# NALAIYA THIRAN PROGRAM

## FERTILLIZER RECOMMENDATION SYSTEM

## FOR DISEASE PREDICTION

## PROJECT REPORT

### SUBMITTED BY

---


| | |
|---|---|
| **SWETHA C** | **722819205052** |
| **NALINA M** | **722819205027** |
| **RENUKA R** | **722819205032** |
| **SUBANANDHANA R** | **722819205051** |


---

| | |
|---|---|
| **INDUSTRY MENTOR** | **Ms. DURGA PRASAD** |
| **FACULTY MENTOR** | **Ms. MINU BALAKRISHNAN** |

---


**SRI ESHWAR COLLEGE OF ENGINEERING**

**AN AUTONOMOUS INSTITUITION**

**COIMBATORE**

# CONTENTS

7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    a. Feature 1
    b. Feature 2
    c. Database Schema (if Applicable)
8. **TESTING**
    a. Test Cases
    b. User Acceptance Testing
9. **RESULTS**
    a. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

# 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

It allows us to predict which crops would be appropriate for a given climate. Using the weather and disease related data sets, the crop quality can also be improved. Prediction algorithms help us to classify the data based on the disease, and data extracted from the classifier is used to predict soil and crop.

# 2. LITERATURE SURVEY

## 2.1. EXISTING PROBLEM

The growth of crops which will earn them most profit. It will help in maintaining nutrients content in the soil. Both quantity and quality will be increased. The prediction of crop yield is based on the soil data and proper implementation of algorithms have proved that higher crop is achieved. the soil is based on soil type, land type, nutrients along with temperature and electrical conductivity of soil. Alternate crops also can be grown for the particular season as requested by the farmers.

## 2.2 REFERENCES


- Gosai, Dhruvi, Chintal Raval, Rikin Nayak, Hardik Jayswal, and Axat Patel. "Crop Recommendation System using Machine Learning." (2021).

- Bondre, D. A., & Mahagaonkar, S. (2019). Prediction of crop yield and fertilizer

recommendation using machine learning algorithms. International Journal of Engineering Applied Sciences and Technology, 4(5), 371-376.

- Archana, K., and K. G. Saranya. "Crop Yield Prediction, Forecasting and Fertilizer Recommendation using Voting Based Ensemble Classifier." SSRG Int. J. Comput. Sci. Eng 7 (2020): 1-4.

- Neela, R., & Nithya, P. (2019). Fertilizers Recommendation System For Disease Prediction In Tree Leaves. International Journal Of Scientific & Technology Research, ISSN, (2277-8616).

- Mane, Vijay, Akash Gajbhiye, Chinmay Deshmukh, and Kunal Gaikwad. "Krishi Mitra: Crop and Fertilizer Recommendations System Using Machine Learning Algorithm." In Pervasive Computing and Social Networking, pp. 309-317. Springer, Singapore, 2023.

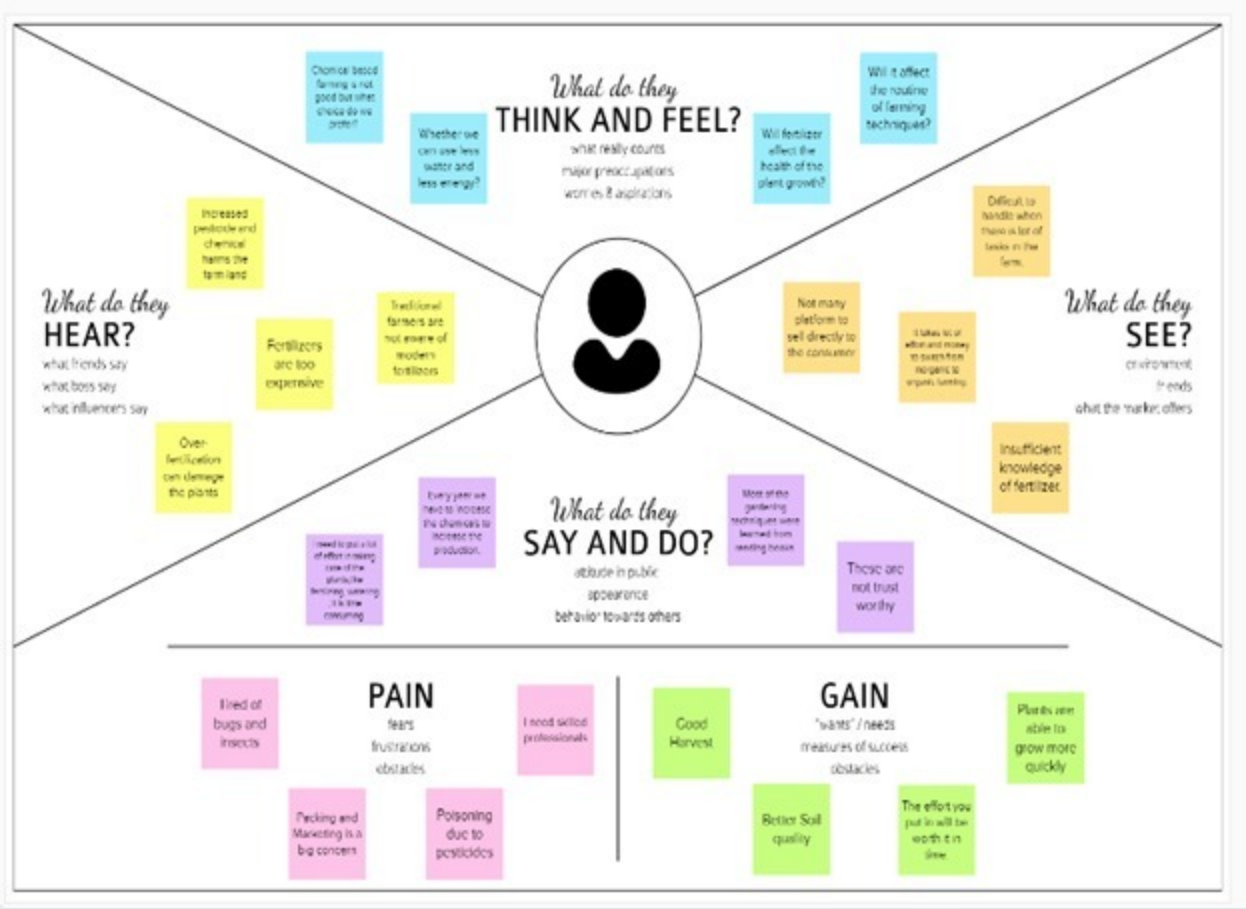
- Chougule, A., Jha, V. K., & Mukhopadhyay, D. (2019). Crop suitability and fertilizers recommendation using data mining techniques. In Progress in Advanced Computing and Intelligent Engineering (pp. 205-213). Springer, Singapore.

## 2.3 PROBLEM STATEMENT DEFINITION

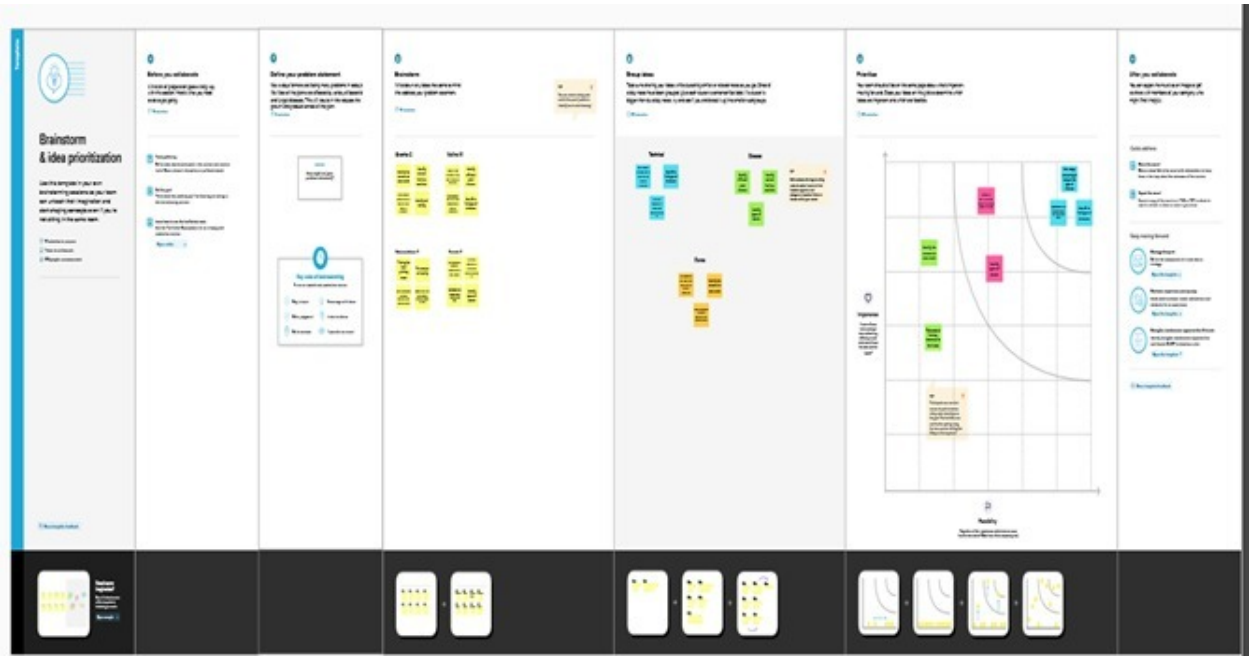
The Agriculture industry is extremely vital and crucial for economic and social development and jobs. In India, the agricultural sector provides a living for almost 48% of the population. As per the 2019-2020 economic survey, an Indian farmer's median wage in 16 states is Rupees 2500. Most of the Indian population depends on agriculture for their livelihood. Agriculture gives an opportunity of employment to the village people to develop a country like India on large scale and give a push in the economic sector. The majority of farmers face the problem of planting an inappropriate crop for their land based on a conventional or non-scientific approach. This is a challenging task for a country like India, where agriculture feeds approximately 42% of the population. And the outcomes for the farmer of choosing the wrong crop for land is moving towards metro city for livelihoods, suicide, quitting the agriculture and give land on lease to industrialist or use for the non-agriculture purpose. The outcome of wrong crop selection is less yield and less profit.

# 3. IDEATION AND PROPOSED SOLUTION

## 3.1 EMPATHY MAP CANVAS



## 3.2 IDEATION AND BRAINSTORMING




## 3.3 PROPOSED SOLUTION

**Problem Statement:** Most of the plants are affected by variety of bacterial and fungal diseases. This will result in the reduces the growth and productiveness of the plant

**Idea / Solution Description:** The solution of the problem is to identifying the diseases of the crops by using image processing and then it provides the correct fertilizers to the farmers.

**Novelty / Uniqueness**: Capturing the image of the crop will be in required picture to identify the disease of the crop and suggest related fertilizers to the farmers.

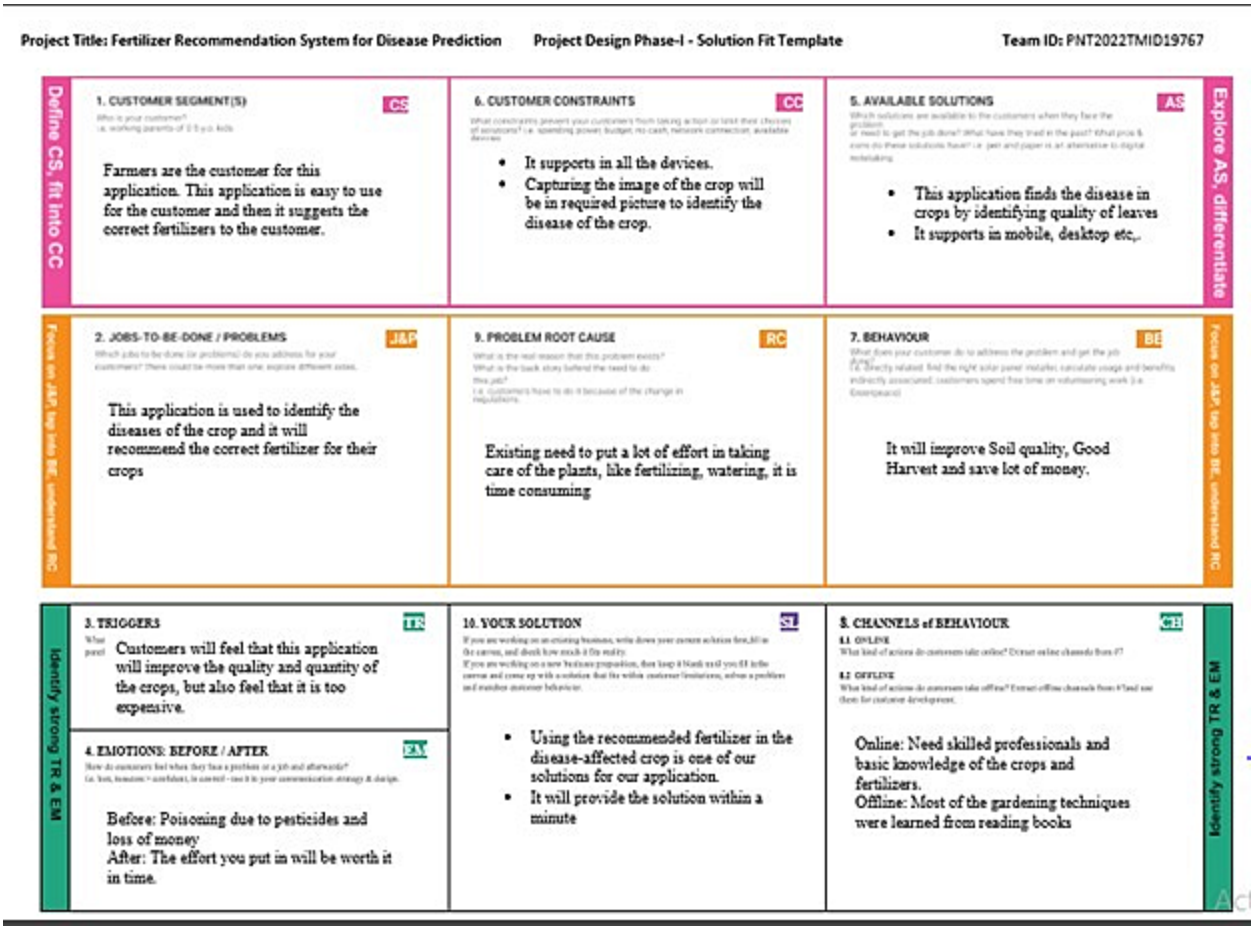
**Social Impact / Customer Satisfaction:** It identifies the diseases of the crops by using image processing and will provide the solution within a minute and also increases the quality and quantity of the crops.

**Business Model:** The application is recommended to farmers in subscription basis

**Scalability:** Application can be improved by introducing online purchases of crops,

fertilizers with door delivery and online payment

### 3.4 PROBLEM SOLUTION FIT




## 4. REQUIREMENT ANALYSIS

### 4.1 FUNCTIONAL REQUIREMENTS

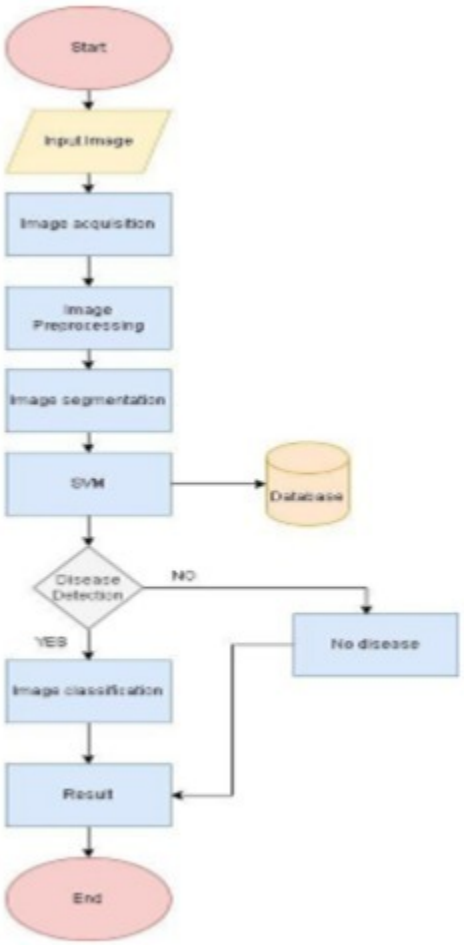
- User Registration
- User Confirmation
- User Details
- Uploading data
- Image processing
- Solution description

### 4.2 NON-FUNCTIONAL REQUIREMENTS

- Usability
- Security
- Performance
- Availability
- Scalability
- Reliability

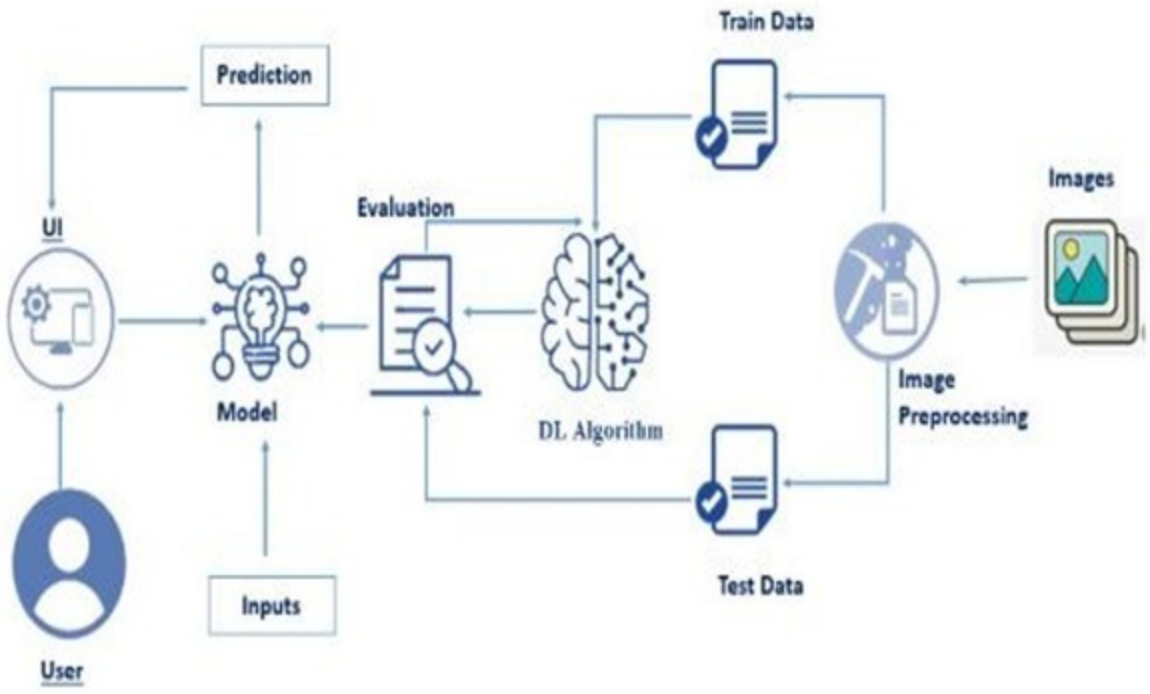
## 5. PROJECT DESIGN

### 5.1 DATA FLOW DIAGRAMS

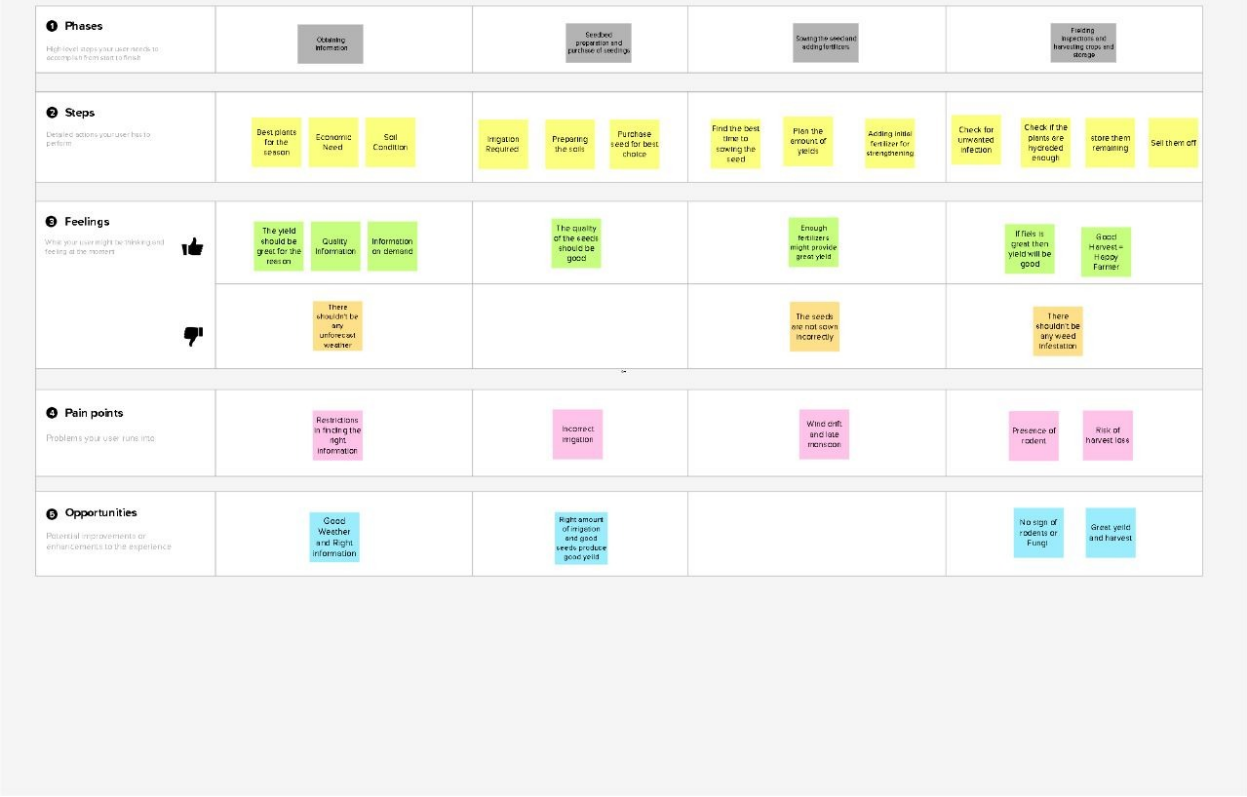



## 5.2 SOLUTION AND TECHNICAL ARCHITECTURE

We are planning to create an application which will be instructing the user to recommend the fertilizer for the prediction of diseases. To do this process without any errors, we will be training our model with more relatable training and testing dataset.




## 5.3 USER STORIES




# 6. SPRINT PLANNING AND SCHEDULING

## 6.1 SPRINT PLANNING AND ESTIMATION

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance 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 |
| | Login | USN-2 | As a user, I can log into the application by entering email & password | I can login using my E-mail ID accounts or user credentials | High | Sprint-1 |
| | Dashboard | USN-3 | As a user, I can view the page of the application where i can upload my images and the fertilizer should be recommended. | I can access my account/ dashboard. | High | Sprint-2 |
| Customer (Web user) | Registration | USN-4 | As a user, I can login to web dashboard just Like website dashboard | I can register using my username and password | High | Sprint-3 |
| | Login | USN-5 | As a user, I can login to my web dashboard with the login credentials | I can login using my User credentials | High | Sprint-3 |
| | Dashboard | USN-6 | As a user, I can view the web application where i can upload my images for getting the suggestion of the fertilizer | I can access my account/ dashboard | High | Sprint-4 |
| | | USN-7 | As a user, the fertilizer recommended to me Is in high accurate. | I can access my account/ dashboard | High | Sprint-4 |

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Administrator | Login | USN-8 | As a admin, I can login to the website using my login credentials. | I can login to the website using my login credentials. | High | Sprint-5 |
| | Dashboard | USN-9 | As a admin, I can view the dashboard of the application. | I can access my dashboard | High | Sprint-5 |

## 6.2 SPRINT DELIVERY SCHEDULE

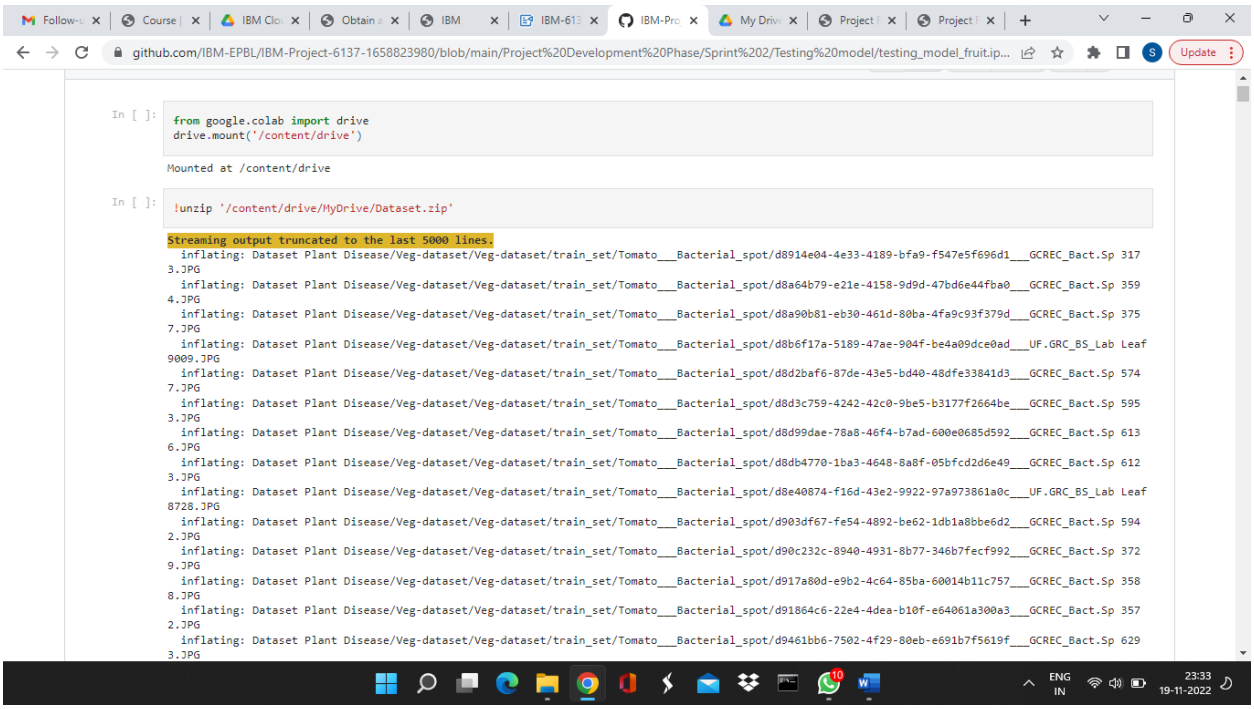
| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points (Total) | Priority | Team members |
|---|---|---|---|---|---|---|
| Sprint-1 | Model Creation and Training (Fruits) | | Create a model which can classify diseased fruit plants from given images. I also need to test the model and deploy it on IBM Cloud | 8 | High | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |
| | Model Creation and Training (Vegetables) | | Create a model which can classify diseased vegetable plants from given images | 2 | High | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points (Total) | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-2 | Model Creation and Training (Vegetables) | | Create a model which can classify diseased vegetable plants from given images and train on IBM Cloud | 6 | High | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |
| | Registration | USN-1 | As a user, I can register by entering my email, password, and confirming my password or via OAuth API | 3 | Medium | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |
| | Upload page | USN-2 | As a user, I will be redirected to a page where I can upload my pictures of crops | 4 | High | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |
| | Suggestion results | USN-3 | As a user, I can view the results and then obtain the suggestions provided by the ML model | 4 | High | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |
| | Base Flask App | | A base Flask web app must be created as an interface for the ML model | 2 | High | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |
| Sprint-3 | Login | USN-4 | As a user/admin/shopkeeper, I can log into the application by entering email & password | 2 | High | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |
| | User Dashboard | USN-5 | As a user, I can view the previous results and history | 3 | Medium | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |
| | Integration | | Integrate Flask, CNN model with Cloudant DB | 5 | Medium | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |
| | Containerization | | Containerize Flask app using Docker | 2 | Low | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |
| Sprint-4 | Dashboard (Admin) | USN-6 | As an admin, I can view other user details and uploads for other purposes | 2 | Medium | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |
| | Dashboard (Shopkeeper) | USN-7 | As a shopkeeper, I can enter fertilizer products and then update the details if any | 2 | Low | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |
| | Containerization | | Create and deploy Helm charts using Docker Image made before | 2 | Low | Swetha C<br>Nalina M<br>Subanandhana R<br>Renuka R |

| 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 | 10 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 10 | 30 Oct 2022 |
| Sprint-2 | 15 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 15 | 06 Nov 2022 |
| Sprint-3 | 15 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 15 | 13 Nov 2022 |
| Sprint-4 | 12 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 10 | 20 Nov 2022 |

# 7. CODING AND SOLUTIONING

## 7.1 FEATURE 1



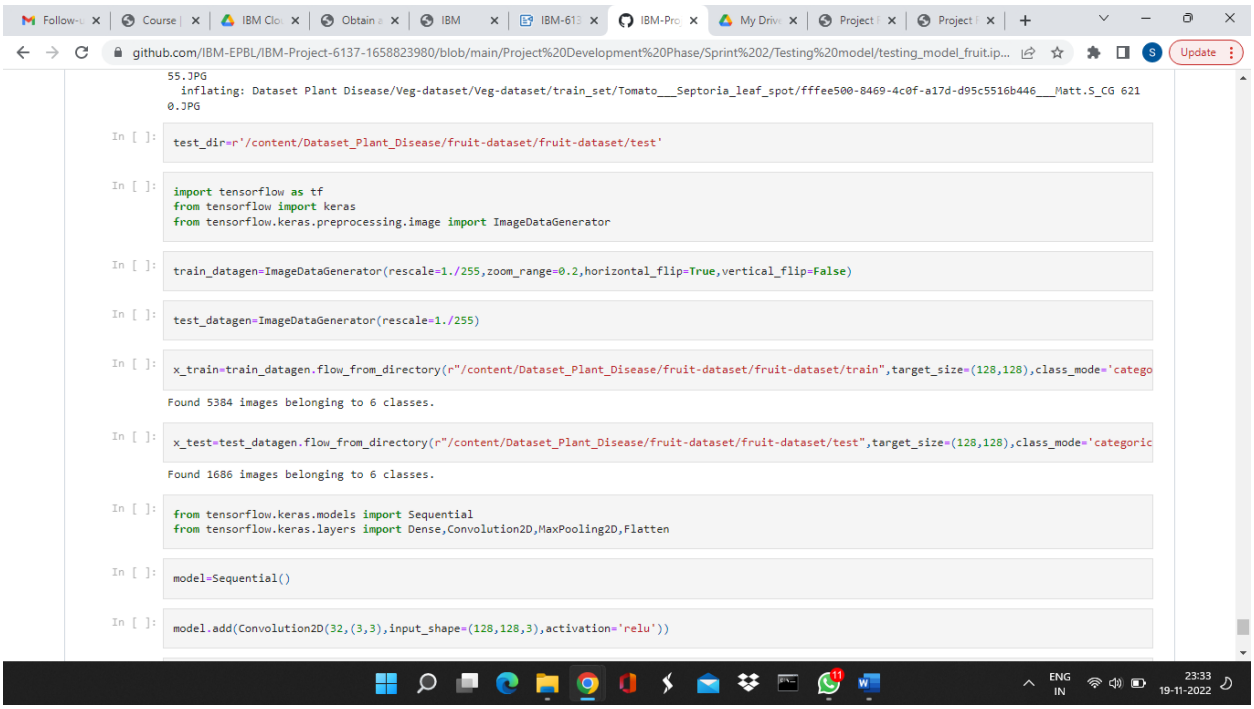
github.com/IBM-EPBL/IBM-Project-6137-1658823980/blob/main/Project%20Development%20Phase/Sprint%202/Testing%20model/testing_model_fruit.ip...

In [ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [ ]:
```
!unzip '/content/drive/MyDrive/Dataset.zip'
```

```
Streaming output truncated to the last 5000 lines.
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d8914e04-4e33-4189-bfa9-f547e5f696d1___GCREC_Bact.Sp 317
3.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d8a64b79-e21e-4158-9d9d-47bd6e44fba0___GCREC_Bact.Sp 359
4.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d8a90b81-eb30-461d-80ba-4fa9c93f379d___GCREC_Bact.Sp 375
7.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d8b6f17a-5189-47ae-904f-be4a09dce0ad___UF.GRC_BS_Lab Leaf
9009.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d8d2baf6-87de-43e5-bd40-48dfe33841d3___GCREC_Bact.Sp 574
7.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d8d3c759-4242-42c0-9be5-b3177f2664be___GCREC_Bact.Sp 595
3.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d8d99dae-78a8-46f4-b7ad-600e0685d592___GCREC_Bact.Sp 613
6.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d8db4770-1ba3-4648-8a8f-05bfcd2d6e49___GCREC_Bact.Sp 612
3.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d8e40874-f16d-43e2-9922-97a973861a0c___UF.GRC_BS_Lab Leaf
8728.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d903df67-fe54-4892-be62-1db1a8bbe6d2___GCREC_Bact.Sp 594
2.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d90c232c-8940-4931-8b77-346b7fecf992___GCREC_Bact.Sp 372
9.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d917a80d-e9b2-4c64-85ba-60014b11c757___GCREC_Bact.Sp 358
8.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d91864c6-22e4-4dea-b10f-e64061a300a3___GCREC_Bact.Sp 357
2.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Bacterial_spot/d9461bb6-7502-4f29-80eb-e691b7f5619f___GCREC_Bact.Sp 629
3.JPG
```



github.com/IBM-EPBL/IBM-Project-6137-1658823980/blob/main/Project%20Development%20Phase/Sprint%202/Testing%20model/testing_model_fruit.ip...

```
55.JPG
  inflating: Dataset Plant Disease/Veg-dataset/Veg-dataset/train_set/Tomato___Septoria_leaf_spot/fffee500-8469-4c0f-a17d-d95c5516b446___Matt.S_CG 621
0.JPG
```

In [ ]:
```
test_dir=r'/content/Dataset_Plant_Disease/fruit-dataset/fruit-dataset/test'
```

In [ ]:
```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [ ]:
```
train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
```

In [ ]:
```
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [ ]:
```
x_train=train_datagen.flow_from_directory(r"/content/Dataset_Plant_Disease/fruit-dataset/fruit-dataset/train",target_size=(128,128),class_mode='catego
```

```
Found 5384 images belonging to 6 classes.
```

In [ ]:
```
x_test=test_datagen.flow_from_directory(r"/content/Dataset_Plant_Disease/fruit-dataset/fruit-dataset/test",target_size=(128,128),class_mode='categoric
```

```
Found 1686 images belonging to 6 classes.
```

In [ ]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Convolution2D,MaxPooling2D,Flatten
```

In [ ]:
```
model=Sequential()
```

In [ ]:
```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
```

```
In [ ]: model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

In [ ]: model.add(MaxPooling2D(pool_size=(2,2)))
        model.add(Flatten())
        model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 flatten (Flatten)           (None, 127008)            0

=================================================================
Total params: 896
Trainable params: 896
Non-trainable params: 0
_________________________________________________________________
```

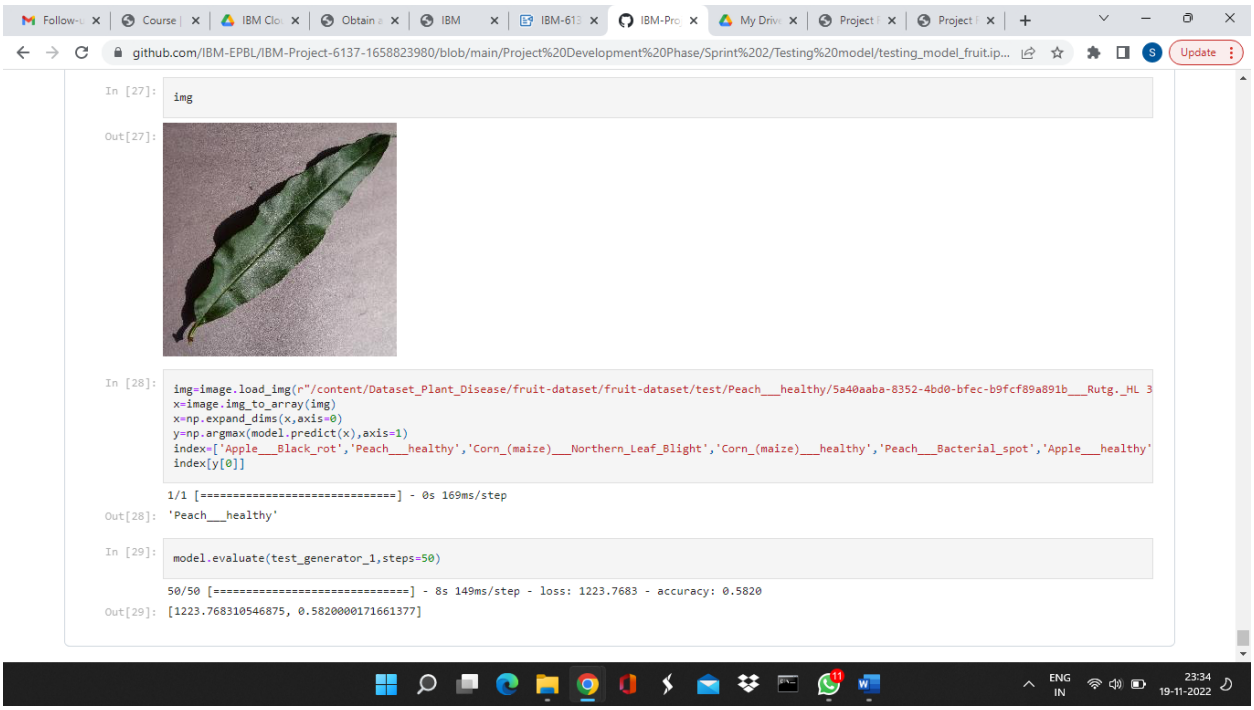
```
In [ ]: 32*(3*3*3+1)
        model.add(Dense(300,activation='relu'))
        model.add(Dense(150,activation='relu'))

In [ ]: model.add(Dense(6,activation='softmax'))
        model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
        len(x_train)

Out[ ]: 225

In [20]: model.fit(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs=10)
```


```
225/225 [==============================] - 164s 727ms/step - loss: 0.0940 - accuracy: 0.9716 - val_loss: 0.0990 - val_accuracy: 0.9656
Epoch 8/10
225/225 [==============================] - 163s 725ms/step - loss: 0.0523 - accuracy: 0.9811 - val_loss: 0.1073 - val_accuracy: 0.9686
Epoch 9/10
225/225 [==============================] - 161s 715ms/step - loss: 0.0499 - accuracy: 0.9807 - val_loss: 0.1402 - val_accuracy: 0.9680
Epoch 10/10
225/225 [==============================] - 164s 727ms/step - loss: 0.0653 - accuracy: 0.9801 - val_loss: 0.1073 - val_accuracy: 0.9680
```

```
Out[20]:

In [21]: model.save('fruitdata.h5')

In [22]: import numpy as np
         from tensorflow.keras.models import load_model
         from tensorflow.keras.preprocessing import image

In [23]: model=load_model('fruitdata.h5')

In [24]: 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'
         )
```

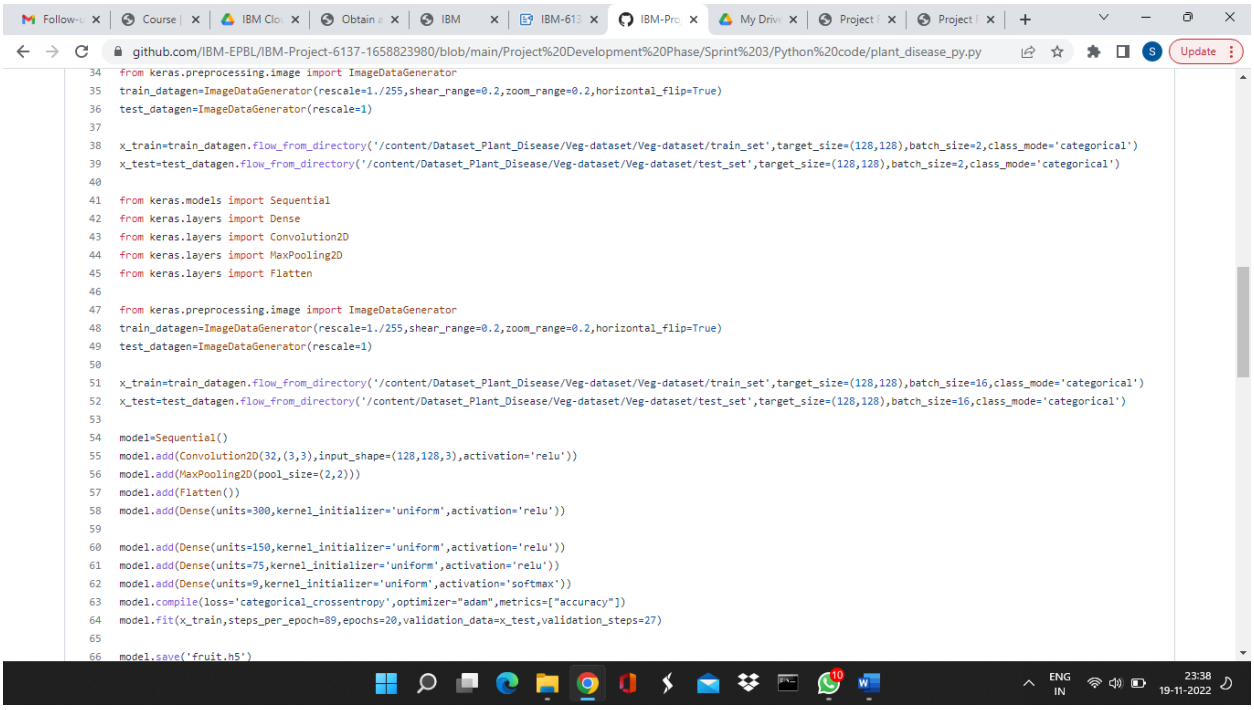
```
Found 1686 images belonging to 6 classes.
```

```
In [25]: import numpy as np
         from tensorflow.keras.models import load_model
         from tensorflow.keras.preprocessing import image

In [26]: img=image.load_img(r"/content/Dataset_Plant_Disease/fruit-dataset/fruit-dataset/test/Peach___healthy/5a40aaba-8352-4bd0-bfec-b9fcf89a891b___Rutg._HL 3
```



```
In [27]: img

Out[27]: 

In [28]: img=image.load_img(r"/content/Dataset_Plant_Disease/fruit-dataset/fruit-dataset/test/Peach___healthy/5a40aaba-8352-4bd0-bfec-b9fcf89a891b___Rutg._HL 3
         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','Peach___healthy','Corn_(maize)___Northern_Leaf_Blight','Corn_(maize)___healthy','Peach___Bacterial_spot','Apple___healthy'
         index[y[0]]

         1/1 [==============================] - 0s 169ms/step
Out[28]: 'Peach___healthy'

In [29]: model.evaluate(test_generator_1,steps=50)

         50/50 [==============================] - 8s 149ms/step - loss: 1223.7683 - accuracy: 0.5820
Out[29]: [1223.768310546875, 0.5820000171661377]
```

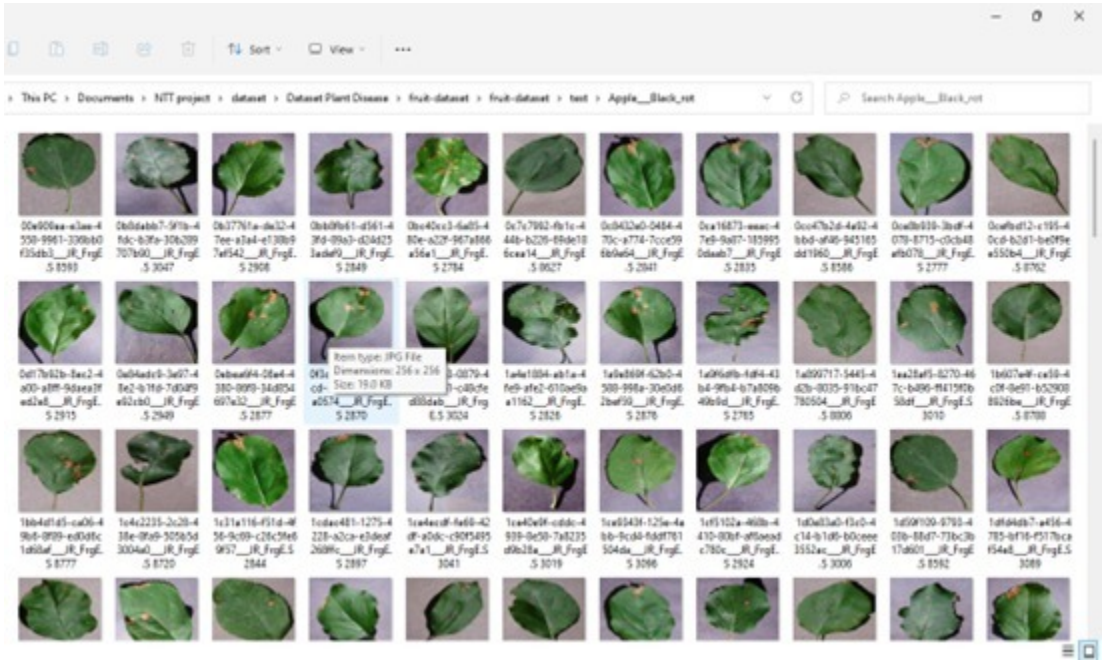
## 7.2 FEATURE 2



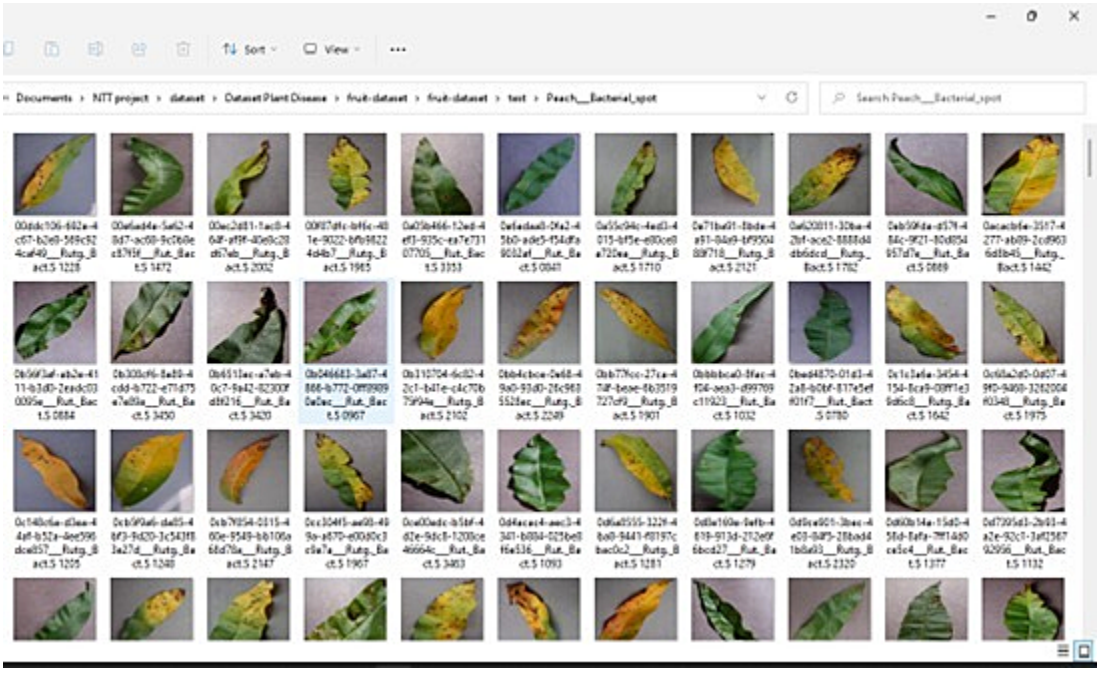
```
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1)

x_train=train_datagen.flow_from_directory('/content/Dataset_Plant_Disease/Veg-dataset/Veg-dataset/train_set',target_size=(128,128),batch_size=2,class_mode='categorical')
x_test=test_datagen.flow_from_directory('/content/Dataset_Plant_Disease/Veg-dataset/Veg-dataset/test_set',target_size=(128,128),batch_size=2,class_mode='categorical')

from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten

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('/content/Dataset_Plant_Disease/Veg-dataset/Veg-dataset/train_set',target_size=(128,128),batch_size=16,class_mode='categorical')
x_test=test_datagen.flow_from_directory('/content/Dataset_Plant_Disease/Veg-dataset/Veg-dataset/test_set',target_size=(128,128),batch_size=16,class_mode='categorical')

model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(units=300,kernel_initializer='uniform',activation='relu'))

model.add(Dense(units=150,kernel_initializer='uniform',activation='relu'))
model.add(Dense(units=75,kernel_initializer='uniform',activation='relu'))
model.add(Dense(units=9,kernel_initializer='uniform',activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer="adam",metrics=["accuracy"])
model.fit(x_train,steps_per_epoch=89,epochs=20,validation_data=x_test,validation_steps=27)

model.save('fruit.h5')
```

# 8. TESTING

## 8.1 TESTCASES

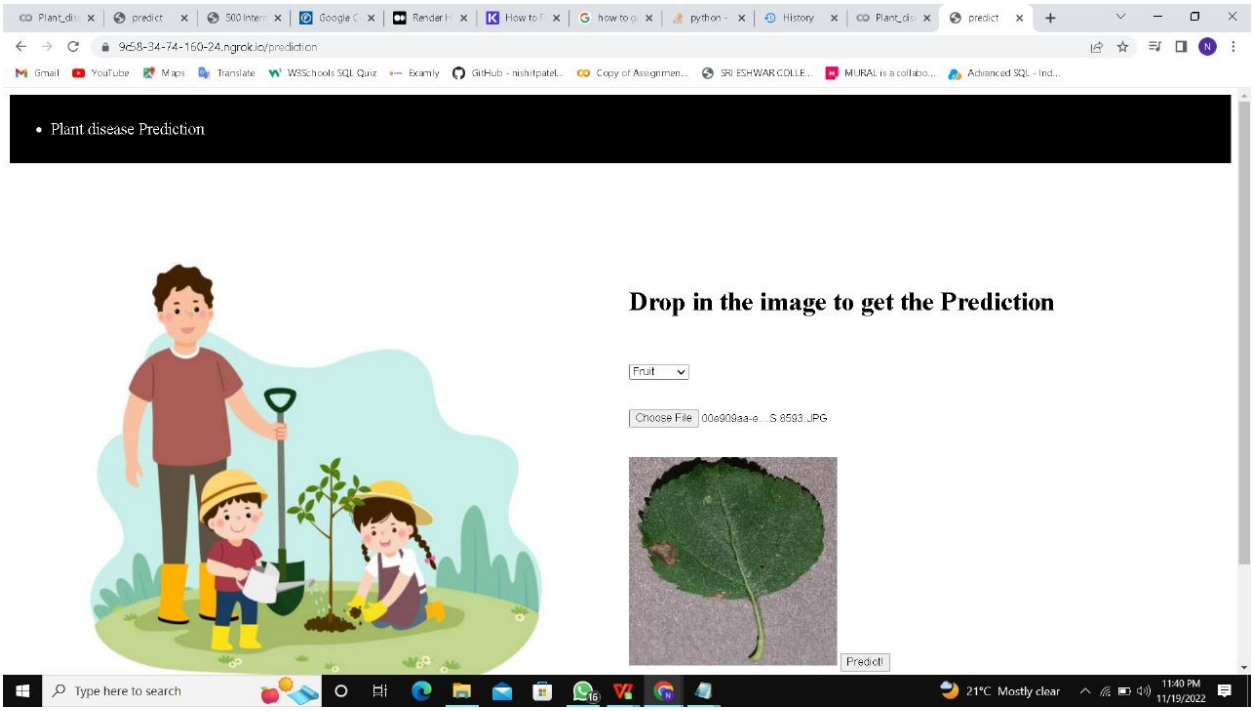



## 8.2 USER ACCEPTANCE TESTING



## 9. RESULTS


Plant Disease Prediction
Home
Predict
Detect if your plant is infected!!
Agriculture is one of the major sectors world wide. Over the years it has developed and the use of new technologies and equipment replaced almost all the tracitional methy of framing. The plant disesi effect the production. Identification of diseases and taking necessary prey is all done through naked eye, which requires labour and laboratories. This application kelps farmers in detecting the diseases by observing the spots on the leaves, which inturn saves effort and labor costs.




## 10. ADVANTAGES

- They are quick in providing plant nutrients and restoring soil fertility.
- They are portable and easy to transport.
- Plants easily absorb fertilizers.
- Fertilizers improve and increase the productivity of many crops such as wheat, maize, and rice.

## 11. CONCLUSION

Generally chemical fertilizers and pesticides are effective and convenient in use for production and disease management of plants but they are potential threat for the health and environment of soil, plant as well as humans.

## 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 Open CV

python library. This project work can  be extended for security applications such as figure print recognition,  iris recognition and face recognition.

## 13. APPENDIX

### GITHUB link

**https://github.com/IBM-EPBL/IBM-Project-6137-1658823980**