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

Date	11 February 2026
Team ID	LTVIP2026TMIDS54415
Project Name	Online Payment Fraud Detection using Machine Learning
Maximum Marks	4 Marks

Technical Architecture:

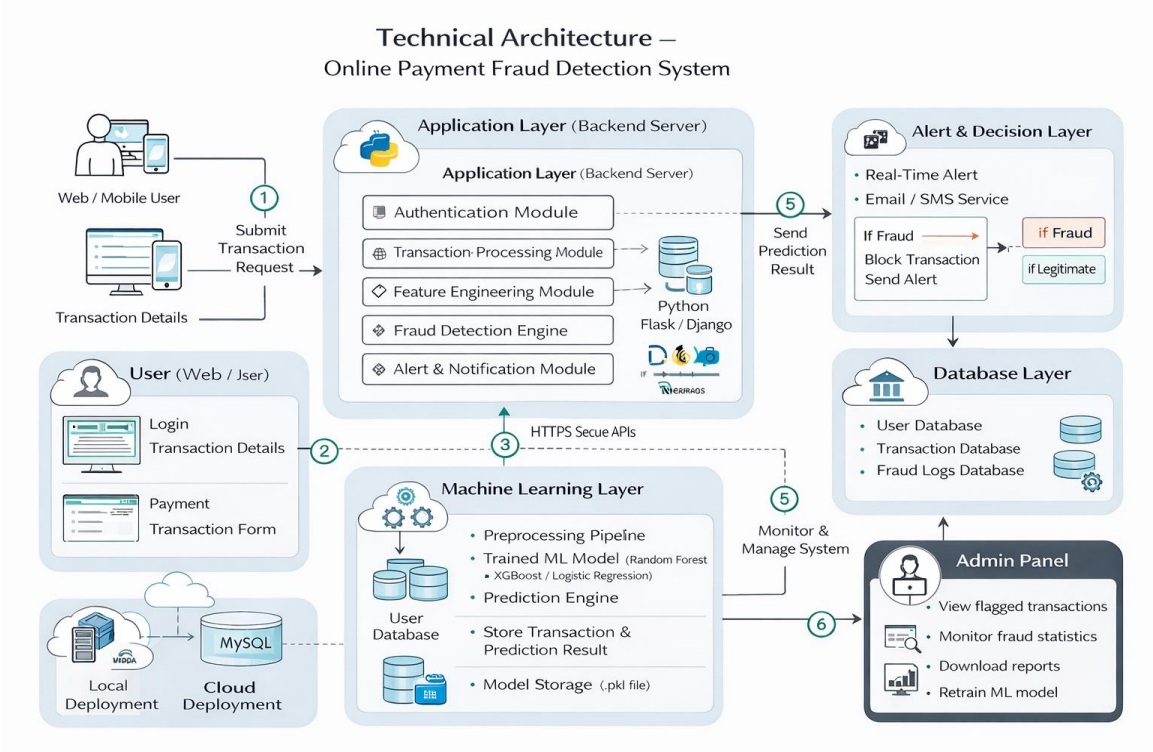

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Web-based interface for users and admin to perform transactions and view fraud alerts	HTML, CSS, JavaScript, Bootstrap / React JS
2	Application Logic-1	Transaction processing and validation logic	Python (Flask / Django)

3	Application Logic-2	Fraud detection logic using trained ML model	Python (Scikit-learn, Pandas, NumPy)
4	Database	Store user details, transaction history, fraud logs	MySQL / PostgreSQL
5	Cloud Database	Cloud-based storage for scalable transaction data	AWS RDS / MongoDB Atlas
6	File Storage	Storage of datasets, trained ML model (.pkl), reports	Local File System / AWS S3
7	External API-1	Payment gateway integration for transaction validation	Razorpay API / Stripe API
8	Machine Learning Model	Predict fraudulent vs legitimate transactions	Random Forest / Logistic Regression / XGBoost
9	Infrastructure (Server / Cloud)	Application deployment environment	Local Server (Windows/Linux), AWS EC2 / Heroku

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Frameworks used for backend, ML, and frontend	Flask, Scikit-learn, Pandas, NumPy, Bootstrap
2	Security Implementations	Encryption, authentication, secure APIs, input validation	HTTPS, SHA-256 Password Hashing, JWT Authentication, OTP Verification
3	Scalable Architecture	3-Tier Architecture (UI → Application → Database) supporting future microservices scaling	Flask (API Layer), MySQL, AWS EC2
4	Availability	System available 24/7 with minimal downtime using cloud deployment	AWS EC2 / Load Balancer
5	Performance	Real-time fraud detection with millisecond-level prediction, optimized queries	Scikit-learn optimized model, Indexed Database, Caching

6	Data Privacy	Secure handling of user financial data	Data Encryption, Secure APIs, Access Control
7	Maintainability	Easy model retraining and system updates	Modular Python Code, Version Control (Git)