“Loan Status Prediction using Random-Forest Classifier”

A Project Report Submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of

Bachelor of Engineering in Computer Science & Engineering

Submitted by:

Ansh Joshi (0827CS201036)
Bhavik Mundra (0827CS201057)
Bhavika Darpe (0827CS201059)

Guided by:

Prof. Juhi Shrivastava
Assistant Professor
Computer Science & Engineering

Acropolis Institute of Technology & Research, Indore

Jan – June 2024

EXAMINER APPROVAL

The Project entitled **"Loan Status Prediction using Random-Forest Classifier"** submitted by **Ansh Joshi (0827CS201036)**, **Bhavik Mundra (0827CS201057)**, **Bhavika Darpe (0827CS201059)** has been examined and is hereby approved towards partial fulfilment for the award of Bachelor of Engineering degree in Computer Science & Engineering discipline, for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner) **(External Examiner)**

Date: **Date:**

GUIDE RECOMMENDATION

This is to certify that the work embodied in this project **"Loan Status Prediction using Random-Forest Classifier"** submitted by **Ansh Joshi (0827CS201036)**, **Bhavik Mundra (0827CS201057)**, **Bhavika Darpe (0827CS201059)** is a satisfactory account of the Bonafede work done under the supervision of **Prof. Juhi Shrivastava** are recommended towards partial fulfilment for the award of the Bachelor of Engineering (Computer Science & Engineering) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

(Project Guide)

(Project Coordinator)

(Dean Academics)

STUDENTS UNDERTAKING

This is to certify that project entitled **"Loan Status Prediction using Random-Forest Classifier"** has developed by us under the supervision of **Prof. Juhi Shrivastava**. The whole responsibility of work done in this project is ours. The sole intension of this work is only for practical learning and research.

We further declare that to the best of our knowledge; this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work found then we are liable for explanation to this.)

Ansh Joshi (0827CS201036)

Bhavik Mundra(0827CS201057)

Bhavika Darpe (0827CS201059)

ACKNOWLEDGEMENT

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely. There are number of people without whom this projects work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support. We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentors **Prof. Juhi Shrivastava**, Associate Professor, AITR, for their motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Dr. Kamal Kumar Sethi**, HOD CSE, AITR Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of CSE Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if do not acknowledge the support and guidance received from **Dr. S.C. Sharma**, Director, AITR, Indore whenever needed. We take opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for project to achieve our objectives.

We are grateful to **our parent** and **family members** who have always loved and supported us unconditionally. To all of them, we want to say, “Thank you”, for being the best family that one could ever have and without whom none of this would have been possible.

Ansh Joshi (0827CS201036)

Bhavik Mundra(0827CS201057)

Bhavika Darpe (0827CS201059)

EXECUTIVE SUMMARY

"Loan Status Prediction using Random-Forest Classifier"

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal (MP), India for partial fulfillment of Bachelor of Engineering in Computer Science & Engineering branch under the sagacious guidance and vigilant supervision of **Prof. Juhi Shrivastava**.

The project is based on Machine Learning concerned with supervised learning and different classification techniques. In this project, the dataset used consists of attributes like income, debt-to-income ratio, credit score and many more. The result includes the identifying if a particular applicant can avail the loan or not.

Key words: Encoding, Classifiers

“Where the vision is one year,

cultivate flowers;

Where the vision is ten years,

cultivate trees;

Where the vision is eternity,

cultivate people.”

- Oriental Saying

List of Figures

List of Tables

List of Abbreviations

Abbr1: AI-Artificial Intelligence

Abbr2: GPU-Graphics Processing Unit

Abbr3: HTML-Hypertext Markup Language

Abbr4: ML-Machine Learning

Abbr5: RAM-Random Access Memory

Abbr6: ROI-Region of Interest

Abbr7: URL-Uniform Resource Locator

Table of Contents

CHAPTER 1:

INTRODUCTION

CHAPTER 1: INTRODUCTION

The project "Loan Status Prediction Using Random-Forest Classifier" addresses the critical need in the banking sector to enhance credit risk assessment processes. With lending being a primary revenue source for financial institutions, the project aims to leverage historical loan data and advanced machine learning techniques to develop a predictive model. This model is designed to accurately forecast loan approval status by analysing various applicant attributes. By classifying borrowers as likely defaulters or non-defaulters, the project seeks to empower banks to mitigate risks and improve decision-making in lending practices. Through the integration of machine learning algorithms and data preprocessing techniques, the project endeavours to revolutionize the loan approval process, ultimately leading to increased customer satisfaction and economic prosperity.

1.1. Overview

The project is based on binary classification where the outcome is classified as approved or not approved. The classification algorithms like random forest classifier, decision tree classifier etc are used and their accuracy is compared to select the one with the most accuracy. The model is trained and tested for a large dataset and finally integrated with frontend to make a fully functional loan status prediction website.

1.2. Background and Motivation

In recent years, the financial landscape has witnessed a paradigm shift in the way lending institutions evaluate loan applications. Traditional methods, reliant on manual assessment and subjective decision-making, are being supplanted by data-driven approaches leveraging machine learning and predictive analytics. This transition is propelled by the need for greater efficiency, accessibility, and fairness in the lending process.

Moreover, with the proliferation of digital platforms and the democratization of financial services, there is a growing demand for faster, more accurate loan approval mechanisms. This report endeavors to explore the development and implementation of a loan status prediction system, aiming to harness the power of machine learning to revolutionize the lending landscape. Through the utilization of advanced algorithms and predictive models, this system seeks to expedite the loan approval process.

1.3. Problem Statement and Objectives

The current loan approval process in many financial institutions is fraught with challenges, including prolonged processing times, subjective decision-making, and a lack of transparency. These inefficiencies often result in delays for borrowers seeking credit, while also hindering lenders' ability to make accurate and timely decisions. Furthermore, the reliance on manual assessments can introduce biases and inconsistencies, impacting the fairness of lending practices. In light of these issues, there is a pressing need for a more efficient, objective, and transparent approach to loan status prediction, which not only enhances the borrower experience but also optimizes the operational efficiency of lending institutions, ensuring better utilization of resources and reducing the risk of defaults.

Objective: The objective of this project is to develop an automated loan status prediction system that leverages machine learning techniques to provide fast, accurate, and unbiased assessments of loan applications. By analyzing historical loan data and utilizing advanced predictive models, the system aims to streamline the lending process, reduce processing times, and improve the overall efficiency of loan approval procedures.

1.4. Scope of the Project

In the future, loan status prediction is set to make big changes in how loans are given out. With better technology, computers will be able to predict who

can get a loan faster and more accurately. They'll use new kinds of information, like what people do online and how they pay their bills, to decide. These systems will also make sure that loans are given out fairly and explain why they make the decisions they do.

1.5. Team Organization

1. **Ansh Joshi:**

   I led the front-end design, creating an intuitive user interface that enhances user experience. My creative approach and attention to visual design made the application user-friendly.

2. **Bhavik Mundra:**

   I managed the back-end and the model integration with it. The backend was developed using Django and also handled the categorical values at the backend. My technical expertise contributed to a responsive and fully functional web application.

3. **Bhavika Darpe:**

   I worked on machine learning domain of this project. I pre-processed the dataset, identified the relevant attributes and applied classification algorithms to develop a predictive model for loan status prediction.

1.6. Report Structure

The project Loan Status Prediction is primarily concerned with handling the categorical values and training the model and whole project report is categorized into five chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature- explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System- starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation- includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designed in project along with their functionality. Further it discusses the experiment results along with testing of the project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion- Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

CHAPTER 2:

REVIEW OF LITERATURE

CHAPTER 2: REVIEW OF LITERATURE

Over the years, technological advancements have revolutionized this area. Machine learning and image recognition have become instrumental in the development of automated systems. Loan status prediction has undergone significant development over the years, driven by advancements in machine learning and predictive analytics. Early approaches to loan prediction primarily relied on traditional statistical methods that provided valuable insights into borrower creditworthiness but often struggled to capture complex patterns in data.

In recent years, researchers have explored a wide range of machine learning algorithms for loan status prediction, including support vector machines, random forests, decision tree classifiers and neural networks. These algorithms offer improved predictive accuracy and scalability, making them well-suited for handling large volumes of loan applications.

The literature review also highlights the importance of performance evaluation. It is a critical aspect of assessing loan prediction models. Studies commonly use metrics such as accuracy, precision and recall to measure model performance. Comparative studies have shown that different machine learning algorithms may perform differently depending on the dataset and problem domain, highlighting the importance of empirical evaluation.

2.1. Preliminary Investigation

2.1.1. Current System

There are certain systems available for loan status prediction as follows:

1. **FICO Score**: Widely used credit scoring system assessing credit risk based on factors like payment history, credit utilization, and length of credit history. Scores range from 300 to 850, with higher scores indicating lower risk.

2. **Manual Loan Approval**: Loan applications are manually reviewed by loan officers who assess various factors such as credit history, income, debt-to-income ratio, and collateral to determine the applicant's eligibility for a loan. app designed to assist farmers, gardeners, and agricultural enthusiasts in identifying plant diseases, pests, and nutrient deficiencies.

2.2. Requirement Identification and Analysis for Project

Requirement identification and analysis for a loan status prediction system involves understanding the needs and expectations of users in the lending process. Here's an analysis of the key requirements for such a system:

- Data input and preprocessing: The system must efficiently process loan application data, including borrower information, financial history, and loan details, to assess creditworthiness accurately.
- Prediction accuracy: The system should utilize advanced machine learning algorithms to predict loan status with high accuracy, minimizing the risk of defaults and maximizing the approval of viable loan applications.
- User friendly interface: The user interface should be intuitive and accessible, allowing users to easily submit their details and get the result in real time.
- Speed and efficiency: Rapid processing and decision-making are crucial. The system should provide quick responses to loan applications to streamline the lending process and enhance customer experience.
- Scalability and performance: Design the system to handle a large volume of loan applications and scale with increasing demand, ensuring consistent performance and reliability.
- Data Security and Privacy: Ensure robust data security and privacy measures to protect user data.

- Continuous Improvement: Implement mechanisms for continuous feedback collection, model improvement, and database updates based on user input and emerging credentials of the user.
- Compatibility and Integration: Ensure the system is compatible with various devices and operating systems.

By analyzing these requirements, you can create a comprehensive and effective loan status prediction system that addresses the needs of users.

2.3. Conclusion

The research undertaken in "Loan Status Prediction" holds significant promise for transforming the loan approval landscape in the financial sector. Through the utilization of historical loan data and cutting-edge machine learning methodologies, the project endeavours to construct a predictive framework adept at reliably determining loan approval outcomes based on diverse applicant characteristics. This endeavour marks a pivotal step towards enhancing the efficiency and accuracy of loan approval processes, thereby offering substantial benefits to both financial institutions and loan applicants alike.

CHAPTER 3:

PROPOSED SYSTEM

CHAPTER 3: PROPOSED SYSTEM

3.1. The Proposal

Our project aims to develop an innovative Loan Status Prediction System leveraging machine learning algorithms and data preprocessing techniques. This system will provide lenders and borrowers with a seamless and efficient platform for predicting loan outcomes and making informed decisions. Borrowers will submit loan credentials through a user-friendly interface, and the system will utilize advanced predictive models to assess creditworthiness and predict loan status in real-time. The primary objective is to streamline the loan approval process, reduce processing times, and enhance transparency and fairness in lending decisions. Key features of the system will include robust data security measures, scalability to handle large volumes of loan applications, and compatibility with existing banking systems. By fulfilling these requirements, our project aims to revolutionize the lending landscape, promoting financial inclusivity and economic growth while mitigating risks for lenders and borrowers alike.

3.2. Benefits of the Proposed System

Here are the benefits of the proposed system, which can instill trust and support among users:

- **Efficiency**: By automating the loan approval process, the system significantly reduces processing times, allowing lenders to make faster decisions.
- **Accuracy**: Leveraging advanced machine learning algorithms, the system provides more accurate assessments of creditworthiness, minimizing the risk of default and enabling lenders to make informed lending decisions based on reliable data.

- **Transparency**: The system promotes transparency in lending practices by providing borrowers with clear insights into the factors influencing their loan status predictions. This transparency enhances trust between lenders and borrowers and fosters a more transparent and equitable lending environment.
- **Fairness**: By using objective criteria and reducing the influence of subjective judgment, the system helps mitigate biases in lending decisions, ensuring fair treatment for all applicants regardless of demographic or socioeconomic factors.
- **Cost Savings**: Streamlining the loan approval process and reducing the need for manual intervention leads to cost savings for lenders, allowing them to allocate resources more efficiently and potentially offer more competitive loan terms to borrowers.
- **Access to Credit**: By expediting the loan approval process and improving the accuracy of credit assessments, the system expands access to credit for borrowers who may have been underserved or overlooked by traditional lending institutions.
- **Risk Mitigation**: The system enables lenders to better assess and mitigate risks associated with lending, helping to reduce the likelihood of defaults and minimize losses from non-performing loans.

3.3. Feasibility Study

Feasibility Study for a Loan Status Prediction

1. **Technical Feasibility**: Technical Feasibility: Assess the availability of technology and resources required for developing the loan status prediction system. Evaluate the feasibility of implementing machine learning algorithms, data processing infrastructure, and user interface design within the project constraints.

2. **Economic Feasibility**: Determine the financial viability of the project by estimating the costs associated with software development, data acquisition, infrastructure setup, and ongoing maintenance. Conduct a cost-benefit analysis to compare the expected benefits of the system, such as improved efficiency and reduced risk, against the projected costs.
3. **Operational Feasibility**: Evaluate the practicality and effectiveness of integrating the loan status prediction system into existing lending processes. Consider factors such as user acceptance, workflow integration, and training requirements to ensure smooth adoption and operation of the system within the organization.

3.3.1. Technical

Data Requirements: Identify the data sources and types of data needed for loan status prediction, including borrower information, financial history, loan details, and historical performance data. Assess the availability, quality, and relevance of data sources to ensure the accuracy and reliability of predictive models.

Machine Learning Algorithms: Evaluate different machine learning algorithms, such as logistic regression, decision trees, decision trees, and neural networks, for their suitability in predicting loan status. Consider factors such as prediction accuracy, scalability and precision to select the most appropriate algorithm for the task.

Infrastructure Requirements: Determine the hardware and software infrastructure needed to support the loan status prediction system, including servers, databases, cloud services, and development tools. Design a scalable and robust architecture that can handle large volumes of data and user requests efficiently.

3.3.2. Economical

Cost-Benefit Analysis: Conduct a thorough cost-benefit analysis to evaluate the economic feasibility. Consider the initial and ongoing costs of system development, including software development, server hosting, and maintenance.

Revenue Generation: Explore potential revenue sources, such as premium subscriptions, ads, or partnerships with banks and lending organizations. Estimate revenue projections and the timeframe for achieving profitability.

Risk Analysis: Identify potential risks and uncertainties that could impact the financial outcomes of the project, such as technology obsolescence, regulatory changes, and market fluctuations.

3.3.3. Operational

Resource Availability: Ensure that human and physical resources necessary for system operation are readily available. Consider factors like developer expertise, data annotation, and server hosting.

User Requirement: Analyse the needs and preferences of end-users, including lenders, borrowers, and system administrators to ensure that the loan status prediction system meets their operational requirements. Gather feedback through interviews, surveys, and usability testing to refine system features and functionalities.

Workflow Integration: Evaluate how the loan status prediction system will integrate into existing lending process and workflows. Identify potential bottlenecks, dependencies and areas for process improvement to optimize the operational efficiency of the system.

3.4. Design Representation

3.4.1. Sequence Diagram

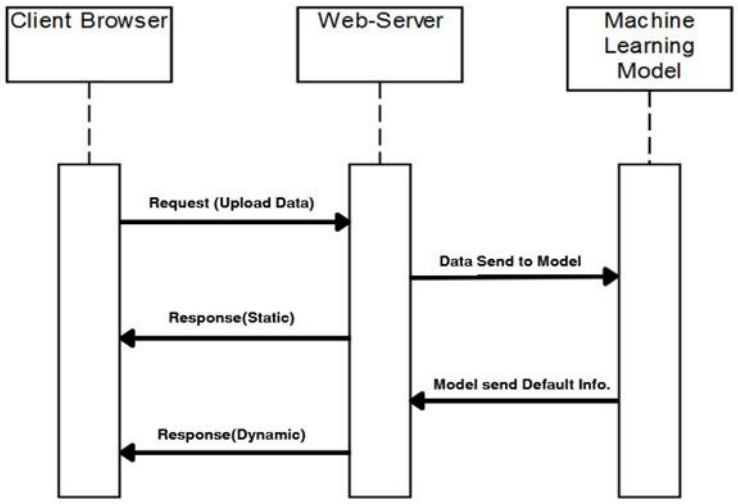


Figure 3.1: Sequence Diagram

3.4.2. Use Case Diagram

Figure 3.2: Use Case Diagram

The use case diagram presents an overview of the interactions between users and the loan status prediction system, illustrating the primary functionalities and relationships within the loan application process. There can be two types of users : one is the common person and another a bank manager.

The bank-side actor is responsible for managing all user data related to loan applications. This actor uploads crucial user details such as income, co-applicant information, loan type, and other relevant data into the loan status prediction system. These details serve as input for the prediction process, aiding in determining whether the applicant should be granted a loan or not.

On the other hand, the ML model acts as another key actor in the system. It receives the uploaded user data from the bank-side actor and utilizes machine learning algorithms to analyze and predict the loan status. Based on the input data, the ML model determines whether the risk associated with granting the loan is low or high.

Overall, the use case diagram emphasizes the interaction between users and the loan status prediction system, highlighting the roles of both the bank-side actor and the ML model in facilitating the loan application process and making informed decisions regarding loan approvals.

3.4.3. Entity Relationship Diagram

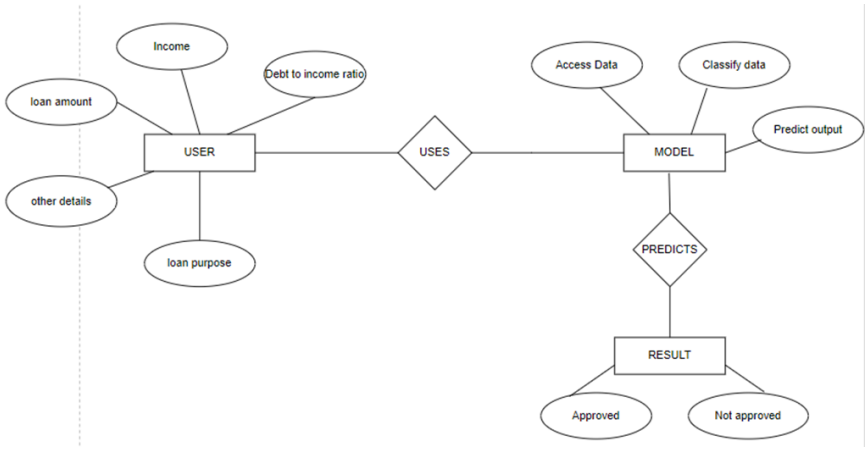


Figure 3.3: Entity Relationship Diagram

The Entity-Relationship (ER) diagram for the loan status prediction system would illustrate the entities, relationships, and attributes involved in the system, providing a clear visual representation of the data flow and structure within the system. In this system, the primary entities would include:

1. **User**: Represents individuals or entities interacting with the system to upload data for loan prediction. Attributes of the User entity may include user ID, username, and other relevant user details.
2. **ML Model**: Represents the machine learning model utilized for predicting loan status. Attributes of the ML Model entity may include model ID, algorithm used, and other relevant details.

The relationships between these entities would illustrate the flow of data within the system. For example:

- **User uploads Data**: This relationship indicates that a user uploads data to the system for loan prediction.
- **ML Model accesses Client Data**: This relationship signifies that the ML model accesses the uploaded user data for prediction. It

would be depicted by a line connecting the ML Model entity to the Client Data entity.

- **ML Model classifies output**: This relationship shows that the ML model classifies the output based on the accessed data. It would be represented by a line connecting the ML Model entity to the predicted output, which may be labelled as "Loan Status" with values like "Approved" or "Not Approved."

  Overall, the ER diagram visually communicates how user data is uploaded, accessed by the ML model, and classified to determine the loan status prediction, helping to understand the structure and flow of data within the loan status prediction system.

3.4.4. Dataset Structure

Dataset is the training and testing data on which our machine learns and test its accuracy. The dataset is the Loan Default which consists of approx. 6100 data entries and 18 columns some of which are income, debt to income ratio, co-applicant income, loan type etc.

Following are the attributes present in the dataset:

- Credit Score
- Income
- Employment Status
- Debt-to-Income Ratio
- Loan Amount
- Loan Term
- Payment History
- Age
- Loan Purpose
- Assets
- Savings
- Housing Status

- Marital Status
- Education Level
- Number of Dependents
- Years of Employment
- Current Debt
- Credit Utilization Ratio

3.5. Deployment Requirements

The deployment hardware and software requirements for the proposed loan status prediction system include the following:

3.5.1. Hardware

- Processor: Intel Core i5 or higher
- RAM: 8 GB or higher
- Hard Disk Space: 1 TB or higher
- Graphics Card: NVIDIA or AMD with a minimum of 2 GB memory
- Internet Connection: Broadband or higher

3.5.2. Software

- Operating System: Windows 10 or Ubuntu 18.04 or higher
- Python 3.7 or higher
- Jupyter Notebook
- VS Code

The software components must be installed and configured properly for the system to function correctly. The Python environment must have all the necessary packages installed like Django, scikit-learn, pandas, numpy.

Overall, the hardware and software requirements for the proposed loan status prediction system require careful consideration to ensure that the system functions correctly and provides users with a satisfactory experience.

CHAPTER 4:

IMPLEMENTATION

CHAPTER 4. IMPLEMENTATION

The implementation process would involve the following steps:

1. **Data Collection and Preparation**: Gather a comprehensive dataset of information which is required to be provided by the user. We have taken a dataset which consists of columns like credit score, age, loan type, income, co-applicant income, number of dependents etc. The dataset overall consists of 18 columns and approx. 10000 records varying from numerical, floating to categorical values.
2. **Preprocessing Data**: The preprocessing step includes identifying missing values and outliers if present. The missing values are replaced with mean or mode of the remaining values in the columns. Also unwanted columns are removed if those don't hold any correlation with the target variable.
3. Convert the categorical values into numerical values so that these can be fed into ML model. The conversion is done by the encoding methods like label encoder or one hot encoder. Through the encoding techniques, the categorical values are mapped to numerical values.
4. Split the dataset into test dataset and training dataset respectively.
5. **Model Training**: The training dataset is used to train the machine learning model by importing scikit-learn library. Through scikit-learn, we can implement various machine learning models for regression, classification, clustering, and statistical tools for analyzing these models.
6. **Model Evaluation**: Evaluate the trained model's performance using test dataset. To evaluate the model's performance, find the accuracy and precision of the test dataset in comparison to training dataset.
7. **Deployment**: To be used for loan status prediction, import the model to enable it to integrate with the frontend and deploy it.

4.1. Technique Used

4.1.1. Machine Learning:

The study of algorithms and statistical models that enable computers to learn and make predictions without explicit programming instructions. Machine learning techniques are categorized into supervised, unsupervised, and reinforcement learning, with the former being most relevant to loan approval prediction.

4.1.2. Encoding Technique

Encoding techniques are used to represent categorical data (data that consists of categories or labels) in a numerical format that can be fed into machine learning models. For example, Label Encoding assigns a unique integer to each category in the feature whereas One-hot encoding converts each category into a binary vector where each category is represented by a binary value (0 or 1).

4.1.3. Random Forest Classifier

Random Forest: A popular machine learning algorithm for classification tasks, such as loan approval prediction. It combines multiple decision trees to improve prediction accuracy and reduce overfitting.

4.2. Tools Used

In a loan status prediction project using machine learning, several tools and libraries are commonly used to develop, train, and deploy the model. Some of the key tools and libraries used in such projects include:

4.2.1. Python

Python is the most commonly used programming language for machine learning and computer vision tasks. It provides a wide range of libraries and frameworks for these tasks.

4.2.2. Jupyter Notebook

Jupyter Notebook is often used for interactive coding and data analysis, making it a valuable tool for experimenting with models and visualizing data.

4.2.3. Django

Django is a high-level Python web framework that enables rapid development of secure and scalable web applications. It follows the Model-View-Template (MVT) architectural pattern, which is a variation of the Model-View-Controller (MVC) pattern. Django automatically generates a customizable admin interface for managing site content. ORM enables developers to interact with the database using Python objects, abstracting away the complexities of SQL queries.

4.2.4. Scikit-learn

Scikit-learn, often abbreviated as sklearn, is a popular open-source machine learning library for Python. It provides simple and efficient tools for data mining and data analysis, built on top of NumPy, SciPy, and matplotlib.

Scikit-learn offers a wide range of machine learning algorithms for classification, regression, clustering, dimensionality reduction, model selection, and preprocessing.

4.3. Language Used

Python language is used in the system due to the following characteristics:

- **Simple**:

  Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English (but very strict English!). This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the syntax i.e. the language itself.

- **Free and Open Source:**

  Python is an example of a FLOSS (Free/Libre and Open-Source Software). In simple terms, you can freely distribute copies of this software, read the software's source code, make changes to it, use pieces of it in new free programs, and that you know you can do these things.

- **Object Oriented**:

  Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simple way of doing object-oriented programming, especially, when compared to languages like C++ or Java.

- **Extensive Libraries**:

  The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, ftp, email, XML,

4.4. Glimpse of Project

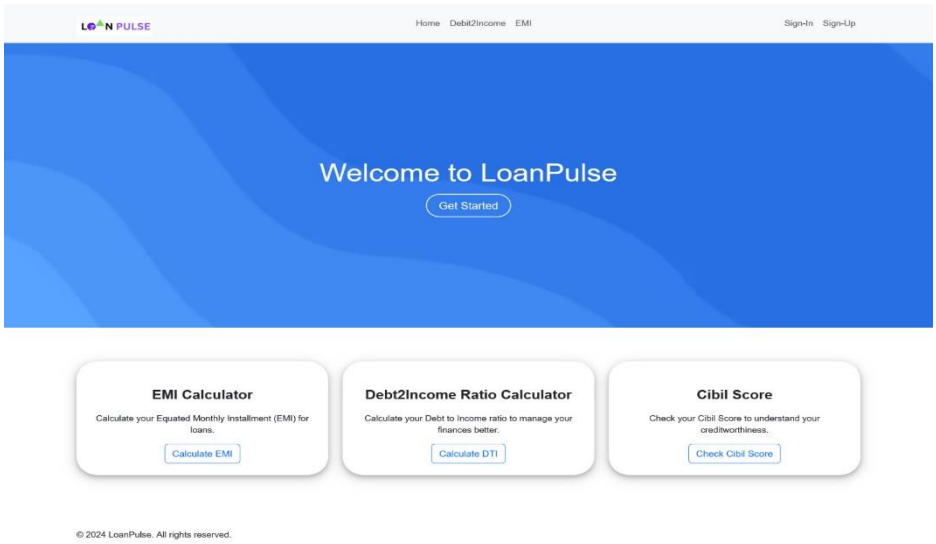


Figure 4.1: Home Page

Pseudocode:

```
<div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav mx-auto">
         <li class="nav-item active">
          <a class="nav-link" href="/" >Home</a>
          </li>
           <li class="nav-item">
            <a class="nav-link" href="{% url 'dbtinc' %}">Debit2Income</a>
            </li>
            <li class="nav-item">
             <a class="nav-link" href="{% url 'emi' %}">EMI</a>
             </li>
             </ul>
        <ul class="navbar-nav ml-auto">
        {% if not request.user.is_authenticated %}
         <li class="nav-item">
          <a class="nav-link" href="{% url 'signin' %}">Sign-In</a>
```

```
            </li>
             <li class="nav-item">
              <a class="nav-link" href="{% url 'register' %}">Sign-Up</a>
              </li>
              {% endif %}
              {% if request.user.is_authenticated %}
               <li class="nav-item">
                <a class="nav-link" href="{% url 'signout' %}">Sign-Out</a>
                </li>
               {% endif %}
          <div>
           <h1>Welcome to LoanPulse</h1>
           <p>Your financial solution partner</p>
           <a href="{% url 'predict' %}"> Get Started </a>
           </div>
</div>
```

Description: The home page consists of a navbar for login, logout and registration purpose. There is EMI calculator, Debt to income ratio calculator and Cibil Score calculator as well.

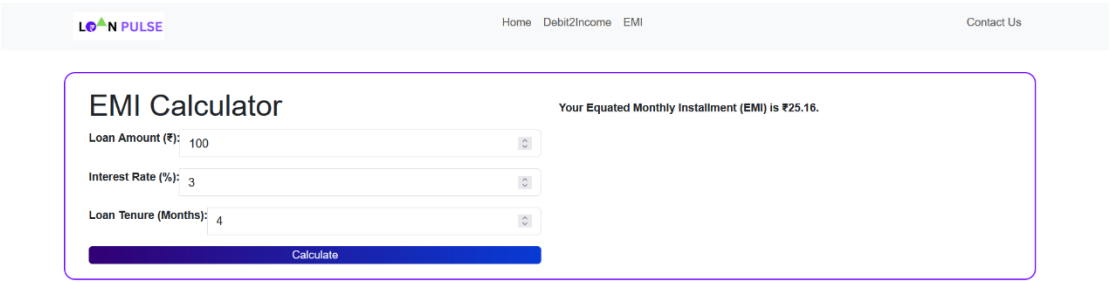


Figure 4.2: EMI Calculator

Pseudocode:

<h1>EMI Calculator</h1>

<label for="loanAmount">Loan Amount (₹):</label>

<input type="number" id="loanAmount" class="form-control" min="0" step="any">

<label for="interestRate">Interest Rate (%):</label>

<input type="number" id="interestRate" class="form-control" min="0" step="any">

<label for="loanTenure">Loan Tenure (Months):</label>

<input type="number" id="loanTenure" class="form-control" min="0" step="any">

<button onclick="calculateEMI()">Calculate</button>

Description: Loan amount, interest rate and loan tenure is provided to calculate EMI.

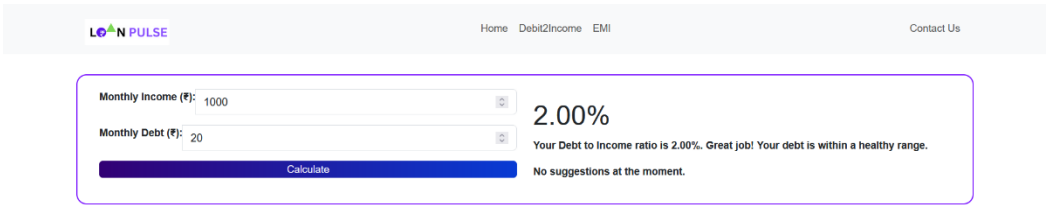


Figure 4.3: Debit2Income Page

Pseudocode:

```
<div >
        <h2>Debit2Income</h2>
        <break></break>
        <div class="input-group">
        <label for="monthlyIncome">Monthly Income (₹):</label>
        <input type="number" id="monthlyIncome" class="form-
        control" min="0" step="any">
        </div>
        <div class="input-group">
        <label for="monthlyDebt">Monthly Debt (₹):</label>
        <input type="number" id="monthlyDebt" class="form-
        control" min="0" step="any">
        </div>
        <button onclick="calculateDTI()">Calculate</button>
</div>
```

Description: This page calculates debt to income ratio when monthly income and monthly debt is fed onto the page.

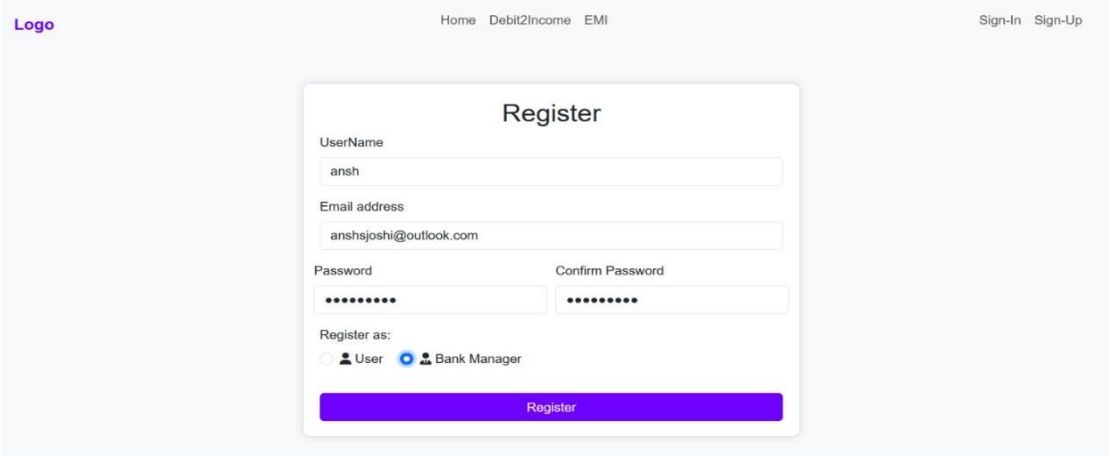


Figure 4.4: Registration Page

Pseudocode:

```
<form id="registerForm" method="post" action="register">
        {% csrf_token %}
<div class="form-group mb-3">
  <label for="username" class="form-label">UserName</label>
<input type="text" name="username" class="form-control"
id="username" required>
</div>
<div class="form-group mb-3">
  <label for="email" class="form-label">Email address</label>
<input type="email" name="email" class="form-control" id="email"
required>
</div>
<div class="form-group row mb-3">
  <div class="col">
    <label for="password" class="form-label">Password</label>
 <input type="password" name="password" class="form-control"
id="password" required>
  </div>
  <div class="col">
```

```
 <label for="confirmPassword" class="form-label">Confirm
Password</label>
<input type="password" name="confirmPassword" class="form-control"
id="confirmPassword" required>
  </div>
</div>
<div class="form-group mb-3">
  <label class="form-label">Register as:</label>
  <div>
    <div class="form-check form-check-inline">
 <input class="form-check-input" type="radio" name="userType"
id="user" value="user" required>
      <label class="form-check-label" for="user">
        <i class="fa fa-user"></i> User
      </label>
    </div>
    <div class="form-check form-check-inline">
<input class="form-check-input" type="radio" name="userType"
id="bankManager" value="bank" required>
      <label class="form-check-label" for="bankManager">
        <i class="fa fa-user-tie"></i> Bank Manager
      </label>
    </div>
  </div>
</div>
```

Description: This code is for registration page. It consists of a form which takes the user details and allows user to register as a common user or a bank manager.

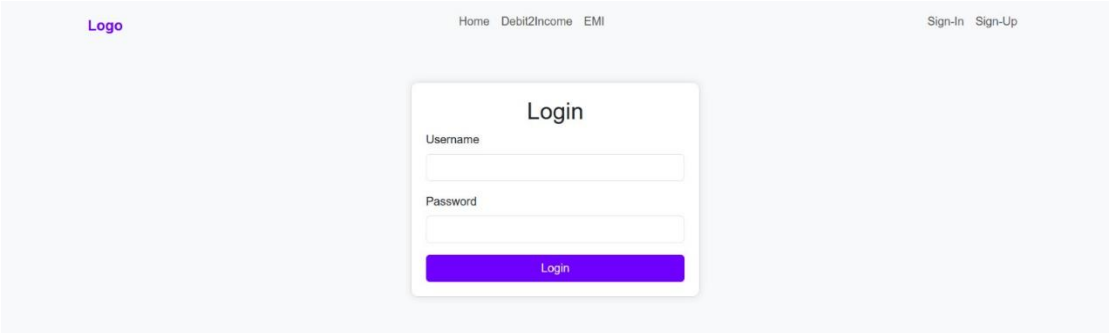


Figure 4.5: Login Page

Pseudocode:

```
    <div class="container login-container">
<h2 class="text-center">Login</h2>
<form id="loginForm" method="post" action="signin">
  {% csrf_token %}
  <div class="mb-3">
    <label for="username" class="form-label">Username</label>
   <input name="username" type="text" class="form-control"
   id="username" required>
  </div>
  <div class="mb-3">
    <label for="password" class="form-label">Password</label>
   <input name='password' type="password" class="form-control"
   id="password" required>
  </div>
  <button type="submit" class="btn btn-primary w-100">Login</button>
</form>
    </div>
```

Description: The login page requires username and password as input.

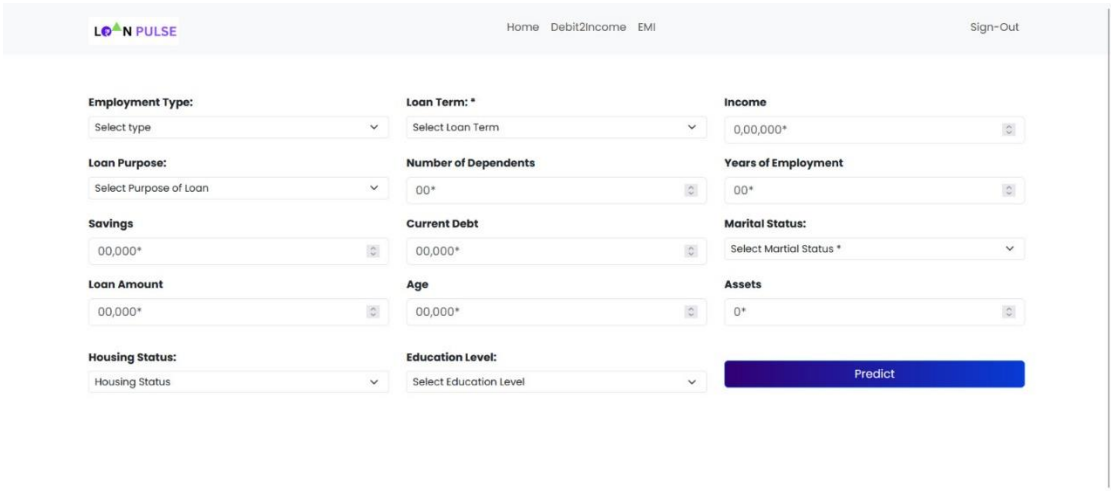


Figure 4.6: Enter Details Page

4.5. Testing

Testing is the process of evaluation of a system to detect differences between given input and expected output and also to assess the feature of the system. Testing assesses the quality of the product. It is a process that is done during the development process.

4.5.1. Strategy Used

Here's how testing can be used in the loan status prediction system:

- The dataset is typically divided into two subsets: a training set and a test set. The training set is used to train the predictive model, while the test set is used to evaluate its performance. This split helps assess how well the model generalizes to new, unseen data.

- Various evaluation metrics can be used to assess the performance of the model on the test set. For binary classification problems like loan status prediction (e.g.,

approved or not approved), common metrics include accuracy, precision, F1 score etc are used.

- Once the model is deployed in a real-world setting, ongoing testing and monitoring are necessary to ensure its continued effectiveness. This involves periodically evaluating the model's performance on new data and detecting any degradation in performance that may require retraining or updating the model.

4.5.2. Results

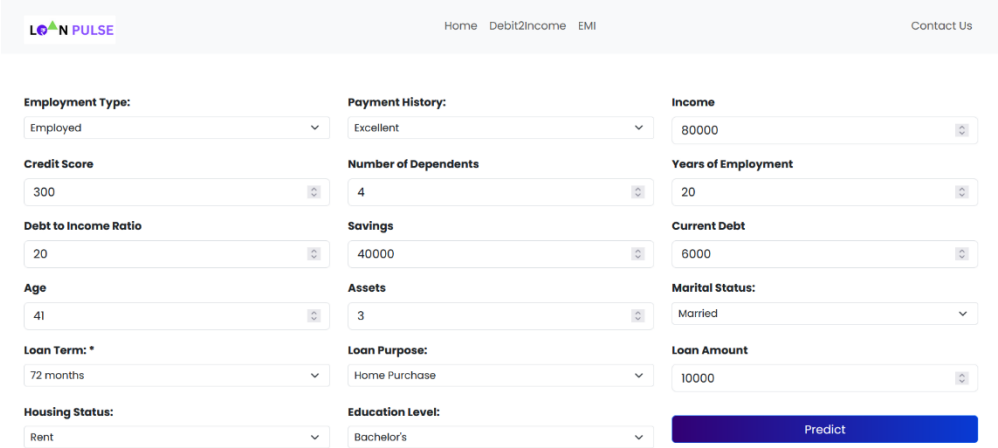


Figure 4.7: Sample Input 1

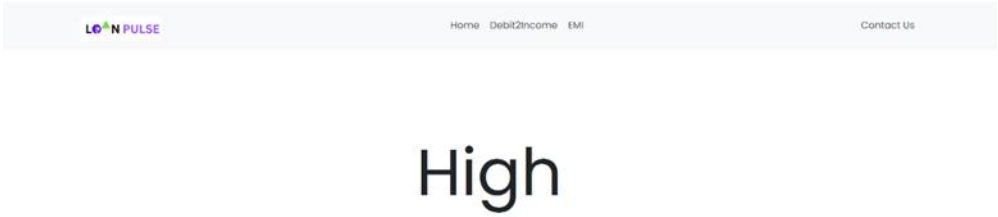


Figure 4.8: Output 1

4.5.3. Testcases – Unit Testing

Table 4.1: Testcase 1

Credit Score	**Income**	**DTI**	**Employment Status**	**CU Ratio**	**Age**	**Output**
600	100000	0.25	Employed	0.25	22	High
Loan Amount	**Current Debt**	**Loan Term**	**Purpose of Loan**	**Educational Level**	**Years of Employment**	
200000	400000	24 M	Car Repair	Bachelor's	2	
Payment History	**Savings**	**Assets**	**Marital Status**	**Housing Status**	**Number of Dependents**	
Excellent	250000	500000	Single	Rent	4	

Table 4.2: Testcase 2

Credit Score	**Income**	**DTI**	**Employment Status**	**CU Ratio**	**Age**	**Output**
350	50000	0.25	Unemployed	0.25	35	Low
Loan Amount	**Current Debt**	**Loan Term**	**Purpose of Loan**	**Educational Level**	**Years of Employment**	
800000	200000	36 M	Business	PhD	10	
Payment History	**Savings**	**Assets**	**Marital Status**	**Housing Status**	**Number of Dependents**	
Fair	200000	150000	Married	Rent	8	

Table 4.3: Testcase 3

Credit Score	**Income**	**DTI**	**Employment Status**	**CU Ratio**	**Age**	**Output**
750	300000	0.15	Self-employed	0.14	25	High
Loan Amount	**Current Debt**	**Loan Term**	**Purpose of Loan**	**Educational Level**	**Years of Employment**	
600000	0	48 M	Car Purchase	Bachelor's	3	
Payment History	**Savings**	**Assets**	**Marital Status**	**Housing Status**	**Number of Dependents**	
Fair	100000	500000	Single	Own	2	

CHAPTER 5:

CONCLUSION

CHAPTER 5. CONCLUSION

5.1. Conclusion

In conclusion, a loan status prediction system serves as a vital tool in the lending industry, offering valuable insights into the creditworthiness of applicants and assisting financial institutions in making informed decisions. By leveraging advanced machine learning techniques and predictive analytics, such systems can effectively assess the risk associated with loan applications, identify potential defaulters, and optimize lending strategies. Through the integration of comprehensive data analysis, model training, and rigorous testing, these systems provide accurate, reliable, and actionable predictions, ultimately enabling lenders to mitigate risk, enhance operational efficiency, and foster responsible lending practices.

5.2. Limitations of the Work

- Since no task can be 100% perfect. The same applies to this project as the predicted status is not 100% accurate.
- The dataset used for loan status prediction consists of limited number of entries and is predefined dataset with static number of entries.
- The model is trained on a dataset that consists of 18 attributes. So the user needs to upload only the entries for those attributes and not anything else otherwise it may give wrong output.

5.3. Suggestion and Recommendations for Future Work

- The future scope of this idea is to scale up the dataset and accommodate more attributes and entries in it.

- Integrating the system with real-time data streams can lead to fast and low latency detection of the status of loan.
- Developing a user-friendly mobile app to complement the web interface would make it more accessible to users allowing them to easily upload details without requiring the need of laptop.

BIBLIOGRAPHY

BIBLIOGRAPHY

- Shoumo S Z H, Dhruba M I M, Hossain S, Ghani N H, Arif H and Islam S 2019 Application of machine learning in credit risk assessment: a prelude to smart banking TENCON 2019 – 2019 IEEE Region 10 Conf. (TENCON) pp 2023–8
- Orgler, Y. E. (1971). Evaluation of bank consumer loans with credit scoring models.
- Cyert, R. M., Davidson, H. J., & Thompson, G. L. (1962). search 6, 180–195. Estimation of allowance for doubtful accounts by Markov chains. Management Science 8, 287–303.

APPENDIX A:

SOURCE CODE

APPENDIX A: SOURCE CODE

S.1. Frontend Code

S.1.1. base.html

```
{% load static %}
<!doctype html>
<html lang="en">

<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;400;600;7
00&display=swap" rel="stylesheet">
  <!-- Bootstrap CSS -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css
" rel="stylesheet" integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2
HN" crossorigin="anonymous">      <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.3.0/css/all.min.css">
```

```
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/ionicons/2.0.1/css/ionicons.min.c
ss">

    <link rel = "stylesheet" href =
"https://cdnjs.cloudflare.com/ajax/libs/bttn.css/0.2.4/bttn.css">

    <title>LoanGuard</title>
    <style>
        .bg {
            height: 100%;
            /* Center and scale the image nicely */
            background-position: center;
            background-repeat: no-repeat;
            background-size: cover;
        }

        .form-label{
            font-weight: bold;
        }

        body {
            background-color:white;
            /* background: -webkit-linear-gradient(to right, #66a776, #043610);
            background: linear-gradient(to right, #5d9c6a, #059b0d); */
            height: 100vh;
            font-family: 'Poppins', sans-serif;
            font-size: 15px;
        }

        .rounded-lg {
```

```
        border-radius: 1rem;
      }

      .custom-file-label.rounded-pill {
        border-radius: 50rem;
      }

      .custom-file-label.rounded-pill::after {
        border-radius: 0 50rem 50rem 0;
      }
      #submit{  width:100%;
        background: linear-gradient(to right, #320074, #083bd4); }
      option :hover{
        background-color: #6f00ff;
        color:#f1f3f5;
      }

      .footer-basic {
        margin-top: 10%;
        padding: 40px 0;
        background-color: #042908;
        color: #f1f3f5;
      }
      {% block extra_css %}{% endblock %}
    </style>
</head>

<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light"
style="background-color: white;">
      <div class="container">
```

```
        <a class="navbar-brand" href="#" style="font-size: 20px;font-weight:
600;color:  #6f00ff;margin-top: 10px;"><img src="{% static 'images/image.png'
%}" style="width: 125px; margin-right: 10px;" srcset=""></a>
        <div class="collapse navbar-collapse" id="navbarNav">


          <ul class="navbar-nav mx-auto">
            <li class="nav-item active">
              <a class="nav-link" href="/">Home</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/dbtinc">Debit2Income</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="/EMI">EMI</a>
            </li>
          </ul>
          <ul class="navbar-nav ml-auto">
            <li class="nav-item">
              <a class="nav-link" href="./contact">Contact Us</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
    {% block body %}

    {% endblock body %}
    <div class="footer-basic">

    </div>
```

```
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></scrip
t>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.3/js/bootstrap.bundle.min.js"></script>
  <script>
    var value = document.getElementById("theInput").value;
    console.log(value);
    </script>
</body>
</html>
```

S.1.2. index.html

```
{% extends "base.html" %}
<!DOCTYPE html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Home Page</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css
">
  <style>
    {% block extra_css %}
    body {
      font-family: 'Poppins', sans-serif;
      font-size: 15px;
    }
```

```
.hero-section {
    background: linear-gradient(90deg, rgba(2,0,36,1) 0%, rgba(9,9,121,1) 35%, rgba(0,212,255,1) 100%);
    background-image: url('https://img.freepik.com/premium-vector/abstract-background-design-perfect-landing-page-background-other-you-want_288336-2.jpg');
    background-size: cover;
    background-position: center;
    height: 75vh;
    display: flex;
    justify-content: center;
    align-items: center;
    text-align: center;
    color: white;
}

.hero-section h1 {
    font-size: 3rem;
    margin-bottom: 20px;
}

.hero-section button {
    padding: 10px 20px;
    font-size: 1.2rem;
    background-color: #357deb;
    border: 2px solid white;
    border-radius: 30px;
    color: white;
    cursor: pointer;
}
```

```
.hero-section button:hover {
    background-color: #320074;
}

.sub-section {
    padding: 50px 20px;
    padding-bottom: 75;
    text-align: center;
    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
    height: 30vh;
    border-radius: 35px;
    margin-top: 10%;
    margin-bottom: 25%;
}

.sub-section h2 {
    font-size: 1.5rem;
    margin-bottom: 20px;
    font-weight: bold;
}

.sub-section a {
    display: block;
    font-size: 1.2rem;
    margin-top: 10px;
    color: #6f00ff;
    text-decoration: none;
}

.sub-section a:hover {
    text-decoration: underline;
```

```
    }
    {% endblock %}
  </style>
</head>
<body>
  {% block body %}
  <div class="hero-section">
    <div>
      <h1>Welcome to LoanPulse</h1>
      <p>Your financial solution partner</p>
      <a href="/home"><button onclick="">Get Started</button></a>
    </div>
  </div>
<br>
  <div class="container">
    <div class="row">
      <div class="col-md-4" >
        <div class="sub-section" >
          <h2>EMI Calculator</h2>
          <p>Calculate your Equated Monthly Installment (EMI) for loans.</p>
          <button class="btn btn-outline-primary" href="/emi.html">Calculate
EMI</button>
        </div>
      </div>
      <div class="col-md-4">
        <div class="sub-section" >
          <h2>Debt2Income Ratio Calculator</h2>
          <p>Calculate your Debt to Income ratio to manage your finances
better.</p>
          <button class="btn btn-outline-primary"
href="/dbtinc.html">Calculate DTI</button>
```

```
            </div>
          </div>
          <div class="col-md-4">
            <div class="sub-section">
              <h2>Cibil Score</h2>
              <p>Check your Cibil Score to understand your creditworthiness.</p>
              <button class="btn btn-outline-primary" href="/cibil.html">Check
Cibil Score</button>
            </div>
          </div>
        </div>
    </div>
    <div class="footer-section">
      <div class="container">
        <p>&copy; 2024 LoanPulse. All rights reserved.</p>
      </div>
    </div>
    {% endblock body %}
    </body>
</html>
```

S.1.3. home.html

```
{% extends 'base.html' %}
{% block pagetitle %}
{% endblock pagetitle %}
{% block body %}
      <div class="container my-3">
            <form method="POST">
                  {% csrf_token %}
                  <div class="row" style="padding-top: 3%;">
```

```
                    <div class="col-md-4">
                        <label class="form-label">Employment Type:</label>
                        <select class="form-select form-select-sm" aria-label="Small
select example" name="Employment_Status" required>
              <option selected>Select type</option>
              <option value="Employed">Employed</option>
              <option value="Self-employed">Self-employed</option>
              <option value="Unemployed">Unemployed</option>
            </select>
                    </div>
                    <div class="col-md-4">
                        <label class="form-label">Payment History:</label>
                        <select class="form-select form-select-sm" aria-label="Small
select example" name="Payment_History" required>
              <option selected>Select Payment History</option>
              <option value="Excellent">Excellent</option>
              <option value="Good">Good</option>
              <option value="Fair">Fair</option>
              <option value="Poor">Poor</option>
            </select>
                    </div>
                    <div class="col-md-4">
                      <label class="form-label">Income</label>
                      <input type="number" name="Annual Income" class="form-
control" aria-describedby="passwordHelpInline"   placeholder="0,00,000*"
required>
                    </div>
                  </div>
                  <!-- Add more rows and elements similarly -->
                  <div class="row mt-3">
```

```
                    <div class="col-md-4">
                     <label class="form-label">Credit Score</label>
                     <input type="number" name="Credit_Score" class="form-
control" aria-describedby="passwordHelpInline"  placeholder="0000*"
required>
                </div>
                    <div class="col-md-4">
                        <label class="form-label">Number of Dependents</label>
                        <input type="number" name="Number_of_Dependents"
class="form-control" aria-describedby="passwordHelpInline
"  placeholder="00*"required>
                    </div>
                    <div class="col-md-4">
                        <label class="form-label">Years of Employment</label>
                        <input type="number" name="Years_of_Employment"
class="form-control" aria-
describedby="passwordHelpInline"  placeholder="00*" required>
                    </div>
                </div>
                <div class="row mt-3">
                    <div class="col-md-4">
                        <label class="form-label">Debt to Income Ratio</label>
                        <input type="number" step="0.01"
name="Debt_to_Income_Ratio" class="form-control" aria-
describedby="passwordHelpInline" placeholder="0.0*" required>
                    </div>
                    <div class="col-md-4">
                        <label class="form-label">Savings</label>
                        <input type="number" name="Savings" class="form-
control" aria-describedby="passwordHelpInline" placeholder="00,000*"
required>
```

```
                    </div>
                    <div class="col-md-4">
                        <label class="form-label">Current Debt</label>
                        <input type="number" name="Current_Debt" class="form-
control" aria-describedby="passwordHelpInline" placeholder="00,000*"
required>
                    </div>
                </div>
                <div class="row mt-3">
                    <div class="col-md-4">
                        <label class="form-label">Age</label>
                        <input type="number" name="Age" class="form-control"
aria-describedby="passwordHelpInline" placeholder="00,000*" required>
                    </div>
                    <div class="col-md-4">
                        <label class="form-label">Assets</label>
                        <input type="number" name="Assets" class="form-control"
aria-describedby="passwordHelpInline"placeholder="0*" required>
                    </div>
                    <div class="col-md-4">
                      <label class="form-label">Marital Status:</label>
                      <select class="form-select form-select-sm" aria-label="Small
select example" name="Marital_Status">
          <option selected>Select Martial Status *</option>
          <option value="Single">Single</option>
          <option value="Married">Married</option>
        </select>
                     </div>
                </div>
                <div class="row mt-3">
                    <div class="col-md-4">
```

```
                    <label class="form-label">Loan Term: *</label>
                    <select class="form-select form-select-sm" aria-label="Small
select example" name="Loan_Term">
            <option selected>Select Loan Term</option>
            <option value="6 months">6 months</option>
            <option value="12 months">12 months</option>
            <option value="18 months">18 months</option>
            <option value="24 months">24 months</option>
            <option value="36 months">36 months</option>
            <option value="48 months">48 months</option>
            <option value="60 months">60 months</option>
            <option value="72 months">72 months</option>
            <option value="84 months">84 months</option>
          </select>
                </div>
                <div class="col-md-4">
                  <label class="form-label">Loan Purpose:</label>
                  <select class="form-select form-select-sm" aria-label="Small
select example" name="Loan_Purpose">
          <option selected>Select Purpose of Loan</option>
          <option value="Business">Business</option>
          <option value="Car Purchase">Car Purchase</option>
          <option value="Car Repair">Car Repair</option>
          <option value="Debt consolidation">Debt consolidation</option>
          <option value="Home improvement">Home improvement</option>
          <option value="Home Purchase">Home Purchase</option>
          <option value="Home Repair">Home Repair</option>
          <option value="Medical">Medical</option>
          <option value="Vacation">Vacation</option>
        </select>
                </div>
```

```
                <div class="col-md-4">
                  <label class="form-label">Loan Amount</label>
                  <input type="number" name="Loan_Amount" class="form-
control" aria-describedby="passwordHelpInline" placeholder="00,000*"
required>
              </div>
                </div>
                <div class="row mt-3">
                  <div class="col-md-4">
                    <label class="form-label">Housing Status:</label>
                    <select class="form-select form-select-sm" aria-label="Small select
example" name="Housing_Status">
      <option selected>Housing Status</option>
      <option value="Rent">Rent</option>
      <option value="Own">Own</option>
    </select>
              </div>
                    <div class="col-md-4">
                        <label class="form-label">Education Level:</label>
                        <select class="form-select form-select-sm" aria-label="Small
select example" name="Education_Level">
              <option selected>Select Education Level</option>
              <option value="PhD">PhD</option>
              <option value="Bachelors">Bachelor's</option>
              <option value="Masters">Master's</option>
              <option value="Associate">Associate's</option>
              <option value="High School">High School</option>
            </select>
                    </div>
                    <div class="col-md-4 mt-3">
```

```
                        <button type="submit" class="btn btn-primary"
id="submit">Predict</button>
                    </div>
                </div>
            </form>
        </div>
        {% endblock body %}
```

S.1.4. final.html

```
{% extends 'base.html' %}
{% block pagetitle %}
{% endblock pagetitle %}

{% block body %}
<div class="container my-3">
   <h1 id="theInput" style="margin-top:10%;margin-left : 40%; margin-bottom :
25% ;font-size: 100px;">{{ct}}</h1>
</div>
{% endblock body %}
```

S.1.5. emi.html

```
{% extends "base.html" %}
<!DOCTYPE html>
<html lang="en">
<head>
   <!-- Required meta tags -->
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <title>EMI Calculator</title>
```

```
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css
">
  <style>
    {% block extra_css %}
    body {
      font-family: 'Poppins', sans-serif;
      font-size: 15px;
    }

    .container-box {
      border: 2px solid #6f00ff;
      border-radius: 15px;
      padding: 20px;
      margin-top: 30px;
    }

    .input-group {
      margin-bottom: 15px;
    }

    label {
      font-weight: bold;
    }

    button {
      width: 100%;
      background: linear-gradient(to right, #320074, #083bd4);
      color: #fff;
      border: none;
      border-radius: 5px;
```

```
        cursor: pointer;
      }

      button:hover {
        background: linear-gradient(to right, #6f00ff, #320074);
      }

      #result {
        margin-top: 15px;
        font-weight: bold;
      }
      {% endblock %}
    </style>
</head>
<body>
    {%block body%}

    <div class="container">
      <div class="row container-box">
        <div class="col-md-6">
          <div >
            <h1>EMI Calculator</h1>
            <div class="input-group">
              <label for="loanAmount">Loan Amount (₹):</label>
              <input type="number" id="loanAmount" class="form-control"
min="0" step="any">
            </div>
            <div class="input-group">
              <label for="interestRate">Interest Rate (%):</label>
              <input type="number" id="interestRate" class="form-control"
min="0" step="any">
```

```
            </div>
            <div class="input-group">
              <label for="loanTenure">Loan Tenure (Months):</label>
              <input type="number" id="loanTenure" class="form-control"
min="0" step="any">
            </div>
            <button onclick="calculateEMI()">Calculate</button>
          </div>
        </div>
        <div class="col-md-6">
          <div>
            <div id="result">
              <p id="calculatedEMI">Your Equated Monthly Installment (EMI) will
appear here after calculation.</p>
            </div>
          </div>
        </div>
      </div>
    </div>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></scrip
t>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.3/js/bootstrap.bundle.min.js"></script>
    <script>
      function calculateEMI() {
        const loanAmount =
parseFloat(document.getElementById('loanAmount').value);
        const interestRate =
parseFloat(document.getElementById('interestRate').value);
```

```
        const loanTenure =
parseFloat(document.getElementById('loanTenure').value);

        if (isNaN(loanAmount) || isNaN(interestRate) || isNaN(loanTenure)) {
           alert('Please enter valid numbers.');
           return;
        }
        const monthlyInterestRate = (interestRate / 100) / 12;
        const emi = loanAmount * monthlyInterestRate * Math.pow(1 +
monthlyInterestRate, loanTenure) / (Math.pow(1 + monthlyInterestRate,
loanTenure) - 1);
        document.getElementById('calculatedEMI').innerText = `Your Equated
Monthly Installment (EMI) is ₹${emi.toFixed(2)}.`;
      }
   </script>
   {%endblock%}
</body>
</html>
```

S.1.6. dbtinc.html

```
{% extends "base.html" %}
<!DOCTYPE html>
<html lang="en">
<head>
   <!-- Required meta tags -->
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <title>Debt to Income Ratio Calculator</title>
   <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css
">
```

```
<style>
  {% block extra_css %}
  body {
    font-family: 'Poppins', sans-serif;
    font-size: 15px;
  }

  .container-box {
    border: 2px solid #6f00ff;
    border-radius: 15px;
    padding: 20px;
    margin-top: 30px;
  }

  .input-group {
    margin-bottom: 15px;
  }

  label {
    font-weight: bold;
  }

  button {
    width: 100%;
    background: linear-gradient(to right, #320074, #083bd4);
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
  }
```

```
      button:hover {
        background: linear-gradient(to right, #6f00ff, #320074);
      }

      #result {
        margin-top: 15px;
        font-weight: bold;
      }
      {% endblock %}
    </style>
</head>
<body>
    {% block body %}

    <div class="container">
      <div class="row container-box">
        <div class="col-md-6">
          <div >
            <h2>Debit2Income</h2>
            <break></break>
            <div class="input-group">
              <label for="monthlyIncome">Monthly Income (₹):</label>
              <input type="number" id="monthlyIncome" class="form-control"
min="0" step="any">
            </div>
            <div class="input-group">
              <label for="monthlyDebt">Monthly Debt (₹):</label>
              <input type="number" id="monthlyDebt" class="form-control"
min="0" step="any">
            </div>
            <button onclick="calculateDTI()">Calculate</button>
```

```
            </div>
          </div>
          <div class="col-md-6">
            <div>
              <div id="result">
                <h1 id="dtiResult">0%</h1>

                <p id="calculatedDTI">Your Debt to Income ratio is 0%. </p>
                <p id="suggestion">Suggestions will appear here after
calculation.</p>
              </div>
            </div>
          </div>
        </div>
      </div>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></scrip
t>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.3/js/bootstrap.bundle.min.js"></script>
    <script>
        function calculateDTI() {
            const monthlyIncome =
parseFloat(document.getElementById('monthlyIncome').value);
            const monthlyDebt =
parseFloat(document.getElementById('monthlyDebt').value);

            if (isNaN(monthlyIncome) || isNaN(monthlyDebt)) {
                alert('Please enter valid numbers.');
                return;
```

```
        }

        const dtiRatio = (monthlyDebt / monthlyIncome) * 100;
        let resultText = '';
        let suggestion = '';
        if (dtiRatio <= 36) {
            resultText = `Your Debt to Income ratio is ${dtiRatio.toFixed(2)}%. Great job! Your debt is within a healthy range.`;
            suggestion = 'No suggestions at the moment.';
        } else {
            resultText = `Your Debt to Income ratio is ${dtiRatio.toFixed(2)}%. Your debt is higher than recommended. Consider reducing it.`;
            suggestion = 'Consider reducing your debt to improve your financial health.';
        }
        document.getElementById('calculatedDTI').innerText = resultText;
        document.getElementById('suggestion').innerText = suggestion;
        document.getElementById('dtiResult').innerText = `${dtiRatio.toFixed(2)}%`;
      }
    </script>
    {% endblock %}
</body>
</html>
```

S.2. Backend Code

S.2.1. urls.py

```
from django.contrib import admin
from django.urls import path
from backendApp import views

from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index),
    path('home/', views.home),
    path('dbtinc/', views.dbtinc),
    path('EMI/',views.emi),
]
urlpatterns += static(settings.STATIC_URL,
document_root=settings.STATIC_ROOT)
```

S.2.2. settings.py

```
"""
Django settings for LoanProject project.

Generated by 'django-admin startproject' using Django 5.0.1.

For more information on this file, see
https://docs.djangoproject.com/en/5.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/5.0/ref/settings/
"""

from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
```

```
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/5.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-&^_1idhhjddtcp($lq((xgxc-
%t4t@j7=2#u*_zd0z*uvobx3$'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'backendApp',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'LoanProject.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
```

```
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'LoanProject.wsgi.application'

# Database
# https://docs.djangoproject.com/en/5.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Password validation
# https://docs.djangoproject.com/en/5.0/ref/settings/#auth-password-
validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/5.0/topics/i18n/

LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.0/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/5.0/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

S.2.3. Model Training Code

```
!pip install numpy
!pip install pandas

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df =
pd.read_csv(r"C:\Users\Jaideep\machine_learning_project\loan_status_predictio
n\data.csv")

df.head()

df.shape

df.info()

print("Number of duplicated rows: ", len(df[df.duplicated()]))

#Finding correlation between the columns
df.corr()

def count_features(data):
    numerical = data.select_dtypes(include = ['int64']).columns
    objects = data.select_dtypes(include = ['object']).columns
    floating = data.select_dtypes(include = ['float']).columns
```

```
    print("Object features are : ", len(objects))
    print("Numerical features are : ", len(numerical))
    print("Floating features are : ", len(floating))

count_features(df)

#Number of missing values
print(df.isnull().sum())

# No missing values present. Now categorical balues need to be converted into
numerical values
#Finding number of unique vlues for each object type
def unique_values(data):
    objects = data.select_dtypes(include = ['object']).columns
    for col in objects:
        unique = data[col].value_counts()
        print(unique, '\n')
        print('*' *50)

unique_values(df)

#Expanding dataset
import random
num_additional_rows = 5000
additional_data = []

for _ in range(num_additional_rows):
    # Generate dummy values for each column
    new_row = {
        'ID': len(df) + 1,  # Assuming ID is incremented sequentially
        'Credit_Score': random.randint(500, 800),
        'Income': random.randint(20000, 100000),
        'Employment_Status': random.choice(['Employed', 'Self-employed',
'Unemployed']),
        'Debt_to_Income_Ratio': round(random.uniform(0.1, 0.6), 2),
        'Loan_Amount': random.randint(5000, 50000),
        'Loan_Term': random.choice([24, 36, 48, 60]),
        'Payment_History': random.choice(['Poor', 'Fair', 'Good', 'Excellent']),
        'Age': random.randint(20, 60),
        'Loan_Purpose': random.choice(['Vacation','Debt consolidation','Home
improvement','Home repair','Business','Medical','Car repair','Education','Home
purchase','Car purchase']),
        'Assets': random.randint(1000, 80000),
        'Savings': random.randint(500, 30000),
        'Housing_Status': random.choice(['Rent', 'Own']),
        'Marital_Status': random.choice(['Single', 'Married', 'Divorced']),
```

```
        'Education_Level': random.choice(["High School", "Associate's", "Bachelor's",
"Master's", "PhD"]),
        'Number_of_Dependents': random.randint(0, 5),
        'Years_of_Employment': random.randint(1, 20),
        'Current_Debt': random.randint(0, 20000),
        'Credit_Utilization_Ratio': round(random.uniform(0.1, 0.8), 2),
        'Default': random.randint(0, 1)
    }
    additional_data.append(new_row)

# Append the additional rows to the existing DataFrame
df = df.append(pd.DataFrame(additional_data), ignore_index=True)

df.shape

num_additional_rows = 5000
additional_data = []

for _ in range(num_additional_rows):
    # Generate dummy values for each column
    new_row = {
        'ID': len(df) + 1,  # Assuming ID is incremented sequentially
        'Credit_Score': random.randint(500, 800),
        'Income': random.randint(20000, 100000),
        'Employment_Status': random.choice(['Employed', 'Self-employed',
'Unemployed']),
        'Debt_to_Income_Ratio': round(random.uniform(0.1, 0.6), 2),
        'Loan_Amount': random.randint(5000, 50000),
        'Loan_Term': random.choice([24, 36, 48, 60]),
        'Payment_History': random.choice(['Poor', 'Fair', 'Good', 'Excellent']),
        'Age': random.randint(20, 60),
        'Loan_Purpose': random.choice(['Vacation','Debt consolidation','Home
improvement','Home repair','Business','Medical','Car repair','Education','Home
purchase','Car purchase']),
        'Assets': random.randint(1000, 80000),
        'Savings': random.randint(500, 30000),
        'Housing_Status': random.choice(['Rent', 'Own']),
        'Marital_Status': random.choice(['Single', 'Married', 'Divorced']),
        'Education_Level': random.choice(["High School", "Associate's", "Bachelor's",
"Master's", "PhD"]),
        'Number_of_Dependents': random.randint(0, 5),
        'Years_of_Employment': random.randint(1, 20),
        'Current_Debt': random.randint(0, 20000),
        'Credit_Utilization_Ratio': round(random.uniform(0.1, 0.8), 2),
        'Default': random.randint(0, 1)
    }
    additional_data.append(new_row)
```

```
# Append the additional rows to the existing DataFrame
df = df.append(pd.DataFrame(additional_data), ignore_index=True)

df.shape

df

# Perform Label Encoding
df_encoded = pd.get_dummies(df, columns=['Employment_Status',
'Payment_History', 'Loan_Purpose', 'Housing_Status', 'Marital_Status',
'Education_Level'])

# Save the mapping of categorical values to encoded counterparts
encoding_mapping = {}
for column in ['Employment_Status', 'Payment_History', 'Loan_Purpose',
'Housing_Status', 'Marital_Status', 'Education_Level']:
    encoding_mapping[column] = dict(zip(df[column], df_encoded[[col for col in
df_encoded.columns if col.startswith(column)]].columns))

# Print encoding mapping
for column, mapping in encoding_mapping.items():
    print(f"{column} encoding mapping:")
    print(mapping)
    print()


from sklearn.preprocessing import LabelEncoder

# Copy DataFrame to avoid modifying the original data
df_label_encoded = df.copy()

# Initialize LabelEncoder
label_encoders = {}

# Initialize a dictionary to store mappings
encoded_mappings = {}

# Iterate over categorical columns and apply Label Encoder
for column in df_label_encoded.select_dtypes(include=['object']).columns:
    label_encoders[column] = LabelEncoder()
    df_label_encoded[column] =
label_encoders[column].fit_transform(df_label_encoded[column])
    # Store the mappings
    encoded_mappings[column] = dict(zip(label_encoders[column].classes_,
label_encoders[column].transform(label_encoders[column].classes_)))
```

```
# Print label mappings
for column, encoder in label_encoders.items():
    print(f"{column}: {encoded_mappings[column]}")


def count_features(data):
    numerical = data.select_dtypes(include = ['int64']).columns
    objects = data.select_dtypes(include = ['object']).columns
    floating = data.select_dtypes(include = ['float']).columns


    print("Object features are : ", len(objects))
    print("Numerical features are : ", len(numerical))
    print("Floating features are : ", len(floating))

count_features(df_label_encoded)

df_label_encoded.shape

# Training model
X=df_label_encoded.drop('Default',axis='columns')
Y=df_label_encoded.Default

X

Y

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.30, random_state
= True)

from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score

from sklearn.linear_model import LogisticRegression
logistic_regression = LogisticRegression(penalty='l1', solver='liblinear')
logistic_regression.fit(X_train, Y_train)
Y_predicted_logistic = logistic_regression.predict(X_test)
print("Accuracy Score: ", accuracy_score(Y_test,Y_predicted_logistic))
print("Precision Score : ",precision_score(Y_test,Y_predicted_logistic))

from sklearn.tree import DecisionTreeClassifier
decision_tree = DecisionTreeClassifier(max_depth = 5)
decision_tree.fit(X_train, Y_train)
Y_predicted_decisionTree = decision_tree.predict(X_test)
accuracy_dt = accuracy_score(Y_test,Y_predicted_decisionTree)
precision_dt = precision_score(Y_test,Y_predicted_decisionTree)
```

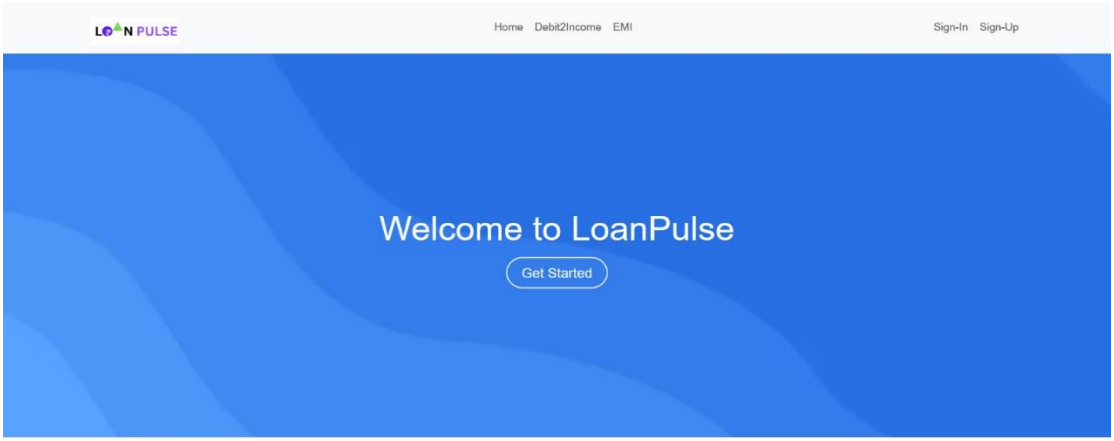
```
print("Acuuracy : ", accuracy_dt)
print("Precision : ", precision_dt)

from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(X_train, Y_train)
Y_pred_rf = rf.predict(X_test)
accuracy_rf = accuracy_score(Y_test, Y_pred_rf)
precision_rf = precision_score(Y_test,Y_predicted_decisionTree)
print("Accuracy:", accuracy_rf)
print("Precision : ", precision_rf)
```

APPENDIX B:

USER MANUAL

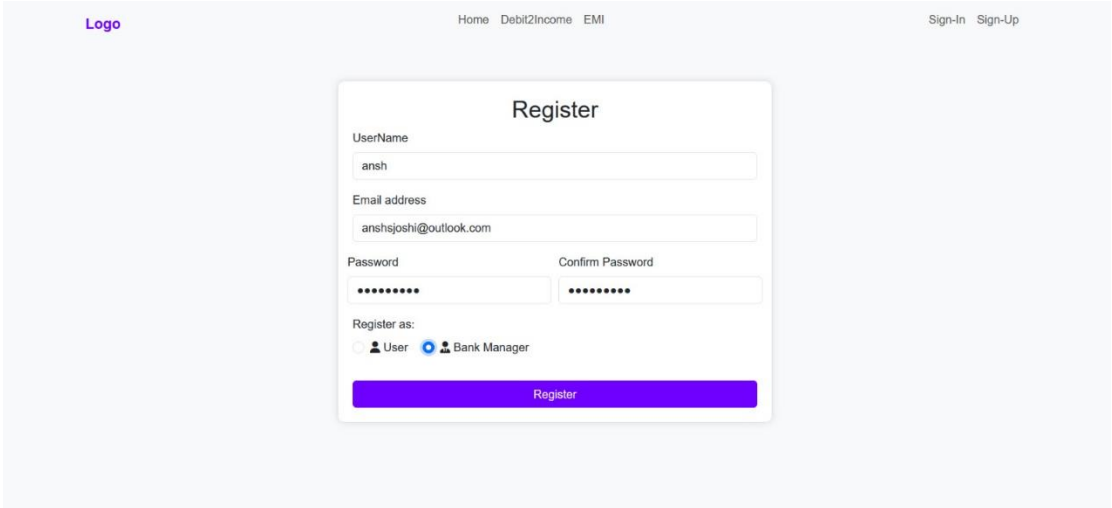
APPENDIX B: USER MANUAL

- Welcome to loan pulse website.
- The user will see this on the home page on the screen.

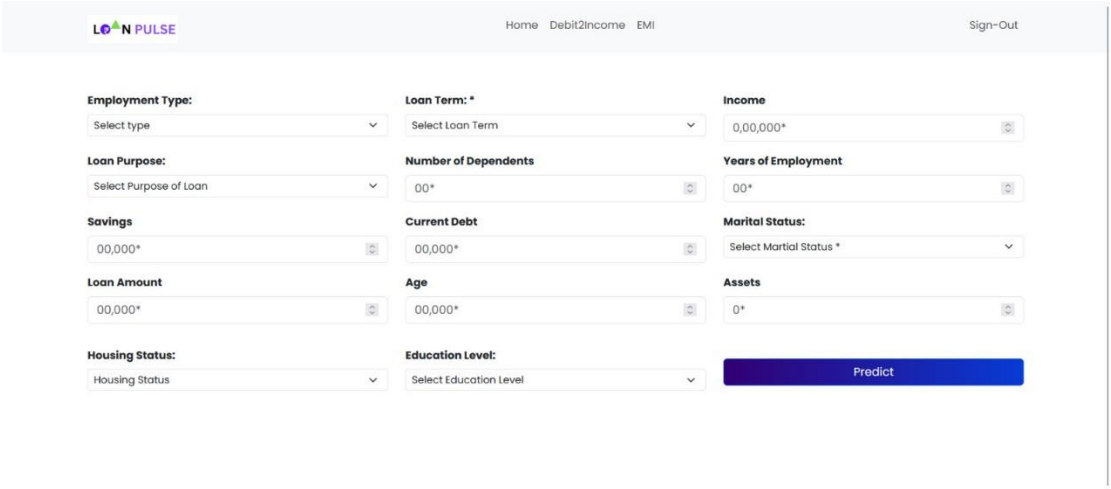


- If there is a new user, he will be required to register first. The navbar on the website has options for sign up, sign in and sign out.

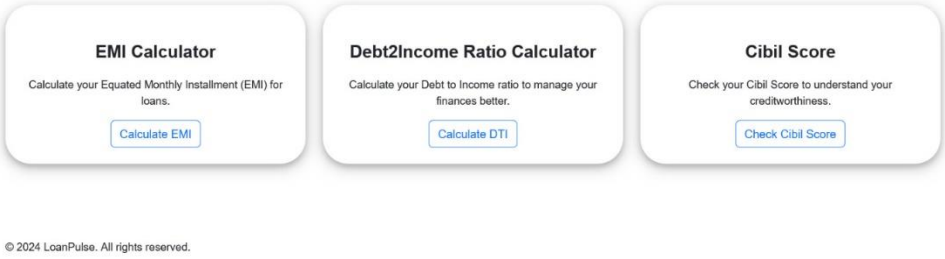
- First the user will sign up i.e. register himself as a user or a bank manager on the website. He will be required to enter username and password for this purpose.



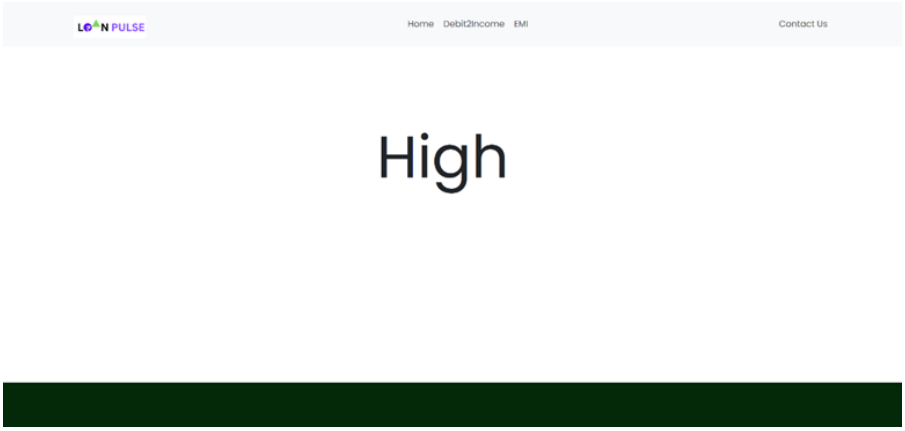
- Next, there occurs a page i.e. a form where input is given in the form of details of the user.



- There are options present in some input fields and some fields require the input manually by the user.
- The input fields like EMI, Cibil Score and debt to income ratio can be calculated by using the calculators provided on the home page.



- After successfully submitting all the details, the output page shows the result as high or low.



This indicates whether the chances of approving the loan are high or low.

APPENDIX C:

RESEARCH PAPER WITH CERTIFICATES

APPENDIX D:

POSTER