RentRead

# Problem Statement

Develop a **RESTful API** service using **Spring Boot** to manage an online book rental system while using **MySQL** to persist the data.

# Key Features

* Please note that this is a simplified version of a book rental system, and you should focus on implementing the specified features effectively within the given constraints
* The service must implement **authentication** and **authorization**
* The service uses **Basic Auth**
* The service must have two roles: **USER** and **ADMIN**
* The service must have two types of API endpoints:
  + Public endpoints - Anyone can access (Ex. Registration, Login)
  + Private endpoints - Only **authenticated** users can access (Ex. GET all books)
* The private endpoints also require **authorization** i.e. only users with specific permissions can access the endpoint (Ex. Creating (POST) a book is only allowed for the admin)

**Note:** Some of the design choices are left to you. For example, the requirement may state that the users must be able to rent a book using the service. You can either let the users with the role “USER” rent a book or both the “USER” and the “ADMIN”. Technically, both approaches are correct but be prepared to defend your design choices. Designing the **database schema** is another critical decision you must make and defend.

The **API** must have the following features:

## User Registration and Login

* Users must be able to register by providing their email address and password
* The password must be hashed and stored using **BCrypt**
* Fields: Email, Password, First Name, Last Name, Role
* The Role must be defaulted to “User” if it is not specified
* Registered users must log in using their email address and password

## Book Management

* Store and manage book details
* Fields: Title, Author, Genre, Availability Status
* Availability Status tells whether the book is available to rent or not
* Any user can browse all the available books
* Only the administrator is allowed to create, update, and delete books

## Rental Management

* Users must be able to rent books using the service
* A user cannot have more than two active rentals i.e. the service should throw an error if a user requests to rent a book while already having two other book rentals
* Users must be able to return books that they have rented

# Additional Requirements

* Use logs to **log** information and errors
* Handle common errors gracefully and return **appropriate HTTP codes** (Ex. 404, User not found)
* Include basic unit tests while making use of **MockMvc** and **Mockito** (Minimum 3)
* Publish your code to a public **GitHub** repository
* Write meaningful, **incremental** commit messages
* Include a descriptive **README.MD** for your application codebase
* Generate a **JAR** file for your application and provide instructions on how to run it
* Create and add a public [**Postman**](https://www.postman.com/) **Collection** in the README.MD (Optional)

# Endpoints

* POST /books/{bookId}/rent - For renting a book
* POST /books/{bookId}/return - For returning a book
* You are required to design other RESTful endpoints based on the requirements

# What to Submit?

* You will be submitting your GitHub code repository for this assignment.
* Note: An activity will be part of your program to collect this submission.

# Additional Resources

* [Local Environment Setup - Backend](https://docs.google.com/document/d/1LbRboQXtkjvto8ftQnX0JnwjQsy96nECqyTimeMX7Fg/edit) - For setting up your local environment
* [Setting Up Applications Using Spring Initializr](https://docs.google.com/document/d/1pUot5Sf6XdY2jDX5oTr5CP-1cZ7eBt0NoyOqpinxAuY/edit#heading=h.h2q5unqavex1) - To learn about generating boilerplate code with Spring Initializr, adding dependencies, integrating databases, and Spring Boot best practices **(Added section on Spring Security integration)**
* [Template for Backend Takehomes](https://docs.google.com/document/d/15FD73sysjd92ubZ50SkQ3wzyeeivMSmCnOaLbNGh9qI/edit#heading=h.3p60com67j8r) (New pointers on **Authentication/Authorization**)
* [Logging with @Slf4j in Spring Boot & Lombok | Medium](https://medium.com/@AlexanderObregon/enhancing-logging-with-log-and-slf4j-in-spring-boot-applications-f7e70c6e4cc7) - Introduction to Logging
* Make sure to initialize a new repository for every project on GitHub. Use one of the below for the necessary steps:
  + [Installing Git and Creating a Repository](https://medium.com/analytics-vidhya/github-tutorial-1-installing-git-and-creating-a-repository-984dc0447684) OR
  + [How to Add a New Project to GitHub Repository with Visual Studio Code](https://www.youtube.com/watch?v=ATR5XJwDyJY&t=271s)
* [Postman Collections - Getting Started](https://learning.postman.com/docs/getting-started/first-steps/creating-the-first-collection/) and [Postman Collections - Learning More](https://learning.postman.com/docs/collections/collections-overview/)
* [Basic writing and formatting syntax for README.MD](https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax) and [Markdown Cheatsheet](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)