

# Smart Plant Monitoring System (Plantify)

## Project Overview

The **Smart Plant Monitoring System** is an integrated software solution designed to assist farmers and agricultural experts in making data-driven decisions. The system combines traditional Data Analysis, Machine Learning for crop recommendation, and Deep Learning for plant disease detection into a unified web application.

The platform provides an intuitive interface for monitoring soil conditions, recommending suitable crops based on environmental factors, diagnosing plant diseases from leaf images, and answering agricultural queries using Generative AI.

## Team Members

- Marwan Mostafa Ismail (Team Leader)
- Fares Emad Abdelnaby Elsayed
- Youhanna George Jacoub Ibrahim
- Mohamed Hisham Hussian Mahmoud
- Adham Jehad Abdelmoneam Eloraby
- Mohamed Ezzat Saad Ghoraba

## Contents

Smart Plant Monitoring System (Plantify) .....	1
Project Overview .....	1
Team Members .....	1
Key Features .....	3
1. Interactive Data Analysis .....	3
2. Smart Crop Recommendation .....	3
3. Plant Disease Detection .....	3
4. AI Agricultural Assistant .....	3
Technical Architecture .....	4
Tech Stack .....	4
File Structure Description .....	4
Installation & Usage .....	4
Prerequisites .....	4
Running the Application .....	4
Model Details .....	5
Crop Recommendation Model .....	5
Disease Detection Model .....	5
Future Improvements .....	5

# Key Features

## 1. Interactive Data Analysis

A dynamic dashboard that allows users to explore agricultural datasets.

- **Capabilities:** Users can upload CSV files to generate instant visualizations.
- **Plot Types:** Histograms, Box Plots, Scatter Plots, and Bar Charts.
- **Tech:** Built using Plotly Express for interactive charts.

## 2. Smart Crop Recommendation

Uses Machine Learning to analyze soil and environmental metrics to suggest the most suitable crop.

- **Input Parameters:** Nitrogen (N), Phosphorus (P), Potassium (K), Temperature, Humidity, pH, and Rainfall.
- **Model:** LightGBM (Gradient Boosting Framework).
- **Performance:** Validated using Stratified 5-Fold Cross-Validation to ensure robustness.

## 3. Plant Disease Detection

A Computer Vision module that identifies plant diseases from uploaded leaf images.

- **Input:** Image files (JPG/PNG) of plant leaves.
- **Model:** Convolutional Neural Network (CNN) trained on the **PlantVillage** dataset.
- **Framework:** TensorFlow / Keras.
- **Output:** Predicts the specific disease class or confirms the plant is healthy.

## 4. AI Agricultural Assistant

An integrated chatbot powered by Google's Generative AI.

- **Functionality:** Acts as an expert consultant, answering questions about pest control, farming techniques, and explaining model predictions.
- **Integration:** google.generativeai API.

# Technical Architecture

## Tech Stack

- **Programming Language:** Python 3.x
- **Web Framework:** Streamlit (for the frontend UI)
- **Data Manipulation:** Pandas, NumPy
- **Visualization:** Plotly, Matplotlib, Seaborn
- **Machine Learning:** Scikit-learn, LightGBM
- **Deep Learning:** TensorFlow, Keras
- **LLM Integration:** Google Gemini API

## File Structure Description

- `FS.py`: The main entry point for the Streamlit application. It handles routing between pages (Home, Analysis, Prediction, Detection, AI Chat) and integrates the backend logic.
- `Plantify_Notebook.ipynb`: The research notebook used for Data Cleaning, Exploratory Data Analysis (EDA), and training the LightGBM Crop Recommendation model.
- `CV_plantify.ipynb`: The notebook used for training the Deep Learning CNN model on the PlantVillage dataset for disease classification.
- `model.pkl`: The serialized LightGBM model for crop recommendation.
- `Color_Images.h5`: The trained Keras model weights for image classification.

## Installation & Usage

### Prerequisites

Ensure you have Python installed. You will need to install the following dependencies:

```
pip install streamlit pandas numpy plotly tensorflow lightgbm google-generativeai matplotlib scikit-learn
```

### Running the Application

1. Clone the repository or download the project files.
2. Navigate to the project directory in your terminal.
3. Run the Streamlit app:  

```
[streamlit run FS.py]
```
4. The application will open in your default web browser at <http://localhost:8501>.

# Model Details

## Crop Recommendation Model

- **Algorithm:** LightGBM Classifier.
- **Training Strategy:** The model was trained on historical crop data correlating soil nutrients and weather conditions with optimal crop types.
- **Validation:** Achieved high accuracy and stability through StratifiedKFold validation (5 splits), ensuring the model performs well across different data distributions.

## Disease Detection Model

- **Architecture:** Deep Convolutional Neural Network (CNN).
- **Preprocessing:** Images are resized and normalized before being fed into the network.
- **Dataset:** PlantVillage (Labeled dataset containing thousands of images of healthy and diseased leaves).
- **Classes:** The model supports multiple classes including Blights, Rusts, Mildews, and Healthy states.

## Future Improvements

- **Real-time IoT Integration:** Connect directly to hardware sensors (Arduino/Raspberry Pi) to fetch N, P, K, and pH values automatically.
- **Mobile App Wrapper:** Convert the Streamlit web app into a mobile-native format for easier field usage.
- **Cloud Deployment:** Deploy the system to AWS or Heroku for public access.