FERTILIZERS RECOMMENDATION SYSTEM FOR DISEASE PREDICTION
TEAM ID PNT2022TMID33888

NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READLINESS FOR INNOVATION EMPLOYNMENT AND ENTERPRENEURSHIP

A PROJECT REPORT
SUBMITTED BY

JANANI M	950819106020
RESHMI FIONA T	950819106060
VANITHA S	950819106102
VIJAY S	950819106105

BACHELOR OF ENGINERING IN ELECTRONICS AND COMMUNICATION ENGINEERING
GOVERNMENT COLLEGE OF ENGINEERING -TIRUNELVELI-627007

INDEX

INTRODUCTION

1.1 PROJECT OVERVIEW

- Fertilizer Recommentation system for disease Prediction is a simple ML and DL based website which recommends the best crop to grow, fertilizers to use and the diseases caught by your crops.

1.2 PURPOSE

The purpose of this model is to predict the plant diseases and recommend the appropriate solution for it.

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Abstract: Agriculture is the main aspect of country development. Plant disease on leaves is one of the major factors of reductions in both quality and quantity of the food crops. Finding the leaf disease is an important role of agriculture preservation. After preprocessing using a median filter, segmentation is done by Guided Active Contour method and finally, the leaf disease is identified by using Support Vector Machine. The disease-based similarity measure is used for fertilizer recommendation. INTRODUCTION: Detection and recognition of plant diseases using machine learning are very efficient in providing symptoms of identifying diseases at its earliest. Plant pathologists can analyze the digital images using digital image processing for diagnosis of plant diseases. Different symptoms and diseases of leaves are predicted by different methods in image processing. These different methods include different fundamental processes ● segmentation ● feature extraction ● classification Mostly, the prediction and diagnosis of leaf diseases are dependent on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves. MATERIAL AND METHODS: A digital camera or similar devices are used to take images of different types, and then those are used to identify the affected area in leaves. Image Classification Steps : ● Image acquisition: To get the image of a leaf to evaluate in the direction of a class. ● Preprocessing: The purpose of image preprocessing is improving image statistics. The preprocessing receives an image as input and generates an output image as a grayscale,an invert and a smoothed one. ● Segmentation: Implements Guided active contour method. Unconstrained active contours applied to the difficult natural images. Dealing with unsatisfying contours, which would try and make their way through every possible grab cut in the border of the leaf. ● Disease Prediction: Leaves are affected by bacteria, fungi, virus, and other insects. Support Vector Machine (SVM) algorithm classifies the leaf image as normal or affected based on leaf features such as color, shape, textures. Then a hyperplane was constructed to categorize the pre-processed leaves and also implement a multiclass classifier, to predict diseases in leaf image with improved accuracy. ● Fertilizer Recommendation: Recommend the fertilizer for affected leaves based on severity level. Fertilizers may be organic or inorganic. Admin can store the fertilizers based on disease categorization with severity levels. The measurements of fertilizers suggested based on disease severity.

2.2 REFERENCE

1.Fertilizers Recommendation System For Disease Prediction In Tree Leave R. Neela, P. Nithya

2.Plant Disease Detection and Fertilizer Suggestion Authors: Apurva Save, Aksham Gupta, Sarthak Pruthi, Divyanjana Nikam, Prof. Dr. Shilpa Paygude

3.Soil Based Fertilizer Recommendation System for Crop Disease Prediction System Dr.P. Pandi Selvi, P. Poornima

4.A Recommended System for Crop Disease Detection and Yield Prediction Using Machine Learning Approach Pooja Akulwar

5. CNN based Leaf Disease Identification and Remedy Recommendation System V Suma, R Amog Shetty, Rishab F Tated, Sanku Rohan, Triveni S Pujar

6.KRISHI RAKSHAN - A Machine Learning based New Recommendation System to the Farmer D. N. V. S. L. S. Indira; M. Sobhana; A. H. L. Swaroop; V Phani Kumar

7. Recommendation System for Agriculture Using Machine Learning and Deep Learning K. SuriyaKrishnaan, L. Charan Kumar & R. Vignesh

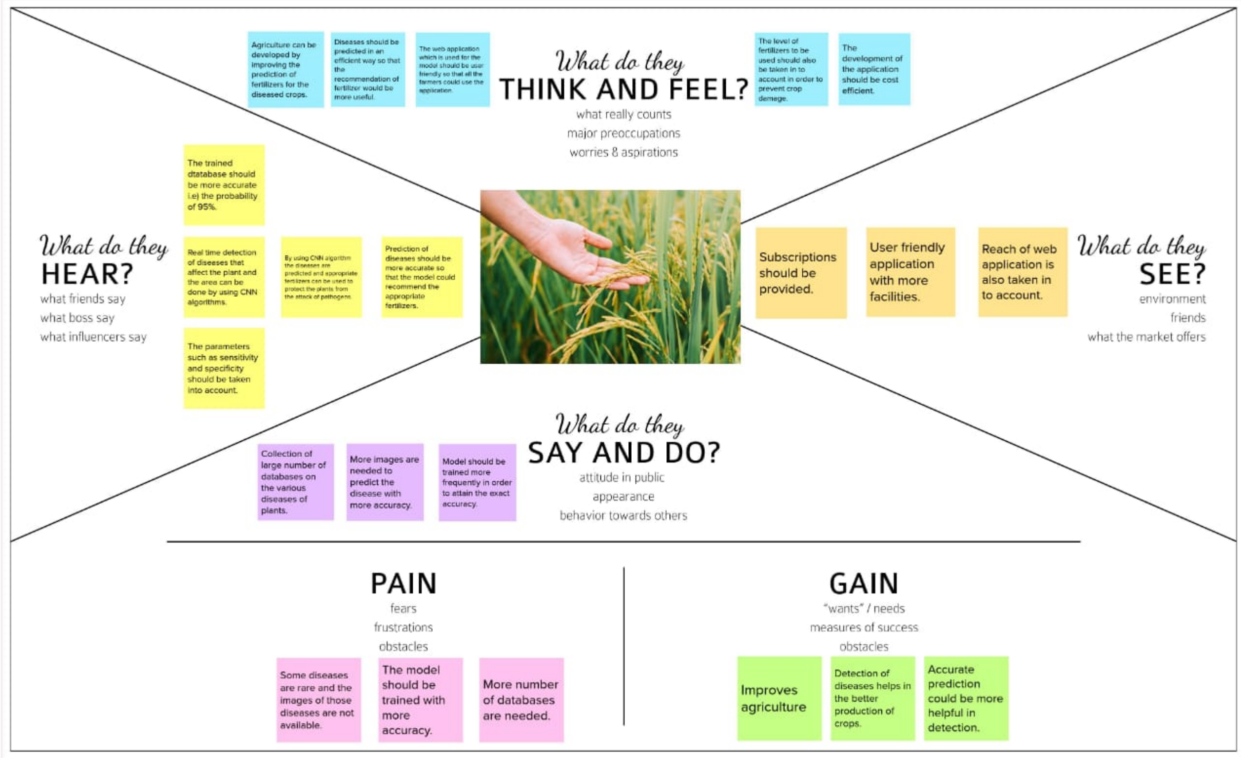
8.Plant Disease Detection and Crop Recommendation Using CNN and Machine Learning Raj Kumar, Neha Shukla

2.3 PROBLEM STATEMENT DEFINATION

- *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 AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

Agriculture can be developed by improving the prediction of fertilizers for the diseased crops.

Diseases should be predicted in an efficient way so that the recommendation of fertilizer would be more useful.

The web application which is used for the model should be user friendly so that all the farmers could use the application.

The level of fertilizers to be used should also be taken in to account in order to prevent crop demage.

The development of the application should be cost efficient.

What do they HEAR?
what friends say
what boss say
what influencers say

The trained dtatabase should be more accurate i.e) the probability of 95%.

Real time detection of diseases that affect the plant and the area can be done by using CNN algorithms.

By using CNN algorithm the diseases are predicted and appropriate fertilizers can be used to protect the plants from the attack of pathogens.

Prediction of diseases should be more accurate so that the model could recommend the appropriate fertilizers.

The parameters such as sensitivity and specificity should be taken into account.

What do they SEE?
environment
friends
what the market offers

Subscriptions should be provided.

User friendly application with more facilities.

Reach of web application is also taken in to account.

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

Collection of large number of databases on the various diseases of plants.

More images are needed to predict the disease with more accuracy.

Model should be trained more frequently in order to attain the exact accuracy.

PAIN
fears
frustrations
obstacles

Some diseases are rare and the images of those diseases are not available.

The model should be trained with more accuracy.

More number of databases are needed.

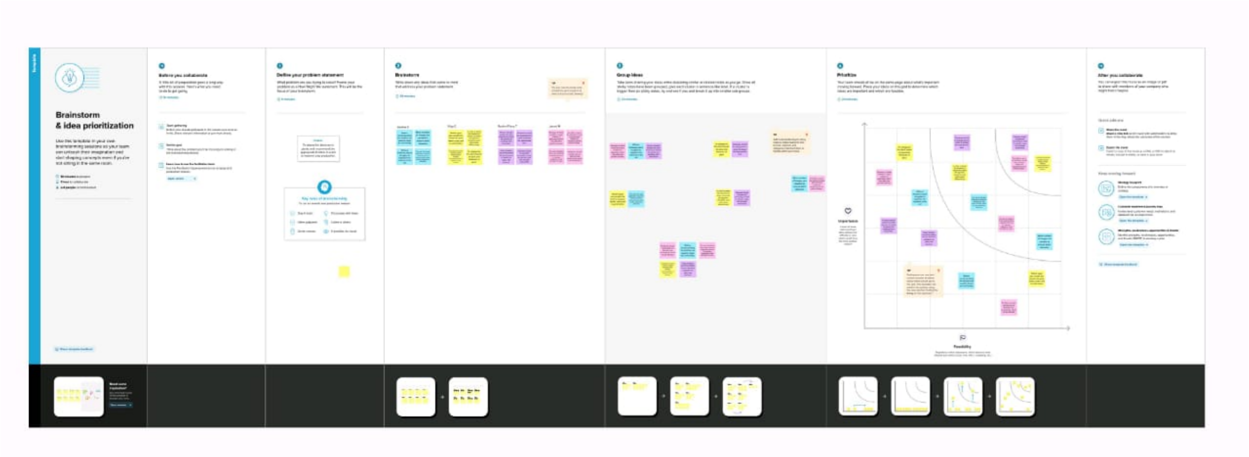
GAIN
"wants" / needs
measures of success
obstacles

Improves agriculture

Detection of diseases helps in the better production of crops.

Accurate prediction could be more helpful in detection.

3.2 IDEATION & BRAINSTORMING

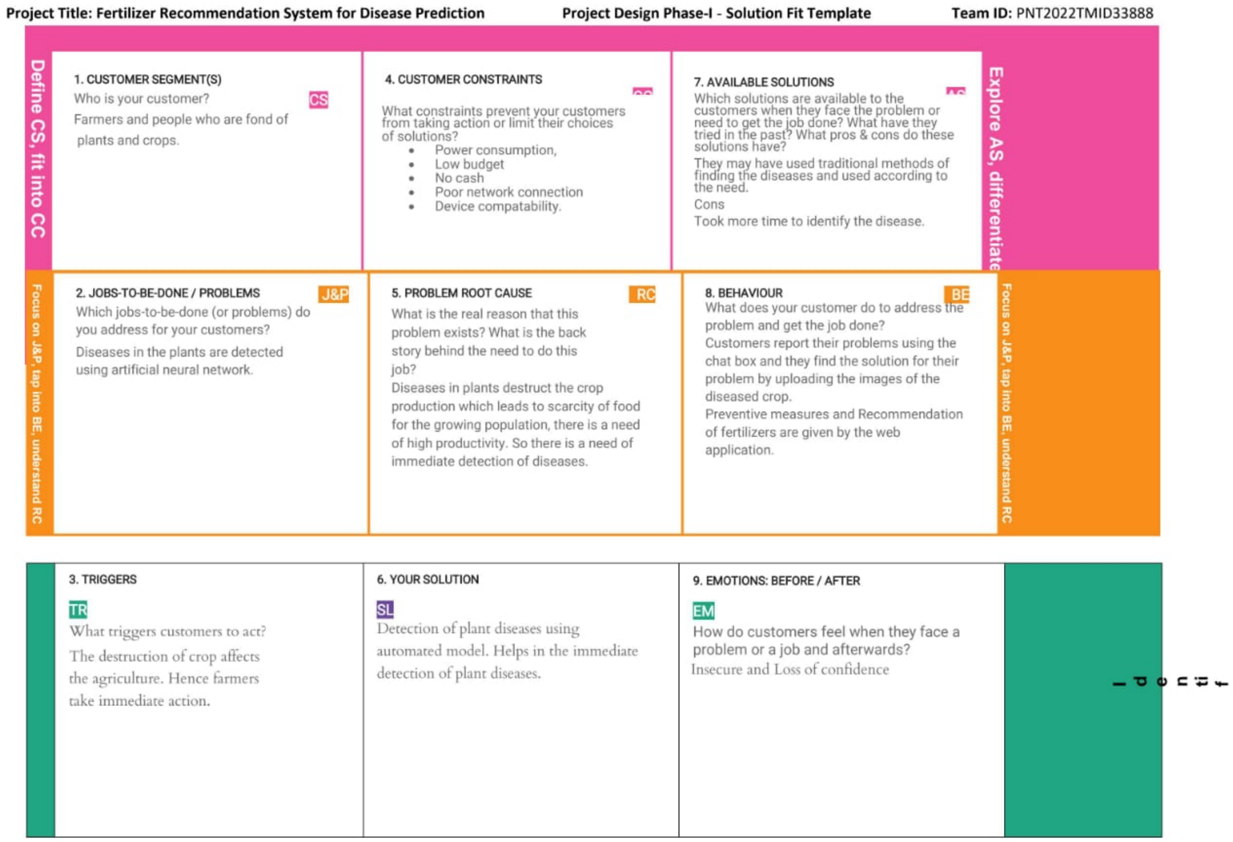


3.3 PROPOSED SOLUTION

S.NO	PARAMETER	DESCRIPTION
1	Problem statement (problem to be solved)	To detect the diseases in the plants at the early stages caused due to several pests and micro organisms and recommand fertilizers to prevent the crops from destruction
2	Idea /solution description	Categorizing the dieases of plant based on the pest. Dataset and images of the plant disease are compared with a existing images using CNN algorthim.
3	novelty / uniqueness	Automatic detection of plant diseases using deep learning models.
4	Social Impact / Customer Satisfaction	useful for farmers of detect the exact diseases of plants at the early stages and taking steps to treat the plants based on theri diseases.
5	Bussiness Model (Revenue model)	web application should be accessed by almost all farmers all over the

		world.Web application should be at low cost.
6.	Scalability of the solution	ased on the previous dataset the model can guess new diseases.

3.4 PROBLEM SOLUTION


Define CS, fit into CC
Focus on J&P, tap into BE, understand RC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Identif

Project Title: Fertilizer Recommendation System for Disease Prediction

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID33888

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

Farmers and people who are fond of plants and crops.

4. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

- Power consumption,
- Low budget
- No cash
- Poor network connection
- Device compatability.

7. AVAILABLE SOLUTIONS AS

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?

They may have used traditional methods of finding the diseases and used according to the need.

Cons

Took more time to identify the disease.

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers?

Diseases in the plants are detected using artificial neural network.

5. 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?

Diseases in plants destruct the crop production which leads to scarcity of food for the growing population, there is a need of high productivity. So there is a need of immediate detection of diseases.

8. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

Customers report their problems using the chat box and they find the solution for their problem by uploading the images of the diseased crop.

Preventive measures and Recommendation of fertilizers are given by the web application.

3. TRIGGERS

TR

What triggers customers to act?

The destruction of crop affects the agriculture. Hence farmers take immediate action.

6. YOUR SOLUTION

SL

Detection of plant diseases using automated model. Helps in the immediate detection of plant diseases.

9. EMOTIONS: BEFORE / AFTER

EM

How do customers feel when they face a problem or a job and afterwards?

Insecure and Loss of confidence

REQULREMENT ANALYSIS

FUNCTIONAL REQUIREMENTS

FR NO.	FUNCTIONAL REQUIREMENT(EPIC)	SUB REUIREMENT (STORY / SUB-TASK)
FR-1	User Registration	Registration through Form
FR-2	User Confirmation	Confirmation via Email
FR-3	User Profile	Filling the profile after logging in
FR-4	Uploading	The uploding images of the

		leaves are compared with the model that is pre-define and the solution is generated.
FR-5	Downloading the solution	The solution is downloaded in the pdf format which contains the recommended fertilizers and the possible diseases of the diseasedplant

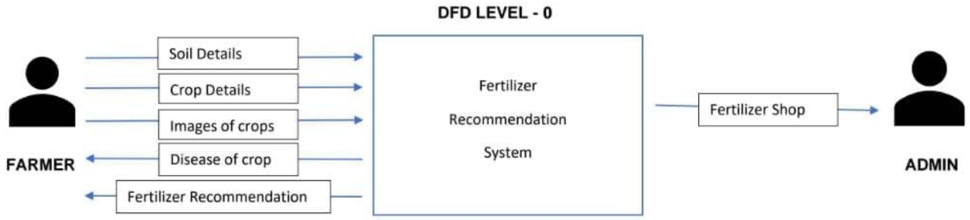
4.2 NON-FUNCTIONAL REQUIREMENTS

FR NO	NON-FUNCTIONAL REQUIREMENT	DESCRIPTION
NFR-1	Usability	The user could perform tasks very easily, efficiently and effectively .They could use this application in a user friendly manner
NFR-2	Security	All the data inside the system or the part will be protected against malware attaks or unauthorized accesas.
NFR-3	Reliability	The failure in the application cannot be recovered easily.
NFR-4	Performance	The processing time and the response is faster.
NFR-5	Availability	The system will be available all the time when the user needs to get the solution for their problem.
NFR-6	Scalability	The web application which is used in the detection of diseases is scalable.

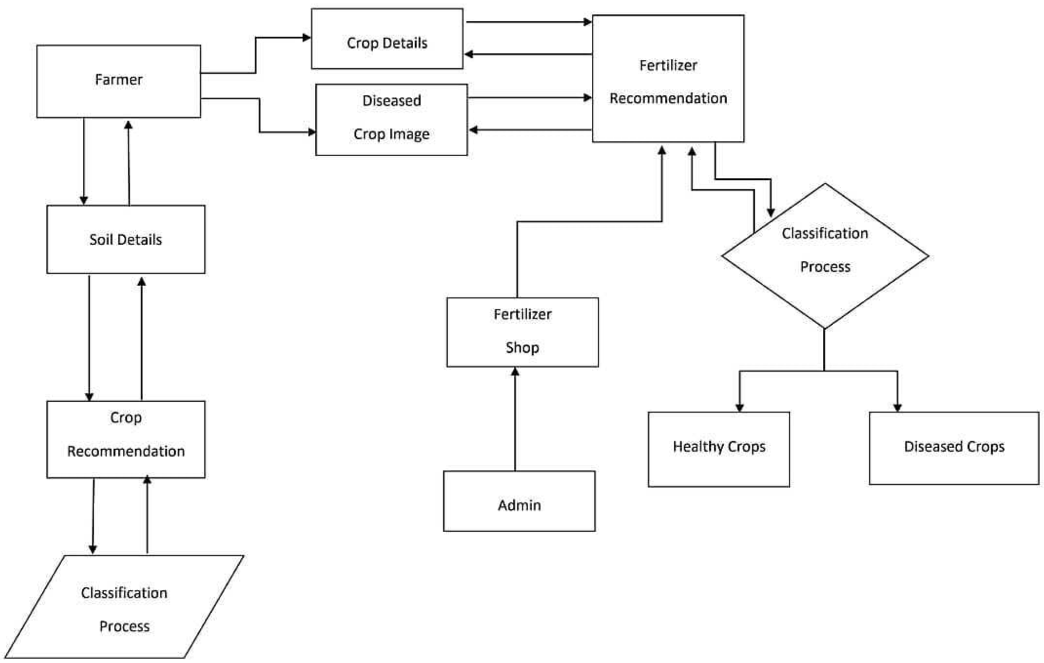
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

Data Flow Diagram:



DFD LEVEL - 1



USER STORIES

USER TYPE	FUNCTIONAL REQUIREMENT(EPIC)	USER STORY NI=UMBER	USER STORY /TASK	ACCEPTANUE CRITERIA	PRIORITY	RELEASE
Customer(mobile user)	Registration	USN-1	As a user ,i can register for the application by entering my email, password , and confirming my password	I can access my account/deshboard	high	sprint-1

	Login	USN-2	As a user ,Ican log into the application by entering email & password	I can login using my email ID accountors or user credential	High	Sprint-1
	Dashboard	USN-3	As a user, I can view application where I can upload my images and the fertilizer should be recommended	I can access my account /deshboard	High	Sprint-2
Customer(web user)	Registration	USN-4	As a web user I can login to my web deshboard.	I can register my username and password	High	Sprin-3
	Login	USN-5	As a web user I can login into the website using login credentials	I can login using login credentials	High	Sprin-4
		USN-6	As a user I can view the web application where the images can be uploaded and the fertilizer should be	I can access my deshboard	High	Sprin-4

			of higher accuracy			
	Dashboard	USN-7	As a user the fertilizer recommended to me should be of higher accuracy	I can access my deshboard	High	Sprin-4
Administrator	Login	USN-8	As a admin I can login to the website using my login credentials	I can login to the website using my login credentials	High	Sprin-5
	Dashboard	USN-9	As a admin I can view the deshboard of the application	I can access my deshboard	High	Sprin-5

PRODUCT BACKLOG,SPRINT SCHEDULE,AND ESTU

SPRINT	FUNCTIONAL RQUIREMEMT(EPIC)	USER STORY NUMBER	USER STORY / TASK	STORY POINT (TOTAL)	PRIORITY	TEAM MEMBERS
Sprint-1	model creation and training(fruits)		creating a model which can classify the diseased fruit plants from the given	8	High	Reshmi FionA T,Vijay s,vanitha s, Janani m.

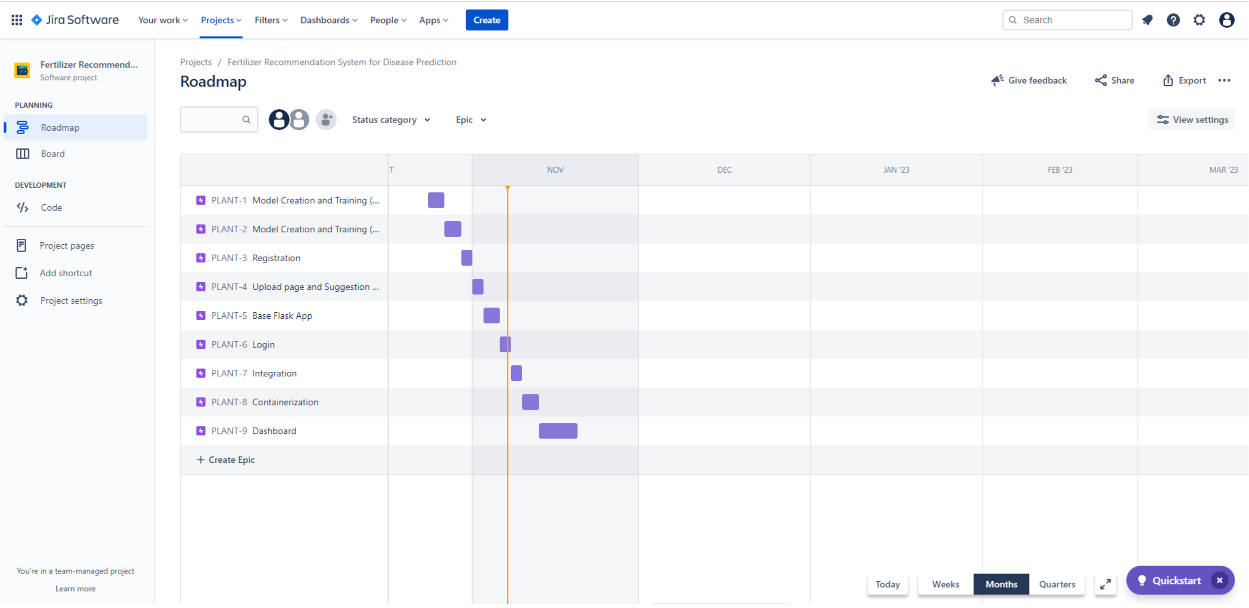
			image ,a so need to test			
	Model Creation and Training (vegetables)		Creating a model which can classify the dieased vegetables plants from the given images . also need to test the model an d deploy it On IBM Cloud	2	High	Reshmi FionaT, Vijay S, Vanitha S, Janani M.
Sprint-2	Model creation and Traning (vegetables)		Creating a model which can classify diseased vegetable plants from the given images and train on IBM Cloud.	6	High	Reshmi FionaT, Vijay S, Vanitha S, Janani M.
	Registration	USN-1	As a user, I can register by entering my email, password, and confirming my password	3	Medium	Reshmi FionaT, Vijay S, Vanitha S, Janani M.

	Upload page	USN-2	As a user, I will be redirected to a page where I can upload my pictures of crops.	4	High	Reshmi FionaT, Vijay S, Vanitha S, Janani M.
	Suggestion results	USN-3	As a user, I can view the results and then obtain the suggestions provided by the ML Model	4	High	Reshmi FionaT, Vijay S, Vanitha S, Janani M.
	Base Flask App		A base Flask web app must be created as an interface for the ML model	2	High	Reshmi FionaT, Vijay S, Vanitha S, Janani M.
Sprint-3	Login	USN-4	As a user/admin/shopkeeper, I can log into the application by entering email & password	2	High	Reshmi FionaT, Vijay S, Vanitha S, Janani M.
	User Dashboard	USN-5	As a user, I can view the previous results and history	3	Medium	Reshmi FionaT, Vijay S, Vanitha S, Janani M.

	Integration		Integrate Flask, CNN model with Cloudant DB	5	Medium	Reshmi Fiona T, Vijay S, Vanitha S, Janani M
	Containerization		Containerize Flask app using Docker	2	Low	Reshmi Fiona T, Vijay S, Vanitha S, Janani M
Sprint-4	Dashboard (Admin)	USN-6	As an admin, I can view other user details and uploads for other purposes		Medium	Reshmi Fiona T, Vijay S, Vanitha S, Janani M
	Dashboard (Shopkeeper)	USN-7	As a shopkeeper, I can enter fertilizer products and then update the details if any	2	Low	Reshmi Fiona T, Vijay S, Vanitha S, Janani M
	Containerization		Create and deploy Helm charts using Docker Image made before	2	Low	Reshmi Fiona T, Vijay S, Vanitha S, Janani M

Create and deploy Helm charts using Docker Image made before

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 October 2022	29 October 2022	10	30 Oct 2022
Sprint-2	15		31 October 2022	05 November 2022	15	06 Nov 2022
Sprint-3	15		07 November 2022	12November 2022	15	13 Nov2022
Sprint-4	12		14 November 2022	05 November 2022	10	20 Nov 2022



Projects / Fertilizer Recommendation System for Disease Prediction

Roadmap

PLANT-1 Model Creation and Training (...
PLANT-2 Model Creation and Training (...
PLANT-3 Registration
PLANT-4 Upload page and Suggestion ...
PLANT-5 Base Flask App
PLANT-6 Login
PLANT-7 Integration
PLANT-8 Containerization
PLANT-9 Dashboard
+ Create Epic

CODING AND SOLUTIONING

App.py

```
import requests
```

```
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
import tensorflow as tf
from flask import Flask, redirect,render_template, request,url_for
import os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session
app= Flask(__name__)
model1 = load_model('fruit.h5')
model = load_model('vegetable.h5')
@app.route('/')
def home():
 return render_template('homepage.html')
@app.route('/prediction')
def prediction():
 return render_template('predict.html')
@app.route('/predict',methods=['GET','POST'])
def upload():
  if request.method =='POST':
    f= request.files['image']
    basepath = os.path.dirname(__file__)
    file_path=os.path.join(basepath, 'uploads',secure_filename(f.filename))
    f.save(file_path)
    print("file save")
    img = image.load_img(filepath, target_size=(128,128))
    x=image.img_to_array(img)
    print("image to gray")
    x=np.expand_dims(x, axis=0)
    plant=request.form['plant']
  if (plant=="fruit"):
   model1.predict_classess(x)
   print(preds)
   df=pd.read_excel('precautions - fruits.xlsx')
   print (df.iloc[preds[0]]['cautions'])
```

```
    else:
      preds=model.predict_classes(x)
      df=pd.read_excel("precautions-veg.xlsx")
      print(df.iloc[preds[0]]['caution'])

    return df.iloc[preds[0]]['caution']
if __name__=="__main__":
 app.run(debug=True)
```

HTML PAGES

HOMEPAGE

```
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>
```

```html
<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="/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:center;padding-top: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;text-align:justify;">Agriculture was the essential development in the rise of human
  civilization, whereby farming of acclimatize species produced food oversupply that enabled people to reside in cities.
  Plants were independently sophisticated in at least 11 regions of the world. Industrial agriculture based on large-scale monocropping in the twentieth century came to influence agricultural output, though
  about 2 billion people still depended on maintaining agriculture.The plant diseases effect the production. Identification of diseases and taking necessary precautions is all done through naked eye, which requires labour and laboratries. This application helps farmers in detecting the diseases by observing the spots on the leaves, which inturn saves effort and labor costs.</div><br><br>
</div>
</div>
<div style="width:40%;float:right;"><br><br>
<img src="https://images.pexels.com/photos/35196/water-plant-green-fine-layers.jpg" style="max-height:100%;max-width: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", "")
  }
}
</script>
</body>
</html>
```

PREDICTION PAGE

```
!DOCTYPE html>
<html lang="en">
<head>
    <title>predict</title>
</head>
<style>
    .container{
        display: flex;
        padding: 60px 70px 60px 70px;
    }
    .card{
        padding: 70px 80px 70px 80px;
    }
```

```
  .menu{
    padding: 10px 10px 10px 10px;
    background-color: black;
    color: white;
    font-size: 15pt;
  }
</style>
<body>
  <div class="menu">
    <ul ><li>Plant disease Prediction</li></ul></div>
  <div class="container">

    <img src="C:\Users\chana\OneDrive\Documents\c\Screenshot 2022-11-09 182944.jpg">
    <div class="card">
    <form>
      <h1>Drop in the image to get the Prediction </h1><br><br>
      <label><select name="Fruit" id="plant">
        <option value="fruit" id="fruit">Fruit</option>
        <option value="vagitable" id="vig">vegitable</option>
        </select>
      </label><br><br><br>
      <input  id="default-btn" type="file" name=""
onchange="document.getElementById('output').src=window.URL.createObjectURL(this.files[0])"><br>
<br><br>
      <img src="" id="output">

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

    </form>




</body>
</html>
```

CONCLUSION:

Thus this system provides solution for the prediction of plant diseases. It is useful for farmers all over the world.