Project Initialization and Planning Phase

Date	20 July 2025
Team ID	SWUID20250184320
Project Title	Online Fraud Payment Detection
Maximum Marks	3 Marks

Project Proposal (Proposed Solution)

To effectively detect fraudulent transactions in real-time and minimize financial loss, this project proposes the development of a **Fraud Detection System** using **machine learning algorithms** such as Random Forest, XG Boost, and Support Vector Machines (SVM). The solution will involve data preprocessing, model training, evaluation, and deployment of the best-performing model.

The system will analyse patterns in transaction data and identify anomalies that indicate fraud. It will be trained on a labelled dataset containing both fraudulent and legitimate transactions, allowing it to learn distinguishing characteristics.

Once deployed, the model will accept transaction inputs, process them through the trained pipeline, and return a prediction — whether the transaction is fraudulent or not — in near real-time.

Project Overview	
Objective	To build a machine learning–based fraud detection system capable of identifying suspicious transactions in real-time, thereby reducing financial loss and enhancing user trust in digital payment platforms.
Scope	This project focuses on developing a classification model trained on historical transaction data to detect fraudulent activities. It includes data preprocessing, model training, performance evaluation, and optional deployment with a user-friendly interface, limited to binary classification (fraudulent vs. legitimate) and does not involve multi-class fraud categorization or network-based fraud detection.
Problem Statement	
Description	In the age of digital transactions, users, especially regular digital payment users and loyal online banking customers — are increasingly vulnerable to fraudulent activities due to delayed detection systems. Current systems often flag fraud after the damage is done, leading to loss of funds, mental distress, and a lack of trust in digital platforms.

Impact	Solving this issue will empower users by ensuring safer financial transactions, minimizing financial loss, and rebuilding trust in digital payment systems. It also supports the financial ecosystem by proactively reducing fraudulent incidents.
Proposed Solution	
Approach	• **Data Collection & Preprocessing**: Clean and prepare the dataset (handle missing values, encode categorical data, normalize features). • **Model Development**: Use and compare algorithms such as Random Forest, XG Boost, SVM, and Decision Trees. • **Model Evaluation**: Evaluate using metrics like precision, recall, F1-score, confusion matrix, and AUC-ROC. • **Model Deployment**: Save the best-performing model and (optionally) deploy it using Streamlit /Flask or Render for user interaction. • **Documentation**: Properly document code, performance, and limitations.
Key Features	• **Real-Time Fraud Detection**: Detects fraudulent transactions as they occur using trained ML models. • **Multi-Model Comparison**: Implements and evaluates various ML models such as Random Forest, XG Boost, and SVM. • **Performance Evaluation**: Uses precision, recall, F1-score, and ROC-AUC to determine the best-performing model. • **Data Preprocessing Pipeline**: Handles missing values, feature scaling, and encoding efficiently. • **Model Saving & Loading**: Saves the best model for future predictions or deployment. • **Interactive Interface (Optional)**: (If deployed) Allows users to input transaction details and receive fraud risk predictions in real-time. • **Detailed Documentation**: Includes structured code, comments, and visualizations to explain performance.

Resource Requirements

Resource Type	Description	Specification/Allocation
Hardware		
Computing Resources	CPU for model training & testing	AMD Ryzen 3 (dual/quad core, 3.5 GHz approx.)
Memory	RAM for dataset processing	8 GB DDR4
Storage	Local disk space for data, models, logs	512 GB
Software		
Frameworks	Python frameworks	Flask / Streamlit/ Render
Libraries	ML and data processing	scikit-learn, xg boost, pandas, NumPy, matplotlib, seaborn
Development Environment	IDE and version control system	Jupyter Notebook, VS Code, Git, GitHub Desktop
Data		
Data	Source, size, format	Kaggle dataset (e.g., "Online Payments Fraud Detection Dataset "), ~284,807 rows, CSV format