Project Title	**Loan Management System**
Technologies	MERN
Domain	Industry
Project Level	Medium

Table

Table of Contents

1. Problem Statement:

Design a web application “Loan Management System” to maintain the information of the loan customers.

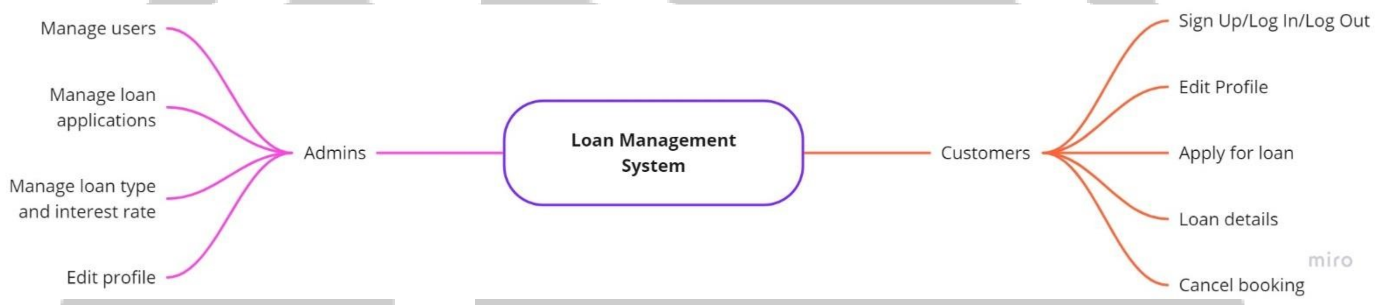
1.1. What is Loan?

A loan is when one accepts money from a bank, a friend, or another financial institution with the promise to pay it back later, along with the principal and interest. Principal is the amount borrowed, while interest is the fee for taking out the loan. Due to the risk that lenders take when they grant you a loan and their concern that you might not be able to return it, they must charge interest as a way to offset their losses.

Now that you understand what Loan is, let’s discuss some of the functionality of the “**Loan Management System**” that you will design.

1. To create a web-based system that will let the users sign up with required information.
2. After successful registration and login customers can apply for a loan.
3. Before applying for a loan customers will be able to see all the loan details (Ex: Loan period, interest, EMI, and other details).
4. After the customer has applied for a loan and it is approved, customers can track their details online.
5. Admins can see all the records of the customers who have taken the loan and can track the status online from admin dashboard.

6. Admin should have an option to approve or reject any loan application.
7. Admin should also be able to manage loans related info and user's info.



Low-Level system flow

1.2. Project Objective

- To reduce the paperwork required for manual loan application because it is inefficient and time consuming.
- To automate tedious tasks of checking the user background, verifying user identity and/or user credit history.
- Customers who have taken the loan can directly check the details online.

1.3. Functional and Non-Functional Requirements: -

1.3.1. Functional Requirements

- Admin can manage the loan type and interest rate in the system.
- Admin can see the loan applications sent by the customers and decide whether to approve the loan or reject it.
- Admin should be able to see the customers details.
- Admin should be able to manage users.
- Admin can manage loan details such as loan type, loan Amount, loan tenure, interest rate, issue date, etc., for the customer.
- Admin should be able to see the paid EMIs by the customer along with the EMI payment date.
- Users should be able to register in the system and upon successful registration they can log in to the system.

- Users after logging in can see the loan info and view interest and the loan type in the system.
- Users should be able to update their personal information.
- Users can apply for the loan and once approved they can see their loan details Ex: EMI, interest, due date, etc.,

1.3.2. Non-Functional Requirements

1. **Security:** The system should have a certain level of security such that not anybody can have access to sensitive information and the passwords should be properly encrypted.
2. **Robustness:** If the user's system crashes, a backup of the user data must be stored on remote database servers to enable recovery.
3. **Performance:** The application must be lightweight and the UI should be fast and responsive.
4. **Authentication Requirements:** There should be an authentication system in place which checks the credentials each time a user logs in.

2. Project Evaluation metrics:

2.1. Code:

- You are supposed to write code in a modular fashion
- Safe: It can be used without causing harm.
- Testable: It can be tested at the code level.
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system).
- You have to maintain your code on GitHub.
- You have to keep your GitHub repo public so that anyone can check your code.
- Proper readme file you have to maintain for any project development.
- You should include the basic workflow and execution of the entire project in the readme file on GitHub.
- Follow the coding standards.

2.2. Database:

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

2.3. API Details or User Interface:

You have to expose your complete solution as an API or try to create a user interface for your model testing. Anything will be fine for us.

2.4. Deployment:

Deploy the application on your preferred service.

2.5. Solutions Design:

You have to submit complete solution design strategies in High-level Document (HLD), Lowlevel Document (LLD), and Wireframe documents.

2.6. System Architecture:
You must submit a system architecture design in your wireframe and architecture documents.

2.7. Optimization of solutions:
Try to optimize your solution on the code level, and architecture level, and mention all of these things in your final submission.

Mention your test cases for your project.

3. Submission requirements:

3.1. High-level Document:
You have to create a high-level document design for your project. You can reference the HLD form below the link.

Sample link: HLD Document Link

3.2. Low-level document:
You have to create a Low-level document design for your project; you can refer to the LLD from the link below.

Sample link: LLD Document Link

3.3. Architecture:

You have to create an Architecture document design for your project; you can refer to the Architecture from the link below.

Sample link: Architecture sample link

3.4. Wireframe:

You have to create a Wireframe document design for your project; refer to the Wireframe from the link below.

Demo link: Wireframe Document Link

3.5. Project code:

You have to submit your code to the GitHub repo in your dashboard when the final submission of your project.

Demo link: Project code sample link

3.6. Detail project report:

You have to create a detailed project report and submit that document as per the given sample.

Demo link: DPR sample link

3.7. Project demo video:

You have to record a project demo video for at least 5 Minutes and submit that link as per the given demo.

3.8. The project LinkedIn a post:

You have to post your project details on LinkedIn and submit that post link in your dashboard in your respective field.