

Project Design Phase-I
Solution Architecture

Date	10 November 2023
Team ID	Team-591594
Project Name	Online Payments Fraud Detection using ML
Maximum Marks	4 Marks

Solution Architecture:

◆ **Best tech solution to solve existing problems:**

- ❖ To combat online payment fraud, a Flask app integrates machine learning for real-time transaction analysis, utilizing algorithms like Random Forest. This secure and scalable solution flags potential fraud and incorporates a feedback loop for continuous enhancement.

◆ **The structure, characteristics, behavior, and other aspects of the software to project stakeholders:**

❖ **Machine Learning Model:**

➤ **Structure:**

- Incorporates user feedback to enhance the model's accuracy through a dynamic feedback loop.
- Flask app serves as a user interface for interacting with the fraud detection system.

➤ **Characteristics:**

- Trained on the Online Payments Fraud Detection dataset from Kaggle.
- Can recognize fraud payments

➤ **Behaviour:**

- Processes input data of fraud payments and produces corresponding predictions.
- The system continuously analyzes incoming transactions in real-time.
- Track patterns and anomalies, enabling prompt identification of potentially fraudulent activities.

❖ **Web Application:**

➤ **Structure:**

- Created with the Flask framework.
- Comprsed of three HTML pages: index.html, logout.html. and, prediction.html

➤ **Characteristics:**

- An interface that is easy to use for interaction.
- Integration with the machine learning model is seamless.

➤ **Behaviour:**

- Allows users to input fraud data for prediction.
- Displays predictions on the prediction.html page.
- Results are presented on the logout.html page.

❖ **CSS and JS files:**

➤ **Structure:**

- JS to improve the web application's functionality and interactivity.
 - CSS is used to style HTML pages.
 - **Characteristics:**
 - JS is used to manage dynamic elements and provide seamless interactions.
 - Used CSS to enhance the user experience and visual attractiveness.
 - **Behaviour:**
 - A visually pleasing and consistent design is guaranteed by CSS.
 - The web application's dynamic features and responsiveness are enhanced by the use of JS.
 - The system dynamically evolves, ensuring resilience against new threats in online payments.
- ◆ **Features, development phases, and solution requirements:**
- ❖ **Features:**
 - Identifies unusual patterns or behaviors in transactions
 - The architecture is designed to handle a high volume of transactions in real-time.
 - Flags transactions deviating from the expected norm for further investigation.
 - ❖ **Development Phases:**
 - Including the trained model in the web application.
 - Flask app development.
 - Model training and evaluation.
 - Data collection and preprocessing.
 - ❖ **Solution Requirements:**
 - Online Payments Fraud Detection dataset.
 - Python with libraries like TensorFlow, Flask, etc.
 - HTML, CSS, and JS for web app development.
- ◆ **Specifications according to which the solution is defined, managed, and delivered:**
- ❖ **Data Specifications:**
 - Online Payments Fraud Detection dataset from Kaggle.
 - ❖ **Model Specifications:**
 - Flask app serves.
 - Training and evaluation metrics.
 - ❖ **Web App Specifications:**
 - Flask framework.
 - Three HTML pages (index, prediction, logout).
 - CSS and JS for design and functionality.
 - ❖ **Delivery:**
 - Documentation for interested parties.
 - Packaged and deployed web app.
 - A deployed trained model.

Solution Architecture Design:

