

## Project Design Phase Solution Architecture

Date	14 February 2026
Team ID	LTVIP2026TMIDS42974
Project Name	Online Payments Fraud Detection Using Machine Learning
Maximum Marks	4 Marks

### Overview

The solution architecture bridges business goals (reducing online payment fraud and protecting customers/financial institutions) and the technical implementation (data preprocessing, machine learning models, and API-based integration). The architecture defines system components, transaction flow, interfaces, and deployment considerations.

### Key Goals

- Provide accurate real-time fraud detection for online transactions
- Minimize false positives while maximizing fraud detection rate
- Ensure modularity across preprocessing, model, and API layers
- Enable scalability for high transaction volumes
- Maintain reproducibility via saved artifacts (model, scaler, encoder)

### Architecture Components

**Frontend (UI):** Web dashboard for monitoring transactions and viewing fraud alerts.

**Backend (API Layer):** Flask/FastAPI server handling transaction requests, validation, and routing.

**Preprocessing Layer:** Imputer, Encoder, Scaler, and feature engineering modules to prepare transaction data consistently.

**ML Model Layer:** Trained fraud detection classifier (Logistic Regression / Random Forest / XGBoost).

**Database & Artifacts:** Transaction records stored in database; serialized ML model files (.pkl/.joblib) for inference.

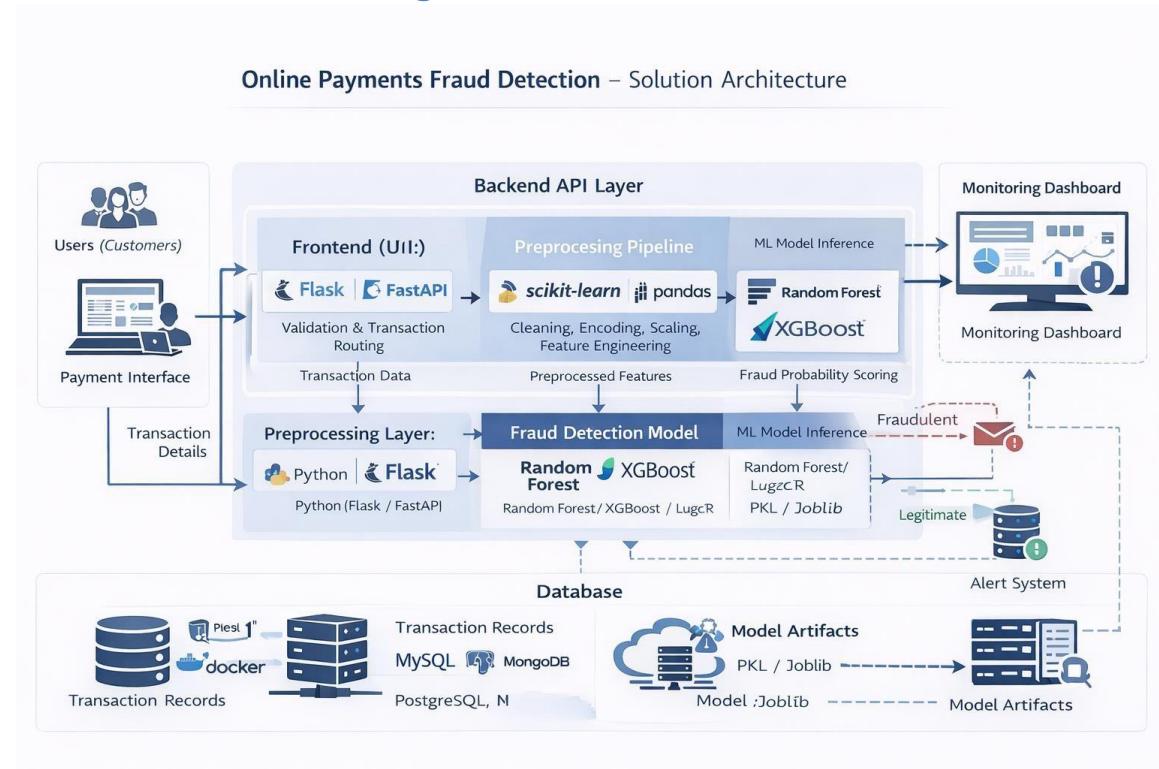
**Alert System:** Notification module for flagging suspicious transactions.

### Data Flow

1) User initiates an online payment transaction.

- 2) Backend API receives transaction details and validates inputs.
- 3) Transaction data passes through preprocessing (cleaning, encoding, scaling, feature generation).
- 4) Preprocessed features are sent to the ML model for fraud probability prediction.
- 5) System classifies transaction as Fraud or Legitimate.
- 6) Result is stored in database and alert is triggered if fraud is detected.

## Solution Architecture Diagram



## Non-Functional Considerations

- **Performance:** Real-time low-latency fraud prediction (<1 second per transaction)
- **Reliability:** Consistent preprocessing using saved artifacts
- **Security:** Encrypted transaction handling, input validation, authentication
- **Maintainability:** Modular and well-documented architecture
- **Scalability:** Cloud-ready design supporting high transaction loads
- **Deployability:** Compatible with Docker, Cloud VMs, and container orchestration in future.