Project Execution Environment Manual
Project Title: Real-Time Fraud Detection System
Team: A. D. Thushani Niwarthana, A. D. Wishani Samadara, Lubabatu Hussaini
Supervisor: Mr. Rameez Raja

🔧 1. System Requirements

Hardware:

- CPU: Intel i5 or higher
- RAM: 8 GB minimum
- Disk: 10 GB free space

Operating System:

- Windows 10/11 OR Ubuntu 20.04+

Software Stack:

- Python 3.9+
- Apache Kafka 3.x
- MongoDB (or PostgreSQL)
- Dash/Flask
- pip (Python package installer)

📦 2. Python Dependencies

Install required libraries:

pip install pandas numpy scikit-learn xgboost lightgbm catboost kafka-python dash flask pymongo imbalanced-learn matplotlib seaborn

Optional for PostgreSQL users:

pip install psycopg2-binary

📂 3. Project Folder Structure

```
FRAUD DETECTION PROJECT/
├── __pycache__/
├── assets/
│   └── style.css
├── Model Reports/
├── venv/
├── consumer.log
├── consumer.py
├── counts.log
├── dashboard.py
├── evaluation.py
├── features.py
├── fraud_model.py
├── m.txt
├── model.txt
├── preprocessed_data.pkl
├── producer.log
├── producer.py
├── Start Kafka.txt
└── transactions.csv
```

🪄 4. Step-by-Step Execution

Step 1: Train and Save ML Model

python model.py

- This cleans data, applies SMOTE, trains the models, and saves the best one.

Step 2: Start Apache Kafka Services

- Run Kafka Server and Zookeeper:

```
bin/zookeeper-server-start.sh config/zookeeper.properties
bin/kafka-server-start.sh config/server.properties
```

Step 3: Start Kafka Producer

python producer.py

- This begins sending transaction data into the Kafka stream.

Step 4: Start Kafka Consumer (Real-Time Prediction)

python consumer.py

- This listens to the Kafka stream, runs the fraud model, and stores predictions.

Step 5: Launch the Dashboard

python dashboard.py

- Open browser at http://127.0.0.1:8050 to monitor transactions live.

🧪 5. Testing the System

- Observe console logs from consumer.py for "Fraud" or "Not Fraud" predictions.
- Dashboard updates in real-time with flagged transactions.
- Modify producer to simulate fraud spikes if needed.

🗂️ 6. Database Options

- **MongoDB:** Stores flagged transaction logs in real time.
- **PostgreSQL (Optional):** Can be used to store structured data and reporting tables.

🚨 7. Troubleshooting Tips

- **Kafka Errors:** Ensure Zookeeper and Kafka services are running.
- **ModuleNotFound:** Reinstall missing Python packages.
- **Dashboard not loading:** Check if port 8050 is already in use.
- **Fraud predictions not showing:** Recheck model.pkl or feature alignment.

✅ 8. Final Notes

- All scripts are compatible with local development.
