# Fertilizers Recommendation System For Disease Prediction

APROJECTREPORT

***Submittedby***

|  |  |
| --- | --- |
| **M.MASANAMUTHU** | **(952319106019)** |
| **A.CHINNA DURAI** | **(952319106006)** |
| **M.SATHISHKUMAR** | **(952319106031)** |
| **L.ESAKKI RAJ** | **(952319106010)** |

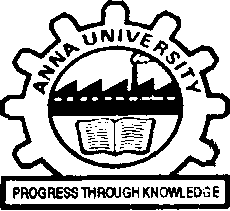
## M.ARUMAI KODI(952319106003)

***In partial fulfillment for the award of the degreeOf***

## BACHELOROFENGINEERING

***In***

ELECTRONICSANDCOMMUNICATIONENGINEERING



PSN ENGINEERING COLLEGE, MELATHEDIYOORTIRUNELVELI-627152

## AFFILIATED TOANNAUNIVERSITY-CHENNAI600025

NOVEMBER2022

# BONAFIDECERTIFICATE

Certified that this report titled “**FERTILIZERS REOMMENTATION SYSTEM FOR DISEASE PREDICTION”**isthebonafideworkof**“M.MASANA MUTHU(952319106019),A.CHINNA DURAI(952319106006),M.SATHISHKUMAR(952319106031),L.ESAKKIRAJ(952319106010),M.ARUMAIKODI(952319106003)”**, who carried out the project workundermysupervision.

SIGNATUREMr.S.PITCHAIAHM.E.,

HEADOFTHEDEPARTMENT,

Department of ECE,Melathediyoor,Tirunelveli-627152

SIGNATUREMr.S.PITCHAIAHM.E.,IBMSPOC,

Department of ECE,Melathediyoor,Tirunelveli-627152

SubmittedfortheProjectworkandViva-Voceexaminationheldon…………

MENTOR EVALUVATOR

**INTRODUCTION**

**1.1 Project Overview**

In today’s world agriculture is very important for life and helps to save the natural resources around as. Doing agriculture is the very hard in current scenario because of many natural disasters are happening every day. Most of the plants are affected by many diseases due to pollution in water, air, soil.

Identifying the disease is one of the huge hurtles in agriculture. Most of the plants are affected by leaf disease and it’s hard to find to correct fertilizer to cure. The main objective of this project is to identify the disease in the plants and cure it in the early stage of the infection. 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**

This project is used to test the fruits and vegetables samples and identify the different diseases. Also, this project recommends fertilizers for prediced diseases

**LITERATURE SURVEY**

**2.1 Existing Problem**

Indumathi proposed a method for leaf disease detection and suggest

fertilizers to cure leaf diseases[1]. But the method involves less number of train and

test sets which results in poor accuracy. Pandi selvi [2] proposed a simple prediction method for soil based fertilizer recommedation system forpredicted crop diseases. This method gives less accuracy and prediction. Shiva reddy [3] proposed an IoT based system for leaf disease detection and fertilizer recommendation which is based on Machine Learning techniques yields less 80 percentage accuracies.

**2.2 Referances**

[1]. R Indumathi.; N Saagari.; V Thejuswini.; R Swarnareka.," Leaf Disease Detection and Fertilizer Suggestion", IEEE International Conference on System, Computation, 9 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 (IRJET), Volume: 06 Issue: 11, Nov 2019, e-ISSN: 2395-

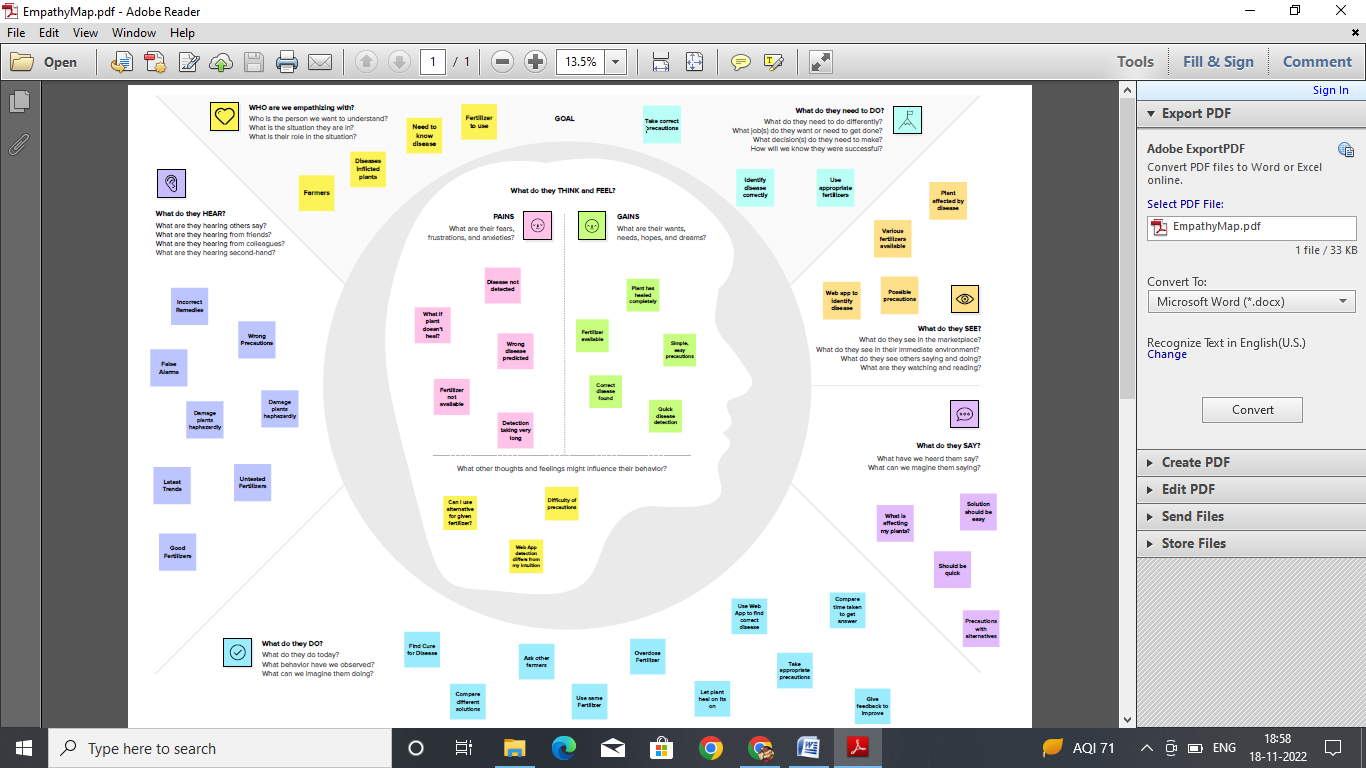
0056.

**2.3 Problem Statement Definition**

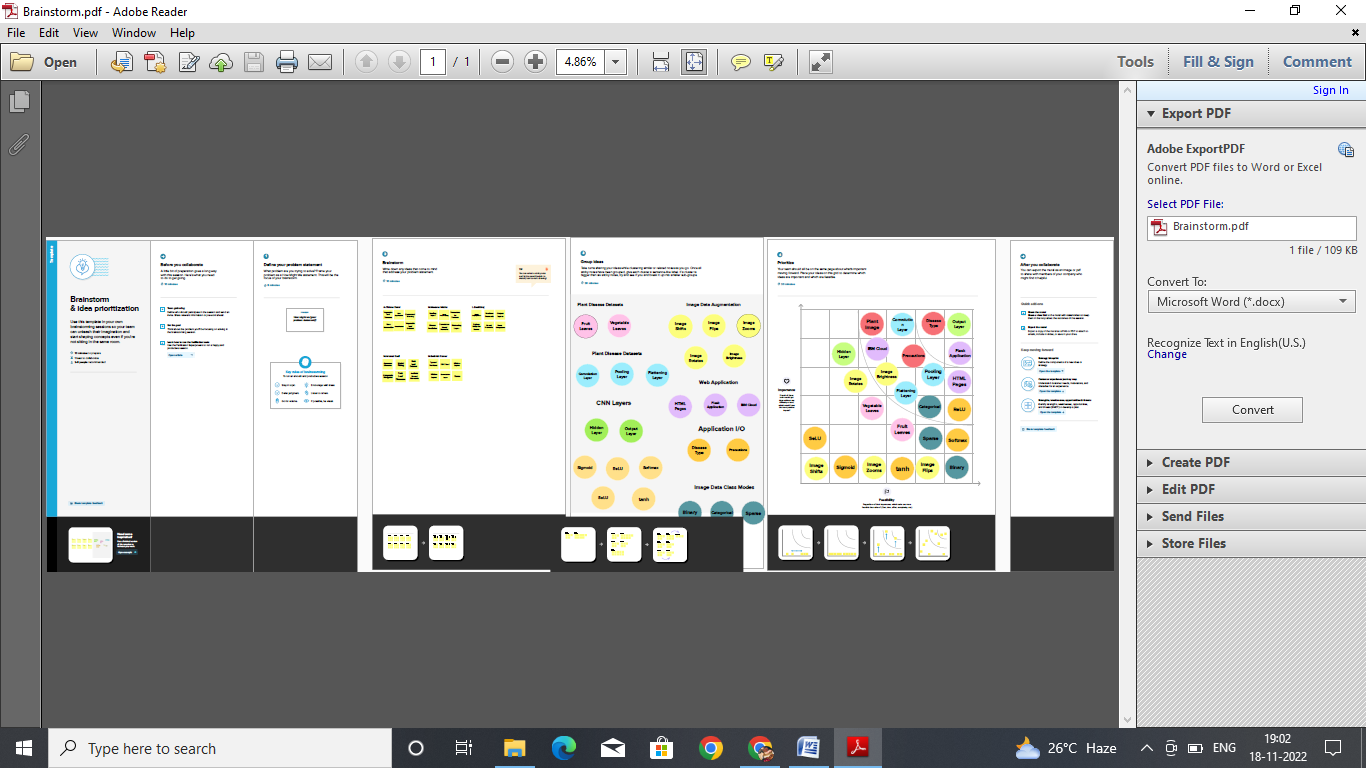
In India, The Agriculture industry is extremely vital and crucial for economic and social development and jobs. In India, the agricultural sector provides a living for almost 48% of the population. As per the 2019-2020 economic survey, an Indian farmer's median wage in 16 states is Rupees 2500. Most 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**.**

**IDEATION & PROPOSED SOLUTION**

**3.1 Empathy Map Canvas**

****

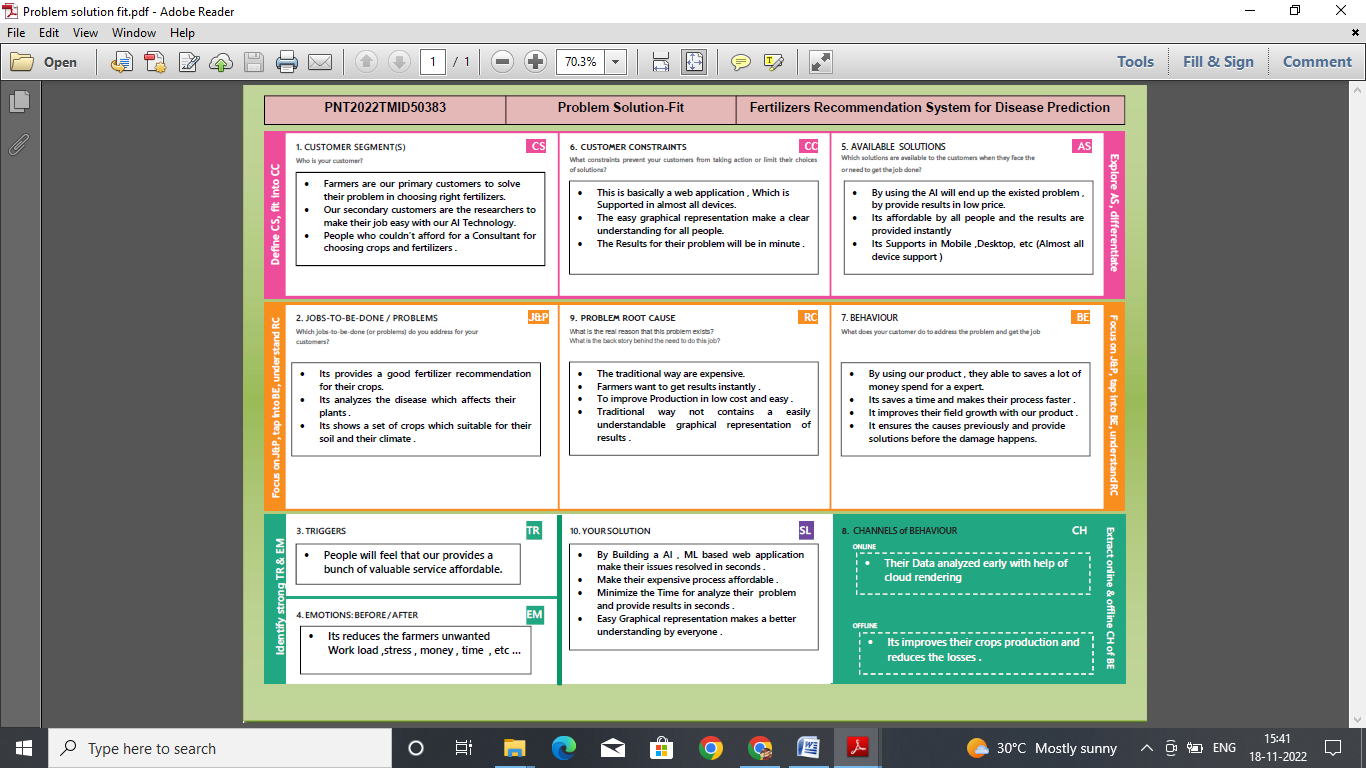
**3.2 Ideation & Brainstroming**



**3.3 Proposed Solution**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | |  | | --- | | **Parameter** | | |  | | --- | | **Description** | |
| **1** | |  | | --- | | Problem Statement (Problem to be solved) | | |  | | --- | | The problem statement is that when a crop’s leaf image is given as the input to an AI model, the essential features from the leaves are taken, analyzed and the AI model will predict the disease and will suggest a suitable fertilizer to cure the disease that the crop has been infected with. | |
| **2** | |  | | --- | | Idea / Solution description | | |  | | --- | | In order to predict the disease and to suggest a suitable fertilizer to cure the disease that the crop has been infected with, an artificial intelligent system has to be introduced to provide farmers with the best solution possible. | |
| **3** | |  | | --- | | Novelty / Uniqueness | | |  | | --- | | AI model developed will be able to predict the disease accurately and able to suggest suitable fertilizer for the disease that the crop has been infected with while the input is only the image of the leaf of the infected crop. | |
| **4** | |  | | --- | | Social Impact / Customer Satisfaction | | |  | | --- | | The AI model is built in a way such that each farmer can get benefitted and fully satisfied in terms of production as well quality of the goods produced without spending huge amount of money. | |
| **5** | |  | | --- | | Business Model (Revenue Model) | | |  | | --- | | Employing an AI model will be a cost-effective solution for agriculture. It eliminates the need for soil testing and the results are provided instantly and much faster than conventional methods for crop disease prediction. | |
| **6** | |  | | --- | | Scalability of the Solution | | |  | | --- | | The AI model developed can be scaled to predict the source of the disease when the crops leaf image is given as an input and also can be scaled to suggest suitable relevant diseases that the crop might get infected in the future as well. | |

**3.4 Problem Solution fit**

****

**REQUIREMENT ANALYSIS**

**4.1 Functional requirement**

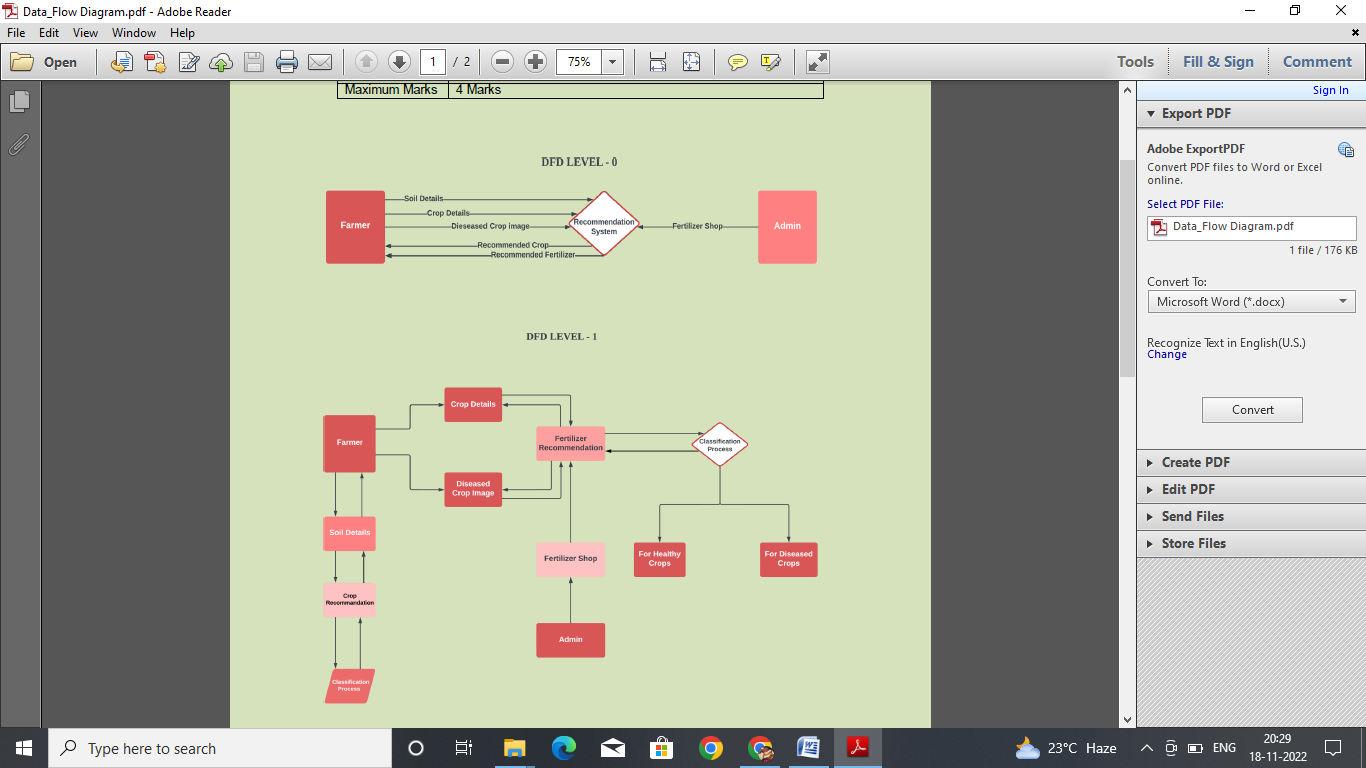
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **FR**  **No** | | |  | | --- | | **Functional Requirement (Epic)** | | |  | | --- | | **Sub Requirement (Story / Sub-Task)** | |
| |  | | --- | | **FR-1** | | |  | | --- | | User Registration | | |  | | --- | | **Registration through Form** | |
| |  | | --- | | **FR-2** | | |  | | --- | | User Confirmation | | |  | | --- | | **Confirmation via Email** | |
| |  | | --- | | **FR-3** | | |  | | --- | | User Profile | | |  | | --- | | **Filling the profile page after logging in** | |
| |  | | --- | | **FR-4** | | |  | | --- | | Uploading Dataset (Leaf) | | |  | | --- | | **Images of the leaves are to be uploaded** | |
| |  | | --- | | **FR-5** | | |  | | --- | | Requesting solution | | |  | | --- | | **Uploaded images is compared with the pre-defined Model and solution is generated** | |
| |  | | --- | | **FR-6** | | |  | | --- | | Downloading Solution | | |  | | --- | | **The Solution in pdf format which contains the recommendations of fertilizers and the possible diseases.** | |

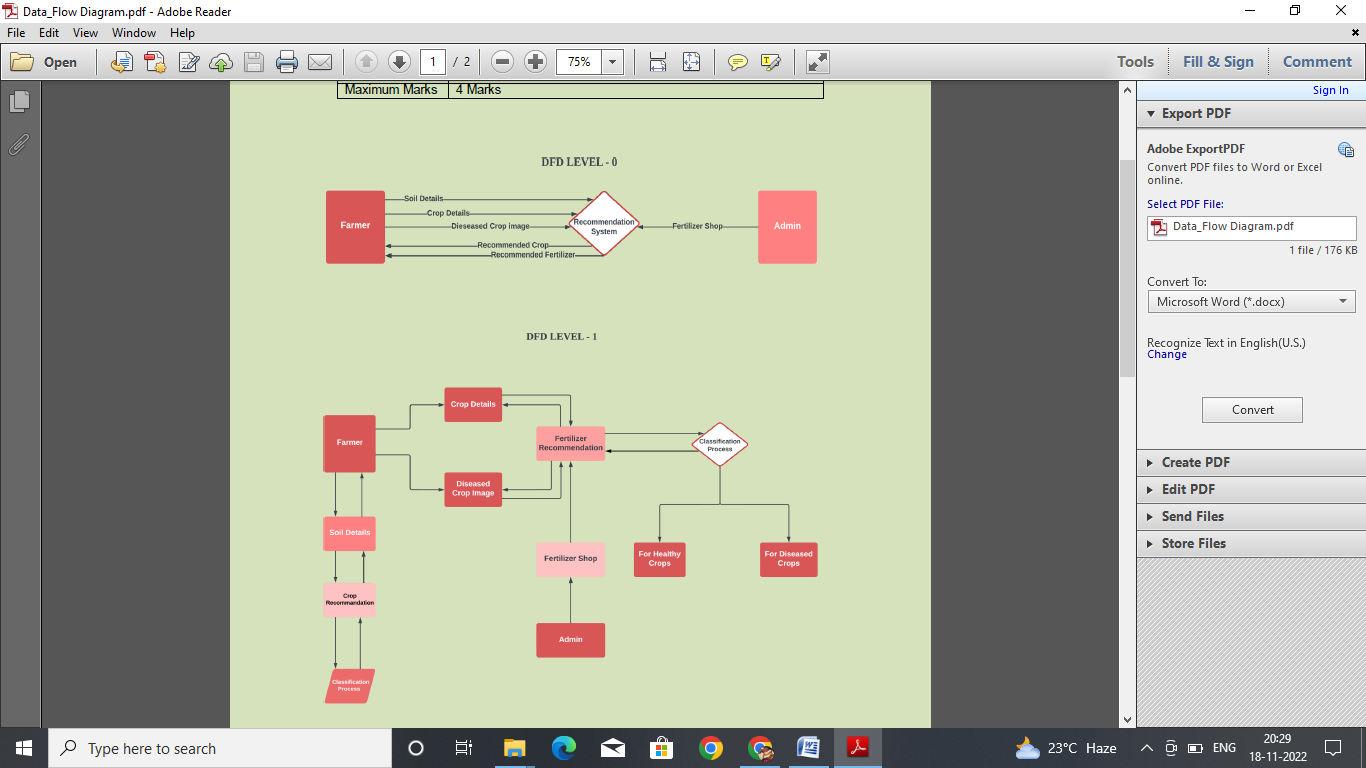
**4.2 Non-Functional requirements**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **FR**  **No.** | | |  | | --- | | **Non-Functional Requirement** | | |  | | --- | | **Description** | |
| |  | | --- | | **NFR-1** | | |  | | --- | | **Usability** | | |  | | --- | | **The system allows the user to perform the tasks easily and efficiently and effectively.** | |
| **NFR-2** | |  | | --- | | **Security** | | |  | | --- | | **Assuring all data inside the system or its part will be**  **protected against malware attacks or unauthorized access.** | |
| **NFR-3** | |  | | --- | | **Reliability** | | |  | | --- | | **The website does not recover from failure quickly**  **,it**  **takes time as the application is running in single server** | |
| **NFR-4** | |  | | --- | | **Performance** | | |  | | --- | | **Response Time and Net Processing Time is Fast** | |
| **NFR-5** | |  | | --- | | **Availability** | | |  | | --- | | **The system will be available up to 95% of the time** | |
| **NFR-6** | |  | | --- | | **Scalability** | | |  | | --- | | **The website is scalable** | |

**PROJECT DESIGN**

**5.1 Data Flow Diagrams**

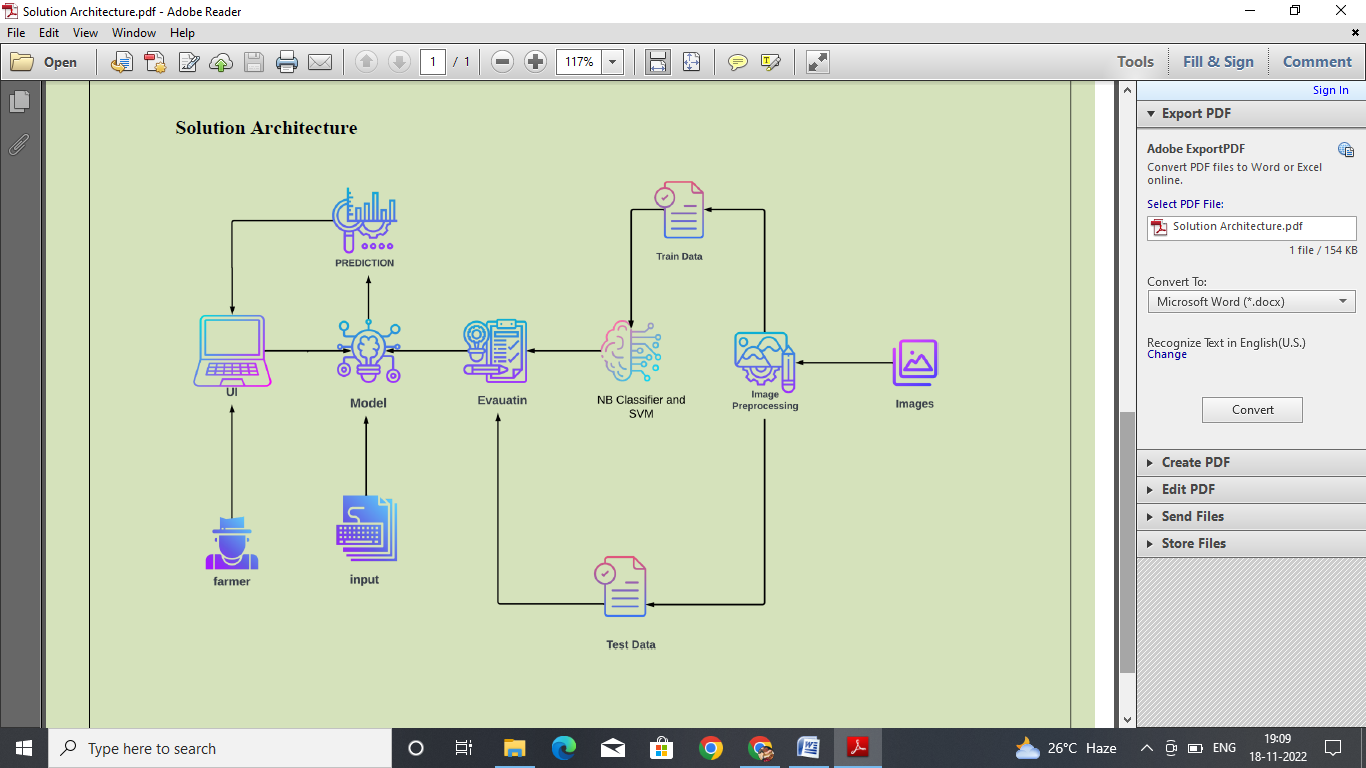
****



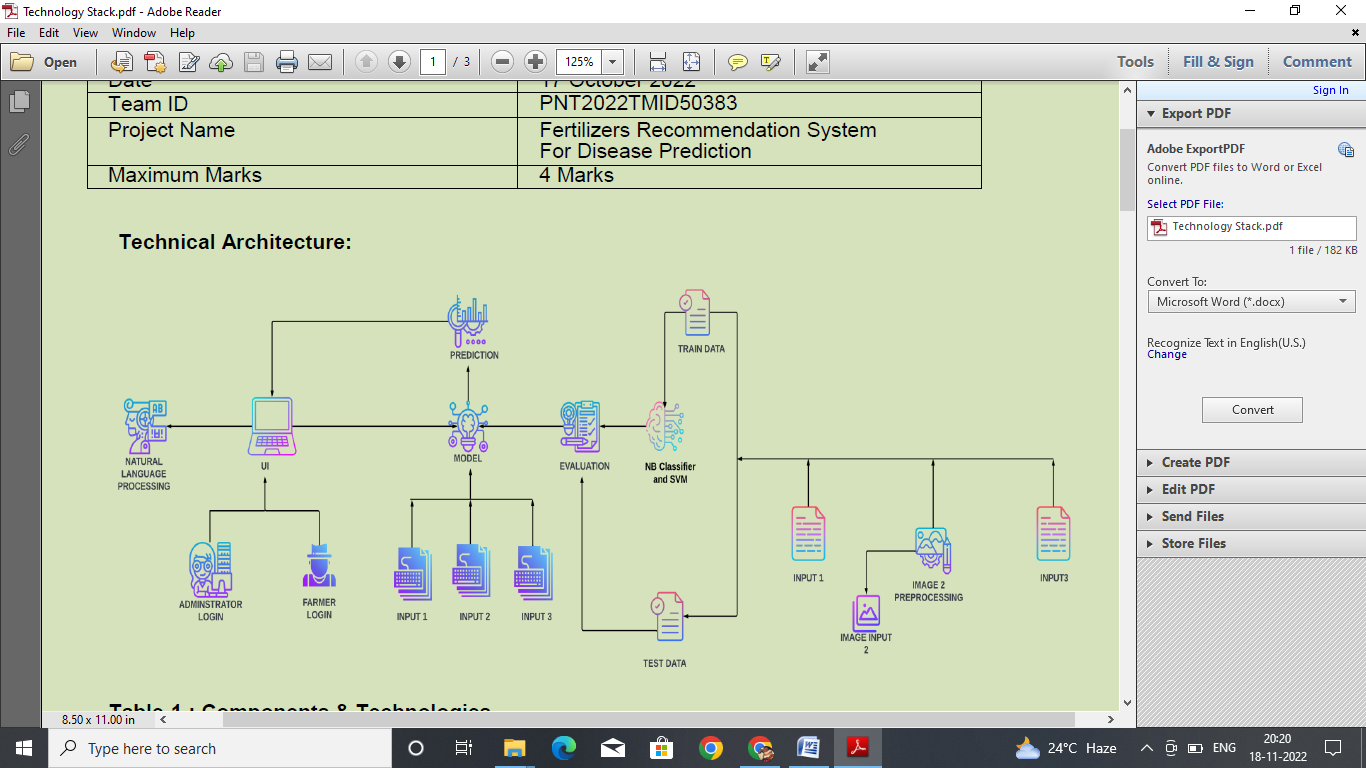
**5.2 Solution & Technical Architecture**

Crop disease in plants is predicted and suitable fertilizer is recommended for better yield. The images of the diseased plants are obtained and it is preprocessed against the dataset of diseased plants. Deep Learning Algorithm is used to process the images and then it is evaluated. Then a model is built on the evaluations, it is then trained using no. of. inputs and predictions are given to the users which subsequently helps in recommending the fertilizers. The Convolutional layers are used to classify and process the images and further helps in recommending the fertilizers. The image classification steps are: Image acquisition Preprocessing Segmentation Disease Prediction Fertilizer Recommendation

**Solution Architecture**

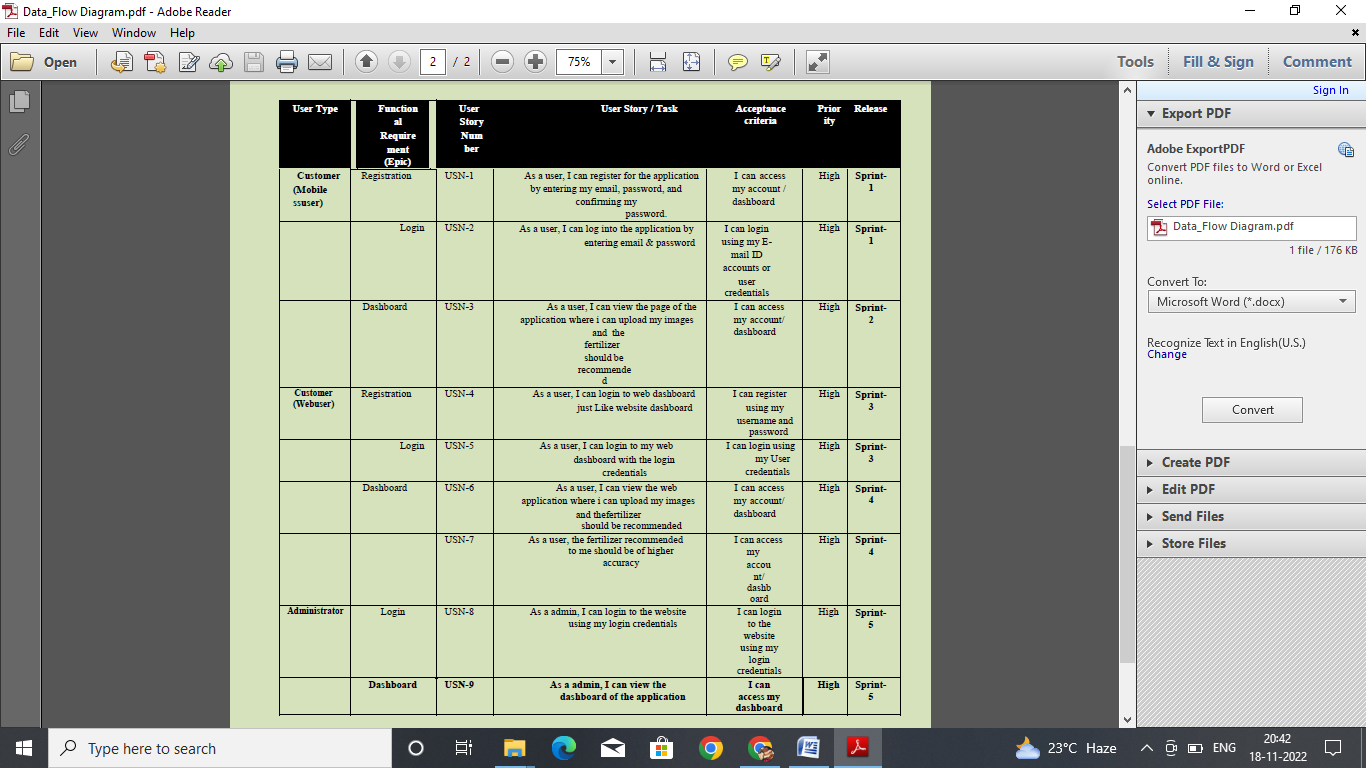
****

**Technical Architecture**

****

**5.3 User Stories**

**Use the below template to list all the user stories for the product.**

****

**PROJECT PLANNING & SCHEDULING**

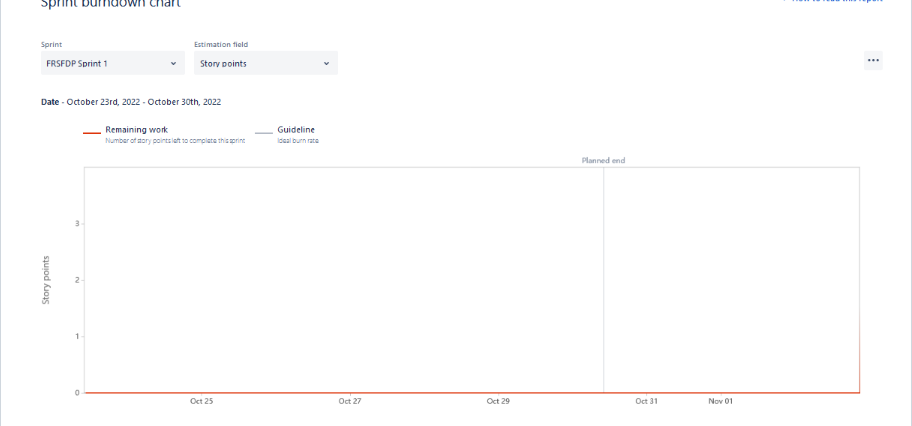
**6.1 Sprint Planning & Estimation**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **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** | | |  | | --- | | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** | |
|  | |  | | --- | | **Model Creation and Training (Vegetables)** | |  | |  | | --- | | **Create a model which can classify diseased vegetable plants from given images** | | |  | | --- | | **2** | | |  | | --- | | **High** | | |  | | --- | | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** | |
| **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** | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** |
|  | **Registration** | **USN-1** | **As a user, I can register by entering my email, password, and confirming my password or via OAuth API** | **3** | **Medium** | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** |
|  | **Upload page** | **USN-2** | **As a user, I will be redirected to a page where I can upload my pictures of crops** | **4** | **High** | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** |
|  | **Suggestion results** | **USN-3** | **As a user, I can view the results and then obtain the suggestions provided by the ML model** | **4** | **High** | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** |
|  | **Base Flask App** |  | **A base Flask web app must be created as an interface for the ML model** | **2** | **High** | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** |
| **Sprint-3** | **Login** | **USN-4** | **As a user/admin/shopkeeper, I can log into the application by entering email & password** | **2** | **High** | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** |
|  | **User Dashboard** | **USN-5** | **As a user, I can view the previous results and history** | **3** | **Medium** | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** |
|  | **Integration** |  | **Integrate Flask, CNN model with Cloudant DB** | **5** | **Medium** | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** |
|  | **Containerization** |  | **Containerize Flask app using Docker** | **2** | **Low** | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** |
| **Sprint-4** | **Dashboard (Admin)** | **USN-6** | **As an admin, I can view other user details and uploads for other purposes** | **2** | **Medium** | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** |
|  | **Dashboard (Shopkeeper)** | **USN-7** | **As a shopkeeper, I can enter fertilizer products and then update the details if any** | **2** | **Low** | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** |
|  | **Containerization** |  | **Create and deploy Helm charts using Docker Image made before** | **2** | **Low** | **Chinna Durai,Masana Muthu,Sathish Kumar,Arumai Kodi, Esakkiraj** |

**6.2 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 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | | 20 | 6 Days | | 31 Oct 2022 | 05 Nov 2022 | 20 | 04 Nov 2022 |
| Sprint-3 | | 20 | 6 Days | | 07 Nov 2022 | 12 Nov 2022 | 20 | 10 Nov 2022 |
| Sprint-4 | 20 | | | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 17 Nov 2022 |

**6.3 Reports from JIRA**



**CODING & SOLUTIONING (Explain the features added in the project along with code)**

**7.1 Feature 1**

**# -\*- coding: utf-8 -\*-**

**"""Copy of Test the Veg model.ipynb**

**Automatically generated by Colaboratory.**

**Original file is located at**

**https://colab.research.google.com/drive/1RHpmLZRIo1sq5mAhS8EUL\_PAcVbNWolZ**

**"""**

**!unzip '/content/drive/MyDrive/ibm dataset/Fertilizers\_Recommendation\_ System\_For\_Disease\_ Prediction.zip'**

**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')**

**model.summary()**

**from keras.preprocessing import image**

**from tensorflow.keras.preprocessing.image import img\_to\_array**

**from tensorflow.keras.preprocessing import image**

**from tensorflow.keras.models import load\_model**

**import numpy as nps**

**model=load\_model('fruit.h5')**

**img=image.load\_img('/content/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple\_\_\_healthy/011d02f3-5c3c-4484-a384-b1a0a0dbdec1\_\_\_RS\_HL 7544.JPG',grayscale=False,target\_size=(128,128))**

**img**

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

**x=nps.expand\_dims(x,axis=0)**

**pred=(model.predict(x) > 0.5).astype("int32")**

**pred**

**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\_\_)**

**model = load\_model("fruit.h5")**

**@app.route('/')**

**def home():**

**return render\_template('home.html')**

**@app.route('/prediction')**

**def prediction():**

**return render\_template('predict.html')**

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

**def predict():**

**if request.method=='POST':**

**f= request.files['images']**

**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=="fruit"):**

**preds=model.predict\_classess(x)**

**print(preds)**

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

**print (df.iloc[preds[0]]['cautions'])**

**else:**

**pred=model1.predict\_classes(x)**

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

**print(df.iloc[preds[0]]['caution'])**

**return df.iloc[preds[0]]['caution']**

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

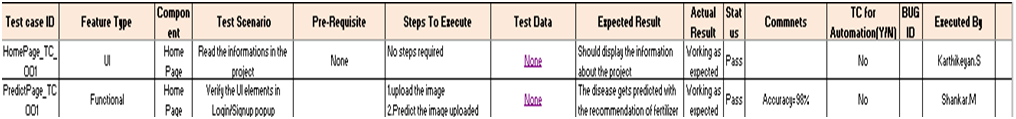
**app.run(debug=False)**

**7.2 Feature 2**

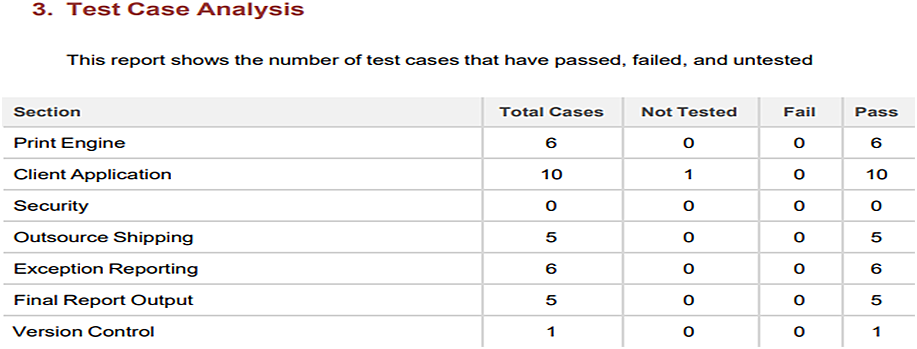
**7.3 Database Schema (if Applicable)**

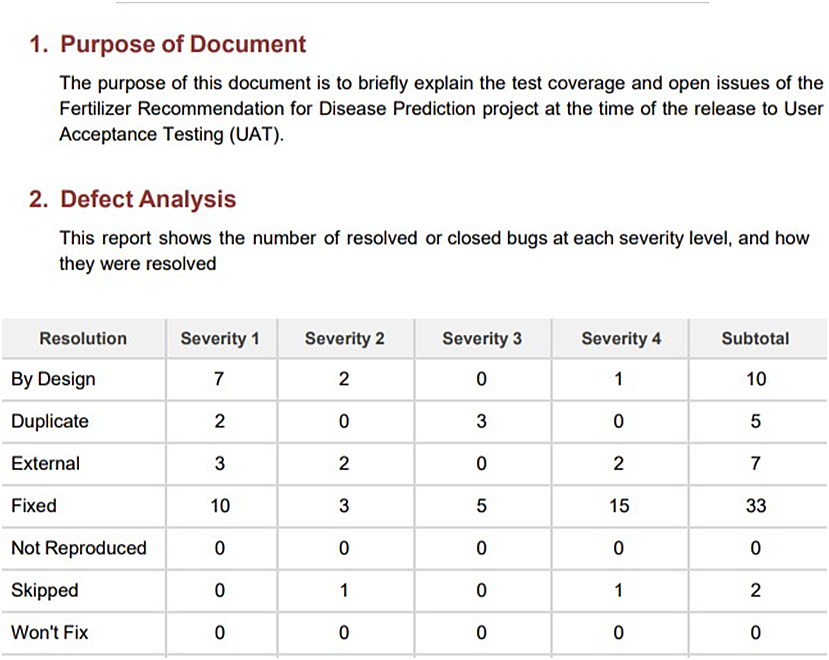
**TESTING**

**8.1 Test Cases**

****

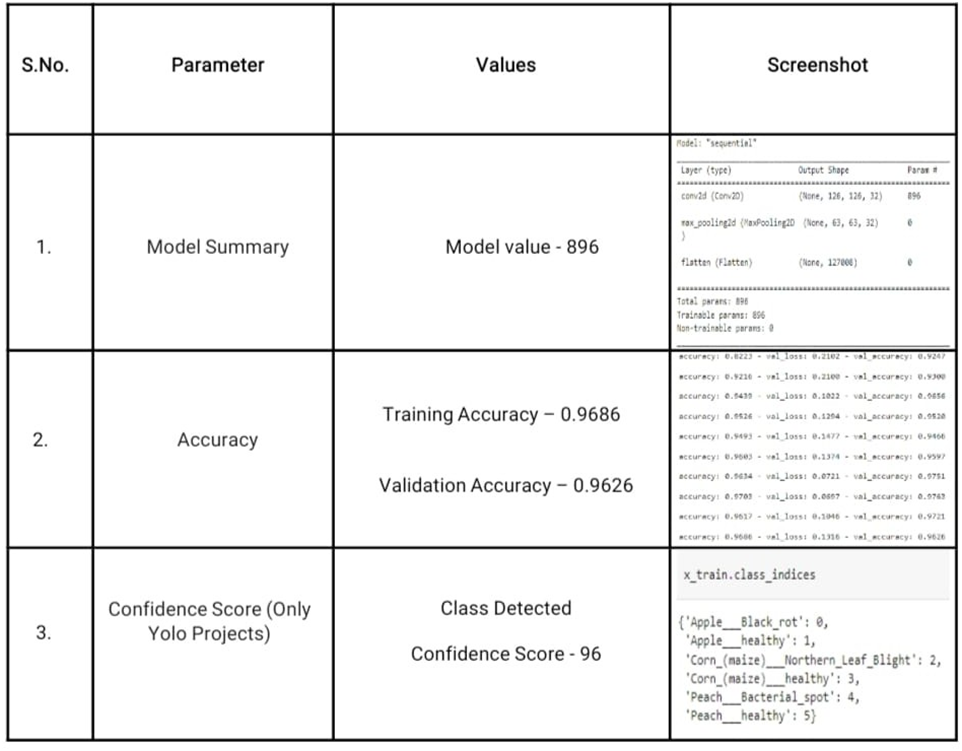
**8.2 User Acceptance Testing**

****

****

**RESULTS**

**9.1 Performance Metrics**

****

**ADVANTAGES & DISADVANTAGES**

**Advantages:**

✓ The system helps to compute the disease severity.

✔ 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.

✔ The prediction and diagnosing of leaf diseases are depending on the segmentation such

as segmenting the healthy tissues from diseased tissues of leaves.

**Disadvantages:**

✔ Due to the changing climatic conditions, accurate results cannot be predicted by this

system.

✔ System only able to detect the disease from citrus leaves.

**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 datasets.

**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 vide processing are possible with help OpenCV python library This project work can be extended for security applications such as figure print recognition, inis recognition and face recognition.

**APPENDIX**

**13.1 SOURCE CODE**

Home.html

1 <!DOCTYPE html>

2 <html >

3

4 <head>

5 <meta charset="UTF-8">

6 <meta name="viewport" content="width=device-width, initialscale=

1">

7 <title> Plant Disease Prediction</title>

8 <link

href='https://fonts.googleapis.com/css?family=Pacifico'

rel='stylesheet' type='text/css'>

9 <link href='https://fonts.googleapis.com/css?family=Arimo'

rel='stylesheet' type='text/css'>

10 <link href='https://fonts.googleapis.com/css?family=Hind:300'

rel='stylesheet' type='text/css'>

11 <link

href='https://fonts.googleapis.com/css?family=Open+Sans+Conden

sed:300' rel='stylesheet' type='text/css'>

12 <link rel="stylesheet" href="{{ url\_for('static',

filename='css/style.css') }}">

13 <link

href='https://fonts.googleapis.com/css?family=Merriweather'

rel='stylesheet'>

14 <link href='https://fonts.googleapis.com/css?family=Josefin

Sans' rel='stylesheet'>

15 <link

href='https://fonts.googleapis.com/css?family=Montserrat'

rel='stylesheet'>

16 <style>

17 .header {

18 top:0;

19 margin:0px;

20 left: 0px;

21 right: 0px;

22 position: fixed;

23 background-color: #28272c;

24 color: white;

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

26 overflow: hidden;

27 padding-left:20px;

28 font-family: 'Josefin Sans';

29 font-size: 2vw;

30 width: 100%;

31 height:8%;

32 text-align: center;

33 }

34 .topnav {

35 overflow: hidden;

36 background-color: #333;

37 }

38

39 .topnav-right a {

40 float: left;

41 color: #f2f2f2;

42 text-align: center;

43 padding: 14px 16px;

44 text-decoration: none;

45 font-size: 18px;

46 }

47

48 .topnav-right a:hover {

49 background-color: #ddd;

50 color: black;

51 }

52

53 .topnav-right a.active {

54 background-color: #565961;

55 color: white;

56 }

57

58 .topnav-right {

59 float: right;

60 padding-right:100px;

61 }

62

63 body {

64

65 background-color:#ffffff;

66 background-repeat: no-repeat;

67 background-size:cover;

68 background-position: 0px 0px;

69 }

70 .button {

71 background-color: #28272c;

72 border: none;

73 color: white;

74 padding: 15px 32px;

75 text-align: center;

76 text-decoration: none;

77 display: inline-block;

78 font-size: 16px;

79 border-radius: 12px;

80 }

81 .button:hover {

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

rgba(0,0,0,0.19);

83 }

84 form {border: 3px solid #f1f1f1; margin-left:400px;marginright:

400px;}

85

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

87 width: 100%;

88 padding: 12px 20px;

89 display: inline-block;

90 margin-bottom:18px;

91 border: 1px solid #ccc;

92 box-sizing: border-box;

93 }

94

95 button {

96 background-color: #28272c;

97 color: white;

98 padding: 14px 20px;

99 margin-bottom:8px;

100 border: none;

101 cursor: pointer;

102 width: 15%;

103 border-radius:4px;

104 }

105

106 button:hover {

107 opacity: 0.8;

108 }

109

110 .cancelbtn {

111 width: auto;

112 padding: 10px 18px;

113 background-color: #f44336;

114 }

115

116 .imgcontainer {

117 text-align: center;

118 margin: 24px 0 12px 0;

119 }

120

121 img.avatar {

122 width: 30%;

123 border-radius: 50%;

124 }

125

126 .container {

127 padding: 16px;

128 }

129

130 span.psw {

131 float: right;

132 padding-top: 16px;

133 }

134

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

small screens \*/

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

137 span.psw {

138 display: block;

139 float: none;

140 }

141 .cancelbtn {

142 width: 100%;

143 }

144 }

145

146 .home{

147 margin:80px;

148

149 width: 84%;

150 height: 500px;

151 padding-top:10px;

152 padding-left: 30px;

153 }

154 .login{

155 margin:80px;

156 box-sizing: content-box;

157 width: 84%;

158 height: 420px;

159 padding: 30px;

160 border: 10px solid blue;

161 }

162 .left,.right{

163 box-sizing: content-box;

164 height: 400px;

165 margin:20px;

166 border: 10px solid blue;

167 }

168

169 .mySlides {display: none;}

170 img {vertical-align: middle;}

171

172 /\* Slideshow container \*/

173 .slideshow-container {

174 max-width: 1000px;

175 position: relative;

176 margin: auto;

177 }

178

179 /\* Caption text \*/

180 .text {

181 color: #f2f2f2;

182 font-size: 15px;

183 padding: 8px 12px;

184 position: absolute;

185 bottom: 8px;

186 width: 100%;

187 text-align: center;

188 }

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

190 .dot {

191 height: 15px;

192 width: 15px;

193 margin: 0 2px;

194 background-color: #bbb;

195 border-radius: 50%;

196 display: inline-block;

197 transition: background-color 0.6s ease;

198 }

199

200 .active {

201 background-color: #717171;

202 }

203

204 /\* Fading animation \*/

205 .fade {

206 -webkit-animation-name: fade;

207 -webkit-animation-duration: 1.5s;

208 animation-name: fade;

209 animation-duration: 1.5s;

210 }

211

212 @-webkit-keyframes fade {

213 from {opacity: .4}

214 to {opacity: 1}

215 }

216

217 @keyframes fade {

218 from {opacity: .4}

219 to {opacity: 1}

220 }

221

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

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

224 .text {font-size: 11px}

225 }

226 </style>

227 </head>

228

229 <body style="font-family:'Times New Roman', Times,

serif;background-color:#C2C5A8;">

230

231 <div class="header">

232 <div style="width:50%;float:left;font-size:2vw;textalign:

left;color:white; padding-top:1%">Plant Disease

Prediction</div>

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

234 <a class="active" href="{{

url\_for('home')}}">Home</a>

235 <a href="{{ url\_for('prediction')}}">Predict</a>

236 </div>

237 </div>

238

239 <div style="backgroundimage:

url("./static/images/images.jpg");">

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

241 <div style="font-size:50px;fontfamily:

Montserrat;padding-left:20px;text-align:center;paddingtop:

10%;">

242 <b>Detect if your plant<br> is infected!!</b></div><br>

243 <div style="font-size:20px;fontfamily:

Montserrat;padding-left:70px;padding-right:30px;textalign:

justify;">Agriculture is one of the major sectors worls

wide. Over the years it has developed and the use of new

technologies and equipment replaced almost all the traditional

methods of farming. 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>

244 </div>

245 </div>

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

247 <img

src="{{url\_for('static',filename='images/12456.png')}}"

style="max-height:100%;max-width:100%;">

248

249 </div>

250 </div>

251

252 <div class="home">

253

254 <br>

255

256 </div>

257

258 <script>

259 var slideIndex = 0;

260 showSlides();

261

262 function showSlides() {

263 var i;

264 var slides =

document.getElementsByClassName("mySlides");

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

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

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

268 }

269 slideIndex++;

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

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

272 dots[i].className = dots[i].className.replace("

active", "");

273 }

274 slides[slideIndex-1].style.display = "block";

275 dots[slideIndex-1].className += " active";

276 setTimeout(showSlides, 2000); // Change image every 2

seconds

277 }

278 </script>

279 </body>

280 </html>

predict.html

1 <!DOCTYPE html>

2 <html >

3

4 <head>

5 <meta charset="UTF-8">

6 <meta name="viewport" content="width=device-width, initialscale=

1">

7 <title> Plant Disease Prediction</title>

8 <link

href='https://fonts.googleapis.com/css?family=Pacifico'

rel='stylesheet' type='text/css'>

9 <link href='https://fonts.googleapis.com/css?family=Arimo'

rel='stylesheet' type='text/css'>

10 <link href='https://fonts.googleapis.com/css?family=Hind:300'

rel='stylesheet' type='text/css'>

11 <link

href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.mi

n.css" rel="stylesheet">

12 <script

src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.j

s"></script>

13 <script

src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></scr

ipt>

14 <script

src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.

js"></script>

15 <link

href='https://fonts.googleapis.com/css?family=Open+Sans+Conden

sed:300' rel='stylesheet' type='text/css'>

16 <link

href='https://fonts.googleapis.com/css?family=Merriweather'

rel='stylesheet'>

17 <link href='https://fonts.googleapis.com/css?family=Josefin

Sans' rel='stylesheet'>

18 <link

href='https://fonts.googleapis.com/css?family=Montserrat'

rel='stylesheet'>

19 <link href="{{ url\_for('static', filename='css/final.css') }}"

rel="stylesheet">

20 <style>

21 .header {

22 top:0;

23 margin:0px;

24 left: 0px;

25 right: 0px;

26 position: fixed;

27 background-color: #28272c;

28 color: white;

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

30 overflow: hidden;

31 padding-left:20px;

32 font-family: 'Josefin Sans';

33 font-size: 2vw;

34 width: 100%;

35 height:8%;

36 text-align: center;

37 }

38 .topnav {

39 overflow: hidden;

40 background-color: #333;

41 }

42

43 .topnav-right a {

44 float: left;

45 color: #f2f2f2;

46 text-align: center;

47 padding: 14px 16px;

48 text-decoration: none;

49 font-size: 18px;

50 }

51

52 .topnav-right a:hover {

53 background-color: #ddd;

54 color: black;

55 }

56

57 .topnav-right a.active {

58 background-color: #565961;

59 color: white;

60 }

61

62 .topnav-right {

63 float: right;

64 padding-right:100px;

65 }

66

67 .login{

68 margin-top:-70px;

69 }

70 body {

71

72 background-color:#ffffff;

73 background-repeat: no-repeat;

74 background-size:cover;

75 background-position: 0px 0px;

76 }

77 .login{

78 margin-top:100px;

79 }

80

81 .container {

82 margin-top:40px;

83 padding: 16px;

84 }

85 select {

86 width: 100%;

87 margin-bottom: 10px;

88 background: rgba(255,255,255,255);

89 border: none;

90 outline: none;

91 padding: 10px;

92 font-size: 13px;

93 color: #000000;

94 text-shadow: 1px 1px 1px rgba(0,0,0,0.3);

95 border: 1px solid rgba(0,0,0,0.3);

96 border-radius: 4px;

97 box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px

rgba(255,255,255,0.2);

98 -webkit-transition: box-shadow .5s ease;

99 -moz-transition: box-shadow .5s ease;

100 -o-transition: box-shadow .5s ease;

101 -ms-transition: box-shadow .5s ease;

102 transition: box-shadow .5s ease;

103 }

104

105 </style>

106 </head>

107

108 <body style="font-family:Montserrat;overflow:scroll;">

109

110 <div class="header">

111 <div style="width:50%;float:left;font-size:2vw;textalign:

left;color:white; padding-top:1%">Plant Disease

Prediction</div>

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

113 </div>

114 </div>

115 <div class="container">

116 <div id="content" style="margin-top:2em">

117 <div class="container">

118 <div class="row">

119 <div class="col-sm-6 bd" >

120

121 <br>

122 <img

src="{{url\_for('static',filename='images/images.jpg')}}"

style="height:450px;width:550px"class="img-rounded" alt="Gesture">

123 </div>

124 <div class="col-sm-6">

125 <div>

126 <h4>Drop in the image to get the prediction

</h4>

127 <form action = "" id="upload-file" method="post"

enctype="multipart/form-data">

128 <select name="plant">

129

130 <option value="select" selected>Select plant

type</option>

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

132 <option value="vegetable">Vegetable</option>

133 </select><br>

134 <label for="imageUpload" class="upload-label"

style="background: #28272c;">

135 Choose...

136 </label>

137 <input type="file" name="image" id="imageUpload"

accept=".png, .jpg, .jpeg">

138 </form>

139

140

141 <div class="image-section" style="display:none;">

142 <div class="img-preview">

143 <div id="imagePreview">

144 </div>

145 </div>

146 <div>

147 <button type="button" class="btn btn-info btn-lg

" id="btn-predict" style="background: #28272c;">Predict!</button>

148 </div>

149 </div>

150

151 <div class="loader" style="display:none;"></div>

152

153 <h3>

154 <span id="result" style="font-size:17px; "> </span>

155 </h3>

156

157 </div>

158 </div>

159

160 </div>

161 </div>

162 </div>

163 </div>

164 </body>

165

166 <footer>

167 <script src="{{ url\_for('static', filename='js/main.js')

}}" type="text/javascript"></script>

168 </footer>

169 </html>

final.css

1 .img-preview {

2 width: 256px;

3 height: 256px;

4 position: relative;

5 border: 5px solid #F8F8F8;

6 box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);

7 margin-top: 1em;

8 margin-bottom: 1em;

9 }

10

11 .img-preview>div {

12 width: 100%;

13 height: 100%;

14 background-size: 256px 256px;

15 background-repeat: no-repeat;

16 background-position: center;

17 }

18

19 input[type="file"] {

20 display: none;

21 }

22

23 .upload-label{

24 display: inline-block;

25 padding: 12px 30px;

26 background: #28272c;

27 color: #fff;

28 font-size: 1em;

29 transition: all .4s;

30 cursor: pointer;

31 }

32

33 .upload-label:hover{

34 background: #C2C5A8;

35 color: #39D2B4;

36 }

37

38 .loader {

39 border: 8px solid #f3f3f3; /\* Light grey \*/

40 border-top: 8px solid #28272c; /\* Blue \*/

41 border-radius: 50%;

42 width: 50px;

43 height: 50px;

44 animation: spin 1s linear infinite;

45 }

46

47 @keyframes spin {

48 0% { transform: rotate(0deg); }

49 100% { transform: rotate(360deg); }

50 }

Main.js

1 $(document).ready(function () {

2 // Init

3 $('.image-section').hide();

4 $('.loader').hide();

5 $('#result').hide();

6

7 // Upload Preview

8 function readURL(input) {

9 if (input.files && input.files[0]) {

10 var reader = new FileReader();

11 reader.onload = function (e) {

12 $('#imagePreview').css('background-image',

'url(' + e.target.result + ')');

13 $('#imagePreview').hide();

14 $('#imagePreview').fadeIn(650);

15 }

16 reader.readAsDataURL(input.files[0]);

17 }

18 }

19 $("#imageUpload").change(function () {

20 $('.image-section').show();

21 $('#btn-predict').show();

22 $('#result').text('');

23 $('#result').hide();

24 readURL(this);

25 });

26

27 // Predict

28 $('#btn-predict').click(function () {

29 var form\_data = new FormData($('#upload-file')[0]);

30

31 // Show loading animation

32 $(this).hide();

33 $('.loader').show();

34

35 // Make prediction by calling api /predict

36 $.ajax({

37 type: 'POST',

38 url: '/predict',

39 data: form\_data,

40 contentType: false,

41 cache: false,

42 processData: false,

43 async: true,

44 success: function (data) {

45 // Get and display the result

46 $('.loader').hide();

47 $('#result').fadeIn(600);

48 $('#result').text('Prediction: '+data);

49 console.log('Success!');

50 },

51 });

52 });

53

54 });

app.py

1 import requests

2 from tensorflow.keras.preprocessing import image

3 from tensorflow.keras.models import load\_model

4 import numpy as np

5 import pandas as pd

6 import tensorflow as tf

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

8 import os

9 from werkzeug.utils import secure\_filename

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

11

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

13 global sess

14

15 global graph

16 graph=tf.compat.v1.get\_default\_graph()

17

18

19

20 model = load\_model("./Models/fruit.h5")

21 model1=load\_model("./Models/vegetable.h5")

22

23

24 @app.route('/')

25 def home():

26 return render\_template('home.html')

27

28

29 @app.route('/prediction')

30 def prediction():

31 return render\_template('predict.html')

32

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

34

35 def predict():

36 if request.method == 'POST':

37

38 f = request.files['image']

39

40

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

42 file\_path = os.path.join(

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

44 f.save(file\_path)

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

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

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

48

49 plant=request.form['plant']

50 print(plant)

51

52 if(plant=="vegetable"):

53 preds = model.predict(x)

54 preds = np.argmax(preds)

55 print(preds)

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

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

58 else:

59 preds = model1.predict(x)

60 preds = np.argmax(preds)

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

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

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

64

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

66 app.run(debug=False)

GitHub Link: <https://github.com/IBM-EPBL/IBM-Project-51926-1660986906>

DEMO VIDEO LINK :

[**https://drive.google.com/file/d/1DaEQXeEDmY9kubNqdZi\_ARl27k84VpAv/view?usp=share\_link**](https://drive.google.com/file/d/1DaEQXeEDmY9kubNqdZi_ARl27k84VpAv/view?usp=share_link)