IBM

NALAIYA THIRAN

PROJECT REPORT

ON

AI-POWERED NUTRITION ANALYZER FOR FITNESS ENTHUSIASTS

TEAM ID: PNT2022TMID03893

SRI SAI RAM ENGINEERING COLLEGE

Team Members

SRIDHAR M

DINESH ROHIT K B

RAVINDRAN E

SURYA P

ANANTHA KIRISHNAN S

1. INTRODUCTION

1.1 Project Overview

Food is a necessity for human life and has been addressed in numerous medical conventions. Modern dietary evaluation and nutrition analysis technologies give consumers more possibilities to explore nutrition patterns, comprehend their daily eating habits, and keep up a balanced diet. Finding out a food's nutritional value is done through nutritional analysis. Information about the chemical make-up, processing, quality assurance, and contamination of food is a crucial component of analytical chemistry.

1.2 Purpose

The primary goal of the project is to develop a model that will be used to categorise fruits according to their various attributes, such as color, shape, and texture. Here, users can take pictures of various fruits, which are subsequently uploaded to a trained algorithm for analysis. The model examines the image to determine the nutrition content of the fruits, such as their sugar, fiber, protein, and calorie content.

2. LITERATURE SURVEY

2.1 Existing Problem

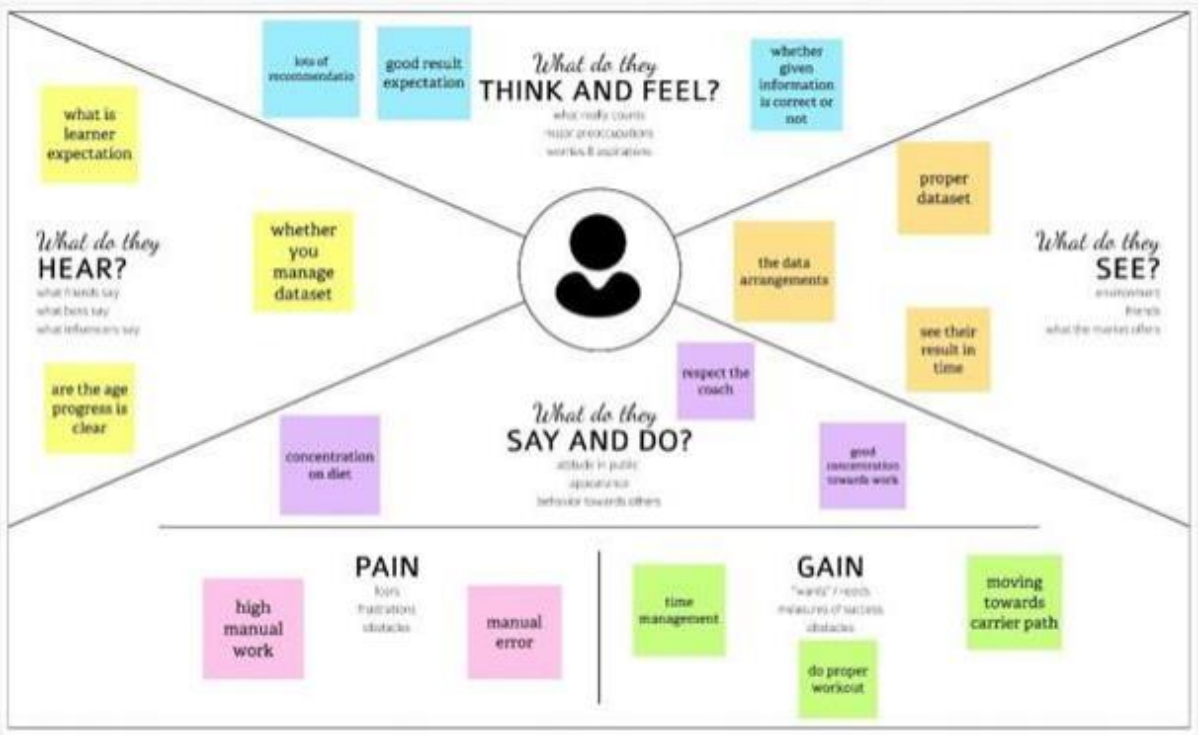
Neutrino offers its users nutrition-based data services and analytics with the goal of becoming a leading platform for the nutrition industry. To enable customized data compilation, the platform uses NLP, mathematical models from optimization theory, and predictive analysis. The software uses artificial intelligence to generate personalized data for a smart calorie counter. Their artificial intelligence picks up on a person's preferences, body type, and taste. An extensive nutrition and exercise tracker has all of this information.

2.2 Problem Statement Definition

The primary goal of the project is to develop a model that will be used to categorise fruits according to their various attributes, such as color, shape, and texture. Here, users can take pictures of various fruits, which are subsequently uploaded to a trained algorithm for analysis. The model examines the image to determine the nutrition content of the fruits, such as their sugar, fiber, protein, and calorie content.

3. IDEATION & PROPOSED SOLUTION

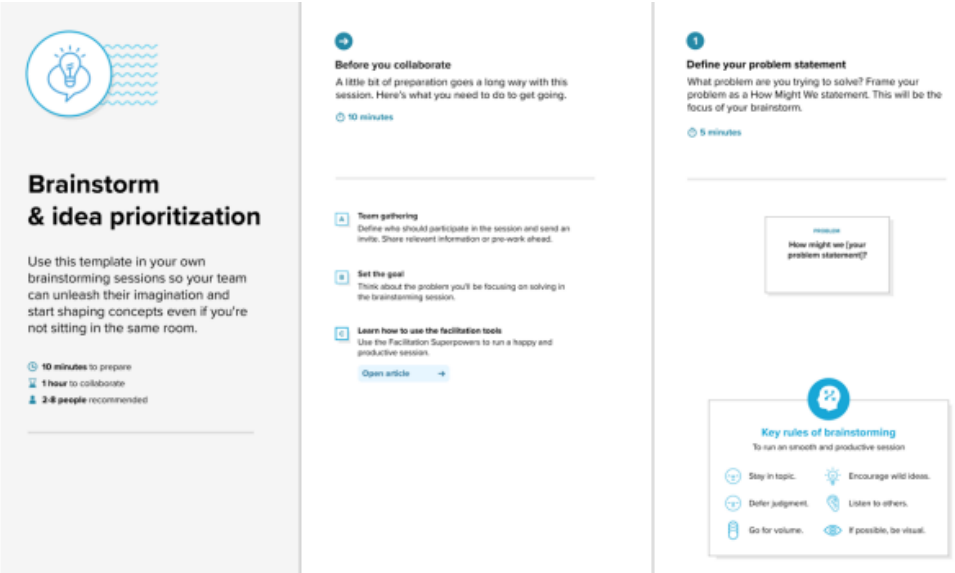
3.1 Empathy Map Canvas



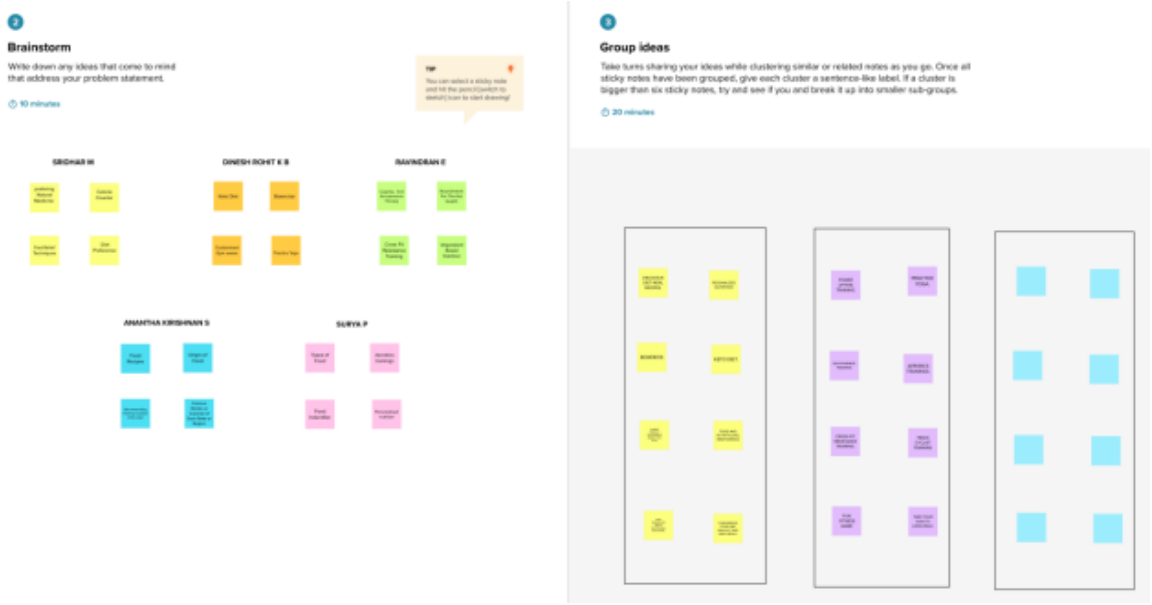
3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

3.3 Proposed Solution

S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	• The main aim of the project is to build a model which is used for identifying the fruit depends on the different characteristics like color, shape, texture etc using image processing. • Here the user can capture the images of different fruits and then the image will be analysed with the trained model. • The model analyses the image and lists out the nutrients present in the fruitlike sugar, vitamins, minerals, protein etc
2.	Idea/Solution description	• The idea of this application is that the user can capture the images of different fruits and vegetables, and then the image will be sent to the trained model. • The model analyses the image and detects the nutrition based on the fruits like (Sugar, Fiber, Protein, Calorie intake, etc.). The above idea is achieved by using the Convolution Neural Network (CNN). • It is used to pick the raw pixels present in the image. Fruit Recognition using Color and Texture Features
3.	Novelty/Uniqueness	• The application has several unique features. The main feature is that the user need not have to visit or consult a Nutritionist(or) a Dietician to follow a fit and healthy diet. • This application has the feature of analysing the entire nutritional content of fruits and vegetables by simply scanning them. It provides for a personalized dietary requirement for individuals who have limited preferences while choosing food
4.	Social Impact / Customer Satisfaction	• This will acquire knowledge and provide information about nutrition. Now a days, no one follows the diet plan. Providing this information, they come to know about the nutrition present in each food item.

		• It is used to schedule a diet plan by taking the image of a food item and if we send it, we can get information about each food
5.	Business Model (Revenue Model)	• Social media is the best way to spread the word about our application and with the help of influencers we can attract normal people. • Clustering and targeting the fitness people with the help of local gyms. Allowing third-party vendors (Nutritional Products) to sell their products through our app via.
6.	Scalability of the Solution	• Artificial intelligence (AI) can be used to predict investment outcomes quickly and effectively, as well as to devise strategies or establish long-term goals. Scalable AI pertains to how data models, infrastructures, and algorithms can increase or decrease their complexity, speed, or size at scale in order to best handle the requirements of the situation at hand. • As improvements continue with data storage capacities as well as computing resources, AI models can be created with billions of parameters. Scaling up nutrition is a global push for action and investment to improve maternal, child nutrition and various health problems.

3.4 Problem Solution Fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators.

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

Following are the functional requirements of the proposed solution

FR.NO	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail or using phone number
FR-2	User Confirmation	Confirmation on Gmail or using phone number
FR-3	User Details	Registration through form
FR-4	Server Calculation	Calculating user details Example: Height, Weight, Age
FR-5	Calculate information from Server	Based on the given information they calculate their nutrition leve
FR-6	Server notification	Based on the nutrition level server provide notification like intake of food and water

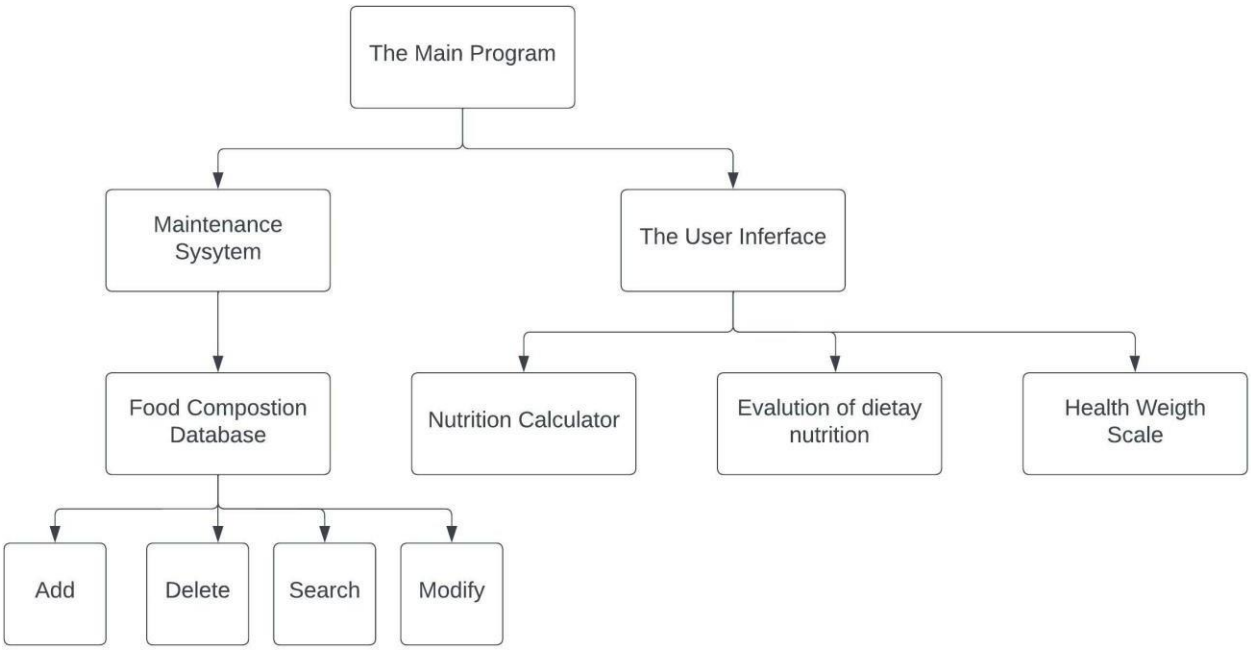
4.2 Non-Functional Requirements

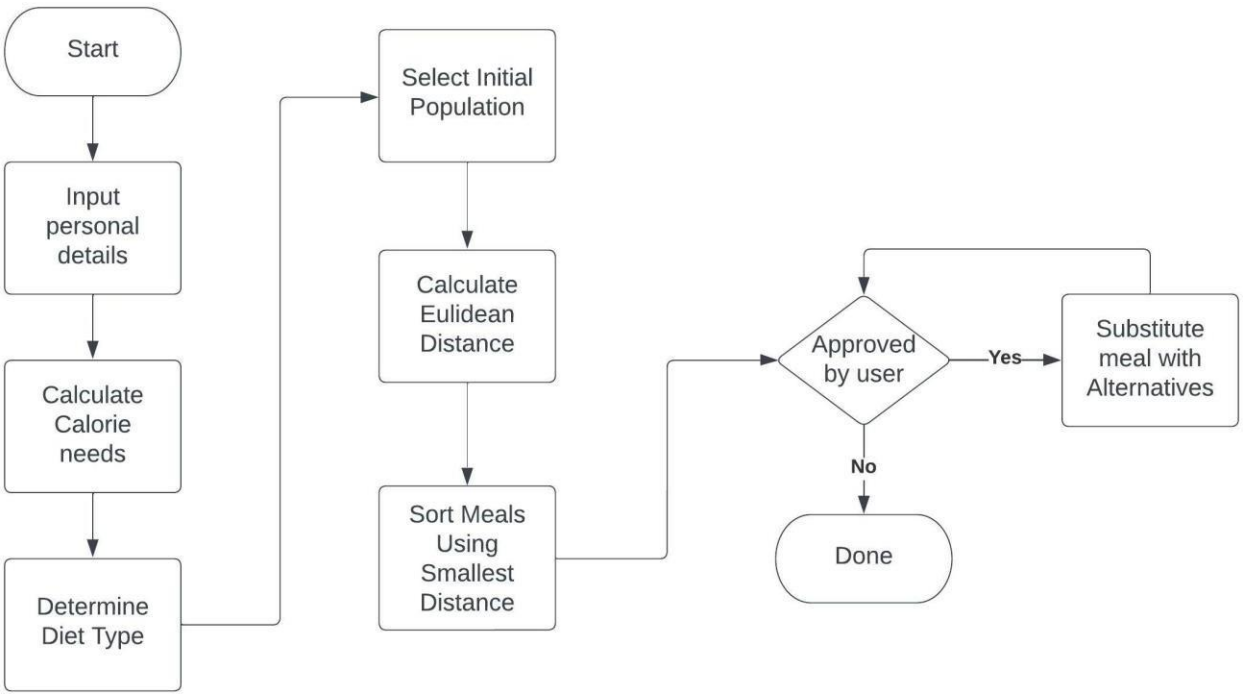
Following are the non-functional requirements of the proposed solution.

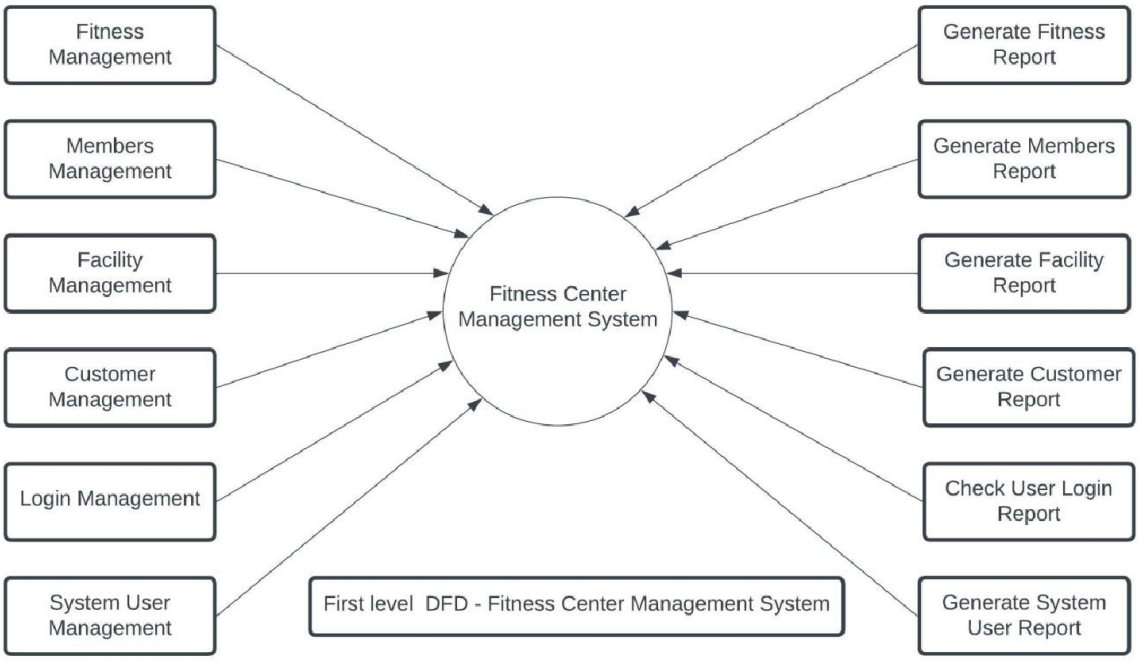
FR.NO.	Non-Functional Requirement	Description
NFR-1	Usability	To maintain your health
NFR-2	Security	User details will be secure from server side
NFR-3	Reliability	Trusted details from server
NFR-4	Performance	Better performance comparing to other apps
NFR-5	Availability	Available on email and chatbot
NFR-6	Scalability	Every Customer must get Healthy Life and Proper Diet Maintenance based on the Healthy Measure and Calorie prediction.

5. PROJECT DESIGN

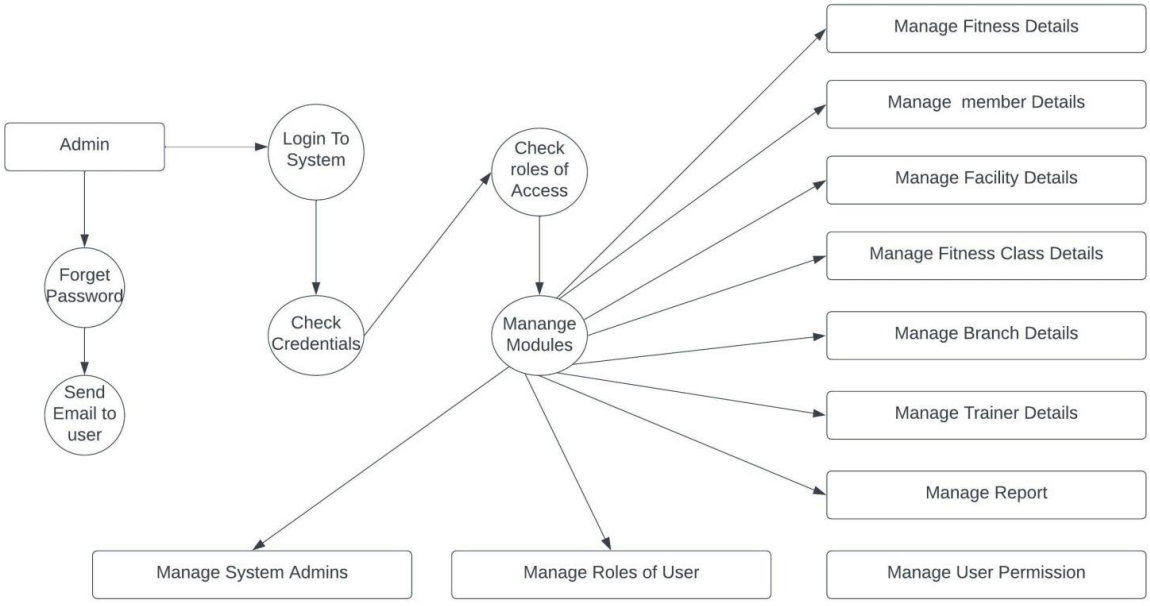
5.1 Data Flow Diagrams





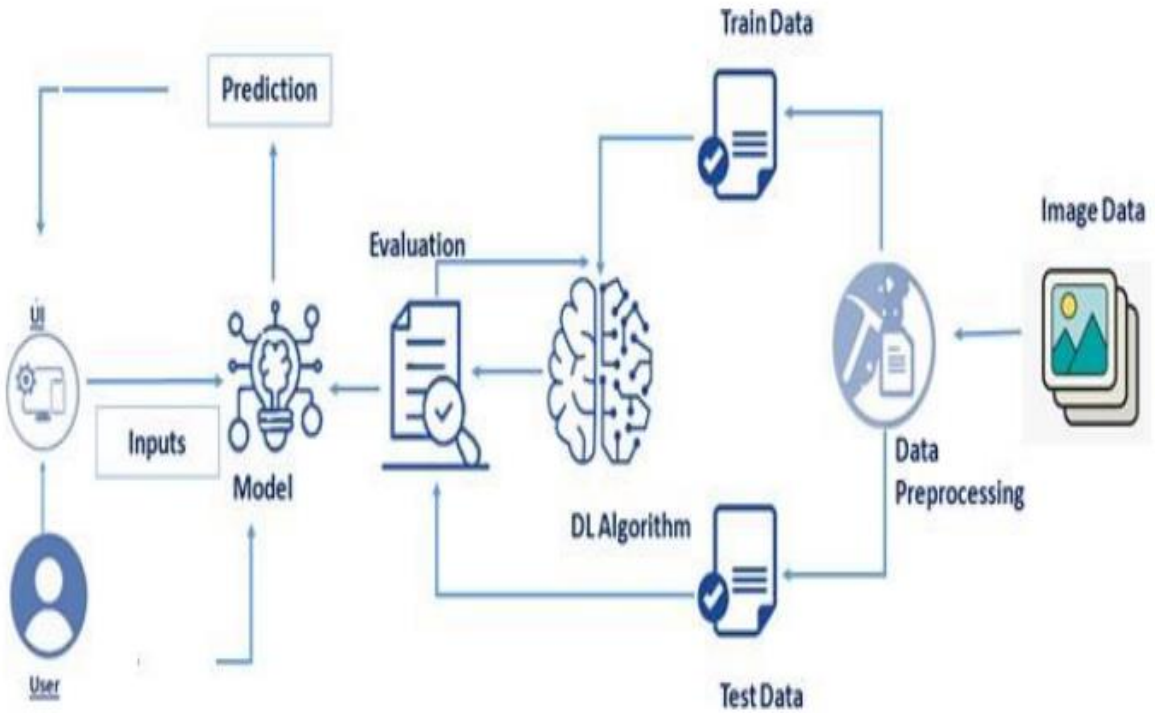


First level DFD - Fitness Center Management System



Second Level DFD - Fitness Center Management System

5.2 Solution & Technical Architecture



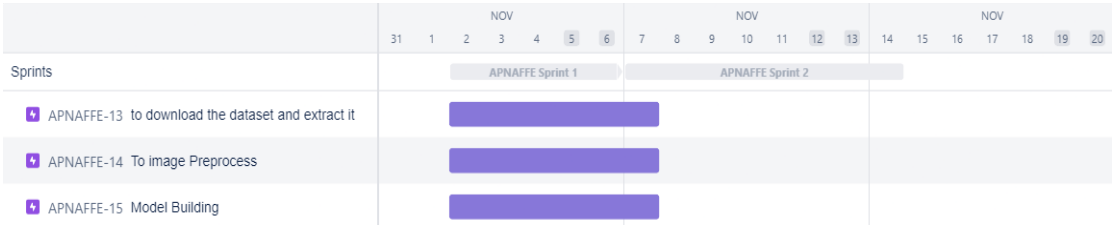
6. PROJECT PLANNING & SCEHEDULING

6.1 Sprint Planning & Estimation

Every project manager should consider the delivery strategy of the project deliverables as a strategic component. Every project's objective is to deliver a product that fulfils a certain need. The word "purpose" can be used to refer to a wide range of objectives, including those for a chair, a building, a translation, etc. Delivery planning is one of the activities used in Project Spirit to finish the project and display the projected timeline. This delivery plan aids in comprehending the team members' workflow and project procedure. Each individual module is given to a team member so they can showcase their efforts and contributions to the project's development.

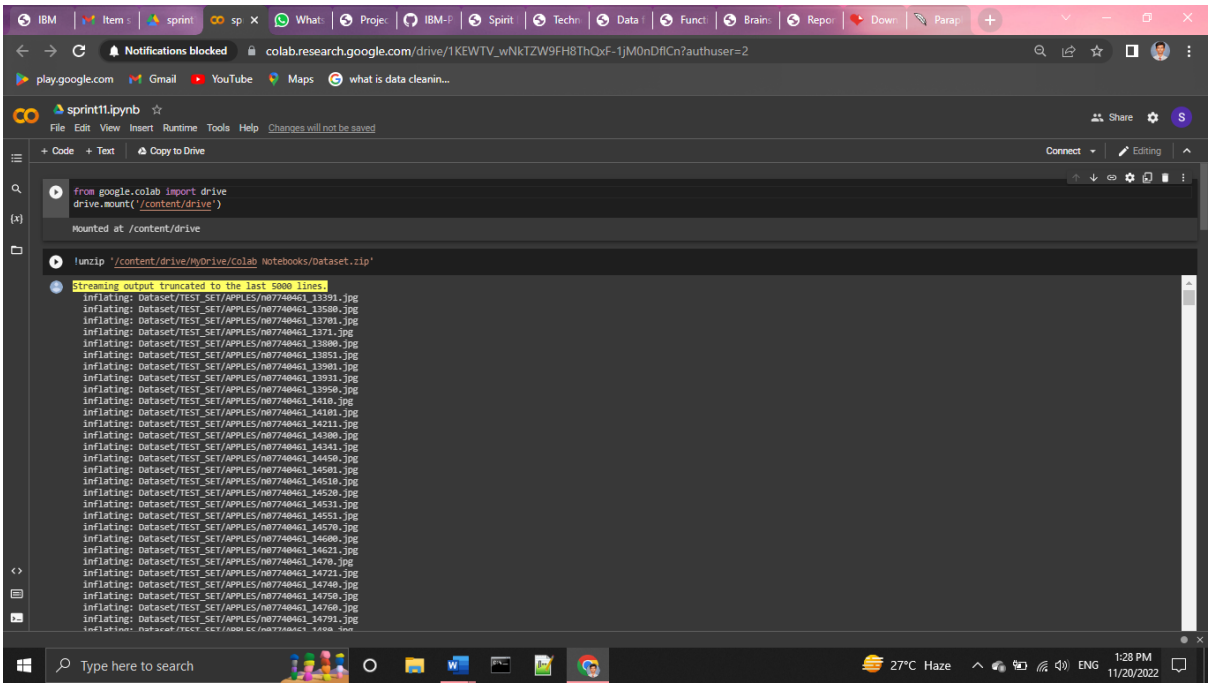
6.2 Sprint Delivery Schedule

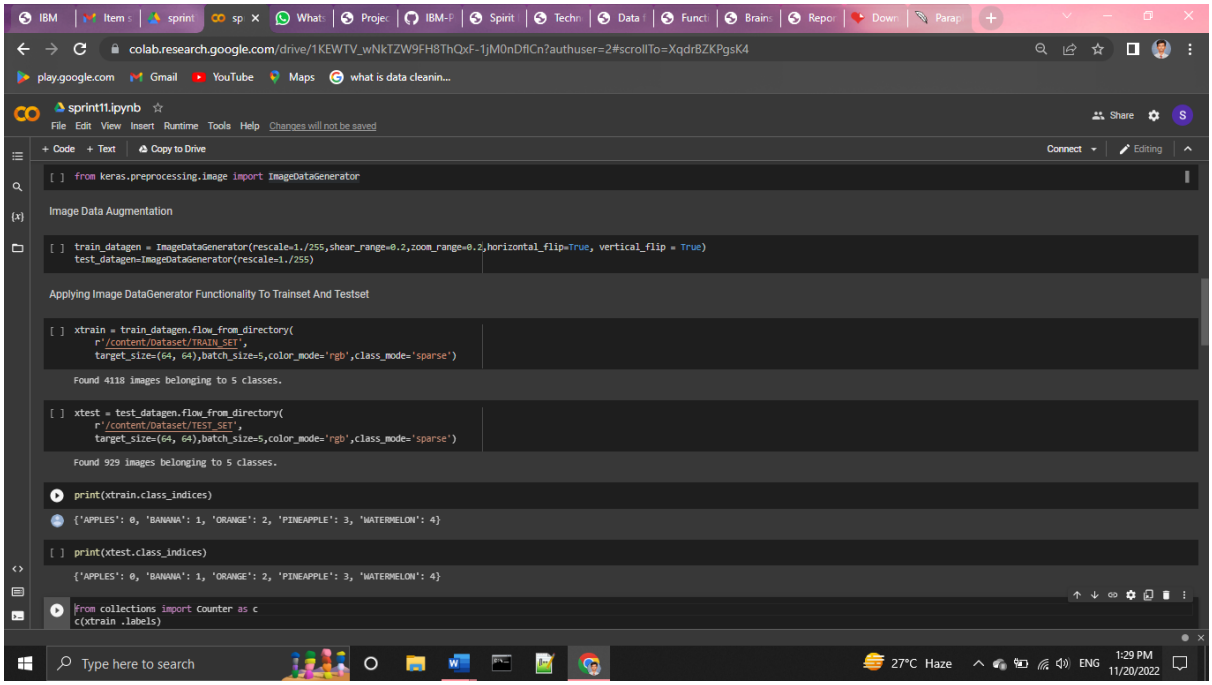
6.3 Report from JIRA

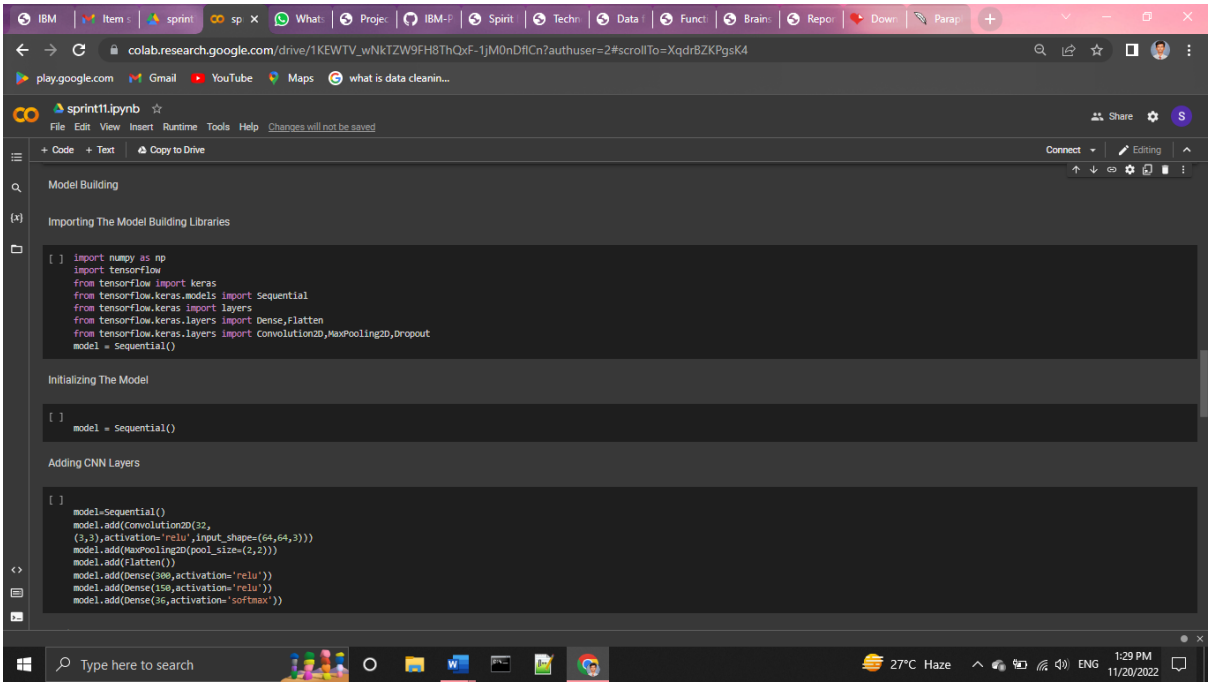


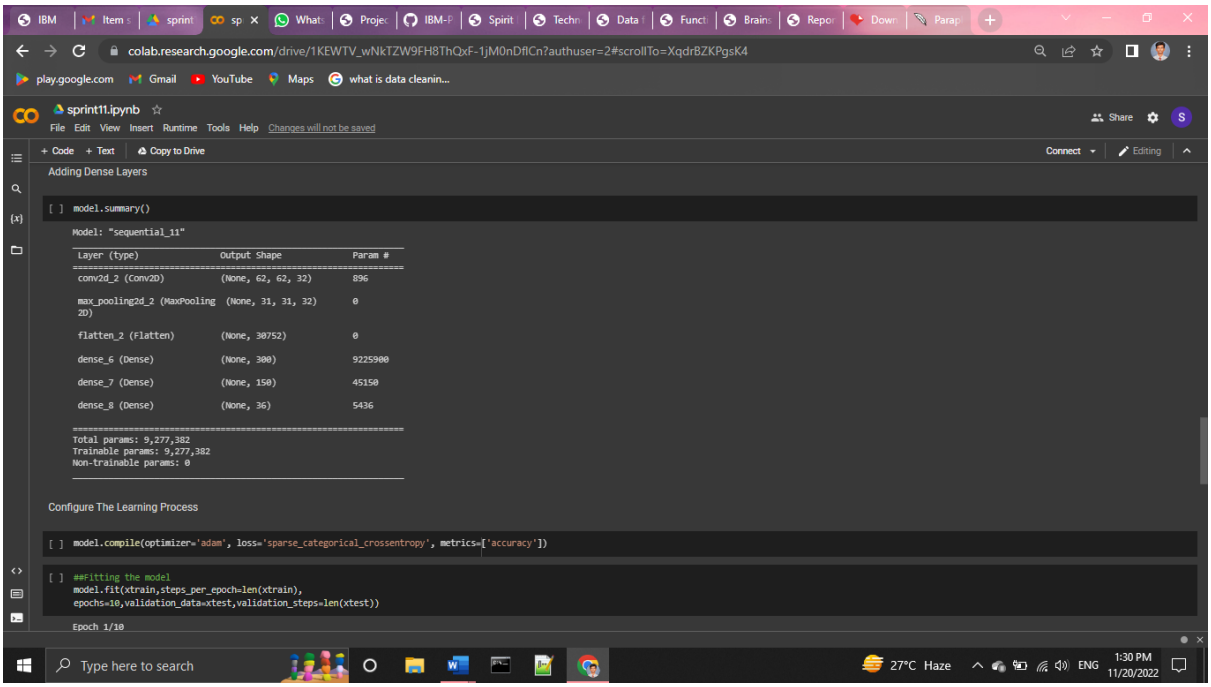
7. CODING & SOLUTION (Explain the features added in the project along with code)

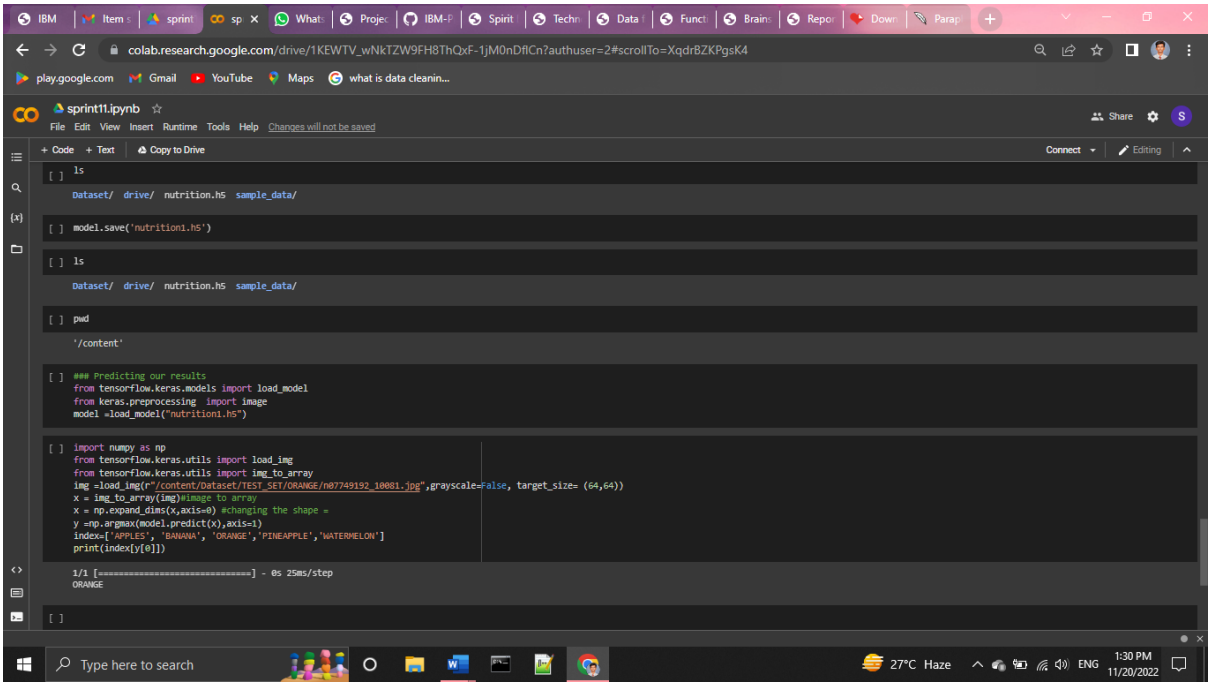
7.1 Feature 1



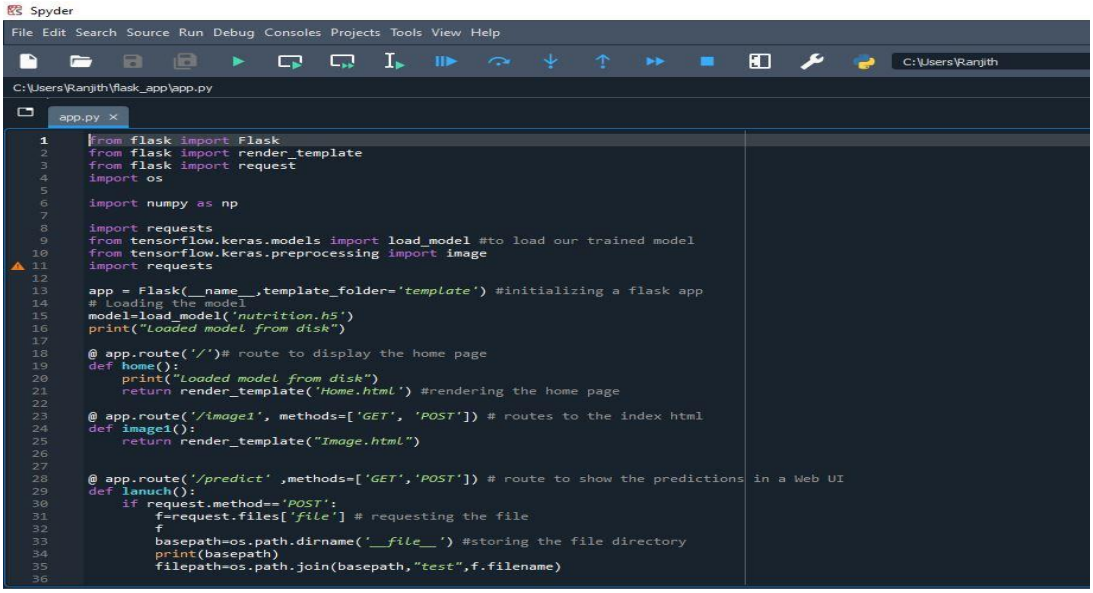



colab.research.google.com/drive/1KEWTV_wNkTZW9FH8ThQxF-1jM0nDflCn?authuser=2#scrollTo=XqdrBZKPgsK4
sprint11.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing
Share
Model Building
Importing The Model Building Libraries
[ ] import numpy as np
import tensorflow
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import Dense,Flatten
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Dropout
model = Sequential()
Initializing The Model
[ ]
model = Sequential()
Adding CNN Layers
[ ]
model=Sequential()
model.add(Convolution2D(32,
(3,3),activation='relu',input_shape=(64,64,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(36,activation='softmax'))
Type here to search
27°C Haze
ENG
1:29 PM
11/20/2022


colab.research.google.com/drive/1KEWTV_wNkTZW9FH8ThQxF-1jM0nDflCn?authuser=2#scrollTo=XqdrBZKPgsK4
sprint11.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing
Share
Adding Dense Layers
[ ] model.summary()
Model: "sequential_11"
Layer (type) Output Shape Param #
conv2d_2 (Conv2D) (None, 62, 62, 32) 896
max_pooling2d_2 (MaxPooling 2D) (None, 31, 31, 32) 0
flatten_2 (Flatten) (None, 30752) 0
dense_6 (Dense) (None, 300) 9225900
dense_7 (Dense) (None, 150) 45150
dense_8 (Dense) (None, 36) 5436
Total params: 9,277,382
Trainable params: 9,277,382
Non-trainable params: 0
Configure The Learning Process
[ ] model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
[ ] ##Fitting the model
model.fit(xtrain,steps_per_epoch=len(xtrain),
epochs=10,validation_data=xtest,validation_steps=len(xtest))
Epoch 1/10
Type here to search
27°C Haze
ENG
1:30 PM
11/20/2022


colab.research.google.com/drive/1KEWTV_wNkTZW9FH8ThQxF-1jM0nDflCn?authuser=2#scrollTo=XqdrBZKPgsK4
sprint1.ipynb
File Edit View Insert Runtime Tools Help Changes will not be saved
+ Code + Text Copy to Drive
Connect Editing
ls
Dataset/ drive/ nutrition.h5 sample_data/
model.save('nutrition1.h5')
ls
Dataset/ drive/ nutrition.h5 sample_data/
pwd
'/content'
Predicting our results
from tensorflow.keras.models import load_model
from keras.preprocessing import image
model =load_model("nutrition1.h5")
import numpy as np
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array
img =load_img(r"/content/Dataset/TEST_SET/ORANGE/n07749192_10081.jpg",grayscale=False, target_size= (64,64))
x = img_to_array(img)#image to array
x = np.expand_dims(x,axis=0) #changing the shape =
y =np.argmax(model.predict(x),axis=1)
index=['APPLES', 'BANANA', 'ORANGE','PINEAPPLE','WATERMELON']
print(index[y[0]])
1/1 [==============================] - 0s 25ms/step
ORANGE
Type here to search
27°C Haze
ENG
1:30 PM
11/20/2022

7.2 Feature 2


Spyder
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Ranjith
C:\Users\Ranjith\flask_app\app.py
app.py
from flask import Flask
from flask import render_template
from flask import request
import os
import numpy as np
import requests
from tensorflow.keras.models import load_model #to load our trained model
from tensorflow.keras.preprocessing import image
import requests
app = Flask(__name__,template_folder='template') #initializing a flask app
# Loading the model
model=load_model('nutrition.h5')
print("Loaded model from disk")
@ app.route('/')# route to display the home page
def home():
print("Loaded model from disk")
return render_template('Home.html') #rendering the home page
@ app.route('/image1', methods=['GET', 'POST']) # routes to the index html
def image1():
return render_template("Image.html")
@ app.route('/predict' ,methods=['GET','POST']) # route to show the predictions in a Web UI
def lanuch():
if request.method=='POST':
f=request.files['file'] # requesting the file
f
basepath=os.path.dirname('__file__') #storing the file directory
print(basepath)
filepath=os.path.join(basepath,"test",f.filename)

Spyder

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Ranjith

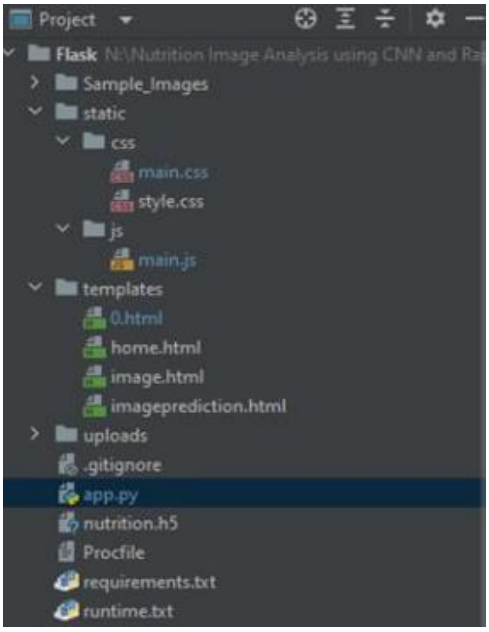
C:\Users\Ranjith\flask_app\app.py

app.py

```
@ app.route('/image1', methods=['GET', 'POST']) # routes to the index html
def image1():
    return render_template("Image.html")


@ app.route('/predict' ,methods=['GET','POST']) # route to show the predictions in a Web UI
def lanuch():
    if request.method=='POST':
        f=request.files['file'] # requesting the file
        f
        basepath=os.path.dirname('__file__') #storing the file directory
        print(basepath)
        filepath=os.path.join(basepath,"test",f.filename)

         #storing the file in uploads folder
        f.save(filepath) #saving the file

        img=image.load_img(filepath,target_size=(32,32)) #load and reshaping the image
        x=image.img_to_array(img) #converting image to an array
        x=np.expand_dims(x,axis=0) #changing the dimensions of the image

        pred=np.argmax(model.predict(x), axis=1)
        print("prediction",pred) #printing the prediction
        index=['APPLE','BANANA','ORANGE','BANANA','WATERMELON','WATERMELON','WATERMELON','APPLE','BANANA','WATERMELON']

        result=str(index[pred[0]])
        print(result)
        x=result
        result=nutrition(result)
        print(result)

        return render_template("0.html",showcase=(result),showcase1=(x))
def nutrition(index):

    import requests
```

8. TESTING

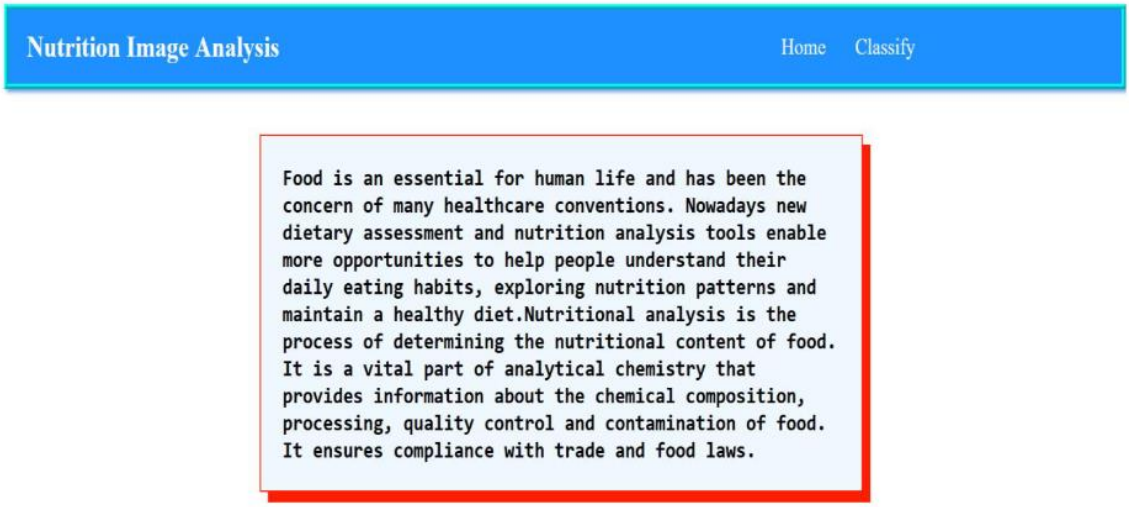
8.1 Test Cases



8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics



10. ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES

- Picture of body identifying benefits of healthy eating for adults.
    - May help you live longer.
    - Keeps skin, teeth, and eyes healthy.
    - Supports muscles
    - Strengthens bones.
    - Lowers risk of heart disease, type 2 diabetes, and some cancers.
    - Supports healthy pregnancies and breastfeeding.

11. CONCLUSION

By the end of this project we will,

- Know fundamental concepts and techniques of Convolutional Neural Network.
- Gain a broad understanding of image data.
- Know how to build a web application using the Flask framework.
- Know how to pre-process data.
- Know how to clean the data using different data pre-processing techniques.

12. FURURE SCOPE

- AI is revolutionizing the health industry.
- It is majorly used in improving marketing and sales decisions, AI is now also being used to reshape individual habits.
- In future we don’t want to go to gym and do any diets. By using this nutrition fitness analyser, we can maintain our diet plans without any help from others and we can lead a happy and healthy life with good wealth.
- AI can easily track health behaviors and repetitive exercise patterns and use the data to guide you towards your fitness journey and diet plans.

13. APPENDIX

Source Code & GitHub Code:

```
from flask import Flask,render_template,request
import os

import numpy as np

import requests
from tensorflow.keras.models import load_model #to load our trained
model
from tensorflow.keras.preprocessing import image
import requests

app = Flask(__name__,template_folder='template') #initializing a flask
app
# Loading the model
model=load_model('nutrition1.h5')
print("Loaded model from disk")

@ app.route('/')# route to display the home page
def home():
    print("Loaded model from disk")
    return render_template('home.html') #rendering the home page

@ app.route('/image1', methods=['GET', 'POST']) # routes to the index
html
def image1():
    return render_template("Image.html")



@ app.route('/predict' ,methods=['GET','POST']) # route to show the
predictions in a Web UI
def lanuch():
    if request.method=='POST':
        f=request.files['file'] # requesting the file
        f
        basepath=os.path.dirname('__file__') #storing the file directory
        print(basepath)
```

```
        filepath=os.path.join(basepath,"test",f.filename)

         #storing the file in uploads folder
        f.save(filepath) #saving the file

        img=image.load_img(filepath,target_size=(32,32)) #load and
reshaping the image
        x=image.img_to_array(img) #converting image to an array
        x=np.expand_dims(x,axis=0) #changing the dimensions of the
image

        pred=np.argmax(model.predict(x), axis=1)
        print("prediction",pred) #printing the prediction

index=['APPLE','BANANA','ORANGE','BANANA','WATERMELON','
WATERMELON','WATERMELON','APPLE','BANANA','WATERMEL
ON']

        result=str(index[pred[0]])
        print(result)
        x=result
        result=nutrition(result)
        print(result)

        return render_template("0.html",showcase=(result),showcase1=(x))
def nutrition(index):

    import requests

    url = "https://calorieninjas.p.rapidapi.com/v1/nutrition"

    querystring = {"query":index}

    headers = {
        "X-RapidAPI-Key":
"85887549f4msh51e7315b280a87ep1f43e0jsn585c940f2ea6",
        "X-RapidAPI-Host": "calorieninjas.p.rapidapi.com"
     }
```

```
    response = requests.request("GET", url, headers=headers,
params=querystring)

    print(response.text)
    return response.json()['items']
if __name__ == "__main__":
    # running the app
    app.run(debug=False)
```