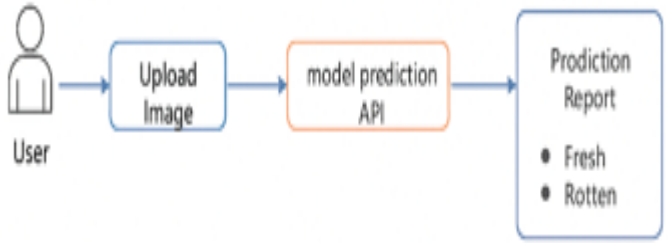
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	26 June 2025
Team ID	LTVIP2025TMID20416
Project Name	Smart Sorting: Transfer Learning for Identifying Rotten Fruits and Vegetables
Maximum Marks	4 Marks

Technical Architecture:

Example:



Guidelines:

Include all the processes (As an application logic / Technology Block)
Provide infrastructural demarcation (Local / Cloud)
Indicate external interfaces (third party API's etc.)
Indicate Data Storage components / services
Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web interface for uploading images and displaying predictions	HTML, CSS, JavaScript
2.	Application Logic-1	Image upload, form handling, and displaying	Python + Flask

		results	
3.	Application Logic-2	Backend logic for preprocessing and model prediction	TensorFlow / Keras
4.	Application Logic-3	Integration with ML model for image classification	Transfer learning
5.	Database	Store user history or logs (optional)	MySQL, NoSQL, etc.
6.	Cloud Database	For scalable storage of logs, user activity	IBM DB2, IBM Cloudant etc.
7.	File Storage	Temporary image uploads or dataset storage	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	(Optional) Weather API for future enhancements	IBM Weather API, etc.
9.	External API-2	(Optional) Nutritional Database (for future health suggestions)	Aadhar API, etc.
10.	Machine Learning Model	Machine Learning model to classify fruits as fresh or rotten	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Deployment on local system or cloud	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	TensorFlow, Flask, OpenCV, Pandas	Python, Flask, TensorFlow
2.	Security Implementations	Basic authentication, input validation, optional HTTPS	SHA-256, SSL, Flask-Security, JWT (optional)
3.	Scalable Architecture	Microservice-ready, loosely coupled (UI, API, ML Model)	Flask (for API), REST Architecture
4.	Availability	Can be hosted with high uptime on cloud platforms	IBM Cloud, Heroku with fallback & auto-restart
5.	Performance	Lightweight model for fast inference, caching predictions (if needed)	CDN (optional), model optimization, Nginx