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

Date	05 November 2023
Team ID	Team-592145
Project Name	Deep learning model for disease detection in Tea leaves
Maximum Marks	4 Marks

Technical Architecture:

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

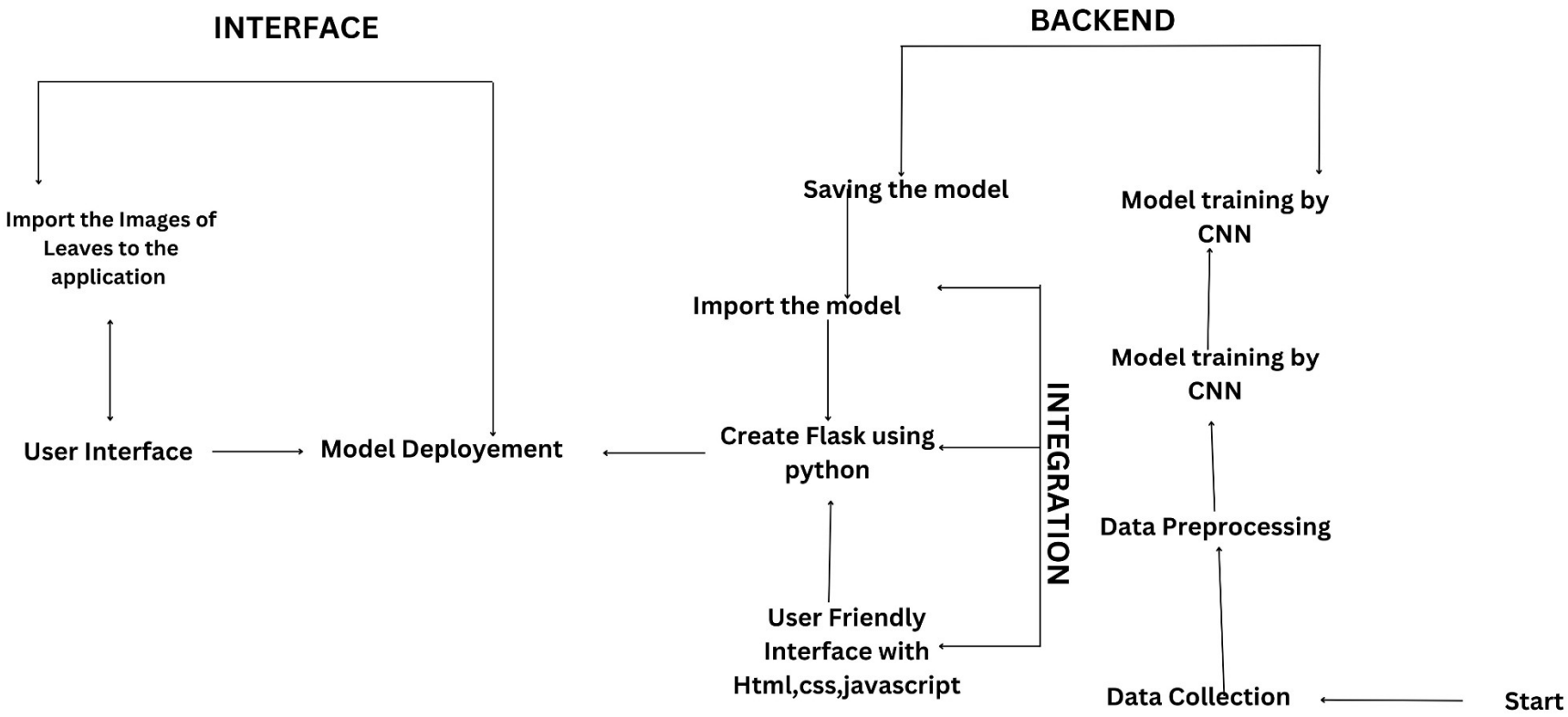
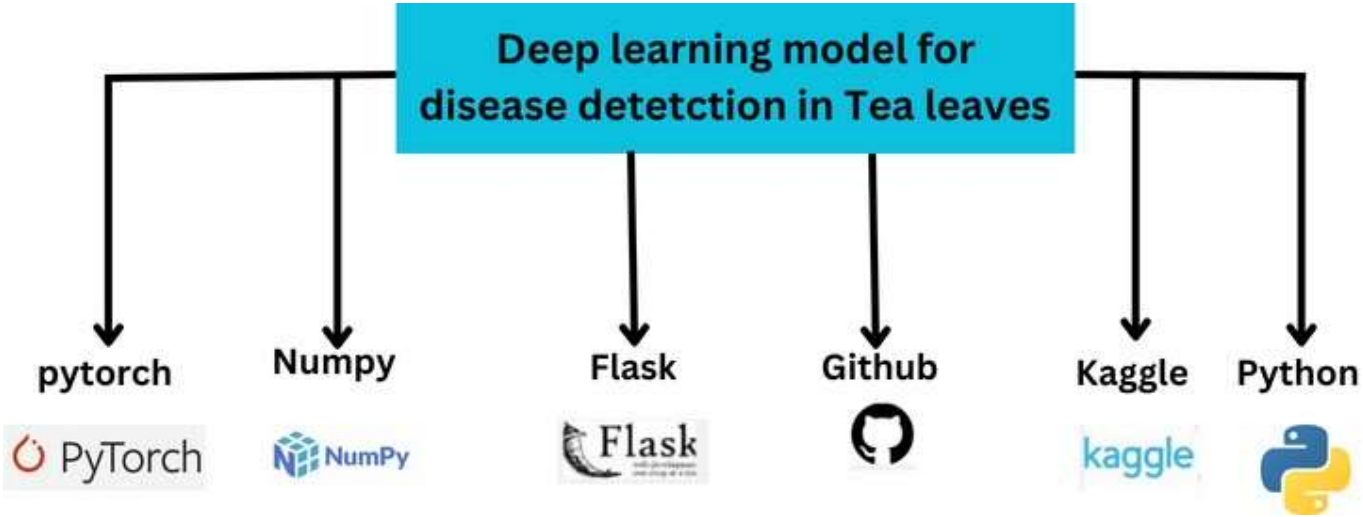

INTERFACE
BACKEND
Import the Images of Leaves to the application
User Interface
Model Deployement
Saving the model
Import the model
Create Flask using python
User Friendly Interface with Html,css,javascript
INTEGRATION
Model training by CNN
Model training by CNN
Data Preprocessing
Data Collection
Start

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Web Application / user interface	An user friendly interface to interact with the application	Html, css, javascript
2.	Data processing	The images need to be preprocessed to enhance their quality, such as resizing, rescaling, cropping, rotating, flipping, and applying principal component analysis (PCA) to reduce noise and redundancy	numpy
3.	Application Logic-1	Logic for a process in the application	Python
4	Database	Collect the Dataset Based on the Problem Statement	File Manager, MySQL etc.
5.	File Storage/ Data	File storage requirements for Storing the dataset	Local System, Google Drive Etc
6	Training	To make it more advanced to detect we ae are improving the CNN model	PyTorch
7	Frame Work	Used to Create a web Application, Integrating Frontend and Back End	Python Flask etc
8.	Deep learning model	Purpose of model	CNN model
9	Data collection	This phase involves collecting images of tea leaves with different types of diseases,	kaggle

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	List the open-source frameworks used	Python's Flask
2	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Controls
3	Scalable Architecture	Justify the scalability of architecture (3 – tier, Microservices)	
4	Availability	Justify the availability of applications (e.g. use of load balancers, distributed servers etc.)	Web service
5	Performance	Design consideration for the performance of the application	



Brief summary of what we did

- We collected and preprocessed images of tea leaves with different diseases using tools like numpy.
- We developed and trained a deep CNN model using frameworks like PyTorch.
- We deployed and tested the CNN model using tools like Flask.