Coding Task: Order Fulfillment System

📘 **Problem Statement**

You are developing a backend service for an **Order Fulfillment System** used by an e-commerce platform. The system is responsible for processing customer orders, checking inventory, and notifying the shipping service.

Your goal is to **unit test the service layer** using **Mockito**, focusing on mocking dependencies, verifying method calls, and stubbing return values.

🎯 **Objectives**

- Implement the OrderService class that:
    - Validates inventory before placing an order
    - Saves the order to the repository
    - Notifies the shipping service if the order is successful
- Write **unit tests** using **Mockito** to:
    - Mock the InventoryService, OrderRepository, and ShippingService
    - Stub method responses to simulate different scenarios
    - Verify interactions between components

Components

1. Model

Java

```java
public class Order {

    private String orderId;

    private String productId;

    private int quantity;

    private String status; // e.g., "PLACED", "FAILED"


    // Constructors, Getters, Setters

}
```

Mockito Unit Testing Task

Create a test class OrderServiceTest to:

- **Mock** all dependencies

- **Stub** inventory responses

- **Verify** that:

  - Order is saved only when inventory is available

  - Shipping is scheduled only for successful orders

  - No interactions happen when inventory is insufficient