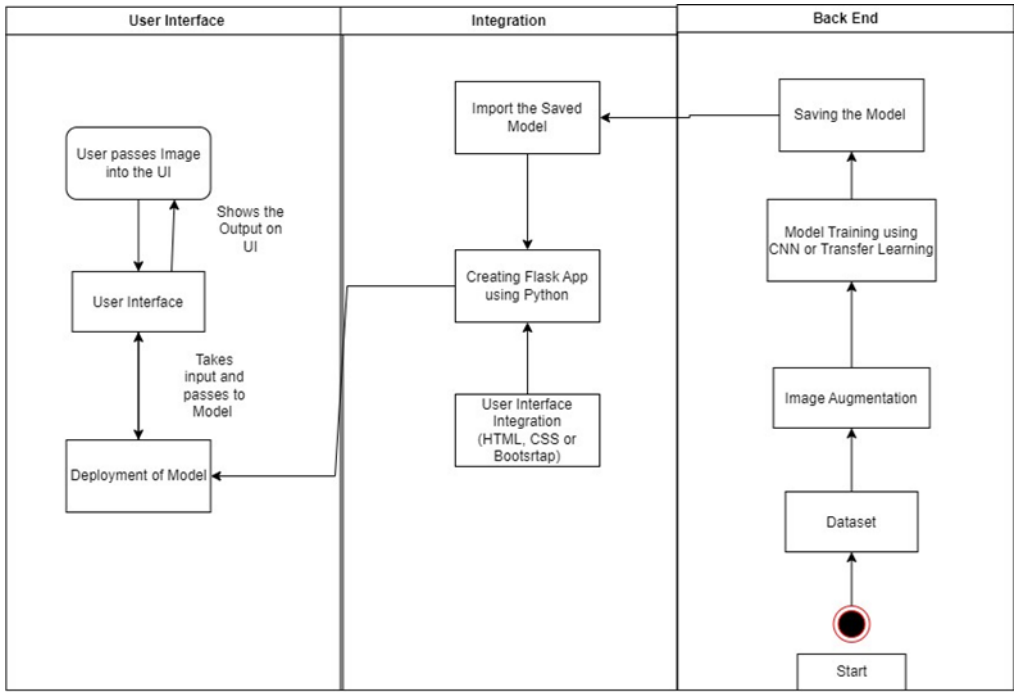
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	19-12-2023
Team ID	Team-591938
Project Name	Online Payments Fraud Detection Using Ml
Maximum Marks	4 Marks

Technical Architecture:

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



Guidelines:

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

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Data Collection	Collects data from various sources such as transactions, user profiles, device info, and IP addresses.	Data Ingestion Tools (e.g., Apache Kafka) - API integrations - Data Scraping Tools.
2.	Data Storage	Stores and manages data efficiently.	Relational Databases (e.g., PostgreSQL, MySQL) - NoSQL Databases (e.g., MongoDB, Cassandra)
3.	Data Processing/ETL	Cleans, preprocesses, and transforms raw data into a suitable format.	Apache Spark - Apache Flink - Talend - Apache Nifi
4.	Feature Engineering	Creates relevant features from raw data to improve model performance.	Python libraries (e.g., Pandas, NumPy) - Feature Scaling and Normalization
5.	Machine Learning Models	Implements supervised and unsupervised models for fraud detection.	Scikit-learn (for traditional ML models)- TensorFlow or PyTorch (for deep learning)- XGBoost, LightGBM (for ensemble models)
6.	Real-time Processing	Processes data in real-time for immediate fraud detection.	Apache Kafka for stream processing - Apache Flink for real-time analytics
7.	Model Deployment	Deploys trained models for online use.	Docker for containerization - Kubernetes for container orchestration - TensorFlow Serving, Flask, FastAPI for model serving
8.	Monitoring and Logging	Monitors system behaviour and logs events for analysis.	Logging frameworks (e.g., ELK Stack - Elasticsearch, Logstash, Kibana) - Prometheus and Grafana for monitoring
9.	Scalability and Performance	Ensures the system can handle increasing loads and demands.	Cloud Platforms (AWS, Azure, GCP) - Auto-scaling mechanisms

10.	Security	Implements measures to secure data and prevent unauthorized access.	Encryption algorithms (e.g., AES) - Access Control (e.g., OAuth, JWT)- SSL/TLS for data in transit
11.	Feedback Loop	Involves human validation and continuous learning for model improvement.	Human-in-the-loop systems - Model retraining pipelines
12.	Visualization and Reporting	Provides tools for visualizing and reporting fraud detection metrics.	Kibana, Tableau, Power BI for dashboard creation
13.	Compliance and Regulation	Ensures adherence to industry regulations and standards	Compliance tools for tracking and managing regulatory requirements

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Real-time Processing	Analyzing data as it is generated for immediate detection of fraudulent activities.	Apache Kafka for real-time data streaming - Apache Flink for stream processing
2.	Machine Learning Models	Utilizing algorithms to identify patterns and anomalies indicative of fraud.	Scikit-learn for traditional machine learning models - TensorFlow or PyTorch for deep learning models - XGBoost, LightGBM for ensemble models
3.	Scalability	The ability of the system to handle increasing amounts of data and user activity.	Cloud Platforms (AWS, Azure, GCP) for scalable infrastructure - Auto-scaling mechanisms for dynamic resource allocation
4.	Security	Implementing measures to secure sensitive data and prevent unauthorized access.	Encryption algorithms (e.g., AES) for data protection - Access control mechanisms (e.g., OAuth, JWT)- SSL/TLS for secure data transmission

5.	Data Collection	Gathering information from various sources to build a comprehensive dataset for analysis.	Data Ingestion Tools (e.g., Apache Kafka, Apache Nifi) - API integrations - Web scraping tools
6.	Feature Engineering	Creating relevant features from raw data to enhance the performance of machine learning models.	Python libraries (e.g., Pandas, NumPy) for data manipulation- Feature scaling and normalization techniques
7.	Model Deployment	The process of making trained machine learning models available for online use.	Docker for containerization- Kubernetes for container orchestration - TensorFlow Serving, Flask, FastAPI for model deployment
8.	Monitoring and Logging	Keeping track of system behaviour, logging events, and setting up alerts for suspicious activities.	Logging frameworks (e.g., ELK Stack - Elasticsearch, Logstash, Kibana) - Prometheus and Grafana for monitoring and visualization
9.	Feedback Loop	Involving human validation and continuously updating models to adapt to evolving fraud patterns.	Human-in-the-loop systems for expert review - Continuous learning mechanisms for model updates
10.	Visualization and Reporting	Providing tools for visualizing fraud detection metrics and generating reports	Kibana, Tableau, Power BI for dashboard creation and reporting
11.	Compliance and Regulation	Adhering to industry regulations and standards to ensure legal and ethical practices.	Compliance tools for tracking and managing regulatory requirements

References:

https://c4model.com/ https://www.leanix.net/en/wiki/ea/technical-architecture

https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d