PREDICTIVE MODELLING FOR LOAN APPROVAL

A PROJECT REPORT

submitted by

AKHNA S J

TVE23MCA-2008

to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the degree

of

Master of Computer Applications

Department of Computer Applications

College of Engineering

Trivandrum-695016

MAY 2025

Declaration

I undersigned hereby declare that the project report titled **"PREDICTIVE MODELLING FOR LOAN APPROVAL"** submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Sunitha S, Professor. This submission represents my ideas in my words and where ideas or words of others have been included. I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity as directed in the ethics policy of the college and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and/or University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title.

Place : Trivandrum **Akhna S J**

Date : 02/04/2025

DEPARTMENT OF COMPUTER APPLICATIONS

COLLEGE OF ENGINEERING
TRIVANDRUM

CERTIFICATE

This is to certify that the report entitled **Predictive Modelling for Loan Approval** submitted by **Akhna S J** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by him under my guidance and supervision. This report in any form has not been submitted to any University or Institute for any purpose.

Internal Supervisor External Supervisor

Head of the Dept

Acknowledgement

First and for most I thank **GOD** almighty and to my parents for the success of this project. I owe a sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely thankful to **Dr Suresh K**, Principal, College of Engineering Trivandrum for providing me with the best facilities and atmosphere which was necessary for the successful completion of this project.

I am extremely grateful to **Prof. Liji P I**, HOD, Dept of Computer Applications, for providing me with best facilities and atmosphere for the creative work guidance and encouragement and also for his valuable guidance, support and advice that aided in the successful completion of my project.

I express our sincere thanks to **Prof. Sunitha D**, Asst. Professor, Department of Computer Applications, College of Engineering Trivandrum for her valuable guidance, support and advice that aided in the successful completion of my project.

I profusely thank other faculty members in the department and all other staffs of CET, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project. No words can express my humble gratitude to my beloved parents and relatives who have been guiding me in all walks of my journey.

Akhna S J

Abstract

Access to credit is a fundamental pillar of economic growth, enabling individuals and businesses to achieve their financial goals. However, traditional loan approval processes often suffer from inefficiencies, subjectivity, and delays due to manual verification and rigid eligibility criteria. This project, Predictive Modelling for Loan Approval, aims to address these challenges by leveraging data-driven techniques to provide a more efficient, transparent, and fair loan assessment system. By analyzing key financial and demographic factors, the model predicts loan approval outcomes with high accuracy, enabling faster decision-making for both lenders and borrowers.

The primary objective of this system is to enhance accessibility to financial services by minimizing human bias and improving risk assessment. Borrowers receive real-time insights into their loan eligibility, allowing them to make informed financial decisions, while lenders benefit from an automated evaluation process that reduces operational costs and enhances credit risk management. The integration of predictive analytics ensures that the approval process remains objective, reducing unnecessary rejections and ensuring fairer access to credit opportunities. Additionally, the system's user-friendly interface facilitates seamless loan applications and provides clear, data-backed justifications for each decision.

Beyond its immediate practical applications, this project contributes to the broader goal of financial inclusion. By leveraging predictive modelling, the system can be adapted to different types of loans, catering to diverse borrower profiles and evolving financial trends. Its scalable nature allows for continuous improvements through machine learning advancements, ensuring higher accuracy and adaptability. Ultimately, this project represents a significant step toward a more reliable and equitable credit assessment framework, fostering a data-driven financial ecosystem that benefits both individuals and institutions.

Contents

List of Figures

Chapter 1

Introduction

In today's fast-paced financial landscape, access to credit is vital for economic growth and personal stability. Traditional loan approval processes often involve manual assessments, extensive paperwork, and rigid criteria, leading to delays and potential bias. As a result, many deserving applicants face rejections despite strong financial profiles. To overcome these challenges, Predictive Modelling for Loan Approval introduces a data-driven approach that enhances efficiency, accuracy, and fairness. By leveraging predictive analytics, this system offers an objective evaluation of a borrower's eligibility, streamlining the approval process for both lenders and applicants.

This project analyzes key financial indicators—such as income, credit history, and employment details—to predict loan approval likelihood. By replacing traditional decision-making with data-driven modeling, it minimizes uncertainty and ensures that creditworthiness is assessed objectively. This approach accelerates loan processing times, strengthens risk management for lenders, and reduces default rates. Borrowers, in turn, receive clearer insights into their eligibility, enabling informed financial decisions and improving financial inclusion.

As digital financial services continue to grow, integrating predictive modeling into loan approval systems is becoming increasingly essential. This project bridges the gap between financial institutions and borrowers by creating a seamless, transparent, and efficient application process. By adopting intelligent lending models, lenders can mitigate risks more effectively, while borrowers benefit from faster, fairer, and more accessible financial opportunities.

Chapter 2

Problem Definition and Motivation

In today's financial landscape, accessing loans is a critical factor for personal growth and economic stability. However, the traditional loan approval process remains slow, subjective, and inefficient. Many applicants are often denied loans due to rigid criteria, a lack of proper evaluation tools, or slow, manual processing. This leads to delays, inaccuracies, and biases in approving loan applications, especially for individuals with limited credit history or those from marginalized communities. These challenges create barriers for individuals who are deserving of loans but face obstacles due to an outdated, inefficient process.

The motivation behind this project is to provide a more efficient, data-driven approach to loan approvals. By leveraging predictive modeling and machine learning, the goal is to automate the loan approval process, ensuring faster and more accurate decisions based on objective financial data. This system will reduce human bias, streamline the approval process, and help lenders make better-informed decisions, while offering borrowers quicker and clearer feedback regarding their loan eligibility.

2.1 Problem Statement

Despite the increasing demand for loans and financial products, the traditional loan approval process remains inefficient, opaque, and heavily reliant on human judgment. The process is often fraught with delays, subjectivity, and manual errors, leading to lengthy processing times and inconsistencies in decision-making. Applicants may face rejections due to non-transparent criteria or subjective evaluations, even if they have the financial means to repay the loan. This lack of a standardized, objective system for assessing creditworthiness often results in missed

opportunities, where deserving applicants are unfairly denied loans, and financial institutions risk losing out on potential customers.

Moreover, traditional loan approval processes do not leverage the full potential of modern data analytics. Factors like income, credit history, employment status, and other financial indicators are assessed in a fragmented, manual manner, often leading to misjudgments. This limits the ability of lenders to evaluate applicants in a holistic and accurate way, affecting their decision-making process. Additionally, human biases often play a significant role in loan approvals, further exacerbating the problem by unfairly impacting applicants from certain demographic or socio-economic backgrounds.

The project's main aim is to tackle these issues by creating a predictive loan approval system that uses machine learning models to assess the likelihood of a borrower's ability to repay a loan. By leveraging a range of financial data points and objective criteria, the system will ensure a more efficient, fair, and transparent loan approval process. The integration of predictive modeling will enable faster decision-making, reduce the risk of defaults, and promote more equitable access to credit for individuals who may otherwise be overlooked in traditional systems. This approach not only benefits financial institutions but also creates a more inclusive financial environment, enabling borrowers from diverse backgrounds to access the financial support they need.

Chapter 3

System Analysis

3.1 Existing System

The existing system for loan approval processes primarily relies on manual assessments and traditional credit scoring methods. In many financial institutions, the process begins with applicants submitting physical or digital documentation, including proof of income, credit history, employment details, and other financial documents. These documents are reviewed by loan officers or credit analysts who assess an applicant's ability to repay the loan based on predefined eligibility criteria. The decision-making process often involves subjective human judgment, where factors like personal biases and interpretation of financial data can influence the outcome. While automated credit scoring systems exist, they typically rely on static, historical data and follow rigid rules without taking into account the broader context or evolving financial situations of applicants.

3.1.1 Limitation of Existing System

- Manual Processes and Subjectivity
- Lack of Personalization
- Limited Data Utilization
- Inefficiency and Delays
- Risk of Bias and Discrimination

- Limited Accessibility

3.2 Proposed System

The proposed system for loan approval aims to leverage predictive modeling and machine learning algorithms to automate and enhance the loan assessment process. By utilizing a broad range of data, including income, credit history, and non-traditional factors like employment status or education, the system provides a more comprehensive evaluation of applicants. The model learns from historical loan data and continuously improves, making it capable of assessing the creditworthiness of applicants with greater accuracy and fairness. The system's integration with digital platforms ensures that the loan approval process is faster, more efficient, and available to a larger audience, including underserved and unbanked individuals.

3.2.1 Advantages of Proposed System

- Increased Accuracy: By using predictive modeling, the system can analyze multiple factors beyond traditional credit scores, resulting in more accurate loan approval decisions.
- Faster Processing: The automation of loan approval reduces manual intervention, speeding up the decision-making process and providing quicker feedback to applicants.
- Greater Fairness: By removing human bias and incorporating a wider range of data, the system ensures fairer evaluations, promoting greater financial inclusion for individuals from diverse backgrounds.
- Improved Accessibility: The system's digital platform makes the loan approval process accessible to a broader audience, particularly underserved or unbanked populations who may otherwise struggle with traditional processes.
- Continuous Learning: The predictive model improves over time by learning from past data, allowing the system to adapt to changing financial conditions and better predict loan defaults or approvals.

Chapter 4

Requirement Analysis

4.1 Purpose

The purpose of this project is to create an efficient, accurate, and fair loan approval system using predictive modeling and machine learning algorithms. By automating the loan assessment process, it aims to reduce manual intervention, enhance decision-making accuracy, and promote financial inclusion. The system ensures that applicants from diverse backgrounds receive fair consideration based on comprehensive data points.

4.2 Objective

- Develop a Predictive Model for Loan Approval: Build a machine learning model capable of analyzing multiple factors such as income, credit history, employment status, and other personal data to predict the likelihood of loan approval. The goal is to create a robust model that can accurately assess applicants' creditworthiness using a variety of relevant indicators.

- Automate the Loan Approval Process: Design an automated system that processes loan applications without the need for manual intervention. By automating the approval process, the system will significantly reduce processing times, enabling quicker decisions for applicants and increasing the efficiency of financial institutions.

- Enhance Accuracy and Fairness in Loan Assessment: Ensure that the loan approval process is objective and free from bias. The system will use data-driven insights to assess each ap-

plicant's eligibility, helping to eliminate human errors and biases that can affect traditional loan assessments, thus promoting fairness in decision-making.

- Promote Financial Inclusion: Create a system that enables underserved and unbanked individuals to have access to loans. By using non-traditional data and alternative methods of assessing creditworthiness, the system will offer financial opportunities to a broader population, ensuring that more individuals can benefit from the credit system.
- Integrate the System into Digital Platforms: Design and deploy the loan approval system in a way that it can be easily accessed via digital platforms, such as mobile apps or web portals. This integration will make the loan approval process more accessible to a larger audience, ensuring users can conveniently apply for loans from anywhere, at any time.

4.2.1 Product Functions

- Loan Application Submission
- Profile Management
- Predictive Loan Approval
- Loan Approval Status Tracking
- Dashboard of our activities

4.2.2 Hardware Requirements

- Processor : 2.0 GHz or faster processor
- Storage : 1 GB or more.
- Memory : 4 GB RAM

4.2.3 Software Requirements

- Operating System : Android
- Platform : Flutter

- IDE : Visual Studio Code
- Language : Dart
- Database : Firebase

4.3 Functional Requirements

The functional requirements includes all the activities or processes that should be achieved by the proposed system. It includes

- **Flutter :** Flutter is an open-source mobile app development framework created by Google that allows developers to build high-performance, cross-platform mobile applications for both Android and iOS platforms. Flutter uses a reactive programming model, which enables developers to build apps with a single code base that runs seamlessly on multiple platforms. This means that developers can write code once and deploy it across different platforms without compromising the quality of the app's user interface and experience.
- **Firebase :** Firebase is a scalable backend platform built on Google's infrastructure, offering real-time data synchronization, secure storage, and seamless Flutter integration. With Firestore and Realtime Database, it ensures efficient data management, offline support, and dynamic user experiences while maintaining security and reliability.
- **Dart:** Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.

4.4 Non Functional Requirements

4.4.1 Performance Requirements

- Accuracy : Accuracy in functioning and the nature of user-friendly should be maintained by the system.
- Speed : The system must be capable of offering speed.

- Low cost: This system is very cheap to implement and is also user-friendly.
- Less Time consuming: It uses very less time comparing to the existing system .
- User Friendly: This proposed system is highly user friendly they enables to create a good environment.

4.4.2 Quality Requirements

- Scalability : The software will meet all of the functional requirements.
- Maintainability : The system should be maintainable. It should keep backups to atone for system failures, and should log its activities periodically.
- Reliability : The acceptable threshold for down-time should be large as possible. i.e. mean time between failures should be large as possible. And if the system is broken, time required to get the system backup again should be minimum.
- Availability: This system is easily available as the core equipments in building the software is easily obtained.
- High- Functionality: This system is highly functional in all environment since, They are highly adaptable.

Chapter 5

Design And Implementation

5.1 Overall Design

The overall design of the loan approval system is modular, comprising key components: a user-friendly interface, backend services, machine learning models, and a database. The UI allows applicants to easily submit and track their loan applications. The backend handles data processing and integrates predictive analytics to assess loan approval. The machine learning models continuously improve based on historical data, while the database securely stores user and loan application information. This integrated design ensures a streamlined, efficient, and fair loan approval process.

5.2 System Design

The system design for the loan approval prediction platform follows a modular architecture, consisting of a user-friendly frontend built with Flutter, a robust backend powered by FastAPI, and a machine learning model for loan approval predictions. The frontend captures essential loan-related information from applicants, providing a seamless interface across multiple devices. The backend processes user data, interfaces with the pre-trained machine learning model, and securely handles data transmission. The model, built using algorithms like XGBoost, analyzes historical loan data to make accurate predictions on loan approvals. A secure relational database stores all user data and loan records, ensuring smooth data management and tracking. This design ensures scalability, security, and efficient processing, offering a reliable and intuitive loan approval system.

5.3 System Architecture

The system architecture for the loan approval prediction platform follows a client-server model, consisting of a mobile client, backend server, and machine learning model. The mobile client, developed using Flutter, serves as the user interface where loan applicants input their personal and financial details. This data is sent to the backend server, built with FastAPI, which processes the inputs and interacts with the pre-trained machine learning model to assess the loan eligibility. The model, stored and deployed on the server, utilizes historical loan data to predict the likelihood of approval. A secure relational database is used for storing user profiles, loan applications, and prediction results, ensuring data integrity and privacy. The architecture allows for efficient data flow, fast processing, and a seamless user experience, while also ensuring scalability and security for future growth.

5.4 Data Flow Diagram

DFD is one of the graphical representation techniques used in a project to show the flow of the data through a project. DFD helps us to obtain an idea about the input, output, and process involved. The things absent in a DFD are control flow, decision rules, and loops. It can be described as a representation of functions, processes that capture, manipulate, store, and distribute data between a system and the surrounding and between the components of the system. The visual representation helps for good communication.

It shows the journey of the data and how will it be stored in the last. It does not provide details about the process timings or if the process shall have a parallel or sequential operation.

In level 0 the basic data flow of the application is showcased. It does not show the flow of data much deeper. It will be evaluated in the higher levels of Data Flow Diagram.

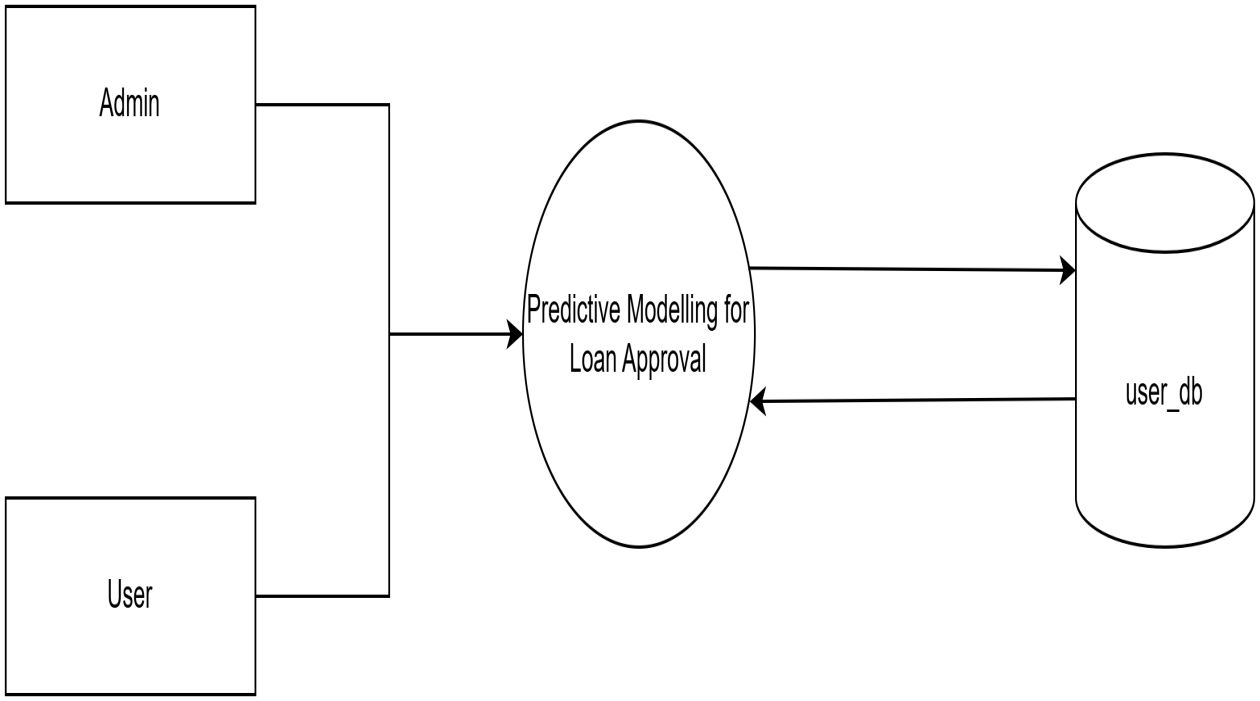

Figure 5.1: Level 0 DFD

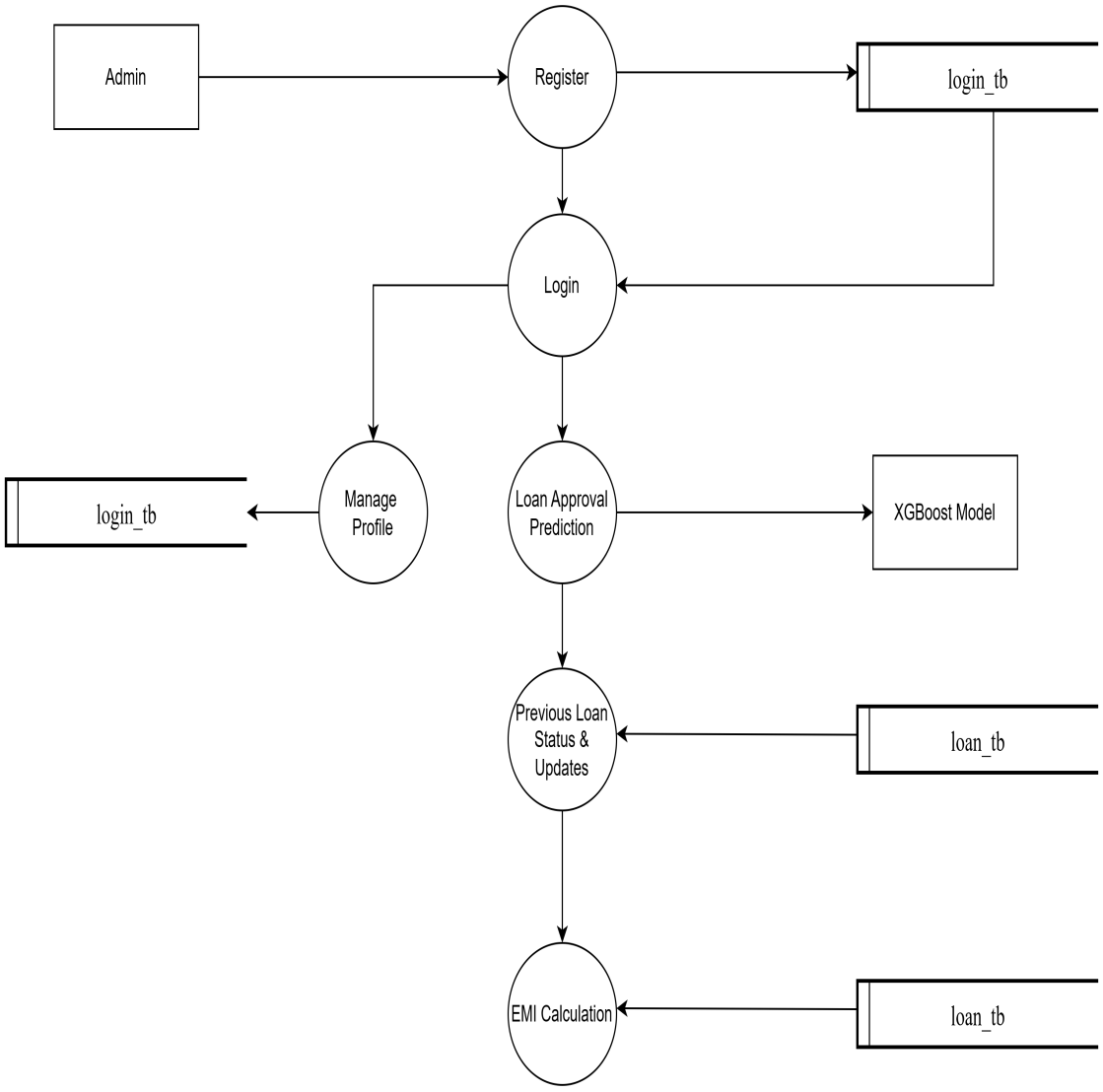

Figure 5.2: Level 1 DFD

5.5 Screenshots of User Interface

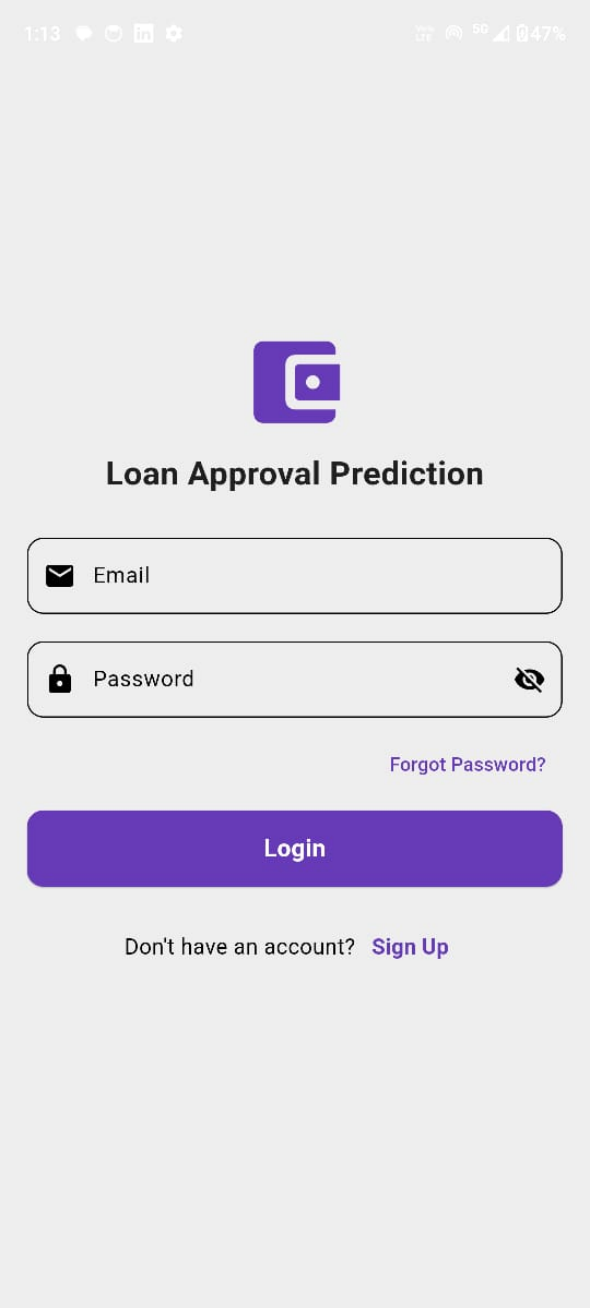

Figure 5.3: Login Page

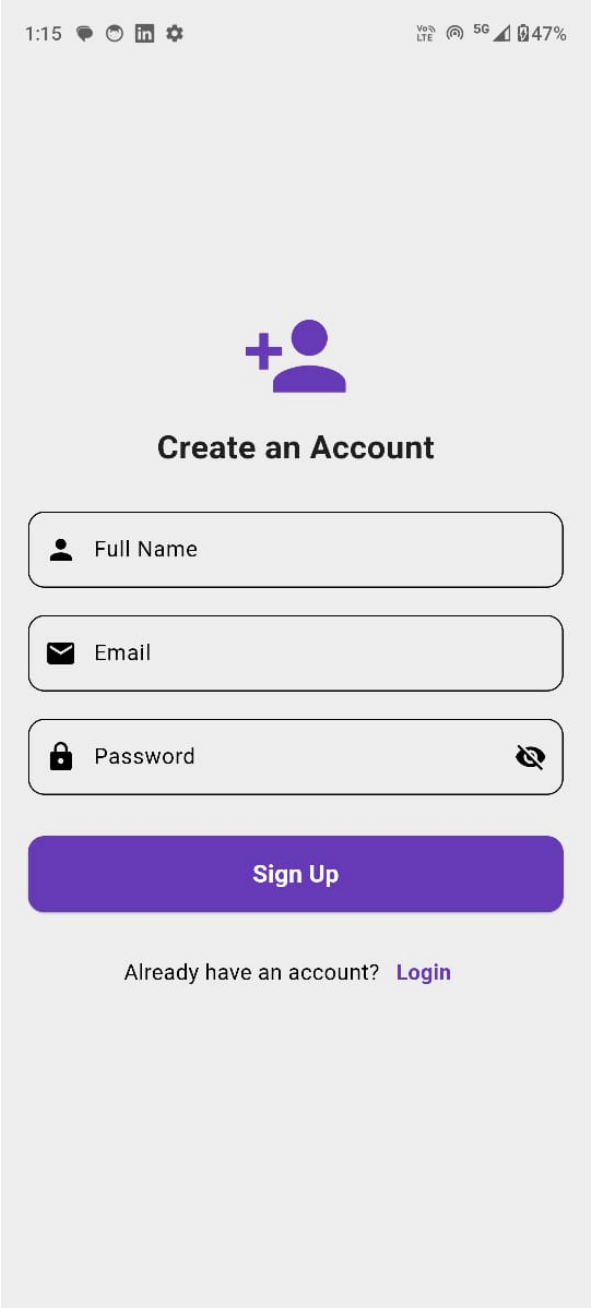

Figure 5.4: Signup Page

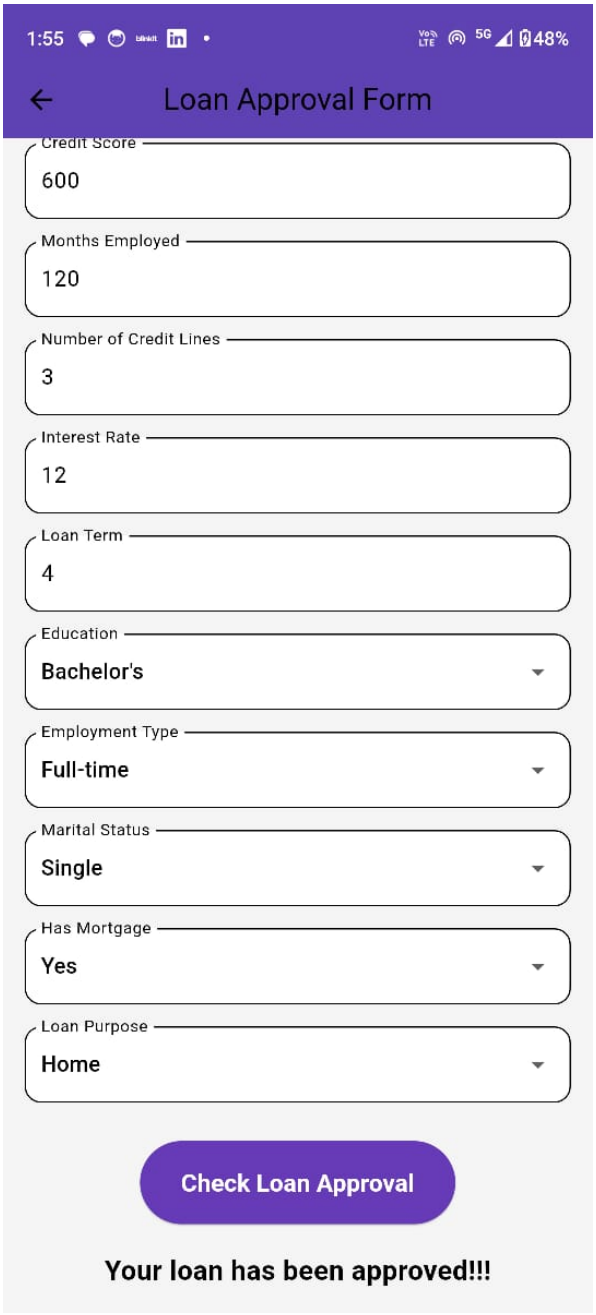

Figure 5.5: Loan Approval Form

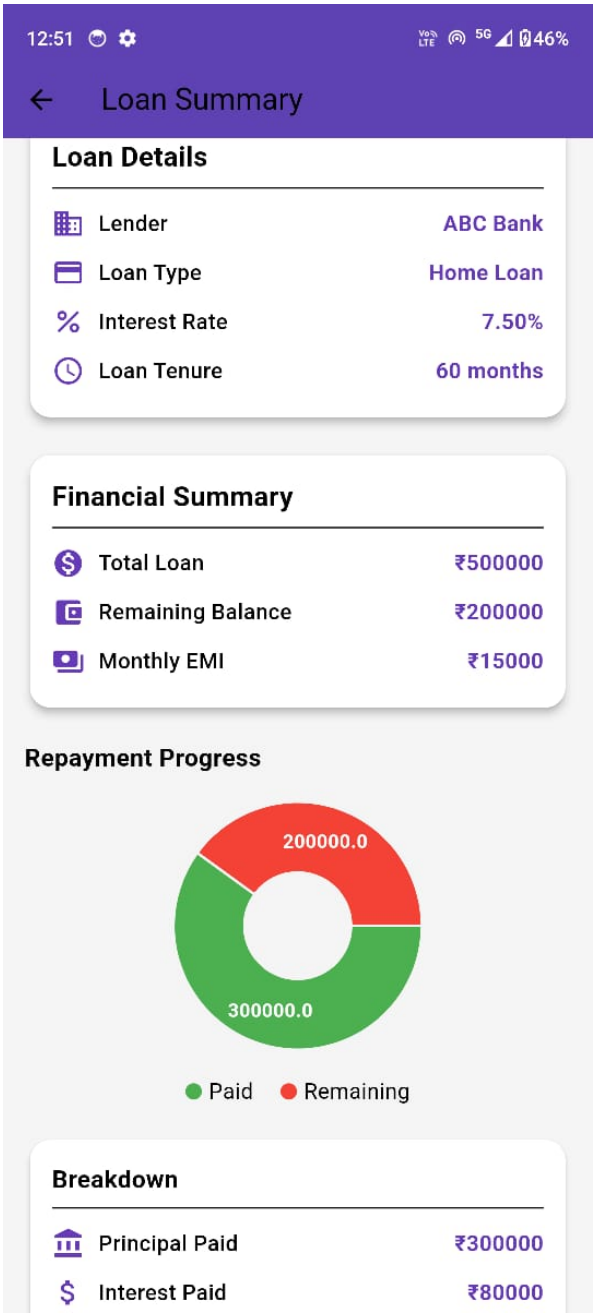

Figure 5.6: Loan Summary

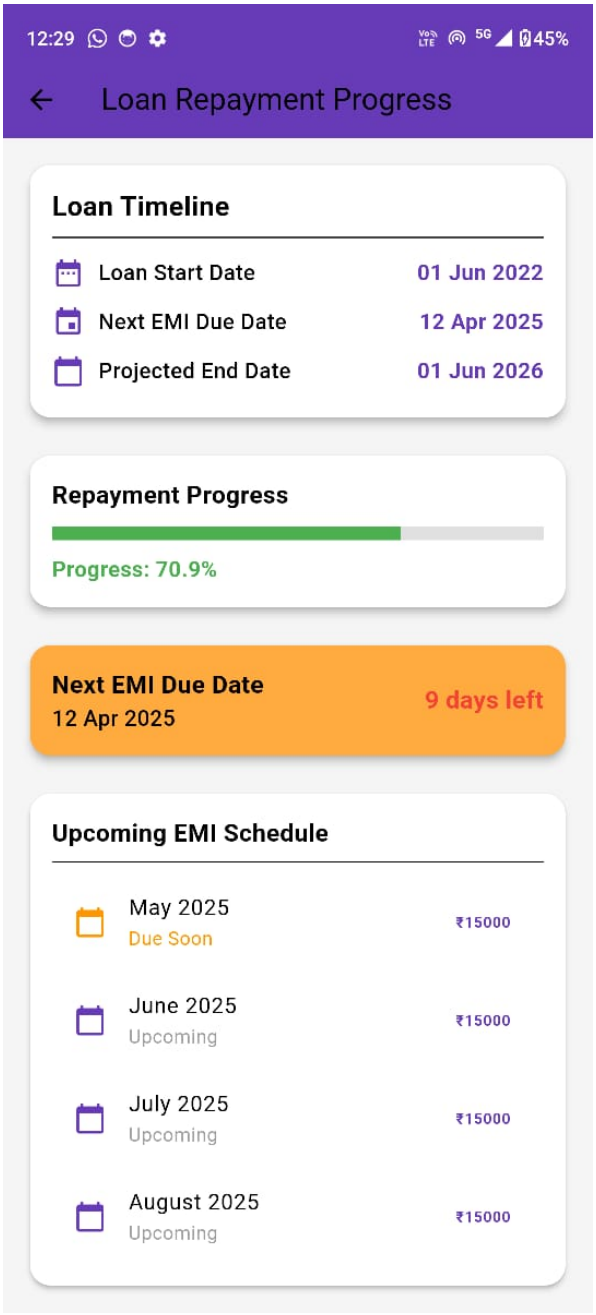

Figure 5.7: Loan Repayment Progress

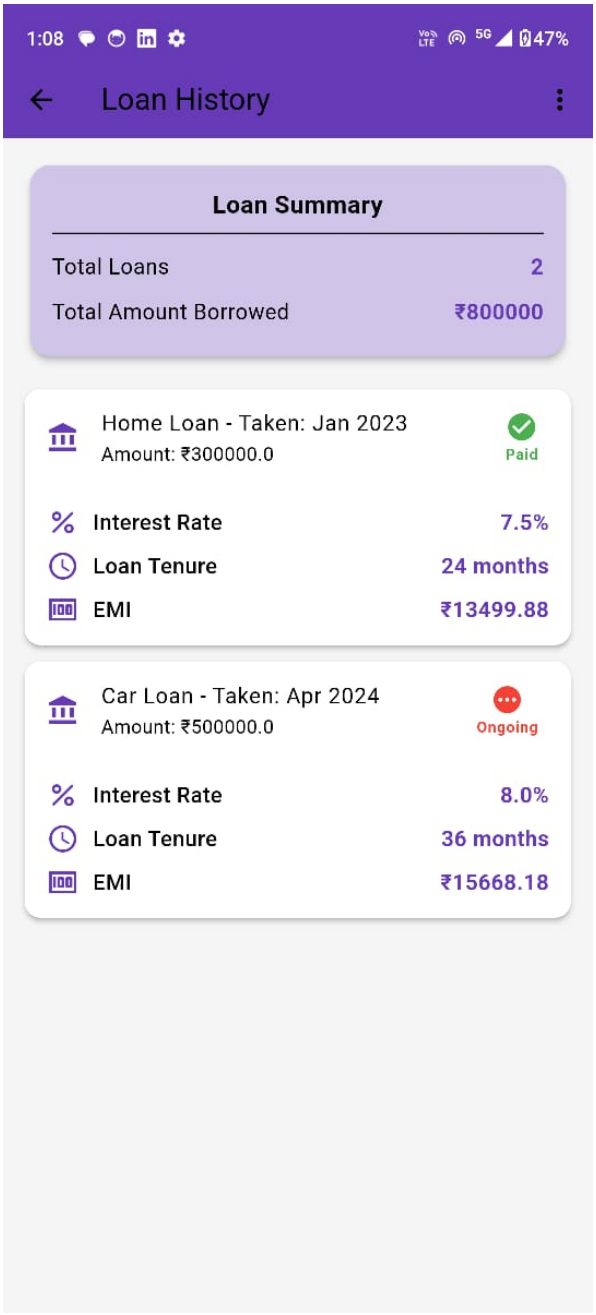

Figure 5.8: Loan History

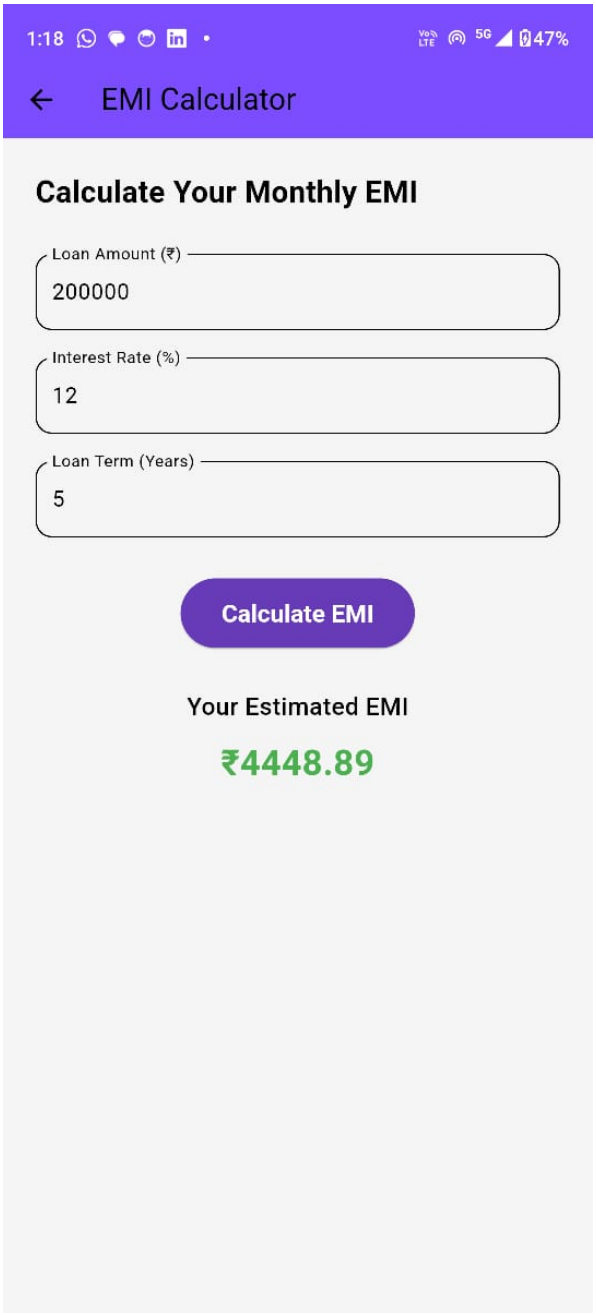

Figure 5.9: EMI Calculator

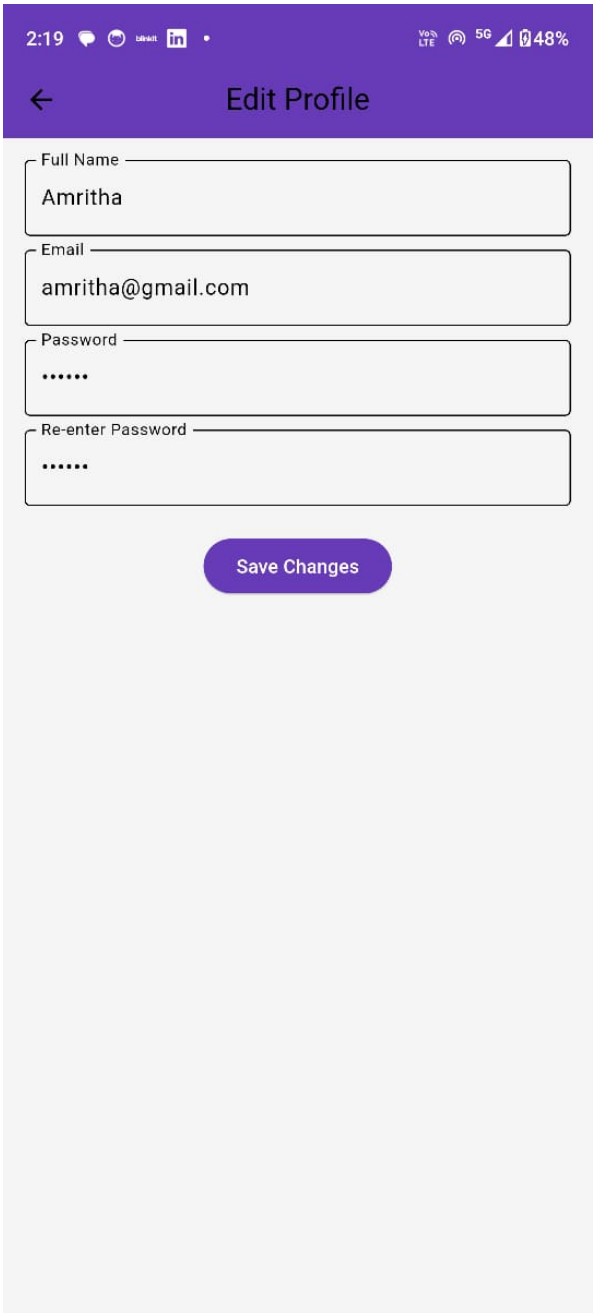

Figure 5.10: Edit Profile

Chapter 6

Coding

```
import 'package:flutter/material.dart';
import 'login_page.dart';
import 'loan_summary.dart';
import 'loan_repayment.dart';
import 'emi_calculator.dart';
import 'loan_history.dart';
import 'loan_page.dart';
import 'profile.dart';

void main() {
  runApp(const LoanManagementApp());
}

class LoanManagementApp extends StatelessWidget {
  const LoanManagementApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Loan Management System',
      theme: ThemeData(
        colorScheme:
            ColorScheme.light(primary: Colors.deepPurple, secondary: Colors.grey),
        scaffoldBackgroundColor: Colors.grey[100],
      ),
      home: const LoginPage(),
    );
  }
}
```

Figure 6.1: main.dart

```
class LoginPage extends StatelessWidget {
  const LoginPage({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Loan Management System",
            style: TextStyle(fontSize: 22, fontWeight: FontWeight.bold)),
        centerTitle: true,
        backgroundColor: Colors.deepPurple,
      ),
      body: Padding(
        padding: const EdgeInsets.all(20.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            _buildNavButton(context, "Loan Summary", const LoanSummaryPage()),
            const SizedBox(height: 20),
            _buildNavButton(context, "Repayment Progress", const LoanRepaymentPage()),
            const SizedBox(height: 20),
            _buildNavButton(context, "EMI Calculator", const EmiCalculatorPage()),
            const SizedBox(height: 20),
            _buildNavButton(context, "Loan History", const LoanHistoryPage()),
            const SizedBox(height: 20),
            _buildNavButton(context, "Loan Page", const LoanPage()),
            const SizedBox(height: 20),
            _buildNavButton(context, "Profile Management", const ProfileManagementPage()),
          ],
        ),
      ),
    );
  }
```

Figure 6.2: main.dart

```
Widget _buildNavButton(BuildContext context, String text, Widget page) {
  return ElevatedButton(
    style: ElevatedButton.styleFrom(
      padding: const EdgeInsets.symmetric(vertical: 20, horizontal: 40),
      textStyle: const TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
      backgroundColor: Colors.deepPurple,
      foregroundColor: Colors.white,
      shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(12)),
    ),
    onPressed: () => Navigator.push(
        context, MaterialPageRoute(builder: (context) => page)),
    child: Text(text),
  );
 }
}
```

Figure 6.3: main.dart code

Chapter 7

Testing and Implementation

7.1 Testing and various types of testing used.

Error detection is a process of testing. Testing is extremely important for quality control and for guaranteeing the dependability of software. The outcomes of testing are also applied later in maintenance. The goal of testing is frequently to establish that a software is functional by proving that it is error-free. The primary goal of the testing step is to find any potential faults in the program. As a result, one should begin testing with the intention of demonstrating that a program does not work rather than the opposite. The act of testing involves running a software with the goal of identifying errors.

7.1.1 Unit Testing

Unit testing concentrates verification work on the module, which is the smallest piece of software. Testing is done to find problems inside the module's perimeter using the design solution and the process standards. Prior to the start of the integration testing, all modules must pass the unit test. Each service in this project can be considered a module. Each module has been tested by receiving various kinds of input. When creating the component as well as completing the development to ensure that every module functions flawlessly. When the user provides input, it is verified. The modules and processes that are put together and integrated to generate a particular function are known as software components in a system. To find faults, testing procedure is first performed on modules separately from one another. Errors can be found thanks to this. By using these methods, the faults brought on by module interaction were initially prevented.

7.1.2 Integration Testing

The stage of software testing where separate software modules are merged and tested as a unit is known as integration testing (sometimes known as testing, abbreviated IT). Integration testing is done to determine whether a system or component complies with a list of functional requirements. It takes place between unit testing and validation testing. The goal of integration testing is to produce an integrated system that is ready for system testing by taking as input modules that have undergone unit testing, grouping them into bigger aggregates, applying the tests outlined in an integration testing phase to those aggregates.

Types of Test Cases

Integration test case : Here, we run test cases that merely describe connectivity between modules and integration across applications, as well as how an application transitions from a parent node to a child node and vice versa.

Functional Test case : Here, we run test cases that describe the application's functionality and describe the desired results. According to the SRS, we provide input and anticipate some output. Here, we thoroughly examine each module by looking over each and every window, textboxes, button, and so forth.

Non-Functional Test case : Examples of user-friendliness testing for font, picture, colour, and how Here, things like ease of use, performance, and security are covered.

User Acceptance test case : These test cases, which are also known as end-to-end business scenario test cases, are critical and extremely important to client side personnel since they discuss the business and strategy of the application to complete a certain client task.

7.1.3 System Testing

The entire application system is tested at this place. The requirements document serves as the process's reference, and its compliance with the requirements will be determined. Here, the complete piece of software has been examined in relation to the project's requirements, and it is determined whether or not all of the requirements have been met.

Alpha Test : After the initial round of defects has been fixed, the newly designed product undergoes its first test with actual users. To make sure the developer accurately captured the client's vision for custom software, the client may be invited to an alpha test.

Beta Test : A test of a new or updated software programme that is carried out by customers at their facilities while it is being used normally. Alpha testing is followed by beta testing. Customers of packaged software vendors are frequently given the chance to beta test.

Acceptance Testing : The customer of the application tests it to see if it was created in accordance with the specifications he or she provided. It is carried out either at the customer site or within the organization's development.

Chapter 8

System Implementation

The deployment of the new system to a selected group of users and the positioning of on going support and functionality of the facility within the performing organisation can be used to summarise the goal of system implementation (the transition). At a more granular level, deploying the system entails carrying out all procedures required to inform Consumers on how to use the new system, putting the new advanced system into production, ensuring that all data needed at the beginning of the procedure is accessible and precise, and validating that business processes that interaction between the system are operating as intended. The ownership of the new system is transferred from the Project to the Performing Organization, and the system sup port responsibilities are changed from system development to system support and maintenance. The fact that all program operations up to this moment have been carried out in safe, secure, and protected settings, where project difficulties that emerge have little to no impact on regular company operations, is a major distinction between Systems Development and all other stages of the lifecycle. However, it's no longer true after the system is live. Any errors made at this point will almost probably have an immediate negative impact on the Performing Organization's operations and/or finances. The Project Team can reduce the risk of these incidents and choose the best backup plans in the event of an issue by carefully planning, carrying out, and managing System Implementation activities.

Chapter 9

Conclusion

In conclusion, this project successfully integrates various functionalities to assist users in managing their loans efficiently. With features like loan application, EMI calculation, repayment tracking, and profile management, the app ensures a seamless and user-friendly experience. The implementation of Firebase for data management, along with Flutter's responsive UI, enhances the application's performance and accessibility. Additionally, the app's structured design allows for scalability, making it adaptable for future enhancements such as AI-driven loan approval predictions, multi-user support, and integration with banking APIs to provide real-time financial insights. Thorough unit testing ensures the reliability of key functionalities, improving the overall user experience. Security measures such as encrypted password storage and secure authentication protocols can be incorporated to enhance data protection. Moreover, integrating machine learning models can further optimize loan approvals by analyzing user data and predicting repayment probabilities. Overall, this project serves as a practical and scalable solution for loan management, benefiting both borrowers and lenders while setting a foundation for further financial technology advancements.

Chapter 10

Future Scope

- Expansion to a Full-Fledged Financial Management Platform: Beyond loan management, the system can be expanded to include features like savings tracking, investment advice, and budgeting tools. This will provide users with a holistic financial management experience, making the app more valuable and increasing user engagement.
- Real-Time Banking and Payment Gateway Integration: By integrating with banking APIs, users can automate loan repayments, check their real-time account balance, and receive payment reminders. Secure payment gateways can also enable borrowers to make EMI payments directly through the app, improving financial discipline.
- Enhanced Security and Compliance with Financial Regulations: Implementing advanced security measures such as biometric authentication, end-to-end encryption, and fraud detection algorithms will ensure user data protection. Additionally, compliance with financial regulations such as RBI guidelines and GDPR will make the platform more trustworthy and suitable for large-scale adoption.

Bibliography

[1] Flutter Documentation

[2] Firebase Documentation

[3] XGBoost Algorithm for Loan Default Prediction – Chen, T., Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[4] Lyn Thomas, David Edelman, and Jonathan Crook. Credit Scoring and its Applications. 2002.

[5] E. I. Altman and A. Saunders. "Credit risk measurement: Developments over the last 20 years". In: Journal of Banking Finance 21.11-12 (1994)

[6] Si Tse Shi et al. "Machine learning-driven credit risk: a systemic review". In: Neural Computing and Applications 34 (2022)