PROJECT DOCUMENTATION

AI-Powered Nutrition Analyzer For Fitness Enthusiasts

TEAM ID : PNT2022TMID26240

Submitted By:-

Pratheebha S - 211519205113 - Team Lead

Thanappriya R L - 211519205172

Monika N S - 211519205098

Divya K - 211519205045

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.

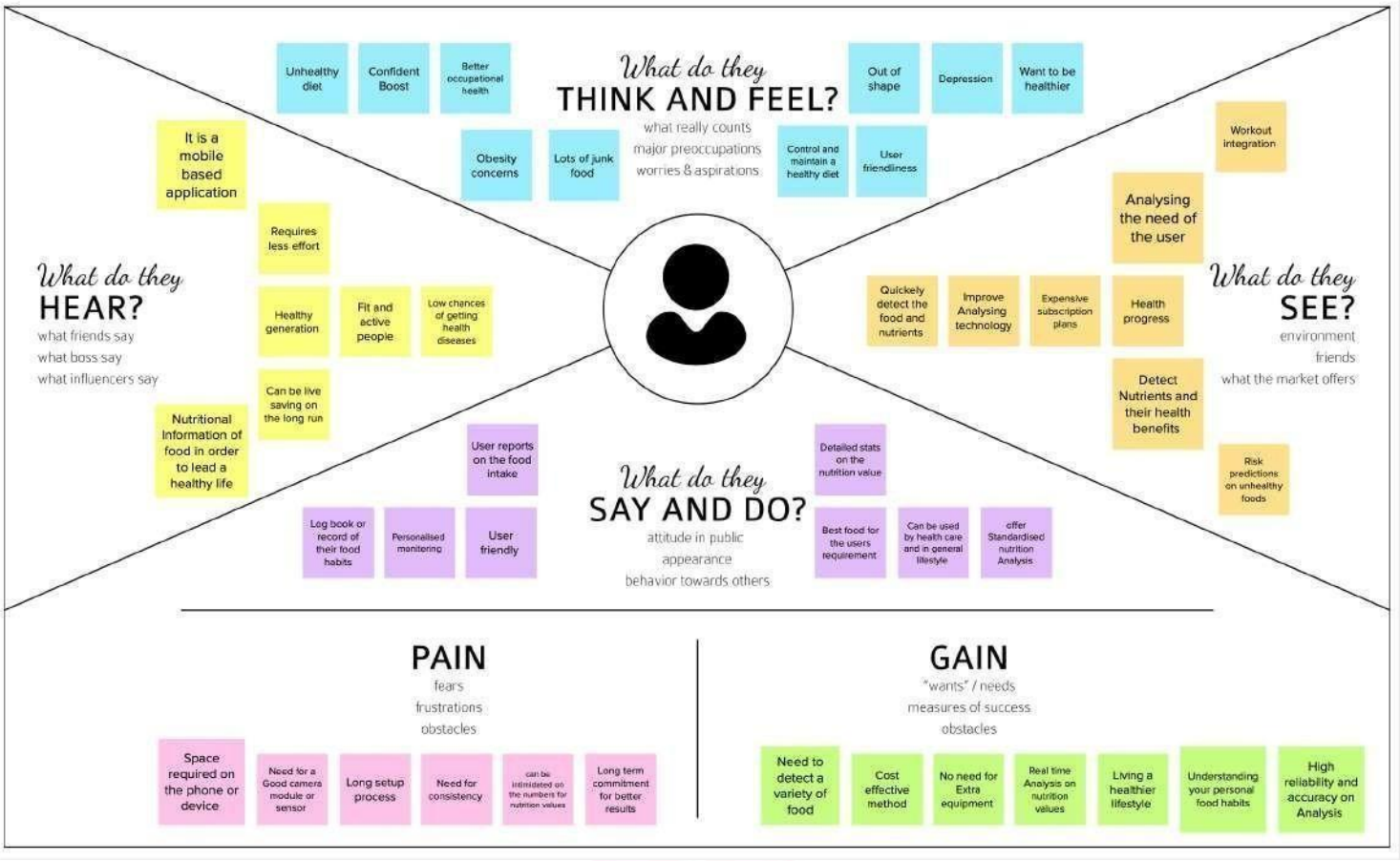
The application relies on Artificial Intelligence to produce custom data related to smart calorie counter powered by AI. Their artificial intelligence learns an individual's tastes, preferences, and body type. All of this is packaged in a comprehensive nutrition and activity tracker.

2.2 Problem Statement Definition

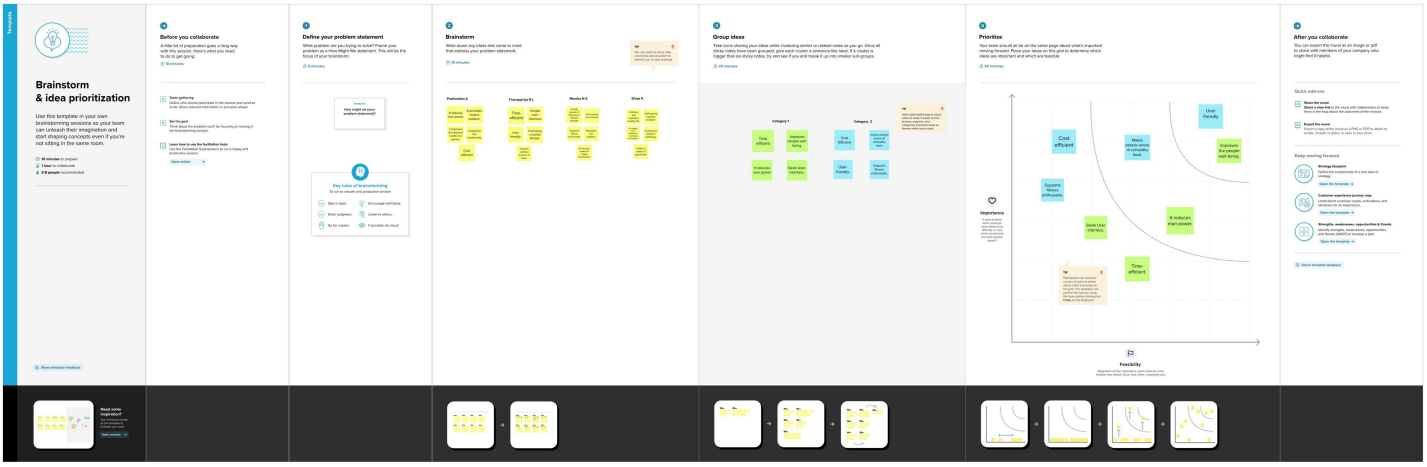
The main aim of the project is to building a model which is used for classifying the fruit depends on the different characteristics like colour, shape, texture etc. Here the user can capture the images of different fruits and then the image will be sent the trained model. The model analyses the image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



3.3 Proposed Solution

S.NO	PARAMETERS	SOLUTIONS
1.	Problem Statement	• Main objective is to detect the nutrition in a fruit from camera captured image. • The identification of nutrition and calories from a image is quite an interesting field. • Since nutrition monitoring plays an important role in leading healthy lifestyle, this product has the potential to become an essential in our day to day life.
2.	Idea / Solution description	• The solution is to develop AI-powered nutrition analyzer application. • By giving the image of the fruit as the input to the application, it will display the nutrition content in it. • By training the model with various inputs, image processing can be improved as well as the accuracy of the result.
3.	Novelty / Uniqueness	• Personalized nutrition for individuals. • Providing science based guidance for healthy living. • Balanced food diet and measured intake. • 24/7 support. • Serving size.

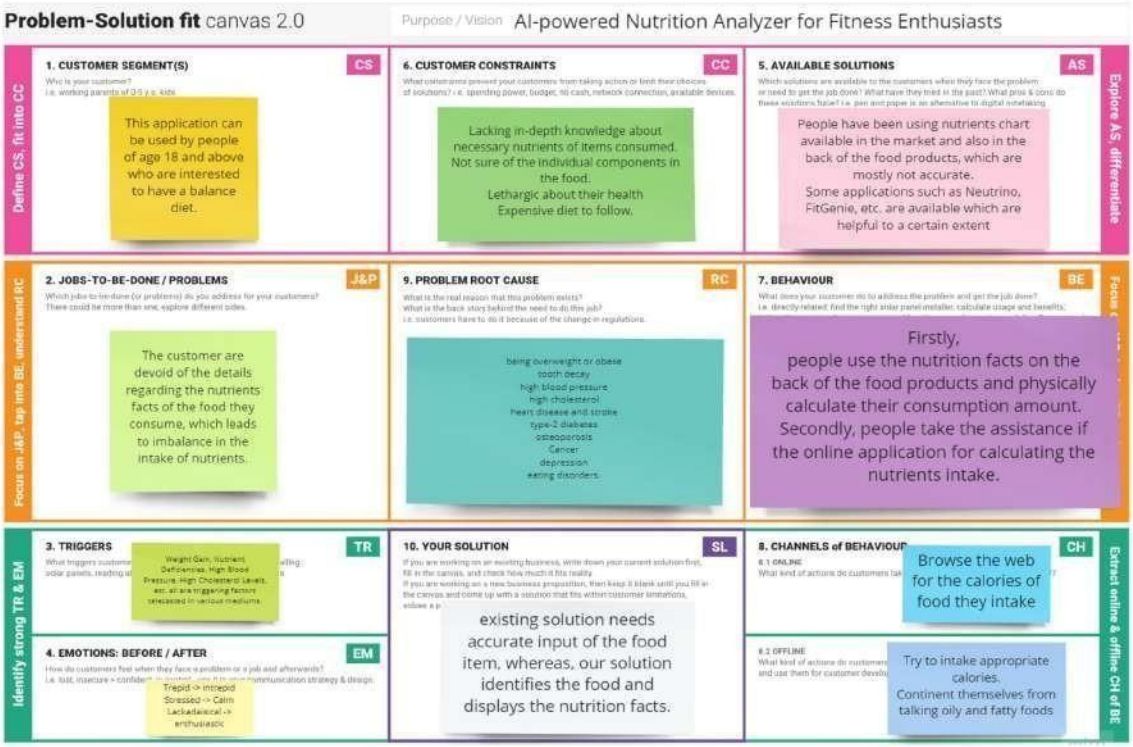
4.	**Social Impact / Customer Satisfaction**	• Economically stable product. • Change one's view towards health and fitness. • Quality of service. • High fiber food. • Accurate amount of nutrition.
5.	**Business Model (Revenue Model)**	• User friendly interface which improves the constant use of the product. • Hence, Economical growth improves. • Product will be delivered in pocket size which results in consuming low memory. • Nutrition and fitness related ads to earn profit
6.	**Scalability of the Solution**	• Offers ingredients substance detail in food • Suggest best health solution and meal plans for different criteria proposed by different individuals. • Virtualization of your long term plan to provide motivation to the customer.

3.4 Problem Solution fit

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

Purpose:

- Solve complex problems in a way that fits the state of your customers.
- Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
- Sharpen your communication and marketing strategy with the right triggers and messaging.
- Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.



4. REQUIREMENT ANALYSIS

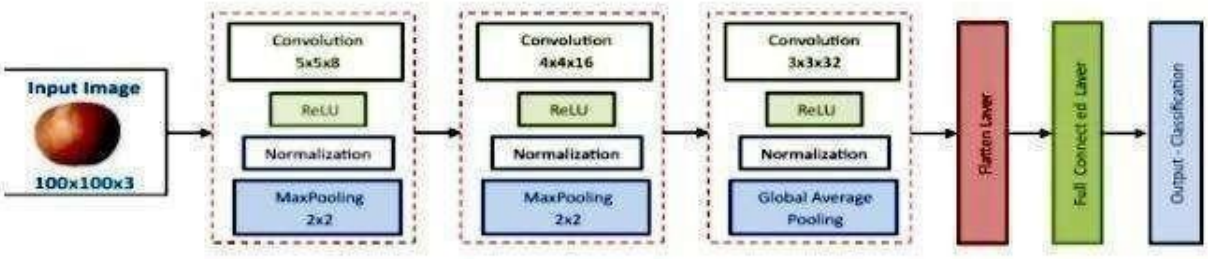
4.1 Functional requirement

- It will generate the diet plan as well as monitor the user's health to classify the category of the disease and to create the diet plan. It will also reduce the cost of consulting the person nutritionist.
- The task of food detection/classification is not easy as it seems. All possible options related to the given Image.
- Image classification, object detection, segmentation, face recognition.
- Classification of crystal structure using a Convolutional neural network
- Nutrition is vital to the growth of the human body. Nutritional analysis guarantees that the meal meets the appropriate vitamin and mineral requirements, and the examination of nutrition in food aids in understanding the fat proportion, carbohydrate dilution, proteins, fiber, sugar, and so on. Another thing to keep in mind is not to exceed our daily calorie requirements
- Computer-Assisted Nutritional Recognize Food Images – In order to solve this issue, a brand-new Convolutional Neural Network (CNN)- based food picture identification system was

created, as described in this study. We utilized our suggested strategy on two sets of actual food picture data.

- Here the user can capture the images of different fruits and then the image will be sent to the trained model. The model analyzes the image and detects the nutrition based on the fruits like (Sugar, Fiber, Protein, Calories, etc.)
- The Ultimate Workout at Home Solution This fitness AI software is designed with personalized training regimens for each individual. It began as “gym only software,” but has now improved its system to satisfy “at home fitness” expectations.
- You take a picture, dial in data such as whether you are eating breakfast or lunch and add a quick text label, and the app estimates the calorie content.
- This software collaborated with IBM’s natural language capability to provide 24-hour assistance and dietary recommendations.

For Example:



- The comparison of the proposed model with the conventional models shows that the results of this model are exceptionally good and promising to use in real-world applications.
- This sort of higher accuracy and precision will work to boost the machine’s general efficiency in fruit recognition more appropriately.
- A generic model for the dietary protein requirement (as with any nutrient) defines the requirement in terms of the needs of the organism,
- i.e. metabolic demands, and the dietary amount which will satisfy those needs, i.e. efficiency of utilization, thus: dietary requirement = metabolic demand/efficiency of utilization.

4.2 Non-functional Requirements:

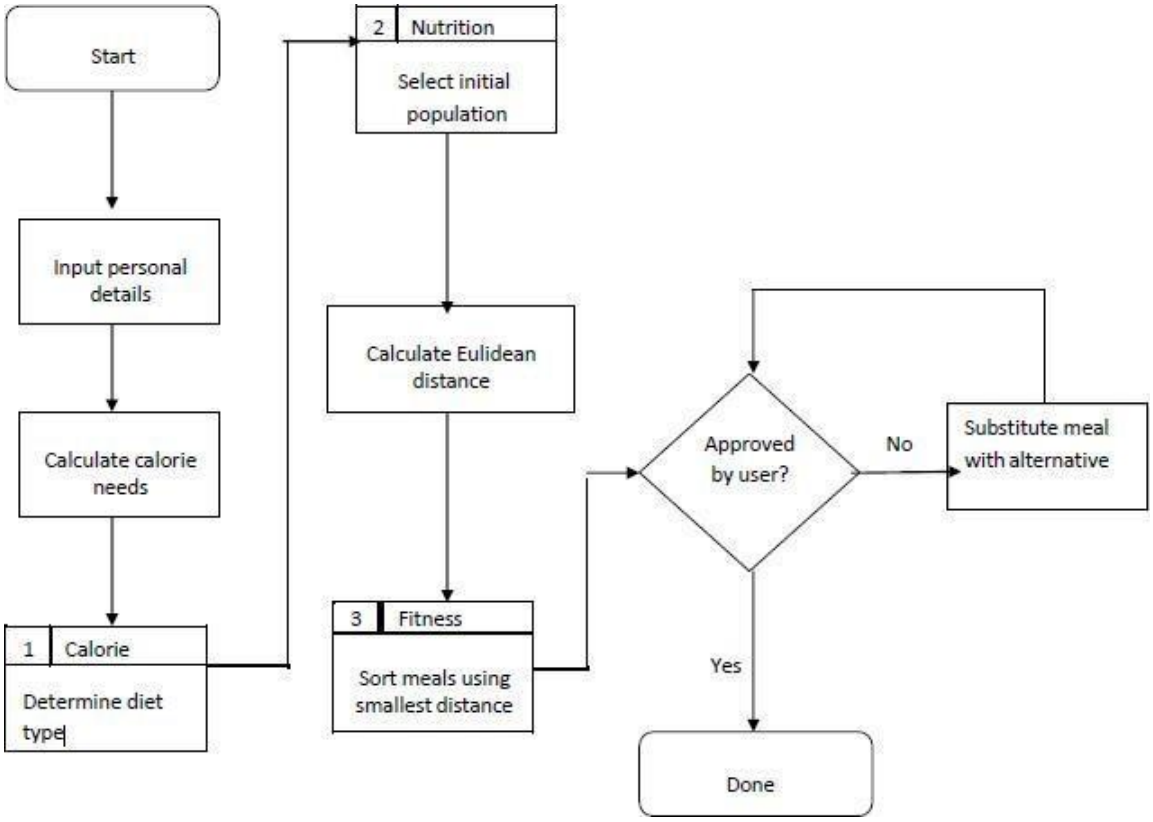
Following are the non-functional requirements of the proposed solution.

- NFR-1 Usability User can know the range before entering their own report and can easily navigate.
- NFR-2 Security The user data is very secured .It is accessed only the user and company
- NFR-3 Reliability The result or the report given is very reliable
- NFR-4 Performance Deals with the front-end load time and fast return of result of the report

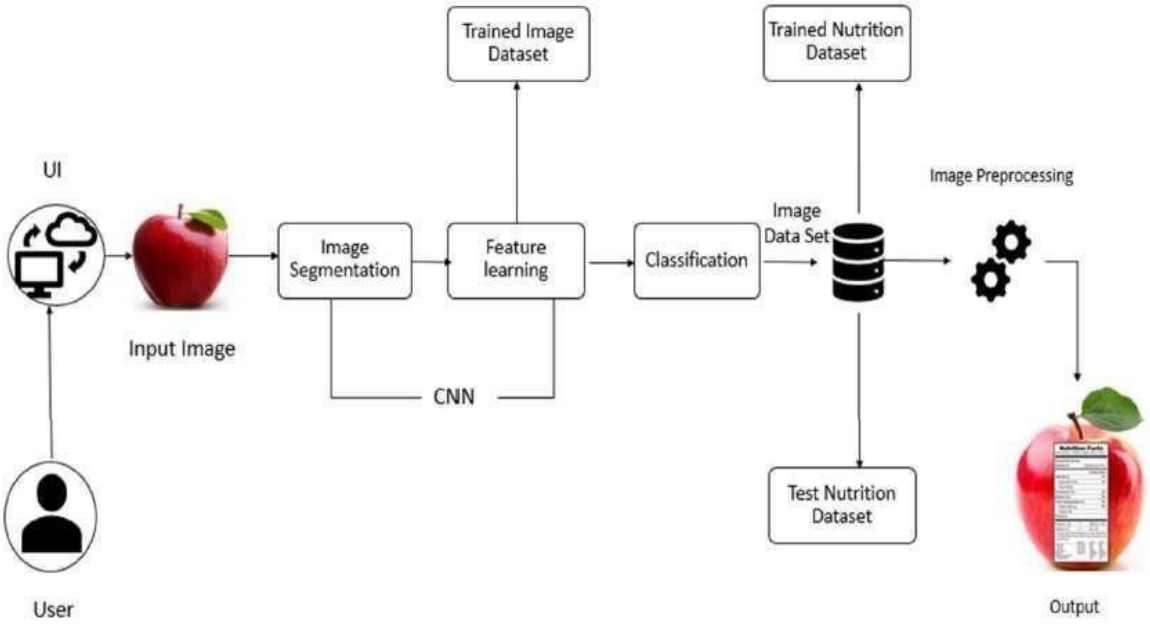
- NFR-5 Availability All the reports and result will be available in user's log in
- NFR-6 Scalability Collecting response from user

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture

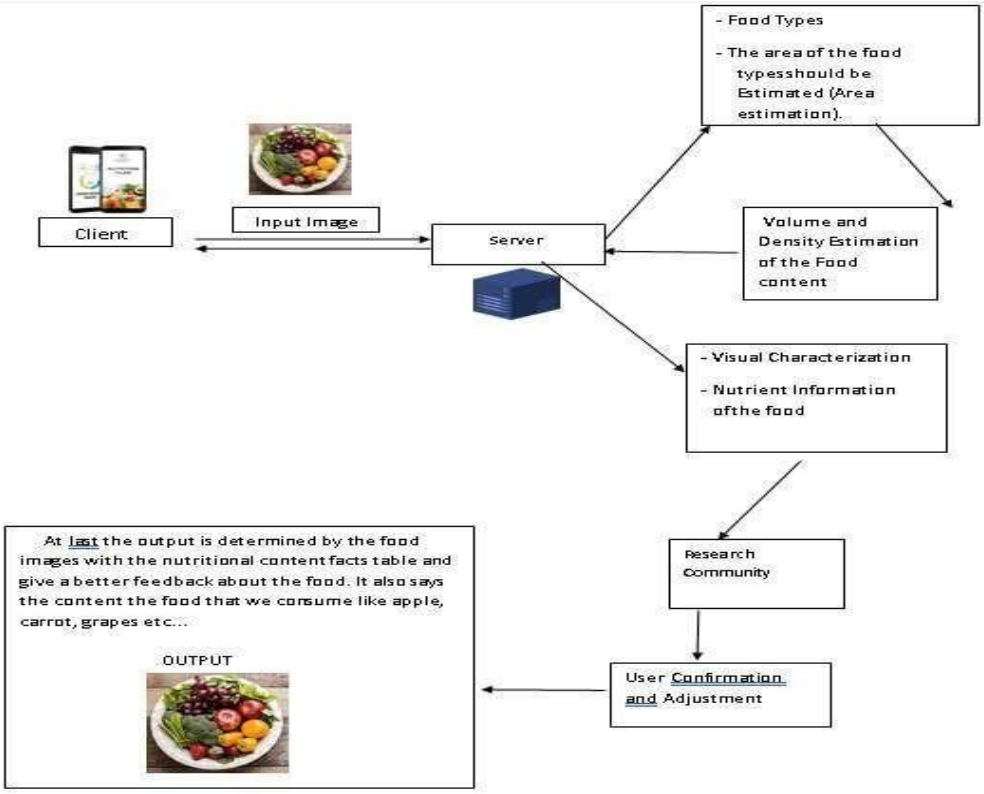


S.No	Component	Description	Technology
1.	App	User interacts with application for the prediction of Nutrition	Python, Java, HTML, SQLite, Android studio
2.	Database	Data Type, Configurations and data will be stored	MySQL, JS
3.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
4.	File Storage	File storage requirements	Cloud -- > drive
5.	Machine Learning Model	Purpose of Machine Learning Model	ANN, CNN, RNN
6.	Notification	Notification will be sent from the server	SendGrid

Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Open-source frameworks used	SendGrid, Python, JQuery
2.	Security Implementations	Request authentication using encryption	Encryptions, SSL certs
3.	Scalable Architecture	The scalability of architecture consists of 3 tiers	Web Server – HTML, CSS ,Javascript Application Server – Python Flask Database Server – IBM Cloud
4.	Availability	Availability is increased by loads balancers in cloud VPS	IBM Cloud hosting
5.	Performance	The application is expected to handle up to 4000 predications per second	IBM Load Balance

5.3 User Stories


- Food Types
- The area of the food typesshould be Estimated (Area estimation).
Client
Input Image
Server
Volume and Density Estimation of the Food content
- Visual Characterization
- Nutrient Information ofthe food
Research Community
User Confirmation and Adjustment
At last the output is determined by the food images with the nutritional content facts table and give a better feedback about the food. It also says the content the food that we consume like apple, carrot, grapes etc...
OUTPUT

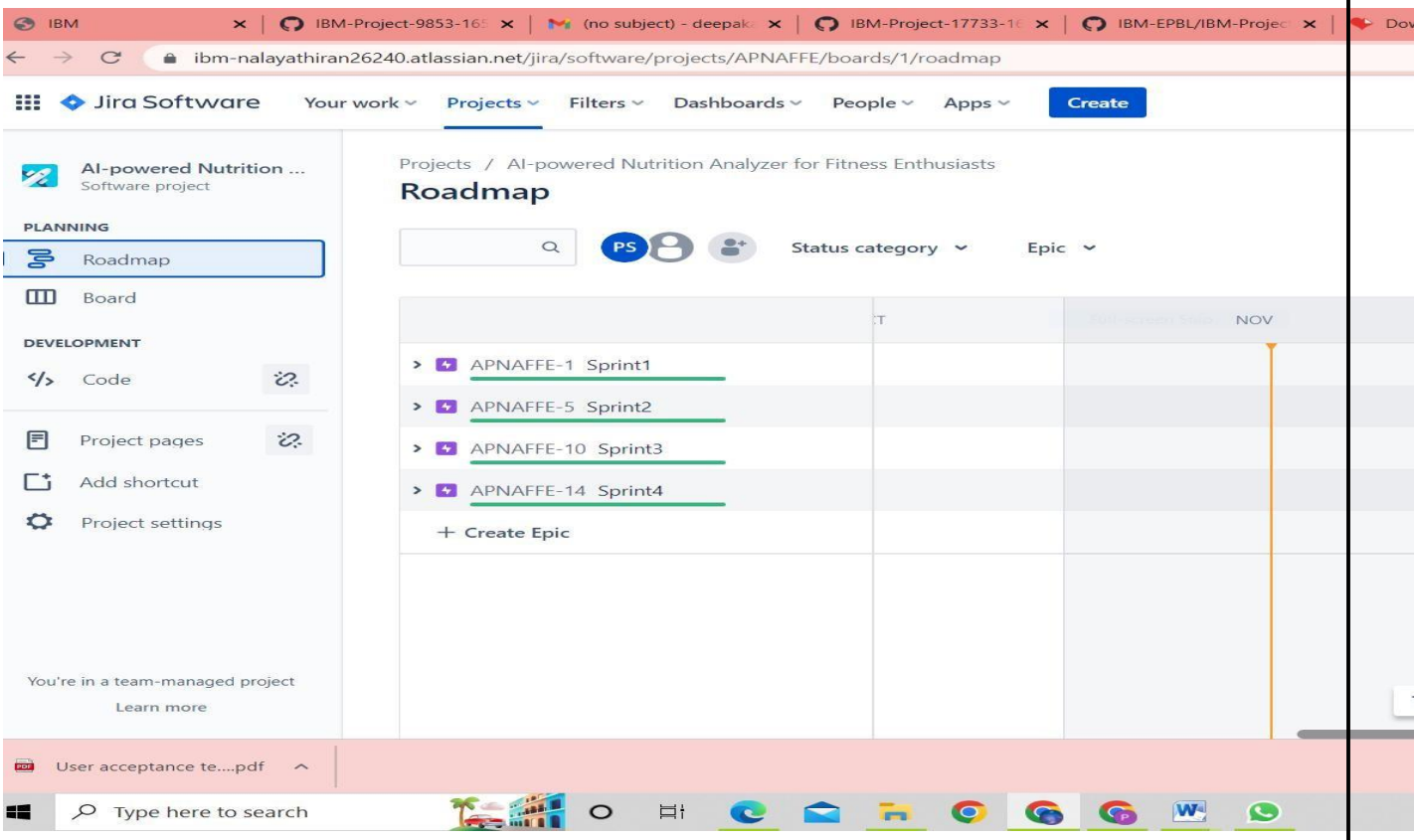
6. PROJECT PLANNING & SCHEDULING

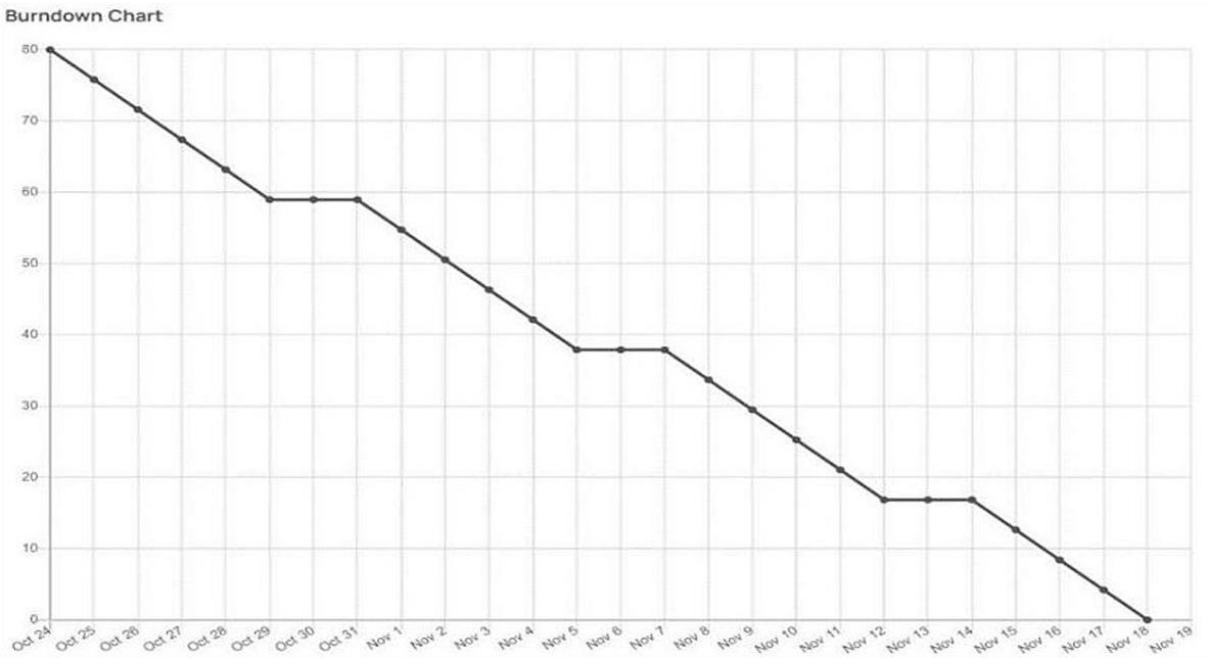
6.1 Sprint Delivery Schedule

Spirit Delivery Plan:

The delivery plan of project deliverables is a strategic element for every Project Manager. The goal of every project is, in fact, to produce a result that serves a specific purpose. With the word "purpose ", we can mean the most disparate goals: a software program, a chair, a building, a translation, etc.... In Project Spirit Delivery Planning is one of the processes of completing the project and Show Casing the Time Line of the Project Planning. This Delivery plan help to understanding the process and Work Flow of the Project working by the Team Mates. Every Single Modules are assigned to the team mates to show case their work and contribution of developing the Project.

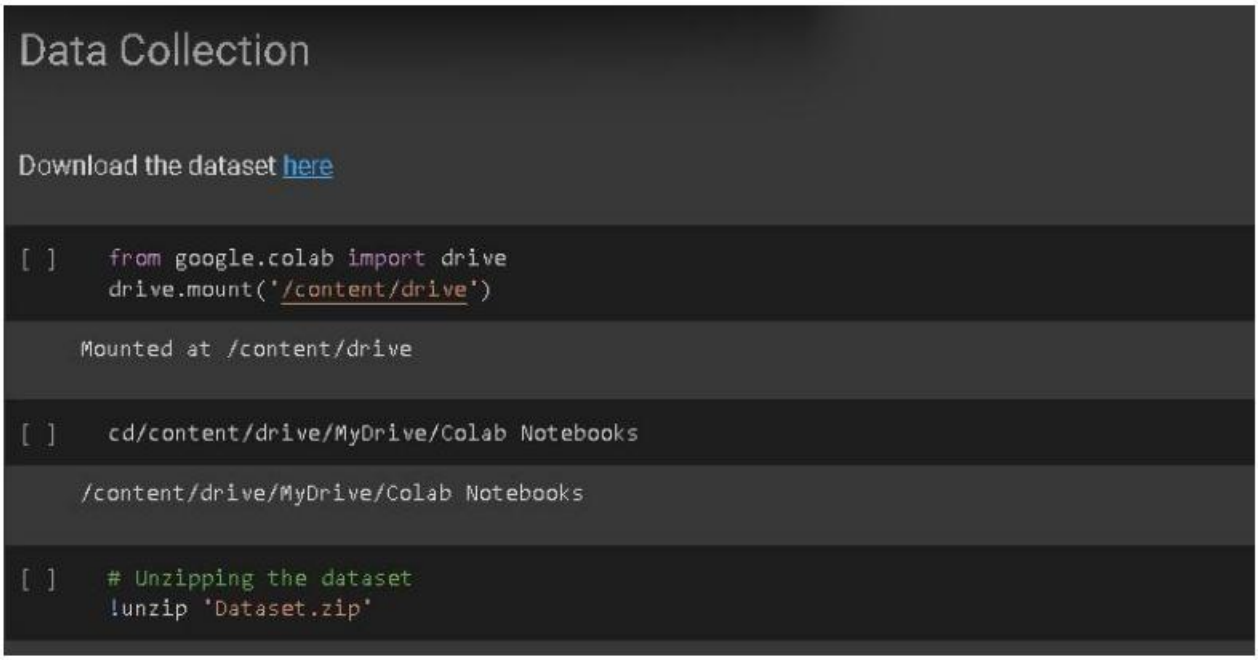
6.2 Jira reports and Burnt chart





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

7.1 Feature 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=20, 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 1/20
496/624 [==============>.............] - ETA: 6:52 - loss: 0.7194 - accuracy: 0.7174
```

7. Saving The Model

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

8. Testing The Model

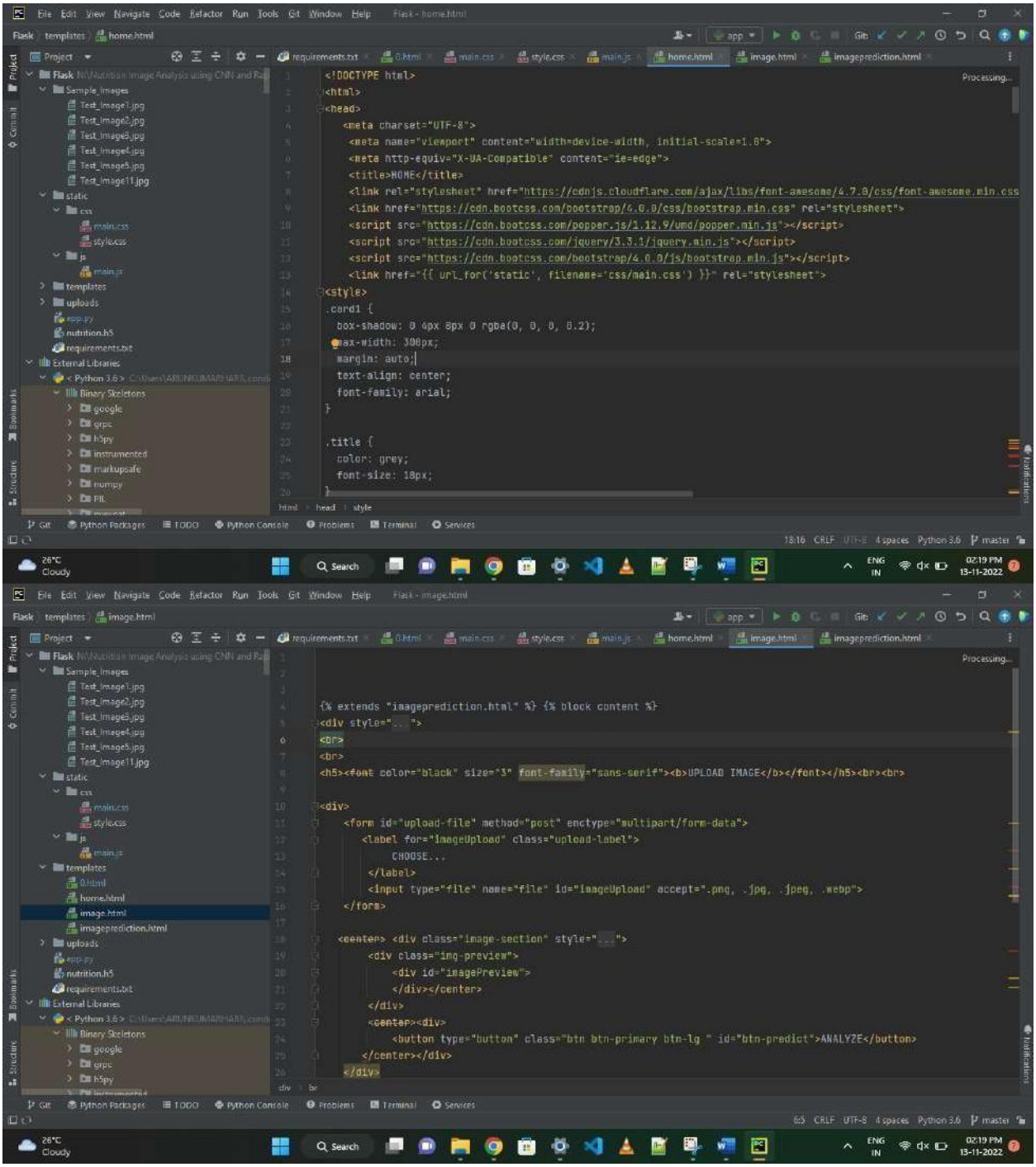
```
[ ]  from tensorflow.keras.models import load_model
     from keras.preprocessing import image
     model = load_model("nutrition.h5")
```

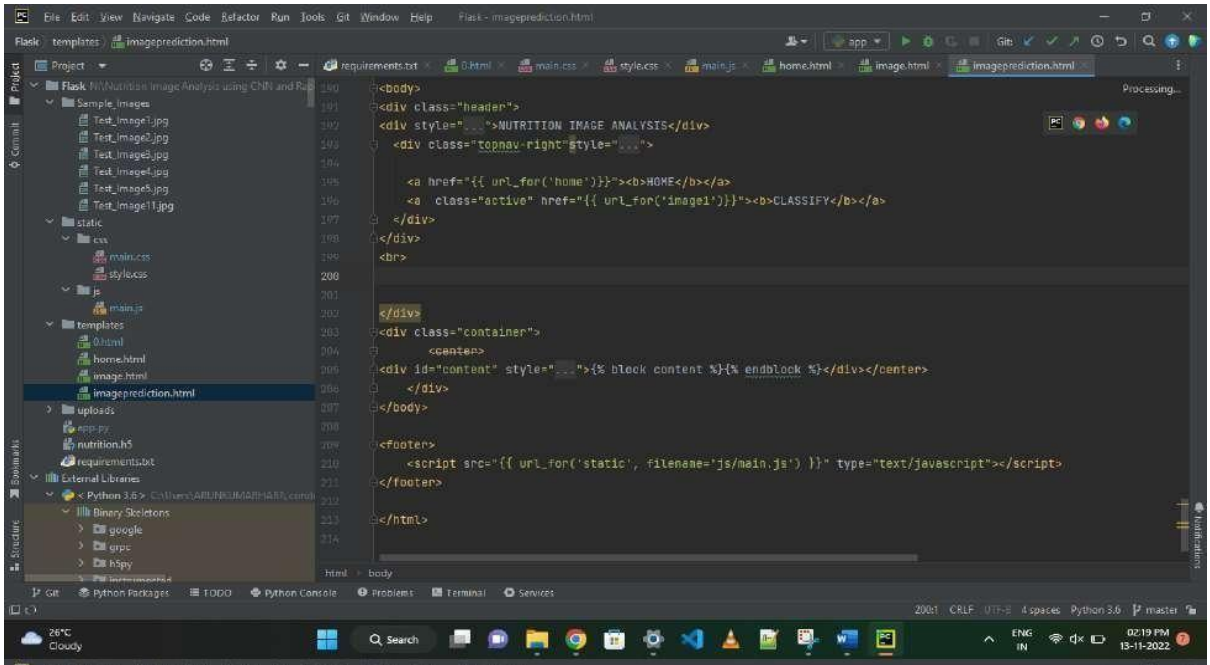
```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model = load_model("nutrition.h5")
img = image.load_img(r'/content/drive/MyDrive/Colab Notebooks/Sample_Images/Test_Image1.jpg',grayscale=False,target_size= (64,64))
x = img_to_array(img)
x = np.expand_dims(x,axis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

7.2 Feature 2

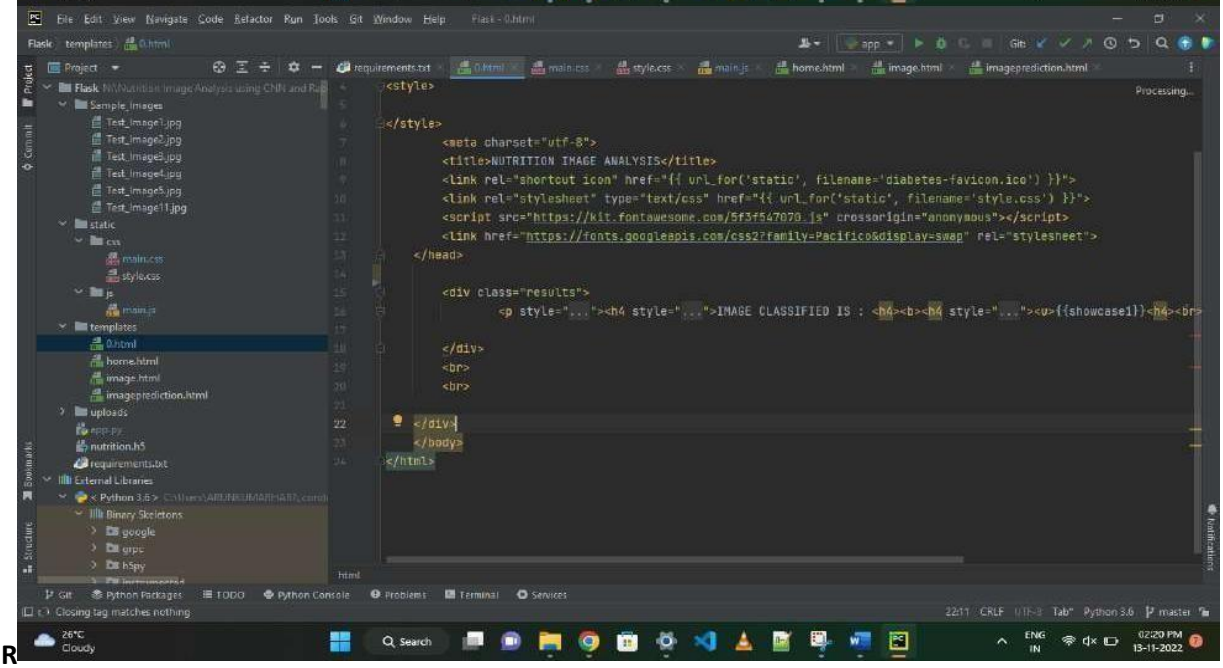



imageprediction.html
<body>

NUTRITION IMAGE ANALYSIS
<div class="topnav-right"style="...">
HOME
CLASSIFY

{% block content %}{% endblock %}

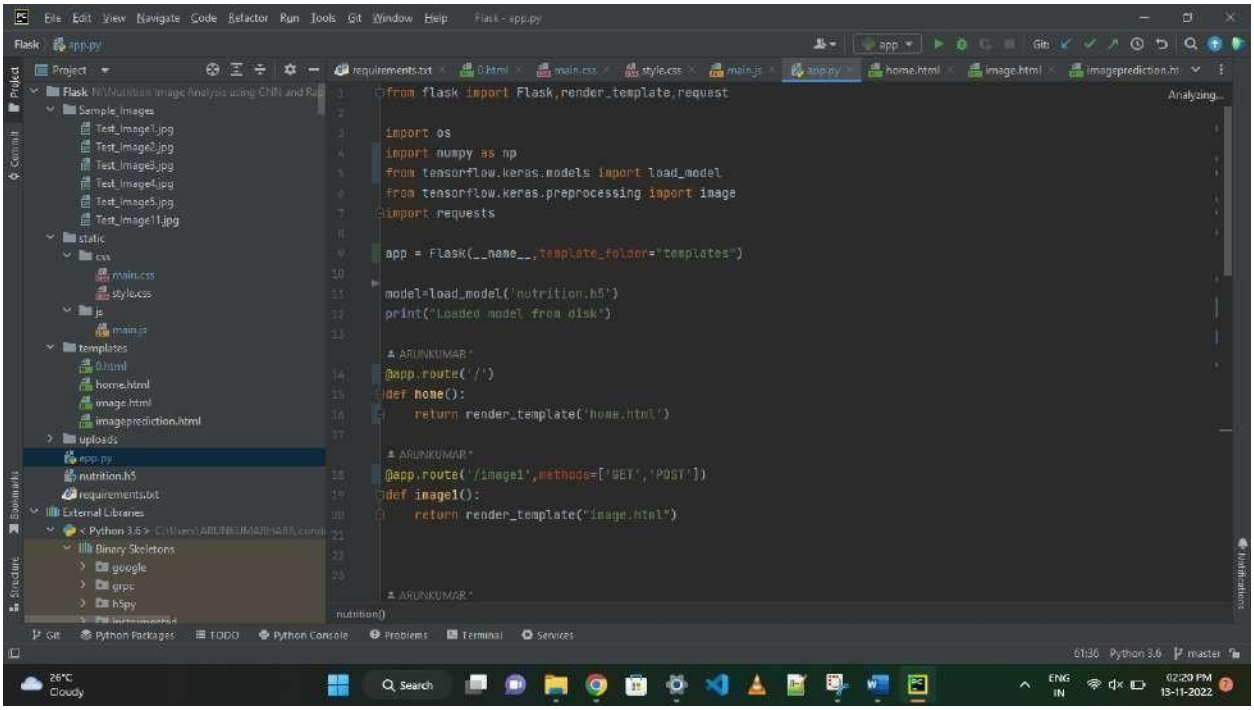
</body>
<footer>
<script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>
</html>


0.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>

<p style="..."><h4 style="...">IMAGE CLASSIFIED IS : <h4><h4 style="..."><u>{{showcase1}}<h4>

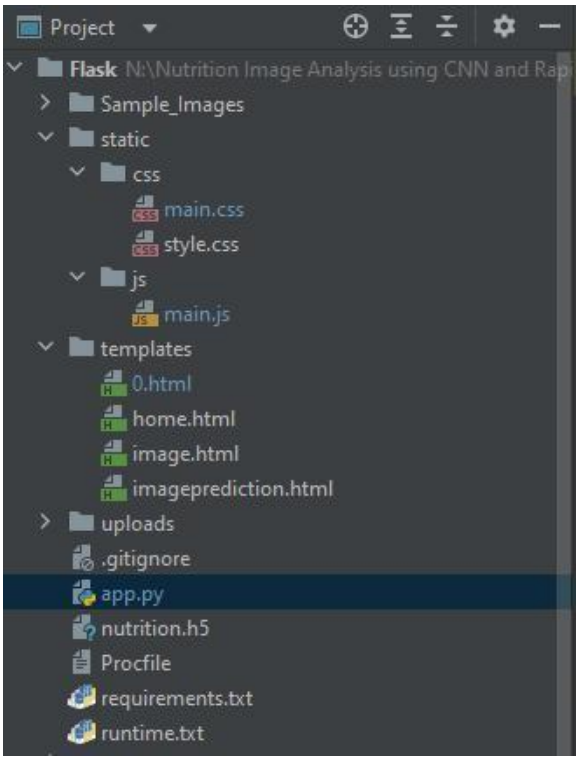
</body>
</html>

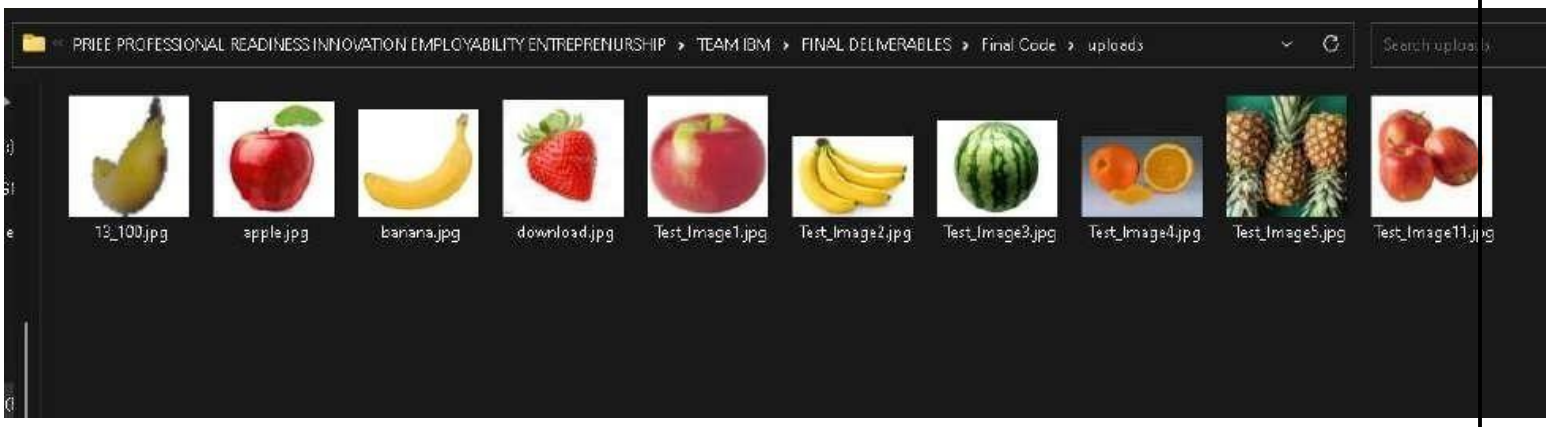
R ES



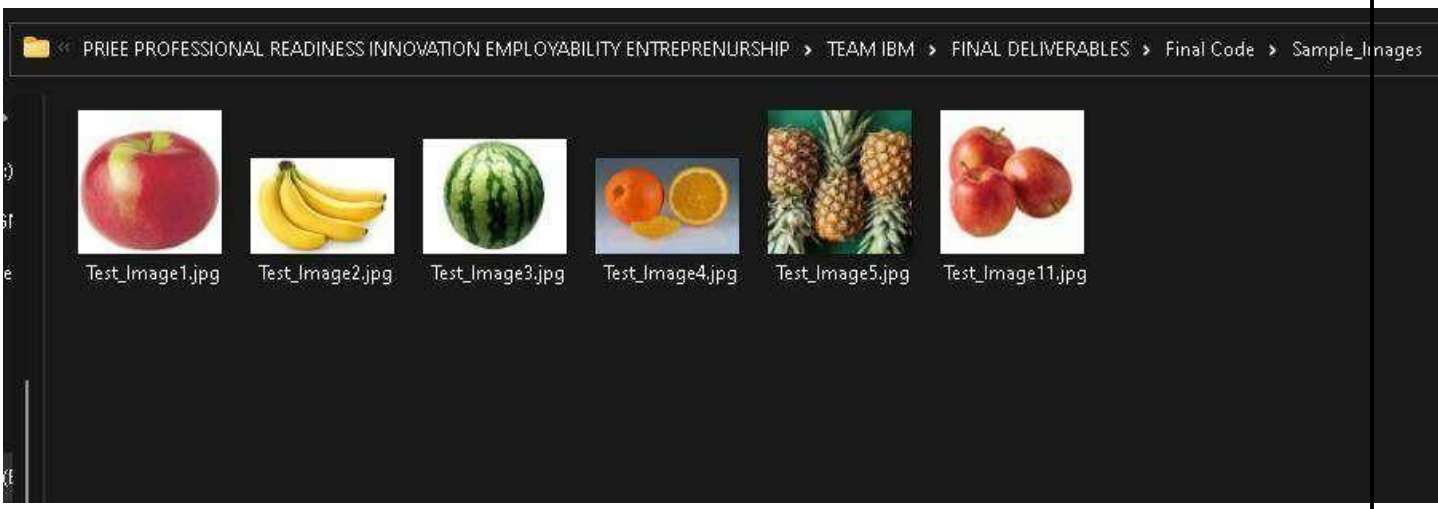
8 TESTING

8.1 TestCases



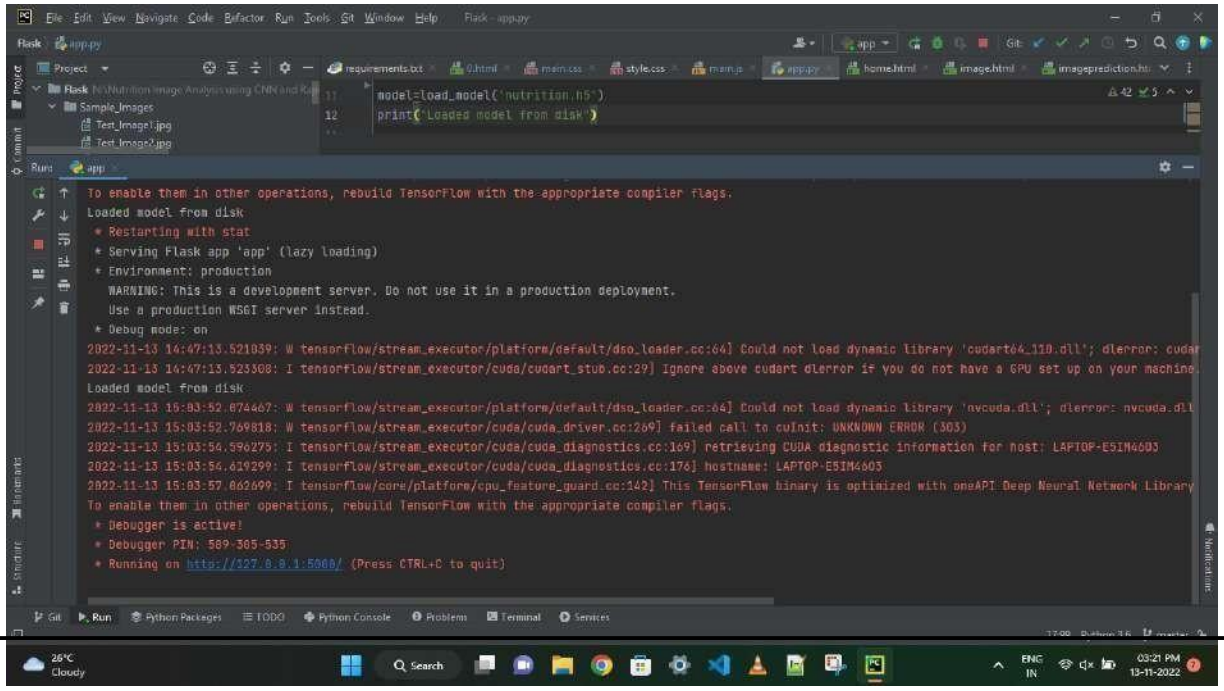


8.2 User AcceptanceTesting



9. RESULTS

9.1 Performance Metrics



10 . ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES:-

- The proposed model here produces very high accuracy of classification.
- Very large datasets can also be trained and tested.
- Images of very high can be resized within the proposed itself

10.2 DISADVANTAGES:-

- For training and testing, the proposed model requires very high computationaltime.
- The neural network architecture used in this project work has high complexity.

11. CONCLUSION

By the end of this project we will

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

12. 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 .

13. APPENDIX

Source code:

```
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")
```

```
@app.route('/predict',methods=['GET', 'POST'])
def launch():
    if request.method=='POST':
        f=request.files['file']
        basepath=os.path.dirname('__file__')
        filepath=os.path.join(basepath,"uploads",f.filename)
        f.save(filepath)

        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)
        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": "tomato"}

    headers = {
        "X-RapidAPI-Key": "f2179b0ee2msh46dd220682815e1p1e6122jsnaea9bb30dd96",
        "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=True)
```

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;
}

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

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

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

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

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

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;
}

.container{
        width:80%;
        margin:auto;
```

```
        overflow:hidden;
}

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

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

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

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

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


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

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

#navbar li{
        display:inline;
}

#navbar a{
        color:#fff;
        text-decoration: none;
        font-size:18px;
```

```
        padding-right:15px;
}

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


#showcase h1{
   width: 100%;
        color:#333;
        font-size:40px;
        text-align: center;
        line-height: 1em;
        padding-top:10px;
}
#showcase h2{
   width: 100%;
        color:#333;
        font-size:30px;
        text-align: center;
        line-height: 1.6em;
        padding-top:10px;
}

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

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


.img-preview {
   width: 10px;
   height: 10px;
   position: relative;
   border: 5px solid #F8F8F8;
   box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
   margin-top: 1em;
   margin-bottom: 1em;
}

.img-preview>div {
```

```
    width: 10%;
    height: 10%;
    background-size: 100px 10px;
    background-repeat: no-repeat;
    background-position: center;
}

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

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

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

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

.myButton span {
  position: relative;
  z-index: 1;
}

.myButton:after {
  content: "";
  position: absolute;
  left: 0;
  top: 0;
  height: 310%;
  width: 150%;
  background: #f2f2f2;
  -webkit-transition: all .5s ease-in-out;
```

```
  transition: all .5s ease-in-out;
  -webkit-transform: translateX(-98%) translateY(-25%) rotate(45deg);
  transform: translateX(-98%) translateY(-25%) rotate(45deg);
}

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

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

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

#main-footer{
        background: #333;
        color:#fff;
        text-align: center;
        padding:1px;
        margin-top:0px;
}

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

        #sidebar{
                width:100%;
                float:none;
        }
}
```

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!');
        },
      });
    });

});
```

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-9853-1659080772

Project Demo Link

https://drive.google.com/file/d/19yMMRlMnSpiPLirDplWa117RsDI5viuK/view?usp=share_linkTE