FruitSure – Smart Agricultural Fruit Grading and Disease Detection

Overview

FruitSure is an AI-powered web platform designed to help apple farmers assess fruit quality and detect leaf diseases.
It leverages computer vision and machine learning to automate the traditionally manual, subjective, and error-prone processes of apple grading and disease diagnosis.

The platform provides three main features:

1. Apple Leaf Disease Detection – identifies diseases from uploaded apple leaf images using a deep learning model such as Vision Transformer (ViT) or YOLOv8.
2. Apple Quality Grading – analyzes apple images (and optionally acoustic vibration data) to determine fruit quality (e.g., good, average, poor).
3. Apple Price Insights – uses web scraping to fetch and display current apple market prices from verified online sources.

Technical Details

1. Disease Detection Module

- Model: Vision Transformer (ViT) or YOLOv8 (tested on datasets such as Plant Pathology 2021 and Apple Leaf datasets).
- Task: Multi-class classification (Healthy, Rust, Scab, Multiple Diseases, etc.).
- Input: Leaf image uploaded by the farmer.
- Output: Predicted disease class and suggested remedies.
- Backend: Python using PyTorch or TensorFlow.

2. Apple Grading Module

- Approach: Combines image-based features (color, texture, shape) and acoustic readings if available.

- Goal: Categorize apples into grades such as A (Premium), B (Average), and C (Low).
- Benefit: Enables farmers to price apples accurately and maintain consistent quality.

3. Price Scraping Module

- Method: Automated web scraping using libraries such as requests, BeautifulSoup, or Selenium.
- Data Source: Agricultural market or government price websites.
- Output: Displays real-time apple prices based on region and grade.

Web Platform Architecture

Component	Technology
Frontend	HTML, CSS, JavaScript or React
Backend	Flask or Django
ML Integration	Model inference through an API endpoint
Database	SQLite or PostgreSQL
Web Scraping	Python (BeautifulSoup or Scrapy)
Chatbot	RAG-based system using LangChain and a Large Language Model (LLM) such as Mistral, LLaMA, or GPT
Hosting	Render, Railway, or local/cloud servers