PROJECT BASED EXPERIENTIAL LEARNING PROGRAM (NALAIYA THIRAN)

Fertilizers Recommendation System For Disease Prediction

A PROJECT REPORT

Submitted by

1.Nisagar-813819106071

2.Poojapriyadharshini-813819106075

3.Praveena-813819106077

4.Suriyasri-813819106112

TEAM ID:PNT2022TMID32773

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Project Report Format

1. INTRODUCTION

1.1PROJECT OVERVIEW

Agriculture is the most important sector in today’s life. Most plants are affected by a wide variety of bacterial and fungal diseases. Diseases on plants placed a major constraint on the production and a major threat to food security. Hence, early and accurate identification of plant diseases is essential to ensure high quantity and best quality. In recent years, the number of diseases on plants and the degree of harm caused has increased due to the variation in pathogen varieties, changes in cultivation methods, and inadequate plant protection techniques.

An automated system is introduced to identify different diseases on plants by checking the symptoms shown on the leaves of the plant. Deep learning techniques are used to identify the diseases and suggest the precautions that can be taken for those diseases.

So the ultimate aim of our project is to help the farmers by identifying the disease before the crop get damaged and to give good quality crops to all the people in the cheap budget so we have choosen this domain and by using deep learning technique we are developing a software project called fertilizer recommedation and disease predication.

1.2 Purpose

We propose a system which helps farmers detect plant disease. We use the EfficientNet [8] deep learning model, which achieves 99.8% validation accuracy on our choice of dataset for plant disease detection **to predict the occurrence or change in severity of plant diseases**. At the field scale, these systems are used by growers to make economic decisions about_disease treatments for control.

2 LITERATURE SURVEY

2.1 EXSISTING PROBLEMS

There are needs to simulation results show that KNN based classifacation method provides an accuracy of **85%** for the blast affected leaf images and **86%** for the normal leaf images.

Disease can be predicted accurately only using complex application. Quality production is increades by using machine learning techniques so that the food security is increased.

Disease can be predicted for onlylimited crops and it is not applicable for all crops.There are unwanted background image so can't able to detect the plant correctly.

2.2 REFRENCES

S. RAMESH ,D. VYDEKI - 1970

PRADEEP SEELWAL,A. SHARMA - 2022

S.MOHANTY,DAVID P., HUQHES, M.SALATHE - 2016

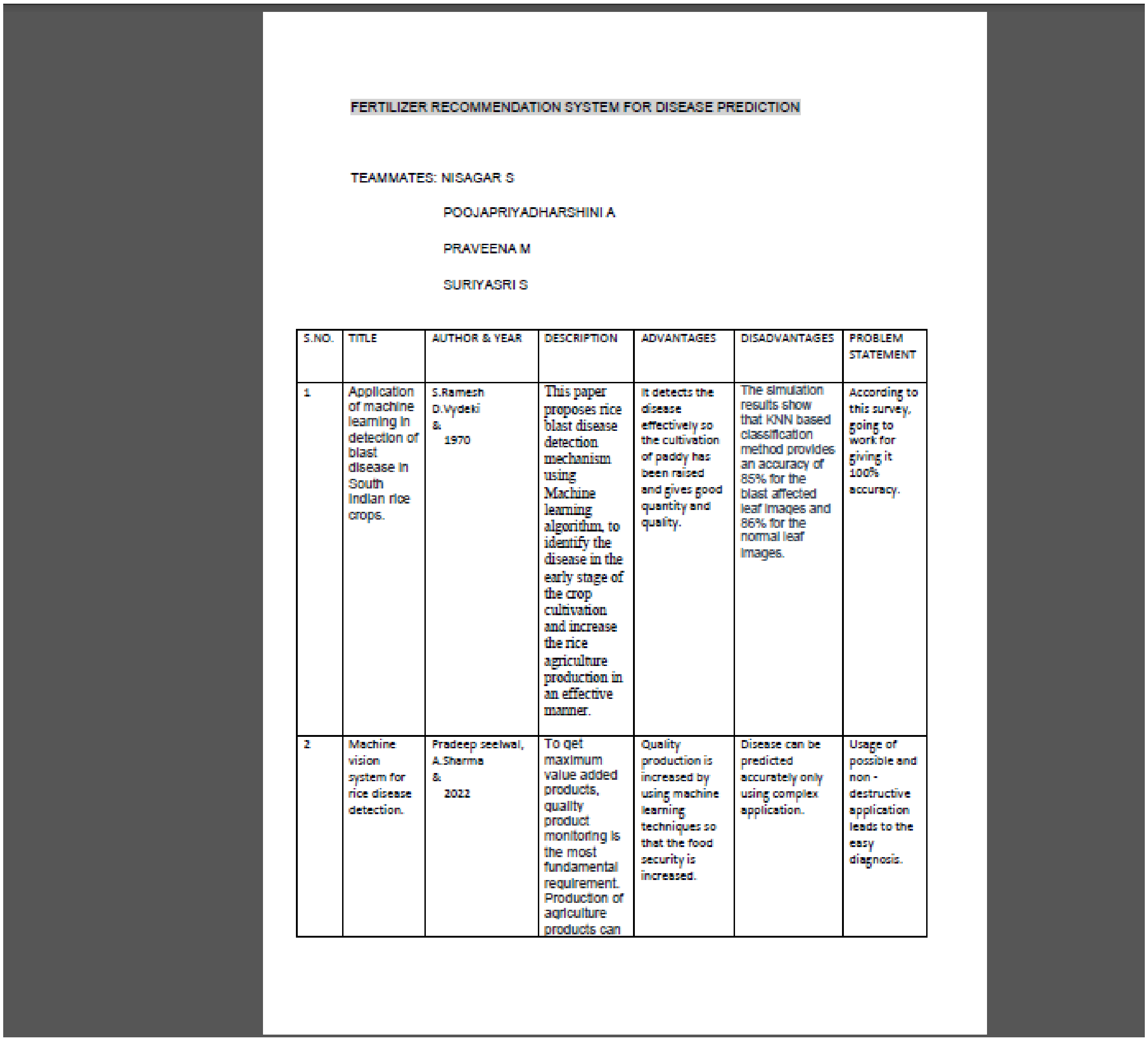
H.AMIN,A.DARIUM,M.SOLLIMAN

2.3 PROBLEM STATMENT DEFINITION

According to this survey, going to work for giving it **100%** accuracy.Usage of possible and non-deatructive application leads to the easy diagnosis.

Trying to detect the disease for more crops.Under reasearch to avoid the unwanted input background image.

Process is going on to detect the disease for more crops.Working for it giving 100% accuracy.



FERTILIZER RECOMMENDATION SYSTEM FOR DISEASE PREDICTION

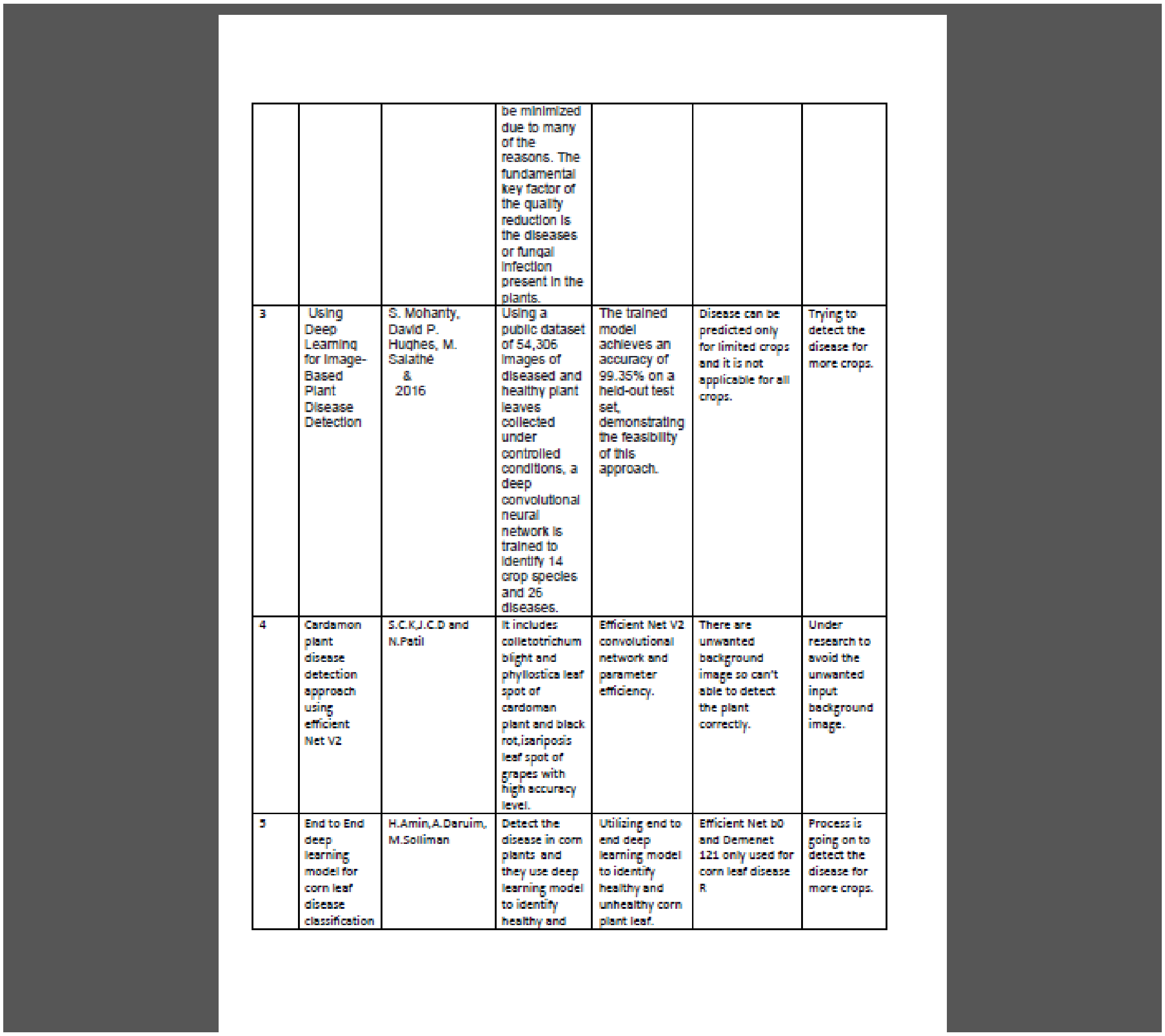
TEAMMATES: NISAGAR S

POOJAPRIYADHARSHINI A

PRAVEENA M

SURIYASRI S

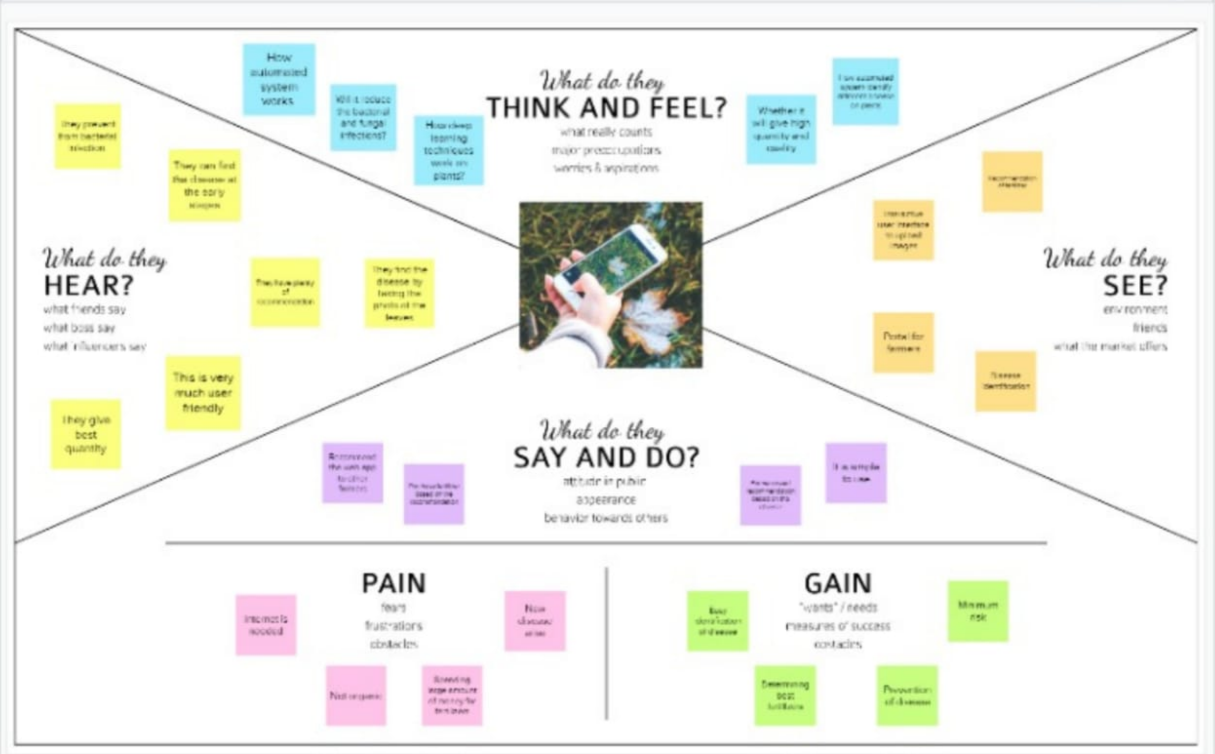
S.NO.	TITLE	AUTHOR & YEAR	DESCRIPTION	ADVANTAGES	DISADVANTAGES	PROBLEM STATEMENT
1	Application of machine learning in detection of blast disease in South Indian rice crops.	S.Ramesh D.Vydeki & 1970	This paper proposes rice blast disease detection mechanism using Machine learning algorithm, to identify the disease in the early stage of the crop cultivation and increase the rice agriculture production in an effective manner.	It detects the disease effectively so the cultivation of paddy has been raised and gives good quantity and quality.	The simulation results show that KNN based classification method provides an accuracy of 85% for the blast affected leaf images and 86% for the normal leaf images.	According to this survey, going to work for giving it 100% accuracy.
2	Machine vision system for rice disease detection.	Pradeep seelwal, A.Sharma & 2022	To get maximum value added products, quality product monitoring is the most fundamental requirement. Production of agriculture products can	Quality production is increased by using machine learning techniques so that the food security is increased.	Disease can be predicted accurately only using complex application.	Usage of possible and non - destructive application leads to the easy diagnosis.



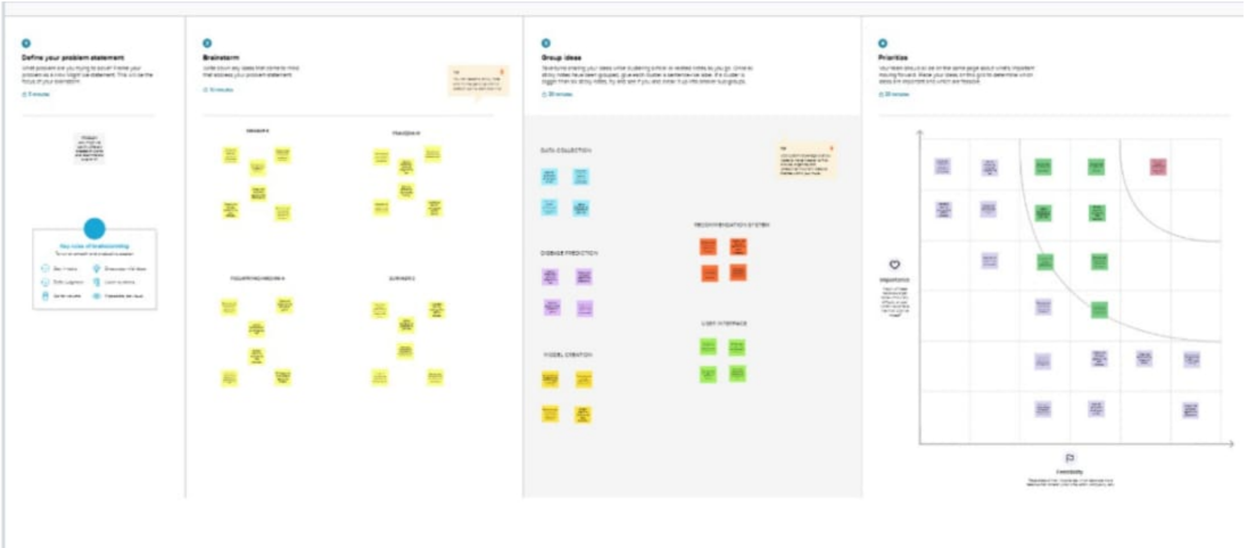
			be minimized due to many of the reasons. The fundamental key factor of the quality reduction is the diseases or fungal infection present in the plants.			
3	Using Deep Learning for Image-Based Plant Disease Detection	S. Mohanty, David P. Hughes, M. Salathé & 2016	Using a public dataset of 54,306 images of diseased and healthy plant leaves collected under controlled conditions, a deep convolutional neural network is trained to identify 14 crop species and 26 diseases.	The trained model achieves an accuracy of 99.35% on a held-out test set, demonstrating the feasibility of this approach.	Disease can be predicted only for limited crops and it is not applicable for all crops.	Trying to detect the disease for more crops.
4	Cardamon plant disease detection approach using efficient Net V2	S.C.K,J.C.D and N.Patil	It includes colletotrichum blight and phyllostica leaf spot of cardoman plant and black rot,isariposis leaf spot of grapes with high accuracy level.	Efficient Net V2 convolutional network and parameter efficiency.	There are unwanted background image so can't able to detect the plant correctly.	Under research to avoid the unwanted input background image.
5	End to End deep learning model for corn leaf disease classification	H.Amin,A.Daruim, M.Solliman	Detect the disease in corn plants and they use deep learning model to identify healthy and	Utilizing end to end deep learning model to identify healthy and unhealthy corn plant leaf.	Efficient Net b0 and Demenet 121 only used for corn leaf disease R	Process is going on to detect the disease for more crops.

3. IDEATION & PROPOSED SOLUTION

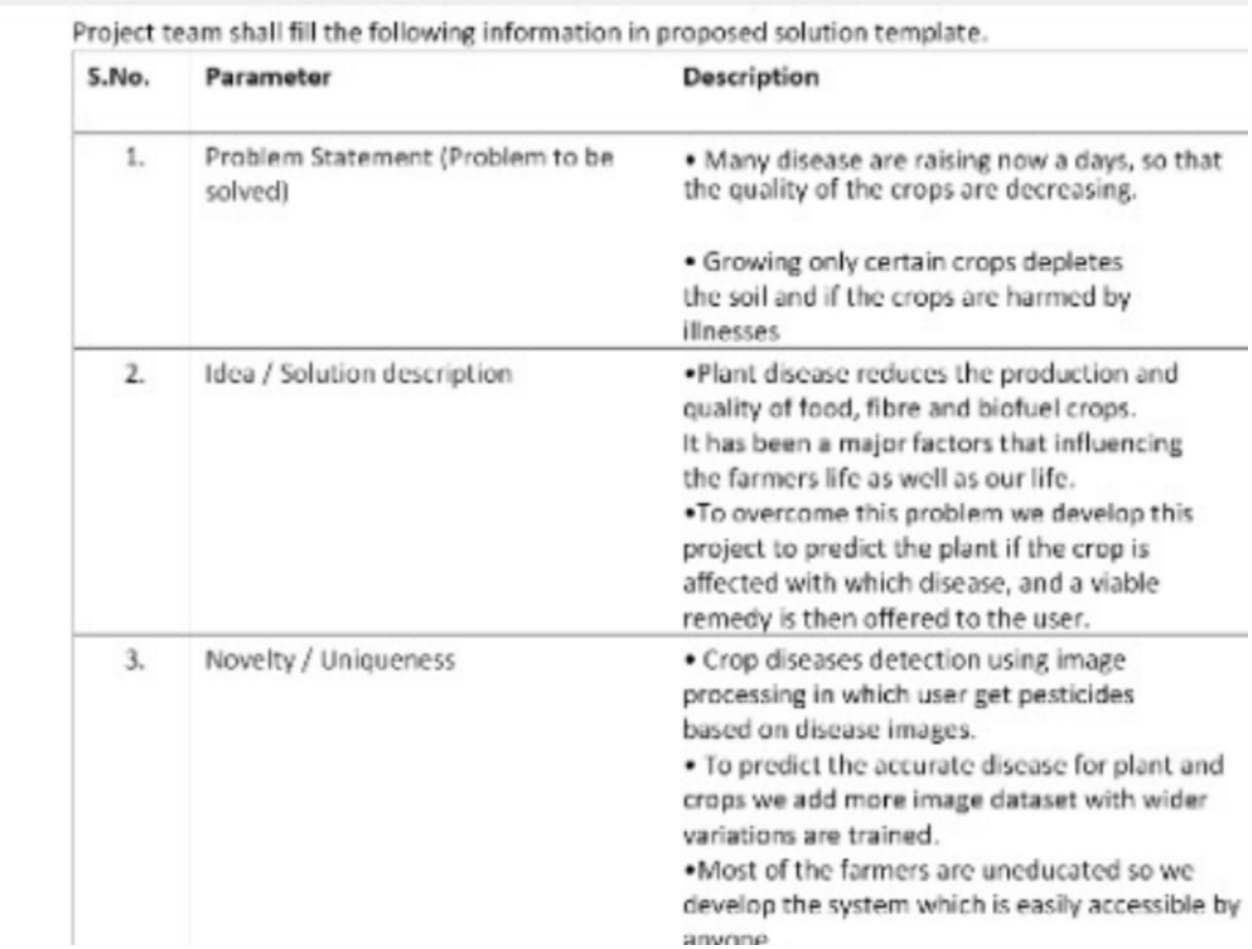
3.1 Empathy map canvas


What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
How automated system works
Will it reduce the bacterial and fungal infections?
How deep learning techniques work on plants?
Whether it will give high quantity and quality
What do they HEAR?
what friends say
what boss say
what influencers say
They prevent from bacterial infection
They can find the disease at the early stages
They find the disease by taking the photo of the leaves
This is very much user friendly
They give best quantity
What do they SEE?
environment
friends
what the market offers
Interactive user interface to upload images
Portal for farmers
Disease identification
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Recommend the web app for other farmers
It is simple to use
PAIN
fears
frustrations
obstacles
Internet is needed
New disease arise
Not organic
Spending large amount of money for fertilizer
GAIN
"wants" / needs
measures of success
obstacles
Easy identification of disease
Minimum risk
Determining best fertilizers
Prevention of disease

3.2 Brainstorming And Ideation

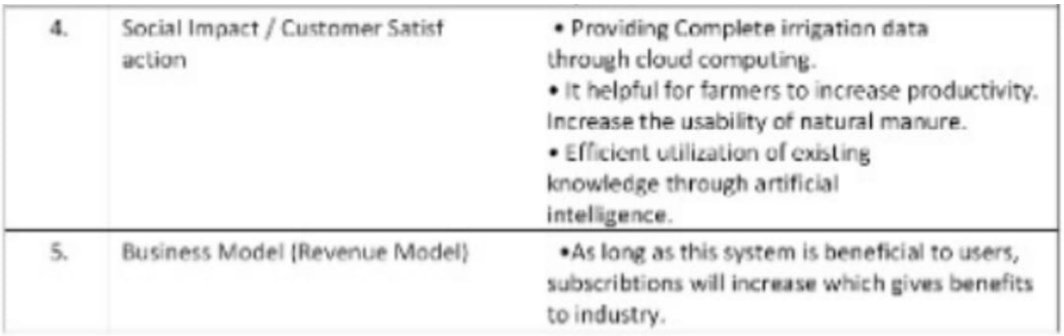


3.3 Proposed solution

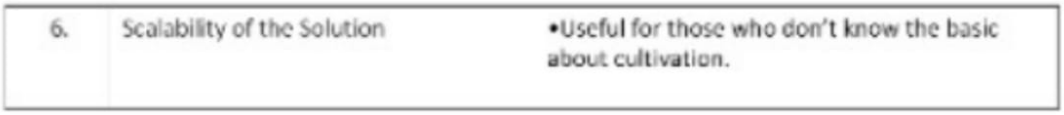


Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Many disease are raising now a days, so that the quality of the crops are decreasing. • Growing only certain crops depletes the soil and if the crops are harmed by illnesses
2.	Idea / Solution description	•Plant disease reduces the production and quality of food, fibre and biofuel crops. It has been a major factors that influencing the farmers life as well as our life. •To overcome this problem we develop this project to predict the plant if the crop is affected with which disease, and a viable remedy is then offered to the user.
3.	Novelty / Uniqueness	• Crop diseases detection using image processing in which user get pesticides based on disease images. • To predict the accurate disease for plant and crops we add more image dataset with wider variations are trained. •Most of the farmers are uneducated so we develop the system which is easily accessible by anyone

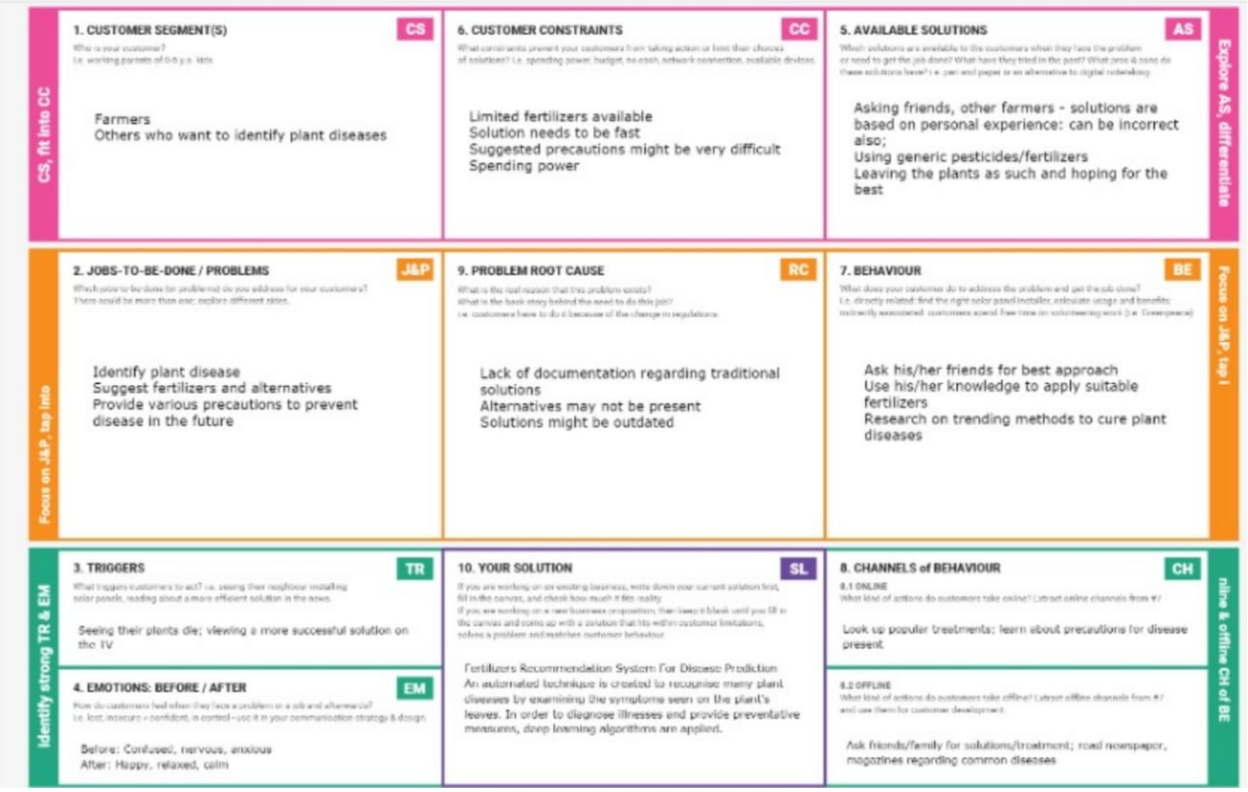


4.	Social Impact / Customer Satisf action	• Providing Complete irrigation data through cloud computing. • It helpful for farmers to increase productivity. Increase the usability of natural manure. • Efficient utilization of existing knowledge through artificial intelligence.
5.	Business Model (Revenue Model)	•As long as this system is beneficial to users, subscribtions will increase which gives benefits to industry.



6.	Scalability of the Solution	•Useful for those who don't know the basic about cultivation.

3.4 Problem solution fit



1. CUSTOMER SEGMENT(S)

Who is your customer?
i.e. working parents of 0-5 y.o. kids

Farmers
Others who want to identify plant diseases

6. CUSTOMER CONSTRAINTS

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Limited fertilizers available
Solution needs to be fast
Suggested precautions might be very difficult
Spending power

5. AVAILABLE SOLUTIONS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Asking friends, other farmers - solutions are based on personal experience: can be incorrect also;
Using generic pesticides/fertilizers
Leaving the plants as such and hoping for the best

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

Identify plant disease
Suggest fertilizers and alternatives
Provide various precautions to prevent disease in the future

9. PROBLEM ROOT CAUSE

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

Lack of documentation regarding traditional solutions
Alternatives may not be present
Solutions might be outdated

7. BEHAVIOUR

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Ask his/her friends for best approach
Use his/her knowledge to apply suitable fertilizers
Research on trending methods to cure plant diseases

3. TRIGGERS

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Seeing their plants die; viewing a more successful solution on the TV

4. EMOTIONS: BEFORE / AFTER

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before: Confused, nervous, anxious
After: Happy, relaxed, calm

10. YOUR SOLUTION

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Fertilizers Recommendation System For Disease Prediction
An automated technique is created to recognise many plant diseases by examining the symptoms seen on the plant's leaves. In order to diagnose illnesses and provide preventative measures, deep learning algorithms are applied.

8. CHANNELS of BEHAVIOUR

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

Look up popular treatments; learn about precautions for disease present

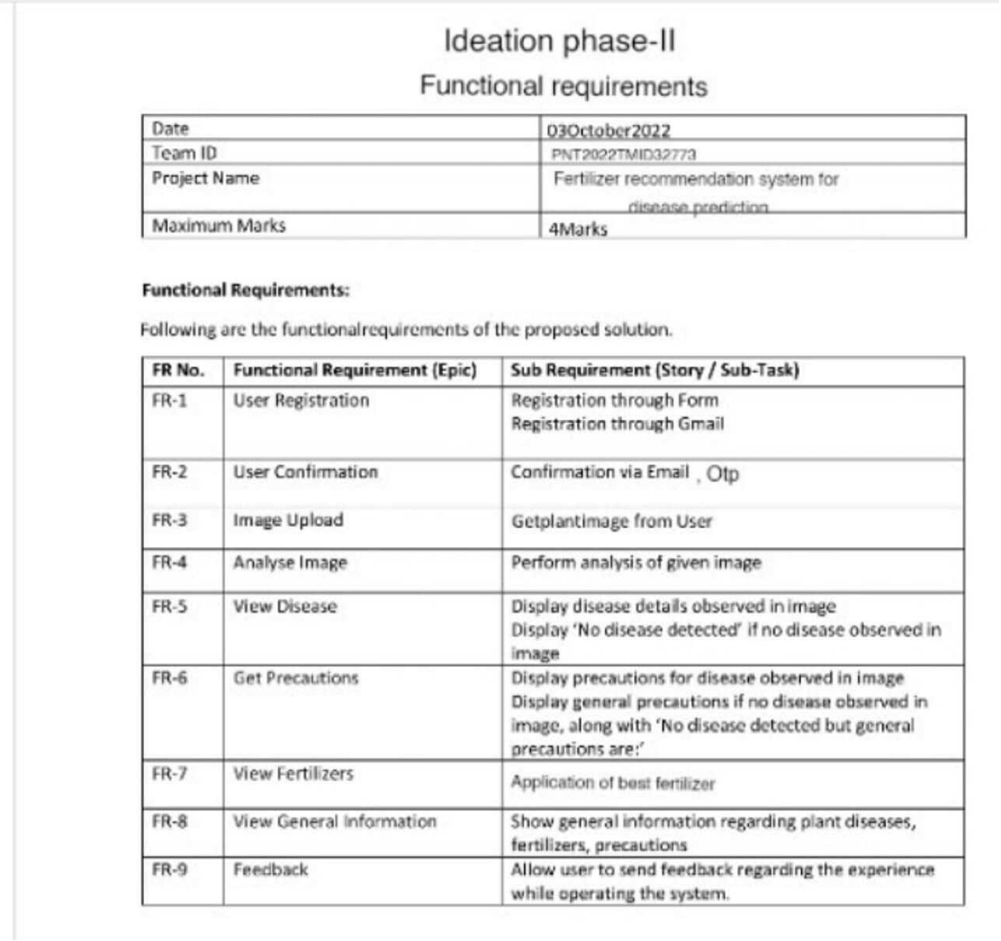
8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Ask friends/family for solutions/treatment; read newspaper, magazines regarding common diseases

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:



Ideation phase-II

Functional requirements

Date	03October2022
Team ID	PNT2022TMID32773
Project Name	Fertilizer recommendation system for disease prediction
Maximum Marks	4Marks

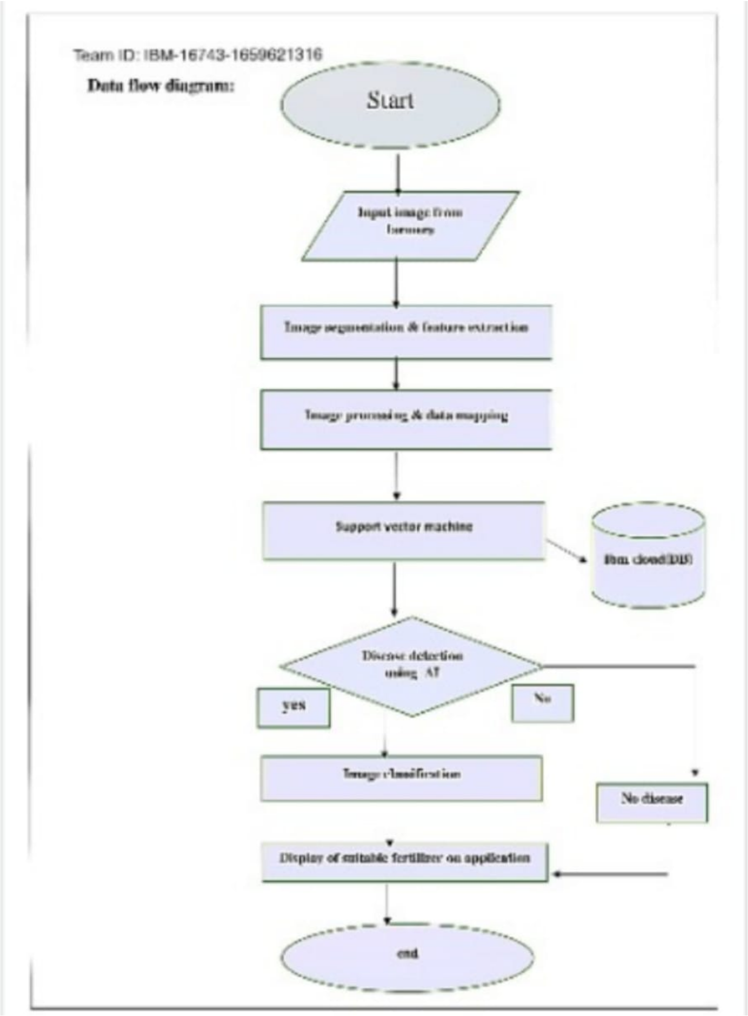
Functional Requirements:

Following are the functionalrequirements of the proposed solution.

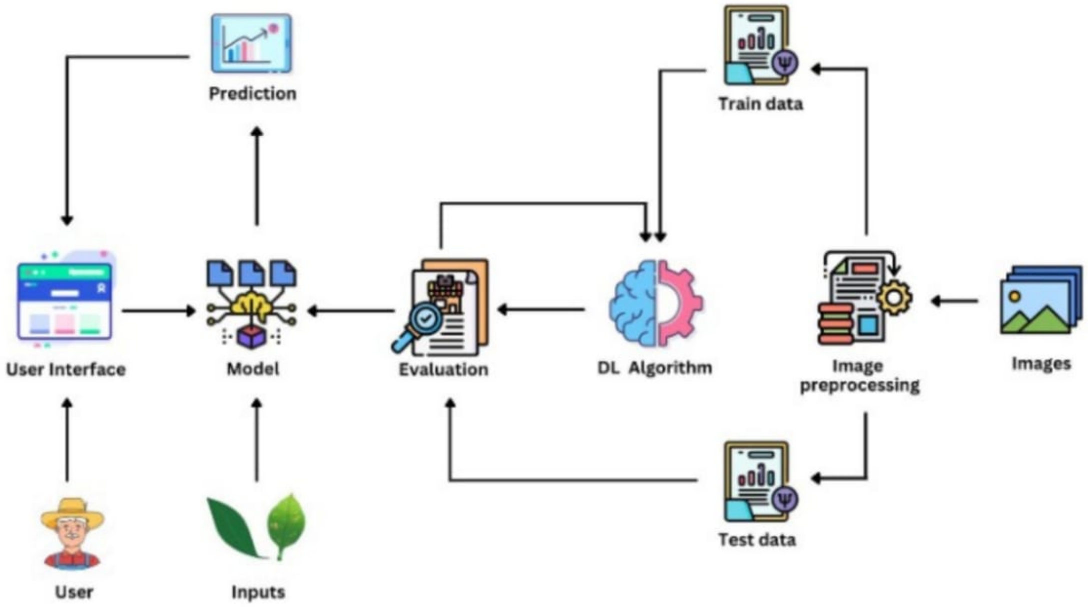
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email , Otp
FR-3	Image Upload	Getplantimage from User
FR-4	Analyse Image	Perform analysis of given image
FR-5	View Disease	Display disease details observed in image Display 'No disease detected' if no disease observed in image
FR-6	Get Precautions	Display precautions for disease observed in image Display general precautions if no disease observed in image, along with 'No disease detected but general precautions are:'
FR-7	View Fertilizers	Application of best fertilizer
FR-8	View General Information	Show general information regarding plant diseases, fertilizers, precautions
FR-9	Feedback	Allow user to send feedback regarding the experience while operating the system.

5. PROJECT DESIGN

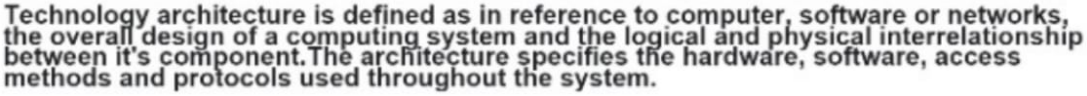
5.1 Data Flow Diagrams


Team ID: IBM-16743-1659621316
Data flow diagram:
Start
Input image from farmers
Image segmentation & feature extraction
Image processing & data mapping
Support vector machine
Ibm cloud(DB)
Disease detection using AI
yes
No
Image classification
No disease
Display of suitable fertilizer on application
end

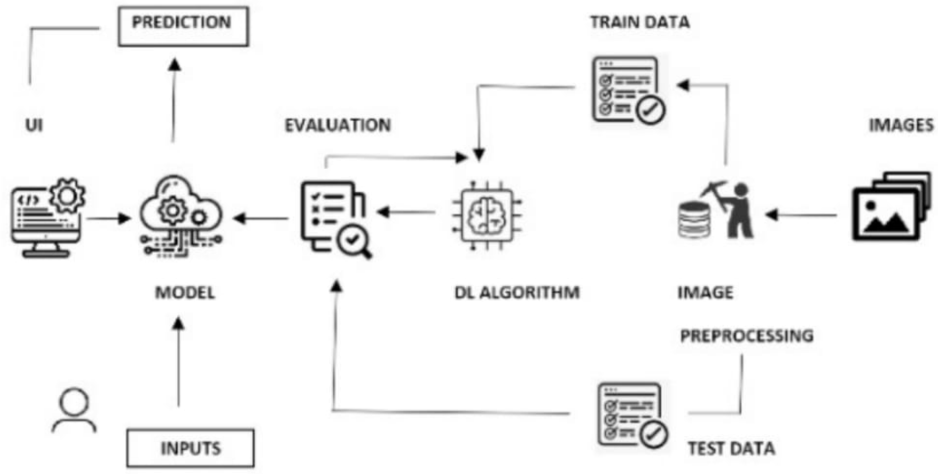
5.2 Solution Architecture



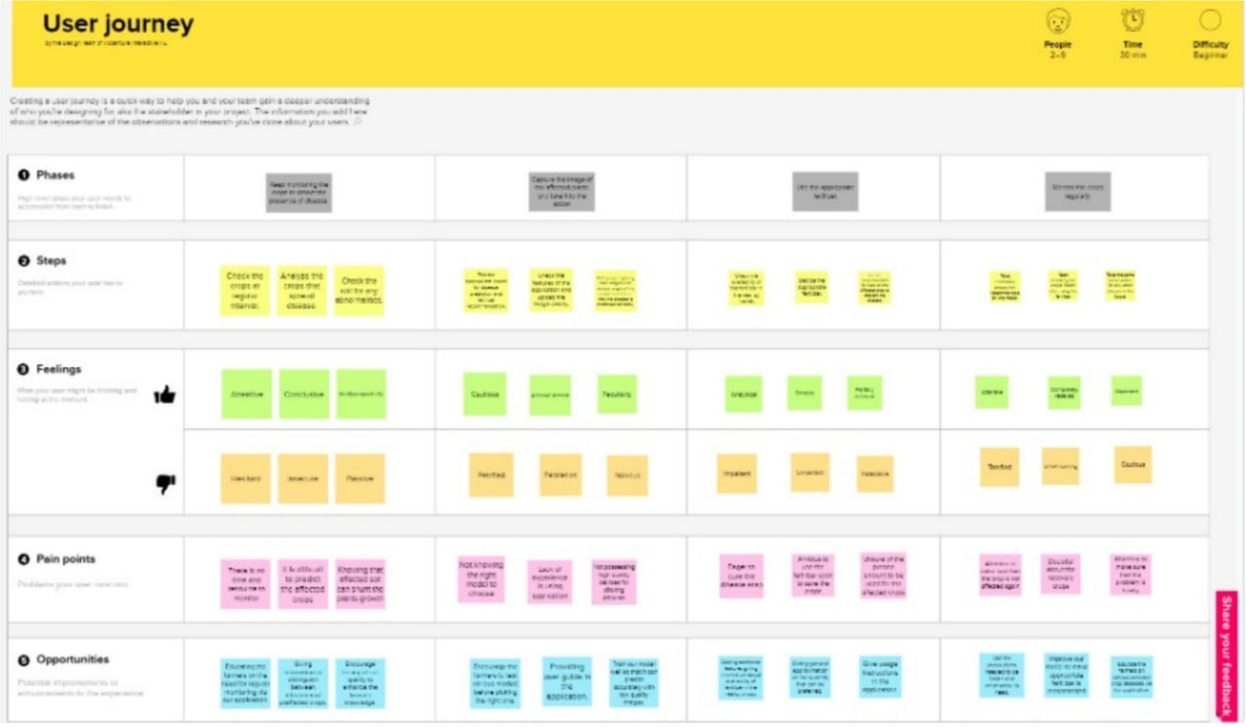
Technical Architecture



Technology architecture is defined as in reference to computer, software or networks, the overall design of a computing system and the logical and physical interrelationship between it's component.The architecture specifies the hardware, software, access methods and protocols used throughout the system.



5.3 User stories



User journey

People 2–5 Time 30 min Difficulty Beginner

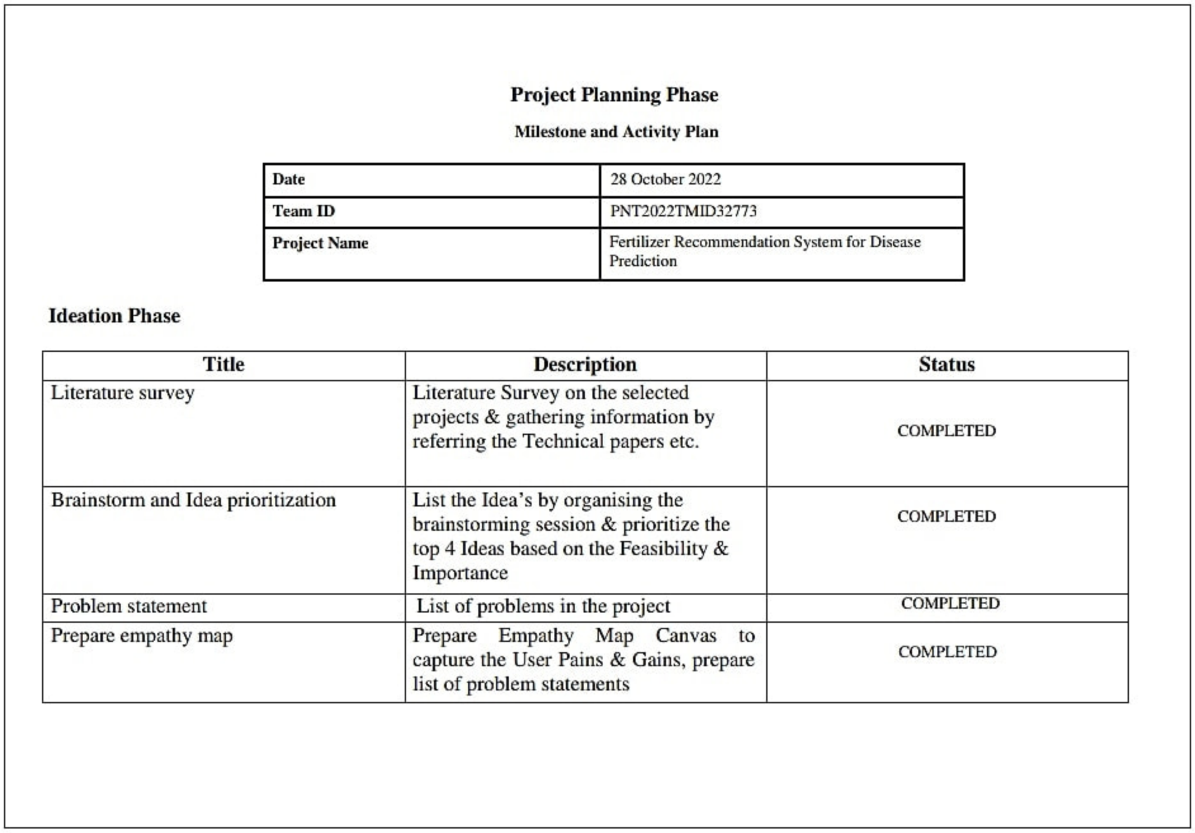
Creating a user journey is a quick way to help you and your team gain a deeper understanding of who you're designing for, aka the stakeholder in your project. The information you add here should be representative of the observations and research you've done about your users.

1 Phases High-level steps your user needs to accomplish from start to finish.	Keep monitoring the crops to detect the presence of disease.	Capture the image of the affected plant and take it to the action	Use the appropriate fertiliser	Monitor the crops regularly
2 Steps Detailed actions your user has to perform.	Check the crops at regular intervals. Analyse the crops that spread disease. Check the soil for any abnormalities.	[illegible] [illegible] [illegible]	[illegible] Decide the appropriate fertiliser. [illegible]	[illegible] [illegible] [illegible]
3 Feelings What your user might be thinking and feeling at the moment. 👍	Attentive Conclusive [illegible]	Cautious [illegible] Resolutely	Analytical [illegible] [illegible]	[illegible] [illegible] [illegible]
👎	[illegible] [illegible] Passive	Terrified Perplexed [illegible]	Impatient Uncertain Indecisive	Terrified [illegible] Cautious
4 Pain points Problems your user runs into.	There is no time and resources to monitor It is difficult to predict the affected crops Knowing that affected soil can stunt the plants growth	Not knowing the right model to choose Lack of experience in using application Not possessing high quality devices for clicking pictures	Eager to cure the disease soon Anxious to use the fertiliser soon to save the crops Unsure of the correct amount to be used for the affected crops	[illegible] Doubtful about the recovery stage Attentive to make sure that the problem is [illegible]
5 Opportunities Potential improvements or enhancements to the experience.	Educate the farmers on the need for regular monitoring via our application Giving [illegible] between affected and unaffected crops Encourage [illegible] quality to enhance the farmer's knowledge	Encourage the farmers to test various models before picking the right one Providing user guide to the application. Train our model well so that it can predict accuracy with low quality images	[illegible] [illegible] Give usage instructions in the application	[illegible] Improve our model to display appropriate fertiliser [illegible] [illegible]

Share your feedback

6.PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION



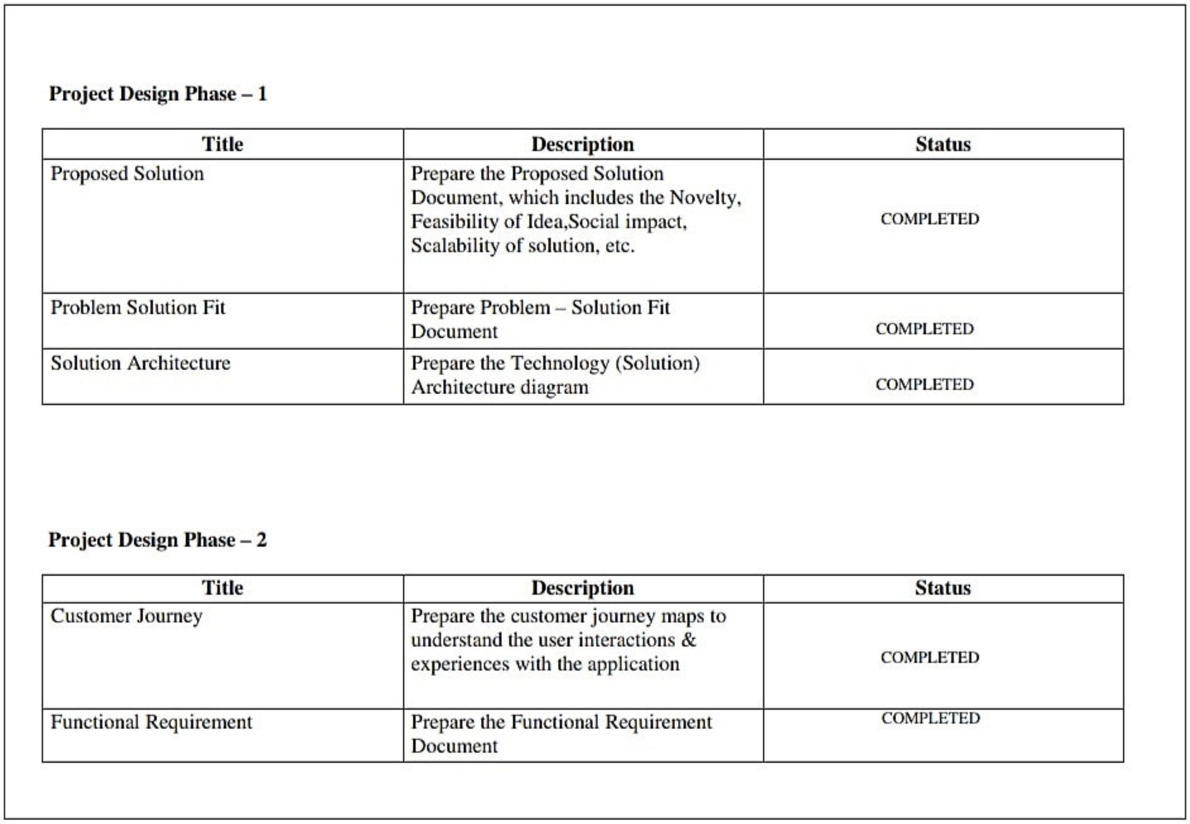
Project Planning Phase

Milestone and Activity Plan

Date	28 October 2022
Team ID	PNT2022TMID32773
Project Name	Fertilizer Recommendation System for Disease Prediction

Ideation Phase

Title	Description	Status
Literature survey	Literature Survey on the selected projects & gathering information by referring the Technical papers etc.	COMPLETED
Brainstorm and Idea prioritization	List the Idea's by organising the brainstorming session & prioritize the top 4 Ideas based on the Feasibility & Importance	COMPLETED
Problem statement	List of problems in the project	COMPLETED
Prepare empathy map	Prepare Empathy Map Canvas to capture the User Pains & Gains, prepare list of problem statements	COMPLETED

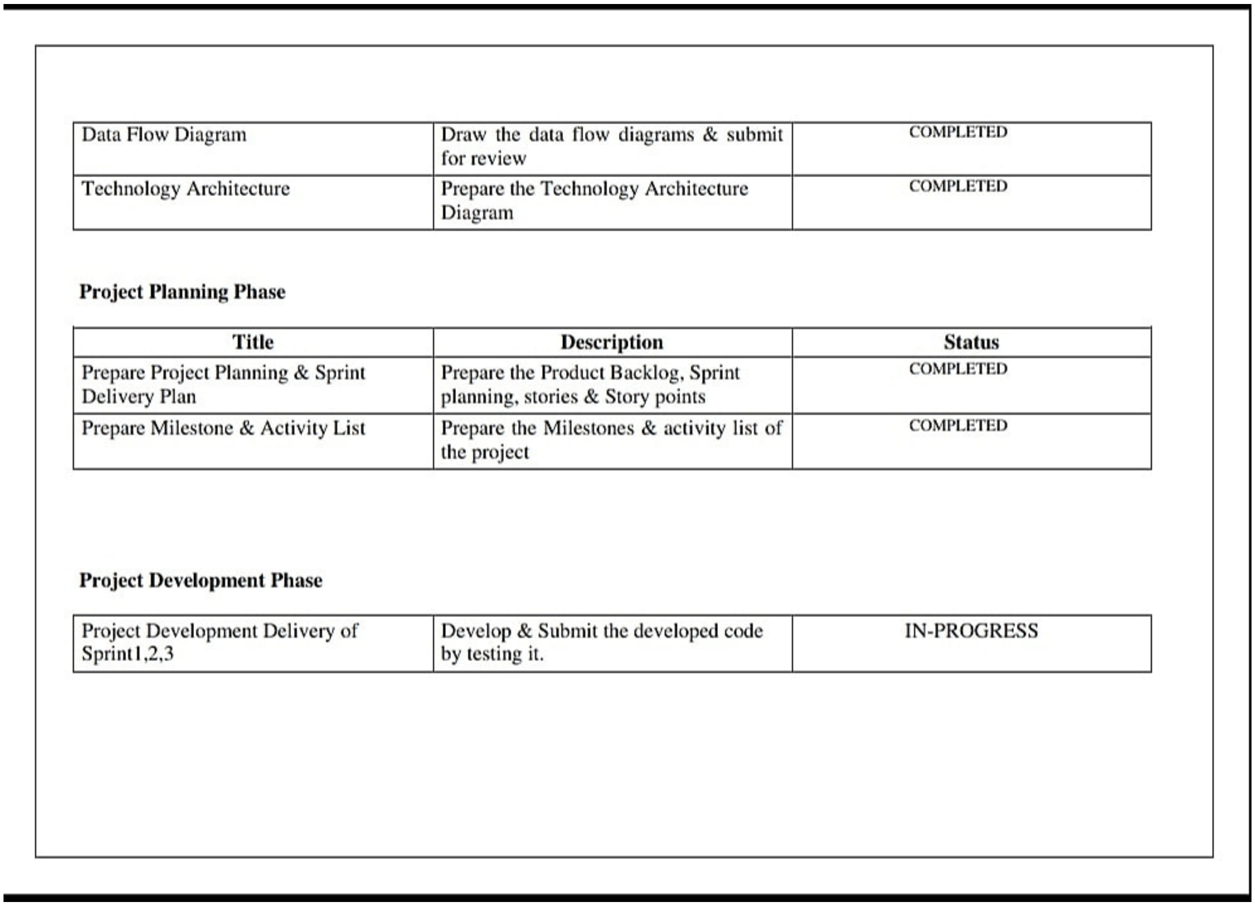


Project Design Phase – 1

Title	Description	Status
Proposed Solution	Prepare the Proposed Solution Document, which includes the Novelty, Feasibility of Idea,Social impact, Scalability of solution, etc.	COMPLETED
Problem Solution Fit	Prepare Problem – Solution Fit Document	COMPLETED
Solution Architecture	Prepare the Technology (Solution) Architecture diagram	COMPLETED

Project Design Phase – 2

Title	Description	Status
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application	COMPLETED
Functional Requirement	Prepare the Functional Requirement Document	COMPLETED



Data Flow Diagram	Draw the data flow diagrams & submit for review	COMPLETED
Technology Architecture	Prepare the Technology Architecture Diagram	COMPLETED

Project Planning Phase

Title	Description	Status
Prepare Project Planning & Sprint Delivery Plan	Prepare the Product Backlog, Sprint planning, stories & Story points	COMPLETED
Prepare Milestone & Activity List	Prepare the Milestones & activity list of the project	COMPLETED

Project Development Phase

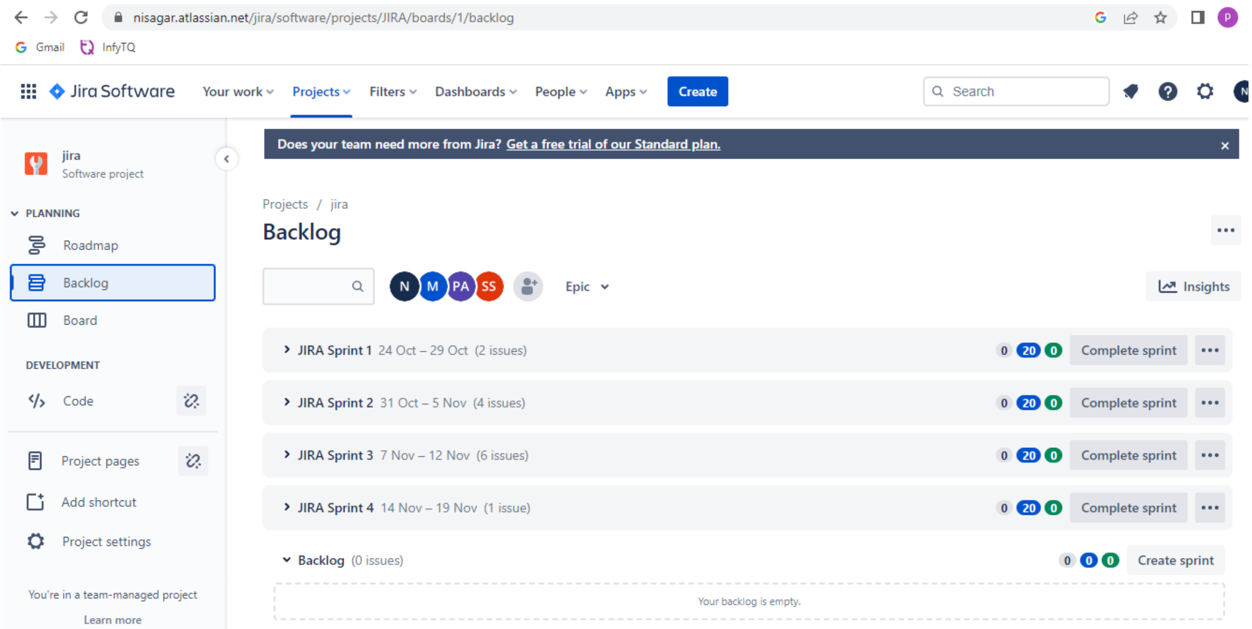
Project Development Delivery of Sprint1,2,3	Develop & Submit the developed code by testing it.	IN-PROGRESS

Sprint	Functional Requirement (Epic)	User story number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dataset collection	USN-1	Collect the dataset	6	Medium	Praveena
	Image Preprocessing	USN-2	Process the images which can be collected from the user	14	High	Nisagar Poojapriyadharshini Suriyasri
Sprint-2	Model Building for fruit disease prediction	USN-4	Create a model which can classify diseased fruit plants from given images.	6	Medium	Poojapriyadharshini Praveena
	Fruit Dataset	USN-5	Datasets with fruits	4	Low	Nisagar Suriyasri
	Model Building for vegetable disease prediction	USN-6	Create a model which can classify diseased vegetable plants from given images.	6	Medium	Nisagar Suriyasri
	Vegetable Dataset	USN-7	Datasets with vegetables	4	Low	Poojapriyadharshini Praveena
Sprint-3	Test both models	USN-8	Test the both vegetables and fruits models using the datasets.	4	Low	Nisagar Suriyasri
	Application buildings	USN-9	Build the application	2	Low	Nisagar
	Registration	USN-10	As a user/admin/shopkeeper,I can log into the application by entering email & password	3	Medium	Poojapriyadharshini
	Redirect the page	USN-11	Page can be redirected to Another page	5	Medium	Poojapriyadharshini Praveena

	Upload Images	USN-12	Here we upload the diseased fruit images	2	Low	Suriyasri
	Recoomend-ed Results	USN-13	As an admin,I can view other user details and uploads for other purposes	4	Medium	Nisagar Suriyasri
Sprint-4	Train the models on IBM cloud	USN-14	Created model can be tested by IBM cloud .	20	High	Nisagar Poojapriyadharshini Praveena Suriyasri

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	31 Nov 2022
Sprint 2	20	6 days	31 Oct 2022	5 Nov 2022	20	6 Nov 2022
Sprint 3	20	6 days	7 Nov 2022	12 Nov 2022	20	13 Nov 2022
Sprint 4	20	6 days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORTS FROM JIRA



nisagar.atlassian.net/jira/software/projects/JIRA/boards/1/backlog

Does your team need more from Jira? Get a free trial of our Standard plan.

You're in a team-managed project

Learn more

Projects / jira

Backlog

JIRA Sprint 1 24 Oct – 29 Oct (2 issues) 0 20 0 Complete sprint

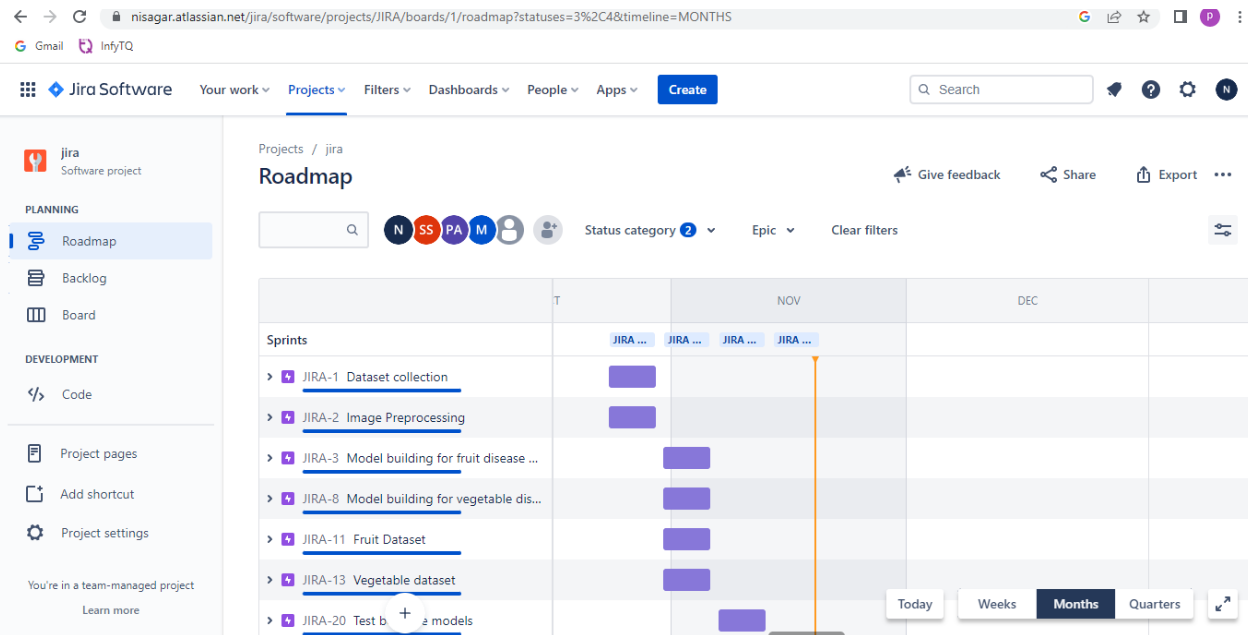
JIRA Sprint 2 31 Oct – 5 Nov (4 issues) 0 20 0 Complete sprint

JIRA Sprint 3 7 Nov – 12 Nov (6 issues) 0 20 0 Complete sprint

JIRA Sprint 4 14 Nov – 19 Nov (1 issue) 0 20 0 Complete sprint

Backlog (0 issues) 0 0 0 Create sprint

Your backlog is empty.



nisagar.atlassian.net/jira/software/projects/JIRA/boards/1/roadmap?statuses=3%2C4&timeline=MONTHS

You're in a team-managed project

Learn more

Projects / jira

Roadmap

JIRA-1 Dataset collection

JIRA-2 Image Preprocessing

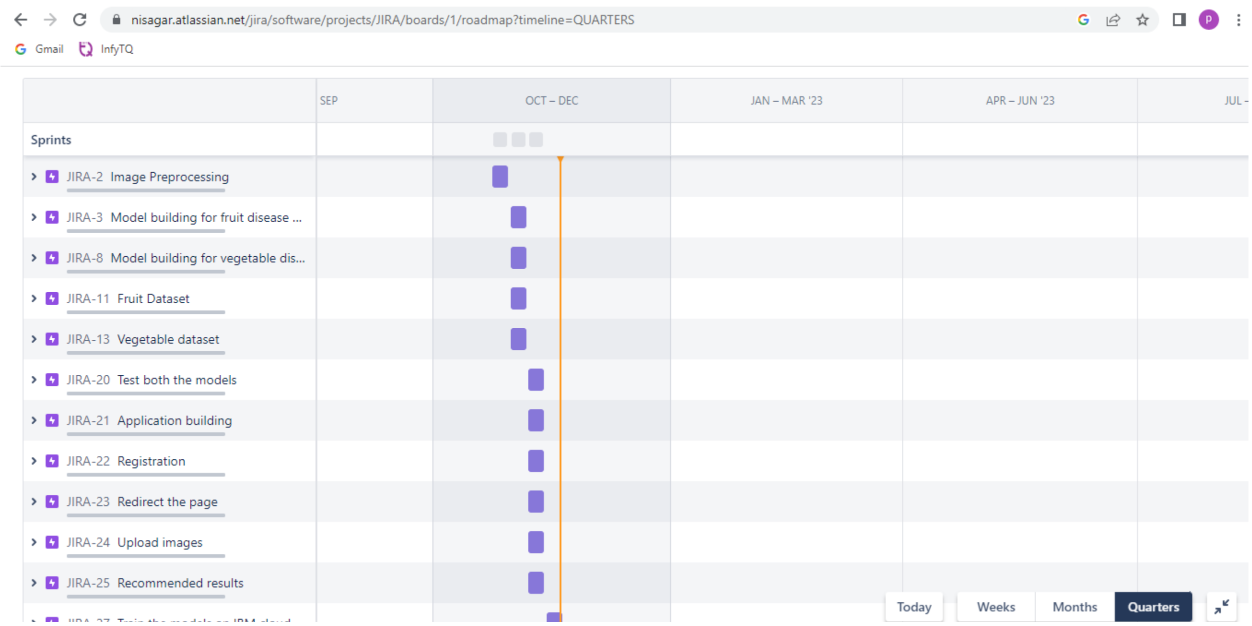
JIRA-3 Model building for fruit disease ...

JIRA-8 Model building for vegetable dis...

JIRA-11 Fruit Dataset

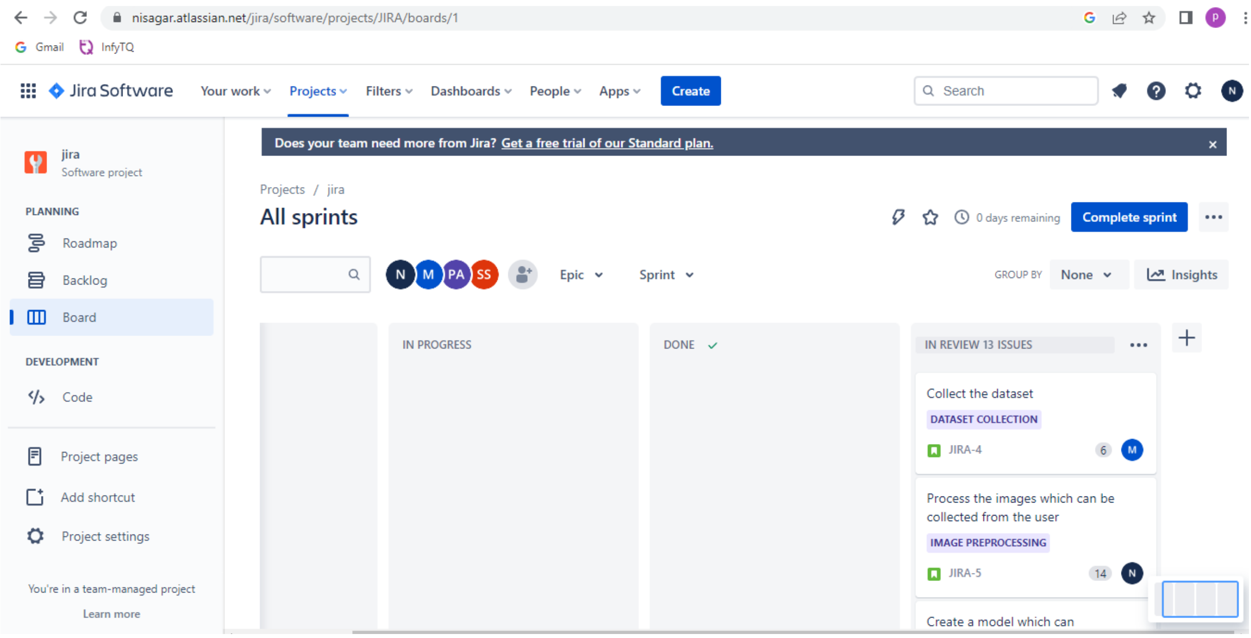
JIRA-13 Vegetable dataset

JIRA-20 Test b + e models


nisagar.atlassian.net/jira/software/projects/JIRA/boards/1/roadmap?timeline=QUARTERS
Gmail
InfyTQ
SEP
OCT – DEC
JAN – MAR '23
APR – JUN '23
JUL –
Today
Weeks
Months
Quarters

Sprints

- JIRA-2 Image Preprocessing
- JIRA-3 Model building for fruit disease ...
- JIRA-8 Model building for vegetable dis...
- JIRA-11 Fruit Dataset
- JIRA-13 Vegetable dataset
- JIRA-20 Test both the models
- JIRA-21 Application building
- JIRA-22 Registration
- JIRA-23 Redirect the page
- JIRA-24 Upload images
- JIRA-25 Recommended results
- JIRA-27 Train the models on IBM cloud



Does your team need more from Jira? Get a free trial of our Standard plan.

Projects / jira

All sprints

IN PROGRESS

DONE ✓

IN REVIEW 13 ISSUES

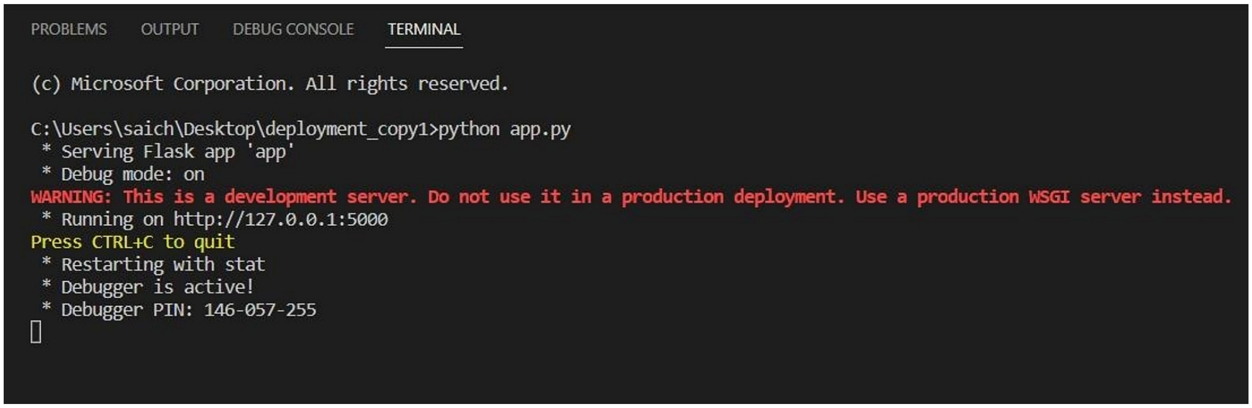
Collect the dataset
DATASET COLLECTION
JIRA-4 6 M

Process the images which can be collected from the user
IMAGE PREPROCESSING
JIRA-5 14 N

Create a model which can

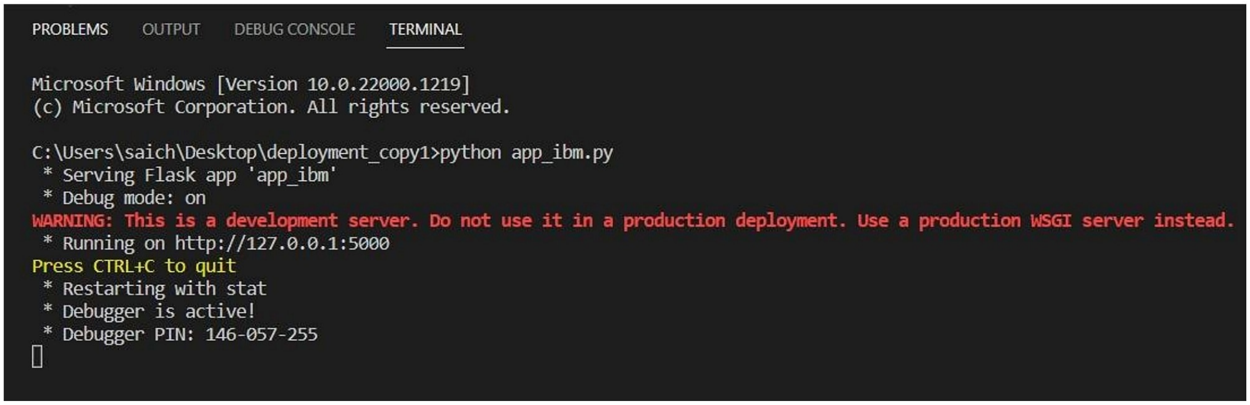
7.CODING AND SOLUTIONING

7.1 Feature 1



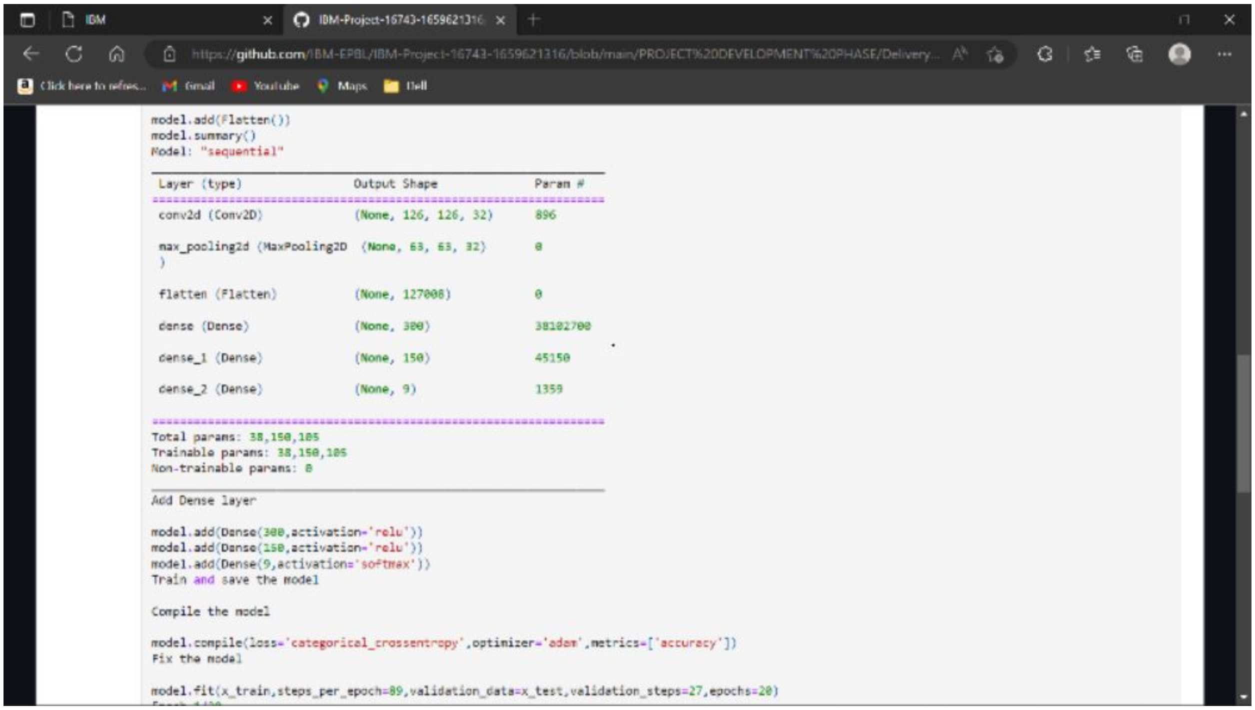
```
(c) Microsoft Corporation. All rights reserved.

C:\Users\saich\Desktop\deployment_copy1>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 146-057-255
```

7.2 Feature 2

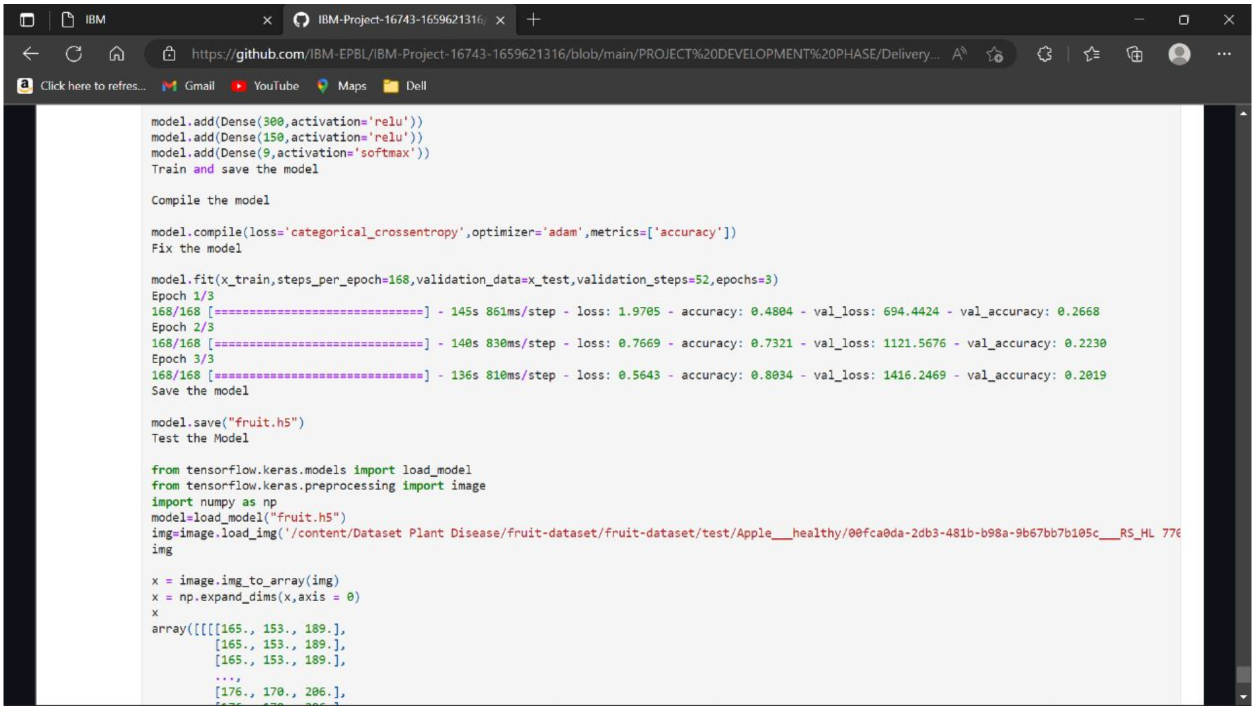


```
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\Users\saich\Desktop\deployment_copy1>python app_ibm.py
 * Serving Flask app 'app_ibm'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 146-057-255
```

8.TEST CASES


IBM
IBM-Project-16743-1659621316
https://github.com/IBM-EPBL/IBM-Project-16743-1659621316/blob/main/PROJECT%20DEVELOPMENT%20PHASE/Delivery...
Click here to refres...
Gmail
Youtube
Maps
Dell

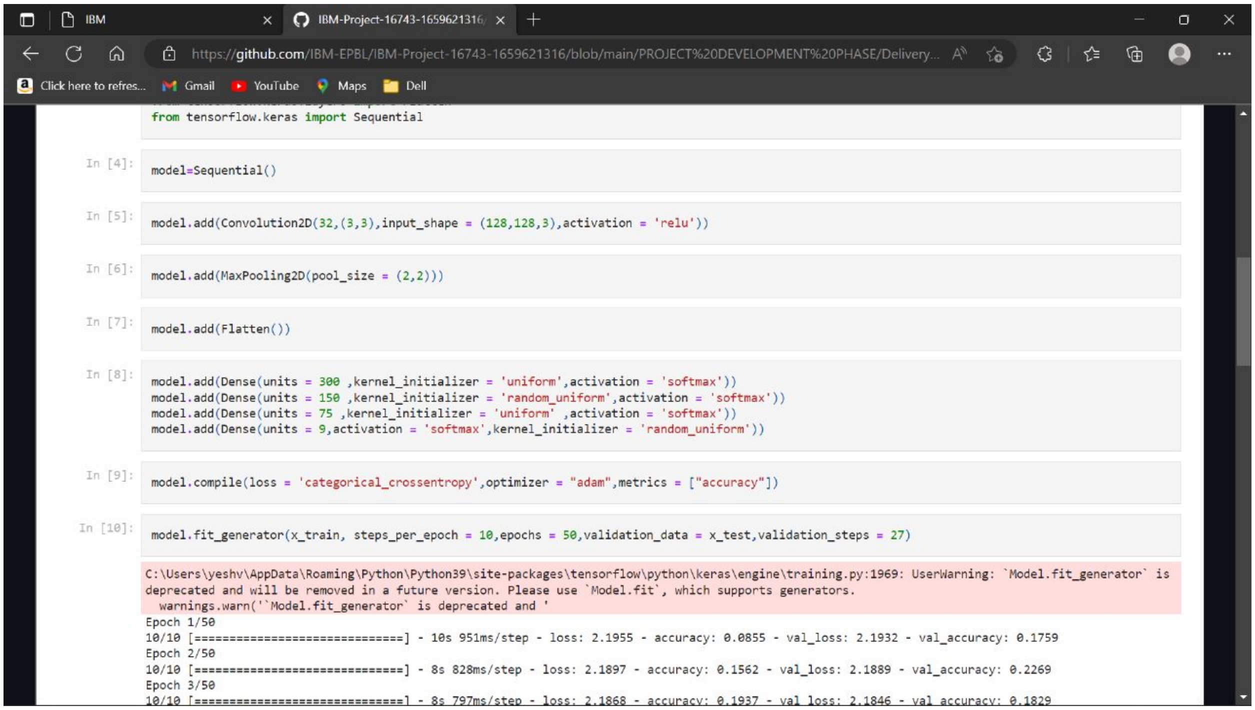
```
model.add(Flatten())
model.summary()
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 126, 126, 32)      896

 max_pooling2d (MaxPooling2D  (None, 63, 63, 32)       0
 )

 flatten (Flatten)           (None, 127008)            0

 dense (Dense)               (None, 300)               38102700

 dense_1 (Dense)             (None, 150)               45150

 dense_2 (Dense)             (None, 9)                 1359

=================================================================
Total params: 38,150,105
Trainable params: 38,150,105
Non-trainable params: 0
_________________________________________________________________
Add Dense layer

model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))
Train and save the model

Compile the model

model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
Fix the model

model.fit(x_train,steps_per_epoch=89,validation_data=x_test,validation_steps=27,epochs=20)
```


IBM
IBM-Project-16743-1659621316
https://github.com/IBM-EPBL/IBM-Project-16743-1659621316/blob/main/PROJECT%20DEVELOPMENT%20PHASE/Delivery...
Click here to refres...
Gmail
YouTube
Maps
Dell

```
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))
Train and save the model

Compile the model

model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
Fix the model

model.fit(x_train,steps_per_epoch=168,validation_data=x_test,validation_steps=52,epochs=3)
Epoch 1/3
168/168 [==============================] - 145s 861ms/step - loss: 1.9705 - accuracy: 0.4804 - val_loss: 694.4424 - val_accuracy: 0.2668
Epoch 2/3
168/168 [==============================] - 140s 830ms/step - loss: 0.7669 - accuracy: 0.7321 - val_loss: 1121.5676 - val_accuracy: 0.2230
Epoch 3/3
168/168 [==============================] - 136s 810ms/step - loss: 0.5643 - accuracy: 0.8034 - val_loss: 1416.2469 - val_accuracy: 0.2019
Save the model

model.save("fruit.h5")
Test the Model

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
model=load_model("fruit.h5")
img=image.load_img('/content/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___healthy/00fca0da-2db3-481b-b98a-9b67bb7b105c___RS_HL 776
img

x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
x
array([[[[165., 153., 189.],
         [165., 153., 189.],
         [165., 153., 189.],
         ...,
         [176., 170., 206.],
```

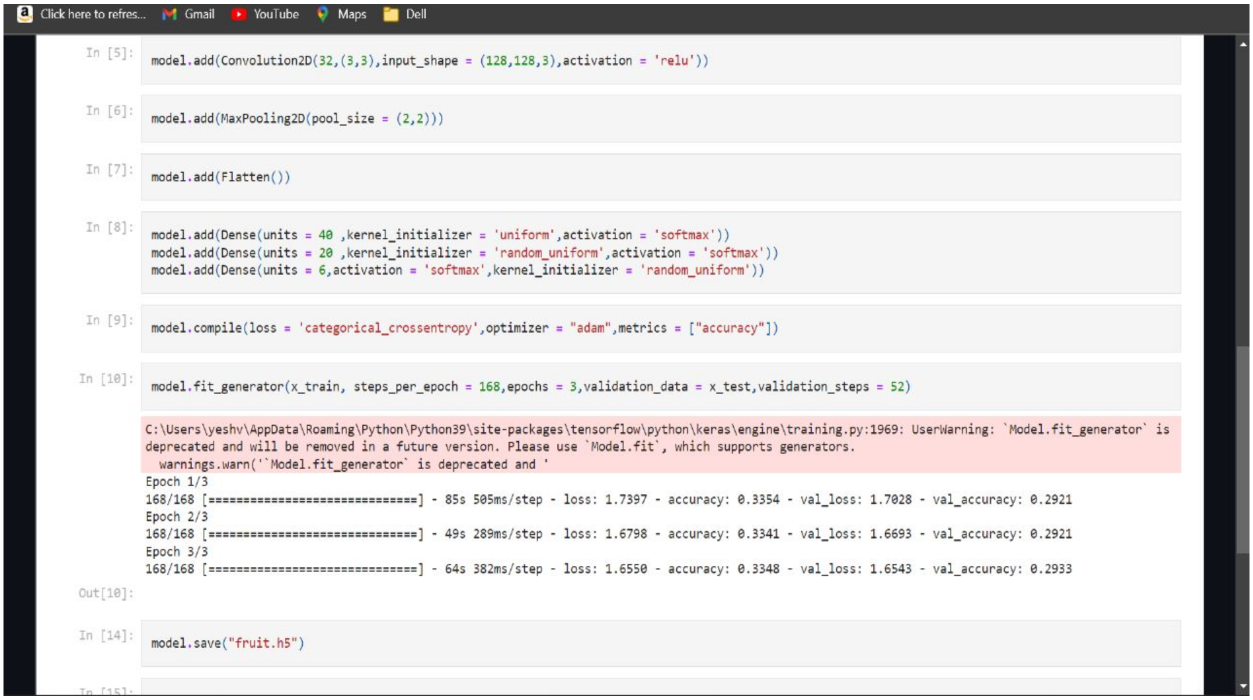
9.RESULTS

9.1 MODEL PERFORMANCE TESTING



IBM
IBM-Project-16743-1659621316
https://github.com/IBM-EPBL/IBM-Project-16743-1659621316/blob/main/PROJECT%20DEVELOPMENT%20PHASE/Delivery...
Click here to refres...
Gmail
YouTube
Maps
Dell

```
from tensorflow.keras import Sequential

In [4]:  model=Sequential()

In [5]:  model.add(Convolution2D(32,(3,3),input_shape = (128,128,3),activation = 'relu'))

In [6]:  model.add(MaxPooling2D(pool_size = (2,2)))

In [7]:  model.add(Flatten())

In [8]:  model.add(Dense(units = 300 ,kernel_initializer = 'uniform',activation = 'softmax'))
         model.add(Dense(units = 150 ,kernel_initializer = 'random_uniform',activation = 'softmax'))
         model.add(Dense(units = 75 ,kernel_initializer = 'uniform' ,activation = 'softmax'))
         model.add(Dense(units = 9,activation = 'softmax',kernel_initializer = 'random_uniform'))

In [9]:  model.compile(loss = 'categorical_crossentropy',optimizer = "adam",metrics = ["accuracy"])

In [10]: model.fit_generator(x_train, steps_per_epoch = 10,epochs = 50,validation_data = x_test,validation_steps = 27)

C:\Users\yeshv\AppData\Roaming\Python\Python39\site-packages\tensorflow\python\keras\engine\training.py:1969: 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 1/50
10/10 [==============================] - 10s 951ms/step - loss: 2.1955 - accuracy: 0.0855 - val_loss: 2.1932 - val_accuracy: 0.1759
Epoch 2/50
10/10 [==============================] - 8s 828ms/step - loss: 2.1897 - accuracy: 0.1562 - val_loss: 2.1889 - val_accuracy: 0.2269
Epoch 3/50
10/10 [==============================] - 8s 797ms/step - loss: 2.1868 - accuracy: 0.1937 - val_loss: 2.1846 - val_accuracy: 0.1829
```



```
In [5]:  model.add(Convolution2D(32,(3,3),input_shape = (128,128,3),activation = 'relu'))

In [6]:  model.add(MaxPooling2D(pool_size = (2,2)))

In [7]:  model.add(Flatten())

In [8]:  model.add(Dense(units = 40 ,kernel_initializer = 'uniform',activation = 'softmax'))
         model.add(Dense(units = 20 ,kernel_initializer = 'random_uniform',activation = 'softmax'))
         model.add(Dense(units = 6,activation = 'softmax',kernel_initializer = 'random_uniform'))

In [9]:  model.compile(loss = 'categorical_crossentropy',optimizer = "adam",metrics = ["accuracy"])

In [10]: model.fit_generator(x_train, steps_per_epoch = 168,epochs = 3,validation_data = x_test,validation_steps = 52)

C:\Users\yeshv\AppData\Roaming\Python\Python39\site-packages\tensorflow\python\keras\engine\training.py:1969: 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 1/3
168/168 [==============================] - 85s 505ms/step - loss: 1.7397 - accuracy: 0.3354 - val_loss: 1.7028 - val_accuracy: 0.2921
Epoch 2/3
168/168 [==============================] - 49s 289ms/step - loss: 1.6798 - accuracy: 0.3341 - val_loss: 1.6693 - val_accuracy: 0.2921
Epoch 3/3
168/168 [==============================] - 64s 382ms/step - loss: 1.6550 - accuracy: 0.3348 - val_loss: 1.6543 - val_accuracy: 0.2933

Out[10]:

In [14]: model.save("fruit.h5")

In [15]:
```

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

1.**They are quick in providing plant nutrients and restoring soil fertility**. They are portable and easy to transport. Plants easily absorb fertilizers. Fertilizers improve and increase the productivity of many crops such as wheat, maize, and rice.

2.**Without fertilizers, nature struggles to replenish the nutrients in the soil**. When crops are harvested, important nutrients are removed from the soil, because they follow the crop and end up at the dinner table. If the soil is not replenished with nutrients through fertilizing, crop yields will deteriorate over time.

3.**To supply each palm with adequate nutrients in balanced proportion to ensure healthy vegetative growth and optimum economic FFB yields**. To apply the fertilisers in the prescribed manner over the areas of the estate that are likely to result in the most efficient uptake of nutrients.

4.The most important characteristics of a fertilizer is to be adapted to the kind of crop you are growing and to the different stages of growth and maturation (ratio of N,P,K) and the physical and chemical nature of the soil (capacity of ions adsorption, pH, presence of humus).

DISADVANTAGES

1.Synthetic fertilizers often contain toxins that can be destructive to the soil, and the chemicals in these fertilizers can be poisonous to humans, wildlife, and marine life if they reach the oceans. Fertilizers can also leach through soil into groundwater, making it very harmful to the surrounding environment.

2.Adverse impacts of fertilizers are mainly caused by their excessive and inefficient use. This leads to nutrient losses to the environment and other adverse impacts, such as drinking water contamination and eutrophication of freshwater systems and coastal zones

11.CONCLUSION

The TPF-CNN dual operator approach makes the insecticide recommendation operation efficient and compact. The proposed system consists of combined insecticide and fertilizer recommendation systems, which will help farmers gain maximum farm yield. Also, the soil nutrients would be managed efficiently, resulting in nutrient-rich soil. The cost incurred for laboratory testing of soil nutrients will reduce. The proposed approach gives the recommendation of insecticides in a short time of 10 s and fertilizer recommendation in 60 s only. Compared to other approaches such as KNN, SVM, and ANN, it gives nearly 20% higher performance. This system can be used anywhere as it is standalone and does not require an internet connection. In the future, the system can be integrated with more sensors such as pH, temperature, humidity, and moisture sensors for open and indoor farming. Also, this system can be used in online and offline modes. This system can be recommended for farmers, soil testing laboratories, and seed hybridizing companies. The limitations of this model are it does not save any data on the system or cloud database.

12.FUTURE SCOPE

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.

13.APPENDIX

SOURCE CODE

HTML CODE

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-
scale=1.0">
   <meta http-equiv="X-UA-Compatible" content="ie=edge">
   <title>Home</title>
   <link
href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.mi
n.css" rel="stylesheet">
   <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.j
s"></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="E:\Plant Disease\flask\static\css\main.css" rel="stylesheet">
<style>
body
{
    background-image: url("https://purepng.com/public/uploads/large/purepng.com-grassnaturerosegreencartoonillustrationgrassfieldlandscape-961524676349chnpr.png");
    background-size: cover;
}
.bar
{
margin: 0px;
padding:20px;
background-color:white;
opacity:0.6;
color:black;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:25px;
```

```
}
h3
{
margin: 0px;
padding:20px;
background-color:#9ACD32;
width: 800px;
opacity:0.6;
color:#000000;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:25px;
}
a
{
color:black;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
color:white;
border-radius:15px;0
font-size:30px;
padding-left:10px;
```

```
}
.div1{
  background-color: lightgrey;
  width: 500px;
  border: 10px solid peach;
  padding: 20px;
  margin: 20px;
  height: 500px;
}
```

```
.header {  position: relative;
                 top:0;
                 margin:0px;
                 z-index: 1;
                 left: 0px;
                 right: 0px;
                 position: fixed;
                 background-color: #8B008B ;
                 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: #FCAD98;
}

.topnav-right a {
  float: left;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 22px;
}

.topnav-right a:hover {
  background-color: #FF69B4;
  color: white;
}

.topnav-right a.active {
  background-color: #DA70D6;
  color: white;
}
```

```
.topnav-right {
  float: right;
  padding-right:100px;
}
</style>
</head>
<body>

<!--Brian Tracy-->

<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-
align:left;color:white; padding-top:1%;padding-left:5%;">Plant
Disease Prediction</div>
  <div class="topnav-right"style="padding-top:0.5%;">

    <a class="active" href="E:\Plant
Disease\flask\template\home.html">Home</a>
    <a   href="E:\Plant
Disease\flask\template\predict.html">Predict</a>
  </div>
</div>
</div>
<br>
<br>
<br>
<br>
<br>
<br>
```

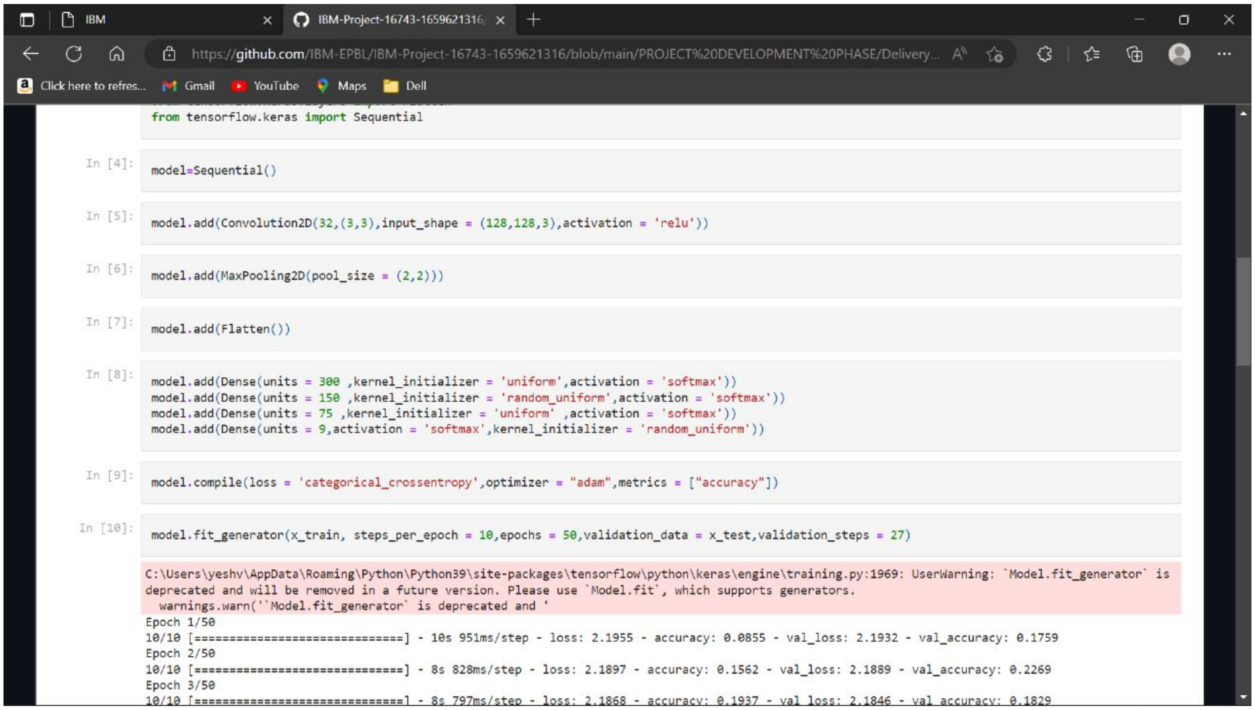
```
<br>
<br>
<h1>
<h2>
<center>
    Detect If your Plant is Infected!!
</center>
</h2>

<center>
<h3>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
    laboratries. This application helps farmers in detecting the
diseases by observing the spots on the leaves, which inturn saves
effort and labor costs.</h3>
</center>


</h1>
</body>
</html>
```

PREDICT CODE

```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "source": [
        "<!DOCTYPE html>\n",
        "<html >\n",
        "\n",
        "<head>\n",
        "  <meta charset=\"UTF-8\">\n",
        "  <meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">\n",
        "  <title> Plant Disease Prediction</title>\n",
        "  <link
href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>\n",
        "<link href='https://fonts.googleapis.com/css?family=Arimo'
rel='stylesheet' type='text/css'>\n",
        "<link
href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>\n",
        "<link
href=\"https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.m
in.css\" rel=\"stylesheet\">\n",
        "    <script
src=\"https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min
.js\"></script>\n",
        "    <script
src=\"https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js\"></scri
pt>\n",
        "    <script
src=\"https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.j
s\"></script>\n",
        "<link
href='https://fonts.googleapis.com/css?family=Open+Sans+Cond
ensed:300' rel='stylesheet' type='text/css'>\n",
        "<link
href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>\n",
        "<link href='https://fonts.googleapis.com/css?family=Josefin
Sans' rel='stylesheet'>\n",
        "<link
href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>\n",
        "<link href=\"{{ url_for('static', filename='css/final.css') }}\"
rel=\"stylesheet\"> \n",
        "<style>\n",
        ".header {\t\n",
        "\t\t\ttop:0;\t\n",
        "\t\t\tmargin:0px;\n",
        "\t\t\tleft: 0px;\n",
        "\t\t\tright: 0px;\n",
        "\t\t\tposition: fixed;\n",
        "\t\t\tbackground-color: #28272c;\n",
        "\t\t\tcolor: white;\n",
        "\t\t\tbox-shadow: 0px 8px 4px grey;\n",
        "\t\t\toverflow: hidden;\n",
        "\t\t\tpadding-left:20px;\n",
      "\t\t\tfont-family: 'Josefin Sans';\n",
        "\t\t\tfont-size: 2vw;\n",
        "\t\t\twidth: 100%;\n",
        "\t\t\theight:8%;\n",
        "\t\t\ttext-align: center;\n",
        "\t\t}\n",
        "\t\t.topnav {\n",
        "  overflow: hidden;\n",
        "  background-color: #333;\n",
        "}\n",
        "\n",
        ".topnav-right a {\n",
        "  float: left;\n",
        "  color: #f2f2f2;\n",
        "  text-align: center;\n",
        "  padding: 14px 16px;\n",
        "  text-decoration: none;\n",
        "  font-size: 18px;\n",
        "}\n",
        "\n",
        ".topnav-right a:hover {\n",

      "\twidth: 100%; \n",
      "\tmargin-bottom: 10px; \n",
      "\tbackground: rgba(255,255,255,255);\n",
      "\tborder: none;\n",
      "\toutline: none;\n",
"  background-color: #ddd;\n",
        "  color: black;\n",
        "}\n",
        "\n",
        ".topnav-right a.active {\n",
        "  background-color: #565961;\n",
        "  color: white;\n",
        "}\n",
        "\n",
        ".topnav-right {\n",
        "  float: right;\n",
        "  padding-right:100px;\n",
        "}\n",
        "\t\t\n",
        ".login{\n",
        "margin-top:-70px;\n",
        "}\n",
        "body {\n",
        "\n",
        "  background-color:#ffffff;\n",
        "  background-repeat: no-repeat;\n",
        "  background-size:cover;\n",
        "  background-position: 0px 0px;\n",
        "  }\n",
        ".login{\n",
        "\tmargin-top:100px;\n",
        "}\n",
        "\n",
        ".container {\n",
        "  margin-top:40px;\n",
        "  padding: 16px;\n",
        "}\n",
        "select { \n",    "\tpadding: 10px;\n",
      "\tfont-size: 13px;\n",
        "\tcolor: #000000;\n",
        "\ttext-shadow: 1px 1px 1px rgba(0,0,0,0.3);\n",
        "\tborder: 1px solid rgba(0,0,0,0.3);\n",
        "\tborder-radius: 4px;\n",
        "\tbox-shadow: inset 0 -5px 45px rgba(100,100,100,0.2), 0
1px 1px rgba(255,255,255,0.2);\n",
        "\t-webkit-transition: box-shadow .5s ease;\n",
        "\t-moz-transition: box-shadow .5s ease;\n",
        "\t-o-transition: box-shadow .5s ease;\n",
        "\t-ms-transition: box-shadow .5s ease;\n",
        "\ttransition: box-shadow .5s ease;\n",
        "}\n",
        "\n",
        "  \n",
        "</style>\n",
        "</head>\n",
        "\n",
      "<body style=\"font-family:Montserrat;overflow:scroll;\">\n",
        "\n",
        "<div class=\"header\">    \n",
        " <div style=\"width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%\">Plant Disease Prediction</div>\n",
        "  <div class=\"topnav-right\" style=\"padding-top:0.5%;\">\n",
        "    \n",
        "    \n",
        "  </div>\n",
        "</div>\n",
        "<div class=\"container\">\n",
        "       <div id=\"content\" style=\"margin-top:2em\">\n",
        "\t\t<div class=\"container\">\n",
        "\t\t  <div class=\"row\">\n",
        "\t\t\t<div class=\"col-sm-6 bd\" >\n",
        "\t\t\t \n",
        "\t\t\t  <br>\n",
        "\t\t\t\t<img
```

```
src=\"{{url_for('static',filename='images/789.jpg')}}\"
style=\"height:450px;width:550px\"class=\"img-rounded\"
alt=\"Gesture\">       \n",
        "\t\t\t</div>\n",
        "\t\t\t<div class=\"col-sm-6\">\n",
        "\t\t\t\t<div>\n",
        "\t\t\t\t\t<h4>Drop in the image to get the prediction
</h4>\n",
        "\t\t\t<form action = \"\" id=\"upload-file\" method=\"post\"
enctype=\"multipart/form-data\">\n",
        "\t\t\t\t<select name=\"plant\">\n",
        "\t\t\t\n",
        "\t\t\t\t\t  <option value=\"select\" selected>Select plant
type</option>\n",
        "\t\t\t\t\t  <option value=\"fruit\">Fruit</option>\n",
        "\t\t\t\t\t  <option
value=\"vegetable\">Vegetable</option>\n",
        "\t\t</select><br>\n",
        "\t\t\t\t<label for=\"imageUpload\" class=\"upload-label\"
style=\"background: #28272c;\">\n",
        "\t\t\t\t\tChoose...\n",
        "\t\t\t\t</label>\n",
        "\t\t\t\t<input type=\"file\" name=\"image\"
id=\"imageUpload\" accept=\".png, .jpg, .jpeg\">\n",
        "\t\t\t</form>\n",
      "\t\t\t\n",
        "\n",
        "\t\t\t<div class=\"image-section\"
```

```
style=\"display:none;\">\n",
        "\t\t\t\t<div class=\"img-preview\">\n",
        "\t\t\t\t\t<div id=\"imagePreview\">\n",
        "\t\t\t\t\t</div>\n",
        "\t\t\t\t</div>\n",
        "\t\t\t\t<div>\n",
        "\t\t\t\t\t<button type=\"button\" class=\"btn btn-info btn-lg \"
id=\"btn-predict\" style=\"background:
#28272c;\">Predict!</button>\n",
        "\t\t\t\t</div>\n",
        "\t\t\t</div>\n",
        "\n",
        "\t\t\t<div class=\"loader\" style=\"display:none;\"></div>\n",
        "\n",
        "\t\t\t<h3>\n",
        "\t\t\t\t<span id=\"result\" style=\"font-size:17px; \">
</span>\n",
        "\t\t\t</h3>\n",
        "\n",
        "\t\t</div>\n",
        "\t\t\t</div>\n",
        "\t\t\t \n",
        "\t\t  </div>\n",
        "\t\t</div>\n",
        "\t\t</div>\n",
        "    </div>\n",
        "</body>\n",
        "\n",
        "<footer>\n",
        "    <script src=\"{{ url_for('static', filename='js/main.js') }}\"
type=\"text/javascript\"></script>    \n",
        "</footer>\n",
        "</html>"
      ],
      "metadata": {
        "id": "AkTNU9_FbnIw"
      },
      "execution_count": null,
      "outputs": []
    }
  ]
}
```


IBM
IBM-Project-16743-1659621316/
https://github.com/IBM-EPBL/IBM-Project-16743-1659621316/blob/main/PROJECT%20DEVELOPMENT%20PHASE/Delivery...
Click here to refres...
Gmail
YouTube
Maps
Dell

```
from tensorflow.keras import Sequential
```

```
In [4]:  model=Sequential()
```

```
In [5]:  model.add(Convolution2D(32,(3,3),input_shape = (128,128,3),activation = 'relu'))
```

```
In [6]:  model.add(MaxPooling2D(pool_size = (2,2)))
```

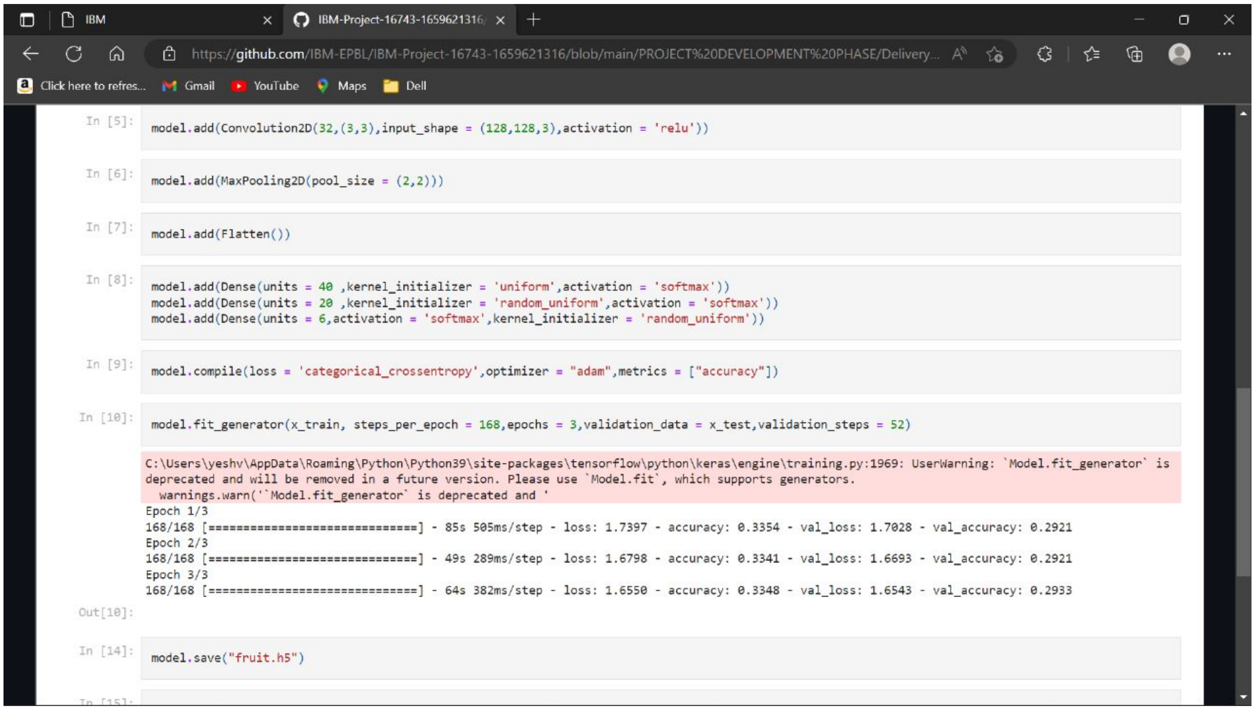
```
In [7]:  model.add(Flatten())
```

```
In [8]:  model.add(Dense(units = 300 ,kernel_initializer = 'uniform',activation = 'softmax'))
         model.add(Dense(units = 150 ,kernel_initializer = 'random_uniform',activation = 'softmax'))
         model.add(Dense(units = 75 ,kernel_initializer = 'uniform' ,activation = 'softmax'))
         model.add(Dense(units = 9,activation = 'softmax',kernel_initializer = 'random_uniform'))
```

```
In [9]:  model.compile(loss = 'categorical_crossentropy',optimizer = "adam",metrics = ["accuracy"])
```

```
In [10]: model.fit_generator(x_train, steps_per_epoch = 10,epochs = 50,validation_data = x_test,validation_steps = 27)
```

```
C:\Users\yeshv\AppData\Roaming\Python\Python39\site-packages\tensorflow\python\keras\engine\training.py:1969: 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 1/50
10/10 [==============================] - 10s 951ms/step - loss: 2.1955 - accuracy: 0.0855 - val_loss: 2.1932 - val_accuracy: 0.1759
Epoch 2/50
10/10 [==============================] - 8s 828ms/step - loss: 2.1897 - accuracy: 0.1562 - val_loss: 2.1889 - val_accuracy: 0.2269
Epoch 3/50
10/10 [==============================] - 8s 797ms/step - loss: 2.1868 - accuracy: 0.1937 - val_loss: 2.1846 - val_accuracy: 0.1829
```


IBM
IBM-Project-16743-1659621316/
https://github.com/IBM-EPBL/IBM-Project-16743-1659621316/blob/main/PROJECT%20DEVELOPMENT%20PHASE/Delivery...
Click here to refres...
Gmail
YouTube
Maps
Dell

```
In [5]:  model.add(Convolution2D(32,(3,3),input_shape = (128,128,3),activation = 'relu'))
```

```
In [6]:  model.add(MaxPooling2D(pool_size = (2,2)))
```

```
In [7]:  model.add(Flatten())
```

```
In [8]:  model.add(Dense(units = 40 ,kernel_initializer = 'uniform',activation = 'softmax'))
         model.add(Dense(units = 20 ,kernel_initializer = 'random_uniform',activation = 'softmax'))
         model.add(Dense(units = 6,activation = 'softmax',kernel_initializer = 'random_uniform'))
```

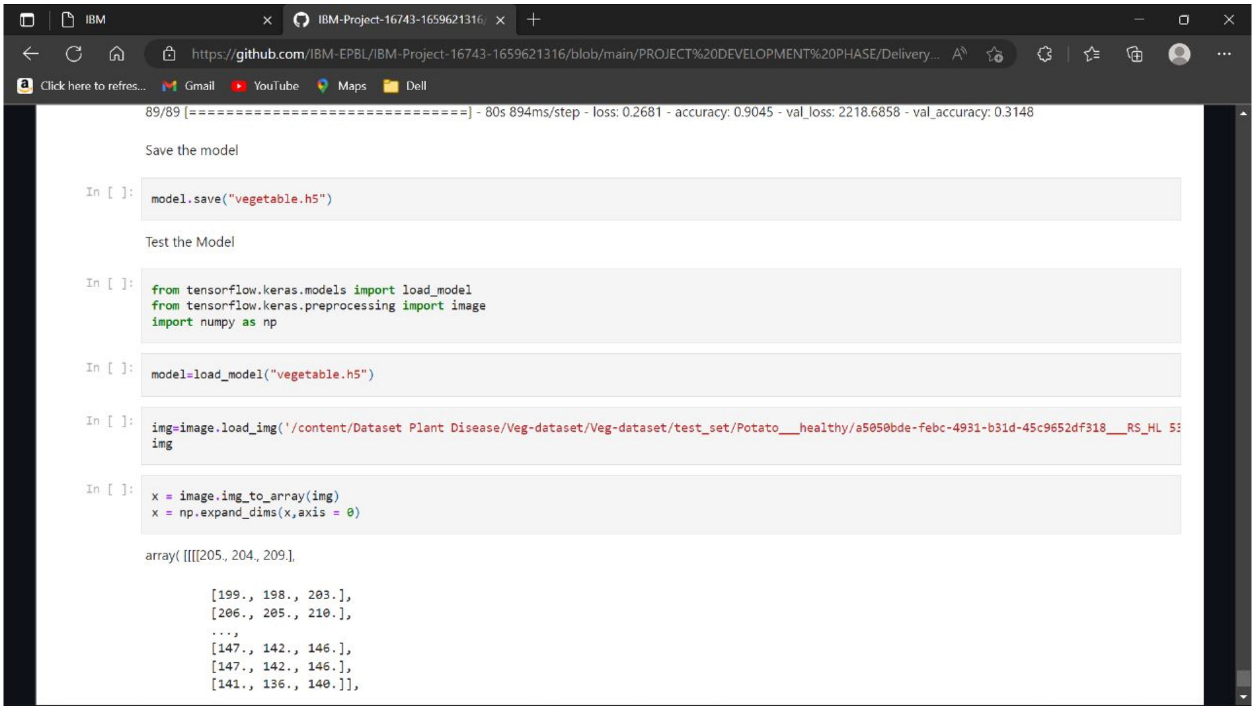
```
In [9]:  model.compile(loss = 'categorical_crossentropy',optimizer = "adam",metrics = ["accuracy"])
```

```
In [10]: model.fit_generator(x_train, steps_per_epoch = 168,epochs = 3,validation_data = x_test,validation_steps = 52)
```

```
C:\Users\yeshv\AppData\Roaming\Python\Python39\site-packages\tensorflow\python\keras\engine\training.py:1969: 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 1/3
168/168 [==============================] - 85s 505ms/step - loss: 1.7397 - accuracy: 0.3354 - val_loss: 1.7028 - val_accuracy: 0.2921
Epoch 2/3
168/168 [==============================] - 49s 289ms/step - loss: 1.6798 - accuracy: 0.3341 - val_loss: 1.6693 - val_accuracy: 0.2921
Epoch 3/3
168/168 [==============================] - 64s 382ms/step - loss: 1.6550 - accuracy: 0.3348 - val_loss: 1.6543 - val_accuracy: 0.2933
Out[10]:
```

```
In [14]: model.save("fruit.h5")
```

```
In [15]:
```


IBM
IBM-Project-16743-1659621316/
https://github.com/IBM-EPBL/IBM-Project-16743-1659621316/blob/main/PROJECT%20DEVELOPMENT%20PHASE/Delivery...
Click here to refres...
Gmail
YouTube
Maps
Dell

```
89/89 [==============================] - 80s 894ms/step - loss: 0.2681 - accuracy: 0.9045 - val_loss: 2218.6858 - val_accuracy: 0.3148
```

Save the model

```
In [ ]: model.save("vegetable.h5")
```

Test the Model

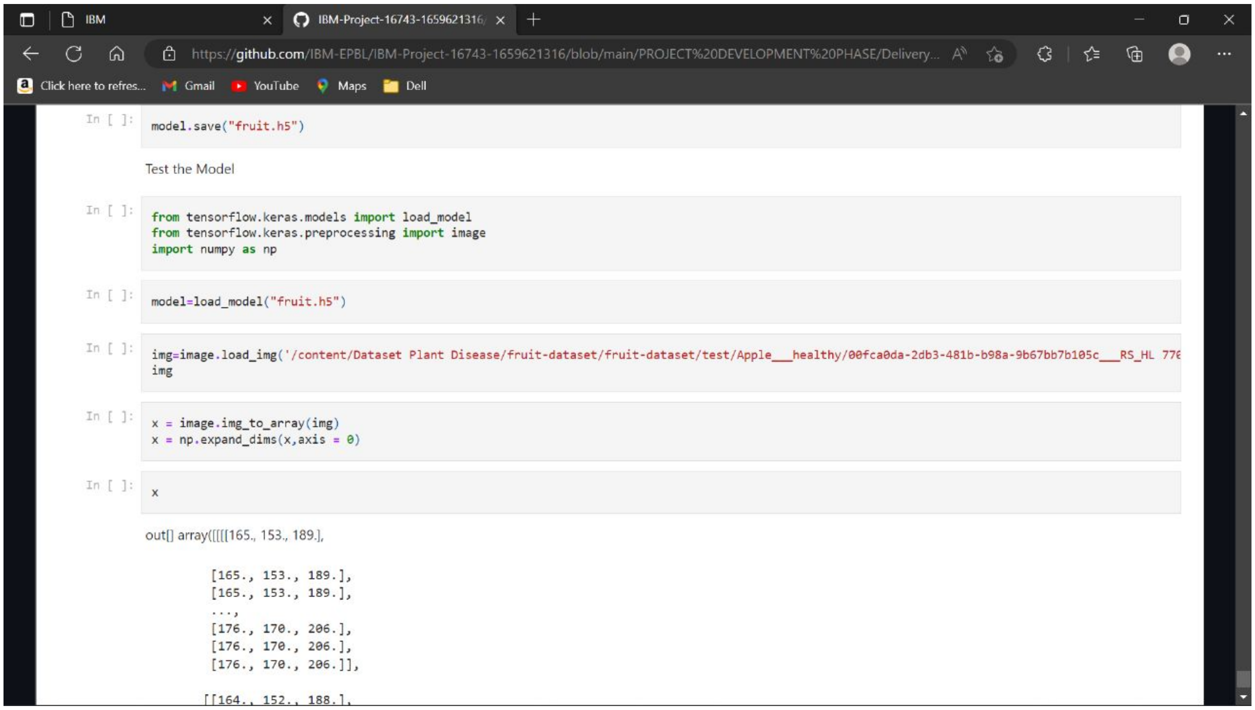
```
In [ ]: from tensorflow.keras.models import load_model
        from tensorflow.keras.preprocessing import image
        import numpy as np
```

```
In [ ]: model=load_model("vegetable.h5")
```

```
In [ ]: img=image.load_img('/content/Dataset Plant Disease/Veg-dataset/Veg-dataset/test_set/Potato___healthy/a5050bde-febc-4931-b31d-45c9652df318___RS_HL 53
        img
```

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

```
array( [[[[205., 204., 209.],

         [199., 198., 203.],
         [206., 205., 210.],
         ...,
         [147., 142., 146.],
         [147., 142., 146.],
         [141., 136., 140.]],
```


IBM
IBM-Project-16743-1659621316/
https://github.com/IBM-EPBL/IBM-Project-16743-1659621316/blob/main/PROJECT%20DEVELOPMENT%20PHASE/Delivery...
Click here to refres...
Gmail
YouTube
Maps
Dell

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

Test the Model

```
In [ ]: from tensorflow.keras.models import load_model
        from tensorflow.keras.preprocessing import image
        import numpy as np
```

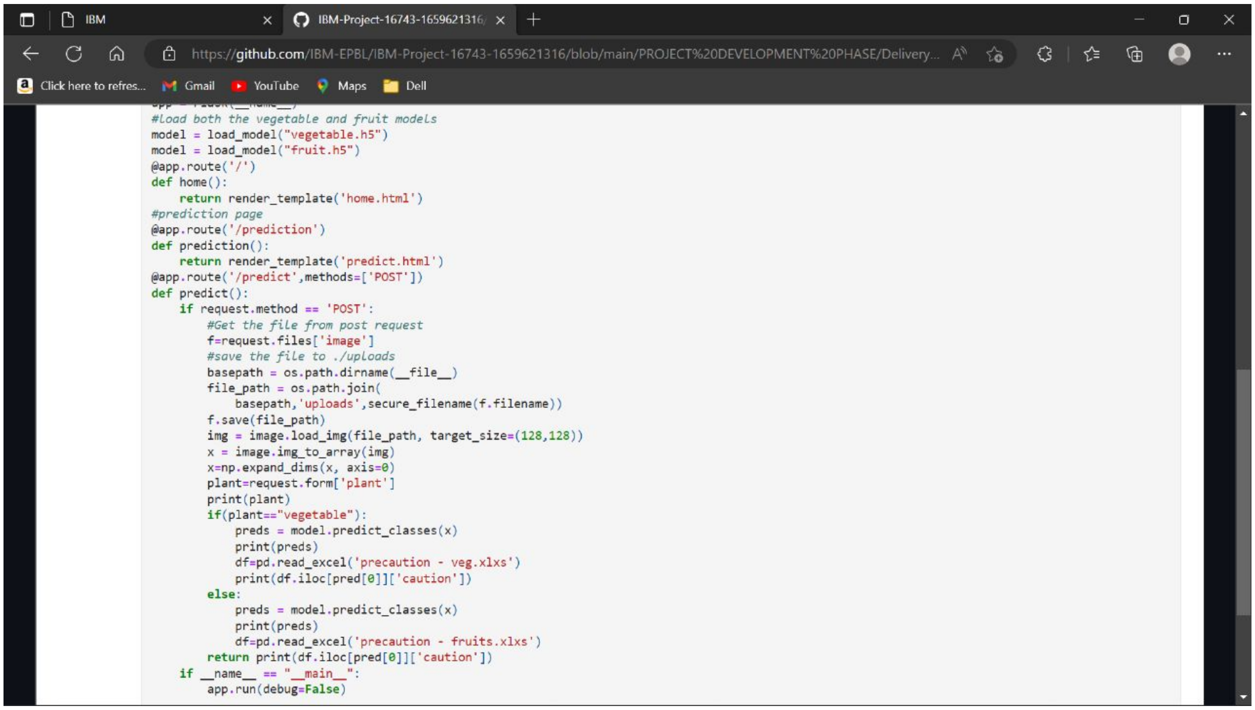
```
In [ ]: model=load_model("fruit.h5")
```

```
In [ ]: img=image.load_img('/content/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___healthy/00fca0da-2db3-481b-b98a-9b67bb7b105c___RS_HL 776
        img
```

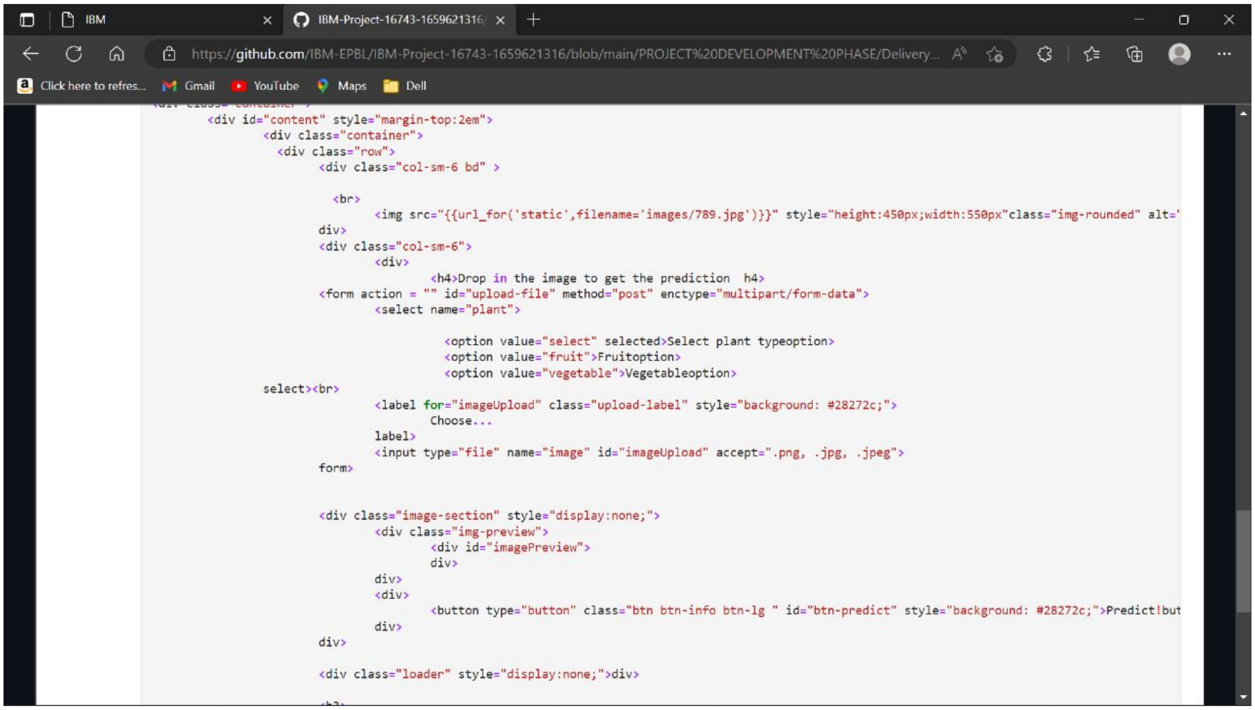
```
In [ ]: x = image.img_to_array(img)
        x = np.expand_dims(x,axis = 0)
```

```
In [ ]: x
```

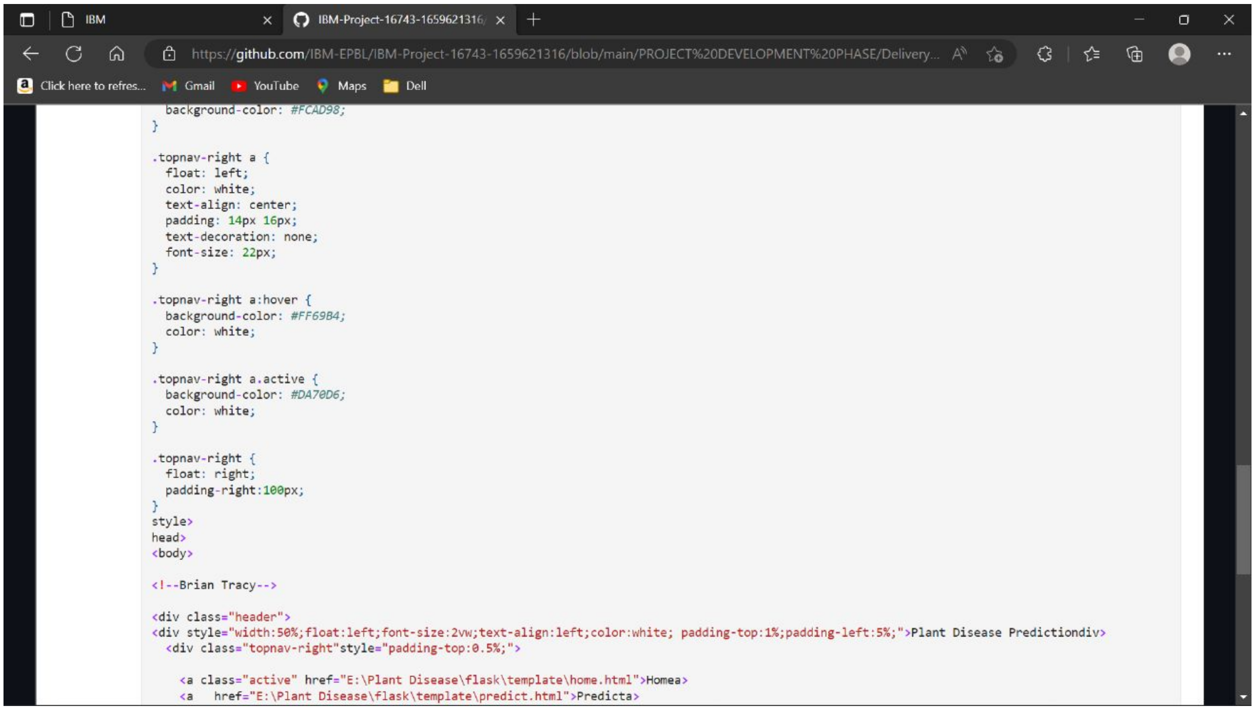
```
out[] array([[[[165., 153., 189.],

         [165., 153., 189.],
         [165., 153., 189.],
         ...,
         [176., 170., 206.],
         [176., 170., 206.],
         [176., 170., 206.]],

        [[164., 152., 188.],
```


IBM
IBM-Project-16743-1659621316/
https://github.com/IBM-EPBL/IBM-Project-16743-1659621316/blob/main/PROJECT%20DEVELOPMENT%20PHASE/Delivery...
Click here to refres...
Gmail
YouTube
Maps
Dell

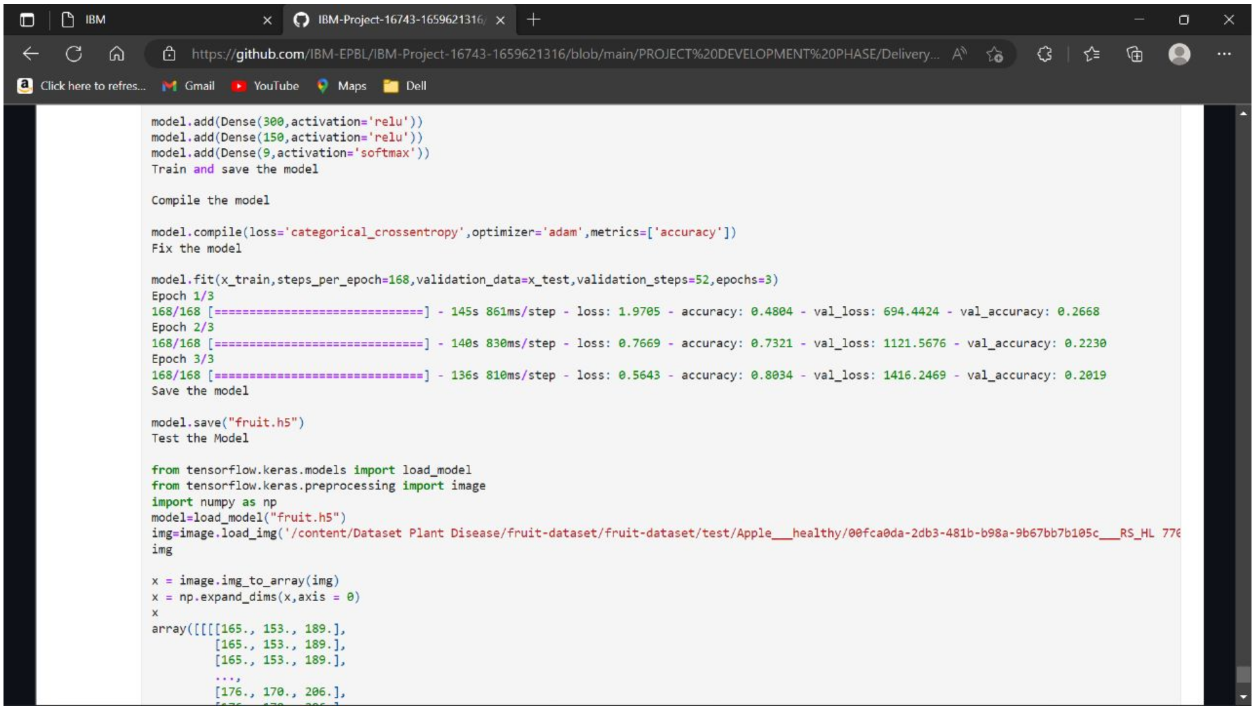
```python
#Load both the vegetable and fruit models
model = load_model("vegetable.h5")
model = load_model("fruit.h5")
@app.route('/')
def home():
    return render_template('home.html')
#prediction page
@app.route('/prediction')
def prediction():
    return render_template('predict.html')
@app.route('/predict',methods=['POST'])
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_classes(x)
            print(preds)
            df=pd.read_excel('precaution - veg.xlxs')
            print(df.iloc[pred[0]]['caution'])
        else:
            preds = model.predict_classes(x)
            print(preds)
            df=pd.read_excel('precaution - fruits.xlxs')
        return print(df.iloc[pred[0]]['caution'])
    if __name__ == "__main__":
        app.run(debug=False)
```


IBM
IBM-Project-16743-1659621316/
https://github.com/IBM-EPBL/IBM-Project-16743-1659621316/blob/main/PROJECT%20DEVELOPMENT%20PHASE/Delivery...
Click here to refres...
Gmail
YouTube
Maps
Dell

```html
        <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/789.jpg')}}" style="height:450px;width:550px"class="img-rounded" alt='
                        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 typeoption>
                                          <option value="fruit">Fruitoption>
                                          <option value="vegetable">Vegetableoption>
                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!but
                                div>
                        div>

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


IBM
IBM-Project-16743-1659621316/
https://github.com/IBM-EPBL/IBM-Project-16743-1659621316/blob/main/PROJECT%20DEVELOPMENT%20PHASE/Delivery...
Click here to refres...
Gmail
YouTube
Maps
Dell

```
  background-color: #FCAD98;
}

.topnav-right a {
  float: left;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 22px;
}

.topnav-right a:hover {
  background-color: #FF69B4;
  color: white;
}

.topnav-right a.active {
  background-color: #DA70D6;
  color: white;
}

.topnav-right {
  float: right;
  padding-right:100px;
}
style>
head>
<body>

<!--Brian Tracy-->

<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-align:left;color:white; padding-top:1%;padding-left:5%;">Plant Disease Predictiondiv>
  <div class="topnav-right"style="padding-top:0.5%;">

    <a class="active" href="E:\Plant Disease\flask\template\home.html">Homea>
    <a   href="E:\Plant Disease\flask\template\predict.html">Predicta>
```


IBM
IBM-Project-16743-1659621316/
https://github.com/IBM-EPBL/IBM-Project-16743-1659621316/blob/main/PROJECT%20DEVELOPMENT%20PHASE/Delivery...
Click here to refres...
Gmail
YouTube
Maps
Dell

```
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))
Train and save the model

Compile the model

model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
Fix the model

model.fit(x_train,steps_per_epoch=168,validation_data=x_test,validation_steps=52,epochs=3)
Epoch 1/3
168/168 [==============================] - 145s 861ms/step - loss: 1.9705 - accuracy: 0.4804 - val_loss: 694.4424 - val_accuracy: 0.2668
Epoch 2/3
168/168 [==============================] - 140s 830ms/step - loss: 0.7669 - accuracy: 0.7321 - val_loss: 1121.5676 - val_accuracy: 0.2230
Epoch 3/3
168/168 [==============================] - 136s 810ms/step - loss: 0.5643 - accuracy: 0.8034 - val_loss: 1416.2469 - val_accuracy: 0.2019
Save the model

model.save("fruit.h5")
Test the Model

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
model=load_model("fruit.h5")
img=image.load_img('/content/Dataset Plant Disease/fruit-dataset/fruit-dataset/test/Apple___healthy/00fca0da-2db3-481b-b98a-9b67bb7b105c___RS_HL 776
img

x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
x
array([[[[165., 153., 189.],
         [165., 153., 189.],
         [165., 153., 189.],
         ...,
         [176., 170., 206.],
```

GITHUB AND PROJECT DEMO LINK

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-16743-1659621316

PROJECT DEMO LINK:

https://drive.google.com/drive/folders/174zHfXGuyKZndKktAyz5Kfrm2gT6_eEp