**FERTILIZERS RECOMMENDATION SYSTEM FOR**

**DISEASE PREDICTION**

**TEAM ID : PNT2022TMID23783**

ULASA POOJITHA A. NAVYA

THARANI P K VEMBUDHARSINI V

**1.INTRODUCTION**

1.1 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.

1.2 PURPOSE :

Plant diseases especially on leaves is one of the main reason for reduction in both quality and quantity of food crops. In Agriculture aspects, if a plant is affected by leaf disease ,then it reduces the growth at agricultural level. Finding the leaf disease is an important role of agriculture preservation.After pre-processing using a median layer, segmentation is done and leaf disease is identified. The disease based similarity measure is used for fertilizer recommendation.

**2.LITERATURE SURVEY**

2.1 EXISTING PROBLEM :

A proposed method for leaf disease detection and suggest fertilizers to cure leaf diseases. But the method involves less number of train and test sets which results in poor accuracy.

Pandi selvi proposed a simple prediction method for soil based fertilizer recommendation system for predicted crop diseases.This method gives less accuracy and prediction.

Shiva reddy proposed an IOT based system for leaf disease detection and fertilizer recommendation which is based on machine learning techniques that yields less than 80 percentage accuracies.

2.2 REFERENCES :

[1]. R Indumathi.; N Saagari.; V Thejuswini.; R Swarnareka.;"Leaf Disease Detection and Fertilizer Suggestion",IEEE International Conference on System , Computation , Automation and Networking (ICSCAN) , 29-30 March 2019, DOI:10.1109/ICSCAN.2019.8878781

[2]. P. Pandi Selvi, P.Poornima, "Soil Based Fertilizer Recommendation System for Crop Disease Prediction System",International Journal of Engineering Trends and Applications(IJETA)-Volume 8 Issue 2 , Mar-Apr 2021

[3]. H Shiva reddy, Ganesh hedge , Prof. DR chinnaya3, "IOT based Leaf Disease Detection and Fertilizer Recommendation", International Research Journal of Engineering and Technology (IJRET), Volume: 06 Issue: 11, Nov 2019,e-ISSN:2395-0056

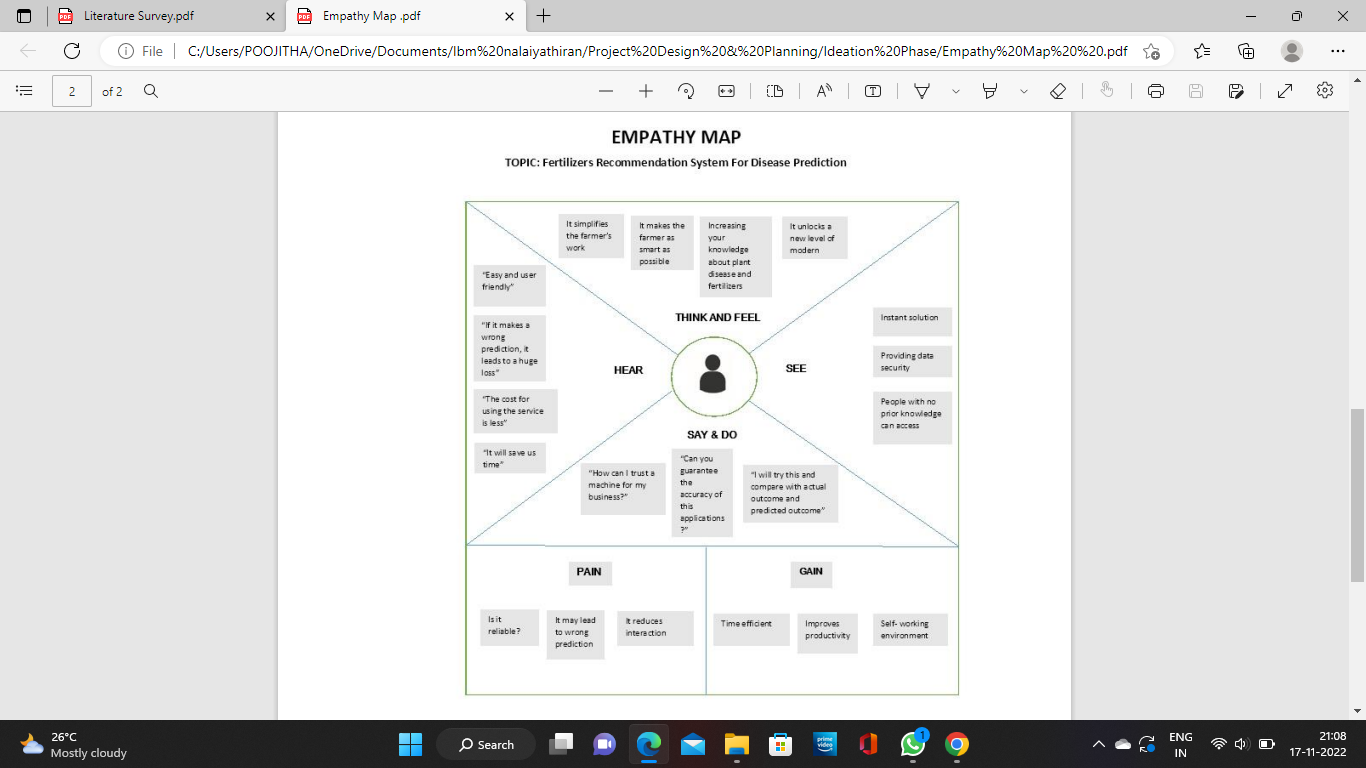
[4]. Detection of Leaf Diseases and Classification using Digital Image Processing International Conference on Innovations in Information, Embedded and Communication Systems(ICIIECS), IEEE, 2017.

2.3 PROBLEM STATEMENT DEFINITION :

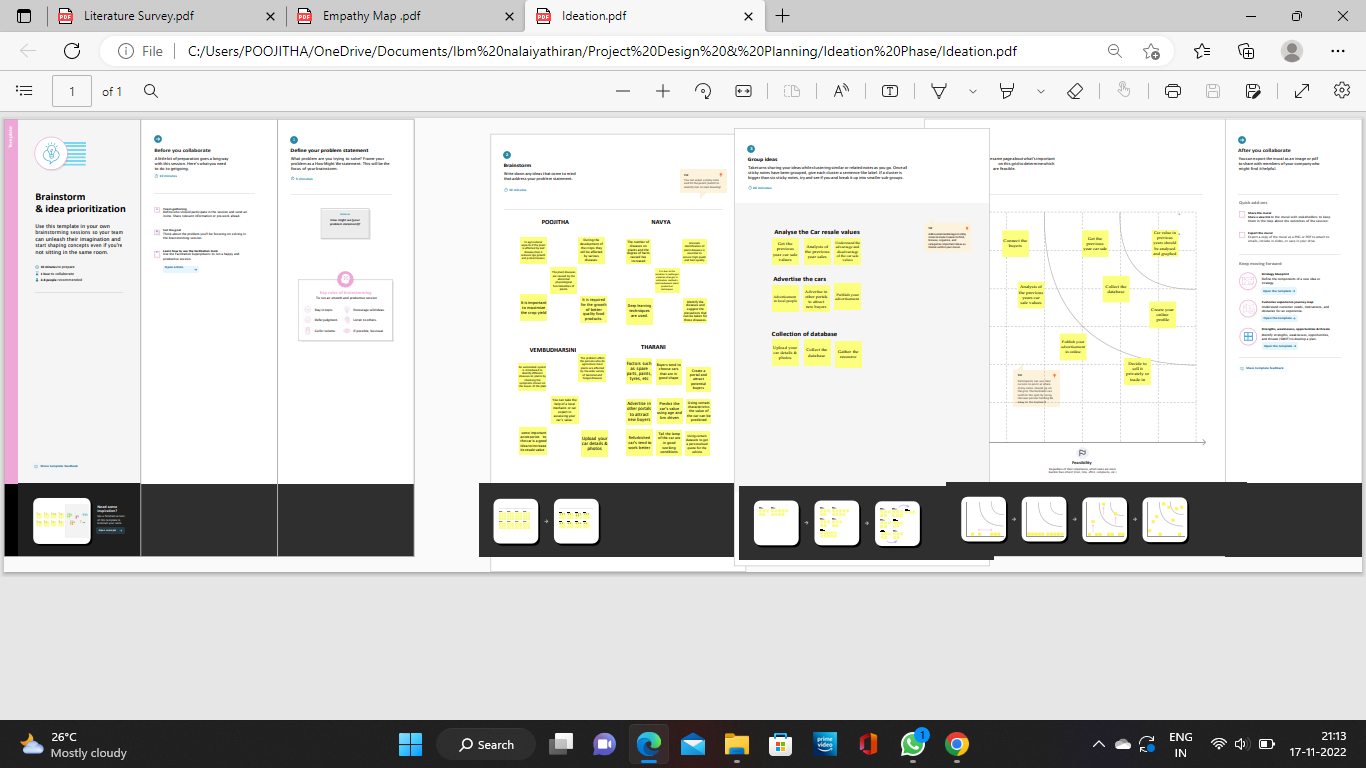
Mostly, the plant leaf diseases are caused by Pathogens which are positioned on the stems of the plants. These different symptoms and diseases of leaves are predicted by different methods in image processing. These different methods include different fundamental processes like segmentation, feature extraction and classification and so on. Mostly, the prediction and diagnosis of leaf diseases are depending on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves.

**3.IDEATION & PROPOSED SOLUTION**

3.1 EMPATHY MAP CANVAS :



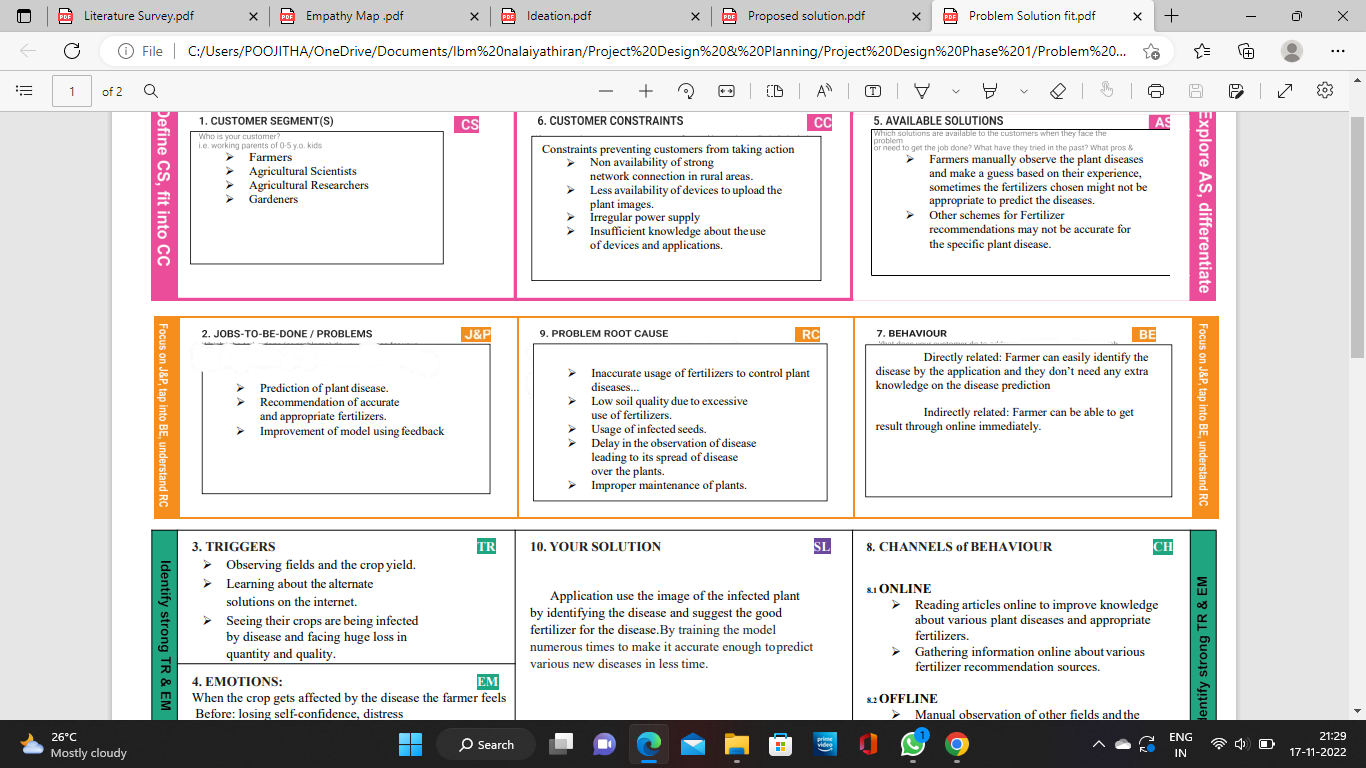
3.2 IDEATION AND BRAINSTORMING :



3.3 PROPOSED SOLUTION :

|  |  |  |
| --- | --- | --- |
| S.no | Parameter | Description |
| 1 | Problem Statement (Problem to be solved) | Disease in plants reduced the quantity and quality of the plants productivity. Identifying the disease in plant is hard to find. |
| 2 | Idea / Solution description | One of the solution of the problem is to identifying the disease in early stage and using the correct fertilizer. |
| 3 | Novelty / Uniqueness | This application can suggest good fertilizer for the disease in the plant by recognizing the images. |
| 4 | Social Impact / Customer Satisfaction | It helps the farmer by identifying the disease in the early stage and increase the quality and quantity of crops in efficient way. |
| 5 | Business Model (Revenue Model) | The application is recommends to farmer in subscription basis. |
| 6 | Scalability of the Solution | This application can be improved by introducing online purchases of crops, fertilizer easily |

3.4 PROBLEM SOLUTION FIT :



**4.REQUIREMENT ANALYSIS**

4.1 FUNCTIONAL REQUIREMENTS:

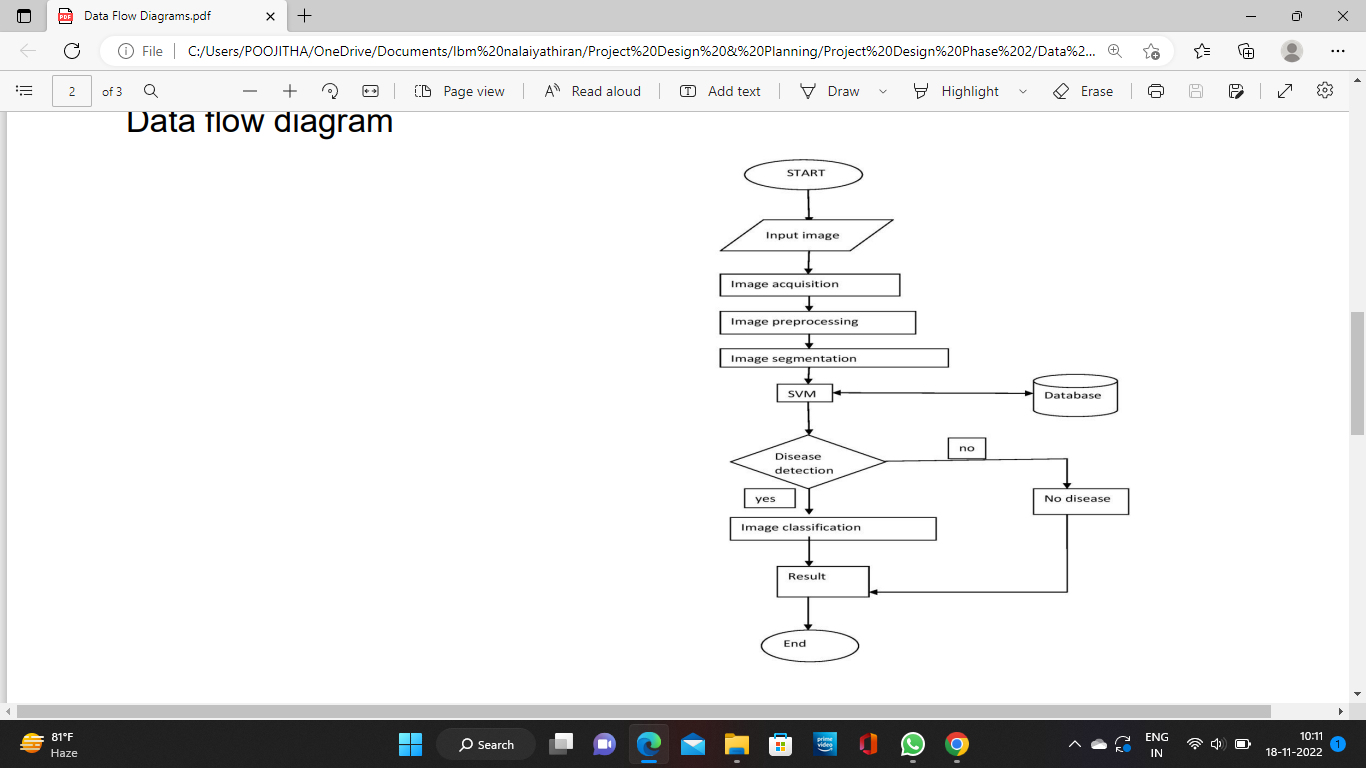
|  |  |
| --- | --- |
| **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| User Registration | Registration through Form ,Gmail , LinkedIN |
| User Confirmation | Confirmation via Email ,Confirmation via OTP |
| Capturing image | Capture the image of the leaf |
| Image processing | Upload the image for the prediction of the disease in the leaf. |
| Leaf identification | Identify the leaf and predict the disease in leaf. |
| Image description | Suggesting the best fertilizer for the disease . |

4.2 NON FUNCTIONAL REQUIREMENTS:

|  |  |
| --- | --- |
| **Non-Functional Requirement** | **Description** |
| Usability | Datasets of all the leaf is used to detecting the disease that present in the leaf. |
| Security | The information belongs to the user and leaf are secured highly. |
| Reliability | The leaf quality is important for the predicting the disease in leaf. |
| Performance | The performance is based on the quality of the leaf used for disease prediction. |
| Availability | It is available for all user to predict the disease in the plant |
| Scalability | Increasing the prediction of the disease in the leaf |

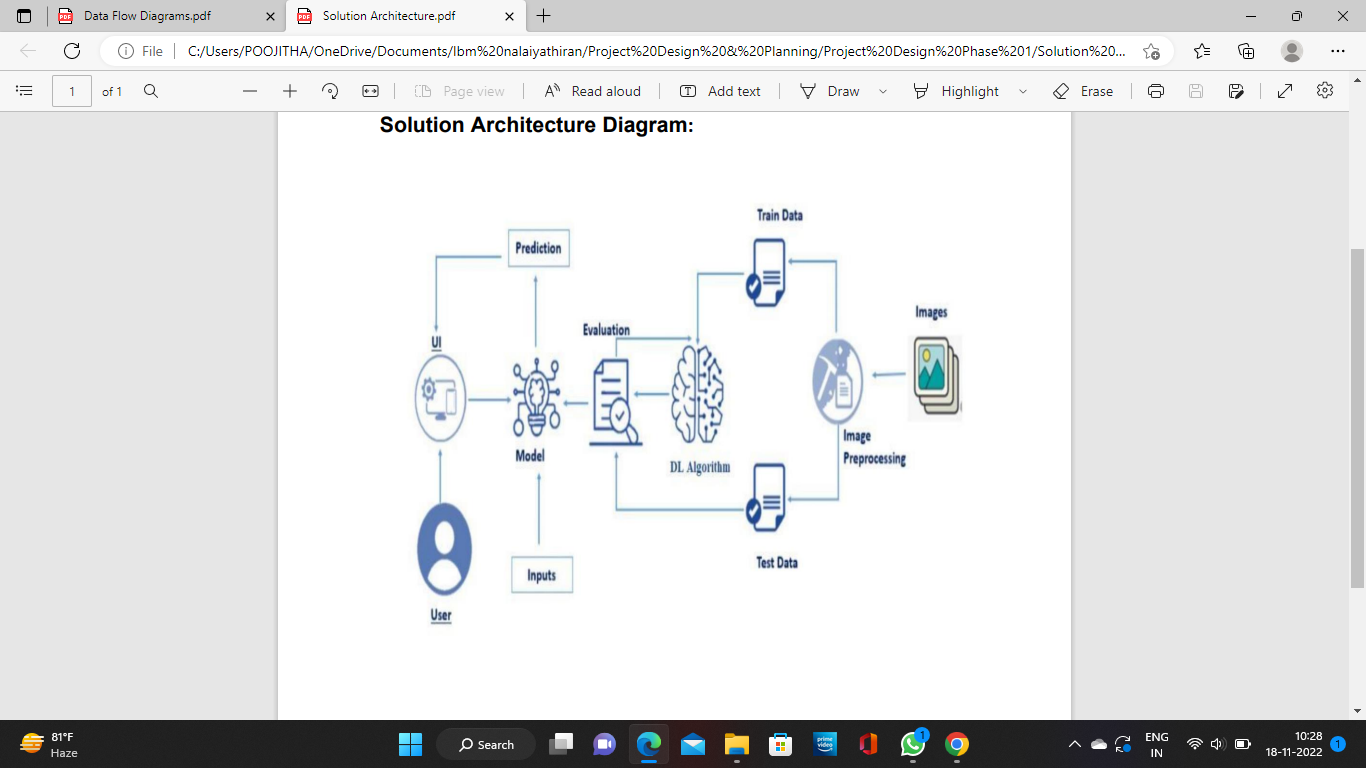
**5.PROJECT DESIGN**

5.1 DATA FLOW DIAGRAMS :

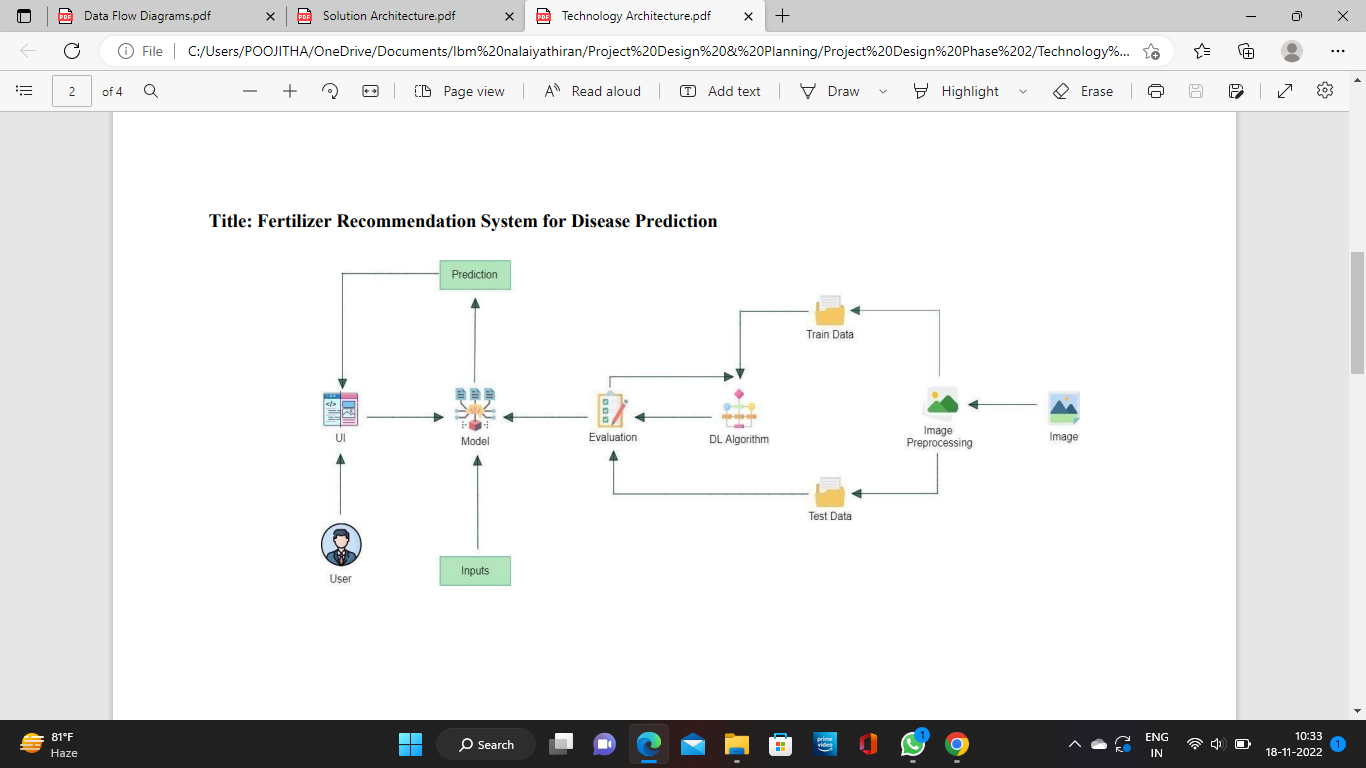


5.2 SOLUTION & TECHNICAL ARCHITECTURE :

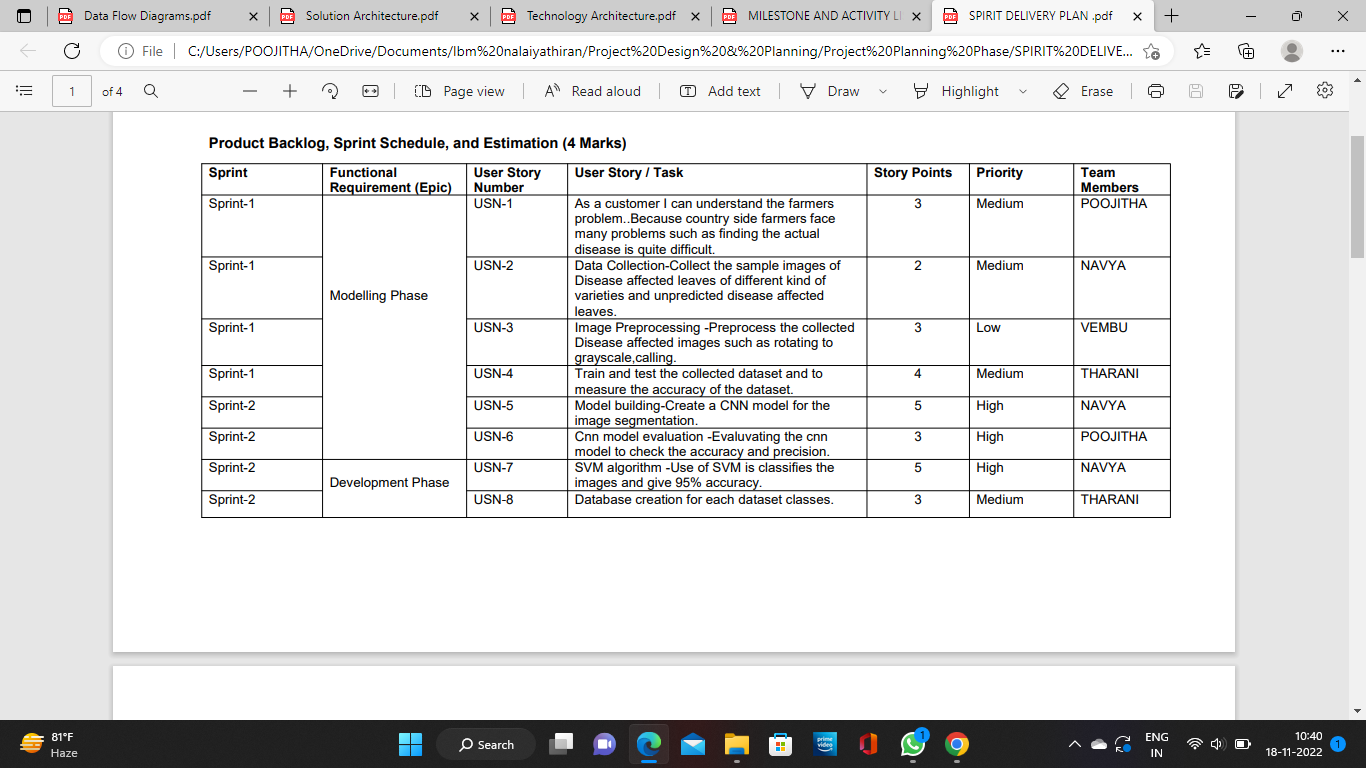
Solution Architecture

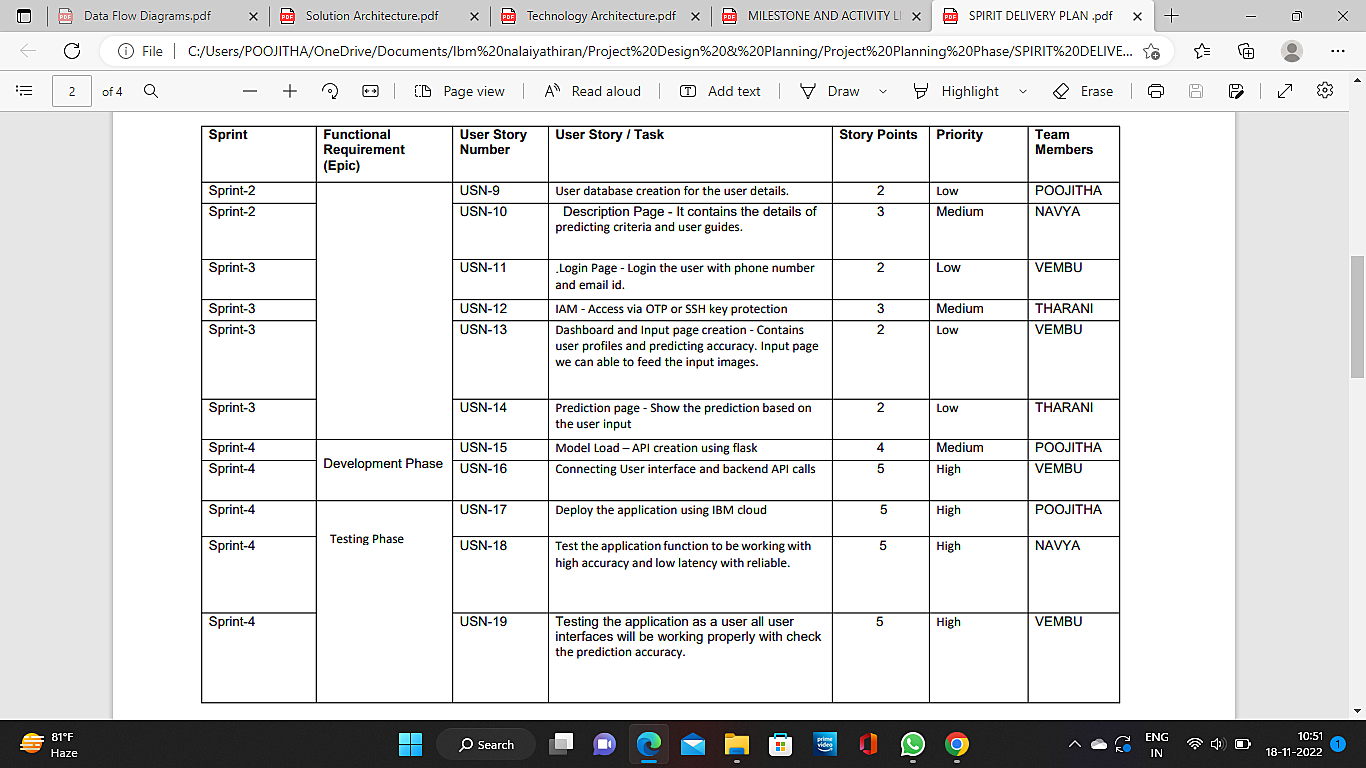


Technical Architecture



5.3 USER STORIES :





**6. PROJECT PLANNING AND SCHEDULING**

6.1 SPRINT PLANNING :

Phrase 1 : Requirement analysis and information collection

Phrase 2: Project planning and developing modules

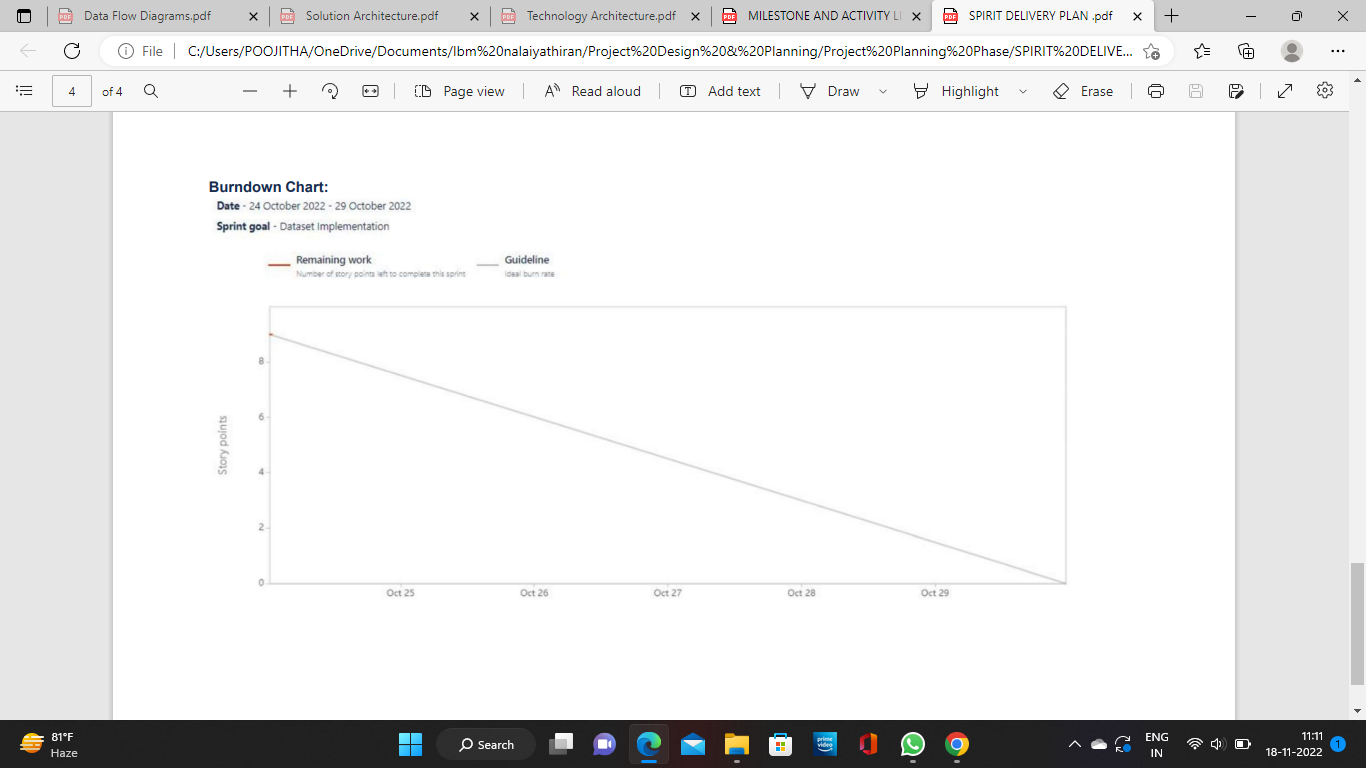
Phrase 3:Implementing highly accurate algorithms

Phrase 4: Deploying the model on cloud and Testing the model

6.2 SPRINT DELIVERY SCHEDULE :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint** | **Story points** | **Duration** | **Start date** | **End date** |
| Sprint 1 | 12 | 6 days | 24 Oct 2022 | 29 Oct 2022 |
| Sprint 2 | 21 | 4 days | 30 Nov 2022 | 02 Nov 2022 |
| Sprint 3 | 09 | 5 days | 03 Nov 2022 | 07 Nov 2022 |
| Sprint 4 | 24 | 5 days | 08 Nov 2022 | 12 Nov 2022 |

6.3 REPORTS FROM JIRA:



Velocity:

Sprint 1 average velocity: Average Velocity = 12 / 6 =2

Sprint 2 average velocity: Average Velocity = 21 / 4 = 5.2

Sprint 3 average velocity: Average Velocity = 09 / 5 = 1.8

Sprint 4 average velocity: Average Velocity = 24 / 5 = 4.8

**7.CODING & SOLUTIONING**

7.1 Feature 1 :

Images in the dataset are preprocessed to ensure efficiency and reliability.

Code :

from keras.preprocessing.image import ImageDataGenerator

train\_datagen = ImageDataGenerator (rescale = 1./255, shear\_range= 0.2,zoom\_range= 0.2, horizontal\_flip = True)

###### test\_datagen =ImageDataGenerator (rescale = 1)

x\_train = train\_datagen.flow\_from\_directory(r'C:\Users\POOJITHA\Downloads\Fertilizers\_Recommendation\_ System\_For\_Disease\_ Prediction\Dataset Plant Disease\fruit-dataset\fruit-dataset\train',target\_size = (128,128), batch\_size = 32, class\_mode = 'categorical')

x\_test = test\_datagen.flow\_from\_directory(r'C:\Users\POOJITHA\Downloads\Fertilizers\_Recommendation\_ System\_For\_Disease\_ Prediction\Dataset Plant Disease\fruit-dataset\fruit-dataset\test',target\_size = (128,128), batch\_size = 32, class\_mode = 'categorical')

Found 5384 images belonging to 6 classes.

Found 1686 images belonging to 6 classes.

x\_train = train\_datagen.flow\_from\_directory(r'C:\Users\POOJITHA\Downloads\Fertilizers\_Recommendation\_ System\_For\_Disease\_ Prediction\Dataset Plant Disease\Veg-dataset\Veg-dataset\train\_set',target\_size = (128,128), batch\_size = 32, class\_mode = 'categorical')

x\_test = test\_datagen.flow\_from\_directory(r'C:\Users\POOJITHA\Downloads\Fertilizers\_Recommendation\_ System\_For\_Disease\_ Prediction\Dataset Plant Disease\Veg-dataset\Veg-dataset\test\_set',target\_size = (128,128), batch\_size = 32, class\_mode = 'categorical')

Found 11386 images belonging to 9 classes.

Found 3416 images belonging to 9 classes.

7.2 FEATURE 2 :

Histogram analysis is performed to identify all dimensions in image processing.

Code :

import matplotlib.pyplot as plt

import numpy as np

from skimage.io import imread

I = imread('/content/23ea1618-d554-47fb-bc03-a1b978f14fbf\_\_\_RS\_HL 6008.JPG')

J = imread('/content/25de086c-ea7e-42b0-83fd-bc7d1e584d0a\_\_\_RS\_HL 5852.JPG')

plt.figure()

plt.subplot(121), plt.imshow(I)

plt.subplot(122), plt.imshow(J)

plt.show()

plt.figure(figsize=(10, 10))

plt.imshow(np.abs(I[:, :, 0].astype(float) - J[:, :, 0].astype(float)), cmap='gray')

plt.show()

d = imread('/content/23ea1618-d554-47fb-bc03-a1b978f14fbf\_\_\_RS\_HL 6008.JPG')

mask = imread('/content/25de086c-ea7e-42b0-83fd-bc7d1e584d0a\_\_\_RS\_HL 5852.JPG')

print(np.amin(d), np.amax(d))

print(np.amin(mask), np.amax(mask))

plt.figure(), plt.imshow(mask), plt.show()

mask = mask[:, :, 0]

maskInv = np.zeros\_like(mask)

maskInv[mask == 0] = 255

maskInv[mask == 255] = 0

plt.figure(), plt.imshow(maskInv, cmap='gray'), plt.show()

**8. TESTING**

8.1 TEST CASES :

Code :

test\_dir=r"C:\Users\POOJITHA\Downloads\Fertilizers\_Recommendation\_ System\_For\_Disease\_ Prediction\Dataset Plant Disease\fruit-dataset\fruit-dataset\test"

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras.preprocessing.image import ImageDataGenerator

model = tf.keras.models.load\_model(r"C:\Users\POOJITHA\Downloads\Fertilizers\_Recommendation\_ System\_For\_Disease\_ Prediction\Dataset Plant Disease\fruit.h5")

test\_datagen\_1=ImageDataGenerator(rescale=1)

test\_generator\_1=test\_datagen\_1.flow\_from\_directory(

test\_dir,

target\_size=(128,128),

batch\_size=20,

class\_mode='categorical'

)

import numpy as np

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import imagei

img=image.load\_img(r"C:\Users\POOJITHA\Downloads\Fertilizers\_Recommendation\_ System\_For\_Disease\_ Prediction\Dataset Plant Disease\fruit-dataset\fruit-dataset\train\Corn\_(maize)\_\_\_healthy\11c1e3ce-73d1-4338-8939-473087e9dcbb\_\_\_R.S\_HL 0604.JPG")

mg=image.load\_img(r"C:\Users\POOJITHA\Downloads\Fertilizers\_Recommendation\_ System\_For\_Disease\_ Prediction\Dataset Plant Disease\fruit-dataset\fruit-dataset\train\Corn\_(maize)\_\_\_healthy\11c1e3ce-73d1-4338-8939-473087e9dcbb\_\_\_R.S\_HL 0604.JPG")

img

img=image.load\_img(r"C:\Users\POOJITHA\Downloads\Fertilizers\_Recommendation\_ System\_For\_Disease\_ Prediction\Dataset Plant Disease\fruit-dataset\fruit-dataset\train\Corn\_(maize)\_\_\_healthy\11c1e3ce-73d1-4338-8939-473087e9dcbb\_\_\_R.S\_HL 0604.JPG",target\_size=(128,128))

x=image.img\_to\_array(img)

x=np.expand\_dims(x,axis=0)

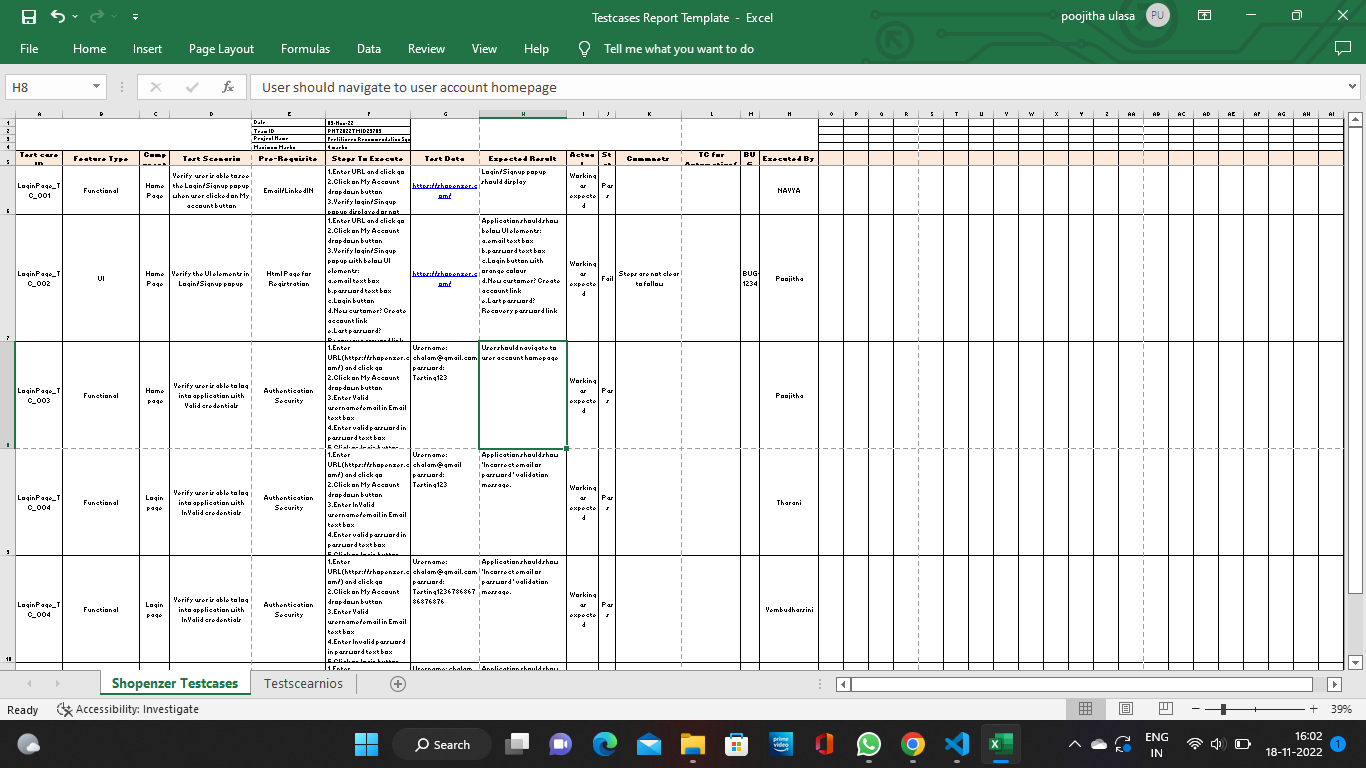
y=np.argmax(model.predict(x),axis=1)

index=['Apple\_\_\_Black\_rot', 'Apple\_\_\_healthy', 'Corn\_(maize)\_\_\_healthy', 'Corn\_(maize)\_\_\_Northern\_Leaf\_Blight', 'Peach\_\_\_Bacterial\_spot', 'Peach\_\_\_healthy']

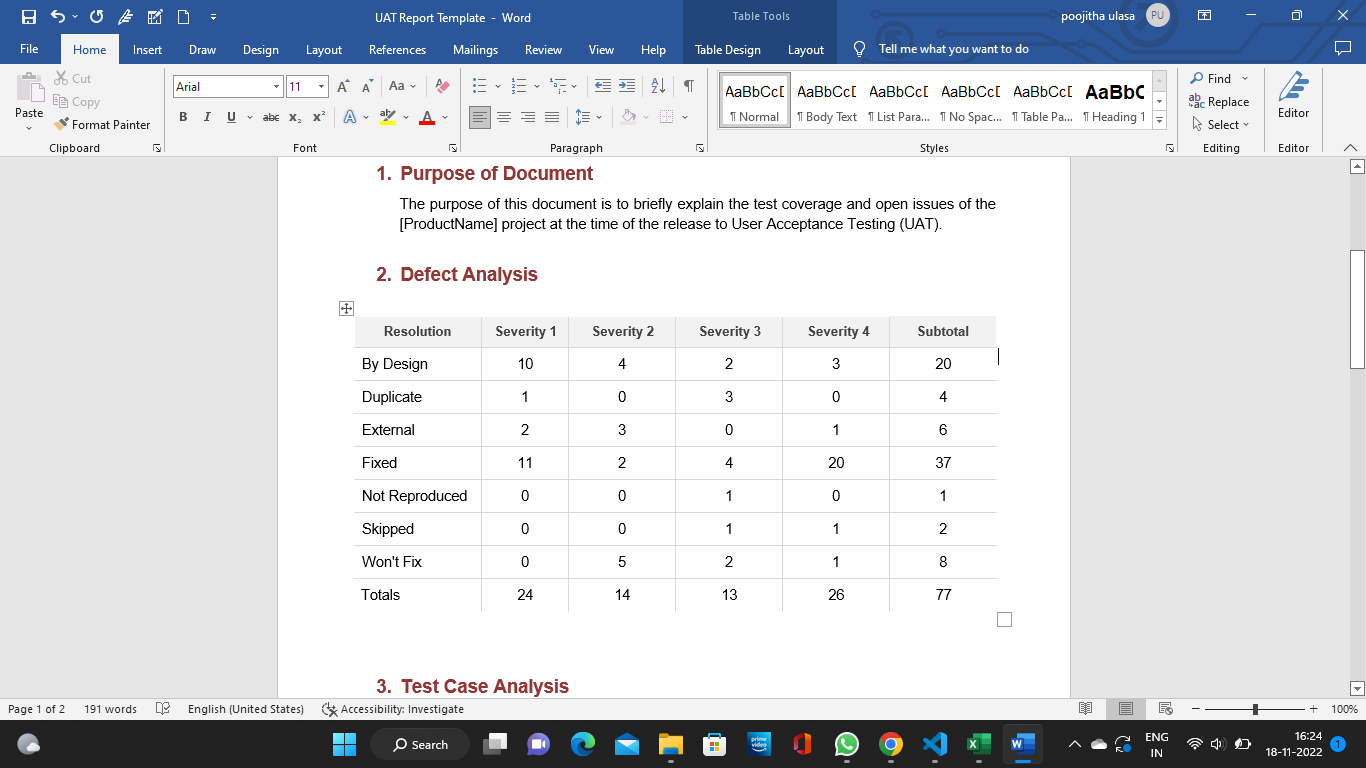
index[y[0]]

model.evaluate(test\_generator\_1,steps=50)

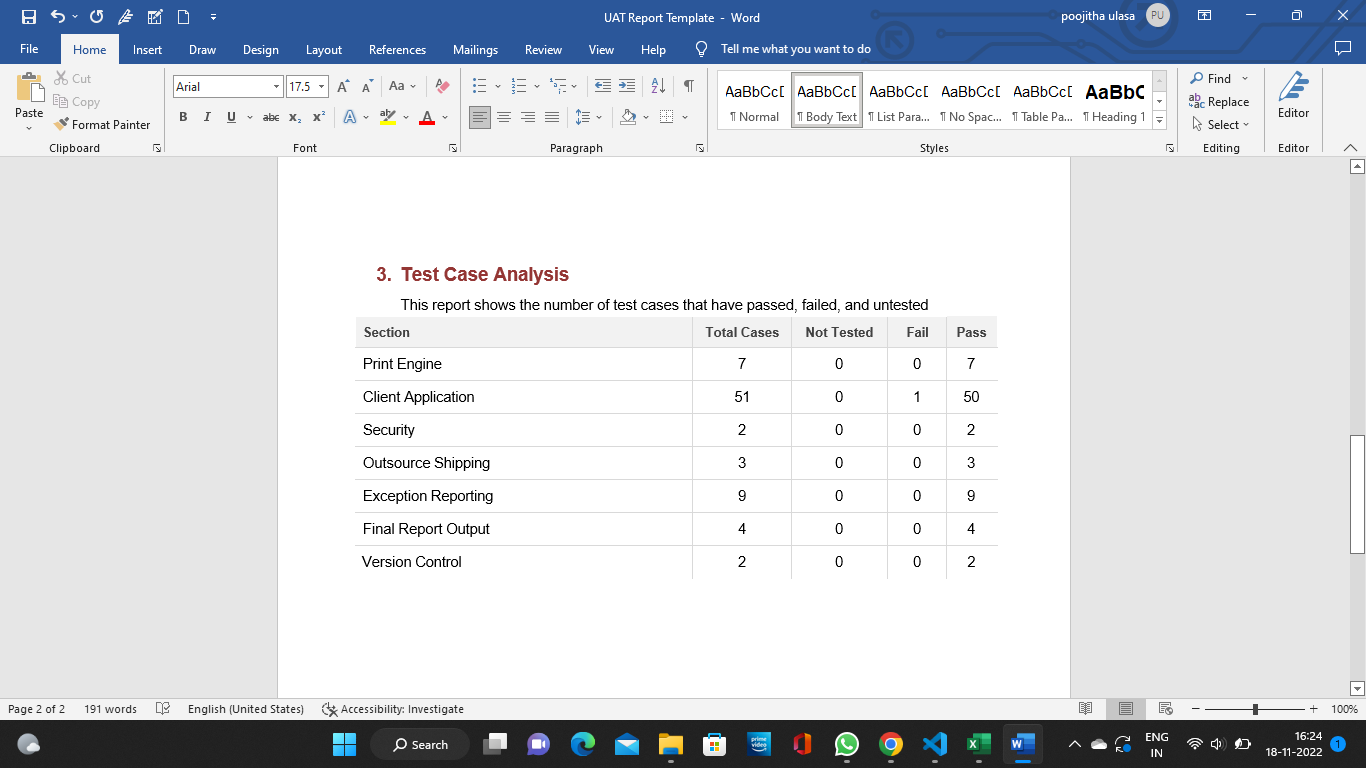
TEST CASES :

8.2 USER ACCEPTANCE TESTING :

Defect Analysis :



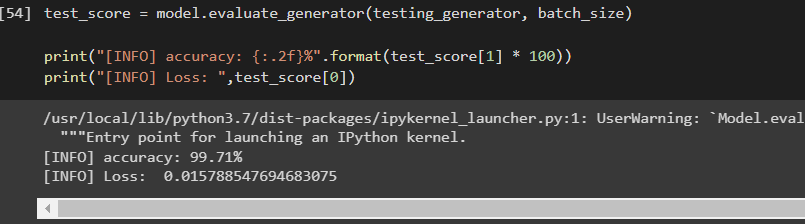
Test Case Analysis :



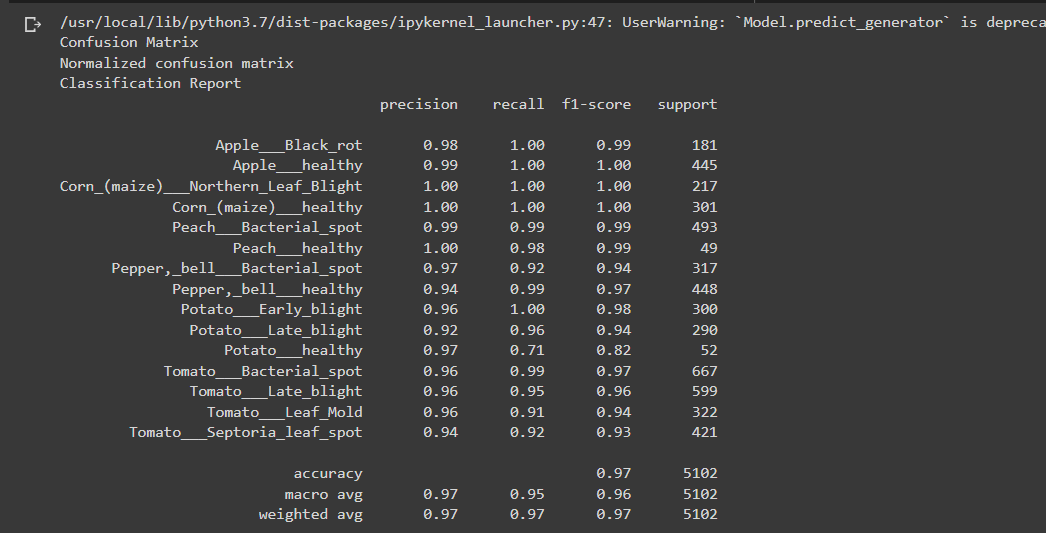
**9.RESULTS**

9.1 PERFORMANCE METRICS :

Model Summary :



Values :



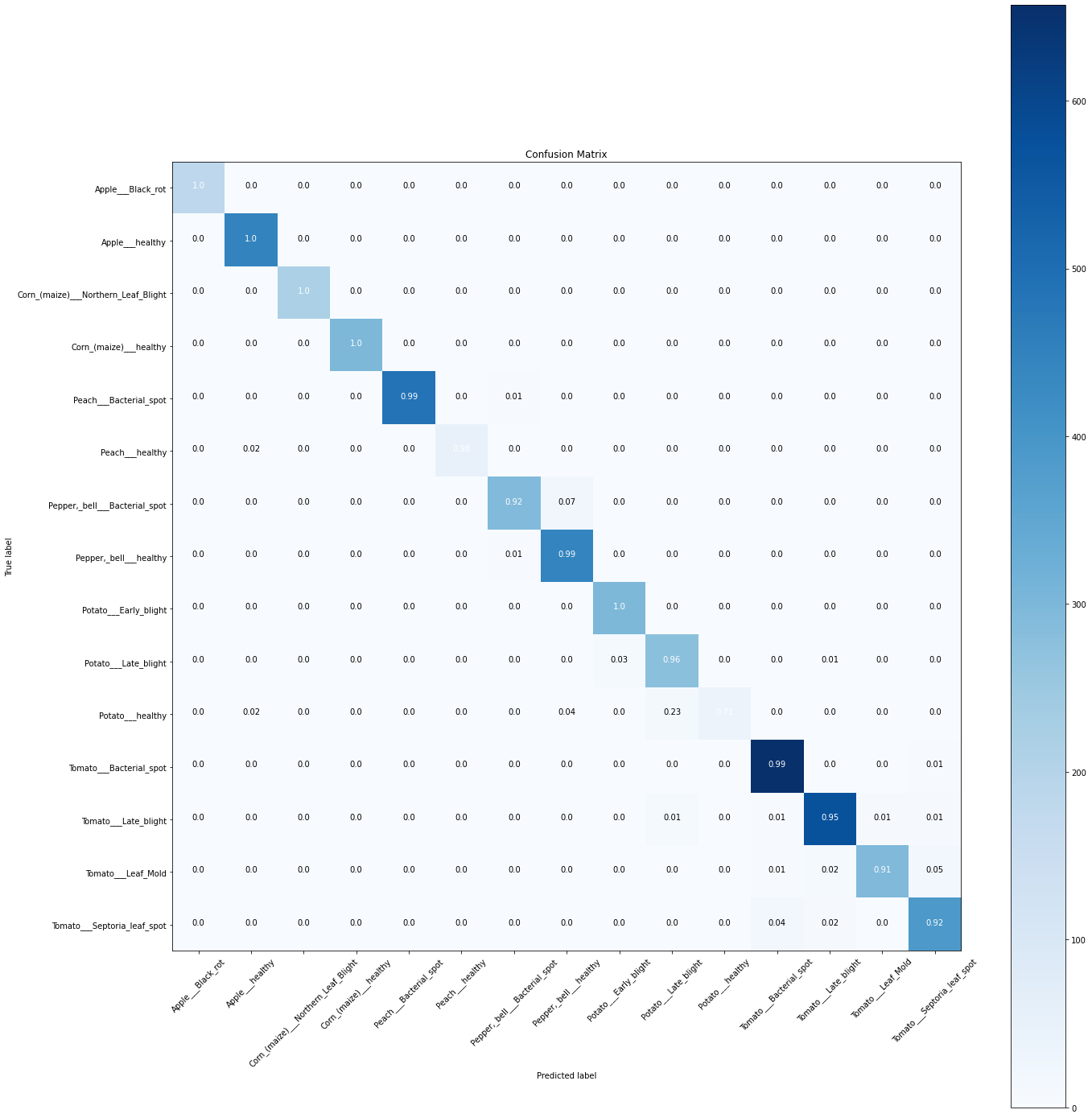
Accuracy :



Training accuracy : 0.95

Validation accuracy : 0.98

Model Confusion Matrix :



**10.ADVANTAGES & DISADVANTAGES**

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.

DISADVANTAGES :

* For training and testing , computational time is a little high.
* The neural network architecture used in this project work is complex.

**11.CONCLUSION**

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 dataset.

**12.FUTURE SCOPE**

This further research is implementing the proposed algorithm with the existing public datasets.Also, various segmentation algorithms can be implemented to improve accuracy. The proposed algorithm can be modified further to identify the disease that affects the various other plant organs such as stems and fruits.

**13.APPENDIX**

SOURCE CODE :

Python Code

app.py

import requests

from tensorflow.keras.preprocessing import image

from tensorflow.keras.models import load\_model

import numpy as np

import pandas as pd

import tensorflow as tf

from flask import Flask, request, render\_template, redirect, url\_for

import os

from werkzeug.utils import secure\_filename

from tensorflow.python.keras.backend import set\_session

app = Flask(\_\_name\_\_)

#load both the vegetable and fruit models

model = load\_model("vegetable.h5")

model1=load\_model("fruit.h5")

#home page

@app.route('/')

def home():

return render\_template('home.html')

#prediction page

@app.route('/prediction')

def prediction():

return render\_template('predict.html')

@app.route('/predict',methods=['POST'])

def predict():

if request.method == 'POST':

# Get the file from post request

f = request.files['image']

# Save the file to ./uploads

basepath = os.path.dirname(\_\_file\_\_)

file\_path = os.path.join(

basepath, 'uploads', secure\_filename(f.filename))

f.save(file\_path)

img = image.load\_img(file\_path, target\_size=(128, 128))

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis=0)

plant=request.form['plant']

print(plant)

if(plant=="vegetable"):

preds = model.predict(x)

preds=np.argmax(preds)

print(preds)

df=pd.read\_excel('precautions - veg.xlsx')

print(df.iloc[preds]['caution'])

else:

preds = model1.predict(x)

preds=np.argmax(preds)

df=pd.read\_excel('precautions - fruits.xlsx')

print(df.iloc[preds]['caution'])

return df.iloc[preds]['caution']

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=False)

HTML CODE :

home.html

<!DOCTYPE html>

<html >

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title> Plant Disease Prediction</title>

<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

<link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">

<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>

<style>

.header {

top:0;

margin:0px;

left: 0px;

right: 0px;

position: fixed;

background-color: #28272c;

color: white;

box-shadow: 0px 8px 4px grey;

overflow: hidden;

padding-left:20px;

font-family: 'Josefin Sans';

font-size: 2vw;

width: 100%;

height:8%;

text-align: center;

}

.topnav {

overflow: hidden;

background-color: skyblue;

}

.topnav-right a {

float: left;

color: #f2f2f2;

text-align: center;

padding: 14px 16px;

text-decoration: none;

font-size: 18px;

}

.topnav-right a:hover {

background-color: #ddd;

color: black;

}

.topnav-right a.active {

background-color: #565961;

color: white;

}

.topnav-right {

float: right;

padding-right:100px;

}

body {

background-color:#ffffff;

background-repeat: no-repeat;

background-size:cover;

background-position: 0px 0px;

}

.button {

background-color: #28272c;

border: none;

color: white;

padding: 15px 32px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 16px;

border-radius: 12px;

}

.button:hover {

box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24), 0 17px 50px 0 rgba(0,0,0,0.19);

}

form {border: 3px solid #f1f1f1; margin-left:400px;margin-right:400px;}

input[type=text], input[type=password] {

width: 100%;

padding: 12px 20px;

display: inline-block;

margin-bottom:18px;

border: 1px solid #ccc;

box-sizing: border-box;

}

button {

background-color: #28272c;

color: white;

padding: 14px 20px;

margin-bottom:8px;

border: none;

cursor: pointer;

width: 15%;

border-radius:4px;

}

button:hover {

opacity: 0.8;

}

.cancelbtn {

width: auto;

padding: 10px 18px;

background-color: #f44336;

}

.imgcontainer {

text-align: center;

margin: 24px 0 12px 0;

}

img.avatar {

width: 30%;

border-radius: 50%;

}

.container {

padding: 16px;

}

span.psw {

float: right;

padding-top: 16px;

}

/\* Change styles for span and cancel button on extra small screens \*/

@media screen and (max-width: 300px) {

span.psw {

display: block;

float: none;

}

.cancelbtn {

width: 100%;

}

}

.home{

margin:80px;

width: 84%;

height: 500px;

padding-top:10px;

padding-left: 30px;

}

.login{

margin:80px;

box-sizing: content-box;

width: 84%;

height: 420px;

padding: 30px;

border: 10px solid blue;

}

.left,.right{

box-sizing: content-box;

height: 400px;

margin:20px;

border: 10px solid blue;

}

.mySlides {display: none;}

img {vertical-align: middle;}

/\* Slideshow container \*/

.slideshow-container {

max-width: 1000px;

position: relative;

margin: auto;

}

/\* Caption text \*/

.text {

color: #f2f2f2;

font-size: 15px;

padding: 8px 12px;

position: absolute;

bottom: 8px;

width: 100%;

text-align: center;

}

/\* The dots/bullets/indicators \*/

.dot {

height: 15px;

width: 15px;

margin: 0 2px;

background-color: #bbb;

border-radius: 50%;

display: inline-block;

transition: background-color 0.6s ease;

}

.active {

background-color: #717171;

}

/\* Fading animation \*/

.fade {

-webkit-animation-name: fade;

-webkit-animation-duration: 1.5s;

animation-name: fade;

animation-duration: 1.5s;

}

@-webkit-keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

@keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

/\* On smaller screens, decrease text size \*/

@media only screen and (max-width: 300px) {

.text {font-size: 11px}

}

</style>

</head>

<body style="font-family:'Times New Roman', Times, serif;background-color:#C2C5A8;">

<div class="header">

<div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Plant Disease Prediction</div>

<div class="topnav-right"style="padding-top:0.5%;">

<a class="active" href="{{ url\_for('home')}}">Home</a>

<a href="C:\Users\POOJITHA\OneDrive\Documents\Predict.html" class="button">Predict</button></a>

</div>

</div>

<div style="background-color:#ffffff;">

<div style="width:60%;float:left;">

<div style="font-size:50px;font-family:Montserrat;padding-left:20px;text-align:center;padding-top:10%;">

<b>Know Your Plant!</b></div><br>

<div style="font-size:20px;font-family:Montserrat;padding-left:70px;padding-right:30px;text-align:justify;">Agriculture was the essential development in the rise of human

civilization, whereby farming of acclimatize species produced food oversupply that enabled people to reside in cities.

Plants were independently sophisticated in at least 11 regions of the world. Industrial agriculture based on large-scale monocropping in the twentieth century came to influence agricultural output, though

about 2 billion people still depended on maintaining agriculture.The plant diseases effect the production. Identification of diseases and taking necessary precautions is all done through naked eye, which requires labour and laboratries. This application helps farmers in detecting the diseases by observing the spots on the leaves, which inturn saves effort and labor costs.</div><br><br>

</div>

</div>

<div style="width:40%;float:right;"><br><br>

<img src="C:\Users\POOJITHA\Downloads\leaves.jpeg"style="max-height:200%;max-width:200%;">

</div>

</div>

<div class="home">

<br>

</div>

<script>

var slideIndex = 0;

showSlides();

function showSlides() {

var i;

var slides = document.getElementsByClassName("mySlides");

var dots = document.getElementsByClassName("dot");

for (i = 0; i < slides.length; i++) {

slides[i].style.display = "none";

}

slideIndex++;

if (slideIndex > slides.length) {slideIndex = 1}

for (i = 0; i < dots.length; i++) {

dots[i].className = dots[i].className.replace(" active", "")

}

}

</script>

</body>

</html>

predict.html :

<!DOCTYPE html>

<html lang="en">

<head>

<title>predict</title>

</head>

<style>

.container{

display: flex;

padding: 60px 70px 60px 70px;

}

.card{

padding: 70px 80px 70px 80px;

}

.menu{

padding: 10px 10px 10px 10px;

background-color: black;

color: white;

font-size: 15pt;

}

</style>

<body>

<div class="menu">

<ul ><li>Plant disease Prediction</li></ul></div>

<div class="container">

<img src="C:\Users\POOJITHA\Downloads\Fields.jpeg">

<div class="card">

<form>

<h1>Drop in the image to get the Prediction </h1><br><br>

<label><select name="Fruit" id="plant">

<option value="fruit" id="fruit">Fruit</option>

<option value="vagitable" id="vig">vegitable</option>

</select>

</label><br><br><br>

<input id="default-btn" type="file" name="" onchange="document.getElementById('output').src=window.URL.createObjectURL(this.files[0])"><br><br><br>

<img src="" id="output">

<br><button id="button" onclick ="display()" >Predict!</button></br>

</form>

</body>

</html>

DATE SET LINK :

<https://drive.google.com/file/d/1fxs7ptI6zh7NTbCOZARKZ7AmYKjnprrY/view?usp=sharing>

FLASK FILES :

<https://drive.google.com/drive/folders/1C0Qu-O8T-zZPO7scXTdUDzs8-vzNC-ge>

GitHub LINK :

<https://github.com/IBM-EPBL/IBM-Project-34603-1660239698>

DEMO VIDEO LINK :

<https://drive.google.com/drive/u/0/folders/1M7q068LcdA6CoEsRpljTBmPQleO8kK0U>