Low-Level Design

1. Introduction:

The Library Management System is a web-based application that simplifies the management of books and students in a library. It provides functionalities such as student registration, book search, and book details viewing. The system is primarily administered by an admin who has privileges to add, delete, and update books, issue books to students, collect fines, and manage student records.

2. System Components:

The Library Management System is composed of the following components:

- Frontend: Utilizes JSP (JavaServer Pages) for creating the user interface and enabling user interaction.

- Backend: Implements the business logic and database operations using the Spring Boot framework.

- Persistence: Utilizes Spring Data JPA for accessing and manipulating data stored in the MySQL database.

3. User Roles:

The system supports two user roles:

- Student: Can register, login, search for books, view book details, and pay fines.

- Admin: Has complete control over the system, including the ability to add, delete, and update books, issue books to students, manage fines, and maintain student records.

4. Student Management:

4.1 Registration and Login:

- Students can register by providing necessary details such as their name, email, and password.

- Upon successful registration, a new student record is created in the database.

- Students can log in using their registered email and password.

4.2 Book Search and Viewing:

- Students can search for books based on keywords that match the book's category, title, or author.

- The system displays a list of books that match the search criteria, showing details such as the book ID, title, author, and category.

- Students can view the complete details of a specific book, including its availability and any fines associated with it.

5. Book Management:

5.1 Admin Authentication:

- Admin credentials are hardcoded into the application for authentication.

- Only the admin can access the book management functionality.

5.2 Book CRUD Operations:

- The admin can add a new book by providing details such as the title, author, category, and availability status.

- The admin can delete an existing book from the system.

- The admin can update the details of a book, including the title, author, category, and availability status.

5.3 Student Management:

- The admin has access to the list of registered students in the system.

- The admin can delete a student record if necessary, provided that the student has no pending fines.

5.4 Book Issuing:

- The admin can issue books to students, with a limit of three books per student.

- The issuing date is stored in the database for each book issued.

- If a student fails to return a book within 15 days, a fine is calculated based on the number of days overdue.

- The fine amount is 10/9 rupees per day for each overdue book.

5.5 Fine Collection:

- The admin can collect fines from students who have paid the outstanding amount.

- By clicking a button next to the student's fine details, the admin can update the student's fine amount to zero.

6. Technology Stack:

The Library Management System utilizes the following technology stack:

- Frontend: JSP (JavaServer Pages)

- Backend: Spring Boot

- Persistence: Spring Data JPA

- Database: MySQL

7. Conclusion:

The Low-Level Design of the Library Management System encompasses components for student management, book management, and fine collection. The system supports user registration, login, book search, book viewing, and CRUD operations for books and student records. Admin credentials are used for system administration, including book issuing, fine management, and student record maintenance. The technology stack consists of JSP for the frontend, Spring Boot for the backend, and Spring Data JPA for data persistence using MySQL.

High-Level Design Document - Library Management System

1. Introduction:

The Library Management System is a web-based application designed to streamline and automate various operations in a library. Its primary goal is to improve the efficiency and effectiveness of tasks such as book cataloging, user management, borrowing and returning books, reservations, and reporting. By automating these processes, the system helps librarians manage the library's resources more effectively and provides users with a seamless experience.

2. System Architecture:

The Library Management System follows a modular and layered architecture approach to ensure scalability, maintainability, and separation of concerns. The system consists of the following layers:

- Presentation Layer: This layer focuses on the user interface and handles user interactions. It provides a web-based interface through which users can access the system and perform various actions. The presentation layer includes components such as web pages, forms, buttons, and menus that allow users to interact with the system. It ensures a user-friendly and intuitive experience.

- Application Layer: Also known as the business logic layer, this layer contains the core logic and business rules of the system. It is responsible for processing and manipulating data, enforcing business rules, and coordinating interactions between different modules or components. The application layer ensures that data is validated, transformed, and stored correctly. It encapsulates the system's functionalities and provides a set of services that can be accessed by the presentation layer.

- Data Layer: This layer manages data storage and retrieval operations. It includes a database or data repository that stores information about books, users, transactions, and other relevant data. The data layer ensures data persistence, integrity, and efficient access. It provides mechanisms to retrieve, store, update, and delete data, and it abstracts the underlying database operations to simplify data management.

3. Component Overview:

The Library Management System consists of several modules or components that work together to provide the necessary functionalities:

- User Management Module: This module handles user accounts, authentication, and authorization. It allows librarians to create and manage user accounts, authenticate users, and control access to different system features. It stores user information, including personal details and borrowing history, ensuring the security and privacy of user data.

- Catalog Management Module: This module is responsible for managing the library's collection of books, magazines, and other resources. It provides functionalities to add new items to the catalog, update existing items, and remove items that are no longer available. It maintains details such as title, author, ISBN, availability status, and location. The module ensures that the catalog is up to date and accurate.

- Borrowing and Return Module: This module handles the borrowing and returning of library materials. It supports functions such as issuing books to users, setting due dates, and tracking returns. It manages the borrowing history and keeps track of the availability of items. The module also sends notifications for overdue items and manages fines for late returns, ensuring that the borrowing process is efficient and fair.

- Reservation Module: This module enables users to reserve books that are currently unavailable. It manages a queue of reservations and automatically notifies users when the reserved item becomes available. The module ensures that users have access to the resources they need, even if they are currently checked out by other users.

- Search Module: This module provides users with an interface to search for books and other resources in the library. It implements search filters based on title, author, category, or keyword, allowing users to find specific items of interest. The module retrieves and displays search results, providing detailed information about each item, including availability status and location.

- Reporting Module: This module generates reports for library administrators, providing insights into various aspects of library operations. It includes functionalities to generate borrowing statistics, inventory status, user activity reports, and other relevant information. The module helps librarians make informed decisions, optimize resource allocation, and improve overall library management.

4. Conclusion:

The Library Management System's high-level design incorporates a modular architecture with distinct components for user management, catalog management, borrowing and returning, reservations, search functionality, and reporting. The system follows a layered approach, ensuring separation of concerns and promoting scalability and maintainability. By automating various library operations, the system aims to enhance efficiency, improve user experience, and empower librarians with valuable insights for effective resource

management.

Architecture

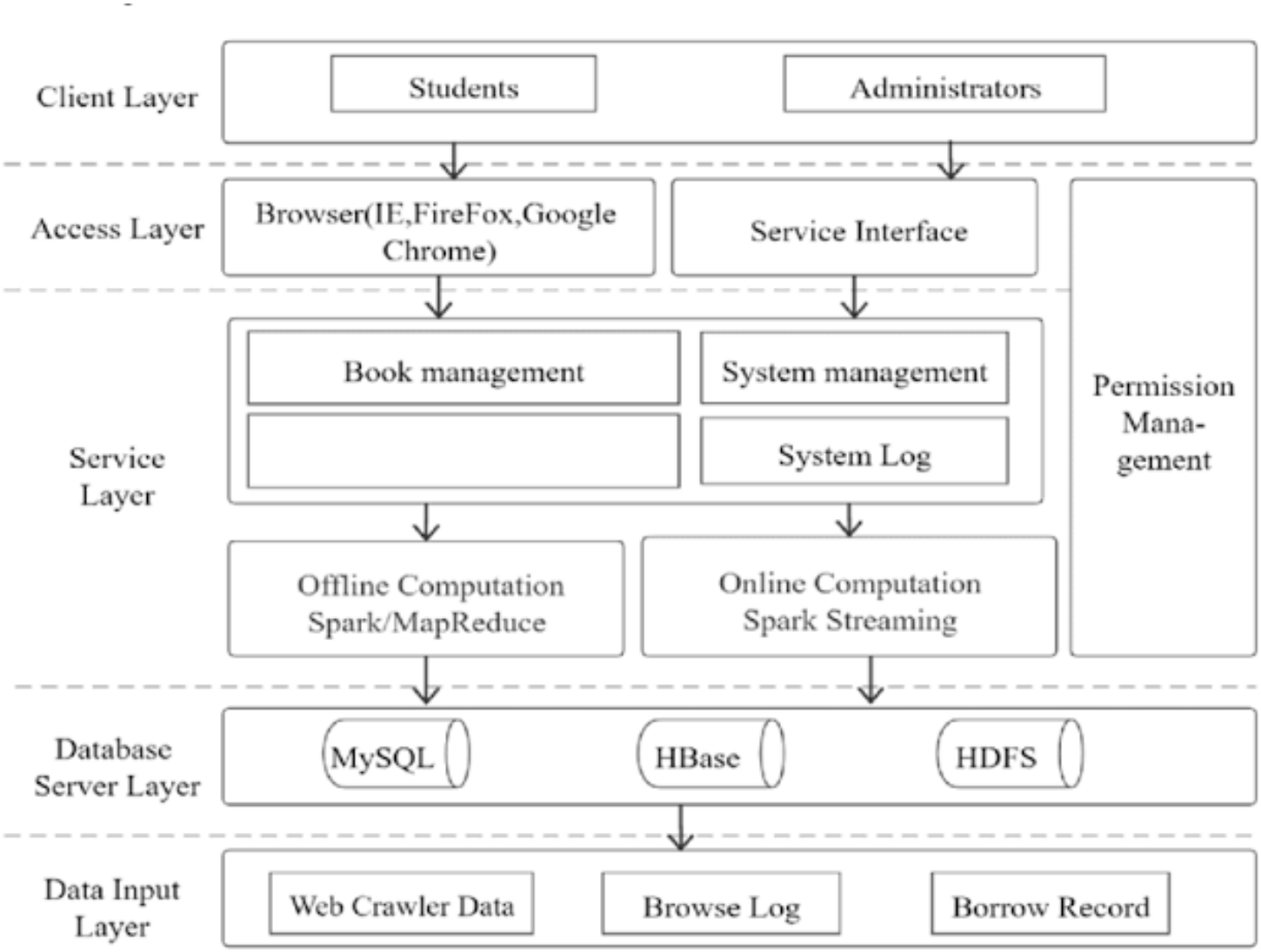

Fig 1. Architecture of Library Management System

Fig 2. Functional Structure of Library Management System

WIREFRAME

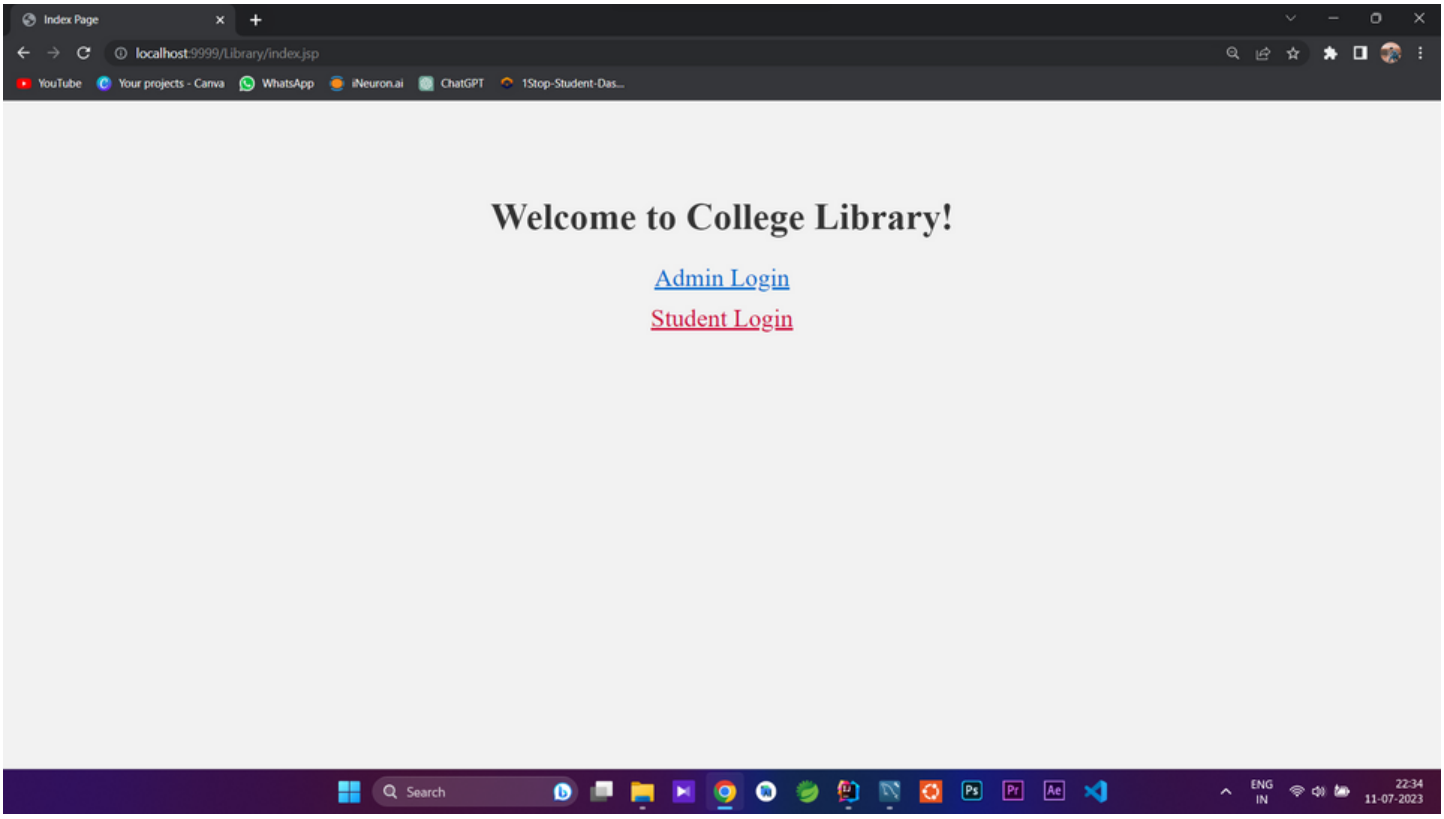

Fig 1. Index page where we can login as admin or student

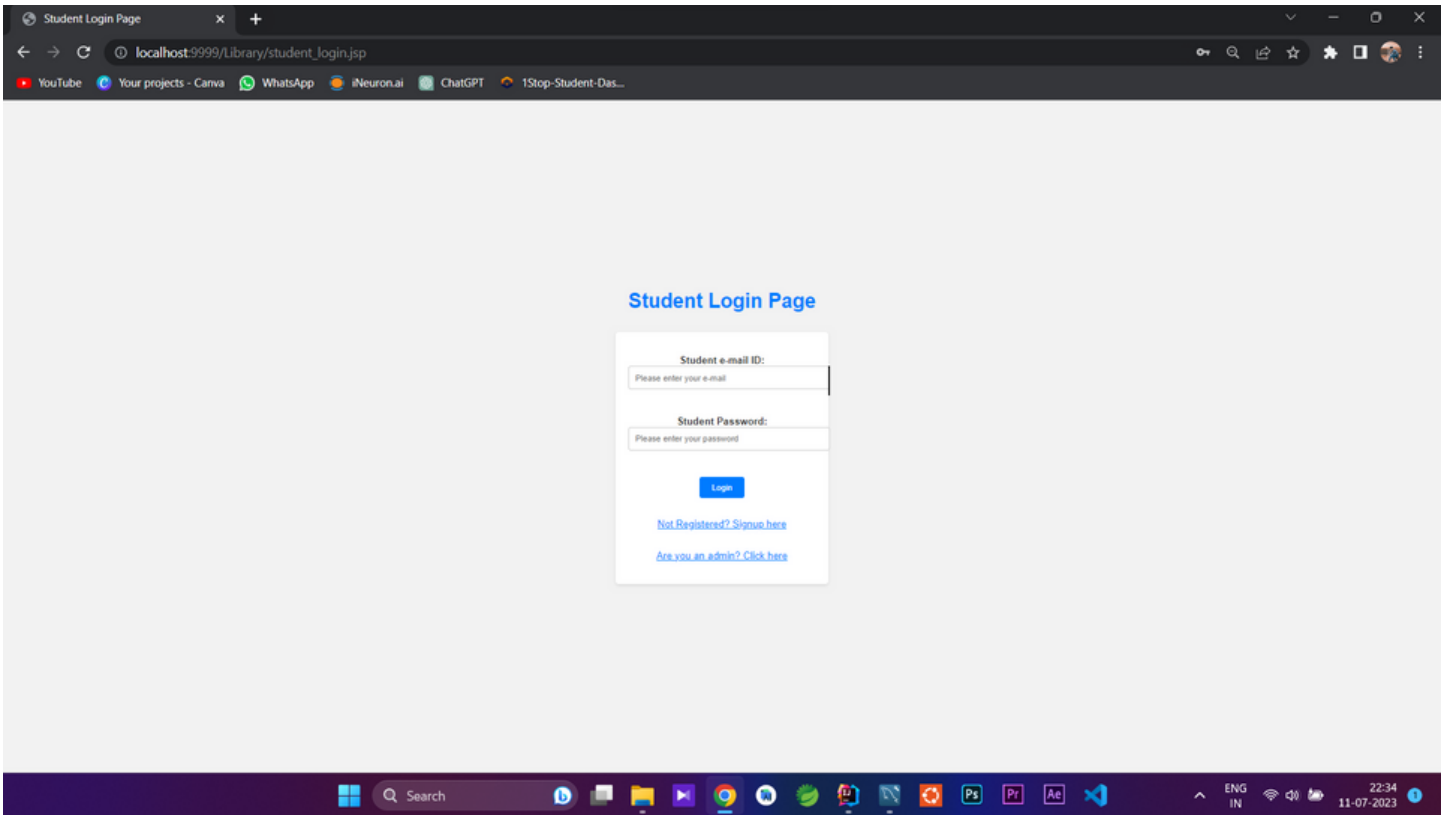

Fig 2. Student Login Page

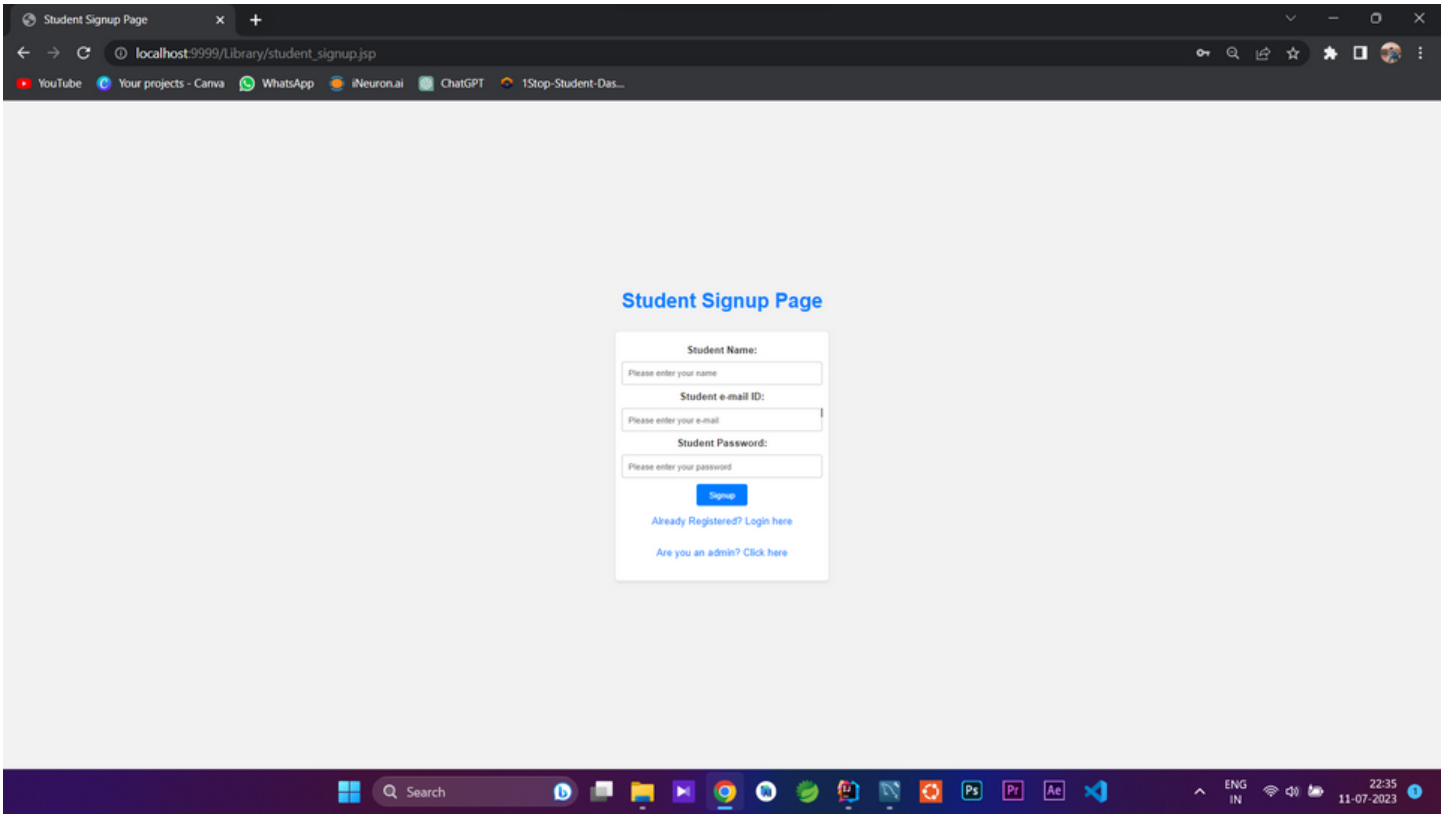

Fig 3. Student Signup Page

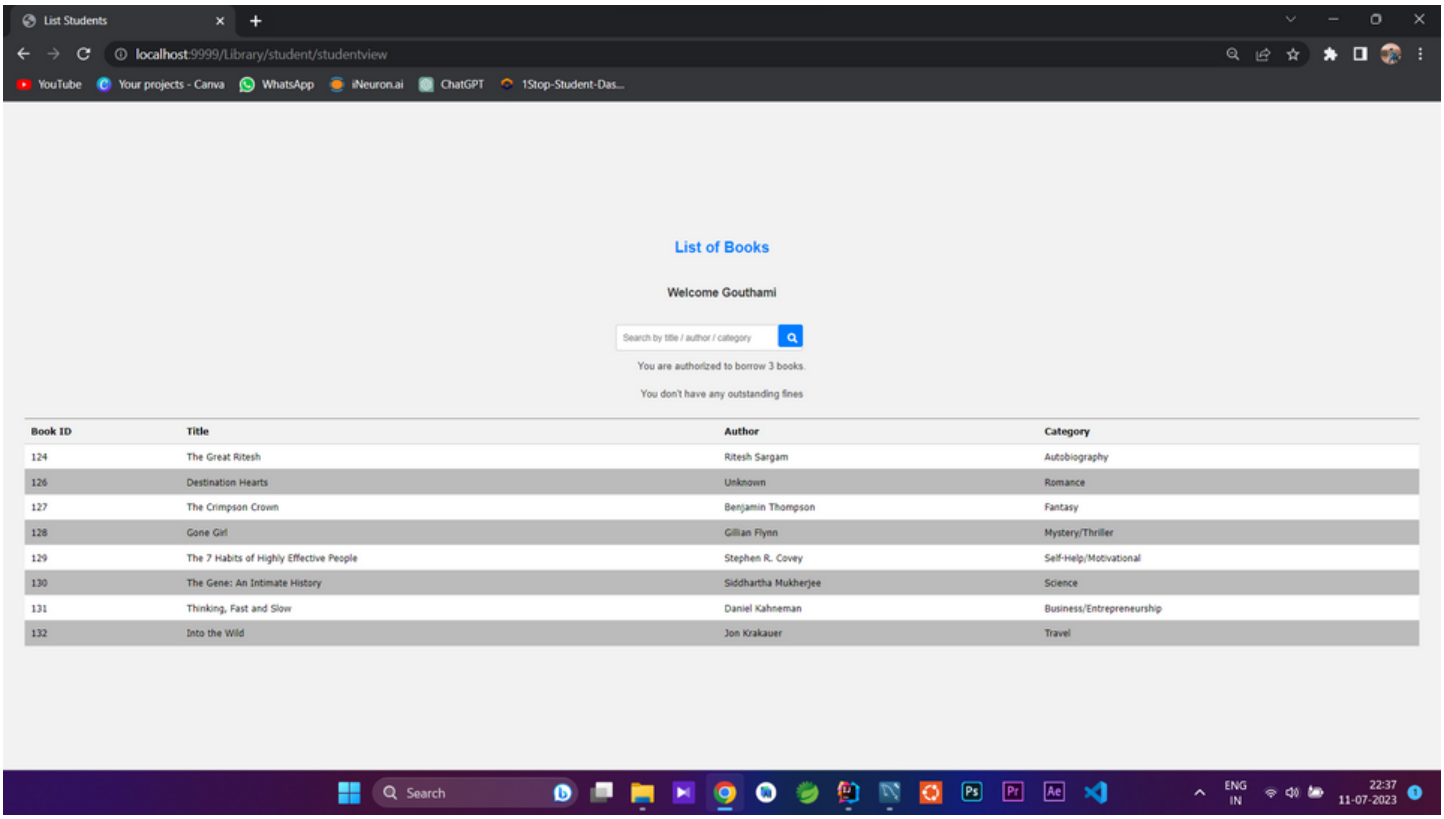

Fig 4. After Student Login he can view available books

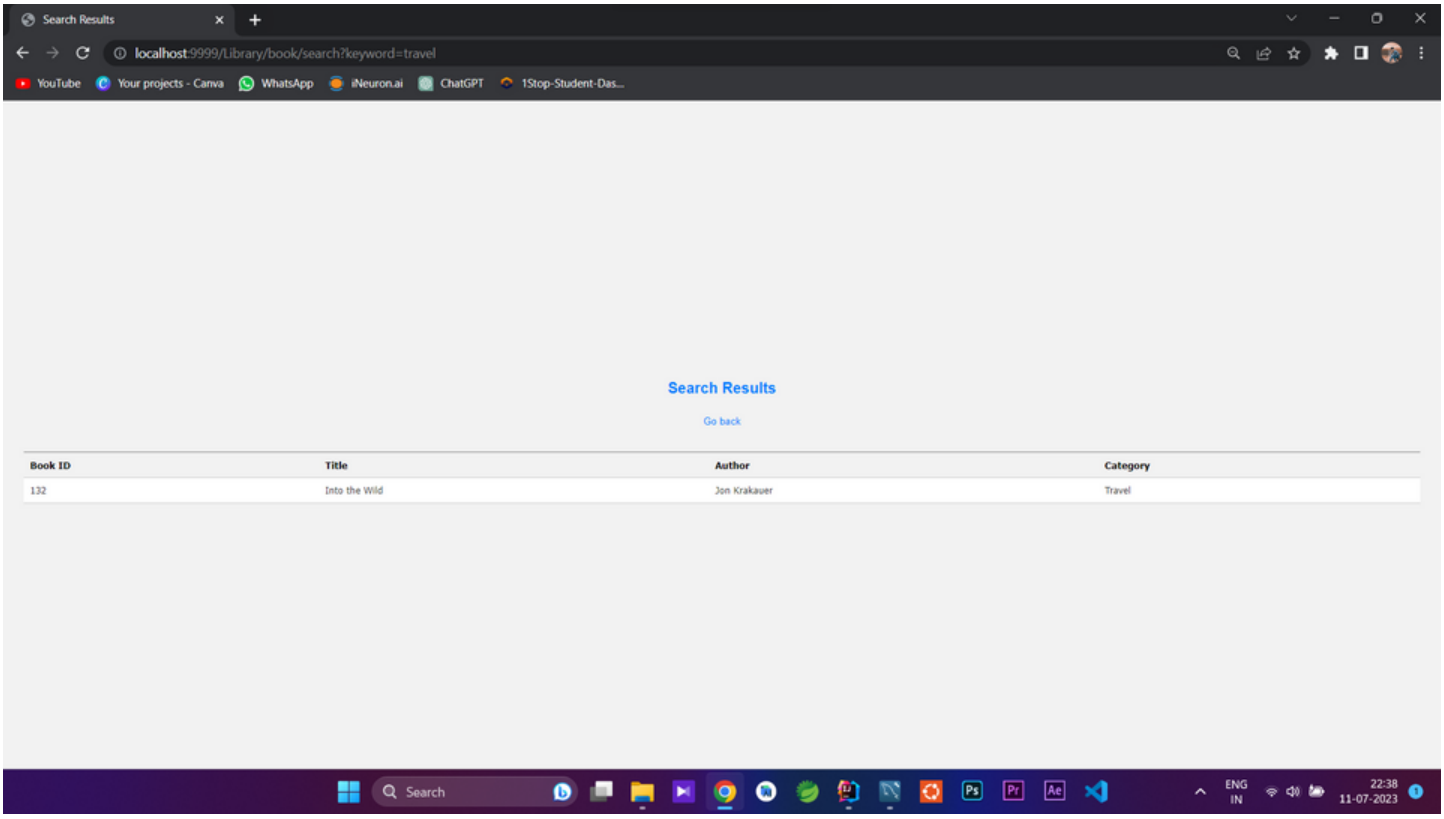

Fig 5. We can search book by Title, Author, Category

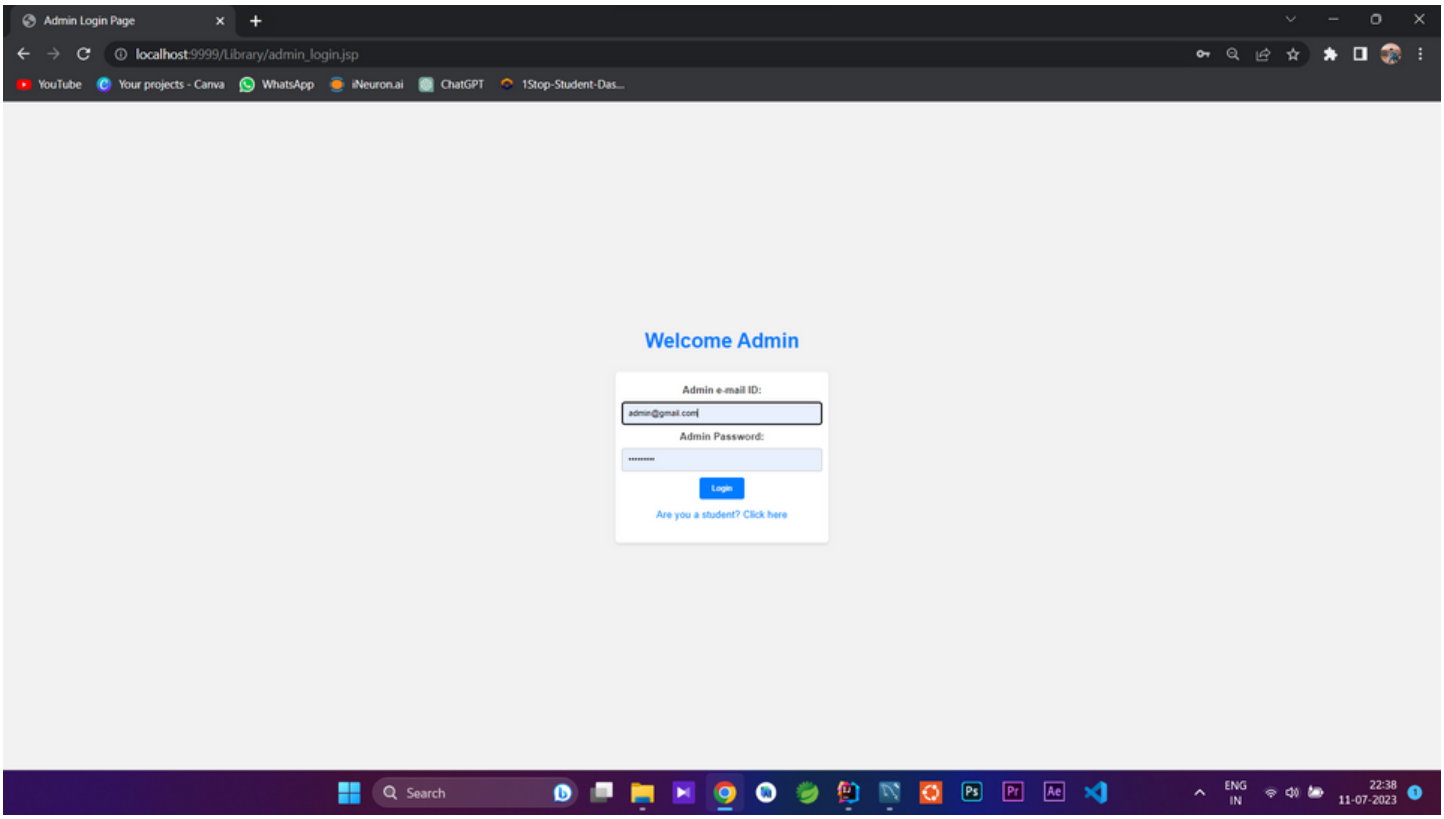

Fig 6. Admin Login Page

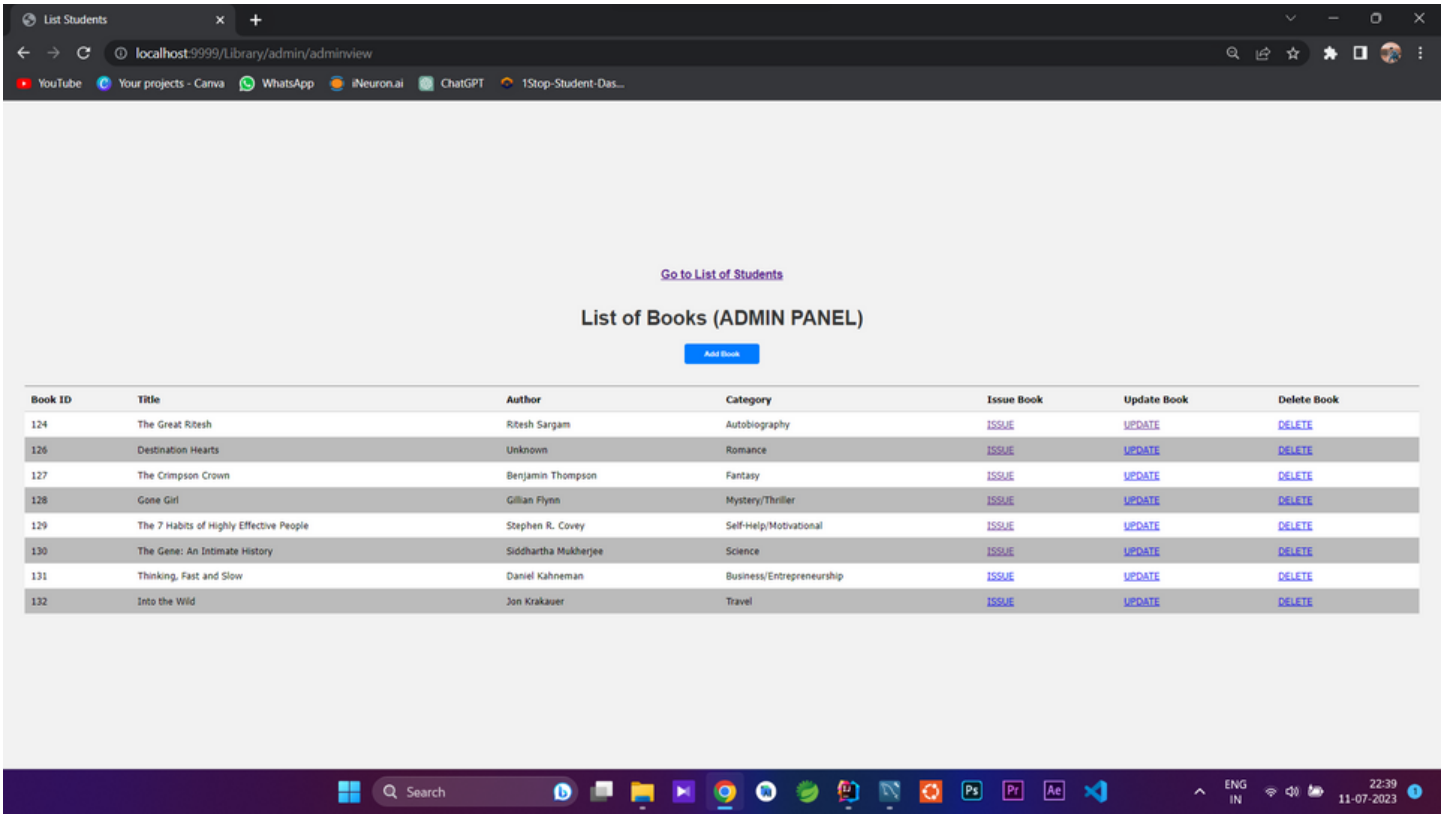

Fig 7. After Admin login successfully

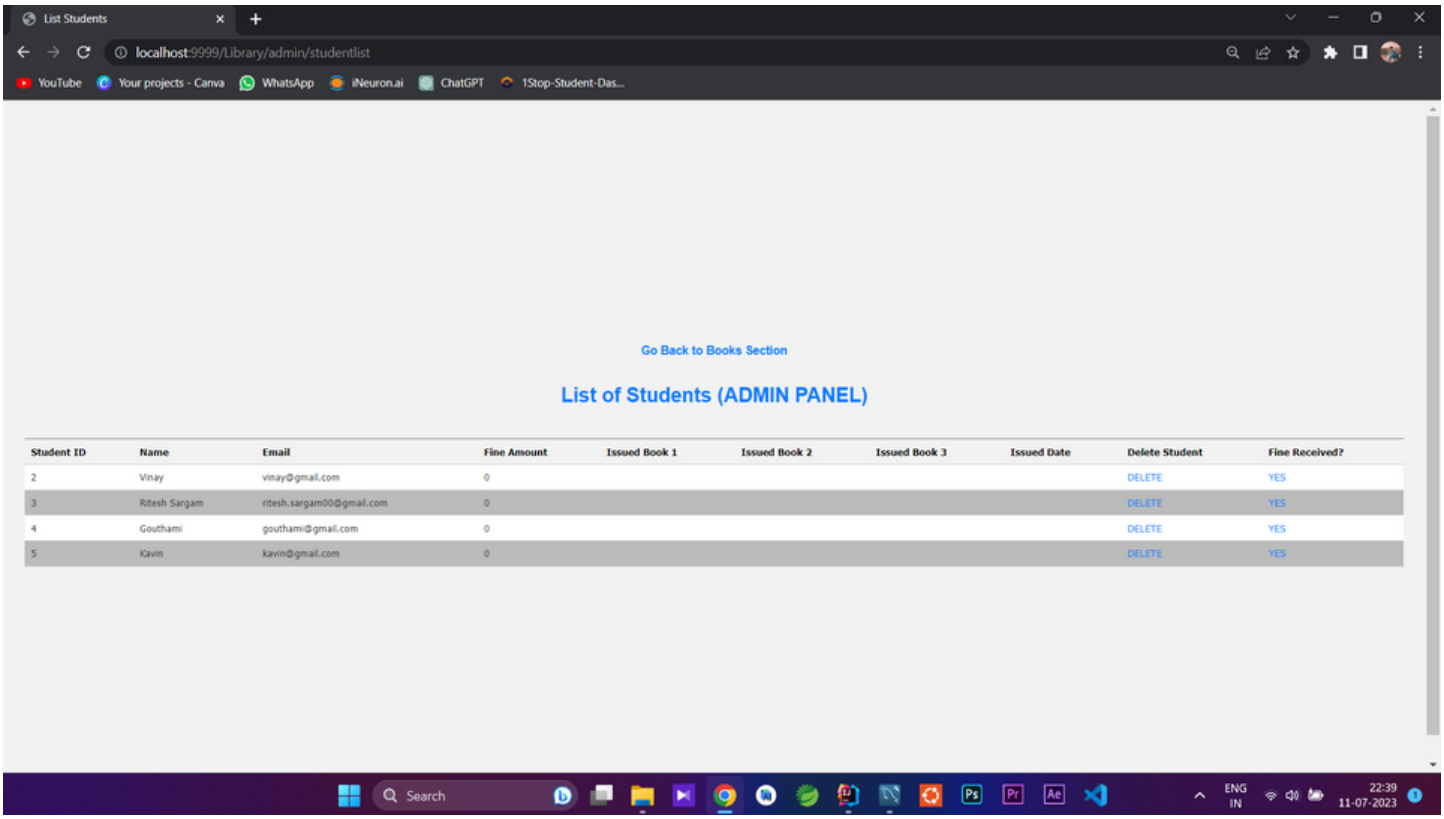

Fig 8. Admin can view list Student

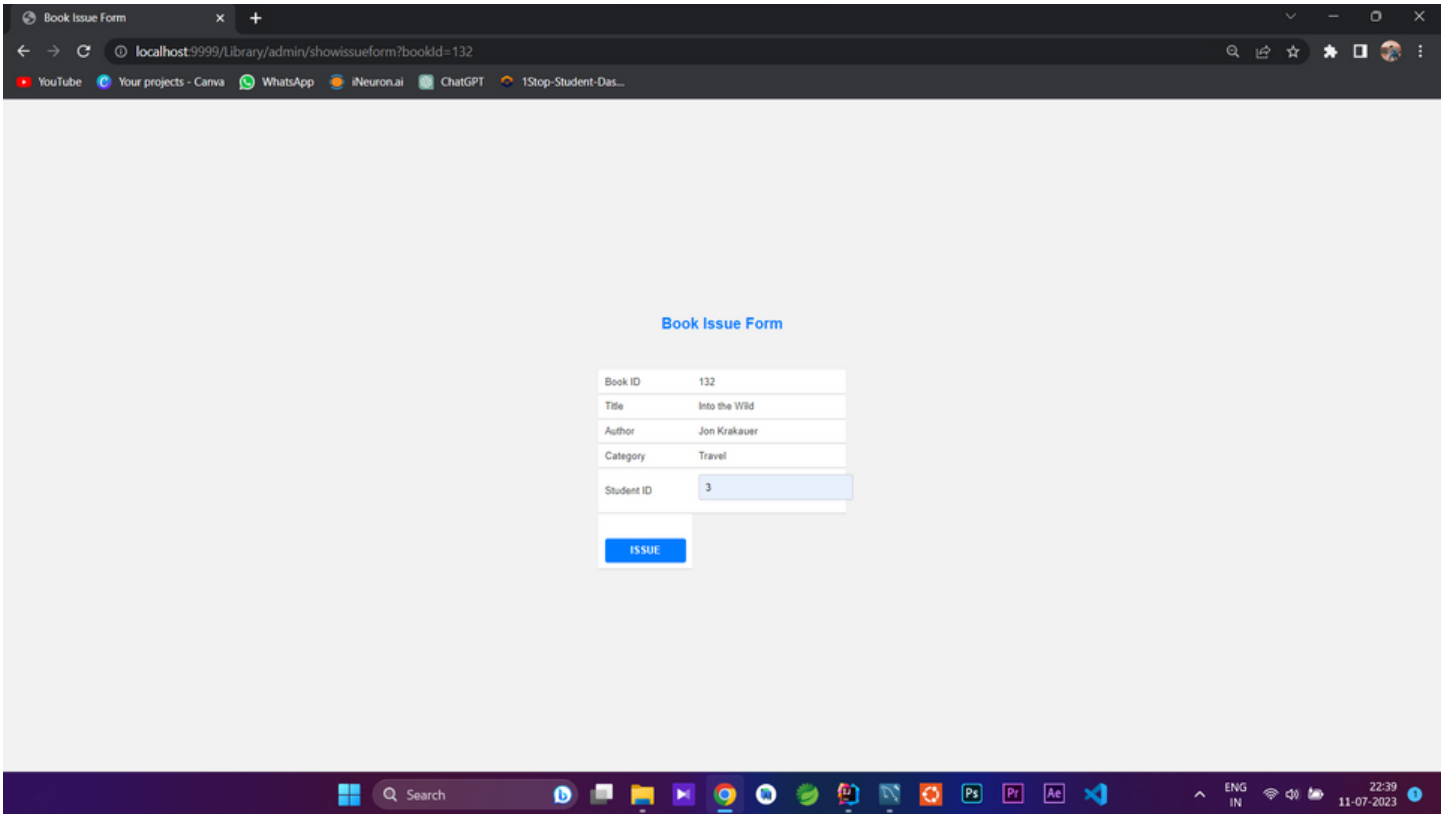

Fig 9. Admin can Issue a book to a student with his ID

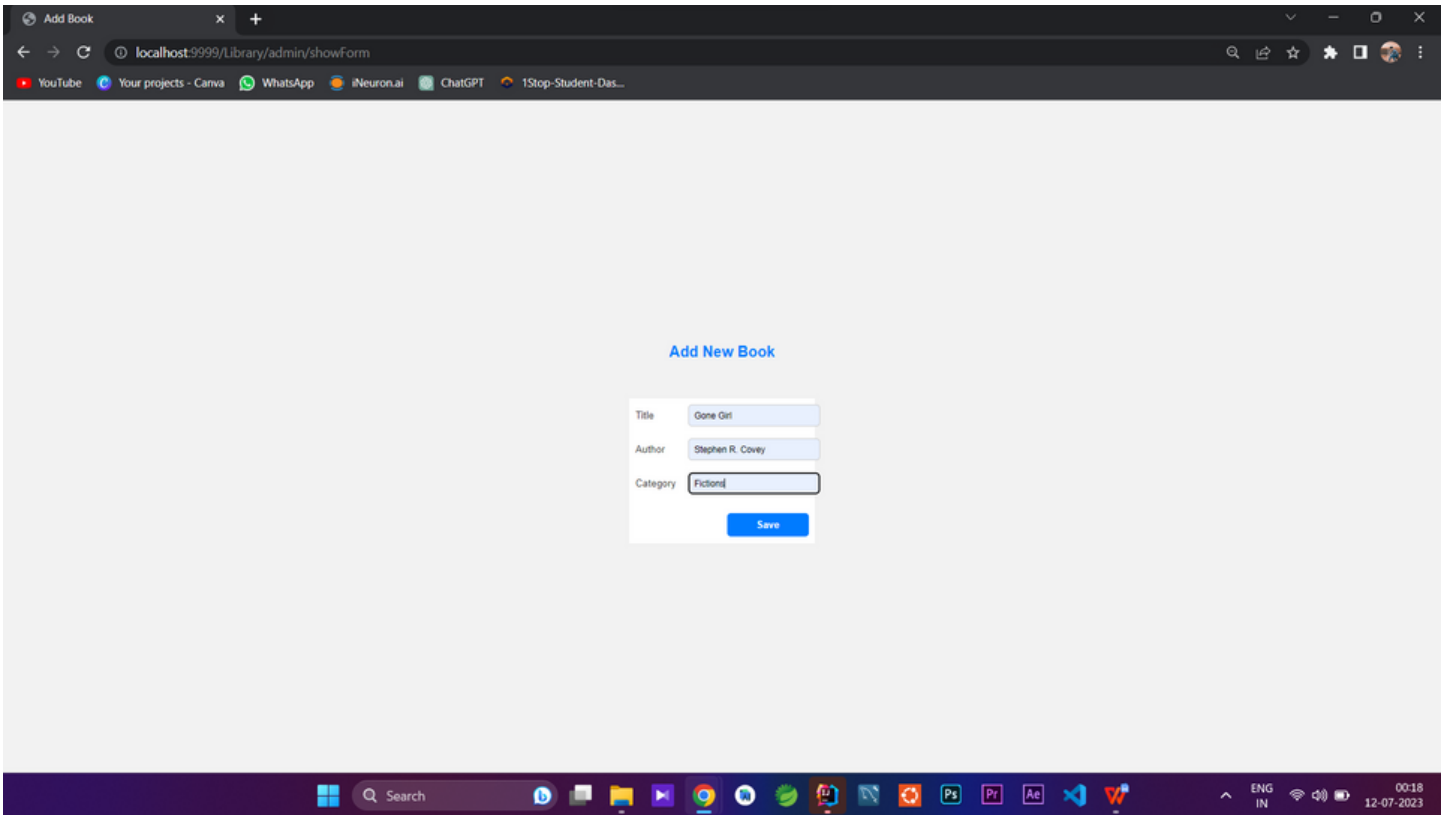

Fig 10. Admin can add a new book

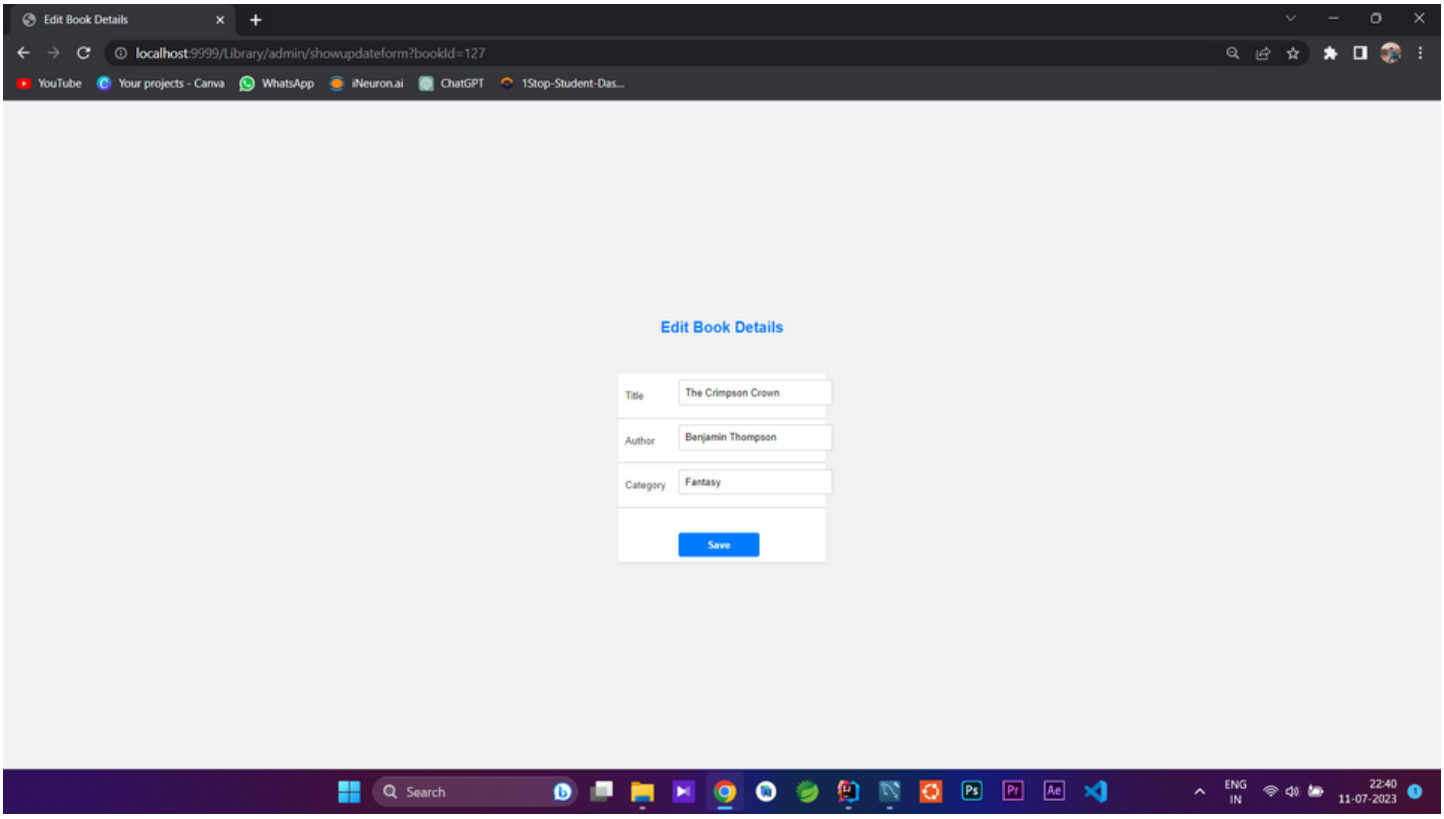

Fig 11. Admin can edit a book details

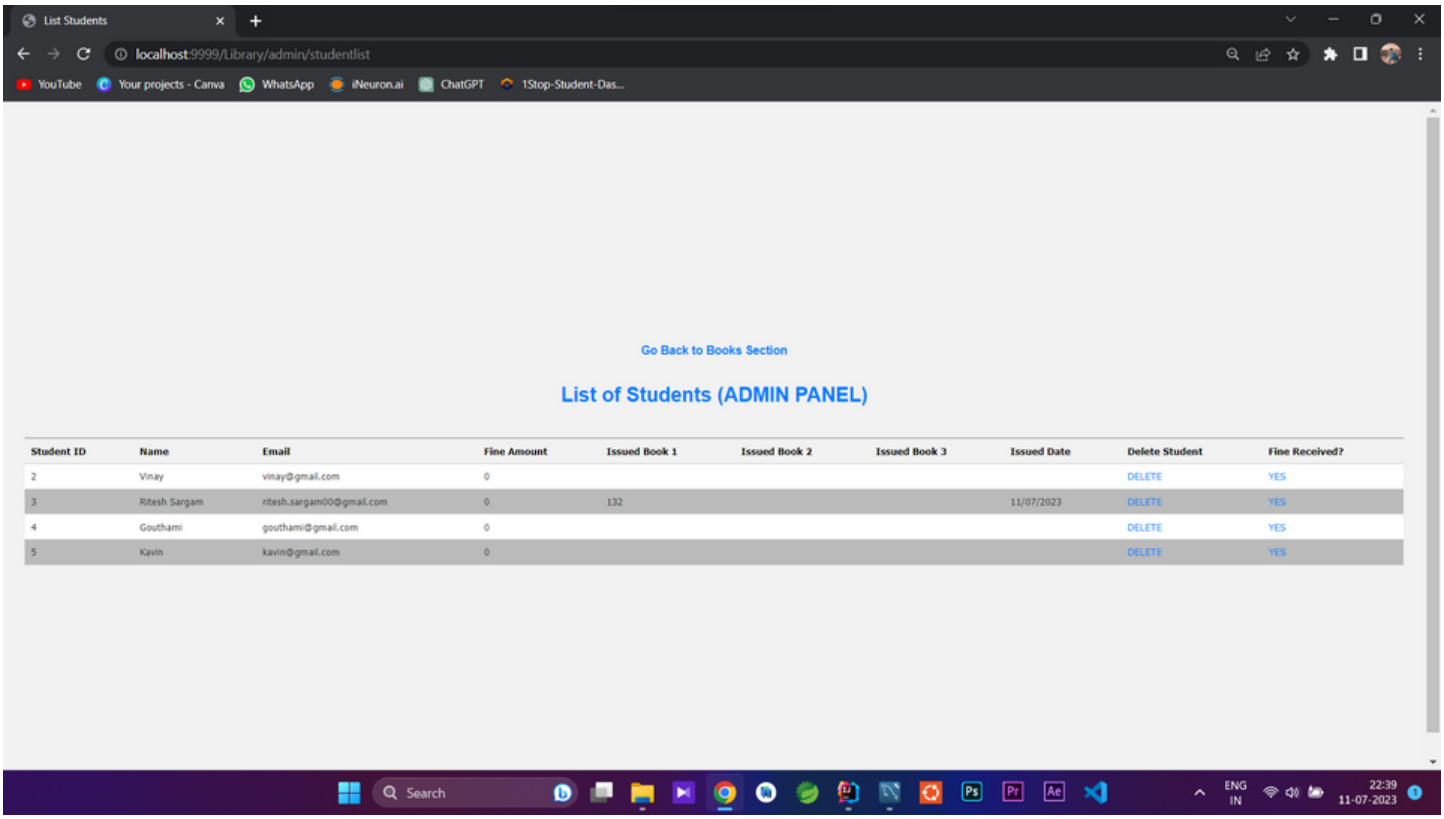

Fig 12. Admin can view which book is issued to students and collect fine if due date is over

Fig 13. Source Code Files

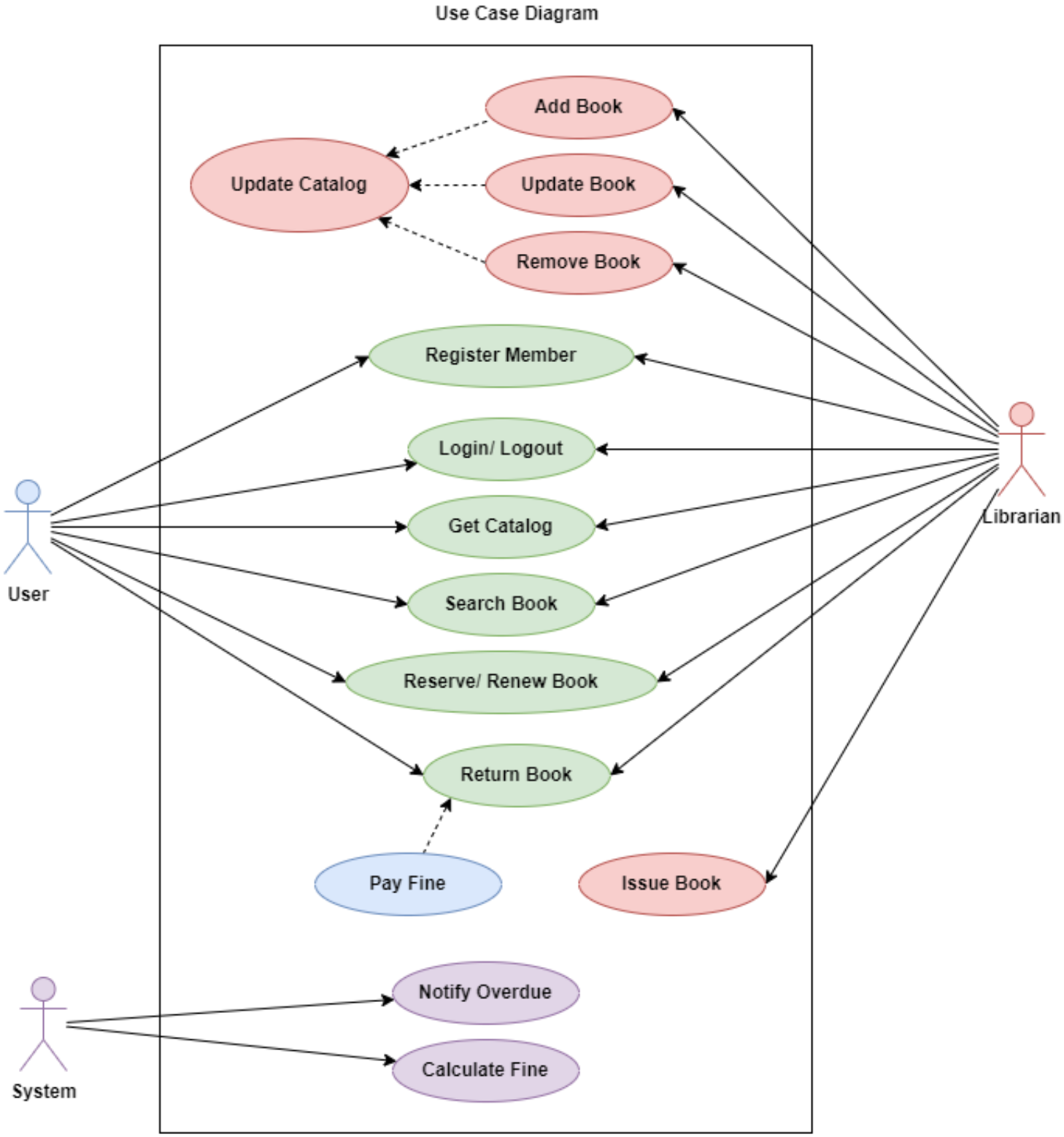

Fig 14. Functional Structure of Library Management System