Project Design Phase
Solution Architecture

Solution Architecture – Rotten Fruits and Vegetables Detection System

---

1. Input Layer (Data Ingestion)

- **Sources**:
    - High-resolution images from cameras (conveyor belt, drone, mobile app)
    - IoT sensors (for temperature, humidity, gas emission)
- **Tools**: Raspberry Pi / Arduino (for sensors), smartphones, CCTV

---

2. Data Storage

- **Raw Image Data**:
    - Cloud Storage (AWS S3, Google Cloud Storage)
- **Structured Data**:
    - Metadata + sensor data (time, location, temperature) in databases
    - **Database**: Firebase, MySQL, PostgreSQL

---

3. Data Processing

- **Preprocessing**:
    - Image resizing, augmentation, normalization
    - Noise reduction in sensor data
- **Tools**:
    - OpenCV, TensorFlow/Keras preprocessing layers, NumPy, Pandas

---

4. Model Layer (AI/ML Engine)

- **Model Type**:
    - CNN (Convolutional Neural Network) for image classification
    - Optional: Object Detection (YOLOv8, SSD) for marking spoiled areas
- **Model Framework**:
    - TensorFlow, PyTorch, Scikit-learn
- **Training**:
    - Labeled dataset of fresh vs rotten produce
    - Techniques: Transfer Learning, Data Augmentation

---

5. Model Serving & Inference

- **Real-Time Inference**:

    - On Edge (NVIDIA Jetson, mobile) or cloud (AWS SageMaker, GCP Vertex AI)

- **Batch Inference**:
    - For daily or batch quality checks in warehouses

---

6. Decision Layer

- **Outputs**:
    - Classification (Rotten / Fresh / Mildly Spoiled)
    - Quality score (0–100)
- **Action Triggers**:
    - Notify staff
    - Automatic sorting
    - Disposal alert
    - Re-routing in supply chain

---

7. User Interface

- **Dashboard**:
    - Quality trends, real-time camera feed, batch inspection results
- **Mobile App**:

    - Upload image, view results, feedback

- **Tools**:
    - Web: React / Angular + Flask / Node.js
    - App: Flutter / Android Studio

---

8. Monitoring & Feedback Loop

- **Monitoring**:
    - Model drift
    - Accuracy tracking
    - Manual override logs
- **Feedback Loop**:
    - Human review of misclassified data
    - Re-train model with new images

Diagram Suggestion

plaintext
CopyEdit

```
[Camera/Input Devices] --> [Preprocessing Layer] --> [CNN Model Inference]
        |                          |                          |
   [Sensor Data] ---------> [Feature Fusion] --------> [Rotten / Fresh Decision]
        |                          |
   [Cloud DB]            <-- [Model Feedback / Logs]
```

Technologies Summary

Layer	Tools/Tech
Input	Cameras, IoT Sensors, Raspberry Pi
Storage	AWS S3, Firebase, SQL
Processing	OpenCV, Pandas, TensorFlow, NumPy
AI/ML Model	CNN, PyTorch, TensorFlow, YOLO
Inference	Edge Devices, Flask APIs, AWS/GCP/Azure
Dashboard/App	React, Angular, Flutter, Power BI, Tableau
Monitoring	Prometheus, Grafana, custom logging