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

Date	26th October 2023
Team ID	Team-592696
Project Name	Detecting COVID-19 From Chest X-Rays Using Deep Learning Techniques
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)

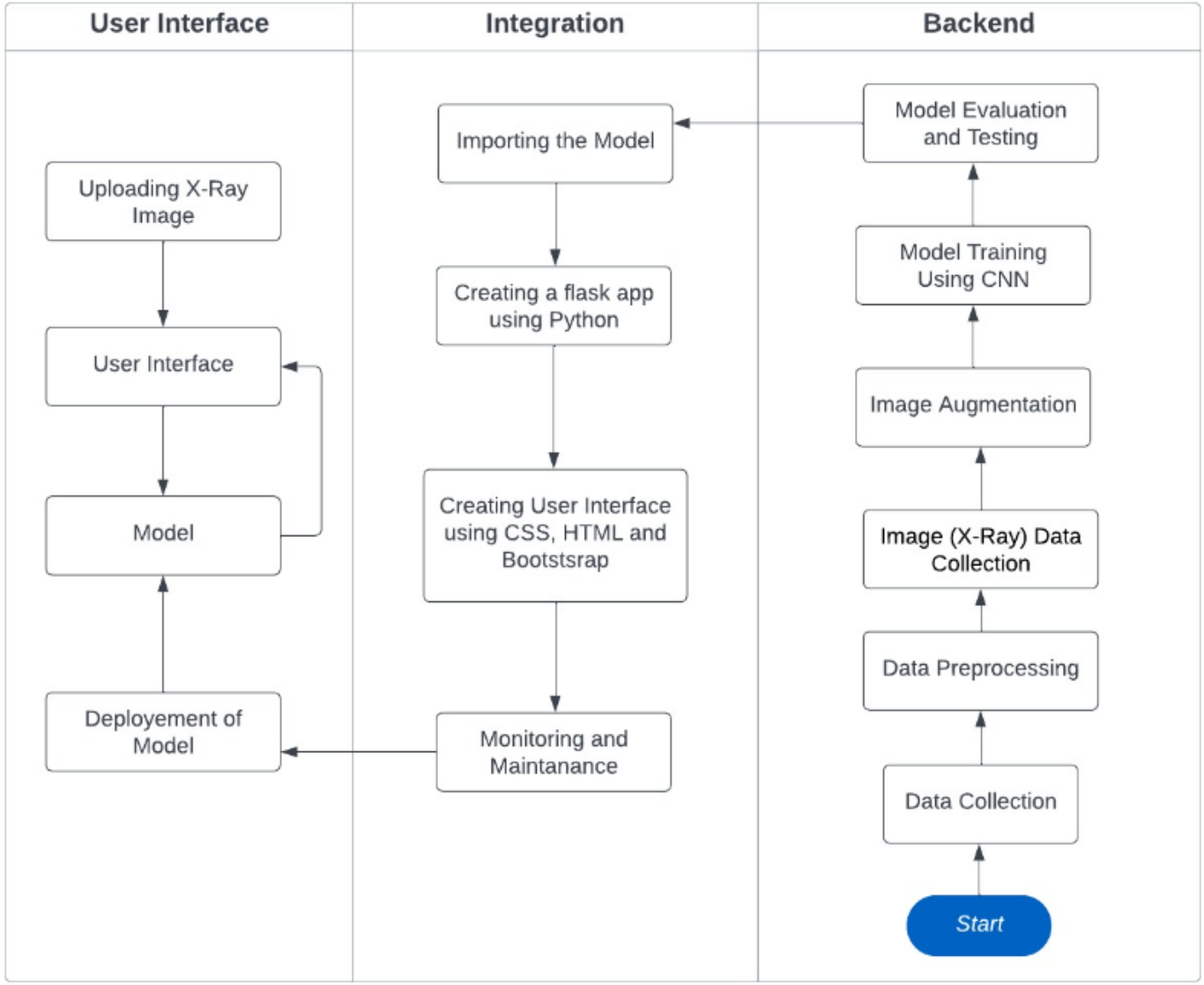
User Interface
Integration
Backend
Uploading X-Ray Image
User Interface
Model
Deployement of Model
Importing the Model
Creating a flask app using Python
Creating User Interface using CSS, HTML and Bootstsrap
Monitoring and Maintanance
Model Evaluation and Testing
Model Training Using CNN
Image Augmentation
Image (X-Ray) Data Collection
Data Preprocessing
Data Collection
Start

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Enable users to effectively control the computer or device they are interacting with.	Web Interface (CSS, Html and Bootstrap)
2.	Data Collection	Gathering chest X-ray images data	Python, OpenCV
3.	Data Pre processing	Image Augmentation, Normalization and data splitting.	Python, OpenCV, TensorFlow, scikit-learn
4.	File Storage	File storing requirements for storing data	Local System, Google Drive
5.	Model Selection	Selection of suitable deep learning model for the problem	CNN (Convolutional Neural Networks)
6.	Model Training	Training the deep learning model with available data	TensorFlow, Keras
7.	Evaluation	Model evaluation on validation and testing model using data sets.	Accuracy, Precision, F1-Score, ROC Curve
8.	Framework	Used to create a web application, Integrating Frontend and Backend	Python Flask
9.	Deployment	Creating an API or application for predicting decisions	Flask API
10.	Monitoring and Maintenance	Continuously monitoring and maintain the model and interface	Model versioning, CI/CD pipelines
11.	Infrastructure	Application Development on local System	Local, Kubernets
12.	Scaling	Deploying in scalable environment	AWS, Azure, Google Cloud Practioner

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilizing open-source frameworks for the development of model	Python Flask, TensorFlow
2.	Security Implementations	Implementing security controls for data privacy	Data Encryption, OWASP and Access controls
3.	Scalable Architecture	Ensuring scalability for increasing workloads	Microservices and Cloud-based Deployment
4.	Availability	Ensuring high availability of the system	Load Balancing and Redundancy