Project Report: Book Management System

BY

PRABHANSHU SINGH

Prabhanshusingh94@gmail.com

Introduction

The Book Management System is designed to manage a collection of books efficiently. The system allows users to view, insert, and manage book records. This report outlines the technologies used to develop the project, the features implemented, and an overview of the architecture.

Technologies Used

1. Backend

- **Java**: The backend of the application is developed using Java, a versatile programming language known for its portability and robustness. It is used to implement the business logic and handle requests from the frontend.
- **Spring Boot**: Spring Boot is utilized to create the RESTful API for the application. It simplifies the setup and development of new applications, allowing for rapid development. The framework provides built-in features for security, data access, and dependency injection.
- **Hibernate**: Hibernate is used as the Object-Relational Mapping (ORM) tool to manage the interaction between the application and the database. It simplifies database operations by allowing developers to work with Java objects instead of SQL queries.
- **MySQL**: MySQL is the chosen relational database management system for storing book data. It provides robust data storage and management capabilities, enabling efficient data retrieval and manipulation.

2. Frontend

- **HTML**: HTML is used to structure the user interface of the application. It provides the necessary elements for displaying book records and forms for data entry.
- **CSS**: CSS is employed for styling the application, ensuring a visually appealing layout and enhancing user experience. Custom styles are applied to make the interface user-friendly.
- **Bootstrap**: Bootstrap is used as a front-end framework to design responsive web pages. It includes pre-defined styles and components, which facilitate rapid development and ensure compatibility across devices.
- **JavaScript**: JavaScript is utilized to implement dynamic functionalities on the frontend, such as fetching book data from the backend and handling form submissions. It enhances user interactivity within the application.

3. Development Tools

- **Eclipse IDE**: Eclipse Integrated Development Environment (IDE) is used for Java development, providing a robust environment for writing, debugging, and managing the codebase.

- **Postman**: Postman is employed for testing the RESTful API endpoints during development. It allows developers to send requests and analyze responses, ensuring that the API functions as expected.

Features Implemented

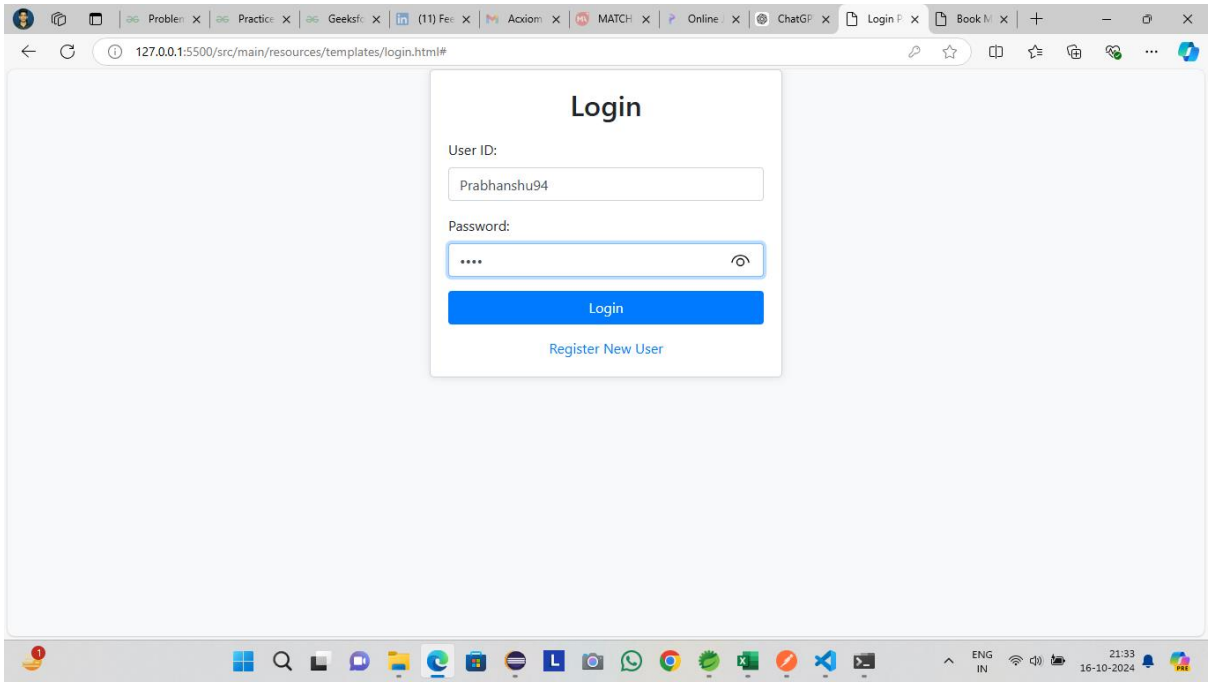
- **Book Listing**: Users can view a table listing all available books with details such as serial number, book name, author, and availability status.
- **Insert New Book**: A modal form allows users to input new book data, including serial, book name, author, and availability. Upon submission, the data is sent to the backend and stored in the database.
- **Responsive Design**: The application is designed to be responsive, ensuring it is accessible and user-friendly on various devices, including desktops, tablets, and smartphones.

Conclusion

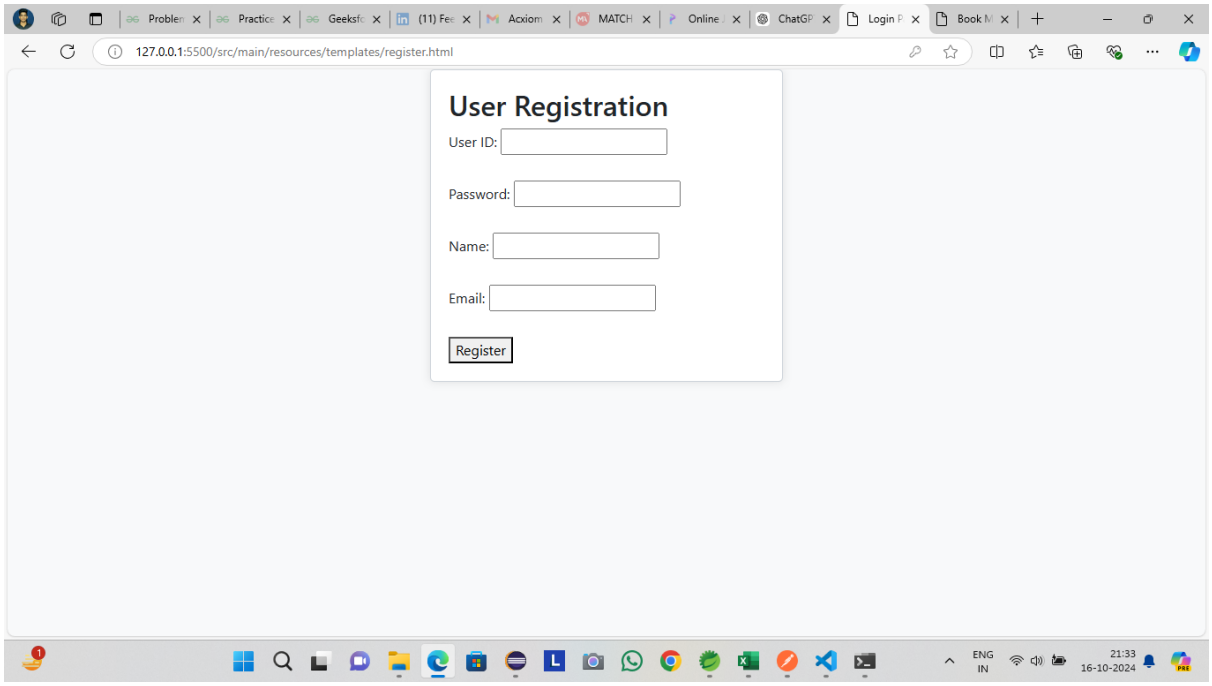
The Book Management System effectively demonstrates the integration of various technologies, providing a comprehensive solution for managing book records. The combination of Java, Spring Boot, Hibernate, MySQL, and frontend technologies such as HTML, CSS, Bootstrap, and JavaScript results in a robust application that is both functional and user-friendly. Future enhancements may include additional features such as user authentication, advanced search capabilities, and integration with external APIs for more extensive book databases.

Screenshots:

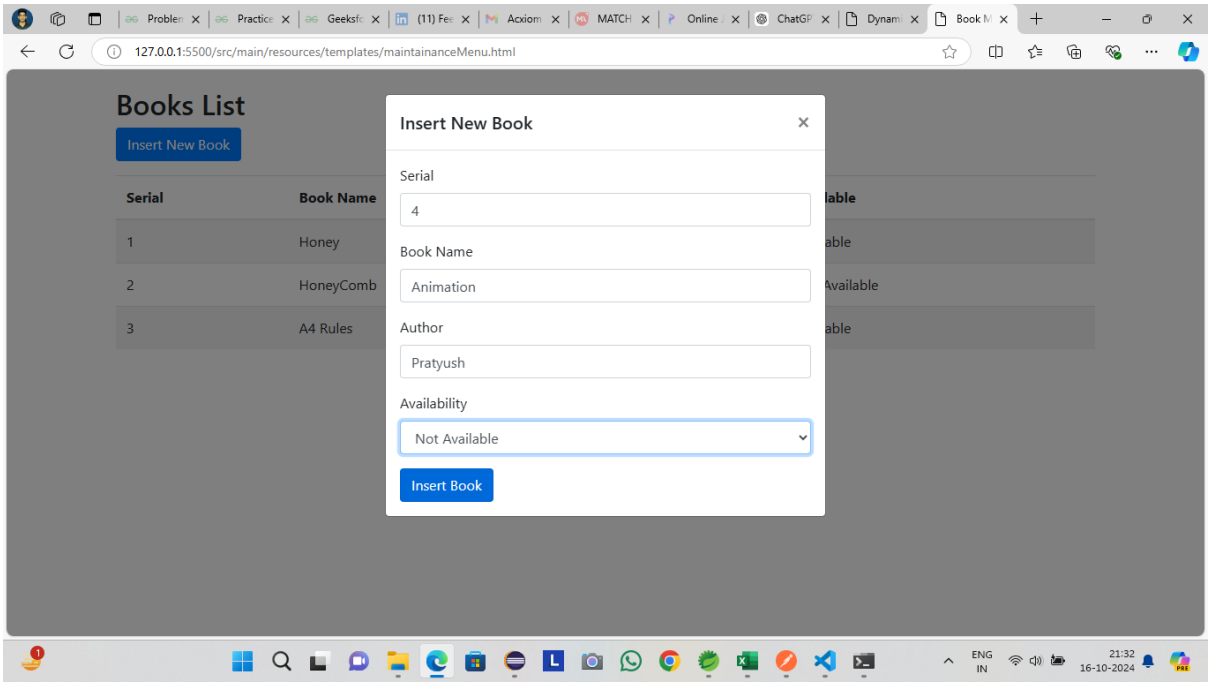
1. The website will start from the LOGIN page



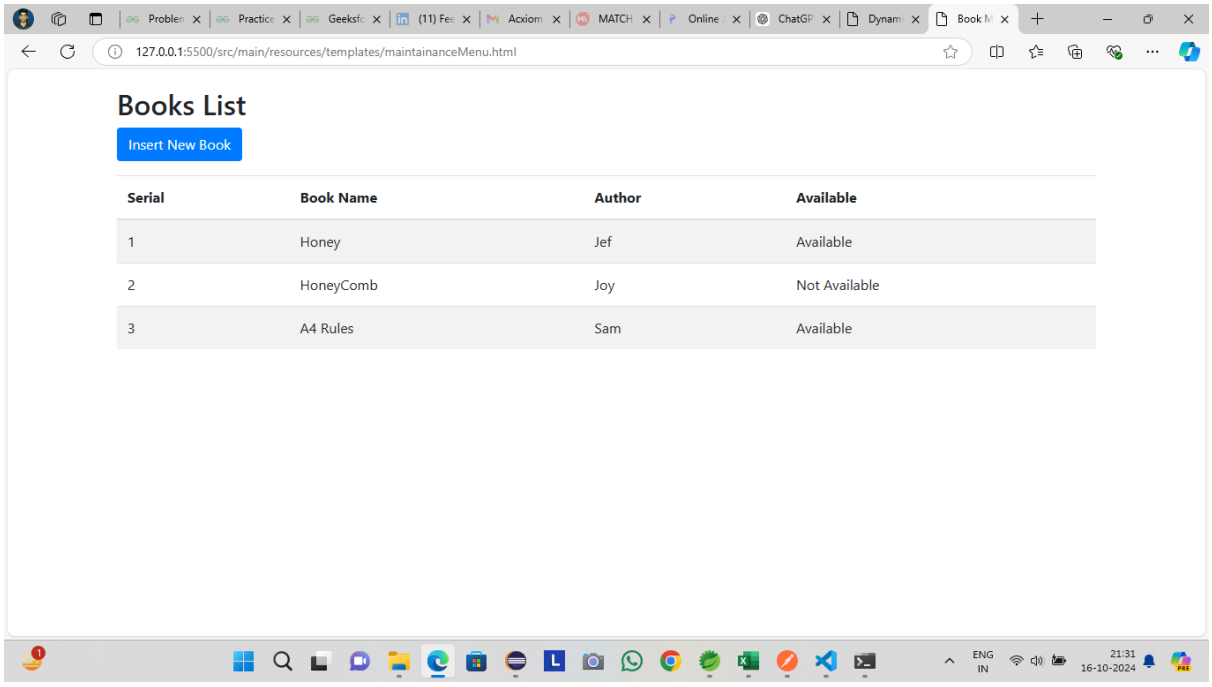
2. **If the user wants to register then it will redirect to the registration page.**



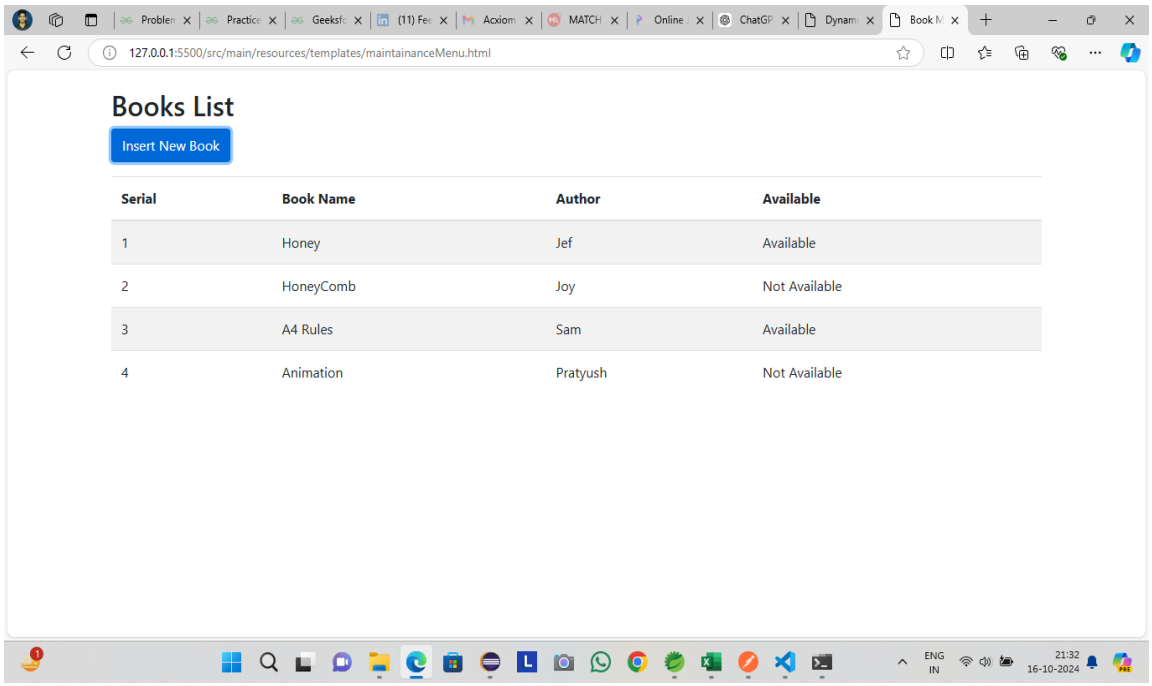
3. Book List View for admin user.



4. Insert New Books to the dataset (Admin Functionality)



5. **Live Update on the book list after updation**



6. Book List view for the user to rent a book.