

CHUBB®

**CAPSTONE PROJECT
(Loan Management System)**

Enterprise-Grade Secure Full-Stack Web Application

**-Riddhima Bhanja
Kalinga Institute of Industrial Technology**

INDEX

Content	Page No.
1. JACOCO Code Coverage Report	1
2. SonarQube Report & Issues	3
3. System Architecture, ER Diagram	5
4. JMeter & RabbitMQ dashboard	6
— JMeter Result Tree	6
— JMeter Summary Report	6
— Apache JMETER Dashboard	7
5. Logs	9
— RabbitMQ Dashboard	9
— Eureka server, Booking Service	10
— Flight Service, API Gateway, Notification service	10
6. MongoDB Screenshots	11
7. Postman Screenshots	11
— Circuit Breaker, Message broker	11
— — JWT security through API Gateway	12
— — Status,Events	12
— — Health Checks	13
— — API Gateway Health	13
— — Booking Service Health	13
— — Eureka Server Health	14
— — Flight Service Health	14
— Add Flight	15
— Book Flight, get flight by PNR, get flight by ID	15
— Cancel Booking, FALBACK CASE	15
— Get Booking by PNR	16
— Get Booking History	16
8. Email	17
— With PDF	17
— Without PDF	17
— Fallback Case	17
9. Eureka Dashboard, MySQL Workbench	17

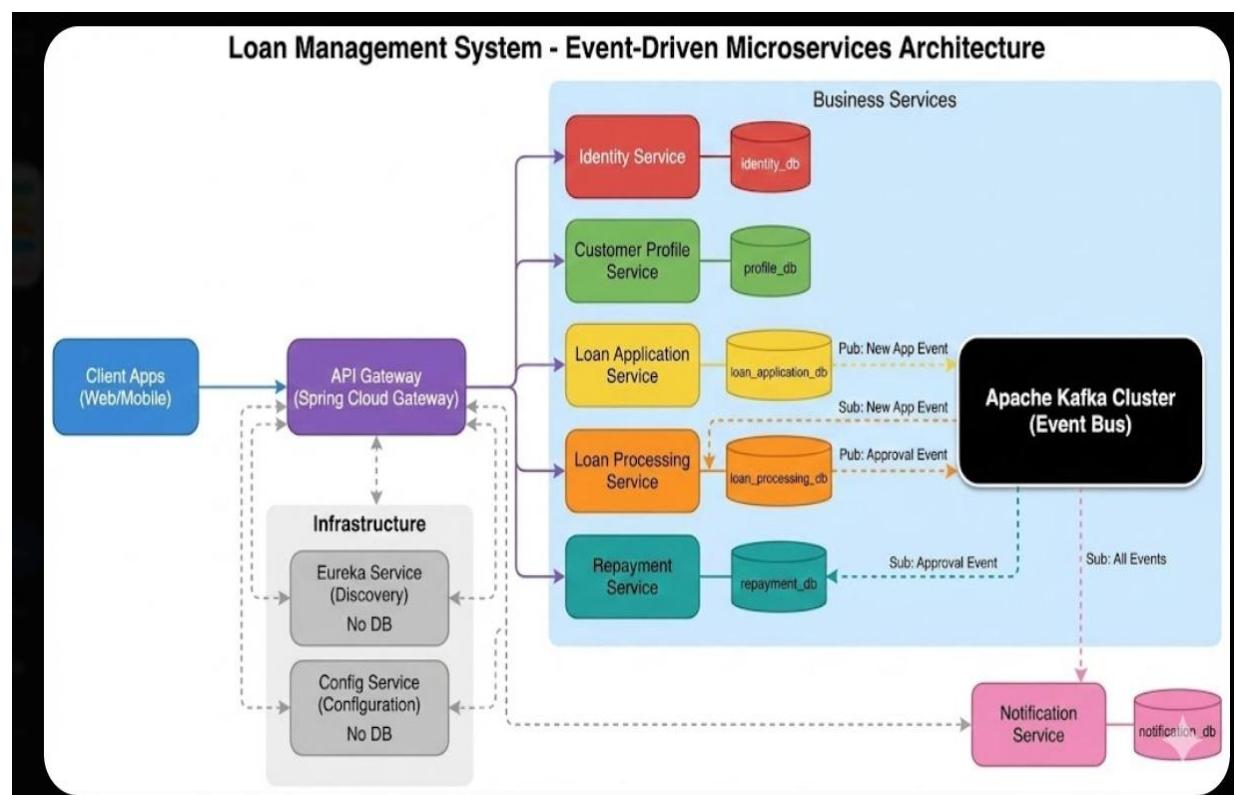
1) Purpose of the document

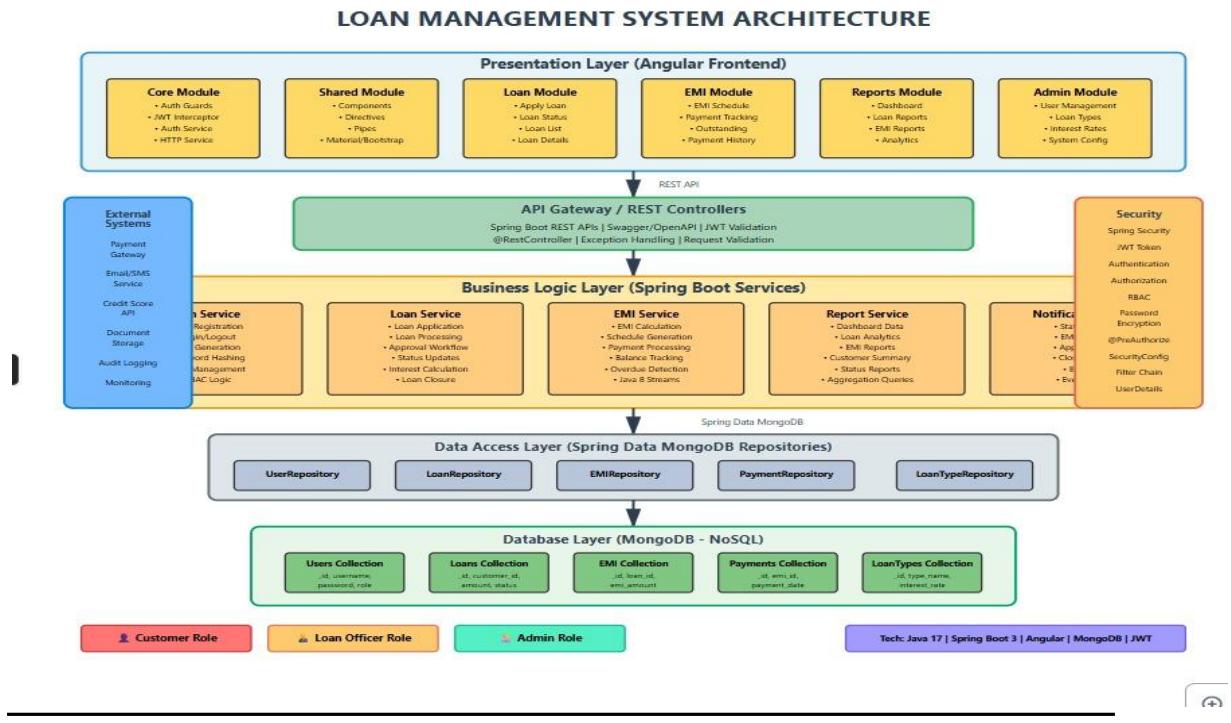
This document serves as the **authoritative technical and architectural reference** for the **Loan Management System (LMS)**.

Its primary objectives are to:

- Explain the **business rationale and system vision**
- Provide a **high-level system overview** for stakeholders
- Describe the **architectural style, core components, and design decisions**
- Detail the **microservice responsibilities, APIs, and data ownership**
- Define **security, error handling, validation, and non-functional requirements**
- Present a **robust testing and quality assurance strategy**

System Overview





2.1 Business Objective

The Loan Management System is designed to **digitize and automate the end-to-end loan lifecycle** for banks, NBFCs, and financial institutions. The system replaces manual and semi-automated workflows with a **secure, scalable, and audit-ready platform**.

Key business goals include:

- Faster loan application processing
- Transparent approval workflows
- Accurate EMI computation and repayment tracking
- Secure handling of sensitive financial data
- Real-time dashboards and operational reports
- Cloud-ready, API-first architecture

2.2 High-Level Features

- Multi-role system: **Admin, Loan Officer, Customer**
- Online loan application and tracking

- Configurable loan types, interest rates, and tenures
- Automated EMI schedule generation
- Repayment tracking and loan closure
- Secure authentication using JWT
- Role-based authorization (RBAC)
- Reporting and dashboard analytics
- Production-ready REST APIs

EUREKA DASHBOARD

The screenshot shows a web browser window for the Eureka Dashboard at localhost:8761. At the top, a red warning message reads: "THE SELF PRESERVATION MODE IS TURNED OFF. THIS MAY NOT PROTECT INSTANCE EXPIRY IN CASE OF NETWORK/OTHER PROBLEMS." Below this, under "DS Replicas", there is a table titled "Instances currently registered with Eureka". The table has columns: Application, AMIs, Availability Zones, and Status. The data is as follows:

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - api-gateway:8080
AUTH-SERVICE	n/a (1)	(1)	UP (1) - auth-service:8081
EMI-SERVICE	n/a (1)	(1)	UP (1) - EMI-SERVICE:8084
LOAN-APPLICATION-SERVICE	n/a (1)	(1)	UP (1) - LOAN-APPLICATION-SERVICE:8082
LOAN-APPROVAL-SERVICE	n/a (1)	(1)	UP (1) - loan-approval-service:8083
NOTIFICATION-SERVICE	n/a (1)	(1)	UP (1) - notification-service:8086
REPORTING-SERVICE	n/a (1)	(1)	UP (1) - b70aac59ae1e:REPORTING-SERVICE:8085

Below the table is a section titled "General Info" with a table showing system metrics:

Name	Value
total-avail-memory	96mb
num-of-cpus	8

HOME PAGE

The screenshot shows a web browser window for the Loan Pro home page at localhost:4200/home. The background features a collage of US dollar bills and a pocket watch. In the center, there is a logo consisting of a blue square with a white arrow pointing down-right, followed by the text "Loan Pro" and the tagline "Your Gateway to Financial Freedom". Below this, a sub-tagline reads: "Fast, secure, and hassle-free loans tailored to your needs. Experience the future of lending with our advanced loan management system." At the bottom, there are two buttons: a purple "LOGIN" button with a user icon and a pink "REGISTER NOW" button with a plus-user icon.

Why Choose Us?

 **Quick Loan Processing**
Get your loan approved in just 24 hours with our streamlined process

 **Secure & Trusted**
Bank-level security with encrypted transactions and data protection

 **Competitive Rates**
Enjoy the lowest interest rates in the market with flexible repayment options

 **24/7 Support**
Our dedicated team is always available to assist you with any queries

10K+
HAPPY CUSTOMERS

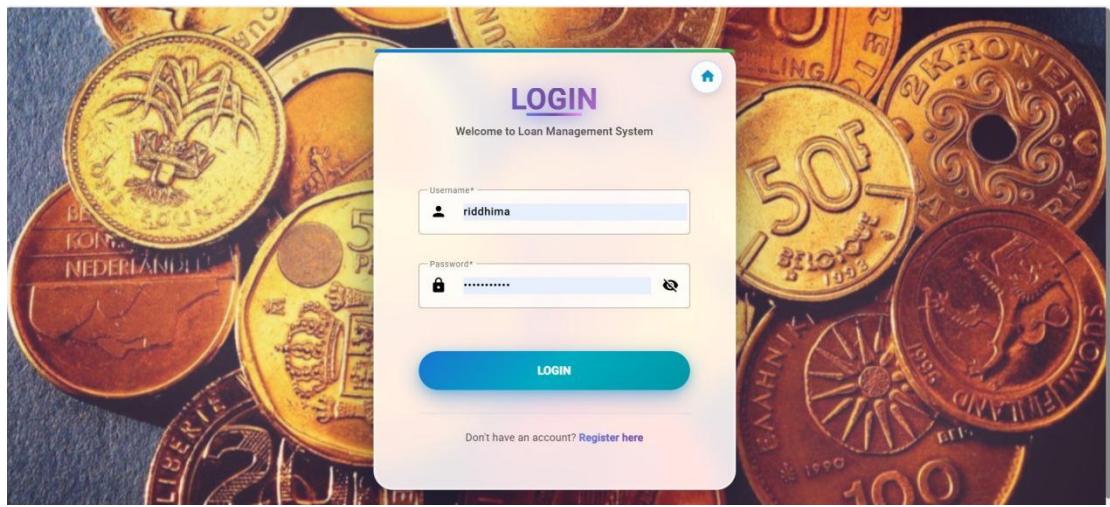
₹500Cr+
LOANS DISBURSED

98%
APPROVAL RATE

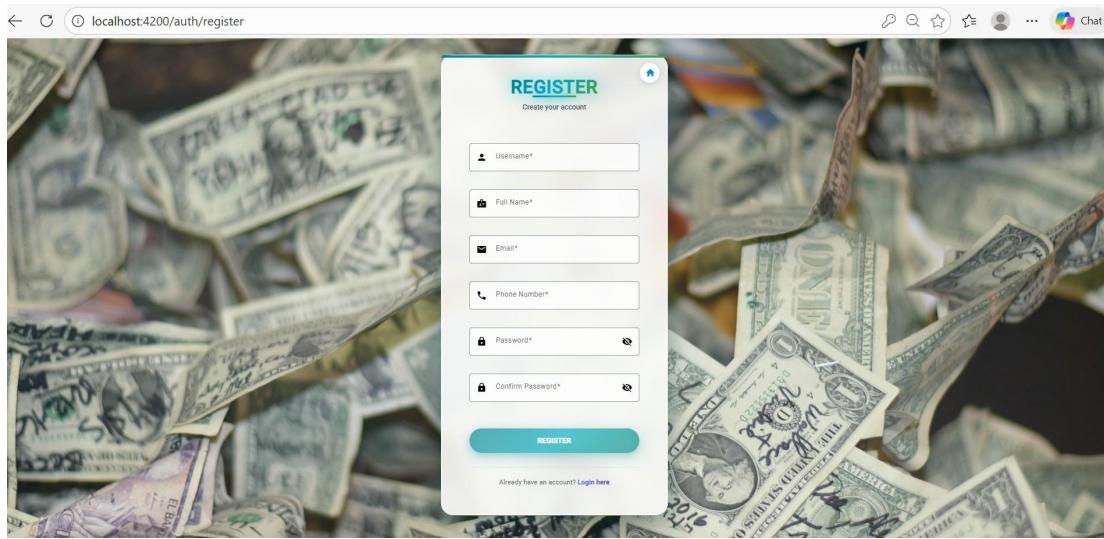
24hrs
QUICK PROCESSING

© 2024 LoanPro Management System. All rights reserved.
Empowering Financial Dreams

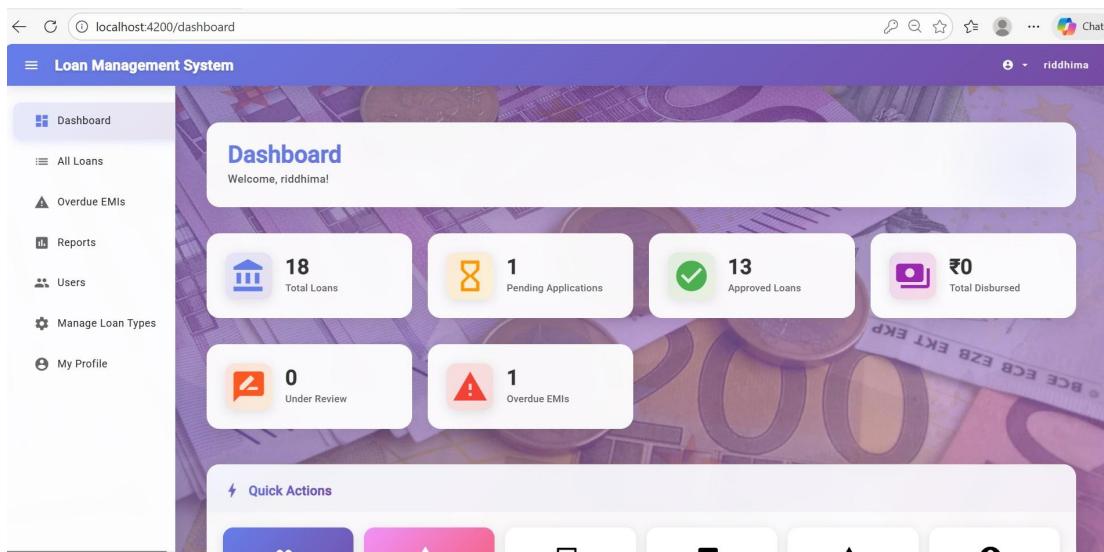
LOGIN PAGE



REGISTER PAGE



ADMIN DASHBOARD



The screenshot shows the 'Loan Management System' dashboard. On the left, a sidebar menu includes: Dashboard, All Loans, Overdue EMIs, Reports, Users, Manage Loan Types, and My Profile. The main area features a 'Quick Actions' bar with buttons for 'Manage Users' (purple), 'Loan Types' (pink), 'All Loans' (grey), 'Reports' (white), 'Overdue EMIs' (white), and 'My Profile' (white). Below this is a section titled 'Recent Applications' with a table:

APPLICATION #	CUSTOMER	LOAN TYPE	AMOUNT	STATUS	APPLIED DATE
	seema		₹9,00,000	PENDING	6 Jan 2026
	seema		₹2,00,000	APPROVED	6 Jan 2026
	seema		₹9,00,000	REJECTED	6 Jan 2026
	seema		₹90,000	APPROVED	6 Jan 2026

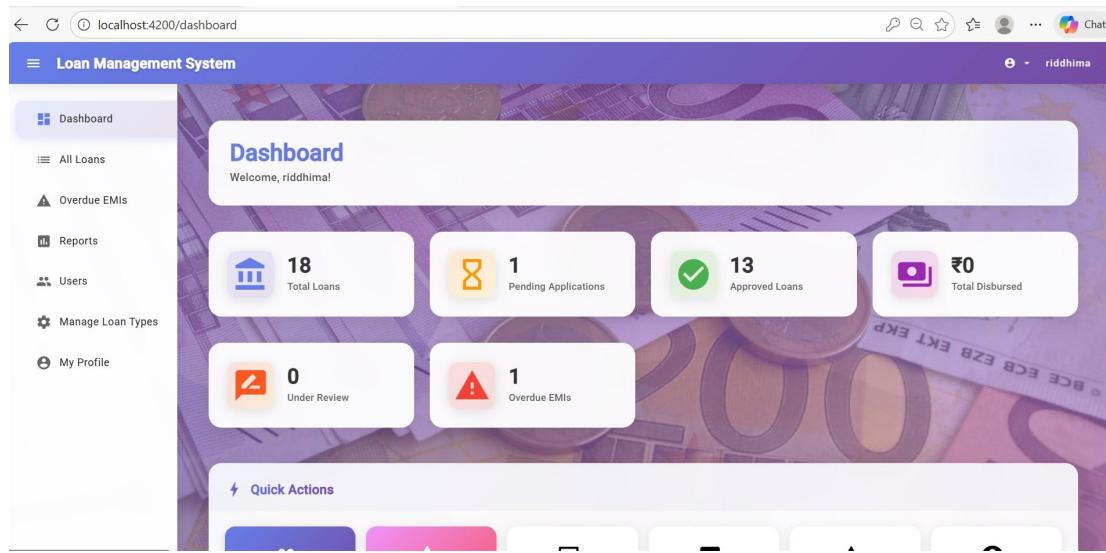
CUSTOMER DASHBOARD

The screenshot shows the 'Loan Management System' dashboard. The sidebar menu is identical to the previous one. The main area features a 'Dashboard' section with a welcome message 'Welcome, riddhima!' and several statistics:

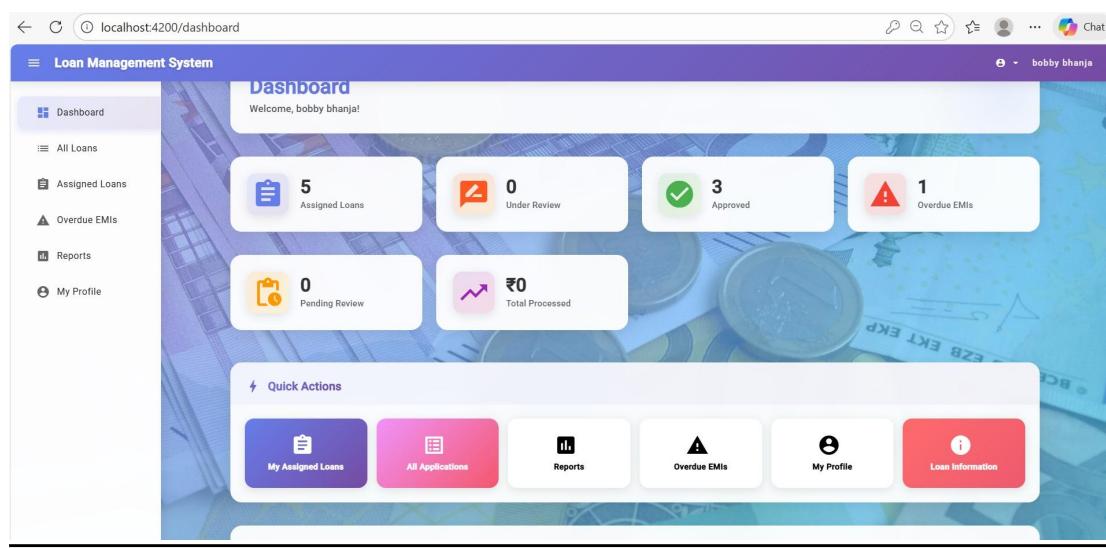
- Total Loans: 18
- Pending Applications: 1
- Approved Loans: 13
- Total Disbursed: ₹0
- Under Review: 0
- Overdue EMIs: 1

Below this is a 'Quick Actions' bar with buttons for 'Manage Users' (purple), 'Loan Types' (pink), 'All Loans' (grey), 'Reports' (white), 'Overdue EMIs' (white), and 'My Profile' (white).

CUSTOMER EMI SCHEDULE



LOAN OFFICER DASHBOARD



The screenshot shows the 'Loan Management System' dashboard. On the left, a sidebar menu includes 'Dashboard', 'All Loans', 'Assigned Loans', 'Overdue EMIs', 'Reports', and 'My Profile'. A 'Quick Actions' bar at the top right contains five buttons: 'My Assigned Loans' (purple), 'All Applications' (pink), 'Reports' (white), 'Overdue EMIs' (white with red triangle), 'My Profile' (white with user icon), and 'Loan Information' (red). The main content area displays a table titled 'Recent Assignments' with columns: APPLICATION #, CUSTOMER, LOAN TYPE, AMOUNT, STATUS, and APPLIED DATE. The data is as follows:

APPLICATION #	CUSTOMER	LOAN TYPE	AMOUNT	STATUS	APPLIED DATE
	seema		₹60,000	REJECTED	4 Jan 2026
	seema		₹9,00,000	DISBURSED	6 Jan 2026
	seema		₹90,00,000	APPROVED	6 Jan 2026
	seema		₹9,00,000	REJECTED	6 Jan 2026
	seema		₹90,000	APPROVED	6 Jan 2026

PER LOAN CUSTOMER DETAILS

The screenshot shows the 'Loan Application - LN000018' page. At the top, it says 'View and manage loan application details'. There are two main sections: 'Basic Information' and 'Customer Information'. In 'Basic Information', fields include APPLICATION NUMBER: LN000018, STATUS: Pending, LOAN TYPE: Education Loan, and APPLIED DATE: 06/01/2026 22:02. In 'Customer Information', fields include NAME: Customer #11, EMPLOYMENT STATUS: Self Employed, and MONTHLY INCOME: ₹80,00,000.00. Below these, under 'Loan Details', REQUESTED AMOUNT is ₹9,00,000.00, TENURE is 90 months, and PURPOSE is home maintenance.

ALL LOAN TYPES CUSTOMER VIEW

The screenshot displays the 'Loan Management System' interface. On the left, a sidebar menu includes 'Dashboard', 'Loan Types' (selected), 'Apply for Loan', 'My Loans', 'My EMI Schedule', and 'My Profile'. The main area shows six loan categories:

- HOME LOAN**: Active. Description: Loan for purchasing or constructing residential property. Lower interest rates with longer tenure options. Details: Loan Amount ₹5,000 - ₹5,00,00,000, Tenure Period 60 - 360 months, Interest Rate 8.5% per annum. Buttons: Apply Now, Calculate EMI.
- Personal Loan**: Active. Description: Unsecured personal loan for any personal needs like medical expenses, home renovation, wedding, vacation, etc. Details: Loan Amount ₹50,000 - ₹25,00,000, Tenure Period 12 - 60 months, Interest Rate 10.5% per annum. Buttons: Apply Now, Calculate EMI.
- Car Loan**: Active. Description: Loan for purchasing new or used vehicles. Quick approval with competitive rates. Details: Loan Amount ₹1,00,000 - ₹50,00,000, Tenure Period 12 - 84 months, Interest Rate 9.25% per annum. Buttons: Apply Now, Calculate EMI.
- Education Loan**: Active. Description: Education loan for higher studies in India or abroad. Covers tuition fees, accommodation, and related. Buttons: Apply Now, Calculate EMI.
- Business Loan**: Active. Description: Working capital and business expansion loan for MSMEs and entrepreneurs. Buttons: Apply Now, Calculate EMI.
- Gold Loan**: Active. Description: for purchasing gold and investing in it. Buttons: Apply Now, Calculate EMI.

PAYMENT RECORDS

The screenshot shows the 'EMI Schedule - LN000001' page. At the top, it says 'View complete EMI payment schedule'. Below is a table of EMI payments:

EMI	Date	Amount
1	04 Jan 2026	₹9,932.82
2	05 Feb 2026	₹10,032.15
3	05 Mar 2026	₹10,132.47
4	05 Apr 2026	₹10,233.79
5	05 May 2026	₹10,336.13
6	05 Jun 2026	₹10,439.49

A modal window titled 'Record EMI Payment' is open, prompting for 'Payment Amount*' (₹ 19932.82) and 'Payment Date*' (1/7/2026). A dropdown menu for 'Payment Method*' lists 'Cash', 'Cheque', 'NEFT', 'RTGS', 'UPI', and 'Debit Card'. To the right, a summary shows 'PENDING EMIS 68' with a grid of payment status cards: Paid (green), Pending (orange).

OVERDUE EMI

The screenshot shows a web-based dashboard titled "Overdue EMIs". At the top, it displays four key metrics: "TOTAL OVERDUE EMIS" (1), "OVERDUE AMOUNT" (₹19,932.82), "TOTAL OUTSTANDING" (₹8,85,108.72), and "AFFECTED CUSTOMERS" (1). Below these, a red banner reads "Attention Required! 1 EMI payment(s) are overdue. Please take immediate action to follow up with customers." A table lists the single overdue EMI: LOAN # 1, CUSTOMER Customer #11, EMI # 11, DUE DATE 01 Jan 2026, DAYS OVERDUE 5 days RECENT, EMI AMOUNT ₹19,932.82, and OUTSTANDING ₹8,85,108.72. Actions for this row include a magnifying glass icon and a calendar icon.

NEXT EMI SCHEDULE

The screenshot shows a web-based dashboard titled "Loan Management System". On the left, a sidebar lists navigation options: Dashboard, Loan Types, Apply for Loan, My Loans, My EMI Schedule, and My Profile. The main area features a "Next EMI Payment" section with a due date of "1 Jan 2026", an amount of "₹19,932.82", and a payment method of "Auto-Debit / Pay Now". A "View Full EMI Schedule" button is present. Below this, a "Quick Actions" section includes buttons for "Apply for Loan" (purple), "Browse Loan Types" (pink), "My Loans" (white), "My EMI Schedule" (white), and "My Profile" (white).

LOAN SUMMARY PER CUSTOMER

localhost:4200/loans/18/review

Review Loan Application
Application: LN000018

Loan Summary

CUSTOMER:	LOAN TYPE:	REQUESTED AMOUNT:
Customer #11	Education Loan	₹9,00,000.00
TENURE:	EMPLOYMENT:	MONTHLY INCOME:
90 months	SELF_EMPLOYED	₹80,00,000.00
PURPOSE:		
home maintenance		

Approve Application

Approved Amount*
₹ 900000

CUSTOMER PROFILE

Account Information

USERNAME seema	ROLES CUSTOMER	ACCOUNT STATUS ACTIVE
-------------------	-------------------	--------------------------

MEMBER SINCE
5 January 2026 at 05:25 pm

Personal Information

Edit Profile

Full Name* seema
Email Address* mousumi.bhanja.31@gmail.com
Phone Number* 1234567890
Enter 10-digit mobile number

Security Notes

CUSTOMER EMI SCHEDULE

The screenshot shows a web application titled "My EMI Schedule". At the top, there are four summary boxes: "TOTAL LOANS 5", "PAID AMOUNT ₹41,763.75", "PENDING EMIS 204", and "PENDING AMOUNT ₹1,24,18,358.15". Below this is a navigation bar with "All EMIs" and "Pending Only (204)". A detailed table follows, showing a single loan entry for "Loan #1" with 70 total EMIs. The table includes columns for ID, Date, Amount, Interest, Total Amount, Status, and Actions. The first two rows are marked as "Paid", while the third row is marked as "Pending".

CREATE CUSTOMER

The screenshot shows a "User Management" page where users can "Manage system users and roles". On the left, a list of existing users is displayed with columns for ID, Name, and Role. On the right, a modal window titled "Create New User" is open, containing fields for "Full Name*", "Email*", "Phone Number*", and "Password*". To the right of the modal, a grid of user status icons (Inactive, Active) is shown, with one icon having a checked checkbox.

CREATE LOAN

The screenshot shows a 'Create New Loan Type' dialog box over a background of a loan management interface. The dialog has a 'Description*' field (placeholder: 'Loan type name is required'), 'Minimum Amount*' and 'Maximum Amount*' fields, 'Interest Rate (% per annum)*' field (placeholder: '%'), and 'Cancel' and 'Create' buttons. In the background, there's a table of loan types like 'HOME LOAN' and 'Personal Loan', and a sidebar with interest rate options (8.5% p.a., 10.5% p.a., etc.) and their status (Active).

APPROVE REJECT LOAN

The screenshot shows two modal dialogs. The top one is for 'Approve Application' with fields for 'Approved Amount*' (₹ 900000), 'Interest Rate (per annum)*' (12%), and 'Approval Remarks (Optional)'. The bottom one is for 'Reject Application' with fields for 'Rejection Reason*' and 'Additional Notes (Optional)'. Both dialogs have 'Approve Loan' and 'Reject Loan' buttons respectively.

APPLY FOR LOAN AS CUSTOMER

The screenshot shows the 'Apply for Loan' page of the 'Loan Management System'. The top navigation bar includes a back arrow, a search icon, a user profile, and a 'Chat' button. The left sidebar has a purple header 'Loan Management System' and links for Dashboard, Loan Types, Apply for Loan (which is highlighted in blue), My Loans, My EMI Schedule, and My Profile. The main content area has a purple background featuring a clock and banknotes. The title 'Apply for Loan' is at the top, with a 'Back to Dashboard' link. Below it is a three-step process: 1. Select Loan Type, 2. Loan Details, 3. Employment Details. Step 1 is active, showing a dropdown menu with 'Personal Loan (₹50,000 - ₹25,00,000)' selected. A 'Choose Your Loan Type' section provides details for Personal Loan: 'Unsecured personal loan for any personal needs like medical expenses, home renovation, wedding, vacation, etc.' It also lists 'AMOUNT RANGE ₹50,000 - ₹25,00,000', 'TENURE RANGE 12 - 60 months', and 'INTEREST RATE 10.5% per annum'. A 'Next' button is at the bottom right.

ALL LOANS(ADMIN)

This screenshot is identical to the one above, showing the 'Apply for Loan' page for an administrator. The layout, sidebar, and content are the same, including the purple header 'Loan Management System', the 'Apply for Loan' link in the sidebar, and the detailed information about the Personal Loan product.

ALL APPROVED/REJECTED LOANS

All Approved/Rejected Loans						
LN000006	Customer #11	Car Loan	₹8,00,000.00	Approved	06/01/2026	⋮
LN000007	Customer #11	Car Loan	₹7,00,000.00	Approved	06/01/2026	⋮
LN000008	Customer #11	Education Loan	₹1,00,000.00	Rejected	06/01/2026	⋮
LN000009	Customer #11	Car Loan	₹9,00,000.00	Rejected	06/01/2026	⋮
LN000010	Customer #11	Education Loan	₹1,00,000.00	Rejected	06/01/2026	⋮
LN000011	Customer #11	Personal Loan	₹60,000.00	Rejected	04/01/2026	⋮
LN000012	Customer #11	Personal Loan	₹9,00,000.00	Disbursed	06/01/2026	⋮
LN000013	Customer #11	Business Loan	₹90,00,000.00	Approved	06/01/2026	⋮
LN000014	Customer #11	Car Loan	₹9,00,000.00	Rejected	06/01/2026	⋮

ADMIN PERSONAL INFO CARD

localhost:4200/loan-types

Loan Management System

- Dashboard
- Loan Types
- + Apply for Loan
- My Loans
- My EMI Schedule
- My Profile

HOME LOAN Active

Loan for purchasing or constructing residential property. Lower interest rates with longer tenure options.

Loan Amount: ₹5,000 - ₹5,00,00,000

Tenure Period: 60 - 360 months

Interest Rate: 8.5% per annum

Apply Now **Calculate EMI**

PERSONAL LOAN Active

Unsecured personal loan for any personal needs like medical expenses, home renovation, wedding, vacation, etc.

Loan Amount: ₹50,000 - ₹25,00,000

Tenure Period: 12 - 60 months

Interest Rate: 10.5% per annum

Apply Now **Calculate EMI**

CAR LOAN Active

Loan for purchasing new or used vehicles. Quick approval with competitive rates

Loan Amount: ₹1,00,000 - ₹50,00,000

Tenure Period: 12 - 84 months

Interest Rate: 9.25% per annum

Apply Now **Calculate EMI**

EDUCATION LOAN Active

Education loan for higher studies in India or abroad. Covers tuition fees, accommodation, and related

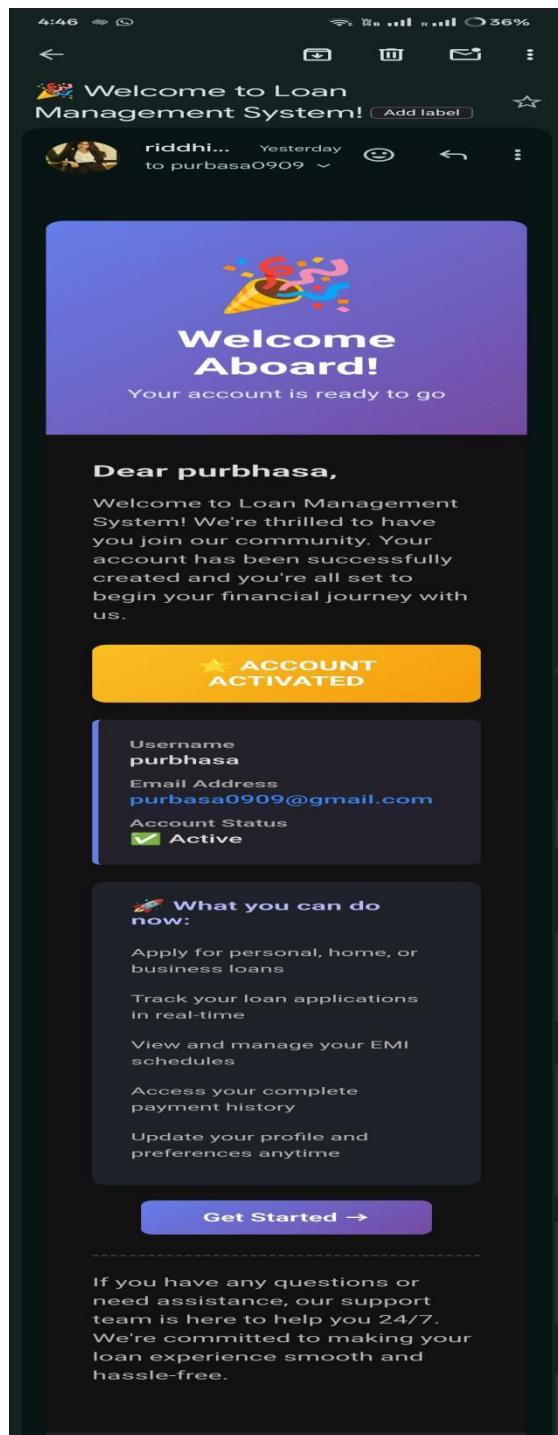
BUSINESS LOAN Active

Working capital and business expansion loan for MSMEs and entrepreneurs

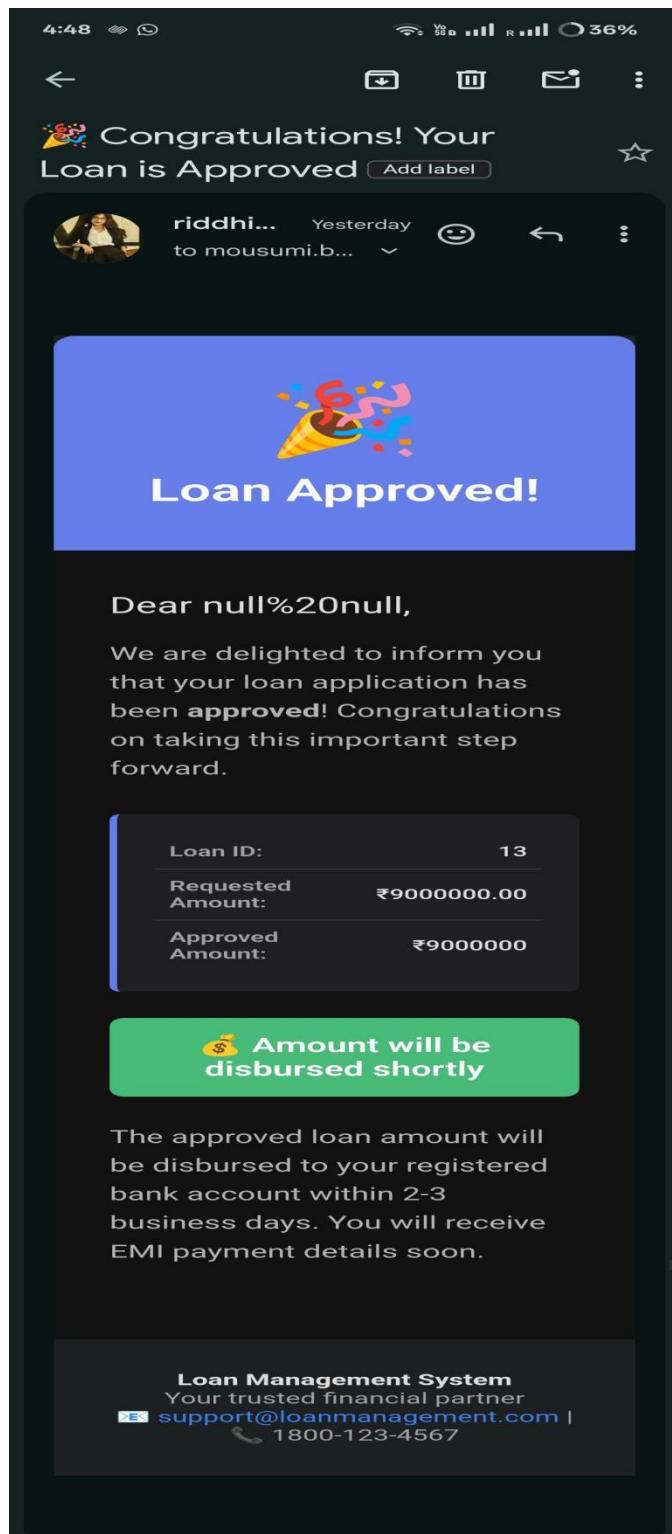
GOLD LOAN Active

for purchasing gold and investing in it

LOAN REGISTRATION EMAIL



LOAN APPROVAL EMAIL



LOAN REJECTION EMAIL

The screenshot shows a mobile phone interface with a dark theme. At the top, the status bar displays the time (6:14), signal strength, battery level (29%), and other icons. Below the status bar is the navigation bar with standard icons for back, home, recent apps, and settings.

The main screen is titled "Loan Application Update". It features a recipient's profile picture and name ("riddhi... Yesterday"), the recipient's name ("to mousumi.b..."), and a reply icon. There are also icons for adding a label and marking the email as important.

A large red rectangular graphic in the center contains a clipboard icon and the text "Application Update".

The body of the email begins with "Dear null%20null," followed by a message of rejection:

Thank you for your interest in our loan services. After careful review of your application, we regret to inform you that we are unable to approve your loan request at this time.

Below this, there is a summary table:

Loan ID:	14
Application Date:	Today

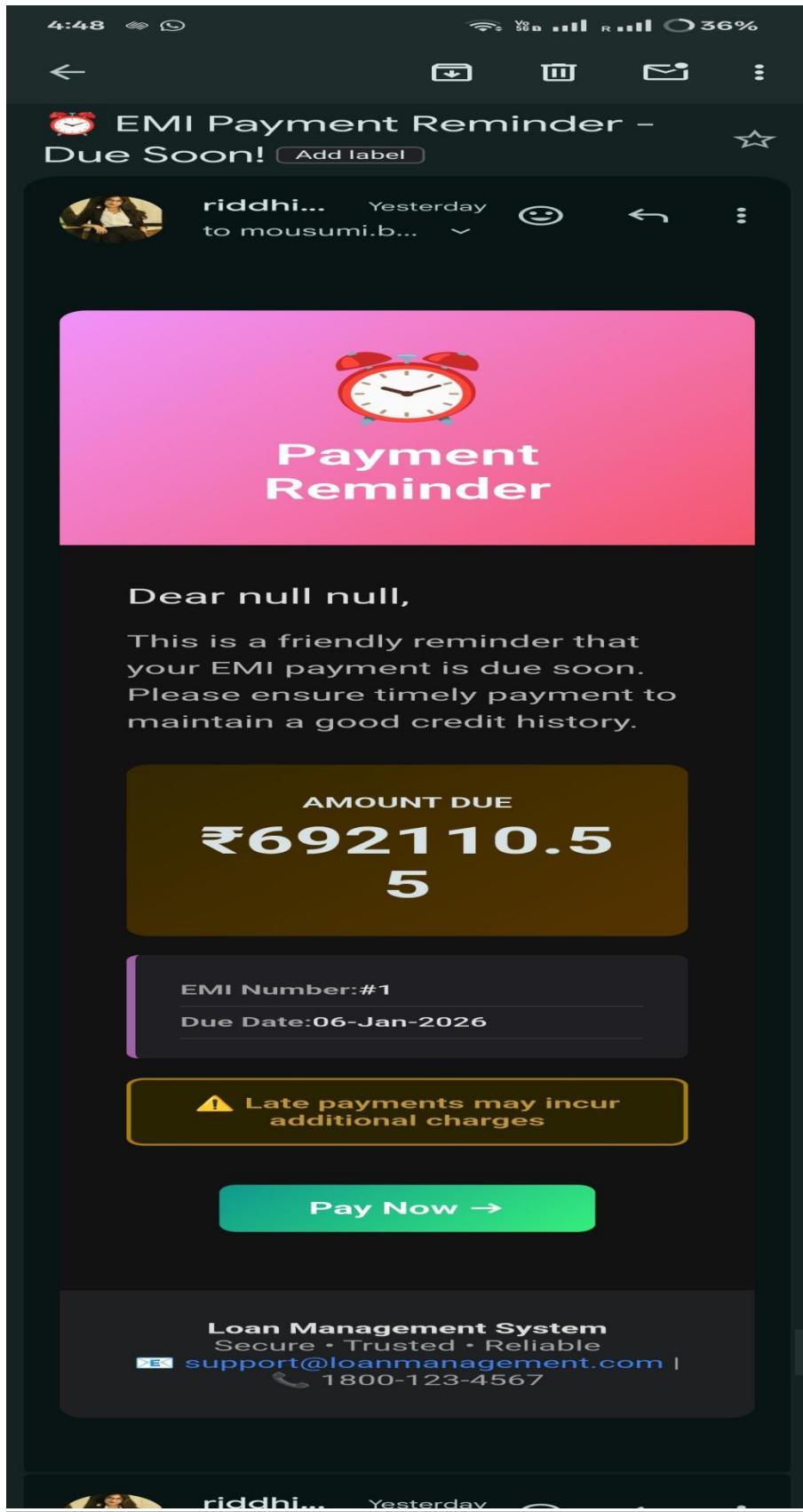
Following the table is a yellow-bordered box containing the reason for rejection:

Reason for Decision:
document%20verification%20failed

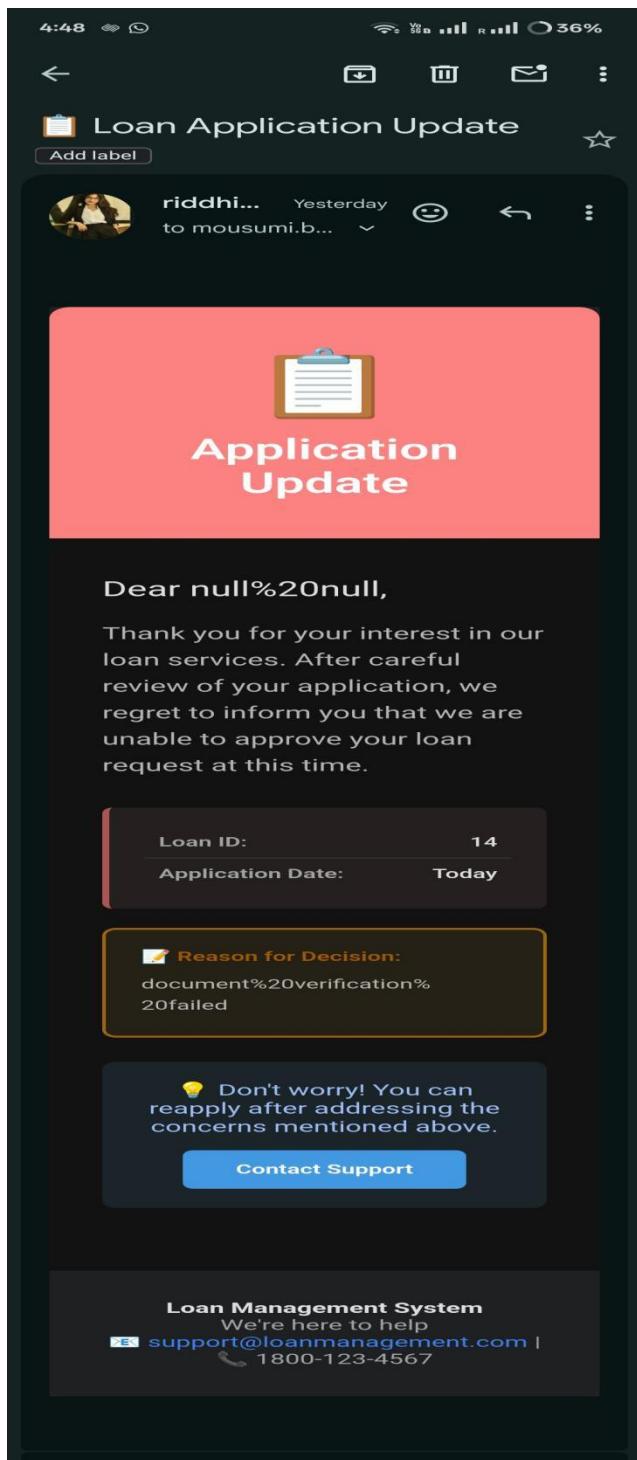
At the bottom, there is a blue button with white text:

💡 Don't worry! You can reapply after addressing the concerns mentioned above.
Contact Support

EMI PAYMENT REMINDER



APPLICATION UPDATE EMAIL



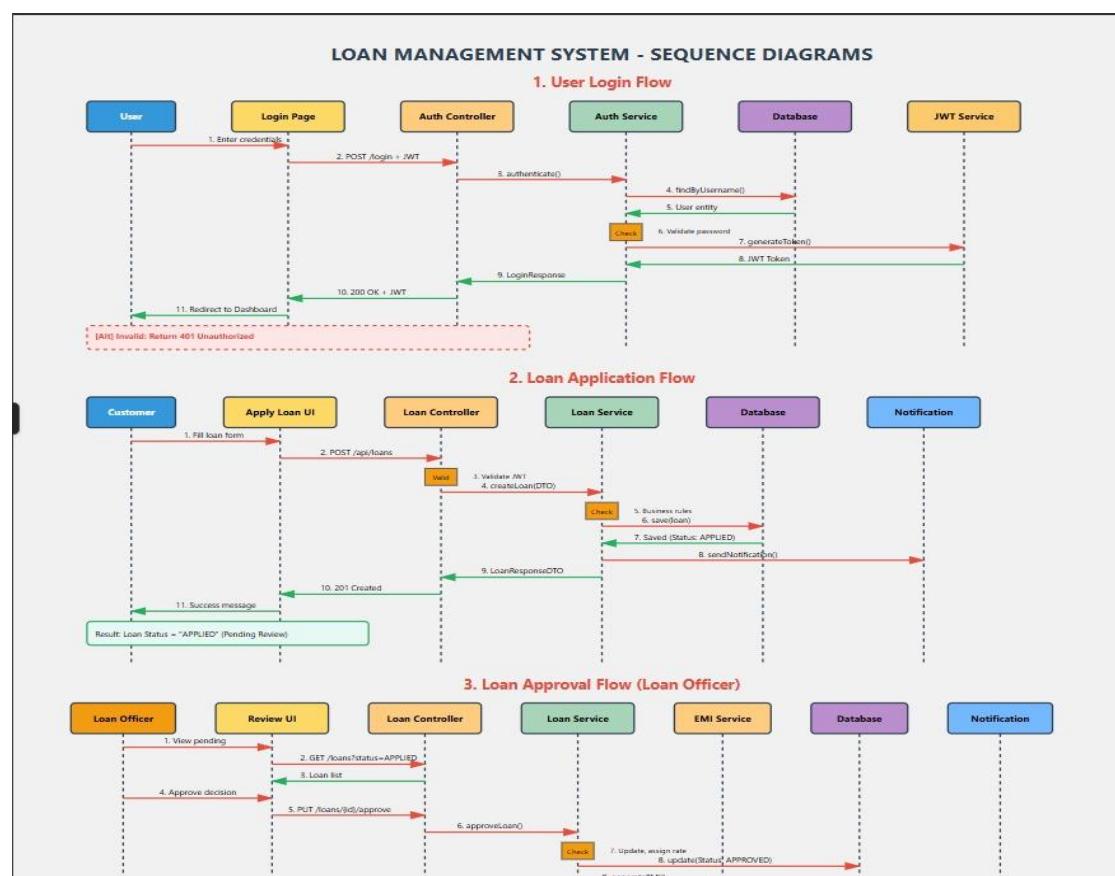
3. Architecture Style

3.1 Architectural Pattern

The system follows a **Microservices-Oriented Architecture** combined with **Layered Design Principles**.

Key characteristics:

- Loose coupling between services
- Independent data ownership
- REST-based inter-service communication
- Stateless backend services
- Frontend-backend separation (SPA + APIs)



3.Core System Components

4.1 Frontend Layer

- Single Page Application (SPA)
- Handles UI rendering, form validation, and role-based navigation
- Communicates with backend via secured REST APIs

4.2 Backend Layer

- Stateless REST services
- Implements business logic, validation, security, and workflows
- Exposes versioned APIs

4.3 Persistence Layer

- Independent databases per service
- Ensures data isolation and scalability

4.4 Security Layer

- · JWT authentication
- · Role-based authorization
- · Encrypted password storage
- · Secure API endpoints

5.Techology Stack

5.1 Backend

- Java 17+
- Spring Boot 3.x
- Spring Web (REST APIs)
- Spring Data JPA
- Hibernate ORM
- Spring Security with JWT
- Swagger / OpenAPI
- Maven

5.2 Frontend

- Angular (latest)
- TypeScript
- Angular Material / Bootstrap
- Reactive Forms
- HTTP Interceptors
- Route Guards

5.3 Database

- PostgreSQL / MySQL
- Separate schema per microservice

5.4 DevOps & Tools

- Git & GitHub
- Postman
- Environment-based configuration (dev/test/prod)

5.5 Testing

- JUnit 5
- Mockito
- Postman collections

6. Microservice Design

6.1 User Service

Responsibilities

- User registration and authentication
- Role and permission management
- JWT generation and validation
- Password hashing and security policies

APIs

ADMIN DASHBOARD

The screenshot shows the Postman interface with the following details:

- Workspace:** riddhima bhanja's Workspace
- Collection:** Loan Management System - Complete API Collection / 8. Reports & ...
- Request:** GET {{baseUrl}} /api/dashboard/admin
- Body:** 200 OK (JSON response)
- Response Data:**

```
1 {  
2     "totalLoans": 19,  
3     "totalCustomers": 12,  
4     "pendingApprovals": 1,  
5     "approvedLoans": 14,  
6     "disbursedLoans": 6,  
7     "totalDisbursedAmount": 0,  
8     "totalEmiCollected": 41763.75,  
9     "pendingEmiAmount": 12431222.96,  
10    "overdueAmount": 25507.61,  
11    "overdueCount": 14,
```

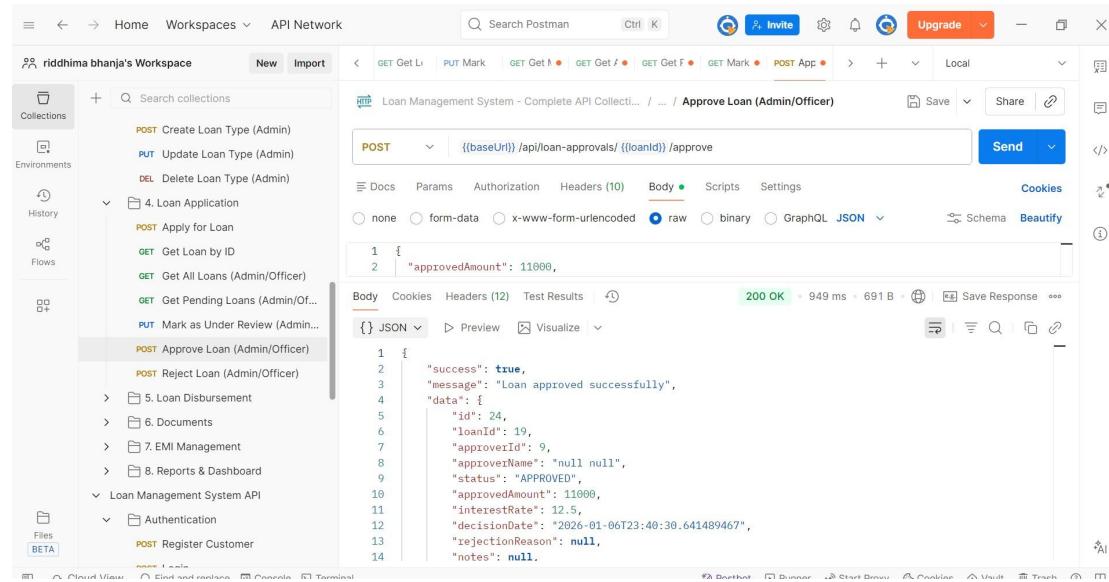
APPLY FOR LOAN

The screenshot shows the Postman interface with the following details:

- Workspace:** riddhima bhanja's Workspace
- Collection:** Loan Management System - Complete API Collection / 8. Reports & ...
- Request:** GET {{baseUrl}} /api/dashboard/admin
- Body:** 200 OK (JSON response)
- Response Data:**

```
1 {  
2     "totalLoans": 19,  
3     "totalCustomers": 12,  
4     "pendingApprovals": 1,  
5     "approvedLoans": 14,  
6     "disbursedLoans": 6,  
7     "totalDisbursedAmount": 0,  
8     "totalEmiCollected": 41763.75,  
9     "pendingEmiAmount": 12431222.96,  
10    "overdueAmount": 25507.61,  
11    "overdueCount": 14,
```

APPROVE LOAN



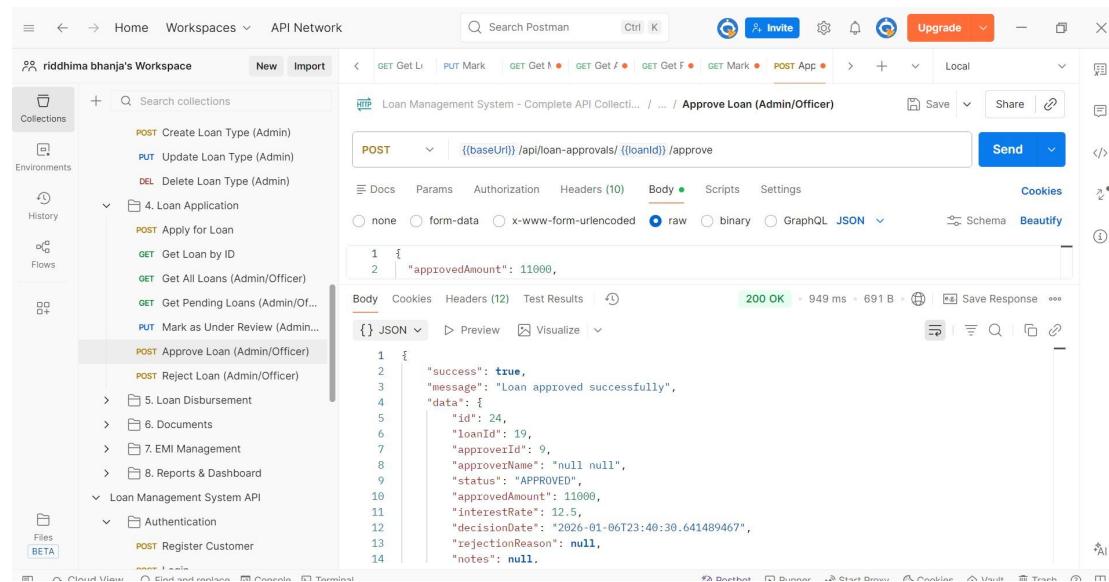
The screenshot shows the Postman application interface with the following details:

- Workspace:** riddhima bhanja's Workspace
- Collection:** Loan Management System - Complete API Collection
- Request:** POST /api/loan-approvals/{loanId}/approve
- Body:** Raw JSON input:

```
{ "approvedAmount": 11000, }
```
- Response:** 200 OK (949 ms, 691 B)

```
{"success": true, "message": "Loan approved successfully", "data": { "id": 24, "loanId": 19, "approverId": 9, "approverName": "null null", "status": "APPROVED", "approvedAmount": 11000, "interestRate": 12.5, "decisionDate": "2026-01-06T23:40:30.641489467", "rejectionReason": null, "notes": null. }}
```

CREATE LOAN



The screenshot shows the Postman application interface with the following details:

- Workspace:** riddhima bhanja's Workspace
- Collection:** Loan Management System - Complete API Collection
- Request:** POST /api/loan-approvals/{loanId}/approve
- Body:** Raw JSON input:

```
{ "approvedAmount": 11000, }
```
- Response:** 200 OK (949 ms, 691 B)

```
{"success": true, "message": "Loan approved successfully", "data": { "id": 24, "loanId": 19, "approverId": 9, "approverName": "null null", "status": "APPROVED", "approvedAmount": 11000, "interestRate": 12.5, "decisionDate": "2026-01-06T23:40:30.641489467", "rejectionReason": null, "notes": null. }}
```

DELETE LOAN

The screenshot shows the Postman application interface with the following details:

- Workspace:** riddhima bhanja's Workspace
- Collection:** Loan Management System - Complete API Collection
- Endpoint:** POST {{baseUrl}}/api/loan-approvals/ {{loanId}}/approve
- Body:** raw JSON (approvedAmount: 11000)
- Response:** 200 OK (Success: true, message: "Loan approved successfully", data: {id: 24, loanId: 19, approveId: 9, approveName: "null null", status: "APPROVED", approvedAmount: 11000, interestRate: 12.5, decisionDate: "2026-01-06T23:40:30.641489467", rejectionReason: null, notes: null})

DISBURSE LOAN

The screenshot shows the Postman application interface with the following details:

- Workspace:** riddhima bhanja's Workspace
- Collection:** Loan Management System - Complete API Collection
- Endpoint:** POST {{baseUrl}}/api/loan-approvals/ {{loanId}}/approve
- Body:** raw JSON (approvedAmount: 11000)
- Response:** 200 OK (Success: true, message: "Loan approved successfully", data: {id: 24, loanId: 19, approveId: 9, approveName: "null null", status: "APPROVED", approvedAmount: 11000, interestRate: 12.5, decisionDate: "2026-01-06T23:40:30.641489467", rejectionReason: null, notes: null})

GET ACTIVE LOAN TYPES

The screenshot shows the Postman interface with a collection named "riddhima bhanja's Workspace". A specific API endpoint, "Approve Loan (Admin/Officer)", is selected. The request method is POST, and the URL is `http://{{baseUrl}}/api/loan-approvals/ {{loanId}}/approve`. The request body contains the JSON object:

```
1 {  
2   "approvedAmount": 11000,  
3 }
```

. The response status is 200 OK, with a response time of 949 ms and a response size of 691 B. The response body is:

```
1 {  
2   "success": true,  
3   "message": "Loan approved successfully",  
4   "data": {  
5     "id": 24,  
6     "loanId": 19,  
7     "approveId": 9,  
8     "approversName": "null null",  
9     "status": "APPROVED",  
10    "approvedAmount": 11000,  
11    "interestRate": 12.5,  
12    "decisionDate": "2026-01-06T23:40:30.641489467",  
13    "rejectionReason": null,  
14    "notes": null.  
15 }
```

GET ALL LOAN TYPES

The screenshot shows the Postman interface with the same collection. A specific API endpoint, "Get All Loan Types (Public)", is selected. The request method is GET, and the URL is `http://{{baseUrl}}/api/loan-types`. The request includes the headers: Accept-Encoding (gzip, deflate, br) and Connection (keep-alive). The response status is 200 OK, with a response time of 177 ms and a response size of 2.68 KB. The response body is:

```
1 {  
2   "success": true,  
3   "message": "Loan types fetched successfully",  
4   "data": [  
5     {  
6       "id": 1,  
7       "name": "HOME LOAN",  
8       "description": "Loan for purchasing or constructing residential property. Lower interest  
9       rates with longer tenure options.",  
10      "minAmount": 5000.00,  
11      "maxAmount": 5000000.00,  
12      "minTenureMonths": 60,  
13      "maxTenureMonths": 360,  
14      "interestRate": 8.50,  
15     }  
16   ]  
17 }
```

GET ALL LOANS

The screenshot shows the Postman application interface with the following details:

- Workspace:** riddhima bhanja's Workspace
- Collection:** Loan Management System - Complete API Collection / 3. L... / Get All Loan Types (Public)
- Method:** GET
- URL:** {{baseUrl}} /api/loan-types
- Headers:** (7)
 - Accept-Encoding: gzip, deflate, br
 - Connection: keep-alive
- Body:** ({} JSON) (12)
 - 1: {
 - 2: "success": true,
 - 3: "message": "Loan types fetched successfully",
 - 4: "data": [
 - 5: {
 - 6: "id": 1,
 - 7: "name": "HOME LOAN",
 - 8: "description": "Loan for purchasing or constructing residential property. Lower interest rates with longer tenure options.",
 - 9: "minAmount": 5000.00,
 - 10: "maxAmount": 5000000.00,
 - 11: "minTenureMonths": 60,
 - 12: "maxTenureMonths": 360,
 - 13: "interestRate": 8.50,

- Test Results:** 200 OK, 177 ms, 2.68 KB
- Buttons:** Save Response, AI

GET ALL USERS

The screenshot shows the Postman application interface with the following details:

- Workspace:** riddhima bhanja's Workspace
- Collection:** Loan Management System - Complete API Collection / 3. L... / Get All Loan Types (Public)
- Method:** GET
- URL:** {{baseUrl}} /api/loan-types
- Headers:** (7)
 - Accept-Encoding: gzip, deflate, br
 - Connection: keep-alive
- Body:** ({} JSON) (12)
 - 1: {
 - 2: "success": true,
 - 3: "message": "Loan types fetched successfully",
 - 4: "data": [
 - 5: {
 - 6: "id": 1,
 - 7: "name": "HOME LOAN",
 - 8: "description": "Loan for purchasing or constructing residential property. Lower interest rates with longer tenure options.",
 - 9: "minAmount": 5000.00,
 - 10: "maxAmount": 5000000.00,
 - 11: "minTenureMonths": 60,
 - 12: "maxTenureMonths": 360,
 - 13: "interestRate": 8.50,

- Test Results:** 200 OK, 177 ms, 2.68 KB
- Buttons:** Save Response, AI

GET CURRENT USER PROFILE

The screenshot shows the Postman application interface with the following details:

- Collection:** riddhima bhanja's Workspace
- API:** Loan Management System - Complete API Collection / 3. L... / Get All Loan Types (Public)
- Method:** GET
- URL:** {{baseUrl}} /api/loan-types
- Headers:** (7)
 - Accept-Encoding: gzip, deflate, br
 - Connection: keep-alive
- Body:** (12) JSON
 - 1: {
2: "success": true,
3: "message": "Loan types fetched successfully",
4: "data": [
5: {
6: "id": 1,
7: "name": "HOME LOAN",
8: "description": "Loan for purchasing or constructing residential property. Lower interest rates with longer tenure options.",
9: "minAmount": 5000.00,
10: "maxAmount": 5000000.00,
11: "minTenureMonths": 60,
12: "maxTenureMonths": 360,
13: "interestRate": 8.50,
- Test Results:** 200 OK, 177 ms, 2.68 KB
- Buttons:** Save, Share, Send

GET DASHBOARD ROLE BASED

The screenshot shows the Postman application interface with the following details:

- Collection:** riddhima bhanja's Workspace
- API:** Loan Management System - Complete API Collection / 3. L... / Get All Loan Types (Public)
- Method:** GET
- URL:** {{baseUrl}} /api/loan-types
- Headers:** (7)
 - Accept-Encoding: gzip, deflate, br
 - Connection: keep-alive
- Body:** (12) JSON
 - 1: {
2: "success": true,
3: "message": "Loan types fetched successfully",
4: "data": [
5: {
6: "id": 1,
7: "name": "HOME LOAN",
8: "description": "Loan for purchasing or constructing residential property. Lower interest rates with longer tenure options.",
9: "minAmount": 5000.00,
10: "maxAmount": 5000000.00,
11: "minTenureMonths": 60,
12: "maxTenureMonths": 360,
13: "interestRate": 8.50,
- Test Results:** 200 OK, 177 ms, 2.68 KB
- Buttons:** Save, Share, Send

GET DISBURSEMENT DETAILS

The screenshot shows the Postman application interface with the following details:

- Left Sidebar:** Shows "riddhima bhanja's Workspace" with a tree view of API collections, environments, history, flows, and files.
- Header:** "GET Get /" and "GET Get Disbursement Details".
- Body:** A JSON response with status 200 OK, 63 ms, and 733 B. The response body is:

```
1 {  
2   "success": true,  
3   "message": "Disbursement details retrieved successfully",  
4   "data": {  
5     "id": 6,  
6     "loanId": 19,  
7     "disbursedBy": 9,  
8     "disbursedByName": "null null",  
9     "amount": 11000.00,  
10    "disbursementDate": "2025-01-01",  
11    "disbursementMethod": "BANK_TRANSFER",  
12    "referenceNumber": "RFF123456"
```

GET DOCUMENTS FOR LOAN

The screenshot shows the Postman application interface with the following details:

- Left Sidebar:** Shows "riddhima bhanja's Workspace" with a tree view of API collections, environments, history, flows, and files.
- Header:** "GET Get /" and "GET Get Disbursement Details".
- Body:** A JSON response with status 200 OK, 63 ms, and 733 B. The response body is:

```
1 {  
2   "success": true,  
3   "message": "Disbursement details retrieved successfully",  
4   "data": {  
5     "id": 6,  
6     "loanId": 19,  
7     "disbursedBy": 9,  
8     "disbursedByName": "null null",  
9     "amount": 11000.00,  
10    "disbursementDate": "2025-01-01",  
11    "disbursementMethod": "BANK_TRANSFER",  
12    "referenceNumber": "RFF123456"
```

GET EMI SCHEDULE

The screenshot shows the Postman interface with the "Loan Management System - Complete API Collection" selected. The left sidebar lists various collections and environments. The main panel displays the "GET Disbursement Details" endpoint under the "6. Documents" section. The request URL is `({{baseUrl}} /api/loan-approvals/disbursement/loan/{loanId})`. The response status is 200 OK, and the JSON body contains the following data:

```
1 {  
2   "success": true,  
3   "message": "Disbursement details retrieved successfully",  
4   "data": {  
5     "id": 6,  
6     "loanId": 19,  
7     "disbursedBy": 9,  
8     "disbursedByName": "null null",  
9     "amount": 11000.00,  
10    "disbursementDate": "2025-01-01",  
11    "disbursementMethod": "BANK TRANSFER",  
12    "referenceNumber": "RFF123456"  
13  }  
14}
```

GET LOAN SUMMARY REPORT

The screenshot shows the Postman interface with the "Loan Management System - Complete API Collection" selected. The left sidebar lists various collections and environments. The main panel displays the "GET Loan Summary Report (Admin/Officer)" endpoint under the "8. Reports & Dashboard" section. The request URL is `({{baseUrl}} /api/reports/loans/summary?startDate=2026-01-01&endDate=2026-01-31)`. The response status is 200 OK, and the JSON body contains the following data:

```
1 {  
2   "totalLoans": 19,  
3   "loansByStatus": {  
4     "PENDING": 1,  
5     "APPROVED": 6,  
6     "DISBURSED": 6,  
7     "REJECTED": 6  
8   },  
9   "loansByType": {  
10    "TYPE_5": 2,  
11    "TYPE_3": 5,  
12    "TYPE_4": 5,  
13    "TYPE_1": 1,  
14  }  
15}
```

GET LOAN TYPE BY ID

The screenshot shows the Postman interface with the following details:

- Workspace:** riddhima bhanja's Workspace
- Request URL:** {{baseUrl}} /api/reports/loans/summary?startDate=2026-01-01&endDate=2026-01-31
- Method:** GET
- Body:** JSON (empty)
- Response Status:** 200 OK
- Response Body (JSON):**

```
1  {
2    "totalLoans": 19,
3    "loansByStatus": {
4      "PENDING": 1,
5      "APPROVED": 6,
6      "DISBURSED": 6,
7      "REJECTED": 6
8    },
9    "loansByType": {
10      "TYPE_5": 2,
11      "TYPE_3": 5,
12      "TYPE_4": 5,
13      "TYPE_1": 1,
14    }
15  }
```

GET LOANS BY ID

The screenshot shows the Postman interface with the following details:

- Workspace:** riddhima bhanja's Workspace
- Request URL:** {{baseUrl}} /api/reports/loans/summary?startDate=2026-01-01&endDate=2026-01-31
- Method:** GET
- Body:** JSON (empty)
- Response Status:** 200 OK
- Response Body (JSON):**

```
1  {
2    "totalLoans": 19,
3    "loansByStatus": {
4      "PENDING": 1,
5      "APPROVED": 6,
6      "DISBURSED": 6,
7      "REJECTED": 6
8    },
9    "loansByType": {
10      "TYPE_5": 2,
11      "TYPE_3": 5,
12      "TYPE_4": 5,
13      "TYPE_1": 1,
14    }
15  }
```

GET OVERDUE EMI

The screenshot shows the Postman API client interface. The left sidebar displays a workspace structure with collections for Loan Disbursement, Documents, and EMI Management. The main area shows a GET request to `https://Loan Management System - Complete API Col... / ... / Get Overdue EMIs (Admin/Officer)`. The request parameters are `GET` and `(({baseUrl})) /api/emis/overdue`. The response status is `200 OK` with a response time of 53 ms and a size of 4.26 KB. The response body is a JSON object:

```
{ "success": true, "message": "Overdue EMIs retrieved successfully", "data": [ { "id": 208, "loanId": 19, "customerId": 9, "emiNumber": 1, "emiAmount": 428.83, "principalComponent": 314.25, "interestComponent": 114.58, "dueDate": "2025-01-01", "outstandingBalance": 10685.75, "status": "PENDING" } ] }
```

GET PAYMENT HISTORY

The screenshot shows the Postman API client interface. The left sidebar displays a workspace structure with collections for Loan Disbursement, Documents, and EMI Management. The main area shows a GET request to `https://Loan Management System - Complete API Col... / ... / Get Overdue EMIs (Admin/Officer)`. The request parameters are `GET` and `(({baseUrl})) /api/emis/overdue`. The response status is `200 OK` with a response time of 53 ms and a size of 4.26 KB. The response body is a JSON object:

```
{ "success": true, "message": "Overdue EMIs retrieved successfully", "data": [ { "id": 208, "loanId": 19, "customerId": 9, "emiNumber": 1, "emiAmount": 428.83, "principalComponent": 314.25, "interestComponent": 114.58, "dueDate": "2025-01-01", "outstandingBalance": 10685.75, "status": "PENDING" } ] }
```

GET PENDING LOANS

The screenshot shows the Postman interface with the following details:

- Collection:** riddhima bhanja's Workspace
- Request Type:** GET
- URL:** {{baseUrl}}/api/emis/overdue
- Body:** None
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 {
  "success": true,
  "message": "Overdue EMIs retrieved successfully",
  "data": [
    {
      "id": 208,
      "loanId": 19,
      "customerId": 9,
      "emiNumber": 1,
      "emiAmount": 428.83,
      "principalComponent": 314.25,
      "interestComponent": 114.58,
      "dueDate": "2025-01-01",
      "outstandingBalance": 10685.75,
      "status": "PENDING"
    }
  ]
}
```

GET USER BY ID

The screenshot shows the Postman interface with the following details:

- Collection:** riddhima bhanja's Workspace
- Request Type:** GET
- URL:** {{baseUrl}}/api/emis/overdue
- Body:** None
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 {
  "success": true,
  "message": "Overdue EMIs retrieved successfully",
  "data": [
    {
      "id": 208,
      "loanId": 19,
      "customerId": 9,
      "emiNumber": 1,
      "emiAmount": 428.83,
      "principalComponent": 314.25,
      "interestComponent": 114.58,
      "dueDate": "2025-01-01",
      "outstandingBalance": 10685.75,
      "status": "PENDING"
    }
  ]
}
```

LOAN OFFICER DASHBOARD

The screenshot shows the Postman interface with the following details:

- Collection:** riddhima bhanja's Workspace
- Request:** GET /api/emis/overdue
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 {
  "success": true,
  "message": "Overdue EMIs retrieved successfully",
  "data": [
    {
      "id": 208,
      "loanId": 19,
      "customerId": 9,
      "emiNumber": 1,
      "emiAmount": 428.83,
      "principalComponent": 314.25,
      "interestComponent": 114.58,
      "dueDate": "2025-01-01",
      "outstandingBalance": 10685.75,
      "status": "PENDING"
    }
  ]
}
```

LOGIN

The screenshot shows the Postman interface with the following details:

- Collection:** riddhima bhanja's Workspace
- Request:** GET /api/emis/overdue
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 {
  "success": true,
  "message": "Overdue EMIs retrieved successfully",
  "data": [
    {
      "id": 208,
      "loanId": 19,
      "customerId": 9,
      "emiNumber": 1,
      "emiAmount": 428.83,
      "principalComponent": 314.25,
      "interestComponent": 114.58,
      "dueDate": "2025-01-01",
      "outstandingBalance": 10685.75,
      "status": "PENDING"
    }
  ]
}
```

MARK AS UNDER REVIEW

The screenshot shows the Postman application interface. On the left, the sidebar lists collections: 5. Loan Disbursement, 6. Documents, 7. EMI Management, 8. Reports & Dashboard, and others under Loan Management System API and Authentication. The main area displays a GET request for 'Get Overdue EMIs (Admin/Officer)' with the URL `http://{{baseUrl}}/api/emis/overdue`. The request has 8 headers and no body. The response status is 200 OK, with a response time of 53 ms and a size of 4.26 KB. The response body is a JSON object:

```
1 {  
2   "success": true,  
3   "message": "Overdue EMIs retrieved successfully",  
4   "data": [  
5     {  
6       "id": 208,  
7       "loanId": 19,  
8       "customerId": 9,  
9       "emiNumber": 1,  
10      "emiAmount": 428.83,  
11      "principalComponent": 314.25,  
12      "interestComponent": 114.58,  
13      "dueDate": "2025-01-01",  
14      "outstandingBalance": 10685.75,  
15      "status": "PENDING",  
..]
```

REGISTER CUSTOMER

The screenshot shows the Postman application interface. On the left, the sidebar lists collections: 5. Loan Disbursement, 6. Documents, 7. EMI Management, 8. Reports & Dashboard, and others under Loan Management System API and Authentication. The main area displays a GET request for 'Get Overdue EMIs (Admin/Officer)' with the URL `http://{{baseUrl}}/api/emis/overdue`. The request has 8 headers and no body. The response status is 200 OK, with a response time of 53 ms and a size of 4.26 KB. The response body is a JSON object:

```
1 {  
2   "success": true,  
3   "message": "Overdue EMIs retrieved successfully",  
4   "data": [  
5     {  
6       "id": 208,  
7       "loanId": 19,  
8       "customerId": 9,  
9       "emiNumber": 1,  
10      "emiAmount": 428.83,  
11      "principalComponent": 314.25,  
12      "interestComponent": 114.58,  
13      "dueDate": "2025-01-01",  
14      "outstandingBalance": 10685.75,  
15      "status": "PENDING",  
..]
```

UPDATE LOAN TYPE

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'riddhima bhanja's Workspace' containing collections, environments, history, flows, files, and a cloud view. The main area displays an API collection for 'Loan Management System - Complete API Collection / 3. L... / Update Loan Type (Admin)'. The collection includes several endpoints: GET Get User by ID (Admin), GET Get All Loan Types (Public), GET Get Active Loan Types (Public), GET Get Loan Type by ID, POST Create Loan Type (Admin), PUT Update Loan Type (Admin), and DEL Delete Loan Type (Admin). The 'PUT Update Loan Type (Admin)' endpoint is selected, showing a 'PUT {{baseUrl}}/api/loan-types/{id}' request. The 'Params' tab is active, showing a key-value pair 'Key' with 'Value' 'Description'. The 'Body' tab shows a JSON payload:

```
1  {
2      "success": true,
3      "message": "Loan type updated successfully",
4      "data": {
5          "id": 1,
6          "name": "HOME LOAN",
7          "description": "Updated description",
8          "minAmount": 15000,
9          "maxAmount": 600000,
10         "minTenureMonths": 60,
11         "maxTenureMonths": 360,
12         "interestRate": 8.50,
13         "lateFeePercentage": 2.00,
14         "gracePeriodDays": 3.
```

6.2 Product Service (Loan Service)

Responsibilities

- Loan type management
- Interest rate and tenure rules
- Loan application submission
- Loan status lifecycle management

APIs

- POST /loans/apply
- GET /loans/{id}
- GET /loans/customer/{customerId}
- PUT /loans/{id}/approve
- PUT /loans/{id}/reject

Database

- Loan table
- Loan type table
- Loan status history table

6.3 Order Service (EMI & Repayment Service)

Responsibilities

- EMI calculation
- EMI schedule generation
- Repayment tracking
- Outstanding balance computation
- Automatic loan closure

APIs

- GET /emis/loan/{loanId}
- POST /emis/pay
- GET /repayments/customer/{customerId}

Database

- EMI schedule table
- Repayment table
- Loan balance table

7. Data Design (Low-Level Design)

7.1 User Document

- userId
- name
- email
- password (hashed)
- role
- status
- createdAt

7.2 Product (Loan) Document

- loanId

- · customerId
- · loanType
- · principalAmount
- · interestRate
- · tenure
- · status
- · approvalRemarks

7.3 Order (EMI) Document

- · emild
- · loanId
- · dueDate
- · emiAmount
- · paidAmount
- · outstandingBalance
- · paymentStatus

8. API Design & Validation

- RESTful conventions followed
- DTOs used for request/response
- Input validation using @Valid
- Custom validators for:
- Loan amount limits
- Tenure constraints
- · EMI payment rules

9. Error Handling Strategy

9.1 Global Exception Handling

- Centralized exception management using @ControllerAdvice
- Custom exceptions:
- ResourceNotFoundException
- ValidationException
- UnauthorizedAccessException
- Standardized error response format:
- timestamp
- status
- errorCode

- message
- Pat

10. Security Design

- JWT-based authentication
- Stateless session management
- Role-based access control
- Password hashing using BCrypt
- Secured endpoints with method-level authorization
- HTTP interceptors for token propagation

11. Non-Functional Requirements

- Scalability: Horizontally scalable services
- Security: Encrypted credentials and secure APIs
- Performance: Optimized queries and pagination
- Maintainability: Clean layered architecture
- Availability: Fault-tolerant stateless services
- Auditability: Complete loan lifecycle traceability

12. Testing Strategy

12.1 Unit Testing

- Service-layer tests using JUnit and Mockito
- Mocked repositories and external dependencies

12.2 API Testing

- Postman collections for all endpoints
- Authentication and authorization test cases

12.3 Validation & Security Testing

- Input validation tests
- Unauthorized access tests
- Token expiry handling

Conclusion

The Loan Management System demonstrates **enterprise-level system design, secure full-stack development, and real-world financial workflows.**

It is **production-ready, cloud-deployable**, and built using **industry best practices**, making it a strong representation of modern backend and frontend engineering capabilities.