Plant Leaf Disease Prediction

Team Members: -

Yeddula Reddy Krishna Reddy - 11700539

Mukkamala Harichandana - 11710326

Yenugula Ruthiksha Reddy - 11714976

**Hemanth Janapala - 11657605**

Pavani Pabboju – 11713231

1. Web Application Screenshots

The following screenshots showcase the interface and functionality of the Plant Disease Detection web application:

1.1: Homepage

This above screenshot shows the home page of the application. It has self-explanatory links to upload a plant leaf image and to check the prediction results. Preferred formats of these files are; PNG or JPG.
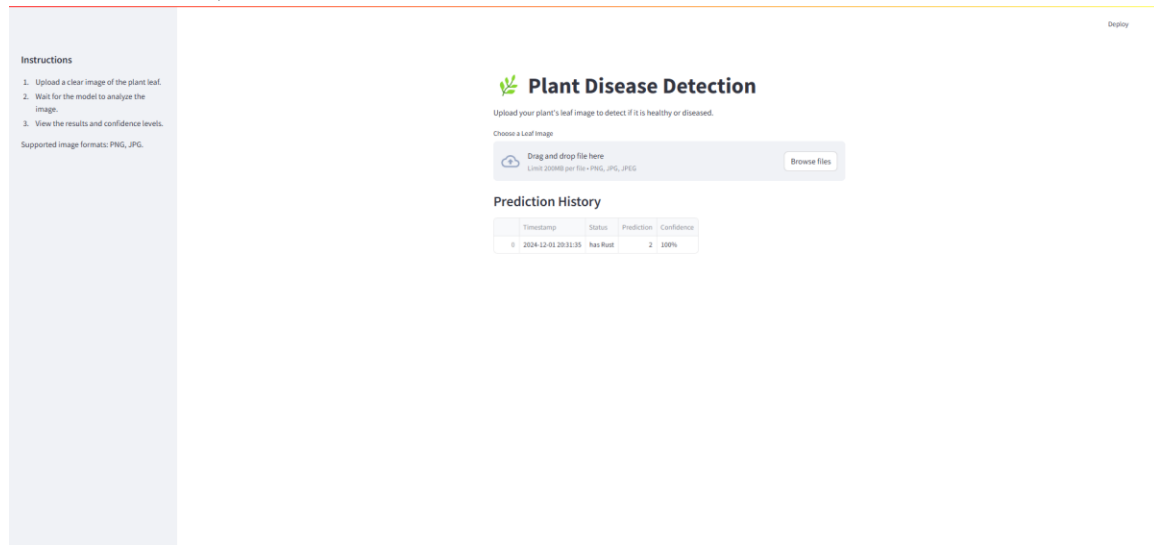

1.2: File Upload

Here, the user is allowed to upload an image of a plant leaf here. This image shall be used in training of the used model to detect diseases. Basically, a clean, which makes the design more informative, and responsive design simplifies it.
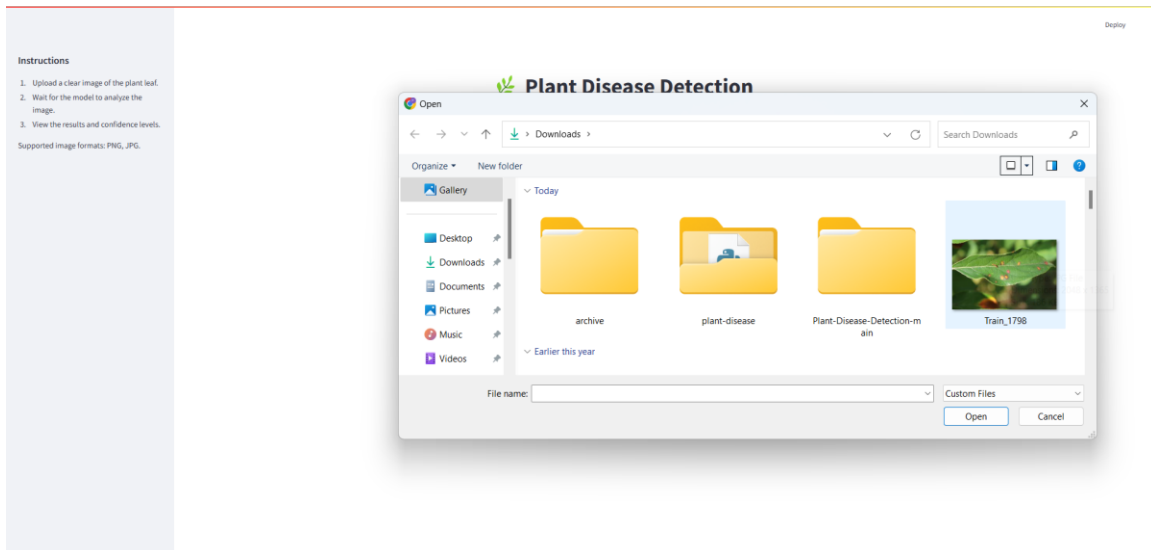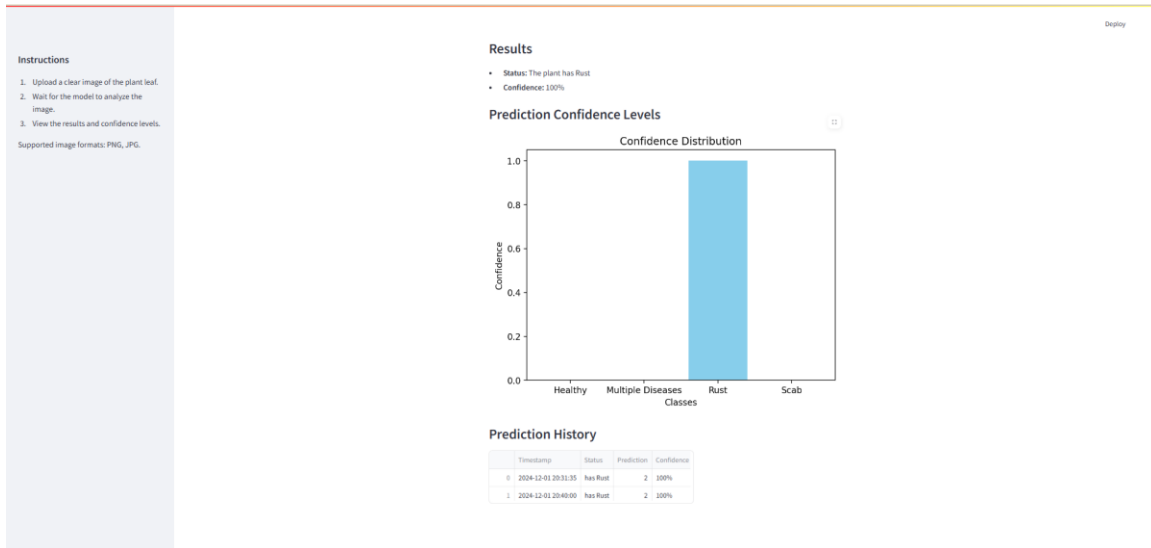
1.3: Prediction Results

This screenshot displays output prediction details ranging from the disease class that was identified to confidence levels of prediction. The estimated result by the model is shown next to the posted image for comparison.
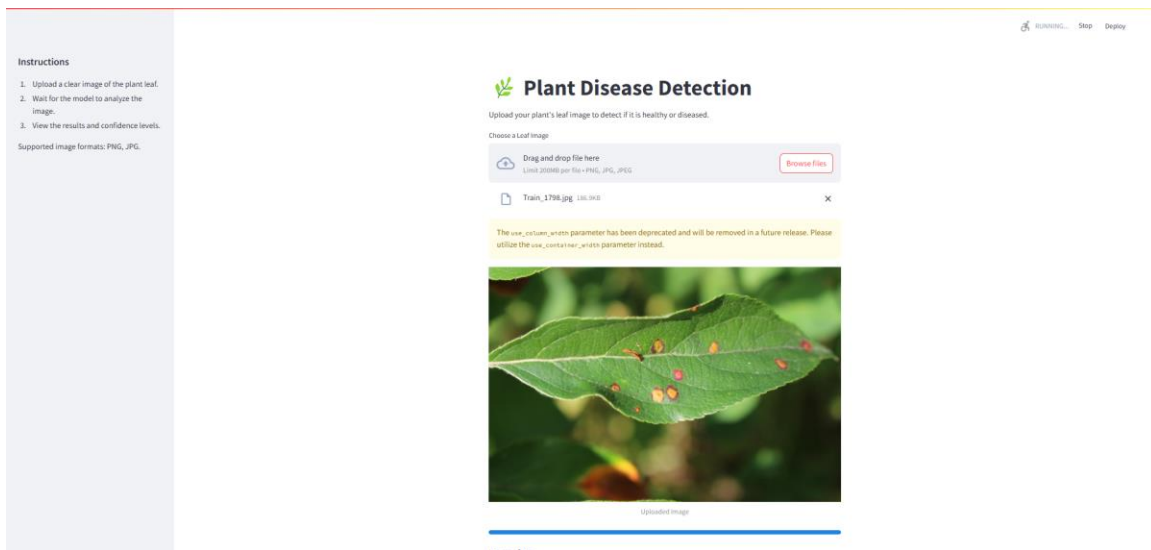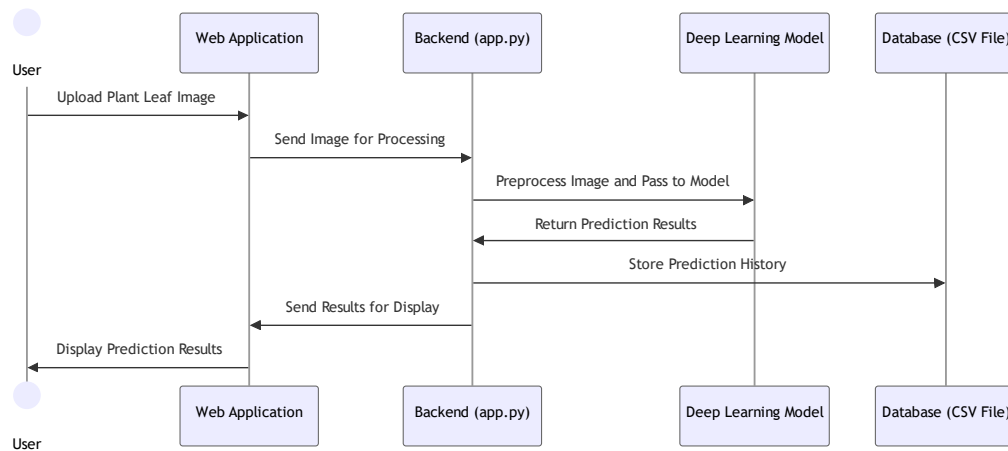

1.4: Prediction History

This table save timestamp, predicted status and the confidence level for each and every prediction made. This makes it possible for users to refer to past analysis results for either future use or even for presentation.

Instructions

1. Upload a clear image of the plant leaf.
2. Wait for the model to analyze the image.
3. View the results and confidence levels.

Supported image formats: PNG, JPG.

Results

- **Status:** The plant has Rust
- **Confidence:** 100%

Prediction Confidence Levels

Confidence Distribution

Prediction History

	Timestamp	Status	Prediction	Confidence
0	2024-12-01 20:31:35	has Rust	2	100%
1	2024-12-01 20:40:00	has Rust	2	100%

2. Communication Diagram

As seen in the following communication diagram the Plant Disease Detection system comprises of different components or entities. It includes the following steps:

1. This is done by the user uploading a plant leaf image through the web application developed here.

2. This web application passes the image to the backend that is the app.py file for further processing.

3. The backend resizes the image as well as normalizes it and then presents it to the trained model.

4. The model looks into an image and estimates the class of the disease along with the confidence level.

5. The backend saves the prediction history in a database that is a CSV file.

6. The web application also retrieves the prediction results and presents them to the user.

These interactions are illustrated in the following diagram: [Diagram is attached below]