Panipat Institute of Engineering and Technology Samalkha, Panipat

Department of Computer Science and Engineering

(Artificail Intelligence and Machine Learning)

Tech T Report File
(Library Management System)

Submitted to:
Ms. Richa
Assistant Professor

Submitted by:
Shubham Ruhal(2823430)
Vaibhav(2823380)
Ajay(2823413)
Anurag(2823325)
B. Tech CSE – AI&ML
4th Semester

Affiliated to:

KURUKSHETRA UNIVERSITY, KURUKSHETRA, INDIA

1. Introduction

The Library Management System (LMS) is a web-based application designed to streamline and automate the management of library resources. It facilitates seamless transactions such as adding, removing, listing, issuing, and returning books while also managing user authentication and book cover images. Built with Django for the backend and React for the frontend, the LMS features a modern, responsive user interface optimized for ease of use.

Libraries often struggle with manual book tracking, resource management, and efficient member services. LMS addresses these challenges by providing an automated, digital platform that reduces manual work, improves accessibility, and enhances user experience for both library staff and members.

2. Objectives

- Efficient Library Operations: Simplify the process of managing library resources and transactions.

- User-friendly Interface: Ensure an intuitive and responsive UI for both librarians and users.

- Secure Transactions: Maintain data integrity and security during book transactions and user authentication.

- Real-time Tracking: Enable real-time updates on issued and available books.

3. Technologies Used

- Backend: Django, Django REST Framework

- Frontend: React, TypeScript, Tailwind CSS

- Database: SQLite

- Version Control: Git & GitHub

- Other Tools: Axios (API calls), React Router (Navigation), Framer Motion (Animations)

4. System Architecture

The LMS architecture is divided into two main components:

- Frontend (React): Handles user interface, form submissions, and API interactions.
- Backend (Django): Manages database operations, authentication, and API endpoints.

The frontend communicates with the backend via REST API calls, enabling CRUD operations and data synchronization.

5. Database Design

The database consists of the following models:

Book Model:

- title: Title of the book
- author: Author's name
- description: A brief summary of the book
- published_date: Date of publication
- is_issued: Status of the book (issued or not)
- issued_to: ForeignKey to User (if issued)
- issued_date: Date of issue
- cover_image: Image of the book cover

User Model:

- Uses Django's built-in User model for authentication and permissions.

6. Key Features

- Book Management: Add, remove, and list books

- Issue and Return Books: Easy management of book transactions

- User Authentication: Secure login and registration

- Book Cover Image Upload: Supports image uploads for better visual management

- Responsive UI: Accessible across devices

7. Implementation Overview

- Backend (Django): API views, serializers, and models are implemented to handle CRUD operations securely.

- Frontend (React): User-friendly components for managing books, issuing, and returning operations.

- State Management: Managed with React hooks for seamless UI updates.

- Security Measures: CSRF protection, CORS handling, and token-based authentication.

8. How to Run the Project

Prerequisites:

- Python 3.x

- Node.js & npm

Backend (Django):

1.Navigate to the `backend` directory:

```
cd backend
```

2. Install dependencies:

```
pip install -r requirements.txt
```

3. Apply migrations:

```
python manage.py migrate
```

4.Start the Django server:

```
python manage.py runserver
```

Frontend (React):

1.Navigate to the `frontend` directory:

```
cd frontend
```

2. Install dependencies:

```
npm install
```

3. Start the React server:

```
npm run dev
```

Access the Application:

- Backend API: http://localhost:8000/

- Frontend app: http://localhost:5173/

9. Challenges Faced

- Handling image uploads for book covers efficiently

- Managing session-based authentication

- Synchronizing frontend and backend data

10. Future Improvements

- Implement search and filter capabilities

- Role-based access control for better security

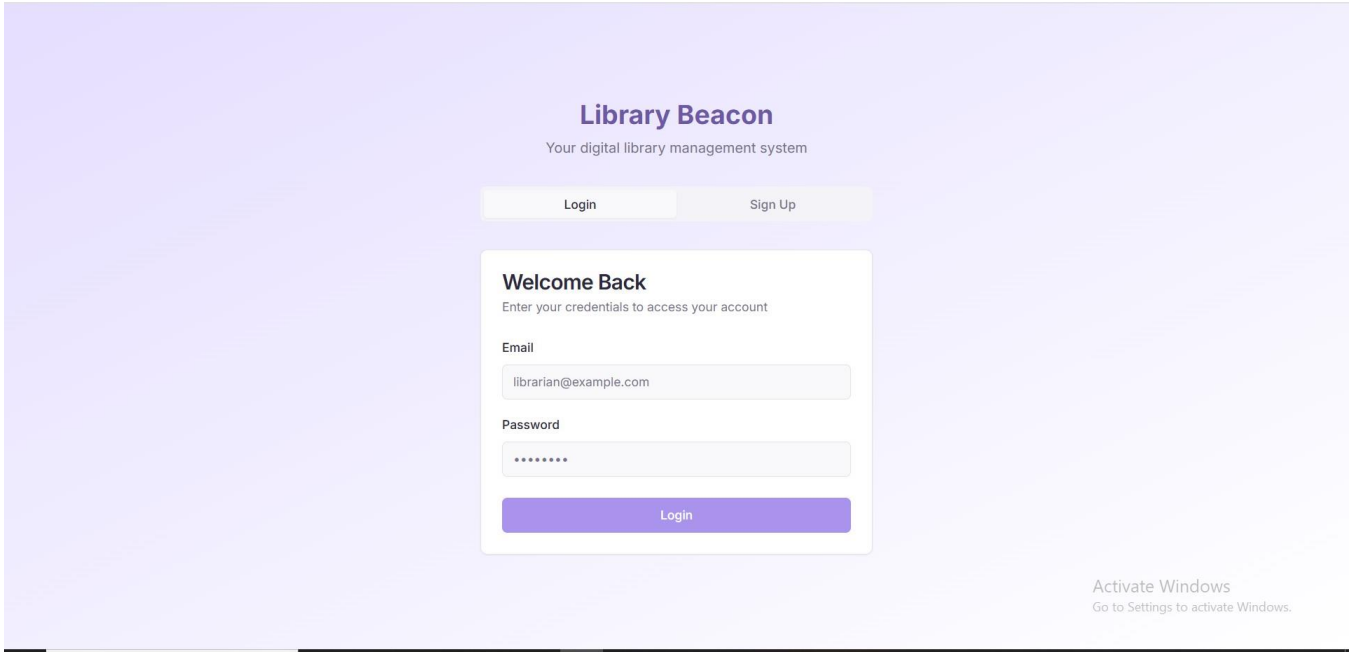
- Email notifications for due dates and reservations

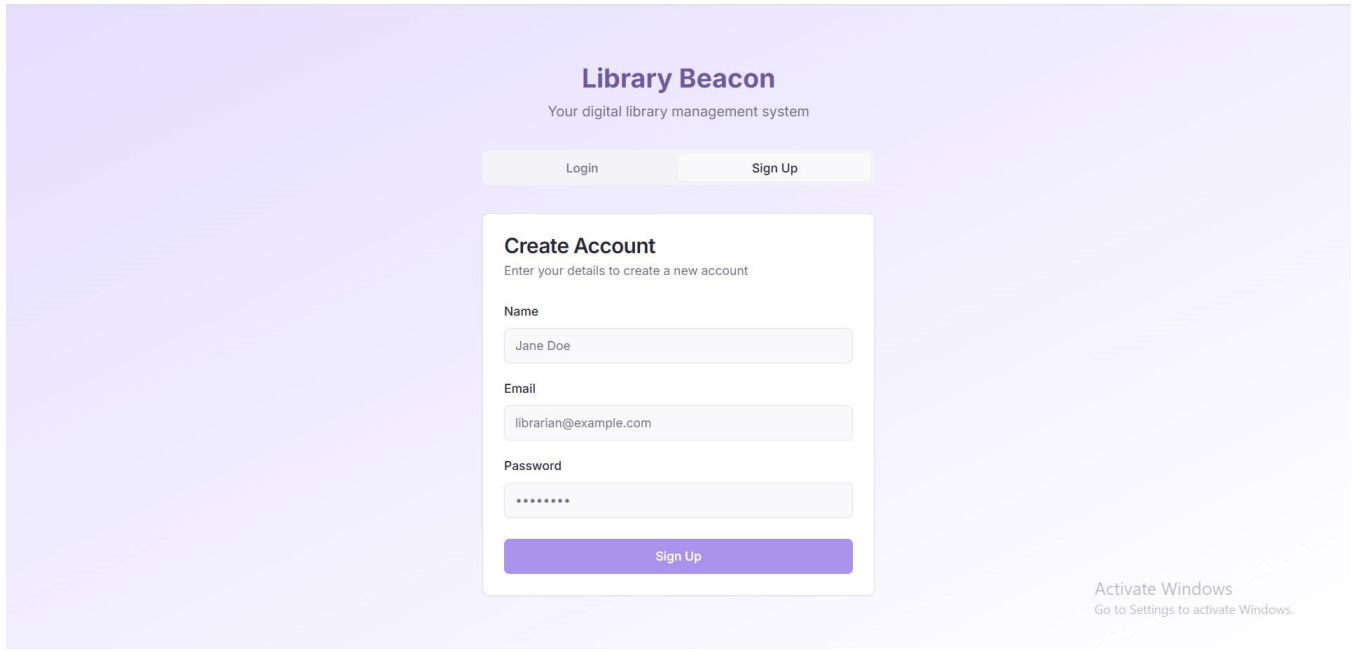
- Analytics dashboard for book statistics and usage

11. Conclusion

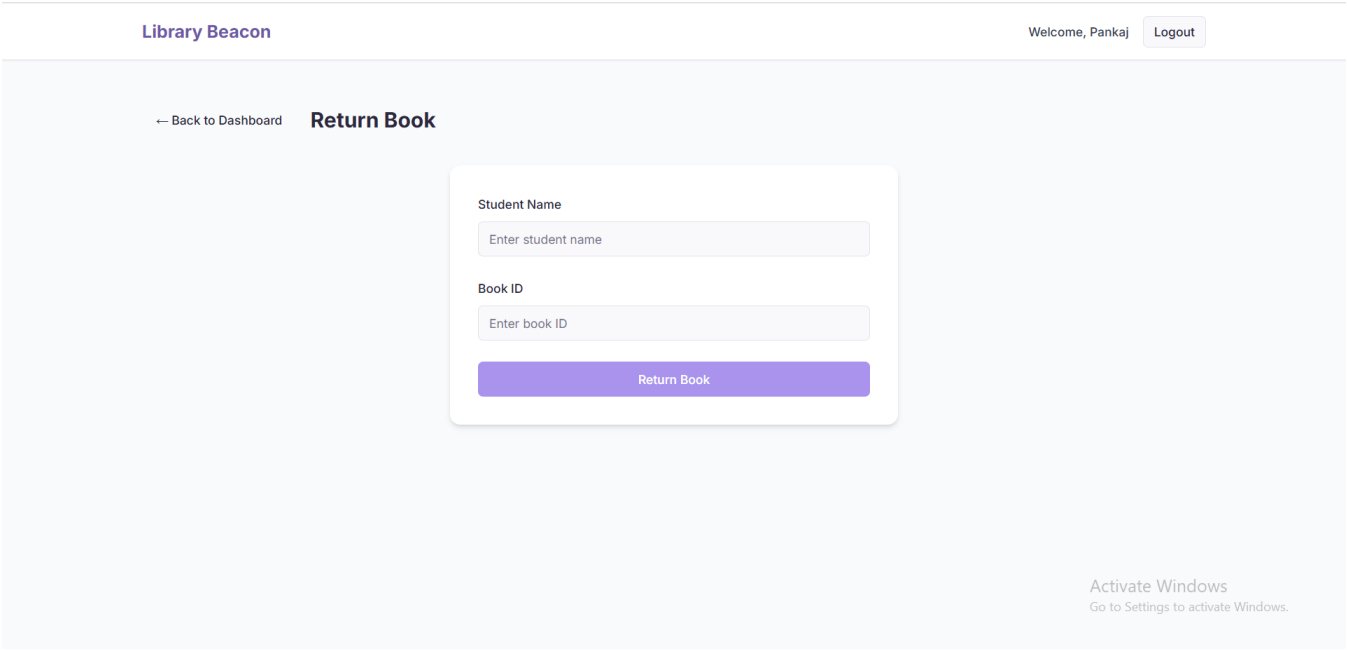
The Library Management System effectively digitizes and streamlines library operations. Its scalable architecture and user-friendly interface make it a valuable tool for modern libraries, with room for further expansion and feature enhancements.

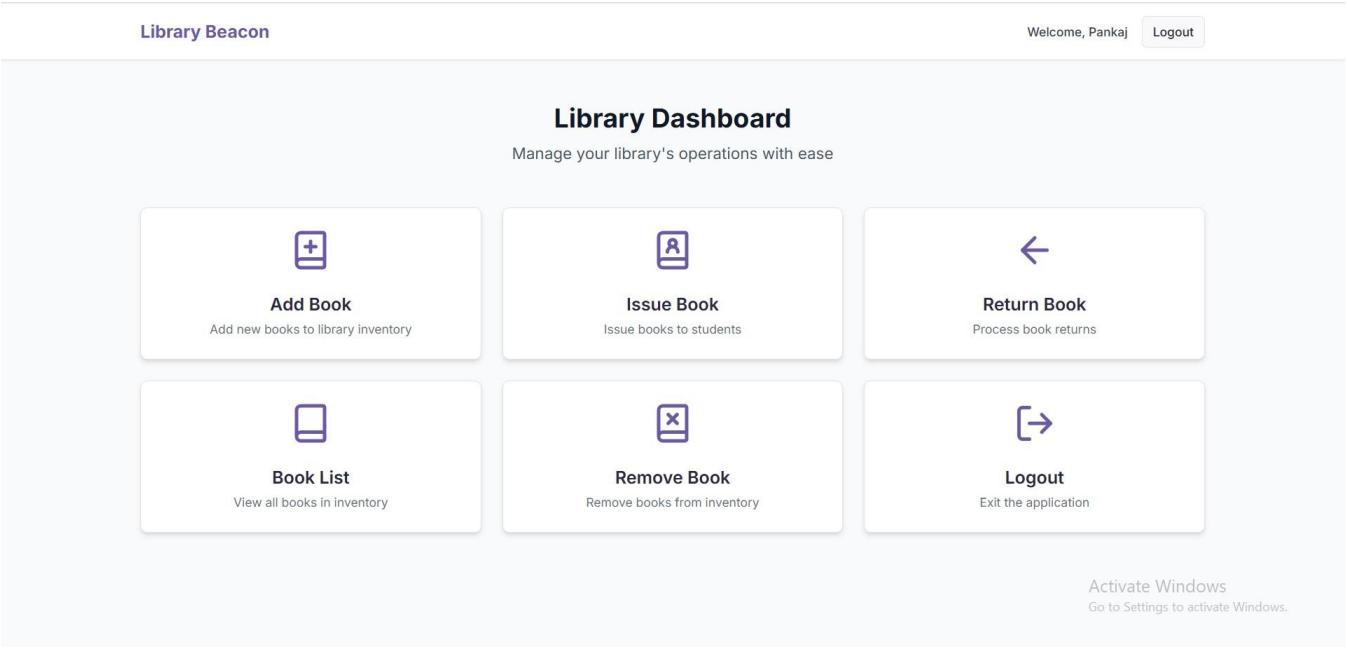
12. User Interface Screenshots



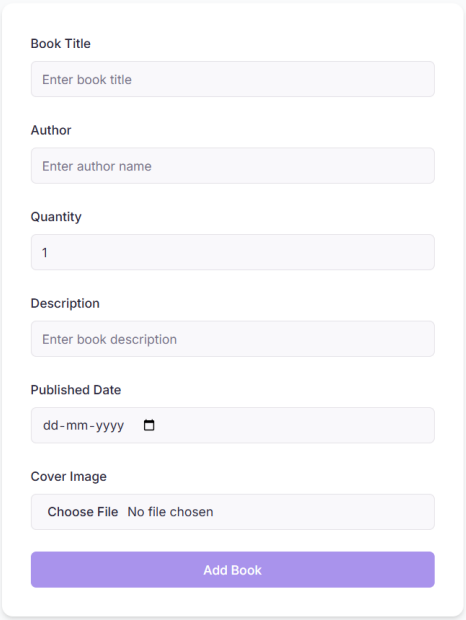


12. User Interface Screenshots

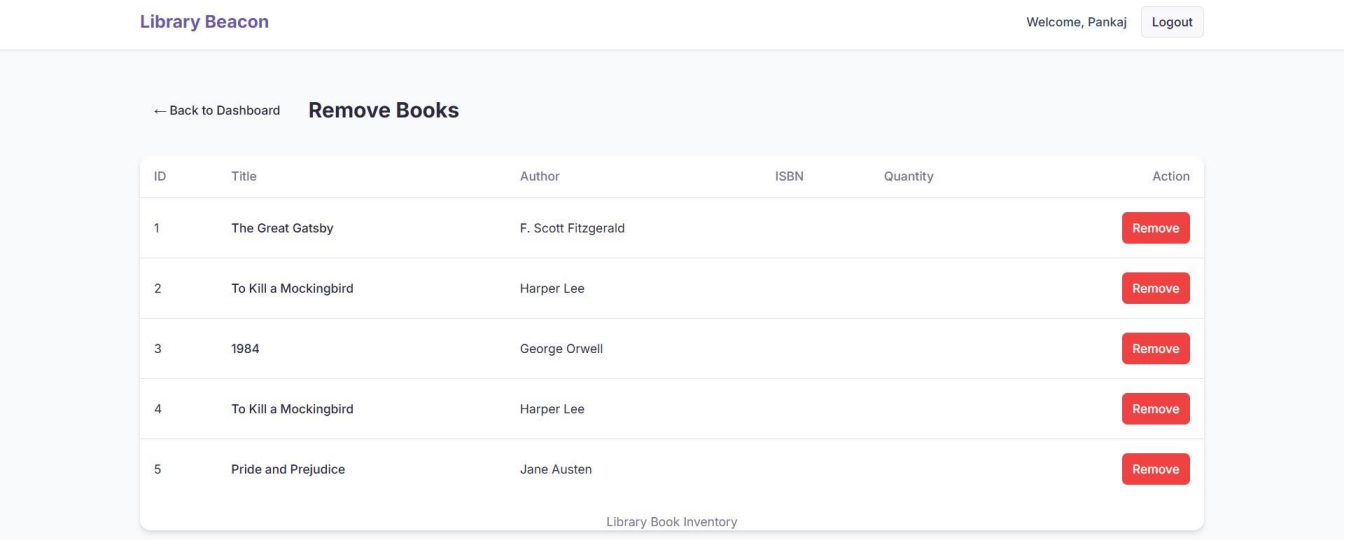




12. User Interface Screenshots



Activate Windows
Go to Settings to activate Windows.



Activate Windows
Go to Settings to activate Windows.

Library Beacon

Welcome, Pankaj Logout

← Back to Dashboard

Book Inventory

+ Add Book

ID	Title	Author	ISBN	Quantity
1	The Great Gatsby	F. Scott Fitzgerald		
2	To Kill a Mockingbird	Harper Lee		
3	1984	George Orwell		
4	To Kill a Mockingbird	Harper Lee		
5	Pride and Prejudice	Jane Austen		

Library Book Inventory

Activate Windows
Go to Settings to activate Windows.