1.INTRODUCTION

1.1 PROJECT OVERVIEW

Plant diseases cause major crop production losses worldwide, and a lot of significant research effort has been directed toward making plant disease identification and treatment procedures more effective. It would be of great benefit to farmers to be able to utilize the current technology in order to leverage the challenges facing agricultural production and hence improve crop production and operation profitability. In this work, we designed and implemented a user-friendly fertilizer recommendation system for disease prediction using AI and a treatment recommendation system using machine learning (ML) techniques. CNN was used for feature extraction while the ANN and KNN were used to classify the plant diseases; a content-based filtering recommendation algorithm was used to suggest relevant treatments for the detected plant diseases after classification. The result of the implementation shows that the system correctly detected and recommended treatment for plant disease.

1.2 PURPOSE

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, 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 dependent on the

segmentation such as segmenting the healthy tissues from diseased tissues of leaves.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

1.Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification.

Author: Srdjan Sladojevic,Marko Arsenovic ,Andras Anderia,Dubravko Culibrk and Darko Stefanovic.

Observation: In this paper, a new approach of using deep learning methods was explored in order to automatically classify and detect plant diseases from leaf images. The developed model was able to detect leaf presence and distinguish between healthy leaves and 13 different diseases, which can be visually diagnosed. The complete procedure was described, respectively, from collecting the images used for training and validation to image preprocessing and augmentation and finally the procedure of training the deep CNN and fine-tuning. Different tests were performed in order to check the performance of the newly created model.

Advantage: Novelty of the developed model lies in its simplicity; healthy leaves and background images are in line with other classes, enabling the model to distinguish between diseased leaves and healthy ones or from the environment by using deep CNN.

Disadvantage: As the presented method has not been exploited, as far as we know, in the field of plant disease recognition, there was no comparison with related results, using the exact technique.

2. Fertilizer Recommendation System For Disease Prediction In Tree Leave. International journal of scientific & technology research volume 8, issue 11 , november 2019.

Author: R.Neela,P.Nithya

Observation: The proposed method uses SVM to classify tree leaves,

identify the disease and suggest fertilizer. The proposed method is compared with the existing CNN based leaf disease prediction. The proposed SVM technique gives a better result when compared to existing CNN. For the same set of images, F-Measure for CNN is 0.7and 0.8 for SVM, the accuracy of identification of leaf disease of CNN is 0.6 and SVM is 0.8.

Advantages: The prediction and diagnosing of leaf diseases are dependent on the segmentation such as segmenting the healthy tissues from diseased tissues of leaves.

Disadvantages: This further research is implementing the proposed algorithm with the existing public datasets. Also, various segmentation algorithms can be implemented to improve accuracy. The proposed algorithm can be modified further to identify the disease that affects the various plant organs such as stems and fruits.

3.Plant Disease Detection and Fertilizer Suggestion

Author: Apurva Save,Aksham Gupta,Sarthak Pruthi.

Observation: This 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.

Advantage: The system successfully interprets various Diseases and is also capable of providing fertilizers suggestion for the respective disease.

Disadvantage: This project is limited to just one crop for now but in the future more crops and even flowers dataset can be added so that it is helpful for every agricultural need.

4.Recognition of Image-Based Plant Leaf Diseases Using Deep Learning Classification Models

Author: Sakshi Takkar,Anuj Kakran,Veerpal Kaur,Manik Rakhra.

Observation: Protection of crops in an agriculture field is a very tedious task and still, there is a need for a qualitative study to know about the crops and

their likely weeds, pathogens, and pests. The present methodology identifies diseases in plants to increase the productivity of crops in fields. The system is developed for the benefit of farmers and the agricultural sector. In this system, deep learning models were used for the detection of plant diseases using different leaf images to identify whether the leaf is healthy or diseased.

Advantage:experimental results and comparison between two models SRCNN and Bicubic demonstrates the accuracy to recognize the correct disease in plants. Out of these two models,SRCNN gives an accuracy rate of 99.175 % in recognizing plant diseases.

Disadvantage: Diseases are not the specific problem in the agricultural sector but crops growing in good soil and getting nutritious food protect the plant from various pest attacks.

5.Farmer’s Assistant: A Machine Learning BasedApplication for Agricultural Solutions.

Author: Shloka Gupta ,Nishit Jain ,Akshay Chopade.

Observation: In this paper, we propose a user-friendly web application system based on machine learning and web-scraping called the ‘Farmer’s Assistant’. With our system, we are successfully able to provide several features - crop recommendation using Random Forest algorithm, fertilizer recommendation using rule based classification system, and crop disease detection using EfficientNet model on leaf images. The user can provide the input using forms on our user interface and quickly get their results. In addition, we also use the LIME interoperability method to explain our predictions on the disease detection image, which can potentially help understand why our model predicts what it predicts, and improve the datasets and models using this information.

Advantages: For crop recommendation and fertilizer recommendation, we can provide the availability of the same on the popular shopping websites, and possibly allow users to buy the crops and fertilizers directly from our application.

Disadvantages: To provide fine-grained segmentation of the diseased

portion of the dataset.This is not possible due to lack of such data. However, in our application,we can integrate a segmentation annotation tool where the users might be able to help us with the lack. Also, we can use some unsupervised algorithms to pin-point the diseased areas in the image. We intend to add these features and fix these gaps in our upcoming work.

6.Plant Disease Detection Using Convolution Neural Network

Author: Prakanshu Srivastava,Kritika Mishra,Vibhav Awasthi,Vivek Kumar Sahu and Mr.Pawan Kumar Pal.

Observation: In this paper, an approach of using deep learning methods was explored to automatically classify and detect plant diseases from leaf images.The entire procedure was described ,respectively,from collecting the pictures used for training and validation to image pre-processing and augmentation and eventually the procedure of coaching the deep CNN and fine-tuning .Different tests were performed to see the performance of the newly created model.

Advantage: The proposed model using CNN was trained using images from plant village dataset and attained an accuracy of 94.87% in identifying the diseased plants with the help of image processing by OpenCV.

Disadvantage: In the field of disease recognition,there was no comparison with related results, using the precise technique.

7. Semi-automatic leaf disease detection and classification system for soybean culture IET Image Processing, 2018 .

Author: Sukhvir Kaur,Shreelekha Pandey,Shivani Goel

Observation: This paper mainly focuses on detecting and classifying the leaf disease of soybean plants. Using SVM the proposed system classifies the leaf disease in 3 classes like i.e. downy mildew, frog eye, and septoria leaf blight etc. The proposed system gives maximum average classification accuracy reported is ~90% using a big dataset of 4775 images.

Advantages: The system helps to compute the disease severity.

Disadvantages: The system uses leaf images taken from an online

dataset, so cannot be implemented in real time.

Algorithm used: SVM.

8.A Brief Review on Plant Disease Detection Using in Image Processing

Author: Rajneet kaur,Miss.Manjeet Kaur

Observation: As, SVM is very complex in calculations and it is not the cost effective testing of each instance and inaccurate to wrong inputs. KNN algorithm is an effective classifier that would be used to minimize the computational cost. In previous research it has proved that KNN has a high accuracy rate. KNN classifier obtains highest result as compared to SVM. The comparison would be based upon two parameters: Accuracy and Detection time. The study reviews and summarizes some techniques that have been used for plant disease detection. A novel approach for classification of plant disease has been proposed.

Advantage: Using digital image processing method, the disease detection in plants is efficient, less time consuming and accurate. This technique saves time, efforts, labors and use of pesticides.

Disadvantage: Detection and recognition of diseases in plants mistreatment digital image method is extremely effective in providing symptoms of characteristic diseases at its.

2.2 REFERENCES:

1. Semi-automatic leaf disease detection and classification system for soybean culture IET Image Processing, 2018 .

2. Cloud Based Automated Irrigation And Plant Leaf Disease Detection System Using An Android Application. International Conference on Electronics, Communication and Aerospace Technology,ICECA 2017.

3. Ms. Kiran R. Gavhale, Ujwalla Gawande, Plant Leaves Disease detection using Image Processing Techniques, January 2014.
https://www.researchgate.net/profile/UjwallaGawande/publication/314436486_An Overview_of_the_Research_on_Plant_Leaves_Disease_detection_using_Image _Processing_Techniques/links/5d3710664585153e591a3d20/An-Overview of-the

-Research-on-Plant-Leaves-Disease detection-using-Image-Processing Techniques.pdf.

4. Duan Yan-e, Design of Intelligent Agriculture Management Information System Based on IOTll, IEEE,4th, Fourth International reference on Intelligent Computation Technology and Automation, 2011 https://ieeexplore.ieee.org/document/5750779.

5. R. Neela, P. Fertilizers Recommendation System For Disease Prediction In Tree Leave International journal of scientific & technology research volume 8, issue 11, November 2019 http://www.ijstr.org/final-print/nov2019/Fertilizers-Recommendation-System-For -Disease-Prediction-In-Tree-Leave.pdf .

6. Swapnil Jori1, Rutuja Bhalshankar2, Dipali Dhamale3, Sulochana Sonkamble , Healthy Farm:Leaf Disease Estimation and Fertilizer Recommendation System using Machine Learning, International Journal of All Research Education and Scientific Methods (IJARESM), ISSN:2455-6211.

7. Detection of Leaf Diseases and Classification using Digital Image Processing International Conference on Innovations in Information, Embedded and Communication Systems(ICIIECS), IEEE,2017.

8. Shloka Gupta ,Nishit Jain ,Akshay Chopade, Farmer's Assistant: A Machine Learning BasedApplication for Agricultural Solutions.

2.3 PROBLEM STATEMENT DEFINITION:

1.A Web Application Based On Artificial Intelligence For Crop Disease Detection And Fertilizer Recommendation.

2.Using Artificial Intelligence For Farmers,Crop Disease Detection, Fertilizer Recommendation, Pest Recognition And Proper Guideline.

3.Predicting Efficient Fertilizer Recommendation Based On Soil Features And Environmental Parameter Using AI.

4.The Project Has Been Proposed To Analyze Available Datasets And To Predict Fertilizer Requirements For Two Different Crops Based On Soil Features And Environmental Parameters Using Appropriate Machine Learning Models.

5.Farmers Do Not Have Enough Of Knowledge About Soil Nutrients

Contents Such As Nitrogen, Phosphorus And Potassium In The Land.

6.A Fertilizer Requirement Prediction Model Would Be Very Useful For Obtaining Higher Yield Of Different Crops.

7.Design and Implementation of Fertilizer Recommendation System for Farmers.

8.A nutrient recommendation system for soil fertilization based on AI.

9.Develop web-based solutions for Cotton Crop health monitoring and suggest remedial actions.

10.Develop smart & affordable solutions to protect crops from wild animals.

3.IDEATION & PROPOSED SOLUTION

3.1EMPATHY MAP CANVAS

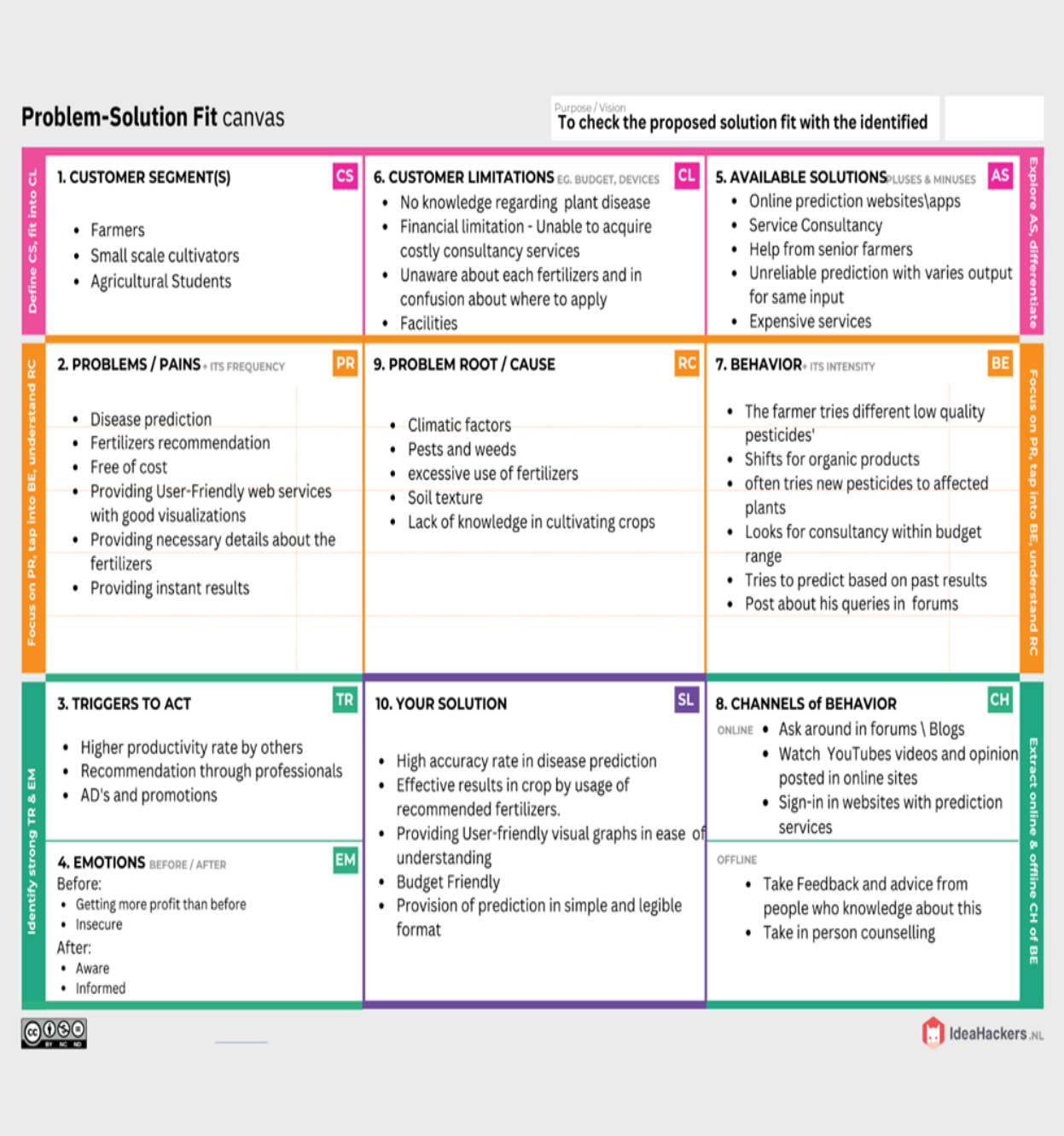
3.2 IDEATION AND BRAINSTORMING:

3.3 PROPOSED SOLUTION:

S. NO	Parameter	Description
1.	Problem Statement (Problem to be solved)	Banks are not able to resolve the queries of customers at all times related to the products or services in a satisfactory way which in turn hinders the customer satisfaction. Customers need to visit banks frequently for simple queries.
2.	Idea / Solution description	In order to guide the customers throughout all the financial services provided by the bank, an intelligent system has to be introduced to provide people with the best solution possible.

3.	Novelty / Uniqueness	Chatbots developed using AI should be able to answer any general banking queries on account creation, loan, net banking, other services etc. It addresses the queries of customers immediately and effectively in a cost efficient manner.
4.	Social Impact/ Customer Satisfaction	In order to attain the user satisfaction issues associated with banking services, chatbot will provide personal and efficient communication between the user and the bank. It is built to be the overall virtual assistant that can facilitate customers to ask banking-related questions without visiting the bank or calling up customer service centers as well as providing them with relevant suggestions.
5.	Business Model (Revenue Model)	Employing a chatbot will be a cost-effective solution to clear customer queries for banks. It eliminates the need for a massive customer care workforce and even reduces the workload of the bank employees whose efforts can be used elsewhere.
6.	Scalability of the Solution	AI Chatbots provides 24/7 service to clear all customer queries and guide them through all the banking processes. It supports voice assistance features and maintains a confidential conversation with customers. It can be scaled as per the requirements of the bank to include answers to queries related to any new feature or service introduced by the bank.

3.4 PROBLEM SOLUTION FIT :



4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

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 Gmail Registration through Link Registration through Google Registration through Mobile number
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP Confirmation via Call
FR-3	Authentication	User Id / Name and Password
FR-4	Authorization	User Id / Name and Password or OTP verification
FR-5	External interfaces	Upload the image and verify with captcha
FR-6	Transaction	Through Net banking, UPI, Debit and Credit
FR-7	Reporting	Click report / Help option

4.2 NON-FUNCTIONAL REQUIREMENTS :

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

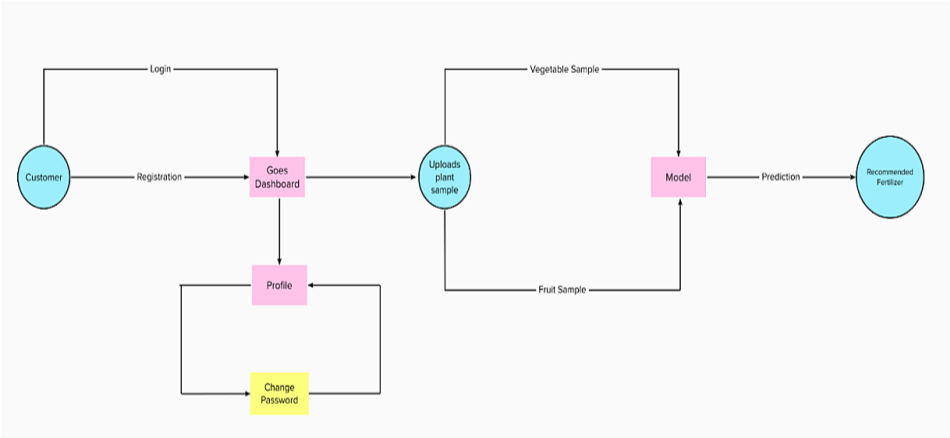
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Even illiterate can use
NFR-2	**Security**	Allow access to Camera / Gallery / Drive to upload image
/NFR-3	**Reliability**	Backup
NFR-4	**Performance**	Splash screen must be not more than 3 seconds for users that access the website using mobile connection
NFR-5	**Availability**	–
NFR-6	**Scalability**	–

5.PROJECT DESIGN

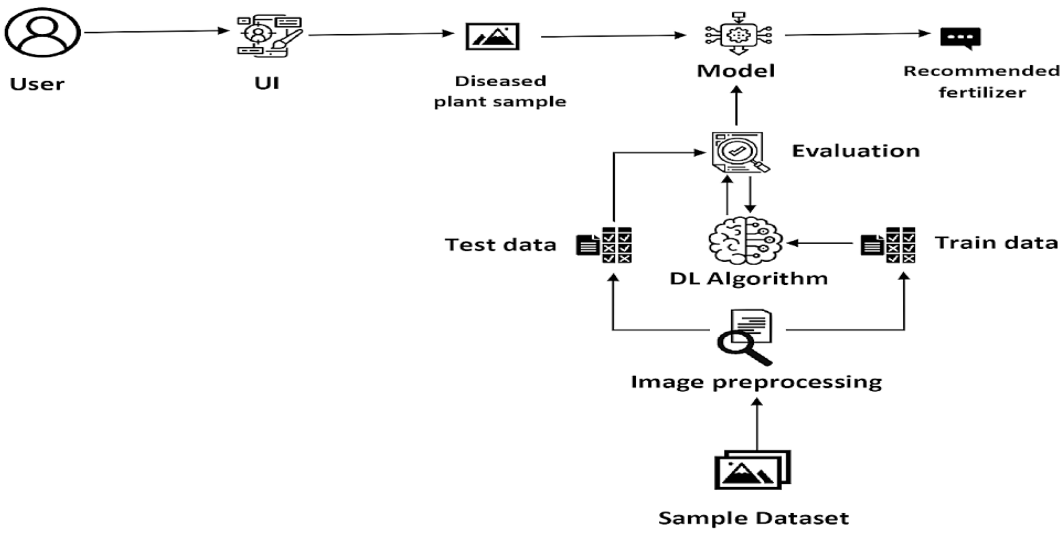
5.1 DATA FLOW DIAGRAMS

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.



5.2 SOLUTION & TECHNICAL ARCHITECTURE :



5.3 USER STORIES :

User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile User / Web 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 / dashboard	Low	Sprint-4
	Login	USN-2	As a user, I can log into the application by entering email & password	I can log into my account using my username / email with password	Low	Sprint-4
	Dashboard	USN-3	As a user, I can access all my personal information which will be stored in my profile by using dashboard	I can access my personal information such as phone number, email, upload history	Medium	Sprint-1,2
	Prediction Page	USD-4	As a user, I can upload my sample plant image to known recommended fertilizers	I can upload the sample plant image	High	Sprint-2,3

6. PROJECT PLANNING & SCHEDULING :

6.1 Sprint Planning & Estimation :

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Arthy, Madhu Preetha, Nandhini, Vijayashree
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	5	High	Arthy, Madhu Preetha, Nandhini, Vijayashree
Sprint-2		USN-3	As a user, I can register for the application through mobile number	5	Low	Arthy, Madhu Preetha, Nandhini, Vijayashree
Sprint-1		USN-4	As a user, I can register for the application through Gmail	5	Medium	Arthy, Madhu Preetha, Nandhini, Vijayashree

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	5	High	Arthy, Madhu Preetha, Nandhini, Vijayashree
Sprint-2	Dashboard	USN-6	As a user, I can access all my personal information which will be stored in my profile by using dashboard	5	Medium	Arthy, Madhu Preetha, Nandhini, Vijayashree
Sprint-3		USN-7	As a user, I can change my password	10	Low	Arthy, Madhu Preetha, Nandhini, Vijayashree
Sprint-4	Support	USN-8	As a user, I can get my help and support from help option	10	Medium	Arthy, Madhu Preetha, Nandhini, Vijayashree
Sprint-3	Landing Page	USD-9	As a user, I can see the free trials and subscription	10	High	Arthy, Madhu Preetha, Nandhini, Vijayashree
Sprint-4		USD-10	As a user, I can access the FAQ asked by other uses	10	Low	Arthy, Madhu Preetha, Nandhini, Vijayashree
Sprint-2	Prediction Page	USD-11	As a user, I can upload my sample plant image to known recommended fertilizers	10	High	Arthy, Madhu Preetha, Nandhini, Vijayashree

6.2 Sprint Delivery Schedule :

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

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

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

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

6.3 Reports from JIRA :

7.CODING & SOLUTIONING :

7.1 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, 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("vegetabledata.h5")
model1=load_model("fruitdata.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('/login')
def login():
    return render_template('login.html')

@app.route('/signup')
def signup():
    return render_template('signup.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('precautions - veg.xlsx')
            result = df.iloc[preds]['caution']
            return result
        else:
            preds = model1.predict(x)
            preds=np.argmax(preds)
            print(preds)
            df=pd.read_excel('precautions - fruits.xlsx')
            result = df.iloc[preds]['caution']
            return result
        return result
if _name_ == "_main_":
    app.run(debug=False)
```

7.2 Home.html:

```
<!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:#C2C5A8;
  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;
}

@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 {
  max-width: 1000px;
  position: relative;
  margin: auto;
}


.text {
  color: #f2f2f2;
```

```
  font-size: 15px;
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}


.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;
}


.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}
}


@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>
    <a href="{{ url_for('login')}}">Login</a>
    <a href="{{ url_for('signup')}}">Signup</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 is one of the major sectors worldwide. 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%;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", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
  setTimeout(showSlides, 2000); // Change image every 2 seconds
}
</script>
```

```
</body>
</html>
```

7.3 Predict.html:

```
<!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://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300'
rel='stylesheet' type='text/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'>
<link href="{{ url_for('static', filename='css/final.css') }}" 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;
}
```

```
.login{
margin-top:-70px;
}
body {

  background-color:#C2C5A8;
  background-repeat: no-repeat;
  background-size:cover;
  background-position: 0px 0px;
  }
.login{
        margin-top:100px;
}

.container {
  margin-top:40px;
  padding: 16px;
}
select {
        width: 100%;
        margin-bottom: 10px;
        background: rgba(255,255,255,255);
        border: none;
        outline: none;
        padding: 10px;
        font-size: 13px;
        color: #000000;
        text-shadow: 1px 1px 1px rgba(0,0,0,0.3);
        border: 1px solid rgba(0,0,0,0.3);
        border-radius: 4px;
        box-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0 1px 1px
rgba(255,255,255,0.2);
        -webkit-transition: box-shadow .5s ease;
        -moz-transition: box-shadow .5s ease;
        -o-transition: box-shadow .5s ease;
        -ms-transition: box-shadow .5s ease;
```

```
        transition: box-shadow .5s ease;
}

</style>
</head>

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

<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%;">
  </div>
</div>
<div class="container">
      <div id="content" style="margin-top:2em">
            <div class="container">
              <div class="row">
                  <div class="col-sm-6 bd" >

                    <br>
                        <img src="{{url_for('static',filename='images/predict.png')}}" style="height:450px;width:550px"class="img-rounded" alt="Gesture">
                  </div>
                  <div class="col-sm-6">
                        <div>
                              <h4>Drop in the image to get the prediction  </h4>
                  <form action = "" id="upload-file" method="post" enctype="multipart/form-data">
                        <select name="plant">

                              <option value="select" selected>Select plant type</option>
                              <option value="fruit">Fruit</option>
                              <option value="vegetable">Vegetable</option>
            </select><br>
                        <label for="imageUpload" class="upload-label"
```

```
style="background: #28272c;">
                                    Choose...
                                </label>
                                <input type="file" name="image" id="imageUpload"
accept=".png, .jpg, .jpeg">
                            </form>


                            <div class="image-section" style="display:none;">
                                <div class="img-preview">
                                    <div id="imagePreview">
                                    </div>
                                </div>
                                <div>
                                    <button type="button" class="btn btn-info btn-lg "
id="btn-predict" style="background: #28272c;">Predict!</button>
                                </div>
                            </div>

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

                            <h3>
                                <span id="result" style="font-size:17px; "> </span>
                            </h3>

                        </div>
                            </div>

                          </div>
                        </div>
                        </div>
    </div>
</body>

<footer>
    <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>
```

```
</html>
```

7.4 Login.html:

```
<!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 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;
}
.sign-up-form
{
    width: 300px;
    align-items: center;
    justify-content: center;
    box-shadow: 0 0 3px 0 rgba(0,0,0,0.3);
    padding: 20px;
```

```
    border-radius: 20px;
    margin: 5% auto 0;
    margin-left: 450px;
    text-align: center;
    }
.sign-up-form h1
{
    color: #1c8adb;
    margin-bottom: 30px;
}
.input-box
{
    border-radius: 20px;
    padding: 10px;
    margin: 10px 0;
    width: 95%;
    border: 1px solid #999;
    outline: none;
}
button
{
    color: #fff;
    width: 100%;
    padding: 10px;
    border-radius: 20px;
    font-size: 15px;
    margin: 10px 0;
    border: none;
    outline: none;
    cursor: pointer;
}
.signup-btn
{
    background-color: #1c8adb;
}
.google-btn
{
```

```
    background-color: #21afde;
}
a{
    text-decoration: none;
}
hr{
    margin-top: 20px;
    width: 80%;
}
.or{
    background: white;
    width: 30px;
    margin: -19px auto 10px;
}
img{
    width:70px;
    margin-top: -60px
}
</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  href="{{ url_for('home')}}">Home</a>
    <a href="{{ url_for('prediction')}}">Predict</a>
    <a class="active" href="{{ url_for('login')}}">Login</a>
    <a href="{{ url_for('signup')}}">Signup</a>

  </div>
</div>

<div style="background-color:#ffffff;">
```

```
<div style="float:left;">
<div style="font-size:50px;font-family:'Times New Roman', Times,
serif;color:#28272c;;text-align:center;margin-left:450px;padding-top:10%;class:sign-up-
form">
<b>Log-in</b></div>
<div class = "sign-up-form">
   <form>
      <input type="email" class="input-box" placeholder="Your email">
      <input type="password" class="input-box" placeholder="Your Password">
      <button type="button" class="signup-btn">Log in</button>
      <hr>
      <p class="or">OR</p>
      <button type="button" class="google-btn">Sign Up with google</button>
      <p>Do you have an account ? <a href="{{ url_for('signup')}}">Sign Up</a></p>
   </form>
 </div><br><br>
</div>
</div>
</div>

<div class="home">

<br>

</div>

</body>
</html>
```

Sign up.html:

```
<!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 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;
}
.sign-up-form
{
    width: 300px;
    align-items: center;
    justify-content: center;
    box-shadow: 0 0 3px 0 rgba(0,0,0,0.3);
    padding: 20px;
    border-radius: 20px;
    margin: 5% auto 0;
    margin-left: 450px;
    text-align: center;
    }
.sign-up-form h1
{
    color: #1c8adb;
    margin-bottom: 30px;
}
```

```
.input-box
{
    border-radius: 20px;
    padding: 10px;
    margin: 10px 0;
    width: 95%;
    border: 1px solid #999;
    outline: none;
}
button
{
    color: #fff;
    width: 100%;
    padding: 10px;
    border-radius: 20px;
    font-size: 15px;
    margin: 10px 0;
    border: none;
    outline: none;
    cursor: pointer;
}
.signup-btn
{
    background-color: #1c8adb;
}
.google-btn
{
    background-color: #21afde;
}
a{
    text-decoration: none;
}
hr{
    margin-top: 20px;
    width: 80%;
}
.or{
```

```
    background: white;
    width: 30px;
    margin: -19px auto 10px;
}
img{
    width:70px;
    margin-top: -60px
}
</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  href="{{ url_for('home')}}">Home</a>
    <a href="{{ url_for('prediction')}}">Predict</a>
    <a href="{{ url_for('login')}}">Login</a>
    <a class="active" href="{{ url_for('signup')}}">Signup</a>

  </div>
</div>

<div style="background-color:#ffffff;">
<div style="float:left;">
<div style="font-size:50px;font-family:'Times New Roman', Times,
serif;color:#28272c;;text-align:center;margin-left:450px;padding-top:10%;class:sign-up-
form">
<b>Sign Up Now </b></div>
<div class = "sign-up-form">
    <form>
        <input type="email" class="input-box" placeholder="Your email">
        <input type="password" class="input-box" placeholder="Your Password">
        <button type="button" class="signup-btn">Sign up</button>
```

```
        <hr>
        <p class="or">OR</p>
        <button type="button" class="google-btn">Sign Up with google</button>
        <p>Do you have an account ? <a href="{{ url_for('login')}}">Sign in</a></p>
    </form>
 </div><br><br>
</div>
</div>
</div>

<div class="home">

<br>

</div>

</body>
</html>
```

Final.css:

```
.img-preview {
    width: 256px;
    height: 256px;
    position: relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}

.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}
```

```
input[type="file"] {
    display: none;
}

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #28272c;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #C2C5A8;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #28272c; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
```

Main.js:

```
$(document).ready(function () {
    // Init
```

```
$('.image-section').hide();
$('.loader').hide();
$('#result').hide();

// Upload Preview
function readURL(input) {
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = function (e) {
            $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
            $('#imagePreview').hide();
            $('#imagePreview').fadeIn(650);
        }
        reader.readAsDataURL(input.files[0]);
    }
}
$("#imageUpload").change(function () {
    $('.image-section').show();
    $('#btn-predict').show();
    $('#result').text('');
    $('#result').hide();
    readURL(this);
});

// Predict
$('#btn-predict').click(function () {
    var form_data = new FormData($('#upload-file')[0]);

    // Show loading animation
    $(this).hide();
    $('.loader').show();

    // Make prediction by calling api /predict
    $.ajax({
        type: 'POST',
        url: '/predict',
        data: form_data,
```

```
            contentType: false,
            cache: false,
            processData: false,
            async: true,
            success: function (data) {
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').text('Prediction: '+data);
                console.log('Success!');
            },
        });
    });

});
```