

# **BEE Project Based Evaluation**

**Project Report**  
**Semester- IV (Batch – 2023)**

**Library Management System**



**Supervised By:**  
Mr.Prabhjot Singh Manocha

**Submitted By:**  
Pranjal 2310990853 G11

Rohit 2310990870 G11

Harshit 2310990876 G11

Vinay 2310990917 G11

**Department of Computer Science and Engineering Chitkara**  
**University Institute of Engineering & Technology,**  
**Chitkara University ,Punjab**

## ABSTRACT

The **Library Management System (LMS)** is a comprehensive and user-centric web-based platform designed to efficiently manage and streamline the core operations of a modern library. Developed using the **MERN stack (MongoDB, Express.js, React, Node.js)**, this system serves the dual purpose of enhancing the borrowing experience for students and simplifying administrative tasks for librarians and educators.

On the user side, students can register accounts, log in securely, and access personalized dashboards. They can browse the library catalog, which showcases an organized collection of books with details like title, author, genre, availability status, and more. Through intuitive filters and search functionality, users can easily find books that match their interests. Each book entry is displayed in a card format with a 'Lend' button that allows users to borrow available books. Borrowed books are tracked and shown in the user's "Lent Books" section, enabling them to monitor their current loans. Additionally, users can receive notifications about due dates and new arrivals, making the borrowing experience smooth and engaging.

On the admin/teacher side, the LMS provides robust tools for managing the library's inventory and user activities. Admins and librarians can add, update, or remove books from the system, monitor who has borrowed what, and manage user accounts. Teachers have access to role-based dashboards, enabling them to view students' borrowing history and assign or recommend reading material if needed. The system is secured with authentication and role-based access control, ensuring that users can only perform actions permitted for their role.

The platform emphasizes a clean UI/UX design with seamless navigation and responsive layouts, ensuring accessibility across devices. By automating traditional manual processes like ledger entries, book lending, and due date tracking, the LMS significantly reduces operational overhead and human error. Furthermore, it fosters a reading culture among students by providing a transparent and interactive experience that encourages regular engagement with the library.

## INDEX

SR. NO	Section	Pg no.
1.	Introduction	4 - 8
2.	Problem Statement	9 - 12
3.	Proposed Design / Methodology	13 – 15
4.	Results	16 - 19
5 .	Key Features	20
6.	Conclusion	21
7.	References	22

# 1. INTRODUCTION

## 1.1 Background

The traditional method of managing library resources has long relied on manual systems involving handwritten records, physical ledgers, and face-to-face interactions. While effective in their time, these methods are time-consuming, error-prone, and inefficient—especially in academic institutions with large student populations. Tasks such as lending books, tracking due dates, managing inventory, and recording student borrowing history were cumbersome and lacked scalability. As student expectations and institutional needs evolved, the limitations of paper-based library systems became increasingly apparent, creating a strong need for digital transformation in library operations.

With the growth of digital technologies and the widespread availability of the internet, **Library Management Systems (LMS)** have emerged as a transformative solution. These systems digitize and automate core library functions, enabling students to search for books, borrow them, and manage their lending activity online. At the same time, administrators can easily manage inventory, track overdue books, and monitor borrowing patterns from a centralized dashboard. This transition from manual to automated systems significantly reduces human error, enhances operational efficiency, and offers greater convenience to both students and librarians.

In educational institutions where the number of enrolled students is rapidly growing, the demand for a system that is efficient, scalable, and user-friendly has never been higher. A web-based LMS, built using technologies like the MERN stack, enables real-time updates, remote access, and seamless integration of various features like authentication, notifications, and user-specific dashboards. These platforms not only save time and reduce operational costs but also provide better visibility into the reading habits and resource utilization across the institution.

In essence, the development of online Library Management Systems addresses the pressing need for modernization in resource management. It empowers students with self-service tools, assists teachers in overseeing student engagement with academic resources, and simplifies library operations. Such systems contribute significantly to the educational environment by promoting a culture of reading, improving information accessibility, and ensuring that the management of knowledge resources keeps pace with technological advancements.

## **1.2 Objectives**

The primary objective of developing this Library Management System (LMS) is to create an efficient, accessible, and user-friendly platform that modernizes the way libraries operate in academic institutions. The project aims to overcome the limitations of traditional manual library systems by leveraging digital technologies to streamline book lending, inventory management, and user interaction. The following are the key objectives of the LMS project:

### **1. Facilitate Seamless Book Search and Lending**

One of the core goals of the LMS is to empower students and library users by providing them with a centralized platform where they can search for available books, filter them based on categories (such as subject, author, or availability), and request or borrow books directly online. This objective is to simplify and speed up the book lending process, making library access more convenient and efficient for users.

### **2. Streamline Library Operations for administrators and staff**

The system is designed to assist librarians, administrators, and teachers in efficiently managing the library's resources. With features such as book catalog management, user tracking, and borrow history monitoring, the platform aims to reduce manual effort and paperwork. It allows administrators to add, update, and remove books, view borrowing activity, and manage users from a single dashboard.

### **3. Ensure Secure and Role Based Access**

To maintain the integrity and security of library data, the LMS incorporates secure authentication and role-based access control. Students, teachers, and administrators each have customized access levels tailored to their roles. This ensures sensitive operations, such as editing inventory or accessing student borrowing records, are only available to authorized personnel.

#### **4. Enhance Accessibility and User Experience**

Another important objective is to make the library system accessible, intuitive, and responsive. The LMS is designed to be easy to use across devices and for users with various levels of technical proficiency. It supports a user-friendly interface that promotes engagement and encourages regular use of library services. By eliminating physical dependencies, the system allows remote access and enhances academic inclusivity.

#### **5. Enable Scalability and Future Enhancement**

The LMS is developed with a modular and scalable architecture to support future growth. It is designed to be adaptable, with room to incorporate advanced features such as due-date reminders, book recommendations, analytics, or integration with external databases or APIs. This ensures the platform can evolve with technological advancements and institutional requirements.

#### **6. Reduce Administrative Burden and Errors**

By automating essential functions like book lending, return tracking, and record maintenance, the LMS significantly reduces the administrative workload for library staff. This minimizes human error, enhances operational efficiency, and frees up resources that can be directed towards improving user experience and expanding the library's offerings.

### 1.3 Significance

The significance of the Library Management System (LMS) project lies in its ability to modernize traditional library operations and improve the overall efficiency, accessibility, and user experience for both library staff and users. The following points highlight the importance of this project:

- **Centralized Resource Management:** The LMS provides a unified platform for managing books, users, and transactions, significantly reducing reliance on manual processes and enabling better control over library operations.
- **Enhancing Accessibility:** By digitizing the library and enabling web-based access, the system allows students and teachers to browse and borrow books from any location, at any time, removing the constraints of physical presence and opening access to a wider user base.
- **Improving Efficiency:** Automating routine tasks such as book lending, return tracking, and inventory updates helps minimize human error, speeds up transactions, and reduces the workload on librarians and administrative staff.
- **Promoting Transparency:** Features like role-based access control, borrowing history tracking, and real-time updates foster transparency in operations and ensure that sensitive data is accessed only by authorized users.
- **Supporting Data Driven Decisions:** The LMS can generate insights from borrowing trends, usage statistics, and user preferences, helping administrators make informed decisions about inventory management, procurement, and policy planning.
- **Adapting to Institutional Growth:** The scalable design of the system allows for future enhancements such as overdue alerts, integration with external academic databases, recommendation engines, or mobile app extensions—ensuring the platform evolves with the institution's needs.

## **2 . Problem Definition and Requirements**

### **2.1 Problem Statement**

- The traditional management of library resources in many academic institutions remains heavily reliant on manual processes or outdated standalone systems. These methods are often time-consuming, error-prone, and inefficient, resulting in significant challenges for students, teachers, and librarians. Tasks like book searching, borrowing, tracking due dates, and maintaining lending records often require human intervention, which leads to delays, misplaced books, and inconsistent records. Students frequently struggle to locate the availability of specific books, and there's often no system in place for setting return reminders or tracking borrowing history.
- Librarians and staff face their own set of issues. Managing book inventories manually increases the risk of data duplication, loss, and inefficiencies in book issuing and returning. Keeping accurate records of overdue books, penalties, and frequently borrowed items becomes difficult without automation. Additionally, generating reports or identifying usage patterns for decision-making is a complex and resource-heavy process when done manually.
- Another major concern is the lack of role-based access and secure authentication in many existing systems. Without clear differentiation between users (students, teachers, admins), data integrity and access control become serious issues. Unauthorized changes, loss of records, or incorrect book status updates can negatively impact the entire system's credibility.



Given these challenges, there is a clear need for a centralized, web-based Library Management System that addresses the shortcomings of traditional library operations. Such a system should:

- Allow students and teachers to search for books, check availability, borrow and return books, and track lending history through a personalized dashboard.
- Enable librarians and administrators to manage book inventories, monitor borrowing activities, and generate detailed reports on usage and overdue items.
- Provide role-based access control to ensure users only see and interact with data relevant to their roles (e.g., admins can add/remove books; students can only borrow/view).
- Ensure real-time updates, reminders for due dates, and notifications for book availability to enhance user engagement.
- Reduce manual errors, paperwork, and delays by automating all core library operations.

## 2.2 Software Requirements

Frontend:

- HTML, CSS, Tailwind CSS: For responsive and attractive UI.
- JavaScript, React: For building dynamic, interactive user interfaces.

Backend:

- Node.js, Express: For server-side logic, RESTful APIs, and authentication.

Database:

- MongoDB: For storing user profiles, job listings, applications, and feedback.

### Other Functional Requirements:

- The system should support secure user registration and login for students, faculty, and librarians/admins, each with distinct roles and access levels.
- Provide separate dashboards for students/faculty (borrowers) and administrators/librarians. Each dashboard should display role-relevant features such as borrowed books, due dates, and inventory management.
- Users should be able to search the library catalog using filters (e.g., title, author, genre, availability), view book details, and request to borrow or return books through the system.
- Admins/librarians must be able to add, update, or delete book records, manage stock availability, and track borrowing and return status in real-time.
- The system should maintain detailed borrowing history for each user, including due dates, return dates, and any penalties for overdue books.
- A centralized panel for librarians/admins to manage users, view system activity logs, track overdue books, and generate reports (e.g., most borrowed books, inactive users).
- Users should receive alerts and reminders for due dates, successful borrow/return actions, and availability of reserved books.

## 2.3 Hardware Requirements

Minimum Server Requirements:

- Processor: 2+ cores
- RAM: 4 GB or higher (expandable as needed)
- Disk Space: 200 MB minimum for application files, more for database and uploads
- Reliable internet connection for hosting and data transfer

Client Requirements:

- Modern web browser (latest versions of Chrome, Firefox, Edge, Safari)

## 2.4 Data Sets

- Book Catalog: Contains detailed information about each book, including book ID, title, author, publisher, ISBN, genre.
- User profiles: Stores personal and academic information of registered users (students, faculty, and librarians), including user ID, name, role, email, department, borrowing history, fines (if any), and current borrowed books.
- Borrowing Records: Tracks the borrowing and returning history of each user, with data fields such as transaction ID, book ID, user ID, issue date, due date, return date, and fine (if applicable).
- Admin data: Includes verified admin credentials and permissions, as well as logs of administrative actions such as book additions, removals, and user management activities

## 3 . Proposed Design / Methodology

### 3.1 Overview

The proposed **Library Management System (LMS)** is a **full-stack web application** developed using the **MERN stack** (MongoDB, Express.js, React, Node.js), with Tailwind CSS for responsive and modern UI styling. This system is designed to manage and streamline the key operations of a library—such as book catalog management, borrowing/returning workflows, due date tracking, and user account management—by offering role-based dashboards for students, faculty, and administrators. It aims to reduce manual overhead, improve user experience, and maintain accurate, real-time library records.

### 3.2 Methodology

- **Agile Development:**

The project follows Agile methodology, emphasizing iterative development in short cycles (sprints), continuous feedback, and regular releases. This approach allows the team to adapt rapidly to evolving user requirements and market trends, ensuring that the portal remains relevant and user-centric.

- **Competitive Analysis:**

Existing library systems (both manual and digital) were analyzed to identify strengths, weaknesses, and best practices. This informed decisions regarding feature prioritization, user experience design, and performance expectations, helping to ensure a more modern, efficient, and scalable LMS.

- **Feature Finalization:**

Based on the analysis and project objectives, the following **core features** were finalized for implementation:

- **User registration and authentication** for students, faculty, and librarians.
- **Role-based dashboards** with personalized views and access permissions.
- **Book catalog search** with filters (e.g., title, author, genre, availability).
- **Borrowing and return system** with due date tracking and automated fine calculation.
- **Inventory management** for librarians to add, update, or remove book records.
- **Notification system** for due date reminders, overdue alerts, and book availability updates.
- **Admin panel** for system monitoring, report generation, and user management.
- **Feedback and support module** to collect user suggestions and handle issues.

### 3.3 System Architecture

Three-Tier Architecture:

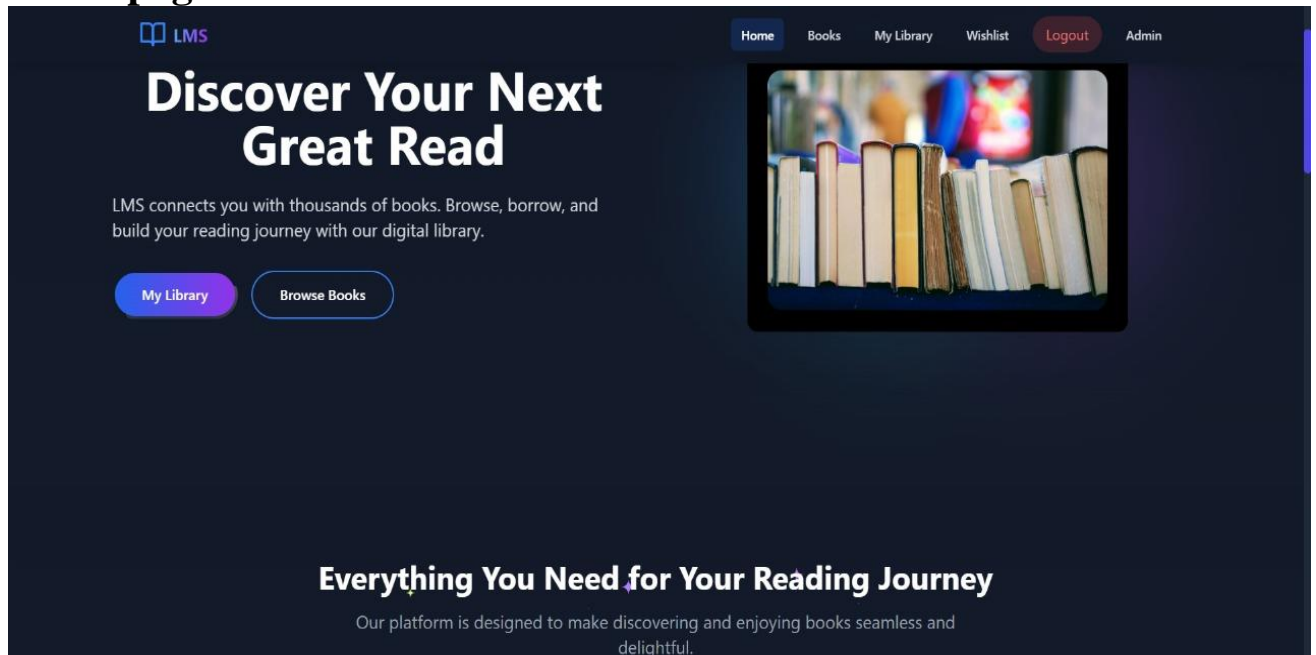
- **Presentation Layer:** React + Tailwind CSS for a responsive and interactive user interface.
- **Application Layer:** Node.js with Express.js providing RESTful APIs for business logic and data processing.
- **Data Layer:** MongoDB for storing structured (user profiles, job postings) and unstructured data (resumes, job descriptions)

#### **Key Benefits of This Architecture:**

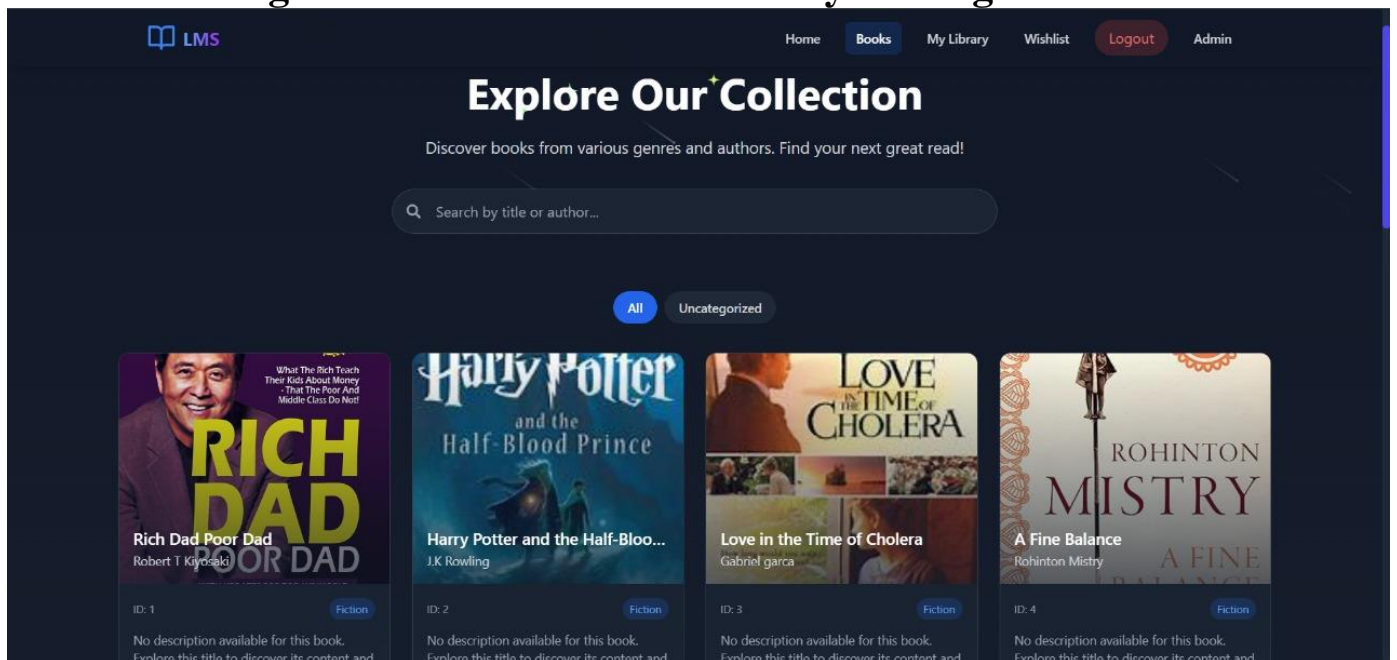
- **Separation of Concerns:** Each tier can be developed, deployed, and scaled independently, improving reliability and maintainability.
- **Security:** The data tier and presentation tier never communicate directly; all requests pass through the application tier, reducing attack surfaces.
- **Scalability and Flexibility:** Each layer can be optimized or scaled based on demand, ensuring the portal can handle growth and evolving requirement

## 4. Results

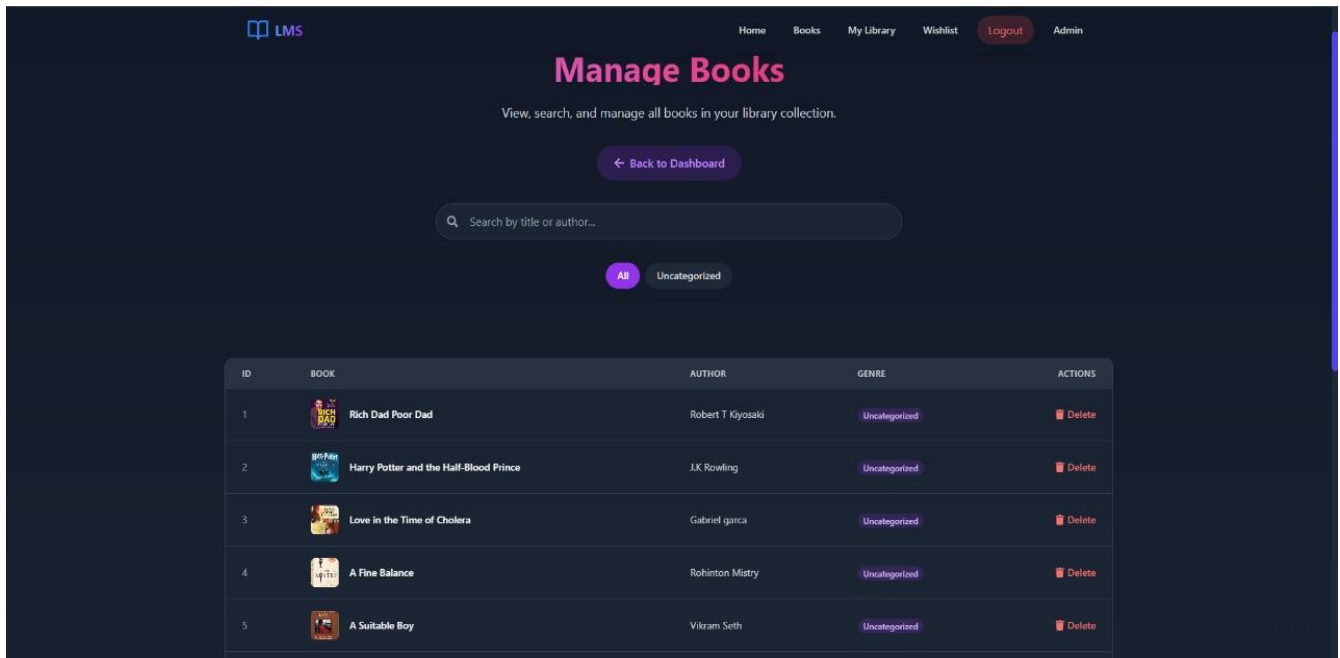
### Home page



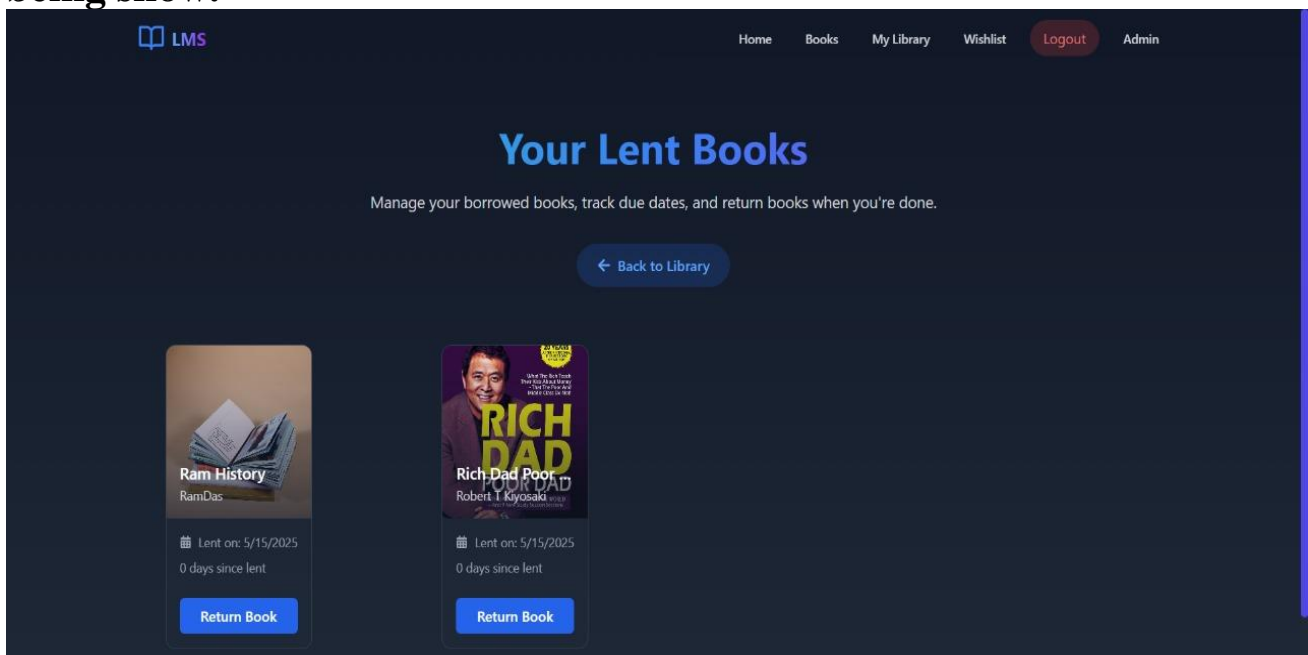
### Books Catalog : Here All the books of library is being shown.



**Page After Login : It is the page that appears after login (Valid Credentials).**

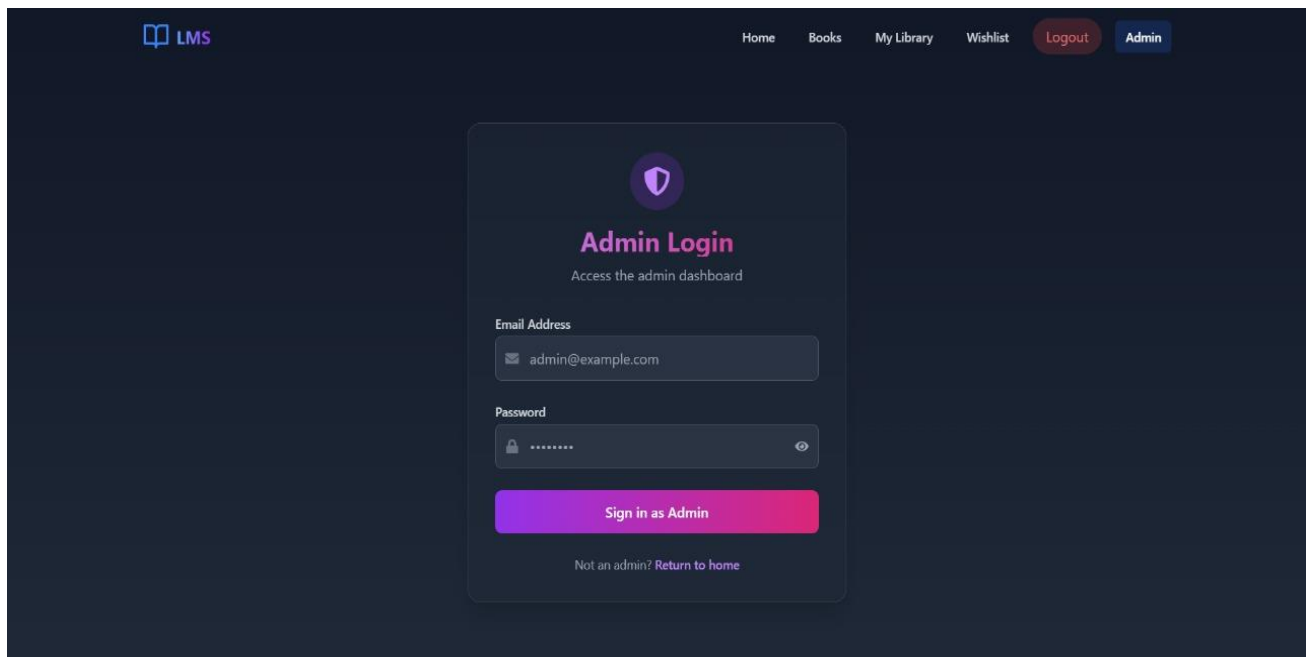


**Lending Books page of the user : Here the books lentend by the user is being show.**





## Admin Dashboard :



The Admin Login form is centered on a dark blue background. It features a purple shield icon at the top, followed by the title 'Admin Login' and the subtitle 'Access the admin dashboard'. The form includes two input fields: 'Email Address' with the placeholder 'admin@example.com' and 'Password' with a masked input '\*\*\*\*\*'. A purple 'Sign in as Admin' button is positioned below the password field. At the bottom, a link 'Not an admin? Return to home' is displayed. The top navigation bar contains links for 'Home', 'Books', 'My Library', 'Wishlist', 'Logout', and 'Admin'.

**Admin Login**  
Access the admin dashboard

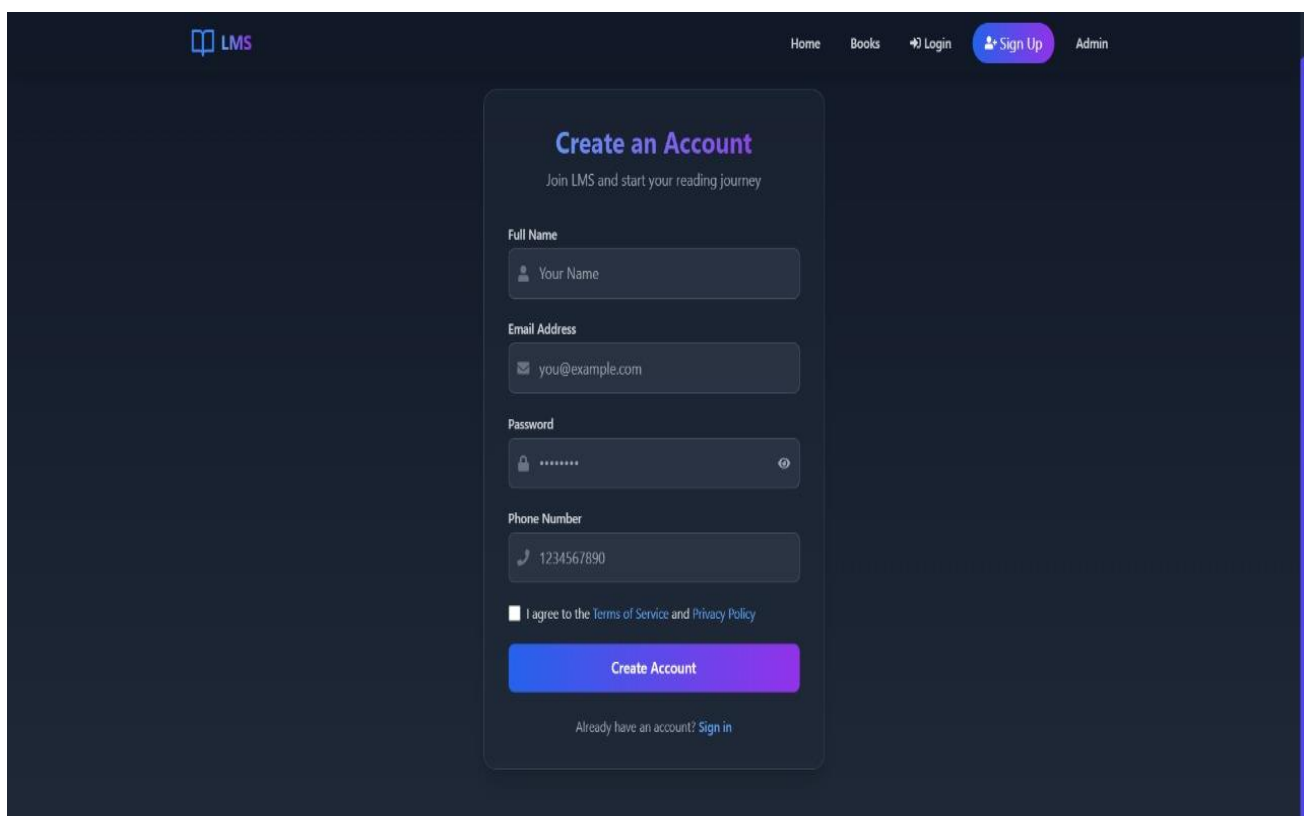
Email Address  
admin@example.com

Password  
\*\*\*\*\*

Sign in as Admin

Not an admin? [Return to home](#)

## Signup Page : Here a new user can make his account.



The 'Create an Account' form is centered on a dark blue background. It features a purple shield icon at the top, followed by the title 'Create an Account' and the subtitle 'Join LMS and start your reading journey'. The form includes four input fields: 'Full Name' with the placeholder 'Your Name', 'Email Address' with the placeholder 'you@example.com', 'Password' with a masked input '\*\*\*\*\*', and 'Phone Number' with the placeholder '1234567890'. A checkbox labeled 'I agree to the Terms of Service and Privacy Policy' is located below the phone number field. A purple 'Create Account' button is positioned below the checkbox. At the bottom, a link 'Already have an account? Sign in' is displayed. The top navigation bar contains links for 'Home', 'Books', 'Login', 'Sign Up', and 'Admin'.

**Create an Account**  
Join LMS and start your reading journey

Full Name  
Your Name

Email Address  
you@example.com

Password  
\*\*\*\*\*

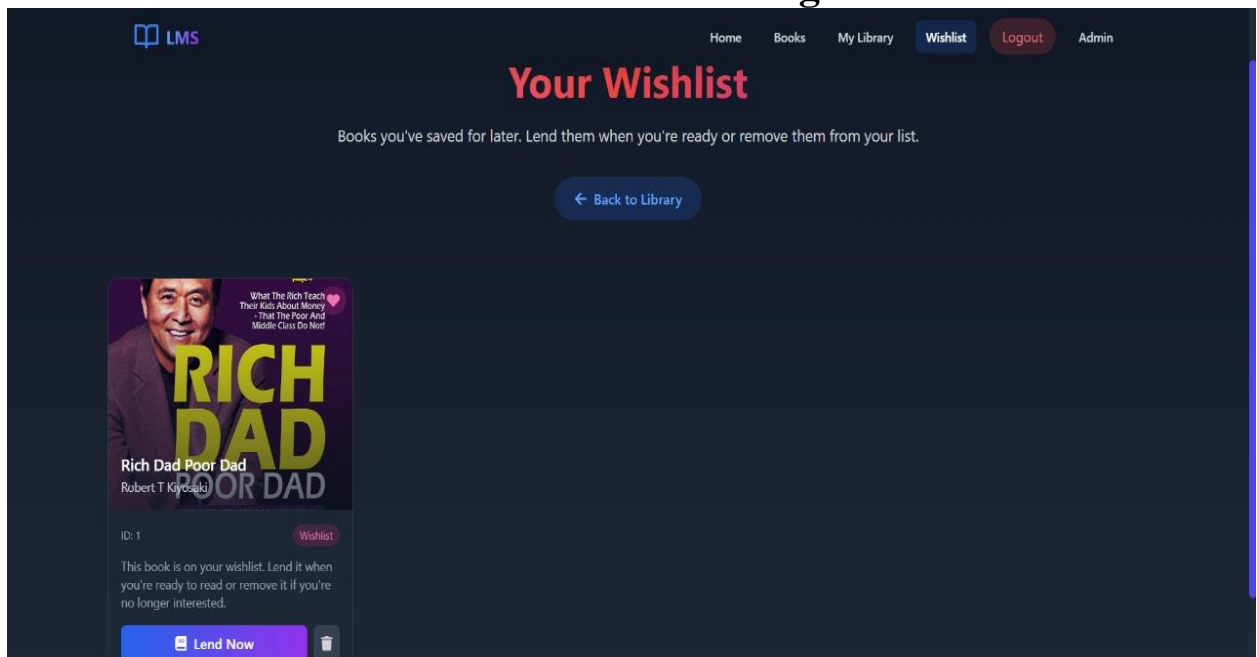
Phone Number  
1234567890

☐ I agree to the Terms of Service and Privacy Policy

Create Account

Already have an account? [Sign in](#)

## Books Data : Here the Data of Books is being shown



# KEY FEATURES

## 1. User Registration & Login

- Separate registration for students, faculty, and librarians/admins, each with distinct roles and access levels.
- Secure login system with role-based access control to protect user accounts and data.

## 2. Book Search and Catalog

- Users can browse and search the book catalog using filters like title, author, genre, and availability.
- Book details page to view more information about each item, including availability and location.

## 3. Book Borrowing & Returning

- Students and faculty can borrow and return books directly through the system, with due date tracking.
- Librarians can monitor borrowed books and track overdue items.

## 4. Inventory Management

- Admins/librarians can add, update, or remove books from the library catalog.
- Maintain accurate records of book stock, including available and issued copies.

## 5. Simple Dashboard

- Students : View borrowed books, due dates, and fines (if applicable).
- Admins: View and manage user accounts, book stock, and transactions.

## Conclusion

Developing this **Library Management System (LMS)** has provided valuable insights into the essential requirements and challenges involved in building an efficient and user-friendly system for managing library operations. Throughout the project, I focused on implementing the core features necessary to streamline the library's daily functions, ensuring a smooth experience for students, faculty, and administrators.

The LMS successfully allows users to register with specific roles (students, faculty, and librarians) and offers role-based access to features. Students and faculty can easily borrow and return books, track due dates, and manage their personal borrowing history, while librarians can update the book catalog, track overdue books, and manage user profiles. The system also ensures accurate record-keeping by maintaining real-time data on book availability and user transactions.

The inclusion of a simple yet effective dashboard for both students and administrators allows users to easily view and manage their borrowed books, fines, and personal profiles. The secure login and logout system enhances privacy and ensures that users' personal information is protected.

While this project covers the basic features of an LMS, it also lays the groundwork for future enhancements such as advanced search filters, automated notifications for due dates and overdue books, and the implementation of detailed analytics for better library management. These improvements could further increase the system's efficiency and user engagement.

In conclusion, this project not only demonstrates the practical application of web development skills but also emphasizes the importance of designing intuitive interfaces, maintaining secure data management, and providing an accessible platform for users. The experience gained through this project will be instrumental in tackling more complex library management systems or other web applications in the future.

## References

1. <https://www.foundit.in/>
2. <https://www.glassdoor.co.in/>
3. <https://fonts.googleapis.com>
4. <https://fonts.gstatic.com>