
**MBARARA UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**FACULTY OF COMPUTING AND INFORMATICS**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**COURSE UNIT: WEB DESIGN AND DEVELOPMENT**

**LECTURER: MR. BWAANA ANTHONY**

**ASSIGNMENT NUMBER: 1**

**ASSIGNMENT TYPE: INDIVIDUAL**

**REGISTRATION NUMBER:**

**2024/BIT/065/PS**

# Library Management System Project

### 1. Introduction

In the field of web development, a well-structured project concept ensures clarity, feasibili ty, and successful implementation. This document outlines a **Library Management Syst em (LMS)** designed to streamline library operations such as book tracking, borrower ma nagement, and transaction recording.

With the increasing number of books and users in libraries, a **web-based system** is esse ntial to improve efficiency, reduce errors, and provide easy access to library resources. T his project will develop an LMS to digitize library operations, making it accessible to adm inistrators, librarians, and students.

### 2. Problem Identification

Traditional library systems rely on **manual record-keeping**, which results in various chall enges, including:

- Difficulty in tracking book availability.
- Inefficient borrower and transaction management.
- Time-consuming book issuance and return processes.
- High risk of lost or misplaced records.
- Limited accessibility for users to check book availability remotely.

These challenges necessitate a **web-based solution** that ensures an organized, user-frie ndly, and efficient library system.

### 3. Proposed Web-Based Solution

In the modern world, libraries play a crucial role in knowledge management, offering boo ks, research materials, and digital content to students and professionals. However, traditi

onal library management relies heavily on manual record-keeping, which leads to ineffici encies.

This project proposes a **web-based Library Management System (LMS)** to **digitize** and **automate** library functions such as **book tracking, borrower management, and transact ion recording.** The system will make library services more accessible, reliable, and efficie nt for administrators, librarians, and students.

A well-developed LMS provides:

- **Automated book tracking** to prevent misplaced books.
- **User-friendly interfaces** for book searching and borrowing.
- **Secure authentication** for students, librarians, and admins.
- **Analytics and reports** for better library management.

By utilizing modern **web development technologies** such as **PHP, React.js, and MySQL**, t his project ensures a scalable and efficient system.

## 2. Problem Identification

### Challenges in Traditional Library Systems

Traditional libraries face numerous challenges due to outdated methods of record-keepin g and book management. Some key issues include:

**1. Manual Record-Keeping:**

Libraries often use paper-based or Excel-based systems for tracking books, which leads t o data inconsistencies, misplaced records, and errors.

**2. Difficulty in Tracking Books:**

Without a digital catalog, students and librarians struggle to check book availability, lead

ing to wasted time and frustration.

**3. Inefficient Borrower Management:**

Libraries often lose track of who has borrowed which books, resulting in delays and mis placed books.

**4. Time-Consuming Book Issuance and Returns:**

Librarians manually enter borrower details and due dates, which takes time and increase s the risk of errors.

**5. High Risk of Lost or Misplaced Records:**

Physical record books can be lost or damaged, leading to a complete loss of transaction history.

**6. Limited User Accessibility:**

Students and faculty members cannot check book availability remotely, requiring them t o visit the library in person.

A **Library Management System** solves these challenges by **automating book issuance, borrower tracking, and record-keeping in a centralized digital platform.**

---

**3. Proposed Web-Based Solution**

To address the issues in traditional libraries, we propose a **Library Management System** with the following features:

**Key Features of the System**

**User Authentication:** Secure login system for three user roles (Admin, Librarian, Studen t).

**Book Management:** Add, update, and delete book records efficiently.

**Borrower Management:** Track book borrowers and their due dates.

**Transaction Handling:** Record issued and returned books automatically.

**Search and Filtering:** Find books by title, author, category, or availability.

**Mobile Responsiveness:** Accessible on desktop and mobile devices.

**Reports & Analytics:** Generate reports on book borrowing trends.

**API Integration:** Use **Google Books API** to fetch book details.

By integrating these features, the **Library Management System** will enhance user experie nce, reduce administrative burden, and improve efficiency.

---

**4. System Architecture & Wireframe**

**System Architecture Overview**

The **LMS** will follow a **three-tier architecture:**

1. **Frontend (Client Side):** React.js, HTML, CSS, JavaScript
2. **Backend (Server Side):** PHP (Laravel Framework)
3. **Database:** MySQL

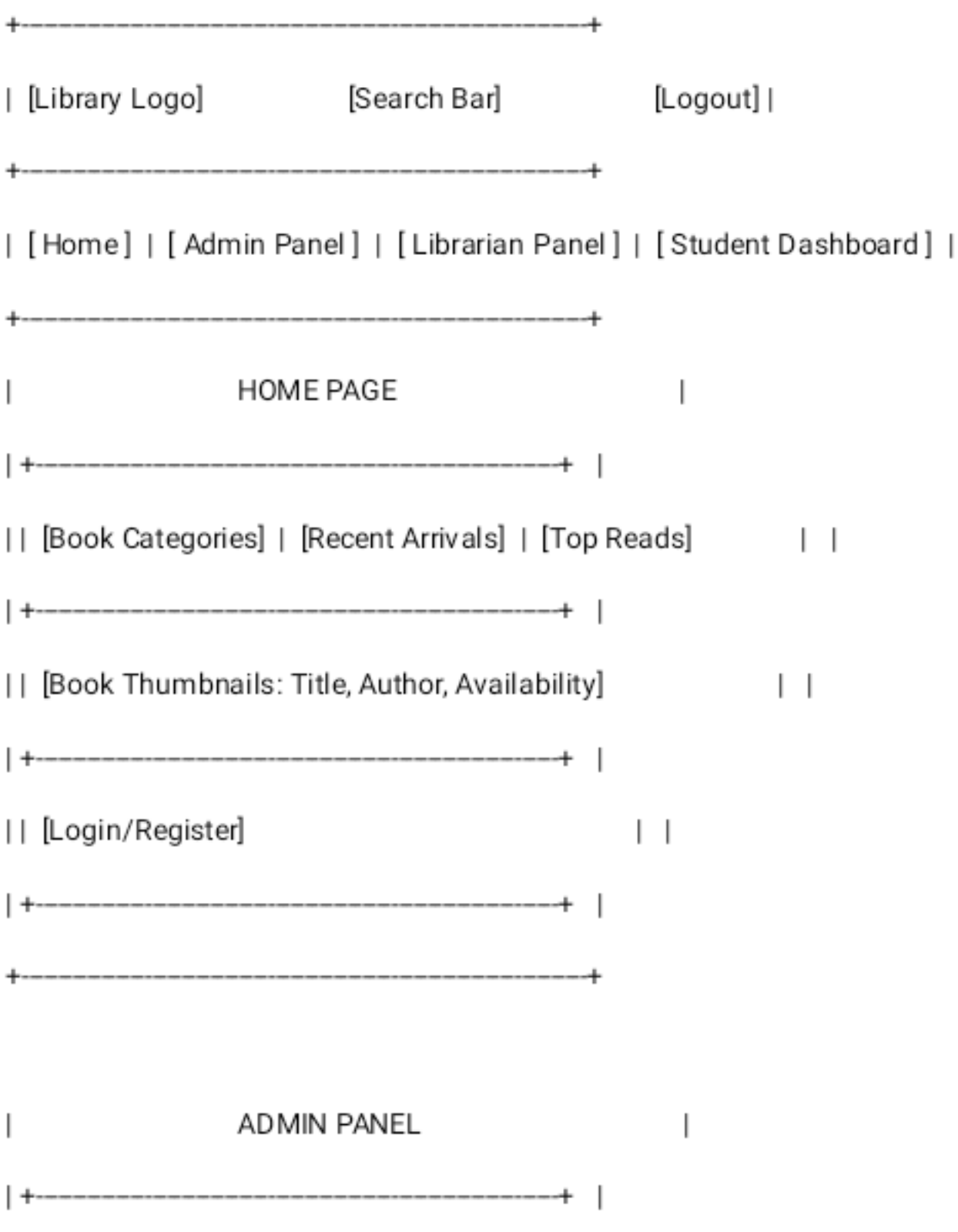
**Database Schema**

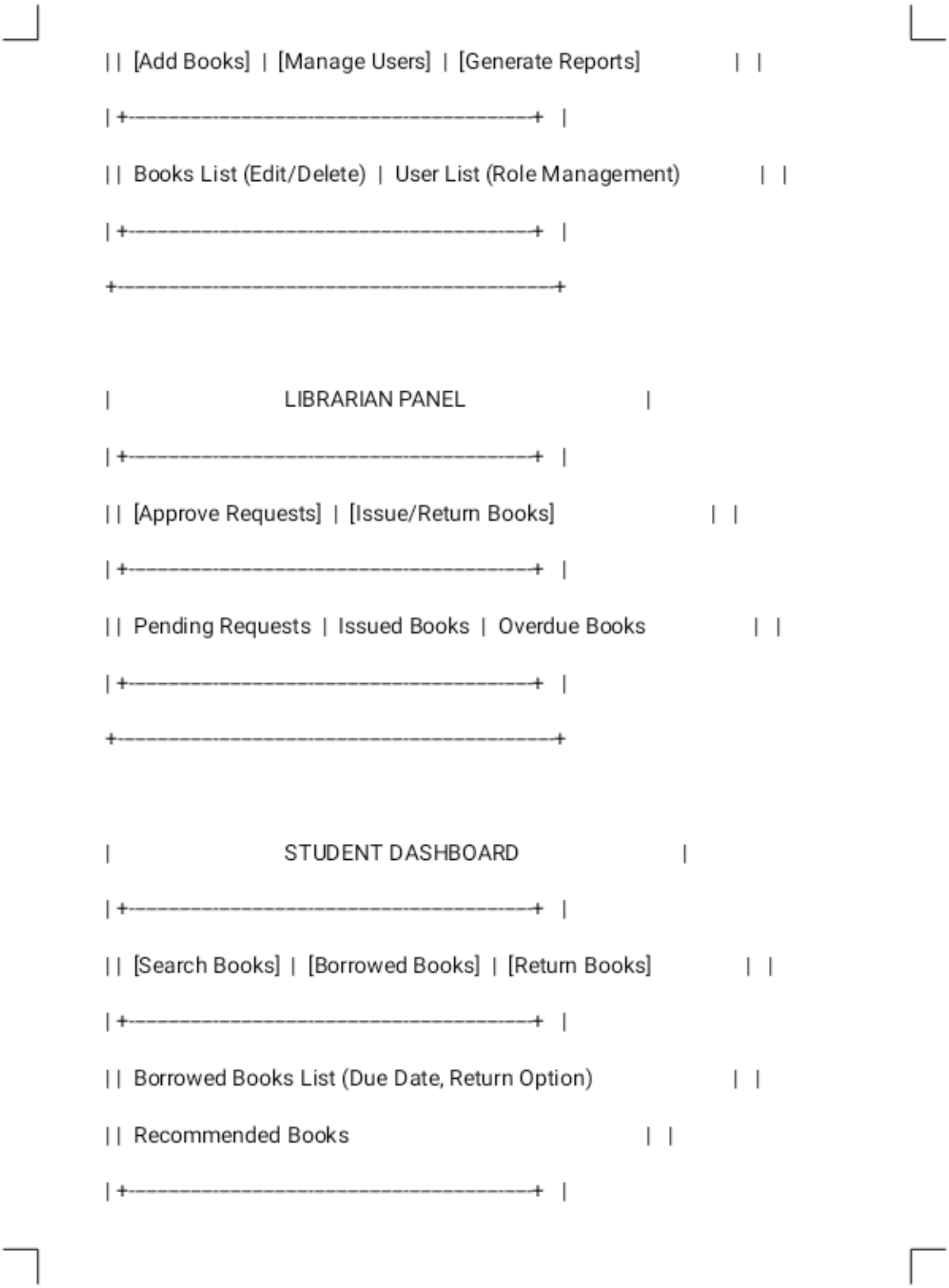
| Table Name | Columns | Description |
|---|---|---|
| Users | user_id, name, email, role | Stores user informatio n |
| Books | book_id, title, author, genre, availability | Stores book details |
| Transaction s | transaction_id, user_id, book_id, borrow_date, return _date | Tracks borrowing histo ry |

**Wireframe Overview**

- **Home Page:** Displays book catalog and search bar.

- **Admin Dashboard:** Manage users, books, and reports.
- **Librarian Panel:** Approve requests and manage books.
- **Student Dashboard:** View borrowed books and request new ones.
- Illustration




| | [Add Books] | [Manage Users] | [Generate Reports] | |
| +----------+ |
| | Books List (Edit/Delete) | User List (Role Management) | |
| +----------+ |
+----------+
| LIBRARIAN PANEL |
| +----------+ |
| | [Approve Requests] | [Issue/Return Books] | |
| +----------+ |
| | Pending Requests | Issued Books | Overdue Books | |
| +----------+ |
+----------+
| STUDENT DASHBOARD |
| +----------+ |
| | [Search Books] | [Borrowed Books] | [Return Books] | |
| +----------+ |
| | Borrowed Books List (Due Date, Return Option) | |
| | Recommended Books | |
| +----------+ |

+—

----------+

Explanation

1. Header Section – Displays the library logo, search bar, and logout button for easy navi gation.

2. Navigation Menu – Provides quick access to Home, Admin Panel, Librarian Panel, and Student Dashboard.

3. Home Page – Showcases book categories, recent arrivals, top reads, and book thumb nails with availability.

4. Admin Panel – Allows book management, user management, and report generation fo r administrators.

5. Librarian Panel – Facilitates approving book requests, issuing/returning books, and tr acking overdue books.

6. Student Dashboard – Enables students to search for books, track borrowed books, an d return books.

---

**5. Technology Stack**

**Frontend**

**HTML, CSS, JavaScript** – For structuring and styling the website.

**React.js** – To build an interactive and dynamic user interface.

**Backend**

**PHP (Laravel Framework)** – Handles API requests and database interactions.

**Database**

**MySQL** – Stores book and user records efficiently.

**Hosting**

**GitHub Pages, Netlify, Heroku, or AWS** – For hosting and deployment.

**API Integration**

**Google Books API** – To fetch additional book details.

---

**6. User Roles & Features**

| User Role | Responsibilities |
| --- | --- |
| **Administrator** | Manages users, books, and reports. |
| **Librarian** | Approves book requests, issues/returns books. |

| User Role | Responsibilities |
| --- | --- |
| **Student/User** | Searches, borrows books, views history. |

---

### 7. Research & References

**Sources Used**

- **MDN Web Docs & W3C** – Web development best practices.
- **IEEE Research Papers** – Digital library management case studies.
- **GitHub & Stack Overflow** – Coding solutions.

---

### 8. Implementation Plan

| Phase | Task | Timeline |
| --- | --- | --- |
| 1 | Research & Planning | 1 Week |
| 2 | Database & Backend Development | 2 Weeks |
| 3 | Frontend Development | 2 Weeks |
| 4 | Testing & Debugging | 1 Week |
| 5 | Deployment & Documentation | 1 Week |
| 6 | Submission & Final Presentation | Last Week |

---

## 9. Conclusion & Reflection

The **Library Management System** is an essential tool for modern libraries. It provides **automation, efficiency, and remote access** to students and librarians.