**โจทย์: ระบบจัดการคำสั่งซื้อออนไลน์ (Online Order Management System)**

**รายละเอียด:**

คุณจะต้องสร้าง API ที่รองรับการทำงานของระบบจัดการคำสั่งซื้อออนไลน์ (Order Management System) ซึ่งจะต้องทำงานกับฐานข้อมูล PostgreSQL โดยต้องใช้ Spring boot

**ข้อกำหนด:**

1. **Database Structure**:
   * ตาราง orders สำหรับเก็บข้อมูลคำสั่งซื้อ
   * ตาราง order\_items สำหรับเก็บข้อมูลรายการสินค้าในคำสั่งซื้อ
2. **API Endpoints**:
   * POST /orders: สร้างคำสั่งซื้อใหม่
     + Body: { "customer\_name": "John Doe", "items": [{"product\_name": "Product A", "quantity": 2, "price": 50.0}, {"product\_name": "Product B", "quantity": 1, "price": 100.0}] }
     + Response: { "order\_id": 1, "status": "created" }
   * GET /orders/{order\_id}: ดึงข้อมูลคำสั่งซื้อจาก ID
     + Response: { "order\_id": 1, "customer\_name": "John Doe", "total\_amount": 200.0, "items": [{"product\_name": "Product A", "quantity": 2, "price": 50.0}, {"product\_name": "Product B", "quantity": 1, "price": 100.0}] }
   * GET /orders: ดึงข้อมูลคำสั่งซื้อทั้งหมด (รองรับการ pagination)
     + Query params: ?page=1&size=10
     + Response: { "orders": [{...}, {...}], "total": 100 }
   * PUT /orders/{order\_id}/status: เปลี่ยนสถานะของคำสั่งซื้อ
     + Body: { "status": "completed" }
     + Response: { "order\_id": 1, "status": "completed" }
3. Unit test
   * เขียน Unit test ด้วย junit.jupiter และ MockK หรือ Mockito
4. ทดสอบ API
   * ใช้ swagger ในการทดสอบ API

**ข้อกำหนดเพิ่มเติม:**

* ต้องมีการเชื่อมต่อกับ PostgreSQL โดยใช้ **JPA**
* **สามารถรันบนเครื่องและทดสอบได้จริง**

**คำแนะนำในการทดสอบ:**

* ในการดำเนินการคำสั่งซื้อหลาย ๆ รายการพร้อมกัน แต่ต้องดูแลเรื่องของการควบคุมการเข้าถึงฐานข้อมูลเพื่อป้องกันปัญหาการเข้าถึงพร้อมกัน (concurrent access)
* ใช้ **PostgreSQL** เพื่อเก็บข้อมูลคำสั่งซื้อและรายการสินค้า

sqlCopyCREATE TABLE order\_items (  
    id SERIAL PRIMARY KEY,  
    order\_id INT REFERENCES orders(id) ON DELETE CASCADE,  
    product\_name VARCHAR(100),  
    quantity INT,  
 price DECIMAL(10, 2)  
);

sqlCopyCREATE TABLE orders (  
    id SERIAL PRIMARY KEY,  
    customer\_name VARCHAR(100),  
    total\_amount DECIMAL(10, 2),  
    status VARCHAR(20),  
    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);