

Assignment

Students will build a Kafka-based system that produces and consumes **order messages**. Each message must use **Avro serialization**, and the system must support:

- Real-time aggregation (running average of prices)
- Retry logic for temporary failures
- Dead Letter Queue (DLQ) for permanently failed messages
- Demonstrates the system **live** and maintains a **Git repository** for submission.
- Students are free to **choose any programming language** for the producer and consumer and are expected to **research and implement solutions independently**.

Order Message Definition

Each order message represents a purchase transaction with the following schema (`order.avsc`):

Field	Type	Description
<code>orderId</code>	string	Unique identifier for the order (e.g., "1001", "1002")
<code>product</code>	string	Name of the purchased item (e.g., "Item1", "Item2")
<code>price</code>	float	Price of the product (randomized for the assignment)