**Main Objective**

Build a real-time streaming fraud detection system using Apache Kafka and Python that processes transaction data and identifies potentially fraudulent transactions.

The assignment teaches you practical Kafka concepts including producers, consumers, topics, and stream processing while building a realistic fraud detection pipeline that could be found in banking systems.

**Key Tasks to Complete**

**1. Infrastructure Setup**

* Set up a local Kafka cluster using Docker Compose
* Create separate Docker Compose files:
  + docker-compose.kafka.yml for the Kafka cluster (Zookeeper + Broker)
  + docker-compose.yml for your applications
* Create a shared Docker network (kafka-network)

**2. Transaction Generator Application**

**Location:** ./generator/ folder

* **Files to create:**
  + Dockerfile (Python 3.6 base image)
  + requirements.txt (kafka-python dependency)
  + app.py (main producer application)
  + transactions.py (utility functions for generating fake transactions)
* **Functionality:**
  + Generate random transactions with: source account, target account, amount, currency
  + Use Kafka Producer to send transactions to queueing.transactions topic
  + Make it configurable via environment variables
  + Run continuously (infinite loop with configurable rate)

**3. Fraud Detector Application**

**Location:** ./detector/ folder

* **Files to create:**
  + Dockerfile (same as generator)
  + requirements.txt (kafka-python dependency)
  + app.py (main consumer/producer application)
* **Functionality:**
  + Consume transactions from queueing.transactions topic
  + Apply fraud detection logic (transactions ≥ $900 are suspicious)
  + Route transactions to appropriate topics:
    - Legitimate: streaming.transactions.legit
    - Fraudulent: streaming.transactions.fraud
  + Use both Kafka Consumer and Producer

**4. Testing & Verification**

* Start the Kafka cluster
* Run both applications using Docker Compose
* Verify message flow using kafka-console-consumer commands
* Test that transactions are properly categorized into legit/fraud topics

**5. Documentation & Deliverables**

* Push all code to your GitHub repository
* Create comprehensive README.md with:
  + Setup instructions
  + Screenshots of code execution
  + Screenshots of console outputs showing transactions flowing through topics
  + Verification that the fraud detection is working correctly

**Expected Project Structure**

.

├── docker-compose.yml

├── docker-compose.kafka.yml

├── detector/

│ ├── Dockerfile

│ ├── app.py

│ └── requirements.txt

└── generator/

├── Dockerfile

├── app.py

├── transactions.py

└── requirements.txt