StayEase

Problem Statement

Develop and **deploy** a **RESTful API** service using **Spring Boot** to streamline the room booking process for a hotel management aggregator application. You are required to use **MySQL** to persist the data.

Key Features

- Please note that this is a simplified version of an online room booking system, and you should focus on implementing the specified features effectively within the given constraints
- You can make the following assumptions:
    - The application only has a single type of room and all bookings are for two guests
    - Any hotel manager can update any hotel details i.e you do not have to keep track of who manages which hotel
    - Another service handles check-in and check-out functionalities
- The service must implement **authentication** and **authorization**
- The service uses **JWT** tokens for **session management**
- The service must have three roles: **CUSTOMER, HOTEL MANAGER,** 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. Book a room)

Note: Some of the design choices are left to you. All design decisions such as designing the database schema, and providing resource access based on roles must have a thorough thought process behind them.

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 encrypted and stored using **BCrypt**
- Fields: Email, Password, First Name, Last Name, Role
- The Role must be defaulted to “Customer” if it is not specified
- A JWT token must be generated upon successful registration or login

Hotel Management

- Store and manage hotel details
- Fields: Hotel Name, Location, Description, Number of Available Rooms
- The number of available rooms indicates whether a booking can be made or not
- Anyone can browse all the available hotels (Public endpoint)
- Only the administrator is allowed to create and delete hotels
- The hotel manager can only update the hotel details

Booking Management

- Customers must be able to book rooms using the service
- A single room can be booked per request
- Only hotel managers are allowed to cancel the booking

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** **Collection** in the README.MD (Optional)

Endpoints

- POST /hotels/{hotelId}/book - For making a booking
- DELETE /bookings/{bookingId} - For cancelling a booking
- 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
- You are also required to submit the deployed link of your application
- Note: An activity will be part of your program to collect this submission

Additional Resources

- Local Environment Setup - Backend - For setting up your local environment

- Setting Up Applications Using Spring Initializr - To learn about generating boilerplate code with Spring Initializr, adding dependencies, integrating databases, and Spring Boot best practices
- Template for Backend Takehomes
- Spring Security: Implementing JWT Authentication in a RESTful Spring Boot Application
- Deploying Spring Boot Applications using Render - **For deploying Spring Boot Applications Using Render (Contains a section on hosting MySQL Database)**
- Logging with @Slf4j in Spring Boot & Lombok | Medium - 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 OR
    - How to Add a New Project to GitHub Repository with Visual Studio Code
- Docker Crash Course for Absolute Beginners [NEW]
- Postman Collections - Getting Started and Postman Collections - Learning More
- Basic writing and formatting syntax for README.MD and Markdown Cheatsheet