iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.69800

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Bank Management System

Ms.Sakshi Kamble[1], Mr.Rushikesh Khedkar[2], Ms.Punam Kolhe[3], Mr. Arya Kothavale[4], Prof. Afrin Sheikh[5]
Savitribai Phule Pune University, Department of Engineering Science KJ College of Engineering and Management Research Pune, India

***Abstract:** This research paper offers a thorough overview of The digital revolution in financial services highlighting the need for flexible, secure, and user-friendly banking systems, especially for co-operative and rural banks that often face infrastructural and technological limitations. This project, "Bank Management System", introduces a comprehensive full-stack web application designed to address these challenges. The system provides end-to-end digital banking functionalities such as secure user registration, encrypted login, balance checking, fund transfers, transaction history, loan applications and approvals, as well as administrative and IT support tools.*

Developed using HTML, CSS, JavaScript for the frontend and Node.js with Express.js for the backend, the application uses SQLite as the database to manage user data, transactions, and loan records efficiently. The interface is made responsive with the help of Bootstrap, while bcrypt is used for password hashing, ensuring data security. The system employs role-based access control, allowing different features and dashboards for users, administrators, and IT admins.

One of the unique aspects of this project is its modular design and scalability. It is designed to be user-friendly and scalable, capable of being enhanced further to include mobile app integration, UPI/payment gateway linkage, AI-based fraud detection, and real-time customer support features.

I. INTRODUCTION

In today's technology-driven world, digital transformation in banking has become imperative to meet the growing expectations of customers. The banking industry is evolving rapidly with the help of modern technologies. Our project titled "Bank Management System" aims to provide a comprehensive web-based banking platform that brings traditional banking services online. The system ensures secure and convenient banking operations including user registration, login, balance inquiries, fund transfers, loan management, and more.

Developed using modern technologies such as HTML, CSS, JavaScript, Node.js, and SQLite, this application simplifies everyday banking activities for both customers and bank staff.

This platform promotes the idea of self-service and 24/7 banking access, minimizing the need for physical visits to the bank and offering users the freedom to manage their finances from any location. This becomes especially crucial in rural or semi-urban areas where banking infrastructure may be limited.

The system offers end-to-end digital banking functionalities, including secure user registration, encrypted login, balance inquiries, fund transfers, transaction history tracking, loan applications and approvals, and tools for administrative and IT support. Built with a robust tech stack—HTML, CSS, JavaScript for the frontend, Node.js with Express.js for the backend, and SQLite for efficient data management—the application ensures both performance and security. Passwords are hashed using bcrypt, and Bootstrap is used to make the interface responsive across devices. Additionally, role-based access control provides customized dashboards and access levels for users, administrators, and IT

II. LITERATURE REVIEW

Traditional banking systems often rely on outdated software and manual processes, which can lead to inefficiencies and increased vulnerability to errors. These legacy systems struggle to handle the growing volumes of transactions and data, making it difficult to process information quickly and accurately.

The lack of modern data management solutions exacerbates these issues, as banks find it challenging to efficiently store, retrieve, and analyse vast amounts of customer data. Additionally, these older systems often lack robust security measures, making them more susceptible to cyberattacks, fraud, and data breaches. As a result, customers' sensitive information is at risk, and banks may face significant financial and reputational damage.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

III. PROBLEM DEFINITION

Traditional banking systems, reliant on manual processes for managing customer accounts, deposits, loans, and transaction histories, face significant challenges such as time-consuming operations, human errors, and limited transparency.
The lack of real-time access to transactions and centralized data creates inefficiencies and increases the risk of discrepancies. Additionally, the restricted accessibility outside banking hours often inconveniences customers who need quick or urgent services. This fragmentation and manual approach not only slow down financial operations but also hinder customer satisfaction and trust. Modern digital banking solutions aim to address these issues by offering streamlined, accurate, and transparent services that are accessible 24/7, enabling better data management and enhancing overall efficiency.

IV. METHODOLOGY

A. Requirement Analysis

This document outlines the key requirements for the development of a web-based application, covering both software and hardware components essential for successful implementation. The application will be built using modern web technologies with a focus on performance, security, and user experience. The stack selected includes lightweight, efficient tools suitable for small to medium-scale applications, making it ideal for startups, prototypes, or academic projects.

1) Frontend Technologies:

- HTML5-Structures the content of the web pages and provides semantic elements for accessibility and search engine optimization.
- CSS3-Styles and visually enhances the application interface with support for modern design techniques such as flexbox, grid layout, transitions, and media queries.
- JavaScript-Enables dynamic functionality on the client side, including real-time form validation, interactive content, and event-driven behavior.
- Bootstrap-A mobile-first CSS framework that helps create responsive layouts quickly using its pre-defined components and grid system.

2) Backend Technologies:

- Node.js-An open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside of a browser, enabling efficient server-side scripting.
- Express.js-A flexible web application framework for Node.js that simplifies API development and server-side routing, and supports middleware for scalable applications.

3) Database:

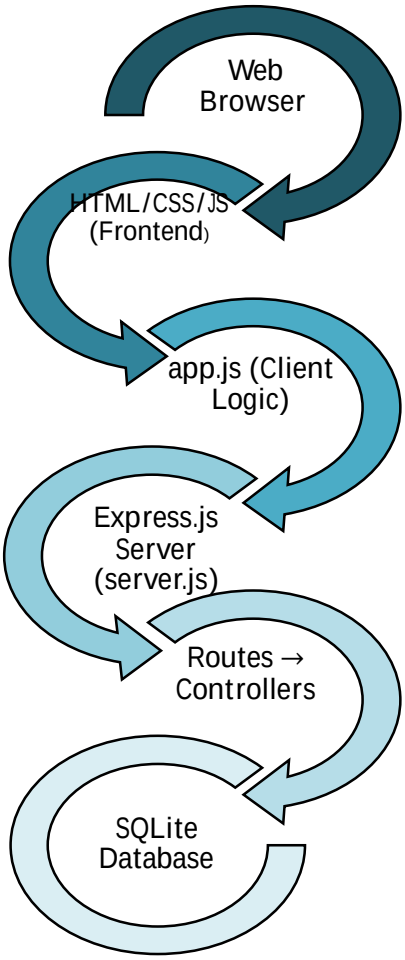
- SQLite-A lightweight, file-based database engine that is easy to set up and requires minimal configuration, ideal for development or low-traffic production environments.

4) Development Environment:

- Visual Studio Code (VS Code)-A robust code editor with extensive support for Node.js, JavaScript, and frontend development. It includes Git integration, debugging tools, and extension support.
- Node.js Runtime-Allows server-side JavaScript execution and supports development using Express.js and other related packages via npm (Node Package Manager).
- Git-Facilitates version control and collaboration, enabling code tracking, branching, and integration with platforms like GitHub for team-based development.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com



V. FUNCTIONS AND MODULES

The banking application is designed using a modular architecture to ensure scalability, maintainability, and ease of development. Each module is responsible for a specific set of tasks, promoting separation of concerns and better organization of code. This modular design also enables independent testing, easier debugging, and the potential to extend the application with new features in the future.

1) User Module: -

This module manages user-related operations such as account creation, authentication, and access to personal banking features.

- User Registration-Users can sign up by providing personal details including name, mobile number, Aadhar number, PAN card, email, and address.
- Secure Login-Authentication is implemented using bcrypt to securely hash and verify user passwords, protecting against common attacks like brute-force and credential theft.
- User Dashboard -After logging in, users are redirected to a personalized dashboard displaying their account summary, current balance, and a list of recent transactions.

2) Role-Based Access Control (RBAC): -

Access to various features is controlled using clearly defined user roles, ensuring secure and appropriate access for each type of user.

- Roles Defined-
  - User: Basic banking operations like balance inquiry and fund transfers.
  - Admin: Manages user accounts, approves or rejects loans, and oversees all transactions.
  - IT Admin: Has unrestricted access, with the ability to manage all users, roles, and system-level settings.

©IJRASET: All Rights are Reserved

- Access Control-Each action or route within the application is protected based on the user's role, enhancing both security and compliance.

3) Transaction Module: -

Handles all financial transactions within the application, ensuring accuracy and traceability.

- Balance Inquiry-Users can check their current account balance instantly through the dashboard or transaction page.
- Fund Transfer-Users can transfer funds to other registered users by entering the recipient's account number and the amount.
- Add/Withdraw Funds-Controlled by Admin or designated Cashier roles. These operations are logged for accountability.
- Transaction History-Users can view the last 10 transactions with detailed information like date, amount, type (debit/credit), and transaction ID.

4) Customer Support Module: -

Ensures users can communicate their concerns or seek assistance through a built-in support system.

- Support Form-A contact form allows users to submit queries, complaints, or requests to the support team.
- Admin Panel-Admins can view and manage incoming queries, assign them to support agents, and mark them as resolved once addressed.

5) Admin Dashboard

A centralized control panel offering administrative features and system-wide insights.

- User Management-Admins can view, edit, or delete user accounts and access detailed profiles of all users.
- Manual Transactions-Admins can perform direct deposits or withdrawals on user accounts, especially in scenarios like branch-level operations.
- Financial Reports-Real-time data ontotal deposits, withdrawals, and other financial metrics help Admins make informed decisions.

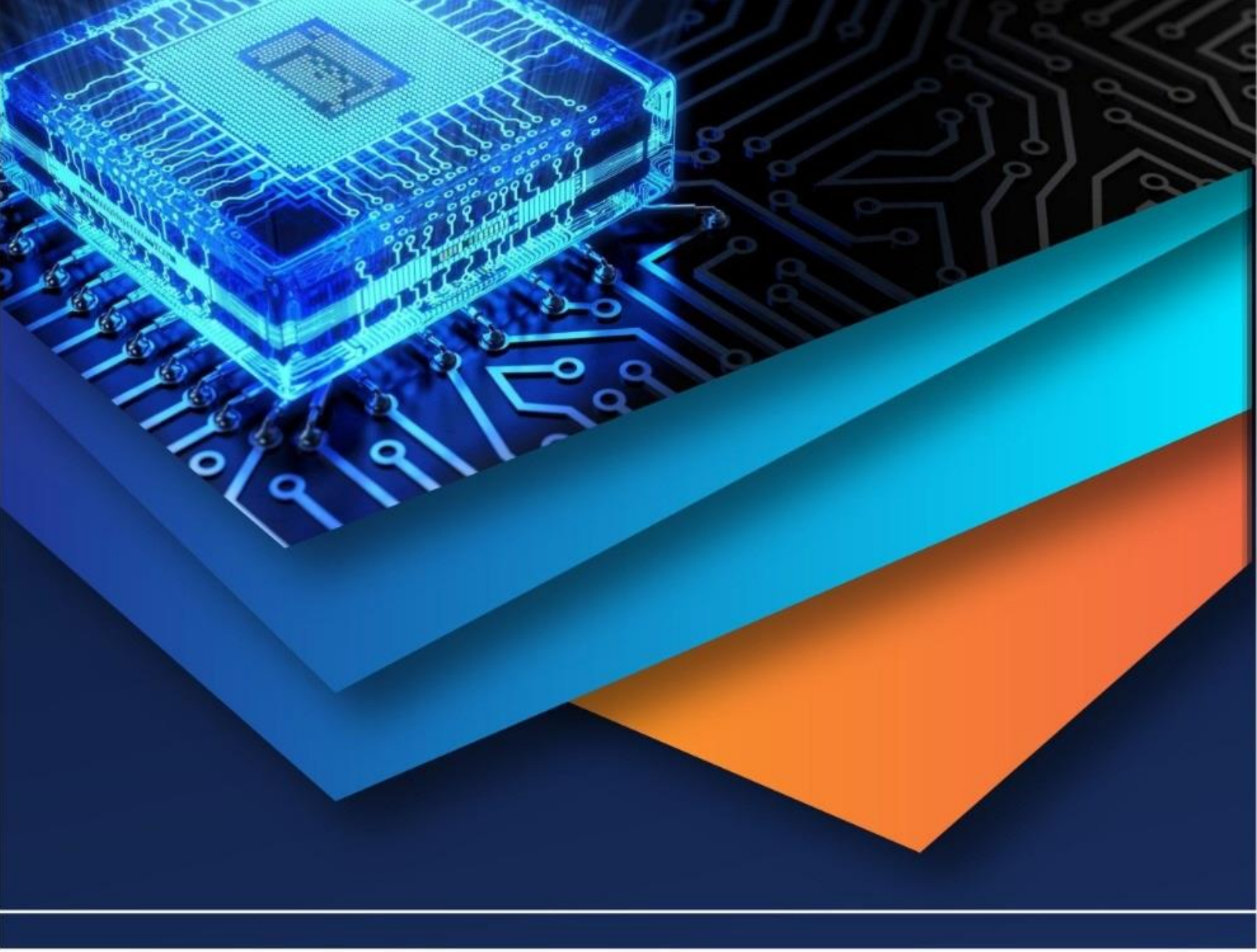
VI. CONCLUSION

The "Bank Management System" successfully provides an end-to-end web-based solution for small to mid-sized banks. It includes all vital components for day-to-day banking operations while ensuring accessibility, scalability, and security. The application demonstrates how technology can streamline traditional banking systems and significantly improve the user experience. It is a practical solution, especially for co-operative banks aiming to digitize their operations affordably.

REFERENCES

[1] Herbert Schildt, "C++: The Complete Reference," McGraw-Hill Education.
[2] Bjarne Stroustrup, "The C++ Programming Language," Addison-Wesley. -
[3] GeeksforGeeks, "File Handling in C++," [https://www.geeksforgeeks.org/file-handling-cpp/](https://www.geeksforgeeks.org /file-handling-cpp/).
[4] TutorialsPoint, "C++ Object-Oriented Programming," [https://www.tutorialspoint.com/cplusplus/cpp_object_oriented.htm] (https://www.tutorialspoint.com/cplusplus/cpp_object_oriented.htm).
[5] https://www.researchgate.net/publication/380214416_Online_banking_management_system
[6] http://www.irjmets.com/

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)