NestJS Practical

1. **Create a MSSQL database with the below tables**
    - Balances Table:
    - Fields:
        - Id (Primary Key)
        - user_id
        - currency_symbol (e.g., BTC, ETH)
        - balance (amount of the cryptocurrency)
    - Orders Table:
    - Fields:
        - Id (Primary Key)
        - user_id
        - order_type (buy/sell)
        - currency_symbol (the cryptocurrency being traded)
        - price (the price at which the order is placed)
        - quantity (the quantity of cryptocurrency being traded)
        - status (open/closed/cancelled)

- Create an API that will place an order with userId (ranging from 1 to 10), Order_type (buy/sell), currency_symbol (BTC, ETH), Price (ranging from 1.123456 to 9.123456), and Quantity. The initial data should be added to the order table and also produced on Confluent Kafka. Another script should read data from Confluent Kafka and, if the order is bought at the same price, then add the quantity. For sell orders, deduct the balance. And add or update balance in the "Balance" Table.

Guidelines:

- Use Nest.js with typescript.
- Use Confluent Kafka
- Focus on code quality, readability, and maintainability.
- Write clean and well-commented code.
- Ensure the application is easy to set up and run locally.

Submission:

- Host your project on GitHub.
- Include a brief README.md file to provide clear instructions on how to run the project locally.

Create a video that shows data inserted into the "Orders" Table and balance is added or updated into the "Balance" Table.