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

Date	19 November 2023
Team ID	Team - 592200
Project Name	Fruits and Vegetables Classification with nutrition analysis using InceptionV3
Maximum Marks	4 Marks

Technical Architecture:

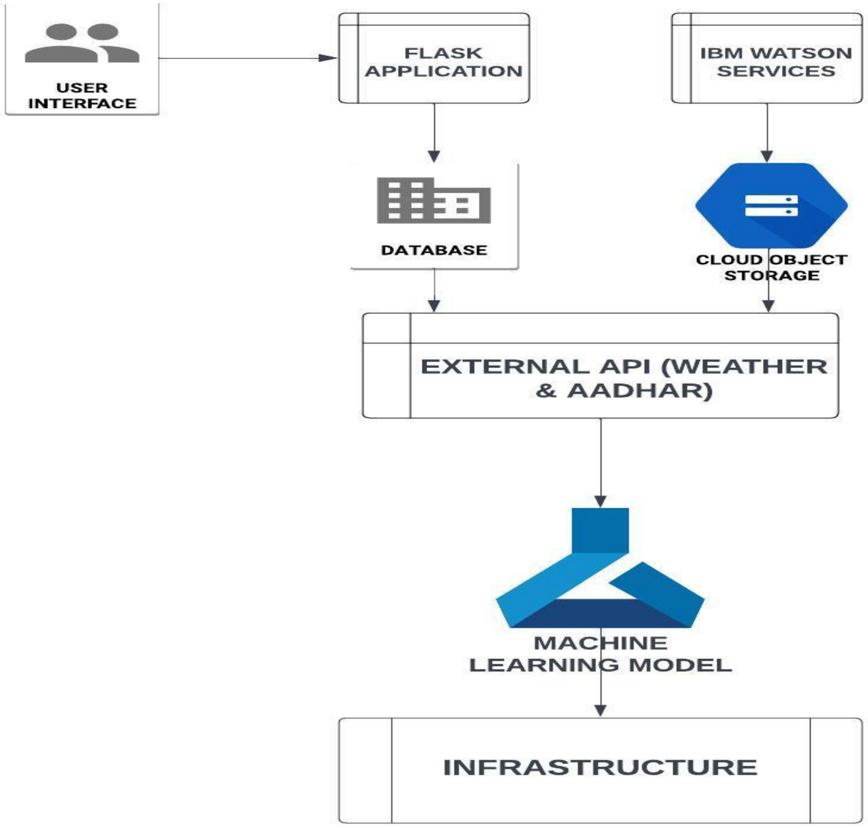

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web Interface for uploading images and viewing results.	HTML, CSS, JavaScript
2.	Flask Server	Backend server to handle image uploads, process requests, and serve results	Flask(Python)
3.	InceptionV3 Model	Pre-trained CNN model for image classification	TensorFlow, Keras
4.	Application Logic	Logic for a process in the application	IBM Watson Assistant
5.	Nutrient Analysis Module	Extracts nutrient information from the classified fruit/vegetable	Python Libraries (NumPy, Pandas), OpenCV (for image processing)
6.	Cloud Database	Stores information about fruits, vegetables, and user interactions	IBM DB2, SQLAlchemy (ORM for Flask), SQLite, MySQL, or other relational database
7.	User Authentication Module	Manages user accounts, authentication, and authorization	Flask-Login, JWT (JSON Web Tokens)
8.	Image Upload and Storage	Handles image uploads and storage	Flask-Uploads, Cloud Storage (e.g., AWS S3, IBM Cloud Object Storage)
9.	API Gateway	Manages and routes API requests to various backend services	Aadhar API, Flask-RESTful, Kong, AWS API Gateway
10.	External Services Integration	Integration with external services, such as IBM Watson for advanced analysis	RESTful APIs
11.	Machine Learning Model	ML model for classification of fruits and vegetables	InceptionV3 for Object Recognition

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	TensorFlow, Keras, Flask
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Encryption(HTTPS), IAM, Firewalls, OWASP
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Kubernetes(for container orchestration)

S.No	Characteristics	Description	Technology
4.	Availability	Justify the availability of applications (e.g. use of load balancers, distributed servers, etc.)	High Availability Clusters, Load Balancers, distributed Servers
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc.	Caching, Load Testing, CDN’s
6.	Integration	Seamless integration between the web interface, Flas server, InceptionV3 model, and external services(IBM Clouds, API’s)	RESTful API standards, Webhooks for external service integration, API Gateway for managing API calls
7.	Cost-Effective	Efficient use of resources, especially in a cloud deployment environment, to minimize costs.	Serverless architecture for cost-efficient scaling, Cloud cost monitoring tools (e.g., AWS Cost Explorer, IBM Cloud Cost and Asset Management)

IMPORTANT POINTS:

- Providing clear instructions on capturing high-quality images.
- Ensuring a diverse dataset for better model generalization.
- Including interesting facts about recognized items to keep users engaged.
- Establishing a feedback mechanism for continuous system improvement.
- Optimizing resource usage, especially in cloud deployments.