PROJECT REPORT

**PROJECT TITLE : Fertilizers Recommendation System**

**For Disease Prediction**

**Team ID : PNT2022TMID21860**

# TEAM MEMBERS:

**1) Vignesh B - 142219106123**

1. **Tamilselvan M A - 311019106116**
2. **Udhayaraj A – 142219106118**

**4) Vasanthakumar V - 142219106122**

**Project Report Format**

* 1. **INTRODUCTION**
     1. Project Overview
     2. Purpose
  2. **LITERATURE SURVEY**
     1. Existing problem
     2. Problem Statement
  3. **IDEATION & PROPOSED SOLUTION**
     1. Empathy Map Canvas
     2. Ideation & Brainstorming
     3. Proposed Solution
     4. Problem Solution fit
  4. **REQUIREMENT ANALYSIS**
     1. Functional requirement
     2. Non-Functional requirements
  5. **PROJECT DESIGN**
     1. Data Flow Diagrams
     2. Solution & Technical Architecture
     3. User Stories
  6. **PROJECT PLANNING & SCHEDULING**
     1. Sprint Planning & Estimation
     2. Sprint Delivery Schedule
     3. Reports from JIRA
  7. **CODING & SOLUTION**
     1. Features
  8. **TESTING**
     1. User Acceptance Testing
  9. **RESULTS**
  10. **ADVANTAGES & DISADVANTAGES**
  11. **CONCLUSION**
  12. **FUTURE SCOPE**
  13. **APPENDIX**

Source Code

GitHub & Project Demo Link

# 1.INTRODUCTION

* 1. **Project Overview**

Agriculture is the most important sector in today’s life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, The number of diseases on plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques. An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

An Automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant.Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases changes in cultivation method and inadequate

Plant protection techniques and suggest all the precautions that can be taken for those diseases :

* Applying the CNN algorithm to the dataset
* How deep neural networks detect the disease
* You will be able to know how to find the accuracy of the model
* You will be able to build web applications using the Flask framework
  1. **Purpose**
     + The main aim of the project is to test the fruits and vegetables samples and identify the type of diseases affected.
     + After finding the disease ,the project recommends fertilizers for predicted diseases.

# LITERATURE SURVEY

* 1. **Existing problem**

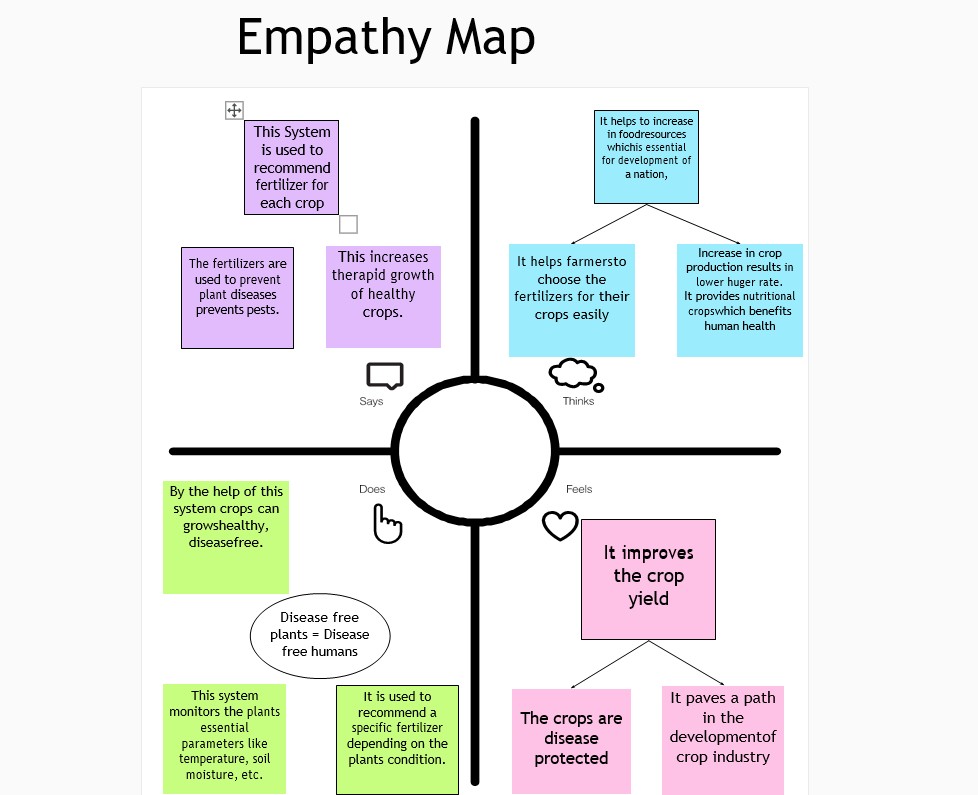
There are many new plant diseases that many farmers have no knowledge of . It takes a lot of time to identify plant disease after sending it to the laboratory. Without proper disease treatment ,many crops get spoiled.So,if there is an application that comes in handy to provide necessary information on plant diseases & best fertilizer for the crops,it would be of huge help for the farmers.

**2.2Problem Statement:**

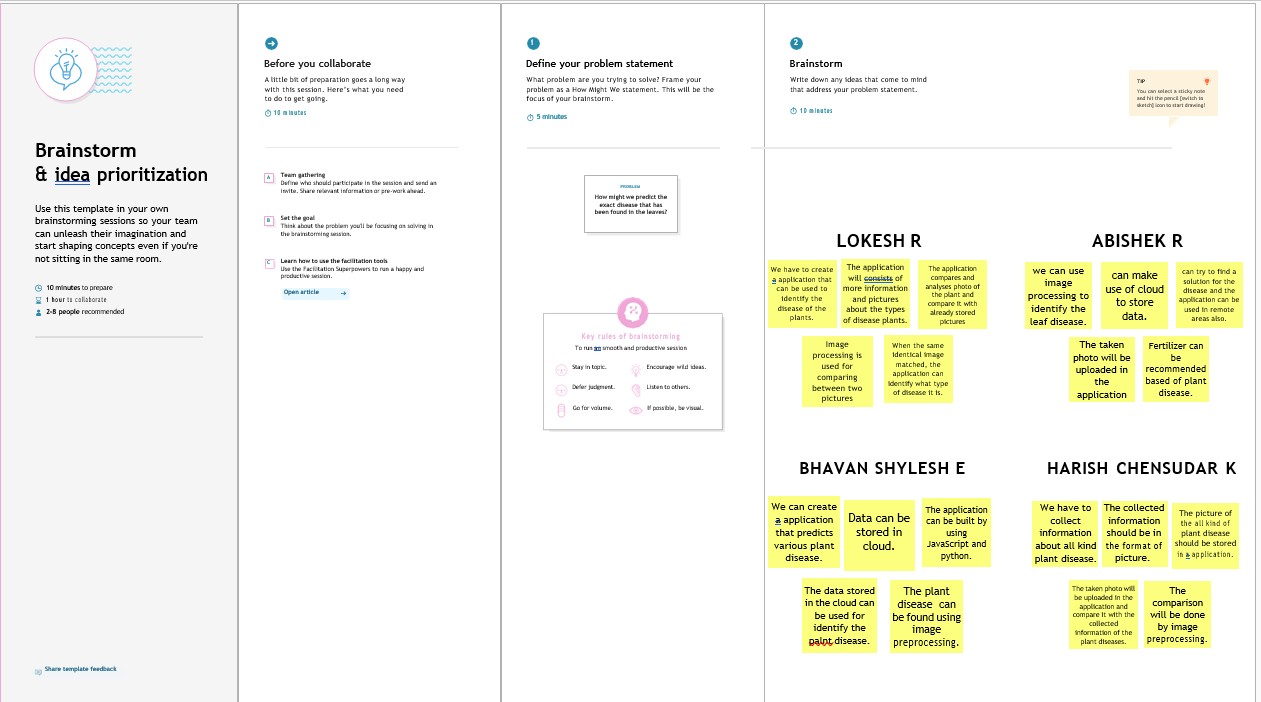
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Iam (Customer)** | **I’m trying to** | **But** | **Because** | **Which makes me feel** |
| PS-1 | Farmer | I want to find out whether my crops have any disease. | I don’t have any source/device to analyze the crops. | I don’t know which device to use, as I need to verify the crops for all kinds of diseases. | worried |
| PS-2 | Botanist | I am trying to research various plant diseases. | I need a device that can identify particular diseases. | I need to differentiate  each and every disease. | Troubled |

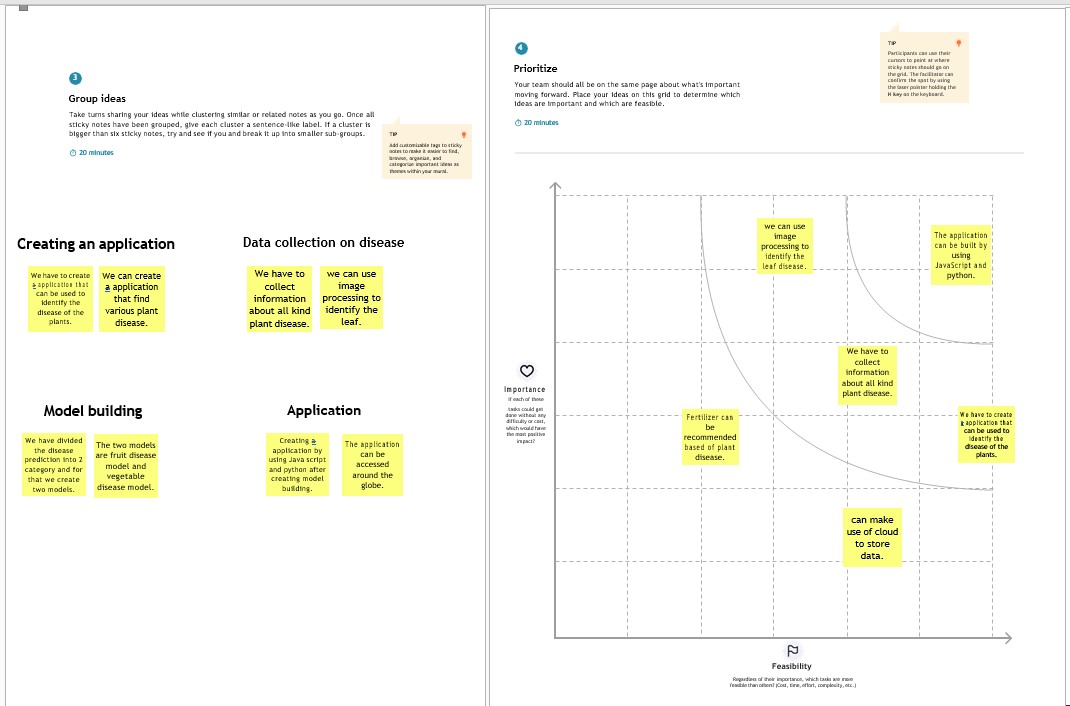
# IDEATION & PROPOSED SOLUTION

* 1. **Empathy Map Canvas**



* 1. **Ideation & Brainstorming**

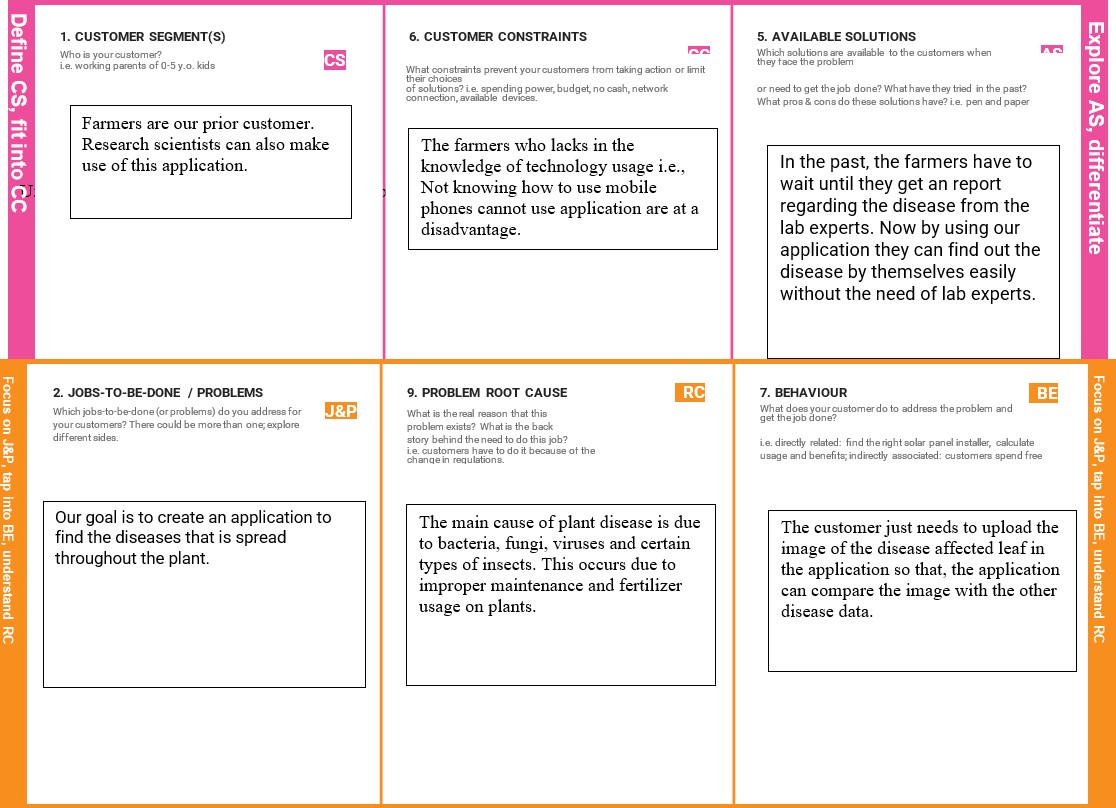


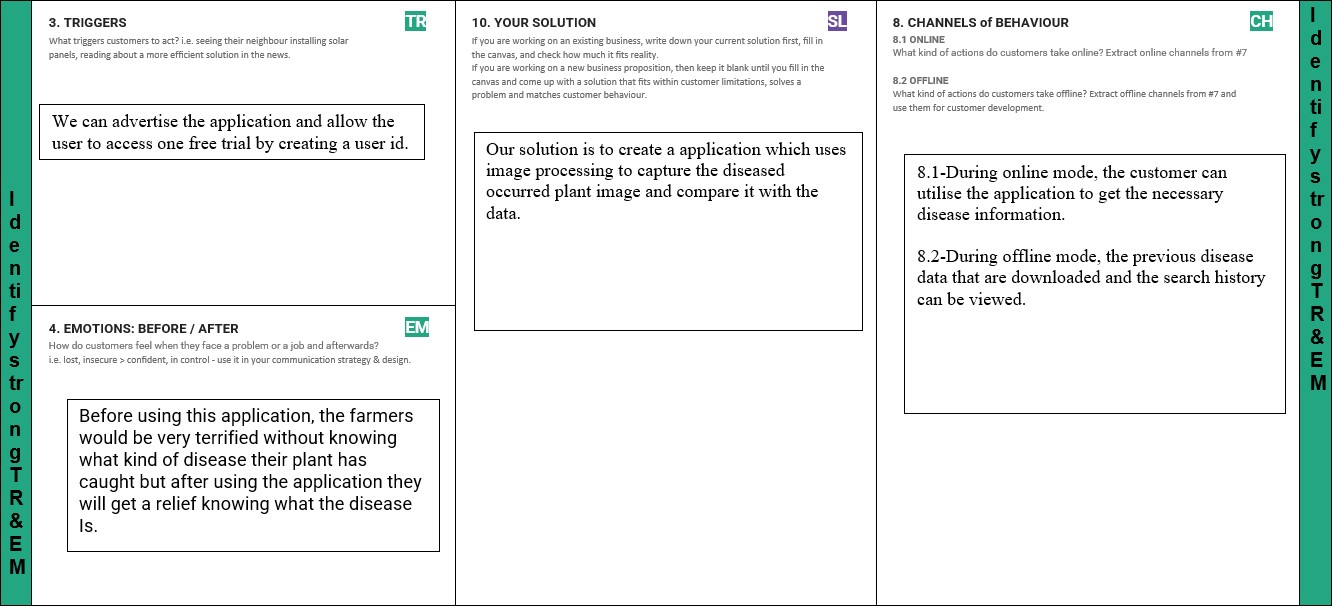


* 1. **Proposed Solution**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Parameter** | **Description** |
| **1** | Problem statement (problem to be solved ) | There are several kinds of plant diseases around the globe. It is not possible to detect all the diseases through a single handy device. |
| **2** | Idea/solution description | This project mainly focuses on creating a device that is used to detect all the plant diseases that  have been so far discovered and it can also update the current discovery of diseases. |
| **3** | Novelty / uniqueness | This device not only detects the disease but it also provides information regarding the fertilizer to be used. |
| **4** | Social impact/customer satisfaction | This device will surely be of useful for all the farmers throughout the globe. It also  creates a path in the plant research field. |
| **5** | Business model(revenue model) | It will be beneficial & profitable when it is manufactured. Many farmers will be benefited and there will be a huge profit on it. |
| **6** | Scalability of the solution | This project comes in handy due to its less weight and compact  size. |

* 1. **Problem Solution Fit**





# REQUIREMENT ANALYSIS

* 1. **Functional Requirement**

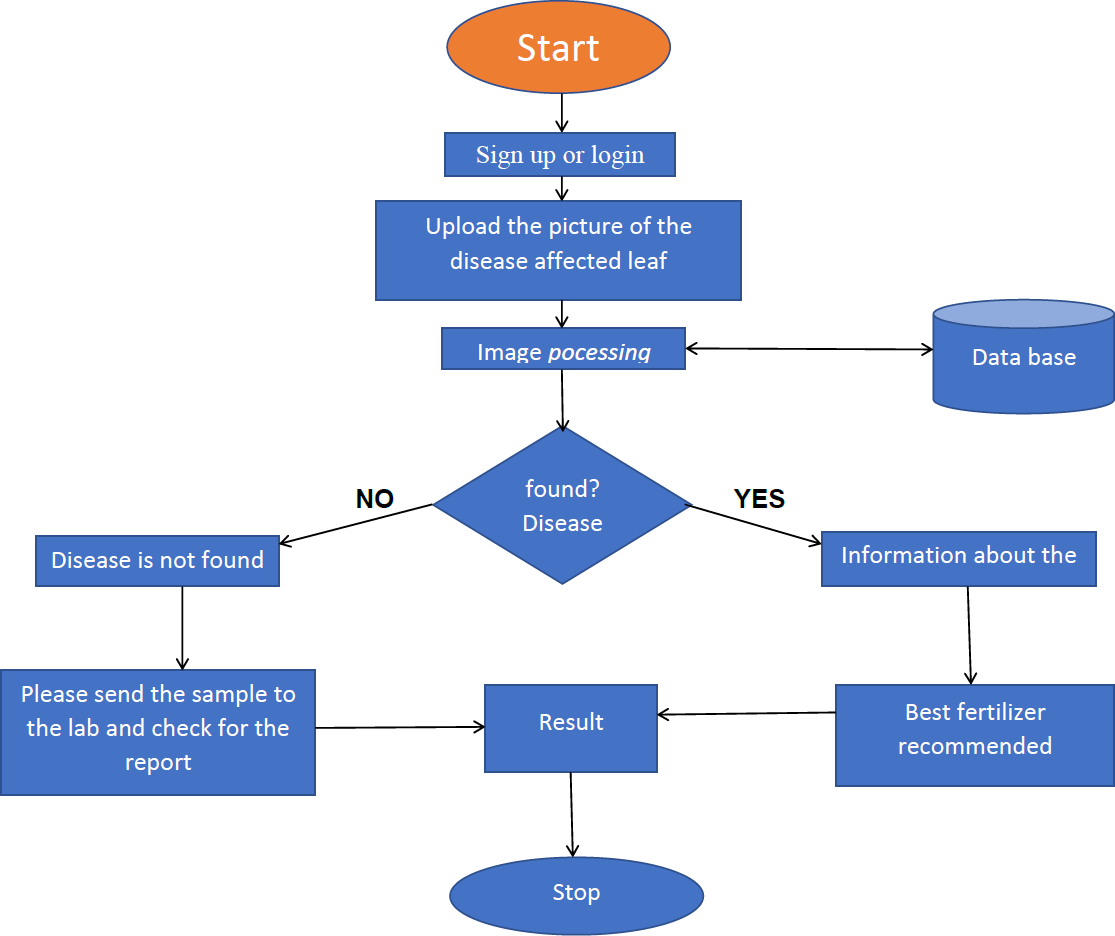
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form Registration through Gmail  Registration through LinkedIN |
| FR-2 | User Confirmation | Confirmation via Email Confirmation via OTP |
| FR-3 | User profile | Fill the profile page by user information after logging in and set a password. |
| FR-4 | Uploading image | Upload the image of the leaf for which the disease is to be found. |
| FR-5 | Image processing | Analysing the image and comparing it with the data base in order to find the disease. |
| FR-6 | Solution | After finding the disease, Best fertilizer is recommended to the crops. |

* 1. **Non-functional requirement**

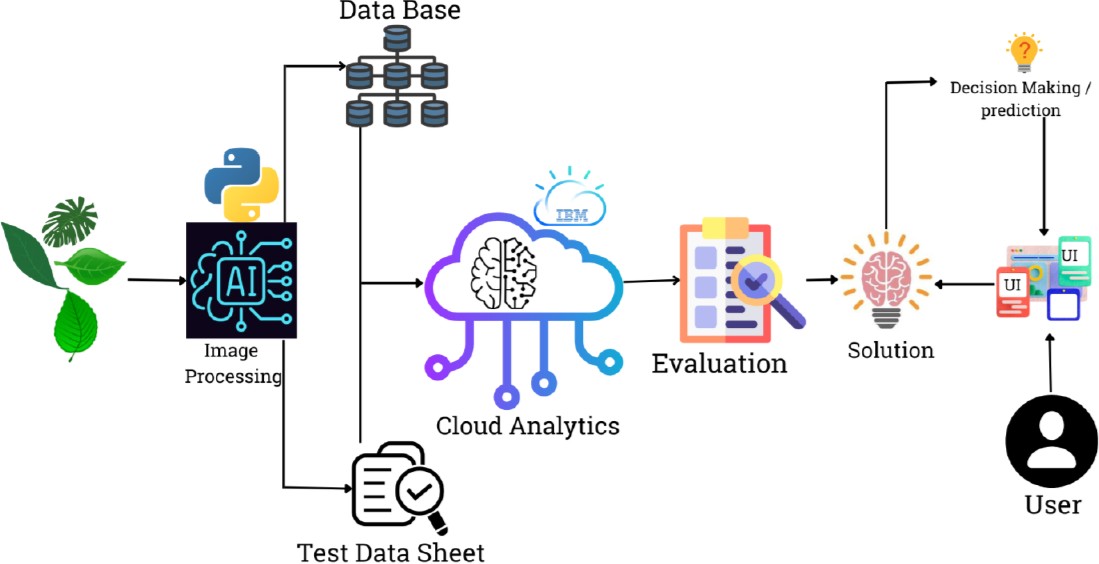
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | Beneficial in finding plant diseases. |
| NFR-2 | **Security** | It is protected with a username and password and it is done by using certain protocols like IPsec,  Application Transparent Transport Layer Security, etc. So it is highly secured. |
| NFR-3 | **Reliability** | The application is well secured and protects the user search information by providing a user id & password. |
| NFR-4 | **Performance** | It compares the image with the data that contains all the disease information and provides its best solution. |

# PROJECT DESIGN

**5.1Data Flow Diagrams**



**5.2.Solution & Technical Architecture**



**Components & Technologies**

**S.No Description Technology Component**

**1.**

**User Interface**

**2.**

**Application Logic-1**

**3.**

**Application Logic-2**

**4.**

**Application Logic-3**

**5.**

**Database**

**How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.**

**Logic for a process in the application**

**Logic for a process in the application**

**Logic for a process in the application**

**Data Type, Configurations etc.**

**HTML, CSS,**

**JavaScript**

**Java / Python**

**IBM Watson STT service**

**IBM Watson Assistant**

**MySQL, NoSQL,**

**etc.**

**6.**

**Cloud Database**

**7.**

**File Storage**

**Database Service on Cloud IBM DB2, IBM**

**Cloudant etc.**

**File storage requirements IBM Block Storage**

**or Other Storage Service or Local Filesystem**

**8.**

**9.**

**10.**

**External API-1**

**External API-2**

**Machine Learning Model**

**Purpose of External API used in the application**

**Purpose of External API used in the application**

**Purpose of Machine Learning Model**

**IBM Weather API, etc.**

**Aadhar API, etc.**

**Object Recognition Model, etc.**

**11.**

**Infrastructure**

**Application Deployment onLocal, Cloud Local System / Cloud LocalFoundry,**

**(Server / Cloud)**

**Server Configuration:**

**Cloud Server Configuration :**

**Kubernetes, etc.**

**5.3User Stories**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional**  **Requireme nt (Epic)** | **User Story Numbe r** | **User Story / Task** | **Acceptan ce criteria** | **Priority** | **Release** |
| Customer (Mobile user) | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can access my account / dashboard | High | Sprint-1 |
|  |  | USN-2 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmati on email & click confirm | High | Sprint-1 |
|  |  | USN-3 | As a user, I can register for the application through Facebook | I can register & access the dashboard with Facebook Login | Low | Sprint-2 |
|  |  | USN-4 | As a user, I can register for the application through Gmail | I can register through mail | Medium | Sprint-1 |
|  | Login | USN-5 | As a user, I can log into the application by entering email & password | I can register through login credentials | High | Sprint-1 |
|  | Interface | HABL-6 | As a user, the interface should be in user friendly manner. | I can access the contents easily | Medium | Sprint-2 |
| Customer (Web user) | Dashboard | HABL-7 | As a user, I can access specific information. | I can check the informatio n of diseases similar to | High | Sprint-1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | my crops. |  |  |
| Customer Care Executive | View image | HABL-0 1 | As a customer care, I can view the images of plant diseases. | I can easily check for the visual images. | High | Sprint-2 |
|  | View data | HABL-0 2 | As a customer care, I can view the data of  plant diseases. | I can view the disease  data. | High | Sprint-2 |
|  | Suggested fertilizer | HABL-0 3 | As a customer care, I can get the recommendation of best fertilizer of the crop. | I can get suggested fertilizer informatio n. | High | Sprint-2 |
| Administrator | Risk monitoring | RM-1 | An administrator is the one who is responsible for handling the application must check for any risk and handle it. | Admin should monitor and maintain the necessary records. | High | Sprint-1 |

1. **PROJECT PLANNING & SCHEDULING**
   1. **Sprint Planning & Estimation**

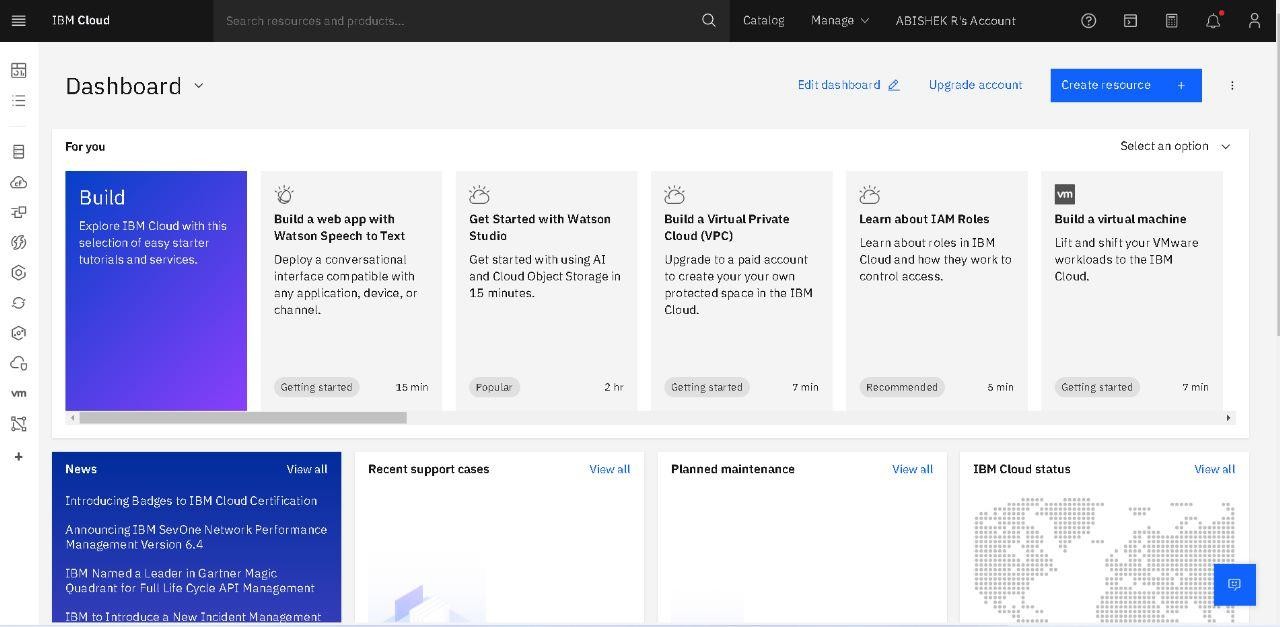
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| **Sprint-1** | **Registration** | **USN-1** | **As a user, I** | **2** | **High** | **HARISH** |
|  | **(Customer)** |  | **can register** |  |  | **CHENSUDAR K,** |
|  |  |  | **for the** |  |  | **ABISHEK R,** |
|  |  |  | **application by** |  |  | **LOKESH R,** |
|  |  |  | **entering my** |  |  | **BHAVAN** |
|  |  |  | **email,** |  |  | **SHYLESH E** |
|  |  |  | **password, and** |  |  |  |
|  |  |  | **confirming my** |  |  |  |
|  |  |  | **password.** |  |  |  |
| **Sprint-1** | **Login** | **USN-2** | **As a user, I** | **1** | **High** | **HARISH** |
|  |  |  | **will receive** |  |  | **CHENSUDAR K,** |
|  |  |  | **confirmation** |  |  | **ABISHEK R,** |
|  |  |  | **email once I** |  |  | **LOKESH R,** |
|  |  |  | **have** |  |  | **BHAVAN** |
|  |  |  | **registered for** |  |  | **SHYLESH E** |
|  |  |  | **the application** |  |  |  |
| **Sprint-1** | **Dashboard** | **USN-4** | **I am a new** | **2** | **High** | **HARISH** |
|  |  |  | **user, ready to** |  |  | **CHENSUDAR** |
|  |  |  | **explore the** |  |  | **K,** |
|  |  |  | **site via** |  |  | **ABISHEK R,** |
|  |  |  | **dashboard** |  |  | **LOKESH R,** |
|  |  |  |  |  |  | **BHAVAN** |
|  |  |  |  |  |  | **SHYLESH E** |
| **Sprint-1** | **Registration** | **USN-1** | **I am in seek of** | **2** | **High** | **HARISH** |
|  | **(Administrator)** |  | **my profile details and my**  **exploring** |  |  | **CHENSUDAR K, ABISHEK R, LOKESH R,** |
|  |  |  | **stuffs** |  |  | **BHAVAN** |
|  |  |  |  |  |  | **SHYLESH E** |
| **Sprint-2** | **Login** | **USN-3** | **As an existing** | **1** | **Low** | **HARISH** |
|  |  |  | **user, I can** |  |  | **CHENSUDAR K,** |
|  |  |  | **access the** |  |  | **ABISHEK R,** |
|  |  |  | **website with** |  |  | **LOKESH R,** |
|  |  |  | **login** |  |  | **BHAVAN** |
|  |  |  | **credentials** |  |  | **SHYLESH E** |
|  |  |  | **that are** |  |  |  |
|  |  |  | **specific to** |  |  |  |

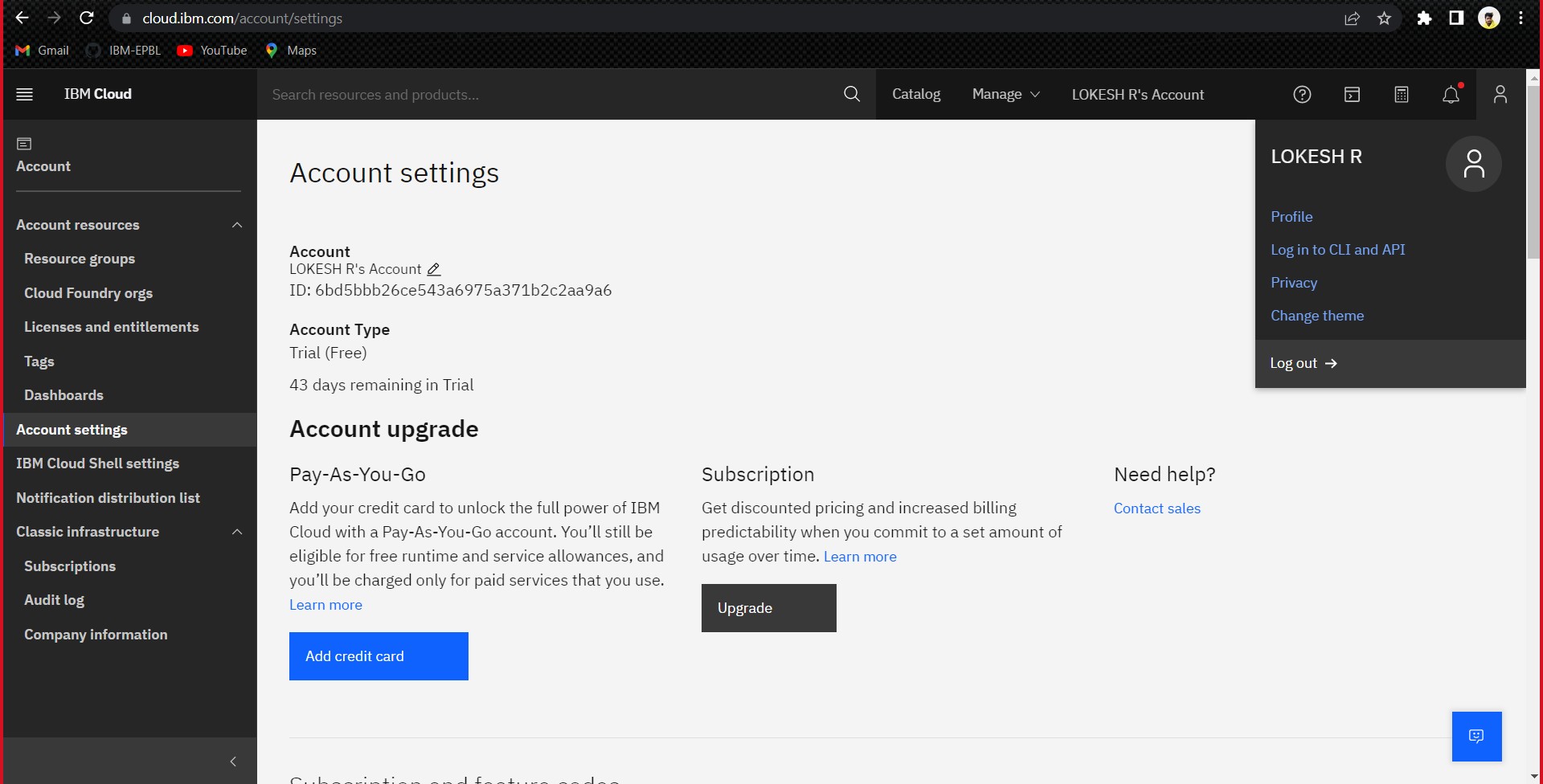
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **myself.** |  |  |  |
| **Sprint-2** | **Dashboard** | **USN-5** | **I am an existing** | **1** | **High** | **HARISH** |
|  |  |  | **user, pick from** |  |  | **CHENSUDAR K,** |
|  |  |  | **where I left** |  |  | **ABISHEK R,** |
|  |  |  |  |  |  | **LOKESH R,** |
|  |  |  |  |  |  | **BHAVAN** |
|  |  |  |  |  |  | **SHYLESH E** |
| **RSprint-3** | **Image Uploading** | **USN-1** | **I am in need of** | **1** | **High** | **HARISH** |
|  |  |  | **suggestions of** |  |  | **CHENSUDAR K,** |
|  |  |  | **fertilizer for the** |  |  | **ABISHEK R,** |
|  |  |  | **crops in my** |  |  | **LOKESH R,** |
|  |  |  | **field. So I am** |  |  | **BHAVAN** |
|  |  |  | **uploading the** |  |  | **SHYLESH E** |
|  |  |  | **images for the** |  |  |  |
|  |  |  | **same.** |  |  |  |
| **Sprint-3** | **Image Uploading** | **USN-2** | **I am in need of suggestions of my land(soil) but I don’t have a clear picture of it. So I am specifying through text format.** | **2** | **High** | **HARISH CHENSUDAR K, ABISHEK R, LOKESH R,**  **BHAVAN SHYLESH E** |
| **Sprint-4** | **View of** | **USN-1** | **I am just an** | **2** | **Low** | **HARISH** |
|  | **recommends** |  | **explorer, view** |  |  | **CHENSUDAR** |
|  |  |  | **of recommends** |  |  | **K,** |
|  |  |  | **are adequate** |  |  | **ABISHEK R,** |
|  |  |  |  |  |  | **LOKESH R,** |
|  |  |  |  |  |  | **BHAVAN** |
|  |  |  |  |  |  | **SHYLESH E** |
| **Sprint-4** | **Feedback Providing** | **USN-1** | **I finished of my journey on the website, time to provide**  **feedback.** | **1** | **Low** | **HARISH CHENSUDAR K, ABISHEK R, LOKESH R,** |
|  |  |  |  |  |  | **BHAVAN** |
|  |  |  |  |  |  | **SHYLESH E** |
| **Sprint -4** | **Feedback** | **USN-2** | **I need to see all** | **2** | **Low** | **HARISH** |
|  | **Collection** |  | **other peer** |  |  | **CHENSUDAR** |
|  |  |  | **members’** |  |  | **K,** |
|  |  |  | **review about** |  |  | **ABISHEK R,** |
|  |  |  | **this site.** |  |  | **LOKESH R,** |
|  |  |  |  |  |  | **BHAVAN** |
|  |  |  |  |  |  | **SHYLESH E** |

* 1. **Sprint Delivery Schedule**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Duration** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint Release Date (Actual)** |
| **Sprint-1** | **20** | **6 Days** | **24 Oct**  **2022** | **30 Oct**  **2022** | **20** | **30 Oct**  **2022** |
| **Sprint-2** | **20** | **6 Days** | **31 Oct**  **2022** | **05 Nov**  **2022** | **20** | **05 Nov**  **2022** |
| **Sprint-3** | **20** | **6 Days** | **07 Nov**  **2022** | **12 Nov**  **2022** | **20** | **12 Nov**  **2022** |
| **Sprint-4** | **20** | **6 Days** | **14 Nov**  **2022** | **19 Nov**  **2022** | **20** | **19 Nov**  **2022** |

* 1. **IBM CLOUD**





# CODING & SOLUTION

## PYTHON CODE

import io import os

import pickle

import config import numpy as np import pandas as pd import requests import torch

from flask import Flask, Markup, render\_template, request from PIL import Image

from torchvision import transforms from utils.disease import disease\_dic

from utils.fertilizer import fertilizer\_dic from utils.model import ResNet9

disease\_classes = ['Apple Apple\_scab',

'Apple Black\_rot',

'Apple Cedar\_apple\_rust', 'Apple healthy', 'Blueberry healthy',

'Cherry\_(including\_sour) Powdery\_mildew', 'Cherry\_(including\_sour) healthy',

'Corn\_(maize) Cercospora\_leaf\_spot Gray\_leaf\_spot', 'Corn\_(maize) Common\_rust',

'Corn\_(maize) Northern\_Leaf\_Blight', 'Corn\_(maize) healthy',

'Grape Black\_rot',

'Grape Esca(Black\_Measles)',

'Grape Leaf\_blight(Isariopsis\_Leaf\_Spot)', 'Grape healthy',

'Orange Haunglongbing(Citrus\_greening)', 'Peach Bacterial\_spot',

'Peach healthy', 'Pepper,bell Bacterial\_spot', 'Pepper,bell healthy', 'Potato Early\_blight', 'Potato Late\_blight', 'Potato healthy',

'Raspberry healthy', 'Soybean healthy', 'Squash Powdery\_mildew', 'Strawberry Leaf\_scorch', 'Strawberry healthy', 'Tomato Bacterial\_spot', 'Tomato Early\_blight', 'Tomato Late\_blight', 'Tomato Leaf\_Mold',

'Tomato Septoria\_leaf\_spot',

'Tomato Spider\_mites Two-spotted\_spider\_mite', 'Tomato Target\_Spot',

'Tomato Tomato\_Yellow\_Leaf\_Curl\_Virus', 'Tomato Tomato\_mosaic\_virus',

'Tomato healthy'] disease\_model\_path = 'models/plant\_disease\_model.pth' disease\_model = ResNet9(3, len(disease\_classes)) disease\_model.load\_state\_dict(torch.load(

disease\_model\_path, map\_location=torch.device('cpu'))) disease\_model.eval()

crop\_recommendation\_model\_path = 'models/RandomForest.pkl'

crop\_recommendation\_model = pickle.load( open(crop\_recommendation\_model\_path, 'rb'))

def weather\_fetch(city\_name): api\_key = config.weather\_api\_key

base\_url = ["http://api.openweathermap.org/data/2.5/weather?"](http://api.openweathermap.org/data/2.5/weather)

complete\_url = base\_url + "appid=" + api\_key + "&q=" + city\_name response = requests.get(complete\_url)

x = response.json()

if x["cod"] != "404": y = x["main"]

temperature = round((y["temp"] - 273.15), 2) return temperature

else:

return None

def predict\_image(img, model=disease\_model): transform = transforms.Compose([

transforms.Resize(256), transforms.ToTensor(),

])

image = Image.open(io.BytesIO(img)) img\_t = transform(image)

img\_u = torch.unsqueeze(img\_t, 0)

# Get predictions from model yb = model(img\_u)

# Pick index with highest probability

\_, preds = torch.max(yb, dim=1)

prediction = disease\_classes[preds[0].item()] # Retrieve the class label

return prediction app=Flask(\_name\_)

@ app.route('/crop-predict', methods=['POST']) def crop\_prediction():

title = 'Four leaf clover - Plant Disease Prediction' if request.method == 'POST':

N = int(request.form['nitrogen'])

P = int(request.form['phosphorous']) K = int(request.form['pottasium']) ph = float(request.form['ph'])

rainfall = float(request.form['rainfall'])

# state = request.form.get("stt") city = request.form.get("city")

if weather\_fetch(city) != None:

temperature, humidity = weather\_fetch(city)

data = np.array([[N, P, K, temperature, humidity, ph, rainfall]]) my\_prediction = crop\_recommendation\_model.predict(data) final\_prediction = my\_prediction[0]

return render\_template('result.html', prediction=final\_prediction, title=title)

else:

return render\_template('try\_again.html', title=title) @ app.route('/fertilizer-predict', methods=['POST'])

def fert\_recommend():

title = 'Four Leaf Clover - Plant Disease Prediction ' crop\_name = str(request.form['cropname'])

N = int(request.form['nitrogen'])

P = int(request.form['phosphorous']) K = int(request.form['pottasium'])

# ph = float(request.form['ph'])

df = pd.read\_csv('Data/fertilizer.csv')

nr = df[df['Crop'] == crop\_name]['N'].iloc[0]

pr = df[df['Crop'] == crop\_name]['P'].iloc[0]

kr = df[df['Crop'] == crop\_name]['K'].iloc[0] n = nr - N

p = pr - P k = kr - K

temp = {abs(n): "N", abs(p): "P", abs(k): "K"} max\_value = temp[max(temp.keys())]

if max\_value == "N": if n < 0:

key = 'NHigh' else:

key = "Nlow" elif max\_value == "P":

if p < 0:

key = 'PHigh' else:

key = "Plow"

else:

if k < 0:

key = 'KHigh' else:

key = "Klow"

response = Markup(str(fertilizer\_dic[key]))

return render\_template('result.html', recommendation=response, title=title) @app.route('/disease-predict', methods=['GET', 'POST'])

def upload():

if request.method=='POST': f=request.files['image'] basepath=os.path.dirname(\_file\_)

filepath=os.path.join(basepath,'uploads',f.filename) f.save(filepath)

print('File Save') img=image.load\_img(filepath,target\_size=(128,128)) x=image.img\_to\_array(img)

print('Image to gray') x=np.expand\_dims(x,axis=0) plant=request.form['plant'] if(plant=='vegetable'):

model=load\_model("vegitable.h5") y=np.argmax(model.predict(x),axis=1) df=pd.read\_excel('precautions\_veg.xlsx')

if(plant=='fruit'): model=load\_model('fruit.h5') y=np.argmax(model.predict(x),axis=1) df=pd.read\_excel('precautions\_fruits.xlsx')

return df.iloc[y[0]]['caution'] if \_name\_=='\_main\_':

temp.run(debug=False)

**HTML CODE**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="utf-8">**

**<meta name="viewport" content="with=device-width, initial-scale=1.0">**

**<title>Four Leaf Clover - Disease Prediction </title>**

**<link rel="stylesheet" href="style.css">**

**<link rel="preconnect" href="https://fonts.googleapis.com">**

**<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin="">**

**<link**

**href=**[**"https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,400;1,200;1,300;1,40**](https://fonts.googleapis.com/css2?family=Poppins%3Aital%2Cwght%400%2C400%3B1%2C200%3B1%2C300%3B1%2C40) **0;1,600&amp;display=swap" rel="stylesheet">**

**<link rel="stylesheet"**

**href=**[**"https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.2.0/css/fontawesome.**](https://cdn.jsdelivr.net/npm/%40fortawesome/fontawesome-free%406.2.0/css/fontawesome) **min.css">**

**<link href=**[**"https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"**](https://cdn.jsdelivr.net/npm/bootstrap%405.2.2/dist/css/bootstrap.min.css) **rel="stylesheet"**

**integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">**

**</head>**

**<body>**

**<section class="header">**

**<nav>**

**<a href="index.html"><img src="image/logo.jpg" ></a>**

**<div class="nav-links" id="navLinks">**

**<i class="fa fa-times" onclick="hidemenu()"></i>**

**<ul>**

**<li><a href="">HOME</a></li>**

**<li><a href="">ABOUT</a></li>**

**<li><a href="">SERVICES</a></li>**

**<li><a href="">FEATURES</a></li>**

**<li><a href="">CONTACT </a></li>**

**</ul>**

**</div>**

**<i class="fa fa-bars" onclick="showmenu()"></i>**

**</nav>**

**<div class="text-box">**

**<h1>Welcome to Four Leaf Clover! </h1>**

**<h2>Creating a Disease Free Plant Environment</h2>**

**<p> Find out more about plant diseases from our page ,<br> let us create a heathly environment for your crops.</p>**

**<a href="login.html" class="login-btn">Log In/ Sign Up</a>**

**</div>**

**</section>**

**<section class="about">**

**<h1>Four Leaf Clover - Disease Prediction </h1>**

**<div class="row">**

**<div class="about-col">**

**<h3>Why use Four Leaf Clover ?</h3>**

**<p> Our page mainly focuses on providing accurate information on plant**

**order to treat**

**diseases and provide the user with a fertilizer to be used in**

**the plant Disease .**

**</p>**

**</div>**

**<div class="about-col">**

**<h3>Why use Four Leaf Clover ?</h3>**

**<p> Our page mainly focuses on providing accurate**

**information on plant order to treat**

**diseases and provide the user with a fertilizer to be used in the plant Disease .**

**</p>**

**</div>**

**<div class="about-col">**

**<h3>Why use Four Leaf Clover ?</h3>**

**<p> Our page mainly focuses on providing accurate information on plant**

**order to treat**

**diseases and provide the user with a fertilizer to be used in**

**the plant Disease .**

**</p>**

**</div>**

**</div>**

**</section>**

**<section class="service">**

**<h2>SERVICE </h2>**

**<div class="row">**

**<div class="service-col">**

**<p>**

**We offer the best solution for most of the plant diseases.**

**Nearly thousand of customer**

**have been satisfied by our page. Our main goal is to help the farmers with their problems and provide**

**them with the best fertilier solution.**

**<br>**

**<br>**

**Not joined yet? Hurry up and be a part of our community. Let us create a disease free plant Environment.**

**</p>**

**</div>**

**<div class="service-col">**

**<img src="image/we.png" width="200px">**

**</div>**

**</div>**

**</section>**

**<section class="features">**

**<h2> FEATURES </h2>**

**<div class="row">**

**<div class="feature-col">**

**</div>**

**</div>**

**</section>**

**<!--------------->**

**<script>**

**var navLinks = document.getElementById("navLinks"); function showmenu(){**

**navLinks.style.right = "0";**

**}**

**function hidemenu(){ navLinks.style.right = "-200px";**

**}**

**</script>**

**</body>**

**</html>**

# 8.TESTING

* 1. **User Acceptance Testing**

## Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Fertilizer Recommendation system for plant disease prediction] project at the time of the release to User Acceptance Testing (UAT).

## Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Total Cases** | **Not Tested** | **Fail** | **Pass** |
| Print Engine | 7 | 0 | 0 | 7 |
| Client Application | 38 | 0 | 2 | 36 |
| Security | 4 | 0 | 0 | 4 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | **Severity 1** | **Severity 2** | **Severity 3** | **Severity 4** | **Subtotal** |
| By Design | 8 | 1 | 2 | 2 | 14 |
| Duplicate | 1 | 1 | 0 | 0 | 2 |
| External | 2 | 2 | 1 | 1 | 6 |
| Fixed | 14 | 2 | 5 | 14 | 35 |
| Not  Reproduced | 0 | 0 | 0 | 0 | 0 |
| Skipped | 0 | 0 | 1 | 0 | 1 |
| Won't Fix | 0 | 5 | 2 | 1 | 8 |
| Totals | 25 | 11 | 11 | 18 | 65 |

## Test Case Analysis

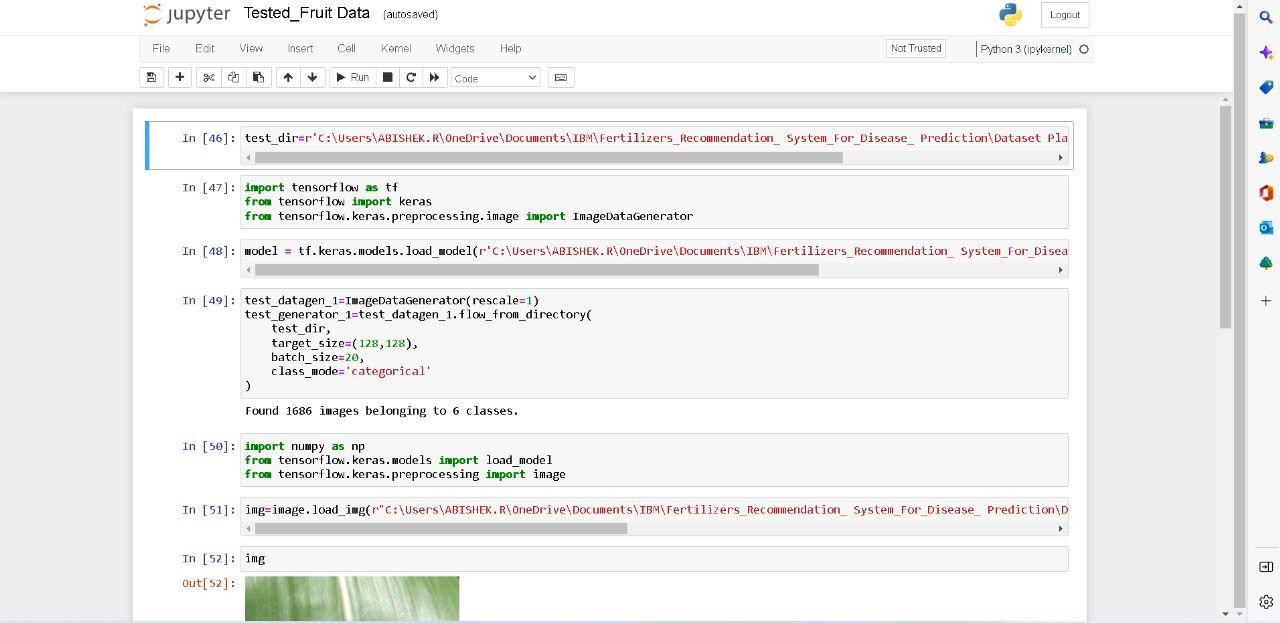
This report shows the number of test cases that have passed, failed, and untested

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Outsource Shipping | 0 | 0 | 0 | 3 |
| Exception Reporting | 12 | 0 | 2 | 10 |
| Final Report Output | 6 | 0 | 0 | 6 |
| Version Control | 3 | 0 | 1 | 2 |

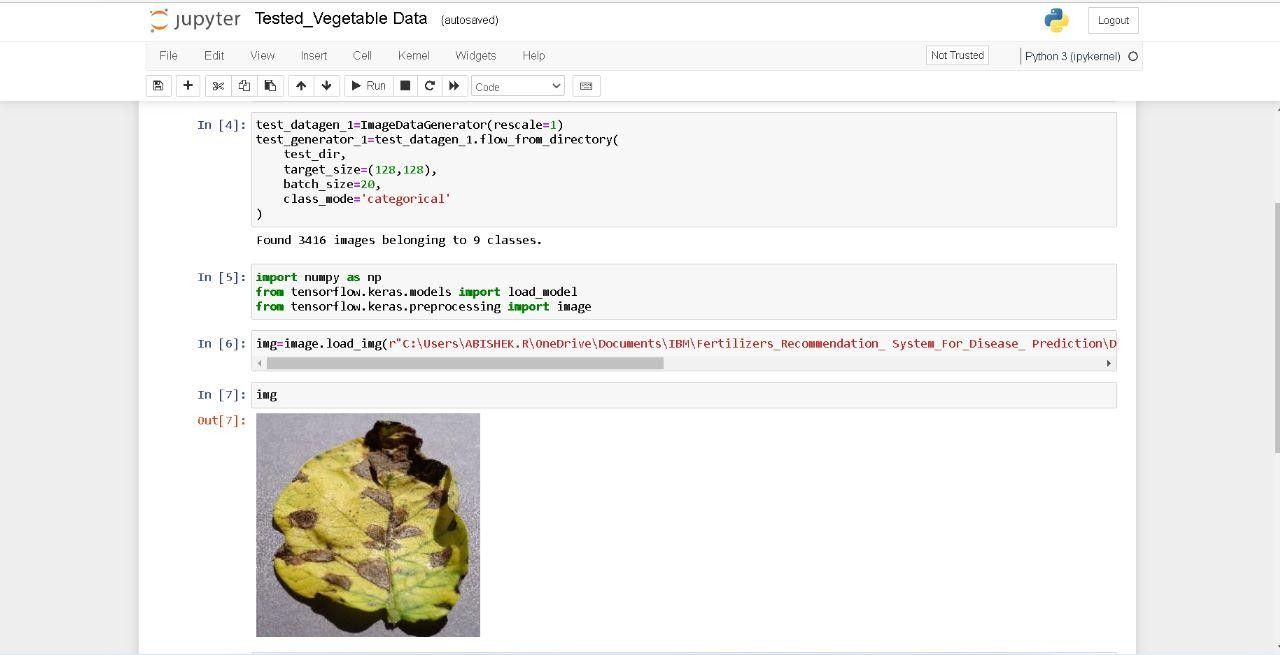
# RESULTS

**Final findings (output) of the project given below in the form of screenshot:**

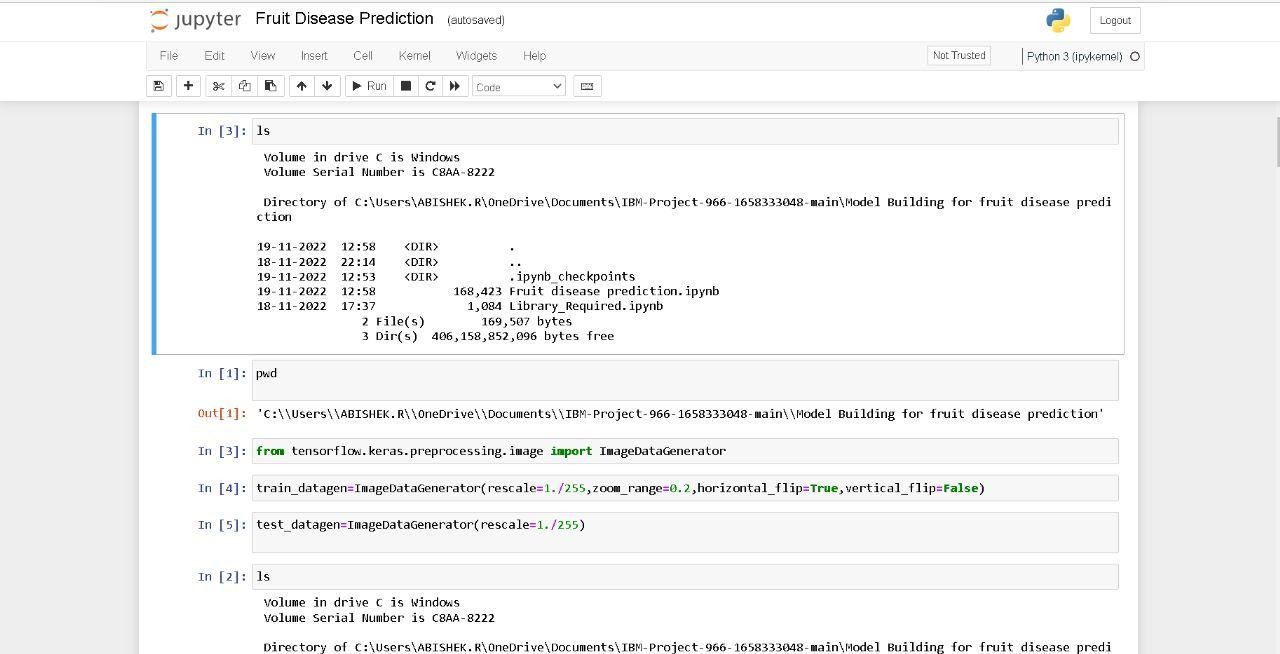
**Tested Fruit Data**



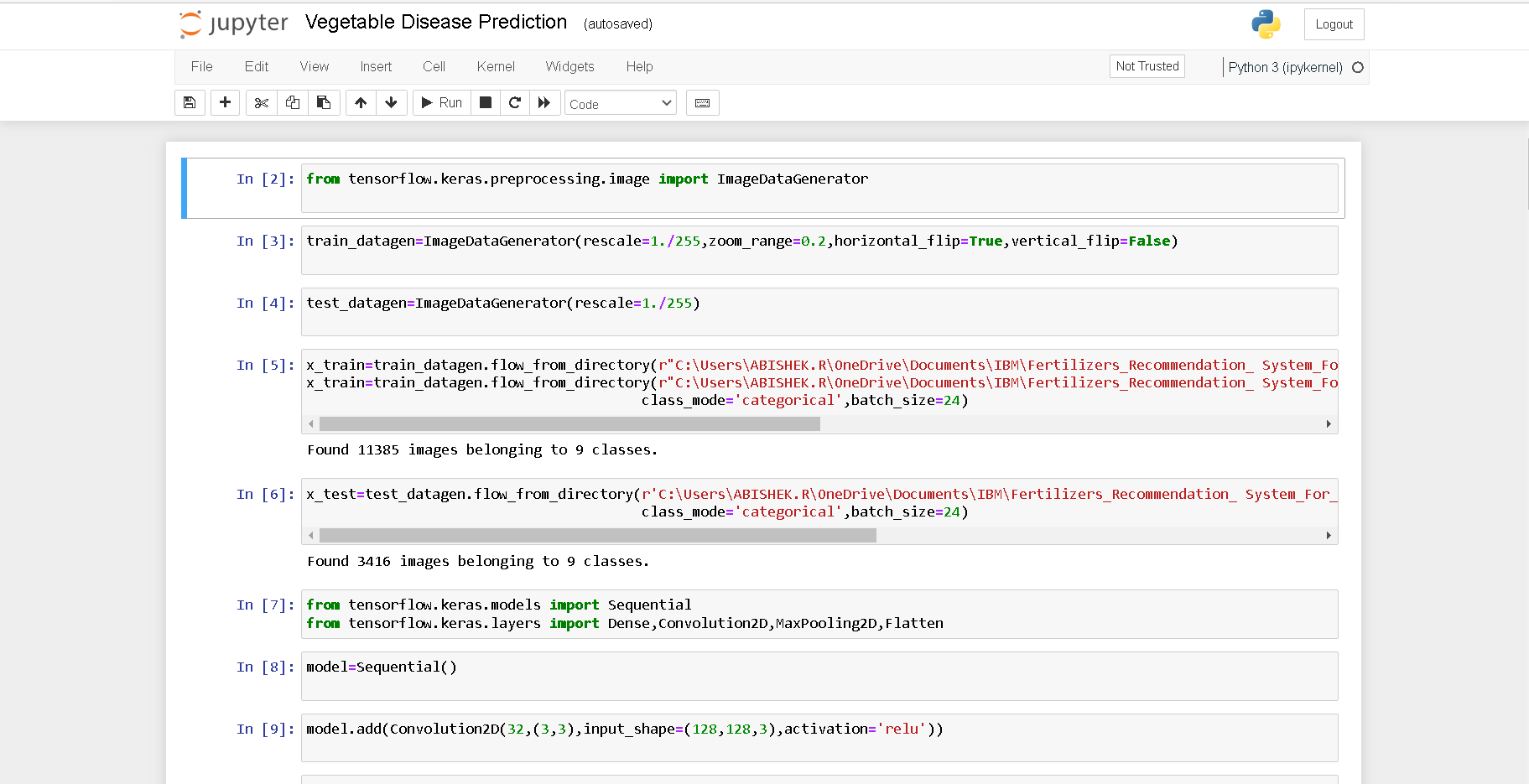
**Tested Vegetable Data**

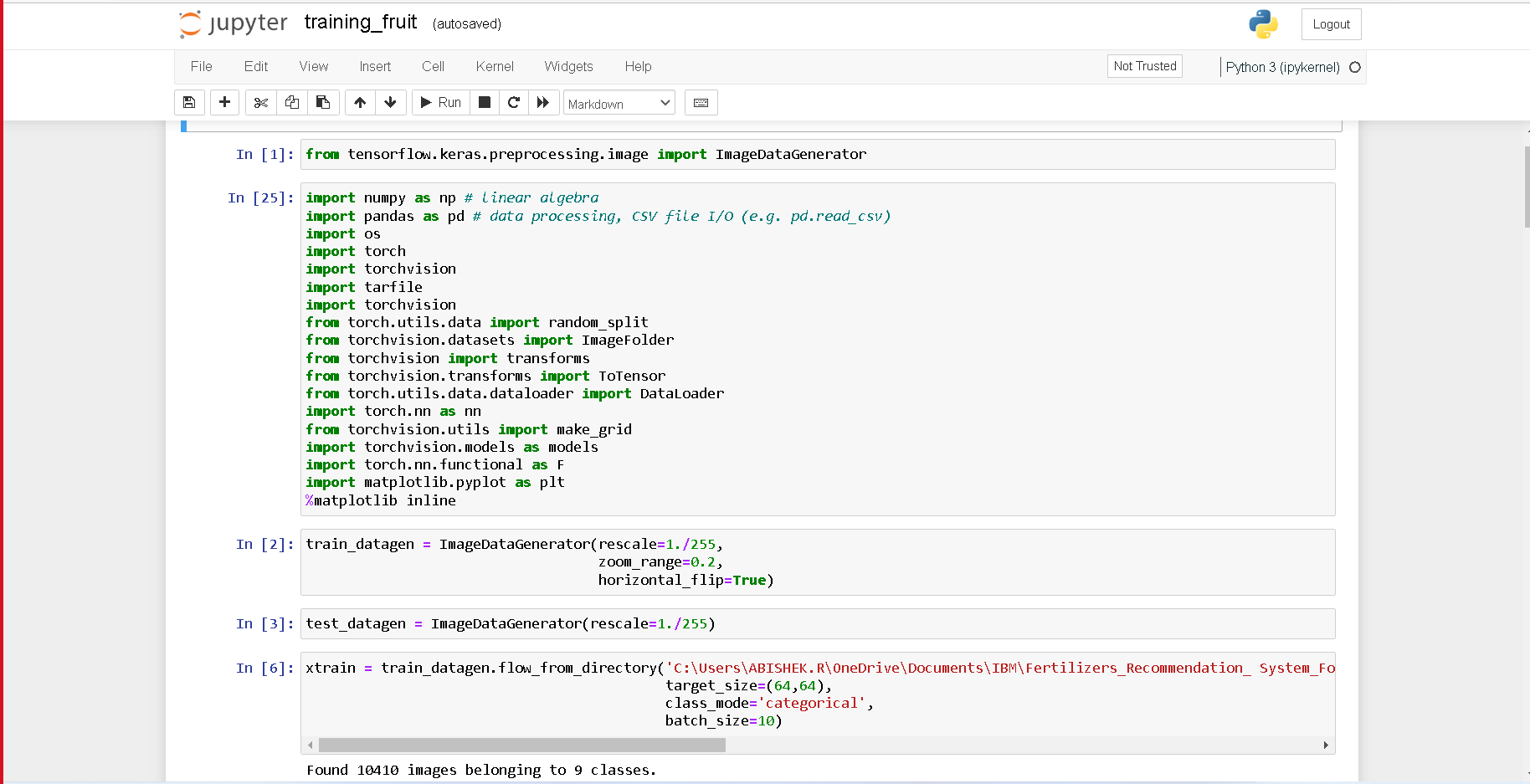


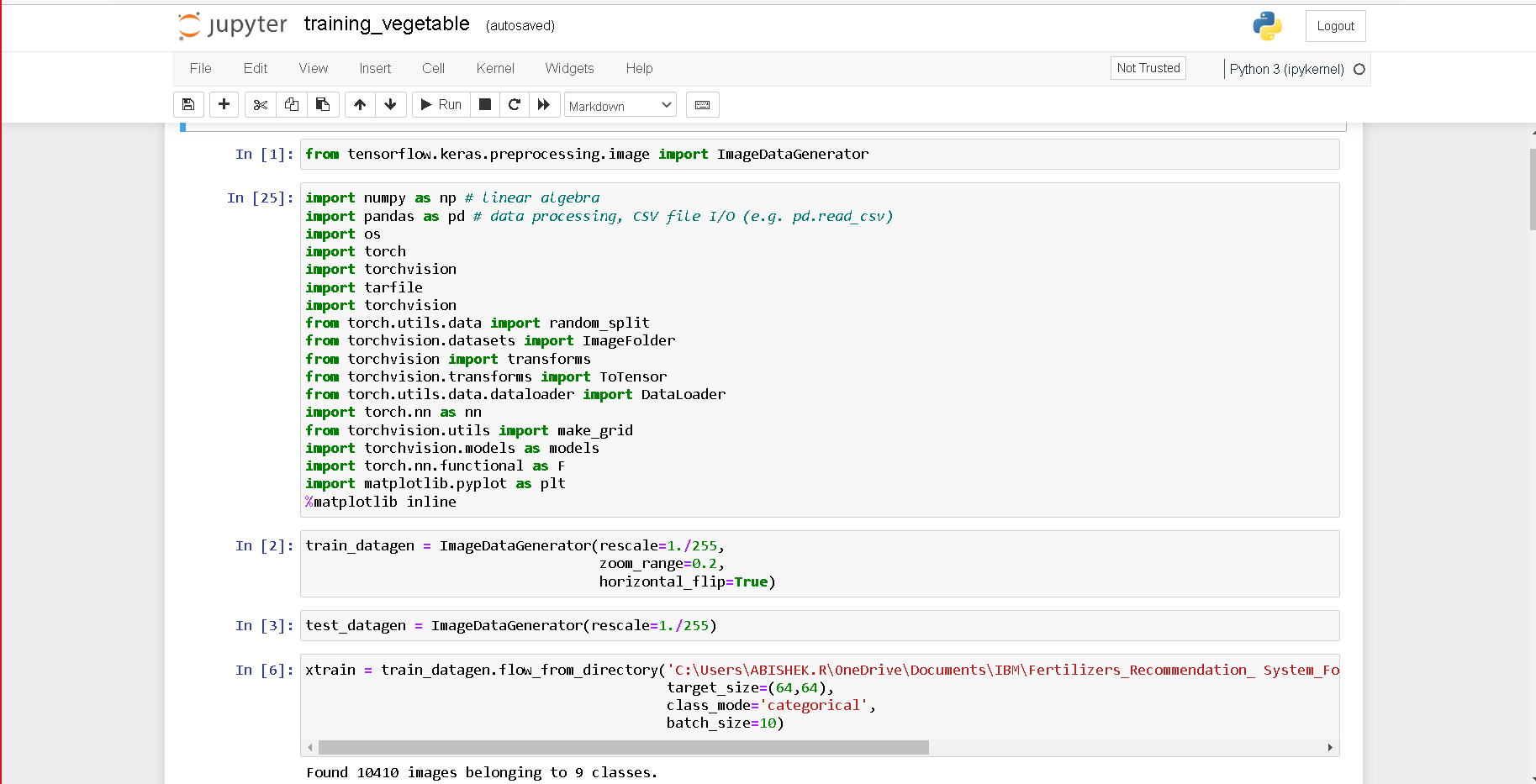
**Fruit Disease Prediction**



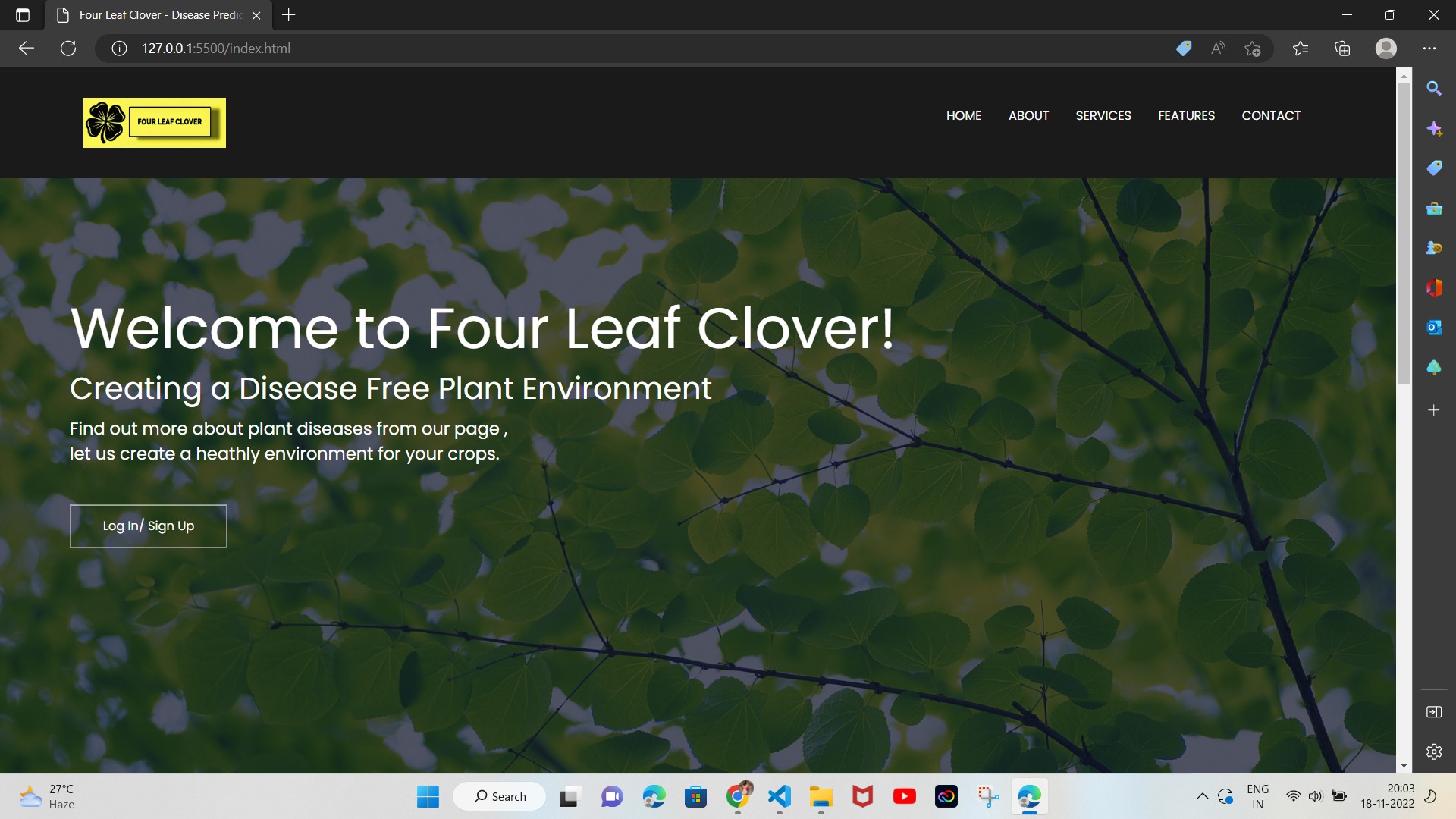
Vegetable Disease Prediction

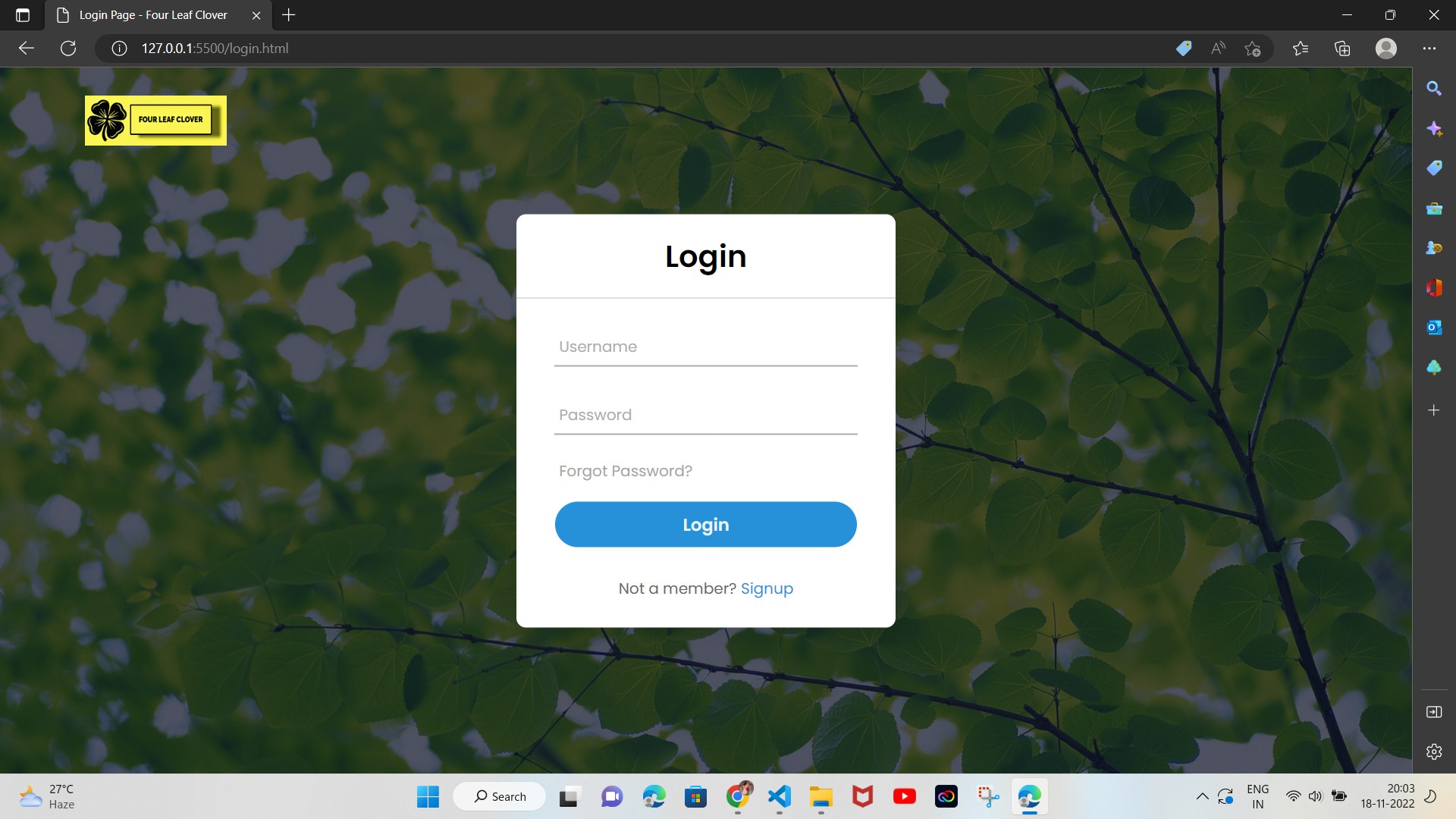


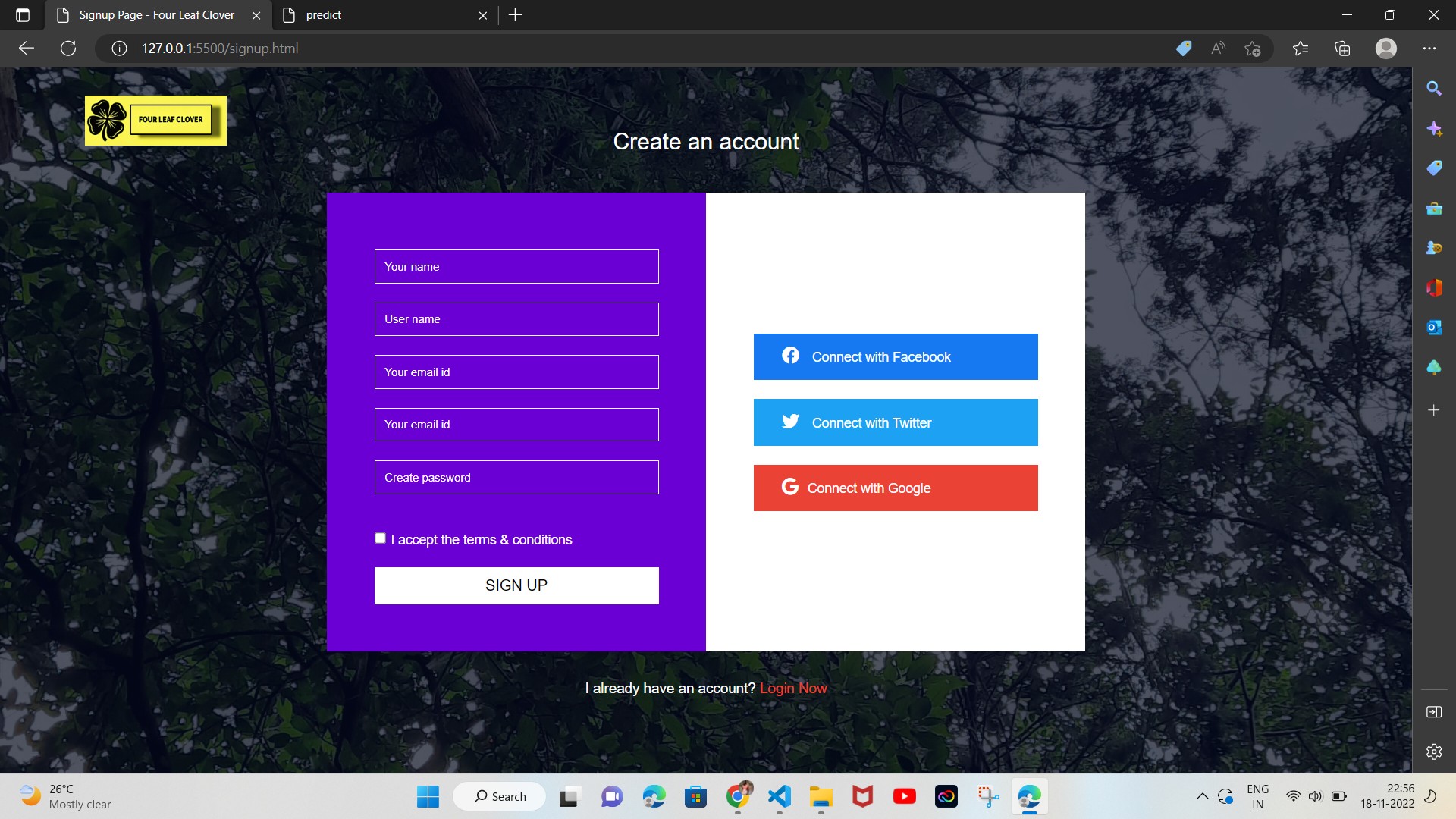
**Training Fruit**

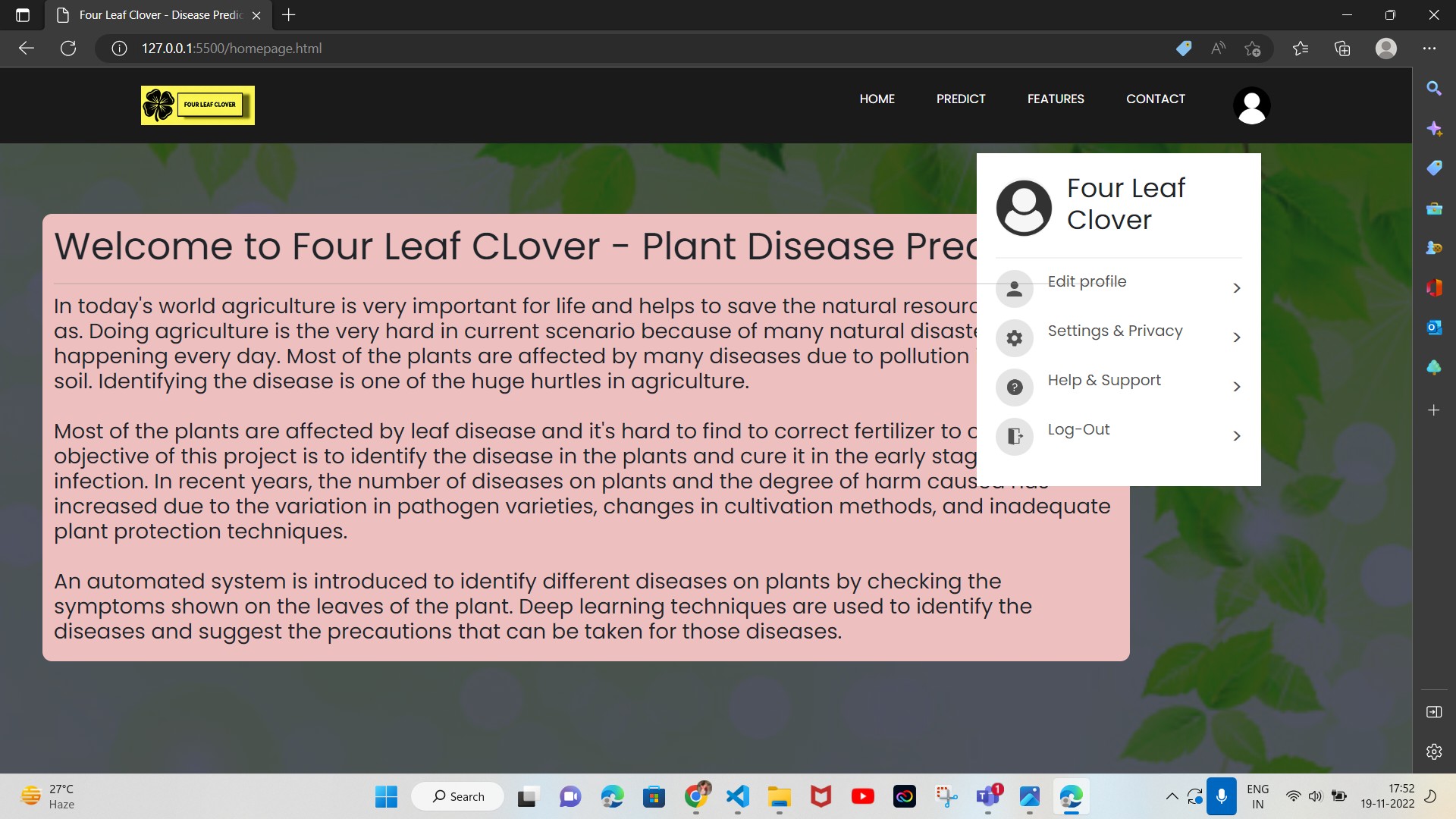
**Training Vegetable**

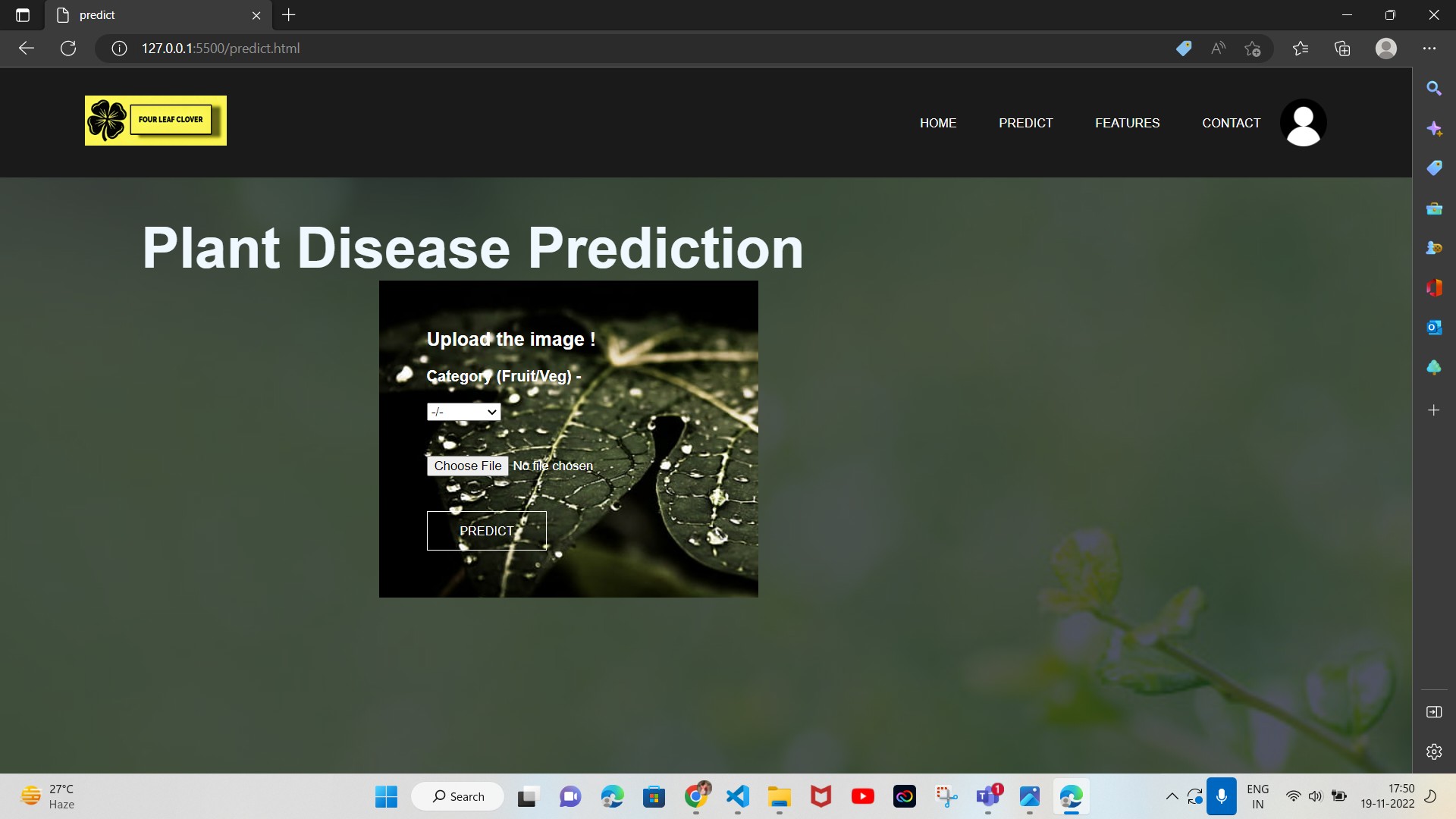
# OUTPUT:

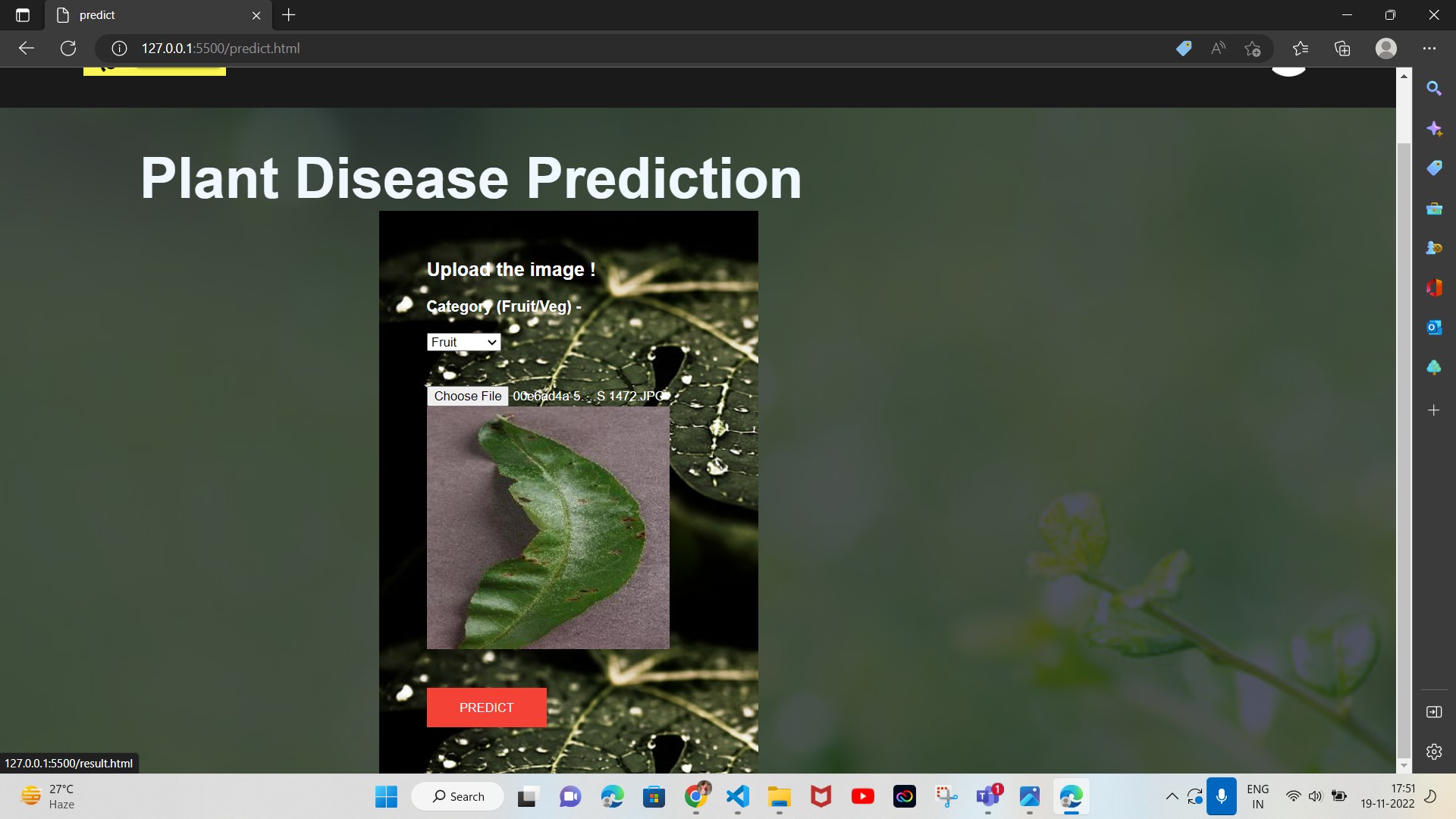


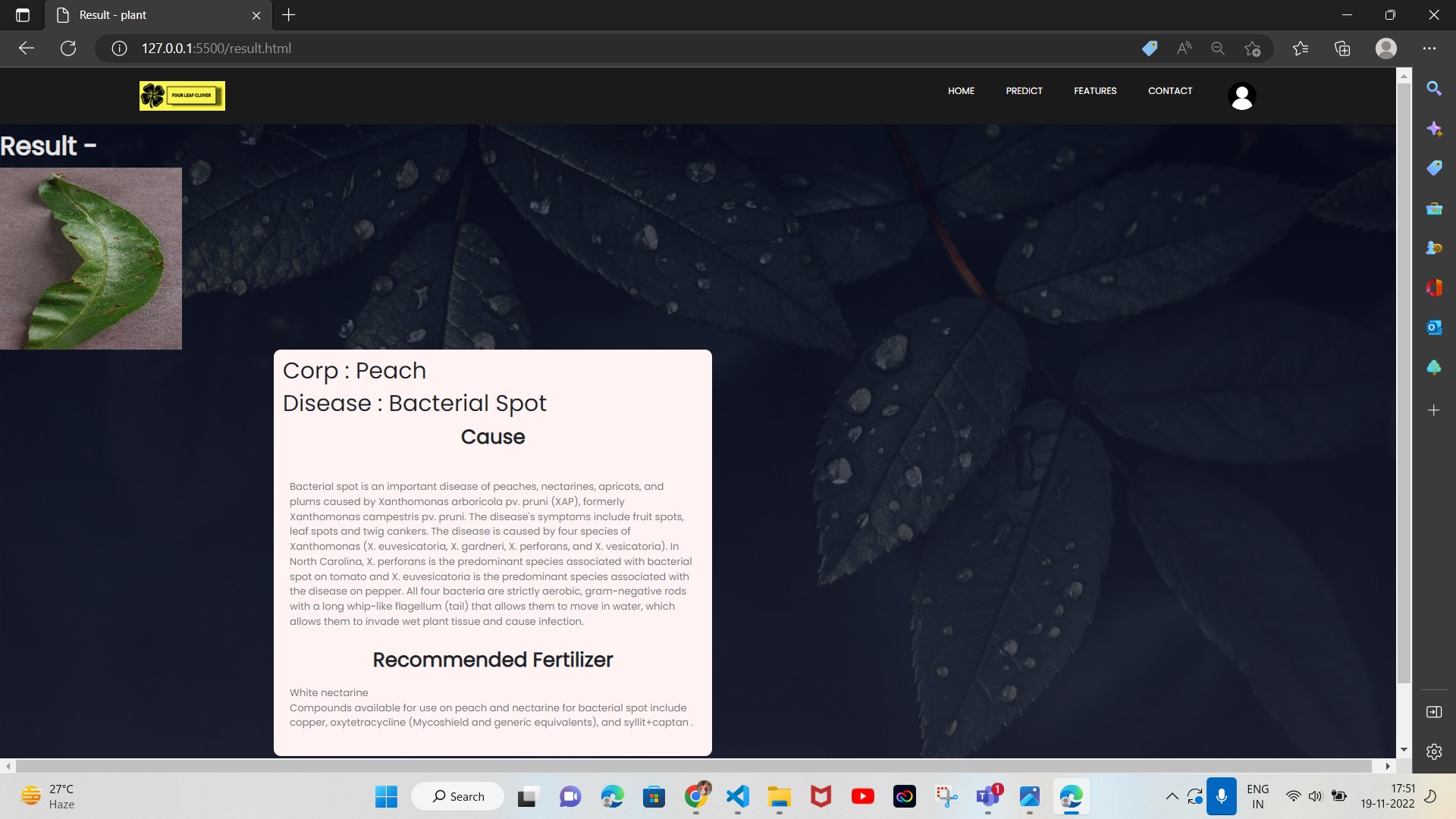












1. **ADVANTAGES & DISADVANTAGES**

**List of advantages**

* The proposed model here produces very high accuracy of classification.
* Very large datasets can also be trained and tested.
* Images of very high can be resized within the proposed itself.

**List of disadvantages**

* For training and testing, the proposed model requires very high computational time.
* The neural network architecture used in this project work has high complexity.

**APPLICATIONS**

1. The trained network model used to classify the image patterns with high accuracy.
2. The proposed model not only used for plant disease classification but also for other image pattern classification such as animal classification.
3. This project work application involves not only image classification but also for pattern recognition.

# 11.CONCLUSIONS

The model proposed here involves image classification of fruit datasets and vegetable datasets. The following points are observed during model testing and training:

* + The accuracy of classification increased by increasing the number of epochs.
  + For different batch sizes, different classification accuracies are obtained.
  + The accuracies are increased by increasing more convolution layers.
  + The accuracy of classification also increased by varying dense layers.
  + Different accuracies are obtained by varying the size of kernel used in the convolution layer output.
* Accuracies are different while varying the size of the train and test datasets**.**

# 12.FUTURE SCOPE

The proposed model in this project work can be extended to image recognition. The entire model can be converted to application software using python to exe software. The real time image

classification, image recognition and video processing are possible with help OpenCV python library. This project work can be extended for security applications such as figure print recognition, iris recognition and face recognition.

# 13.APPENDIX

## Source Code

[**https://github.com/IBM-EPBL/IBM-Project-16563-1659617520**](https://github.com/IBM-EPBL/IBM-Project-16563-1659617520)

## GitHub & Project Demo Link

[**https://github.com/IBM-EPBL/IBM-Project-16563-1659617520**](https://github.com/IBM-EPBL/IBM-Project-16563-1659617520)

**Demo Link** [**https://youtu.be/nIi9Lm8jvHQ**](https://youtu.be/nIi9Lm8jvHQ)