System Design and implementation interview Task:

Room Booking Platform

You are expected to design and implement a scalable and fault-tolerant Room Booking Platform.

The design system should support user registration, room search, booking, cancellation, and booking history retrieval. Consider realistic constraints such as rate limiting, multi-region deployment, data consistency, and fault tolerance.

Implementation should support:

user registration, room search, and booking over 2 microservices one for UI and one for BE.

Preferred technologies (not a must): TS, react.

There's no need to implement cancellation, and booking history retrieval.

1. High-Level Architecture

Describe the overall architecture of the system. Include components such as frontend, backend services, databases, cache, load balancers, and any other relevant infrastructure. Consider multi-region deployment and fault tolerance.

2. API Design

List and describe the key API endpoints. Include request and response formats, authentication mechanisms, and rate limiting strategies.

3. Database Schema

Design the schema for storing users, rooms, bookings, and availability. Include relationships, indexing strategies, and data consistency considerations.

4. Concurrency Handling

Explain how you would prevent double-booking and ensure consistency during concurrent booking requests. Discuss locking mechanisms, transactions, or optimistic concurrency control.

5. Scalability Strategies

Describe how you would scale the system to handle high traffic, especially for search and booking operations. Include caching, replication, and load balancing strategies.

6. Optional Components

Discuss additional components such as monitoring, logging, notifications (e.g., booking confirmation emails), and analytics. Explain how these integrate with the core system.