```
FRAUD_DETECTION/
|
├── 1_data_ingestion/
|   ├── producers/
|   |   ├── producer.py
|   |   ├── config.yaml
|   |   └── requirements.txt
|   ├── spark_jobs/
|   |   ├── clean_stream.py
|   |   ├── kafka_test.py
|   |   ├── stream_test_console.py
|   |   └── venv/
|   └── schemas/
|
├── 2_stream_processing/
|   ├── model_training/
|   |   ├── train_lightgbm.ipynb
|   |   ├── lightgbm_model.pkl
|   |
|   ├── spark_jobs/
|   |   └── predict_step1.py
|   |
|   ├── dashboard/
|   |   └── dashboard.py
|   |
|   ├── alert_service/
|   |   └── email_alert_service.py
|   |
|   └── venv/
|
```

├── 3_batch_processing/

│ (To be implemented next)

│

├── airflow/

│ (To be implemented next)

│

├── infrastructure/

│ └── docker/

│ ├── docker-compose.ingestion.yml

│ ├── docker-compose.batch.yml

│ └── spark-docker/

│

└── data/

└── fraud_data/

├── all_transactions.csv

└── fraud_transactions.csv

✅ **UPDATED WORK-DONE REPORT (Corrected Data Flow)**

Real-Time Fraud Detection System

📁 **FRAUD_DETECTION PROJECT — WORK COMPLETED**

1️⃣ **1_data_ingestion/**

Handles **real-time ingestion → cleaning → S3 storage**.

📌 **producers/**

✓ **producer.py**

- Reads the Kaggle credit card fraud CSV.
- Streams transactions row-by-row into Kafka topic: **creditcard-transactions**
- Supports **slowing down stream** via rows_per_second.

- Injects **synthetic fraud events** every 20 transactions.
- Uses Confluent Kafka Producer.

✓ config.yaml

- Contains streaming rate, topic name, CSV path, Kafka broker config.

✓ requirements.txt

Dependencies for producer.

📌 spark_jobs/

✓ clean_stream.py (VERY IMPORTANT — your pipeline heart)

This performs **real-time Spark Structured Streaming**:

It takes raw data from:

- creditcard-transactions (Producer output)

It produces cleaned data to:

OUTPUT 1 → Kafka topic: cleaned-transactions

Used by real-time ML predictor (predict_step1.py)
This is your **real-time path**.

OUTPUT 2 → S3 bucket (Parquet files)

This is your **batch layer** input for Snowflake.

⭐ CORRECT FLOW YOU DESCRIBED:

Producer → creditcard-transactions → clean_stream.py →

(a) cleaned-transactions → Stream Processing (ML prediction)

(b) S3 → Batch Layer (Snowflake)

This cleaned_stream is doing **dual writes**, which is exactly what enterprise pipelines do.

✓ kafka_test.py + stream_test_console.py

Used to debug Kafka and Spark stream handling.

✓ schemas/

Stores schemas for Spark/Kafka messages.

2️⃣ 2_stream_processing/

Consumes cleaned stream from Kafka and performs **real-time inference, dashboard visualization, and fraud alerting**.

📌 model_training/

✓ train_lightgbm.ipynb

Trained LightGBM model on Kaggle fraud dataset.

✓ lightgbm_model.pkl

Saved model + feature list.

📌 spark_jobs/predict_step1.py

✓ Consumes cleaned data:

Topic → **cleaned-transactions**
(comes from clean_stream.py)

✓ The predictor performs:

- Feature extraction
- LightGBM prediction
- Fraud probability calculation

✓ Output:

- Sends real-time predictions to Kafka topic:
  predicted-transactions
  (used by Dashboard)
- Sends fraud alerts to:
  fraud-alerts
- Writes CSV logs into:
- data/fraud_data/all_transactions.csv
- data/fraud_data/fraud_transactions.csv

✓ Sends email alert for fraud

This is your **real-time ML engine**.

📌 dashboard/dashboard.py

Streamlit-based real-time enterprise dashboard showing:

- Fraud Count
- Safe Count
- Total Transactions
- Fraud %
- Live transaction table
- Live charts with fraud markers

Consumes:
➞ Kafka topic **predicted-transactions**

📌 alert_service/email_alert_service.py

Consumes fraud alerts from:
➞ Kafka topic **fraud-alerts**

Sends HTML email using Gmail SMTP.

3️⃣ 3_batch_processing/

(Not implemented yet)

Will handle:

- Loading S3 data → Snowflake (Snowpipe)
- Daily/Hourly analytics (SQL)
- Aggregated fraud metrics
- Historical dashboards

4️⃣ airflow/

(Not implemented yet)

Will orchestrate:

- Producer starting
- Spark jobs
- ML predictor
- Dashboard refresh
- Snowflake batch pipelines

This becomes your **control plane**.

5️⃣ infrastructure/docker/

Contains Docker environments:

- Kafka + Zookeeper for ingestion
- Spark cluster (optional)
- Batch processing setup

6️⃣ data/fraud_data/

Stores predicted output from ML pipeline:

- all_transactions.csv
- fraud_transactions.csv

These will be used for:

- Batch analytics
- Manual inspection
- Loading into Snowflake

🎯 CURRENT PIPELINE STATUS (Realistic Representation)

RAW TRANSACTIONS

↓

Kafka Topic: creditcard-transactions

↓

clean_stream.py (Spark)

↓ ↘

cleaned-transactions S3 Parquet (Batch)

↓

predict_step1.py (ML)

↓ ↓

predicted-transactions fraud_data CSVs

↓ ↓

Dashboard Email Alerts

🚀 **NEXT STEPS (Your Remaining Work)**

✓ A. Batch Processing Layer (3_batch_processing)

You will build:

1. Snowflake tables
2. Snowpipe for auto-loading from S3
3. Batch SQL analytics:
    - Daily fraud summary
    - Amount statistics
    - High-risk transaction patterns
4. Optional: BI Dashboard (Superset, Power BI)

✓ B. Airflow Orchestration

Create DAGs that run:

1. Start producer
2. Start clean_stream.py
3. Start predictor
4. Trigger email service
5. Batch job to move S3 → Snowflake
6. Daily reporting job

This will make your project feel like **production-grade**.

🌟 **You have completed ~70% of a real enterprise fraud detection pipeline.**

This is **extremely strong** for a fresher portfolio.