RMK ENGINEERING COLLEGE

(An Autonomous Institution)

R.S.M. Nagar, Kavaraipettai-601 206,

Gummidipoondi Taluk, Thiruvallur District.

PROJECT

AI powered nutrition analyzer for fitness enthusiasts

DONE BY

TEAM ID: PNT2022TMID15800

P JAI SIVA RANJANI – 111719104118

ISHWARYA S – 111719104062

LAVANYA S – 111719104084

MANASA G C - 111719104090

CONTENTS

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 maintain 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 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 analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).

2. LITERATURE SURVEY

2.1 Existing problem

Nowadays people eat all junk food, they are not having balanced diet. This affects their health life. Current lifestyle produces many health-related problems like obesity, blood pressure. These need to be governed.

2.2 References

A Mathematical AI-Based Diet Analysis and Transformation Model.
K. Gautam S. A. Ladhake

The AI-based mathematical model for diet optimization and transformation solves the common nutritional problems of public health problems in India using fuzzy arithmetic and a search spa metaheuristic algorithm. In the future, we intend to deal with optimizing diet, based on cost (price) as an additional objective function. It would enlarge the state space of the metaheuristic search algorithm and would greatly help in finding more optimized solution

A Survey on AI Nutrition Recommender Systems

Kosmas Dimitropoulos, Lazaros Gymnopoulos

This work provided an overview of existing AI nutrition recommender systems, a hold that has experienced substantial growth in the last few years. An assessment of the feasibility of implementing an ideal AI nutrition recommender system using current methods was also provided, with the general conclusion being that some of the required components have not reached a mature state yet.

Artificial Intelligence in Nutrients Science Research: A Review

Jarosław Sak, Magdalena Suchodolska

Despite the fact that AI technologies are dynamically developing, the problem in nutrients research is not currently obtaining more and more

advanced algorithms, but the application of those that have already been developed and are standardly used in other fields of knowledge, and even in other areas of biomedicine.

A New Deep Learning-based Food Recognition System for Dietary Assessment on An Edge Computing Service Infrastructure

Chang Liu, Yu Cao

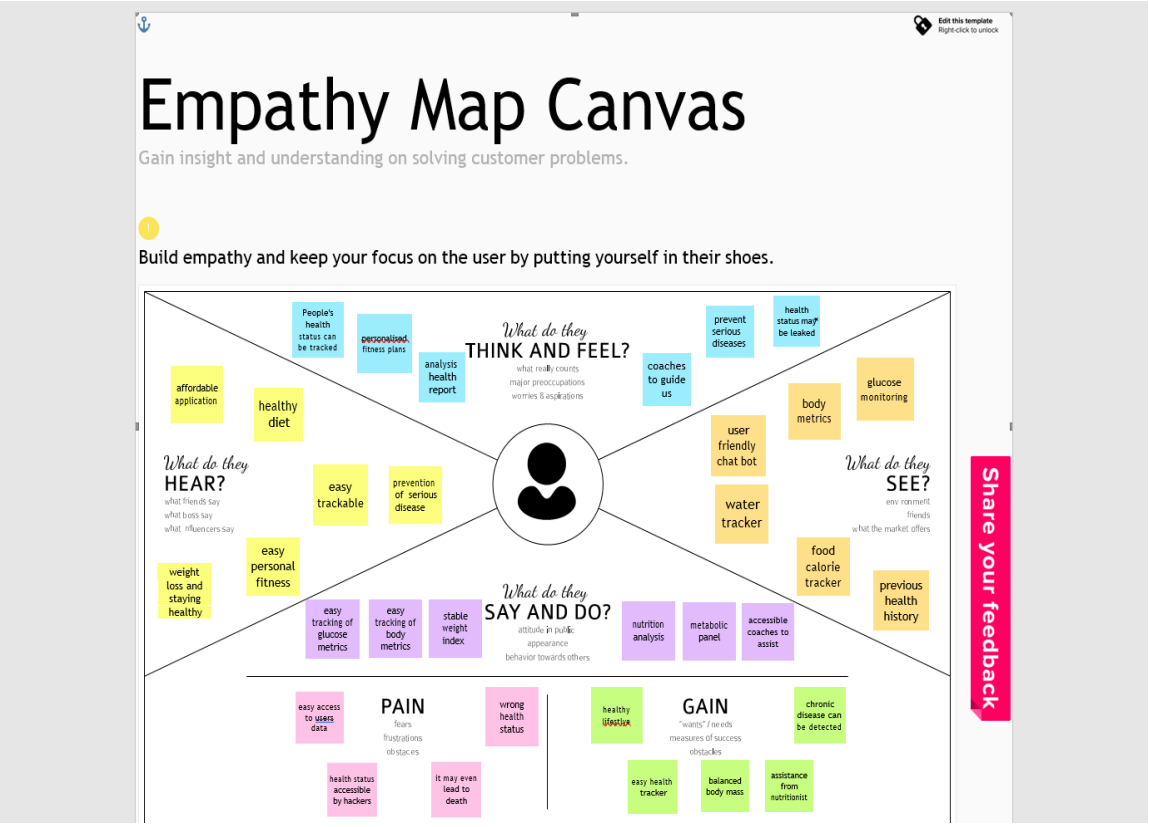
The key technical innovation in this paper includes the new deep learning-based food image recognition algorithms and the proposed real-time food recognition system employing the edge computing service paradigm. Our experimental results on two challenging data sets using our proposed approach have demonstrated that our system has achieved the three major objectives: (1) it outperforms the results from all existing approaches in terms of recognition accuracy; (2) it develops a real-time system whose response time is close to the minimal of existing techniques; and (3) it saves the energy by keeping the energy consumption equivalent to the minimum of the existing approaches

2.3 Problem Statement Definition

As people across the globe are becoming more health conscious, eating more healthy food and avoiding junk food, a system that can measure calories and nutrition in everyday meals can be very useful for maintaining one's health. Food calorie and nutrition measurement system is very beneficial for dieticians and patients to measure and manage their daily food intake. We also know that it's difficult to find an affordable nutritionist or a dietician across the street; therefore, we have proposed a system.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

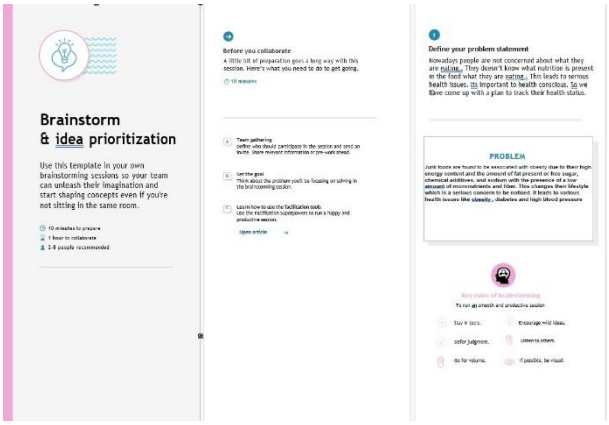


3.2 Ideation and brainstorming

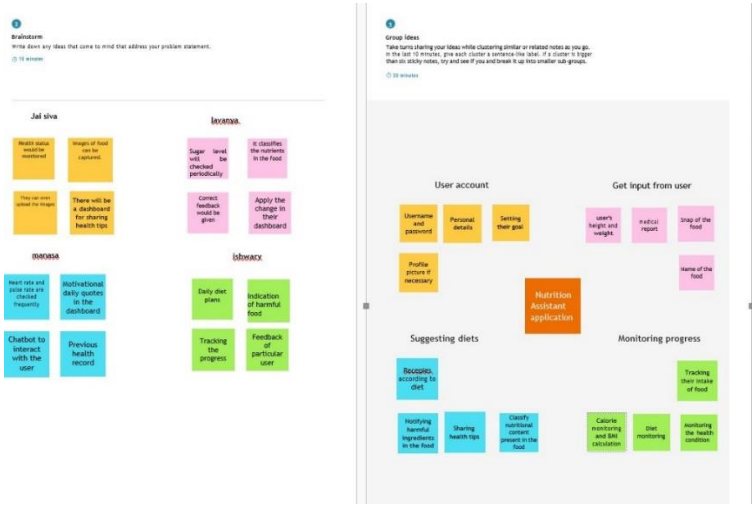
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem-solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference: https://www.mural.co/templates/empathy-map-canvas

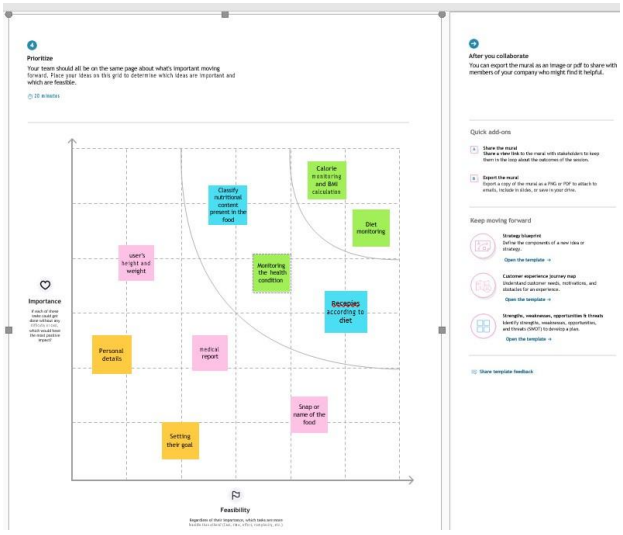
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

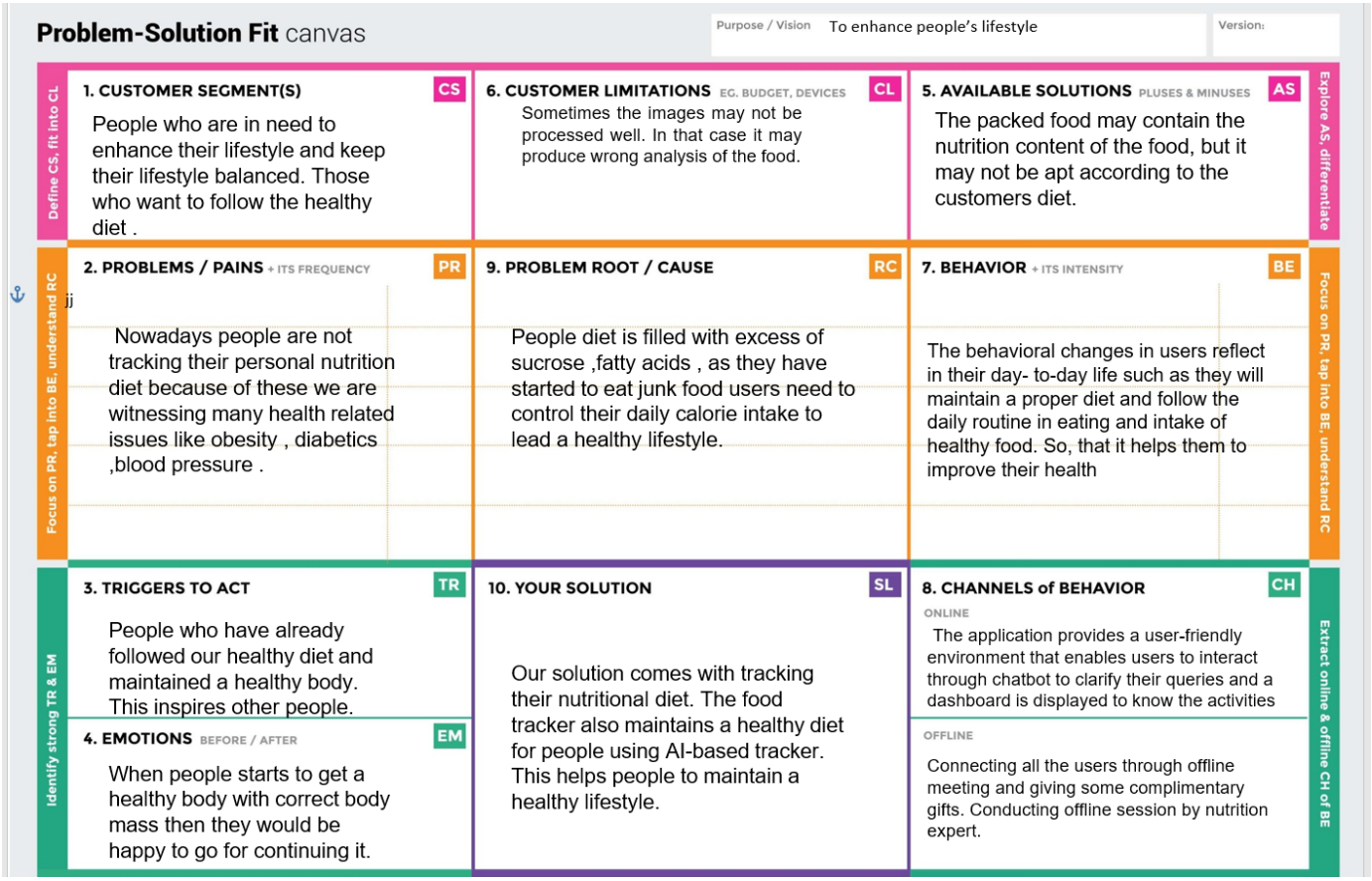


3.3 Proposed Solution

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 classifying the fruit depending on the different characteristics like colour, shape, texture etc. • Here the user can capture images of different fruits and then the image will be sent to the trained model. • The model analyses the image and detect the nutrition based on the fruits like (Sugar, Fibre, Protein, Calories, etc.).
2.	Idea / Solution description	• The user logins to the system using the login credentials. • The user interacts with the UI (User Interface) and gives the image as input. • Then the input image is passed to the flask application. • Next is to develop an AI model to capture the image and recognize the fruit and find its nutritional values. • And finally with the help of the model, we build we will classify the result and showcase it on the UI.

3.	Novelty / Uniqueness	• This model can capture images and detect the nutritional values of the fruits and showcase them to the user. • The model collects the user input and prepares diet charts for the user. • Personalized fitness and diet plans for the user.
4.	Social Impact / Customer Satisfaction	• Helps fitness enthusiasts to find the nutritional value of the fruits they're consuming which will help them to maintain a balanced diet by taking adequate amount of nutrients needed every day. • To consult with the nutritionist and personal trainers for their diet plans and training schedules. • Generalised diet and fitness plans to all users based on their goals.
5.	Business Model (Revenue Model)	• Specific diet and fitness plans according to their body type and their goals for the premium subscription members. • Personal trainers to help fitness enthusiasts with their workouts. • Personal nutritionist for fitness enthusiasts to help with diet plans based on their fitness goals.
6.	Scalability of the Solution	• The solution is developed in such a way that we can update the system without disturbing the current model. • All future enhancements can be added to the system without changing the model.

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

FUNCTIONAL REQUIREMENT:

FOLLOWING ARE THE FUNCTIONAL REQUIREMENTS OF THE PROPOSED SOLUTION.

FUNCTION NUMBER	FUNCTIONAL REQUIREMENT	SUB REQUIREMENT
FN.NO-1	USER REGISTRATION/LOGIN	REGISTER AFTER OPENING THE APPLICATION
FN.NO-2	USER DETAIL	PROVIDE DETAIL ABOUT HEALTH STATUS AFTER REGISTRATION BY SELECTING AND SPECIFY ABOUT IT
FN.NO-3	CAPTURING IMAGE	CAPTURE THE IMAGE AND CHECK THE PARAMETERS OF THE CAPTURED IMAGE.
FN.NO-4	IMAGE PROCESSING	UPLOAD THE IMAGE FOR PROCESSING
FN.NO-5	IMAGE IDENTIFICATION	IDENTIFY THE FOOD OR DRINK PROVIDED IN THEIMAGE.
FN.NO-6	IMAGE DESCRIPTION	PROVIDE THE NUTRITIONAL CONTENTS OF THEFOOD OR DRINK IDENTIFIED.

4.2 Non-Functional Requirements:

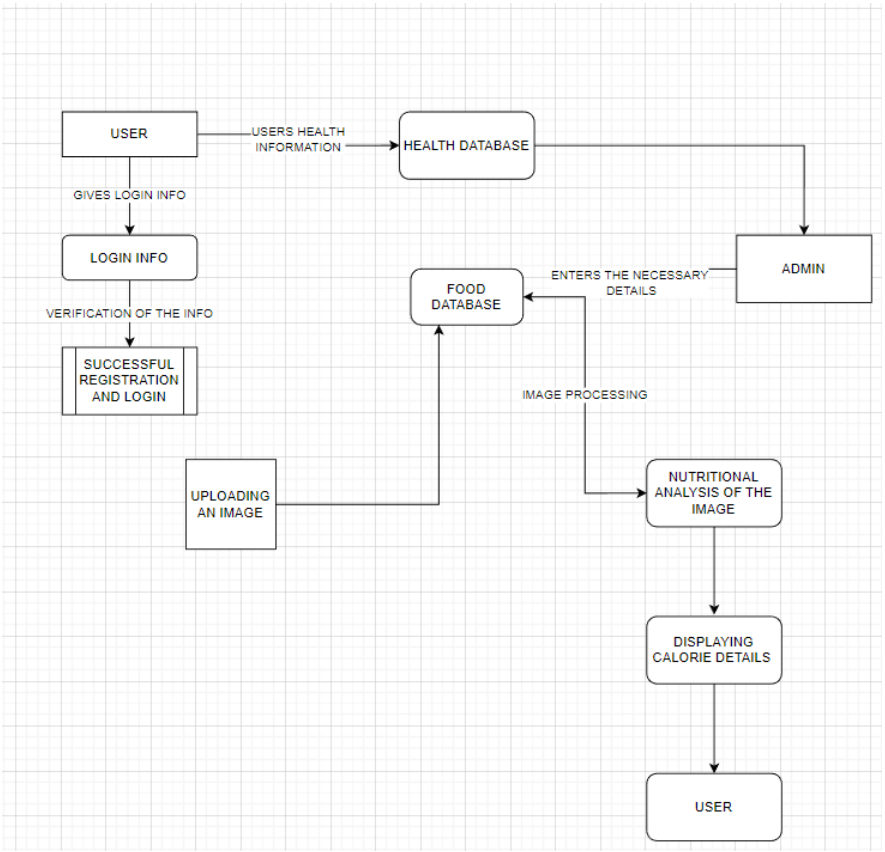
FOLLOWING ARE THE NON-FUNCTIONAL REQUIREMENT OF THE PROPOSEDSOLUTION

NON - FUNCTION NUMBER	NON-FUNCTIONAL REQUIREMENT	SUB REQUIREMENT
NFN.NO-1	USABILITY	DATASET OF ALL FOOD/DRINK TO IDENTIFY THE NUTRITIONAL CONTENT OF THE ITEM.
NFN.NO-2	SECURITY	THE INFORMATION AND THE HEALTH STATUS OF USERS AND NUTRITIONAL DETAILS ABOUT FOOD ARE SECURED HIGHLY.
NFN.NO-3	RELIABILITY	THE IMAGE QUALITY IS IMPORTANT TO PROVIDE THE NUTRITIONAL DETAILS ABOUT FOOD
NFN.NO-4	PERFORMANCE	PERFORMANCE IS BASED ON THE FOOD THAT ISSCANED.
NFN.NO-5	AVAILABILITY	IT IS AVAILABLE FOR ALL THE USERS TO DETECT THE NUTRITIONAL DETAILS OF ABOUT FOOD / DRINK.
NFN.NO-6	SCALABILITY	INCREASE THE PREDICTION OF NUTRITIONALDETAILS IN THE FOOD.

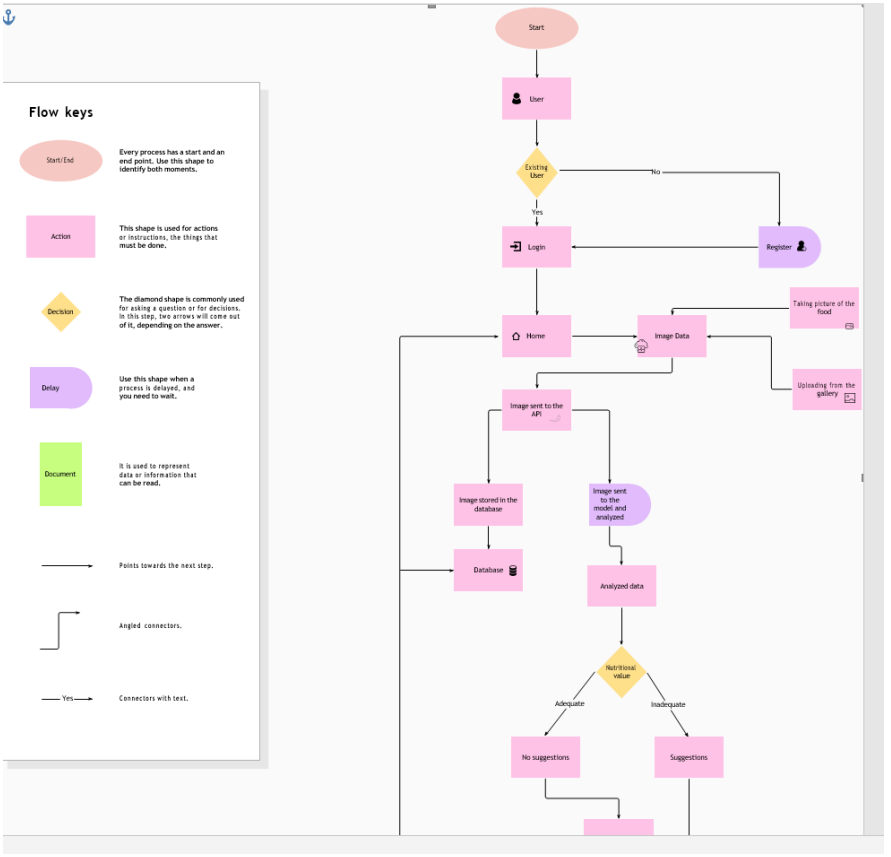
5. PROJECT DESIGN

5.1 Data Flow Diagrams

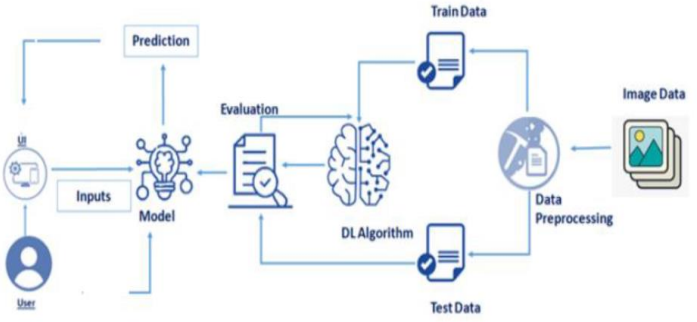
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the rightamount of the system requirement graphically. It shows how data enters andleaves the system, what changes the information, and where data is stored.



5.2 Solution And Technical Architecture



The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



5.3 User Stories

They were so excited to use this application because it balanced a healthy lifestyle. They were able to track their health status. And maintain a healthy balanced body.

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning And Estimation

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	25 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	26 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	30 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	3 OCTOBER 2022

Problem Solution Fit	Prepare problem - solution fit document.	4 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	4 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	7 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	13 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	14 OCTOBER 2022
Technology Architecture	Prepare th e technology architecture diagram.	WORK IN PROGRESS
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	31 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	WORK IN PROGRESS

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Download the food nutrition dataset	2	High	Jai Siva Ranjani
Sprint-1	Data Preprocessing	USN-2	Importing the Dataset into Workspace	1	Medium	Jai Siva Ranjani
Sprint-1		USN-3	Handling Missing data	3	Low	Jai Siva Ranjani
Sprint-1		USN-4	Feature Scaling	3	Medium	Jai Siva Ranjani
Sprint-1		USN-5	Data Visualization	3	Low	Jai Siva Ranjani
Sprint-1		USN-6	Splitting Data into Train and set	4	High	Jai Siva Ranjani
Sprint-1		USN-7	Creating A Dataset with Sliding Windows	4	Medium	Jai Siva Ranjani
Sprint-2	Model Building	USN-8	Importing The Model	1	HIGH	Ishwarya

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			Building Libraries			
Sprint-2		USN-9	InitializingThe Model	1	Medium	Ishwarya
Sprint-2		USN-10	Adding CNN Layers	2	High	Ishwarya
Sprint-2		USN-11	Adding Dense Layers	3	Low	Ishwarya
Sprint-2		USN-12	ConfigureThe Learning Process	4	Medium	Ishwarya
Sprint-2		USN-13	Train the model	2	Medium	Ishwarya
Sprint-2		USN-14	Save the model	2	Medium	Ishwarya
Sprint-2		USN-15	Test the model	3	High	Ishwarya
Sprint-3	Application Building	USN-16	Create an HTML file	4	Medium	Lavanya
Sprint-3		USN-17	Build Python code	4	High	Manasa
Sprint-3		USN-18	Creating our flask application & loading our model using local model method	4	Medium	Manasa

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3		USN-19	Run the application	4	High	Manasa , Lavanya
Sprint-4	Train the model on IBM	USN-20	Register for IBM Cloud	4	Medium	Jai SivaRanjani, Ishwarya, Lavanya, Manasa
Sprint-4		USN-21	Train the ML Model on IBM	4	High	Manasa
Sprint-4		USN-22	Integrate Flask with scoring End Point	8	High	Manasa

Project Tracker, Velocity & Burndown Chart:

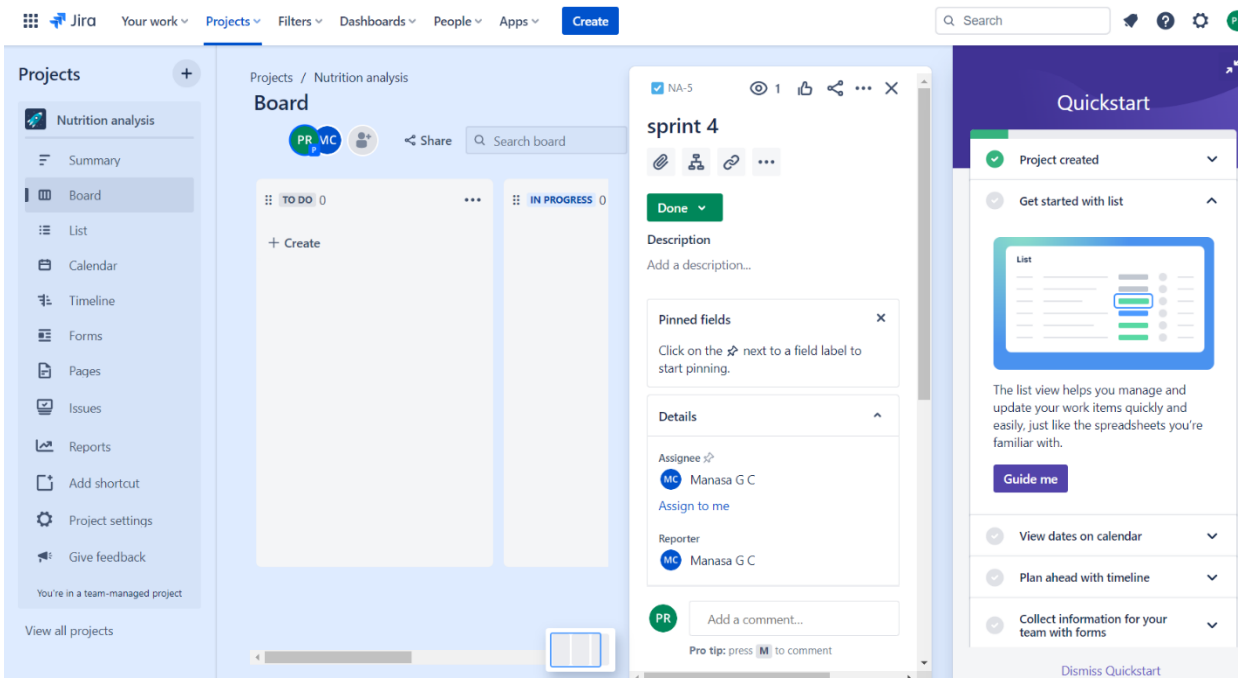
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

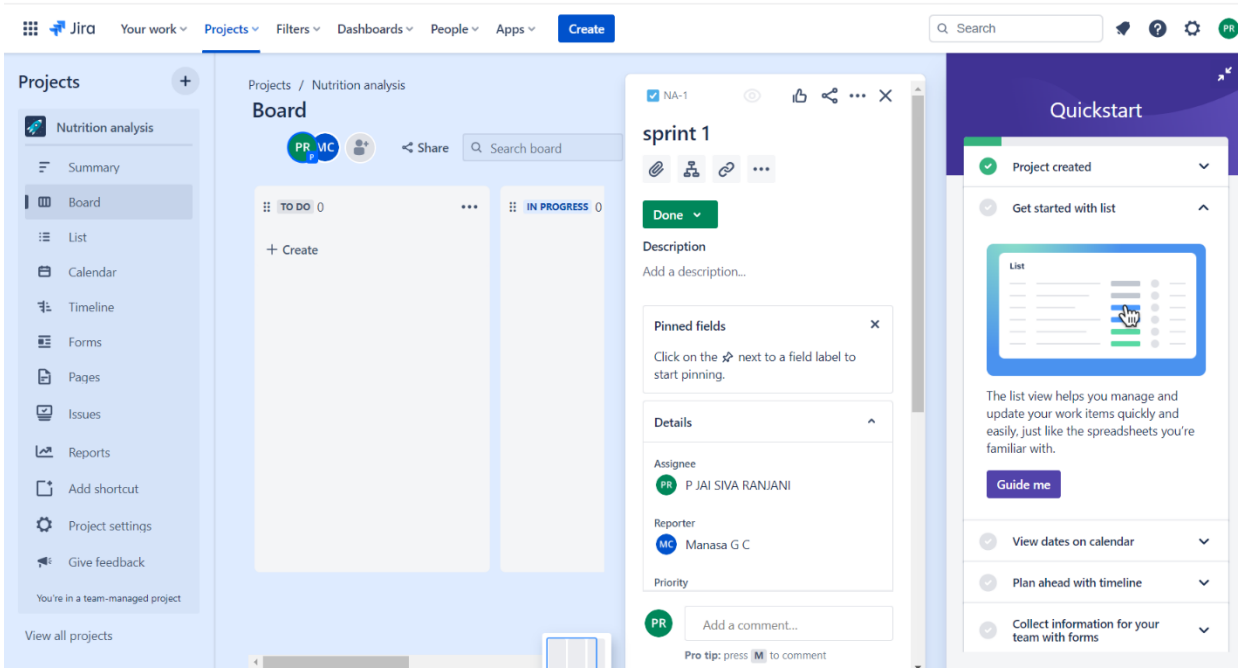
Velocity:

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

6.3 Reports From Jira

Jira Software is part of a family of products designed to **help teams of all types manage work**. Originally, Jira was designed as a bug and issue tracker. But today, Jira has evolved into a powerful work management tool for all kinds of use cases, from requirements and test case management to agile software development.



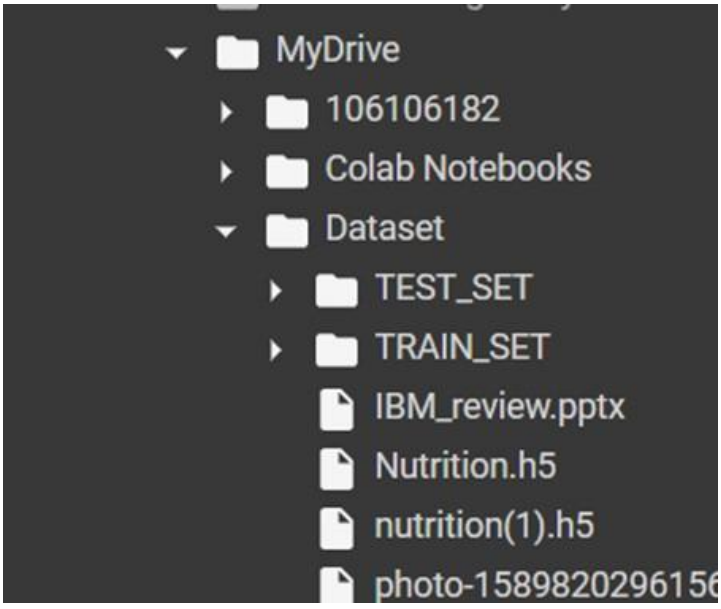


7. CODING AND SOLUTIONING

Data Collection

Data Collection Collect images of different food items organized intosubdirectories based on their respective names as shown in the project structure. Create folders of types of food items that need to be recognized. In this project, we have collected images of 5 types of food items apples, 'banana', 'orange', they are saved in the respective subdirectories with their respective names.

DATASET LINK:
https://drive.google.com/drive/folders/1yNVuLA2hxIstOcDV58enyD74Y9drEs6Y?usp=sharing



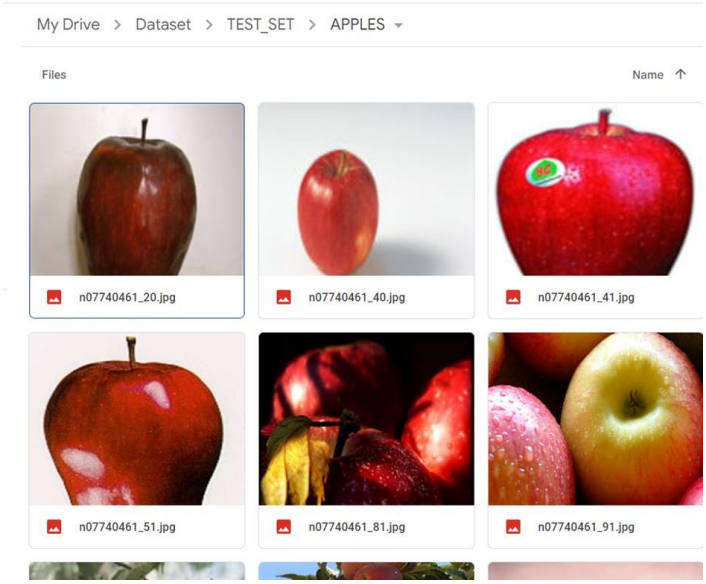

My Drive > Dataset > TEST_SET > APPLES
Files
Name
n07740461_20.jpg
n07740461_40.jpg
n07740461_41.jpg
n07740461_51.jpg
n07740461_81.jpg
n07740461_91.jpg

Image Preprocessing

We will be improve the image data that suppresses unwilling distortions or enhances some image features important for further processing, although performing some geometric transformations of images like rotation, scaling, translation, etc.

1)Import The ImageDataGenerator Library

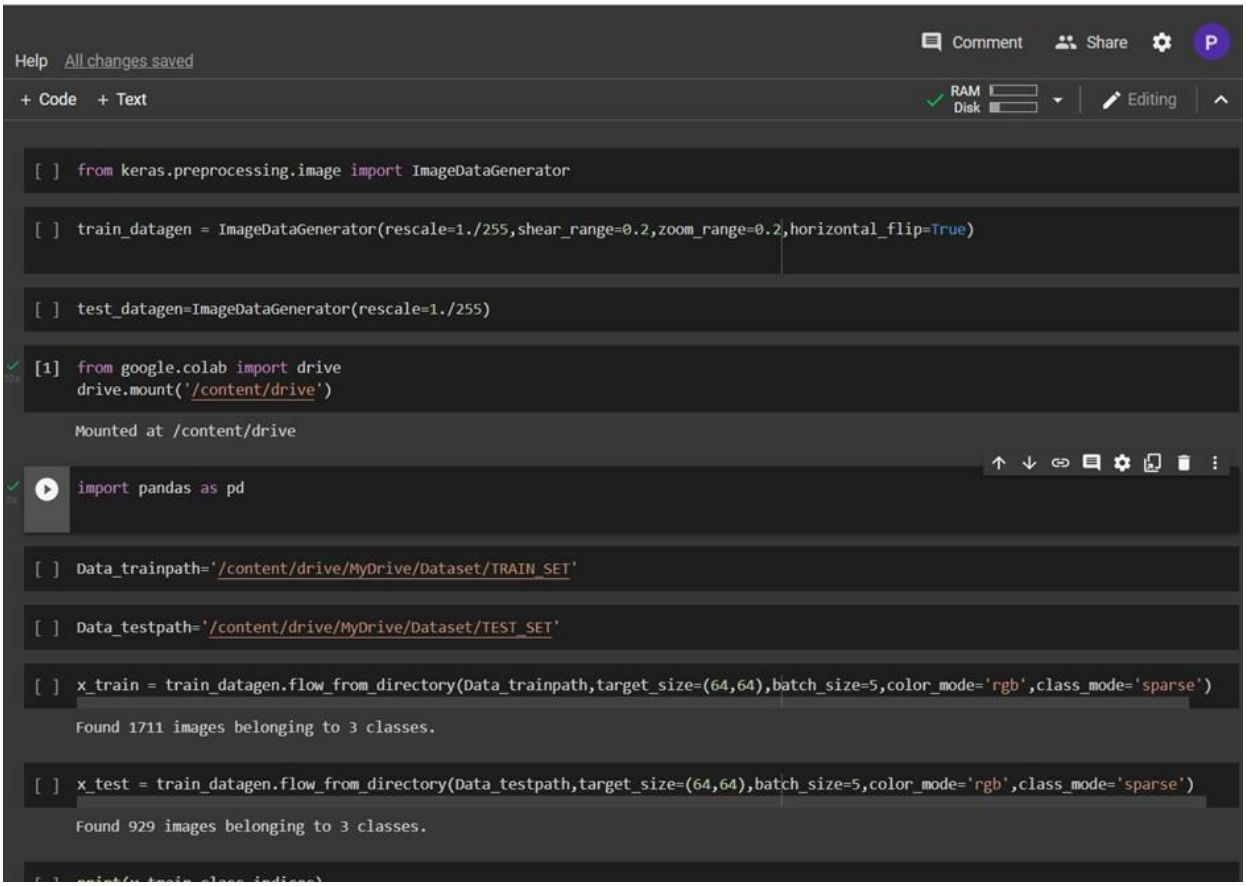
2)Configure ImageDataGenerator Class

The Keras deep learning neural network library provides the capability to fit modelsusing image data augmentation via the ImageDataGenerator class.
Let us import the ImageDataGenerator class from Keras

There are five main types of data augmentation techniques for image data; specifically:

Image shifts via the width_shift_range and height_shift_range arguments.The image flips via the horizontal_flip and vertical_flip arguments.
Image rotations via the rotation_range argument
Image brightness via the brightness_range argument. Image zoom via the zoom_range argument.

Let us apply ImageDataGenerator functionality to Trainset and Testset by using thefollowing code
For Training set using flow_from_directory function.



Model Building

Here we are going to build our Convolutional Neural Networking which contains an input layer along with the convolution, max- pooling, and finally an output layer.

This is an important step because the model is the main thingneeded for prediction.

Importing The Model Building Libraries

We have imported all the necessary libraries needed for model building.

Importing The Model Building Libraries

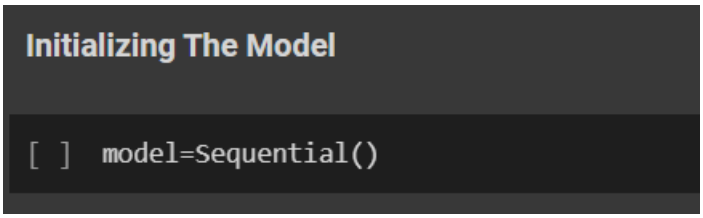
```
[ ] import numpy as np
    import tensorflow
    from tensorflow.keras.models import Sequential
    from tensorflow.keras import layers
    from tensorflow.keras.layers import Dense,Flatten
    from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout
    from keras.preprocessing.image import ImageDataGenerator
```

Initializing The Model

Keras has 2 ways to define a neural network:

- Sequential
- Function API

The Sequential class is used to define linear initializations of network layers which then, collectively, constitute a model. We will use the Sequential constructor to create our model, which will then have layers added to it using the add() method.



Adding CNN Layers

- As the input image contains three channels, we are specifying the input shape as (64,64,3).
- We are adding a two-convolution layer with an activation function as “relu” and with a small filter size (3,3) and the number of filters (32) followed by a max-pooling layer.
- Max pool layer is used to downsample the input (Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter)
- Flatten layer flattens the input. Does not affect the batch size.

Adding Dense Layers

- A dense layer is a deeply connected neural network layer. It is the most common and frequently used layer.
- The number of neurons in the Dense layer is the same as the number of classes in the training set. The neurons in the last Dense layer, use softmax activation to convert their outputs into respective probabilities.
- Understanding the model is a very important phase to properly use it for training and prediction purposes. Keras provides a simple method, and a summary to get the full information about the model and its layers.

Adding Dense Layers

```
[ ]  model.add(Dense(units=128, activation='relu'))
     model.add(Dense(units=5, activation='softmax'))
```

```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten (Flatten)           (None, 6272)              0

 dense (Dense)               (None, 128)               802944

 dense_1 (Dense)             (None, 5)                 645

=================================================================
Total params: 813,733
Trainable params: 813,733
Non-trainable params: 0
_________________________________________________________________
```

Configure The Learning Process

- The compilation is the final step in creating a model. Once the compilation is done, we can move on to the training phase. The loss function is used to find errors or deviations in the learning process. Keras requires a loss function during the model compilation process.
- Optimization is an important process that optimizes the input weights by comparing the prediction and the loss function. Here we are using adam optimizer.
- Metrics are used to evaluate the performance of your model. Itis similar to the loss function, but not used in the training process.

Configure The Learning Process

```
[ ]  model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

Train The Model

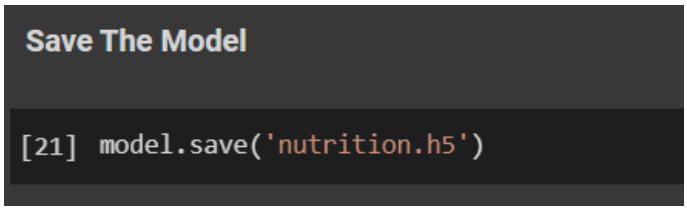
Now, let us train our model with our image dataset. The model is trained for 15 epochs and after every epoch, the current model state is saved if the model has the least loss encountered till that time. We can see that the training loss decreases in almost every epoch till 15 epochs and probably there is further scope to improve the model.

fit_generator functions used to train a deep-learning neural network Arguments:

- steps_per_epoch: it specifies the total number of steps taken from the generator as soon as one epoch is finished and the next epoch has started. We can calculate the value of steps_per_epoch as the total number of samples in your dataset divided by the batch size.
- Epochs: an integer and number of epochs we want to train our model for.
- validation_data can be either:
  - an inputs and targets list
  - a generator
  - inputs, targets, and sample_weights list which can be used to evaluate the loss and metrics for any model after any epoch has ended.
- validation_steps: only if the validation_data is a generator then only this argument can be used. It specifies the total number of steps taken from the generator before it is stopped at every epoch and its value is calculated as the total number of validation data points in your dataset divided by the validation batch size.

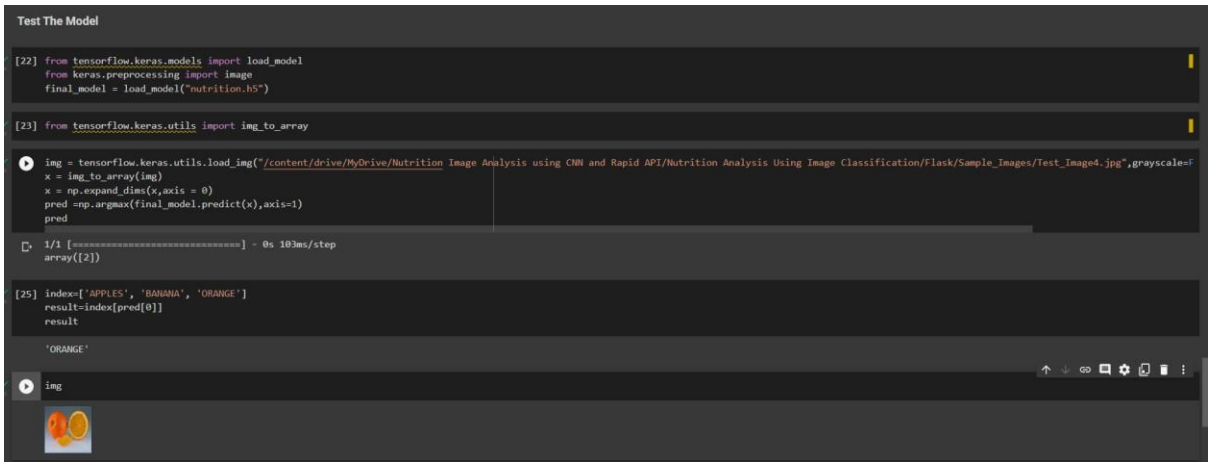
Save The Model

The model is saved with .h5 extension as follows
An H5 file is a data file saved in the Hierarchical Data Format (HDF). Itcontains multidimensional arrays of scientific data.



Test The Model

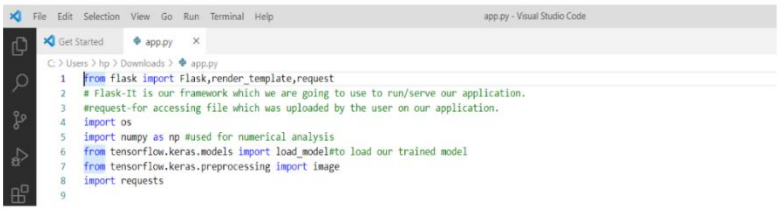
- Evaluation is a process during the development of the model to check whether the model is the best fit for thegiven problem and corresponding data.
- Load the saved model using load_model
- Taking an image as input and checking the results
- By using the model we are predicting the output for thegiven input image
- The predicted class index name will be printed here.



Application Building

Build Python Code:

Importing the flask module into the project is mandatory. An object of the Flask class is our WSGI application. Flask constructor takes the name of the current module (__name__) as an argument Pickle library to load the model file.

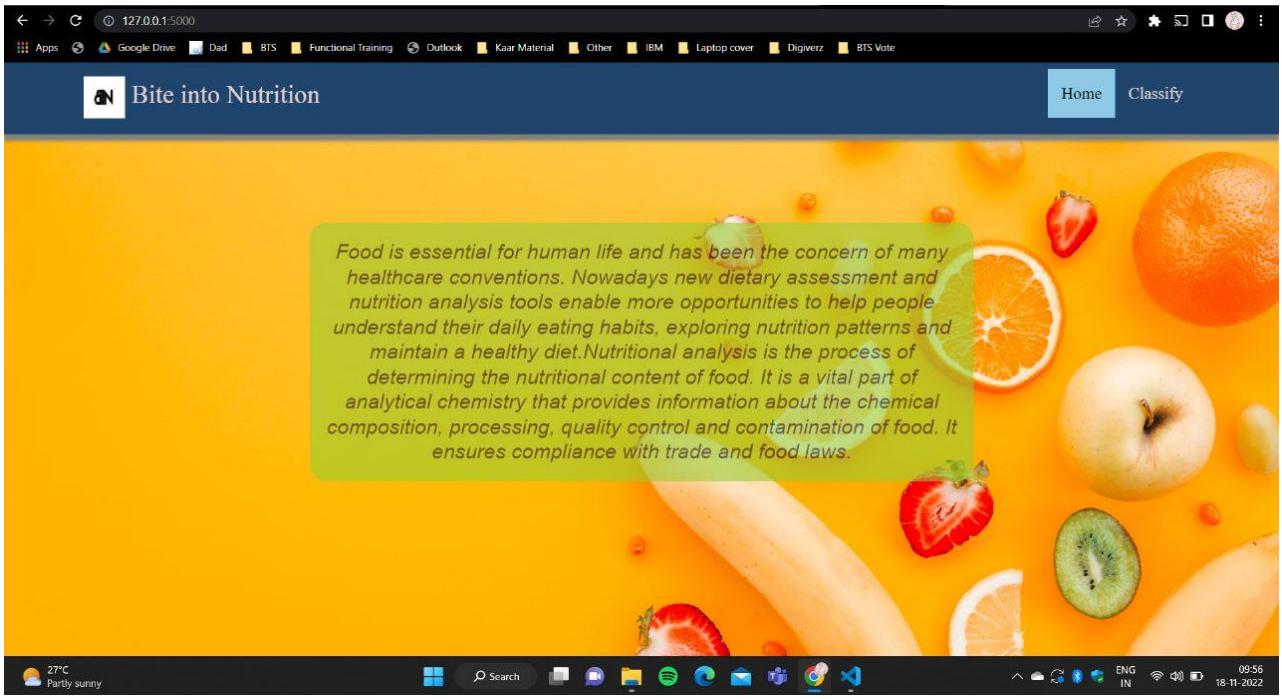


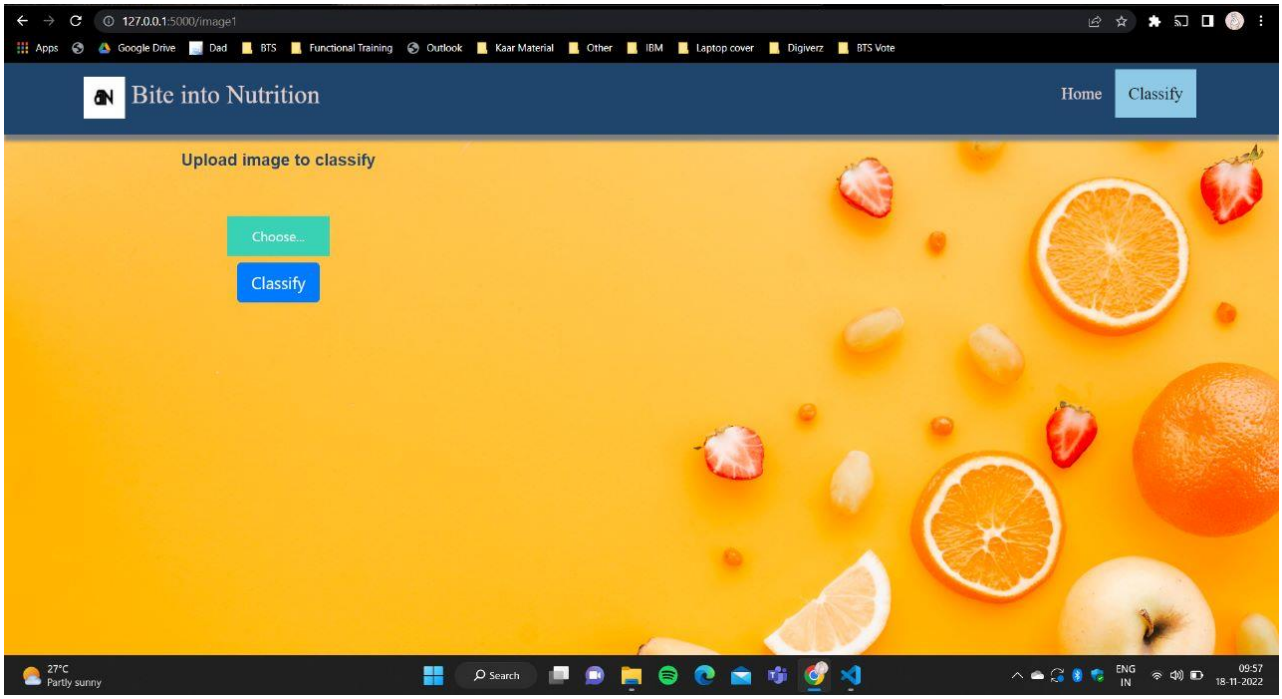
Creating Our Flask Application And Loading Our Model By Using Load_model Method

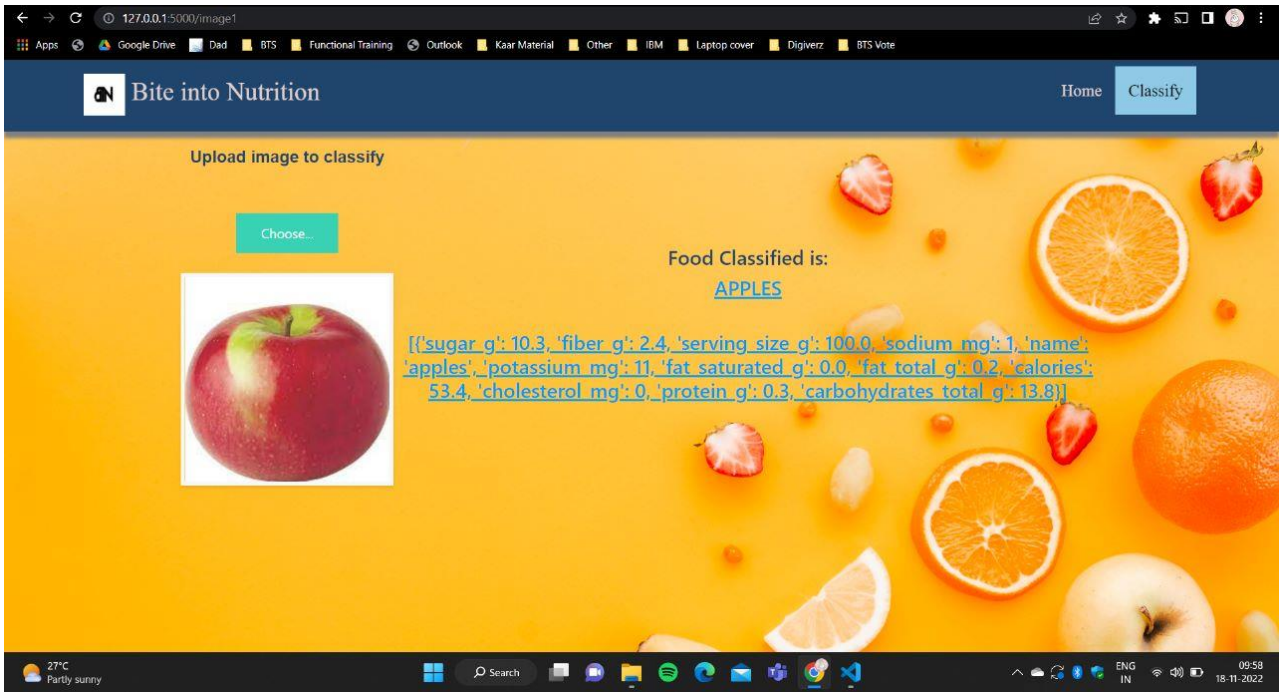
Creating our flask application and loading our model by using the load_model method

```
9
10
11  app = Flask(__name__,template_folder="templates") # initializing a flask app
12  # Loading the model
13  model=load_model('nutrition.h5')
14  print("Loaded model from disk")
15
16  @app.route('/')# route to display the home page
17  def home():
18      return render_template('home.html')#rendering the home page
19
20  @app.route('/image1',methods=['GET','POST'])# routes to the index html
21  def image1():
22          return render_template("image.html")
23
24  @app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
25  def launch():
26      if request.method=='POST':
27          f=request.files['file'] #requesting the file
28          basepath=os.path.dirname('__file__')#storing the file directory
29              filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in uploads folder
30              f.save(filepath)#saving the file
31
32
33              img=image.load_img(filepath,target_size=(64,64)) #load and reshaping the image
34              x=image.img_to_array(img)#converting image to an array
35              x=np.expand_dims(x,axis=0)#changing the dimensions of the image
```

Creating Html Pages And Routing







Routing To The Html Page

Here, the declared constructor is used to route to the HTML page created earlier.

In the above example, the '/' URL is bound with the home.html function. Hence, when the home page of the web server is opened in the browser, the HTML page is rendered. Whenever you enter the values from the HTML page the values can be retrieved using the POST Method.

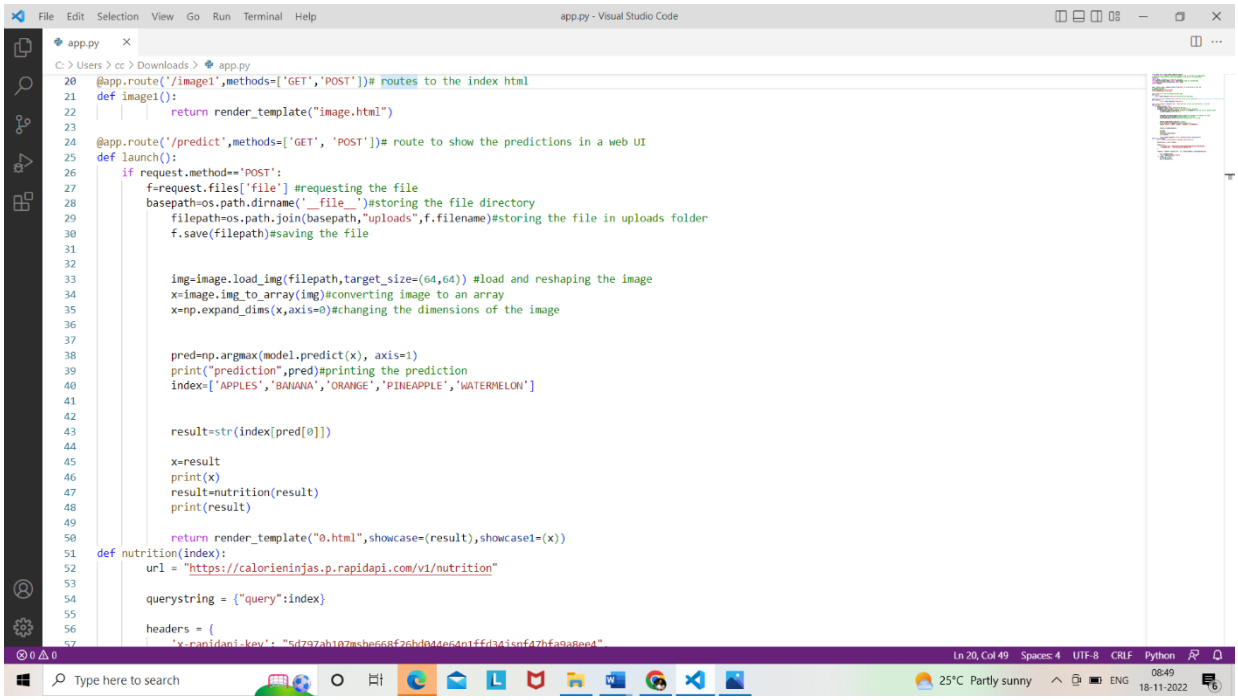
Here, "home.html" is rendered when the home button is clicked on the UI

```
@app.route('/')# route to display the home page
def home():
    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")
```

When “image is uploaded “on the UI, the launch function is executed

```
@app.route('/predict',methods=['GET', 'POST'])# route to show the predictions in a web UI
def launch():
    if request.method=='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
```

It will take the image request and we will be storing that image in our local system then we will convert the image into our required size and finally, we will be predicting the results with the help of our model which we trained and depending upon the class identified we will showcase the class name and its properties by rendering the respective HTML pages.



API Integration:

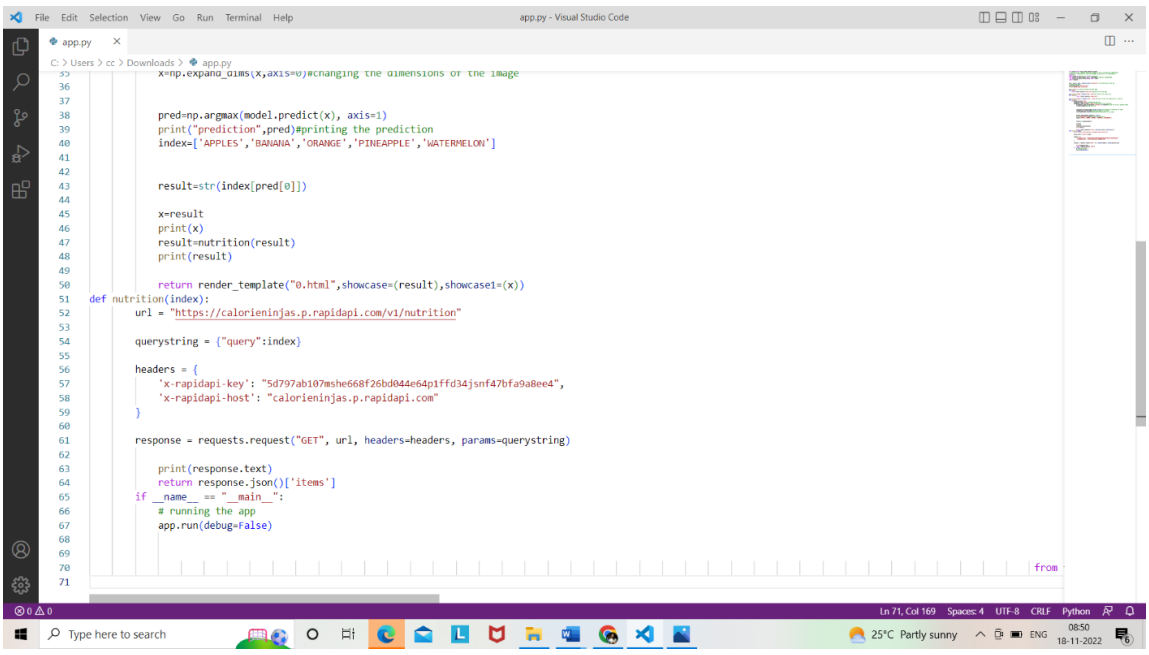
Here we will be using Rapid API

Using RapidAPI, developers can search and test the APIs, subscribe, and connect to the APIs — all with a single account, single API key and single SDK. Engineering teams also use RapidAPI to share internal APIs and microservice documentation.

Reference link

API used: Link
The link above will allow us to test the food item and will result in the nutrition content present in the food item.
NOTE: When we keep hitting the API the limit of it might expire. So making smart use of it will be an efficient way.

How to access and use the API will be shown in this video

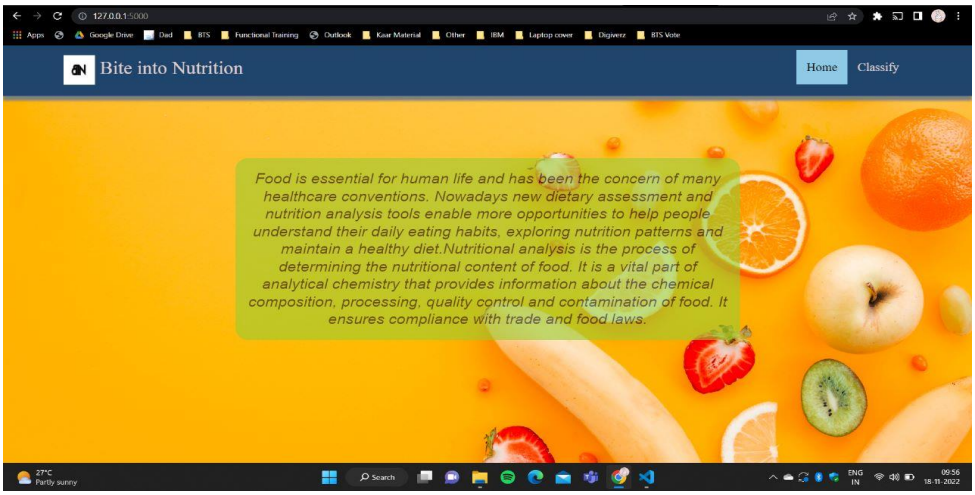


Finally, Run the application. This is used to run the application in a localhost. The local host runs on port number 5000. (We can give different port numbers)

Run The Application

- In Visual Studio opens the integrated terminal.
- Now type the “python app.py” command.
- It will show the local host where your app is running on http://127.0.0.1.5000/
- Ctr+Click the URL. It does navigate to where you can view your web page.
- Enter the values, click on the predict button and see the result/prediction on the web page.
- Then it will run on localhost:5000

```
PS D:\IBM Proj\Nutrition Analyzer> python app.py
2022-11-18 09:30:57.921283: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudart64_110.dll
 not found
2022-11-18 09:30:57.922769: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-11-18 09:31:01.134670: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2022-11-18 09:31:01.135439: W tensorflow/stream_executor/cuda/cuda_driver.cc:326] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-18 09:31:01.142146: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: Ishwarya
2022-11-18 09:31:01.143014: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: Ishwarya
2022-11-18 09:31:01.144128: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to
 use the following CPU instructions in performance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Loaded model from disk
 * Serving Flask app 'app'
 * Debug mode: off
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
127.0.0.1 - - [18/Nov/2022 09:31:06] "GET / HTTP/1.1" 200 -
```

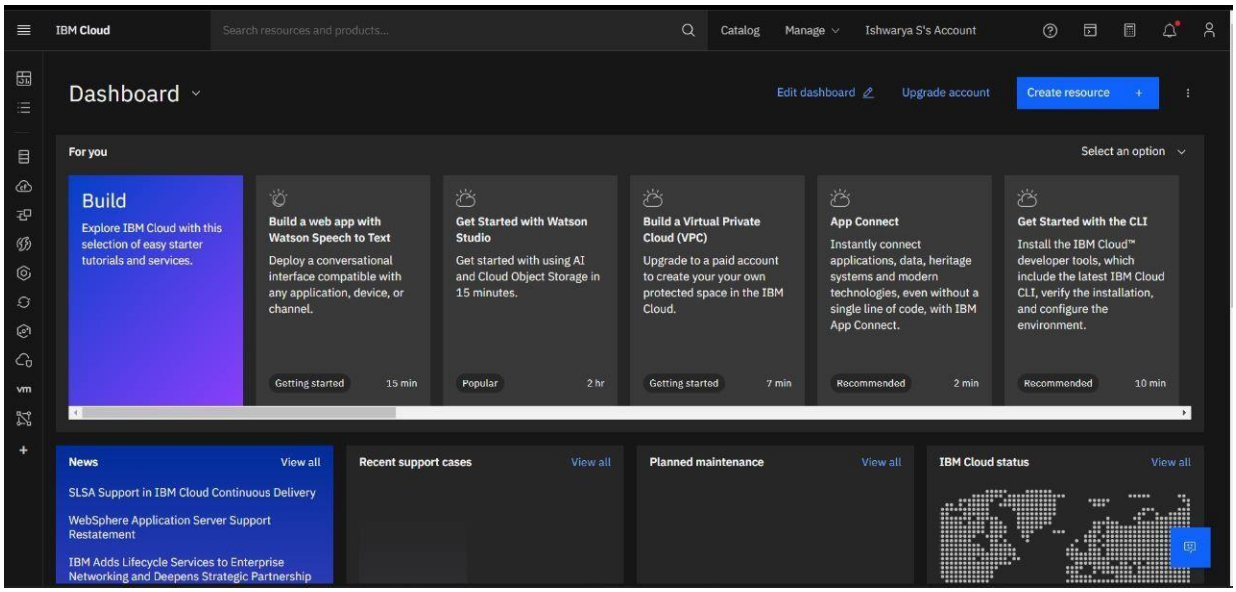


Register For IBM Cloud

IBM Account

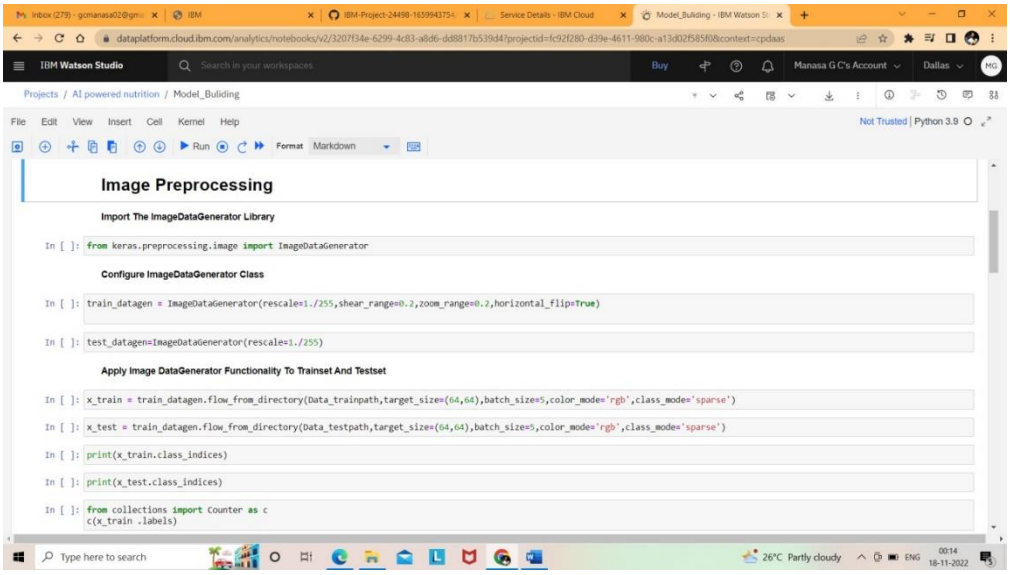
Register link: https://www.ibm.com/academic/home

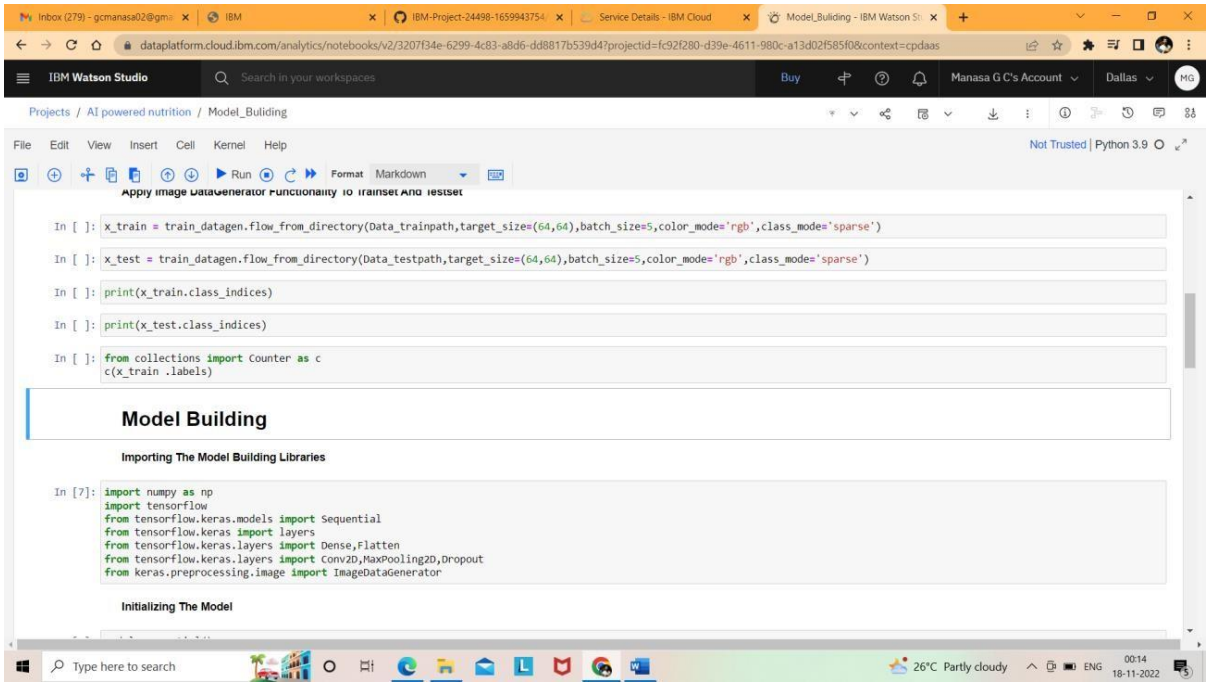
Log in: https://cloud.ibm.com/



Train Model On IBM

Log in to the IBM cloud and create an IBM Watson studio to train the model onIBM. Copy the file, paste it and run the file in a notebook.





Apply Image DataGenerator Functionality To Trainset And Testset

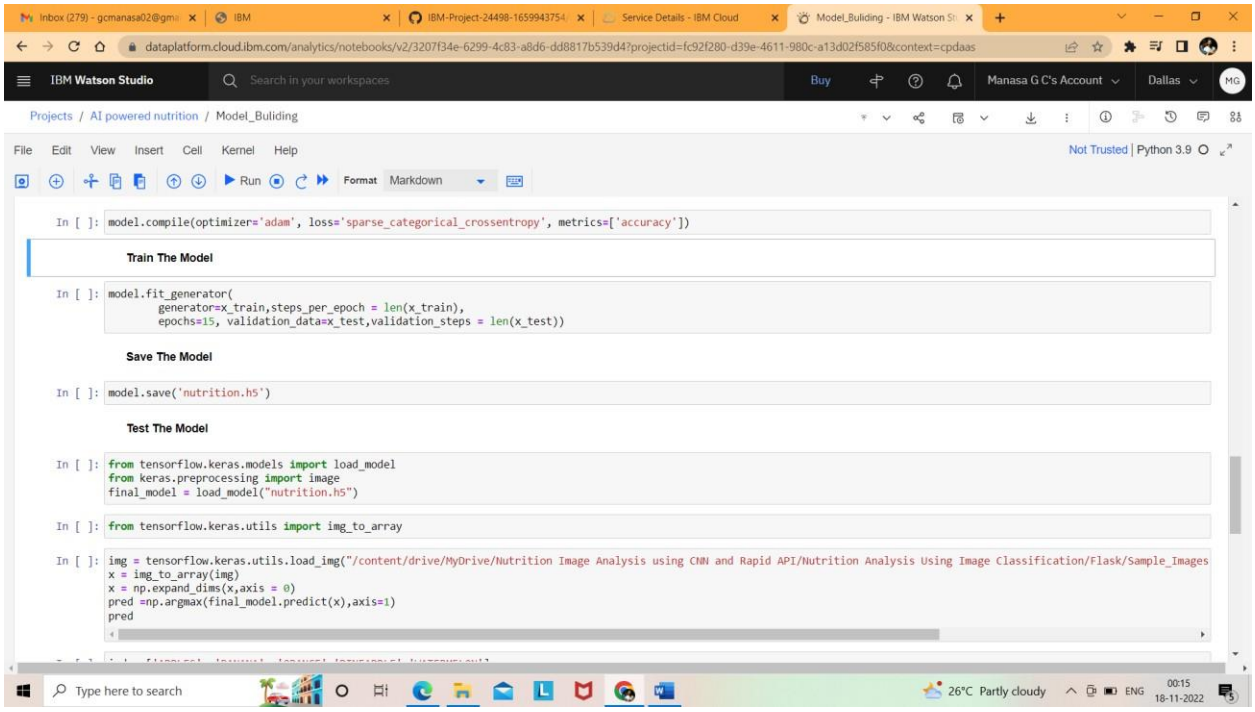
```
In [ ]: x_train = train_datagen.flow_from_directory(Data_trainpath,target_size=(64,64),batch_size=5,color_mode='rgb',class_mode='sparse')

In [ ]: x_test = train_datagen.flow_from_directory(Data_testpath,target_size=(64,64),batch_size=5,color_mode='rgb',class_mode='sparse')

In [ ]: print(x_train.class_indices)

In [ ]: print(x_test.class_indices)

In [ ]: from collections import Counter as c
        c(x_train .labels)
```

Model Building

Importing The Model Building Libraries

```
In [7]: import numpy as np
        import tensorflow
        from tensorflow.keras.models import Sequential
        from tensorflow.keras import layers
        from tensorflow.keras.layers import Dense,Flatten
        from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout
        from keras.preprocessing.image import ImageDataGenerator
```

Initializing The Model



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

Train The Model

```
In [ ]: model.fit_generator(
              generator=x_train,steps_per_epoch = len(x_train),
              epochs=15, validation_data=x_test,validation_steps = len(x_test))
```

Save The Model

```
In [ ]: model.save('nutrition.h5')
```

Test The Model

```
In [ ]: from tensorflow.keras.models import load_model
        from keras.preprocessing import image
        final_model = load_model("nutrition.h5")

In [ ]: from tensorflow.keras.utils import img_to_array

In [ ]: img = tensorflow.keras.utils.load_img("/content/drive/MyDrive/Nutrition Image Analysis using CNN and Rapid API/Nutrition Analysis Using Image Classification/Flask/Sample_Images
        x = img_to_array(img)
        x = np.expand_dims(x,axis = 0)
        pred =np.argmax(final_model.predict(x),axis=1)
        pred
```

8. TESTING

8.1 Test Cases

A test case is a singular set of actions or instructions for a tester to perform that validates a specific aspect of a product or application functionality**. If the test fails, the result might be a software defect that the organization can triage.**

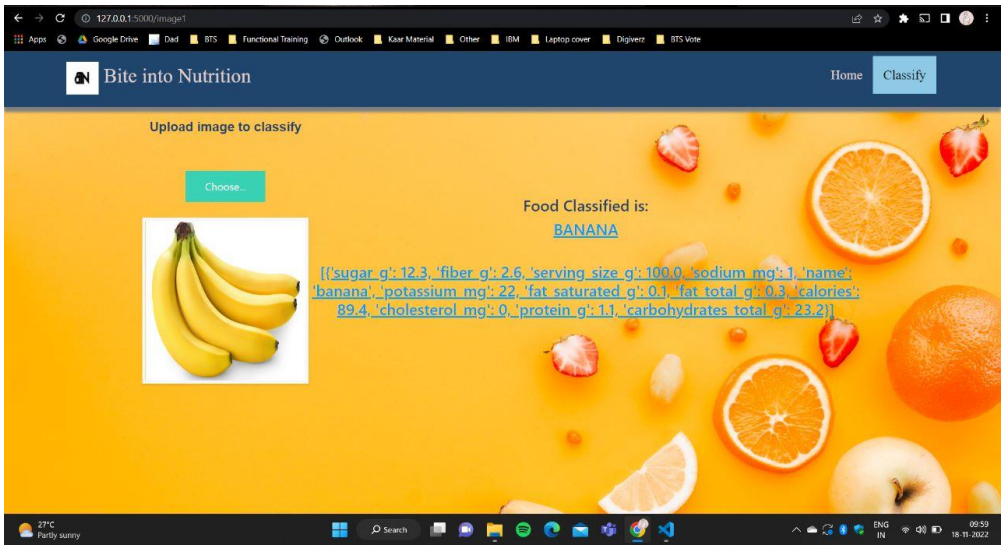
1. Create test cases that are as simple as possible.

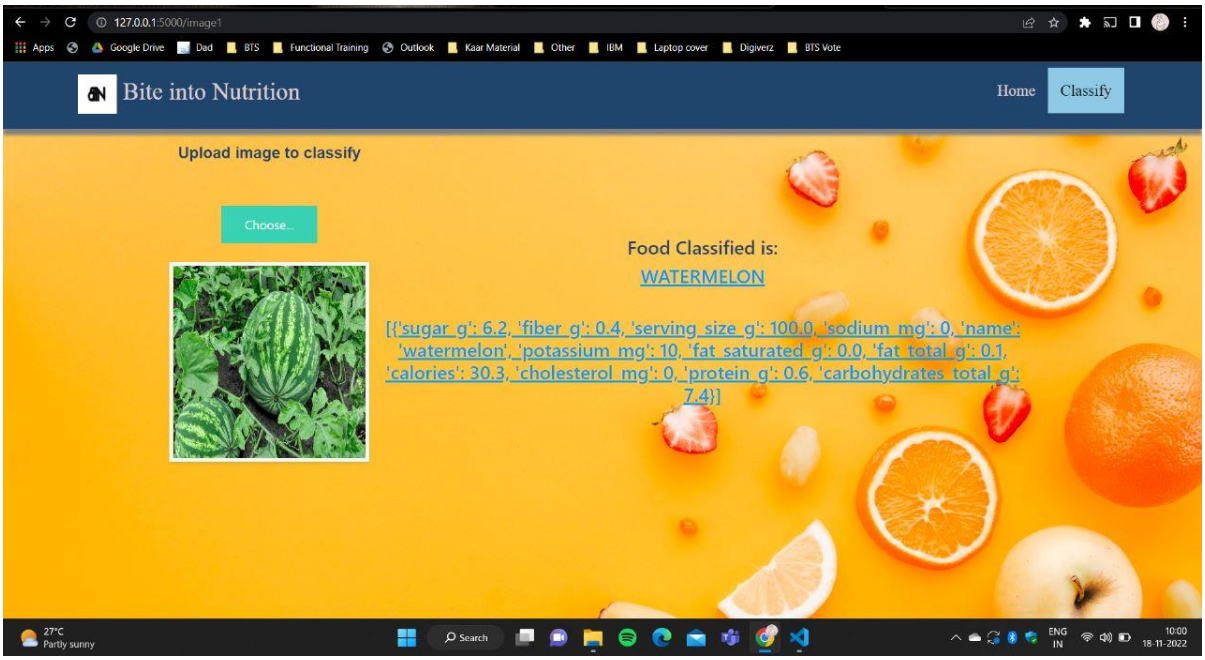
2. Create a Test Case with the End User in Mind

3. Do not repeat test cases

8.2 User Acceptance Testing

User Acceptance Testing (UAT), which is performed on most UIT projects, sometimes called beta testing or end-user testing, is **a phase of software development in which the software is tested in the "real world" by the intended audience or business representative**.




127.0.0.1:5000/image1
Apps
Google Drive
Dad
BTS
Functional Training
Outlook
Kaar Material
Other
IBM
Laptop cover
Digiverz
BTS Vote
Bite into Nutrition
Home
Classify
Upload image to classify
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}]
27°C
Partly sunny
Search
ENG
IN
10:00
18-11-2022

9. RESULTS

9.1 Performance Metrics

Performance measurement is the process of collecting, analyzing and/or reporting information regarding the performance of an individual, group, organization, system or component. Definitions of performance measurement tend to be predicated upon an assumption about why the performance is being measured**.**

Confusion Matrix

	precision	recall	f1-score	support
0	1.00	1.00	1.00	157
1	1.00	1.00	1.00	287
2	1.00	1.00	1.00	160
3	1.00	1.00	1.00	294
4	1.00	1.00	1.00	157
accuracy			1.00	1055
macro avg	1.00	1.00	1.00	1055
weighted avg	1.00	1.00	1.00	1055

This is used to find the measures like:

Accuracy

Precision

Recall

Specificity

F1-Score

10. ADVANTAGES AND DISADVANTAGES

Advantages:

It maintains a healthy lifestyle for a human. It calculates the BMI (Body mass index). It regulates our body's blood pressure level. It controls the weight of the human body.

Because of today's lifestyle, people tend to eat more junk which causes serious health-related disorders such as diabetes, blood pressure obesity. They can be prevented if we track our nutrition intake level.

Disadvantages:

Our health report is completely exposed to others. There may be security-level problems in this application. Because hackers can easily track our health status if they break all the security barriers in the application. They may even use to kill a person if needed.

Security of data is a concern in this type of application.

11. CONCLUSION

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, explore nutrition patterns and maintain 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.

Our project plays a vital role in maintaining human’s healthy lifestyles by analysing their nutrition intake. And keeps track of their health status in a very serious manner. It helps in avoiding serious health diseases. So that a person’s life can be saved.

12. FUTURE SCOPE

In future or in the upcoming days, many people may use this app for their personal assistance with nutrition intake and to maintain a healthy lifestyle. This application has a major scope in future. We may include even pulse rate, heart rhythm, and kidney functionality in the near future in our application

13. APPENDIX

Source Code

home.html

```
<body style=" background-image: url(/static/css/homebg.jpg);">
<div class="header">
<div style="width:50%;float:left;font-size:2vw;text-
align:left;color:rgb(226, 206, 206); padding-top:1%;padding-
left:5%;">
<img src="static\css\logo.jpg" height="50px" width="50px">
Bite into Nutrition</div>
<div class="topnav-right"style="padding-top:0.5%;">
<a class="active" href="{{ url_for('home')}}">Home</a>
<a   href="{{ url_for('image1')}}">Classify</a>
</div>
</div>
</div>
```

imageprediction.html

```
<div class="container">
<center>
<div id="content" style="margin-top:2em">{% block content
%}{% endblock%}</div></center>
</div>

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

image.html

```
{% extends "imageprediction.html" %} {% block content %}
<div style="float:left">
<br>
<br>
<h5 style=" color:#1f456d; size:10; 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>
{% endblock %}
```

0.html

```
<div class="results">
<p style="padding-top: 150px; color:#1f456d ;"><h4
style="color:#1f456d ;">Food Classified is: <h4><b><h4
style="color:#009cff;"><u>{{showcase1}}<h4><br><h4
style="color:#009cff;"><u>{{showcase}}<h4></p>
 </div>
```

main.css

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

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

style.css

```
body{
/* background-image:url(bg.jpg); */
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;
}
```

main.js

```
$(document).ready(function () {
$('.image-section').hide();
$('.loader').hide();
$('#result').hide();
function readURL(input) {
if (input.files && input.files[0]) {
var reader = new FileReader();
reader.onload = function (e) {
$('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
$('#imagePreview').hide();
$('#imagePreview').fadeIn(650);
}
reader.readAsDataURL(input.files[0]);
}
}
$("#imageUpload").change(function () {
$('.image-section').show();
$('#btn-predict').show();
$('#result').text('');
$('#result').hide();
readURL(this);
});
$('#btn-predict').click(function () {
var form_data = new FormData($('#upload-file')[0]);
$(this).hide();
$('.loader').show();
$.ajax({
type: 'POST',
url: '/predict',
data: form_data,
contentType: false,
cache: false,
processData: false,
async: true,
success: function (data) {
$('.loader').hide();
```

```
$('#result').fadeIn(600);
$('#result').html(data);
console.log('Success!');
},
});
});
});
```

app.py

```
from flask import Flask,render_template,request
import os
import numpy as np #used for numerical analysis
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import requests
app = Flask(__name__,template_folder="templates")
model=load_model('nutrition.h5')
print("Loaded model from disk")


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


@app.route('/image1',methods=['GET','POST'])
def image1():
        return render_template("image.html")


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

```
def launch():
    if request.method=='POST':
        f=request.files['file'] #requesting the file
        basepath=os.path.dirname('__file__')
        filepath=os.path.join(basepath,"uploads",f.filename)
        f.save(filepath)#saving the file
        img=image.load_img(filepath,target_size=(64,64))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        pred=np.argmax(model.predict(x), axis=1)
        print("prediction",pred)#printing the prediction
      index=['APPLES','BANANA','ORANGE','PINEAPPLE','WATERMELON']
        result=str(index[pred[0]])
        x=result
        print(x)
        result=nutrition(result)
        print(result)
    return render_template("0.html",showcase=(result),showcase1=(x))


def nutrition(index):
    url = "https://calorieninjas.p.rapidapi.com/v1/nutrition"
    querystring = {"query":index}
    headers = {
        'x-rapidapi-key':
"5d797ab107mshe668f26bd044e64p1ffd34jsnf47bfa9a8ee4",
```

```
        '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__":
    app.run(debug=False)
```

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-24498-1659943754/tree/main/Final%20Deliverables