references
        public int CategoryId { get; set; }

        [ForeignKey("CategoryId")]
        4 references
        public Category? Category { get; set; }



        [Required]
        [DisplayName("Supplier")]
        5 references
        public string Supplier { get; set; }
        4 references
        public DateTime Restock{ get; set; }

        10 references
        public float Rating { get; set; }
    }
}
```

Figure 5.3.7 Product Model (ProductCrud.cs)

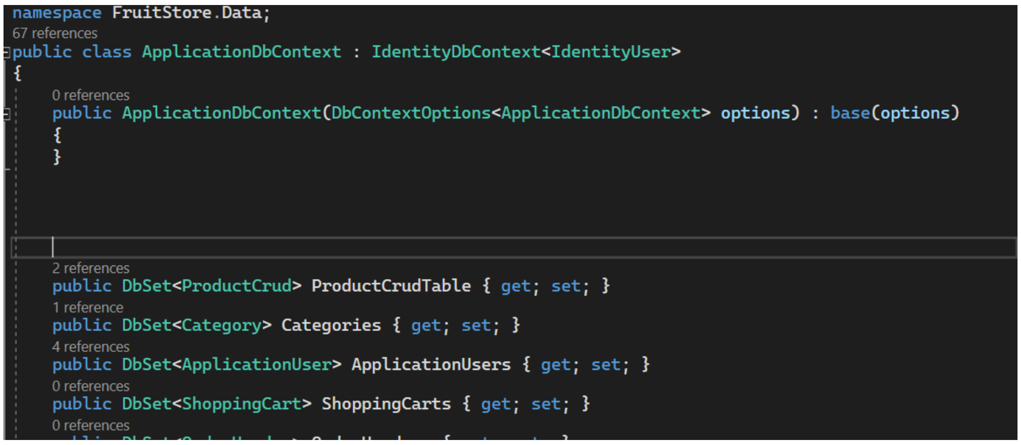
ii. Add the connection string to appsettings.json file in order to connect the product model and SQL server.



```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Server=DESKTOP-R4DBFN3;Database=FruitStore;Trusted_Connection=True;TrustServerCertificate=True;"
  },
```

Figure 5.3.8 Default Connection is the connection string (appsettings.json)

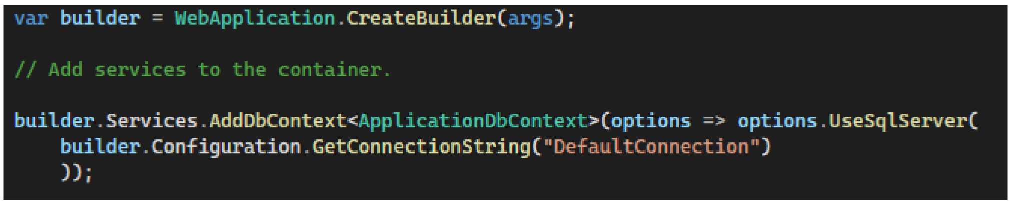
iii. Create ApplicationDbContext file and set up the Product entity in which this will create the Product table in database later after migration.



```
namespace FruitStore.Data;
67 references
public class ApplicationDbContext : IdentityDbContext<IdentityUser>
{
    0 references
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options)
    {
    }



    2 references
    public DbSet<ProductCrud> ProductCrudTable { get; set; }
    1 reference
    public DbSet<Category> Categories { get; set; }
    4 references
    public DbSet<ApplicationUser> ApplicationUsers { get; set; }
    0 references
    public DbSet<ShoppingCart> ShoppingCarts { get; set; }
    0 references
```

Figure 5.3.9 ApplicationDbContext (ApplicationDbContext.cs)

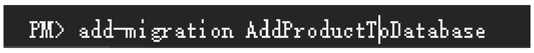
iv. Setup Program.cs to use the ApplicationDbContext before executing the web application.



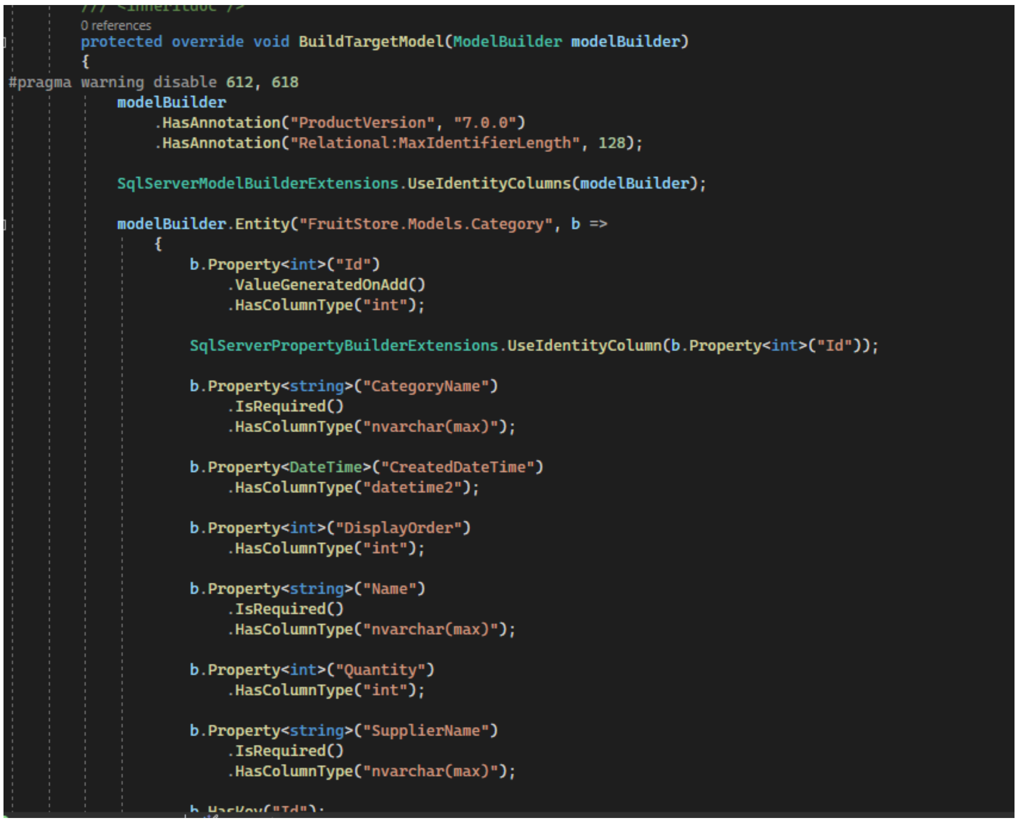
```
var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer(
    builder.Configuration.GetConnectionString("DefaultConnection")
    ));
```

Figure 5.3.10 Include the ApplicationDbContext in the configuration (Program.cs)

v. Use the command "add-migration -meaningfulname" in the Package manager console to perform migration in which migration will sync the database schema with the Entity Framework model. For instance, always make sure the product schema and number of field member is keep sync with the product model.



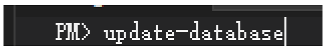
```
PM> add-migration AddProductToDatabase
```



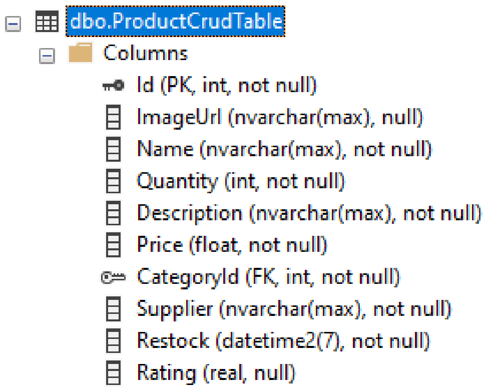
```
        /// <inheritdoc />
        0 references
        protected override void BuildTargetModel(ModelBuilder modelBuilder)
        {
#pragma warning disable 612, 618
            modelBuilder
                .HasAnnotation("ProductVersion", "7.0.0")
                .HasAnnotation("Relational:MaxIdentifierLength", 128);

            SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

            modelBuilder.Entity("FruitStore.Models.Category", b =>
                {
                    b.Property<int>("Id")
                        .ValueGeneratedOnAdd()
                        .HasColumnType("int");

                    SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

                    b.Property<string>("CategoryName")
                        .IsRequired()
                        .HasColumnType("nvarchar(max)");

                    b.Property<DateTime>("CreatedDateTime")
                        .HasColumnType("datetime2");

                    b.Property<int>("DisplayOrder")
                        .HasColumnType("int");

                    b.Property<string>("Name")
                        .IsRequired()
                        .HasColumnType("nvarchar(max)");

                    b.Property<int>("Quantity")
                        .HasColumnType("int");

                    b.Property<string>("SupplierName")
                        .IsRequired()
                        .HasColumnType("nvarchar(max)");

                    b.HasKey("Id");
```

Figure 5.3.11 Include the ApplicationDbContext in the configuration (Program.cs)

vi. Use the command "update-database". The migrations which contain the database schema will automatically create the table and update the database in the SQL Server Management Studio. The command is vital and important to use whenever any new tables need to be added or new updates provided.

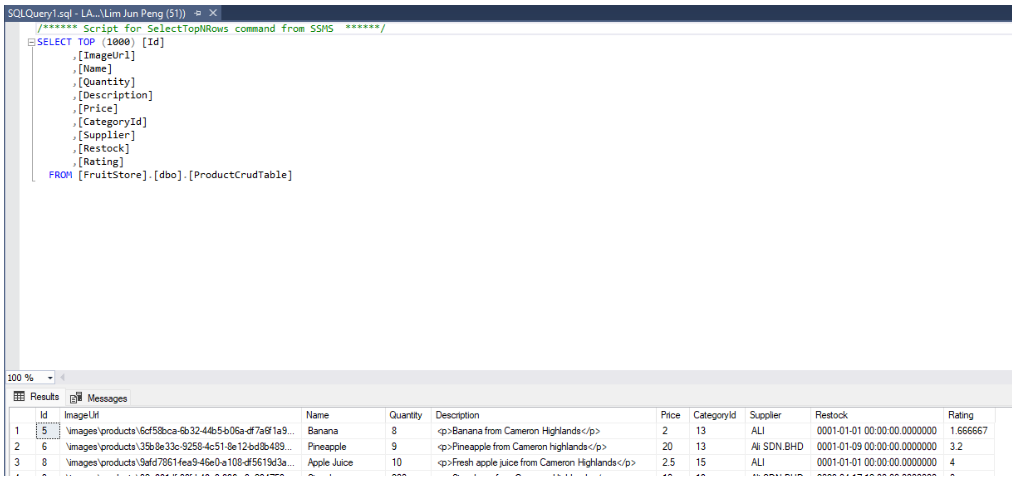


```
PM> update-database
```



```
dbo.ProductCrudTable
    Columns
        Id (PK, int, not null)
        ImageUrl (nvarchar(max), null)
        Name (nvarchar(max), not null)
        Quantity (int, not null)
        Description (nvarchar(max), not null)
        Price (float, not null)
        CategoryId (FK, int, not null)
        Supplier (nvarchar(max), not null)
        Restock (datetime2(7), not null)
        Rating (real, null)
```

Figure 5.3.12 Product table is added to the database in SSMS



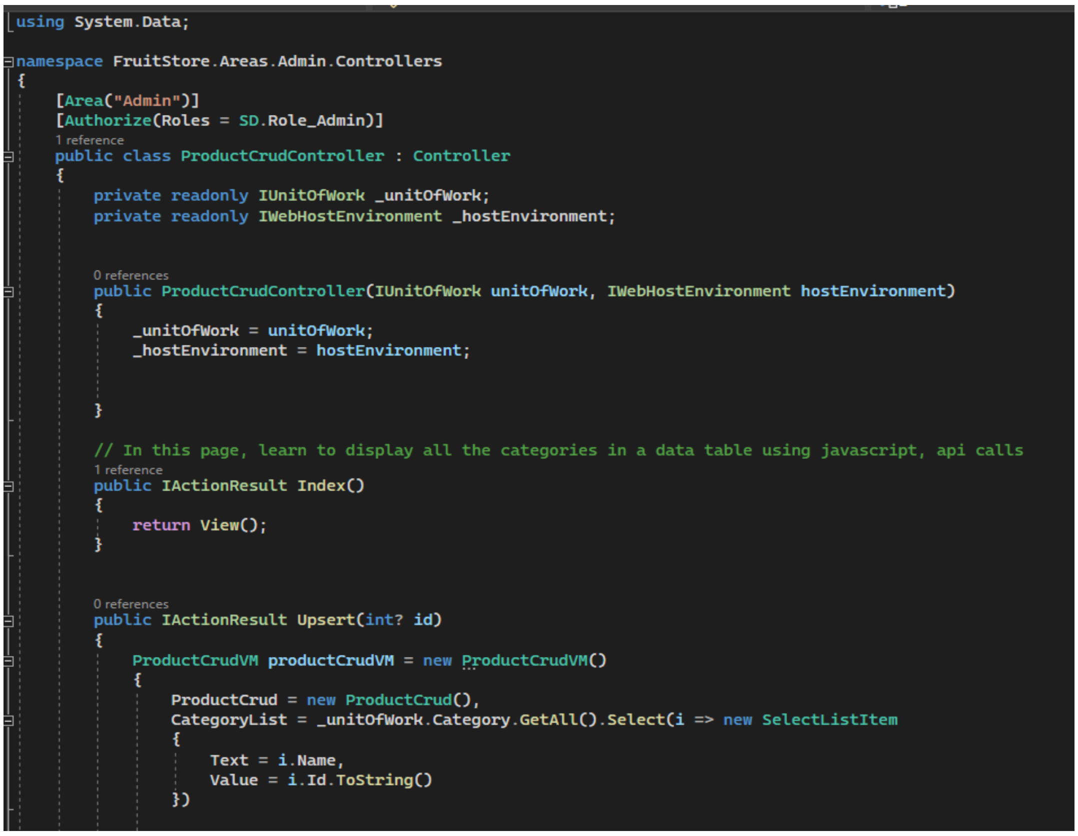
SQLQuery1.sql - LA...\Lim Jun Peng (51))

```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [Id]
      ,[ImageUrl]
      ,[Name]
      ,[Quantity]
      ,[Description]
      ,[Price]
      ,[CategoryId]
      ,[Supplier]
      ,[Restock]
      ,[Rating]
  FROM [FruitStore].[dbo].[ProductCrudTable]
```

	Id	ImageUrl	Name	Quantity	Description	Price	CategoryId	Supplier	Restock	Rating
1	5	\images\products\6cf58bca-6b32-44b5-b06a-df7a6f1a9...	Banana	8	<p>Banana from Cameron Highlands</p>	2	13	ALI	0001-01-01 00:00:00.0000000	1.666667
2	6	\images\products\35b8e33c-9258-4c51-8e12-bd8b489...	Pineapple	9	<p>Pineapple from Cameron highlands</p>	20	13	Ali SDN.BHD	0001-01-09 00:00:00.0000000	3.2
3	8	\images\products\9afd7861-fea9-46e0-a108-df5619d3a...	Apple Juice	10	<p>Fresh apple juice from Cameron Highlands</p>	2.5	15	ALI	0001-01-01 00:00:00.0000000	4

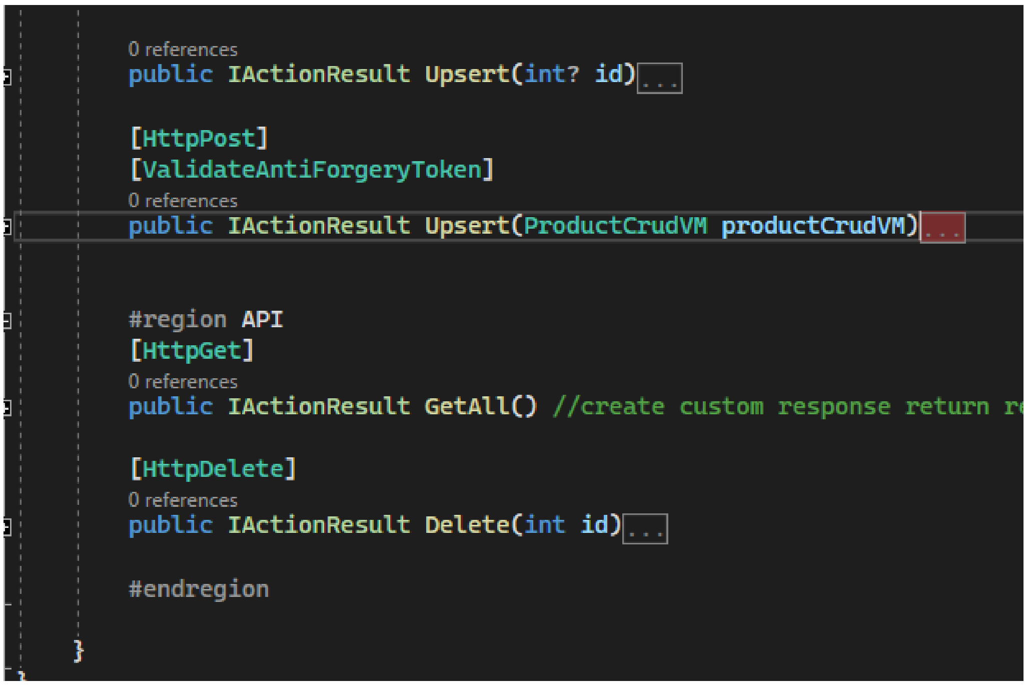
Figure 5.3.13 Product table is created

vii. After this step, the product table and the database are successfully created from the model. Next, controller and view needed to be created in order to perform CRUD operation on the products from the database. Controller is a middle person which redirects request and response between model and view. From the Figure 4.4.8, Index() in ProductCrudController.cs is a function to display all the product details which can be indicated as the view operation while Upsert() will be reflected as create and update operation while Delete() will be indicated as delete operation.



```
using System.Data;

namespace FruitStore.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize(Roles = SD.Role_Admin)]
    1 reference
    public class ProductCrudController : Controller
    {
        private readonly IUnitOfWork _unitOfWork;
        private readonly IWebHostEnvironment _hostEnvironment;

        0 references
        public ProductCrudController(IUnitOfWork unitOfWork, IWebHostEnvironment hostEnvironment)
        {
            _unitOfWork = unitOfWork;
            _hostEnvironment = hostEnvironment;

        }

        // In this page, learn to display all the categories in a data table using javascript, api calls
        1 reference
        public IActionResult Index()
        {
            return View();
        }

        0 references
        public IActionResult Upsert(int? id)
        {
            ProductCrudVM productCrudVM = new ProductCrudVM()
            {
                ProductCrud = new ProductCrud(),
                CategoryList = _unitOfWork.Category.GetAll().Select(i => new SelectListItem
                {
                    Text = i.Name,
                    Value = i.Id.ToString()
                })
```

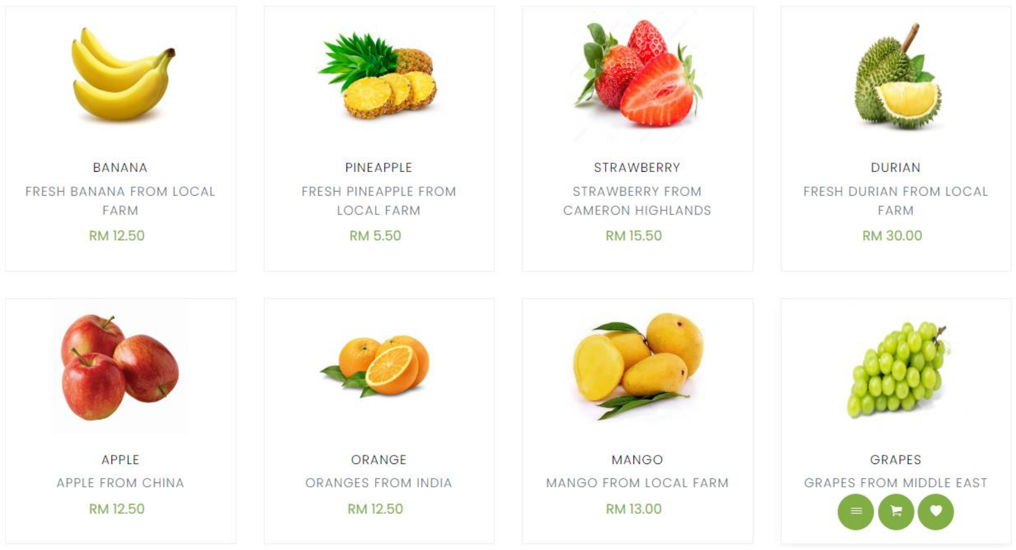
Figure 5.3.14 Index() method (ProductController.cs)



```
        0 references
        public IActionResult Upsert(int? id)...

        [HttpPost]
        [ValidateAntiForgeryToken]
        0 references
        public IActionResult Upsert(ProductCrudVM productCrudVM)...


        #region API
        [HttpGet]
        0 references
        public IActionResult GetAll() //create custom response return re

        [HttpDelete]
        0 references
        public IActionResult Delete(int id)...

        #endregion

    }
}
```

Figure 5.3.15 Other methods (ProductController.cs)

viii. The final step is to create the view component in MVC. It is basically a generic client-side HTML5 page designed with Bootstrap and CSS which serve for display purposes. From Figure 4.4.10 and 4.4.11, all the products in the product table are being displayed in the Product page which can be accessed by all users and Product Dashboard page only accessible by administrators.


BANANA
FRESH BANANA FROM LOCAL FARM
RM 12.50

PINEAPPLE
FRESH PINEAPPLE FROM LOCAL FARM
RM 5.50

STRAWBERRY
STRAWBERRY FROM CAMERON HIGHLANDS
RM 15.50

DURIAN
FRESH DURIAN FROM LOCAL FARM
RM 30.00

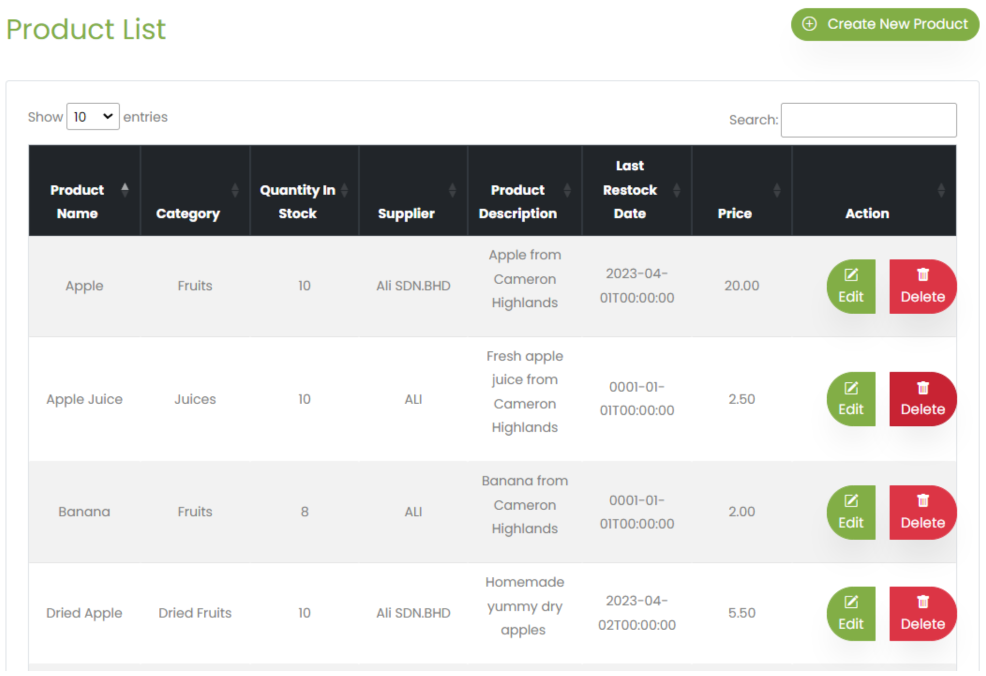
APPLE
APPLE FROM CHINA
RM 12.50

ORANGE
ORANGES FROM INDIA
RM 12.50

MANGO
MANGO FROM LOCAL FARM
RM 13.00

GRAPES
GRAPES FROM MIDDLE EAST

Figure 5.3.16 Product page for all users (Index.cshtml)


Product List

Create New Product

Show 10 entries

Search:

Product Name	Category	Quantity In Stock	Supplier	Product Description	Last Restock Date	Price	Action
Apple	Fruits	10	Ali SDN.BHD	Apple from Cameron Highlands	2023-04-01T00:00:00	20.00	Edit Delete
Apple Juice	Juices	10	ALI	Fresh apple juice from Cameron Highlands	0001-01-01T00:00:00	2.50	Edit Delete
Banana	Fruits	8	ALI	Banana from Cameron Highlands	0001-01-01T00:00:00	2.00	Edit Delete
Dried Apple	Dried Fruits	10	Ali SDN.BHD	Homemade yummy dry apples	2023-04-02T00:00:00	5.50	Edit Delete

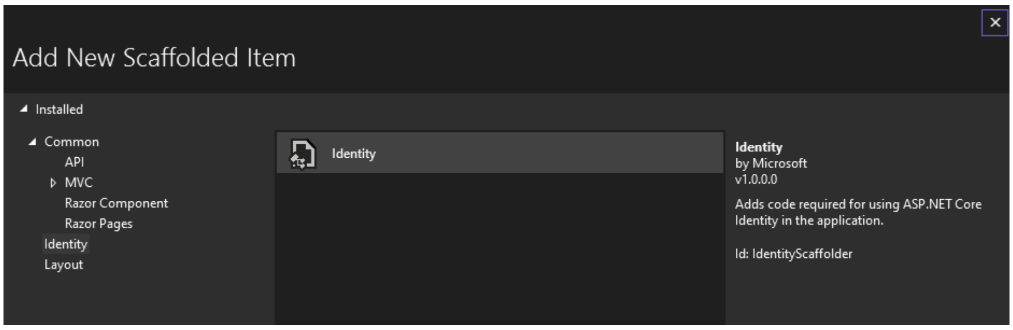
Figure 5.3.17 Product dashboard page for administrators only (Index.cshtml)

5.3.3 Role-based authentication module using ASP.NET Core Identity Scaffolding

In the proposed web application, it is a must and compulsory to have login, register, profile module in order to make sure every user or customer will have account with their own login credentials and profile information. ASP.NET Core Identity is a defined API in ASP.NET CORE 6 which provide the functionality of authentication and authorization module. By scaffolding the register, login, logout,register and other files generated by ASP.NET Identity, it reduces the burden to manually hardcode all those HTML pages and controllers to have those functions. However, it still requires some configurations so that user can be able to successfully create an account, login and perform other operations like forget password and manage profile. For instance, identity is required to be configured in order to store the table column name that are needed into the SQL database in SSMS. Moreover, role also being assigned in two types which are basically normal user and administrators in this web application to show different level of access control and restrict the page access by authorizing users with defined role.

The following are the steps of creating and implementing the authentication module:

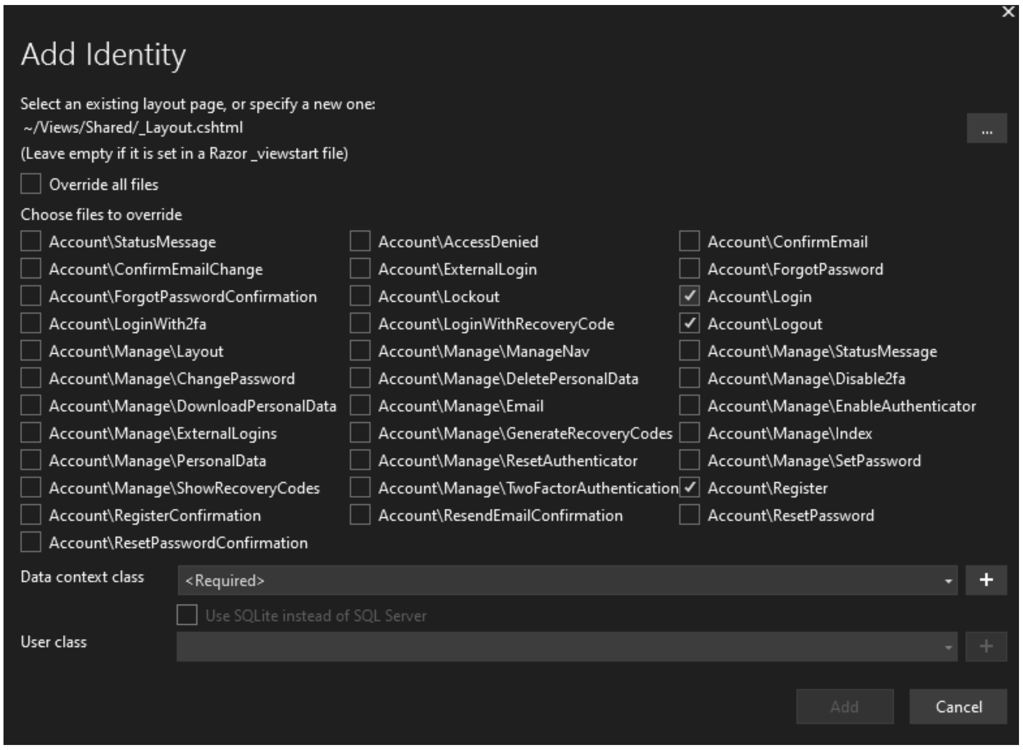
i. The first step is to create a new scaffolded Identity item by adding the functionality we need from Identity provided solution.


Add New Scaffolded Item

Identity
by Microsoft
v1.0.0.0
Adds code required for using ASP.NET Core Identity in the application.

Id: IdentityScaffolder

Figure 5.3.18 Add a new scaffolded Identity in the file solution



Add Identity

Select an existing layout page, or specify a new one:
~/Views/Shared/_Layout.cshtml
(Leave empty if it is set in a Razor _viewstart file)

☐ Override all files

Choose files to override

☐ Account\StatusMessage	☐ Account\AccessDenied	☐ Account\ConfirmEmail
☐ Account\ConfirmEmailChange	☐ Account\ExternalLogin	☐ Account\ForgotPassword
☐ Account\ForgotPasswordConfirmation	☐ Account\Lockout	✓ Account\Login
☐ Account\LoginWith2fa	☐ Account\LoginWithRecoveryCode	✓ Account\Logout
☐ Account\Manage\Layout	☐ Account\Manage\ManageNav	☐ Account\Manage\StatusMessage
☐ Account\Manage\ChangePassword	☐ Account\Manage\DeletePersonalData	☐ Account\Manage\Disable2fa
☐ Account\Manage\DownloadPersonalData	☐ Account\Manage\Email	☐ Account\Manage\EnableAuthenticator
☐ Account\Manage\ExternalLogins	☐ Account\Manage\GenerateRecoveryCodes	☐ Account\Manage\Index
☐ Account\Manage\PersonalData	☐ Account\Manage\ResetAuthenticator	☐ Account\Manage\SetPassword
☐ Account\Manage\ShowRecoveryCodes	☐ Account\Manage\TwoFactorAuthentication	✓ Account\Register
☐ Account\RegisterConfirmation	☐ Account\ResendEmailConfirmation	☐ Account\ResetPassword
☐ Account\ResetPasswordConfirmation		

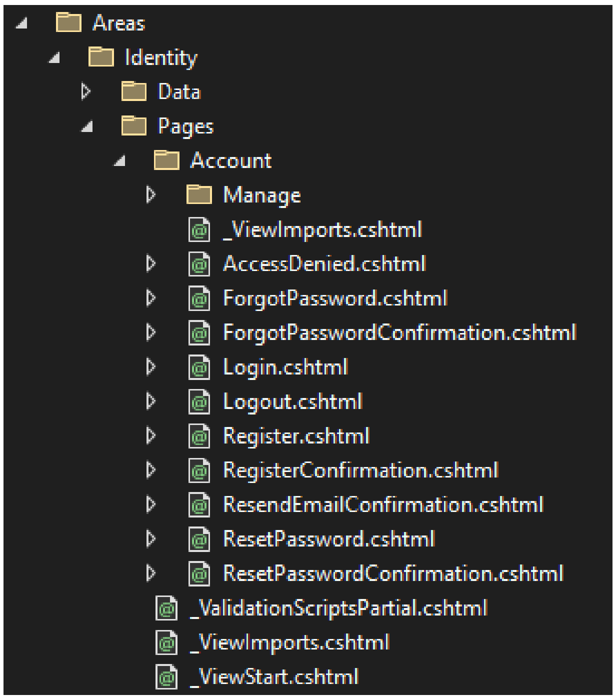
Data context class: <Required>

☐ Use SQLite instead of SQL Server

User class:

Figure 5.3.19 Add the functionality we need from Identity provided solution

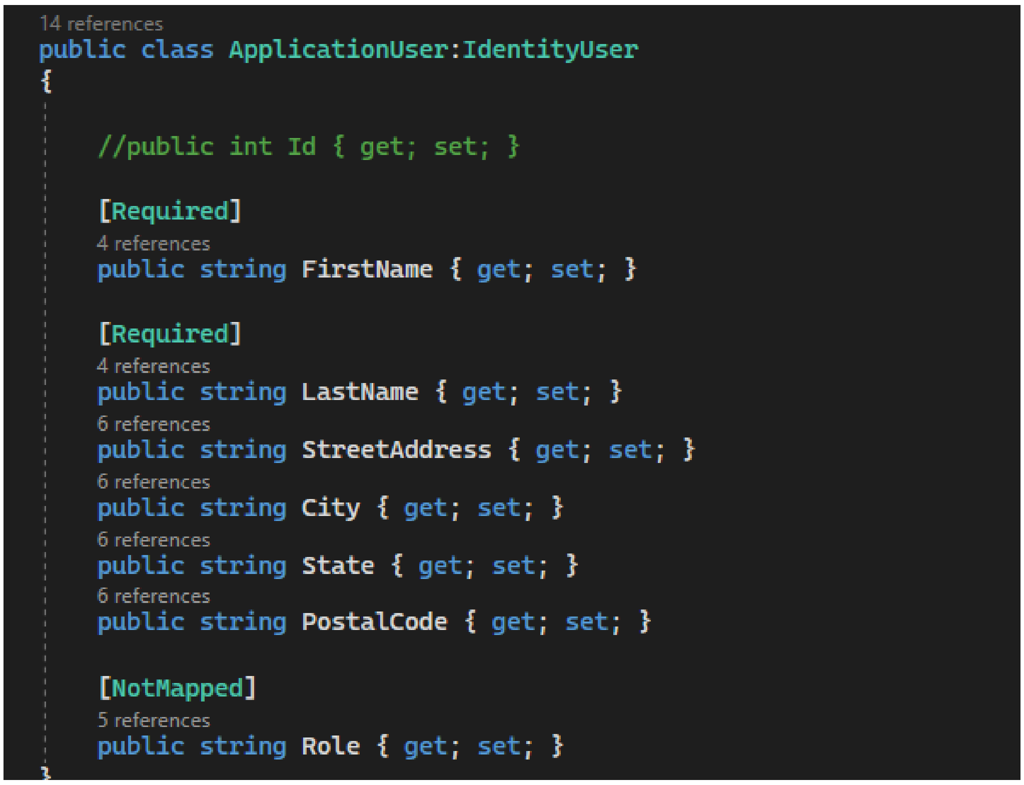
ii. After we add the Identity item, the controller and view components will auto be generated according to the functionality we want to implement.



```
Areas
  Identity
    Data
    Pages
      Account
        Manage
        _ViewImports.cshtml
        AccessDenied.cshtml
        ForgotPassword.cshtml
        ForgotPasswordConfirmation.cshtml
        Login.cshtml
        Logout.cshtml
        Register.cshtml
        RegisterConfirmation.cshtml
        ResendEmailConfirmation.cshtml
        ResetPassword.cshtml
        ResetPasswordConfirmation.cshtml
      _ValidationScriptsPartial.cshtml
      _ViewImports.cshtml
      _ViewStart.cshtml
```

Figure 5.3.20 Folder which contain the view and controller pages

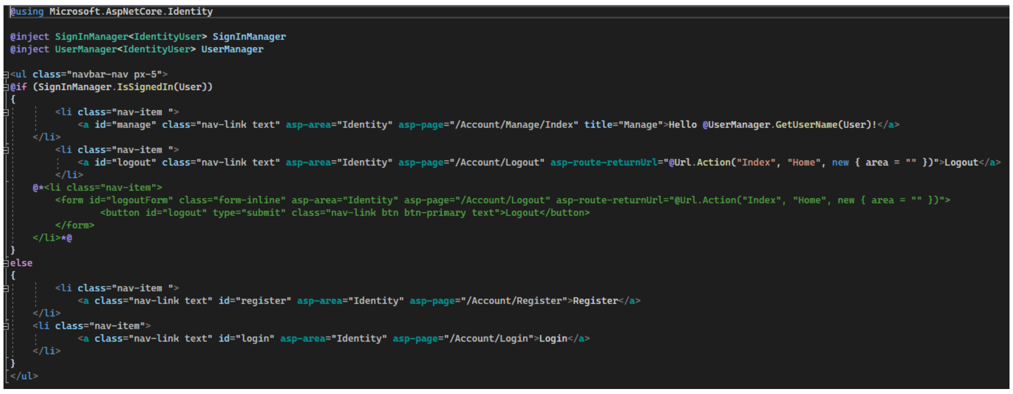
iii. Create a class to add the class member that we require the user to fill up so that these details will be stored in the database. It is pretty similar to a model class.



```
14 references
public class ApplicationUser:IdentityUser
{

    //public int Id { get; set; }

    [Required]
    4 references
    public string FirstName { get; set; }

    [Required]
    4 references
    public string LastName { get; set; }
    6 references
    public string StreetAddress { get; set; }
    6 references
    public string City { get; set; }
    6 references
    public string State { get; set; }
    6 references
    public string PostalCode { get; set; }

    [NotMapped]
    5 references
    public string Role { get; set; }
}
```

Figure 5.3.21 Model class that contains the class member (ApplicationUser.cs)

iv. Besides, it will auto-generate a partial view that already contains the login, register and sign out elements in which we included in previous section. We can add the partial view in any of the full view page such as putting at the navigation bar of every page by adding one line to include it, "<partial name="_LoginPartial" />".



```
@using Microsoft.AspNetCore.Identity

@inject SignInManager<IdentityUser> SignInManager
@inject UserManager<IdentityUser> UserManager

<ul class="navbar-nav px-5">
@if (SignInManager.IsSignedIn(User))
{
        <li class="nav-item ">
            <a id="manage" class="nav-link text" asp-area="Identity" asp-page="/Account/Manage/Index" title="Manage">Hello @UserManager.GetUserName(User)!</a>
    </li>
        <li class="nav-item ">
            <a id="logout" class="nav-link text" asp-area="Identity" asp-page="/Account/Logout" asp-route-returnUrl="@Url.Action("Index", "Home", new { area = "" })">Logout</a>
        </li>
    @*<li class="nav-item">
        <form id="logoutForm" class="form-inline" asp-area="Identity" asp-page="/Account/Logout" asp-route-returnUrl="@Url.Action("Index", "Home", new { area = "" })">
                <button id="logout" type="submit" class="nav-link btn btn-primary text">Logout</button>
        </form>
    </li>*@
}
else
{
        <li class="nav-item ">
            <a class="nav-link text" id="register" asp-area="Identity" asp-page="/Account/Register">Register</a>
    </li>
    <li class="nav-item">
            <a class="nav-link text" id="login" asp-area="Identity" asp-page="/Account/Login">Login</a>
    </li>
}
</ul>
```

Figure 5.3.22 Partial view for the Identity components (_LoginPartial.cshtml)

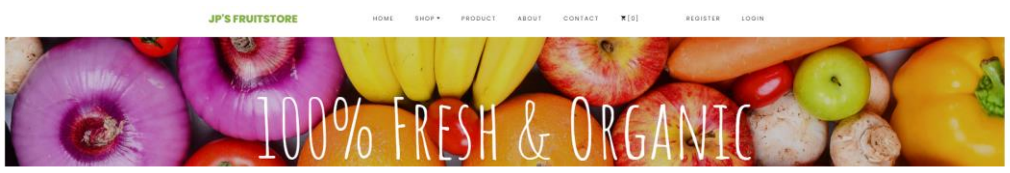
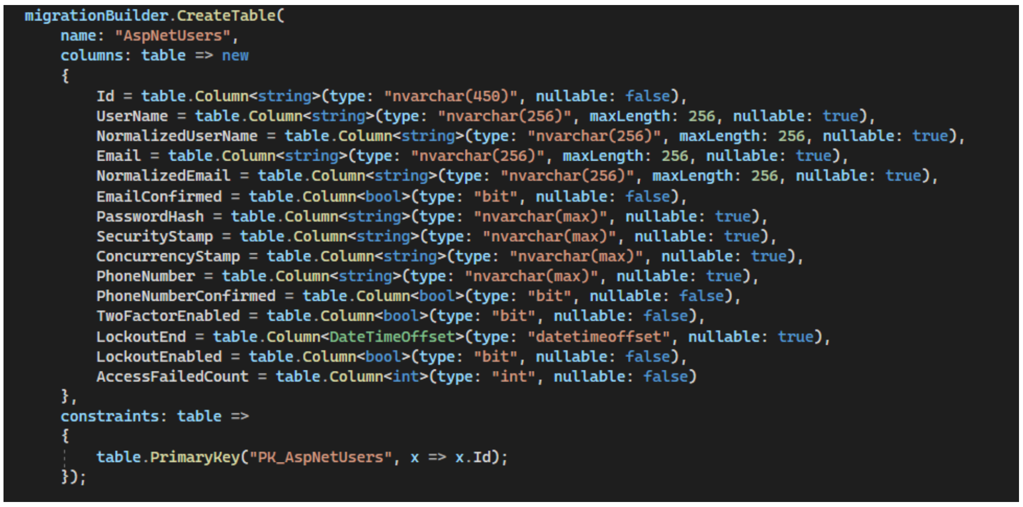


Figure 5.3.23 Partial view being added to navigation bar (_Layout.cshtml)

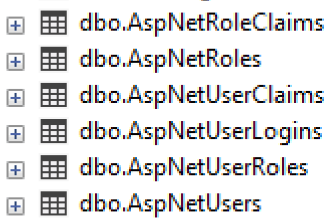
v. After that, a database context is required to store the database schema of the identity components. Then, simply use the command "add-migration -meaningfulname" and "update-database" are able to perform migration so that it can create all the table related to user module in the behind logic.



```
migrationBuilder.CreateTable(
    name: "AspNetUsers",
    columns: table => new
    {
        Id = table.Column<string>(type: "nvarchar(450)", nullable: false),
        UserName = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),
        NormalizedUserName = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),
        Email = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),
        NormalizedEmail = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),
        EmailConfirmed = table.Column<bool>(type: "bit", nullable: false),
        PasswordHash = table.Column<string>(type: "nvarchar(max)", nullable: true),
        SecurityStamp = table.Column<string>(type: "nvarchar(max)", nullable: true),
        ConcurrencyStamp = table.Column<string>(type: "nvarchar(max)", nullable: true),
        PhoneNumber = table.Column<string>(type: "nvarchar(max)", nullable: true),
        PhoneNumberConfirmed = table.Column<bool>(type: "bit", nullable: false),
        TwoFactorEnabled = table.Column<bool>(type: "bit", nullable: false),
        LockoutEnd = table.Column<DateTimeOffset>(type: "datetimeoffset", nullable: true),
        LockoutEnabled = table.Column<bool>(type: "bit", nullable: false),
        AccessFailedCount = table.Column<int>(type: "int", nullable: false)
    },
    constraints: table =>
    {
        table.PrimaryKey("PK_AspNetUsers", x => x.Id);
    });
```

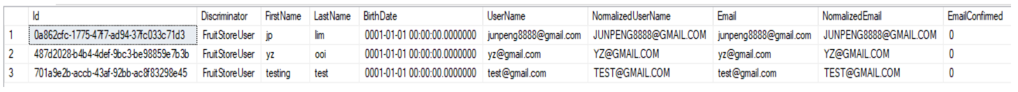
Figure 5.3.24 AddIdentityToDb.cs migration file

vi. Now, the tables for user-related tables are done created in the database.



```
dbo.AspNetRoleClaims
dbo.AspNetRoles
dbo.AspNetUserClaims
dbo.AspNetUserLogins
dbo.AspNetUserRoles
dbo.AspNetUsers
```

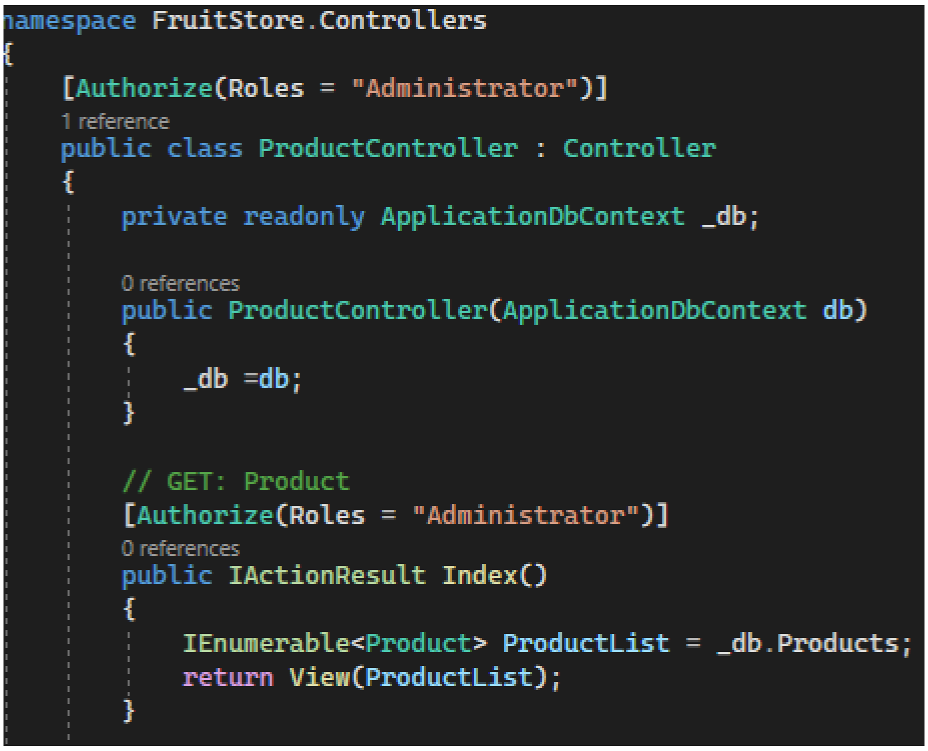
Figure 5.3.25 User table is added to the database in SSMS



	Id	Discriminator	FirstName	LastName	BirthDate	UserName	NormalizedUserName	Email	NormalizedEmail	EmailConfirmed
1	0a862cfc-1775-47f7-ad94-37fc033c71d3	FruitStoreUser	jp	lim	0001-01-01 00:00:00.0000000	junpeng8888@gmail.com	JUNPENG8888@GMAIL.COM	junpeng8888@gmail.com	JUNPENG8888@GMAIL.COM	0
2	487d2028-b4b4-4def-9bc3-be98859e7b3b	FruitStoreUser	yz	ooi	0001-01-01 00:00:00.0000000	yz@gmail.com	YZ@GMAIL.COM	yz@gmail.com	YZ@GMAIL.COM	0
3	701a9e2b-accb-43af-92bb-ac8f83298e45	FruitStoreUser	testing	test	0001-01-01 00:00:00.0000000	test@gmail.com	TEST@GMAIL.COM	test@gmail.com	TEST@GMAIL.COM	0

Figure 5.3.26 User table is created and some example rows

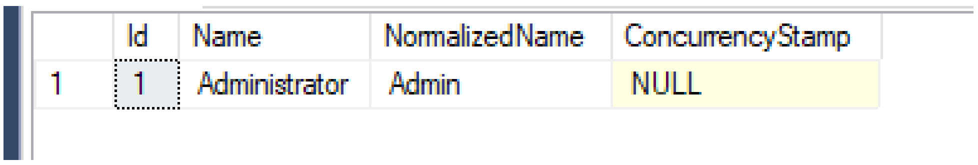
vii. Until this stage, the user is capable to perform generic action like register, login, manage profile and logout function but there must be some access control so that only administrator can view certain pages which should not be accessible by normal users. Here, we need to assign some role to certain controller actions so that it can restrict the action of users to access that method and view its page.



```
namespace FruitStore.Controllers
{
    [Authorize(Roles = "Administrator")]
    1 reference
    public class ProductController : Controller
    {
        private readonly ApplicationDbContext _db;

        0 references
        public ProductController(ApplicationDbContext db)
        {
            _db =db;
        }

        // GET: Product
        [Authorize(Roles = "Administrator")]
        0 references
        public IActionResult Index()
        {
            IEnumerable<Product> ProductList = _db.Products;
            return View(ProductList);
        }
```

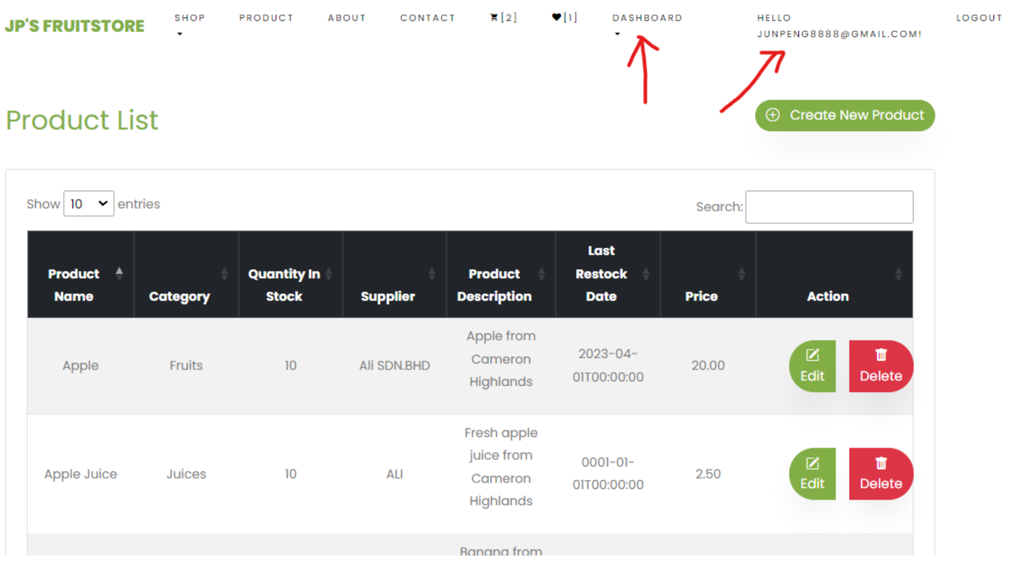
Figure 5.3.27 Code snippet to assign role for certain methods

viii. After implementing the role-based methods, an example will be given to demonstrate the different level of access by administrator and normal user. According to Figure 4.4.23, only JUNPENG8888@GMAIL.COM which has assigned an administrator role in database can access the product dashboard page in order to manage all the product details. For normal user like TEST@GMAIL.COM in Figure 4.4.24 will not be able to see the product dashboard navigation link but only can see a normal product display page. Even when users try to access the dashboard page by using the link, they are not able to do so due to access denied to its role as shown in Figure 4.4.25.



	Id	Name	NormalizedName	ConcurrencyStamp
1	1	Administrator	Admin	NULL

Figure 5.3.28 Administrator role being created in the database and others will be normal user by default



Product List

Product Name	Category	Quantity In Stock	Supplier	Product Description	Last Restock Date	Price	Action
Apple	Fruits	10	Ali SDN.BHD	Apple from Cameron Highlands	2023-04-01T00:00:00	20.00	Edit Delete
Apple Juice	Juices	10	ALI	Fresh apple juice from Cameron Highlands	0001-01-01T00:00:00	2.50	Edit Delete
				Banana from			

Figure 5.3.29 Administrator account view

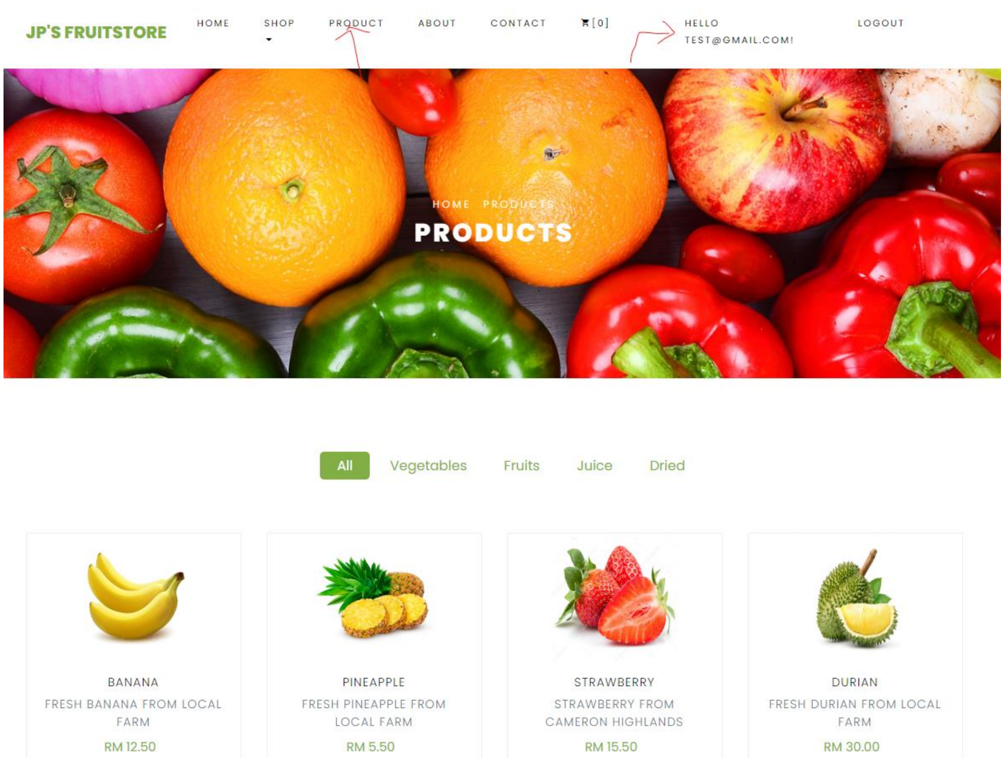
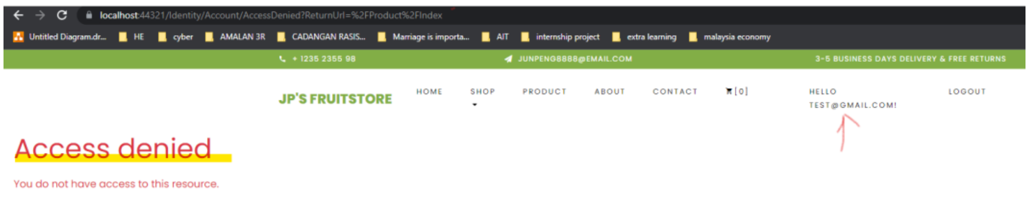


Figure 5.3.30 Normal account view


localhost:44321/Identity/Account/AccessDenied?ReturnUrl=%2FProduct%2FIndex

Untitled Diagram.dr... HE cyber AMALAN 3R CADANGAN RASIS... Marriage is importa... AIT internship project extra learning malaysia economy

+ 1235 2355 98 JUNPENG8888@EMAIL.COM 3-5 BUSINESS DAYS DELIVERY & FREE RETURNS

JP'S FRUITSTORE HOME SHOP PRODUCT ABOUT CONTACT [0] HELLO TEST@GMAIL.COM! LOGOUT

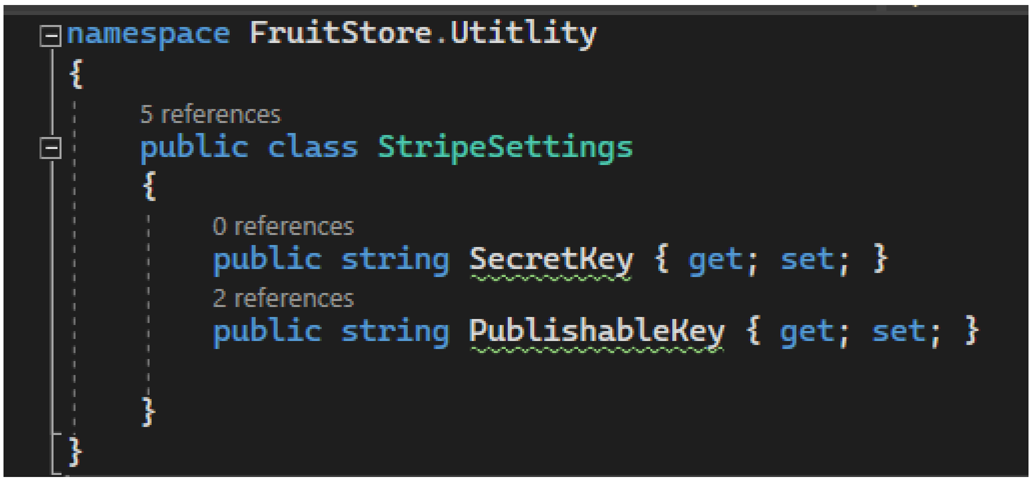
Access denied

You do not have access to this resource.

Figure 5.3.31 Access denied page if user tried to access the product dashboard using the link

5.3.4 Payment module using Stripe

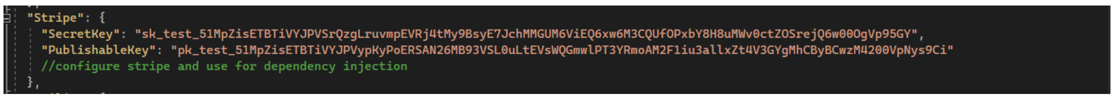
i. First, we need to create the StripeSettings configuration to create a service which stores the SecretKey and PublishableKey.



```
namespace FruitStore.Utitlity
{
    5 references
    public class StripeSettings
    {
        0 references
        public string SecretKey { get; set; }
        2 references
        public string PublishableKey { get; set; }

    }
}
```

Figure 5.3.32 StripeSettings configuration for payment option

ii. In the appsettings.json, we will store the value for SecretKey and PublishableKey which obtained from the Stripe account.



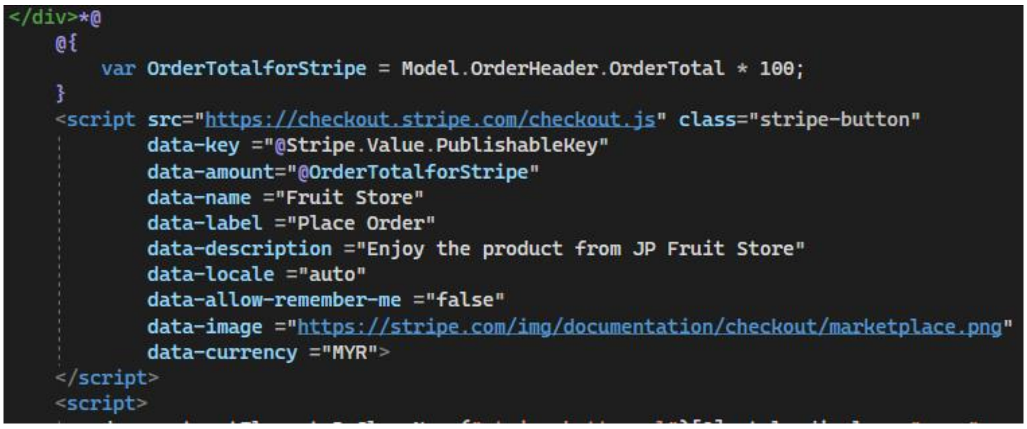
```
"Stripe": {
  "SecretKey": "sk_test_51MpZisETBTiVYJPVSrQzgLruvmpEVRj4tMy9BsyE7JchMMGUM6ViEQ6xw6M3CQUfOPxbY8H8uMWv0ctZOSrejQ6w00OgVp95GY",
  "PublishableKey": "pk_test_51MpZisETBTiVYJPVypKyPoERSAN26MB93VSL0uLtEVsWQGmwlPT3YRmoAM2F1iu3allxZt4V3GYgMhCByBCwzM4200VpNys9Ci"
  //configure stripe and use for dependency injection
},
```

Figure 5.3.33 appsettings.json configuration for stripe

iii. We can use dependency injection and configure the stripe settings by getting the key value from appsettings.json when the web application is being built with the code below. Noted that the steps are similar when configuring the Twilio option for sending order success message.

builder.Services.Configure<StripeSettings>(builder.Configuration.GetSection("Stripe"));

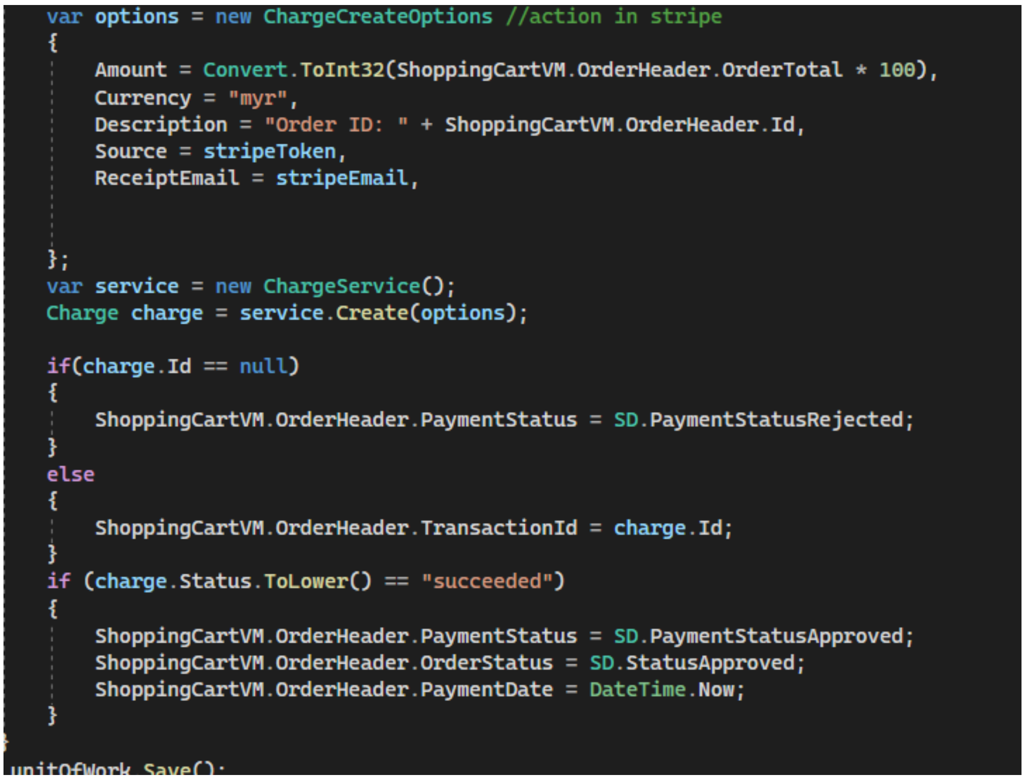
iv. Figure 5.3.34 shows the payment code on order summary view which can request customers to fill up the required card and personal information before making payment and the information will pass back to CartController to charge the customers for payment.



```
</div>*@
    @{
        var OrderTotalforStripe = Model.OrderHeader.OrderTotal * 100;
    }
    <script src="https://checkout.stripe.com/checkout.js" class="stripe-button"
            data-key ="@Stripe.Value.PublishableKey"
            data-amount="@OrderTotalforStripe"
            data-name ="Fruit Store"
            data-label ="Place Order"
            data-description ="Enjoy the product from JP Fruit Store"
            data-locale ="auto"
            data-allow-remember-me ="false"
            data-image ="https://stripe.com/img/documentation/checkout/marketplace.png"
            data-currency ="MYR">
    </script>
    <script>
```

Figure 5.3.34 payment section code

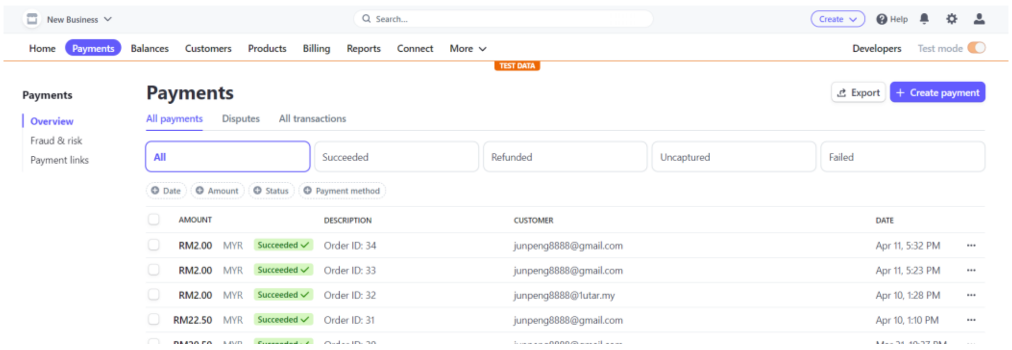
v. The information being passed to the CartController will be processed and callout the ChargeService() method to charge customer who had paid.



```
    var options = new ChargeCreateOptions //action in stripe
    {
        Amount = Convert.ToInt32(ShoppingCartVM.OrderHeader.OrderTotal * 100),
        Currency = "myr",
        Description = "Order ID: " + ShoppingCartVM.OrderHeader.Id,
        Source = stripeToken,
        ReceiptEmail = stripeEmail,


    };
    var service = new ChargeService();
    Charge charge = service.Create(options);

    if(charge.Id == null)
    {
        ShoppingCartVM.OrderHeader.PaymentStatus = SD.PaymentStatusRejected;
    }
    else
    {
        ShoppingCartVM.OrderHeader.TransactionId = charge.Id;
    }
    if (charge.Status.ToLower() == "succeeded")
    {
        ShoppingCartVM.OrderHeader.PaymentStatus = SD.PaymentStatusApproved;
        ShoppingCartVM.OrderHeader.OrderStatus = SD.StatusApproved;
        ShoppingCartVM.OrderHeader.PaymentDate = DateTime.Now;
    }
}
_unitOfWork.Save();
```

Figure 5.3.35 CartController code to call the ChargeService() method

vi. After customer being charged for the payment, the record can be checked on the Stripe website.



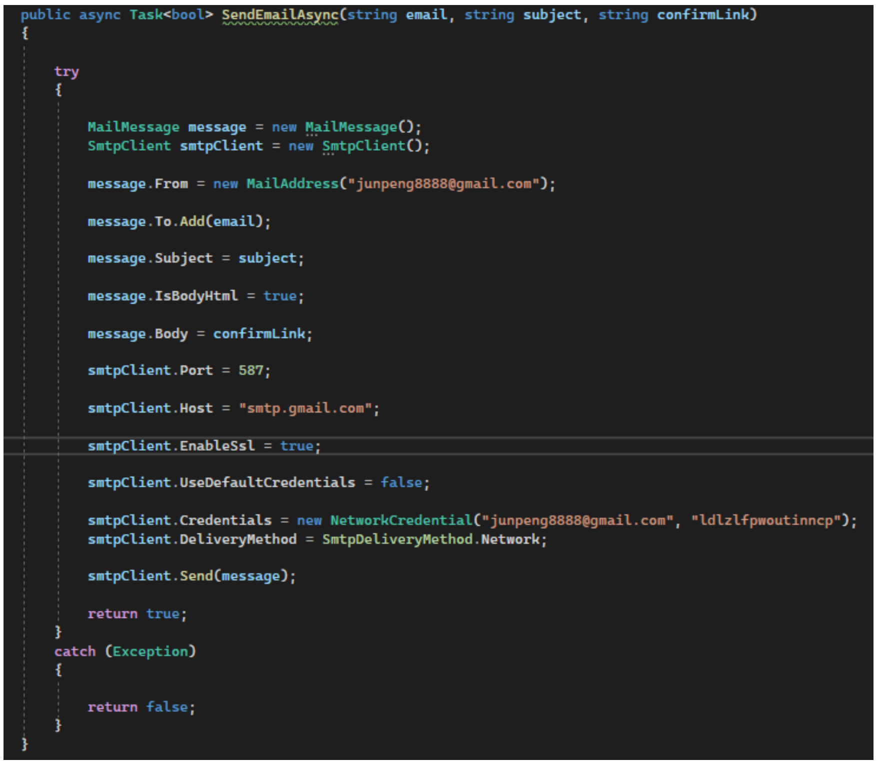
Payments

AMOUNT			DESCRIPTION	CUSTOMER	DATE
RM2.00	MYR	Succeeded ✓	Order ID: 34	junpeng8888@gmail.com	Apr 11, 5:32 PM
RM2.00	MYR	Succeeded ✓	Order ID: 33	junpeng8888@gmail.com	Apr 11, 5:23 PM
RM2.00	MYR	Succeeded ✓	Order ID: 32	junpeng8888@1utar.my	Apr 10, 1:28 PM
RM22.50	MYR	Succeeded ✓	Order ID: 31	junpeng8888@gmail.com	Apr 10, 1:10 PM

Figure 5.3.36 Stripe Website to check the payment record

5.3.5 Email verification using HTML template

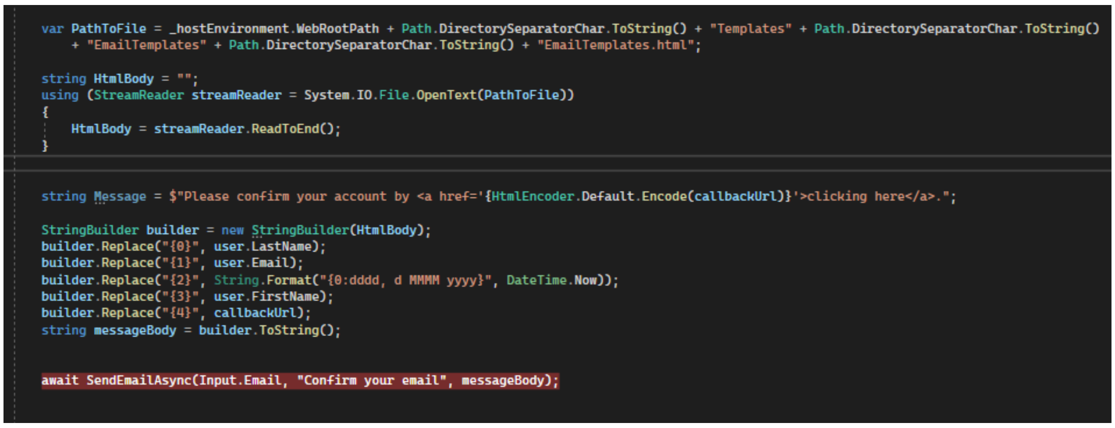
i. When the user completely filled up the sign-up form and click the sign-up button, the Register.cshtml.cs page will call the SendEmailAsync() method to send an email verification to the user's email. Figure 5.3.37 shows the configuration of the smtp server information like port, host name, subject and the message.



```
public async Task<bool> SendEmailAsync(string email, string subject, string confirmLink)
{

    try
    {

        MailMessage message = new MailMessage();
        SmtpClient smtpClient = new SmtpClient();

        message.From = new MailAddress("junpeng8888@gmail.com");

        message.To.Add(email);

        message.Subject = subject;

        message.IsBodyHtml = true;

        message.Body = confirmLink;

        smtpClient.Port = 587;

        smtpClient.Host = "smtp.gmail.com";

        smtpClient.EnableSsl = true;

        smtpClient.UseDefaultCredentials = false;

        smtpClient.Credentials = new NetworkCredential("junpeng8888@gmail.com", "ldlzlfpwoutinncp");
        smtpClient.DeliveryMethod = SmtpDeliveryMethod.Network;

        smtpClient.Send(message);

        return true;
    }
    catch (Exception)
    {

        return false;
    }
}
```

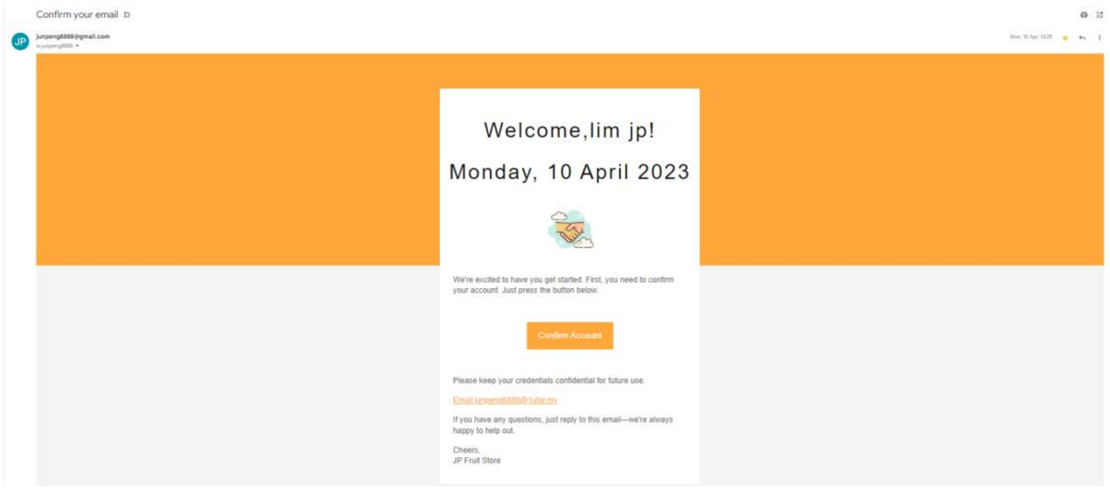
Figure 5.3.37 SendEmailAsync() method to configure the smtp server

ii. Create EmailTemplates.html as shown in Figure 5.3.39 for the email verification template and pass the SendEmailAsync() method as the messageBody.



```
var PathToFile = _hostEnvironment.WebRootPath + Path.DirectorySeparatorChar.ToString() + "Templates" + Path.DirectorySeparatorChar.ToString()
    + "EmailTemplates" + Path.DirectorySeparatorChar.ToString() + "EmailTemplates.html";

string HtmlBody = "";
using (StreamReader streamReader = System.IO.File.OpenText(PathToFile))
{
    HtmlBody = streamReader.ReadToEnd();
}

string Message = $"Please confirm your account by <a href='{HtmlEncoder.Default.Encode(callbackUrl)}'>clicking here</a>.";

StringBuilder builder = new StringBuilder(HtmlBody);
builder.Replace("{0}", user.LastName);
builder.Replace("{1}", user.Email);
builder.Replace("{2}", String.Format("{0:dddd, d MMMM yyyy}", DateTime.Now));
builder.Replace("{3}", user.FirstName);
builder.Replace("{4}", callbackUrl);
string messageBody = builder.ToString();

await SendEmailAsync(Input.Email, "Confirm your email", messageBody);
```

Figure 5.3.38 pass the EmailTemplates.html to SendEmailAsync() method



Confirm your email

junpeng8888@gmail.com
to junpeng8888

Welcome,lim jp!

Monday, 10 April 2023

We're excited to have you get started. First, you need to confirm your account. Just press the button below.

Confirm Account

Please keep your credentials confidential for future use.

Email:junpeng8888@1utar.my

If you have any questions, just reply to this email—we're always happy to help out.

Cheers,
JP Fruit Store

Figure 5.3.39 EmailTemplates.html

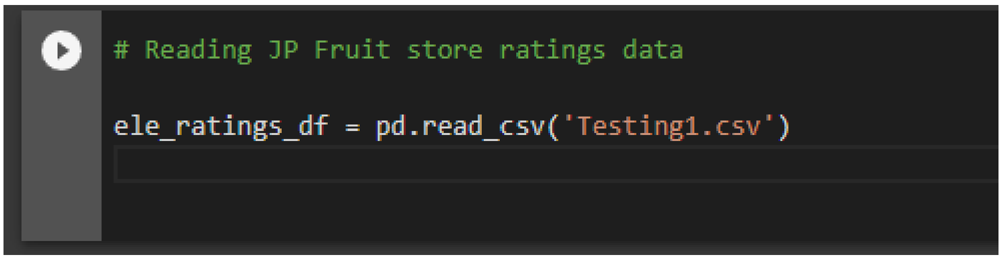
iii. It is noteworthy that this approach is also applicable when users subscribe to the newsletter on the proposed web application.

5.3.6 Implementation of proposed recommender model (User-based Collaborative Filtering)

i. These are the software and packages being installed and downloaded in my laptop for recommender model development:

```
from flask import Flask,render_template,request,jsonify
import numpy as np
import pandas as pd
import pickle
from gevent.pywsgi import WSGIServer
from sklearn.model_selection import train_test_split
from surprise.model_selection import train_test_split
from scipy.sparse.linalg import svds
from surprise import Dataset,Reader
from surprise import SVD, KNNWithMeans
from surprise import accuracy
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.neighbors import NearestNeighbors
from sklearn.cluster import KMeans
from sklearn.metrics import adjusted_rand_score
```

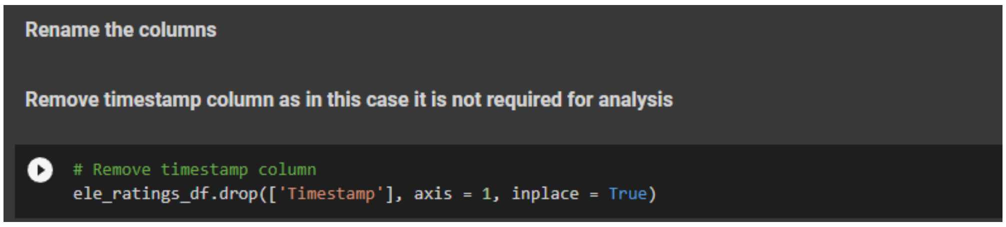
ii. Load the data from the dataset (JP's Fruit store rating data)



```
# Reading JP Fruit store ratings data

ele_ratings_df = pd.read_csv('Testing1.csv')
```

Figure 5.3.40 Loading data from dataset and create the dataframe

iii. Perform data cleaning and data preprocessing like removal of timestamp column from the dataframe.



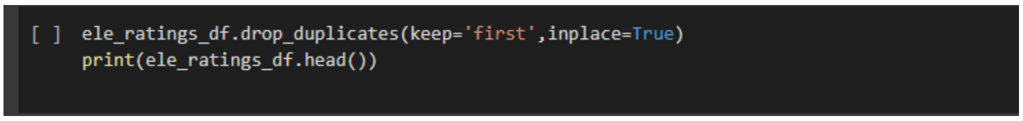
Rename the columns

Remove timestamp column as in this case it is not required for analysis

```
# Remove timestamp column
ele_ratings_df.drop(['Timestamp'], axis = 1, inplace = True)
```

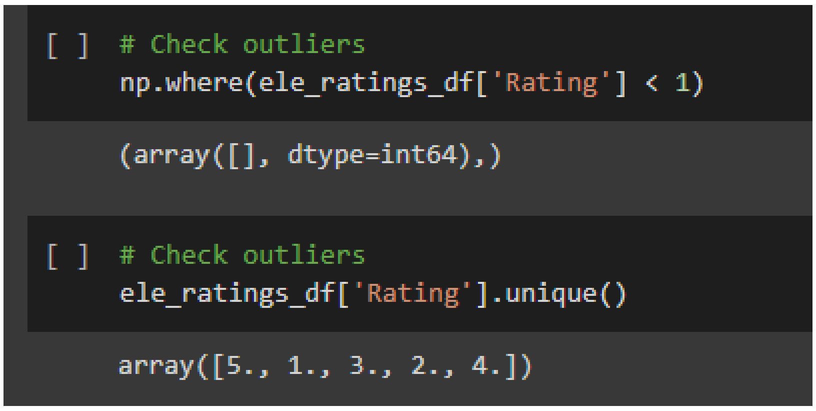
Figure 5.3.41 Loading data from dataset and create the dataframe

iv. Remove the duplicated rows if any and always check for null values and outliers. Assuming the ProductRatings table will check for null value so it is not necessary to perform null removal here.



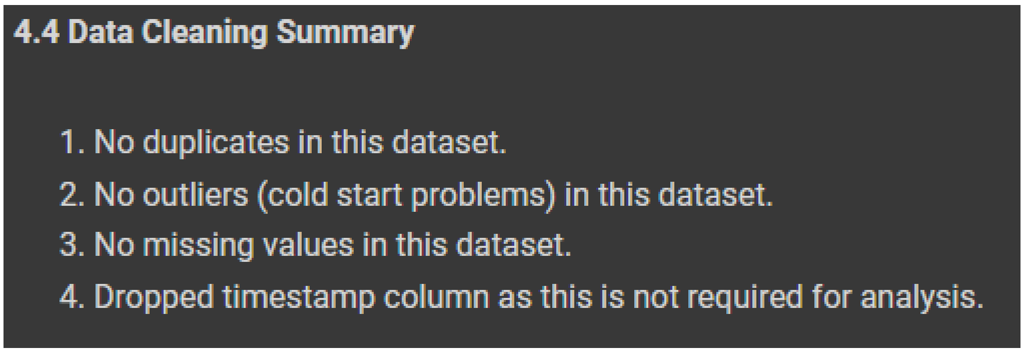
```
[ ]  ele_ratings_df.drop_duplicates(keep='first',inplace=True)
     print(ele_ratings_df.head())
```

Figure 5.3.42 Remove duplicate rows



```
[ ]  # Check outliers
     np.where(ele_ratings_df['Rating'] < 1)

     (array([], dtype=int64),)

[ ]  # Check outliers
     ele_ratings_df['Rating'].unique()

     array([5., 1., 3., 2., 4.])
```

Figure 5.3.43 Check the outliers

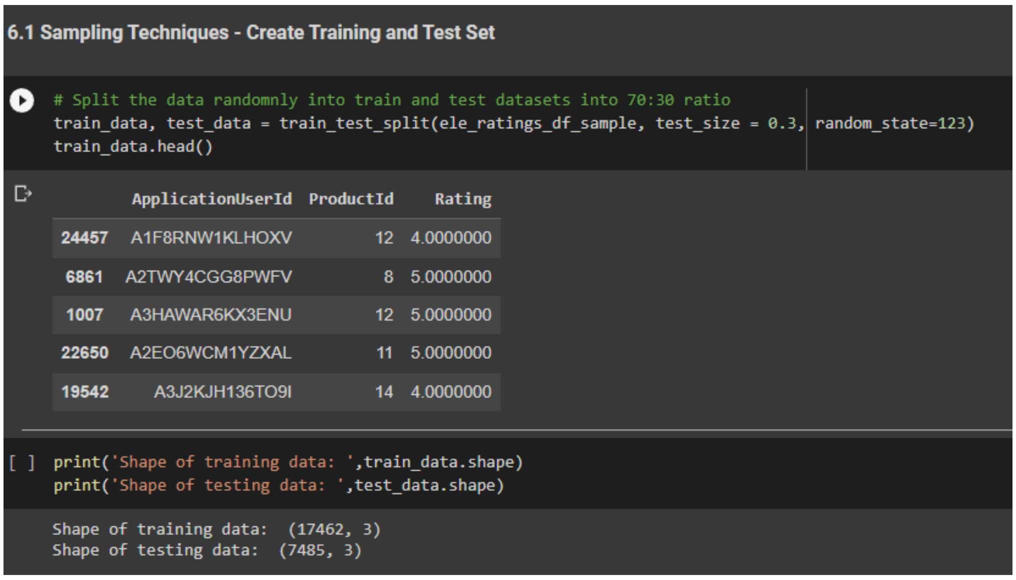


4.4 Data Cleaning Summary

1. No duplicates in this dataset.
2. No outliers (cold start problems) in this dataset.
3. No missing values in this dataset.
4. Dropped timestamp column as this is not required for analysis.

Figure 5.3.44 Data cleaning summary

v. Perform the data spitting into training and testing set with 7:3 ratio and the shape of training and testing data is being shown.



6.1 Sampling Techniques - Create Training and Test Set

```
# Split the data randomnly into train and test datasets into 70:30 ratio
train_data, test_data = train_test_split(ele_ratings_df_sample, test_size = 0.3, random_state=123)
train_data.head()
```

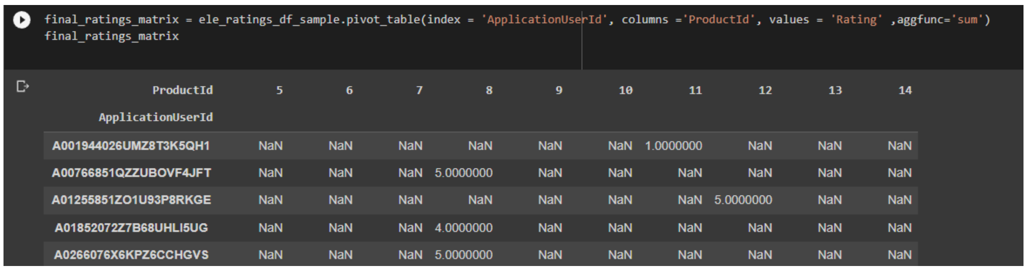
	ApplicationUserId	ProductId	Rating
24457	A1F8RNW1KLHOXV	12	4.0000000
6861	A2TWY4CGG8PWFV	8	5.0000000
1007	A3HAWAR6KX3ENU	12	5.0000000
22650	A2EO6WCM1YZXAL	11	5.0000000
19542	A3J2KJH136TO9I	14	4.0000000

```
print('Shape of training data: ',train_data.shape)
print('Shape of testing data: ',test_data.shape)
```

```
Shape of training data:  (17462, 3)
Shape of testing data:  (7485, 3)
```

Figure 5.3.45 Split the data into training and testing sets

vi. Convert the data into rating matrix using pivot_table() method where index is the ApplicationUserId and column is the ProdutId and the value is the actual ratings of product being rated by the user. As shown in the figure below, there are NaN values for products which are not rated by users.

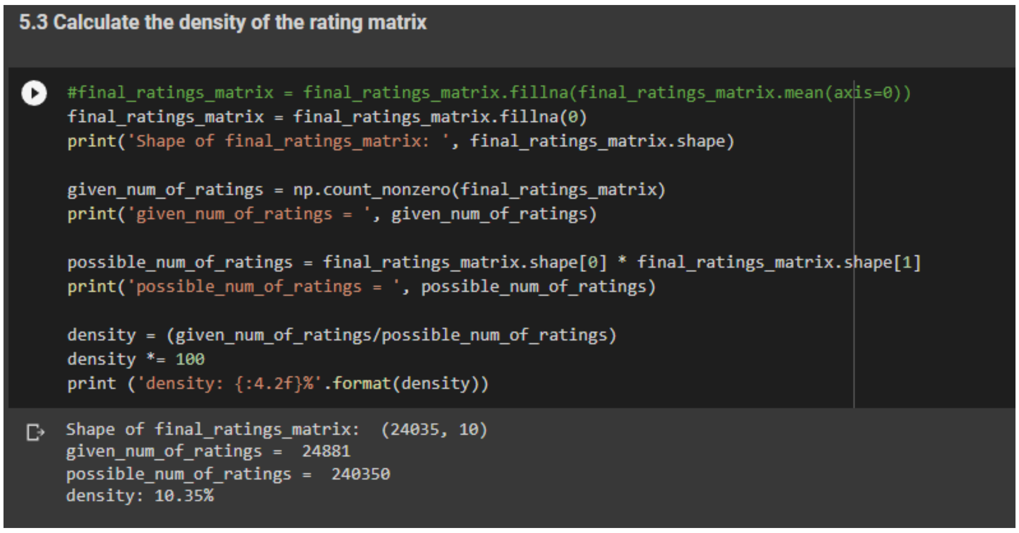


```
final_ratings_matrix = ele_ratings_df_sample.pivot_table(index = 'ApplicationUserId', columns ='ProductId', values = 'Rating' ,aggfunc='sum')
final_ratings_matrix
```

ProductId	5	6	7	8	9	10	11	12	13	14
ApplicationUserId										
A001944026UMZ8T3K5QH1	NaN	NaN	NaN	NaN	NaN	NaN	1.0000000	NaN	NaN	NaN
A00766851QZZUBOVF4JFT	NaN	NaN	NaN	5.0000000	NaN	NaN	NaN	NaN	NaN	NaN
A01255851ZO1U93P8RKGE	NaN	NaN	NaN	NaN	NaN	NaN	NaN	5.0000000	NaN	NaN
A01852072Z7B68UHLI5UG	NaN	NaN	NaN	4.0000000	NaN	NaN	NaN	NaN	NaN	NaN
A0266076X6KPZ6CCHGVS	NaN	NaN	NaN	5.0000000	NaN	NaN	NaN	NaN	NaN	NaN

Figure 5.3.46 final_ratings_matrix

vii. Calculate the density of rating matrix and we can see the result below. Besides, fill up all the NaN values with value 0. The rating matrix clearly shows the relationship between the product and user by ratings.

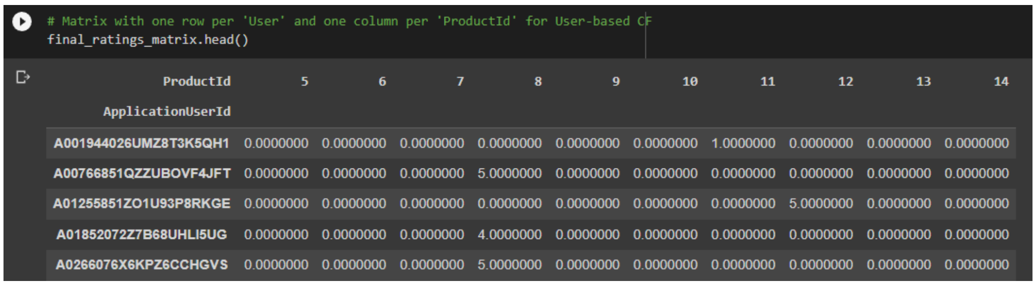


5.3 Calculate the density of the rating matrix

```
#final_ratings_matrix = final_ratings_matrix.fillna(final_ratings_matrix.mean(axis=0))
final_ratings_matrix = final_ratings_matrix.fillna(0)
print('Shape of final_ratings_matrix: ', final_ratings_matrix.shape)

given_num_of_ratings = np.count_nonzero(final_ratings_matrix)
print('given_num_of_ratings = ', given_num_of_ratings)

possible_num_of_ratings = final_ratings_matrix.shape[0] * final_ratings_matrix.shape[1]
print('possible_num_of_ratings = ', possible_num_of_ratings)

density = (given_num_of_ratings/possible_num_of_ratings)
density *= 100
print ('density: {:4.2f}%'.format(density))
```

```
Shape of final_ratings_matrix:  (24035, 10)
given_num_of_ratings =  24881
possible_num_of_ratings =  240350
density: 10.35%
```

Figure 5.3.47 Fill up NaN values with 0 and show the density of rating matrix

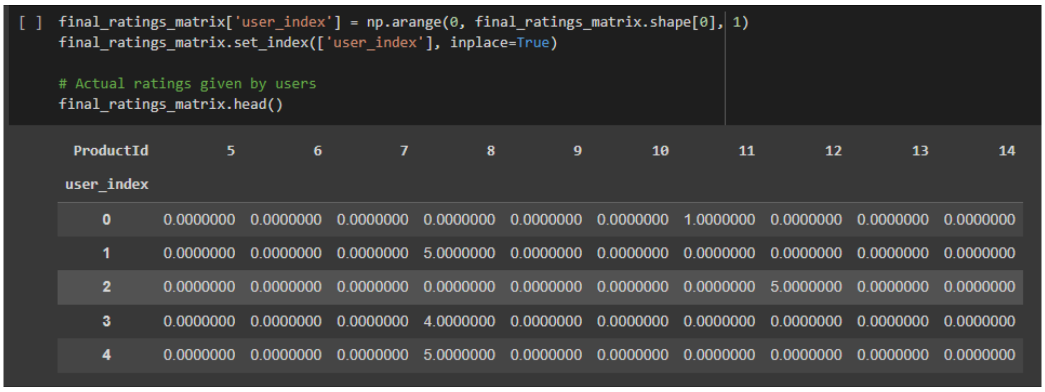


```
# Matrix with one row per 'User' and one column per 'ProductId' for User-based CF
final_ratings_matrix.head()
```

ProductId	5	6	7	8	9	10	11	12	13	14
ApplicationUserId										
A001944026UMZ8T3K5QH1	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000	0.0000000	0.0000000	0.0000000
A00766851QZZUBOVF4JFT	0.0000000	0.0000000	0.0000000	5.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
A01255851ZO1U93P8RKGE	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	5.0000000	0.0000000	0.0000000
A01852072Z7B68UHLI5UG	0.0000000	0.0000000	0.0000000	4.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
A0266076X6KPZ6CCHGVS	0.0000000	0.0000000	0.0000000	5.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000

Figure 5.3.48 final_ratings_matrix

viii. Define user index for each row of user and replace the ApplicationUserId with the user index. As shown in the figure below, it was a sparse matrix with low density, so SVD is one of the approaches to apply on this sparse matrix.



```
final_ratings_matrix['user_index'] = np.arange(0, final_ratings_matrix.shape[0], 1)
final_ratings_matrix.set_index(['user_index'], inplace=True)

# Actual ratings given by users
final_ratings_matrix.head()
```

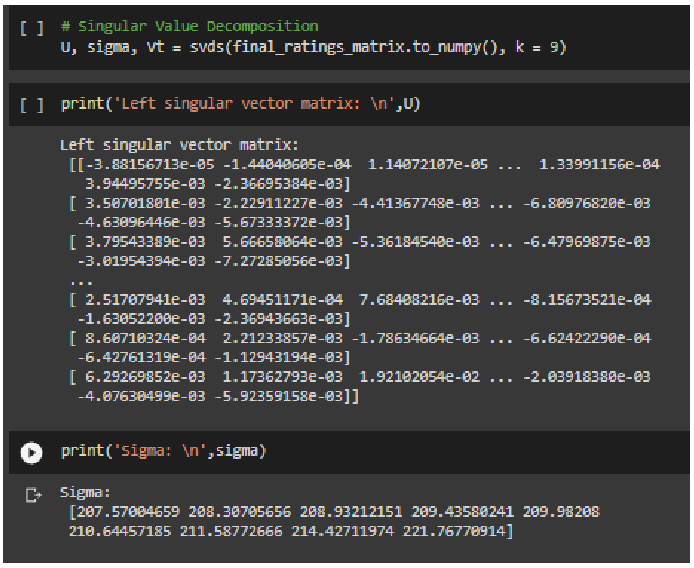
ProductId	5	6	7	8	9	10	11	12	13	14
user_index										
0	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	1.0000000	0.0000000	0.0000000	0.0000000
1	0.0000000	0.0000000	0.0000000	5.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
2	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	5.0000000	0.0000000	0.0000000
3	0.0000000	0.0000000	0.0000000	4.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
4	0.0000000	0.0000000	0.0000000	5.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000

Figure 5.3.49 Replace the ApplicationUserId with the user index

ix. Singular value decomposition (SVD) is one of the famous matrix decomposition techniques to break down a matrix into 3 other matrices and reduce the dimensionality as you can in the formula in figure shown below [25]. It is mainly used to discover the relationship between users and products.

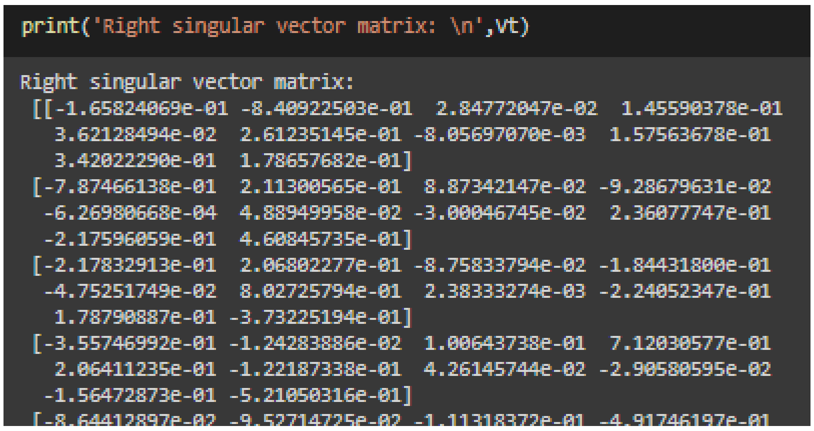
$$A = U\Sigma V^T$$

Figure 5.3.50 Singular value decomposition (SVD) formula



```
[ ]  # Singular Value Decomposition
     U, sigma, Vt = svds(final_ratings_matrix.to_numpy(), k = 9)

[ ]  print('Left singular vector matrix: \n',U)

     Left singular vector matrix:
      [[-3.88156713e-05 -1.44040605e-04  1.14072107e-05 ...  1.33991156e-04
        3.94495755e-03 -2.36695384e-03]
      [ 3.50701801e-03 -2.22911227e-03 -4.41367748e-03 ... -6.80976820e-03
       -4.63096446e-03 -5.67333372e-03]
      [ 3.79543389e-03  5.66658064e-03 -5.36184540e-03 ... -6.47969875e-03
       -3.01954394e-03 -7.27285056e-03]
      ...
      [ 2.51707941e-03  4.69451171e-04  7.68408216e-03 ... -8.15673521e-04
       -1.63052200e-03 -2.36943663e-03]
      [ 8.60710324e-04  2.21233857e-03 -1.78634664e-03 ... -6.62422290e-04
       -6.42761319e-04 -1.12943194e-03]
      [ 6.29269852e-03  1.17362793e-03  1.92102054e-02 ... -2.03918380e-03
       -4.07630499e-03 -5.92359158e-03]]

     print('Sigma: \n',sigma)

     Sigma:
      [207.57004659 208.30705656 208.93212151 209.43580241 209.98208
      210.64457185 211.58772666 214.42711974 221.76770914]
```

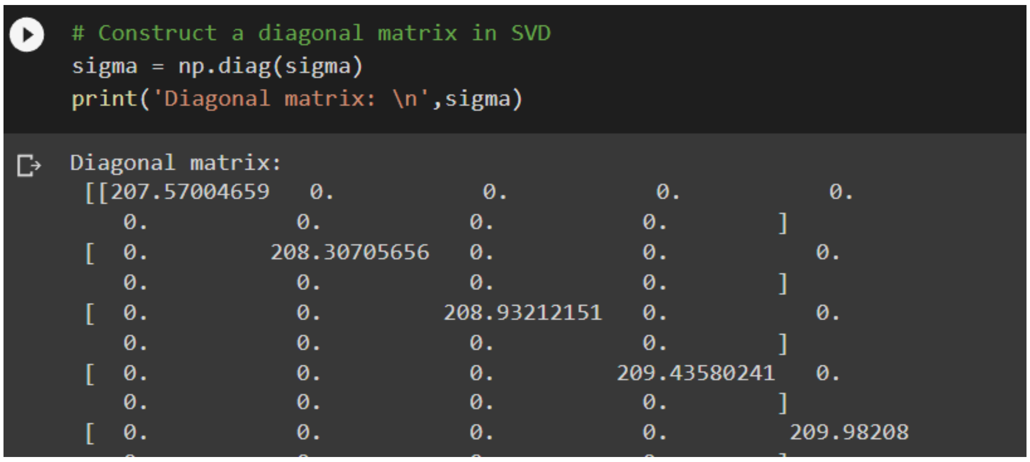
Figure 5.3.51 Singular value decomposition (SVD) code and each matrix after decomposed



```
print('Right singular vector matrix: \n',Vt)

Right singular vector matrix:
 [[-1.65824069e-01 -8.40922503e-01  2.84772047e-02  1.45590378e-01
   3.62128494e-02  2.61235145e-01 -8.05697070e-03  1.57563678e-01
   3.42022290e-01  1.78657682e-01]
 [-7.87466138e-01  2.11300565e-01  8.87342147e-02 -9.28679631e-02
  -6.26980668e-04  4.88949958e-02 -3.00046745e-02  2.36077747e-01
  -2.17596059e-01  4.60845735e-01]
 [-2.17832913e-01  2.06802277e-01 -8.75833794e-02 -1.84431800e-01
  -4.75251749e-02  8.02725794e-01  2.38333274e-03 -2.24052347e-01
   1.78790887e-01 -3.73225194e-01]
 [-3.55746992e-01 -1.24283886e-02  1.00643738e-01  7.12030577e-01
   2.06411235e-01 -1.22187338e-01  4.26145744e-02 -2.90580595e-02
  -1.56472873e-01 -5.21050316e-01]
 [-8.64412897e-02 -9.52714725e-02 -1.11318372e-01 -4.91746197e-01
```

Figure 5.3.52 Right singular vector matrix

x. As the sigma in SVD must be in diagonal, use the np.diag() to convert it into diagonal matrix.

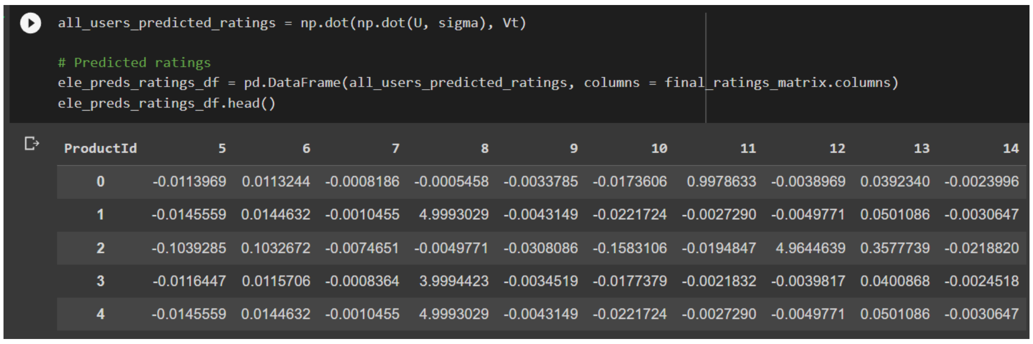


```
# Construct a diagonal matrix in SVD
sigma = np.diag(sigma)
print('Diagonal matrix: \n',sigma)
```

```
Diagonal matrix:
 [[207.57004659   0.           0.           0.           0.
    0.           0.           0.           0.         ]
 [  0.         208.30705656   0.           0.           0.
    0.           0.           0.           0.         ]
 [  0.           0.         208.93212151   0.           0.
    0.           0.           0.           0.         ]
 [  0.           0.           0.         209.43580241   0.
    0.           0.           0.           0.         ]
 [  0.           0.           0.           0.         209.98208
```

Figure 5.3.53 Convert sigma into diagonal matrix

xi. Then, predict the user rating of the user index against each ProductId and convert it into dataframe.

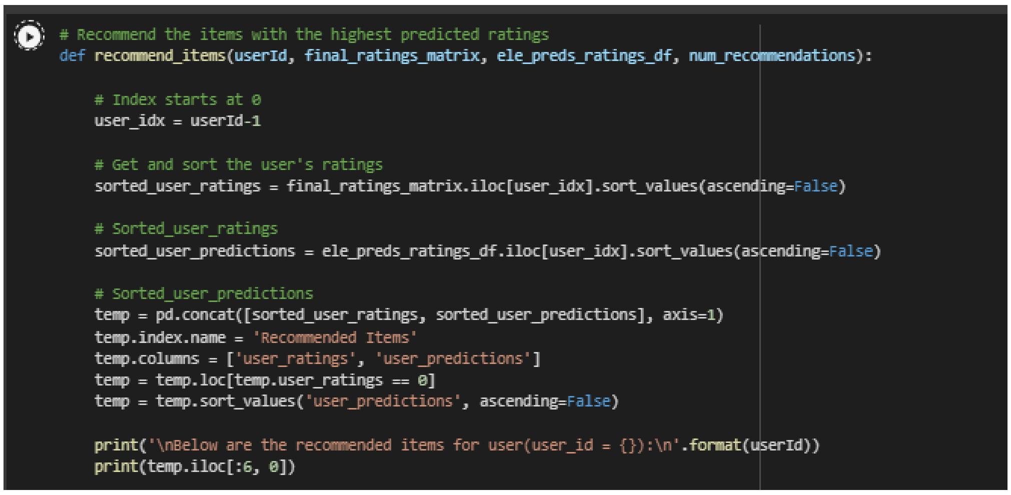


```
all_users_predicted_ratings = np.dot(np.dot(U, sigma), Vt)

# Predicted ratings
ele_preds_ratings_df = pd.DataFrame(all_users_predicted_ratings, columns = final_ratings_matrix.columns)
ele_preds_ratings_df.head()
```

ProductId	5	6	7	8	9	10	11	12	13	14
0	-0.0113969	0.0113244	-0.0008186	-0.0005458	-0.0033785	-0.0173606	0.9978633	-0.0038969	0.0392340	-0.0023996
1	-0.0145559	0.0144632	-0.0010455	4.9993029	-0.0043149	-0.0221724	-0.0027290	-0.0049771	0.0501086	-0.0030647
2	-0.1039285	0.1032672	-0.0074651	-0.0049771	-0.0308086	-0.1583106	-0.0194847	4.9644639	0.3577739	-0.0218820
3	-0.0116447	0.0115706	-0.0008364	3.9994423	-0.0034519	-0.0177379	-0.0021832	-0.0039817	0.0400868	-0.0024518
4	-0.0145559	0.0144632	-0.0010455	4.9993029	-0.0043149	-0.0221724	-0.0027290	-0.0049771	0.0501086	-0.0030647

Figure 5.3.54 Dataframe of user rating of the user index against each ProductId

xii. Sort the user predicted ratings and recommend the top-N products which have the higher user-predicted ratings suggesting to user.

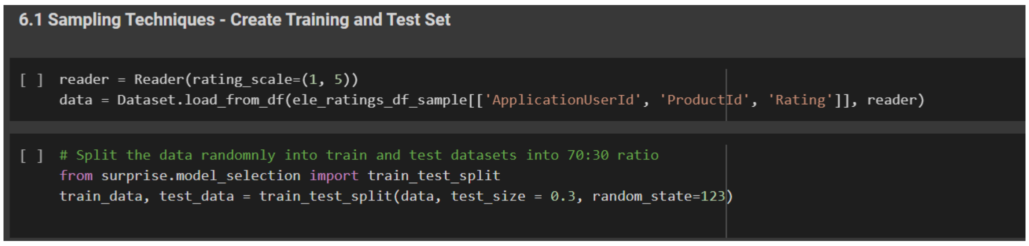


```
# Recommend the items with the highest predicted ratings
def recommend_items(userId, final_ratings_matrix, ele_preds_ratings_df, num_recommendations):

    # Index starts at 0
    user_idx = userId-1

    # Get and sort the user's ratings
    sorted_user_ratings = final_ratings_matrix.iloc[user_idx].sort_values(ascending=False)

    # Sorted_user_ratings
    sorted_user_predictions = ele_preds_ratings_df.iloc[user_idx].sort_values(ascending=False)

    # Sorted_user_predictions
    temp = pd.concat([sorted_user_ratings, sorted_user_predictions], axis=1)
    temp.index.name = 'Recommended Items'
    temp.columns = ['user_ratings', 'user_predictions']
    temp = temp.loc[temp.user_ratings == 0]
    temp = temp.sort_values('user_predictions', ascending=False)

    print('\nBelow are the recommended items for user(user_id = {}):\n'.format(userId))
    print(temp.iloc[:6, 0])
```

Figure 5.3.55 Generate product recommendations

5.3.7 Implementation of proposed recommender model (Item-based Collaborative Filtering)

i. The data loading, pre-processing and other steps are like the previous model starting from step i-iv. This time another sampling technique is used from the surprise library. As usual, split the data randomly into train and test datasets into 70:30 ratios.



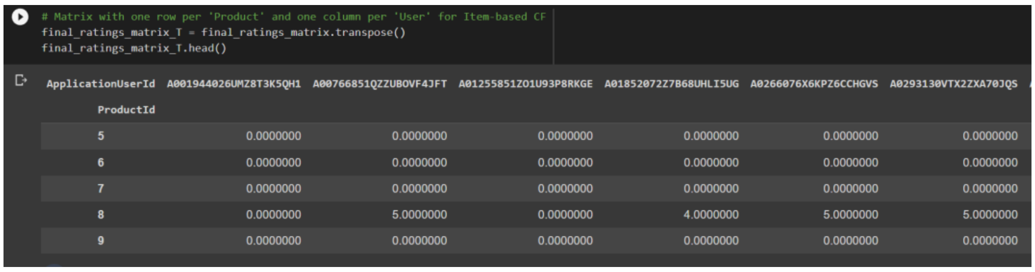
6.1 Sampling Techniques - Create Training and Test Set

```
[ ] reader = Reader(rating_scale=(1, 5))
    data = Dataset.load_from_df(ele_ratings_df_sample[['ApplicationUserId', 'ProductId', 'Rating']], reader)
```

```
[ ] # Split the data randomnly into train and test datasets into 70:30 ratio
    from surprise.model_selection import train_test_split
    train_data, test_data = train_test_split(data, test_size = 0.3, random_state=123)
```

Figure 5.3.56 Split the data into training and testing sets

ii. For item-based recommender, transpose the matrix in the previous section with ProductId as the row and ApplicationUserId as the column and ratings as the values by using numpy .transpose() method.

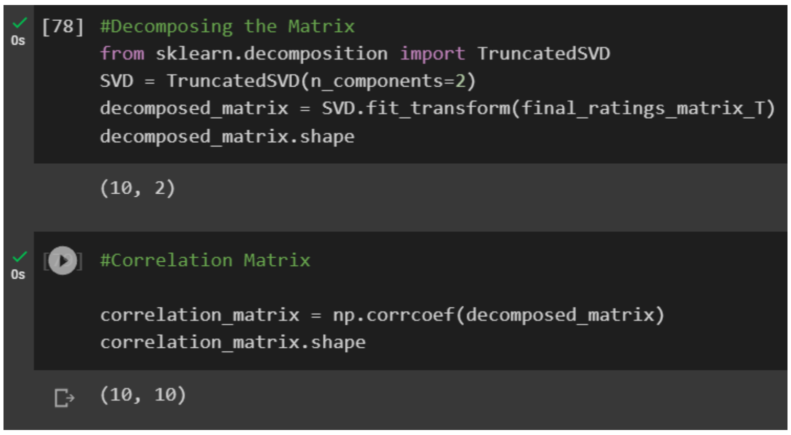


```
# Matrix with one row per 'Product' and one column per 'User' for Item-based CF
final_ratings_matrix_T = final_ratings_matrix.transpose()
final_ratings_matrix_T.head()
```

```
ApplicationUserId  A001944026UMZ8T3K5QH1  A00766851QZZUBOVF4JFT  A01255851ZO1U93P8RKGE  A01852072Z7B68UHLI5UG  A0266076X6KPZ6CCHGVS  A0293130VTX2ZXA70JQS
        ProductId
        5                      0.0000000              0.0000000              0.0000000              0.0000000             0.0000000             0.0000000
        6                      0.0000000              0.0000000              0.0000000              0.0000000             0.0000000             0.0000000
        7                      0.0000000              0.0000000              0.0000000              0.0000000             0.0000000             0.0000000
        8                      0.0000000              5.0000000              0.0000000              4.0000000             5.0000000             5.0000000
        9                      0.0000000              0.0000000              0.0000000              0.0000000             0.0000000             0.0000000
```

Figure 5.3.57 final_ratings_matrix_T after transposed

iii. As usual, decompose the final_ratings_matrix_T but this time use the TruncatedSVD technique from sklearn library to reduce the dimensionality. Besides, use Pearson Correlation to check the similarity between the products this time using numpy np.corrcoef() method.

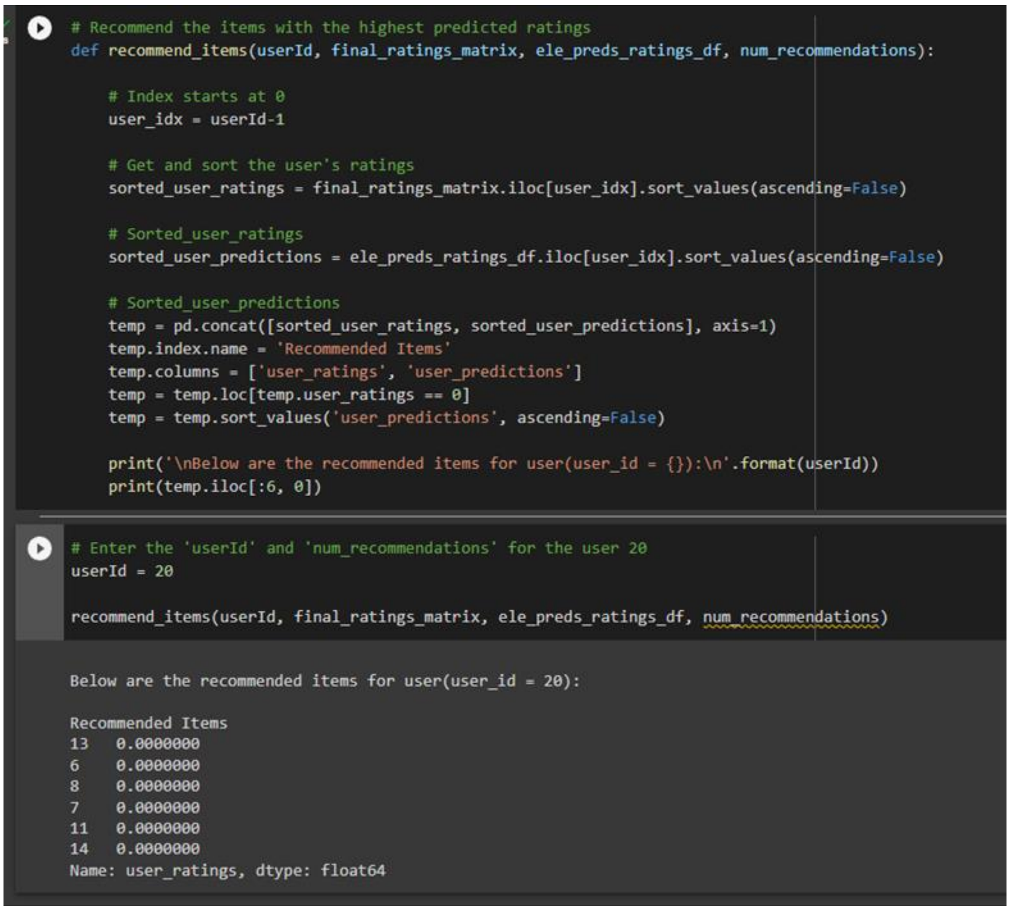


```
[78] #Decomposing the Matrix
     from sklearn.decomposition import TruncatedSVD
     SVD = TruncatedSVD(n_components=2)
     decomposed_matrix = SVD.fit_transform(final_ratings_matrix_T)
     decomposed_matrix.shape

     (10, 2)
```

```
     #Correlation Matrix

     correlation_matrix = np.corrcoef(decomposed_matrix)
     correlation_matrix.shape

     (10, 10)
```

Figure 5.3.58 correlation_matrix

iv. Take the product_ID and based on that, recommend the products which has the correlation coefficient up to 0.90 and above which indicates that it is a strong positive correlation strength with the product_ID input. This will tell that user A who rates product A with a high rating will most likely also like the products being recommended by the recommender.



```
# Recommend the items with the highest predicted ratings
def recommend_items(userId, final_ratings_matrix, ele_preds_ratings_df, num_recommendations):

    # Index starts at 0
    user_idx = userId-1

    # Get and sort the user's ratings
    sorted_user_ratings = final_ratings_matrix.iloc[user_idx].sort_values(ascending=False)

    # Sorted_user_ratings
    sorted_user_predictions = ele_preds_ratings_df.iloc[user_idx].sort_values(ascending=False)

    # Sorted_user_predictions
    temp = pd.concat([sorted_user_ratings, sorted_user_predictions], axis=1)
    temp.index.name = 'Recommended Items'
    temp.columns = ['user_ratings', 'user_predictions']
    temp = temp.loc[temp.user_ratings == 0]
    temp = temp.sort_values('user_predictions', ascending=False)

    print('\nBelow are the recommended items for user(user_id = {}):\n'.format(userId))
    print(temp.iloc[:6, 0])
```

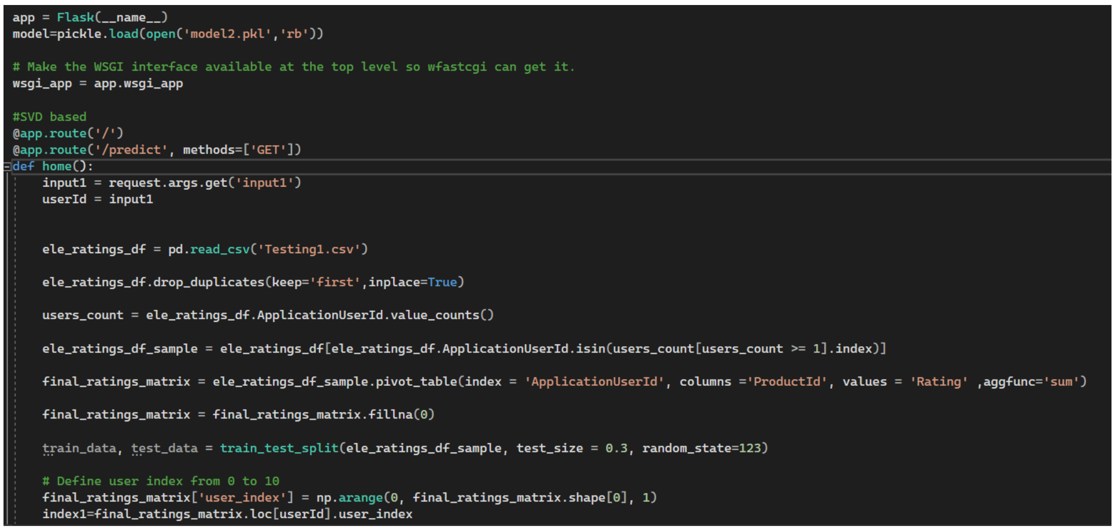
```
# Enter the 'userId' and 'num_recommendations' for the user 20
userId = 20

recommend_items(userId, final_ratings_matrix, ele_preds_ratings_df, num_recommendations)
```

```
Below are the recommended items for user(user_id = 20):

Recommended Items
13   0.0000000
6    0.0000000
8    0.0000000
7    0.0000000
11   0.0000000
14   0.0000000
Name: user_ratings, dtype: float64
```

Figure 5.3.59 Generate product recommendations for user_id=20

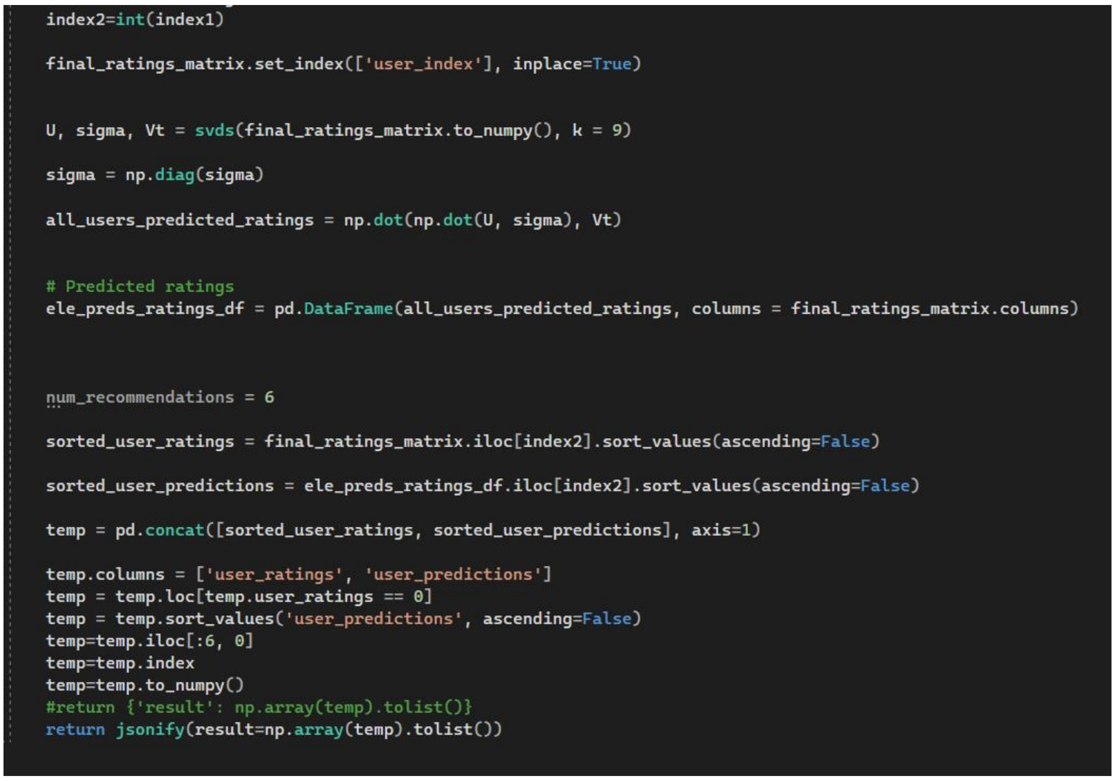
5.3.8 Deployment of proposed recommender model to flask and pass the result back to web application through HTTP Request

i. The two recommender models in the previous section will be transferred to Flask which acts as Restful API and ready for deployment. Firstly, it is important to create the routes in which the web application would route to pass the input. Figure 5.3.60 below shows the app.route(“/predict”) is defined so that the web application will know where to route where there is a “GET” request incoming. Besides, the input is provided from the MVC web application before running the prediction and the output will be returned to the web application in JSON format int array of ProductId as shown in Figure 5.3.61. Noted that the below example is for user-based collaborative filtering recommender model.



```
app = Flask(__name__)
model=pickle.load(open('model2.pkl','rb'))

# Make the WSGI interface available at the top level so wfastcgi can get it.
wsgi_app = app.wsgi_app

#SVD based
@app.route('/')
@app.route('/predict', methods=['GET'])
def home():
    input1 = request.args.get('input1')
    userId = input1


    ele_ratings_df = pd.read_csv('Testing1.csv')

    ele_ratings_df.drop_duplicates(keep='first',inplace=True)

    users_count = ele_ratings_df.ApplicationUserId.value_counts()

    ele_ratings_df_sample = ele_ratings_df[ele_ratings_df.ApplicationUserId.isin(users_count[users_count >= 1].index)]

    final_ratings_matrix = ele_ratings_df_sample.pivot_table(index = 'ApplicationUserId', columns ='ProductId', values = 'Rating' ,aggfunc='sum')

    final_ratings_matrix = final_ratings_matrix.fillna(0)

    train_data, test_data = train_test_split(ele_ratings_df_sample, test_size = 0.3, random_state=123)

    # Define user index from 0 to 10
    final_ratings_matrix['user_index'] = np.arange(0, final_ratings_matrix.shape[0], 1)
    index1=final_ratings_matrix.loc[userId].user_index
```

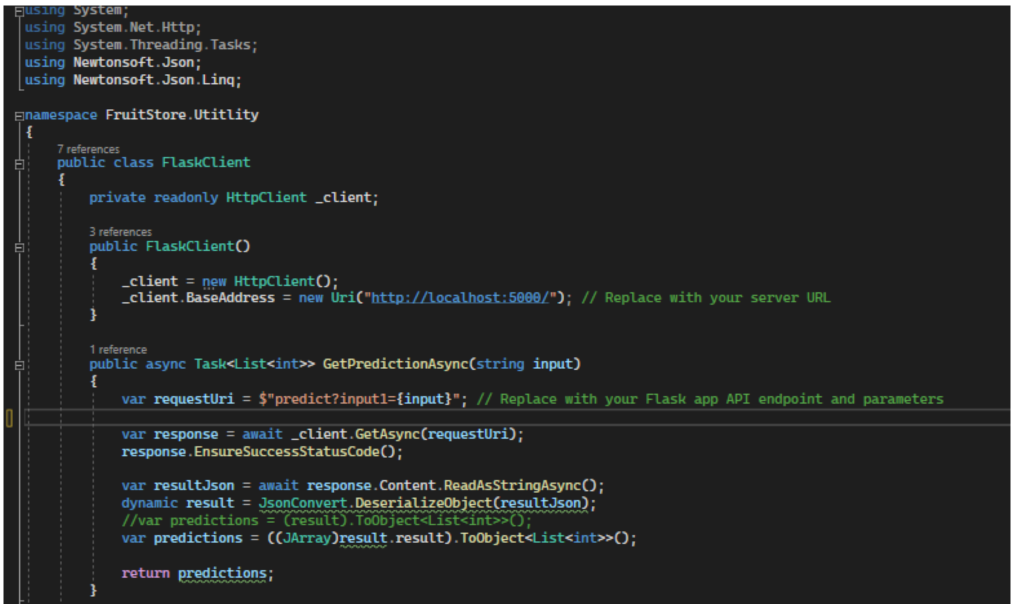
Figure 5.3.60 Flask API python code to deploy the user-based collaborative filtering recommender Part 1



```
    index2=int(index1)

    final_ratings_matrix.set_index(['user_index'], inplace=True)


    U, sigma, Vt = svds(final_ratings_matrix.to_numpy(), k = 9)

    sigma = np.diag(sigma)

    all_users_predicted_ratings = np.dot(np.dot(U, sigma), Vt)


    # Predicted ratings
    ele_preds_ratings_df = pd.DataFrame(all_users_predicted_ratings, columns = final_ratings_matrix.columns)



    num_recommendations = 6

    sorted_user_ratings = final_ratings_matrix.iloc[index2].sort_values(ascending=False)

    sorted_user_predictions = ele_preds_ratings_df.iloc[index2].sort_values(ascending=False)

    temp = pd.concat([sorted_user_ratings, sorted_user_predictions], axis=1)

    temp.columns = ['user_ratings', 'user_predictions']
    temp = temp.loc[temp.user_ratings == 0]
    temp = temp.sort_values('user_predictions', ascending=False)
    temp=temp.iloc[:6, 0]
    temp=temp.index
    temp=temp.to_numpy()
    #return {'result': np.array(temp).tolist()}
    return jsonify(result=np.array(temp).tolist())
```

Figure 5.3.61 Flask API python code to deploy the user-based collaborative filtering recommender Part 2

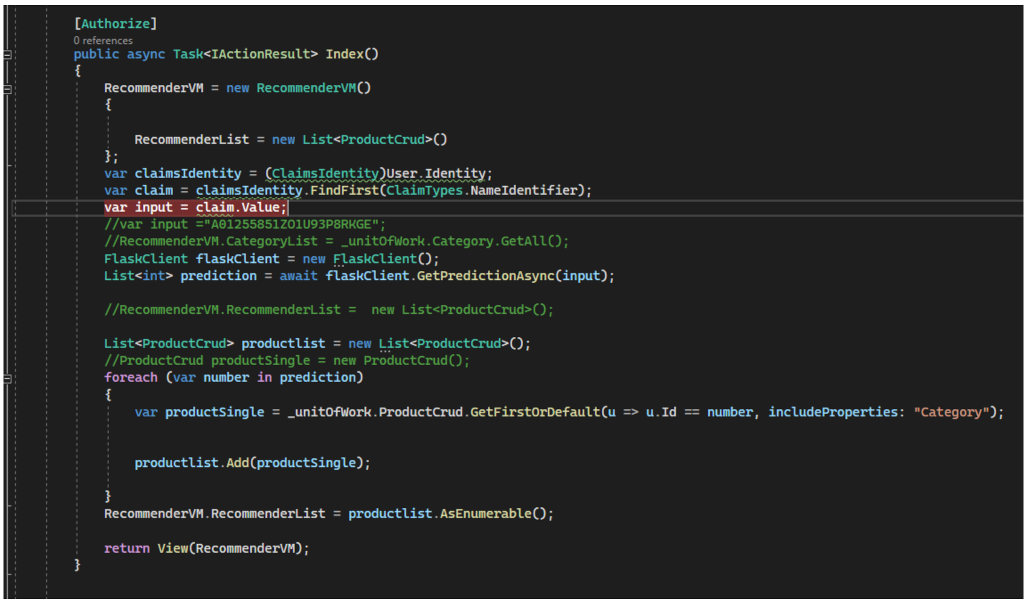
ii. This is the FlaskClient class in the web application which passes and redirects routes to the Flask API using HttpClient. The figure below show a method GetPredictionAsync(string input) which will passes the ApplicationUserId as a string parameter to the Flask app API endpoints for prediction. The Flask API result or response is passed back to the web application in JSON format. It needs further parsing and deserialization before passing back to the controller for application logic.



```
using System;
using System.Net.Http;
using System.Threading.Tasks;
using Newtonsoft.Json;
using Newtonsoft.Json.Linq;

namespace FruitStore.Utitlity
{
    7 references
    public class FlaskClient
    {
        private readonly HttpClient _client;

        3 references
        public FlaskClient()
        {
            _client = new HttpClient();
            _client.BaseAddress = new Uri("http://localhost:5000/"); // Replace with your server URL
        }

        1 reference
        public async Task<List<int>> GetPredictionAsync(string input)
        {
            var requestUri = $"predict?input1={input}"; // Replace with your Flask app API endpoint and parameters

            var response = await _client.GetAsync(requestUri);
            response.EnsureSuccessStatusCode();

            var resultJson = await response.Content.ReadAsStringAsync();
            dynamic result = JsonConvert.DeserializeObject(resultJson);
            //var predictions = (result).ToObject<List<int>>();
            var predictions = ((JArray)result.result).ToObject<List<int>>();

            return predictions;
        }
```

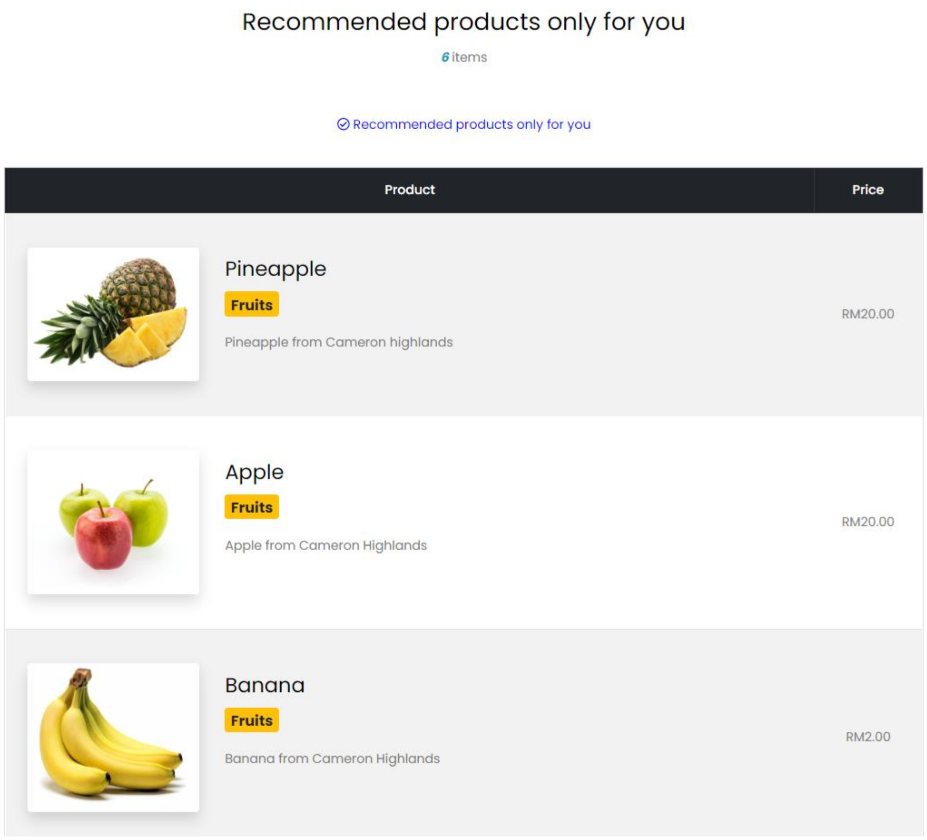
Figure 5.3.62 FlaskClient class

iii. The GetPredictionAsync(input) method is being called in this RecommenderController and the input is the ApplicationUserId for the user. The prediction being returned will be used as the productId for retrieveing its associated information from the database. All these productId with the information will be suggested to customers as the recommended products in the Index View as shown in Figure 5.3.64 below.



```
[Authorize]
0 references
public async Task<IActionResult> Index()
{
    RecommenderVM = new RecommenderVM()
    {

        RecommenderList = new List<ProductCrud>()
    };
    var claimsIdentity = (ClaimsIdentity)User.Identity;
    var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);
    var input = claim.Value;
    //var input ="A01255851Z01U93P8RKGE";
    //RecommenderVM.CategoryList = _unitOfWork.Category.GetAll();
    FlaskClient flaskClient = new FlaskClient();
    List<int> prediction = await flaskClient.GetPredictionAsync(input);

    //RecommenderVM.RecommenderList =  new List<ProductCrud>();

    List<ProductCrud> productlist = new List<ProductCrud>();
    //ProductCrud productSingle = new ProductCrud();
    foreach (var number in prediction)
    {
        var productSingle = _unitOfWork.ProductCrud.GetFirstOrDefault(u => u.Id == number, includeProperties: "Category");


        productlist.Add(productSingle);

    }
    RecommenderVM.RecommenderList = productlist.AsEnumerable();

    return View(RecommenderVM);
}
```

Figure 5.3.63 RecommenderController



Recommended products only for you

6 items

Recommended products only for you

Product	Price
Pineapple Fruits Pineapple from Cameron highlands	RM20.00
Apple Fruits Apple from Cameron Highlands	RM20.00
Banana Fruits Banana from Cameron Highlands	RM2.00

Figure 5.3.64 Index Page to show the recommended products for user-based collaborative filtering recommender

5.4 System Operation (With Screenshot)

5.4.1 Register

1. Newcomers or first-time users in this website can register via the "REGISTER" in the navigation bar being highlighted.

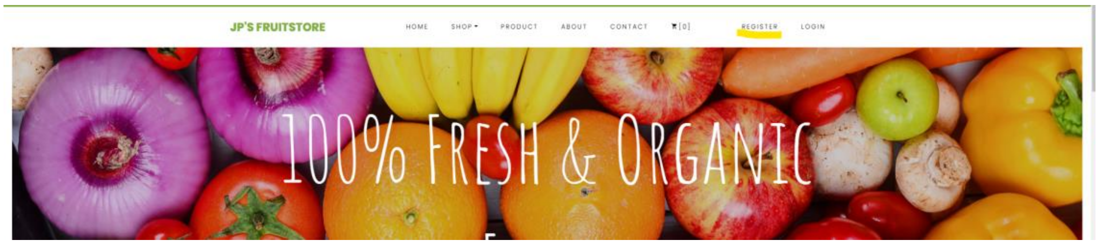
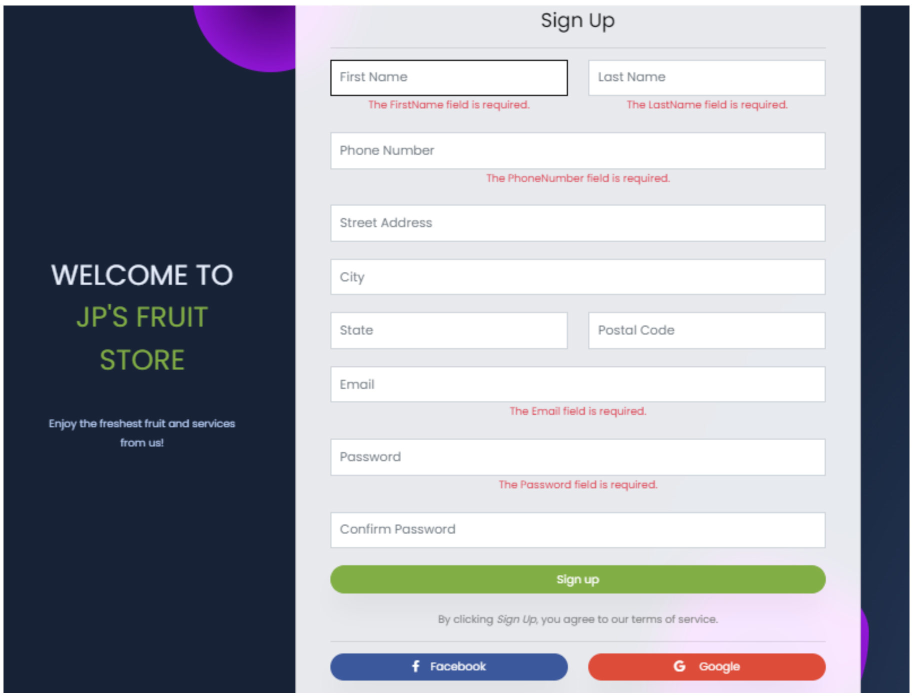


Figure 5.4.1 Home page and navigation bar

2. User will be redirected to this page and must fill up all the required details. They must make sure the password and confirmation password are matched and also all the required fields are filled up or else it will give an error validation message just like image shown in Figure 5.4.2 and 5.4.3. After making sure the password is secure and every field is filled up, user can click register to proceed. Other than that, there are other options like Facebook and Google accounts sign up. In this example, the system sign-up method will be shown only.



WELCOME TO JP'S FRUIT STORE

Enjoy the freshest fruit and services from us!

Sign Up

First Name
The FirstName field is required.

Last Name
The LastName field is required.

Phone Number
The PhoneNumber field is required.

Street Address

City

State

Postal Code

Email
The Email field is required.

Password
The Password field is required.

Confirm Password

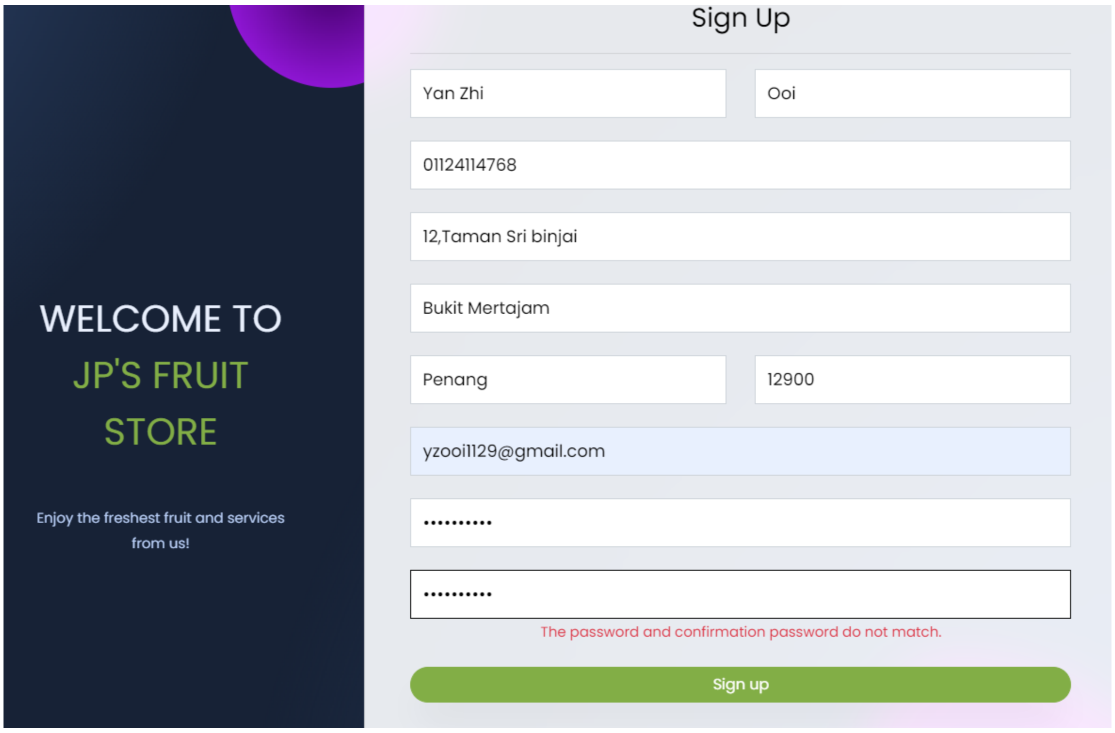
Sign up

By clicking *Sign Up*, you agree to our terms of service.

Facebook

Google

Figure 5.4.2 The required field error validation message



WELCOME TO
JP'S FRUIT
STORE

Enjoy the freshest fruit and services from us!

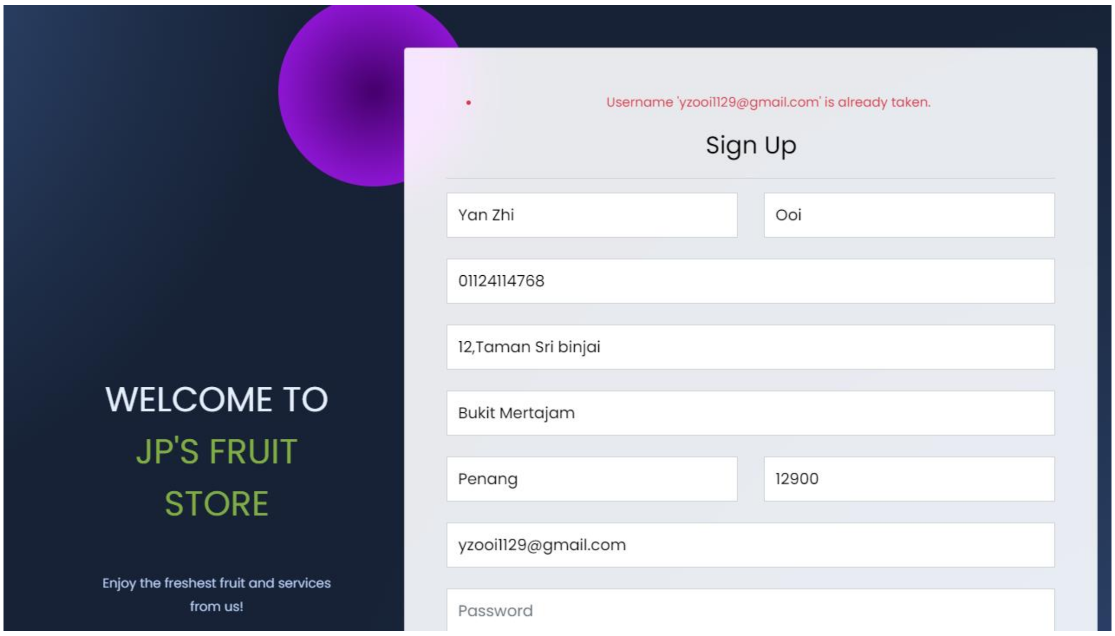
Sign Up

Yan Zhi | Ooi
01124114768
12,Taman Sri binjai
Bukit Mertajam
Penang | 12900
yzooil129@gmail.com
••••••••••
••••••••••

The password and confirmation password do not match.

Figure 5.4.3 Password and Confirm Password error validation message

3. System also will check whether the username/Gmail is already taken or not. If taken, the error validation message will show.



- Username 'yzooil129@gmail.com' is already taken.

Sign Up

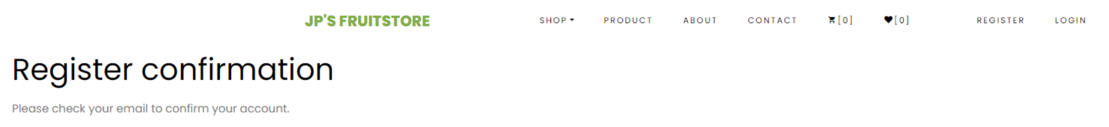
Yan Zhi | Ooi
01124114768
12,Taman Sri binjai
Bukit Mertajam
Penang | 12900
yzooil129@gmail.com
Password

WELCOME TO
JP'S FRUIT
STORE

Enjoy the freshest fruit and services from us!

Figure 5.4.4 Email being taken error validation message

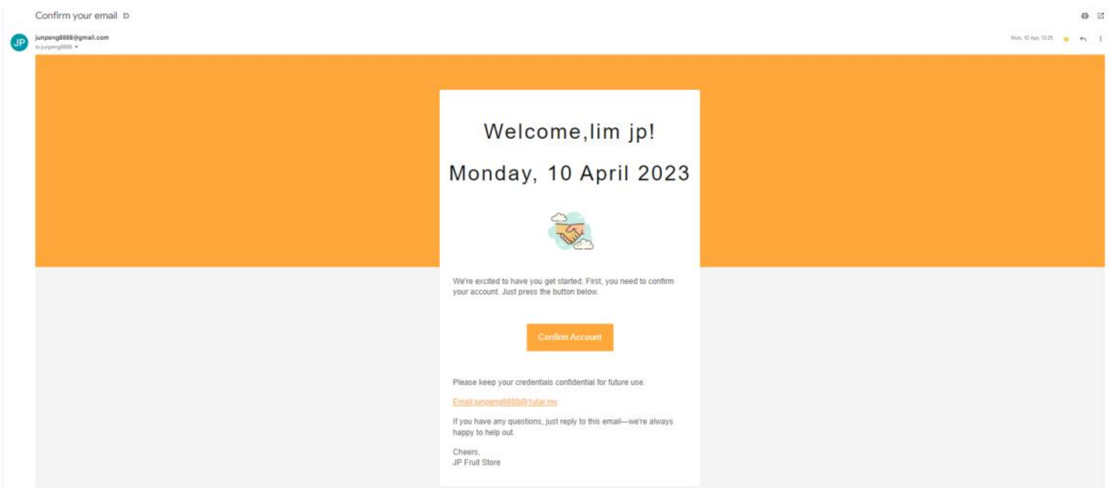
3. System will require user to verify their account in the email before successfully sign up as shown in Figure 5.4.6.



Register confirmation

Please check your email to confirm your account.

Figure 5.4.5 Register confirmation message



Confirm your email

junpeng8888@gmail.com

Welcome,lim jp!

Monday, 10 April 2023

We're excited to have you get started. First, you need to confirm your account. Just press the button below.

Please keep your credentials confidential for future use.

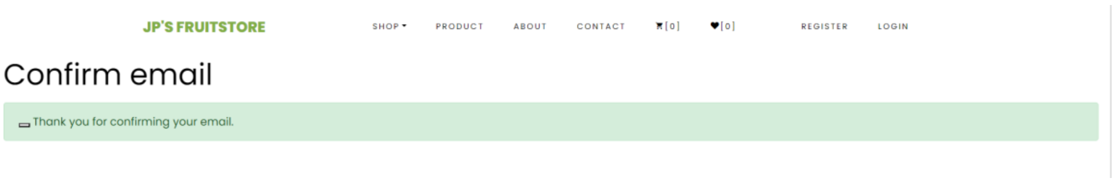
Email:junpeng8888@1utar.my

If you have any questions, just reply to this email—we're always happy to help out.

Cheers,
JP Fruit Store

Figure 5.4.6 Email verification

4. Once user confirms the email verification, the success message will show and user can login now.



Confirm email

Thank you for confirming your email.

Figure 5.4.7 Confirm email success message

5. Once user login successfully, system will verify the account and automatically redirect to the homepage with their username/Gmail as shown in the navigation bar.

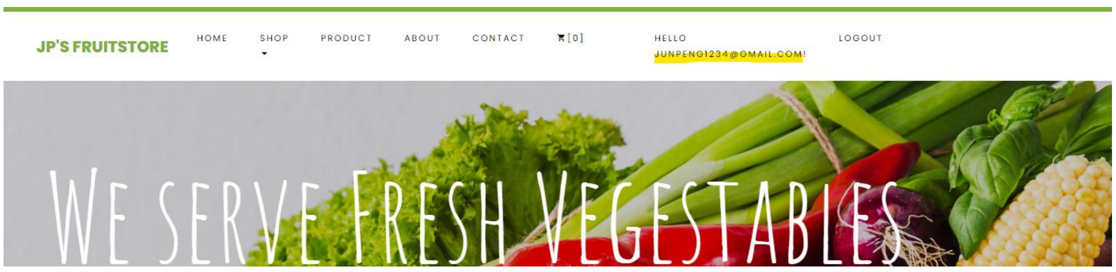
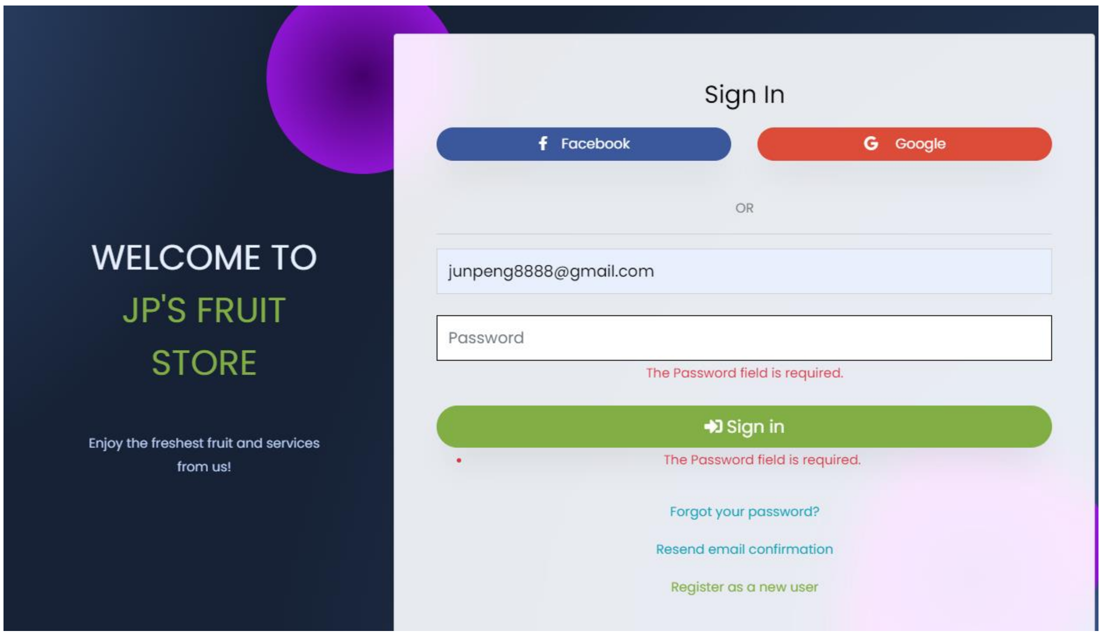


Figure 5.4.8 Home page with their username/Gmail being shown after login

5.4.2 Login

1. Returned customer or not first-time user can click the “LOGIN” link and proceed to login their account. They will be redirected to a login form and they must fill up the email and password. It is compulsory to fill up both field or else the error validation message will pop out.



junpeng8888@gmail.com

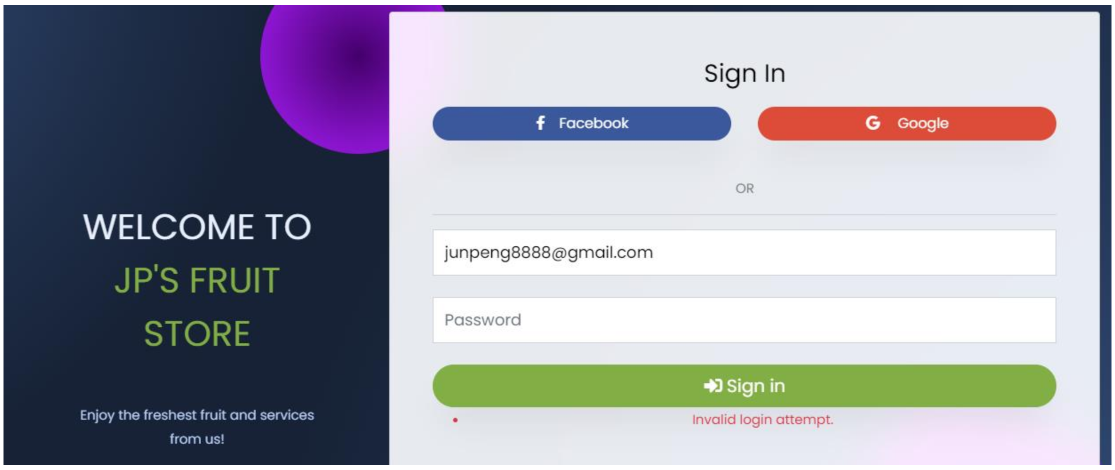
Password

The Password field is required.

• The Password field is required.

Figure 5.4.9 Password does not fill up error validation message

2. User must fill up the correct password and email. System will prompt “Invalid login attempt” error message if there is something wrong on the email and password.



junpeng8888@gmail.com

Password

• Invalid login attempt.

Figure 5.4.10 Invalid login attempt error validation message

3. After filling up the proper and accurate email and password, normal user will verified by system and successfully login their account. Their username/Gmail will be shown in the navigation bar.

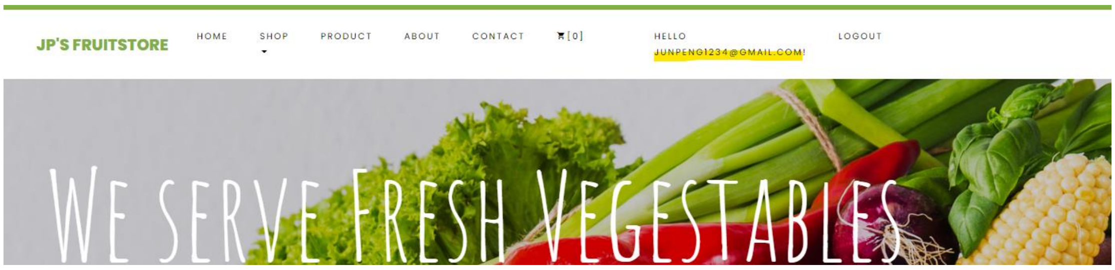
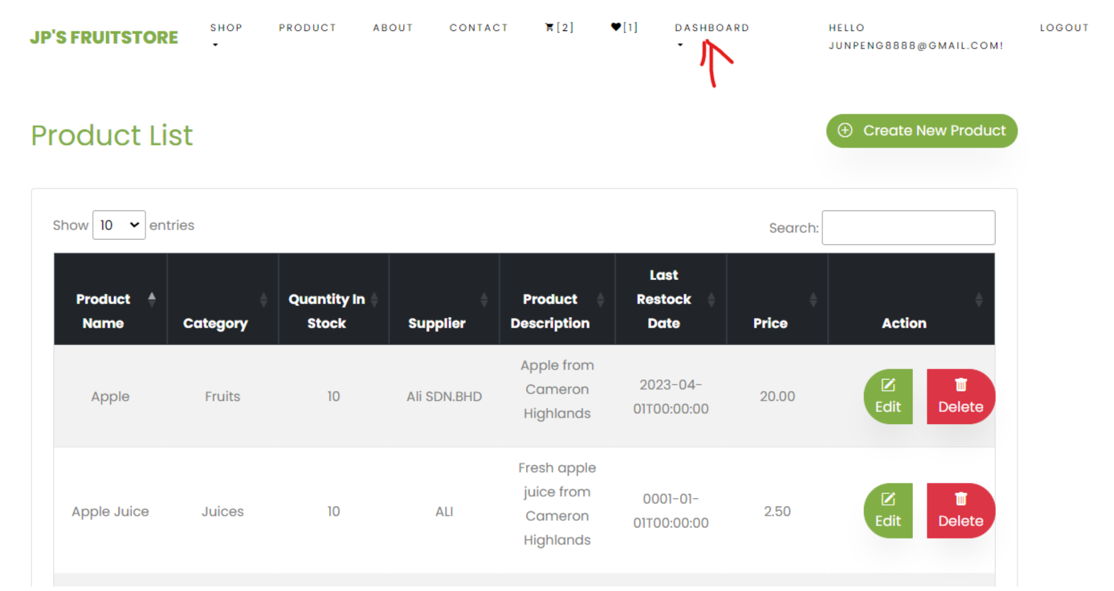


Figure 5.4.11 Home page with their username/Gmail being shown after successfully login

4. The previous steps show how normal user login and they will not able to see the dashboard navigation which only accessible by administrator. For administrator case, they will see the "Dashboard" for management purposes.



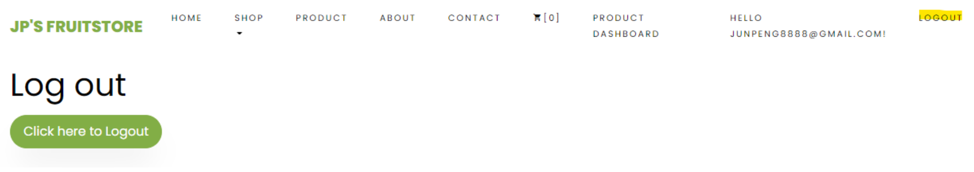
Product List

Product Name	Category	Quantity In Stock	Supplier	Product Description	Last Restock Date	Price	Action
Apple	Fruits	10	Ali SDN.BHD	Apple from Cameron Highlands	2023-04-01T00:00:00	20.00	Edit Delete
Apple Juice	Juices	10	ALI	Fresh apple juice from Cameron Highlands	0001-01-01T00:00:00	2.50	Edit Delete

Figure 5.4.12 Product Dashboard view for administrators only after successfully login

5.4.3 Logout

1. Anytime when user want to log out their account, they can just simply click the "LOGOUT" on the top right of navigation bar and will be redirected to a page with a button to confirm to log out.



Log out

Figure 5.4.13 Logout page

2. They will be redirected to home page without any account information. User can login again anytime.

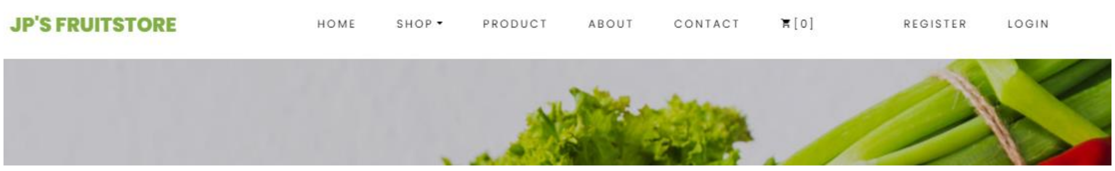
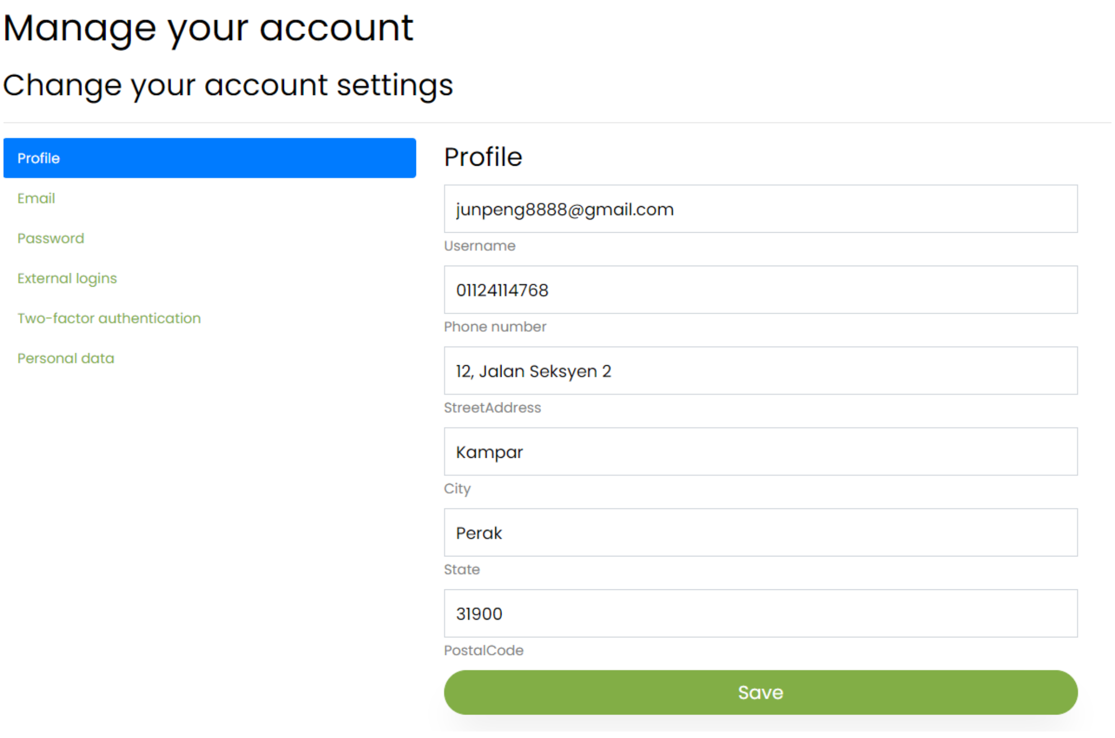


Figure 5.4.14 Home page after logout

5.4.4 Profile

1. User can click their username/Gmail and will be redirected to manage profile page. At here, user can save and update their personal information or even change password.



Manage your account

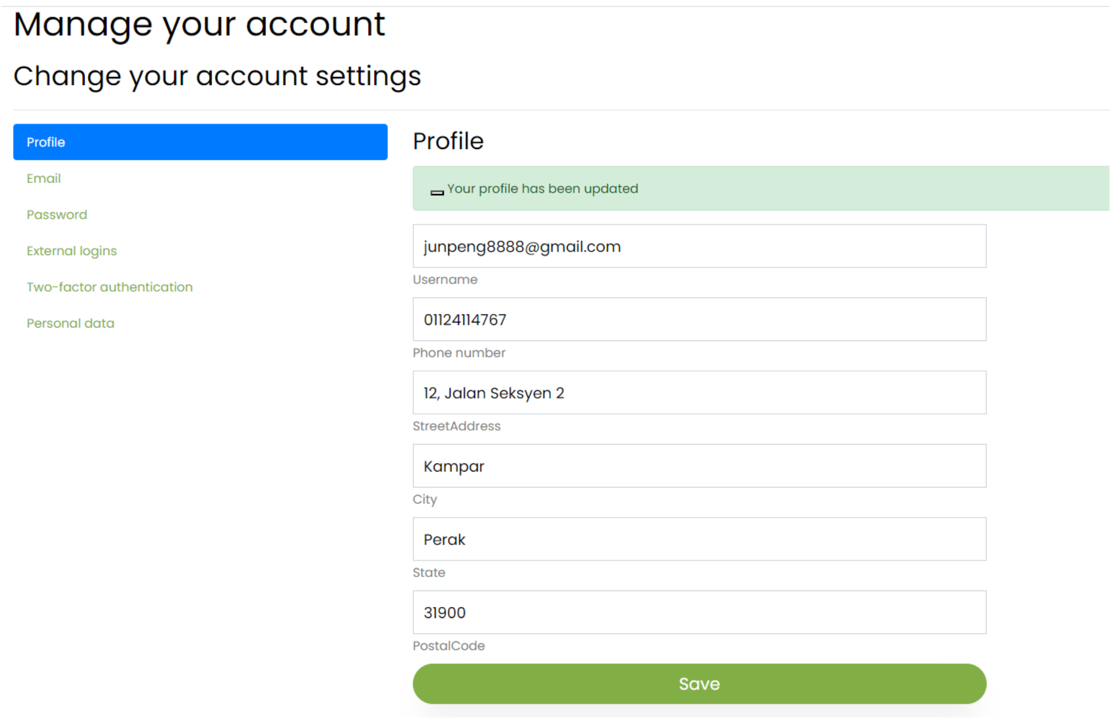
Change your account settings

Profile

Username	junpeng8888@gmail.com
Phone number	01124114768
StreetAddress	12, Jalan Seksyen 2
City	Kampar
State	Perak
PostalCode	31900

Figure 5.4.15 Profile page

2. User must fill up the required filed such as the telephone number before click save and the success message will be shown after user update their information.



Manage your account

Change your account settings

Profile

Your profile has been updated

Username: junpeng8888@gmail.com

Phone number: 01124114767

StreetAddress: 12, Jalan Seksyen 2

City: Kampar

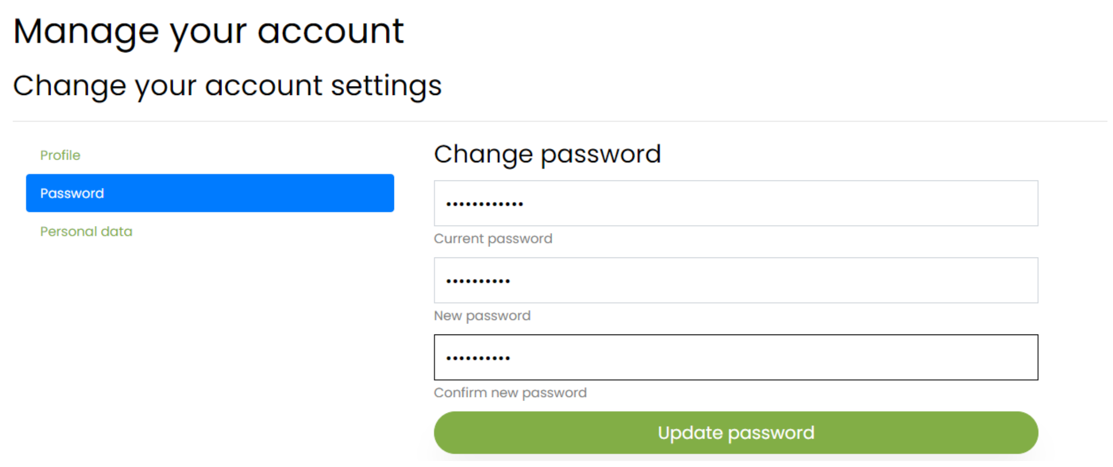
State: Perak

PostalCode: 31900

Figure 5.4.16 Success message being shown and information is updated

5.4.5 Change Password in Profile

1. User can click "Password" and they will be redirected to change password page. User has to fill up current password, new password and confirm new password field. It is compulsory to fill up the field and ensure password matching to prevent error validation message like in Figure 5.4.18 and 5.4.19.



Manage your account

Change your account settings

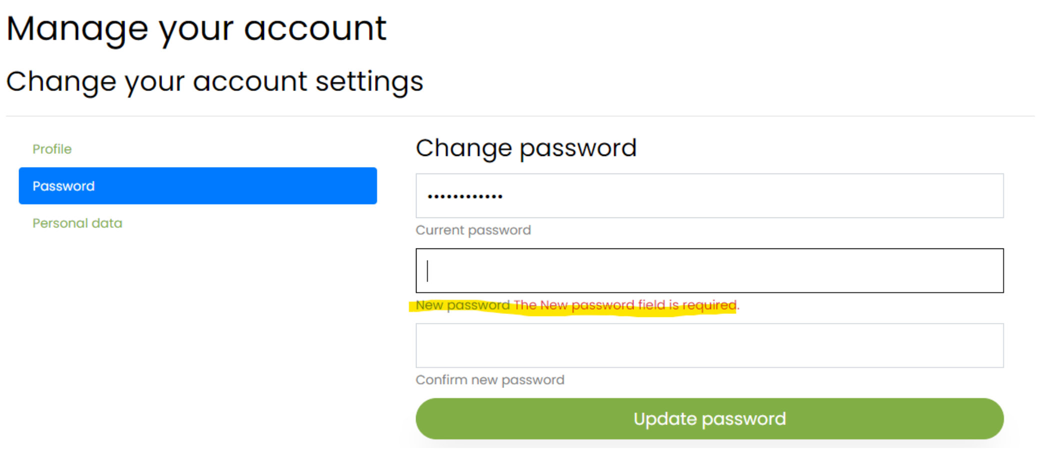
Change password

Current password: ••••••••••••

New password: ••••••••••

Confirm new password: ••••••••••

Figure 5.4.17 Change password page



Manage your account

Change your account settings

Profile
Password
Personal data

Change password

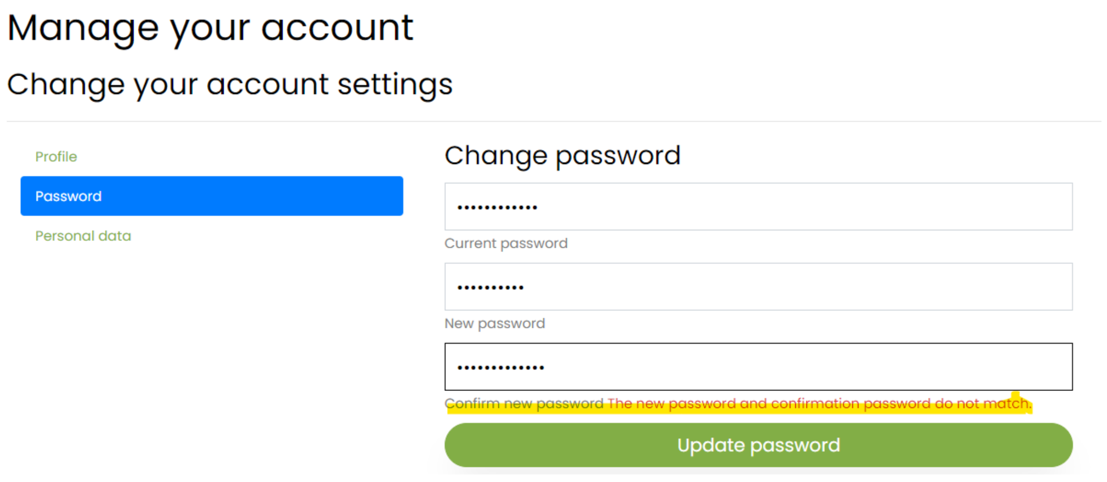
••••••••••••
Current password

New password The New password field is required.

Confirm new password

Update password

Figure 5.4.18 Password empty error validation message



Manage your account

Change your account settings

Profile
Password
Personal data

Change password

••••••••••••
Current password

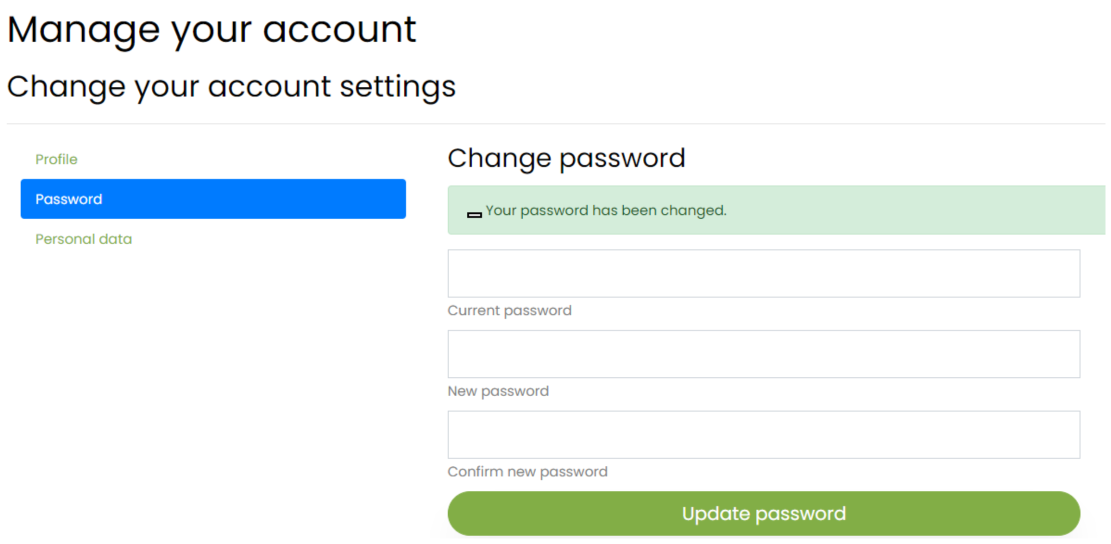
••••••••••
New password

•••••••••••••
Confirm new password The new password and confirmation password do not match.

Update password

Figure 5.4.19 New and confirm new password does not match error validation message

2. The password changed success message will be shown after user click the button.



Manage your account

Change your account settings

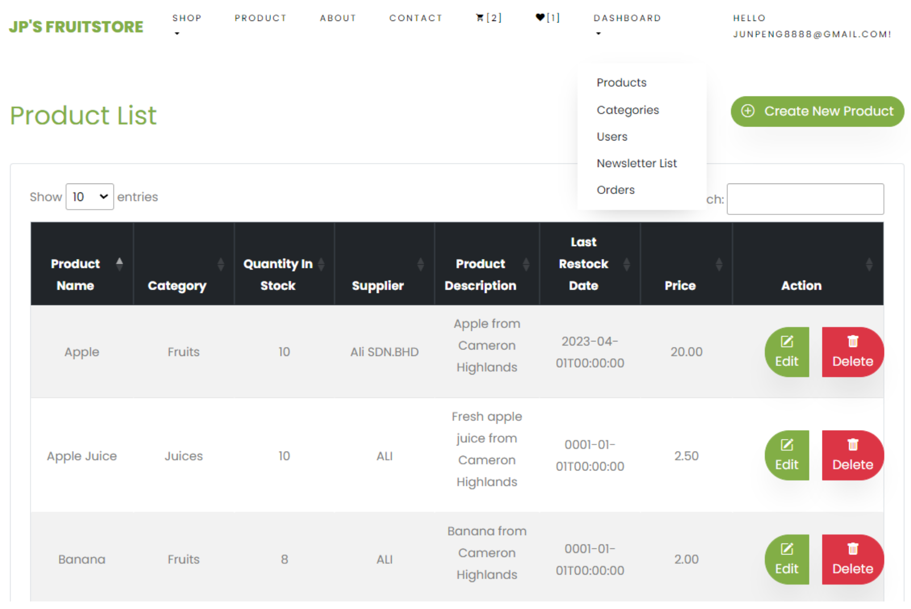
Change password

Your password has been changed.

Figure 5.4.20 Success message being shown and password is updated

5.4.6 Create new product by administrator

1. Administrator can click the "Dashboard" and they will see the "Products" option. This is where they can manage the product. In this section, we focus on the create operation. Administrators can click the "Create New Product" button and they will be redirected to the page in Figure 5.4.22.

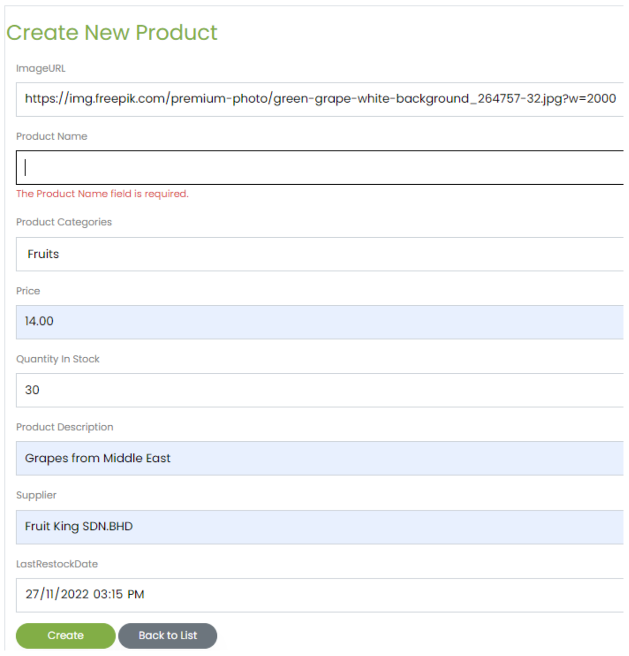


Product List

Product Name	Category	Quantity In Stock	Supplier	Product Description	Last Restock Date	Price	Action
Apple	Fruits	10	Ali SDN.BHD	Apple from Cameron Highlands	2023-04-01T00:00:00	20.00	Edit Delete
Apple Juice	Juices	10	ALI	Fresh apple juice from Cameron Highlands	0001-01-01T00:00:00	2.50	Edit Delete
Banana	Fruits	8	ALI	Banana from Cameron Highlands	0001-01-01T00:00:00	2.00	Edit Delete

Figure 5.4.21 Product Dashboard page

2. Administrator is compulsory fill up all the new product details or else the error validation will be prompted in Figure 5.4.22.



Create New Product

ImageURL: https://img.freepik.com/premium-photo/green-grape-white-background_264757-32.jpg?w=2000

Product Name:

The Product Name field is required.

Product Categories: Fruits

Price: 14.00

Quantity In Stock: 30

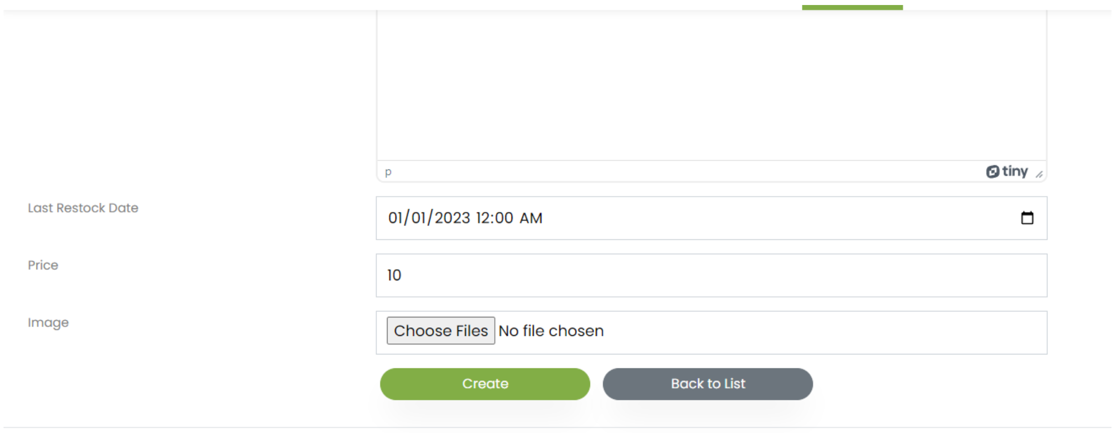
Product Description: Grapes from Middle East

Supplier: Fruit King SDN.BHD

LastRestockDate: 27/11/2022 03:15 PM

Figure 5.4.22 Field empty error validation message

3. After filling up all the required details, user can click "Create" button to create the product.



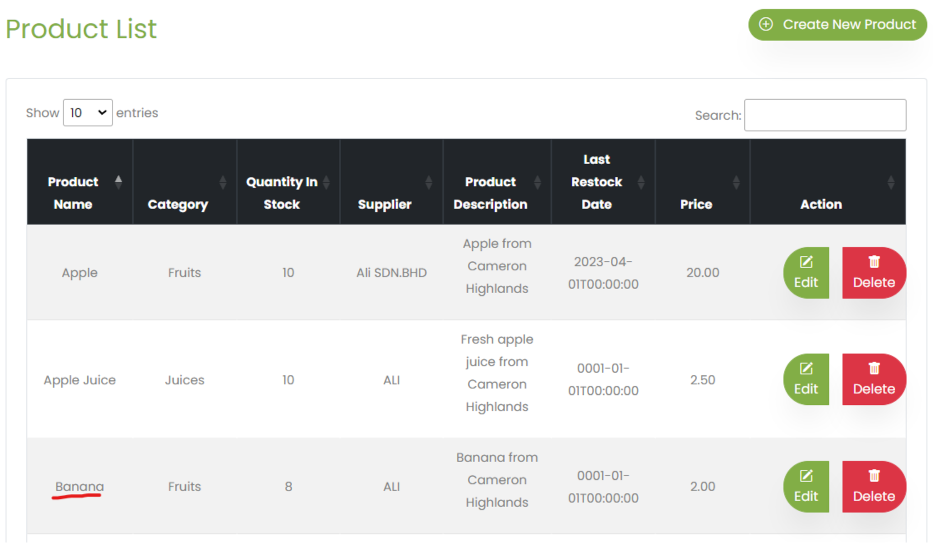
Last Restock Date: 01/01/2023 12:00 AM

Price: 10

Image: Choose Files No file chosen

Figure 5.4.23 Create new product form

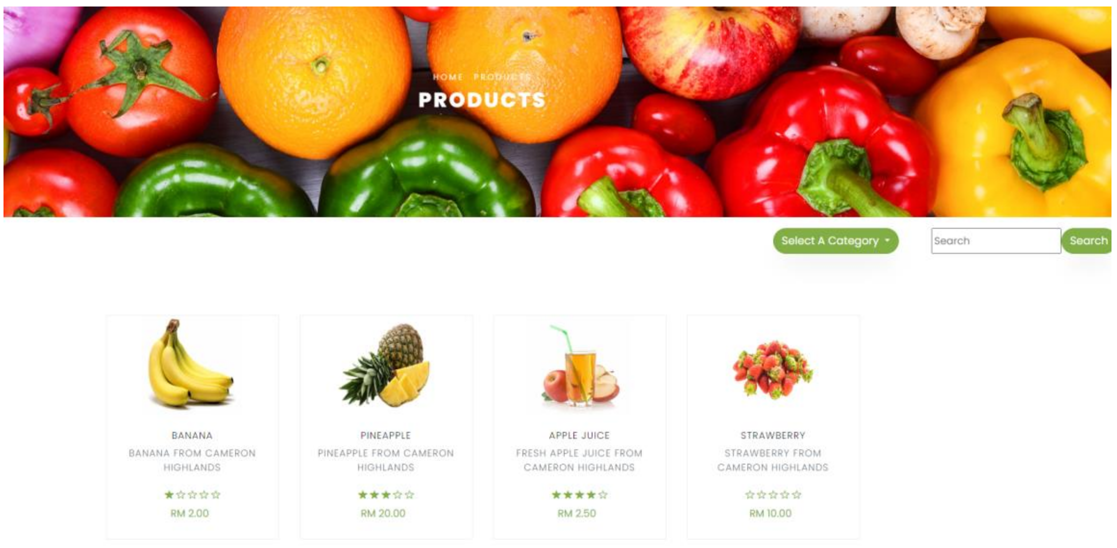
4. After the new product is created successfully, it will display in the product list under the Product Dashboard page and also being uploaded under the Product page in customer or client side.



Product List

Product Name	Category	Quantity In Stock	Supplier	Product Description	Last Restock Date	Price	Action
Apple	Fruits	10	Ali SDN.BHD	Apple from Cameron Highlands	2023-04-01T00:00:00	20.00	Edit Delete
Apple Juice	Juices	10	ALI	Fresh apple juice from Cameron Highlands	0001-01-01T00:00:00	2.50	Edit Delete
Banana	Fruits	8	ALI	Banana from Cameron Highlands	0001-01-01T00:00:00	2.00	Edit Delete

Figure 5.4.24 New product being created in the Product Dashboard page



BANANA
BANANA FROM CAMERON HIGHLANDS
★☆☆☆☆
RM 2.00

PINEAPPLE
PINEAPPLE FROM CAMERON HIGHLANDS
★★★☆☆
RM 20.00

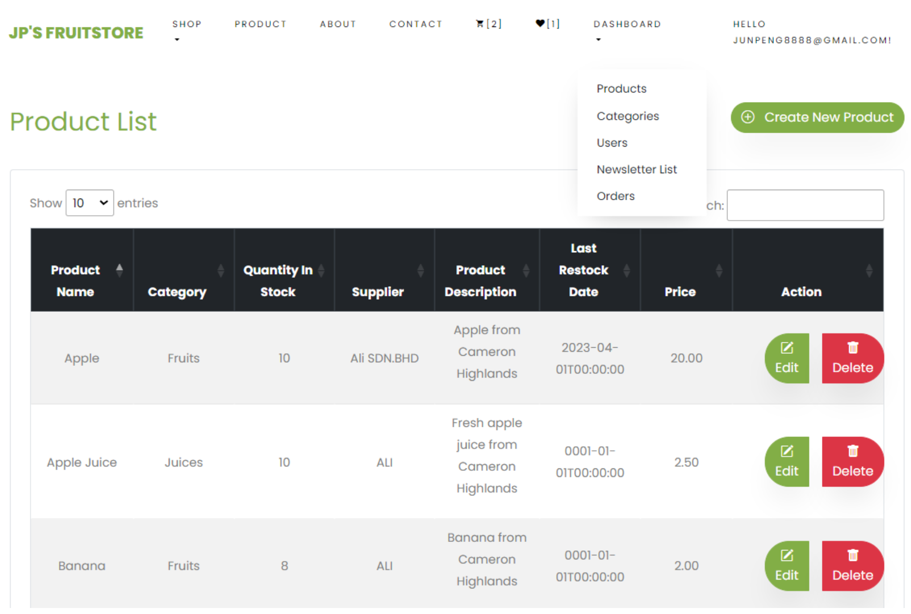
APPLE JUICE
FRESH APPLE JUICE FROM CAMERON HIGHLANDS
★★★★☆
RM 2.50

STRAWBERRY
STRAWBERRY FROM CAMERON HIGHLANDS
☆☆☆☆☆
RM 10.00

Figure 5.4.24 New product being created in the Product page

5.4.7 Update product details by administrator

1. In this section, we focus on the update operation. Administrators can click the “Edit” button and they will be redirected to the page in Figure 5.4.26.

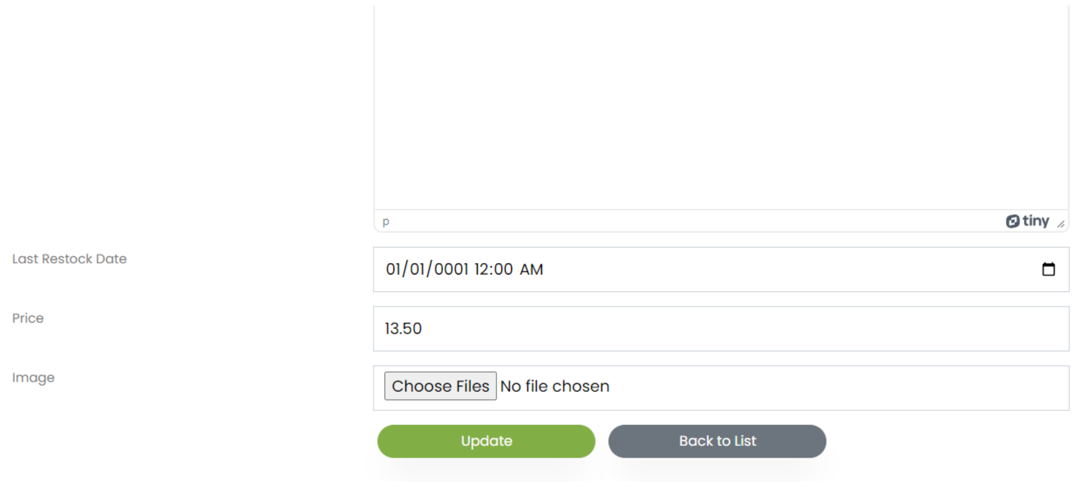


Product List

Product Name	Category	Quantity in Stock	Supplier	Product Description	Last Restock Date	Price	Action
Apple	Fruits	10	Ali SDN.BHD	Apple from Cameron Highlands	2023-04-01T00:00:00	20.00	Edit Delete
Apple Juice	Juices	10	ALI	Fresh apple juice from Cameron Highlands	0001-01-01T00:00:00	2.50	Edit Delete
Banana	Fruits	8	ALI	Banana from Cameron Highlands	0001-01-01T00:00:00	2.00	Edit Delete

Figure 5.4.25 Product Dashboard page

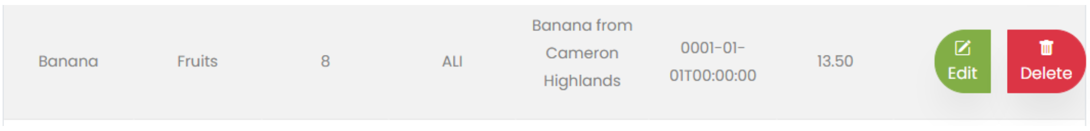
2. Administrator can edit the product details accordingly. For instance, administrator update the price from RM12.50 to RM13.50.



Last Restock Date	01/01/0001 12:00 AM
Price	13.50
Image	No file chosen

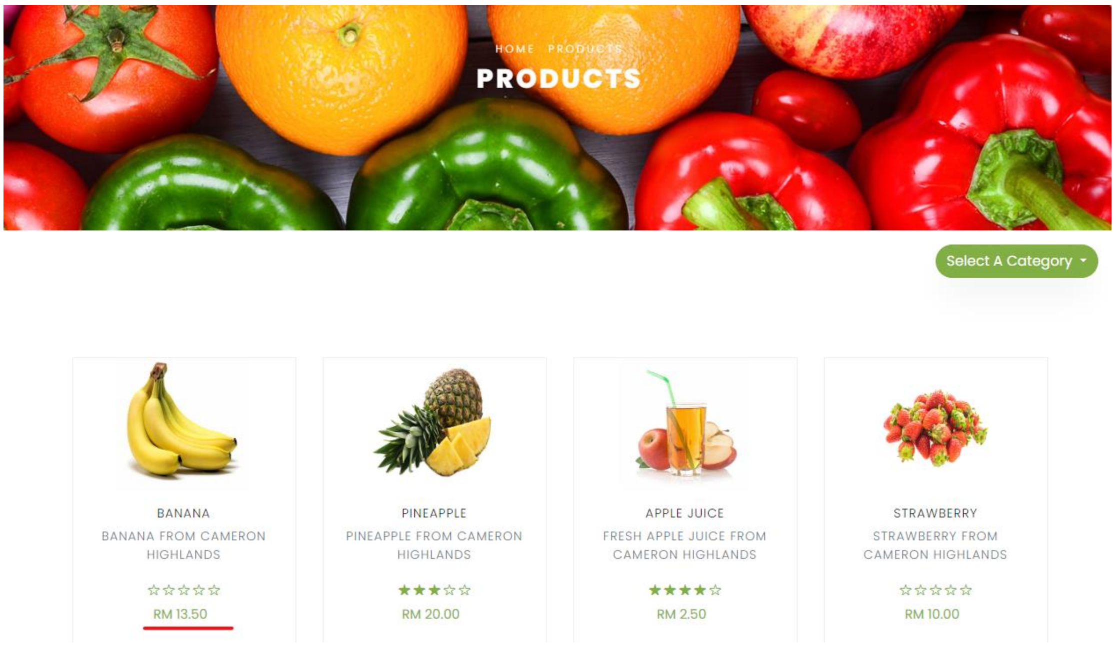
Figure 5.4.26 Edit product details form

3. The price of banana is being successfully updated in the produst list under the Product Dashboard page and also being uploaded under the Product page in user or client side.



Banana	Fruits	8	ALI	Banana from Cameron Highlands	0001-01-01T00:00:00	13.50	Edit Delete

Figure 5.4.27 Product details being updated in the Product Dashboard page



BANANA
BANANA FROM CAMERON HIGHLANDS
RM 13.50

PINEAPPLE
PINEAPPLE FROM CAMERON HIGHLANDS
RM 20.00

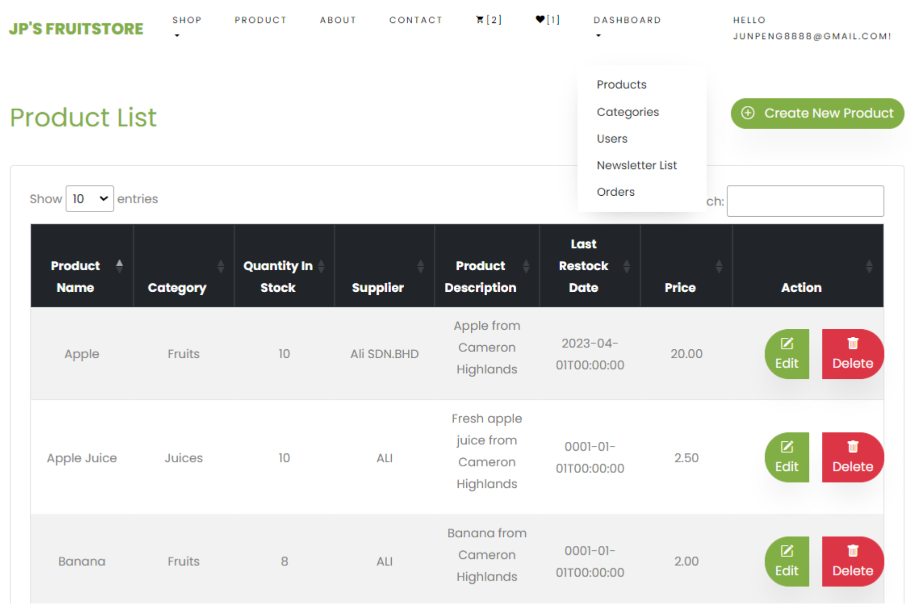
APPLE JUICE
FRESH APPLE JUICE FROM CAMERON HIGHLANDS
RM 2.50

STRAWBERRY
STRAWBERRY FROM CAMERON HIGHLANDS
RM 10.00

Figure 5.4.28 Product details being updated in the Product page

5.4.8 Delete product by administrator

1. In this section, we focus on the delete operation. Administratos can click the “Delete” button and they will be redirected to the page in Figure 5.4.31.

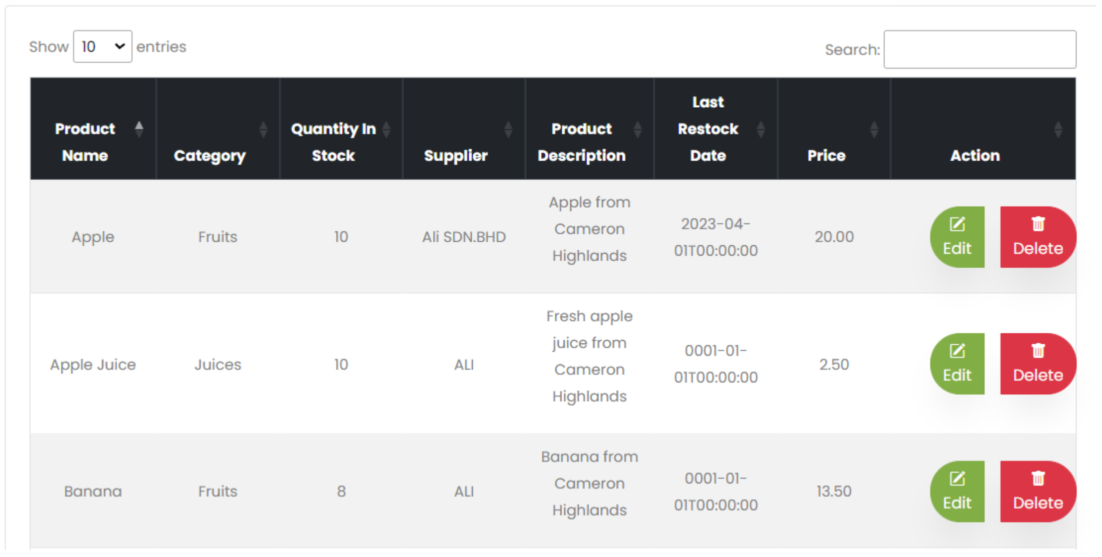


Product List

Product Name	Category	Quantity In Stock	Supplier	Product Description	Last Restock Date	Price	Action
Apple	Fruits	10	Ali SDN.BHD	Apple from Cameron Highlands	2023-04-01T00:00:00	20.00	Edit Delete
Apple Juice	Juices	10	ALI	Fresh apple juice from Cameron Highlands	0001-01-01T00:00:00	2.50	Edit Delete
Banana	Fruits	8	ALI	Banana from Cameron Highlands	0001-01-01T00:00:00	2.00	Edit Delete

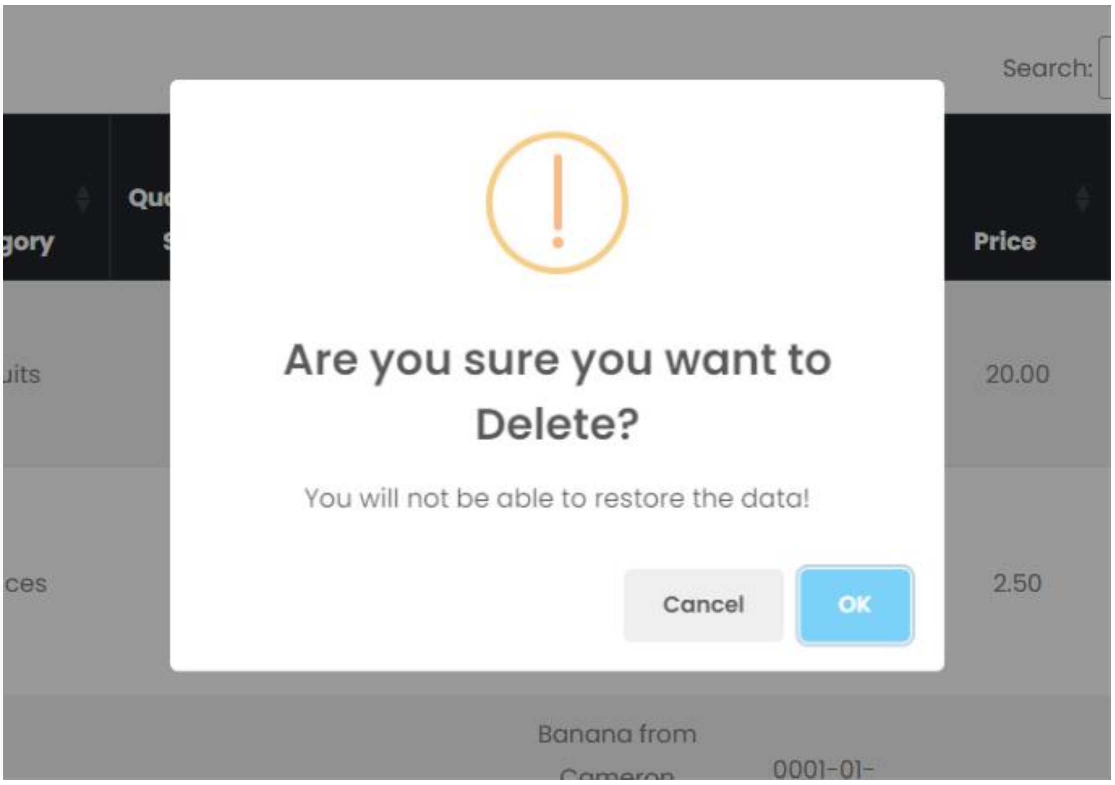
Figure 5.4.29 Product Dashboard page

2. Administrators can delete the product which may not be selling or existing in the store. For instance, administrator wants to delete the product in Figure 5.4.30. After clicking the delete button in Figure 5.4.30, the product will be deleted from the database and will not be shown in the product list anymore as well as the product page.



Product Name	Category	Quantity In Stock	Supplier	Product Description	Last Restock Date	Price	Action
Apple	Fruits	10	Ali SDN.BHD	Apple from Cameron Highlands	2023-04-01T00:00:00	20.00	Edit Delete
Apple Juice	Juices	10	ALI	Fresh apple juice from Cameron Highlands	0001-01-01T00:00:00	2.50	Edit Delete
Banana	Fruits	8	ALI	Banana from Cameron Highlands	0001-01-01T00:00:00	13.50	Edit Delete

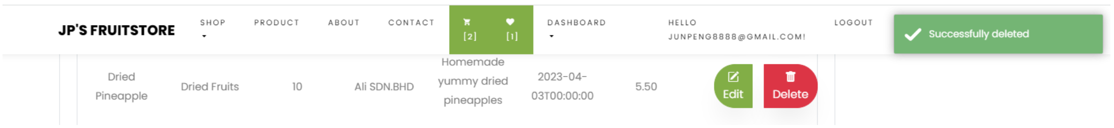
Figure 5.4.30 Testing product for delete operation



Are you sure you want to Delete?

You will not be able to restore the data!

Figure 5.4.31 Delete product form



Successfully deleted

Dried Pineapple	Dried Fruits	10	Ali SDN.BHD	Homemade yummy dried pineapples	2023-04-03T00:00:00	5.50

Figure 5.4.32 Product deleted successfully message and product is no longer exist in the product list

5.4.9 View single product

1. User or customer can view all the products which had been created by administrators on the Product page. All the products available here are dynamically fetched from the database which means the product is not hardcoded one by one. Customer can click the button as shown in Figure 5.4.34 to view the single product page.

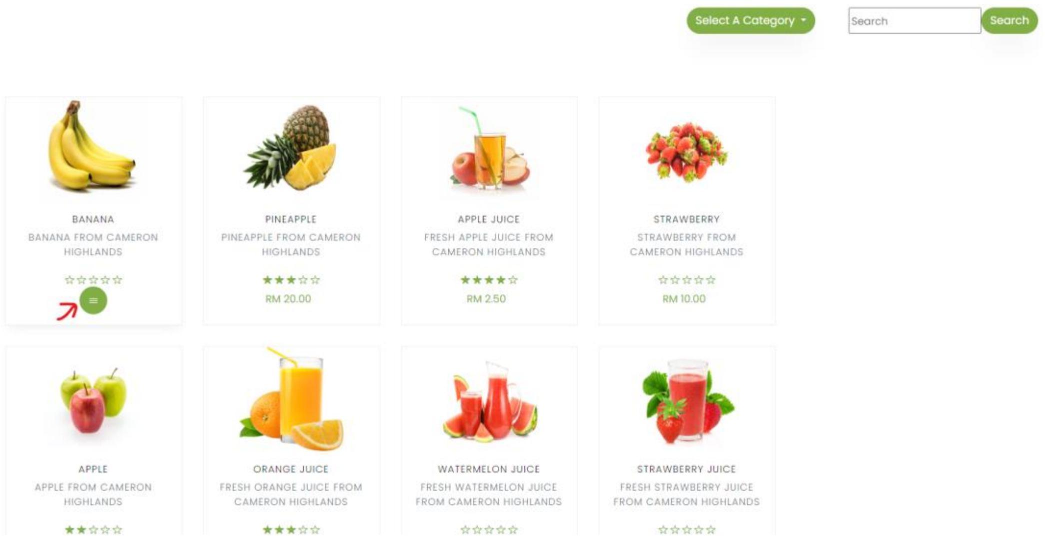

Select A Category
Search
Search
BANANA
BANANA FROM CAMERON HIGHLANDS
PINEAPPLE
PINEAPPLE FROM CAMERON HIGHLANDS
RM 20.00
APPLE JUICE
FRESH APPLE JUICE FROM CAMERON HIGHLANDS
RM 2.50
STRAWBERRY
STRAWBERRY FROM CAMERON HIGHLANDS
RM 10.00
APPLE
APPLE FROM CAMERON HIGHLANDS
ORANGE JUICE
FRESH ORANGE JUICE FROM CAMERON HIGHLANDS
WATERMELON JUICE
FRESH WATERMELON JUICE FROM CAMERON HIGHLANDS
STRAWBERRY JUICE
FRESH STRAWBERRY JUICE FROM CAMERON HIGHLANDS

Figure 5.4.33 Product page which shows all the products in database

2. After users click the button, they will be redirected to single product page which associated with its product details such as the product name, price, description and the image. All these details are updatable and deletable. The administrators can edit the details in the Product Dashboard page and the changes will automatically applied here.

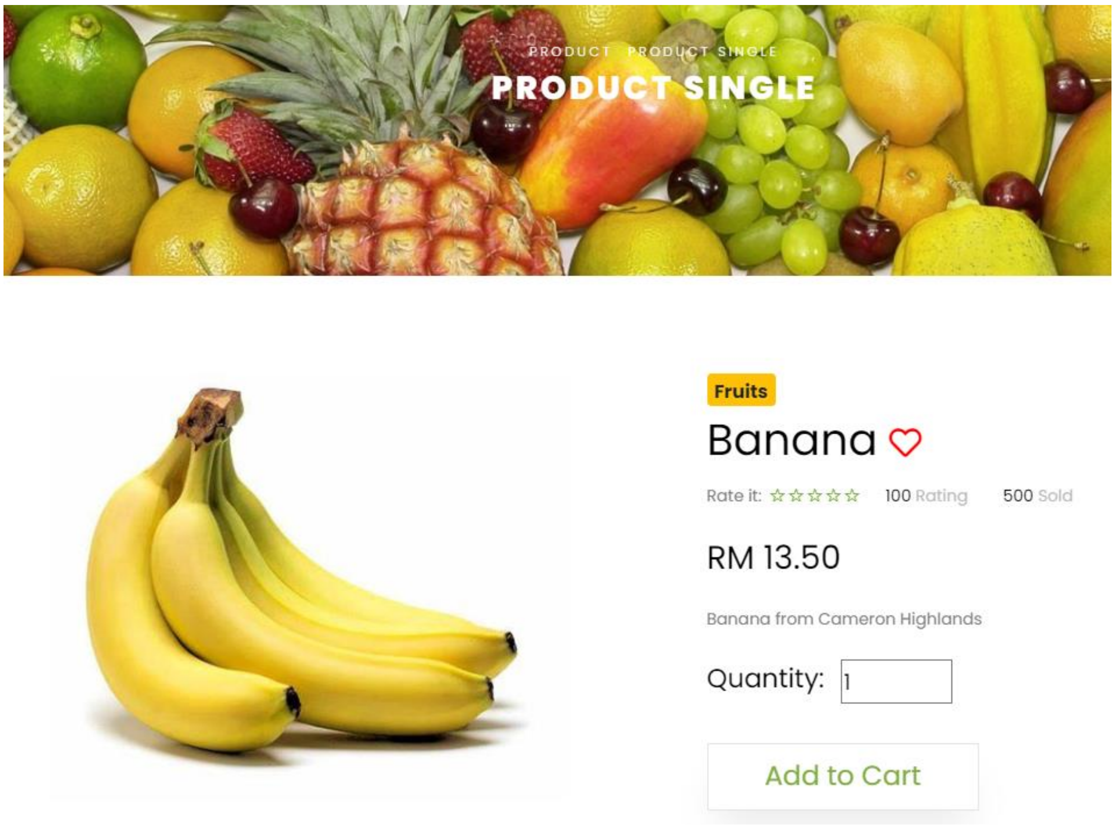


Figure 5.4.34 Single product page which shows its detail

5.4.10 Add to cart

1. After users click the “Add to Cart” button with the appropriate quantity, the single product information will stored inside the cart as shown in Figure 5.4.36.

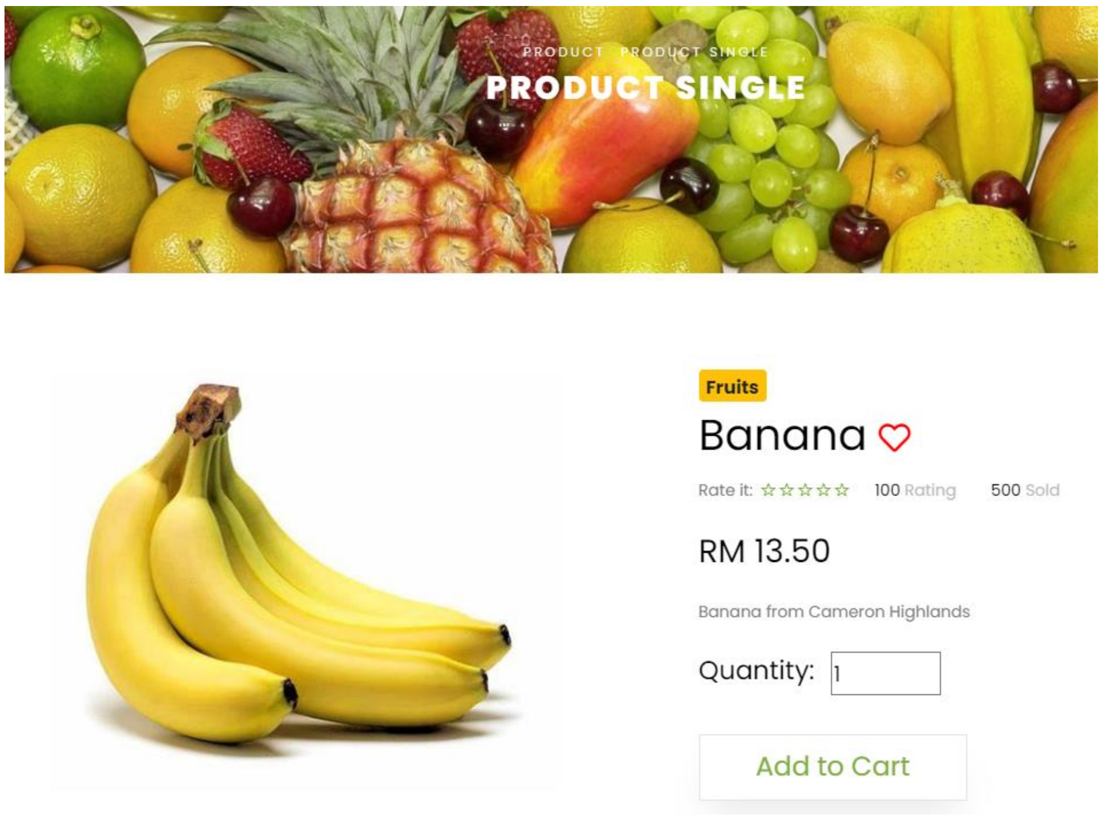
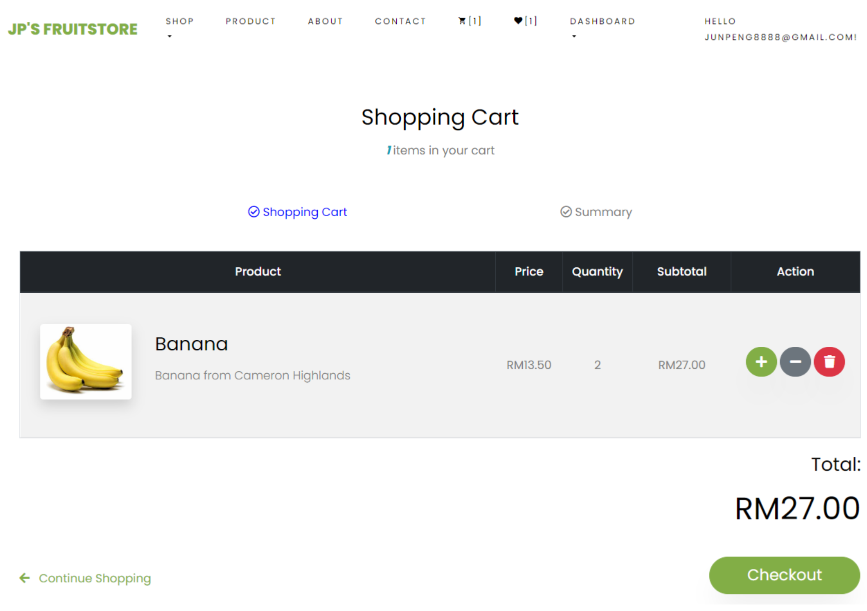


Figure 5.4.35 Single product page which shows its detail

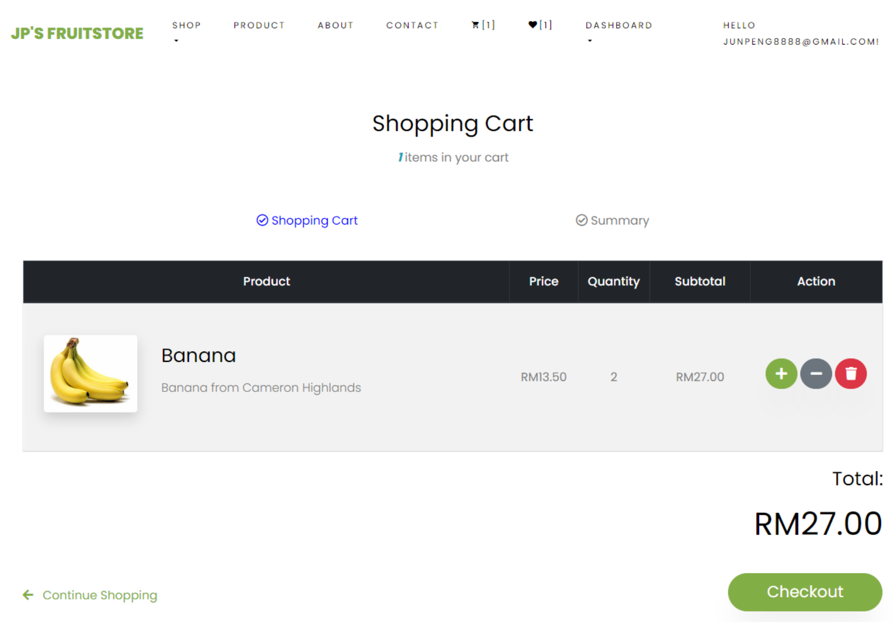


Product	Price	Quantity	Subtotal	Action
Banana Banana from Cameron Highlands	RM13.50	2	RM27.00	+ − 🗑

Figure 5.4.36 The product will stored into the shopping cart after customer clicks the button

5.4.11 Manage cart

1. In the cart, there are 3 actions in which customers can either add more quantity, reduce the quantity or directly remove the product from the cart.



Shopping Cart

1 items in your cart

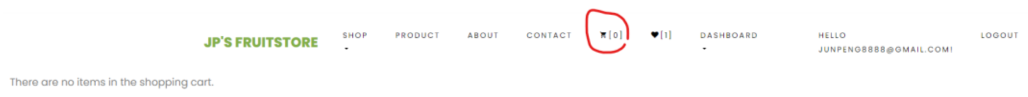
Product	Price	Quantity	Subtotal	Action
Banana Banana from Cameron Highlands	RM13.50	2	RM27.00	+ − 🗑

Total:

RM27.00

Figure 5.4.37 The shopping cart

2. if customer clicks the remove button, the product will be totally removed from the shopping cart and display “There are no items in the shopping cart”. The number for the cart has changed from 1 to 0.

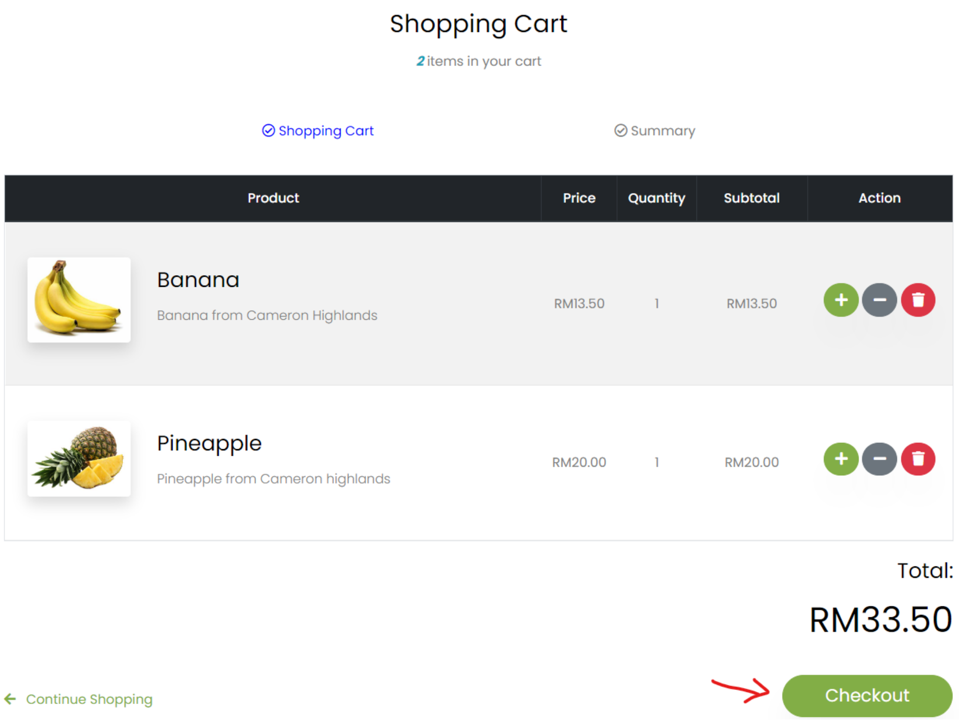


There are no items in the shopping cart.

Figure 5.4.38 Display no items if customer delete it from the cart

5.4.12 Check out and place order

1. When a customer clicks the checkout button, it will redirect customer to the order summary page in Figure and requires customer to fill up all the required personal and shipping information on the left-hand side. Besides, there will be showing the product subtotal price and total to pay amount on the right-hand side.



Shopping Cart

2 items in your cart

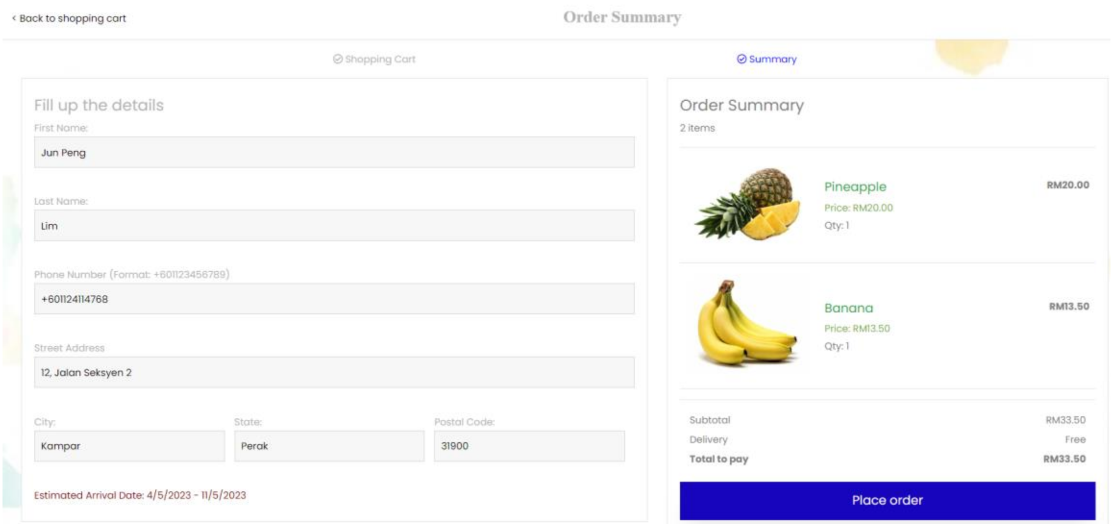
Shopping Cart | Summary

Product		Price	Quantity	Subtotal	Action
	Banana Banana from Cameron Highlands	RM13.50	1	RM13.50	
	Pineapple Pineapple from Cameron highlands	RM20.00	1	RM20.00	

Total:

RM33.50

Figure 5.4.39 Shopping cart with checkout button



‹ Back to shopping cart

Order Summary

Shopping Cart | Summary

Fill up the details

First Name: Jun Peng		
Last Name: Lim		
Phone Number (Format: +601123456789): +60124114768		
Street Address: 12, Jalan Seksyen 2		
City: Kampar	State: Perak	Postal Code: 31900

Estimated Arrival Date: 4/5/2023 - 11/5/2023

Order Summary

2 items

Pineapple Price: RM20.00 Qty: 1	RM20.00
Banana Price: RM13.50 Qty: 1	RM13.50
Subtotal	RM33.50
Delivery	Free
Total to pay	RM33.50

Figure 5.4.50 Order Summary page

2. If customer fills up all the information and clicks “Place order” button, the following page will be displayed and request user to submit the card and personal information in order to place order and make payment successfully.

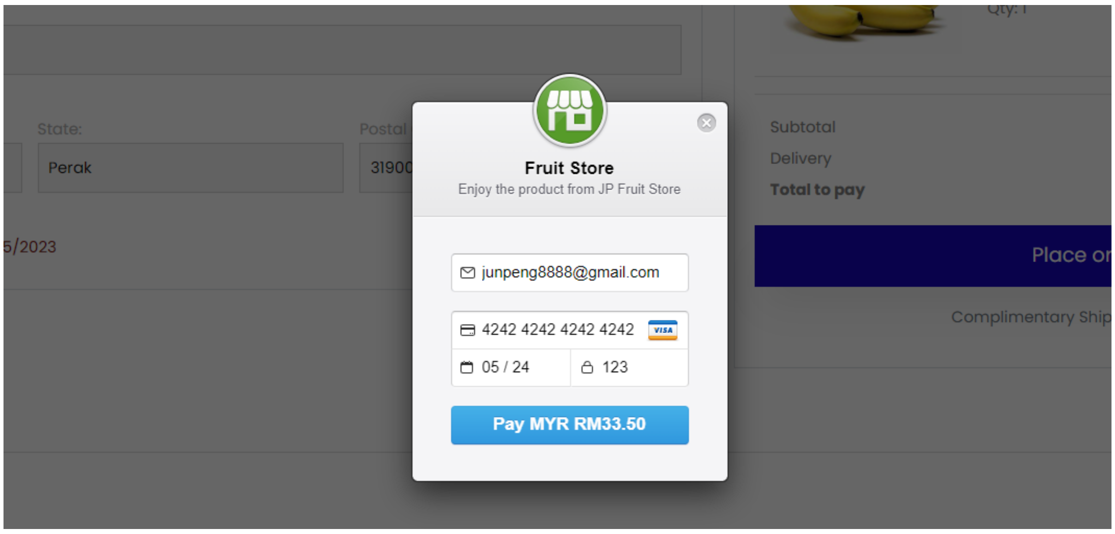
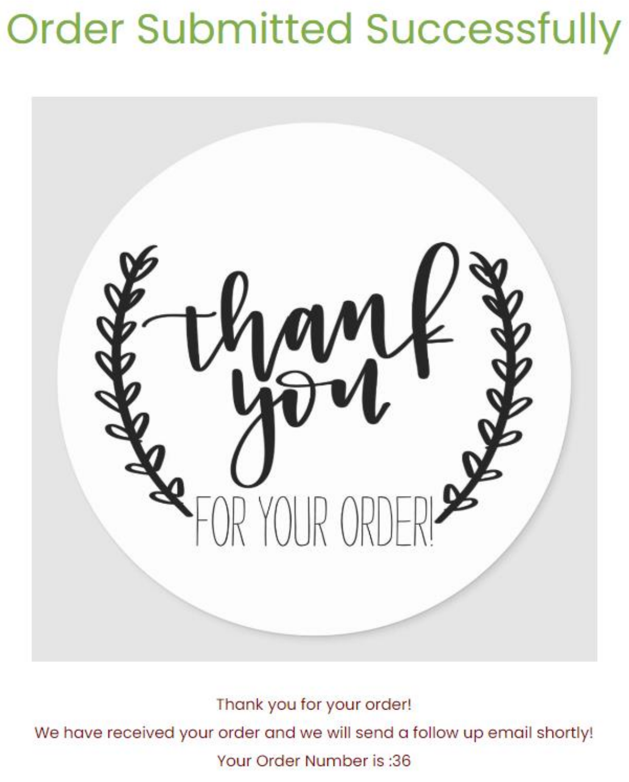


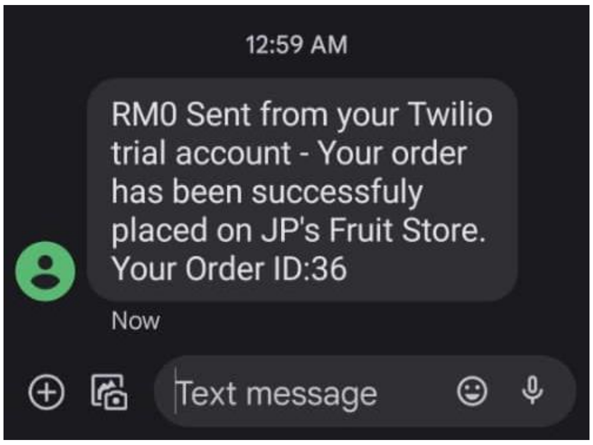
Figure 5.4.41 Payment page

3. If the customer has successfully made the payment, an order success page will be displayed and a Twilio success message will be sent to the customer's phone. The new order record with order number 36 has been added into database and can be displayed in the order history as shown in Figure 5.4.44 and payment record can be viewed in Stripe dashboard.


Order Submitted Successfully

Thank you for your order!
We have received your order and we will send a follow up email shortly!
Your Order Number is :36

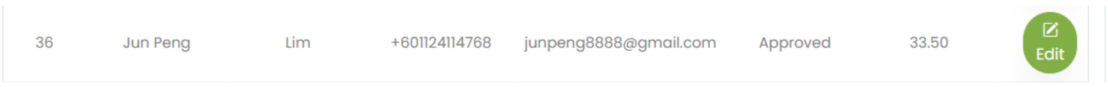
Figure 5.4.42 Order success page


12:59 AM

RM0 Sent from your Twilio trial account - Your order has been successfuly placed on JP's Fruit Store. Your Order ID:36

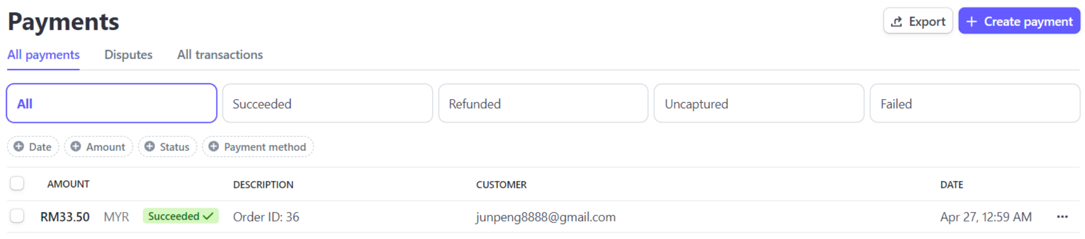
Now

Figure 5.4.43 Twilio order success message



36	Jun Peng	Lim	+601124114768	junpeng8888@gmail.com	Approved	33.50	Edit

Figure 5.4.44 Order record with order number 36

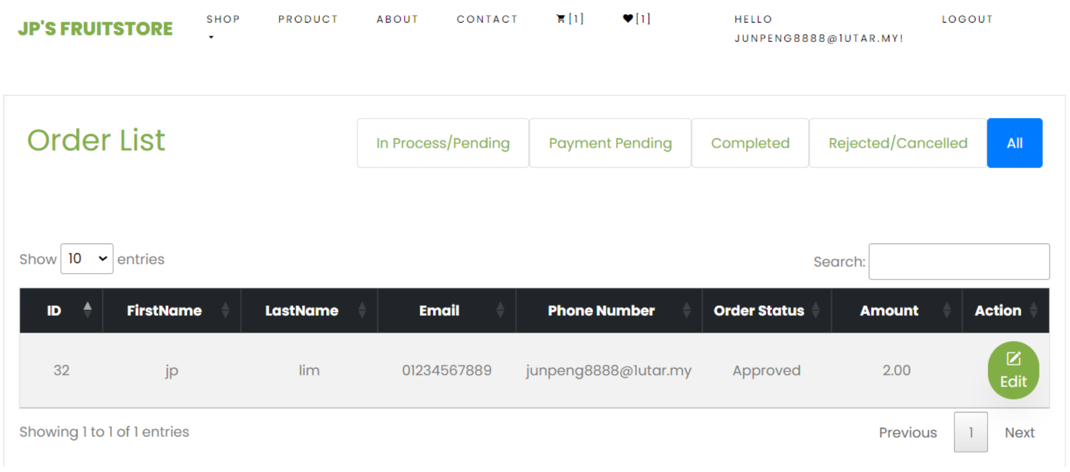

Payments

AMOUNT			DESCRIPTION	CUSTOMER	DATE
RM33.50	MYR	Succeeded ✓	Order ID: 36	junpeng8888@gmail.com	Apr 27, 12:59 AM

Figure 5.4.45 Payment record with order number 36

5.4.13 Customer view order and cancel order operation for special scenario

1. Customers can view their order history on this page.



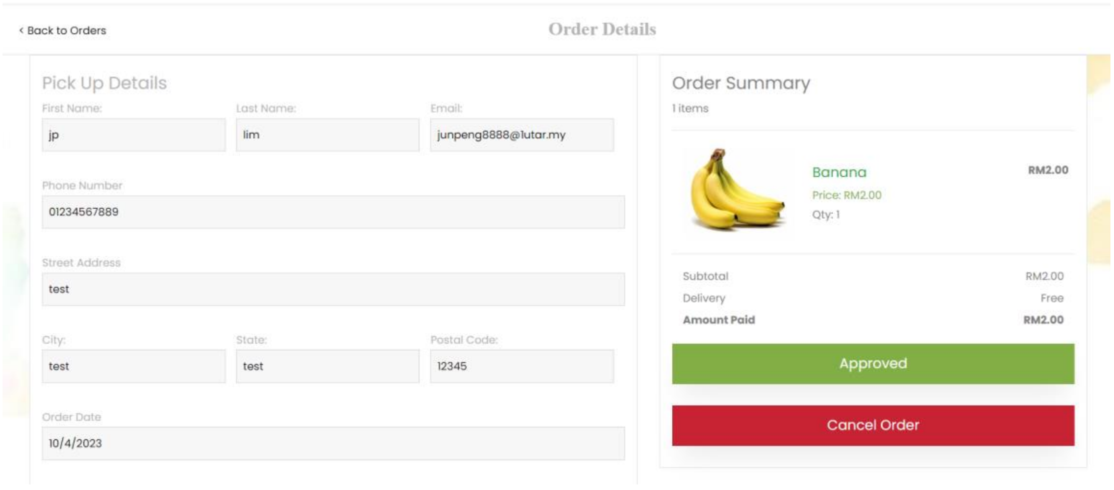
Order List

ID	FirstName	LastName	Email	Phone Number	Order Status	Amount	Action
32	jp	lim	01234567889	junpeng8888@lutar.my	Approved	2.00	Edit

Showing 1 to 1 of 1 entries

Figure 5.4.46 Order history table

2. Customers will be able to check the status of the order and there is a cancel order button below. Noted that they only can cancel order before the items are shipped.


‹ Back to Orders

Order Details

Pick Up Details

First Name: jp	Last Name: lim	Email: junpeng8888@lutar.my
Phone Number: 01234567889		
Street Address: test		
City: test	State: test	Postal Code: 12345
Order Date: 10/4/2023		

Order Summary

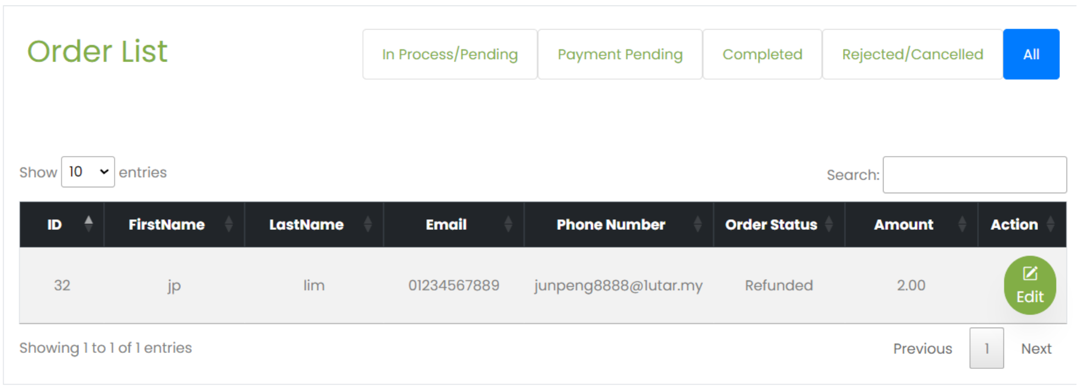
1 items

Banana RM2.00
Price: RM2.00
Qty: 1

Subtotal	RM2.00
Delivery	Free
Amount Paid	RM2.00

Figure 5.4.47 Order details

3. If a customer clicks the cancel order button, the order will be cancelled, and the money will be refunded soon back to the customer's card account. The proof of payment refund record is shown below.

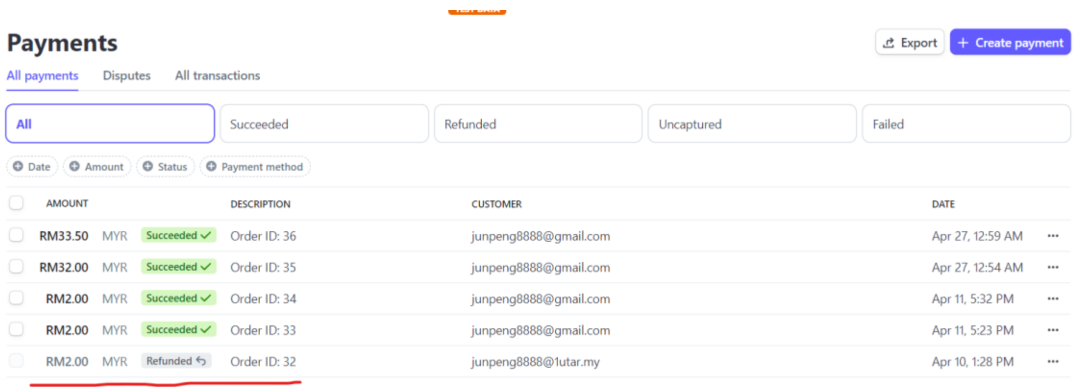

Order List

Show 10 entries Search:

ID	FirstName	LastName	Email	Phone Number	Order Status	Amount	Action
32	jp	lim	01234567889	junpeng8888@lutar.my	Refunded	2.00	Edit

Showing 1 to 1 of 1 entries

Figure 5.4.48 Order history table with payment refund record

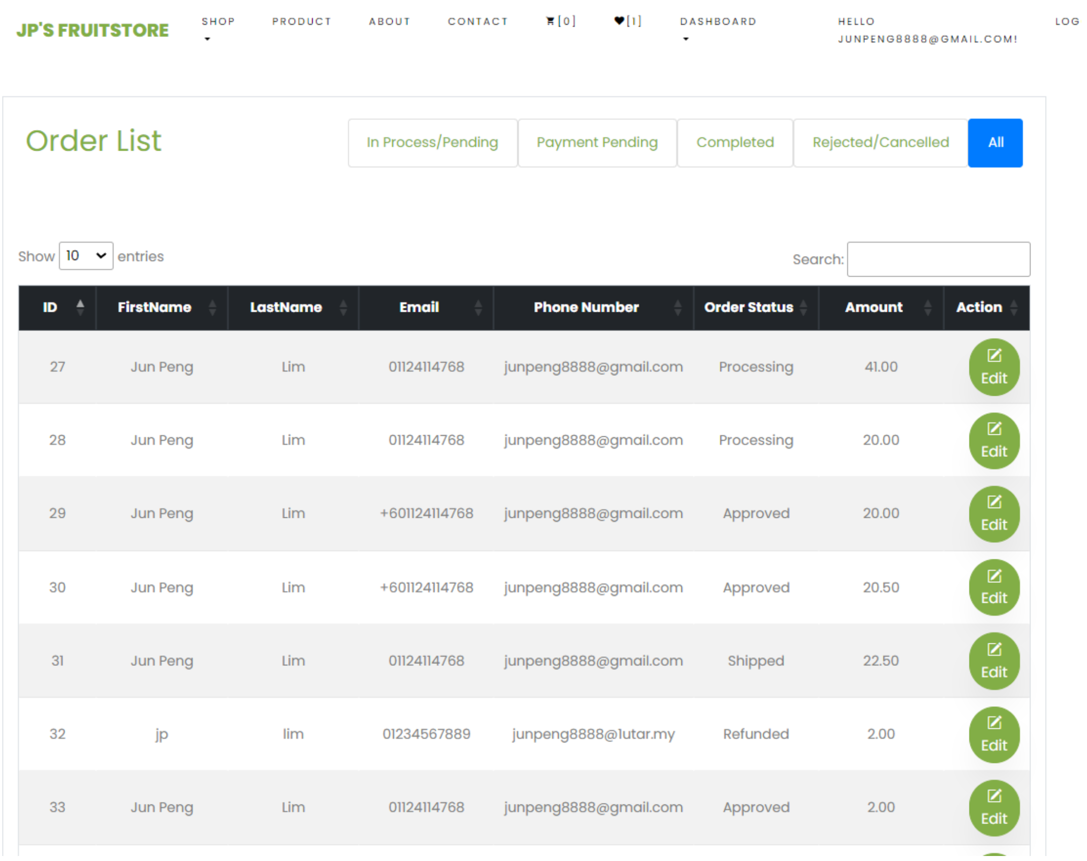

Payments

AMOUNT			DESCRIPTION	CUSTOMER	DATE	
RM33.50	MYR	Succeeded ✓	Order ID: 36	junpeng8888@gmail.com	Apr 27, 12:59 AM	···
RM32.00	MYR	Succeeded ✓	Order ID: 35	junpeng8888@gmail.com	Apr 27, 12:54 AM	···
RM2.00	MYR	Succeeded ✓	Order ID: 34	junpeng8888@gmail.com	Apr 11, 5:32 PM	···
RM2.00	MYR	Succeeded ✓	Order ID: 33	junpeng8888@gmail.com	Apr 11, 5:23 PM	···
RM2.00	MYR	Refunded	Order ID: 32	junpeng8888@1utar.my	Apr 10, 1:28 PM	···

Figure 5.4.49 Payment refund record in Stripe dashboard

5.4.14 Order module managed by administrator

1. The order list with all the orders placed by customers will only be accessible by the administrator. Administrator can click edit button to view and update the order status.

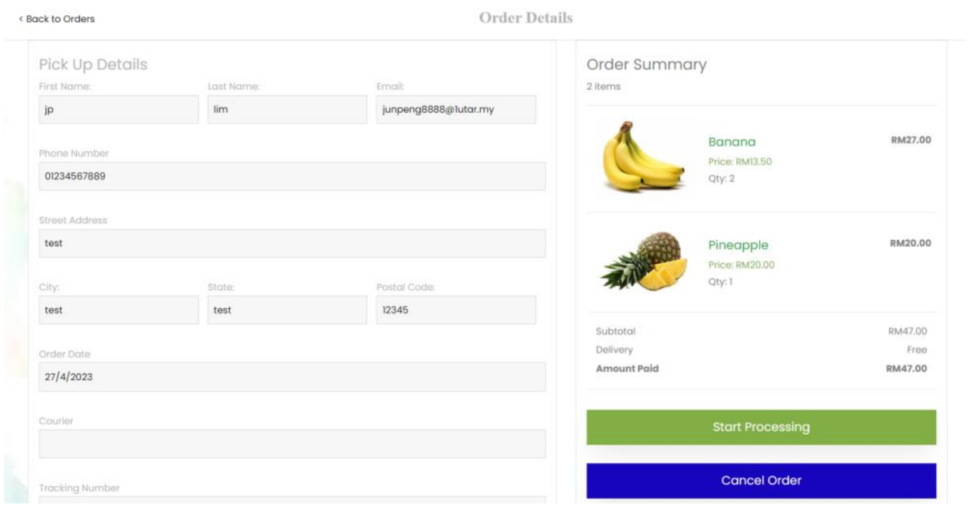


Order List

ID	FirstName	LastName	Email	Phone Number	Order Status	Amount	Action
27	Jun Peng	Lim	01124114768	junpeng8888@gmail.com	Processing	41.00	Edit
28	Jun Peng	Lim	01124114768	junpeng8888@gmail.com	Processing	20.00	Edit
29	Jun Peng	Lim	+601124114768	junpeng8888@gmail.com	Approved	20.00	Edit
30	Jun Peng	Lim	+601124114768	junpeng8888@gmail.com	Approved	20.50	Edit
31	Jun Peng	Lim	01124114768	junpeng8888@gmail.com	Shipped	22.50	Edit
32	jp	lim	01234567889	junpeng8888@1utar.my	Refunded	2.00	Edit
33	Jun Peng	Lim	01124114768	junpeng8888@gmail.com	Approved	2.00	Edit

Figure 5.4.50 Order history dashboard in administrator account

2. Administrators can click the start processing button to change the order status from “Approved” to “Processing”. They can cancel the order for special scenario like the customers.


‹ Back to Orders

Order Details

Pick Up Details

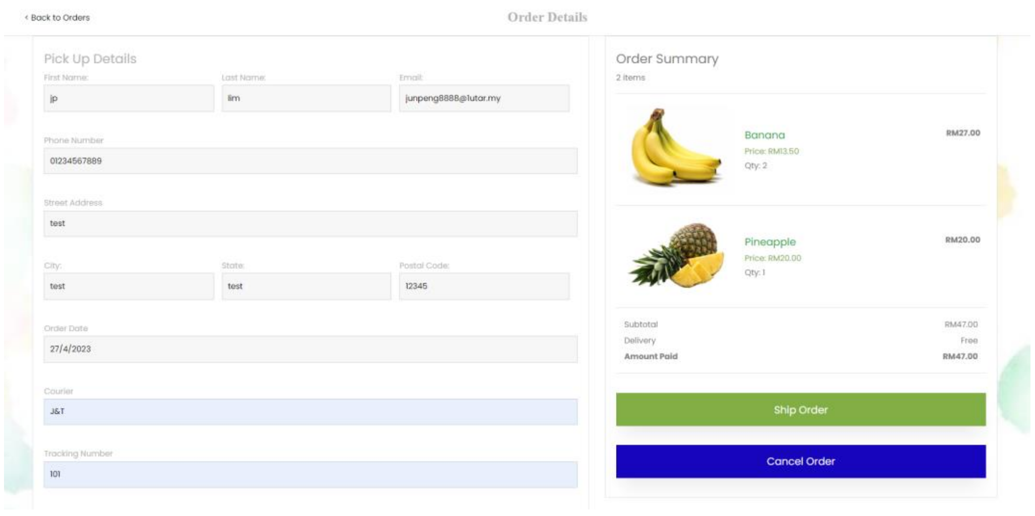
First Name: jp
Last Name: lim
Email: junpeng8888@1utar.my
Phone Number: 01234567889
Street Address: test
City: test
State: test
Postal Code: 12345
Order Date: 27/4/2023
Courier:
Tracking Number:

Order Summary
2 items

Banana Price: RM13.50 Qty: 2	RM27.00
Pineapple Price: RM20.00 Qty: 1	RM20.00
Subtotal	RM47.00
Delivery	Free
Amount Paid	RM47.00

Figure 5.4.51 Order details with approved status

3. If the order status has changed to “Processing”, administrators can click the ship order button associated with filling up the shipping information to change the order status to “Shipped”.


‹ Back to Orders

Order Details

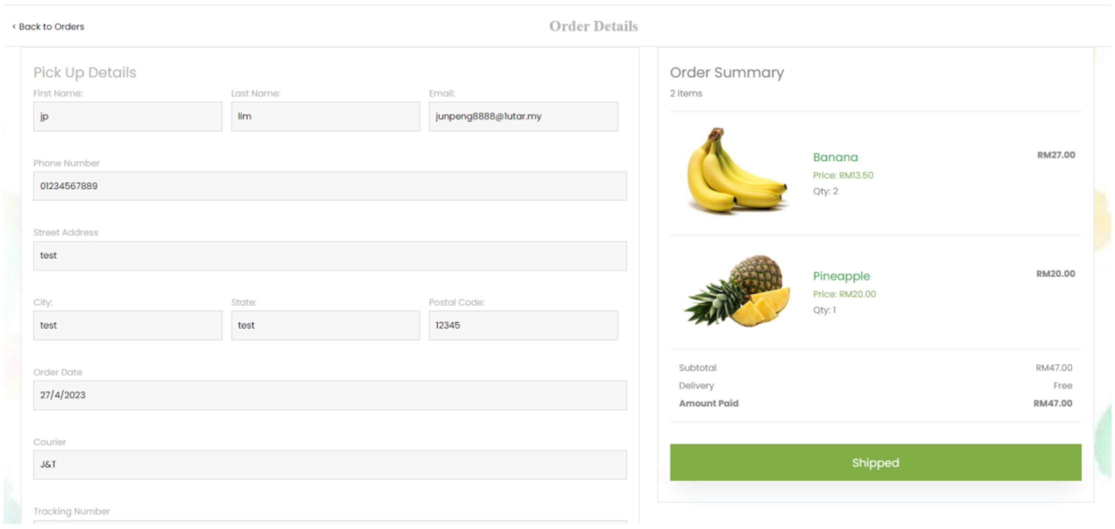
Pick Up Details

First Name: jp
Last Name: lim
Email: junpeng8888@1utar.my
Phone Number: 01234567889
Street Address: test
City: test
State: test
Postal Code: 12345
Order Date: 27/4/2023
Courier: J&T
Tracking Number: 101

Order Summary
2 items

Banana Price: RM13.50 Qty: 2	RM27.00
Pineapple Price: RM20.00 Qty: 1	RM20.00
Subtotal	RM47.00
Delivery	Free
Amount Paid	RM47.00

Figure 5.4.52 Order details with processing status



Order Details

Pick Up Details

First Name: jp	Last Name: lim	Email: junpeng8888@1utar.my
Phone Number: 01234567889		
Street Address: test		
City: test	State: test	Postal Code: 12345
Order Date: 27/4/2023		
Courier: J&T		
Tracking Number		

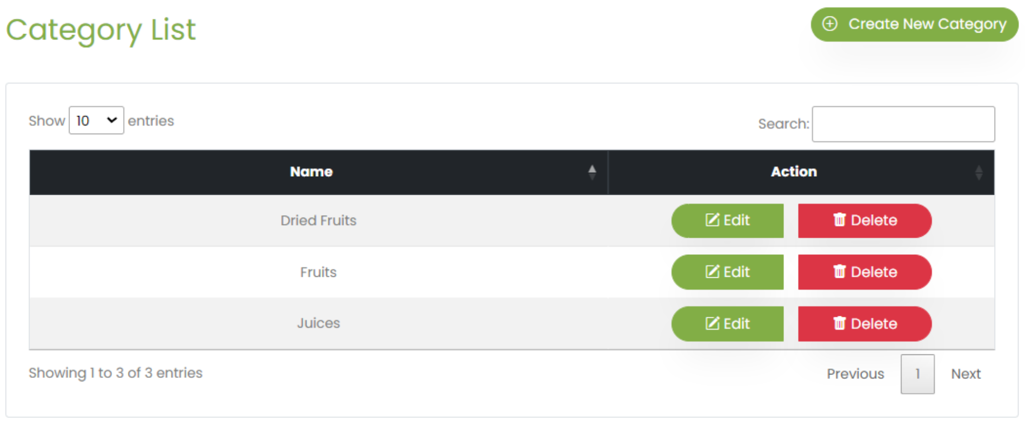
Order Summary
2 items

Banana Price: RM13.50 Qty: 2	RM27.00
Pineapple Price: RM20.00 Qty: 1	RM20.00
Subtotal	RM47.00
Delivery	Free
Amount Paid	**RM47.00**

Figure 5.4.53 Order details with shipped status

5.4.15 Category module managed by administrator

1. The administrator can also manage the category using the category dashboard. The action which can be done is similar to section 5.4.6 – 5.4.8 in which they can create, update and delete the category.


Category List

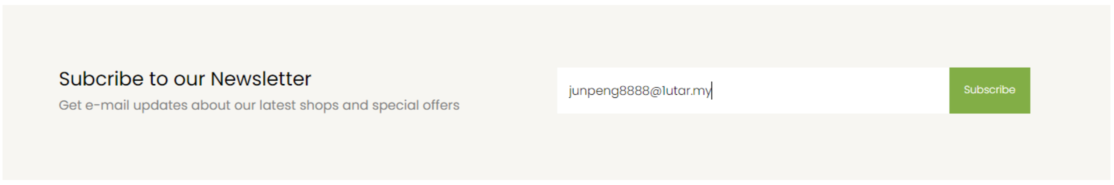
Name	Action
Dried Fruits	Edit Delete
Fruits	Edit Delete
Juices	Edit Delete

Showing 1 to 3 of 3 entries

Figure 5.4.54 Category dashboard in administrator account

5.4.16 Subscribe Newsletter

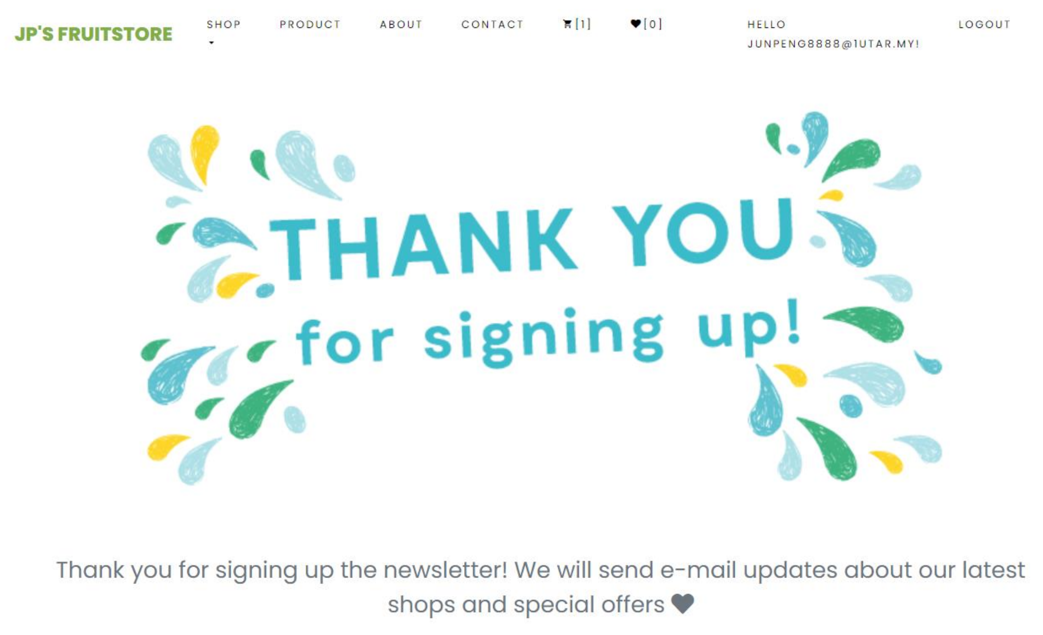
1. Customers can provide their email and subscribe to the newsletter so that they can get e-mail updates about the latest shops and special offers. After they click the subscribe button, the sign-up newsletter success page will be displayed.


Subcribe to our Newsletter

Get e-mail updates about our latest shops and special offers

junpeng8888@1utar.my

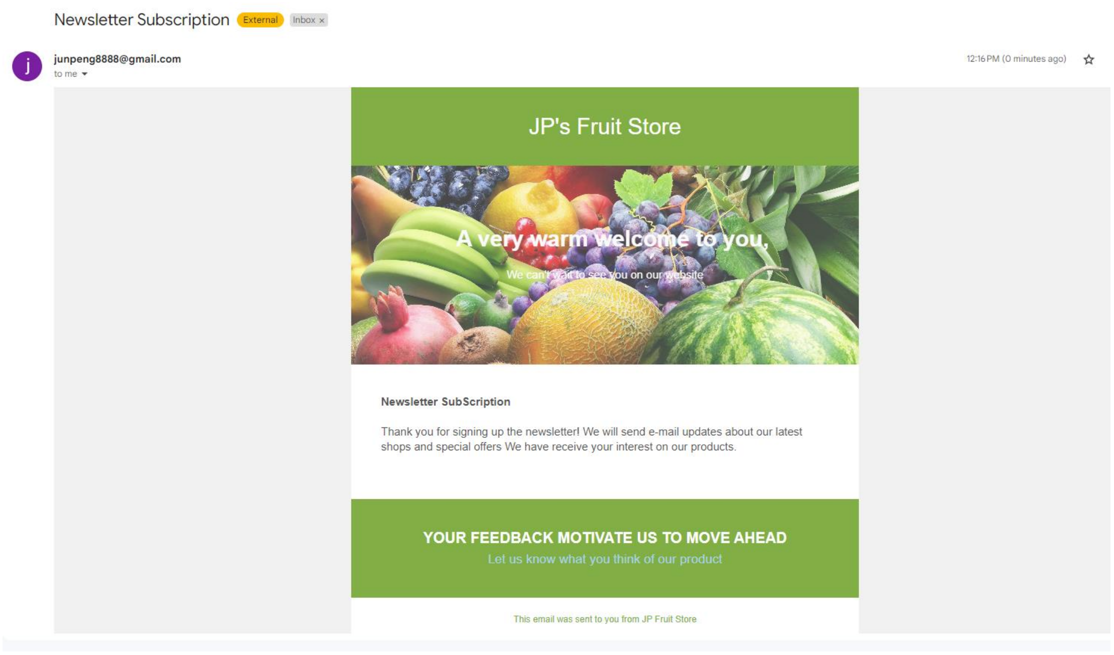
Figure 5.4.55 Subscribe newsletter section



Thank you for signing up the newsletter! We will send e-mail updates about our latest shops and special offers

Figure 5.4.56 Subscribe newsletter success page

2. They will receive an email which informs users that the newsletter has subscribed successfully.


Newsletter Subscription

junpeng8888@gmail.com

to me

12:16 PM (0 minutes ago)

JP's Fruit Store

A very warm welcome to you,

We can't wait to see you on our website

Newsletter SubScription

Thank you for signing up the newsletter! We will send e-mail updates about our latest shops and special offers We have receive your interest on our products.

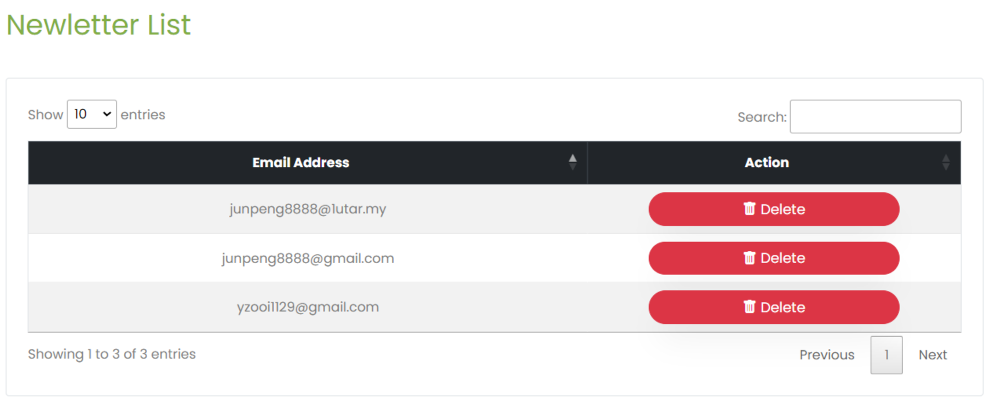
YOUR FEEDBACK MOTIVATE US TO MOVE AHEAD

Let us know what you think of our product

This email was sent to you from JP Fruit Store

Figure 5.4.57 Subscribe newsletter success message in email

3. After the user subscribed the newsletter, their email will appear in the newsletter dashboard and only the administrators can view or delete them.


Newletter List

Show 10 entries   Search:

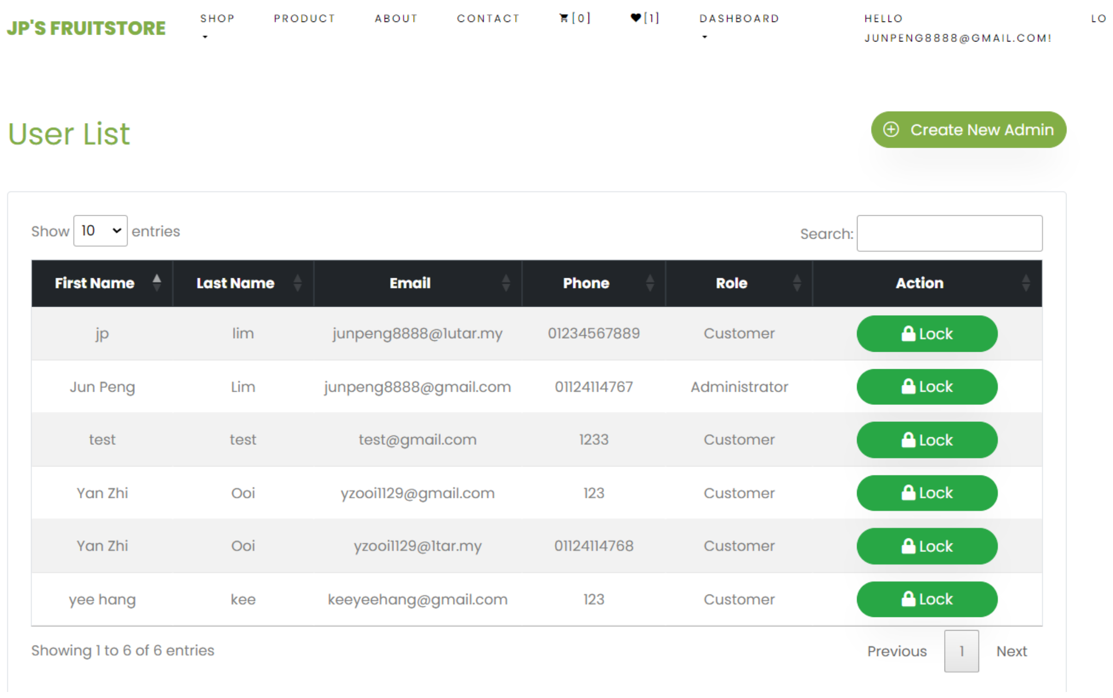
Email Address	Action
junpeng8888@1utar.my	Delete
junpeng8888@gmail.com	Delete
yzooi1129@gmail.com	Delete

Showing 1 to 3 of 3 entries   Previous 1 Next

Figure 5.4.58 Newsletter dashboard in administrator account

5.4.17 User module managed by administrator

1. The administrator can also manage the user using the user dashboard. All the user details will be retrieved from the database and displayed here. Administrators can perform lock and unlock operation.



User List

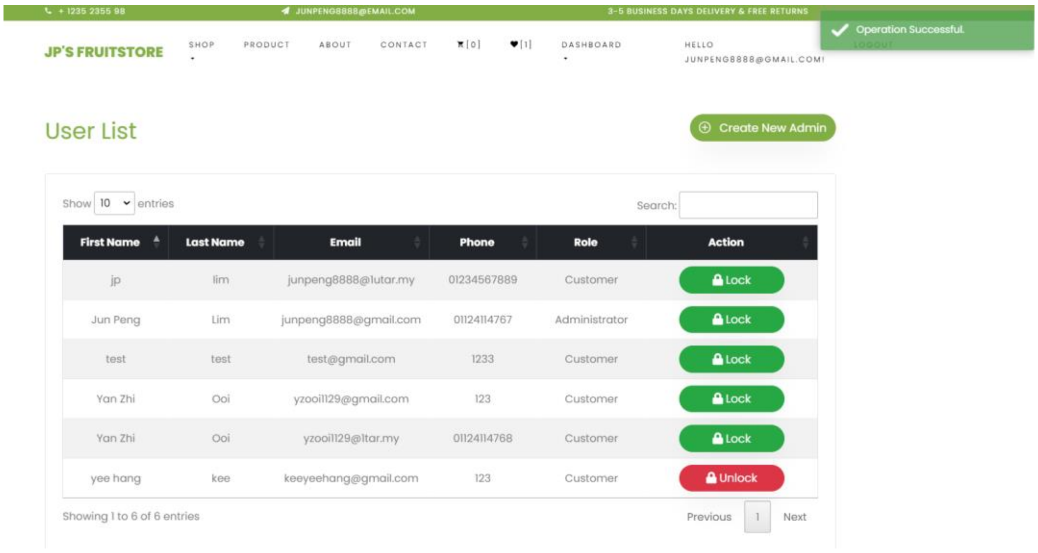
Show 10 entries   Search:

First Name	Last Name	Email	Phone	Role	Action
jp	lim	junpeng8888@1utar.my	01234567889	Customer	Lock
Jun Peng	Lim	junpeng8888@gmail.com	01124114767	Administrator	Lock
test	test	test@gmail.com	1233	Customer	Lock
Yan Zhi	Ooi	yzooi1129@gmail.com	123	Customer	Lock
Yan Zhi	Ooi	yzooi1129@1tar.my	01124114768	Customer	Lock
yee hang	kee	keeyeehang@gmail.com	123	Customer	Lock

Showing 1 to 6 of 6 entries   Previous 1 Next

Figure 5.4.59 User dashboard in administrator account

2. If they click the lock button, customers will not be able to login using the account anymore in future. However, the account still can be activated back by clicking the unlock button.



User List

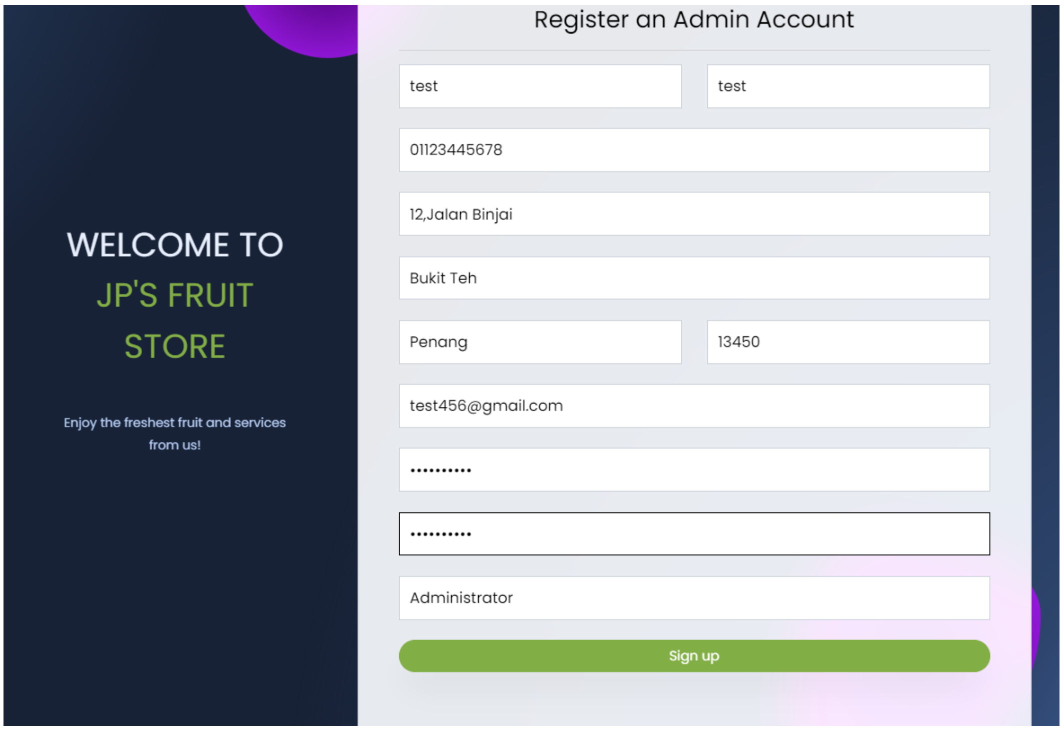
First Name	Last Name	Email	Phone	Role	Action
jp	lim	junpeng8888@1utar.my	01234567889	Customer	Lock
Jun Peng	Lim	junpeng8888@gmail.com	01124114767	Administrator	Lock
test	test	test@gmail.com	1233	Customer	Lock
Yan Zhi	Ooi	yzooi1129@gmail.com	123	Customer	Lock
Yan Zhi	Ooi	yzooi1129@1tar.my	01124114768	Customer	Lock
yee hang	kee	keeyeehang@gmail.com	123	Customer	Unlock

Showing 1 to 6 of 6 entries

Previous 1 Next

Figure 5.4.60 User dashboard in administrator account after lock the user account

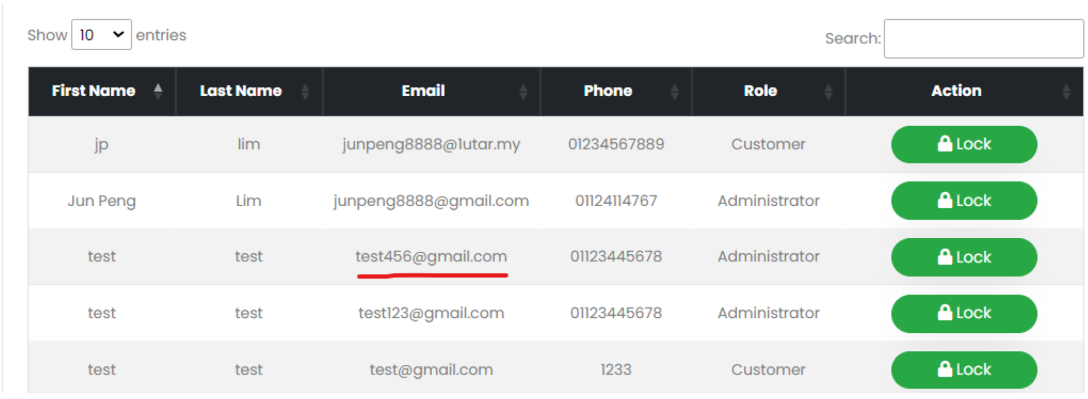
3. Only administrators can click the create new admin button and register another administrator by filling in all the required information. After that, the administrator needs to verify the email account as usual.


Register an Admin Account

test	test
01123445678	
12,Jalan Binjai	
Bukit Teh	
Penang	13450
test456@gmail.com	
••••••••••	
••••••••••	
Administrator	

Figure 5.4.61 Sign-up page to create new administrator

4. The new administrator account is successfully created.


Show 10 entries Search:

First Name	Last Name	Email	Phone	Role	Action
jp	lim	junpeng8888@1utar.my	01234567889	Customer	Lock
Jun Peng	Lim	junpeng8888@gmail.com	01124114767	Administrator	Lock
test	test	test456@gmail.com	01123445678	Administrator	Lock
test	test	test123@gmail.com	01123445678	Administrator	Lock
test	test	test@gmail.com	1233	Customer	Lock

Figure 5.4.62 New administrator added in the user dashboard

5.4.18 Multi-factor Authentication

1. The customer can set up the Two-factor authentication(2FA) for double layer security and provide their personal information when they want to login on the web application. They need to set up the authentication in the settings.

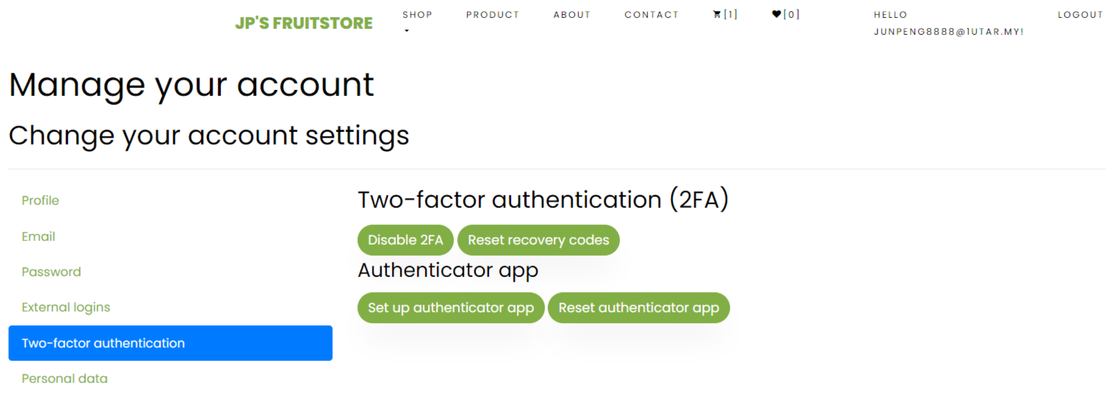
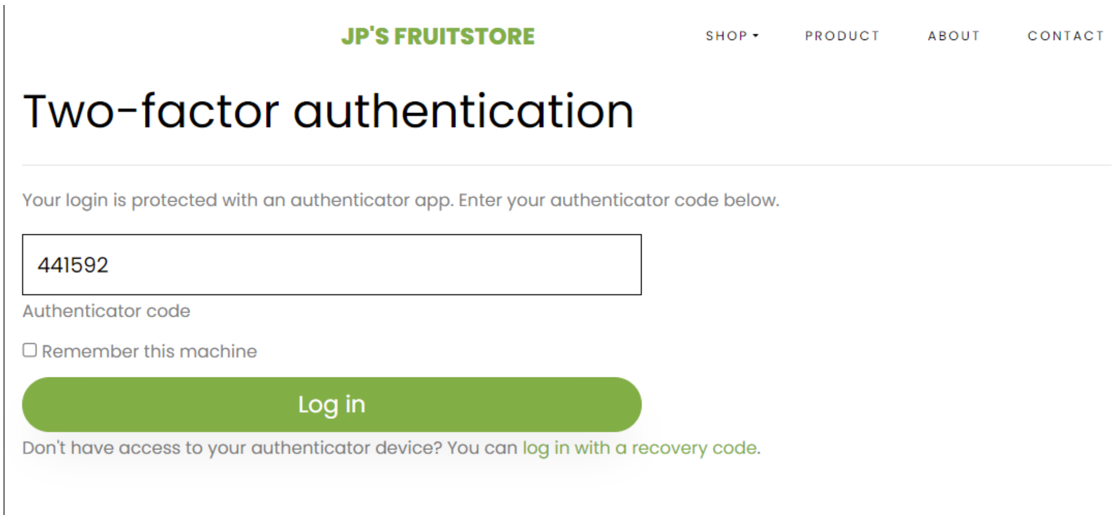


Figure 5.4.63 2FA settings is profile page

2. After setting up the 2FA, they need to download the authenticator generator tool like Microsoft authenticator to get the 6 digits code in order to login their account successfully.



Your login is protected with an authenticator app. Enter your authenticator code below.

Authenticator code: 441592

☐ Remember this machine

Don't have access to your authenticator device? You can log in with a recovery code.

Figure 5.4.64 Enter 6 digits code to pass the 2FA

5.4.19 Wish list

1. Customers can click the heart icon beside the product name to add the product to the wish list. Once clicked it, the number beside the heart icon on the navigation will change from 0 to 1.

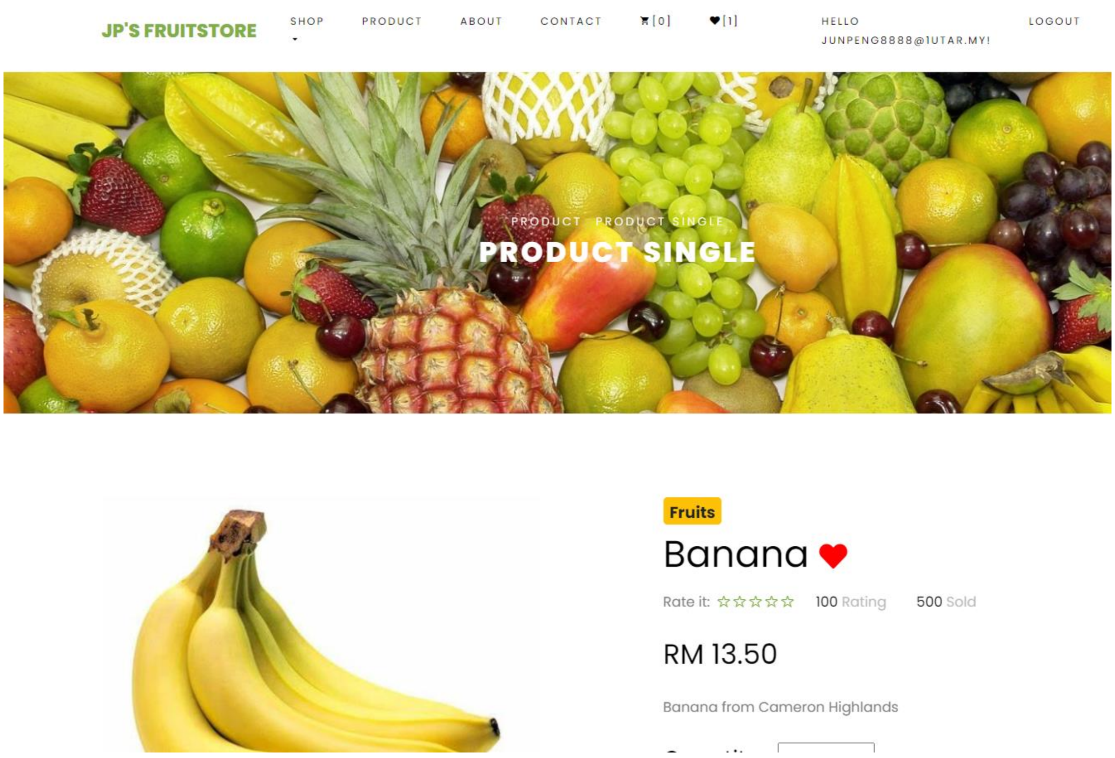
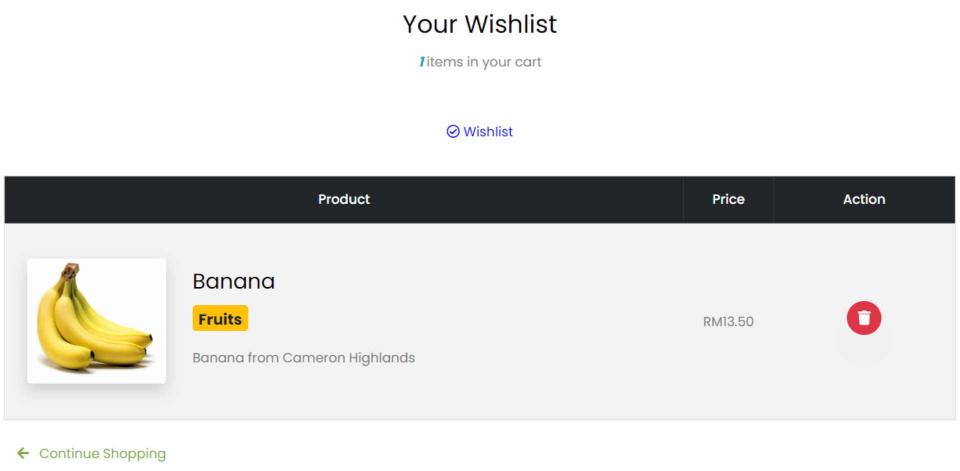


Figure 5.4.65 Product single page which the product has been added to wish list

2. When a customer clicks the heart icon on the navigation, the people will be redirected to the wish list page and the customer can view or delete it from the wish list as they wish.



Your Wishlist

1 items in your cart

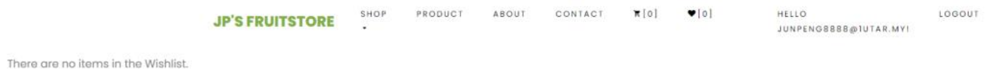
Wishlist

Product	Price	Action
Banana Fruits Banana from Cameron Highlands	RM13.50	

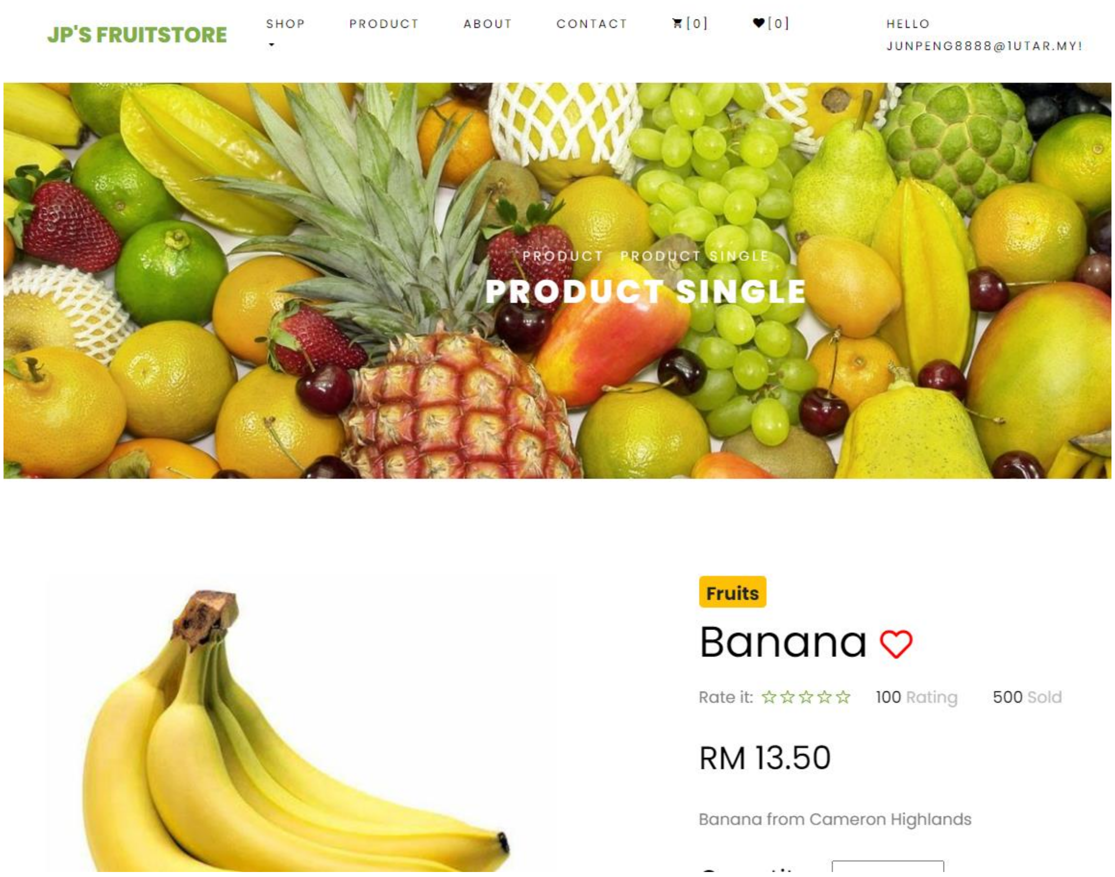
Continue Shopping

Figure 5.4.66 Wish list page

3. If a customer clicks the delete icon, the item will be gone and the number beside the heart icon on the navigation will change back to 0. The red color heart icon in the product single page will also change back to empty unless customer adds it to wish list again in future.



There are no items in the Wishlist.



Banana

Rate it: 100 Rating 500 Sold

RM 13.50

Banana from Cameron Highlands

Figure 5.4.67 Product single page which the product has not been added to wish list

5.4.20 Rate Product

1. Customer can rate the product from 1 star up to 5 stars on the single product page.

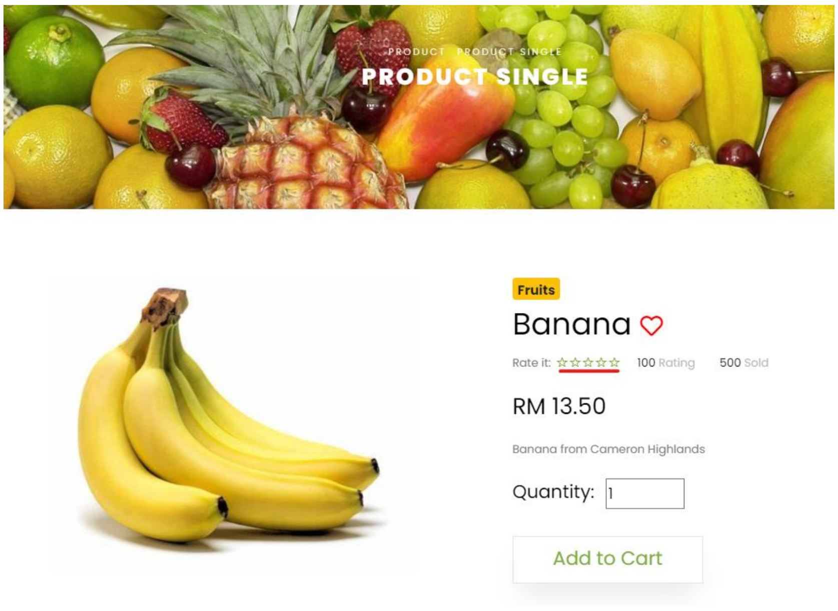


Figure 5.4.68 Product single page

2. Once the customer gives the rating, the ratings will be processed and counted in the behind logic. When a customer returns to the product view page, they will see the ratings number will change. For this example, initially there is no rating but after clicking 5 stars for banana, the changes of rating will display in advance.

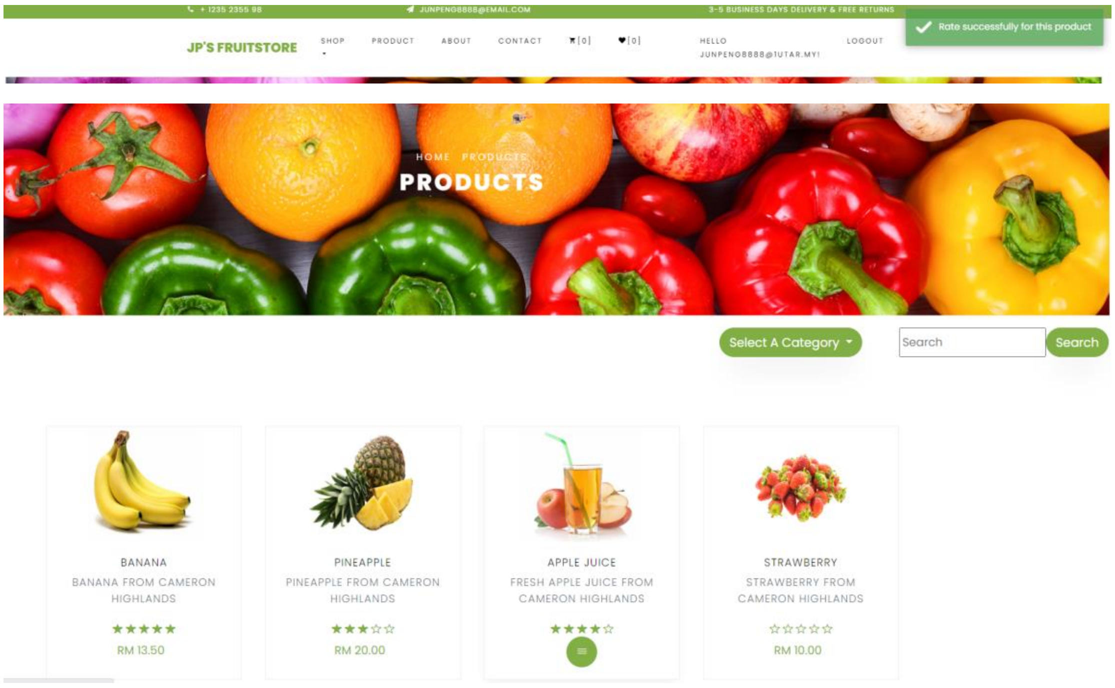

+ 1235 2355 98
JUNPENG8888@EMAIL.COM
3-5 BUSINESS DAYS DELIVERY & FREE RETURNS
Rate successfully for this product
JP'S FRUITSTORE
SHOP
PRODUCT
ABOUT
CONTACT
[0]
[0]
HELLO JUNPENG8888@1UTAR.MY!
LOGOUT
HOME PRODUCTS
PRODUCTS
Select A Category
Search
Search
BANANA
BANANA FROM CAMERON HIGHLANDS
★★★★★
RM 13.50
PINEAPPLE
PINEAPPLE FROM CAMERON HIGHLANDS
★★★☆☆
RM 20.00
APPLE JUICE
FRESH APPLE JUICE FROM CAMERON HIGHLANDS
★★★★☆
STRAWBERRY
STRAWBERRY FROM CAMERON HIGHLANDS
☆☆☆☆☆
RM 10.00

Figure 5.4.69 Product view page

5.5 Implementation Issues and Challenges

Along the way of implementation and programming, indeed I had encountered some challenges and problems. First and foremost, it was quite complicated for me to start up the system during the initial project development. This is primarily because there is less exposure on certain software framework, tools and programing language. For instance, MVC is the design pattern and architecture used in this proposed web application. Thus, appropriate knowledge about the theory and practical skills should be possessed. I need to pay effort and time on understanding the concept of Model-View-Controller and way to implement it in real coding. In addition, I also spent some time on connecting and configuring the database to the web application. It requires me to be familiarise with the Microsoft Visual Studio IDE and also SQL Server Management Studio (SSMS). This is a mandatory step for me to program the CRUD operation for all the table. Understanding migrations in entity framework and entity transformation into SQL has never been an easy-peasy task along the implementation. Fortunately, with the help of online tutorials and helpful resources, I can manage to build up tables in the database, complete all the CRUD operation in this project and finally display the output in the client-side part of apps. Besides, some effort and time are required to learn the syntax of C# language and the implementation in the client and server-side coding, especially all the module needed to be finished in this semester. Moreover, I also face some challenge on configuring all the services like stripe payment, email services and Twilio message services. Fortunately, there are some tutorial videos online that provide me some guideline and help.

Other than that, it costs me a lot of time to learn how to create the recommender models using python code. At first, I face some difficulty on how to start the project. Thus, I need to allocate more time on learning the concept of recommender system and the suitable machine learning algorithm. Other than that, I quite struggled when deploying the web application in Flask and treat it as the API to return product recommendations. It requires some basic knowledge in Flask and Restful API. Much time has been used for experiments and development to integrate Flask and the proposed web application

CHAPTER 6

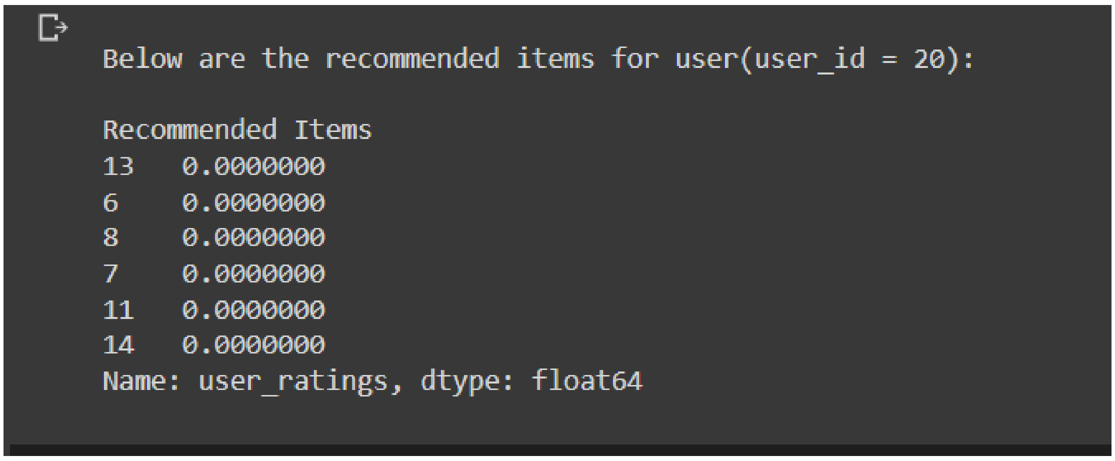
System Evaluation and Discussion

In this chapter, there will be some discussion on system testing and performance metrics. Testing setup and result will be included. Besides, there will be some discussion on the objective evaluation.

6.1 System Testing, Testing Setup and Results

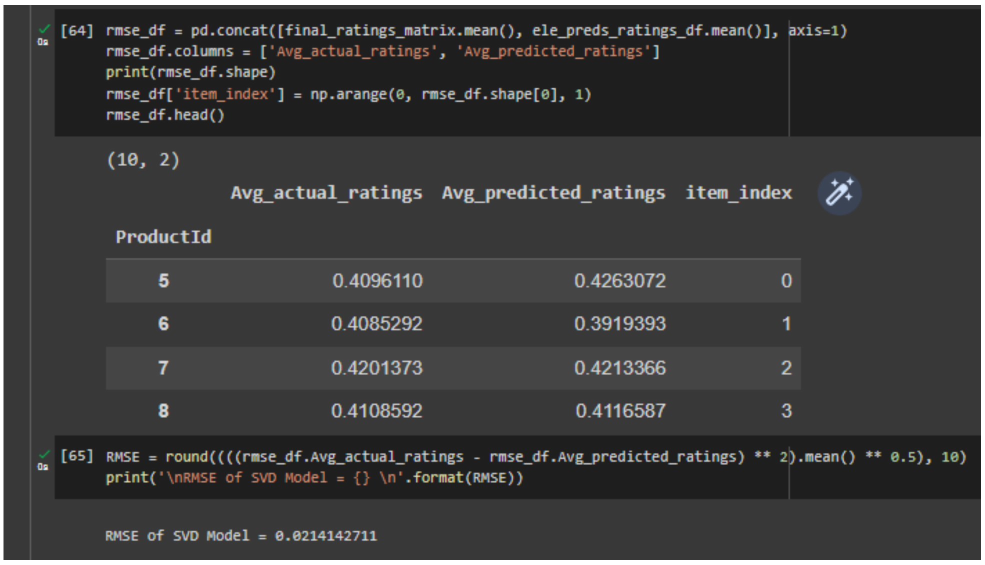
6.1.1 Results of User-Based Collaborative Filtering Recommender using SVD algorithm from Scipy library

For Scipy does not provide an implementation of the SVD-based recommender algorithm. So, we can use the RMSE formula to count the RMSE with the information of actual and predicted ratings. The lower the RMSE indicates a better result. The first model is already good to go in which it can provide the product recommendations when userID is passing as input as shown in Figure 6.1.1. User-based collaborative filtering recommender will generate some recommendations which customers may give high ratings based on the customer rating history by finding the group of customers who share the similar product ratings and compare their similarity between two users. However, the RMSE we get is around 0.021 might seem to be over satisfied which may have overfitting problem. In future, this model can be improved by using other algorithm like KNN or SVD which provided in the surprise library as shown in the next section.



```
Below are the recommended items for user(user_id = 20):

Recommended Items
13   0.0000000
6    0.0000000
8    0.0000000
7    0.0000000
11   0.0000000
14   0.0000000
Name: user_ratings, dtype: float64
```

Figure 6.1.1 Recommendations by user-based collaborative filtering recommender



```
[64] rmse_df = pd.concat([final_ratings_matrix.mean(), ele_preds_ratings_df.mean()], axis=1)
     rmse_df.columns = ['Avg_actual_ratings', 'Avg_predicted_ratings']
     print(rmse_df.shape)
     rmse_df['item_index'] = np.arange(0, rmse_df.shape[0], 1)
     rmse_df.head()

(10, 2)
```

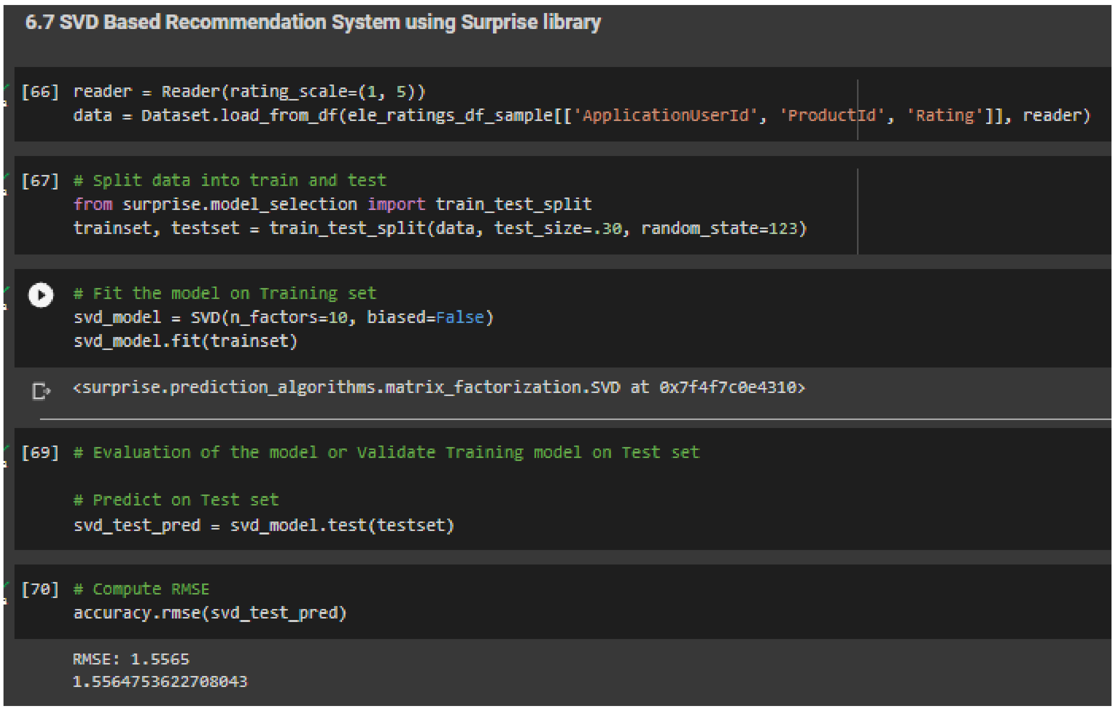
ProductId	Avg_actual_ratings	Avg_predicted_ratings	item_index
5	0.4096110	0.4263072	0
6	0.4085292	0.3919393	1
7	0.4201373	0.4213366	2
8	0.4108592	0.4116587	3

```
[65] RMSE = round((((rmse_df.Avg_actual_ratings - rmse_df.Avg_predicted_ratings) ** 2).mean() ** 0.5), 10)
     print('\nRMSE of SVD Model = {} \n'.format(RMSE))

RMSE of SVD Model = 0.0214142711
```

Figure 6.1.2 Results by user-based collaborative filtering recommender

6.1.2 Results of User-Based Collaborative Filtering Recommender Model using SVD algorithm from Surprise library

The SVD algorithm from Surprise library generate a more neutral result which may not suffer from the overfitting problem. The RMSE of the SVD algorithm from Surprise library is around 1.5565.



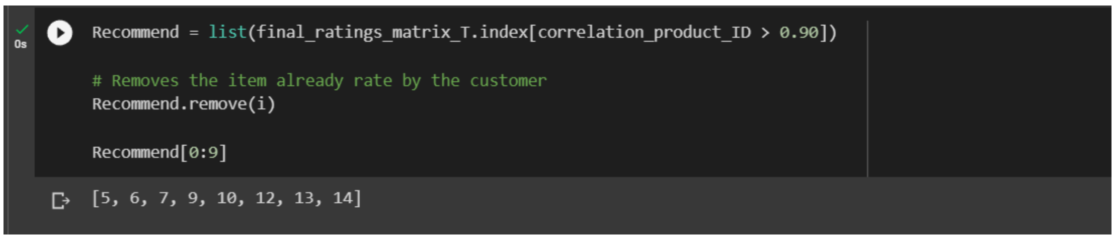
6.7 SVD Based Recommendation System using Surprise library

```
[66] reader = Reader(rating_scale=(1, 5))
     data = Dataset.load_from_df(ele_ratings_df_sample[['ApplicationUserId', 'ProductId', 'Rating']], reader)

[67] # Split data into train and test
     from surprise.model_selection import train_test_split
     trainset, testset = train_test_split(data, test_size=.30, random_state=123)

     # Fit the model on Training set
     svd_model = SVD(n_factors=10, biased=False)
     svd_model.fit(trainset)

     <surprise.prediction_algorithms.matrix_factorization.SVD at 0x7f4f7c0e4310>

[69] # Evaluation of the model or Validate Training model on Test set

     # Predict on Test set
     svd_test_pred = svd_model.test(testset)

[70] # Compute RMSE
     accuracy.rmse(svd_test_pred)

     RMSE: 1.5565
     1.5564753622708043
```

Figure 6.1.3 Results by user-based collaborative filtering recommender using surprise library

6.1.3 Results of Item-Based Collaborative Filtering Recommender Model using SVD algorithm from Surprise library

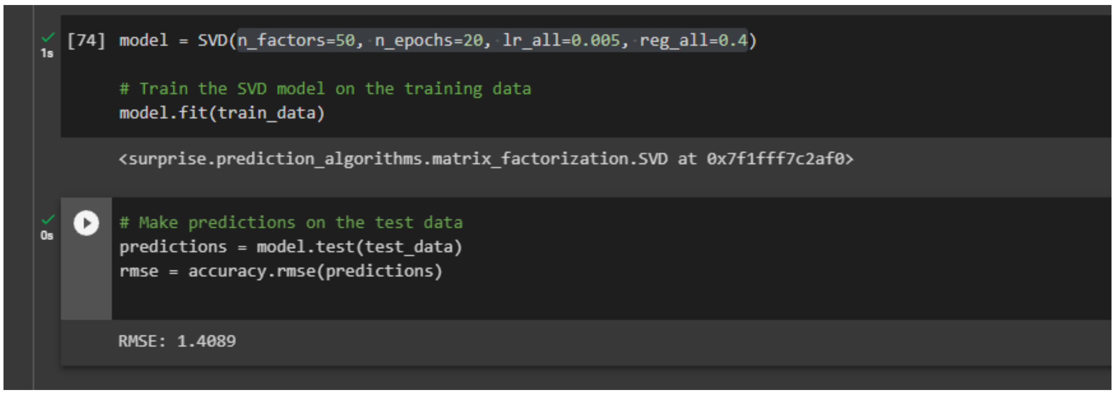
This is the item-based collaborative filtering recommender in which it can provide the product recommendations when productID is passing as input. Item-based collaborative filtering recommender will generate some recommendations which customers may give high ratings based on the customer rating history by finding every pair of the products in which one of them or both has rated by the customer and measure the similarity of the products' rating across every customer who have rated both of them. Pearson-Correlation is being used here to measure the similarity strength between the products. The RMSE we get is around 1.4089 which considered not bad about the accuracy and not suffered from the overfitting problem.



```
Recommend = list(final_ratings_matrix_T.index[correlation_product_ID > 0.90])

# Removes the item already rate by the customer
Recommend.remove(i)

Recommend[0:9]
```

```
[5, 6, 7, 9, 10, 12, 13, 14]
```

Figure 6.1.4 Recommendations by item-based collaborative filtering recommender



```
[74] model = SVD(n_factors=50, n_epochs=20, lr_all=0.005, reg_all=0.4)

     # Train the SVD model on the training data
     model.fit(train_data)
```

```
<surprise.prediction_algorithms.matrix_factorization.SVD at 0x7f1fff7c2af0>
```

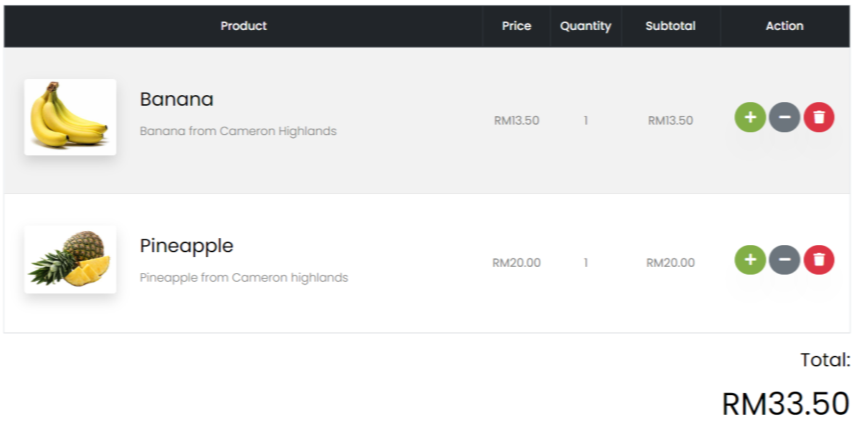
```
# Make predictions on the test data
predictions = model.test(test_data)
rmse = accuracy.rmse(predictions)
```

```
RMSE: 1.4089
```

Figure 6.1.5 Results by item-based collaborative filtering recommender

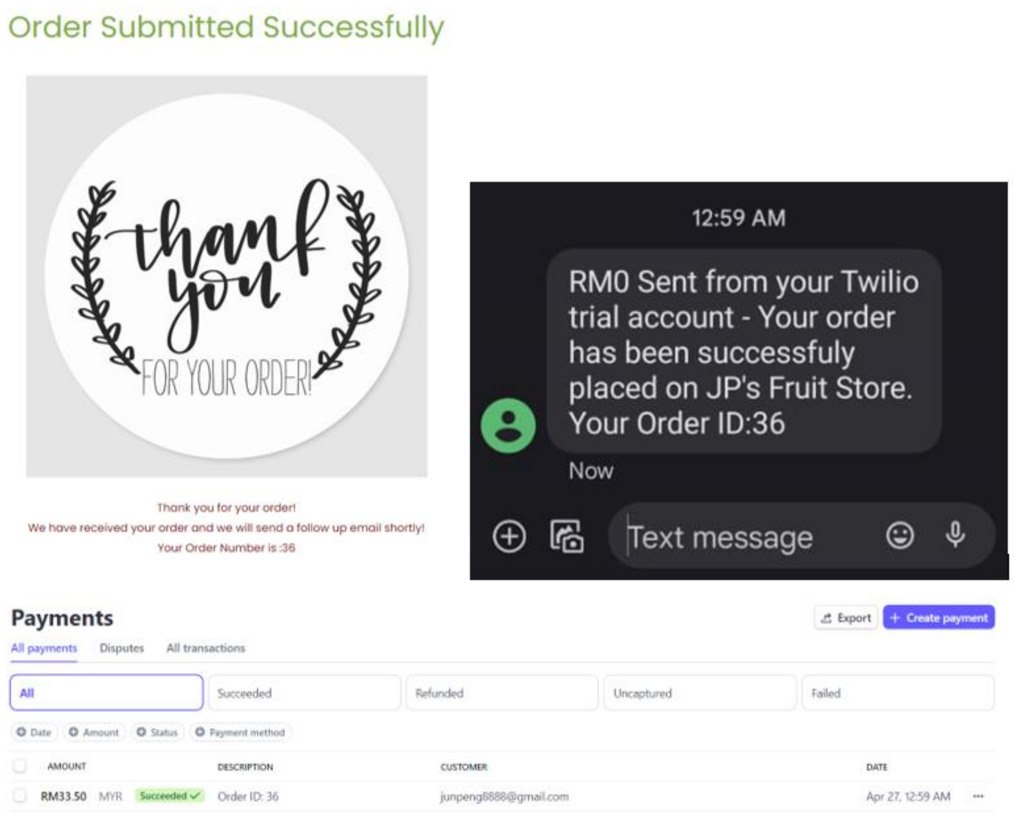
6.2 Test Cases and Results

Project Name	Modern Fruits Web Store with Personalized Recommender System
Module Name	**Place order Module (Customer)**
Reference Document	Controller: CartController.cs View: Index.cshtml, Summary.cshtml, OrderConfirmation.cshtml (all views under Cart Views folder)
Created by	Lim Jun Peng
Date of Creation	23/4/2022
Date of Review	23/4/2022
Test Case ID	TC_POM_01
Test Scenario	To verify that the customers can be able to place order on the proposed web application and the order is stored into the database.
Test Case Description	This test case is created to test the functionality of the online fruits web application with actions like add to cart, checkout, make payment and place order. This test case is also prepared to ensure the output correctness and the customer buying flow is smooth and correct.
Pre-condition	1. Customer has registered, verified their email account and successfully logged into the online web application.

	2. Customer has added some products into the shopping cart. 3. Customer clicks the cart icon and redirects to the shopping cart page to view the products in cart.
Test Step:	1. Customer clicks the “Checkout” button to proceed to order summary page. 2. Customer fills in all the required personal and shipping information in the order summary page and ready to place order. 3. Customer double checks all details and clicks the “Place Order” button to place the order. 4. Customer fills in all the personal and card information in the Stripe payment page. 5. Customer clicks the “Pay MYR RMxxx.xx” in order to make payment.
Test data	Products: Banana x1, Pineapple x1 Price: RM12.50 (Banana), RM20.00 (Pineapple) Total to pay amount: RM33.50  Product · Price · Quantity · Subtotal · Action Banana — Banana from Cameron Highlands · RM13.50 · 1 · RM13.50 Pineapple — Pineapple from Cameron highlands · RM20.00 · 1 · RM20.00 Total: RM33.50
Expected Result	After the customer has clicked the “Place Order” button to place the order and make the payment, the system will validate the transaction. Once the payment is successful, a new order record will be created and stored in the database. A message will be sent to the customer’s mobile

	phone and an order success page will be displayed. The payment transaction will also be being added into the Stripe dashboard.
Actual Result	The system is able to create a new order record with order number 36 in this example. A message will be sent to the customer's mobile phone and an order success page will be displayed. The payment transaction will also be being added into the Stripe dashboard.
Test Result	PASSED

Screenshots of final results:



Order Submitted Successfully

Thank you for your order!
We have received your order and we will send a follow up email shortly!
Your Order Number is :36

12:59 AM

RM0 Sent from your Twilio trial account - Your order has been successfuly placed on JP's Fruit Store. Your Order ID:36

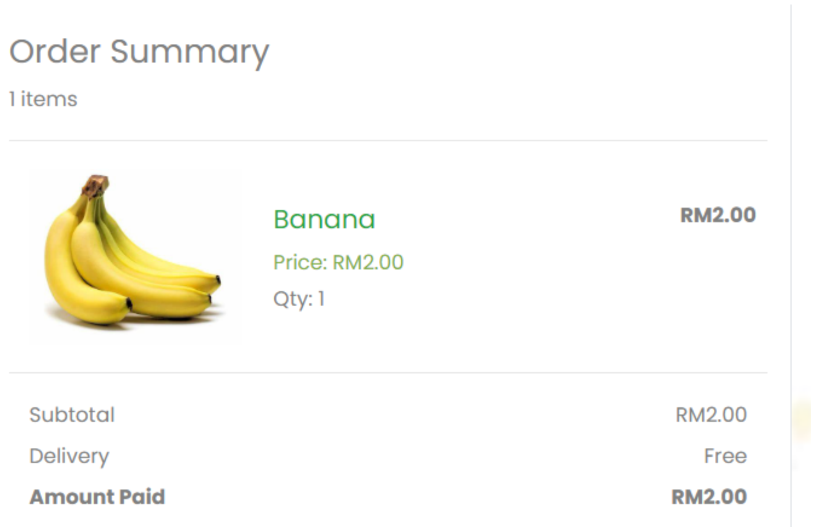
Now

Payments

AMOUNT			DESCRIPTION	CUSTOMER	DATE	
RM33.50	MYR	Succeeded ✓	Order ID: 36	junpeng8888@gmail.com	Apr 27, 12:59 AM	···

Figure 6.2.1 Test result of TC_POM_01

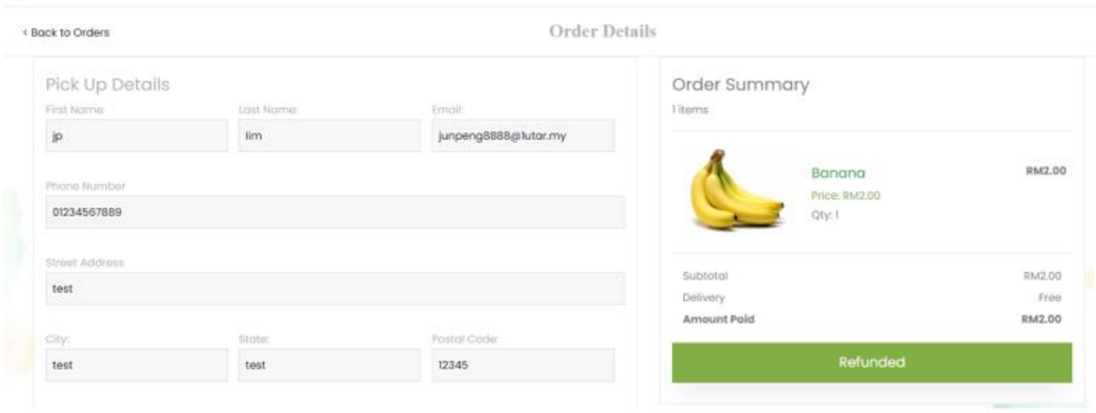
Project Name	Modern Fruits Web Store with Personalized Recommender System
Module Name	**Cancel order Module (Customer)**
Reference Document	Controller: OrderController.cs View: Index.cshtml, Details.cshtml (all views under Order Views folder)
Created by	Lim Jun Peng
Date of Creation	23/4/2022
Date of Review	23/4/2022
Test Case ID	TC_COM_01
Test Scenario	To verify that the customers can be able to cancel order on the proposed web application and the money shall be refunded to customer card account.
Test Case Description	This test case is created to test the functionality of the online fruits web application with actions like check order history and cancel order. This test case is also prepared to ensure the output correctness and the customer cancel order flow is smooth and correct.
Pre-condition	1. Customer has registered, verified their email account and successfully logged into the online web application.

	2. Customer clicks the order history navigation and redirects to the order history page to view the order which has been made. 3. The order only can be cancelled if the order status is under "Approved" or "Processing".
Test Step:	1. Customer clicks the order record that he or she wants to cancel order. 2. Customer clicks "Cancel order" button to cancel the order
Test data	Order ID: 32 Products: Banana x1 Price: RM2.00 (Banana) Total amount paid: RM2.00  Order Summary 1 items Banana RM2.00 Price: RM2.00 Qty: 1 Subtotal RM2.00 Delivery Free Amount Paid RM2.00
Expected Result	After the customer has clicked the "Cancel order" button to cancel the order, the system will process the request and refund the amount to customer. Once the cancel order request is verified, the order status will change to "Refunded". The payment transaction status will also be being updated in the Stripe dashboard.
Actual Result	The system is able to accept and verify the cancel order request for order number 32 in this example. Once the cancel order request is

	verified, the order status will change to “Refunded”. The payment transaction status will also be being updated in the Stripe dashboard.
Test Result	PASSED

Screenshots of final results:

1)



‹ Back to Orders

Order Details

Pick Up Details

First Name:	Last Name:	Email:
jp	lim	junpeng8888@lutar.my

Phone Number: 01234567889

Street Address: test

City:	State:	Postal Code:
test	test	12345

Order Summary

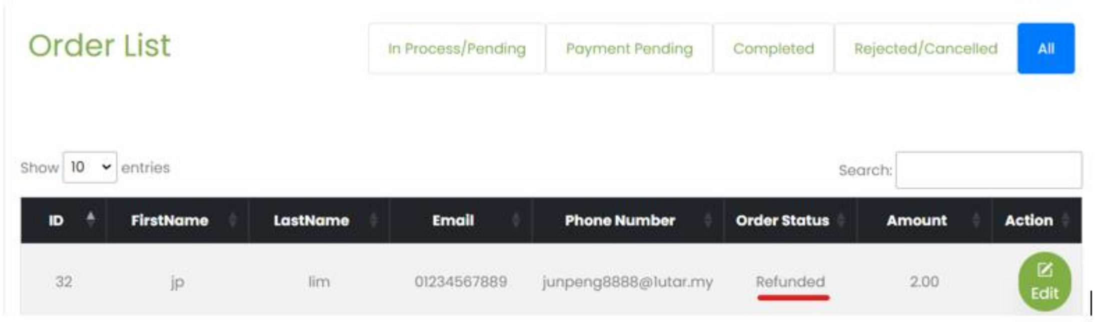
1 items

Banana — RM2.00
Price: RM2.00
Qty: 1

Subtotal	RM2.00
Delivery	Free
Amount Paid	**RM2.00**

Refunded

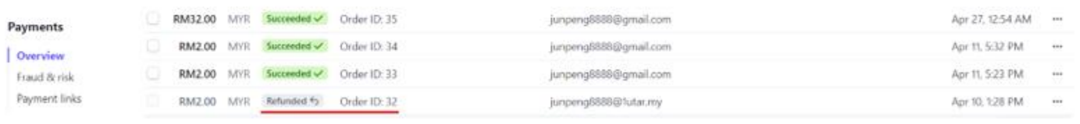
2)



Order List

ID	FirstName	LastName	Email	Phone Number	Order Status	Amount	Action
32	jp	lim	01234567889	junpeng8888@lutar.my	Refunded	2.00	Edit

3)



RM32.00	MYR	Succeeded ✓	Order ID: 35	junpeng8888@gmail.com	Apr 27, 12:54 AM
RM2.00	MYR	Succeeded ✓	Order ID: 34	junpeng8888@gmail.com	Apr 11, 5:32 PM
RM2.00	MYR	Succeeded ✓	Order ID: 33	junpeng8888@gmail.com	Apr 11, 5:23 PM
RM2.00	MYR	Refunded	Order ID: 32	junpeng8888@1utar.my	Apr 10, 1:28 PM

Figure 6.2.2 Test result of TC_COM_01

Project Name	Modern Fruits Web Store with Personalized Recommender System
Module Name	**Manage Product Module (Administrator)**
Reference Document	Controller: ProductCrudController.cs View: Index.cshtml, Upsert.cshtml (all views under ProductCrud Views folder)
Created by	Lim Jun Peng
Date of Creation	23/4/2022
Date of Review	23/4/2022
Test Case ID	TC_MPM_01
Test Scenario	To verify that the administrator can be able to perform product CRUD operation and management on the proposed web application.
Test Case Description	This test case is created to test the functionality of the product dashboard of online fruits web application with actions like create, update and delete products. This test case is also prepared to ensure the output correctness and the actions mentioned can done smoothly and correctly.
Pre-condition	1. Administrator has successfully logged into the online web application.

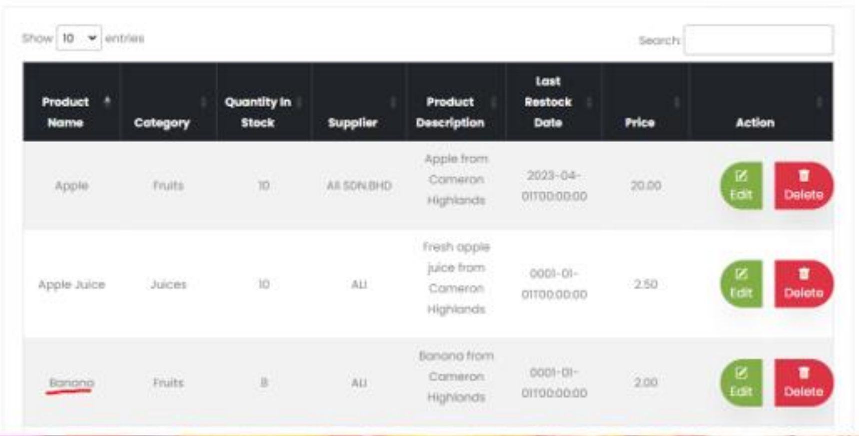
	2. Administrator clicks the dashboard navigation and redirects to the product dashboard page to perform management on products which are currently selling. 3. Only administrator accounts can access the dashboard.
Test Step:	<u>Create new product:</u> 1. Administrator clicks “Create New Product” button. 2. Administrator fills in all the required information of new products and upload the associated image. 3. Administrator clicks “Create” button. <u>Edit existing product:</u> 1. Administrator selects the product which needs to be edited from the dashboard and clicks the “Edit” button. 2. Administrator fills in the updated information of products or uploads new image. 3. Administrator clicks “Update” button. <u>Delete existing product:</u> 1. Administrator selects the product which needs to be deleted from the dashboard and clicks the “Delete” button. 2. Administrator clicks “OK” button when the delete product confirmation message pops out.
Test data	<u>Create new product:</u> Banana with its associated information. <u>Edit existing product:</u> Change the price of banana from RM2.00 to RM13.50.

	<u>Delete existing product:</u> Delete the test product called test with its associated information.
Expected Result	<u>Create new product:</u> The new product should be added successfully after administrator clicks the “Create” button. The new product will also be displayed on the product view page. <u>Edit existing product:</u> The product information should be able to be updated successfully after administrator clicks the “Update” button. The new product information will also be displayed on the product view page. <u>Delete existing product:</u> The product should be able to be deleted successfully after administrator clicks the “Delete” button. The product will no longer exist in the dashboard or displayed in the product view page.
Actual Result	<u>Create new product:</u> The new product “banana” is added successfully after administrator clicks the “Create” button. It is then displayed on the product view page. <u>Edit existing product:</u> The price of banana is updated successfully from RM2.00 to RM13.50 after administrator clicks the “Update” button. The new changes in the price of banana will also be displayed on the product view page. <u>Delete existing product:</u>

	The product "test" is deleted successfully after administrator clicks the "Delete" button. It no longer exists in the dashboard or displayed in the product view page.
Test Result	PASSED

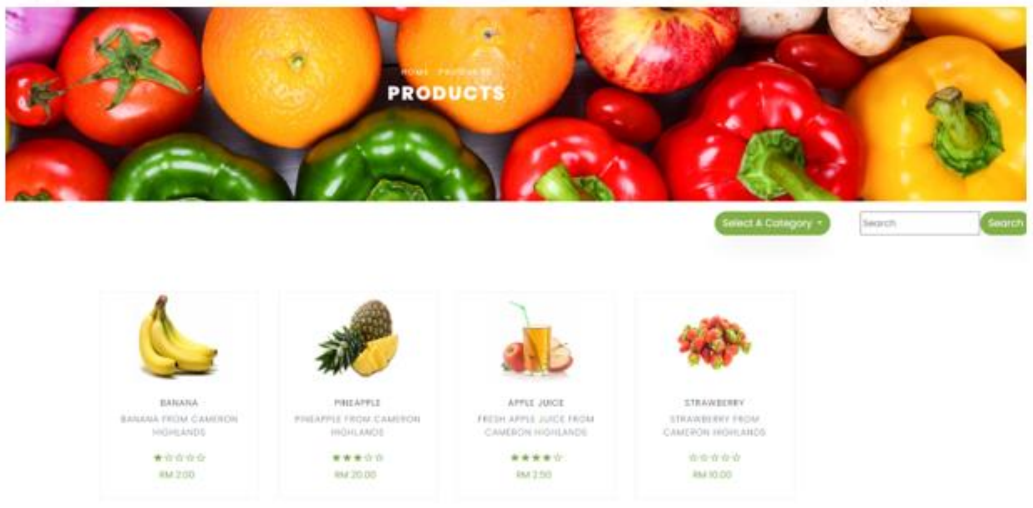
Screenshots of final results:

1) After creating the new product "banana":



Show 10 entries Search:

Product Name	Category	Quantity in Stock	Supplier	Product Description	Last Restock Date	Price	Action
Apple	Fruits	10	Ali SDN BHD	Apple from Cameron Highlands	2023-04-01T00:00:00	20.00	Edit Delete
Apple Juice	Juices	10	ALI	Fresh apple juice from Cameron Highlands	0001-01-01T00:00:00	2.50	Edit Delete
Banana	Fruits	8	ALI	Banana from Cameron Highlands	0001-01-01T00:00:00	2.00	Edit Delete



BANANA
BANANA FROM CAMERON HIGHLANDS
★☆☆☆☆
RM 2.00

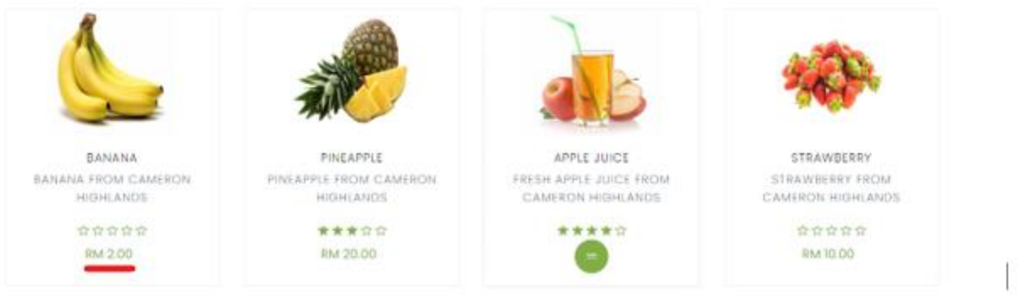
PINEAPPLE
PINEAPPLE FROM CAMERON HIGHLANDS
★★★☆☆
RM 20.00

APPLE JUICE
FRESH APPLE JUICE FROM CAMERON HIGHLANDS
★★★★☆
RM 2.50

STRAWBERRY
STRAWBERRY FROM CAMERON HIGHLANDS
☆☆☆☆☆
RM 10.00

Figure 6.2.3 Test result of TC_MPM_01

2a) Before editing the price of product “banana”:



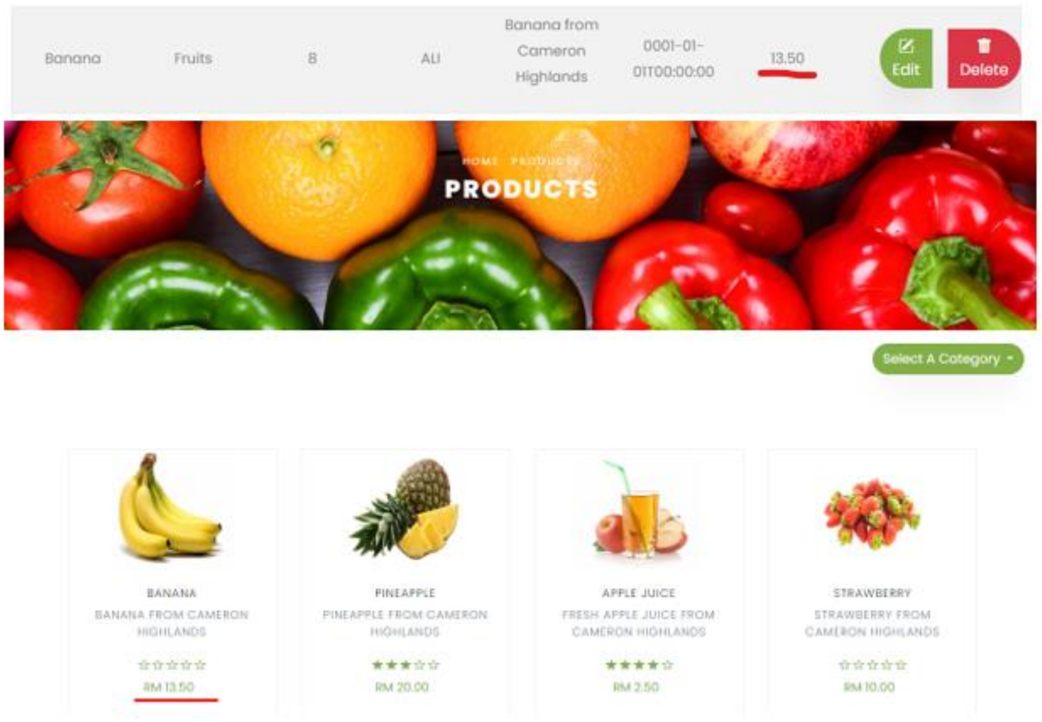
BANANA
BANANA FROM CAMERON HIGHLANDS
RM 2.00

PINEAPPLE
PINEAPPLE FROM CAMERON HIGHLANDS
RM 20.00

APPLE JUICE
FRESH APPLE JUICE FROM CAMERON HIGHLANDS

STRAWBERRY
STRAWBERRY FROM CAMERON HIGHLANDS
RM 10.00

2b) After editing the price of product “banana”:



Banana	Fruits	8	ALI	Banana from Cameron Highlands	0001-01-01T00:00:00	13.50	Edit Delete

BANANA
BANANA FROM CAMERON HIGHLANDS
RM 13.50

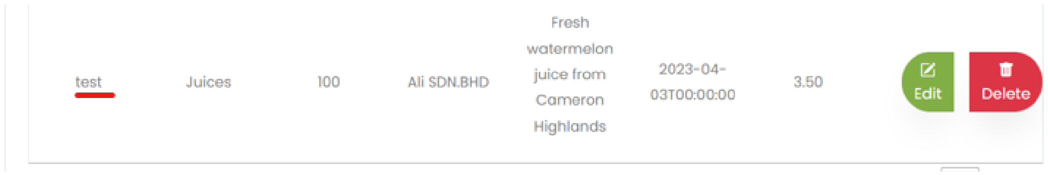
PINEAPPLE
PINEAPPLE FROM CAMERON HIGHLANDS
RM 20.00

APPLE JUICE
FRESH APPLE JUICE FROM CAMERON HIGHLANDS
RM 2.50

STRAWBERRY
STRAWBERRY FROM CAMERON HIGHLANDS
RM 10.00

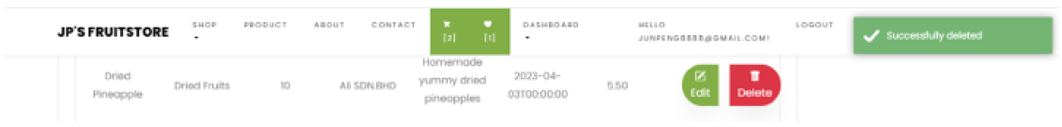
Figure 6.2.4 Test result of TC_MPM_01

3a) Before deleting the product “test”:



test	Juices	100	Ali SDN.BHD	Fresh watermelon juice from Cameron Highlands	2023-04-03T00:00:00	3.50	Edit Delete

3b) After deleting the product “test”:

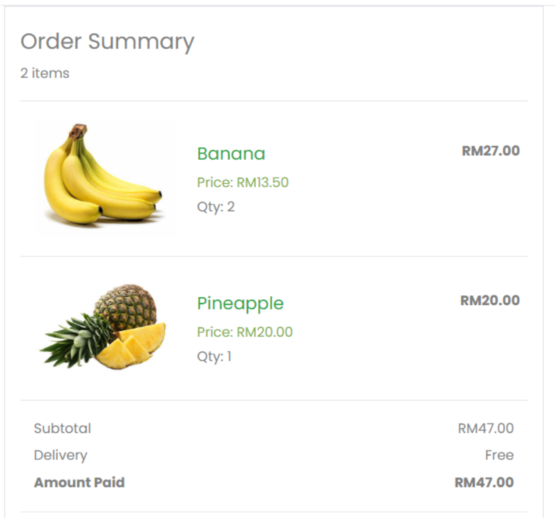


Successfully deleted

Dried Pineapple	Dried Fruits	10	Ali SDN.BHD	Homemade yummy dried pineapples	2023-04-03T00:00:00	5.50	Edit Delete

Figure 6.2.5 Test result of TC_MPM_01

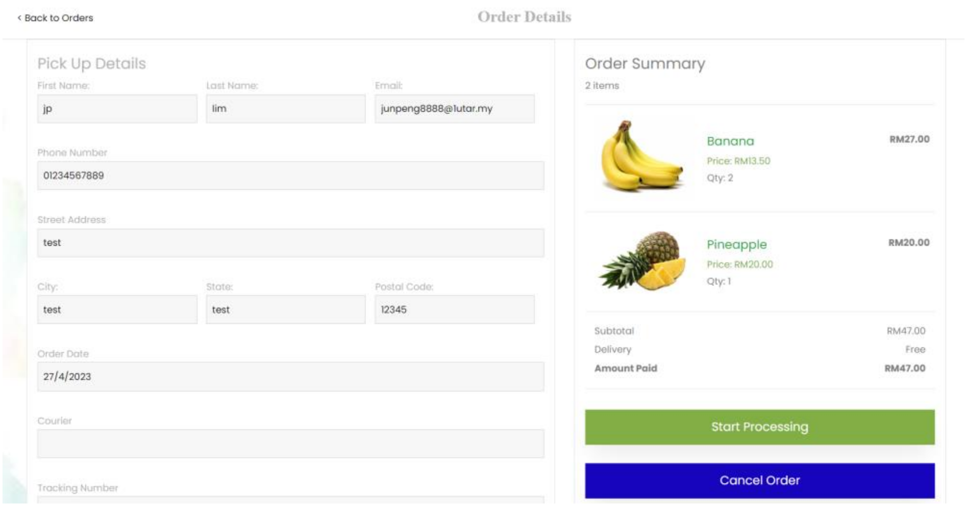
Project Name	Modern Fruits Web Store with Personalized Recommender System
Module Name	**Update order status Module (Administrator)**
Reference Document	Controller: OrderController.cs View: Index.cshtml, Details.cshtml (all views under Order Views folder)
Created by	Lim Jun Peng
Date of Creation	23/4/2022
Date of Review	23/4/2022
Test Case ID	TC_UOSM_01
Test Scenario	To verify that the administrator can be able to update the order status of customer.
Test Case Description	This test case is created to test the functionality of the order dashboard of online fruits web application with action like update the order status. This test case is also prepared to ensure the output correctness and the action mentioned can be done smoothly and correctly.
Pre-condition	1. Administrator has successfully logged into the online web application.

	2. The administrator clicks the dashboard navigation and redirects to the order dashboard page to perform update on order of customers. 3. Only administrator accounts can access the dashboard. 4. Administrator only updates the status of order which is "Approved". Other status like "Cancelled", "Pending" and "Refunded" will not be able to be updated manually by the administrators.
Test Step:	1. Administrator selects the order record which needs to be updated from the dashboard and clicks the "Edit" button. 2. If the order status is "Approved", the administrator can click the "Start Processing" button. 2. If the order status is "Processing", the administrator needs to fill up the shipping information, tracking number before they can click the "Ship Order" button.
Test data	Order ID: 37 Products: Banana x2, Pineapple x2 Price: RM27.00 (Banana), RM20.00 (Pineapple) Total amount paid: RM47.00  Order Summary 2 items Banana — Price: RM13.50 — Qty: 2 — RM27.00 Pineapple — Price: RM20.00 — Qty: 1 — RM20.00 Subtotal RM47.00 Delivery Free Amount Paid RM47.00

Expected Result	1. If the order status is “Approved”, the administrator can click the “Start Processing” button, the status should change from “Approved” to “Processing”. 2. If the order status is “Processing”, after the administrator fills up the shipping information, tracking number and click the “Ship Order” button, the status should change from “Processing” to “Shipped”.
Actual Result	If the order status is “Approved”, the administrator clicks the “Start Processing” button, the status successfully changes from “Approved” to “Processing”. 2. If the order status is “Processing”, after the administrator fills up the shipping information, tracking number and click the “Ship Order” button, the status successfully changes from “Processing” to “Shipped”.
Test Result	PASSED

Screenshots of final results:

1) The administrator can click the “Start Processing” button if the order status is “Approved”.


< Back to Orders

Order Details

Pick Up Details

First Name: jp

Last Name: lim

Email: junpeng8888@1utar.my

Phone Number: 01234567889

Street Address: test

City: test

State: test

Postal Code: 12345

Order Date: 27/4/2023

Courier:

Tracking Number:

Order Summary

2 items

Banana RM27.00
Price: RM13.50
Qty: 2

Pineapple RM20.00
Price: RM20.00
Qty: 1

Subtotal	RM47.00
Delivery	Free
Amount Paid	RM47.00

Figure 6.2.6 Test result of TC_ UOSM _01

2) The order status successfully changes from “Approved” to “Processing”. The administrator fills in the shipping information, tracking number and click the “Ship Order” button.

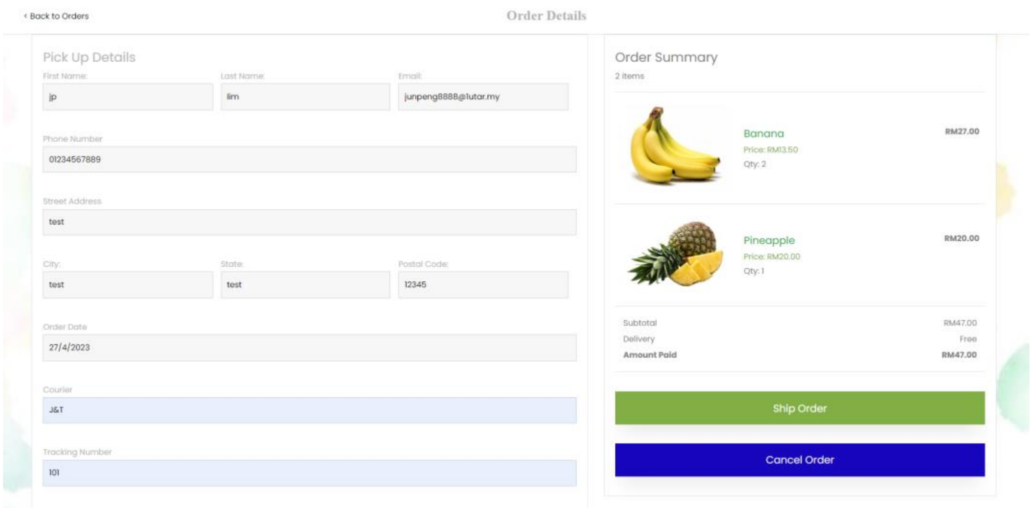


Figure 6.2.7 Test result of TC_ UOSM _01

3) The order status successfully changes from “Processing” to “Shipped”.

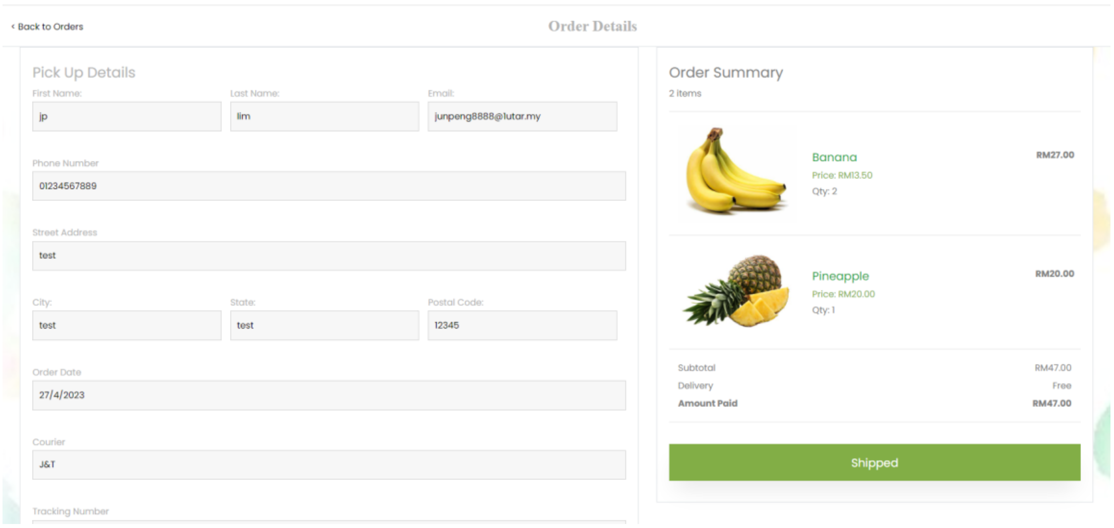


Figure 6.2.8 Test result of TC_ UOSM _01

Project Name	Modern Fruits Web Store with Personalized Recommender System
Module Name	**Generate product recommendations Module**
Reference Document	Controller: RecommenderController.cs, SingleProductController View: Index.cshtml (under Recommender Views folder), Main.cshtml, Details,cshtml (under SingleProduct Views folder)
Created by	Lim Jun Peng
Date of Creation	23/4/2022
Date of Review	23/4/2022
Test Case ID	TC_GPRM_01
Test Scenario	To verify that the proposed web application can be able to generate the product recommendations.
Test Case Description	This test case is created to test the functionality of the recommender model and the integration with online fruits web application. It is to make sure that the web application can suggest product recommendations to customers. This test case is also prepared to ensure the output correctness and the action mentioned can be done smoothly and correctly.
Pre-condition	Item-based collaborative filtering recommender

	1. Customer has registered, verified their email account and successfully logged into the online web application. 2. The customer must have rated some products before. User-based collaborative filtering recommender 1. Customer has registered, verified their email account and successfully logged into the online web application. 2. The customer must have rated some products before. Popularity-based recommender 1. Customer has registered, verified their email account and successfully logged into the online web application. 2. The customer is not necessary to rate some products before.
Test Step:	Item-based collaborative filtering recommender 1. Customer clicks "Product" on navigation and redirected the product view page. 2. Customer clicks the single product which has been rated before. 3. Customer can view and click the product being recommended under the section "Customer who rated this product may also like...". User-based collaborative filtering recommender 1. Customer clicks "Shop" on navigation and further click the list item "Recommended only for you". 2. Customer will be redirected to the "Recommended only for you" page. 3. Customer can view and click the product being recommended. Popularity-based recommender

	1. Customers scroll the home page and there will be a section called "Most Popular Products". Customers can view the products available there and check out the top-N most popular products with the highest sales. 2. Customer can view and click the product being recommended.
Test data	<u>Item-based collaborative filtering recommender</u> Product:Banana Productid=5 <u>User-based collaborative filtering recommender</u> User: junpeng8888@gmail.com ApplicationUserId ="ef48a2e2-42b2-4cf6-914d-ad1dfb5e5769"
Expected Result	<u>Item-based collaborative filtering recommender</u> Customer should be able to view and click the product being recommended under the section "Customer who rated this product may also like...". <u>User-based collaborative filtering recommender</u> Customer should be able to view and click the product being recommended in the "Recommended only for you" page. <u>Popularity-based recommender</u> Customer should be able to view and click the product being recommended in the "Most Popular Products" page.
Actual Result	<u>Item-based collaborative filtering recommender</u> Customer is able to view and click the product being recommended under the section "Customer who rated this product may also like...". <u>User-based collaborative filtering recommender</u> Customer is able to view and click the product being recommended in the "Recommended only for you" page.

	<u>Popularity-based recommender</u> Customer is able to view and click the product being recommended in the “Most Popular Products” page.
Test Result	PASSED

Screenshots of final results:

1) Item-based collaborative filtering recommender

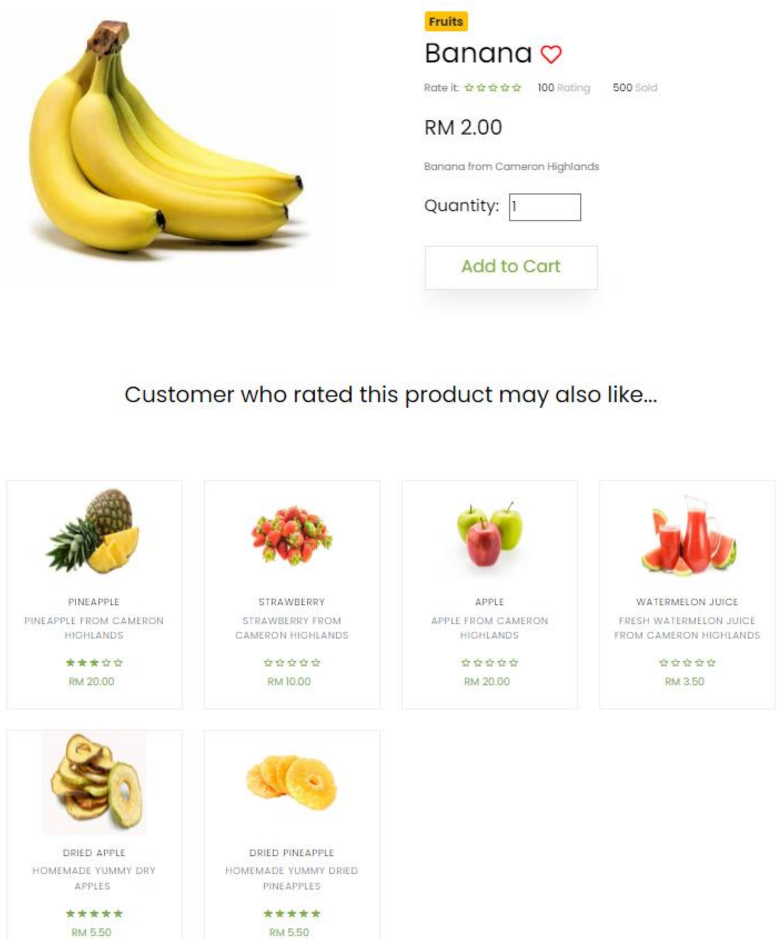
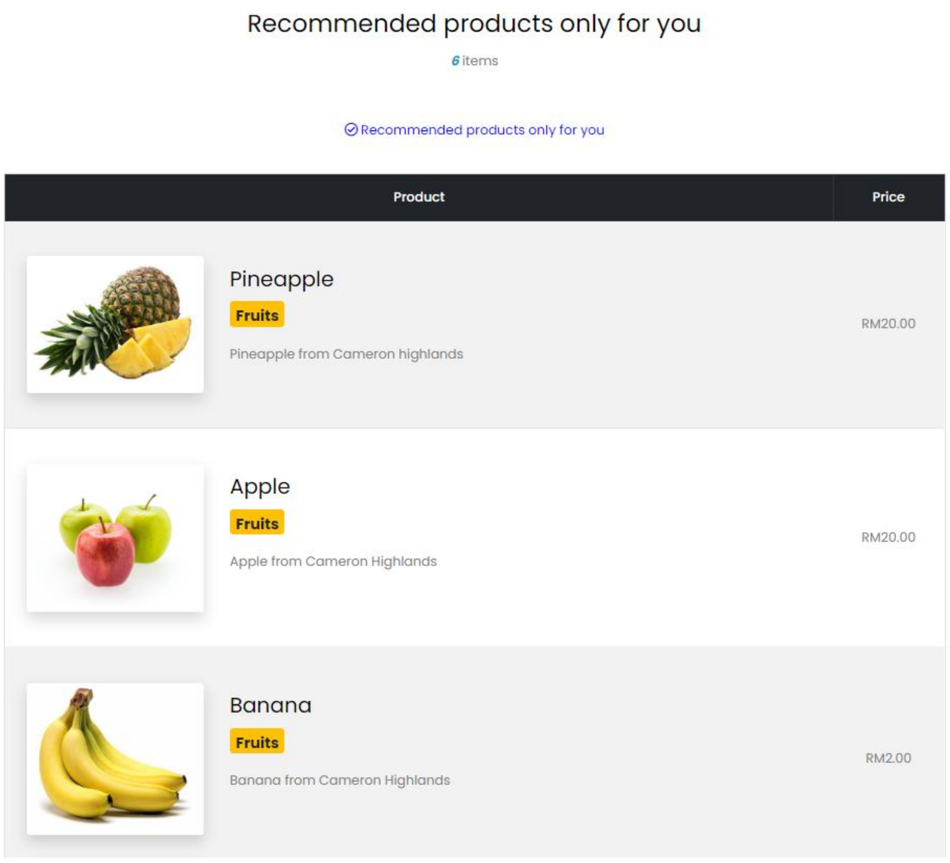

Fruits
Banana
Rate it: ☆☆☆☆☆ 100 Rating 500 Sold
RM 2.00
Banana from Cameron Highlands
Quantity: 1
Add to Cart
Customer who rated this product may also like...
PINEAPPLE
PINEAPPLE FROM CAMERON HIGHLANDS
★★★☆☆
RM 20.00
STRAWBERRY
STRAWBERRY FROM CAMERON HIGHLANDS
☆☆☆☆☆
RM 10.00
APPLE
APPLE FROM CAMERON HIGHLANDS
☆☆☆☆☆
RM 20.00
WATERMELON JUICE
FRESH WATERMELON JUICE FROM CAMERON HIGHLANDS
☆☆☆☆☆
RM 3.50
DRIED APPLE
HOMEMADE YUMMY DRY APPLES
★★★★★
RM 5.50
DRIED PINEAPPLE
HOMEMADE YUMMY DRIED PINEAPPLES
★★★★★
RM 5.50

Figure 6.2.9 Test result of TC_ GPRM _01

2) User-based collaborative filtering recommender



Recommended products only for you

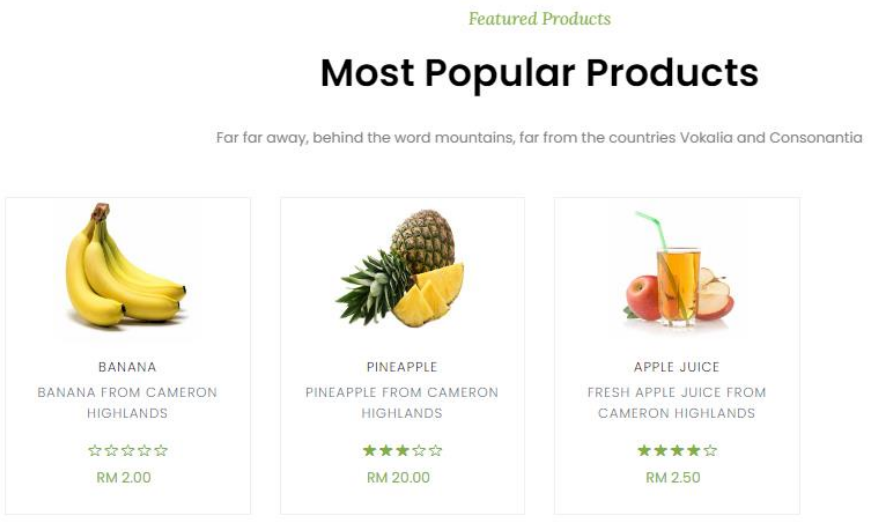
6 items

Recommended products only for you

Product		Price
	Pineapple Fruits Pineapple from Cameron highlands	RM20.00
	Apple Fruits Apple from Cameron Highlands	RM20.00
	Banana Fruits Banana from Cameron Highlands	RM2.00

Figure 6.2.10 Test result of TC_ GPRM _01

3) Popularity-based recommender



Featured Products

Most Popular Products

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia

BANANA
BANANA FROM CAMERON HIGHLANDS
☆☆☆☆☆
RM 2.00

PINEAPPLE
PINEAPPLE FROM CAMERON HIGHLANDS
★★★☆☆
RM 20.00

APPLE JUICE
FRESH APPLE JUICE FROM CAMERON HIGHLANDS
★★★★☆
RM 2.50

Figure 6.2.11 Test result of TC_ GPRM _01

6.3 Objectives Evaluation and Concluding Remarks

In this chapter, a few test cases are done to make sure all the components of web application are working smoothly. There are in total 5 test cases being tested in which 2 test cases are to make sure the customers can place order and cancel order in the web applications which are the Place order Module and Cancel order Module. These are the 2 most important operations in this proposed web application in customer module. Manage product Module and update order status Module are responsible to take care the administrator's operation especially in online selling web application. The last test case, which is the generate product recommendations Module is to test whether the recommender system can generate product recommendations to user or not on the web application. All the modules have passed the system testing, the main functionalities of the system work smoothly and the actual result able to achieve the expected result.

The first objective is to analyse different algorithms or techniques that can be used as the recommender system. Section 6.1 had discussed and analyzed the results and performance using RMSE as the performance metrics. There are also results on the products being recommended by those algorithms. Besides, there are also literature reviews on these recommender techniques before starting to build the models. It can be roughly said that the first objective is achieved so that these algorithms can be deployed and act as the recommender system that provides product recommendation to customers which is the second objective. Generate product recommendations Module test case plays the important role to verify this objective. As from the results shown, product recommendations are being generated and displayed on customers side originated from the user-based and item-based collaborative filtering recommender which deployed in the Flask. There is also a non-personalized recommender which is the popularity-based recommender applied in the web application to cool start the start-up business by simply recommending the top-N hot sales products which having the greatest number of sales in the order history. Lastly, the third objective is to develop an online fruit and fruit juice web application which should have all the required features using ASP.NET framework. From our system operation and also the first 4 test cases, it proves that the proposed web application has completed all the project scope tasks and it can be ready to be deployed in Azure and go online in future for business. After the system testing, it can be indicated that all the objectives being proposed at the beginning of the project are well-achieved. In summary, the project work can be well wrapped up after testing the proposed web application.

CHAPTER 7

Conclusion

In short, there are huge amount of preliminary work had been done in this trimester especially building up the web application prototype and the recommender system. It took me a about 13 weeks to complete all the project work. The main motivation for me to start with this project is because of the rapid growth and of e-commerce. As nowadays people prefer purchase products online, the situation encourage me to propose a modern fruit store which will empowered with a lot of expert technologies and API that can really ease people to purchase the fruits from the web application. In order to be different as compared to others, a good recommender system shall propose in the web application to personalise the customer shopping experiences and understand the customer shopping behaviour.

With the progress and work done in the last trimester, it provided me the basic knowledge which indeed strengthen my base and help me to grab the skills to create the remaining modules in next trimester. In a short summary, for the role of user, current web application allows them to register, login and logout in the website. Meanwhile, they can view all the available products and each product details in a single product page. Correspondingly, administrator is capable to manage all the products with CRUD operations. All these efforts are trying to meet the third objective which is to develop an online fruit and fruit juice web application which should have all the required features using ASP.NET framework. More and more effort has been paid off in this semester in order to make the web application has a lot of noteworthy features integrated with modern technologies and API to achieve the last objective. Meanwhile, the current web application has 2 role of users which are the customer and administrator. The associated actions which can be done by these 2 roles of user have been done based on the project scope in Chapter 1. For first and second objective, the research and development work has been completed. Comparison among the different algorithms for example like item-based and user-based collaborative-based filtering recommender in terms of their accurateness is being discovered. Other than that, the recommenders are being deployed in Flask and able to provide recommendations to customers in the web application.

In addition, there are some future work or recommendations which can further improve the proposed web application. Firsly, chatbot which acts like a middleman who can directly communicate with customers. Some Frequently Asked Questions (FAQs) and answers can also be included in the chatbot to ease customers to find that section. Next, extra learning and research are needed to provide a better recommender system in terms of the accuracy of prediction. The reason why content-based filtering is not being implemented is because of the products in the proposed web application are quite niche in terms of the categories. The performance of content-based recommender may not be really good as compared to collaborative as it only focuses on the product description and categories. However, the number and type of products are not really much which causes the content-based recommender cannot provide the product recommendations which is out of the scope of products profile. Thus, the hybrid recommenders involving both techniques integrated should be proposed in the future to improve the recommender's performance. Then, I will recommend a useful filter in the product view page so that customers can sort their products according to ratings, popularity or any other useful filters under certain time range like past one week or one month. This filter is handy to help customer to understand which product is more popular and assist them buying the best choice and suitable product. In terms of the web design and interface, more time need to be allocated to make it more eyes-catchy and interactive. The best case is that the proposed web application can display responsively among different screen sizes.

Last but not least, I strongly hope the proposed web application and solution is adequate in solving the problem at the end especially provide a platform for customer to buy fruits easily and safely without physical contact through online delivery. They can order the fruit anytime and anywhere they want through the web application. Other than that, I hope this proposed web application can also boost the trend and demand for selling and buying fruits online and delivery. Apart from that, I sincerely hope that this proposed web application can also benefit the administrators in terms of management in business. They can be easily tracking their inventory stock level and orders at once in this web application. In addition, it would also be better that the proposed solution can provide recommendations according to customer preferences and the developed feature can meet their shopping goals and provide comfortable and personalized shopping experience.

REFERENCES

[1] S. Sanyala and M. W. Hisamb, “Factors Affecting Customer Satisfaction with E-commerce Websites - An Omani Perspective,” *Proceeding 2019 Int. Conf. Digit. Landscaping Artif. Intell. ICD 2019*, pp. 232–236, 2019, doi: 10.1109/ICD47981.2019.9105780.

[2] J. N. Sheth, “Future of brick and mortar retailing: how will it survive and thrive?,” *J. Strateg. Mark.*, vol. 29, no. 7, pp. 598–607, 2021, doi: 10.1080/0965254X.2021.1891128.

[3] X. Zhao, “A study on E-commerce recommender system based on big data,” *2019 IEEE 4th Int. Conf. Cloud Comput. Big Data Anal. ICCCBDA 2019*, no. 1, pp. 222–226, 2019, doi: 10.1109/ICCCBDA.2019.8725694.

[4] S. Sivapalan, A. Sadeghian, H. Rahnama, and A. M. Madni, “Recommender systems in e-commerce,” *World Autom. Congr. Proc.*, pp. 179–184, 2014, doi: 10.1109/WAC.2014.6935763.

[5] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” *Egypt. Informatics J.*, vol. 16, no. 3, pp. 261–273, 2015, doi: 10.1016/j.eij.2015.06.005.

[6] M. Diaby, E. Viennet, and T. Launay, “Toward the next generation of recruitment tools: An online social network-based job recommender system,” *Proc. 2013 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2013*, pp. 821–828, 2013, doi: 10.1145/2492517.2500266.

[7] A. Alkhatib, A. Al Sabbagh, and R. Maraqa, “Pubic Cloud Computing: Big Three Vendors,” *2021 Int. Conf. Inf. Technol. ICIT 2021 - Proc.*, pp. 230–237, 2021, doi: 10.1109/ICIT52682.2021.9491680.

[8] B. S. Dordević, S. P. Jovanović, and V. V. Timčenko, “Cloud Computing in Amazon and Microsoft Azure platforms: Performance and service comparison,” *2014 22nd Telecommun. Forum, TELFOR 2014 - Proc. Pap.*, pp. 931–934, 2014, doi: 10.1109/TELFOR.2014.7034558.

[9] S. Rautmare and D. M. Bhalerao, “MySQL and NoSQL database comparison for IoT application,” *2016 IEEE Int. Conf. Adv. Comput. Appl. ICACA 2016*,

pp. 235–238, 2017, doi: 10.1109/ICACA.2016.7887957.

[10] D. Laksono, “Testing Spatial Data Deliverance in SQL and NoSQL Database Using NodeJS Fullstack Web App,” *Proc. - 2018 4th Int. Conf. Sci. Technol. ICST 2018*, 2018, doi: 10.1109/ICSTC.2018.8528705.

[11] A. J. Poulter, S. J. Johnston, and S. J. Cox, “Using the MEAN stack to implement a RESTful service for an Internet of Things application,” *IEEE World Forum Internet Things, WF-IoT 2015 - Proc.*, pp. 280–285, 2015, doi: 10.1109/WF-IoT.2015.7389066.

[12] Z. Chen, L. Jia, and W. Zheng, “Research and development of the long distance coach management system based on ASP.net technology,” *2012 2nd Int. Conf. Consum. Electron. Commun. Networks, CECNet 2012 - Proc.*, pp. 1992–1995, 2012, doi: 10.1109/CECNet.2012.6201721.

[13] A. Wang and H. Pan, “The teaching in ASP.NET programming and the development in E-commerce project,” *2nd Int. Work. Educ. Technol. Comput. Sci. ETCS 2010*, vol. 1, pp. 386–389, 2010, doi: 10.1109/ETCS.2010.111.

[14] “What is ASP.NET?”, *Microsoft*, 2022. [Online]. Available: https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet. [Accessed: 05- Sep- 2022].

[15] “What is ASP.NET Core?”, *Microsoft*, 2022. [Online]. Available: https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet-core. [Accessed: 05- Sep- 2022].

[16] “Understanding Mean Stack, its Advantages, Disadvantages, and Use Cases”, *Knowledgenile*, 2022. [Online]. Available: https://www.knowledgenile.com/blogs/understanding-mean-stack-applications/. [Accessed: 05- Sep- 2022].

[17] M. Baldwin, “Azure Key Vault Overview - Azure Key Vault,” *Microsoft*, 2022. [Online]. Available: https://learn.microsoft.com/en-us/azure/key-vault/general/overview [Accessed: 23- Nov- 2022].

[18] R. Boucher, “Azure Monitor Overview - Azure Monitor,” *Microsoft*, 2022. [Online]. Available: https://learn.microsoft.com/en-us/azure/azure-monitor/overview. [Accessed: 30-Nov-2022].

[19] A. Shaw, “Product recommendation system for e-commerce,” *Kaggle,* 30-Aug-2019. [Online]. Available: https://www.kaggle.com/code/shawamar/product-recommendation-system-for-e-commerce#Recommendation-System---Part-I. [Accessed: 28-Mar-2023].

[20] I. Ahmed, “Getting started with a movie recommendation system,” *Kaggle,* 16-May-2020. [Online]. Available: https://www.kaggle.com/code/ibtesama/getting-started-with-a-movie-recommendation-system/notebook#Collaborative-Filtering. [Accessed: 28-Apr-2023].

[21] S. Huang, “Introduction to recommender system. part 1 (collaborative filtering, singular value decomposition),” *HackerNoon,* 24-Jan-2018. [Online]. Available: https://hackernoon.com/introduction-to-recommender-system-part-1-collaborative-filtering-singular-value-decomposition-44c9659c5e75. [Accessed: 28-Apr-2023].

[22] H. Roggero, “Comparing AWS SQL server with Azure SQL database,” *C# Corner*, 15-Oct-2019. [Online]. Available: https://www.c-sharpcorner.com/article/comparing-aws-sql-server-with-azure-sql-database/. [Accessed: 28-Apr-2023].

[23] “ASP.NET MVC Pattern: .NET,” *Microsoft*. [Online]. Available: https://dotnet.microsoft.com/en-us/apps/aspnet/mvc. [Accessed: 28-Apr-2023].

[24] R. Anderson, S. Addie, N. Schonning, A. Pasic, and T. Dykstra, “ASP.NET MVC overview,” *Microsoft Learn*. [Online]. Available: https://learn.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview. [Accessed: 28-Apr-2023].

[25] V. Kumar, “Singular value decomposition (SVD) & its application in Recommender System,” *Analytics India Magazine*, 26-Mar-2022. [Online]. Available: https://analyticsindiamag.com/singular-value-decomposition-svd-application-recommender-system. [Accessed: 28-Apr-2023].

APPENDIX

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T2	**Study week no.: 2**
Student Name & ID: Lim Jun Peng (19ACB03930)	
Supervisor: Ts Sun Teik Heng	
Project Title: Modern Fruits Web Store with Personalized Recommender System	

1. WORK DONE
[Please write the details of the work done in the last fortnight.]

Brainstorming on project ideas and propose a plan to derive the project work needed and evaluate the progress of this semester. This is to make sure there is an achievable target amount of project development that shall be done at the end. Refine the report content as requested in last trimester.

2. WORK TO BE DONE

Finalising the required workload and output of the system for this trimester.
Program email verification and forgot password module.

3. PROBLEMS ENCOUNTERED

So far do not have any big issues. Solution should be able to find online.

4. SELF EVALUATION OF THE PROGRESS

Still manageable and able to fix the issue of configuration of email services. Able to finalise the required workload and output of the system for this trimester.

SunTeikHeng

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT
(Project II)

Trimester, Year: Y3T2	**Study week no.: 4**
Student Name & ID: Lim Jun Peng (19ACB03930)	
Supervisor: Ts Sun Teik Heng	
Project Title: Modern Fruits Web Store with Personalized Recommender System	

1. WORK DONE
[Please write the details of the work done in the last fortnight.]

Program email verification and forgot password module and customer profile done.

2. WORK TO BE DONE

Program user management CRUD module.

3. PROBLEMS ENCOUNTERED

So far also do not have any big issues. Coding examples and the online tutorials and YouTube videos are helpful for my learning curve especially those beginner tutorials with continuous learning scope that can constantly increase my knowledge topic by topic.

4. SELF EVALUATION OF THE PROGRESS

Still able to cope with the learning and development.

SunTeikHeng
Supervisor's signature

[illegible]
Student's signature

FINAL YEAR PROJECT WEEKLY REPORT
(Project II)

Trimester, Year: Y3T2	**Study week no.: 6**
Student Name & ID: Lim Jun Peng (19ACB03930)	
Supervisor: Ts Sun Teik Heng	
Project Title: Modern Fruits Web Store with Personalized Recommender System	

1. WORK DONE [Please write the details of the work done in the last fortnight.] Program user management CRUD module done.
2. WORK TO BE DONE -Program product category and shopping cart module. -Program product wish list module.
3. PROBLEMS ENCOUNTERED Need some time to build the shopping cart and wish list module.
4. SELF EVALUATION OF THE PROGRESS Everything is in progress and should be able to have a prototype next week.

SunTeikHeng
Supervisor's signature

[illegible]
Student's signature

FINAL YEAR PROJECT WEEKLY REPORT
(Project II)

Trimester, Year: Y3T2	**Study week no.: 8**
Student Name & ID: Lim Jun Peng (19ACB03930)	
Supervisor: Ts Sun Teik Heng	
Project Title: Modern Fruits Web Store with Personalized Recommender System	

1. WORK DONE
[Please write the details of the work done in the last fortnight.]

-Program product category and shopping cart module done.
-Program product wish list module done.

2. WORK TO BE DONE

-Program search engine module
-Program newsletter module
-Program make order, cancel order module

3. PROBLEMS ENCOUNTERED

Need to arrange the time better as there are other assignments and presentations ongoing. Have to optimize the time for development and learning.

4. SELF EVALUATION OF THE PROGRESS

A little bit stressful but trying to complete everything according to the timeline.

SunTeikHeng
Supervisor's signature

[illegible]
Student's signature

FINAL YEAR PROJECT WEEKLY REPORT
(Project II)

Trimester, Year: Y3T2	**Study week no.: 10**
Student Name & ID: Lim Jun Peng (19ACB03930)	
Supervisor: Ts Sun Teik Heng	
Project Title: Modern Fruits Web Store with Personalized Recommender System	

1. WORK DONE
[Please write the details of the work done in the last fortnight.]

-Program search engine module done.
-Program newsletter module done.
-Program make order, cancel order module done.

2. WORK TO BE DONE

-Program order CRUD module
-Program payment module
-Understand and familiarize with different type of recommender systems
-Compare the pros and cons of recommender systems implemented with different techniques

3. PROBLEMS ENCOUNTERED

Need to allocate more time for completing the development especially the order CRUD module and payment module using Stripe so that can move on to the recommender system later.

4. SELF EVALUATION OF THE PROGRESS

The progress is good but must allocate time on the recommender models soon.

SunTeikHeng
Supervisor's signature

Lim
Student's signature

FINAL YEAR PROJECT WEEKLY REPORT
(Project II)

Trimester, Year: Y3T2	**Study week no.: 12**
Student Name & ID: Lim Jun Peng (19ACB03930)	
Supervisor: Ts Sun Teik Heng	
Project Title: Modern Fruits Web Store with Personalized Recommender System	

1. WORK DONE
[Please write the details of the work done in the last fortnight.]

-Program order CRUD module done.
-Program payment module done.
-Understand and familiarize with different type of recommender systems done.
- Compare the pros and cons of recommender systems implemented with different techniques done.

2. WORK TO BE DONE

-Design and program the recommender system
-Deploy the recommender system into Flask and testing the overall web application
-Improve and wrap up the web application

3. PROBLEMS ENCOUNTERED

-Need time to revise back the machine learning basics and building the model
-Face some difficulty in Flask

4. SELF EVALUATION OF THE PROGRESS

-Have to allocate more time for the recommender model building, training and testing. After that, need to speed up the process and integrate it with the web application.

SunTeikHeng

Supervisor's signature

[illegible]

Student's signature

POSTER

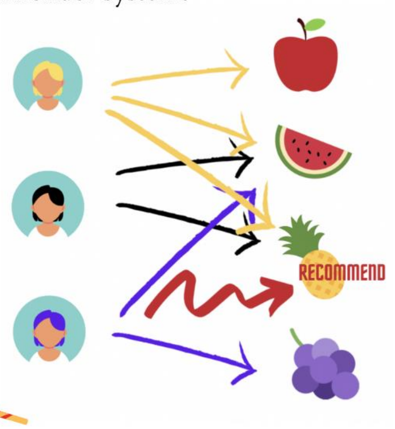
MODERN FRUITS WEB STORE WITH PERSONALIZED RECOMMENDER SYSTEM

UTAR UNIVERSITI TUNKU ABDUL RAHMAN

Faculty of Information Communication and Technology

INTRODUCTION

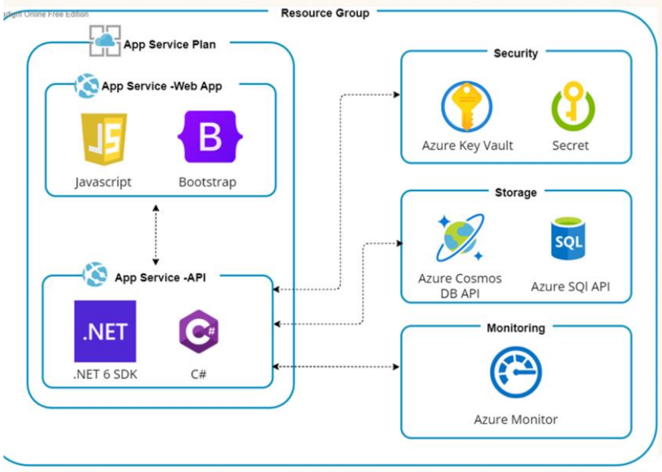
In this day and age, Ecommerce is substantially popular and not only limited to those big retail companies. So the main goal of this project is to develop a well-developed online fruit and fruit juice web application with the recommender system.



OBJECTIVES

- To analyse different algorithm that can be used as the recommender system.
- To design a recommender system that provides product recommendation to customers.
- To develop an online fruit and fruit juice web application which should have all the required features using ASP.NET framework.

METHODS



Architectural Diagram for Modern Fruit Web Store

CONCLUSION

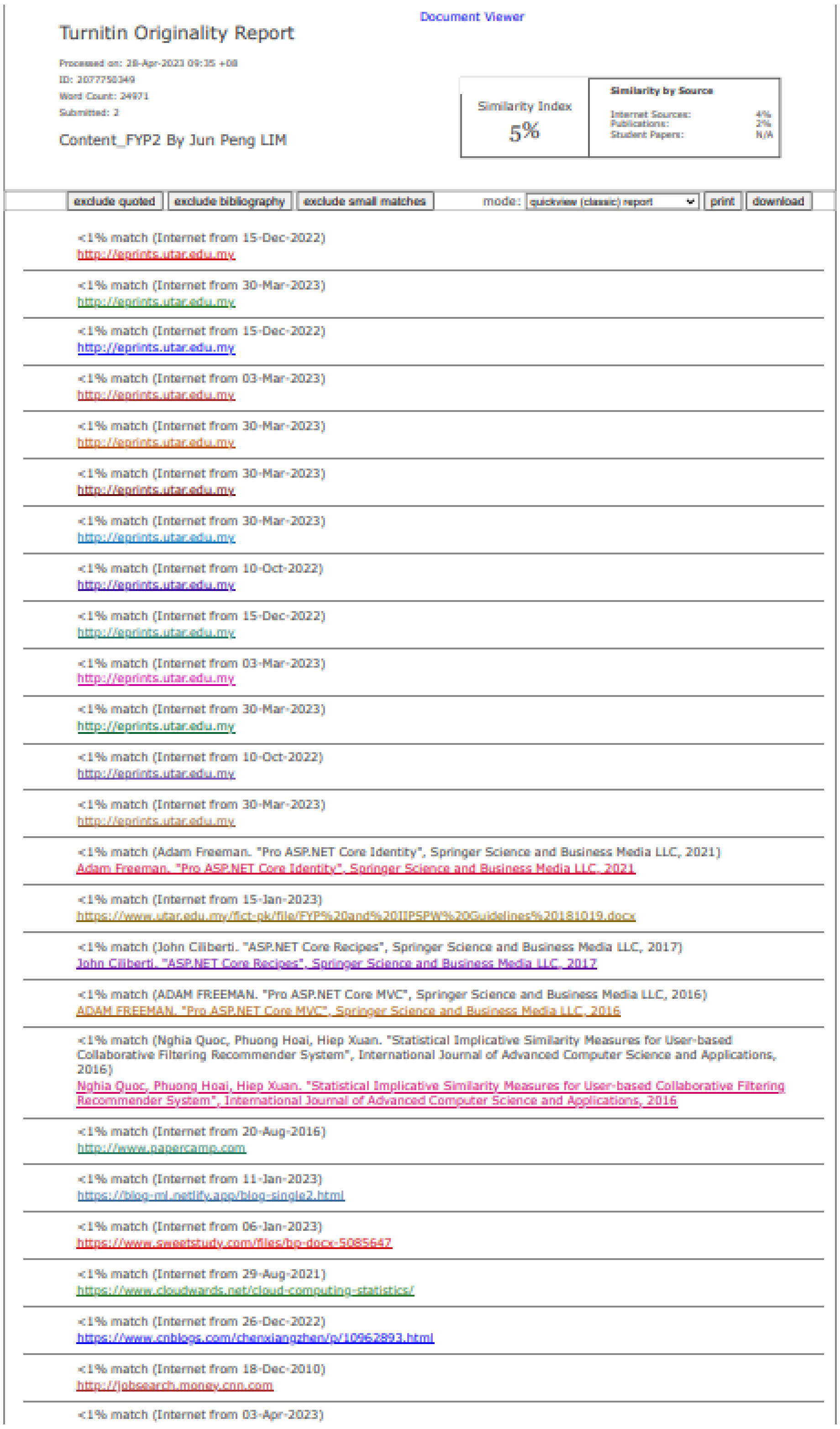
✓ Provide an online platform for customer to buy fruits easily and safely without physical contact.

✓ Provide recommendations according to customer preferences.

✓ Meet their shopping goals and provide comfortable shopping experience.

PROJECT DEVELOPER: LIM JUN PENG
PROJECT SUPERVISOR: TS SUN TEIK HENG

PLAGIARISM CHECK RESULT



Document Viewer

Turnitin Originality Report

Processed on: 28-Apr-2023 09:35 +08
ID: 2077750349
Word Count: 24971
Submitted: 2

Content_FYP2 By Jun Peng LIM

Similarity Index
5%

Similarity by Source
Internet Sources: 4%
Publications: 2%
Student Papers: N/A

<1% match (Internet from 15-Dec-2022)
http://eprints.utar.edu.my

<1% match (Internet from 30-Mar-2023)
http://eprints.utar.edu.my

<1% match (Internet from 15-Dec-2022)
http://eprints.utar.edu.my

<1% match (Internet from 03-Mar-2023)
http://eprints.utar.edu.my

<1% match (Internet from 30-Mar-2023)
http://eprints.utar.edu.my

<1% match (Internet from 30-Mar-2023)
http://eprints.utar.edu.my

<1% match (Internet from 30-Mar-2023)
http://eprints.utar.edu.my

<1% match (Internet from 10-Oct-2022)
http://eprints.utar.edu.my

<1% match (Internet from 15-Dec-2022)
http://eprints.utar.edu.my

<1% match (Internet from 03-Mar-2023)
http://eprints.utar.edu.my

<1% match (Internet from 30-Mar-2023)
http://eprints.utar.edu.my

<1% match (Internet from 10-Oct-2022)
http://eprints.utar.edu.my

<1% match (Internet from 30-Mar-2023)
http://eprints.utar.edu.my

<1% match (Adam Freeman. "Pro ASP.NET Core Identity", Springer Science and Business Media LLC, 2021)
Adam Freeman. "Pro ASP.NET Core Identity", Springer Science and Business Media LLC, 2021

<1% match (Internet from 15-Jan-2023)
https://www.utar.edu.my/fict-pk/file/FYP%20and%20IIPSPW%20Guidelines%20181019.docx

<1% match (John Ciliberti. "ASP.NET Core Recipes", Springer Science and Business Media LLC, 2017)
John Ciliberti. "ASP.NET Core Recipes", Springer Science and Business Media LLC, 2017

<1% match (ADAM FREEMAN. "Pro ASP.NET Core MVC", Springer Science and Business Media LLC, 2016)
ADAM FREEMAN. "Pro ASP.NET Core MVC", Springer Science and Business Media LLC, 2016

<1% match (Nghia Quoc, Phuong Hoai, Hiep Xuan. "Statistical Implicative Similarity Measures for User-based Collaborative Filtering Recommender System", International Journal of Advanced Computer Science and Applications, 2016)
Nghia Quoc, Phuong Hoai, Hiep Xuan. "Statistical Implicative Similarity Measures for User-based Collaborative Filtering Recommender System", International Journal of Advanced Computer Science and Applications, 2016

<1% match (Internet from 20-Aug-2016)
http://www.papercamp.com

<1% match (Internet from 11-Jan-2023)
https://blog-ml.netlify.app/blog-single2.html

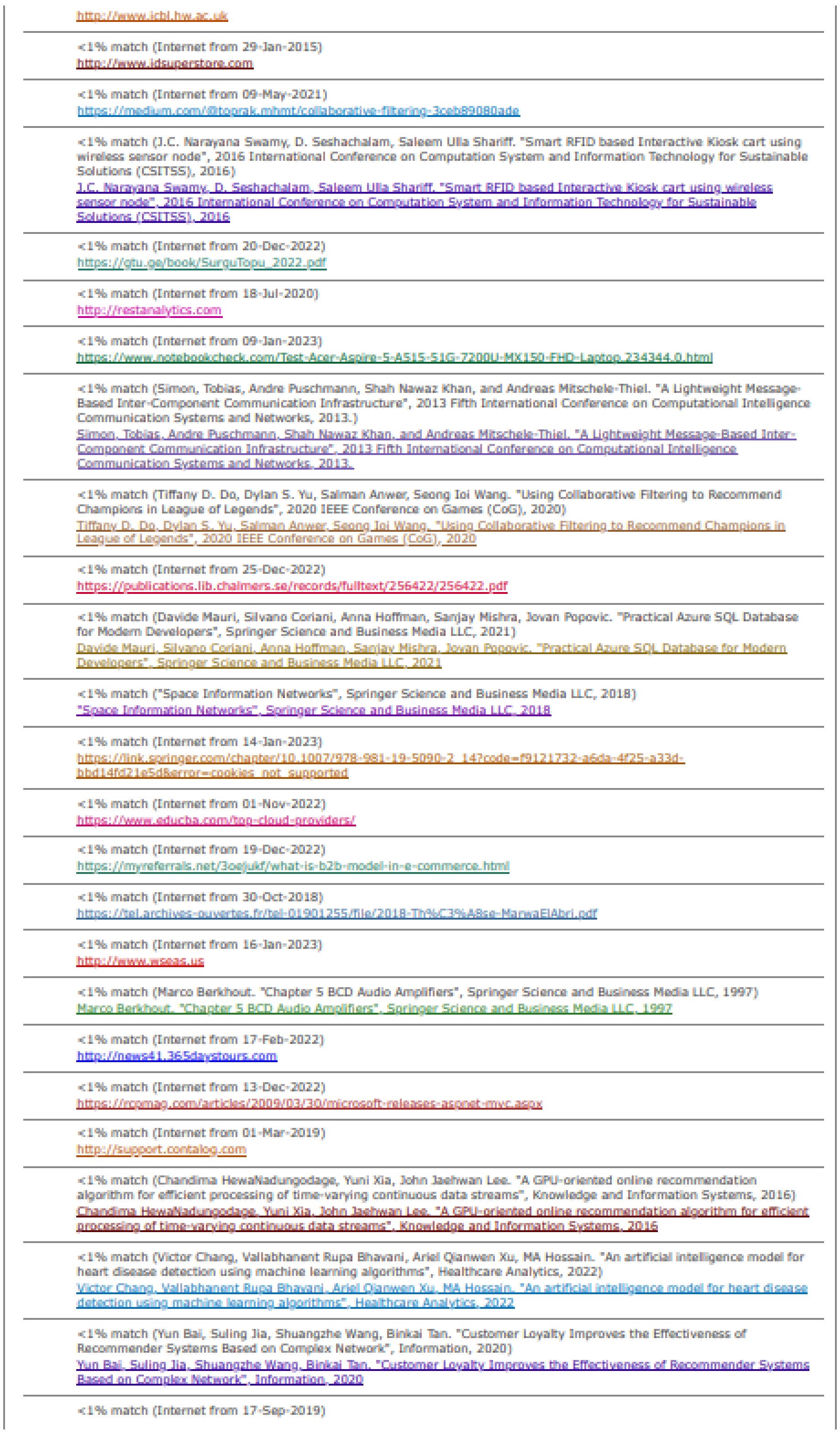
<1% match (Internet from 06-Jan-2023)
https://www.sweetstudy.com/files/bo-docx-5085647

<1% match (Internet from 29-Aug-2021)
https://www.cloudwards.net/cloud-computing-statistics/

<1% match (Internet from 26-Dec-2022)
https://www.cnblogs.com/chenxiangzhen/p/10962893.html

<1% match (Internet from 18-Dec-2010)
http://jobsearch.money.cnn.com

<1% match (Internet from 03-Apr-2023)



http://www.icbl.hw.ac.uk

<1% match (Internet from 29-Jan-2015)
http://www.idsuperstore.com

<1% match (Internet from 09-May-2021)
https://medium.com/@toprak.mhmt/collaborative-filtering-3ceb89080ade

<1% match (J.C. Narayana Swamy, D. Seshachalam, Saleem Ulla Shariff. "Smart RFID based Interactive Kiosk cart using wireless sensor node", 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), 2016)
J.C. Narayana Swamy, D. Seshachalam, Saleem Ulla Shariff. "Smart RFID based Interactive Kiosk cart using wireless sensor node", 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), 2016

<1% match (Internet from 20-Dec-2022)
https://gtu.ge/book/SurguTopu_2022.pdf

<1% match (Internet from 18-Jul-2020)
http://restanalytics.com

<1% match (Internet from 09-Jan-2023)
https://www.notebookcheck.com/Test-Acer-Aspire-5-A515-51G-7200U-MX150-FHD-Laptop.234344.0.html

<1% match (Simon, Tobias, Andre Puschmann, Shah Nawaz Khan, and Andreas Mitschele-Thiel. "A Lightweight Message-Based Inter-Component Communication Infrastructure", 2013 Fifth International Conference on Computational Intelligence Communication Systems and Networks, 2013.)
Simon, Tobias, Andre Puschmann, Shah Nawaz Khan, and Andreas Mitschele-Thiel. "A Lightweight Message-Based Inter-Component Communication Infrastructure", 2013 Fifth International Conference on Computational Intelligence Communication Systems and Networks, 2013.

<1% match (Tiffany D. Do, Dylan S. Yu, Salman Anwer, Seong Ioi Wang. "Using Collaborative Filtering to Recommend Champions in League of Legends", 2020 IEEE Conference on Games (CoG), 2020)
Tiffany D. Do, Dylan S. Yu, Salman Anwer, Seong Ioi Wang. "Using Collaborative Filtering to Recommend Champions in League of Legends", 2020 IEEE Conference on Games (CoG), 2020

<1% match (Internet from 25-Dec-2022)
https://publications.lib.chalmers.se/records/fulltext/256422/256422.pdf

<1% match (Davide Mauri, Silvano Coriani, Anna Hoffman, Sanjay Mishra, Jovan Popovic. "Practical Azure SQL Database for Modern Developers", Springer Science and Business Media LLC, 2021)
Davide Mauri, Silvano Coriani, Anna Hoffman, Sanjay Mishra, Jovan Popovic. "Practical Azure SQL Database for Modern Developers", Springer Science and Business Media LLC, 2021

<1% match ("Space Information Networks", Springer Science and Business Media LLC, 2018)
"Space Information Networks", Springer Science and Business Media LLC, 2018

<1% match (Internet from 14-Jan-2023)
https://link.springer.com/chapter/10.1007/978-981-19-5090-2_14?code=f9121732-a6da-4f25-a33d-bbd14fd21e5d&error=cookies_not_supported

<1% match (Internet from 01-Nov-2022)
https://www.educba.com/top-cloud-providers/

<1% match (Internet from 19-Dec-2022)
https://myreferrals.net/3oejukf/what-is-b2b-model-in-e-commerce.html

<1% match (Internet from 30-Oct-2018)
https://tel.archives-ouvertes.fr/tel-01901255/file/2018-Th%C3%A8se-MarwaElAbri.pdf

<1% match (Internet from 16-Jan-2023)
http://www.wseas.us

<1% match (Marco Berkhout. "Chapter 5 BCD Audio Amplifiers", Springer Science and Business Media LLC, 1997)
Marco Berkhout. "Chapter 5 BCD Audio Amplifiers", Springer Science and Business Media LLC, 1997

<1% match (Internet from 17-Feb-2022)
http://news41.365daystours.com

<1% match (Internet from 13-Dec-2022)
https://rcpmag.com/articles/2009/03/30/microsoft-releases-aspnet-mvc.aspx

<1% match (Internet from 01-Mar-2019)
http://support.contalog.com

<1% match (Chandima HewaNadungodage, Yuni Xia, John Jaehwan Lee. "A GPU-oriented online recommendation algorithm for efficient processing of time-varying continuous data streams", Knowledge and Information Systems, 2016)
Chandima HewaNadungodage, Yuni Xia, John Jaehwan Lee. "A GPU-oriented online recommendation algorithm for efficient processing of time-varying continuous data streams", Knowledge and Information Systems, 2016

<1% match (Victor Chang, Vallabhanent Rupa Bhavani, Ariel Qianwen Xu, MA Hossain. "An artificial intelligence model for heart disease detection using machine learning algorithms", Healthcare Analytics, 2022)
Victor Chang, Vallabhanent Rupa Bhavani, Ariel Qianwen Xu, MA Hossain. "An artificial intelligence model for heart disease detection using machine learning algorithms", Healthcare Analytics, 2022

<1% match (Yun Bai, Suling Jia, Shuangzhe Wang, Binkai Tan. "Customer Loyalty Improves the Effectiveness of Recommender Systems Based on Complex Network", Information, 2020)
Yun Bai, Suling Jia, Shuangzhe Wang, Binkai Tan. "Customer Loyalty Improves the Effectiveness of Recommender Systems Based on Complex Network", Information, 2020

<1% match (Internet from 17-Sep-2019)

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-security-tutorial

<1% match (Internet from 07-Dec-2022)
https://ipfs.io/ipfs/bafykbzaceaq5zrvlr5upvsjusvd566mq5rekwjea4ca6qzigjzluu2umtjcg?filename=%5BLecture+Notes+in+Computer+Science+%E2%84%964496%5D+Paul+Davidsson%2C+Jan+A.+Persson%2C+Johan+Holmg+Agent+and+Multi-Agent+Systems%3A+Technologies+and+Applications%3A+First+KES+International+Symposium%2C+KES-AMSTA+2007%2C+Wroclaw%2C+Poland%2C+May+31%E2%80%93+June+1%2C+2007.+Proceedings+%282007%2C+Springer%29+%!3-540-72830-6%5D.pdf

<1% match (Internet from 08-Sep-2021)
http://www.fikt.uklo.edu.mk

<1% match (Internet from 13-Apr-2023)
http://www.ir.juit.ac.in:8080

<1% match ("Emerging Research in Computing, Information, Communication and Applications", Springer Science and Business Media LLC, 2019)
"Emerging Research in Computing, Information, Communication and Applications", Springer Science and Business Media LLC, 2019

<1% match (Manolis Vozalis. "On the combination of user-based and item-based collaborative filtering", International Journal of Computer Mathematics, 9/1/2004)
Manolis Vozalis. "On the combination of user-based and item-based collaborative filtering", International Journal of Computer Mathematics, 9/1/2004

<1% match (Rajesh P.N. Rao. "An optimal estimation approach to visual perception and learning", Vision Research, 1999)
Rajesh P.N. Rao. "An optimal estimation approach to visual perception and learning", Vision Research, 1999

<1% match (Wu Jin. "User interest modeling and collaborative filtering algorithms application in English personalized learning resource recommendation", Research Square Platform LLC, 2023)
Wu Jin. "User interest modeling and collaborative filtering algorithms application in English personalized learning resource recommendation", Research Square Platform LLC, 2023

<1% match (Internet from 05-Nov-2022)
https://coek.info/pdf-big-sensor-data-applications-in-urban-environments-.html

<1% match (Internet from 21-Nov-2022)
https://cpdeportfolio.rcog.org.uk/cloud-software.html

<1% match (Internet from 18-Jan-2023)
https://documents1.worldbank.org/curated/ru/434831469792299796/SFG2362-REVISED-EA-P153473-Box402881B-PUBLIC-Disclosed-1-25-2017.docx

<1% match ()
"Recent Advances in Social Data and Artificial Intelligence 2019", 'MDPI AG', 2022

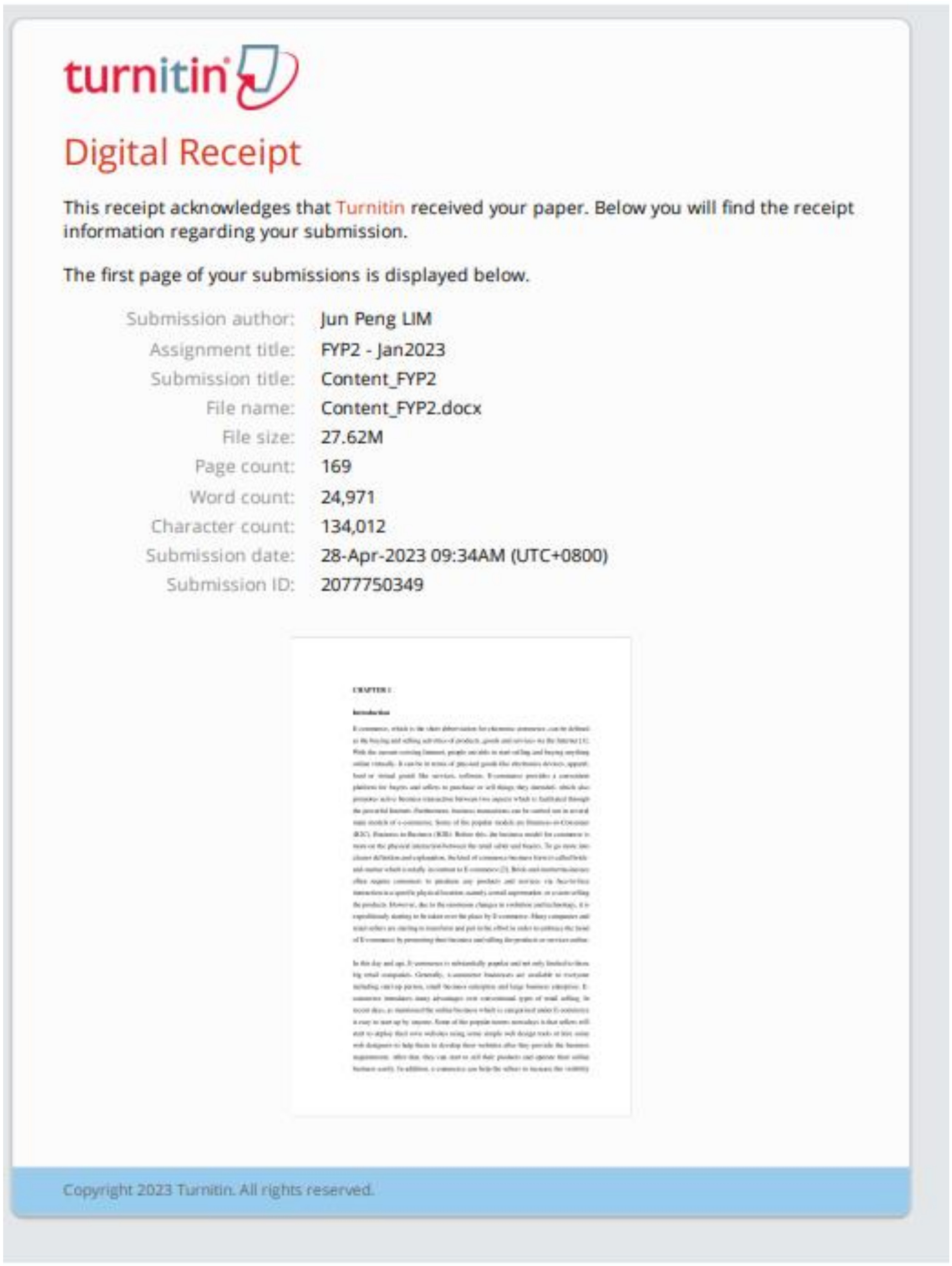
<1% match (Internet from 17-Oct-2021)
http://repository.aust.edu.ng

<1% match (Internet from 07-Apr-2021)
https://www.coursehero.com/file/50958398/web-serverpy/

<1% match (Internet from 31-May-2020)
https://www.seedworkshops.com/managing-data-cloud-using-microsoft-azure/

<1% match (Internet from 14-Jan-2023)
https://www.theses.fr/2014CERG0683.pdf

<1% match (Ying Bai. "Practical Database Programming with Java", Wiley, 2011)
Ying Bai. "Practical Database Programming with Java", Wiley, 2011



turnitin

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Jun Peng LIM
Assignment title:	FYP2 - Jan2023
Submission title:	Content_FYP2
File name:	Content_FYP2.docx
File size:	27.62M
Page count:	169
Word count:	24,971
Character count:	134,012
Submission date:	28-Apr-2023 09:34AM (UTC+0800)
Submission ID:	2077750349

Copyright 2023 Turnitin. All rights reserved.

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date:	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	LIM JUN PENG
ID Number(s)	19ACB03930
Programme / Course	Bachelor Degree of Computer Science (CS)
Title of Final Year Project	Modern Fruits Web Store with Personalized Recommender System

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: __5__ **%** **Similarity by source** Internet Sources: __4__ % Publications: __2__ % Student Papers: __N/A__ %	
Number of individual sources listed of more than 3% similarity: _0____	
Parameters of originality required, and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8***	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

SunTeikHeng	
Signature of Supervisor	Signature of Co-Supervisor
Name: __Ts. Sun Teik Heng__	Name: __________
Date: __28/4/2023	Date: __________

FYP 2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student ID	19ACB03930
Student Name	Lim Jun Peng
Supervisor Name	Ts Sun Teik Heng

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

[illegible]

(Signature of Student)
Date: 28/4/2023