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

Date	27 October 2023
Team ID	Team - 592909
Project Name	Potato Disease Classification
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable includes the architectural diagram shown as below and the information as per the table1 & table 2

Example: Order processing during pandemics for offline mode

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

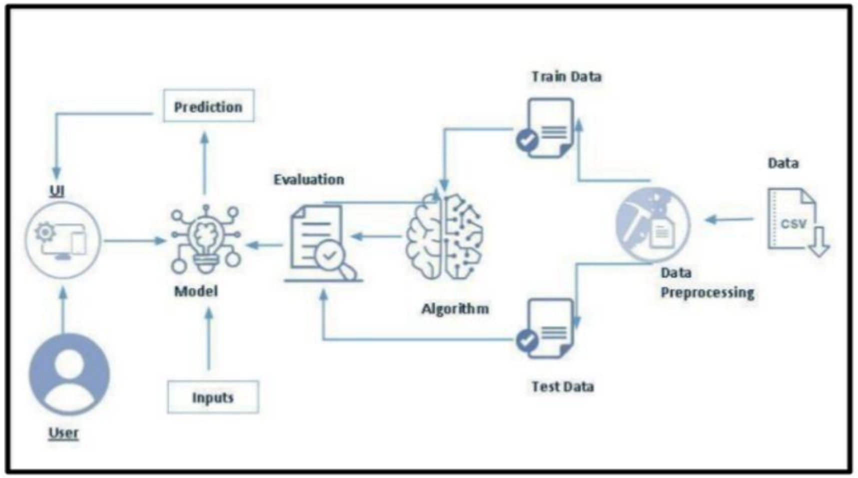

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Image processing Module	Module for processing potato plant images	Python, OpenCV
3.	Disease Classification Module	Module for classification potato Diseases	Python, TensorFlow, Keras
5.	Database	Data storage for users and image information	My SQL, MongoDB
6.	Cloud Database	Cloud based storage for processed data.	AWS S3, Google Cloud Storage
7.	External API	Integration with external disease databases	PlantVillage API, Crop Knowledge Center API
8.	Reporting module	Module for generating disease classification reports	IPython, Pandas, Matplotlib
9.	Notification	Service for notifying users about classification results	Email service, SMS service.
10.	Machine Learning Model	Model for disease classification	Convolutional Neural Networks (CNN), Transfer Learning
11.	Infrastructure (Server / Cloud)	Deployment and management of the application	Docker, Kubernetes, AWS EC2

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology

1.	Open-Source Frameworks	Implementation using open-source frameworks	Django, Flask, TensorFlow
2.	Security Implementations	Encryption and security protocols for data protection	SHA-256, SSL/TLS, Role based access controls
3.	Scalable Architecture	Utilization of microservices for scalability	Docker, Kubernetes, AWS Auto Scaling
4.	Availability	Ensuring high availability through load balancing and redundant servers	Nginx, AWS Elastic Load Balancer, Multi-region deployment
5.	Performance	Optimization for handling large data and traffic loads .	Caching (Redis), Content Delivery Networks (CDNs), Asynchronous processing with Celery

References:

https://c4model.com/

https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/

https://www.ibm.com/cloud/architecture

https://aws.amazon.com/architecture

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