



Internship Report

Web Development (MERN Stack)

DLithe Consultancy Services Pvt. Ltd.





Internship Report

Trainee/Intern Name: Deepthi K M

Reg. no: 4CB21IS012

Period: 4 Months

Job Assignment: Banking application for Loan recommendation based on user Transactions

Organization: DLithe Consultancy Services Pvt. Ltd.

Supervisor's Name: Mr. Purushottam Pattanashetty

Observations:

- The report introduces a modern approach to digital banking by implementing a full-stack web application that automates loan recommendations through transaction analysis.
- The various factors that are taken into consideration in loan recommendation models are deposits, withdrawals, account balance patterns, and transaction frequency.
- The system presents future opportunities for enhancing personalization through AI/ML integration.

Submitted to

Signature of Training Supervisor

Date: 05/05/2025

Signature of Co-ordinator

Date:05/05/2025



Letter of Transmittal

To,

Program Co-ordinator
DLithe Consultancy services
Bengaluru

Dear Sir,

I am writing to submit my report on Web development(MERN Stack). The training program was an invaluable learning experience, and I am grateful for the opportunity to participate.

The training program covered various aspects of full-stack development using MongoDB, Express.js, React.js, and Node.js. I gained hands-on experience in designing and developing a functional banking application with both user and admin modules. The user side enables users to perform transactions such as deposits and withdrawals, and based on their financial activity, loan recommendations are made. The admin side allows for viewing user details, transaction history, loan application management, and handling user messages and settings.

The report includes a detailed overview of the training program, including the topics covered, the learning objectives, and the outcomes achieved. It also provides observations and insights into the practical implementation of a full-stack application and how transaction data can be utilized for intelligent decision-making like loan recommendations.

I believe that the knowledge and skills that I acquired during the training program will be valuable to our organization. As digital banking continues to evolve, the ability to build intelligent web applications using MERN Stack and data-driven logic will play a vital role in delivering modern financial services.

I hope that the report provides useful insights into the benefits of on-job training and the potential of intelligent web applications.

Sincerely,

Name: Deepthi K M

Table of Contents

Introduction	5
Literature survey	5
Proposed work	6
Implementation	11
Results	12
Conclusion	14
References	14

1. INTRODUCTION

Banking plays a critical role in the financial stability of individuals and the economy. With the growing demand for digital financial services, understanding user transactions has become essential in offering efficient and personalized banking solutions. Loan recommendation is a key functionality in banking systems that helps users access financial support based on their account activities. Accurate evaluation of deposits, withdrawals, and transaction history is crucial for determining loan eligibility and maintaining financial transparency.

Loan recommendation is one of the essential but challenging tasks in banking applications. Since it depends on multiple financial factors such as transaction patterns, balance consistency, and user behavior, implementing an effective recommendation system requires handling various types of data. This makes loan recommendation a structured yet detailed process that must ensure accuracy, fairness, and responsiveness. While traditional systems relied on manual processing, modern web-based platforms now allow for real-time processing and tracking of financial data to support better decision-making.

The application was developed using the MERN Stack—MongoDB for database management, Express.js and Node.js for server-side functionality, and React.js for building a dynamic user interface. The system consists of user and admin modules, where users can perform transactions and view loan recommendations, while the admin can manage user data, review transactions, and handle loan applications efficiently.

2. LITERATURE SURVEY

The reviewed research papers and published articles collectively address the following aspects. These surveys aim to assess the current landscape of web-based banking systems, highlighting prominent methodologies, existing challenges, and opportunities for advancement. The main focus areas typically include:

1. **Evolution of Web-Based Banking Applications:** An overview of the literature tracing the shift from traditional banking to digital platforms, emphasizing how online systems manage user transactions and improve accessibility.

2. **Overview of Current Loan Recommendation Systems:** A detailed review of existing frameworks that use rule-based mechanisms and financial behavior profiling to suggest suitable loan options.
3. **User Behavior Analysis via Transactions:** Examination of studies analyzing deposits, withdrawals, and other financial activities to assess creditworthiness and identify spending patterns.
4. **Web Technologies Empowering Digital Banking:** Insights into modern tech stacks like the MERN (MongoDB, Express.js, React.js, Node.js), which facilitate seamless frontend-backend communication, real-time processing, and secure data operations.
5. **Implementation of Role-Based Access Control:** Exploration of role-based user segregation in banking platforms to ensure data confidentiality, streamlined operations, and customized user interfaces for administrators and customers.
6. **Security and Data Protection Mechanisms:** Analysis of secure practices for managing sensitive user information, including data encryption, transaction safety, and defenses against unauthorized system access.
7. **Shortcomings of Conventional Banking Methods:** Identification of inefficiencies in manual loan assessments and customer interactions, reinforcing the importance of automated, digital banking tools.
8. **Prospective Developments in Banking Applications:** Discussion on emerging trends such as intelligent loan recommendation engines, more intuitive UI designs, enhanced administrative functionalities, and the integration of third-party financial services.

3. PROPOSED WORK

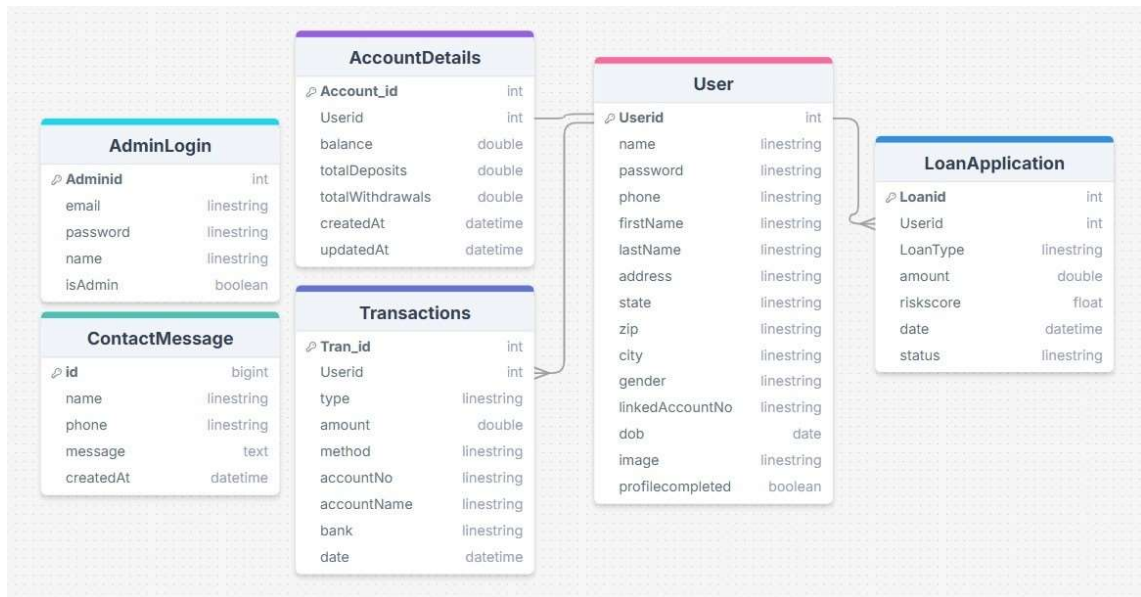
One of the key challenges in the Banking Application project is designing an efficient system that manages user transactions and provides accurate loan recommendations based on transaction history. The project includes creating a database schema, defining workflows, and designing the user interface to ensure smooth operation and user experience.

The system consists of two main modules: the User side and the Admin side. Users can perform deposits and withdrawals, and based on their transaction history, the system

recommends suitable loan options. The Admin has access to all user details, transaction records, and loan applications, which can be approved or rejected.

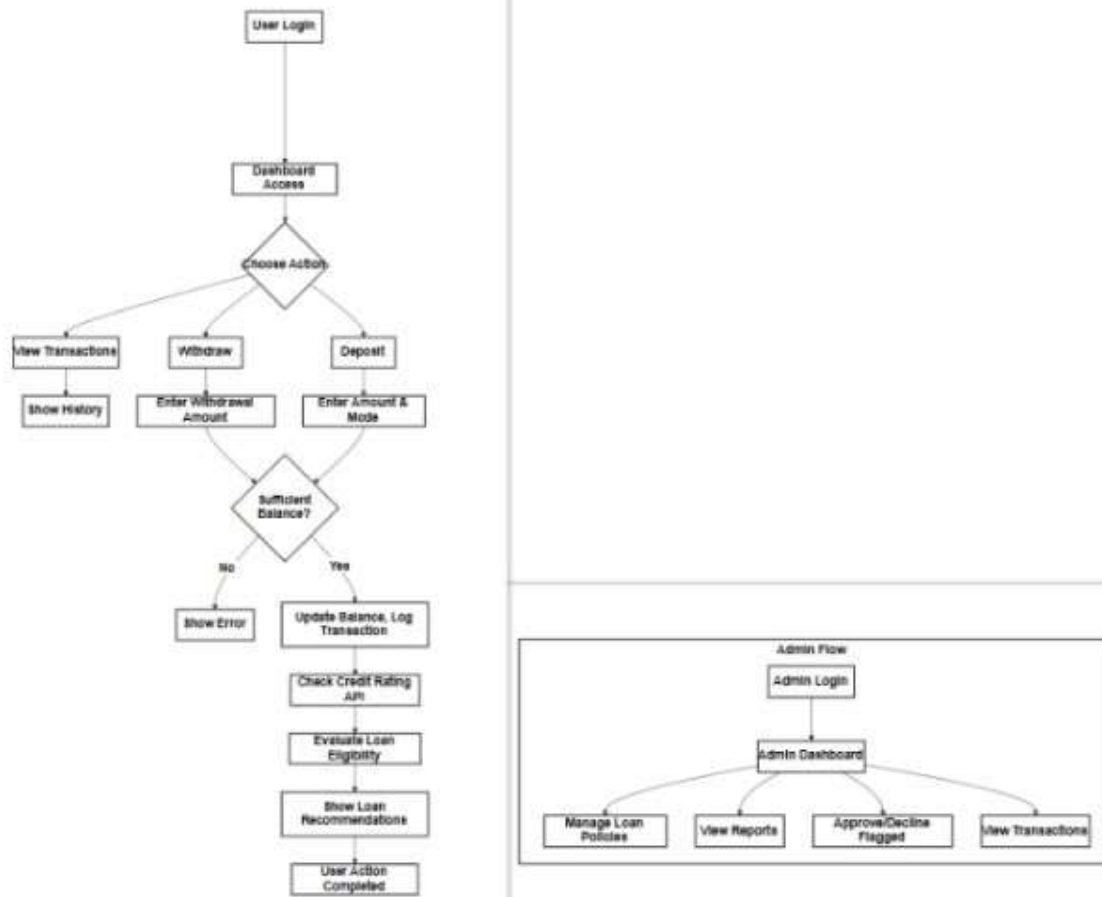
To illustrate the overall design and flow of the application, we have created several diagrams and models:

- **Database Design:** Structured collections in MongoDB to store user profiles, transactions, loan applications, and admin data efficiently.



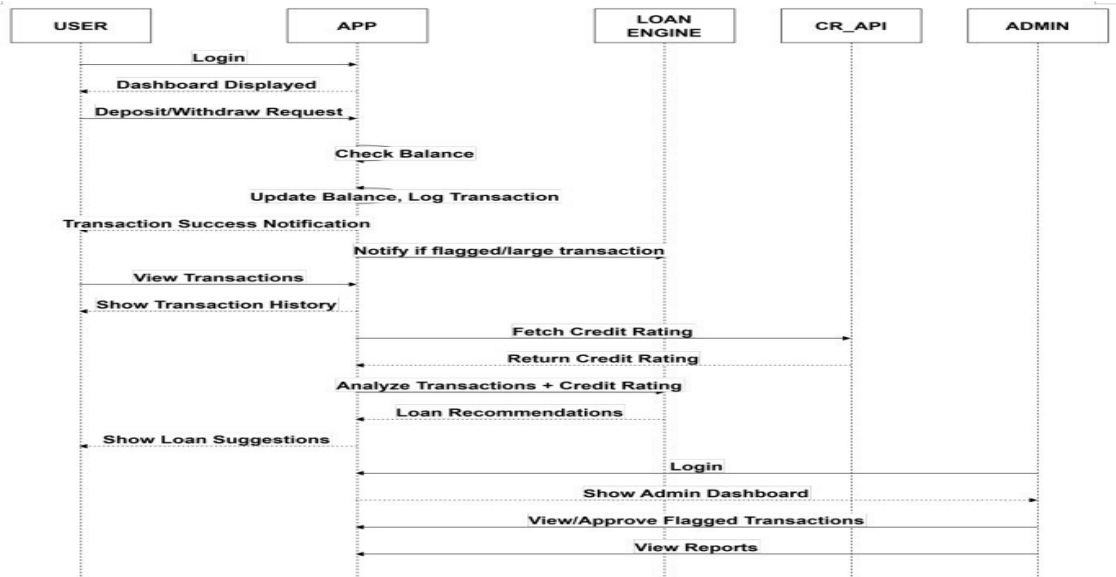
Database Design

- **Workflow Diagram:** Visual representation of user actions, transaction processing, and loan recommendation logic.



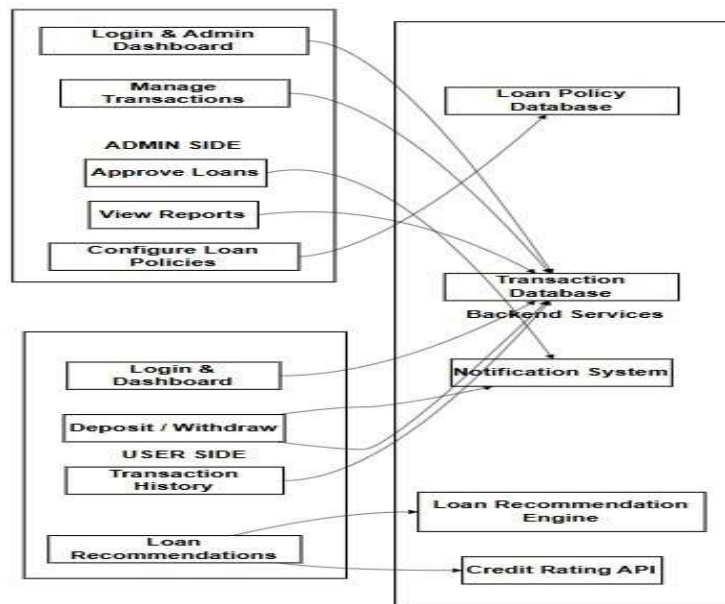
Workflow Diagram

- **Sequence Diagram:** Depicts the interaction between users, admin, and the backend system during key operations like loan application and approval.



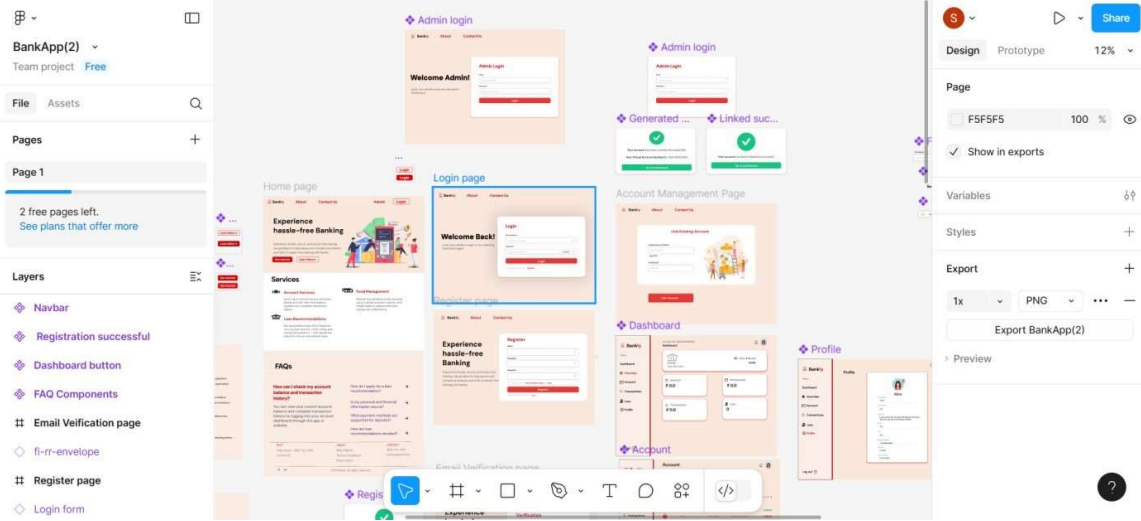
Sequence Diagram

- **Functional Architecture:** Shows the components of the MERN stack and how they integrate to support the application's functionality.



Functional Achitecture Diagram

- **Figma Design:** UI/UX prototype demonstrating the layout and navigation for both user and admin interfaces.



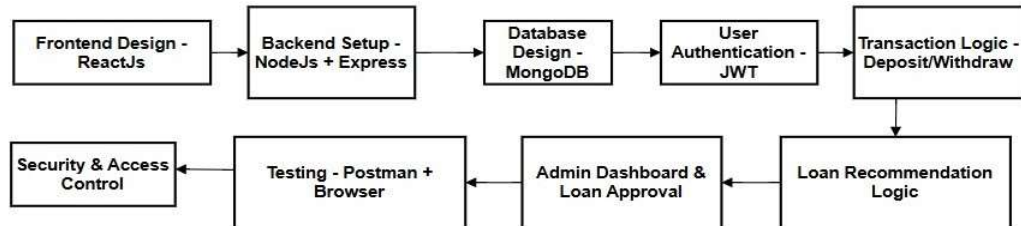
Figma Design

The loan recommendation logic is based on predefined criteria analyzing the user's transaction patterns, including frequency and volume of deposits and withdrawals. This helps the system determine eligibility and suggest loan amounts accordingly.

Here is an overview of the proposed process:

- Step 1: Users perform transactions such as deposits and withdrawals.
- Step 2: Transaction data is stored and updated in the MongoDB database.
- Step 3: The system analyzes transaction history to generate loan recommendations dynamically.
- Step 4: Users can apply for loans based on these recommendations.
- Step 5: Admin reviews loan applications, verifies user data, and approves or rejects the requests.

4. IMPLEMENTATION



1. Database Design: A MongoDB database was used for storing user data, transaction records, and loan application details. Collections were structured for Users, Transactions, Admin, and Loans, ensuring normalized and scalable data management.
2. Frontend and UI Design: The interface was designed using Figma, then implemented using React.js. User-friendly forms and dashboards were created for both the user and admin, allowing for smooth navigation and functionality like login, deposits, withdrawals, and loan applications.
3. Backend Development: Node.js with Express.js was used to handle API requests, perform server-side operations, and manage authentication and data routing. Backend endpoints were created for handling user registration, login, transaction operations, and loan management.
4. Loan Recommendation Logic: Loan recommendations are based on transaction frequency and volume. If a user's deposit/withdrawal history meets predefined criteria, the system recommends suitable loan options. This logic was implemented using conditional checks and rules in the backend.
5. Workflow and Sequence Implementation: The full workflow was built with the following process: user actions (deposit, withdrawal) → transaction logging → data analysis → loan suggestion → admin approval system. This process flow was implemented based on the workflow and sequence diagrams created during design.
6. Admin Dashboard Functionality: The admin panel allows viewing of all users, their transaction summaries, and pending loan requests. Admins can approve or reject loans, and send messages or status updates to users, all through secure endpoints.
7. Testing and Validation: The application was tested using Postman and browser-based

manual testing. Transactions, logins, and loan logic were validated to ensure proper working conditions across different users.

8. Security and Access Control: Authentication was handled using JWT (JSON Web Tokens) to secure user sessions. Role-based access ensured that only admins could manage users and loan approvals.

5. RESULTS

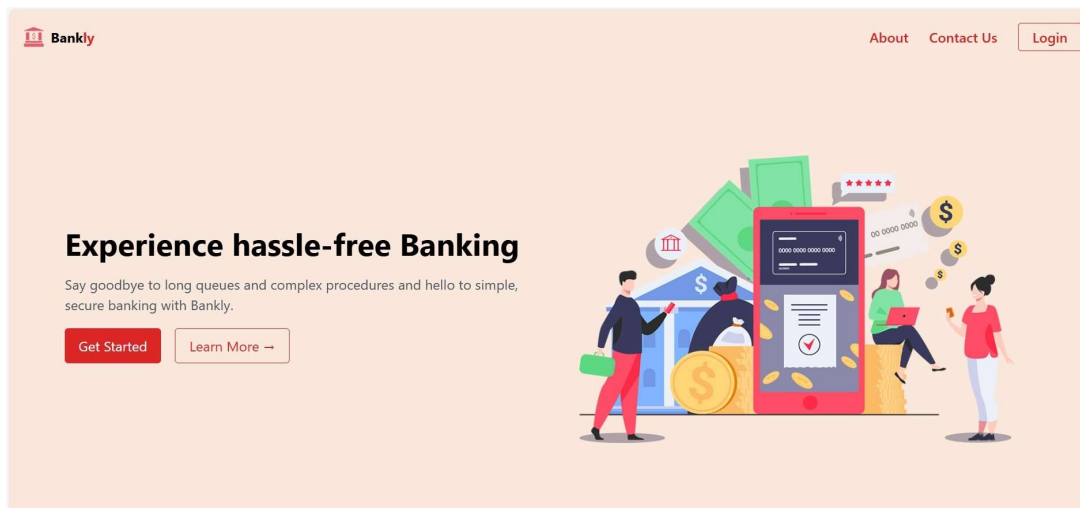


Figure 1: Landing page of Bankly

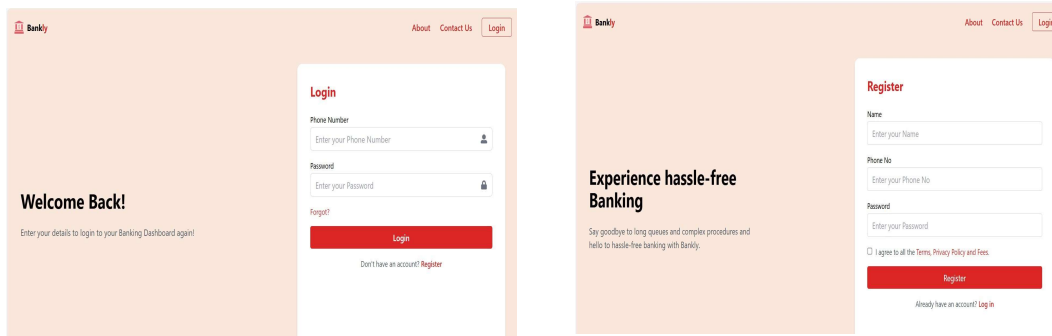


Figure 2: Register and Login page for a user of Bankly

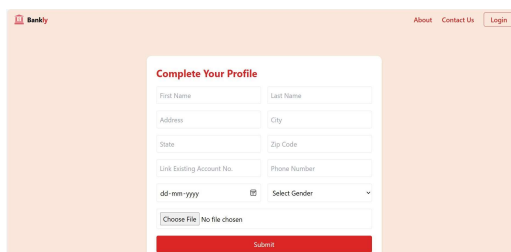


Figure 3: Profile Setup for a User

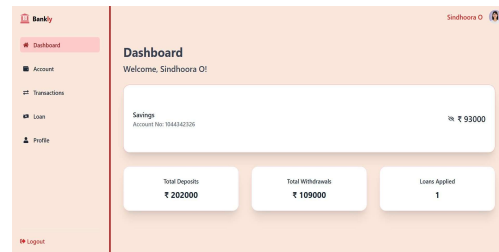


Figure 4: User Dashboard

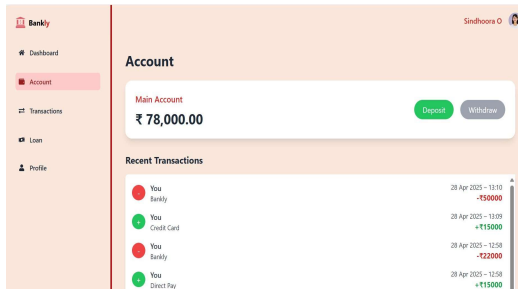


Figure 5: User Account section

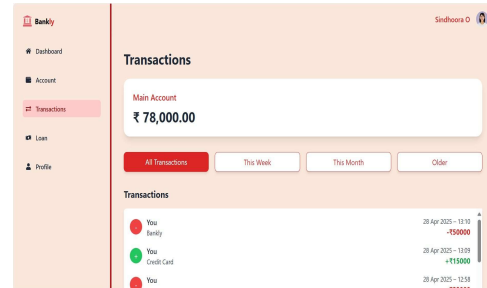


Figure 6: User Transactions section

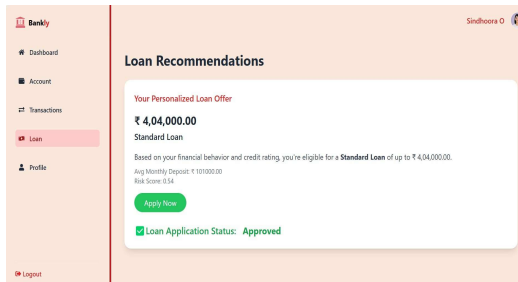


Figure 7: User Loan Recommendations

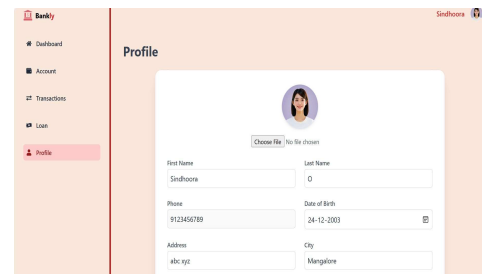


Figure 8: User profile

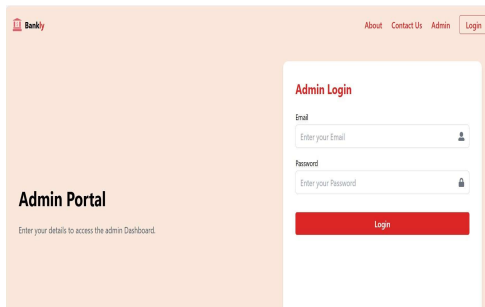


Figure 9: Admin Login

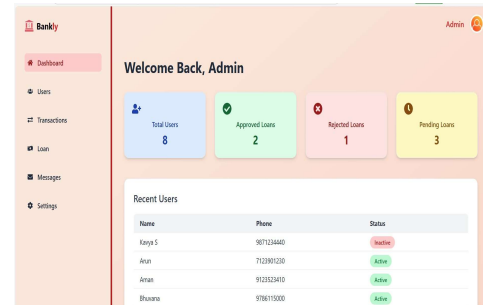


Figure 10: Admin Dashboard

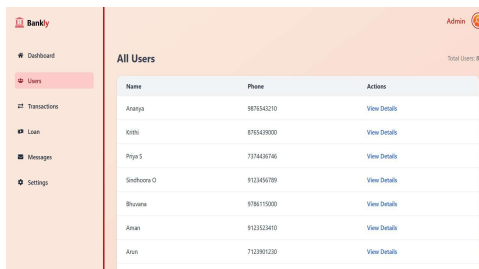


Figure 11: All Users - Admin

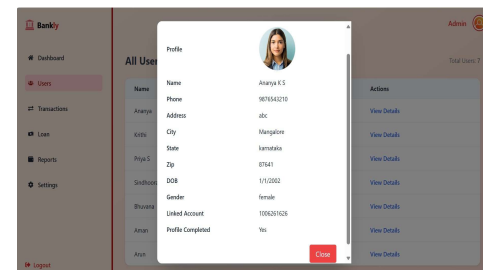


Figure 12: User details



The screenshot shows the Admin dashboard with a sidebar menu containing Dashboard, Users, Transactions, Loan, Messages, and Settings. The main content area is titled 'All Transactions' and displays a table of transactions.

User	Type	Amount	Method	Account No	Date	Time
Sindhuora D	Deposit	₹100000	Direct Pay	1044342326	27 Apr 2025	23:30
Sindhuora D	Deposit	₹10000	Direct Pay	1044342326	27 Apr 2025	23:30
Sindhuora D	Withdraw	₹100000		1044342326	27 Apr 2025	23:30
Sindhuora D	Withdraw	₹10000		1044342326	27 Apr 2025	23:43
Sindhuora D	Withdraw	₹10000		1044342326	27 Apr 2025	23:43
Sindhuora D	Deposit	₹10000	Credit Card	1044342326	27 Apr 2025	23:44
Akanya V S	Deposit	₹100000	Direct Pay	1000261926	27 Apr 2025	23:46
Sindhuora D	Deposit	₹100000	Direct Pay	1044342326	28 Apr 2025	00:25
Sindhuora D	Deposit	₹100000	Direct Pay	1044342326	28 Apr 2025	10:19

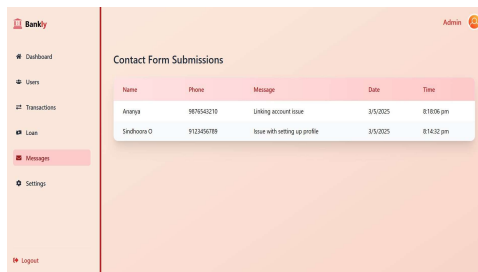
Figure 13: All transactions - Admin



The screenshot shows the Admin dashboard with a sidebar menu. The main content area is titled 'Loan Applications' and displays a table of loan applications.

User	Loan Type	Amount	Risk Score	Date	Status	Action
Bhavana	Premium Loan	₹4,950,000	0.52	3/5/2025	Applied	Approve Reject
Bhavana	Standard Loan	₹2,200,000	0.52	3/5/2025	Applied	Approve Reject
Anan	Premium Loan	₹6,000,000	0.50	3/5/2025	Applied	Approve Reject
Anan	Premium Loan	₹20,840,000	0.50	3/5/2025	Rejected	Approve Reject
Akanya	Standard Loan	₹6,000,000	0.43	3/5/2025	Approved	Approve Reject
Sindhuora D	Standard Loan	₹7,740,000	0.58	3/5/2025	Approved	Approve Reject

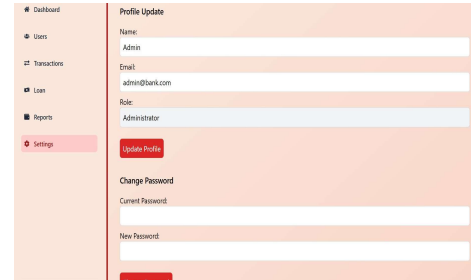
Figure 14: Applied Loan details



The screenshot shows the Admin dashboard with a sidebar menu. The main content area is titled 'Contact Form Submissions' and displays a table of submissions.

Name	Phone	Message	Date	Time
Akanya	9876543210	Waiting account issue	3/5/2025	8:18:08 pm
Sindhuora D	9123456789	Issue with setting up profile	3/5/2025	8:14:32 pm

Figure 15: Messages/Queries



The screenshot shows the Admin dashboard with a sidebar menu. The main content area contains two forms: 'Profile Update' and 'Change Password'.

Profile Update

Name: Admin

Email: admin@bank.com

Role: Administrator

Change Password

Current Password: [Input Field]

New Password: [Input Field]

Figure 16: Admin Settings

6. CONCLUSION

The development of this banking application focused on building an efficient and scalable system using the MERN stack. The project followed a structured development lifecycle—from database design and frontend prototyping to implementing secure and robust backend logic.

The final application enables users to carry out essential banking operations and receive automated loan recommendations tailored to their transaction history. Administrators have the ability to manage user accounts and oversee the loan approval workflow. With a streamlined process, real-time operations, and strong data security measures, the system effectively meets its objective of improving loan recommendation accuracy and supporting informed decision-making.

Future enhancements may include features like real-time analytics, integrated notification systems, credit score integration, and AI-driven loan recommendation engines to further elevate the application's capabilities.

7. REFERENCES

- Roadmap.sh. (n.d.). *Developer Roadmaps*. <https://roadmap.sh>
- Mongoose. (n.d.). *Mongoose ODM v7.6.1 Documentation*. <https://mongoosejs.com/docs/>
- npm Documentation. (n.d.). *npm Docs*. <http://docs.npmjs.com>
- Express.js Official Docs. *Express Web Framework*. <https://expressjs.com/>