PROJECT REPORT

on

AI-Powered Nutrition Analyzer for Fitness Enthusiasts

TEAM ID: PNT2022TMID43309

TEAM MEMBERS:

- KIRUTHIK VISHAAL S (TEAM LEADER)
- CIBI SIDDAARTH R (TEAM MEMBER I)
- JAI PRAKASH R (TEAM MEMBER II)
- AJAY RAJ R (TEAM MEMBER III)

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

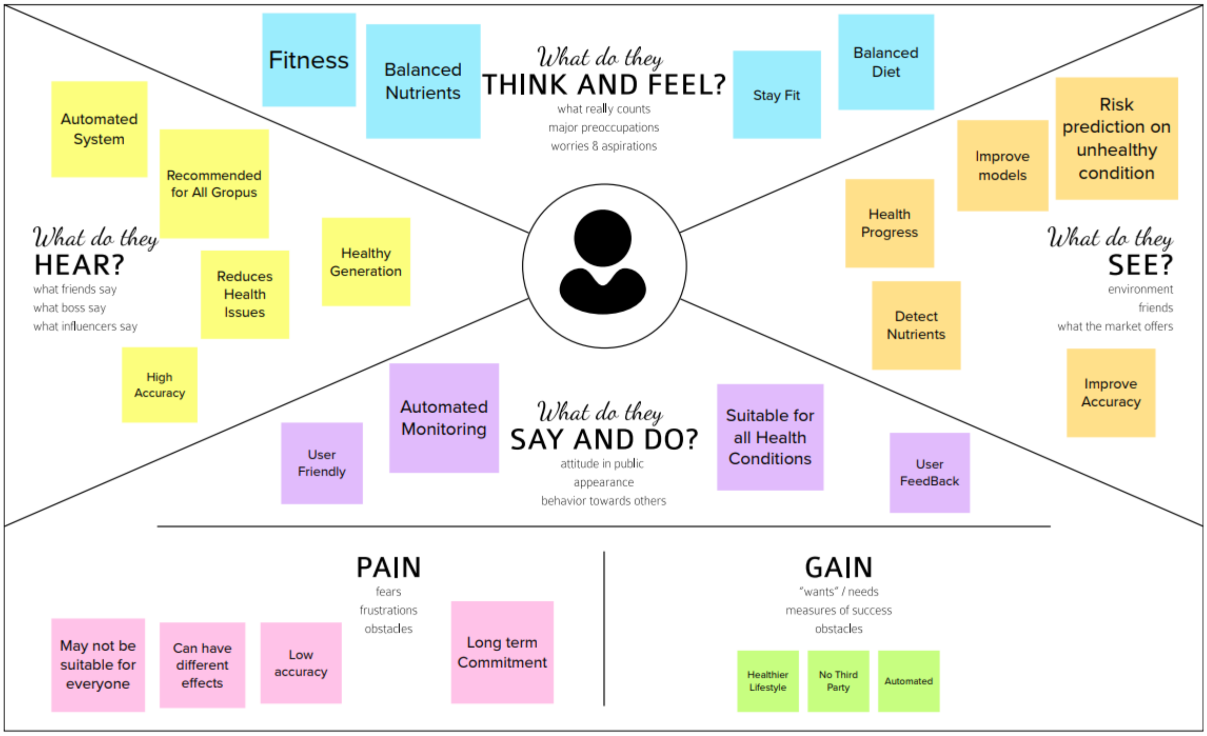
Neutrino delivers nutrition-based data services and analytics to its users and wants to turn into a leading source of the nutrition-related platform. The platform employs NLP and mathematical models from the optimization theory as well as predictive analysis to enable individualized data compilation. The application relies on Artificial Intelligence to produce custom data related to smart calorie counter powered by AI. Their artificial intelligence learns an individual's tastes, preferences, and body type. All of this is packaged in a comprehensive nutrition and activity tracker.

2.2 Problem Statement Definition

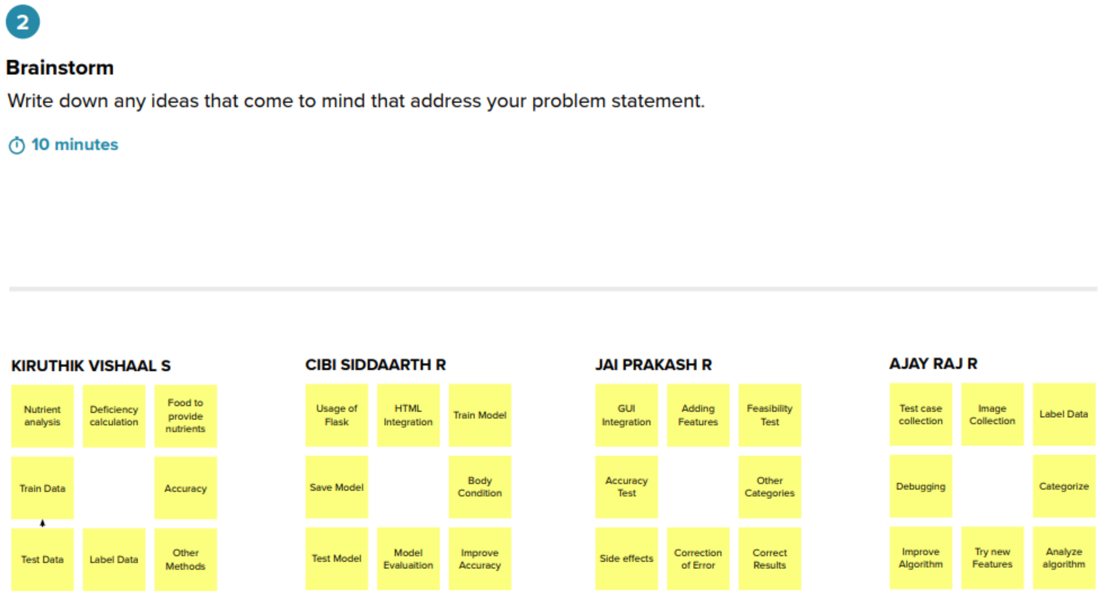
The main aim of the project is to building a model which is used for classifying the fruit depends on the different characteristics like colour, shape, texture etc. Here the user can capture the images of different fruits and then the image will be sent the trained model. The model analyses the image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

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

KIRUTHIK VISHAAL S

Nutrient analysis
Deficiency calculation
Food to provide nutrients
Train Data
Accuracy
Test Data
Label Data
Other Methods

CIBI SIDDAARTH R

Usage of Flask
HTML Integration
Train Model
Save Model
Body Condition
Test Model
Model Evaluaition
Improve Accuracy

JAI PRAKASH R

GUI Integration
Adding Features
Feasibility Test
Accuracy Test
Other Categories
Side effects
Correction of Error
Correct Results

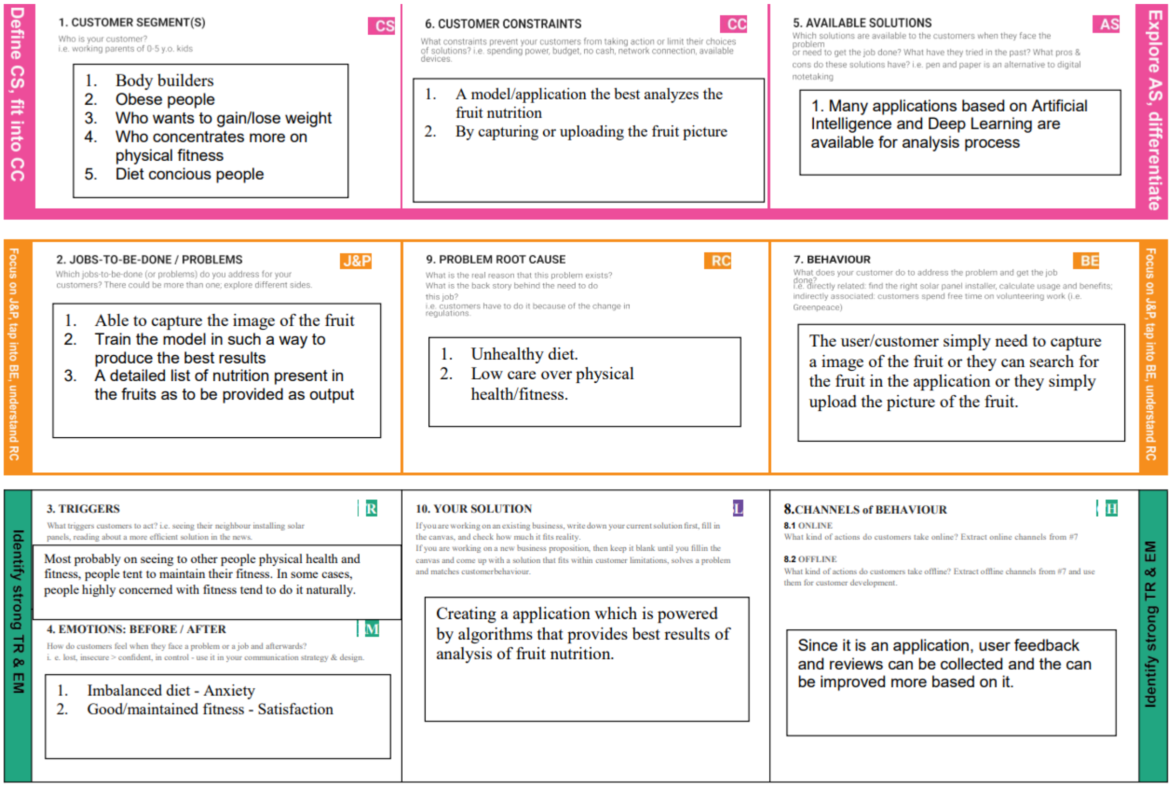
AJAY RAJ R

Test case collection
Image Collection
Label Data
Debugging
Categorize
Improve Algorithm
Try new Features
Analyze algorithm

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	AI-Powered Nutrition Analyzer For Fitness Enthusiasts.
2.	Idea / Solution description	The proposed solution is to classify the fruits using CNN based model and detect the nutrition based on the fruit (like sugar ,calories ,fibre ,protein ,etc)
3.	Novelty / Uniqueness	For detecting the nutrition more images are used to train which is very efficient than other methods for nutrition detection model.
4.	Social Impact / Customer Satisfaction	By this model the customer can efficiently keep track of their health and accordingly get required consultations in a less time consuming manner.
5.	Business Model (Revenue Model)	● Input module ● Image pre processing and segmentation module ● Feature extraction module ● Data set training module ● Nutrition level estimation API module ● Suggestion module
6.	Scalability of the Solution	The accuracy of the result for the training data set is 99% .We can build a large data set which includes different fruit images to get a better result.

3.4 Problem Solution fit


Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Identify strong TR & EM

1. CUSTOMER SEGMENT(S) CS

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

1. Body builders
2. Obese people
3. Who wants to gain/lose weight
4. Who concentrates more on physical fitness
5. Diet concious people

6. CUSTOMER CONSTRAINTS CC

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

1. A model/application the best analyzes the fruit nutrition
2. By capturing or uploading the fruit picture

5. AVAILABLE SOLUTIONS AS

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

1. Many applications based on Artificial Intelligence and Deep Learning are available for analysis process

2. JOBS-TO-BE-DONE / PROBLEMS J&P

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

1. Able to capture the image of the fruit
2. Train the model in such a way to produce the best results
3. A detailed list of nutrition present in the fruits as to be provided as output

9. PROBLEM ROOT CAUSE RC

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

1. Unhealthy diet.
2. Low care over physical health/fitness.

7. BEHAVIOUR BE

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

The user/customer simply need to capture a image of the fruit or they can search for the fruit in the application or they simply upload the picture of the fruit.

3. TRIGGERS TR

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

Most probably on seeing to other people physical health and fitness, people tent to maintain their fitness. In some cases, people highly concerned with fitness tend to do it naturally.

4. EMOTIONS: BEFORE / AFTER EM

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

1. Imbalanced diet - Anxiety
2. Good/maintained fitness - Satisfaction

10. YOUR SOLUTION SL

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

Creating a application which is powered by algorithms that provides best results of analysis of fruit nutrition.

8.CHANNELS of BEHAVIOUR CH

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

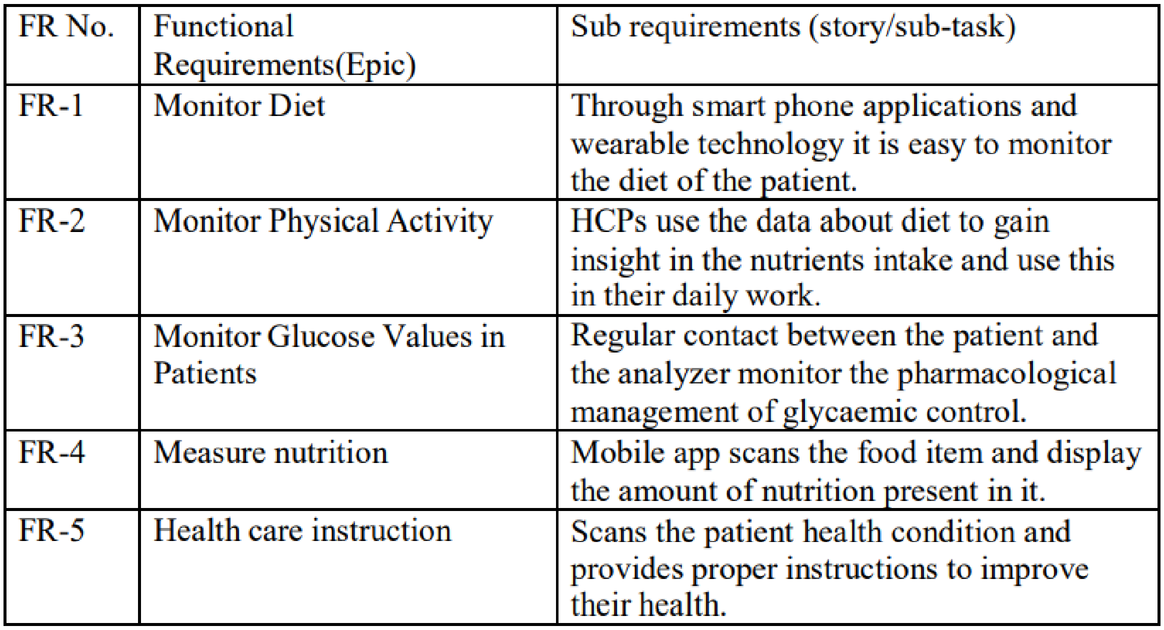
8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Since it is an application, user feedback and reviews can be collected and the can be improved more based on it.

4. REQUIREMENT ANALYSIS

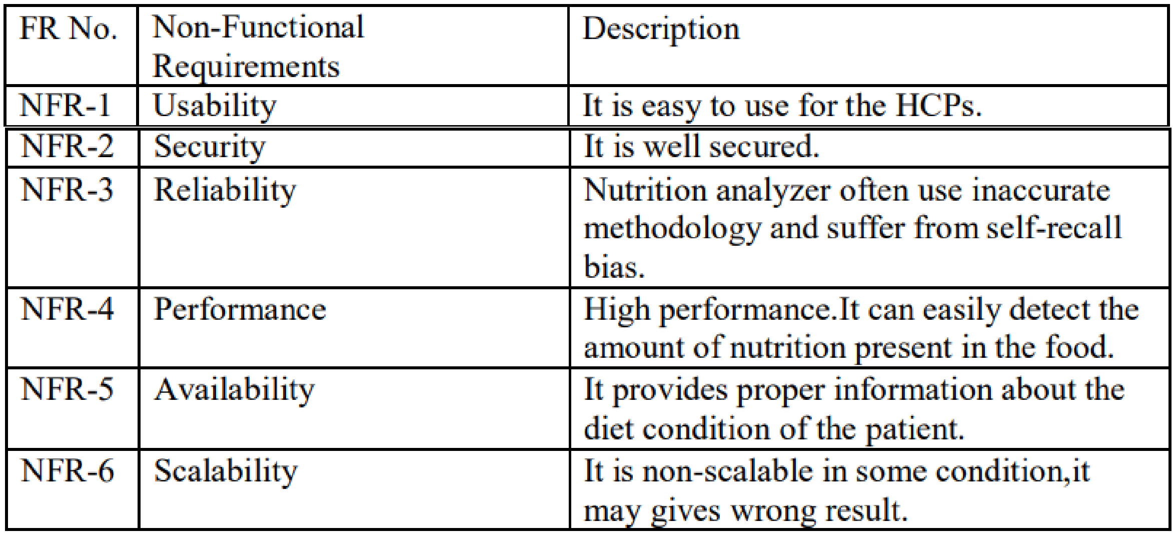
4.1 Functional requirement

Following are the functional requirements of the proposed solution.



FR No.	Functional Requirements(Epic)	Sub requirements (story/sub-task)
FR-1	Monitor Diet	Through smart phone applications and wearable technology it is easy to monitor the diet of the patient.
FR-2	Monitor Physical Activity	HCPs use the data about diet to gain insight in the nutrients intake and use this in their daily work.
FR-3	Monitor Glucose Values in Patients	Regular contact between the patient and the analyzer monitor the pharmacological management of glycaemic control.
FR-4	Measure nutrition	Mobile app scans the food item and display the amount of nutrition present in it.
FR-5	Health care instruction	Scans the patient health condition and provides proper instructions to improve their health.

4.2 Non-Functional requirement

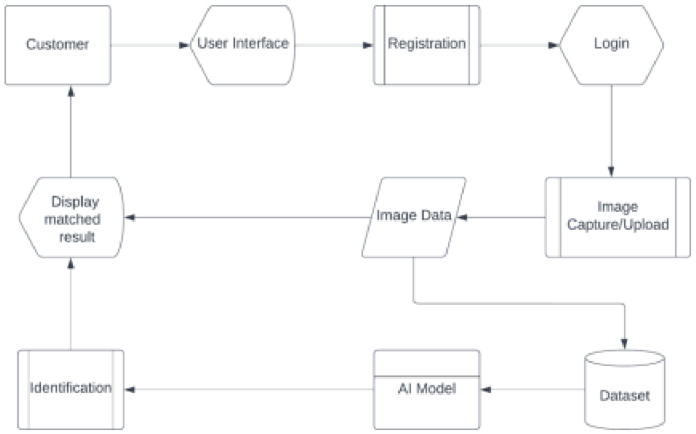


FR No.	Non-Functional Requirements	Description
NFR-1	Usability	It is easy to use for the HCPs.
NFR-2	Security	It is well secured.
NFR-3	Reliability	Nutrition analyzer often use inaccurate methodology and suffer from self-recall bias.
NFR-4	Performance	High performance.It can easily detect the amount of nutrition present in the food.
NFR-5	Availability	It provides proper information about the diet condition of the patient.
NFR-6	Scalability	It is non-scalable in some condition,it may gives wrong result.

5. PROJECT DESIGN

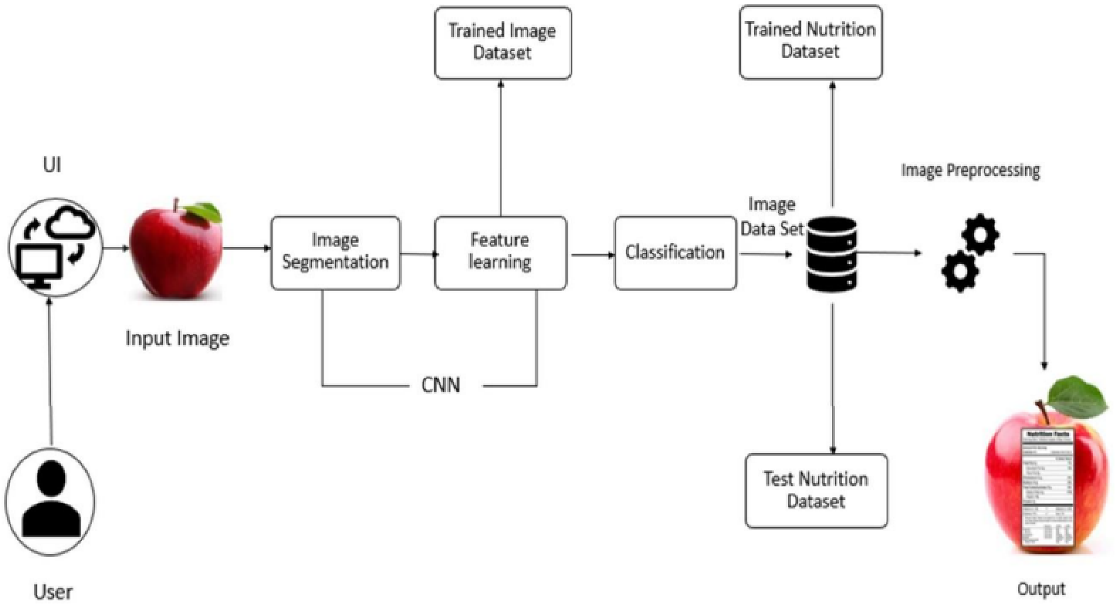
5.1 Data Flow Diagrams

Example: DFD Level 0 (Industry Standard)

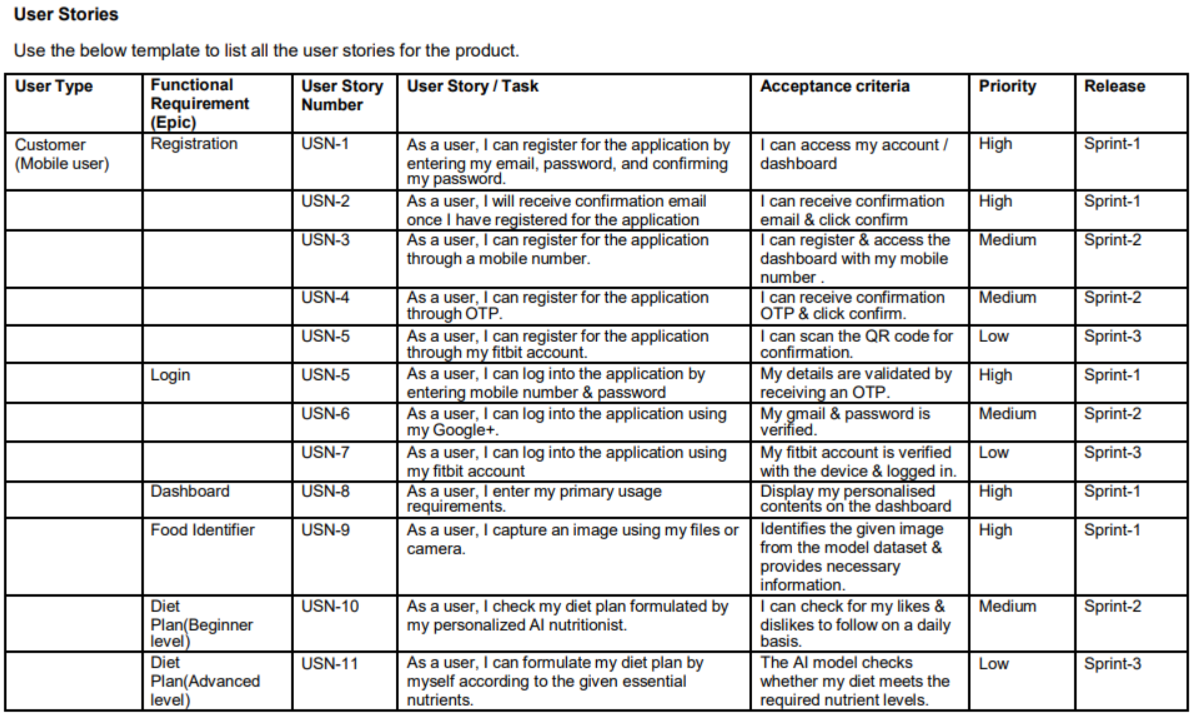


5.2 Solution & Technical Architecture

Solution Architecture:



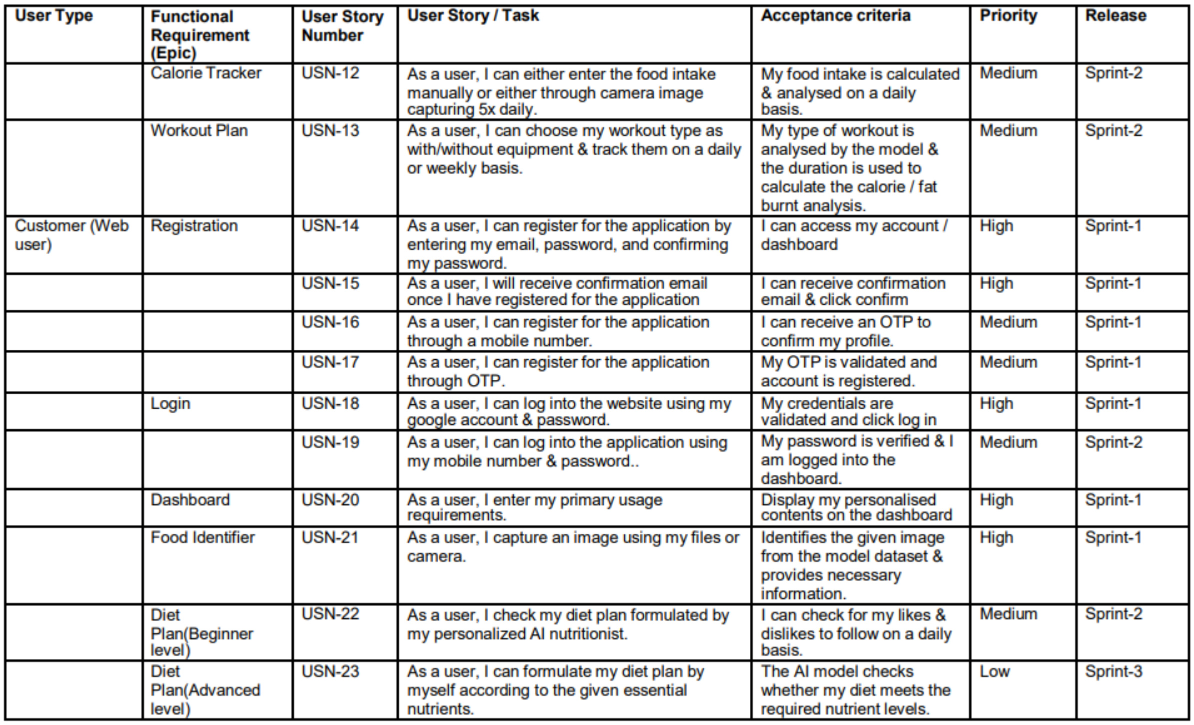
5.3 User Stories



User Stories

Use the below template to list all the user stories for the product.

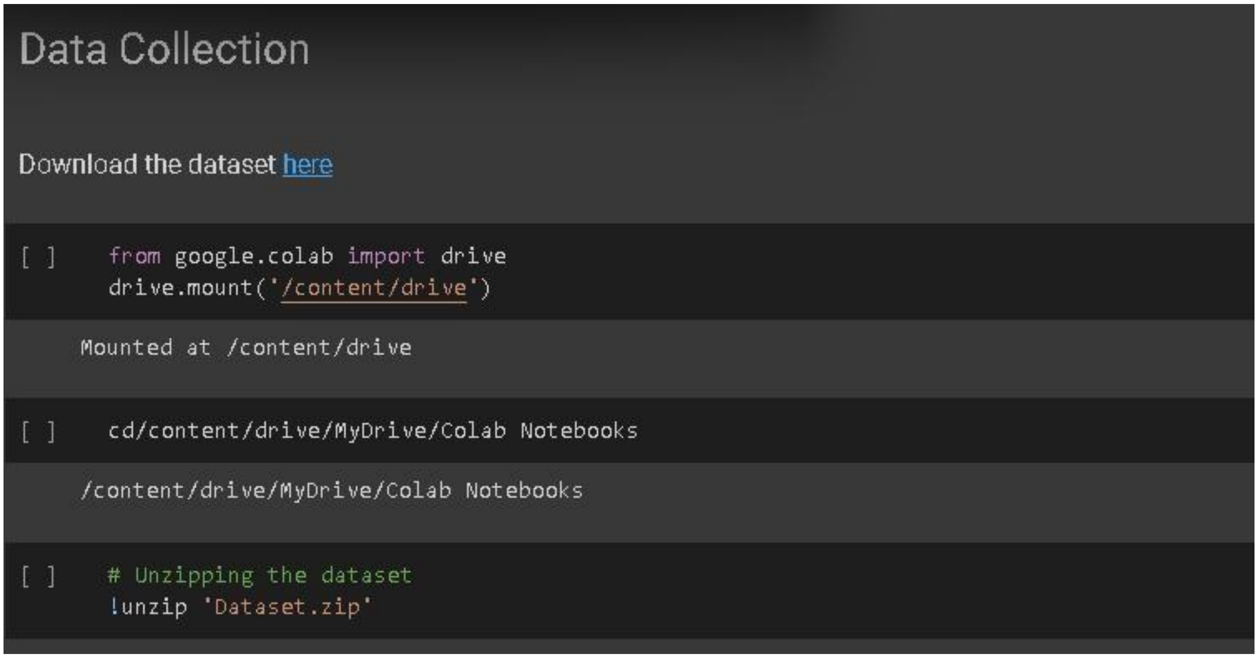
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		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 a mobile number.	I can register & access the dashboard with my mobile number .	Medium	Sprint-2
		USN-4	As a user, I can register for the application through OTP.	I can receive confirmation OTP & click confirm.	Medium	Sprint-2
		USN-5	As a user, I can register for the application through my fitbit account.	I can scan the QR code for confirmation.	Low	Sprint-3
	Login	USN-5	As a user, I can log into the application by entering mobile number & password	My details are validated by receiving an OTP.	High	Sprint-1
		USN-6	As a user, I can log into the application using my Google+.	My gmail & password is verified.	Medium	Sprint-2
		USN-7	As a user, I can log into the application using my fitbit account	My fitbit account is verified with the device & logged in.	Low	Sprint-3
	Dashboard	USN-8	As a user, I enter my primary usage requirements.	Display my personalised contents on the dashboard	High	Sprint-1
	Food Identifier	USN-9	As a user, I capture an image using my files or camera.	Identifies the given image from the model dataset & provides necessary information.	High	Sprint-1
	Diet Plan(Beginner level)	USN-10	As a user, I check my diet plan formulated by my personalized AI nutritionist.	I can check for my likes & dislikes to follow on a daily basis.	Medium	Sprint-2
	Diet Plan(Advanced level)	USN-11	As a user, I can formulate my diet plan by myself according to the given essential nutrients.	The AI model checks whether my diet meets the required nutrient levels.	Low	Sprint-3



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Calorie Tracker	USN-12	As a user, I can either enter the food intake manually or either through camera image capturing 5x daily.	My food intake is calculated & analysed on a daily basis.	Medium	Sprint-2
	Workout Plan	USN-13	As a user, I can choose my workout type as with/without equipment & track them on a daily or weekly basis.	My type of workout is analysed by the model & the duration is used to calculate the calorie / fat burnt analysis.	Medium	Sprint-2
Customer (Web user)	Registration	USN-14	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-15	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-16	As a user, I can register for the application through a mobile number.	I can receive an OTP to confirm my profile.	Medium	Sprint-1
		USN-17	As a user, I can register for the application through OTP.	My OTP is validated and account is registered.	Medium	Sprint-1
	Login	USN-18	As a user, I can log into the website using my google account & password.	My credentials are validated and click log in	High	Sprint-1
		USN-19	As a user, I can log into the application using my mobile number & password..	My password is verified & I am logged into the dashboard.	Medium	Sprint-2
	Dashboard	USN-20	As a user, I enter my primary usage requirements.	Display my personalised contents on the dashboard	High	Sprint-1
	Food Identifier	USN-21	As a user, I capture an image using my files or camera.	Identifies the given image from the model dataset & provides necessary information.	High	Sprint-1
	Diet Plan(Beginner level)	USN-22	As a user, I check my diet plan formulated by my personalized AI nutritionist.	I can check for my likes & dislikes to follow on a daily basis.	Medium	Sprint-2
	Diet Plan(Advanced level)	USN-23	As a user, I can formulate my diet plan by myself according to the given essential nutrients.	The AI model checks whether my diet meets the required nutrient levels.	Low	Sprint-3

1. CODING & SOLUTIONING

1.1Feature 1



Data Collection

Download the dataset here

```
[ ]   from google.colab import drive
      drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[ ]   cd/content/drive/MyDrive/Colab Notebooks
```

```
/content/drive/MyDrive/Colab Notebooks
```

```
[ ]   # Unzipping the dataset
      !unzip 'Dataset.zip'
```

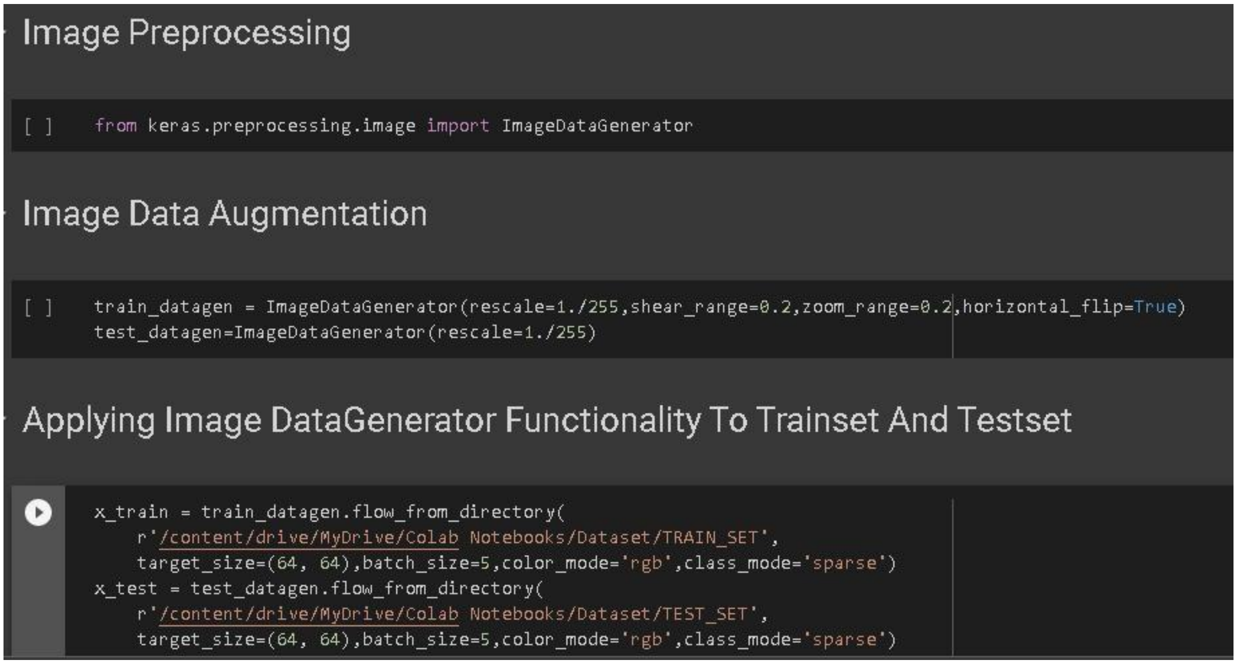


Image Preprocessing

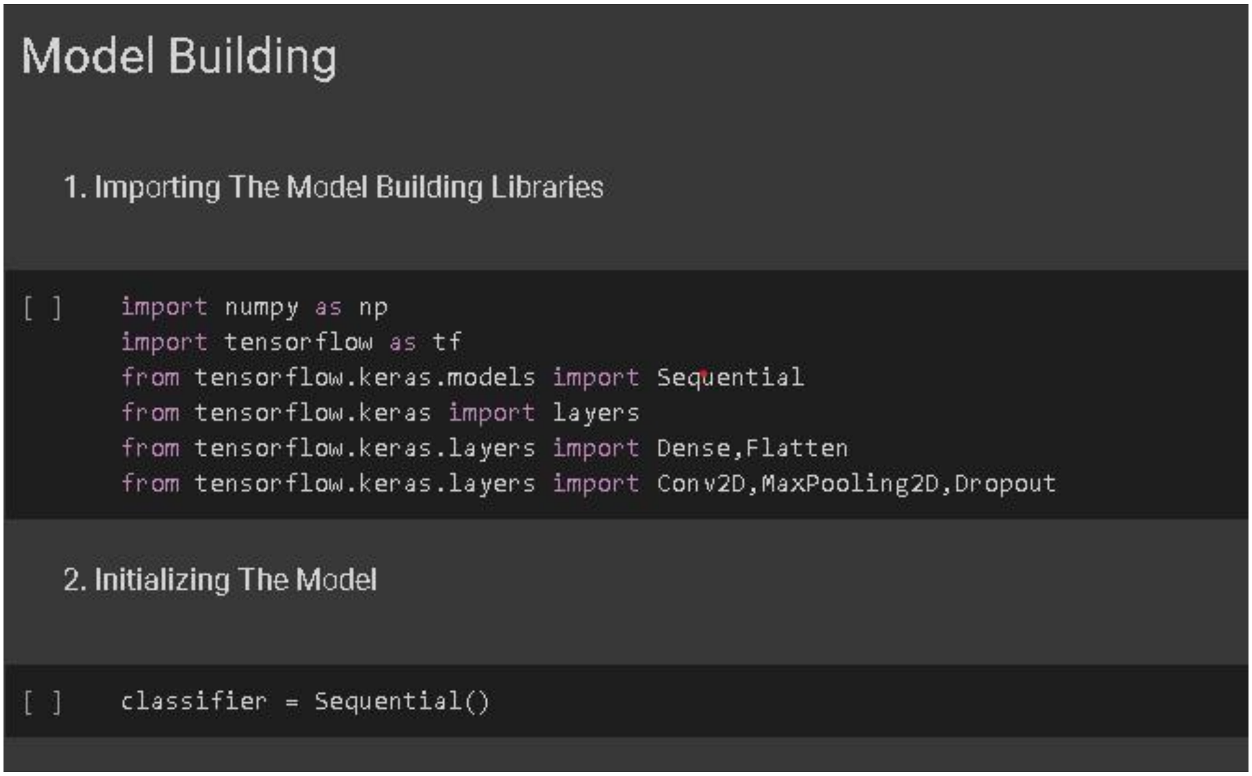
```
[ ]   from keras.preprocessing.image import ImageDataGenerator
```

Image Data Augmentation

```
[ ]   train_datagen = ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
      test_datagen=ImageDataGenerator(rescale=1./255)
```

Applying Image DataGenerator Functionality To Trainset And Testset

```
x_train = train_datagen.flow_from_directory(
    r'/content/drive/MyDrive/Colab Notebooks/Dataset/TRAIN_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
x_test = test_datagen.flow_from_directory(
    r'/content/drive/MyDrive/Colab Notebooks/Dataset/TEST_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```



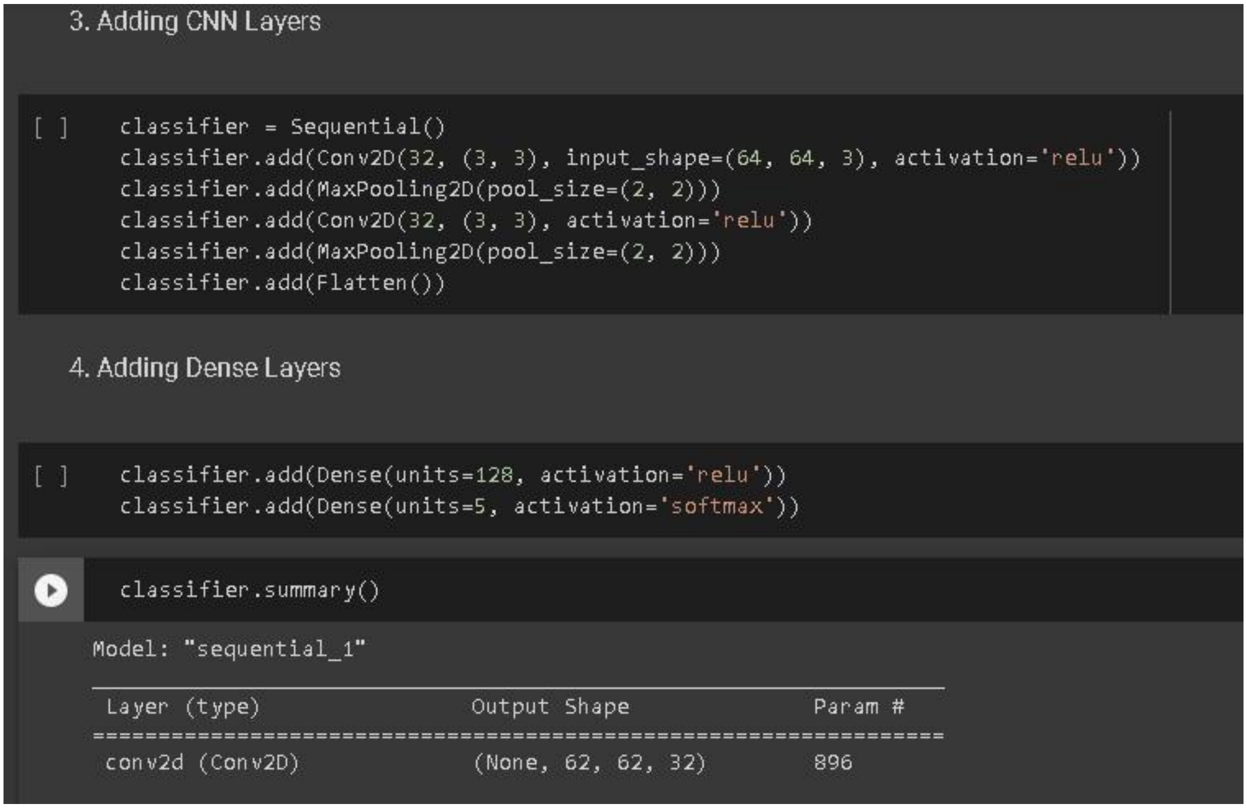
Model Building

1. Importing The Model Building Libraries

```
import numpy as np
import tensorflow as tf
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
```

2. Initializing The Model

```
classifier = Sequential()
```



3. Adding CNN Layers

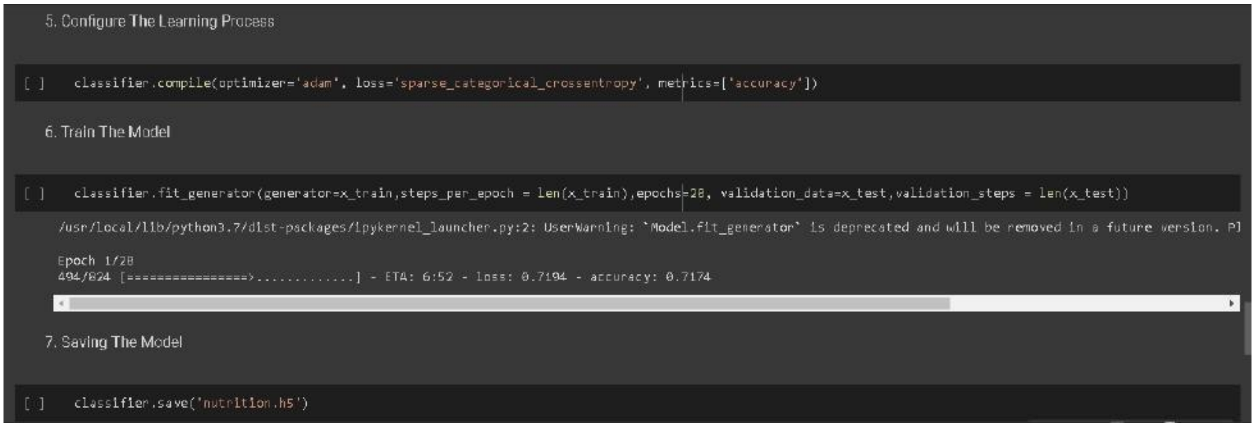
```
classifier = Sequential()
classifier.add(Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
classifier.add(Conv2D(32, (3, 3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2, 2)))
classifier.add(Flatten())
```

4. Adding Dense Layers

```
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=5, activation='softmax'))
```

```
classifier.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896
```



5. Configure The Learning Process

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

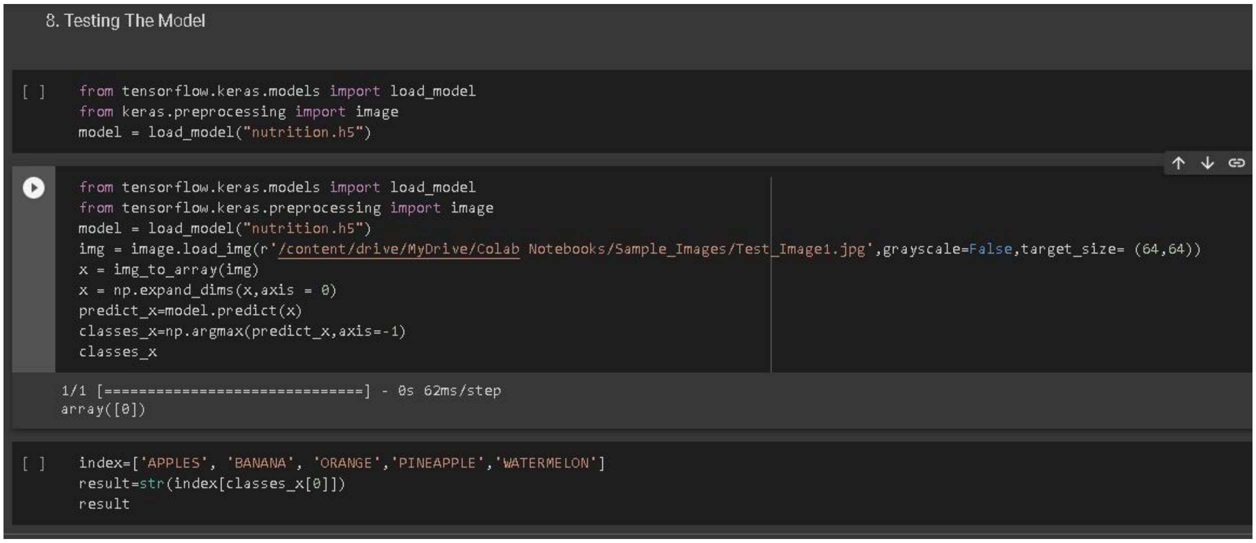
6. Train The Model

```
[ ]   classifier.fit_generator(generator=x_train,steps_per_epoch = len(x_train),epochs=28, validation_data=x_test,validation_steps = len(x_test))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Pl

Epoch 1/28
494/824 [================>.............] - ETA: 6:52 - loss: 0.7194 - accuracy: 0.7174
```

7. Saving The Model

```
[ ]   classifier.save('nutrition.h5')
```



8. Testing The Model

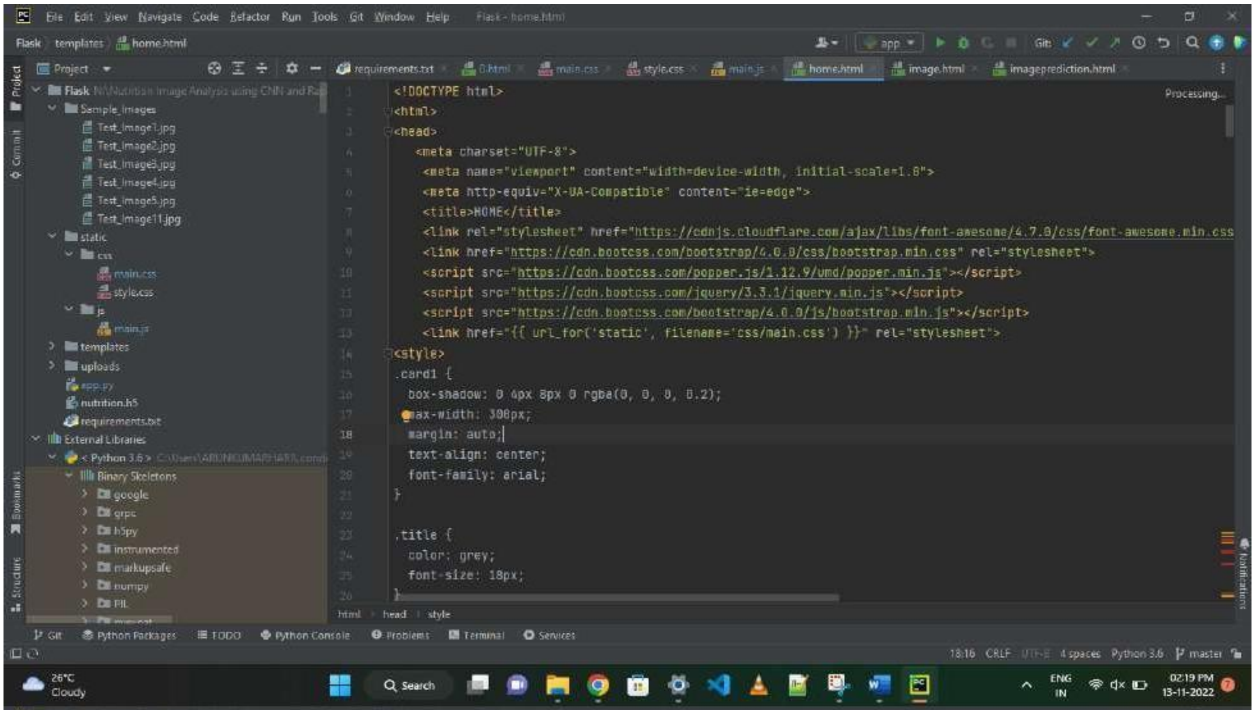
```
[ ]   from tensorflow.keras.models import load_model
      from keras.preprocessing import image
      model = load_model("nutrition.h5")
```

```
      from tensorflow.keras.models import load_model
      from tensorflow.keras.preprocessing import image
      model = load_model("nutrition.h5")
      img = image.load_img(r'/content/drive/MyDrive/Colab Notebooks/Sample_Images/Test_Image1.jpg',grayscale=False,target_size= (64,64))
      x = img_to_array(img)
      x = np.expand_dims(x,axis = 0)
      predict_x=model.predict(x)
      classes_x=np.argmax(predict_x,axis=-1)
      classes_x
```

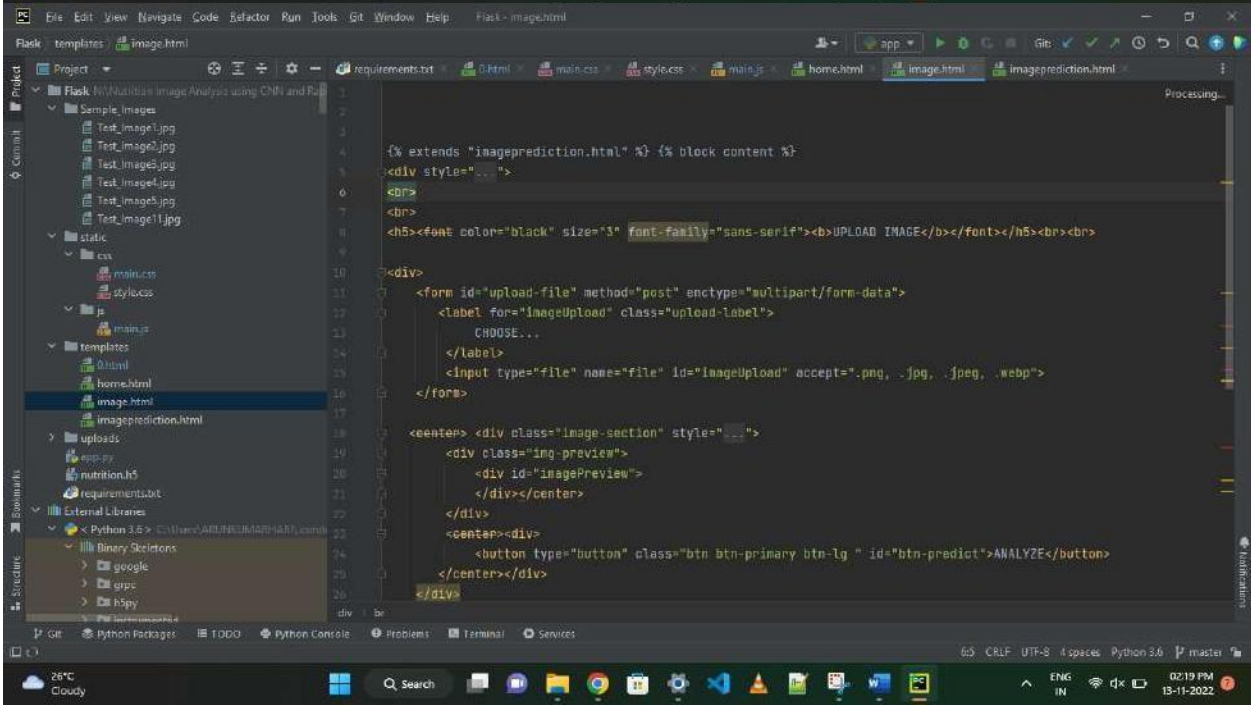
```
1/1 [==============================] - 0s 62ms/step
array([0])
```

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

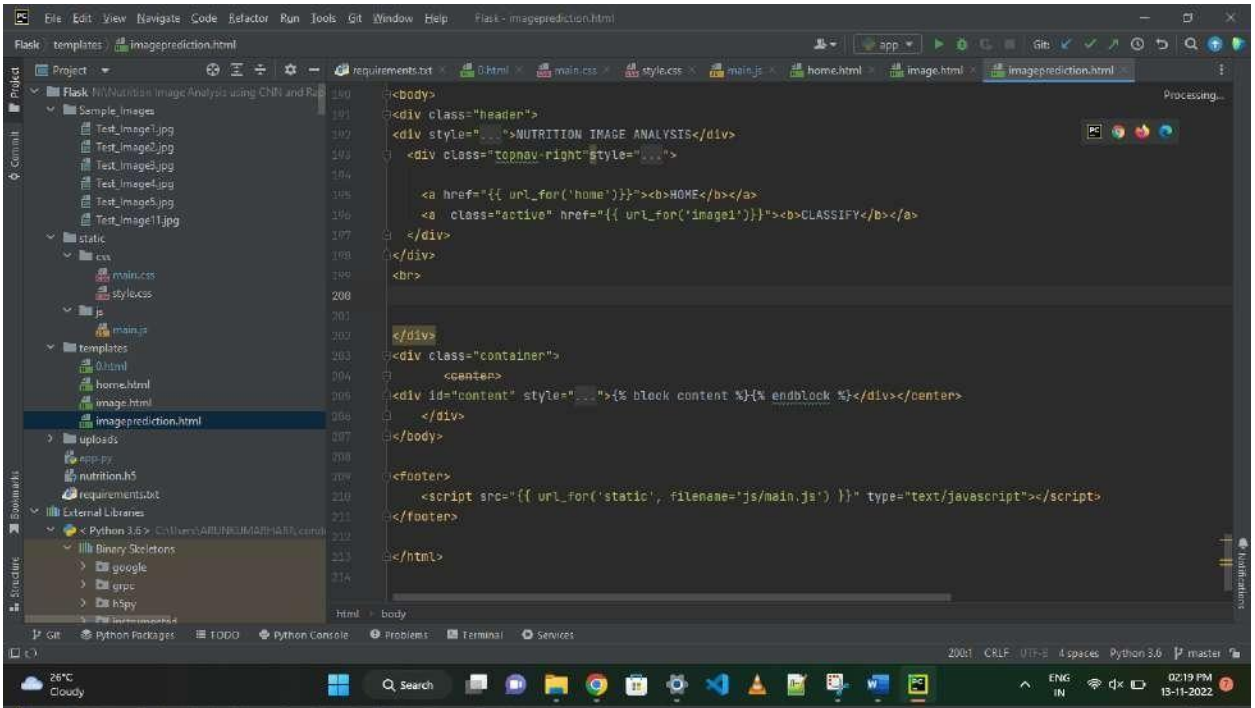
1.2 Feature 2



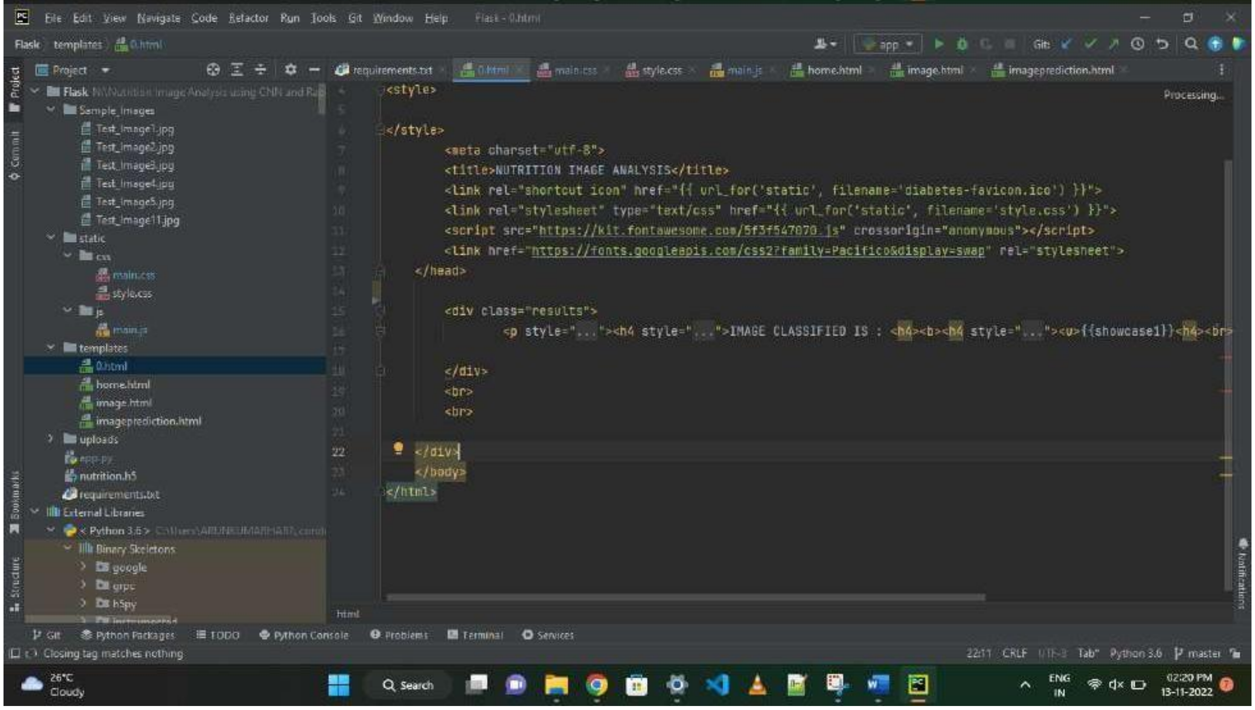
```
<!DOCTYPE html>
<html>
<head>
   <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>HOME</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css
    <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>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
<style>
.card1 {
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
  max-width: 300px;
  margin: auto;
  text-align: center;
  font-family: arial;
}

.title {
  color: grey;
  font-size: 18px;
}
```



```


{% extends "imageprediction.html" %} {% block content %}
<div style="...">
<br>
<br>
<h5><font color="black" size="3" 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, .webp">
    </form>

   <center> <div class="image-section" style="...">
        <div class="img-preview">
            <div id="imagePreview">
            </div></center>
        </div>
        <center><div>
            <button type="button" class="btn btn-primary btn-lg " id="btn-predict">ANALYZE</button>
       </center></div>
    </div>
```

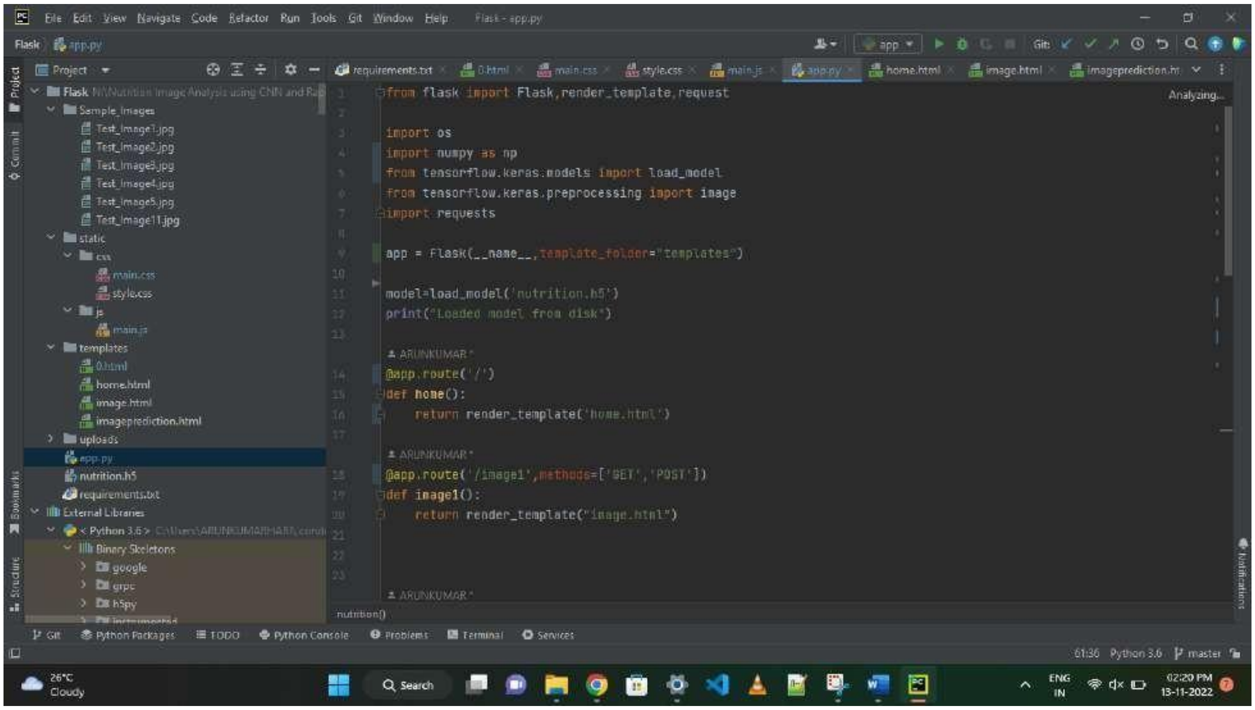


```
<body>
<div class="header">
<div style="...">NUTRITION IMAGE ANALYSIS</div>
  <div class="topnav-right"style="...">

    <a href="{{ url_for('home')}}"><b>HOME</b></a>
    <a  class="active" href="{{ url_for('image1')}}"><b>CLASSIFY</b></a>
  </div>
</div>
<br>


</div>
<div class="container">
        <center>
<div id="content" style="...">{% block content %}{% endblock %}</div></center>
    </div>
</body>

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

</html>
```



```
<style>

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

        <div class="results">
                <p style="..."><h4 style="...">IMAGE CLASSIFIED IS : <h4><b><h4 style="..."><u>{{showcase1}}<h4><br>

        </div>
        <br>
        <br>

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

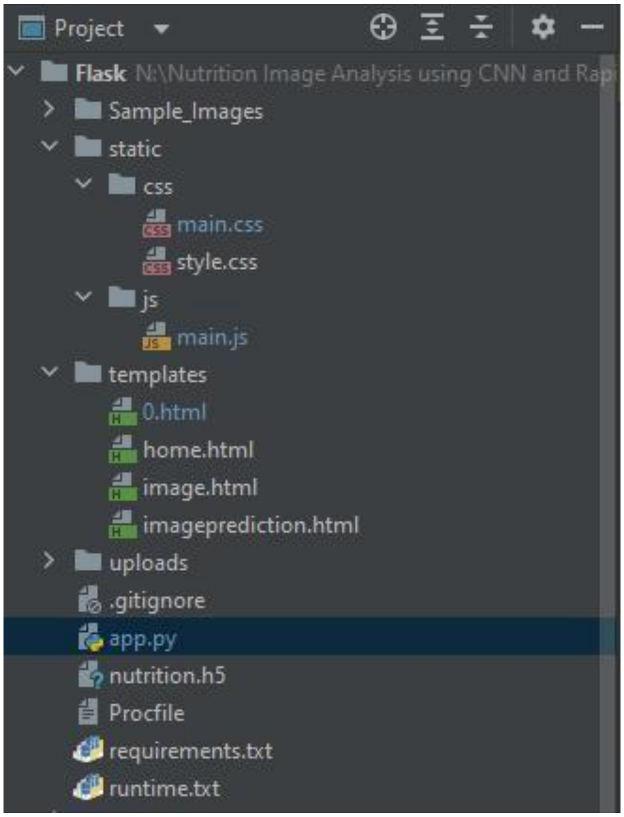
Closing tag matches nothing

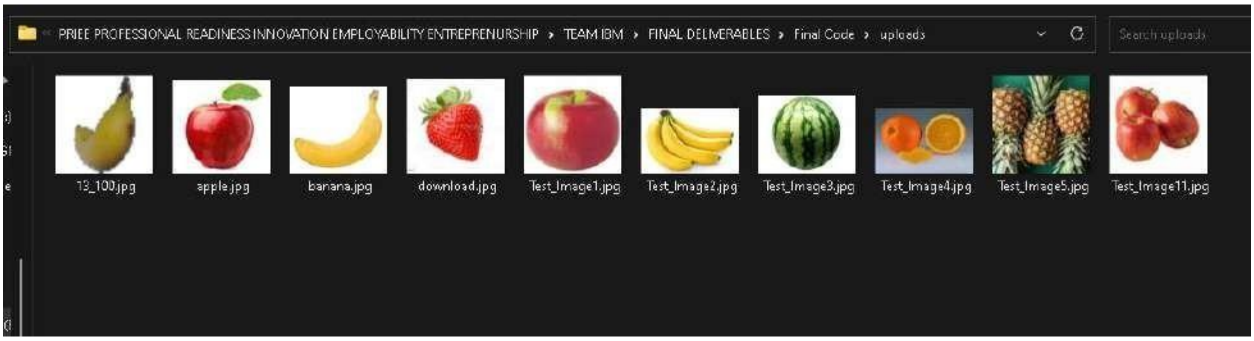


```
from flask import Flask,render_template,request

import os
import numpy as np
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")
```

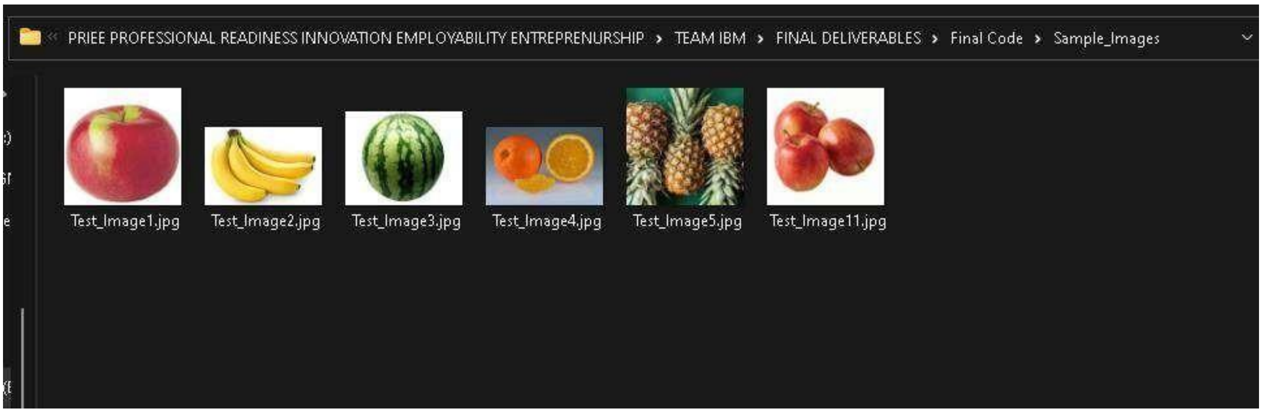
2. TESTING

2.1 Test Cases



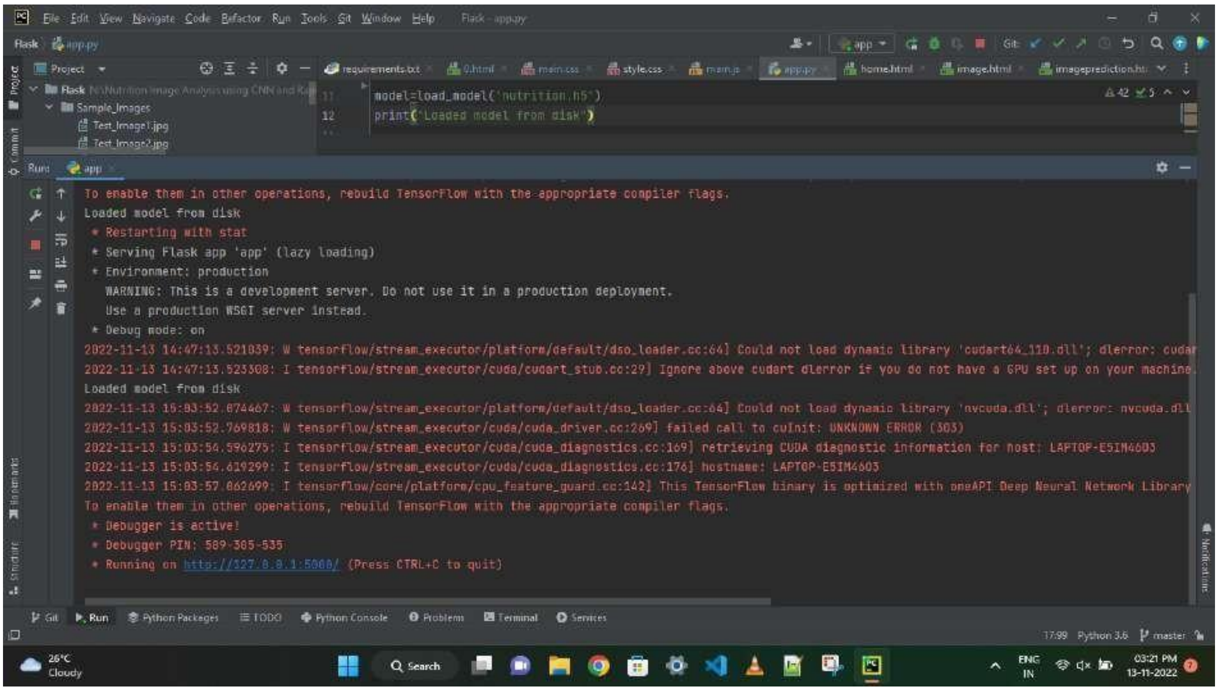


2.2User Acceptance Testing


PRIEE PROFESSIONAL READINESS INNOVATION EMPLOYABILITY ENTREPRENURSHIP › TEAM IBM › FINAL DELIVERABLES › Final Code › Sample_Images
Test_Image1.jpg
Test_Image2.jpg
Test_Image3.jpg
Test_Image4.jpg
Test_Image5.jpg
Test_Image11.jpg

3. RESULTS

3.1 Performance Metrics



```
model=load_model('nutrition.h5')
print('Loaded model from disk')
```

```
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
Loaded model from disk
 * Restarting with stat
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
2022-11-13 14:47:13.521039: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dlerror: cudar
2022-11-13 14:47:13.523308: 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.
Loaded model from disk
2022-11-13 15:03:52.074467: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll
2022-11-13 15:03:52.769818: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2022-11-13 15:03:54.596275: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: LAPTOP-E5IM4603
2022-11-13 15:03:54.619299: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: LAPTOP-E5IM4603
2022-11-13 15:03:57.862699: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
 * Debugger is active!
 * Debugger PIN: 589-305-535
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

10. CONCLUSION

By the end of this project we will

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

11. FUTURE SCOPE

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

Source Code

```
from flask import Flask,render_template,request

import os

import numpy as np

from keras.models import load_model

from keras_preprocessing import image

import requests

from werkzeug.utils import secure_filename

app = Flask(__name__,template_folder="template")

model = load_model(r'C:\Users\KIRUTHIK VISHAAL S\Desktop\own\nutrition.h5')

print('loaded model from disk')

app.config['IMAGE_UPLOADS'] = r"C:\Users\KIRUTHIK VISHAAL S\Desktop\own\Sprint-
4\Sample_Images\\"

@app.route('/')

def home():

    return render_template('index.html')

@app.route('/image1' ,methods=['Get','Post'])

def image1():

    return render_template("Image.html")

@app.route('/predict',methods=['Get','Post'])

def launch():

    f = request.files['file']

    filename = secure_filename(f.filename)

    basedir =  os.path.abspath(os.path.dirname(__file__))

    f.save(os.path.join(basedir,app.config["IMAGE_UPLOADS"],filename))

    p = r"C:\Users\KIRUTHIK VISHAAL S\Desktop\own\Sprint-4\Sample_Images\\"+filename

    img = image.load_img(p,grayscale=False,target_size=(64,64))

    x= image.img_to_array(img)

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

    pred =model.predict(x)

    pred = pred.astype('int32')
    n = np.array(pred[0])
    s = np.where(n==1)
    index= ['APPLE','BANANA','ORANGE','PINEAPPLE','WATERMELON']
    n=int(s[0])
    result=(index[n])
    apiResult=nutrition(result)
    final_result = {
            "result" : result,
            "apiResult" : apiResult
        }
    return final_result

def nutrition(index):
    url = "https://calorieninjas.p.rapidapi.com/v1/nutrition"
    querystring = {"query":index}
    headers = {
            "X-RapidAPI-Key": "7c2fb6a502msh4e99d771797d074p173659jsnf288c18cf37c",
            "X-RapidAPI-Host": "calorieninjas.p.rapidapi.com"
    }
    response = requests.request("GET", url, headers=headers, params=querystring)
    print(response.text)
    return response.text

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

GitHub & Project Demo Link

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-39372-1660408877

Project Demo Video Link: https://drive.google.com/file/d/1qQnVJdc4phzfr9qw9Zvmg3-6oG2eMl43/view?usp=sharing