ABSTRACT

The FinEdge Application is an advanced software solution designed to improve financial management for small and medium-sized enterprises (SMEs). Utilizing cutting-edge technologies such as React.js for the frontend, Node.js for the backend, and MySQL Workbench for data storage, this platform offers a comprehensive and secure environment for handling digital banking, investment management, and loan processing.

This application consists of two primary panels: a user panel and an admin panel. The admin panel streamlines the management of user applications, allowing administrators to approve or reject applications and view user information. The user panel enables users to log in, access their dashboard, and engage in various financial activities. Existing users authenticate by entering their account number, username, and password, while any new users can register by providing personal details, setting up credentials, and receiving a generated account number for future use.

The user dashboard offers a range of functionalities, including transferring money to different accounts, viewing transaction history, paying utility bills, and applying for credit or debit cards and loans. For credit card and loan applications, users must submit the required information and await processing, with administrators responsible for reviewing and making decisions on these applications.

By automating and integrating various financial tasks, the FinEdge Application provides crucial insights and tools for managing an enterprise, ensuring compliance, tracking client data, and enhancing performance while adhering to regulatory standards. This application marks a significant improvement in the efficient and secure management of financial operations for SMEs.

CHAPTER1

INTRODUCTION

1.1 PROJECT DESCRIPTION

The FinEdge Application is a standard platform designed to enhance financial management for both small and also for medium enterprises (SMEs). Built with React.js for the frontend, Node.js for the backend, and MySQL Workbench for data storage, the application offers a secure, user-friendly environment to streamline various financial operations such as digital banking, investment management, and loan processing. It features two main interfaces: a user panel for accessing personal financial tools and an admin panel for managing user applications, reviewing, and approving or rejecting submissions.

Users can log in using their account number, username, and password, or register as new users to access the platform's functionalities. The dashboard allows users to transfer money, view transaction history, pay utility bills, and apply for credit or debit cards and loans. Admins oversee these applications, ensuring efficient management and compliance with regulatory standards. By integrating these functions, the FinEdge Application empowers SMEs to manage all finances effectively, improving operational efficiency and financial health.

1.2 COMPANY PROFILE

AgileMinds IT Solutions is a leading force in the IT training, services, and solutions industry. Backed by professionals with engineering and IIM backgrounds, the team brings over two decades of industry experience. The company's mission is to manage corporate needs and institutional capabilities through innovative training initiatives. With expertise in Human Resources, Technology, Sales, and Marketing, AgileMinds delivers customized solutions to address the ever-evolving demands of the IT industry.

AgileMinds' training programs are led by experienced instructors who provide deep industry insights and practical knowledge. These programs are designed to offer a comprehensive learning experience, enhanced by hands-on mentorship. Beyond training, AgileMinds also undertakes client projects, delivering high-quality, tailored solutions that consistently exceed expectations. With a diverse team, robust training programs, and a proven track record in client success, AgileMinds IT Solutions stands as a leader in the IT sector.

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING AND PROPOSED SYSTEM

Existing System

Current financial management for SMEs often involves fragmented systems and manual processes, leading to inefficiencies and data silos. Traditional systems lack integration, making it difficult for SMEs to have a comprehensive view of their financial health. This fragmentation results in high administrative burdens and a lack of timely, informed decision-making due to the absence of automation.

Proposed System

The FinEdge Application addresses these challenges by providing a unified platform that integrates digital banking, investment management, and loan processing. Utilizing modern technologies like React.js, Node.js, and MySQL, FinEdge ensures robust performance, scalability, and security. It features distinct user and admin panels, enhancing functionality and security with role-based access controls. Automation reduces administrative overhead, while comprehensive financial insights and regulatory compliance help SMEs make better decisions and maintain legal standards. This approach improves efficiency and data management compared to existing systems (IJMSSSR) (International Journal of Research (IJR)).

2.2 FEASIBILITY STUDY

The FinEdge Application was tested to ensure security and reliability, with many SMEs expressing high satisfaction with its performance and user experience.

Technical Feasibility

The application leverages modern technologies like React.js, Node.js, and MySQL, ensuring robust performance and scalability. These technologies were widely used, well-documented, and supported by active communities and also making them ideal for the application's development and future scalability. The architecture's modularity facilitates easy integration with existing financial systems and future enhancements.

Economic Feasibility

From an economic perspective, the FinEdge Application offers substantial cost savings by automating financial processes, thereby reducing the need for extensive administrative tasks and minimizing errors. Initial development costs are offset by long-term operational efficiencies and potential revenue from subscription-based services. The return on investment (ROI) is projected to be favorable due to improved financial management and reduced operational costs for SMEs.

Operational Feasibility

Operationally, the application is designed with a user-friendly interface that minimizes the need for extensive training. The clear separation of user and admin panels ensures efficient management of financial tasks and user applications. Automated features streamline daily operations, reduce manual intervention, and enhance overall productivity.

2.3 TOOLS AND TECHNOLOGIES USED

The FinEdge Application is being developed using VS Code as the primary code editor. The backend of application is built using Node.js for server-side scripting, while the frontend is developed with React.js to create a dynamic and responsive user interface. MySQL Workbench is used for database management and data storage. The application is designed to run on a Node.js server, providing a robust and scalable environment for handling various financial operations.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 INTRODUCTION

3.1.1 PURPOSE

The purpose of this Software Requirements Specification (SRS) document is to systematically capture the requirements of the FinEdge Application. This document aims to clearly define the application's functionalities, ensuring they align with the project goals and user needs. It serves as a comprehensive guide for developers and stakeholders, establishing a foundation for the system's development and subsequent validation. All modifications to the requirements will adhere to a formal change management process to ensure consistency and accuracy throughout the project lifecycle.

3.1.2 DOCUMENT ABBREVATIONS/ DEFINITIONS

- **API**: Application Programming Interface
- **HTTP**: Hypertext Transfer Protocol
- **JS**: JavaScript
- **MySQL**: My Structured Query Language
- **Node.js**: Node JavaScript Runtime
- **React.js**: React JavaScript Library
- **SRS**: Software Requirements Specification
- **SQL**: Structured Query Language
- **UI**: User Interface
- **VS Code**: Visual Studio Code

3.1.3 TARGET AUDIENCE

3.1.3.1 Developer

Developers should use this document to understand the system requirements and make sure all functionalities were implemented as specified. They will refer to this document to make necessary adjustments or updates during the development and testing phases.

3.1.3.2 Admin.

Admin were responsible for overseeing the security of the application and managing user accounts. They have the authority to add or remove users and are crucial for maintaining the integrity and smooth function of the system.

3.1.3.3 User

End user of the FinEdge Application will make use of this system for various financial activities, including managing transactions, viewing financial reports, and applying for credit or loans. They should be familiar with the application's features and requirements to effectively use its functions.

3.1.4 PROJECT SCOPE

The Software Requirements Specifications (SRS) document consolidates all the essential requirements for the FinEdge Application into a comprehensive framework. It provides clear guidelines and specifications for developers to follow, ensuring that the system meets the outlined objectives and user needs. The document is designed to support developers in understanding and implementing the system's functionalities while serving as the foundation for validating the final product. Any modifications to the requirements will follow a formal change management process to maintain the integrity of the project. This document aims to offer a thorough overview of the FinEdge Application's operations, ensuring that all stakeholders grasp the core functionalities and vision of the system effectively.

3.1.5 BENEFITS

The primary aim of this document is to guide the development and testing teams in creating the FinEdge Application effectively. The FinEdge Application offers several advantages, including:

- **Efficiency in Financial Management**: Streamlines financial processes for small and medium enterprises, reducing manual effort and time spent on managing financial activities.
- **Enhanced User Experience**: Provides a user-friendly interface for easy navigation and access to financial services, including transactions, bill payments, and loan applications.
- **Comprehensive Financial Oversight**: Allows users and administrators to manage and track financial operations efficiently, improving overall control and transparency.

3.1.6 REFERENCES

1. **Software Engineering: A Practitioner's Approach, 9th Edition** by Roger S. Pressman and Bruce R. Maxim, McGraw-Hill Education.

3.2 OVERALL DESCRIPTION

3.2.1 IDENTIFICATION OF PRE-EXISTING WORK

The FinEdge Application offers a streamlined, web-based solution for financial management in small and medium enterprises. It is a web-based platform and integrates functions such as transaction management, bill payments, and loan applications, automating processes to reduce manual effort and improve efficiency. By replacing traditional methods with a more cohesive system. This application provides a more efficient and integrated approach compared to existing systems, enhancing overall user experience and operational efficiency.

3.2.2 PERSPECTIVE ON PRODUCT

The FinEdge Application is a web-based platform designed for financial management within small and medium enterprises. It supports users in managing transactions, bill payments, and loan applications efficiently. Users interact with a streamlined interface for those function, which were automated to minimize manual effort and paperwork. Unlike traditional financial management methods, which can be cumbersome and time-consuming, FinEdge provides a direct and integrated approach, significantly speeding up financial processes and enhancing overall efficiency. The application eliminates redundant steps, offering a user-friendly experience for all users and administrators.

3.2.3 PRODUCT ATTRIBUTES

The FinEdge Application features a user-friendly interface accessible from various platforms by both users and administrators. Key attributes include:

- **Login**: Users log in using their credentials for secure access.
- **New User Registration**: Allows new users to register by providing personal information and setting up their account.
- **Add Money**: Users can add money into their account.
- **Check Balance**: Users have the ability to view their current account balance.
- **Money Transfers**: Users can initiate and manage transfers between accounts.
- **Transaction History**: Provides a view of past transactions for users.
- **Pay Bills**: Facilitates payment of utility bills such as electricity, water, and mobile recharge.
- **Credit and Loan Applications:**Users have the option to apply for credit cards and loans, with applications reviewed by administrators.
- **Add Debit Cards**: Users can apply for and manage debit cards.

- **Loan Processing**: Administrators review and process loan applications submitted by users.
- **Logout**: Provides users with the option to securely log out of their accounts.

3.2.4 END USER CHARACTERISTICS

The FinEdge application is designed to be user-friendly, requiring minimal training for both administrators and end-users. The system comprises two primary user roles: Administrators and End-Users.

3.2.4.1 Administrator

- Manages the overall FinEdge system.
- Creates, modifies, and deletes user accounts.
- Oversees user applications for credit cards and loans.
- Generates system reports and analytics.
- Maintains system security and compliance.

3.2.4.2 End-User

- Performs various financial transactions within the application.
- Accesses account information and transaction history.
- Applies for credit cards and loans.
- Manages personal profile and settings.

3.2.5 OPERATING ENVIRONMENT

3.2.5.1 General Constraints

- All required fields must be completed with accurate information.
- Only administrators have access to modify system settings and user management functions.
- Users must adhere to the application's terms of service and privacy policy.
- System performance may be affected by factors such as network connectivity and device capabilities.
- The application is designed for use on desktop and mobile devices with compatible browsers and operating systems.

3.2.6 CONSTRAINTS IN DESIGN AND IMPLEMENTATION

- All required fields must be completed.
- Relevant fields must be filled out accurately.
- Users must enter a strong password and a valid email address.
- User registration must be validated to ensure all necessary information is provided.

3.2.7 ASSUMPTIONS AND DEPENDENCIES

- Users of the application must have appropriate access rights.
- Users should have basic computer literacy.
- Internet access is required to use the application.

3.3 PRODUCT FUNCTIONALITY

3.3.1 USER MODULE

3.3.1.1 Login

Users must log in with their account number, username, and password to move to dashboard.

3.3.1.2 New User Registration.

New users must apply by providing their personal details and setting up their account. If a user already exists, an appropriate message is displayed.

3.3.1.3 Money Transfers

Users can initiate and manage money transfers to different accounts.

3.3.1.4 Transaction History

Users may have option to check past transactions.

3.3.1.5 Pay Bills

Users can also pay their utility bills such as electricity, water, and mobile recharge.

3.3.1.6 Submit applications for credit cards and loans

Users are able to apply for credit cards and loans. Applications will be reviewed by administrators.

3.3.1.7 Add Debit Cards

Users can apply for and manage their debit cards.

3.3.1.8 Add Money

Users can add money to their account.

3.3.1.9 Check Balance

Users can view their account balance.

3.3.1.10 Logout

Users can securely log out of their accounts.

3.3.2 ADMIN MODULE

3.3.2.1 Login

Administrators must log in with their credentials to access the admin panel.

3.3.2.2 Manage Users

Administrators can add, view, and manage users, including setting roles and permissions.

3.3.2.3 Manage Transactions

Administrators can monitor and review transactions.

3.3.2.4 Approve or Reject Applications

Administrators can review and either approve or reject applications for credit cards and loans.

3.3.2.7 Logout

Administrators can securely log out of their accounts.

3.4 EXTERNAL INTERFACE REQUIREMENTS

3.4.1 USER INTERFACES

- Users can access the FinEdge Application via a web browser.
- The homepage contains links to other sections such as account management, transaction history, and bill payments.
- Users can submit transaction requests, view their account balance, and manage their financial activities.
- The login form includes validation to guarantee that only authorized users have access to the system after successful authentication.

3.4.2 HARDWARE INTERFACES

- Any device with a web browser installed, such as computers, tablets, or smartphones.

3.4.3 SOFTWARE INTERFACES

- The application requires support from web browsers such as Chrome and Firefox, Safari, or Edge.

3.4.4 COMMUNICATION INTERFACES

- FinEdge Application needs a web browser, an HTTP-compatible internet connection, and a server for communication.

3.5 OTHER NON-FUNCTIONAL REQUIREMENTS

3.5.1 PERFORMANCE REQUIREMENTS

The possibility of a system crash, which results in data loss, exists. There can be a backup to recover the data to prevent this.

The overall system performance should be quick and error-free, with built-in:

- Error checking and correction facilities.
- Running this program requires the following:
  - A minimum 56 kbps bandwidth internet.
  - We need a browser with IE6 or a higher version to access this page.

3.5.2 SAFETY REQUIREMENTS

A user can login to the system only after logging in successfully. Access is restricted to authorized users only to use the application, preventing access to the system is granted.

3.5.3 SOFTWARE QUALITY ATTRIBUTES

3.5.3.1 Reliability

Verify the product's durability to ensure it can endure any challenges. It should continuously produce accurate findings. The project's capacity to function in various working environments and situations serves as a measure of product reliability.

3.5.3.2 Maintainability

Iterations of the product should be straightforward to maintain. It should be easy to add code to the existing system for development and to upgrade the system for new features and developing technologies. It should be affordable and easy to maintain. Updating the program or fixing faults in the system should be straightforward.

3.5.3.3 Portability

This can be evaluated in terms of the financial difficulties, challenges with technology, and behavioral difficulties associated with porting.

3.5.3.4 Usability

This can be evaluated using the ease of use. The software ought to be easy to use. Learning ought to be easy. It should be simple to navigate. The system must be:

- Easy to operate, prepare the input, and evaluate the result.
- Provide user interface guidelines or standards that align with other frequently used systems.
- FinEdge Application should be easy for new or infrequent users to learn and use.

3.5.3.5 Flexibility

Must be have to change and adapt.It should have the capability to be altered to communicate with various components. Connecting to other widely used third-party components make it simple.

3.6 SPECIFIC REQUIREMENTS

3.6.1 OPERATING ENVIRONMENT

3.6.1.1 Hardware

• **Processor**	:	Minimum Intel Pentium 4 processor
• **RAM**	:	Minimum 1GB
• **Hard Disk**	:	Minimum 10GB

3.6.1.2 Software

• **Language**	:	HTML, CSS, JavaScript, Node.js
• **Frontend**	:	HTML, CSS, JavaScript, React.js
• **Backend**	:	Node.js, Express.js
• **Database**	:	MySQL

3.7 DELIVERY PLAN

Develop and launch a user-friendly financial management application for small and medium enterprises within X months. The application will enable efficient management of transactions, bill payments, and loan processing. Prioritize simplicity and accessibility to ensure ease of use for all users.

CHAPTER 4

SYSTEM DESIGN

4.1.SYSTEM DESIGN

4.1.1 INTRODUCTION

System design specifies the elements of the system, such as its architecture, modules, and data. It must be sometimes referred to as creating a subapplication. System design focuses on selecting the necessary components and ensuring consistency among these modules.

4.1.2 SCOPE

This document is essential for helping end users understand the FinEdge Application in a methodical way. It describes the system's context flow, allowing us to visualize the system's data flow and direction. Additionally, it explains the connections between each module of the software and use-case realization to guide the developer in adopting the best practices during programming.

4.1.3 AUDIENCE

The primary users of the FinEdge Application during the design phase include the developer, testing team, and maintenance team.

4.2 SOFTWARE PRODUCT ARCHITECTURE

4.2.1 ARCHITECTURAL DESIGN

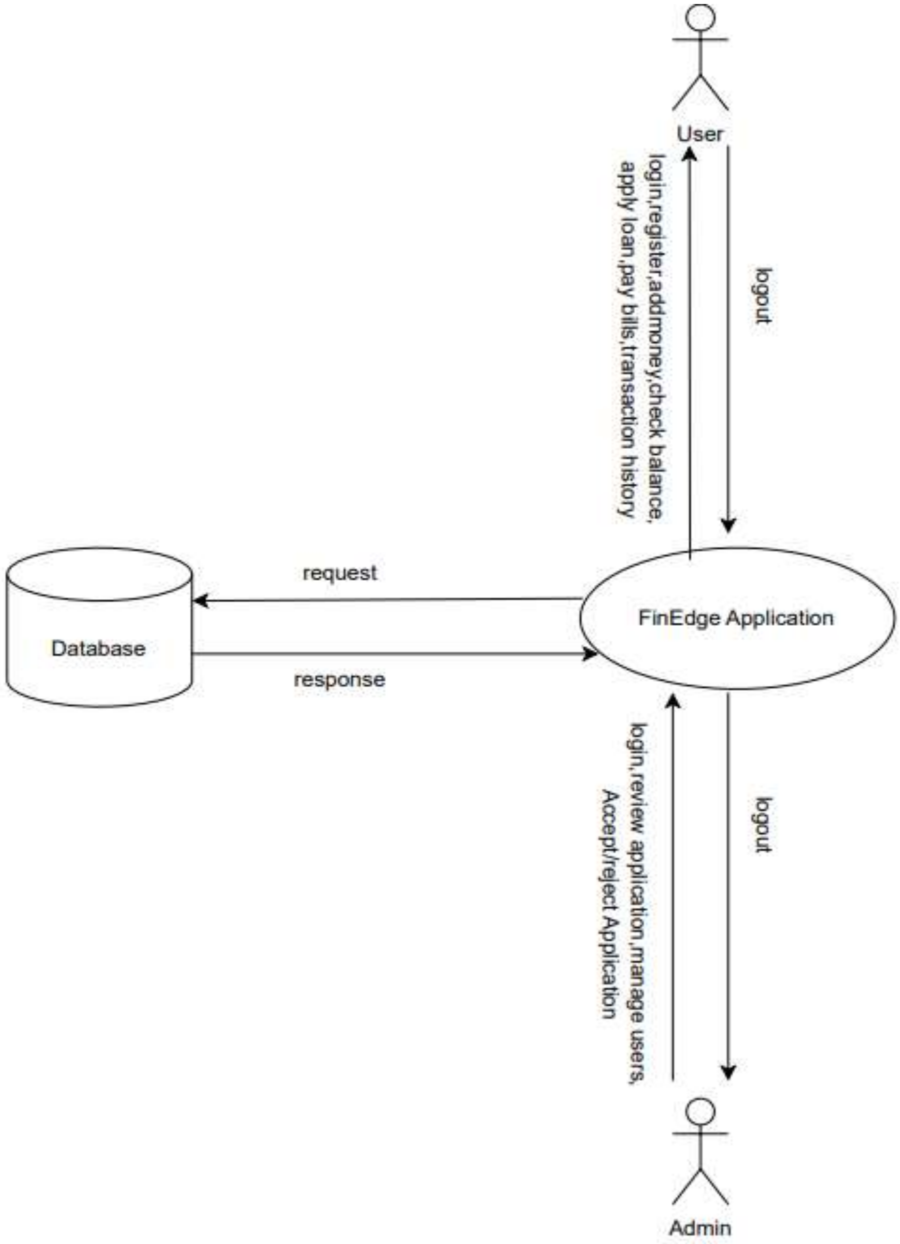

User
login,register,addmoney,check balance, apply loan,pay bills,transaction history
logout
request
Database
FinEdge Application
response
login,review application,manage users, Accept/reject Application
logout
Admin

Figure 4.1:FinEdge Application Architectural Design

4.2.1.1 View Layer

This layer encompasses both the Admin Panel also the User Panel of the FinEdge Application. The Admin Panel allows administrators to manage user applications, including the ability to accept or reject applications and view user details. The User Panel provides functionalities such as login, dashboard access, and various financial activities for the end users.

4.2.1.2 Business Layer

The Business Layer encompasses the core functionality of the FinEdge Application, which includes processing user registrations, managing financial transactions, and handling applications for credit cards and loans. This layer also ensures secure communication and transaction processing between the front-end interface and the backend services.

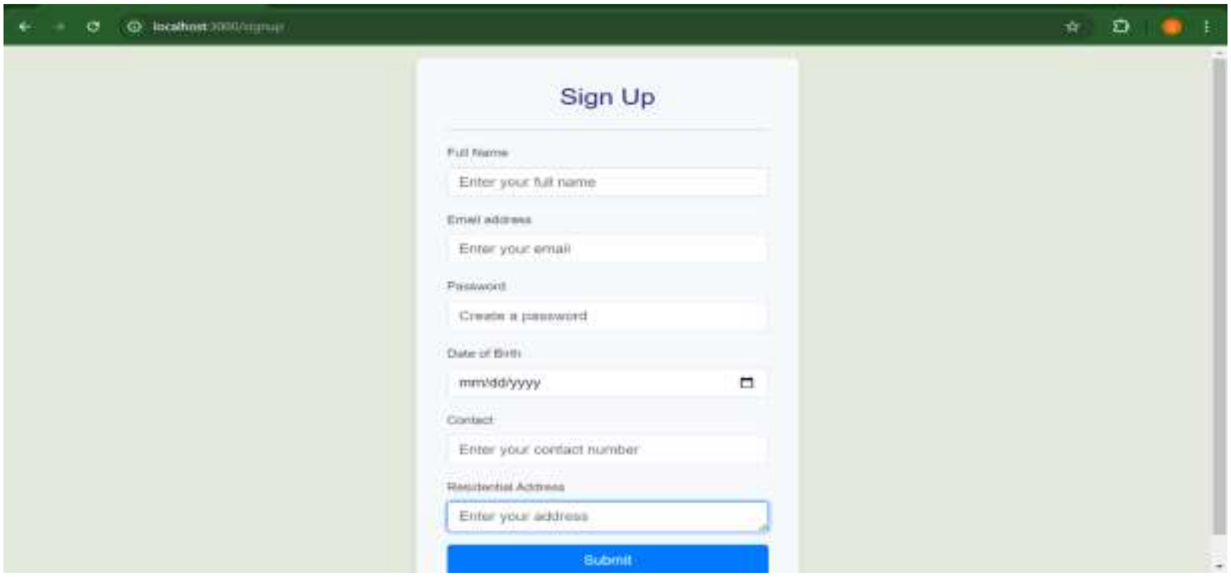
4.2.1.3 Access Layer

The Access Layer interacts with the MySQL Workbench database to store and retrieve user data. It manages user authentication, account information, transaction history, and application status, ensuring data integrity and security throughout the system.

4.3 ARCHITECTURE OF COMPONENTS

4.3.1 INTERFACE DESIGN

The primary objectives of the user interface are to facilitate interaction between the system as well as the user, and also to enable users to quickly understand how to navigate and utilize its features.



localhost:3000/signup

Sign Up

Full Name: Enter your full name

Email address: Enter your email

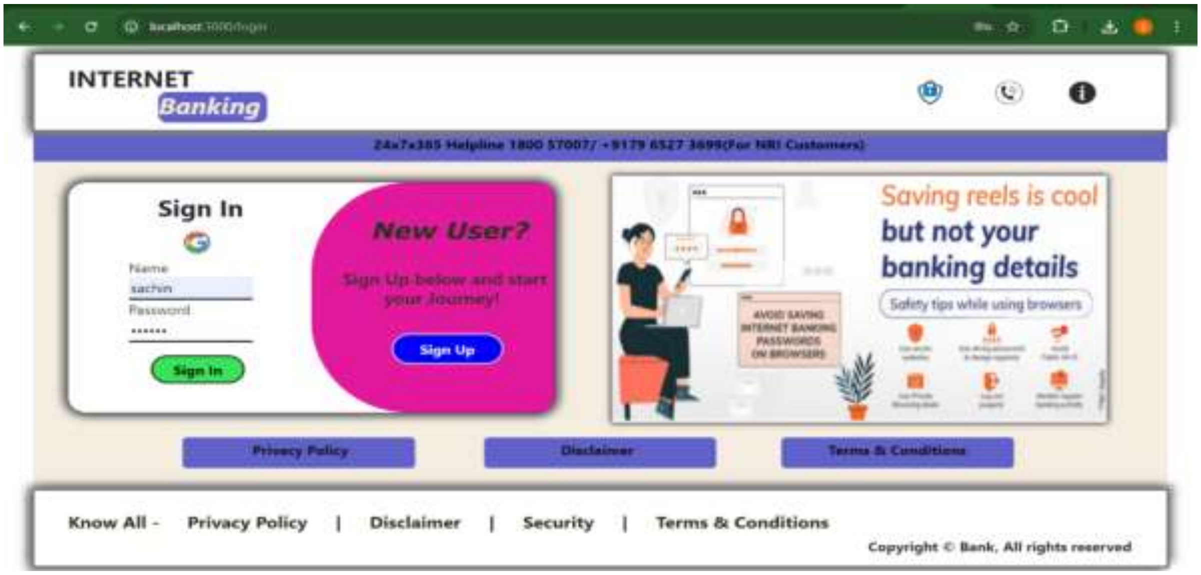
Password: Create a password

Date of Birth: mm/dd/yyyy

Contact: Enter your contact number

Residential Address: Enter your address

Figure 4.2:Sign Up User Interface



localhost:3000/login

INTERNET Banking

24x7x365 Helpline 1800 57007/ +9179 6527 3699(For NRI Customers)

Sign In

Name: sachin

Password: ******

New User?

Sign Up below and start your Journey!

Saving reels is cool but not your banking details

Safety tips while using browsers

AVOID SAVING INTERNET BANKING PASSWORDS ON BROWSERS

Know All - Privacy Policy | Disclaimer | Security | Terms & Conditions

Copyright © Bank, All rights reserved

Figure 4.3 :Login User Interface

4.4. DATA FLOW DIAGRAM

4.4.1 CONTEXT DIAGRAM FOR FLOWFINEDGE APPLICATION

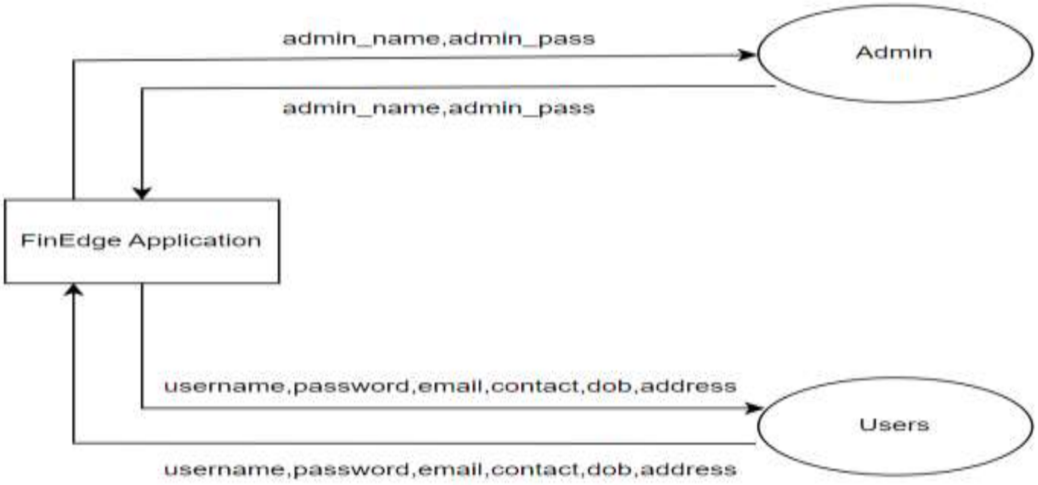

admin_name,admin_pass
Admin
admin_name,admin_pass
FinEdge Application
username,password,email,contact,dob,address
Users
username,password,email,contact,dob,address

Figure 4.4:Cntext Flow Diagram for FinEdge Application

4.4.2 LEVEL 1 DFD USER

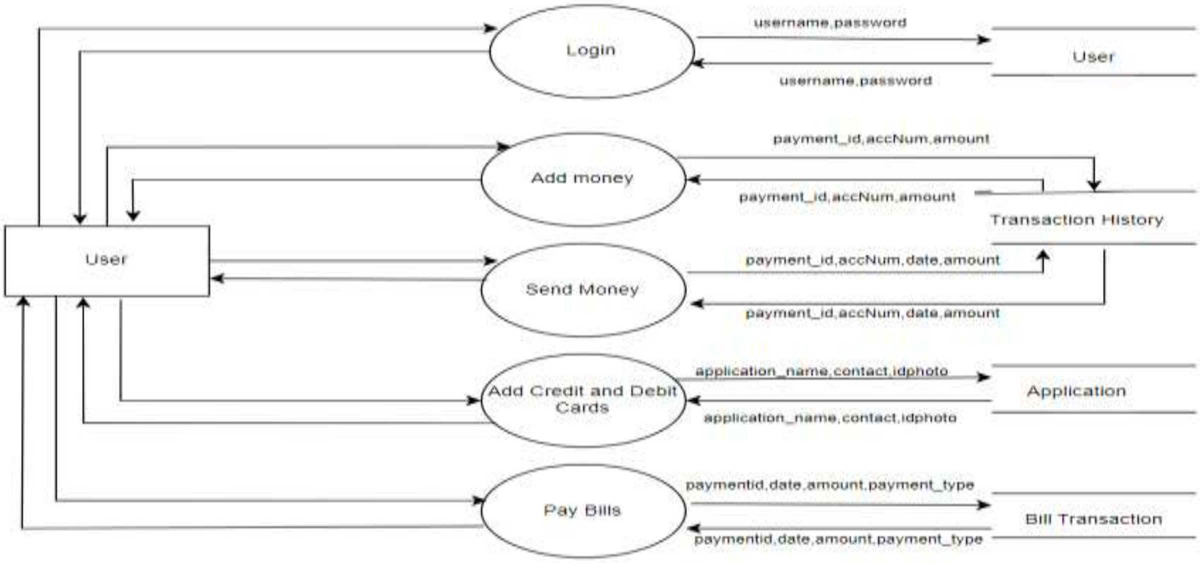

Login
username,password
User
username,password
Add money
payment_id,accNum,amount
payment_id,accNum,amount
Transaction History
User
Send Money
payment_id,accNum,date,amount
payment_id,accNum,date,amount
Add Credit and Debit Cards
application_name,contact,idphoto
Application
application_name,contact,idphoto
Pay Bills
paymentid,date,amount,payment_type
Bill Transaction
paymentid,date,amount,payment_type

Figure 4.5:Level 1 Dfd for User

CHAPTER 5

DETAILED DESIGN

5.1 USE CASE DIAGRAM

The diagram illustrating use cases for the FinEdge Application illustrates the interactions between users, administrators, and the system.

System: Represented by a rectangle, the FinEdge Application encompasses all functionalities and services, including user and admin interactions, financial transactions, and application processing.

Actor: Depicted as stick figures, the primary actors are Users and Admins.

Use case: Use cases, shown as ellipses, represent various operations within the FinEdge Application

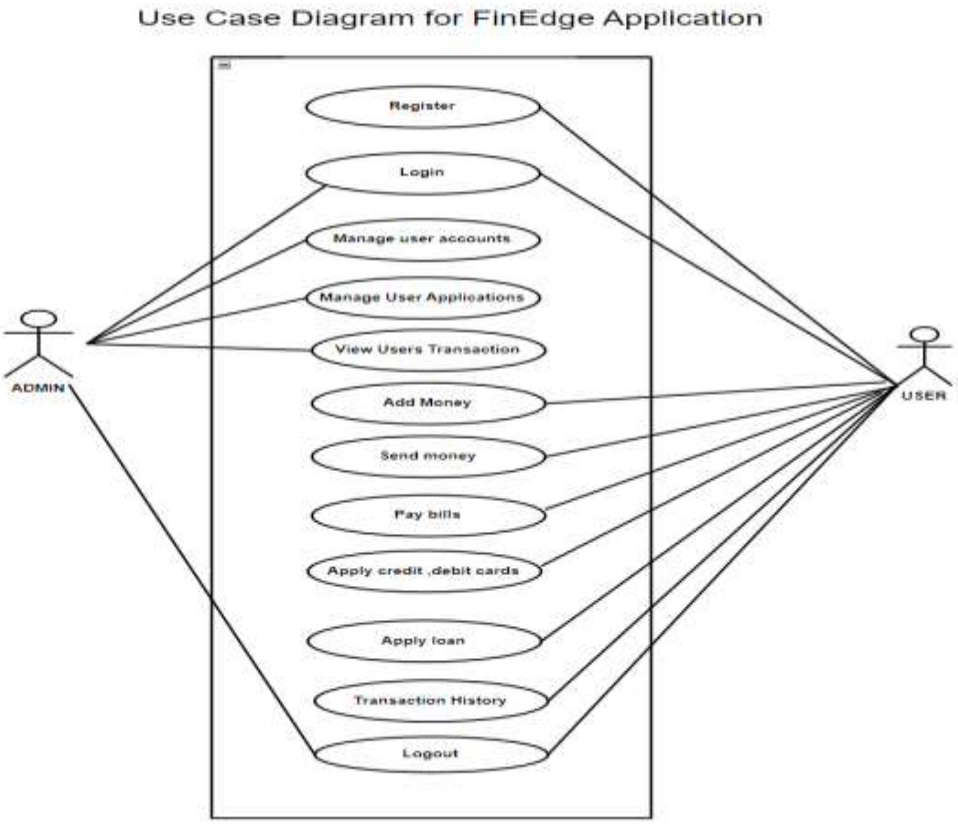


Figure 5.1:Use case diagram of FinEdge Application

5.2 SEQUENCE DIAGRAM

The diagram depicting the sequence illustrates how the components of the FinEdge Applications interact with each other over time. It describes the flow of communication and exchange of messages between objects within the system. In the framework of the FinEdge Application, the diagram illustrating the sequence highlights the interactions between users, administrators, and the system components as they perform various financial activities and manage user accounts. This diagram, also known as an event diagram, helps in visualizing the step-by-step execution of processes within the application.

5.2.1 SEQUENCE DIAGRAM FOR ADMINISTRATOR

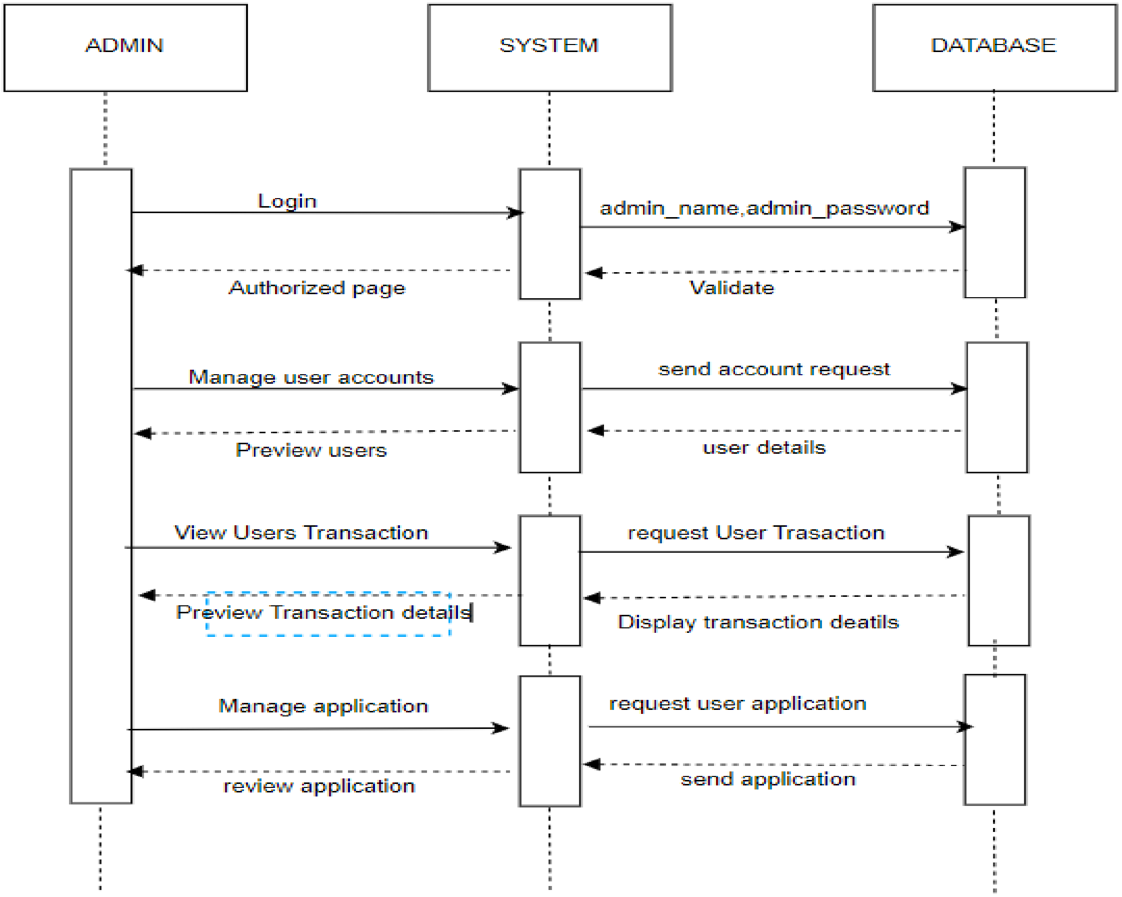


Figure 5.2.1:Sequence Diagram for Administrator

5.2.2 SEQUENCE DIAGRAM FOR THE USER

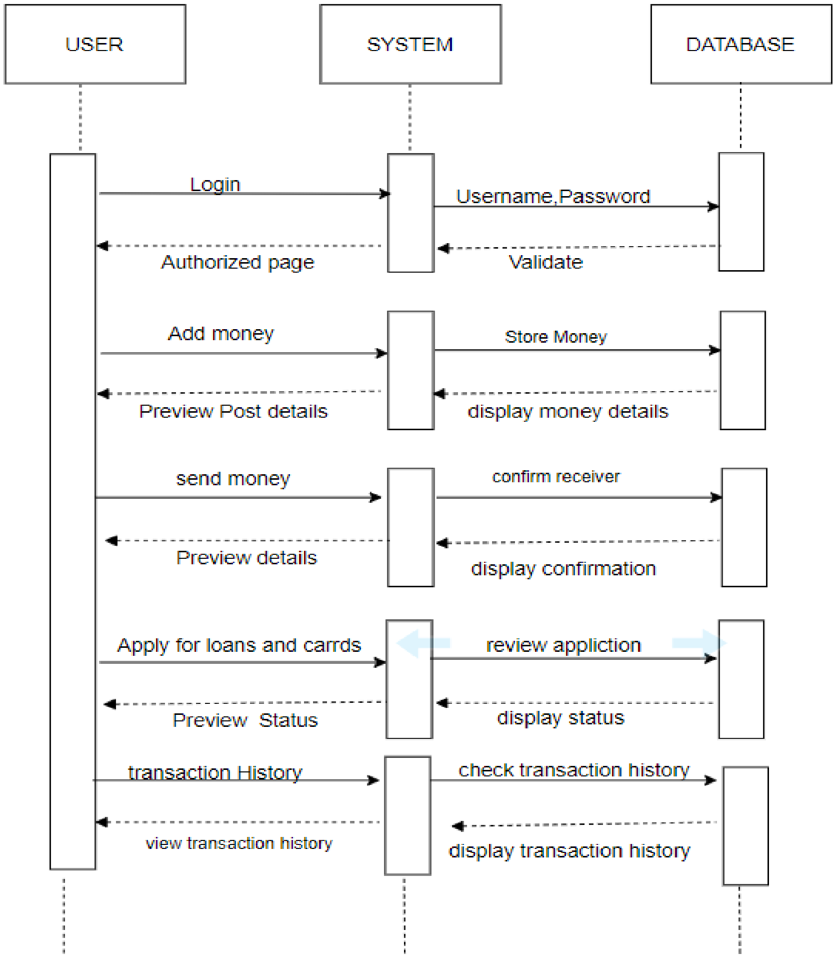


Figure 5.2.2: Sequence Diagram for the User

5.3 ACTIVITY DIAGRAMS

An activity diagram visually represents the workflow in the FinEdge Application, showing how one action connects to the next. It illustrates the steps involved in processes such as user registration, adding money, checking balance, transferring money, viewing transaction history, paying bills, applying for cards, and applying for loans. This helps us understand how the system functions, ensuring all steps are logically connected and providing a clear overview of the application's processes.

5.3.1 USERS AND ADMININSTRATOR

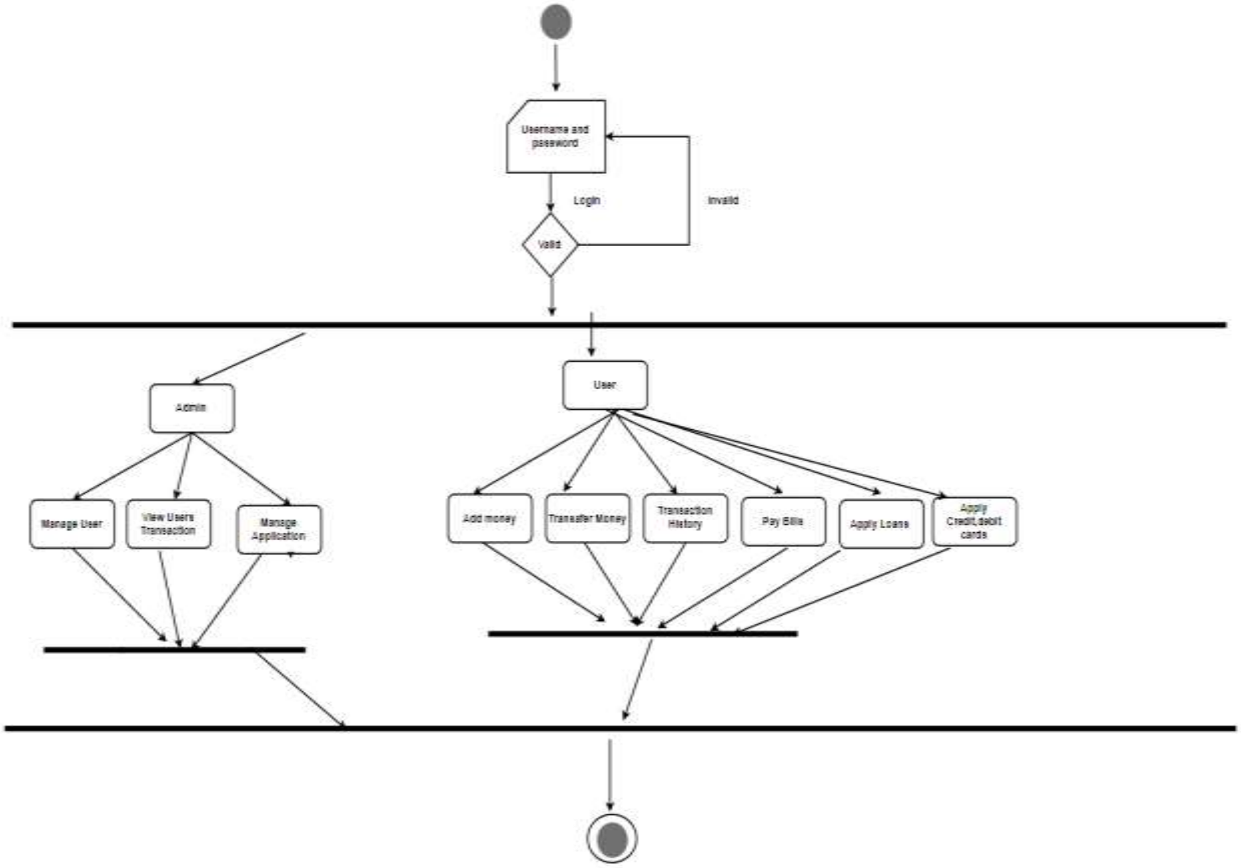


Figure 5.3.1: Activity Diagram for Users and Admin

5.4 DATABASE DESIGN

The FinEdge Application's database design includes several key tables crucial for the application's functionality. These tables manage data related to users, transactions, and applications, ensuring efficient and secure data handling.

5.1 USER TABLE

Table 5.1:User Table

Attribute	Data Type	Constraint	Descriptions
Id	Int	Primary Key	Unique Id For Each User
Name	Varchar(100)	Not Null	Name
Email	Varchar(100)	Not Null	Email of the User
Pass	Varchar(100)	Not Null	Passwords
Dob	Varchar(100)	Not Null	Date of Birth
Contact	Varchar(100)	Not Null	Contact Number
Address	Varchar(100)	Not Null	Address

5.2 BANK REGISTER TABLE

Table 5.2:NEWBANKUSER Table

Attribute	Data Type	Constraints	Description
Id	Int	PK,NN,AI	Primar Key,Auto-Incremented
Userid	Int	Not Null	User Id, References The User
Name	Varchar(200)	Not Null	Full name Of the User
Dob	Varchar(200)	Not Null	Date of Birth Of the User
City	Varchar(200)	Not Null	City Of The User
Pincode	Varchar(200)	Not Null	Postal Code
Phnumber	Varchar(200)	Not Null	Phone Number
Email	Varchar(200)	Not Null	Email Address
Aadhar	Varchar(200)	Not Null	Aadhar Number
Gender	Varchar(200)	Not Null	Gender
Initialamt	Varchar(200)	Not Null	Initial Amount Deposited

Username	Varchar(200)	Not Null	Username For Login
Password	Varchar(200)	Not Null	Password For Login
Accnum	Varchar(200)	Not Null	Account Number
Pin_Number	Varchar(200)	Not Null	Pin For The Account

5.3 LOGIN HISTORY TABLE

Table 5.3:LOGIN HISTORY Table

Attribute	Data Type	Constraint	Descriptions
Id	Int	PK,NN,AI	Primar Key,Auto-Incremented
User_Id	Int	Not Null	Unique User Id
Logintime	Varchar(200)	Not Null	Notes The Time When Logged In

5.4 EXISTING BANK USERS TABLE

Table 5.4:EXISTINGBANKUSERS Table

Attribute	Data Type	Constraint	Descriptions
Id	Int	PK,NN,AI	Primar Key,Auto-Incremented
User_Id	Int	Not Null	Unique User Id
Username	Varchar(200)	Not Null	Existing User Name
Password	Varchar(200)	Not Null	Password
Accountnum	Varchar(200)	Not Null	Existing Account Number

5.5 TRANSACTION HISTORY TABLE

Table 5.5:TRANSACTIONDETAILS Table

Attribute	Data Type	Constraints	Description
Id	Int	PK,NN,AI	Primar Key,Auto-Incremented
Payment_Id	Varchar(200)	Not Null	Payment Id
Accnum	Varchar(200)	Not Null	Account Number
Date	Timestamp	Not Null	Transaction Date
Description	Varchar(200)	Not Null	Description About The Payment
Amount	Varchar(200)	Not Null	Amount
Type	Varchar(200)	Not Null	Type Of Payment Method

5.6 CREDIT CARD APPLICATION TABLE

Table 5.6:CREDICARDAPPLICATION Table

Attribute	Data Type	Constraints	Descriptions
Application_Id	Int	PK,NN,AI	Primar Key,Auto-Incremented
Name	Varchar(200)	Not Null	Name of The Applicant
Dateofbirth	Date	Not Null	Date of Birth
Contact	Varchar(200)	Not Null	Contact Number
Address	Text	Not Null	Address Of The Applicant
Id	Varchar(200)	Not Null	Id Type
Idnumber	Varchar(200)	Not Null	Id Card Number
Professionname	Varchar(200)	Not Null	Name of The Profession Of Applicant

Monthlysalary	Decimal(10,2)	Not Null	Monthly Income
Idphoto	Varchar(200)	Not Null	Photo Of Id Card
Status	Enum	Not Null	Current Status Of Application
Application_Date	Timestamp	Not Null	Date Of Submissiom

5.7 DEBIT CARD APPLICATION TABLE

<table>
<tr><th colspan="4">5.7:DEBITCARDAPPLICATION
Table</th></tr>
<tr><th>Attribute</th><th>Data Type</th><th>Constraints</th><th>Descriptions</th></tr>
<tr><td>Application_Id</td><td>Int</td><td>PK,NN,AI</td><td>Primar Key,Auto-Incremented</td></tr>
<tr><td>Name</td><td>Varchar(200)</td><td>Not Null</td><td>Name of The Applicant</td></tr>
<tr><td>Bankacc</td><td>Varchar(200)</td><td>Not Null</td><td>Bank Account Number</td></tr>
<tr><td>Dateofbirth</td><td>Varchar(200)</td><td>Not Null</td><td>Date of Birth</td></tr>
<tr><td>Contact</td><td>Varchar(200)</td><td>Not Null</td><td>Contact Number</td></tr>
<tr><td>Address</td><td>Varchar(200)</td><td>Not Null</td><td>Address Of The Applicant</td></tr>
<tr><td>Id</td><td>Varchar(200)</td><td>Not Null</td><td>Id Type</td></tr>
<tr><td>Idnumber</td><td>Varchar(200)</td><td>Not Null</td><td>Id Card Number</td></tr>
<tr><td>Idphoto</td><td>Varchar(200)</td><td>Not Null</td><td>Photo Of Id Card</td></tr>
<tr><td>Created_On</td><td>Timestamp</td><td>Not Null</td><td>Created Time Of Application</td></tr>
<tr><td>Debit_Pin</td><td>Varchar(200)</td><td>Not Null</td><td>Dabit Card Pin Set By User</td></tr>
</table>

5.8 CREDIT CARD DETAILS TABLE

5.8:CREDITCARD Table			
Attribute	**Data type**	**Constraint**	**Description**
Creditcard_Id	Int	PK,NN,AI	Primar Key,Auto-Incremented
Userid	Int	Not Null	Unique User Id
Application_Id	Int	Not Null	Application Id For Application
Name	Varchar(200)	Not Null	Name of The Credit Card Holder
Dateofbirth	Date	Not Null	Date Of Birth
Contact	Varchar(200)	Not Null	Contact Number
Address	Text	Not Null	Address Of The Applicant
Id_Type	Varchar(200)	Not Null	Id Type
Id_Number	Varchar(200)	Not Null	Id Card Number
Profession_Name	Varchar(200)	Not Null	Name Of The Profession
Monthly_Salary	Decimal(10,2)	Not Null	Monthly Salasy
Id_Photo	Varchar(200)	Not Null	Id Card Photo
Credit_Card_Number	Varchar(200)	Not Null,Unique	Unique 16 Digit Credit Card Number
Card_Type	Varchar(200)	Not Null	Credit Card Type
Expiry_Date	Date	Not Null	Expiry Date
Cvv	Varchar(3)	Not Null	Ccv Of Card
Credit_Limit	Decimal(10,2)	Not Null	Maximum Credit Limit
Available_Credit	Decimal(10,2)	Not Null	Available Amount Of Credit
Issuance_Date	Date	Not Null	Date Of Issue Of Credit Card
Billing_Cycle_Start	Date	Not Null	Monthly Billing Cycle Start
Billing_Cycle_End	Date	Not Null	Monthly Billing Cycle End
Created_At	Timestamp	Not Null	Created Date And Time

5.9 DEBIT CARD DETAILS TABLE

5.9:DEBITCARD Table

Attribute	Data type	Constraints	Description
Card_Id	Int	PK,NN,AI	Primar Key,Auto-Incremented
Userid	Int	Not Null	Unique User Id
Application_Id	Int	Not Null	Application Id For Application
Name	Varchar(200)	Not Null	Name of The Debit Card Holder
Dateofbirth	Date	Not Null	Date Of Birth
Contact	Varchar(200)	Not Null	Phone Number
Address	Text	Not Null	Address of The Applicant
Id_Type	Varchar(200)	Not Null	Id Type
Id_Number	Varchar(200)	Not Null	Id Card Number
Id_Photo	Varchar(200)	Not Null	Id Card Photo
Debit_Card_Number	Varchar(200)	Not Null,Unique	Unique 16 Digit Debit Card Number
Expiry_Date	Date	Not Null	Expiry Date
Cvv	Varchar(3)	Not Null	Ccv Of Card
Issuance_Date	Date	Not Null	Date Of Issue Of Debit Card

5.10 UTILITY BILLS TRANSACTION DETAILS TABLE

Table 5.10:UTILITYBILLS Table			
Attribute	**Data Type**	**Constraints**	**Description**
Id	Int	PK,NN,AI	Primar Key,Auto-Incremented
Paymentid	Varchar(200)	Not Null	Payment Id
Phone_Number	Varchar(200)	Not Null	Phone Number
Date	Timestamp	Not Null	Transaction Date
Description	Varchar(200)	Not Null	Description About The Payment
Amount	Varchar(200)	Not Null	Amount
Payment_Type	Varchar(200)	Not Null	Type Of Payment Method

5.11 ADMINN LOGIN TABLE

Table 5.3:LOGIN HISTORY Table			
Attribute	**Data Type**	**Constraints**	**Description**
Id	Int	PK,NN,AI	Primar Key,Auto-Incremented
Adminname	Varchar(200)	Not Null	Enter A Name
Password	Varchar(200)	Not Null	Password

5.4.1 ER DIAGRAM

Here, I've just included the key components of my project and have illustrated how they relate to one another.

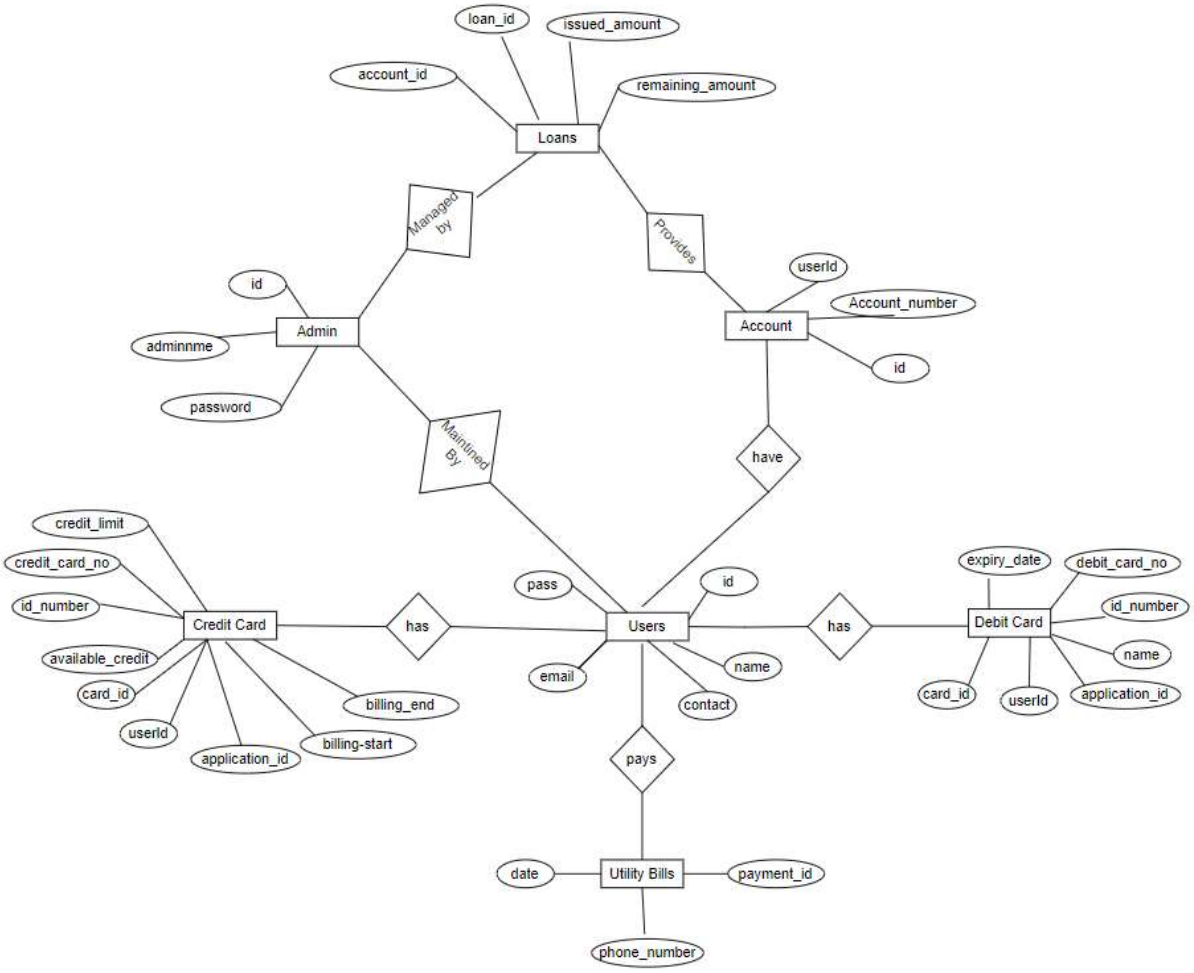


Figure 5.4.1:ER Diagram for FinEdge Appliction

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

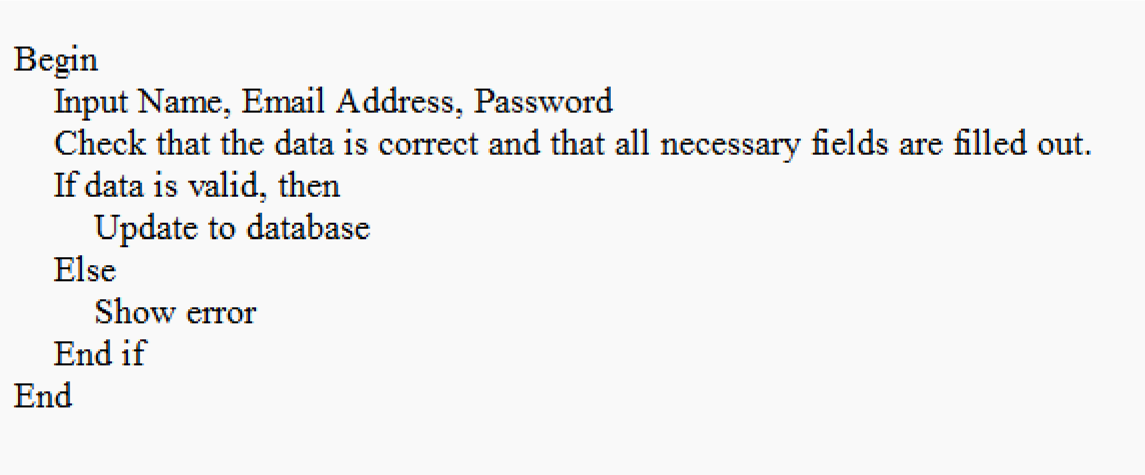
Implementation is the procedure of transforming a designed plan into a working application. For the FinEdge Application, the implementation involves converting all the design specifications and requirements into actual code. This procedure encompasses not only coding but also testing various scenarios to guarantee that the system functions as intended.

Adhering to coding standards is crucial during the implementation phase. These standards provide a clear and consistent approach to writing code, making it understandable and maintainable by developers other than the original authors. By following these standards, we ensure that the FinEdge Application can be easily modified and updated in the future, similar to how uniform naming conventions are used in system components like classes and functions.

6.2 PSEUDO CODE

6.2.1 PSEUDOCODE FOR USER REGISTRATION PAGE

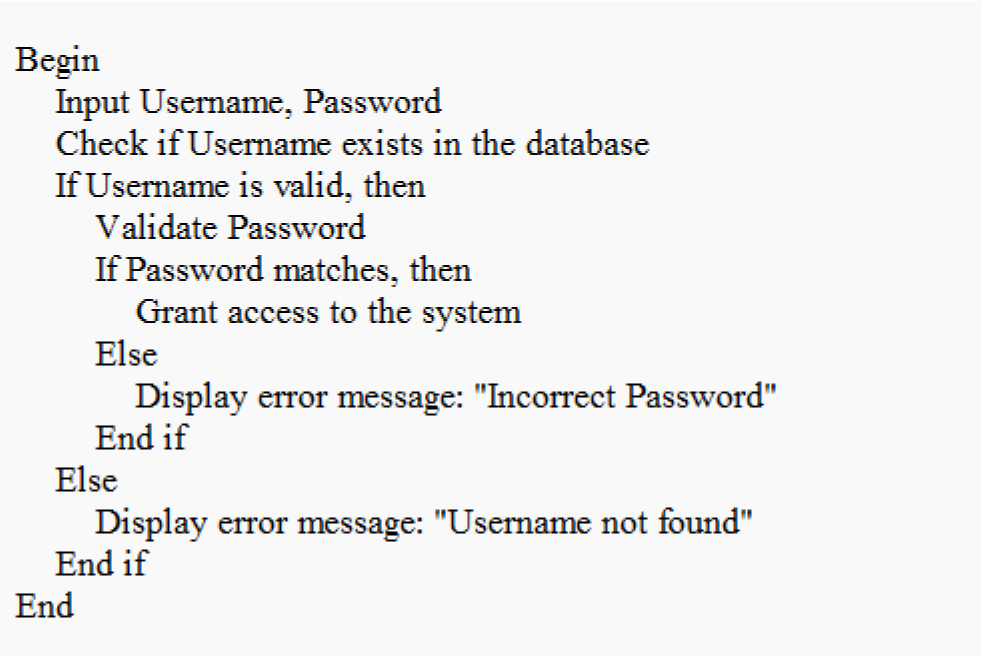
Before accessing the FinEdge Application, users must register by providing essential details such as their name, email address, and password. The system checks whether all necessary fields are filled out and if the provided information is valid. If the data is accurate, it is stored in the database; otherwise, a issue is displayed. Once registration is complete, users are prompted to update additional information like their contact number, address, and date of birth. This ensures the system has comprehensive user details before allowing further interactions



```
Begin
    Input Name, Email Address, Password
    Check that the data is correct and that all necessary fields are filled out.
    If data is valid, then
        Update to database
    Else
        Show error
    End if
End
```

6.2.2 PSEUDOCODE FOR LOGIN PAGE

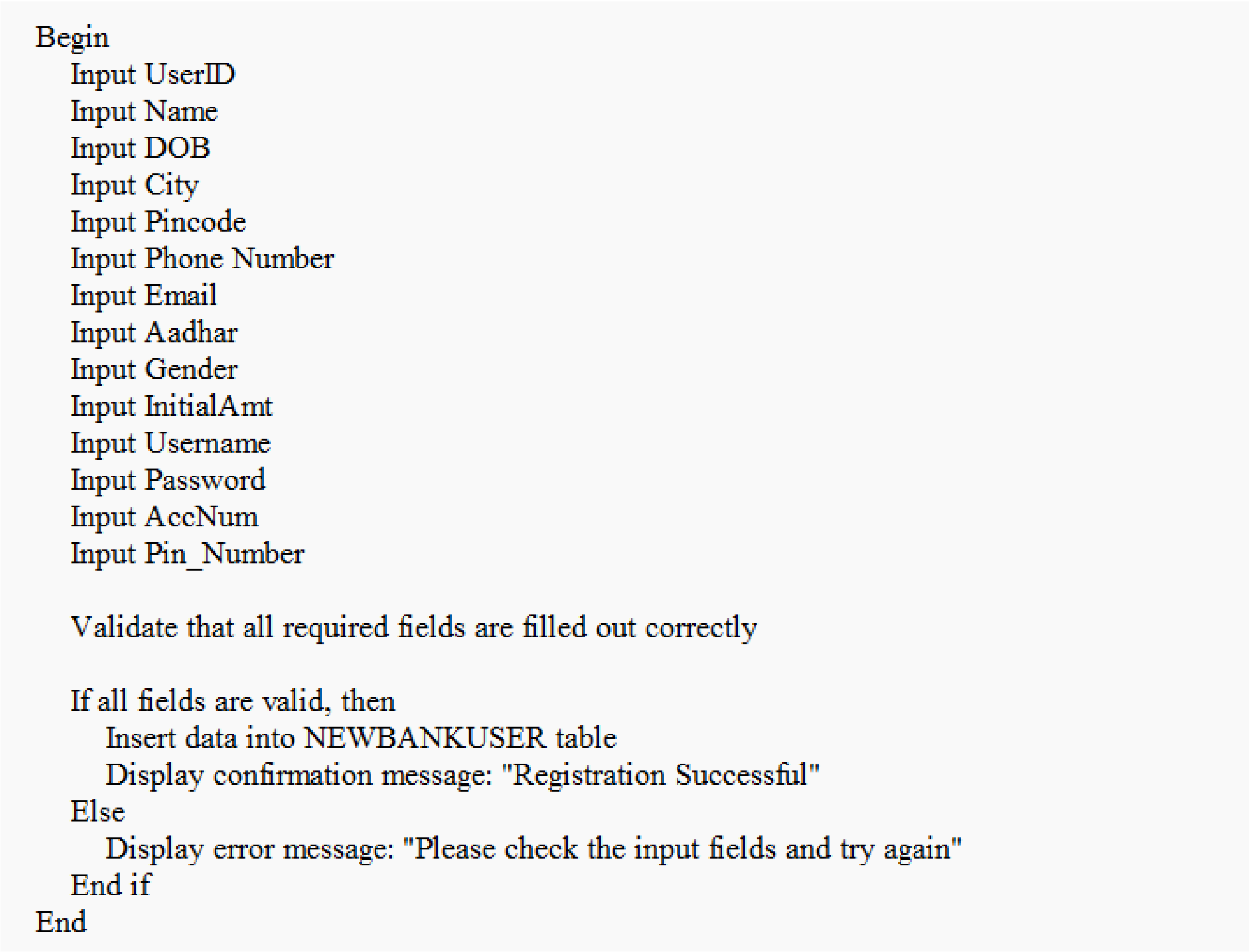
In the FinEdge Application, users must log in with their username and password to log in to the system. The login process verifies whether the provided username exists in the database. If the username exists , the system then checks if the corresponding password is correct. Upon successful validation, user granted access to application. If the credentials are incorrect, an error message displayed.



```
Begin
    Input Username, Password
    Check if Username exists in the database
    If Username is valid, then
        Validate Password
        If Password matches, then
            Grant access to the system
        Else
            Display error message: "Incorrect Password"
        End if
    Else
        Display error message: "Username not found"
    End if
End
```

6.2.3 PSEUDOCODE FOR BANK REGISTRATION

After successfully logging in, users must complete the new bank registration by providing additional details such as name, date of birth, city, and other personal information. The system validates each input and stores the data in the NEWBANKUSER table. If the data is valid, it is saved; otherwise, an issue will be displayed.



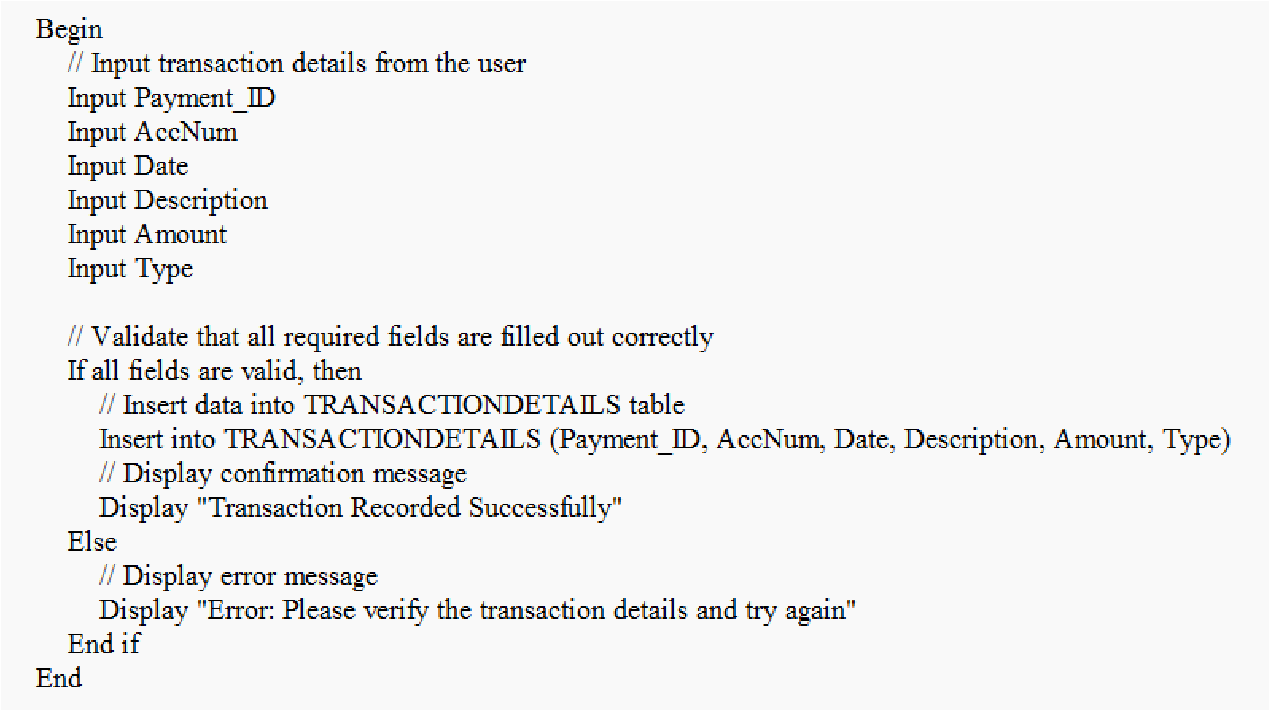
```
Begin
    Input UserID
    Input Name
    Input DOB
    Input City
    Input Pincode
    Input Phone Number
    Input Email
    Input Aadhar
    Input Gender
    Input InitialAmt
    Input Username
    Input Password
    Input AccNum
    Input Pin_Number

    Validate that all required fields are filled out correctly

    If all fields are valid, then
        Insert data into NEWBANKUSER table
        Display confirmation message: "Registration Successful"
    Else
        Display error message: "Please check the input fields and try again"
    End if
End
```

6.2.4 PSEUDO CODE FOR TRANSACTION DETAILS

This pseudo code describes the process of recording a transaction in the TRANSACTIONDETAILS table of the FinEdge Application. The process involves inputting necessary transaction details, validating the input, and updating the database. If the input is valid, the transaction is successfully recorded; otherwise, an error message is displayed.

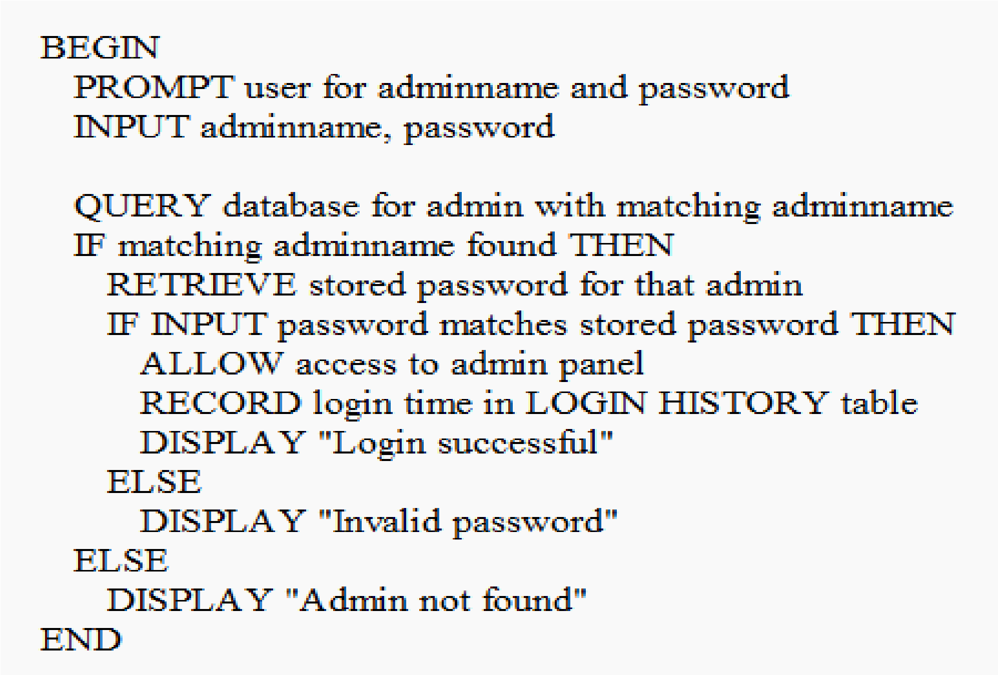


```
Begin
   // Input transaction details from the user
   Input Payment_ID
   Input AccNum
   Input Date
   Input Description
   Input Amount
   Input Type

   // Validate that all required fields are filled out correctly
   If all fields are valid, then
      // Insert data into TRANSACTIONDETAILS table
      Insert into TRANSACTIONDETAILS (Payment_ID, AccNum, Date, Description, Amount, Type)
      // Display confirmation message
      Display "Transaction Recorded Successfully"
   Else
      // Display error message
      Display "Error: Please verify the transaction details and try again"
   End if
End
```

6.2.4 PSEUDOCODE FOR ADMIN LOGIN

The system alerts the admin for their username and password, checks these credentials against the database, and grants access if they match.



```
BEGIN
   PROMPT user for adminname and password
   INPUT adminname, password

   QUERY database for admin with matching adminname
   IF matching adminname found THEN
      RETRIEVE stored password for that admin
      IF INPUT password matches stored password THEN
         ALLOW access to admin panel
         RECORD login time in LOGIN HISTORY table
         DISPLAY "Login successful"
      ELSE
         DISPLAY "Invalid password"
   ELSE
      DISPLAY "Admin not found"
END
```

CHAPTER 7

SOFTWARE TESTING

7.1 INTRODUCTION

Testing is a crucial stage in software development, verifying that the program functions as discussed. The primary objective of Testing aims to identify and correct any errors or issues within the software, thereby ensuring its reliability and effectiveness. In FinEdge application, extensive testing was conducted to confirm that the system fulfills user requirements and operates seamlessly. We employed different testing methods, includes unit testing to verify individual components and integration testing to address the system's overall performance. By simulating various scenarios and creating test data, we ensured that the application behaves at different conditions.

7.2 TESTING OBJECTIVE

The main goal of testing aims to ensure that the software aligns with user requirements and performs effectively. Through rigorous testing, we aim to enhance the quality of the FinEdge application, ensuring that the final product is free of errors and meets all validation and verification standards. This involves confirming that each module adheres to the specified requirements during implementation. Our testing process for this project involves Unit Testing, Integration Testing, functional Testing, and also regression Testing, where we systematically evaluate each module

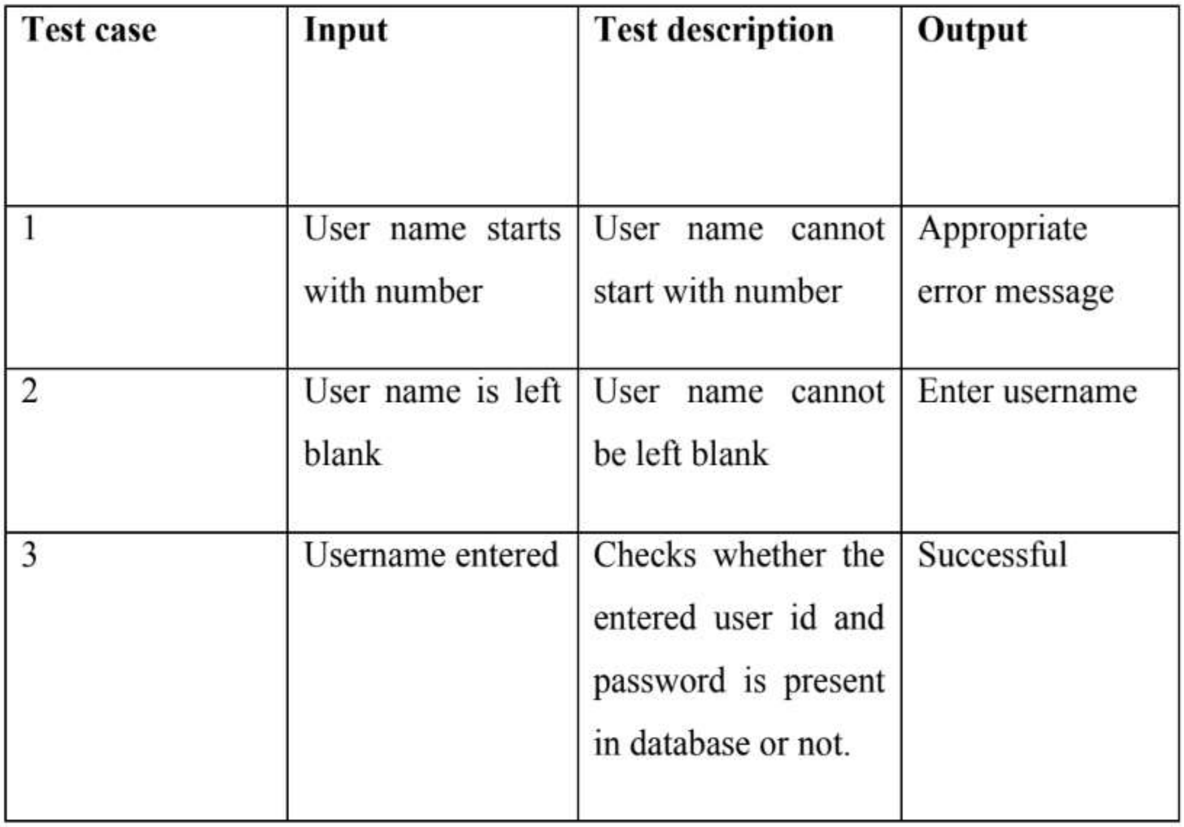
7.3 UNIT TESTING

Unit testing focuses on validating the functionality of specific code segments, often at the individual function level. In the FinEdge application, developers conducted unit tests to ensure that each function within the software operates correctly. This practice helps in detecting defects early in the development process, reducing risks, costs, and time. By testing the smallest units of the software—each with defined inputs and outputs—we ensured that every component functions as intended, contributing to the overall reliability that the system provides.

7.3.1 UNIT TEST CASES

This involves examining the smallest individual components of the FinEdge Application. At this stage, identifying and fixing issues is more manageable, time-efficient, and cost-effective. Each module within the FinEdge Application was tested separately with specific inputs, ensuring they produce the expected results. This thorough approach allows for the early detection of errors, contributing to a smoother integration of modules later in the development process.

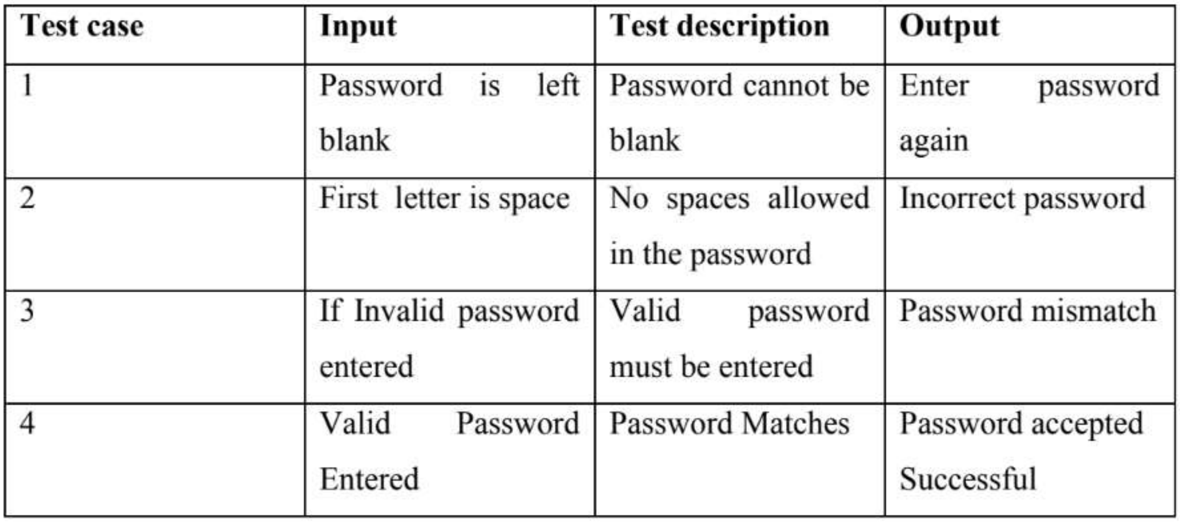
7.3.2 TESTING FOR VALID USER NAME



Test case	Input	Test description	Output
1	User name starts with number	User name cannot start with number	Appropriate error message
2	User name is left blank	User name cannot be left blank	Enter username
3	Username entered	Checks whether the entered user id and password is present in database or not.	Successful

Table 7.3.2 : Testing for Valid User Name

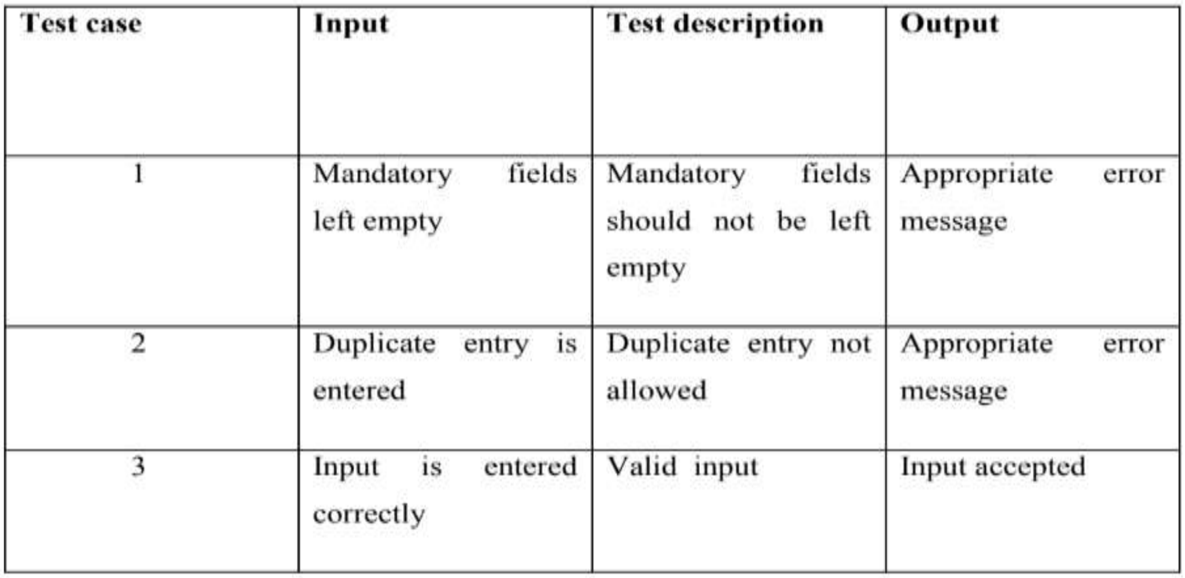
7.3.3 TSTING FOR VALID PASSWORD



Test case	Input	Test description	Output
1	Password is left blank	Password cannot be blank	Enter password again
2	First letter is space	No spaces allowed in the password	Incorrect password
3	If Invalid password entered	Valid password must be entered	Password mismatch
4	Valid Password Entered	Password Matches	Password accepted Successful

Table 7.3.3: Test for valid password

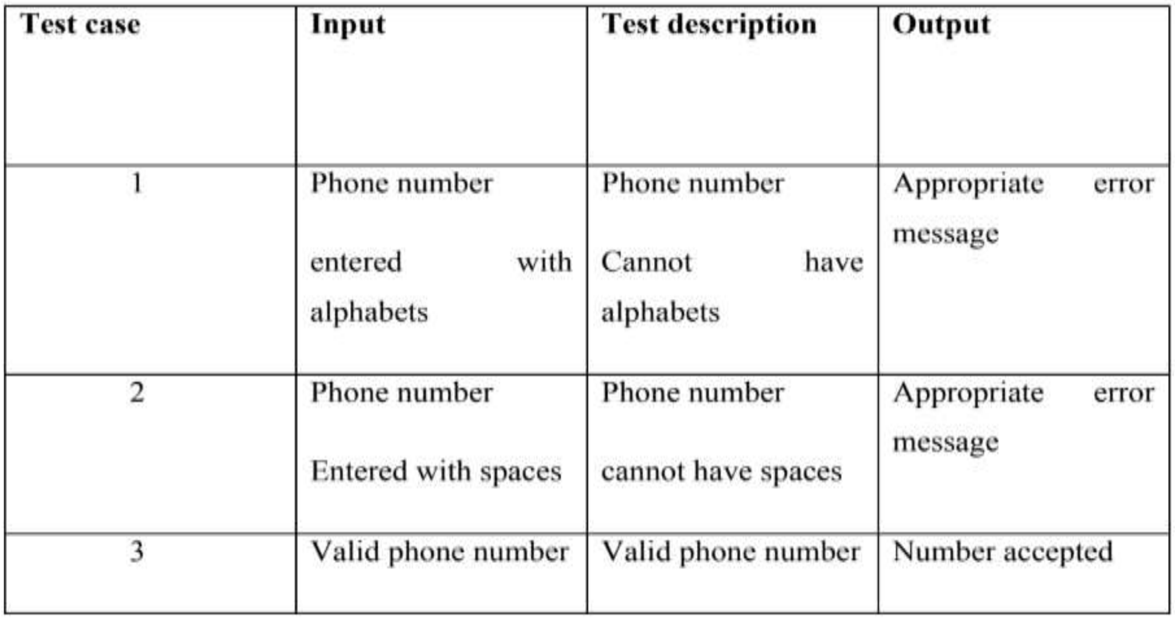
7.3.4 TESTING FOR DATA INSERTION



Test case	Input	Test description	Output
1	Mandatory fields left empty	Mandatory fields should not be left empty	Appropriate error message
2	Duplicate entry is entered	Duplicate entry not allowed	Appropriate error message
3	Input is entered correctly	Valid input	Input accepted

Table 7.3.4:Testing for Data Insertion

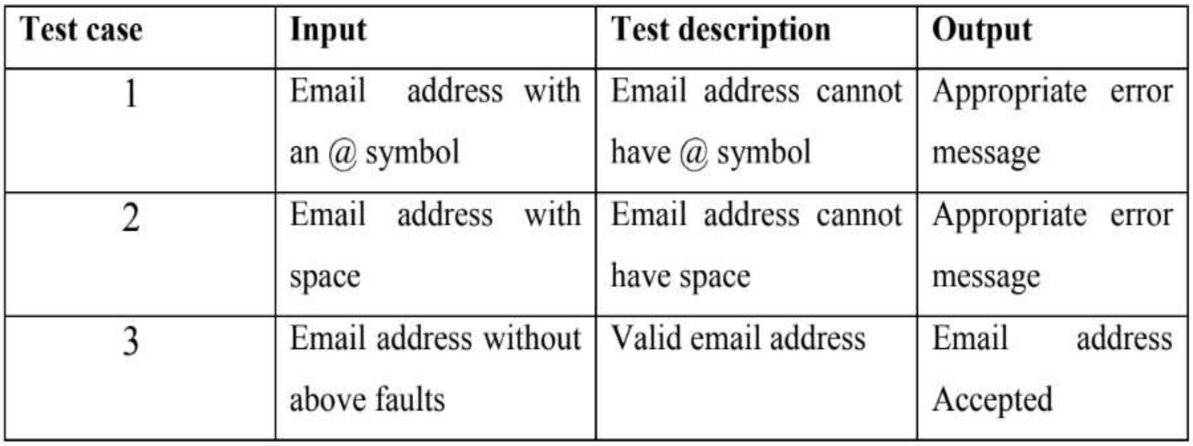
7.3.5 TESTING FOR CONTACT NUMBER



Test case	Input	Test description	Output
1	Phone number entered with alphabets	Phone number Cannot have alphabets	Appropriate error message
2	Phone number Entered with spaces	Phone number cannot have spaces	Appropriate error message
3	Valid phone number	Valid phone number	Number accepted

Table 7.3.5: Testing for contact number

7.3.6 TESTING FOR EMAIL ADDRESS



Test case	Input	Test description	Output
1	Email address with an @ symbol	Email address cannot have @ symbol	Appropriate error message
2	Email address with space	Email address cannot have space	Appropriate error message
3	Email address without above faults	Valid email address	Email address Accepted

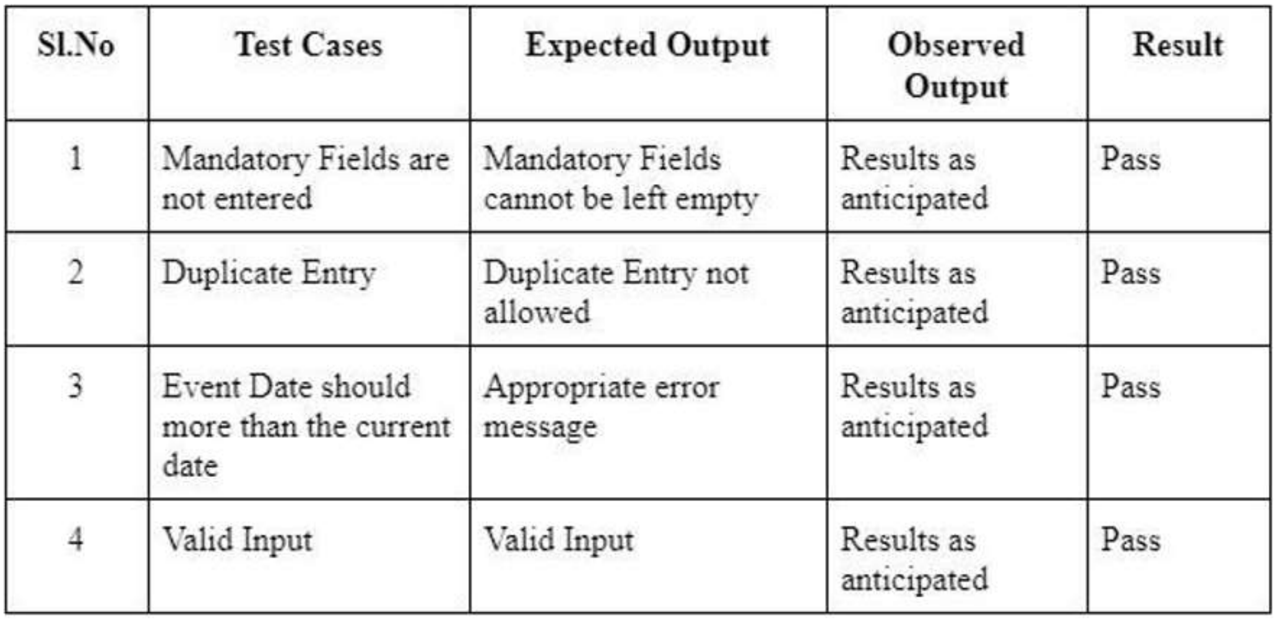
Table 7.3.6: Testing for email address

7.4 INTEGRATION TESTING

In the FinEdge Application, integration testing was conducted to confirm that all modules and components function cohesively as a unified system. This testing process was crucial in identifying and Handling any challenges that arose from the interaction between different part of the software. We began by testing smaller, related groups of components and then gradually integrated these groups to ensure they function correctly together as planned. This method allowed us to confirm that the entire system operates smoothly and effectively.

7.4.1 INTEGRATION TEST CASES

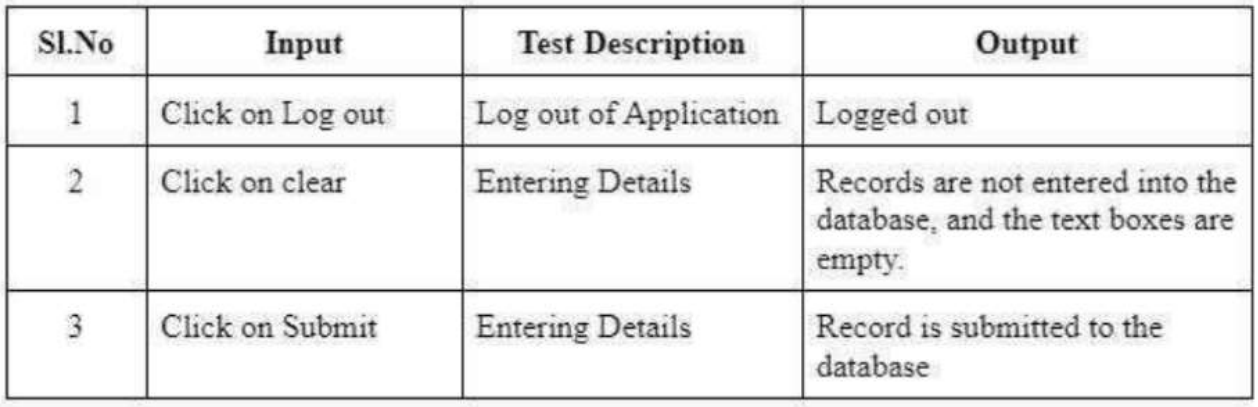
In the FinEdge Applicationintegration test cases are used to designed , examine how different components of system interact when combined. Each test case was carefully constructed to simulate real-world scenarios where multiple modules need to work together. During these tests, we acted as detectives, uncovering any issues that arose when integrating different parts of the applicationThis process was crucial for verifying that all components of the system, when combined, deliver the expected results and operate without errors.



Sl.No	Test Cases	Expected Output	Observed Output	Result
1	Mandatory Fields are not entered	Mandatory Fields cannot be left empty	Results as anticipated	Pass
2	Duplicate Entry	Duplicate Entry not allowed	Results as anticipated	Pass
3	Event Date should more than the current date	Appropriate error message	Results as anticipated	Pass
4	Valid Input	Valid Input	Results as anticipated	Pass

Table 7.4.1: Integration Test Cases for Data Insertion

7.4.2 OTHER TEST CASES



Sl.No	Input	Test Description	Output
1	Click on Log out	Log out of Application	Logged out
2	Click on clear	Entering Details	Records are not entered into the database, and the text boxes are empty.
3	Click on Submit	Entering Details	Record is submitted to the database

Table 7.4.2: Other Test Cases

CHAPTER 8

CONCLUSION

The FinEdge Application was tested on Windows 10 and later versions to ensure optimal performance and reliability. In this system, the admin is central to managing all user activities, financial transactions, and service approvals. By integrating key roles such as department heads and financial managers, the system significantly reduces manual paperwork and streamlines operational workflows. Users, particularly SMEs, can securely log in with their credentials, apply for some services like differet types of loans, manage investments, and conduct digital banking operations. The application's approval workflow is designed to be efficient, allowing for quick processing of requests and minimizing delays. A thorough approach not only enhances productivity and Additionally offers users a dependable platform for managing their financial operations, ultimately saving time, reducing costs, and improving decision-making for businesses.

CHAPTER 9

FUTURE ENHANCEMENTS

In line with the iterative development model, the FinEdge Application will continuously evolve to meet user needs and industry standards. Future enhancements include introducing automated reminders for users, such as SMEs and financial managers, to alert them of pending financial transactions or approvals before deadlines. Additionally, the system will be adapted to incorporate emerging technologies like AI-driven analytics and blockchain for enhanced security and efficiency. These improvements will ensure the FinEdge Application remains a robust and forward-thinking solution, continuously enhancing user experience and operational effectiveness.

APPENDIX A REFERENCES

[1] Subramanian, Vasan. *Pro MySQL Database Design.* Apress, 2017. Available online at: https://link.springer.com/book/10.1007/978-1-4842-3138-8

[2] Paul DuBois. *MySQL Cookbook: Solutions for Database Developers and Administrators.* O'Reilly Media, 2014. Available online at: https://www.oreilly.com/library/view/mysql-cookbook/059652708X/

[3] Bartholomew, Richard, and Sheeri Cabral. *Learning MySQL.* O'Reilly Media, 2015. Available online at: https://www.oreilly.com/library/view/learning-mysql/0596008643/

[4] Silva, Edward. *Mastering MySQL.* Packt Publishing, 2018. Available online at: https://www.packtpub.com/product/mastering-mysql/9781788622660

[5] Jalote, Pankaj. A Unified Approach to Software Engineering. Available online at: https://www.academia.edu/4660479/an_integral_approach_to_software_engineering_BY_PANKAJ_JALOTE

[6] *Learned tutorials of React and Node.js* on YouTube. Available at: https://www.youtube.com/

APPENDEX B USER MANUAL

1.Home Page

The homepage of the FinEdge Application serves as the gateway to all functionalities, featuring links such as services, about, customer care, and login. It includes engaging advertising images and an EMI calculator for users to quickly calculate EMIs. Additionally, the homepage displays the company's vision, mission, and values, ensuring users have a comprehensive understanding of the brand's goals and services.

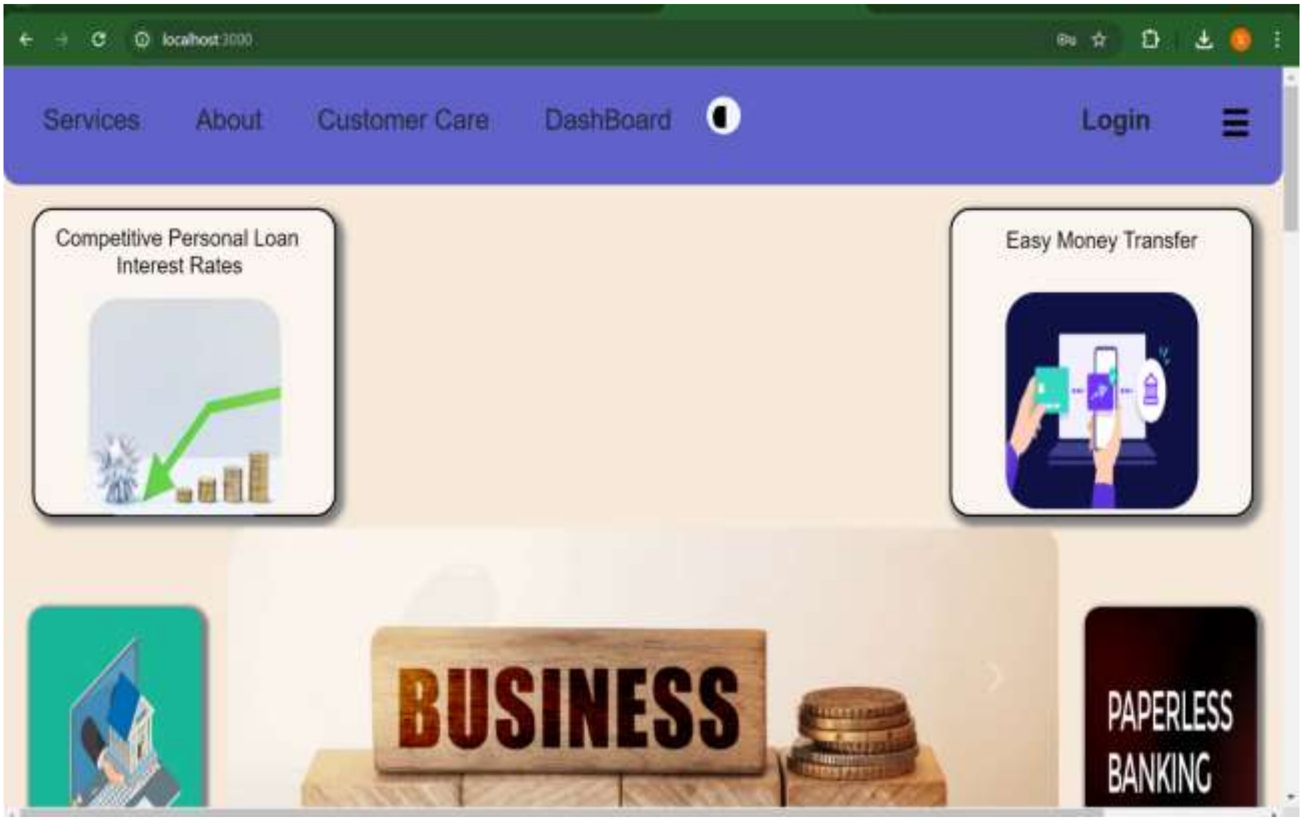
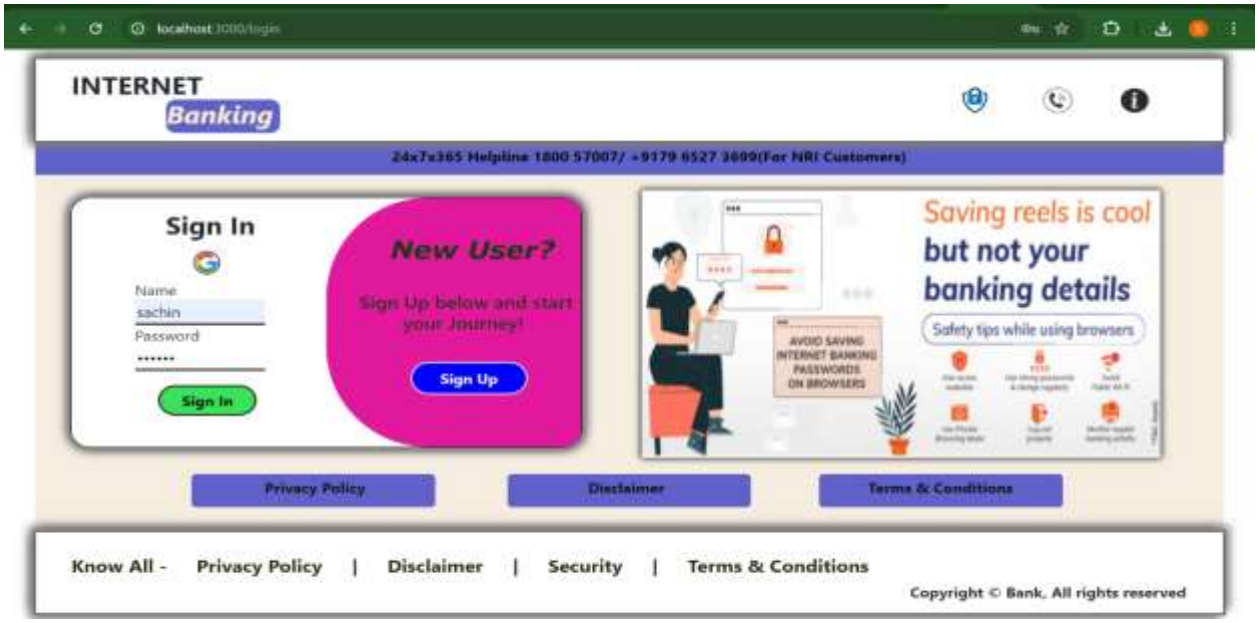


Figure 1:Home page

2. User Login Page

The FinEdge Application's user login page lets users securely access their accounts by entering their account number, username, and password, leading them directly to their personalized dashboard.

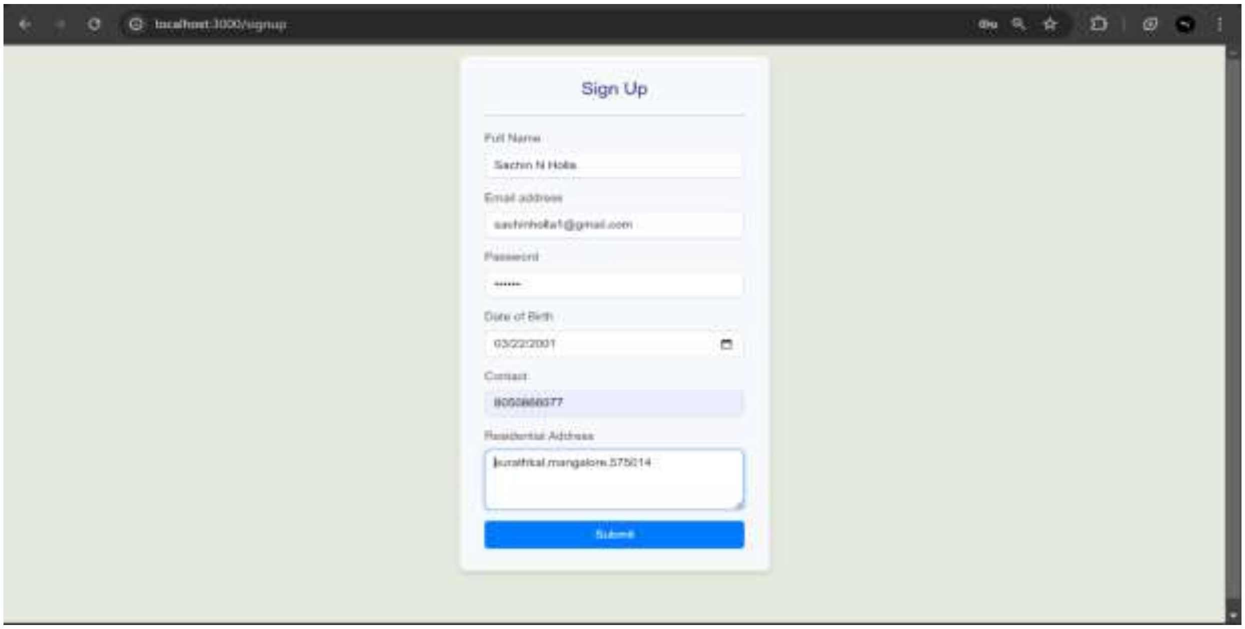

localhost:3000/login
INTERNET
Banking
24x7x365 Helpline 1800 57007/ +9179 6527 3699(For NRI Customers)
Sign In
New User?
Sign Up below and start your Journey!
Sign Up
Sign In
Saving reels is cool
but not your
banking details
Safety tips while using browsers
AVOID SAVING INTERNET BANKING PASSWORDS ON BROWSERS
Privacy Policy
Disclaimer
Terms & Conditions
Know All - Privacy Policy | Disclaimer | Security | Terms & Conditions
Copyright © Bank. All rights reserved

Name
sachin
Password
••••••

Figure 2: Login page

3.Sign Up Page for New Users

The New User Sign-Up page allows users to register by entering their personal details, setting up a username and password, and receiving an account number, enabling them to become an existing user



Full Name
Sachin N Holla

Email address
sachinholla1@gmail.com

Password
••••••

Date of Birth
03/22/2001

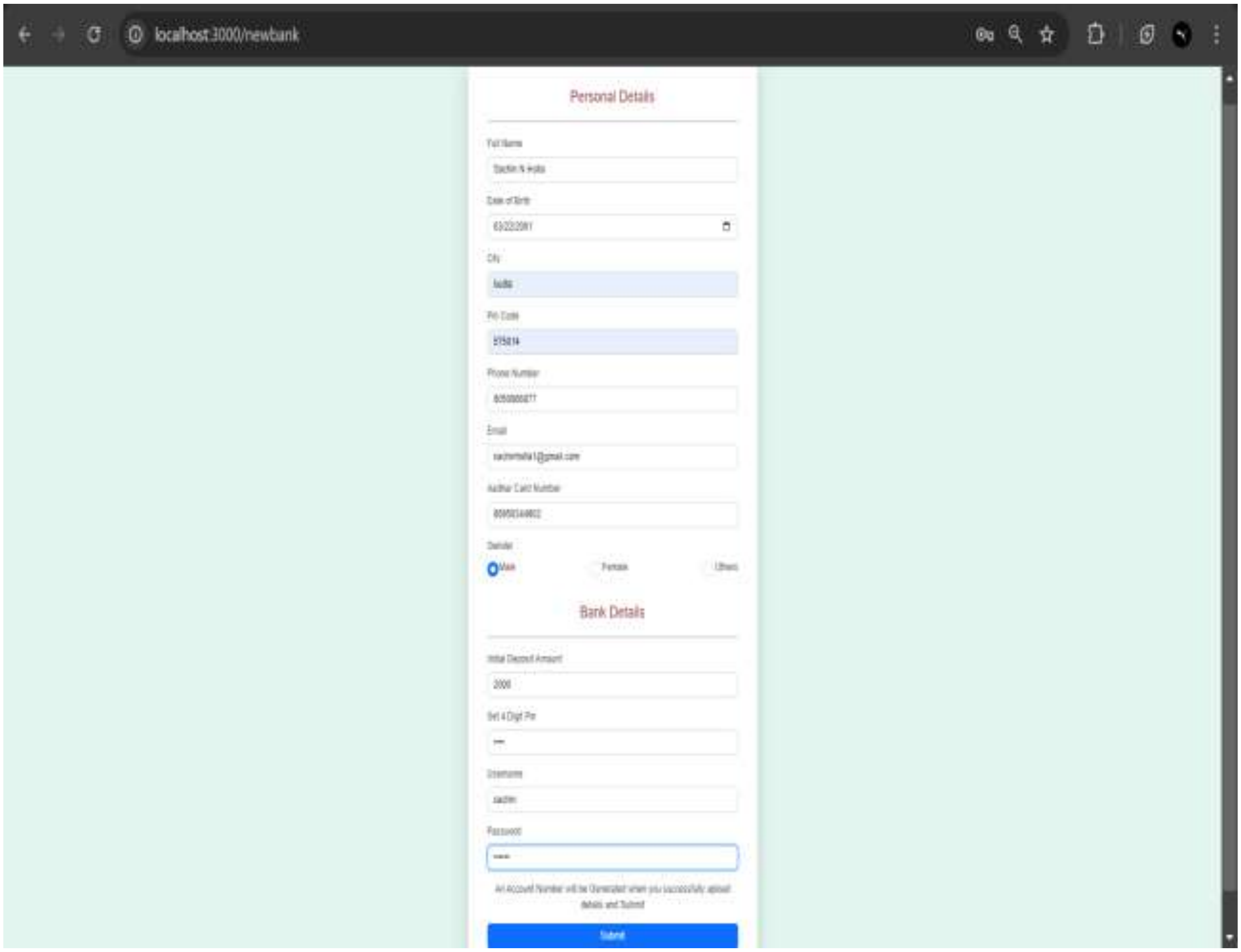
Contact
8050866077

Residential Address
surathkal.mangalore.575014

Figure 3: Sign Up Page

4.New Bank Registration

After a successful login, users are guided to the Bank Registration page, where users can add personal details to login as a new bank user. Upon completion, the system provides an account number, allowing users to set their username and password for future access.



Personal Details

Full Name
Sachin N Hotla

Date of Birth
03/22/2001

City
[illegible]

Pin Code
575014

Phone Number
[illegible]

Email
sachinhotla1@gmail.com

Aadhar Card Number
[illegible]

Gender
● Male ○ Female ○ Others

Bank Details

Initial Deposit Amount
2000

Set 4 Digit Pin
••••

Username
sachin

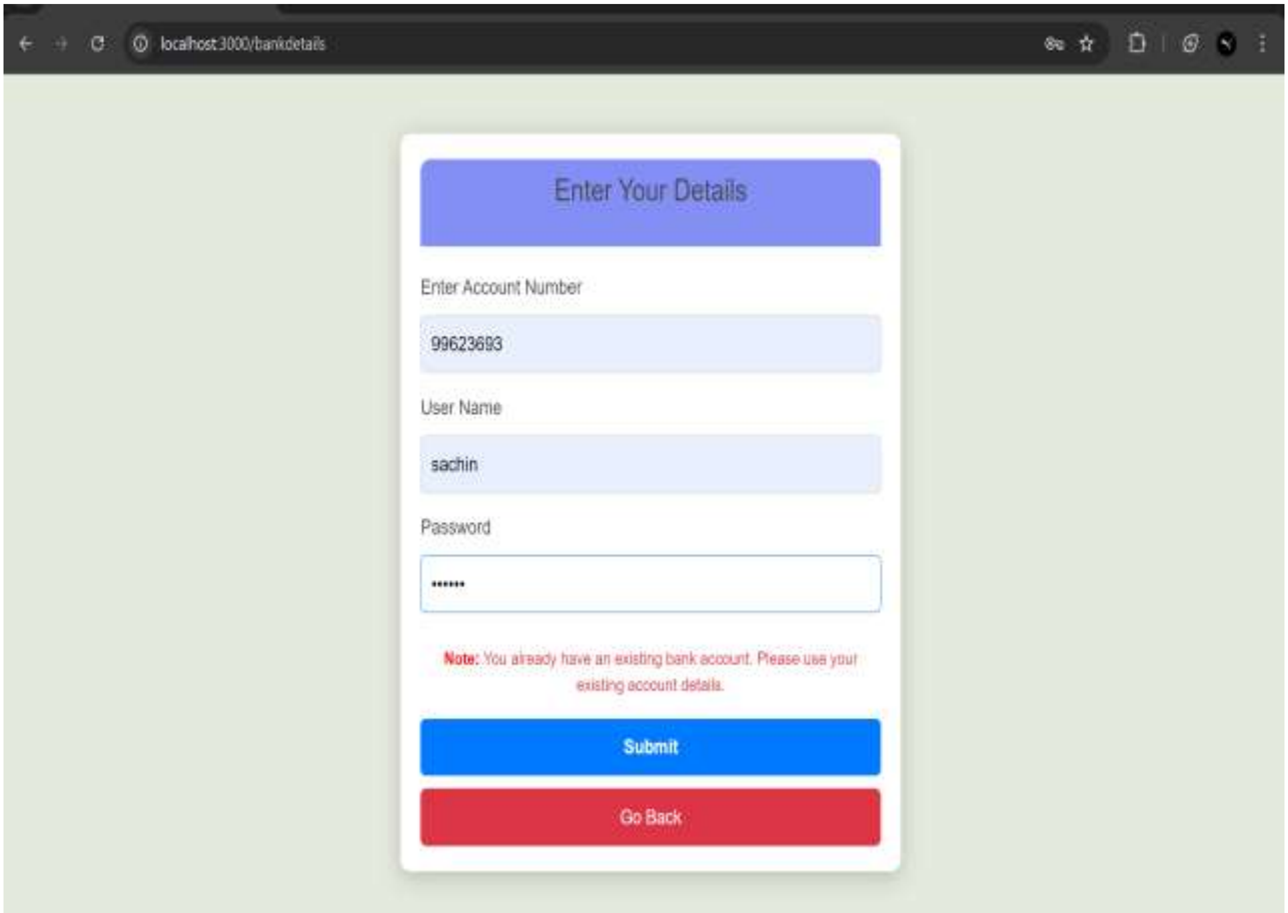
Password
••••

An Account Number will be Generated when you successfully upload details and Submit

Figure 4:Bank Registration page

5.Login Using Credentials

After successfully registering as a new bank user, you can log into the dashboard using your existing credentials. To log in, enter your registered username, password, and account number on the login page. Once authenticated, you'll be redirected to dashboard, and you can manage your account, view transaction details, and perform other banking activities.



Enter Your Details

Enter Account Number

99623693

User Name

sachin

Password

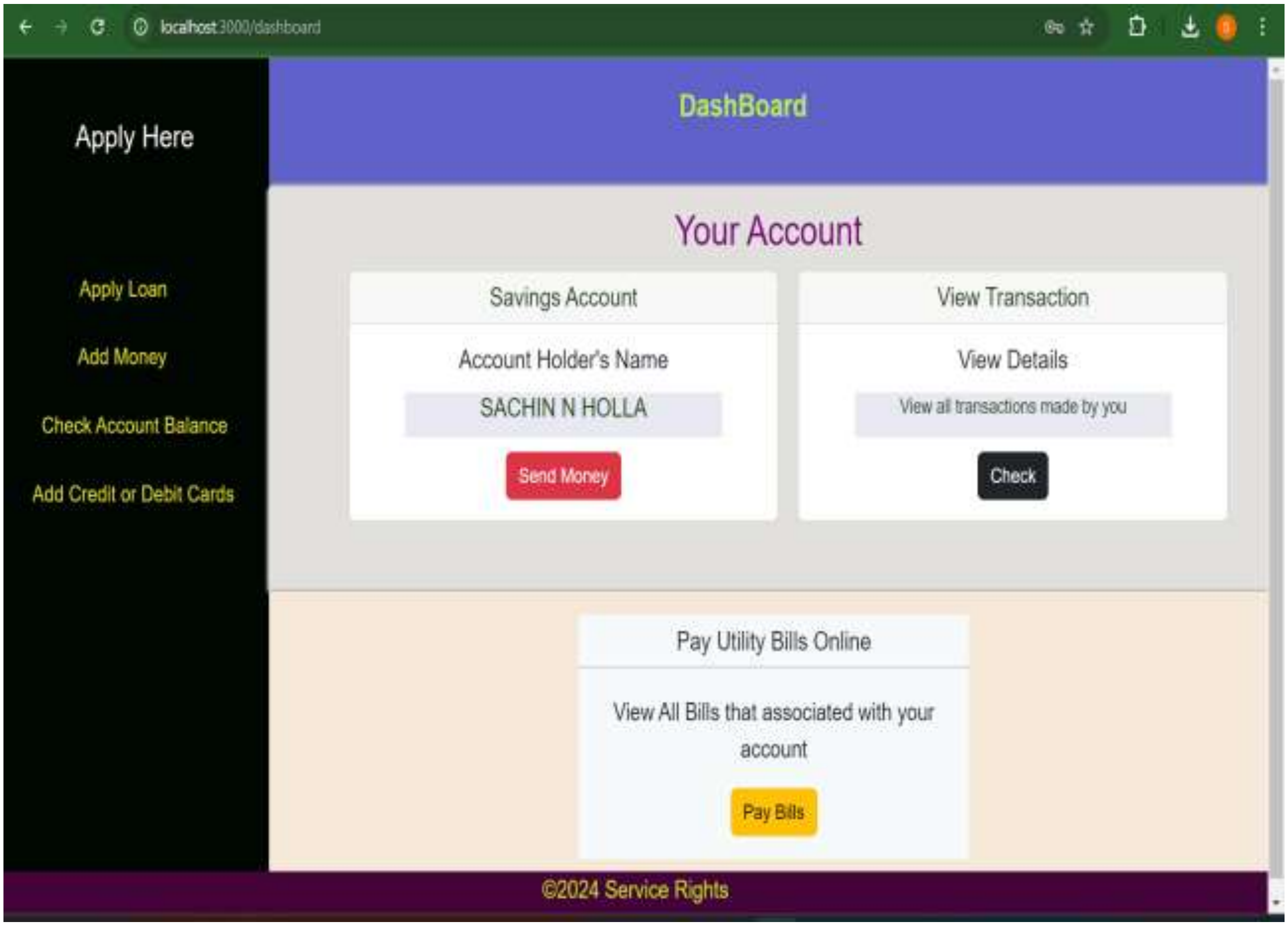
••••••

Note: You already have an existing bank account. Please use your existing account details.

Figure 5: Existing Account Login

6.Access DashBoard

After logging in, you'll be taken to the dashboard, which serves as the main center for handling your banking activities. The dashboard offers a user-friendly interface, giving you a quick overview of your account balance, recent transactions, and other financial details. From this central location, you can easily access features like money transfers, bill payments, and loan applications. The dashboard is crafted for simplicity, ensuring that all key functions are readily available for effective financial management.



Your Account

Savings Account

Account Holder's Name

SACHIN N HOLLA

View Transaction

View Details

View all transactions made by you

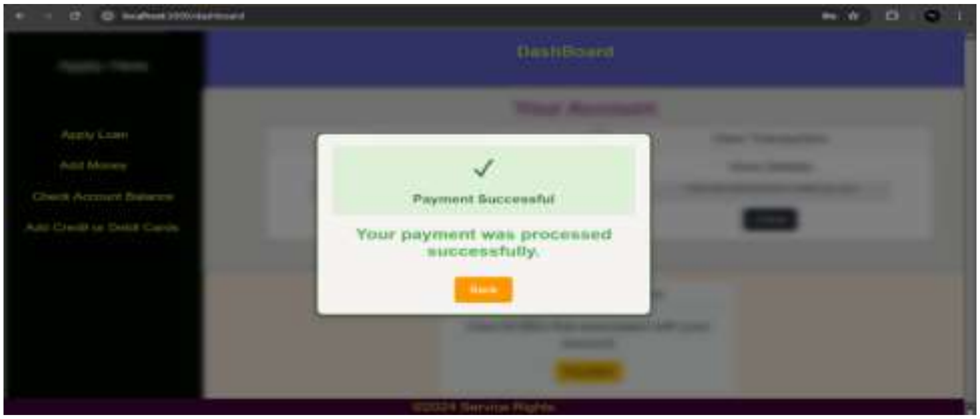
Pay Utility Bills Online

View All Bills that associated with your account

Figure 6: Dashboard

7.Add Money

In the dashboard, when you click the "Add Money" button, you will be prompted to enter your correct pin number along with the amount you wish to add to your account. After submitting this information accurately, the transaction will be processed successfully, and the amount will be added to your balance.



✓

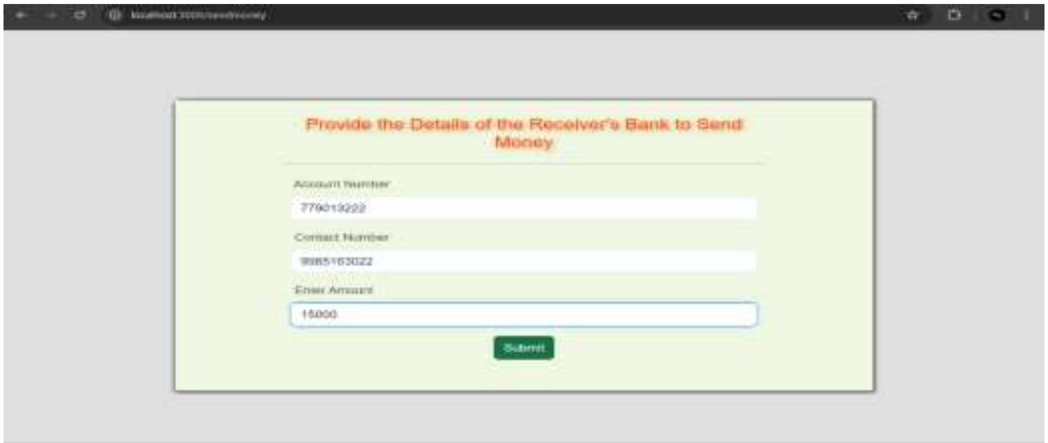
Payment Successful

Your payment was processed successfully.

Figure 7: Add money

8.Send Money to Others

In the dashboard, you can send money to others by entering the receiver's account number or choosing the mobile number transaction option. Simply provide the required details, and the money will be transferred securely to the recipient's account.



Provide the Details of the Receiver's Bank to Send Money

Account Number
779013222

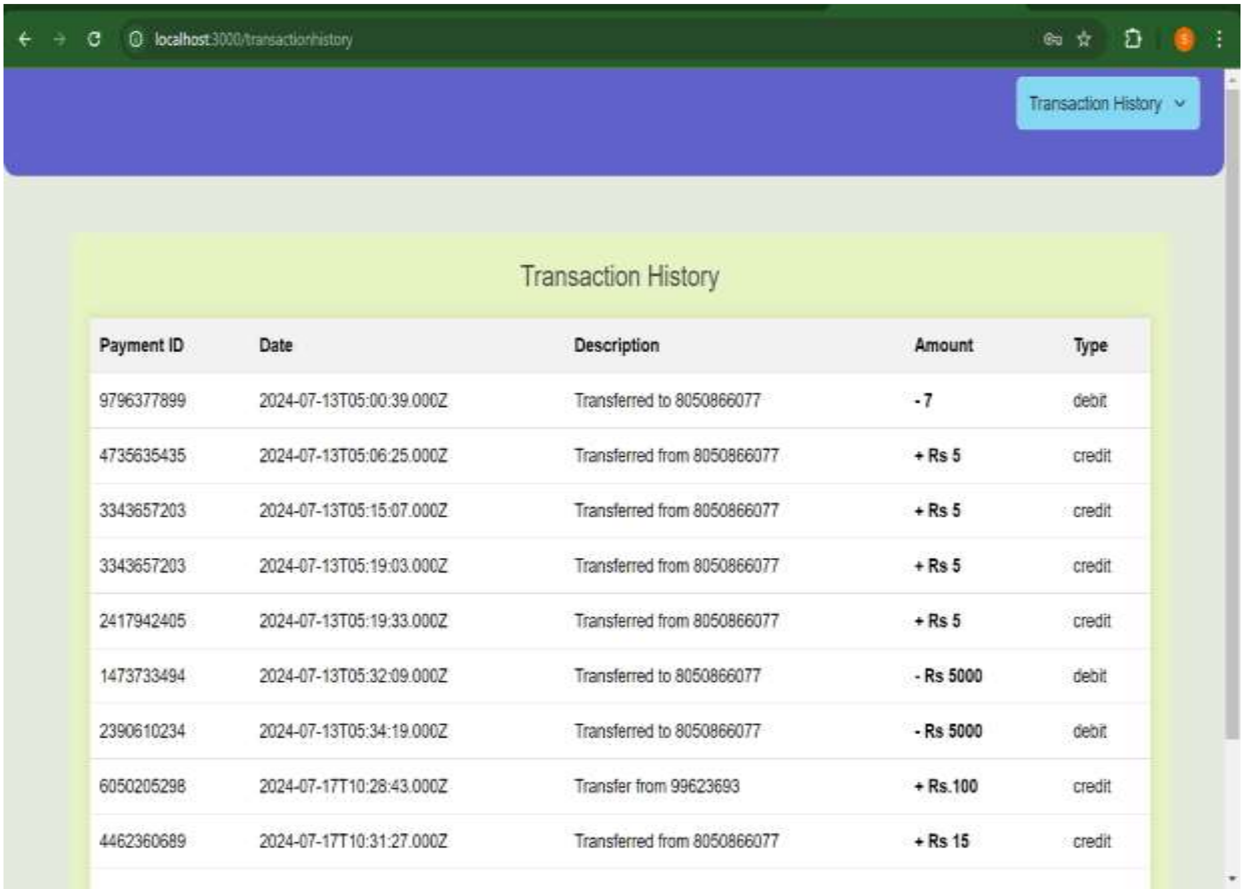
Contact Number
9985163022

Enter Amount
15000

Figure 8: Send money Page

9.Transaction History

The transaction history feature provides a clear and concise record of all your previous transactions, including dates, amounts, and types of transactions, making it easy to track and review your financial activities.



Transaction History

Payment ID	Date	Description	Amount	Type
9796377899	2024-07-13T05:00:39.000Z	Transferred to 8050866077	**- 7**	debit
4735635435	2024-07-13T05:06:25.000Z	Transferred from 8050866077	**+ Rs 5**	credit
3343657203	2024-07-13T05:15:07.000Z	Transferred from 8050866077	**+ Rs 5**	credit
3343657203	2024-07-13T05:19:03.000Z	Transferred from 8050866077	**+ Rs 5**	credit
2417942405	2024-07-13T05:19:33.000Z	Transferred from 8050866077	**+ Rs 5**	credit
1473733494	2024-07-13T05:32:09.000Z	Transferred to 8050866077	**- Rs 5000**	debit
2390610234	2024-07-13T05:34:19.000Z	Transferred to 8050866077	**- Rs 5000**	debit
6050205298	2024-07-17T10:28:43.000Z	Transfer from 99623693	**+ Rs.100**	credit
4462360689	2024-07-17T10:31:27.000Z	Transferred from 8050866077	**+ Rs 15**	credit

Figure 9: Transaction History

10.Pay Utility Bills

The utility bills feature enables you to conveniently pay your bills directly through the application, ensuring timely payments and easy management of recurring expenses. This feature simplifies the process by allowing you to handle multiple bills from a single platform.

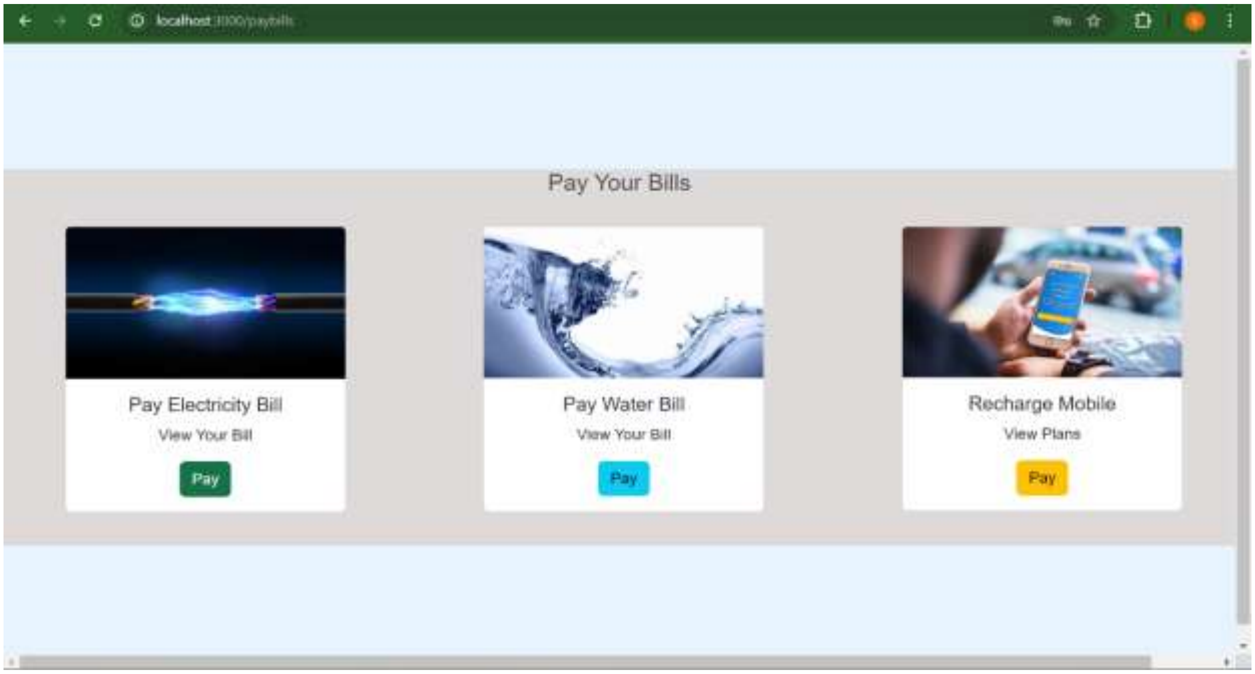
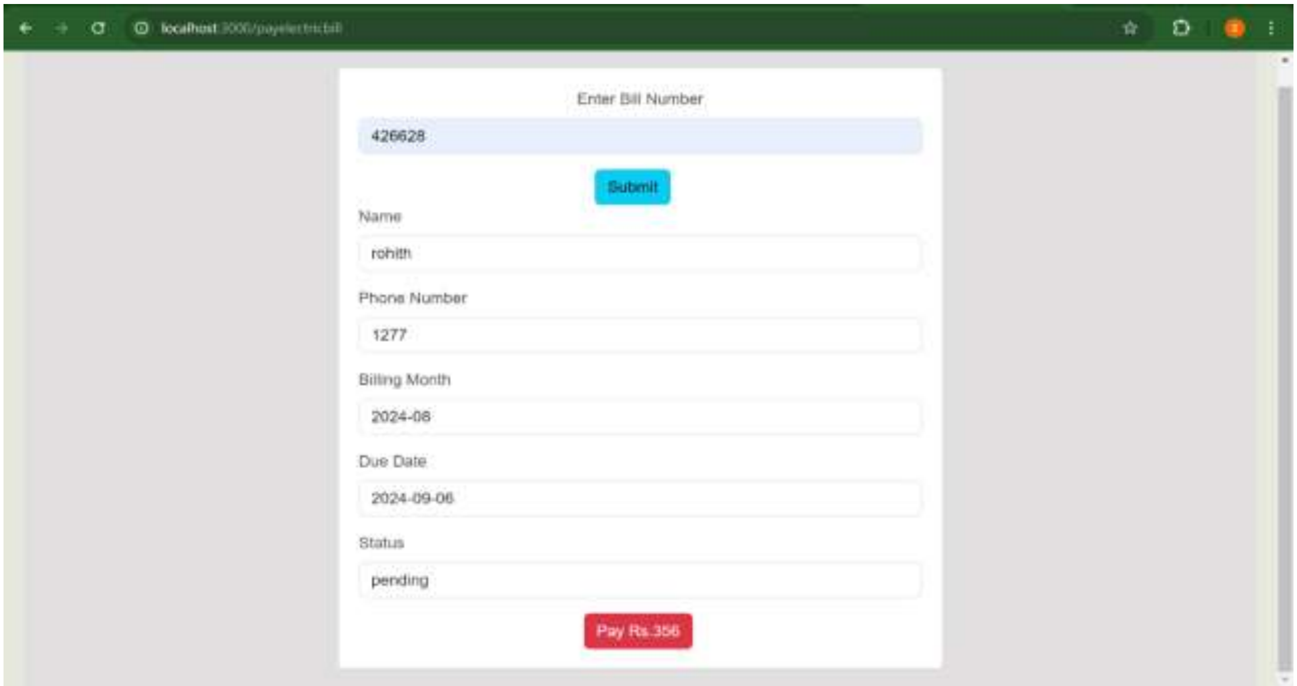


Figure 10: Utility Bills page

11.Pay Electricity Bill

For paying electricity bills, users can enter their bill number in the designated field. After entering the bill number, the relevant bill details will be displayed, allowing the user to review the information and proceed with the payment directly through the application.



localhost:3000/payelectricbill

Enter Bill Number

426628

Name

rohith

Phone Number

1277

Billing Month

2024-08

Due Date

2024-09-06

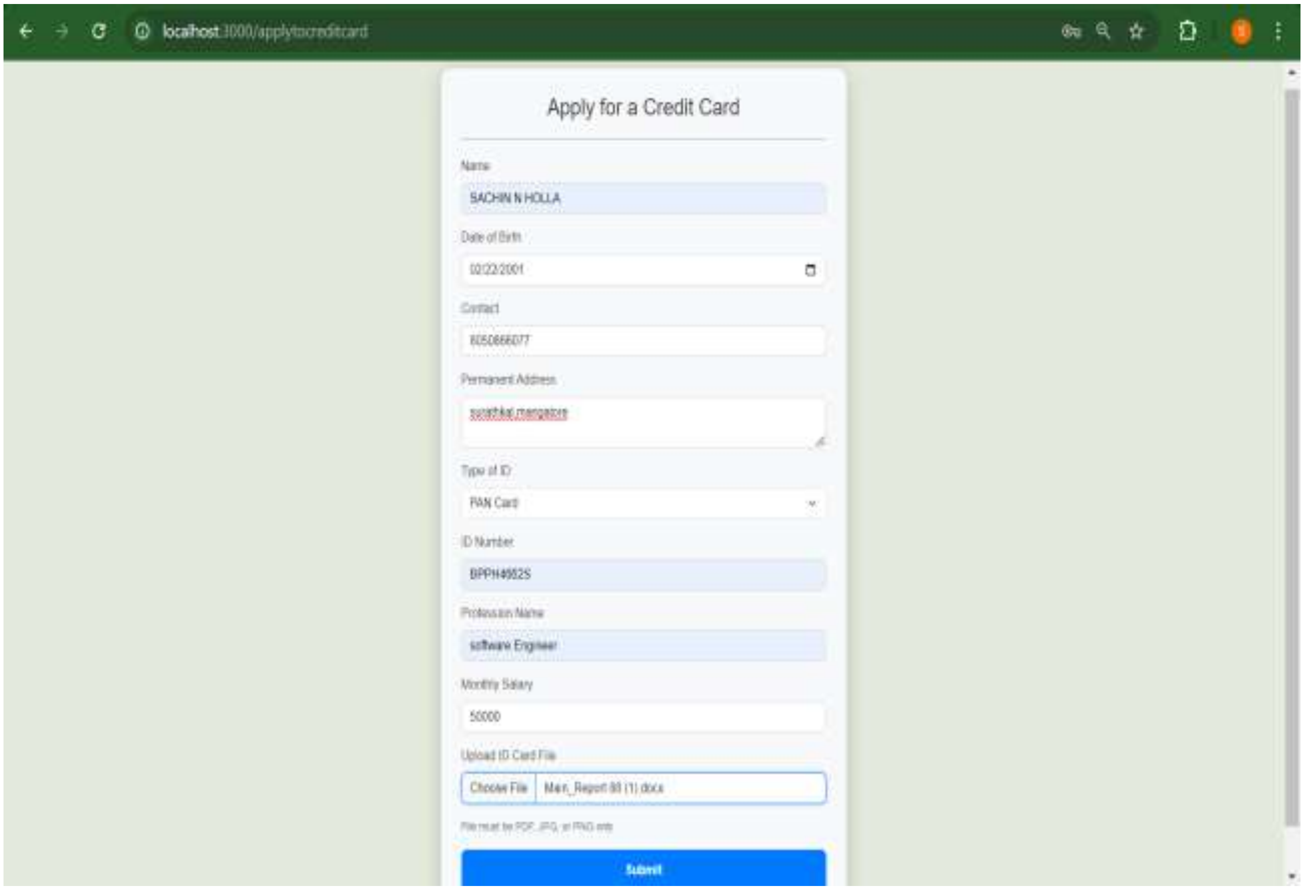
Status

pending

Figure 11:Electicity Bill Page

12.Apply Credit/Debit Card Application

End Users can apply for credit and debit cards directly through the application. They need to fill out the required details in the application form and submit it for processing. The submitted application will then be reviewed by the admin, who will either accept or reject it based on the provided information.



localhost:3000/applytocreditcard

Apply for a Credit Card

Name
SACHIN N HOLLA

Date of Birth
02/22/2001

Contact
8050866077

Permanent Address
surathkal,mangalore

Type of ID
PAN Card

ID Number
BPPH4652S

Profession Name
software Engineer

Monthly Salary
50000

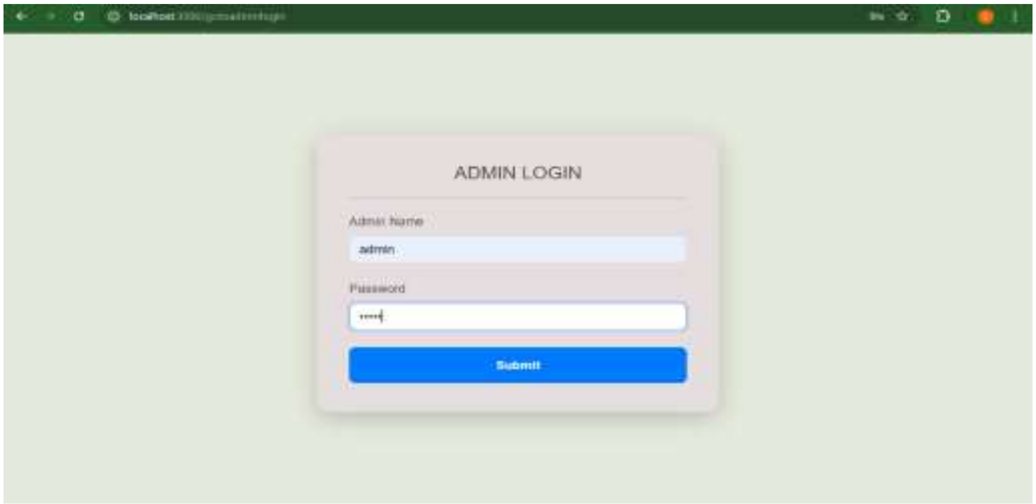
Upload ID Card File
Choose File Main_Report 80 (1).docx

File must be PDF, JPG, or PNG only

Figure 12: Apply Credit Card Application Page

13.Admin Login

The admin login page allows administrators to securely access the admin panel by entering their username and password. Upon successful login, admins can manage user applications, monitor transactions, and perform other administrative tasks within the FinEdge Application.



ADMIN LOGIN

Admin Name

admin

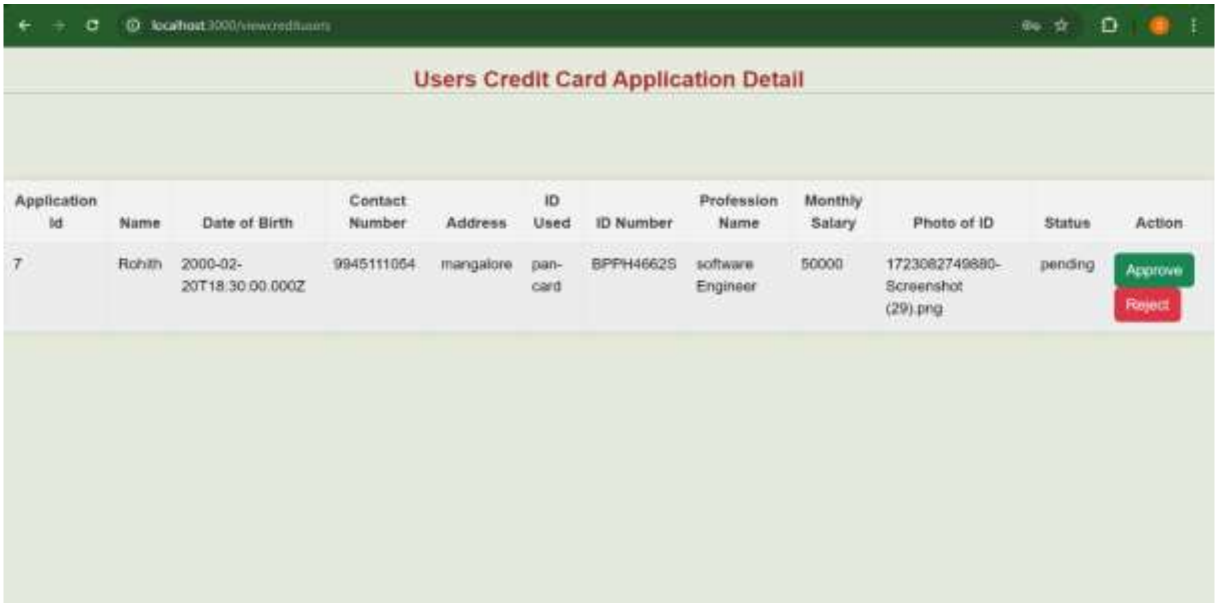
Password

•••••

Figure 13: Admin Login page

14.Review User Applications

Admins can review user applications by accessing the admin panel. They can view detailed information submitted by users, such as applications for credit/debit cards or loans, and decide to accept or reject them on basis of the provided details. This tells that all user requests are carefully evaluated before approval.



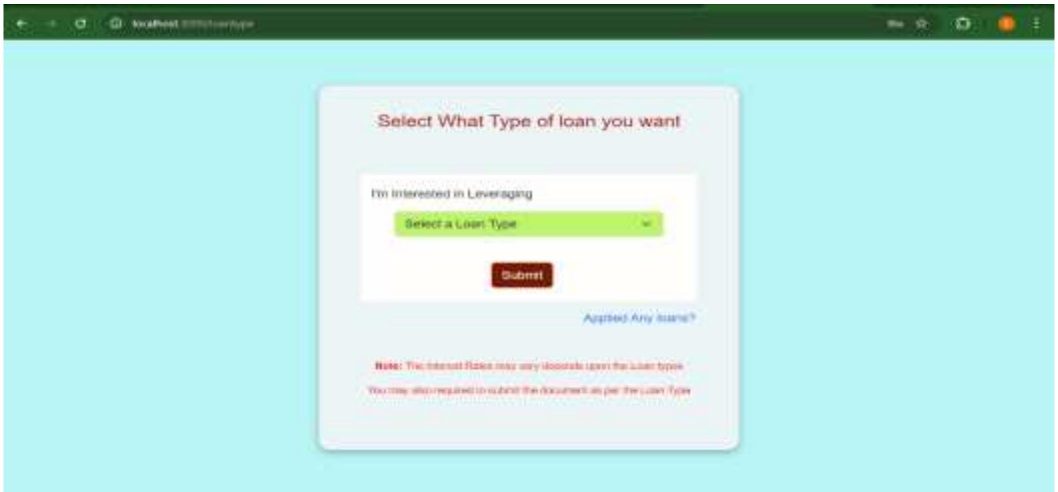
Users Credit Card Application Detail

Application Id	Name	Date of Birth	Contact Number	Address	ID Used	ID Number	Profession Name	Monthly Salary	Photo of ID	Status	Action
7	Rohith	2000-02-20T18:30:00.000Z	9945111064	mangalore	pan-card	BPPH4662S	software Engineer	50000	1723082749880-Screenshot (29).png	pending	Approve Reject

Figure 14:Review Credit Card Application Page

15.Apply Loan

Users can apply for loans by choosing the loan type and entering required financial details. The process is designed to allow quick submission and prompt review.



localhost:3000/loantype

Select What Type of loan you want

I'm Interested in Leveraging

Select a Loan Type

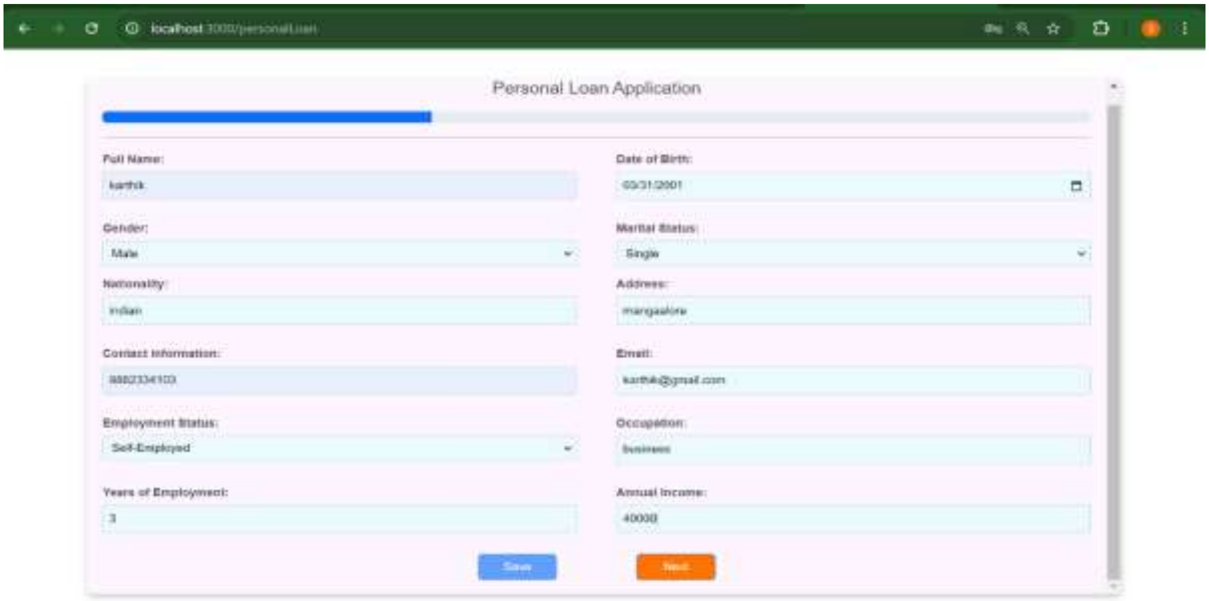
Note: The Interest Rates may vary depends upon the Loan types

You may also required to submit the document as per the Loan Type

Figure 15 : Apply Loan

16.Provide Loan Application Details

Users submit personal details when applying for a loan, ensuring accurate processing. This step includes filling in essential information for identity verification and eligibility assessment.



localhost:3000/personalLoan

Personal Loan Application

Full Name: karthik

Date of Birth: 03/31/2001

Gender: Male

Marital Status: Single

Nationality: indian

Address: mangaalore

Contact Information: 8882334103

Email: karthik@gmail.com

Employment Status: Self-Employed

Occupation: business

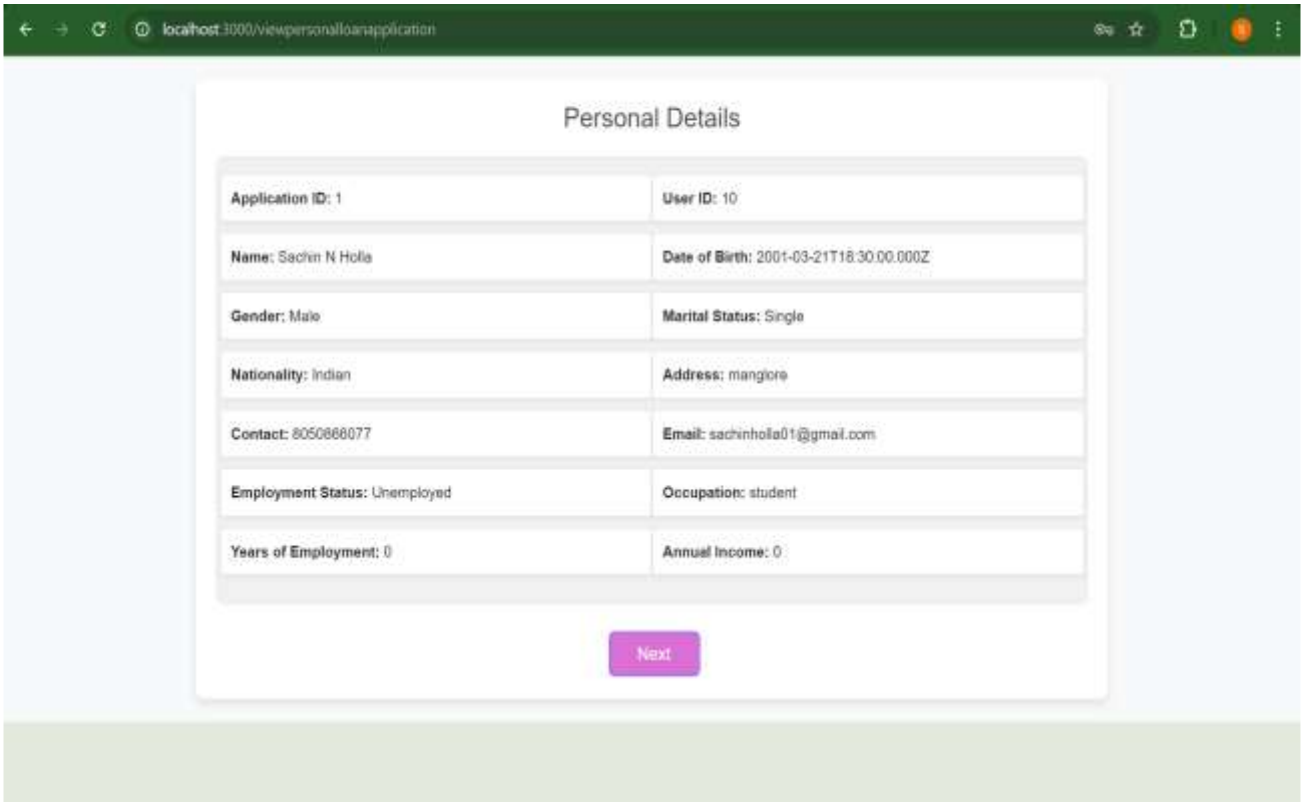
Years of Employment: 3

Annual Income: 40000

Figure 16:Provide Personal Details

17.Application Review By Administrator

Adminstrator review loan applications by evaluating the provided details and verifying the applicant's eligibility. They can then approve or reject the application based on the assessment.



Personal Details

Application ID: 1	**User ID:** 10
Name: Sachin N Holla	**Date of Birth:** 2001-03-21T18:30:00.000Z
Gender: Male	**Marital Status:** Single
Nationality: Indian	**Address:** manglore
Contact: 8050868077	**Email:** sachinholla01@gmail.com
Employment Status: Unemployed	**Occupation:** student
Years of Employment: 0	**Annual Income:** 0

Figure 17:Application Review By Administrator