

# Bookstore System

Sprint 2 projects for online Bookstore system using Spring boot and MySQL Database.

user credentials stored in MySQL database. Tokens (persistent-logins) are stored in the database as well. Passwords are encrypted with the BCrypt algorithm.

**A Spring Boot web app with the following features:**

- Login and registration pages
- Password reset workflows
- Edit profile page
- Thymeleaf templates

## Getting Started

To install this application, run the following commands:

```
git clone https://github.com/janeodum/BookstoreSystemEclipse.git
```

## Required Technologies

- Eclipse EE
- Spring Boot Tools
- Mysql
- Mysql Workbench.

## Libraries used:

- Spring boot
- Spring Data JPA and Hibernate
- MYSQL (for database management)
- Spring security

## Database description:

Below are the main tables that will be used in the authentication/authorization process.

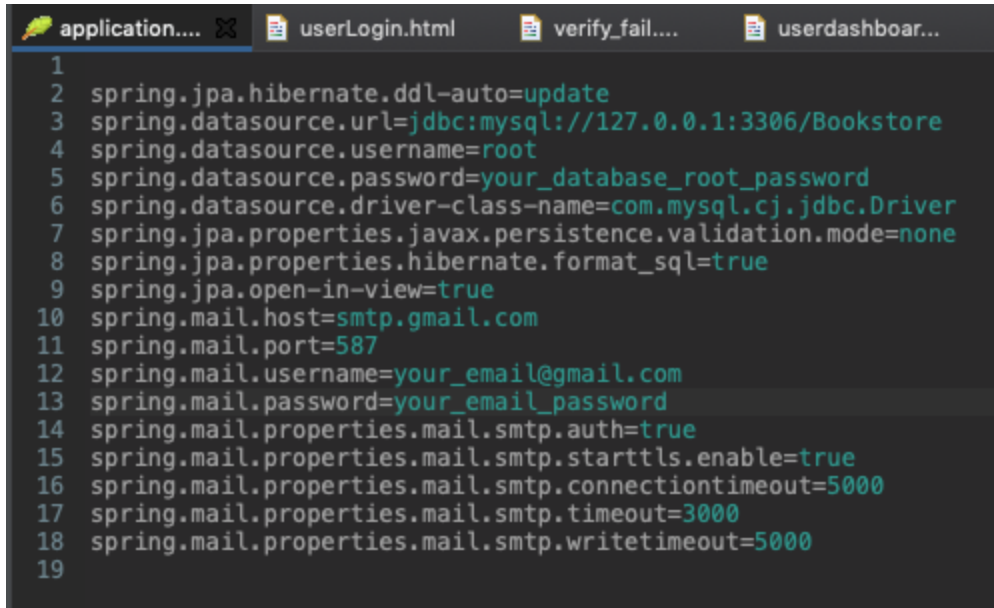
- *Users* table: Contains basic user details (email and password) used for the user to login. Here BCrypt encrypted passwords are stored and also *user\_type*(admin or user) used for the user privileges.
- *featured\_books* table: Contains books details displayed on the home page.
- *Payment\_information* table: Contains user payments details it also stores the user's email which serves as a foreign key in the payment table.
- *Persistent\_token* table: Contains remember-me access tokens for authenticated users.
- *Address\_information* table: Contains information about user address details.
- *Promotion\_information*: Contains information about users who have subscribed for promotional email. It stores the user's email address and phone number as a foreign key.

## NOTE:

You would need to create a Database called Bookstore in your MYSQL Workbench and add the connection details in the *app.properties* file. Since the requirement states that the admin does not need to register to have an account open, you would need to manually enter admin information in the database after the user table has been created. To do so simply enter "admin" in the *user\_type* column after the user account has been created and when the user is authenticated, based on their type they would be redirected to the appropriate page.

## Project configurations - *application.properties*:

- Database configuration: You can modify database-related configurations in the *application.properties*.
- For email SMTP I used Gmail for this project.
- Note that once you add your Gmail and run the code you might encounter an SMTP error depending on your Gmail security configuration. To correct this simply turn on Less secure app setting on your Gmail account. If you also have a two-factor authentication you would need to turn off that too. Or as I advise, open a new Gmail account for the purpose of testing this application.



```
1
2 spring.jpa.hibernate.ddl-auto=update
3 spring.datasource.url=jdbc:mysql://127.0.0.1:3306/Bookstore
4 spring.datasource.username=root
5 spring.datasource.password=your_database_root_password
6 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
7 spring.jpa.properties.javax.persistence.validation.mode=none
8 spring.jpa.properties.hibernate.format_sql=true
9 spring.jpa.open-in-view=true
10 spring.mail.host=smtp.gmail.com
11 spring.mail.port=587
12 spring.mail.username=your_email@gmail.com
13 spring.mail.password=your_email_password
14 spring.mail.properties.mail.smtp.auth=true
15 spring.mail.properties.mail.smtp.starttls.enable=true
16 spring.mail.properties.mail.smtp.connectiontimeout=5000
17 spring.mail.properties.mail.smtp.timeout=3000
18 spring.mail.properties.mail.smtp.writetimeout=5000
19
```

## How to run this project:

To run the project, if you cloned my repo on GitHub, open the command-line at the project's root directory, and run this command: `mvnw spring-boot:run`. It will automatically create the database, required tables, and insert preliminary data. Or simply open your eclipse IDE and click on FILE → OpenFile and open the file folder where you have it saved. You can also use IntelliJ IDEA to import it as a Maven project. Right-click on the Bookstoresystem file and click Run As spring boot application. Once you have had the spring boot application ran, note that the featured books table has not been populated with book details because the admin page has not been implemented so for now we would need to run an SQL script to insert book details in the table. To populate the featured books table open the database folder under the resources folder and copy the script. Run the script in your MySQL workbench.

Now open your browser and go to [localhost:8080/](http://localhost:8080/) to access the homepage of our website! Enjoy!