PROJECT REPORT

AI-Powered Nutrition Analyzer For Fitness Enthusiasts

Submitted By

PNT2022TMID03918

Srija G - 412519104139

Monisha G - 412519104079

Pooja kumari M B - 412519104096

Varshitha P V - 412519104095

Priyadharshini K - 412519104100

Project Report Format

INTRODUCTION
Project Overview
Purpose
LITERATURE SURVEY
Existing problem
References
Problem Statement Definition
IDEATION & PROPOSED SOLUTION
Empathy Map Canvas
Ideation & Brainstorming
Proposed Solution
Problem Solution fit
REQUIREMENT ANALYSIS
Functional requirement
Non-Functional requirements
PROJECT DESIGN
Data Flow Diagrams
Solution & Technical Architecture
User Stories
PROJECT PLANNING & SCHEDULING
Sprint Planning & Estimation
Sprint Delivery Schedule
Reports from JIRA
CODING & SOLUTIONING (Explain the features added in the project along with code)
Feature 1
Feature 2
Database Schema (if Applicable)
TESTING
Test Cases
User Acceptance Testing
RESULTS
Performance Metrics
ADVANTAGES & DISADVANTAGES
CONCLUSION
FUTURE SCOPE
APPENDIX
Source Code
GitHub & Project Demo Link

1.INTRODUCTION

1.1 Project Overview

Food is essential for human life and has been the concern of many healthcare conventions. Nowadays new dietary assessment and nutrition analysis tools enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and maintaining a healthy diet. Nutritional analysis is the process of determining the nutritional content of food. It is a vital part of analytical chemistry that provides information about the chemical composition, processing, quality control and contamination of food.

1.2 Purpose

The main aim of the project is to build a model which is used for classifying the fruit depending on the different characteristics like colour, shape, texture etc. Here the user can capture the images of different fruits and then the image will be sent to the trained model. The model analyzes the image and detects the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).

2. LITERATURE SURVEY

2.1 Existing problem

The calculation of food calories is critical in the medical field, because the calories in this food are beneficial to one's health. This measurement is taken from the image of food in various objects, such as fruits and vegetables. This measurement is performed with the assistance of a neural network. Tensor flow is one of the most effective methods for classifying machine learning methods. This method is used to calculate food calories using Convolutional Neural Networks. Despite this, their informational collection is moderately small, and with further growth in the informational index, the mean error can be reduced even further. Meng-Lin Chiang enhanced the performance of both analytics and graphics, the system is built on a Mask Region-based Convolutional Neural Network (R-CNN) with a union postprocessing that alters the extracted bounding boxes and masks without employing non-maximum suppression (NMS). The combination of the Ville Cafe and Food-256 datasets had a recognition accuracy of 99.86 percent, and the intersection over union (IoU) was 97.17 percent.M. Sundaramurthy, created and constructed a graphical user interface (GUI) based interactive tool that can accurately detect the type of food. Using a deep learning method, they successfully developed a compact and cost-effective Food Classifier and Nutrition Interpreter Tool (FCNI). Through a multimedia response, the FCNI Tool provides the user with the most information about the recognized food. The FCNI Tool can analyze a variety of foods, and the system can recommend a healthy diet based on the user's health. Haoyu Hu proposed a method for recognizing food and estimating calories. Following the analysis of raw data, the chosen dataset is subjected to data cleaning. In the second section, they employed an SSD for object detection. They used SSD to speed up the process Default boxes and multi-scale features were introduced

to compensate for the loss of accuracy. Convolutional Neural Network (CNN) is one of the most well-known deep learning techniques. Mang Ning proposed a YOLOv4 object model that detects all objects in an image by changing the output range of YOLOv4 and the associated image label. Experiments with the COCO dataset show the effectiveness of our method by achieving a 67.97% recall. Incomplete labels (COCO is only labels in 80 categories) impair the learning process of object search and point out that if the dataset is fully labeled, their method can achieve higher recall. N Murali Krishna's paper focuses on deep learning and how to apply it for object detection and classification. The advantage of using such an algorithm is that it improves performance as the amount of data increases. This is different from traditional learning algorithms, where the performance of the set data increases and the performance stabilizes. YOLO implements object detection as a regression problem and provides the class probability of the detected image.

2.2 References

[1] Fahira, P. K., Rahmadhani, Z. P., Mursanto, P., Wibisono, A., & Wisesa,H. A. "Classical Machine Learning Classification for Javanese Traditional Food Image". 2020 4th International Conference on Informatics and Computational Sciences (ICICoS) (pp. 1-5).

[2] First Teddy Surya Gunawan., Mira Kartiwi, Noreha Abd Malik., Nanang Ismail," Food Intake Calorie Prediction using Generalized Regression Neural Network" 2018 IEEE 5th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)

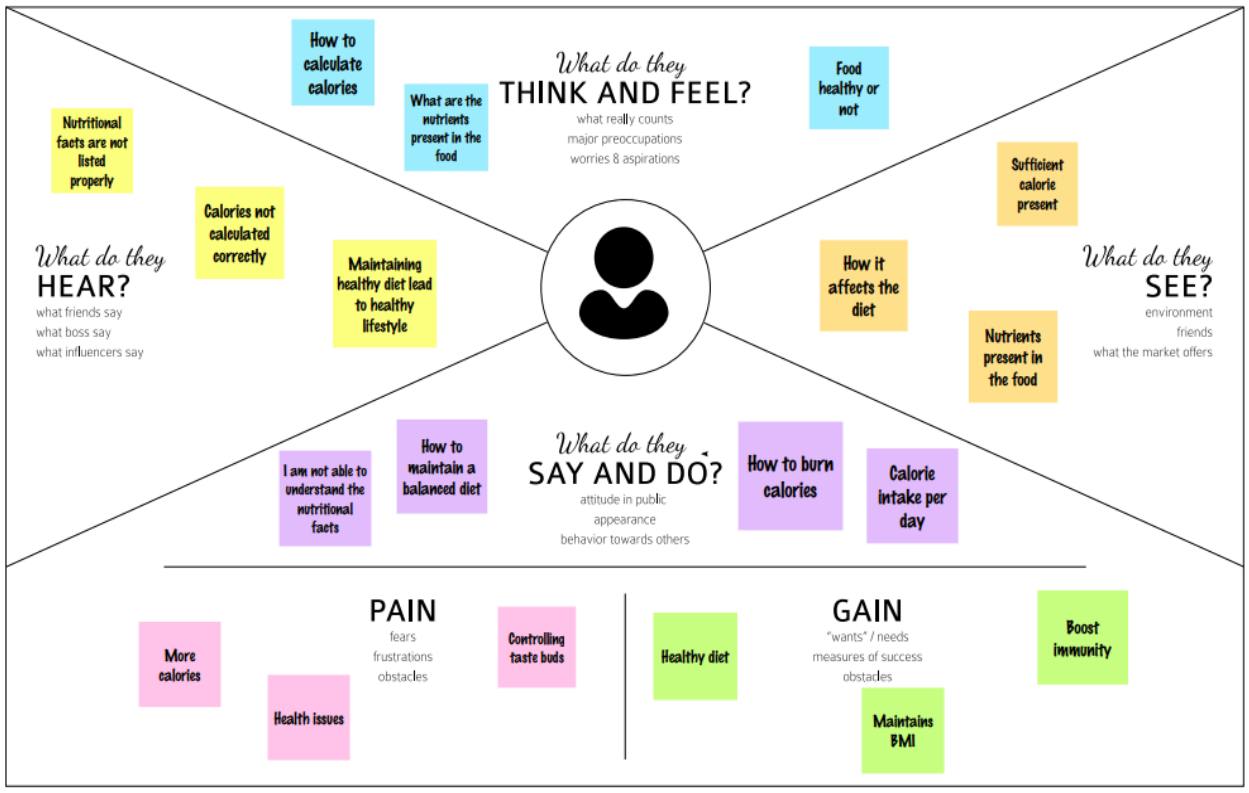
[3] Haoyu Hu., Yulin Song.,Zihao Zhang "Image Based Food Calories Estimation Using Various Models of Machine Learning" 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICM CCE) (pp. 1874-1878)

2.3 Problem Statement Definition

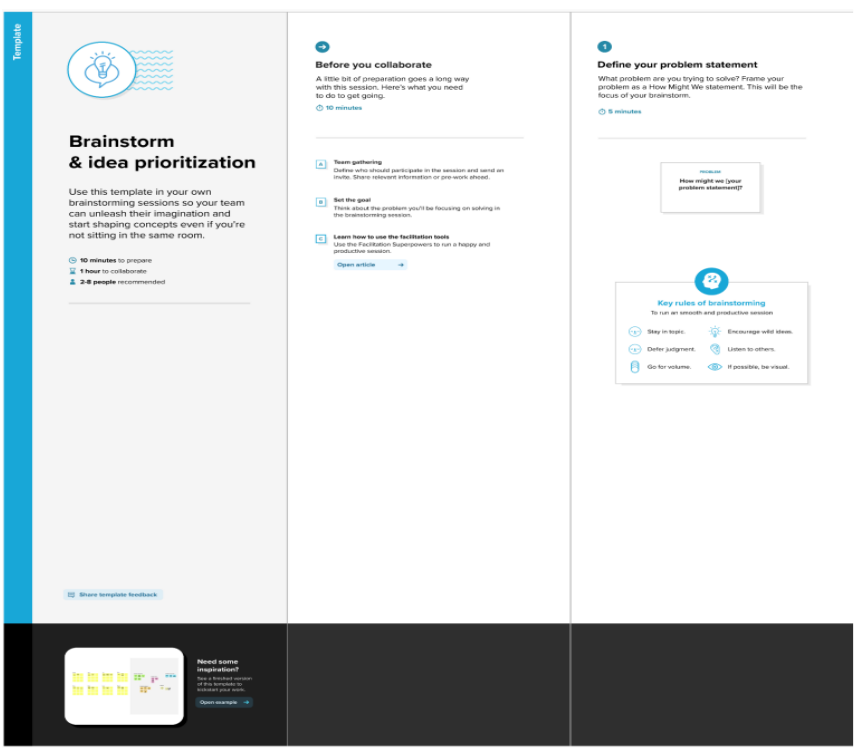
Obesity has become a serious problem in recent decades. Obesity is caused by consuming more calories than that are burned through exercise and daily activities. Convolutional Neural Network (CNN) is a type of neural network that is optimized to process pixel data and thus identify the given input image. To aid with this problem, we create a food nutrition system that can analyze the composition of a food based on a sample image provided.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Srija G

- Predicts nutritional facts
- Helps to maintain diet
- Calorie consciousness
- Health and fitness tracking
- Helps to attain fitness goals
- Giving diet plans according to customer requirements

Priyadharshini K

- Simple diet tracking
- Helps to lead healthy life
- Helps in reducing foodborne diseases
- Evaluation of nutritional qualities of diets, food habits, and/or food supplies
- Enables better food fortification
- It provides a better quality check of the product.

Monisha G

- Healthy intakes
- follow a specific diet plan
- Feedback is given to the user based on the diet
- Helps to stay fit and healthy
- update the user with the diet they intake
- Helps you to reach dietary goals

Varshitha P V

- Quality food
- Estimates calories and nutrients
- Helps to reduce excess consumption
- Helps people to act accordingly for allergies and diabetes
- Suggests correct intake levels
- Helps to consume healthy food

Pooja Kumari M B

- Monitor dietary patterns
- Analyze food intake
- Limit calorie consumption
- helps in strengthening immunity
- Improving body metabolisms
- Detailed nutritional analysis

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

User Requirement

Reduce weight/Burn unwanted fat

Gain weight/ build muscle

To maintain healthy lifestyle

Suggesting diet and nutritional plan

Intake of low calorie food

Eat Nutrient and protien rich food

Proper amount of food intake

TIP

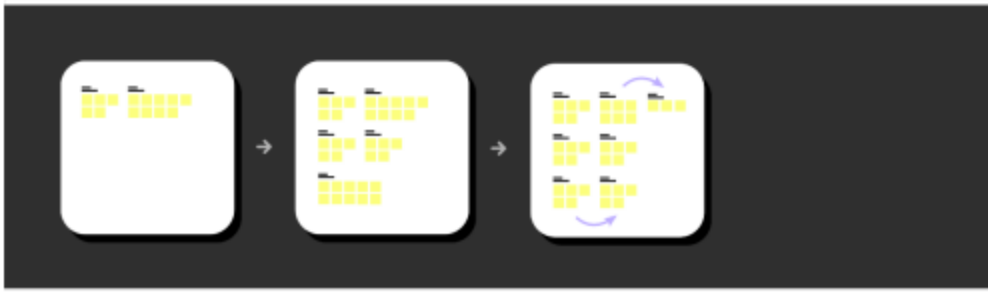
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Following plans and reaching goals

Sticking to diet plan

Consume nutritionous food and stay healthy

Intake protien rich food and build strong muscle

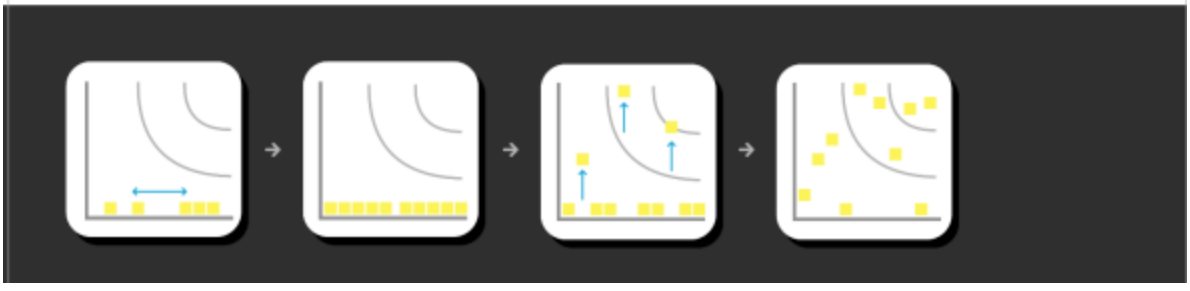


4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

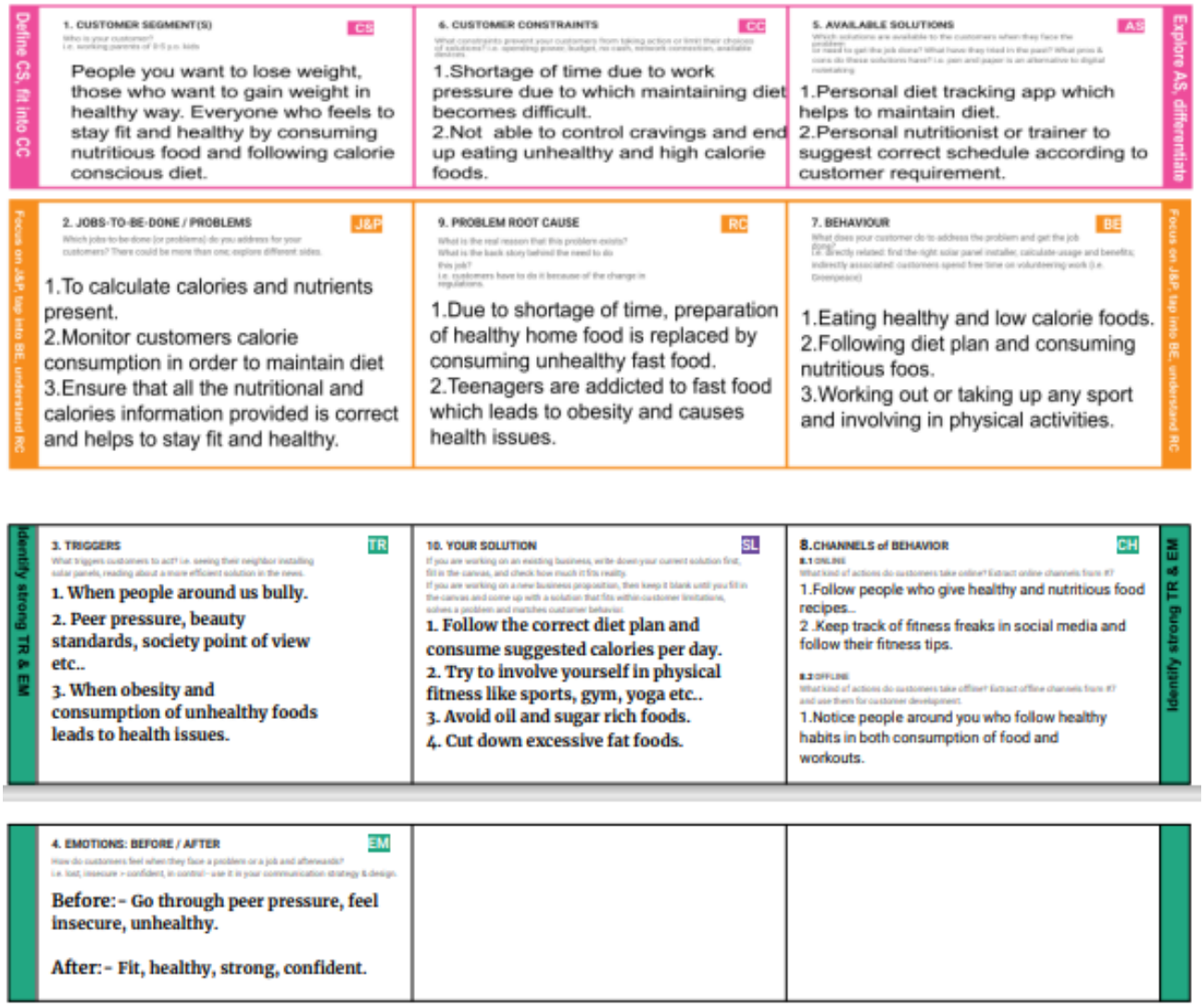


3.3 Proposed Solution

S.No.	Parameter	Description
1.	**Problem Statement (Problem to be solved)**	**Statement:-** To create a food nutrition system that can analyze the composition of a food based on a sample image provided. **Description:-** To design a system that accepts sample food images from the user and compares it with the provided dataset.To identify the kind of food present in the provided sample and display their names along with their nutritional content.
2.	**Idea / Solution description**	We are up to implementing an user-friendly interface which provides the nutritional contents and calories content in the food which the user intakes.
3.	**Novelty / Uniqueness**	1. We will be using Nanonet API as a back-end tool to train and build our image classification model using the deep learning approach. We will be using the YOLO(You Only Look Once) algorithm for object localization. 2.Yolo algorithm is found to be more accurate than R-CNN, FAST R-CNN and FASTER R-CNN.
4.	**Social Impact / Customer Satisfaction**	1.Nutrition Analyser is a web-app that was created using Artificial Intelligence. 2.The main aim of the project is to build a model which is used for classifying the fruit depending on the different characteristics like colour, shape, texture etc. Here the user can capture the images of different fruits and then the image will be sent to the trained model. The model analyzes the image and detects the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).
5.	**Business Model (Revenue Model)**	With widespread advancements in data gathering and manipulation, as well as machine learning becoming widely used and low cost, AI is one of today's most heralded technologies. For AI technologies to be profitable, the correct revenue models still

		need to be chosen. Using these standards, recurring income prospects that are subscription-, service-, and usage-based.
6.	**Scalability of the Solution**	Using AI algorithms the nutritional analysis of the food is predicted and analyzed in a better way. All the nutritional facts and calorie contents are calculated properly.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration and User Confirmation	The user can register into the fitness website using his/her login credentials. Registered users are verified through Email or mobile number using OTP. Each user who has registered is provided with a personal nutritionist to guide them according to their fitness requirements. Users who have not registered can access the website to check out the nutritional values of food.
FR-2	User Interaction With nutritionist	All the registered users can interact with their personalized nutritionist through chat with nutritionists. Unregistered users can ask general queries in the chat box.
FR-3	User Satisfaction	The UI/UX of the website must be simple and user friendly to use. The nutritionist assigned to each user should be friendly to the user so that they feel comfortable sharing their fitness problems and requirements.
FR-4	User Engagement/Feedback	The users should keep themselves engaged by logging into the website everyday to follow their fitness routine. Users should give feedback about their everyday routine as it helps the nutritionist to know the user's perspective.

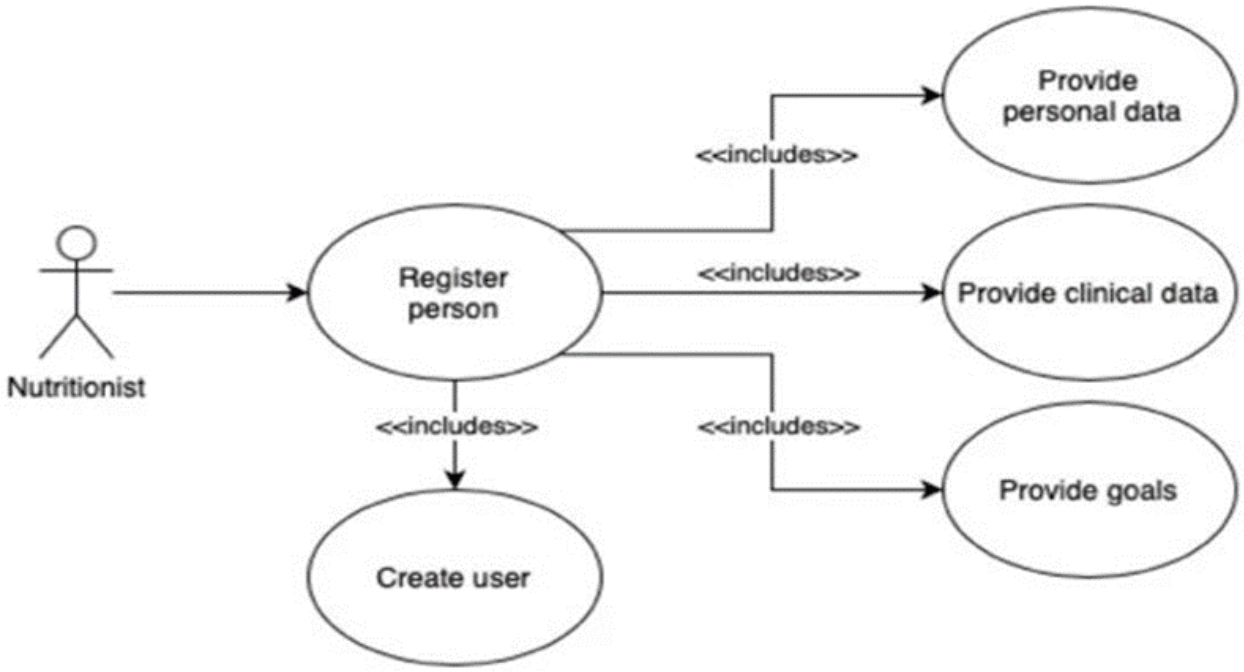
4.2 Non Functional Requirements

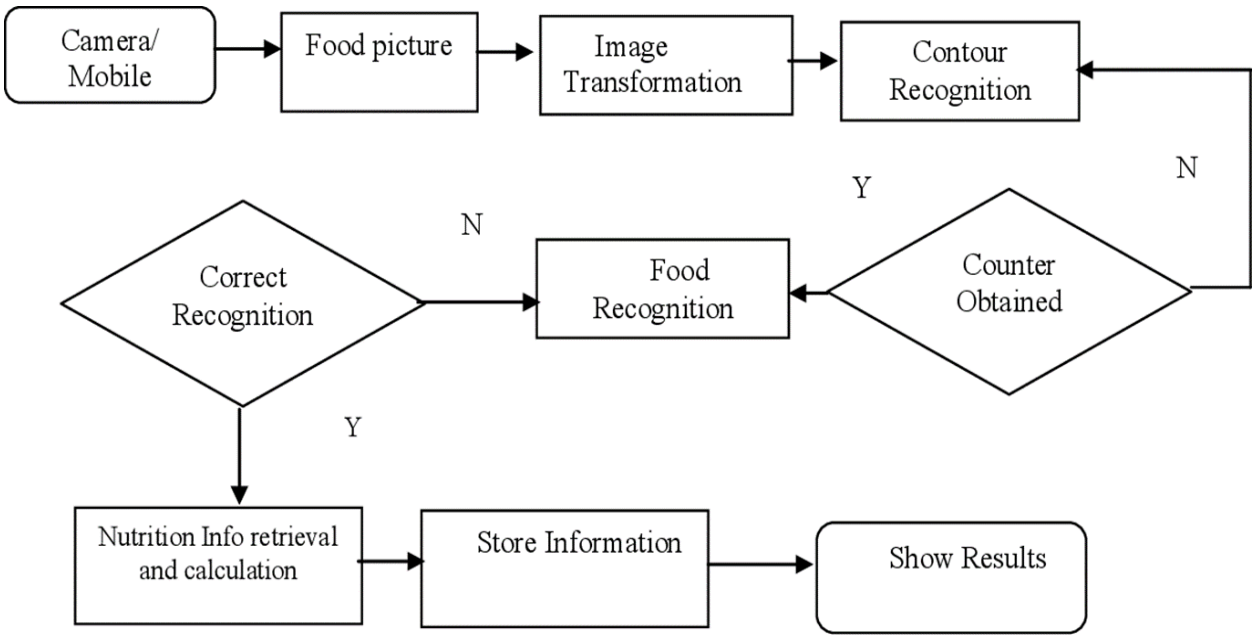
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The website should be user friendly to use Anyone from age group above 13 must be able to access The nutritional facts mentioned must be clear and correct
NFR-2	**Security**	All the user's data who have registered are confidential and secure. Each registered user is provided with unique login credentials to maintain security.
NFR-4	**Performance**	The website must be able to show the results of nutritional facts of food within a few seconds.
NFR-5	**Availability**	Personal nutritionist guidance is restricted to unregistered users
NFR-6	**Scalability**	The system should be able to handle a large number of users accessing the website at same time.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.


Provide personal data
<<includes>>
Nutritionist
Register person
<<includes>>
Provide clinical data
<<includes>>
<<includes>>
Provide goals
Create user


Camera/ Mobile
Food picture
Image Transformation
Contour Recognition
N
Y
Counter Obtained
N
Food Recognition
Correct Recognition
Y
Nutrition Info retrieval and calculation
Store Information
Show Results

5.2 Solution & Technical Architecture

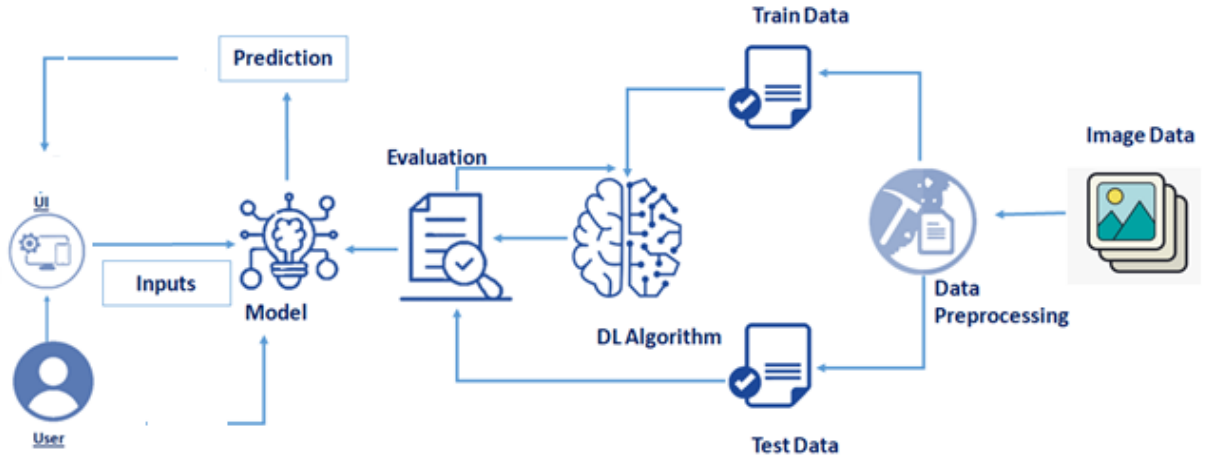


Table-1 : Components and Technologies:

S. No.	Component	Description	Technology
1.	User Interface	An app that lets users make profiles, upload photos of the components they use in their food, and obtain a personalized nutrition calendar.	HTML, CSS, JavaScript
2.	Image Capture	Users are required to take a photo of the ingredient(s) they eat.	IBM Maximo Image Inspection
3.	Ingredient Detection Model	The ingredients used must be identified from the captured image.	Machine Learning & Image Processing using Python
4.	Calorie Consumption Monitoring	The software monitors the user's daily calorie intake and alerts them when there is an excess.	IBM Push Notifications
5.	Database of Ingredients	Ingredient information and the relevant calories are kept on file.	MySQL
6.	Cloud Database for Back-up	Here, backup copies of the application's data are kept, and consolidated reports of monthly calendars are also kept.	IBM Cloudant
7.	File Storage	A file system is used to keep track of the products consumed each day as well as the daily caloric intake. Additionally, a customized calorie calendar	IBM Block Storage

S. No.	Component	Description	Technology
		is created using this.	
8.	Calorie Value Consolidation	To determine the calorie counts of components that are saved in the database, a web-scraping API is used.	Beautiful Soup
9.	Machine Learning Model	To detect substances, captured photos are analyzed using machine learning algorithms.	Object Recognition Model to Label Ingredients
10.	Infrastructure (Server / Cloud)	The program is deployed to the cloud for use. Configuration of the cloud server:	Cloud Foundry

Table-2: Application Characteristics:

S. No.	Characteristics	Description	Technology
1.	Open-Source Frameworks	Google Colab, VS Code, Online Websites	Python, HTML, CSS, JavaScript
2.	Security Implementations	Email-based data access authentication and text encryption before file storage	SMTP, Encryption Algorithms
3.	Scalable Architecture	Applications are updated, bugs are fixed, and new features are added in response to user experience and input.	Customer feedback, reviews, andratings
4.	Availability	Users should always be able to access the cloud-hosted application, and they shouldn't experience any problems like application crashes.	IBM Cloud
5.	Performance	The application should be able to process many requests without sacrificing the speed or quality of the results.	Testing - Black, White, and Beta Revise application in spiral model

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming	I can access my account / dashboard	High	Sprint-1

			my password.			
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	Visit the dashboard for accessing the features of the application	The dashboard is user-friendly	Low	Sprint-1
	Upload Image	USN-7	Uploading the images of the food in the website	As a user, they are able to attend the food images	High	Sprint-3
	Predict	USN-8	The model is able to classify and predict the nutrients present in the food	From the images uploaded, I can access the nutrition facts	High	Sprint-3
		USN-9	We train the model to test various images and ensure that the output accuracy is good	I'm able to train and test the application till the results are as accurate as possible	Medium	Sprint-4
Customer (Web user)	Login	USN-10	As a user, I can use the application by entering my email and password.	I can access my account	Medium	Sprint-4
Customer Care Executive	Dashboard	USN-11	Upload the image	Recognizing and get the output	High	Sprint-1
Administrator	Security	USN-12	updated the features	checking the security	Medium	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

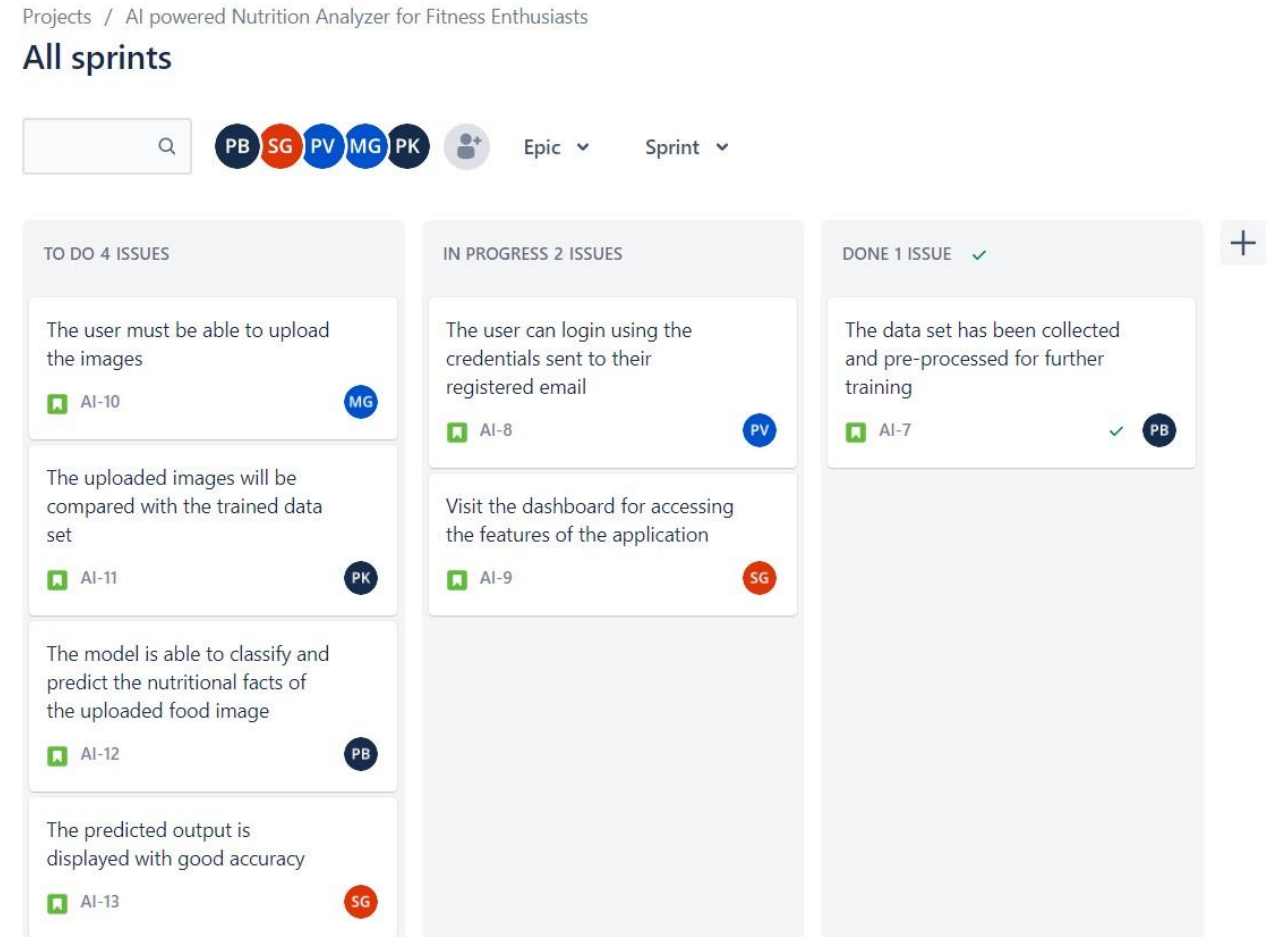
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password	2	High	Srija G Monisha G Varshitha P V Pooja Kumari M B Priyadharshini K
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Srija G Monisha G Varshitha P V Pooja Kumari M B Priyadharshini K
Sprint-1		USN-3	As a user, I can register for the application through Gmail	2	Medium	Srija G Monisha G Varshitha P V Pooja Kumari M B

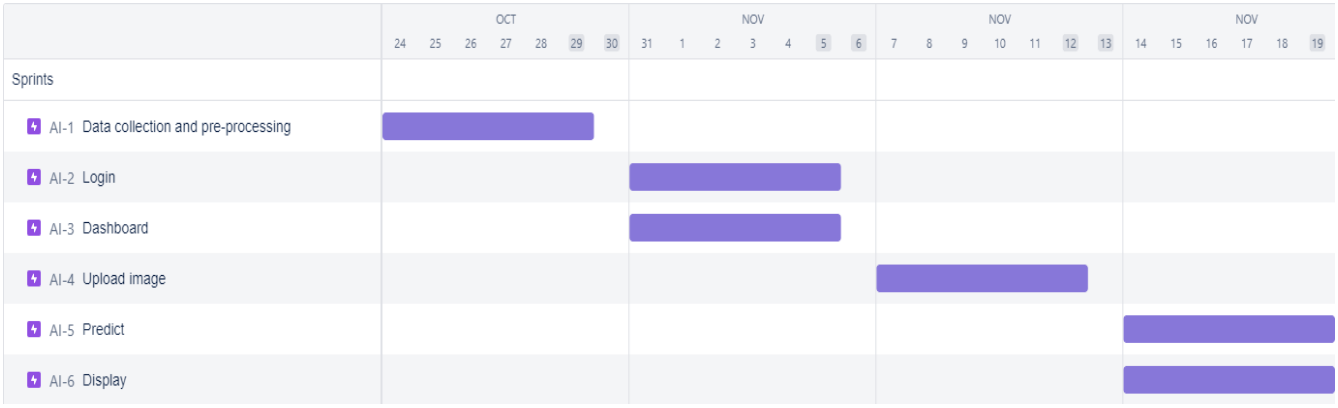
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
						Priyadharshini K
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	2	Medium	Srija G Monisha G Varshitha P V Pooja Kumari M B Priyadharshini K
Sprint-1	Dashboard	USN-5	Visit the dashboard for accessing the features of the application		Low	Srija G Monisha G Varshitha P V Pooja Kumari M B Priyadharshini K
Sprint-2	Dashboard	USN-6	Upload the image		High	Srija G Monisha G Varshitha P V Pooja Kumari M B Priyadharshini K
Sprint-2		USN-7	updated the features		Medium	Srija G Monisha G Varshitha P V Pooja Kumari M B Priyadharshini K
Sprint-3	Upload image	USN-8	Uploading the images of the food in the website		High	Srija G Monisha G Varshitha P V Pooja Kumari M B

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
						Priyadharshini K
Sprint-3	Predict	USN-9	The model is able to classify and predict the nutrients present in the food		High	Srija G Monisha G Varshitha P V Pooja Kumari M B Priyadharshini K
Sprint-4		USN-10	We train the model to test various images and ensure that the output accuracy is good		Medium	Srija G Monisha G Varshitha P V Pooja Kumari M B Priyadharshini K
Sprint-4	Login	USN-11	As a user, I can use the application by entering my email and password.		Medium	Srija G Monisha G Varshitha P V Pooja Kumari M B Priyadharshini K

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

6.3 Reports from JIRA





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

● AI-powered Nutrition Analyzer for Fitness Enthusiasts
● The main aim of the project is to building a model which is used for classifying
The fruit depends on the different characteristics like colour, shape, texture etc.
Here the user can capture the images of different fruits and then the image will
be sent the trained model. The model analyzes the image and detect the nutrition
based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).
● Languages : Python
● Tools/IDE : Google collaboratory , Spyder
● Libraries : Recommendation

from flask import Flask,render_template,request
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
import os
import numpy as np #used for numerical analysis
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="templates") # 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():
return render_template('home.html')
@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 launches():
if request.methods=='POST':
f=request.files['file'] #requesting the file
basepath=os.path.dirname('__file__')#storing the file directory
filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in uploads folder
f.save(filepath)#saving the file

```
        img=image.load_img(filepath,target_size=(64,64)) #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(&quot;prediction&quot;,pred)#printing the prediction
        index=[&#39;APPLES&#39;,&#39;BANANA&#39;,&#39;ORANGE&#39;,&#39;PINEAPPLE&#39;,
&#39;WATERMELON&#39;]
        result=str(index[pred[0]])
        x=result
        print(x)
        result=nutrition(result)
        print(result)
        return render_template(&quot;0.html&quot;,showcase=(result))
        import http.client
        conn = http.client.HTTPSConnection(&quot;calorieninjas.p.rapidapi.com&quot;)
        headers = {
        &#39;X-RapidAPI-Key&#39;:
&quot;e5805fbf62mshf8d7308c0600c2dp197087jsn93407e3cce35&quot;,
        &#39;X-RapidAPI-Host&#39;: &quot;calorieninjas.p.rapidapi.com&quot;
        }
        conn.request(&quot;GET&quot;, &quot;/v1/nutrition?query=Pineapple&quot;, headers=headers)
        res = conn.getresponse()
        data = res.read()

        print(data.decode(&quot;utf-8&quot;))
        import requests
        url = &quot;https://calorieninjas.p.rapidapi.com/v1/nutrition&quot;
        querystring = {&quot;query&quot;:&quot;Pineapple&quot;}
        headers = {
        &quot;X-RapidAPI-Key&quot;:
&quot;e5805fbf62mshf8d7308c0600c2dp197087jsn93407e3cce35&quot;,
        &quot;X-RapidAPI-Host&quot;: &quot;calorieninjas.p.rapidapi.com&quot;
        }
        response = requests.request(&quot;GET&quot;, url, headers=headers, params=querystring
        print(response.text)
        if __name__ == &quot;__main__&quot;:
        # running the app
        app.run(debug=False)
```

8.TESTING

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('train.h5')
model=load_model('dataset.h5')
model=load_model('nutrition.h5')
img=image.load_img(r"/content/drive/MyDrive
/CNN/Dataset/TEST_SET/WATERMELON/3_100.jpg")
img
```

```
img=image.load_img(r"/content/drive/MyDrive
/CNN/Dataset/TEST_SET/WATERMELON/3_100.jpg",
target_size=(64,64))
img
```

```
x=image.img_to_array(img)
x
array([[[[255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.],
        ...,
        [255., 255., 255.],
        [255., 255., 255.],
        [255., 255., 255.]],

        [[255., 255., 255.],
        [255., 255., 255.],
```

```
 [255., 255., 255.],
 ...,
 [255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.]],

[[255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.],
 ...,
 [255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.]],
   ...,

[[255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.],
 ...,
 [255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.]],

[[255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.],
 ...,
 [255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.]],

 [[255., 255., 255.],
 [255., 255., 255.],
 [255., 255., 255.],
 ...,
 [255., 255., 255.],
```

```
        [255., 255., 255.],
        [255., 255., 255.]]]], dtype=float32)
```

```
x=np.expand_dims(x,axis=0)
```

```
array([[[[255., 255., 255.],
         [255., 255., 255.],
         [255., 255., 255.],
         ...,
         [255., 255., 255.],
         [255., 255., 255.],
         [255., 255., 255.]],

        [[255., 255., 255.],
         [255., 255., 255.],
         [255., 255., 255.],
         ...,
         [255., 255., 255.],
         [255., 255., 255.],
         [255., 255., 255.]],

        [[255., 255., 255.],
         [255., 255., 255.],
         [255., 255., 255.],
         ...,
         [255., 255., 255.],
         [255., 255., 255.],
         [255., 255., 255.]]]], dtype=float32)
```

```
pred = model.predict
```

```
pred
```

```
array
```

```
([[0.25227112, 0.17414774, 0.15219809, 0.20493415, 0.21644896],
 [0.26760292, 0.1759095 , 0.15206912, 0.19424875, 0.21016978],
 [0.26474723, 0.165203  , 0.14452063, 0.20434381, 0.2211853 ],
        ...,
 [0.24550524, 0.1721549 , 0.16282505, 0.21065485, 0.20885986],
 [0.25395462, 0.1735253 , 0.16055605, 0.20655352, 0.20541045],
 [0.24495909, 0.15889102, 0.16927534, 0.20705006, 0.21982446]],
  dtype=float32
```

```
<bound method Model.predict of <keras.engine.
sequential.Sequential object at 0x7f94abfd7c10>>
```

```
predict_x=model.predict(x_test)
```

```
classes_x=np.argmax(predict_x,axis=1)
```

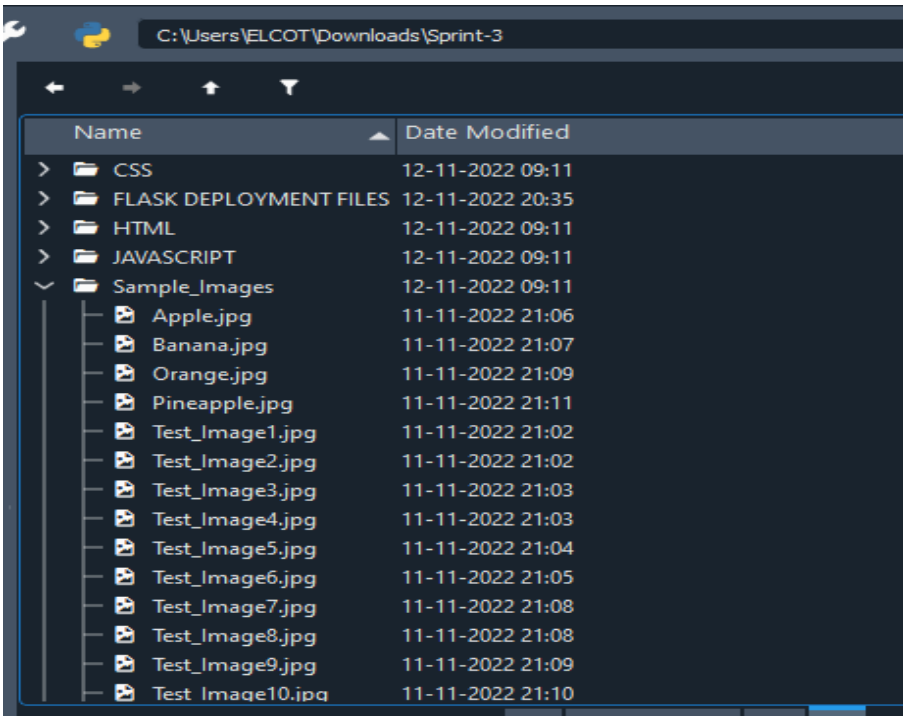
```
classes_x
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

```
x_test.class_indices
```

```
index=['APPLE','BANANA','ORANGE','WATERMELON','PINEAPPLE']
result=str(index[classes_x[0]])
result
'Watermelon''
```

8.1 TEST CASES



8.2 USER ACCEPTANCE TESTING

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done. The main Purpose of UAT is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is a kind of black box testing where two or more end-users will be involved. Need of User Acceptance Testing arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.


APPLE


BANANA


ORANGE


PINEAPPLE


WATERMELON


TEST_IMAGE1


TEST_IMAGE2


TEST_IMAGE3


TEST_IMAGE4

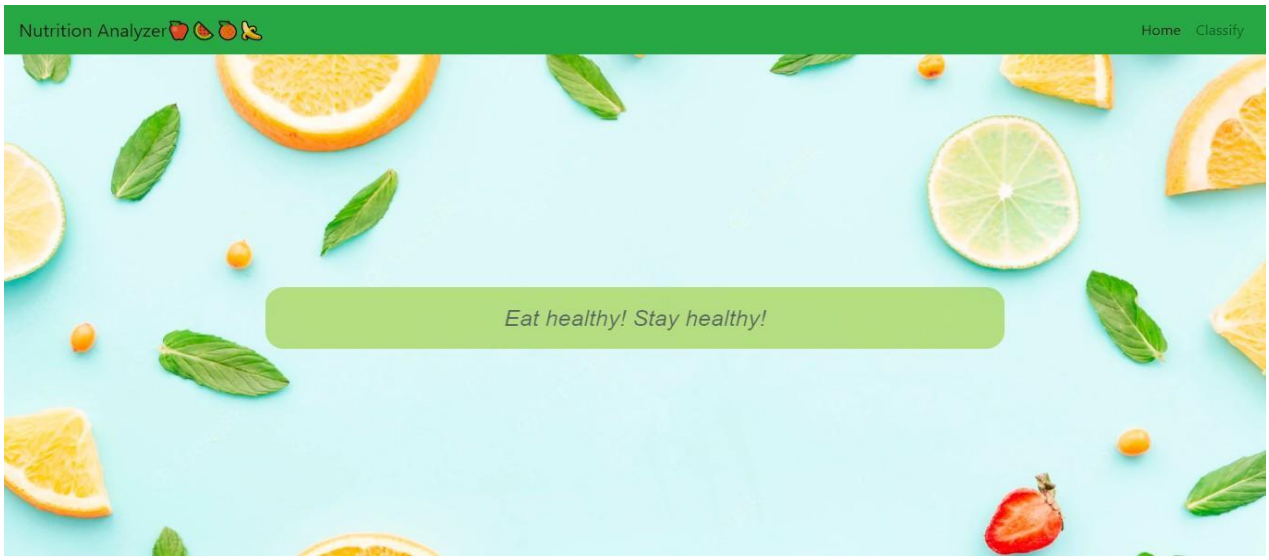

TEST_IMAGE5

PERFORMANCE TESTING:

Epoch 1/10
110/110 [==============================] - 27s 242ms/step - loss: 0.4205 - accuracy: 0.8861 - val_loss: 48.9065 - val_accuracy: 0.1488
Epoch 2/10
110/110 [==============================] - 27s 245ms/step - loss: 0.0082 - accuracy: 0.9989 - val_loss: 62.1670 - val_accuracy: 0.1280
Epoch 3/10
110/110 [==============================] - 28s 255ms/step - loss: 0.0014 - accuracy: 1.0000 - val_loss: 66.6759 - val_accuracy: 0.1488
Epoch 4/10
110/110 [==============================] - 27s 242ms/step - loss: 3.3364e-04 - accuracy: 1.0000 - val_loss: 70.6794 - val_accuracy: 0.1488
Epoch 5/10
110/110 [==============================] - 27s 248ms/step - loss: 1.9990e-04 - accuracy: 1.0000 - val_loss: 74.1865 - val_accuracy: 0.1488
Epoch 6/10
110/110 [==============================] - 26s 236ms/step - loss: 4.5090e-04 - accuracy: 1.0000 - val_loss: 75.5190 - val_accuracy: 0.1308
Epoch 7/10
110/110 [==============================] - 27s 248ms/step - loss: 1.0600e-04 - accuracy: 1.0000 - val_loss: 78.4789 - val_accuracy: 0.1488
Epoch 8/10
110/110 [==============================] - 26s 237ms/step - loss: 7.9529e-05 - accuracy: 1.0000 - val_loss: 80.7918 - val_accuracy: 0.1403

```
Epoch 9/10
110/110 [==============================] - 26s 236ms/step - loss: 9.2201e-05 - accuracy: 1.0000 - val_loss:
80.3610 - val_accuracy: 0.1431
Epoch 10/10
110/110 [==============================] - 29s 266ms/step - loss: 9.1324e-05 - accuracy: 1.0000 - val_loss:
83.0943 - val_accuracy: 0.1393
<keras.callbacks.History at 0x7fbcb5cb4b10>
```

RESULTS




Nutrition Analyzer
Home
Classify
UPLOAD IMAGE
Choose...
Food Classified is:
WATERMELON
[{'sugar_g': 6.2, 'fiber_g': 0.4, 'serving_size_g': 100.0, 'sodium_mg': 0, 'name': 'watermelon', 'potassium_mg': 10, 'fat_saturated_g': 0.0, 'fat_total_g': 0.1, 'calories': 30.3, 'cholesterol_mg': 0, 'protein_g': 0.6, 'carbohydrates_total_g': 7.4}]

10. ADVANTAGES AND DISADVANTAGES

Advantages:

➢ Food and food habits are ever-changing and evolving. People and professionals need to quickly adapt to new food products, diets, and changing preferences. The best way to instantly adapt to these changes is to have software that changes and adapts with you.

➢ Using automated nutrition analysis software will allow you to free up more time to innovate or grow your business. If you find a nutrition analysis software that has all the features you need, you can create much more time to focus on improving your business.

➢ Features such as a quick preview of nutrients while adding foods to diets, menus, and recipes give you the ability to save time when new recipes and food products are introduced.

➢ Having quick and easy software to help them plan their meals will save you tons of time.

Disadvantages:

➢ This methodology is still limited by its dependency on time-consuming and error-prone manual video annotations, with many studies resorting to the use of multiple human annotators.

➢ Often suffers from reliability issues.

➢ It is extremely expensive due to semantics analysis model and nutritional analysis model.

➢ In order to make recommendations, the system needs to collect nutritional needs from users. Most of the information is only provided through continuous

interactions with users. However, in reality, recording nutritional intake from users cannot avoid faults because users usually forget or give wrong information about the foods they have consumed.

➢ Moreover deep learning requires expensive GUIs and hundreds of machines. This increases the cost to the users.

11. CONCLUSION

Food is essential for human life and has been the concern of many health care conventions. In this project we have built a nutrition analysis model that classifies the nutritional content of the food through the image uploaded by the user. Such Nutritional analysis helps people understand their daily eating habits, exploring nutrition patterns and maintaining a healthy diet. It is a vital part of analytical chemistry that provides information about the chemical composition, processing, quality control and contamination of food. The nutritional analysis model is implemented using a Convolutional neural network and the web application is built and implemented using the Flask framework. As for the future work, the model can be trained and tested on more datasets to provide accurate results and better performance.

FUTURE SCOPE

The future scope of this project is very broad. Few of them are:

- The model could be trained using vast database in order to increase the accuracy of results.
- The Backend framework of the web application can be improved so that the uploaded images can be handled appropriately.
- In addition to the nutrition analysis, the application can also be designed to provide recipes that can be prepared using the nutrient-rich foods
- A database can also be implemented for the system so that users can save their data and relook into it later.
- The Web application can be further developed and launched as an Android App so that anyone anywhere with or without internet connection can access it and benefit from its use cases.

APPENDIX

Source Code

1. **app.py**

```
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
os.environ["SM_FRAMEWORK"] = "tf.keras"
# 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)

             #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

        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)
```

2. **0.html**

```
<html lang="en" dir="ltr">
<head>
<style>

</style>
    <meta charset="utf-8">
    <title>Nutrition Image Analysis</title>
                <link    rel="shortcut    icon"    href="{{    url_for('static',
filename='diabetes-favicon.ico') }}">
                    <script     src="https://kit.fontawesome.com/5f3f547070.js"
crossorigin="anonymous"></script>
     <link href="https://fonts.googleapis.com/css2?family=Pacifico&display=swap"
rel="stylesheet">
    <style>

body{

background-image:url("https://img.freepik.com/free-photo/set-fruits-seeds-leave
s_23-2148145087.jpg?w=2000");
    background-size: 400% auto;
  background-repeat: no-repeat;
  background-position:center;
  color:#555;
  font-family:Arial, Helvetica, sans-serif;
  font-size:16px;
  line-height:1.6em;
  margin:0;
}

.container{
```

```
  width:80%;
  margin:auto;
  overflow:hidden;
}

.justify{
    text-align:justify;
    text-justify: auto;
}

.parallax {
  /* The image used */
     background-image: url("doc.jpg");

  /* Set a specific height */
  min-height: 750px;

  /* Create the parallax scrolling effect */
  background-attachment: fixed;
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}

html {
  scroll-behavior: smooth;
}
#section2 {
  height: 500px;
  background:#34495E;
}
div.background {
  background: url("static/bgg2.jpg");
  min-height: 5px;
 background-attachment: fixed;
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}
```

```
#navbar{
  background-color:#fff;
  color:#333;
}

#navbar ul{
  padding:0;
  list-style: none;
}

#navbar li{
  display:inline;
}

#navbar a{
  color:#fff;
  text-decoration: none;
  font-size:18px;
  padding-right:15px;
}

#showcase{
  min-height:300px;
  margin-bottom:30px;
}


#showcase h1{
    width: 100%;
  color:#333;
  font-size:40px;
  text-align: center;
  line-height: 1em;
  padding-top:10px;
}
#showcase h2{
    width: 100%;
  color:#333;
  font-size:30px;
```

```
  text-align: center;
  line-height: 1.6em;
  padding-top:10px;
}

#main{
  float:left;
  color:#fff;
  width:65%;
  padding:0 30px;
  box-sizing: border-box;
}

#sidebar{
  float:right;
  width:35%;
  background-color: #ffcccc;
  color:#000;
  padding-left:10px;
  padding-right:10px;
  padding-top:1px;
  box-sizing: border-box;
}


.img-preview {
    width: 10px;
    height: 10px;
    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: 10%;
    height: 10%;
    background-size: 100px 10px;
    background-repeat: no-repeat;
```

```
    background-position: center;
}

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

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #34495E;
    color: #39D2B4;
}

.myButton {
  border: none;
  text-align: center;
  cursor: pointer;
  text-transform: uppercase;
  outline: none;
  overflow: hidden;
  position: relative;
  color: #fff;
  font-weight: 700;
  font-size: 12px;
  background-color: #ff0000;
  padding: 10px 15px;
  margin: 0 auto;
  box-shadow: 0 5px 15px rgba(0,0,0,0.20);
}

.myButton span {
```

```
  position: relative;
  z-index: 1;
}


.myButton:after {
  content: "";
  position: absolute;
  left: 0;
  top: 0;
  height: 310%;
  width: 150%;
  background: #f2f2f2;
  -webkit-transition: all .5s ease-in-out;
  transition: all .5s ease-in-out;
  -webkit-transform: translateX(-98%) translateY(-25%) rotate(45deg);
  transform: translateX(-98%) translateY(-25%) rotate(45deg);
}

.myButton:hover:after {
  -webkit-transform: translateX(-9%) translateY(-25%) rotate(45deg);
  transform: translateX(-9%) translateY(-25%) rotate(45deg);
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #ff0000; /* Red */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

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

#main-footer{
  background: #333;
```

```
  color:#fff;
  text-align: center;
  padding:1px;
  margin-top:0px;
}

@media(max-width:600px){
  #main{
    width:100%;
    float:none;
  }

  #sidebar{
    width:100%;
    float:none;
  }
}
    </style>
  </head>

    <!-- Result -->
    <div class="results">
                              <p  style="padding-top:  150px;  color:blue;"><h4
style="color:blue;">Food           Classified          is:           <h4><b><h4
style="color:red;"><u>{{showcase1}}<h4><br><h4
style="color:red;"><u>{{showcase}}<h4></p>

    </div>
    <br>
    <br>

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

3. **predict.html**

```
<!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.min.css"
rel="stylesheet">
                                                                    <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
                                                                    <script
src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

<style>
body
{
                                                          background-image:
url("https://img.freepik.com/free-photo/set-fruits-seeds-leaves_23-2148145087.j
pg?w=2000");
    background-size: cover;
}
/* .bar
{
margin: 0px;
padding:20px;
background-color:rgb(147, 173, 177);
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:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
/* a:hover{
background-color:black;
color:white;
border-radius:15px;
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;
```

```
      width: 100%;
      height:8%;
      size: 2vw;
      text-align: center;
      position: fixed;
      background-color:  pink;
      color: white;
      box-shadow: 0px 8px 4px grey;
      overflow: hidden;
      padding-left:20px;
      font-family: 'Josefin Sans'



    }
    .topnav {
  overflow: hidden;
  background-color: pink;
}

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

.topnav-right a:hover {
  background-color:  pink;
  color: black;
}

.topnav-right a.active {
  background-color:  pink;
  color: black;
}

.topnav-right {
```

```
  float: right;
  padding-right:100px;
}

.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: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #34495E;
    color: #39D2B4;
}
```

```
.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

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

</style>
</head>
<body>

<!--Brian Tracy-->

<!-- <div class="header">
<div      style="width:50%;float:left;font-size:2vw;text-align:left;color:black;
padding-top:1%;padding-left:25%;">Nutrition Analyzer 🍎🍉🍊🍌</div>
  <div class="topnav-right"style="padding-top:0.5%;">

    <a class="active" href="/">Home</a>
    <a   href="image1">Classify</a>
  </div>
</div>
</div> -->
<nav class="navbar navbar-expand-lg navbar-light bg-success">
  <a class="navbar-brand" href="#">Nutrition Analyzer🍎🍉🍊🍌</a>
      <button   class="navbar-toggler"   type="button"   data-toggle="collapse"
data-target="#navbarNav"     aria-controls="navbarNav"     aria-expanded="false"
aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse justify-content-end" id="navbarNav">
    <ul class="navbar-nav">
```

```
      <li class="nav-item active">
                                  <a    class="nav-link"    href="#">Home    <span
class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="image1">Classify</a>
      </li>
    </ul>
  </div>
</nav>

<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<h1>

<center>


<h3>Eat healthy! Stay healthy!</h3>

</center>


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

4. **Image.html**

```
<!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>Predict</title>
     <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
                                                                    <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
                                                                    <script
src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

<style>
body
{
                                                          background-image:
url("https://img.freepik.com/free-photo/set-fruits-seeds-leaves_23-2148145087.j
pg?w=2000");
    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;
}
a
{
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
```

```
}
/* a:hover{
background-color:black;
color:white;
border-radius:15px;
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: black;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}

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

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

.topnav-right {
  float: right;
  padding-right:100px;
}

.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: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #34495E;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
</style>
</head>
```

```
<body>
<!-- <div class="header">
<div      style="width:50%;float:left;font-size:2vw;text-align:left;color:black;
padding-top:1%;padding-left:5%;">Nutrtion Image Analysis</div>
  <div class="topnav-right"style="padding-top:0.5%;">

    <a href="/">Home</a>
    <a  class="active" href="ImagePrediction.html">Classify</a>
  </div>
</div> -->
<nav class="navbar navbar-expand-lg navbar-light bg-success">
    <a class="navbar-brand" href="#">Nutrition Analyzer🍎🍉🍊🍌</a>
         <button  class="navbar-toggler"  type="button"  data-toggle="collapse"
data-target="#navbarNav"     aria-controls="navbarNav"     aria-expanded="false"
aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse justify-content-end" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item active">
                                  <a    class="nav-link"    href="/">Home    <span
class="sr-only">(current)</span></a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="image1">Classify</a>
        </li>
      </ul>
    </div>
  </nav>
<br>


</div>
<div class="container">
      <center>
<div id="content" style="margin-top:2em">

    <div style="float:left">
        <br>
        <br>
```

```
            <h5><font color="black" size="4" font-family="sans-serif"><b>UPLOAD
IMAGE</b></font></h5><br><br>

        <div>
            <form id="upload-file" method="post" enctype="multipart/form-data">
               <label for="imageUpload" class="upload-label">
                    Choose...
                </label>
                  <input type="file" name="file" id="imageUpload" accept=".png,
.jpg, .jpeg">
            </form>

           <center> <div class="image-section" style="display: contents;">
                <div class="img-preview">
                    <div id="imagePreview">
                    </div></center>
                </div>
                <center><div>
                          <button type="button" class="btn btn-primary btn-lg "
id="btn-predict" onclick="clickEvent()" >Classify</button>
                    <script>
                        function clickEvent()  {
    // Init
    $('.image-section').show();
    $('.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').show();
                $('#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 A 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').html(data);
                console.log('Success!');
            },
        });
    });

}

                </script>
            </center></div>
        </div>
```

```
            <div class="loader" style="display:none;margin-left: 450px;"></div>

            <h3 id="result">

                    <span><p style="padding-top: 25px;"><h4>Food Classified is :
<h4><b><u>{{showcase}}{{showcase1}}</p> </span>
            </h3>

        </div>
        </div>



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



</html>
```

5. **Image prediction.html**

```
<!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>Predict</title>
     <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
                                                                    <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
                                                                    <script
src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
```

```
<style>
body
{
                                                    background-image:
url("https://img.freepik.com/free-photo/set-fruits-seeds-leaves_23-2148145087.j
pg?w=2000");
    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;
}
a
{
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
/* a:hover{
background-color:black;
color:white;
border-radius:15px;
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: black;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 18px;
}
```

```css
.topnav-right a:hover {
  background-color: #FF69B4;
  color: black;
}

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

.topnav-right {
  float: right;
  padding-right:100px;
}

.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: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #34495E;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
</style>
</head>
<body>
<div class="header">
<div      style="width:50%;float:left;font-size:2vw;text-align:left;color:black;
padding-top:1%;padding-left:5%;">Nutrtion Image Analysis</div>
  <div class="topnav-right"style="padding-top:0.5%;">

    <a href="/">Home</a>
    <a  class="active" href="/image1">Classify</a>
  </div>
</div>
<br>
```

```
</div>
<div class="container">
      <center>
<div id="content" style="margin-top:2em">

  <div style="float:left">
    <br>
    <br>
     <h5><font color="black" size="3" font-family="sans-serif"><b>Upload image
to classify</b></font></h5><br><br>

    <div>
        <form id="upload-file" method="post" enctype="multipart/form-data">
           <label for="imageUpload" class="upload-label">
                Choose...
            </label>
             <input type="file" name="file" id="imageUpload" accept=".png, .jpg,
.jpeg">
        </form>

       <center> <div class="image-section" style="display:none;">
            <div class="img-preview">
                <div id="imagePreview">
                </div></center>
            </div>
            <center><div>
                        <button type="button" class="btn btn-primary btn-lg "
id="btn-predict">Classify</button>
           </center></div>
        </div>

        <div class="loader" style="display:none;margin-left: 450px;"></div>

        <h3 id="result">

                 <span><p style="padding-top: 25px;"><h4>Food Classified is :
<h4><b><u>{{showcase}}{{showcase1}}</p> </span>
        </h3>

    </div>
```

```
    </div>




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

<footer>
    <script src="E:\19euec050\main.js" type="text/javascript"></script>
</footer>

</html>
```

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-25480-1659964974

Project Demo Link: https://youtu.be/1Len-3fogNA