

# **FERTILIZERS RECOMMENDATION SYSTEM FOR DISEASE PREDICTION**

**A PROJECT REPORT**

**Submitted By**

DEEPAN CHAKARAVARTHY H	210819205007
SHANJAY M	210819205046
SATHISH KUMAR P	210819205051
SHAKTHIVEL B	210819205042
PARASURAMAN E	210819205034

**TEAM ID : PNT2022TMID25497**

**In partial fulfilment for the award of the degree**

**Of**

**BACHELOR OF TECHNOLOGY**

**In**



**INFORMATION TECHNOLOGY**

**KINGS ENGINEERING COLLEGE, IRUNGATTUKOTAI**

**ANNA UNIVERSITY: CHENNAI 600025**

## **BONAFIDE CERTIFICATE**

**Certified that mini project report “Fertilizers Recommendation System For Disease Prediction” is bonafide work of “DEEPAN CHAKARAVARTHY H, SHANJAY M, SATHISH KUMAR P, SHAKTHIVEL B, PARASURAMAN E” who carried out this Nalaiyathiran project work under my supervision.**

### **SIGNATURE**

**Dr.G. MANIKANDAN**

### **HEAD OF THE DEPARTMENT**

**Professor**

Dept of Information Technology,  
Kings Engineering College,  
Irungattukottai,  
Chennai-602 117

### **SIGNATURE**

**Mr. J BRISO BECKY BELL**

### **SUPERVISOR**

**Associate Professor**

Dept of Information Technology  
Kings Engineering College,  
Irungattukottai,  
Chennai-602 117.

## ACKNOWLEDGEMENT

We thank God for his unstoppable blessings and also for giving us good knowledge and strength in enabling us to finish our project. Our deep gratitude goes to our founder late **Dr. D. SELVARAJ, M.A., M.Phil.**, for his patronage in the completion of our project. We like to take this opportunity to thank our honourable chairperson **Dr.S. NALINI SELVARAJ, M.COM., MPhil., Ph.D.** and honourable director, **MR.S. AMIRTHARAJ, M.Tech., M.B.A** for their support given to us to finish our project successfully. We wish to express our sincere thanks to our beloved principal. **Dr.T.JOHN ORAL BASKAR., M.E., Ph.D** for his kind encouragement and his interest towards us.

We are extremely grateful and thanks to our professor **Dr.G.MANIKANDAN**, head of Information Technology, Kings Engineering College, for his valuable suggestion, guidance and encouragement and to our SPOC **DR. MRS. PARAMASHWARI**, head of Artificial Intelligence and Data Science. We wish to express our sense of gratitude to our project supervisor **Dr. J Briso Becky Bell**, Assistant Professor of Information Technology Department, Kings Engineering College and project evaluator **Dr. M. ROBINSON JOEL**, Associate Professor of Information Technology Department, Kings Engineering College whose idea and direction made our project a grand success, also our Industrial Mentor **DURGA PRASAD** . We express our sincere thanks to our parents, friends and staff members who have helped and encouraged us during the entire course of completing this project work successfully.

## **TABLE OF THE CONTENT**

<b>CHAPTER</b>	<b>CONTENTS</b>	<b>PAGE NO</b>
<b>1</b>	<b>INTRODUCTION</b>	
	1.1 Project overview	1
	1.2 Purpose	2
<b>2</b>	<b>LITERATURE SURVEY</b>	
	2.1 Existing problems	
		2
	2.2 References	2
<b>3</b>	<b>IDEATION AND PROPOSED SOLUTION</b>	
	3.1 Empathy map	4
	3.2 Ideation and Brainstorming	5
	3.3 Proposed solution	6
	3.4 Problem solution fit	6

<b>4</b>	<b>REQUIREMENT ANALYSIS</b>	
	4.1 Functional requirement	8
	4.2 Non-functional requirements	8
<b>5</b>	<b>PROJECT DESIGN</b>	
	5.1 Data flow diagrams	8
	5.2 Solution & technical architecture	8
	5.3 User stories	9
<b>6</b>	<b>PROJECT PLANNING &amp; SCHEDULING</b>	
	6.1 Sprint planning & estimation	11
	6.2 Sprint delivery schedule	11
	6.3 Reports from jira	12
<b>7</b>	<b>CODING &amp; SOLUTIONING</b>	13
<b>8</b>	<b>TESTING</b>	
	8.1 Test cases	15
	8.2 User acceptance testing	15
<b>9</b>	<b>RESULTS</b>	
	9.1 Performance metrics	16
<b>10</b>	<b>ADVANTAGES &amp; DISADVANTAGES</b>	17
<b>11</b>	<b>CONCLUSION</b>	18
<b>12</b>	<b>FUTURE SCOPE</b>	18
<b>13</b>	<b>APPENDIX</b>	18
	Source code & project demo link	



# 1. 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. Application of computer vision and image processing strategies simply assist farmers in all of the regions of agriculture. Generally, the plant diseases are caused by the abnormal physiological functionalities of plants. Therefore, the characteristic symptoms are generated based on the differentiation between normal physiological functionalities and abnormal physiological functionalities of the plants. Mostly, the plant leaf diseases are caused by Pathogens which are positioned on the stems of the plants. These different symptoms and diseases of leaves are predicted by different methods in image processing. These different methods include different fundamental processes like segmentation, feature extraction and classification and so on. Mostly, the prediction and diagnosis of leaf diseases are depending on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves.

## 1.1. PROJECT OVERVIEW

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 Convolutional 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 web pages is written in Spyder-Anaconda python and tested. 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. Agriculture is the main aspect of country development. Many people lead their life from agriculture field, which gives fully related to agricultural products. Plant disease, especially on leaves, is one of the major factors of reductions in both quality and quantity of the food crops. In agricultural aspects, if the plant is affected by leaf disease then it reduces the

growth of the agricultural level. Finding the leaf disease is an important role of agriculture preservation. After pre-processing 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.

## **1.2 PURPOSE**

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

## **2. LITERATURE SURVEY**

### **2.1 EXISTING PROBLEM**

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,a simple prediction method for soil based fertilizer recommendation system for predicted crop diseases. This method gives less accuracy and prediction, an IoT based system for leaf disease detection and fertilizer recommendation whichbased on Machine Learning techniques yields less 80 percentage accuracies.

### **2.2 REFERENCES**

[1] Luca Bencini, Davide Di Palma, Giovanni Collodi, G. Manes and Antonio Manes, "Agricultural monitoring based on wireless sensor network technology: Real long life deployments for physiology and pathogens control.". Third International Conference on Sensor Technologies and Applications. IEEE, 2009. [2] Journal Article Mrs. N. Hemageetha, Dr. G.M. Nasira, "Analysis of soil condition based on pH value using Classification Technique", IOSRJCE, Volume 18, Issue 6, Nov-Dec 2016. <https://www.iosrjournals.org/iosrjce/papers/Vol18-issue6/Version3/I1806035054.pdf> [3]



International Journal of Computer Science and Informatics. Jay Gholap, Anurag Ingole, Jayesh Gohil, Shailesh Gargade and Vahida Attar, "Soil Data Analysis Using Classification Techniques and Soil Attribute Prediction", IJCSI, Vol. 9, Issue 3, No 3, ISSN: 1694-0814, May 2012. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.402.2833&rep=rep1&type=pdf> [4] Duan Yan-e, Design of Intelligent Agriculture Management Information System Based on IOT, IEEE, 4th, Fourth International reference on Intelligent Computation Technology and Automation, 2011 <https://ieeexplore.ieee.org/document/5750779> [5] Bindu Garg and Tanya Sah, "Prediction of Crop Yield Using Fuzzy-Neural System" , 19th October, 2019. [https://link.springer.com/chapter/10.1007/978-3-030-19562-5\\_21](https://link.springer.com/chapter/10.1007/978-3-030-19562-5_21) [6] Bindu Garg, B., Beg, M. M. S. & Ansari, A. Q. "Fuzzy time series model to forecast rice production, July-2013" [https://www.researchgate.net/publication/258282994\\_Fuzzy\\_Time\\_Series\\_Model\\_to\\_Forecast\\_Rice\\_Production](https://www.researchgate.net/publication/258282994_Fuzzy_Time_Series_Model_to_Forecast_Rice_Production) [7] Website DAVIS, L. E..25 1943. MEASUREMENTS OF pH WITH THE GLASS ELECTRODE AS AFFECTED BY SOIL MOISTURE Soil Sci. 56: 405-422, Illus. [8] James. N. Mugo, Nancy N. Karanja, Charles K. Gachene, Klaus Dittert, Shadrack O. Nyawade, and Elmar Schulte- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7210878/Geldermann> - Assessment of soil fertility and potato crop nutrient status in central and eastern highlands of Kenya, 8th May, 2020

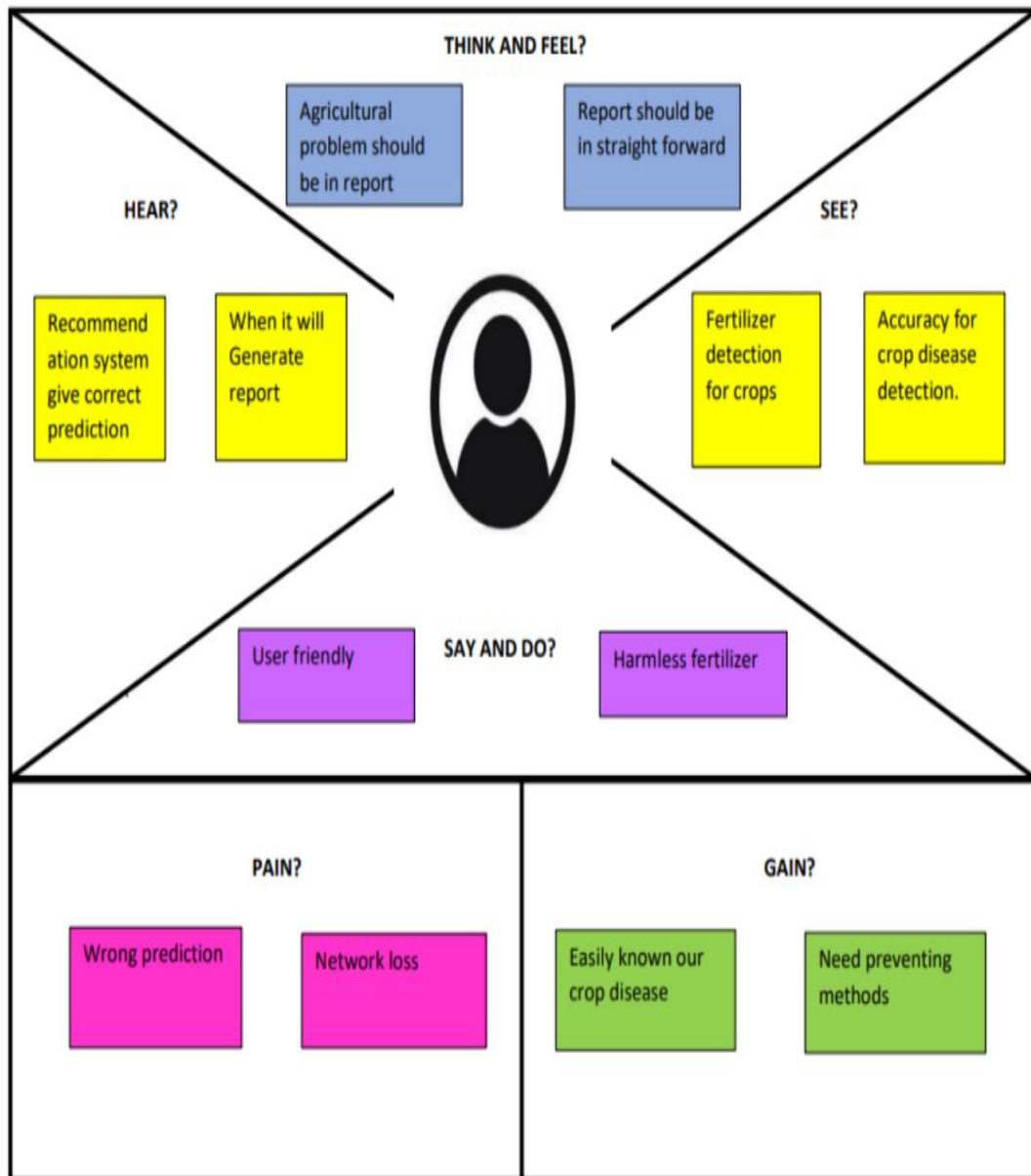
## **2.3 PROBLEM STATEMENT DEFINITION**

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 & PROPOSED SOLUTION**

### **3.1 EMPATHY MAP CANVAS**

## EMPATHY MAP FOR FERTILIZER RECOMMENDATION SYSTEM



**Fertilizer Recommendation System for Disease Prediction**

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.

**1. Overview and Problem Statement**

**Before you start:** A little bit of preparation goes a long way with the design. Here's what you need to do to get going.

**Define your problem statement:** What problem are you trying to solve? What is your problem as a user? What is the business? What is the goal of your solution?

**2. User Flow and Features**

**User Flow:** The user flow is a sequence of steps that a user takes to achieve a goal. In this case, the goal is to identify a plant disease and get a fertilizer recommendation.

**Features:** The features are the functionalities that the app provides. These include:
 

- Image Upload:** Users can upload a photo of a plant with a disease.
- Disease Identification:** The app uses a machine learning model to identify the disease.
- Fertilizer Recommendation:** Based on the identified disease, the app recommends a suitable fertilizer.
- Interactive User Interface:** The app has a user-friendly interface with clear instructions and feedback.

**3. System Architecture**

The system architecture shows the components of the app and how they interact. It includes the user interface, the data layer, the business logic, and the machine learning model.

**4. Data Flow and Model Training**

The data flow shows how data is collected, processed, and used to train the machine learning model. The model is trained on a dataset of plant diseases and their corresponding fertilizer recommendations.

### **3.3 PROPOSED SOLUTION**

#### Description

##### 1.Problem Statement (Problem to be solved)

Disease in plants reduced the quantity and quality of the plant's productivity. Identifying the disease in plant is hard to find.

##### 2.Idea / Solution description

One solution of the problem is to identifying the disease in early stage and using the correct fertilizer.

##### 3.Novelty / Uniqueness

This application can suggest good fertilizer for the disease in the plant by recognizing the images.

##### 4.Social Impact / Customer Satisfaction

It helps the farmer by identifying the disease in the early stage and increase the quality and quantity of crops in efficient way.

##### 5.Scalability of the Solution

The application is recommending to farmer in subscription basis      This application can be improved by introducing online purchases of crops, fertilizer easily.

### **3.4 PROBLEM SOLUTION FIT**



## **4.REQUIREMENT ANALYSIS**

### **4.1 FUNCTIONAL REQUIREMENT**

FR-1 User Registration Registration through Form Registration through Gmail Registration through LinkedIN

FR-2 User Confirmation Confirmation via Email Confirmation via OTP

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.

FR-5 Leaf identification Identify the leaf and predict the disease in leaf.

FR-6 Image description Suggesting the best fertilizer for the disease.

### **NON-FUNCTIONAL REQUIREMENT**

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 for the predicting the disease 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 leaf

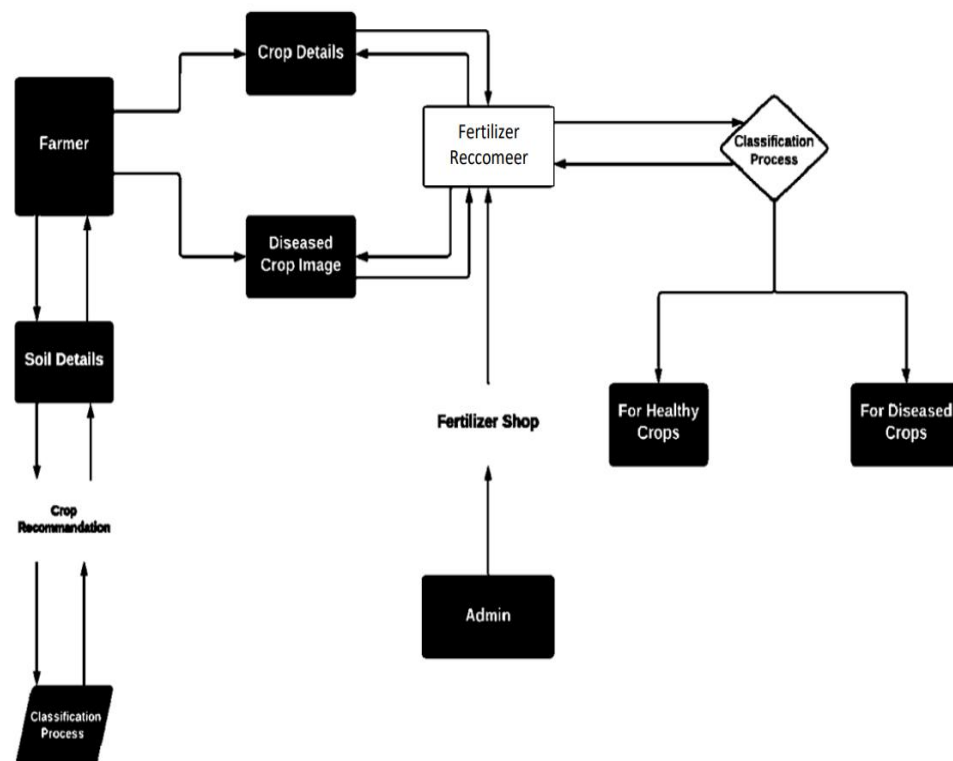
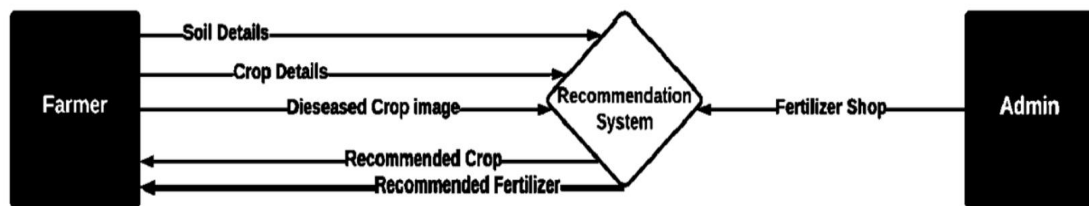
## **5.PROJECT DESIGN**

### **5.1 DATA FLOW DIAGRAM**

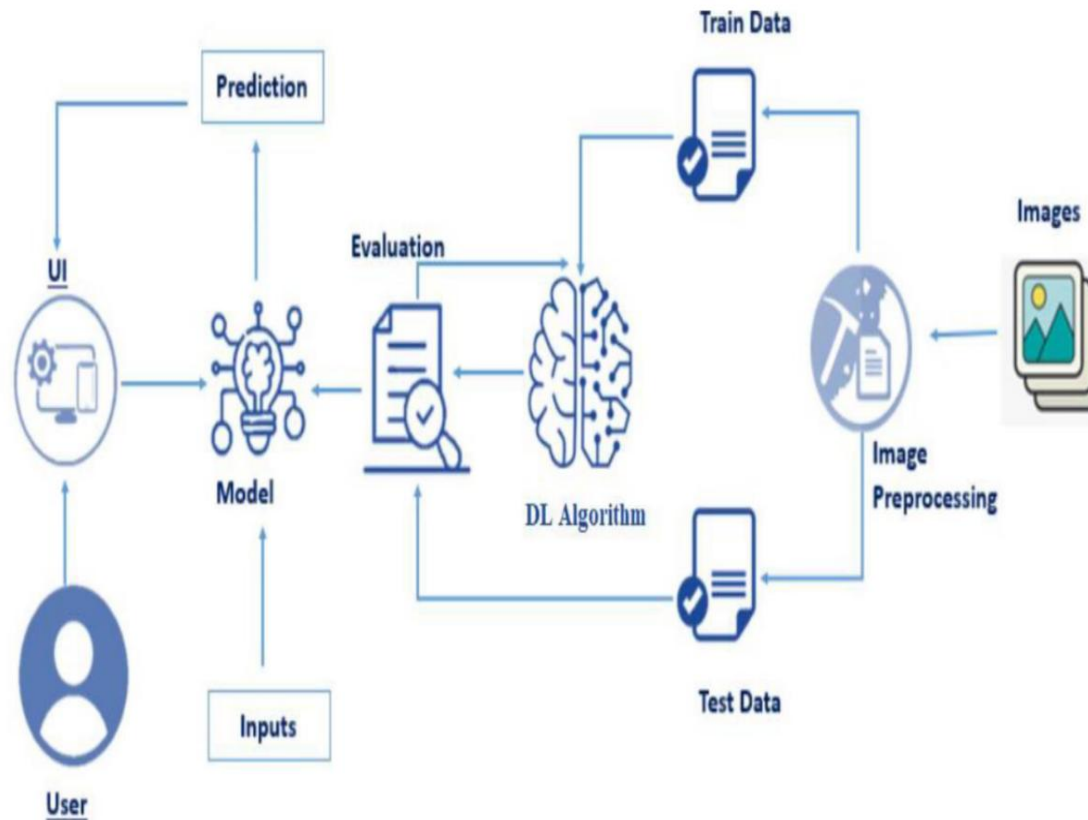
Data Flow Diagrams: A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the

system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

### DFD LEVEL - 0



## 5.2 SOLUTION & TECHNICAL ARCHITECTURE



## 5.3 USER STORIES

- As a user, I can register for the application by entering my email, password, confirming my password and mobile number
- As a user, I can log into the application by entering email & password
- As a user, I can view the page of the application where I can upload my images and the fertilizer should be recommended
- As a user , I can login to web dashboard just like website dashboard
- As a user, I can login to my web dashboard with the login credentials



- As a user, I can view the web application where I can upload my images and fertilizer should be recommended
- As a user, the fertilizer recommended to me should be of higher accuracy
- As a admin, I can login to the website using my login credentials

## 6.PROJECT PLANNING & SCHEDULING

### 6.1 SPRINT PLANNING AND ESTIMATION

#### User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile ssuser)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering email & password	I can login using my E-mail ID accounts or user credentials	High	Sprint-1
	Dashboard	USN-3	As a user, I can view the page of the application where i can upload my images and the fertilizer should be recommended	I can access my account/ dashboard	High	Sprint-2
Customer (Webuser)	Registration	USN-4	As a user, I can login to web dashboard just Like website dashboard	I can register using my username and password	High	Sprint-3
	Login	USN-5	As a user, I can login to my web dashboard with the login credentials	I can login using my User credentials	High	Sprint-3
	Dashboard	USN-6	As a user, I can view the web application where i can upload my images and thefertilizer should be recommended	I can access my account/ dashboard	High	Sprint-4
		USN-7	As a user, the fertilizer recommended to me should be of higher accuracy	I can access my accou nt/ dashb oard	High	Sprint-4
Administrator	Login	USN-8	As a admin, I can login to the website using my login credentials	I can login to the	High	Sprint-2

## 6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	10	30 Oct 2022
Sprint-2	15	6 Days	31 Oct 2022	05 Nov 2022	15	06 Nov 2022
Sprint-3	15	6 Days	07 Nov 2022	12 Nov 2022	15	13 Nov 2022
Sprint-4	12	6 Days	14 Nov 2022	19 Nov 2022	10	20 Nov 2022

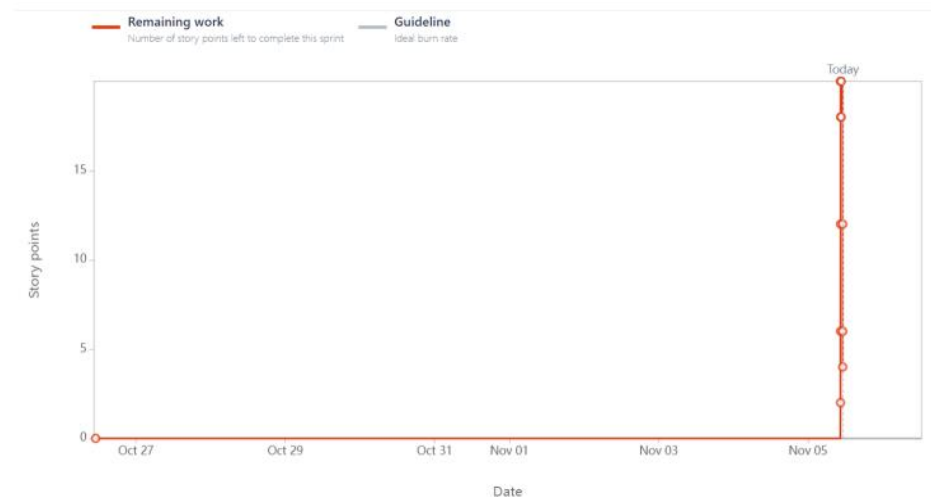
## 6.3 REPORTS FROM JIRA

### 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)

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

Burndown chart:



## 7. CODING AND SOLUTIONING

### Import Required Libraries

```
In [52]: import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
from keras.utils import np_utils
import tensorflow as tf
from tensorflow.keras.layers import Conv2D, Dense, Flatten
```

```
In [2]: print(tf.__version__)
```

2.9.2

```
In [3]: mnist_ds = tf.keras.datasets.mnist.load_data()
```

Downloading data from <https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz>  
11490434/11490434 [\*\*\*\*\*] - 0s 0us/step

```
In [4]: mnist_ds
```

### Building The Model

```
In [53]: model = tf.keras.models.Sequential([tf.keras.layers.Conv2D(64, (3, 3), input_shape=(28, 28, 1), activation="relu"),
tf.keras.layers.Conv2D(32, (3, 3), activation="relu"),
tf.keras.layers.Flatten(),
tf.keras.layers.Dense(10, activation=tf.nn.softmax)])
```

```
In [54]: model.compile(loss='categorical_crossentropy', optimizer="Adam", metrics=["accuracy"])
```

```
In [55]: model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 64)	640
conv2d_1 (Conv2D)	(None, 24, 24, 32)	18464
flatten (Flatten)	(None, 18432)	0
dense (Dense)	(None, 10)	184330

\*\*\*\*\*  
Total params: 203,434  
Trainable params: 203,434  
Non-trainable params: 0

## Training The Model

```
In [56]: model.fit(training_images, training_labels, batch_size=32, epochs=5, validation_data=(test_images, test_labels))

Epoch 1/5
1875/1875 [*****] - 16s 4ms/step - loss: 0.1254 - accuracy: 0.9625 - val_loss: 0.0519 - val_accuracy: 0.9831
Epoch 2/5
1875/1875 [*****] - 7s 4ms/step - loss: 0.0464 - accuracy: 0.9861 - val_loss: 0.0385 - val_accuracy: 0.9868
Epoch 3/5
1875/1875 [*****] - 7s 4ms/step - loss: 0.0296 - accuracy: 0.9907 - val_loss: 0.0409 - val_accuracy: 0.9872
Epoch 4/5
1875/1875 [*****] - 7s 4ms/step - loss: 0.0281 - accuracy: 0.9937 - val_loss: 0.0403 - val_accuracy: 0.9878
Epoch 5/5
1875/1875 [*****] - 7s 4ms/step - loss: 0.0139 - accuracy: 0.9957 - val_loss: 0.0457 - val_accuracy: 0.9872

Out[56]:
```

## Test The Model

```
In [58]: metrics = model.evaluate(test_images, test_labels, verbose=0)

print("Test Loss -> {} \nTest Accuracy -> {}".format(metrics[0], metrics[1]))

Test Loss -> 0.04573516175150871
Test Accuracy -> 0.9872000217437744
```

```
In [67]: model.predict(test_images[2:8])

1/1 [*****] - @s 15ms/step
Out[67]: array([[2.32065427e-08, 9.99983430e-01, 8.10439190e-07, 1.28179977e-07,
 9.64923492e-06, 1.83879649e-06, 1.62038040e-07, 1.56461965e-06,
 2.34936374e-06, 3.22469944e-08],
 [9.99998927e-01, 1.04071238e-13, 7.69856399e-07, 1.84126245e-09,
 3.37900085e-13, 4.71777106e-09, 8.84182239e-09, 2.02508791e-11,
 3.22932721e-07, 9.56373647e-09],
 [8.73478698e-13, 4.26847549e-13, 1.15858136e-10, 3.97662771e-11,
 9.99999081e-01, 2.68545906e-12, 8.96004648e-11, 7.41609482e-11,
 4.87753553e-08, 1.05269102e-07],
 [2.15423035e-09, 9.99581635e-01, 2.15949945e-06, 1.00863390e-08,
 2.10376020e-05, 2.63231090e-08, 3.26978977e-08, 3.81208694e-04,
 1.27808356e-05, 1.17525337e-06],
 [2.41032138e-18, 9.36788000e-11, 3.97475330e-10, 3.54850779e-13,
 9.99299288e-01, 6.94019900e-09, 6.61158953e-14, 4.28452246e-10,
 7.00477336e-04, 2.29253416e-07],
 [1.89875802e-16, 2.21634187e-11, 1.76906703e-09, 2.65193867e-09,
 2.5687592e-06, 2.08839956e-08, 1.60236594e-14, 2.67538752e-11,
 5.46009005e-06, 9.99991894e-01]], dtype=float32)
```

```
In [74]: history=model.predict(np.array([test_images[7]]))
history

1/1 [*****] - @s 17ms/step
Out[74]: array([[1.8987580e-16, 2.2163419e-11, 1.7690670e-09, 2.6519387e-09,
 2.5687532e-06, 2.0883996e-08, 1.6023662e-14, 2.6753875e-11,
 5.4600901e-06, 9.9999189e-01]], dtype=float32)
```

```
In [75]: np.argmax(history, axis=1)
```

```
Out[75]: array([9])
```

```
In [73]: #It predicted as 9
```

let us see, it is correct or not?

```
In [78]: t1=test_labels[7]
t1
```

```
Out[78]: array([0., 0., 0., 0., 0., 0., 0., 0., 1.], dtype=float32)
```

```
In [81]: np.argmax(t1)
```

```
Out[81]: 9
```

It Predicted Correctly!!!

## 8. TESTING

### 8.1 TEST CASES

				Date	3-Nov-22								
				Team ID	PNT2022TMD25407								
				Project Name	Fertilizers Recommendation System For Disease Prediction								
				Maximum Marks	4marks								
Test case ID	Feature Type	Component	Test Scenario	Pre-Requlite	Steps To Exeucite	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation (Y/N)	BUG ID	Executed By
Register	Functional	Register Page	Verify user is able to see the Login/signup popup when user clicked on My account button	Mail id,password	1. Enter URL and click go 2. Click on My Account dropdown button 3. Verify login/signup popup displayed or not	<a href="http://127.0.0.1:5000/signup">http://127.0.0.1:5000/signup</a>	Login/signup popup should display	Working as expected	Pass	Steps are follow properly			
RegisterPage	UI	Register Page	Verify the UI elements in Login/signup popup	Mail id,password	1. Enter URL and click go 2. Click on My Account dropdown button 3. Verify login/signup popup with below UI elements: a. email text box b. password text box c. login button d. New customer? Create account link e. Last password? Recovery passwor d link	<a href="http://127.0.0.1:5000/signup">http://127.0.0.1:5000/signup</a>	Application should show below UI elements: a. email text box b. password text box c. login button with orange colour d. New customer? Create account link	Working as expected	Fail	Steps are not clear to follow		BUG-1	
LoginPage_TC_003	Functional	Login page	Verify user is able to log into application with Valid credentials	Mail id,password	1. Enter URL( <a href="http://127.0.0.1:5000/signup">http://127.0.0.1:5000/signup</a> ) and click go 2. Click on My Account dropdown button 3. Enter Valid username/email in Email text box 4. Enter valid password in password text box 5. Click on login button	Username: csxp@gmail.com password: Testing123	User should navigate to user account homepage	Working as expected	Pass	Steps are follow properly			
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with Invalid credentials	Mail id,password	1. Enter URL( <a href="http://127.0.0.1:5000/signup">http://127.0.0.1:5000/signup</a> ) and click go 2. Click on My Account dropdown button 3. Enter Invalid username/email in Email text box 4. Enter valid password in password text box 5. Click on login button	Username: csxp@gmail.com password: Testing123	Application should show "Incorrect email or password" validation message.	Working as expected	Fail	Steps are not clear to follow		BUG-2	
Login	UI	Login page	Verify user is able to log into application with Invalid credentials	Mail id,password	1. Enter URL( <a href="http://127.0.0.1:5000/signup">http://127.0.0.1:5000/signup</a> ) and click go 2. Click on My Account dropdown button 3. Enter Valid username/email in Email text box 4. Enter Invalid password in password text box 5. Click on login button	Username: csxp@gmail.com password: Testing123787686786876	Application should show "Incorrect email or password" validation message.	Working as expected	Fail	Steps are not clear to follow		BUG-3	
LoginPage	Functional	Login page	Verify user is able to log into application with Invalid credentials	Mail id,password	1. Enter URL( <a href="http://127.0.0.1:5000/signup">http://127.0.0.1:5000/signup</a> ) and click go 2. Click on My Account dropdown button 3. Enter Invalid username/email in Email text box 4. Enter Invalid password in password text box 5. Click on login button	Username: csxp password: Testing123787686786876	Application should show "Incorrect email or password" validation message.	Working as expected	Fail	Steps are not clear to follow			
Home	Functional	Dashboard	verify a user is able to predict and disease	web camera, videos, images	1. Enter upload image 2. click predict 3. show the disease	infected leaf image	detect the infected leaf	Working as expected	Pass	Steps are follow properly			
Home	Functional	Dashboard	verify a user is able to predict and disease	web camera, videos, images	1. Enter upload image 2. click predict 3. show the disease	infected leaf image	detect the infected leaf	Working as expected	Pass	Steps are follow properly			
Home	Functional	Dashboard	verify a user is able to predict and disease	web camera, videos, images	1. Enter select video 2. click predict 3. show the disease	infected leaf image	detect the infected leaf	Working as expected	Pass	Steps are follow properly			
Home	Functional	Dashboard	verify a user is able to predict and disease	web camera, videos, images	1. Enter select video 2. click predict 3. show the disease	infected leaf image	detect the infected leaf	Working as expected	Pass	Steps are follow properly			
Home	Functional	Dashboard	verify a user is able to predict and disease	web camera, videos, images	1. click camera 2. show the disease and further	infected leaf image	detect the infected leaf	Working as expected	Pass	Steps are follow properly			
Home	Functional	Dashboard	verify a user is able to predict and disease	web camera, videos, images	1. click camera 2. show the disease and further	infected leaf image	detect the infected leaf	Working as expected	Pass	Steps are follow properly			

### 8.2 USER ACCEPTANCE TESTING

#### 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
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

### 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
Print Engine	7	0	0	7
Client Application	41	0	0	41
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

## 9. RESULTS

### 9.1 PERFORMANCE METRICS

#### ▾ Observing the metrics

```
[ ] # Final evaluation of the model
    metrics = model.evaluate(x_test, y_test, verbose=0)
    print("Metrics (Test loss &Test Accuracy) : ")
    print(metrics)
```

```
Metrics (Test loss &Test Accuracy) :
[0.08848220854997635, 0.9772999882698059]
```



## 10.ADVANTAGES & DISADVANTAGES

### ADVANATGES

- 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

## **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 accuracies are obtained.
- The accuracies are increased by increasing more convolution layers.
- The accuracy of classification also increased by varying dense layers.
- Different accuracies are obtained by varying the size of kernel used in the convolution layer output.
- Accuracies are different while varying the size of the train and test datasets.

## **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

(Jupyter notebook python code)

fruit.ipynb (due to limited page size the code vegetable.ipynb uploaded in github)

```
#!/usr/bin/env python
```

```
# coding: utf-8
```

```
# In[1]: pwd
```

```
# In[2]: cd E:/IBM_MY_COURSE/Project/Dataset Plant Disease/fruit
```

```
dataset/fruit-dataset
```

```
# # Apply ImageDataGenerator functionality to Train and Test set
```

```
# # Preprocessing # In[3]: from keras.preprocessing.image import
```

```
ImageDataGenerator
```

```
train_datagen
```

```
=17
```

```
ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal
```

```
flip=True) test_datagen = ImageDataGenerator(rescale=1) # In[4]: pwd
```

```
#
```

```
In[5]: x_train =
```

```
train_datagen.flow_from_directory('E:/IBM_MY_COURSE/Project/Dataset
```

```
Plant Disease/fruit
```

```
dataset/fruitdataset/train', target_size=(128, 128), batch_size=32, class_mode='cate
```

```
gorical')
```

```
#In[6]:
```

```

x_test=test_datagen.flow_from_directory('E:/IBM_MY_COURSE/Project/Dataset
Plant Disease/fruit-dataset/fruit-dataset/test',target_size=(128,128),
batch_size=32,class_mode='categorical') ## Import the models

# In[7]: from tensorflow.keras.models

Import Sequential      from  tensorflow.keras.layers

import Dense,Convolution2D,MaxPool2D,Flatten

## Initializing the models 10

# In[8]: model=Sequential()

## Add CNN Layers

#In[9]: model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

# In[10]: x_train.class_indices

## Add Pooling layer

# In[11]: model.add(MaxPool2D(pool_size=(2,2)))

## Add Flatten layer # In[12]: model.add(Flatten())

## Add Dense Layer18

# In[21]: model.add(Dense(40, kernel_initializer='uniform',activation='relu'))

model.add(Dense(20, kernel_initializer='random_uniform',activation='relu'))

## Add Output Layer # In[24]: model.add(Dense(6,activation='softmax',
kernel_initializer='random_uniform'))

##Compile the model #

In[25]:

model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accur
acy' ]) # In[26]: len(x_train)

```

```

# In[27]: 5384/32

# # Fit the Model

#In[28]:

model.fit_generator(x_train,steps_per_epoch=168,validation_data=x_test,validation_steps=52,epochs=3)

# # Save the Model

# In[29]: model.save("fruit.h5")

# In[30]: ls

# # Test the Model

#

In[32]: from keras.preprocessing

import image from tensorflow.keras.preprocessing.image

import img_to_array

from tensorflow.keras.models import load_model import numpy as np

# In[33]: model = load_model("fruit.h5")

# # Test Apple_Healthy Class images19

# In[37]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant
Disease/fruitdataset/fruit-dataset/test/Apple
healthy/00fca0da-2db3-481b
b98a9b67bb7b105c
RS_HL_7708.JPG',target_size=(128,128)) 11

# In[39]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0)

# In[40]: pred = model.predict_classes(x)

```

```

# In[41]: pred

#In[45]:index         =['Apple          Black_rot','Apple          healthy','Corn_(maize)
Northern_Leaf_Blight','Corn_( maize)

healthy','Peach

Bacterial_spot','Peach

healthy']

# In[46]: print('the given image belongs to=',index[pred[0]])

# # Test Apple Black Rot class images # In[54]: img =

image.load_img('E:/IBM_MY_COURSE/Project/Dataset

Plant Disease/fruitdataset/fruit-dataset/test/Apple

Black_rot/0f3d45f4-e121-42cd

a5b6- be2f866a0574

JR_FrgE.S 2870.JPG',target_size=(128,128))

# In[55]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred =

model.predict_classes(x) print('the given image belongs to=',index[pred[0]])

# # Test Corn Northern leaf Blight class images

# In[56]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant

Disease/fruitdataset/fruit

dataset/test/Corn_(maize)

Northern_Leaf_Blight/00a14441-7a62- 4034-bc40-

b196aeab2785

RS_NLB 3932.JPG',target_size=(128,128))

# In[57]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred =

```

```

model.predict_classes(x) print('the given image belongs to=',index[pred[0]])20

## Test Corn Healthy class

images

# In[58]:

img

=

image.load_img('E:/IBM_MY_COURSE/Project/Dataset
Plant
Disease/fruitdataset/fruit-dataset/test/Corn_(maize)
healthy/0a68ef5a-027c-
41ae-b227- 159dae77d3dd
R.S_HL 7969 copy.jpg',target_size=(128,128))

# In[59]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred =
model.predict_classes(x) print('the given image belongs to=',index[pred[0]]) # #
Test Peach Bacterial spot class images

In[60]: img =
image.load_img('E:/IBM_MY_COURSE/Project/Dataset
Plant Disease/fruitdataset/fruit-dataset/test/Peach Bacterial_spot/00ddc106-692e-
4c67-b2e8- 569c924caf49 Rutg._Bact.S 1228.JPG',target_size=(128,128)) 12

# In[61]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred =
model.predict_classes(x) print('the given image belongs to=',index[pred[0]])

# # Test Peach Healthy class images

# In[62]: img = image.load_img('E:/IBM_MY_COURSE/Project/Dataset Plant

```

Disease/fruitdataset/fruit-dataset/test/Peach

healthy/1a07ce54-f4fd-41cf

b088-144f6bf71859\_Rutg.\_HL\_3543.JPG',target\_size=(128,128))

```
# In[63]: x=image.img_to_array(img) x=np.expand_dims(x,axis=0) pred =  
model.predict_classes(x) print('the given image belongs to=',index[pred[0]])
```

## **GITHUB & PROJECT DEMO LINK**

**GITHUB LINK :** <https://github.com/IBM-EPBL/IBM-Project-26546-1660029331>

**PROJECT DEMO LINK :**