

Project: Online Bookstore Management System

Background: In the age of digital transformation, brick-and-mortar bookstores are looking for innovative ways to manage their inventory and sales while offering online browsing options for their customers. Your task is to build an Online Bookstore Management System that not only helps the bookstore staff manage their inventory but also provides an interface for customers to browse available books and make purchase requests.

Requirements:

1. **Maven Project using Spring Initializer:** Start by setting up a Maven project via the Spring Initializer. This will ensure that all required dependencies and plugins are set up properly.
2. **Spring Boot REST API:** Develop a RESTful API using Spring Boot that will handle all CRUD operations related to books, such as adding a new book, fetching details of a book, updating book details, and deleting a book.
3. **Connection to a Database:** Connect your application to a database (e.g., MySQL or PostgreSQL) where all the book details will be stored. Implement CRUD operations to manage the book inventory.
4. **Spring MVC for the Backend:** Implement the backend logic using Spring MVC. This should manage request mappings, handle responses, and integrate with the database.
5. **Thymeleaf, CSS, and Bootstrap for the Frontend:**
 - Design a frontend that allows staff to manage the inventory and customers to browse the books.
 - Use Thymeleaf as the template engine to display the book data fetched from the backend.
 - Incorporate CSS for styling and Bootstrap to ensure the website is responsive and has a modern design.
6. **Spring Security:**
 - Implement a login form for the bookstore staff and customers.
 - Include a logout button to ensure users can log out securely.
 - Set up role-based controls: Staff should have permissions to add, update, or delete books, while customers should only be able to browse and make purchase requests.
 - Use JDBC authentication to verify user credentials against the database.
 - Implement password security using the bcrypt algorithm.
 - Create a custom access denied page for users trying to access unauthorized sections.
7. **DevTools and Actuator Dependencies:** Integrate Spring Boot DevTools to ensure faster development cycles by automatically restarting the application upon changes. Use the Actuator dependency to gain insights into the application's health, metrics, and other details.

Expected Outcomes:

1. A fully functional Online Bookstore Management System where staff can manage the inventory and customers can browse books.
2. Secure authentication and role-based authorization for staff and customers.
3. An intuitive and responsive frontend interface, streamlined backend, and stable database connection.

Evaluation Criteria:

- Proper implementation and structure of the Spring Boot application.
- Successful CRUD operations on the database.
- The effectiveness of Spring Security measures implemented.
- The design, responsiveness, and user-friendliness of the frontend.
- Proper usage of ThymeLeaf, CSS, and Bootstrap in the frontend design.
- Effective use of Spring MVC in handling backend logic.

Good luck! I'm excited to see the innovative solutions you develop using the Spring ecosystem.

Reza