Submitted by: Prachi Mehta (202318008)

Assignment Report on
Real-Time E-commerce Order Processing System Using Kafka

To develop a Kafka-based system for managing e-commerce orders in real-time, you'll need to set up producers, consumers, and implement message filtering logic. Below are the steps you can follow to achieve this:

Step 1: Set Up Kafka

1. **Install Kafka:** Ensure Kafka is installed and running on your system or a server.
2. **Create Kafka Topics:** Create Kafka topics named **inventory_orders** and **delivery_orders** for each producer to send messages to.

Step 2: Implement Kafka Producers

1. **Inventory Orders Producer (inventory_orders_producer):**
    - This producer should filter messages where the **type** field is **inventory**.
    - Implement a Kafka producer that reads inventory-related events from a data source (like a database or event stream) and sends messages with **type** set to **inventory** to the **inventory_orders** topic.
2. **Delivery Orders Producer (delivery_orders_producer):**
    - This producer should filter messages where the **type** field is **delivery**.
    - Develop a Kafka producer that reads delivery-related events and sends messages with **type** set to **delivery** to the **delivery_orders** topic.

Step 3: Implement Kafka Consumers

1. **Inventory Data Consumer (inventory_data_consumer):**
    - Configure a Kafka consumer that subscribes to the **inventory_orders** topic.
    - Implement logic to process inventory messages received by updating inventory databases or systems accordingly.
2. **Delivery Data Consumer (delivery_data_consumer):**

    - Set up a Kafka consumer for the **delivery_orders** topic.
    - Develop logic to handle delivery-related messages such as scheduling deliveries, updating delivery status, and notifying customers.

Step 4: Develop Message Filtering Logic

1. **Producer Message Filtering:**
    - Implement logic within each producer (**inventory_orders_producer** and **delivery_orders_producer**) to filter messages based on the **type** field from the incoming data source.
    - Only send messages to Kafka if they match the desired **type** (i.e., **inventory** or **delivery**).

Additional Considerations

- **Error Handling:** Implement error handling within producers and consumers to manage exceptions or failed operations gracefully.
- **Scalability:** Design your system to handle increasing loads by considering Kafka partitioning, consumer groups, and scaling strategies.
- **Monitoring and Logging:** Utilize Kafka monitoring tools and logging frameworks to monitor system performance and troubleshoot issues effectively.

By following these steps and best practices, you'll be able to develop a robust Kafka-based e-commerce order management system capable of real-time inventory management and delivery processing.