# **Online Bookstore**

The **Online Bookstore** project is a comprehensive application that allows users to browse, search, and purchase books online. It involves multiple features and modules, providing a rich learning experience in Spring MVC and Thymeleaf.

**Project Overview:**

The Online Bookstore will have a user-friendly interface where users can browse a catalog of books, add them to their cart, and make purchases. The project will include an admin panel for managing the book inventory, user roles, and order processing.

**Key Modules & Features:**

1. **User Authentication & Authorization:**
   * **User Roles:** There will be at least two roles: *Admin* and *Customer*.
     + *Admin* can manage books, users, and orders.
     + *Customers* can browse books, add them to the cart, and place orders.
   * **Authentication:** Implement login and registration features with Spring Security.
   * **Authorization:** Control access to different sections of the website based on user roles.
2. **Book CatLog Management:**
   * **Book Listings:** Display a list of available books with details like title, author, price, and availability.
   * **Search & Filters:** Implement search functionality to find books by title, author, or genre. Filters can include price range, genre, and rating.
   * **Book Details Page:** When a user clicks on a book, they should see a detailed page with a description, cover image, author info, and reviews.
3. **Shopping Cart & Checkout:**
   * **Shopping Cart:** Users can add books to their cart, view items, update quantities, and remove items.
   * **Checkout Process:** The checkout process should include order summary, shipping details, and payment options. You can use mock payment integration (e.g., PayPal, Stripe).
   * **Order History:** Users can view their past orders and track the status of ongoing orders.
4. **Admin Panel:**
   * **Manage Books:** Admins can add new books, edit existing book details, and remove books from the inventory.
   * **Order Management:** Admins can view all orders, update their status (e.g., processing, shipped, delivered), and manage returns/refunds.
   * **User Management:** Admins can manage customer accounts, reset passwords, and deactivate accounts if necessary.
5. **Customer Reviews & Ratings:**
   * **Reviews:** Customers can leave reviews for books they have purchased.
   * **Ratings:** Implement a rating system for books. The average rating can be displayed on the book details page.
6. **Search Engine Optimization (SEO):**
   * Implement SEO-friendly URLs, meta tags, and sitemaps to make the bookstore easily discoverable by search engines.
7. **Responsive Design:**
   * Ensure that the bookstore is mobile-friendly and responsive across different screen sizes using Thymeleaf templates.
8. **Email Notifications:**
   * Send email notifications for order confirmation, shipping updates, and promotional offers.

**Technologies & Tools:**

* **Backend:**
  + **Spring MVC:** To manage the business logic and route requests.
  + **Spring Security:** For handling authentication and authorization.
  + **Spring Data JPA:** For database interaction, managing entities, and queries.
  + **MySQL:** As the relational database to store user data, book details, orders, etc.
* **Frontend:**
  + **Thymeleaf:** To render dynamic content on the frontend with templates.
  + **HTML/CSS/Bootstrap:** For creating a responsive and attractive user interface.
  + **JavaScript:** For interactive elements like dynamic search filters, cart updates, etc.
* **Other Tools:**
  + **Maven/Gradle:** For project build and dependency management.
  + **Git:** For version control.
  + **JUnit:** For unit testing the application.

**Learning Outcomes:**

By working on this project, you'll gain hands-on experience with:

* **Spring MVC architecture:** Understanding how to manage controllers, services, and repositories.
* **Thymeleaf:** Mastering the template engine for generating dynamic content.
* **Spring Security:** Implementing secure authentication and authorization.
* **RESTful APIs:** (Optional) If you choose to separate frontend and backend, you can create REST APIs for the frontend to consume.
* **Database Design:** Designing relational databases with entities like Users, Books, Orders, etc.
* **Full-Stack Development:** Combining both backend and frontend skills to create a complete web application.

This project will be an impressive addition to your portfolio, demonstrating your ability to build a robust and secure e-commerce application from scratch.