



Internship Report

Web Development (MERN Stack)

DLithe Consultancy Services Pvt. Ltd.



Internship Report

Trainee/Intern Name: Nayana G Hegde

Reg. no:

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

Observations:

- The application helps users perform banking tasks and get personalized loan recommendations.
- Factors like transaction history, account activity, income, and credit score influence loan suggestions.
- Built using the MERN stack, it ensures real-time updates, secure processing, and a responsive UI.
- Future improvements include adding ML models, credit risk analysis, and predictive loan features.

Submitted to

Signature of Training Supervisor

Date:

Signature of Co-ordinator

Date:



Letter of Transmittal

To,

Program Coordinator
DLithe Consultancy services
Bengaluru

Dear Sir,

I am pleased to submit my project report titled "**Banking Application for Loan Recommendation Based on User Transactions.**" This project was undertaken as a part of the training program, and it has been a valuable and enriching experience in enhancing both my technical and analytical skills.

The project focuses on building a MERN stack-based web application that allows users to perform core banking functions such as deposits and withdrawals while also receiving personalized loan recommendations. These recommendations are generated based on transaction history, account activity, and credit ratings, with the integration of a third-party credit rating API. The application also includes admin features for monitoring and managing users and transactions.

The report provides a comprehensive overview of the application's architecture, use cases, functionalities, and system design. It also highlights the significance of applying AI/ML-based logic in financial systems to provide smart, data-driven services.

I believe the skills and insights gained through this project will be highly beneficial for future roles in the fintech domain and beyond. I hope this report reflects the technical depth and practical knowledge acquired during the training.

Sincerely,

Name: Nayana G Hegde

Table of Contents

Introduction	5
Literature survey	5
Proposed work	6
Implementation	8
Conclusion	9
References	10

1. INTRODUCTION

Banking plays a vital role in ensuring the financial stability of both individuals and the broader economy. With the increasing demand for digital financial services, understanding user transactions has become essential for delivering efficient and personalized banking experiences. One of the key features in modern banking systems is loan recommendation, which provides users with tailored financial support based on their account activity. Accurate analysis of deposits, withdrawals, and transaction history is fundamental to determining loan eligibility and promoting financial transparency.

Loan recommendation is a critical yet complex feature in banking applications. It relies on a variety of financial indicators such as transaction trends, balance stability, and user behavior. Developing an effective recommendation system requires processing and analyzing diverse types of data, making it a structured and detailed task that must prioritize accuracy, fairness, and responsiveness. While traditional systems depended on manual evaluation, modern web-based platforms enable real-time processing and monitoring of financial data, significantly enhancing decision-making capabilities.

This application was built using the MERN Stack, which includes MongoDB for database management, Express.js and Node.js for backend development, and React.js for creating a responsive and interactive user interface. The system is divided into two modules: a user module and an admin module. Users can perform transactions and receive personalized loan recommendations, while admins have the ability to manage user data, monitor transactions, and efficiently process loan applications.

2. LITERATURE SURVEY

All the published papers and research articles cover the following points. These survey aim to identify the current state of the field, key methodologies, challenges, and potential areas for improvement. They typically includes:

- **Web Technologies in Banking:**

MERN stack is widely used for building responsive and scalable web applications. It enables full-stack development using JavaScript for both frontend and backend.

- **Rule-Based Loan Recommendation:**

Traditional systems use conditional logic based on user transactions and credit scores. These rules are implemented using backend logic in Node.js and stored in MongoDB.

- **Core System Features:**

Key features include deposits, withdrawals, transaction history, and loan suggestions. Admin functionalities like user management and transaction approval are also included.

- **Application Architecture:**

RESTful APIs connect the frontend (React) to backend services (Node/Express). MongoDB handles data storage, while routing and processing are managed via Express.js.

- **Security and Data Handling:**

JWT tokens are used for secure user sessions and authentication. Proper validation and error handling are essential for safe transaction processing.

- **Challenges in Existing Systems:**

Issues include managing concurrent transactions and ensuring data consistency. Real-time updates and efficient UI rendering also remain key concerns.

- **Enhancement Opportunities:**

Systems can be improved with better UX, notifications, and third-party integrations. Admins can be given tools to modify loan criteria dynamically through the UI.

- **Scalability and Maintenance:**

Modular code and database indexing are important for scaling transaction systems.

Version control and clean API structures help in long-term maintenance.

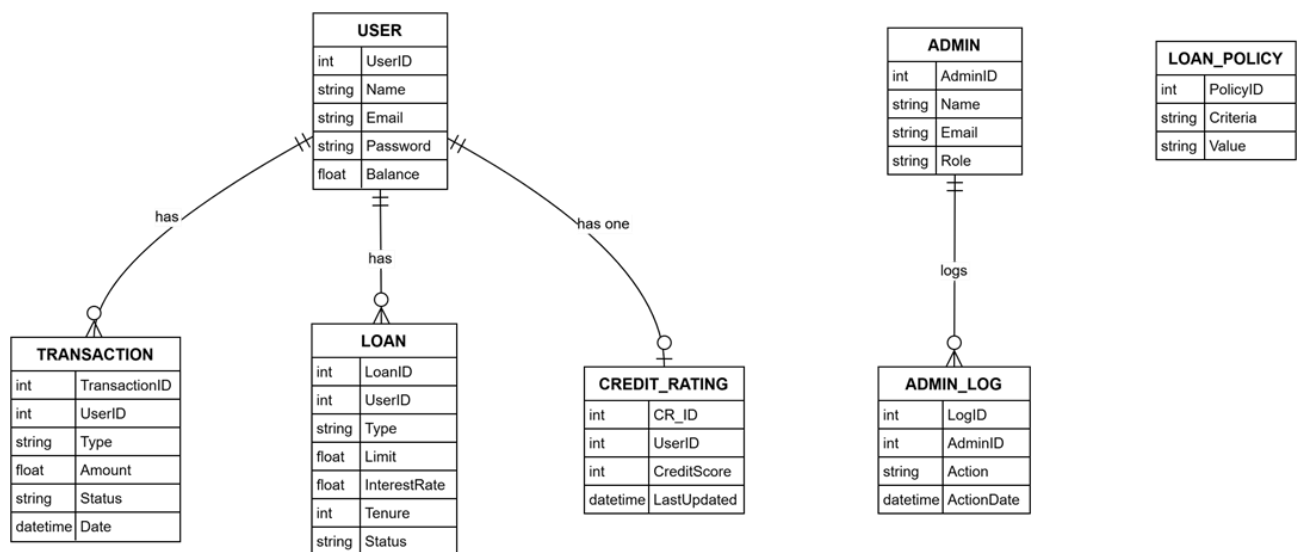
3. PROPOSED WORK

One of the primary challenges in the Banking Application project is developing an efficient system that effectively manages user transactions while offering accurate loan recommendations based on users' transaction histories. This project involves designing a robust database schema, defining clear workflows, and creating a user-friendly interface to ensure seamless functionality and a positive user experience.

The application is divided into two main modules: the User module and the Admin module. Users can perform actions such as deposits and withdrawals, and the system analyzes their transaction history to suggest suitable loan options. Meanwhile, the Admin has comprehensive access to user information, transaction records, and loan applications, with the ability to approve or reject loan requests.

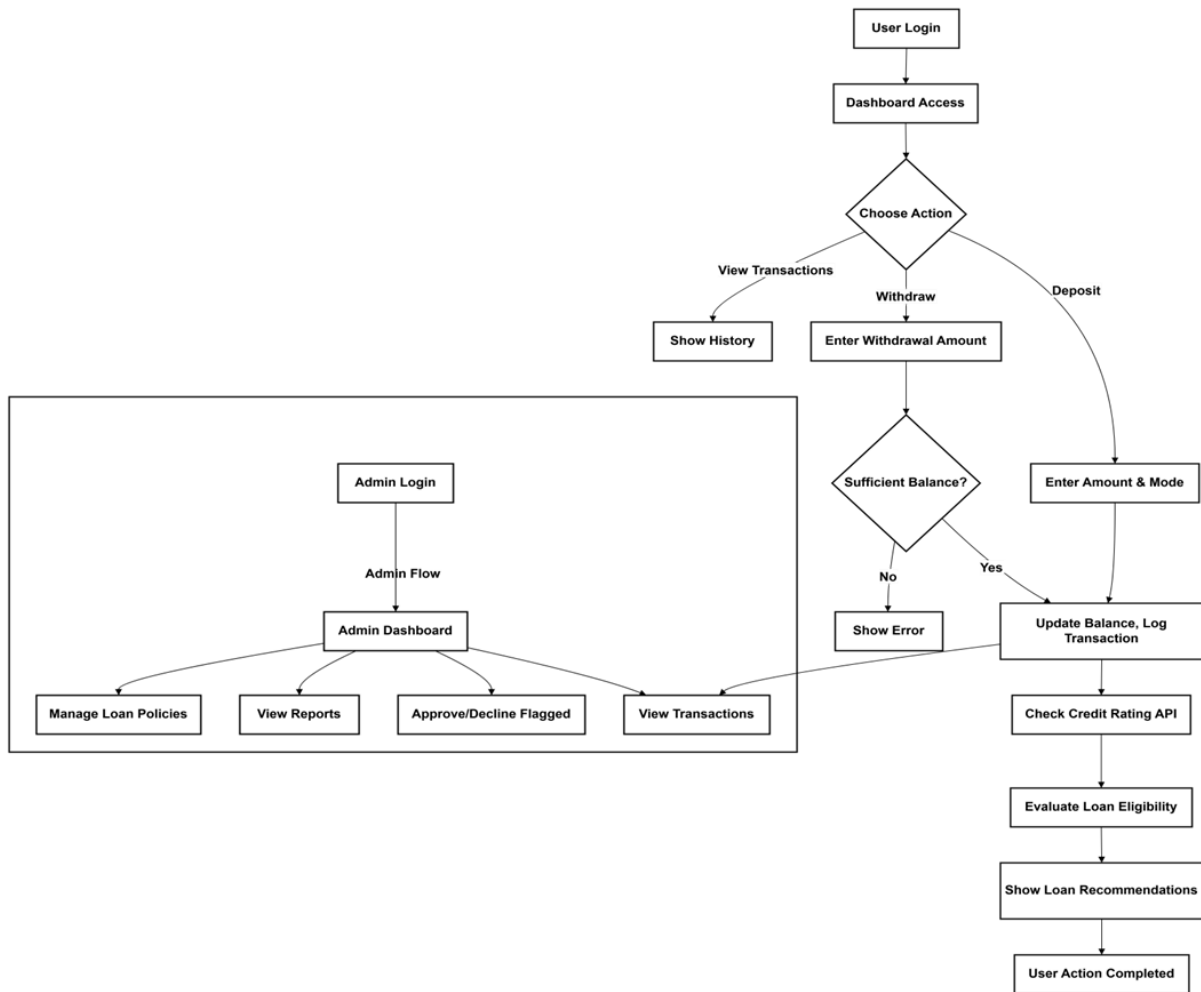
To clearly represent the design and workflow of the application, we have developed a set of diagrams and models.

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



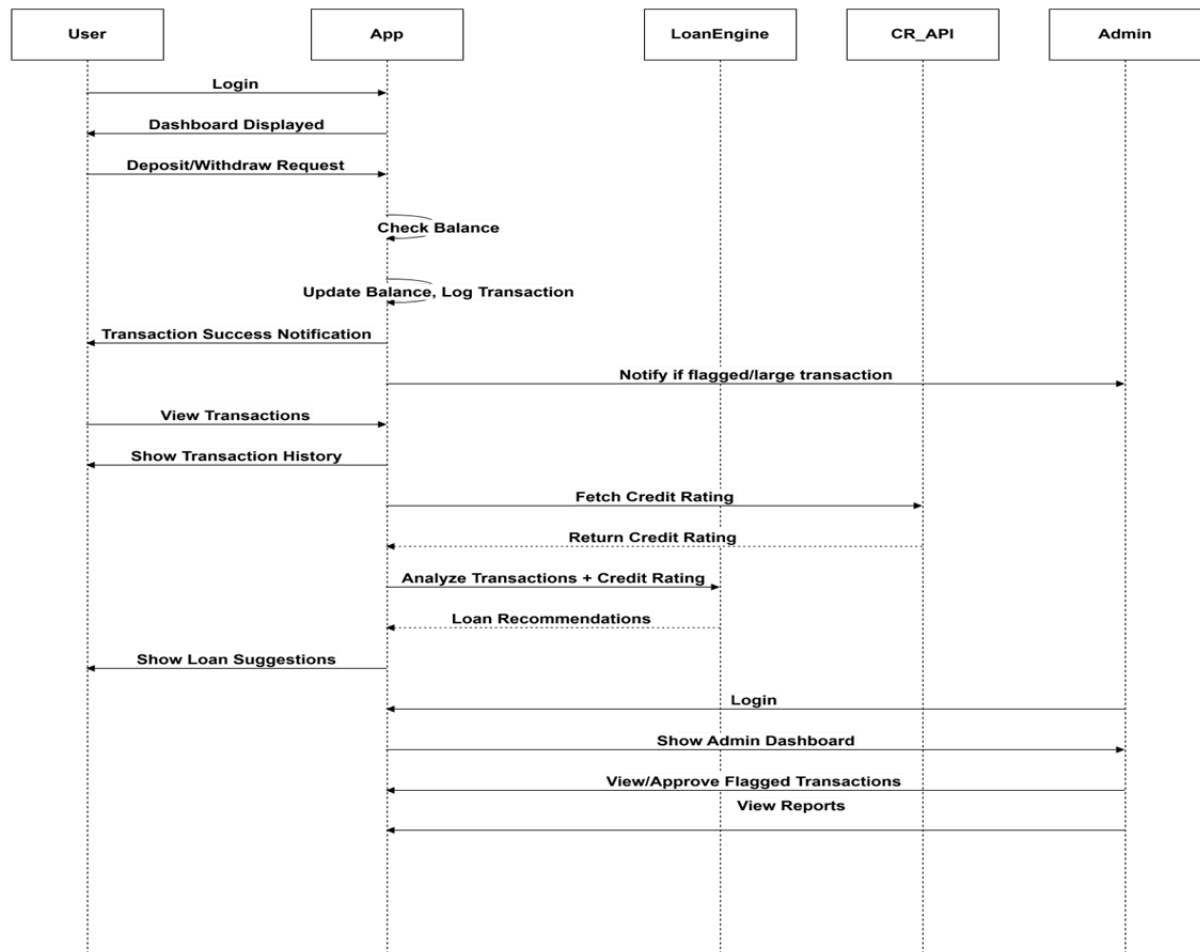
Database Design

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



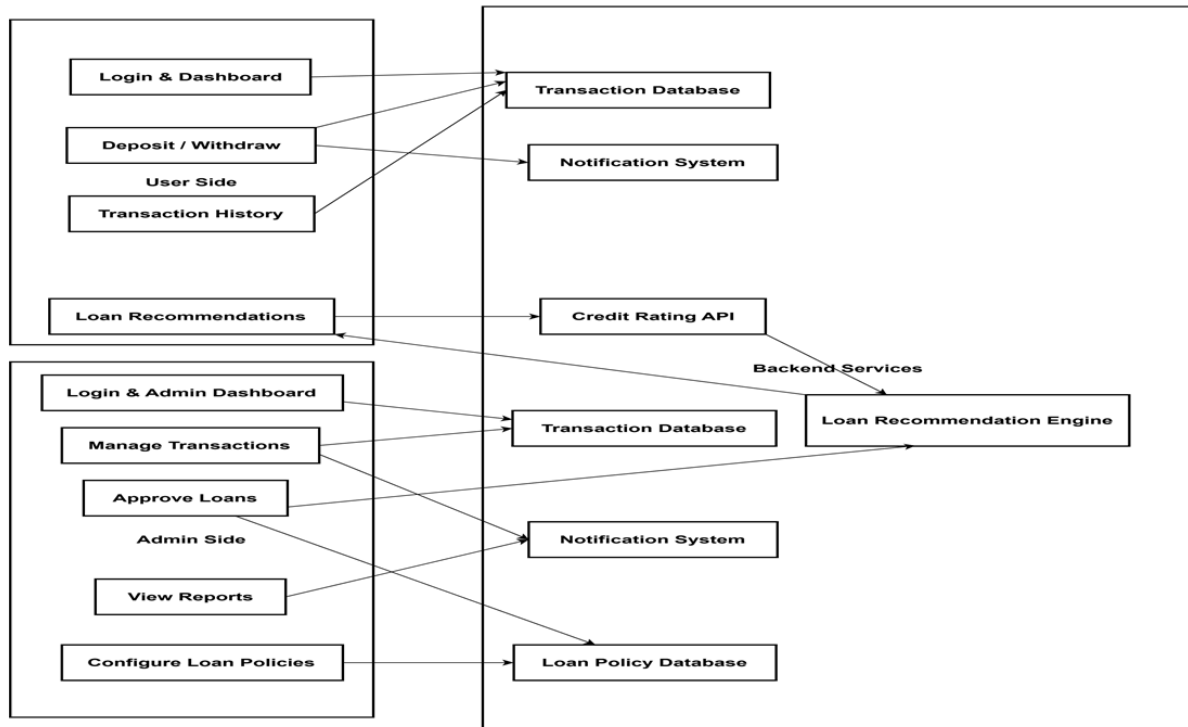
Workflow design

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



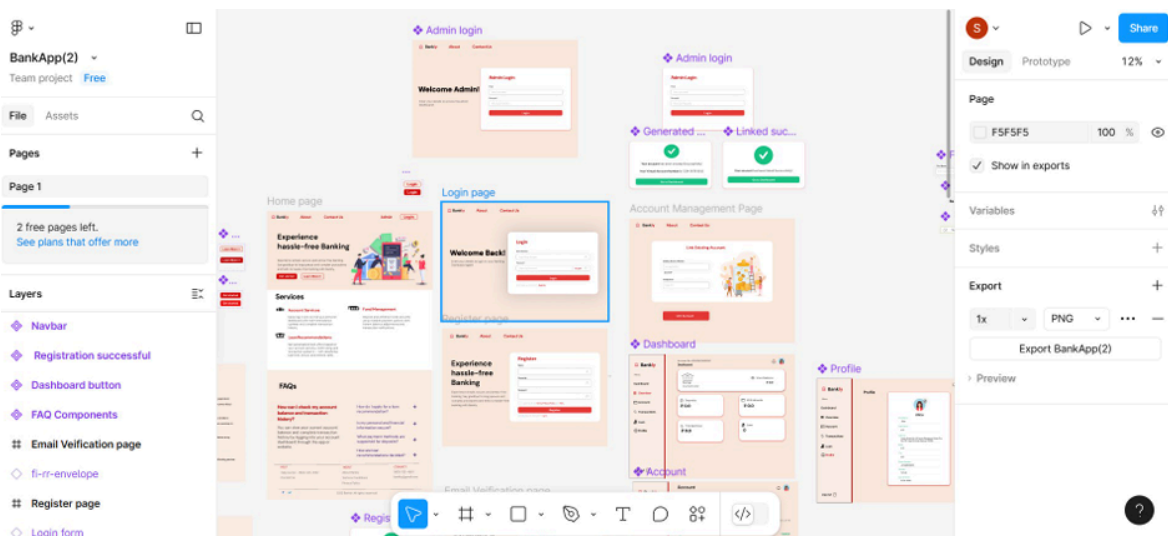
Sequence diagram

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



Functional Architecture

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

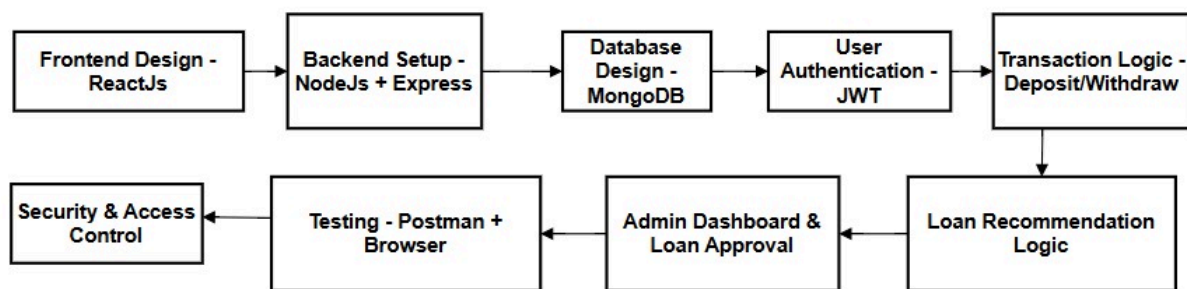


Figma Design

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's how it works:

- 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. CONCLUSION

The Banking Application developed using the MERN stack provides a secure and user-friendly platform for managing banking transactions and recommending loans based on user activity. By utilizing rule-based logic, the system ensures reliable loan suggestions without the need for complex AI models.

The project demonstrates the effectiveness of modern web technologies in building real-time, responsive applications for the financial domain. With features for both users and administrators, the system is scalable, maintainable, and ready for further enhancements to



meet evolving banking needs.

6. REFERENCES

- A. Sharifhosseini, "A Case Study for Presenting Bank Recommender Systems based on Bon Card Transaction Data", study on bank recommender systems utilizing transaction data to enhance customer experience, 2019
- Y. Lai and W. Yu, "Recommendation system based on user preference and prediction rating", explores recommendation systems that leverage user preferences and predictive ratings to enhance decision-making processes, 2021
- Roadmap.sh. (n.d.). Developer Roadmaps. <https://roadmap.sh>
- Mongoose. (n.d.). Mongoose <https://mongoosejs.com/docs/> ODM v7.6.1 Documentation.
- npmDocumentation. (n.d.). npm Docs. <http://docs.npmjs.co>