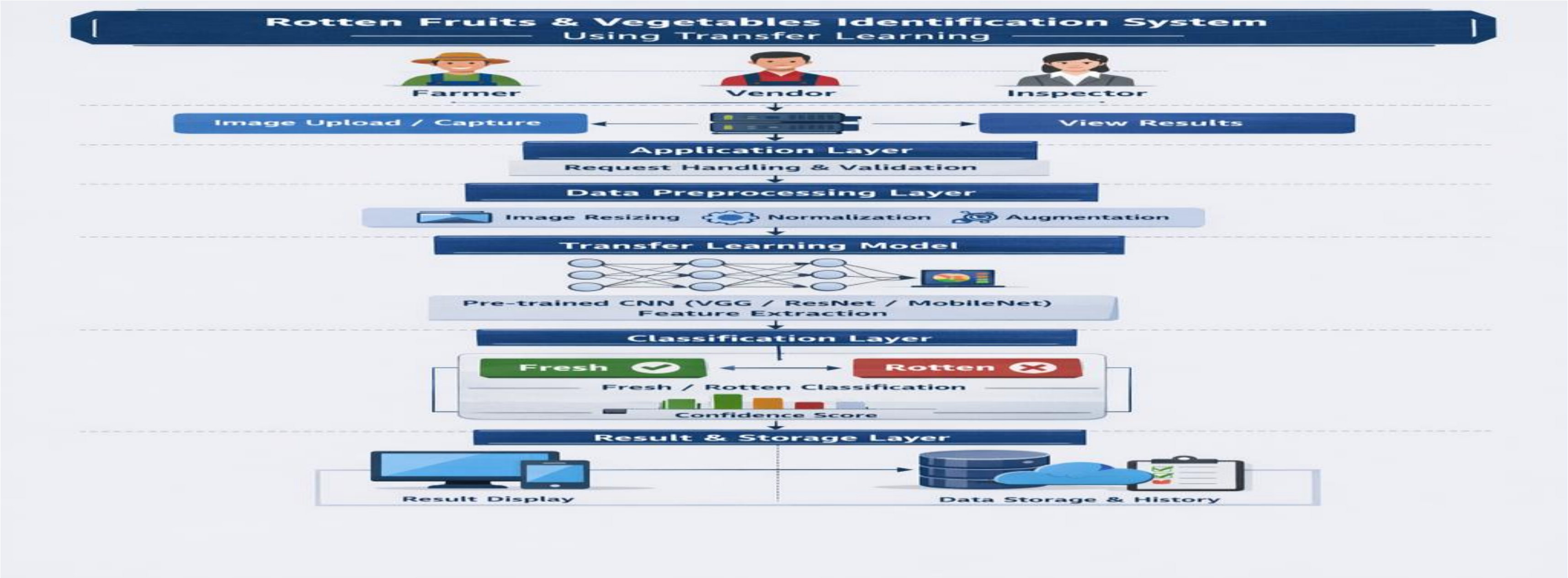
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	31 January 3035
Team ID	LTVIP2026TMID91218
Project Name	TRANSFERING LEARNING IDENTIFICATION OF FRUITS AND VEGETABLES
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



The technical architecture of the Rotten Fruits and Vegetables Identification System using Transfer Learning illustrates the complete flow of data from image input to result generation. The system is designed in a layered manner to ensure modularity, scalability, and efficient processing.

The process begins at the User Interface layer, where users such as farmers, vendors, or quality inspectors upload or capture images of fruits and vegetables. These images are sent to the Application Layer, which handles request validation, user authentication, and communication with the backend system.

Next, the images pass through the Data Preprocessing Layer, where operations such as image resizing, normalization, and augmentation are performed. This step ensures that the input images are in a suitable format for the deep learning model and improves classification accuracy.

The processed images are then fed into the Transfer Learning Model Layer, which uses a pre-trained convolutional neural network (such as VGG, ResNet, or MobileNet) to extract meaningful features like texture, color, and surface defects. These extracted features are forwarded to the Classification Layer, where the model classifies the produce as either Fresh or Rotten along with a confidence score.

Finally, the results are displayed to the user through the Result & Storage Layer, and relevant data such as images and prediction history are stored for future reference. This architecture ensures accurate, fast, and automated detection of spoiled fruits and vegetables, reducing manual effort and food wastage.

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.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, 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	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, 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	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used

S.No	Characteristics	Description	Technology
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN’s) etc.	Technology used