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

Date	8 February 2026
Team ID	LTVIP2026TMIDS76029
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	4 Marks

Technical Architecture:

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

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	**User Interface**	Web interface where user enters transaction details and views fraud prediction result.	HTML, CSS, Bootstrap
2	**Application Logic-1**	Handles form submission, input validation, and routing between pages (Home, Predict, Submit).	Python (Flask Framework)
3	**Application Logic-2**	Loads trained ML model and processes transaction input data.	Python (Scikit-learn, NumPy, Pandas)
4	**Application Logic-3**	Performs prediction logic and returns Fraud / Not Fraud result to UI.	Flask Backend + payments.pkl Model
5	**Database**	Stores historical transaction dataset used for training and analysis.	CSV Dataset (PS_20174392719_1491204439457_logs.csv)
6	**Cloud Database (Optional)**	Can be extended to store transaction logs in cloud for future scalability.	AWS RDS / MongoDB Atlas (Future Enhancement)

S.No	Component	Description	Technology
7	**File Storage**	Stores trained ML model file used for prediction.	Local File System (payments.pkl)
8	**External API-1**	Not used in current implementation. (System works independently without third-party APIs.)	N/A
9	**External API-2**	Not applicable in current system.	N/A
10	**Machine Learning Model**	Predicts whether a transaction is Fraud or Not Fraud using trained classification algorithms.	Decision Tree, Random Forest, SVM, Extra Trees (Scikit-learn)
11	**Infrastructure (Server / Cloud)**	Application deployed locally using Flask server. Can be deployed on cloud for production use.	Local System (Python 3.8+), Flask Server

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	**Open-Source Frameworks**	The application is built completely using open-source technologies for development, machine learning, and deployment.	Python, Flask, Scikit-learn, Pandas, NumPy, Matplotlib
2	**Security Implementations**	The system validates user input before processing. The trained model file (payments.pkl) is stored securely in the backend. Application runs on localhost server and can be secured using HTTPS in production.	Flask Input Validation, Python Security Practices, HTTPS (Future Deployment)
3	**Scalable Architecture**	The project follows a simple 3-layer architecture: Presentation Layer (UI), Application Layer (Flask Backend), and ML Model Layer (Prediction Engine). It can be extended to cloud deployment for handling more users.	Flask (Backend), Scikit-learn Model, 3-Tier Architecture Design

S.No	Characteristics	Description	Technology
4	**Availability**	The application is available whenever the Flask server is running. It can be deployed to cloud platforms for higher uptime and continuous availability.	Local Flask Server, Can be deployed on AWS / Heroku (Future Enhancement)
5	**Performance**	The trained ML model is preloaded to ensure fast real-time predictions. The system processes input instantly with minimal delay. Dataset preprocessing is done offline to improve runtime speed.	Scikit-learn Optimized Model, Pre-trained payments.pkl File