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

Date	16 November 2023
Team ID	592286
Project Name	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 table1 & table 2

The technical architecture of a fraud detection system involves data collection, preprocessing, model development with supervised or unsupervised learning, real-time scoring, alert generation, case management, a feedback loop for continuous learning, integration with external systems, security and compliance measures, reporting, and visualization within a scalable and cloud-based infrastructure

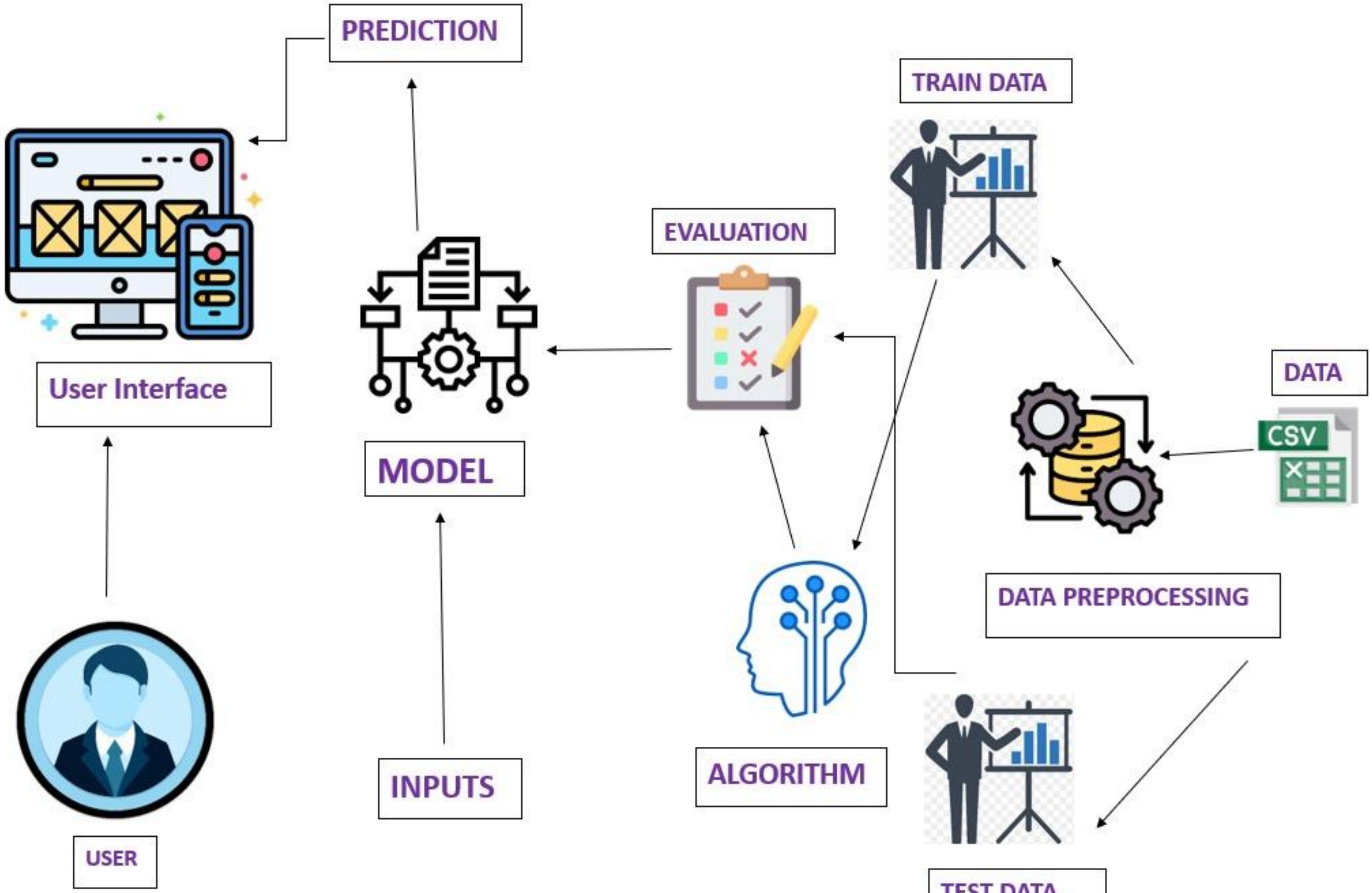
PREDICTION
TRAIN DATA
EVALUATION
User Interface
DATA
CSV
MODEL
DATA PREPROCESSING
ALGORITHM
INPUTS
TEST DATA
USER

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, mobile app etc.	HTML, CSS, SCSS
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	Render cloud service
4.	Application Logic-3	Logic for a process in the application	NIL
5.	Database	Data Type, Configurations etc.	No database is connected directly accessed from dataset through python
6.	Cloud Database	Database Service on Cloud	Render Cloud
7.	File Storage	File storage requirements	Deployed files through commands on render
8.	External API-1	Purpose of External API used in the application	ML API, etc.
9.	External API-2	Purpose of External API used in the application	API
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Executing on local machine and final deploying on render cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask, pytorch, sklearn, pandas	Flask is a lightweight web application framework written in Python.
2.	Security Implementations	Use secure protocols like HTTPS to encrypt data during transmission. For data at rest, employ encryption algorithms to protect stored information.	Ensure compliance with relevant data protection regulations such as GDPR, HIPAA, or PCI DSS.
3.	Scalable Architecture	A 3-tier architecture provides scalability, offers flexibility in technology stack.	load balancers, web server Nginx

S.No	Characteristics	Description	Technology
4.	Availability	Load balancers distribute incoming traffic across multiple servers, preventing any single server from becoming a bottleneck. This enhances availability by ensuring that the application can handle increased loads and maintains consistent performance.	Render cloud deployment
5.	Performance	Consider using CDNs for caching dynamic content to reduce server load.	Render cloud deployment