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

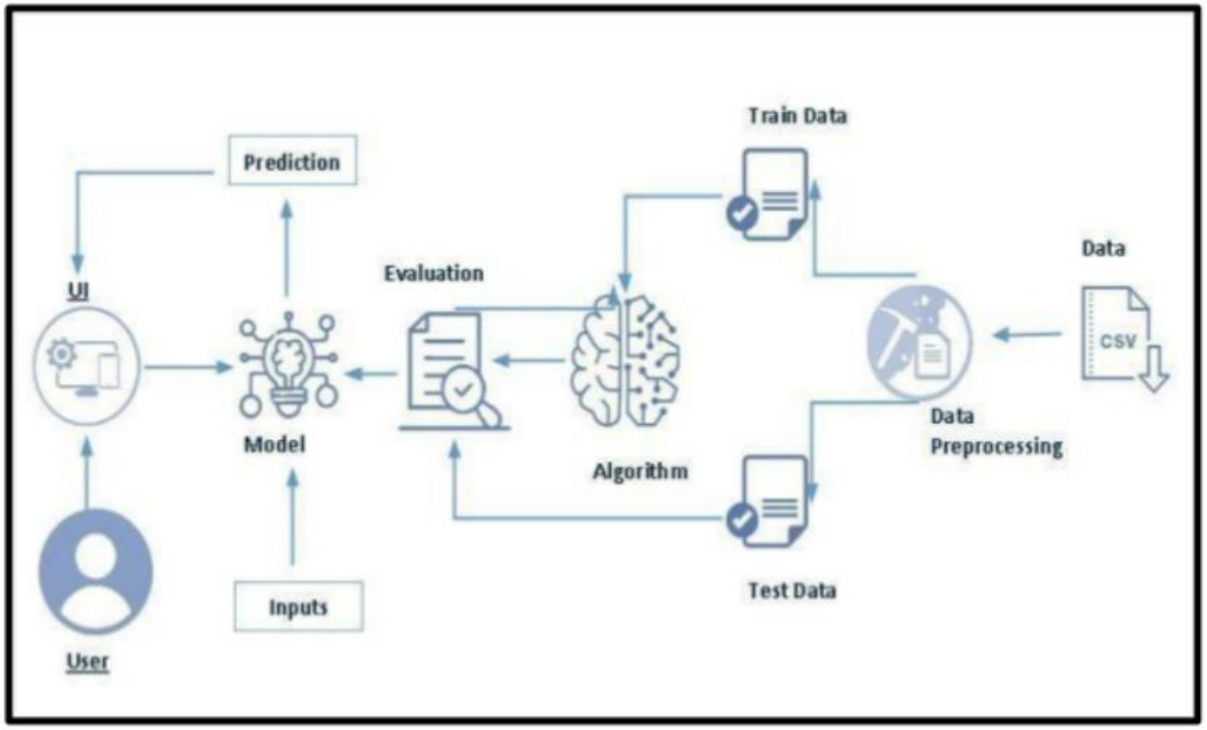
Date	27 October 2023
Team ID	Team - 592472
Project Name	Project Name- Potato Disease Classification
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram 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/



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)

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 ModulePython, TensorFlow, Keras	Module for classifying potato diseases	Python, TensorFlow, Keras
4.	Database	Data storage for user and image information	MySQL, MongoDB
5.	Cloud Storage	Cloud-based storage for processed data.	AWS S3, Google Cloud Storage
6.	External API	Integration with external disease databases	PlantVillage API, Crop Knowledge Center API
7.	Reporting Module	Module for generating disease classification reports	Python, Pandas, Matplotlib
8.	Notification	Service for notifying users about classification results	Email service, SMS service.
9.	Machine Learning Model	Model for disease classification	Convolutional Neural Networks (CNN), Transfer Learning
10.	Infrastructure	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

S.No	Characteristics	Description	Technology
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