Exercises on Microservices with Spring Boot 3.0

1. Build a User and Order Management System

Problem: Create two microservices:

- **User Service** to manage users.
- **Order Service** to manage orders placed by users.

Requirements:

- Use **REST APIs**.
- Communicate between services using **WebClient (Spring WebFlux)** or **OpenFeign**.
- Store data in **MySQL** or **PostgreSQL**.

---

2. Inventory Management System with Service Discovery

Problem: Create:

- **Product Service**: Manage products and stock.
- **Inventory Service**: Track stock levels for each product.

Requirements:

- Use **Spring Cloud Netflix Eureka** for **service discovery**.
- Implement **centralized configuration** using Spring Cloud Config Server.

---

3. Implement an API Gateway

Problem: Create an **API Gateway** to route requests to:

- **Customer Service**
- **Billing Service**

Requirements:

- Use **Spring Cloud Gateway**.
- Implement **rate limiting**, **caching**, and **path rewriting**.

4. Resilient Microservices with Circuit Breaker

Problem: A **Payment Service** calls a slow third-party API.

Requirements:

- Implement **Circuit Breaker** and **fallback logic** using **Resilience4j**.
- Log and monitor fallback events.