RentVideo

**Problem Statement**

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

# Key Features

This is a simplified version of a video rental system. You should focus on implementing the specified features effectively while adhering to the following design principles:

* The service must implement authentication and authorization.
* The service uses **JWT** tokens for **stateless** **authentication.**
* The service must have two roles: **CUSTOMER** and **ADMIN**.
* The service must have two types of API endpoints:
  + **Public endpoints**: Anyone can access them (e.g., Registration, Login).
  + **Private endpoints**: Only authenticated users can access (e.g., Viewing available videos).
  + Private endpoints must also enforce authorization, i.e., only specific roles can access certain endpoints (e.g., managing videos should only be allowed for the **ADMIN** role).

**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, password, and role.
* The password must be hashed using BCrypt.
* Fields: Email, Password, First Name, Last Name, Role.
* The Role should default to **CUSTOMER** if not specified.
* Registered users must log in using their email address and password (Basic Auth).

## Video Management

* The system must store and manage video details.
* Fields: Title, Director, Genre, Availability Status (whether the video is available for rent or not).
  + Assume that all videos are available to rent
* Any user can browse the list of available videos.
* Only the **ADMIN** is allowed to create, update, and delete videos.

## Rental Management

* Users must be able to rent videos using the service
* A user can have up to two active rentals i.e. the service should throw an error if a user requests to rent a video while already having two other video rentals.
* Users must be able to return the videos that they have rented

# Endpoints

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

# Additional Requirements

* Handle common errors and return appropriate HTTP codes (e.g., 404, entity not found).
* Unit tests for essential operations. (Optional).

# 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
* [Template for Backend Takehomes](https://docs.google.com/document/d/15FD73sysjd92ubZ50SkQ3wzyeeivMSmCnOaLbNGh9qI/edit#heading=h.3p60com67j8r)
* [What is a Foreign Key Constraint? Understanding Primary & Foreign Keys](https://www.youtube.com/watch?v=5kiMg7GXAsY)
* [Spring Boot Global Exception Handler | by Ahmet Emre DEMİRŞEN | Medium](https://medium.com/@aedemirsen/spring-boot-global-exception-handler-842d7143cf2a)
* 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)
* If you are new to building Spring Boot Applications using Spring Data JPA and MySQL, you can take a look at this project: [Spring Boot Project | Banking Application using Spring Boot 3, Spring Data JPA (Hibernate), & MySQL](https://www.youtube.com/watch?v=5i379k0Xh_s)