Project Planning Phase

Date	29 June 2025
Team ID	LTVIP2025TMID41462
Project Name	Smart Sorting : Transfer Learning For Identifying Rotten Fruits And Vegetables
Maximum Marks	

1. Project Description

Smart Sorting AI is an image classification system developed using deep learning that classifies input images of fruits and vegetables into four categories:

- Fresh Fruit
- Rotten Fruit
- Fresh Vegetable
- Rotten Vegetable

The model uses MobileNetV2 and a balanced dataset to achieve ~72% accuracy, with future scope to improve it further.

2. Project Objectives

- Create a working deep learning model for smart sorting.
- Design a web interface using Flask for real-time image predictions.
- Enable users to upload images and get class predictions.
- Train a model on a 4-class labeled dataset from Kaggle.
- Make the solution lightweight and deployable locally.

3. Technology Stack

Layer	Tools/Frameworks Used
Programming	Python 3.10
Deep Learning	TensorFlow, Keras
Model Architecture	MobileNetV2 (Pre-trained on ImageNet)
Dataset	Kaggle Dataset (Fresh & Stale Fruits/Vegetables)
Web Framework	Flask (Python microframework)
Frontend	HTML, CSS (custom styled form)
Image Handling	Pillow (PIL), base64 for rendering images
Hosting (optional)	Localhost / Flask-Ngrok (for demo)

---

4. Resources Required

Resource Type	Description
Hardware	Google Colab (GPU), Local PC with Anaconda

Resource Type	Description
Dataset	Kaggle: Fresh and Stale Images Dataset
IDEs/Editors	Google Colab, VS Code, Notepad++
Libraries	TensorFlow, Pillow, Flask, Flask-Ngrok
Collaboration Tools	Google Drive, GitHub, PDF editor

---

5. Risk Analysis & Mitigation

Risk	Mitigation Strategy
Overfitting due to small subset	Use dropout, image augmentation
Accuracy too low (<70%)	Tune layers, try more data, longer training
Deployment errors (Flask 404, etc.)	Test each route locally, add validations
File compatibility (image types, .h5)	Standardize formats and handle exceptions

---

6. Success Criteria

- Accuracy ≥ 70% on validation set
- Smooth prediction interface (image → result)
- Working website with upload and result display
- Fully documented GitHub repo and demo video