Itroduction

Project Overview

The **Online Payment Fraud Detection System** is a machine learning-based application designed to detect fraudulent financial transactions in real time. With the rapid growth of digital payment platforms, the risk of online fraud has increased significantly. Traditional rule-based systems are often unable to detect complex and evolving fraud patterns.

This project utilizes the **XGBoost classification algorithm** to analyze transaction data and predict whether a transaction is genuine or fraudulent. The system includes data preprocessing, feature engineering, model training, performance evaluation, and deployment through a web-based interface. By integrating machine learning techniques, the system enhances accuracy and reduces financial risks.

The application architecture consists of a frontend interface for users, a backend API for processing requests, a trained ML model for prediction, and a database for storing transaction records.

Purpose

1. To develop a reliable machine learning system capable of detecting fraudulent online transactions.
2. To improve fraud detection accuracy compared to traditional rule-based approaches.
3. To reduce financial losses caused by unauthorized or suspicious transactions.
4. To provide real-time fraud prediction and risk assessment.
5. To enhance customer trust and strengthen security in digital payment systems.

Ideation Phase
Brainstorm & Idea Prioritization Template

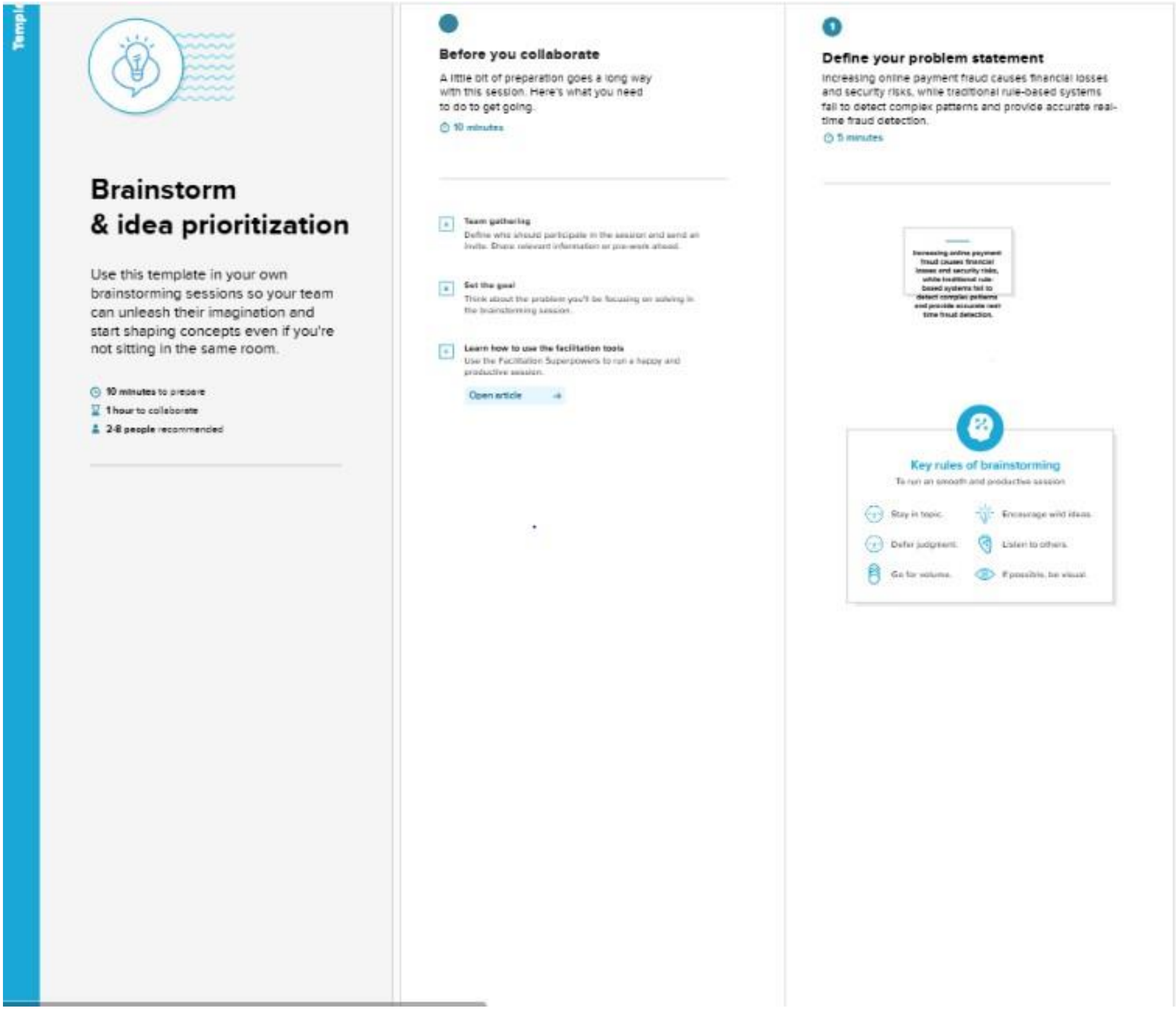
Brainstorm & Idea Prioritization Template:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference:https://app.mural.co/invitation/room/1771508067399943?code=8e77c3079f8947bc8e73326469a5c43f&sender=u8bfd3d70dc48c34b8d019375

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Define your problem statement

Increasing online payment fraud causes financial losses and security risks, while traditional rule-based systems fail to detect complex patterns and provide accurate real-time fraud detection.

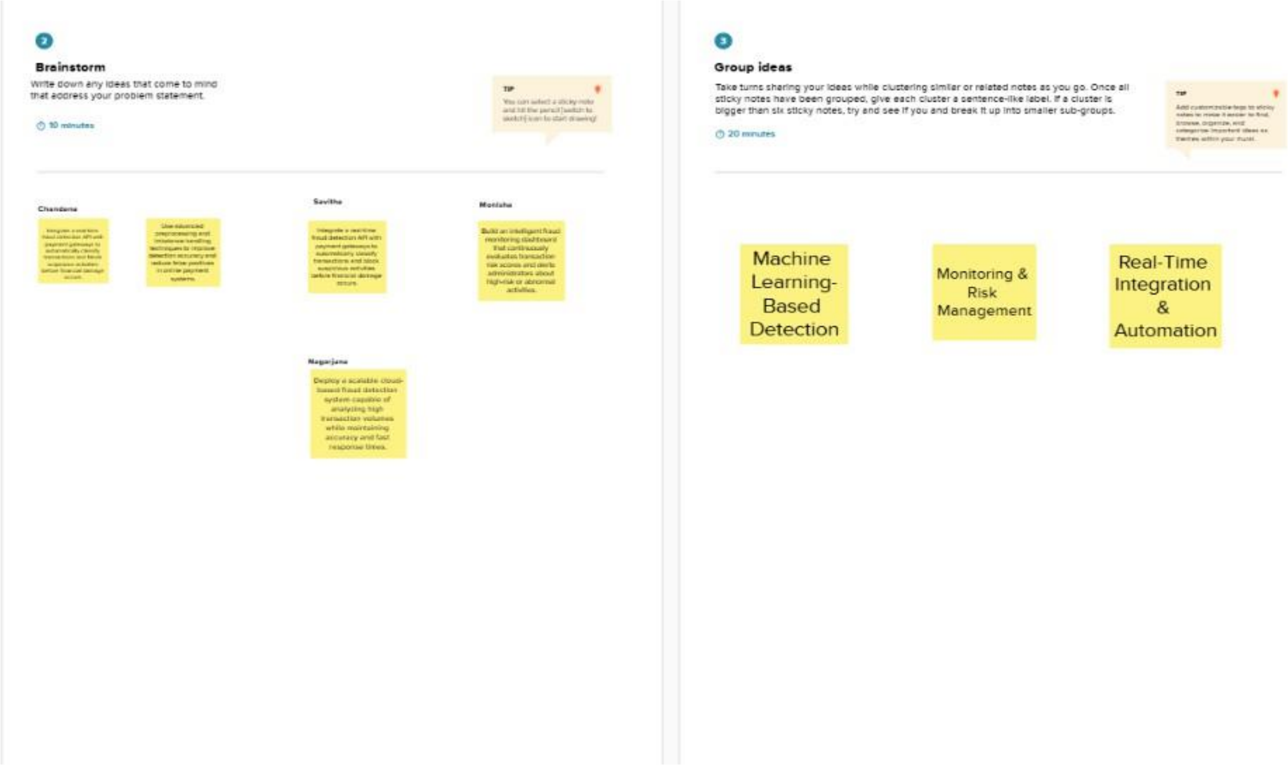
5 minutes

Increasing online payment fraud causes financial losses and security risks, while traditional rule-based systems fail to detect complex patterns and provide accurate real-time fraud detection.

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Chandana

Integrate a real-time fraud detection API with payment gateways to automatically classify transactions and block suspicious activities before financial damage occurs.

Use advanced preprocessing and imbalance handling techniques to improve detection accuracy and reduce false positives in online payment systems.

Savitha

Integrate a real-time fraud detection API with payment gateways to automatically classify transactions and block suspicious activities before financial damage occurs.

Monisha

Build an intelligent fraud monitoring dashboard that continuously evaluates transaction risk scores and alerts administrators about high-risk or abnormal activities.

Nagarjuna

Deploy a scalable cloud-based fraud detection system capable of analyzing high transaction volumes while maintaining accuracy and fast response times.

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

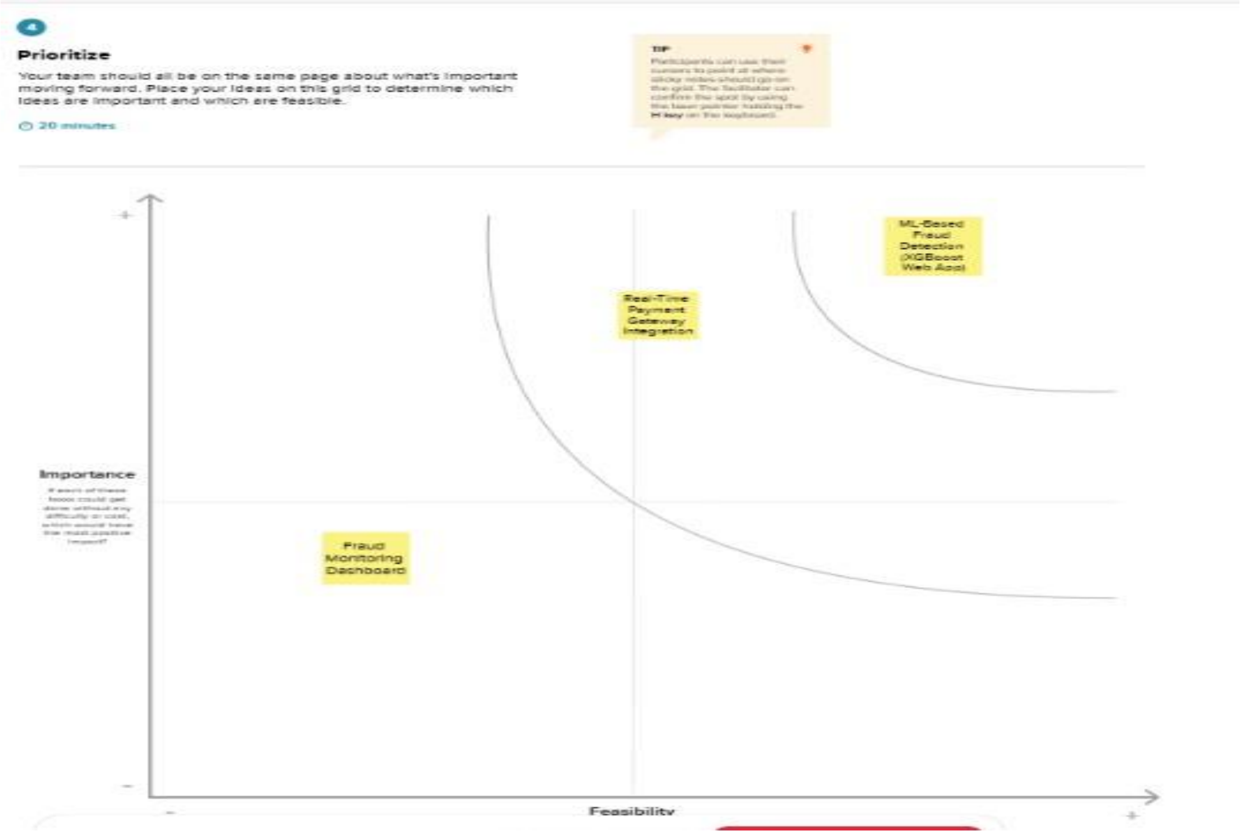
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Machine Learning-Based Detection

Monitoring & Risk Management

Real-Time Integration & Automation

Step-3: Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.

+
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
-
-
Feasibility
+

ML-Based Fraud Detection (XGBoost Web App)

Real-Time Payment Gateway Integration

Fraud Monitoring Dashboard

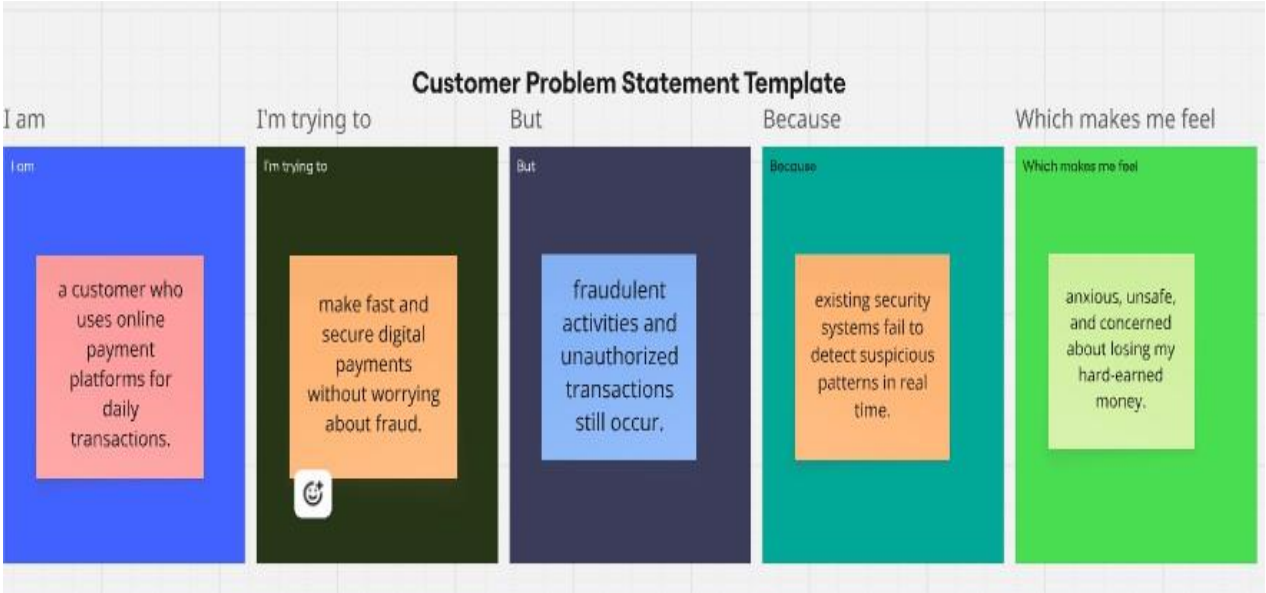
Ideation Phase
Define the Problem Statements

Customer Problem Statement Template:

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

Reference: https://miro.com/templates/customer-problem-statement/



Customer Problem Statement Template

I am	I'm trying to	But	Because	Which makes me feel
I am	I'm trying to	But	Because	Which makes me feel
a customer who uses online payment platforms for daily transactions.	make fast and secure digital payments without worrying about fraud.	fraudulent activities and unauthorized transactions still occur.	existing security systems fail to detect suspicious patterns in real time.	anxious, unsafe, and concerned about losing my hard-earned money.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	a customer who uses online payment platforms for daily transactions.	make fast and secure digital payments without worrying about fraud.	fraudulent activities and unauthorized transactions still occur.	existing security systems fail to detect suspicious patterns in real time.	anxious, unsafe, and concerned about losing my hard-earned money.

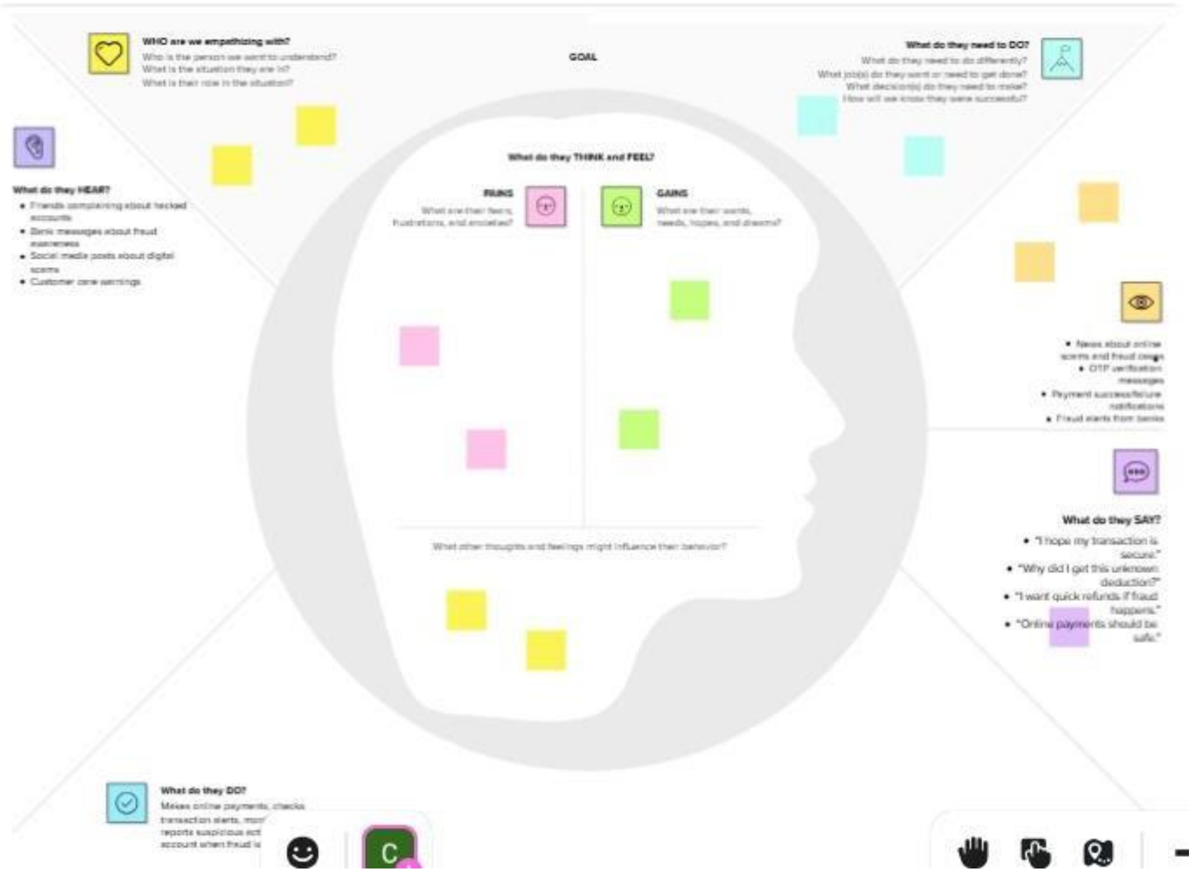
Ideation Phase
Empathize & Discover

Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users.
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Example:



WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?

What do they HEAR?
- Friends complaining about hacked accounts
- Bank messages about fraud awareness
- Social media posts about digital scams
- Customer care warnings

What do they THINK and FEEL?

PAINS
What are their fears, frustrations, and anxieties?

GAINS
What are their wants, needs, hopes, and dreams?

What other thoughts and feelings might influence their behavior?

- News about online scams and fraud cases
- OTP verification messages
- Payment success/failure notifications
- Fraud alerts from banks

What do they SAY?
- "I hope my transaction is secure."
- "Why did I get this unknown deduction?"
- "I want quick refunds if fraud happens."
- "Online payments should be safe."

What do they DO?
Makes online payments, checks transaction alerts, mon[illegible] reports suspicious act[illegible] account when fraud is

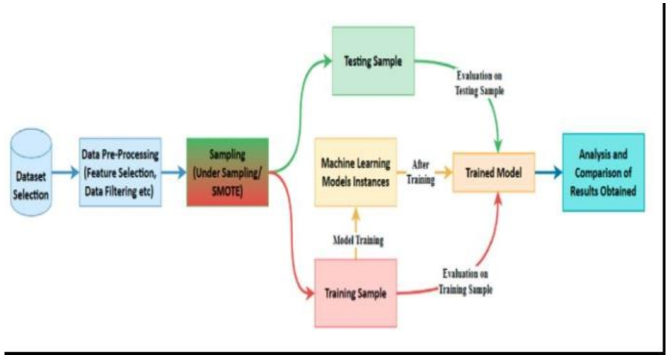
Reference: https://www.mural.co/templates/empathy-map-canvas

Project Design Phase-II
Data Flow Diagram & User Stories

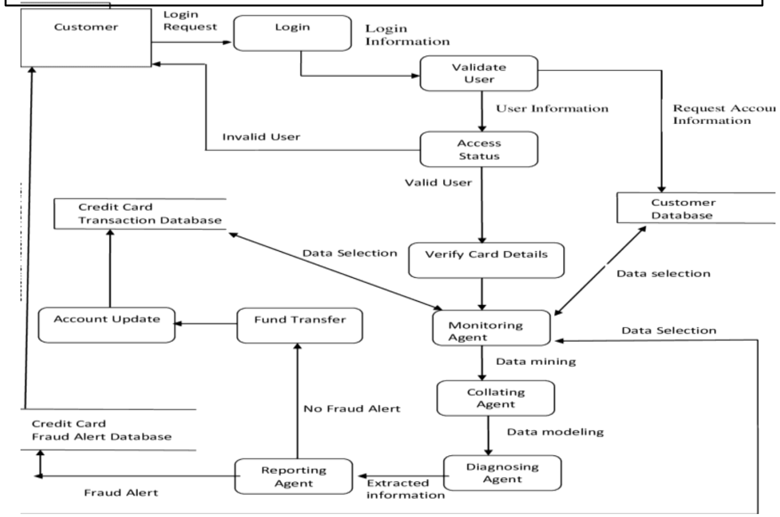
Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example


Dataset Selection
Data Pre-Processing (Feature Selection, Data Filtering etc)
Sampling (Under Sampling/ SMOTE)
Testing Sample
Machine Learning Models Instances
Training Sample
Model Training
After Training
Evaluation on Testing Sample
Evaluation on Training Sample
Trained Model
Analysis and Comparison of Results Obtained

Example: DFD Level 0(Industry standard)



Flow

1. Input Transaction Details
2. Data Preprocessing
3. Fraud Prediction using ML Model
4. Display Result

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Transaction Fraud Check	USN-1	As a user, I want to enter my transaction details so that I can check whether my transaction is safe or fraudulent.	• User can enter all required transaction fields. System validates input Submit button works	High	Sprint-1
		USN-2	As a user, I want the system to analyze my transaction using machine learning so that I get an accurate fraud probability.	• Every transaction is stored Risk percentage is stored Decision status is stored	High	Sprint-1
		USN-3	As a user, I want to know whether my transaction is approved, under review, or blocked.	• Decision is shown clearly Risk level is shown UI displays result instantly	Low	Sprint-2
		USN-4	As a user, I want high-risk transactions to be blocked automatically for security.	• High-risk transactions are marked BLOCKED Account status changes to SUSPENDED	Medium	Sprint-1
	Monitoring Transactions	USN-5	As an admin, I want all transactions stored in the database so that I can monitor fraud patterns.	• Every transaction is stored Risk percentage	High	Sprint-1

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Transaction Input Management	The system shall allow users to enter transaction details (Time, Amount).
FR-2	Input Validation	The system shall validate all required fields before submission.
FR-3	Fraud Prediction Processing	The system shall process input data using the trained ML model.
FR-4	Data Preprocessing	The system shall scale and transform input data using the saved scaler before prediction.
FR-5	Fraud Probability Calculation	The system shall generate fraud probability using the trained XGBoost model.
FR-6	Result Display	The system shall display whether the transaction is Legitimate or Fraudulent
FR-7	Risk Score Display	The system shall display the fraud risk percentage on the UI.
FR-8	Error Handling	The system shall display meaningful error messages for invalid or incomplete input.
FR-9	Model Loading	The system shall load the trained model (.pkl file) at application startup.
FR-10	Web Interface Access	The system shall allow users to access the application through a web browser
FR-11	Prediction Response	The system shall return prediction results instantly after submission.
FR-12	Transaction Logging (Optional Enhancement)	The system shall store transaction results for future monitoring (if implemented).

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	The user interface shall be simple, intuitive, and easy to use.
NFR-2	**Security**	The system shall prevent unauthorized modification of the trained ML model.
NFR-3	**Reliability**	The system shall operate without crashing during valid user input.
NFR-4	**Performance**	The system shall generate prediction results within 2–3 seconds.
NFR-5	**Availability**	The web application shall be accessible whenever the server is running.
NFR-6	**Scalability**	The system shall allow future integration with real-time payment gateways.

Project Design Phase-II Technology Stack (Architecture & Stack)

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

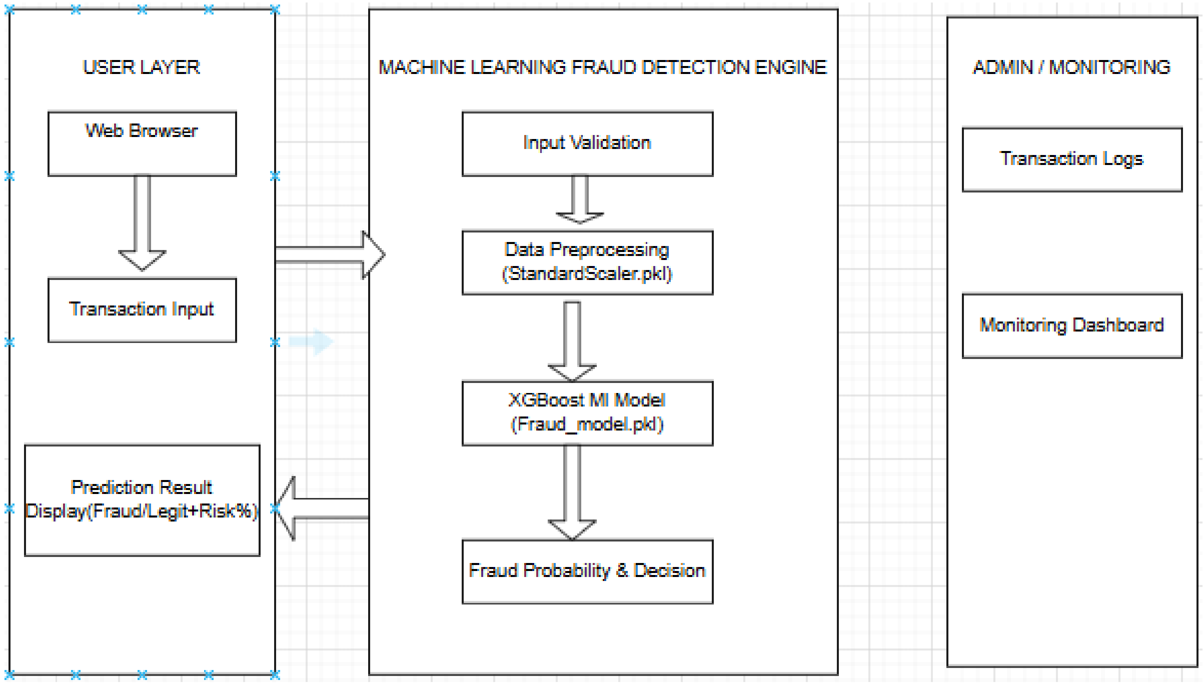

USER LAYER
Web Browser
Transaction Input
Prediction Result Display(Fraud/Legit+Risk%)
MACHINE LEARNING FRAUD DETECTION ENGINE
Input Validation
Data Preprocessing (StandardScaler.pkl)
XGBoost Ml Model (Fraud_model.pkl)
Fraud Probability & Decision
ADMIN / MONITORING
Transaction Logs
Monitoring Dashboard

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web interface used to input transaction details and display prediction results	HTML, CSS
2.	Application Logic-1	Handles routing, request processing, and response generation	Python,Flask
3.	Application Logic-2	Input validation and data formatting before model prediction	Flask Backend (Python)
4.	Application Logic-3	Fraud prediction processing using trained model	XGBoost (Scikit-learn)
5.	Database	Stores transaction logs and prediction results (if implemented)	MySQL / SQLite
6.	Cloud Database	Stores trained ML model and scaler files	Pickle (.pkl files)
7.	File Storage	Scales and transforms input data before prediction	StandardScaler (Scikit-learn)
8.	External API-1	Not applicable in current version (future integration with payment gateways possible)	REST API (Future Scope)
9.	Machine Learning Model	Performs fraud classification and probability prediction	XGBoost Classifier
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: http://127.0.0.1:5000	. Localhost (Flask Server) / Cloud (Future Deployment)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frameworks used for development of the application	Flask, Scikit-learn, XGBoost
2.	Security Implementations	Input validation, model protection, secure backend processing	Flask Validation, Python Backend Security
3.	Scalable Architecture	System designed to support future integration with real-time payment systems	Layered Architecture (Frontend + Backend + ML)
4.	Availability	Application runs continuously when Flask server is active	Flask Development Server
5.	Performance	Generates fraud prediction within few seconds	XGBoost Optimized Model

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d

Project Design Phase
Problem – Solution Fit Template

Problem – Solution Fit Template:

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

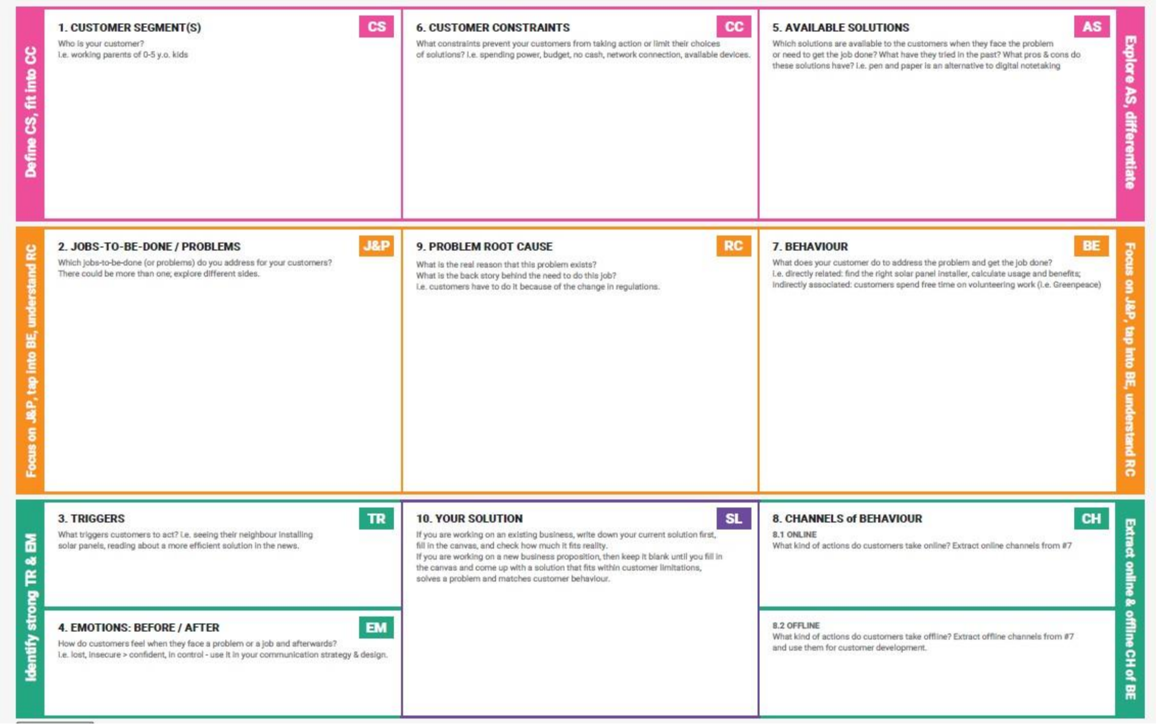
Purpose:

The purpose of this project is to develop a machine learning-based fraud detection system that accurately identifies fraudulent online payment transactions in real time.

The system aims to analyze transaction data using advanced algorithms to reduce financial losses, improve transaction security, and enhance user trust in digital payment platforms.

By replacing traditional rule-based detection methods with an intelligent predictive model, the project seeks to provide faster, more accurate, and scalable fraud detection solutions for modern financial systems.

Template:



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS
Who is your customer?
i.e. working parents of 0-5 y.o. kids

6. CUSTOMER CONSTRAINTS CC
What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

5. AVAILABLE SOLUTIONS AS
Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P
Which jobs-to-be-done (or problems) do you address for your customers?
There could be more than one; explore different sides.

9. PROBLEM ROOT CAUSE RC
What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

7. BEHAVIOUR BE
What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR
What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

4. EMOTIONS: BEFORE / AFTER EM
How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

10. YOUR SOLUTION SL
If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

8. CHANNELS of BEHAVIOUR CH
8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Extract online & offline CH of BE

References:

1. https://www.ideahackers.network/problem-solution-fit-canvas/
2. https://medium.com/@epicantus/problem-solution-fit-canvas-aa3dd59cb4fe

Project Design Phase Proposed Solution Template

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)	Increasing online payment fraud causes financial loss, reduced trust, and security risks due to ineffective traditional rule-based detection systems.
2.	Idea / Solution description	A machine learning-based fraud detection system using XGBoost that analyzes transaction data, calculates fraud probability, and provides real-time classification through a Flask web application.
3.	Novelty / Uniqueness	Uses advanced ML model (XGBoost) for detecting complex fraud patterns instead of static rules. Provides probability-based risk scoring and real-time prediction through a web interface.
4.	Social Impact / Customer Satisfaction	Reduces financial fraud losses, improves digital transaction security, enhances trust in online payments, and supports safer digital economy growth.
5.	Business Model (Revenue Model)	Can be offered as a Fraud Detection API service to banks and payment gateways with subscription-based or usage-based pricing.
6.	Scalability of the Solution	Easily scalable for integration with banking systems, payment gateways, and cloud deployment for handling high transaction volumes.

Project Design
Phase Solution
Architecture

Solution Architecture:

The solution architecture defines the structure and workflow of the Online Payment Fraud Detection System. It integrates a machine learning model with a web-based application to provide real-time fraud detection.

The architecture bridges the gap between online payment users and intelligent fraud detection using a layered design approach.

The goals of the solution architecture are:

- Identify and detect fraudulent transactions accurately using machine learning.
- Provide real-time fraud prediction through a web interface.
- Ensure modular and scalable system design.
- Maintain system reliability and fast response time.
- Enable future integration with banking systems and payment gateways.

Example - Solution Architecture Diagram:

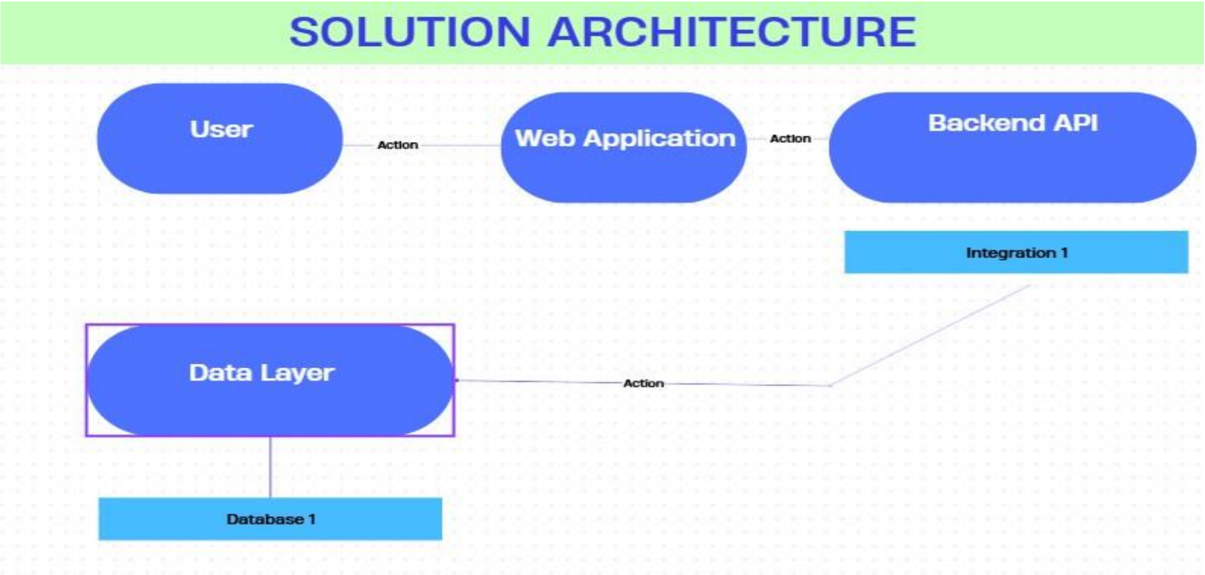


Figure 1: Architecture and data flow of the online payments fraud detection using machine learning

Reference: [**https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research- powered-by-ai-on-aws-part-1-architecture-and-design-considerations/**](https://aws.amazon.com/blogs/industries/voice-applications-in-clinical-research-powered-by-ai-on-aws-part-1-architecture-and-design-considerations/)

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	15 February 2025
Team ID	LTVIP2026TMIDS79606
Project Name	online payments fraud detection using machine learning
Maximum Marks	5 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dataset Preparation	User 01	Collect and explore credit card fraud dataset	3	High	Chandana
Sprint-1	Data Preprocessing	User 01	Perform data cleaning and handle missing values	5	High	Chandana
Sprint-1	Feature Engineering	User 01	Apply feature scaling using StandardScaler	5	High	Savitha
Sprint-1	Model Development	User 02	Train XGBoost fraud detection model	8	High	Savitha
Sprint-1	Model Evaluation	User 01	Evaluate model using accuracy, precision, recall, ROC-AUC	5	High	Chandana
Sprint-2	Model Optimization	User 01	Tune hyperparameters for better performance	5	Medium	Monisha
Sprint-2	Backend Development	User 01	Develop Flask backend application	8	High	Monisha
Sprint-2	API Integration	User 02	Integrate ML model with Flask backend	5	High	Nagarjuna
Sprint-2	Frontend Development	User 02	Design HTML/CSS web interface	5	High	Nagarjuna
Sprint-2	Input Validation	User 02	Implement input validation in Flask	3	Medium	Chandana

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Prediction Display	User 01	Display fraud result and probability on UI	5	High	Chandana

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	26	2 weeks	20 Dec 2025	4 Jan 20226	25	5 Jan 2026
Sprint-2	21	2 Weeks	6 Jan 2026	14 Jan 2026	21	15 Jan 2026
Sprint-3	18	2 weeks	16 Jan 2026	30 Jan 2026	18	31 Jan 2026
Sprint-4	13	2 weeks	01 Feb 2026	14-Feb 2026	13	15 Feb 2026

Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let’s calculate the team’s average velocity (AV) per iteration unit (story points per day)

Average Velocity = 75 /4

Average Velocity =18.7

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

https://www.visual-paradigm.com/scrum/scrum-burndown-chart/

https://www.atlassian.com/agile/tutorials/burndown-charts

Reference:

https://www.atlassian.com/agile/project-management

https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software

https://www.atlassian.com/agile/tutorials/epics

https://www.atlassian.com/agile/tutorials/sprints

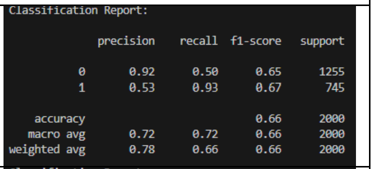
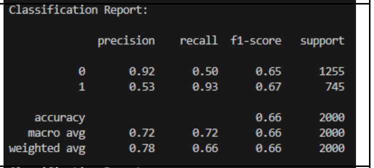
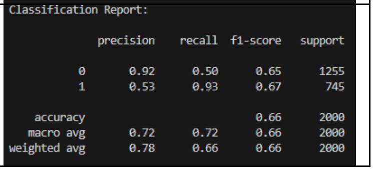
https://www.atlassian.com/agile/project-management/estimation

https://www.atlassian.com/agile/tutorials/burndown-charts

Project Development Phase
Model Performance Test

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

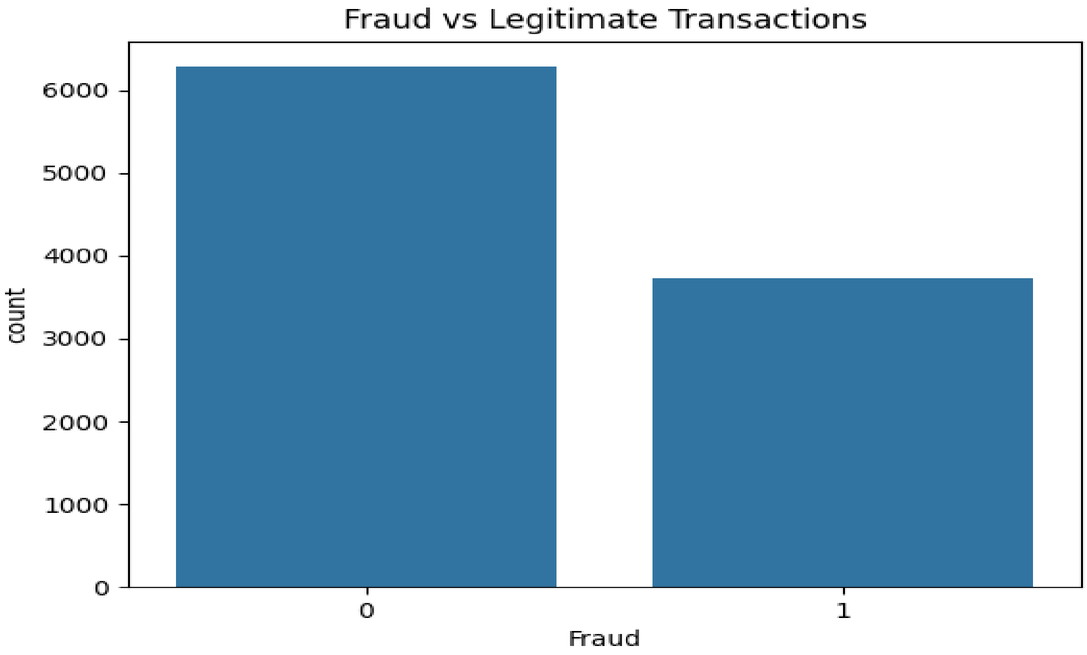
S.No.	Parameter	Values	Screenshot
1.	Model Summary		
2.	Accuracy	Training Accuracy - 80 Validation Accuracy -80	
3.	Fine Tunning Result(if Done)	Validation Accuracy -90	

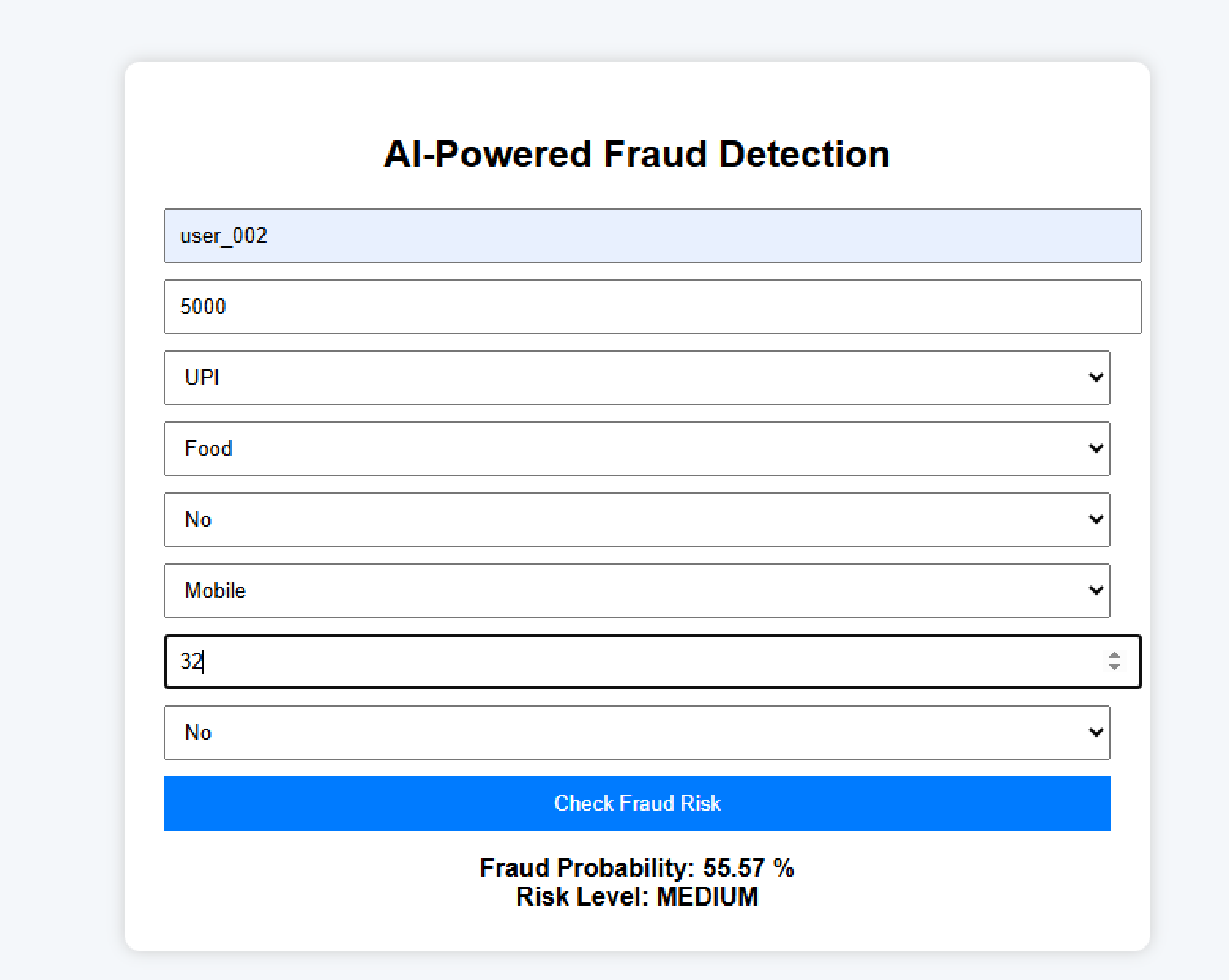
```
Classification Report:

              precision    recall  f1-score   support

           0       0.92      0.50      0.65      1255
           1       0.53      0.93      0.67       745

    accuracy                           0.66      2000
   macro avg       0.72      0.72      0.66      2000
weighted avg       0.78      0.66      0.66      2000
```

```
Classification Report:

              precision    recall  f1-score   support

           0       0.92      0.50      0.65      1255
           1       0.53      0.93      0.67       745

    accuracy                           0.66      2000
   macro avg       0.72      0.72      0.66      2000
weighted avg       0.78      0.66      0.66      2000
```

```
Classification Report:

              precision    recall  f1-score   support

           0       0.92      0.50      0.65      1255
           1       0.53      0.93      0.67       745

    accuracy                           0.66      2000
   macro avg       0.72      0.72      0.66      2000
weighted avg       0.78      0.66      0.66      2000
```

RESULTS
OUTPUT SCREENSHOTS





AI-Powered Fraud Detection

user_002

5000

UPI

Food

No

Mobile

32

No

Fraud Probability: 55.57 %
Risk Level: MEDIUM

ADVANTAGES:

1. Provides real-time fraud detection to prevent financial losses.
2. Improves accuracy compared to traditional rule-based systems.
3. Reduces false positives using advanced machine learning techniques.
4. Scalable system capable of handling large transaction volumes.
5. Enhances customer trust in digital payment platforms.
6. Automated risk assessment minimizes manual monitoring effort.
7. Adaptive model can learn evolving fraud patterns.

DISADVANTAGES:

1. Requires high-quality and large datasets for effective training.
2. Model performance may decrease if fraud patterns change significantly.
3. Implementation cost may be high for small-scale businesses.
4. Risk of false positives affecting genuine customers.
5. Requires continuous monitoring and model retraining.

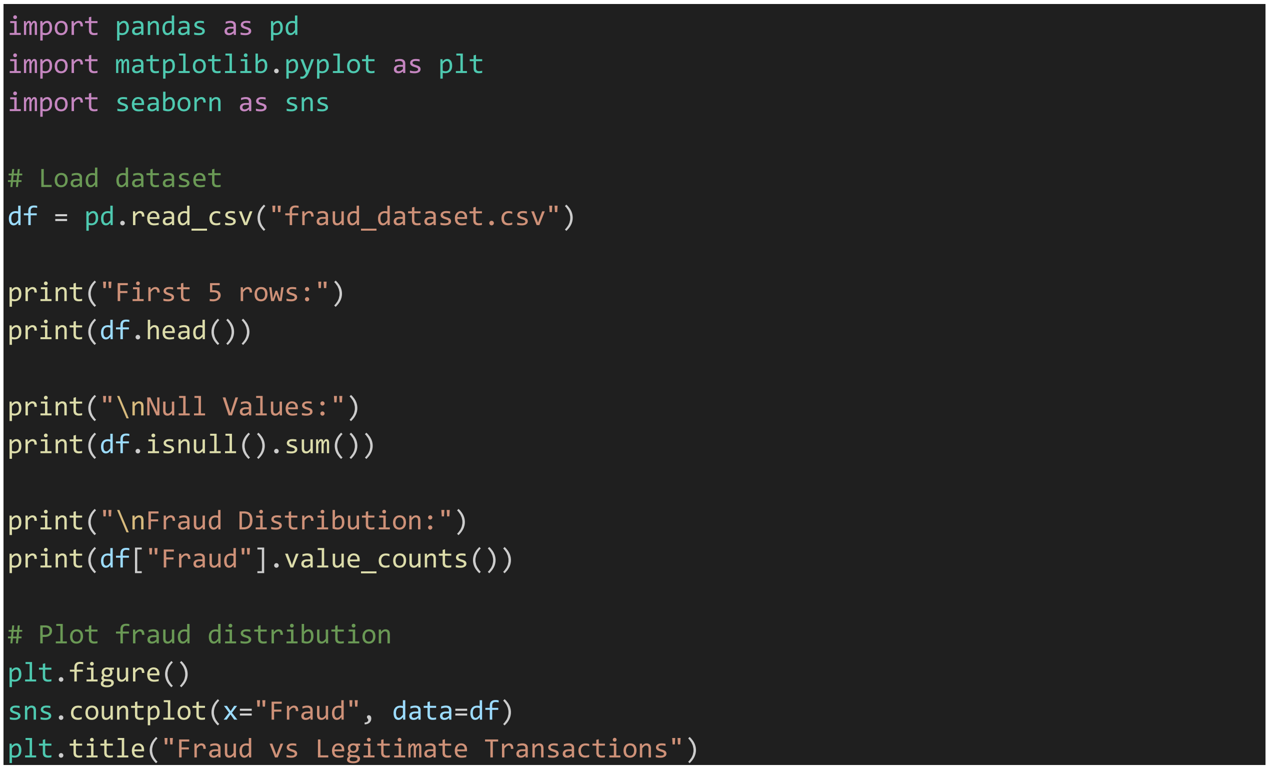
CONCLUSION:

The Online Payment Fraud Detection System using XGBoost provides an efficient and scalable solution to detect fraudulent transactions in real time. By replacing traditional rule-based systems with a machine learning approach, the system improves accuracy, reduces financial losses, and enhances customer trust in digital payments. The integration of preprocessing, imbalance handling, and performance evaluation ensures reliable fraud prediction. Overall, the system contributes to strengthening security in digital financial transactions.

FUTURE SCOPE:

- Integration with real-time banking and payment gateway APIs.
- Deployment on cloud platforms for large-scale transaction monitoring.
- Implementation of deep learning models for improved accuracy.
- Addition of behavioral biometrics for enhanced fraud detection.
- Development of a mobile-based fraud monitoring application.
- Integration of explainable AI (XAI) for model transparency.
- Continuous automated model retraining using live transaction data.

APPENDIX:
SOURCE CODE

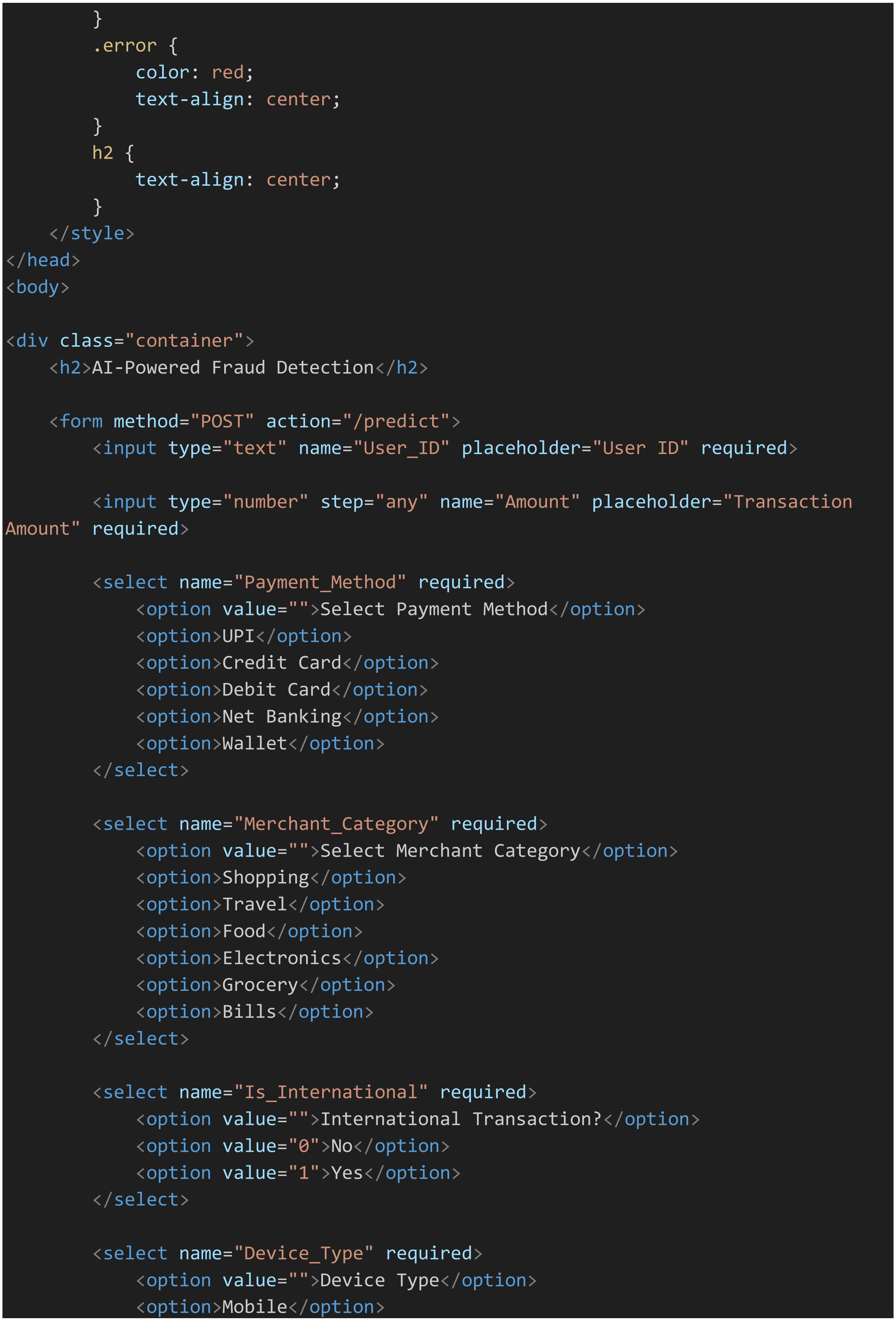


```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset
df = pd.read_csv("fraud_dataset.csv")

print("First 5 rows:")
print(df.head())

print("\nNull Values:")
print(df.isnull().sum())

print("\nFraud Distribution:")
print(df["Fraud"].value_counts())

# Plot fraud distribution
plt.figure()
sns.countplot(x="Fraud", data=df)
plt.title("Fraud vs Legitimate Transactions")
plt.show()

# -------------------------
# Encode categorical columns
# -------------------------
df_encoded = pd.get_dummies(df, columns=[
    "Payment_Method",
    "Merchant_Category",
    "Device_Type"
], drop_first=True)

# Save encoded dataset
df_encoded.to_csv("fraud_dataset_encoded.csv", index=False)

print("\nEncoded dataset saved successfully!")
```

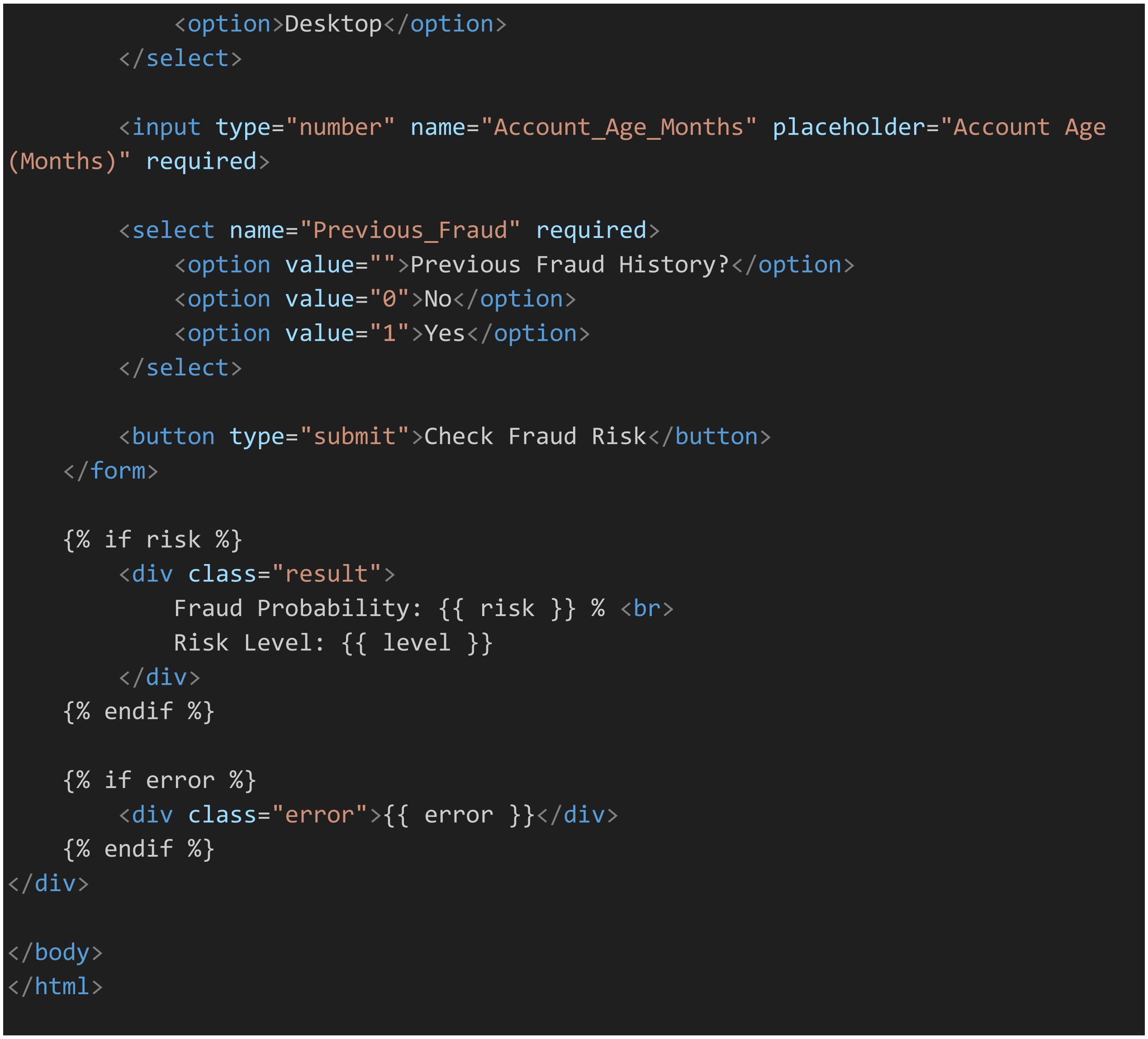
```
import pandas as pd
import joblib
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from imblearn.over_sampling import SMOTE

# Load encoded dataset
df = pd.read_csv("fraud_dataset_encoded.csv")

X = df.drop("Fraud", axis=1)
y = df["Fraud"]

# Train Test Split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# 🔥 Apply SMOTE
smote = SMOTE(random_state=42)
X_train_sm, y_train_sm = smote.fit_resample(X_train, y_train)

# 🔥 Use class_weight
model = RandomForestClassifier(
    n_estimators=200,
    max_depth=10,
    class_weight="balanced",
    random_state=42
)

model.fit(X_train_sm, y_train_sm)
```

```python
# Predict
y_probs = model.predict_proba(X_test)[:, 1]

# 🔥 Adjust threshold here
threshold = 0.35
y_pred = (y_probs > threshold).astype(int)

print("\nClassification Report:\n")
print(classification_report(y_test, y_pred))


# Save model
joblib.dump(model, "fraud_model.pkl")

print("\nModel saved successfully!")

import pandas as pd
import joblib
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from imblearn.over_sampling import SMOTE

# Load encoded dataset
df = pd.read_csv("fraud_dataset_encoded.csv")

X = df.drop("Fraud", axis=1)
y = df["Fraud"]

# Train Test Split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

# 🔥 Apply SMOTE
smote = SMOTE(random_state=42)
X_train_sm, y_train_sm = smote.fit_resample(X_train, y_train)

# 🔥 Use class_weight
model = RandomForestClassifier(
    n_estimators=200,
    max_depth=10,
    class_weight="balanced",
    random_state=42
)

model.fit(X_train_sm, y_train_sm)

# Predict
```

```
y_probs = model.predict_proba(X_test)[:, 1]

# 🔥 Adjust threshold here
threshold = 0.35
y_pred = (y_probs > threshold).astype(int)

print("\nClassification Report:\n")
print(classification_report(y_test, y_pred))


# Save model
joblib.dump(model, "fraud_model.pkl")

print("\nModel saved successfully!")

<!DOCTYPE html>
<html>
<head>
    <title>AI Fraud Detection System</title>
    <style>
        body {
            font-family: Arial;
            background: #f4f7fa;
        }
        .container {
            width: 600px;
            margin: 40px auto;
            background: white;
            padding: 25px;
            border-radius: 10px;
            box-shadow: 0px 0px 10px #ccc;
        }
        input, select {
            width: 100%;
            padding: 8px;
            margin-bottom: 10px;
        }
        button {
            width: 100%;
            padding: 10px;
            background: #007bff;
            color: white;
            border: none;
            cursor: pointer;
        }
        .result {
            margin-top: 15px;
            font-weight: bold;
            text-align: center;
```



```
        }
        .error {
            color: red;
            text-align: center;
        }
        h2 {
            text-align: center;
        }
    </style>
</head>
<body>

<div class="container">
    <h2>AI-Powered Fraud Detection</h2>

    <form method="POST" action="/predict">
        <input type="text" name="User_ID" placeholder="User ID" required>

        <input type="number" step="any" name="Amount" placeholder="Transaction
Amount" required>

        <select name="Payment_Method" required>
            <option value="">Select Payment Method</option>
            <option>UPI</option>
            <option>Credit Card</option>
            <option>Debit Card</option>
            <option>Net Banking</option>
            <option>Wallet</option>
        </select>

        <select name="Merchant_Category" required>
            <option value="">Select Merchant Category</option>
            <option>Shopping</option>
            <option>Travel</option>
            <option>Food</option>
            <option>Electronics</option>
            <option>Grocery</option>
            <option>Bills</option>
        </select>

        <select name="Is_International" required>
            <option value="">International Transaction?</option>
            <option value="0">No</option>
            <option value="1">Yes</option>
        </select>

        <select name="Device_Type" required>
            <option value="">Device Type</option>
            <option>Mobile</option>
```



```
            <option>Desktop</option>
        </select>

        <input type="number" name="Account_Age_Months" placeholder="Account Age (Months)" required>

        <select name="Previous_Fraud" required>
            <option value="">Previous Fraud History?</option>
            <option value="0">No</option>
            <option value="1">Yes</option>
        </select>

        <button type="submit">Check Fraud Risk</button>
    </form>

    {% if risk %}
        <div class="result">
            Fraud Probability: {{ risk }} % <br>
            Risk Level: {{ level }}
        </div>
    {% endif %}

    {% if error %}
        <div class="error">{{ error }}</div>
    {% endif %}
</div>

</body>
</html>
```

DATASET LINK:

https://drive.google.com/file/d/1wmjaN91OcwQcWFkuQhIpsbqbGBXyZvJB/view?usp=sharing