* Project Scope and Objectives:

A : The project is 'Hotel room booking application' which is used to book rooms for a single hotel. we will be creating a microservices-based backend for the application.

* Project Requirements:

A : We are breaking the Hotel room booking application into three different microservices , which are as follows:

1 : API-Gateway - This service is exposed to the outer world and is responsible for routing all requests to the microservices internally.

2 : Booking service - This service is responsible for collecting all information related to user booking and sending a confirmation message once the booking is confirmed.

3 : Payment service - This is a dummy payment service; this service is called by the booking service for initiating payment after confirming rooms.

4 : Eureka Server : connect all api to eureka server.

* Technical Details :

- Technology : Java, SpringBoot, Mixroservices, Mysql, SonarQube, JMETER, Git.
- Optional Technology - UI developement using Angular.

Booking Service:

1 :Spring Cloud Netflix Eureka Client
2 :Spring Boot Web
3 :Spring Boot Data JPA.

Payment Service:

1 :Spring Cloud Netflix Eureka Client
2 :Spring Boot Web
3 :Spring Boot Data JPA

API Gateway:

1 :Dependencies: Spring Boot Actuator, Spring Cloud Netflix Eureka Client
2 :Configure properties

Eureka Server:

1 :Dependencies: Spring Cloud Netflix Eureka Client
2 :Open the Eureka server and annotate the main class with proper annotation so that the Eureka server gets enabled.
3 :Set properties for running standalone Eureka servers.
4 :Set port for Eureka server as 8761.

* Proof of Concept Scope:

1 : Booking Service :

This service is responsible for taking input from users like- toDate, fromDate, aadharNumber and the number of rooms required (numOfRooms) and save it in its database.

1.1 Model Classes:

Refer to the “booking” table in the schema to create the entity class named “BookingInfoEntity”.

1.2 Controller Layer:

Endpoint 1 : This endpoint is responsible for collecting information like fromDate, toDate, aadharNumber, numOfRooms from the user and save it in its database.

URI: /booking
HTTP METHOD: POST
RequestBody: UserName, aadharNumber, fromDate, toDate,numOfRooms.
Response Body : UserName, aadharNumber, fromDate, toDate,numOfRooms, totalPrice,

2. Payment Service:

This service is responsible for taking payment-related information- paymentMode, upiId or cardNumber, bookingId and returns a unique transactionId to the booking service. It saves the data in its database and returns the transactionId as a response.

1.1 Model Classes:

Refer to the “transaction” table in the schema to create the entity class named “TransactionDetailsEntity”.

1.2 Controller Layer:

Endpoint 1: This endpoint is used to imitate performing a transaction for a particular booking. After receiving the transactionId from 'Payment' service, confirmation message is printed on the console.

URL: /transaction
HTTP METHOD: POST
RequestBody: paymentMode, bookingId, upiId, cardNumber.
Response Body : confirmation message is printed on the console.

* Summary

Hope you have now understood the problem statement well.

All The Best