Names: **Nkunda Christelle**

ID: **23474**

Group: **Monday, 18h (F)**

Lecturer: **Dushimimana Patrick**

Course: **WEB TECHNOLOGY Semester 1 (2023-2024)**

Final Exam Project

Topic: BOOK STORE MANAGAMENT SYSTEM

Notice: This documentation provides and explain all the 6 Requirements to submit the Project on GitHub.

I) Project Requirements:

Purpose: The purpose of the Book Management System is to provide users with a platform to manage their personal book collection. Users can sign up, log in, add new books, view available books, and manage their book list.

1. **Expected Outcomes:**

Users can sign up for an account and log in securely.

User Dashboard with options for adding a new book, viewing available books, and managing their book list.

Users can add, delete, and update books in their personal list.

Data validation to ensure the integrity of user input.

Secure storage of user data and book information.

2. **Constraints/Limitations:**

Use Java for the backend.

Spring MVC framework for application structure.

HTML, CSS, and JavaScript for the frontend.

GitHub for version control and collaboration.

II) Project Plan:

Scope: Develop a Book Management System focusing on user interactions.

1. **Timeline:**

Week 1: Set up the Spring MVC project, database, and user authentication

Week 2: Implement the user dashboard and functionalities (Add, delete, update books).

Week 3: Frontend Development

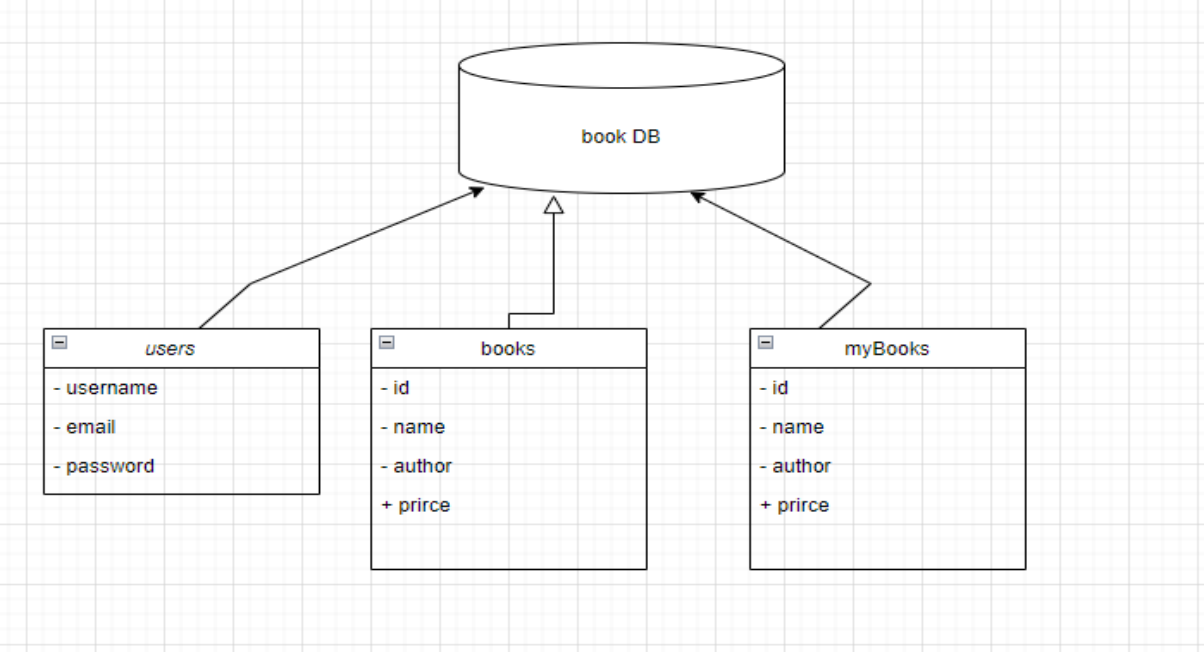
2. **Resources:**

- Java for backend development.
- Spring MVC for application structure.
- HTML, CSS, and JavaScript for frontend development.
- GitHub for version control and collaboration.

III) Source Code:

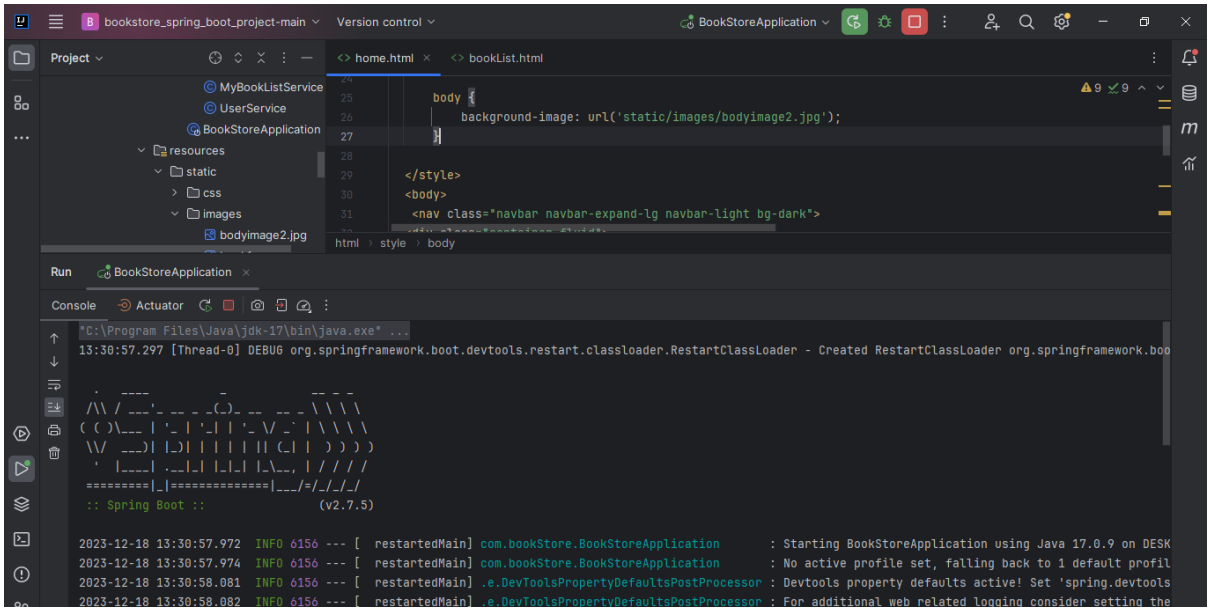
➔ GitHub Link:

IV) Database Schema:

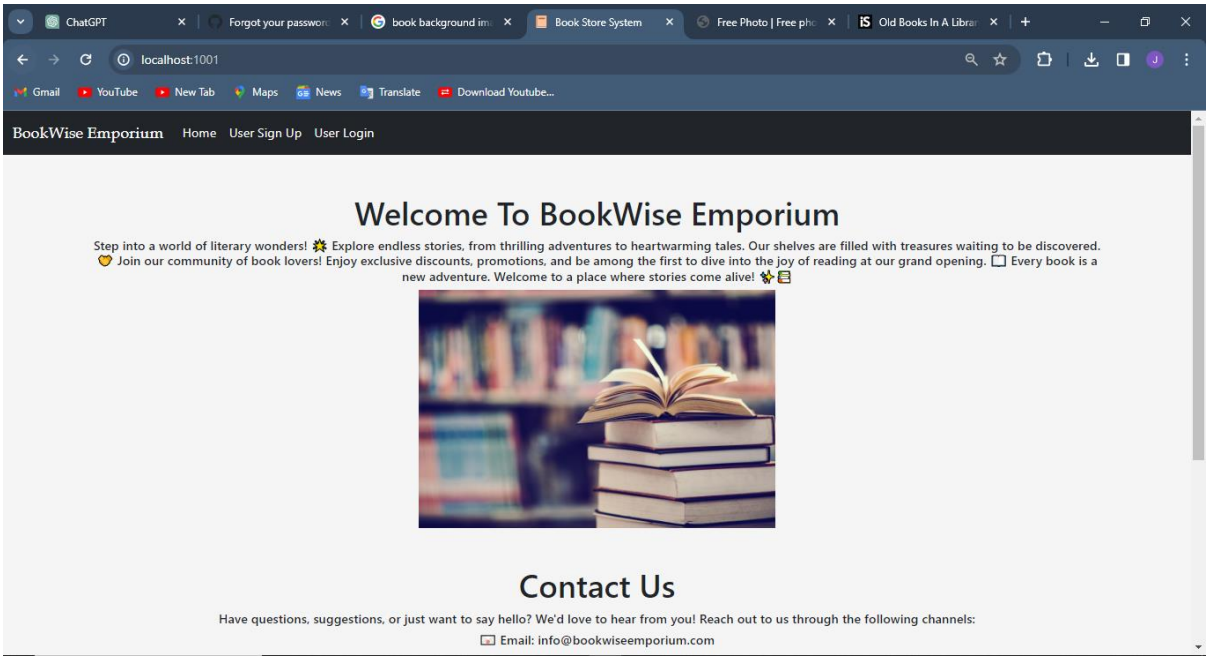


V) User Documentation:

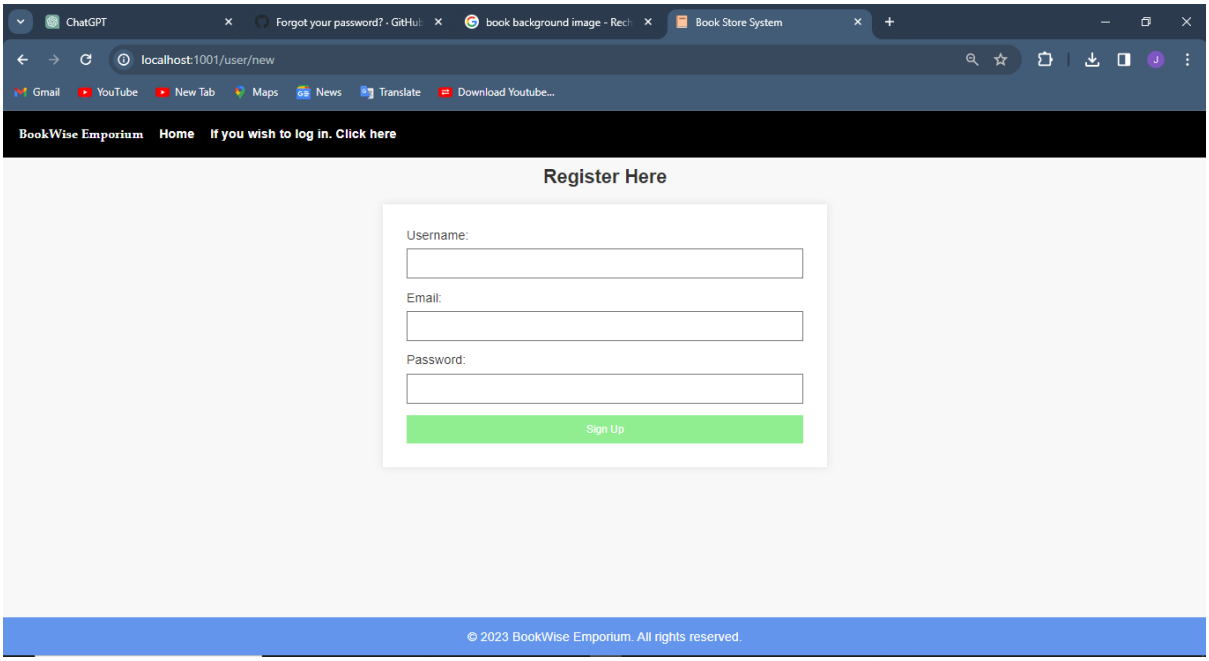
➔ **First is of all, we run the application, using localhost:1001 in any Browser.**



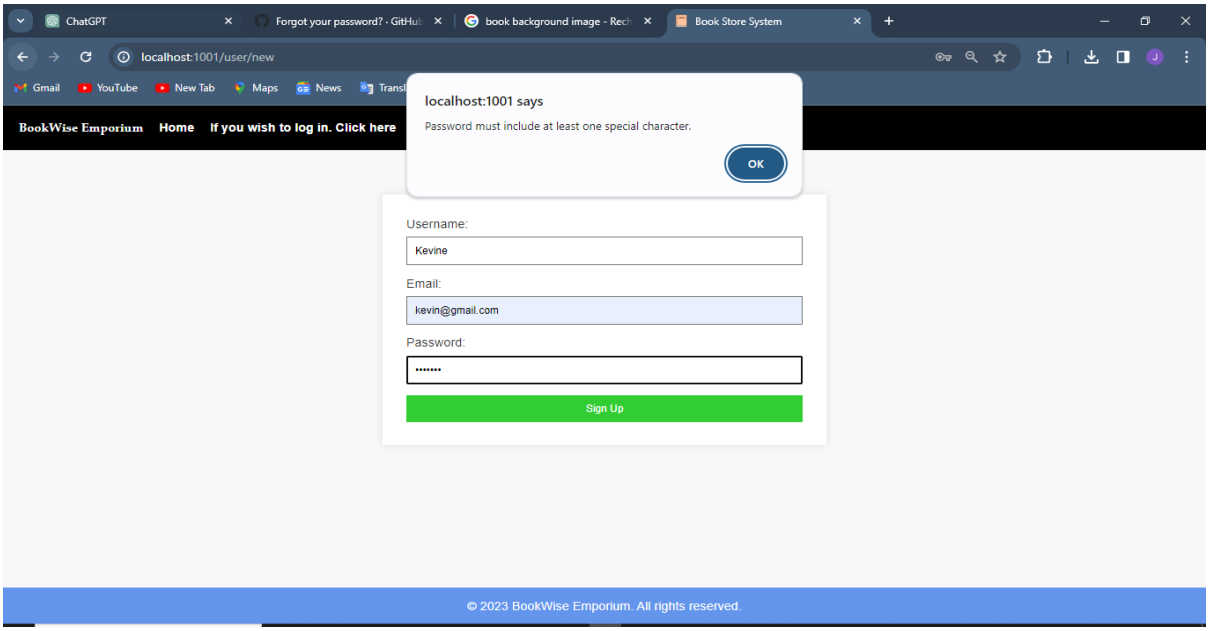
➔ **Then, this is Home Page:**



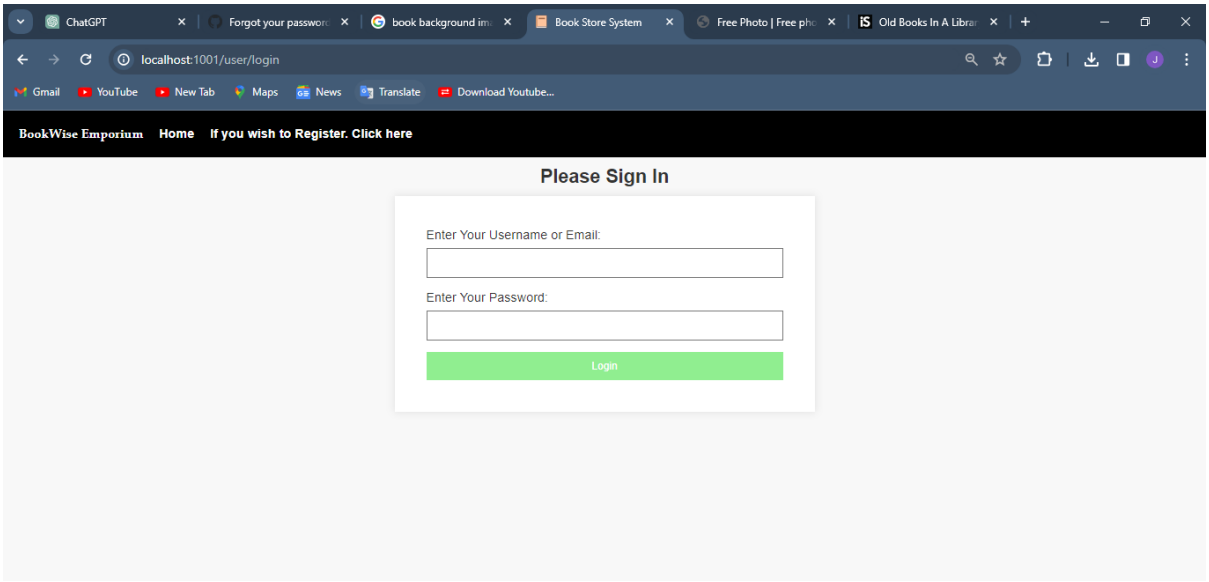
➔ **You can go to Sign Up page, to register if you don't have an account. This the Sign Up page:**



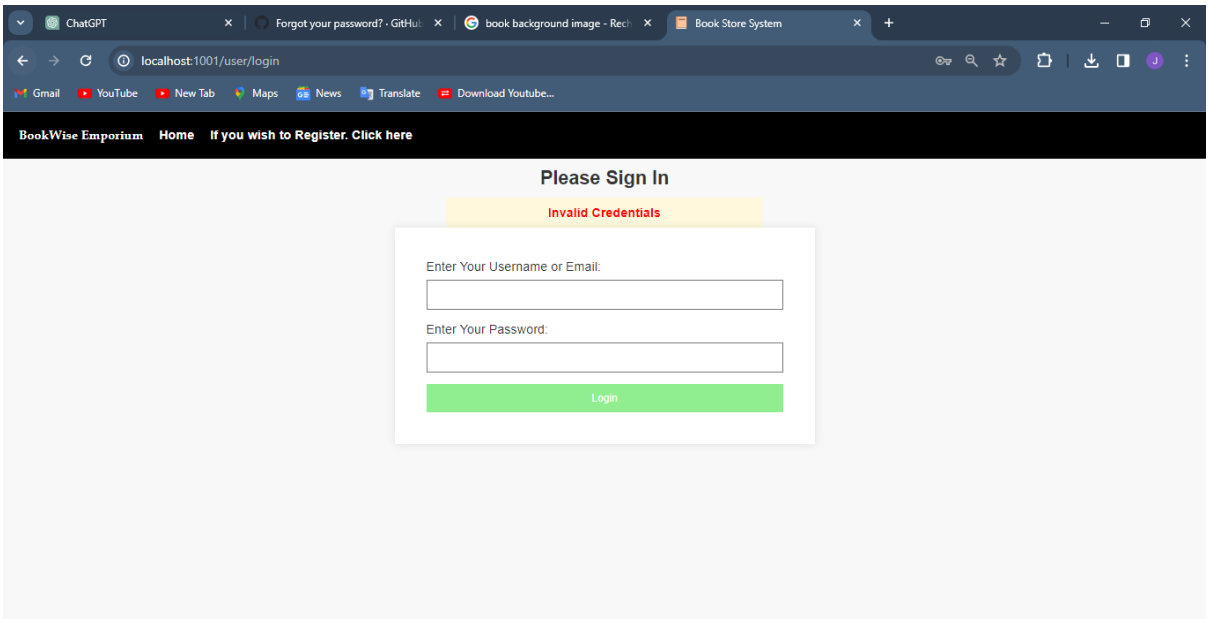
➔ **If you enter a password which doesn't have at least One Character, it will bring a message on the browser, and that is Validation:**



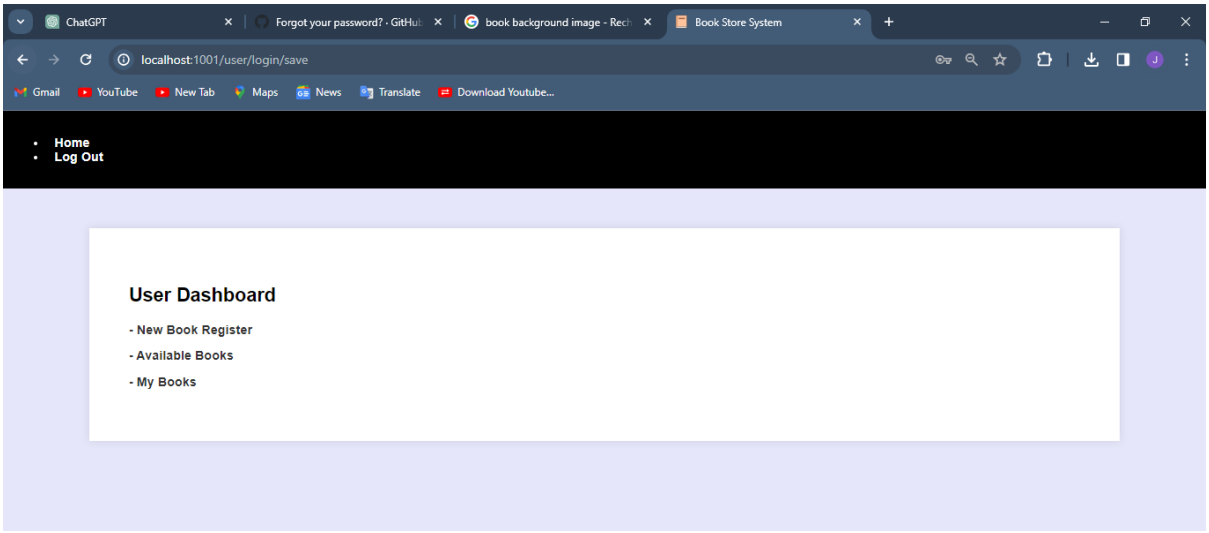
➔ **Then, after if successfully Signed Up, it redirects you to the Login page. This is the Login Page:**



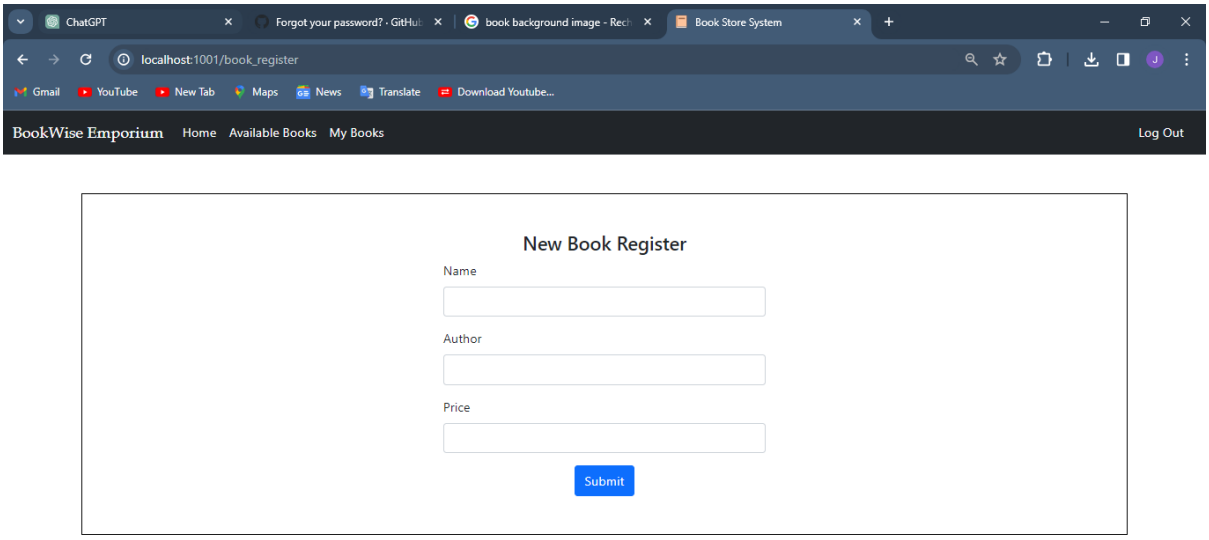
➔ **After, if the Credentials entered are not correct, it will bring a message Error (That' Authentication and Authorization):**



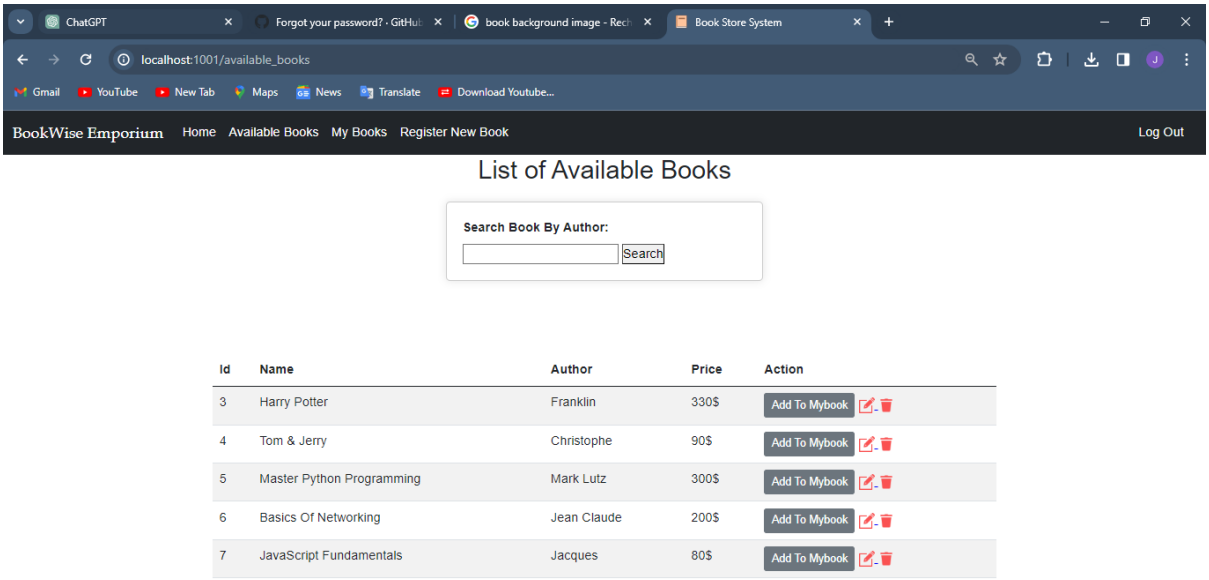
➔ **After logged in Successfully, it will bring you to the User Dashboard:**



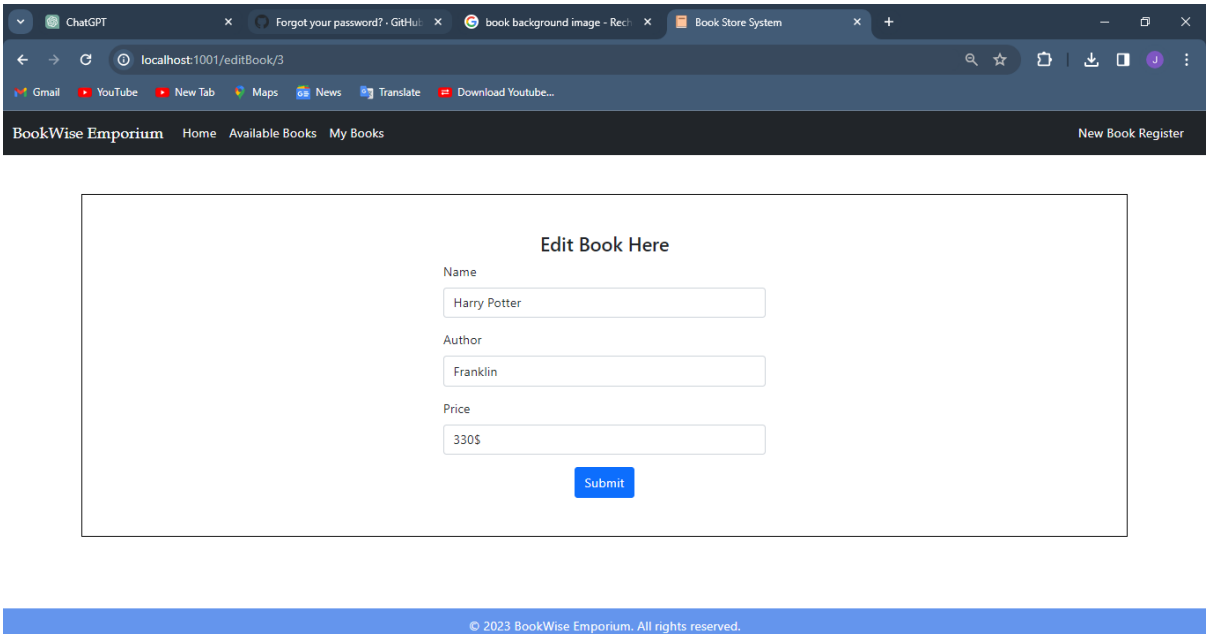
➔ **Then, if you want you can Register a Book, by Clicking there. Then, this is the Register Book page:**



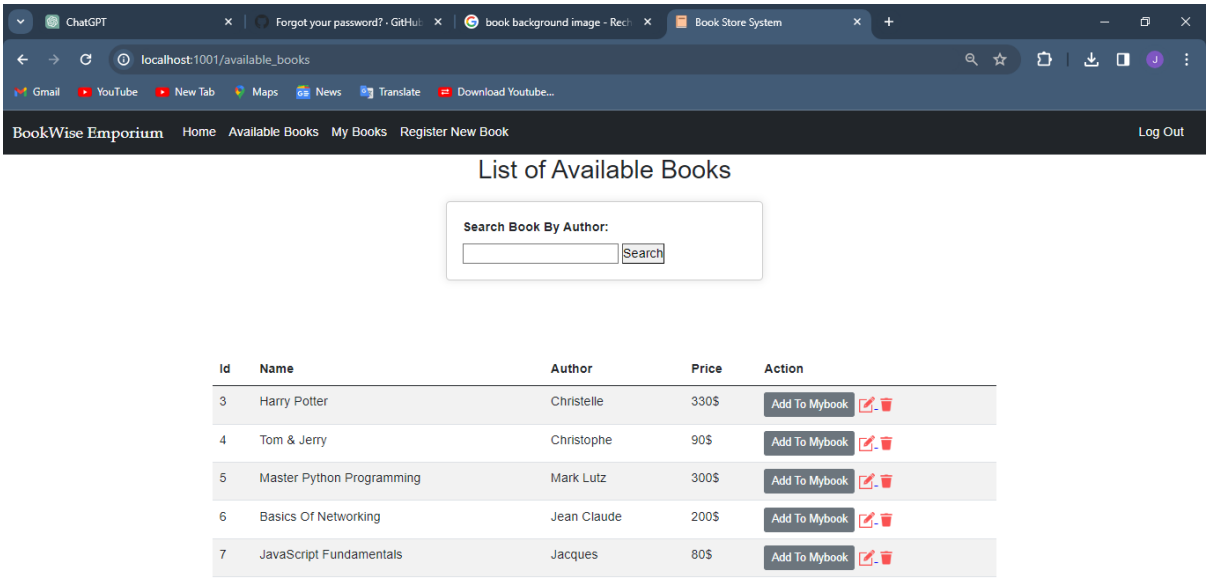
➔ **After, it redirects to the list of Available, where if you wish you can search a book by the Author, you add a book to your list of Books, or even delete a book, and edit a book. This is the page:**



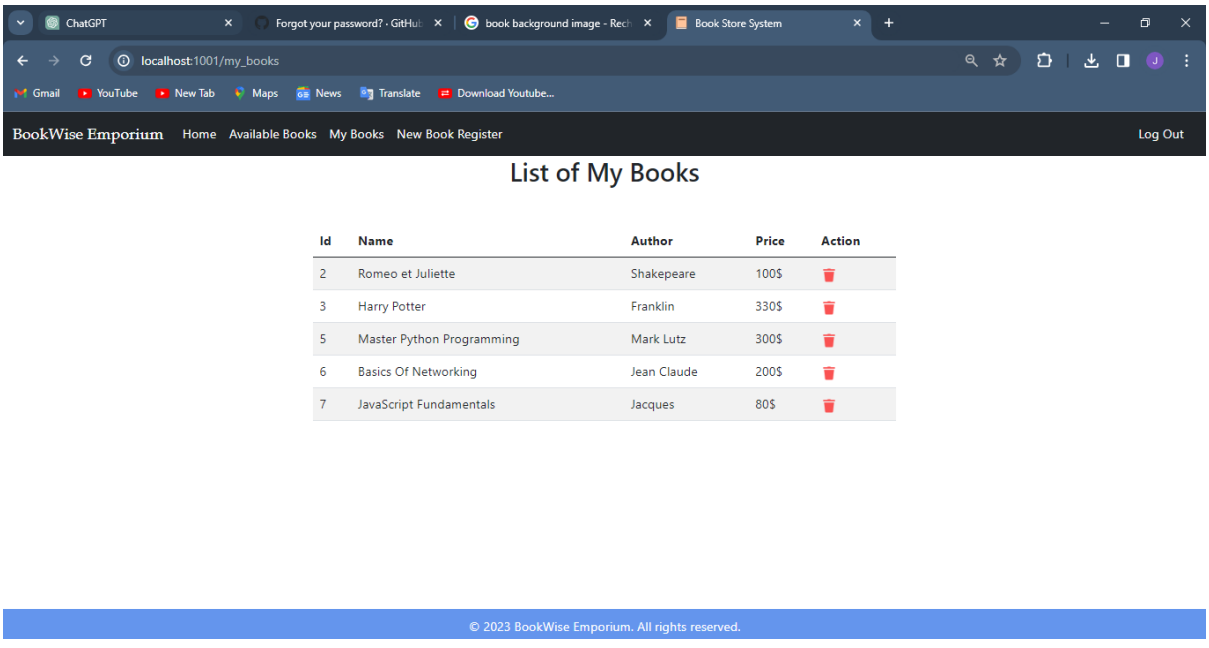
➔ **If you wish you can edit a book. This is the Edit Book page:**



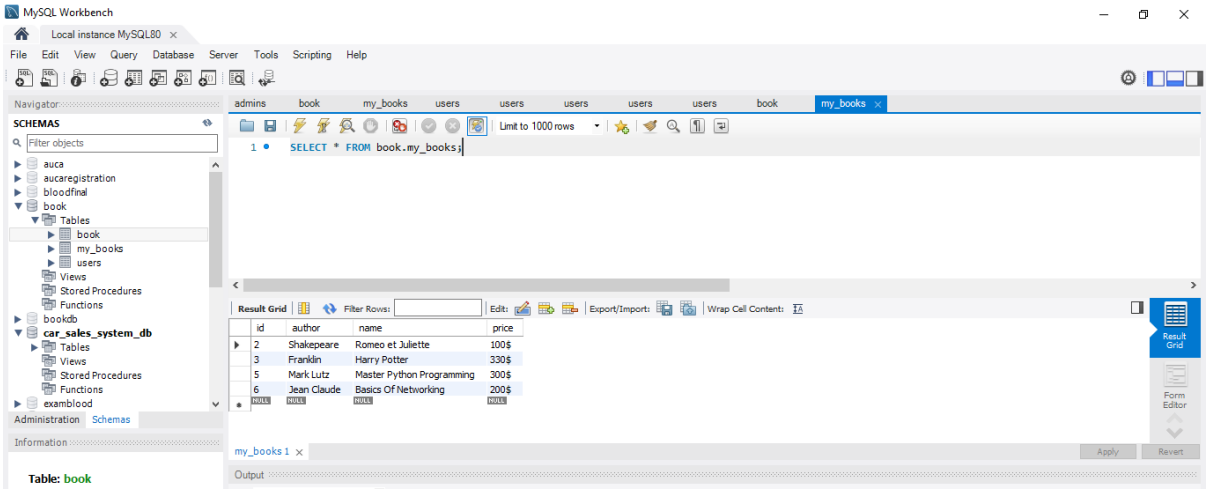
➔ **Then, if successfully edited, it will appear differently on the Available list:**



➔ **If you wish, you can also explore your own books, added from the Available Books. This is the list of my Books:**



➔ **This is the Database, MySQL view:**



VI) Technical Documentation:

1. Architecture of the Application:

1.1 Backend Architecture:

The backend of the Book Management System is developed using Java and the Spring MVC framework.

Java: The primary programming language used for backend development.
Spring MVC: Framework for building modular and scalable web applications. Follows the Model-View-Controller (MVC) architectural pattern.

1.2 Frontend Architecture:

The frontend of the application is developed using HTML, CSS, and JavaScript.

HTML/CSS: Responsible for structuring and styling the user interface.
JavaScript: Used for client-side interactions, form validations, and asynchronous requests.

1.3 Database Architecture:

The application uses a relational database to store user data and book information.

Database: MySQL is used as the relational database management system.
Database Schema: Follows a straightforward schema with tables for users and books. Refer to the provided database schema diagram for details.

Database Schema: Follows a straightforward schema with tables for users and books. Refer to the provided database schema diagram for details.

2. Implementation Details:

2.1 Backend Implementation:

User Authentication: Implemented secure user authentication and authorization using Spring Security.

User Management: CRUD operations for user accounts, utilizing Spring Data JPA for data access.
Book Management: CRUD operations for book data, ensuring data integrity and consistency.

2.2 Frontend Implementation:
User Interface: HTML templates for user registration, login, and dashboard views.
User Interaction: JavaScript for form validations and dynamic content updates.
Communication with Backend: AJAX requests to communicate with the backend for seamless data exchange.

3. Libraries/Frameworks Integrated:

3.1 Backend Libraries/Frameworks:
Spring Boot: Used for rapid application development and simplified configuration.
Spring MVC: Framework for building robust and scalable web applications.
Spring Data JPA: Simplifies data access layer implementation using Java Persistence API (JPA).

3.2 Frontend Libraries/Frameworks:
Bootstrap: Utilized for responsive and mobile-first frontend development.
jQuery: Used for simplifying DOM manipulation and AJAX requests.

___DONE___