

**Student ID: 202318047**

**Name: Srushti Bhagchandani**

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

To construct a Kafka-based system for real-time management of e-commerce orders, it's essential to configure producers and consumers, along with integrating message filtering mechanisms. Here are the structured steps to guide you through this setup:

#### **Step 1: Initialize Kafka**

1. Kafka Installation: Confirm that Kafka is properly installed and operational either on your local machine or a remote server.
2. Topic Creation: Establish Kafka topics named `inventory_orders` and `delivery_orders` to cater to specific producers.

#### **Step 2: Configure Kafka Producers**

1. Producer for Inventory Orders (`inventory_orders_producer`):  
Filter and process only those messages that have the type attribute set to inventory.  
Set up a Kafka producer to capture inventory-related events from sources such as databases or live event streams, and publish these messages to the `inventory_orders` topic.
2. Producer for Delivery Orders (`delivery_orders_producer`):  
Ensure this producer filters and handles messages where the type field is delivery.  
Create a Kafka producer that gathers delivery-related events and dispatches messages marked as delivery to the `delivery_orders` topic.

#### **Step 3: Set Up Kafka Consumers**

1. Consumer for Inventory Data (`inventory_data_consumer`):  
Configure a Kafka consumer to listen to the `inventory_orders` topic.  
Craft logic to process incoming inventory messages by updating relevant inventory databases or systems.
2. Consumer for Delivery Data (`delivery_data_consumer`):  
Initialize a Kafka consumer dedicated to the `delivery_orders` topic.  
Formulate logic to manage delivery-related messages, which includes scheduling deliveries, updating delivery statuses, and notifying customers.

#### **Step 4: Implement Message Filtering Logic**

1. Filtering within Producers:  
Integrate message filtering logic in each producer (`inventory_orders_producer` and `delivery_orders_producer`) to selectively send messages based on the type attribute from the data source.
2. Messages should only be dispatched to Kafka topics if they correspond to the predefined types (inventory or delivery).