**LibraCore – Library Management System Project Report**

**Title Page**

Project Title: LibraCore – Library Management System for a college

Institution / Organization: IBM Summer Internship Program Mentor/Supervisor: Mr. Jaivik Panchal Submission Date: 21 July 2025

**Certificate**

This is to certify that the project titled “LibraCore – Library Management System for a college” is an original work completed by the Hely Prajapati as part of the Full Stack Development Internship under the guidance of Mr. Jaivik Panchal.

**Acknowledgment**

We would like to express our sincere gratitude to everyone who supported and guided us throughout the development of this Library Management System project.

First and foremost, we are deeply thankful to our project guide and faculty members for their valuable suggestions, constant encouragement, and expert guidance, which were instrumental in the successful completion of this project.

We also extend our heartfelt thanks to our peers and teammates for their collaborative efforts, ideas, and constructive feedback. Their contributions have been vital in shaping the system effectively.

Lastly, we would like to acknowledge the support and resources provided by our institution, which enabled us to explore, design, and implement this project in a real-world context.

**Table of Contents**

1. Abstract
2. Objective
3. Technology Stack
4. Modules
5. Frontend Development
6. Backend Development
7. Database Design
8. Data Flow Diagrams
9. Security Measures
10. Limitations
11. Future Enhancements
12. Screenshots

**1. Abstract**

The Library Management System is a full-stack web application designed to automate the day-to-day operations of a college or school library.  
It simplifies book issuance, return tracking, student management, and fine calculation.

* **What**: A digital solution for managing library operations.
* **Why**: To reduce manual workload, prevent data inconsistency, and offer a better user experience.
* **Key Features**:
  + Student/Admin login
  + Book issue/return tracking
  + Fine calculation
  + Book Catalog
  + Feedback system

**2. Objective**

The primary objective of this project is to design and develop a web-based Library Management System that streamlines and automates the core functions of a library. This includes user registration, book inventory management, book issue/return tracking, fine calculation, and user feedback.

The system aims to minimize manual work, reduce errors, ensure data accuracy, and provide an intuitive interface for both students and administrators. By leveraging modern web technologies, the project enhances efficiency, accessibility, and transparency in library operations.

**3. Technology Stack**

| **Layer** | **Technology Used** |
| --- | --- |
| Frontend | HTML, CSS, JavaScript, Bootstrap |
| Backend | Node.js, Express.js |
| Database | MongoDB |
| Tools | GitHub, VS Code, MongoDB Compass |

**4. Modules/Features**

**1. Authentication**

* Login/Signup with role-based access (Admin/Student)
* Technologies: Node.js, Express, MongoDB

**2. Admin Panel**

* Manage books, students, issue/return logs

**3. Student Dashboard**

* View issued books, return dates, and fines
* Submit feedback

**4. Book Management**

* Add, edit, delete books
* search by department

**5. Issue/Return System**

* Track issued books
* Calculate fines for late return

**5. Frontend Development**

* **Framework**: HTML, CSS, Vanilla JS (Bootstrap for styling)
* **Layout**: Modular pages – login.html, dashboard.html, issuebook.html, explore Books, fine calculation, profile
* **UI/UX**: Simple, teal-themed interface; user-focused design

6. **Backend Development**

* **Framework**: Node.js with Express.js
* **API Style**: RESTful API
* **Controllers**:
  + auth.js: login/signup
  + books.js: add/update/delete books
  + issue.js: issue/return operations
  + feedback.js: share user experience
  + fine.js: calculate fine for late returns

7. **Database Design**

* **DBMS**: MongoDB
* **ER Diagram**: (Can be drawn using dbdiagram.io or manually)
* **Collections**:
  + users: {name, email, password, role, studentId, department}
  + books: {bookId, title, author, department, status}
  + issuedBooks: {studentId, bookId, issueDate, returnDate}
  + fines: {studentId, bookId, fineAmount}
  + feedback: {studentId, message}
* **Relationships**:
  + One student → Many issuedBooks
  + One book → One or zero issued record at a time

8. **Data Flow Diagrams**

**Level 0**

* User → Login → Dashboard (via Backend)
* Admin → Manage Students/Books
* Student → Issue/Return → DB update

**Level 1**

* Issue Book: studentId/bookId → fetch book → validate → update status → insert into issuedBooks
* Return Book: fetch record → calculate fine → update DB

9. **Security Measures**

* **Encryption**: Passwords hashed using Bcrypt
* **Authentication**: JWT token for sessions
* **Validation**: Input validation using middleware

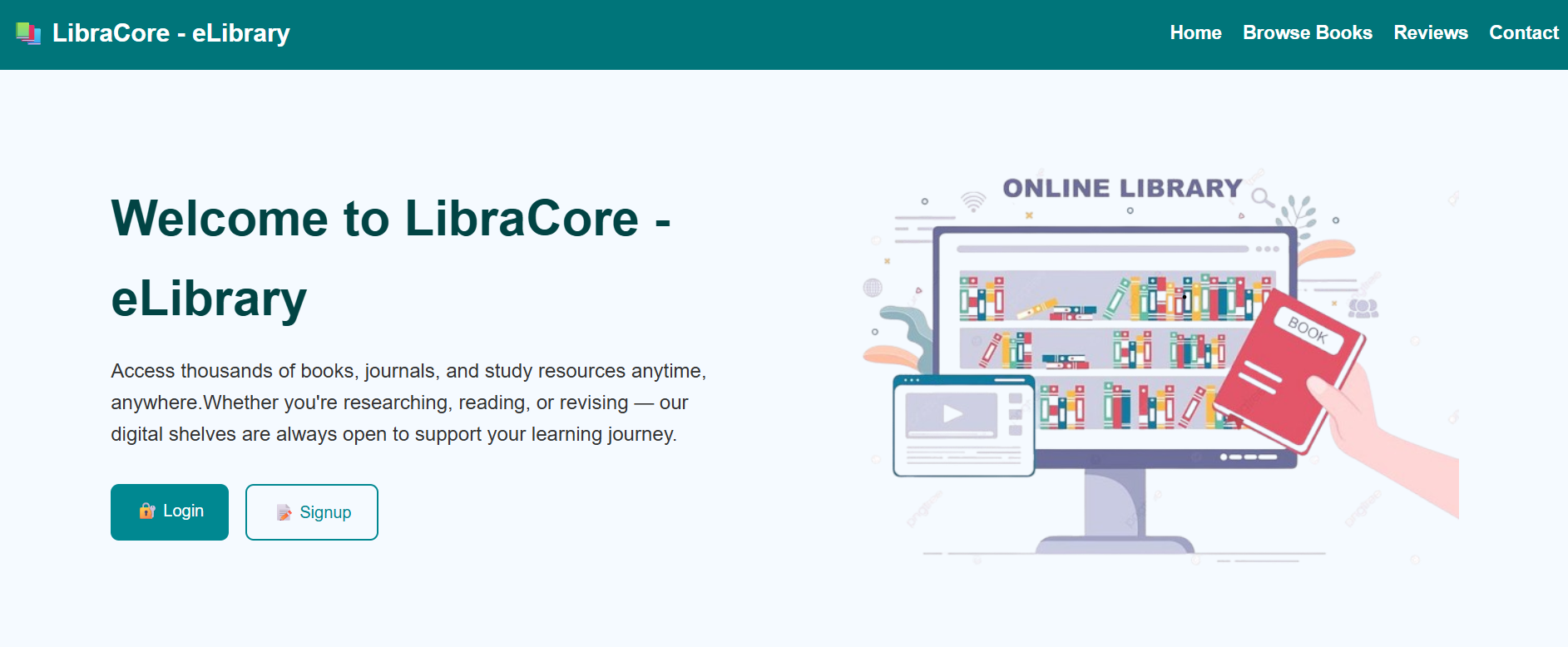
10. **Limitations**

* No real-time notifications or alerts
* No role creation from UI
* Works best on desktop (not responsive)
* Books Available only for computer department

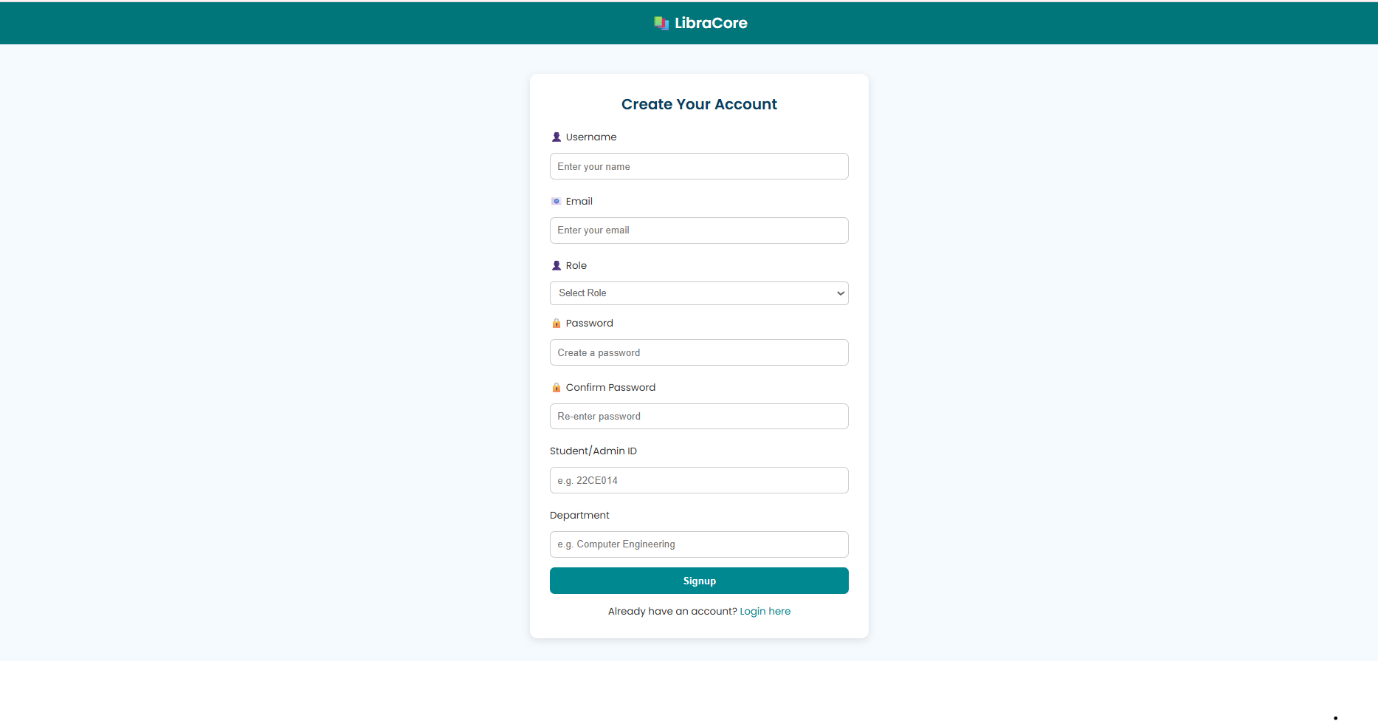
11. **Future Enhancements**

* Mobile responsive design
* QR code for book scanning
* Email reminders for due books
* Admin analytics dashboard
* Advanced Search & Filtering
* Real-time Availability Tracking
* AI-Based Recommendations

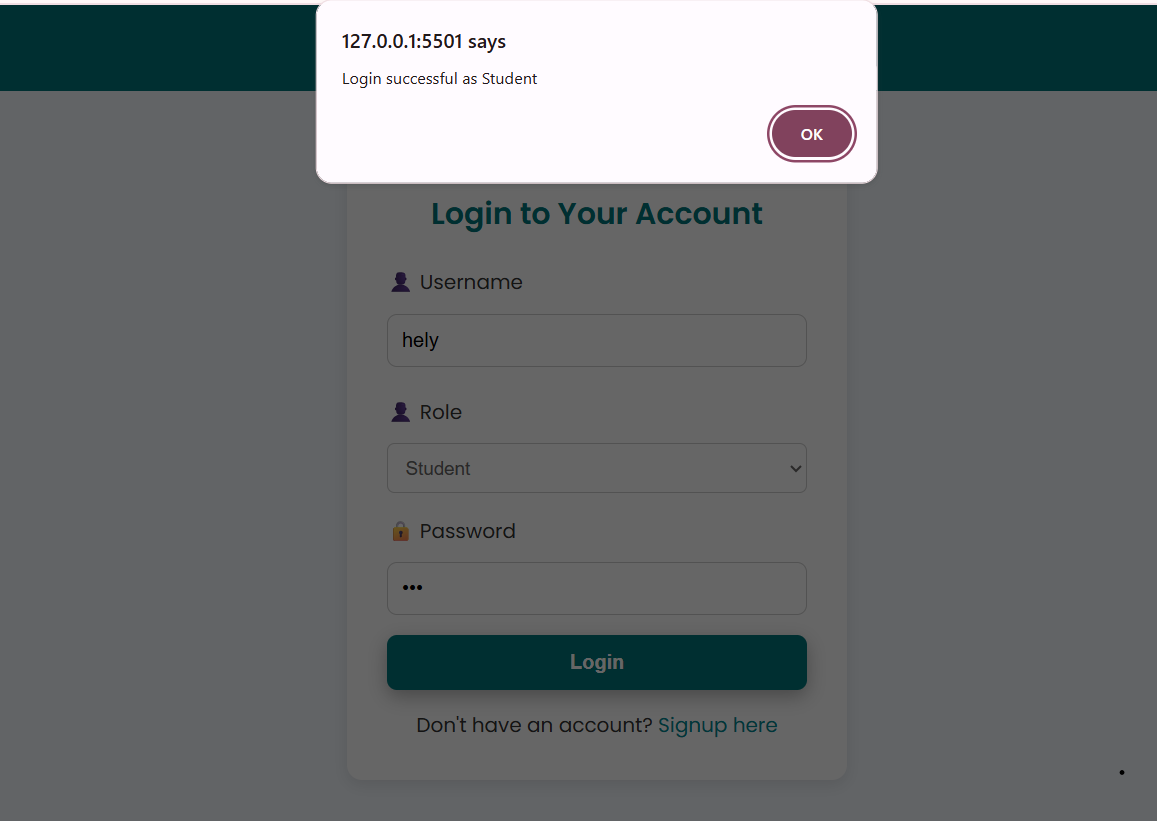
**12.Screenshots**

****

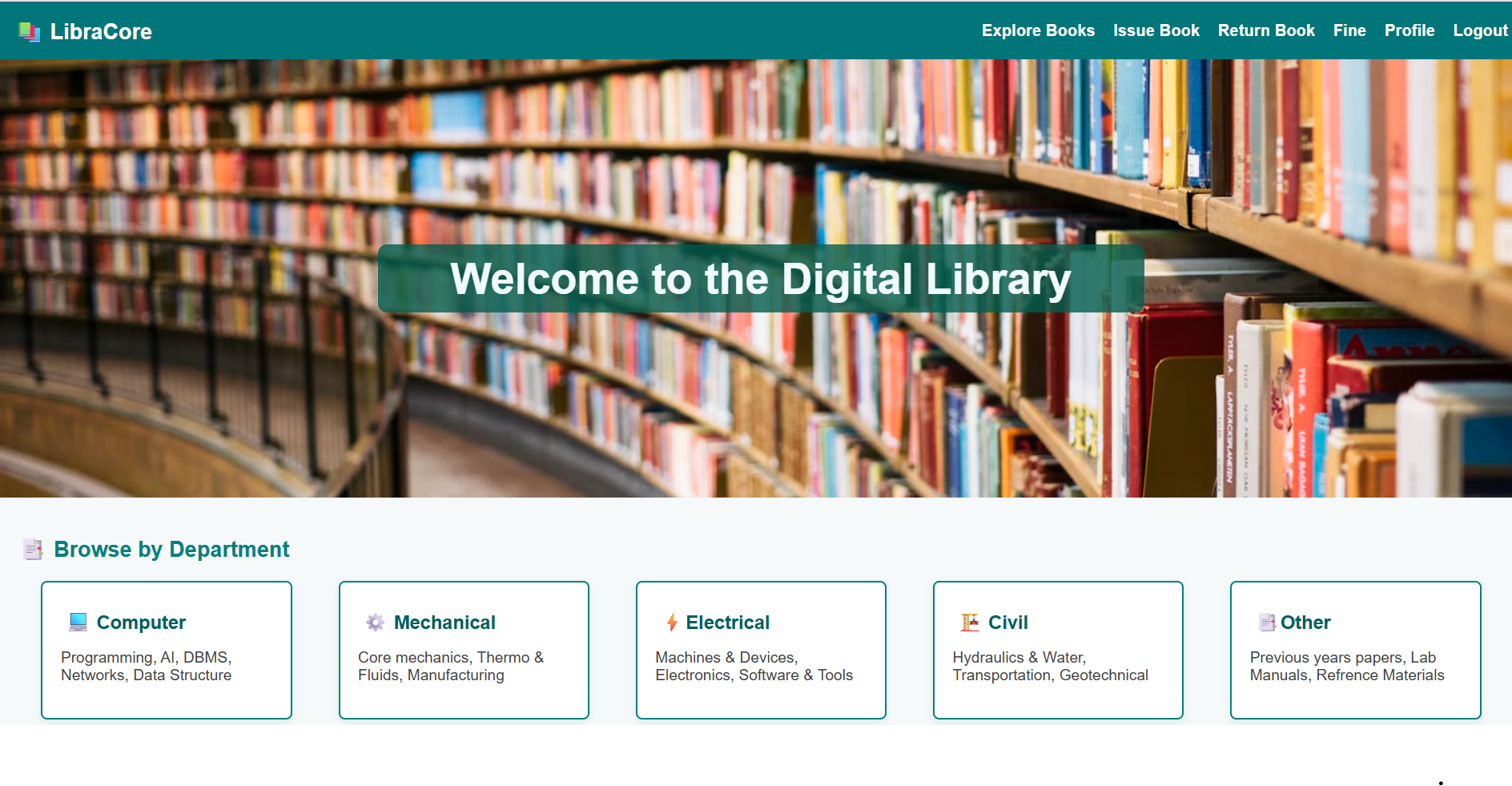
Landing Page



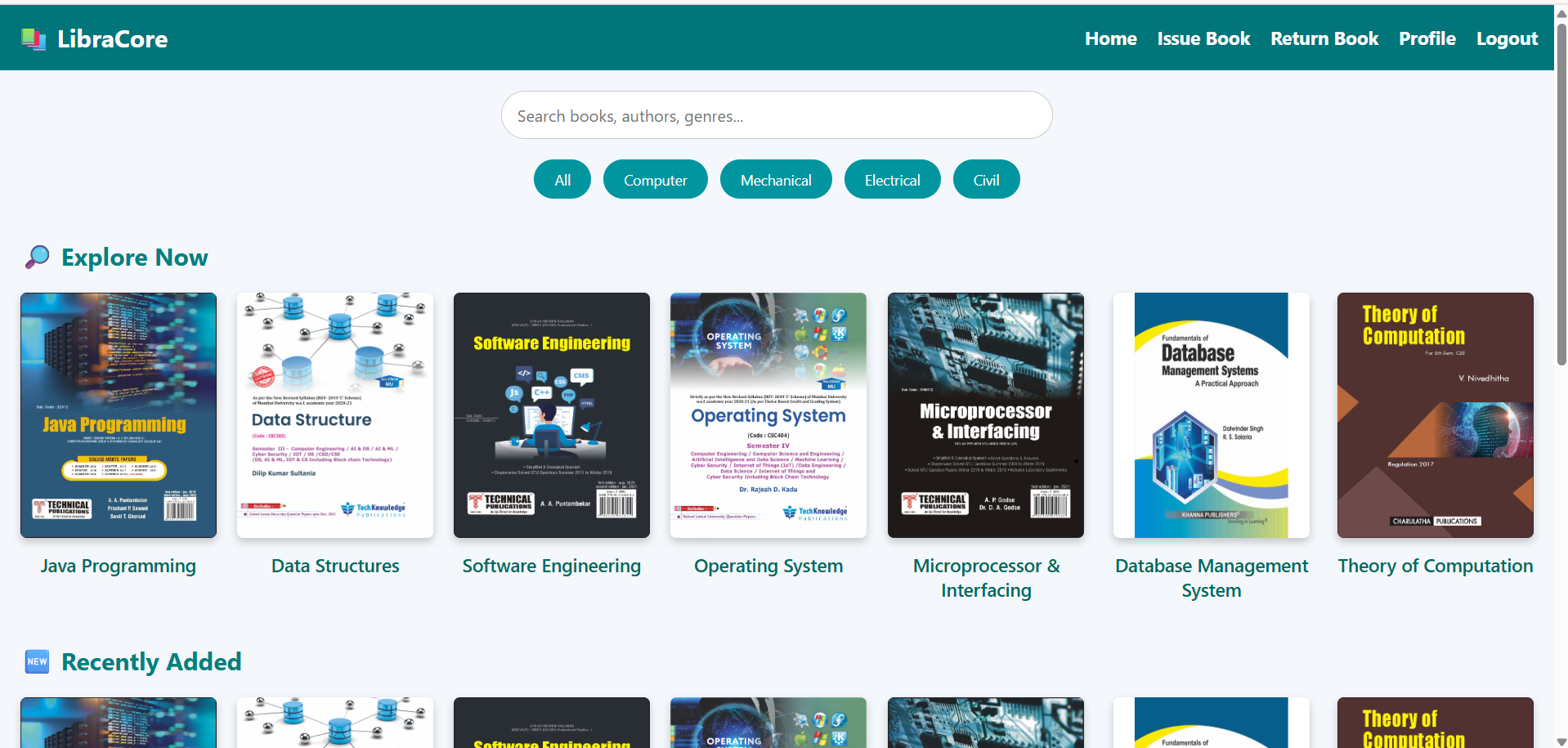
Signup Page



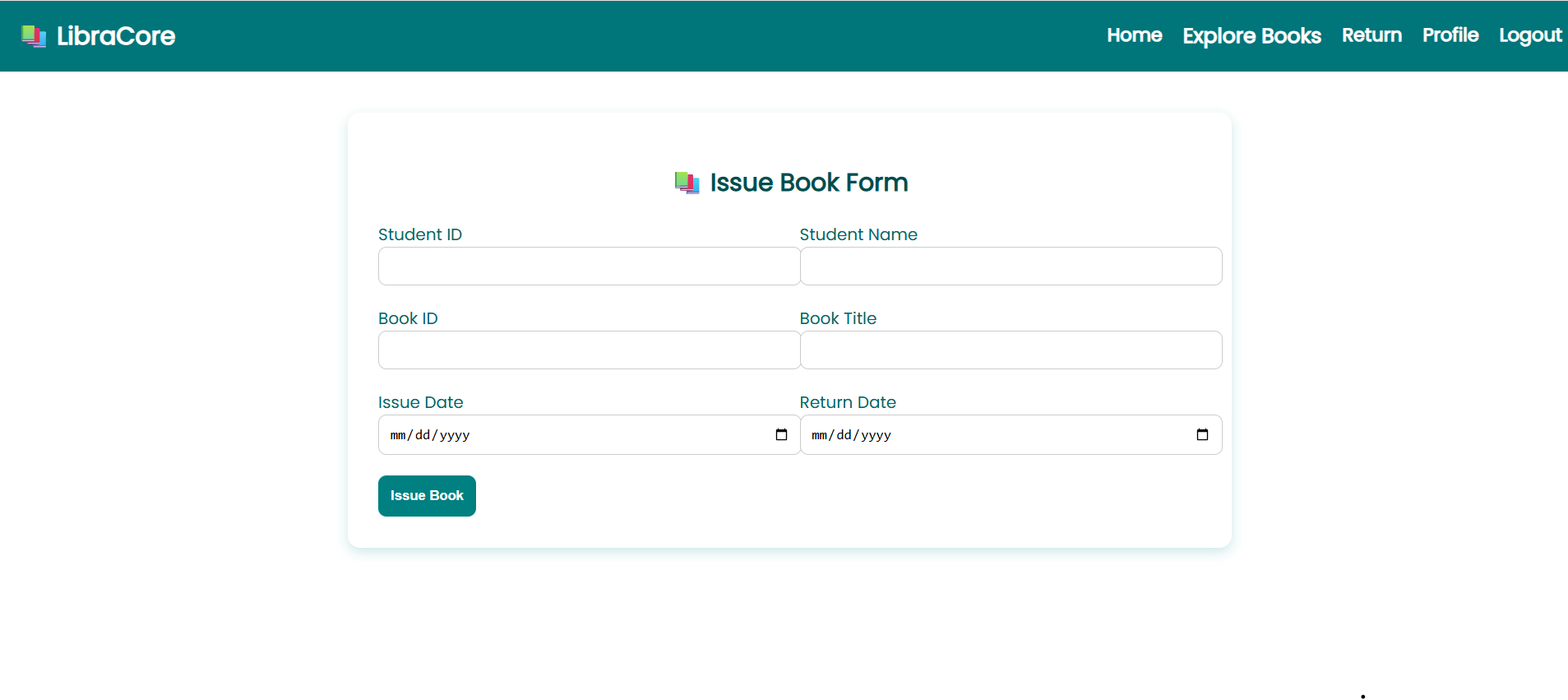
Successful Login



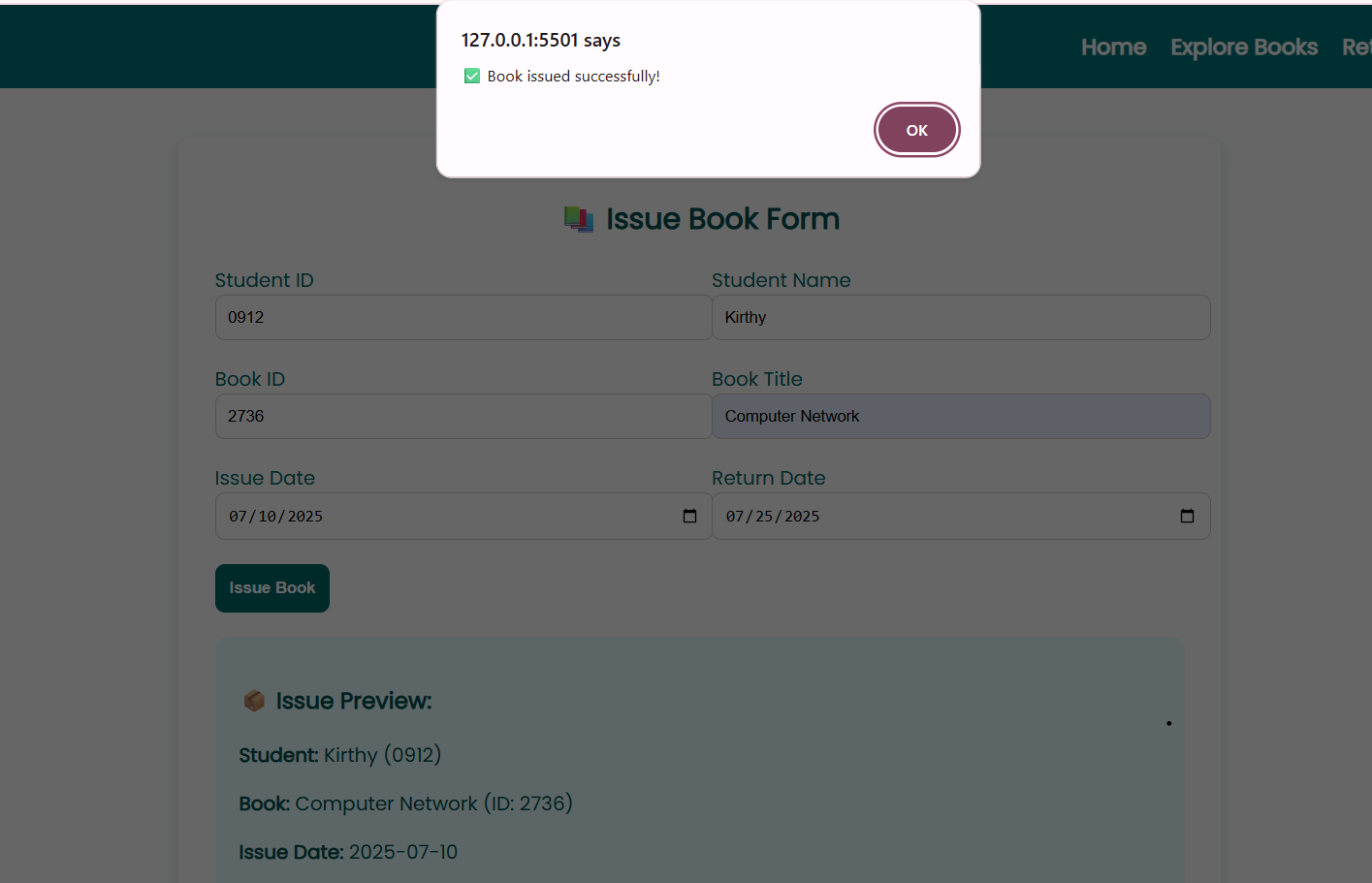
Index Page



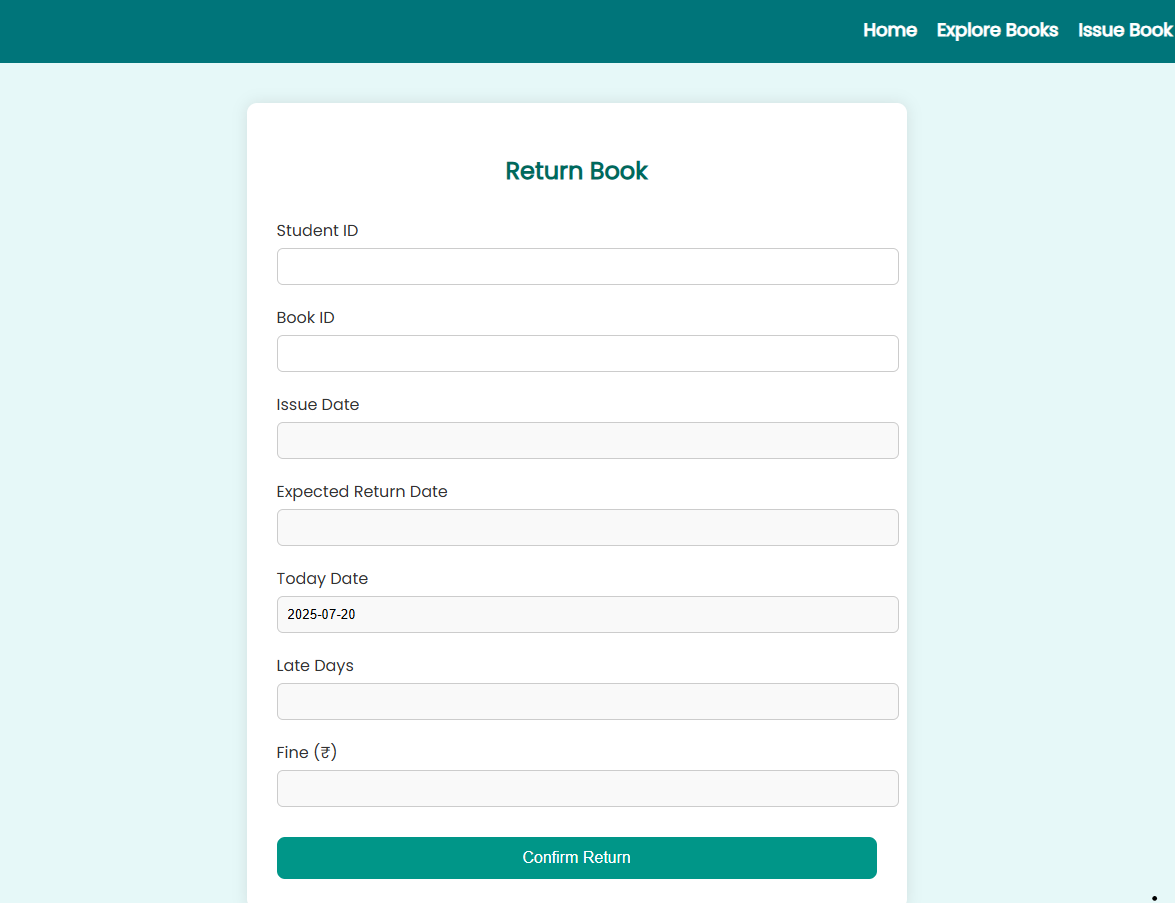
Explore Books



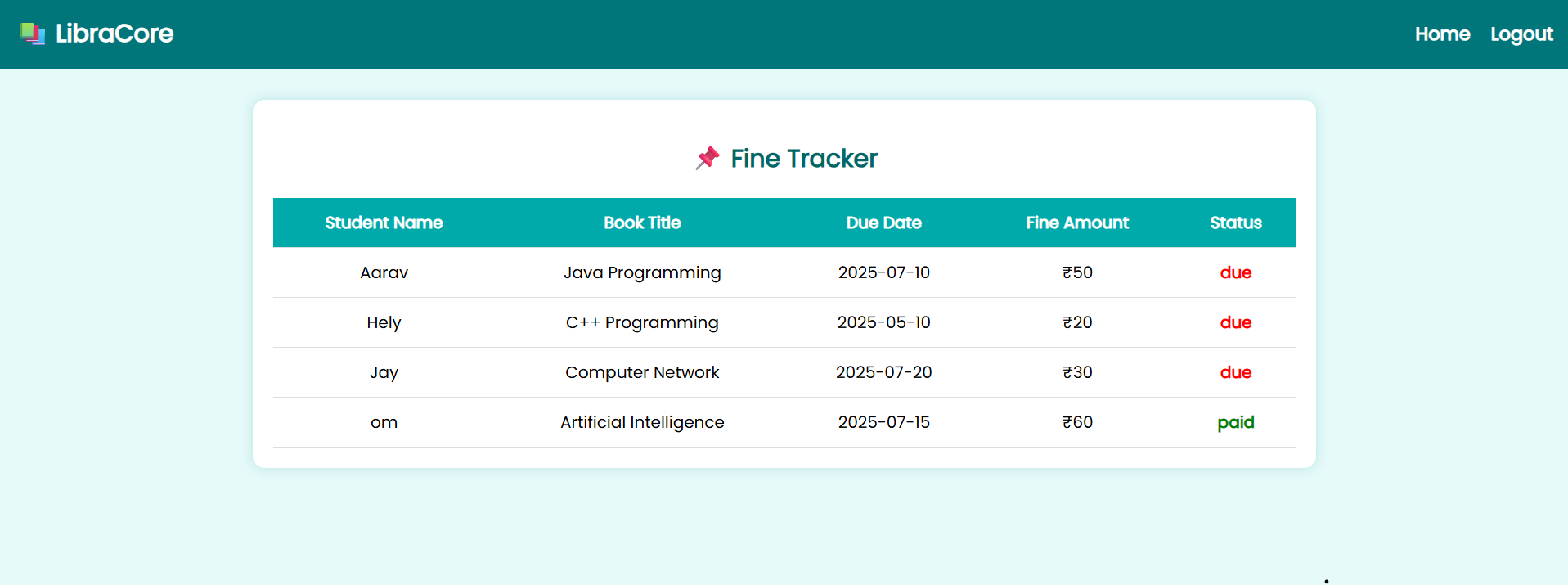
Issue Book Form



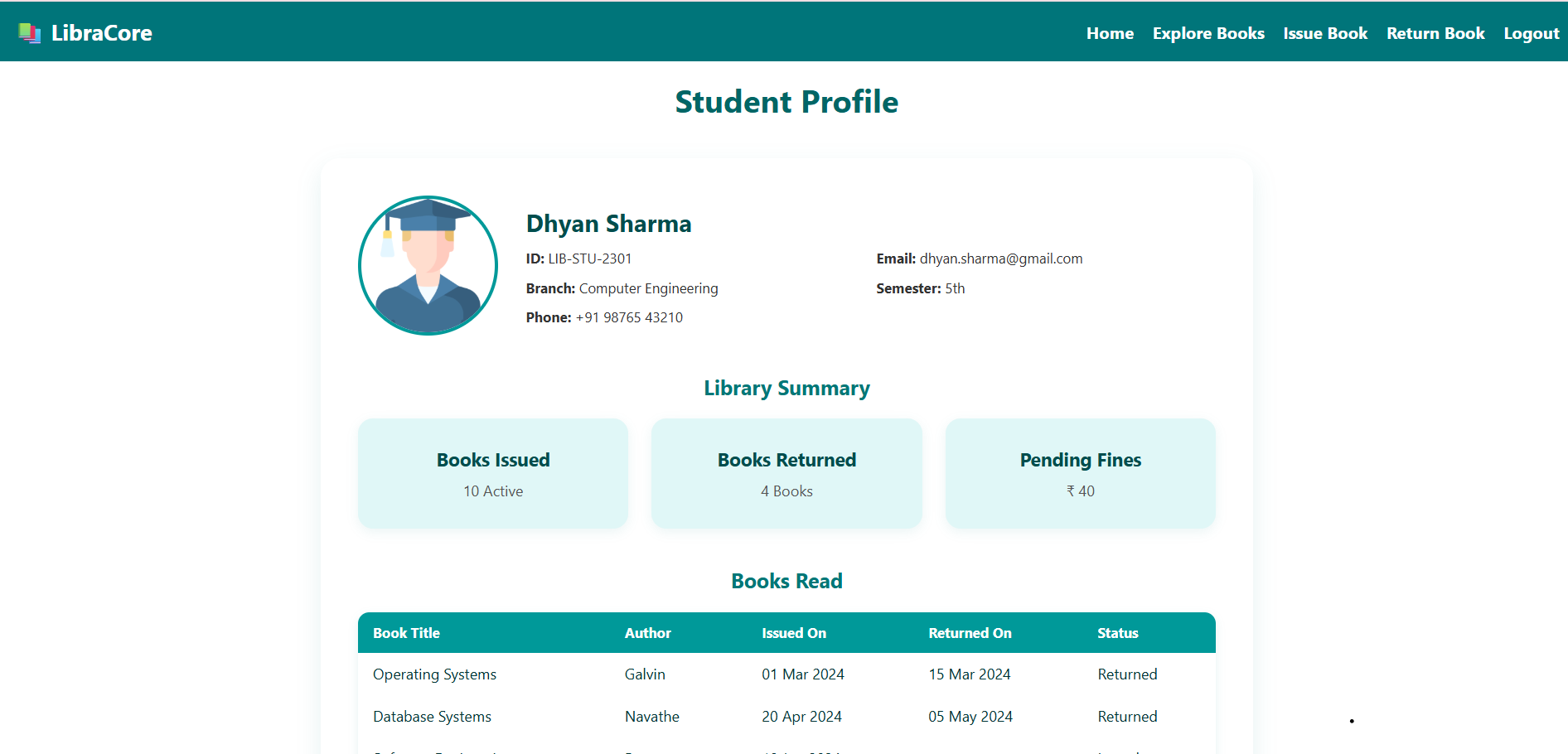
Book Issued Successfully



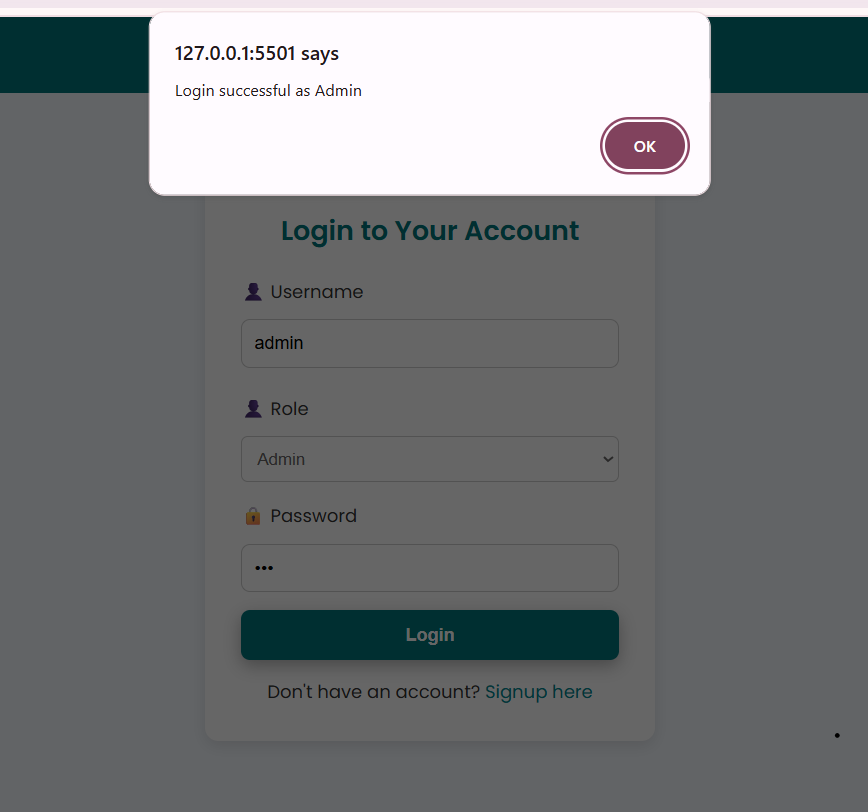
Return Book



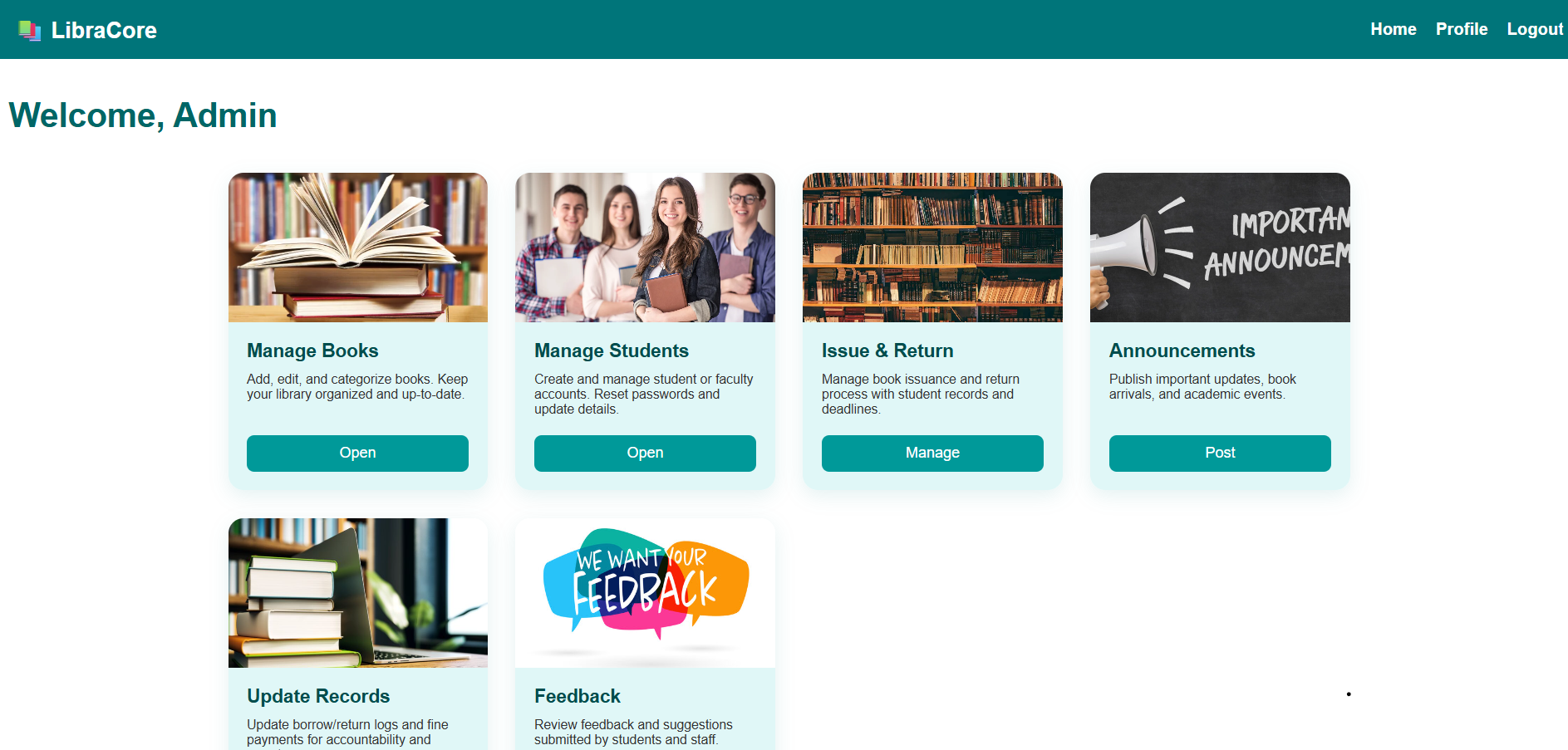
Fine Tracker

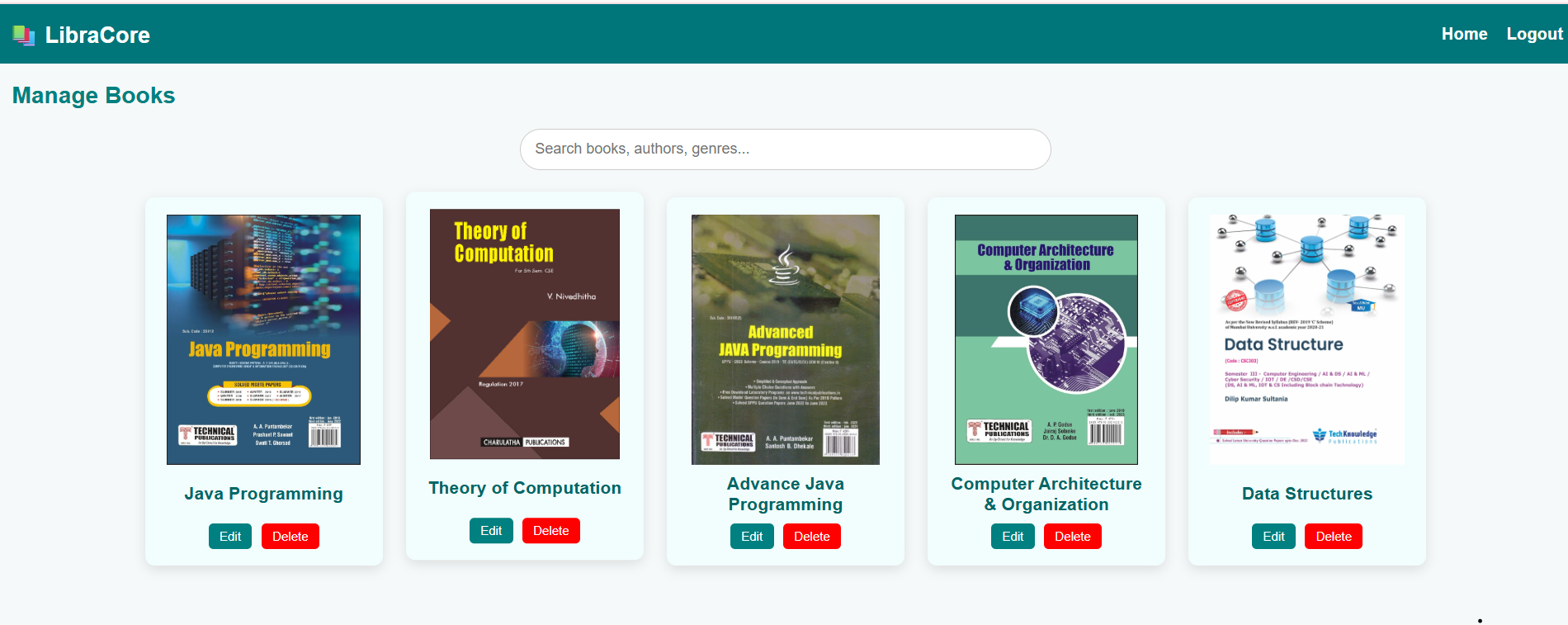


Student Profile

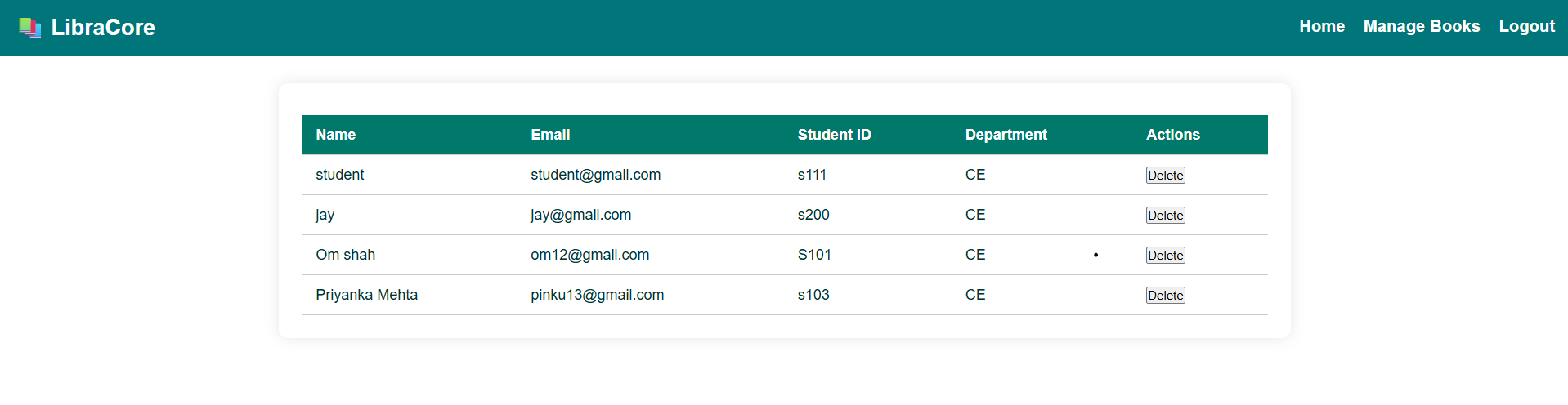


Admin Login Page

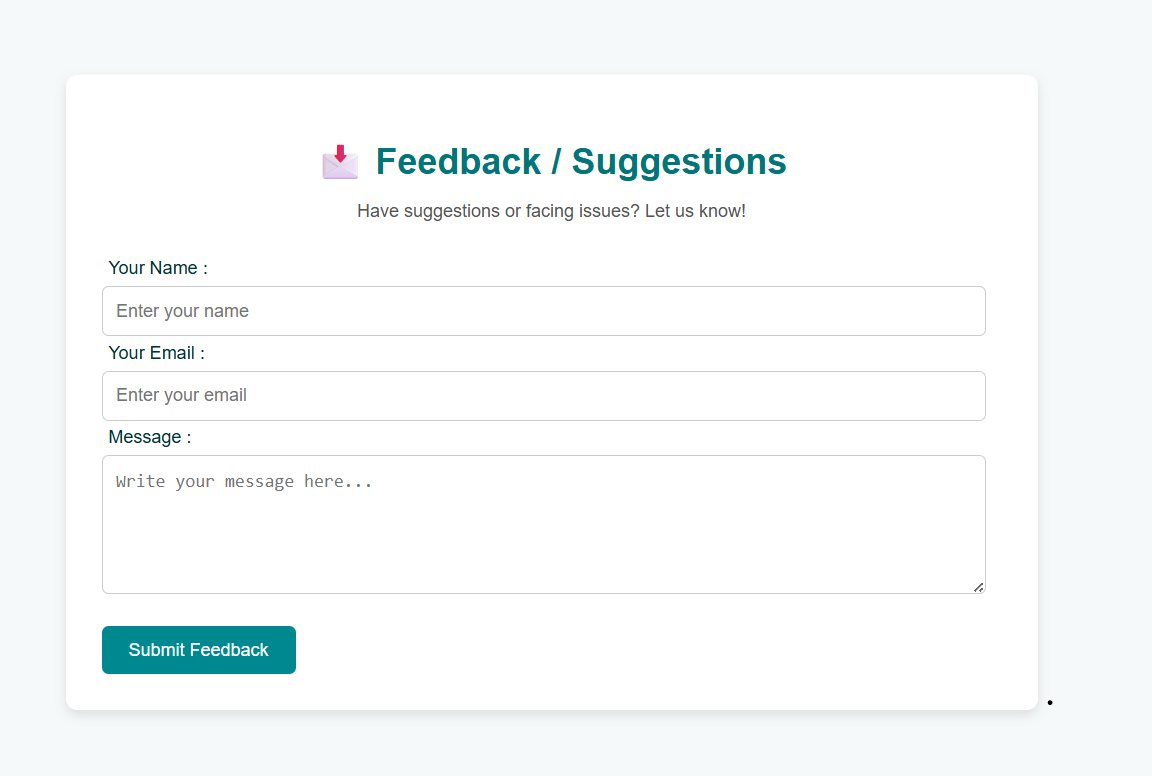




Manage Books



Manage Students



**13.Annexures**

https://github.com/herlin125/LibraCore---Project-.git

**14.Refrences**

<https://nodejs.org/en/docs/>

<https://expressjs.com/>

<https://www.mongodb.com/docs/>

<https://code.visualstudio.com/>

<https://chat.openai.com/>