AI-BASED SOLUTION FOR ADDRESSING PLANT DISEASES IN EGYPT'S FRUIT AND VEGETABLE CROPS

Turning Small Steps into Big Impact

Supervised by: Dr. Mahmoud Basiony

Assistant Supervisor: Mrs. Toka Ashraf

Tabel of contents

OUR TEAM

Kholoud Mohsen 21-00882
Team_Leader

Eslam Ayman 21-01854

Bishoy Ezzat 21-00584

George Hany 21-00724

Hager Abdelqader 21-02131

Mariam Hany 21-01011

Farah Mohamed 21-00686

Introduction

For generations, farmers have depended on personal experience and expert consultations to manage crop diseases. However, this traditional approach is often slow, costly, and prone to misdiagnosis—especially during peak seasons when quick intervention is critical.

To address these challenges, we developed Planty Care—an AI-powered mobile application that revolutionizes plant disease detection and agricultural decision-making. With a simple photo of a leaf, the app delivers:

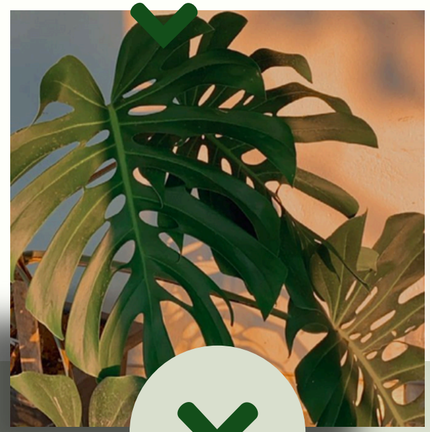
- Instant disease diagnosis
- Treatment recommendations
- In addition to these features, Planty Care also provides:
- Crop suggestions based on soil and environmental conditions
- Fertilizer guidance tailored to the plant's nutrient needs

This solution empowers farmers, gardeners, and agronomists with accurate, data-driven insights—helping them save time, reduce losses, and promote sustainable farming.

Proplem Defination

Plant diseases cause serious crop losses and are often hard to identify due to similar symptoms. Farmers usually rely on manual inspection or expert help, which can be slow, inaccurate, and costly.

There’s a need for a fast, reliable, and accessible solution to help farmers detect diseases early and apply the right treatment to reduce losses and boost productivity.



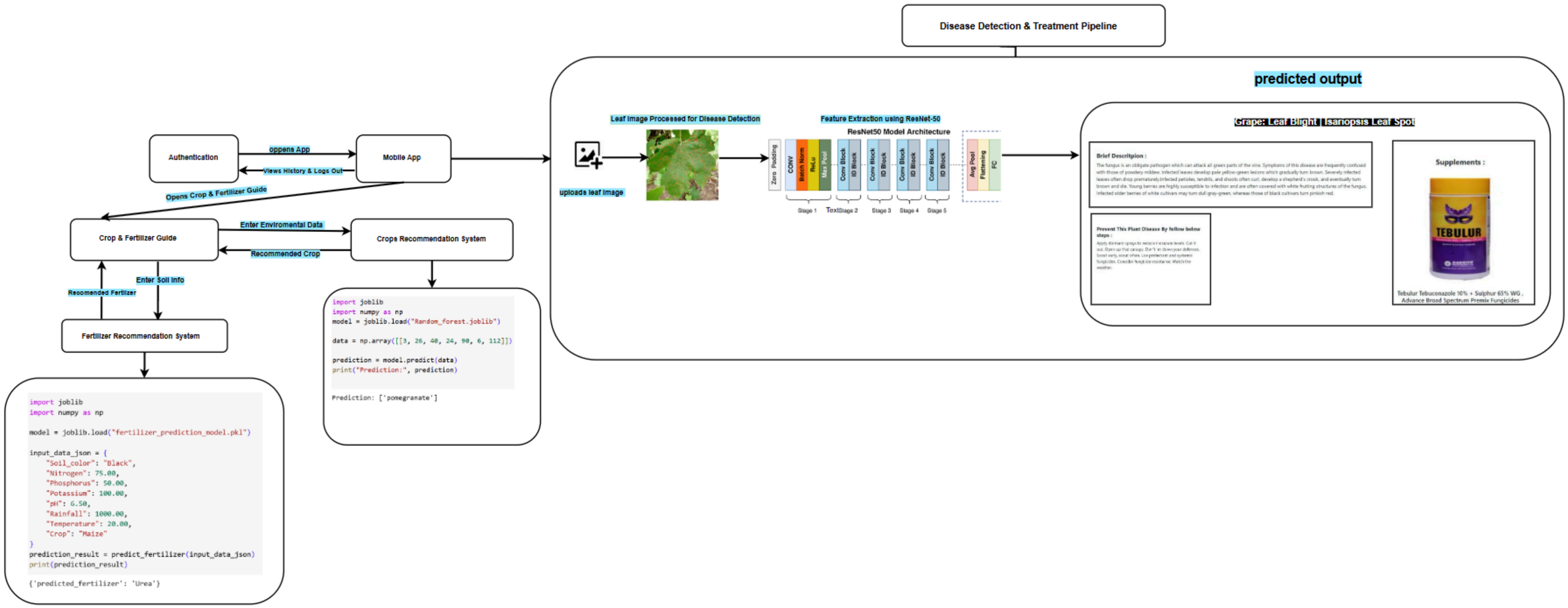
Objectives

1. **Detect plant diseases using AI (ResNet-50) from leaf images with high accuracy.**
2. **Recommend the best crop to plant based on soil and weather data using Random Forest.**
3. **Suggest suitable fertilizer based on soil nutrients using XGBoost.**
4. **Optimize models for mobile use, making them fast and easy to use in real farming conditions.**
5. **Ensure continuous updates to add new diseases, crops, and fertilizers as needed.**
6. **Data Security & Privacy Protection – Keeps user data safe and confidential.**

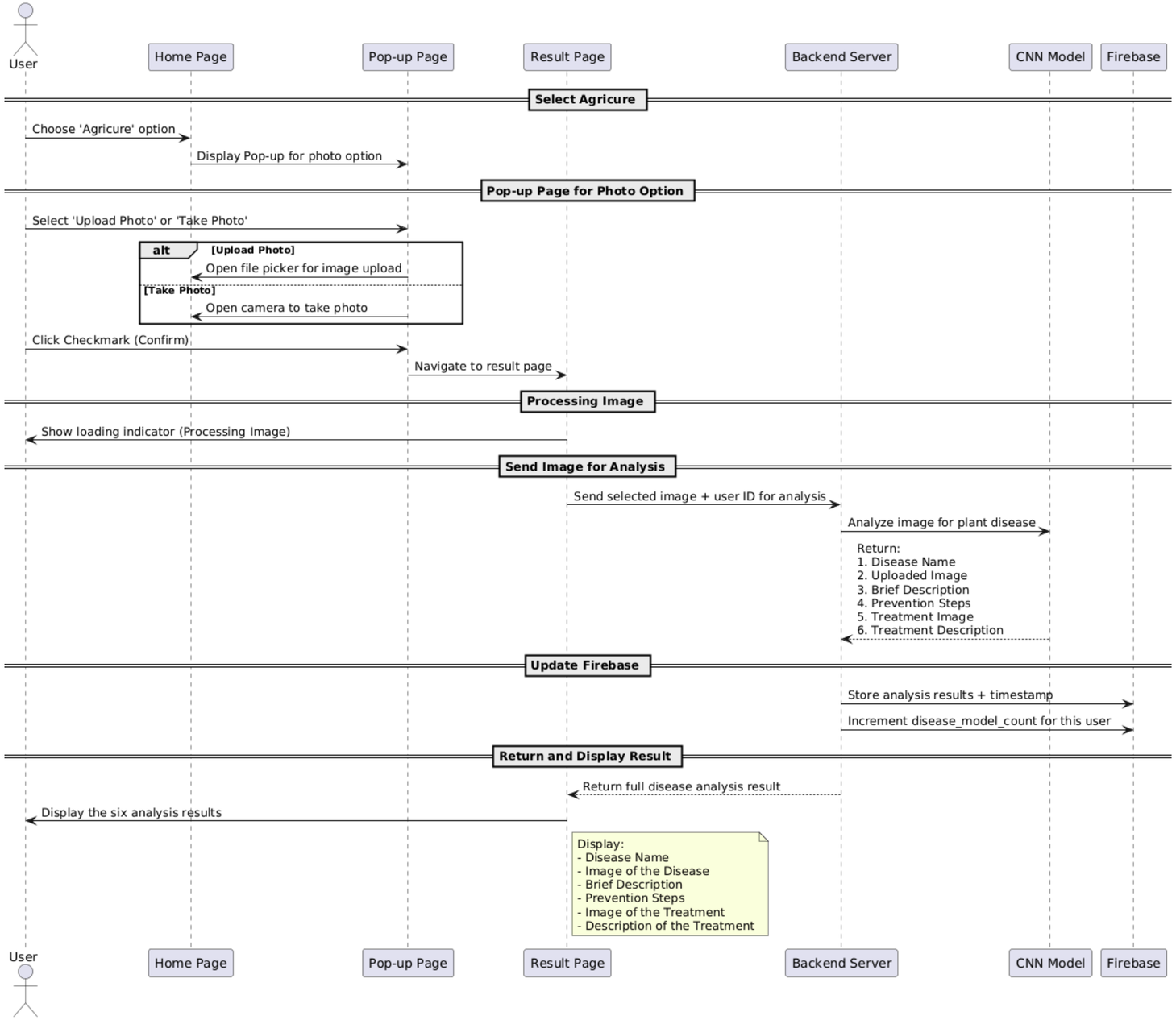
SYSTEM ANALYSIS AND DESIGN

System Analysis and Design is the process of understanding a problem, identifying its requirements, and then building the system's structure, including its architecture, data flow, and interactions, before implementation.

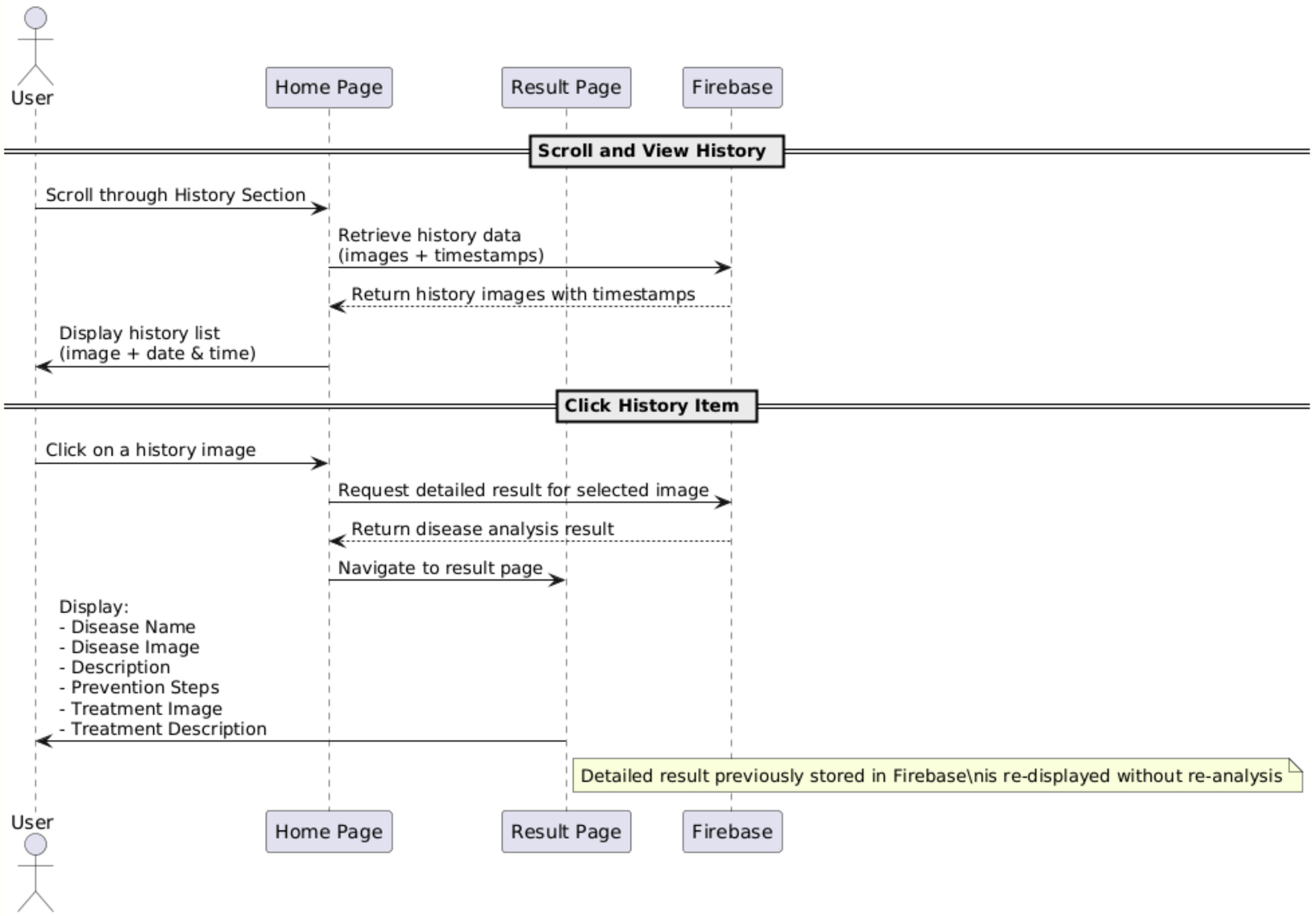
System Architecture



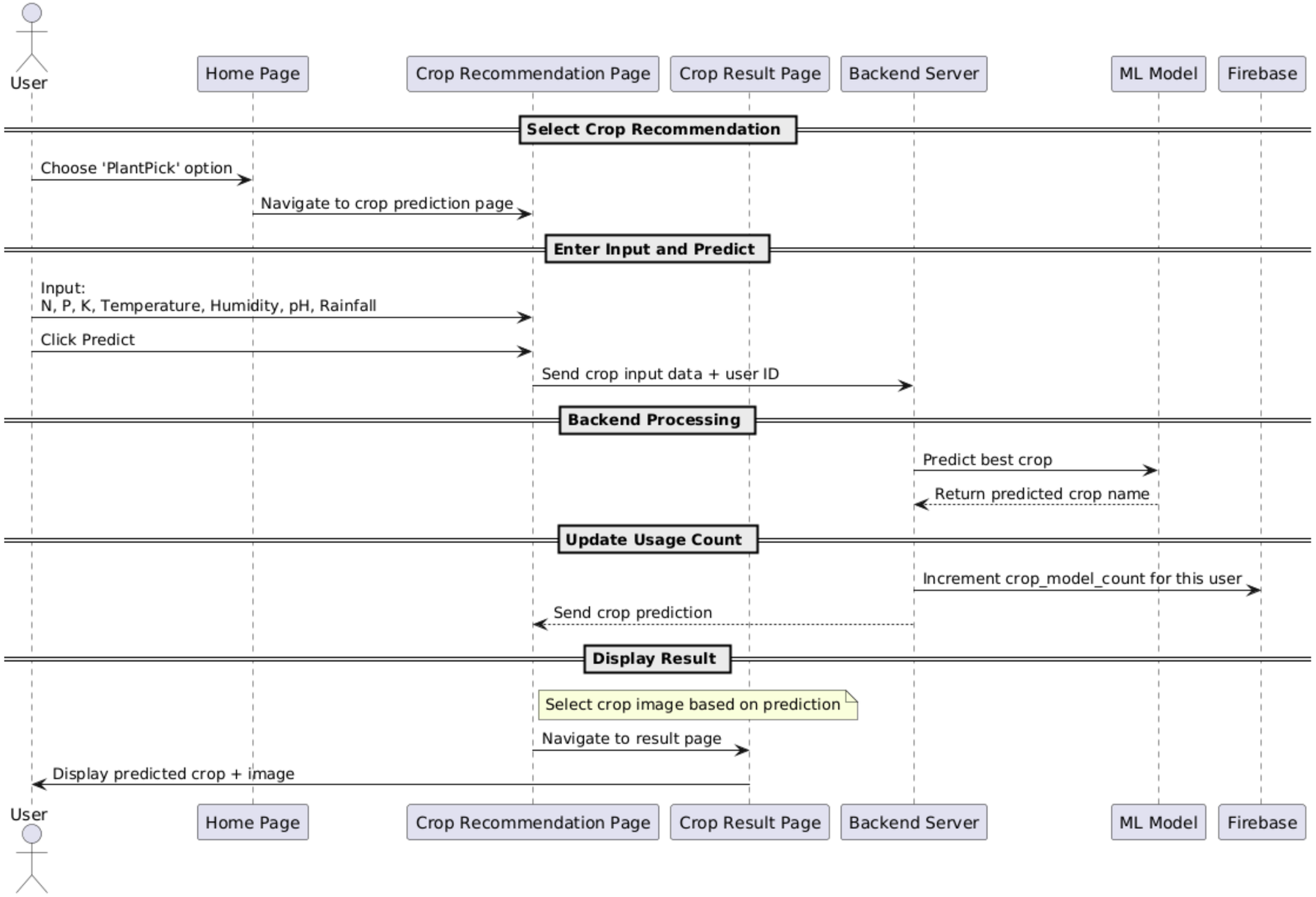
Sequence Diagram for Plant Disease Detection Model



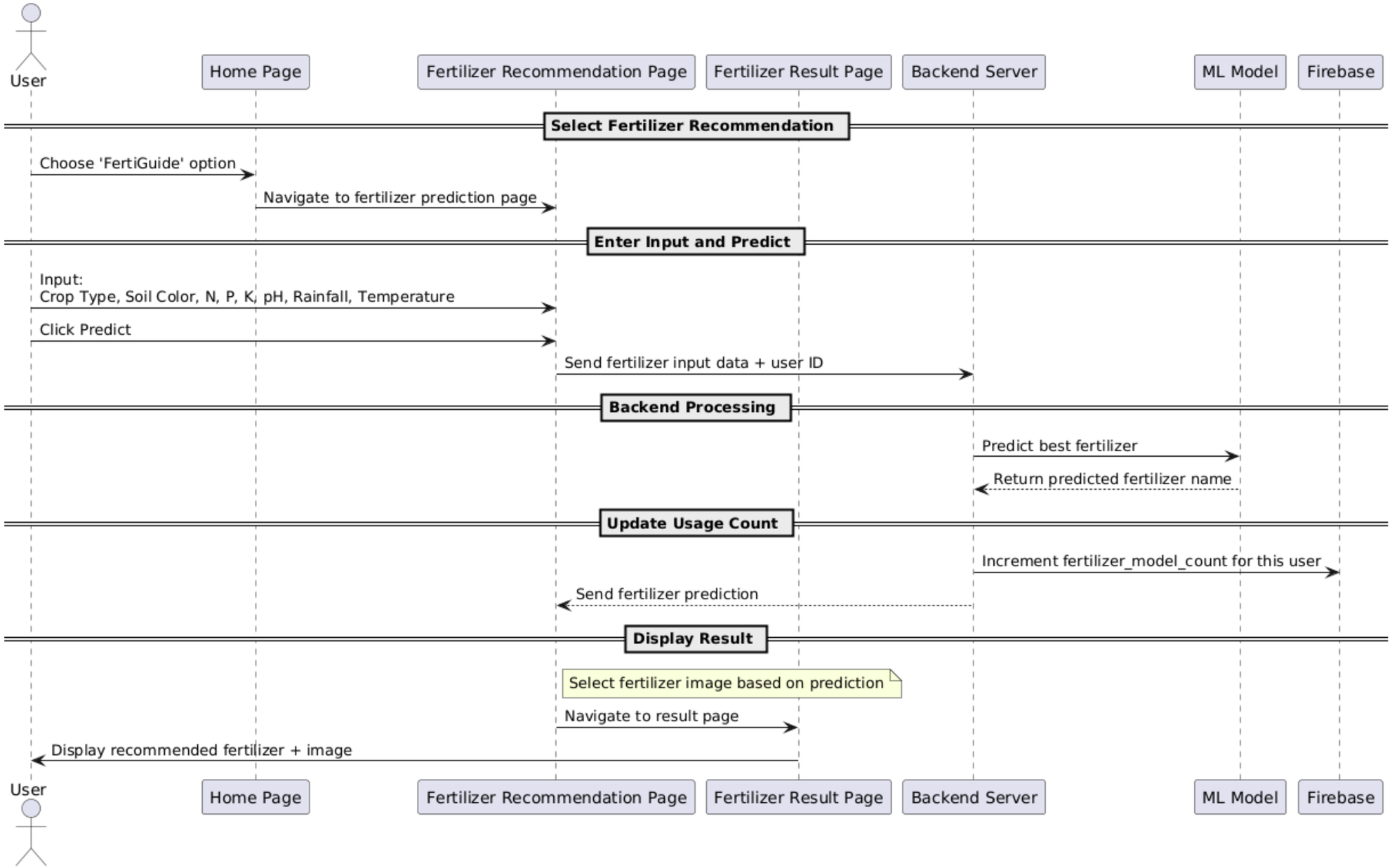
Sequence Diagram for History



Sequence Diagram for Crop Recommendation Model



Sequence Diagram for Fertilizer Prediction Model



DATASETS

In this model we use 3 datasets:

1- new plant disease dataset (images).

  1.1- supplement_info(csv).

  1.2- disease_info(csv).

2- Crop Recommendation dataset(csv).

3- Crop and fertilizer dataset(csv).

Model 1 – Plant Disease Detection

🗂️ **Dataset Overview (New Plant Diseases Dataset)**
Source: Kaggle
Size: 86,818
Classes: 38 (includes healthy and diseased leaf categories)
Crops Covered: Tomato, Potato, Corn, Apple, etc.
Split:
Training: 69,428
Validation: 17,336
Testing: 54
Image Size: Resized to 128×128 pixels
✅ **Purpose: Used for image-based classification of plant health conditions.**

📚 **Supporting Dataset (Plant Disease & Supplement Info Dataset)**
Includes:

1. **Disease name**
2. **Symptoms description**
3. **Treatment steps**
4. **Reference image URLs**
5. **Goal: Provides educational feedback and actionable treatment recommendations post-diagnosis.**

model implementation

ResNet-50
Algorithm: ResNet-50 (Deep Convolutional Neural Network)
Architecture: 48 conv layers, MaxPooling, AvgPooling, and fully connected layers
Key Feature: Uses residual blocks to solve the vanishing gradient problem, allowing deeper and more stable training
Purpose: High-accuracy classification of plant diseases from leaf images

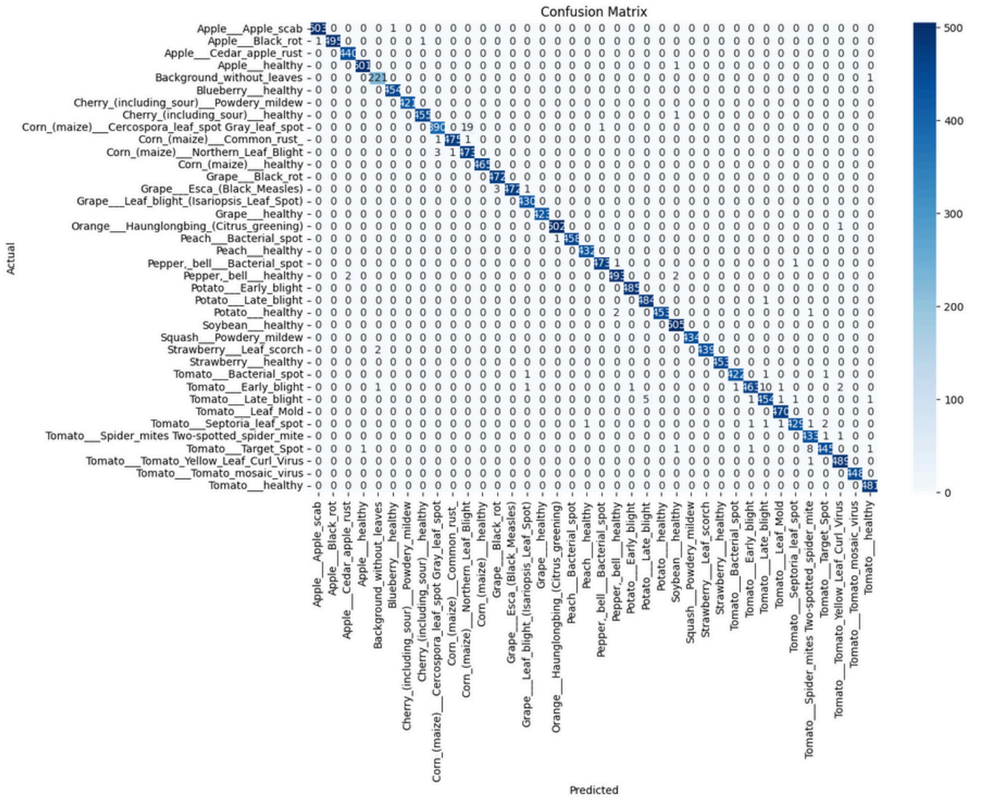
⚙ **Training Details**

Activation Function: Softmax
Optimizer: AdamW (learning rate: 0.001)
Train/Validation Split: 80% / 20%
Reproducibility: Random seed set for consistent training results

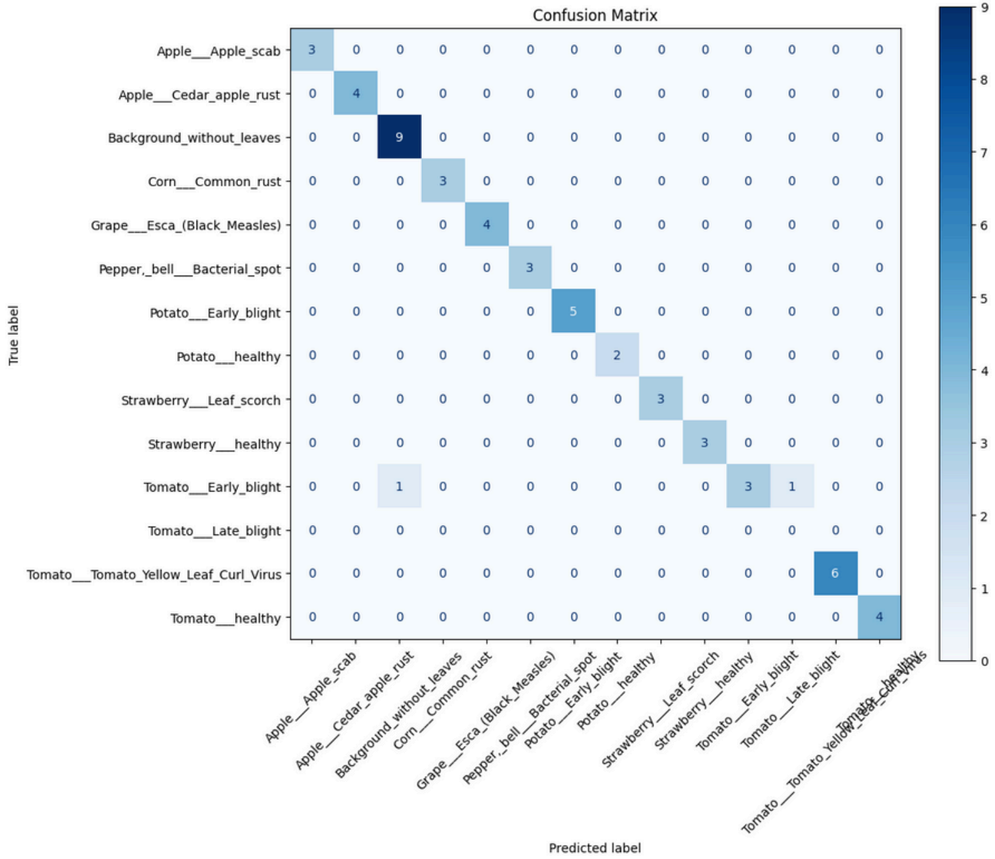
```
[Validation] Epoch 17, Batch 210: Loss = 0.0000
[Validation] Epoch 17, Batch 220: Loss = 0.0597
[Validation] Epoch 17, Batch 230: Loss = 0.0042
[Validation] Epoch 17, Batch 240: Loss = 0.0002
[Validation] Epoch 17, Batch 250: Loss = 0.0009
[Validation] Epoch 17, Batch 260: Loss = 0.0709
[Validation] Epoch 17, Batch 270: Loss = 0.0007
[Validation] Epoch 17, Batch 280: Loss = 0.0023
Epoch 17, Train Accuracy: 99.33%, Validation Accuracy: 99.42%
✅ Best model checkpoint saved at /content/best_model_checkpoint_epoch_17.pth
```

Epoch 17, Train Accuracy: 99.33%, Validation Accuracy: 99.42%

validation confusion matrix



test confusion matrix



Test Accuracy: 96.30%

Average Validation Loss: 0.0191

Validation Accuracy: 99.42%

Here We Show Inputs And Predicted Outputs

Strawberry Leaf scorch

Breif Description

Leaf scorch symptoms are very similar to the early stages of common (Mycosphaerella) leaf spot, with irregular dark purple spots being scattered over the upper leaf surface. As the spots enlarge, they begin to look like drops of tar, and are actually the accumulations of black fruiting bodies (acervuli) of the fungus. The centers of the spots remain purple (in Mycosphaerella leaf spot they are white) and there is no well-defined lesion border. In heavy infections, these regions coalesce and the tissue between the lesions often takes on a purplish to bright red color that is dependent on cultivar, temperature, or other factors. The leaves eventually turn brown, dry up, and curl at the margins giving the leaf a scorched appearance. Examination of the acervuli and conidial morphology can help to distinguish between leaf spot and leaf scorch at this advanced stage of disease. On the upper leaf surfaces of leaf scorch lesions, the acervuli are dark with glistening spore masses and dark apothecia. Petiole lesions are elongate, sunken, with a purplish to brown color and can kill the leaf by girdling the petiole. Runners, fruit stalks, fruit and caps can also become infected. Plants may become weakened and the number and vigor of crowns reduced. Infection predisposes the plants to winter and drought stress. In severe infestations, flowers and fruit may die.

Prevent This plant disease by follow below steps

"While leaf scorch on strawberry plants can be frustrating, there are some strategies which home gardeners may employ to help prevent its spread in the garden. The primary means of strawberry leaf scorch control should always be prevention. Since this fungal pathogen overwinters on the fallen leaves of infected plants, proper garden sanitation is key. This includes the removal of infected garden debris from the strawberry patch, as well as the frequent establishment of new strawberry transplants. The creation of new plantings and strawberry patches is key to maintaining a consistent strawberry harvest, as older plants are more likely to show signs of severe infection.",

Supplements :

SWISS GREEN ORGANIC PLANT GROWTH PROMOTER
STRAWBERRY Fertilizer

second model (Crop recommendation)

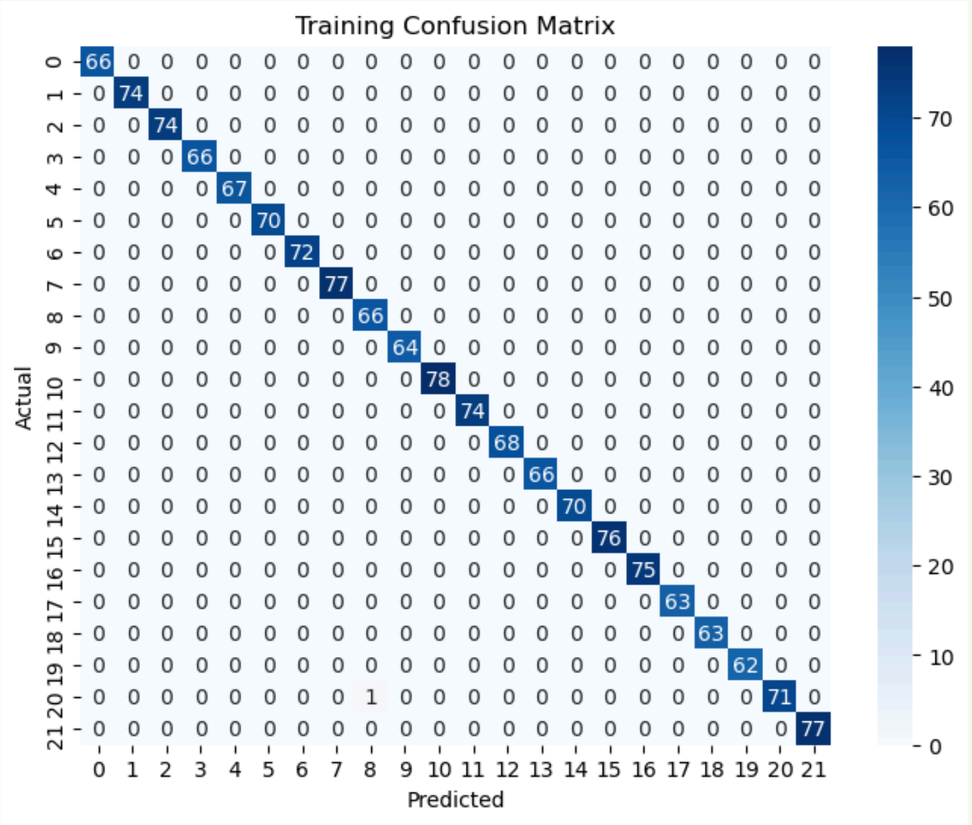
📊 Dataset Overview

- **Name: Crop Recommendation Dataset**
- **Size: 2,200 instances (1540 train,594 val,66test)**
- **Features (7 total):**
  - **Soil nutrients: Nitrogen (N), Phosphorus (P), Potassium (K) [kg/ha]**
  - **Climate factors: Temperature (°C), Humidity (%), Rainfall (mm)**
  - **Soil pH**
- **Target: Multi-class classification (e.g., rice, maize, cotton, etc.)**
- **Split: 70% training – 27% validation-3% testing**

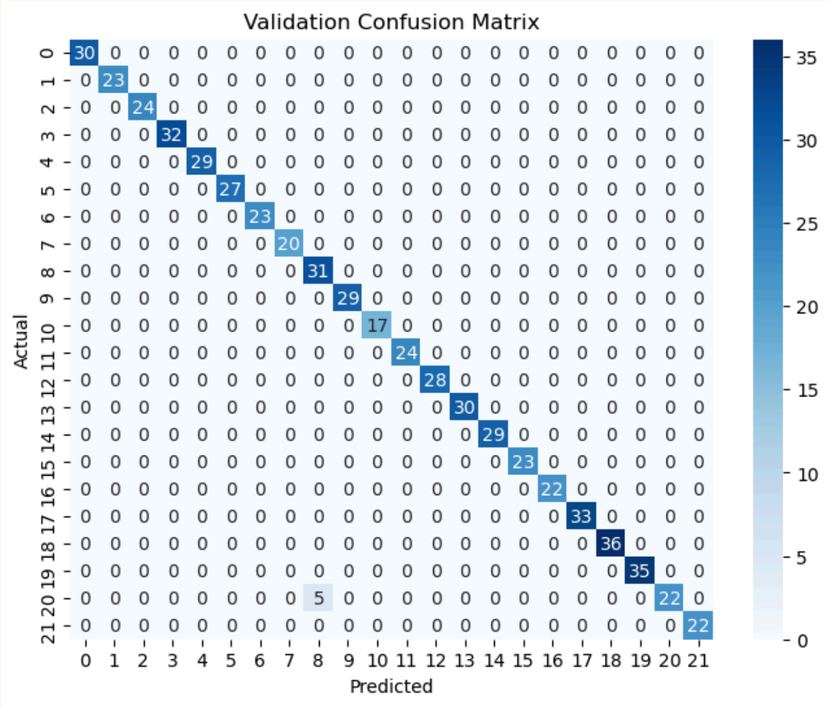
- **Algorithm:** **Random Forest (Ensemble Machine Learning Model)**
- **Why Used:**
- **High prediction accuracy**
- **Handles complex and varied environmental data**
- **Robust under different farming conditions**

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn import metrics
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

random_forest = RandomForestClassifier(random_state=2, n_jobs=-1)

param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [3, 5, 10, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'criterion': ['gini', 'entropy'],
}


grid_search = GridSearchCV(random_forest, param_grid, cv=5, scoring='accuracy', n_jobs=-1, verbose=1)
grid_search.fit(Xtrain, Ytrain)


best_params = grid_search.best_params_



best_random_forest = RandomForestClassifier(**best_params, random_state=2, n_jobs=-1)
best_random_forest.fit(Xtrain, Ytrain)
predicted_values_tarin = best_random_forest.predict(Xtrain)
```

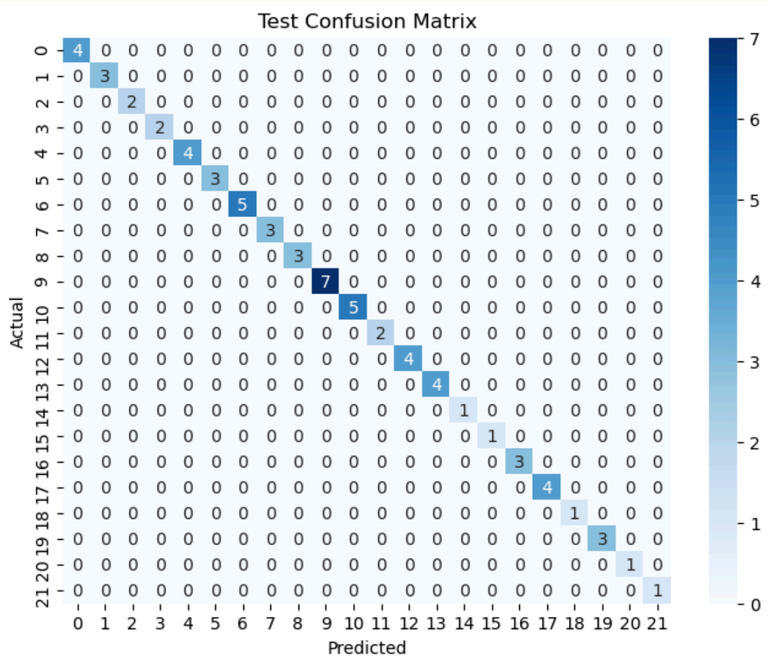
```
Training Accuracy: 99.94%
```



```
Validation Accuracy: 99.16%
```




```
Test Accuracy: 100.00%
```



Output

```
import joblib
import numpy as np
model = joblib.load("Random_forest.joblib")

data = np.array([[3, 26, 40, 24, 90, 6, 112]])

prediction = model.predict(data)
print("Prediction:", prediction)


Prediction: ['pomegranate']
```

🌿 Fertilizer Recommendation Model

📊 Dataset Overview

Name: Crop and Fertilizer Recommendation Dataset
Size: 4,513 records
Key Features:
Soil and Environment: Nitrogen (N), Phosphorus (P), Potassium (K), pH, Rainfall, Temperature
Categorical Data: District Name, Soil Color (encoded numerically)
Target: Recommended Fertilizer type (multi-class classification)
Additional Fields: Recommended Crop, Educational Link
Diversity: Data collected from multiple districts covering various geographical and climate conditions

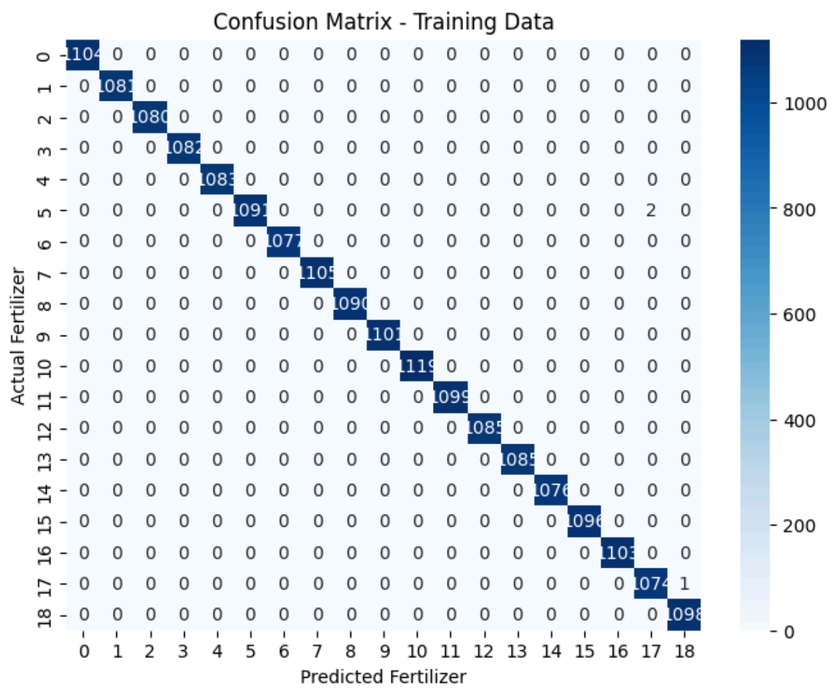
🧠 Model Used: XGBoost

We selected XGBoost due to its:

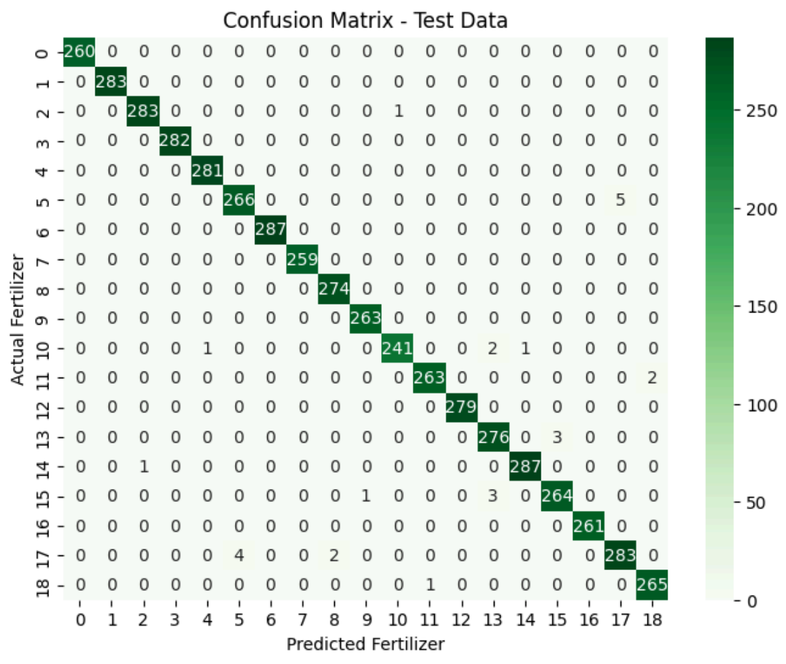
1. **High accuracy and efficiency**
2. **Excellent performance on imbalanced datasets**
3. **Built-in regularization that reduces overfitting and improves generalization**

```python
from xgboost import XGBClassifier


model = XGBClassifier(n_estimators=300, learning_rate=0.5, max_depth=5, random_state=42)
model.fit(X_train, y_train)


y_train_pred_encoded = model.predict(X_train)
y_train_pred = le.inverse_transform(y_train_pred_encoded)
y_train_actual = le.inverse_transform(y_train)

train_accuracy = accuracy_score(y_train_actual, y_train_pred)
print(f"✅ Training Accuracy: {train_accuracy:.2%}")
print("\n📊 Classification Report (Training):\n")
print(classification_report(y_train_actual, y_train_pred))



plt.figure(figsize=(8, 6))
sns.heatmap(confusion_matrix(y_train_actual, y_train_pred), annot=True, fmt="d", cmap="Blues")
plt.xlabel("Predicted Fertilizer")
plt.ylabel("Actual Fertilizer")
plt.title("Confusion Matrix - Training Data")
plt.show()
```

```
Training Accuracy: 99.99%
```




```
Test Accuracy: 99.48%
```



Output

```python
import joblib
import numpy as np

model = joblib.load("fertilizer_prediction_model.pkl")

input_data_json = {
    "Soil_color": "Black",
    "Nitrogen": 75.00,
    "Phosphorus": 50.00,
    "Potassium": 100.00,
    "pH": 6.50,
    "Rainfall": 1000.00,
    "Temperature": 20.00,
    "Crop": "Maize"
}
prediction_result = predict_fertilizer(input_data_json)
print(prediction_result)
```

```
{'predicted_fertilizer': 'Urea'}
```

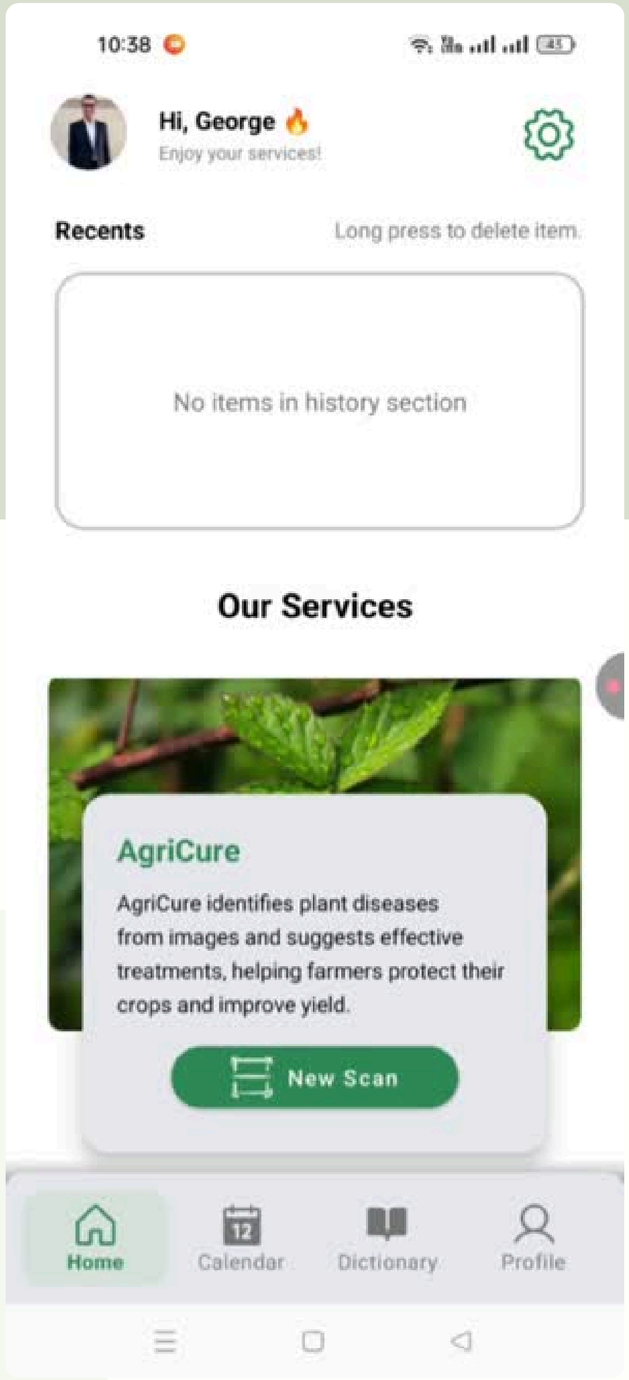
Demo of App

Some shots from our app

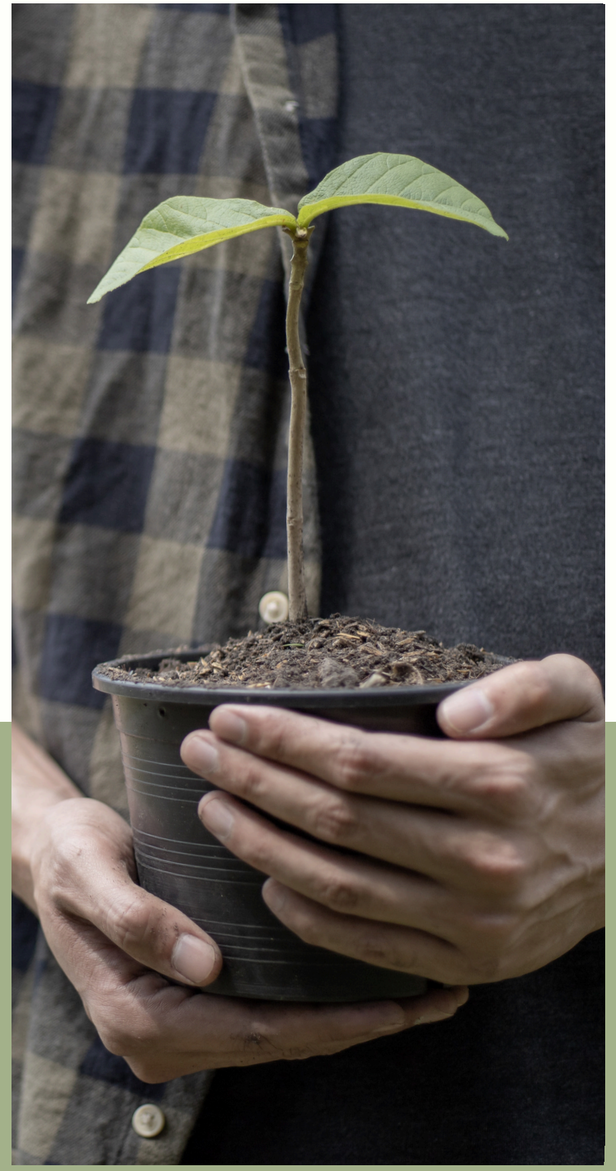
Light Mode

Dark Mode


7:58
Photos
Part (1)
UI & Feature Overview
PLANTY
CARE
Close all


10:38
Hi, George
Enjoy your services!
Recents
Long press to delete item.
No items in history section
Our Services
AgriCure
AgriCure identifies plant diseases from images and suggests effective treatments, helping farmers protect their crops and improve yield.
New Scan
Home
Calendar
Dictionary
Profile

Future Improvements – Planty Care



- **GPS Alerts:** Notify nearby users of potential disease outbreaks.
- **Smart Sensor Integration:** Real-time data from soil moisture, temperature, pH, and light sensors.
- **Real-time Image Scanning using Ai:** To enhance user experience by enabling real-time plant disease detection through the phone's camera, eliminating the need for manual image uploads.
- **In-App Agricultural Marketplace:** Buy recommended fertilizers and pesticides directly.
- **Smart Scheduling System:** Helps plan irrigation, fertilization, and harvesting tasks.
- **Scalable Disease Progress Monitoring:** Visual reports dynamically track treatment impact and disease progression over time, designed for expanding datasets.

Conclusion

Planty Care is an AI-powered mobile app that helps farmers detect plant diseases, choose the right crops, and get fertilizer recommendations — all in one place.

With advanced models like ResNet-50, Random Forest, and XGBoost, it delivers fast, accurate support to improve productivity and promote sustainable farming — anytime, anywhere.

www.planty care.com

THANK YOU!

Turning Small Steps into Big Impact