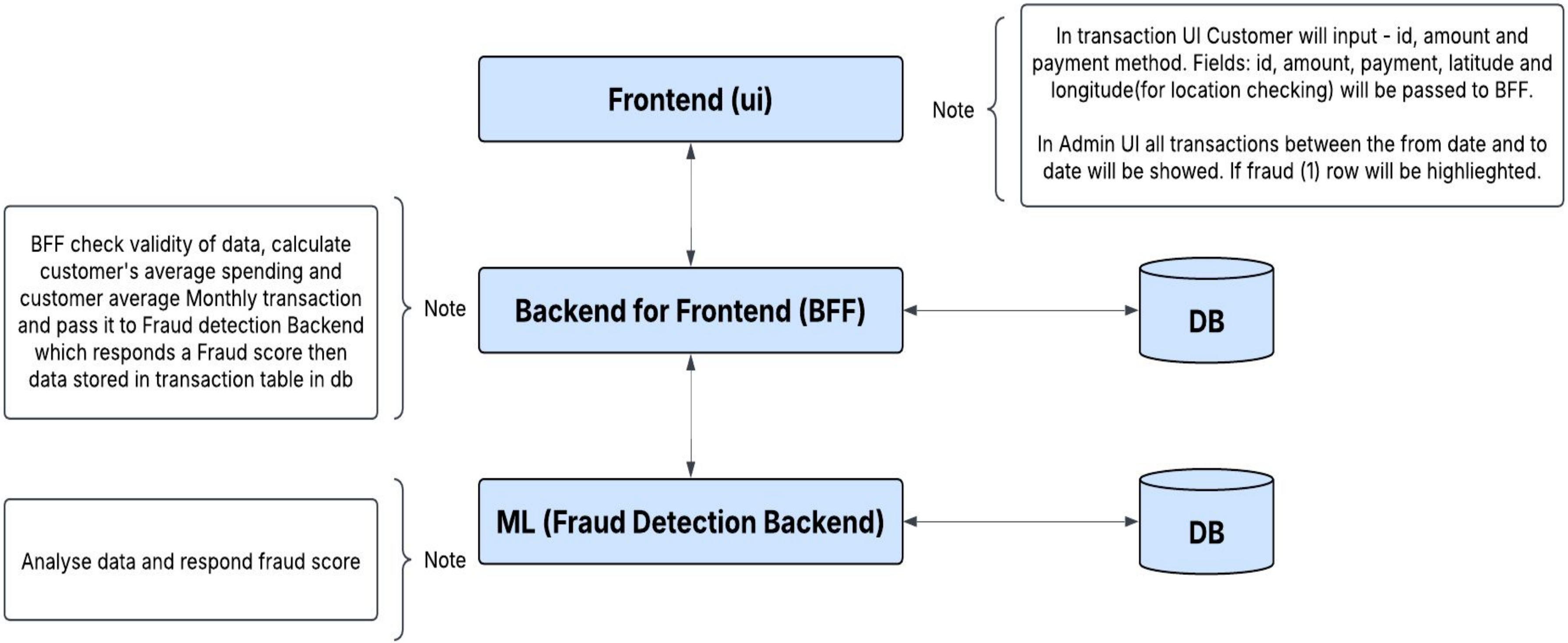
Bank Transaction Fraud Detection Solution

Team Cobra OS:
Krisha Shah- Team lead | Phone: 9727470882 | email: krishabr2007@gmail.com

Krina Shah | Phone: 9726470882 | email: krinabr2007@gmail.com

Anshul patel | Phone: 8160445737 | email: anshul2266patel@gmail.com

ARCHITECHTURE



Frontend (ui)

Note

In transaction UI Customer will input - id, amount and payment method. Fields: id, amount, payment, latitude and longitude(for location checking) will be passed to BFF.

In Admin UI all transactions between the from date and to date will be showed. If fraud (1) row will be highlieghted.

BFF check validity of data, calculate customer's average spending and customer average Monthly transaction and pass it to Fraud detection Backend which responds a Fraud score then data stored in transaction table in db

Note

Backend for Frontend (BFF)

DB

Analyse data and respond fraud score

Note

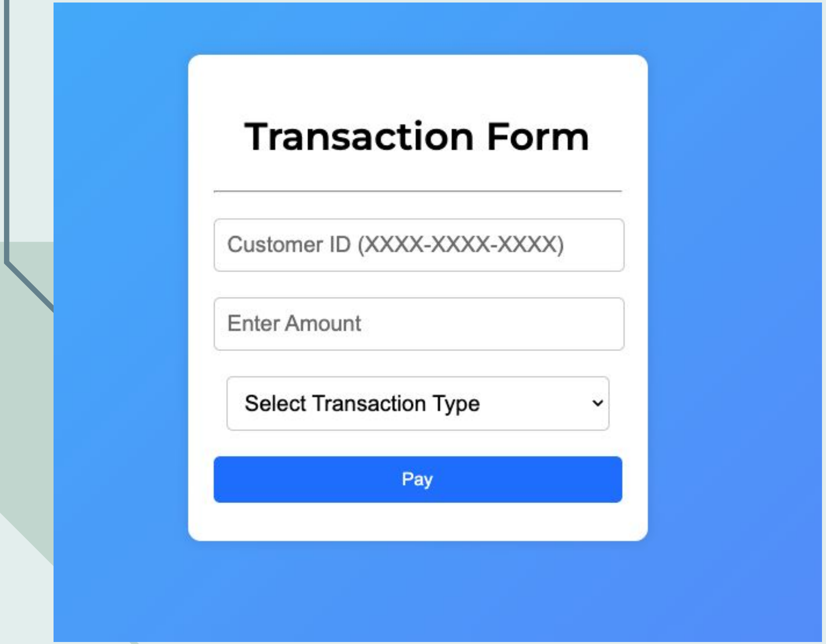
ML (Fraud Detection Backend)

DB

UI

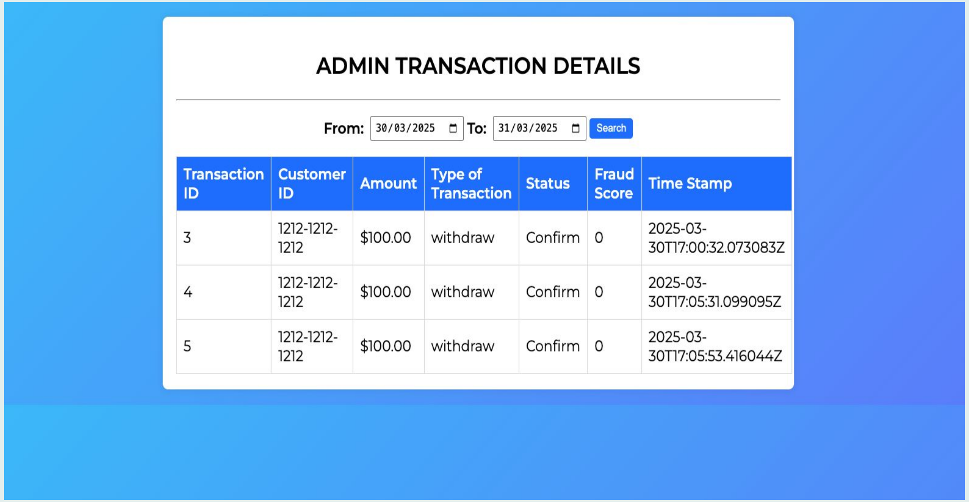
Technologies: HTML, CSS, Angular JS

Screen 1: Transaction/User UI



Transaction Form

Screen 2: Admin UI



ADMIN TRANSACTION DETAILS

From: 30/03/2025 To: 31/03/2025

Transaction ID	Customer ID	Amount	Type of Transaction	Status	Fraud Score	Time Stamp
3	1212-1212-1212	$100.00	withdraw	Confirm	0	2025-03-30T17:00:32.073083Z
4	1212-1212-1212	$100.00	withdraw	Confirm	0	2025-03-30T17:05:31.099095Z
5	1212-1212-1212	$100.00	withdraw	Confirm	0	2025-03-30T17:05:53.416044Z

BFF

- Database Schema:
  - Transaction Table:
    - Id (transaction id), Customer Id, Amount, Time Stamp, Transaction Type, IP_address, Transaction Status, Fraud score, Latitude
      Longitude
  - Customer Table:
    - Customer Id, Customer Name, Phone, Email, Balance
- Technologies: django-rest framework
- End points:
  - Transaction endpoint
    - Request payload (customer id, amount, transaction type, latitude, longitude)
    - Request payload to AI backend (customer id, amount, transaction type, latitude, longitude, IP_address, Time stamp, Customer Average Spending, Customer’s Monthly Average Transactions)
    - Response payload from AI backend (fraud score)
    - Response payload (Status:200)

AI BACKEND

- Use of traditional Single ML model approach, we used the Random Forest Classifier after evaluating other models such as SVM, XGBoost performance, KNN, ANN, NULL, Logistic Regression.

- Dataset (csv file for ml model):
  - (transaction id, customer id, amount, transaction type, latitude, longitude, transaction timestamp, hour of day, day of week, ip address, average spending, customer monthly average transaction, Fraud score

Rules and Guidelines for Model

- Transaction Amount & Frequency Rules
  - Unusually High Transaction Amounts: Transactions significantly above the customer’s average spending limit are flagged.

    Sudden Increase in Transaction Count: A sharp spike in the number of daily/monthly transactions compared to the customer’s history raises suspicion.
- Behavioral Pattern Rules
  - Deviations from Customer’s Normal Behavior: If a customer suddenly transacts at odd hours or from an unusual location, it is flagged.

    Changes in Transaction Type Usage: If a customer primarily withdraws cash but suddenly starts making high-value online transfers, it raises a red flag.
- Time-Based Rules
  - Unusual Transaction Timing: Transactions at odd hours (e.g., midnight or early morning) that do not match past behavior may indicate fraud.
- Rule-Based & AI-Driven Scoring
  - Predefined Fraud Patterns: The model applies industry-standard fraud rules (e.g., card-not-present fraud, ATM skimming).

    Risk Scoring: The model assigns a fraud risk score based on multiple factors, such as transaction type, location, device, and frequency.

Technologies Used

- **Frontend:** HTML, CSS, AngularJS **(User submits transactions, admin monitors fraud.)**

- **Backend (API & Business Logic):** Django (Django REST Framework) **(Handles user transactions, connects to AI model.)**

- **Database:** MySQL **(Stores transactions, fraud history.)**

- **Mode Selection:** Pandas , Numpy, Scikit-learn, graphviz, seaborn, TensorFlow, XGBoost, SMOOTE

- **Training Model:** Python, Pandas, Numpy, Scikit-learn, SciPy **(Detects fraudulent transactions.)**

THANK YOU!!!