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 stateless **authentication.**
* 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 hashed 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**](https://www.postman.com/) **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](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)
* [Spring Security: Implementing JWT Authentication in a RESTful Spring Boot Application](https://medium.com/@inni.chang95/spring-security-implementing-jwt-authentication-in-a-restful-spring-boot-application-657493fc0ae6)
* [Deploying Spring Boot Applications using Render](https://docs.google.com/document/d/18ZWLVcauCESi3TEU13O4SJb5mivVp1mKkQt7zx_pZLU/edit#heading=h.urmr0qrpl4mt) - **For deploying Spring Boot Applications Using Render (Contains a section on hosting MySQL Database)**
* [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)
* [Docker Crash Course for Absolute Beginners [NEW]](https://www.youtube.com/watch?v=pg19Z8LL06w)
* [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)