DESIGN AND IMPLEMENTATION OF PERSONAL LOAN PROCESSING SYSTEM USING AI TECHNIQUE

By

Jason Lee Chia Shen

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology
(Kampar Campus)

JAN 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ___Design and Implementation of Personal Loan Processing____
____System Using AI Technique _____________________________

Academic Session: __Jan 2024____

I _____JASON LEE CHIA SHEN__________________________
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

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

Verified by,

_____Jason___

(Author's signature)

(Supervisor's signature)

Address:
__12, Lorong Bertam Flora 3,
Taman Bertam Flora, 13200,
Kepala Batas, Pulau Pinang___

Phan Koo Yuen
Supervisor's name

Date: ___25/4/2024___________

Date: 26/4/2024

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: ___25/4/2024____________

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ______***Jason Lee Chia Shen***__________________________ (ID No: __***20ACB03695***__) has completed this final year project entitled "____*Design and Implementation of Personal Loan Processing System using AI Technique*_____" under the supervision of _____________Ts Dr Phan Koo Yuen___________ (Supervisor) from the Department of __Computer Science___________, Faculty of _Information and Communication Technology________ , and ____________________ (Co-Supervisor)* from the Department of _________________________, Faculty/Institute* of __________________________.

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

Yours truly,

Jason

(*Jason Lee Chia Shen*)

DECLARATION OF ORIGINALITY

I declare that this report entitled “**DESIGN AND IMPLEMENTATION OF PERSONAL LOAN PROCESSING SYSTEM USING AI TECHNIQUE**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ______________________

Name : JASON LEE CHIA SHEN

Date : 25/4/2024

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Ts. Dr Phan Koo Yuen, who has given me this bright opportunity to engage in a machine learning project. It is my first step to establish a career in the machine learning and artificial intelligence field. A million thanks to you.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

This project focuses on the design and implementation of personal loan processing system using artificial intelligence technique. With the issue of manual and tedious tasks in traditional personal loan processing operations, many banking and financial institutions have not seen substantial increase in their productivity. As for that, this project has proposed a personal loan system to automate the manual loan processes. The borrower can get their creditworthiness and eligibility for a loan checked in an accurate manner and in a very short time. This project integrates a gradient boosting model (XGBoost) and SHAP (Shapley Additive exPlanations) values to accurately predict the creditworthiness of the borrower in the context of loan approval. Furthermore, to tackle the issue of inadequate credit history, the digital footprints of the borrower will also be used to compute their eligibility for a loan. The proposed personal loan system is light weight to be deployed in real world conditions where time and performance matter. This system will comply to all the ethical standards and regulations to give the borrower a peace of mind when using the system. The system will be transparent enough to be used in real banking and financial organizations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 Project Background

1.1 Introduction

In the financial industry, when an individual applies for a personal loan, this loan application will usually go through the initial pre-qualification assessment, to the underwriting process, and to the funding of the loan which is widely known as the loan origination process. But with the implementation of personal loan system using the Artificial Intelligence (AI) technique, these manual, labour-intensive, and time-consuming processes are vastly automated. Not only boosting the productivity of the financial industry, but the AI-based personal loan system will also lessen the overall time it takes for an individual to get funded from the bank.

This chapter is structured as follows. In chapter 1.2 the overview of the issues with the current personal loan, challenges facing by banks, and how AI can help in these situations are presented. The problem statement and solutions are discussed in chapter 1.3. In chapter 1.4, and chapter 1.5, the project scope and objectives of this project are presented respectively. Motivation and contribution are discussed in chapter 1.6 and chapter 1.7.

1.2 Overview of Personal Loan and Artificial Intelligence

The following definition of personal loan is based on [1]. The authors have defined that personal loan is the fund that a borrower obtains from a lender which can be banks or other financial institutions for the sake of personal use. Personal loan is one of the many services offered by lenders. This type of loan is usually aimed to assist borrowers that are having short-term difficulties in personal finances.

Although the world has seen the widespread availability of personal loans, but, there is still one major issue, *passing the pre-qualification assessment*. A recent survey done by [2] has shown that the application rate for any type of loans has slumped from 40.9 percent to 40.3 percent. It was the lowest reading recorded since October 2020. Not only that, from the same survey, the rejection rate for loan applications has increased significantly to 21.8 percent ever since the highest rate recorded in June 2018. The main contributor for personal loan applications to be rejected usually stems from the applicant for not having a good credit score [3]. In Malaysia, the applicant's history such as existing debts, loans, and other borrowings can be accessed through the Central Credit Reference Information System (CCRIS) under the Credit Bureau of Bank Negara Malaysia. Banks will usually refer to this report to determine the creditworthiness of the application which eventually will lead to either the approval or the rejection of the loan application.

Moreover, the traditional lending practices often lead to the inefficiency of the loan origination process in banks. As stated in [4], most of the paper-based loan origination process involves manual data entry. Having to enter huge volumes of data by hand exposes the banks and financial institutions to huge risks of human errors that could potentially delay the overall lending process. Not only that, as banks are still reliant on paper-based documents and manual tasks, these situations have forced loan officers to continue handle, stare, and compare applicants' data manually which are both redundant and inefficient [5]. While some banks have automated some processes, these inefficiencies still persist. According to a Forbes Global Insight survey, out of 60% of the senior executives interviewed, only 38% claimed that they had already achieved the automation of the loan origination process [6]. Other than the aforementioned issues, credit risk is regarded as the biggest headache for the banks. Assessing creditworthiness and the risk of defaulting as accurately as possible are the well-known challenges in the industry. Insufficient amount of quality data often ends up having the loan application wrongfully rejected by the traditional lenders [7].

From a recent survey by Credit Tip-Off Service (CTOS) State of Consumer Credit in Malaysia 2022, it had been reported that consumers were borrowing more than ever [8]. Three out of ten people had a minimum of one personal loan compared to previous years. Furthermore, the same report observed the outstanding personal loan balance had increased 5.4% every year, reaching an all-time high of RM67,470 per person. As in [8], both income groups such as Bottom 40% (B40) and Middle 40% (M40) had their average outstanding balance climbed 3.1% to RM35,531 per person and 3.7% to RM93,202 per person respectively. This report clearly outlines the pre- and post-Covid 19 pandemic situation in Malaysia and the consumers' urgent needs for financial aid. Not only is the increasing trend of personal loans prevalent in Malaysia, but on the international stage as well. An analysis conducted by [9] shows the global personal loan market was worth $47.79 billion in 2020. In the following year 2021, the market is projected to be valued at a staggering estimation of $719.31 billion by 2030, growing at a compound annual growth rate (CAGR) of 31.7%.

Due to the existing issues with personal loans and the increasing trend of personal loans, AI is widely used in the financial industry in recent years. AI goes hand in hand with big data, which is why the adoption of this technology is generally welcomed as the banking industry is heavily dependent on data and information [10]. At the surface level, AI can be applied to serve customers. When it comes to the technical aspects, AI could support various kind of operations which include the loan approval, the analysis of decisions and the diversification of financial transactions [11]. The rapid development of Internet of Things (IOT) has shifted people's attention to AI, initiating a widespread usage of AI in numerous domains. This technological trend records over 30% of recent AI patent applications, reflecting the banking industry's growing interest in the application of AI [12]. There are several AI techniques for loan processing system available out in the market. Some of the most common ones are machine learning for credit scoring [13], Natural Language Processing (NLP) [14], robotic process automation (RPA) [15], and fraud detection [16]. In general, many of the big banks use AI to accelerate the underwriting process which can increase the profit for every loan that has been approved and can reduce the delay to some extent. Not only that, but some companies also automate the entire loan process using AI [17]. For instance, in Malaysia, CIMB Bank Berhad has implemented the machine learning credit models using the AI platform developed by DataRobot [18]. It enables the bank to have a better credit risk model and improves the retail underwriting speed across all loan products.
In short, the rapid advancement of AI has transformed the financial technology industry, revolutionizing how banks operate from the surface level, such as customer service, to the deep technical aspect of the whole loan processing operations.

1.3 Problem Statement

1.3.1 Fairness and Ethical Considerations

Concerns are starting to arise when AI has the power to approve or reject a loan application. Decisions made by AI can be accurate to some extent, but the question will be on how AI interprets fairness and whether the decisions made are biased in a more complex scenario. As machine learning models are trained using historical data, they are subject to bias as the sample provided often reflects the prejudices of the individual who originally made the decision. This presents a challenge for the AI community as it is immensely difficult to evaluate the fairness of a machine learning model.

Fairness is a nuanced concept; it does not have a one-size-fits-all definition. What is considered fair in one scenario may be deemed unfair in another. One of the rather famous cases of a biased model is the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) algorithm which has been highly criticised for its racial bias. Based on its prediction, the system had unfairly judged black offenders and falsely labeled them as high-risk individuals twice as likely than white defendants [19]. This shows that any machine learning models could just be as bias and unfair if the provided sample is already biased to begin with.

In many fields, AI is thought to be more superior and outperforms humans in many ways. But the application of AI is still regarded with suspicion and the human experience is not replaceable [20]. Due to these concerns, there is a debate on whether AI should adhere to a set of ethical principles. It is inadequate to dictate the performance of the machine learning model if considered only a black box. Loan officers has to understand the reasoning behind on every prediction made by the machine learning model due to the high-risk nature of loan approval processes [21].

To address the issues relating to fair lending practices, it is crucial to include diverse and representative data during the development and training of the machine learning model. This method can help to mitigate the issue of biased training data and contribute to more fair lending decisions.

1.3.2 High Rejection Rate

The worst thing that can happen is being turned down for a personal loan while in desperate need of financial assistance. There are many factors that could lead to a personal loan to be rejected. It is difficult to determine the creditworthiness of an individual due to lack of credit history [22]. Because of this, the high rejection rate presents a significant challenge particularly for those with limited credit history. Financial institutions cannot accurately assess and determine whether the loan should be granted to the applicant. As for that, it leads to the conservative lending decision where the applicant with limited credit history will be automatically rejected which contributes to the rejection rate. This is a big headache for those in need of financial assistance.

To mitigate this particular issue, the project aims to explore the use of digital footprints to enhance the accuracy of the creditworthiness assessment. A more inclusive and accurate evaluation process will be developed to help reduce the high rejection rate and provide opportunities for those with limited credit history.

1.3.3 Efficiency and Automation

Despite pandemic-driven digital pushes, some of the financial institutions do not offer online loan application [23]. Moreover, the current loan processing system still relies heavily on manual data entries, reviews, and credit assessments. From the same article, the finding has concluded that entering the same data repeatedly is the biggest hindrance in the commercial lending process. Some financial institutions have reportedly had to re-enter the same data for a loan application up to five or six times. The lack of automation and inefficiencies of traditional manual lending method hampers the loan approval process, resulting in increased operational costs, delays, and suboptimal user experiences.

To address the efficiency and automation issue, this project will implement machine learning method to reduce manual tasks and assess loan applications more accurately.

1.4 Project Scope

This project aims to develop a personal loan system using a machine learning method to replace the manual and tedious loan process. The proposed personal loan system will automate the entire loan processing operation in which when the borrower provides their relevant personal information, the proposed system will automatically process the data and determine the creditworthiness of the borrower using the machine learning method.

In addition, a simple web application will be developed together with the machine learning method. The borrower can browse the web application and check their eligibility for a loan. Although it is optional, but the borrower is advised to provide their digital footprints in order for their final credit score to be accurate and precise.

For the proposed system, all the private and confidential information will be handled in compliance with the ethical standards and with laws and regulations. In case the loan application is rejected, the proposed system will utilize the SHAP values to provide meaningful feedback to the borrower on why it is rejected.

Finally, since the proposed system will be deployed and used in real-time, the proposed system should be light weight enough so that the borrower will not have to wait long before they can know their creditworthiness.

In this project, a gradient boosting model (XGBoost) and the SHAP values will be used to determine the creditworthiness of the borrower. The project will be completed by May 2024.

1.5 Project Objectives

The objectives of this project are as follows:

1. To develop a personal loan system using a gradient boosting model (XGBoost) and with SHAP values to determine the riskiness, eligibility, and creditworthiness of the borrower as well as to provide explicit explanation on reasons of the final decisions made by the machine learning method.
   - Collect and prepare a dataset containing information about loan applicants.
   - Train the XGBoost model with the prepared dataset.
   - To ensure the final result can be interpreted and explained, SHAP values can be used. The values can be used to express the final decisions in human-readable format.

2. To develop a simple and interactive web application for the borrower to check their creditworthiness and eligibility for a personal loan by providing their relevant personal details and the overall process should be fast, efficient, and user-friendly.
   - Login Module. The user will be required to log in when using the personal loan system. The user has to provide an email address and a password for registering an account.
   - Data Collection Module. A form will be made available for the user to enter their personal details, including financial information and any other criteria.
   - Personal Loan Module. This dashboard will show relevant information to the user including their past loan analysis history, explanation on each result of the loan application, etc. The loan processing will run once the user has submitted their relevant information.

3. To integrate digital footprints data into the proposed personal loan system and the system will be developed with accordance to regulatory compliance and fairness considerations and the transparency of the model is crucial for ethical reasons.

- The user has the choice to provide their digital footprints data into the personal loan system manually. The digital footprints data will be used as part of the training dataset to train the XGBoost model.
- To prevent potential bias in the digital footprints data, the data will be analyzed and evaluated of its impact on the lending decisions to ensure fairness. If the dataset is deemed fair, then it will be used for training.

1.6 Motivation

The aim of the project is to propose an improved version of the existing personal loan system using AI technique. As traditional lending practices often impede efficiency, it leads to unwanted delays and increased operational costs for banking institutions. Due to that, this project will examine existing algorithms and develop a machine learning model that can streamline and automate the tedious loan processes. Bias and fairness in lending decisions is a critical concern. This project aims to create a personal loan system that makes decisions based on objective criteria rather than demographic factors. Not only that, but this project's motivation also extends to gaining a better understanding of the overall situation of the personal loan market in the lending industry. This proposed system can increase the overall efficiency of the loan process and reduce the risk of borrowers defaulting.

1.7 Impact, Significance and Contribution

1.7.1 Implications to the Practitioners

This paper presents several contributions. Firstly, the proposed approach can automate almost the entire loan origination process. It saves the hassle of the loan officer from having to perform those operations manually which are time-consuming and tedious. With that, the proposed approach can also speed up the loan origination process significantly with its intelligent decision-making skills. This model not only benefits the lender, but also it helps the borrower indirectly by computing their digital footprint data which could potentially boost their creditworthiness to some extent.

As most of the machine learning models in the context of loan applications are geared towards the lenders, it can lower the operational cost of the banking industry tremendously. The proposed approach can empower loan officers in helping them to serve their clients better. In the traditional loan origination process, most of the time is spent on finding and formatting the relevant data and information of the borrower. By incorporating the model into their work, the overall productivity will be boosted as AI is best at sourcing accurate and reliable data which it is prone to human errors if done manually by loan officers. Not only that, but AI is also capable of retrieving data much faster than a loan officer.

Initially, loan origination was an entirely manual operation. The dependence on loan officers would sometimes affect the outcome of the loan application. Without a doubt, human generally has bias in one way or another and often enough this would seep into the loan origination process, causing a thought to be eligible applications to be wrongly rejected or to have lower interest rates applied to the specific application due to favouritism and bias. By introducing the machine learning algorithm, bias will be eliminated given that the training dataset is not bias in the first place which can reduce the error rates and can profit lenders in the long run.

1.7.2 Knowledge of the body

This project aims to contribute to the field of AI in the banking industry. Other researchers could use this paper as the basis of their research. This proposed system could also assist other machine learning model to better predict the risk of defaulting and to assess the creditworthiness of the applicant.

1.8 Report Organization

The details of this research are shown in the following chapters. In Chapter 2, some related backgrounds are reviewed. Then, the system methodology and approach of this personal loan system are discussed in Chapter 3. And then, Chapter 4 describes the system design of the personal loan processing system. Besides, Chapter 5 shows the system implementation of this project. The system evaluation and discussion are presented in the Chapter 6. Lastly, Chapter 7 will be on the conclusion of this project and also recommendations for future work.

CHAPTER 2 Literature Review

2.1 Previous Works

2.1.1 Artificial Intelligence and Bank Credit Analysis: A Review

In [24], the author has analyzed the impact of AI on credit analysis procedures. In the analysis, the author has reviewed the work done by [25]. The work can be divided into three stages. The first stage is to treat the data in different ways. Among different steps, it is first checked for any missing or irregular data which is then grouped under discrete explanatory variables and discretization of continuous variables. In the second stage, the variables are analyzed to make sure it is not too correlated to each other. Any redundant variables are removed by using the principle of parsimony. The last stage involves selecting the explanatory variables for the credit score model. The author has commented that the use of AI algorithm makes the setup above obsolete as the use of tree-based algorithm is able to determine the modalities and optimal discretization automatically. Thus, this leads to high productivity and predictive performance gains. Not only that, but AI also makes the automation of credit granting process possible. To have a better creditworthiness assessment, a combination of AI and the use of big data has emerged. The big data is sourced from the digital fingerprint data of the customer which usually includes their social network activity information. Other than using the traditional variables such as age, income and marital status, the AI will also process the big data together during the credit analysis phase. However, the author has argued that there will be a problem of finding the balance between the protection of the customer's privacy data and the ethical aspects of the usage of AI and big data and the tension from regulators.

2.1.2 A Novel Hybrid Ensemble Model Based on Tree-Based Method and Deep Learning Method for Default Prediction

[26] has proposed a novel ensemble model to increase the predictive performance of the default prediction. The proposed model consists of a tree-based method, a deep learning method, classifier ensemble and classifier setting. In the tree-based method,

LightGBM is used to generate features for the classifiers. The author uses convolutional neural network (CNN) as their feature generation method for the deep learning classifier. In the study, RF, XGBoost, and LightGBM are used as the foundation to ensemble with the deep learning model so that the performance and robustness of the proposed model could be further improved. The experimental result of this proposed model shows that: 1) the feature generation based on tree (LightGBM) can improve the prediction performance of the classifiers; 2) features can be generated effectively with the use of deep learning feature generation method based on CNN which can also improve the default prediction; 3) the classifier ensemble improves the final overall results of the default prediction. However, the feature generation methods used in this proposed study have not been optimized for hyperparameters. Hyperparameters optimization can improve the predictive performance of the model to some extent but without optimization, the model might suffer from performance loss as features might not be able to capture the underlying pattern of the data effectively.

2.1.3 Improvement of Personal Loans Granting Methods in Banks Using Machine Learning Methods and Approaches in Palestine

Predicting credit defaulters is a hard challenge for many banking organizations. Loan approval and risk assessment require complex processing as they need high efforts and deep understanding of the matter, and it becomes a challenge for the employee to make an accurate decision due to the traditional and manual methods used in the banks and financial institutions. As for that, [27] has proposed to study three different machine learning algorithms: Decision Tree (DT), Logistic Regression (LR), and Random Forest (RT), by using real-life data sourced from Quds Bank. For estimation using logistic regression, the author has used the property of Bernoulli distribution and Maximum likelihood. The proposed decision tree consists of three steps: 1) choose attribute in the root; 2) divide training set into subsets, such that each subset has same value for an attribute; 3) step one and step two are iterated until leaf nodes are traversed. The random forest used by the author is a combination of tree classifiers which is generated using a random vector. As for accuracy, the decision tree implemented by the author is the highest among the other machine learning algorithms which is essential for banking institutions. The proposed system works effectively and complies with all the requirements and criteria of banks. However, the proposed system might not be efficient if the borrower has very little credit history. The proposed system might have a high rejection rate due to the inadequate data of the borrower as the system might correlate with the pattern of lack of credit history with low repayment behavior.

2.1.4 Credit Score Prediction System using Deep Learning and K-Means Algorithms

With the increasing growth of banking and financial industry, it becomes challenging to identify the insights on the performance of the financial industry. Hence, [28] has proposed their own credit scoring system by using K-Means algorithm and deep learning. Credit scoring data might contain redundant or unrelated data which sometimes could lead to the performance loss of the model. To overcome the issue, the author uses the feature selection technique that will only select the relevant and most important features from the training dataset. The model is then trained using the processed dataset and a deep neural network. This allows the model to learn for a low dimensional embedding of input vector and weight vector [28]. The proposed credit scoring algorithm will apply decision tree-based classification on the selected features from the deep learning network. The predictive model is then deployed in real-word scenarios by using actual credit scoring dataset. Figure 2.1.4.1 shows the overall architecture of the proposed model. The result of this proposed credit scoring system shows that it can predict the credit score for every customer with a great accuracy. The deep neural network of the proposed credit scoring system can assist the prediction model to better predict by utilizing the root mean square function and activation function which can reduce the error. However, it is hard to interpret how the deep learning network has arrived at their decisions where in the context of the personal loan system, transparency is crucial. This may be the drawback due to the lack of interpretability.

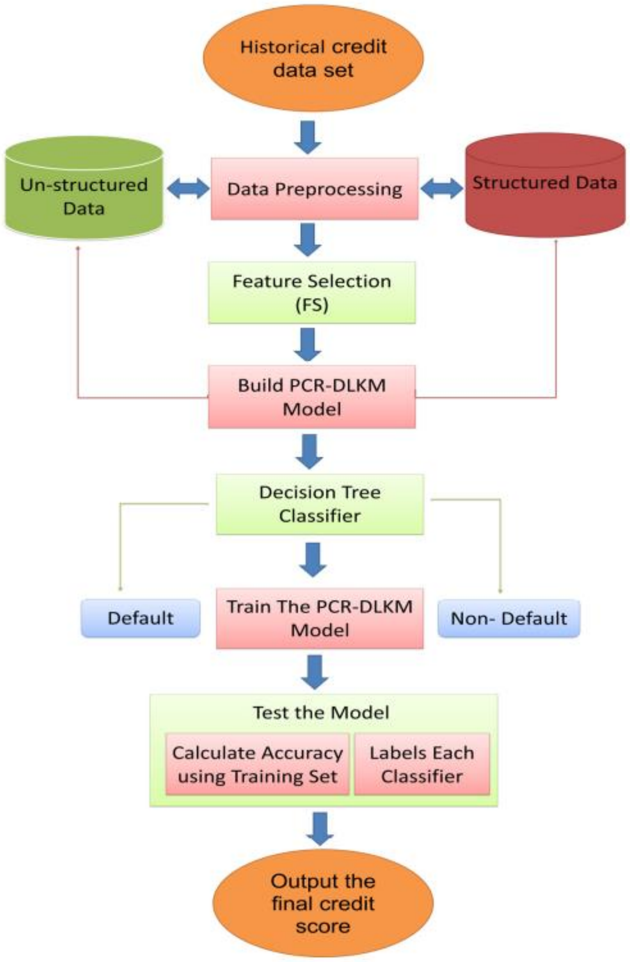


Figure 2.1.4.1: Block diagram of the proposed predictive model [28].

2.1.5 Methodology and Models for Individuals' Creditworthiness Management Using Digital Footprint Data and Machine Learning Methods

The key challenge of the current situation in banking and financial sectors is to reduce banks' credit risks associated with the risk of defaulting of borrowing individuals [29]. To mitigate the challenge, the author in [29] has proposed a new methodology and models to better assess the creditworthiness of an individual as well as incorporate additional digital footprint data into their approach. The methodology is divided into four stages. The first stage will yield three groups of indicators from the analysis and acquisition of the data: financial, anthropometric, and digital footprint data. The correlation analysis between the borrower's riskiness and the influence of the three groups of indicators obtained earlier is carried out in the second stage. In the third stage, a homogenous cluster will be formed on borrowers that share the same characteristics. Classification will be made, and riskiness are determined. The last stage involves the development of the management decisions. From the results obtained, it can be concluded that the proposed model outperforms the regression model (R-model). The proposed model also increases the efficiency of financial decisions. However, since the digital footprint data of the borrower is used in the proposed model, the accuracy maybe not always precisely represent a borrower's financial situation. For instance, one of the digital footprint data used, "music style" might not necessarily contribute to the creditworthiness of the borrower. It is very subjective, and the accuracy of the final output is heavily dependent on the back-end user of the proposed model.

2.1.6 The Use of Responsible Artificial Intelligence Techniques in the Context of Loan Approval Processes

With the skepticism about the use of artificial intelligence in loan approval processes, [21] uses *fairness* and *explainability* techniques that can lead to the increase in trust and reliance on an artificial intelligence system. The proposed proprietary framework applies the principles of *fairness* and *explainability.* The users will be able to get explanations on each prediction and results made by the proposed model. With that, the model will also check for fairness and mitigate any presence of biases in the loan approval process. The author also allows the loan officer to give feedback about a certain prediction in which will help to build a better and new machine learning model. During the machine learning model training, several models are built by using different machine learning algorithms. Confusion matrix is used to evaluate the performance of each of them and to select the best one that will be implemented in the final workflow. The proposed approach is proven to be effective through experimental results from several user case studies and field tests. Hence, it leads to the growth of trust and reliance of domain experts on artificial intelligence systems. However, the dataset size of the explainability algorithms evaluation is small (2440 samples) which the proposed approach might be prone to overfitting issues.

2.1.7 An Approach for Prediction of Loan using Machine Learning Algorithm

Loan prediction is a challenge that every banking and financial institution faces. They can only gain profit if the prediction is accurate, and the borrower is expected to not default [30]. [30] has proposed a model that can predict and classify loan by using Logistic Regression with sigmoid function. The author uses Pandas and Numpy library to generate the features required for the prediction. During the feature engineering, logarithm transformation is applied to handle the skewed data to decrease the effect of the outliers. Among different machine learning model, the author has adopted the logistic regression model as it meets the requirements and constraints of the stakeholder (borrowers and banks). The model is then evaluated with various methods such as Accuracy, Precision, Recall, F1 score and Confusion matrix. During the performance evaluation phase, it is found that the proposed model performs well with high accuracy of 81.1 percent and conclusion has been made that applicants with lower credit score will not be granted with a loan and vice versa. However, logistic regression assumes the relationship between the independent variables and the dependent variables. For instance, a person that earns a very high income might not necessarily correlate to high likelihood of loan approval.

2.1.8 Comparative Analysis of Bank Loan Defaulter Prediction Using Machine Learning Techniques

Due to the rapid expansion of banking and financial industry, there is an emerging need to discover the parameters used by banks and financial institutions to evaluate a candidate’s profile whether they are trustworthy enough to be granted with a loan [31]. In [31], the author has proposed a comparative study of Random Forest, Naïve Bayes, and the Support Vector Machine to predict the creditworthiness of the borrower. Figure 2.1.8.1 shows the bank loan defaulter prediction methodology.

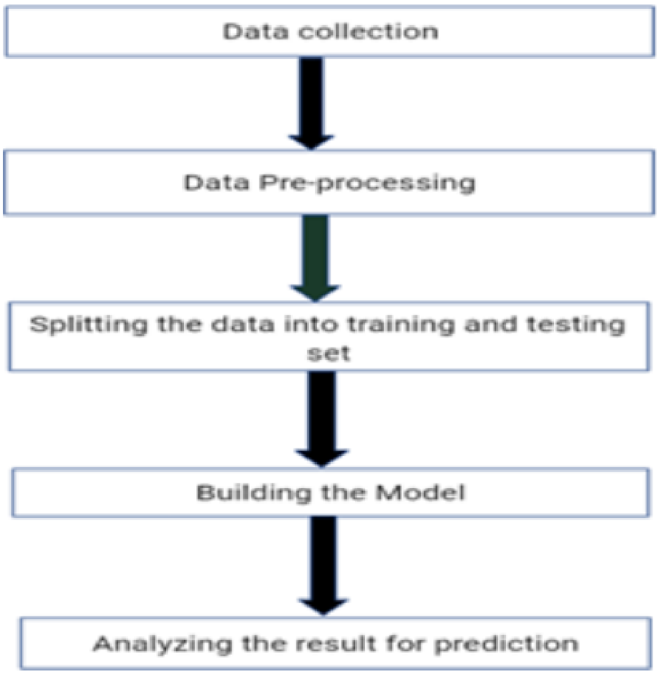

Data collection
Data Pre-processing
Splitting the data into training and testing set
Building the Model
Analyzing the result for prediction

Figure 2.1.8.1: Bank Loan Defaulter Prediction Methodology [31]

As for the naïve bayes, this classification strategy is classified into three models: Gaussian Model, Multinomial Model, and Bernoulli Model. For the support vector machine, the author has selected Linear Kernel, Gaussian RBF Kernel, and Polynomial Kernel for their study. From the results, the support vector machine is significantly better than its counterparts (Random Forest, Gaussian NB, Multinomial NB, SVM RBF Kernel, and SVM Poly Kernel). The SVM linear kernel achieves the precision score of 0.819 and the recall score of 0.985. However, SVM is computationally expensive by nature. In real world applications, the amount of data that SVM needs to process is a lot of times larger than the dataset used in proposed model. Training time and memory usage might become a concern for real-time loan approval systems.

2.1.9 Research on Financial Field Integrating Artificial Intelligence: Application Basis, Case Analysis, and SVR Model-Based Overnight

Non-performing loans have been observed to be in an increasing trend in recent years. Moreover, many small and medium enterprises in China have been facing difficulties in obtaining financing [12]. The author in [12] has demonstrated that how the intelligent risk control has been established as the standard practice for all banking and financial institutions with the help of AI to further improve the risk management. Table 2.1.9.1 shows the intelligent risk control flow sheet.

Pre-Loan	Loan period	Post-loan
-Marketing customer -Loan application -Machine learning -Face recognition	-Loan Review -Micro expression recognition -Speech recognition -Knowledge maping	-Loan management -Overdue collection -Knowledge mapping -Learning speech communication

Table 2.1.9.1: Intelligent risk control flow sheet [12]

During the pre-loan stage, it will first identify potential customers and promote loan services to them. When the intrigued customer submits the loan application, it will be reviewed and evaluated thoroughly. During loan period, machine learning method is deployed in the risk control process. Face recognition will authenticate the identity of the borrowers. In the post-loan stage, it focuses more on the management and collection of overdue loan payments. This intelligence risk control enhances risk control and improves the decision-making process, making sure the overall risk of defaulter is reduced and the decision is accurate. However, the limitation of the [12] is the dependency on the SVR algorithm. While SVR is suitable for small scale learning, its performance may vary depending on the size of the dataset and the parameters [12].

2.1.10 An Explainable AI Decision-Support-System to Automate Loan Underwriting

The widespread adoption of AI has brought many benefits to the community, but the transparency of the machine learning models has faced some roadblocks when it comes to lending decisions [32]. To overcome the issue, the author in [32] has proposed an explainable AI decision-support-system with belief-rule-base (BRB) to automate the entire loan underwriting process. The BRB system can explicitly work with the knowledge of experts. Not only that, through supervised learning, the system can also learn from the historical data. The system can explain the events that lead to the final decision for a loan application with the important of rules. From a business case study, the proposed system has proven to provide a balance between explainability and prediction accuracy. If any rejected applicants were denied for their loan applications, the textual explanation could be sent to them with reasons on why the system had done so. Hence, even non-technical people can understand the BRB structure and the decision-making process and its transparency assures companies to deploy this system in their companies. However, the development of the system is very time-consuming, and it requires deep technical knowledge. An expert is needed to finalize the final structure of the BRB system. Not only that, but the system is also not able to reject loan applications due to the inconsistency in the applications [32].

2.2 Comparison of Reviewed Techniques

2.2.1 Comparison of Reviewed Techniques using Deep Learning and Machine Learning Methods

Paper Title	**Techniques Involved**	**Advantages**	**Disadvantages**
A novel hybrid ensemble model based on tree-based method and deep learning method for default prediction [26]	Ensemble model consisting of RF, XGBoost, and LightGBM with Convolutional Neural Network (CNN).	The proposed model achieves good performance and is robust. The ensemble model improves the final overall results of the default prediction.	Hyperparameters are not optimized. It could lead to performance loss as features might not be able to capture the underlying pattern of the data effectively. Not only that, but the use of ensemble model also requires a lot of resources

			and is very computationally expensive.
Credit Score Prediction System using Deep Learning and K-Means Algorithms [28]	Deep learning model and K-Means Algorithm.	Able to predict customer with high accuracy even if it is deployed in the real-world scenarios. The deep learning model further enhances the accuracy of the proposed system by reducing the error.	Lack of interpretability. It is hard to interpret how the deep learning network has arrived at their decisions where in the context of the personal loan system, transparency is crucial.

Table 2.2.1.1: Comparison of techniques using both deep learning and machine learning methods.

Both [26] and [28] uses the help of deep learning model to extract meaningful and important features from the given training dataset. However, the difference is that [26] uses ensemble of classifier (RF, XGBoost, and LightGBM) while [28] only uses one classifier which is K-Means Algorithms. A great performance could be achieved with ensemble of classifier, but this method requires extensive number of resources and also the time needed for training and evaluation. Although the use of single classifier might perform worse than the ensemble of classifier, but in some cases where time is crucial and with limited resources, using only one classifier is considered good enough.

2.2.2 Comparison of Reviewed Techniques using Machine Learning methods

Paper Title	**Techniques Involved**	**Advantages**	**Disadvantages**
Improvement of personal loans	Logistic Regression,	Out of three machine learning	Performance might be impacted if the

granting methods in banks using machine learning methods and approaches in Palestine [27]	Decision Tree, and Random Forest algorithm.	algorithms, the decision tree achieves with highest accuracy. Furthermore, the author's proposed system complies with all the requirements and criteria of banks, which can be deemed suitable to be deployed in banks.	borrower has little credit history. Due to the inadequate data, the proposed system might have high rejection rate.
Methodology and Models for Individuals' Creditworthiness Management Using Digital Footprint Data and Machine Learning Methods [29]	Hierarchical clustering and k-means method, classification based on the stochastic gradient boosting (SGB) method, and with the digital footprints of the borrower.	As digital footprints are included in the analysis, it helps to better design the individual credit trajectories. Not only that, but it also improves the loan's quality as well as credit risks are reduced, and bank's stability is increased. The proposed system also helps with the efficiency of financial decisions.	Accuracy might be impacted as sometimes digital footprint do not necessarily represent the borrower's financial situation.

An Approach for Prediction of Loan using Machine Learning Algorithm [30]	Logistic Regression with Sigmoid function	Performs well with high accuracy of 81.1 percent with the given dataset.	The assumption made by Logistic Regression might result in high rejection rate.
Comparative Analysis of Bank Loan Defaulter Prediction Using Machine Learning Techniques [31]	Random Forest, Naïve Bayes, and the Support Vector Machine	Support Vector Machine performs the best than the other two models. It achieves high precision score as well as high recall rate.	Support Vector Machine is computationally expensive. Training time and memory usage become a concern for real-time loan approval systems.
An Explainable AI Decision-Support-System to Automate Loan Underwriting [32]	Belief-rule-based (BRB) system	This proposed system can provide explanation on the events that lead to the final decision. It provides a balance between explainability and prediction accuracy. A non-technical person could also operate the BRB system.	The development of the system is very time-consuming. It requires deep technical knowledge. The system could not reject loan due to the inconsistency in the loan applications.

Table 2.2.2.1: Comparison of techniques using machine learning methods.

All five of the papers compared in Table 2.2.2.1 use machine learning methods without deep learning model. There is no ensemble technique used for all the techniques reviewed. As in [27] and [31], several machine learning methods are used but each of them is compared and the best performing one is picked out as the final proposed model. Out of five papers, [29] has implemented a more modern approach as digital footprints of the borrower is accounted for the final decision of the loan approval process. Unlike the proposed model in [32], the other proposed models do not give explanation on how the final decision has arrived. With the belief-rule-based system, the borrower can get explanations on why their loan application has been rejected [32].

2.3 Limitation of Previous Studies

The author in [24] has argued that is it critical to find the right balance between the protection of the customer's personal data and the ethical aspects of the usage of AI. Moreover, the use of big data in AI in the context of loan system has to be in compliance with the laws as this issue has received tension from the regulators. Furthermore, the proposed work done by [26] did not optimize the hyperparameters, which could lead to the loss of performance as features might not be able to capture the underlying pattern of the data effectively. The proposed system of [27] might have high rejection rate if the borrower does not have extensive credit history. In [28], the proposed credit scoring system lacks interpretability. Its biggest drawback is that the final decision made by the deep learning network could not be explained explicitly to the borrowers. Although the use of digital footprints can boost the overall performance of the machine learning model, but without carefully selecting the meaningful and important aspects from the digital data, it could impact the final accuracy of the model which could affect the borrower's creditworthiness directly [29]. The author in [21] uses the fairness and explainability techniques in their proposed work. But the dataset size is small where the proposed approach could be prone to overfitting issues. The proposed logistic regression from [30] assumes the relationship between the independent and dependent variables. For instance, a person that is earning a high income might not necessarily correlate to high likelihood of loan approval. In [31], the proposed work is computationally expensive. The training time and memory required are the major concern when it is applied in real-time loan approval systems. The Support Vector Machine for Regression (SVR) in [12] has some limitation. While SVR is suitable for small scale learning, its performance may vary depending on the size of the dataset and the parameters. For the last literature reviewed, the technique used in [32] needs extensive amount of time and deep technical knowledges of experts during the development phase. Moreover, it could not reject loan applications due to the inconsistency in the loan application itself [32].

2.4 Proposed Solutions

This project aims to build a fast and efficient personal loan system using machine learning algorithm. With some of the limitations mentioned in the literature review, this project also aims to solve those limitations that are applicable to this project. With the issue of high rejection rate and the inadequate credit history, this project has proposed to incorporate the borrower's existing digital footprints data (social media, personal shopping preferences, online behaviours, habits etc.) into the personal loan system with the help of machine learning algorithm to predict the eligibility and the creditworthiness of the borrower with high accuracy. Furthermore, although there is no single best machine learning algorithm that applies universally to all scenarios, but this project will make use of the existing machine learning algorithm that is light-weight and could process borrower's data rapidly in real-time scenarios. With the laws, regulations, and ethical concerns regarding the use of AI in personal loan systems, this project will take these concerns into consideration during the development of the proposed personal loan system. This project will make sure the final deliverable is compliance with the regulations and strikes the balance between the borrower's personal data and the ethical aspects of it. With the issue of interpretation, this project hopes to incorporate SHAP values into the final personal loan system. If the borrower's loan application is rejected by the proposed personal loan system, they will be provided with explanation and meaningful feedback on why the system has made that decision.

For this project, a gradient boosting model (XGBoost) will be used as the sole machine learning algorithm for the personal loan system. It is a strong choice as it yields high predictive accuracy. Furthermore, incomplete, and missing data is common in real-world datasets. XGBoost can automatically handle and impute missing data and make predictions on the missing features. Lastly, XGBoost provides insights into feature importance. In the context of this project, XGBoost allows the developers to understand which features are the most impactful to determine the credibility of the borrower.

CHAPTER 3 System Methodology / Approach

This chapter will briefly describe the system methodology of the whole project. All the details and approaches are shown in the system architecture diagram.

3.1 System Design Diagram

3.1.1 System Architecture Diagram

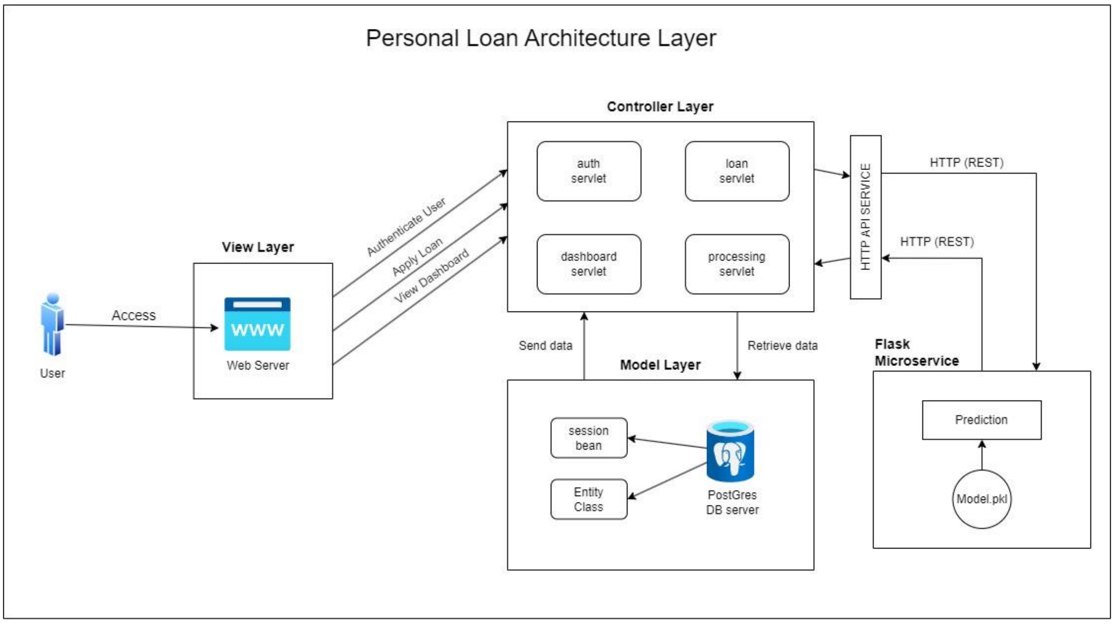


Figure 3.1.1.1: System architecture diagram of the Personal Loan Processing System

According to Figure 3.1.1.1, the methodology of this project is divided into two main phases. The first phase is the training and tuning of the machine learning model which will then be packaged into a microservice using Flask framework. The second phase will be the implementation of the web application using the Model-View-Controller (MVC) framework.

Initially, a real loan prediction dataset will be procured from the Internet. The dataset will then be split into training and testing sets. Both of these sets will be cleaned and preprocessed for any null values, outliers, etc. Any relevant features will be identified and engineered to enhance the predictive power of the model. Once the major tasks are done, the dataset will be used to train the XGBoost model and test it on unseen data. Hyperparameter tuning will be conducted to further optimize the model's performance. When the first model is complete, the original dataset will be altered with new synthetic digital footprint data that simulates the real-world conditions by following closely to the assumptions and rules defined. A second XGBoost model will be trained on the newly altered dataset. The final output is the probability of the loan approval given the financial details and digital footprint data provided by the applicant. After that, the final model will be packaged into a pickle file and wrapped in a Flask-based microservice.

The second phase is the development of the personal loan web application using the MVC framework. A login module will be created to only allow legitimate loan applicant to access the web application. A dashboard will be developed to allow the applicant to see their pending, accepted, and rejected application and a general overview on the overall application sent to the system. The applicant can access the data collection module to type in their relevant financial details and digital footprint data if consented. Once the applicant has submitted their application form, the web server will then connect to the Flask-based microservice through the HTTP API Service and send all the data to the model. The microservice will procced to send back the prediction result to the controller and the controller will interpret it to the applicant.

3.1.2 Use Case Diagram and Description

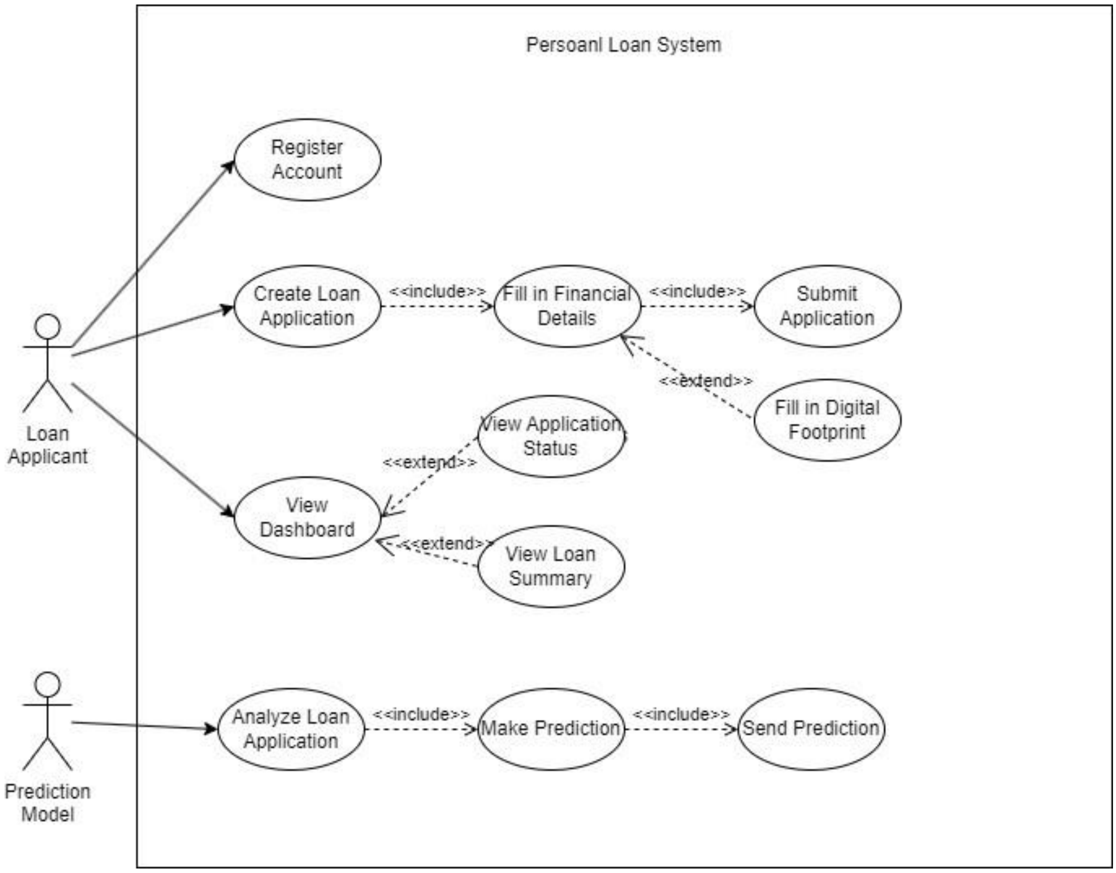


Figure 3.1.2.1: Use case diagram of the personal loan web application.

Figure 3.1.2.1 shows how the user can interact with the web application. The main user, which is the loan applicant, has several functions. Loan applicants will be required to register an account before they can make use of the personal loan system. Once their accounts have been verified, the user can create a new loan application. The user will be asked to provide relevant financial details as well as their digital footprint data if applicable. After that, the loan application will then be submitted to the personal loan system for further action. The prediction model will analyze and review the loan application. Once the model has made the prediction, the final result will be sent to relevant controllers for further processing. Lastly, the user can view the dashboard and check for any approved, rejected or pending loan applications. The user can look at their applied loan application history too.

3.1.3 Activity Diagram

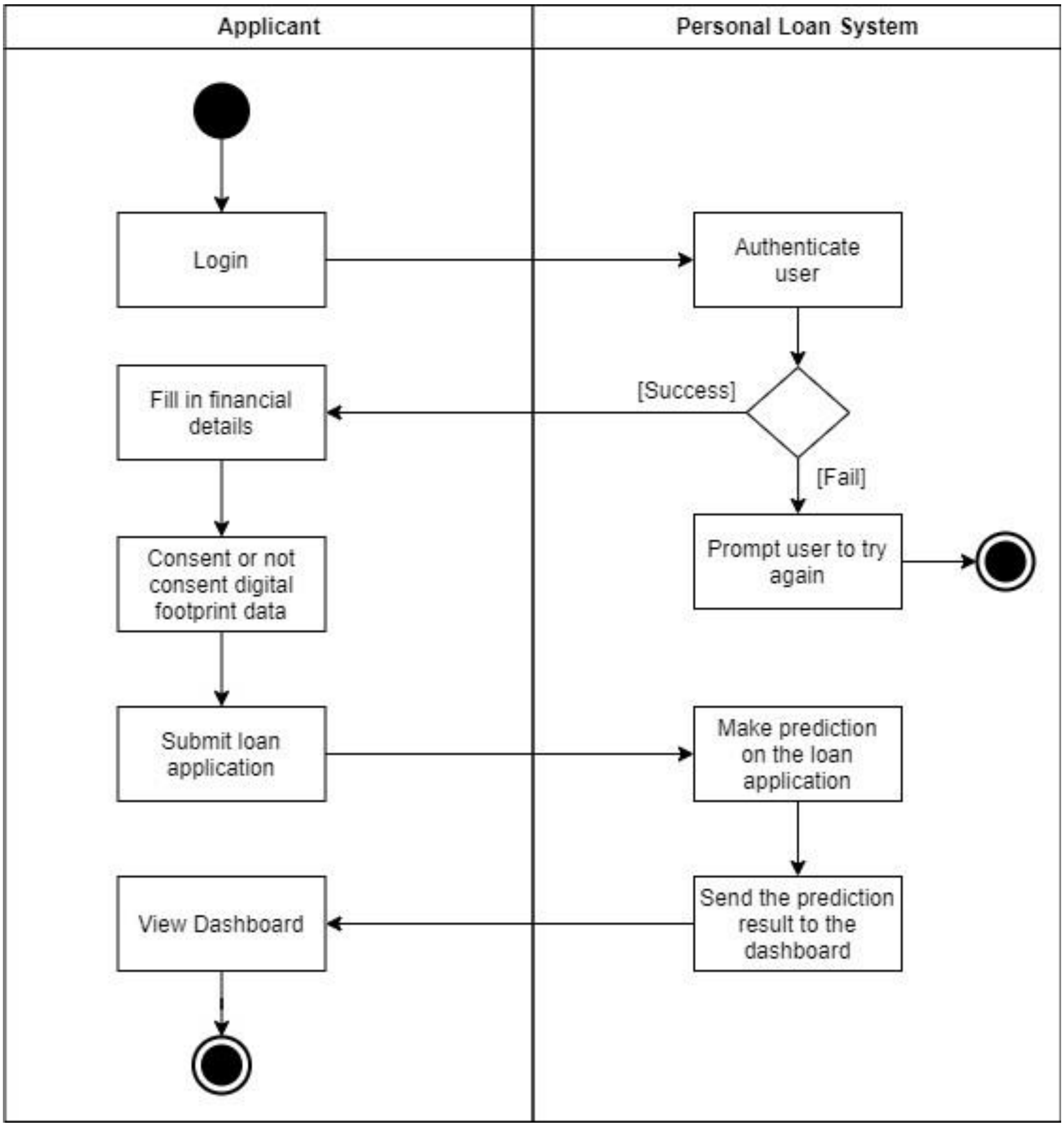

Applicant
Personal Loan System
Login
Authenticate user
[Success]
Fill in financial details
[Fail]
Prompt user to try again
Consent or not consent digital footprint data
Submit loan application
Make prediction on the loan application
View Dashboard
Send the prediction result to the dashboard

Figure 3.1.3.1: The activity diagram of the personal loan web application.

Figure 3.1.3.1 describes the overall flow from the user logging into the web application to start applying for a personal loan and viewing their final result. Before users can start applying for a personal loan, they need to register for a valid account first. Once they have done that, they will be prompted to sign in. The personal loan system will authenticate the user and redirects them back to the login page if they provide the wrong login details. If they have been successfully authenticated, they can start applying for a personal loan. Users will be required to fill in compulsory personal and financial information. They will be prompted to provide their digital footprint data if they consented to do so. The submitted loan application will be processed by the personal loan system and a prediction will be made. This result will then be sent back to the dashboard. Users can view their loan application through the dashboard.

3.2 Project Timeline

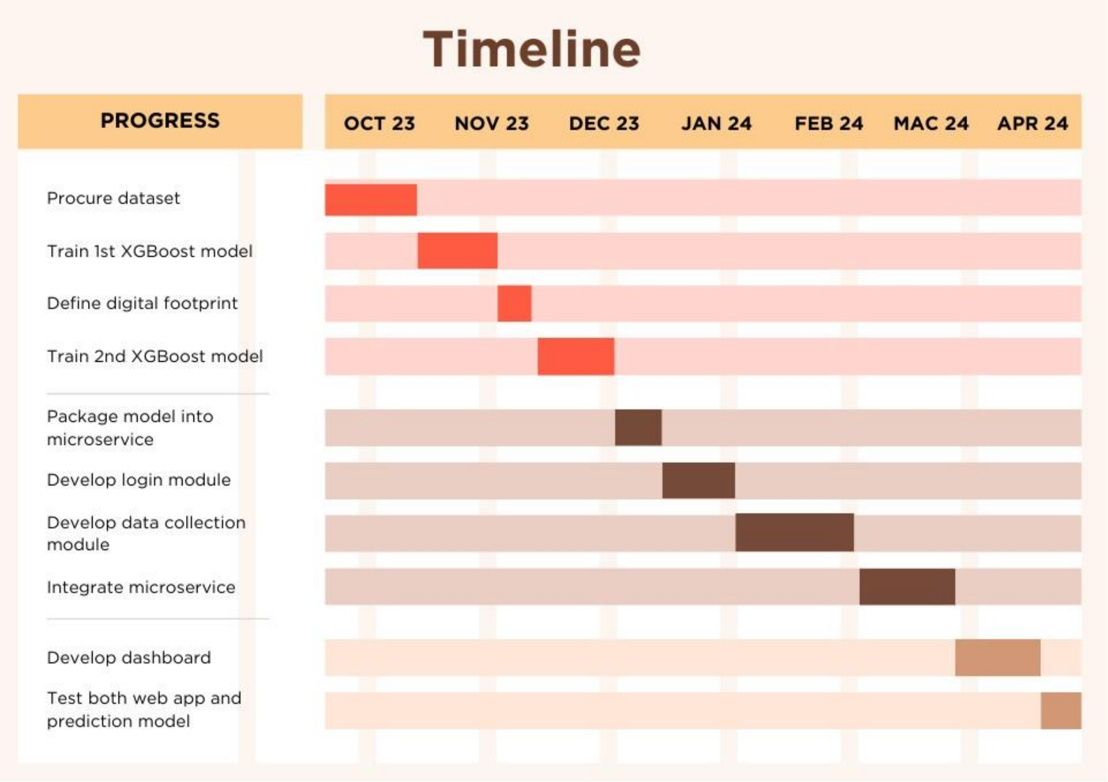


Figure 3.2.1: Timeline of the project.

The timeline of this project is divided into two phases. The first phase was executed during year three and second trimester. In the first phase, the training and testing of the XGBoost model were completed by 9th December 2023. In the second phase (Year Three Trimester Three), a new XGBoost model was retrained on a more diverse and larger dataset. Once done, an online web application of the personal loan system was developed. The final XGBoost model was deployed to the web application in April 2024. Then, it is made available for public demonstration.

CHAPTER 4 System Design

4.1 System Block Diagram

4.1.1 Personal Loan AI Block Diagram

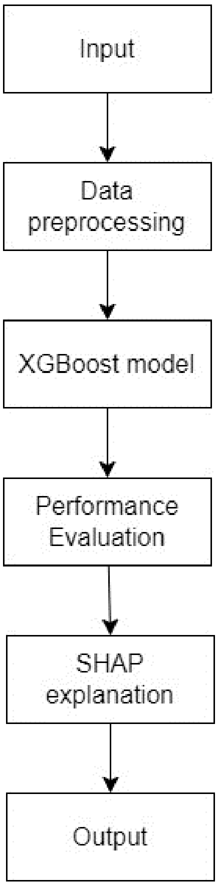


Figure 4.1.1.1: Block diagram of the personal loan system.

4.1.2 Web Application Block Diagram

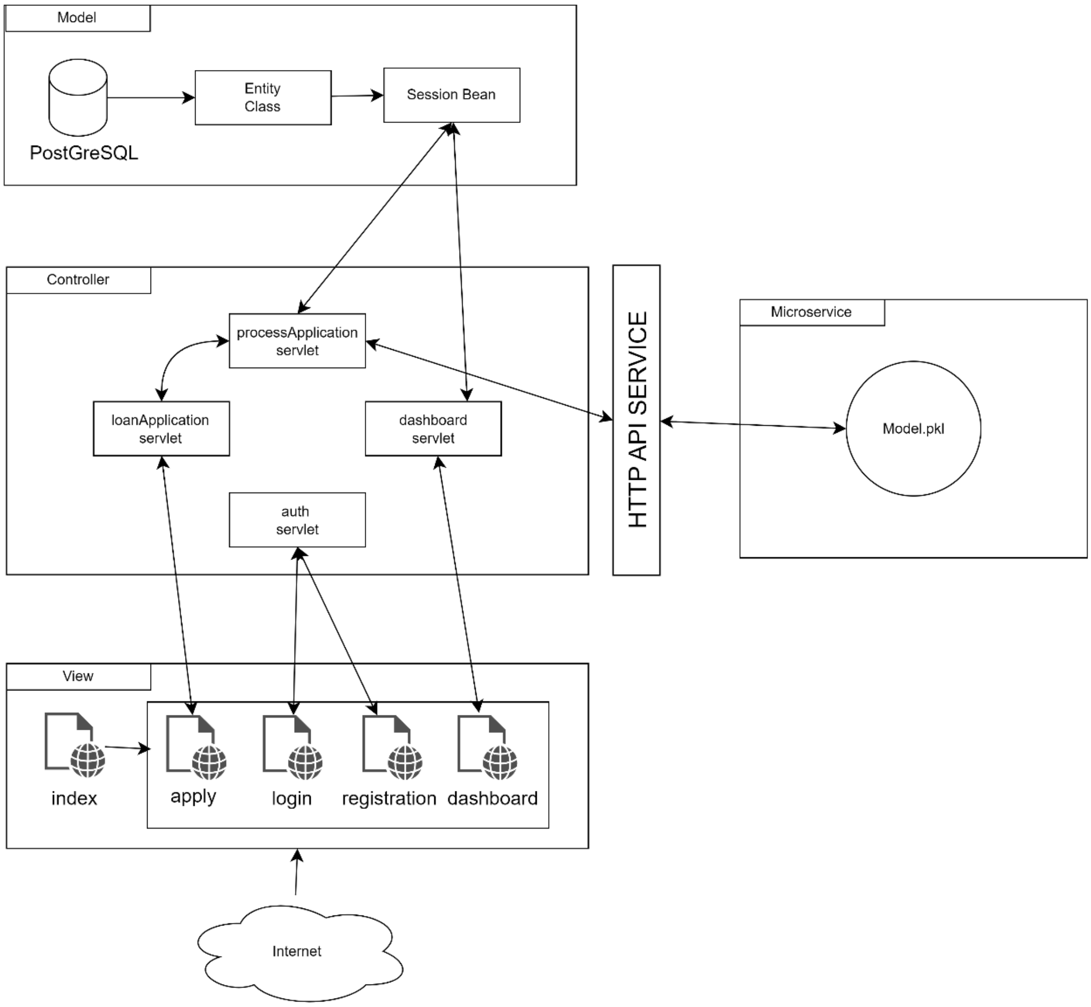


Figure 4.1.2.1: Block diagram of the web application.

4.2 System Components Specifications

4.2.1 Personal Loan AI Components Specifications

Input:

The required input was the personal loan prediction dataset procured from Kaggle. It contained the historical data about the borrowers of LendingClub, a financial services company headquartered in San Francisco, California. This dataset contained 2,260,701 instances and it had 151 features. It was then loaded using into the Jupyter Notebook using the *read_csv* function from the Pandas library.

Data preprocessing:

In this step, raw data collected from the input stage will be processed for analysis. Tasks involved are dropping unmeaningful features, handling missing data, normalizing data, removing noise and outliers, encoding categorical variables, and performing feature engineering before feeding it to the XGBoost model.

The initial total number of features were 151. Due to the fact that having these many features might be deteriorating to the performance of the machine learning model, some of the not meaningful columns have been dropped. These dropped features were mostly empty. The final concluded features have been reduced down to 30 features. These features are as follows:

1. loan_amnt
2. term
3. int_rate
4. installment
5. grade
6. sub_grade
7. emp_title
8. emp_length
9. home_ownership
10. annual_inc
11. verification_status
12. issue_d
13. loan_status
14. purpose
15. title
16. zip_code
17. addr_state
18. dti
19. earliest_cr_line
20. open_acc
21. pub_rec
22. revol_bal

23. revol_util
24. total_acc
25. initial_list_status
26. application_type
27. mort_acc
28. pub_rec_bankruptcies
29. fico_range_high
30. fico_range_low

Further data processing is needed to ensure all these features prove to be significant to the machine learning model. Each feature is carefully analyzed for its importance to the final prediction of the loan application. Here is the list of features that were dropped after a careful analysis:

1. emp_title

   This column contains 378,007 unique values which was way too much to create dummies for it. Moreover, it proved to be not as significant as some of its values do not make sense as well. It might due the typing error from the borrower or just straight up ignorance.

2. emp_length

   This column was analyzed using count plot from Seaborn library. It was visualized alongside the target variable, loan status, and from the graph, it can be concluded that there was not much of a difference between the loan status and the employment length. So, it was dropped as it did not add anything useful to the prediction.

3. title

   Similarly to the column "emp_title", it contained 61,682 unique values. Due to the later trouble of encoding it, it was dropped.

4. zip_code

The correlation between the zip code and loan status was very low. Additionally, the format of the zip code only contained the first three digits, with the remaining removed due to privacy concerns. Because of this, the feature was dropped.

5. grade & sub_grade

   Since the grade was a sub feature of sub_grade, it was dropped.

6. issue_d

   This column "issue_d" would be a major issue for data leakage if it was to be included in the training of the machine learning model. According to the definition from the dataset, this column indicated when the borrower was issued the loan.

After dropping the six unmeaningful features, the final features were once again reduced down to 24. The next preprocessing step is to encode the categorical features. It is important to encode the categorical features before performing the train-test-split as to ensure both of the final train and test datasets have the same numbers of features. The following are the columns that have been encoded using the "get_dummies" function from the Pandas library:

1. addr_state
2. sub_grade
3. application_type
4. initial_list_status
5. verification_status
6. purpose
7. home_ownership
8. term

Before splitting the dataset, a random user was extracted to be used in later testing stage. The dataset was then split using the train_test_split function from SKLearn. The following are the parameters of the function:

1. test_size = 0.2
2. random_state = 42
3. stratify = y

Missing data imputation was only done after the splitting stage as to avoid the issue of the data leakage. Below is the list of imputed features:

1. dti

   This feature was imputed by using the mean of the column.

2. revol_util

   This feature was imputed by using the mean of the column.

3. mort_acc

   The missing values were imputed based on the most common value of mort_acc within groups defined by the ‘open_acc’ column.

4. pub_rec_bankruptcies

   As for this column, five categories were created for annual income. The feature “pub_rec_bankruptcies” was then imputed based on the mean value of its column within groups defined by the five categories of the annual income defined earlier.

Once all features that had missing values had been imputed, the numerical features such as “annual_inc” and “revol_bal” were transformed using log transformation. The feature “earliest_cr_line” was altered to only include month and year. Moreover, the average of these two “fico_range_high” and “fico_range_low” features had been calculated and placed into a new feature named “fico”.

After that, a new dataset copied from the existing dataset will be altered with digital footprint data based on rules and assumptions defined that simulate the real-world conditions. The rules and assumptions are as follows:

1. Generally, iOS user is perceived to be in the top quartile of income as compared to Android User. But these may not be necessarily true anymore in this day and age as people can now buy apple products on a contract. Although more and more people can afford apple products, but the assumption still holds true to some extent. People with higher income are more willing to purchase expensive products as compared to low-income earners. This also applies to Windows and Macintosh users [33].

2. People with higher incomes tend to have paid email hosts. It might be due to them holding higher positions in their company; hence, owning a paid email host or even business owners [33].

3. Although digital wallet transaction and shopping history cannot replace credit history, but it is still a good indicator to tell whether the loan applicant is willing to pay back the loan or not. Extensive transaction and shopping history might indicate that the loan applicant is financially stable which allows them to spend more on things they like.

4. The impact of social media activity is subjective depending on the scenario. In future, with a more robust deep learning model, it could learn the nuances in the posts on the applicant's social media pages. The pattern of defaulting users could be deduced from there. But for the sake of simplicity and clarity, this project will assume whether the user is active or inactive on social media.

5. The number of subscriptions that the applicant has could give an insight into their financial situation. High income earners are more willing to subscribe to

services while low-income earners might prioritize these funds on more important things.

With the rules and assumptions defined, seven new digital data footprint features were then created. These features are as follows:

1. Device_Type

   Type of device the applicant is using (mobile or tablet or computer). When the applicant uses a computer, the probability of the operating system being Windows or Macintosh are 43% and 57% respectively. When the applicant uses a tablet, the probability of the operating system being iOS or Android are 57% and 43% respectively. When the applicant uses a mobile phone, the probability of the operating system being iOS or Android are 55% and 45% respectively.

   Windows and Macintosh have the least probability difference among other devices as not many people use or know how to use Macintosh computer. In this case, Windows is a much popular choice among the low- and high-income households.

   Android tablet is not as popular as iPad. Most of tablets that people own are iPad as compared to an android tablet. Since tablet is not considered as important as mobile phone, many low-income households do not even bother purchasing tablet in the first place. Most of the high-income household will consider picking up iPad for their personal use or even to supplement their business purposes. In this case, iPad has higher probability than windows.

2. Operating_System

   Operating system of the device (Windows or Macintosh or Android or iOS).

3. Email_Host_Type

The email host of the applicant can be either paid or free version. When the loan status is true, the probability of applicants using paid email host is 53% as compared to free email host which is 47%.

The probability difference between free and paid is not large as correlation does not imply causation. Being rich does not mean the person will own a paid email address. But higher income might indicate that the person is a professional or an entrepreneur which usually have their own professional email address.

4. Online_Shopping_Frequency
   How often the applicant shops online (less or moderate or frequent).

5. Social_Media_Activity
   Whether the applicant is active or inactive on social media.

6. Digital_Wallet_Transaction
   Similar concept to credit history but in the sense of history of purchases using digital wallet (1 or 0).

   Although transaction and shopping history cannot replace credit history, but it is still a good indicator to tell whether the loan applicant is willing to pay back the loan or not. Extensive transaction and shopping history might indicate that the loan applicant is financially stable which allows them to spend more on things they like.

7. Online_Subscription
   Number of services that the user has subscribed to (1 or 2 or 3 – more).

The preprocessing of the digital footprint dataset was exactly the same as the original unaltered dataset above.

XGBoost Model:

XGBoost is a gradient boosting model, and it is a powerful machine learning algorithm that is usually used for predictive modeling. In this stage, the historical loan data, including the processed raw data, digital footprints were used to train the XGBoost model. Relationships and patterns in the data will be learned and be used to predict the creditworthiness of the borrower and to reject or approve the loan application. The hyperparameters of the model are as follows:

1. Learning_rate = 0.01
2. N_estimators = 150
3. Max_depth = 4
4. Objective = 'binary:logistic'
5. Eval_metric = 'auc'

The XGBoost model was then fitted into a pipeline. Alongside model, a Synthetic Minority Oversampling Technique (SMOTE) was used to mitigate the class imbalance issue. The pipeline was then used to train on the dataset and predict the testing dataset.

Performance Evaluation:

Once the training of the XGBoost model was done, its performance was evaluated. Some of the popular evaluation metrics including confusion matrix, accuracy, precision, F1-score, recall, ROC-AUC plot, etc. were used for the evaluation purpose. Adjustments and fine-tuning of the XGBoost model may be necessary based on the final evaluated results.

SHAP explanation:

SHAP values (Shapley Additive exPlanations) was applied to the final loan approval process. It allows expert knowledge and rules to refine the decision-making process. With that, factors such as credit score, income, and the information extracted could be weighed of its importance which will eventually enhance the final decision. Any rejected loan application can then refer to the SHAP values to explain which features contributed the most to the rejection of the application. The SHAP values were interpreted using the feature importance bar plot and beeswarm plot.

Output:

At the last stage was the final output. The proposed personal loan system delivered the loan approval decision to the applicant. In case the loan application is rejected, relevant explanations and reasons for the said decision will be provided as necessary.

4.2.2 Web Application Components Specifications

View:

The view component contains five jsp pages which were index, apply, login, registration, and dashboard. When the user browses the personal loan web application, they will always land at the index page first. From the index page, the user can navigate to the remaining pages. The apply and dashboard pages only allowed registered users to access it. If they tried to access it without signing in, they will be redirected to the login page. Once signed in, the user can start applying for a personal loan and view the application status in the dashboard page. If the user does not have an account, they can register for one.

Controller:

This component will manage the user requests from the view component. It communicates with the database for data storage or retrieval and interacts with the XGBoost model for making loan predictions. In this component, four java servlets had been defined which were the authentication servlet (auth), loan application servlet (loanApplication), process application servlet (processApplication), and the dashboard servlet. The authentication servlet is responsible for authenticating the login information and also registering new users to the database. When the user starts to apply for a personal loan from the index page, all the data typed in will be sent to the loan application servlet and these data will then be displayed in the application form. Once the user has done filling in the remaining required details, the loan application will be processed by the process application servlet. This servlet handles and formats necessary details before sending it to the loan prediction model. The Flask-based microservice component contains the trained personal loan machine learning model and it connects to the web application through REST application interface programming (API). All the loan application will be processed and predicted by the model and these data will be sent back to the calling servlet. The dashboard servlet is responsible for displaying the prediction made by the model.

4.3 Modules and Sub-modules of the Web Application

In a web application, it is important to identify the important modules and submodules so that the users will not be confused when navigating around. There are three main modules in this web application.

The first module is the login module. This module enables for the user to gain access to the personal loan system. It is used to verify and authenticate the loan applicant before granting them permission to make use of the system. The submodules of this module are user registration and user login. When the user registers for an account, the user registration submodule will then be invoked. The user can then login to the web application by using the user login submodule. The basic information such as the applicant's first name, last name, date of birth, gender will be automatically saved into the database.

The second module is the data collection module. This module is necessary for the user before they can apply for a loan. The user will interact with the submodules like financial details verification, employment verification, and digital footprints data upload. Different submodules serve different purposes. As for financial details verification, the user will have to provide their necessary financial details such as annual income, credit history, etc., to the system. If the user has provided their consent to the system, they can provide their relevant digital footprints data to the system so that the final result will be more accurate.

The third module is the personal loan module. This module will process the loan application automatically. The submodule includes XGBoost model for assessing creditworthiness, making prediction, and providing explanation based on SHAP values. These submodules will only be invoked when the user submits the loan application to the system. After that, the loan system will interpret the final result to the user along with the explanations based on the status of the loan application. The interpreted information can let the applicant know why their loan application has been rejected.

4.4 Class Diagram of the Personal Loan System

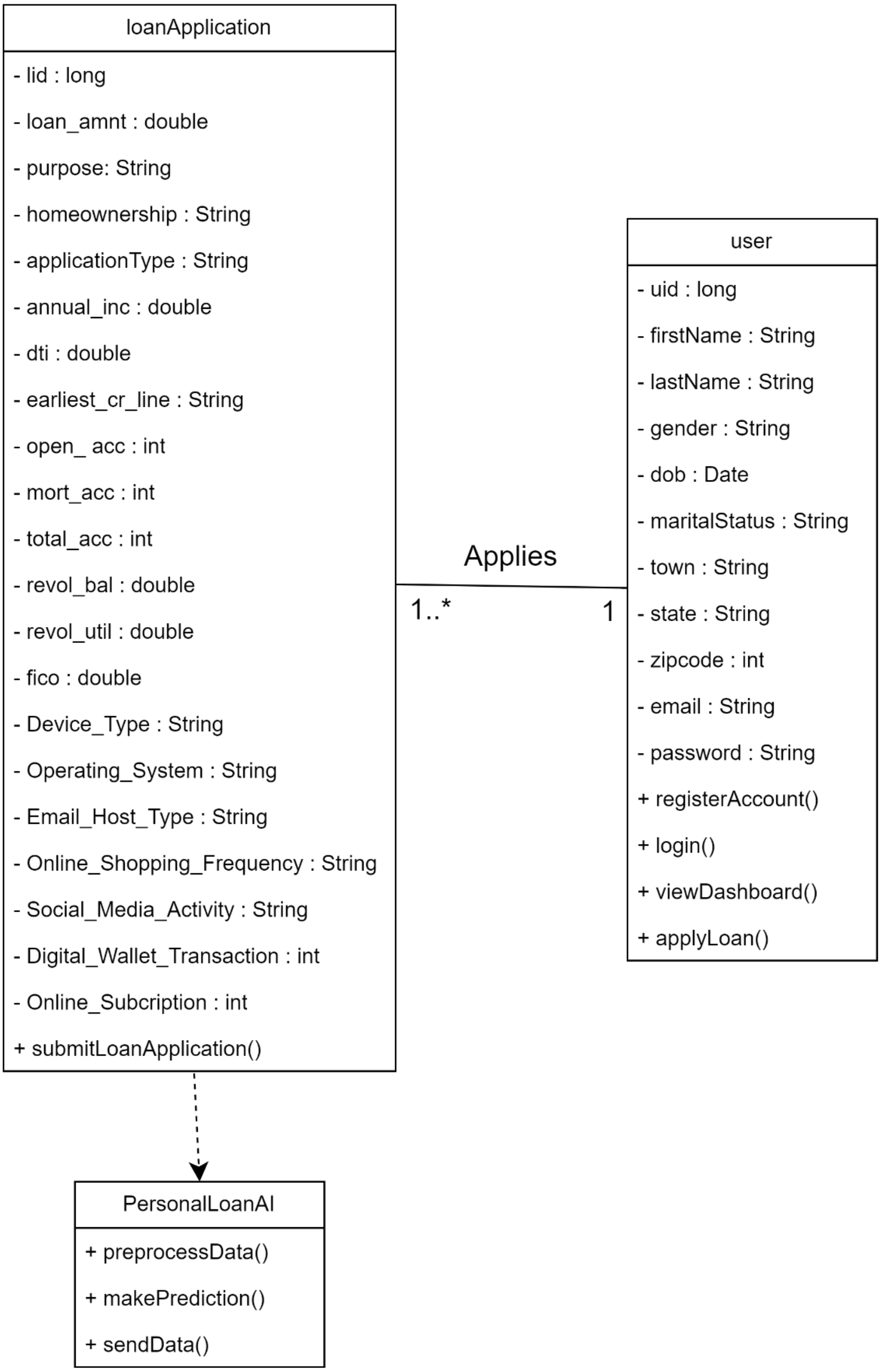


Figure 4.4.1: Class diagram of the Personal Loan System.

For the personal loan system, there are three main classes which is shown in Figure 4.4.1. The user class contains all the basic information of themselves. Four main functions that can be performed by the user are login, register, view dashboard, and apply for a personal loan. The loanApplication class has all the necessary personal, financial, and digital footprint data information. The only function of this class is to send information to the personal loan AI class. This prediction class will pre-process the received data, make a prediction on it and send the results back to the controller.

CHAPTER 5 System Implementation

In this chapter, the hardware and software setup will be discussed. Moreover, the setting and configuration of the web application as well as the personal loan machine learning model will be included in the code snippets. The screen capture of the user interface of the web application will showcase the system operation. Last but not least, the implementation issues and challenges faced during this project will be outlined as well.

5.1 Hardware Setup

The hardware involved in this project is a laptop. The laptop will be used throughout the development of the personal loan system as well as during testing and deployment phases. The specification of the laptop is stated in Table 5.1.1.

Description	Specifications
Model	LAPTOP-3I6F1UI3
Processor	AMD Ryzen 5 4500U
Operating System	Windows 11 Home Single Language
Graphic	Radeon Graphics
Memory	12GB DDR4 RAM
Storage	512 GB SSD

Table 5.1.1: Specifications of laptop.

5.2 Software Setup

The software setup is as follows:

Component Type	Software / Tools
Integrated Development Environment (IDE)	IntelliJ IDEA 2021.3, Jupyter Notebook
Framework	Flask, Bootstrap, Maven, Java Servlet, MVC architecture
Database	PostgreSQL
Language	Java, Python, HTML, CSS, Javascript, JQuery

Table 5.2.1: Software specifications of the personal loan system.

5.2.1 Software Libraries and Packages

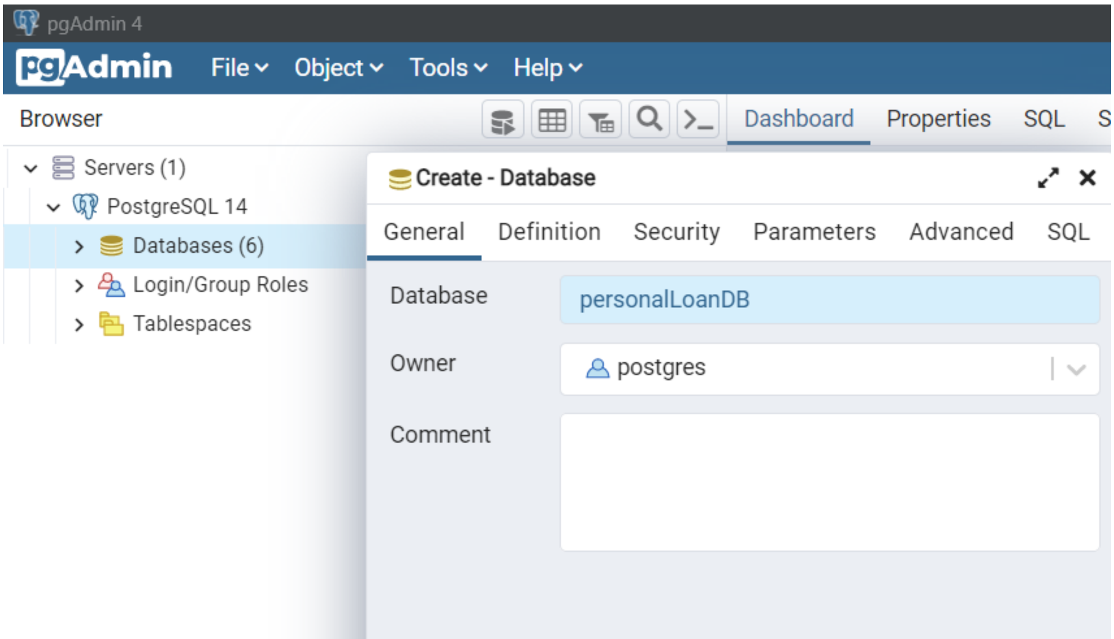
The following libraries and packages are necessary to develop the web application and also the personal loan prediction model:

1. from sklearn.preprocessing import LabelEncoder
2. from sklearn.metrics import roc_curve, roc_auc_score, balanced_accuracy_score
3. from sklearn.model_selection import train_test_split
4. from sklearn.model_selection import cross_val_score, StratifiedKFold, cross_validate, KFold
5. from sklearn.metrics import confusion_matrix
6. from sklearn.metrics import classification_report
7. from sklearn.metrics import accuracy_score
8. from imblearn.over_sampling import SMOTE
9. from imblearn.over_sampling import RandomOverSampler
10. from imblearn.pipeline import Pipeline
11. from xgboost import XGBClassifier
12. from xgboost import plot_importance
13. import shap
14. import numpy as np
15. import pandas as pd
16. Maven

17. Hibernate
18. Java EE: Enterprise Java Beans (EJB)
19. JPA buddy 2022.1.3 haulmont
20. JBoss/WildFly 17.0.1 Final

5.3 Setting and Configuration

5.3.1 PostgreSQL Database Setup

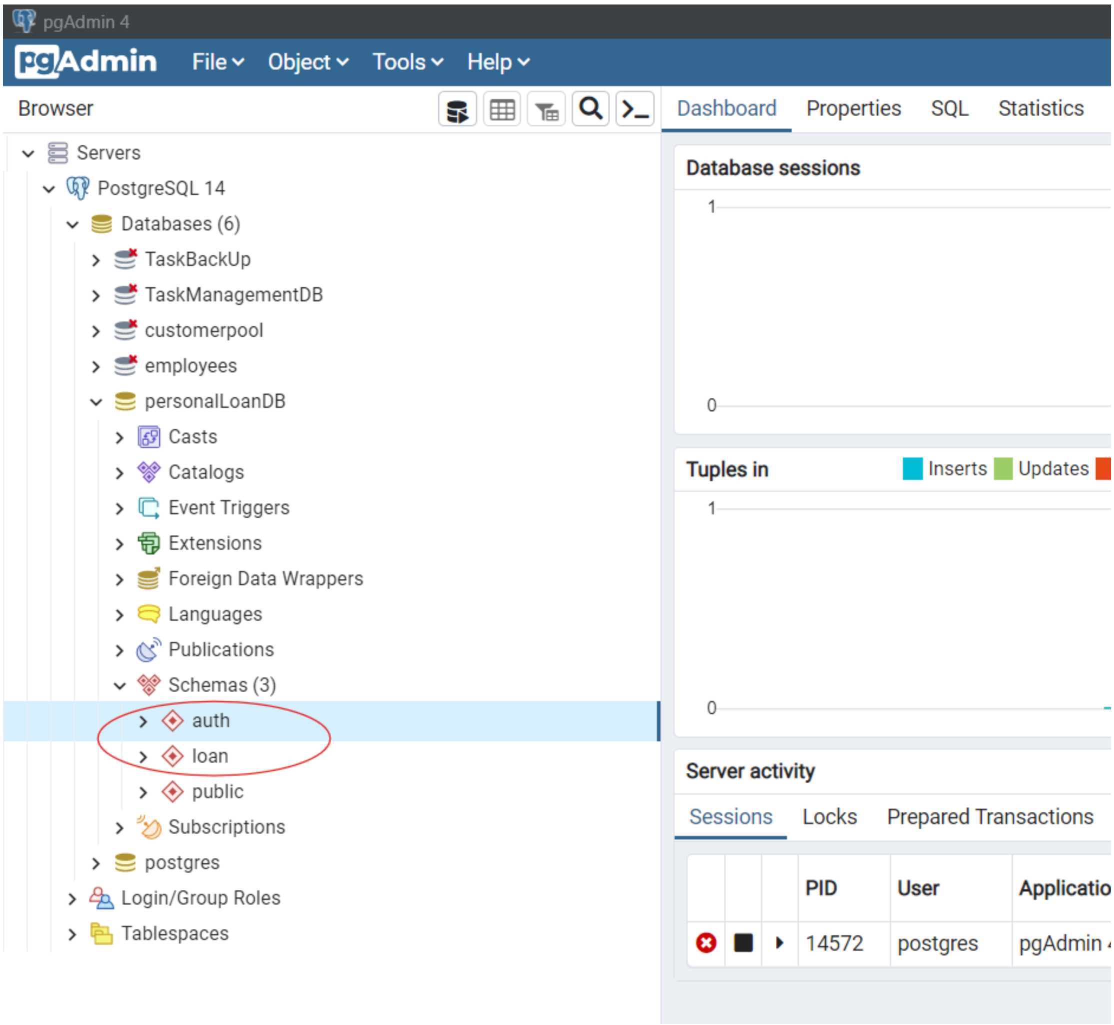


Database: personalLoanDB

Owner: postgres

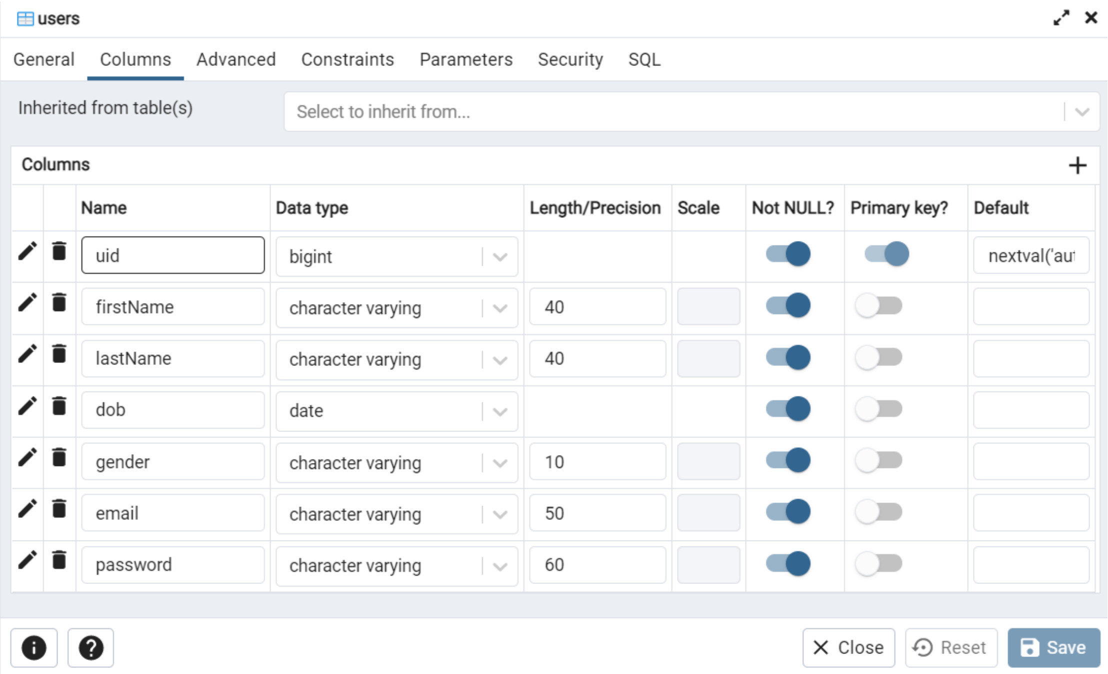
Comment:

Figure 5.3.1.1: Screenshot of the creation of project’s main database.



			PID	User	Applicatio
✕	■	▸	14572	postgres	pgAdmin

Figure 5.3.1.2: Screenshot of two newly created schemas.

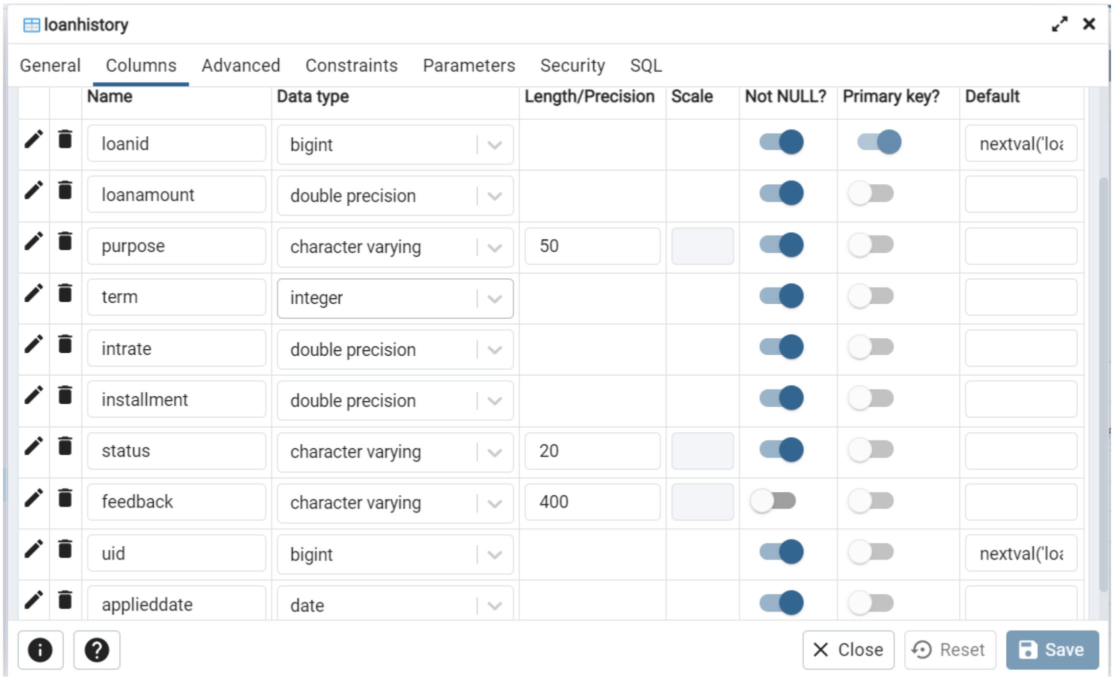


Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
uid	bigint			☑	☑	nextval('au
firstName	character varying	40		☑	☐	
lastName	character varying	40		☑	☐	
dob	date			☑	☐	
gender	character varying	10		☑	☐	
email	character varying	50		☑	☐	
password	character varying	60		☑	☐	

Figure 5.3.1.3: Screenshot of the definition of the user table under the auth schema.

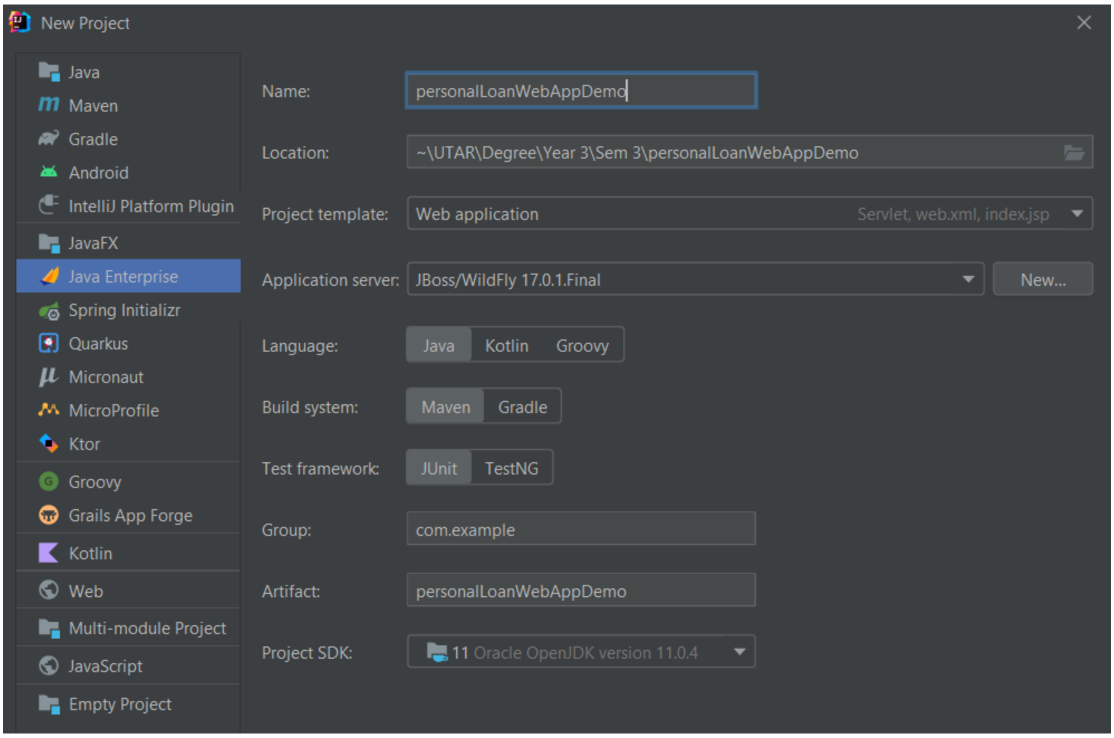


Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?	Default
loanid	bigint			☑	☑	nextval('loa
loanamount	double precision			☑	☐	
purpose	character varying	50		☑	☐	
term	integer			☑	☐	
intrate	double precision			☑	☐	
installment	double precision			☑	☐	
status	character varying	20		☑	☐	
feedback	character varying	400		☐	☐	
uid	bigint			☑	☐	nextval('loa
applieddate	date			☑	☐	

Figure 5.3.1.4: Screenshot of the definition of the loanhistory table under the loan schema.

For the column “uid”, it was the foreign key which was referencing to the primary key of the user table under the auth schema.

5.3.2 Web Application Setup



Name: personalLoanWebAppDemo

Location: ~\UTAR\Degree\Year 3\Sem 3\personalLoanWebAppDemo

Project template: Web application Servlet, web.xml, index.jsp

Application server: JBoss/WildFly 17.0.1.Final

Language: Java

Build system: Maven

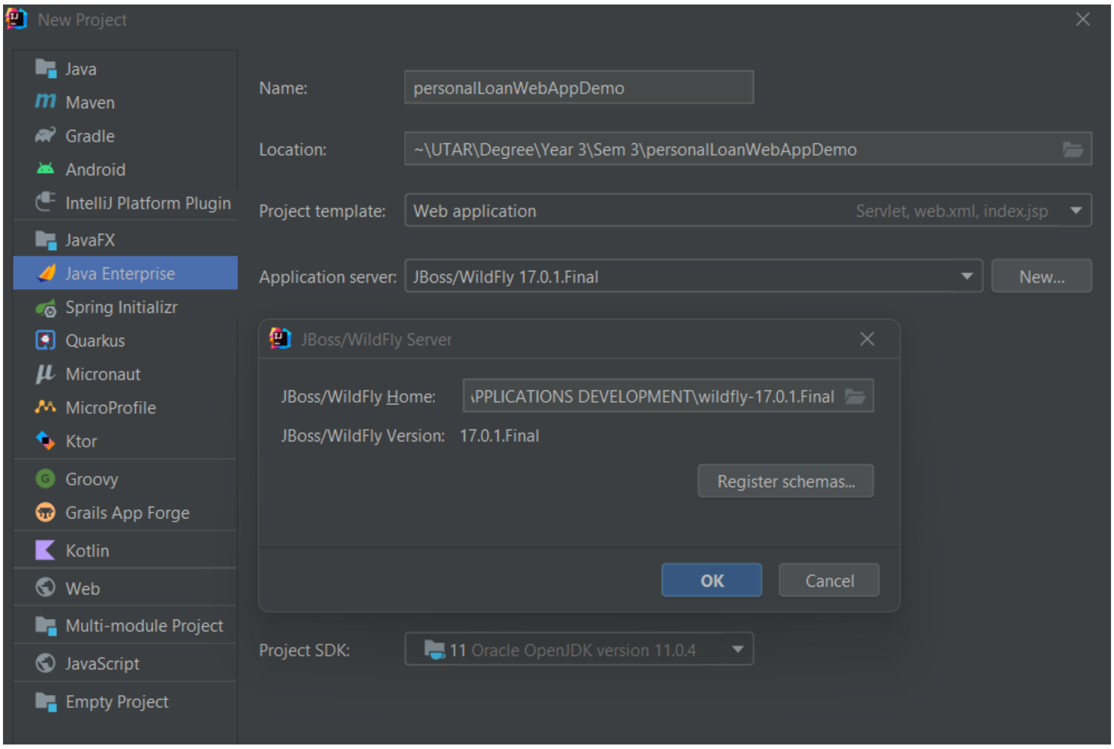
Test framework: JUnit

Group: com.example

Artifact: personalLoanWebAppDemo

Project SDK: 11 Oracle OpenJDK version 11.0.4

Figure 5.3.2.1: Screenshot of the new project's configuration.



Name: personalLoanWebAppDemo

Location: ~\UTAR\Degree\Year 3\Sem 3\personalLoanWebAppDemo

Project template: Web application Servlet, web.xml, index.jsp

Application server: JBoss/WildFly 17.0.1.Final

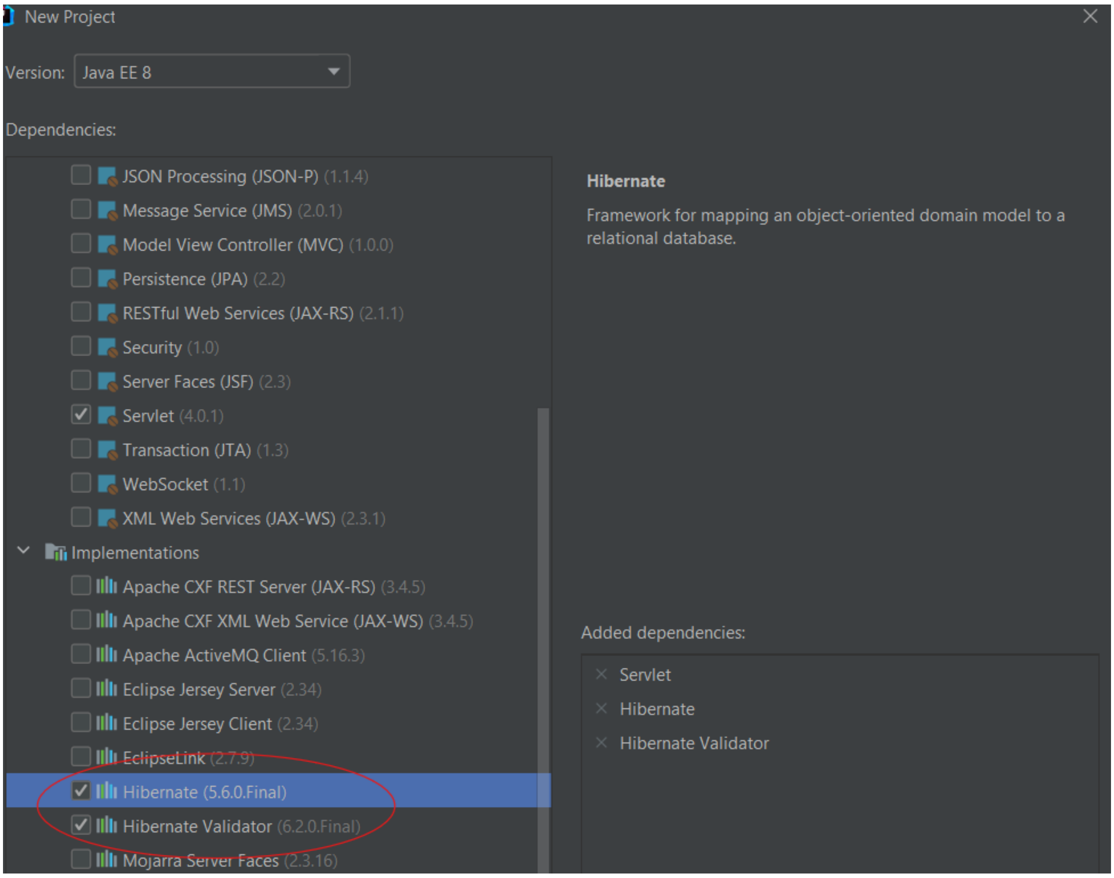
JBoss/WildFly Home: \PPLICATIONS DEVELOPMENT\wildfly-17.0.1.Final

JBoss/WildFly Version: 17.0.1.Final

Project SDK: 11 Oracle OpenJDK version 11.0.4

Figure 5.3.2.2: Screenshot of the JBoss/WildFly configuration

The wildfly server “wildfly-17.0.1.Final” was obtained from official wildfly website. The JBoss/WildFly Home directory linked to where the downloaded folder resided.



New Project

Version: Java EE 8

Dependencies:

- ☐ JSON Processing (JSON-P) (1.1.4)
- ☐ Message Service (JMS) (2.0.1)
- ☐ Model View Controller (MVC) (1.0.0)
- ☐ Persistence (JPA) (2.2)
- ☐ RESTful Web Services (JAX-RS) (2.1.1)
- ☐ Security (1.0)
- ☐ Server Faces (JSF) (2.3)
- ☑ Servlet (4.0.1)
- ☐ Transaction (JTA) (1.3)
- ☐ WebSocket (1.1)
- ☐ XML Web Services (JAX-WS) (2.3.1)

Implementations

- ☐ Apache CXF REST Server (JAX-RS) (3.4.5)
- ☐ Apache CXF XML Web Service (JAX-WS) (3.4.5)
- ☐ Apache ActiveMQ Client (5.16.3)
- ☐ Eclipse Jersey Server (2.34)
- ☐ Eclipse Jersey Client (2.34)
- ☐ EclipseLink (2.7.9)
- ☑ Hibernate (5.6.0.Final)
- ☑ Hibernate Validator (6.2.0.Final)
- ☐ Mojarra Server Faces (2.3.16)

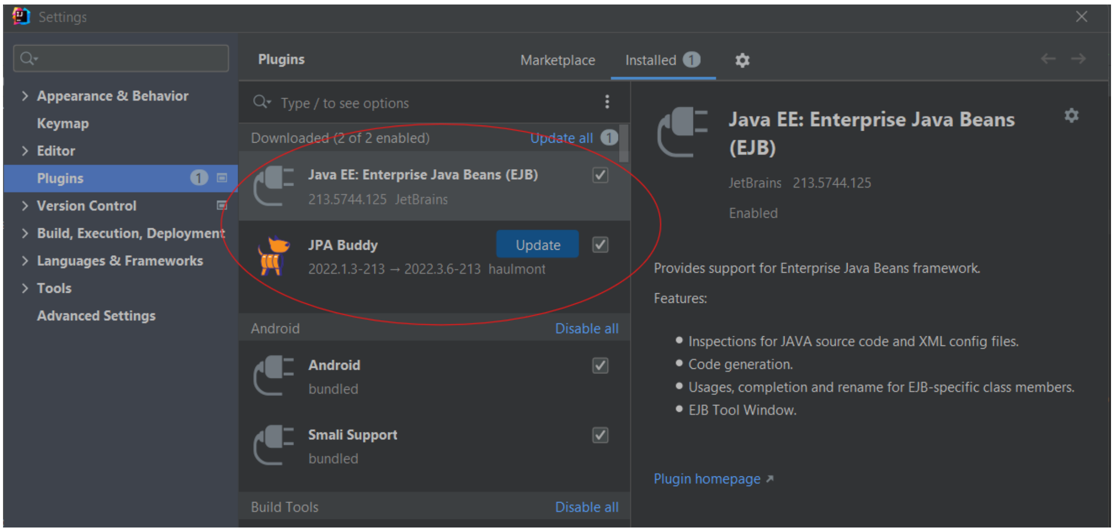
Hibernate

Framework for mapping an object-oriented domain model to a relational database.

Added dependencies:

- Servlet
- Hibernate
- Hibernate Validator

Figure 5.3.2.3: Screenshot of required implementations.



Settings

- Appearance & Behavior
- Keymap
- Editor
- Plugins 1
- Version Control
- Build, Execution, Deployment
- Languages & Frameworks
- Tools
- Advanced Settings

Plugins — Marketplace — Installed 1

Type / to see options

Downloaded (2 of 2 enabled) — Update all 1

- Java EE: Enterprise Java Beans (EJB) ☑ — 213.5744.125 JetBrains
- JPA Buddy — Update ☑ — 2022.1.3-213 → 2022.3.6-213 haulmont

Android — Disable all

- Android ☑ — bundled
- Smali Support ☑ — bundled

Build Tools — Disable all

Java EE: Enterprise Java Beans (EJB)

JetBrains 213.5744.125

Enabled

Provides support for Enterprise Java Beans framework.

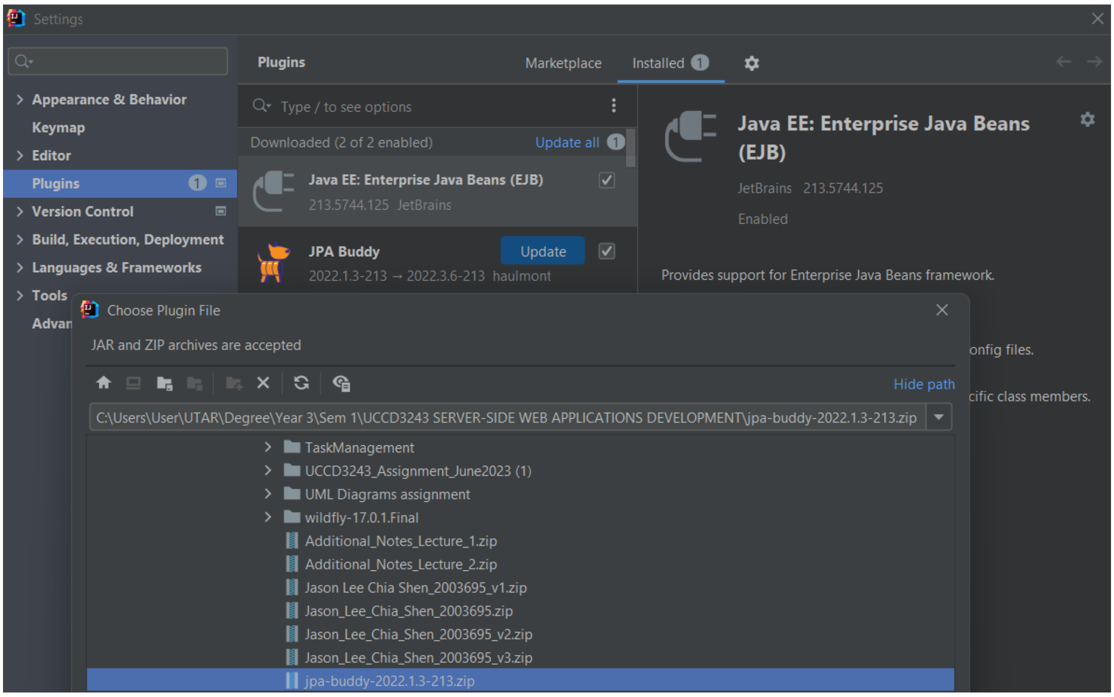
Features:

- Inspections for JAVA source code and XML config files.
- Code generation.
- Usages, completion and rename for EJB-specific class members.
- EJB Tool Window.

Plugin homepage ↗

Figure 5.3.2.4: Screenshot of required plugins to be downloaded.

The “Java EE: Enterprise Java Beas (EJB)” plugin can be found under File > Settings > Plugins.



Java EE: Enterprise Java Beans (EJB) ☑
213.5744.125 JetBrains

JPA Buddy ☑
2022.1.3-213 → 2022.3.6-213 haulmont

Java EE: Enterprise Java Beans (EJB)
JetBrains 213.5744.125
Enabled

Provides support for Enterprise Java Beans framework.

onfig files.

cific class members.

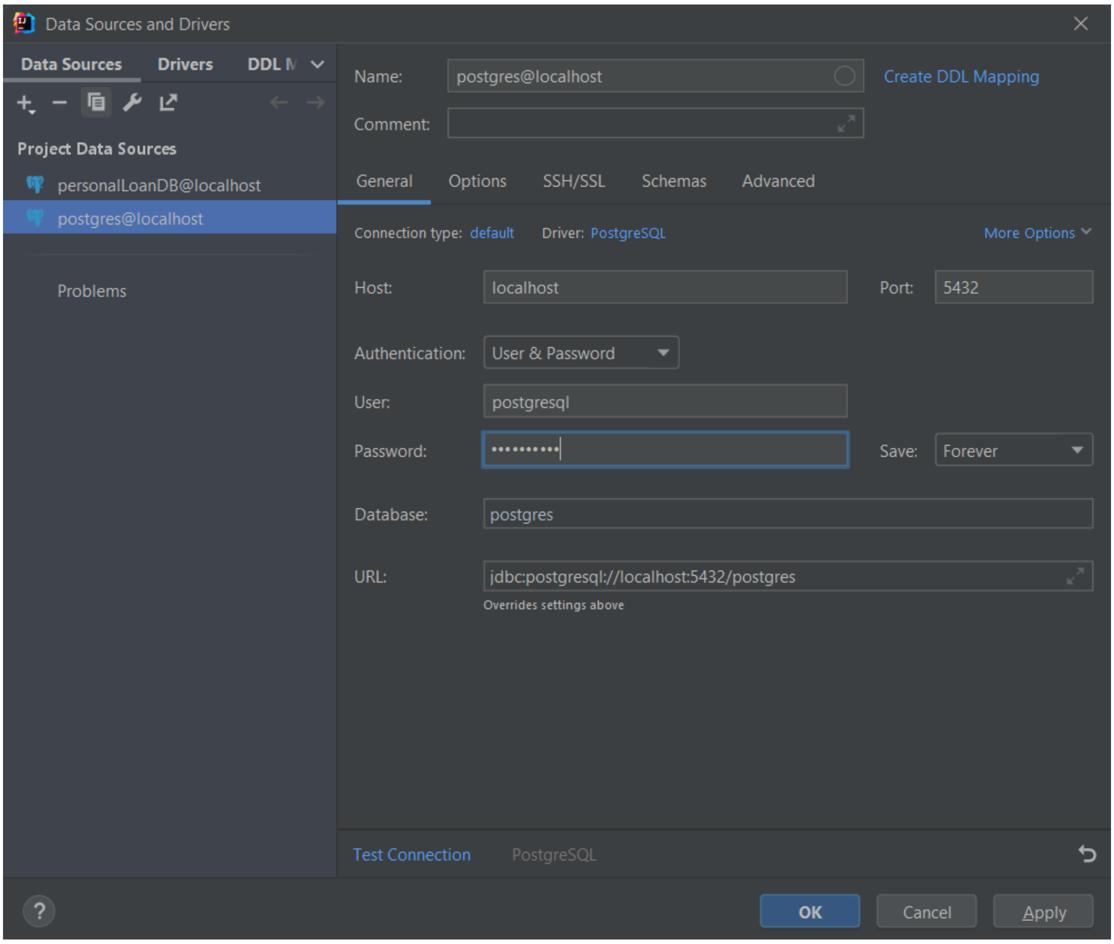
JAR and ZIP archives are accepted

C:\Users\User\UTAR\Degree\Year 3\Sem 1\UCCD3243 SERVER-SIDE WEB APPLICATIONS DEVELOPMENT\jpa-buddy-2022.1.3-213.zip

- TaskManagement
- UCCD3243_Assignment_June2023 (1)
- UML Diagrams assignment
- wildfly-17.0.1.Final
- Additional_Notes_Lecture_1.zip
- Additional_Notes_Lecture_2.zip
- Jason Lee Chia Shen_2003695_v1.zip
- Jason_Lee_Chia_Shen_2003695.zip
- Jason_Lee_Chia_Shen_2003695_v2.zip
- Jason_Lee_Chia_Shen_2003695_v3.zip
- jpa-buddy-2022.1.3-213.zip

Figure 5.3.2.5: Screenshot of the installation of JPA Buddy.

Due to the fact that this project used the older version of JPA Buddy, the manual installation had to be done by clicking on the setting icon beside the installed menu, and click on the “Install Plugin From Disk..” option. The directory path is where the installed package resides.



Name: postgres@localhost

Comment:

Connection type: default Driver: PostgreSQL

Host: localhost Port: 5432

Authentication: User & Password

User: postgresql

Password: •••••••••• Save: Forever

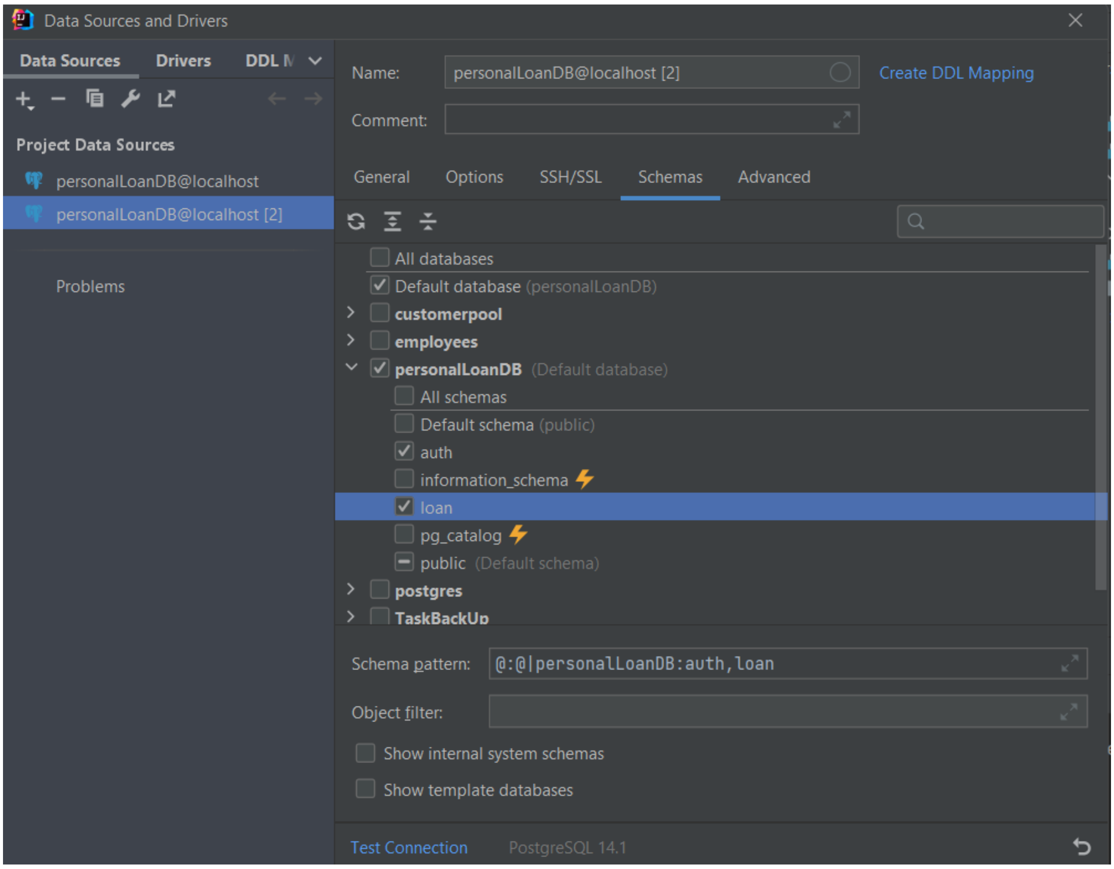
Database: postgres

URL: jdbc:postgresql://localhost:5432/postgres

Overrides settings above

Figure 5.3.2.6: Screenshot of the creation of data source mapping to the PostgreSQL database.

This data sources and drivers menu can be found under File > New > Data Source > PostgreSQL (do not choose PostgreSql which looks almost the same but last two letters are lowercase). The default port for PostgreSQL is 5432. The user and password are “postgres” and “postgresql” respectively. As for the URL, the final section would be the name of the database which would be used to generate entity classes.



Data Sources and Drivers

Name: personalLoanDB@localhost [2]

Comment:

- □ All databases
- ☑ Default database (personalLoanDB)
- □ customerpool
- □ employees
- ☑ personalLoanDB (Default database)
  - □ All schemas
  - □ Default schema (public)
  - ☑ auth
  - □ information_schema
  - ☑ loan
  - □ pg_catalog
  - ⊟ public (Default schema)
- □ postgres
- □ TaskBackUp

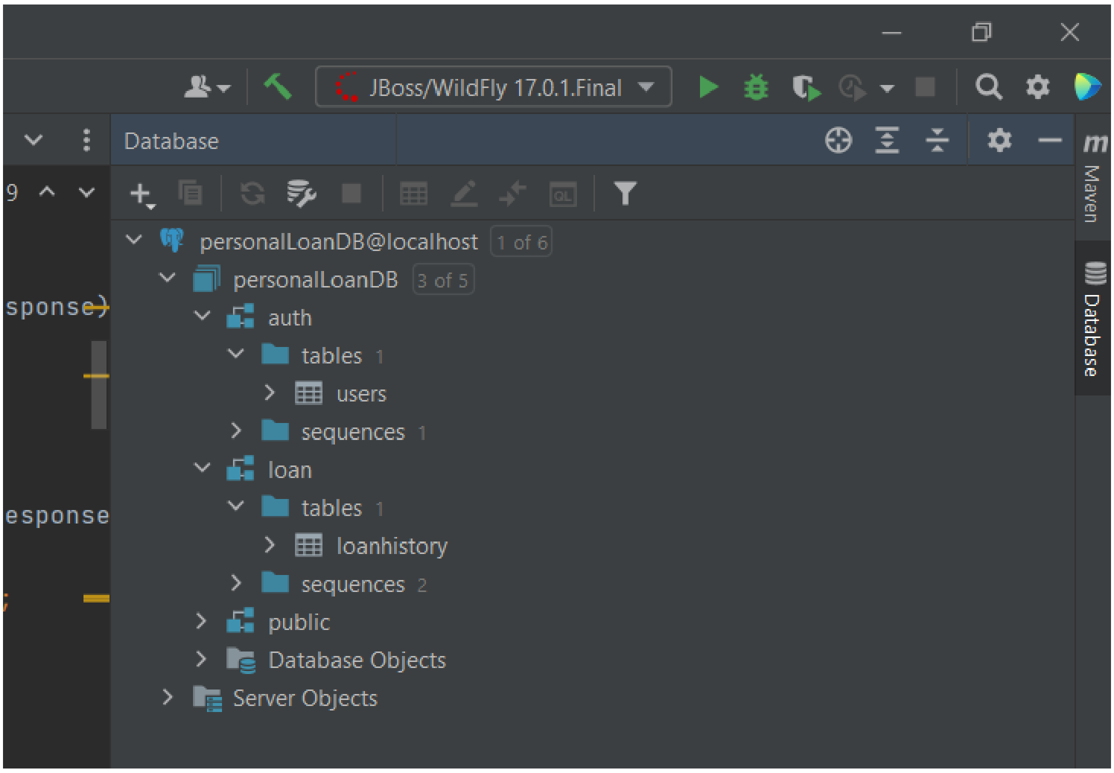
Schema pattern: `@:@|personalLoanDB:auth,loan`

Object filter:

□ Show internal system schemas

□ Show template databases

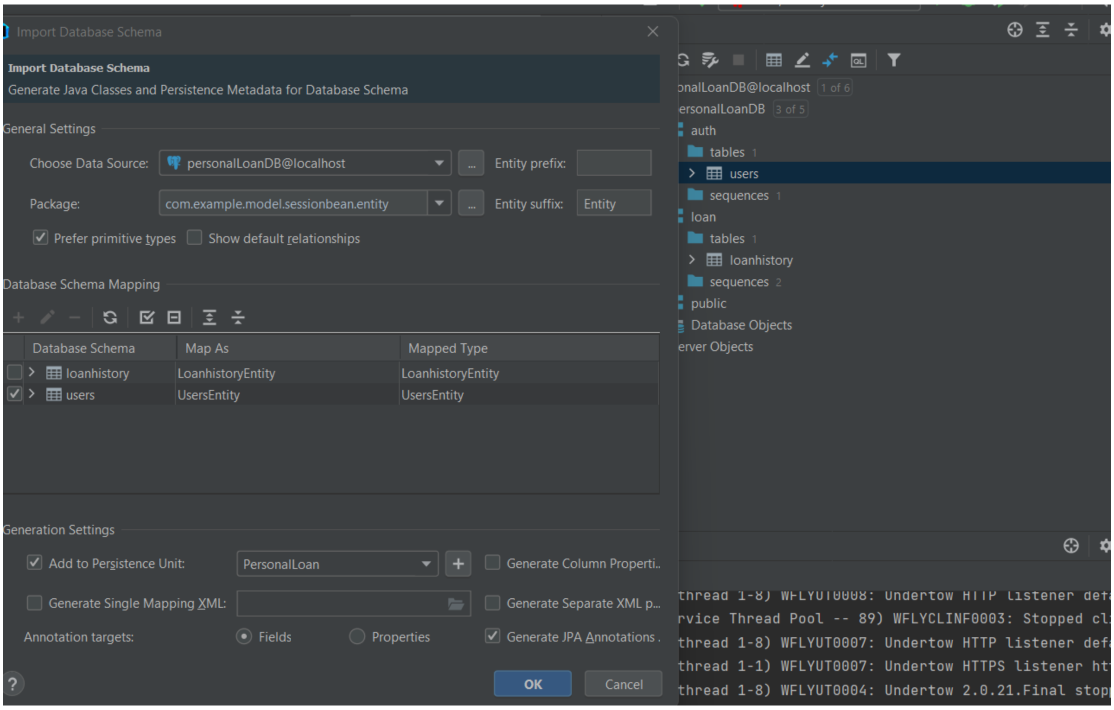
Figure 5.3.2.7: Screenshot of the included schemas.



```
personalLoanDB@localhost  1 of 6
  personalLoanDB  3 of 5
    auth
      tables  1
        users
      sequences  1
    loan
      tables  1
        loanhistory
      sequences  2
    public
    Database Objects
  Server Objects
```

Figure 5.3.2.8: Screenshot of the connected database and the scheme included.

By clicking on the database tab on the right, the connected database was shown along with its schemas and tables. Both users and loanhistory tables were needed for generating persistence mapping to its entity classes.



Import Database Schema

Import Database Schema

Generate Java Classes and Persistence Metadata for Database Schema

General Settings

Choose Data Source: personalLoanDB@localhost

Entity prefix:

Package: com.example.model.sessionbean.entity

Entity suffix: Entity

☑ Prefer primitive types ☐ Show default relationships

Database Schema Mapping

	Database Schema	Map As	Mapped Type
☐	loanhistory	LoanhistoryEntity	LoanhistoryEntity
☑	users	UsersEntity	UsersEntity

Generation Settings

☑ Add to Persistence Unit: PersonalLoan

☐ Generate Column Properti..

☐ Generate Single Mapping XML:

☐ Generate Separate XML p...

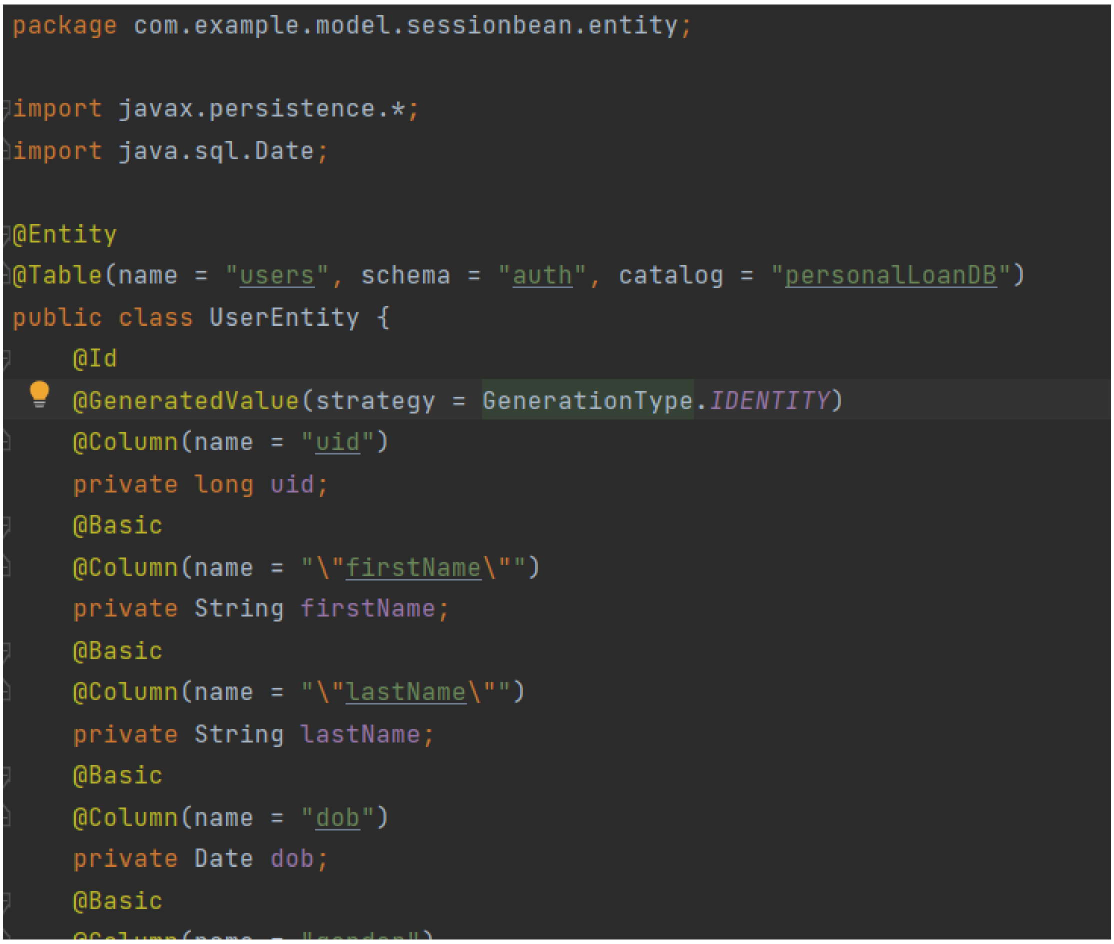
Annotation targets: ● Fields ○ Properties

☑ Generate JPA Annotations .

```
thread 1-8) WFLYUT0008: Undertow HTTP listener def
rvice Thread Pool -- 89) WFLYCLINF0003: Stopped cl
thread 1-8) WFLYUT0007: Undertow HTTP listener def
thread 1-1) WFLYUT0007: Undertow HTTPS listener ht
thread 1-8) WFLYUT0004: Undertow 2.0.21.Final stop
```

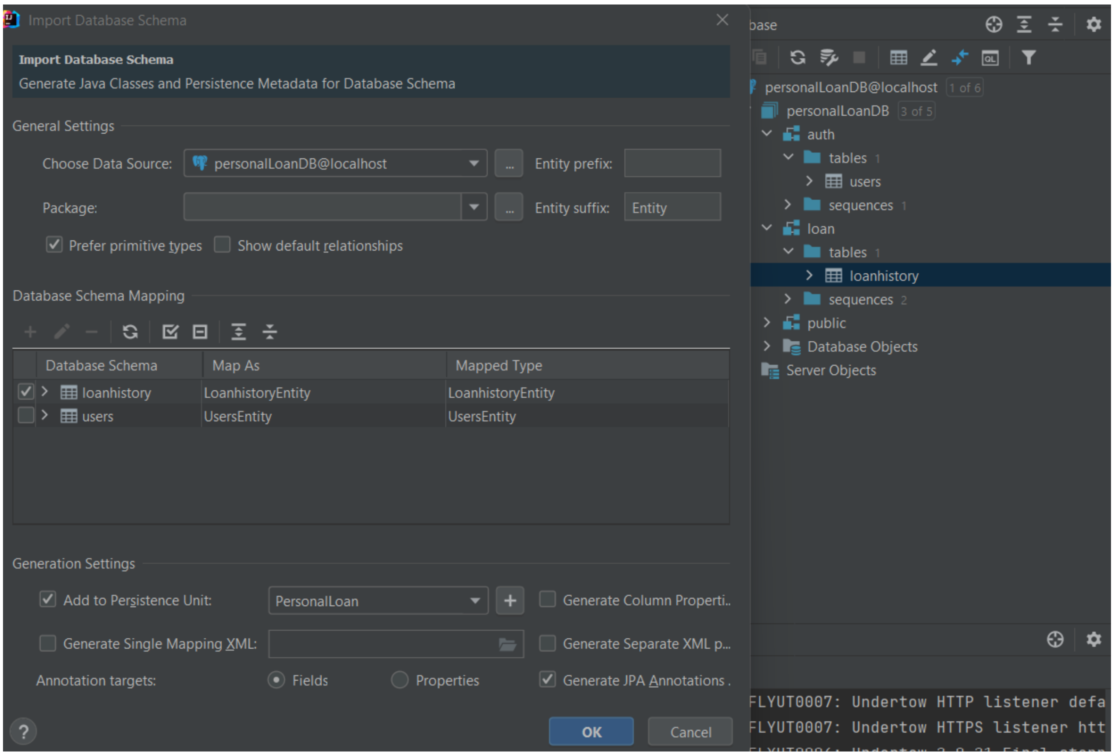
Figure 5.3.2.9: Screenshot of the generation of persistence mapping for users table.

Click on the ‘generate persistence mapping’ when right clicking on the users table. In the dialog box, the package was linked to the directory where this entity class would be residing. In the MVC Framework, it is necessary to create a new package under the session bean package to store the generated entity classes.



```
package com.example.model.sessionbean.entity;

import javax.persistence.*;
import java.sql.Date;

@Entity
@Table(name = "users", schema = "auth", catalog = "personalLoanDB")
public class UserEntity {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "uid")
    private long uid;
    @Basic
    @Column(name = "\"firstName\"")
    private String firstName;
    @Basic
    @Column(name = "\"lastName\"")
    private String lastName;
    @Basic
    @Column(name = "dob")
    private Date dob;
    @Basic
    @Column(name = "gender")
```

Figure 5.3.2.10: Screenshot of the UserEntity class.



Import Database Schema

Import Database Schema

Generate Java Classes and Persistence Metadata for Database Schema

General Settings

Choose Data Source: personalLoanDB@localhost

Entity prefix:

Package:

Entity suffix: Entity

☑ Prefer primitive types ☐ Show default relationships

Database Schema Mapping

	Database Schema	Map As	Mapped Type
☑	loanhistory	LoanhistoryEntity	LoanhistoryEntity
☐	users	UsersEntity	UsersEntity

Generation Settings

☑ Add to Persistence Unit: PersonalLoan

☐ Generate Column Properti..

☐ Generate Single Mapping XML:

☐ Generate Separate XML p...

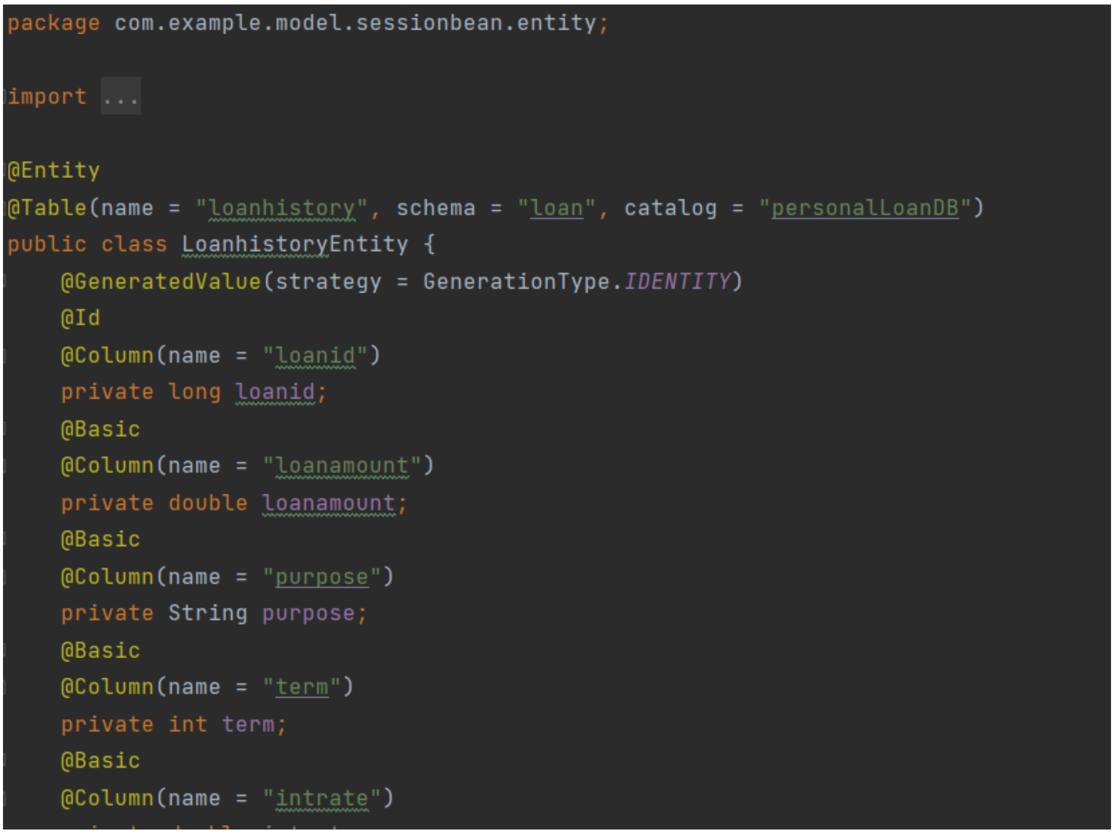
Annotation targets: ● Fields ○ Properties

☑ Generate JPA Annotations .

```
FLYUT0007: Undertow HTTP listener defa
FLYUT0007: Undertow HTTPS listener htt
```

Figure 5.3.2.11: Screenshot of the generation of persistence mapping for loanhistory table.

The configuration step is same as to generating persistence mapping as shown in Figure 5.3.2.9.



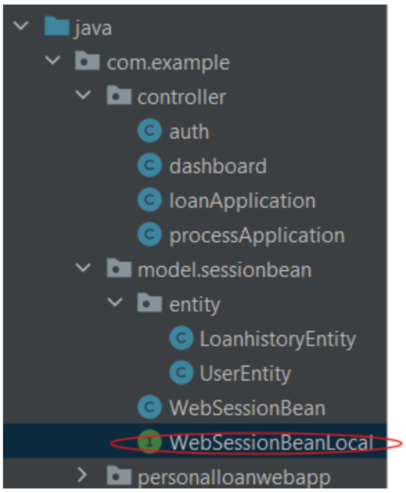
```
package com.example.model.sessionbean.entity;

import ...

@Entity
@Table(name = "loanhistory", schema = "loan", catalog = "personalLoanDB")
public class LoanhistoryEntity {
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Id
    @Column(name = "loanid")
    private long loanid;
    @Basic
    @Column(name = "loanamount")
    private double loanamount;
    @Basic
    @Column(name = "purpose")
    private String purpose;
    @Basic
    @Column(name = "term")
    private int term;
    @Basic
    @Column(name = "intrate")
```

Figure 5.3.2.12: Screenshot of the LoanhistoryEntity class.

5.3.3 Session Bean

The session bean is necessary to allow CRUD operations to be performed on the connected database. The following steps depict the creation of session bean:

1.

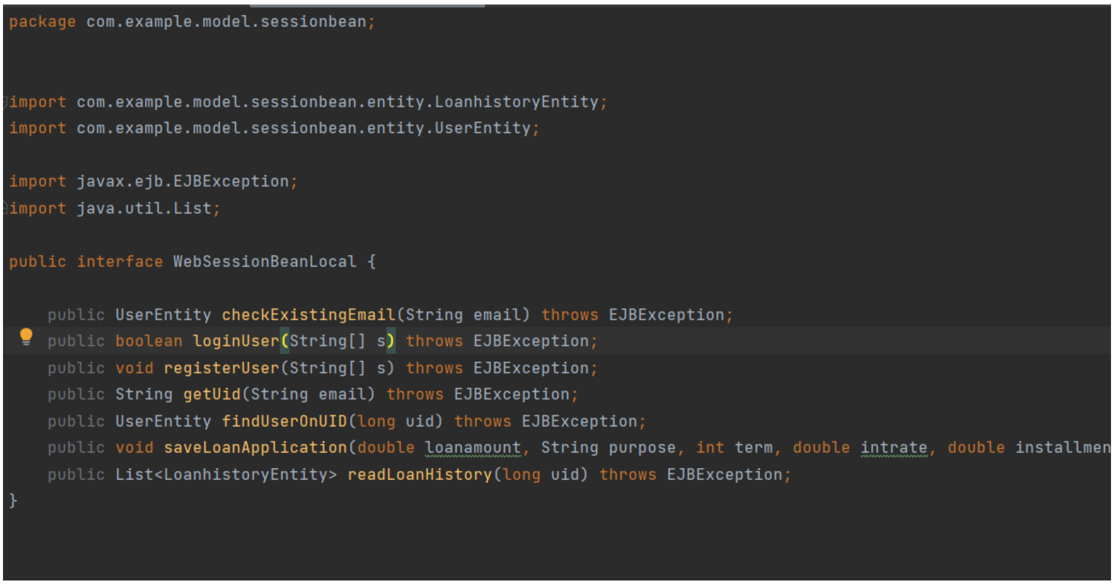


```
java
  com.example
    controller
      auth
      dashboard
      loanApplication
      processApplication
    model.sessionbean
      entity
        LoanhistoryEntity
        UserEntity
      WebSessionBean
      WebSessionBeanLocal
    personalloanwebapp
```

Figure 5.3.3.1: Screenshot of the WebSessionBeanLocal package.

An interface named "WebSessionBeanLocal" was created under the "model.sessionbean" pacakage. Another java class named "WebSessionBean" was created and implemented the functions of the interface "WebSessoinBeanLocal".

2.

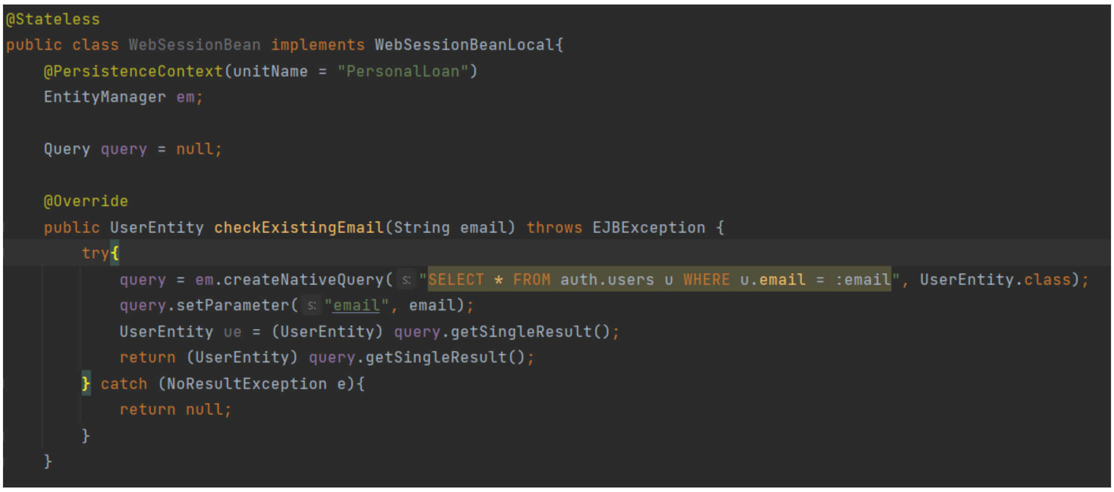


```java
package com.example.model.sessionbean;


import com.example.model.sessionbean.entity.LoanhistoryEntity;
import com.example.model.sessionbean.entity.UserEntity;

import javax.ejb.EJBException;
import java.util.List;

public interface WebSessionBeanLocal {

    public UserEntity checkExistingEmail(String email) throws EJBException;
    public boolean loginUser(String[] s) throws EJBException;
    public void registerUser(String[] s) throws EJBException;
    public String getUid(String email) throws EJBException;
    public UserEntity findUserOnUID(long uid) throws EJBException;
    public void saveLoanApplication(double loanamount, String purpose, int term, double intrate, double installmen
    public List<LoanhistoryEntity> readLoanHistory(long uid) throws EJBException;
}
```

Figure 5.3.3.2: Screenshot of the content of WebSessionBeanLocal.

A total of seven functions were defined to perform CRUD operations on the database. The first function was the checkExistingEmail(). This function will check for matching email in the database based on the provided email address in the parameter. If it is true, then it will return the user who has the matching email address or else it will return null. This function was used for registration where it did not allow users to register an account with the existing email address. The second function was loginUser(). This function will authenticate users based on the provided login credentials. Moreover, the third function was registerUser(). A new account will be registered to the database when this function is called. The fourth function was the getUid(). It will return the user id based on the provided email address. The fifth function was the findUserBasedOnUID(). The information of the user will be returned based on the provided user id parameter. The sixth function is the saveLoanApplication(). Any loan application that has been applied will be automatically saved to the database. The final function is the readLoanHistory(). It will return a list of loan records based on the provided user id.

3.

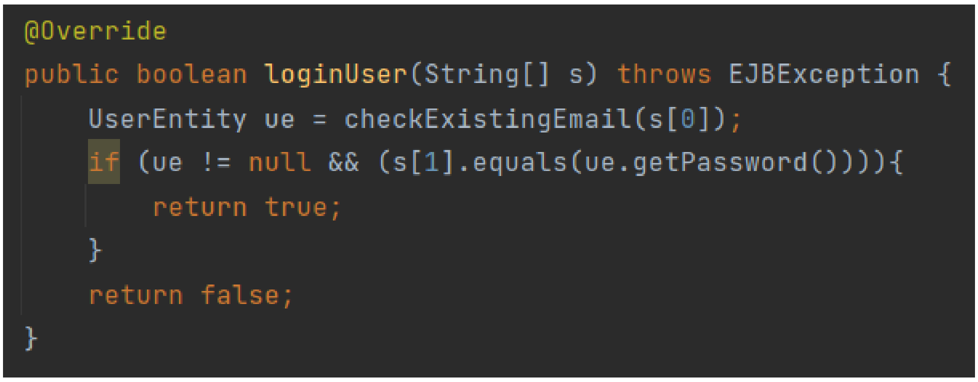


```java
@Stateless
public class WebSessionBean implements WebSessionBeanLocal{
    @PersistenceContext(unitName = "PersonalLoan")
    EntityManager em;

    Query query = null;

    @Override
    public UserEntity checkExistingEmail(String email) throws EJBException {
        try{
            query = em.createNativeQuery( s: "SELECT * FROM auth.users u WHERE u.email = :email", UserEntity.class);
            query.setParameter( s: "email", email);
            UserEntity ue = (UserEntity) query.getSingleResult();
            return (UserEntity) query.getSingleResult();
        } catch (NoResultException e){
            return null;
        }
    }
```

Figure 5.3.3.3: Code snippet of the checkExistingEmail function.

Once the “WebSessionBeanLocal” interface had been created, all the functions would be implemented in the “WebSessionBean” java class. In Figure 5.3.3.3, a query had been created to retrieve the information from the table users under the auth schema with the condition of matching email address. When there is a record, the function will return the user information wrapped in the UserEntity class back to the calling servlet.

4.

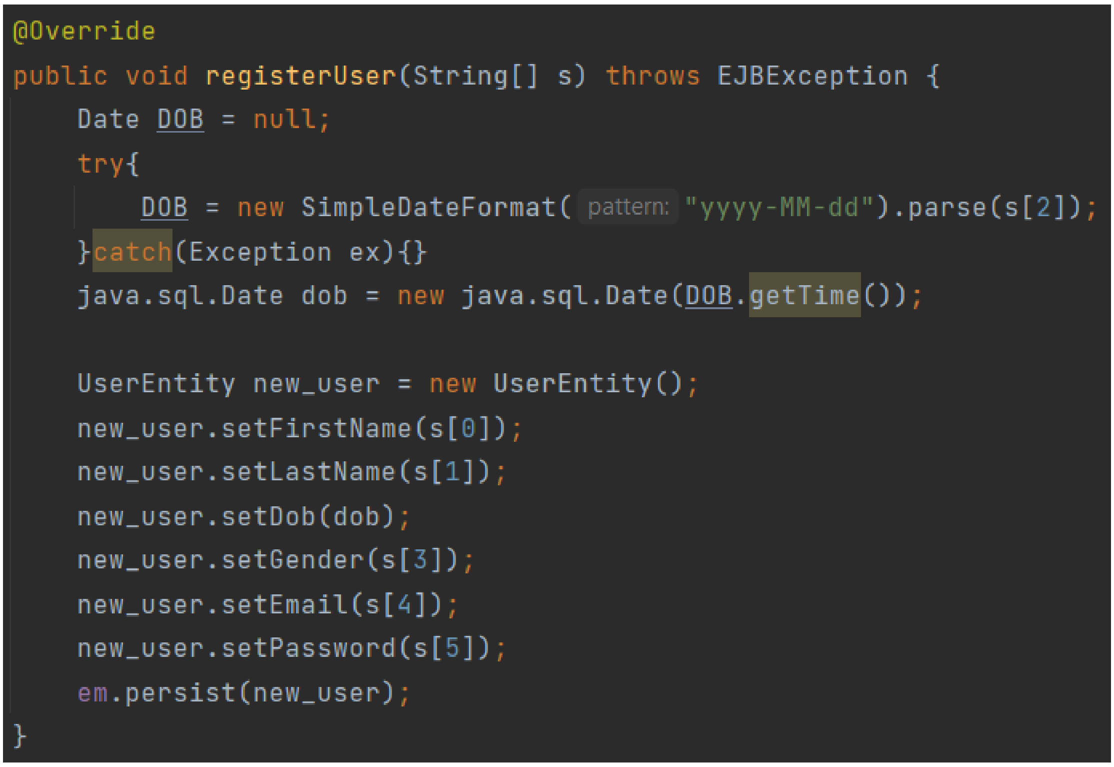


```
@Override
public boolean loginUser(String[] s) throws EJBException {
    UserEntity ue = checkExistingEmail(s[0]);
    if (ue != null && (s[1].equals(ue.getPassword()))){
        return true;
    }
    return false;
}
```

Figure 5.3.3.4: Code snippet of the loginUser function.

Upon getting the user from the checkExistingEmail function, the user’s password was then compared with the provided password in the parameter. If the password was valid, then this function will return true and authenticate the user.

5.

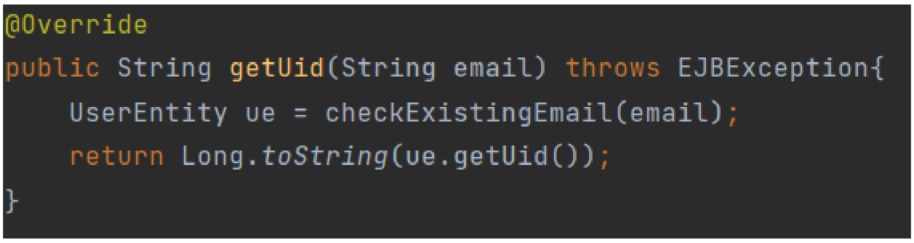


```
@Override
public void registerUser(String[] s) throws EJBException {
    Date DOB = null;
    try{
        DOB = new SimpleDateFormat( pattern: "yyyy-MM-dd").parse(s[2]);
    }catch(Exception ex){}
    java.sql.Date dob = new java.sql.Date(DOB.getTime());

    UserEntity new_user = new UserEntity();
    new_user.setFirstName(s[0]);
    new_user.setLastName(s[1]);
    new_user.setDob(dob);
    new_user.setGender(s[3]);
    new_user.setEmail(s[4]);
    new_user.setPassword(s[5]);
    em.persist(new_user);
}
```

Figure 5.3.3.5: Code snippet of the registerUser function.

This function will register the provided user from the parameter into the database by using the persist function from the EntityManager.

6.

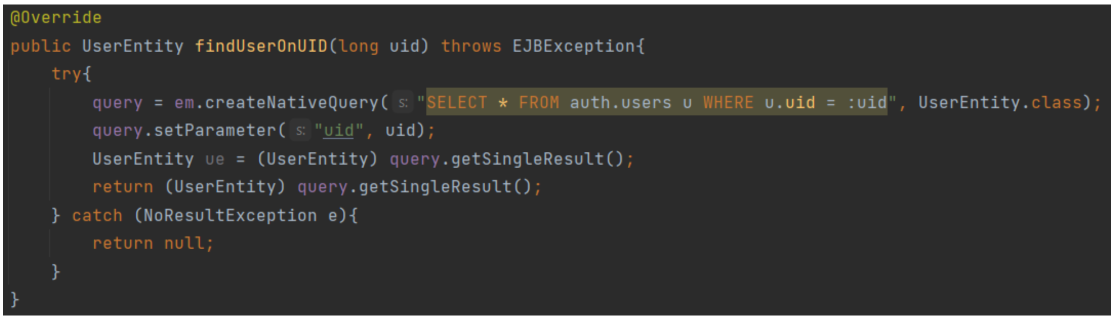


```
@Override
public String getUid(String email) throws EJBException{
    UserEntity ue = checkExistingEmail(email);
    return Long.toString(ue.getUid());
}
```

Figure 5.3.3.6: Code snippet of the getUid function.

This function will return the user id of the retrieved user from the checkExistingEmail function based on the email provided in the parameter.

7.

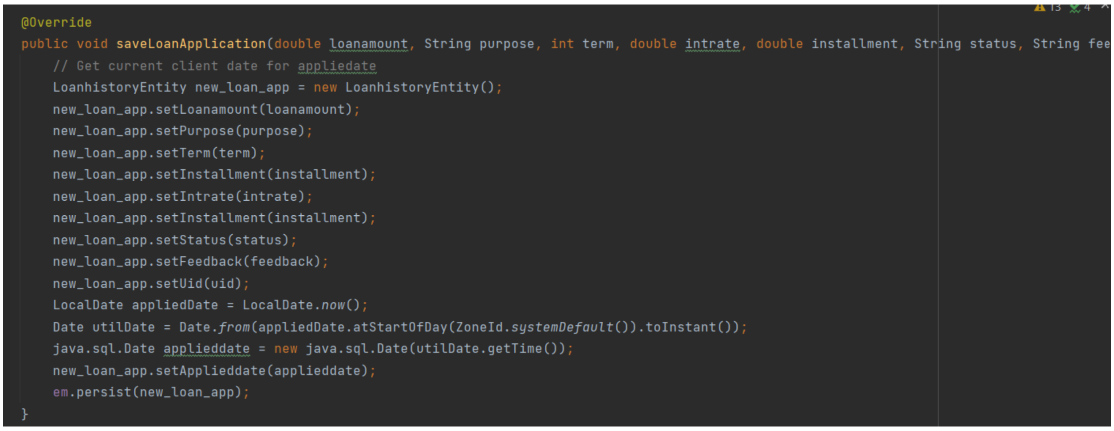


```
@Override
public UserEntity findUserOnUID(long uid) throws EJBException{
    try{
        query = em.createNativeQuery( s: "SELECT * FROM auth.users u WHERE u.uid = :uid", UserEntity.class);
        query.setParameter( s: "uid", uid);
        UserEntity ue = (UserEntity) query.getSingleResult();
        return (UserEntity) query.getSingleResult();
    } catch (NoResultException e){
        return null;
    }
}
```

Figure 5.3.3.7: Code snippet of the findUserOnUID function.

A query was created to retrieve the user information based on the user id provided in the parameter.

8.

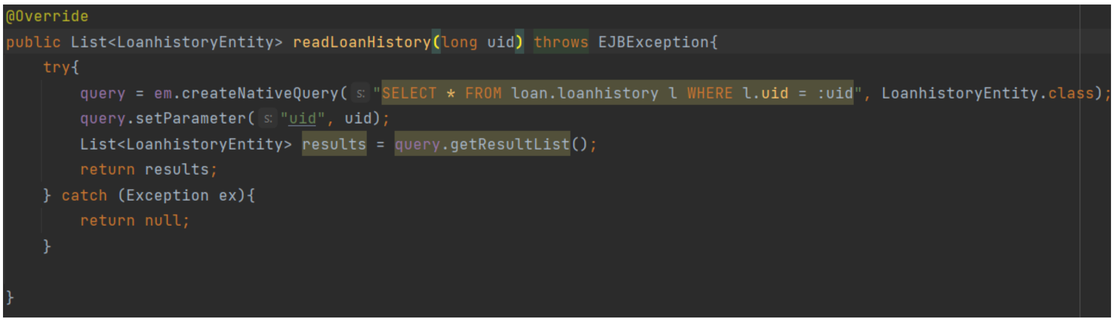


```
@Override
public void saveLoanApplication(double loanamount, String purpose, int term, double intrate, double installment, String status, String fee
    // Get current client date for appliedate
    LoanhistoryEntity new_loan_app = new LoanhistoryEntity();
    new_loan_app.setLoanamount(loanamount);
    new_loan_app.setPurpose(purpose);
    new_loan_app.setTerm(term);
    new_loan_app.setInstallment(installment);
    new_loan_app.setIntrate(intrate);
    new_loan_app.setInstallment(installment);
    new_loan_app.setStatus(status);
    new_loan_app.setFeedback(feedback);
    new_loan_app.setUid(uid);
    LocalDate appliedDate = LocalDate.now();
    Date utilDate = Date.from(appliedDate.atStartOfDay(ZoneId.systemDefault()).toInstant());
    java.sql.Date applieddate = new java.sql.Date(utilDate.getTime());
    new_loan_app.setApplieddate(applieddate);
    em.persist(new_loan_app);
}
```

Figure 5.3.3.8: Code snippet of the saveLoanApplication function.

The saveLoanApplication function will persist the provided information of the valid loan application to the database.

9.



```
@Override
public List<LoanhistoryEntity> readLoanHistory(long uid) throws EJBException{
    try{
        query = em.createNativeQuery( "SELECT * FROM loan.loanhistory l WHERE l.uid = :uid", LoanhistoryEntity.class);
        query.setParameter( "uid", uid);
        List<LoanhistoryEntity> results = query.getResultList();
        return results;
    } catch (Exception ex){
        return null;
    }

}
```

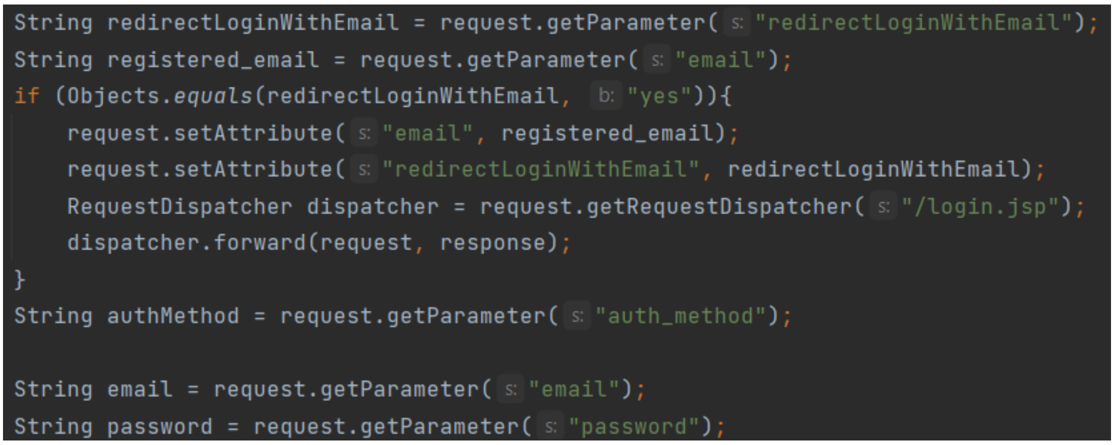
Figure 5.3.3.9: Code snippet of the readLoanHistory function.

This function will return a list of LoanhistoryEntity based on the user id provided in the parameter. A query was defined to retrieve any loan application that was applied by the user from the loanhistory table under the loan schema.

5.3.4 Login and Registration Module

The login and registration modules are necessary to prevent any unauthorized user from accessing the personal loan web application. A new user will need to register for an account first. Once registered, the user will need to use that account to sign in into the web application. The loan application and the dashboard pages are only accessible by the authorized user. The flow from registration to login process are as follows:

1.

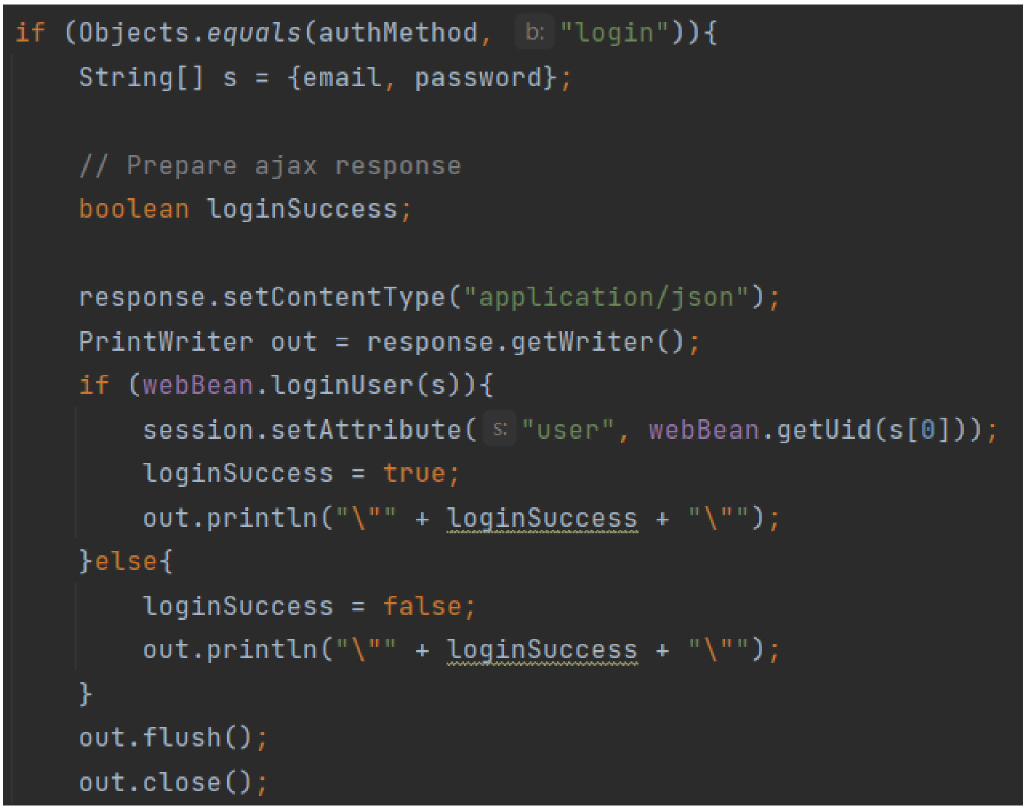


```
String redirectLoginWithEmail = request.getParameter( s: "redirectLoginWithEmail");
String registered_email = request.getParameter( s: "email");
if (Objects.equals(redirectLoginWithEmail,  b: "yes")){
    request.setAttribute( s: "email", registered_email);
    request.setAttribute( s: "redirectLoginWithEmail", redirectLoginWithEmail);
    RequestDispatcher dispatcher = request.getRequestDispatcher( s: "/login.jsp");
    dispatcher.forward(request, response);
}
String authMethod = request.getParameter( s: "auth_method");

String email = request.getParameter( s: "email");
String password = request.getParameter( s: "password");
```

Figure 5.3.4.1: Code snippet of the auth servlet (part 1).

The auth servlet is responsible for both login and registration operations. When the request is redirected to this servlet, several variables will be retrieved from the request parameters. The most crucial variable is the “authMethod” which will determine whether the request is either from the login page or the registration page. The “redirectLoginWithEmail” variable will redirect the user to the login page after they have done registering and their email address will be automatically entered in the email address input field. This is to save the user time from having to re-enter their email address.

2.

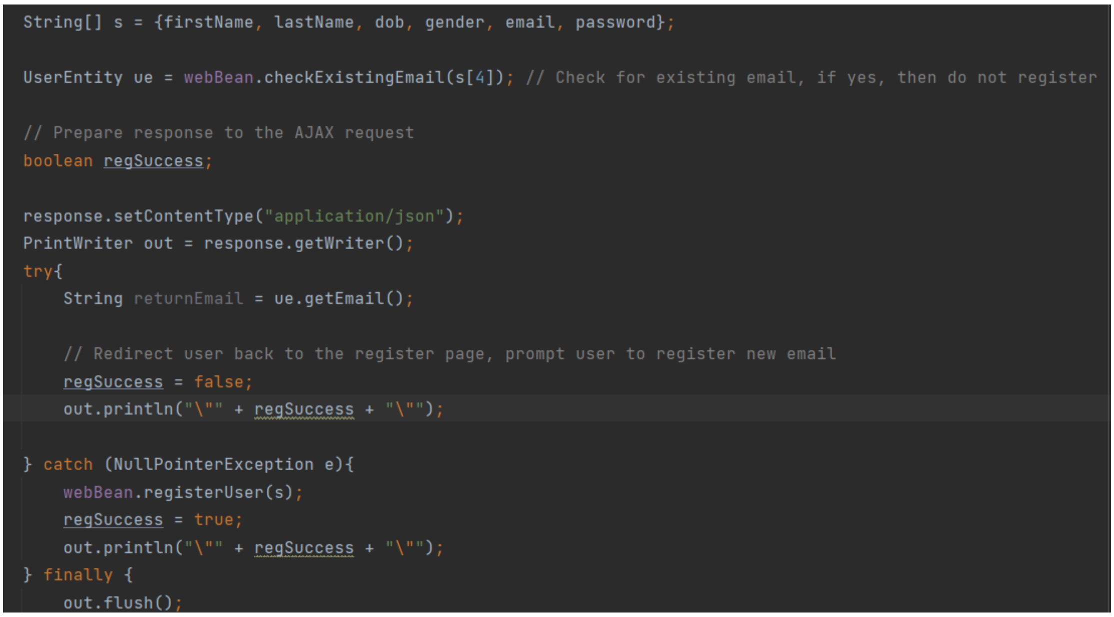


```
if (Objects.equals(authMethod, b: "login")){
    String[] s = {email, password};

    // Prepare ajax response
    boolean loginSuccess;

    response.setContentType("application/json");
    PrintWriter out = response.getWriter();
    if (webBean.loginUser(s)){
        session.setAttribute( s: "user", webBean.getUid(s[0]));
        loginSuccess = true;
        out.println("\"" + loginSuccess + "\"");
    }else{
        loginSuccess = false;
        out.println("\"" + loginSuccess + "\"");
    }
    out.flush();
    out.close();
```

Figure 5.3.4.2: Code snippet of the auth servlet (part 2).

If the “authMethod” is from the login page, then the auth servlet will know that the request should be processed in the respective login processing code. If the login credentials such as email and password are valid, the user will be authorized, and a session will be created with their user id. If not, the user will remain in the same login page and will be prompted to re-enter their login credentials.

3.



```
String[] s = {firstName, lastName, dob, gender, email, password};

UserEntity ue = webBean.checkExistingEmail(s[4]); // Check for existing email, if yes, then do not register

// Prepare response to the AJAX request
boolean regSuccess;

response.setContentType("application/json");
PrintWriter out = response.getWriter();
try{
    String returnEmail = ue.getEmail();

    // Redirect user back to the register page, prompt user to register new email
    regSuccess = false;
    out.println("\"" + regSuccess + "\"");

} catch (NullPointerException e){
    webBean.registerUser(s);
    regSuccess = true;
    out.println("\"" + regSuccess + "\"");
} finally {
    out.flush();
```

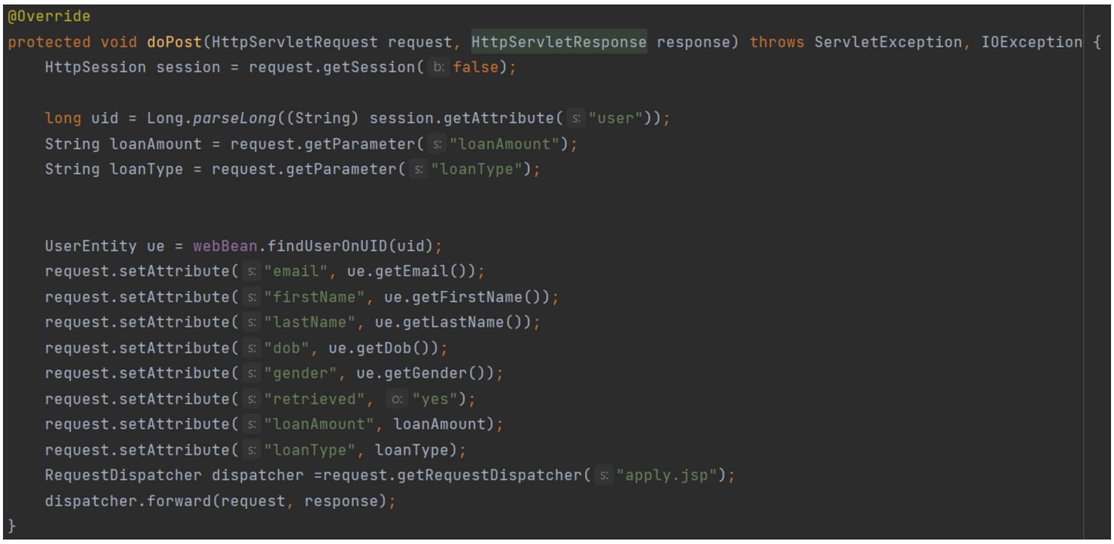
Figure 5.3.4.3: Code snippet of the auth servlet (part 3).

If the request is sent from the registration page, then the auth servlet will process the request parameters in the registration block. The servlet will first send the provided email address to the “WebSessionBeanLocal” to check for any existing email address. If the email address has already been registered by the other user in the database, then the auth servlet will prompt the user to re-enter another email address. If the email address does not exist, then the user will be registered into the database using the registerUser function from the session bean.

5.3.5 Data collection and Application Module

Two servlets that are responsible for both of these modules are the loanApplication servlet and also the processApplication servlet. The loanApplication servlet will receive request parameters and then dispatch it to the processApplication servlet for further processing. The flow from loanApplication to the processApplication are depicted as follows:

1.

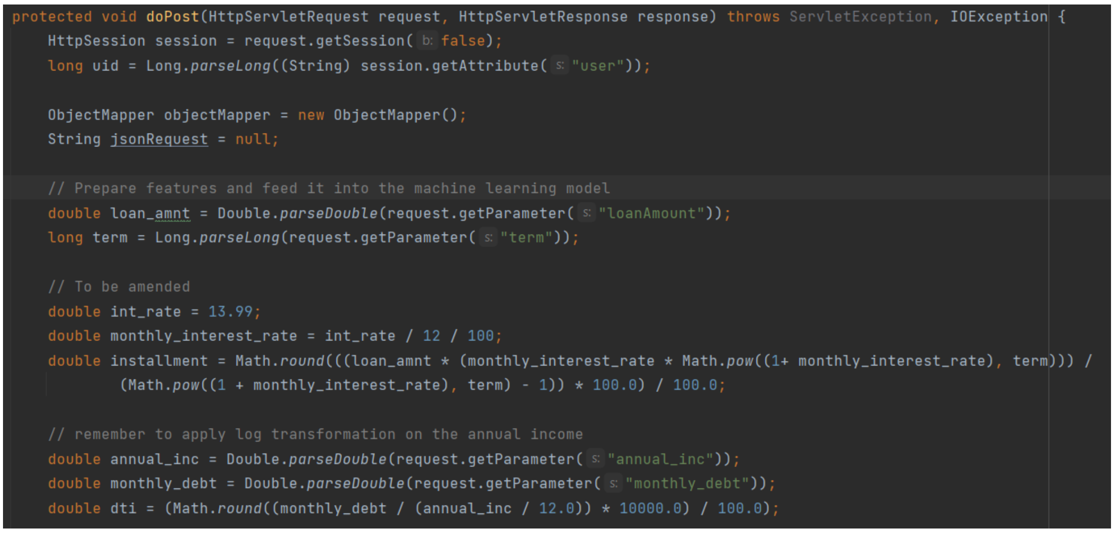


```java
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    HttpSession session = request.getSession( b: false);

    long uid = Long.parseLong((String) session.getAttribute( s: "user"));
    String loanAmount = request.getParameter( s: "loanAmount");
    String loanType = request.getParameter( s: "loanType");


    UserEntity ue = webBean.findUserOnUID(uid);
    request.setAttribute( s: "email", ue.getEmail());
    request.setAttribute( s: "firstName", ue.getFirstName());
    request.setAttribute( s: "lastName", ue.getLastName());
    request.setAttribute( s: "dob", ue.getDob());
    request.setAttribute( s: "gender", ue.getGender());
    request.setAttribute( s: "retrieved",  o: "yes");
    request.setAttribute( s: "loanAmount", loanAmount);
    request.setAttribute( s: "loanType", loanType);
    RequestDispatcher dispatcher =request.getRequestDispatcher( s: "apply.jsp");
    dispatcher.forward(request, response);
}
```

Figure 5.3.5.1: Code snippet of the loginApplication servlet.

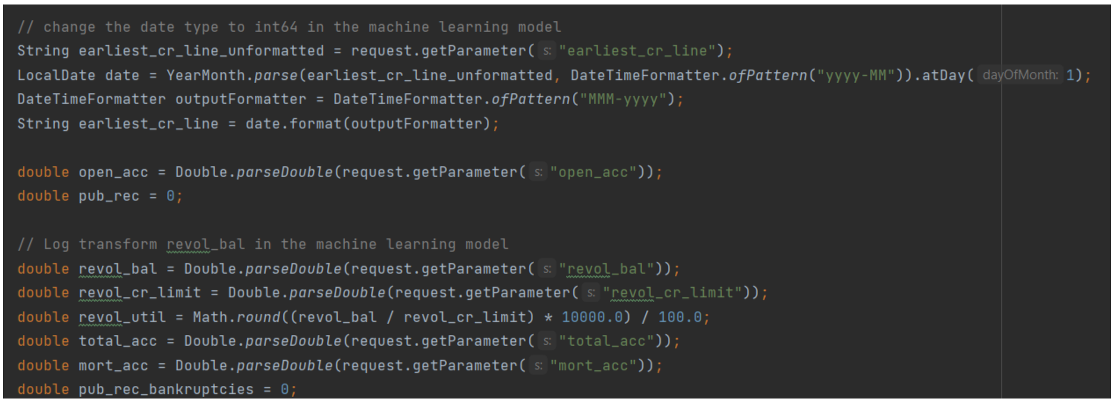
This servlet will handle all the requests coming from the index page where the user will fill in two basic information of the application such as the loan amount and the type of loan they wish to apply. Then, these attributes will be dispatched to the application page.

2.



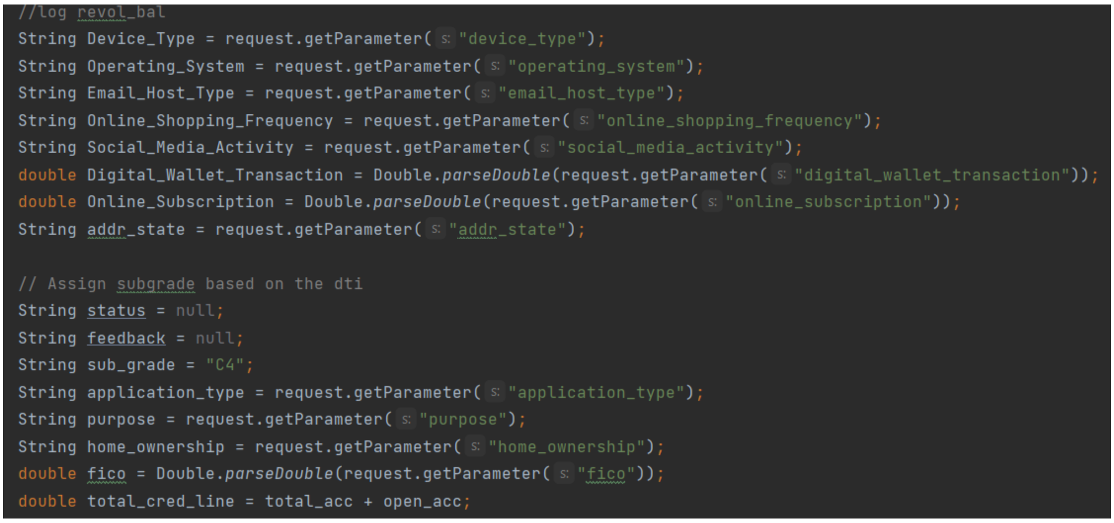
```java
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    HttpSession session = request.getSession( b: false);
    long uid = Long.parseLong((String) session.getAttribute( s: "user"));

    ObjectMapper objectMapper = new ObjectMapper();
    String jsonRequest = null;

    // Prepare features and feed it into the machine learning model
    double loan_amnt = Double.parseDouble(request.getParameter( s: "loanAmount"));
    long term = Long.parseLong(request.getParameter( s: "term"));

    // To be amended
    double int_rate = 13.99;
    double monthly_interest_rate = int_rate / 12 / 100;
    double installment = Math.round(((loan_amnt * (monthly_interest_rate * Math.pow((1+ monthly_interest_rate), term))) /
            (Math.pow((1 + monthly_interest_rate), term) - 1)) * 100.0) / 100.0;

    // remember to apply log transformation on the annual income
    double annual_inc = Double.parseDouble(request.getParameter( s: "annual_inc"));
    double monthly_debt = Double.parseDouble(request.getParameter( s: "monthly_debt"));
    double dti = (Math.round((monthly_debt / (annual_inc / 12.0)) * 10000.0) / 100.0);
```

Figure 5.3.5.2: Code snippet of the processApplication servlet (part 1).



```
// change the date type to int64 in the machine learning model
String earliest_cr_line_unformatted = request.getParameter( s: "earliest_cr_line");
LocalDate date = YearMonth.parse(earliest_cr_line_unformatted, DateTimeFormatter.ofPattern("yyyy-MM")).atDay( dayOfMonth: 1);
DateTimeFormatter outputFormatter = DateTimeFormatter.ofPattern("MMM-yyyy");
String earliest_cr_line = date.format(outputFormatter);

double open_acc = Double.parseDouble(request.getParameter( s: "open_acc"));
double pub_rec = 0;

// Log transform revol_bal in the machine learning model
double revol_bal = Double.parseDouble(request.getParameter( s: "revol_bal"));
double revol_cr_limit = Double.parseDouble(request.getParameter( s: "revol_cr_limit"));
double revol_util = Math.round((revol_bal / revol_cr_limit) * 10000.0) / 100.0;
double total_acc = Double.parseDouble(request.getParameter( s: "total_acc"));
double mort_acc = Double.parseDouble(request.getParameter( s: "mort_acc"));
double pub_rec_bankruptcies = 0;
```

Figure 5.3.5.3: Code snippet of the processApplication servlet (part 2).

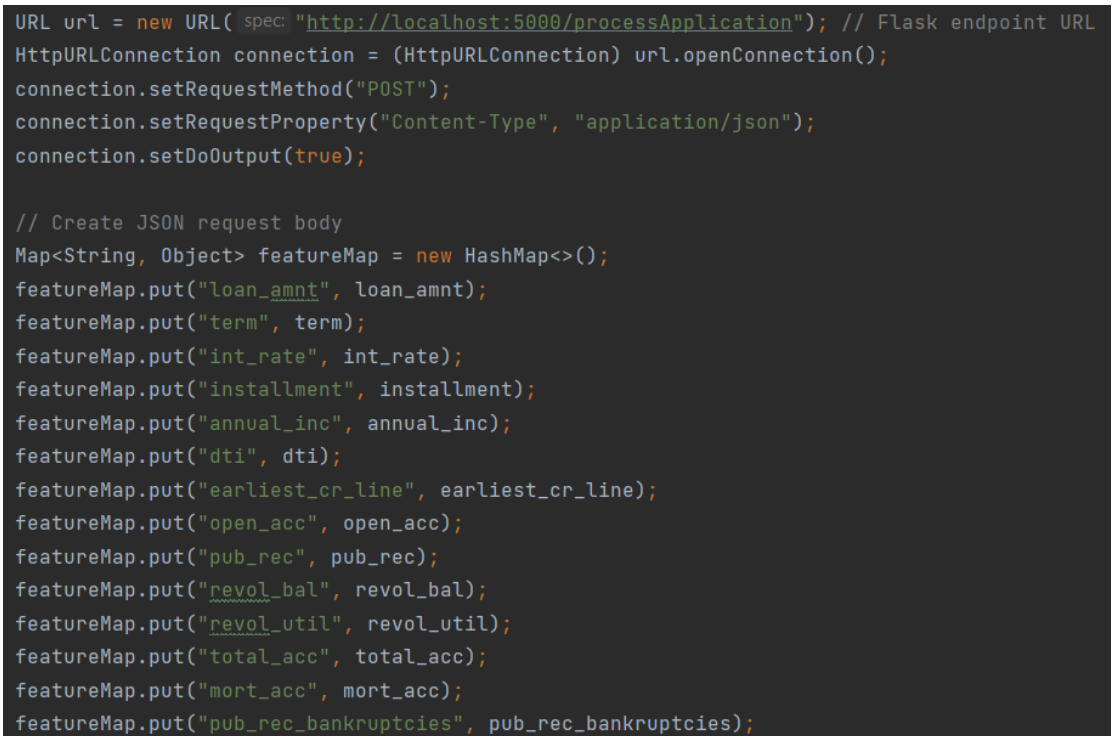


```
//log revol_bal
String Device_Type = request.getParameter( s: "device_type");
String Operating_System = request.getParameter( s: "operating_system");
String Email_Host_Type = request.getParameter( s: "email_host_type");
String Online_Shopping_Frequency = request.getParameter( s: "online_shopping_frequency");
String Social_Media_Activity = request.getParameter( s: "social_media_activity");
double Digital_Wallet_Transaction = Double.parseDouble(request.getParameter( s: "digital_wallet_transaction"));
double Online_Subscription = Double.parseDouble(request.getParameter( s: "online_subscription"));
String addr_state = request.getParameter( s: "addr_state");

// Assign subgrade based on the dti
String status = null;
String feedback = null;
String sub_grade = "C4";
String application_type = request.getParameter( s: "application_type");
String purpose = request.getParameter( s: "purpose");
String home_ownership = request.getParameter( s: "home_ownership");
double fico = Double.parseDouble(request.getParameter( s: "fico"));
double total_cred_line = total_acc + open_acc;
```

Figure 5.3.5.4: Code snippet of the processApplication servlet (part 3).

Once the user had done filling out the remaining application form in the application page, the form would then be submitted to this processApplication servlet. These request parameters were then assigned to the local variables with the appropriate data type for later use.

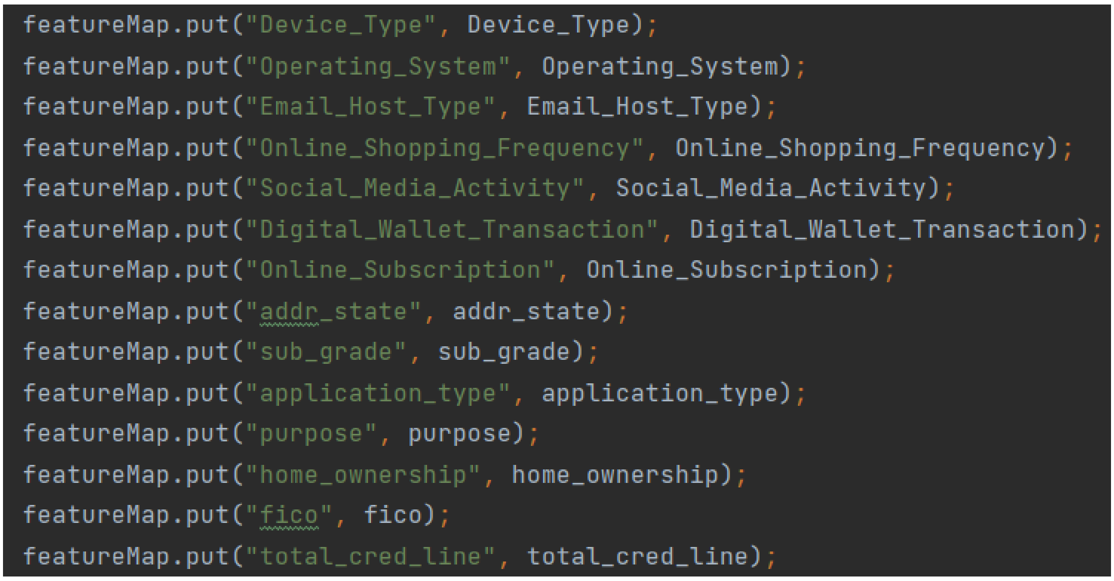
3.



```
URL url = new URL( spec: "http://localhost:5000/processApplication"); // Flask endpoint URL
HttpURLConnection connection = (HttpURLConnection) url.openConnection();
connection.setRequestMethod("POST");
connection.setRequestProperty("Content-Type", "application/json");
connection.setDoOutput(true);

// Create JSON request body
Map<String, Object> featureMap = new HashMap<>();
featureMap.put("loan_amnt", loan_amnt);
featureMap.put("term", term);
featureMap.put("int_rate", int_rate);
featureMap.put("installment", installment);
featureMap.put("annual_inc", annual_inc);
featureMap.put("dti", dti);
featureMap.put("earliest_cr_line", earliest_cr_line);
featureMap.put("open_acc", open_acc);
featureMap.put("pub_rec", pub_rec);
featureMap.put("revol_bal", revol_bal);
featureMap.put("revol_util", revol_util);
featureMap.put("total_acc", total_acc);
featureMap.put("mort_acc", mort_acc);
featureMap.put("pub_rec_bankruptcies", pub_rec_bankruptcies);
```

Figure 5.3.5.5: Code snippet of the processApplication servlet (part 4).

The servlet will open a connection to the endpoint of the Flask-based microservice. The API endpoint will be used to send the data to the trained model in the microservice.

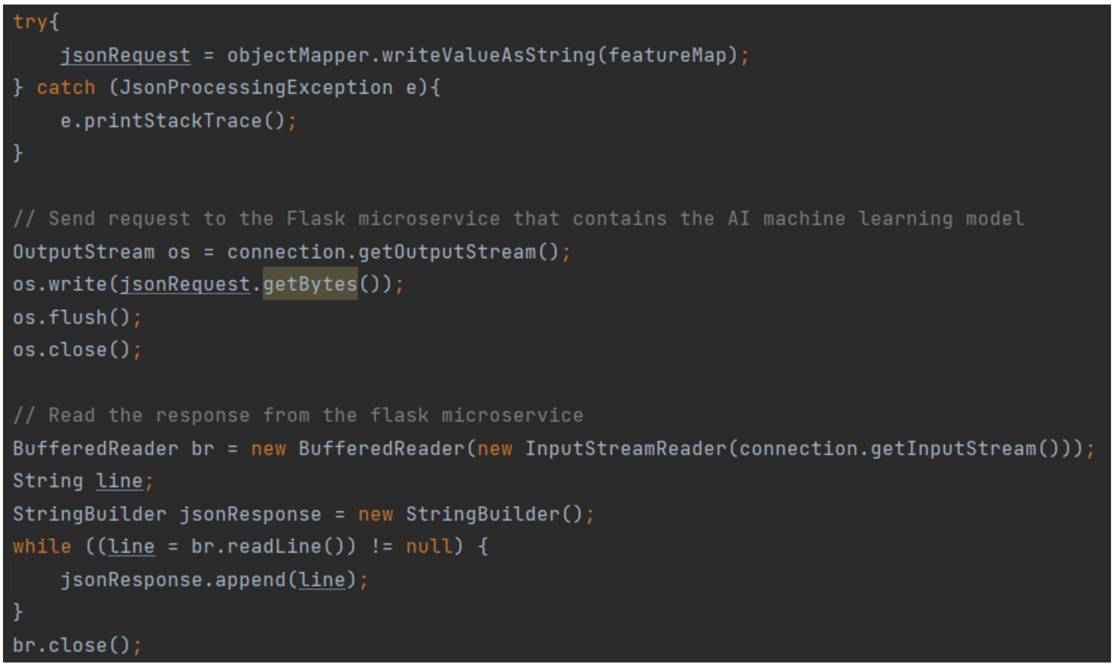


```
featureMap.put("Device_Type", Device_Type);
featureMap.put("Operating_System", Operating_System);
featureMap.put("Email_Host_Type", Email_Host_Type);
featureMap.put("Online_Shopping_Frequency", Online_Shopping_Frequency);
featureMap.put("Social_Media_Activity", Social_Media_Activity);
featureMap.put("Digital_Wallet_Transaction", Digital_Wallet_Transaction);
featureMap.put("Online_Subscription", Online_Subscription);
featureMap.put("addr_state", addr_state);
featureMap.put("sub_grade", sub_grade);
featureMap.put("application_type", application_type);
featureMap.put("purpose", purpose);
featureMap.put("home_ownership", home_ownership);
featureMap.put("fico", fico);
featureMap.put("total_cred_line", total_cred_line);
```

Figure 5.3.5.6: Code snippet of the processApplication servlet (part 5).

All the local variables defined earlier were then be put into a HashMap.

4.

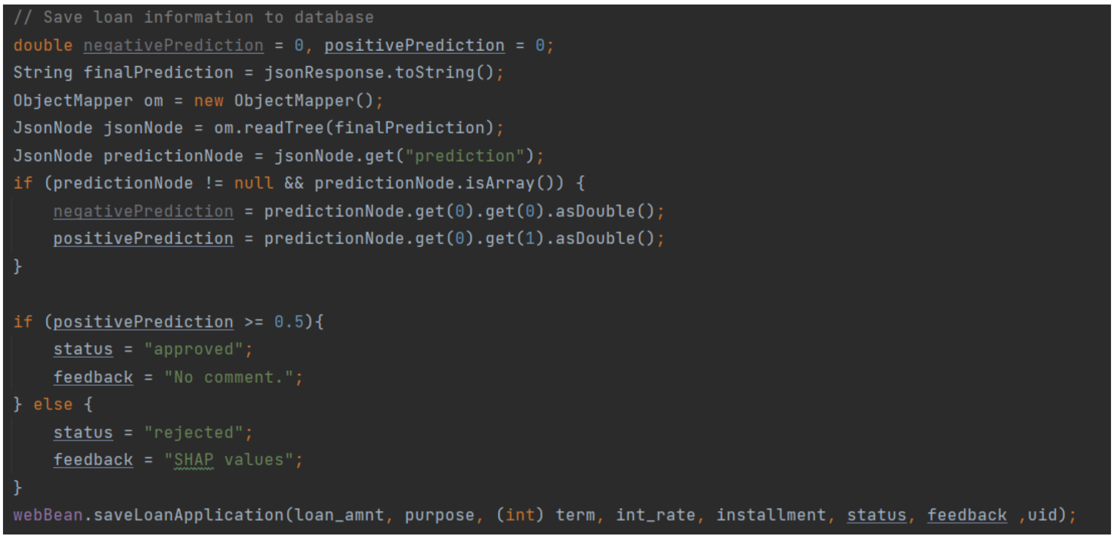


```
try{
    jsonRequest = objectMapper.writeValueAsString(featureMap);
} catch (JsonProcessingException e){
    e.printStackTrace();
}

// Send request to the Flask microservice that contains the AI machine learning model
OutputStream os = connection.getOutputStream();
os.write(jsonRequest.getBytes());
os.flush();
os.close();

// Read the response from the flask microservice
BufferedReader br = new BufferedReader(new InputStreamReader(connection.getInputStream()));
String line;
StringBuilder jsonResponse = new StringBuilder();
while ((line = br.readLine()) != null) {
    jsonResponse.append(line);
}
br.close();
```

Figure 5.3.5.7: Code snippet of the processApplication servlet (part 6).

The hash map was then converted into a Json format by using the object mapper function. The final list of user's data was sent to the trained model through API endpoint. Once the model had made its prediction, the result was then sent back to this servlet for further processing.

5.

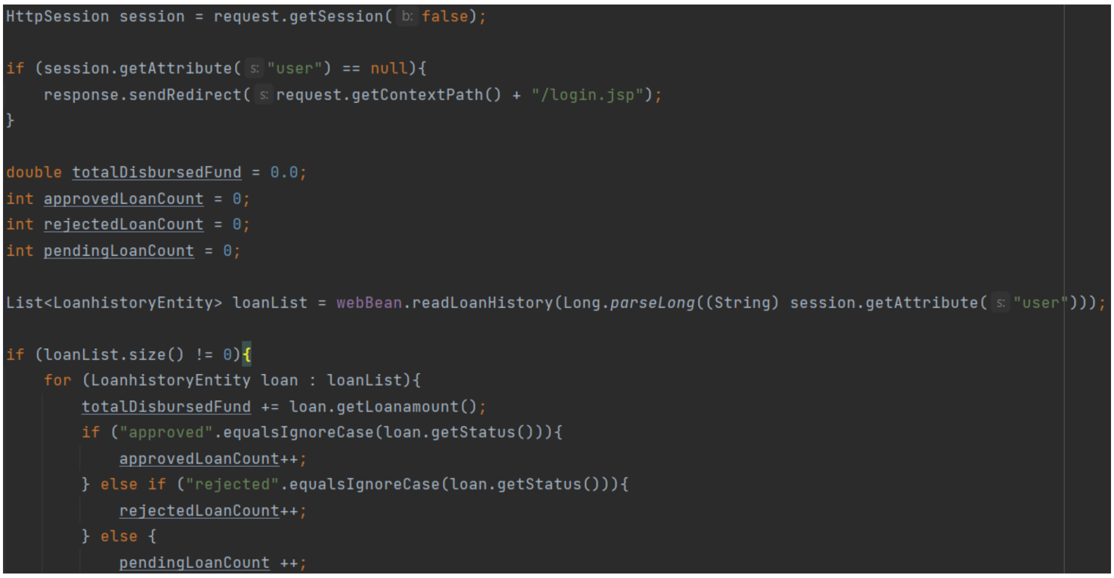


```
// Save loan information to database
double negativePrediction = 0, positivePrediction = 0;
String finalPrediction = jsonResponse.toString();
ObjectMapper om = new ObjectMapper();
JsonNode jsonNode = om.readTree(finalPrediction);
JsonNode predictionNode = jsonNode.get("prediction");
if (predictionNode != null && predictionNode.isArray()) {
    negativePrediction = predictionNode.get(0).get(0).asDouble();
    positivePrediction = predictionNode.get(0).get(1).asDouble();
}

if (positivePrediction >= 0.5){
    status = "approved";
    feedback = "No comment.";
} else {
    status = "rejected";
    feedback = "SHAP values";
}
webBean.saveLoanApplication(loan_amnt, purpose, (int) term, int_rate, installment, status, feedback ,uid);
```

Figure 5.3.5.8: Code snippet of the processApplication servlet (part 7).

The prediction result was then interpreted and if the probability of the positive class was equal or more than 50%, then the application will be approved. If not, then the loan application will be rejected and the SHAP values will be provided for explanation.

5.3.6 Dashboard Module

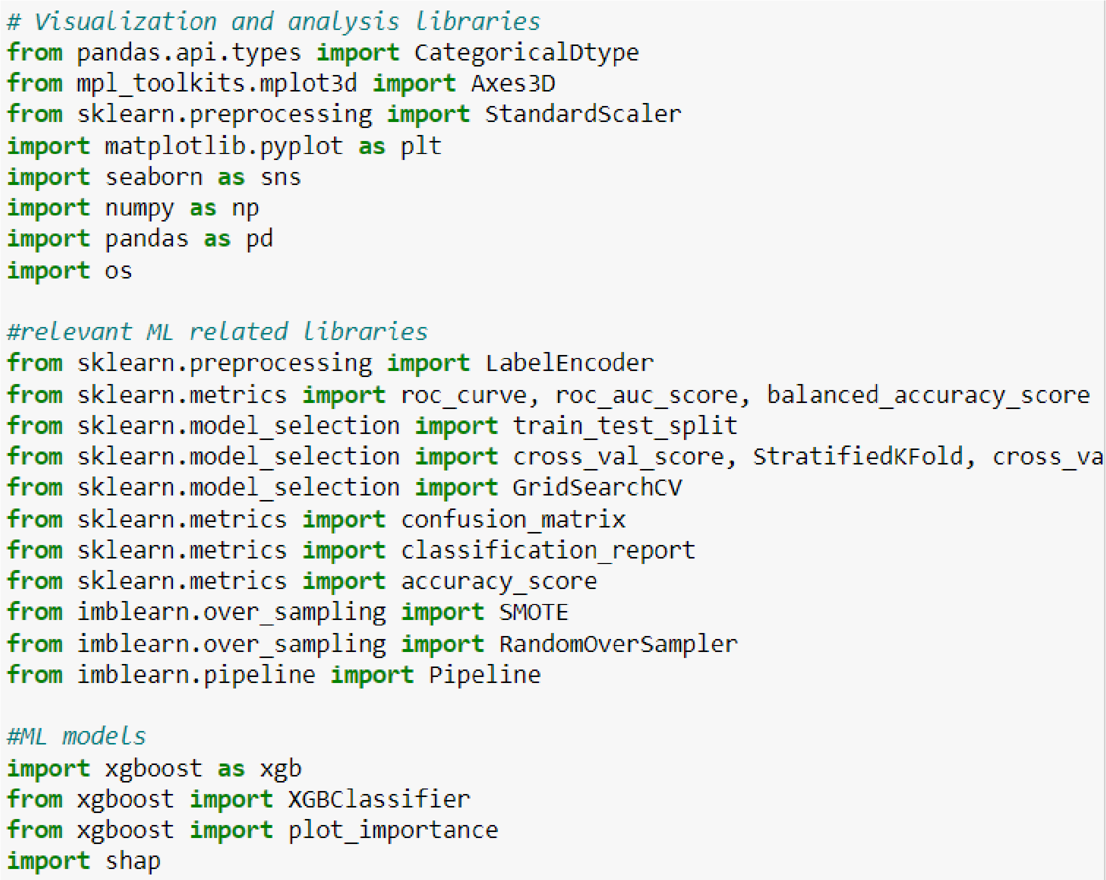


```
HttpSession session = request.getSession( b: false);

if (session.getAttribute( s: "user") == null){
    response.sendRedirect( s: request.getContextPath() + "/login.jsp");
}

double totalDisbursedFund = 0.0;
int approvedLoanCount = 0;
int rejectedLoanCount = 0;
int pendingLoanCount = 0;

List<LoanhistoryEntity> loanList = webBean.readLoanHistory(Long.parseLong((String) session.getAttribute( s: "user")));

if (loanList.size() != 0){
    for (LoanhistoryEntity loan : loanList){
        totalDisbursedFund += loan.getLoanamount();
        if ("approved".equalsIgnoreCase(loan.getStatus())){
            approvedLoanCount++;
        } else if ("rejected".equalsIgnoreCase(loan.getStatus())){
            rejectedLoanCount++;
        } else {
            pendingLoanCount ++;
```

Figure 5.3.6.1: Code snippet of the dashboard servlet.

5.3.7 Personal Loan Machine Learning Model Setup

1.



```
# Visualization and analysis libraries
from pandas.api.types import CategoricalDtype
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
import os

#relevant ML related libraries
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import roc_curve, roc_auc_score, balanced_accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score, StratifiedKFold, cross_va
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from imblearn.over_sampling import SMOTE
from imblearn.over_sampling import RandomOverSampler
from imblearn.pipeline import Pipeline

#ML models
import xgboost as xgb
from xgboost import XGBClassifier
from xgboost import plot_importance
import shap
```

Figure 5.3.7.1: Screenshot of the required libraries to build the personal loan model.

The dataset was then loaded into the Jupyter Notebook by using the “read_csv” function from Pandas. The shape of the initial dataset was 2,260,701 rows and 151 columns. A simple analysis was then conducted on the dataset.

2.

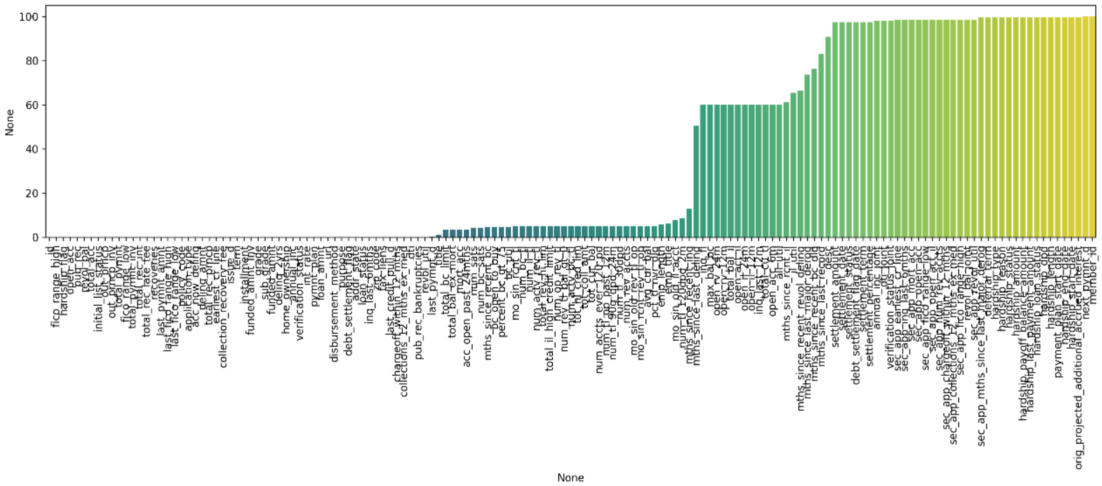


Figure 5.3.7.2: The graph shows the number of missing values across the features.

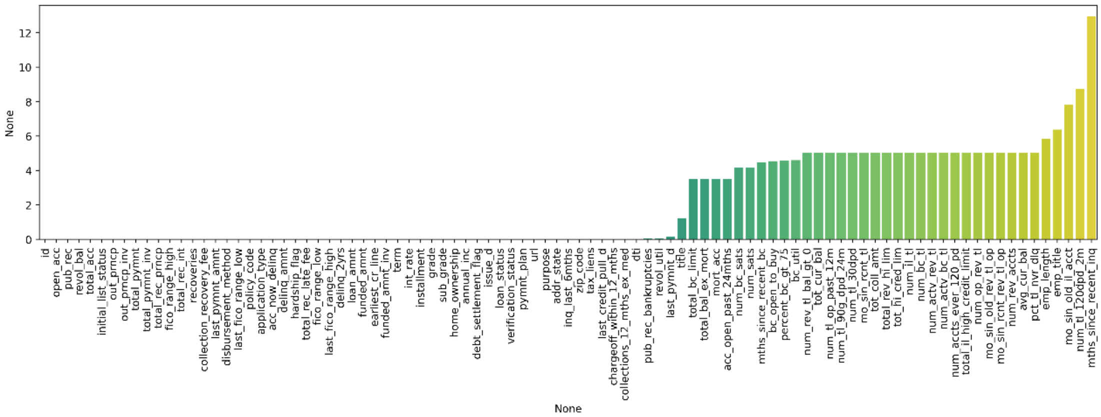
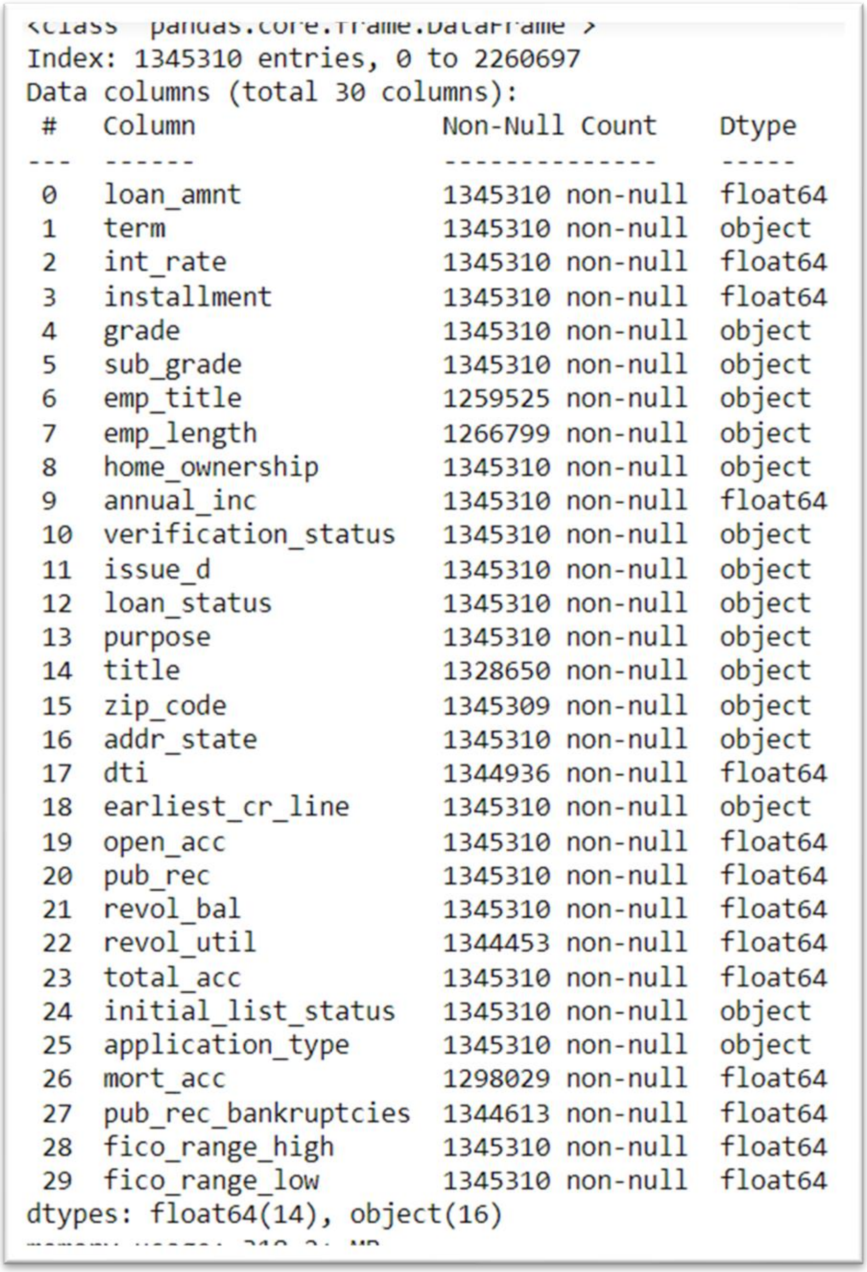


Figure 5.3.7.3: The graph shows the number of missing values across the features after dropping columns with more than 50% missing values.

After dropping unmeaningful features, there were still 93 features which were still a lot. It was not ideal to include all as the current project did not have access to domain/expert knowledge. Moreover, some of the features were not even made available to the borrower before the loan was granted. Hence, the final 30 features had been concluded in this project.

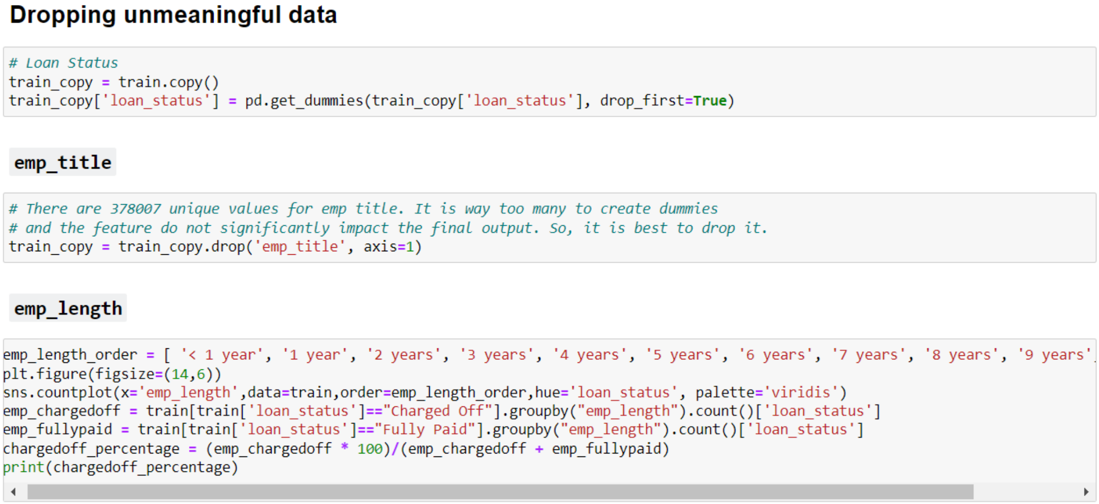


```
<class 'pandas.core.frame.DataFrame'>
Index: 1345310 entries, 0 to 2260697
Data columns (total 30 columns):
 #   Column                Non-Null Count    Dtype  
---  ------                --------------    -----  
 0   loan_amnt             1345310 non-null  float64
 1   term                  1345310 non-null  object 
 2   int_rate              1345310 non-null  float64
 3   installment           1345310 non-null  float64
 4   grade                 1345310 non-null  object 
 5   sub_grade             1345310 non-null  object 
 6   emp_title             1259525 non-null  object 
 7   emp_length            1266799 non-null  object 
 8   home_ownership        1345310 non-null  object 
 9   annual_inc            1345310 non-null  float64
 10  verification_status   1345310 non-null  object 
 11  issue_d               1345310 non-null  object 
 12  loan_status           1345310 non-null  object 
 13  purpose               1345310 non-null  object 
 14  title                 1328650 non-null  object 
 15  zip_code              1345309 non-null  object 
 16  addr_state            1345310 non-null  object 
 17  dti                   1344936 non-null  float64
 18  earliest_cr_line      1345310 non-null  object 
 19  open_acc              1345310 non-null  float64
 20  pub_rec               1345310 non-null  float64
 21  revol_bal             1345310 non-null  float64
 22  revol_util            1344453 non-null  float64
 23  total_acc             1345310 non-null  float64
 24  initial_list_status   1345310 non-null  object 
 25  application_type      1345310 non-null  object 
 26  mort_acc              1298029 non-null  float64
 27  pub_rec_bankruptcies  1344613 non-null  float64
 28  fico_range_high       1345310 non-null  float64
 29  fico_range_low        1345310 non-null  float64
dtypes: float64(14), object(16)
memory usage: 318.2+ MB
```

Figure 5.3.7.4: Screenshot of the info of the final 30 features.

Data pre-processing was needed to ensure that features would contribute positively to the final prediction model. Missing values imputation, dropping unmeaningful features and encoding categorical data were necessary for this dataset.

3.



Dropping unmeaningful data

```
# Loan Status
train_copy = train.copy()
train_copy['loan_status'] = pd.get_dummies(train_copy['loan_status'], drop_first=True)
```

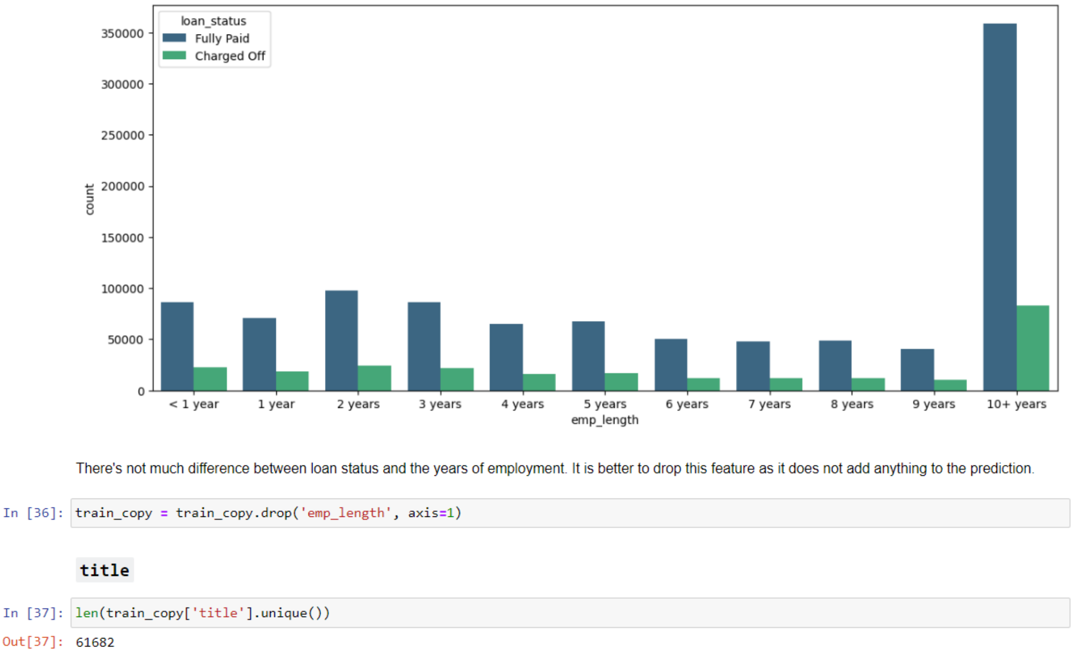
emp_title

```
# There are 378007 unique values for emp title. It is way too many to create dummies
# and the feature do not significantly impact the final output. So, it is best to drop it.
train_copy = train_copy.drop('emp_title', axis=1)
```

emp_length

```
emp_length_order = [ '< 1 year', '1 year', '2 years', '3 years', '4 years', '5 years', '6 years', '7 years', '8 years', '9 years'
plt.figure(figsize=(14,6))
sns.countplot(x='emp_length',data=train,order=emp_length_order,hue='loan_status', palette='viridis')
emp_chargedoff = train[train['loan_status']=="Charged Off"].groupby("emp_length").count()['loan_status']
emp_fullypaid = train[train['loan_status']=="Fully Paid"].groupby("emp_length").count()['loan_status']
chargedoff_percentage = (emp_chargedoff * 100)/(emp_chargedoff + emp_fullypaid)
print(chargedoff_percentage)
```

Figure 5.3.7.5: Code snippet of dropping unmeaningful data (part 1).



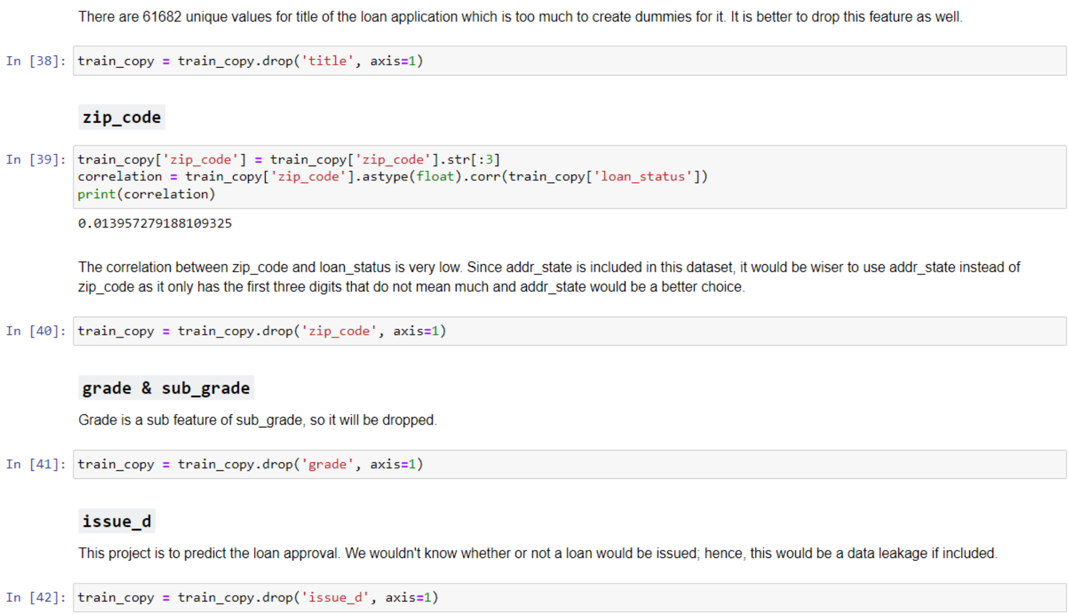
There's not much difference between loan status and the years of employment. It is better to drop this feature as it does not add anything to the prediction.

```
In [36]: train_copy = train_copy.drop('emp_length', axis=1)
```

title

```
In [37]: len(train_copy['title'].unique())

Out[37]: 61682
```

Figure 5.3.7.6: Code snippet of dropping unmeaningful data (part 2).



There are 61682 unique values for title of the loan application which is too much to create dummies for it. It is better to drop this feature as well.

```
In [38]: train_copy = train_copy.drop('title', axis=1)
```

zip_code

```
In [39]: train_copy['zip_code'] = train_copy['zip_code'].str[:3]
         correlation = train_copy['zip_code'].astype(float).corr(train_copy['loan_status'])
         print(correlation)

         0.013957279188109325
```

The correlation between zip_code and loan_status is very low. Since addr_state is included in this dataset, it would be wiser to use addr_state instead of zip_code as it only has the first three digits that do not mean much and addr_state would be a better choice.

```
In [40]: train_copy = train_copy.drop('zip_code', axis=1)
```

grade & sub_grade

Grade is a sub feature of sub_grade, so it will be dropped.

```
In [41]: train_copy = train_copy.drop('grade', axis=1)
```

issue_d

This project is to predict the loan approval. We wouldn't know whether or not a loan would be issued; hence, this would be a data leakage if included.

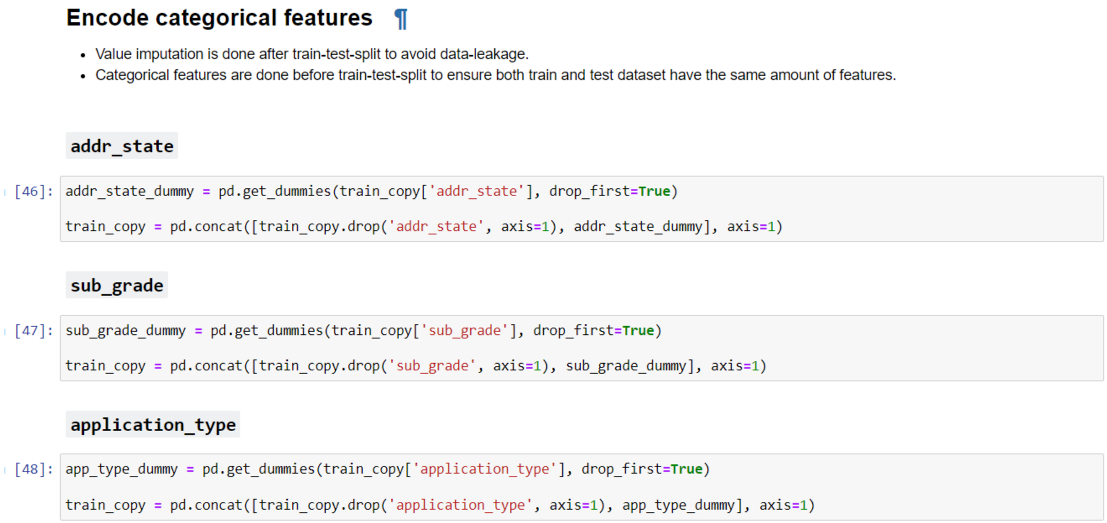
```
In [42]: train_copy = train_copy.drop('issue_d', axis=1)
```

Figure 5.3.7.7: Code snippet of dropping unmeaningful data (part 3).

Unmeaningful features that were dropped were emp_title, emp_length, title, zip_code, grade, and issue_d. Emp_title, emp_length, and title had too many unique values which were way too much to create dummies for these features. Moreover, only the first three digits of the zip_code were visible, and it proved to not add anything to the final prediction model. Grade was a sub feature of sub_grade, hence; it was dropped. Lastly, the issue_d was a problem of data leakage and it needed to be removed as well.

4.

After removing these features, only important features were kept as the final feature. Some of these features were categorical and needed to be encoded.



Encode categorical features ¶

- Value imputation is done after train-test-split to avoid data-leakage.
- Categorical features are done before train-test-split to ensure both train and test dataset have the same amount of features.

`addr_state`

```
[46]: addr_state_dummy = pd.get_dummies(train_copy['addr_state'], drop_first=True)

      train_copy = pd.concat([train_copy.drop('addr_state', axis=1), addr_state_dummy], axis=1)
```

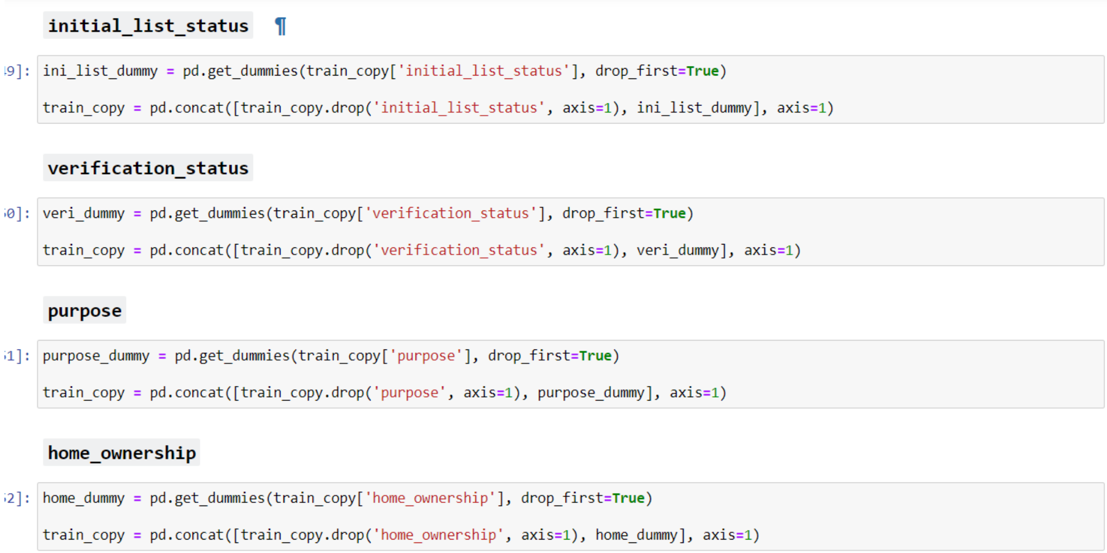
`sub_grade`

```
[47]: sub_grade_dummy = pd.get_dummies(train_copy['sub_grade'], drop_first=True)

      train_copy = pd.concat([train_copy.drop('sub_grade', axis=1), sub_grade_dummy], axis=1)
```

`application_type`

```
[48]: app_type_dummy = pd.get_dummies(train_copy['application_type'], drop_first=True)

      train_copy = pd.concat([train_copy.drop('application_type', axis=1), app_type_dummy], axis=1)
```

Figure 5.3.7.8: Code snippet of encoding categorical features (part 1).



`initial_list_status` ¶

```
9]: ini_list_dummy = pd.get_dummies(train_copy['initial_list_status'], drop_first=True)

    train_copy = pd.concat([train_copy.drop('initial_list_status', axis=1), ini_list_dummy], axis=1)
```

`verification_status`

```
0]: veri_dummy = pd.get_dummies(train_copy['verification_status'], drop_first=True)

    train_copy = pd.concat([train_copy.drop('verification_status', axis=1), veri_dummy], axis=1)
```

`purpose`

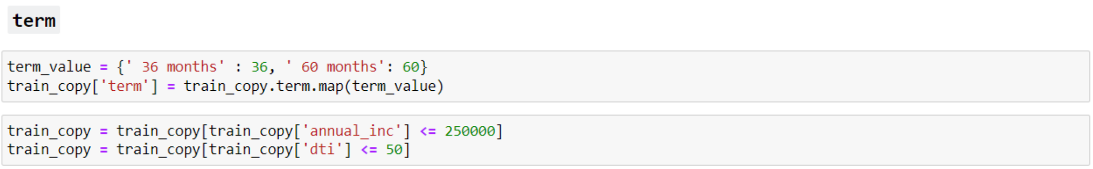
```
1]: purpose_dummy = pd.get_dummies(train_copy['purpose'], drop_first=True)

    train_copy = pd.concat([train_copy.drop('purpose', axis=1), purpose_dummy], axis=1)
```

`home_ownership`

```
2]: home_dummy = pd.get_dummies(train_copy['home_ownership'], drop_first=True)

    train_copy = pd.concat([train_copy.drop('home_ownership', axis=1), home_dummy], axis=1)
```

Figure 5.3.7.9: Code snippet of encoding categorical features (part 2).

Once all the categorical features had been encoded, the next step was to split the dataset into train and test datatest.



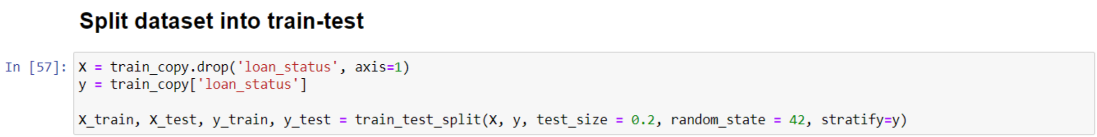
term

```
term_value = {' 36 months' : 36, ' 60 months': 60}
train_copy['term'] = train_copy.term.map(term_value)
```

```
train_copy = train_copy[train_copy['annual_inc'] <= 250000]
train_copy = train_copy[train_copy['dti'] <= 50]
```

Figure 5.3.7.10: Code snippet of encoding categorical features (part 3).

5.



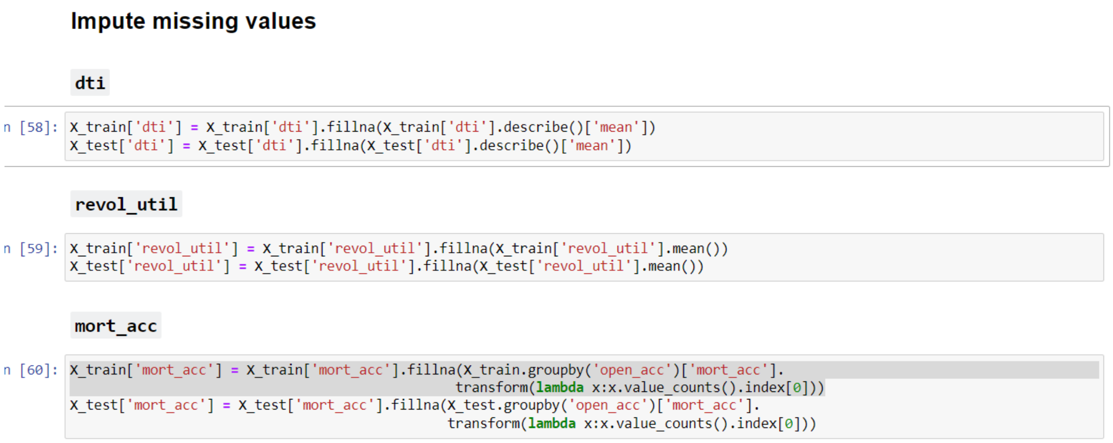
Split dataset into train-test

```
In [57]: X = train_copy.drop('loan_status', axis=1)
         y = train_copy['loan_status']

         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42, stratify=y)
```

Figure 5.3.7.11: Code snippet of splitting the dataset.

The dataset was split into training and testing set with 8:3 ratio. Data imputation could only be done after splitting the dataset to avoid the issue of data leakage during the training and testing of the model.

6.



Impute missing values

dti

```
n [58]: X_train['dti'] = X_train['dti'].fillna(X_train['dti'].describe()['mean'])
        X_test['dti'] = X_test['dti'].fillna(X_test['dti'].describe()['mean'])
```

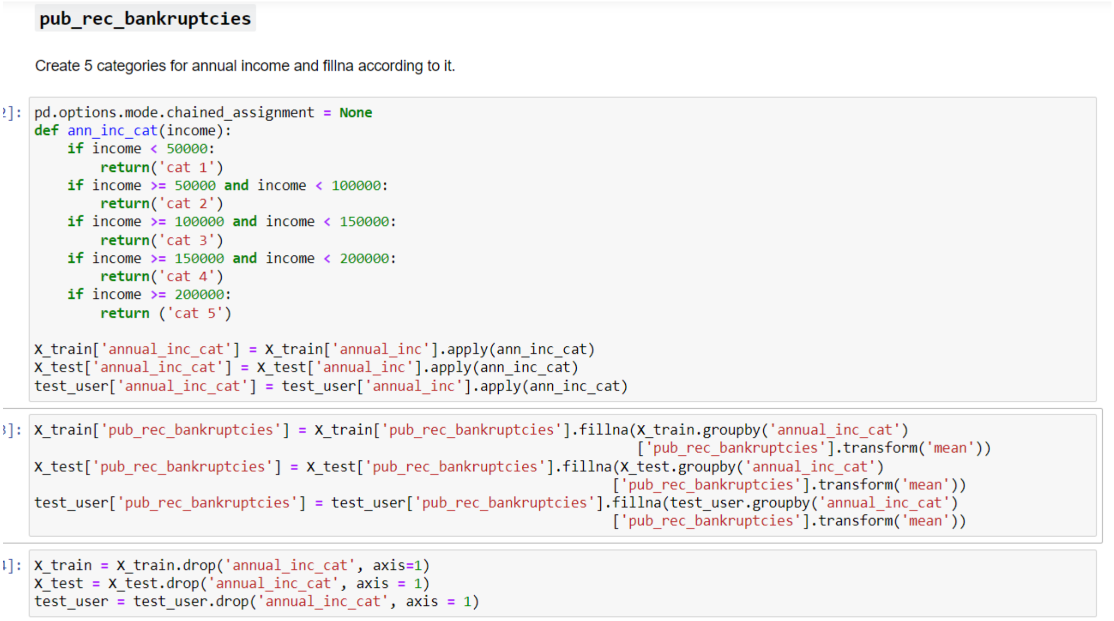
revol_util

```
n [59]: X_train['revol_util'] = X_train['revol_util'].fillna(X_train['revol_util'].mean())
        X_test['revol_util'] = X_test['revol_util'].fillna(X_test['revol_util'].mean())
```

mort_acc

```
n [60]: X_train['mort_acc'] = X_train['mort_acc'].fillna(X_train.groupby('open_acc')['mort_acc'].
                                                 transform(lambda x:x.value_counts().index[0]))
        X_test['mort_acc'] = X_test['mort_acc'].fillna(X_test.groupby('open_acc')['mort_acc'].
                                                transform(lambda x:x.value_counts().index[0]))
```

Figure 5.3.7.12: Code snippet of missing values imputation (part 1).



`pub_rec_bankruptcies`

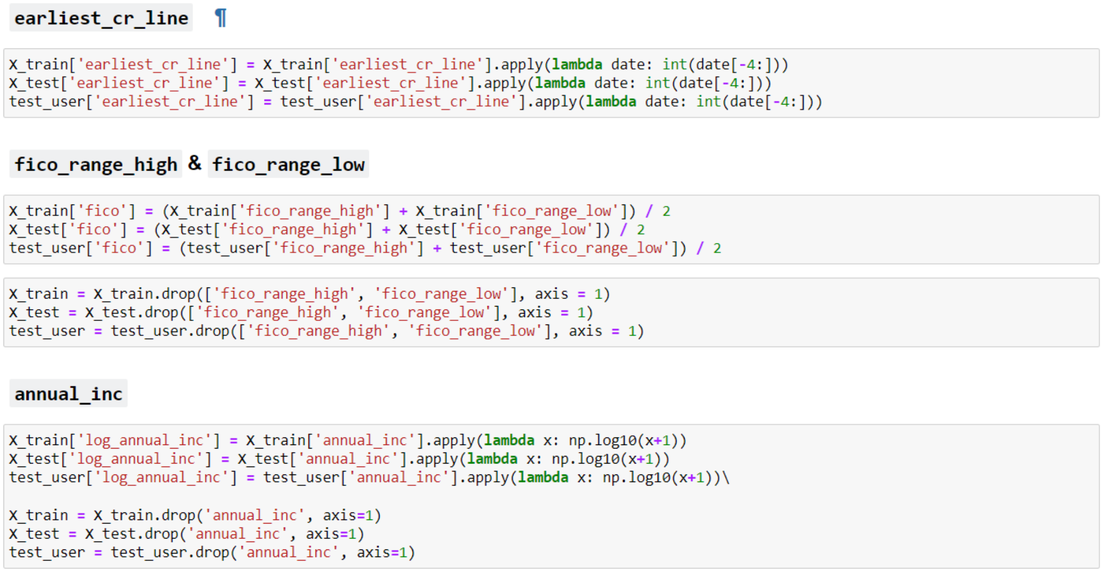
Create 5 categories for annual income and fillna according to it.

```
pd.options.mode.chained_assignment = None
def ann_inc_cat(income):
    if income < 50000:
        return('cat 1')
    if income >= 50000 and income < 100000:
        return('cat 2')
    if income >= 100000 and income < 150000:
        return('cat 3')
    if income >= 150000 and income < 200000:
        return('cat 4')
    if income >= 200000:
        return ('cat 5')

X_train['annual_inc_cat'] = X_train['annual_inc'].apply(ann_inc_cat)
X_test['annual_inc_cat'] = X_test['annual_inc'].apply(ann_inc_cat)
test_user['annual_inc_cat'] = test_user['annual_inc'].apply(ann_inc_cat)
```

```
X_train['pub_rec_bankruptcies'] = X_train['pub_rec_bankruptcies'].fillna(X_train.groupby('annual_inc_cat')
                                                                          ['pub_rec_bankruptcies'].transform('mean'))
X_test['pub_rec_bankruptcies'] = X_test['pub_rec_bankruptcies'].fillna(X_test.groupby('annual_inc_cat')
                                                                       ['pub_rec_bankruptcies'].transform('mean'))
test_user['pub_rec_bankruptcies'] = test_user['pub_rec_bankruptcies'].fillna(test_user.groupby('annual_inc_cat')
                                                                       ['pub_rec_bankruptcies'].transform('mean'))
```

```
X_train = X_train.drop('annual_inc_cat', axis=1)
X_test = X_test.drop('annual_inc_cat', axis = 1)
test_user = test_user.drop('annual_inc_cat', axis = 1)
```

Figure 5.3.7.13: Code snippet of missing values imputation (part 2).



`earliest_cr_line` ¶

```
X_train['earliest_cr_line'] = X_train['earliest_cr_line'].apply(lambda date: int(date[-4:]))
X_test['earliest_cr_line'] = X_test['earliest_cr_line'].apply(lambda date: int(date[-4:]))
test_user['earliest_cr_line'] = test_user['earliest_cr_line'].apply(lambda date: int(date[-4:]))
```

`fico_range_high` & `fico_range_low`

```
X_train['fico'] = (X_train['fico_range_high'] + X_train['fico_range_low']) / 2
X_test['fico'] = (X_test['fico_range_high'] + X_test['fico_range_low']) / 2
test_user['fico'] = (test_user['fico_range_high'] + test_user['fico_range_low']) / 2
```

```
X_train = X_train.drop(['fico_range_high', 'fico_range_low'], axis = 1)
X_test = X_test.drop(['fico_range_high', 'fico_range_low'], axis = 1)
test_user = test_user.drop(['fico_range_high', 'fico_range_low'], axis = 1)
```

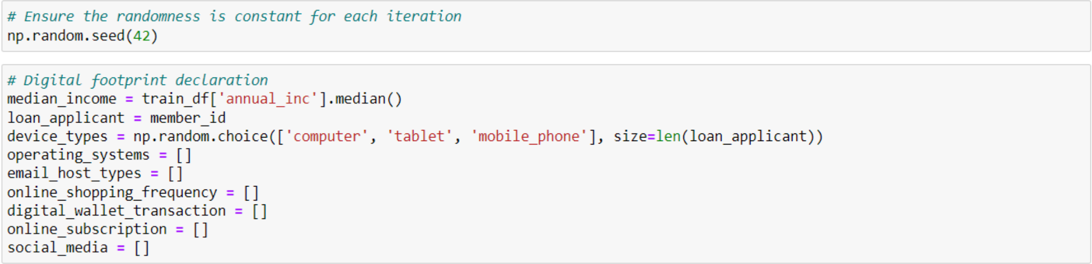
`annual_inc`

```
X_train['log_annual_inc'] = X_train['annual_inc'].apply(lambda x: np.log10(x+1))
X_test['log_annual_inc'] = X_test['annual_inc'].apply(lambda x: np.log10(x+1))
test_user['log_annual_inc'] = test_user['annual_inc'].apply(lambda x: np.log10(x+1))\

X_train = X_train.drop('annual_inc', axis=1)
X_test = X_test.drop('annual_inc', axis=1)
test_user = test_user.drop('annual_inc', axis=1)
```

Figure 5.3.7.14: Code snippet of missing values imputation (part 3).

The original dataset had been pre-processed and cleaned and it was ready to be used for the training and testing of the first prediction model. Before the training stage, it was needed to create a new dataset by reusing the original dataset and amend it with digital footprint data.

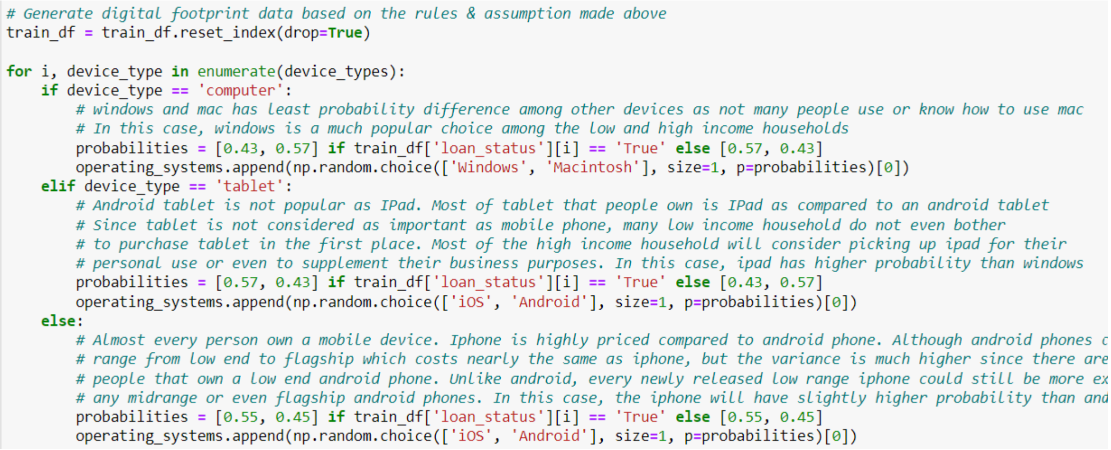
7.



```
# Ensure the randomness is constant for each iteration
np.random.seed(42)
```

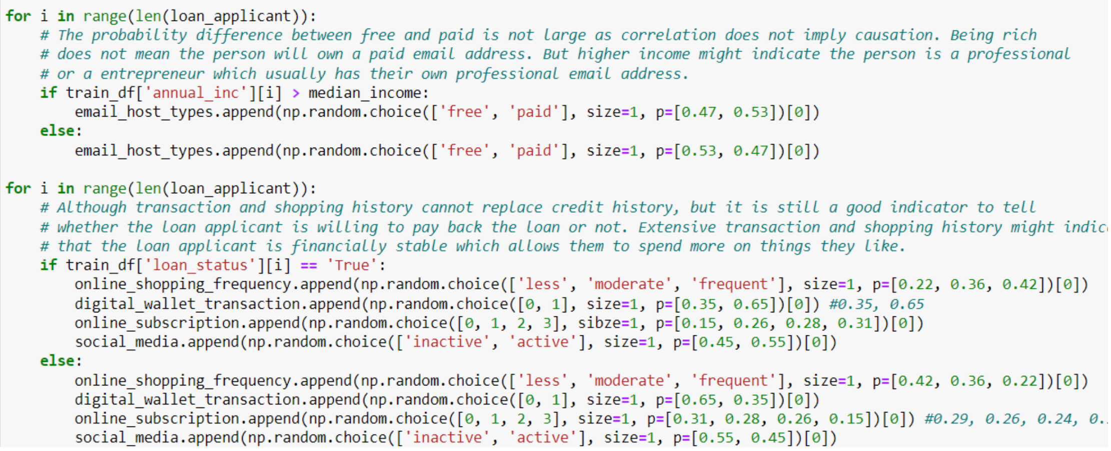
```
# Digital footprint declaration
median_income = train_df['annual_inc'].median()
loan_applicant = member_id
device_types = np.random.choice(['computer', 'tablet', 'mobile_phone'], size=len(loan_applicant))
operating_systems = []
email_host_types = []
online_shopping_frequency = []
digital_wallet_transaction = []
online_subscription = []
social_media = []
```

Figure 5.3.7.15: Code snippet for creating digital footprint data (part 1).



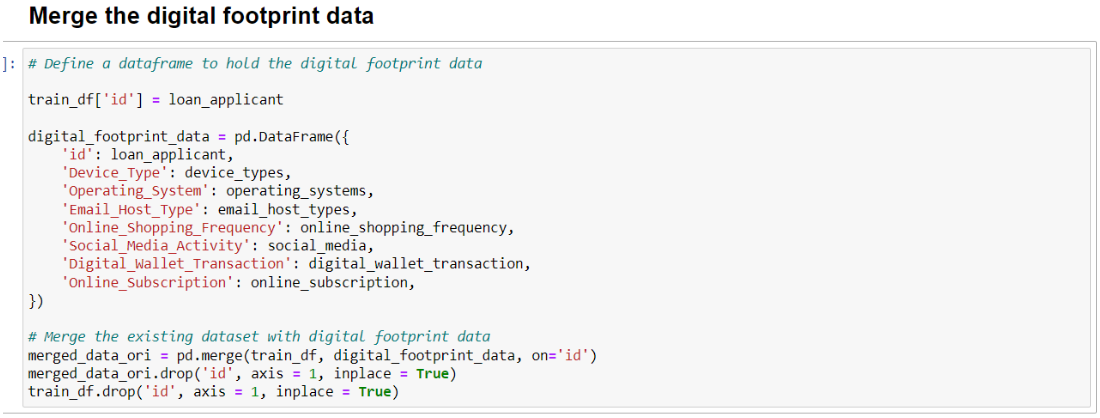
```
# Generate digital footprint data based on the rules & assumption made above
train_df = train_df.reset_index(drop=True)

for i, device_type in enumerate(device_types):
    if device_type == 'computer':
        # windows and mac has least probability difference among other devices as not many people use or know how to use mac
        # In this case, windows is a much popular choice among the low and high income households
        probabilities = [0.43, 0.57] if train_df['loan_status'][i] == 'True' else [0.57, 0.43]
        operating_systems.append(np.random.choice(['Windows', 'Macintosh'], size=1, p=probabilities)[0])
    elif device_type == 'tablet':
        # Android tablet is not popular as IPad. Most of tablet that people own is IPad as compared to an android tablet
        # Since tablet is not considered as important as mobile phone, many low income household do not even bother
        # to purchase tablet in the first place. Most of the high income household will consider picking up ipad for their
        # personal use or even to supplement their business purposes. In this case, ipad has higher probability than windows
        probabilities = [0.57, 0.43] if train_df['loan_status'][i] == 'True' else [0.43, 0.57]
        operating_systems.append(np.random.choice(['iOS', 'Android'], size=1, p=probabilities)[0])
    else:
        # Almost every person own a mobile device. Iphone is highly priced compared to android phone. Although android phones c
        # range from low end to flagship which costs nearly the same as iphone, but the variance is much higher since there are
        # people that own a low end android phone. Unlike android, every newly released low range iphone could still be more ex
        # any midrange or even flagship android phones. In this case, the iphone will have slightly higher probability than and
        probabilities = [0.55, 0.45] if train_df['loan_status'][i] == 'True' else [0.55, 0.45]
        operating_systems.append(np.random.choice(['iOS', 'Android'], size=1, p=probabilities)[0])
```

Figure 5.3.7.16: Code snippet for creating digital footprint data (part 2).



```
for i in range(len(loan_applicant)):
    # The probability difference between free and paid is not large as correlation does not imply causation. Being rich
    # does not mean the person will own a paid email address. But higher income might indicate the person is a professional
    # or a entrepreneur which usually has their own professional email address.
    if train_df['annual_inc'][i] > median_income:
        email_host_types.append(np.random.choice(['free', 'paid'], size=1, p=[0.47, 0.53])[0])
    else:
        email_host_types.append(np.random.choice(['free', 'paid'], size=1, p=[0.53, 0.47])[0])

for i in range(len(loan_applicant)):
    # Although transaction and shopping history cannot replace credit history, but it is still a good indicator to tell
    # whether the loan applicant is willing to pay back the loan or not. Extensive transaction and shopping history might indic
    # that the loan applicant is financially stable which allows them to spend more on things they like.
    if train_df['loan_status'][i] == 'True':
        online_shopping_frequency.append(np.random.choice(['less', 'moderate', 'frequent'], size=1, p=[0.22, 0.36, 0.42])[0])
        digital_wallet_transaction.append(np.random.choice([0, 1], size=1, p=[0.35, 0.65])[0]) #0.35, 0.65
        online_subscription.append(np.random.choice([0, 1, 2, 3], sibze=1, p=[0.15, 0.26, 0.28, 0.31])[0])
        social_media.append(np.random.choice(['inactive', 'active'], size=1, p=[0.45, 0.55])[0])
    else:
        online_shopping_frequency.append(np.random.choice(['less', 'moderate', 'frequent'], size=1, p=[0.42, 0.36, 0.22])[0])
        digital_wallet_transaction.append(np.random.choice([0, 1], size=1, p=[0.65, 0.35])[0])
        online_subscription.append(np.random.choice([0, 1, 2, 3], size=1, p=[0.31, 0.28, 0.26, 0.15])[0]) #0.29, 0.26, 0.24, 0..
        social_media.append(np.random.choice(['inactive', 'active'], size=1, p=[0.55, 0.45])[0])
```

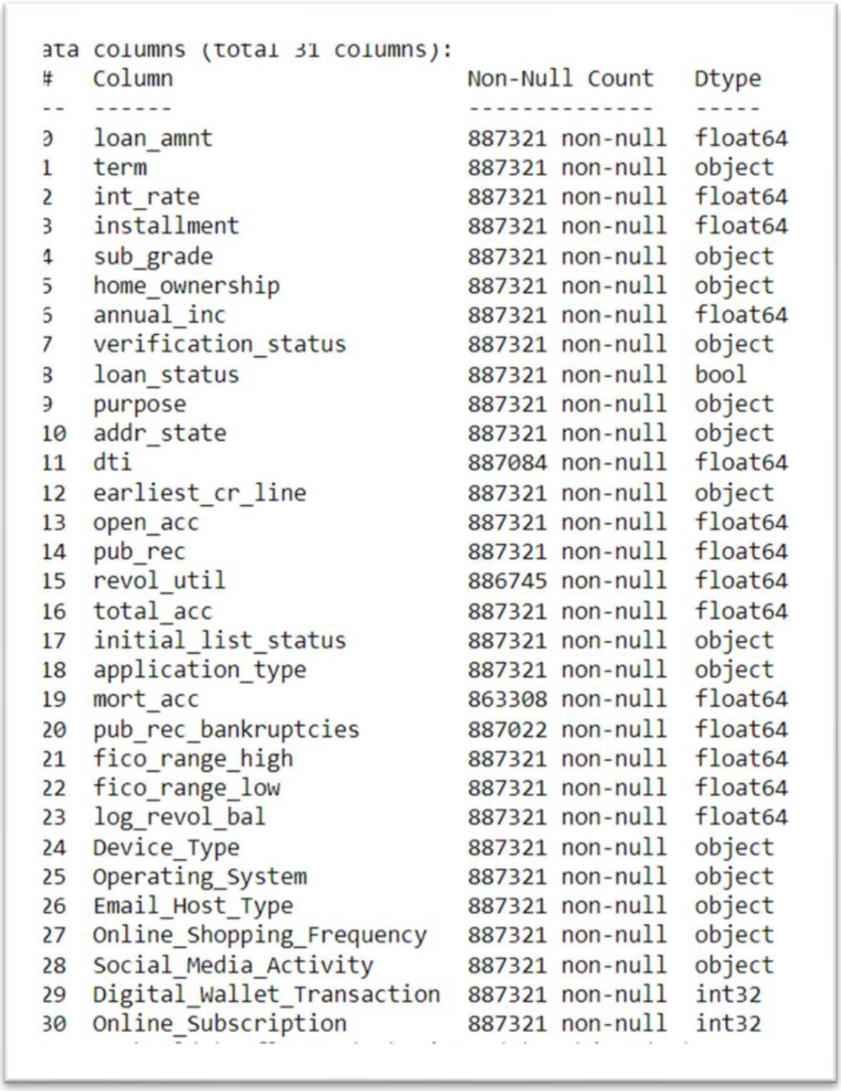
Figure 5.3.7.17: Code snippet for creating digital footprint data (part 3).



Merge the digital footprint data

```
# Define a dataframe to hold the digital footprint data

train_df['id'] = loan_applicant

digital_footprint_data = pd.DataFrame({
    'id': loan_applicant,
    'Device_Type': device_types,
    'Operating_System': operating_systems,
    'Email_Host_Type': email_host_types,
    'Online_Shopping_Frequency': online_shopping_frequency,
    'Social_Media_Activity': social_media,
    'Digital_Wallet_Transaction': digital_wallet_transaction,
    'Online_Subscription': online_subscription,
})

# Merge the existing dataset with digital footprint data
merged_data_ori = pd.merge(train_df, digital_footprint_data, on='id')
merged_data_ori.drop('id', axis = 1, inplace = True)
train_df.drop('id', axis = 1, inplace = True)
```

Figure 5.3.7.18: Code snippet for creating digital footprint data (part 4).

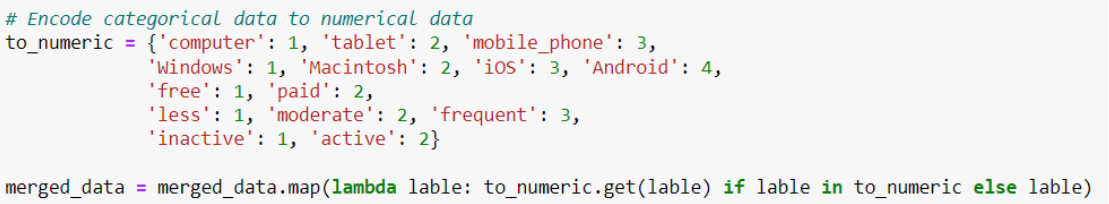


```
ata columns (total 31 columns):
#   Column                      Non-Null Count   Dtype
--  ------                      --------------   -----
0   loan_amnt                   887321 non-null  float64
1   term                        887321 non-null  object
2   int_rate                    887321 non-null  float64
3   installment                 887321 non-null  float64
4   sub_grade                   887321 non-null  object
5   home_ownership              887321 non-null  object
6   annual_inc                  887321 non-null  float64
7   verification_status         887321 non-null  object
8   loan_status                 887321 non-null  bool
9   purpose                     887321 non-null  object
10  addr_state                  887321 non-null  object
11  dti                         887084 non-null  float64
12  earliest_cr_line            887321 non-null  object
13  open_acc                    887321 non-null  float64
14  pub_rec                     887321 non-null  float64
15  revol_util                  886745 non-null  float64
16  total_acc                   887321 non-null  float64
17  initial_list_status         887321 non-null  object
18  application_type            887321 non-null  object
19  mort_acc                    863308 non-null  float64
20  pub_rec_bankruptcies        887022 non-null  float64
21  fico_range_high             887321 non-null  float64
22  fico_range_low              887321 non-null  float64
23  log_revol_bal               887321 non-null  float64
24  Device_Type                 887321 non-null  object
25  Operating_System            887321 non-null  object
26  Email_Host_Type             887321 non-null  object
27  Online_Shopping_Frequency   887321 non-null  object
28  Social_Media_Activity       887321 non-null  object
29  Digital_Wallet_Transaction  887321 non-null  int32
30  Online_Subscription         887321 non-null  int32
```

Figure 5.3.7.19: Screenshot of the info of the newly altered dataset with digital footprint data.

The original dataset was altered and merged with the digital footprint data that were generated based on the rules and assumptions defined above. The newly altered dataset required pre-processing before it can be fed into the prediction model.

8.

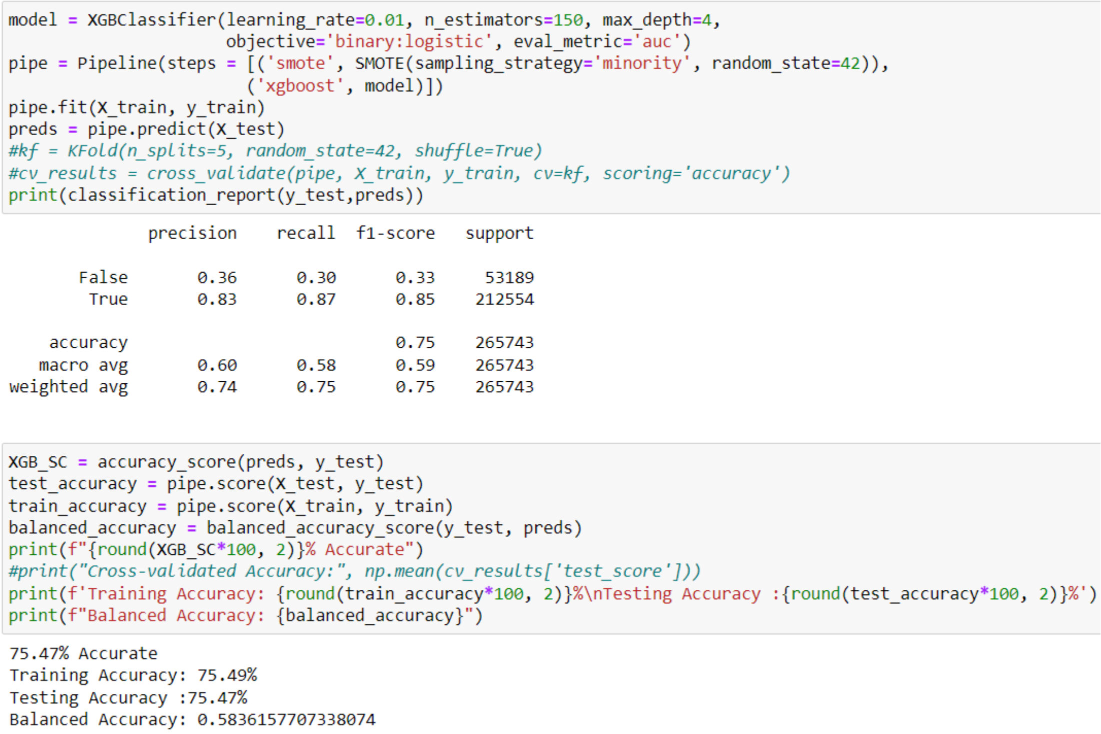


```
# Encode categorical data to numerical data
to_numeric = {'computer': 1, 'tablet': 2, 'mobile_phone': 3,
              'Windows': 1, 'Macintosh': 2, 'iOS': 3, 'Android': 4,
              'free': 1, 'paid': 2,
              'less': 1, 'moderate': 2, 'frequent': 3,
              'inactive': 1, 'active': 2}

merged_data = merged_data.map(lambda lable: to_numeric.get(lable) if lable in to_numeric else lable)
```

Figure 5.3.7.20: Code snippet of encoding the digital footprint data on the newly merged dataset.

After everything were done, two XGBoost models were trained and tested on both of the datasets.

9.



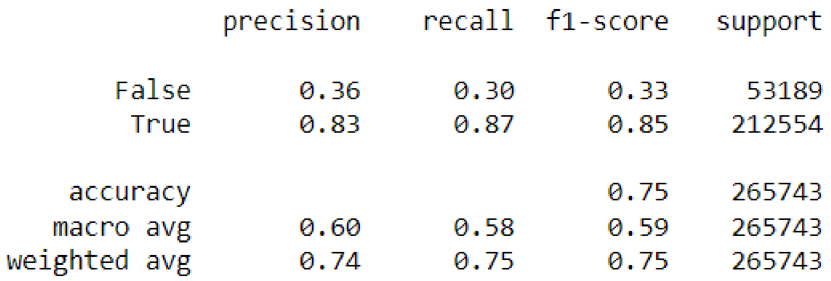
```
model = XGBClassifier(learning_rate=0.01, n_estimators=150, max_depth=4,
                      objective='binary:logistic', eval_metric='auc')
pipe = Pipeline(steps = [('smote', SMOTE(sampling_strategy='minority', random_state=42)),
                         ('xgboost', model)])
pipe.fit(X_train, y_train)
preds = pipe.predict(X_test)
#kf = KFold(n_splits=5, random_state=42, shuffle=True)
#cv_results = cross_validate(pipe, X_train, y_train, cv=kf, scoring='accuracy')
print(classification_report(y_test,preds))
```

```
              precision    recall  f1-score   support

       False       0.36      0.30      0.33     53189
        True       0.83      0.87      0.85    212554

    accuracy                           0.75    265743
   macro avg       0.60      0.58      0.59    265743
weighted avg       0.74      0.75      0.75    265743
```

```
XGB_SC = accuracy_score(preds, y_test)
test_accuracy = pipe.score(X_test, y_test)
train_accuracy = pipe.score(X_train, y_train)
balanced_accuracy = balanced_accuracy_score(y_test, preds)
print(f"{round(XGB_SC*100, 2)}% Accurate")
#print("Cross-validated Accuracy:", np.mean(cv_results['test_score']))
print(f'Training Accuracy: {round(train_accuracy*100, 2)}%\nTesting Accuracy :{round(test_accuracy*100, 2)}%')
print(f"Balanced Accuracy: {balanced_accuracy}")
```

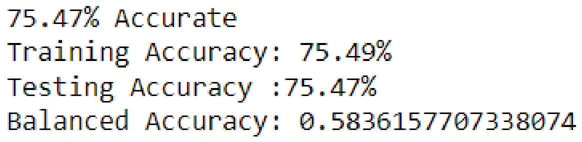
```
75.47% Accurate
Training Accuracy: 75.49%
Testing Accuracy :75.47%
Balanced Accuracy: 0.5836157707338074
```

Figure 5.3.7.21: Code snippet for training and testing of the first XGBoost model.



```
              precision    recall  f1-score   support

       False       0.36      0.30      0.33     53189
        True       0.83      0.87      0.85    212554

    accuracy                           0.75    265743
   macro avg       0.60      0.58      0.59    265743
weighted avg       0.74      0.75      0.75    265743
```

Figure 5.3.7.22: Screenshot of the classification report of the first XGBoost model.



```
75.47% Accurate
Training Accuracy: 75.49%
Testing Accuracy :75.47%
Balanced Accuracy: 0.5836157707338074
```

Figure 5.3.7.23: Screenshot of the training and testing accuracy of the first XGBoost model.

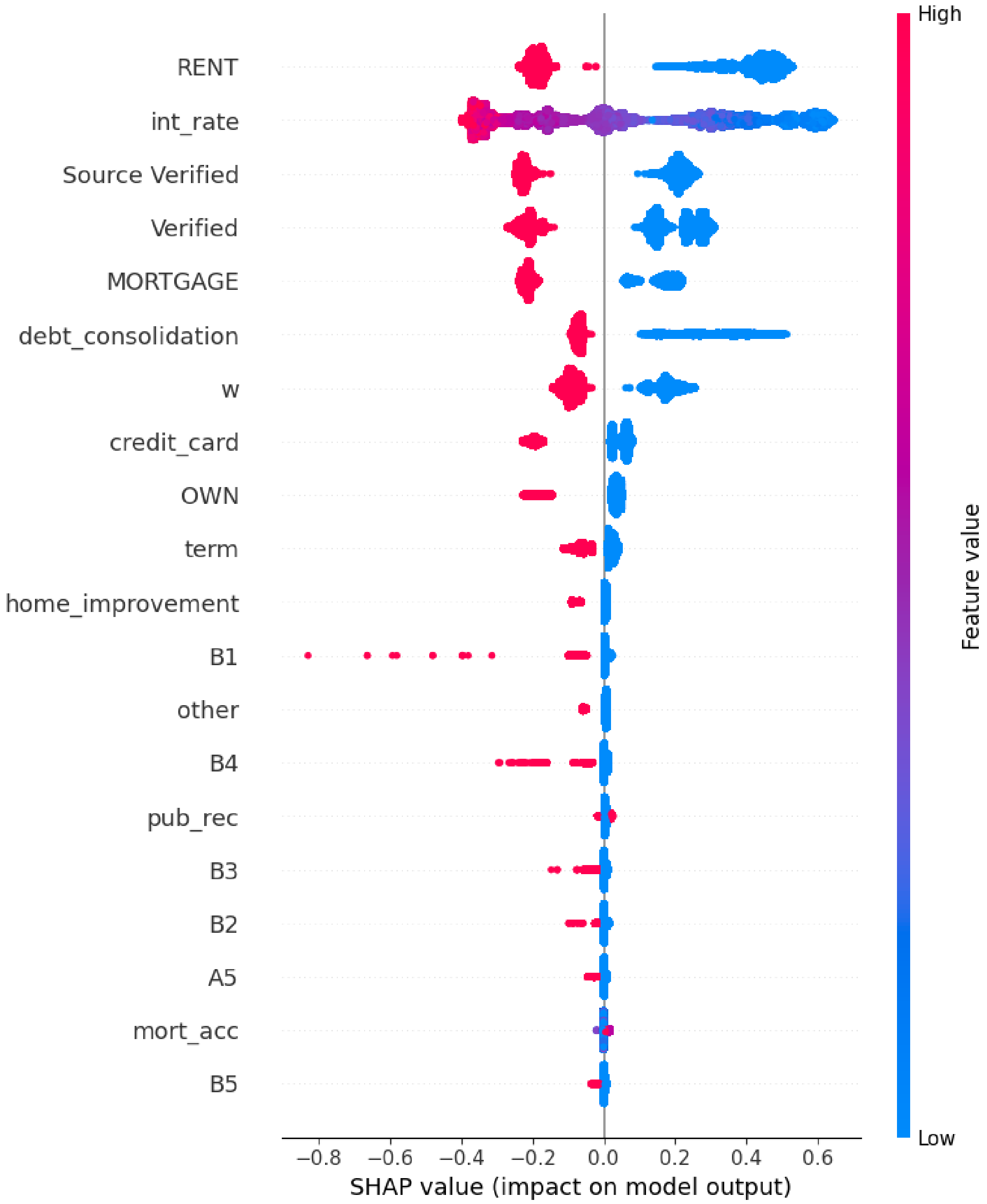
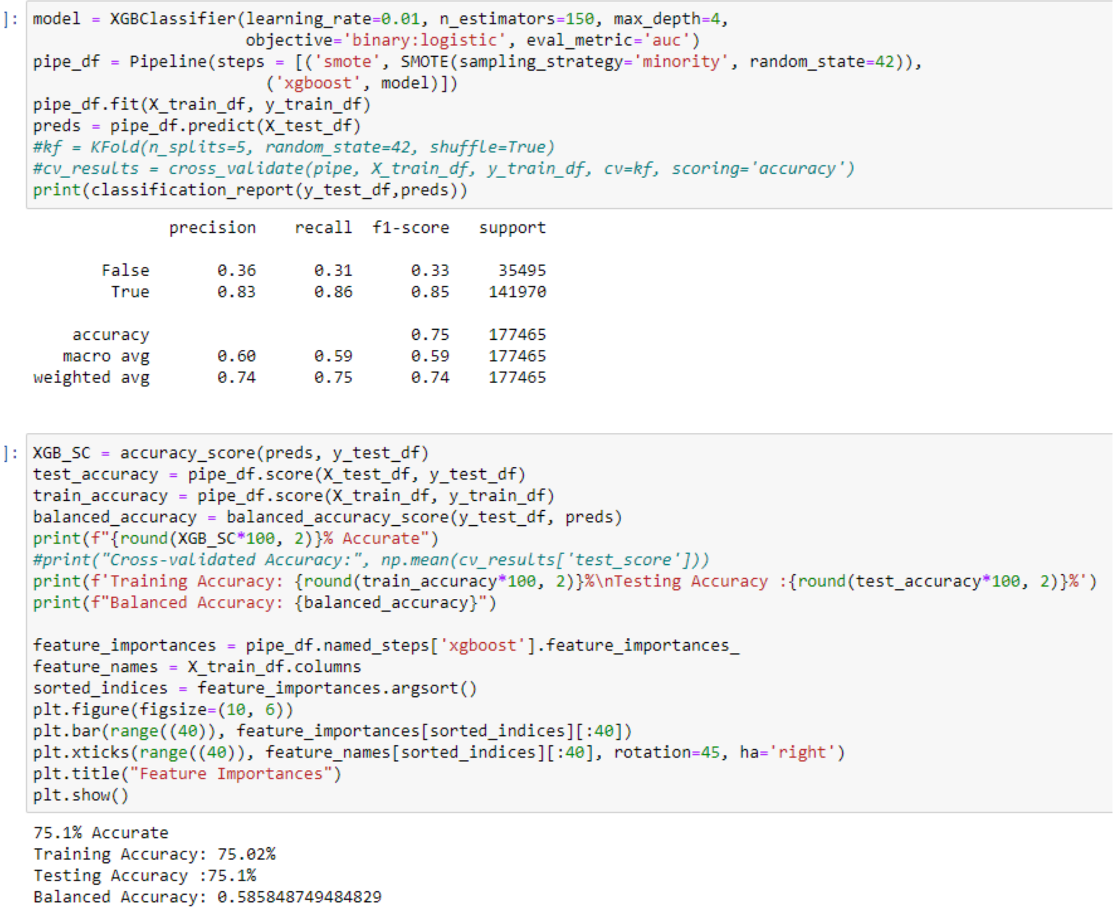


Figure 5.3.7.24: Graph of the SHAP values of the first XGBoost model.

The final result of the first XGBoost model can be seen in Figure 5.3.7.22 and Figure 5.3.7.23. The figure 5.3.7.24 depicts the SHAP values of the XGBoost model. The next step was to train the second XGBoost model for the altered dataset with the digital footprint data.

10.



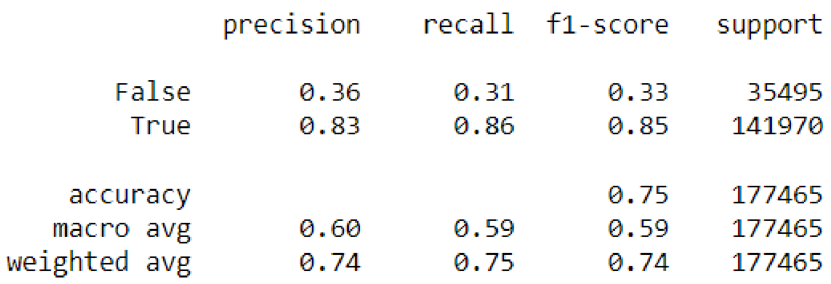
```
]: model = XGBClassifier(learning_rate=0.01, n_estimators=150, max_depth=4,
                         objective='binary:logistic', eval_metric='auc')
   pipe_df = Pipeline(steps = [('smote', SMOTE(sampling_strategy='minority', random_state=42)),
                               ('xgboost', model)])
   pipe_df.fit(X_train_df, y_train_df)
   preds = pipe_df.predict(X_test_df)
   #kf = KFold(n_splits=5, random_state=42, shuffle=True)
   #cv_results = cross_validate(pipe, X_train_df, y_train_df, cv=kf, scoring='accuracy')
   print(classification_report(y_test_df,preds))
```

```
              precision    recall  f1-score   support

       False       0.36      0.31      0.33     35495
        True       0.83      0.86      0.85    141970

    accuracy                           0.75    177465
   macro avg       0.60      0.59      0.59    177465
weighted avg       0.74      0.75      0.74    177465
```

```
]: XGB_SC = accuracy_score(preds, y_test_df)
   test_accuracy = pipe_df.score(X_test_df, y_test_df)
   train_accuracy = pipe_df.score(X_train_df, y_train_df)
   balanced_accuracy = balanced_accuracy_score(y_test_df, preds)
   print(f"{round(XGB_SC*100, 2)}% Accurate")
   #print("Cross-validated Accuracy:", np.mean(cv_results['test_score']))
   print(f'Training Accuracy: {round(train_accuracy*100, 2)}%\nTesting Accuracy :{round(test_accuracy*100, 2)}%')
   print(f"Balanced Accuracy: {balanced_accuracy}")

   feature_importances = pipe_df.named_steps['xgboost'].feature_importances_
   feature_names = X_train_df.columns
   sorted_indices = feature_importances.argsort()
   plt.figure(figsize=(10, 6))
   plt.bar(range((40)), feature_importances[sorted_indices][:40])
   plt.xticks(range((40)), feature_names[sorted_indices][:40], rotation=45, ha='right')
   plt.title("Feature Importances")
   plt.show()
```

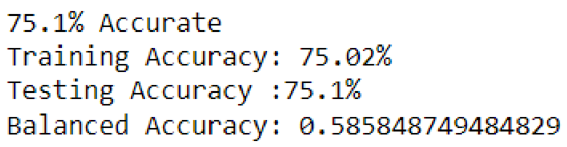
```
75.1% Accurate
Training Accuracy: 75.02%
Testing Accuracy :75.1%
Balanced Accuracy: 0.585848749484829
```

Figure 5.3.7.25: Code snippet for training and testing of the second XGBoost model.



```
              precision    recall  f1-score   support

       False       0.36      0.31      0.33     35495
        True       0.83      0.86      0.85    141970

    accuracy                           0.75    177465
   macro avg       0.60      0.59      0.59    177465
weighted avg       0.74      0.75      0.74    177465
```

Figure 5.3.7.26: Screenshot of classification report of the second XGBoost model.



```
75.1% Accurate
Training Accuracy: 75.02%
Testing Accuracy :75.1%
Balanced Accuracy: 0.585848749484829
```

Figure 5.3.7.27: Screenshot of the training and testing accuracy of the second XGBoost model.

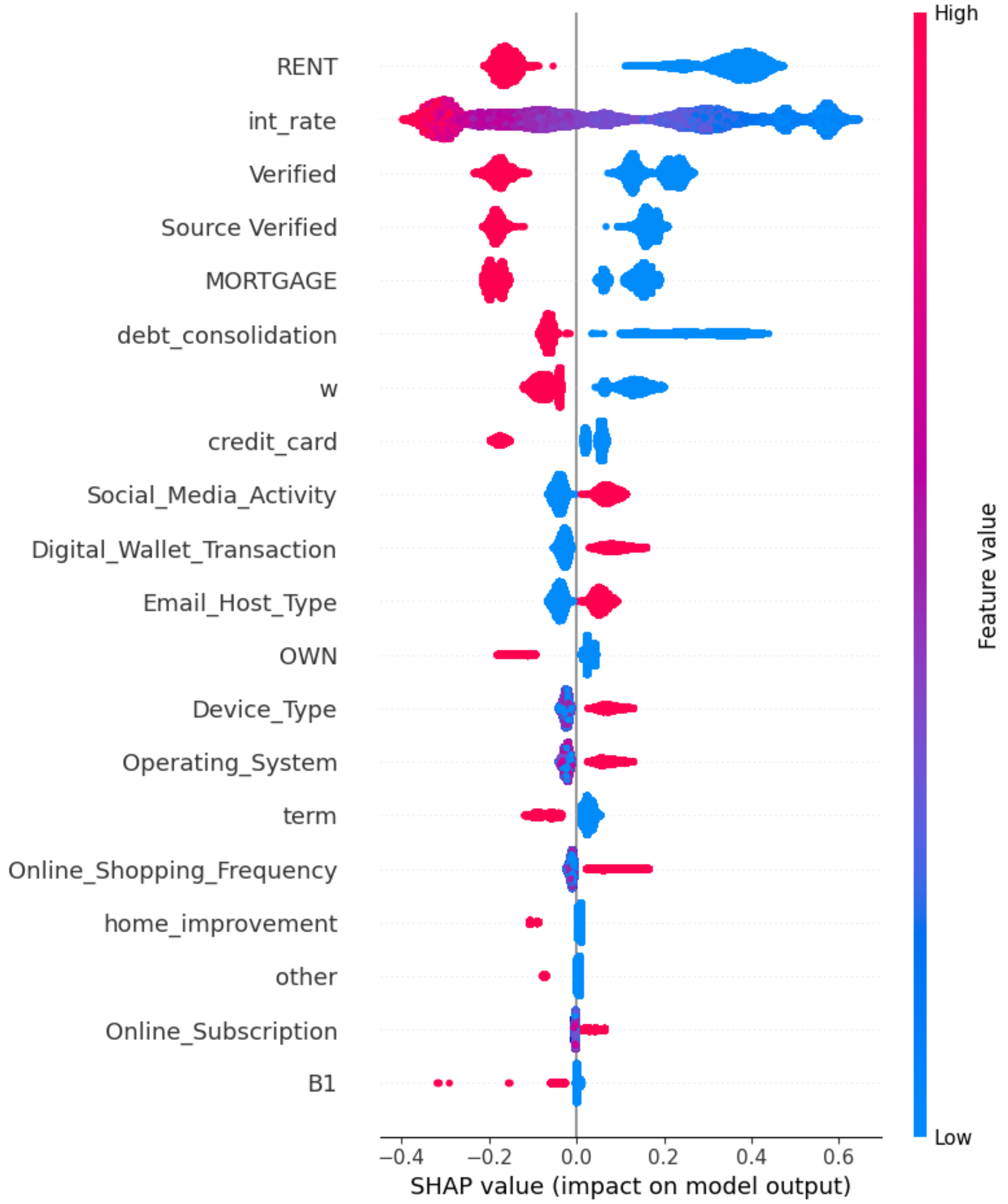
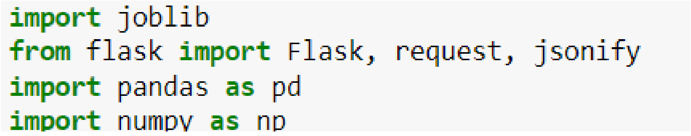


Figure 5.3.7.28: Graph of SHAP values of the second XGBoost model.

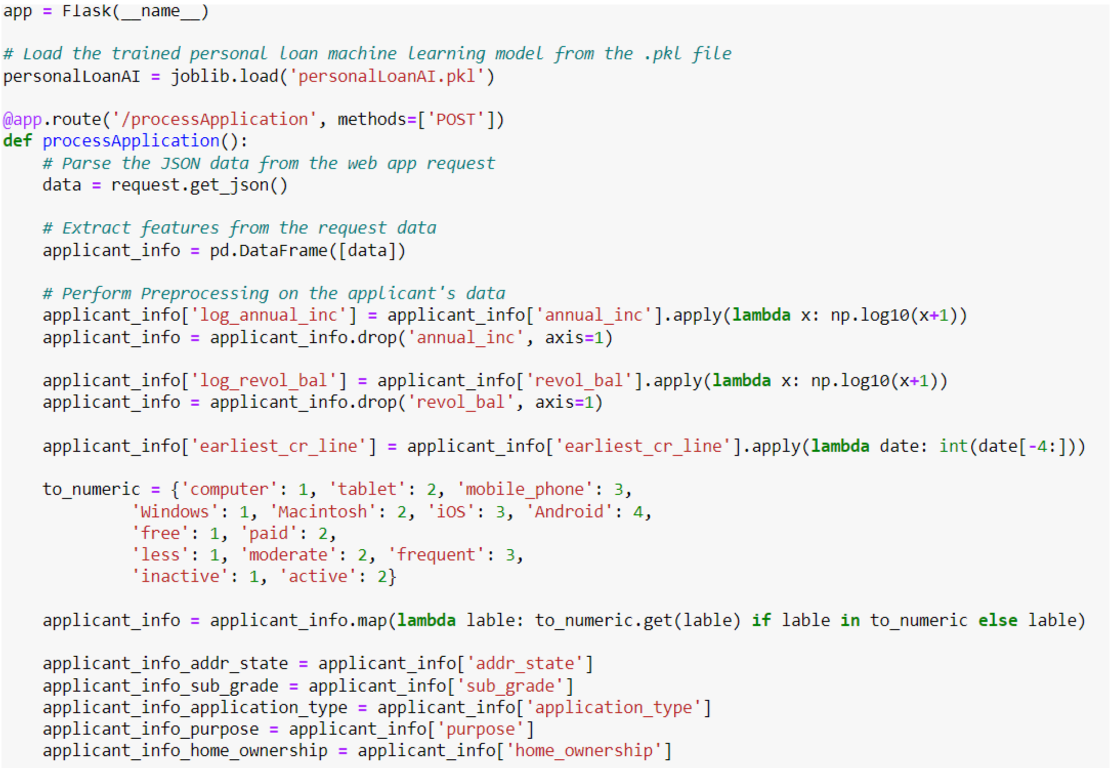
The result of the second XGBoost model can be seen in Figure 5.3.7.26 and Figure 5.3.7.27. The SHAP values of the second XGBoost model are interpreted in Figure 5.3.7.28. Finally, the XGBoost model was packaged into a pickle file.

5.3.8 Flask-based Microservice Setup



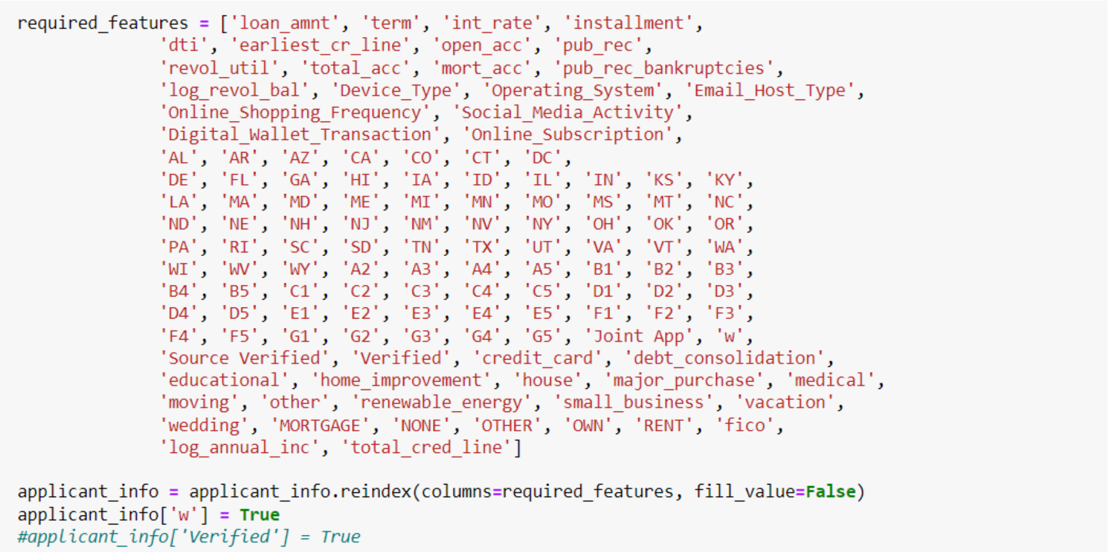
```
import joblib
from flask import Flask, request, jsonify
import pandas as pd
import numpy as np
```

Figure 5.3.8.1: Code snippet of required libraries for the Flask-based microservice.



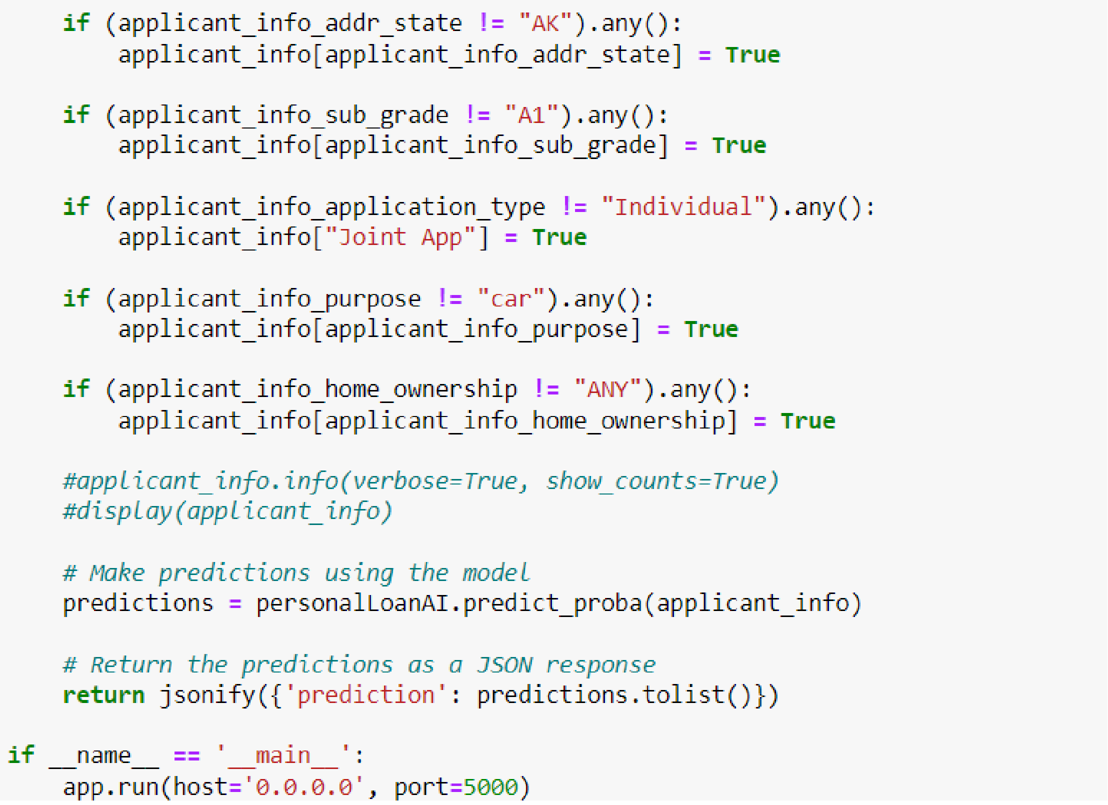
```
app = Flask(__name__)

# Load the trained personal loan machine learning model from the .pkl file
personalLoanAI = joblib.load('personalLoanAI.pkl')

@app.route('/processApplication', methods=['POST'])
def processApplication():
    # Parse the JSON data from the web app request
    data = request.get_json()

    # Extract features from the request data
    applicant_info = pd.DataFrame([data])

    # Perform Preprocessing on the applicant's data
    applicant_info['log_annual_inc'] = applicant_info['annual_inc'].apply(lambda x: np.log10(x+1))
    applicant_info = applicant_info.drop('annual_inc', axis=1)

    applicant_info['log_revol_bal'] = applicant_info['revol_bal'].apply(lambda x: np.log10(x+1))
    applicant_info = applicant_info.drop('revol_bal', axis=1)

    applicant_info['earliest_cr_line'] = applicant_info['earliest_cr_line'].apply(lambda date: int(date[-4:]))

    to_numeric = {'computer': 1, 'tablet': 2, 'mobile_phone': 3,
             'Windows': 1, 'Macintosh': 2, 'iOS': 3, 'Android': 4,
             'free': 1, 'paid': 2,
             'less': 1, 'moderate': 2, 'frequent': 3,
             'inactive': 1, 'active': 2}

    applicant_info = applicant_info.map(lambda lable: to_numeric.get(lable) if lable in to_numeric else lable)

    applicant_info_addr_state = applicant_info['addr_state']
    applicant_info_sub_grade = applicant_info['sub_grade']
    applicant_info_application_type = applicant_info['application_type']
    applicant_info_purpose = applicant_info['purpose']
    applicant_info_home_ownership = applicant_info['home_ownership']
```

Figure 5.3.8.2: Code snippet of creating the Flask-based microservice (part 1).



```
required_features = ['loan_amnt', 'term', 'int_rate', 'installment',
                     'dti', 'earliest_cr_line', 'open_acc', 'pub_rec',
                     'revol_util', 'total_acc', 'mort_acc', 'pub_rec_bankruptcies',
                     'log_revol_bal', 'Device_Type', 'Operating_System', 'Email_Host_Type',
                     'Online_Shopping_Frequency', 'Social_Media_Activity',
                     'Digital_Wallet_Transaction', 'Online_Subscription',
                     'AL', 'AR', 'AZ', 'CA', 'CO', 'CT', 'DC',
                     'DE', 'FL', 'GA', 'HI', 'IA', 'ID', 'IL', 'IN', 'KS', 'KY',
                     'LA', 'MA', 'MD', 'ME', 'MI', 'MN', 'MO', 'MS', 'MT', 'NC',
                     'ND', 'NE', 'NH', 'NJ', 'NM', 'NV', 'NY', 'OH', 'OK', 'OR',
                     'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VA', 'VT', 'WA',
                     'WI', 'WV', 'WY', 'A2', 'A3', 'A4', 'A5', 'B1', 'B2', 'B3',
                     'B4', 'B5', 'C1', 'C2', 'C3', 'C4', 'C5', 'D1', 'D2', 'D3',
                     'D4', 'D5', 'E1', 'E2', 'E3', 'E4', 'E5', 'F1', 'F2', 'F3',
                     'F4', 'F5', 'G1', 'G2', 'G3', 'G4', 'G5', 'Joint App', 'w',
                     'Source Verified', 'Verified', 'credit_card', 'debt_consolidation',
                     'educational', 'home_improvement', 'house', 'major_purchase', 'medical',
                     'moving', 'other', 'renewable_energy', 'small_business', 'vacation',
                     'wedding', 'MORTGAGE', 'NONE', 'OTHER', 'OWN', 'RENT', 'fico',
                     'log_annual_inc', 'total_cred_line']

applicant_info = applicant_info.reindex(columns=required_features, fill_value=False)
applicant_info['w'] = True
#applicant_info['Verified'] = True
```

Figure 5.3.8.3: Code snippet for creating the Flask-based microservice (part 2).



```
    if (applicant_info_addr_state != "AK").any():
        applicant_info[applicant_info_addr_state] = True

    if (applicant_info_sub_grade != "A1").any():
        applicant_info[applicant_info_sub_grade] = True

    if (applicant_info_application_type != "Individual").any():
        applicant_info["Joint App"] = True

    if (applicant_info_purpose != "car").any():
        applicant_info[applicant_info_purpose] = True

    if (applicant_info_home_ownership != "ANY").any():
        applicant_info[applicant_info_home_ownership] = True

    #applicant_info.info(verbose=True, show_counts=True)
    #display(applicant_info)

    # Make predictions using the model
    predictions = personalLoanAI.predict_proba(applicant_info)

    # Return the predictions as a JSON response
    return jsonify({'prediction': predictions.tolist()})

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

Figure 5.3.8.4: Code snippet for creating the Flask-based microservice (part 3).

The microservice was created to host the personal loan prediction model. All the received data from the servlet would undergone another pre-processing stage to ensure the final format of the input matched the expected format of the model. Once the model had predicted the loan application, the result was result back as a Json response to the servlet.

5.4 System Operation (with Screenshot)

5.4.1 Registration

When the user browses the personal loan web application, they will always land at the home page first which can be seen in Figure 5.4.1.1.

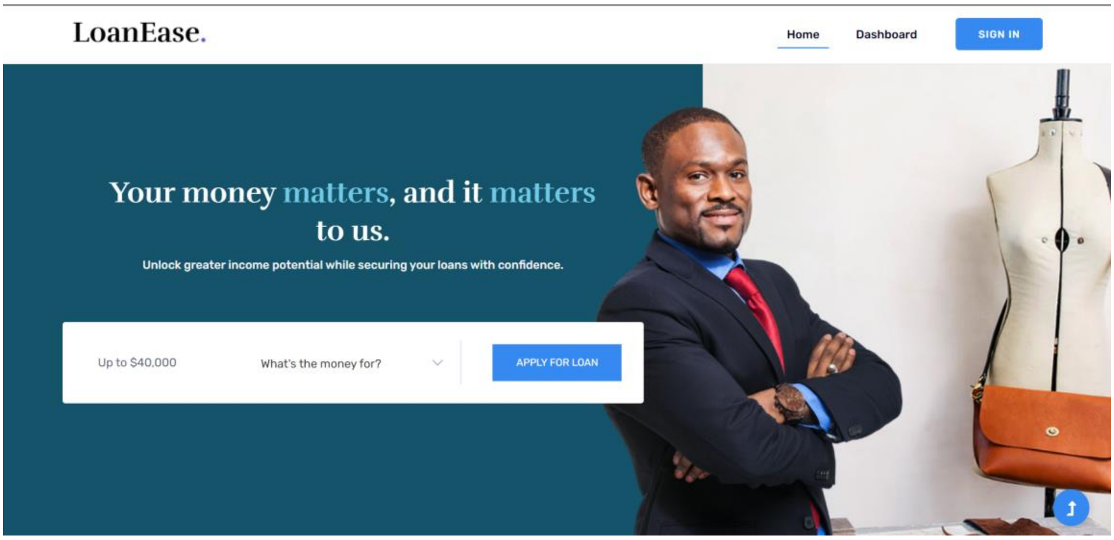

LoanEase.
Home
Dashboard
SIGN IN
Your money matters, and it matters to us.
Unlock greater income potential while securing your loans with confidence.
Up to $40,000
What's the money for?
APPLY FOR LOAN

Figure 5.4.1.1: Home page of the personal loan web application.

To register for an account, the user will have to click on the blue sign in button located at the top right of the page. Once done, the user will see the member sign-in page first.

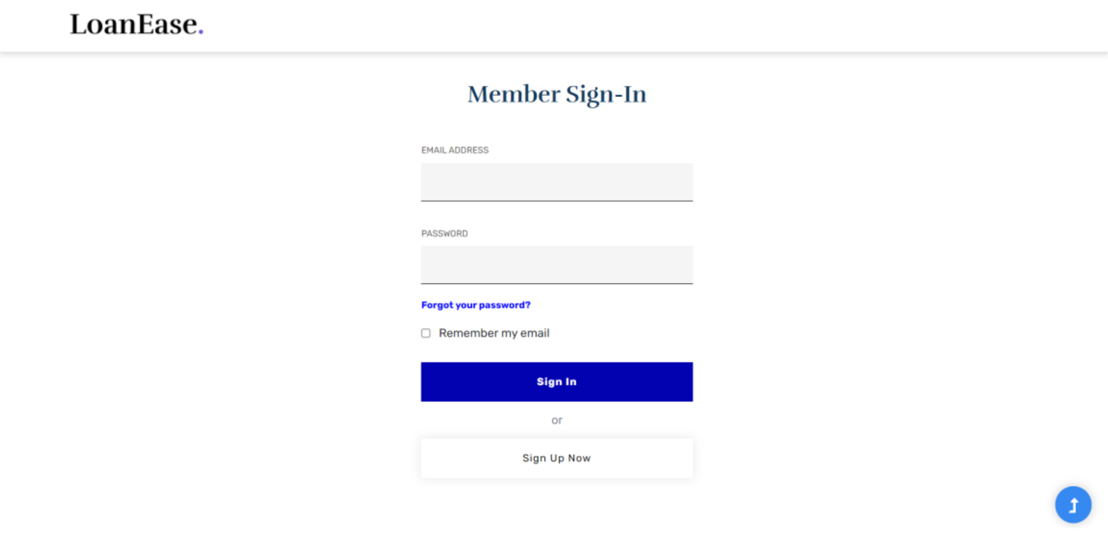


Figure 5.4.1.2: Sign in page of the personal loan web application.

At the sign-in page, the user will have to click on the "Sign Up Now" button to register for an account. The user then has to fill in the required information and click on the register button once done.

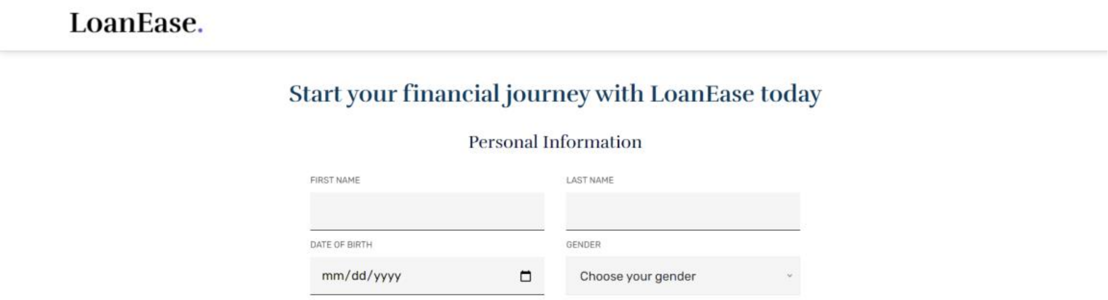


Figure 5.4.1.3: Sign up page of the personal loan web application (part 1).

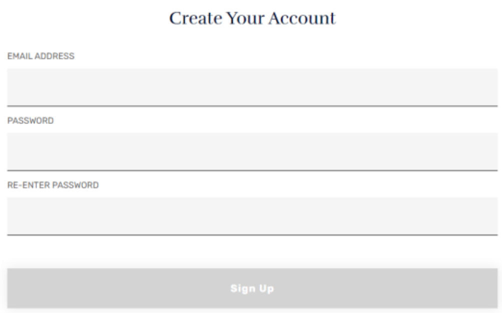
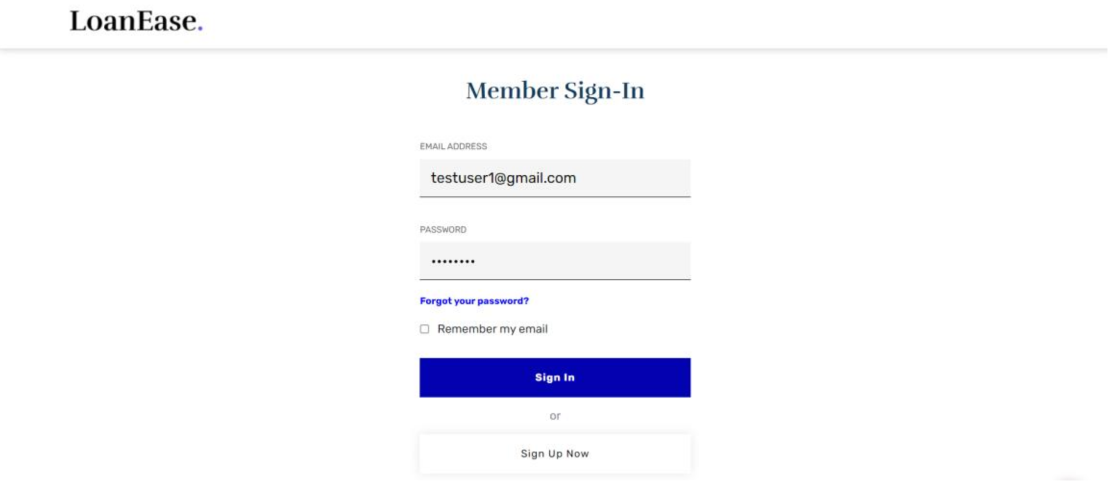


Figure 5.4.1.4: Sign up page of the personal loan web application (part 2).

Once the account registration is successful, the user will be redirected to the login page as shown in Figure 5.4.1.2.

5.4.2 Login



EMAIL ADDRESS

testuser1@gmail.com

PASSWORD

••••••••

Figure 5.4.2.1: Sign in page of the personal loan web application with credentials entered.

Once the user has a valid account, they can type in the valid login credentials. If they have been authorized by the authentication servlet, then they will be redirected back to the index page as shown in Figure 5.4.3.1.

5.4.3 Logout

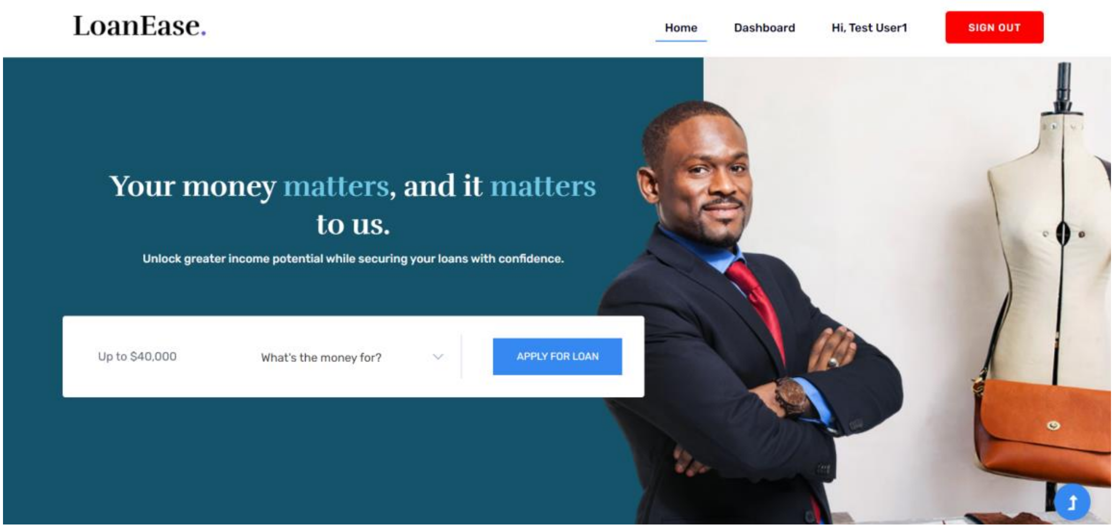
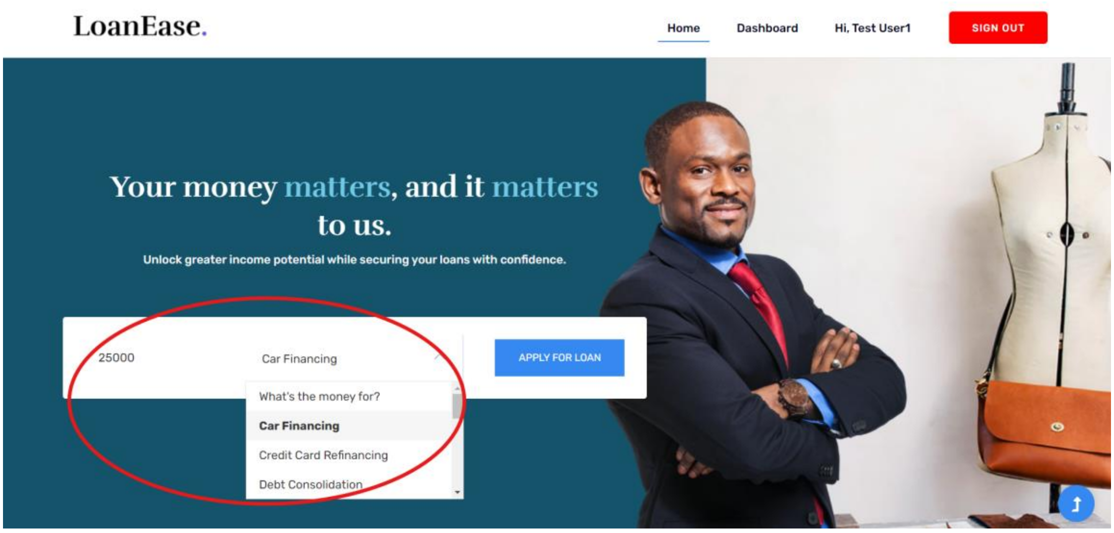

LoanEase.
Home
Dashboard
Hi, Test User1
SIGN OUT
Your money matters, and it matters to us.
Unlock greater income potential while securing your loans with confidence.
Up to $40,000
What's the money for?
APPLY FOR LOAN

Figure 5.4.3.1: Home page of the personal loan web application after the user has signed in.

If the user wishes to sign out of their account, they can click on the red sign out button located at the top right of the page. Once signed out, the user will be redirected back to the login page as shown in Figure 5.4.1.2.

5.4.4 Loan Application



Your money matters, and it matters to us.

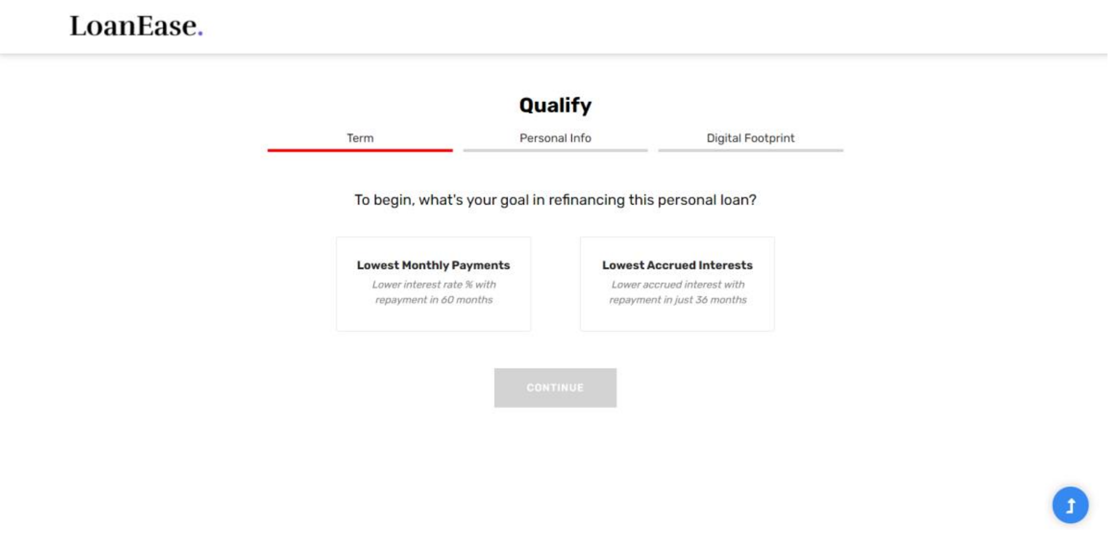
Unlock greater income potential while securing your loans with confidence.

25000

Car Financing

Figure 5.4.4.1: Home page of the personal loan web application with filled out loan amount and loan type.

If the user wants to apply for a personal loan, they can go to the index page and fill out the loan amount field and pick a loan purpose. Once both of the fields are done, then they can click on the apply for loan button.



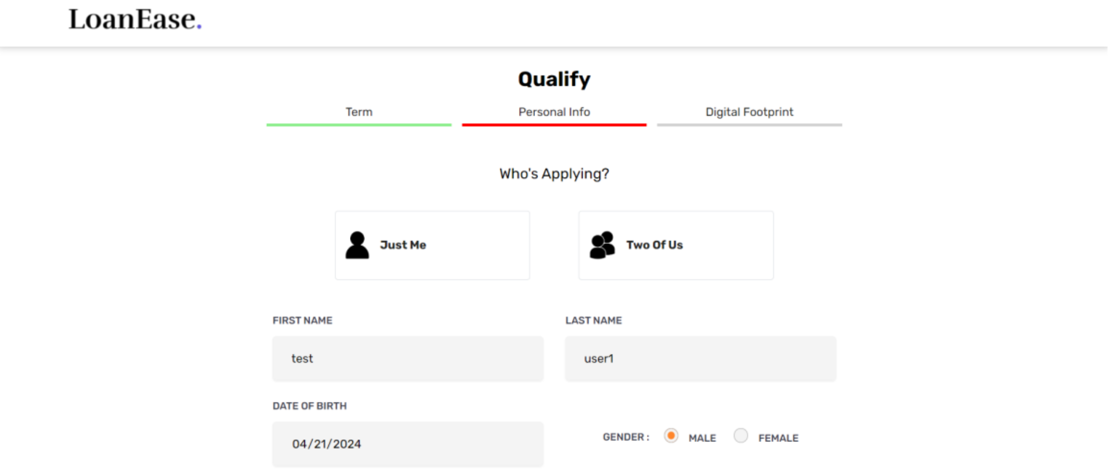
To begin, what's your goal in refinancing this personal loan?

Lowest Monthly Payments
Lower interest rate % with repayment in 60 months

Lowest Accrued Interests
Lower accrued interest with repayment in just 36 months

Figure 5.4.4.2: Term of the loan application

In the first section of the loan application, the user has to choose either to pay back the borrowed funds in 60 months or in 32 months. After that, they can proceed by clicking on the continue button.



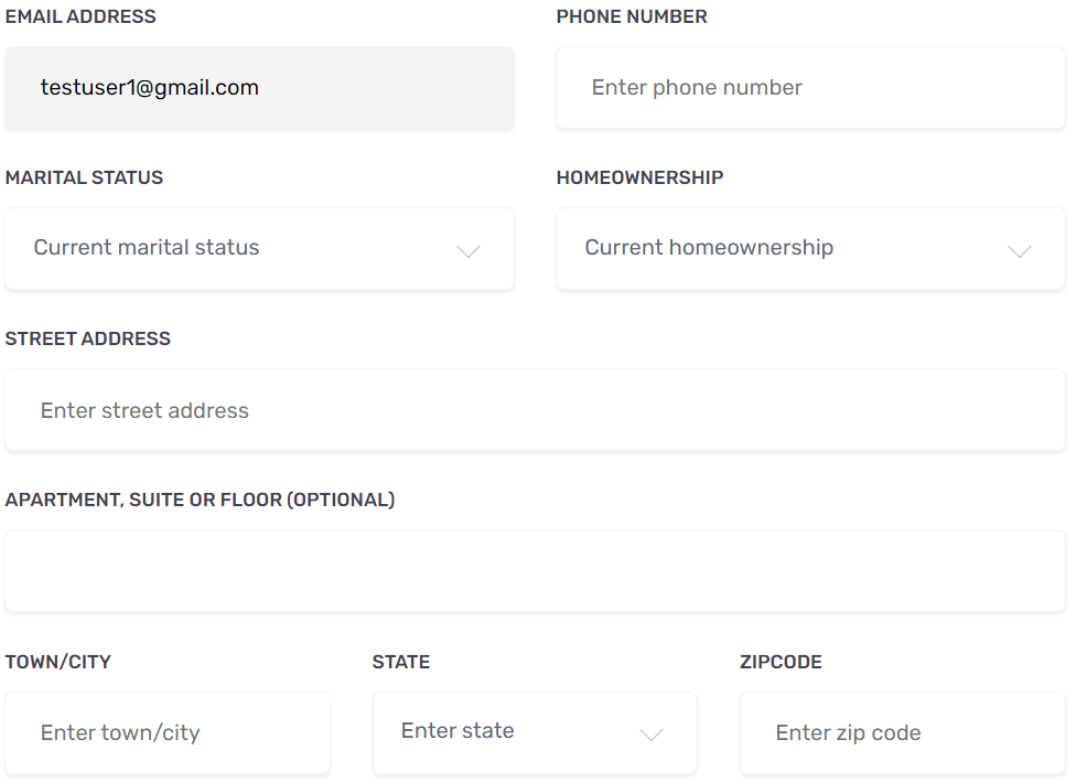
FIRST NAME: test

LAST NAME: user1

DATE OF BIRTH: 04/21/2024

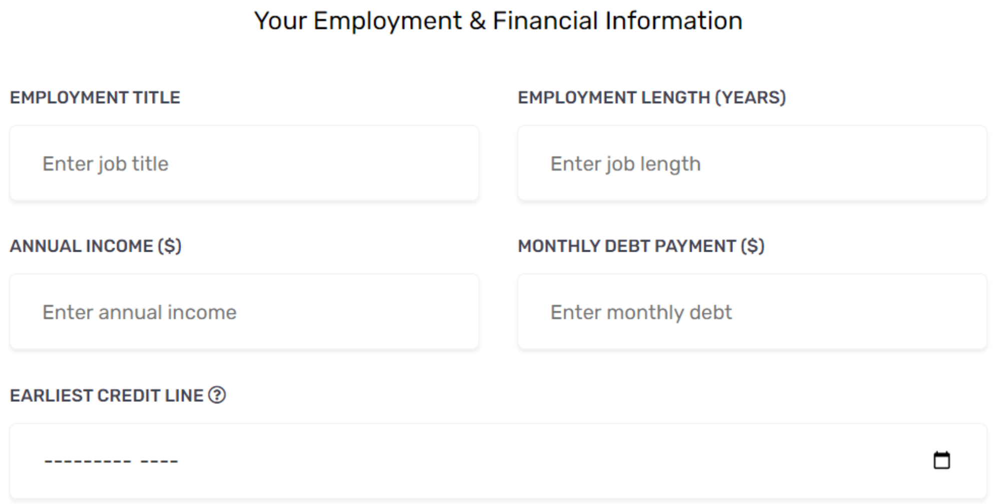
GENDER : ● MALE ○ FEMALE

Figure 5.4.4.3: Personal info of the loan application (part 1).



EMAIL ADDRESS: testuser1@gmail.com

Figure 5.4.4.4: Personal info of the loan application (part 2).



Your Employment & Financial Information

EMPLOYMENT TITLE: Enter job title

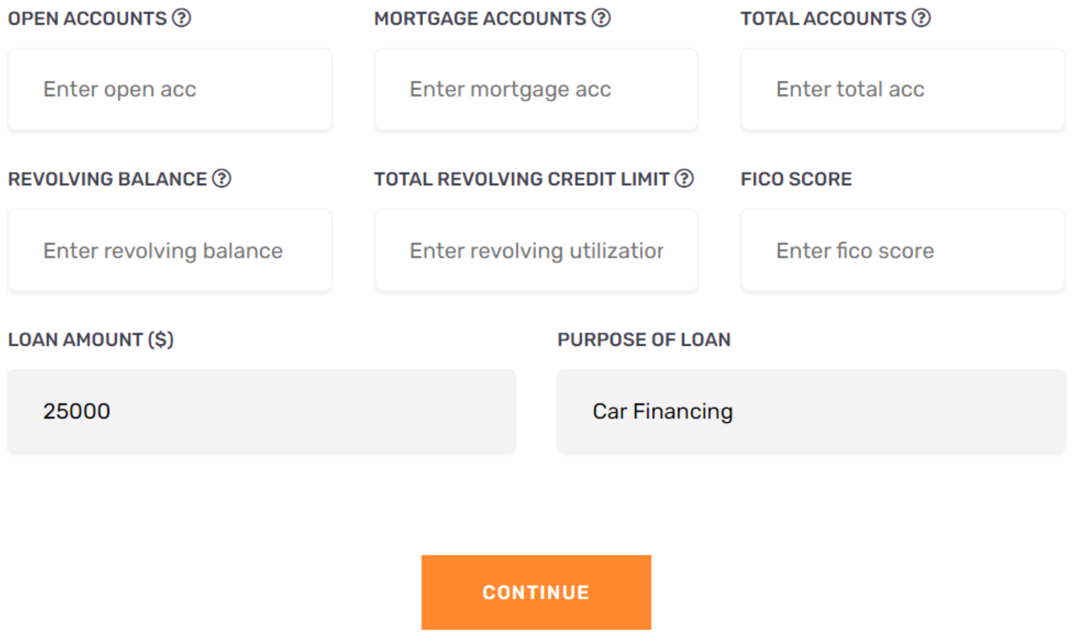
EMPLOYMENT LENGTH (YEARS): Enter job length

ANNUAL INCOME ($): Enter annual income

MONTHLY DEBT PAYMENT ($): Enter monthly debt

EARLIEST CREDIT LINE ⓘ: --------- ----

Figure 5.4.4.5: Personal info of the loan application (part 3).



OPEN ACCOUNTS ⓘ: Enter open acc

MORTGAGE ACCOUNTS ⓘ: Enter mortgage acc

TOTAL ACCOUNTS ⓘ: Enter total acc

REVOLVING BALANCE ⓘ: Enter revolving balance

TOTAL REVOLVING CREDIT LIMIT ⓘ: Enter revolving utilizatior

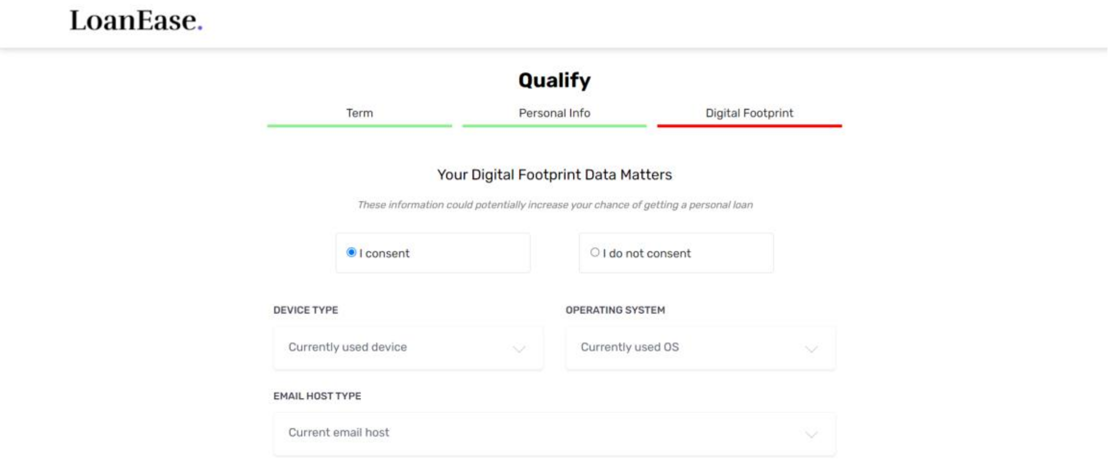
FICO SCORE: Enter fico score

LOAN AMOUNT ($): 25000

PURPOSE OF LOAN: Car Financing

Figure 5.4.4.6: Personal info of the loan application (part 4).

Once the user has done filling out the necessary personal and financial information, they can click on the continue button and proceed to the final step of the loan application.



LoanEase.

Qualify

Term | Personal Info | Digital Footprint

Your Digital Footprint Data Matters

These information could potentially increase your chance of getting a personal loan

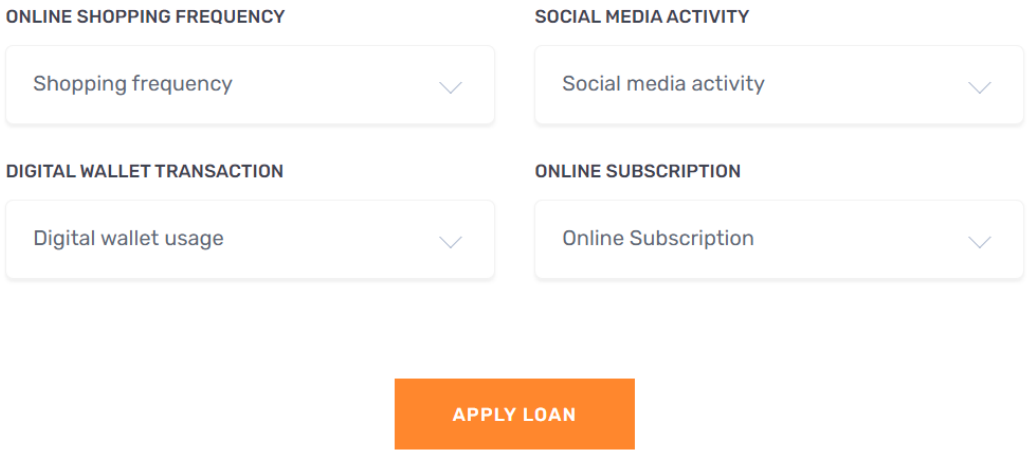
● I consent ○ I do not consent

DEVICE TYPE: Currently used device

OPERATING SYSTEM: Currently used OS

EMAIL HOST TYPE: Current email host

Figure 5.4.4.7: Digital footprint of the loan application if consented (part 1).



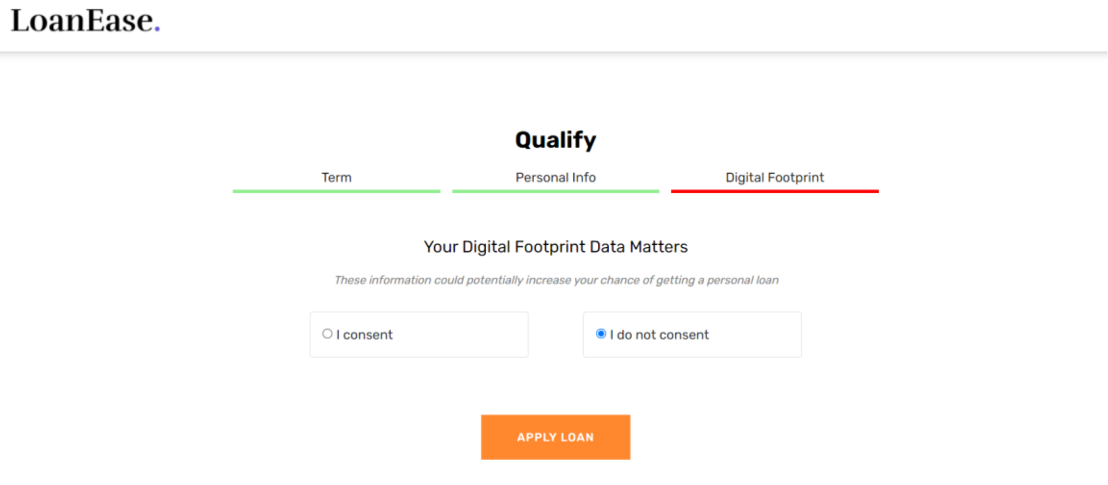
ONLINE SHOPPING FREQUENCY: Shopping frequency

SOCIAL MEDIA ACTIVITY: Social media activity

DIGITAL WALLET TRANSACTION: Digital wallet usage

ONLINE SUBSCRIPTION: Online Subscription

Figure 5.4.4.8: Digital footprint of the loan application if consented (part 2).



LoanEase.

Qualify

Term | Personal Info | Digital Footprint

Your Digital Footprint Data Matters

These information could potentially increase your chance of getting a personal loan

○ I consent

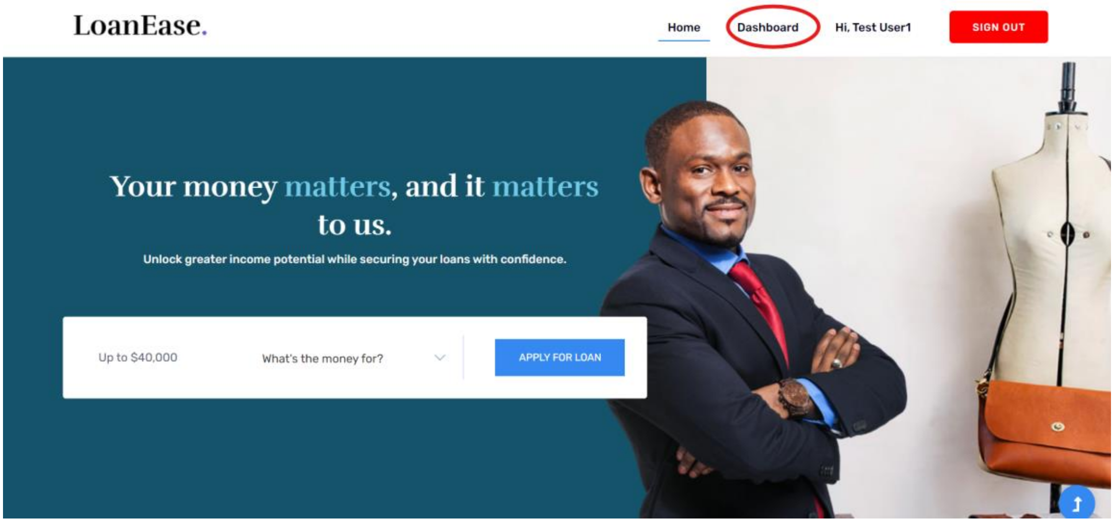
● I do not consent

APPLY LOAN

Figure 5.4.4.9: Digital footprint of the loan application if not consented.

In the final section of the loan application, the user has the option to either consent their digital footprint data be used to assist the prediction of the loan application or not consent to. If consented, the user has to fill out necessary information as shown in Figure 5.4.4.7 and Figure 5.4.4.8. If not consented, the user can just click on the apply loan button and wait for the AI model to make prediction.

5.4.5 Dashboard



LoanEase.

Home | Dashboard | Hi, Test User1 | SIGN OUT

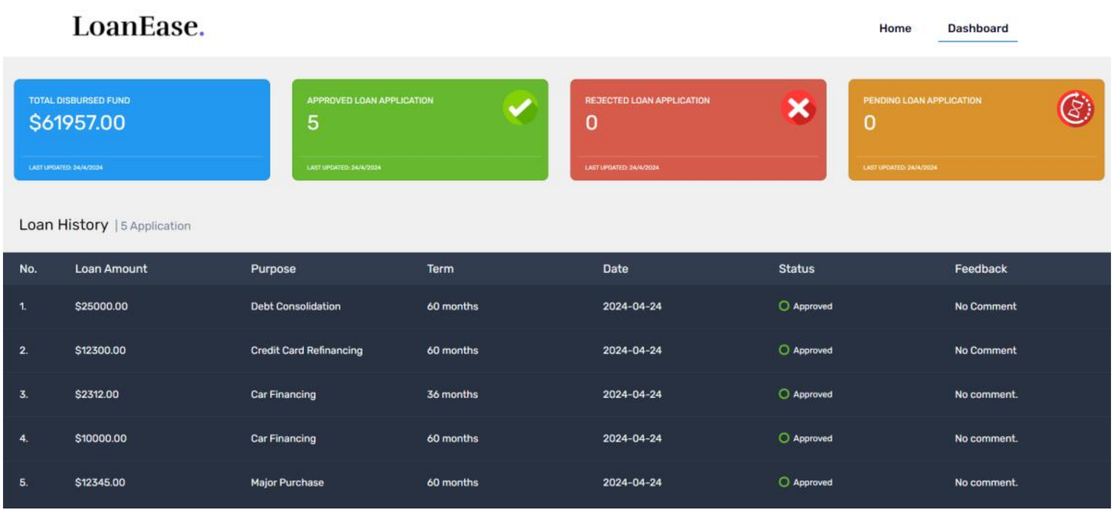
Your money matters, and it matters to us.

Unlock greater income potential while securing your loans with confidence.

Up to $40,000 | What's the money for? | APPLY FOR LOAN

Figure 5.4.5.1: Home page of the web application with dashboard menu highlighted.

If the user wants to see their loan application history, they can click on the Dashboard located in the navigation bar.



TOTAL DISBURSED FUND
$61957.00
LAST UPDATED: 24/4/2024

APPROVED LOAN APPLICATION
5
LAST UPDATED: 24/4/2024

REJECTED LOAN APPLICATION
0
LAST UPDATED: 24/4/2024

PENDING LOAN APPLICATION
0
LAST UPDATED: 24/4/2024

Loan History | 5 Application

No.	Loan Amount	Purpose	Term	Date	Status	Feedback
1.	$25000.00	Debt Consolidation	60 months	2024-04-24	○ Approved	No Comment
2.	$12300.00	Credit Card Refinancing	60 months	2024-04-24	○ Approved	No Comment
3.	$2312.00	Car Financing	36 months	2024-04-24	○ Approved	No comment.
4.	$10000.00	Car Financing	60 months	2024-04-24	○ Approved	No comment.
5.	$12345.00	Major Purchase	60 months	2024-04-24	○ Approved	No comment.

Figure 5.4.5.2: Dashboard of the user.

Once the user has arrived at the dashboard view, they can look at the total amount of disbursed fund from approved loan application, total number of approved, rejected, and pending loan application. The history of previously applied loan application is shown at the lower part of the page.

5.5 Implementation Issues and Challenges

There were few issues and challenges faced during the implementation of this project. First of all, finding a loan prediction dataset online with digital footprint data is hard or almost impossible. Most of the dataset do not come with said data. To overcome this issue, the most logical way to achieve this is to generate synthetic digital footprint data. But this does come with another challenge. It generally requires domain knowledge and deep understanding on how these features could affect the final outcome of the loan approval process. The relationship of these features has to be defined from a set of rules and assumptions. Not only that, as this project does not have access to expert domain knowledge, the rules and assumptions can only be derived from studying various sources online which then tries to simulate the real-world conditions as closely as possible. Moreover, during the training of the XGBoost model, the altered dataset with the synthetic digital footprint data seemed to perform slightly worse than the original dataset but with only 3% difference in accuracy. This might be due to the fact the dataset had millions of instances and hundreds of features which made the relationship between features incredibly hard to interpret without any domain or expert knowledge at hand.

However, the approval rate seemed to be slightly better than the original XGBoost model. Ideally, the historical data such as this dataset should have come with the digital footprint data already in it but as it seemed to be extremely hard to find any publicly available quality loan prediction dataset that contained digital footprint data due to major privacy concerns.

5.6 Concluding Remark

In summary, the implementation of the personal loan AI with the web application involved the use of MVC framework, Flask-based framework, and also the use of machine learning libraries from Python to create a robust personal loan web application. The XGBoost-based personal loan prediction model can accurately assess the borrower’s creditworthiness together with their digital footprint data. The final output is then presented to the dashboard of the user.

CHAPTER 6 System Evaluation and Discussion

6.1 System Testing, Testing Setup and Results

6.1.1 Personal Loan Machine Learning Model Testing and Results

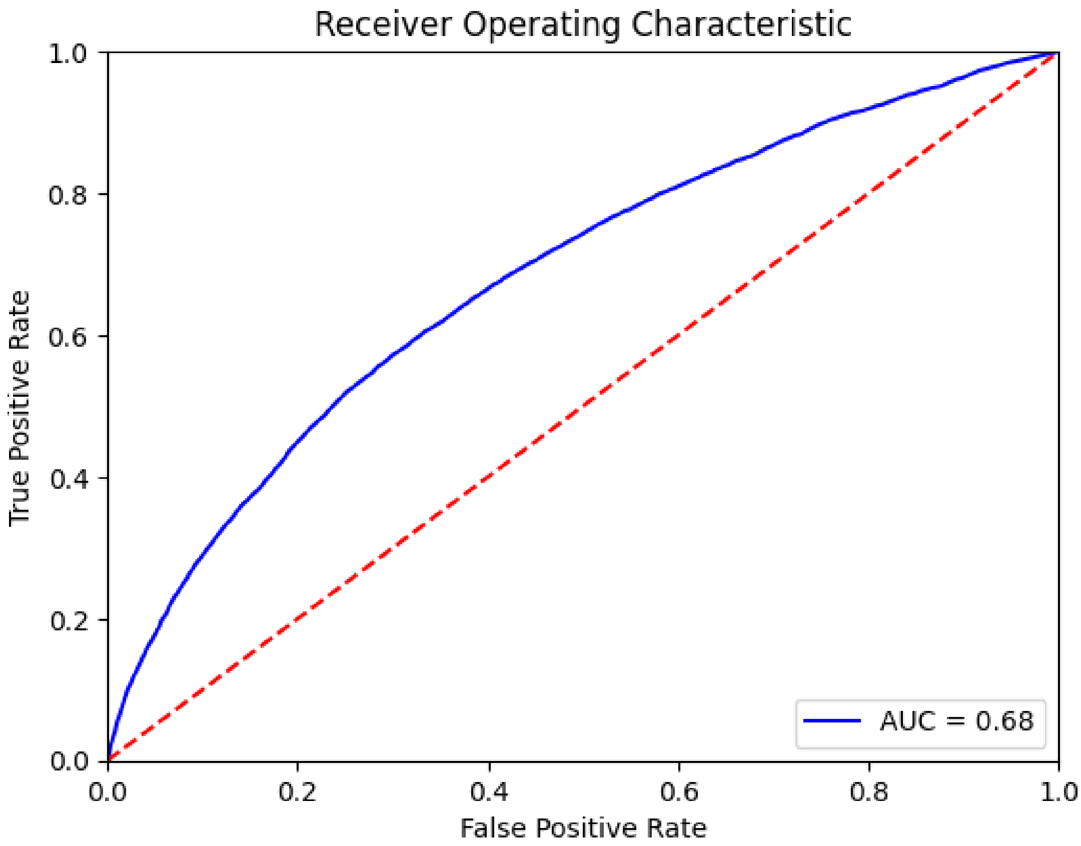


Figure 6.1.1.1: ROC-AUC graph of the first XGBoost model.

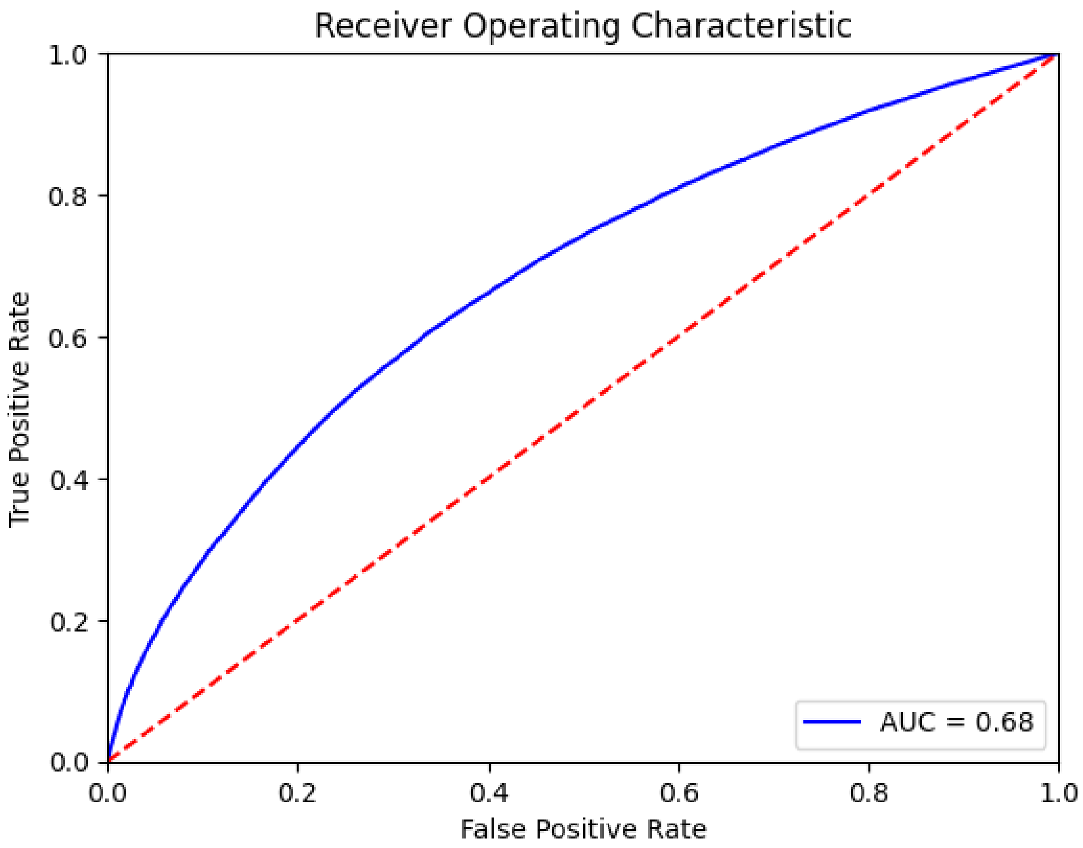
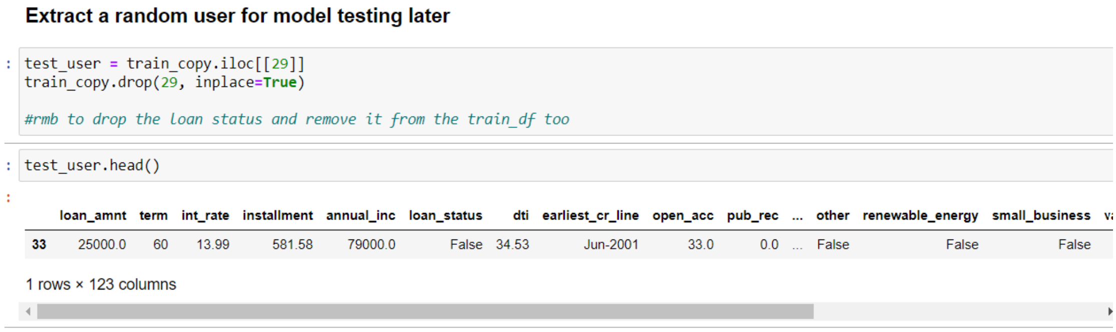


Figure 6.1.1.2: ROC-AUC graph of the second XGBoost model.

The AUC score of the first XGBoost model was 0.66 and the AUC score of the second XGBoost model was 0.66. Both models had the same ability to differentiate between the positive and negative classes as compared to the first XGBoost model. This might be due to the fact the there were underlying complex relationship among the many features in the dataset. Although both had the same AUC score, but these models were then tested on an unseen test data shown in Figure 6.1.1.3. The first XGBoost predicted that the test user had 56% of approval rate which can be seen in Figure 6.1.1.4. But after adding additional digital footprint data into the test user information, the approval rate was then increased to 60% which was shown in Figure 6.1.1.5. The digital footprint data information is shown in Figure 6.1.1.6.



Extract a random user for model testing later

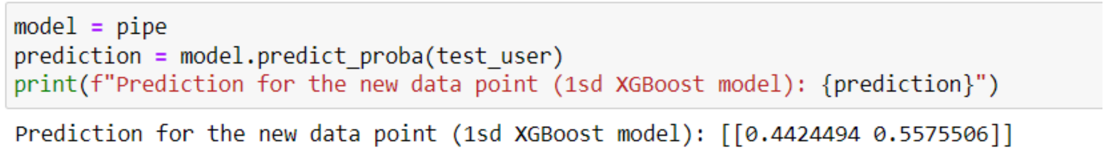
```
test_user = train_copy.iloc[[29]]
train_copy.drop(29, inplace=True)

#rmb to drop the loan status and remove it from the train_df too
```

```
test_user.head()
```

	loan_amnt	term	int_rate	installment	annual_inc	loan_status	dti	earliest_cr_line	open_acc	pub_rec	...	other	renewable_energy	small_business	va
33	25000.0	60	13.99	581.58	79000.0	False	34.53	Jun-2001	33.0	0.0	...	False	False	False	

1 rows × 123 columns

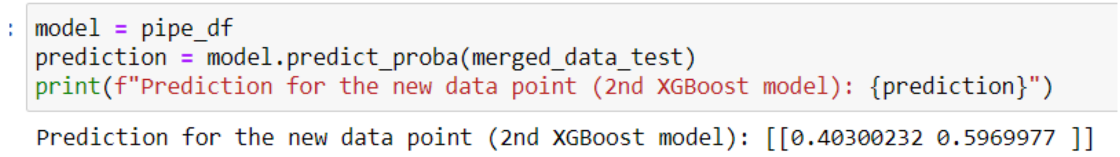
Figure 6.1.1.3: Sample test user extracted from the dataset.



```
model = pipe
prediction = model.predict_proba(test_user)
print(f"Prediction for the new data point (1sd XGBoost model): {prediction}")
```

```
Prediction for the new data point (1sd XGBoost model): [[0.4424494 0.5575506]]
```

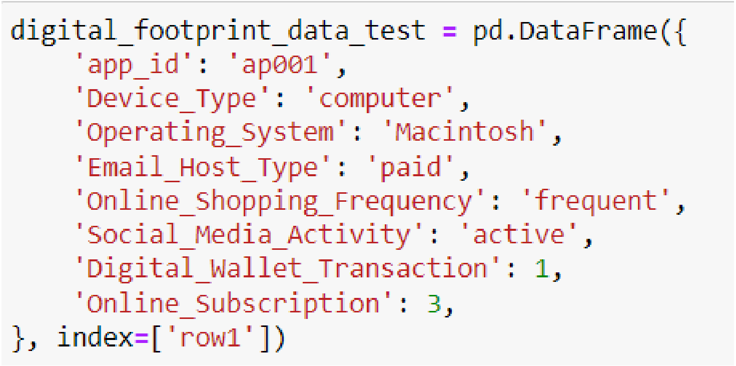
Figure 6.1.1.4: Prediction result made by the first XGBoost model on an unseen test user data.



```
model = pipe_df
prediction = model.predict_proba(merged_data_test)
print(f"Prediction for the new data point (2nd XGBoost model): {prediction}")
```

```
Prediction for the new data point (2nd XGBoost model): [[0.40300232 0.5969977 ]]
```

Figure 6.1.1.5: Prediction result made by the second XGBoost model after it has been altered with digital footprint data.



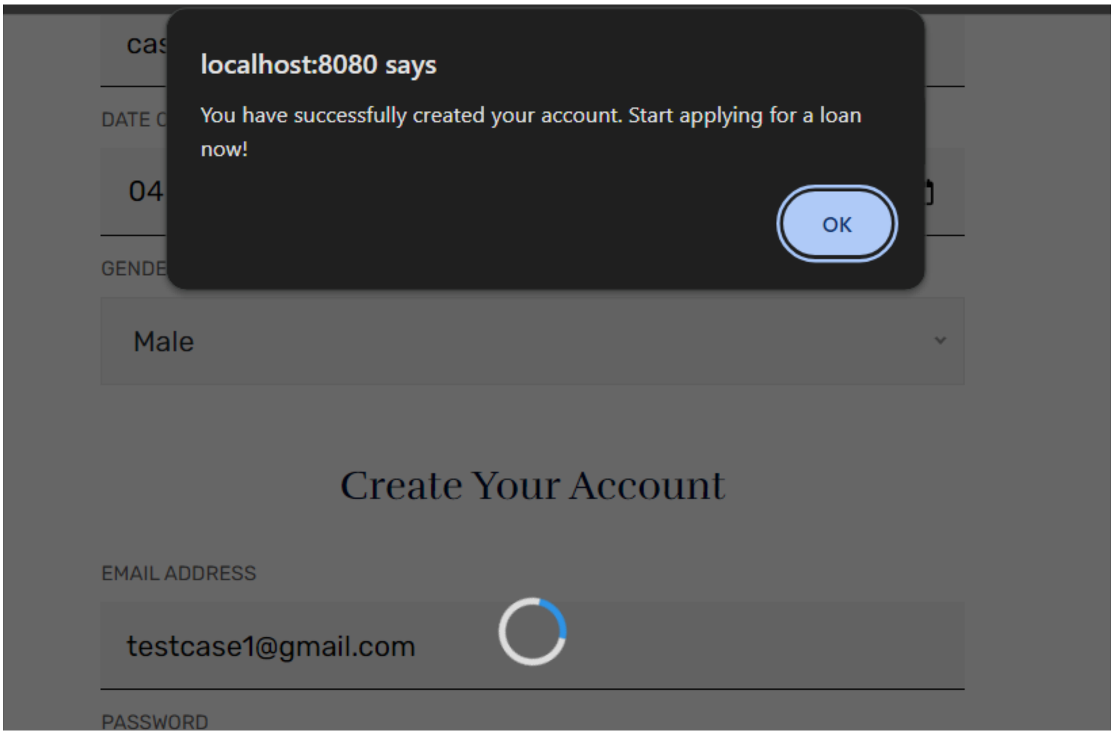
```
digital_footprint_data_test = pd.DataFrame({
    'app_id': 'ap001',
    'Device_Type': 'computer',
    'Operating_System': 'Macintosh',
    'Email_Host_Type': 'paid',
    'Online_Shopping_Frequency': 'frequent',
    'Social_Media_Activity': 'active',
    'Digital_Wallet_Transaction': 1,
    'Online_Subscription': 3,
}, index=['row1'])
```

Figure 6.1.1.6: Digital footprint data of the test user.

6.1.2 Web Application Testing and Results

Test Case 1.0 Registration Page

Test No.	1.1
Test Case	Verify if the user can register a valid new account.
Description	This test case is created to test the functionality of the registration controller.
Test Step	1. Navigate to the sign in page. 2. Click on the "Sign Up Now" button. 3. Fill up all of the required fields with valid information. 4. Click the Sign-Up button.
Expected Result	If all of the information is valid, then the account will be successfully registered to the database and an alert message will be popped up indicating the user has successfully registered.
Actual Result	An alert message box appeared, and the message was "You have successfully created your account. Start applying for a loan now!"
Status	**PASS**



localhost:8080 says

You have successfully created your account. Start applying for a loan now!

Figure 6.1.2.1: Alert box appears upon successful registration.

Test No.	1.2
Test Case	Verify that the user should not be able to register for an account when the re-entered password does not match the initial password
Description	This test case is created to test the functionality of the registration controller at error validation.
Test Step	1. Navigate to the sign in page. 2. Click on the “Sign Up Now” button. 3. Fill up the all the required information with valid information except for the password and re-enter password field. 4. For the password, type in “testuser” and the re-enter password is “testuser1” 5. Click the Sign-Up button.
Expected Result	If the password and the re-enter password do not match, the sign-up button will remain disabled, and the re-enter password field will be appeared red in color.

Actual Result	The sign-up button is disabled, and the re-enter password field have changed its color to red.
Status	**PASS**

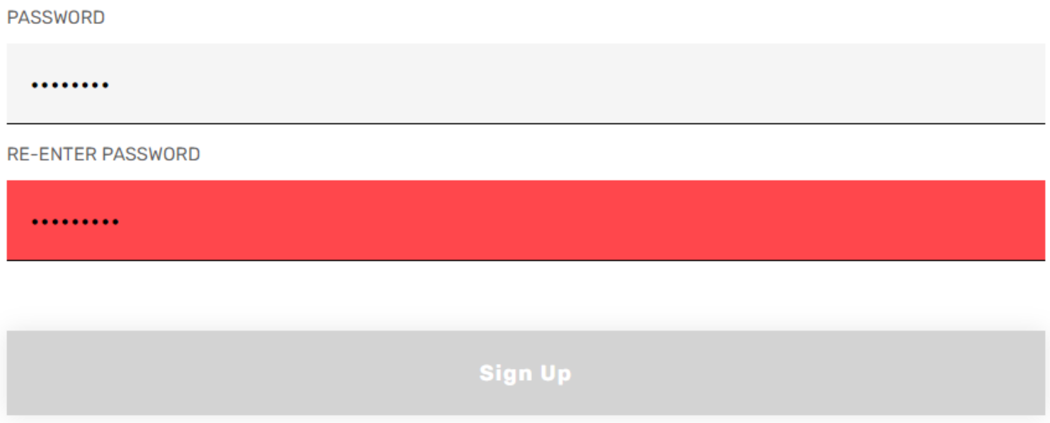
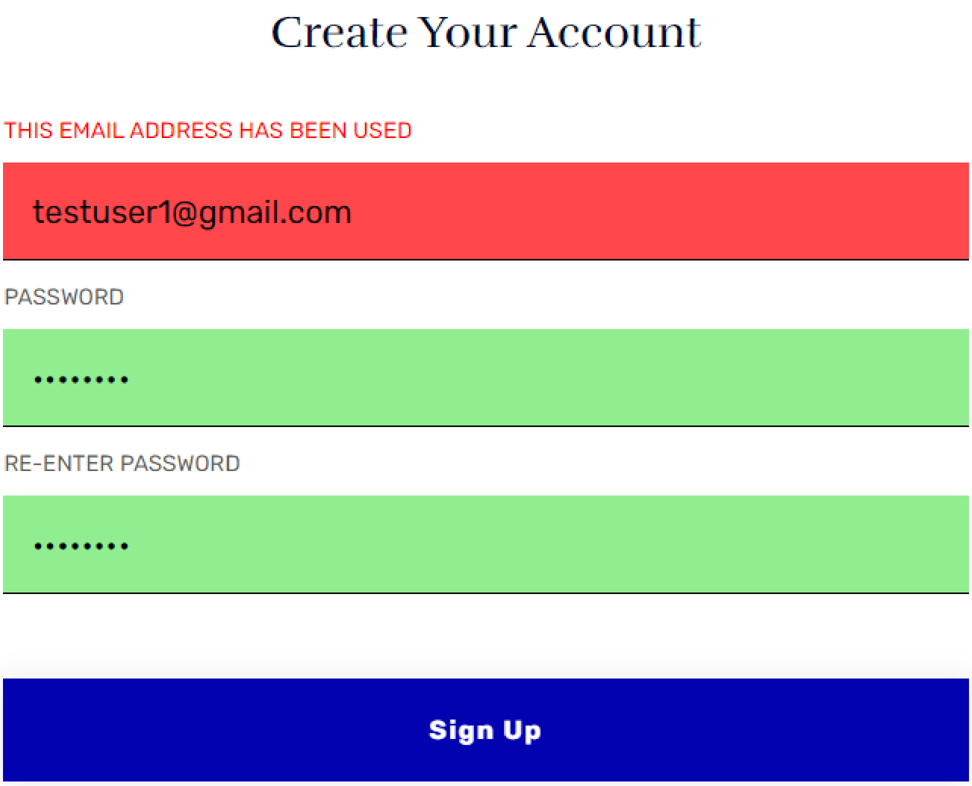


Figure 6.1.2.2: Mismatch password during registration.

Test No.	1.3
Test Case	Verify that the user should not be able to register for an account that has already existed in the database
Description	This test case is created to test the functionality of the registration controller at preventing the same email address from registering again, which may potentially cause some troubles in the future.
Test Step	1. Navigate to the sign in page. 2. Click on the "Sign Up Now" button. 3. Fill up the all the required information with valid information except for the email field. 4. For the email field, type in testuser1@gmail.com which has already existed in the database. 5. Click the Sign-Up button.
Expected Result	The error message will tell the user that the email address is already in use.

Actual Result	A red error message "THIS EMAIL HAS BEEN USED" did popped up on top of the input field.
Status	**PASS**



Create Your Account

THIS EMAIL ADDRESS HAS BEEN USED

testuser1@gmail.com

PASSWORD

••••••••

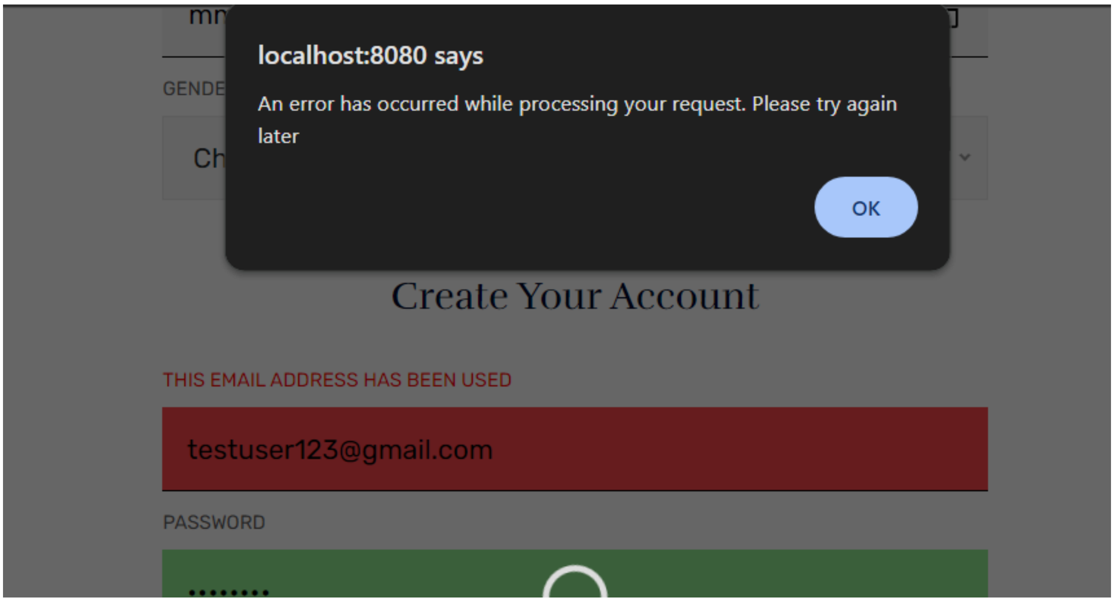
RE-ENTER PASSWORD

••••••••

Figure 6.1.2.3: Email address has been used during registration.

Test No.	1.4
Test Case	Verify that the user should not be able to register for an account if some of the input fields are left empty intentionally.
Description	This test case is created to test the functionality of the registration controller at preventing uploading null values to the database.
Test Step	1. Navigate to the sign in page. 2. Click on the "Sign Up Now" button. 3. Leave all the input fields blank. 4. Click the Sign-Up button.

Expected Result	The error alert message should pop up saying the there is an error while trying to process the request.
Actual Result	An alert message box appeared, and the message was "An error has occurred while processing your request. Please try again later."
Status	**PASS**



localhost:8080 says

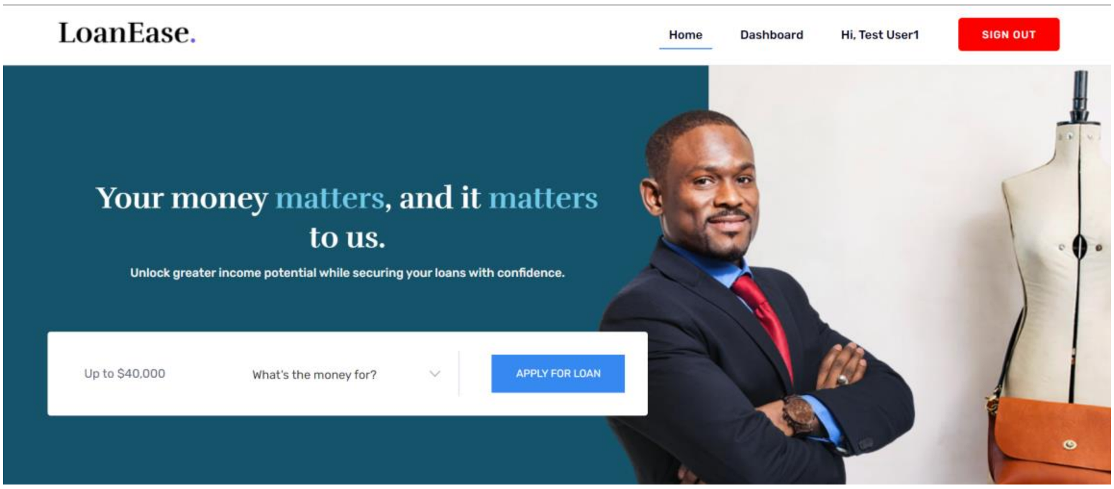
An error has occurred while processing your request. Please try again later

Figure 6.1.2.4: Error when field are left empty during registration.

Test Case 2.0 Login Page

Test No.	2.1
Test Case	Verify that the user should be able to login with valid credentials
Description	This test case is created to test the functionality of the authentication controller at authorizing only valid users.
Test Step	1. Navigate to the sign in page. 2. Email: testuser1@gmail.com Password: testuser 3. Click the sign in button

Expected Result	The user should be able to sign in without any problem and should see the index as the landing page.
Actual Result	The user was redirected to the login page. No error message was shown during signing in.
Status	**PASS**



LoanEase.

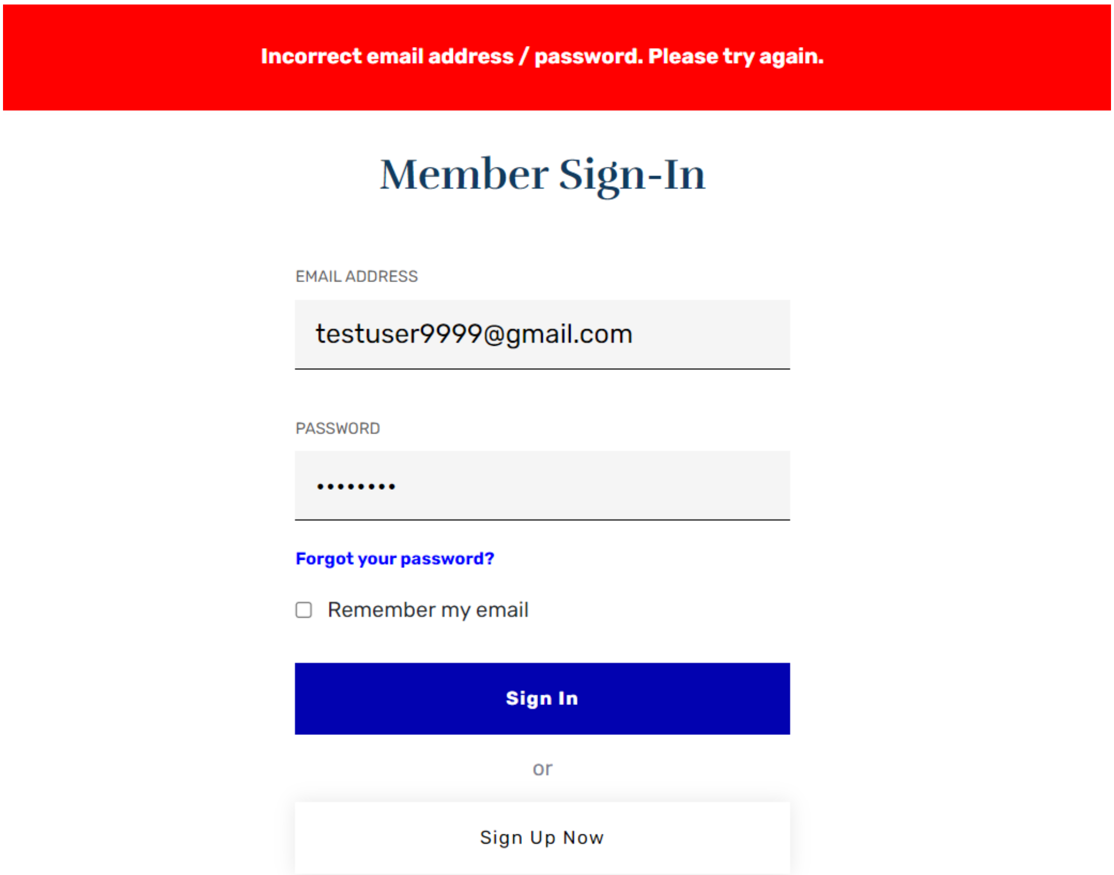
Your money matters, and it matters to us.

Unlock greater income potential while securing your loans with confidence.

Figure 6.1.2.5: Valid user login.

Test No.	2.2
Test Case	Verify that the user should not be able to log in with invalid credentials
Description	This test case is created to test the functionality of the authentication controller at handling invalid information.
Test Step	1. Navigate to the sign in page. 2. Email: testuser9999@gmail.com Password: testuser 3. Click the sign in button
Expected Result	The user should not be able to sign in and an error prompt will be shown to the user.
Actual Result	The user remained at the login page and a red error message was shown on the top. Its error message was "Incorrect email address / password. Please try again."

Status	**PASS**



Incorrect email address / password. Please try again.

Member Sign-In

EMAIL ADDRESS

testuser9999@gmail.com

PASSWORD

••••••••

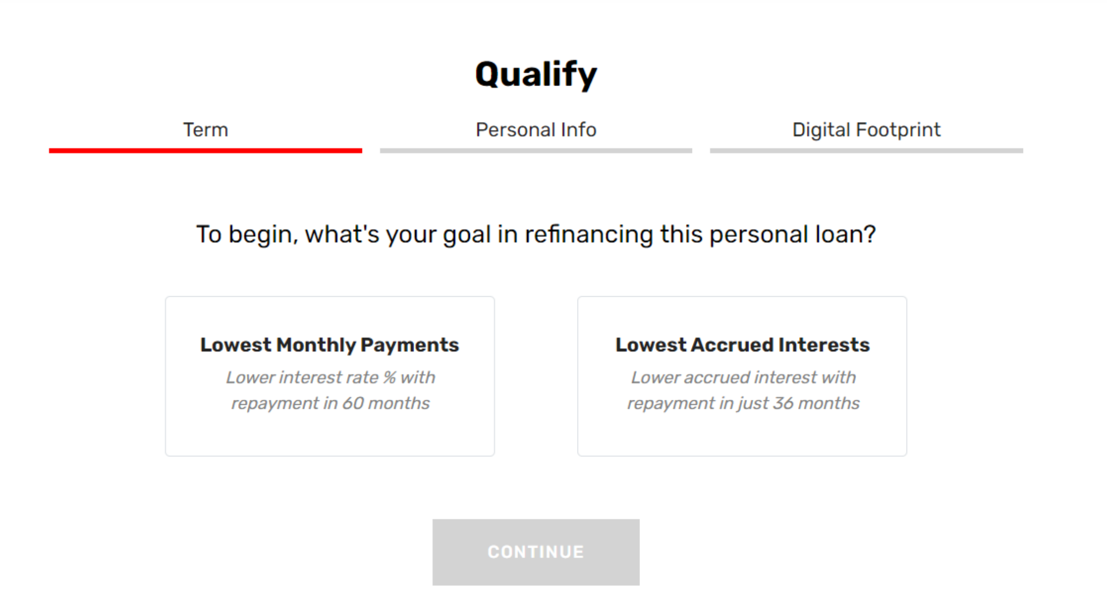
Figure 6.1.2.6: Invalid user login.

Test Case 3.0 Index page

Test No.	3.1
Test Case	Verify that the signed-in user should be able to log out from the index page.
Description	This test case is created to test the functionality of the authentication controller at invalidating the user session.
Test Step	1. Navigate to log in. 2. Sign in with email: testuser1@gmail.com Password: testuser 3. Click the sign out button on the top right of the page

Expected Result	The user should be able to log out and will be redirect back to the login page.
Actual Result	The user was redirected to the login page after clicking the sign out button
Status	**PASS**

Test No.	3.2
Test Case	Verify that the user should be able to start applying for a loan after inserting valid loan amount and also the type of loan.
Description	This test case is created to test the functionality of the loan application servlet at redirecting user to the application form.
Test Step	1. Navigate to log in. 2. Sign in with email: testuser1@gmail.com Password: testuser 3. Type in the loan amount that falls within the $1,000 and $40,000 range. 4. Choose any of the loan type. 5. Click the apply for loan button.
Expected Result	The user should be redirected to the loan application form where the first section is to choose the term of the loan.
Actual Result	The user was redirected to the login application form and prompted to choose either lowest monthly payments or lowest accrued interests.
Status	**PASS**



Qualify

Term | Personal Info | Digital Footprint

To begin, what's your goal in refinancing this personal loan?

Lowest Monthly Payments
Lower interest rate % with repayment in 60 months

Lowest Accrued Interests
Lower accrued interest with repayment in just 36 months

Figure 6.1.2.7: Term page.

Test No.	3.3
Test Case	Verify that the user should not be redirected to the loan application page when either the loan amount or the type of loan provided is incorrect.
Description	This test case is created to test the functionality of the index page at preventing incorrect values from being passed to the servlet.
Test Step	1. Navigate to log in. 2. Sign in with email: testuser1@gmail.com Password: testuser 3. Type in the loan amount that falls outside the $1,000 and $40,000 range. 4. Do not choose any loan type 5. Click the apply for loan button.
Expected Result	The user should not be redirected, and error messages will be shown.

Actual Result	The user remained at the same page and two error messages "Enter $1,000 to $40,000" and "Select a loan type" were shown respectively.
Status	**PASS**

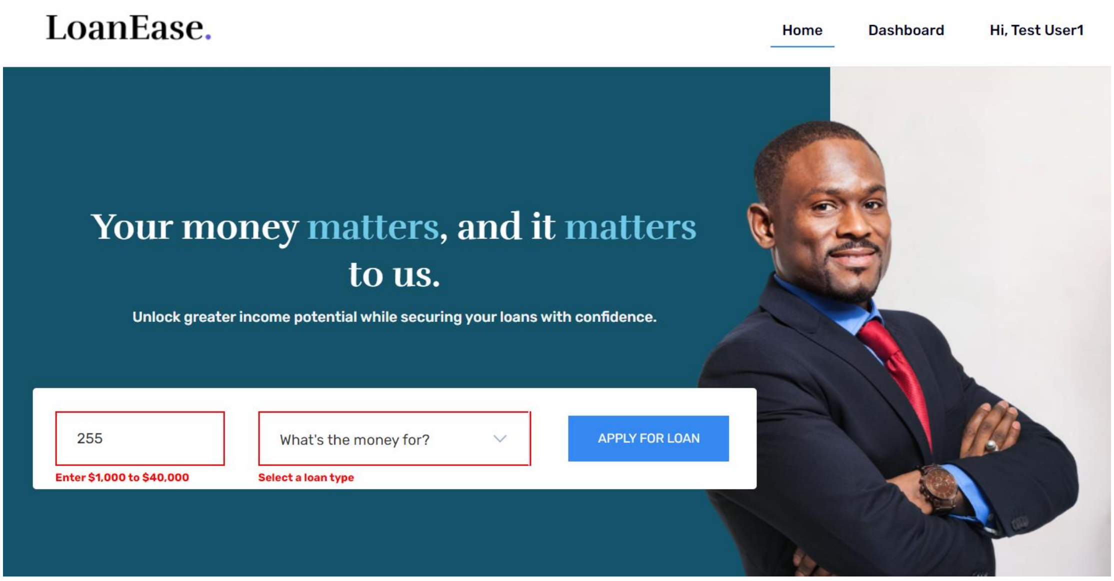
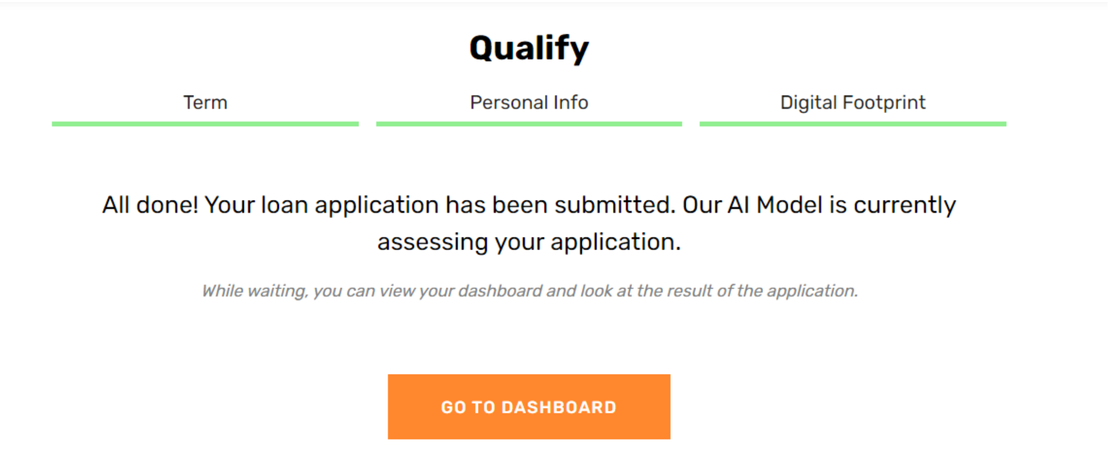

LoanEase.
Home
Dashboard
Hi, Test User1
Your money matters, and it matters to us.
Unlock greater income potential while securing your loans with confidence.
255
What's the money for?
APPLY FOR LOAN
Enter $1,000 to $40,000
Select a loan type

Figure 6.1.2.8: Error messages when the loan amount or the type of loan is incorrect.

Test Case 4.0 Apply Page

Test No.	4.1
Test Case	Verify that the user should be able to submit for a loan application with the correct personal and financial information.
Description	This test case is created to test the functionality of the process application servlet to send data to the microservice and retrieve back the prediction result.
Test Step	1. Navigate to log in. 2. Sign in with email: testuser1@gmail.com Password: testuser 3. Type in the loan amount that falls inside the $1,000 and $40,000 range. 4. Choose any type of loan. 5. Click the apply for loan button.

	6. Choose any term and click continue. 7. Fill in the personal and financial details and click continue. 8. Fill in the digital footprint data and submit the loan application.
Expected Result	The user should be prompted with a message indicating the loan application has been submitted and currently being processed by the prediction model.
Actual Result	The user was prompted with "All done! Your loan application has been submitted. Our AI model is currently assessing your application."
Status	**PASS**



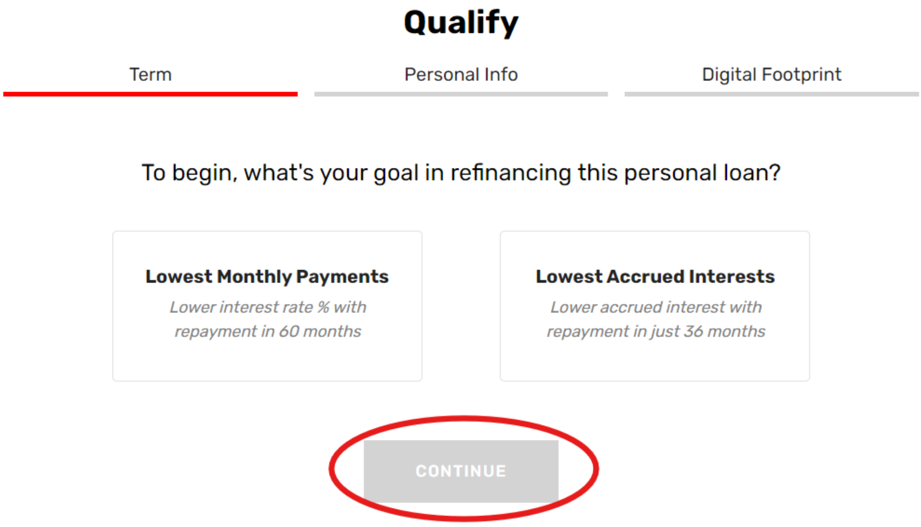
All done! Your loan application has been submitted. Our AI Model is currently assessing your application.

While waiting, you can view your dashboard and look at the result of the application.

Figure 6.1.2.9: Successfully submitted the loan application.

Test No.	4.2
Test Case	Verify that the user should not be able to continue the loan application without selecting a term.
Description	This test case is created to test the functionality of the process application servlet to prevent null value from submitting to the prediction model.
Test Step	1. Navigate to log in. 2. Sign in with email: testuser1@gmail.com

	Password: testuser 3. Type in the loan amount that falls within the $1,000 and $40,000 range. 4. Choose any type of loan. 5. Click the apply for loan button. 6. Leave the term selection blank. 7. Click the continue button.
Expected Result	The user should not be able to click the continue button as it is disabled due to none of the term is selected.
Actual Result	The user was not able to proceed with the loan application. The continue button was grayed out.
Status	**PASS**



To begin, what's your goal in refinancing this personal loan?

Lowest Monthly Payments

Lower interest rate % with repayment in 60 months

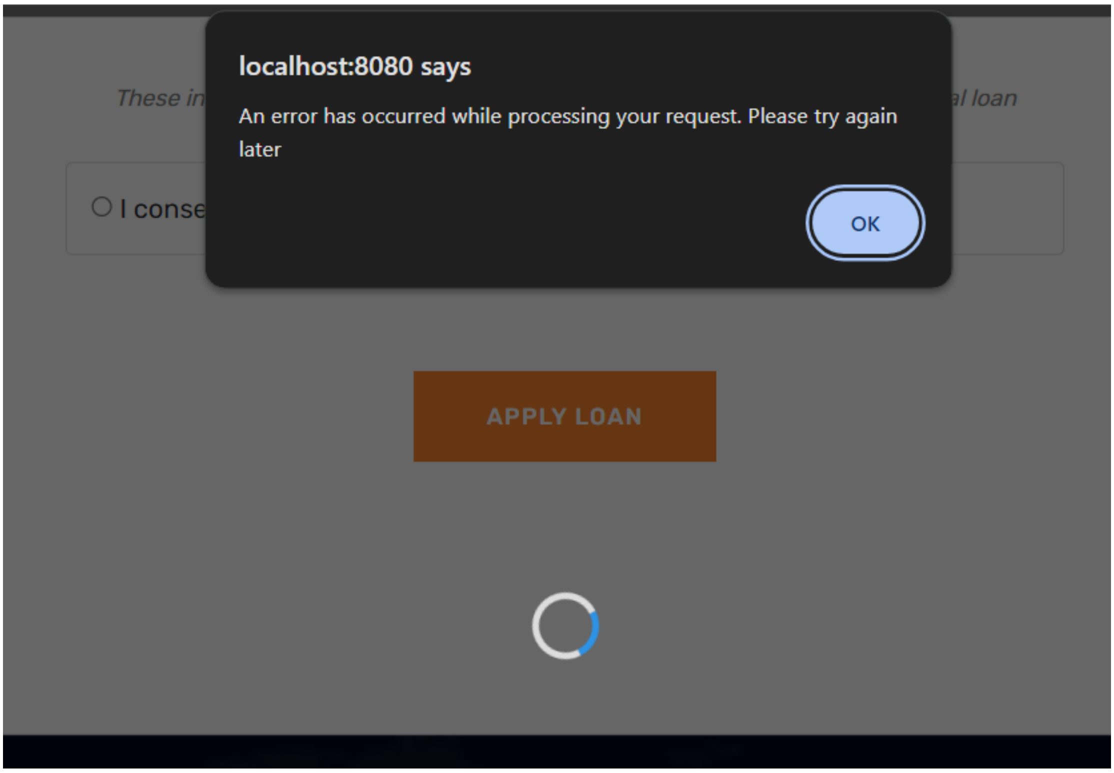
Lowest Accrued Interests

Lower accrued interest with repayment in just 36 months

Figure 6.1.2.10: Disabled continue button at the term page.

Test No.	4.3
Test Case	Verify that the user should not be able to submit the loan application with all of the input fields left empty.
Description	This test case is created to test the functionality of the process application servlet to prevent null value from submitting to the prediction model.
Test Step	1. Navigate to log in.

	2. Sign in with email: testuser1@gmail.com Password: testuser 3. Type in the loan amount that falls within the $1,000 and $40,000 range. 4. Choose any type of loan. 5. Click the apply for loan button. 6. Choose any term value. 7. Click the continue button. 8. Leave all the input fields empty. 9. Click the continue button. 10. Leave all the digital footprint data empty. 11. Click the apply loan button.
Expected Result	The user should not be able to submit the loan application and the error alert message will be prompted.
Actual Result	The user was not able to proceed with the loan application. An error alert message was shown to the user. Its error message was “An error has occurred while processing your request. Please try again later.”
Status	**PASS**



localhost:8080 says

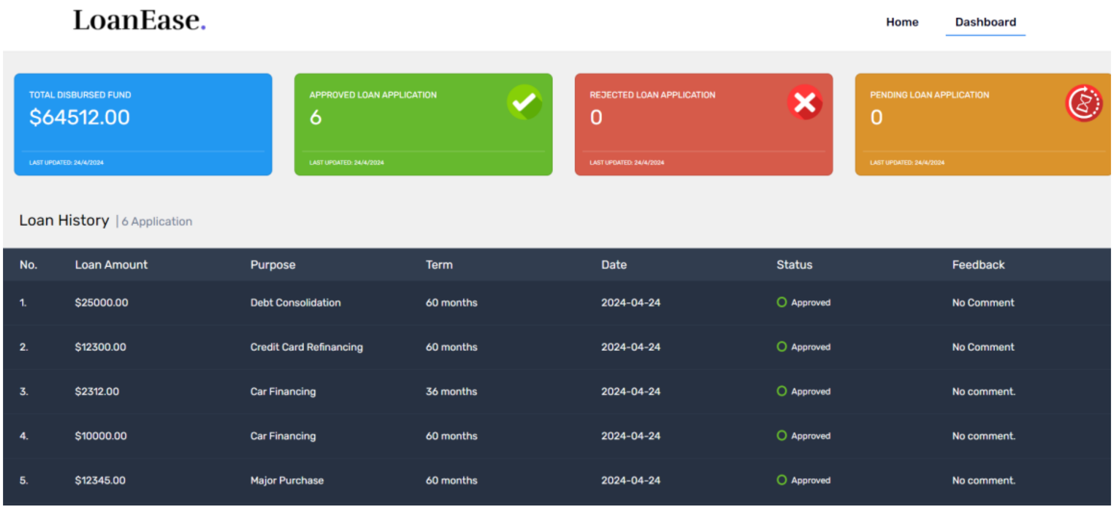
An error has occurred while processing your request. Please try again later

Figure 6.1.2.11: Error message when submitting an empty loan application.

Test Case 5.0 Dashboard

Test No.	5.1
Test Case	Verify that the dashboard should correctly display loan application history based on the user id.
Description	This test case is created to test the functionality of the dashboard to display correct loan application history regarding the user.
Test Step	1. Navigate to log in. 2. Sign in with email: testuser1@gmail.com Password: testuser 3. Click on the dashboard menu.
Expected Result	The user should be able to see their total approved, rejected, and pending loan application. They can also see the total

	amount of disbursed fund. Moreover, the loan application history will be located at the bottom.
Actual Result	The user was able to see their loan application history. All of the information was correct, and it matched the records stored in the database.
Status	**PASS**



TOTAL DISBURSED FUND
$64512.00
LAST UPDATED: 24/4/2024

APPROVED LOAN APPLICATION
6
LAST UPDATED: 24/4/2024

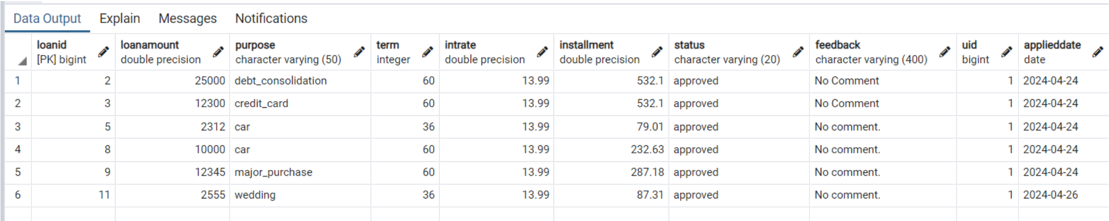
REJECTED LOAN APPLICATION
0
LAST UPDATED: 24/4/2024

PENDING LOAN APPLICATION
0
LAST UPDATED: 24/4/2024

Loan History | 6 Application

No.	Loan Amount	Purpose	Term	Date	Status	Feedback
1.	$25000.00	Debt Consolidation	60 months	2024-04-24	○ Approved	No Comment
2.	$12300.00	Credit Card Refinancing	60 months	2024-04-24	○ Approved	No Comment
3.	$2312.00	Car Financing	36 months	2024-04-24	○ Approved	No comment.
4.	$10000.00	Car Financing	60 months	2024-04-24	○ Approved	No comment.
5.	$12345.00	Major Purchase	60 months	2024-04-24	○ Approved	No comment.

Figure 6.1.2.12: The user dashboard.



	loanid [PK] bigint	loanamount double precision	purpose character varying (50)	term integer	intrate double precision	installment double precision	status character varying (20)	feedback character varying (400)	uid bigint	applieddate date
1	2	25000	debt_consolidation	60	13.99	532.1	approved	No Comment	1	2024-04-24
2	3	12300	credit_card	60	13.99	532.1	approved	No Comment	1	2024-04-24
3	5	2312	car	36	13.99	79.01	approved	No comment.	1	2024-04-24
4	8	10000	car	60	13.99	232.63	approved	No comment.	1	2024-04-24
5	9	12345	major_purchase	60	13.99	287.18	approved	No comment.	1	2024-04-24
6	11	2555	wedding	36	13.99	87.31	approved	No comment.	1	2024-04-26

Figure 6.1.2.13: The database record of the user named “testuser1”.

6.3 Objectives Evaluation

The first objective is the development of personal loan system using a gradient boosting model (XGBoost) and with SHAP values to assess the borrower’s creditworthiness. The final XGBoost model that was trained on the dataset along with the digital footprint data proved to slightly improve the approval rate of the borrowers. Moreover, SHAP values were used to determine the importance of the features to the model. The first objective can be roughly said that it has been achieved but major improvement could still be done to the SHAP interpretation side.

The second objective is to develop a simple and interactive web application for borrowers to apply loan application. Several test cases had been performed on the web application in Chapter 6.1.2. All the test cases had passed, and the functionality of the web application was running smoothly. The user can easily apply for a loan and obtain the result within minutes. This objective has been achieved.

The third objective is to implement the digital footprint data into the personal loan system. Rules and assumptions had been defined and the digital footprint data were generated based on that. The digital footprint data had proven to slightly increase the approval rate of the borrower. This objective has been achieved in this project.

6.4 Concluding Remark

In summary, the prediction model had been evaluated and it had shown to slightly improve the approval rate of the borrower if the digital footprint data was included in the prediction. Moreover, the web application had passed the test cases defined in the Chapter 6.1.2 and all the functions were working correctly and there were no issues. The web application had no problem connecting to the database and as well as the API endpoint of the microservice where the prediction model resided. All of the objectives had been achieved in this project although there were some improvements can be done which will be discussed in Chapter 7 recommendation.

CHAPTER 7 Conclusion and Recommendation

7.1 Conclusion

In conclusion, the development of this personal loan system using XGBoost model and SHAP values are aimed to address the three main critical problems in the lending industry. These problems are the fairness and ethical issues, high rejection rate, and the inefficiency of the traditional loan processes. Motivated by the need for a more efficient and sophisticated approach, this project delves into the usage of XGBoost machine learning algorithm and SHAP values for interpretability and explanation. The proposed solution consists of a meticulous data collection process, as well as preprocessing and feature engineering of the data acquired. Then, to simulate the dynamic nature of real-world data, the digital footprint data is synthesized based on the predefined rules and assumptions. The choice of XGBoost as the primary machine learning model is driven by its performance in handling structured data and its ability to provide interpretable results based on SHAP values. Once the XGBoost model has undergone rigorous training and testing, it is then deployed to the web application. The web application allows the loan applicant to check their eligibility for a loan in real time. As the project moves forward, the results obtained can provide valuable insights into the effectiveness of the proposed solutions, contributing to the modern lending industry.

7.2 Analysis of Results Based on Objectives

7.2.1 First Objective

The first objective is to develop a personal loan system using XGBoost model alongside SHAP values to determine the creditworthiness and eligibility of the borrower. During the implementation stage, two XGBoost models had been developed. The first model was trained on unaltered, original dataset. The second model was trained on altered dataset with digital footprint data. According to Figure 6.1.1.1 and Figure 6.1.1.2, the AUC score of both models were similar. Moreover, the second model seemed to perform slightly worse than the first model as the accuracy is 0.3% lower as seen in Figure 5.3.7.22 and Figure 5.3.7.26, but the impact might not be substantial. However, the balanced accuracy of the second model is slightly higher than the first model. Both of these models performed above average with accuracy score more than 70 percent. The SHAP values were visualized to measure the importance of each feature to the model. From the Figure 5.3.7.24, the int_rate was one of the most important features in making prediction. The higher the interest rate, the lower the chance of having the loan application approved.

7.2.2 Second Objective

The second objective is to develop a simple and interactive web application for the borrower to check their creditworthiness and eligibility for a personal loan by providing relevant information to the personal loan system. Three main modules such as login module, data collection module, and personal loan module were created. All of the modules were developed and implemented successfully without any errors. The user could sign up for a valid account and log in with valid credentials. Not only that, but the loan application form could also prevent null values from being submitted to the database. The user could apply for a loan and view its status in the dashboard. A connection between the web application and the Flask-based microservice was established through the API endpoint. All the loan application data were successfully sent to the prediction model in the microservice, and the prediction result was sent back to the web application. The web application had passed all the test cases defined in the Chapter 6.1.2.

7.2.3 Third Objective

The third objective is the integration of digital footprint into the personal loan system. All the rules and assumptions were defined with fairness in consideration and merged into the existing dataset based on the loan status. Moreover, the addition of digital footprint data proven to be beneficial to the borrower. According to Figure 6.1.1.4 and Figure 6.1.1.5, the test user who had consented their digital footprint data to be included in the loan application had their approval rate slightly increased. The digital footprint data were meant to only supplement the loan application.

7.3 Contribution of the Study

This project is able to automate the loan origination process. All the time-consuming and tedious manual operations are handled by the personal loan prediction model. With that, it speeds up the entire process, reducing the waiting time for the approval of the loan. It also helps the borrower indirectly by integrating their digital footprint data which could potentially boost their creditworthiness to some extent. Moreover, this project could potentially be used to reduce the operational cost of the lending industry. The overall productivity will be boosted by the personal loan system. Most importantly, favouritism and bias could be eliminated with the prediction model as there would be no human bias that would seep into the loan origination process.

7.4 Limitation of this Project

The limitation of this project is the quality of the digital footprint data. Since the data is generated synthetically based on pre-defined rules and assumptions, it might not work well with other datasets as there are underlying hidden relationships between features that need to be taken into consideration. Moreover, the explanation using SHAP values was not as detailed as expected. Due to the complexity of tree-based model, some of the results are hard to interpret. Simple feature importance can be interpreted but a more detail explanation would need additional domain-specific knowledge and SHAP values need to be further analyzed in greater details.

7.5 Future Recommendation

There are some recommendations for this project. Firstly, the quality of the digital footprint data can be further refined. The underlying relationship of the existing features can be analyzed and be used to assist the generation of the synthetic digital footprint data. Moreover, the tuning of hyperparameters of the prediction model can be improved. Last but not least, the SHAP values can be further analyzed and used to explain the decision-making process of the machine learning model in more details.

REFERENCES

[1] S. Ismail, N. Ezaili Alias, W.-L. Koe, R. Othman, and M. Halim Mahphoth, "Empirical investigation on attitude towards personal loans borrowing," KnE Social Sciences, vol. 3, no. 10, p. 114, 2018. doi:10.18502/kss.v3i10.3123

[2] "SCE Credit Access Survey," FEDERAL RESERVE BANK of NEW YORK - Serving the Second District and the Nation - FEDERAL RESERVE BANK of NEW YORK, https://www.newyorkfed.org/microeconomics/sce/credit-access (accessed Sep. 10, 2023).

[3] D. Nair, "Why was your personal loan not approved?," RinggitPlus, https://ringgitplus.com/en/blog/personal-loans/why-was-your-personal-loan-not-approved.html (accessed Sep. 10, 2023).

[4] M. E. Biery, "3 inefficiencies in loan origination and how to solve them," Abrigo, https://www.abrigo.com/blog/3-inefficiencies-in-loan-origination-and-how-to-solve-them/ (accessed Sep. 10, 2023).

[5] "Kofax Intelligent Automation for digital workflow transformation," Kofax, https://www.kofax.com/-/media/files/white-papers/en/wp_kofax-mortage-processing_en.pdf (accessed Sep. 10, 2023).

[6] "Forbes Insights: Accelerate business value with intelligent automation: RPA," Kofax, https://www.kofax.com/Learn/Reports/rp_forbes-insights-accelerate-business-value-with-intelligent-automation_en (accessed Sep. 10, 2023).

[7] "How to overcome the challenges in underwriting with open banking," Overcoming the challenges in underwriting | Tink blog, https://tink.com/blog/open-banking/underwriting-challenges-solution/ (accessed Sep. 10, 2023).

[8] N. Alam et al., "Malaysia State of Consumer Credit 2022," CTOS, https://ctoscredit.com.my/wp-content/uploads/2023/02/CTOS-State-of-Consumer-Credit-2022-Malaysia.pdf (accessed Sep. 10, 2023).

[9] “Personal loans market size, share and Industry Forecast - 2030,” Allied Market Research, https://www.alliedmarketresearch.com/personal-loans-market-A07580 (accessed Sep. 10, 2023).

[10] F. Königstorfer and S. Thalmann, “Applications of artificial intelligence in commercial banks – a research agenda for Behavioral Finance,” *Journal of Behavioral and Experimental Finance*, vol. 27, p. 100352, 2020. doi:10.1016/j.jbef.2020.100352

[11] B. Ghosh and E. Kozarević, “Identifying explosive behavioral trace in the CNX nifty index: A Quantum Finance Approach,” *Investment Management and Financial Innovations*, vol. 15, no. 1, pp. 208–223, 2018. doi:10.21511/imfi.15(1).2018.18

[12] X. Yan, “Research on financial field integrating artificial intelligence: Application basis, case analysis, and SVR model-based overnight,” *Applied Artificial Intelligence*, vol. 37, no. 1, 2023. doi:10.1080/08839514.2023.2222258

[13] “Credit scoring using machine learning,” Credit Scoring Using Machine Learning, https://www.datrics.ai/credit-scoring-using-machine-learning (accessed Sep. 10, 2023).

[14] A. Arif, “NLP in Finance: Examining the impact of natural language processing in financial and banking services,” John Snow Labs, https://www.johnsnowlabs.com/examining-the-impact-of-nlp-in-financial-services (accessed Sep. 10, 2023).

[15] “RPA in lending: The Complete Guide,” Inscribe, https://www.inscribe.ai/robotic-process-automation/rpa-in-lending (accessed Sep. 10, 2023).

[16] “Fraud detection: A lender’s Ultimate Guide,” Ocrolus, https://www.ocrolus.com/fraud-detection (accessed Sep. 10, 2023).

[17] J. Levin, “Council post: Three ways ai will impact the lending industry,” Forbes, https://www.forbes.com/sites/forbesrealestatecouncil/2019/10/30/three-ways-ai-will-impact-the-lending-industry/?sh=39f712216899 (accessed Sep. 10, 2023).

[18] "CIMB Bank recognised for Best Financial Artificial Intelligence Project at the asset triple A digital awards 2022," Media Outreach, https://www.media-outreach.com/news/malaysia/2022/06/09/141976/cimb-bank-recognised-for-best-financial-artificial-intelligence-project-at-the-asset-triple-a-digital-awards-2022 (accessed Sep. 10, 2023).

[19] J. Angwin, J. Larson, L. Kirchner, and S. Mattu, "Machine bias," ProPublica, https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing (accessed Sep. 10, 2023).

[20] M. H. Jarrahi, "Artificial Intelligence and the future of work: Human-AI symbiosis in organizational decision making," *Business Horizons*, vol. 61, no. 4, pp. 577–586, 2018. doi:10.1016/j.bushor.2018.03.007

[21] E. Purificato, F. Lorenzo, F. Fallucchi, and E. W. De Luca, "The use of responsible artificial intelligence techniques in the context of loan approval processes," *International Journal of Human–Computer Interaction*, vol. 39, no. 7, pp. 1543–1562, 2022. doi:10.1080/10447318.2022.2081284

[22] Finscore.ph, "What is credit scoring using digital footprints?: FinScore," FinScore.ph, https://www.finscore.ph/digital-footprint-credit-score/ (accessed Sep. 10, 2023).

[23] Abrigo, "Survey: Lending process challenges remain, despite pandemic-driven digital pushes," PR Newswire: press release distribution, targeting, monitoring and marketing, https://www.prnewswire.com/news-releases/survey-lending-process-challenges-remain-despite-pandemic-driven-digital-pushes-301280466.html (accessed Dec. 7, 2023).

[24] H. Sadok, F. Sakka, and M. E. El Maknouzi, "Artificial Intelligence and Bank Credit Analysis: A Review," *Cogent Economics &amp; Finance*, vol. 10, no. 1, 2022. doi:10.1080/23322039.2021.2023262

[25] S. Athey, “The impact of machine learning on economics,” *The Economics of Artificial Intelligence*, pp. 507–552, 2019. doi:10.7208/chicago/9780226613475.003.0021

[26] H. He and Y. Fan, “A novel hybrid ensemble model based on tree-based method and deep learning method for default prediction,” *Expert Systems with Applications*, vol. 176, p. 114899, 2021. doi:10.1016/j.eswa.2021.114899

[27] M. J. Hamayel, M. A. Abu Mohsen, and M. Moreb, “Improvement of personal loans granting methods in banks using machine learning methods and approaches in Palestine,” *2021 International Conference on Information Technology (ICIT)*, 2021. doi:10.1109/icit52682.2021.9491636

[28] A. Kumar, D. Shanthi, and P. Bhattacharya, “Credit Score Prediction System using deep learning and K-means algorithms,” *Journal of Physics: Conference Series*, vol. 1998, no. 1, p. 012027, 2021. doi:10.1088/1742-6596/1998/1/012027

[29] E. V. Orlova, “Methodology and models for individuals’ creditworthiness management using digital footprint data and machine learning methods,” *Mathematics*, vol. 9, no. 15, p. 1820, 2021. doi:10.3390/math9151820

[30] M. A. Sheikh, A. K. Goel, and T. Kumar, “An approach for prediction of loan approval using machine learning algorithm,” *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2020. doi:10.1109/icesc48915.2020.9155614

[31] B. Spoorthi, S. S. Kumar, A. P. Rodrigues, R. Fernandes, and N. Balaji, “Comparative analysis of Bank Loan Defaulter prediction using Machine Learning Techniques,” *2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, 2021. doi:10.1109/discover52564.2021.9663662

[32] S. Sachan, J.-B. Yang, D.-L. Xu, D. E. Benavides, and Y. Li, “An explainable AI decision-support-system to automate loan underwriting,” *Expert Systems with Applications*, vol. 144, p. 113100, 2020. doi:10.1016/j.eswa.2019.113100

[33] Berg, T. *et al.* (2018) 'On the rise of fintechs credit scoring using digital footprints', *SSRN Electronic Journal* [Preprint]. doi:10.2139/ssrn.3163781.

APPENDIX

Weekly Report

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 2**
Student Name & ID: Jason Lee Chia Shen 20ACB03695	
Supervisor: Ts. Dr Phan Koo Yuen	
Project Title: DESIGN AND IMPLEMENTATION OF PERSONAL LOAN PROCESSING SYSTEM USING AI TECHNIQUE	

1. WORK DONE

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

- A new dataset procured from Kaggle

- Dataset loaded into the Jupyter Notebook

2. WORK TO BE DONE

- Perform explanatory analysis on the dataset

- Plot the dataset for visualization and analysis

- Analyze the relationship and correlation of the features

- Check for any null values, data types, and uniqueness of the dataset

3. PROBLEMS ENCOUNTERED

- There was still no publicly available loan dataset with digital footprint data

4. SELF EVALUATION OF THE PROGRESS

- Progress made was good.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 4**
Student Name & ID: Jason Lee Chia Shen 20ACB03695	
Supervisor: Ts. Dr Phan Koo Yuen	
Project Title: DESIGN AND IMPLEMENTATION OF PERSONAL LOAN PROCESSING SYSTEM USING AI TECHNIQUE	

1. WORK DONE

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

- Performed explanatory analysis on the dataset

- Plotted the dataset for visualization and analysis

- Analyzed the relationship and correlation of the features

- Checked for any null values, data types, and uniqueness of the dataset

- Performed preprocessing on the dataset

2. WORK TO BE DONE

- Split the original dataset into training and testing

- Perform data imputation on the features.

- Encode categorical features

- Generate digital footprint data based on predefined rules and assumptions

3. PROBLEMS ENCOUNTERED

- EDA was harder due to the fact that the new dataset contained more than two million of rows.

4. SELF EVALUATION OF THE PROGRESS

- Progress made was good.

Jason

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 6**
Student Name & ID: Jason Lee Chia Shen 20ACB03695	
Supervisor: Ts. Dr Phan Koo Yuen	
Project Title: DESIGN AND IMPLEMENTATION OF PERSONAL LOAN PROCESSING SYSTEM USING AI TECHNIQUE	

1. WORK DONE

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

- Encoded the categorical data
- Dataset had been split into training and testing set
- Feature analysis and engineering were performed
- Missing values had been imputed
- Train the XGBoost model

2. WORK TO BE DONE

- Test the XGBoost model
- Develop the web application
- Package the trained model into a microservice

3. PROBLEMS ENCOUNTERED

- Without domain knowledge, it was quite hard to determine which feature was useful for the loan application.

4. SELF EVALUATION OF THE PROGRESS

- Was slightly behind schedule in these two weeks as assignments from other core subjects were piling up.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 8**
Student Name & ID: Jason Lee Chia Shen 20ACB03695	
Supervisor: Ts. Dr Phan Koo Yuen	
Project Title: DESIGN AND IMPLEMENTATION OF PERSONAL LOAN PROCESSING SYSTEM USING AI TECHNIQUE	

1. WORK DONE

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

- Tested the XGBoost model
- Tuned the hyperparameters

2. WORK TO BE DONE

- Develop the web application
- Package the trained model into a microservice

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS

- Was getting even behind the schedule. Many presentations, midterms, and assignments were ongoing.

Supervisor's signature

Jason

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 10**
Student Name & ID: Jason Lee Chia Shen 20ACB03695	
Supervisor: Ts. Dr Phan Koo Yuen	
Project Title: DESIGN AND IMPLEMENTATION OF PERSONAL LOAN PROCESSING SYSTEM USING AI TECHNIQUE	

1. WORK DONE

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

- Created the index page

- Defined schema and created tables for storing user and loan history.

- Defined the auth, dashboard, loanApplication, and processApplication servlet but had yet to implement those functions

2. WORK TO BE DONE

- Package the trained model into a microservice

- Develop the apply page

- Develop the dashboard

- Develop the login and registration module

- Implement the functions of the servlet.

- Create session bean and entity class

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS

- Was getting really close to the deadline. Had to rush and work extremely fast.

Supervisor's signature

Jason

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Trimester 3, Year 3	**Study week no.: 12**
Student Name & ID: Jason Lee Chia Shen 20ACB03695	
Supervisor: Ts. Dr Phan Koo Yuen	
Project Title: DESIGN AND IMPLEMENTATION OF PERSONAL LOAN PROCESSING SYSTEM USING AI TECHNIQUE	

1. WORK DONE

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

- Implemented the functions of the servlet
- Developed the apply page
- Developed the dashboard
- Develop the login and registration module
- Package the trained model into a Flask microservice

2. WORK TO BE DONE

- Finish the remaining FYP 2 report
- Connect the microservice to the web application
- Perform testing on the machine learning model in the microservice

3. PROBLEMS ENCOUNTERED

- Needed to learn new framework which was the Flask

4. SELF EVALUATION OF THE PROGRESS

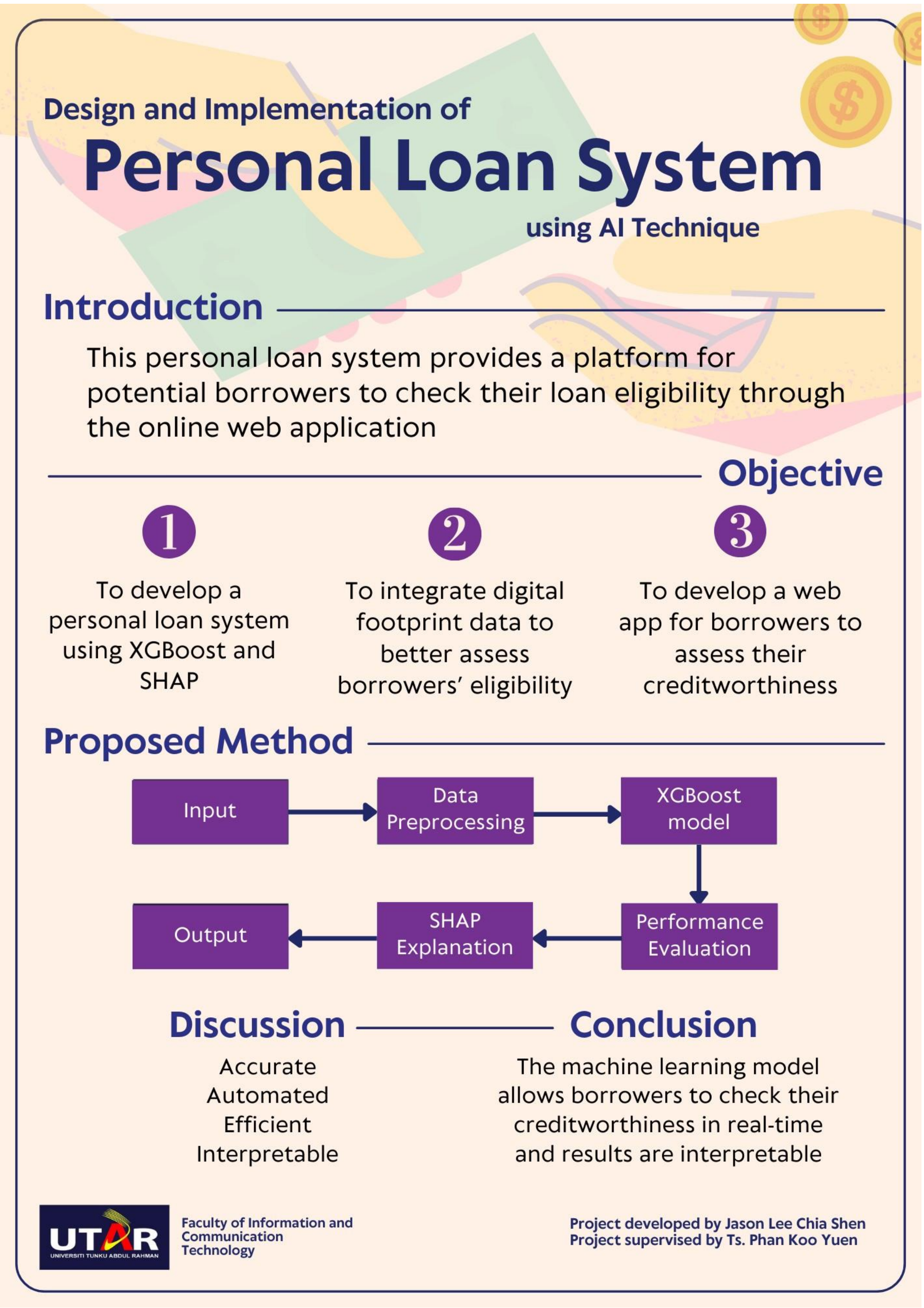
- Slow progress overall. Need to allocate most of the time to finish the remaining system and the project report.

Supervisor's signature

Jason

Student's signature

Poster



Design and Implementation of

Personal Loan System

using AI Technique

Introduction

This personal loan system provides a platform for potential borrowers to check their loan eligibility through the online web application

Objective

1. To develop a personal loan system using XGBoost and SHAP
2. To integrate digital footprint data to better assess borrowers' eligibility
3. To develop a web app for borrowers to assess their creditworthiness

Proposed Method

Input → Data Preprocessing → XGBoost model → Performance Evaluation → SHAP Explanation → Output

Discussion

Accurate
Automated
Efficient
Interpretable

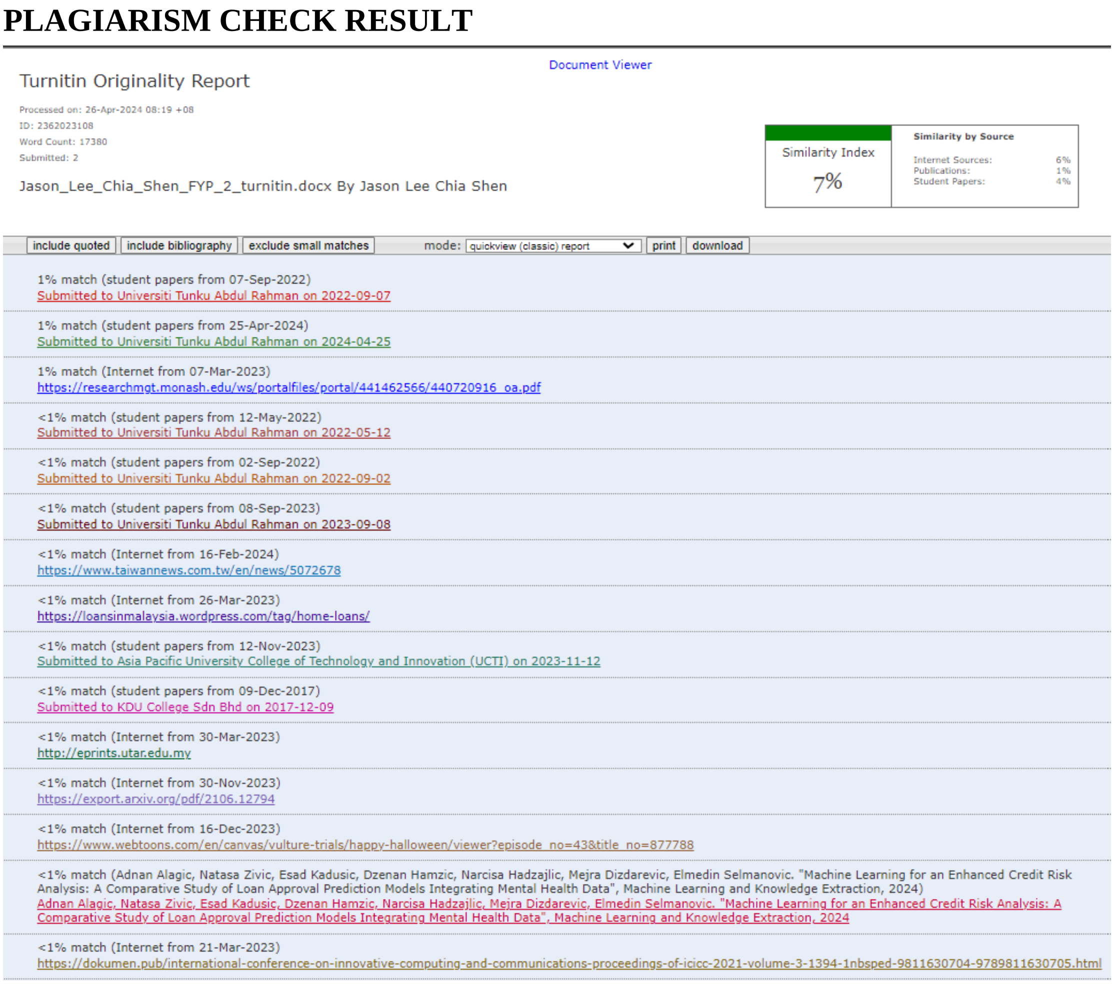
Conclusion

The machine learning model allows borrowers to check their creditworthiness in real-time and results are interpretable

UTAR UNIVERSITI TUNKU ABDUL RAHMAN

Faculty of Information and Communication Technology

Project developed by Jason Lee Chia Shen
Project supervised by Ts. Phan Koo Yuen



PLAGIARISM CHECK RESULT

Turnitin Originality Report

Processed on: 26-Apr-2024 08:19 +08
ID: 2362023108
Word Count: 17380
Submitted: 2

Jason_Lee_Chia_Shen_FYP_2_turnitin.docx By Jason Lee Chia Shen

Similarity Index
7%

Similarity by Source

Internet Sources:	6%
Publications:	1%
Student Papers:	4%

mode: quickview (classic) report

1% match (student papers from 07-Sep-2022)
Submitted to Universiti Tunku Abdul Rahman on 2022-09-07

1% match (student papers from 25-Apr-2024)
Submitted to Universiti Tunku Abdul Rahman on 2024-04-25

1% match (Internet from 07-Mar-2023)
https://researchmgt.monash.edu/ws/portalfiles/portal/441462566/440720916_oa.pdf

<1% match (student papers from 12-May-2022)
Submitted to Universiti Tunku Abdul Rahman on 2022-05-12

<1% match (student papers from 02-Sep-2022)
Submitted to Universiti Tunku Abdul Rahman on 2022-09-02

<1% match (student papers from 08-Sep-2023)
Submitted to Universiti Tunku Abdul Rahman on 2023-09-08

<1% match (Internet from 16-Feb-2024)
https://www.taiwannews.com.tw/en/news/5072678

<1% match (Internet from 26-Mar-2023)
https://loansinmalaysia.wordpress.com/tag/home-loans/

<1% match (student papers from 12-Nov-2023)
Submitted to Asia Pacific University College of Technology and Innovation (UCTI) on 2023-11-12

<1% match (student papers from 09-Dec-2017)
Submitted to KDU College Sdn Bhd on 2017-12-09

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

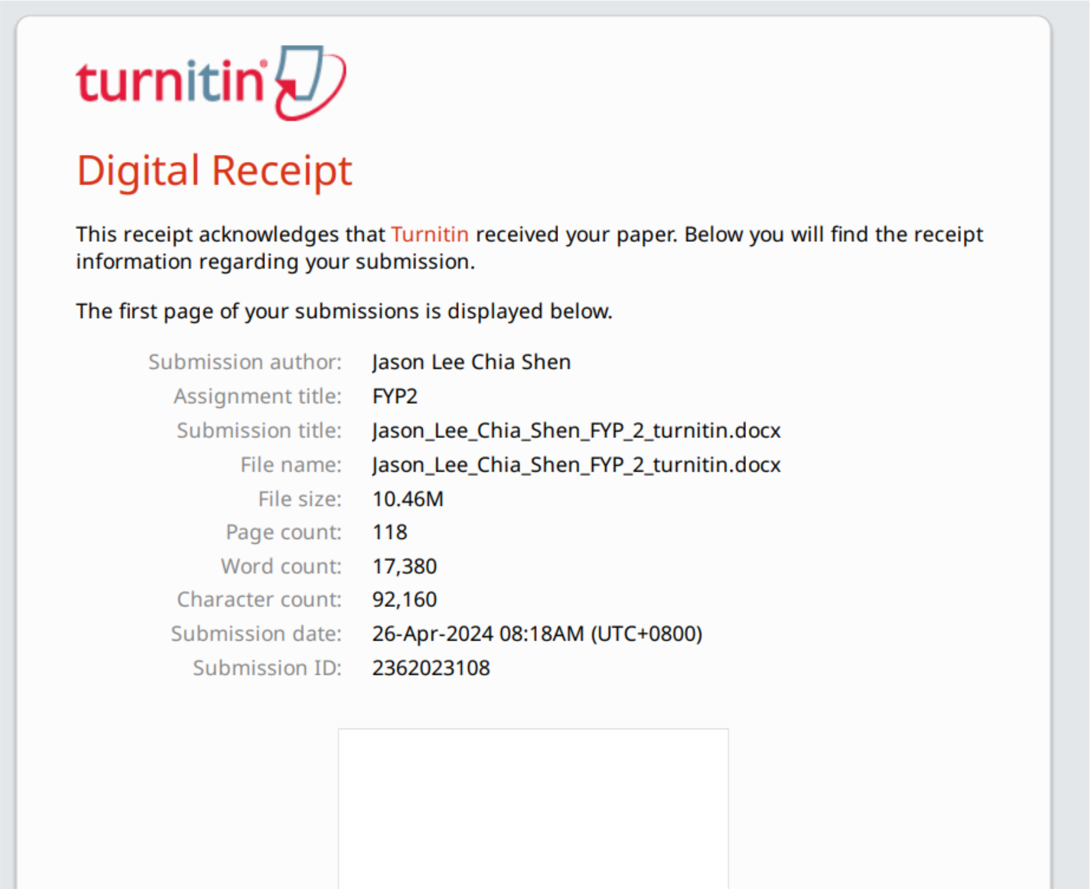
<1% match (Internet from 30-Nov-2023)
https://export.arxiv.org/pdf/2106.12794

<1% match (Internet from 16-Dec-2023)
https://www.webtoons.com/en/canvas/vulture-trials/happy-halloween/viewer?episode_no=43&title_no=877788

<1% match (Adnan Alagic, Natasa Zivic, Esad Kadusic, Dzenan Hamzic, Narcisa Hadzajlic, Mejra Dizdarevic, Elmedin Selmanovic. "Machine Learning for an Enhanced Credit Risk Analysis: A Comparative Study of Loan Approval Prediction Models Integrating Mental Health Data", Machine Learning and Knowledge Extraction, 2024)
Adnan Alagic, Natasa Zivic, Esad Kadusic, Dzenan Hamzic, Narcisa Hadzajlic, Mejra Dizdarevic, Elmedin Selmanovic. "Machine Learning for an Enhanced Credit Risk Analysis: A Comparative Study of Loan Approval Prediction Models Integrating Mental Health Data", Machine Learning and Knowledge Extraction, 2024

<1% match (Internet from 21-Mar-2023)
https://dokumen.pub/international-conference-on-innovative-computing-and-communications-proceedings-of-icicc-2021-volume-3-1394-1nbsped-9811630704-9789811630705.html

PLAGIARISM CHECK RESULT



turnitin

Digital Receipt

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

The first page of your submissions is displayed below.

Submission author:	Jason Lee Chia Shen
Assignment title:	FYP2
Submission title:	Jason_Lee_Chia_Shen_FYP_2_turnitin.docx
File name:	Jason_Lee_Chia_Shen_FYP_2_turnitin.docx
File size:	10.46M
Page count:	118
Word count:	17,380
Character count:	92,160
Submission date:	26-Apr-2024 08:18AM (UTC+0800)
Submission ID:	2362023108

Universiti Tunku Abdul Rahman			
Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Jason Lee Chia Shen
ID Number(s)	20ACB03695
Programme / Course	Bachelor of Computer Science (Honours) / Project II
Title of Final Year Project	Design and Implementation of Personal Loan Processing System using AI Technique

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 7 **%** **Similarity by source** Internet Sources: 6 % Publications: 1 % Student Papers: 4 %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required, and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and

(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

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

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

Signature of Supervisor

Name: Phan Koo Yuen

Date: 26/4/2024

Signature of Co-Supervisor

Name: ____________

Date: ____________

FYP 2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB03695
Student Name	Jason Lee Chia Shen
Supervisor Name	Ts Dr Phan Koo Yuen

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

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

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

____

(Signature of Student)
Date:26/4/2024