Fraud Transaction Detection – Project Report

1. Introduction

The Fraud Transaction Detection System is a machine-learning project designed to identify whether a financial transaction is **fraudulent or legitimate**. The goal was to build a complete end-to-end solution—from data processing to model training and deployment through a web interface (Streamlit). This project simulates a real-world fraud-analytics workflow and demonstrates my ability to work with large datasets, extract features, build models, and deploy interactive ML apps.

2. Dataset Overview

The dataset provided consisted of **daily transaction logs** stored as .pkl files, totaling about **1.75 million** records from April to September.
Each transaction contained:

- Transaction ID
- Customer ID
- Terminal ID
- Transaction Date/Time
- Transaction Amount
- Fraud label (TX_FRAUD)

The dataset was **synthetically generated**, meaning fraud labels were created using predefined rules rather than real fraud events.

3. Fraud Simulation Logic

While exploring the data, I found that fraud cases followed specific simulated rules.
One of the most important rules was:

🔎 Rule: Any transaction with an amount greater than 220 is considered fraud.

This means high-value transactions (≥220) were automatically marked as fraudulent during dataset creation.
Because the model learns patterns directly from the data, it naturally learns this rule as well.
As a result, when entering values above ~200 in the app, the model correctly identifies them as high-risk transactions.

Other hidden fraud patterns included:

- Certain terminals being flagged as fraudulent for 28 days
- Some customers having amounts multiplied artificially

These rules helped the model learn both simple and complex fraud behaviors.

4. Methodology

4.1 Data Processing

- Loaded and combined all .pkl files into a single DataFrame
- Extracted useful time features such as:
    - **Transaction day (TX_DAY)**
    - **Transaction hour (TX_HOUR)**
- Selected relevant features for model training:
    - TX_AMOUNT, TX_DAY, TX_HOUR

4.2 Model Training

- Algorithm used: **RandomForestClassifier**
- Trained on 80% of data, tested on 20%
- Performance was strong due to clear patterns in the labeled dataset

4.3 Deployment

A simple Streamlit application (app.py) allows users to:

- Input transaction amount
- Enter transaction date/time
- Get an immediate fraud prediction with a risk score

5. Results

The model achieved **high accuracy**, primarily because the dataset contains strong and clear fraud patterns (especially the amount >220 rule).
The Streamlit app successfully predicts fraud based on learned characteristics and provides an easy-to-use interface for testing different scenarios.

6. Conclusion

This project helped me understand key concepts in fraud analytics, including:

- Working with large, multi-file datasets
- Extracting time-based features
- Training and evaluating ML models
- Deploying interactive ML apps using Streamlit

The system performs well for the dataset provided, and with additional behavioral features (customer history, terminal statistics), it could be extended to real-world fraud detection tasks.