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

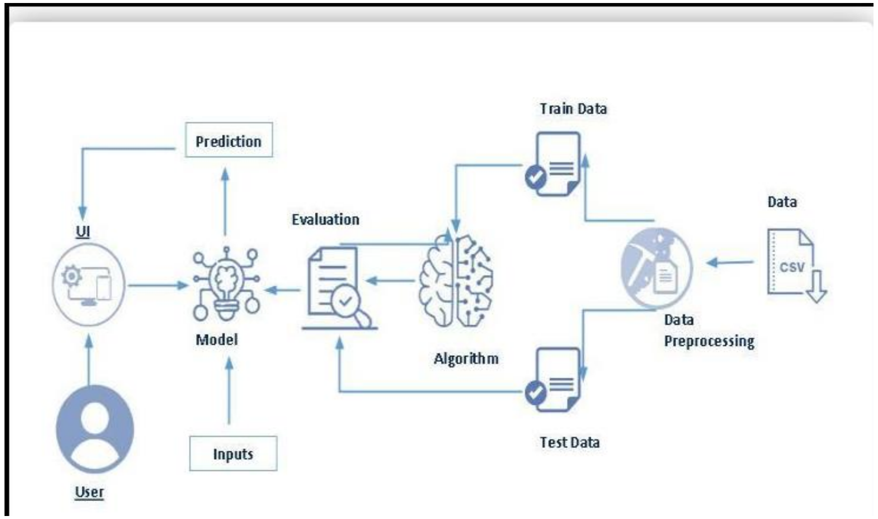
Date	31 January 2026
Team ID	LTVIP2026TMIDS90948
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	4 Marks

Technical Architecture:

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

Example: Order processing during pandemics for offline mode

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



Guidelines:

- Include all the processes (As an application logic / Technology Block)
- Provide infrastructural demarcation (Local / Cloud)
- Indicate external interfaces (third party API’s etc.)
- Indicate Data Storage components / services
- Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web-based dashboard for inputting transaction data and viewing risk reports.	HTML5, CSS3
2.	Application Logic-1	Server-side logic for data processing, routing, and feature engineering.	Python 3.x, Flask Framework
3.	Application Logic-2	Real-time automated decision logic (Thresholding for Risk levels).	Python (Conditional Logic)
4.	Machine Learning Model	High-performance fraud classifier optimized for 99% F1-Score.	XGBoost & Random Forest
5.		.	
6.			
7.			
8.			
9.			
10.			
11.			

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilization of robust, community-backed libraries for ML and Web.	Scikit-learn, XGBoost, Flask, Pandas
2.	Security Implementations	Enhancing data privacy by dropping sensitive identifiers (nameOrig, nameDest) and using the	Data Anonymization, Flask (POST method)

S.No	Characteristics	Description	Technology
		POST method to securely transfer transaction details from the UI to the backend.	
3.	Scalable Architecture	A modular **3-Tier Architecture** that separates the frontend (HTML), the application logic (Flask), and the prediction engine (XGBoost).	Client-Server-Model (3-Tier)
4.	Availability	Implementation of a persistent web server that can be accessed via a browser to perform fraud checks at any time.	Flask Local Server
5.	Performance	High-speed processing enabled by serializing the trained model to allow instant loading and sub-millisecond predictions.	Pickle (.pkl), Vectorized NumPy operations

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