Online Payment Fraud Detection using Machine Learning

Ideation Phase

Define the Problem Statements

Date	31 January 2026
Team ID	LTVIP2026TMIDS55701
Project Name	Online Payment Fraud Detection using Machine Learning
Maximum Marks	2 Marks

Customer Problem Statement:

Problem Statemen (PS)	**I am t (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**	
PS-1	Financial Institution / Payment Gateway Operator	Detect and block fraudulent online payment transactions in real time	Existing systems rely on manual rule-based filtering	They cannot analyze compl behavioral patterns across multiple featur simultaneously	Concerned about xfinancial losses nd regulatory non-ompliance es	
PS-2	Online Shopper / End User	Conduct secure online payments without fear of fraud	Fraudulent transactions are not always caught before processing	Fraud detection systems are either unavailable or not easily accessible	Anxious and vulnerable about financial security during online purchases	

Ideation Phase Empathy Map

	31 January 2026
Team ID	LTVIP2026TMIDS55701
Project Name	Online Payment Fraud Detection using Machine Learning
Maximum Marks	2 Marks

Empathy Map:

An Empathy Map is a collaborative visualization used to articulate what we know about a particular type of user. It externalizes knowledge about users in order to create a shared understanding, and to aid in decision making. For this project, the primary user is a Financial Institution or Payment Gateway Operator trying to secure digital transactions using an automated fraud detection system.

SAYS	**THINKS**
• "We need faster, automated fraud alerts." • "Rule-based systems are generating too many false positives." • "We lose customer trust every time a fraud slips through." • "Our manual review team is overwhelmed."	• An ML model could identify patterns humans miss. • Real-time detection would reduce chargebacks significantly. • Regulatory pressure is increasing; we must act now. • A predictive model trained on our data would outperform generic tools.
DOES	**FEELS**
• Manually reviews flagged transactions daily. • Adjusts rule thresholds reactively after fraud incidents. • Spends significant resources on fraud chargeback resolution. • Integrates basic threshold-based fraud filters in payment pipeline.	• **Before:** Frustrated with reactive, slow detection; anxious about regulatory fines. • **After (with ML system):** Confident in real-time protection; relieved by reduced false positives and automated alerts.

Project Design Phase

Proposed Solution Template

	31 January 2026
Team ID	LTVIP2026TMIDS55701
Project Name	Online Payment Fraud Detection using Machine Learning
Maximum Marks	2 Marks

Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Online payment fraud causes significant financial losses to businesses and consumers. Existing
2.	Idea / Solution description	A Flask-based web application that uses a machine learning classifier (SVM/Random Forest) tr
3.	Novelty / Uniqueness	Combines multiple classification algorithms (Random Forest, SVM, Decision Tree, ExtraTrees
4.	Social Impact / Customer Satisfaction	Protects consumers from financial harm and identity theft. Reduces chargeback rates for merch
5.	Business Model (Revenue Model)	Can be offered as a SaaS (Software as a Service) to banks, fintech companies, and e-commerce
6.	Scalability of the Solution	The modular architecture allows retraining on expanded or region-specific datasets. Additional

Project Design Phase Problem – Solution Fit Template

	31 January 2026
Team ID	LTVIP2026TMIDS55701
Project Name	Online Payment Fraud Detection using Machine Learning
Maximum Marks	2 Marks

Problem – Solution Fit:

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why.

Purpose:

1. **Customer Segment(s):** Financial institutions (banks, credit card companies), payment gateway operators, e-commerce platforms, and end consumers conducting online transactions.

2. **Jobs-to-be-done / Problems:** Detecting fraudulent online payment transactions with high accuracy and providing real-time actionable classification (FRAUD / NOT FRAUD) to prevent financial loss.

3. **Triggers:** Unusual transaction amounts, abnormal account balance changes, high-frequency transactions in a short period, or transaction types inconsistent with user history.

4. **Emotions (Before/After):** Before: Anxious and financially exposed due to reactive, rule-based fraud systems. After: Confident and secure due to real-time ML-powered detection and early alerts.

5. **Available Solutions:** Manual transaction review processes, basic threshold-based rule engines, or generic fraud screening tools that do not adapt to evolving transaction patterns.

6. **Problem Root Cause:** Lack of automated, adaptive tools that can simultaneously analyze transaction amount, type, account balances, and behavioral patterns to classify fraud in real time.

7. **Your Solution:** A machine learning model achieving up to 79% accuracy (SVM) with five algorithms compared, deployed as a Flask web application that classifies transactions as FRAUD or NOT FRAUD through a simple, easy-to-use web interface.

Template:

1. CUSTOMER SEGMENT(S) CS • Financial institutions / banks • Payment gateway operators • E-commerce platforms • Online consumers	**5. CUSTOMER CONSTRAINTS CC** • Limited ML expertise in fraud teams • High cost of enterprise fraud platforms • Lack of labeled real-time transaction data	**5. AVAILABLE SOLUTIONS AS** • Manual transaction review • Basic rule-based threshold filters • Generic fraud screening APIs (static)
2. JOBS-TO-BE-DONE / PROBLEMS JAP • Detect fraud transactions with high accuracy • Provide real-time FRAUD / NOT FRAUD alerts • Reduce false positive rate for genuine transactions	**9. PROBLEM ROOT CAUSE RC** • Lack of automated ML tools analyzing transaction amount, type, and account balance features simultaneously	**6. BEHAVIOUR BE** • Manual review of flagged transactions • Reactive rule adjustment post-incident • Delayed fraud resolution, customer disputes
3. TRIGGERS TR • Unusual transaction amounts or types • Abnormal account balance changes • High transaction frequency anomalies • BEFORE: Anxious, financially exposed	**10. YOUR SOLUTION SL** A Flask web application with ML model (79% accuracy — SVM) that processes 7 transaction features and classifies as "FRAUD" or "NOT FRAUD" in real time.	**8. AVAILABLE SOLUTION CH ONLINE:** No directly comparable open-source real-time fraud web app **OFFLINE:** Rule-based bank systems, manual fraud investigation teams

Fig. 4: Problem-Solution Fit Canvas for Online Payment Fraud Detection. This illustrates alignment between customer problems and ML-based web application solution.

Project Design Phase
Solution Architecture

Date	31 January 2026
Team ID	LTVIP2026TMIDS55701
Project Name	Online Payment Fraud Detection using Machine Learning
Maximum Marks	4 Marks

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- **Structural Components**: The system is built on a **3-tier architecture**:
- **Presentation Layer**: HTML5/Bootstrap 5 interface for user transaction input.
- **Logic Layer**: Flask server (**app.py**) managing data flow, Label Encoding, Feature Scaling, and model inference.

- **Data/Model Layer**: Pickle-serialized SVM/Random Forest model and CSV training dataset.

- **Behavior**: When a user inputs transaction data (type, amount, account balances), the system performs **Label Encoding** on the transaction type and **Feature Scaling** before passing a 7-feature vector to the model for real-time classification as FRAUD or NOT FRAUD.

- **Specifications**: The solution is delivered as a local web application running on port **5000**, requiring Python packages like **scikit-learn**, **Flask**, **pandas**, **numpy**, and **pickle**.

Example - Solution Architecture Diagram:

Rising Waters: Solution Architecture Overview
Presentation Layer HTML5/BOOTSTRAP 5 DASHBOARD WEB BROWSER (User Interface) Input: step, type, amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest, newbalanceDest
Logic Layer FLASK WEB SERVER (app.py) REQUEST ROUTING ↓ DATA CAPTURE ↓ DATA PREPROCESSING (Label Encoding, Feature Scaling) 13-FEATURE VECTOR ALIGNMENT ↓ INFERENCE CALL → PREDICTION RESULT (FRAUD / NOT FRAUD)
Data / Model Layer fraud_dataset.csv (Training Data) → ML MODEL (model.pkl) Random Forest / SVM Classifier PREPROCESSING ASSETS (scaler.pkl, label_encoder.pkl)
EXTERNAL INTERFACES BOOTSTRAP CDN (Responsive UI Styles)

Fig. 6: Online Payment Fraud Detection – Solution Architecture Diagram. This diagram illustrates the 3-tier architecture, data flow, and technology stack for the fraud prediction system, from user input to real-time classification.

Online Payment Fraud Detection using Machine Learning

Team Members: -

Y Nagarjuna Reddy: 238X5A4404
S. Siva Sruthi: 228X1A4430
G Siva Gopiraju : 228X1A4450
C Omkar manikanta: 228X1A4447

Team ID:-LTVIP2026TMIDS55701

Project Overview:

Online Payment Fraud Detection using Machine Learning is a proactive, data-driven approach to identify and prevent fraudulent activities during online transactions. By leveraging historical transaction data, customer behavior patterns, and machine learning algorithms, this project aims to detect potential fraud in real time, ensuring secure and trustworthy online payment experiences for users and businesses alike. The goal is to provide timely alerts and actionable predictions to financial institutions, payment gateways, and e-commerce platforms to minimize the financial and reputational impact of online fraud.

Scenario 1: Real-time Fraud Monitoring

One of the primary use cases is the development of a real-time fraud monitoring system for online payment platforms. By analyzing transaction features such as transaction amount, type, and account balance changes, the system can flag suspicious transactions for immediate review, preventing fraudulent activities before they are completed.

Scenario 2: Fraudulent Account Detection

Machine learning models can detect patterns indicative of fraudulent accounts. By analyzing behavioral patterns over time — including unusual transaction types, sudden large transfers, and inconsistencies between old and new account balances — the system can identify and flag potentially compromised accounts, protecting legitimate users and businesses from financial harm.

Scenario 3: Adaptive Fraud Prevention

The system adapts and improves its fraud detection capabilities over time. By continuously learning from new transaction data and comparing multiple classification algorithms, it can stay ahead of evolving fraud techniques and provide ongoing protection for businesses and their customers against online payment fraud.

Technical Architecture:

The system is built on a 3-tier architecture comprising a Presentation Layer (HTML5/Bootstrap 5 web interface), a Logic Layer (Flask server managing data preprocessing and model inference), and a Data/Model Layer (Pickle-serialized ML model and CSV training dataset). User inputs transaction data through the web UI; the Flask backend performs Label Encoding and passes the feature vector to the trained classifier, which returns a real-time FRAUD or NOT FRAUD prediction displayed on the UI.

Pre-Requisites:

To complete this project, you must require the following software, concepts, and packages.

VS Code / Anaconda Navigator:

- Refer to the link to download VS Code: https://www.youtube.com/watch?v=mIVB-SNycKI
- Or download Anaconda Navigator from: https://www.anaconda.com/products/navigator

Python packages:

- Open Anaconda prompt as administrator and type "pip install numpy" and click enter.• Open Anaconda prompt as administrator and type "pip install pandas" and click enter.
- Open Anaconda prompt as administrator and type "pip install scikit-learn" and click enter.
- Open Anaconda prompt as administrator and type "pip install matplotlib" and click enter.
- Open Anaconda prompt as administrator and type "pip install pickle-mixin" and click enter.
- Open Anaconda prompt as administrator and type "pip install seaborn" and click enter.
- Open Anaconda prompt as administrator and type "pip install Flask" and click enter.
- Open Anaconda prompt as administrator and type "pip install xgboost" and click enter.

Prior Knowledge:

You must have prior knowledge of the following topics to complete this project.

- ML Concepts — Supervised learning, classification algorithms
- Supervised learning: https://www.youtube.com/watch?v=QeKshry8pWQ&t;=3s
- Model Evaluation Metrics (Accuracy, Precision, Recall, F1-Score)
- Flask web framework basics: https://www.youtube.com/watch?v=lj4I_CvBnt0
- Python programming fundamentals and pandas/numpy data manipulation

Final Project Overview Online Payment Fraud Detection using Machine Learning

Date	31 January 2026
Team ID	LTVIP2026TMIDS55701
Project Name	Online Payment Fraud Detection using Machine Learning
Maximum Marks	10 Marks

Project Summary:

This project delivers a complete, end-to-end machine learning solution for detecting fraudulent online payment transactions. Starting from raw transactional data sourced from Kaggle, the project progresses through systematic phases of data exploration, preprocessing, multi-model training, evaluation, and Flask-based web deployment. The final system classifies transactions as FRAUD or NOT FRAUD in real time through a clean web interface.

Key Outcomes:

- Trained and compared five classification algorithms: Random Forest, Decision Tree, ExtraTrees, SVM, and XGBoost on the Kaggle fraud detection dataset.
- Achieved up to 79% classification accuracy with the SVM (Support Vector Classifier) selected asthe best-performing model.
- Conducted comprehensive EDA including univariate, bivariate, and descriptive analysis with 20+visualizations.
- Built and deployed a Flask web application with three HTML pages (home, predict, result) forreal-time fraud prediction.
- Saved the trained model using Pickle for efficient loading and inference in the production Flaskapplication.
- Delivered complete 7-phase project documentation covering Ideation, Requirements, Design, Planning, Development, Documentation, and Demonstration.

Technology Stack:

Component	Technology
Programming Language	Python 3.x
ML Framework	Scikit-learn, XGBoost
Data Processing	Pandas, NumPy
Visualization	Matplotlib, Seaborn
Web Framework	Flask
Model Storage	Pickle (.pkl)
Frontend	HTML5, CSS3, Bootstrap
Dataset	Kaggle — PS_20174392719 (6.3M records)
Development Env	Anaconda Navigator / VS Code

Project Flow:

Step 1: Data Collection → Download Kaggle fraud dataset (PS_20174392719_1491204439457_logs.csv)

Step 2: Data Preprocessing → Remove unnecessary columns, handle null values, encode categorical variables, treat outliers

Step 3: Exploratory Data Analysis → Univariate, bivariate, and descriptive statistical analysis with visualizations

Step 4: Model Building → Train and evaluate 5 classification algorithms

Step 5: Model Evaluation → Compare accuracy, confusion matrix, classification report; save best model with Pickle

Step 6: Application Building → Develop Flask app with 3 HTML pages; integrate model for real-time prediction

Step 7: Documentation & Demonstration → Complete 7-phase documentation + end-to-end demo video