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

Date	10 October 2022
Team ID	PNT2022TMID15472
Project Name	Fertilizer Recommendation System For Disease Prediction
Maximum Marks	4 Marks

Technical Architecture:

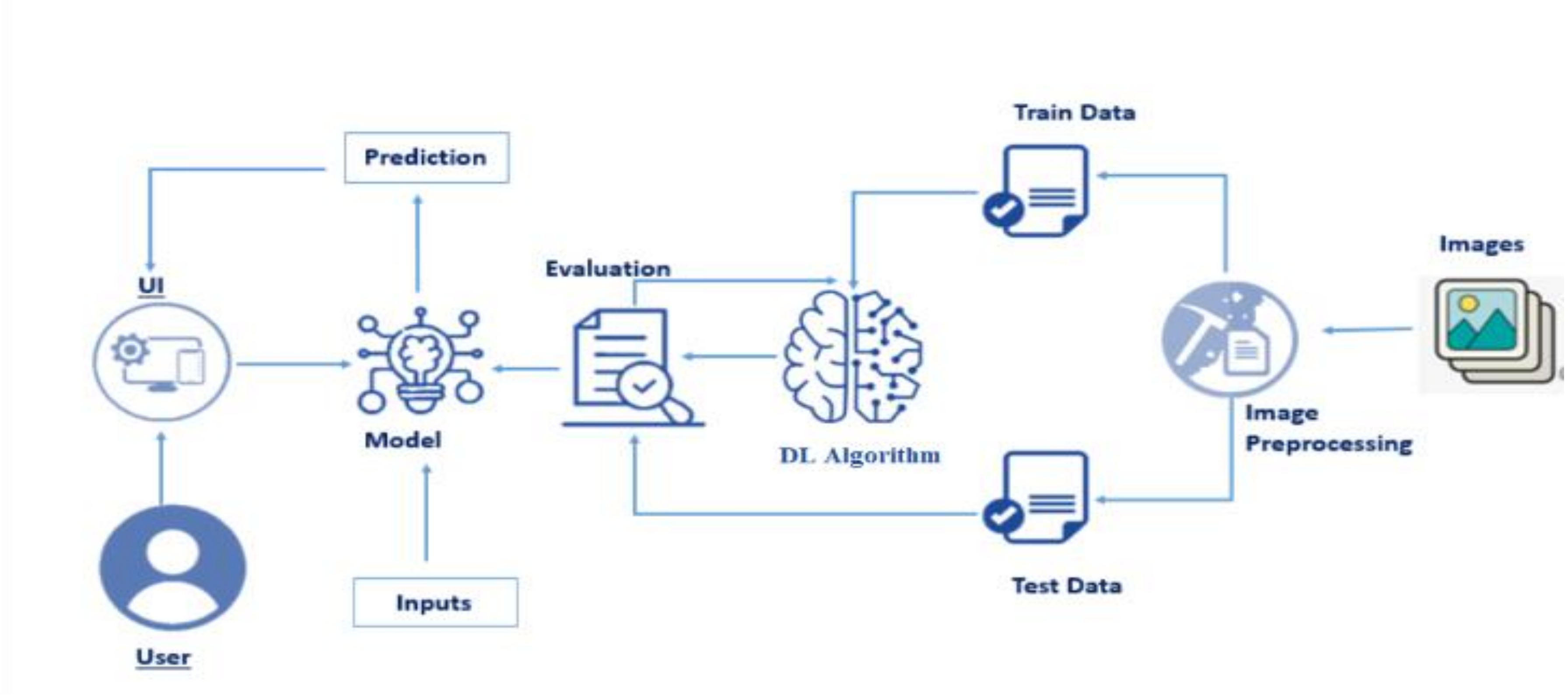

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web UI to feed in the image input	HTML, CSS, JavaScript, Bootstrap, Flask.
2.	Application Logic-1	Users can upload an image of a diseased leaf	HTML, CSS ,Bootstrap
3.	Application Logic-2	The image is then fed to the model through flask	IBM Watson STT service
4.	Application Logic-3	The image is analyzed by the model deployed and a certain fertilizer is recommended	IBM Watson Assistant
5.	Database	Data Type, Configurations, etc.	MySQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant, etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Allows the user to input the image for the necessary plant disease prediction	IBM Weather API, etc.
9.	External API-2	Provides the necessary action to be taken	IBM Weather API, etc
10.	Machine Learning Model	Uses various DL models for image prediction	Object Recognition Model, Image Classification, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2 : Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Used to design the web page and read image data through these pages	HTML, CSS, Bootstrap, Flask, etc…
2.	Security Implementations	Login verification through the login page	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Based on the scalability of architecture (3 – tier, Micro-services)	Various Technologies Used
4.	Availability	The application is scalable changes based on the cloud host and database used	IBM Watson
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDNs) etc.	Flask, Jupyter Notebook