🛡 ONLINE PAYMENTS FRAUD DETECTION USING MACHINE LEARNING

PROJECT REPORT

INTRODUCTION

1.1 Project Overview

The Online Payments Fraud Detection System is a Machine Learning–based application developed to detect fraudulent financial transactions in real time. With the rapid growth of digital payments, fraudulent activities such as unauthorized transfers and cash-out scams are increasing significantly.

This project uses supervised learning algorithms to classify transactions as **Fraud** or **Not Fraud** based on transaction features such as transaction type, amount, sender balance, and receiver balance.

The final selected model is integrated into a Flask-based web application that allows users to input transaction details and receive instant fraud prediction.

1.2 Purpose

The purpose of this project is to build an intelligent fraud detection system that reduces financial risk and improves transaction security.

The system aims to:
• Detect fraudulent transactions accurately
• Handle imbalanced datasets effectively
• Compare multiple ML algorithms
• Deploy the best model into a working web application

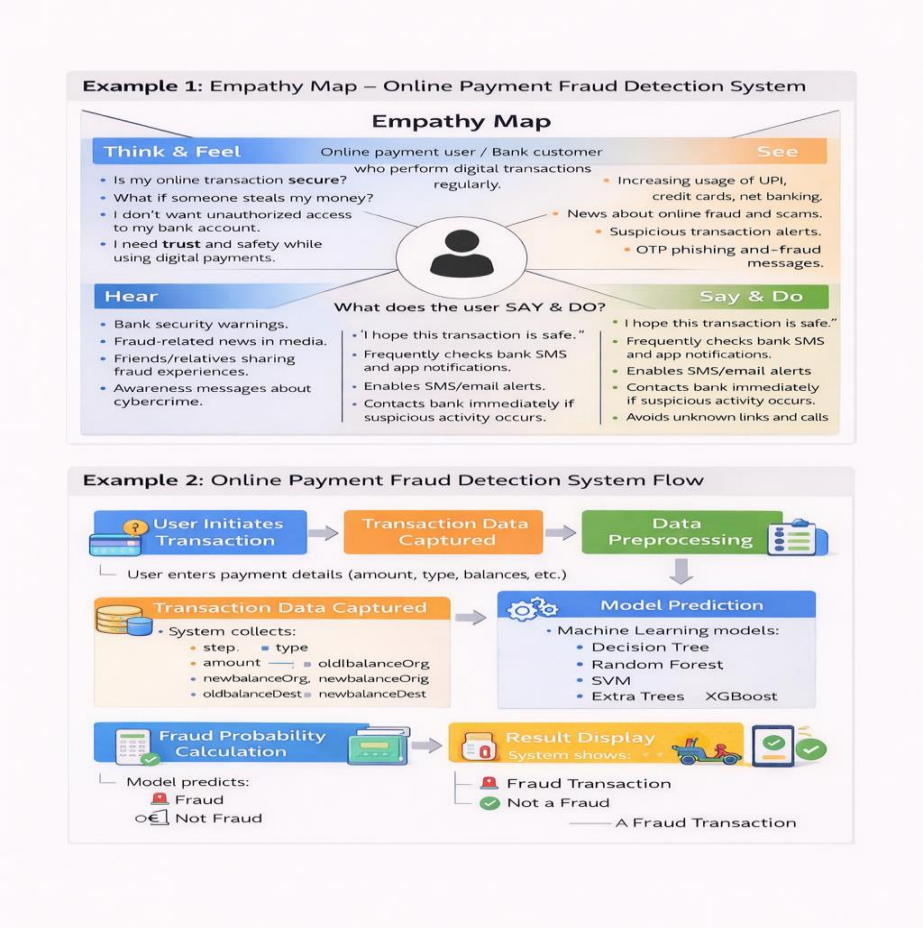
2. IDEATION PHASE

2.1 Problem Statement

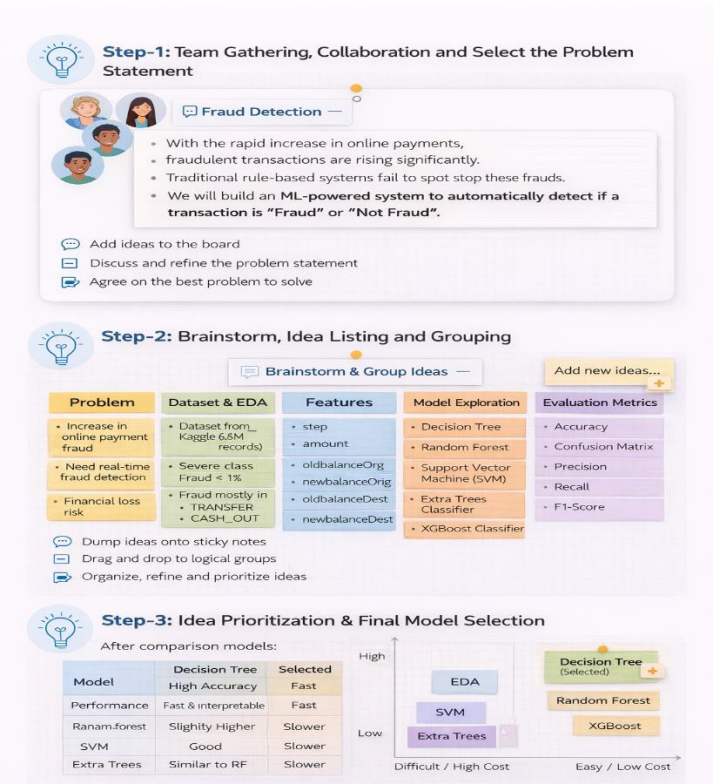
Online payment platforms face severe financial losses due to fraudulent transactions. Traditional rule-based systems fail to detect complex fraud patterns.

Therefore, this project focuses on building a Machine Learning–based fraud detection system capable of predicting fraud transactions efficiently and accurately.

2.2 Empathy Map Canvas



2.3 Brainstorming



During brainstorming, the following ideas were identified:

- Use Kaggle dataset (~6.3M records)
- Handle severe class imbalance
- Apply Decision Tree, Random Forest, SVM, XGBoost
- Compare models using evaluation metrics
- Deploy using Flask

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

3.2 Solution Requirements

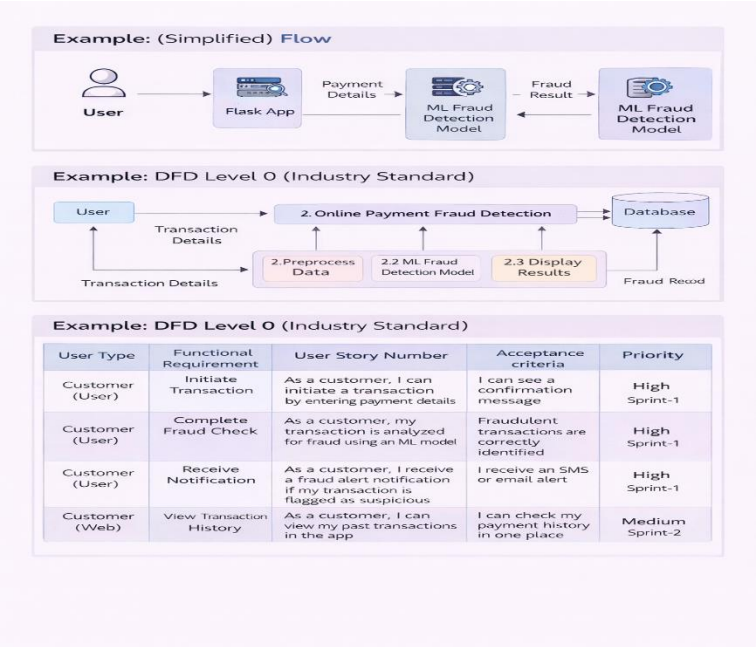
Functional Requirements

• Accept transaction details from user
• Preprocess transaction data
• Predict fraud probability
• Display result in UI

Non-Functional Requirements

• Fast response time
• Secure processing
• Accurate classification
• Easy-to-use interface

3.3 Data Flow Diagram



User Type	Functional Requirement	User Story Number	Acceptance criteria	Priority
Customer (User)	Initiate Transaction	As a customer, I can initiate a transaction by entering payment details	I can see a confirmation message	High Sprint-1
Customer (User)	Complete Fraud Check	As a customer, my transaction is analyzed for fraud using an ML model	Fraudulent transactions are correctly identified	High Sprint-1
Customer (User)	Receive Notification	As a customer, I receive a fraud alert notification if my transaction is flagged as suspicious	I receive an SMS or email alert	High Sprint-1
Customer (Web)	View Transaction History	As a customer, I can view my past transactions in the app	I can check my payment history in one place	Medium Sprint-2

3.4 Technology Stack

• Python
• Pandas
• NumPy
• Scikit-Learn
• XGBoost
• Matplotlib
• Seaborn
• Flask
• HTML/CSS

4. PROJECT DESIGN

4.1 Problem–Solution Fit

Problem – Solution Fit Template:

1 Customer Segments
- Online payment users (UPI, banking app, wallet users)
- Banks & Financial Institutions
- Digital payment platforms
- Fraud monitoring teams

2 Customer Problems
- Rising online fraud transactions
- Fear of unauthorized money transfer
- Delay in detecting suspicious transactions
- Financial loss and lack of trust in digital payments

3 Customer Constraints
- Traditional rule-based systems fail to detect new fraud patterns.
- Huge volume of transactions makes manual checking impossible
- Severe class imbalance (Fraud < 1%)
- Need for real-time detection without slowing transactions.

4 Available Solutions (Existing)
- Manual verification
- Rule-based fraud detection systems
- OTP-based confirmation systems
- Transaction limits (But these are not fully efficient.)

4 Available Solutions (Existing)
- Manual verification
- Rule-based fraud detection systems
- OTP-based confirmation systems
- Transaction limits
- (But these are not fully efficient.)

5 Triggers
- Sudden large transaction
- Transaction from unusual account pattern
- Transfer or CASH_OUT type transactions
- Rapid balance changes

6 Jobs-To-Be-Done / User Needs
- Detect fraud instantly
- Prevent financial loss
- Maintain smooth & fast transactions
- Increase customer trust

7 Our Proposed Solution
- Machine Learning-based Fraud Detection System
- Usec Kaggle Online Payments dataset (6M+ records)
- Models tested:
  - Decision Tree
  - Random Forest
  - Extra
  - SVM
  - XGBoost
- Final selected model: **Decision Tree**

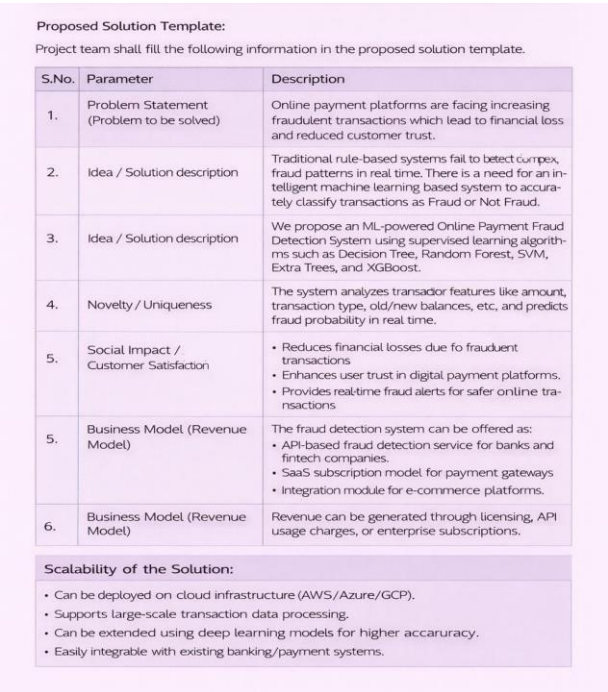
8 Channels of Behaviour
- Web-based prediction system
- Real-time transaction input form
- Model-based fraud probability prediction

9 Emotions Before / After
- Before: Fear, Frustration
- After: Trust, Security, Confidence in online payments

References:

1. https://www.ideahackers.network/problem-solution-fit-canvas/
2. https://medium.com/@ap.cantus/problem-solution-fit-canvas-aa9ad55cb-ffe

4.2 Proposed Solution



Proposed Solution Template:

Project team shall fill the following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Online payment platforms are facing increasing fraudulent transactions which lead to financial loss and reduced customer trust.
2.	Idea / Solution description	Traditional rule-based systems fail to betect complex, fraud patterns in real time. There is a need for an intelligent machine learning based system to accurately classify transactions as Fraud or Not Fraud.
3.	Idea / Solution description	We propose an ML-powered Online Payment Fraud Detection System using supervised learning algorithms such as Decision Tree, Random Forest, SVM, Extra Trees, and XGBoost.
4.	Novelty / Uniqueness	The system analyzes transador features like amount, transaction type, old/new balances, etc, and predicts fraud probability in real time.
5.	Social Impact / Customer Satisfaction	• Reduces financial losses due fo frauduent transactions • Enhances user trust in digital payment platforms. • Provides real-time fraud alerts for safer online transactions
5.	Business Model (Revenue Model)	The fraud detection system can be offered as: • API-based fraud detection service for banks and fintech companies. • SaaS subscription model for payment gateways • Integration module for e-commerce platforms.
6.	Business Model (Revenue Model)	Revenue can be generated through licensing, API usage charges, or enterprise subscriptions.

Scalability of the Solution:
- Can be deployed on cloud infrastructure (AWS/Azure/GCP).
- Supports large-scale transaction data processing.
- Can be extended using deep learning models for higher accaruracy.
- Easily integrable with existing banking/payment systems.

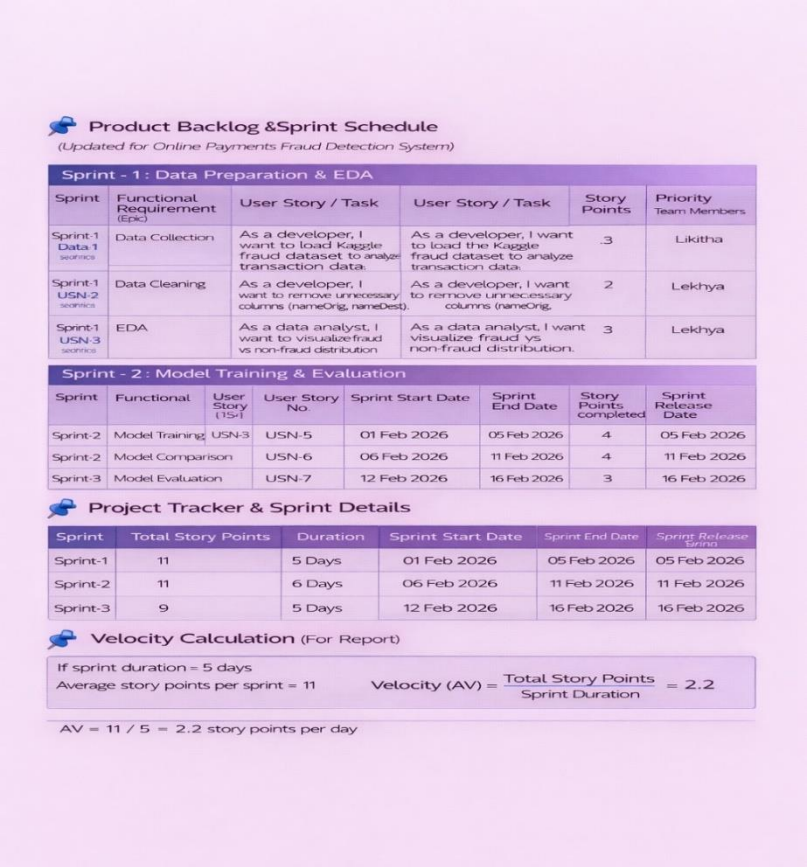
The proposed system uses Machine Learning algorithms to classify transactions. The model is trained on historical transaction data and deployed through a Flask web application for real-time prediction.

4.3 Solution Architecture

The solution architecture connects business requirements with technical implementation.

Transaction data flows through preprocessing steps, feature encoding, and classification model prediction. The selected Decision Tree model processes encoded features and outputs fraud probability, which is displayed to the user via Flask interface.

5. PROJECT PLANNING & SCHEDULING



Product Backlog &Sprint Schedule

(Updated for Online Payments Fraud Detection System)

Sprint - 1: Data Preparation & EDA

Sprint	Functional Requirement (Epic)	User Story / Task	User Story / Task	Story Points	Priority Team Members
Sprint-1 Data 1 seonics	Data Collection	As a developer, I want to load Kaggle fraud dataset to analyze transaction data.	As a developer, I want to load the Kaggle fraud dataset to analyze transaction data.	3	Likitha
Sprint-1 USN-2 seonics	Data Cleaning	As a developer, I want to remove unnecessary columns (nameOrig, nameDest).	As a developer, I want to remove unnecessary columns (nameOrig,	2	Lekhya
Sprint-1 USN-3 seonics	EDA	As a data analyst, I want to visualize fraud vs non-fraud distribution	As a data analyst, I want visualize fraud vs non-fraud distribution.	3	Lekhya

Sprint - 2 : Model Training & Evaluation

Sprint	Functional	User Story (15-1	User Story No.	Sprint Start Date	Sprint End Date	Story Points completed	Sprint Release Date
Sprint-2	Model Training	USN-3	USN-5	01 Feb 2026	05 Feb 2026	4	05 Feb 2026
Sprint-2	Model Comparison		USN-6	06 Feb 2026	11 Feb 2026	4	11 Feb 2026
Sprint-3	Model Evaluation		USN-7	12 Feb 2026	16 Feb 2026	3	16 Feb 2026

Project Tracker & Sprint Details

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date	Sprint Release tarina
Sprint-1	11	5 Days	01 Feb 2026	05 Feb 2026	05 Feb 2026
Sprint-2	11	6 Days	06 Feb 2026	11 Feb 2026	11 Feb 2026
Sprint-3	9	5 Days	12 Feb 2026	16 Feb 2026	16 Feb 2026

Velocity Calculation (For Report)

If sprint duration = 5 days
Average story points per sprint = 11

$$\text{Velocity (AV)} = \frac{\text{Total Story Points}}{\text{Sprint Duration}} = 2.2$$

AV = 11 / 5 = 2.2 story points per day

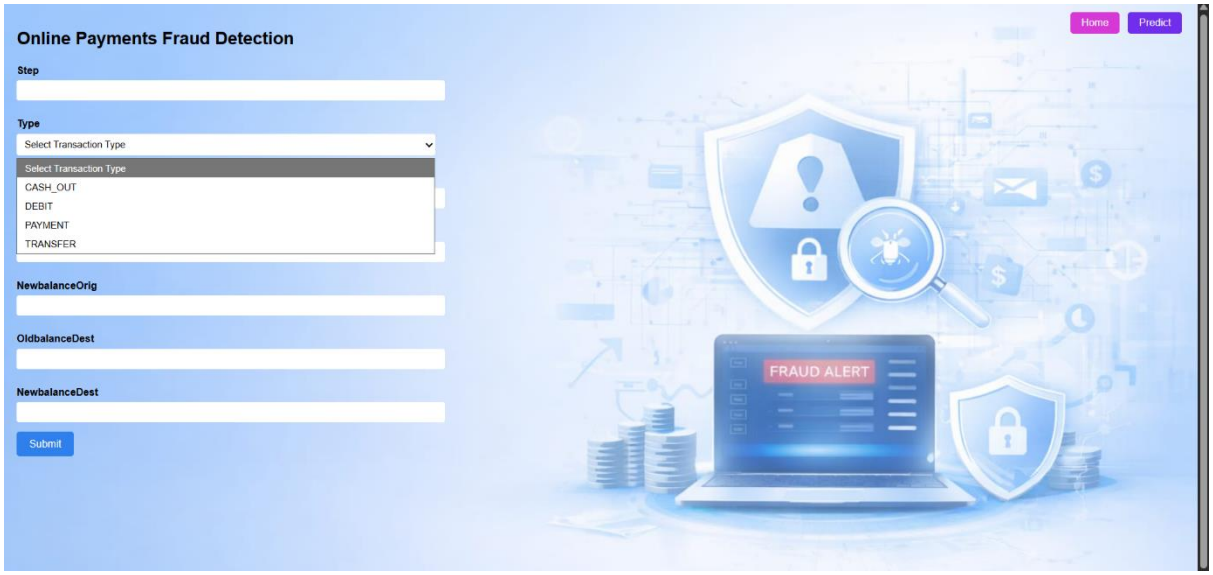
6. FUNCTIONAL AND PERFORMANCE TESTING

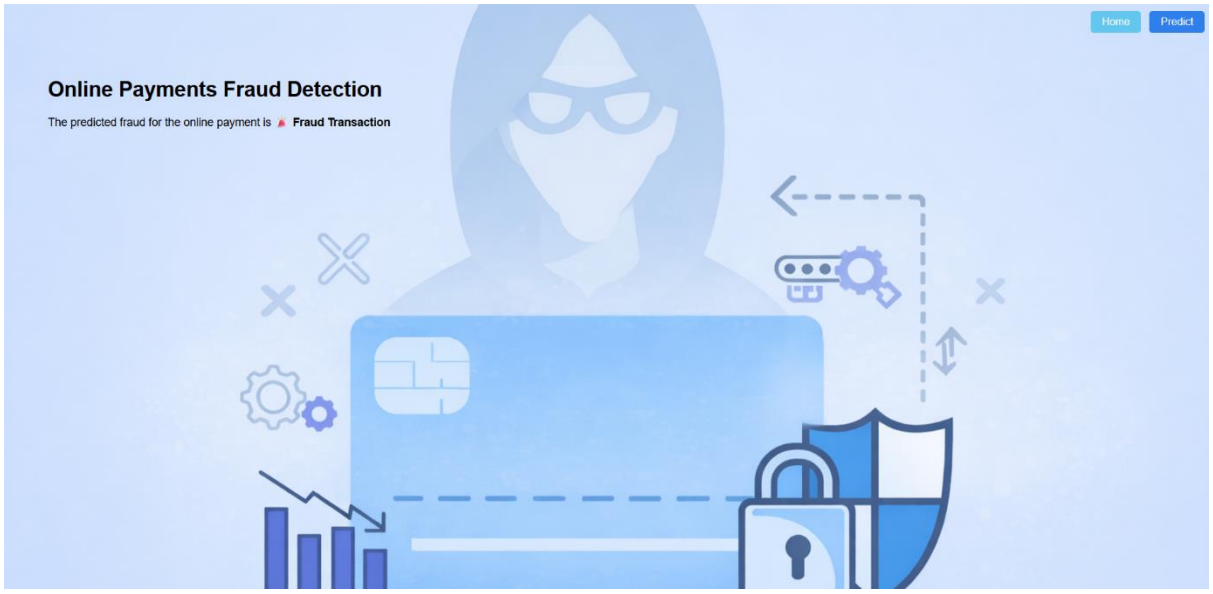
6.1 Performance Testing

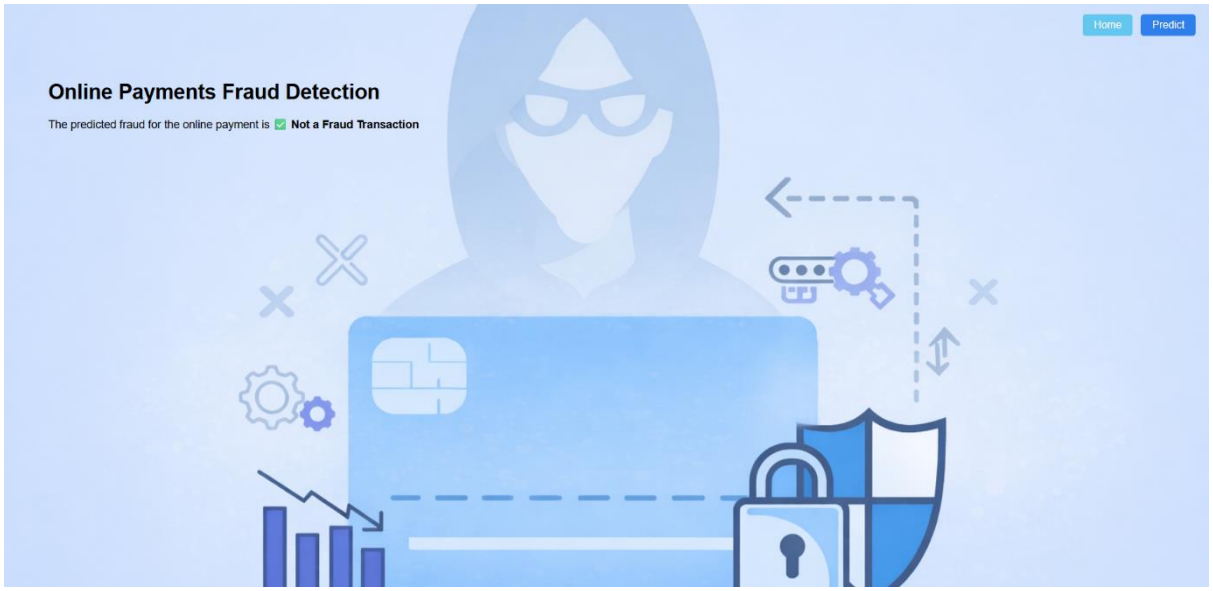
- Accuracy
- Confusion Matrix
- Precision
- Recall
- F1-Score

7. RESULTS


Home
Predict
Online Payments
Fraud Detection


Online Payments Fraud Detection
Home
Predict
Step
Type
Select Transaction Type
Select Transaction Type
CASH_OUT
DEBIT
PAYMENT
TRANSFER
NewbalanceOrig
OldbalanceDest
NewbalanceDest
Submit
FRAUD ALERT


Home
Predict
Online Payments Fraud Detection
The predicted fraud for the online payment is Fraud Transaction



8. ADVANTAGES & DISADVANTAGES

Advantages

- Real-time fraud detection
- Multiple model comparison
- Handles imbalanced dataset
- User-friendly interface

Disadvantages

- Dependent on dataset quality
- High computational cost for large data
- Requires retraining for new fraud patterns

9. CONCLUSION

The Online Payments Fraud Detection System successfully detects fraudulent transactions using Machine Learning algorithms. After comparison, Decision Tree was selected due to its performance and real-time capability. The system is deployed using Flask and provides an efficient fraud prediction solution.

10. FUTURE SCOPE

- Use Deep Learning models
- Deploy on cloud
- Implement real-time streaming detection
- Improve imbalance handling using SMOTE

11. APPENDIX

https://github.com/Likitha456/online_payments_fraud_detection