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

Date	4 November 2023
Team ID	PNT2023TMID592341
Project Name	Project – Online Payments Fraud Detection
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the Table 1 and Table 2

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Interface for administrators to review flagged transactions and take necessary actions.	HTML, CSS, JavaScript / React Js
2	Application Logic-1	Handling data collection and preprocessing logic.	Python
3	Application Logic-2	Deep learning model for online payments fraud detection.	TensorFlow, PyTorch
4	Application Logic-3	Real-time monitoring and alerting logic.	Python, IBM Watson STT service, IBM Watson Assistant
5	Database	Storage for transaction data and model-related information.	CSV files
6	Cloud Database	Cloud-based database service for scalability and accessibility.	IBM Db2, IBM Cloudant
7	File Storage	Storage for system-related files and configurations.	IBM Block Storage or Other Storage Service or Local Filesystem
8	External API-1	Integration with external APIs for additional data sources (e.g., transaction history).	IBM Weather API
9	External API-2	Integration with external APIs for identity verification and additional data.	Aadhar API

10	Machine Learning Model	Core machine learning model for fraud detection.	Supervised Learning Model (e.g., Random Forest, Gradient Boosting)
11	Infrastructure (Server / Cloud)	Deployment of the application on a server or cloud infrastructure.	Cloud Foundry, Kubernetes, IBM Cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Utilization of open-source frameworks for machine learning and web development.	Scikit-learn, TensorFlow, Flask
2	Security Implementations	Implementation of robust security measures including SHA-256 encryption, role-based access controls, and adherence to OWASP security standards.	Encryption algorithms, IAM Controls, OWASP practices
3	Scalable Architecture	Adoption of a scalable architecture, utilizing microservices for flexibility and scalability to handle varying loads.	Microservices architecture, Kubernetes
4	Availability	Ensuring high availability through the use of load balancers, distributed servers, and redundancy in critical components.	Load balancers, Distributed server architecture
5	Performance	Design considerations for optimal performance, including caching mechanisms, CDN utilization, and efficient handling of a high volume of requests per second.	Caching strategies, Content Delivery Networks (CDN), Efficient request handling mechanisms