

**Project School Certificate**

**Title :** Crop Doctor(Plant Disease Detection - Web Page)

**Faculty Incharge :** Dr .Devika Rubi

**Session Duration :** 10/09/2022- 17/12/2022

Name : Vemula Rishi Preetham

Roll Number : 20BD1A6760

Class : III-CSD-A

Signature of Faculty                                 Signature of student

OBJECTIVE

* To Design a responsive web application for automated plant disease detection.
* To help farmers to detect the diseases of plants

TECHNICAL DESCRIPTION

* Plant disease is identified by using the RESNET 50 Deep Learning Model.
* The frontend provides farmers information and predicted results which is developed by HTML, CSS, JavaScript.
* MongoDB is used to store the data and Flask as a backend framework.
* The deep learning model is trained using tensorflow library.

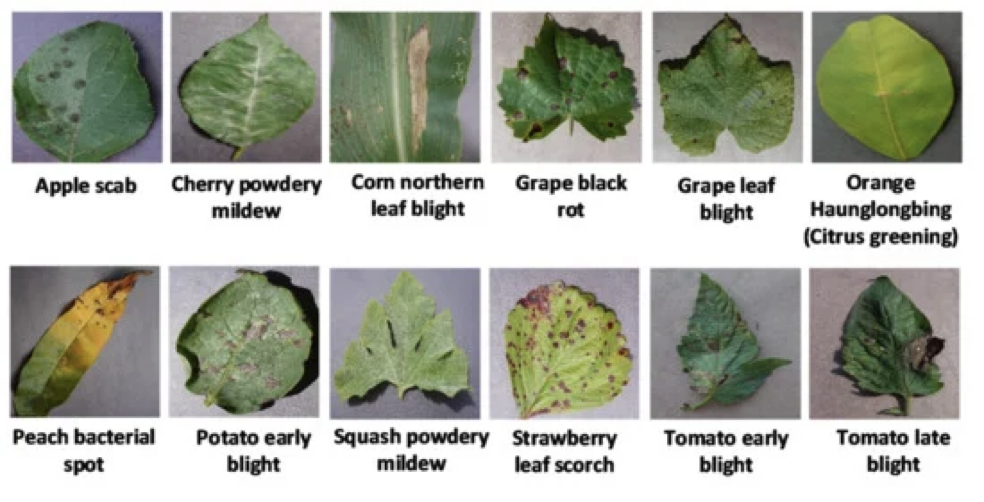
DATASET DESCRIPTION

Name:- New Plant Disease Dataset

Source:- Kaggle

Link:- <https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset>

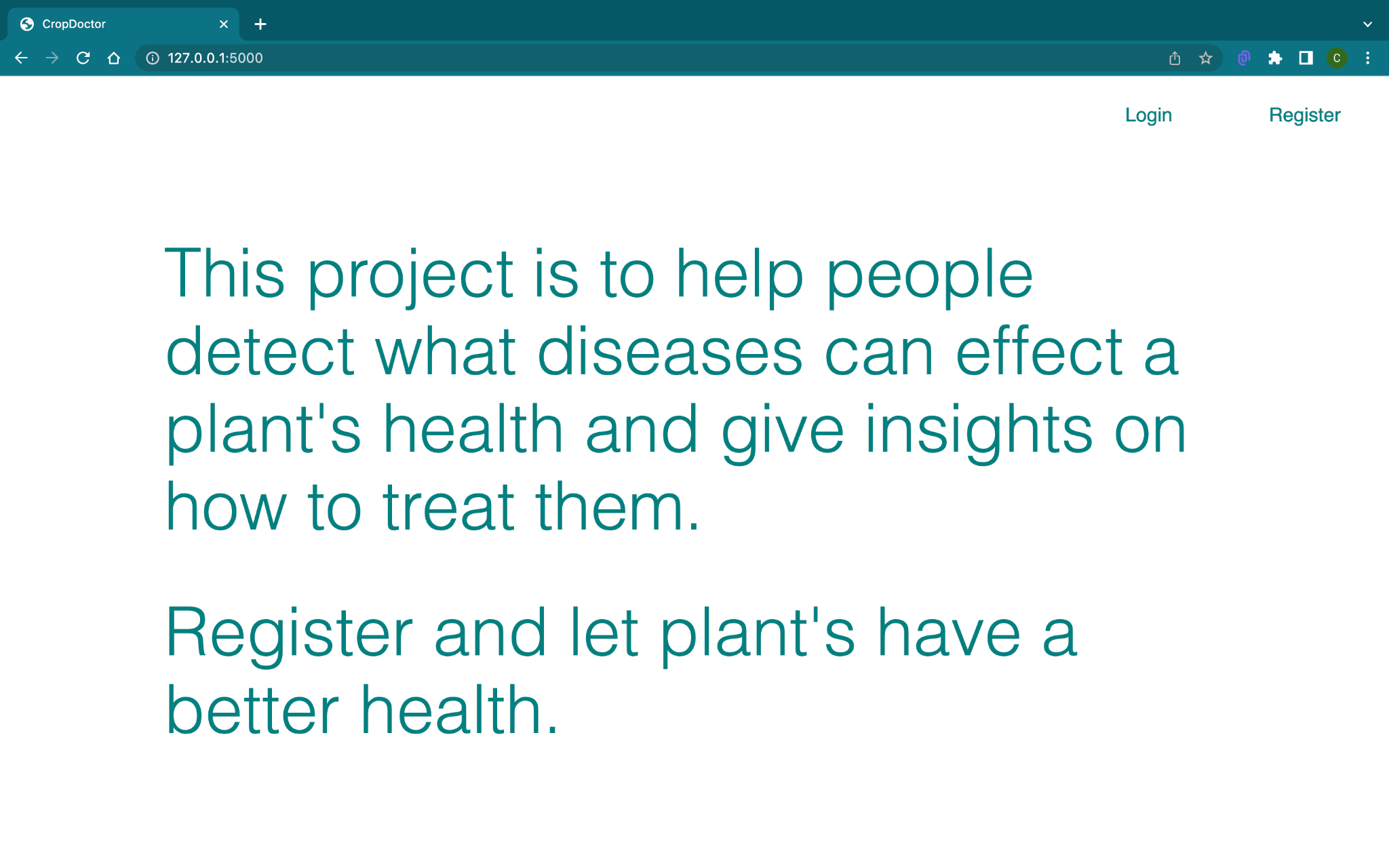
* This dataset consists of about 87000 RGB images of healthy and diseased crop leaves which is categorized into 38 different classes. The total dataset is divided into 80/20 ratio of training and validation set preserving the directory structure.



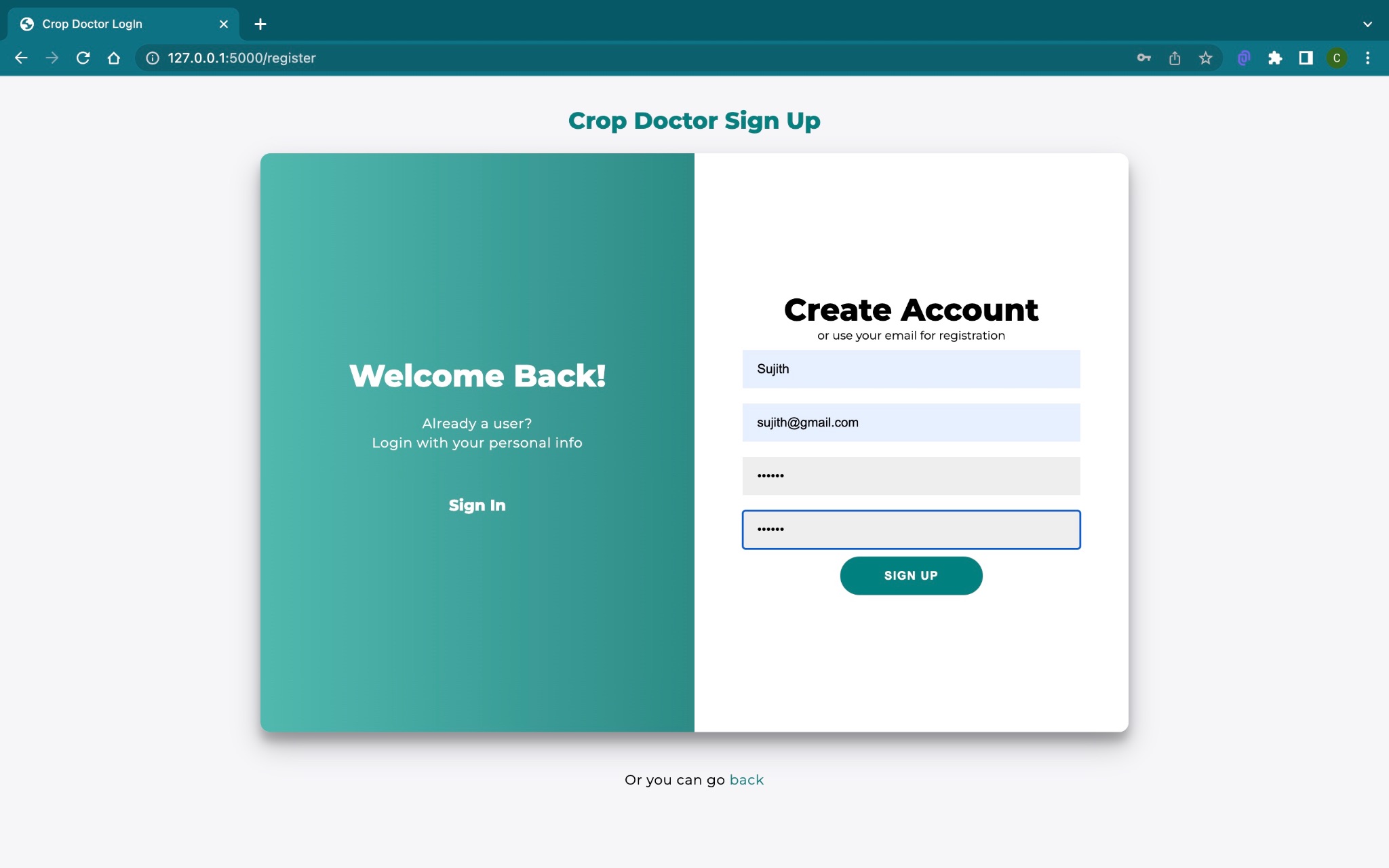
IMPLEMENTATION

* Home Page :- This is the landing page for the project. The page describes the project and asks the user to proceed to the Login/Register page.

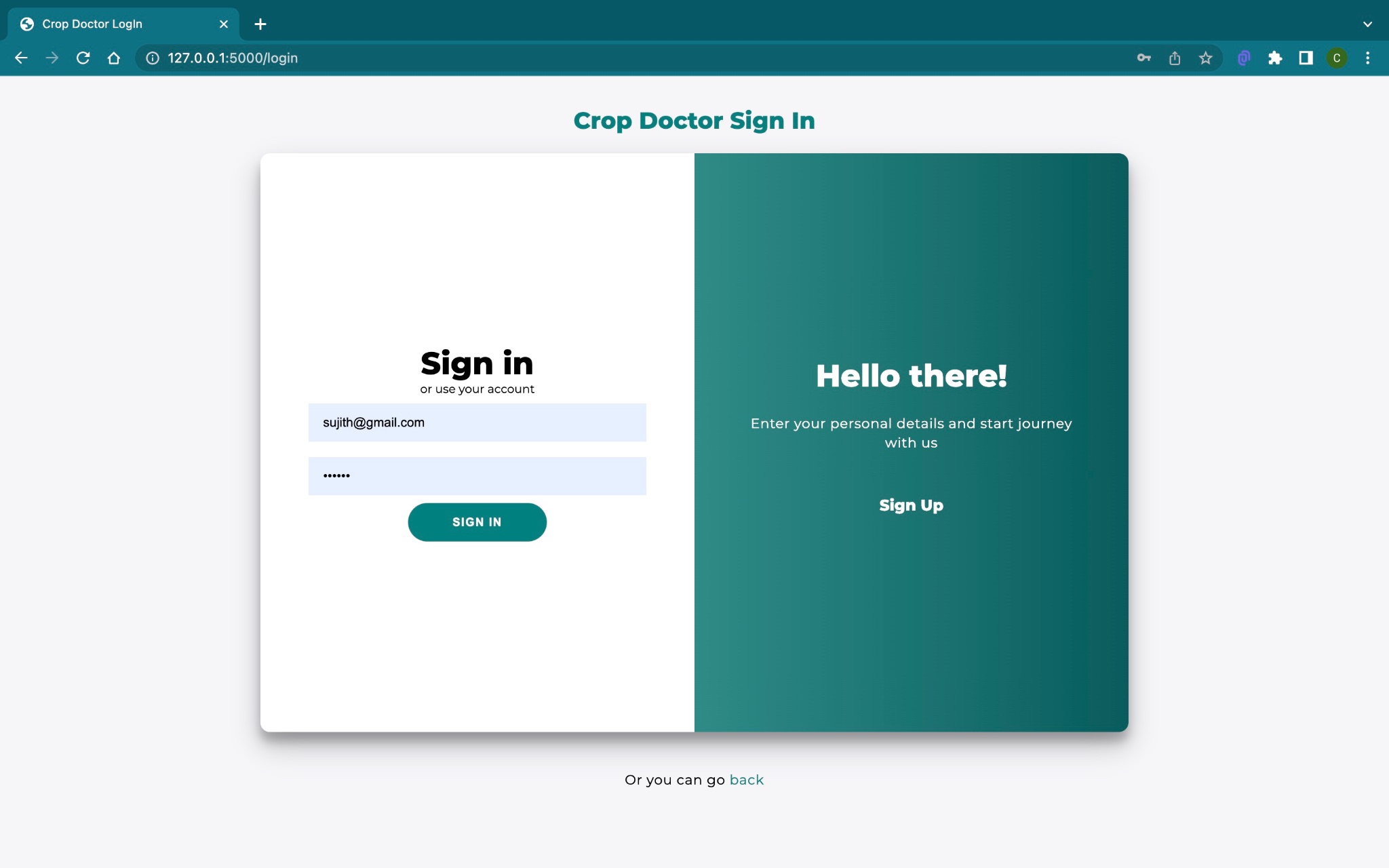




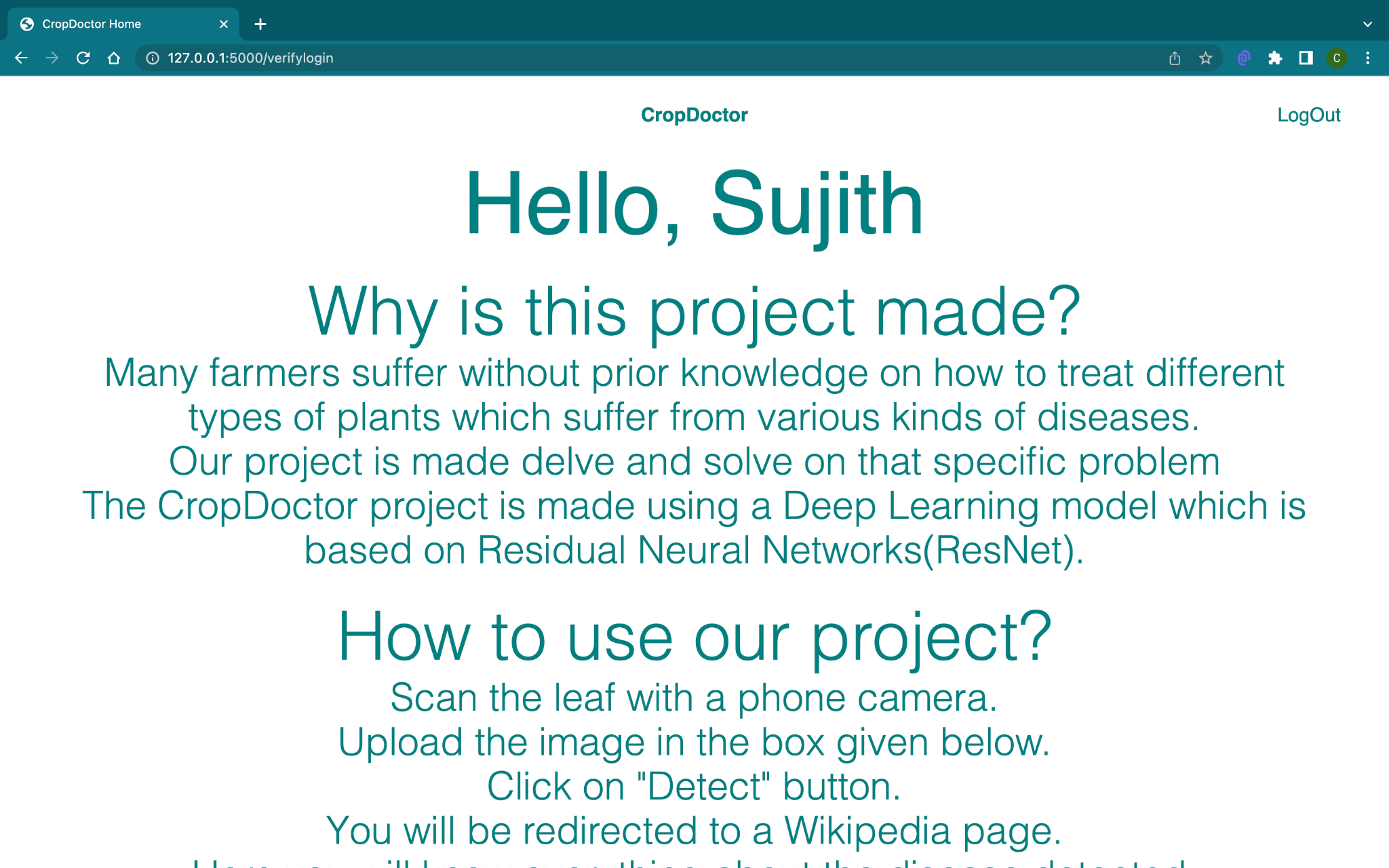
* Register Page:- If the user is new he can register his email and password in this page. These details will be passed to Flask and creates a new user and adds these credentials to the user database in MongoDB.

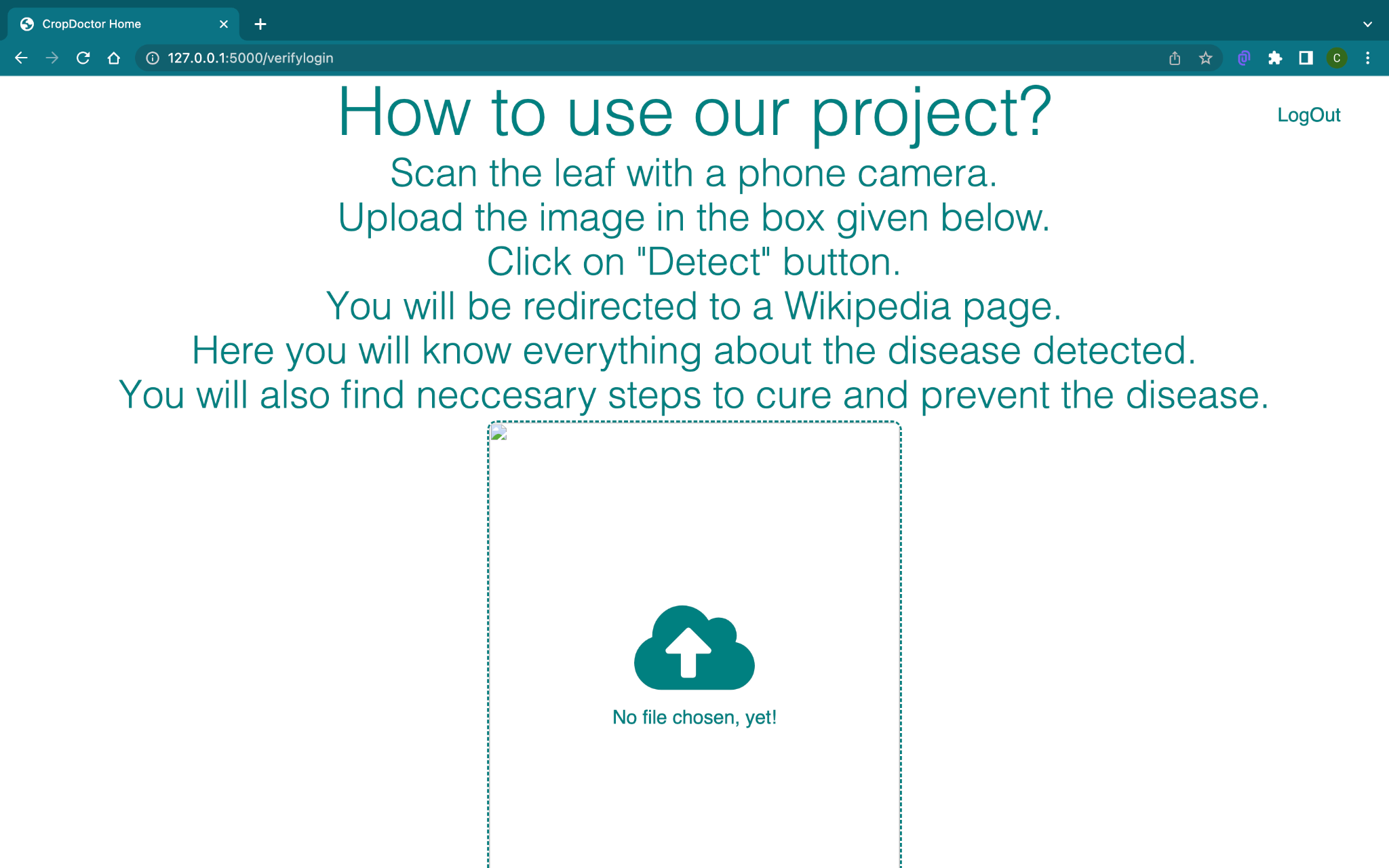


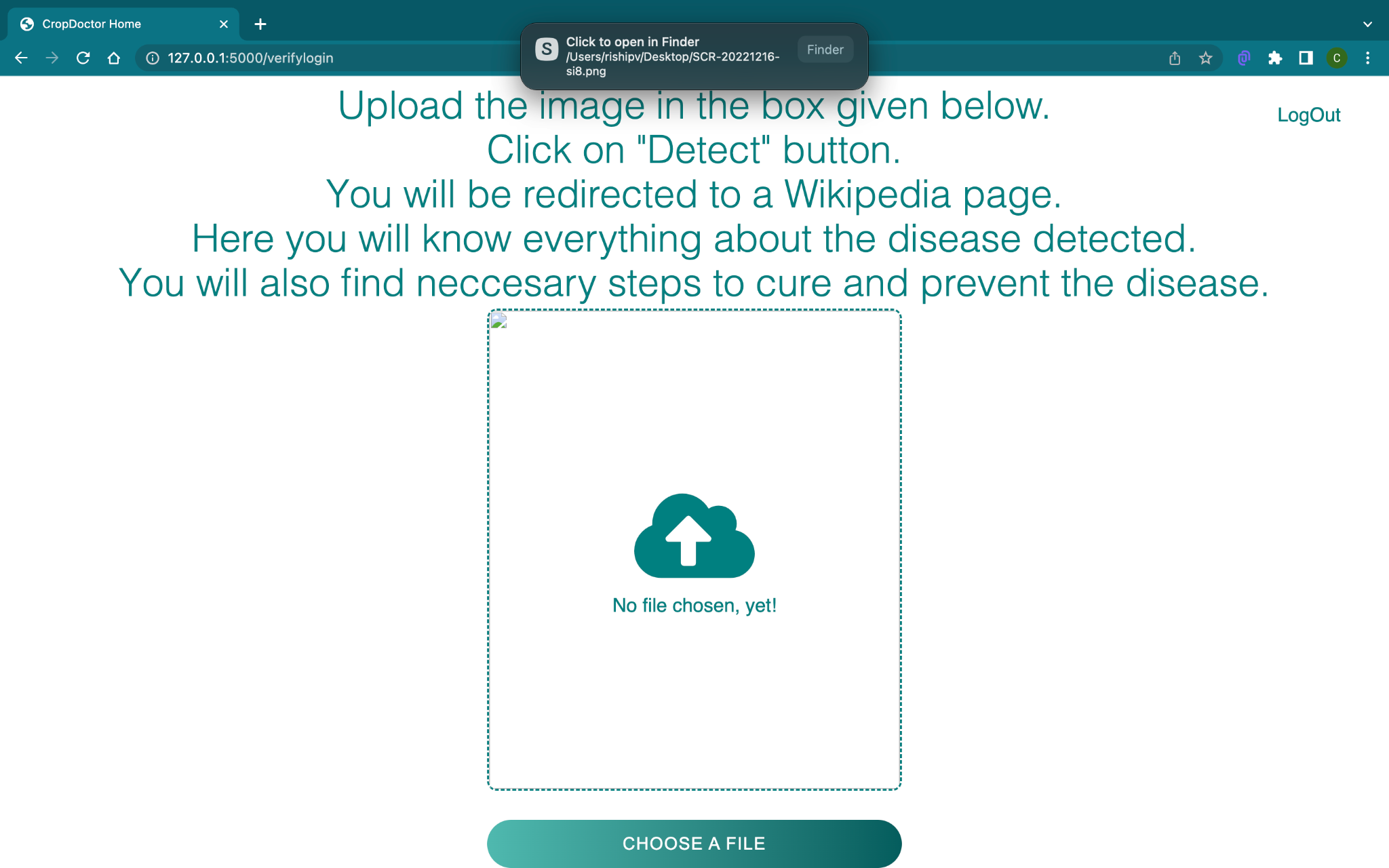
* Login Page:- The user will be asked to login. This functionality will be implemented using Flask and MongoDB. Email-id and password from the user will be passed to Flask and will be checked present in MongoDB. This helps if the user exists. If the user doesn’t exist, an error is thrown otherwise, all the credentials are matched. If not, then a relevant error will be thrown. If everything matches it will navigate to our prediction page.



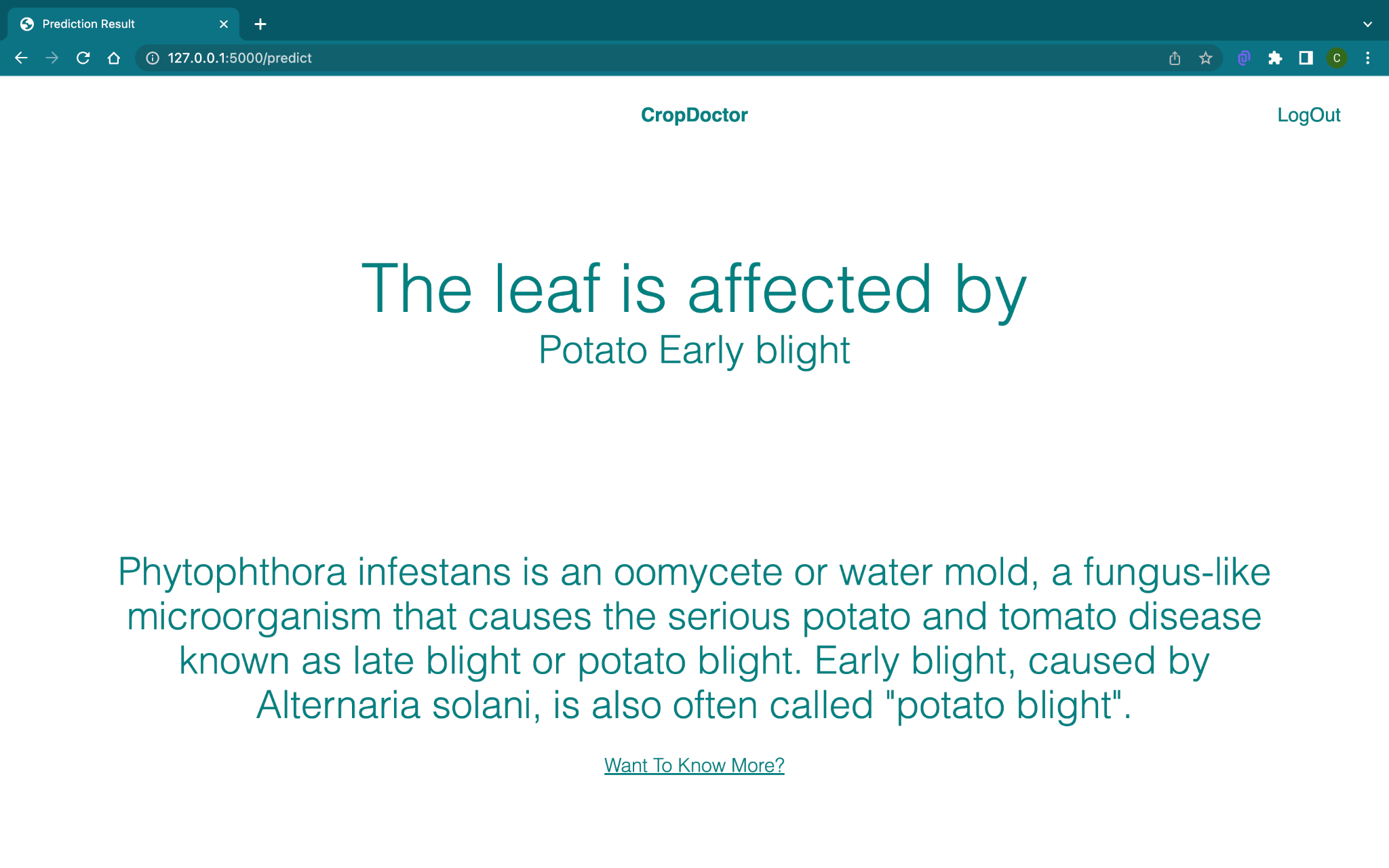
* Home Page :- This is the most important page in our project.In this page we describe why the project is made and give instructions for the user to work with our project.Here we upload an image from the Upload section of the page.







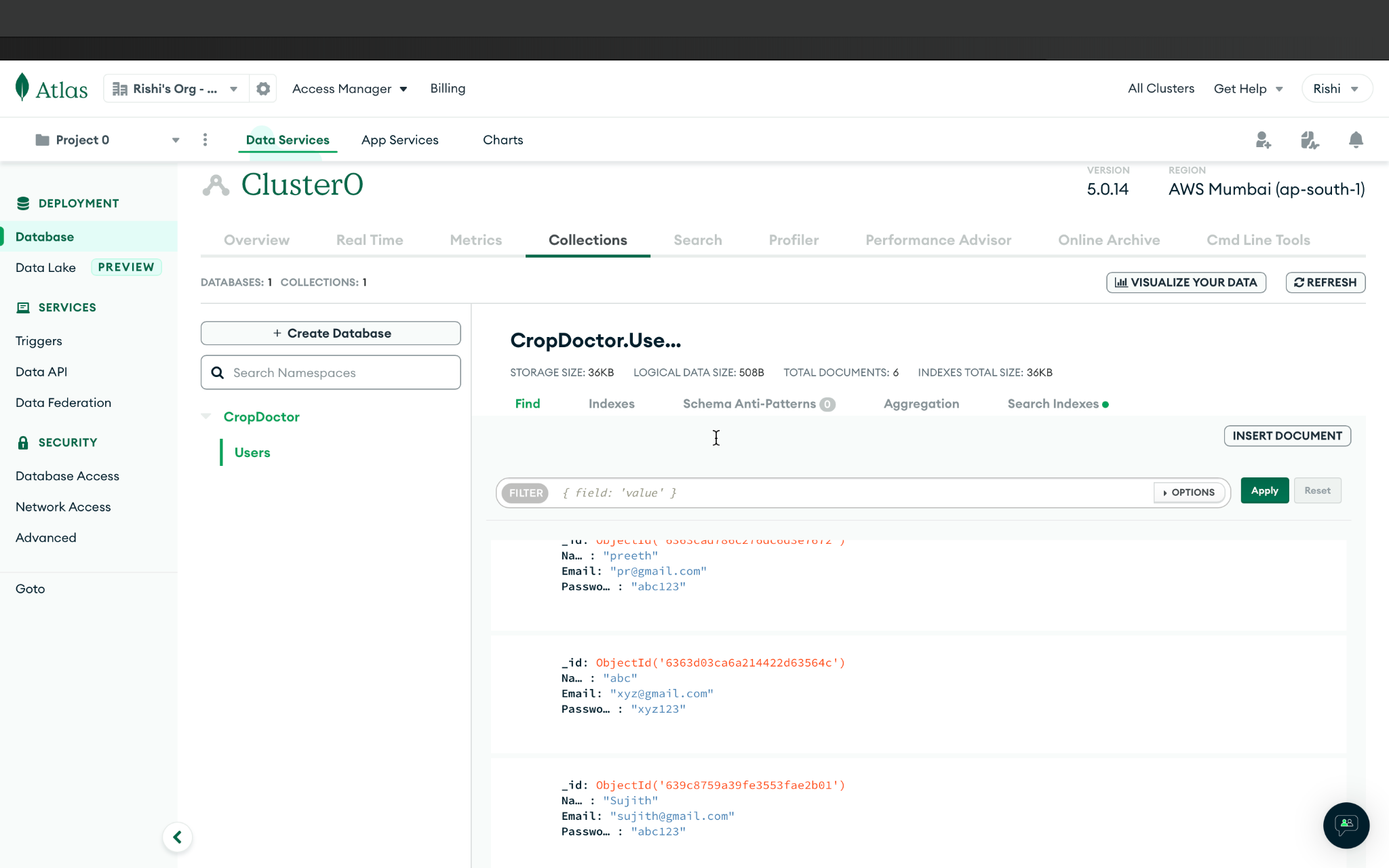
* Prediction Page :- After passing it to our model, a prediction is shown. The user will be able to get the name of the disease and brief description of the disease.



* Wiki Page :- If the user wishes to view a wikipedia page of our predicted disease. Clicking on “Want to know more?”on the prediction page will take him to the wikipedia page of our predicted disease.



* Database :- Any credentials used to Register to the project will be saved in MongoDB. Login credentials are verified by checking the credentials present in the database.



PLATFORM USED

* Visual Studio Code for Front-end and Back-end development.
* Kaggle for Dataset
* Jupyter lab for model building

SYSTEM CONFIGURATION

* Name:- Apple MacBook Air
* Processor :- Apple Silicon M1
* RAM :- 8GB Unified Memory

BUILDING THE MODEL

**Model Summary:-**

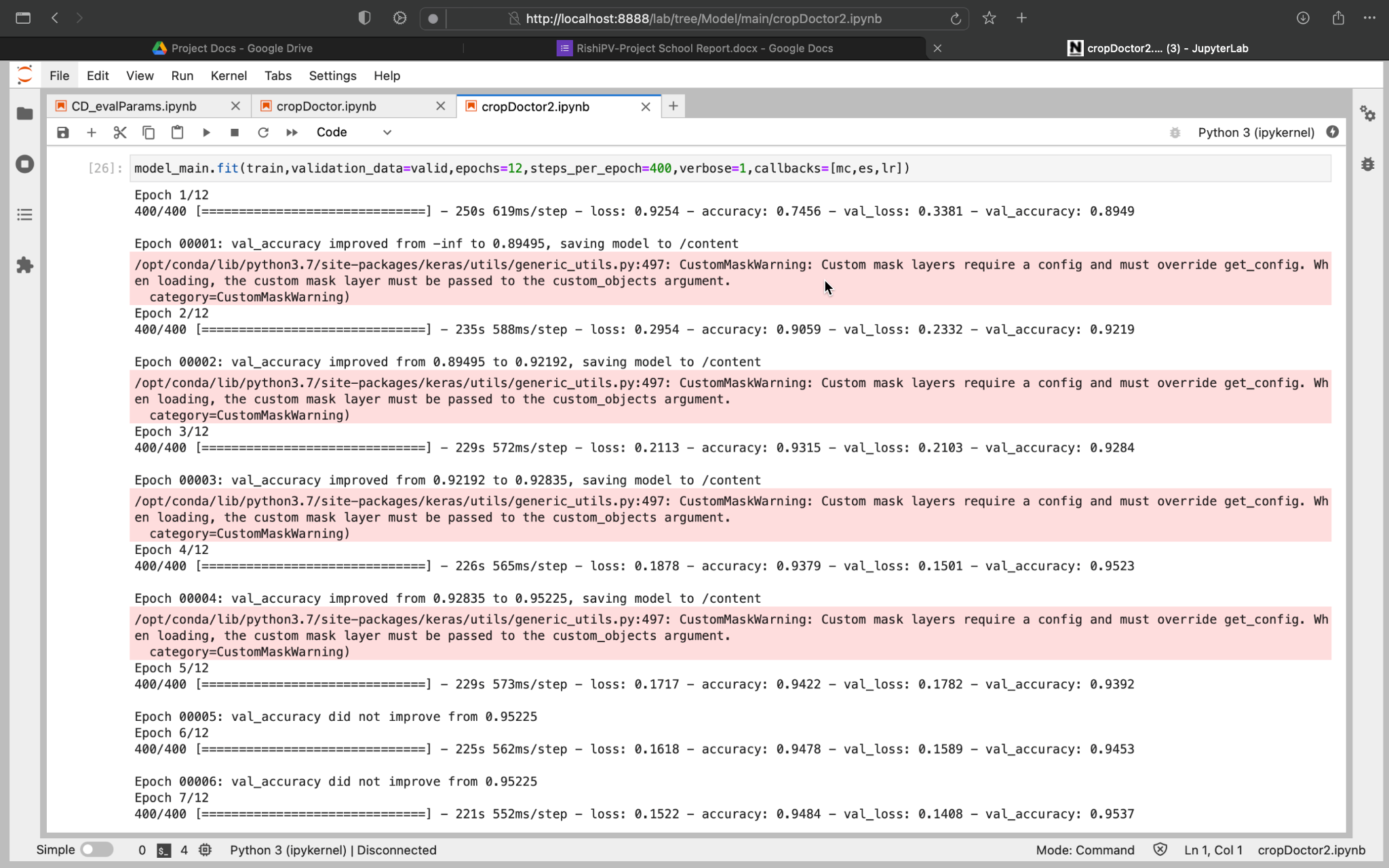
****

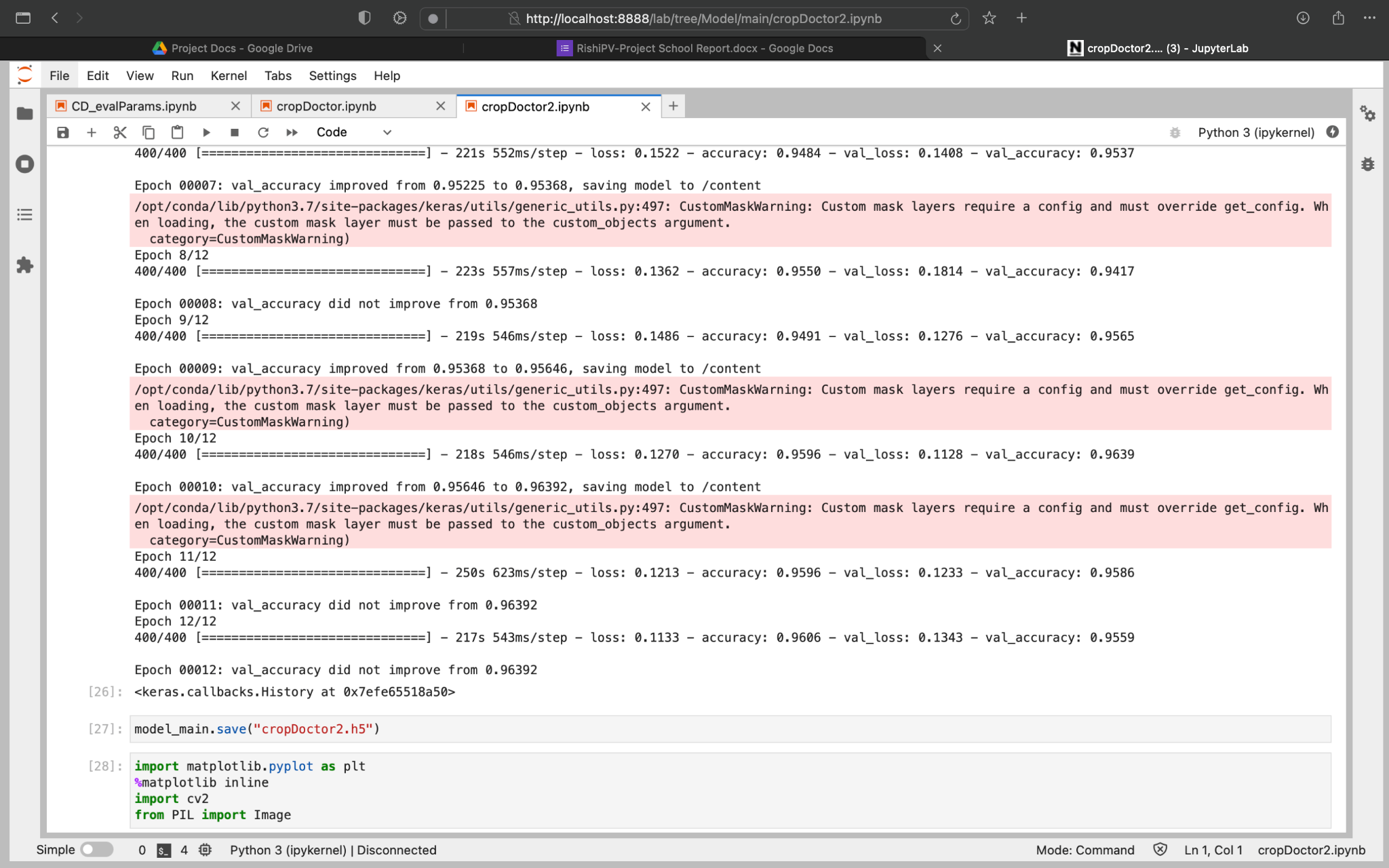
No. of. Epochs:- 12

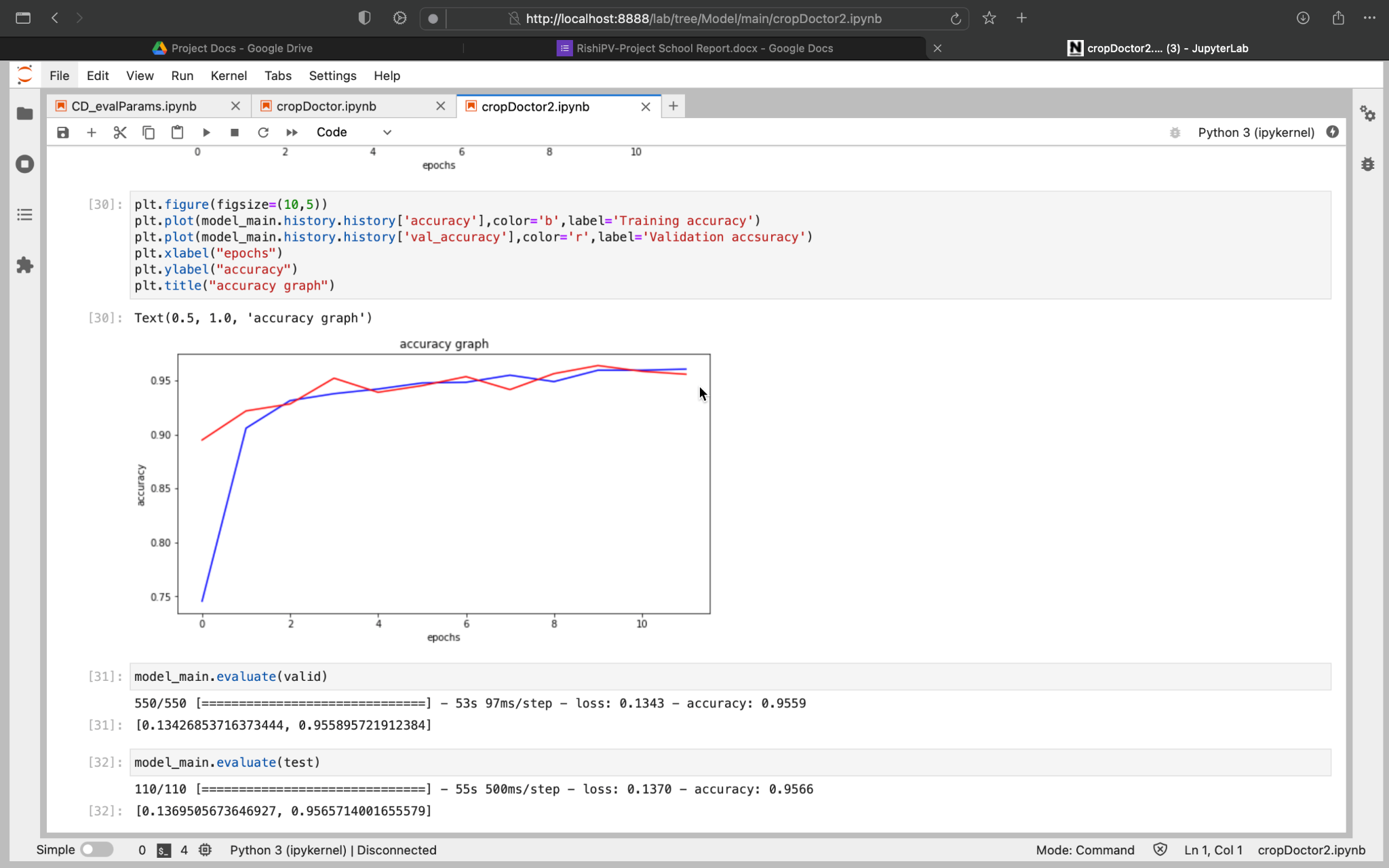
Steps for Epochs:- 400

Train Accuracy:- 96.06%

Test Accuracy:- 95.65%







SUMMARY

The web application for detecting diseases in plants is designed and can be used for detecting disease by just uploading an image onto the website.