DEVELOPMENT OF CURRENCY EXCHANGE RATE TRACKING SYSTEM

PHAN ZHAN YAN

UNIVERSITI TUNKU ABDUL RAHMAN

DEVELOPMENT OF CURRENCY EXCHANGE RATE TRACKING SYSTEM

PHAN ZHAN YAN

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Software Engineering with Honours

Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2024

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 

Name : Phan Zhan Yan

ID No. : 20UEB01584

Date : 25/4/2024

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"DEVELOPMENT OF CURRENCY EXCHANGE RATE TRACKING SYSTEM"** was prepared by **PHAN ZHAN YAN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Software Engineering with Honours at Universiti Tunku Abdul Rahman.

Approved by,

Signature : 

Supervisor : MOHAMMAD BABRDEL BOANB

Date : 16/05/2024

Signature :

Co-Supervisor :

Date :

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Mohammad Babrdel Bonab for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents and friends who had helped and given me encouragement.....

ABSTRACT

Currency exchange rate tracking systems are responsible to provide visualization for price movements of multiple pairs of currencies. In this day and age of rapid globalization, tracking currency exchange rates has become an integral part of most people's daily routines. However, there are some potential issues with existing currency exchange rate tracking applications in the market which are complex interfaces which are non-beginner friendly, insufficient data visualization, and lack flexibility in creating the exchange rate notification alert. To address all the issues mentioned above, the currency exchange rate tracking system developed in this system uses a dashboard design to summarize the currency exchange rate performance for users to obtain as many insights as possible at first glance. Besides that, the system is embedded with an alert management screen for users to perform CRUD actions on the alerts. Lastly, the system uses various data visualization charts like line charts, bar charts, and heatmaps to convey information to the users. The proposed system is developed using prototyping methodology with Flask being the backend server. After the development stage, unit testing, integration testing, user acceptance testing, and usability testing are conducted to evaluate the functionality and usability of the output system the test results are documented in the report.

TABLE OF CONTENTS

DECLARATION **i**
APPROVAL FOR SUBMISSION **ii**
ACKNOWLEDGEMENTS **iv**
ABSTRACT **v**
TABLE OF CONTENTS **vi**
LIST OF FIGURES **x**
LIST OF TABLES **xvi**

CHAPTER

1 INTRODUCTION **17**
1.1 General Introduction 17
1.2 Project Background 17
1.3 Problem Background 18
1.4 Problem Statement 19
1.4.1 Complex User Interface 19
1.4.2 Lacking Flexibility for Personalized Notifications 20
1.4.3 Insufficient Data Visualization 20
1.5 Project Objectives 21
1.6 Proposed Solution 21
1.7 Proposed Approach 23
1.8 Project Scope 24
1.8.1 Login Module 24
1.8.2 Data Visualization Module 24
1.8.3 Notification Module 25
2 LITERATURE REVIEW **26**
2.1 Introduction 26

	2.2	Review of Existing Currency Exchange Rate Tracking Application	26
		2.2.1 Existing Features in XE.com	26
		2.2.2 Strength and Weakness of XE.com	27
		2.2.3 Existing Features in Wise	27
		2.2.4 Strength and Weakness of Wise	27
		2.2.5 Existing Features in AvaTrade	28
		2.2.6 Strength and Weakness of AvaTrade	28
		2.2.7 Comparison of Applications by Features	29
		2.2.8 Comparison of Applications by Strengths and Weaknesses	30
	2.3	Review on Backend Framework	30
		2.3.1 Flask	30
		2.3.2 Laravel	31
		2.3.3 Django	31
		2.3.4 Comparison Between Backend Frameworks	31
		2.3.5 Summary on Backend Framework	32
	2.4	Review on Software Development Model	32
		2.4.1 Prototyping	32
		2.4.2 Waterfall	33
		2.4.3 Iterative	33
		2.4.4 Comparison Between Development Model	34
		2.4.5 Summary on Development Model	34
3	**METHODOLOGY AND WORK PLAN**		**35**
	3.1	Introduction	35
	3.2	Project Development Methodology	35
		3.2.1 Phase 1: Planning	35
		3.2.2 Phase 2: Requirement Analysis	36
		3.2.3 Phase 3: Design	36
		3.2.4 Phase 4: Build Prototype & Prototype Evaluation	36
		3.2.5 Phase 5: Implementation	37
		3.2.6 Phase 6: Testing	37

	3.3	Project Work Plan	37
4	**PROJECT SPECIFICATION & DESIGN**		**38**
	4.1	Introduction	38
	4.2	Requirement Specification	38
		4.2.1 Functional Requirements	38
		4.2.2 Non-Functional Requirements	40
	4.3	Design & Modelling	40
		4.3.1 Use Case Diagram	41
		4.3.2 Use Case Description	42
		4.3.3 System Architecture Design	64
		4.3.4 Entity Relationship Diagram (ERD)	65
		4.3.5 User Interface Design	67
5	**IMPLEMENTATION**		**72**
	5.1	Introduction	72
	5.2	User Authentication Module	72
		5.2.1 Login Account	72
		5.2.2 Sign Up Account	74
		5.2.3 Reset Password	77
		5.2.4 Google Authentication	79
	5.3	User Profile Module	82
		5.3.1 Edit Profile	82
		5.3.2 Add / Change Password	84
	5.4	Currency Tracking Module	88
		5.4.1 Currency Dashboard	88
		5.4.2 Currency Converter	98
		5.4.3 Currency Exchange Rate Tracking	100
		5.4.4 Currency Relative Strength Index (RSI) Tracking	102
		5.4.5 Currency Changes Comparison	105
		5.4.6 Currency Changes Correlation Analysis	110
		5.4.7 Extra Chart Features	113
	5.5	Alert Module	114
		5.5.1 Exchange Rate Alert	114
		5.5.2 RSI Value Alert	116

		5.5.3 Alert Setting	118
		5.5.4 Scheduler	121
	5.6	Notification Module	122
		5.6.1 Email Notifications	122
		5.6.2 App Notifications	123
	5.7	Project Structure & Project Configurations	124
	5.8	Data Migration	128
	5.9	Data Seeding	128
6	**TESTING**		**129**
	6.1	Introduction	129
	6.2	Unit Testing	130
	6.3	Integration Testing	154
	6.4	User Acceptance Testing	157
	6.5	Usability Testing	161
7	**CONCLUSION AND RECOMMENDATIONS**		**167**
	7.1	Conclusion	167
	7.2	Objective Fulfilment	167
	7.3	Limitations	168
	7.4	Recommendations for Future Work	169
REFERENCES			**171**
APPENDICES			**173**

LIST OF FIGURES

Figure 1.6.1: Overview of CRUD Operations for Notification 22

Figure 1.6.2: High-Level Architectural Design of Application (William, 2022) 22

Figure 1.7.1: Prototyping Workflow 23

Figure 3.2.1 Project Prototyping Methodology Overview 35

Figure 4.3.1: Use Case Diagram 41

Figure 4.3.2: System Architecture Design 64

Figure 4.3.3: Entity Relationship Diagram 65

Figure 4.3.4: Login Screen 67

Figure 4.3.5: Sign Up Screen 67

Figure 4.3.6: Dashboard Screen 68

Figure 4.3.7: Converter Screen 68

Figure 4.3.8: Exchange Rate Changes Screen 69

Figure 4.3.9: RSI Value Screen 69

Figure 4.3.10: Comparison Screen 70

Figure 4.3.11: Currency Correlation Screen 70

Figure 4.3.12: Notification Setting Screen 71

Figure 5.2.1: Login Page 72

Figure 5.2.2: Login Page with Frontend Validation 73

Figure 5.2.3: Login Page with Backend Validation 73

Figure 5.2.4: Code for Frontend Login Validation 73

Figure 5.2.5: Sign Up Page 74

Figure 5.2.6: Sign Up Page with Frontend Validation 74

Figure 5.2.7: Sign Up Page with Backend Validation 75

Figure 5.2.8: Code for Frontend Sign Up Validation 75

Figure 5.2.9: Account Verification Email from System 75

Figure 5.2.10: Code for Generating Activation Token in Sign Up Function 76

Figure 5.2.11: Code for Verifying Activation Token 76

Figure 5.2.12: Forget Password Page 77

Figure 5.2.13: Password Reset Email from System 77

Figure 5.2.14: Reset Password Page 78

Figure 5.2.15: Reset Password Page with Frontend Validation 78

Figure 5.2.16: Code for Frontend Validation on Resetting Password 78

Figure 5.2.17: Success Message upon Password Reset 79

Figure 5.2.18: Google Sign In Button on Login Page 79

Figure 5.2.19: Google Authentication Page 80

Figure 5.2.20: Code for Google OAuth Initiation 80

Figure 5.2.21: Code for Google OAuth Callback Handling 80

Figure 5.2.22: Add Profile Page 81

Figure 5.2.23: Validation on Adding Profile 81

Figure 5.3.1: Edit Profile Button on Home Page 82

Figure 5.3.2: Edit Profile Page 83

Figure 5.3.3: Frontend Validation of Edit Profile Page 83

Figure 5.3.4: Code for Frontend Validation on Edit Profile Page 83

Figure 5.3.5: Success Message upon Saving Profile Changes 84

Figure 5.3.6: Add Password Button on Home Page 84

Figure 5.3.7: Change Password Page for Google OAuth Users 85

Figure 5.3.8: Change Password Button on Home Page 85

Figure 5.3.9: Change Password Page for User Accounts with Password 86

Figure 5.3.10: Validation on Change Password Page 86

Figure 5.3.11: Code for Frontend Validation on Change Password Page 87

Figure 5.3.12: Code for Backend Validation on Change Password Page 87

Figure 5.3.13: Success Message upon Changing Passwords 87

Figure 5.4.1: Currency Dashboard Page 88

Figure 5.4.2: Currency Pair Overview Section on Dashboard Page 89

Figure 5.4.3: Currency Selection on Dashboard Page 89

Figure 5.4.4: Code for Currencies Input Customization 90

Figure 5.4.5: Code for Applying Select2 Package on Currencies Input 90

Figure 5.4.6: Line Chart on Dashboard Page 91

Figure 5.4.7: Code for Line Chart Customization 91

Figure 5.4.8: Code for Line Chart Initialization with ApexCharts.js 92

Figure 5.4.9: Popular Currencies Section on Dashboard Page 92

Figure 5.4.10: Favourite Currencies Section on Dashboard Page 92

Figure 5.4.11: Bar Chart on Dashboard Page 93

Figure 5.4.12: Code for Bar Charts Initialization with ApexCharts.js 93

Figure 5.4.13: Code for Bar Chart Customization 94

Figure 5.4.14: Code for Retrieving Latest Exchange Rate through Ajax 95

Figure 5.4.15: Code for Retrieving Historical Exchange Rates through Ajax 95

Figure 5.4.16: Code for Retrieving Historical RSI Values through Ajax 96

Figure 5.4.17: Code for Getting Popular Currencies Comparison through Ajax 97

Figure 5.4.18: Code for Getting Favourite Currencies Comparison through Ajax 97

Figure 5.4.19: Currency Converter Page 98

Figure 5.4.20: Result Displayed in Currency Converter Page 98

Figure 5.4.21: Code for Retrieving Conversion Result through Ajax 99

Figure 5.4.22: Code for Conversion API Endpoint 99

Figure 5.4.23: Currency Exchange Rate Changes Page 100

Figure 5.4.24: Code for Getting Historical Exchange Rates through Ajax 100

Figure 5.4.25: Code for Historical Exchange Rates API Endpoint 101

Figure 5.4.26: Update after Changes on Duration 101

Figure 5.4.27: Code for Detecting Changes of User Input 102

Figure 5.4.28: Currency RSI Values Changes Page 102

Figure 5.4.29: Code for Getting Historical RSI Values through Ajax 103

Figure 5.4.30: Code for Historical RSI Values API Endpoint 103

Figure 5.4.31: Formula of RSI Value Calculation (West, 2023) 104

Figure 5.4.32: Code for Performing RSI Value Calculation 104

Figure 5.4.33: Update after Changes on Duration 105

Figure 5.4.34: Currencies Comparison Page 105

Figure 5.4.35: Code for Getting Currencies Comparison through Ajax 106

Figure 5.4.36: Code for Currencies Comparison API Endpoint 107

Figure 5.4.37: Comparison by Favourite Currencies 107

Figure 5.4.38: Code for Enabling Comparison by Favourite / Popular Currencies 108

Figure 5.4.39: Adding Extra Currency Input 108

Figure 5.4.40: Code for Adding Extra Currency Input 109

Figure 5.4.41: Code for Removing Currency Input 109

Figure 5.4.42: Code for Detecting Changes on Currencies Input & Duration 109

Figure 5.4.43: Currency Correlation Analysis Page 110

Figure 5.4.44: Code for Getting Currencies Correlation through Ajax 110

Figure 5.4.45: Code for Currencies Correlation Analysis API Endpoint 111

Figure 5.4.46: Heatmap Chart Configuration 112

Figure 5.4.47: Currency Correlation Analysis by Favourite Currencies 112

Figure 5.4.48: Zoom In & Zoom Out Buttons for Line Charts 113

Figure 5.4.49: Download Feature 113

Figure 5.5.1: Setting Periodic Alerts for Exchange Rate 114

Figure 5.5.2: Setting Conditional Alerts for Exchange Rate 114

Figure 5.5.3: Success Message on Setting Alert 115

Figure 5.5.4: Error Message on Setting Alert 115

Figure 5.5.5: Code for Creating Alerts through Ajax 116

Figure 5.5.6: Setting Periodic Alerts for RSI Value 116

Figure 5.5.7: Setting Conditional Alerts as Overbought 117

Figure 5.5.8: Setting Conditional Alerts as Oversold 117

Figure 5.5.9: Alerts Setting Page 118

Figure 5.5.10: Code for Toggling Alerts through Ajax 118

Figure 5.5.11: Code for Deleting Alerts through Ajax 119

Figure 5.5.12: Code for Editing Notes of Alert through Ajax 119

Figure 5.5.13: Edit Modal for Conditional Alert 119

Figure 5.5.14: Edit Modal for Periodic Alert 120

Figure 5.5.15: Code for Validation & Updating Alert through Ajax 120

Figure 5.5.16: Scheduler Configurations 121

Figure 5.5.17: Scheduler Tasks Running on Celery 121

Figure 5.6.1: Email for Periodic Notification 122

Figure 5.6.2: Email for Conditional Notification 122

Figure 5.6.3: View In-App Notifications 123

Figure 5.6.4: Code for Fetching Notifications through Ajax 123

Figure 5.7.1: Structure of Base Directory 124

Figure 5.7.2: Structure of Alert Module 124

Figure 5.7.3: Structure of Auth Module 125

Figure 5.7.4: Structure of Currency Module 125

Figure 5.7.5: Structure of Notification Module 125

Figure 5.7.6: Config.py File 126

Figure 5.7.7: Flask Application Initialization Code 127

Figure 5.8.1: Migration Files 128

Figure 5.9.1: Seeding Files 128

LIST OF TABLES

Table 2-1: Comparison of existing currency exchange rate tracking applications by features 29

Table 2-2: Comparison of existing currency exchange rate tracking system by strengths and weaknesses 30

Table 2-3: Comparison between Flask, Laravel and Django 32

Table 2-4: Comparison between Prototyping, Waterfall and Iterative Model 34

Table 4-1: Data Dictionary 65

Table 6-1: Use Case Traceability Matrix 129

Table 6-2: Unit Testing Test Cases 130

Table 6-3: Integration Testing Test Cases 154

Table 6-4: User Acceptance Test (UAT) Test Cases 157

Table 6-5: Usability Test Scenarios 161

Table 6-6: Participants Demographic Summary 164

Table 6-7: Results of Users Satisfaction Survey 165

Table 6-8: Summary of Participant Comments 166

CHAPTER 1

INTRODUCTION

1.1 General Introduction

The currency exchange rate which is also commonly known as the foreign exchange rate as well as the forex rate, shows how valuable one currency is in relation to other currencies. In this day and age of rapid globalization and interconnected economies, tracking real-time currency exchange rates has become an integral part of most people's daily routines. People need this information to achieve various purposes, ranging from company executives who can make decisions for the future direction of their company's development, to individuals planning for their personal finance savings. Besides that, currency exchange rates are exclusively critical in the realm of international trade, cross-country investment, and financial activities.

Along with the rise of the globalize economy, the currency exchange rate tracking system is designed to provide monitoring and analysis for movements in the prices of currencies for multiple countries. Thus, the expected output of this project is a functional currency exchange rate tracking web application which able to collect historical currency exchange rate data from a reliable source and virtualize the trend of the exchange rate using a variety of methods to provide as much insight as possible.

1.2 Project Background

Currency exchange rate tracking system can basically be generalized by three varieties of systems in the existing market which are currency converting system, money transferring system and forex trading system. Currency converting system covers the most basic functionalities of currency exchange rate tracking system which allows users to check conversion rate of currencies and visualize historical conversion rate data using line charts. Some examples of this kind of application are Currency Converter Plus, All Currency Converter, MyCurrency, and so on.

Moving on, money transferring system and forex trading system can be said as an evolution of currency converting system. They not only provide basic functionalities of currency converting system as their sub-modules, furthermore, they also have additional features as their main module. The main purpose of money transferring applications is to allow people to complete transactions internationally while forex trading applications is to provide various forex indicators at the same time permit users to trade their money into different currencies. Xe and Wise are two instances of money transferring applications that are leading the market. Meanwhile, Oanda, IG Forex, and Avatrade are some examples of forex trading system.

In this project, the currency exchange rate tracking system being developed will be mainly focus on notifying currency conversion rates, visualizing historical currency exchange rates and providing a range of forex indicators to attract potential users which are forex traders. The system will also additionally grant users the ability to generate and manage customized alerts as sub-module. However, this project will not cover extra functionalities like sending money across countries and trading currencies.

1.3 Problem Background

An ideal currency exchange rate tracking system should be a cutting-edge tool that utilizes advanced technologies to accommodate users' various needs in the ever-changing landscape of worldwide finance. Such system should be prioritizing real-time data accuracy, comprehensive historical data visualization, and flexibility in creating notifications at the same time ensuring the user experience.

However, most if not all existing currency exchange rate tracking systems or similar systems fall short of achieving all of the necessary criteria mentioned above to be qualified as an ideal system. Some are having a complex user interface and overloaded functionalities which will confuse the user. There are also websites having poor visualization on the data which will cause users to misinterpret and make the wrong decisions. Besides that, the vast majority of current systems do not support features of creating customized conditions to send alerts or notifications. The users will have no choice but to glare at the

screen of the application periodically in order to get acknowledged by the real-time data. In this case, there is a need for a currency exchange rate tracking system that can provide real-time currency exchange rate data, comprehensive historical data visualization, and the ability to create customizable notifications.

1.4 Problem Statement

There are mainly three problems identified in the existing currency exchange rate tracking system and similar systems in the market based on observation on their web applications.

The issues identified are as follows:

1.4.1 Complex User Interface

The first issue is the complexity of the graphical user interface of most of the present applications. The root cause of this issue is due to large amounts of the present system not mainly focusing on tracking and analysing currency exchange rates. Indeed, they are serving as a multi-purpose application by including the tracking of currency exchange rates as one of their supporting functionalities. For example, Wise and XE.com serves transferring money overseas as their main feature besides tracking the currency exchange rate. This causes the user interface of the application to be overloaded with modules resulting in disruption to users who only need to access the exchange rate tracking functionality.

Additionally, a certain amount of exchange rate tracking systems like DailyFX and AvaTrade is consisting of financial jargons which will induce unnecessary complexity to the user interface since a portion of users do not have financial background. These financial terminologies should be avoided or explained to avoid user confusion. Pratama and Cahyadi (2010) mentioned that people will only tend to try the application when the display looks comfortable, similar to the cover of a book. Meanwhile, a study has also indicated that 77% of respondents were unsatisfied with their time being wasted while 62% felt stress and frustration due to complex user interface design (Suduc, Bizoi and Filip, 2010). Thus, a straightforward user interface is always a crucial factor for attracting users and maintaining their experience.

1.4.2 Lacking Flexibility for Personalized Notifications

The second issue is the lacking flexibility or unavailability of customizable notifications in the system. By and large, users of currency exchange rate tracking systems shall be given the capability of generating their preferred and personalized alerts. For example, XE.com allows users to generate their notifications by specifying the condition of exchange rate of pair currency. Furthermore, XE.com also grants users the ability to monitor the trend in the exchange rate by sending notifications to email periodically. However, this feature can be continuously improved by adding extra flexibility to refine the system. For instance, users are unable to disable the notification once it is being set up and this will end up causing undesired disturbance. Another problem arises when users set up multiple notifications, viewing and managing all these personalized notifications will become a challenging and troublesome task.

Meanwhile, many existing systems do not support personalized notifications. Without a question, having personalized notifications is crucial so users can receive instant notification of significant rate changes. Immediate notification of real-time exchange rates supports users to perform real-time decision-making so they can keep an eye on potential profit and avoid potential loss.

1.4.3 Insufficient Data Visualization

The third issue is the inadequate data visualization techniques used to showcase the trend of the historical exchange rate data. To illustrate the problem, most of the widely used systems like XE, Wise, Oanda, and so on are solely displaying the historical currency exchange rate in the form of a line chart. It is undeniable that the insight to be gained is limited if there is only one chart. Therefore, extra chart like heat map can be applied into the system to do benchmarking between currency exchange rates. Nevertheless, it is understandable because the functionality of tracking currency trends is only one of their supporting modules that constitute a small part of the entire system so excessive charts will only lead to higher complexity.

Moreover, although some foreign exchange applications do provide more information and analysis on the exchange rate data, they are having poor

data visualization techniques. The currency exchange rate data should be visualized interactively to maximize user experience.

1.5 Project Objectives

The objectives of this project are as follows:

1) To understand the requirements of the currency exchange rate tracking system
2) To develop a web-based application for the currency exchange rate tracking system
3) To evaluate the currency exchange rate tracking system using different testing strategies

1.6 Proposed Solution

First and foremost, to tackle the problem of having a complex user interface, the currency exchange rate tracking website will be using a dashboard design to visualize the data. It is widely acknowledged that a dashboard is a medium of delivering at-a-glance visibility which can transform even the most complicated data into a form that is simple to read and interpret (DashboardFox, 2020). As a result, users do not need to be skilled in data analysis to easily obtain their desired insight through a glance at a dashboard. Furthermore, a dashboard comes in handy for establishing data-driven decision-making, which means that users can identify potential problems and opportunities in the early stage to support their decision-making process (Yellowfin, 2022). Besides that, financial terminologies included should be provided an explanation and usage to facilitate users without financial knowledge and reduce the complexity of the user interface.

Moreover, to mitigate the issue of lacking flexibility in creating personalized notifications, notifications setting screen should be prepared in the system for users to view and manage all their enabled notifications at once. Within the setting screen, the system should also allow users to perform CRUD operations to manage the notifications as in the figure below.

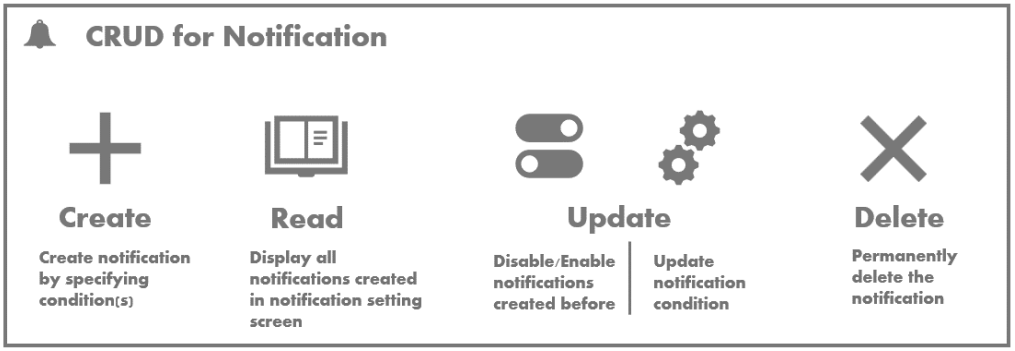


Figure 1.6.1: Overview of CRUD Operations for Notification

Last but not least, to address the issue of inadequate data visualization, the currency exchange rate data should be presented in various forms to act as additional indicators so more valuable insights can be provided to users. For instance, line charts can be used to display the historical exchange rate and relative strength index of selected pair currencies. In a similar manner, a heat map can be used to perform benchmarking between pair currencies so the relationship between pair currencies can be identified. Bar charts can also be used for the straightforward comparison of pair currencies. The charts should also be built in an interactive manner to maximize user experience.

The high-level architectural design of the currency exchange rate tracking web-based application is as follows:

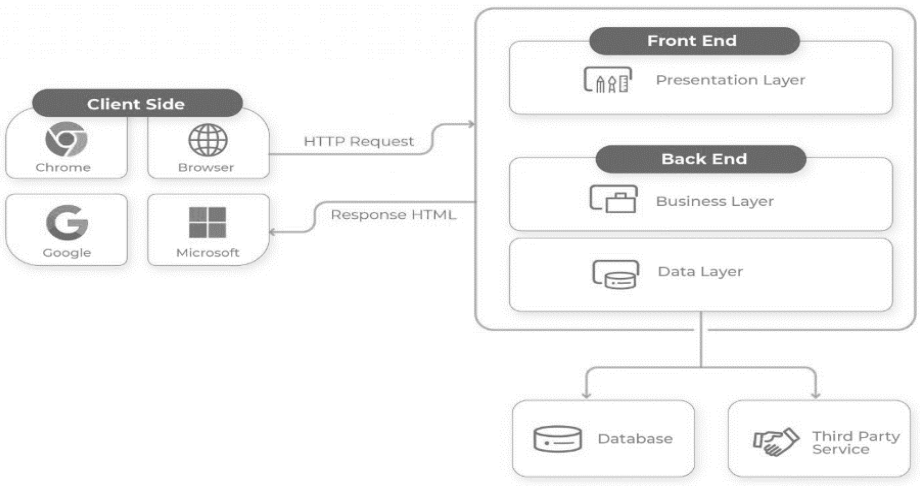


Figure 1.6.2: High-Level Architectural Design of Application (William, 2022)

The web-based application will be utilizing a 3-tier architecture which includes three layers: presentation layer, business layer, and data layer. The front end of the application will be developed using Javascript and Bootstrap while the back end of the application will be implemented using a Python framework, Flask. The role of the front end is presenting the GUI to clients while the back end aims to handle the logical operations and communicate with both database and third-party service through API.

1.7 Proposed Approach

Prototyping is selected as the proposed methodology for this project. According to this software development methodology, a prototype is produced, tested, and revised iteratively until it is acceptable (Martin, 2019). The reason for choosing this methodology is that it is one of the most effective methodologies in software development especially when the requirements are not clarified and finalized. By involving users in prototype evaluation, users can give early feedback on refining the requirements and prototype before implementation of the system takes place. The figure below shows the workflows of the system development using prototyping as the methodology.

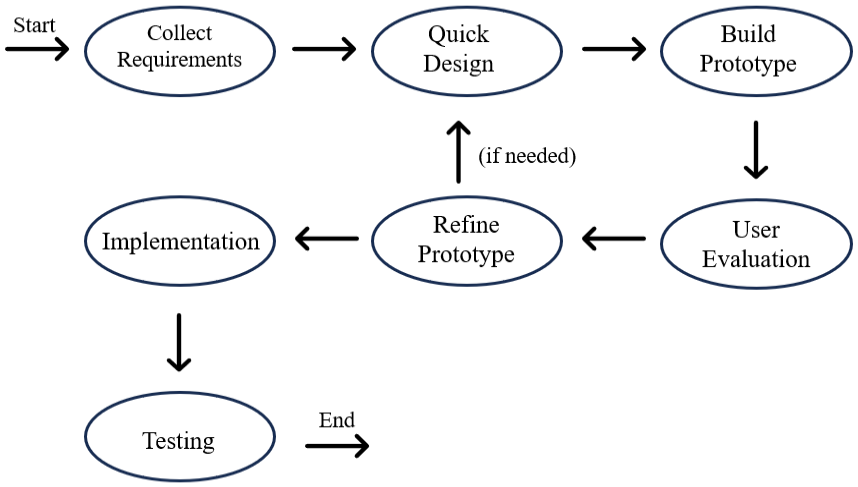


Figure 1.7.1: Prototyping Workflow

1.8 Project Scope

This project aims to develop a web-based application that implements the currency exchange rate tracking system. The application will be built using Flask which is a Python backend framework while JavaScript and Bootstrap will be used for the frontend part. The target users of this currency exchange rate tracking system are mainly foreign exchange market traders because this application focuses on visualizing real-time data as forex indicators that provide valuable insights to forex traders. Moreover, the application is also targeting business executives so they will not miss out on any opportunity to grow their companies' market share and maximize their profit. Additionally, the application is also suitable for individuals who only need the basic functionalities of the system like looking at the past currency exchange rate. The proposed modules included in the currency exchange rate tracking system are as follows:

1.8.1 Login Module

Below are the functionalities for the login module of the system:

- Sign Up Account
- Login Account
- Logout Account
- Reset Password
- Google Sign In

1.8.2 Data Visualization Module

Below are the functionalities for the data visualization module of the system:

- Fetch currency exchange rate data via external API
- Display historical data in the form of line chart
- Compute and display the rate of changes of pair currency within the duration
- Compute the historical Relative Strength Index (RSI) of currencies pair
- Display the Relative Strength Index (RSI) of currencies pair in the form of line chart
- Display the Currencies Correlation Matrix in the form of heatmap
- Compare the exchange rate of pairs of currencies in the form of bar chart

1.8.3 Notification Module

Below are the functionalities for the notification module of the system:

- Set condition to send notification
- Set notifications to be sent periodically
- Write notes to be sent with the notification
- View and manage all notifications on one screen
- Turn off the notification temporary
- Turn on the notification that was turned off
- Update the condition or period of the notification
- Delete the notification that was set before
- Send notifications through email
- View in-app notifications

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, a variety of existing currency exchange rate tracking applications or similar systems will be reviewed and compared in terms of their features, strengths and weaknesses in section 2.2. Furthermore, different commonly used backend frameworks and software development methodologies will be reviewed separately in section 2.3 and section 2.4 to select the most appropriate one to be applied in this project.

2.2 Review of Existing Currency Exchange Rate Tracking Application

In this section, XE.com, Wise and Avatrade are chosen to undergo literature review. However, these applications might have been deployed on both mobile platform and website, since this project is solely to develop a web-based application, hence the literature review will only be conducted based on the website versions of these applications.

2.2.1 Existing Features in XE.com

XE.com provides services for both personal usage and business usage. Some common features embedded in the application for personal usage are sending money abroad, checking live currency exchange rates, reviewing historical currency exchange rates in the form of line chart and setting up rate alerts. On the other hand, XE.com also supports business transactions to overseas, company payments, company forex risk management and API integration to the company's system for business usage. Besides that, XE.com keeps their continuous effort in monitoring currency news and performs analysis. XE.com also prepares multiple calculators like credit card charges calculator, forex charges calculator, travel expense calculator and so on as their extra features which is free of charge.

2.2.2 Strength and Weakness of XE.com

Without a question, most people subscribe to the view that the biggest strength of XE.com is the user-friendly interface. All the functions in the application are organized into personal section and business section. Different pages and headers will be shown based on user selection on either personal view or business view. This comes in handy to filter out excessive function based on user needs and mitigate the interface's complexity. Moreover, majority users have come to a consent that XE.com is one of the platforms that provides the most accurate and up-to-date exchange rates.

However, the weakness is the system only offers the fundamental forex market analysis on historical exchange rate data. Limited advanced analysis and inadequate data visualization will cause the application to lose potential users like forex traders. They might prefer other similar systems with more forex indicators and analysis to aid them in getting profits through the forex market. Another weakness is the unavailability of alert management feature in the system. Users unable to update and delete the notifications once initialized.

2.2.3 Existing Features in Wise

Wise offers the most basic functionalities corresponding to other currency transferring systems like transferring money internationally, paying overseas invoices, converting currencies, checking real-time exchange rates, visualizing historical exchange rates and generating rate alerts. Most importantly, once a user registers successfully and owns an account, the user can store more than forty currencies in the Wise account. This feature leads to the birth of Wise debit card which has made Wise to become outstanding in the competitive market. Users can apply for the Wise debit card to spend abroad without paying any subscription fee.

2.2.4 Strength and Weakness of Wise

Similar to XE.com, the features provided by Wise are categorized into personal usage and business usage. Users can eliminate extra functionalities and complexities by switching between personal view and business view resulting in a comprehensive interface. Another advantage of Wise is the transparency of

its fee structure, the transfer amount, fee and exchange rate will be displayed clearly for confirmation before they proceed to transaction process to minimise the hidden costs. At the same time, in perspective of currency exchange rate tracking system, Wise does a good job by providing an alert management screen to facilitate users in viewing and managing all the previously generated alerts on one screen although there might be lacking accessibility of the feature since users can only navigate to this screen through email.

However, it is undeniable that the weakness of the application is that Wise only provides historical currency exchange rates without any further analysis and additional forex indicators. Hence, the targeting users of this applications are always the companies which have moved to the international stage and people with requirement of international transactions but not forex traders.

2.2.5 Existing Features in AvaTrade

AvaTrade is well-known as a forex trading platform that offers trading service for a wide variety of financial instruments, including shares, commodities, indices, currencies and even cryptocurrencies. AvaTrade also provides a risk management tool named AvaProtect which will reimburse users for losing trade at the end of protection period. Besides that, AvaTrade is also built in with financial instruments tracking dashboard which comes with the functionalities data visualization based on variety of financial indicators, displaying currently most popular trading financial instruments, showing historical trading records, giving financial signals and financial news. Furthermore, AvaTrade has prepared a variety of education resources inclusive of some paid courses as well as free trading notes for beginners which make this application favourable by huge amounts of forex traders.

2.2.6 Strength and Weakness of AvaTrade

Majority of users do agree that AvaTrade which simultaneously acts as a forex trading system has successfully provided a sufficient number of financial indicators and data visualization in assisting them to make decisions on their forex trading. AvaTrade has offered more than seventy financial indicators to

cater users' different needs. Some of these popular indicators include Relative Strength Index (RSI), SMA, Bollinger Bands and so forth.

However, it comes to a debate when talking about the user interface of the website. Some users argue that the website is overloaded with functionalities, and it causes extra workloads when finding the desired service through the website especially those new users who are unfamiliar with the website. Furthermore, the dashboard part also faces the overload issue by having the popular trading financial instruments and data visualization of various charts in the same page. Moreover, the website version of AvaTrade does not support the functionality of setting up rate alerts which might cause users to miss potential profit and suffer from potential loss.

2.2.7 Comparison of Applications by Features

The table below shows the overall comparison of XE.com, Wise and AvaTrade in terms of features.

Table 2-1: Comparison of existing currency exchange rate tracking applications by features

Features	**Applications**		
	XE.com	**Wise**	**AvaTrade**
Check live currency exchange rate	Yes	Yes	Yes
Set rate alerts	Yes	Yes	No (website)
Display historical currency exchange rate	Yes	Yes	Yes
Advance analysis of currency exchange rate (extra financial indicator)	No	No	Yes
Risk management tool	Yes	Yes	Yes
Provide API for integration	Yes	Yes	No
Overseas financial transaction	Yes	Yes	No
Currency conversion tool	Yes	Yes	Yes
Foreign currency trading	No	No	Yes

Provide economic news	Yes	No	Yes
Apply for debit card	No	Yes	No
Offer learning resources	No	No	Yes
Multi-currency account	Yes	Yes	No

2.2.8 Comparison of Applications by Strengths and Weaknesses

The table below shows the overall comparison of XE.com, Wise and AvaTrade by strengths and weaknesses in context of currency exchange rate tracking system.

Table 2-2: Comparison of existing currency exchange rate tracking system by strengths and weaknesses

	XE.com	**Wise**	**AvaTrade**
Complexity of User Interface	Strength	Strength	Moderate
Flexibility of Creating Personalized Alerts / Notifications	Moderate	Strength	Weakness
Sufficiency of Data Visualization	Weakness	Weakness	Strength

2.3 Review on Backend Framework

In this section, Flask, Laravel and Django are chosen to undergo literature review and comparison in terms of their characteristic, complexity and use cases. Eventually, the most suitable backend framework will be decided and applied during the development of the system.

2.3.1 Flask

Flask is a web application framework which utilizes programming language of Python. Flask is also commonly known as microframework which is designed to maintain the application's scalability and complexity (Python Tutorial, 2021). Although it does not provide as many built-in features as compared to other common backend frameworks, but it offers the fundamental tools required to build the web applications and API. By using Flask, developers can enjoy the

largest flexibility to control over the components to be used based on the system requirements.

2.3.2 Laravel

Laravel is a well-known open-source PHP web application framework which adopts the Model-View-Controller (MVC) architectural pattern . Besides that, Laravel is built in with a variety of tools, features and conventions which aid developers in building robust application with high maintainability. One example of assisting tools is the Blade Template engine which provides features like creating dynamic HTML templates and template inheritance. Laravel also comes with Artisan CLI which supports automation in general development tasks.

2.3.3 Django

Similar to Flask, Django is a high-level web application framework written in Python. Django follows Model-View-Template design pattern, and it is designed to assist developers in building web application rapidly by offering tools and conventions for common development tasks. Django emphasizes on DRY (Don’t Repeat Yourself) practices by providing a set of ready-to-use features such as login system, database connection, CRUD operations and so on (W3schools, n.d.). These features improve the reusability of components and shorten the development time of a web-application.

2.3.4 Comparison Between Backend Frameworks

The table below shows the overall comparison of Flask, Laravel and Django in terms of language used, number of built-in functions, complexity and their use case.

Table 2-3: Comparison between Flask, Laravel and Django

	Flask	Laravel	Django
Language	Python	PHP	Python
Built-in Functions	Limited	Large Number	Large Number
Complexity	Low	Moderate	High
Use Case	Small - Medium Application	Large Application	Large Application

2.3.5 Summary on Backend Framework

Considering the learning curve of the backend framework and the nature of the project, Flask is chosen to be used as the backend framework in this project. This is due to Flask is a microframework which is more beginner friendly as compared to other two frameworks. At the same time, Flask is the most suitable framework for this small project and Flask is capable of setting up endpoints for fetching currency exchange rate data easily which can fulfil the requirements of this system.

2.4 Review on Software Development Model

In this section, prototyping model, waterfall model and iterative model are chosen to undergo literature review and comparison in terms of their characteristic, flexibility and use cases. The most suitable software development model will eventually be concluded and adopted in this project.

2.4.1 Prototyping

The basic idea of prototyping model is about repeating the process of gathering user requirements, performing quick design, building prototypes, involving user for evaluation and refining prototype before the prototype includes all the features of the application and successfully obtains customer satisfaction. This model is to ensure the prototype is built according to clients' preferences before moving to the implementation stage which is highly vulnerable to changes in system requirements.

The core benefit of using prototyping as development methodology is early and active user involvement. Thus, defects and misunderstandings can be minimized in the initial stage of the project to avoid any cost overrun and delay in schedule. Besides that, missing functionalities from clients can be identified as well when clients interact with the prototype resulting in lower possibility of product failure (Martin, 2019).

2.4.2 Waterfall

Waterfall model is a widely recognized traditional software development methodology which follows a sequential approach during the software development lifecycle. Waterfall model is highly dependent on the development team to follow a sequence of phases and never move on until the preceding phase is finished which contributes to a clearly defined project structure and comprehensive documentation (Lucidchart, 2018). This model is best suited for small projects with clear requirements at the beginning stage.

However, the main disadvantage of waterfall model is due to its inflexibility as a linear model. The model is rigid and struggles to handle changes because it is changing and expensive to make modifications once that particular phase is already been done. The second disadvantage of Waterfall model is late user involvement in the development process. According to waterfall model, users may only be able to commit their feedbacks in later stages of the project which causes the project to be exclusively risky because the deliverables might not suit client's expectation.

2.4.3 Iterative

Iterative model basically is a software development methodology that incorporates consecutive cycles of planning, designing, implementing, testing and refining in order to build an application. The project is broken down into smaller and manageable parts named iterations. Each iteration outputs a workable version of software which is refined and added with functionalities upon the succeeding iterations.

Tha primary benefit of Iterative model is the functionalities with high priority can be included in the first iteration and developed quickly in the

software development lifecycle (Tutorialpoints, n.d.). Deliverables obtained in each iteration can be used for testing and user evaluation to find defects and gather user feedbacks for continuous refinement of the system. However, this model is not compatible with project which consists of undefined major requirements. Moreover, this model is apparently inappropriate for small projects due to its complexity, and it might consume more resources compared to other models.

2.4.4 Comparison Between Development Model

The table below shows the overall comparison of Prototyping model, Waterfall model and Iterative model in terms of nature, ability of user involvement, flexibility and suitable project size.

Table 2-4: Comparison between Prototyping, Waterfall and Iterative Model

	Prototyping	Waterfall	Iterative
Model Nature	Cycle on Planning & Designing	Linear	Cycle on Whole SDLC Phases
Early User Involvement	Yes	No	Yes
Flexibility	High	Low	High
Project Size	Small	Small	Large

2.4.5 Summary on Development Model

By considering the nature of this project, prototyping will be selected as the software development methodology to be used in this project due to a number of convincing reasons. Firstly, prototyping works best when the user requirements are remained ambiguous because user can be involved in the prototype evaluation to give their feedback and suggestions. Along with each prototype developed, client will end up providing more specific requirements of their demanding system. Secondly, this project is strictly bounded by limited timeframe. It would be less risky to make refinement on designing phase and prototyping phase instead of implementation phase.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.1 Introduction

In this chapter, all phases within the selected project development methodology are discussed from top to bottom in section 3.2. A figure which illustrates the overall phases of the methodology workflow will be provided as well. Moreover, the project work plan including the work breakdown structure (WBS) and Gantt chart will be included in section 3.3.

3.2 Project Development Methodology

As concluded in the literature review after comparison between multiple development methodologies, prototyping will be adopted for this project of building a currency exchange rate tracking system. The figure below shows the overview workflow for all phases in software development lifecycle (SDLC) using prototyping.

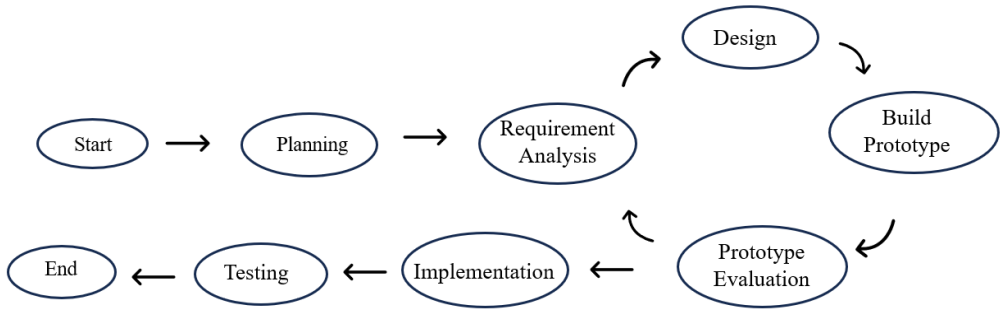


Figure 3.2.1 Project Prototyping Methodology Overview

3.2.1 Phase 1: Planning

Planning phase is the first phase in this project. During this phase, the problem statement, project objectives and the project scopes will be identified. Secondly, literature review is conducted among existing currency exchange rate tracking systems to further understand the project background and requirements. Thirdly, multiple backend frameworks and development methodologies will be compared and the most suitable one will be decided to be used in this project.

Lastly, a work-breakdown structure (WBS) following by Gantt chart will be generated by breaking down each activity in all phases into smaller tasks. Each activity start date and end date will be planned and listed in the Gantt chart. The Gantt chart is then acting as a reference to ensure that the actual progress is always align with the Gantt chart so the project can be finished within the limited timeframe.

3.2.2 Phase 2: Requirement Analysis

During the requirement analysis phase, the initial functional requirements and non-functional requirements will be generated based on the review on existing systems and discussion with supervisor. The requirements will be refined along with the evaluation of the prototype in the following phase developed.

3.2.3 Phase 3: Design

During the design phase, a use-case diagram (UCD) will be generated according to the system requirements identified earlier and the actors involved in the system. Each use cases come with its own use-case description to further specify the system workflow. Moving on, the web application architecture will be designed and provided with detail elaborations. Lastly, an entity relationship diagram (ERD) will be constructed to conceptualize the database design. These designs might undergo refinement according to the result of prototype evaluation.

3.2.4 Phase 4: Build Prototype & Prototype Evaluation

In this phase, the prototype of the currency exchange rate tracking system will be developed using Figma to showcase the user interface and system functionalities to the users and supervisor. The static prototype will be included in the proposal report as screenshots and the dynamic prototype will be used for presentation to the supervisor. After that, the prototype is evaluated, and feedbacks are gathered from supervisor and users. The refinement of the requirement specifications, system design and user interface design will be conducted to develop the consecutive version of the prototype. The process will be repeated for multiple times before moving on to the following phases.

3.2.5 Phase 5: Implementation

The implementation phase will be focusing on development of the currency exchange rate tracking web-application. The frontend client-side will be implemented according to the user interface design. Besides that, APIs for the backend server-side and database connectivity will be covered in this phase as well.

3.2.6 Phase 6: Testing

In the beginning of the testing phase, the test cases for unit testing, integration testing and user acceptance testing will be identified and designed. Once test cases are organized, the unit testing is conducted to make sure that all features in the currency exchange rate tracking system can work independently as intended. Defects identified in this process will be fixed immediately. After the system passed the unit test, integration testing will be performed to ensure multiple modules within the system can work together to provide the complete implementation of system functionalities without any errors. Lastly, user acceptance test is carried out to test whether users are feeling comfortable with the current system. This phase is also signified as the end of this project.

3.3 Project Work Plan

In this section, work breakdown structure (WBS) will be constructed by decomposing the important activities in all phases into smaller components. After that a Gantt chart is built by planning the start date and end date of each activity. The Gantt chart is then compared with the actual progress throughout the whole project lifecycle.

Gantt Project which is a project management software will be used to construct the Gantt Chart of this project. The work breakdown structure (WBS) of the project is included in this report as Appendix A while the Gantt chart of the project is included in this report as Appendix B.

CHAPTER 4

PROJECT SPECIFICATION & DESIGN

4.1 Introduction

In this chapter, the functional requirements and non-functional requirements are analysed and listed in section 4.2: Requirement Specification. Next, the process of designing and modelling will be conducted in section 4.3 based on the requirements collected. Eventually, various models like use case diagram, entity relationship diagram and prototype will be constructed. The requirements and models might undergo refinement after prototype evaluation.

4.2 Requirement Specification

The system requirements of currency exchange rate tracking system are gathered based on reviewing of existing similar systems in Chapter 2. Among all the features embedded in the existing similar systems, only those features with relation to tracking process of currency exchange rate will be considered as requirement of the proposed system like currency conversion while others like sending money and exchanging money will be excluded. The functional requirements collected and analysed are listed in section 4.2.1 while non-functional requirements are documented in section 4.2.2.

4.2.1 Functional Requirements

1. The system shall allow users to sign in through google account.
2. The system shall allow users to sign up account using their email and password.
3. The system shall allow users to change their profile details including personal details and currency preferences.
4. The system shall allow users to reset password if they forgot it.
5. The system shall allow users to login by entering email and password.
6. The system shall allow users to log out of their accounts.
7. The system shall allow users to select their desired currencies to be visualized for every chart.

8. The system shall allow users to set durations to display only historical exchange rate data within that period.
9. The system shall allow users to view the historical exchange rate of selected currencies in the form of line chart.
10. The system shall allow users to view the historical RSI values of selected currencies in the form of line chart.
11. The system shall allow users to convert their input amount of money from one currency to another currency.
12. The system shall allow users to view the comparison of changes of exchange rate between multiple pairs of currencies in the form of bar chart.
13. The system shall allow users to view the correlation between multiple pairs of currencies in the form of heatmap.
14. The system shall allow users to download the charts.
15. The system shall provide feedback to users when user interact with the charts.
16. The system shall allow users to add a currency as their default currency.
17. The system shall allow users to add another currency as their secondary currency.
18. The system shall allow users to set the favourite currencies.
19. The system shall display favourite currencies over other currencies for currency selection input.
20. The system shall allow users to set alert to notify about the exchange rate value or RSI value periodically.
21. The system shall allow users to define the condition to notify about the exchange rate value or RSI value conditionally.
22. The system shall allow users to view all their alerts in one screen.
23. The system shall allow users to enable or disable the alerts set before.
24. The system shall allow users to delete the alerts set before.
25. The system shall allow users to update the alerts duration or condition.

4.2.2 Non-Functional Requirements

1. Reliability

   The system shall be able to ensure the currency exchange rate accuracy and consistency.

2. Performance

   The system shall be able to provide exchange rate updates with a response time of less than 10 seconds.

3. Usability

   The system shall be able to provide an intuitive and easy-to-learn interface for both proficient users and novice users to use.

4.3 Design & Modelling

In section 4.3.1, all the use cases for the currency exchange rate tracking system are identified based on the functional requirements of the system which contribute to the development of use case diagram (UCD). In section 4.3.2, use case description will be generated for each use cases identified to elaborate the workflow in the system. Moreover, in section 4.3.3, entity relationship diagram (ERD) will be constructed to identify the data to be stored and tables to be used in database server. The data is then further described with the data type used and data length to be stored in the database. Lastly, a prototype is constructed in section 4.3.4 to showcase the user interfaces of the system.

4.3.1 Use Case Diagram

Figure 4.3.1: Use Case Diagram

4.3.2 Use Case Description

4.3.1 Sign Up Account

<table>
<tr><td colspan="2">Use Case Name: Sign Up Account</td><td>ID: 01</td><td>Importance Level: High</td></tr>
<tr><td colspan="2">Primary Actor: User</td><td colspan="2">Use Case Type: Detail, Real</td></tr>
<tr><td colspan="4">Stakeholders and Interests:

User – Register an account in order to access the system</td></tr>
<tr><td colspan="4">Brief Description:

This use case describes how user can sign up an account in the system.</td></tr>
<tr><td colspan="4">Trigger: User wants to use the currency exchange rate tracking system as a new user</td></tr>
<tr><td colspan="4">Relationships:

Association: User

Include: N/A

Extend: N/A

Generalization: N/A</td></tr>
<tr><td colspan="4">Normal Flow of Events:

1. If users want to sign up with Google account, sub-flow 1.1 and 1.2 are performed.
2. If users want to do regular sign up, sub-flow from 2.1 to 2.6 are performed.</td></tr>
</table>

Sub-flows:

1.1. System integrates the users google account.

1.2. System redirects users to add profile page

2.1. User enters the email, password and confirm password.

2.2. User inputs personal information like name, nationality, profile pic

2.3. User inputs currency preference like default currency, second currency and favourite currencies

2.4. System sends a verification link to user's email address.

2.5. User clicks on the verification link.

2.6. User is redirected to login page

Alternate/Exceptional Flows:

1.1.a. If the user's google account cannot be verified, the system will prompt error.

2.1.a. If any each of email, password, name or other information is entered in an invalid format, the system will prompt the user to re-enter.

2.1.b. If any each of email is already existing in the system, the system will prompt the customer to re-enter.

2.5.a. If the verification link is invalid, the system will prompt error

4.3.2 Login Account

Use Case Name: Login Account	ID: 02	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Real	
Stakeholders and Interests: User – Login the previously registered account in order to access the system with user records.		
Brief Description: This use case describes how user can login to an existing account in the system.		
Trigger: User wants to use the currency exchange rate tracking system as existing user.		
Relationships: Association: User Include: N/A Extend: Reset Password Generalization: N/A		
Normal Flow of Events: 1. If user chooses to login with Google account, proceed to step 4. 2. If user chooses to login with email and password, sub-flow 2.1 is performed. 3. If user forgot their password and wish to reset them, sub-flow from 3.1 to is performed. 4. User login successfully and the dashboard is prompt to the user.		

Sub-flows: 2.1. User enters the email and password in corresponding field. 3.1. User clicks on forget password 3.2. System sends a link for password reset to the verified email. 3.3. User clicks on the link to navigate to reset password page. 3.4. User enters new password and confirm password
Alternate/Exceptional Flows: 1.a. If the user's google account cannot be verified, the system will prompt error. 2.1.a. If email and password entered is not correct credentials, the system will prompt the user to re-enter. 3.4.a. If the new password entered is not matching with confirm password, the system will prompt the user to re-enter.

4.3.3 Add / Change Profile Detail

Use Case Name: Change Profile Detail	ID: 03	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Real	
Stakeholders and Interests: User – Edit the information in user profile		
Brief Description: This use case describes how user can edit the information stored in user profile		
Trigger: User wants to edit the information stored in user profile		
Relationships: Association: User Include: N/A Extend: N/A Generalization: N/A		
Normal Flow of Events: 1. If current user having profile, sub-flow from 1.1 to 1.4 are performed 2. If current user not having profile, sub-flow from 2.1 to 2.3 are performed 3. If current user wants to change password, sub-flow from 3.1 to 3.2 are perform 4. If current user signed in with Google wants to add password to account, sub-flow from 4.1 to 4.2 are performed		

Sub-flows:

1.1. User clicks on edit profile button.

1.2. User changes their personal information including name, nationality, profile picture and etc.

1.3. User changes their currency preferences including default currency, second currency and favourite currencies.

1.4. User saves changes.

2.1. User being redirected to add profile page if not owning profile.

2.2. User enters their currency preferences.

2.3. User adds profile.

3.1. User enters the old password, new password and confirm password.

3.2. User saves password changes.

4.1. User enters the new password and confirm password.

4.2. User adds password.

Alternate/Exceptional Flows:

1.4.a. If the user input is invalid or empty for required field, the system will prompt error

3.1.a. If the current password entered is incorrect, the system will prompt the user to re-enter.

3.1.b. If the new password is not matching with confirm password, the system will prompt the user to re-enter.

4.1.a. If the current password entered is incorrect, the system will prompt the user to re-enter.

4.2.a. If the new password is not matching with confirm password, the system will prompt the user to re-enter.

4.3.4 Logout System

Use Case Name: Logout System	ID: 04	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Real	
Stakeholders and Interests: User – Logout from the current account.		
Brief Description: This use case describes how user can logout from the current account.		
Trigger: User wants to change account or stop using the system.		
Relationships: Association: User Include: N/A Extend: N/A Generalization: N/A		
Normal Flow of Events: 1. User clicks on the “Logout” button. 2. System navigates user to the login page.		

4.3.5 View Historical Exchange Rate

Use Case Name: View Historical Exchange Rate	ID: 05	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Real	
Stakeholders and Interests: User – View the line chart that visualizes the historical currency exchange rate.		
Brief Description: This use case describes how user can do setting and view the historical currency exchange rate in the form of line chart.		
Trigger: User wants to view the line chart that visualizes the historical exchange rate of selected currency.		
Relationships: Association: User Include: N/A Extend: Set Alert Generalization: N/A		
Normal Flow of Events: 1. User navigates to the “Exchange Rate Changes” screen. 2. User sets the pair of currency 3. User sets the duration for the historical data to be included.		

4. The line chart is shown when user finishes the setting of currencies and duration 5. The changes of the exchange rate within the selected duration will be computed and displayed in percentage. 6. If user wants to create alert, proceed to alternate flow 6.1, 6,2, 6.3, 6.4. 7. If the alert is triggered, alert will be sent either through in-app notification or email depending on user's preference.
Sub-flows:
Alternate/Exceptional Flows: 6.1. If user wants to create alert periodically, user selects the interval duration for the alert to be sent. 6.2. If user wants to create alert which triggered by conditionally, user sets the condition for the alert to be sent. 6.3. User chooses whether to receive the alert via in-app notification, email notification or both. 6.4. User can write notes to be sent along with the alert or leave the notes blank.

4.3.6 View Historical RSI Value

Use Case Name: View Historical RSI Value	ID: 06	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Real	
Stakeholders and Interests: User – View the line chart that visualizes the historical RSI value trend		
Brief Description: This use case describes how user can do setting and view the historical currency RSI values in the form of line chart.		
Trigger: User wants to view the line chart that visualizes the historical RSI values of selected currency.		
Relationships: Association: User Include: N/A Extend: Set Alert Generalization: N/A		
Normal Flow of Events: 1. User navigates to the RSI values screen. 2. User sets the pair of currency 3. User sets the duration for the historical data to be included.		

4. The line chart is shown when user finishes the setting of currencies and duration 5. Two constant lines are shown in the line chart which act as oversold and overbought indicators. 6. The changes of the RSI value within the selected duration will be computed and displayed in percentage. 7. If user wants to create alert, proceed to alternate flow 7.1, 7,2, 7.3, 7.4. 8. If the alert is triggered, alert will be sent either through in-app notification or email depending on user's preference.
Sub-flows:
Alternate/Exceptional Flows: 7.1. If user wants to set alert periodically, user selects the interval duration for the alert to be sent. 7.2. If user wants to set alert which triggered by conditionally, user sets the condition for the alert to be sent. 7.3. User chooses whether to receive the alert via in-app notification, email notification or both. 7.4. User can write notes to be sent along with the alert or leave the notes blank.

4.3.7 View Exchange Rate Comparison

Use Case Name: View Exchange Rate Comparison	ID: 07	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Real	
Stakeholders and Interests: User – View the bar chart that visualizes the comparison of exchange rate changes between multiple pairs of currencies.		
Brief Description: This use case describes how user can do setting and view the comparison of exchange rate changes between multiple pairs of currencies in the form of bar chart.		
Trigger: User wants to view the line chart that visualizes the comparison of exchange rate changes between selected multiple pairs of currencies.		
Relationships: Association: User Include: N/A Extend: N/A Generalization: N/A		

Normal Flow of Events: 1. User navigates to the Comparison screen. 2. User selects the currencies automatically by choosing either comparison of favourite currencies or comparison of popular currencies. 3. If user want to select the currencies manually, proceed to sub-flow 3.1 and onwards. 4. User sets the duration to compute the changes of exchange rate. 5. The bar chart is shown when user finishes the setting of currencies and duration
Sub-flows: 3.1. User selects a base currency. 3.2. User selects multiple comparing currencies for comparison.
Alternate/Exceptional Flows:

4.3.8 View Currency Correlation

Use Case Name: View Currency Correlation	ID: 08	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Real	
Stakeholders and Interests: User – View the heat map that visualizes the correlation of exchange rate changes between multiple pairs of currencies.		
Brief Description: This use case describes how user can do setting and view the correlation of exchange rate changes between multiple pairs of currencies in the form of heat map.		
Trigger: User wants to view the heat map that visualizes the correlation of exchange rate changes between selected multiple pairs of currencies.		
Relationships: Association: User Include: N/A Extend: N/A Generalization: N/A		

Normal Flow of Events: 1. User navigates to the Currency Correlation screen. 2. User selects multiple currencies 3. User sets the duration for the historical data to be included in computation. 4. The correlation coefficient value of these combination of currencies are calculated. 5. The correlation values are shown as heap map.
Sub-flows:
Alternate/Exceptional Flows:

4.3.9 Convert Currency

Use Case Name: Convert Currency	ID: 09	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Real	
Stakeholders and Interests: User – Convert an entered amount of money from one currency to another.		
Brief Description: This use case describes how user can convert entered amount of money from one currency to another.		
Trigger: User wants to convert their desired amount of money from one currency to another.		
Relationships: Association: User Include: N/A Extend: N/A Generalization: N/A		
Normal Flow of Events: 1. User navigates to the Converter screen. 2. User enters the amount of money he wishes to convert. 3. User selects the currency to be converted from 4. User selects the currency to be converted into.		

5. The exchange rate for that pair of currencies is shown to user. 6. The converted value of money is shown to the user.
Sub-flows:
Alternate/Exceptional Flows:

4.3.10 Toggle Alert

Use Case Name: Enable or Disable Alert	ID: 10	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Real	
Stakeholders and Interests: User – Toggle the functionality of a previously set alert.		
Brief Description: This use case describes how user can enable or disable a previously set alert temporarily by toggling the switch button.		
Trigger: User wants to enable or disable a previously set alert temporarily.		
Relationships: Association: User Include: N/A Extend: N/A Generalization: N/A		
Normal Flow of Events: 1. User navigates to the Alert Setting screen. 2. User views all the alerts set before in the screen. 3. User presses on the switch button to toggle the functionality of alert.		

4.3.11 Edit Alert

Use Case Name: Edit Alert	ID: 11	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Real	
Stakeholders and Interests: User – Edit the setting of a previously set alert.		
Brief Description: This use case describes how user can edit the setting of a previously set alert.		
Trigger: User wants to edit the setting of a previously set alert.		
Relationships: Association: User Include: N/A Extend: N/A Generalization: N/A		
Normal Flow of Events: 1. User navigates to the Alert Setting screen. 2. User views all the alerts set before in the screen. 3. User presses on the edit button to change the setting of the alert. If the alert is set to be periodic, proceed to 3.1. If the alert is set to be conditional, proceed to 3.2. 4. User confirms the edit by click save button if he wants to save the edit.		

6. User cancels the edit by clicking cancel button if he does not wish to save it. 7. User can edit the note to be sent together with alert directly.
Sub-flows: 3.1.a. User can update the duration by selecting from drop-down list. 3.2.a User can update the condition by inputting the condition.
Alternate/Exceptional Flows:

4.3.12 Delete Alert

Use Case Name: Delete Alert	ID: 12	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Real	
Stakeholders and Interests: User – Remove a previously set alert permanently.		
Brief Description: This use case describes how user can remove a previously set alert permanently.		
Trigger: User wants to remove a previously set alert which is useless.		
Relationships: Association: User Include: N/A Extend: N/A Generalization: N/A		
Normal Flow of Events: 1. User navigates to the Alert Setting screen. 2. User views all the alerts set before in the screen. 3. User searches for the alert that he wishes to remove permanently 4. User presses on the delete button to remove it permanently. 5. System prompt users on success deletion.		

4.3.13 View In-App Notifications

Use Case Name: View In-App Notifications	ID: 13	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Real	
Stakeholders and Interests: User – View in-app notifications sent to his/her account		
Brief Description: This use case describes how user can view in-app notifications		
Trigger: User wants to view in-app notifications.		
Relationships: Association: User Include: N/A Extend: N/A Generalization: N/A		
Normal Flow of Events: 1. User checks on the notification icon on the top bar. 2. If there is unread message, a red badge with the number of unread messages will be shown 3. User clicks on the notification icon to open notification modal. 4. Unread notifications are appeared as light blue background while old notifications are appeared as white background.		

4.3.3 System Architecture Design

Figure below describes the overview of the system architecture design of the currency exchange rate tracking system to be developed in this project.

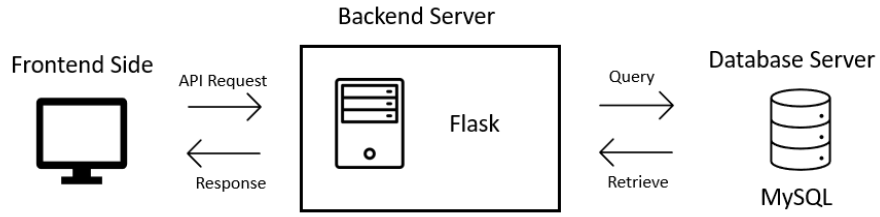


Figure 4.3.2: System Architecture Design

The frontend side is responsible for presenting the web pages to the clients and performing client-side logic like form validation. The frontend side of the system will be built using HTML, Bootstrap and JavaScript. On the other hand, the backend server is responsible for processing request from frontend side, handling server-side logic and interacting with the database server. The backend server will be built using Flask, which is a Python library that prioritizes on developing web applications. Simultaneously, the database server is responsible for storing and managing the user information and other relevant data as tables. MySQL which is a relational database management system (RDBMS) will be used for the database server in this project.

4.3.4 Entity Relationship Diagram (ERD)

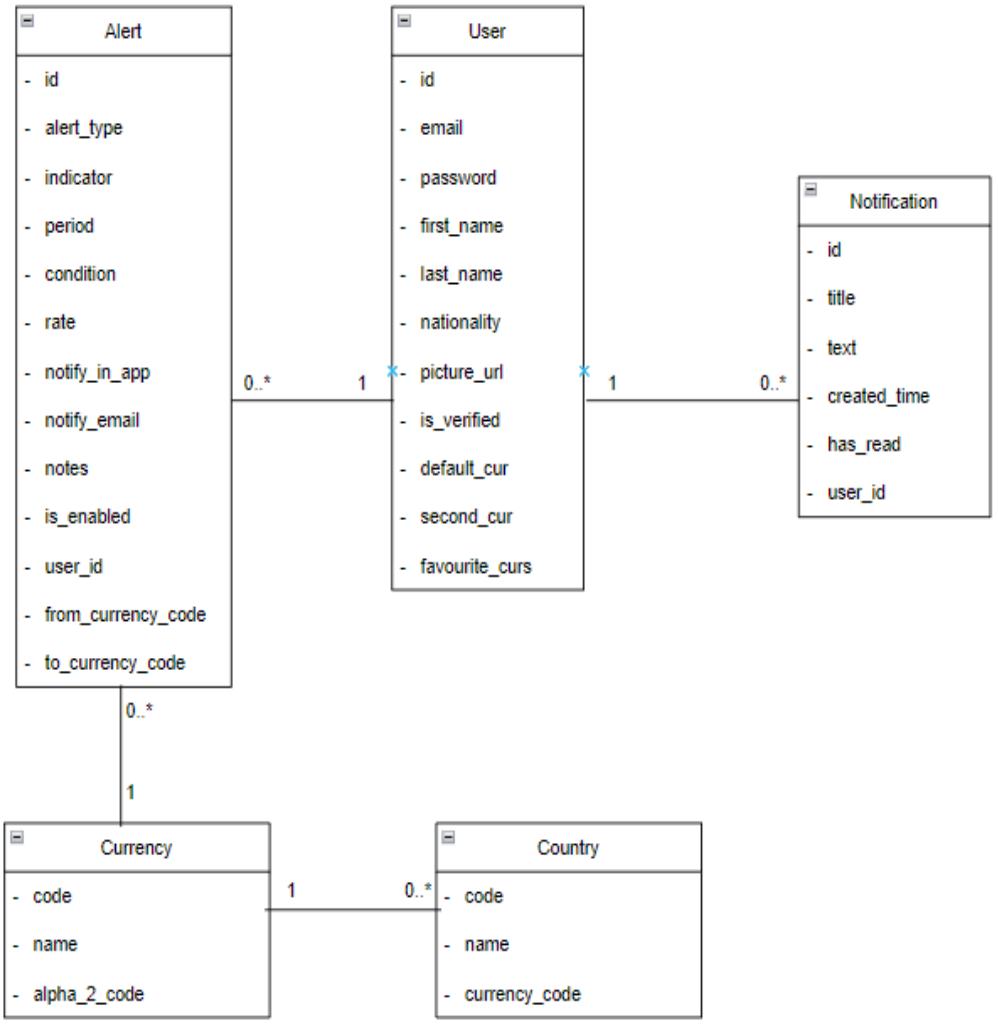


Figure 4.3.3: Entity Relationship Diagram

Table 4-1: Data Dictionary

Table Name	Description	Columns	Data Type
Users	This table contains the record of existing users of the system.	id (PK) email password first_name last_name nationality picture_url	int varchar (150) varchar (150) varchar (150) varchar (150) varchar (3) varchar (255)

		is_verified default_cur second_cur favourite_curs	boolean varchar (3) varchar (3) varchar (255)
Currency	This table contains the information of available currencies in the system.	code (PK) name alpha_2_code	varchar (3) varchar (150) varchar (2)
Country	This table contains the information of countries that using available currencies in the system.	code (PK) name currency_code (FK)	varchar (3) varchar (150) varchar (2)
Alert	This table contains the record of alerts set by users.	id (PK) alert_type indicator period condition rate notify_in_app notify_email notes is_enabled from_cur_code (FK) to_cur_code (FK) user_id (FK)	int varchar (13) varchar (6) varchar (7) varchar (4) float boolean boolean varchar (255) boolean varchar (3) varchar (3) int
Notification	This table contains the record of notifications sent to user.	id created_time title	int datetime varchar (150)

		text has_read user_id	varchar (255) boolean int

4.3.5 User Interface Design

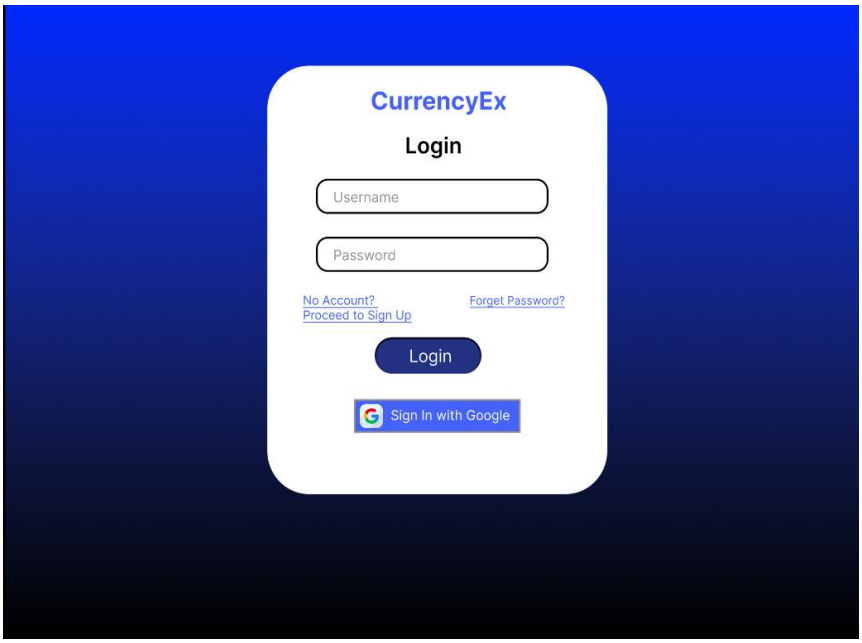


Figure 4.3.4: Login Screen

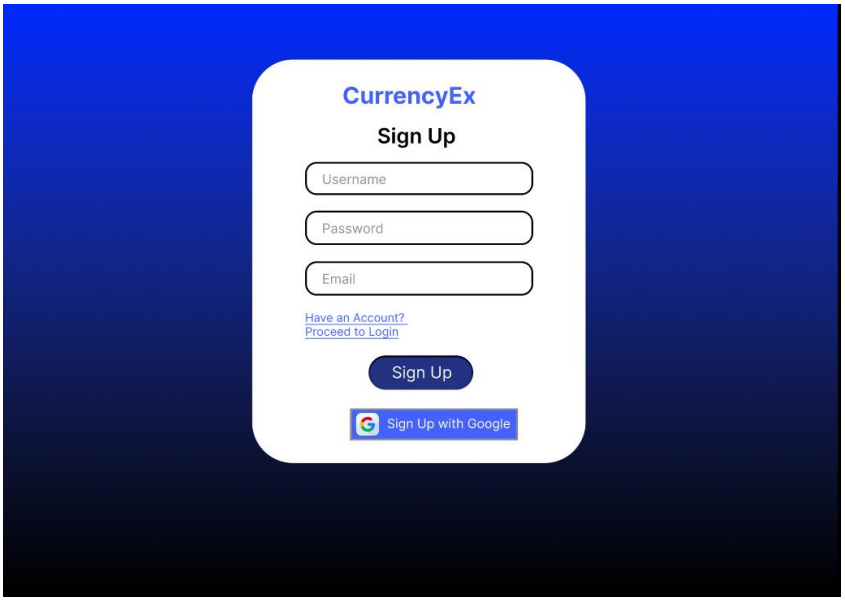


Figure 4.3.5: Sign Up Screen

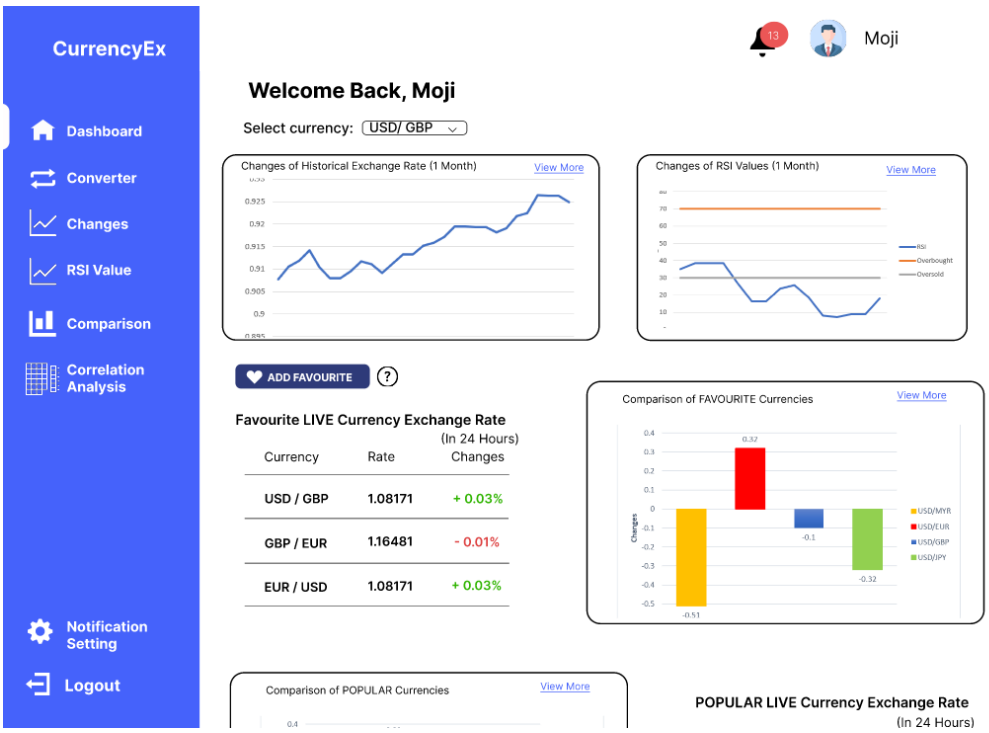


Figure 4.3.6: Dashboard Screen

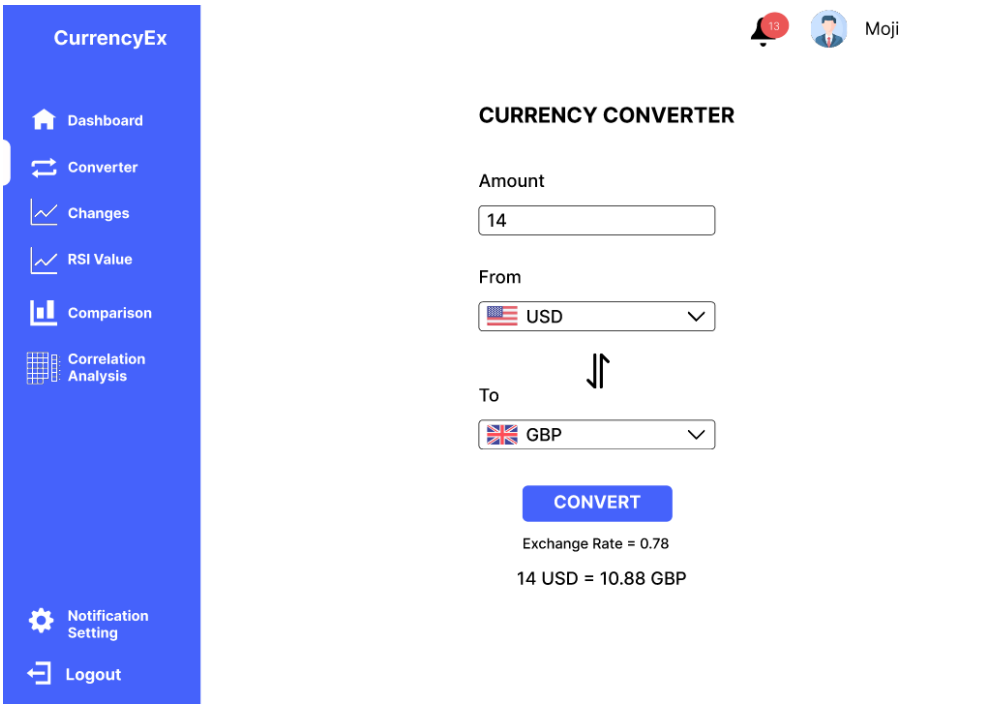


Figure 4.3.7: Converter Screen

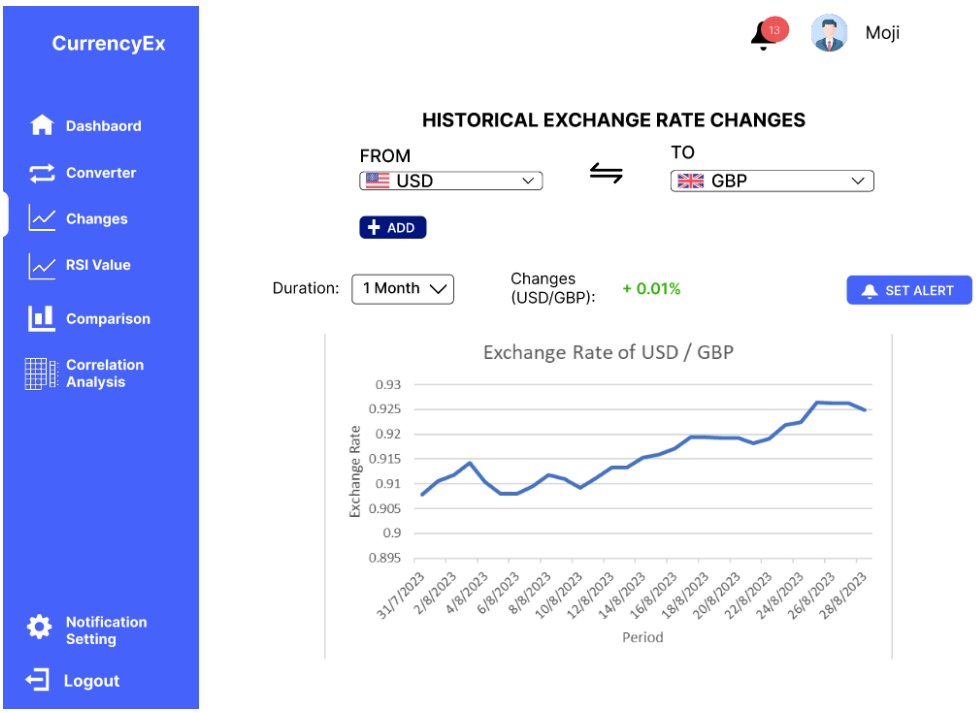


Figure 4.3.8: Exchange Rate Changes Screen

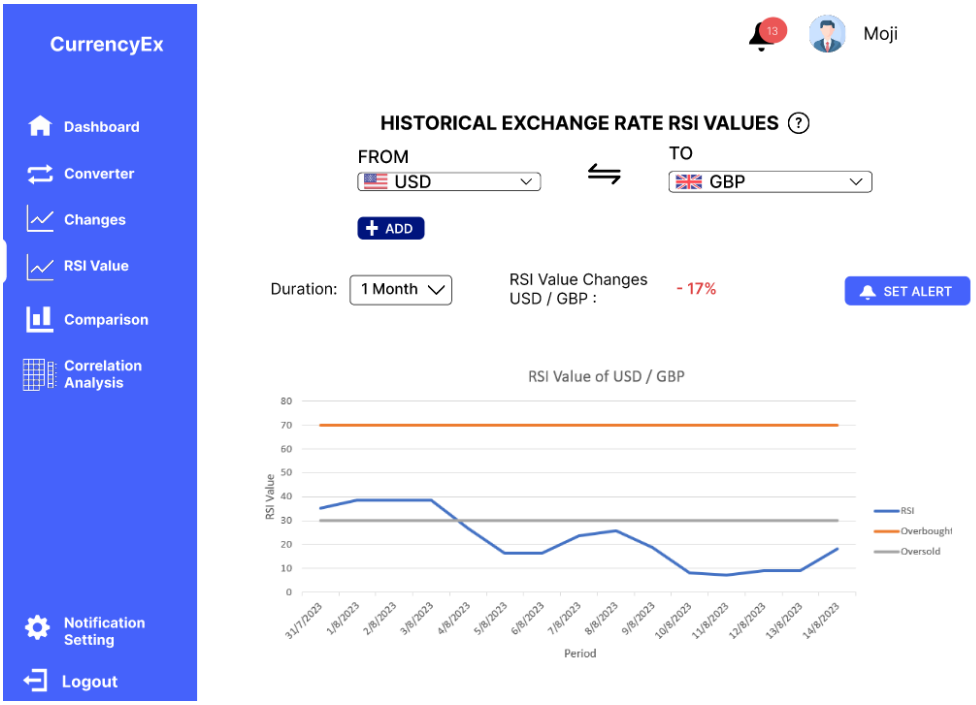


Figure 4.3.9: RSI Value Screen

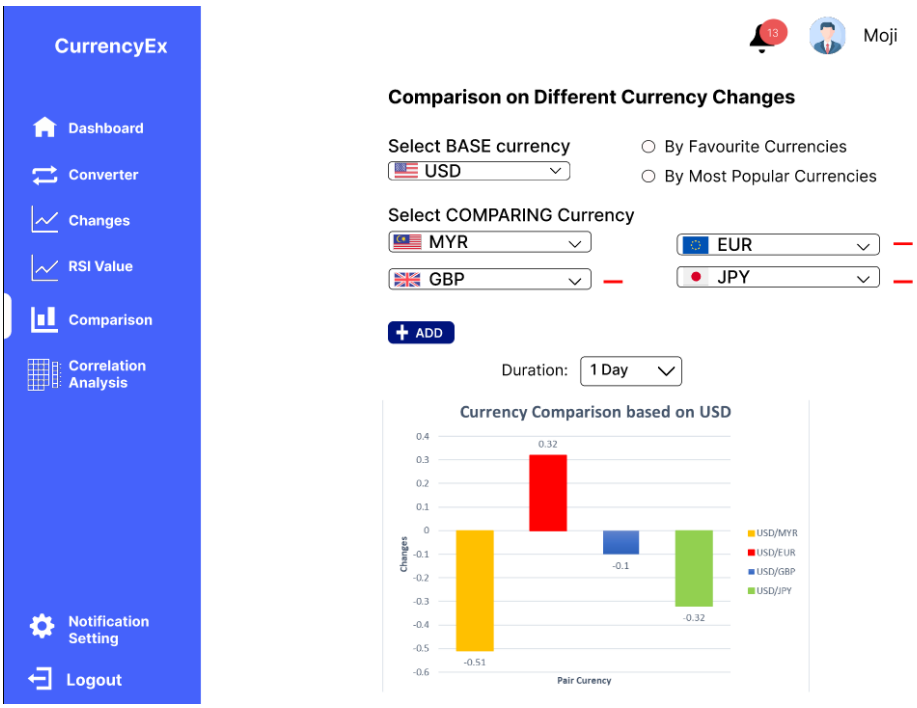
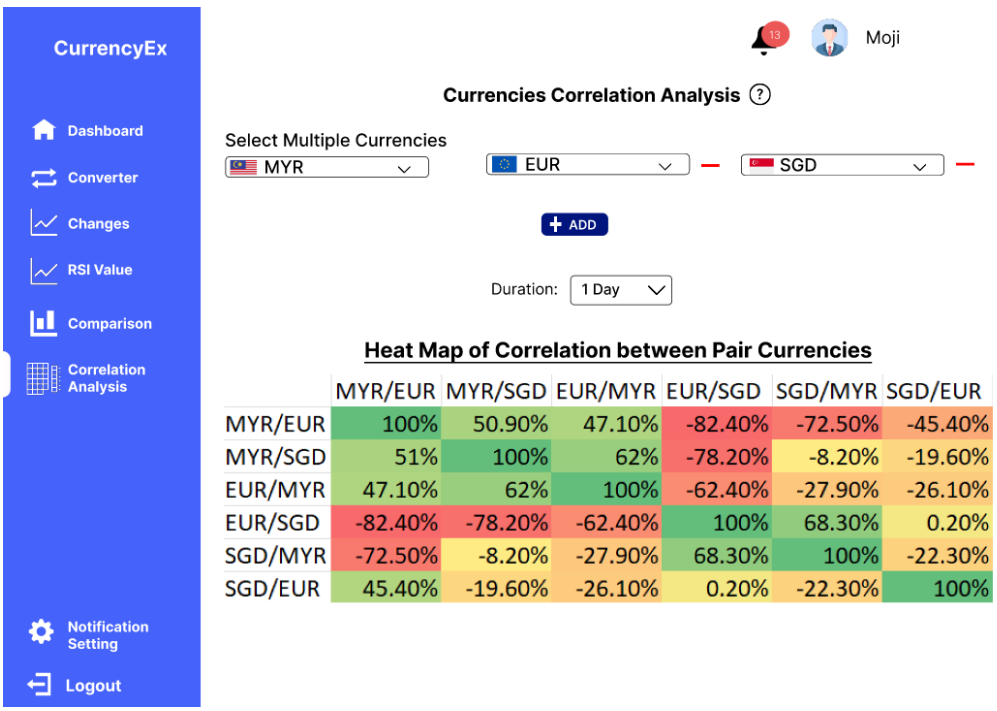


Figure 4.3.10: Comparison Screen



	MYR/EUR	MYR/SGD	EUR/MYR	EUR/SGD	SGD/MYR	SGD/EUR
MYR/EUR	100%	50.90%	47.10%	-82.40%	-72.50%	-45.40%
MYR/SGD	51%	100%	62%	-78.20%	-8.20%	-19.60%
EUR/MYR	47.10%	62%	100%	-62.40%	-27.90%	-26.10%
EUR/SGD	-82.40%	-78.20%	-62.40%	100%	68.30%	0.20%
SGD/MYR	-72.50%	-8.20%	-27.90%	68.30%	100%	-22.30%
SGD/EUR	45.40%	-19.60%	-26.10%	0.20%	-22.30%	100%

Figure 4.3.11: Currency Correlation Screen

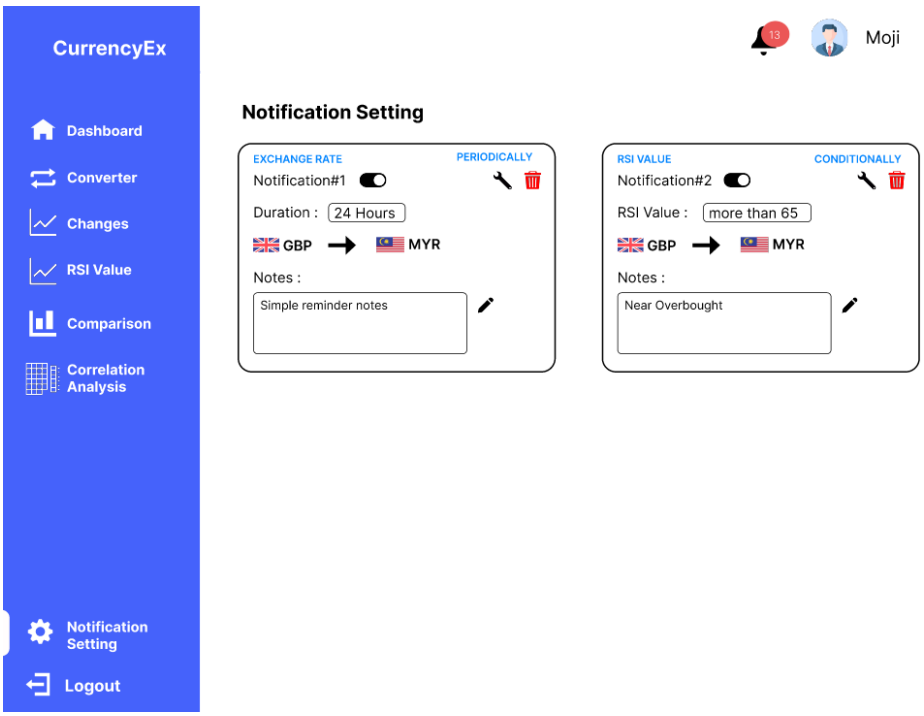


Figure 4.3.12: Notification Setting Screen

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

This chapter focuses on the implementation of the currency exchange rate tracking system, which covers the implementation of the frontend client-side code using Jinja2 templating engine which is built-in inside Flask and also APIs from the Flask server.

5.2 User Authentication Module

5.2.1 Login Account

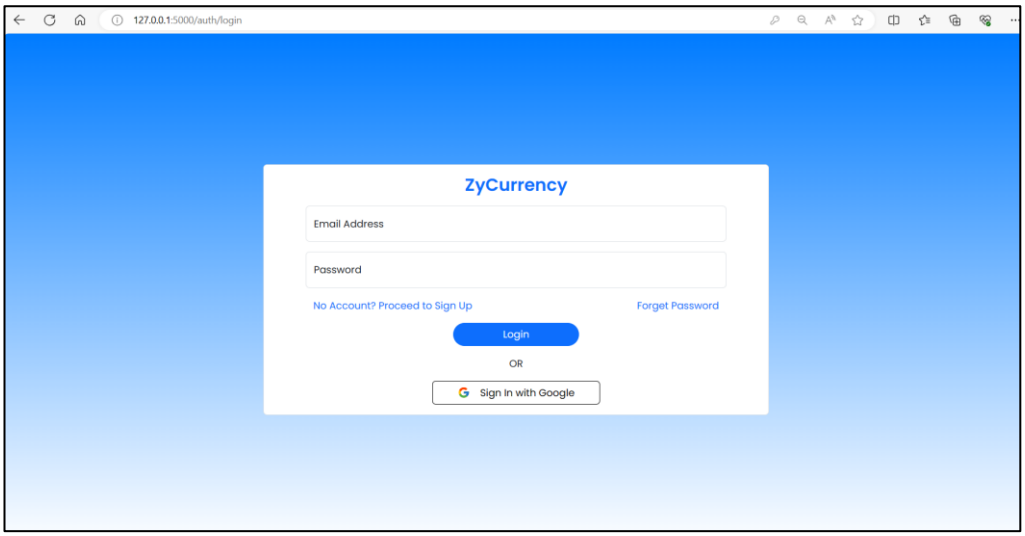


Figure 5.2.1: Login Page

If users are having existing account in the system, user can enter the email address used for registration and also password to login the system.

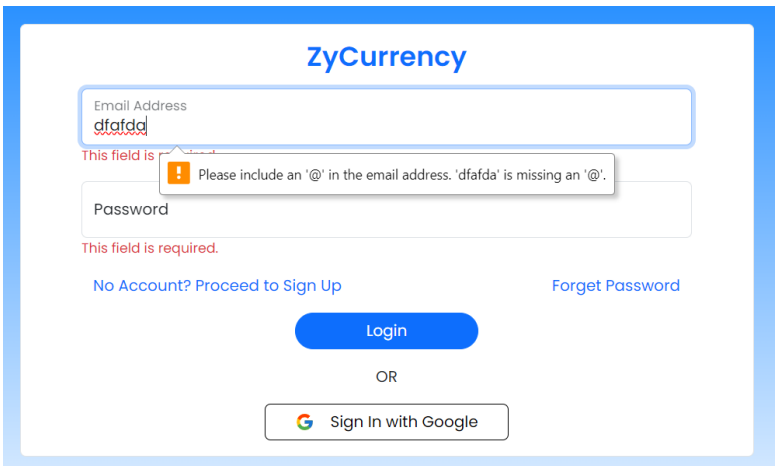


Figure 5.2.2: Login Page with Frontend Validation

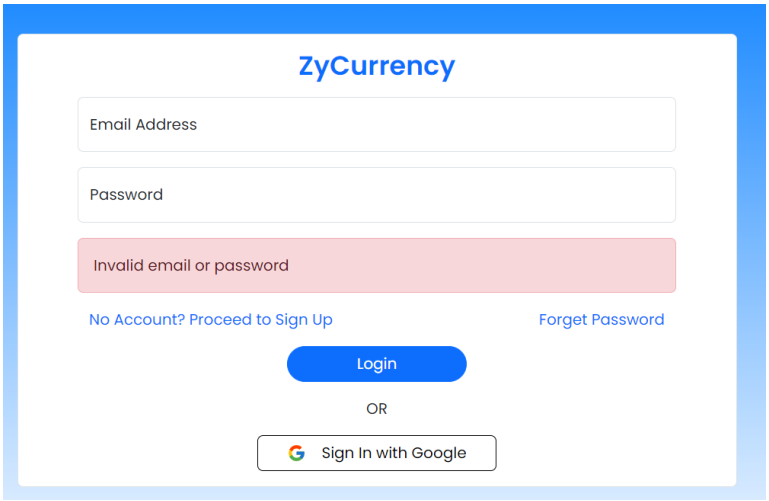


Figure 5.2.3: Login Page with Backend Validation

The validation logics are applied on both frontend and backend to make sure the login data are filled in required format and also to display error messages if the login credentials provided are incorrect.

```
class LoginForm(FlaskForm):
    email = StringField('Email', [
        validators.DataRequired(),
        validators.Email(message='Invalid email address')
    ])
    password = PasswordField('Password', [validators.DataRequired()])
    submit = SubmitField('Login')
```

Figure 5.2.4: Code for Frontend Login Validation

5.2.2 Sign Up Account

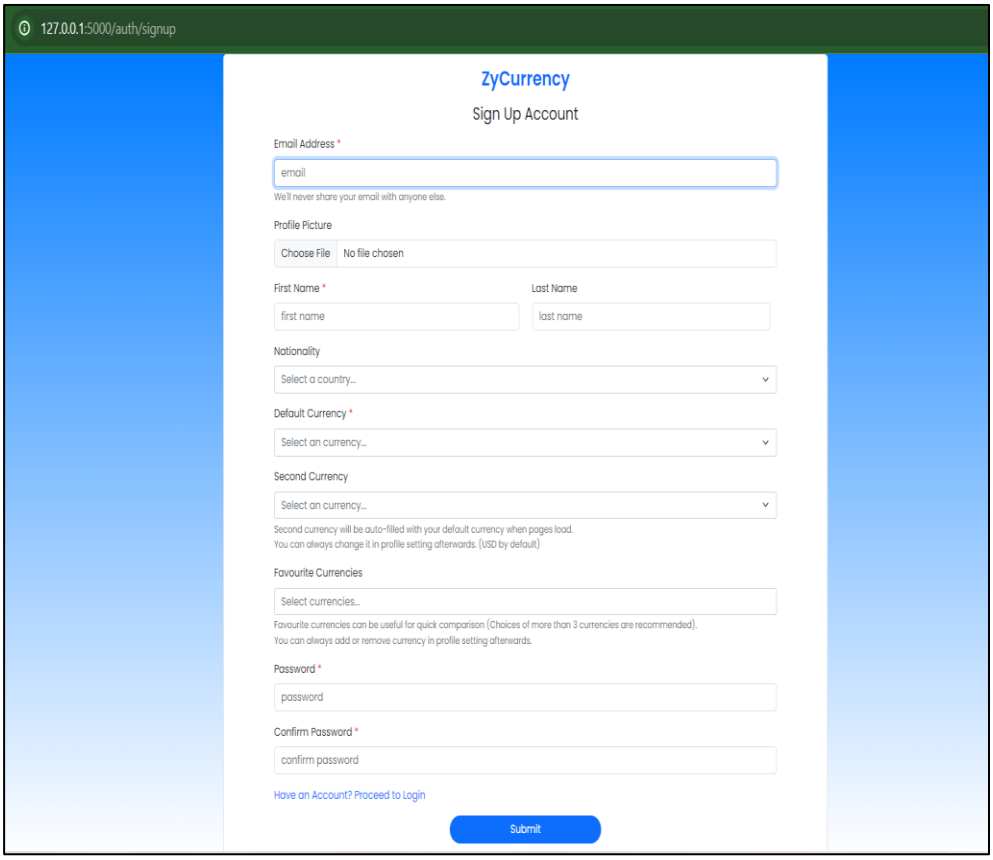


Figure 5.2.5: Sign Up Page

This is the sign-up page where users can register their new account in the system by providing all the required information including their email, name, default currency to be used in system and lastly the passwords. Some optional inputs like profile picture, second currency, favourite currency and so on can enhance the user experience while using the system afterwards.

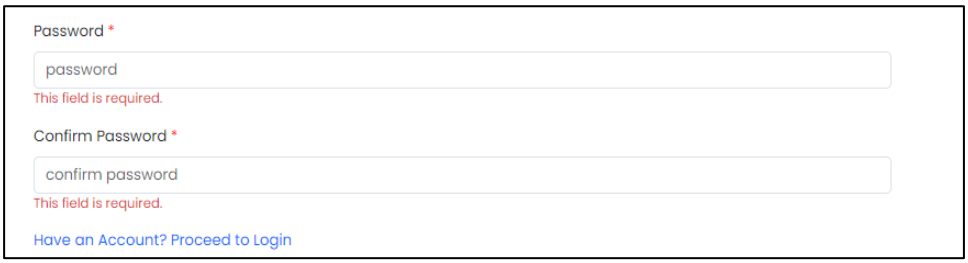


Figure 5.2.6: Sign Up Page with Frontend Validation

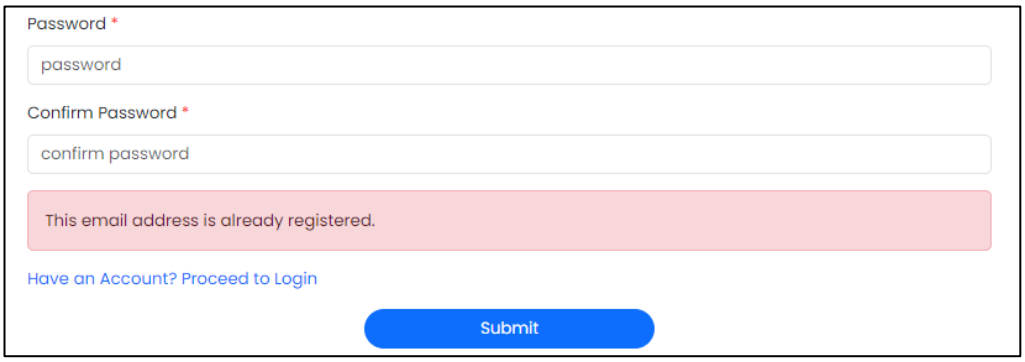


Figure 5.2.7: Sign Up Page with Backend Validation

```
class SignUpForm(FlaskForm):
    email = EmailField('Email', validators=[DataRequired(), Email(message='Invalid email address')])
    first_name = StringField('First Name', validators=[DataRequired()])
    last_name = StringField('Last Name')
    nationality = QuerySelectField(query_factory=lambda: Country.query.all(), allow_blank=True, get_label="name")
    default_cur = QuerySelectField(query_factory=lambda: Currency.query.all(), allow_blank=False)
    second_cur = QuerySelectField(query_factory=lambda: Currency.query.all(), allow_blank=True)
    fav_curs = QuerySelectMultipleField(query_factory=lambda: Currency.query.all(), allow_blank=True)
    password = PasswordField('Password', validators=[DataRequired(), EqualTo('confirm_password', message='Passwords
    confirm_password = PasswordField('Confirm Password', validators=[DataRequired()])
    profile_picture = FileField('Profile Picture', validators=[FileAllowed(['jpg', 'png'], 'Only images with jpg and
    submit = SubmitField('Sign Up')
```

Figure 5.2.8: Code for Frontend Sign Up Validation

Similarly, the validation logics are applied on both frontend and backend to make sure the user's data required are filled and also to display error messages if there is any logic error from the backend side like existing email, and same currency entered for default currency and second currency.

Figure 5.2.9: Account Verification Email from System

After completing registration, an activation email will be sent to the user's inbox for verification purpose. Upon user's activation, user will be redirected to the login page. The figure below shows the code for generating the token dedicated for account verification using serializer from external libraries.

```
    serializer = current_app.config['SERIALIZER']
    token = serializer.dumps(email, salt='verify-account')
    verify_account_url = current_app.config['URL_DOMAIN_WITH_PROTOCOL'] + url_for('auth.verify_account', token=token, external=True)
    html_template = render_template('mails/verify_account_mail.html', verify_account_url=verify_account_url)
    send_verify_account_mail.delay(email, html_template)
    flash('Sign up successful. You may check your email inbox to verify your account.', 'success')
    return redirect(url_for('auth.login'))
```

Figure 5.2.10: Code for Generating Activation Token in Sign Up Function

```
@auth_bp.route('/verify-account/<token>', methods=['GET'])
def verify_account(token):
    serializer = current_app.config['SERIALIZER']
    try:
        email = serializer.loads(token, salt='verify-account', max_age=1800)
        user = User.query.filter_by(email=email).first()
        if user:
            user.email_verified = True
            db.session.commit()
            flash('Your account has been verified. You can proceed to log in to the system.', 'success')
            return redirect(url_for('auth.login'))
        else:
            abort(404)
    except SignatureExpired:
        flash('Verification link has expired. Please sign up again.', 'danger')
        return redirect(url_for('auth.login'))

    except BadSignature:
        abort(404)
```

Figure 5.2.11: Code for Verifying Activation Token

5.2.3 Reset Password

If users forget their password of their account on ZyCurrency, they can navigate to the forget password screen from the login screen for resetting their password.

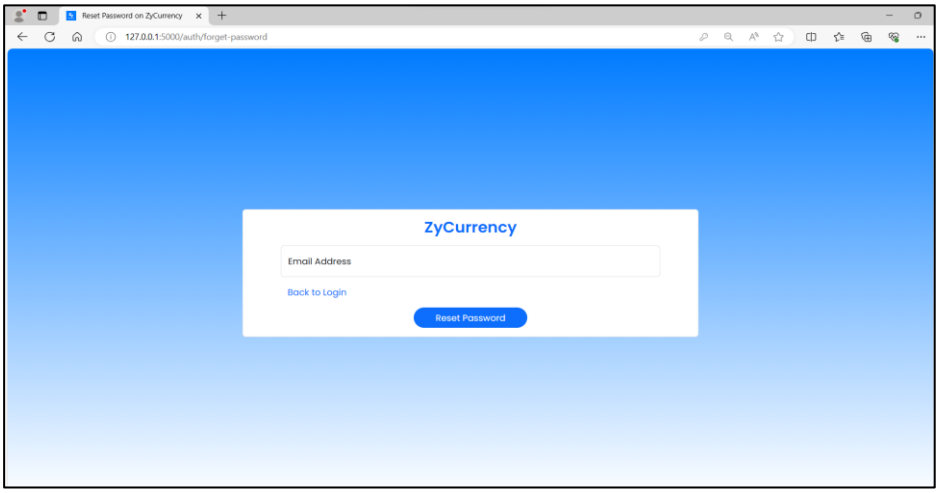


Figure 5.2.12: Forget Password Page

Once the users submitted the email binding with the lost account, a mail will be sent to the email inbox which contains the token generated for resetting password similar to the process of activating their account.

Figure 5.2.13: Password Reset Email from System

By clicking on the button, user will be redirected to the system's reset password page, which is inaccessible without the presence of the token for resetting password.

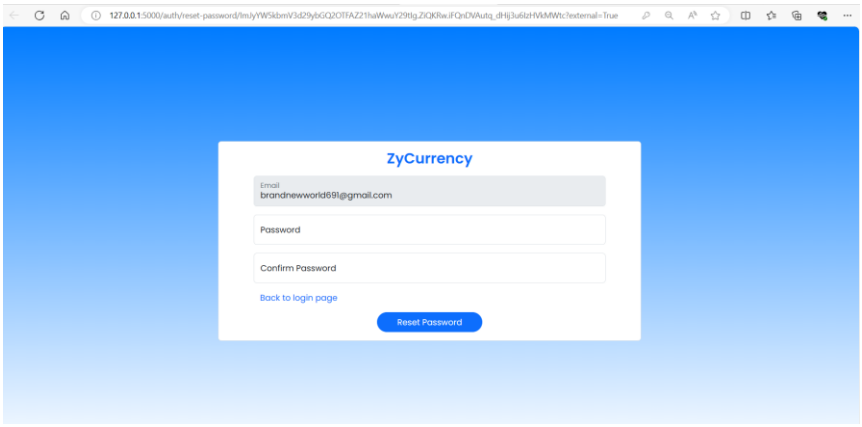


Figure 5.2.14: Reset Password Page

Validation is also done on the form to ensure users entering matching passwords.

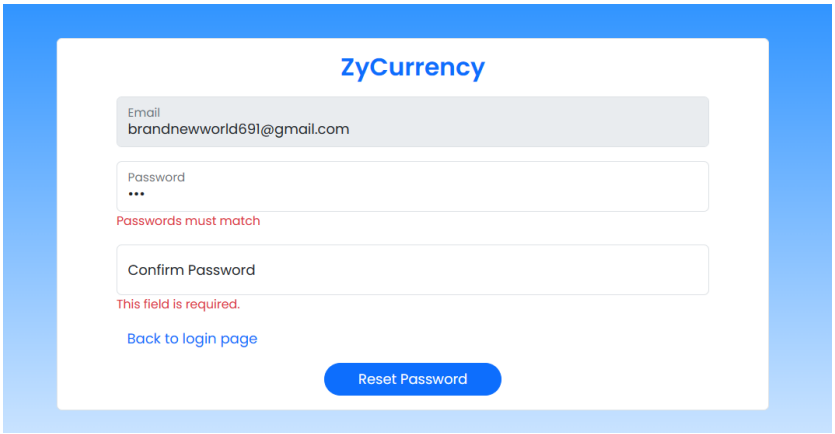


Figure 5.2.15: Reset Password Page with Frontend Validation

```
class ResetPasswordForm(FlaskForm):
    email = StringField('Email', [
        validators.DataRequired(),
        validators.Email(message='Invalid email address')
    ])
    password = PasswordField(' Password', [
        validators.DataRequired(),
        validators.EqualTo('confirm_password', message='Passwords must match')
    ])
    confirm_password = PasswordField('Confirm Password', [
        validators.DataRequired(),
        validators.EqualTo('password', message='Passwords must match')
    ])
```

Figure 5.2.16: Code for Frontend Validation on Resetting Password

Upon successful reset of the password, flash message will be used to notify about the password changes.

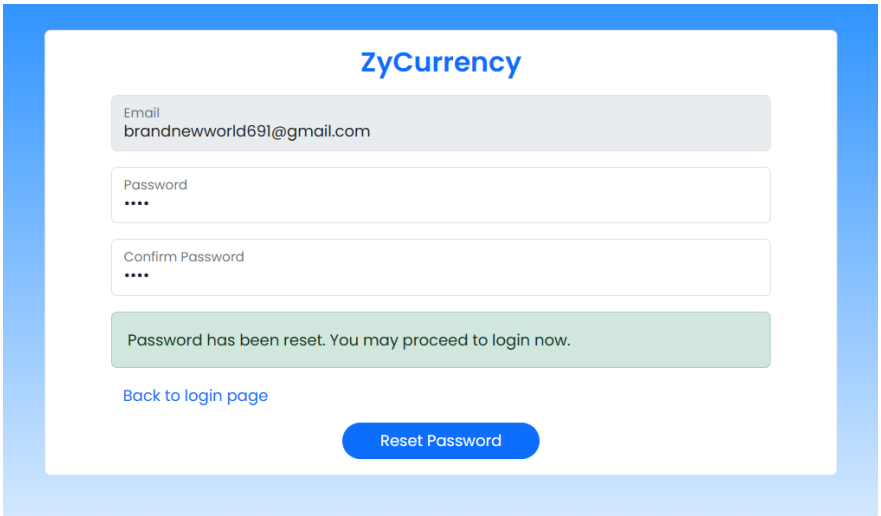


Figure 5.2.17: Success Message upon Password Reset

5.2.4 Google Authentication

Users can choose to sign in using their Google accounts on the login page.

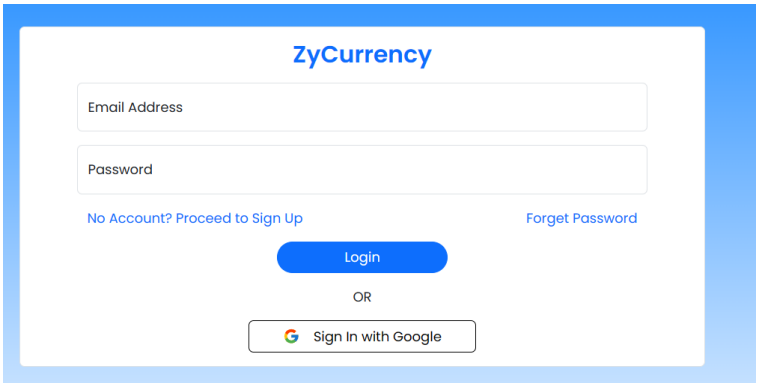


Figure 5.2.18: Google Sign In Button on Login Page

Next, the OAuth flow will be initiated, and user will be redirected to the Google authentication page. Upon receiving the OAuth callback, the application retrieves the authorization code from the callback URL and exchanges it for an access token from Google. Using this token, the system fetches user information, including email, name, and profile picture, from Google's user info endpoint.

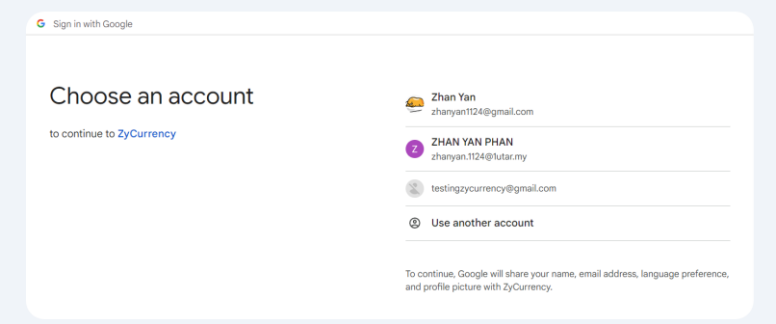


Figure 5.2.19: Google Authentication Page

```python
@auth_bp.route('/google_login')
def google_login():
    client_id, client_secret, google_url = get_google_credentials()
    client = WebApplicationClient(client_id)
    google_provider_cfg = get_google_provider_cfg(google_url)
    authorization_endpoint = google_provider_cfg["authorization_endpoint"]
    request_uri = client.prepare_request_uri(
        authorization_endpoint,
        redirect_uri=request.base_url + "/callback",
        scope=["email", "profile"],
    )
    return redirect(request_uri)
```

Figure 5.2.20: Code for Google OAuth Initiation

```python
@auth_bp.route('/google_login/callback')
def google_login_callback():
    client_id, client_secret, google_url = get_google_credentials()
    client = WebApplicationClient(client_id)
    code = request.args.get("code")
    google_provider_cfg = get_google_provider_cfg(google_url)
    token_endpoint = google_provider_cfg["token_endpoint"]
    token_url, headers, body = client.prepare_token_request(
        token_endpoint,
        authorization_response=request.url,
        redirect_url=request.base_url,
        code=code
    )
    token_response = requests.post(
        token_url,
        headers=headers,
        data=body,
        auth=(client_id, client_secret),
    )
    client.parse_request_body_response(json.dumps(token_response.json()))

    userinfo_endpoint = google_provider_cfg["userinfo_endpoint"]
    uri, headers, body = client.add_token(userinfo_endpoint)
    userinfo_response = requests.get(uri, headers=headers, data=body)
```

Figure 5.2.21: Code for Google OAuth Callback Handling

Once user's Google account is verified, user will be redirected to add profile page to provide some additional information that are not available from Google profile like the nationality and currency preference.

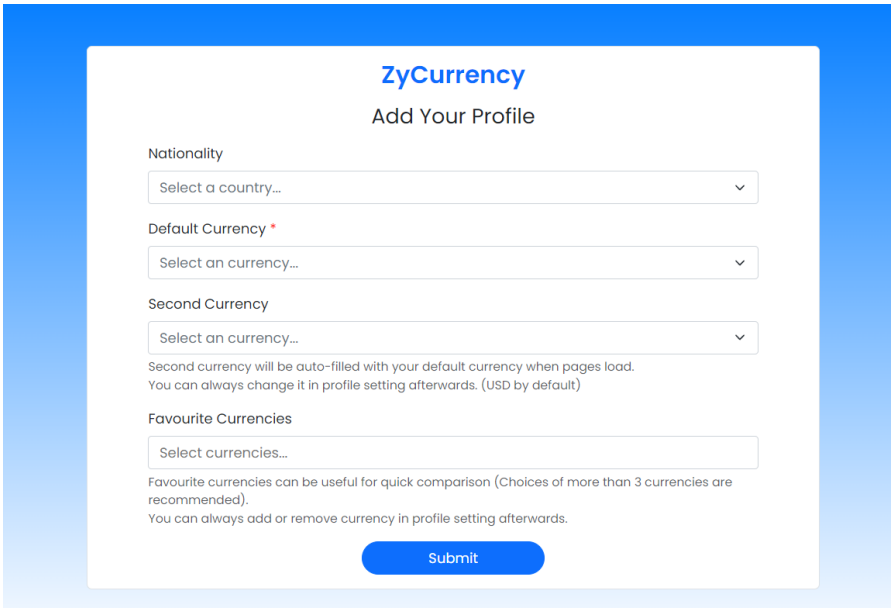


Figure 5.2.22: Add Profile Page

Input validation is also done to ensure the default currency and user's secondly preferred currency do not conflict.

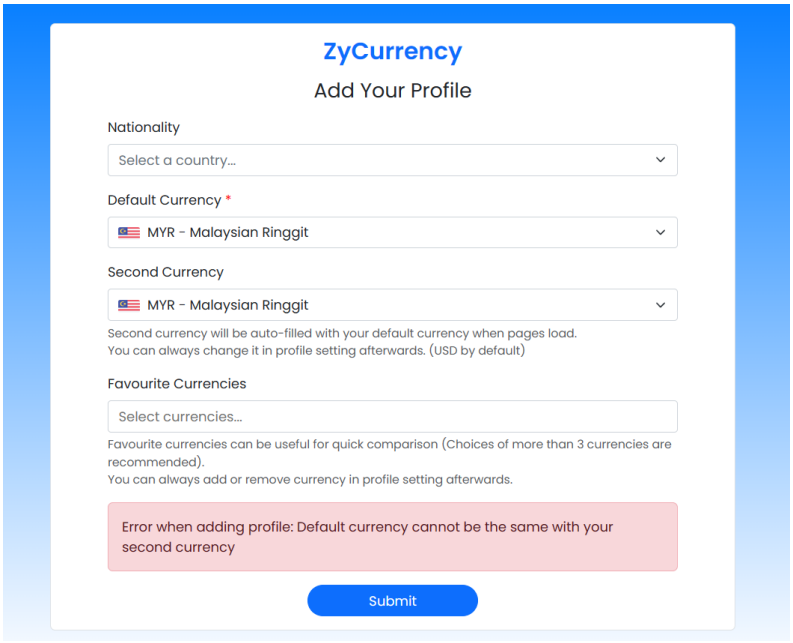


Figure 5.2.23: Validation on Adding Profile

5.3 User Profile Module

5.3.1 Edit Profile

Users can choose to edit their profile by pressing at their profile picture or name section after they login to the system.

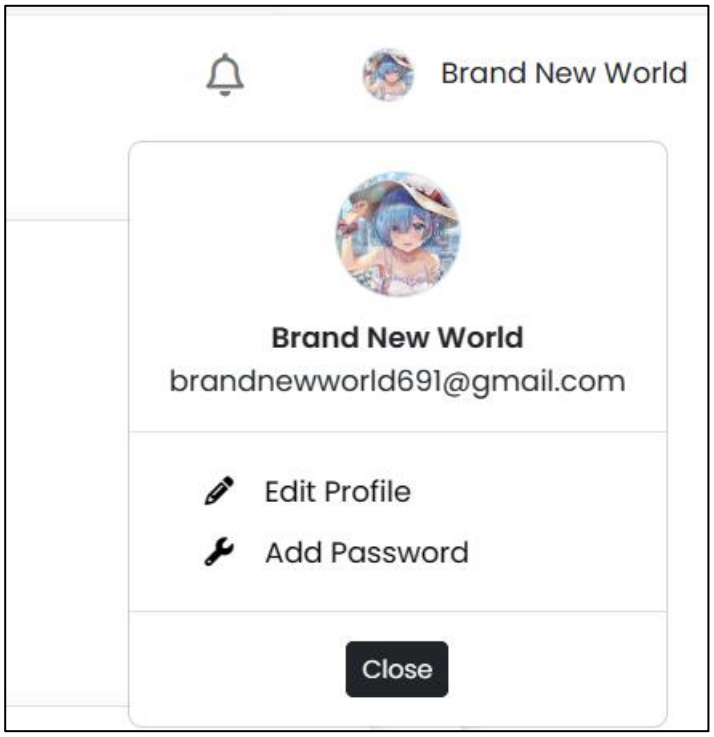


Figure 5.3.1: Edit Profile Button on Home Page

Upon pressing the button, users will be navigating to the edit profile page to order to change their profile details like profile picture, name, nationality and currency preferences.

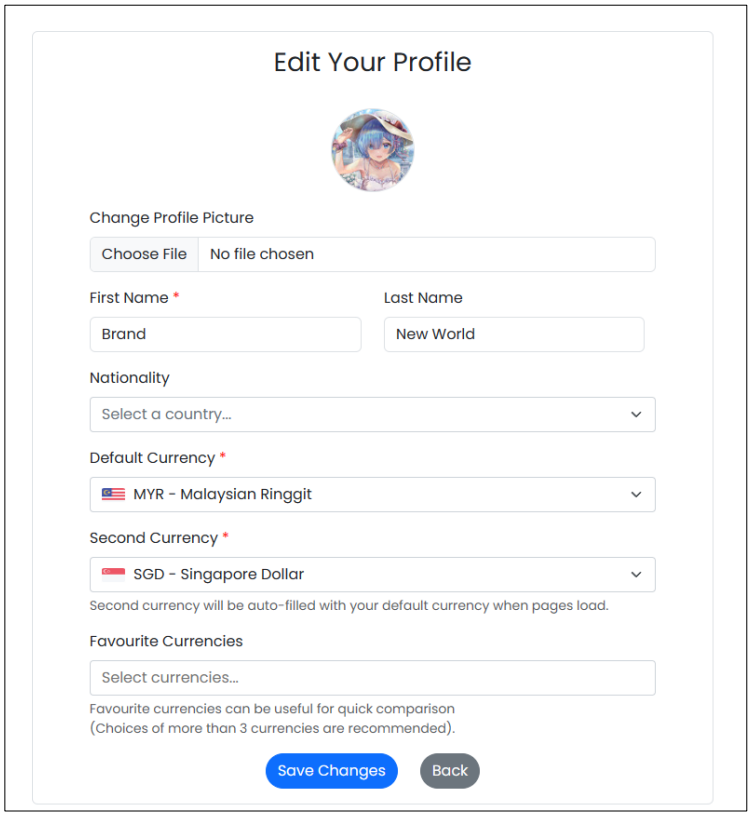


Figure 5.3.2: Edit Profile Page

Users input validation is also done to ensure the necessary data are filled in and they are in a correct format.

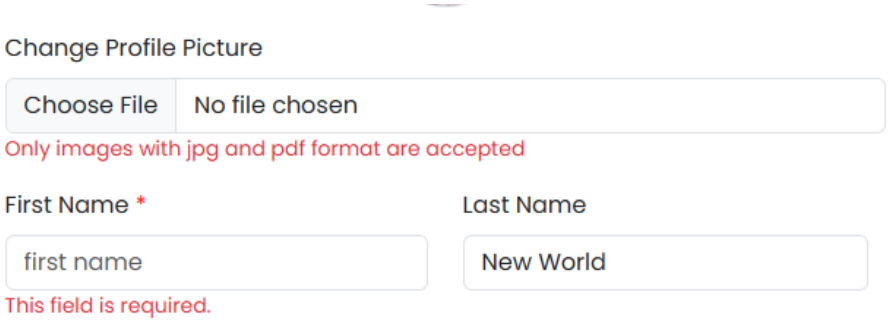


Figure 5.3.3: Frontend Validation of Edit Profile Page

```
class EditProfileForm(FlaskForm):
    profile_picture = FileField('Profile Picture', validators=[
        FileAllowed(['jpg', 'png'], 'Only images with jpg and pdf format are accepted')
    ])
    first_name = StringField('First Name', [validators.DataRequired()])
    last_name = StringField('Last Name')
    nationality = QuerySelectField(query_factory=lambda: Country.query.all(), allow_blank=True, get_label="name")
    default_cur = QuerySelectField(query_factory=lambda: Currency.query.all(), allow_blank=False)
    second_cur = QuerySelectField(query_factory=lambda: Currency.query.all(), allow_blank=False)
    fav_curs = QuerySelectMultipleField(query_factory=lambda: Currency.query.all(), allow_blank=True)
    submit = SubmitField('Edit')
```

Figure 5.3.4: Code for Frontend Validation on Edit Profile Page

Upon saving changes of user profile successfully in the system, a flash message will be shown to notify users.

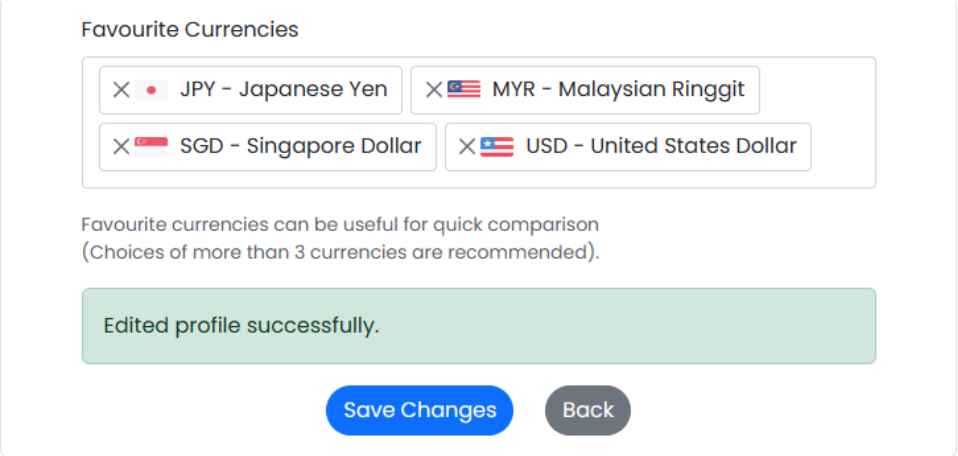


Figure 5.3.5: Success Message upon Saving Profile Changes

5.3.2 Add / Change Password

For users who sign up with their Google account, users can choose to add passwords to their account so they will have an extra option to login with email and password next time.

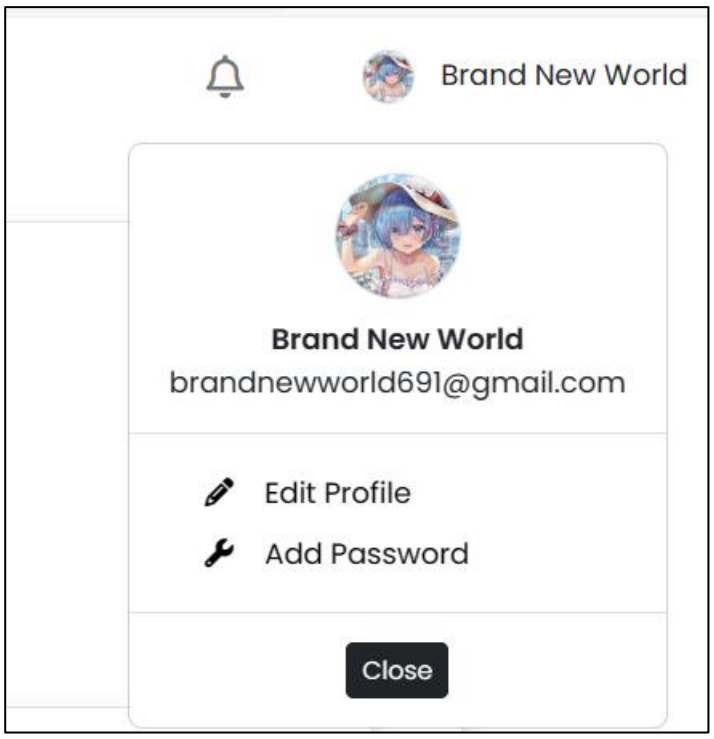


Figure 5.3.6: Add Password Button on Home Page

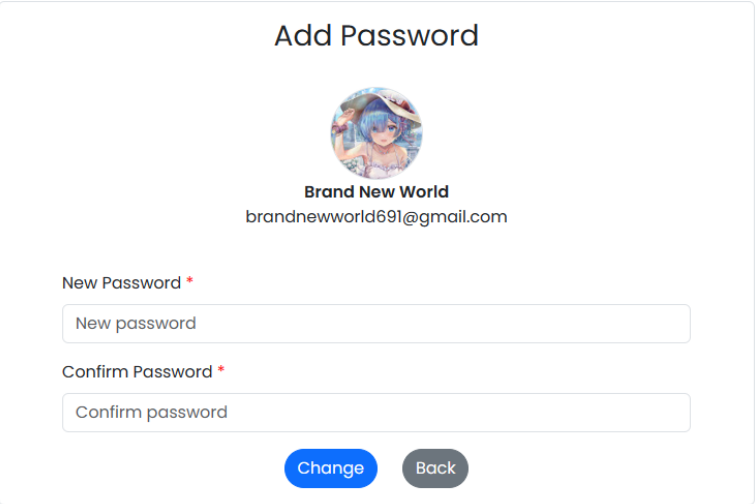


Figure 5.3.7: Change Password Page for Google OAuth Users

On the other hand, for users who already have an existing password binding with the account, they can choose to change their password on this page.

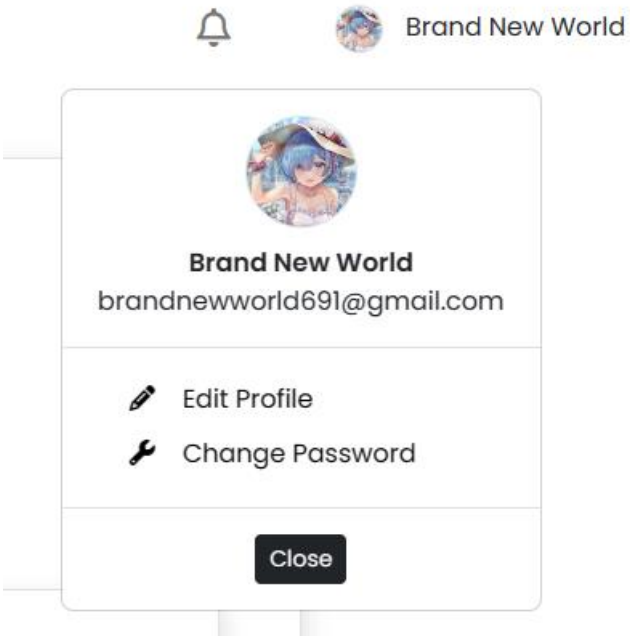


Figure 5.3.8: Change Password Button on Home Page

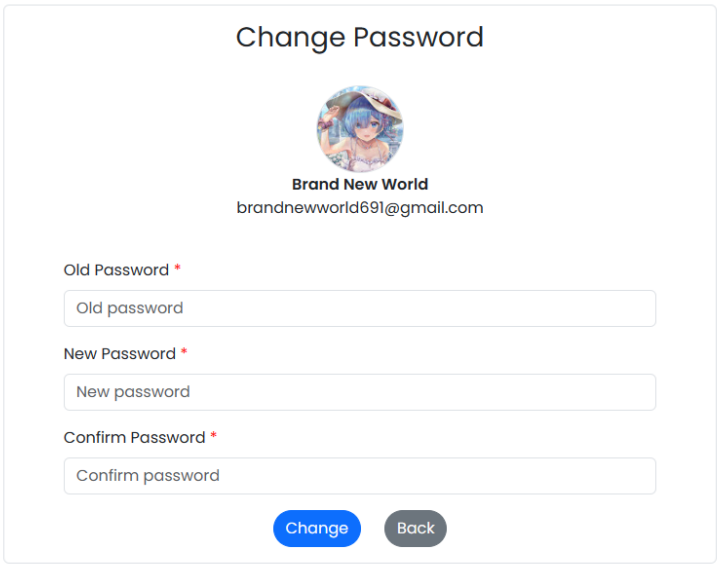


Figure 5.3.9: Change Password Page for User Accounts with Password

If user is having existing password and they wish to change to the new one, they are requested to input their old password for verification purpose.

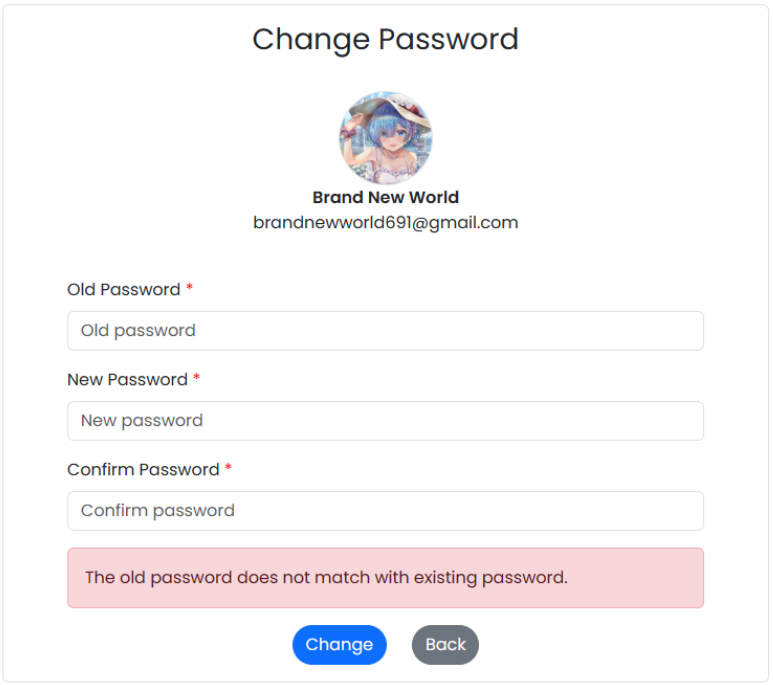


Figure 5.3.10: Validation on Change Password Page

```python
class ChangePasswordForm(FlaskForm):
    old_password = PasswordField('Old Password')
    new_password = PasswordField(' NewPassword', [
        validators.DataRequired(),
        validators.EqualTo('confirm_password', message='Passwords must match')
    ])
    confirm_password = PasswordField('Confirm Password', [
        validators.DataRequired(),
        validators.EqualTo('new_password', message='Passwords must match')
    ])
```

Figure 5.3.11: Code for Frontend Validation on Change Password Page

```python
@auth_bp.route('/change-password', methods=['GET', 'POST'])
@login_required
def change_password():
    form = ChangePasswordForm()
    try:
        if request.method == 'POST' and form.validate_on_submit():
            if(current_user.password):
                if(form.old_password.data == None):
                    raise InvalidPasswordException ('Old password is required for validation.')
                if(check_password_hash(current_user.password, form.old_password.data) == False):
                    raise InvalidPasswordException ('The old password does not match with existing password.')
                if(check_password_hash(current_user.password, form.new_password.data)):
                    raise InvalidPasswordException ('You are using an old password')
            hashed_password = generate_password_hash(form.new_password.data, method='scrypt')
            current_user.password = hashed_password
            db.session.commit()

            flash('Changed password successfully.', 'success')
```

Figure 5.3.12: Code for Backend Validation on Change Password Page

Upon changing password successfully in the system, a flash message will be shown to notify users.

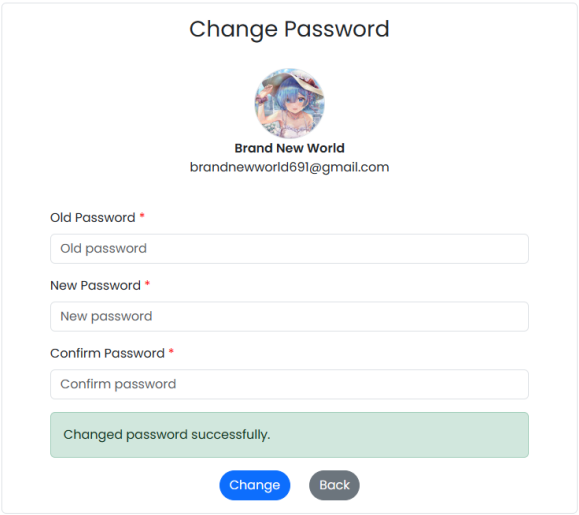


Figure 5.3.13: Success Message upon Changing Passwords

5.4 Currency Tracking Module

5.4.1 Currency Dashboard

The currency dashboard screen will be the first screen presented to the users once they successfully login to the system. The currency dashboard will be consisting of three parts which are the currency pair's overview section, popular currencies comparison section and favourite currencies comparison section respectively.

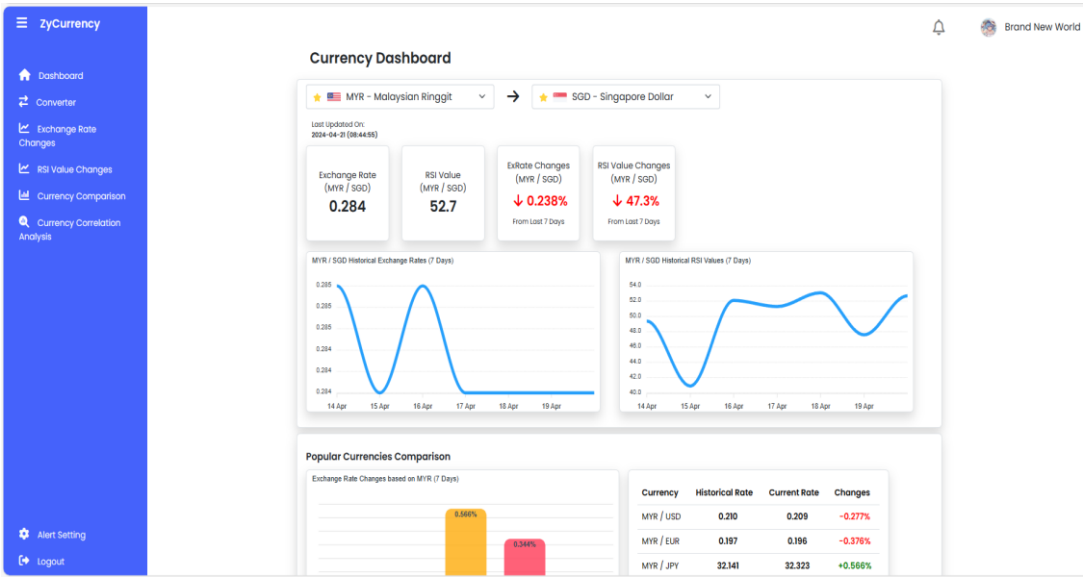


Figure 5.4.1: Currency Dashboard Page

This section will be consisting of the latest exchange rate, latest relative strength index (RSI) value, updated time, overall changes and line charts over past seven days. Furthermore, user can select their preferred currency pair by clicking on the currencies input. By default, default currency set by user will be used as the base currency while second currency of the user will be used as the comparing currency.

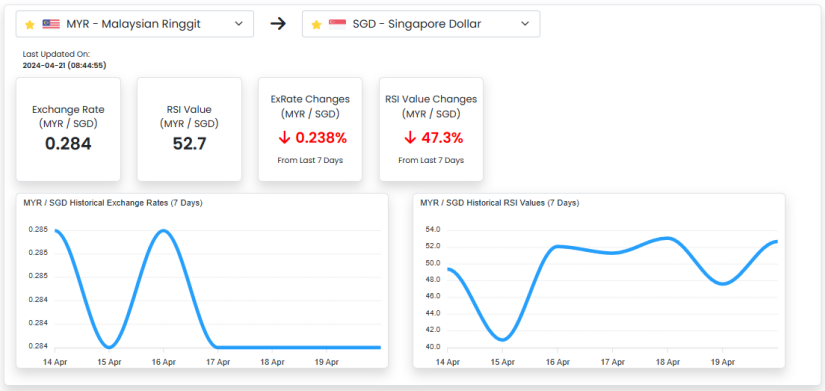


Figure 5.4.2: Currency Pair Overview Section on Dashboard Page

The currencies selection inputs are built in with searching function for the name or code to quickly locate the preferred currency which will save up the users' time. Besides that, users' favourite currencies will be placed on top of other currencies with an additional star icon to enforce quick selection.

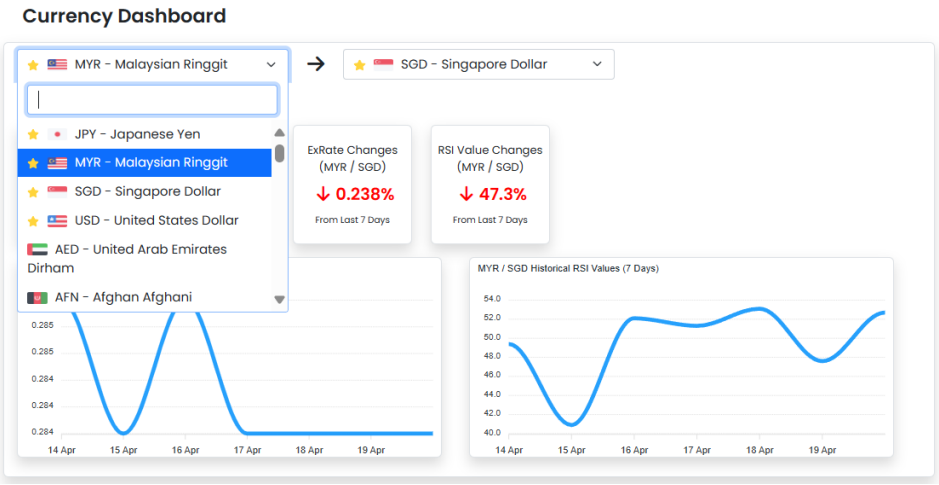


Figure 5.4.3: Currency Selection on Dashboard Page

To customize the currency selection input, a library named select 2 is being used for displaying the flag image and the star icon in the option field due to the plain html code has limited the flexibility of customization.

```
function currencySelectionOption(option) {
    if (!option.id) {
    return option.text;
    }

    var imageUrl = $(option.element).data("image");
    if (!imageUrl) {
        return option.text;
    }

    var fav = $(option.element).data("fav")
    if (fav){
        return $(
            '<span> <img src="' + fav +'" style="width:15px;height:15px;margin-right:10px" /><img src="' +
                imageUrl +
                '" class="img-flag" style="width:25px;height:15px;margin-right:5px"/> ' +
                option.text +
                '</span> '
            );
    }
    else{
        return $(
        '<span><img src="' +
            imageUrl +
            '" class="img-flag" style="width:25px;height:15px;margin-right:5px"/> ' +
            option.text +
            "</span>"
        );
    }
};
```

Figure 5.4.4: Code for Currencies Input Customization

```
//Use Select2 package to define the select inputs
$("#from_cur").select2({
    theme: "bootstrap-5",
    templateResult: currencySelectionOption,
    templateSelection: currencySelectionOption,
    width: "30%",
});

$("#to_cur").select2({
    theme: "bootstrap-5",
    templateResult: currencySelectionOption,
    templateSelection: currencySelectionOption,
    width: "30%",
});
```

Figure 5.4.5: Code for Applying Select2 Package on Currencies Input

The charts embedded in dashboard screen are being displayed interactively so it will show the particular detail for that data point including the date, currency pair and the exchange rate when the cursor hovers towards the specific point in exchange rate chart.

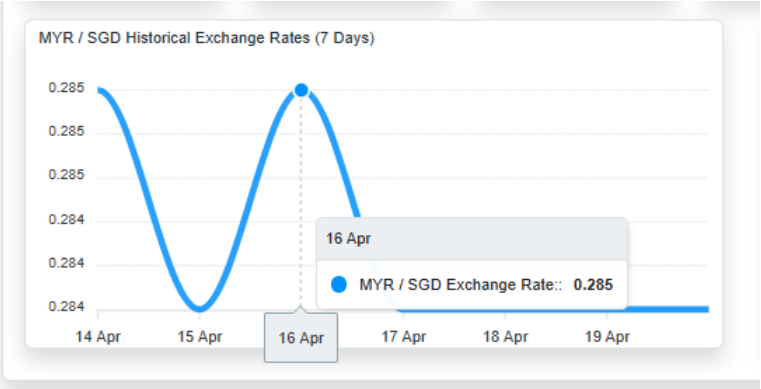


Figure 5.4.6: Line Chart on Dashboard Page

To achieve this design and interactivity embedded in the line chart, a JavaScript library named ApexCharts.js is being imported into the project and used to customize the line chart.

```
var lineChartOptions = {
    chart: {
        type: "line",
        animations: {
            initialAnimation: {
            enabled: false
            }
        },
        toolbar: {
            show: false,
            tools: {
                download: false,
                selection: false,
                zoom: false,
                zoomin: false,
                zoomout: false,
                pan: false,
                reset:false
            },
        },
    },
    stroke: {
        curve: 'smooth',
    },
    title: {
        align: 'left',
        margin: 10,
        offsetX: 0,
        offsetY: 0,
        floating: false,
        style: {
            fontSize:  '12px',
            fontWeight:  'normal',
            color:  '#263238'
        },
    },
    dataLabels: {
        enabled: false
    },
    noData: {
        text: 'Requesting Data...'
    },
    series: [],
    xaxis: {
        type: "datetime",
    },
    yaxis:{
        labels: {
            show: true,
        }
    },
    grid: {
        show: true,
        strokeDashArray: 1,
    }
};
```

Figure 5.4.7: Code for Line Chart Customization

```
var exchange_chart = new ApexCharts(document.querySelector("#exchangeChart"), lineChartOptions);
exchange_chart.render();
var rsi_chart = new ApexCharts(document.querySelector("#rsiChart"), lineChartOptions);
rsi_chart.render();
```

Figure 5.4.8: Code for Line Chart Initialization with ApexCharts.js

Moving on the next section which is the popular currencies comparison section. A bar chart and a table will be used to display the comparison of a list of popular currencies based on the default currency of the user. The popular currencies are fixed in the system including USD, EUR, JPY, GBP and lastly CNY.

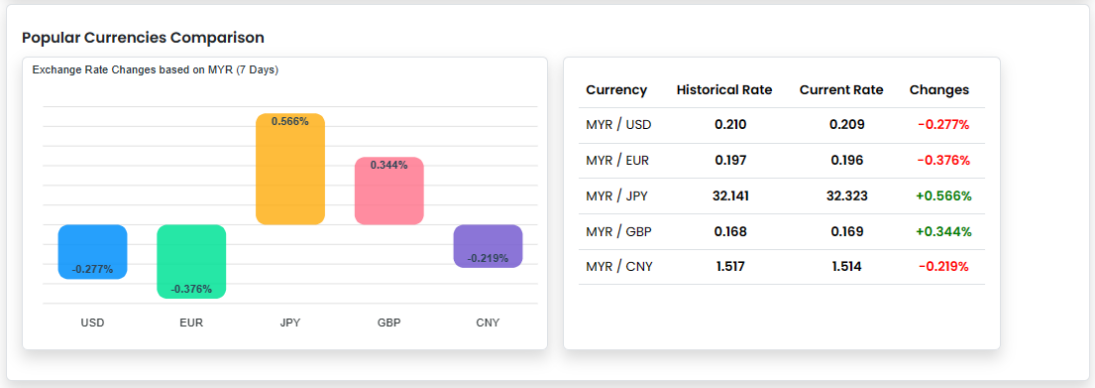


Figure 5.4.9: Popular Currencies Section on Dashboard Page

The last section of the dashboard screen will be the favourite currencies comparison based on the default currency of user. This section will only appear if users have set more than one favourite currencies in their user profiles. Similarly to the previous section, a bar chart and also a table will be used for displaying the result for one-glance comparison.

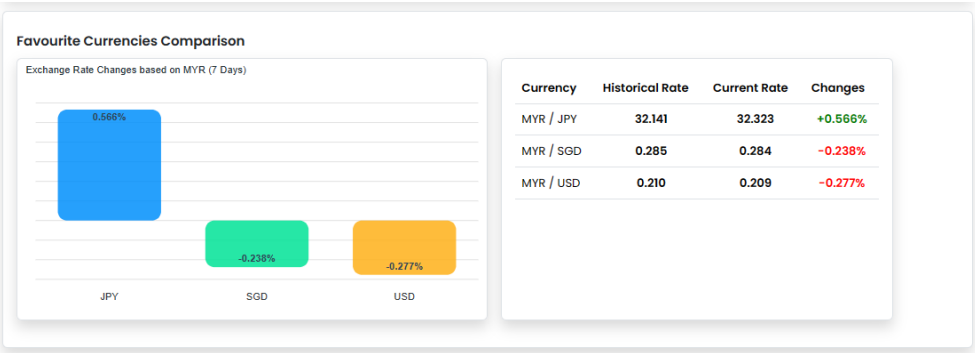


Figure 5.4.10: Favourite Currencies Section on Dashboard Page

The bar charts are also constructed interactively so the detail of the bar will be shown including the comparing currency, base currency, and changes for past seven days when the user hovers their cursor towards the bar.

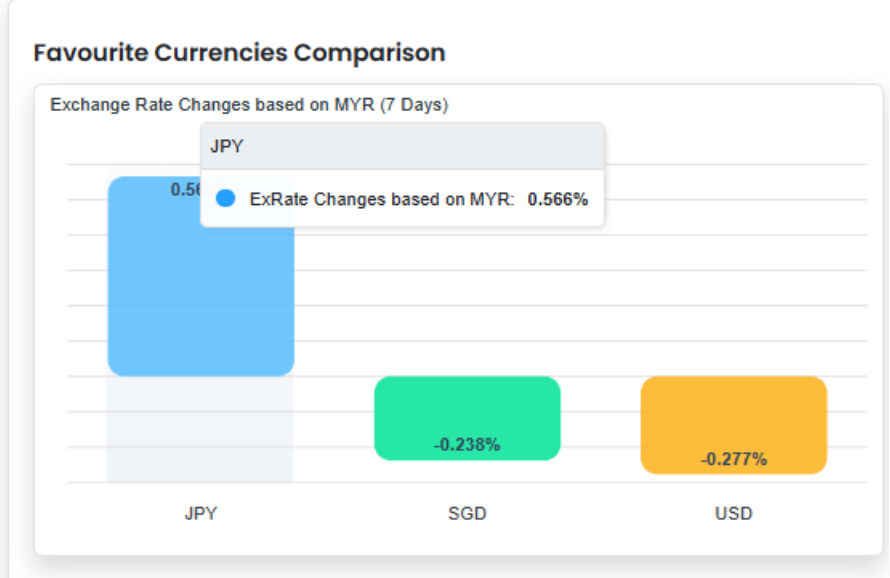


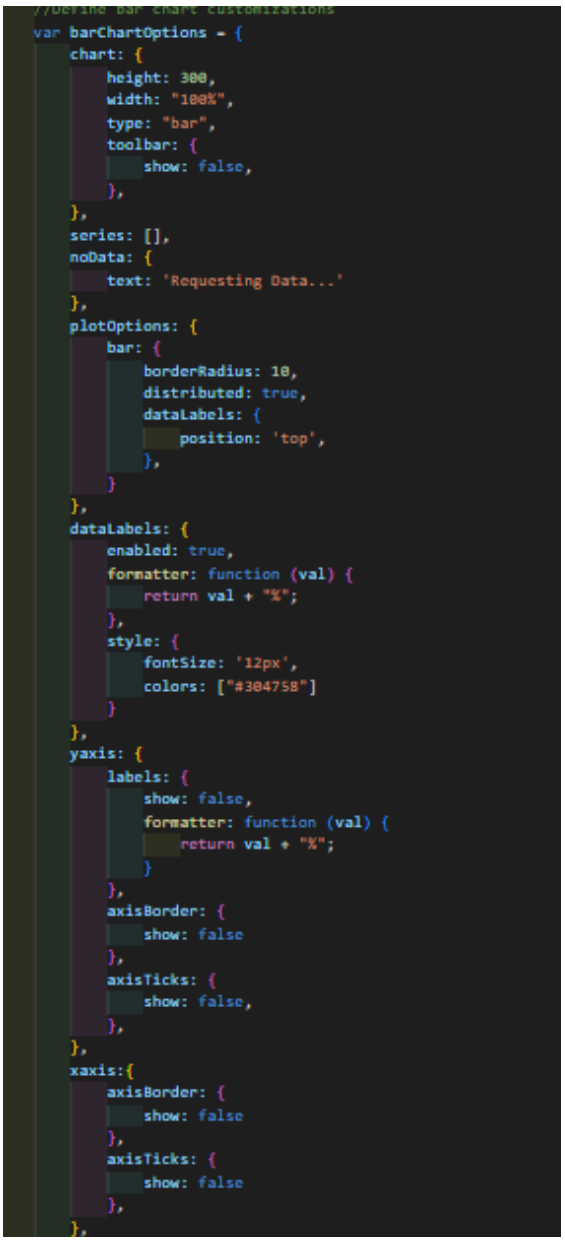
Figure 5.4.11: Bar Chart on Dashboard Page

To achieve this design and interactivity embedded in the bar charts, ApexCharts.js is being used to customize the bar charts.

```
var compare_by_popular_chart = new ApexCharts(document.querySelector("#comparisonByPopularChart"), barChartOptions);
compare_by_popular_chart.render();

var compare_by_fav_chart = new ApexCharts(document.querySelector("#comparisonByFavChart"), barChartOptions);
compare_by_fav_chart.render();
```

Figure 5.4.12: Code for Bar Charts Initialization with ApexCharts.js



```
//Define bar chart customizations
var barChartOptions = {
    chart: {
        height: 300,
        width: "100%",
        type: "bar",
        toolbar: {
            show: false,
        },
    },
    series: [],
    noData: {
        text: 'Requesting Data...'
    },
    plotOptions: {
        bar: {
            borderRadius: 10,
            distributed: true,
            dataLabels: {
                position: 'top',
            },
        }
    },
    dataLabels: {
        enabled: true,
        formatter: function (val) {
            return val + "%";
        },
        style: {
            fontSize: '12px',
            colors: ["#304758"]
        }
    },
    yaxis: {
        labels: {
            show: false,
            formatter: function (val) {
                return val + "%";
            }
        },
        axisBorder: {
            show: false
        },
        axisTicks: {
            show: false,
        },
    },
    xaxis:{
        axisBorder: {
            show: false
        },
        axisTicks: {
            show: false
        },
    },
```

Figure 5.4.13: Code for Bar Chart Customization

When the page initially loads, apex charts will be initialized and ajax requests will be sent to the API endpoints in the Flask server to obtain the currencies data. DOM manipulation techniques are being used to modify the html elements once the response are received from the API endpoints.

```
$.ajax({
    url: "/currency/latest-exrate",
    type: "POST",
    dataType: 'json',
    data: latestExRateFormData,
    success: function (response){
        console.log(response);
        $("#latest_rate").text(response.exchange_rate.toFixed(3));
        $(".updated_date").text(response.updated_date + "\n (" + response.updated_time + ") ");
        $(".currency_pair").text("("+ from_cur + " / " + to_cur +")")
    },
    error:function (xhr, status, error) {
        console.error(xhr.responseText);
    },
});
```

Figure 5.4.14: Code for Retrieving Latest Exchange Rate through Ajax

```
$.ajax({
    url: "/currency/historical-exrate",
    type: "POST",
    dataType: 'json',
    data: historicalFormData,
    success: function (response) {
        console.log(response);
        fillExRateChanges(response.changes);
        const exchange_chart_data = convertObjectToList(response.results)
        exchange_chart.updateSeries([{
            name: $("#from_cur").val() + " / " + $("#to_cur").val() + " Exchange Rate:",
            data: exchange_chart_data
        }])
        exchange_chart.updateOptions({
            title: {
                text: $("#from_cur").val() + " / " + $("#to_cur").val() + " Historical Exchange Rates (7 Days)",
            }
        })
    },
    error: function (xhr, status, error) {
        console.error(xhr.responseText);
    },
});
```

Figure 5.4.15: Code for Retrieving Historical Exchange Rates through Ajax

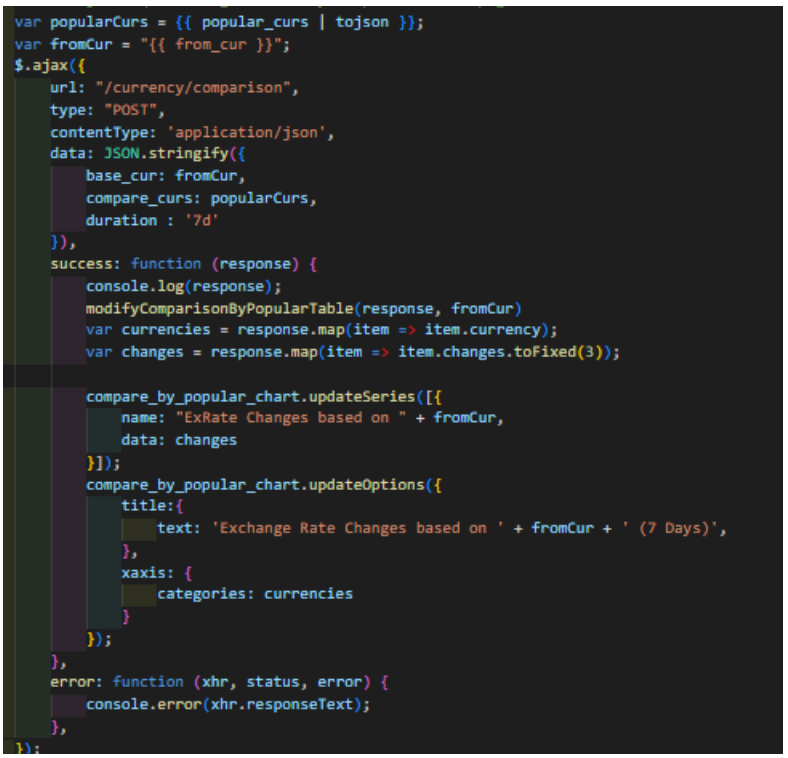
```
$.ajax({
    url: "/currency/historical-rsi",
    type: "POST",
    data: historicalFormData,
    success: function (response) {
        console.log(response);
        fillRSIChanges(response.changes)
        var rsi_values = response.rsi_values;
        var dates = response.dates;

        latest_rsi = rsi_values[rsi_values.length - 1]
        $("#latest_rsi").text(latest_rsi.toFixed(1))

        var rsi_chart_data = rsi_values.map(function(value, index) {
            return [dates[index], value.toFixed(1)];
        });

        rsi_chart.updateSeries([{
            name: $("#from_cur").val() + " / " + $("#to_cur").val() + " RSI Value: ",
            data: rsi_chart_data
        }])
        rsi_chart.updateOptions({
            title: {
                text: $("#from_cur").val() + " / " + $("#to_cur").val() + " Historical RSI Values (7 Days)",
            },
            annotations: {
                yaxis: [
                    {
                        y: 70,
                        strokeDashArray: 1,
                        borderColor: 'black',
                        label: {
                            borderColor: 'black',
                            style: {
                                color: '#fff',
                                background: 'black'
                            },
                            text: 'Overbought'
                        }
                    },
                    {
                        y: 30,
                        strokeDashArray: 1,
                        borderColor: 'black',
                        label: {
                            borderColor: 'black',
                            style: {
                                color: '#fff',
                                background: 'black'
                            },
                            text: 'Oversold'
                        }
                    },
                    ]
                }
        })
    },
```

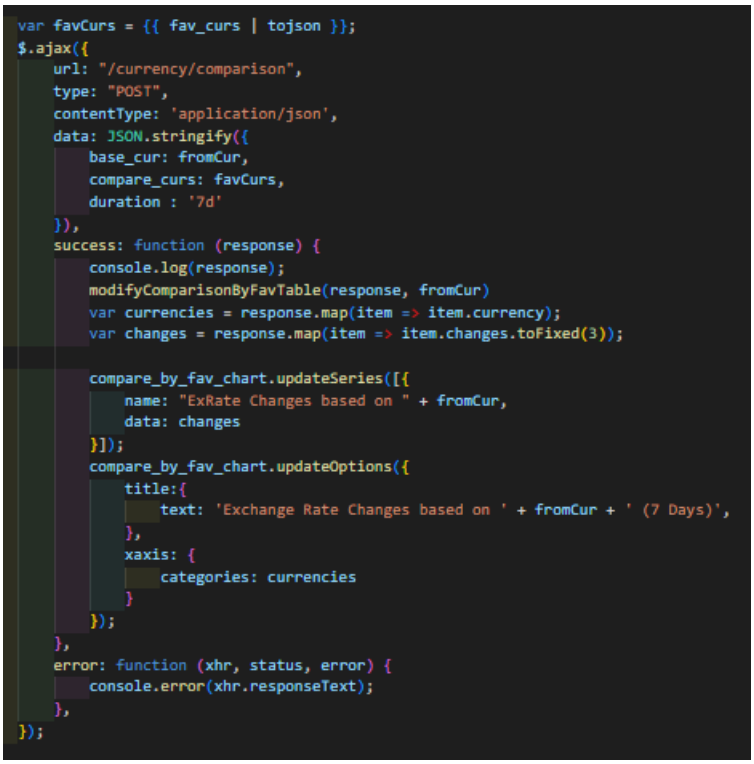
Figure 5.4.16: Code for Retrieving Historical RSI Values through Ajax



```
var popularCurs = {{ popular_curs | tojson }};
var fromCur = "{{ from_cur }}";
$.ajax({
    url: "/currency/comparison",
    type: "POST",
    contentType: 'application/json',
    data: JSON.stringify({
        base_cur: fromCur,
        compare_curs: popularCurs,
        duration : '7d'
    }),
    success: function (response) {
        console.log(response);
        modifyComparisonByPopularTable(response, fromCur)
        var currencies = response.map(item => item.currency);
        var changes = response.map(item => item.changes.toFixed(3));

        compare_by_popular_chart.updateSeries([{
            name: "ExRate Changes based on " + fromCur,
            data: changes
        }]);
        compare_by_popular_chart.updateOptions({
            title:{
                text: 'Exchange Rate Changes based on ' + fromCur + ' (7 Days)',
            },
            xaxis: {
                categories: currencies
            }
        });
    },
    error: function (xhr, status, error) {
        console.error(xhr.responseText);
    },
});
```

Figure 5.4.17: Code for Getting Popular Currencies Comparison through Ajax



```
var favCurs = {{ fav_curs | tojson }};
$.ajax({
    url: "/currency/comparison",
    type: "POST",
    contentType: 'application/json',
    data: JSON.stringify({
        base_cur: fromCur,
        compare_curs: favCurs,
        duration : '7d'
    }),
    success: function (response) {
        console.log(response);
        modifyComparisonByFavTable(response, fromCur)
        var currencies = response.map(item => item.currency);
        var changes = response.map(item => item.changes.toFixed(3));

        compare_by_fav_chart.updateSeries([{
            name: "ExRate Changes based on " + fromCur,
            data: changes
        }]);
        compare_by_fav_chart.updateOptions({
            title:{
                text: 'Exchange Rate Changes based on ' + fromCur + ' (7 Days)',
            },
            xaxis: {
                categories: currencies
            }
        });
    },
    error: function (xhr, status, error) {
        console.error(xhr.responseText);
    },
});
```

Figure 5.4.18: Code for Getting Favourite Currencies Comparison through Ajax

5.4.2 Currency Converter

Users can navigate to the currency converter page by clicking on the converter button located on the sidebar.

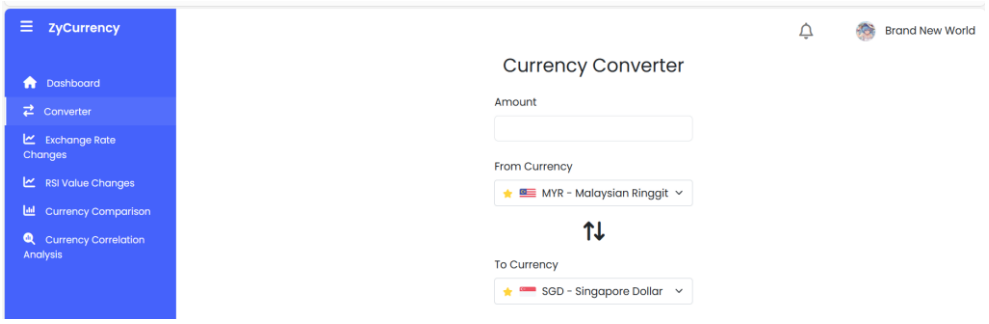


Figure 5.4.19: Currency Converter Page

When users input an amount, the ajax request will be sent to call the API endpoint in flask server to perform the conversion.

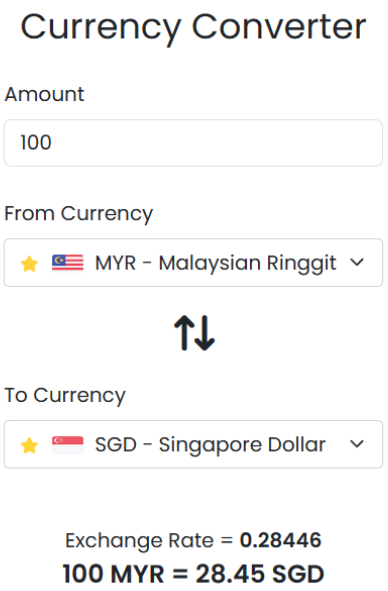


Figure 5.4.20: Result Displayed in Currency Converter Page

```javascript
$("#convert_form").submit(function (event) {
  event.preventDefault();

  var formData = $(this).serialize();

  $.ajax({
    url: "/currency/convert",
    type: "POST",
    data: formData,
    success: function (response) {
      console.log(response);
      var exchange_rate = response.exchange_rate;
      var convert_result = response.convert_result;
      $("#convert_result_section").show();
      var from_cur = $("#from_cur").val();
      var to_cur = $("#to_cur").val();
      var amount = $("#amount").val();

      $("#exchange_rate_display").text(exchange_rate);
      $("#result").text(amount + ' ' +  from_cur + ' = ' + convert_result + ' ' + to_cur);

    },
    error: function (xhr, status, error) {
      console.error(xhr.responseText);
    },
  });
});
```

Figure 5.4.21: Code for Retrieving Conversion Result through Ajax

Once the ajax request is called, the Flask server will be sending a request to external API from Fast Forex to retrieve the conversion result by passing in the currencies and amount as parameter. Eventually, the results or error messages will be returned to the frontend as the response of the request.

```python
@currency_bp.route('/convert', methods=['POST'])
@login_required
def retrieve_convertion():
    try:
        from_cur = request.form.get('from_cur')
        to_cur = request.form.get('to_cur')
        amount=request.form.get('amount')

        if from_cur is None or to_cur is None:
            raise DataMissingException('Select currency to proceed')

        elif amount is None:
            raise DataMissingException('Please input amount')

        headers = {"accept": "application/json"}
        convert_url = "{}/convert?api_key={}&from={}&to={}&amount={}".format(current_app.config['FAST_FOREX_API_URL']
        , current_app.config['FAST_FOREX_API_KEY'], from_cur, to_cur, amount)
        response = requests.get(convert_url, headers=headers)

        if response.status_code == 200:
            json_obj = response.json()
            print(json_obj)
            result= json_obj['result']
            exchange_rate = result['rate']
            convert_result = result[to_cur]
            data = {
                "convert_result": convert_result,
                "exchange_rate": exchange_rate,
            }
            return jsonify(data), 200
        else:
            print(f"Error: {response.status_code} - {response.text}")
            return jsonify({'error': str(e)}), 400

    except DataMissingException as e:
        print(str(e))
        return jsonify({'error': str(e)}), 400

    except Exception as e:
        print(str(e))
        return jsonify({'error': str(e)}), 400
```

Figure 5.4.22: Code for Conversion API Endpoint

5.4.3 Currency Exchange Rate Tracking

Users can navigate to the currency exchange rate tracking page by clicking on the exchange rate changes button located on the sidebar. By default, the from currency will be set to user's default currency while to currency will be set to user's second currency. Duration will be set to one week by default which is the minimum duration supported by this system.

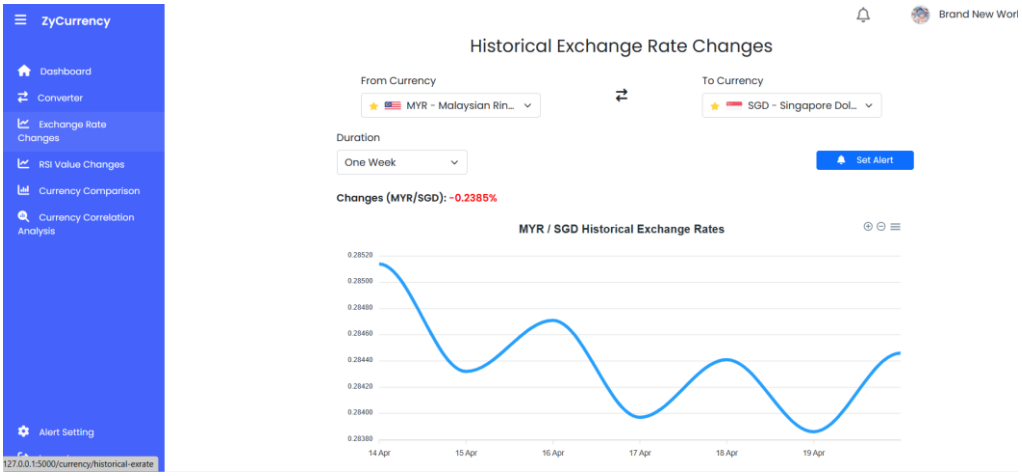


Figure 5.4.23: Currency Exchange Rate Changes Page

An ajax request will be sent to the historical exchange rates API endpoint in Flask server when the page loads initially. The chart and html elements are then updated once the response is received from the backend.

```
var formData = $("#get_historical_form").serialize();
$.ajax({
  url: "/currency/historical-exrate",
  type: "POST",
  data: formData,
  success: function (response) {
    console.log(response);
    fillExRateChanges(response.changes)
    const chart_data = convertObjectToList(response.results)
    const latest_rate = chart_data[chart_data.length-1][1];
    $('#current_rate').text(latest_rate.toFixed(4));
    $('#current_rate_section').show();

    chart.updateSeries([{
      name: $("#from_cur").val() + " / " + $("#to_cur").val() + " Exchange Rate:",
      data: chart_data
    }])
    chart.updateOptions({
      title: {
        text: $("#from_cur").val() + " / " + $("#to_cur").val() + " Historical Exchange Rates",
      }
    })

  },
  error: function (xhr, status, error) {
    console.error(xhr.responseText);
  },
```

Figure 5.4.24: Code for Getting Historical Exchange Rates through Ajax

Once the ajax request is called, the Flask server will be sending a request to external API from Fast Forex to retrieve the historical exchange rates by passing in the currencies and duration as parameter. Eventually, the results or error messages will be returned to the frontend as the response of the request.

```python
@currency_bp.route('/historical-exrate', methods=['POST'])
@login_required
def retrieve_historical_exchange_rate():
    try:
        from_cur = request.form.get('from_cur')
        to_cur = request.form.get('to_cur')
        duration=request.form.get('duration')
        end_date = datetime.now()
        if from_cur is None or to_cur is None:
            raise DataMissingException('Select currency to proceed')
        if duration is None:
            raise DataMissingException('Select duration to proceed')

        if duration.endswith('d'):
            start_date = end_date - timedelta(days=int(duration[:-1]))
        elif duration.endswith('m'):
            start_date = end_date - relativedelta(months=int(duration[:-1]))
        elif duration.endswith('y'):
            start_date = end_date - relativedelta(years=int(duration[:-1]))
        headers = {"accept": "application/json"}
        response=None

        # If currency for current date is not updated, thus need to use the previous date as latest currency
        while not response or 'future' in response.text:
            time_series_url = "{}/time-series?api_key={}&from={}&to={}&start={}&end={}".format(current_app.config['FAST_FOREX_API_URL'],
            current_app.config['FAST_FOREX_API_KEY'], from_cur, to_cur, start_date.strftime('%Y-%m-%d'), end_date.strftime('%Y-%m-%d'))
            response = requests.get(time_series_url, headers=headers)
            if response.status_code == 200:
                json_obj = response.json()
                results= json_obj['results']
                end_date = max(results[to_cur].keys())
                start_date = min(results[to_cur].keys())
                changes=(results[to_cur][end_date] - results[to_cur][start_date])/results[to_cur][start_date] * 100
                changes = round(changes, 4)
                data = {
                    "results": results,
                    "changes": changes
                }
                return jsonify(data), 200
            else:
                if 'future' in response.text:
                    end_date = end_date - timedelta(days=1)
                print(f"Error: {response.status_code} - {response.text}")
```

Figure 5.4.25: Code for Historical Exchange Rates API Endpoint

Whenever there are any changes of from currency, to currency or duration, the ajax request will be sent to the API endpoint again to cater the changed requirements.

Figure 5.4.26: Update after Changes on Duration

```
//Submit the form when data changes
$('#to_cur, #from_cur, #duration').change(function() {
  var fromCurValue = $('#from_cur').val();
  var toCurValue = $('#to_cur').val();
  var durationValue = $('#duration').val();
  if (fromCurValue != "" && toCurValue!="" && durationValue != "" && fromCurValue!=toCurValue)
    $('#get_historical_form').submit();
});
```

Figure 5.4.27: Code for Detecting Changes of User Input

5.4.4 Currency Relative Strength Index (RSI) Tracking

Users can navigate to the currency RSI values tracking page by clicking on the RSI value changes button located on the sidebar. By default, the from currency will be set to user's default currency while to currency will be set to user's second currency. Duration will be set to one week by default.

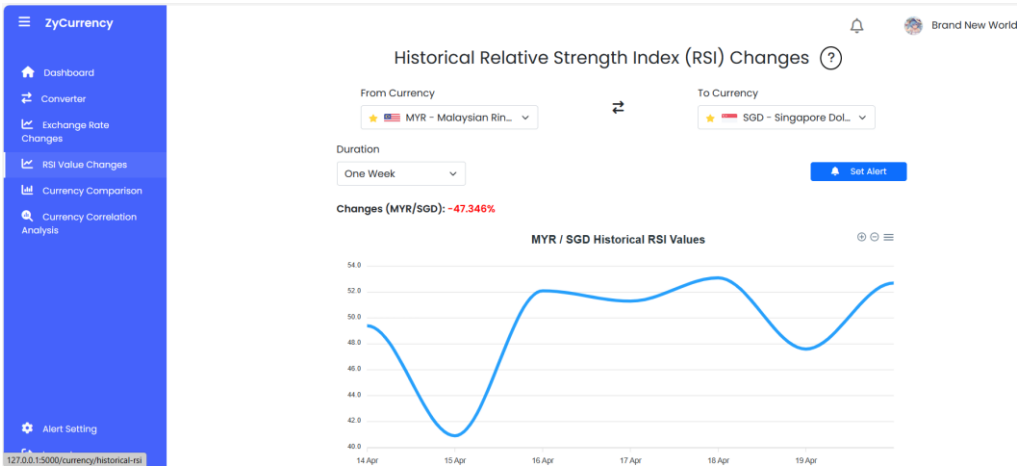


Figure 5.4.28: Currency RSI Values Changes Page

Similarly to exchange rate changes page, an ajax request will be sent to the historical RSI values API endpoint in Flask server when the page loads initially. The chart and html elements are then updated once the response is received from the backend.

```
var formData = $("#get_historical_form").serialize();
$.ajax({
  url: "/currency/historical-rsi",
  type: "POST",
  data: formData,
  success: function (response) {
    console.log(response);
    fillRSIChanges(response.changes);

    var rsi_values = response.rsi_values;
    latest_rsi = rsi_values[rsi_values.length-1];
    $("#current_rsi").text(latest_rsi.toFixed(1));
    $("#current_rsi_section").show();

    var dates = response.dates;

    var chart_data = rsi_values.map(function(value, index) {
        return [dates[index], value.toFixed(1)];
    });

    chart.updateSeries([{
      name: $("#from_cur").val() + " / " + $("#to_cur").val() + " RSI Value: ",
      data: chart_data
    }])
    chart.updateOptions({
      title: {
        text: $("#from_cur").val() + " / " + $("#to_cur").val() + " Historical RSI Values",
      }
    })
  },
  error: function (xhr, status, error) {
    console.error(xhr.responseText);
  },
});
```

Figure 5.4.29: Code for Getting Historical RSI Values through Ajax

However, due to the FastForex API does not have endpoint for retrieval of historical RSI values, the RSI value calculation had to be done in the system after historical exchange rates are received. The system will be using the exchange rates from past 14 days to support calculation of RSI values.

```
@currency_bp.route('/historical-rsi', methods=['POST'])
@login_required
def retrieve_historical_rsi_values():
    try:
        from_cur = request.form.get('from_cur')
        to_cur = request.form.get('to_cur')
        duration=request.form.get('duration')

        if from_cur is None or to_cur is None:
            raise DataMissingException('Select currency to proceed')
        if duration is None:
            raise DataMissingException('Select duration to proceed')

        changes, dates, rsi_values = get_rsi_value(from_cur, to_cur, duration)
        data = {
            "dates": dates,
            "rsi_values": rsi_values,
            "changes" : changes
        }
        return jsonify(data), 200

    except DataMissingException as e:
        print(str(e))
        return jsonify({'error': str(e)}), 400

    except Exception as e:
        print(str(e))
        return jsonify({'error': str(e)}), 400
```

Figure 5.4.30: Code for Historical RSI Values API Endpoint

$$RS = \frac{Avg.Gain}{Avg.Loss}$$

$$RSI = 100 - \frac{100}{1 + RS}$$

Figure 5.4.31: Formula of RSI Value Calculation (West, 2023)

```
def get_rsi_value(from_cur, to_cur, duration):

    time_series_url = "{}/time-series?api_key={}&from={}&to={}&start={}&end={}".format(current_app.config['FAST_FOREX_API_URL'],
    current_app.config['FAST_FOREX_API_KEY'], from_cur, to_cur, start_date.strftime('%Y-%m-%d'), end_date.strftime('%Y-%m-%d'))

    response = requests.get(time_series_url, headers=headers)

    if response.status_code == 200:
        json_obj = response.json()
        results= json_obj['results']
        dates =  list(list(results.values())[0].keys())
        exchange_rates = list(list(results.values())[0].values())
        # print("Dates:", list(dates))
        # print("Exchange Rates:", list(exchange_rates))

        changes = []
        for i in range(1, len(exchange_rates)):
            if exchange_rates[i] - exchange_rates[i - 1] != 0:
                change = (exchange_rates[i] - exchange_rates[i - 1])/exchange_rates[i-1] * 100
            else:
                change = 0
            changes.append(change)
        # print("Changes:", list(changes))

        rsi_values = []
        for i in range(14, len(changes)+1):
            gain = 0
            loss = 0
            for j in range(i-14, i):
                if  changes[j] > 0:
                    gain += changes[j]
                else:
                    loss -= changes[j]
            average_gain = gain / 14
            average_loss = loss / 14
            relative_strength = average_gain/average_loss
            rsi = 100-(100/(1+relative_strength))
            # print('Avg gain for '+ str(i) + ': ' + str(average_gain))
            # print('Avg loss for '+ str(i) + ': ' + str(average_loss))
            # print('RS for '+ str(i)  + ': ' + str(relative_strength))
            # print('RSI for '+ str(i)  + ': ' + str(rsi))
            rsi_values.append(rsi)
        changes = round(rsi_values[-1] - rsi_values[0] / rsi_values[0] * 100,4)
        dates = dates[14:]
        rounded_rsi_values = [round(rsi, 1) for rsi in rsi_values]
        return changes, dates, rounded_rsi_values
```

Figure 5.4.32: Code for Performing RSI Value Calculation

Similarly to exchange rate changes page, the system will also detect for changes in from cur, to cur and duration and perform update on the user interface correspondingly.

Figure 5.4.33: Update after Changes on Duration

5.4.5 Currency Changes Comparison

Users can navigate to the currency comparison page by clicking on the currency comparison button located on the sidebar. By default, the base currency will be set to user's default currency while the comparing currencies will be set to the popular currencies defined in the system. Duration will be set to one week by default.

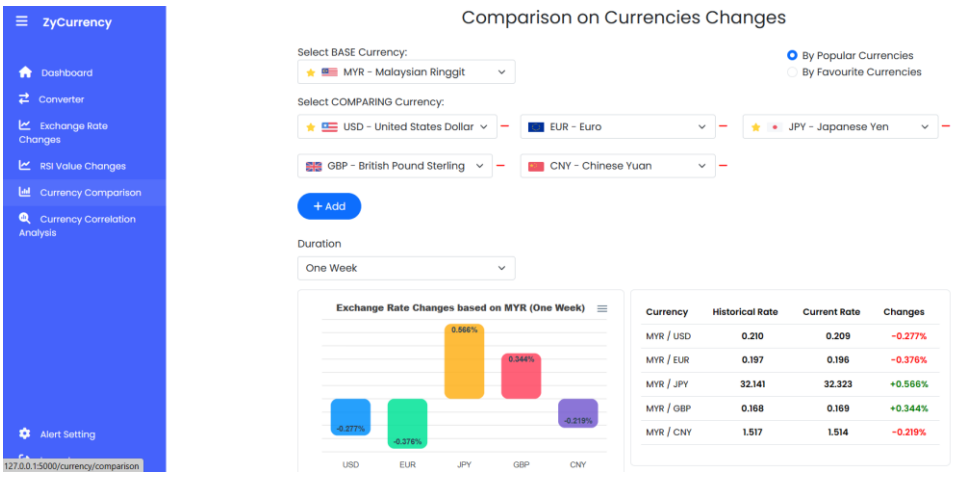


Figure 5.4.34: Currencies Comparison Page

An ajax request will be sent to the comparison API endpoint in Flask server when the page loads initially. The chart and html elements are then updated once the response is received from the backend.

```
//Send first ajax request when page initially load
var baseCur = $('#base_cur').val();
var compareCurValues = $('.compare_cur').map(function() {
    return $(this).val();
}).get();
var duration = $('#duration').val();
$.ajax({
    url: "/currency/comparison",
    type: "POST",
    contentType: 'application/json',
    data: JSON.stringify({
        base_cur: baseCur,
        compare_curs: compareCurValues,
        duration : duration
    }),
    success: function (response) {
        console.log(response);
        modifyTable(response);

        var currencies = response.map(item => item.currency);
        var changes = response.map(item => item.changes.toFixed(3));

        chart.updateSeries([{
            name: "ExRate Changes based on " + baseCur,
            data: changes
        }]);
        chart.updateOptions({
            title:{
                text: 'Exchange Rate Changes based on ' + $("#base_cur").val()+ ' (' + $("#duratio
            },
            xaxis: {
                categories: currencies
            }
        });
    },
    error: function (xhr, status, error) {
        console.error(xhr.responseText);
    },
});
```

Figure 5.4.35: Code for Getting Currencies Comparison through Ajax

Once the ajax request is called, the Flask server will be sending a request to external API from Fast Forex to retrieve the historical exchange rates for each comparing currencies based on the base currency.

```
def retrieve_currency_comparison():
                return jsonify({'error': str(e)}), 400

        fetch_historical_exrate_url = "{}/historical?api_key={}&date={}&from={}".format(current_app.config['FAST_FOREX_API_URL'],
        current_app.config['FAST_FOREX_API_KEY'], start_date.strftime('%Y-%m-%d'), base_cur)
        response = requests.get(fetch_historical_exrate_url, headers=headers)
        if response.status_code == 200:
            json_obj = response.json()
            historical_results= json_obj['results']

        else:
            print(f"Error: {response.status_code} - {response.text}")
            return jsonify({'error': str(e)}), 400

        data = []
        for currency in compare_curs:
            current_rate = current_results[currency]
            historical_rate = historical_results[currency]
            changes = (current_rate- historical_rate)/historical_rate * 100
            data.append({'currency': currency, 'exchange_rate': current_rate, 'historical_rate': historical_rate, 'changes': changes })
        return jsonify(data), 200

    except DataMissingException as e:
        print(str(e))
        return jsonify({'error': str(e)}), 400

    except Exception as e:
        print(str(e))
        return jsonify({'error': str(e)}), 400
```

Figure 5.4.36: Code for Currencies Comparison API Endpoint

If the users have done setting of their favourite currencies, users can choose to compare with favourite currencies, else the button will be disabled.

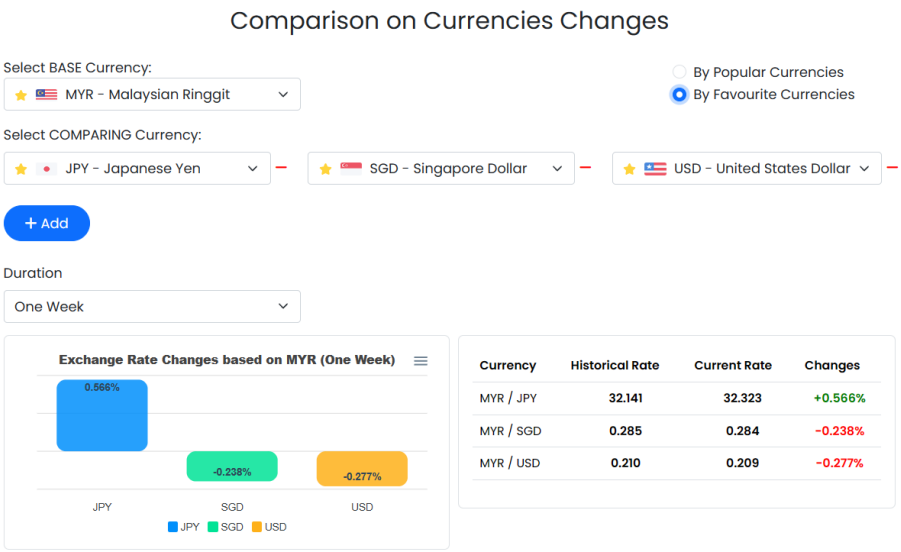


Figure 5.4.37: Comparison by Favourite Currencies

```
//Update currency selection when user choose to use popular currencies or favourite currencies
function modifyCurrencySelection(compare_curs){
    $.each(compare_curs, function(index, compare_cur) {
        var $newCompareCurDiv = $('<div>').addClass('form-group col');
        var $layoutDiv = $('<div>').addClass('d-flex align-items-center');

        var $select = $('<select>').addClass('form-select compare_cur').attr('name', 'compare_cur');
        var $defaultOption = $('<option>').attr('value', '').attr('disabled', 'disabled').attr('selected', 'selected').text('Select Currency ...');

        $select.append($defaultOption);
        {% for currency in fav_currencies %}
            var $option = $('<option>').attr('value', '{{ currency.code }}').attr('data-image', '{{ url_for('static', filename='flags/' + currency
            if (compare_cur === '{{ currency.code }}') {
                $option.prop('selected', true);
            }
            $select.append($option);
        {% endfor %}

        {% for currency in currencies %}
            var $option = $('<option>').attr('value', '{{ currency.code }}').attr('data-image', '{{ url_for('static', filename='flags/' + currency
            if (compare_cur === '{{ currency.code }}') {
                $option.prop('selected', true);
            }
            $select.append($option);
        {% endfor %}

        var $removeIcon = $('<i>').addClass('fa-solid fa-minus clickable-icon').attr('style','color:#ff0000;margin-left:5px')

        $layoutDiv.append($select);
        $layoutDiv.append($removeIcon);

        $newCompareCurDiv.append($layoutDiv);

        $('#comparing_currencies_section').append($newCompareCurDiv);
    });
    $('.compare_cur').select2({
        placeholder: 'Select currency...',
        theme: "bootstrap-5",
        templateResult: currencySelectionOption,
        templateSelection: currencySelectionOption,
        width: "100%",
    });
    $('#get_comparison_form').submit();
```

Figure 5.4.38: Code for Enabling Comparison by Favourite / Popular Currencies

At the same time, users can also perform addition or deletion of the currency input and both the chart, and the table will be updated instantly by clicking on add button or remove icon.

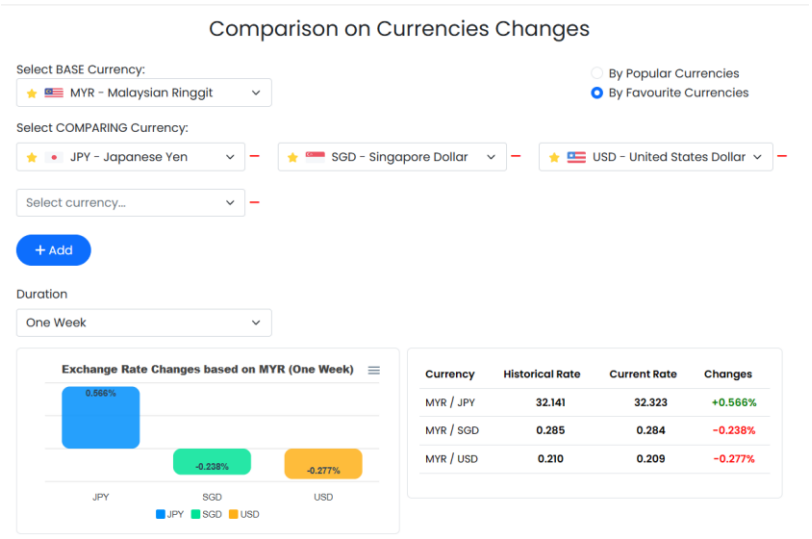


Figure 5.4.39: Adding Extra Currency Input

```
//Add Currency
$('#add_button').click(function(){
    var $newCompareCurDiv = $('<div>').addClass('form-group col');
    var $layoutDiv = $('<div>').addClass('d-flex align-items-center');

    var $select = $('<select>').addClass('form-select compare_cur').attr('name', 'compare_cur');
    var $defaultOption = $('<option>').attr('value', '').attr('disabled', 'disabled').attr('selected', 'selected').text('Select Currency ...');

    $select.append($defaultOption);
    {% for currency in fav_currencies %}
    var $option = $('<option>').attr('value', '{{ currency.code }}').attr('data-image', '{{ url_for('static', filename='flags/' + currency.alph
    $select.append($option);
    {% endfor %}

    {% for currency in currencies %}
    var $option = $('<option>')
                  .attr('value', '{{ currency.code }}')
                  .attr('data-image', '{{ url_for('static', filename='flags/' + currency.alpha_2_code.lower() + '.png') }}')
                  .text('{{ currency.code }} - {{ currency.name }}');
    $select.append($option);
    {% endfor %}

    var $removeIcon = $('<i>').addClass('fa-solid fa-minus clickable-icon').attr('style','color:#ff0000;margin-left:5px')

    $layoutDiv.append($select);
    $layoutDiv.append($removeIcon);

    $newCompareCurDiv.append($layoutDiv);

    $('#comparing_currencies_section').append($newCompareCurDiv);
    $('.compare_cur').select2({
        placeholder: 'Select currency...',
        theme: "bootstrap-5",
        templateResult: currencySelectionOption,
        templateSelection: currencySelectionOption,
        width: "100%",
    });
});
```

Figure 5.4.40: Code for Adding Extra Currency Input

```
//Remove Currency
$(document).on('click', '.fa-minus', function() {
    var selectInput = $(this).closest('.form-group').find('.compare_cur')
    selectInput.val(null);
    $(this).closest('.form-group').remove();
    $('#base_cur').change();
    $('input[name="selection"]').prop('checked', false);
});
```

Figure 5.4.41: Code for Removing Currency Input

The ajax request will be sent again once changes in either of any currencies input and duration are detected.

```
//Submit the form when data changes
$(document).on('change', '#base_cur, .compare_cur, #duration', function() {
    var baseCurValue = $('#base_cur').val();
    var compareCurValues = $('.compare_cur').map(function() {
        return $(this).val();
    }).get();
    var durationValue = $('#duration').val();
    if (baseCurValue != "" && compareCurValues.length>0 && durationValue != null)
        $('#get_comparison_form').submit();
});
```

Figure 5.4.42: Code for Detecting Changes on Currencies Input & Duration

5.4.6 Currency Changes Correlation Analysis

Users can navigate to the currency correlation analysis page by clicking on the currency correlation analysis button located on the sidebar. By default, the base currency will be set to user's default currency while the comparing currencies will be set to the popular currencies defined in the system. Duration will be set to one week by default.

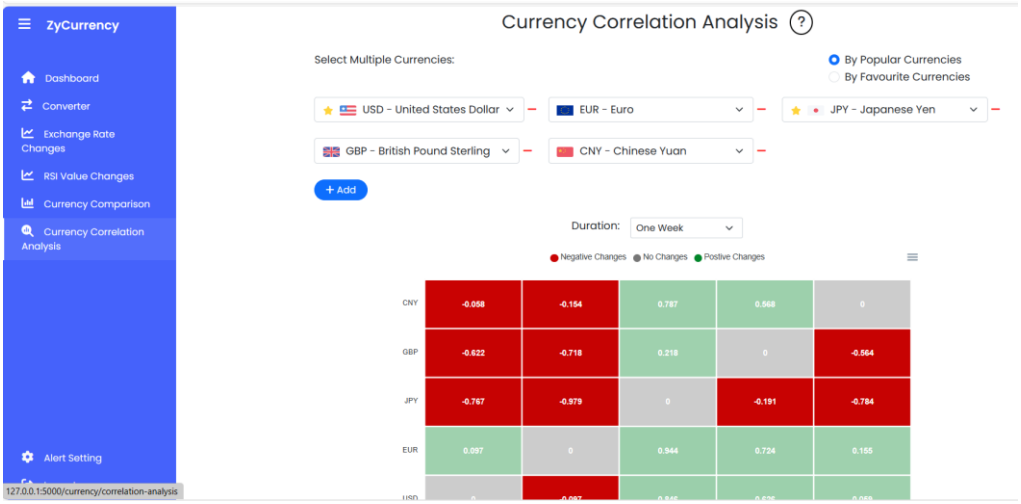


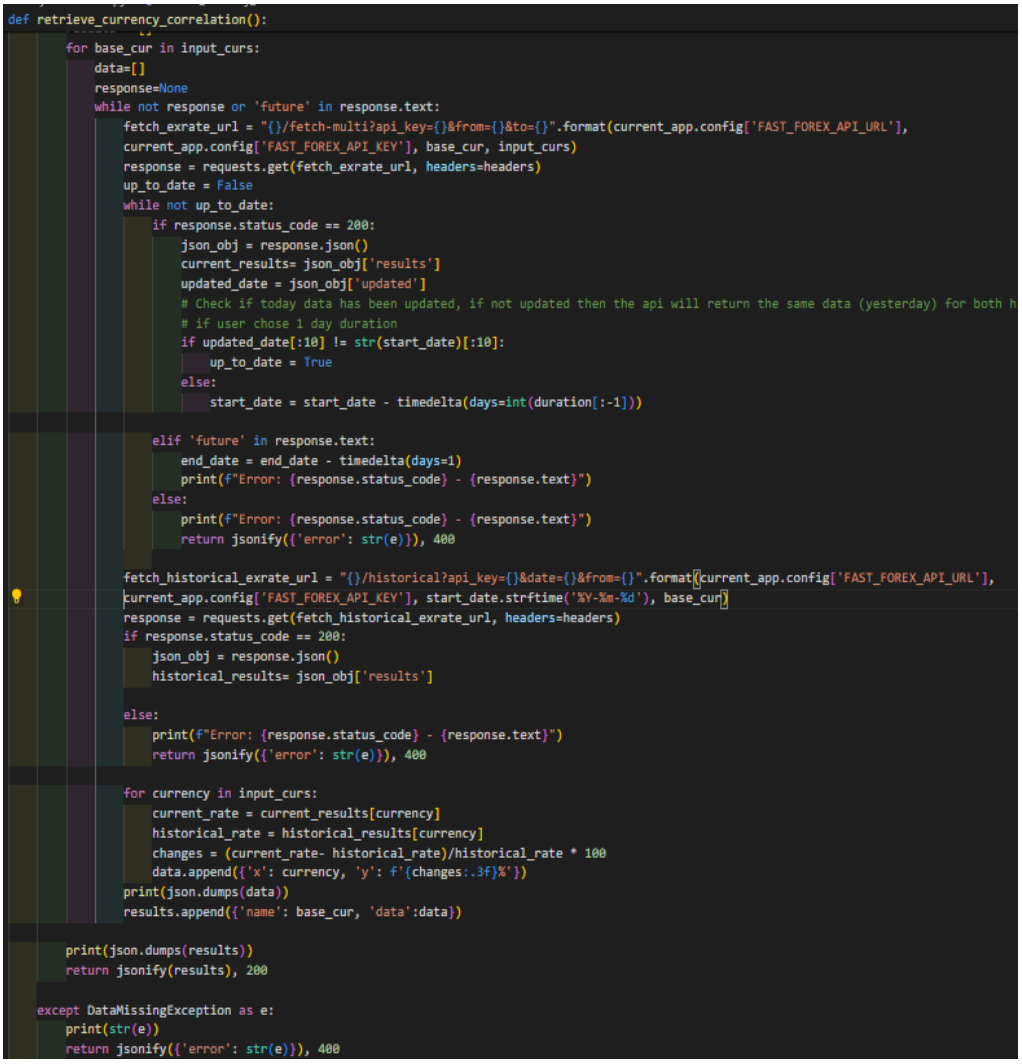
Figure 5.4.43: Currency Correlation Analysis Page

An ajax request will be sent to the correlation analysis API endpoint in Flask server when the page loads initially.

```
//Send first ajax request when page initially load
var curValues = $('.cur_input').map(function() {
    return $(this).val();
}).get();
var duration = $('#duration').val();
$.ajax({
    url: "/currency/correlation-analysis",
    type: "POST",
    contentType: 'application/json',
    data: JSON.stringify({
        input_curs: curValues,
        duration : duration
    }),
    success: function (response) {
        console.log(response)
        chart.updateSeries(response)
    },
    error: function (xhr, status, error) {
        console.error(xhr.responseText);
    },
});
```

Figure 5.4.44: Code for Getting Currencies Correlation through Ajax

Once the ajax request is called, the Flask server will be sending a request to external API from Fast Forex to retrieve the historical exchange rates for all the currencies with each currency being the base currency in every iteration.



```
def retrieve_currency_correlation():
    for base_cur in input_curs:
        data=[]
        response=None
        while not response or 'future' in response.text:
            fetch_exrate_url = "{}/fetch-multi?api_key={}&from={}&to={}".format(current_app.config['FAST_FOREX_API_URL'],
            current_app.config['FAST_FOREX_API_KEY'], base_cur, input_curs)
            response = requests.get(fetch_exrate_url, headers=headers)
            up_to_date = False
            while not up_to_date:
                if response.status_code == 200:
                    json_obj = response.json()
                    current_results= json_obj['results']
                    updated_date = json_obj['updated']
                    # Check if today data has been updated, if not updated then the api will return the same data (yesterday) for both h
                    # if user chose 1 day duration
                    if updated_date[:10] != str(start_date)[:10]:
                        up_to_date = True
                    else:
                        start_date = start_date - timedelta(days=int(duration[:-1]))

                elif 'future' in response.text:
                    end_date = end_date - timedelta(days=1)
                    print(f"Error: {response.status_code} - {response.text}")
                else:
                    print(f"Error: {response.status_code} - {response.text}")
                    return jsonify({'error': str(e)}), 400

            fetch_historical_exrate_url = "{}/historical?api_key={}&date={}&from={}".format(current_app.config['FAST_FOREX_API_URL'],
            current_app.config['FAST_FOREX_API_KEY'], start_date.strftime('%Y-%m-%d'), base_cur)
            response = requests.get(fetch_historical_exrate_url, headers=headers)
            if response.status_code == 200:
                json_obj = response.json()
                historical_results= json_obj['results']

            else:
                print(f"Error: {response.status_code} - {response.text}")
                return jsonify({'error': str(e)}), 400

            for currency in input_curs:
                current_rate = current_results[currency]
                historical_rate = historical_results[currency]
                changes = (current_rate- historical_rate)/historical_rate * 100
                data.append({'x': currency, 'y': f'{changes:.3f}%'})
            print(json.dumps(data))
            results.append({'name': base_cur, 'data':data})

        print(json.dumps(results))
        return jsonify(results), 200

    except DataMissingException as e:
        print(str(e))
        return jsonify({'error': str(e)}), 400
```

Figure 5.4.45: Code for Currencies Correlation Analysis API Endpoint

The heatmap chart also will be configured initially when the page loads and the data of the chart will be updated after the response received from the backend server.

```
//Define chart customizations
var options = {
    chart: {
        height: 450,
        width:800,
        type: "heatmap",
    },
    series: [],
    tooltip: {
        custom: function({series, seriesIndex, dataPointIndex, w}) {
            return '<div class="custom-tooltip">' +
                '<span class="tooltip-label">' + w.globals.seriesNames[seriesIndex] + ' / '  + w.globals.labels[dataPointIndex] + '</span>' + ' : ' +
                '<span class="tooltip-value">' +  series[seriesIndex][dataPointIndex] +'%' +  '</span>'
                '</div>';
        },
    },
    noData: {
        text: 'Requesting Data...'
    },
    plotOptions: {
        heatmap: {
            colorScale: {
                ranges: [{
                    from: -30,
                    to: -0.0001,
                    color: '#c90202',
                    name: 'Negative Changes',
                },
                {
                    from: -0.0001,
                    to: 0.0001,
                    color: '#767676',
                    name: 'No Changes',
                },
                {
                    from: 0.0001,
                    to: 30,
                    color: '#018529',
                    name: 'Postive Changes',
                }
                ]
            },
            reverseNegativeShade: true,
        }
    }
```

Figure 5.4.46: Heatmap Chart Configuration

Similarly to the currency comparison page, user can choose to perform correlation analysis by their favourite currencies if the favourite currencies are set previously. Besides that, users can add or remove any currency input for extra flexibility when utilizing this feature.

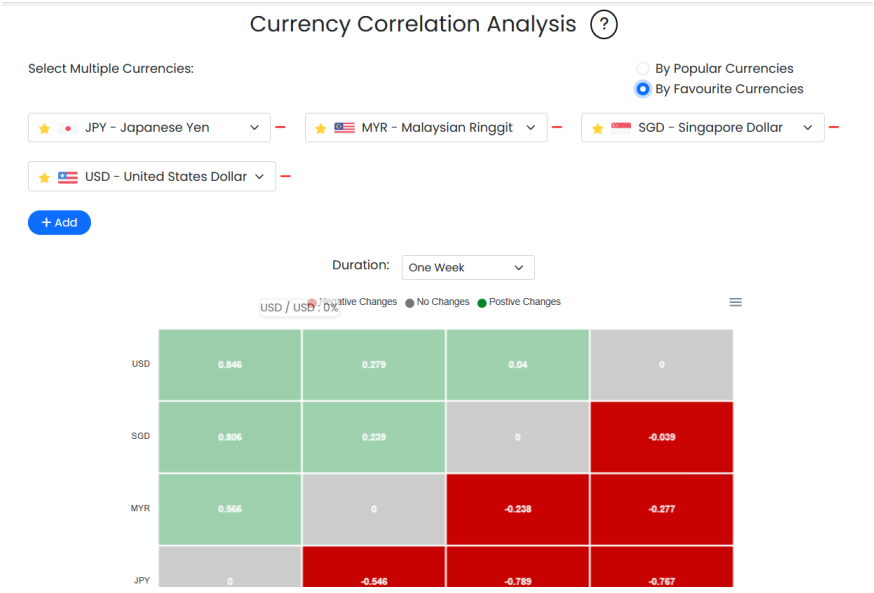


Figure 5.4.47: Currency Correlation Analysis by Favourite Currencies

5.4.7 Extra Chart Features

For the line charts in exchange rates changes page and RSI values changes page, there will be built-in buttons for zooming in and zooming out the chart.

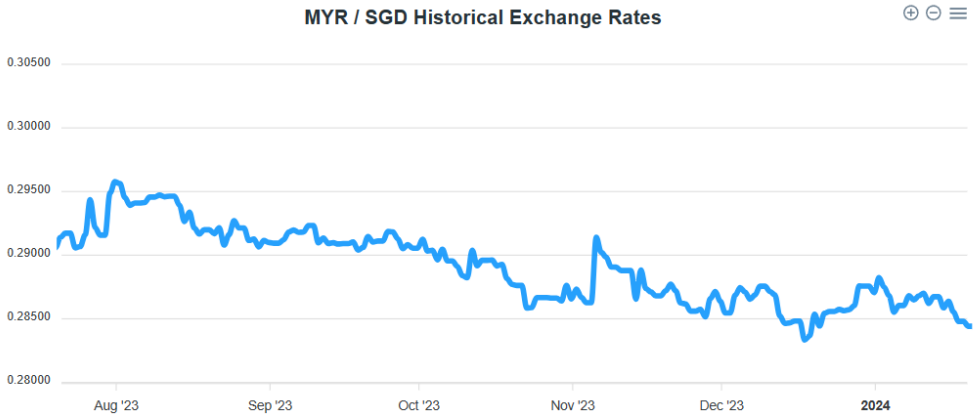


Figure 5.4.48: Zoom In & Zoom Out Buttons for Line Charts

At the same time, all the Apex charts other than the charts on dashboard page, the download feature is available. The charts can be downloaded in form of svg. png or csv format.

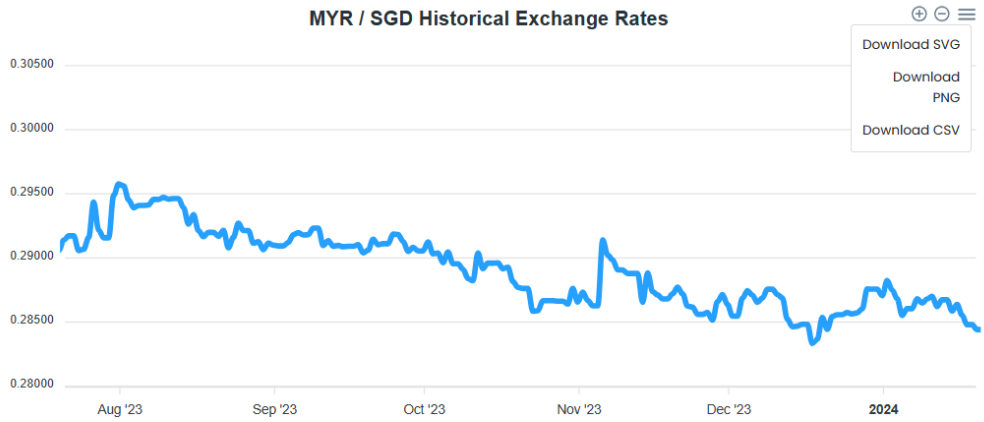


Figure 5.4.49: Download Feature

5.5 Alert Module

5.5.1 Exchange Rate Alert

At the exchange rate changes page, users can click on the set alert button for setting an alert to be triggered either periodically or conditionally.

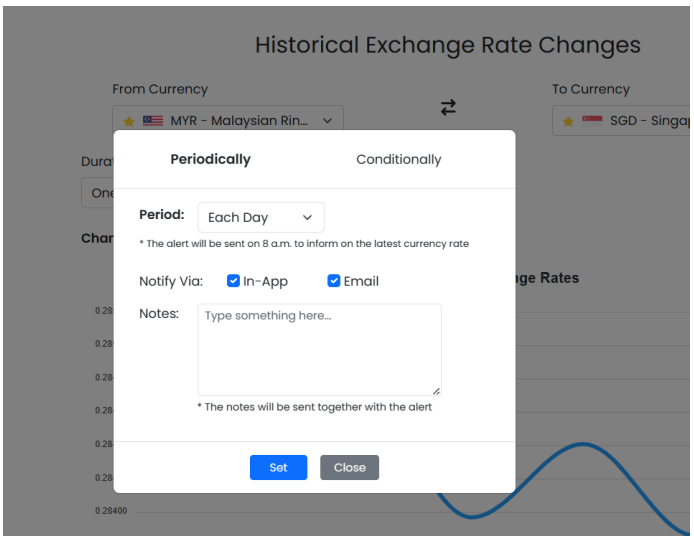


Figure 5.5.1: Setting Periodic Alerts for Exchange Rate

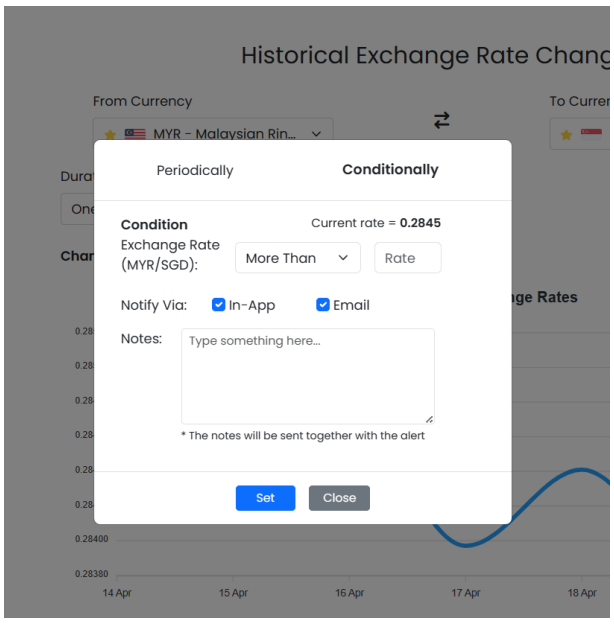


Figure 5.5.2: Setting Conditional Alerts for Exchange Rate

Users will be notified when they set the alert successfully. Otherwise, the error message will be displayed for the users if the system failed to set the alert.

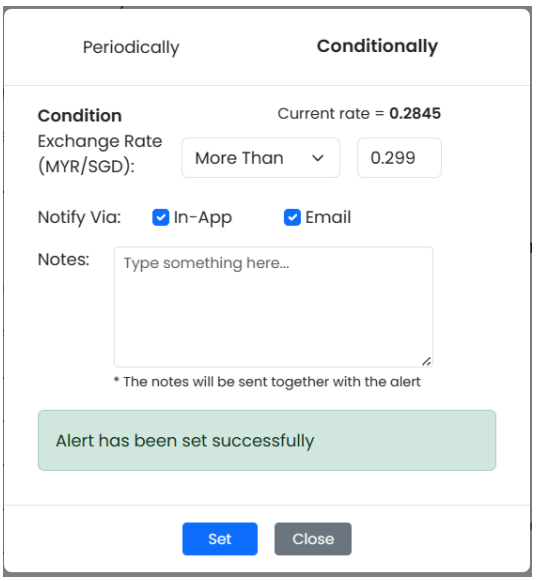


Figure 5.5.3: Success Message on Setting Alert

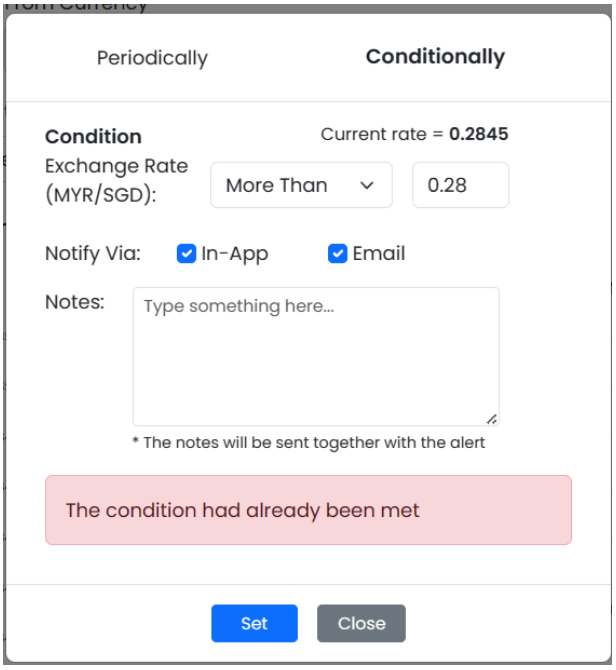


Figure 5.5.4: Error Message on Setting Alert

An ajax request will be sent to the Flask API endpoint for creating the alert once the user inputs pass the validation.

```
$('#create_alert_form').submit(function(event) {
  event.preventDefault();
  if ((($("#condition").val() == "more" &&  $("#current_rate").text() > $("#rate").val()) ||
     ($("#condition").val() == "less" && $("#current_rate").text() < $("#rate").val()))) {
     $("#errorMessage").text('The condition had already been met');
     $("#errorMessage").show();
  }
  else{
    var formData = $(this).serialize();
    $.ajax({
        url: '/alert/create',
        type: 'POST',
        data: formData,
        success: function(response) {
          $('#successMessage').text(response.message);
          $('#successMessage').show();
          $('#errorMessage').hide();
          console.log(response);
        },
        error: function(xhr, status, error) {
          $('#errorMessage').text(xhr.responseJSON.error);
          $('#errorMessage').show();
          $('#successMessage').hide();
          console.error(xhr.responseText);
        }
    });
  }
});
```

Figure 5.5.5: Code for Creating Alerts through Ajax

5.5.2 RSI Value Alert

Similar to the exchange rate changes page, users can click on the set alert button on RSI values changes page for setting an alert to be triggered either periodically or conditionally. The only difference is user can select the condition to be set as oversold or overbought while setting a conditional alert.

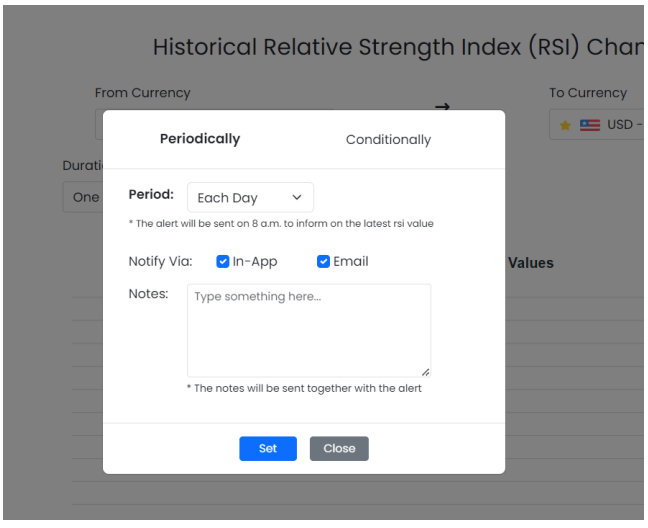


Figure 5.5.6: Setting Periodic Alerts for RSI Value

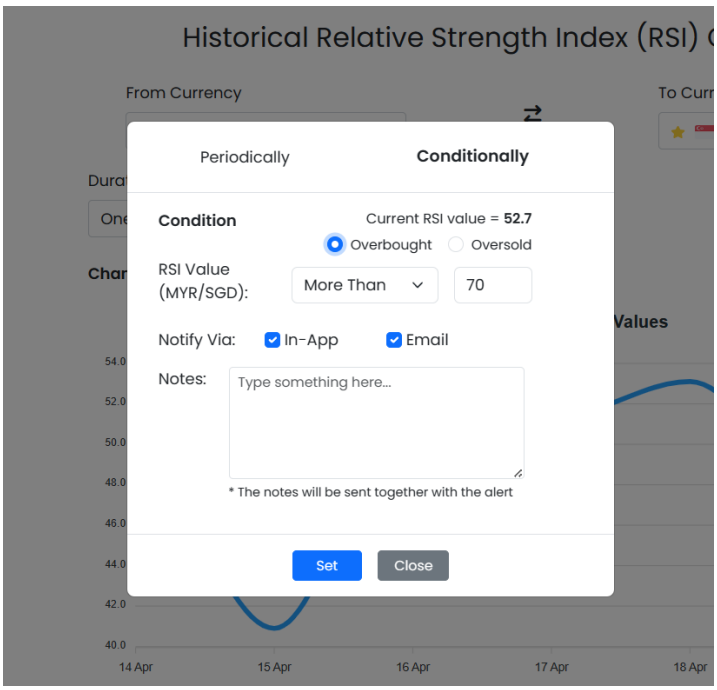


Figure 5.5.7: Setting Conditional Alerts as Overbought

Figure 5.5.8: Setting Conditional Alerts as Oversold

5.5.3 Alert Setting

Users can navigate to the alert setting page by clicking on the alert setting button located on the sidebar. All alerts that belong to the current user will be shown on this page.

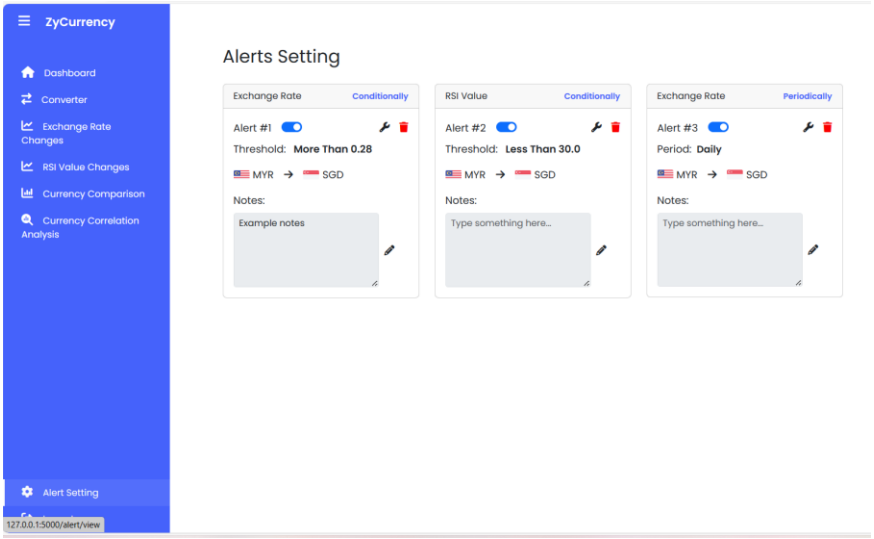


Figure 5.5.9: Alerts Setting Page

In addition, users are able to perform actions like toggling, removing and also updating the alerts on this page.

```
$('.toggle_alert').change(function() {
  var alertId = $(this).data('alert-id');
  $.ajax({
      url: '/alert/toggle/' + alertId,
      type: 'PUT',
      success: function(response) {
        console.log(response);
      },
      error: function(xhr, status, error) {
        console.error(xhr.responseText);
      }
  });
});
```

Figure 5.5.10: Code for Toggling Alerts through Ajax

```
$('.delete_alert').click(function() {
  var alertId = $(this).data('alert-id');
  var $card = $(this).closest('.card');
  $.ajax({
      url: '/alert/delete/' + alertId,
      type: 'DELETE',
      success: function(response) {
        $card.remove();
        location.reload();
        alert(response.message);
        console.log(response);
      },
      error: function(xhr, status, error) {
        alert('Error: ' + xhr.responseText);
        console.error(xhr.responseText);
      }
  });
});
```

Figure 5.5.11: Code for Deleting Alerts through Ajax

```
$('.save_notes').click(function() {
  var notesId = $(this).data('id');
  var $notes = $('#notes_' + notesId);
  $notes.prop('disabled', true);
  var $editIcon = $('#edit_icon_'+notesId)
  $editIcon.show();
  $(this).hide();

  var notesContent = $notes.val();
  $.ajax({
    url: '/alert/edit-notes/' + notesId,
    type: 'PUT',
    contentType: 'application/json',
    data: JSON.stringify({ notes: notesContent }),
    success: function(response) {
      console.log(response);
    },
    error: function(xhr, status, error) {
      console.error(xhr.responseText);
    }
  });
})
```

Figure 5.5.12: Code for Editing Notes of Alert through Ajax

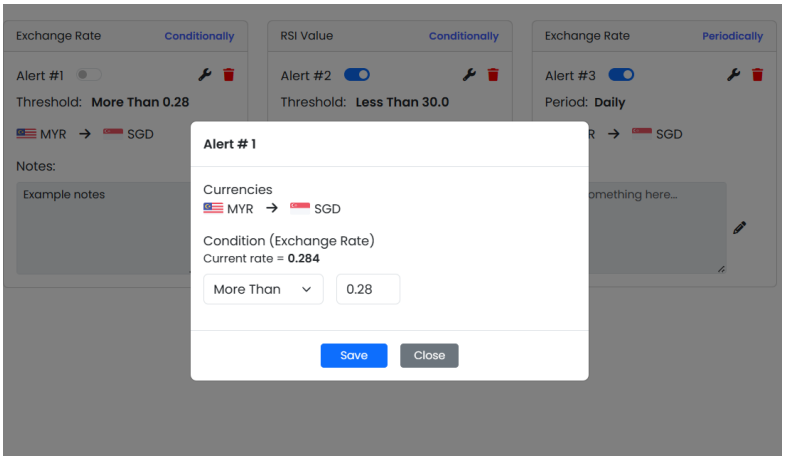


Figure 5.5.13: Edit Modal for Conditional Alert

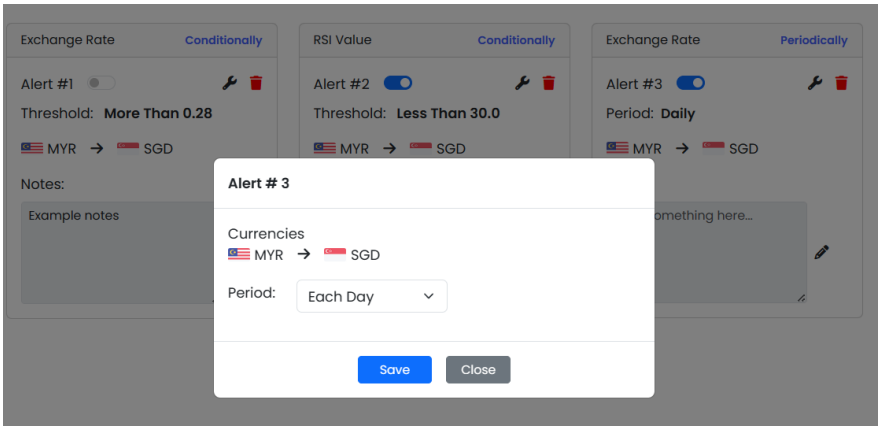


Figure 5.5.14: Edit Modal for Periodic Alert

```
$(document).on('click','.update_alert',function(){
  var alertId = $(this).data('id');
  var period = $('#period').val();
  var condition = $('#condition').val();
  var value = $('#value').val();

  var validRequest = true;
  if ($('#alert_type').val() == 'conditionally'){
    if ( $('#indicator').val() == 'exrate'){
      if (((condition== "more" &&  $("#current_rate").text() > value) ||
        (condition== "less" && $("#current_rate").text() < value))) {
        $("#errorMessage").text('The condition had already been met');
        $("#errorMessage").show();
        validRequest = false;
      }
    }
    else if ( $('#indicator').val() == 'rsi'){
      if (((condition == "more" &&  $("#current_rsi").text() > value) ||
      (condition == "less" && $("#current_rsi").text() < value))) {
      $("#errorMessage").text('The condition had already been met');
      $("#errorMessage").show();
      validRequest = false;
    }
    }
  }

  if (validRequest){
    $.ajax({
      url: '/alert/edit/' + alertId,
      type: 'PUT',
      contentType: 'application/json',
      data: JSON.stringify({
        period: period,
        condition: condition,
        value: value,
      }),
      success: function(response) {
        console.log(response);
        $('#errorMessage').hide()
        $('#successMessage').text("The alert setting has been updated successfully")
        $('#successMessage').show()
      },
      error: function(xhr, status, error) {
        console.error(xhr.responseText);
        $('#successMessage').hide()
        $('#errorMessage').text(xhr.responseText)
        $('#errorMessage').show()
      }
    })
  }
```

Figure 5.5.15: Code for Validation & Updating Alert through Ajax

5.5.4 Scheduler

Flask APScheduler is being installed and used for sending periodic notifications at a specific timing and also performing conditional checking for conditional alerts in 5 minutes interval.

```
from apps.alert.tasks import periodically_currency_update, alert_condition_check
scheduler.add_job(func=periodically_currency_update, args=(app,'daily'), trigger='cron', day_of_week='*', hour='8', minute='0', id='daily_currency_update')
scheduler.add_job(func=periodically_currency_update, args=(app,'weekly'), trigger='cron', day_of_week='MON', hour='8', minute='0', id='weekly_currency_update')
scheduler.add_job(func=periodically_currency_update, args=(app,'monthly'), trigger='cron', month='*', day=1, hour='8', minute='0', id='monthly_currency_update')

scheduler.add_job(func=alert_condition_check, args=(app,), trigger='interval', seconds=300, id='alert_condition_check')

scheduler.start()
```

Figure 5.5.16: Scheduler Configurations

```
mingle: searching for neighbors
mingle: all alone
celery@ZY ready.
Task apps.notification.tasks.send_conditional_notification_mail[d5a5a663-fb1f-4f28-9f0d-cffff5cd21cf] received
ss] Sending notification email
Task apps.notification.tasks.send_conditional_notification_mail[d5a5a663-fb1f-4f28-9f0d-cffff5cd21cf] succeeded in 3.5469999999622814s:

Running job "alert_condition_check (trigger: interval[0:00:30], next run at: 2024-04-21 21:11:23 +08)" (scheduled at 2024-04-21 21:11:23

ss] Exchange rate for MYR to USD: 0.20908
ss] RSI value for MYR to SGD: 52.7
Job "alert_condition_check (trigger: interval[0:00:30], next run at: 2024-04-21 21:11:53 +08)" executed successfully
Running job "alert_condition_check (trigger: interval[0:00:30], next run at: 2024-04-21 21:12:23 +08)" (scheduled at 2024-04-21 21:11:53

ss] Exchange rate for MYR to USD: 0.20908
ss] RSI value for MYR to SGD: 52.7
Job "alert_condition_check (trigger: interval[0:00:30], next run at: 2024-04-21 21:12:23 +08)" executed successfully
Running job "alert_condition_check (trigger: interval[0:00:30], next run at: 2024-04-21 21:12:53 +08)" (scheduled at 2024-04-21 21:12:23

ss] Exchange rate for MYR to USD: 0.20908
ss] RSI value for MYR to SGD: 52.7
Job "alert_condition_check (trigger: interval[0:00:30], next run at: 2024-04-21 21:12:53 +08)" executed successfully
```

Figure 5.5.17: Scheduler Tasks Running on Celery

5.6 Notification Module

5.6.1 Email Notifications

If user checked email as the alert's notification method, when the alert is being triggered, it will send email to the user's inbox.

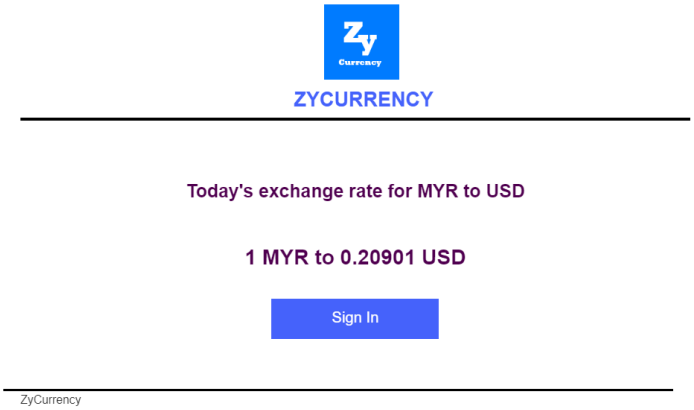


Figure 5.6.1: Email for Periodic Notification

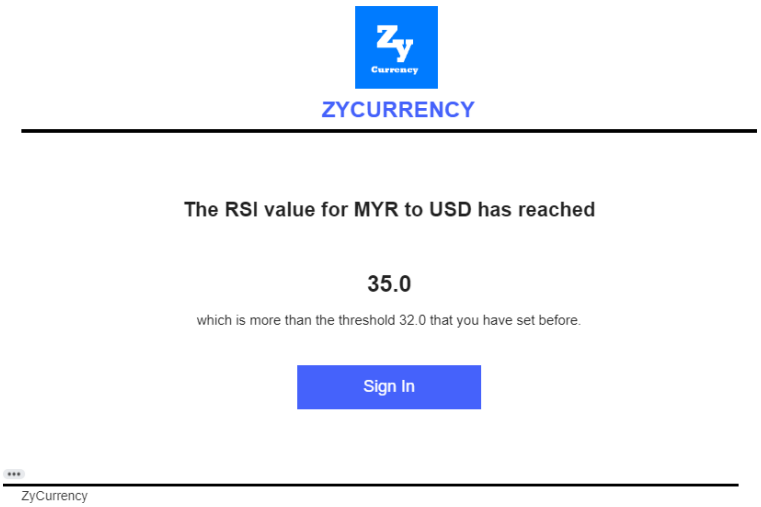


Figure 5.6.2: Email for Conditional Notification

5.6.2 App Notifications

If user checked in-app notification as the alert's notification method, when the alert is being triggered, the system will create notification data for that particular user. Users are able to view their notification by clicking the bell icon on the top bar.

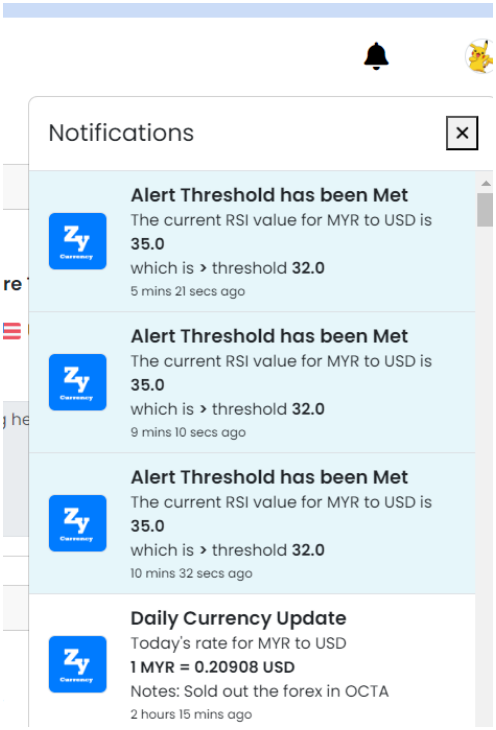


Figure 5.6.3: View In-App Notifications

When the page loads, the system will send the ajax request to fetch all the notifications.

```
$.ajax({
  url: '/notification/get-all',
  type: 'GET',
  success: function(response) {
    console.log(response);
    notifications = response.notifications;
    var unreadCount = response.unread_count;
    showNotifications(unreadCount, notifications);
  },
  error: function(xhr, status, error) {
    console.error(xhr.responseText);
  }
});
```

Figure 5.6.4: Code for Fetching Notifications through Ajax

5.7 Project Structure & Project Configurations

The project structure will be consisting of the base directory and four addition modules which are alert module, auth module, currency module and notification module. The structure of base directory and each module are shown in the figures below.

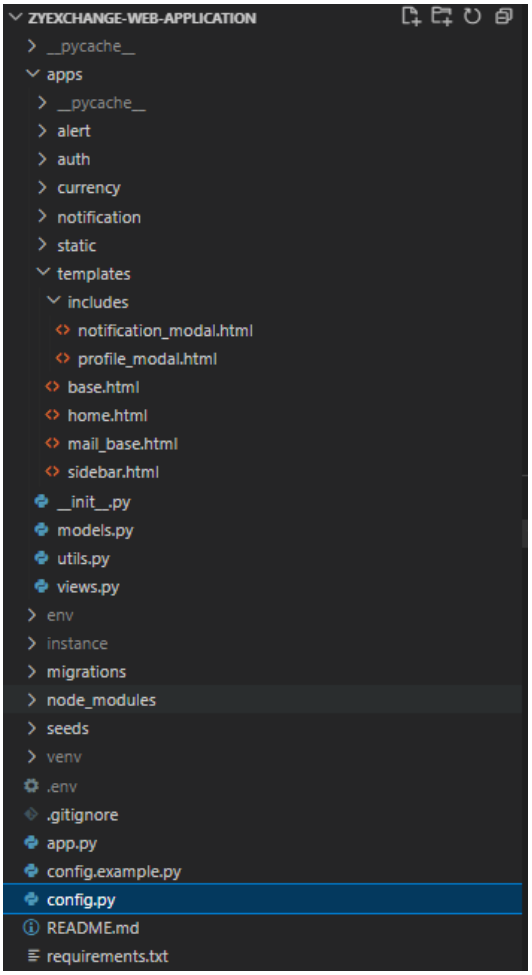


Figure 5.7.1: Structure of Base Directory

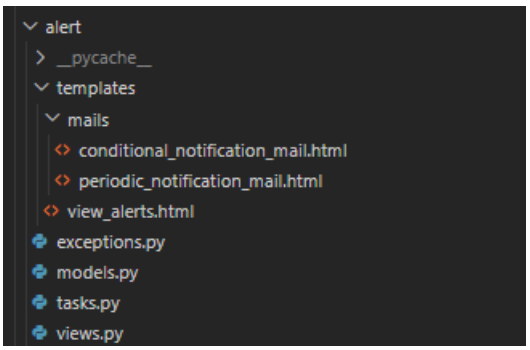


Figure 5.7.2: Structure of Alert Module

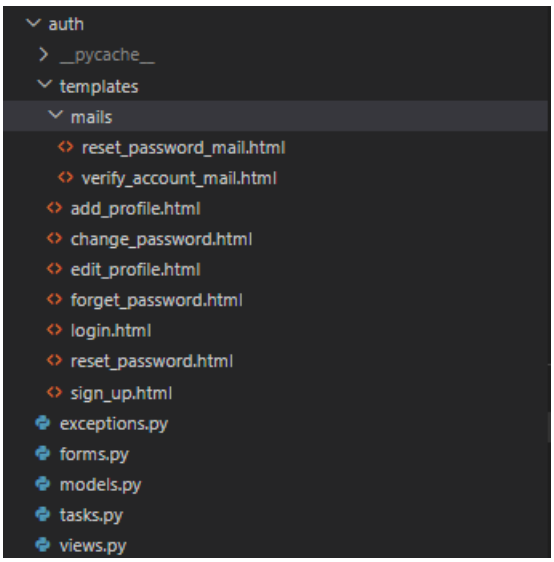


Figure 5.7.3: Structure of Auth Module

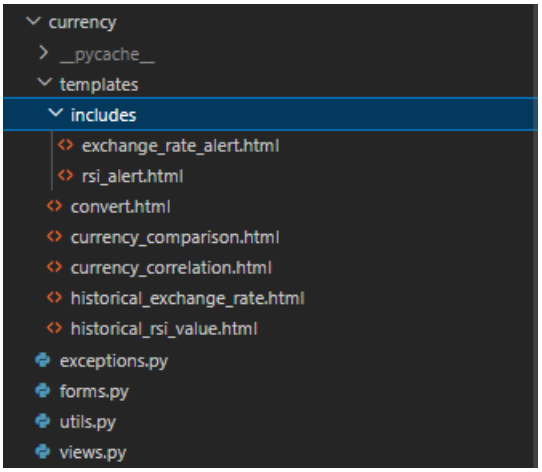


Figure 5.7.4: Structure of Currency Module

Figure 5.7.5: Structure of Notification Module

The configuration of the projects is shown in the figures below.

```python
import os
from dotenv import load_dotenv

load_dotenv()

class Config:
    SECRET_KEY = os.getenv("SECRET_KEY")
    SQLALCHEMY_DATABASE_URI = f'sqlite:///zycurrency.db'
    URL_DOMAIN_WITH_PROTOCOL = 'http://127.0.0.1:5000'
    SERVER_NAME = "127.0.0.1:5000"

    GOOGLE_CLIENT_ID=os.getenv("GOOGLE_CLIENT_ID")
    GOOGLE_CLIENT_SECRET=os.getenv("GOOGLE_CLIENT_SECRET")
    GOOGLE_DISCOVERY_URL="https://accounts.google.com/.well-known/openid-configuration"

    FAST_FOREX_API_URL="https://api.fastforex.io"
    FAST_FOREX_API_KEY=os.getenv("FAST_FOREX_API_KEY")

    MAIL_SERVER = 'smtp-relay.brevo.com'
    MAIL_PORT = 587
    MAIL_USE_TLS = True
    MAIL_USE_SSL = False
    MAIL_USERNAME = os.getenv("MAIL_USERNAME")
    MAIL_PASSWORD = os.getenv("MAIL_PASSWORD")

    CELERY_BROKER_URL='redis://localhost:6379/0',
    CELERY_RESULT_BACKEND='redis://localhost:6379/0'
    CELERY_TASK_IGNORE_RESULT = False

    POPULAR_CURRENCIES= ['USD', 'EUR', 'JPY', 'GBP', 'CNY']
    POPULAR_CURRENCIES_PAIRS = ['EUR/USD', 'USD/JPY', 'GBP/USD', 'AUD/USD', 'USD/CAD']

    SCHEDULER_API_ENABLED = True
```

Figure 5.7.6: Config.py File

```python
def create_app():
    app = Flask(__name__)
    app.config.from_object(Config)

    db.init_app(app)
    migrate.init_app(app, db)
    seeder.init_app(app, db)
    scheduler.init_app(app)

    mail.init_app(app)

    app.config.from_mapping(
        CELERY=dict(
            broker_url="redis://127.0.0.1:6379/0",
            result_backend="redis://127.0.0.1:6379/0",
            task_ignore_result=False,
        ),
    )

    celery_init_app(app)
    serializer = URLSafeTimedSerializer(app.config['SECRET_KEY'])
    app.config['SERIALIZER'] = serializer

    from apps.views import views
    app.register_blueprint(views, url_prefix='/')
    from apps.auth.views import auth_bp
    app.register_blueprint(auth_bp, url_prefix='/auth')
    from apps.currency.views import currency_bp
    app.register_blueprint(currency_bp, url_prefix='/currency')
    from apps.alert.views import alert_bp
    app.register_blueprint(alert_bp, url_prefix='/alert')
    from apps.notification.views import notification_bp
    app.register_blueprint(notification_bp, url_prefix='/notification')

    from apps.auth.models import User
    from apps.models import Country, Currency
    from apps.alert.models import Alert
    from apps.notification.models import Notification
    with app.app_context():
        db.create_all()
        print('Connected to Database')


    login_manager = LoginManager()
    login_manager.login_view = 'auth.login'
    login_manager.init_app(app)

    from apps.alert.tasks import periodically_currency_update, alert_condition_che
    scheduler.add_job(func=periodically_currency_update, args=(app,'daily'), trigg
    scheduler.add_job(func=periodically_currency_update, args=(app,'weekly'), trig
    scheduler.add_job(func=periodically_currency_update, args=(app,'monthly'), tri

    scheduler.add_job(func=alert_condition_check, args=(app,), trigger='interval',

    scheduler.start()


    @login_manager.user_loader
    def load_user(id):
        return User.query.get(int(id))

    return app
```

Figure 5.7.7: Flask Application Initialization Code

5.8 Data Migration

By using Flask Migrate Library, data migration can be implemented into the project by running migrate and update commands. The figure below shows a list of migration files generated through the development of the project.

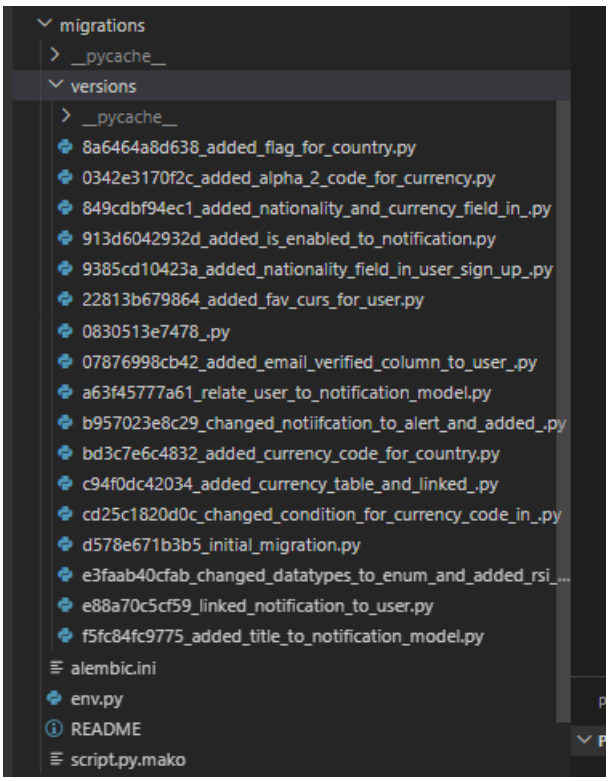


Figure 5.8.1: Migration Files

5.9 Data Seeding

By using Flask Seeder Library, data seeding can be implemented into the project by inserting default data into the country table and the currency table.

Figure 5.9.1: Seeding Files

CHAPTER 6

TESTING

6.1 Introduction

In this chapter, unit testing, integration testing, and user acceptance testing were conducted to evaluate the functionality and usability of the developed currency exchange rate tracking system in the previous chapter. Unit testing was performed to test the system by separation into individual units. Next, integration testing was implemented to test the interoperability between different modules of the system. Furthermore, the user acceptance test (UAT) was organized to determine if the application can meet end-users' requirements. Lastly, the usability testing was carried out to evaluate the usability of the system and also collect feedback from the testers. The table below incorporates a traceability matrix to link each test case in the unit testing to a respective use case.

Table 6-1: Use Case Traceability Matrix

Use Case ID	**Use Case Name**
UC01	Sign Up Account
UC02	Login Account
UC03	Reset Password
UC04	Add / Change Profile Detail
UC05	Logout System
UC06	View Historical Exchange Rate
UC07	View Historical RSI Vales
UC08	View Exchange Rate Comparison
UC09	View Currency Correlation
UC10	Convert Currency

UC11	Set Favourite Currencies
UC12	Create Alert
UC13	Toggle Alert
UC14	Edit Alert
UC15	Delete Alert
UC16	Send Email Notification
UC17	View In-App Notifications

6.2 Unit Testing

Unit testing was performed as the first testing methodology to evaluate every component of the system in isolation. The currency exchange rate tracking system is broken down into units which can be classified using use cases to introduce traceability in testing documentation.

Table 6-2: Unit Testing Test Cases

Test Case ID	Test Case Name	Test Case Description	Expected Result	Use Case	Status
UTC001	Test sign up with existing email	Sign up with an existing email in the database	System will prompt this email has been registered	UC01	Pass
UTC002	Test sign up with non-existing email	Input email field in invalid format	System will prompt invalid email address		Pass
UTC003	Test sign up with incorrect	Upload file with format other than jpg	System will prompt unaccepted file format		Pass

	profile pic format	and png as profile picture			
UTC004	Test sign up with empty input	Submit the form without filling in email, password, first name and default currency	System will prompt the field is required for empty inputs		Pass
UTC005	Test sign up with unmatching confirm password	Submit the form with different values for password and confirm password	System will prompt password must match on password field		Pass
UTC006	Test sign up with same currencies for default currency and second currency	Choose the same currency for default currency and second currency inputs	System will prompt default currency cannot be same with second currency		Pass
UTC007	Test sign up with correct inputs	Enter all the required inputs in correct format and submit form	System will redirect user to the login page and prompt account verification mail has been sent to inbox		Pass

UTC008	Test account verification with expired link	Click on the link of account verification for email which has already been verified	System will prompt the verification link has expired		Pass
UTC009	Test account verification with false token	Use random string as fake account verification token	Requested URL not found		Pass
UTC010	Test account verification with correct token	Click on the link sent to the email inbox for unverified account	User is redirected to login page with success message		Pass
UTC011	Test login account with empty inputs	Attempt to login without filling in email or password	System will prompt the field is required for empty inputs	UC02	Pass
UTC012	Test login account with invalid email format	Enter email field in invalid format and login	System will prompt the email is invalid		Pass
UTC013	Test login account with unverified account	Enter correct email and password for the unverified account	System will prompt the account is not verified		Pass
UTC014	Test login account with	Enter wrong email or password	System will prompt invalid		Pass

	incorrect credentials		email or password		
UTC015	Test login account with correct credentials	Enter correct email and password	System will redirect users to the dashboard page		Pass
UTC016	Test login account with Google sign in	Click sign in with Google on login page and select Google account	User is redirected to add profile page		Pass
UTC017	Test login account with password for Google OAuth accounts	Attempt to login without filling in password using account signed in with Google	System will prompt the account does not have any passwords		Pass
UTC018	Test forget password with invalid email format	Enter email field in invalid format and submit	System will prompt the email is invalid	UC03	Pass
UTC019	Test forget password with empty input	Leave email input in blank and submit	System will prompt the field is required		Pass
UTC020	Test forget password with non-existing email	Enter a valid email format but the email is never used	System will prompt email cannot be recognised		Pass

		before for registration			
UTC021	Test forget password with correct email	Enter correct email and submit	System will prompt users to check email inbox		Pass
UTC022	Test reset password with expired link	Click on the link for resetting password which has already been used	System will prompt the reset password link has expired		Pass
UTC023	Test reset password with false token	Use random string as fake reset password token	Requested URL not found		Pass
UTC024	Test reset password with correct token	Click on the unused link sent to the email inbox	User is redirected to reset password page		Pass
UTC025	Test reset password with empty inputs	Reset password without filling in new password or confirm password	System will prompt the field is required		Pass
UTC026	Test reset password with unmatching passwords	Enter different inputs for new password and confirm password.	System will prompt users to match the passwords		Pass

UTC027	Test reset password with old password	Using old password as new password and confirm password	System will prompt the user that old password is not acceptable		Pass
UTC028	Test reset password with matching passwords	Enter same inputs for both new password and confirm password	System will prompt success message		Pass
UTC029	Test change password with empty inputs	Change password without filling in old password new password	System will prompt the field is required		Pass
UTC030	Test change password with incorrect old password	Enter the wrong current password for old password field	System will prompt the old password is not matching with current one		Pass
UTC031	Test change password with unmatching passwords	Enter different inputs for new password and confirm password.	System will prompt users to match the passwords		Pass
UTC032	Test change password with current password as	Using current password as new password	System will prompt the user that old		Pass

	new password	and confirm password	password is not acceptable		
UTC033	Test change password with correct inputs	Enter correct old password and ensure new password and confirm password are matching	System will prompt success message		Pass
UTC034	Test add profile with required input being empty	Add profile without choosing a default currency	System will prompt the field is required	UC04	Pass
UTC035	Test add profile with same default currency and second currency	Choose same currency for both default currency input and second currency input	System will prompt default currency cannot be same with second currency		Pass
UTC036	Test edit profile with incorrect profile pic format	Upload file with format other than jpg and png as profile picture	System will prompt unaccepted file format		Pass
UTC037	Test edit profile with same default currency and second currency	Choose same currency for both default currency input and second currency input	System will prompt default currency cannot be same with second currency		Pass

UTC038	Test edit profile with empty input for first name	Blank out the field for first name	System will prompt the field is required for first name		Pass
UTC039	Test edit profile with correct and valid inputs for required fields	Modify the value of first name to another string	System will prompt success message		Pass
UTC040	Test logout	Click on logout button on sidebar	System will redirect user to the login page	UC05	Pass
UTC041	Test view exchange rate chart on initial page loads	Click on exchange rate changes on sidebar	System will load the charts with default currency and second currency	UC06	Pass
UTC042	Test view exchange rate with from currency changes	Choose another currency for from currency input	System will update the chart based on the chosen currency		Pass
UTC043	Test view exchange rate with to currency changes	Choose another currency for to currency input	System will update the chart based on the chosen currency		Pass

UTC044	Test view exchange rate with duration changes	Choose another duration for duration input	System will update the chart based on the chosen duration		Pass
UTC045	Test view exchange rate with immediate exchange of currencies	Press exchange icon to instantly exchange the value of from currency with to currency	System will update the chart based on the new pair of currencies		Pass
UTC046	Test exchange rate chart intractability with cursor hovering	Hovers the cursor towards any data points in the line chart	Details of the data point including date, currency pair and exchange rate will be displayed.		Pass
UTC0467	Test exchange rate chart interactivity by zooming in chart	Press the zoom in icon at the top right corner of chart	The chart will be zoomed in		Pass
UTC048	Test exchange rate chart intractability by zooming out chart	Press the zoom out icon at the top right corner of chart	The chart will be zoomed out		Pass
UTC049	Test download	Press the download icon	The svg file with the chart		Pass

	exchange rate chart as svg	and select download svg	image will be installed on user's machine		
UTC050	Test download exchange rate chart as png	Press the download icon and select download png	The png file with the chart image will be installed on user's machine		Pass
UTC051	Test download exchange rate chart as csv	Press the download icon and select download csv	The csv file with the chart data will be installed on user's machine		Pass
UTC052	Test exchange rate chart intractability by dragging chart	Enable the panning feature by clicking on the drag icon	The chart can be dragged towards right or left once it is zoomed in		Pass
UTC053	Test exchange rate chart intractability by resetting chart	Click on the home icon at the top right corner of chart	The chart will be reset to default		Pass
UTC054	Test view RSI values chart on initial page loads	Click on RSI value changes on sidebar	System will load the charts with default currency and	UC07	Pass

			second currency		
UTC055	Test view RSI values with from currency changes	Choose another currency for from currency input	System will update the chart based on the chosen currency		Pass
UTC056	Test view RSI values with to currency changes	Choose another currency for to currency input	System will update the chart based on the chosen currency		Pass
UTC057	Test view RSI values with duration changes	Choose another duration for duration input	System will update the chart based on the chosen duration		Pass
UTC058	Test view RSI values with immediate exchange of currencies	Press exchange icon to instantly exchange the value of from currency with to currency	System will update the chart based on the new pair of currencies		Pass
UTC059	Test RSI value chart intractability with cursor hovering	Hover the cursor towards any data points in the line chart	Details of the data point including date, currency pair and exchange rate will be displayed.		Pass
UTC060	Test RSI values chart	Press the zoom in icon at the	The chart will be zoomed in		Pass

	interactivity by zooming in chart	top right corner of chart			
UTC061	Test RSI values chart intractability by zooming out chart	Press the zoom out icon at the top right corner of chart	The chart will be zoomed out		Pass
UTC062	Test download RSI value chart as svg	Press the download icon and select download svg	The svg file with the chart image will be installed on user's machine		Pass
UTC063	Test download RSI value chart as png	Press the download icon and select download png	The png file with the chart image will be installed on user's machine		Pass
UTC064	Test download RSI value chart as csv	Press the download icon and select download csv	The csv file with the chart data will be installed on user's machine		Pass
UTC065	Test RSI value chart intractability by dragging chart	Enable the panning feature by clicking on the drag icon	The chart can be dragged towards right or left once it is zoomed in		Pass

UTC066	Test RSI value chart intractability by resetting chart	Click on the home icon at the top right corner of chart	The chart will be reset to default		Pass
UTC067	Test currencies comparison on initial page load.	Click on currency comparison on sidebar	The chart will be loaded with default currency as base currency and popular currencies as comparing currencies	UC08	Pass
UTC068	Test currencies comparison by user's favourite currencies.	Select the radio button for favourite currencies	The chart will be updated with user's favourite currencies being comparing currencies		Pass
UTC069	Test currencies comparison by popular currencies	Select the radio button for popular currencies	The chart will be updated with popular currencies being comparing currencies		Pass
UTC070	Test currencies comparison by changing	Choose another currency for	The chart will be updated based on the		Pass

	base currency	base currency input	chosen base currency		
UTC071	Test currencies comparison by changing duration	Choose another duration for duration input	The chart will be updated based on the chosen duration		Pass
UTC072	Test currencies comparison by removing comparing currency	Press on remove icon to remove a currency from the selected set of comparing currencies	The chart will be updated based on the remaining comparing currencies		Pass
UTC073	Test currencies comparison by adding comparing currency	Press on add button to add a currency as comparing currency	The chart will be updated based on the updated set of comparing currencies		Pass
UTC074	Test currency comparison chart intractability with cursor hovering	Hover the cursor towards any bar in the bar chart	Details of the bar including the currency pair and exchange rate changes will be displayed.		Pass
UTC075	Test download currency comparison chart as svg	Press the download icon and select download svg	The svg file with the chart image will be installed on user's machine		Pass

UTC076	Test download currency comparison chart as png	Press the download icon and select download png	The png file with the chart image will be installed on user's machine		Pass
UTC077	Test download currency comparison chart as csv	Press the download icon and select download csv	The csv file with the chart data will be installed on user's machine		Pass
UTC078	Test currencies correlation on initial page load.	Click on currency correlation analysis on sidebar	The chart will be loaded with popular currencies being selected.	UC09	Pass
UTC079	Test currencies correlation by user's favourite currencies.	Select the radio button for favourite currencies	The chart will be updated with user's favourite currencies being selected		Pass
UTC080	Test currencies correlation by popular currencies	Select the radio button for popular currencies	The chart will be updated with popular currencies being selected		Pass
UTC081	Test currencies correlation	Choose another	The chart will be updated based on the		Pass

	by changing duration	duration for duration input	chosen duration		
UTC082	Test currencies correlation by removing selected currency	Press on remove icon to remove a currency from the selected currencies	The chart will be updated based on the remaining selected currencies		Pass
UTC083	Test currencies correlation by adding currency	Press on add button to add a currency into the selected currencies	The chart will be updated based on the updated set of selected currencies		Pass
UTC084	Test currency correlation chart intractability with cursor hovering	Hover the cursor towards any data in the heatmap chart	Details of the data including the currency pair and exchange rate changes will be displayed.		Pass
UTC085	Test download currency correlation chart as svg	Press the download icon and select download svg	The svg file with the chart image will be installed on user's machine		Pass
UTC086	Test download currency correlation chart as png	Press the download icon and select download png	The png file with the chart image will be installed on user's machine		Pass

UTC087	Test download currency correlation chart as csv	Press the download icon and select download csv	The csv file with the chart data will be installed on user's machine		Pass
UTC088	Test convert currency on page initial load	Click on the converter at sidebar	From and to currency are filled with user's default currency and second currency respectively	UC10	Pass
UTC089	Test convert currency on amount input	Input a valid number into the amount field	Conversion result will be displayed		Pass
UTC090	Test convert currency on amount changes	Change the existing number of amount field	Conversion result will be updated based on the changed amount		Pass
UTC091	Test convert currency by changing from currency	Choose another currency for from currency input	Conversion result will be updated based on chosen from currency		Pass
UTC092	Test convert currency by changing to currency	Choose another currency for to currency input	Conversion result will be updated based		Pass

			on chosen to currency		
UTC093	Test convert currency with immediate exchange of currencies	Press exchange icon to instantly exchange the value of from currency with to currency	Conversion result will be updated based on the new pair of currencies		Pass
UTC094	Test view favourite currencies	Click on edit profile button	Previously chosen favourite currencies will be displayed as the data in its field	UC11	Pass
UTC095	Test add favourite currencies	Click on the favourite currencies selection input to select another currency and save changes	The system will prompt success message		Pass
UTC096	Test remove favourite currencies	Click on the cancel icon beside the selected currency and saves changes	The system will prompt success message		Pass
UTC097	Test view favourite currencies	Click on home button on the sidebar and	Comparison chart and table among		Pass

	comparison section on dashboard	scroll to bottom	favourite currencies based on user's default currency will be shown		
UTC098	Test view favourite currencies in currency selection	Click on any currency input in the system	Favourite currencies will be shown on top of other currencies		Pass
UTC100	Test create alert without selection of notification method	Uncheck both checkboxes of notify via email and app	System will prompt user to select at least one notification method	UC12	Pass
UTC101	Test create periodic alert	Choose any duration and check any notification methods and set alert	System will prompt success message		Pass
UTC102	Test create conditional alert with current rate meeting the condition	Select more than as condition and input the value of target rate to be smaller than the current rate	System will prompt the condition had already been met		Pass
UTC103	Test create conditional	Set alert without filling	System will prompt the		Pass

	alert for exchange rate with empty target rate	in the target exchange rate	target rate is required		
UTC104	Test create conditional alert for RSI value with empty target value	Set alert without filling in the target RSI value	System will prompt the target RSI value is required		Pass
UTC105	Test create conditional alert for RSI value by selection of overbought	Click on the radio button for overbought	The condition will be selected as more than while target RSI value will be inputted as 70		Pass
UTC106	Test create conditional alert for RSI value by selection of oversold	Click on the radio button for oversold	The condition will be selected as less than while target RSI value will be inputted as 30		Pass
UTC107	Test toggle periodic alert	Click on the switch button corresponding to the periodic alert for either exchange rate or RSI value	The periodic alert will be disabled or enabled depending on its current state	UC13	Pass

UTC108	Test toggle conditional alert with unmet condition	Click on the switch button corresponding to the conditional alert for either exchange rate or RSI value	The conditional alert will be disabled or enabled depending on its current state		Pass
UTC109	Test enable conditional alert with met condition	Click on the switch button corresponding to the conditional alert conditional with the condition already been met	The state of the switch remains unchanged.		Pass
UTC110	Test edit periodic alert	Click on edit button, change the period and save changes for either exchange rate or RSI value	System will prompt success message	UC14	Pass
UTC111	Test edit conditional alert with unmet condition	Click on edit button, change the condition and target value for either	System will prompt success message		Pass

		exchange rate or RSI value			
UTC112	Test edit conditional alert with met condition	Click on edit icon, change the condition to more than and target value to be smaller than the current rate or RSI value	System will prompt the condition has already been met		Pass
UTC113	Test delete alert	Click on delete icon corresponding to any alert	System will prompt the alert being deleted successfully	UC15	Pass
UTC114	Test sending email notification for daily periodic alert	Set daily periodic alert and wait until 8 a.m. and check email inbox	Email for periodic notification can be seen in inbox for every day 8 a.m.	UC16	Pass
UTC115	Test sending email notification for weekly periodic alert	Set weekly periodic alert and wait until Monday 8 a.m. and check email inbox	Email for periodic notification can be seen in inbox for every Monday 8 a.m.		Pass
UTC116	Test sending email notification	Set monthly periodic alert and change the	Email for periodic notification		Pass

	for monthly periodic alert	system time to first day 8 a.m. of the month and check email inbox	can be seen in inbox for every first day 8 a.m. of the month		
UTC117	Test sending email notification for conditional alert	Set conditional alert and manually change the alert condition to be met in database	Email for conditional notification can be seen in email inbox		Pass
UTC118	Test sending in-app notification for daily periodic alert	Set daily periodic alert and view apps notification on 8 a.m.	Notification of daily update will be shown in notification modal.	UC17	Pass
UTC119	Test sending in-app notification for weekly periodic alert	Set weekly periodic alert and view apps notification on Monday 8a.m..	Notification of weekly update will be shown in notification modal.		Pass
UTC120	Test sending in-app notification for monthly periodic alert	Set monthly periodic alert and view apps notification after changing system time to first day 8a.m. of the month. .	Notification of monthly update will be shown in notification modal.		Pass

UTC121	Test sending in-app notification for conditional alert	Set conditional alert and manually change the alert condition to be met in database	Notification for conditional alert will be shown in notification modal.		Pass

6.3 Integration Testing

The second testing strategy used is integration testing. Integration testing was conducted by testing the units of the currency exchange rate tracking system as groups, with the aim of ensuring seamless integration between different modules in the system.

Table 6-3: Integration Testing Test Cases

Test Case ID	**Test Case Name**	**Test Case Description**	**Status**
ITC01	Test Login Page (Unauthenticated)	Users are redirected to login page if they are unauthenticated.	Pass
ITC02	Test Login Page (Authenticated)	Users are redirected to dashboard page if they are authenticated	Pass
ITC03	Test Sign Up Page (Unauthenticated)	Users can navigate to sign up page from login page.	Pass
ITC04	Test Sign Up Page (Authenticated)	Users are redirected to dashboard page if they are authenticated	Pass
ITC05	Test Forget Password Page (Unauthenticated)	Users can navigate to forget password page from login page.	Pass
ITC06	Test Forget Password Page (Authenticated)	Users are redirected to dashboard page if they are authenticated	Pass
ITC07	Test Successful Login	Users are redirected to dashboard page if they login successfully.	Pass
ITC08	Test Successful Sign Up	Users are redirected to login page if they sign up their account successfully.	Pass
ITC09	Test Successful Account Verification	Users are redirected to login page from verify account page if users click on the link that sent to users' inbox for account verification.	Pass

ITC10	Test Successful Password Reset	Users are redirected to login page from reset password page if they reset password successfully.	Pass
ITC11	Test Google Sign In Page (Unauthenticated)	Users can navigate to Google authentication page from login page.	Pass
ITC12	Test Google Sign In Page (Authenticated)	Users are redirected to dashboard page if they are authenticated	Pass
ITC13	Test Successful Google Sign In	Users are redirected to add profile page if they sign in with their Google accounts successfully.	Pass
ITC14	Test Failed Google Sign In	Users are redirected back to login page if any error occurs during signing in with Google account.	Pass
ITC15	Test Logout	Users are redirected to login page if users logout their accounts.	Pass
ITC16	Test Add Profile Page (Accounts without Profile)	Users are redirected to add profile page if users do not have their profile being set previously.	Pass
ITC17	Test Add Profile Page (Accounts with Profile)	Users are redirected to home page if users have their profile being set previously.	Pass
ITC18	Test Successful Add Profile	Users are redirected to dashboard page if they add their profiles successfully.	Pass
ITC19	Test Change Password Page	Users can navigate to change password page if they are authenticated.	Pass
ITC20	Test Edit Profile Page	Users can navigate to edit profile page if they are authenticated	Pass
ITC21	Test Dashboard Page	Users can navigate to dashboard page if they are authenticated.	Pass
ITC22	Test Converter Page	Users can navigate to currency converter page if they are authenticated.	Pass

ITC23	Test Exchange Rate Changes Page	Users can navigate to currency exchange rate tracking page if they are authenticated.	Pass
ITC24	Test RSI Values Changes Page	Users can navigate to currency RSI values tracking page if they are authenticated.	Pass
ITC25	Test Currency Comparison Page	Users can navigate to currency comparison page if they are authenticated.	Pass
ITC26	Test Currency Correlation Analysis Page	Users can navigate to currency correlation analysis page if they are authenticated.	Pass
ITC27	Test Alert Setting Modal for Exchange Rate (Open)	Users can activate the alert setting modal on currency exchange rate changes page.	Pass
ITC28	Test Alert Setting Modal for Exchange Rate (Close)	Users can deactivate the alert setting modal on currency exchange rate changes page.	Pass
ITC29	Test Alert Setting Modal for RSI Values (Open)	Users can activate the alert setting modal on currency RSI values changes page.	Pass
ITC30	Test Alert Setting Modal for RSI Values (Close)	Users can deactivate the alert setting modal on currency RSI values changes page.	Pass
ITC31	Test Alert Setting Page	Users can navigate to alert setting page if they are authenticated.	Pass
ITC32	Test Edit Alert Modal (Open)	Users can activate the alert edit modal on alert setting page.	Pass
ITC33	Test Edit Alert Modal (Close)	Users can deactivate the alert edit modal on alert setting page.	Pass
ITC34	Test Notification Modal (Open)	Users can activate the notification modal from top bar if they are authenticated.	Pass

ITC35	Test Notification Modal (Close)	Users can deactivate the notification modal.	Pass
ITC36	Test Profile Modal (Open)	Users can activate the profile modal from top bar if they are authenticated.	Pass
ITC37	Test Profile Modal (Close)	Users can deactivate the profile modal.	Pass

6.4 User Acceptance Testing

User Acceptance Test (UAT) is being chosen as the third testing methodology for the currency exchange rate tracking system. This testing was performed by end-users based on a series of test cases. to assess the usability and learnability of the system. A total number of five participants were invited for the test, and they are requested to perform each task listed in the test cases. The time consumed and the result of each test case were recorded.

Table 6-4: User Acceptance Test (UAT) Test Cases

Test Case ID	Test Form ID	Test Module	Test Description
UATC01	F01 F13 F25 F37 F49	Authentication	Able to sign up new account
			Able to verify account
			Able to login the verified account
UATC02	F02 F14 F26 F38 F50	Google Authentication	Able to sign in using Google account
			Able to add profile after signing in with Google
UATC03	F03		Able to reset new password

	F15 F27 F39 F51	Reset Password	Able to login using new password
UATC04	F04 F16 F28 F40 F52	User Profile	Able to change personal information
			Able to change preference currencies
			Able to change password
UATC05	F05 F17 F29 F41 F53	Dashboard	Able to view the responding exchange rate data and chart
			Able to view the responding RSI data and chart
			Able to view the popular currencies comparison chart and table
			Able to view the favourite currencies comparison chart and table
			Able to change the selection of input currencies
UATC06	F06 F18 F30 F42 F54	Currency Converter	Able to get the correct conversion result
			Able to change the selection of input currencies or amount
UATC07	F07 F19	Currency Exchange Rate	Able to view the responding exchange rate chart

	F31 F43 F55	Tracking	
			Able to interact with the exchange rate chart
			Able to change the selection of input currencies or duration
			Able to set a periodic alert
			Able to set a conditional alert
UATC08	F08 F20 F32 F44 F56	Currency RSI Values Tracking	Able to view the responding RSI values chart
			Able to interact with the RSI values chart
			Able to change the selection of input currencies or duration
			Able to set a periodic alert
			Able to set a conditional alert
UATC09	F09 F21 F33 F45 F57	Currencies Comparison	Able to view the responding currency comparison chart and table
			Able to interact with the currency comparison chart
			Able to quick select all the favourite or popular currencies for comparison

			Able to change the selection of base currency and duration
			Able to add or remove currency from comparing currencies
UATC10	F10 F22 F34 F46 F58	Currencies Correlation Analysis	Able to view the responding currency correlation chart and table
			Able to interact with currency correlation chart
			Able to quick select all the favourite or popular currencies for analysis
			Able to add or remove currency for correlation analysis
			Able to change the input of duration
UATC11	F11 F23 F35 F47 F59	Alert	Able to toggle the alert’s status
			Able to delete the alert
			Able to modify the alert
			Able to modify the notes to be sent with the notification
UATC12	F12 F24	Notification	Able to receive email notification

	F36 F48 F60	(Through direct modification of system time and alert's condition for testing)	Able to view in-app notification

6.5 Usability Testing

Usability testing is being chosen as the last testing strategy for the currency exchange rate tracking system. This testing was performed by end-users based on a set of test scenarios which has been sent to the participants one day before the testing. The purpose of usability testing is to further evaluate the usability of the system by testing whether users are able to perform basic operations on the system according to the test scenarios without external guidance. A total number of five participants were invited for the test, and they were requested to perform the whole testing without any external guidance. Before the testing, participants were asked to fill out the questionnaire to obtain their demographic data and also experience in using similar system. After the testing, the participants were requested to answer the user satisfaction survey to obtain their feedback. The table below shows the test scenarios that were performed by the participants.

Table 6-5: Usability Test Scenarios

No	**Test Scenarios**
1	Sign Up & Login Account You wish to use the currency exchange rate tracking system, but you do not have any registered accounts in the system. What would you do to sign up for an account and log into the system?
2	Edit Profile (Default & Second Currency) After you log in, you are redirected to the dashboard page. You notice that the dashboard is loaded with charts and exchange rate data based on default currencies that you set earlier. Now, you wish to set the base

	currency to MYR and the second currency to SGD so this pair of currencies will be used to load the chart every time you load the page. What would you do?
3	Currency Converter You wonder how much 130 Singaporean dollars (SGD) converts to in Malaysian Ringgit (MYR). What would you do?
4	Exchange Rate Changes You wish to view the historical exchange rate chart for Malaysian Ringgit (MYR) to Japanese Yen (JPY) for the past six months. What would you do?
5	Set Periodic Alert: Exchange Rate You wish to be notified via email to get the daily update on the currency exchange rate for Malaysian Ringgit (MYR) to Japanese Yen (JPY). What would you do to set the alert?
6	RSI Values Changes You wish to view the historical relative strength index (RSI) chart for the Malaysian Ringgit (MYR) to the United States dollar (USD) to check the performance of this pair of currencies over the last two years. What would you do?
7	Set Conditional Alert: RSI You want to buy in the forex of Malaysian Ringgit (MYR) / United States dollar (USD). You wish to be notified in the application when this pair of currencies is oversold. What would you do to set up the alert?
8	Currency Comparison You are wondering how well the Malaysian Ringgit (MYR) has performed over the last three months. You wish to use the currency comparison feature of the system to compare Malaysian Ringgit (MYR)

	with currencies like Singaporean Dollar (SGD), United State Dollar (USD), Chinese Yuan (CNY), Great British Bound (GBP) and Thai Baht (THB). What would you do?
9	Set Favourite Currencies From the previous test, you noticed that there is an option to do a comparison by your favourite currencies. Thus, you wish to set the currencies used for the comparison above as your favourite currencies so you can perform a quick selection next time. What would you do to set your favourite currencies?
10	Currency Correlation Analysis Now you have your favourite currencies set, you are curious about the correlation among these currencies like which currency is performing well also when a specific currency is outperforming. Thus, you wish to use the currency correlation analysis feature of the system to obtain insights into the correlation among your favourite currencies over the last three months. What would you do?
11	Toggle Alert For temporarily, you wish to disable the periodic alert for the exchange rate that you set previously. What would you do?
12	Edit Alert You noticed that the RSI value of the Malaysian Ringgit (MYR) to the United States dollar (USD) is close to the overbought line rather than the oversold line. You wish to be notified when the RSI value of this pair of currencies hits the overbought line so you can sell out your forex at the right time. set? At the same time, you also wish to send your notes together with the notification to remind yourself of performing the forex trading activity. What would you do to modify the condition and notes of the alert that you have previously set?

The summary and results of the usability testing are shown in the tables below.

Table 6-6: Participants Demographic Summary

Attribute	Value	Percentage
Age	Below 18 years old	0 %
	19 to 30 years old	80 %
	31 to 50 years old	20%
	Above 51 years old	0%
Gender	Male	80 %
	Female	20 %
Occupation	Students (Software Engineering)	80 %
	Primary School Teacher	20 %
Relevant Experience	Yes	60 %
	No	40 %
System Used	OctaFX	40 %
	XE	20 %
	Moomoo	20 %
Frequency Checking Exchange Rate	Always	40 %
	Sometimes	20 %
	Seldom	20 %
	Never	20 %
Purpose	Forex Trading	40 %
	Purchase Oversea Products	40 %
	Travel Planning	40 %
	Cross-Border Transaction	20 %

Table 6-7: Results of Users Satisfaction Survey

Questions	Participant No.					Average
	1	2	3	4	5	
I think the system serves its responsibility as an exchange rate tracking system	3	4	4	4	4	
I think the system does not cover the basic requirements for exchange rate tracking	3	4	4	3	4	
I think the system is user-friendly	3	4	4	3	4	
I think the system is complex and confusing	4	3	4	4	3	
I think the charts in the system are helpful for gaining insights	4	3	4	3	4	
I think the data visualization method used in the system is terrible	4	4	4	4	4	
I think the system is flexible in managing alerts	3	4	4	4	4	
I think the system is inflexible in creating and managing alerts.	3	4	4	4	4	
I am satisfied with the additional features added (Currency correlation analysis and RSI chart visualization etc.)	4	4	3	2	3	
I think the additional features are useless	4	2	4	2	3	
Sum	35	36	39	33	37	36
SUS Score	87.5	90	97.5	82.5	92.5	90

Table 6-8: Summary of Participant Comments

		Percentage
Positive Feedback	- Good user interface - Attractive & interactive charts - Efficient currencies searching	40 % 40 % 20 %
Negative feedback	- Not supporting intraday exchange rate updates - Too less forex indicator - Long loading time for heatmap chart	40 % 20 % 20 %

CHAPTER 7

CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion

This project began in June 2023 and ended in April 2024, which lasted for about seven months, excluding the internship period in October semester last year. At the beginning stage of the project, the main objectives, the problem statement, and the overall project scope are identified. Moving on to the planning stage, a literature review is conducted to decide on the methodology, programming framework, and functional requirements to be used in this project. After that, the use case diagram, system architecture design, and entity relationship diagram are defined in the design phase. A prototype is also developed using Figma to have a rough idea of how the user interface of the system looks. Using the developed prototype, feedback is gathered from the supervisor and integrated into this project by refining the requirements and diagrams of the project. The next stage is the implementation stage, which is the development of the currency exchange rate tracking system using Flask as the backend framework. Lastly, unit testing, integration testing, user acceptance testing, and usability testing are conducted to evaluate the functionality and usability of the developed system.

In this chapter, the objective fulfilment of the project, the limitations, and future works about this project will be discussed and outlined.

7.2 Objective Fulfilment

All of the objectives have been successfully achieved upon the completion of the project. The first objective of the project is to understand the requirements of the currency exchange rate tracking system. In Chapter 4 which is the topic related to project specifications and design, the functional requirements and use cases are identified through the conduction of a literature review on existing exchange rate tracking applications.

The second objective of this project would be to develop a web-based application for the currency exchange rate tracking system which is also fulfilled after the implementation and testing stage. The codebase of the developed system would be deliverable for this objective, and it is available to the public on my personal GitHub profile.

The third objective of this project would be to evaluate the currency exchange rate tracking system using different testing strategies. At the end of the project, this objective is achieved after implementing various testing methodologies like unit testing, integration testing, user acceptance testing and usability testing. The developed system has passed all of the test cases written in Chapter 6 which has proven the functionality and usability of this system.

7.3 Limitations

Although all of the project's objectives were effectively met, it is undeniable that the system developed is still suffering some limitations.

The first limitation is the currency exchange rate tracking system does not support intraday exchange rate tracking features currently due to the unavailability of intraday updates for Fast Forex API, which is the external API provider for this project, and the unaffordable price offered for other intraday API. However, based on their documentation, they are planning to provide an intraday exchange rate update shortly. By that time, the system could be refined for integration of their intraday API.

The second limitation is the lack of financial indicators to attract forex traders. The currency exchange rate tracking system developed only supports a common indicator which is the relative strength index (RSI). In the future, more financial indicators that come in handy to support decision-making in forex trading like moving averages, Bollinger bands, Fibonacci retracement levels, and so forth can be added to the system.

The third limitation is the system only supports web applications currently which is inconvenient for mobile phone users. Therefore, a mobile currency exchange rate tracking application dedicated to mobile phone users can be developed and it should be allowed to establish communications with the Flask backend server which consists of the RESTful APIs.

7.4 Recommendations for Future Work

Besides the solutions for the limitations mentioned above, the currency exchange rate tracking system can be further improved by adding extra functionalities.

The first recommendation for future work of the currency exchange rate tracking system is to add a financial news corner by integrating news API from reputable sources. By having this feature, users can stay informed about the up-to-date significant economic events or policy changes that may impact the currency market. Furthermore, this valuable information can help users understand the rationale behind the movements of the currency market and thus support them in making informed trading decisions.

The second recommendation for future work of the system is to incorporate artificial intelligence predictions for the currency exchange rates. By utilizing AI algorithms, the system can evaluate a massive volume of historical and identify the complicated patterns that may impact the currency exchange rates. Our own predictive machine learning model can be trained by feeding large amounts of historical data into it in the Flask backend server. Another choice is to integrate a well-trained and high-accuracy model from external sources into the currency exchange rate tracking system.

The third recommendation for future work of the system is to expand the system's scope for covering cryptocurrencies. In this day and age, along with the advancement of microprocessor chips, cryptocurrencies have emerged as a prominent asset class which is an appealing option to worldwide traders. Fast Forex API which is the API currently used by the system provides the APIs and

documentation for retrieving the latest exchange rate of the cryptocurrency. Therefore, the integration of their cryptocurrency API into the system can be considered a wise option due to the vast community of cryptocurrency traders.

REFERENCES

DashboardFox, 2020. *Benefits and Challenges of Dashboards*. [online] Available at: <https://dashboardfox.com/guides/business-dashboard-guide/benefits-and-challenges-of-dashboards/> [Accessed 2 August 2023].

Lucidchart, 2018. *The Pros and cons of waterfall methodology*. [online] Available at: <https://www.lucidchart.com/blog/pros-and-cons-of-waterfall-methodology> [Accessed: 27 August 2023].

Martin, M., 2019. *Prototyping Model in Software Engineering: Methodology, Process, Approach*. [online] Guru99.com. Available at: <https://www.guru99.com/software-engineering-prototyping-model.html> [Accessed 8 August 2023].

Pratama, M.A.T. and Cahyadi, A.T., 2020. Effect of User Interface and User Experience on Application Sales. *IOP Conference Series: Materials Science and Engineering*, 879. https://doi.org/10.1088/1757-899x/879/1/012133.

Python Tutorial, 2021. *What is Flask Python - Python Tutorial*. [online] Available at: <https://pythonbasics.org/what-is-flask-python/> [Accessed: 27 August 2023].

Suduc, A.-M., Bizoi, M. and Filip, F.G., 2010. User Awareness about Information Systems Usability. *Studies in Informatics and Control*, 19(2). https://doi.org/10.24846/v19i2y201004.

Tutorialspoint, n.d. *SDLC - iterative model.* [online] Available at: <https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm> [Accessed: 27 August 2023].

W3schools, n.d. *Introduction to Django*. [online] Available at: <https://www.w3schools.com/django/django_intro.php> [Accessed: 27 August 2023].

West, Z. (2023). *Calculating the RSI in Python: 3 Ways to Predict Market Status & Price Movement*. [online] αlphαrithms. Available at: <https://www.alpharithms.com/relative-strength-index-rsi-in-python-470209/#google_vignette> [Accessed 19 Apr. 2024].

William, 2022. *Web Application Architecture: The Latest Guide 2022*. [online] ClickIT. Available at: <https://www.clickittech.com/devops/web-application-architecture/> [Accessed 2 August 2023].

Yellowfin, 2022. *8 Business Benefits of BI Dashboards.* [online] Available at: <https://www.yellowfinbi.com/blog/bi-dashboards-business-

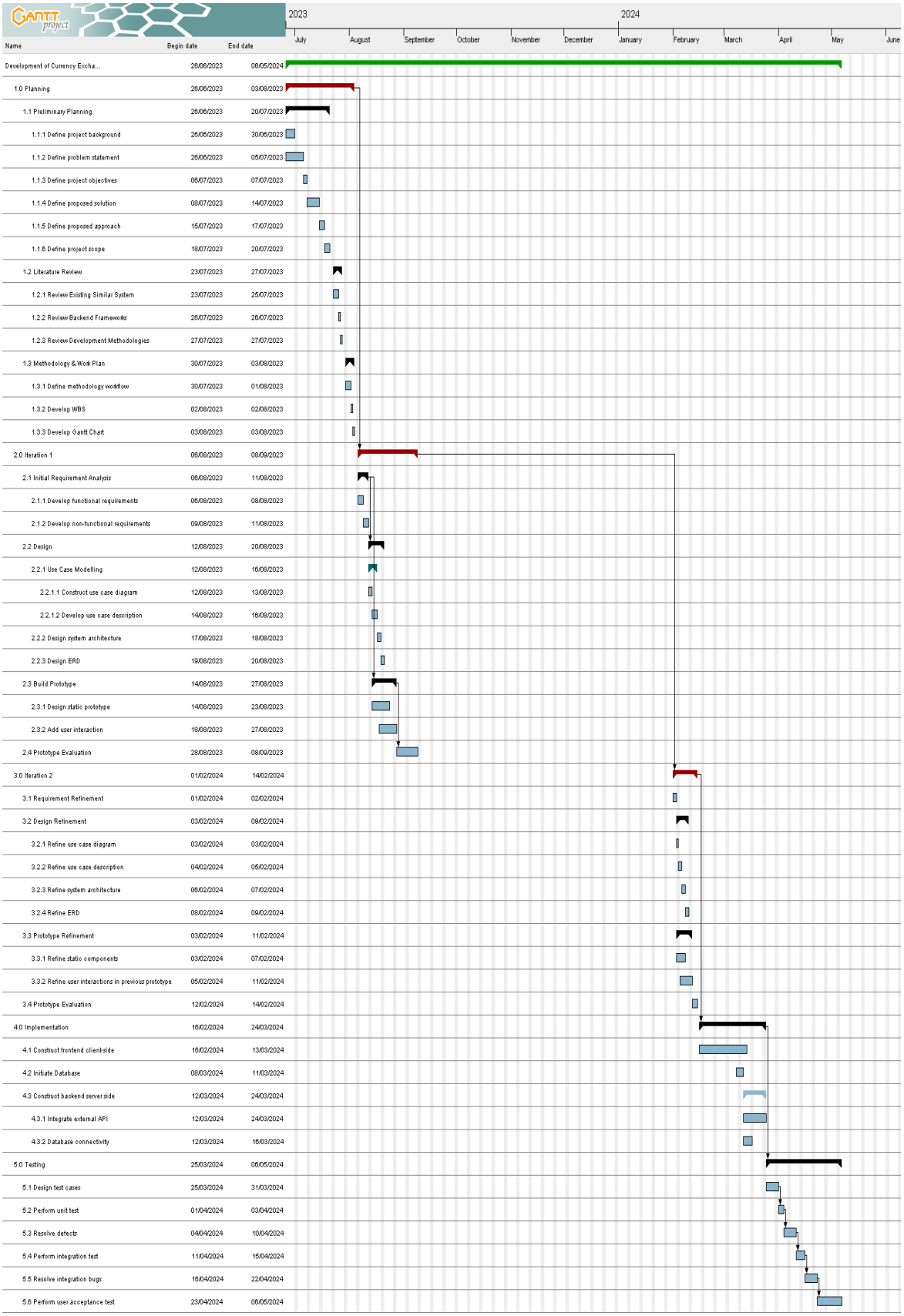
benefits#:~:text=Dashboards%20can%20help%20users%20monitor>
[Accessed 2 August 2023].

APPENDICES

Appendix A: Work Breakdown Structure (WBS)

- Development of Currency Exchange Rate Tracking System
  - 1.0 Planning
    - 1.1 Preliminary Planning
      - 1.1.1 Define project background
      - 1.1.2 Define problem statement
      - 1.1.3 Define project objectives
      - 1.1.4 Define proposed solutions
      - 1.1.5 Define proposed approach
      - 1.1.6 Define project scope
    - 1.2 Literature Review
      - 1.2.1 Review existing similar system
      - 1.2.2 Review backend frameworks
      - 1.2.3 Review development methodologies
    - 1.3 Methodologies & Work Plan
      - 1.3.1 Define methodology workflow
      - 1.3.2 Develop WBS
      - 1.3.3 Develop Gantt Chart
  - 2.0 Iteration 1
    - 2.1 Initial Requirement Analysis
      - 2.1.1 Develop functional requirements
      - 2.1.2 Develop non-functional requirements
    - 2.2 Design
      - 2.2.1 Use Case Modelling
        - 2.2.1.1 Construct use case diagram
        - 2.2.1.2 Develop use case description
      - 2.2.2 Design system architecture
      - 2.2.3 Design ERD
    - 2.3 Build Prototype
      - 2.3.1 Design static prototype
      - 2.3.2 Add user interaction
    - 2.4 Prototype Evaluation
  - 3.0 Iteration 2
    - 3.1 Requirement Refinement
    - 3.2 Design Refinement
      - 3.2.1 Refine use case diagram
      - 3.2.2 Refine use case description
      - 3.2.3 Refine system architecture
      - 3.2.4 Refine ERD
    - 3.3 Prototype Refinement
      - 3.3.1 Refine static components
      - 3.3.2 Refine user interaction
    - 3.4 Prototype Evaluation
  - 4.0 Implementation
    - 4.1 Construct frontend client side
    - 4.2 Initiate Database
    - 4.3 Construct backend server side
      - 4.3.1 Integrate external API
      - 4.3.2 Database connectivity
  - 5.0 Testing
    - 5.1 Design test cases
    - 5.2 Perform unit test
    - 5.3 Resolve defects
    - 5.4 Perform integration test
    - 5.5 Resolve integration bugs
    - 5.6 Perform user acceptance test

Appendix B: Gantt Chart



Appendix C: Consent Forms

Consent Form on UAT and Usability Testing

My name is Phan Zhan Yan, and I am currently studying Bachelor of Science (Honours) in Software Engineering at Universiti Tunku Abdul Rahman (UTAR).

The title of my final year project is about the development of a currency exchange rate tracking system. The purpose of this project is to demonstrate the currency exchange rate data with various data visualization methods to provide users with as many insights as possible at first glance. The system is also embedded with the features of managing alerts that notify users about exchange rate changes to provide the highest flexibility.

To evaluate the usability of the system, I would like to invite you to participate in the user acceptance test (UAT) and usability test of my currency exchange rate tracking system. Your valuable feedback will play a crucial role in helping me refine the functionality of the system in the future.

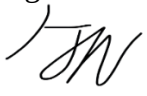
If you agree to participate, you will receive a testing guideline that includes the testing scenario that you will be asked to perform while conducting the usability test. Both the user acceptance test and usability test will be conducted physically by using my device on either day between April 14th and April 20th, depending on your schedule. During the testing, you possess the right to stop your participation at any time and you may take a break whenever you need. You will also be requested to fill in a questionnaire to obtain feedback regarding the system that you tested. Your personal information including age and occupation will be collected and it is solely for internal use. The findings of test results will be included in my final-year project report. If you have any further inquiries, feel free to reach out to me at zhanyan.1124@1utar.my or via my contact number, +60-109390564.

I have read and understand on the information above, and I agree to take part in this testing. I am aware of my rights as a participant in the testing. My signature does not represent a granting of any legal rights. Additionally, I understand that a copy of the informed consent form would be required for my records.

Date: 11/4/2024

Name: Yap Chia Hau

Signature:



Thank you for your attention and interest! Your participation is deeply appreciated as it will contribute to the completion of my final year project.

Consent Form on UAT and Usability Testing

My name is Phan Zhan Yan, and I am currently studying Bachelor of Science (Honours) in Software Engineering at Universiti Tunku Abdul Rahman (UTAR).

The title of my final year project is about the development of a currency exchange rate tracking system. The purpose of this project is to demonstrate the currency exchange rate data with various data visualization methods to provide users with as many insights as possible at first glance. The system is also embedded with the features of managing alerts that notify users about exchange rate changes to provide the highest flexibility.

To evaluate the usability of the system, I would like to invite you to participate in the user acceptance test (UAT) and usability test of my currency exchange rate tracking system. Your valuable feedback will play a crucial role in helping me refine the functionality of the system in the future.

If you agree to participate, you will receive a testing guideline that includes the testing scenario that you will be asked to perform while conducting the usability test. Both the user acceptance test and usability test will be conducted physically by using my device on either day between April 14th and April 20th, depending on your schedule. During the testing, you possess the right to stop your participation at any time and you may take a break whenever you need. You will also be requested to fill in a questionnaire to obtain feedback regarding the system that you tested. Your personal information including age and occupation will be collected and it is solely for internal use. The findings of test results will be included in my final-year project report. If you have any further inquiries, feel free to reach out to me at zhanyan.1124@1utar.my or via my contact number, +60-109390564.

I have read and understand on the information above, and I agree to take part in this testing. I am aware of my rights as a participant in the testing. My signature does not represent a granting of any legal rights. Additionally, I understand that a copy of the informed consent form would be required for my records.

Date: 17/4/2024

Name: Ngeh Kai Bin

Signature:

Thank you for your attention and interest! Your participation is deeply appreciated as it will contribute to the completion of my final year project.

Consent Form on UAT and Usability Testing

My name is Phan Zhan Yan, and I am currently studying Bachelor of Science (Honours) in Software Engineering at Universiti Tunku Abdul Rahman (UTAR).

The title of my final year project is about the development of a currency exchange rate tracking system. The purpose of this project is to demonstrate the currency exchange rate data with various data visualization methods to provide users with as many insights as possible at first glance. The system is also embedded with the features of managing alerts that notify users about exchange rate changes to provide the highest flexibility.

To evaluate the usability of the system, I would like to invite you to participate in the user acceptance test (UAT) and usability test of my currency exchange rate tracking system. Your valuable feedback will play a crucial role in helping me refine the functionality of the system in the future.

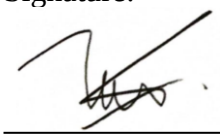
If you agree to participate, you will receive a testing guideline that includes the testing scenario that you will be asked to perform while conducting the usability test. Both the user acceptance test and usability test will be conducted physically by using my device on either day between April 14th and April 20th, depending on your schedule. During the testing, you possess the right to stop your participation at any time and you may take a break whenever you need. You will also be requested to fill in a questionnaire to obtain feedback regarding the system that you tested. Your personal information including age and occupation will be collected and it is solely for internal use. The findings of test results will be included in my final-year project report. If you have any further inquiries, feel free to reach out to me at zhanyan.1124@1utar.my or via my contact number, +60-109390564.

I have read and understand on the information above, and I agree to take part in this testing. I am aware of my rights as a participant in the testing. My signature does not represent a granting of any legal rights. Additionally, I understand that a copy of the informed consent form would be required for my records.

Date: 17/4/2024

Name: Tham Kar Weng

Signature:



Thank you for your attention and interest! Your participation is deeply appreciated as it will contribute to the completion of my final year project.

Consent Form on UAT and Usability Testing

My name is Phan Zhan Yan, and I am currently studying Bachelor of Science (Honours) in Software Engineering at Universiti Tunku Abdul Rahman (UTAR).

The title of my final year project is about the development of a currency exchange rate tracking system. The purpose of this project is to demonstrate the currency exchange rate data with various data visualization methods to provide users with as many insights as possible at first glance. The system is also embedded with the features of managing alerts that notify users about exchange rate changes to provide the highest flexibility.

To evaluate the usability of the system, I would like to invite you to participate in the user acceptance test (UAT) and usability test of my currency exchange rate tracking system. Your valuable feedback will play a crucial role in helping me refine the functionality of the system in the future.

If you agree to participate, you will receive a testing guideline that includes the testing scenario that you will be asked to perform while conducting the usability test. Both the user acceptance test and usability test will be conducted physically by using my device on either day between April 14th and April 20th, depending on your schedule. During the testing, you possess the right to stop your participation at any time and you may take a break whenever you need. You will also be requested to fill in a questionnaire to obtain feedback regarding the system that you tested. Your personal information including age and occupation will be collected and it is solely for internal use. The findings of test results will be included in my final-year project report. If you have any further inquiries, feel free to reach out to me at zhanyan.1124@1utar.my or via my contact number, +60-109390564.

I have read and understand on the information above, and I agree to take part in this testing. I am aware of my rights as a participant in the testing. My signature does not represent a granting of any legal rights. Additionally, I understand that a copy of the informed consent form would be required for my records.

Date: 18/4/2024

Name: Yeoh Wei Bin

Signature:

Thank you for your attention and interest! Your participation is deeply appreciated as it will contribute to the completion of my final year project.

Consent Form on UAT and Usability Testing

My name is Phan Zhan Yan, and I am currently studying Bachelor of Science (Honours) in Software Engineering at Universiti Tunku Abdul Rahman (UTAR).

The title of my final year project is about the development of a currency exchange rate tracking system. The purpose of this project is to demonstrate the currency exchange rate data with various data visualization methods to provide users with as many insights as possible at first glance. The system is also embedded with the features of managing alerts that notify users about exchange rate changes to provide the highest flexibility.

To evaluate the usability of the system, I would like to invite you to participate in the user acceptance test (UAT) and usability test of my currency exchange rate tracking system. Your valuable feedback will play a crucial role in helping me refine the functionality of the system in the future.

If you agree to participate, you will receive a testing guideline that includes the testing scenario that you will be asked to perform while conducting the usability test. Both the user acceptance test and usability test will be conducted physically by using my device on either day between April 14th and April 20th, depending on your schedule. During the testing, you possess the right to stop your participation at any time and you may take a break whenever you need. You will also be requested to fill in a questionnaire to obtain feedback regarding the system that you tested. Your personal information including age and occupation will be collected and it is solely for internal use. The findings of test results will be included in my final-year project report. If you have any further inquiries, feel free to reach out to me at zhanyan.1124@1utar.my or via my contact number, +60-109390564.

I have read and understand on the information above, and I agree to take part in this testing. I am aware of my rights as a participant in the testing. My signature does not represent a granting of any legal rights. Additionally, I understand that a copy of the informed consent form would be required for my records.

Date: 13/4/2024

Name: Mok Lai Hoong

Signature:

Thank you for your attention and interest! Your participation is deeply appreciated as it will contribute to the completion of my final year project.

Appendix D: Result of UAT

Participant ID	Participant Name	Date
1	Yap Chia Hau	15/4/2024
2	Ngeh Kai Bin	17/4/2024
3	Tham Kar Weng	17/4/2024
4	Yeoh Wei Bin	18/4/2024
5	Mok Lai Hoong	20/4/2024

Test Case ID	UATC01	**Test Form ID**	F01	**Participant ID**	1
Start Time	1:03 p.m.	**End Time**		1:07 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Authentication	Able to sign up new account	Pass		-	
	Able to verify account	Pass		-	
	Able to login the verified account	Pass		-	

Test Case ID	UATC02	**Test Form ID**	F02	**Participant ID**	1
Start Time	1:09 p.m.	**End Time**		1:11 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Google Authentication	Able to sign in using Google account	Pass		-	
	Able to add profile after signing in with Google	Pass		-	

Test Case ID	UATC03	**Test Form ID**	F03	**Participant ID**	1
Start Time	1:11 p.m.	**End Time**		1:13 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Reset Password	Able to reset new password	Pass		-	
	Able to login using new password	Pass		-	

Test Case ID	UATC04	**Test Form ID**	F04	**Participant ID**	1
Start Time	1:13 p.m.	**End Time**		1:14 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
User Profile	Able to change personal information	Pass		-	
	Able to change preference currencies	Pass		-	
	Able to change password	Pass		-	

Test Case ID	UATC05	**Test Form ID**	F05	**Participant ID**	1
Start Time	1:14 p.m.	**End Time**		1:16 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Dashboard	Able to view the responding exchange rate data and chart	Pass		-	
	Able to view the responding RSI data and chart	Pass		-	
	Able to view the popular currencies comparison chart and table	Pass		-	

	Able to view the favourite currencies comparison chart and table	Pass	-
	Able to change the selection of input currencies	Pass	-

Test Case ID	UATC06	**Test Form ID**	F06	**Participant ID**	1
Start Time	1:16 p.m.	**End Time**		1:17 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency Converter	Able to get the correct conversion result	Pass		-	
	Able to change the selection of input currencies and amount	Pass		-	

Test Case ID	UATC07	**Test Form ID**	F07	**Participant ID**	1
Start Time	1:17 p.m.	**End Time**		1:19 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency Exchange Rate Tracking	Able to view the responding exchange rate chart	Pass		-	
	Able to interact with the exchange rate chart	Pass		-	
	Able to change the selection of input currencies or duration	Pass		-	
	Able to set a periodic alert	Pass		-	
	Able to set a conditional alert	Pass		-	

Test Case ID	UATC08	**Test Form ID**	F08	**Participant ID**	1
Start Time	1:19 p.m.	**End Time**		1:20 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency RSI Values Tracking	Able to view the responding RSI values chart	Pass		-	
	Able to interact with the RSI values chart	Pass		-	
	Able to change the selection of input currencies or duration	Pass		-	
	Able to set a periodic alert	Pass		-	
	Able to set a conditional alert	Pass		-	

Test Case ID	UATC09	**Test Form ID**	F09	**Participant ID**	1
Start Time	1:21 p.m.	**End Time**		1:23 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currencies Comparison	Able to view the responding currency comparison chart and table	Pass		-	
	Able to interact with the currency comparison chart	Pass		-	
	Able to quick select all the favourite or popular currencies for comparison	Pass		-	
	Able to change the selection of base currency and duration	Pass		-	

	Able to add or remove currency from comparing currencies	Pass	-

Test Case ID	UATC10	**Test Form ID**	F10	**Participant ID**	1
Start Time	1:23 p.m.	**End Time**		1:24 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currencies Correlation Analysis	Able to view the responding currency correlation chart and table	Pass		-	
	Able to interact with the currency correlation chart	Pass		-	
	Able to quick select all the favourite or popular currencies for analysis	Pass		-	
	Able to add or remove currency for correlation analysis	Pass		-	
	Able to change the input of duration	Pass		-	

Test Case ID	UATC11	**Test Form ID**	F11	**Participant ID**	1
Start Time	1:24 p.m.	**End Time**		1:25 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Alert	Able to toggle the alert's status	Pass		-	
	Able to delete the alert	Pass		-	
	Able to modify the alert	Pass		-	
	Able to modify the notes to be sent with notification	Pass		-	

Test Case ID	UATC12	**Test Form ID**	F12	**Participant ID**	1
Start Time	1:27 p.m.	**End Time**		1:27 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Notification	Able to receive email notification	Pass		-	
	Able to view in-app notifications	Pass		-	

Test Case ID	UATC01	**Test Form ID**	F13	**Participant ID**	2
Start Time	2:42 p.m.	**End Time**		2:46 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Authentication	Able to sign up new account	Pass		-	
	Able to verify account	Pass		-	
	Able to login the verified account	Pass		-	

Test Case ID	UATC02	**Test Form ID**	F14	**Participant ID**	2
Start Time	2:48 p.m.	**End Time**		2:49 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Google Authentication	Able to sign in using Google account	Pass		-	
	Able to add profile after signing in with Google	Pass		-	

Test Case ID	UATC03	**Test Form ID**	F15	**Participant ID**	2
Start Time	2:49 p.m.	**End Time**		2:52 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Reset Password	Able to reset new password	Pass		-	
	Able to login using new password	Pass		-	

Test Case ID	UATC04	**Test Form ID**	F16	**Participant ID**	2
Start Time	2:53 p.m.	**End Time**		2:55 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
User Profile	Able to change personal information	Pass		-	
	Able to change preference currencies	Pass		-	
	Able to change password	Pass		-	

Test Case ID	UATC05	**Test Form ID**	F17	**Participant ID**	2
Start Time	2:55 p.m.	**End Time**		2:57 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Dashboard	Able to view the responding exchange rate data and chart	Pass		-	
	Able to view the responding RSI data and chart	Pass		-	
	Able to view the popular currencies comparison chart and table	Pass		-	

	Able to view the favourite currencies comparison chart and table	Pass	-
	Able to change the selection of input currencies	Pass	-

Test Case ID	UATC06	**Test Form ID**	F18	**Participant ID**	2
Start Time	2:57 p.m.	**End Time**		2:57 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency Converter	Able to get the correct conversion result	Pass		-	
	Able to change the selection of input currencies and amount	Pass		-	

Test Case ID	UATC07	**Test Form ID**	F19	**Participant ID**	2
Start Time	2:57 p.m.	**End Time**		2:59 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency Exchange Rate Tracking	Able to view the responding exchange rate chart	Pass		-	
	Able to interact with the exchange rate chart	Pass		-	
	Able to change the selection of input currencies or duration	Pass		-	
	Able to set a periodic alert	Pass		-	
	Able to set a conditional alert	Pass		-	

Test Case ID	UATC08	**Test Form ID**	F20	**Participant ID**	2
Start Time	3:00 p.m.	**End Time**		3:01 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency RSI Values Tracking	Able to view the responding RSI values chart	Pass		-	
	Able to interact with the RSI values chart	Pass		-	
	Able to change the selection of input currencies or duration	Pass		-	
	Able to set a periodic alert	Pass		-	
	Able to set a conditional alert	Pass		-	

Test Case ID	UATC09	**Test Form ID**	F21	**Participant ID**	2
Start Time	3:01 p.m.	**End Time**		3:02 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currencies Comparison	Able to view the responding currency comparison chart and table	Pass		-	
	Able to interact with the currency comparison chart	Pass		-	
	Able to quick select all the favourite or popular currencies for comparison	Pass		-	
	Able to change the selection of base currency and duration	Pass		-	

	Able to add or remove currency from comparing currencies	Pass	-

Test Case ID	UATC10	**Test Form ID**	F22	**Participant ID**	2
Start Time	3:02 p.m.	**End Time**		3:05 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currencies Correlation Analysis	Able to view the responding currency correlation chart and table	Pass		-	
	Able to interact with the currency correlation chart	Pass		-	
	Able to quick select all the favourite or popular currencies for analysis	Pass		-	
	Able to add or remove currency for correlation analysis	Pass		-	
	Able to change the input of duration	Pass		-	

Test Case ID	UATC11	**Test Form ID**	F23	**Participant ID**	2
Start Time	3:05 p.m.	**End Time**		3:07 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Alert	Able to toggle the alert's status	Pass		-	
	Able to delete the alert	Pass		-	
	Able to modify the alert	Pass		-	
	Able to modify the notes to be sent with notification	Pass		-	

Test Case ID	UATC12	**Test Form ID**	F24	**Participant ID**	2
Start Time	3:10 p.m.	**End Time**		3:10 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Notification	Able to receive email notification	Pass		-	
	Able to view in-app notifications	Pass		-	

Test Case ID	UATC01	**Test Form ID**	F25	**Participant ID**	3
Start Time	3:15 p.m.	**End Time**		3:17 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Authentication	Able to sign up new account	Pass		-	
	Able to verify account	Pass		-	
	Able to login the verified account	Pass		-	

Test Case ID	UATC02	**Test Form ID**	F26	**Participant ID**	3
Start Time	3:17 p.m.	**End Time**		3:17 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Google Authentication	Able to sign in using Google account	Pass		-	
	Able to add profile after signing in with Google	Pass		-	

Test Case ID	UATC03	**Test Form ID**	F27	**Participant ID**	3
Start Time	3:18 p.m.	**End Time**		3:19 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Reset Password	Able to reset new password	Pass		-	
	Able to login using new password	Pass		-	

Test Case ID	UATC04	**Test Form ID**	F28	**Participant ID**	3
Start Time	3:20 p.m.	**End Time**		3:20 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
User Profile	Able to change personal information	Pass		-	
	Able to change preference currencies	Pass		-	
	Able to change password	Pass		-	

Test Case ID	UATC05	**Test Form ID**	F29	**Participant ID**	3
Start Time	3:20 p.m.	**End Time**		3:21 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Dashboard	Able to view the responding exchange rate data and chart	Pass		-	
	Able to view the responding RSI data and chart	Pass		-	
	Able to view the popular currencies comparison chart and table	Pass		-	

	Able to view the favourite currencies comparison chart and table	Pass	-
	Able to change the selection of input currencies	Pass	-

Test Case ID	UATC06	**Test Form ID**	F30	**Participant ID**	3
Start Time	3:21 p.m.	**End Time**		3:22 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency Converter	Able to get the correct conversion result	Pass		-	
	Able to change the selection of input currencies and amount	Pass		-	

Test Case ID	UATC07	**Test Form ID**	F31	**Participant ID**	3
Start Time	3:22 p.m.	**End Time**		3:23 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency Exchange Rate Tracking	Able to view the responding exchange rate chart	Pass		-	
	Able to interact with the exchange rate chart	Pass		-	
	Able to change the selection of input currencies or duration	Pass		-	
	Able to set a periodic alert	Pass		-	
	Able to set a conditional alert	Pass		-	

Test Case ID	UATC08	**Test Form ID**	F32	**Participant ID**	3
Start Time	3:23 p.m.	**End Time**		3:23 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency RSI Values Tracking	Able to view the responding RSI values chart	Pass		-	
	Able to interact with the RSI values chart	Pass		-	
	Able to change the selection of input currencies or duration	Pass		-	
	Able to set a periodic alert	Pass		-	
	Able to set a conditional alert	Pass		-	

Test Case ID	UATC09	**Test Form ID**	F33	**Participant ID**	3
Start Time	3:24 p.m.	**End Time**		3:24 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currencies Comparison	Able to view the responding currency comparison chart and table	Pass		-	
	Able to interact with the currency comparison chart	Pass		-	
	Able to quick select all the favourite or popular currencies for comparison	Pass		-	
	Able to change the selection of base currency and duration	Pass		-	

	Able to add or remove currency from comparing currencies	Pass	-

Test Case ID	UATC10	**Test Form ID**	F34	**Participant ID**	3
Start Time	3:24 p.m.	**End Time**		3:25 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currencies Correlation Analysis	Able to view the responding currency correlation chart and table	Pass		-	
	Able to interact with the currency correlation chart	Pass		-	
	Able to quick select all the favourite or popular currencies for analysis	Pass		-	
	Able to add or remove currency for correlation analysis	Pass		-	
	Able to change the input of duration	Pass		-	

Test Case ID	UATC11	**Test Form ID**	F35	**Participant ID**	3
Start Time	3:25 p.m.	**End Time**		3:26 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Alert	Able to toggle the alert's status	Pass		-	
	Able to delete the alert	Pass		-	
	Able to modify the alert	Pass		-	
	Able to modify the notes to be sent with notification	Pass		-	

Test Case ID	UATC12	**Test Form ID**	F36	**Participant ID**	3
Start Time	3:29 p.m.	**End Time**		3:29 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Notification	Able to receive email notification	Pass		-	
	Able to view in-app notifications	Pass		-	

Test Case ID	UATC01	**Test Form ID**	F37	**Participant ID**	4
Start Time	11:07 a.m.	**End Time**		11:11 a.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Authentication	Able to sign up new account	Pass		-	
	Able to verify account	Pass		-	
	Able to login the verified account	Pass		-	

Test Case ID	UATC02	**Test Form ID**	F38	**Participant ID**	4
Start Time	11:11 a.m.	**End Time**		11:12 a.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Google Authentication	Able to sign in using Google account	Pass		-	
	Able to add profile after signing in with Google	Pass		-	

Test Case ID	UATC03	**Test Form ID**	F39	**Participant ID**	4
Start Time	11:12 a.m	**End Time**		11:15 a.m	
Test Modules	**Test Description**	**Status**		**Comments**	
Reset Password	Able to reset new password	Pass		-	
	Able to login using new password	Pass		-	

Test Case ID	UATC04	**Test Form ID**	F40	**Participant ID**	4
Start Time	11:15 a.m.	**End Time**		11:16 a.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
User Profile	Able to change personal information	Pass		-	
	Able to change preference currencies	Pass		-	
	Able to change password	Pass		-	

Test Case ID	UATC05	**Test Form ID**	F41	**Participant ID**	4
Start Time	11:16 a.m.	**End Time**		11:17 a.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Dashboard	Able to view the responding exchange rate data and chart	Pass		-	
	Able to view the responding RSI data and chart	Pass		-	
	Able to view the popular currencies comparison chart and table	Pass		-	

	Able to view the favourite currencies comparison chart and table	Pass	-
	Able to change the selection of input currencies	Pass	-

Test Case ID	UATC06	**Test Form ID**	F42	**Participant ID**	4
Start Time	11:17 a.m.	**End Time**		11:18 a.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency Converter	Able to get the correct conversion result	Pass		-	
	Able to change the selection of input currencies and amount	Pass		-	

Test Case ID	UATC07	**Test Form ID**	F43	**Participant ID**	4
Start Time	11:18 a.m.	**End Time**		11:20 a.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency Exchange Rate Tracking	Able to view the responding exchange rate chart	Pass		-	
	Able to interact with the exchange rate chart	Pass		-	
	Able to change the selection of input currencies or duration	Pass		-	
	Able to set a periodic alert	Pass		-	
	Able to set a conditional alert	Pass		-	

Test Case ID	UATC08	**Test Form ID**	F44	**Participant ID**	4
Start Time	11:20 a.m.	**End Time**		11:22 a.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency RSI Values Tracking	Able to view the responding RSI values chart	Pass		-	
	Able to interact with the RSI values chart	Pass		-	
	Able to change the selection of input currencies or duration	Pass		-	
	Able to set a periodic alert	Pass		-	
	Able to set a conditional alert	Pass		-	

Test Case ID	UATC09	**Test Form ID**	F45	**Participant ID**	4
Start Time	11:22 a.m.	**End Time**		11:24 a.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currencies Comparison	Able to view the responding currency comparison chart and table	Pass		-	
	Able to interact with the currency comparison chart	Pass		-	
	Able to quick select all the favourite or popular currencies for comparison	Pass		-	
	Able to change the selection of base currency and duration	Pass		-	

	Able to add or remove currency from comparing currencies	Pass	-

Test Case ID	UATC10	**Test Form ID**	F46	**Participant ID**	4
Start Time	11:24 a.m.	**End Time**		11:27 a.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currencies Correlation Analysis	Able to view the responding currency correlation chart and table	Pass		-	
	Able to interact with the currency correlation chart	Pass		-	
	Able to quick select all the favourite or popular currencies for analysis	Pass		-	
	Able to add or remove currency for correlation analysis	Pass		-	
	Able to change the input of duration	Pass		-	

Test Case ID	UATC11	**Test Form ID**	F47	**Participant ID**	4
Start Time	1:24 p.m.	**End Time**		1:25 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Alert	Able to toggle the alert's status	Pass		-	
	Able to delete the alert	Pass		-	
	Able to modify the alert	Pass		-	
	Able to modify the notes to be sent with notification	Pass		-	

Test Case ID	UATC12	**Test Form ID**	F48	**Participant ID**	4
Start Time	11:31 a.m.	**End Time**		11:31 a.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Notification	Able to receive email notification	Pass		-	
	Able to view in-app notifications	Pass		-	

Test Case ID	UATC01	**Test Form ID**	F49	**Participant ID**	5
Start Time	8:52 p.m.	**End Time**		8:57 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Authentication	Able to sign up new account	Pass		-	
	Able to verify account	Pass		-	
	Able to login the verified account	Pass		-	

Test Case ID	UATC02	**Test Form ID**	F50	**Participant ID**	5
Start Time	8:57 p.m.	**End Time**		8:58 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Google Authentication	Able to sign in using Google account	Pass		-	
	Able to add profile after signing in with Google	Pass		-	

Test Case ID	UATC03	**Test Form ID**	F51	**Participant ID**	5
Start Time	8:58 p.m.	**End Time**		9:02 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Reset Password	Able to reset new password	Pass		-	
	Able to login using new password	Pass		-	

Test Case ID	UATC04	**Test Form ID**	F52	**Participant ID**	5
Start Time	9:02 p.m.	**End Time**		9:04 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
User Profile	Able to change personal information	Pass		-	
	Able to change preference currencies	Pass		-	
	Able to change password	Pass		-	

Test Case ID	UATC05	**Test Form ID**	F53	**Participant ID**	5
Start Time	9:04 p.m.	**End Time**		9:06 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Dashboard	Able to view the responding exchange rate data and chart	Pass		-	
	Able to view the responding RSI data and chart	Pass		-	
	Able to view the popular currencies comparison chart and table	Pass		-	

	Able to view the favourite currencies comparison chart and table	Pass	-
	Able to change the selection of input currencies	Pass	-

Test Case ID	UATC06	**Test Form ID**	F54	**Participant ID**	5
Start Time	9:06 p.m.	**End Time**		9:07 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency Converter	Able to get the correct conversion result	Pass		-	
	Able to change the selection of input currencies and amount	Pass		-	

Test Case ID	UATC07	**Test Form ID**	F55	**Participant ID**	5
Start Time	9:08 p.m.	**End Time**		9:11 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency Exchange Rate Tracking	Able to view the responding exchange rate chart	Pass		-	
	Able to interact with the exchange rate chart	Pass		-	
	Able to change the selection of input currencies or duration	Pass		-	
	Able to set a periodic alert	Pass		-	
	Able to set a conditional alert	Pass		-	

Test Case ID	UATC08	**Test Form ID**	F56	**Participant ID**	5
Start Time	9:11 p.m.	**End Time**		9:13 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currency RSI Values Tracking	Able to view the responding RSI values chart	Pass		-	
	Able to interact with the RSI values chart	Pass		-	
	Able to change the selection of input currencies or duration	Pass		-	
	Able to set a periodic alert	Pass		-	
	Able to set a conditional alert	Pass		-	

Test Case ID	UATC09	**Test Form ID**	F57	**Participant ID**	5
Start Time	9:13 p.m.	**End Time**		9:15 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currencies Comparison	Able to view the responding currency comparison chart and table	Pass		-	
	Able to interact with the currency comparison chart	Pass		-	
	Able to quick select all the favourite or popular currencies for comparison	Pass		-	
	Able to change the selection of base currency and duration	Pass		-	

	Able to add or remove currency from comparing currencies	Pass	-

Test Case ID	UATC10	**Test Form ID**	F58	**Participant ID**	5
Start Time	9:16 p.m.	**End Time**		9:18 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Currencies Correlation Analysis	Able to view the responding currency correlation chart and table	Pass		-	
	Able to interact with the currency correlation chart	Pass		-	
	Able to quick select all the favourite or popular currencies for analysis	Pass		-	
	Able to add or remove currency for correlation analysis	Pass		-	
	Able to change the input of duration	Pass		-	

Test Case ID	UATC11	**Test Form ID**	F59	**Participant ID**	5
Start Time	9:18 p.m.	**End Time**		9:21 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Alert	Able to toggle the alert's status	Pass		-	
	Able to delete the alert	Pass		-	
	Able to modify the alert	Pass		-	
	Able to modify the notes to be sent with notification	Pass		-	

Test Case ID	UATC12	**Test Form ID**	F60	**Participant ID**	5
Start Time	9:23 p.m.	**End Time**		9:24 p.m.	
Test Modules	**Test Description**	**Status**		**Comments**	
Notification	Able to receive email notification	Pass		-	
	Able to view in-app notifications	Pass		-	

Appendix E: Usability Testing Questionnaire and Surveys

Usability Testing

Section A: Introductory Questions and Survey

Name: Yap Chia Hau Email: chiahau29@1utar.my

Gender: (Male) / Female Age: 22

Occupation: Students (Bachelor of Science Software Engineering in UTAR)

1. Do you have any experience in using currency exchange rate tracking system?

   (Yes) / No

2. What are the applications that you used (if yes for previous question)?

   Moomoo and OctaFx

3. What is the frequency of checking currency exchange rate?

   ⊘ Always (Daily) ○ Sometimes (Weekly) ○ Seldom (Monthly) ○ Never

4. What are the purposes of checking currency exchange rate?

   ☑ Forex Trading ☐ Economic Analysis ☐ Travel Planning

   ☐ Cross-border Payments ☑ Purchase Overseas Products

   Others:

   _____

Section B: Test Scenarios

Test Scenarios
Scenario 1: Sign Up & Login Account You wish to use the currency exchange rate tracking system, but you do not have any registered accounts in the system. What would you do to sign up for an account and log into the system?
Scenario 2: Edit Profile (Default & Second Currency) After you log in, you are redirected to the dashboard page. You notice that the dashboard is loaded with charts and exchange rate data based on default currencies that you set earlier. Now, you wish to set the base currency to MYR and the second currency to SGD so this pair of currencies will be used to load the chart every time you load the page. What would you do?
Scenario 3: Currency Converter You wonder how much 130 Singaporean dollars (SGD) converts to in Malaysian Ringgit (MYR). What would you do?
Scenario 4: Exchange Rate Changes You wish to view the historical exchange rate chart for Malaysian Ringgit (MYR) to Japanese Yen (JPY) for the past six months. What would you do?
Scenario 5: Set Periodic Alert: Exchange Rate You wish to be notified via email to get the daily update on the currency exchange rate for Malaysian Ringgit (MYR) to Japanese Yen (JPY). What would you do to set the alert?
Scenario 6: RSI Values Changes You wish to view the historical relative strength index (RSI) chart for the Malaysian Ringgit (MYR) to the United States dollar (USD) to

check the performance of this pair of currencies over the last two years. What would you do?
Scenario 7: Set Conditional Alert: RSI You want to buy in the forex of Malaysian Ringgit (MYR) / United States dollar (USD). You wish to be notified in the application when this pair of currencies is oversold. What would you do to set up the alert?
Scenario 8: Currency Comparison You are wondering how well the Malaysian Ringgit (MYR) has performed over the last three months. You wish to use the currency comparison feature of the system to compare Malaysian Ringgit (MYR) with currencies like Singaporean Dollar (SGD), United State Dollar (USD), Chinese Yuan (CNY), Great British Bound (GBP) and Thai Baht (THB). What would you do?
Scenario 9: Set Favourite Currencies From the previous test, you noticed that there is an option to do a comparison by your favourite currencies. Thus, you wish to set the currencies used for the comparison above as your favourite currencies so you can perform a quick selection next time. What would you do to set your favourite currencies?
Scenario 10: Currency Correlation Analysis Now you have your favourite currencies set, you are curious about the correlation among these currencies like which currency is performing well also when a specific currency is outperforming. Thus, you wish to use the currency correlation analysis feature of the system to obtain insights into the correlation among your favourite currencies over the last three months. What would you do?

Scenario 11: Toggle Alert For temporarily, you wish to disable the periodic alert for the exchange rate that you set previously. What would you do?
Scenario 12: Edit Alert You noticed that the RSI value of the Malaysian Ringgit (MYR) to the United States dollar (USD) is close to the overbought line rather than the oversold line. You wish to be notified when the RSI value of this pair of currencies hits the overbought line so you can sell out your forex at the right time. set? At the same time, you also wish to send your notes together with the notification to remind yourself of performing the forex trading activity. What would you do to modify the condition and notes of the alert that you have previously set?

Section C: User Satisfaction Survey

Questions	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
I think the system serves its responsibility as an exchange rate tracking system				/	
I think the system does not cover the basic requirements for exchange rate tracking		/			
I think the system is user-friendly				/	
I think the system is complex and confusing	/				

I think the charts in the system are helpful for gaining insights					✓
I think the data visualization method used in the system is terrible	✓				
I think the system is flexible in managing alerts				✓	
I think the system is inflexible in creating and managing alerts.		✓			
I am satisfied with the additional features added (Currency correlation analysis and RSI chart visualization etc.)					✓
I think the additional features are useless	✓				

Comments on the Overall System

- No intraday exchange rate available is weakness of the system
- The dashboard is well-organized
- Heatmap takes too long to load

Usability Testing

Section A: Introductory Questions and Survey

Name: Ngeh Kai Bin Email: kaibin0826@1utar.my

Gender: Male / Female Age: 22

Occupation: Students (Bachelor of Science Software Engineering in UTAR)

1. Do you have any experience in using currency exchange rate tracking system?

Yes / No

5. What are the applications that you used (if yes for previous question)?

_____

6. What is the frequency of checking currency exchange rate?

◯ Always (Daily) ◯ Sometimes (Weekly) ◯ Seldom (Monthly) ☑ Never

7. What are the purposes of checking currency exchange rate?

☐ Forex Trading ☐ Economic Analysis ☐ Travel Planning

☐ Cross-border Payments ☐ Purchase Overseas Products

Others:

_____

Section B: Test Scenarios

Test Scenarios
Scenario 1: Sign Up & Login Account You wish to use the currency exchange rate tracking system, but you do not have any registered accounts in the system. What would you do to sign up for an account and log into the system?
Scenario 2: Edit Profile (Default & Second Currency) After you log in, you are redirected to the dashboard page. You notice that the dashboard is loaded with charts and exchange rate data based on default currencies that you set earlier. Now, you wish to set the base currency to MYR and the second currency to SGD so this pair of currencies will be used to load the chart every time you load the page. What would you do?
Scenario 3: Currency Converter You wonder how much 130 Singaporean dollars (SGD) converts to in Malaysian Ringgit (MYR). What would you do?
Scenario 4: Exchange Rate Changes You wish to view the historical exchange rate chart for Malaysian Ringgit (MYR) to Japanese Yen (JPY) for the past six months. What would you do?
Scenario 5: Set Periodic Alert: Exchange Rate You wish to be notified via email to get the daily update on the currency exchange rate for Malaysian Ringgit (MYR) to Japanese Yen (JPY). What would you do to set the alert?
Scenario 6: RSI Values Changes You wish to view the historical relative strength index (RSI) chart for the Malaysian Ringgit (MYR) to the United States dollar (USD) to

check the performance of this pair of currencies over the last two years. What would you do?
Scenario 7: Set Conditional Alert: RSI You want to buy in the forex of Malaysian Ringgit (MYR) / United States dollar (USD). You wish to be notified in the application when this pair of currencies is oversold. What would you do to set up the alert?
Scenario 8: Currency Comparison You are wondering how well the Malaysian Ringgit (MYR) has performed over the last three months. You wish to use the currency comparison feature of the system to compare Malaysian Ringgit (MYR) with currencies like Singaporean Dollar (SGD), United State Dollar (USD), Chinese Yuan (CNY), Great British Bound (GBP) and Thai Baht (THB). What would you do?
Scenario 9: Set Favourite Currencies From the previous test, you noticed that there is an option to do a comparison by your favourite currencies. Thus, you wish to set the currencies used for the comparison above as your favourite currencies so you can perform a quick selection next time. What would you do to set your favourite currencies?
Scenario 10: Currency Correlation Analysis Now you have your favourite currencies set, you are curious about the correlation among these currencies like which currency is performing well also when a specific currency is outperforming. Thus, you wish to use the currency correlation analysis feature of the system to obtain insights into the correlation among your favourite currencies over the last three months. What would you do?

Scenario 11: Toggle Alert For temporarily, you wish to disable the periodic alert for the exchange rate that you set previously. What would you do?
Scenario 12: Edit Alert You noticed that the RSI value of the Malaysian Ringgit (MYR) to the United States dollar (USD) is close to the overbought line rather than the oversold line. You wish to be notified when the RSI value of this pair of currencies hits the overbought line so you can sell out your forex at the right time. set? At the same time, you also wish to send your notes together with the notification to remind yourself of performing the forex trading activity. What would you do to modify the condition and notes of the alert that you have previously set?

Section C: User Satisfaction Survey

Questions	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
I think the system serves its responsibility as an exchange rate tracking system					✓
I think the system does not cover the basic requirements for exchange rate tracking	✓				
I think the system is user-friendly					✓
I think the system is complex and confusing		✓			

I think the charts in the system are helpful for gaining insights				✓	
I think the data visualization method used in the system is terrible	✓				
I think the system is flexible in managing alerts					✓
I think the system is inflexible in creating and managing alerts.	✓				
I am satisfied with the additional features added (Currency correlation analysis and RSI chart visualization etc.)					✓
I think the additional features are useless			✓		

Comments on the Overall System

- Charts are attractive and interactable
- Efficient in searching for currency

Usability Testing

Section A: Introductory Questions and Survey

Name: Tham Kar Weng Email: weng01260126@1utar.my

Gender: (Male) / Female Age: 22

Occupation: Students (Bachelor of Science Software Engineering in UTAR)

1. Do you have any experience in using currency exchange rate tracking system?

(Yes) / No

8. What are the applications that you used (if yes for previous question)?

XE

9. What is the frequency of checking currency exchange rate?

◯ Always (Daily) ◯ Sometimes (Weekly) ◯ Seldom (Monthly) ◯ Never

10. What are the purposes of checking currency exchange rate?

☐ Forex Trading ☐ Economic Analysis ☑ Travel Planning

☑ Cross-border Payments ☐ Purchase Overseas Products

Others:

Section B: Test Scenarios

Test Scenarios
Scenario 1: Sign Up & Login Account You wish to use the currency exchange rate tracking system, but you do not have any registered accounts in the system. What would you do to sign up for an account and log into the system?
Scenario 2: Edit Profile (Default & Second Currency) After you log in, you are redirected to the dashboard page. You notice that the dashboard is loaded with charts and exchange rate data based on default currencies that you set earlier. Now, you wish to set the base currency to MYR and the second currency to SGD so this pair of currencies will be used to load the chart every time you load the page. What would you do?
Scenario 3: Currency Converter You wonder how much 130 Singaporean dollars (SGD) converts to in Malaysian Ringgit (MYR). What would you do?
Scenario 4: Exchange Rate Changes You wish to view the historical exchange rate chart for Malaysian Ringgit (MYR) to Japanese Yen (JPY) for the past six months. What would you do?
Scenario 5: Set Periodic Alert: Exchange Rate You wish to be notified via email to get the daily update on the currency exchange rate for Malaysian Ringgit (MYR) to Japanese Yen (JPY). What would you do to set the alert?
Scenario 6: RSI Values Changes You wish to view the historical relative strength index (RSI) chart for the Malaysian Ringgit (MYR) to the United States dollar (USD) to

check the performance of this pair of currencies over the last two years. What would you do?
Scenario 7: Set Conditional Alert: RSI You want to buy in the forex of Malaysian Ringgit (MYR) / United States dollar (USD). You wish to be notified in the application when this pair of currencies is oversold. What would you do to set up the alert?
Scenario 8: Currency Comparison You are wondering how well the Malaysian Ringgit (MYR) has performed over the last three months. You wish to use the currency comparison feature of the system to compare Malaysian Ringgit (MYR) with currencies like Singaporean Dollar (SGD), United State Dollar (USD), Chinese Yuan (CNY), Great British Bound (GBP) and Thai Baht (THB). What would you do?
Scenario 9: Set Favourite Currencies From the previous test, you noticed that there is an option to do a comparison by your favourite currencies. Thus, you wish to set the currencies used for the comparison above as your favourite currencies so you can perform a quick selection next time. What would you do to set your favourite currencies?
Scenario 10: Currency Correlation Analysis Now you have your favourite currencies set, you are curious about the correlation among these currencies like which currency is performing well also when a specific currency is outperforming. Thus, you wish to use the currency correlation analysis feature of the system to obtain insights into the correlation among your favourite currencies over the last three months. What would you do?

Scenario 11: Toggle Alert For temporarily, you wish to disable the periodic alert for the exchange rate that you set previously. What would you do?
Scenario 12: Edit Alert You noticed that the RSI value of the Malaysian Ringgit (MYR) to the United States dollar (USD) is close to the overbought line rather than the oversold line. You wish to be notified when the RSI value of this pair of currencies hits the overbought line so you can sell out your forex at the right time. set? At the same time, you also wish to send your notes together with the notification to remind yourself of performing the forex trading activity. What would you do to modify the condition and notes of the alert that you have previously set?

Section C: User Satisfaction Survey

Questions	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
I think the system serves its responsibility as an exchange rate tracking system					
I think the system does not cover the basic requirements for exchange rate tracking					
I think the system is user-friendly					
I think the system is complex and confusing					

I think the charts in the system are helpful for gaining insights					✓
I think the data visualization method used in the system is terrible	✓				
I think the system is flexible in managing alerts					✓
I think the system is inflexible in creating and managing alerts.	✓				
I am satisfied with the additional features added (Currency correlation analysis and RSI chart visualization etc.)				✓	
I think the additional features are useless	✓				

Comments on the Overall System

- Charts looks very nice
- Overall user interface very neat

Usability Testing

Section A: Introductory Questions and Survey

Name: Yeoh Wei Bin Email: weibin02@1utar.my

Gender: (Male) / Female Age: 22

Occupation: Students (Bachelor of Science Software Engineering in UTAR)

1. Do you have any experience in using currency exchange rate tracking system?

(Yes) / No

11. What are the applications that you used (if yes for previous question)?

Octa FX

12. What is the frequency of checking currency exchange rate?

⊘ Always (Daily) ○ Sometimes (Weekly) ○ Seldom (Monthly) ○ Never

13. What are the purposes of checking currency exchange rate?

☑ Forex Trading ☐ Economic Analysis ☐ Travel Planning

☐ Cross-border Payments ☐ Purchase Overseas Products

Others:

Section B: Test Scenarios

Test Scenarios
Scenario 1: Sign Up & Login Account You wish to use the currency exchange rate tracking system, but you do not have any registered accounts in the system. What would you do to sign up for an account and log into the system?
Scenario 2: Edit Profile (Default & Second Currency) After you log in, you are redirected to the dashboard page. You notice that the dashboard is loaded with charts and exchange rate data based on default currencies that you set earlier. Now, you wish to set the base currency to MYR and the second currency to SGD so this pair of currencies will be used to load the chart every time you load the page. What would you do?
Scenario 3: Currency Converter You wonder how much 130 Singaporean dollars (SGD) converts to in Malaysian Ringgit (MYR). What would you do?
Scenario 4: Exchange Rate Changes You wish to view the historical exchange rate chart for Malaysian Ringgit (MYR) to Japanese Yen (JPY) for the past six months. What would you do?
Scenario 5: Set Periodic Alert: Exchange Rate You wish to be notified via email to get the daily update on the currency exchange rate for Malaysian Ringgit (MYR) to Japanese Yen (JPY). What would you do to set the alert?
Scenario 6: RSI Values Changes You wish to view the historical relative strength index (RSI) chart for the Malaysian Ringgit (MYR) to the United States dollar (USD) to

check the performance of this pair of currencies over the last two years. What would you do?
Scenario 7: Set Conditional Alert: RSI You want to buy in the forex of Malaysian Ringgit (MYR) / United States dollar (USD). You wish to be notified in the application when this pair of currencies is oversold. What would you do to set up the alert?
Scenario 8: Currency Comparison You are wondering how well the Malaysian Ringgit (MYR) has performed over the last three months. You wish to use the currency comparison feature of the system to compare Malaysian Ringgit (MYR) with currencies like Singaporean Dollar (SGD), United State Dollar (USD), Chinese Yuan (CNY), Great British Bound (GBP) and Thai Baht (THB). What would you do?
Scenario 9: Set Favourite Currencies From the previous test, you noticed that there is an option to do a comparison by your favourite currencies. Thus, you wish to set the currencies used for the comparison above as your favourite currencies so you can perform a quick selection next time. What would you do to set your favourite currencies?
Scenario 10: Currency Correlation Analysis Now you have your favourite currencies set, you are curious about the correlation among these currencies like which currency is performing well also when a specific currency is outperforming. Thus, you wish to use the currency correlation analysis feature of the system to obtain insights into the correlation among your favourite currencies over the last three months. What would you do?

Scenario 11: Toggle Alert For temporarily, you wish to disable the periodic alert for the exchange rate that you set previously. What would you do?
Scenario 12: Edit Alert You noticed that the RSI value of the Malaysian Ringgit (MYR) to the United States dollar (USD) is close to the overbought line rather than the oversold line. You wish to be notified when the RSI value of this pair of currencies hits the overbought line so you can sell out your forex at the right time. set? At the same time, you also wish to send your notes together with the notification to remind yourself of performing the forex trading activity. What would you do to modify the condition and notes of the alert that you have previously set?

Section C: User Satisfaction Survey

Questions	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
I think the system serves its responsibility as an exchange rate tracking system					
I think the system does not cover the basic requirements for exchange rate tracking					
I think the system is user-friendly					
I think the system is complex and confusing					

I think the charts in the system are helpful for gaining insights					
I think the data visualization method used in the system is terrible					
I think the system is flexible in managing alerts					
I think the system is inflexible in creating and managing alerts.					
I am satisfied with the additional features added (Currency correlation analysis and RSI chart visualization etc.)					
I think the additional features are useless					

Comments on the Overall System

- Not enough forex indicator, not appealing to short-term forex traders due to intraday exchange rates unavailable

Usability Testing

Section A: Introductory Questions and Survey

Name: Mok Lai Hoong Email: moklaihoong@gmail.com

Gender: Male / (Female) Age: 46

Occupation: Primary School Teacher

1. Do you have any experience in using currency exchange rate tracking system?

Yes / (No)

14. What are the applications that you used (if yes for previous question)?

None

15. What is the frequency of checking currency exchange rate?

◯ Always (Daily) ◯ Sometimes (Weekly) ⊘ Seldom (Monthly) ◯ Never

16. What are the purposes of checking currency exchange rate?

☐ Forex Trading ☐ Economic Analysis ☑ Travel Planning

☐ Cross-border Payments ☑ Purchase Overseas Products

Others:

Section B: Test Scenarios

Test Scenarios
Scenario 1: Sign Up & Login Account You wish to use the currency exchange rate tracking system, but you do not have any registered accounts in the system. What would you do to sign up for an account and log into the system?
Scenario 2: Edit Profile (Default & Second Currency) After you log in, you are redirected to the dashboard page. You notice that the dashboard is loaded with charts and exchange rate data based on default currencies that you set earlier. Now, you wish to set the base currency to MYR and the second currency to SGD so this pair of currencies will be used to load the chart every time you load the page. What would you do?
Scenario 3: Currency Converter You wonder how much 130 Singaporean dollars (SGD) converts to in Malaysian Ringgit (MYR). What would you do?
Scenario 4: Exchange Rate Changes You wish to view the historical exchange rate chart for Malaysian Ringgit (MYR) to Japanese Yen (JPY) for the past six months. What would you do?
Scenario 5: Set Periodic Alert: Exchange Rate You wish to be notified via email to get the daily update on the currency exchange rate for Malaysian Ringgit (MYR) to Japanese Yen (JPY). What would you do to set the alert?
Scenario 6: RSI Values Changes You wish to view the historical relative strength index (RSI) chart for the Malaysian Ringgit (MYR) to the United States dollar (USD) to

check the performance of this pair of currencies over the last two years. What would you do?
Scenario 7: Set Conditional Alert: RSI You want to buy in the forex of Malaysian Ringgit (MYR) / United States dollar (USD). You wish to be notified in the application when this pair of currencies is oversold. What would you do to set up the alert?
Scenario 8: Currency Comparison You are wondering how well the Malaysian Ringgit (MYR) has performed over the last three months. You wish to use the currency comparison feature of the system to compare Malaysian Ringgit (MYR) with currencies like Singaporean Dollar (SGD), United State Dollar (USD), Chinese Yuan (CNY), Great British Bound (GBP) and Thai Baht (THB). What would you do?
Scenario 9: Set Favourite Currencies From the previous test, you noticed that there is an option to do a comparison by your favourite currencies. Thus, you wish to set the currencies used for the comparison above as your favourite currencies so you can perform a quick selection next time. What would you do to set your favourite currencies?
Scenario 10: Currency Correlation Analysis Now you have your favourite currencies set, you are curious about the correlation among these currencies like which currency is performing well also when a specific currency is outperforming. Thus, you wish to use the currency correlation analysis feature of the system to obtain insights into the correlation among your favourite currencies over the last three months. What would you do?

Scenario 11: Toggle Alert For temporarily, you wish to disable the periodic alert for the exchange rate that you set previously. What would you do?
Scenario 12: Edit Alert You noticed that the RSI value of the Malaysian Ringgit (MYR) to the United States dollar (USD) is close to the overbought line rather than the oversold line. You wish to be notified when the RSI value of this pair of currencies hits the overbought line so you can sell out your forex at the right time. set? At the same time, you also wish to send your notes together with the notification to remind yourself of performing the forex trading activity. What would you do to modify the condition and notes of the alert that you have previously set?

Section C: User Satisfaction Survey

Questions	Strongly Disagree 1	Disagree 2	Neutral 3	Agree 4	Strongly Agree 5
I think the system serves its responsibility as an exchange rate tracking system					
I think the system does not cover the basic requirements for exchange rate tracking					
I think the system is user-friendly					
I think the system is complex and confusing					

I think the charts in the system are helpful for gaining insights					
I think the data visualization method used in the system is terrible					
I think the system is flexible in managing alerts					
I think the system is inflexible in creating and managing alerts.					
I am satisfied with the additional features added (Currency correlation analysis and RSI chart visualization etc.)					
I think the additional features are useless					

Comments on the Overall System
