Project Requirements Document (PRD)

Title: Library Management System

Overview

This web application helps libraries manage books, members, loans, and fines efficiently. Admins can manage the entire library, while users (members) can view their dashboard, check their loans, and see any fines. The system supports book CRUD, member management, loan tracking, and fine calculation, all with a modern, responsive UI.

Goals & Objectives

- Allow admins to manage books, members, loans, and fines.
- Enable users to view their loans and fines.
- Provide secure authentication and role-based access (admin vs. user).
- Offer a clean, responsive UI for both desktop and mobile.

Target User

- Library staff (admin)
- Library members (users)
- Age: 18+
- Users needing to manage or track library resources

Features

Feature	Description	Priority
Book Management	Add, update, delete, and list books	High
Member Management	Add, update, delete, and list members	High
Loan Management	Issue, return, and track book loans	High
Fine Management	Calculate and track fines for overdue books	High
User Dashboard	Members can view their loans and fines	High
Admin Dashboard	Admins see summary and manage all resources	High
Authentication	Secure login/logout for users and admins	High
Responsive Design	Works on both desktop and mobile devices	High

Functional Requirements

Book Management

- CRUD operations for books (title, author, category, year, copies)
- Admin only

Member Management

- CRUD operations for members (name, email, phone)
- Admin only

Loan Management

- Issue and return books
- Track due dates and status
- Admin only

Fine Management

- Calculate fines for overdue books
- Track paid/unpaid status
- Admin only

User Dashboard

- Members see their current loans and fines

Authentication & Access Control

- Only admins can access management features
- Members can only view their dashboard

UI/UX Requirements

- Clean, modern, mobile-first design (Tailwind CSS)
- Separate dashboards for admin and users
- Responsive tables and forms

Tech Stack

- Frontend: HTML, Tailwind CSS
- Backend: Django, PostgreSQL/SQLite
- Authentication: Django built-in
- Deployment: Render.com or similar

Timeline

Task	Deadline
Requirement Finalization	Day 1
UI Wireframes	Day 2
Build Book/Member CRUD	Day 3
Build Loan/Fine Management	Day 4
User/Admin Dashboard	Day 5
Testing & Responsive Design	Day 6
Final Touches & Report	Day 7

Success Criteria

- Admin can manage books, members, loans, and fines
- User can view their loans and fines
- Secure authentication and role-based access
- Fully responsive layout
- No major bugs during testing

Deliverables

- Working app (deployed)
- Code (GitHub)
- Short demo video (2-3 minutes)
- Final Report (PDF with screenshots and features)