ONLINE PAYMENTS FRAUD DETECTION USING MACHINE LEARNING

Project Title: "Online Payments Fraud Detection using Machine Learning **"**is a bona fide work carried out by the following students:

- TEAM ID**:** LTVIP2026TMIDS61121

- TEAM LEADER: SHAIK KHADEER (228B1A0550)
- TEAM MEMBER: KANAMARLAPUDI VENKATA SAI NAGA LOKESH (228B1A0542)
- TEAM MEMBER: VARAGANI SIVA KUMAR (218B1A0556)
- TEAM MEMBER: KODURI VIJAY (218B1A0544)

Date Of Submission: 20-02-2026

PROJECT REPORT

1.INTRODUCTION

→ PROJECT OVERVIEW:

- The rapid growth of digital payment platforms has significantly increased the risk of fraudulent online transactions. Detecting fraudulent activities in real-time has become a critical challenge for financial institutions and payment gateways. This project presents a **Machine Learning–based Online Payments Fraud Detection System** designed to accurately identify whether a transaction is **fraudulent or legitimate (not fraud)** based on historical transaction data.
- The system uses a well-known financial transactions dataset containing features such as transaction type, amount, sender and receiver balances, and account details. By applying data preprocessing, feature engineering, and multiple classification algorithms, the model learns hidden patterns that distinguish fraudulent transactions from genuine ones.
- Several ML models such as **Random Forest**, **Decision Tree**, **Extra Trees**, and **Support Vector Machine (SVM)** are trained and evaluated to achieve high detection accuracy. Performance is measured using metrics like **Accuracy**, **F1-Score**, **Confusion Matrix**, and **Classification Report** to ensure reliable fraud prediction, especially for the minority fraud class.
- The final outcome of the project is a trained model that can be used to predict fraud in new online payment transactions, helping reduce financial losses and enhancing security in digital payment systems.

→ PURPOSE:

- The purpose of this project is to build a simple Machine Learning–based web application that predicts whether an online payment transaction is Fraud or Not Fraud using a Kaggle dataset.
- This application is developed as an academic project submission to demonstrate how Machine Learning can be applied to real-world problems like online payment fraud detection.

  The system:

  - Uses historical transaction data from Kaggle for training the model.
  - Applies ML algorithms to learn patterns between fraud and genuine transactions.
  - Provides a web interface where transaction details can be entered.
  - Predicts the result as Fraud or Not Fraud.
- This project mainly focuses on implementing ML concepts, model training, and deploying the model using Flask in the form of a basic working application.

2. IDEATION PHASE

→ PROBLEM STATEMENT:

In today's digital payment ecosystem, a large number of online transactions occur every second. Among these, fraudulent transactions are increasing rapidly, causing significant financial losses to users and organizations. Traditional rule-based fraud detection methods are often slow, less accurate, and unable to adapt to new fraud patterns. There is a need for an intelligent system that can automatically analyze transaction details and accurately predict whether a transaction is **Fraud** or **Not Fraud** in real time.

→ EMPATHY MAP CANAVS:

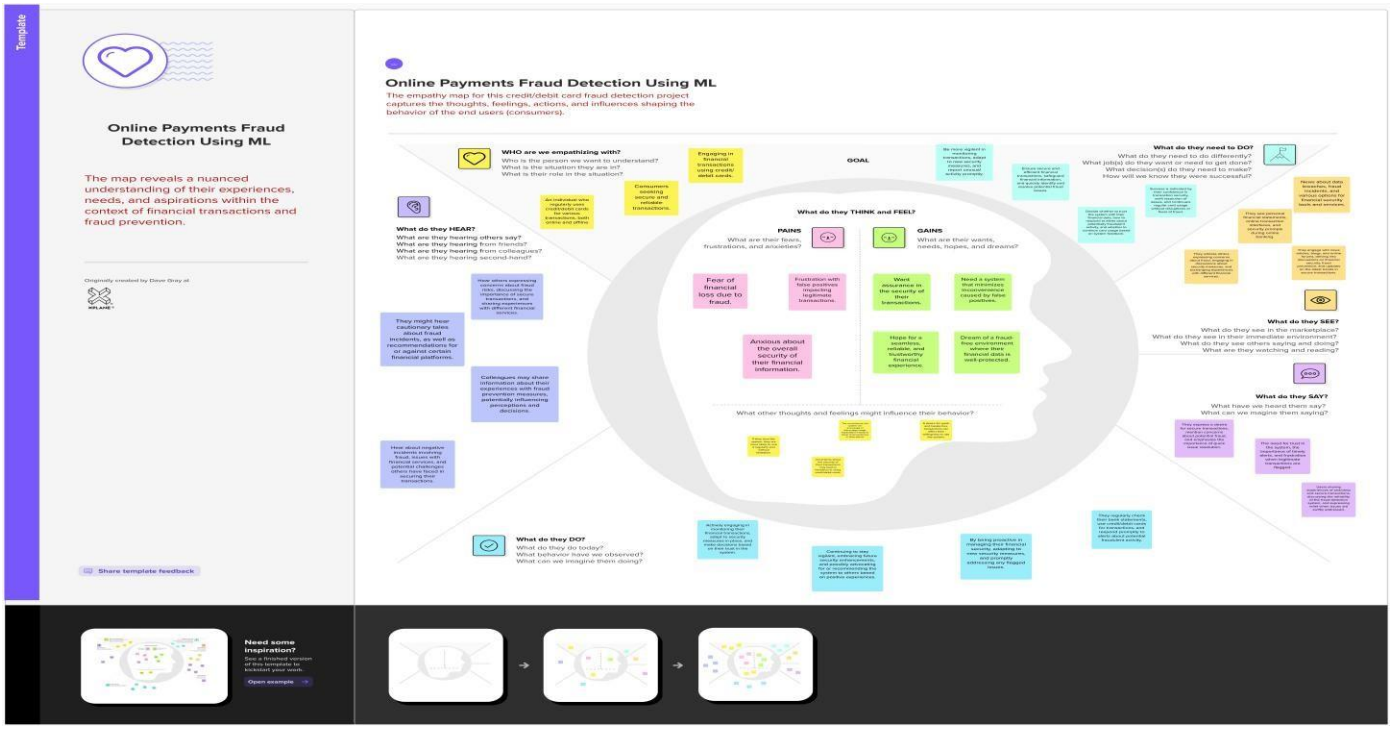
Ideation Phase Empathize & Discover

Date	25 January 2026
Team ID	LTVIP2026TMIDS61121
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	4 Marks

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.



→ BRAINSTROMING:

Ideation Phase Brainstorm & Idea Prioritization Template

Date	**27 January 2026**
Team ID	LTVIP2026TMIDS61121
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	4 Marks

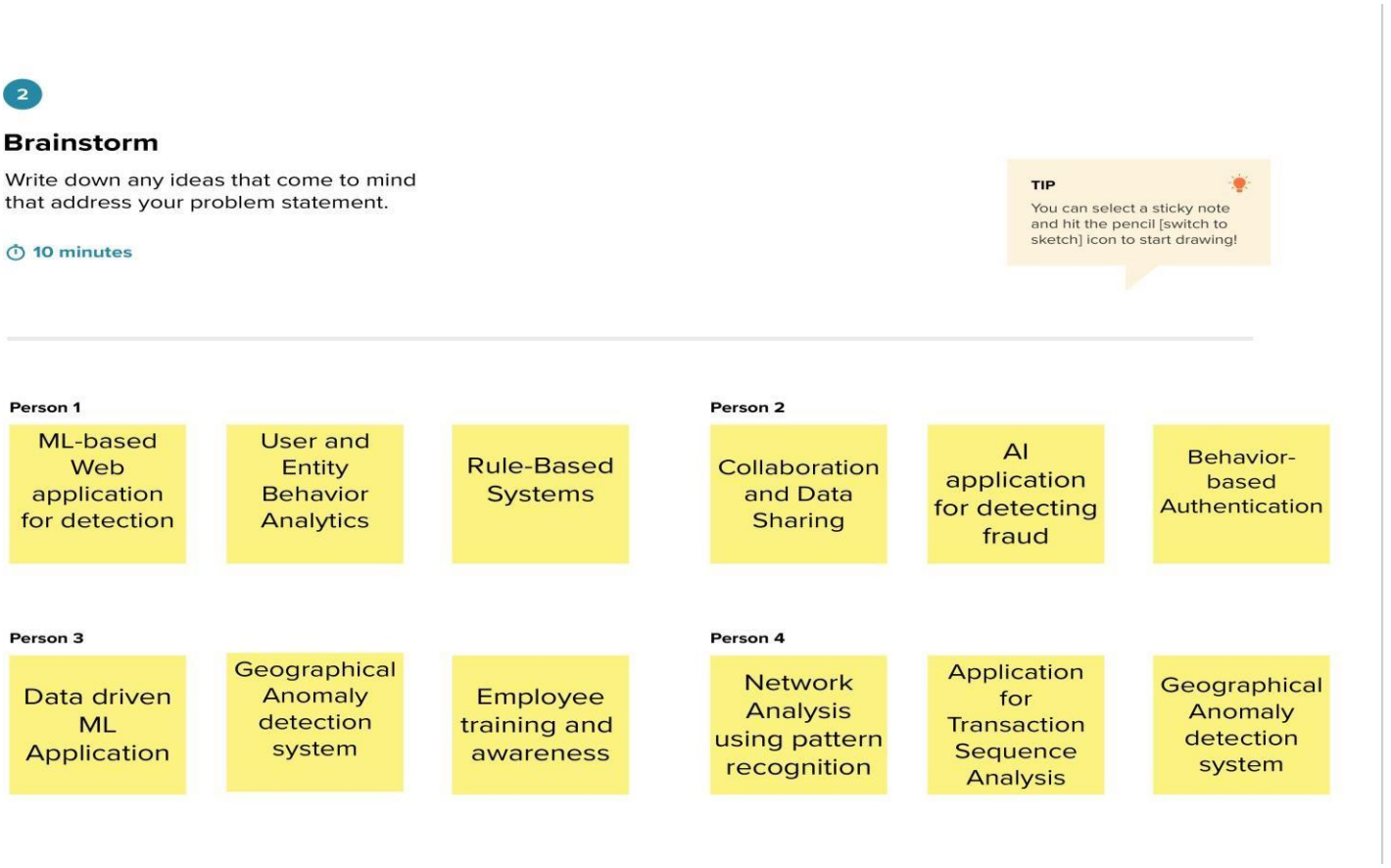
Brainstorm & Idea Prioritization Template:

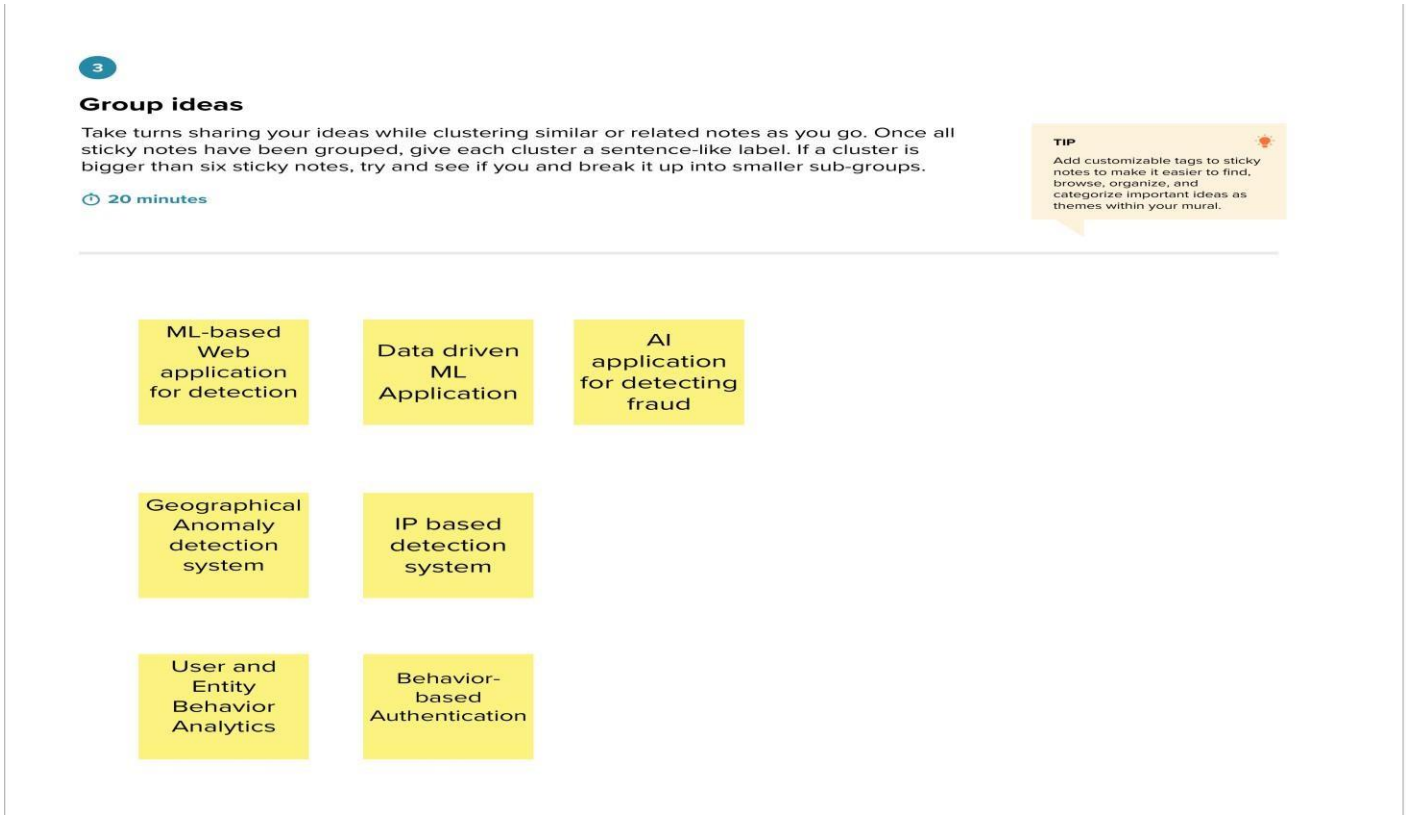
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate,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

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

Step-2: Brainstorm, Idea Listing and Grouping:





Step-3: Idea Prioritization

4

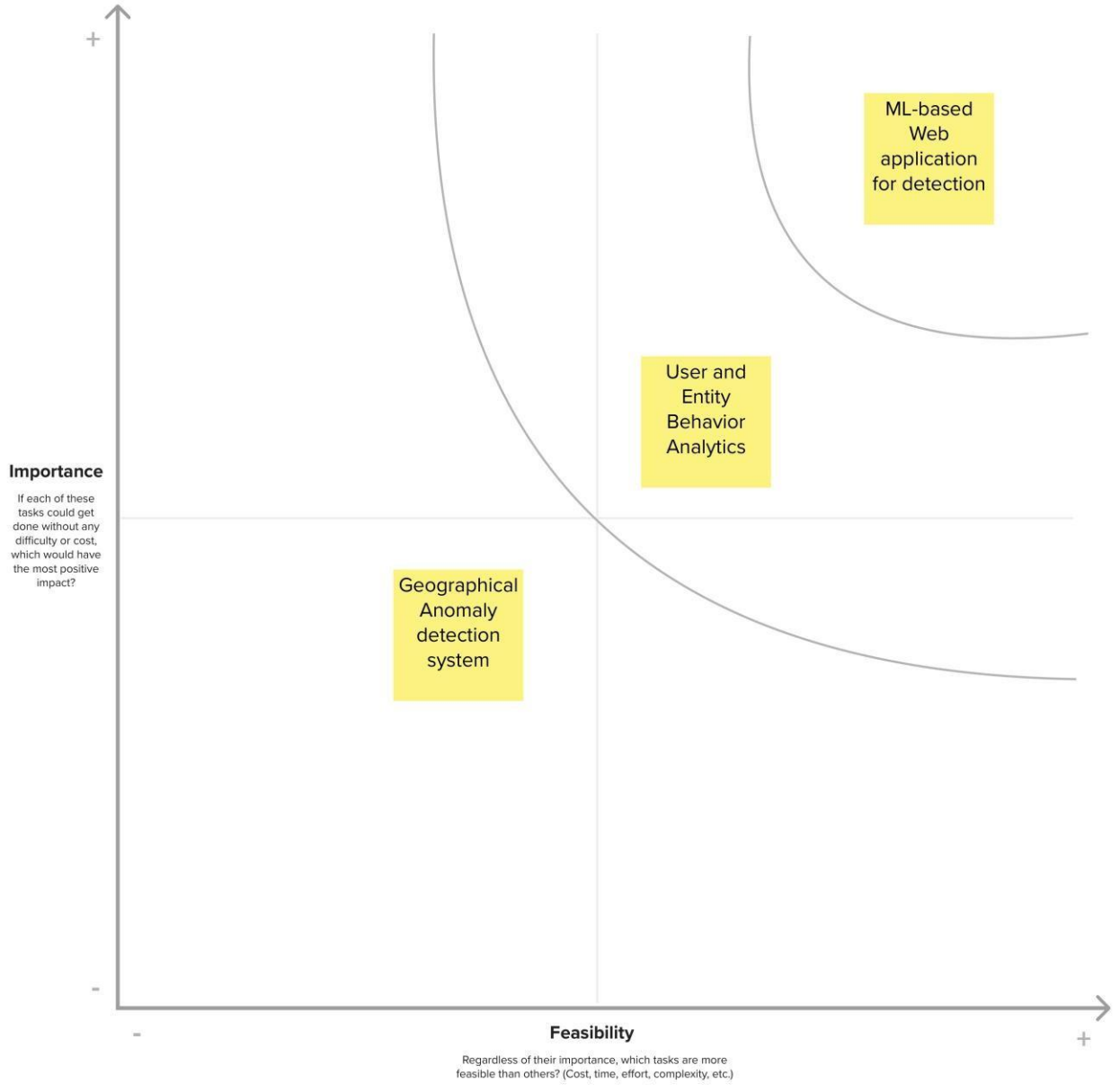
Prioritize

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

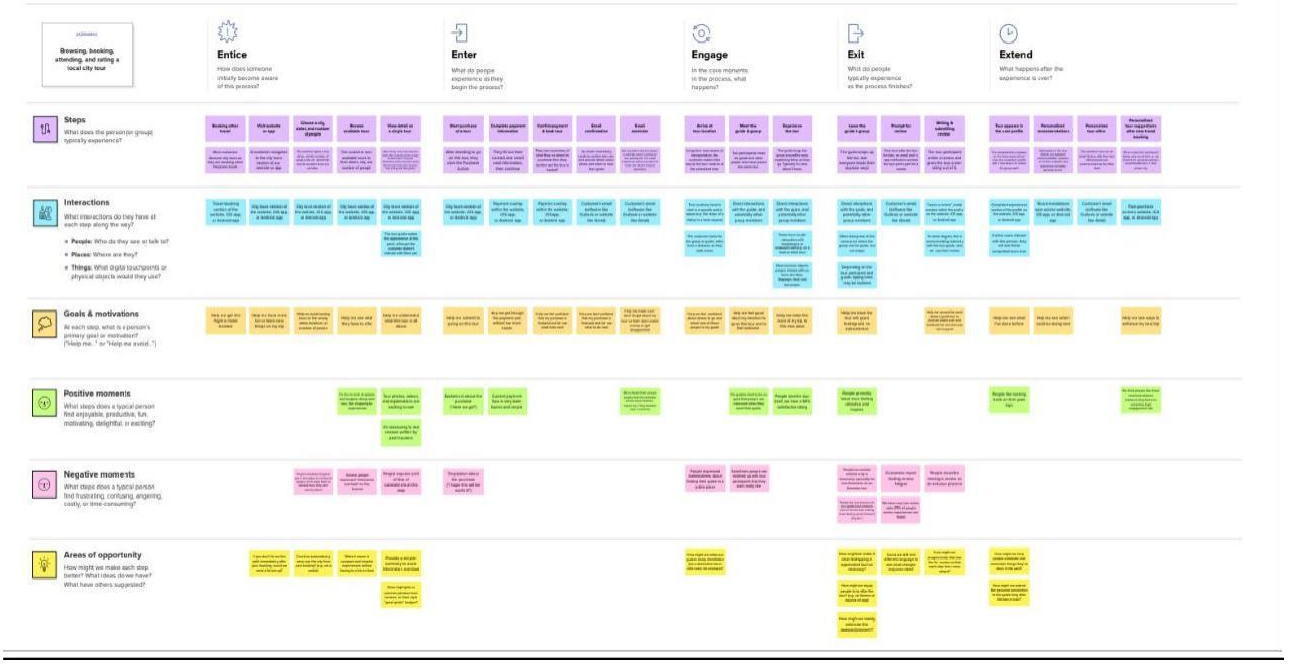
TIP

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.



3.REQIREMENT ANALYSIS

→ Customer Journey map:



→ Solution Requirement:

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

Date	31 January 2026
Team ID	LTVIP2026TMIDS61121
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

Sl. No	Requirement	Description
1	**User Authentication & Authorization**	The system must authenticate users such as administrators and fraud analysts. Role-based access control must restrict actions to authorized users only.
2	**Data Collection & Storage**	Data retrieval must be supported for analysis and reporting.

Non-functional Requirements:

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

Sl. No	Requirement	Description
1	**Performance**	The system must process high volumes of transactions in real-time with minimal response delay for fraud alerts.
2	**Scalability**	The architecture should support horizontal scaling to handle increased transaction loads without performance degradation.
3	**Security**	Ensure encryption of data in transit and at rest. Strong access control and audit trails must be maintained.
4	**Availability**	High system uptime with redundancy and failover mechanisms to minimize downtime.
5	**Compliance**	Must comply with regulations such as GDPR and PCI DSS. Regular audits and compliance reports are required.
6	**Usability**	Provide user-friendly interfaces for fraud analysts. Include user training and support.

→ Data Flow Diagram:

Project Design Phase-II

Data Flow Diagram & User Stories

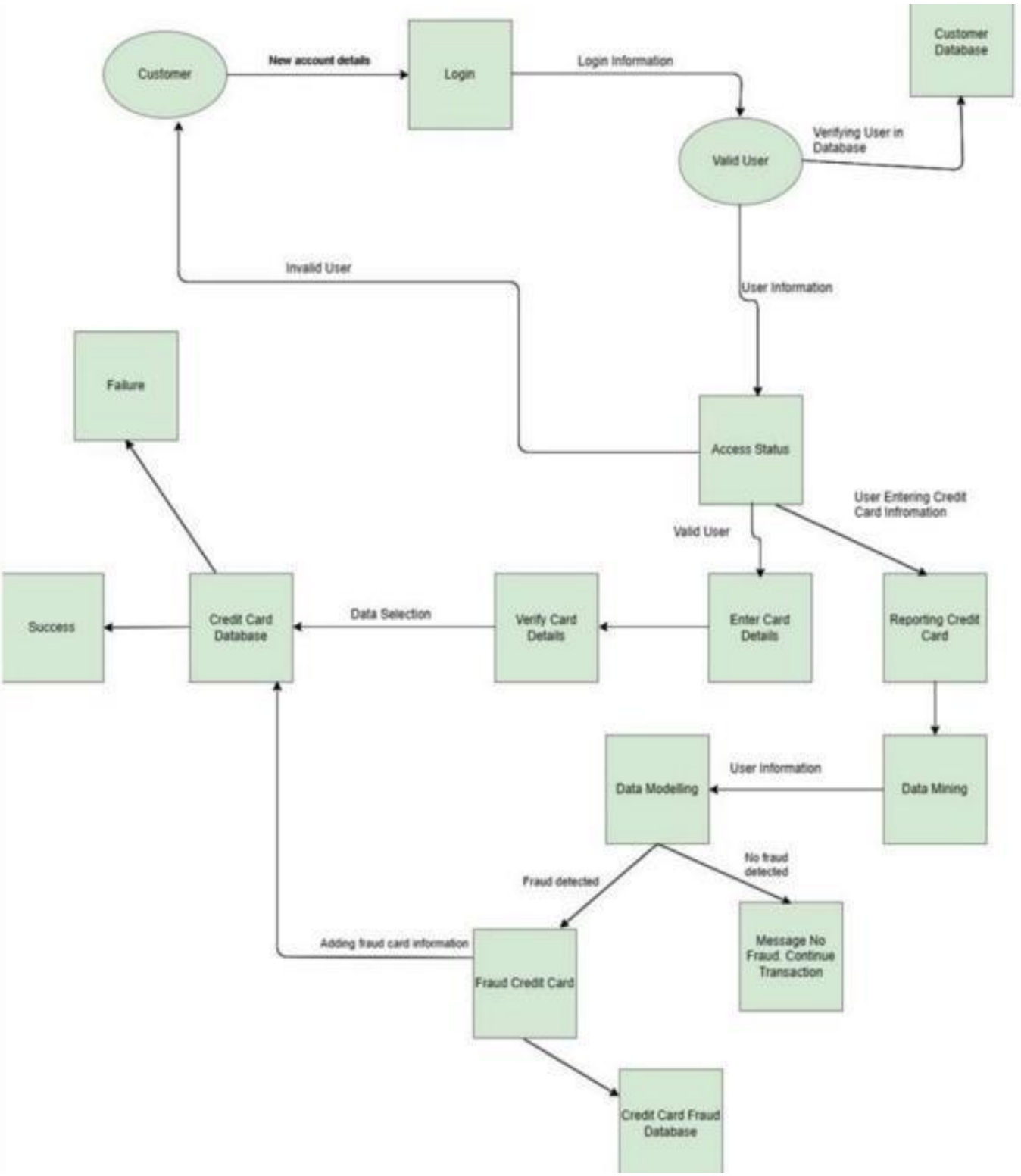
Date	1 February 2026
Team ID	LTVIP2026TMIDS61121
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	4 Marks

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

Example:



User Stories:

Sl. No	User Role	User Story	Expected Outcome
1	User	As a user, I want to open the home page to understand the purpose of the fraud detection system	User understands what the application does
2	User	As a user, I want to navigate to the prediction page	User can access the fraud prediction form
3	User	As a user, I want to enter transaction details like step, type, amount, balances	User can provide required inputs for prediction
4	User	As a user, I want to submit the form	Data is sent to Flask backend for processing
5	System	As a system, I want to take the input values and convert them into model format	Proper data preprocessing before prediction
6	System	As a system, I want to load the trained pickle model	Model is ready to predict fraud
7	System	As a system, I want to predict whether the transaction is fraud or not	Prediction result is generated
8	User	As a user, I want to see the result of the prediction	User knows if the transaction is Fraud / Not Fraud
9	User	As a user, I want the application to be simple and fast	Quick response without delays
10	Developer	As a developer, I want the model integrated with Flask	Real-time fraud detection through web app

→ Technology Stack:

Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	4 February 2026
Team ID	LTVIP2026TMIDS61121
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	4 Marks

Table 1: Components & Technologies

Sl. No	Component	Description	Technology Used in Your Project
1	User Interface	Web pages for Home, Predict and Result	HTML, CSS
2	Data Collection	Online payments fraud dataset	Kaggle Dataset
3	Data Preprocessing	Cleaning data, feature selection, splitting, visualization	Pandas, Numpy, Matplotlib, Seaborn, Scikit-learn
4	Application Logic	Handling routes, form data, prediction	Python Flask
5	Machine Learning Model	Fraud detection trained models	Decision Tree, Random Forest, SVM, XGBoost, Extra Tree
6	Model Storage	Saving trained model for reuse	Pickle (.pkl file)
7	Prediction API	Flask route to accept inputs and return prediction	Flask API (/pred route)
8	Evaluation	Checking model performance	Accuracy Score, Confusion Matrix, Classification Report
9	Infrastructure	Running the application	Local System (Anaconda / VS Code)

Table 2: Application Characteristics

Sl. No	Characteristics	Description	Technology Used
1	Open-Source Frameworks	Using open-source tools for building web app and ML model	Python, Flask, Scikit-learn, XGBoost, Pandas, Numpy
2	Simple Monolithic Architecture	Single Flask application handling UI, logic, and prediction	Flask (app.py)
3	Availability	Application runs locally and is accessible via browser	Localhost (127.0.0.1)
4	Performance	Fast prediction using pre-trained pickle model without retraining	Pickle (. pkl model), Numpy
5	User Interaction	Web form to take transaction inputs and display prediction	HTML, CSS Forms
6	Model Integration	ML model integrated into Flask for real-time prediction	Flask + Pickle Model
7	Lightweight Deployment	No server setup required, runs directly from Python environment	Anaconda / VS Code

4. PROJECT DESIGN

→ Problem Solution Fit:

Project Design Phase
Problem – Solution Fit Template

Date	7 February 2026
Team ID	LTVIP2026TMIDS61121
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	2 Marks

1. The Problem

In today's rapidly growing digital payment environment:

- Thousands of online transactions happen every second.
- Fraudulent transactions are increasing with new and evolving attack patterns.
- Traditional rule-based systems fail to detect modern fraud patterns effectively.
- Manual verification of transactions is slow, costly, and impractical.
- Users and organizations suffer financial losses due to delayed fraud detection.

There is a clear need for an automated, intelligent, and real-time fraud detection system.

2. The Solution

This project provides a Machine Learning–based Flask web application that:

- Uses a trained ML model on a Kaggle payment fraud dataset.
- Accepts transaction details from users through a web interface.
- Predicts instantly whether the transaction is Fraud or Not Fraud.
- Eliminates manual checking by automating fraud detection.
- Demonstrates how ML can be integrated into real-time payment security.

3. Behavioral Insights

- Online payments are increasing, and users expect secure transactions.
- Organizations prefer automated systems over manual verification.
- Fraud patterns change frequently, requiring adaptable ML models.
- Simple web interfaces improve usability and accessibility.

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

→ Proposed Solution:

Project Design Phase
Proposed Solution Template

Date	8 February 2026
Team ID	LTVIP2026TMIDS61121
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	2 Marks

Sl.No.	Parameter	Description
1	**Problem Statement (Problem to be solved)**	In the rapidly growing digital payments ecosystem, fraudulent transactions are increasing due to sophisticated attack patterns. Traditional rule-based or manual verification methods are slow, inefficient, and unable to detect complex fraud behaviors in real time. Organizations need an automated way to identify fraudulent transactions instantly to prevent financial loss.
2	**Idea / Solution Description**	This project presents a **Flask-based web application** integrated with a **Machine Learning model** trained on a Kaggle online payments fraud dataset. Users enter transaction details through a simple web form, and the system predicts in real time whether the transaction is **Fraud** or **Not Fraud** using a trained model (Random Forest/XGBoost).
3	**Novelty / Uniqueness**	✓ Integration of ML model with a live Flask web interface for instant predictions. ✓ Use of high-accuracy models like Random Forest and XGBoost for fraud detection. ✓ Clear separation of modules: Training, Model, and Web App. ✓ Demonstrates real-time practical use of ML in payment security.
4	**Social Impact / Customer Satisfaction**	The system highlights how technology can reduce financial fraud in digital transactions. By enabling instant fraud detection, it helps protect users and organizations from monetary loss and builds trust in online payment systems.
5	**Business Model (Revenue Model)**	• Although developed as an academic project, future scope may include: • **API-based Fraud Detection Service** for payment gateways and fintech apps. • **SaaS Model** for banks and e-commerce platforms to integrate fraud detection.

→ Solution Architecture:

Project Design Phase
Solution Architecture

Date	12 February 2026
Team ID	LTVIP2026TMIDS61121
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	2 Marks

➢ Overview

The solution architecture is designed as a simple, modular, and scalable ML web application that predicts whether a given online payment transaction is Fraud or Not Fraud in real time.

It cleanly separates Model Training**,** Model Serving (Flask)**,** and User Interface (HTML/CSS) for easy maintenance and extension.

➢ Goals of the Architecture

- Integrate a trained ML model into a live web app for instant predictions
- Keep training, model, and web layers independent
- Ensure fast response for real-time fraud checks
- Allow future deployment as an API/service for payment systems
- Maintain simplicity for academic demonstration and scalability for future use

➢ Architecture Layers:

1. Presentation Layer (Frontend)
2. Application Layer (Flask Backend)
3. Data & Model Layer (ML Training & Model Storage)

5. PROJECT PLANNING & SCHEDULING

→ Project Planning:

Project Planning Phase

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

Date	15 February 2026
Team ID	LTVIP2026TMIDS61121
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	5 Marks

1. Product Backlog and Sprint Schedule

Sprint	Functional Requirement (Epic)	User Story No.	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Understanding	USN-1	As a developer, I will study the Kaggle fraud dataset and understand all features	3	High	
Sprint-1	Data Preprocessing	USN-2	As a developer, I will clean and preprocess the dataset	5	High	
Sprint-1	Feature Engineering	USN-3	As a developer, I will encode type and select important features	3	High	
Sprint-2	Model Training	USN-4	As a developer, I will train ML models (DT, RF, SVM, XGB)	8	High	
Sprint-2	Model Evaluation	USN-5	As a developer, I will compare accuracies and select best model	5	High	
Sprint-2	Model Saving	USN-6	As a developer, I will save the trained model as model.pkl	2	Medium	
Sprint-3	Flask UI	USN-7	As a user, I can open the web page and enter transaction details	5	High	
Sprint-3	Flask Integration	USN-8	As a system, I will load model.pkl and predict Fraud/Not Fraud	8	High	
Sprint-3	Result Display	USN-9	As a user, I can see the prediction result on the result page	3	High	
Sprint-3	Testing	USN-10	As a developer, I will test the application with sample inputs	3	Medium	
Sprint-3	Documentation	USN-11	As a developer, I will prepare README and project documentation	2	Medium	

2. Project Tracker, Velocity & Burndown Chart:

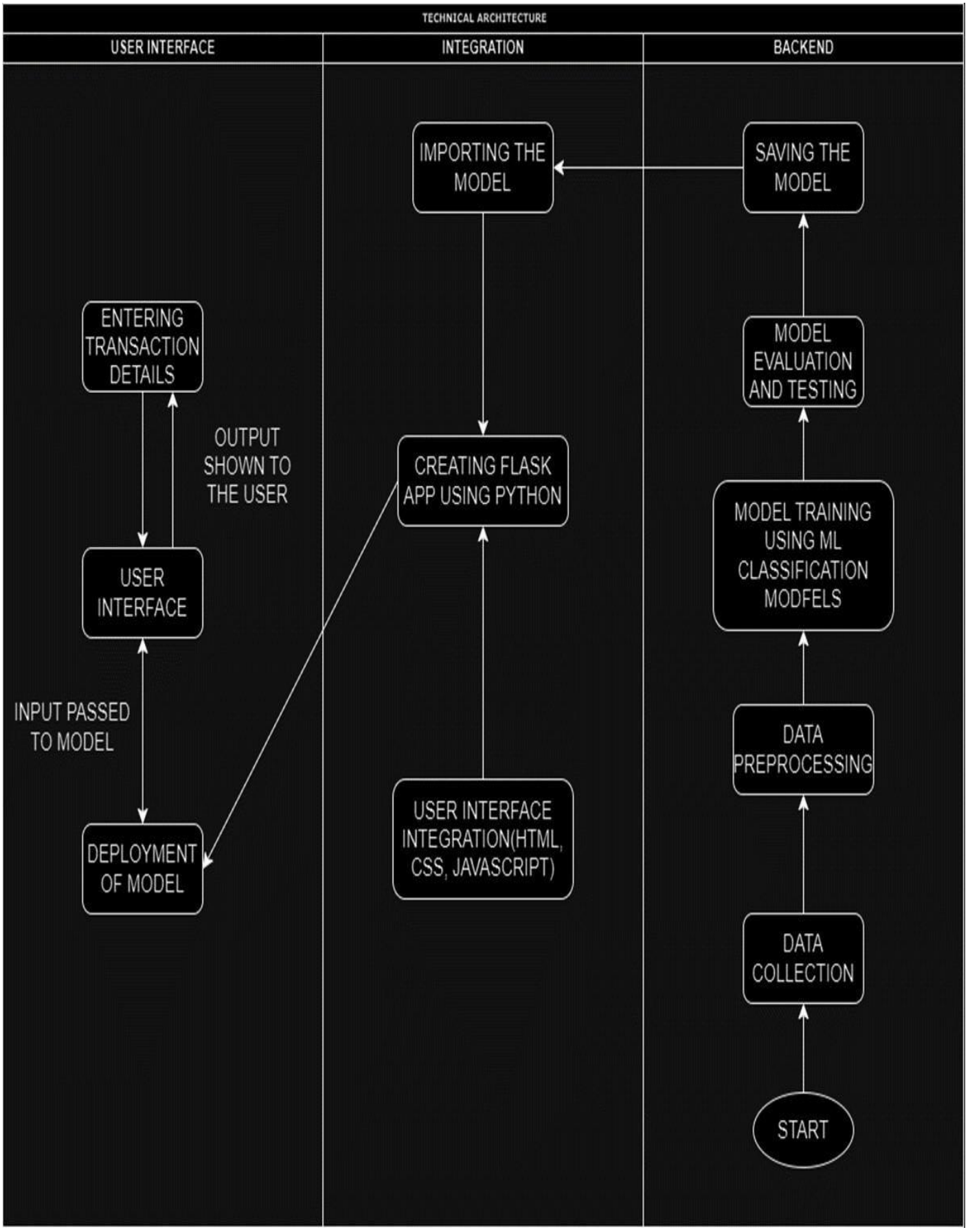
Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed**	**Sprint Release Date (Actual)**
Sprint-1	15	1 Day	15 Feb 2026	16 Feb 2026	15	15 Feb 2026
Sprint-2	15	1 Day	16 Feb 2026	17 Feb 2026	15	16 Feb 2026
Sprint-3	17	1 Day	17 Feb 2026	18 Feb 2026	17	17 Feb 2026

3. Velocity Calculation

- ✓ Given a 3-day sprint duration and the completed work across these sprints:
  - Total Story Points Completed: 47
  - Total Duration: 3 days
  - Average Velocity (AV) per day is calculated as:
  - Average Velocity = Total Story Points / Duration

    AV = 47 / 3 ≈ 16 story points per day

→TECHNICAL ARCHITECTURE:



6. FUNCTIONAL AND PERFORMANCE TESTING

→ Performance Testing:

User Acceptance Testing (UAT)

Date	18 February 2026
Team ID	LTVIP2026TMIDS61121
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	4 Marks

Project Overview

- Project Name: Online Payments Fraud Detection using Machine Learning
- Project Description: A Flask-based web app that predicts whether a transaction is Fraud or Not Fraud using a trained ML model from a Kaggle dataset.
- Project Version: 1.0
- Testing Period: 18 February 2026 to 20 February 2026

Testing Scope

Features and Functionalities Tested

- Home page loading
- Navigation to prediction page
- Transaction input form submission
- Model prediction using model.pkl
- Display of result page (Fraud / Not Fraud)
- Handling invalid or empty inputs

User Stories / Requirements Tested

- User can enter transaction details
- System predicts fraud status instantly
- User can clearly view the prediction result
- Smooth navigation between pages

Testing Environment

- URL/Location: http://127.0.0.1:5000
- Platform: Localhost (Flask Development Server)
- Browser Used: Chrome / Edge
- Backend: Python Flask
- Model: Random Forest (model.pkl)

Test Cases

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
TC-001	Home Page Load	Open 127.0.0.1:5000	Home page should load	Loaded successfully	Pass
TC-002	Navigate to Prediction	Click Predict option	Prediction form should open	Opened correctly	Pass
TC-003	Valid Transaction Input	Enter all fields → Submit	Result page with prediction	Fraud/Not Fraud shown	Pass
TC-004	Empty Input Handling	Submit without values	Error / validation message	Validation worked	Pass
TC-005	Multiple Predictions	Submit different values repeatedly	Quick response each time	No delay observed	Pass

Bug Tracking

Bug ID	Bug Description	Steps to Reproduce	Severity	Status	Additional Feedback
BG-001	Page reloads after prediction	Submit form	Low	Open	Can improve UX with AJAX
BG-002	No input range validation	Enter extreme values	Medium	Open	Add frontend validation
BG-003	Basic UI alignment issue	Open on small screen	Low	Open	Improve CSS responsiveness

Sign-Off:

- **Tester Name**: K.V.S. Naga Lokesh
- **Date**: 20 February 2026
- **Signature**: K.V.S.N. LOKESH

Notes

- All test cases covered positive and negative scenarios.
- Bugs logged with steps, severity, and current status.
- Project is ready for deployment, pending final sign-off from the project manager and product owner.

.

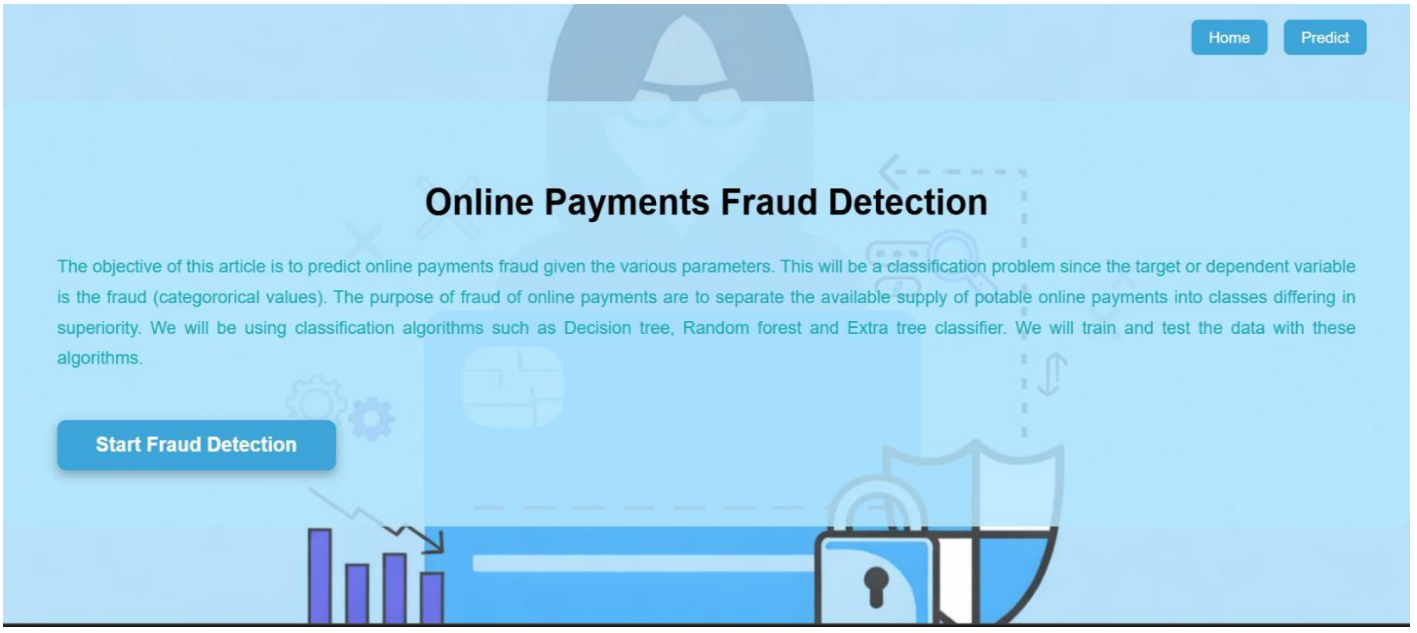
7. RESULTS

➢ OUTPUT SCREENSHOTS:

Project Implementation:

On completing the development part, we then run the application one last time to verify all the functionalities and look for any bugs in it. The user interface of the application looks a bit like the one's provided below.

- **LANDING PAGE:**



- **FORM PAGE (USER INPUT):**

Form: Display a form with fields for users to input information related to a transaction for fraud detection:

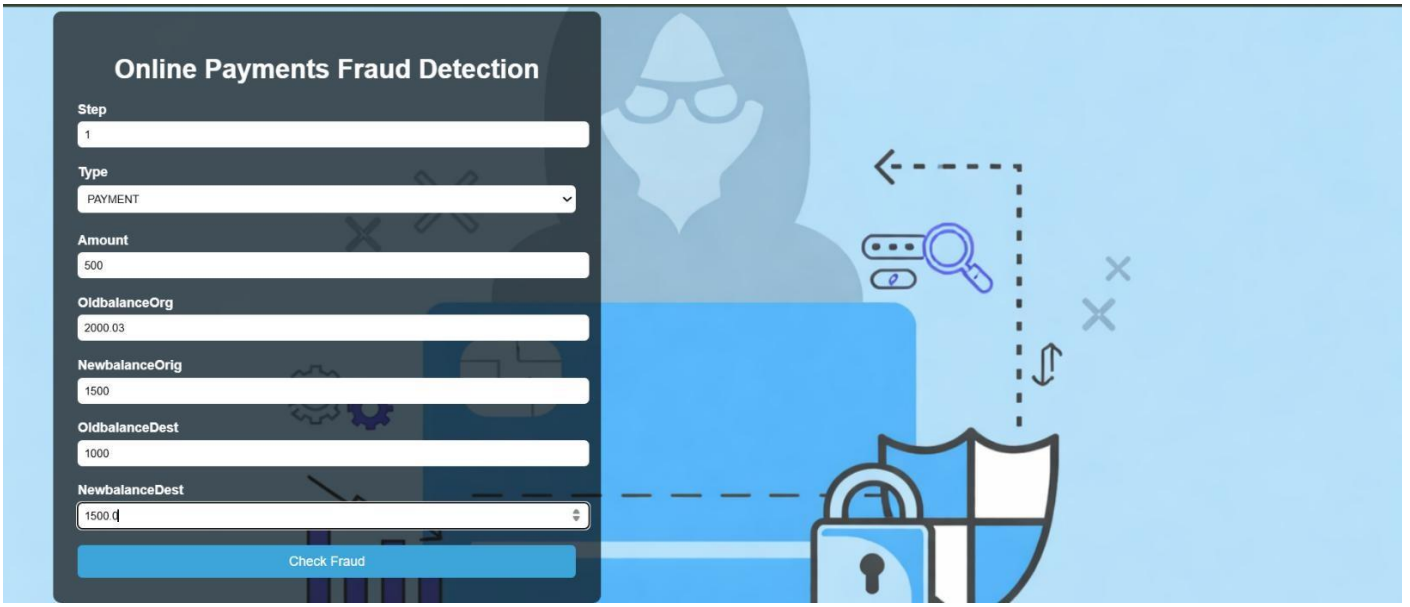
1. 'step': [Input Field]
2. 'type': [Input Field]
3. 'amount': [Input Field]
4. 'oldbalanceDest': [Input Field]
5. 'newbalanceDest': [Input Field]

6. Submit Button: Provide a "Submit" button that allows the user to submit the transaction details for fraud detection.

Result Display: After submitting the form, display the result on the same page. It may indicate whether the provided transaction information is classified as potential fraud or not. For example:

→ "Result: No Fraud Detected" or "Result: Possible Fraud Alert"

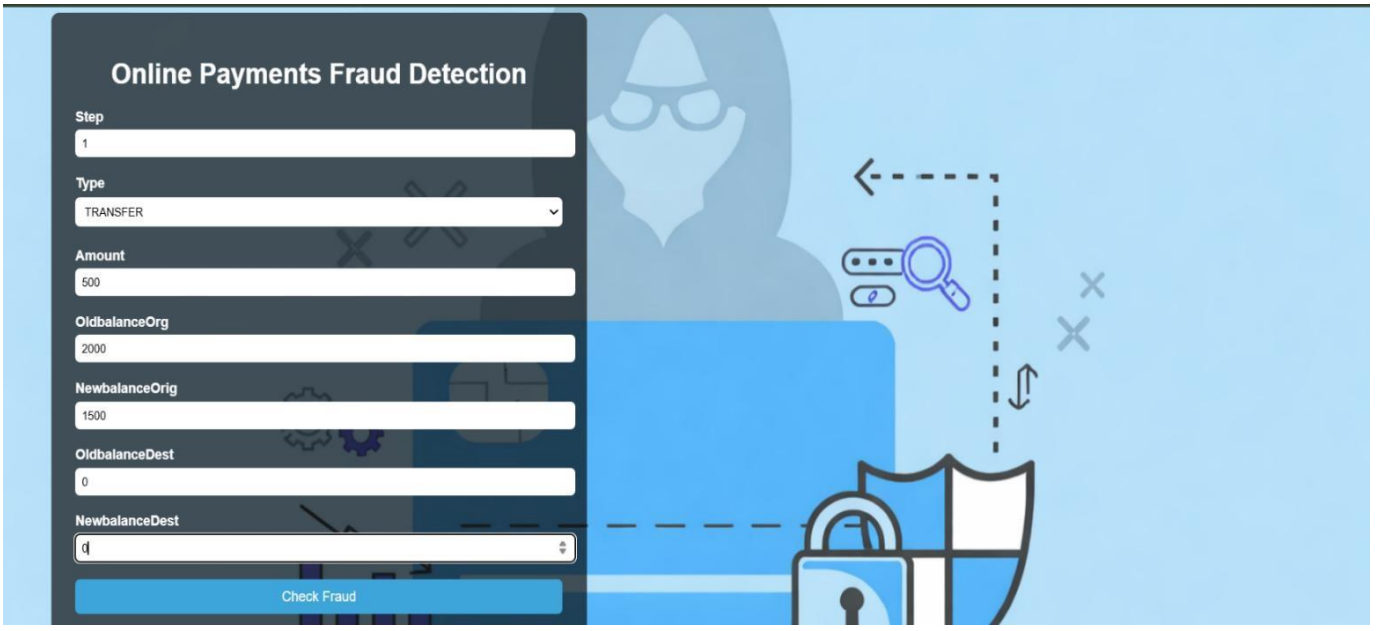
1. NO FRAUD DETECTED:



➢ When the system classifies a transaction as "Not a Fraud," it means that the provided transaction details do not exhibit suspicious or fraudulent behavior.

- Users can be reassured that the transaction appears legitimate, and they can proceed with confidence.

2. FRAUD DETECTED:



- When the system classifies a transaction as "Fraud," it indicates that the provided transaction details raise suspicions of fraudulent activity.

- Users should be alerted to the potential risk and advised to take immediate action to secure their accounts and prevent further damage.

8. ADVANTAGES AND DISADVANTAGES

Advantages:

1. **Improved Security**

   The system enhances transaction security by identifying and preventing fraudulent activities, thereby reducing financial losses and protecting the organization's reputation.

2. **Real-Time Fraud Detection**

   Fraudulent transactions are detected instantly at the time of processing, allowing immediate action to block suspicious activities before damage occurs.

3. **Cost Savings**

   By preventing fraud, the system minimizes losses related to chargebacks, refunds, and stolen funds, resulting in significant cost savings for businesses.

4. **Customer Trust and Satisfaction**

   Accurate fraud detection ensures that genuine transactions are not wrongly declined, improving customer experience and building trust.

5. **Operational Efficiency**

   Automation through machine learning reduces the need for manual fraud reviews, saving time and effort for fraud analysis teams.

6. **Customization and Adaptability**

   The system can be tuned and updated according to evolving fraud patterns by adjusting model parameters and retraining with new data.

7. **Regulatory Compliance**

   Helps organizations meet compliance requirements such as PCI-DSS and other financial security regulations.

8. **Insightful Data Analysis**

   Provides analytical reports and patterns of fraudulent behavior, helping in strengthening future fraud prevention strategies.

Disadvantages:

1. **False Positives**

   Legitimate transactions may sometimes be incorrectly flagged as fraudulent, which can lead to customer dissatisfaction and potential revenue loss.
2. **Resource Intensive**

   Developing and maintaining a fraud detection system requires skilled personnel, significant computational resources, and continuous monitoring.
3. **System Complexity**

   The setup, configuration, and tuning of machine learning models and infrastructure can be complex and may require a steep learning curve.
4. **Cost of Implementation**

   Initial deployment costs, including hardware, software, cloud services, and staffing, can be relatively high.
5. **Adapting to New Fraud Techniques**

   Fraudsters constantly change their methods, requiring frequent updates, retraining of models, and rule adjustments.
6. **Data Privacy and Security Concerns**

   Handling sensitive financial and personal data demands strict data protection policies and compliance with privacy regulations.
7. **Potential Downtime**

   System updates, maintenance, or failures may temporarily affect transaction processing and service availability.
8. **Training Requirements**

   Staff must be trained to understand, monitor, and manage the system effectively, which can be time-consuming.

9. CONCLUSION

This project addresses the critical need for **online payment fraud detection** in the rapidly growing e-commerce ecosystem. By leveraging **machine learning techniques** and real-time anomaly detection, the system provides a proactive and effective mechanism to secure financial transactions. Strengthening these security measures helps maintain the **trust, reliability, and safety** of digital payment platforms for users and businesses alike.

Multiple machine learning algorithms were implemented and evaluated on the given dataset to identify the most suitable model for deployment. The models tested include **Random Forest, Decision Tree, Support Vector Machine (SVM), and XGBoost**. The observed accuracies were:

- **Random Forest** – 99.97%
- **Decision Tree** – 99.96%
- **SVM** – 80%
- **XGBoost** – 99.97%

Based on comparative performance, **Random Forest** was selected as the best-fit model for this dataset due to its **high accuracy, stability, and reliable prediction capability**.

10. FUTURE SCOPE

1. Phishing Scams:

In a common phishing scam, individuals receive seemingly legitimate emails or messages that lead them to fake websites designed to steal their login credentials or credit card information. Such scams have led to unauthorized transactions and identity theft.

2. Credit Card Skimming:

Criminals install skimming devices on ATMs or card readers at gas stations and retail stores. These devices capture card details and PINs, which are then used to make unauthorized purchases or withdrawals.

3. Unauthorized Subscription Charges:

Some businesses may engage in unethical practices by signing up users for subscriptions without their knowledge or consent, resulting in recurring charges to their credit cards.

4. Seller Fraud in Online Marketplaces:

In e-commerce platforms, fraudulent sellers may list products that don't exist or send counterfeit goods after receiving payment. This leaves buyers at a loss with no way to get their money back.

5. Stolen Payment Information:

Hackers breach the security of organizations, gaining access to vast databases of credit card information. This information is then sold on the dark web or used to make unauthorized purchases.

6. Ransomware Attacks:

Ransomware attacks can encrypt a victim's data and demand a ransom to provide the decryption key. These attacks can target individuals or businesses, causing significant financial harm.

RUNNING THE APPLICATION

✓ Start the server:

1. Run Flask
2. Python app.py

✓ Server runs at:

http://127.0.0.1:5000

11. APPENDIX

- Video Demo Link:

https://drive.google.com/drive/folders/1z81Lfh6clpPSn3WnHNXUaaWSygvUz88y

- Project Links:

https://drive.google.com/drive/folders/10UTjdwDCXMZGaekMi8mHwCMTx_YCLKmJ

- Drive Link:

https://drive.google.com/drive/folders/1skJYT59yNn8cndJrCy2ZkgNVtoAsucrZ

- Git Hub Link:

https://github.com/lokesh-143-sai/ONLINE-PAYMENTS-FRAUD-DETECTION-USING-MACHINE-LEARNING

- Dataset Link:

https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset