UNIVERSITY COLLEGE OF ENGINEERING

BIT Campus, Anna University

TIRUCHIRAPALLI – 620 024

AI-POWERED NUTRITION ANALYZER FOR FITNESS ENTHUSIASTS

IBM NALAIYA THIRAN

PROJECT REPORT

Submitted By

KARTHIKEYAN M	**(810019104303)**
SHRIRAM MA	**(810019104305)**
YUGHENDRAN T	**(810019104306)**
ALEXANDER AS	**(810019104301)**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

Nutrition 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. 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.

A fruit dataset describes variety of fruit images apple, banana, orange, pineapple, watermelon. The fruit nutrition is done by analysis. The analysis can be done using convolution neural network. It has four layers as convolution layer, pooling layer, flattening layer and fully connected layer.

Project Overview

The overview of the project is to give the output by doing the image processing and make the list of nutrients to show off.The user interacts with the UI (User Interface) and give the image as input.Then the input image is then pass to our flask application, and finally with the help of the model which we build we will classify the result and showcase it on the UI.By using the Pycharm the Input is been predicted and output is showcased as nutrition content.

The process involved in this is Data Collection,Image Preprocessing,Model Building,Application Building.Dataset folder contains the training and testing images for training our model.

We are building a Flask Application that under project name Flask , HTML pages are stored in the templates folder and css codings are stored in static folder and python file app.py, Sample_images, uploads, nutrition.h5 are stored in under Flask folder, templates folder contains home.html, image.html, imageprediction.html, aboutus.html pages.

Static folder had the css files which are necessary for styling the html page and for executing the actions.Uploads folder will have the uploaded images(which are already tested).Sample_images will have the images which are used to test or upload.

Purpose

The purpose of this project is to make it beneficial for the fitness enthusiasts.Sometimes,the people tempt to eat more and make their calories increase in their diet and make the diet plan collapse.So at this point this NutriFact website make them to create awareness and make the sense of giving the correct nutrition chart.So that they will be eating only up to the level.This website gives the nutrient-content list like Potassium, Carbohydrates, Fat, Satuarted Fat, Mono-unsaturated Fat, Poly-unsaturated Fat, Energy, Protein, Sugar and Sodium.

CHAPTER 2

LITERATURE SURVEY

Fruit Detection Using Convolution Neural Network (Prashant Kumar,Tamal Datta,Sujata Chakravarty , 2020)

Fruit classification is done by an algorithm based on convolution neural network has been applied for fruit detection. In this we use high-quality, fruit-containing image dataset for training a neural network to detect fruits. The image regions are retrieved using a selective search algorithm. Computer vision is one of the most popular technologies in this era of innovation. The experimental results show that deep neural networks provide more accurate results compared with other machine learning algorithms. This model works efficiently with this architecture for fruit recognition. And also we use various hidden layer combinations to compare them.

As a part of artificial intelligence and machine learning algorithm, deep neural network is used for fruit identification from images. Efficiency of DNN is better than other machine learning algorithms. Convolutional neural network is the most commonly used algorithm in DNN which also performs efficiently for visual recognition including photo and video,face recognition, handwritten digit recognition.

The efficiency of CNN can match human level perfection. CNN has a similar architecture like another deep learning algorithm i.e., ANN. In ANN, several neurons are there in each layer and also the fact is that all the neurons are not fully connected; instead they are connected locally as a part of the receptive field. After that the cost function is also generated for training purposes.

Fast algorithms for texture analysis using co-occurrence matrices (F. Argenti, L.Alparone, G. Benelli , 1990)

Texture analysis may be of great importance for the problem of image classification and recognition. Co-occurrence matrices are quite effective for discriminating different textures but have the disadvantage of a high computational cost. In the paper a fast algorithm for calculating parameters of co-occurrence matrices is presented.This method has been applied to the problem of classification and segmentation of artificial and natural scenes: the classification, based on cooccurrence matrix parameters, is implemented pixel-by-pixel by using supervised learning and maximum likelihood estimates.

The problem of texture boundary recognition has also been considered and a classification scheme based on more than one window for each pixel is presented. Experimental results show the improvements of classification rates that can be achieved by using this method when compared to a single window classification.

Fruit Recognition using Color and Texture Features (S.Arivazhagan, R.Newlin Shebiah, S.Selva Nidhyanandhan, L.Ganesan 2010)

Fruit recognition system need a change in the color space of the images, in order to obtain one channel containing the luminance information and two other channels containing chrominance information. The HSV representation is often selected for its invariant properties. The hue is invariant under the orientation of an object with respect to the illumination and camera direction and hence more suited for object retrieval. Discrete Wavelet Transform and the co-occurrence matrix is constructed from the approximation subband by estimating the pair wise statistics of pixel intensity. The use of the co-occurrence matrix is based on the hypotheses that the same grey-level configuration is repeated in a texture.

There exist 5 co-occurrence features i.e., texture features for an image. Statistical features such as Mean, Standard Deviation, Skewness and Kurtosis are derived from H and S components. Hence there will be 8 chrominance or color statistical features for an image. Thus a total of 13 features characterize one fruit image. In the classification phase, for the test fruit image, color and texture features are derived as that of the training phase and compared with corresponding feature values, stored in the feature library.

The classification is done using the Minimum Distance Criterion. The image from the training set which has the minimum distance when compared with the test image says that the test image belongs to the category of that training image.

A New Method for Fruits Recognition System(Woo Chaw Seng, Seyed Hadi Mirisaee Published, 2009)

A new Fruit recognition system has been proposed, which combines three features analysis methods: colour-based, shape based and size-based in order to increase accuracy of recognition. Proposed method classifies and recognizes fruit images based on obtained features values by using nearest neighbours classification. Consequently, system shows the fruit name and a short description to user. Proposed fruit recognition system analyses, classifies and identifies fruits successfully up to 90% accuracy.

A recognition approach for 2D fruit images is proposed, which combines color-based, shape-based, and size-based methods in order to increase the accuracy of the recognition result. The k-Nearest Neighbors algorithm is the methodology that has been used to develop the Fruit Recognition System.The Fruit Recognition System using the KNN algorithm as a classifier to classify fruit based on mean color values, shape roundness value, area and perimeter values of the fruit.

Food Calorie and Nutrition Analysis System based on MaskR-CNN (Meng-Lin Chiang,Chia-An Wu,Jian-Kai Feng,Chiung-Yao Fang,Sei-Wang Chen, 2019)

This study develops a food calorie and nutrition system that can analyze the composition of a food based on a provided image. Further, we introduce a newly collected dataset, The system is based on a Mask Region-based Convolutional Neural Network(RCNN) with a union postprocessing, which modifies the extracted bounding boxes and masks, without the non-maximum suppression (NMS), to provide a better result in both analytics and visualization.

Once the food image is input into the system, the system scales the image to appropriate size. The resized image is then fed into Mask R-CNN to capture the food features and perform food detection and classification. At this step, Mask R-CNN detects and recognizes the food class and the box regression of the food based on the captured features. The system then estimates the weight of the object through the image of the recognized food. After obtaining the weight of the food, the food calorie and nutrition analysis system is estimated according to the Ministry of Health and Welfare and the US Department of Agriculture's Food Nutrition Database.

Color , Shape and Texture based Fruit Recognition System (Ruaa Adeeb Abdulmunem Al-falluji 2016)

An automated system is used for classification of fruits. A dataset containing five different fruits was constructed using an ordinary camera. All the fruits were analyzed on the basis of their color (RGB space), shape and texture and then classified using different classifiers to find the classifier that gives the best accuracy. GLCM is used to calculate texture features. Best accuracy was achieved by support vector machine .All the processing was carried out in Matlab.

Advancement in the field of cameras and sensors, in recent years, has led to an increase in intelligent systems. The main objective of these systems is to understand and perceive an image as done by humans i.e. understanding the symbolic meaning of images by the help of statistics, models, geometry. The main goal of this work is to automatically recognize fruit image by classifying it according to its features using machine learning techniques.

Existing Problem

Unfortunately, some nutritional software packages are of poor quality, and the technical support provided to users is sometimes inadequate. In addition, although many excellent software packages and databases are available, they are open to misuse by users who do not understand or appreciate the limitations of such systems. This review examines some of the sources of error associated with the use of nutritional analysis software. In many build sources the information of nutrient content is very less. So it can be improvised .

References

1)Sak. J, Suchodolska. M, "Artificial Intelligence in Nutrients Science Research: A Review" Nutrients. 2021;13[2]:322.

2) Nilsson. NJ, "The quest for artificial intelligence" Cambridge University Press. 2009.

3) Matusheski. N, Caffrey. A, "Christensen L, Mezgec S, Surendran S, Hjorth MF, et al. Diets, nutrients, genes and the microbiome: Recent advances in personalised nutrition" British Journal of Nutrition. 2021:1-24.

4) Adams. SH, Anthony. JC, Carvajal. R, Chae. L, Kho.o CSH,Latulippe. ME, "Perspective: guiding principles for the implementation of personalized nutrition approaches that benefit health and function" Advances in Nutrition. 2020;11[1]:25-34.

5) Rozga M, Latulippe ME, Steiber A. "Advancements in Personalized Nutrition Technologies: Guiding Principles for Registered Dietitian Nutritionists" Journal of the Academy of Nutrition and Dietetics. 2020;120

6):1074-85.[6] Shim. JS, Oh. K, Kim. HC, "Dietary assessment methods in epidemiologic studies. Epidemiol Health. 2014;36:e2014009

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.).

CHAPTER 3

IDEATION & PROPOSED SOLUTION

Empathy Map Canvas

In this following empathy map we gave a clear idea about our AI-Nutrition Analyzer Website. The outlook of this is what the people think and feel, see, say

For example:

Pain: Slow process for some people, negative feedbacks and accuracy may vary.

Gain: Extra things like consultants, quick and easy and many food journals.

Think and Feel: Using a dedicated smart phone app, easy to work with it, it can reduce man's power.

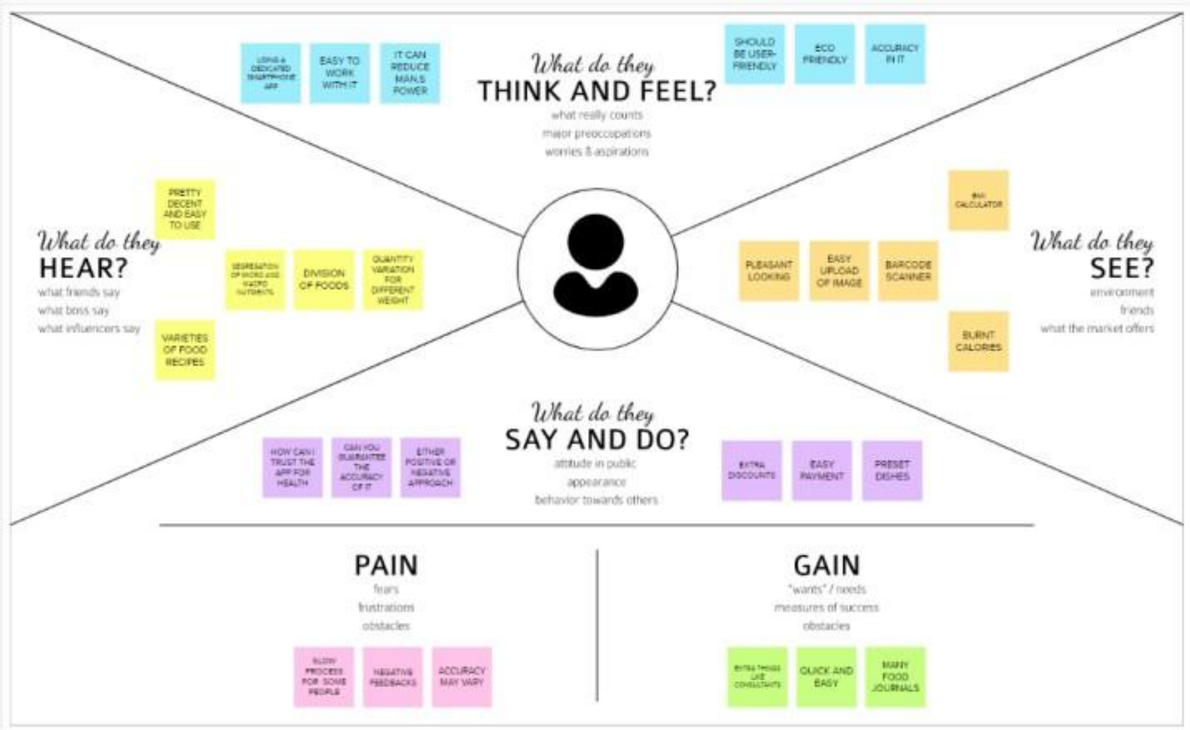
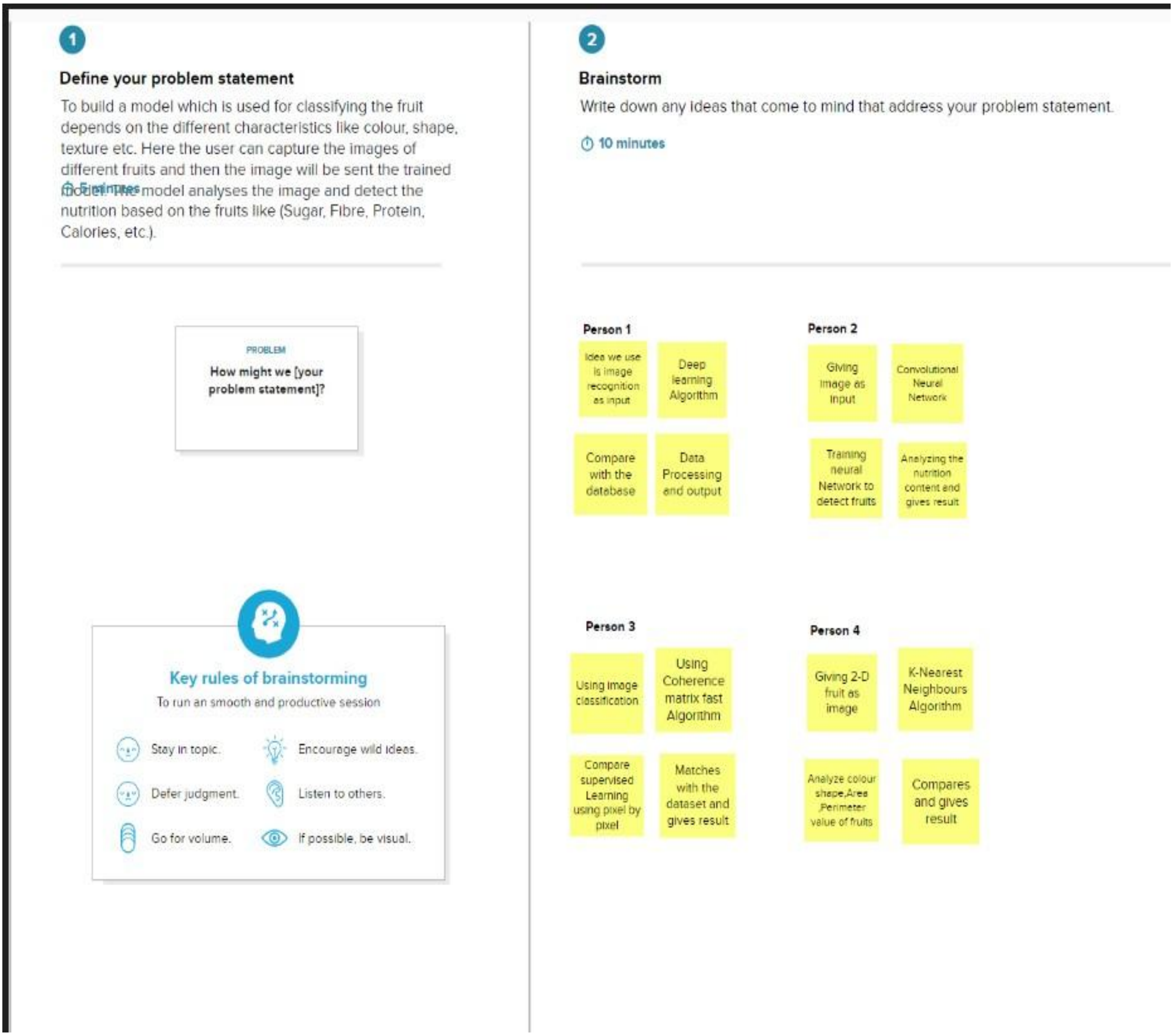


figure 3.1.1

Ideation & Brainstorming

We have discussed different ideas during ideation phase and organised under person 1 and so on .We have discussed about various algorithms for implementation.This ideas give a clear view about the process.



1

Define your problem statement

To build 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.).

PROBLEM

How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.

Encourage wild ideas.

Defer judgment.

Listen to others.

Go for volume.

If possible, be visual.

2

Brainstorm

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

10 minutes

Person 1

Idea we use is image recognition as input

Deep learning Algorithm

Compare with the database

Data Processing and output

Person 2

Giving image as input

Convolutional Neural Network

Training neural Network to detect fruits

Analyzing the nutrition content and gives result

Person 3

Using image classification

Using Coherence matrix fast Algorithm

Compare supervised Learning using pixel by pixel

Matches with the dataset and gives result

Person 4

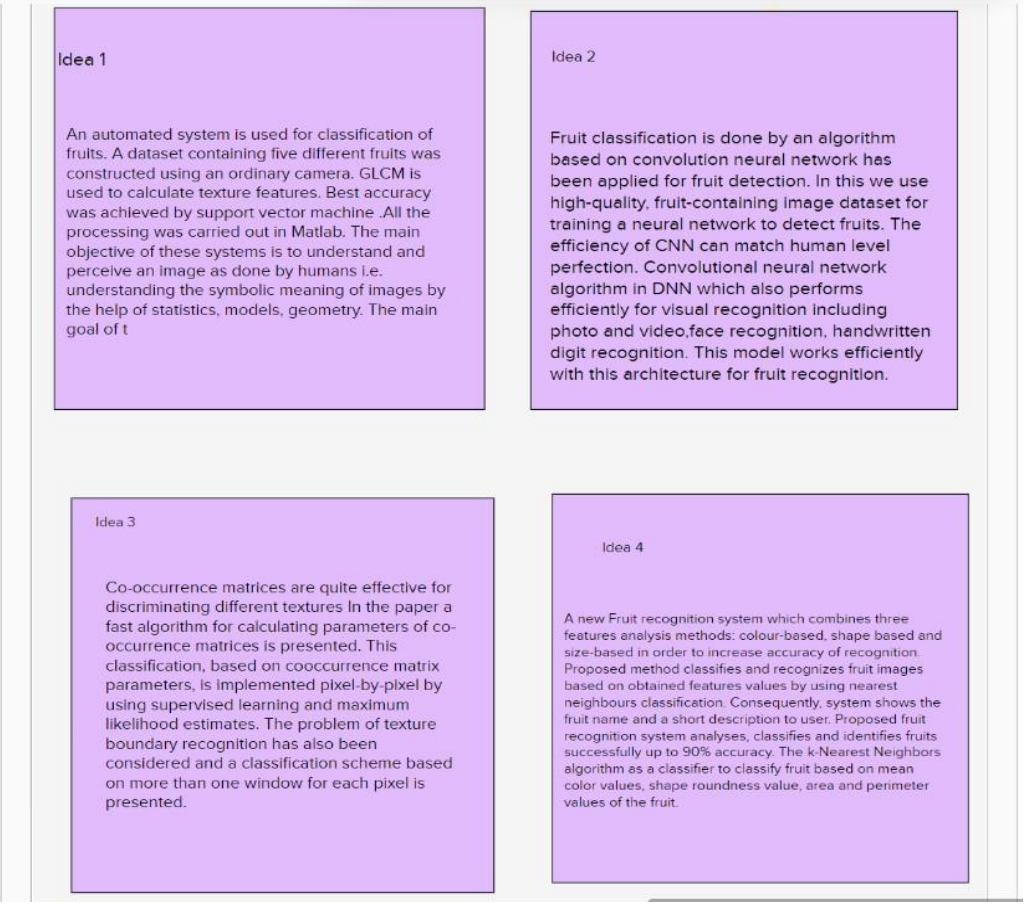
Giving 2-D fruit as image

K-Nearest Neighbours Algorithm

Analyze colour shape,Area ,Perimeter value of fruits

Compares and gives result

figure 3.2.1



Idea 1

An automated system is used for classification of fruits. A dataset containing five different fruits was constructed using an ordinary camera. GLCM is used to calculate texture features. Best accuracy was achieved by support vector machine .All the processing was carried out in Matlab. The main objective of these systems is to understand and perceive an image as done by humans i.e. understanding the symbolic meaning of images by the help of statistics, models, geometry. The main goal of t

Idea 2

Fruit classification is done by an algorithm based on convolution neural network has been applied for fruit detection. In this we use high-quality, fruit-containing image dataset for training a neural network to detect fruits. The efficiency of CNN can match human level perfection. Convolutional neural network algorithm in DNN which also performs efficiently for visual recognition including photo and video,face recognition, handwritten digit recognition. This model works efficiently with this architecture for fruit recognition.

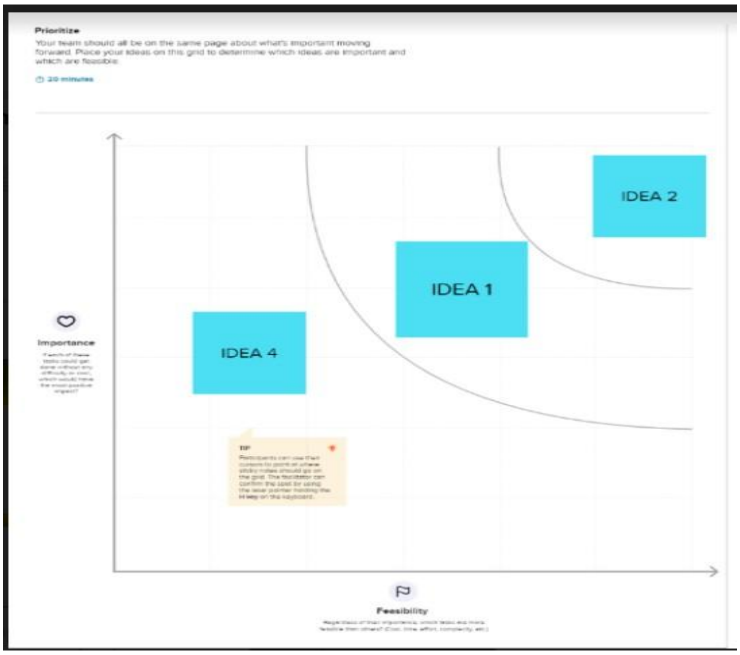
Idea 3

Co-occurrence matrices are quite effective for discriminating different textures In the paper a fast algorithm for calculating parameters of co-occurrence matrices is presented. This classification, based on cooccurrence matrix parameters, is implemented pixel-by-pixel by using supervised learning and maximum likelihood estimates. The problem of texture boundary recognition has also been considered and a classification scheme based on more than one window for each pixel is presented.

Idea 4

A new Fruit recognition system which combines three features analysis methods: colour-based, shape based and size-based in order to increase accuracy of recognition. Proposed method classifies and recognizes fruit images based on obtained features values by using nearest neighbours classification. Consequently, system shows the fruit name and a short description to user. Proposed fruit recognition system analyses, classifies and identifies fruits successfully up to 90% accuracy. The k-Nearest Neighbors algorithm as a classifier to classify fruit based on mean color values, shape roundness value, area and perimeter values of the fruit.

figure 3.2.2



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

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

figure 3.2.3

Proposed solution

We have used convolution neural network to analyse the image.It has four layers.They are convolution layer, pooling layer, flattening layer, fully connected layer.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	1. Identifying the food through the Image processing . 2. Listing out the Nutrition present in it .
2.	Idea / Solution description	1. Idea used here is by using the Convolution Neural Network . 2. It is used to pick the raw pixels present in the image . 3. Fruit Recognition using Colour and Texture Features .
3.	Novelty / Uniqueness	There are three steps present in it. 1. Pooling. 2. Flattening 3. Getting output.
4.	Social Impact / Customer Satisfaction	1. It should be low of cost. 2. Socially It should help others to help with their Fitness.
5.	Business Model (Revenue Model)	1. Creating using the Web Development .
6.	Scalability of the Solution	1. It should work good. 2. There should be no trouble in using apps. 3. It is easy to work with the app. 4. It should be accurate

Table 3.3.1

Convolution layer:

It has a filter passes over the image,scanning a few pixels at a time and creating a feature map that predicts the class to which each feature belongs.

Pooling Layer:

It reduces the amount of information in each feature obtained in the convolution layer while maintaining the most important information.

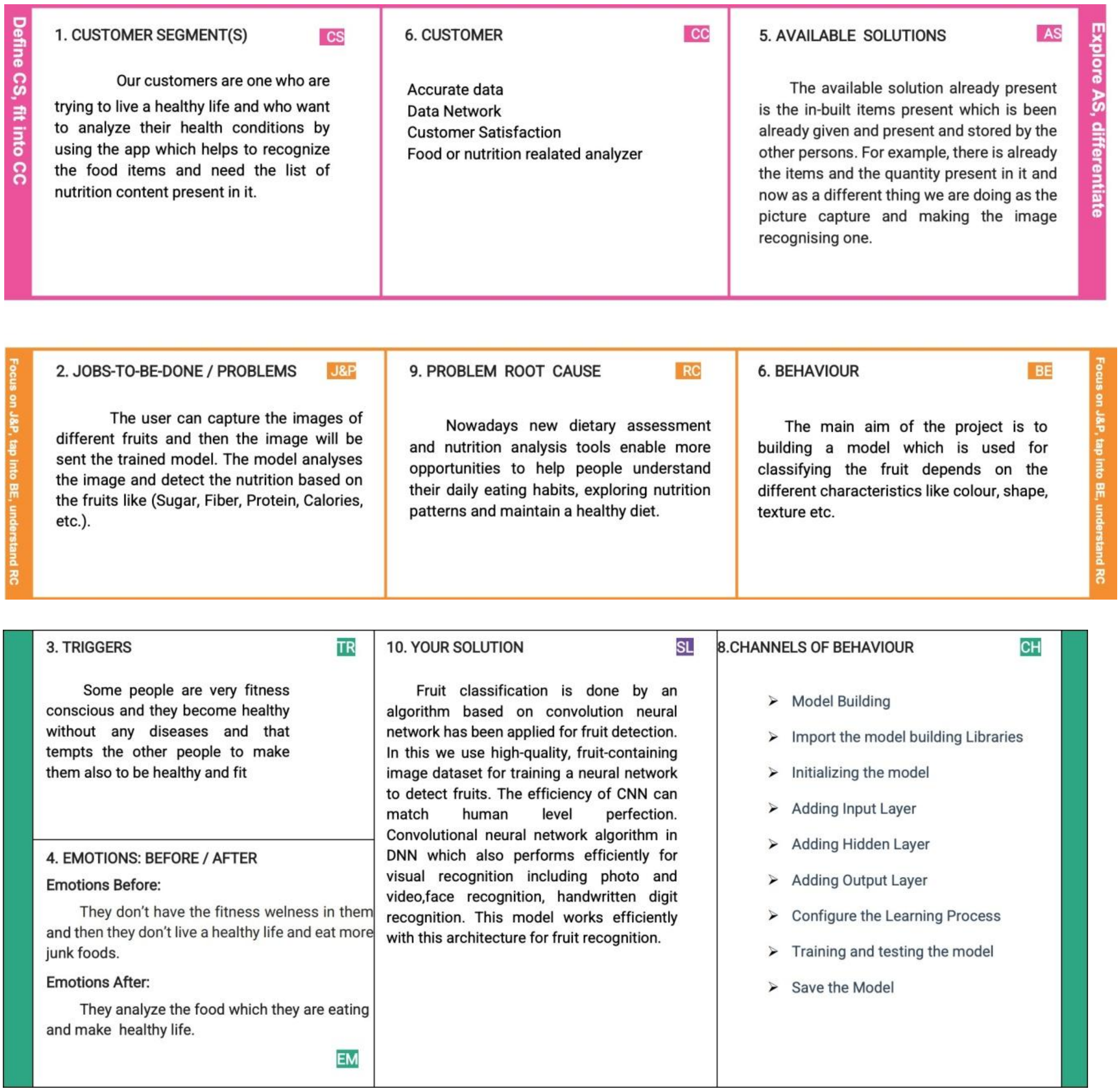
Flatten Layer:

It takes the output of the previous layers,flattens them and turns them into a single vector that can be input for the next stage.

Fully Connected Layer:

It applies weights to predict the correct label and gives final probabilities.

Problem Solution fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Our customers are one who are trying to live a healthy life and who want to analyze their health conditions by using the app which helps to recognize the food items and need the list of nutrition content present in it.

6. CUSTOMER CC

Accurate data
Data Network
Customer Satisfaction
Food or nutrition realated analyzer

5. AVAILABLE SOLUTIONS AS

The available solution already present is the in-built items present which is been already given and present and stored by the other persons. For example, there is already the items and the quantity present in it and now as a different thing we are doing as the picture capture and making the image recognising one.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

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

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, Fiber, Protein, Calories, etc.).

9. PROBLEM ROOT CAUSE RC

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.

6. BEHAVIOUR BE

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.

Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR

Some people are very fitness conscious and they become healthy without any diseases and that tempts the other people to make them also to be healthy and fit

4. EMOTIONS: BEFORE / AFTER

Emotions Before:

They don't have the fitness welness in them and then they don't live a healthy life and eat more junk foods.

Emotions After:

They analyze the food which they are eating and make healthy life.

EM

10. YOUR SOLUTION SL

Fruit classification is done by an algorithm based on convolution neural network has been applied for fruit detection. In this we use high-quality, fruit-containing image dataset for training a neural network to detect fruits. The efficiency of CNN can match human level perfection. Convolutional neural network algorithm in DNN which also performs efficiently for visual recognition including photo and video,face recognition, handwritten digit recognition. This model works efficiently with this architecture for fruit recognition.

8.CHANNELS OF BEHAVIOUR CH

- ➢ Model Building
- ➢ Import the model building Libraries
- ➢ Initializing the model
- ➢ Adding Input Layer
- ➢ Adding Hidden Layer
- ➢ Adding Output Layer
- ➢ Configure the Learning Process
- ➢ Training and testing the model
- ➢ Save the Model

CHAPTER 4

REQUIREMENT ANALYSIS

Functional Requirement:

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	HTML Page creation	To give input image
FR-2	Flask Module	To run the application
FR-3	Database	For storing the information
FR-4	Dataset	Uploading the dataset consists of variety of fruit images
FR-5	Image Input	Analysing the image input provided by the user
FR-6	Process	Testing the image by various convolution layers
FR-7	Result	The nutrient content in the particular fruit is displayed

Table 4.1.1

In this functional requirement,we first collect the dataset and do image preprocessing and build model using CNN layers and predict the required output.

Non-Functional Requirements:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The ways in which the system will accessible to users with particular or non standard accessibility requirements
NFR-2	**Security**	Extend to which data needs to be protected and kept confidential.
NFR-3	**Capacity**	Ability to continue to function well as it changes in according to input that we give.
NFR-4	**Performance**	Fast response is achieved.
NFR-5	**Availability**	The minimum proportion of time given in online service hours that the service should be available.
NFR-6	**Scalability**	Ability of the application to handle an increase in workload without performance degradation, or its ability to quickly enlarge.

Table 4.2.1

CHAPTER 5

PROJECT DESIGN

Data Flow Diagrams

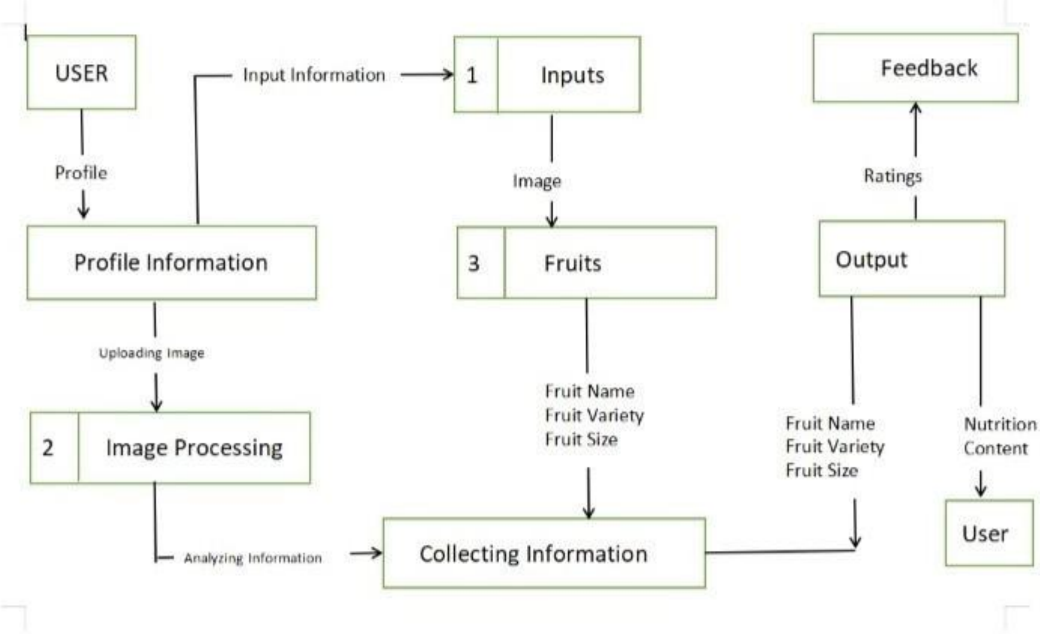


figure 5.1.1

Our team used the flask module to build the application and used colab for model building and we have added some addition nutritional information .

Solution & Technical Architecture

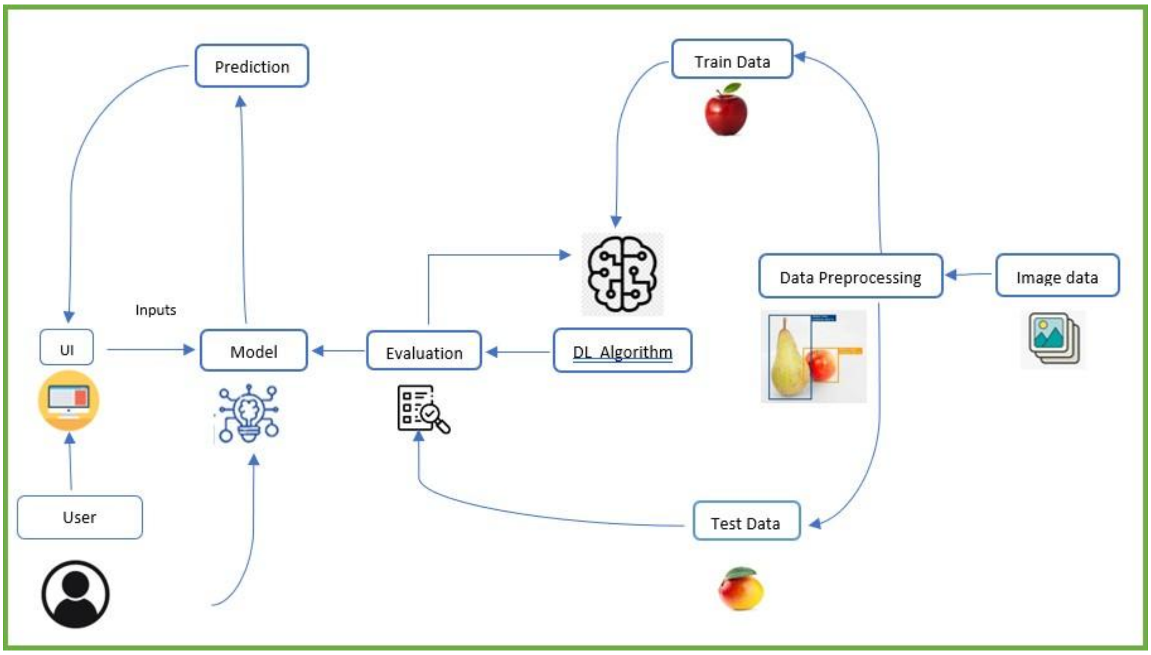


figure 5.2.1

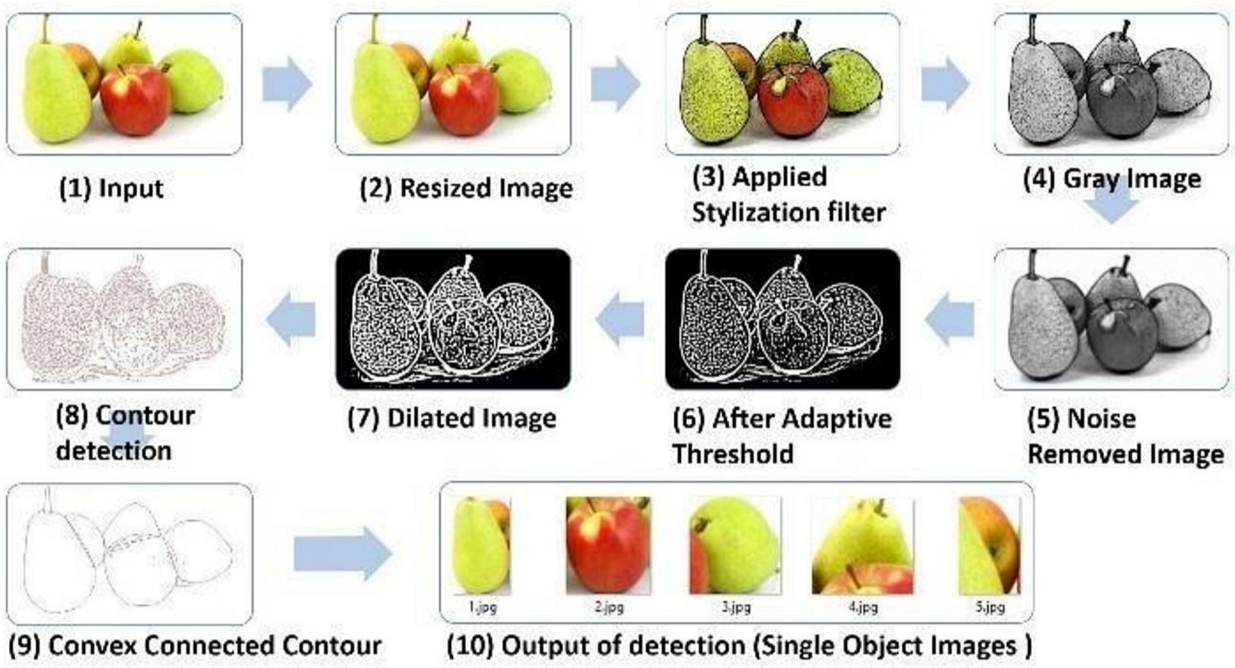
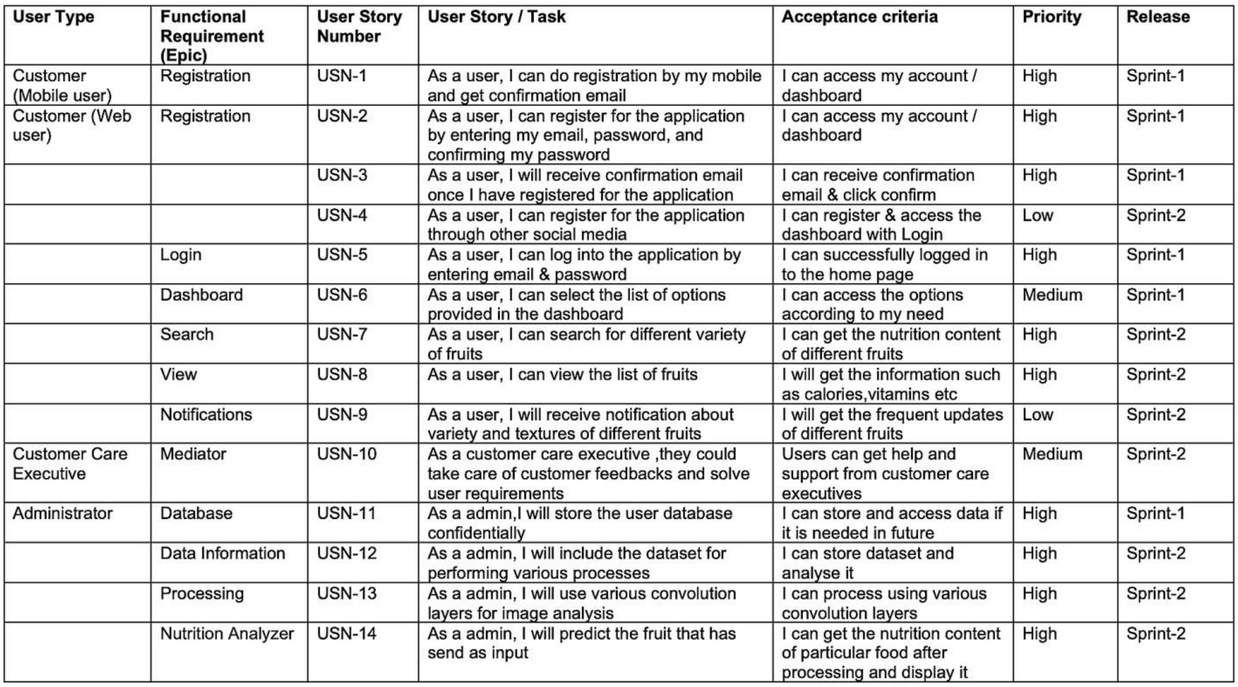

(1) Input
(2) Resized Image
(3) Applied Stylization filter
(4) Gray Image
(5) Noise Removed Image
(6) After Adaptive Threshold
(7) Dilated Image
(8) Contour detection
(9) Convex Connected Contour
1.jpg
2.jpg
3.jpg
4.jpg
5.jpg
(10) Output of detection (Single Object Images)

figure 5.2.2

User Stories



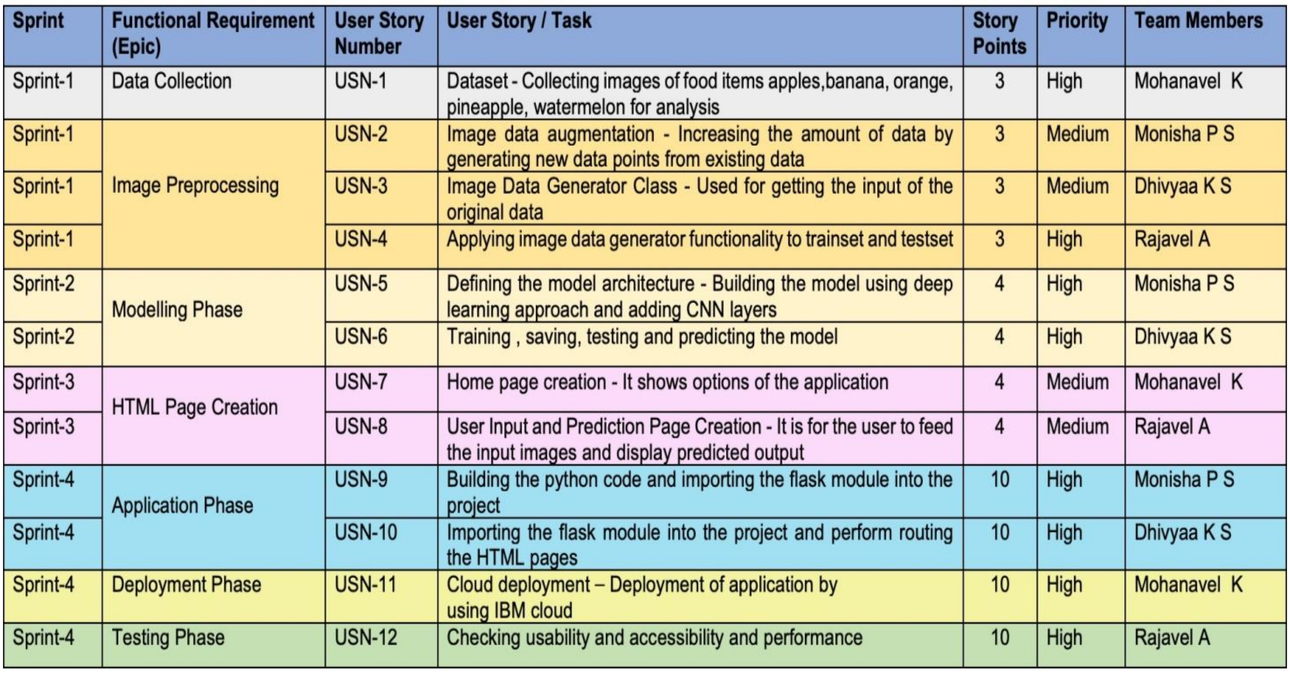
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can do registration by my mobile and get confirmation email	I can access my account / dashboard	High	Sprint-1
Customer (Web user)	Registration	USN-2	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-3	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-4	As a user, I can register for the application through other social media	I can register & access the dashboard with Login	Low	Sprint-2
	Login	USN-5	As a user, I can log into the application by entering email & password	I can successfully logged in to the home page	High	Sprint-1
	Dashboard	USN-6	As a user, I can select the list of options provided in the dashboard	I can access the options according to my need	Medium	Sprint-1
	Search	USN-7	As a user, I can search for different variety of fruits	I can get the nutrition content of different fruits	High	Sprint-2
	View	USN-8	As a user, I can view the list of fruits	I will get the information such as calories,vitamins etc	High	Sprint-2
	Notifications	USN-9	As a user, I will receive notification about variety and textures of different fruits	I will get the frequent updates of different fruits	Low	Sprint-2
Customer Care Executive	Mediator	USN-10	As a customer care executive ,they could take care of customer feedbacks and solve user requirements	Users can get help and support from customer care executives	Medium	Sprint-2
Administrator	Database	USN-11	As a admin,I will store the user database confidentially	I can store and access data if it is needed in future	High	Sprint-1
	Data Information	USN-12	As a admin, I will include the dataset for performing various processes	I can store dataset and analyse it	High	Sprint-2
	Processing	USN-13	As a admin, I will use various convolution layers for image analysis	I can process using various convolution layers	High	Sprint-2
	Nutrition Analyzer	USN-14	As a admin, I will predict the fruit that has send as input	I can get the nutrition content of particular food after processing and display it	High	Sprint-2

figure 5.3.1

CHAPTER 6

PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Dataset - Collecting images of food items apples,banana, orange, pineapple, watermelon for analysis	3	High	Mohanavel K
Sprint-1	Image Preprocessing	USN-2	Image data augmentation - Increasing the amount of data by generating new data points from existing data	3	Medium	Monisha P S
Sprint-1		USN-3	Image Data Generator Class - Used for getting the input of the original data	3	Medium	Dhivyaa K S
Sprint-1		USN-4	Applying image data generator functionality to trainset and testset	3	High	Rajavel A
Sprint-2	Modelling Phase	USN-5	Defining the model architecture - Building the model using deep learning approach and adding CNN layers	4	High	Monisha P S
Sprint-2		USN-6	Training , saving, testing and predicting the model	4	High	Dhivyaa K S
Sprint-3	HTML Page Creation	USN-7	Home page creation - It shows options of the application	4	Medium	Mohanavel K
Sprint-3		USN-8	User Input and Prediction Page Creation - It is for the user to feed the input images and display predicted output	4	Medium	Rajavel A
Sprint-4	Application Phase	USN-9	Building the python code and importing the flask module into the project	10	High	Monisha P S
Sprint-4		USN-10	Importing the flask module into the project and perform routing the HTML pages	10	High	Dhivyaa K S
Sprint-4	Deployment Phase	USN-11	Cloud deployment – Deployment of application by using IBM cloud	10	High	Mohanavel K
Sprint-4	Testing Phase	USN-12	Checking usability and accessibility and performance	10	High	Rajavel A

figure 6.1.1

SPRINT - 1

DataCollection:

Dataset:

1. In our dataset we have collected images of the five variety of fruits.

- Apple
- Orange
- Pineapple
- Watermelon
- Banana
- Drive link:

https://drive.google.com/file/d/1hgEWyKicgrntbY5LSkuW_v6G4C93AQfN/view?usp=share_link

Apple Banana Orange Pineapple Watermelon

figure 6.1.2

Image Preprocessing:

a. **Importing The ImageDataGenerator Library**

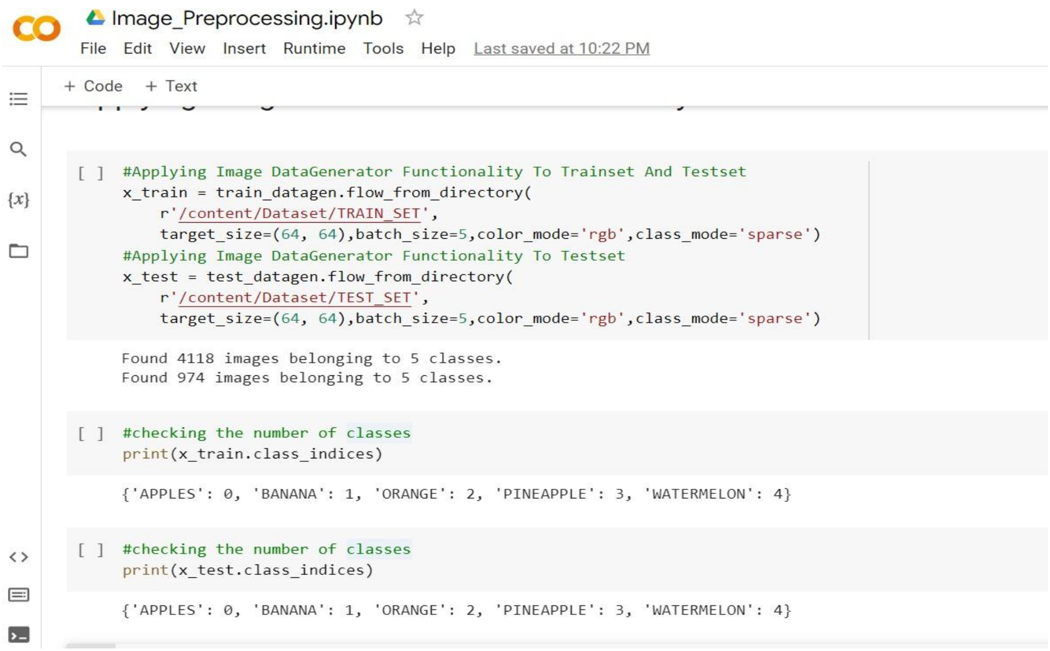
from keras.preprocessing.image import ImageDataGenerator

b. **Configuring ImageDataGenerator Class**

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

c. **Applying Image DataGenerator Functionality To Trainset And Testset**

```
x_train = train_datagen.flow_from_directory(r'/content/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/Dataset/TEST_SET',
   target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```



CO
Image_Preprocessing.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at 10:22 PM
+ Code + Text

```
#Applying Image DataGenerator Functionality To Trainset And Testset
x_train = train_datagen.flow_from_directory(
    r'/content/Dataset/TRAIN_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
#Applying Image DataGenerator Functionality To Testset
x_test = test_datagen.flow_from_directory(
    r'/content/Dataset/TEST_SET',
    target_size=(64, 64),batch_size=5,color_mode='rgb',class_mode='sparse')
```

```
Found 4118 images belonging to 5 classes.
Found 974 images belonging to 5 classes.
```

```
#checking the number of classes
print(x_train.class_indices)
```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
```

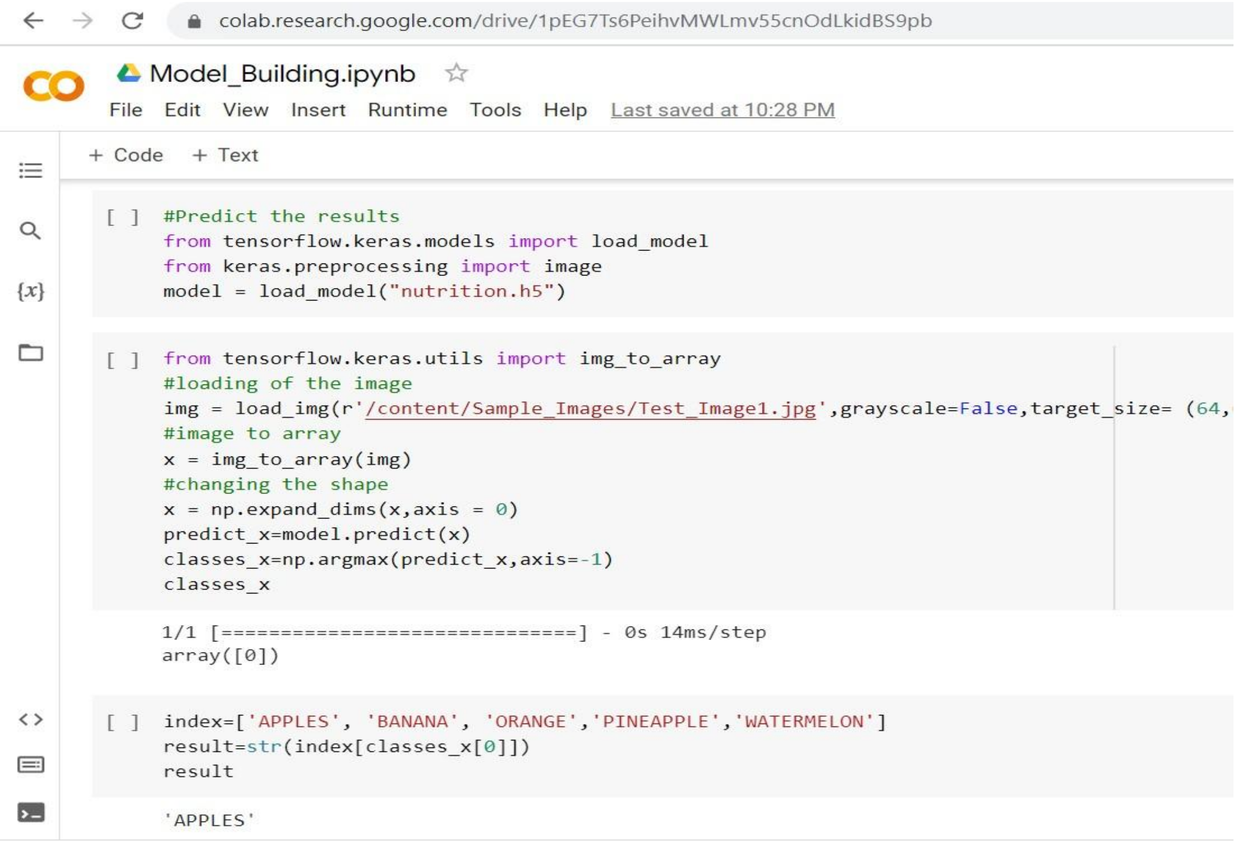
```
#checking the number of classes
print(x_test.class_indices)
```

```
{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4}
```

figure 6.1.3

SPRINT - 2

Model Building:



colab.research.google.com/drive/1pEG7Ts6PeihvMWLmv55cnOdLkidBS9pb
Model_Building.ipynb
File Edit View Insert Runtime Tools Help Last saved at 10:28 PM
+ Code + Text

```
#Predict the results
from tensorflow.keras.models import load_model
from keras.preprocessing import image
model = load_model("nutrition.h5")
```

```
from tensorflow.keras.utils import img_to_array
#loading of the image
img = load_img(r'/content/Sample_Images/Test_Image1.jpg',grayscale=False,target_size= (64,
#image to array
x = img_to_array(img)
#changing the shape
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 14ms/step
array([0])
```

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

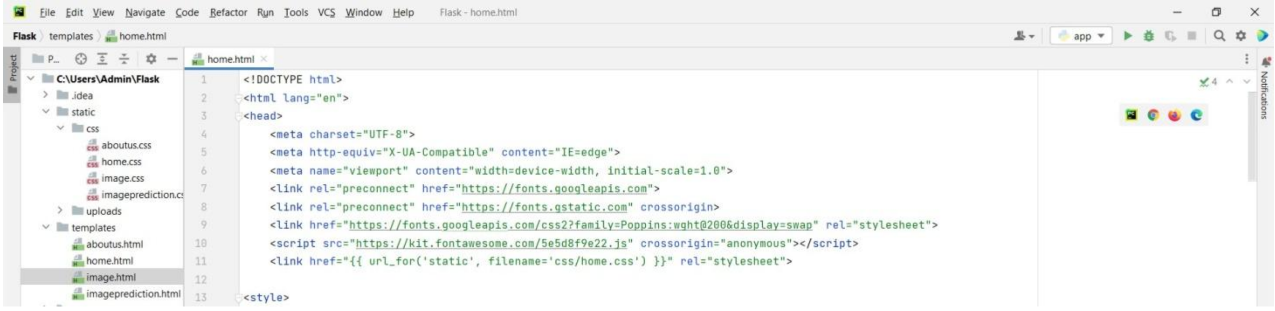
```
'APPLES'
```

figure 6.1.4

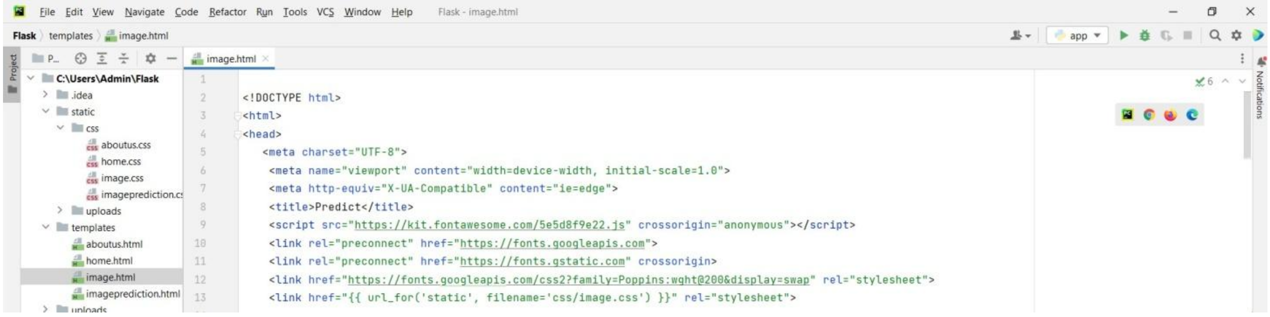
1. Importing The Model Building Libraries
2. Initializing The Model
3. Adding CNN Layers
4. Adding Dense Layers
5. Configure The Learning Process
6. Train the model
7. Save the model
8. Test the model

SPRINT - 3

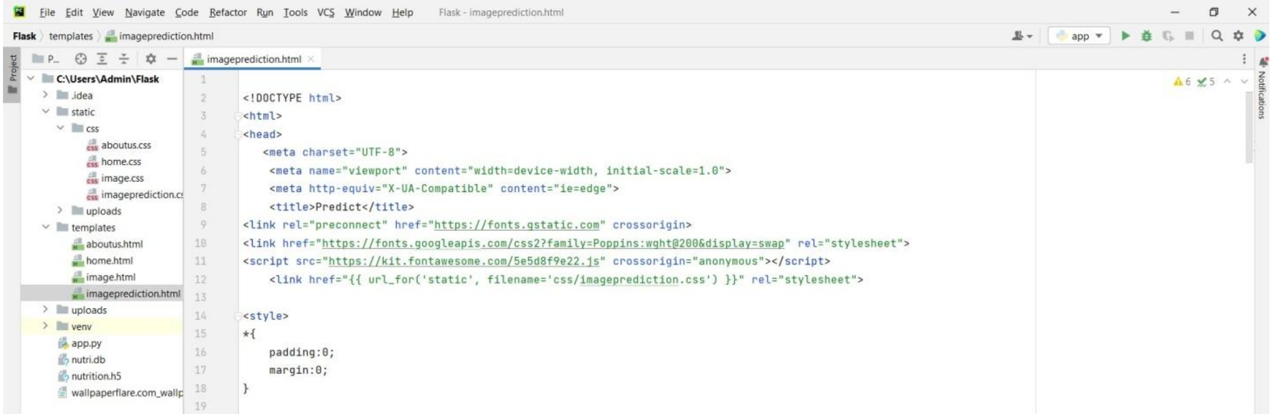
HTML Page Creation



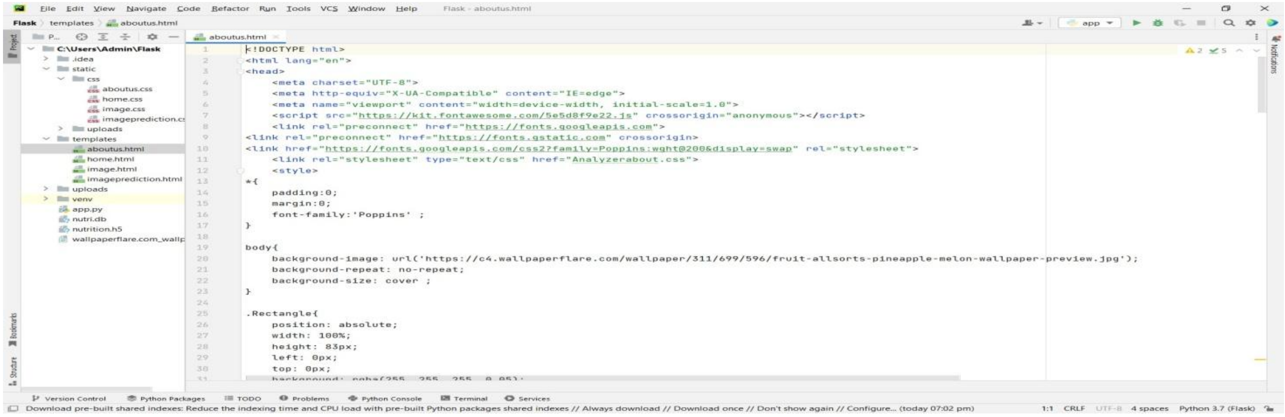
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@200&display=swap" rel="stylesheet">
    <script src="https://kit.fontawesome.com/5e5d8f9e22.js" crossorigin="anonymous"></script>
    <link href="{{ url_for('static', filename='css/home.css') }}" rel="stylesheet">

<style>
```



```

<!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>
    <script src="https://kit.fontawesome.com/5e5d8f9e22.js" crossorigin="anonymous"></script>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@200&display=swap" rel="stylesheet">
    <link href="{{ url_for('static', filename='css/image.css') }}" rel="stylesheet">
```



```

<!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 rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@200&display=swap" rel="stylesheet">
<script src="https://kit.fontawesome.com/5e5d8f9e22.js" crossorigin="anonymous"></script>
    <link href="{{ url_for('static', filename='css/imageprediction.css') }}" rel="stylesheet">

<style>
*{
    padding:0;
    margin:0;
}

```



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://kit.fontawesome.com/5e5d8f9e22.js" crossorigin="anonymous"></script>
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@200&display=swap" rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="Analyzerabout.css">
    <style>
*{
    padding:0;
    margin:0;
    font-family:'Poppins' ;
}

body{
    background-image: url('https://c4.wallpaperflare.com/wallpaper/311/699/596/fruit-allsorts-pineapple-melon-wallpaper-preview.jpg');
    background-repeat: no-repeat;
    background-size: cover ;
}

.Rectangle{
    position: absolute;
    width: 100%;
    height: 83px;
    left: 0px;
    top: 0px;
```

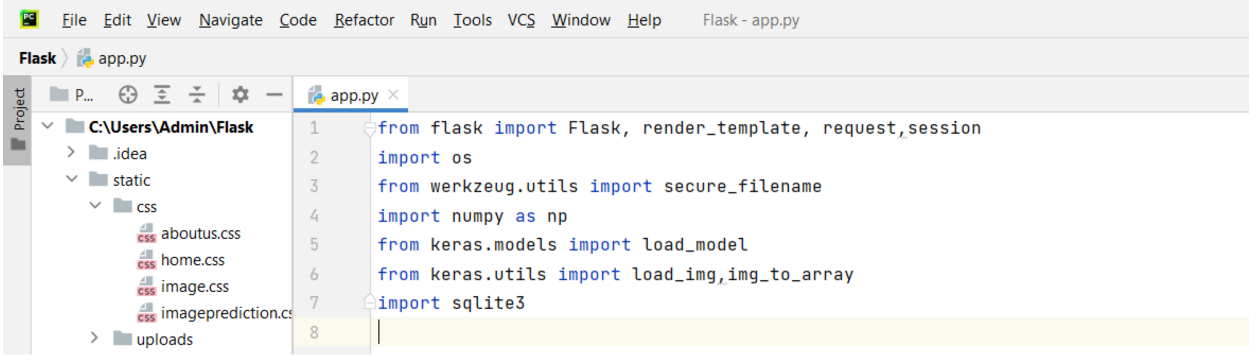
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure... (today 07:02 pm)

figure 6.1.5

SPRINT - 4

Application Building

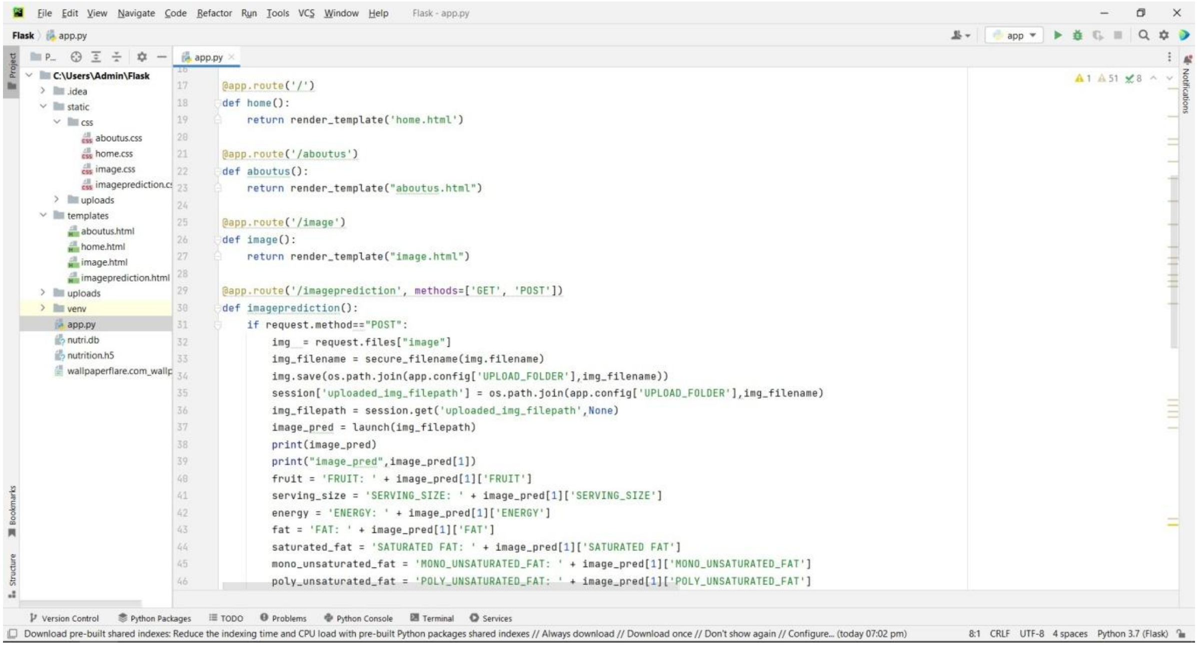
Importing Flask into app.py



```
from flask import Flask, render_template, request,session
import os
from werkzeug.utils import secure_filename
import numpy as np
from keras.models import load_model
from keras.utils import load_img,img_to_array
import sqlite3
```

figure 6.1.6

Routing to HTML Pages



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

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

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

@app.route('/imageprediction', methods=['GET', 'POST'])
def imageprediction():
    if request.method=="POST":
        img  = request.files["image"]
        img_filename = secure_filename(img.filename)
        img.save(os.path.join(app.config['UPLOAD_FOLDER'],img_filename))
        session['uploaded_img_filepath'] = os.path.join(app.config['UPLOAD_FOLDER'],img_filename)
        img_filepath = session.get('uploaded_img_filepath',None)
        image_pred = launch(img_filepath)
        print(image_pred)
        print("image_pred",image_pred[1])
        fruit = 'FRUIT: ' + image_pred[1]['FRUIT']
        serving_size = 'SERVING_SIZE: ' + image_pred[1]['SERVING_SIZE']
        energy = 'ENERGY: ' + image_pred[1]['ENERGY']
        fat = 'FAT: ' + image_pred[1]['FAT']
        saturated_fat = 'SATURATED FAT: ' + image_pred[1]['SATURATED FAT']
        mono_unsaturated_fat = 'MONO_UNSATURATED_FAT: ' + image_pred[1]['MONO_UNSATURATED_FAT']
        poly_unsaturated_fat = 'POLY_UNSATURATED_FAT: ' + image_pred[1]['POLY_UNSATURATED_FAT']
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure... (today 07:02 pm)

figure 6.1.7

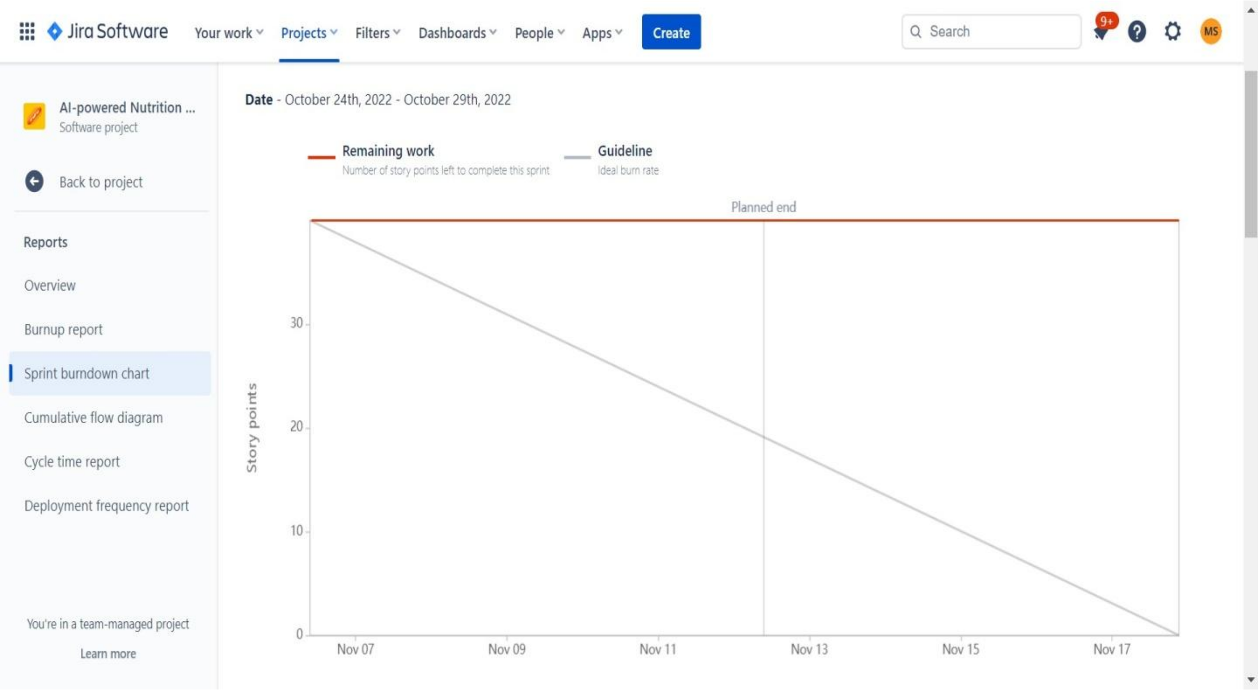
Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)
Sprint-1	12	6 Days	24 Oct 2022	29 Oct 2022
Sprint-2	8	4 Days	30 Oct 2022	02 Nov 2022
Sprint-3	8	3 Days	03 Nov 2022	05 Nov 2022
Sprint-4	40	7 Days	06 Nov 2022	12 Nov 2022

figure 6.2.1

Reports from JIRA

Burndown Chart



Date - October 24th, 2022 - October 29th, 2022

figure 6.3.1

RoadMap

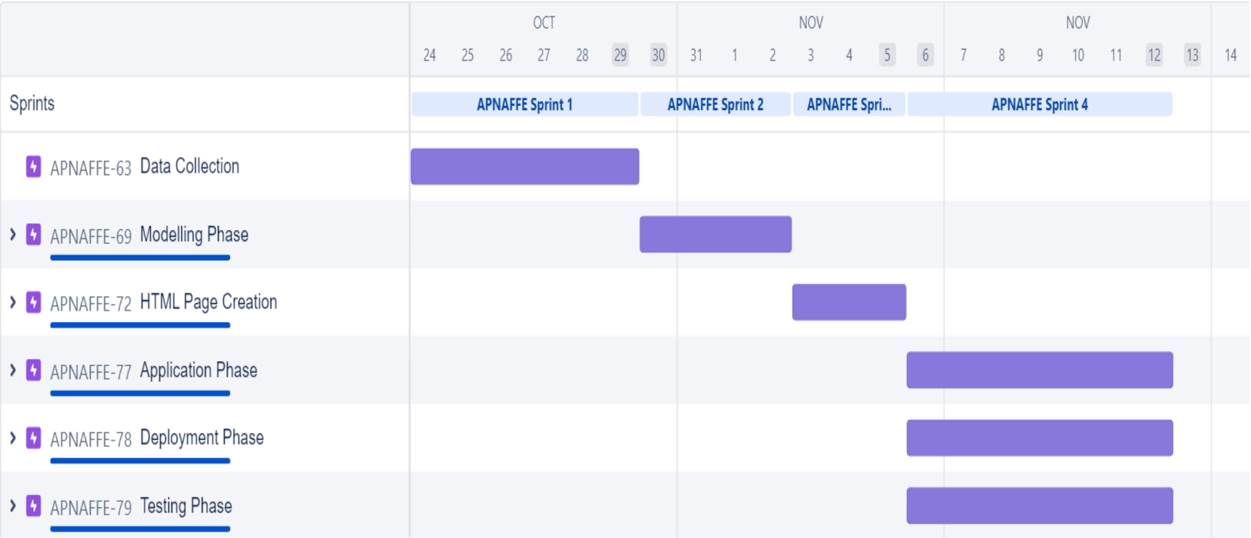


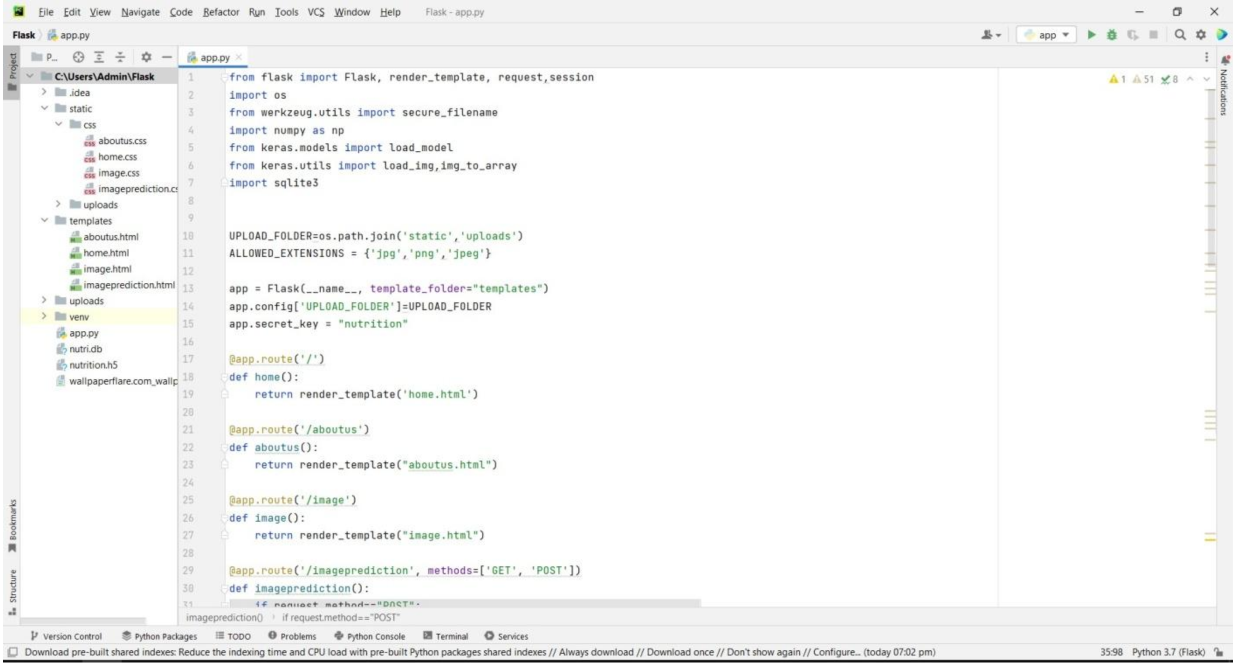
figure 6.3.2

CHAPTER 7

CODING & SOLUTIONING

Feature 1

Python Flask Code



```python
from flask import Flask, render_template, request,session
import os
from werkzeug.utils import secure_filename
import numpy as np
from keras.models import load_model
from keras.utils import load_img,img_to_array
import sqlite3


UPLOAD_FOLDER=os.path.join('static','uploads')
ALLOWED_EXTENSIONS = {'jpg','png','jpeg'}

app = Flask(__name__, template_folder="templates")
app.config['UPLOAD_FOLDER']=UPLOAD_FOLDER
app.secret_key = "nutrition"

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

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

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

@app.route('/imageprediction', methods=['GET', 'POST'])
def imageprediction():
    if request.method=="POST":
```

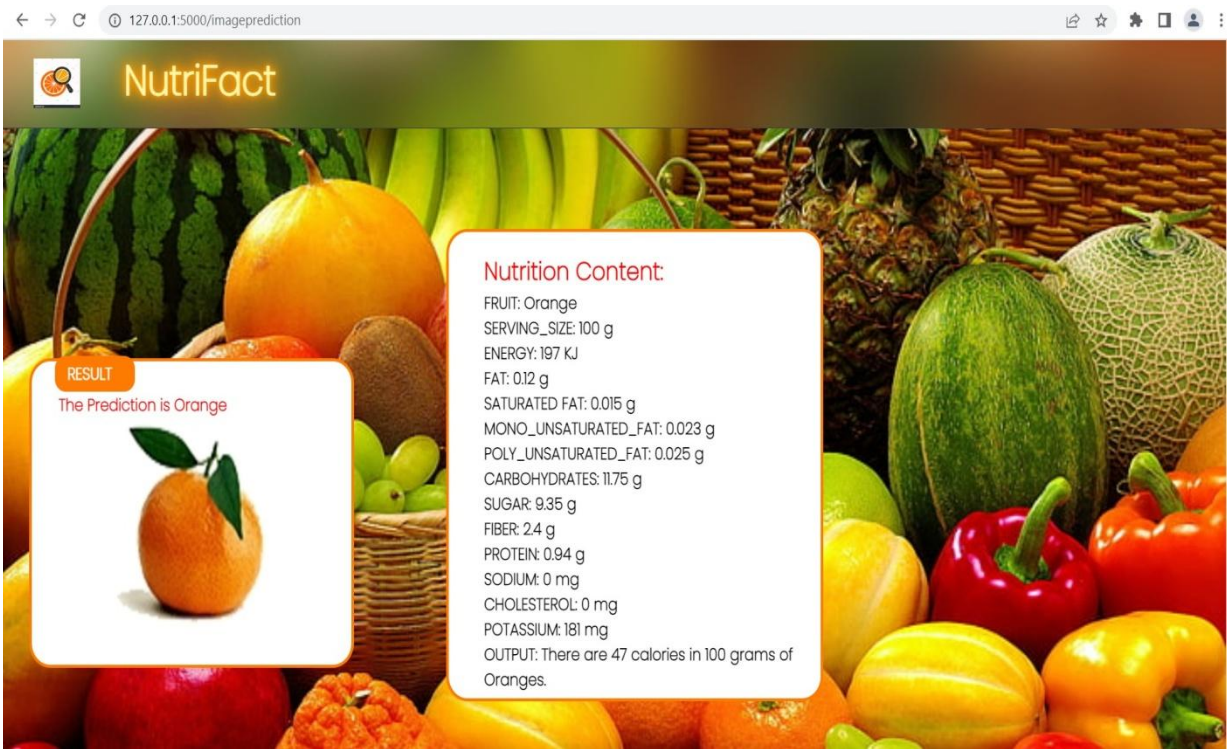
figure 7.1.1

```python
from flask import Flask, render_template, request,session
import os
from werkzeug.utils import secure_filename
import numpy as np
from keras.models import load_model
from keras.utils import load_img,img_to_array
import sqlite3
UPLOAD_FOLDER=os.path.join('static','uploads')
ALLOWED_EXTENSIONS = {'jpg','png','jpeg'}
app = Flask(__name__, template_folder="templates")
app.config['UPLOAD_FOLDER']=UPLOAD_FOLDER
app.secret_key = "nutrition"
@app.route('/')
def home():
    return render_template('home.html')

@app.route('/aboutus')
```

```
def aboutus():

   return render_template("aboutus.html")

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

@app.route('/imageprediction', methods=['GET', 'POST'])
def imageprediction():
    if request.method=="POST":
        img  = request.files["image"]
        img_filename = secure_filename(img.filename)
        img.save(os.path.join(app.config['UPLOAD_FOLDER'],img_filename))
        session['uploaded_img_filepath'] =
os.path.join(app.config['UPLOAD_FOLDER'],img_filename)
        img_filepath = session.get('uploaded_img_filepath',None)
        image_pred = launch(img_filepath)
        print(image_pred)
        print("image_pred",image_pred[1])
        fruit = 'FRUIT: ' + image_pred[1]['FRUIT']
        serving_size = 'SERVING_SIZE: ' + image_pred[1]['SERVING_SIZE']
        energy = 'ENERGY: ' + image_pred[1]['ENERGY']
        fat = 'FAT: ' + image_pred[1]['FAT']
        saturated_fat = 'SATURATED FAT: ' + image_pred[1]['SATURATED FAT']
        mono_unsaturated_fat = 'MONO_UNSATURATED_FAT: ' +
image_pred[1]['MONO_UNSATURATED_FAT']
        poly_unsaturated_fat = 'POLY_UNSATURATED_FAT: ' +
image_pred[1]['POLY_UNSATURATED_FAT']
        carbohydrates = 'CARBOHYDRATES: ' + image_pred[1]['CARBOHYDRATES']
        sugar = 'SUGAR: ' + image_pred[1]['SUGAR']
        fiber = 'FIBER: ' + image_pred[1]['FIBER']
        protein = 'PROTEIN: ' + image_pred[1]['PROTEIN']
        sodium = 'SODIUM: ' + image_pred[1]['SODIUM']
        cholesterol = 'CHOLESTEROL: ' + image_pred[1]['CHOLESTEROL']
        potassium = 'POTASSIUM: ' + image_pred[1]['POTASSIUM']
        output = 'OUTPUT: ' + image_pred[1]['OUTPUT']
        return render_template("imageprediction.html", value=img_filepath,
pred=image_pred[0],
                fruit=fruit, serving_size=serving_size, energy=energy, fat=fat,
saturated_fat=saturated_fat,
                mono_unsaturated_fat=mono_unsaturated_fat,
poly_unsaturated_fat=poly_unsaturated_fat,
                carbohydrates=carbohydrates, sugar=sugar, fiber=fiber, protein=protein,
sodium=sodium,
                cholesterol=cholesterol, potassium=potassium, output=output, flag=True)
def launch(img_filepath):
    model = load_model('nutrition.h5')
    img = load_img(img_filepath, 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)
    index = ['Apple', 'Banana', 'Orange', 'Pineapple', 'Watermelon']
    values = nutrition(index[classes_x])
    return [index[classes_x],  values]

def nutrition(x):
    conn = sqlite3.connect('nutri.db')
    cursor = conn.execute(f'''SELECT * FROM NUTRI WHERE FRUIT=="{x}"''')
    for row in cursor:
       rec =
{"FRUIT":row[0],"SERVING_SIZE":row[1],"ENERGY":row[2],"FAT":row[3],"SATURATED
FAT":row[4], "MONO_UNSATURATED_FAT":row[5],
"POLY_UNSATURATED_FAT":row[6],"CARBOHYDRATES":row[7],"SUGAR":row[8],"FIBER":row[9],"PRO
TEIN":row[10],"SODIUM":row[11],"CHOLESTEROL":row[12],"POTASSIUM":row[13],"OUTPUT":row[1
4]}
    return rec
if __name__ == "__main__":
    app.run(debug=False)
```

Feature 2

Our team has introduced some additional information like monounsaturated fat,polyunsaturated fat,energy and total calories in particular fruit.So that the user will get more information about fruits.The name we give for this application is NutriFact.



The Prediction is Orange

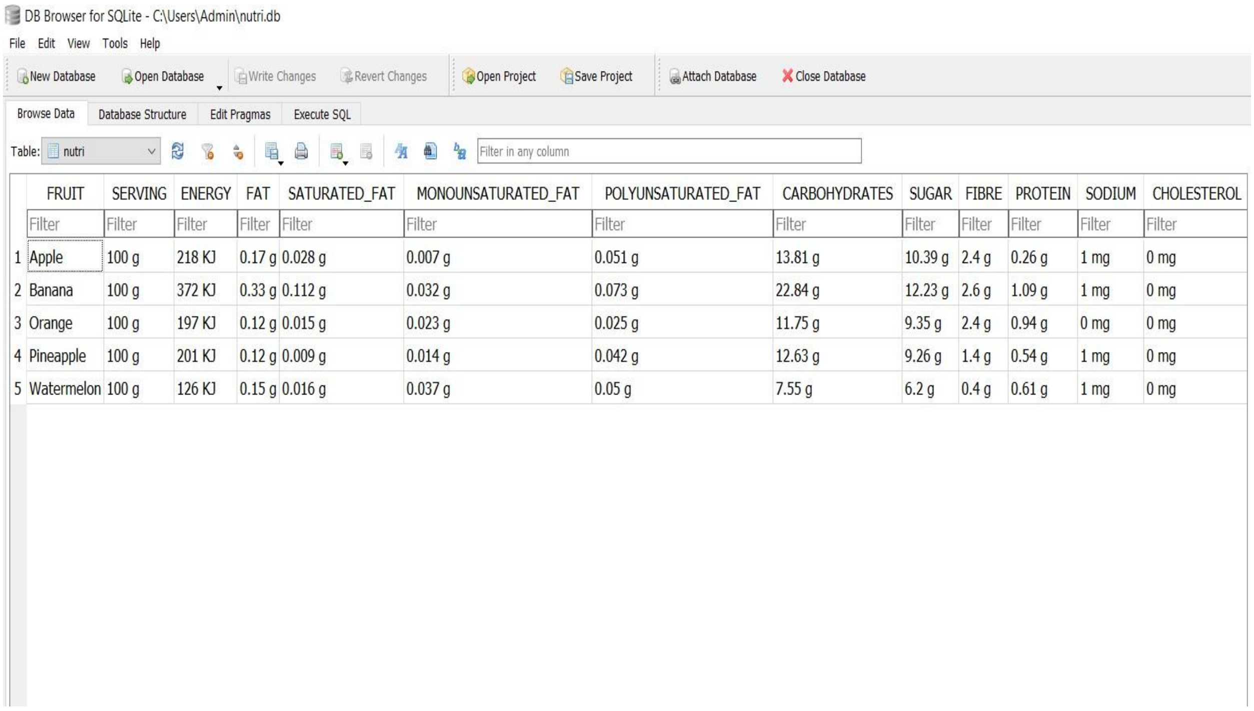
Nutrition Content:

FRUIT: Orange
SERVING_SIZE: 100 g
ENERGY: 197 KJ
FAT: 0.12 g
SATURATED FAT: 0.015 g
MONO_UNSATURATED_FAT: 0.023 g
POLY_UNSATURATED_FAT: 0.025 g
CARBOHYDRATES: 11.75 g
SUGAR: 9.35 g
FIBER: 2.4 g
PROTEIN: 0.94 g
SODIUM: 0 mg
CHOLESTEROL: 0 mg
POTASSIUM: 181 mg
OUTPUT: There are 47 calories in 100 grams of Oranges.

figure 7.2.1

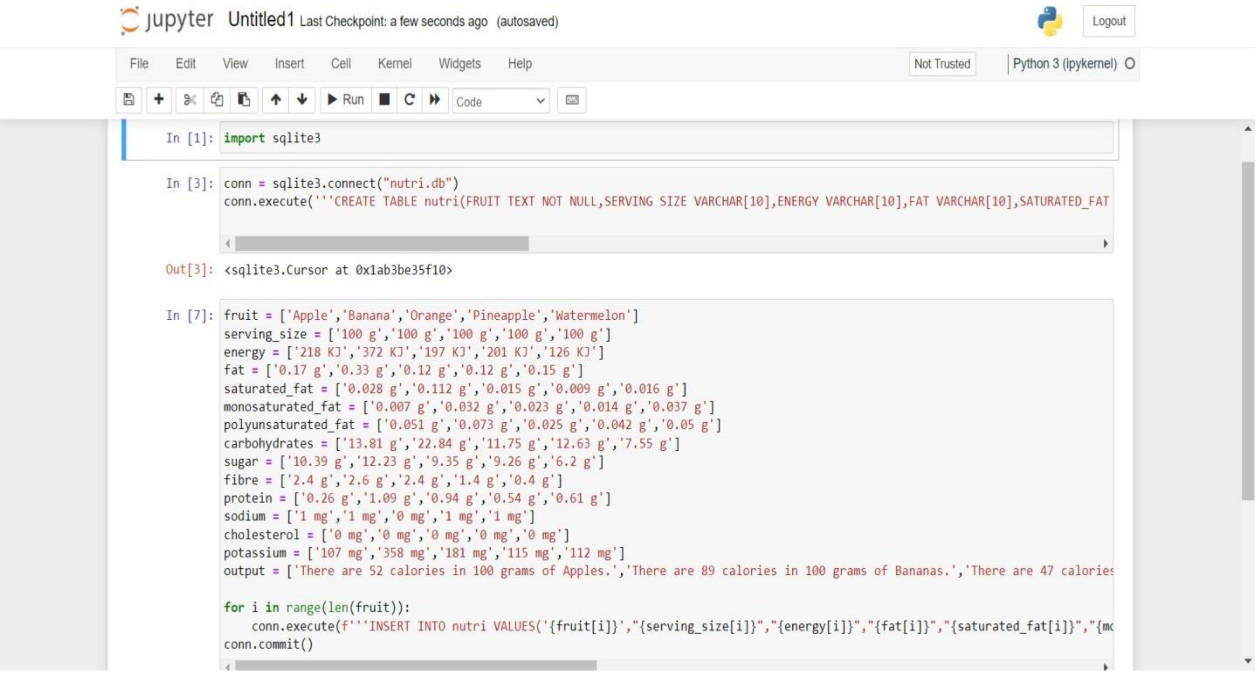
Database Schema

Database of fruits was created using jupyter notebook and implemented and an nutri.db file is generated after execution.The table for storing database is created and it can be viewed by using SQLlite Software.



	FRUIT	SERVING	ENERGY	FAT	SATURATED_FAT	MONOUNSATURATED_FAT	POLYUNSATURATED_FAT	CARBOHYDRATES	SUGAR	FIBRE	PROTEIN	SODIUM	CHOLESTEROL
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	Apple	100 g	218 KJ	0.17 g	0.028 g	0.007 g	0.051 g	13.81 g	10.39 g	2.4 g	0.26 g	1 mg	0 mg
2	Banana	100 g	372 KJ	0.33 g	0.112 g	0.032 g	0.073 g	22.84 g	12.23 g	2.6 g	1.09 g	1 mg	0 mg
3	Orange	100 g	197 KJ	0.12 g	0.015 g	0.023 g	0.025 g	11.75 g	9.35 g	2.4 g	0.94 g	0 mg	0 mg
4	Pineapple	100 g	201 KJ	0.12 g	0.009 g	0.014 g	0.042 g	12.63 g	9.26 g	1.4 g	0.54 g	1 mg	0 mg
5	Watermelon	100 g	126 KJ	0.15 g	0.016 g	0.037 g	0.05 g	7.55 g	6.2 g	0.4 g	0.61 g	1 mg	0 mg

figure 7.3.1



```
In [1]: import sqlite3

In [3]: conn = sqlite3.connect("nutri.db")
        conn.execute('''CREATE TABLE nutri(FRUIT TEXT NOT NULL,SERVING SIZE VARCHAR[10],ENERGY VARCHAR[10],FAT VARCHAR[10],SATURATED_FAT

Out[3]: <sqlite3.Cursor at 0x1ab3be35f10>

In [7]: fruit = ['Apple','Banana','Orange','Pineapple','Watermelon']
        serving_size = ['100 g','100 g','100 g','100 g','100 g']
        energy = ['218 KJ','372 KJ','197 KJ','201 KJ','126 KJ']
        fat = ['0.17 g','0.33 g','0.12 g','0.12 g','0.15 g']
        saturated_fat = ['0.028 g','0.112 g','0.015 g','0.009 g','0.016 g']
        monosaturated_fat = ['0.007 g','0.032 g','0.023 g','0.014 g','0.037 g']
        polyunsaturated_fat = ['0.051 g','0.073 g','0.025 g','0.042 g','0.05 g']
        carbohydrates = ['13.81 g','22.84 g','11.75 g','12.63 g','7.55 g']
        sugar = ['10.39 g','12.23 g','9.35 g','9.26 g','6.2 g']
        fibre = ['2.4 g','2.6 g','2.4 g','1.4 g','0.4 g']
        protein = ['0.26 g','1.09 g','0.94 g','0.54 g','0.61 g']
        sodium = ['1 mg','1 mg','0 mg','1 mg','1 mg']
        cholesterol = ['0 mg','0 mg','0 mg','0 mg','0 mg']
        potassium = ['107 mg','358 mg','181 mg','115 mg','112 mg']
        output = ['There are 52 calories in 100 grams of Apples.','There are 89 calories in 100 grams of Bananas.','There are 47 calories

        for i in range(len(fruit)):
            conn.execute(f'''INSERT INTO nutri VALUES('{fruit[i]}',"{serving_size[i]}","{energy[i]}","{fat[i]}","{saturated_fat[i]}","{mo
        conn.commit()
```

figure 7.3.2

CHAPTER 8

TESTING

Test Cases

A test case might be created as an automated script to verify the functionality as per the original acceptance criteria.

Test case ID	Test case ID	Test case ID	Test case ID
Model Building_TC_OO1	Training and Testing	Python	Verify whether the image prediction is proper or not
Backend_TC_OO2	App Configuration	Python	It will get data from front end and process it
Frontend_TC_OO3	UI	Home page(user),user input Page,image prediction page page,about us page	user can give input as jpg,jpeg,png format and display output
Datebase_TC_OO4	Prediction	Python	Verify that it display the information as correct

Table 8.1.1

Steps To Execute	Test Data	Expected Result	Status	Executed By
1. Importing dataset and unzip it 2. Image preprocessing 3. Add convolution layers and predict fruit	http://127.0.0.1:5000/	Predict the fruit	Pass	Rajavel A
1. APP configuration 2. APP Route	http://127.0.0.1:5000/	Users data should process In Backend it should get data from frontend and display output	Pass	Monisha P S
1. Enter the input image 2. Pick the image format as jpg,png,jpeg 3. Click submit	http://127.0.0.1:5000/ http://127.0.0.1:5000/image http://127.0.0.1:5000/imageprediction http://127.0.0.1:5000/aboutus	User should navigate to home page and required pages they want to go.	Pass	Dhivyaa K S
1. Declaring the table using database code 2. Data should store with the various features of fruit	http://127.0.0.1:5000/	Display data from given database created	Pass	Mohanavel K

Table 8.1.2

User Acceptance Testing

Defect Analysis:

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	3	2	1	14
Duplicate	0	1	0	2	3
External	3	2	0	1	6
Fixed	6	1	3	12	22
Not Reproduced	0	0	0	1	1
Skipped	0	0	1	0	1
Won't Fix	0	1	0	0	1
Totals	15	08	6	17	48

Table 8.2.1

Testcase Analysis:

Section	Total Cases	Not Tested	Fail	Pass
Routing to pages	2	0	0	2
Input Page	4	0	0	4
Storing Image	4	0	0	4
Image Prediction	5	0	0	5
Final Output	5	0	0	5

Table 8.2.2

CHAPTER 9

RESULTS

Performance Metrics

NFT - Risk Assessment

Project Name	Scope	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Risk Score
AI-powered Nutrition Analyzer for Fitness Enthusiats	New	Low	Moderate	Moderate	Low	GREEN

Table 9.1.1

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	Display nutrient content	Stress	App Crash/ Developer team/ Site Down	Approved
2	Display nutrient content	Load	Server Crash/ Developer team	Approved

Table 9.1.2

End Of Test Report

NFR - Met	Test Outcome	GO/NO-GO Decision	Recommendations	Identified Defects (Detected/Closed/Open)
Performance	CPU -01	GO	High Performance	Closed
Database Information	Storage	NO-GO	SQLlite for access	Closed

Table 9.1.3

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

Advantages:

- The platform provides nutrition-based data services, analytics, and technologies to its consumers .
- To enable individualised compilation of data, the platform uses NLP and mathematical models from the optimation theory and predictive analysis.
- The app uses AI and image classification technology to identify the food correctly and accurately and also calculated the amount the calories just from the picture.
- API with diverse data sets, the food which is automatically recognised by the platform is paired with detailed nutrition information.
- Online platform which uses deep learning and image recognition to analyse what the users eat and determine what is trending in terms of each popular dish that the user eats and consumption time.
- Time is saved and managed for the people who are using the website like this.

Disadvantages:

- Sometimes the fruits which are rotten that is also been analyzed.
- Only for the limited number of fruits the image & data has been processed and coded to give the output.
- It supports only the jpeg, png,jpeg file not the other format.If the file which is given in any other format it does not support it.
- It does not say the accurate fruits if the different types of fruit variety such as green apple or cut fruits it will not show the accurate fruit variety in it

CHAPTER 11

11. CONCLUSION

NutriFact website help others to create awareness for the fitness enthusiasits and created successfully using Pycharm and image input is accepted and it does image preprocessing and under convolution layers and predict the output .It will display the name of fruit that we give as image input and its nutritional facts. User can know more details and get quality information from our website.It helps in maintaining health and fitness.

NutriFact is understandable to all people so anyone can use this.It is very simple and easy to use. Users will definitely enjoy this application and know more intersting facts about fruits.

CHAPTER 12

FUTURE SCOPE

1. **Based On Reviews:**

In future the feedbacks are asked from users to meet their expectations like additional features such as BMI calculation,daily calorie intake calculator various features.

2. **Nutritional Chatbot:**

In future the nutrition chatbot will be used.It can have various type of input such as speech recognition,image input,input as name and will give output in form of voice or text .Analyzing according to the bowl size.We will try to take the disadvantages as the challenge and make it possible in the upcomings.

CHAPTER 13

APPENDIX

Source Code

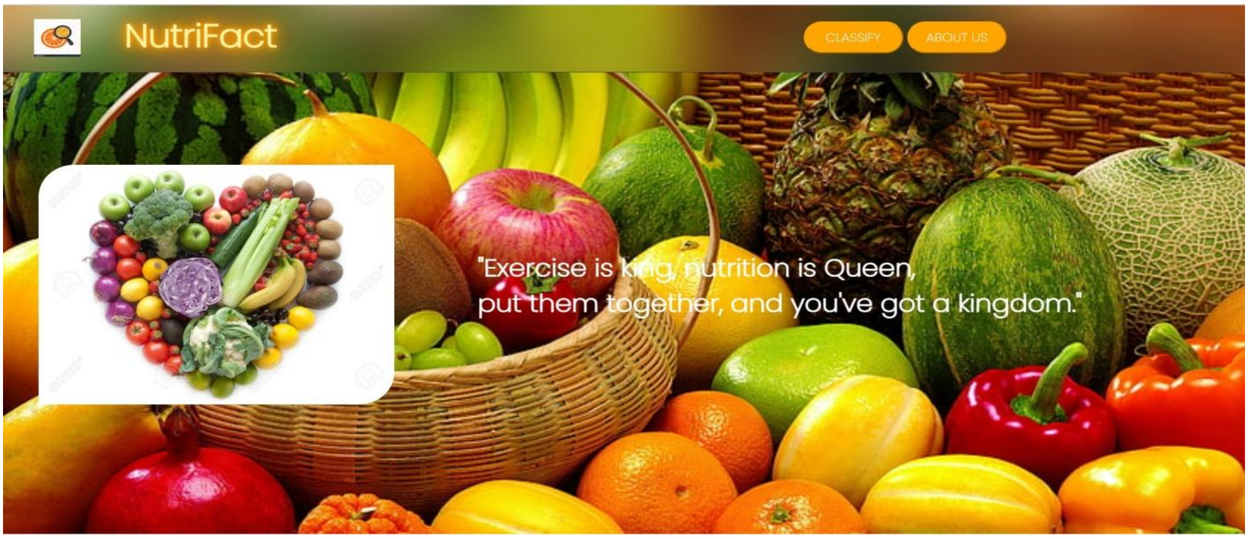
1. Python flask code - app.py

```
from flask import Flask, render_template, request,session
import os
from werkzeug.utils import secure_filename
import numpy as np
from keras.models import load_model
from keras.utils import load_img,img_to_array
import sqlite3


UPLOAD_FOLDER=os.path.join('static','uploads')
ALLOWED_EXTENSIONS = {'jpg','png','jpeg'}
app = Flask(__name__, template_folder="templates")
app.config['UPLOAD_FOLDER']=UPLOAD_FOLDER
app.secret_key = "nutrition"
@app.route('/')
def home():
    return render_template('home.html')
@app.route('/aboutus')
def aboutus():
    return render_template("aboutus.html")
@app.route('/image')
def image():
    return render_template("image.html")
@app.route('/imageprediction', methods=['GET', 'POST'])
def imageprediction():
    if request.method=="POST":
        img  = request.files["image"]
        img_filename = secure_filename(img.filename)
        img.save(os.path.join(app.config['UPLOAD_FOLDER'],img_filename))
        session['uploaded_img_filepath'] =
os.path.join(app.config['UPLOAD_FOLDER'],img_filename)
        img_filepath = session.get('uploaded_img_filepath',None)
        image_pred = launch(img_filepath)
        print(image_pred)
        print("image_pred",image_pred[1])
        fruit = 'FRUIT: ' + image_pred[1]['FRUIT']
        serving_size = 'SERVING_SIZE: ' + image_pred[1]['SERVING_SIZE']
        energy = 'ENERGY: ' + image_pred[1]['ENERGY']
        fat = 'FAT: ' + image_pred[1]['FAT']
        saturated_fat = 'SATURATED FAT: ' + image_pred[1]['SATURATED FAT']
```

```
 mono_unsaturated_fat = 'MONO_UNSATURATED_FAT: ' +
image_pred[1]['MONO_UNSATURATED_FAT']
poly_unsaturated_fat = 'POLY_UNSATURATED_FAT: ' + image_pred[1]['POLY_UNSATURATED_FAT']
        carbohydrates = 'CARBOHYDRATES: ' + image_pred[1]['CARBOHYDRATES']
        sugar = 'SUGAR: ' + image_pred[1]['SUGAR']
        fiber = 'FIBER: ' + image_pred[1]['FIBER']
        protein = 'PROTEIN: ' + image_pred[1]['PROTEIN']
        sodium = 'SODIUM: ' + image_pred[1]['SODIUM']
        cholesterol = 'CHOLESTEROL: ' + image_pred[1]['CHOLESTEROL']
        potassium = 'POTASSIUM: ' + image_pred[1]['POTASSIUM']
        output = 'OUTPUT: ' + image_pred[1]['OUTPUT']
        return render_template("imageprediction.html", value=img_filepath,
pred=image_pred[0],
                fruit=fruit, serving_size=serving_size, energy=energy, fat=fat,
saturated_fat=saturated_fat,
                mono_unsaturated_fat=mono_unsaturated_fat,
poly_unsaturated_fat=poly_unsaturated_fat,
                carbohydrates=carbohydrates,  sugar=sugar,  fiber=fiber,  protein=protein,
sodium=sodium,
                cholesterol=cholesterol,  potassium=potassium,  output=output,  flag=True)

def launch(img_filepath):
    model = load_model('nutrition.h5')
    img = load_img(img_filepath, 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)
    index = ['Apple', 'Banana', 'Orange', 'Pineapple', 'Watermelon']
    values = nutrition(index[classes_x])
    return  [index[classes_x],  values]


def nutrition(x):
    conn = sqlite3.connect('nutri.db')
    cursor = conn.execute(f'''SELECT * FROM NUTRI WHERE FRUIT=="{x}"''')
    for row in cursor:
       rec =
{"FRUIT":row[0],"SERVING_SIZE":row[1],"ENERGY":row[2],"FAT":row[3],"SATURATED
FAT":row[4], "MONO_UNSATURATED_FAT":row[5],
"POLY_UNSATURATED_FAT":row[6],"CARBOHYDRATES":row[7],"SUGAR":row[8],"FIBER":row[9],"PRO
TEIN":row[10],"SODIUM":row[11],"CHOLESTEROL":row[12],"POTASSIUM":row[13],"OUTPUT":row[1
4]}
    return rec
if _name___ == "_main_":
    app.run(debug=False)
```

2. HTML and CSS Pages Screenshots:

Home Page



"Exercise is king, nutrition is Queen, put them together, and you've got a kingdom."

figure 13.2.1

Input Page

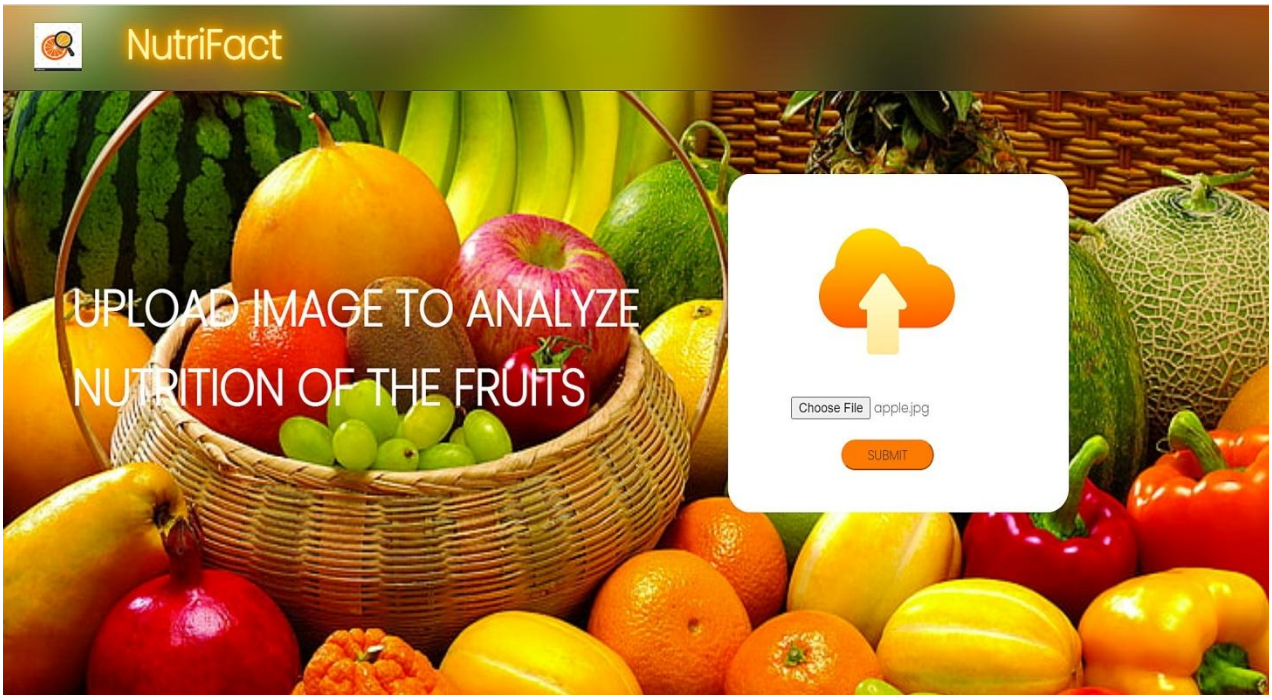
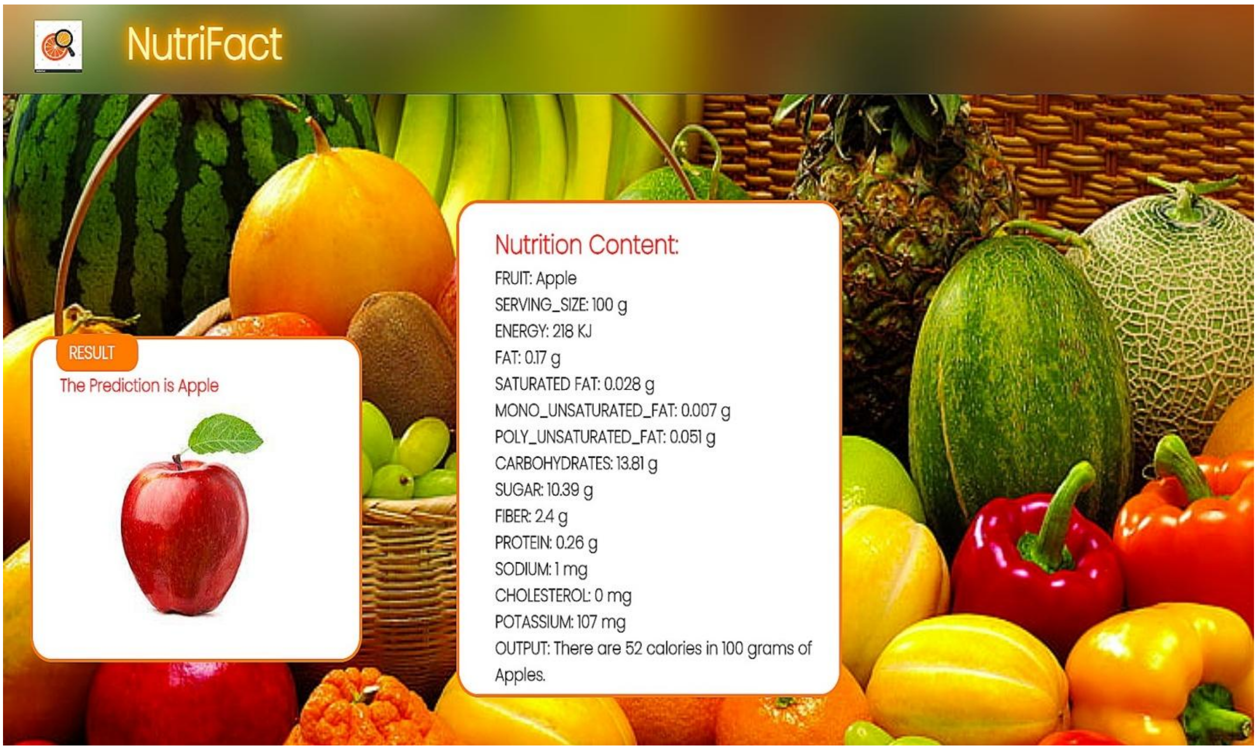


figure 13.2.2

Imageprediction Page



RESULT

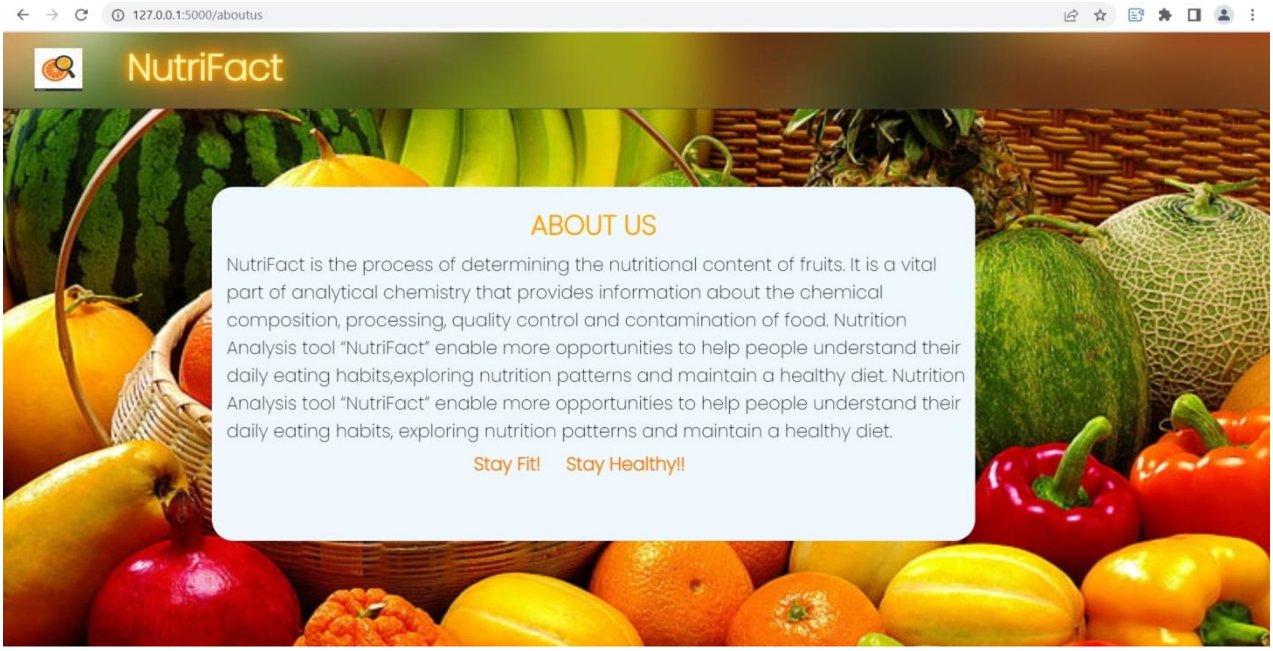
The Prediction is Apple

Nutrition Content:

FRUIT: Apple
SERVING_SIZE: 100 g
ENERGY: 218 KJ
FAT: 0.17 g
SATURATED FAT: 0.028 g
MONO_UNSATURATED_FAT: 0.007 g
POLY_UNSATURATED_FAT: 0.051 g
CARBOHYDRATES: 13.81 g
SUGAR: 10.39 g
FIBER: 2.4 g
PROTEIN: 0.26 g
SODIUM: 1 mg
CHOLESTEROL: 0 mg
POTASSIUM: 107 mg
OUTPUT: There are 52 calories in 100 grams of Apples.

figure 13.2.3

About us Page



ABOUT US

NutriFact is the process of determining the nutritional content of fruits. It is a vital part of analytical chemistry that provides information about the chemical composition, processing, quality control and contamination of food. Nutrition Analysis tool "NutriFact" enable more opportunities to help people understand their daily eating habits,exploring nutrition patterns and maintain a healthy diet. Nutrition Analysis tool "NutriFact" enable more opportunities to help people understand their daily eating habits, exploring nutrition patterns and maintain a healthy diet.

Stay Fit! Stay Healthy!!

figure 13.2.4

GitHub & Project Demo Link

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-7468-1658857709

Project Demo Link