

“Project Report”

“Fertilizers-Recommendation-System-For-Disease-Prediction ”

Team ID: PNT2022TMID0681

Submitted by

1) Team Leader Sowmiya.K

2) Team Member

puviyaal.V
Sivapriya.S
Shalini.B

Table of contents

1. INTRODUCTION

- 1.1 Project Overview
- 1.2 Purpose

2 LITERATURE SURVEY

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

3 IDEATION & PROPOSED SOLUTION

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

4 REQUIREMENT ANALYSIS

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

5 PROJECT DESIGN

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

6 PROJECT PLANNING & SCHEDULING

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

7 CODING & SOLUTIONING (Explain the features added in the project along with code)

- 7.1 Feature 1
- 7.2 Feature 2
- 7.3 Database Schema (if Applicable)

8 TESTING

- 8.1 Test Cases
- 8.2 User Acceptance Testing

9 RESULTS

- 9.1 Performance Metrics

10 ADVANTAGES & DISADVANTAGES

11 CONCLUSION**12 FUTURE SCOPE****13 APPENDIX**

Source Code

GitHub & Project Demo Link

1)Introduction**1.1.**

Project Overview

Plant disease prediction helps in the detection and recognition of the plant diseases. The images of plants are captured and analyzed for certain symptoms using Computer vision and image processing. By identifying the disease, the deficit nutrients that lead to the disease are found. Based on the available data on fertilizers, the necessary nutrient rich fertilizers are recommended.

1.2.Purpose

The plant diseases may lead to abnormal functionalities which may end up with the death of the plant. The project aims at recognizing the symptoms at the early stages. The project also aims at guiding the farmers with the proper choice of the fertilizers that are required to counter the deficiency of the nutrients that cause the disease.

2)Literature Survey**2.1 Existing Problem****2.2) References**

Title	Technique	Links
Soil Based Fertilizer Recommendation System for Crop Disease Prediction System – P.Pandi Selvi,P.Poornima	Long or Short Term Memory Algorithm	http://www.ijetjournal.org/volume-8/issue-2/IJETA-V8I2P1
IOT based Crop Recommendation,Crop Disease Prediction and Its Solution – Rani Holambe, Pooja Patil, Padmaja Pawar Hrushikesh Joshi,Saurabh Salunkhe	Crop Recommendation System,Crop Disease Prediction,Internet of things,Machine Learning	https://arxiv.org/pdf/2204.11340.pdf
Farmer's Assistant:A Machine Learning Based Application for Agricultural Solutions-Shloka Gupta,Aparna Bhonde,Akshay Chopade,Nishit Jain	Image Analysis,Deep Learning,Machine Learning	https://www.irjet.net/archives/V7/i10/IRJET-V7I1004.pdf
R. Neela, P. Fertilizers Recommendation System For Disease Prediction In Tree Leave International journal of scientific & technology research volume 8, issue 11, november 2019	Adding a CNN(Convolutional neural network) and SVM(Support Vector Machine)	http://www.ijstr.org/final-print/nov2019/
Plant Disease Detection Using Image Processing and Machine Learning	Random Forest classifier, a combination of multiple decision trees is used where each tree is trained by using different subsets of the whole dataset to reduce the overfitting and improves the accuracy of the classifier.	https://arxiv.org/abs/2106.10698
Fertilizers Recommendation System for Disease Prediction in Tree Leaves	Support Vector Machine (SVM) algorithm classifies the leaf image as normal or affected. And it is used to identify a function F_x which obtain the hyper-plane.	https://www.semanticscholar.org/paper/Fertilizers-Recommendation-Disease-In-Neela-Nithya/495379d3ef2b461fabd2de8d0605c16

[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, Ujwala Gawande, Plant Leaves Disease detection using Image Processing Techniques, January 2014.
<https://www.researchgate.net/profile/UjwalaGawande/publication/31>

[4436486 An Overview of the Research on Plant Leaves Disease detection using Image Processing Techniques/links/5d37106_64585153e591a3d20/An-Overview-of-the-Research-on-Plant-Leaves-Diseae detection-using-ImageProcessing Techniques.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] 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-RecommendationSystem-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 Based Application for Agricultural Solution

2.3 Problem Statement Definition

This project aims at providing a system to

support the cultivators in choosing the right fertilizers for their plants to counter the deficiency of nutrients that cause various infections and diseases. The below blocks define the problems faced by the different users and the solutions that are provided by the system.

I am	Describe customer with 3-4 key characteristics - who are they?	Describe the customer and their attributes here
I'm trying to	List their outcome or "job" the core about - what are they trying to achieve?	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way - what bothers them most?	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists - what needs to be solved?	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view - how does it impact them emotionally?	Describe the emotions the result from experiencing the problems or barriers

Reference: <https://miro.com/templates/customer-problem-statement/> Example:



3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

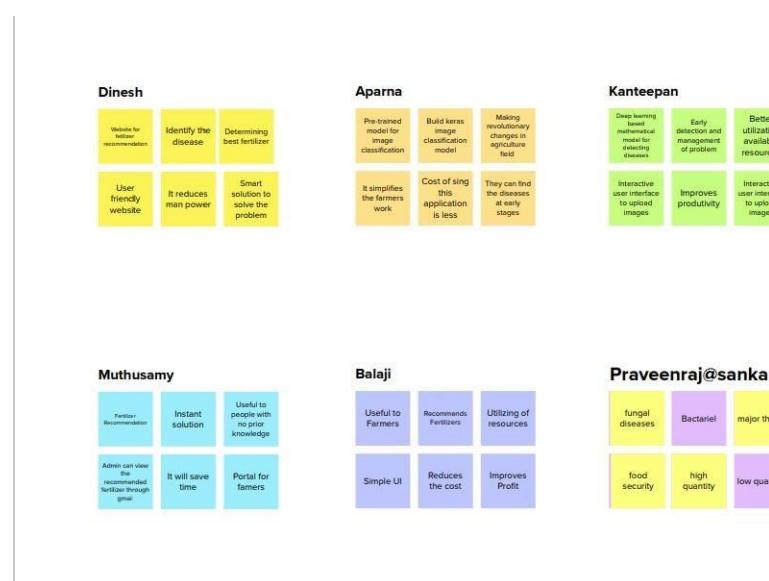
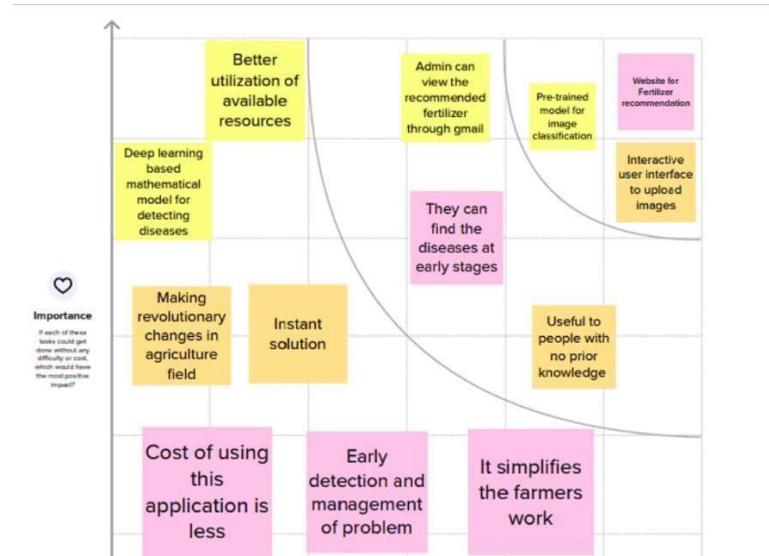
Agriculture is the main aspect of the economic development of a country. Agriculture is the heart and life of most Indians.

By understanding their feelings and problems, we can create a better product and contribute to their lives. For our project, we are getting surveys from farmers to understand what they truly require and desire.



3.2 Ideation & Brainstorming

Ideation and Brainstorming are performed to generate ideas and solutions. Brainstorming is a group activity unlike ideation.



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.

10 minutes
1-6 Collaborate

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

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

Set a goal
Think about the problems you'll be focusing on solving in the brainstorming session.

Fungal diseases

Interested in learning more?
Check out the Meta Think Kit website for additional tools and resources to help your team collaborate, innovate, and move ideas forward with confidence.

Define your problem statement
What problems are you trying to solve? Frame your problem as a question. This will be the focus of your brainstorming.

15 minutes

Problem
Identify the disease on plants using deep learning techniques and to recommend the fertilizers for reducing the diseases.
2. Provide website information for recommended fertilizer.

Key values of collaboration
Focus on mutual and productive success.

- Stay on topic
- Encourage solid ideas
- Challenge others
- Get to the point
- Promote the win

...
More weight now... Define problem statement first!

3. Proposed Solution

An automated system that takes the images of plant parts as input identifies different diseases on plants by checking the symptoms shown on the leaves of the plant is built . Deep learning techniques are used to identify the diseases and suggest the fertilizes that can help cure the disease. The user need not consult any

specialist for identification of diseases that affected the leaves or for the recommendation of the fertilizers.

Proposed Solution Template:

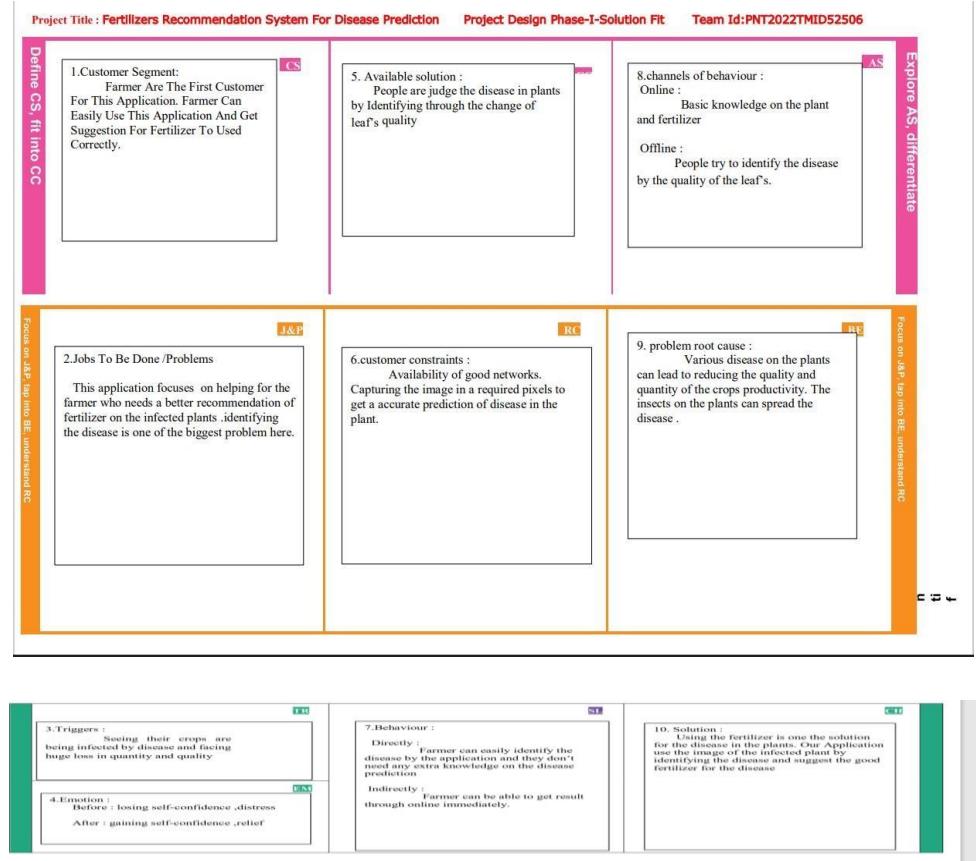
Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Agriculture is having a great impact on the country's economy. Different diseases effect plant that reduces their production and is a major threat to food security. The major problems that the farmers of our country are currently facing includes Crop Failure, Lack of adequate knowledge, Crop damage due to ignorance/carelessness, Lack of

		professional assistance. Inaccessibility to agro-tech solutions. Most of the diseases are detected in later stage that is manually which is time consuming and results in heavy loss so it is important to build an automated system that detects disease at early stage and provides fertilizer recommendation accordingly.
2.	Idea / Solution description	An automated system is built that takes the input as picture of leaves which is uploaded by the user, identifies different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the fertilizer needed for the plant.
3.	Novelty / Uniqueness	It does not require user to consult any specialist for identification of diseases that affected the leaves
4.	Social Impact / Customer Satisfaction	and the fertilizers that is required for the same. It detects Plant disease at their early stage.
5.	Business Model (Revenue Model)	The whole process of identifying disease and recommendation of fertilizer happens just by uploading image so it is user friendly. It helps farmers to get good yield out of the crop. People will get good quality food products.
6.	Scalability of the Solution	Social media is the best way to spread the word about our application. And with the influencers we can reach out to people. Clustering and targeting the farmers for identifying diseases on their plants and recommending them fertilizers for the same.

Problem Solution fit:

The Problem-Solution Fit means that the solution that is realized can actually solve the problem that the customer faces.



4. Requirement Analysis

4.1 Functional requirement

Functional requirement :

Following are the functional requirements of the proposed solution .

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

Following are the non-functional requirement of the proposed solution

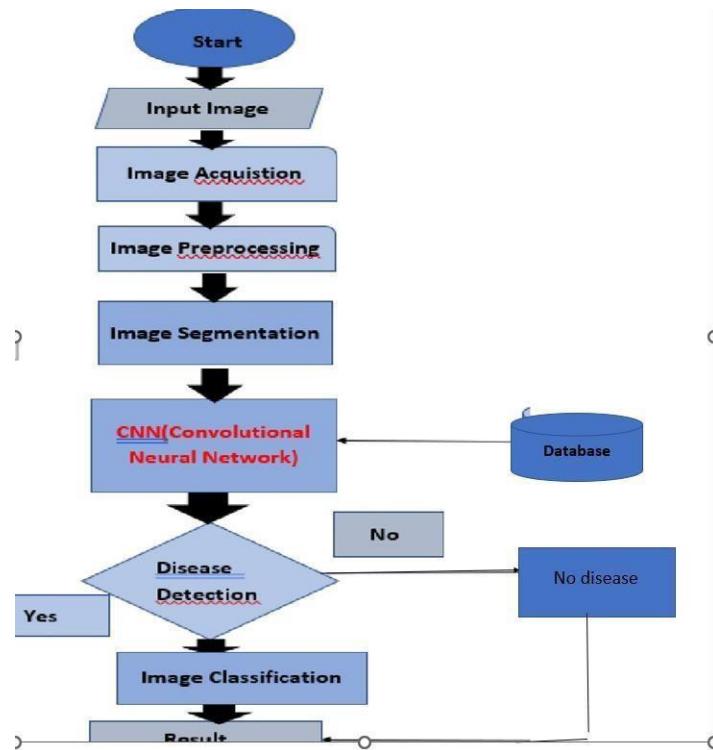
NFr.no	Non-functional requirement	Description
Nfr-1	Usability	Datasets of all the leaf is used to detecting the disease that present in the leaf.
Nfr-2	Security	The information belongs to the user and leaf are secured highly.
Nfr-3	Reliability	The leaf quality is important for the predicting the disease 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 Diagrams

A data flow diagram or DFD(s) maps out the flow of information for any process or system. DFDs help you

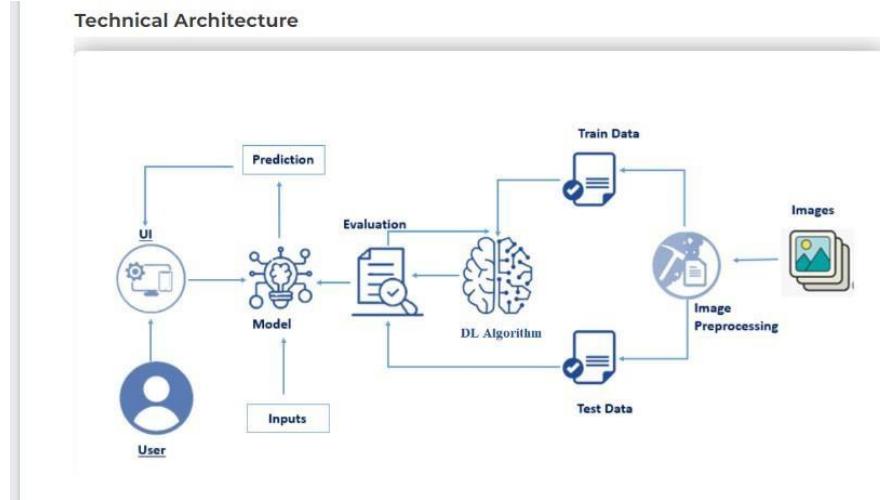
better understand process or system operation to discover potential problems, improve efficiency, and develop better processes.



5.2 Solution & Technical Architecture

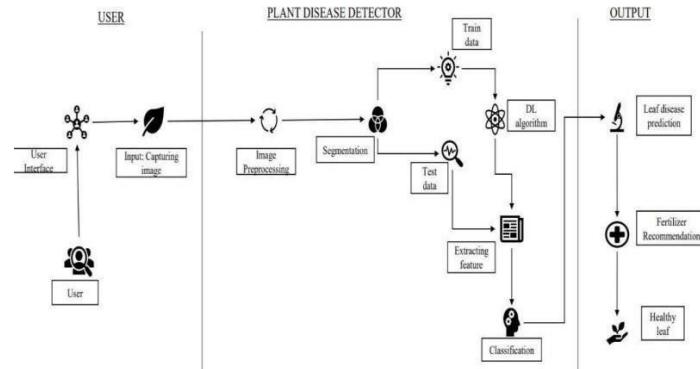
Solution Architecture:

Solution architecture is the process of developing solutions based on predefined processes, guidelines and best practices with the objective that the developed solution fits within the enterprise architecture in terms of information architecture, system portfolios, integration requirements, etc.



5.2 Technical Architecture:

Technical architecture involves the development of a technical blueprint regarding the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.



5.3 User Stories

An informal, generic explanation of a software feature written from the viewpoint of the end user is known as a user story. Its objective is to explain how a software feature will benefit the user

Table -1: Components & Technologies :

S.NO	Component	Description	Technology
1,	User Interface	How user interacts with the website.	HTML,CSS, etc.,
2,	Disease Prediction	Here the disease in the leaf is predicted	Keras,CNN.
3.	Fertilizer Recommendation	The fertilizer is recommended for the predicted disease	User interface, HTML, CSS.
4.	Dataset	The training and testing data are collectively stored	Kaggle.com, data.gov, UCI machine learning repository, etc.
5.	File Storage	File storage requirements	IBM, Local File system.
6,	Modules	Purpose of deep learning modules	Image Recognition Modules,etc.

Table – 2: Application Characteristics:

S.NO	Characteristics	Description	Technology
1.	Opensource Framework	List of the opensource framework used	Open source-PyCharm, anaconda navigator, flask framework.
2.	Login	List of the access control implementation	Security - OWASP
3.	Scalable Architecture	Justify the scalable architecture	PyCharm
4.	Availability	Justify the availability of website	Web application access to all.
5.	Performance	Design consideration for the performance of the website	Convolutional Neural Networks.

6. Project Planning and Scheduling

6.1 Sprint Planning & Estimation

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

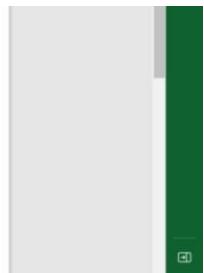
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Image Processing.	USN-1	As a user, I can retrieve useful information about the images.	1	Low	C.J.Dinesh kumar T.R.S.Praveenraj@sankaran T.S.Balaji S.Apama A.Muthusamy P.Kanteepan
Sprint-2	Model Building for Fruit Disease Prediction.	USN-2	As a user, I can able to predict fruit disease using this model.	1	Medium	C.J.Dinesh kumar T.R.S.Praveenraj@sankaran T.S.Balaji S.Apama A.Muthusamy P.Kanteepan
Sprint-2	Model Building for Vegetable Disease Prediction.	USN-3	As a user, I can able to predict vegetable disease using this model.	2	Medium	C.J.Dinesh kumar T.R.S.Praveenraj@sankaran T.S.Balaji S.Apama A.Muthusamy P.Kanteepan
Sprint-3	Application Building.	USN-4	As a user, I can see a web page for Fertilizers Recommendation System for Disease Prediction.	2	High	C.J.Dinesh kumar T.R.S.Praveenraj@sankaran T.S.Balaji S.Apama A.Muthusamy P.Kanteepan
Sprint-4	Train The Model on IBM Cloud.	USN-5	As a user, I can save the information about Fertilizers and crops on IBM cloud	2	High	C.J.Dinesh kumar T.R.S.Praveenraj@sankaran T.S.Balaji S.Apama A.Muthusamy P.Kanteepan

Project Tracker, Velocity & Burndown Chart (4 Marks)

6.2 Sprint Delivery Schedule

Agile sprints typically last from one week to one month. The goal of sprints is to put pressure on teams to innovate and deliver more quickly, hence the shorter the sprint, the better

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	26 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	30 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	05 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	10 Nov 2022



6.3 Reports from JIRA

Backlog:

A backlog is a list of issues that's related to the project and the functions of the system. It makes it simple to make, store, manage a variety of problems including the ones the team is working on.

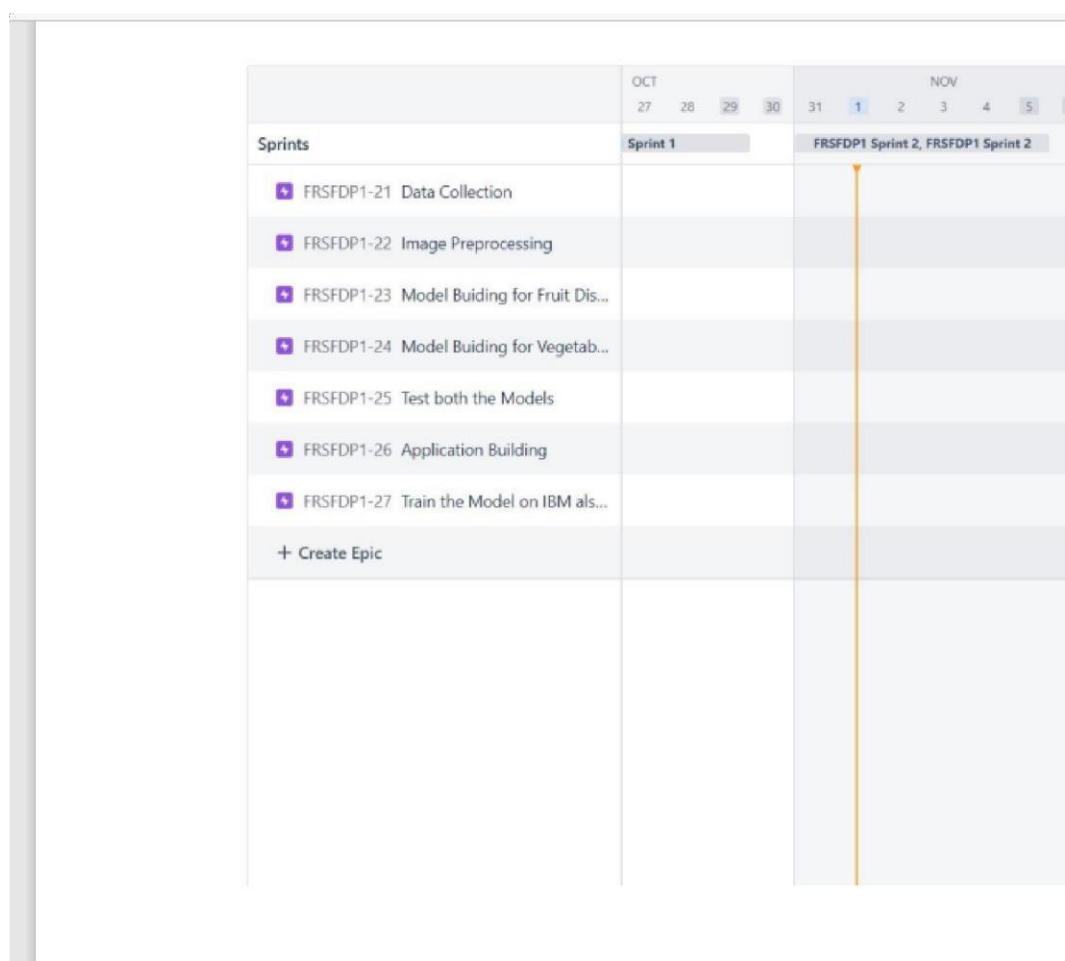
The screenshots show the Jira Software interface for a project named "Fertilizers-Recommendation-System-For-Disease-Prediction". The left sidebar includes sections for Planning (Roadmap, Backlog), Development (Code), and Project pages. The main area displays the Backlog, which is organized by sprints:

- Sprint 3 (7 Nov – 12 Nov):** Contains 4 issues (FRSFDP1-9, FRSFDP1-10, FRSFDP1-11, FRSFDP1-12) all labeled "TO DO".
- Sprint 4 (13 Nov – 16 Nov):** Contains 4 issues (FRSFDP1-13, FRSFDP1-14, FRSFDP1-15, FRSFDP1-16) all labeled "TO DO".
- Sprint 1 (24 Oct – 29 Oct):** Contains 4 issues (FRSFDP1-1, FRSFDP1-2, FRSFDP1-3, FRSFDP1-4) all labeled "TO DO".

Each issue is described with a brief summary and has a "Create" button next to it.

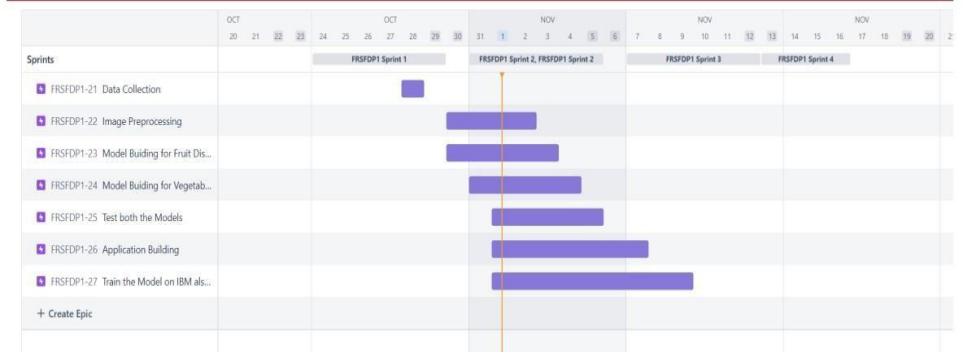
Board: A board reflects your team's process, tracking the status of work. The columns on the board represent the status of

your team's issues. The visual representation of the work helps in discussing and tracking of the progress of the project from start to finish.



Roadmap:

A roadmap offers quick and easy planning that helps teams better manage their dependencies and track progress on the big picture in real-time.



7. Coding and Solutioning Python – app.py:

```

import os
import numpy as np
import pandas as pd
from tensorflow.keras.models import load_model
# from tensorflow.keras.preprocessing import image
from werkzeug.utils import secure_filename

from flask import Flask, render_template, request

app = Flask(__name__)

#load both the vegetable and fruit models
model = load_model("vegetable.h5")
model1=load_model("fruit.h5")

#home page

```

```
@app.route('/')
def home():
    return render_template('home.html')

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

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

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

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

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

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

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

Feature 1: 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'>
    <script type="text/javascript" src="https://gc.kis.v2.scr.kaspersky-labs.com/FD126C42-EBFA-4E12-B309-
BB3FDD723AC1/main.js?attr=AMFGethlf4Q6r2IdpTrTqcDQGNLDU5Cbc3diYnUdLkg5mQrVB_td220
HUAsBJSd0oo80R0zM3rIPeFWfnEY4XCxQu4K0xMSqlshEoIB0zvYw0SsMypyUv4fnvKEjmJoj_Y6cI4ov
-6AM0kz3Sh3epkfq0gltnAPvvQBRdXqRmdqePVjlvvql280NZciS0qr5t0XGxJ0bSiwWT-
rH3cqaCKk05eP1Dx04mie1CjjsA_TtFLx15PUu0ed6soaj-F006-
1d40QxbJYBXUBefiUhzmOYCpsGIs10yQvA0huo8AUyWYB72dvs07U302hq8BmYBv98h13sSo8iXKxyKx4
FUsoMKixjxYP6hu0wwi7yv1E2rei3GhtP15YwHkWioQIPqvAmrlmaPtFZmF-
jE4_UUCi9IEKws8IduDiqQIFkxf03YT_sUC9gWmxKSpGbiewbWcgV-
wvdGENbuUxY18p9Db6jC6FVKRhqdMBianq63qv-
zZRMZbEpjzQT0DQAH3Yho4o4A00FIW2004q8Q80xt2kv928P_nBgS9H0gHI5EZxenbjfqANTs1rh8GGhB
d7RJaE8-
2AaqT6zbLf2tILJ8j4fk3bV1qsdw0fPmp6foJbDu4343XH36a0VGhsMLeVqcc30PSSe1pJbGE4_C_ExQd
0_uRSA40mRjnFwHdLo9SJc1qghyc5YQQil_utG48olMy9cC6z-iyKg1EeLKB43u-
q4S1UimRnuUsZW7drNWaijSfJPdmkm7lUj0POwQXPfnLa2_spc3FisWCOZ7dFuIgDciIu0yF8rio2X0Pz
6pZkGQW4Fwl6vWKRlplmHagJE1KXg58YSWwAT2DILlBjuSPiTbCHR9Ya_mAWX4C03v7xzJlaSK9jneEC
qctvKnH3RFgDS8ocfDcY651XNRkq6v1hrcdv5sM2ek4Kjq40FgX-wijr-0JdpSDpZ1bIK00sPb4-
u1B8c7MaCqBcbJAhfmg4utLU67fn5GLoCX_-5TAWV0ID-_sC1Vs9glWRPkKmmktJMbV98XqC5-
DhtE3yd5I9ZM1SEH1gGYL1RjxwzPjWwHE-YH1Nx9lm-
Esa27TK7M86uT8iAe7Lgtvi02YsCB0buShHwmih3RzwMGaNaevmFSxPRK_sDmTEoVicaYnGa0kaMwhmmF
```

```
9AtPwGmFaGg1v3rryVg0X0bGoXRetnrPpDG7jUoq5zQuXQSelBf9hmNwEqWsS2tI4zNTxjiEkxU0djhPX  
qByZbnelp_3z6pqqniLzqj9jzAkvX6wDOW7ZycfDzOt-  
zNgTxWdtf41P6ZjVu8EWSf65Wqgen5jD4IPXgXGtxkjrSbrqiX-  
NxxxkfKVJUo0cE00F6n3DWD0BMWS8UG0Q08gZZeXcfpuTIGYTD6okyD91kLk5AmhaNTJVkjkHO-  
dHzqMHxikvhdK6C2PIfg4lEY0yuE3Fjj_5NNX5ZalIp013LN6YQ8Jqis_UmC_OXmjW2F5Y4p8VRRKc1HW  
2DFaUxBrEgfSwe_keyaofodrjde_pfPuDQDryEgGy9DNIhpGUv_bQJ8jlPxRL7WSpmPU7-  
IZ1mVN_onhqq2oI-WTl7ep-8w0GsJH30hSRyyJC0XC9xtetqVjIHzcKYFsxOaXT-  
LLe7U9oHaXHzjDK3hn-ZNFYwzV_aq8180eb" charset="UTF-8"></script><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(0,0,0,0.24);
}
form {border: 3px solid #f1f1f1; margin-left:400px; margin-right:400px; }

input[type=text], input[type=password] {
width: 100%;
padding: 12px 20px;
display: inline-block;
margin-bottom:18px;
border: 1px solid #ccc;
box-sizing: border-box;
}

button {
background-color: #28272c;
color: white;
padding: 14px 20px;
```



```
    margin-bottom:8px;
    border: none;
    cursor: pointer;
    width: 15%;
    border-radius:4px;
}

button:hover {
    opacity: 0.8;
}

.cancelbtn {
    width: auto;
    padding: 10px 18px;
    background-color: #f44336;
}

.imgcontainer {
    text-align: center;
    margin: 24px 0 12px 0;
}

img.avatar {
    width: 30%;
    border-radius: 50%;
}

.container {
    padding: 16px;
}

span.psw {
    float: right;
    padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
    span.psw {
        display: block;
        float: none;
    }
    .cancelbtn {
        width: 100%;
    }
}
```

```
}

.home{
    margin:80px;
    width: 84%;
    height: 500px;
    padding-top:10px;
    padding-left: 30px;
}

.login{
    margin:80px;
    box-sizing: content-box;
    width: 84%;
    height: 420px;
    padding: 30px;
    border: 10px solid blue;
}

.left,.right{
    box-sizing: content-box;
    height: 400px;
    margin:20px;
    border: 10px solid blue;
}

.mySlides {display: none;}
img {vertical-align: middle; }

/* Slideshow container */
.slideshow-container {
    max-width: 1000px;
    position: relative;
    margin: auto;
}

/* Caption text */
.text {
    color: #f2f2f2;
    font-size: 15px;
    padding: 8px 12px;
    position: absolute;
    bottom: 8px;
    width: 100%;
    text-align: center;
```

```
}

/* The dots/bullets/indicators */
.dot {
    height: 15px;
    width: 15px;
    margin: 0 2px;
    background-color: #bbb;
    border-radius: 50%;
    display: inline-block;
    transition: background-color 0.6s ease;
}

.active {
    background-color: #717171;
}

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

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

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

/* On smaller screens, decrease text size */
@media only screen and (max-width: 300px) {
    .text {font-size: 11px}
}

</style>
</head>

<body style="font-family:'Times New Roman', Times, serif;background-color:#C2C5A8;">

<div class="header">
```

```

<div style="width:50%;float:left;font-size:2vw;text-align:left;color:white;padding-top:1%">Plant Disease Prediction</div>
<div class="topnav-right" style="padding-top:0.5%;">
    <a class="active" href="{{ url_for('home') }}>Home</a>
    <a href="{{ url_for('prediction') }}>Predict</a>
</div>
</div>

<div style="background-color:#ffffff;">
<div style="width:60%;float:left;">
<div style="font-size:50px;font-family:Montserrat;padding-left:20px;text-align:center;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 world wide. Over the years it has developed and the use of new technologies and equipment replaced almost all the traditional methods of farming. The plant diseases effect the production. Identification of diseases and taking necessary precautions is all done through naked eye, which requires labour and laboratories. 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>

</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");

```

Feature 2:

```

        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>
```

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 type="text/javascript" src="https://gc.kis.v2.scr.kaspersky-labs.com/FD126C42-EBFA-4E12-B309-BB3FDD723AC1/main.js?attr=3wvf44XdejigWHFj22ANQmgfA-L5oa67wZhZwPtEITSo6t8o-DPZwNcHRFhpa2tgGpDJGis4-1IHYYxyIAN2GE0-kSZKkCLRkbKttCLVN9mKhGFVtGJ3auoiByn_jJ-mA447x4TmdjGgz8XvMdLSPF4Gu5xwt0joGxwdxuOEF18Sa5usZGgj4TdTfDHpElX3P1eH-lsevFhUJQEZe3981VXjRKYRn2FrxsYwXGSMBn0sRR9IYup35XYNQkvA6DLQV1lwLc4XuAo0B1JYAFI75R405LwTWuT-uaft0DEQeuV_F3rKvkrcBkalcpWnyXVLeLyjMz5CqpZ1aSCy1MgVAzWxGb-GX3eQb0F5q0ksANddV_vhz1Ai4RgptuAfB8mVyuZ0nWZzpmwam34lc4NL4tfyWGncKz2taMyGfsK4Mrn0zfPlY9_n9FP0lM1AX0IQ8TfbVp4B1vbwnA-

```

```
RVJq8mxoTjgMgqhKhp6NQY_8gZULkbqqA0pqUMvfL3_fZC1PFipLNjCyCGe9Y0aU9L7QF4CXeKsRhJXmI  
898fhpkB1oI7z0xvndsDLPRsqbnuse_eGL9tz0Te5HLGhtoXSns508pHC99_XHYofrlismcByzZ1mVqVkc  
NfmbnMjaD9IQf6xAACyjkQ927AOvyDVCZKr-  
tV6wRZyy_z7Z1J9AG7SGSLoB34AkMytkYXvpgGn21pGFNhl3YSmyKYc2XJs89zHbp5fSyXsfasogSEYL  
bpxCmuvezZK04haaqouKDcLwBGMFp_Br095f-  
AlhhW0dPDx1ezvTMx1NgS4Q0970mbyQCqHUFwZLYNgjQ8zpfdbXB17L_v_1fmruWhUiUVc9tRcJy-  
lpchFJe8Gz7TUOKCRDjbIWtiqXryDeENrJgQ31laXp-  
VVYp0I1L55pek2fgk50CGNzVges5oG4PpMyCIXtJpv32E5r1PTktG4hD8eXmYQECVU1HvSmEiKvuY6T6i  
9wdpqg_AnycRzUXmYdahFT3W7zToIn2RXzNfdOU0zbYBvtJ70TpR4PjfU751J0FsnpdDuCnero3UY0ak7  
vYvGYD9YY2md5v-3AmP-e0or2m55JRH_Hxpn28x-nDNOHqVBC6leYuYFBVV_vL51-  
E8n92uWUqwMEzDPZtAyRaCfz3D2Y0IYn-  
ZrnfNTg2M_zV3ePmUu1xdjYh7d1dx7nwclm7wJrBPb3JnX2kvEGYs9SM17MlwzoY1VJq4UzJ2D6oEvhQw  
HvG4e1et1S6iLWzhy8RVMfBlTa4D PDOHmT1HhsKbn0uaMyFFCppe79rtIVRctcommVmQysUwUOhjzlAq3  
0-hXJCTqdCWJe2xnxjAuUHVqHSiHiZllZaoOWNCV5Ypx_eqzn-KyZS3u-  
2_hGLHHNA2AVBWn_hF3Gz16dw6zA4QSmsWZSfDucNoB1JGOSTaDS3Z8jPTloYPFmu8oES6TL1dL1EK5Yhc  
SGaX4iv6o95drsZGb6bBcWgT7sNFHN6dve9wdjoDfUbergPIAm0sKaZQ2Ex6j150WCbE6UaPg-  
VNfziA2FEPpJaI9hEPI2gdaSuHqovlEot5mjufBBB0xpK0t8k0ZRtsVzqUuJw3VcLjaP6SfG_KZfgX_g8T  
Ps6CcFh1LRz63oXMQFW6AA7euDwfygndazedq5B-  
6DqSk0T04GTUJNqLcElg6KEEWqxd88BzoQoK28jrAf-xWHNIZv5HmQQYEnyX0U_cW8HX-  
hde54TuY_fY3e5QYU4be-JxTkA4JxWLagSa7-zs" charset="UTF-8"></script><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:#ffffff;
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>
        
    </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>
</body>

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

```
$("#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!');
        },
    });
});

});
```

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!');
                },
            });
        });
    });
});
```

8. Testing

8.1 Test Cases

Test cases are a set of actions performed on a system to determine if it satisfies software requirements and functions correctly as it claimed to perform

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute		Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
HomePage_TC_00	Functional	Home Page	Verify user is able to see the home page or not.		1.Enter URL and click go 2.Verify whether the user is able to see the home page.	Enter URL and click go	User able to see the home page	Working as expected	Pass	Nil	N	-	Rajapal S	
HomePage_TC_00	UI	Home Page	Verify the UI elements in Home Page		1.Enter URL and click go 2.Verify the UI elements in Home Page.	Enter URL and click go	Application should show below UI elements: Home Tab & Predict Tab	Working as expected	Pass	Nil	N	-	C.J.Dinesh Kumar	
PredictPage_TC_00	Functional	Predict page	Verify user is able to redirect to predict page or not.		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user is redirect to predict page or not.	Click the predict button in home page	User should navigate to Predict page	Working as expected	Pass	Nil	N	-	S.Aparna	
PredictPage_TC_00	UI	Predict page	Verify the UI elements in Predict Page		1.Enter URL and click go 2.Verify the UI elements in Predict Page.	Click the predict button and redirect to predict page	Application should show below UI elements: Dropdown List, Upload file Button, Predict button.	Working as expected	Pass	Nil	N	-	C.J.Dinesh Kumar	
PredictPage_TC_00	Functional	Predict page	Verify user is able to select the dropdown value or not.		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user is redirect to predict page or not 4.Verify user is able to select the dropdown value or not.	Fruit or Vegetable	Application should shows user to choose fruit or vegetable option in dropdown list.	Working as expected	Pass	Nil	N	-	P.Karthikeyan	
PredictPage_TC_00	Functional	Predict page	Verify user is able to upload the image or not.		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user is redirect to predict page or not 4.Verify user is able to select the dropdown value or not 5.Verify user is able to upload the image or not	Images to be uploaded	Application should shows the uploaded image.	Working as expected	Pass	Nil	N	-	A.Muthusamy	
PredictPage_TC_00	Functional	Predict page	Verify whether the image is predicted correctly or not		1.Enter URL and click go 2.Click on Predict button 3.Verify whether the user is redirect to predict page or not 4.Verify user is able to select the dropdown value or not 5.Verify user is able to upload the image or not 6.Verify whether the image is predicted correctly or not	Click the Predict Button	Application shows the predicted output	Working as expected	Pass	Nil	N	-	T.R.S.PraveenRajuSankaran	

8.2 User Acceptance Testing

Before deploying the software application to a production environment the end user or client performs a type of testing known as user acceptance testing, or UAT to ensure whether the software functionalities serve the purpose of development.

**Acceptance Testing
UAT Execution & Report Submission**

Date	03 November 2022
Team ID	PNT2022TMID52506
Project Name	Fertilizers recommendation system for disease prediction
Maximum Marks	4 Marks

1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
Yellow Leaves	10	4	5	15	34
Blight	1	5	2	4	12
Fruit rots	3	1	0	2	6
Leaf spots	9	2	4	18	33
Mosaic leaf pattern	3	9	6	6	24
Fruit Spots	3	1	5	1	10
Leaves misshapen	0	7	2	1	10
Totals	29	29	24	47	129

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
Yellow Leaves	20	0	0	20
Blight	43	0	0	43
Fruit rots	9	0	0	9
Leaf spots	5	0	0	5
Mosaic leaf pattern	19	0	0	19
Fruit Spots	2	0	0	2
Leaves misshapen	4	0	0	4

9. Results

Performance Metrics:

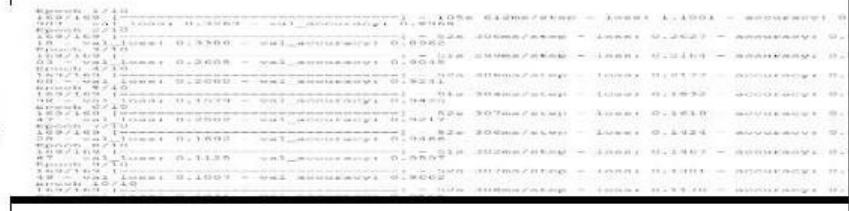
metrics are a baseline for performance tests.

Monitoring the correct parameters will help you detect areas that require increased attention and find ways to improve them.

Project Development Phase Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID52506
Project Name	Fertilizers recommendation system for disease prediction
Maximum Marks	10 Marks

Model Performance Testing:

S.No.	Parameter	Values	Screen Shot
1.	Model Summary	Total Params:896 Trainable Params:896 Non-Trainable Params:0	<pre>model.summary() Model: "sequential" Layer (type) Output Shape Param # conv2d (Conv2D) (None, 126, 126, 32) 896 max_pooling2d (MaxPooling2D) (None, 63, 63, 32) 0 flatten (Flatten) (None, 127008) 0 Total params: 896 Trainable params: 896 Non-trainable params: 0</pre>
2.	Accuracy	Training Accuracy = 90.3 Validation Accuracy = 89.62	
3.	Confidence Score (Only Yolo Projects)	Class Detected - NA Confidence Score - NA	

10. Advantages & Disadvantage

Advantages:

- Early detection of plant diseases.
- Proper fertilizer recommendation to prevent or cure the plant infection or disease.
- No need to consult any specialists.

- Fully automated system.

Disadvantages:

- Requires training the system with large dataset.
- Works only on the pretrained diseases.
- When a plant is infected with multiple diseases the system may not predict all the diseases due to the mixed symptoms.
- Requires a good device connected to the internet.

11. Conclusion:

Hence a system that takes in images as user input, analyses those for certain symptoms and identifies the disease, recommends the fertilizer to counter the deficiency of the nutrients is built and deployed.

12. Future Scope:

The system must be trained with numerous images of plant disease symptoms. In case of presence of multiple diseases, suitable classification must be done to predict each disease accurately and recommend separate fertilizers as a solution to each deficiency or infection.

13. Appendix:

Source Code:

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'> <script  
    type="text/javascript"  
    src="https://gc.kis.v2.scr.kaspe  
    rsky-labs.com/FD126C42-EBFA-  
    4E12-  
    B309-  
    BB3FDD723AC1/main.js?attr=A  
    MFGethIf4Q6r2IdpTrTqcDQGN  
    LDU5Cbc3diYnUdLkg5mQrVB_t  
    d  
    22OHUAsBJSd0oo8OR0zM3rlP  
    eFWfnEY4XCxQu4KOxMSqlshE
```

oIBOzvYw0SsMYpyUv4fnvKEjm
Joj_Y6cl4ov6AMOkz3Sh3epkfq
0gltnAPvvQBRdXqRmdqePVjlv
vqL28ONZCiS0Qr5t0XGxJ0bSiW
VTrH3cqaKCk05eP1Dx04mieTcj
sA_TtFLx15PUu0ed6soaj-
FOO6-
1d4OQxbJYBXUBefiUhzmOYCp
sGIs1OyQvA0huo8AUYwYB72d
vs07U3O2hq8BmYBv98h13sSo
8
iXKxyKx4FUUsOMkixjxYP6hu0w
wi7yv1E2rei3GHtPI5YwHkWio
QIPqvAmrlmaPtFZmFjE4_UUCi
9IEKws8IduDiqQIFkxfO3YT_sU
C9gWmxKSpGbiefwCgVwvdGE
nbUxY18p9Db6jC6FVKRhqdMB
ianq63qvzZRMZbEpjzQT0DQAH
3Yho4o4A00FIW2004q8Q80xt2
kV928P_nBgS9HOgHI5EZxenbjf
qANTs1r h8GGhBd7RJaE8-

**2AaqT6zbLf2tILJ8j4fk3bV1qsd
w0fPmp6foJbDu4343XH36a0V
GHsMLeVqcc30PSsE1pJbGE4_C
_E
xQd0_uRSA40mRjnFwHdLo9SJ
c1qghyc5YGQil_utG48oIMy9cC
6z-
iyKg1EeLKB43uq4SIUimRnuUsZ
W7drNWaijSfJPDmkm7IUJ0PO
wQXPfnLa2_spc3FisWCOZ7dFu
IgDciliu0yF8rio2X
OPz6pZkGQW4Fwl6vWKrLplmH
agJEIKXg58YSWwAT2DILilBjuSP
iTbCHR9Ya_mAXW4C03v7x
zJlaSK9jneECqctvKnH3RFgDS8o
cfDcY65IXNRkq6v1hrcdv5sM2e
k4Kjq4OFgXwijr0JdpSDpZlbIK0
0sPb4u1B8c7MaCqBcbJAhfmg4
utLU6 7fn5GLoCX_-5TAWV0ID-
_sC1Vs9glWRPkKmmktJMbVy9
8XqC5-**

DhtE3yd5I9ZM1SEH1gGYLIRjxw
zPjWwHE-

YH1Nx9lmEsq27TK7M86uT8iA
e7LgtviO2YsCB0buShHWmjh3R
zwMGqNqeymFSxPRK_sDmTFo
VjcaYpGa0

kaMwhmmF9AtPwGmFaGglv3r
ryVg0X0bGoXRetnrPpDG7jUoq
5zQuXQSedBf9hmNwEqWsSztl

4z

NTxjiEkxU0djhPXqByZbnelp_3z
6pqqnILzqj9jzAkvX6wDOW7Zy
cfDzOtzNgTxWdtf41P6ZjVu8E
WSf65Wqgen5jD4IPXgXGtxkjrS
brqiXNxxxkfKVJUOoOcEO0F6n3
DWD0BMWS8UGOQO8gZZeXC
fpuTIGYTD6okyD91kLk5AmhaN

TJV

KjkHOdHZqMHxikVhdK6C2Plfg

4lEY0yuE3Fjj_5NNX5ZallpOl3L
N6YQ8Jqis_UmC_OXmjW2F5Y4
p8VR

RKc1HW2DFaUxBrEgfSwe_key
aofodrjde_pfPuDQDryEgGy9D
NIhpGUV_bQJ8jlPxRL7WSpmP
U7

-IZ1mVN_onhqq2oI-
WTI7ep8w0GsJH3OhSRyyJC0X
C9xtetq

VjlHzcbKYFsxOaXTLLe7U9oHaX
HzjDK3hnZNFYwzV_aoq8180eb
" charset="UTF-
8"></script><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; marginright: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  
(maxwidth: 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  
(maxwidth: 300px) {  
    .text {font-size: 11px}  
}  
  
</style>  
</head>
```

```
<body style="fontfamily:'Times  
New Roman', Times,  
serif;backgroundcolor:#C2C5A8  
;">  
<div class="header">  
<div  
style="width:50%;float:left;fo  
nt-t-  
size:2vw;textalign:left;color:w  
hite; paddingtop:1%">Plant  
Disease Prediction</div> <div  
class="topnavright"style="pad  
dingtop:0.5%;">  
  
<a class="active" href="{{  
url_for('home')}}">Home</a>  
<a href="{{  
url_for('prediction')}}">Predict  
</div>  
</div>
```

```
<div  
style="backgroundcolor:#ffffff;  
> <div  
style="width:60%;float:left;">  
<div style="font-  
size:50px;fontfamily:Montserr  
at;paddingleft:20px;textalign:c  
enter;paddingtop:10%;"  
>  
<b>Detect if your plant<br> is  
infected!!</b></div><br> <div  
style="font-  
size:20px;fontfamily:Montserr  
at;paddingleft:70px;paddingrig  
ht:30px;textalign:justify;">A  
griculture is one of the major  
sectors worls wide. Over the  
years it has developed and the  
use of new technologies and  
equipment replaced almost all  
the traditional methods of
```

farming. The plant diseases effect the production.

Identification of diseases and taking necessary precautions is all done through naked eye, which requires labour and laboratories. This application helps farmers in detecting the diseases by observing the spots on the leaves, which in turn saves effort and labor costs.</div>

</div>

</div> <div

>

</div>
</div>
<div class="home">

</div> <script> var
slideIndex = 0;
showSlides(); function
showSlides() {
 var i; var
 slides =
document.getElementsByClassName
Name("mySlides");
 var dots =
document.getElementsByClassName
Name("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[slideIndex1].style.displa
y = "block"; dots[slideIndex-
1].className
+= " active";
setTimeout(showSlides, 2000);
// Change image every 2
seconds
}
</script>
</body>
</html>
```

**predict.html:**

```
<!DOCTYPE html>

<html >

<head>

<meta charset="UTF-8">

<meta name="viewport"
content="width=devicewid
th, initial-scale=1">

<title> Plant Disease
Prediction</title>

<link
href='https://fonts.googlea
pis.com/css?family=Pacific
o' rel='stylesheet'
type='text/css'>

<link
href='https://fonts.googlea
pis.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.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
 rel="stylesheet">
```

```
<script
 type="text/javascript"
 src="https://gc.kis.v2.scr.kasperskylabs.com/FD126C42-EBFA4E12-B309BB3FDD723AC1/main.js?at=tr=3wvf44XdejigWHFj22ANQmgfA-L5oa67wZhZwPtEITsot6t8o-"
```

**DPZwNcHRFhp<sub>a</sub>2tgGpDJGis  
4-1IHYxyIAN2GE0-  
kSZKkCLRkbKttCLVN9mKh  
GFVtGJ3auoiiByn\_jJmA447  
x4TmdjGgz8XvMdLS  
PF4Gu5xwt0joGxWDXuOEF  
18Sa5usZGgj4TdDiTfDHpEl  
X3P1eHlsevFhUJQEZe3981  
VXjRKYR  
n2FrxsYwXGSMBn0sRR9IY  
up35XYNQkvA6DLQV1lwLc  
4XuAo0B  
IJYAfl75R4O5LwTWuT-  
uaft0DEQeuV\_f3rKvkrcBkal  
cpWnyXVLeLyjMz5CqpZ1aS  
Cy1MgVAzWxGbGX3eQb0F  
5qOksANddV\_vh  
z1Ai4RgptuAfB8mVyuz0n  
WZzpmwam34lc4NL4tfyW  
GncKz2taMyGfs  
K4Mrn0zfPIY9\_n9FP0IMIAx**

**0IQ8TfbVp4B1vbwnARVJq8  
mxoTjgMgqhKhp6N  
QY\_8gZULkbqqA0pqUMvfL  
3\_fZC1PFipLNjCyCGe9YOa  
U9L7QF4CXe**

**KsRhJXml898FhpxB1oI7z0x  
vnndsDLPRsqbNuse\_eGL9tz  
0Te5HLGhtoXSn5O8pHC99  
\_XHYofrlismc**

**ByzZImVqVkCNfmbnMjaD9  
IQf6xAACyjkQ927AOvyDVC  
ZKrtV6wRZyv\_z7Z1J9AG7S  
GSL**

**oB34AkMytkYXvpgGn21pG  
FNhvl3YSmyKYc2XJs89zHb  
p5fSyXfas  
ogSEYLbpvxCmuvezZKO4haa  
qouKDcLwBGMFp\_Br095fA  
IhhWOdPDx1ezvTMx1Ng  
S4QO97OmbyQCqHUFWW**

ZLYNgjQ8zpfdBXB17L\_v\_If  
mrUWhUiUV

c9tRcJylpchFJe8Gz7TUOKC  
RDjbIW

tiqXryDeENrJgQ31laXpVVY  
pOI1L55pek2fgk5OCGN  
zVges5oG4PpMyCIxtJpv32  
E5rlPTktG4hD8eXmYQECV  
U1HvSmEiK

vuY6T6i9wdpqg\_AnycRzUX  
mYdahFT3W7zToIn2RXzNf  
dOU0zbYBvtJ70TpR4PjfU75  
IJ0FsnpbDu

Cnero3UYOak7vYvGYD9YV  
2md5v-

3AmPeOor2m55JZRH\_Hxp  
n28xnDNCOHqVBC6leYuYF  
BVV\_  
vL5I-

E8n92uWUqwMEzdZPZtAy

**RaCfz3D2Y0IYn-**  
**ZrnfNTg2M\_zVJePmUu1xdj**  
**Yh7d1dx7nwclm7wJrBPb3J**  
**nX2kvEGYs9SM17MlwzoY1**  
**VJq4UzJ2D6o**

**EvhQwHvG4e1etlS6iLWzhy**  
**8RVMfBITa4DPDOHmTIHhs**  
**Kbn0UaMyFFCppe79rtlVRc**  
**tcommVmQy**  
**sUwUOhjzlAq30hXJCTqdC**  
**WJe2xnxjAuUHV**  
**qHSiHiZlIZaoOWNCV5Ypx\_**  
**eqzn-KyZS3u-**  
**2\_hGLHHNA2AVBWn\_hF3**  
**Gz16dw6zA4QSmWZSfDUc**  
**NObLJGOSTaDS3Z8jPTloYP**  
**Fmu8oES6T**

**L1dLIEK5YhcSGaX4iv6o95d**  
**rsZGb6bBcWgT7sNFHW6d**  
**VE9wdjoDFuBergPIAm0sKa**

**ZQ2Ex6j15O**

**WCbE6UaPg-**

**VNfziA2FEPpJaI9hEPI2gdaS**

**uHqovlEOt5mjuFBBOxpK0t**

**8kOZRtsVzqUuJw3VcLjaP6S**

**fG\_KZfgX\_**

**g8TPs6CcFhILRz63oXMQFP**

**W6AA7eudWfygndazedq5**

**B-**

**6DqSkOT04GTUJNqLcElg6K**

**EEWqxd88BzoQoK28jrAfx**

**WHNIZv5HmQQYEnyX0U\_**

**cW8HXhde54TuY\_fY3e5QY**

**u4beJxTkA4JxWLEagSa7-**

**zs" charset="UTF-**

**8"></script><script**

**src="https://cdn.bootcss.c**

**om/popper.js/1.12.9/umd/**

**popper.min.js"></script>**

```
<script
src="https://cdn.bootcss.c
om/jquery/3.3.1/jquery.mi
n.js"></script>
```

```
<script
src="https://cdn.bootcss.c
om/bootstrap/4.0.0/js/bo
otstrap.min.js"></script>
```

```
<link
href='https://fonts.googlea
pis.com/css?family=Open+
Sans+Condensed:300'
rel='stylesheet'
type='text/css'>
```

```
<link
href='https://fonts.googlea
pis.com/css?family=Merri
weather' rel='stylesheet'>
```

```
<link
href='https://fonts.googlea
```

**pis.com/css?family=Josefin  
Sans' rel='stylesheet'>**

**<link  
href='https://fonts.googlea  
pis.com/css?family=Monts  
errat' 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:#ffffff; background-
repeat: norepeat;
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:
boxshadow .5s ease;

-moz-transition:
boxshadow .5s ease;

-o-transition: box-shadow
.5s ease;

-ms-transition: boxshadow
.5s ease; transition: box-
shadow .5s ease;

}

</style>

</head>
```

```
<body
style="fontfamily:Montserr
at;overflow w:scroll;">

<div class="header">

<div
style="width:50%;float:left
;font-
size:2vw;textalign:left;colo
r:white; padding-
top:1%">Plant
Disease Prediction</div>
<div class="topnav-right"
style="paddingtop: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" >

 </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/formdata">

<select name="plant">

 <option value="select" selected>Select plant type</option>
 <option value="fruit">Fruit</option>
 <option value="vegetable">Vegetable</option>

</select>

<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="btnpredict"
style="background:
#28272c;">Predict!</butto
n>
```

```
</div> </div>

<div
 class="loader"
 ">

 style="displa
 y:none;"></d
 i v>

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

 </h3>

 </div> </div>

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

 </div>
```

```
</body>

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

</footer> </html>
```

## 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('ba
ckground-image', 'url(' +
 e.target.result + ')');

 $('#imagePreview').hide();

 $('#imagePreview').fadeIn(
 650);
 }

 reader.readAsDataURL(inp
 ut.files[0]);
```

```
}
```

```
}
```

```
$("#imageUpload").change
```

```
(function () {
```

```
$('.image-section').show();
```

```
$('#btn-predict').show();
```

```
$('#result').text('');
```

```
$('#result').hide();
```

```
readURL(this);
```

```
});
```

```
// Predict
```

```
$('#btnclick').click(function
```

```
on () { 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!');

},

});

});});
```

## 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:
 norepeat; 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);
}
```

## Python – app.py:

```
import os

import numpy as np
import pandas as pd
from
tensorflow.keras.mod
```

```
e ls import
load_model

from
tensorflow.keras.prepr
ocessing import image
from werkzeug.utils
import
secure_filename from
flask import
Flask,
render_template,
request app =
Flask(__name__)

#load both the
vegetable and fruit
models model =
load_model("vegetabl
```

```
e.h5")
model1=load_model("f
ruit.h5")

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

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

```
@app.route('/predict',
methods=['POST']) def
predict():

 if request.method ==
 'POST':

 # Get the file from
 post request

 f =
 request.files['image']

 # Save the file to
 ./uploads basepath =
 os.path.dirname(__file
 __) file_path =
 os.path.join(
 basepath, 'uploads',
```

```
secure_filename(f.filename))
```

```
f.save(file_path)
```

```
img =
```

```
image.load_img(file_path, target_size=(128, 128))
```

```
x =
```

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

```
plant=request.form['plant']
```

```
print(plant)
if(plant=="vegetable")
```

```
:
```

```
preds =
model.predict(x)
```

```
preds=np.argmax(pred
s)
```

```
print(preds)
```

```
df=pd.read_excel('prec
autions - veg.xlsx')
```

```
print(df.iloc[preds]['ca
ution']) else:
```

```
preds =
model1.predict(x)
```

```
preds=np.argmax(preds)
```

```
df=pd.read_excel('precations - fruits.xlsx')
```

```
print(df.iloc[preds]['caution'])
```

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

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

**DEPLOYMENT MODEL CODE:**

**Fruit model:**

ls

sample\_data/ pwd

'/home/wsuser/work'

!pip install keras==2.7.0

!pip install tensorflow==2.5.0

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>

Requirement already satisfied: keras==2.7.0 in  
/usr/local/lib/python3.7/dist-packages (2.7.0)

Looking in indexes: <https://pypi.org/simple>, <https://us-python.pkg.dev/colab-wheels/public/simple/>

Requirement already satisfied: tensorflow==2.5.0 in  
/usr/local/lib/python3.7/dist-packages (2.5.0)

Requirement already satisfied: h5py~=3.1.0 in  
/usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (3.1.0)

Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-packages  
(from tensorflow==2.5.0) (3.19.6)

Requirement already satisfied: typing-extensions~=3.7.4 in /usr/local/lib/python3.7/dist-packages  
(from tensorflow==2.5.0) (3.7.4.3)

Requirement already satisfied: keras-nightly~=2.5.0.dev in /usr/local/lib/python3.7/dist-packages  
(from tensorflow==2.5.0) (2.5.0.dev2021032900)

Requirement already satisfied: flatbuffers~=1.12.0 in /usr/local/lib/python3.7/dist-packages  
(from tensorflow==2.5.0) (1.12)

Requirement already satisfied: gast==0.4.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (0.4.0)

Requirement already satisfied: absl-py~=0.10 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (0.15.0)

Requirement already satisfied: astunparse~=1.6.3 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (1.6.3)

Requirement already satisfied: tensorflow-estimator<2.6.0,>=2.5.0rc0 in  
/usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (2.5.0)  
Requirement already satisfied: tensorboard~=2.5 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (2.9.1)

Requirement already satisfied: opt-einsum~=3.3.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (3.3.0)

Requirement already satisfied: six~=1.15.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (1.15.0)

Requirement already satisfied: google-pasta~=0.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (0.2.0)

Requirement already satisfied: grpcio~=1.34.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (1.34.1)

Requirement already satisfied: wrapt~=1.12.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (1.12.1)

Requirement already satisfied: termcolor~=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (1.1.0)

Requirement already satisfied: keras-preprocessing~=1.1.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (1.1.2)

Requirement already satisfied: wheel~=0.35 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (0.38.3)

Requirement already satisfied: numpy~=1.19.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow==2.5.0) (1.19.5)

Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packages (from h5py~=3.1.0->tensorflow==2.5.0) (1.5.2)

Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/dist-packages

(from tensorboard~=2.5->tensorflow==2.5.0) (2.14.1)

Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in  
/usr/local/lib/python3.7/dist-packages (from tensorboard~=2.5->tensorflow==2.5.0) (0.6.1)

Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in  
/usr/local/lib/python3.7/dist-packages (from tensorboard~=2.5->tensorflow==2.5.0) (1.8.1)

Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in  
/usr/local/lib/python3.7/dist-packages (from tensorboard~=2.5->tensorflow==2.5.0) (0.4.6)

Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.7/dist-packages  
(from tensorboard~=2.5->tensorflow==2.5.0) (1.0.1)

Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages  
(from tensorboard~=2.5->tensorflow==2.5.0) (3.4.1)

Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist packages  
(from tensorboard~=2.5->tensorflow==2.5.0) (2.23.0)

Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.7/dist-packages  
(from tensorboard~=2.5->tensorflow==2.5.0) (57.4.0)

Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-packages (from  
google-auth<3,>=1.6.3->tensorboard~=2.5->tensorflow==2.5.0) (4.9) Requirement already  
satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist packages (from  
googleauth<3,>=1.6.3->tensorboard~=2.5->tensorflow==2.5.0) (0.2.8) Requirement already  
satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.7/dist packages (from  
googleauth<3,>=1.6.3->tensorboard~=2.5->tensorflow==2.5.0) (5.2.0) Requirement already  
satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist packages  
(from googleauth-oauthlib<0.5,>=0.4.1->tensorboard~=2.5->tensorflow==2.5.0) (1.3.1)

Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist  
packages (from markdown>=2.6.8->tensorboard~=2.5->tensorflow==2.5.0) (4.13.0)

Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from  
importlib-metadata>=4.4>markdown>=2.6.8->tensorboard~=2.5->tensorflow==2.5.0) (3.10.0)

Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist packages  
(from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard~=2.5-  
>tensorflow==2.5.0) (0.4.8)

```
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.5-
>tensorflow==2.5.0) (1.24.3)

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from
requests<3,>=2.21.0->tensorboard~=2.5->tensorflow==2.5.0) (2.10)

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages
(from requests<3,>=2.21.0->tensorboard~=2.5->tensorflow==2.5.0) (3.0.4) Requirement
already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from
requests<3,>=2.21.0->tensorboard~=2.5->tensorflow==2.5.0) (2022.9.24) Requirement already
satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.5- >tensorflow==2.5.0)
(3.2.2)
```

## Image Augmentation

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,zoom_range=0.2,horizontal_flip=True,ver-
tical_flip=False) import os, types import pandas as pd
from botocore.client import Config import ibm_boto3
def __iter__(self): return 0 from
ibm_watson_machine_learning import APIClient
wml_credentials = {
"url": "https://eu-gb.ml.cloud.ibm.com",
"apikey": "IxG8u1Z4LCtAiOn_gDLZiKeWP4i5xM7uxjf1jN_dTOIn"
}
```

```
client = APIClient(wml_credentials) client model=Sequential()

model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))

model.add(MaxPooling2D(pool_size=(2,2))) model.add(Flatten())

model.summary()

Model: "sequential"
```

---

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d (MaxPooling2D)	(None, 63, 63, 32)	0

---

flatten (Flatten)	(None, 127008)	0
-------------------	----------------	---

---

Total params: 896

Trainable params: 896

Non-trainable params: 0

---

## Hidden Layers

```
model.add(Dense(300,activation='relu')) model.add(Dense(150,activation='relu'))
```

## Output Layers

```
model.add(Dense(6,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
len(x_train) 169
```

1238/24

51.58333333333336

```
C:\Users\LonelyDinesh\anaconda3\lib\sitepackages\tensorflow\python\keras\engine\training.py:1940: UserWarning:
'Model.fit_generator` is deprecated and will be removed in a future version.
Please use `Model.fit`, which supports generators.
```

warnings.warn(`Model.fit\_generator` is deprecated and '

Epoch 1/10

```
169/169 [=====] - 138s 812ms/step - loss: 1.5129 -
accuracy: 0.7643 - val_loss: 0.2199 - val_accuracy: 0.9199
```

Epoch 2/10

```
169/169 [=====] - 74s 438ms/step - loss: 0.2590 -
accuracy: 0.9107 - val_loss: 0.2204 - val_accuracy: 0.9265
```

Epoch 3/10

```
169/169 [=====] - 74s 438ms/step - loss: 0.2134 -
accuracy: 0.9277 - val_loss: 0.1829 - val_accuracy: 0.9442
```

Epoch 4/10

```
169/169 [=====] - 76s 450ms/step - loss: 0.1779 -
accuracy: 0.9419 - val_loss: 0.1587 - val_accuracy: 0.9514
```

Epoch 5/10

```
169/169 [=====] - 76s 452ms/step - loss: 0.1547 -
accuracy: 0.9458 - val_loss: 0.1422 - val_accuracy: 0.9514
```

Epoch 6/10

```
169/169 [=====] - 74s 435ms/step - loss: 0.1072 -
accuracy: 0.9660 - val_loss: 0.2070 - val_accuracy: 0.9288
```

Epoch 7/10

```
169/169 [=====] - 77s 456ms/step - loss: 0.1018 -
accuracy: 0.9638 - val_loss: 0.1305 - val_accuracy: 0.9531
```

Epoch 8/10

```
169/169 [=====] - 80s 473ms/step - loss: 0.1088 -
accuracy: 0.9666 - val_loss: 0.1520 - val_accuracy: 0.9502
```

Epoch 9/10

```
169/169 [=====] - 80s 474ms/step - loss: 0.0835 -
accuracy: 0.9697 - val_loss: 0.1126 - val_accuracy: 0.9668
```

Epoch 10/10

```
169/169 [=====] - 77s 458ms/step - loss: 0.0691 -
accuracy: 0.9764 - val_loss: 0.1282 - val_accuracy: 0.9620
```

```
<tensorflow.python.keras.callbacks.History at 0x2c9806e6df0>
```

**Saving Model** model.save('fruit.h5')

```
!tar -zcvf Train-model_new.tgz fruit.h5
```

```
a fruit.h5 ls -1
```

**IBM Cloud Deployment Model**

```
!pip install watson-machine-learning-client --upgrade
```

Requirement already satisfied: watson-machine-learning-client in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (1.0.391)  
Requirement already satisfied: requests in c:\users\lonelydinesh\anaconda3\lib\site-packages  
(from watson-machinelearning-client) (2.28.1)

Requirement already satisfied: lomond in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from watson-  
machinelearning-client) (0.3.3)

Requirement already satisfied: tabulate in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from watson-  
machinelearning-client) (0.8.10)

Requirement already satisfied: urllib3 in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from watson-  
machinelearning-client) (1.26.11)

Requirement already satisfied: boto3 in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from watson-  
machinelearning-client) (1.24.28)

Requirement already satisfied: ibm-cos-sdk in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from watson-  
machinelearning-client) (2.11.0)

Requirement already satisfied: tqdm in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from watson-  
machinelearning-client) (4.64.1)

Requirement already satisfied: certifi in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from watson-  
machinelearning-client) (2022.9.14)

Requirement already satisfied: pandas in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from watson-  
machinelearning-client) (1.4.4)

Requirement already satisfied: s3transfer<0.7.0,>=0.6.0 in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from boto3->watsonmachine-learning-client) (0.6.0)

Requirement already satisfied: botocore<1.28.0,>=1.27.28 in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from boto3->watsonmachine-learning-client) (1.27.28)

Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from boto3->watsonmachine-learning-client) (0.10.0)

Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from ibm-cos-sdk->watsonmachine-learning-client) (2.11.0)

Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from ibm-cos-sdk->watsonmachine-learning-client) (2.11.0)

Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from ibm-cos-sdkcore==2.11.0->ibm-cos-sdk->watson-machine-learning-client) (2.8.2)

Requirement already satisfied: charset-normalizer<3,>=2 in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from requests->watsonmachine-learning-client) (2.0.4)

Requirement already satisfied: idna<4,>=2.5 in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from requests->watsonmachine-learning-client) (3.3)

Requirement already satisfied: six>=1.10.0 in  
c:\users\lonelydinesh\anaconda3\lib\site-packages (from lomond->watsonmachine-learning-client) (1.15.0)

```
Requirement already satisfied: numpy>=1.18.5 in
c:\users\lonelydinesh\anaconda3\lib\site-packages (from pandas->watsonmachine-learning-client) (1.19.5)
```

```
Requirement already satisfied: pytz>=2020.1 in
c:\users\lonelydinesh\anaconda3\lib\site-packages (from pandas->watsonmachine-learning-client) (2022.1)
```

```
Requirement already satisfied: colorama in
c:\users\lonelydinesh\anaconda3\lib\site-packages (from tqdm->watsonmachine-learning-client) (0.4.5) from
ibm_watson_machine_learning import APIClient
```

```
wml_credentials = {

 "url": "https://eu-gb.ml.cloud.ibm.com",

 "apikey": "IxG8u1Z4LCtAiOn_gDLZiKeWP4i5xM7uxjf1jN_dTOIn"

}

client = APIClient(wml_credentials) client

<ibm_watson_machine_learning.client.APIClient at 0x2c9f5130a00>

client = APIClient(wml_credentials) def

guid_from_space_name(client, space_name): space =

 client.spaces.get_details()

 return(next(item for item in space['resources'] if
item['entity'][“name”]==space_name)[‘metadata’][‘id’]) space_uid =
guid_from_space_name(client, ‘Trainmodel’) print(“Space UID = ” + space_uid)

```