FERTILISER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

TEAM ID : PNT2022TMID27049
TEAM LEAD : Dhivya Shri Thendral G
TEAM MEMBERS : Arivumathi R
Hamlin Anshika A
Kavya K

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 PUPRPOSE:

- It allows us to predict which crops would be appropriate for a given climate.
- The proposed method uses SVM to classify tree leaves, identify the disease and suggest the fertilizer.
- The proposed method is compared with the existing CNN based leaf disease prediction. The proposed SVM technique gives a better result.

2.LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

In our case When a pathogen that is already present or which invades successfully to plant host tissues and cells results in plant disease. It is important to fix the problem because Plant diseases Reduce the amount of food available to humans by ultimately interfering with crop yields.This can cause inadequate food to humans which result in starvation or death in worst cases.

2.2 REFERENCES:

[1] Tanha Talaviya, Dhara Shah, Nivedita Patel, Manan Shah, “Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides” Volume 4, 2020, Pages 58-73.

[2] Anurag Saxena, Truptimayee Suna and Dipankar Saha Regi, “Application of Artificial Intelligence in Indian Agriculture” 2020.

[3] Nilay Ganatra and Atul Patel, “A Survey on Diseases Detection and Classification of Agriculture Products using Image Processing and Machine Learning”, International Journal of Computer Applications (0975 – 8887) Volume 180 – No.13, January 2018.

[4] Ngozi Clara Eli-Chukwu, “Applications of Artificial Intelligence in Agriculture: A Review”, Engineering, Technology & Applied Science Research Vol. 9, No. 4, 2019, 4377-4383.

5] Dr.K.Thangadurai, K.Padmavathi, “Computer Visionimage Enhancement For Plant Leaves Disease Detection”, 2014 World Congress on Computing and Communication Technologies.

2.3 PROBLEM STATEMENT DEFINITION:

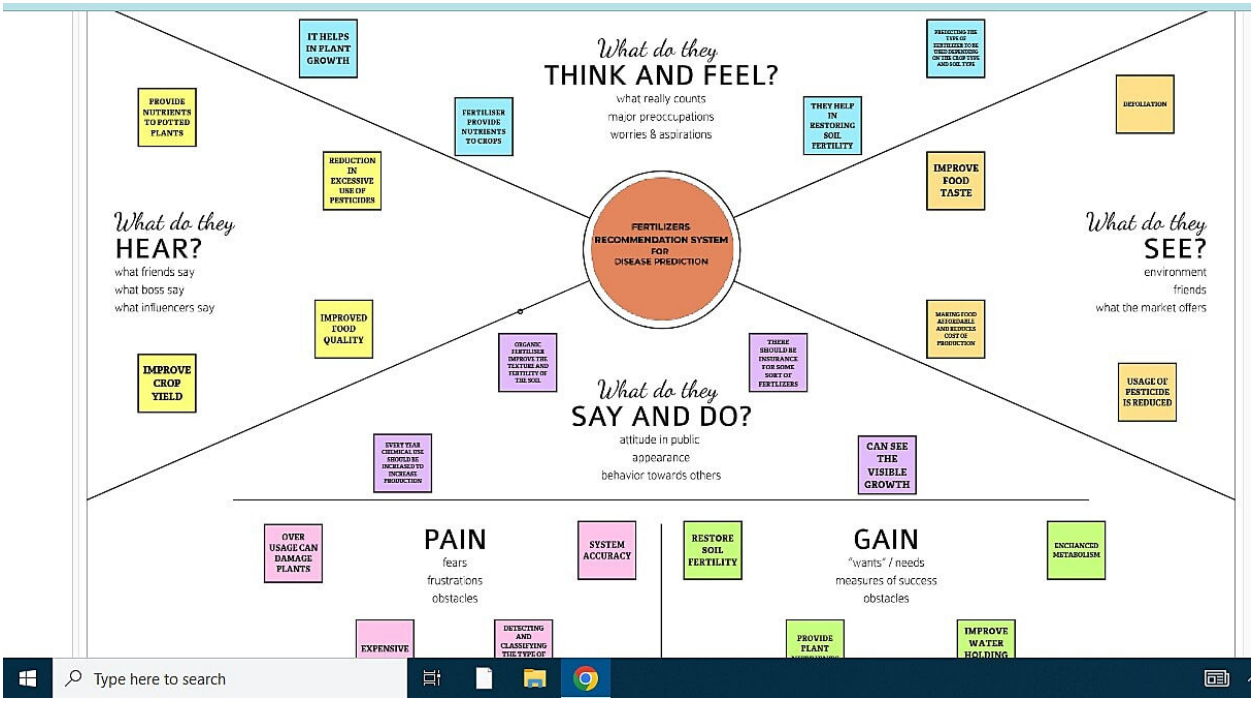
The Problem statement Comprises set of questions which the project seeks to address.It identifies the current state and future state and any gaps between the two. The Problem arised here in this project is:

1.Where does the problem affect?
2.What is the impact of the issues?
3.What would happen if we did'nt solve the problem?
4.When does the issue occur?
5.Where is the issue occuring ?

3.IDEATION AND PROPOSED SOLUTION:

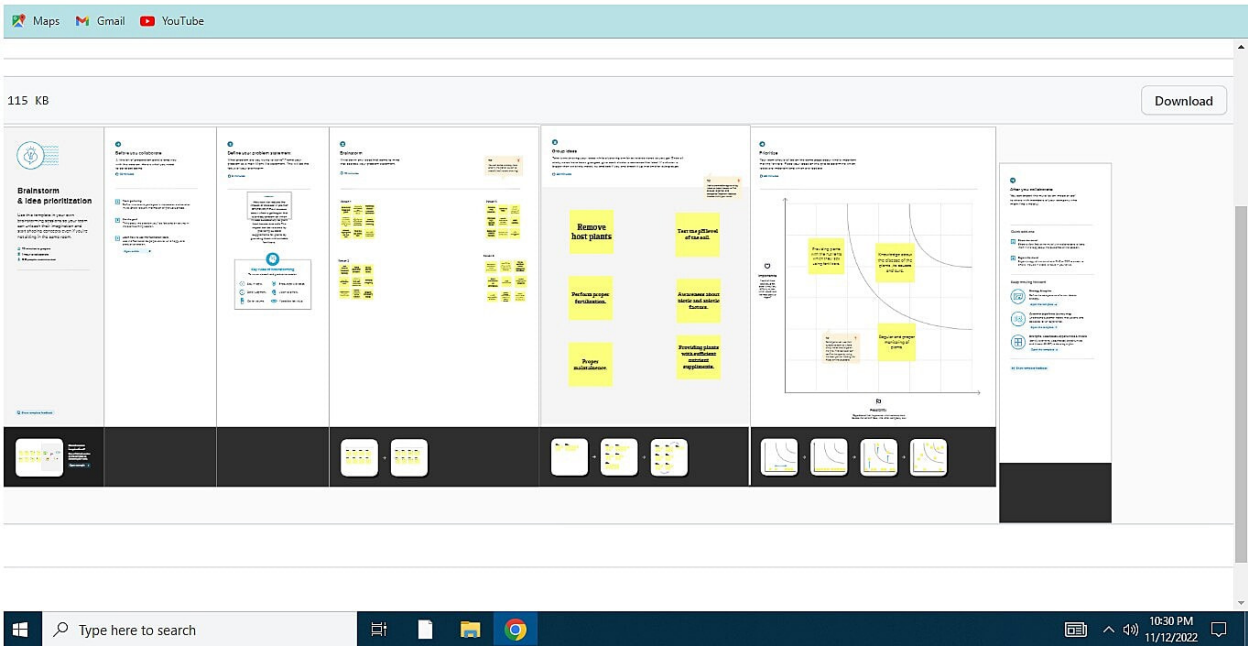
3.1 EMPATHY MAP:

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers.



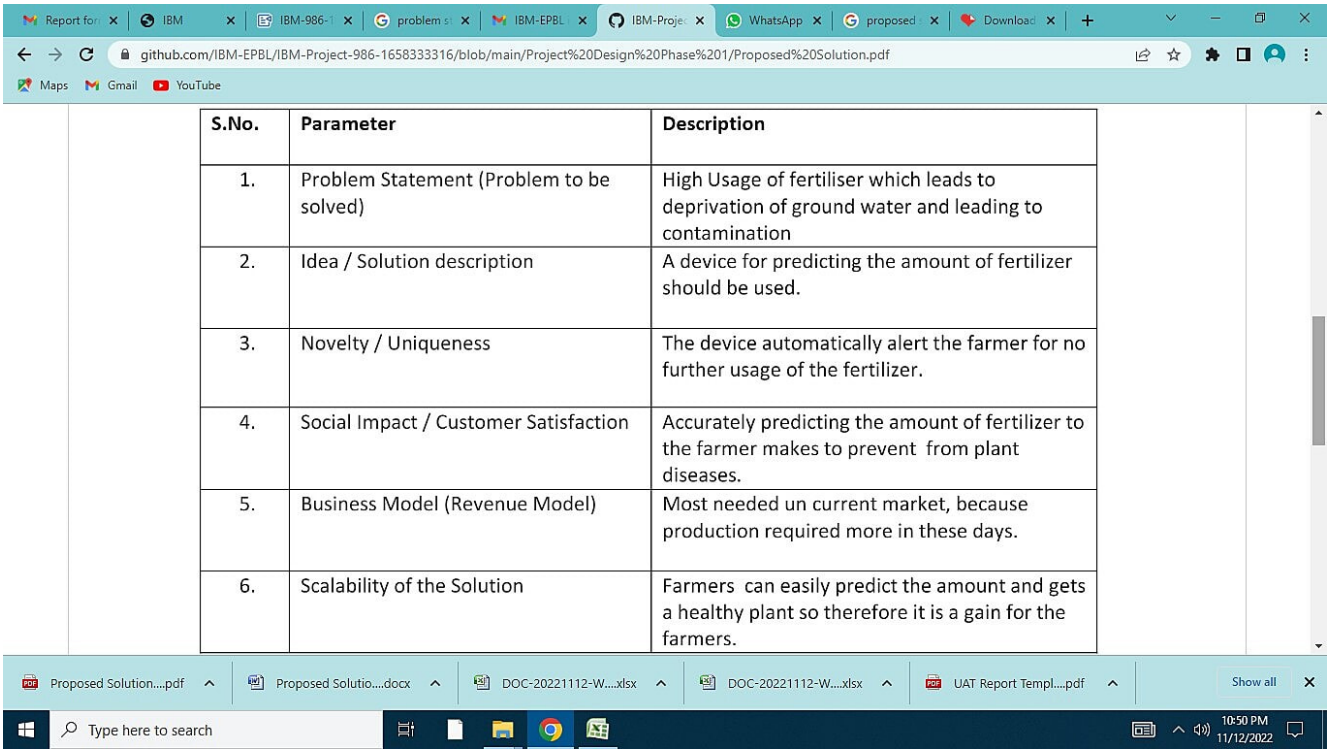
3.2 IDEATION AND PROPOSED SOLUTION:

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.



3.3 Proposed Solution:

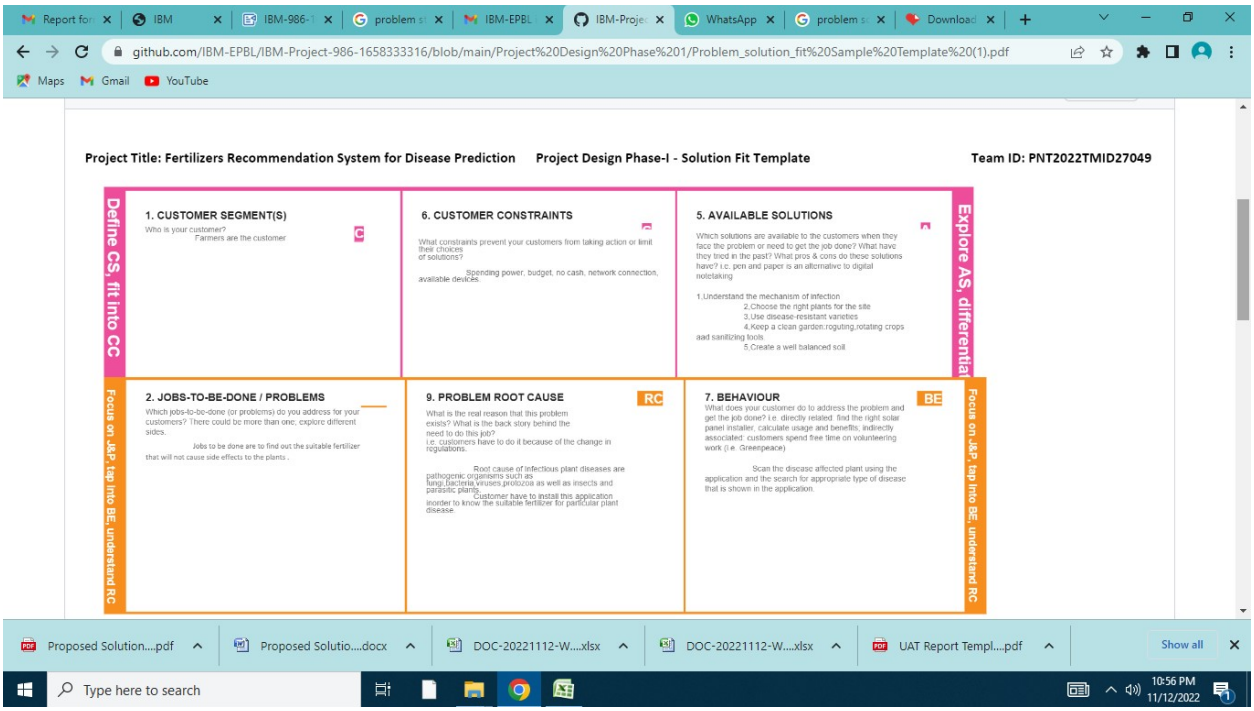
The proposed solution should relate the current situation to a desired result and describe the benefits that will accrue when the desired result is achieved. So, begin your proposed solution by briefly describing this desired result.

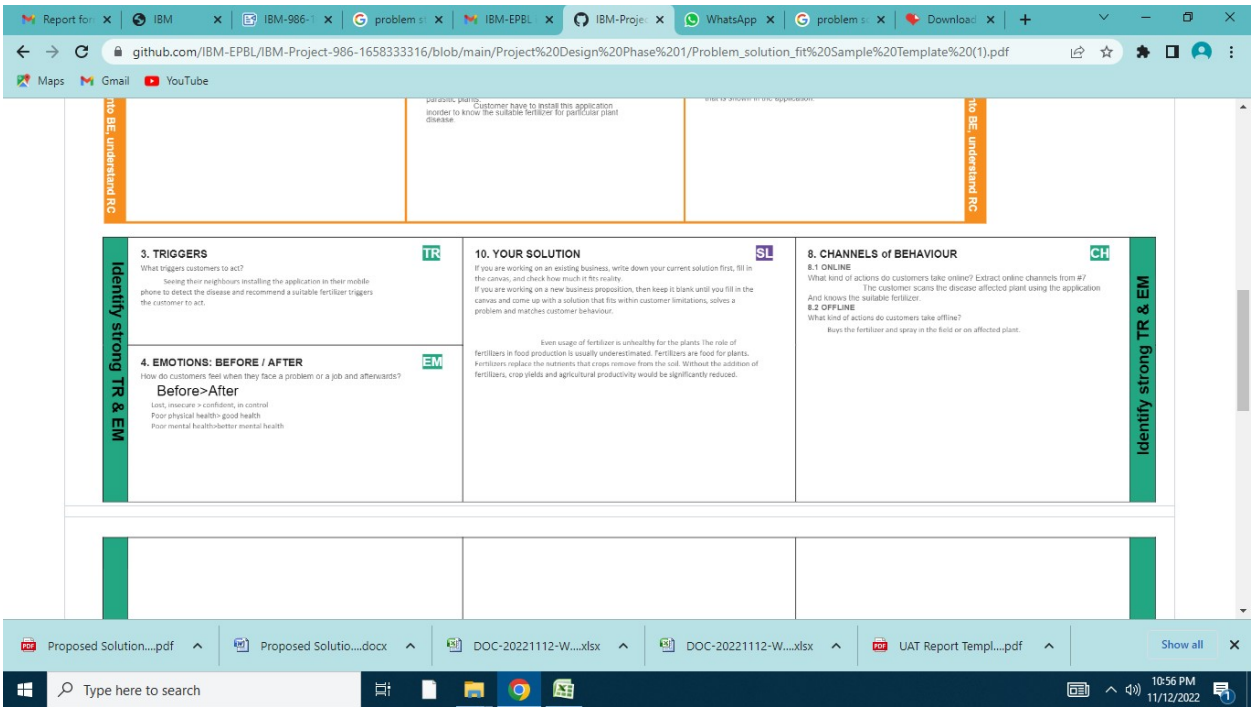


S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	High Usage of fertiliser which leads to deprivation of ground water and leading to contamination
2.	Idea / Solution description	A device for predicting the amount of fertilizer should be used.
3.	Novelty / Uniqueness	The device automatically alert the farmer for no further usage of the fertilizer.
4.	Social Impact / Customer Satisfaction	Accurately predicting the amount of fertilizer to the farmer makes to prevent from plant diseases.
5.	Business Model (Revenue Model)	Most needed un current market, because production required more in these days.
6.	Scalability of the Solution	Farmers can easily predict the amount and gets a healthy plant so therefore it is a gain for the farmers.

3.4 PROBLEM SOLUTION FIT:

Problem-Solution canvas is a tool for entrepreneurs, marketers and corporate innovators, which helps them identify solutions with higher chances for solution adoption, reduce time spent on solution testing and get a better overview of current situation.


github.com/IBM-EPBL/IBM-Project-986-1658333316/blob/main/Project%20Design%20Phase%201/Problem_solution_fit%20Sample%20Template%20(1).pdf
Project Title: Fertilizers Recommendation System for Disease Prediction
Project Design Phase-I - Solution Fit Template
Team ID: PNT2022TMID27049
Define CS, fit into CC
1. CUSTOMER SEGMENT(S)
Who is your customer?
Farmers are the customer
6. CUSTOMER CONSTRAINTS
What constraints prevent your customers from taking action or limit their choices of solutions?
Spending power, budget, no cash, network connection, available devices.
5. AVAILABLE SOLUTIONS
Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking
1.Understand the mechanism of infection
2.Choose the right plants for the site
3.Use disease-resistant varieties
4.Keep a clean garden:rogueing,rotating crops aad sanitizing tools
5.Create a well balanced soil
Explore AS, differentiat
Focus on J&P, tap into BE, understand RC
2. JOBS-TO-BE-DONE / PROBLEMS
Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one, explore different sides.
Jobs to be done are to find out the suitable fertilizer that will not cause side effects to the plants .
9. PROBLEM ROOT CAUSE
RC
What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.
Root cause of infectious plant diseases are pathogenic organisms such as fungi,bacteria,viruses,protozoa as well as insects and parasitic plants.
Customer have to install this application inorder to know the suitable fertilizer for particular plant disease.
7. BEHAVIOUR
BE
What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)
Scan the disease affected plant using the application and the search for appropriate type of disease that is shown in the application.
Focus on J&P, tap into BE, understand RC
Proposed Solution....pdf
Proposed Solutio....docx
DOC-20221112-W....xlsx
DOC-20221112-W....xlsx
UAT Report Templ....pdf
Show all
Type here to search
10:56 PM
11/12/2022

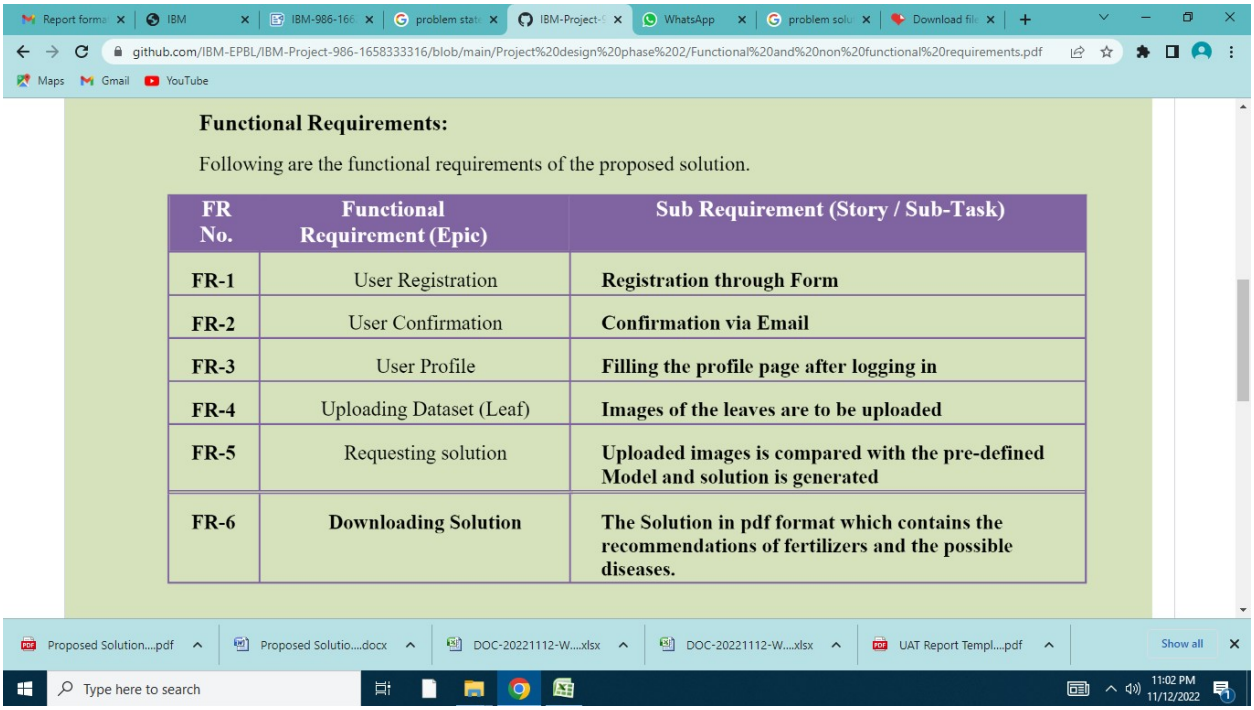

github.com/IBM-EPBL/IBM-Project-986-1658333316/blob/main/Project%20Design%20Phase%201/Problem_solution_fit%20Sample%20Template%20(1).pdf
parasitic plants.
Customer have to install this application inorder to know the suitable fertilizer for particular plant disease.
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
3. TRIGGERS
TR
What triggers customers to act?
Seeing their neighbours installing the application in their mobile phone to detect the disease and recommend a suitable fertilizer triggers the customer to act.
4. EMOTIONS: BEFORE / AFTER
EM
How do customers feel when they face a problem or a job and afterwards?
Before>After
Lost, insecure > confident, in control
Poor physical health> good health
Poor mental health>better mental health
10. YOUR SOLUTION
SL
If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.
Even usage of fertilizer is unhealthy for the plants The role of fertilizers in food production is usually underestimated. Fertilizers are food for plants. Fertilizers replace the nutrients that crops remove from the soil. Without the addition of fertilizers, crop yields and agricultural productivity would be significantly reduced.
8. CHANNELS of BEHAVIOUR
CH
8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7
The customer scans the disease affected plant using the application And knows the suitable fertilizer.
8.2 OFFLINE
What kind of actions do customers take offline?
Buys the fertilizer and spray in the field or on affected plant.
Identify strong TR & EM
Proposed Solution....pdf
Proposed Solutio....docx
DOC-20221112-W....xlsx
DOC-20221112-W....xlsx
UAT Report Templ....pdf
Show all
Type here to search
10:56 PM
11/12/2022

4.REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL REQUIREMENTS:

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases

.



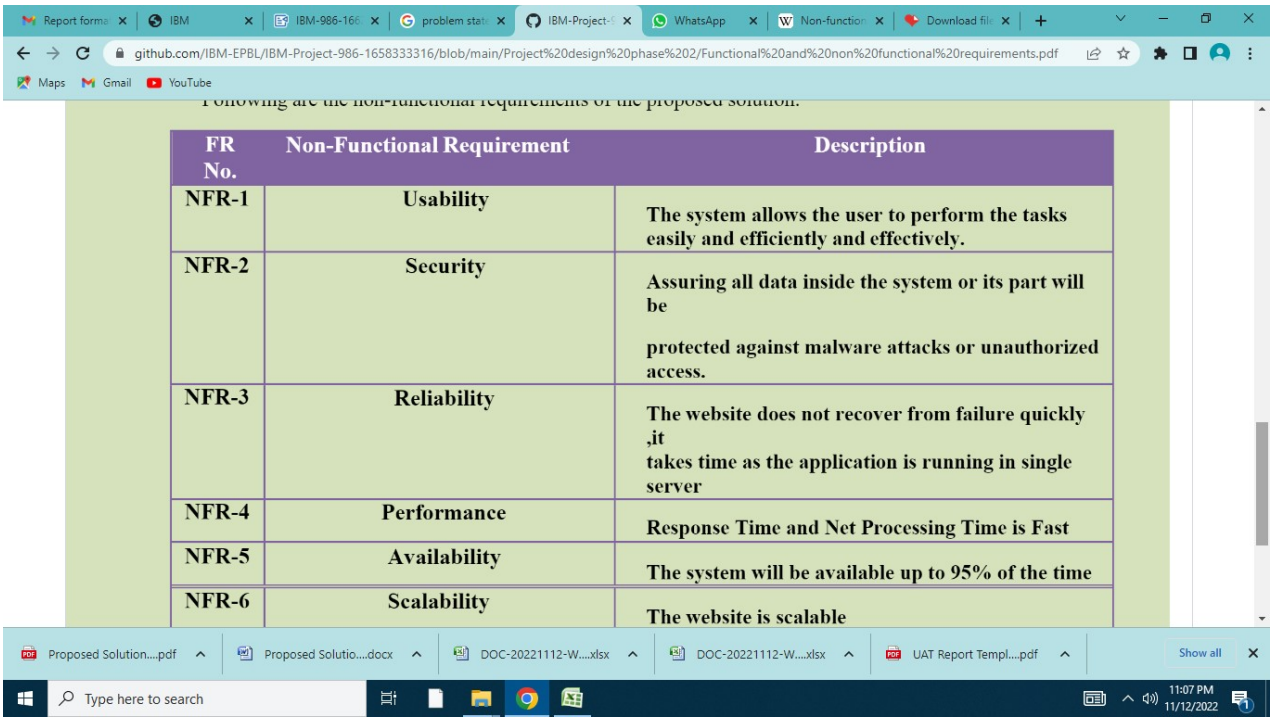
Functional Requirements:

Following are the functional requirements of the proposed solution.

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:

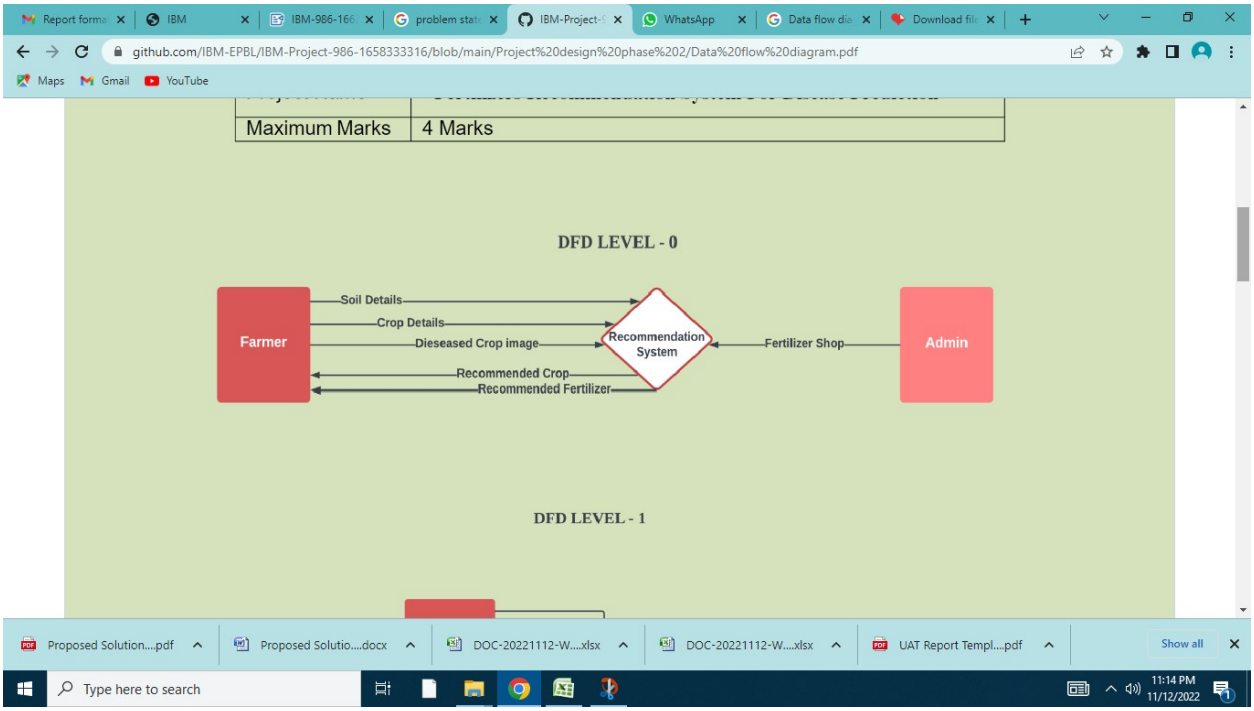
A Non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours.

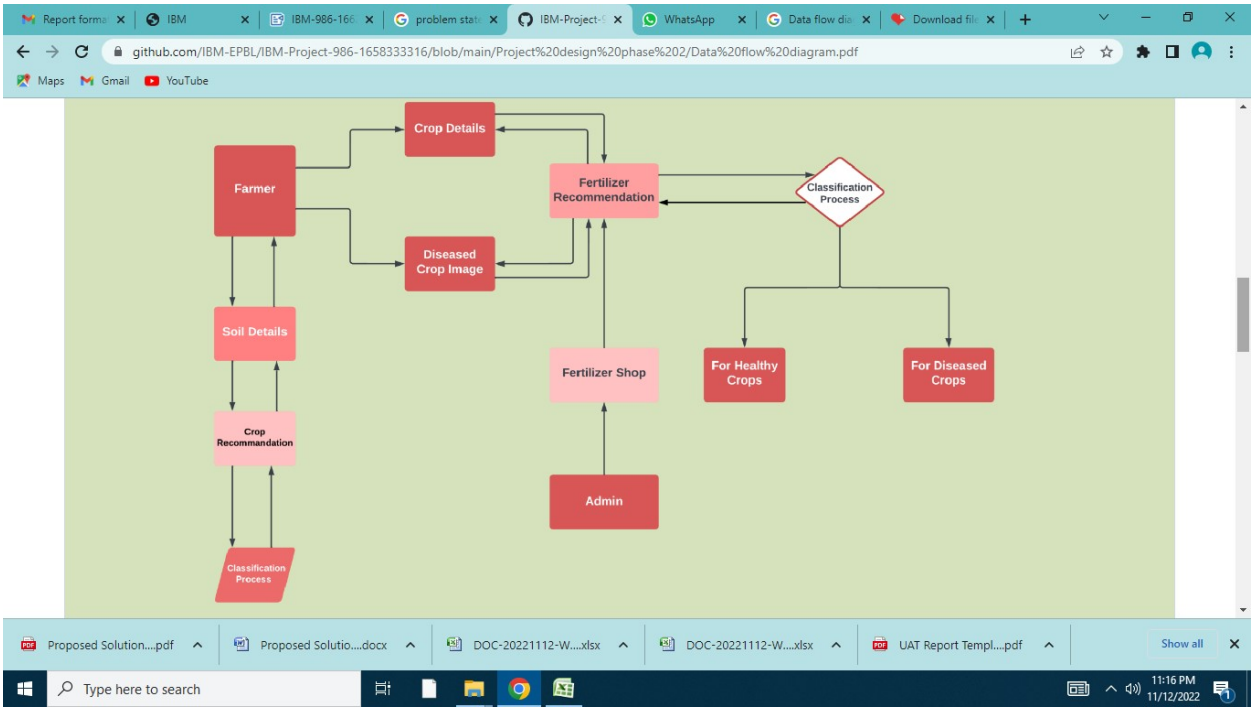


5.PROJECT DESIGN:

5.1 DATA FLOW DIAGRAMS:

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various subprocesses the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships.

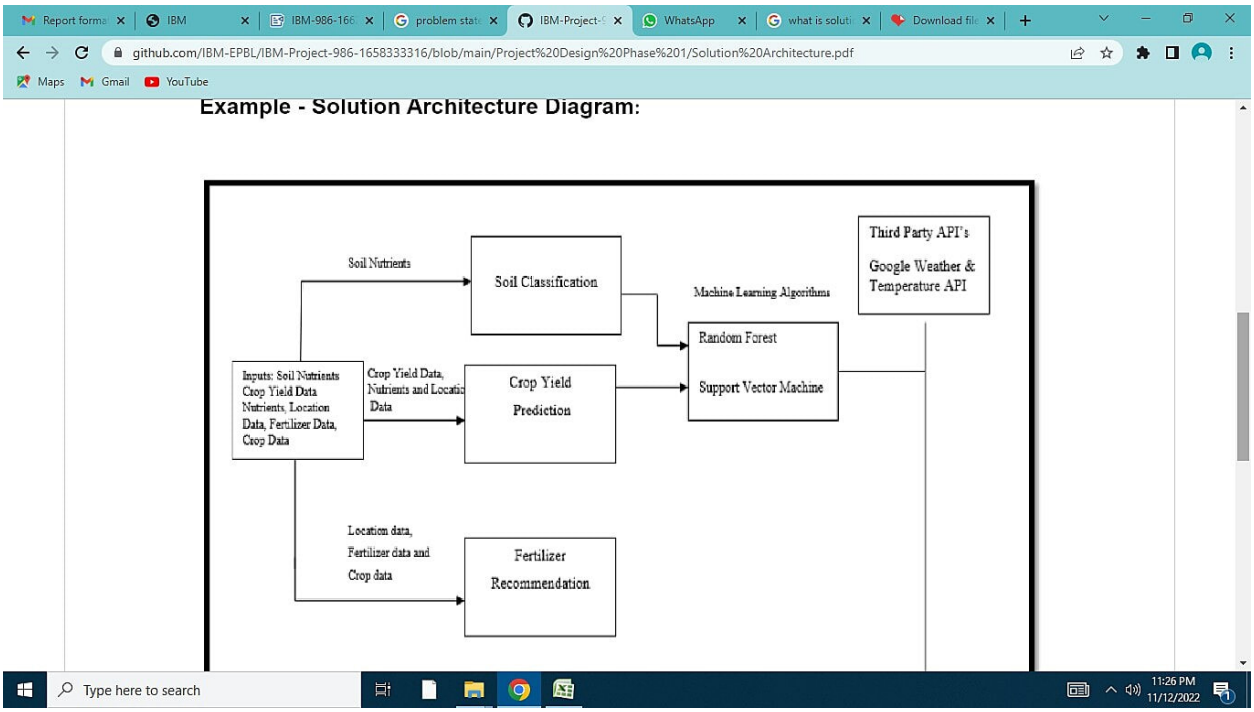


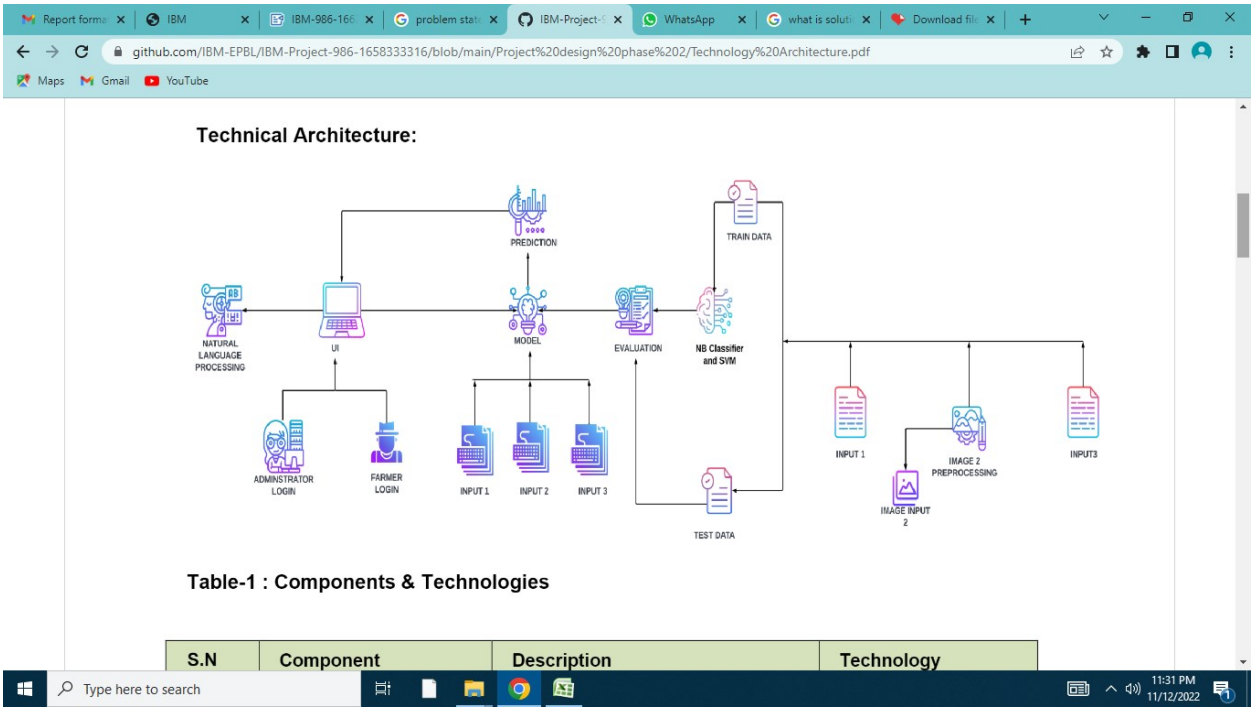


5.2 SOLUTION AND TECHNICAL ARCHITECTURE:

A solution architecture (SA) is antectural description archiof a specific solution. SAs combine guidance from different enterprise architecture viewpoints (business,

information and technical), as well as from the enterprise solution architecture (ESA).





5.3 USER STORIES:

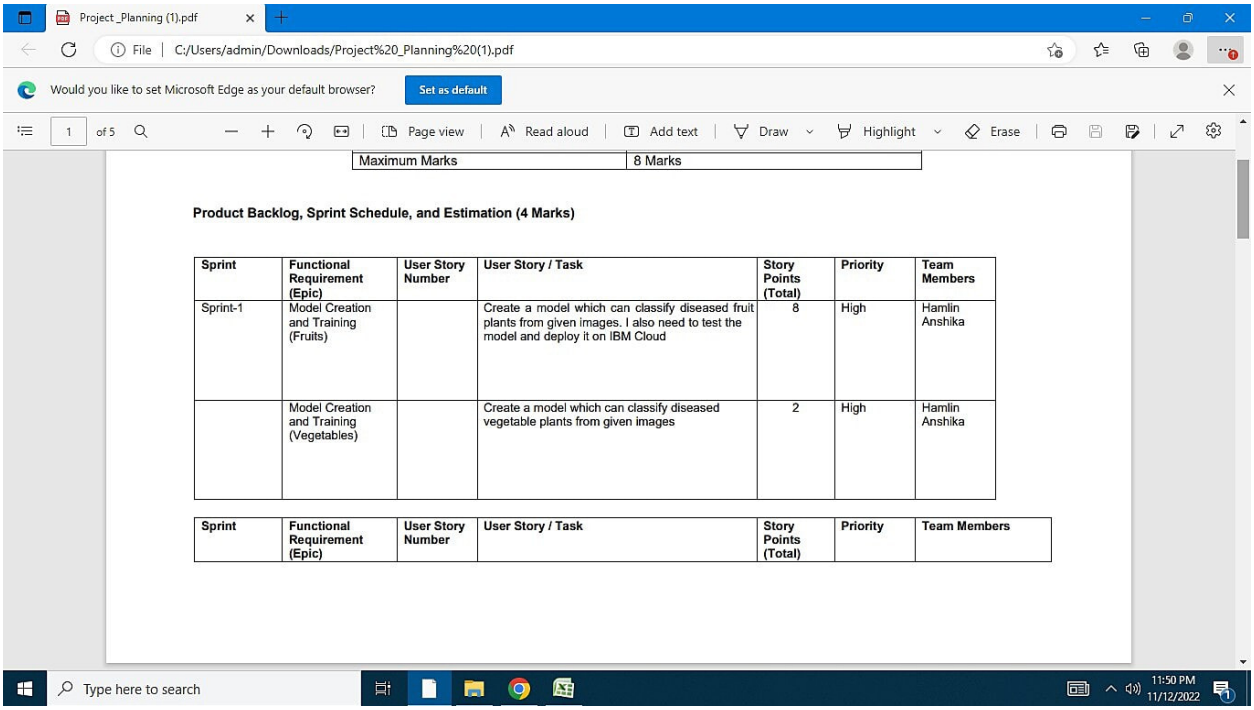
A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to

articulate how a piece of work will deliver a particular value back to the customer.

6.PROJET PLANNING AND SCHEDULING:

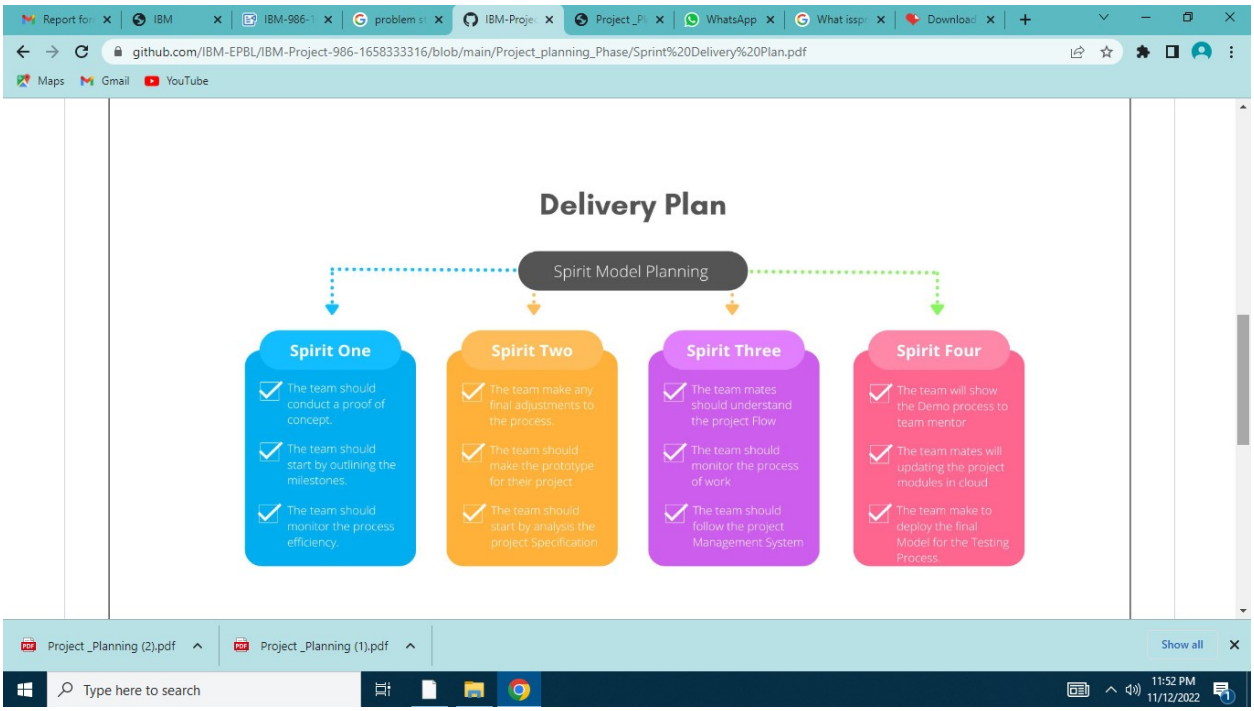
6.1 SPRINT PLANNING AND ESTIMATION:

The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.



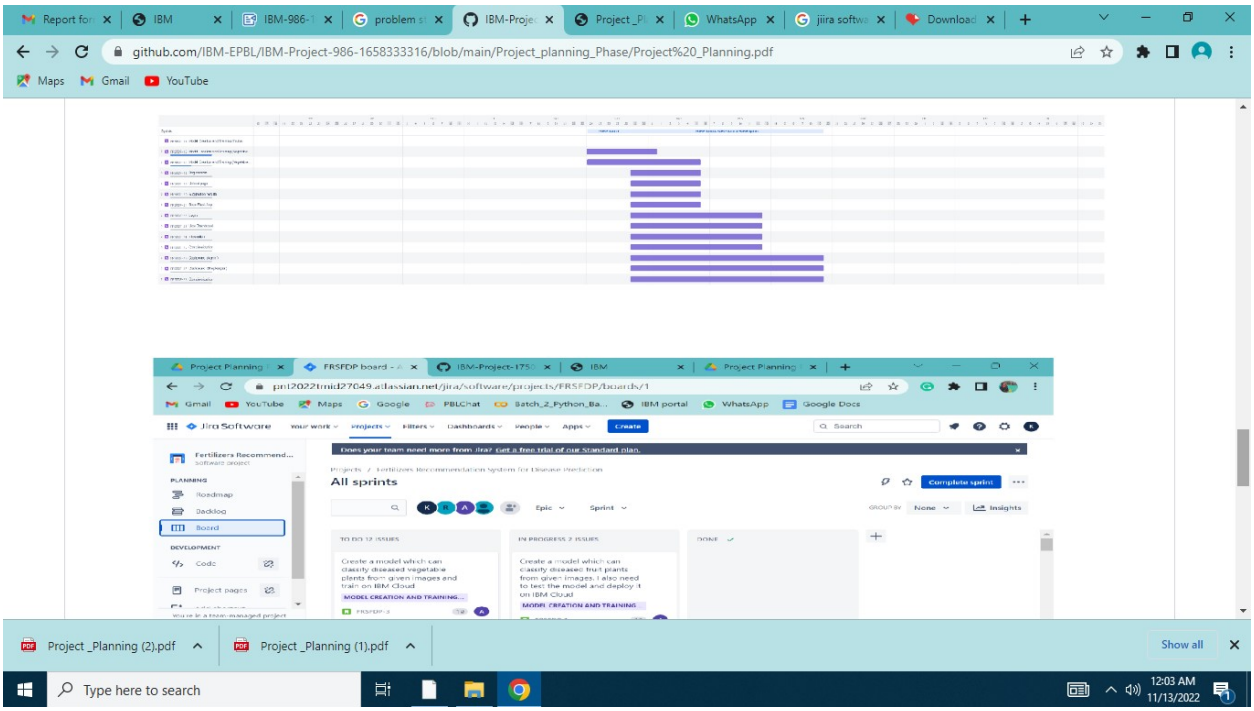
6.2 SPRINT DELIVERY SCHEDULE:

The Objectives of the project must have to be must be separated in forms of sprints and separated to all the team members accordingly.



6.3 REPORTS FROM JIIRA:

Jira Software is part of a family of products designed to help teams of all types manage work. Originally, Jira was designed as a bug and issue tracker. But today, Jira has evolved into a powerful work management tool for all kinds of use cases, from requirements and test case management to agile software development.



7.CODING AND SOLUTIONING:

7.1 PYTHON CODE:

```
# -*- 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/ibmdataset/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)
```

STEP 1: Build a flask application.
STEP 2: Initialise the flask app and load the model.
STEP 3:Configure the home page
STEP 4:Pre process the frame and run

7.2 HTML PAGE:

```
 <!DOCTYPE html>
<html>

<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1">
 <title> Plant Disease Prediction</title>
 <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet'
type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>
<style>
.header {
                                top:0;
                                margin:0px;
                                left: 0px;
                                right: 0px;
                                position: fixed;
                                background-color: #28272c;
                                color: white;
                                box-shadow: 0px 8px 4px grey;
                                overflow: hidden;
```

```
                        padding-left:20px;
                        font-family: 'Josefin Sans';
                        font-size: 2vw;
                        width: 100%;
                        height:8%;
                        text-align: center;
        }
        .topnav {
overflow: hidden;
 background-color: #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%;
```

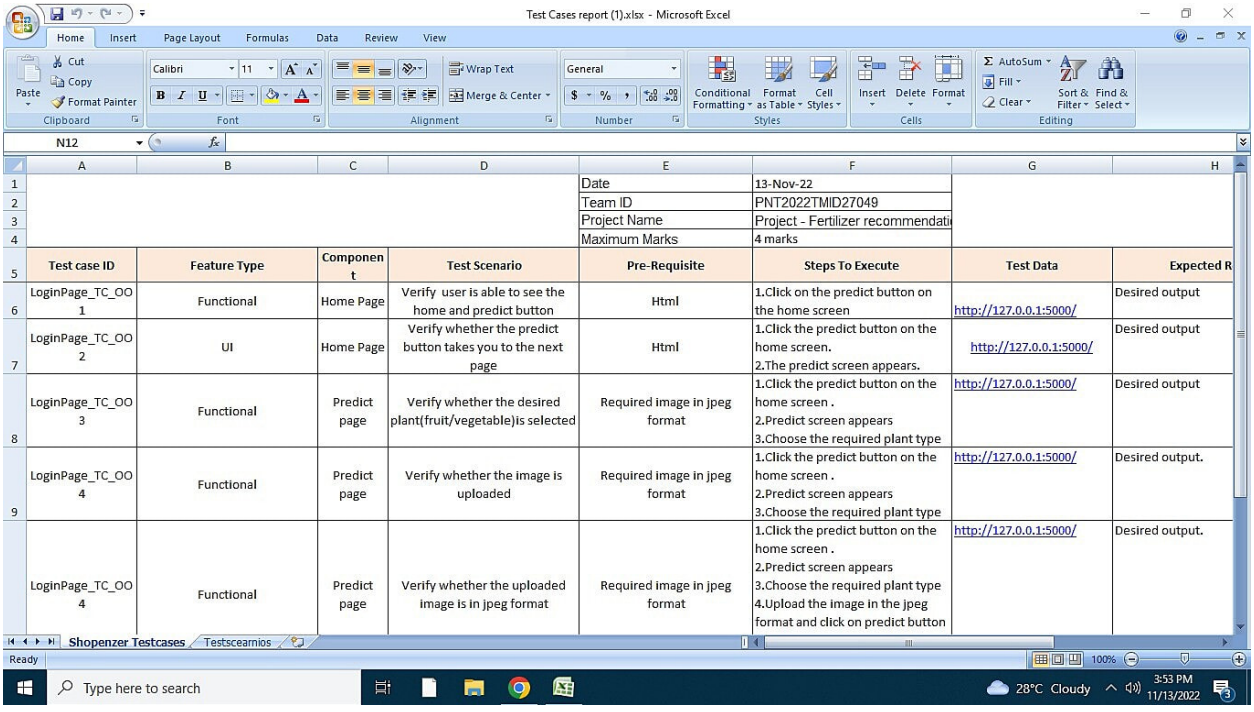
```
display: inline-block;
 transition: background-color 0.6s ease;
}

.active {
 background-color: #717171;
 }

/* Fading animation */
 .fade { -webkit-animation-name: fade;
 -webkit-animation-duration: 1.5s;
 animation-name: fade;
 animation-duration: 1.5s;
 }

 @-webkit-keyframes fade {
from {opacity: .4}
to {opacity: 1}
}
@keyframes fade {
from {opacity: .4}
to {opacity: 1}
 }

 /* On smaller screens, decrease text size */
 @media only screen and (max-width: 300px) {
 .text {font-size: 11px}
}
</style>
</head>
<body style="font-family:'Times New Roman', Times, serif;background-color:#C2C5A8;">
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%">Plant
Disease Prediction</div>
 <div class="topnav-right"style="padding-top:0.5%;">

 <a class="active" href="{{ url_for('home')}}">Home</a>
 <a href="{{ url_for('prediction')}}">Predict</a>
 </div>
</div>
```

```
<div style="background-color:#ffffff;">
<div style="width:60%;float:left;">
<div style="font-size:50px;font-family:Montserrat;padding-left:20px;text-
align:center;paddingtop:10%;">
<b>Detect if your plant<br> is infected!!</b></div><br>
<div style="font-size:20px;font-family: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>
</div>
</div>
<div style="width:40%;float:right;"><br><br>
<img src="{{url_for('static',filename='images/12456.png')}}" style="max-
height:100%;maxwidth:100%;">
</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", "");
 }
 slides[slideIndex-1].style.display = "block";
 dots[slideIndex-1].className += " active";
 setTimeout(showSlides, 2000); // Change image every 2 seconds
}
</script>
</body>
</html>
```

8.TESTING:

8.1 TEST CASES:

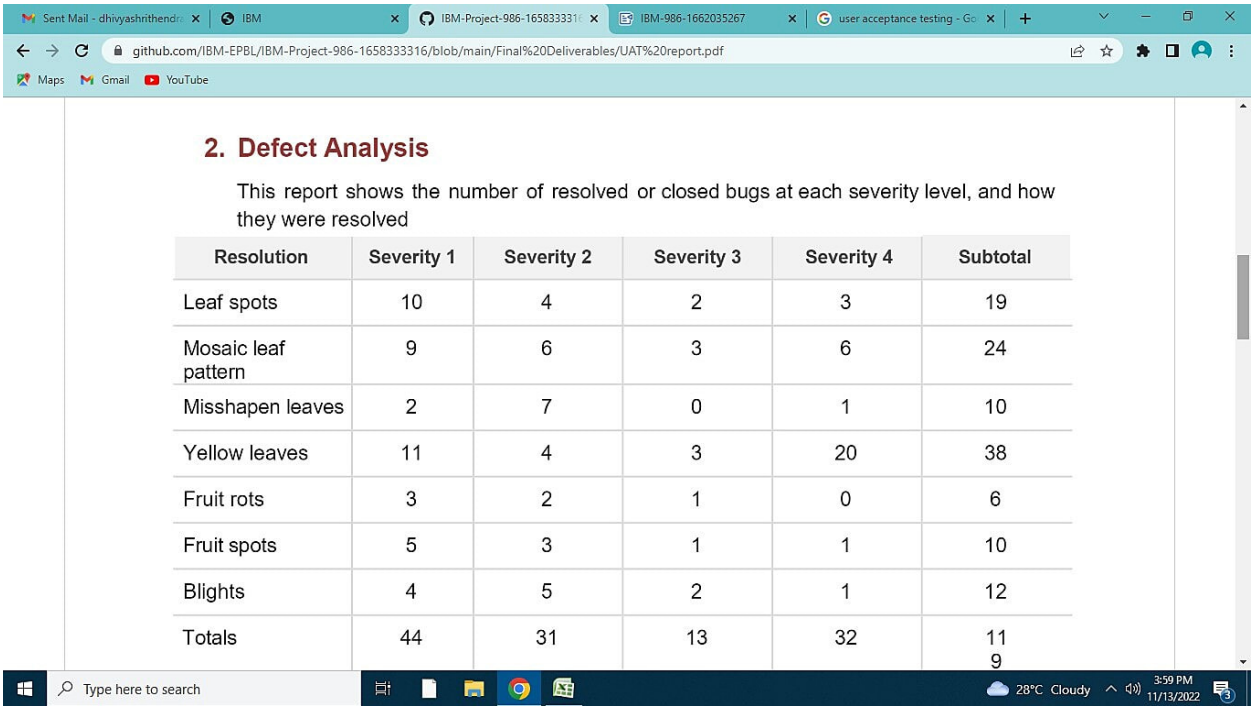


				Date	13-Nov-22		
				Team ID	PNT2022TMID27049		
				Project Name	Project - Fertilizer recommendati		
				Maximum Marks	4 marks		
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected R
LoginPage_TC_OO 1	Functional	Home Page	Verify user is able to see the home and predict button	Html	1.Click on the predict button on the home screen	http://127.0.0.1:5000/	Desired output
LoginPage_TC_OO 2	UI	Home Page	Verify whether the predict button takes you to the next page	Html	1.Click the predict button on the home screen. 2.The predict screen appears.	http://127.0.0.1:5000/	Desired output
LoginPage_TC_OO 3	Functional	Predict page	Verify whether the desired plant(fruit/vegetable)is selected	Required image in jpeg format	1.Click the predict button on the home screen . 2.Predict screen appears 3.Choose the required plant type	http://127.0.0.1:5000/	Desired output
LoginPage_TC_OO 4	Functional	Predict page	Verify whether the image is uploaded	Required image in jpeg format	1.Click the predict button on the home screen . 2.Predict screen appears 3.Choose the required plant type	http://127.0.0.1:5000/	Desired output.
LoginPage_TC_OO 4	Functional	Predict page	Verify whether the uploaded image is in jpeg format	Required image in jpeg format	1.Click the predict button on the home screen . 2.Predict screen appears 3.Choose the required plant type 4.Upload the image in the jpeg format and click on predict button	http://127.0.0.1:5000/	Desired output.

8.2 USER ACCEPTANCE TESTING:

User Acceptance Testing (UAT), which is performed on most UIT projects, sometimes called beta testing or end-user testing, is a phase of software

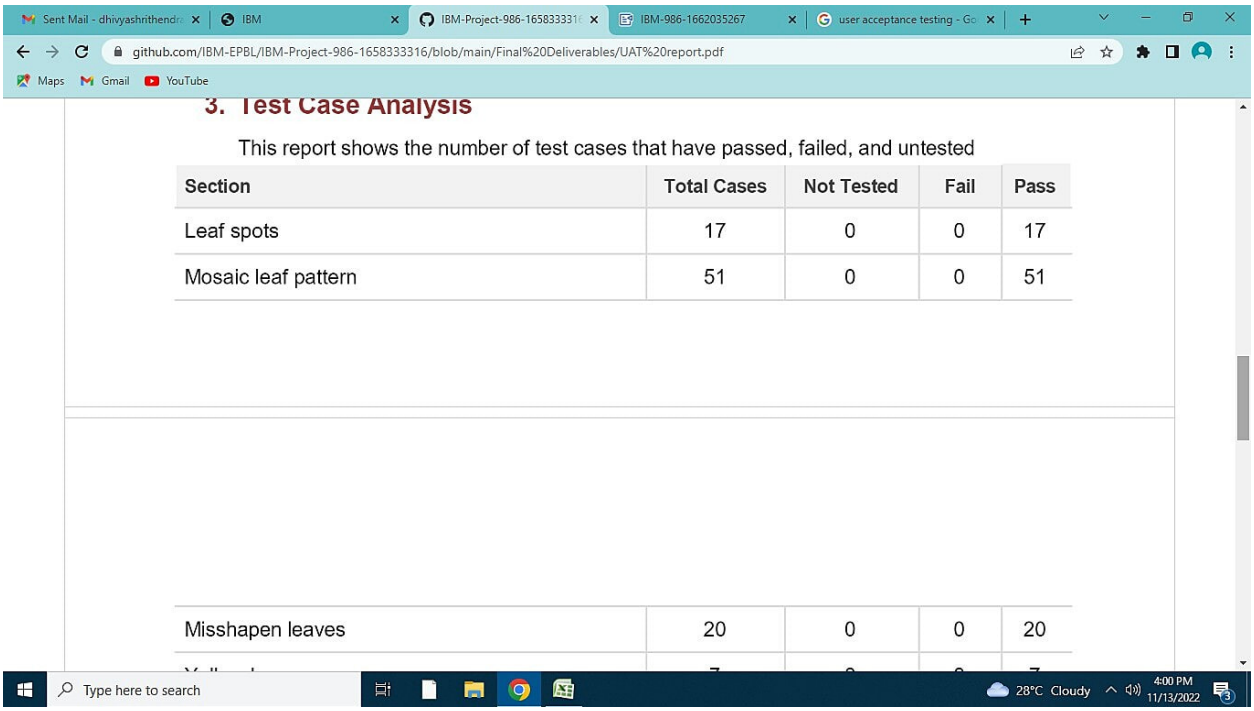
development in which the software is tested in the "real world" by the intended audience or business representative.



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

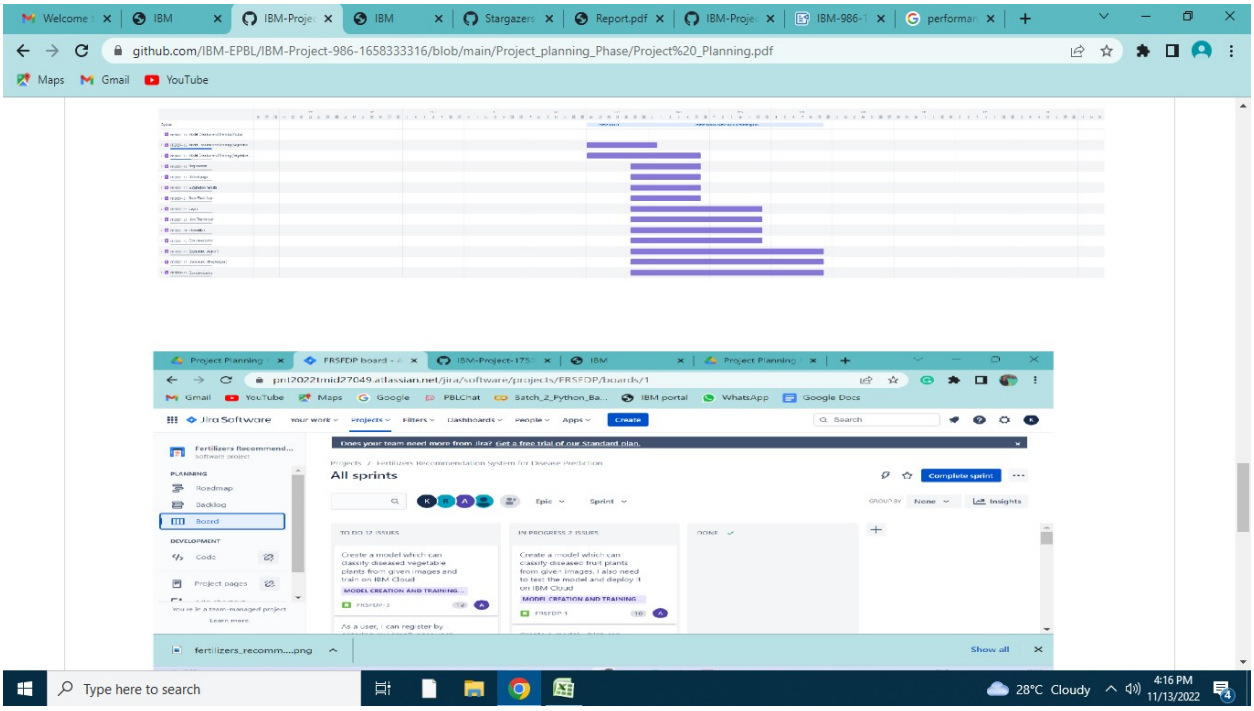
This report shows the number of test cases that have passed, failed, and untested

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

9.RESULT:

9.1 PERFORMANCE METRICS:

Performance metrics are defined as figures and data representative of an organization's actions, abilities, and overall quality.



10.ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

1.Fertilizers provide crops with nutrients like potassium, phosphorus, and nitrogen, which allow crops to grow bigger, faster, and to produce more food. Nitrogen in particular is an essential nutrient for the growth of every organism on Earth. Nitrogen is all around us and makes up about 78% of the air you breathe.

2.Sometimes plants need a quick fix to survive, in this type of cases fertilizers play a vital role to improve plants' health. plants need nutrients that can be absorbed quickly which is fulfilled by fertilizers. They are easily soluble and fastly absorbed by plants and as soon as possible it helps to regain and boost plant health.

3.As the population is increasing, there is a huge demand for food, so good yield is required to fulfill the demand. Here fertilizers become helpful for the good production of crops due to their numerous benefits which promote the fast and healthy growth of plants. For large production, fertilizers become compulsory.

DISADVANTAGES:

1.Fertilizers are man-made so they need production in factories which makes them costlier than naturally made manure. But it is important for plant nutrients so it is in demand and thus it has high value.

2.Fertilizers are used in moderate quantities if we use excessive fertilizers it surely damages the roots of plants and their tissues and thus plants can die. fertilizers are used according to the need of the plant. Unnecessary use of them can affect the plant's health specially if plants have good fertile soil.

3.There are many types of fertilizers in the market, some of them are chemically made. These chemical fertilizers are harmful to humans and plants also. Skin irritation, respiratory problems commonly occur due to fertilizers. Can pass harmful chemical in our food which affects.

4.Fertilizers can reduce the quality of soil and can harm microorganisms in the soil. Long-term use disturbs the pH of the soil and also reduces the microbial activities which are naturally good for plants.

11.CONCLUSION:

The authors proposed a new approach for the soil based fertilizer prediction system. The proposed system was able to analyze the soil nutrient type efficiently, kind of leaf disease present in the crop and predict the fertilizer in a proficient manner. The approach was flexible, and can be extended to the needs of the users in a better manner. The proposed method was carried out with five different crops.

12.FUTURE SCOPE:

Different approaches and models of Deep Learning methods were explored and used in this project so that it can detect and classify plant diseases correctly through image processing of leaves of the plants. The procedure starts from collecting the images used for training, testing and validation to image preprocessing and augmentation and finally comparison of different pretrained models over their accuracy. Finally, at the end , our model detects and distinguishes between a healthy plant and different diseases and provides suitable remedies so as to cure the disease.This paper proposed and developed a system which uses plant leaf images to detect different types of disease in tomato crops, and also provides appropriate fertilizer suggestions.

13.APPENDIX:

The Project deliverables are uploaded in Git repository and in the IBM dashboard.

GIT LINK:https://github.com/IBM-EPBL/IBM-Project-986-1658333316

DEMOLINK:https://drive.google.com/drive/folders/1hmuNNe6PwEA3DGM60TrG-6SERXdRYoAU

SOURCE CODE:

```
import requests

from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
```

```
import tensorflow as tf
from flask import Flask, request, render_template, redirect, url_for
import os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session

app = Flask(__name__)

#load both the vegetable and fruit models
model = load_model("vegetable.h5")
model1=load_model("fruit.h5")
#home page
@app.route('/')
def home():
        return render_template('home.html')

#prediction page
@app.route('/prediction')
def prediction():
return render_template('predict.html')

@app.route('/predict',methods=['POST'])
def predict():
if request.method == 'POST':
        # Get the file from post request
        f = request.files['image']

# Save the file to ./uploads
basepath = os.path.dirname(__file__)
file_path = os.path.join(
        basepath, 'uploads', secure_filename(f.filename))
f.save(file_path)
img = image.load_img(file_path, target_size=(128, 128))

x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
```

```
plant=request.form['plant']
print(plant)
if(plant=="vegetable"):
        preds = model.predict(x)
        preds=np.argmax(preds)
        print(preds)
        df=pd.read_excel(r'precautions - veg.xlsx',engine='openpyxl')
        print(df.iloc[preds]['caution'])
else:
        preds = model1.predict(x)
        preds=np.argmax(preds)
        print(preds)
        df=pd.read_excel(r'precautions - fruits.xlsx',engine='openpyxl')
        print(df.iloc[preds]['caution'])

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

if __name__ == "__main__":
          app.run(debug=False)
```