Fertilizers Recommendation System For Disease Prediction

PROJECT REPORT

Team ID: PNT2022TMID19068

TEAM members:

1) Shri varsha priya M 1919103111

2) Savitha E 1919103104

3) Srinith R 1919103120

4)Kamal C 1919103046

5) Santhosh S 1919103098

1. INTRODUCTION

1.1 Project Overview

In this project, two datasets name fruit dataset and vegetable dataset are collected. The collected datasets are trained and tested with deep learning neural network named Convolution al Neural Networks(CNN). First, the fruit dataset is trained and then tested with CNN.It has 6 classes and all the classes are trained and tested.Second, the vegetable dataset is trained and tested.The software used for training and testing of datasets is Python.All the Python codes are first written in Jupyter notebook supplied along with Anaconda Python and then the codes are tested in IBM cloud. Finally a web based framework is designed with help Flask a Python library.There are 2 html files are created in templates folder along with their associated files in static folder.The Python program 'app.py' used to interface with these two webpages is written in Spy der-Anaconda python and tested.

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 predicted diseases.

2. 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 recommendation system for predicted 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 ac curacies.

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-Apr2021.

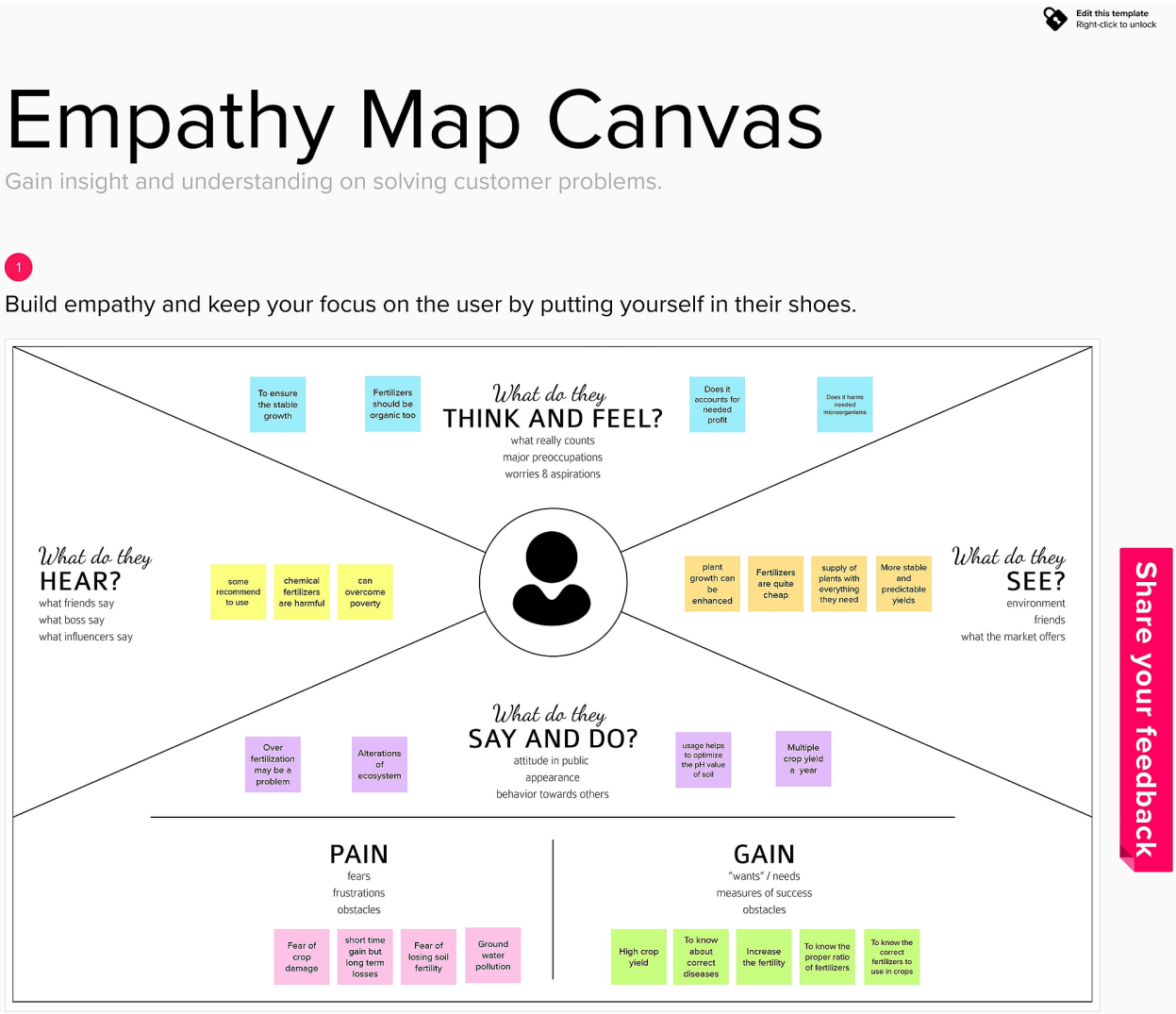
[3]."H Shiva reddy, Ganesh hedge, Prof. DR Chinnaya3 IoT based Leaf Disease Detection and Fertilizer Recommendation", International Research Journal of Engineering and Technology (IRJET), Volume: 06 Issue:11,Nov 2019, e-ISSN:2395-
0056.

2.3 Problem Statement Definition

Farmers' conventional methods of agricultural cultivation are ineffective. It does not make proper use of all available resources. Farmers are unable to detect crop diseases due to a lack of knowledge and old practices, which often result in soil nutrient deterioration and exhaustion. As a result, crop failure occurs. Growing only certain crops depletes the soil, and if the crops are harmed by illnesses, farmers are uninformed of how to recover such crops. Food needs cannot be met until and unless efficient resource management and use is implemented.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming

Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

⏱ 5 minutes

PROBLEM

Diseases on plants placed a major constraint on the production and a major threat to food security and for farmers. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality.

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

⏱ 10 minutes

TIP

You can selec
and hit the pe
sketch] icon t

varsha

In recent years, the number of diseases on plants and the degree of their occurrence has increased due to the variation in pathogen varieties.

SVM technique gives a better result when compared to existing CNN

By providing an image of a leaf farmer gets an idea of which disease caught their crop

Data Mining is a rising studies area in crop yield analysis

The disease-based similarity measure is used for fertilizer recommendation

savitha

algorithms have proved that a higher crop yield can be achieved

providing details like potassium, pH level farmer get the idea of which crop is best for their soil

Supervised Learning algorithms like Classification and Regression can be used for prediction of crop yield

AI enables better decision-making. AI reduces cost savings

srinith

We can use various deep learning techniques

algorithm can be modified further to identify the disease and also pests, insects, stems and fruits

svm is a binary classifieranalyze the data and recognize the pattern for classification.

kamal

What are the different techniques used in fertilizers/crop recommendation system ?

santhosh

Leaves are affected by bacteria, fungi, virus, and other insects.

The measurements of fertilizer recommendation based on disease severity

The prediction of crop yield based on soil data and proper implementation of algorithms

Person 6

Person 7

Person 8

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

⏱ 20 minutes

TIP

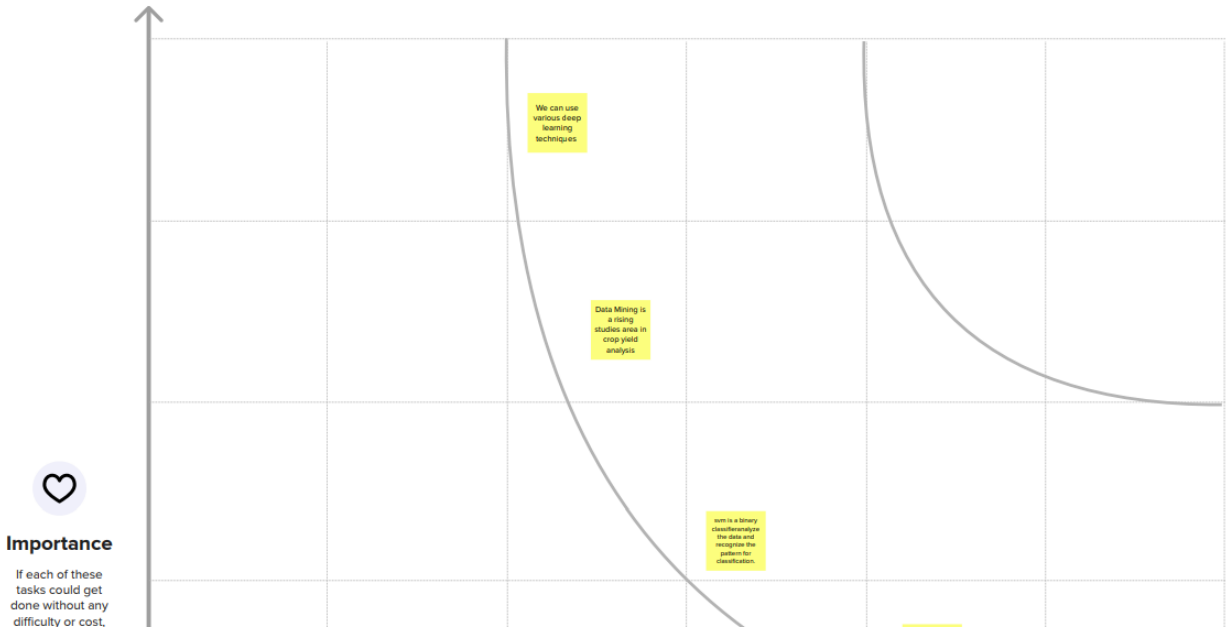
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

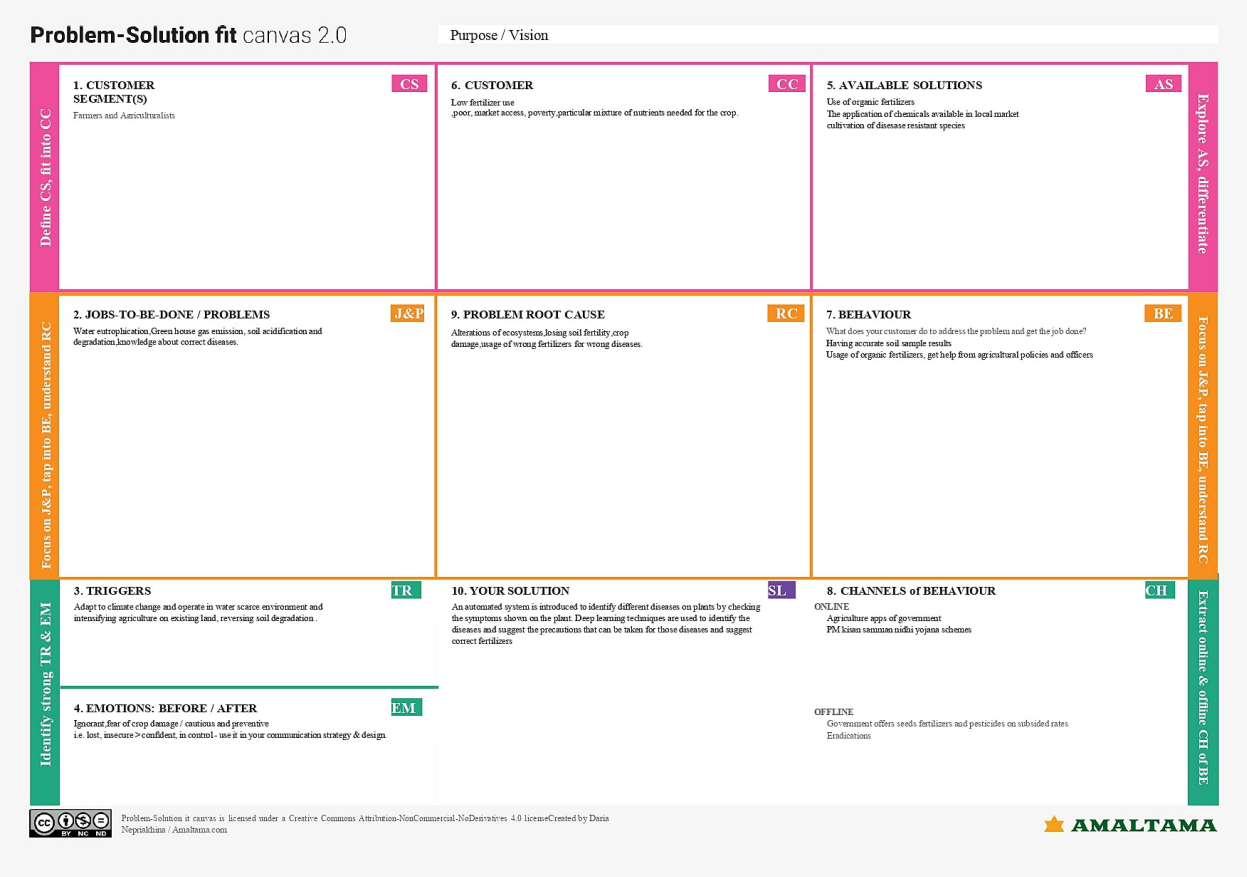


3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Diseases on plants placed a major constraint on the production and a major threat to food security and for farmers. Hence, early and accurate identification of plant diseases and usage of correct fertilizers is essential to ensure high quantity and best quality of crops.
2.	Idea / Solution description	An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the plant. Deep learning techniques are used to identify the diseases and suggest the fertilizers and precautions that can be taken for those diseases.
3.	Novelty / Uniqueness	The CNN model learns the filters whereas traditional algorithms used various activation functions to train and classify the output. Farmers can interact with the portal build Interacts with the user interface to upload images of diseased leaf Our model built analyses the Disease and suggests the farmer with fertilizers are to be used
4.	Social Impact / Customer Satisfaction	The aim of this project is early prediction of crop disease with greater accuracy and prevention of further damage done to the crops. The area of the disease affected is also found so that fertilizers application can be optimized. so the Farmers can be cautious and preventive.
5.	Business Model (Revenue Model)	This system detects the presence of disease in leaf at early stages. The camera connected to Raspberry Pi kit captures the leaf image and is processed in both anaconda navigator and ARM processor for classification of disease using CNN. Further the processed image is clustered using clustering algorithm in MATLAB to find the area affected.

6.	Scalability of the Solution	This system will help he farmers to choose the right crop for their land and to give the suitable amount of fertilizer to produce the maximum yield.

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution .

Fr.no	Functional requirement	Sub requirement (story/subtask)
Fr-1	User registration	Registration through form Registration through Gmail
Fr-2	User confirmation	Confirmation via OTP Confirmation via Email
Fr-3	Capturing image	Capture the image of the leaf and check the parameter of the captured image .
Fr-4	Image processing	Upload the image for the prediction of the disease in the Leaf and plant.
Fr-5	Leaf identification	Identify the leaf and predict the disease in leaf.
Fr-6	Image description	Suggesting the best fertilizer for the disease .

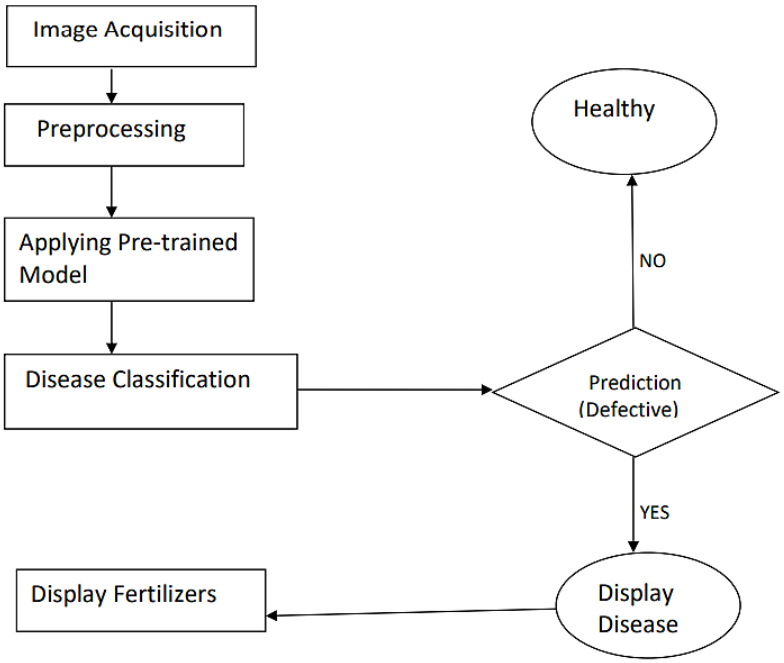
4.2 Non - Functional requirement

Following is the non-functional requirement of the proposed solution

NFr.no	Non-functional requirement	Description
Nfr-1	Usability	Datasets of all the leaf is used to detecting the disease that present in the leaf.
Nfr-2	Security	The information belongs to the user and leaf are secured highly.
Nfr-3	Reliability	The leaf quality is important unpredicting the malnutrition in leaf.
Nfr-4	Performance	The performance is based on the quality of the leaf used for disease prediction.
Nfr-5	Availability	It is available for all user to predict the disease in the plant.
Nfr-6	Scalability	Increasing the prediction of the disease in the leaves and plants.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

Technical Architecture:

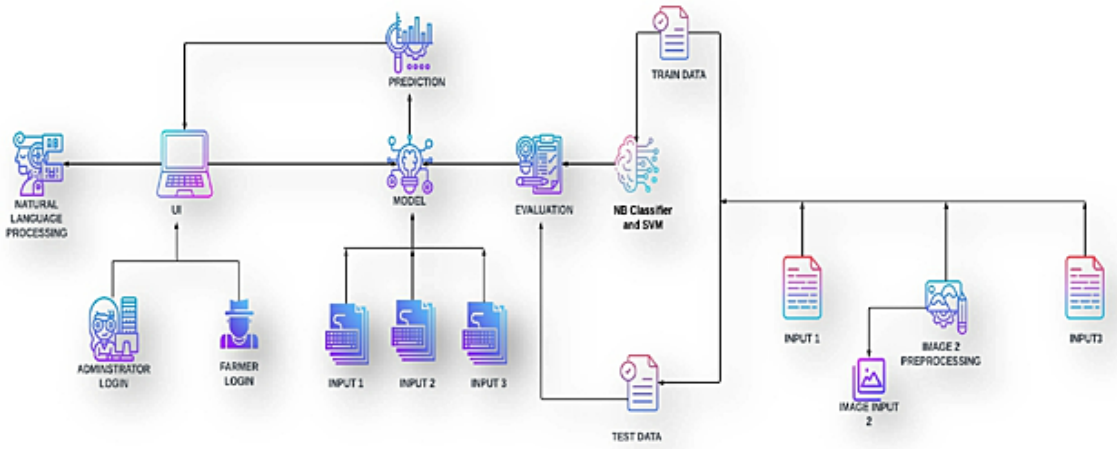


Table-1 : Components & Technologies

S.No	Component	Description	Technology
1.	User Interface	How the user interacts with the application .To depict the human-computer interaction and communication.	HTML, CSS,JSP
2.	Application Logic-1	A page to upload images as input	Python
3.	Application Logic-2	To use the Machine Learning model and predicting the result	Python

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	I can register as a user on the website with eitheran email address or a phone number and password.	I can createmy account.	High	Sprint-3
	Login	USN-2	With theprovided Login credentials, I canaccessthe website as a user.	I can log in andaccess myaccount .	High	Sprint-3
	Upload image	USN-3	I can post my data as a userin formats likepdf and doc.	I can uploadmy data.	Medium	Sprint-3
Customer (Web user)	Admin Login	USN-4	. As a user, I can login to web dashboard just Like website dashboard	I can log in and analyze the user data.	High	Sprint-3
	Data collection	USN-5	As a user, I can login to myweb dashboard with the login credentials	I can collect the dataset.	Low	Sprint-1
	Create model	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 create andtrain the model.	High	Sprint-1
	Test the model	USN-7	As a user, the fertilizer recommended to me Is in high accurate	I can testthe model.	High	Sprint-2
Administrator	Diagnosis	USN-8	I can access the application's diagnosisresults as a userand continue with treatments..	I can access my dashboard	High	Sprint-2

6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Milestone and Activity List:

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	17 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	18 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	19 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	02 OCTOBER 2022
Problem Solution Fit	Prepare problem - solution Fit document.	03 0CTOBER 2022

Solution Architecture	Prepare solution Architecture document.	03 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application.	12 OCTOBER 2022
Functional Requirements	Prepare the functional requirement document.	09 OCTOBER 2022
Data Flow Diagrams	Draw the data flow Diagrams and submit for review.	10 OCTOBER 2022
Technology Architecture	Architecture diagram.	17 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & Activity list of the project.	21 OCTOBER 2022
Project Development Delivery of Sprint- 1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS.

6.2 Sprint Delivery Schedule

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	22 October 2022
Team ID	PNT2022TMID19068
Project Name	Fertilizers Recommendation System for Disease Prediction
Maximum Marks	8 Marks

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	I need to collect data (images of crops and with disease stages).	5	High	Savitha Kamal
Sprint-1	Data Pre-Processing	USN-2	I need to clean my data and prepare it for model building by doing pre-processing activities such as resizing etc.	5	High	Srinith Santhosh
Sprint-2	Model Building	USN-3	I need to build the model using Convolutional Neural Network (CNN).	10	High	Shri varsha priya Savitha
Sprint-3	Model Deployment	USN-4	I need to deploy the Machine Learning model that was built.	10	Medium	Srinith Santhosh
Sprint-4	Application Building	USN-5	I need to build the website for the application using HTML, CSS etc.	8	High	Shri varsha priya Kamal

Sprint-4	Home Page	USN-6	As a user, I can view the home page which hasa description of the plant disease as well as options to sign up or log in.	2	Low	Kamal Santhosh Savitha Shri varsha priya Srinith
Sprint-4	Sign Up Page	USN-7	As a user, I can register for the application byentering my name, phone number, email, password, and confirming my password.	2	High	Kamal Santhosh Savitha Shri varsha priya Srinith
Sprint-4	Authorization	USN-8	As a user, I will receive confirmation email oncel have registered for the application.	2	High	Kamal Santhosh Savitha Shri varsha priya Srinith
Sprint-4	Login	USN-9	As a user, I can log into the application byentering email & password.	2	High	Kamal Santhosh Savitha Shri varsha priya Srinith
Sprint-4	Dashboard	USN-10	As a user, I can upload images of the affected crop to the website in order to receive the plantdisease.	4	High	Kamal Santhosh Savitha Shri varsha priya Srinith

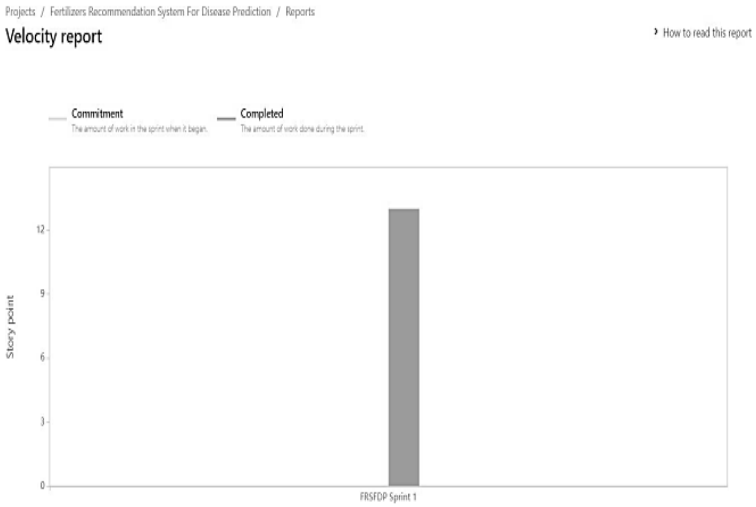
Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	10	6 Days	31 Oct 2022	05 Nov 2022	10	06 Nov 2022
Sprint-3	10	6 Days	07 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

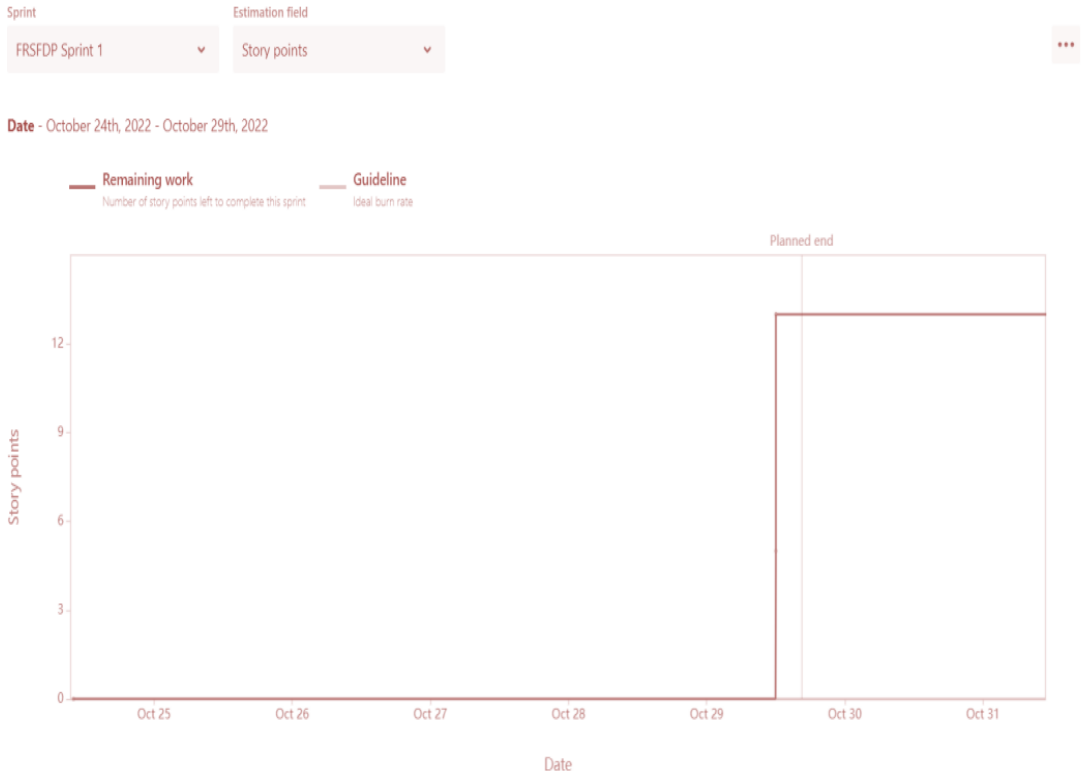
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

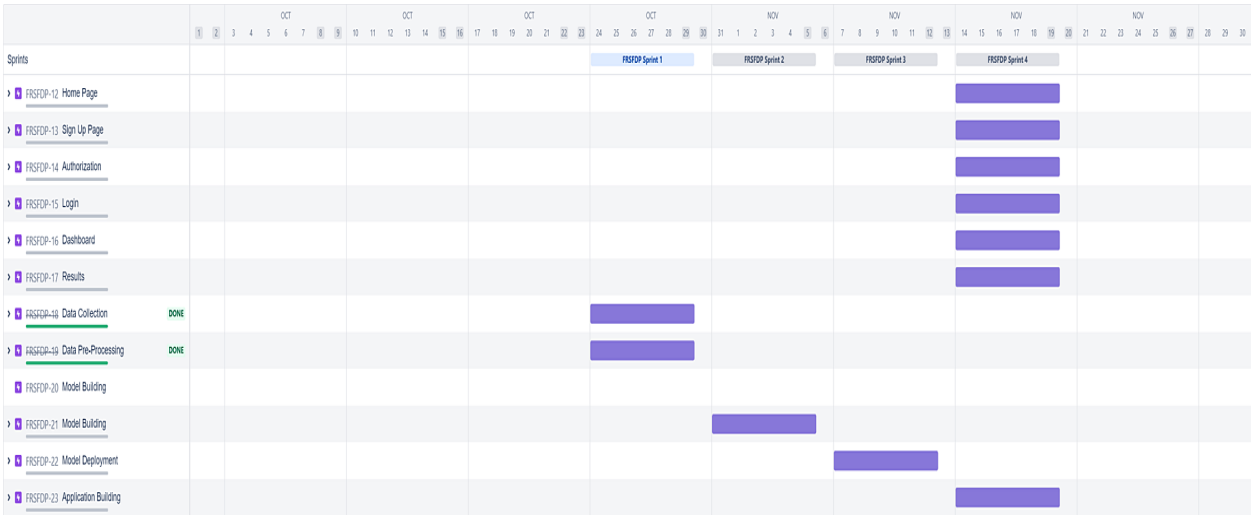


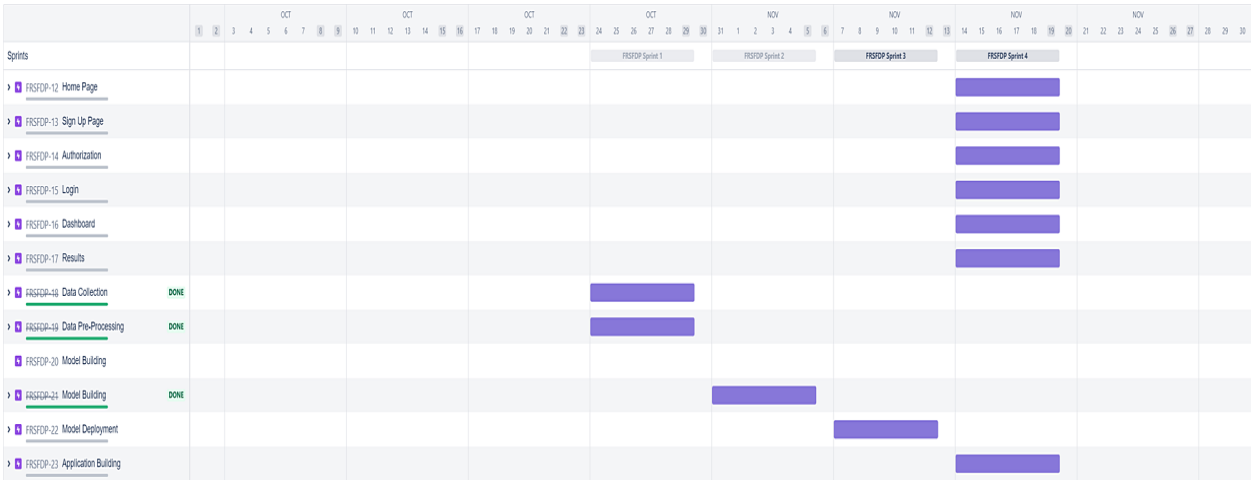
Burndown Chart:

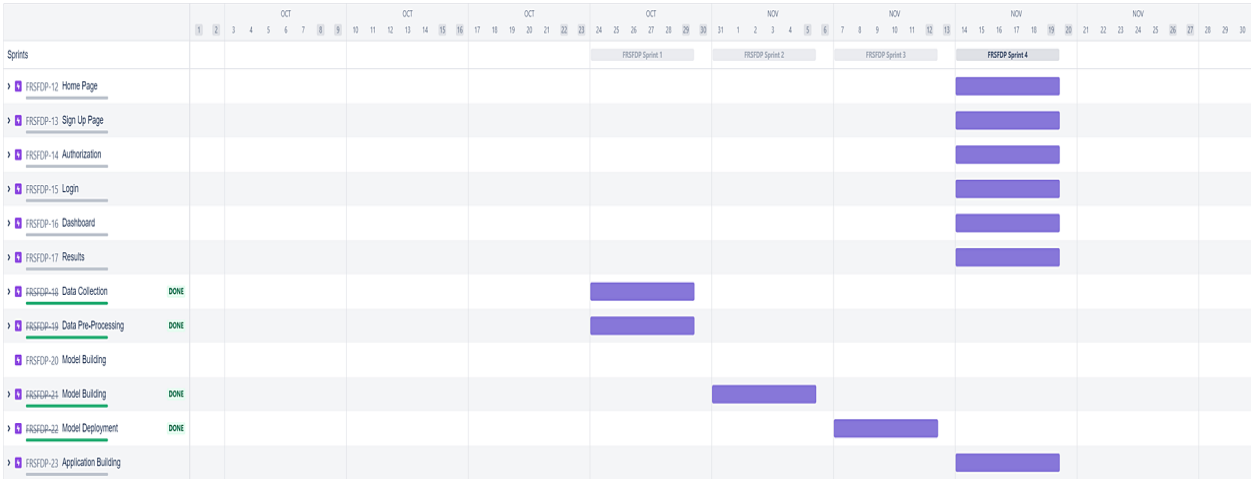
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

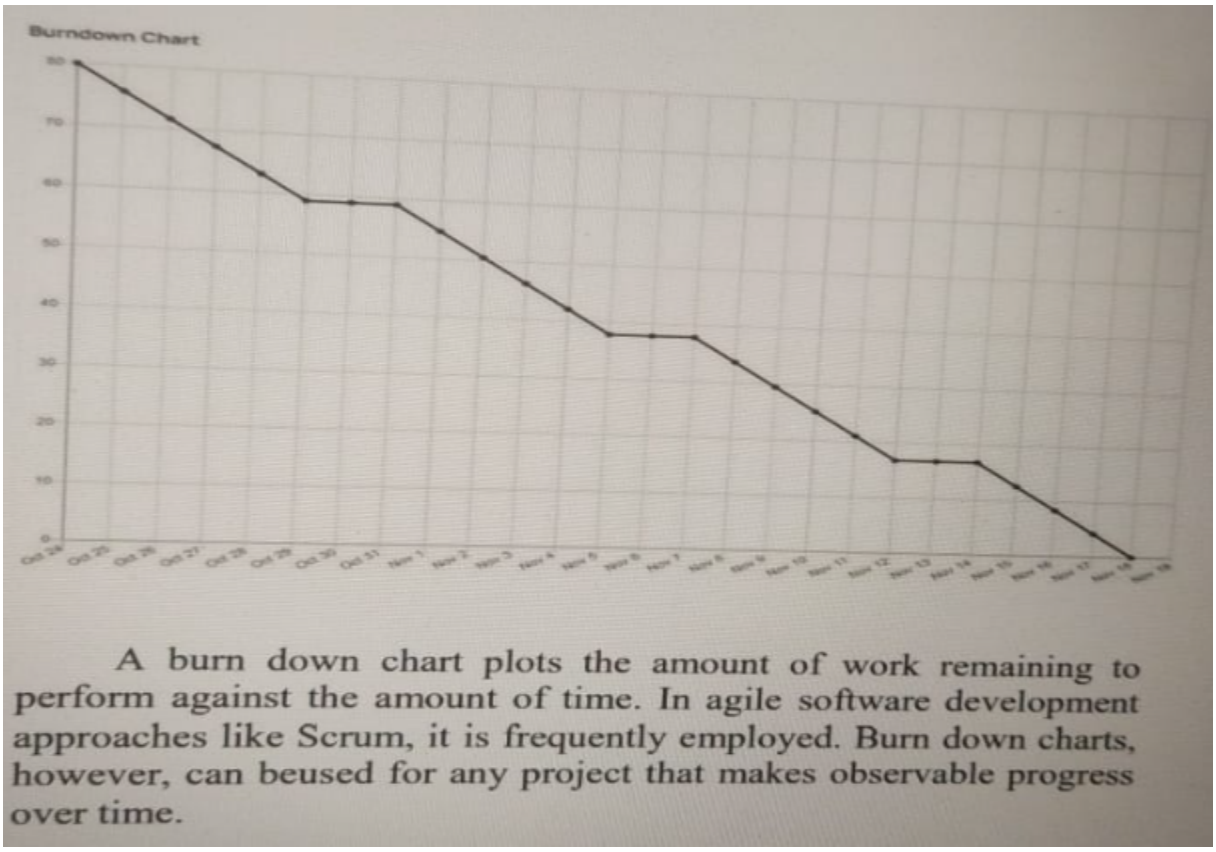


6.3 Reports from JIRA

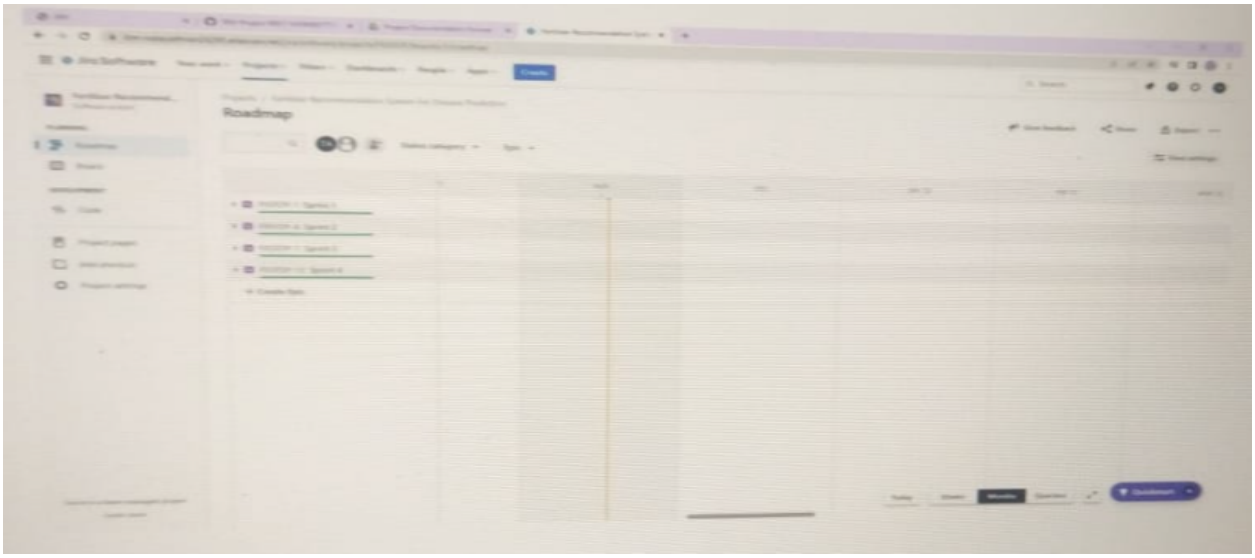








A burn down chart plots the amount of work remaining to perform against the amount of time. In agile software development approaches like Scrum, it is frequently employed. Burn down charts, however, can beused for any project that makes observable progress over time.



7.CODING AND SOLUTIONING

7.1 Feature 1

We have devloped a website which authenticates users and help them upload and predict the disease.

```
<!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' t
                <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='
                <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' typ
                <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='
                <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: #333;
}

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

    <a class="active" href="{{ url_for('home')}}">Home</a>
    <a href="/Predict" 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:cen
<b>Detect if your plant<bar> is infected!!</b></div><br>
<div style="font-size:20px;font-family:Montserrat;padding-left:70px;padding-right:
  civilization, whereby farming of acclimatize species produced food oversupply th
  Plants were independently sophisticated in at least 11 regions of the world. Ind
  about 2 billion people still depended on maintaining agriculture.The plant disea
effort and labor costs.</div><br><br>
</div>
</div>
<div style="width:40%;float:right;"><br><br>
<img src="https://images.pexels.com/photos/35196/water-plant-green-fine-layers.jpg

</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>
```

7.2 Feature 2

We have devloped a multilayer deep convolutional nueral network that classifies the user image of leaves to which extense has the disease has been affected.The model will classify the images into and report them on asking for prediction.

```
# -*- 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_PAcVbNWo
                        """

                        !unzip '/content/drive/MyDrive/ibm dataset/Fertilizers_Recommendation_ Syst

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

                        x_train=train_datagen.flow_from_directory('/content/Dataset Plant Disease/V
                        x_test=test_datagen.flow_from_directory('/content/Dataset Plant Disease/Veg

                        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=
                        test_datagen=ImageDataGenerator(rescale=1)
```

```
x_train=train_datagen.flow_from_directory('/content/Dataset Plant Disease/V
x_test=test_datagen.flow_from_directory('/content/Dataset Plant Disease/Veg

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=["ac
model.fit(x_train,steps_per_epoch=89,epochs=20,validation_data=x_test,valid

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-data
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)
```

8.TESTING

8.1 Test Cases

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

Section	TotalCases	Not Tested	Fail	Pass
Leaf spots	17	0	0	17
Mosaic leaf pattern	51	0	0	51
Misshapen leaves	20	0	0	20
Yellow leaves	7	0	0	7
Fruit rots	9	0	0	9
Fruit spots	4	0	0	4
Blights	2	0	0	2

8.2 User Acceptance Testing

1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Leaf spots	10	4	2	3	19
Mosaic leaf pattern	9	6	3	6	24
Misshapen leaves	2	7	0	1	10
Yellow leaves	11	4	3	20	38
Fruit rots	3	2	1	0	6
Fruit spots	5	3	1	1	10
Blights	4	5	2	1	12
Totals	44	31	13	32	119

3. Test Case Analysis

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

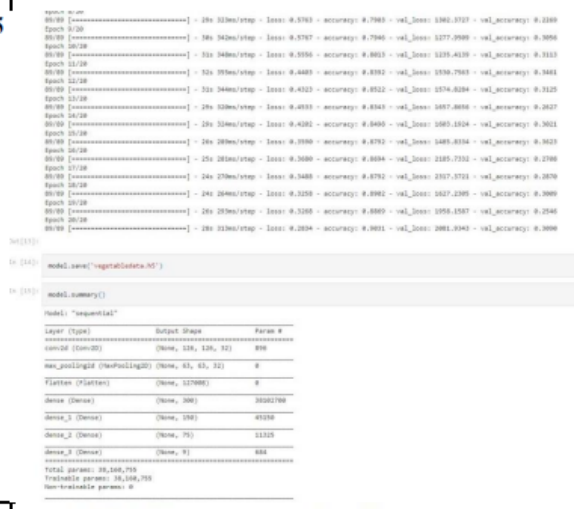
Section	Total Cases	Not Tested	Fail	Pass
Leaf spots	17	0	0	17
Mosaic leaf pattern	51	0	0	51
Misshapen leaves	20	0	0	20
Yellow leaves	7	0	0	7
Fruit rots	9	0	0	9
Fruit spots	4	0	0	4
Blights	2	0	0	2

9.RESULTS

9.1 Performance Metrics

In this project, we are obtaining adequate findings for proper crop production and fertilizer to recommend to farmers for crop cultivation. The

disease detection tool also provides the finest advice for recovering from crop disease, ensuring that the crop or specific land is not ruined and that soil fertility and crop yield are increased.

S. NO	Parameter	Values	Screenshot
1.	Model Summary	**Total params: 38,160,755** **Trainable params: 38,160,755** **Non-trainable params: 0**	
2.	Accuracy	Training Accuracy – **0.9031** Validation Accuracy – **loss 2081.9343**	loss: 0.2834 - accuracy: 0.9031
3.	Confidence Score(Only Yolo Projects)	Class Detected - Confidence Score -	-- -- -- --

10.ADVANTAGES & DISADVANTAGES

List of advantage:

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

List of disadvantages:

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

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 ac curacies are obtained.
- The ac curacies are increased by increasing more convolution layers.
- The accuracy of classification also increased by varying dense layers.
- Different ac curacies are obtained by varying the size of kernel used in the convolution layer output.

12.FUTURE SCOPE

The proposed model in this project work can be extended to image recognition.The entire model can be converted to application software using python to exe software.The real time image classification, image recognition and video processing are possible with help OpenCV python library. This project work can be extended for security applications such as figure print recognition,iris recognition and face recognition.

13.APPENDIX

Source Code

app.py:- !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,z oom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1)
x_train=train_datagen.flow_from_directory('/content/Dataset Plant Disease/Veg-dataset/Vegdataset/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/Vegdataset/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,z oom_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/Vegdataset/test_set',target_size=(128,128),batch_size=16,cla
ss_mode= 'categorical') model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activati
on='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",
me trics=["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/fruitdataset/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[press[0]]['caution'] if __name__=="__main__": app.run(debug=False) home.

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-26649-1660032435

PROJECT DEMO LINK:

https://drive.google.com/file/d/1-

noW9eJtcaJGqlnRnIC9aP847W8xPop-/view?usp=drivesdk