VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“Jnana Sangama”, Belagavi-590018, Karnataka

Database Management Systems

Mini Project Report on

“BANK MANAGEMENT SYSTEM”

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

Bachelor of Engineering
in
Artificial Intelligence & Machine Learning

Submitted by

KUSHAL.T.R	**MANOJ.N.J**	**PRATHAM.P.N**
1BI22AI023	**1BI22AI026**	**1BI22AI038**

Under the Guidance of

Prof. Sahana M.
Assistant Professor
Department of AI&ML, BIT
Bengaluru-560 004

Department of Artificial Intelligence & Machine Learning

Bangalore Institute of Technology
K.R. Road, V.V. Pura, Bengaluru-560 004
2023-24

INDEX

Abstract i

1. Introduction 1
2. System Analysis 2
   - 2.1 Proposed System 2
   - 2.2 Objective of Proposed System 3
3. System Requirement Specification 4
   - 3.1 Hardware Requirement 4
   - 3.2 Software Requirement 4
4. System Design 7
   - 4.1 Database Design 7
     - 4.1.1 Schema Diagram 8
     - 4.1.2 E R Diagram 9
     - 4.1.3 Data-flow Diagram 10
   - 4.2 Implementation 11
     - 4.2.1 Modules Description 11
5. Result with Screen-shots 13
6. Conclusion and future enhancement 19

References

TABLE OF FIGURE

4.1 Schema Diagram 8

4.2 ER Diagram 9

4.3 Data Flow Diagram 10

5.1 Login page 13

5.2 User dashboard 13

5.3 Account Summary 14

5.4 Transfer money page 14

5.5 Check Notification page 15

5.6 Contact Us page 15

5.7 Manager dashboard 16

5.8 New Account Form page 16

5.9 Feedback page 17

5.10 Send Notice page 17

5.11 Staff Account page 17

5.12 Cashier dashboard 18

5.13 After entering account number 18

5.14 After transaction (withdraw or deposit) 18

Abstract

The Bank Management System is a comprehensive database management system (DBMS) designed to streamline and automate the various processes involved in managing a bank's operations. The primary objective of this project is to create a robust, scalable, and secure database system that enhances the efficiency and accuracy of banking transactions, customer management, account handling, and financial reporting.

The system leverages the capabilities of modern DBMS technologies to manage large volumes of data and support complex queries and transactions. Key features include customer account management, transaction processing, loan management, and support for multiple branches. The system also incorporates security measures to protect sensitive financial data and ensure compliance with regulatory requirements.

Customer account management encompasses the creation, updating, and deletion of customer records, alongside handling various account types such as savings, current, and fixed deposit accounts. Transaction processing involves recording deposits, withdrawals, fund transfers, and tracking account balances in real-time.

By implementing this Bank Management System, banks can achieve greater operational efficiency, reduce manual errors, and provide better customer service. The system's design prioritizes data integrity, security, and user-friendliness, making it an essential tool for modern banking operations.

Chapter 1

Introduction

In the contemporary financial landscape, banks play a crucial role in the economic stability and growth of societies. Efficiently managing a bank's operations, including customer accounts, transactions, loans, and financial reporting, is essential to maintaining trust and reliability. Traditional banking processes often involve a significant amount of manual work, which can lead to inefficiencies, errors, and increased operational costs. To address these challenges, the development of a comprehensive Bank Management System (BMS) using a Database Management System (DBMS) has become imperative.

The Bank Management System is designed to automate and streamline the various banking processes, ensuring that data is managed efficiently and accurately. This system utilizes the power of modern DBMS technologies to handle large volumes of data, support complex queries, and ensure the integrity and security of sensitive financial information.

This project focuses on creating a robust and scalable database system that can handle the diverse needs of a bank. Key functionalities of the system include customer account management, transaction processing, loan administration, and financial reporting. By integrating these functionalities into a cohesive system, banks can improve their operational efficiency, reduce manual errors, and enhance the overall customer experience.

The system is designed with a strong emphasis on data security and regulatory compliance, ensuring that sensitive financial information is protected against unauthorized access and breaches. Additionally, the system supports multi-branch operations, allowing customers to access banking services seamlessly across different locations.

In summary, the Bank Management System aims to revolutionize traditional banking operations by leveraging advanced DBMS technologies to create a secure, efficient, and user-friendly platform. This system not only enhances the bank's operational capabilities but also provides a solid foundation for future growth and innovation in the banking sector.

Chapter 2

System Analysis

The system analysis phase is critical in the development of the Bank Management System (BMS). It involves understanding the requirements, defining the system's functionalities, identifying potential challenges, and designing a system architecture that meets the bank's needs. This phase ensures that the final system is robust, efficient, and capable of handling the complexities of modern banking operations.

2.1 Proposed System

The proposed Bank Management System (BMS) is a comprehensive and efficient solution designed to streamline and automate core banking operations using advanced Database Management System (DBMS) technologies. It aims to handle large volumes of data, support complex transactions, and ensure data integrity and security. Key features of the system include customer account management, transaction processing, loan management, financial reporting, and multi-branch support. Customer account management involves creating, updating, and deleting customer records, and handling various account types such as savings, current, and fixed deposits. Transaction processing ensures real-time recording of deposits, withdrawals, and fund transfers, with automated reconciliation for accuracy.

The system's loan management functionality covers the entire loan lifecycle, from application and approval to disbursement and repayment tracking. Financial reporting tools enable the generation of various reports, including balance sheets, income statements, and transaction summaries, with customizable options to meet specific requirements. Multi-branch support allows seamless inter-branch transactions and centralized data management to ensure consistency across all branches. User authentication and authorization are implemented through role-based access control and strong authentication mechanisms like multi-factor authentication.

The system architecture follows a client-server model, with a user-friendly client interface, an application server for business logic, and a centralized database server for data management. The relational database design includes normalized tables for key entities like Customers, Accounts, Transactions, Loans, and Branches, with indexing for enhanced query performance. Security measures encompass data encryption both at rest and in transit, detailed audit trails for user activities, and robust backup and recovery strategies to prevent data loss and ensure business continuity.

The implementation strategy involves a phased approach, starting with a pilot phase in a single branch to test the system, followed by a full rollout based on feedback and improvements. Comprehensive training programs and detailed user manuals will support bank staff in adopting the new system, while a dedicated technical support team and regular system maintenance will ensure ongoing smooth operation. Overall, the proposed BMS aims to revolutionize banking operations by enhancing efficiency, reducing errors, and improving customer service, providing a robust foundation for modern banking needs.

2.2 Objective of Proposed System

The primary objective of the proposed Bank Management System (BMS) is to develop a comprehensive, efficient, and secure platform that automates and streamlines the core banking operations. This system aims to improve operational efficiency, enhance data accuracy, and provide superior customer service. The specific objectives include:

Streamline Banking Operations: Automate routine banking processes such as customer account management, transaction processing, and loan management to reduce manual effort, minimize errors, and increase efficiency.

Enhance Data Management: Utilize advanced DBMS technologies to handle large volumes of data, support complex queries, and ensure data integrity and consistency across all branches.

Improve Customer Service: Provide a user-friendly interface and real-time transaction processing to enhance the customer experience, enabling customers to manage their accounts, conduct transactions, and access banking services seamlessly.

Ensure Data Security: Implement robust security measures including data encryption, access control, and audit trails to protect sensitive financial information from unauthorized access and breaches.

Support Multi-Branch Operations: Facilitate seamless operations and data synchronization across multiple bank branches, enabling inter-branch transactions and centralized data management.

Scalability and Reliability: Design a scalable and reliable system architecture to accommodate future growth in the number of customers and transactions, ensuring minimal downtime and robust backup and recovery mechanisms

Compliance with Regulations: Ensure the system adheres to all relevant financial regulations and standards to maintain compliance and avoid legal issues.

By achieving these objectives, the proposed BMS will revolutionize traditional banking operations, fostering greater efficiency, security, and customer satisfaction, and positioning the bank for future growth and innovation.

.

Chapter 3

System Requirement Specification

The successful deployment and operation of the Bank Management Database System (BMDS) hinge upon a clear understanding of the hardware, software, and infrastructure requirements necessary to support its functionalities and performance objectives. Through a meticulous process of requirement elicitation, documentation, and validation, stakeholders can ensure that the system meets their needs and expectations effectively.

3.1 Hardware Requirement

The hardware requirements define the necessary physical components to support the functionality and performance of the Loan Management Database System. These include:

- **Server:** A dedicated or cloud-based server with a minimum of 4 cores, 8 GB RAM, and sufficient storage capacity for database management and application hosting.
- **Storage:** At least 100 GB of SSD storage for storing the database and system-related files.
- **Network:** High-speed internet connectivity to ensure seamless communication between users and the server.
- **User Devices:** Support for various user devices, including computers, tablets, and smartphones. The system should be compatible with a wide range of browsers.

3.2 Software Requirement

The software requirements detail the technologies, frameworks, and tools necessary for the development, deployment, and operation of the Bank Management Database System. These include:

1. **PHP (Hypertext Preprocessor):** The core server-side scripting language for dynamic content and database interactions.
2. **MySQL Database:** A robust relational database management system storing crucial data related to loan lending, interest, and loans pending.

3. **Bootstrap:** Used for responsive and visually appealing user interface design, providing a grid system and pre-designed components.

4. **JavaScript:** Enhances interactivity, enabling client-side validation, dynamic updates, and an improved user experience.

5. **HTML (Hypertext Markup Language):** Structures web page content, defining the layout and presentation of system elements.

6. **Ajax (Asynchronous JavaScript and XML):** Facilitates asynchronous communication, enabling real-time updates without page refresh.

7. **jQuery:** Simplifies DOM manipulation and event handling in JavaScript for concise and efficient client-side code.

8. **Browsers Compatibility:** Supports popular browsers such as Opera, Mozilla Firefox, Google Chrome, and IE8 for broad user accessibility.

9. **WAMP/XAMPP/LAMP/MAMP:** Deployable on various web server environments, providing the necessary infrastructure for hosting the PHP-based application.

Chapter 4

System Design

The system design for the proposed Bank Management System (BMS) focuses on creating a robust, scalable, and secure architecture to meet the functional and non-functional requirements of modern banking operations. The architecture follows a client-server model, separating the user interface from server-side processing and database management. The client interface, accessible via web or desktop applications, provides a user-friendly platform for bank staff and customers to perform various banking operations such as account management, transaction processing, and report generation. The application server handles business logic, processes client requests, and enforces business rules and workflows, ensuring efficient execution of banking operations. The database server uses a relational DBMS to manage data storage, retrieval, and manipulation, ensuring data integrity and consistency.

4.1 Database Design

Users of the Bank Management Database System (BMDS) can register and log in to the system to access a range of functionalities, including the ability to create account, view account status, and make payments. The system provides a user-friendly interface that allows users to easily navigate through user account, view detailed information, and securely submit their applications.

The database design for the Bank Management Database System (BMDS) is crucial for its efficient operation. The design should consider the following entities:

User account: This entity stores information about individual id, name, account no, branch, email, password, address, contact, date.

Transaction: This entity stores information about transaction id, user id, balance, credit, debit, balance, date.

Login: This entity stores information about id, email, password, type, date.

Notice: This entity stores information about user id, text of notice, date.

Feedback: This entity stores information about user id, text of feedback, date.

Branch: This entity stores information about branch id, branch number, branch name.

Manager: This entity stores information about manager id, manager name, address, contact, email, password, date.

Cashier: This entity stores information about Cashier id, name, email, password, address, contact, date.

The database design should also consider data integrity, normalization, and indexing for efficient querying and reporting. It should be scalable to accommodate growth in loan volumes and adaptable to changing business requirements. Additionally, the design should ensure data security and compliance with regulatory requirements, including encryption, access controls, and audit logging.

4.1.1 Schema Diagram

The schema diagram provides a visual representation of the database schema, illustrating the tables, fields, relationships, and constraints that comprise the system's data model. It helps users and admin understand the logical structure of the database and how data entities are organized and related to each other.

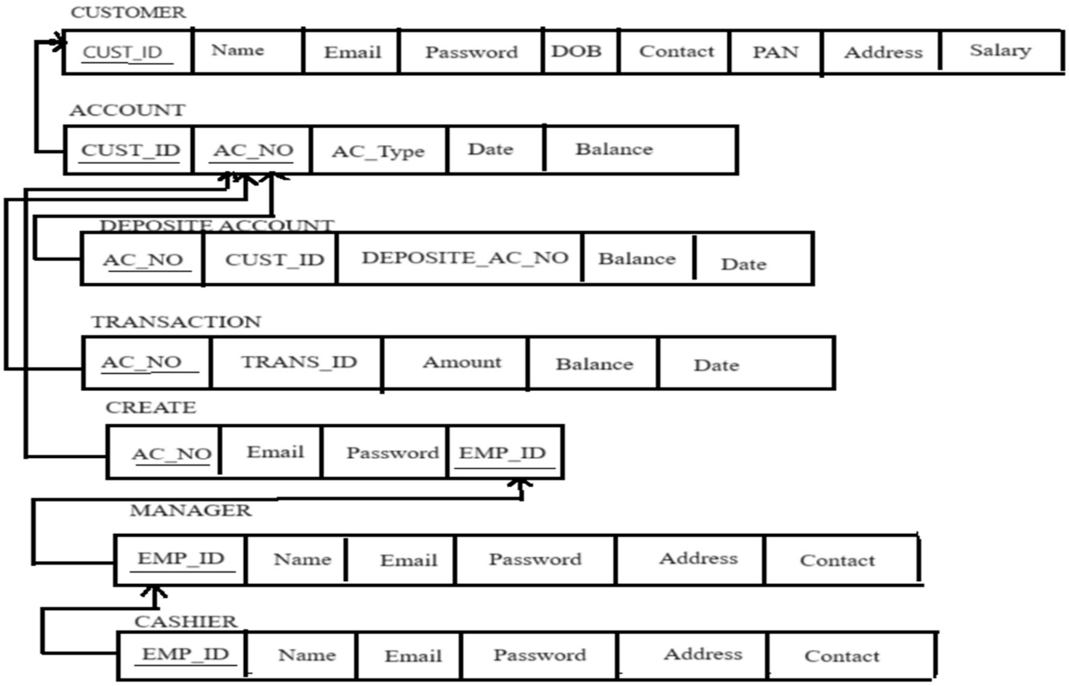

CUSTOMER
CUST_ID | Name | Email | Password | DOB | Contact | PAN | Address | Salary
ACCOUNT
CUST_ID | AC_NO | AC_Type | Date | Balance
DEPOSITE ACCOUNT
AC_NO | CUST_ID | DEPOSITE_AC_NO | Balance | Date
TRANSACTION
AC_NO | TRANS_ID | Amount | Balance | Date
CREATE
AC_NO | Email | Password | EMP_ID
MANAGER
EMP_ID | Name | Email | Password | Address | Contact
CASHIER
EMP_ID | Name | Email | Password | Address | Contact

Figure 4.1 – Schema Diagram

4.1.2 Entity-Relationship (ER) Diagram

The entity-relationship (ER) diagram depicts the entities, attributes, and relationships within the database schema, providing insights into the data model's conceptual design. It helps stakeholders visualize the entities and their relationships, identify dependencies, and refine the data model as needed.

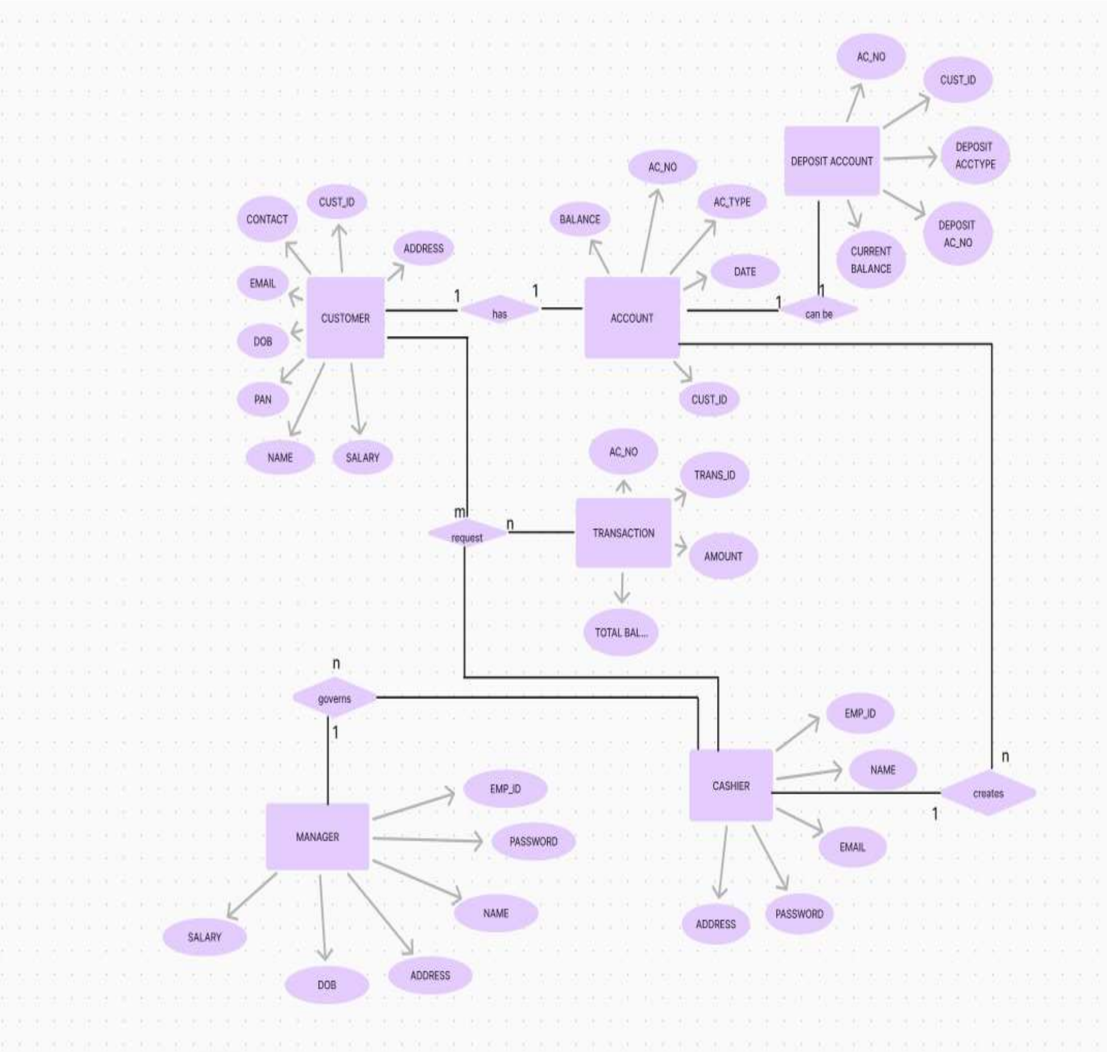


Figure 4.2 – ER Diagram

4.1.3 Data-flow Diagram

The data-flow diagram (DFD) illustrates the flow of data within the system, depicting how information moves between processes, data stores, and external entities. It provides a high-level view of the system's data processing and interaction mechanisms, helping stakeholders understand the data flow and identify potential bottlenecks or inefficiencies.

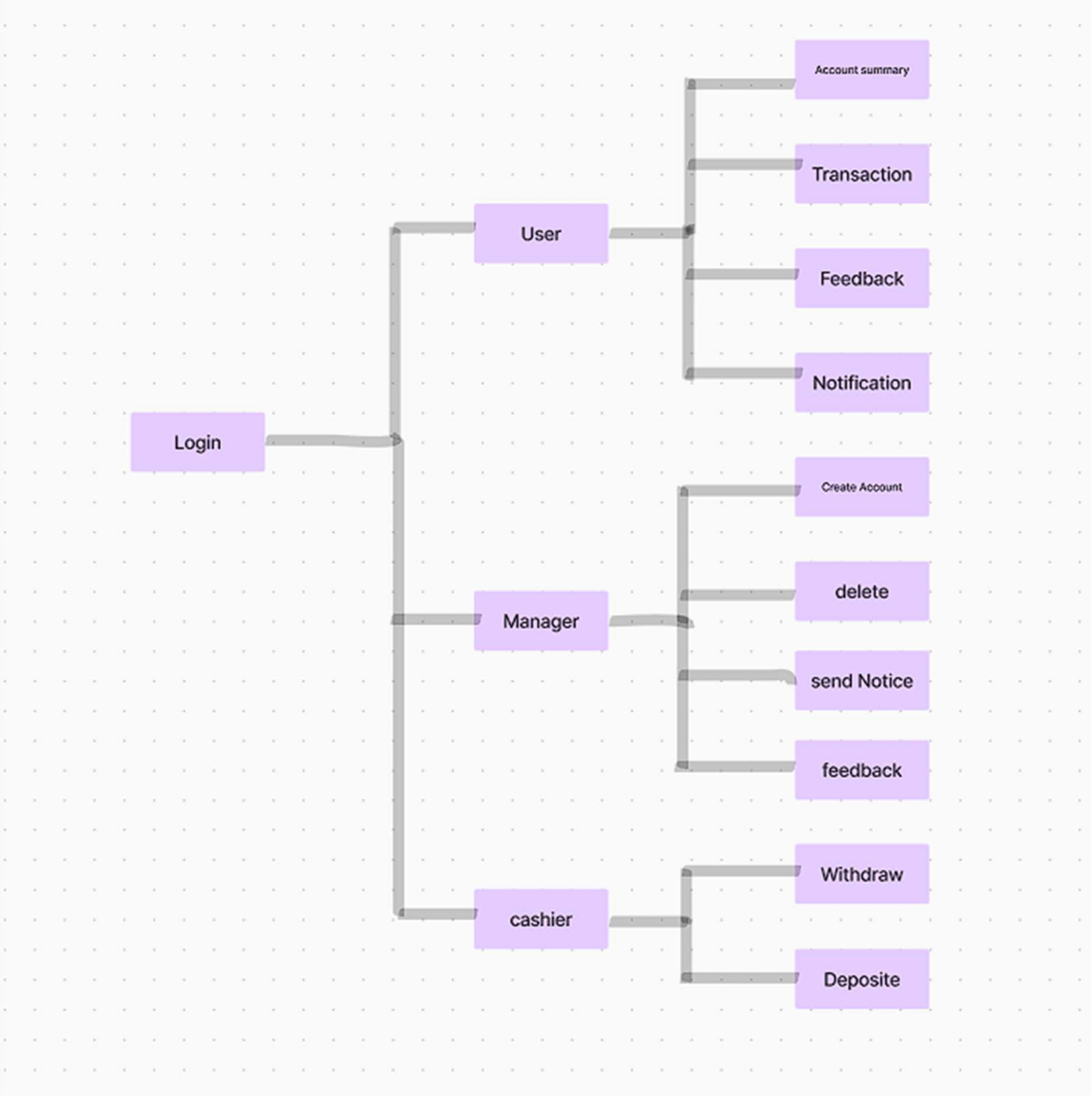


Figure 4.3 – Data-flow Diagram

4.2 Implementation

Implementing a Loan Management Database System involves several steps, and it requires a web development environment, knowledge of PHP, HTML, JavaScript, and MySQL. Below is a simplified guide to help you get started with the implementation. Note that this is a basic outline, and a complete implementation may involve additional considerations such as security, user authentication, and more.

4.2.1 Modules Description

Customer Account Management Module: The Customer Account Management Module is dedicated to handling all operations related to customer accounts, ensuring that the creation, management, and maintenance of account details are performed efficiently. This module enables bank staff to open new accounts by capturing necessary customer information such as name, address, contact details, and initial deposit. It also facilitates the updating of customer information and account settings, allowing for the maintenance of accurate and up-to-date records. Additionally, the module supports the process of closing accounts, ensuring that all necessary procedures are followed. With tools for viewing account details and transaction history, this module provides a comprehensive solution for managing customer accounts.

Transaction Processing Module: The Transaction Processing Module is responsible for managing all types of financial transactions within the bank, ensuring they are processed securely and efficiently. It handles customer deposits and withdrawals, updating account balances accordingly. The module supports the transfer of funds between accounts within the bank or to other banks, facilitating seamless transactions for customers. It maintains a detailed record of all transactions, including date, type, amount, and involved accounts, ensuring transparency and traceability. Additionally, automated reconciliation ensures that all transactions are accurately recorded, maintaining consistency and reliability in the bank's financial operations.

Multi-Branch Support Module: The Multi-Branch Support Module ensures seamless operations and data synchronization across multiple bank branches. It maintains a centralized database to ensure data consistency across all branches, allowing for accurate and unified record-keeping. The module facilitates transactions between different branches of the bank, supporting customers who need to perform inter-branch transactions. It also supports branch-specific operations while maintaining overall integration, enabling each branch to operate effectively within the larger banking network.

User Authentication and Authorization Module: The User Authentication and Authorization Module ensures that only authorized users can access the system and manages user permissions to maintain security and integrity. It assigns roles to users and restricts access based on these roles, ensuring that users can only access the functionalities relevant to their responsibilities. The module implements multi-factor authentication (MFA) for secure access, adding an extra layer of protection against unauthorized access. Detailed audit trails are kept to monitor user activities, providing transparency and enabling the detection of any unauthorized actions.

Customer Feedback Module: The Customer Feedback Module manages customer feedback, enabling the bank to collect and respond to customer inputs effectively. It allows customers to submit feedback and suggestions, providing a platform for communication and improvement. The module provides tools for bank staff to review and respond to customer feedback, ensuring that customer concerns are addressed promptly. Additionally, it generates reports on customer feedback for analysis, helping the bank identify areas for improvement and enhancing overall customer satisfaction.

Notice Management Module: The Notice Management Module handles the creation and distribution of notices to customers and bank staff. It allows bank staff to create notices regarding updates, events, or important information, ensuring that relevant parties are informed. The module distributes notices to the appropriate customers or staff members, facilitating efficient communication. It also keeps a record of all notices sent and their recipients, ensuring that communication is tracked and documented. This module enhances the bank's ability to communicate effectively with its customers and staff.

Chapter 5

Result with Screenshots

The login page takes the input of email and password, where the password is encrypted. After logging in the user, manager and cashier will be able to view and access the dashboard.

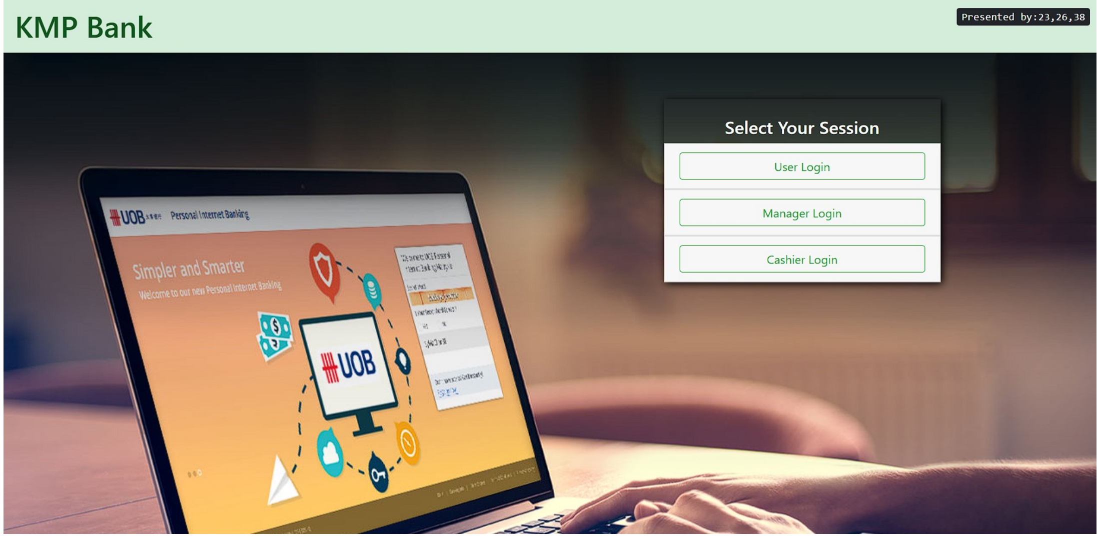


Figure 5.1 – Login page

After user login with his email and password dashboard will shown in figure 5.2

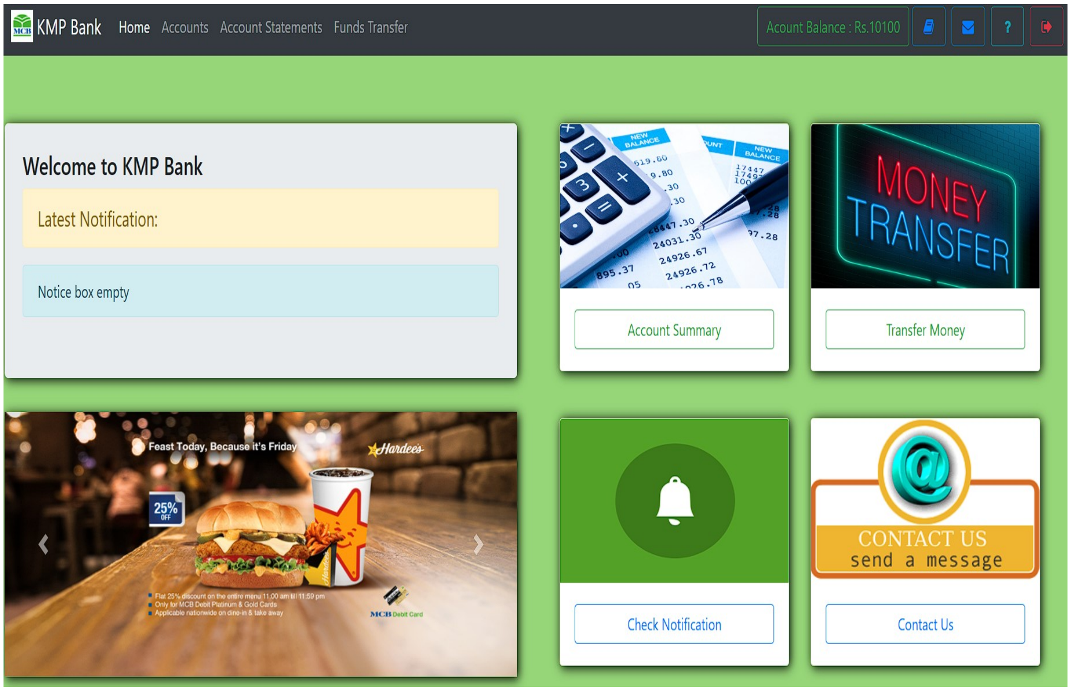
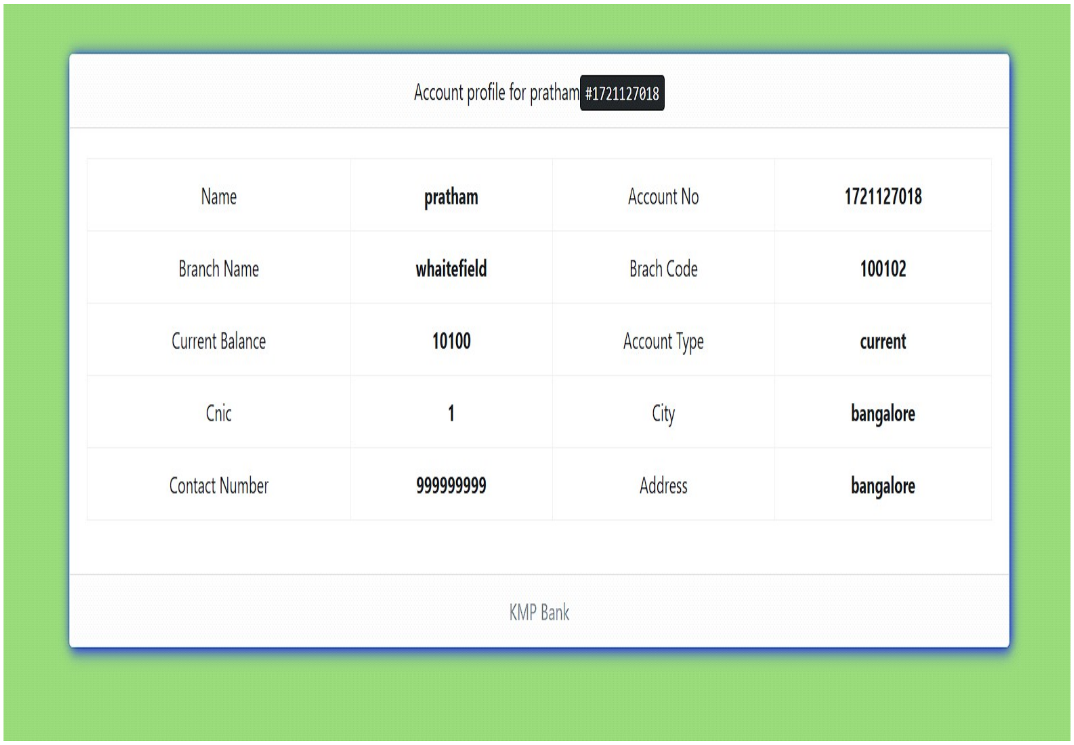


Figure 5.2 – User dashboard

User can check his account summer by clicking summer tab show in figure 5.3



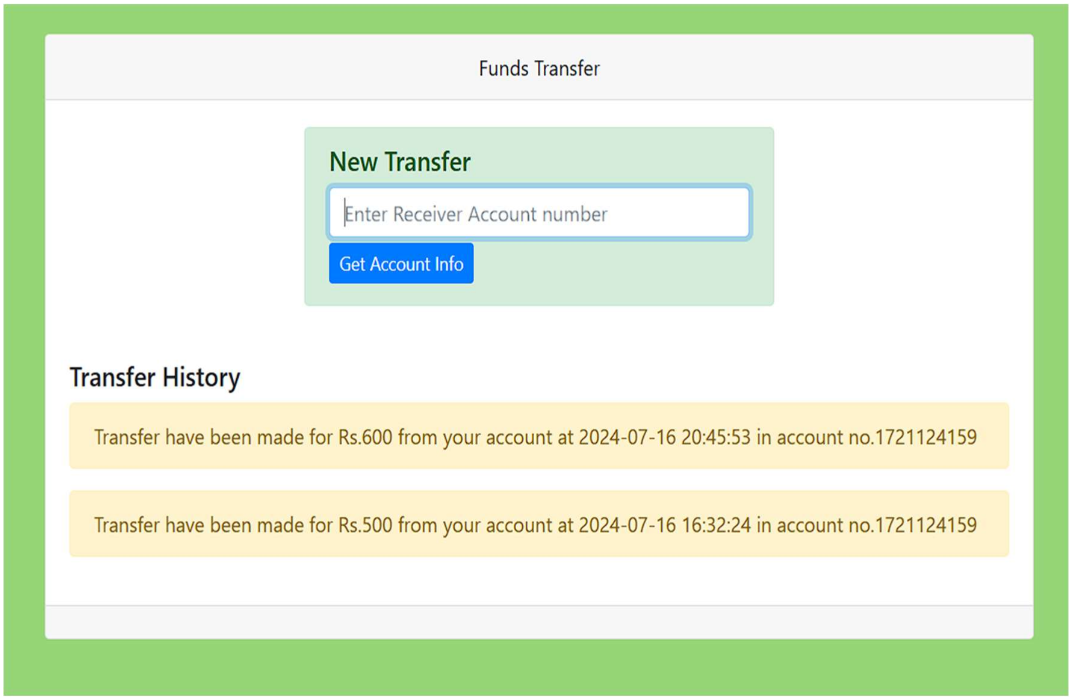
Account profile for pratham #1721127018

Name	**pratham**	Account No	**1721127018**
Branch Name	**whaitefield**	Brach Code	**100102**
Current Balance	**10100**	Account Type	**current**
Cnic	**1**	City	**bangalore**
Contact Number	**999999999**	Address	**bangalore**

KMP Bank

Figure 5.3 – Account Summary

User can transfer money from his account to another account shown figure 5.4



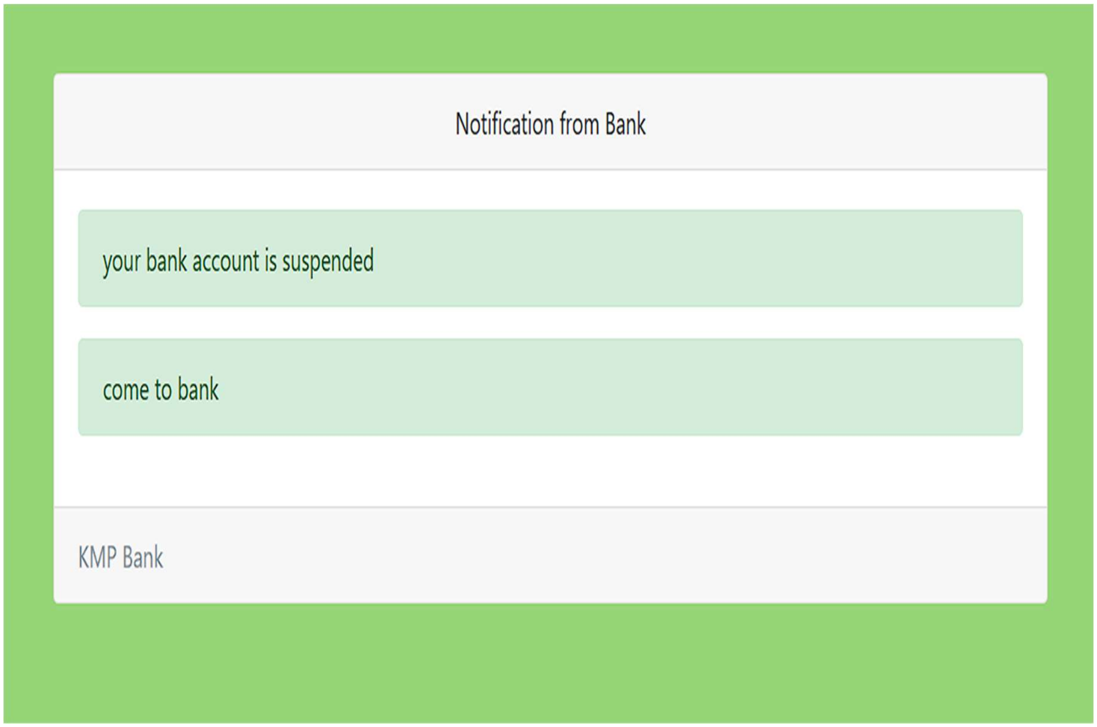
Transfer History

Transfer have been made for Rs.600 from your account at 2024-07-16 20:45:53 in account no.1721124159

Transfer have been made for Rs.500 from your account at 2024-07-16 16:32:24 in account no.1721124159

Figure 5.4 – Transfer money page

In Notification tab user can read any bank notifications shown in figure 5.5



Notification from Bank

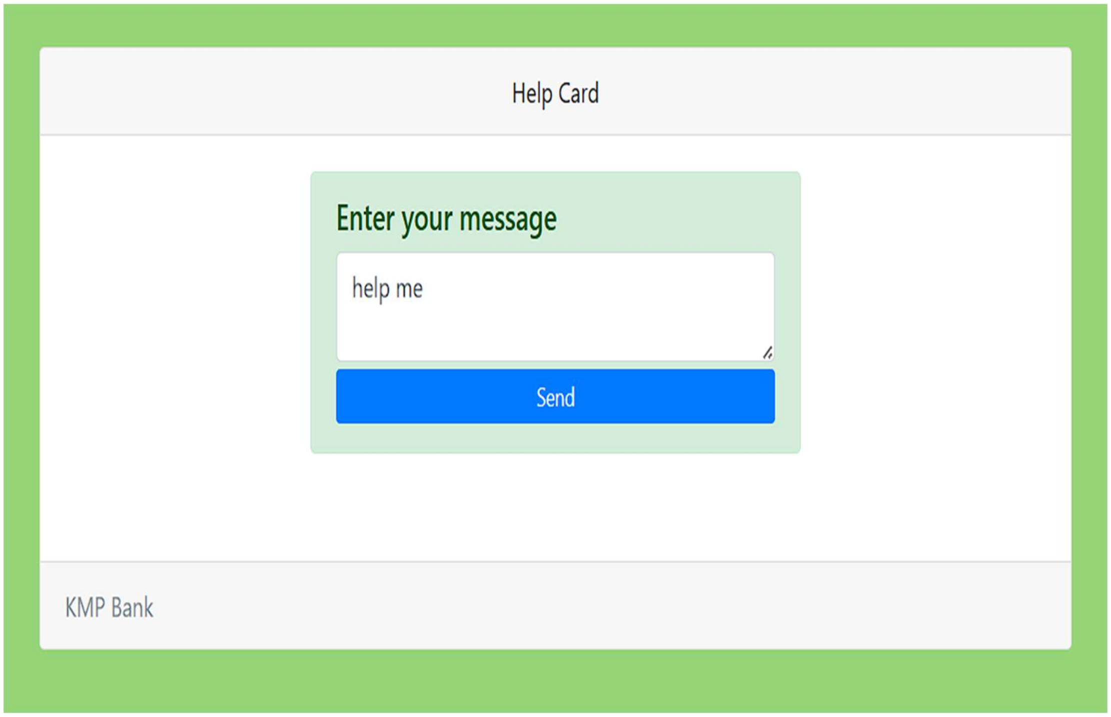
your bank account is suspended

come to bank

KMP Bank

Figure 5.5 – Check Notification page

If user need any help from bank he can send message in contact us tab shown in figure 5.6



Help Card

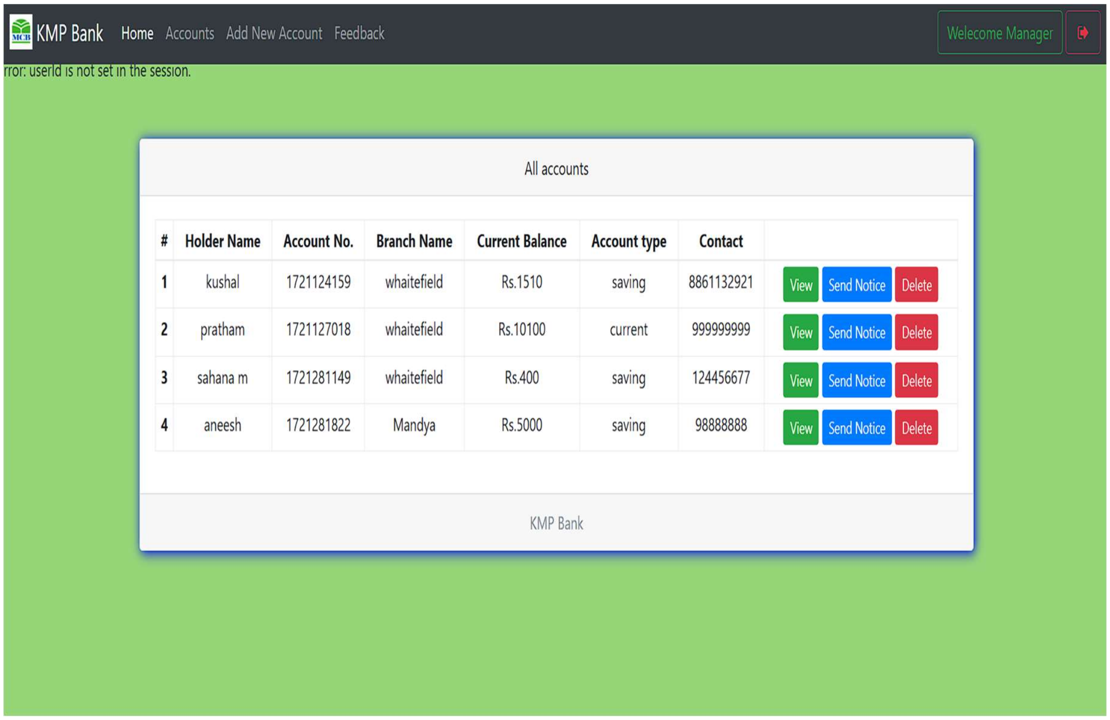
Enter your message

help me

KMP Bank

Figure 5.6 – Contact Us page

After Manager login with his email and password dashboard shown in figure 5.7



rror: userId is not set in the session.

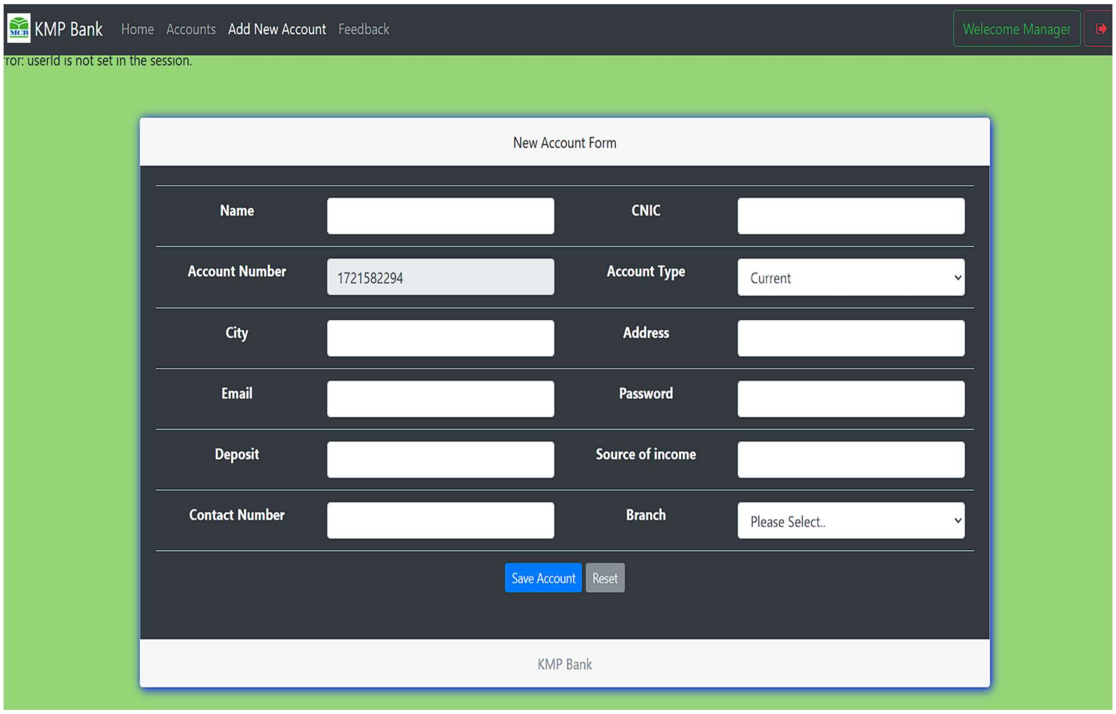
All accounts

#	Holder Name	Account No.	Branch Name	Current Balance	Account type	Contact	
1	kushal	1721124159	whaitefield	Rs.1510	saving	8861132921	View Send Notice Delete
2	pratham	1721127018	whaitefield	Rs.10100	current	999999999	View Send Notice Delete
3	sahana m	1721281149	whaitefield	Rs.400	saving	124456677	View Send Notice Delete
4	aneesh	1721281822	Mandya	Rs.5000	saving	98888888	View Send Notice Delete

KMP Bank

Figure 5.7 – Manager dashboard

Manager can add new user bank account by entering his details shown in figure 5.6



rror: userId is not set in the session.

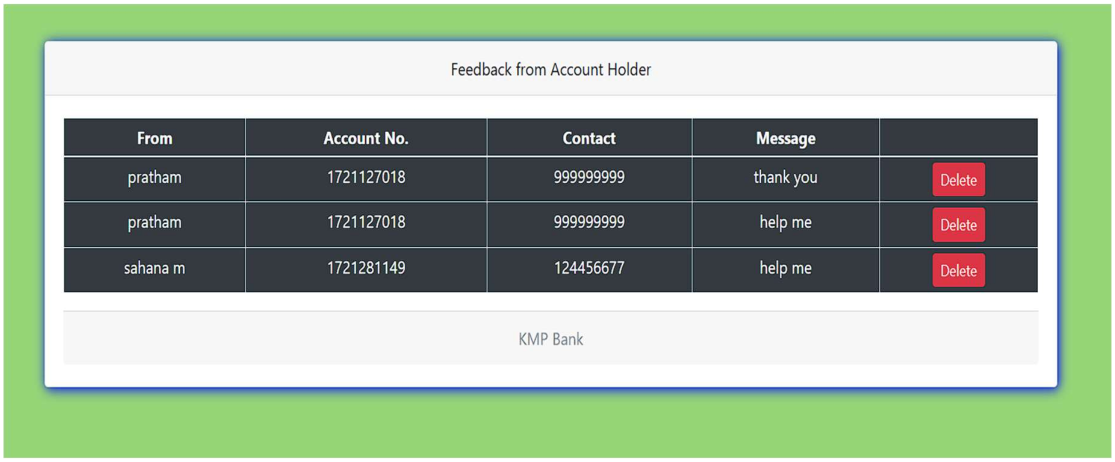
New Account Form

Name		CNIC	
Account Number	1721582294	Account Type	Current
City		Address	
Email		Password	
Deposit		Source of income	
Contact Number		Branch	Please Select..

KMP Bank

Figure 5.8 – New Account Form Page

Manager can read messages sent from his users in feedback page shown in figure 5.9

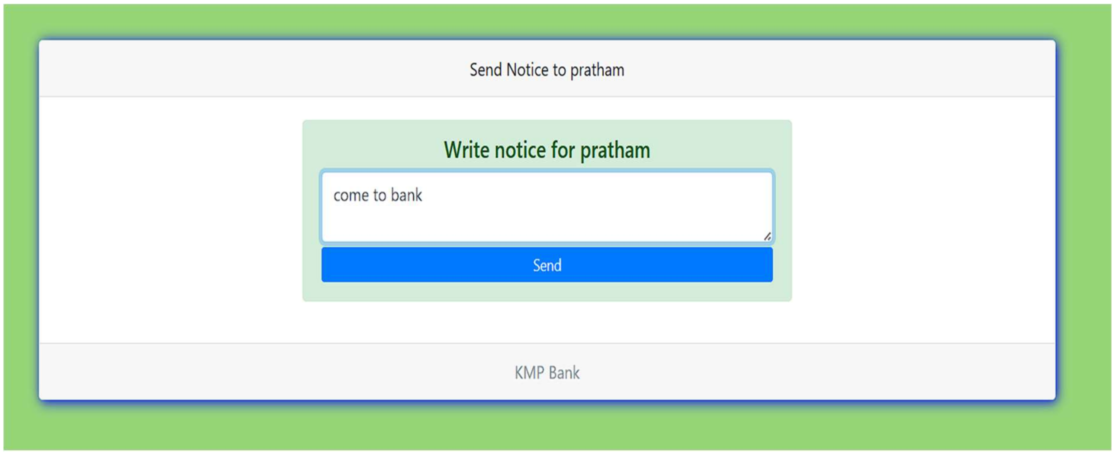


Feedback from Account Holder

From	Account No.	Contact	Message	
pratham	1721127018	999999999	thank you	Delete
pratham	1721127018	999999999	help me	Delete
sahana m	1721281149	124456677	help me	Delete

Figure 5.9 – Feedback Page

Manager can send notice to user in send notice tab shown in figure 5.10



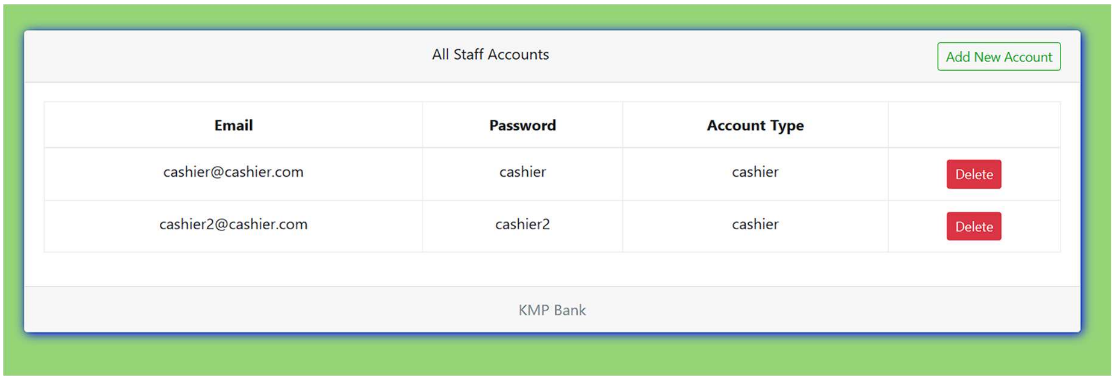
Send Notice to pratham

Write notice for pratham

come to bank

Figure 5.10 – Send Notice Page

Manager can manage bank staff shown in figure 5.11



All Staff Accounts

Email	Password	Account Type	
cashier@cashier.com	cashier	cashier	Delete
cashier2@cashier.com	cashier2	cashier	Delete

Figure 5.11 – Staff Account Page

After Cashier login with his email and password dashboard shown in figure 5.12

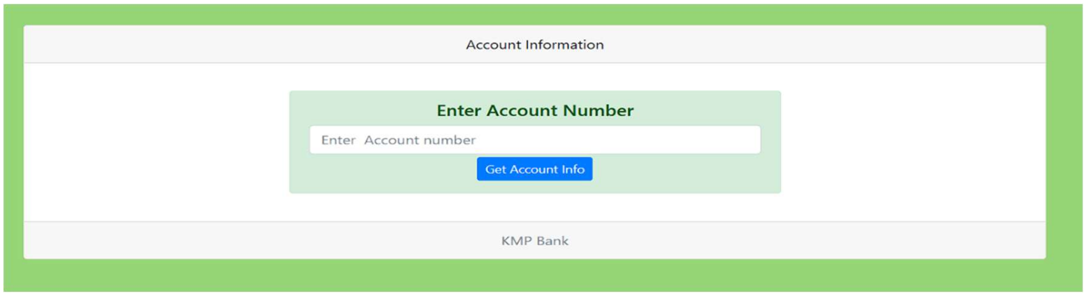
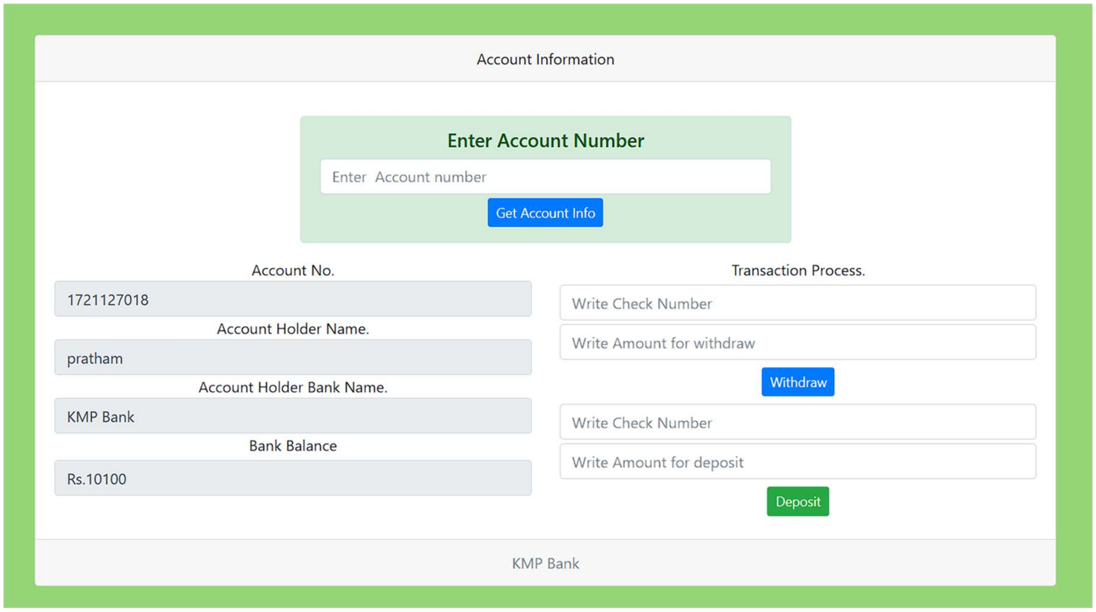


Figure 5.12 – Cashier dashboard

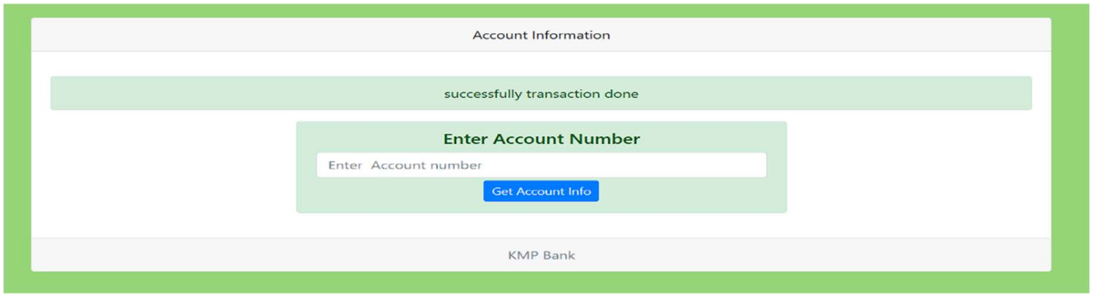
Cashier can withdraw or deposit money of user by entering user account number there he have option to withdraw or deposit, enter check no and amount to do transaction shown in figure 5.13



Account No.
1721127018
Account Holder Name.
pratham
Account Holder Bank Name.
KMP Bank
Bank Balance
Rs.10100

Figure 5.13 – After entering account number

After successful transaction the message will pop up shown in figure 5.14



successfully transaction done

Figure 5.14 – After transaction (withdraw or deposit)

Chapter 6

Conclusion

The proposed Bank Management System (BMS) represents a comprehensive solution designed to streamline and enhance the core banking operations through the effective use of Database Management System (DBMS) technologies. This system addresses the critical needs of modern banking institutions by offering robust features for customer account management, transaction processing, loan management, financial reporting, multi-branch support, user authentication, customer feedback, and notice management. By automating routine banking processes, the BMS significantly reduces manual effort and minimizes errors, leading to improved operational efficiency and accuracy.

The system's architecture, based on a client-server model, ensures scalability, reliability, and security. The centralized database design, with its normalized schema and indexing, supports efficient data management and retrieval, while stringent security measures, including data encryption and role-based access control, protect sensitive information from unauthorized access. The user-friendly interface further enhances the usability of the system, making it easier for bank staff and customers to perform various banking operations seamlessly.

Incorporating advanced features such as real-time transaction processing, automated reconciliation, comprehensive financial reporting, and detailed audit trails, the BMS provides a robust platform that supports informed decision-making, regulatory compliance, and strategic planning. The multi-branch support ensures consistent and accurate data management across all branches, facilitating inter-branch transactions and centralized control.

Future Enhancements:

1. Integration with Payment Gateways: To further enhance user convenience, future developments could include the integration of secure online payment gateways. This would allow users to complete the entire loan application process, including payment, within the Bank Management System platform.

2. Mobile Application Development: Creating a dedicated mobile application for Bank Management System can extend the reach and accessibility of the system. Users could easily apply for loans on the go, receive push notifications, and access the system from their mobile devices.

3. Social Media Integration: Integrating the Bank Management System with social media platforms allows users to share their loan application experiences, reviews, and ratings. This can contribute to increased visibility and user engagement.

4. Multi-Language Support: To cater to a diverse audience, incorporating multi-language support in the system ensures that users can interact with the Bank Management System in their preferred language, expanding its accessibility.

Overall, the Bank Management System is poised to revolutionize traditional banking operations by enhancing efficiency, reducing errors, and improving customer service. It provides a solid foundation for the bank to grow and adapt to the evolving needs of the financial industry, ensuring sustained business continuity and customer satisfaction. Through the effective implementation of this system, the bank can achieve significant operational improvements, better risk management, and a competitive edge in the market.

References

[1] https://chat.openai.com/share/472c1802-ef70-4eec-ad66-4475eb8180f1

[2] https://www.campcodes.com/loan-management-system-in-php-mysql

[3] https://www.apachefriends.org/

[4] https://www.cittabase.com/phpdbinsights/

[5] https://www.w3schools.com/MySQL/default.as

[6] https://www.youtube.com