**Library Rent Book Application**

**Question Scenario**

The ***Library Rent Book Application*** is built using Java Spring Boot with Thymeleaf*.* This application is used to maintain the relationship between books and recipients.

The application consists of 4 entities; Book, User, Admin and Notification. This application allows the user to rent the books for a certain period of time, if books are available (Booking screen will show the book details, Book title, Book description and Cost of Rent per hour) then user has to pay the specified rent on the books for lending which will be credited to admin. Once the user rents any book, notification will be sent to both the admin and user accordingly.

1. User wants to search a book by its name or author’s name, if books are available display the list of available books.

2. On click of any book, the user should be navigated to the Book Details page, where the user has an option to rent the selected book by giving the number of hours that user wants to rent.

3. Once the user rents any book, notification should be sent to both the admin and user (Data should be inserted in the notification table and consider the first user data present in database always). Amount specified on that book based on his rental hours (Number of hours\*rental price of the book) should be credited to the admin.

**Objectives**

***Database*** layer with required code exists in the application. Complete the logic of methods present in ***Presentation*** (which contains views and controllers) and ***Business*** (service) layers. You are allowed to create the ***Persistence/DAO*** (repository)layer according to the requirement. You are not allowed to change the method signature of ***Presentation*** and ***Business*** layers.

We have defined below methods in ***Presentation*** and ***Business*** layers.

**1**. You must complete the body of the function to implement ***search*** functionality. You also need to create a **Book *List*** screen as a result of ***search.***

**Presentation Layer:**

**@PostMapping("/searchBooks")**

**public String searchBooks(BookDTO bookDTO, ModelMap map)**

**Business Layer:**

**public Page<Book> searchBooksByBookName(String bookName, Pageable pageable)**

**public Page<Book> findByAuthorName(String authorName, Pageable pageable)**

**2**. You must complete the body of the function to fetch the details of a single book. You also need to create a **Book *Details*** screen which also displays the ***Total Cost*** based on the number of hours given by the user.

**Presentation Layer:**

**@GetMapping("/getBookDetails/{slNo}")**

**public String bookDetails(@PathVariable Integer slNo, ModelMap map)**

**Business Layer:**

**public BookDTO findBySlNo(Integer slNo)**

**3**. You must complete the body of the function to send the notification to both user and admin after the click of the Rent ***Book*** button (A record has to be inserted in the notificationtable). Increment the ***amount*** column of the admintable. Change the availability of books to ***false*** in the booktable.

**Presentation Layer:**

**@GetMapping("/rentBooks/{noOfHours}/{bookPrice}/{userId}/{adminId}/{bookId}")**

**public String rentedBooks(@PathVariable Integer noOfHours, @PathVariable Integer bookPrice,@PathVariable Integer userId, @PathVariable Integer adminId, @PathVariable Integer bookId, ModelMap map)**

**Business Layer:**

**public String rentBook(Integer userId, Integer adminId, Integer bookId, Integer amount)**

**Instructions**

You are not allowed to make any changes in below files.

1. application.properties
2. pom.xml

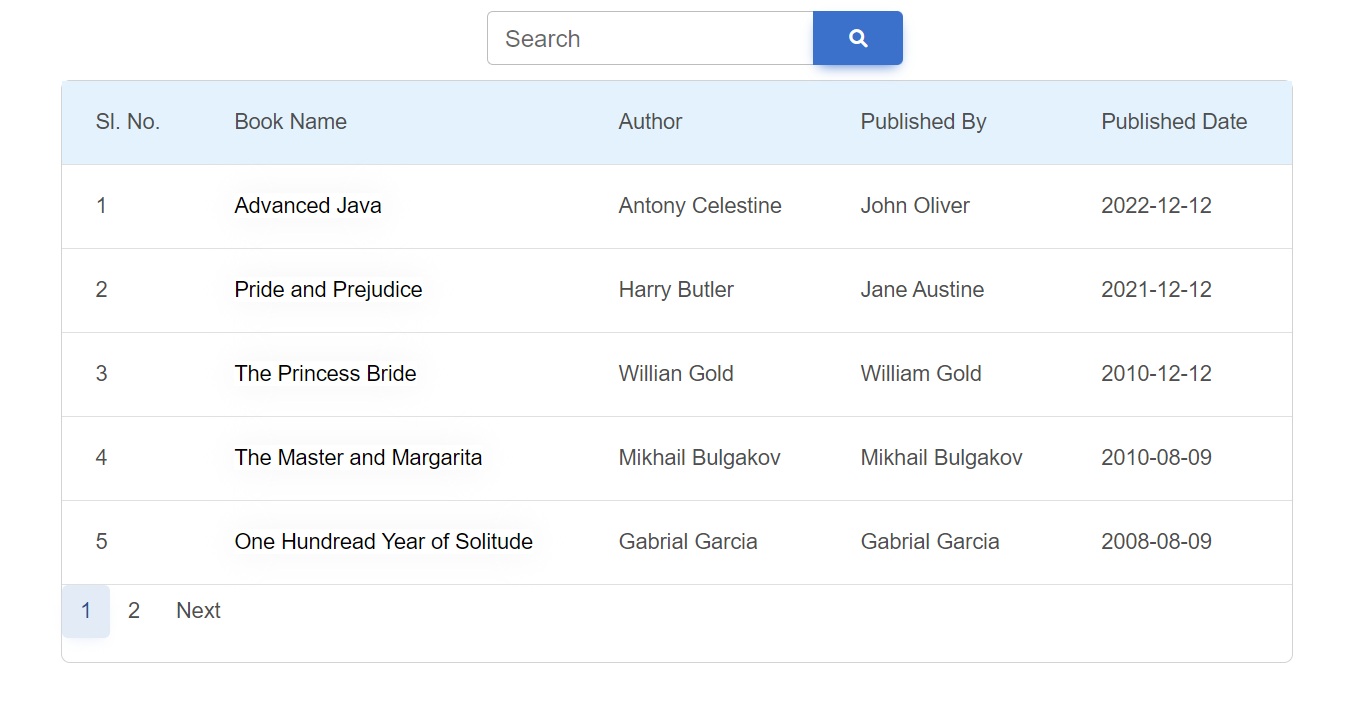
You are not allowed to make any changes in the files present in below packages.

1. com.demo.example.pojo
2. com.demo.example.dto
3. com.demo.example.repository

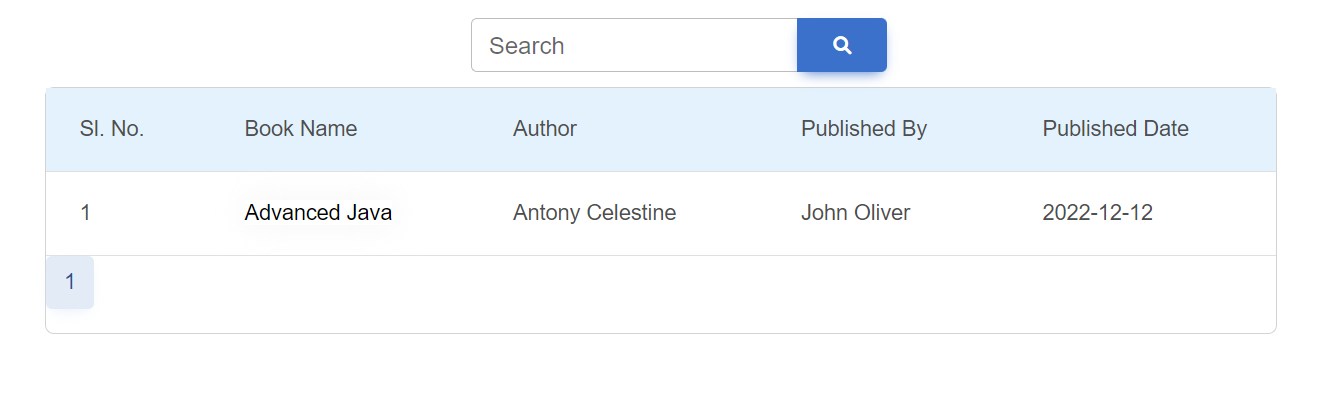
Also, you are not allowed to perform any action which manipulates the existing data present in the database and insert new data to the other tables except the notification table.

**Output**

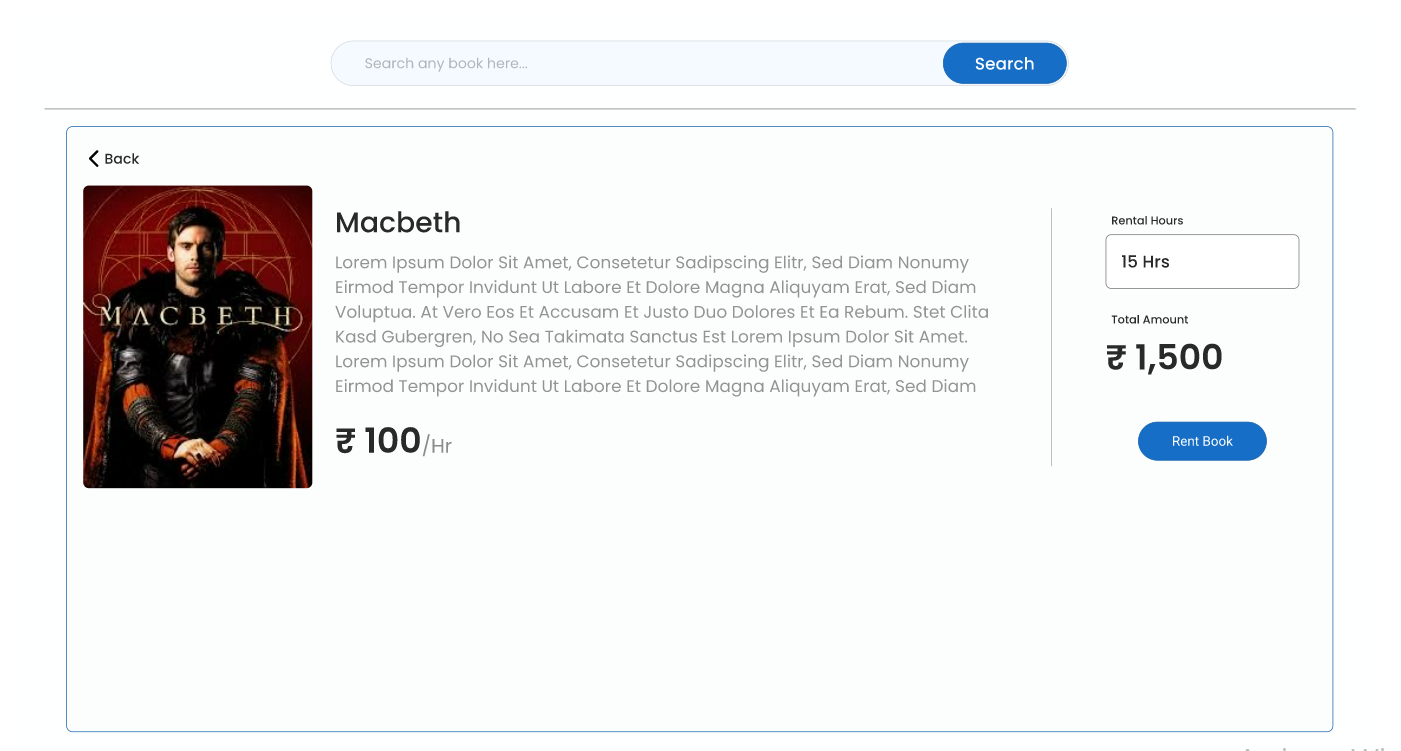
**Before clicking on Search, the page will look as shown below, where it displays all the available books.**



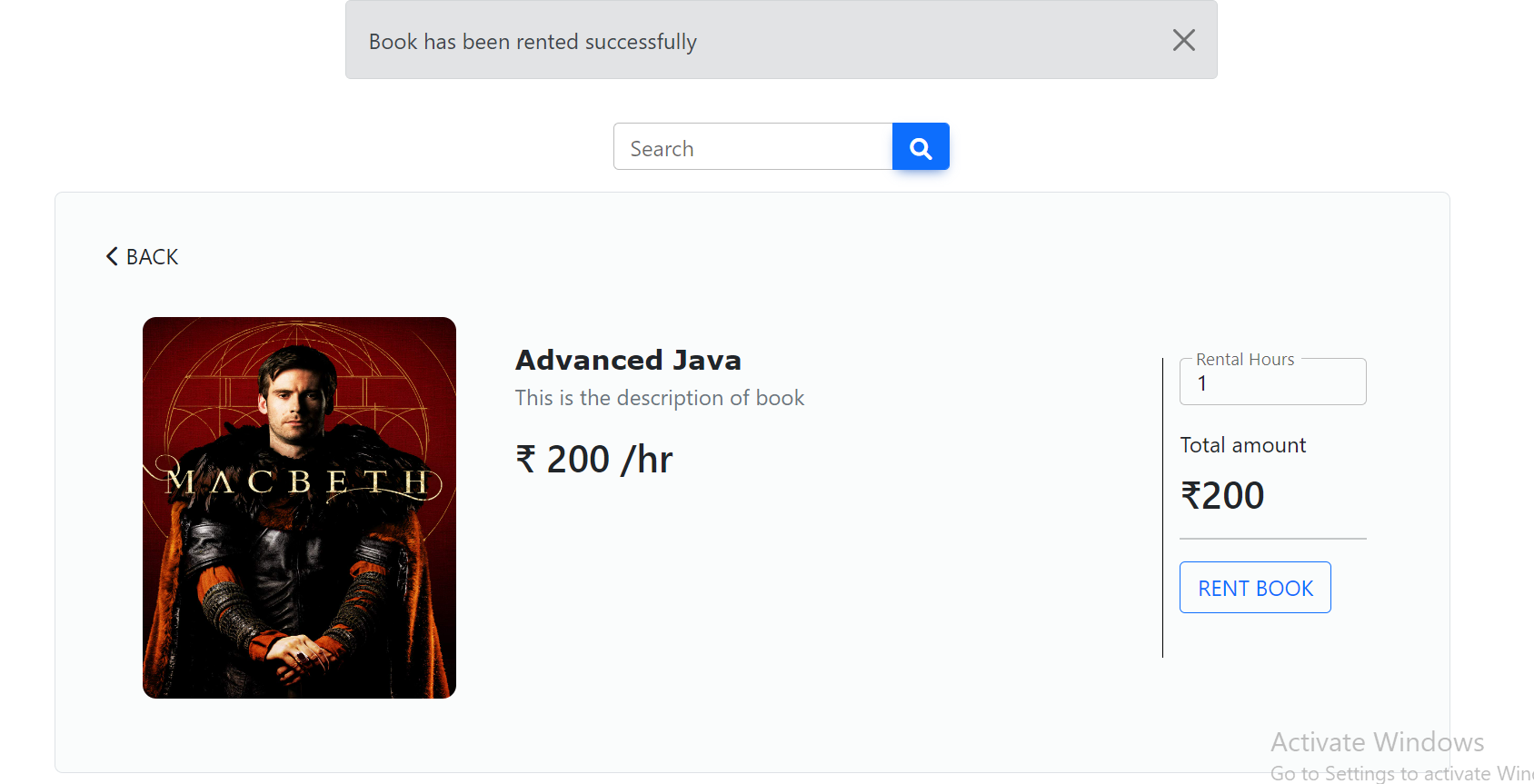
**User can search the books either by its name or author’s name.**



**On click of book name, the user will be navigated to the *Book Details* page. User can enter the number of hours to rent the book. Once he clicks on enter, total cost will be displayed. Book image can be any static image.**

****

**On click of the *Rent Book* button, success message will be displayed.**

****

**Project Files**

You can only work upon the below files.

1. BookController.java
2. BookService.java