



Loan Management System

A fullstack **Loan Management System** for a microfinance or lending company. It manages client records, loan applications, payment schedules, reports, and balances using modern technologies with a focus on **stored procedures** for database operations.

Tech Stack

Frontend

- **Framework:** Next.js (App Router)
- **Styling:** Tailwind CSS
- **UI Components:** Shadcn UI
- **Data Fetching:** TanStack Query
- **Validation:** Zod + React Hook Form
- **Charts:** Recharts
- **Report Tools:** react-pdf, papaparse, Excel (via blob or download API)

Backend

- **Framework:** NestJS (TypeScript)
 - **Database:** MySQL/MariaDB (with stored procedures ONLY)
 - **Validation:** class-validator
 - **Auth:** JWT + Role-based Guards
 - **File Export:** pdfkit, exceljs
 - **Logging:** Winston or Pino
-

Project Structure

Frontend (Next.js)

```
frontend/  
├── app/  
│   └── dashboard/  
├── components/  
├── hooks/  
├── lib/  
├── schemas/  
└── styles/
```

```
├─ utils/  
└─ public/
```

Backend (NestJS)

```
backend/  
├─ src/  
│   ├── auth/  
│   ├── clients/  
│   ├── loans/  
│   ├── payments/  
│   ├── schedules/  
│   ├── reports/  
│   ├── common/  
│   └─ main.ts  
└─ test/
```

Core Modules

Clients

- CRUD operations
- Client portal login

Loans

- Apply, approve, reject loans
- Set interest rate and terms

Payments

- Record payments
- Apply penalties

Schedules

- Generate repayment schedules
- View overdue items

Reports & Charts

- Charts with Recharts
- Reports in PDF, Excel, CSV
- Printable views using Tailwind's `print:` utilities

Auth

- Role-based (Admin, Client)
 - JWT auth with route guards
-

📊 Reports

Types

Report	Format	Viewer
Loan Summary	PDF, Excel	Admin, Client
Client List	Excel	Admin
Payments (monthly)	CSV, Excel, Chart	Admin
Payment Schedule	PDF, Print	Client
Defaulters	Excel, Chart	Admin

Tools

- `pdfkit`, `puppeteer` for backend PDF
- `exceljs` for backend Excel
- `react-pdf`, `papaparse` for frontend

API Endpoints (Sample)

```
GET /reports/loan-summary/:loanId
GET /reports/clients/export
GET /reports/payments/monthly
```

Backend Utilities

- `@nestjs/config` for env
 - `mysql2` for stored procedure calls
 - `@nestjs/swagger` for API docs
 - `@nestjs/schedule` for cron jobs
 - `winston` or `pino` for logging
-

Frontend Utilities

- TanStack Query with fetch or Ky
 - `react-hook-form` for form logic
 - `zod` for schema validation
 - `lucide-react` for icons
 - `tailwind-variants` for Shadcn theming
-

Validation & Error Handling

- Backend: DTOs with `class-validator`, exception filters
 - Frontend: Zod + TanStack `isError`, `isLoading` handling
-

Testing

Layer	Tool
Backend Unit	Jest
Backend E2E	Supertest
Frontend E2E	Playwright or Cypress
Frontend Unit	Vitest + Testing Library

Development Roadmap

Phase 1 – Setup & Auth

- Project scaffolding
- NestJS + Next.js base
- JWT Auth (Client + Admin)

Phase 2 – Core Modules

- Clients, Loans, Payments, Schedule
- Stored procedure integration

Phase 3 – UI & Dashboards

- Build pages with Shadcn UI
- Integrate Recharts + Table views

Phase 4 – Reporting & Exports

- Generate PDFs, Excel, CSV
- Print pages

Phase 5 – Testing & Deployment

- Write tests
- Deploy to Vercel + Railway

Summary

A scalable, secure and feature-rich loan management platform using modern TypeScript stack with emphasis on stored procedures, role-based access, and powerful reporting tools.

Built with ❤️ using NestJS, Next.js, MySQL, and your vision.