Library Management System 28/8/25

roblem Statement

raditional library system rely on manual processes for managing books, issuing, returns and tracking which often leads to errors, delays and inefficiency. The LMS is required to automate these tasks, ensuring accurate record-keeping, faster transactions and better accessibility for students, librarians and administrators.

SRS Document

1. Introduction

1.1 Purpose of the Document

The purpose of this document is to define the requirements for the library Management System which automates library operations such as book cataloging, issue/return management, user registration, fine calculation etc. This document will guide developers, testers and stakeholders in designing ~~and~~ implementing the system.

1.2 Scope

LMS will
- Maintain a digital ~~copy~~ catalog of books and resources.
- Allow students to search, borrow and return books.
- Manage inventory; accounts and fines

- Generate reports
- Provide role-based access for students and administrators.

The system will be web-based, accessible browsers, and scalable for schools and universities.

1.3 Overview

The LMS will replace manual library record keeping with a centralized data and user-friendly interface. It will incl

- Student Portal: Book search, borrowing, ren return requests.
- Librarian dashboard: Book management, Use management, fines and reports
- Administrator Panel: System monitoring, da backup and configuration.

2. General Description

- The system will be web-based and accessible on desktops and mobile devices.
- Users include students, librarians and administrators.
- Maintain a centralized book database
- Interfaces include a web dashboard and REST APIs for integration.
- Data must be stored securely, role based access enforced and transactions logged.

3. Functional Requirements
- Add, update, delete and catalog books
- Register new members, manage accounts
- Search by title / author / category or ISBN
- Track issued books, due dates and returns
- Automatically calculate fine for overdue books
- Generate reports on book availability, usage and overdue lists.
- Role-based login

4. Interface Requirements

User Interfaces:
- Student Portal, librarian dashboard & admin panel

Software Interfaces,
- SQL based database for storage
- REST API for external integrations.

Hardware
- Barcode scanner support for issuing / returning books
- HTTPS for secure communication

5. Performance Requirements.
- Search results displayed in < 2 seconds
- Support upto 500 concurrent users
- Database must handle 100000+ records
- Uptime of 99.9% required during operational hours.

6. Design constraints

- Must be developed using Java, Python
- Should comply with data protection
- Mobile - friendly responsive design
- Daily backups

7. Non Functional Attributes

- Security - Password encryption, role-based
- Intuitive interface ~~for~~ ,
- Automated backup and recovery system
- Modular design for easy updates

8. Preliminary Schedule: and Budget

Week 1-2: Requirement gathering and design
3-5: Database and core module design
6-7: Issue/return module and fine calculation
8-9: Reports and search functional
10-11: Testing and debugging
12: Deployment and user training.

Budget

Requirement Analysis	-	₹3,00,000
System Design	-	₹5,00,000
Implementation	-	₹6,00,000
Testing	-	₹4,00,000
Deployment	-	₹3,50,000
Total	.	₹24,50,00