NUTRITION ASSISTANT APPLICATION

S.NO	REG.NO	NAME	DEPARTMENT	TEAM
1.	211419106031	ARIVANANTHA PANDIAN R	ECE	Team Lead
2.	211419106036	ARUNKUMAR S	ECE	Team Member 1
3.	211419106028	ANTONY KEVIN S	ECE	Team Member 2
4.	211419106039	BALAJI B	ECE	Team Member 3

DONE BY
TEAM ID: PNT2022TMID00941

1. INTRODUCTION

The objective of this study is to identify dietary self-monitoring implementation strategies on a mobile application. Nutritional knowledge is essential for promoting good eating habits since it ensures that necessary nutrient requirements are met to avoid malnutrition.

Wellness and healthy lifestyles have become mainstream. Interest in fitness applications and revenue from them grow as fast as the number of people striving to be fit.

2. PROJECT OVERVIEW

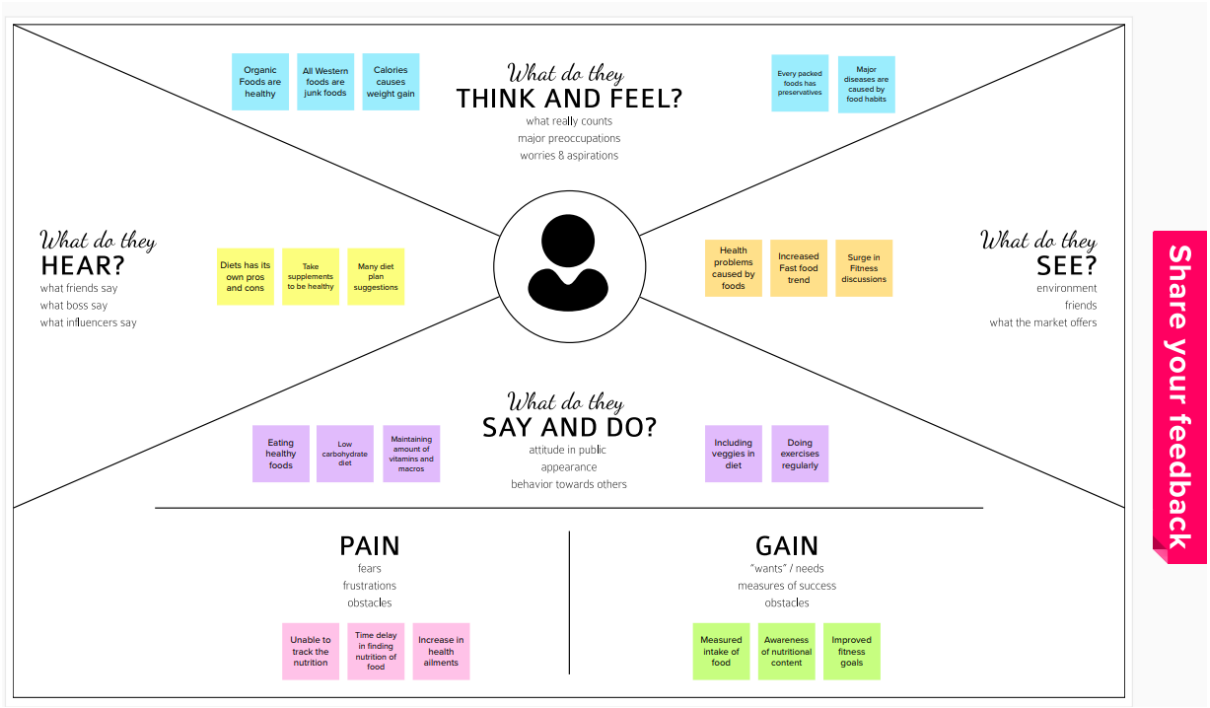
This project aims at building a web App that automatically estimates food attributes such as ingredients and nutritional value by classifying the input image of food. Our method employs **Clarifai's AI-Driven Food Detection Model** for accurate food identification and Food API's to give the nutritional value of the identified food.

3. PURPOSE

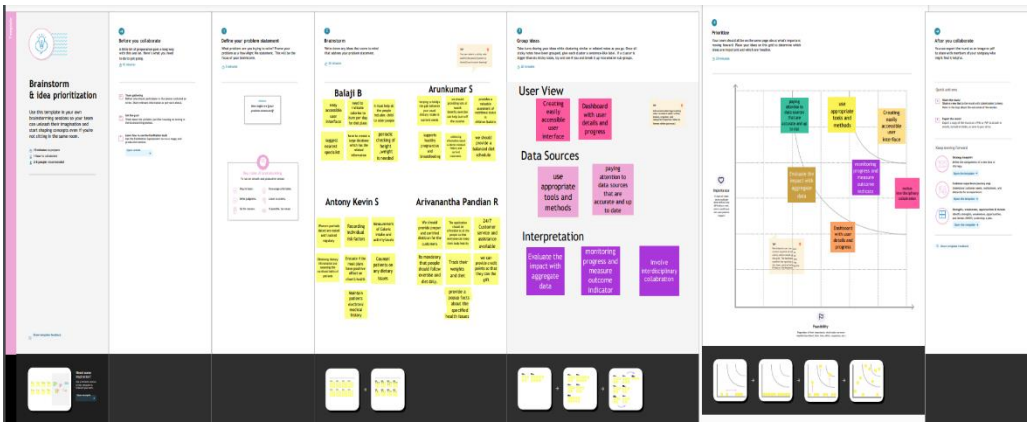
You can automatically calculate the nutritional information for any recipe, analyze recipe costs, visualize ingredient lists, find recipes for what's in your fridge, find recipes based on special diets, nutritional requirements, or favorite ingredients, classify recipes into types and cuisines, convert ingredient amounts, or even compute an entire meal plan.

4. IDEATION PHASE

4.1 EMPATHY MAP



4.2 BRAINSTORMING



4.3 LITERATURE SURVEY

PAPER TITLE	AUTHOR	DESCRIPTION	ADVANTAGES	DISADVANTAGES
1.Use of artificial intelligence in precision nutrition and fitness.	De Moraes Lopes, Maria Helena Baena and Ferreira, Danton Diego and Ferreira, Ana Claudia Barbosa Honorio and da Silva, Giuliano Roberto and Caetano, Aletha Silva and Braz.	Among the available computational tools, artificial intelligence (AI) has gained more and more attention recently, since it is able to learn and model linear and nonlinear relationships between variables by constructing an input-output mapping such that hidden and extremely useful information for decision-making is revealed and interpreted.	A large amount of data is collected by these technologies	AI is not yet widely used in the areas of nutrition and fitness
2. Predicting calorific value for mixed food using image processing.	Kohila, R and Meenakumari, R.	The objective of this paper is to predict and to fix diet control for various diseases by measuring the calorific value to help the patients and nutritionists. The image captured through a mobile phone/tablet camera will provide information concerning the	It increases the security for the information provided by the user. It reduces cost.	They have a limited variations for the prediction of calories in the food. The application lack the support from the user.

		calorie rate of the food.		
3. Mobile cloud based system recognizing nutrition and freshness of food image	Kumbhar, Diptee and Patil, Sarita	Mobile cloud computing (MCC) has been introduced to be a potential paradigm for mobile health services to overcome the interoperability issues over distinctive information formats. In this,we propose a mobile cloud-based food calorie measurement framework.	They have a multiple platform support for the end users. It is cost-efficient .	They have a connectivity and performance issues.
4. Enhancing Cloud and healthy Food Nutrition Information Systems Practice.	Paul, PK and Aithal, PS and Bhuimali, A.	Among the common mass food information systems are not yet popularized as a domain and thus there are huge potentialities to work on this.	Regarding manpower development there are a lot of things are pending and possible to work with. Hence cloud will do an attention on skill and manpower development for sophisticated development of food information systems.	

5. An image analysis system for dietary assessment and evaluation	Fengqing , Marc Bosch, Carol J. Boushey	Measuring accurate dietary intake is considered to be an open research problem in the nutrition and health fields. This approach includes the use of image analysis tools for identification and quantification of food that is consumed at a meal. The mobile device provides a unique vehicle for collecting dietary information that reduces the burden on respondents that are obtained using more classical approaches for dietary assessment. This approach uses image analysis that includes the segmentation of food items, features used to identify foods, a method for automatic portion estimation, and our overall system architecture for collecting the food intake information.	Useful for replacing the traditional food record methods currently used.	Measuring accurate dietary intake is an open research problem in the nutrition and health fields.

4.4 PROBLEM STATEMENTS

Problem Statement(PS)	I am	I'm tryingto	but	Because	Which makes me feel
PS-I	Nutritional assistant	To determine nutritional needs	Issues in eating proper healthy food	People donot get proper guidelines to intakeof food	Being overweight
PS-II	Nutritional assistant	To assess risk factor	Problem in getting proper nutrition	People do not have proper education about nutrition's important	Anxiety and depression

5. REQUIREMNT ANALYSIS

5.1 FUNCTIONAL REQUIRMENTS

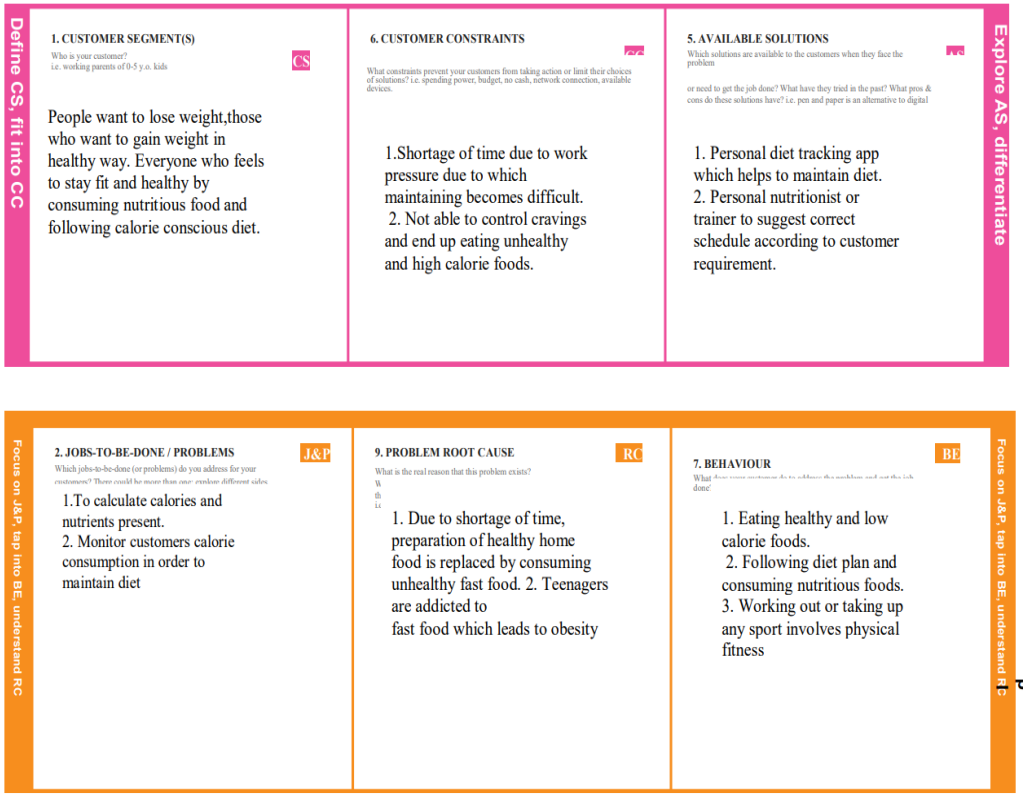
- User Registration
- User Confirmation
- Update Profile
- User Authentication
- Report

5.2 NON FUNCTIONAL REQUIRMENTS

- Usability
- Security
- Reliability
- Performance
- Availability
- Scalability

6. PROJECT DESIGNS

6.1 PROBLEM SOLUTION FIT



6.2 PROPOSED SOLUTION

The project aims at developing an application that helps people to lead a healthy lifestyle by providing information about the ingredients and their nutritional content in the food they are consuming. By this people can avoid various health-related issues like obesity, heart attack, diabetics etc. Monitoring and tracking of goal and diet plans will be provided for the user based on the data collected from them.

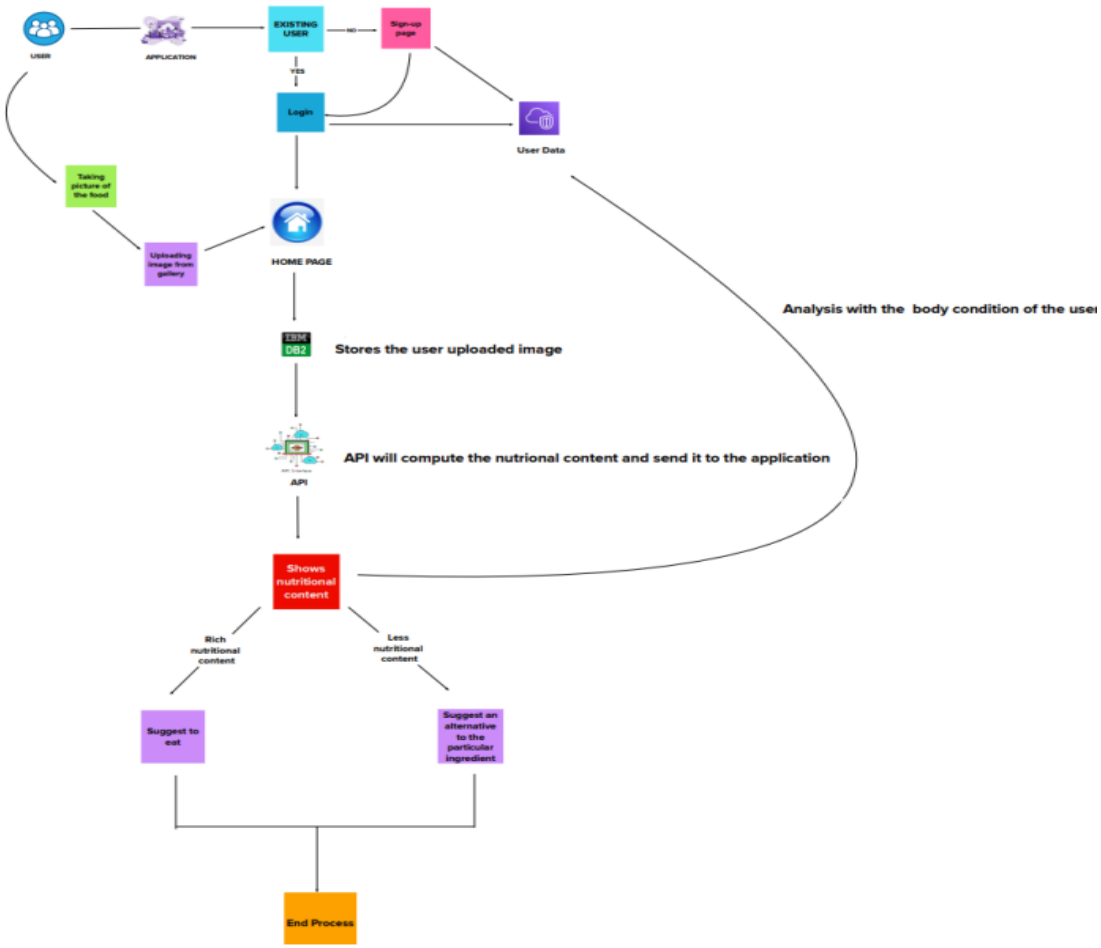
S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• It is easy to fall into a trap of eating unhealthy foods which is heavy in calories. Once the nutritional value is replaced by foods high in sugar, bad fats and salt it leads to various health issues so users need to control their daily calorie intake to lead a healthy lifestyle.
2.	Idea / Solution description	• The solution is a responsive Web application that can be used in any PCdevices. • The website provides a user-friendly interface and accepts multiple samplespredicting them simultaneously. • Our method uses Clarifai's AI- driven food recognition model to accurately identify food suggestions. • A detailed report of the concerned person's health will be generated.
3.	Novelty / Uniqueness	• Keep a food journal. • Providing individual diet charts for users based on their BMI and medical condition if any. • Provides recipes according to their diet. • Providing a user-friendly environment.
4.	Social Impact / Customer Satisfaction	• Getting feedback from the users for enhancement and giving notification on their diet plans and goal tracking. • Nutrition focused food banking & targetedin-depth reporting reviews that

		paid subscriptions the best.
5.	Business Model (Revenue Model)	• Advertising membership option for users to get more benefits like diet- plans or consultation from experts and In-app advertisements. • Revenue is generated on a subscription basis, with big data processing and targetedin-depth reporting reviews that paid subscriptions the best.
6.	Scalability of the Solution	• Providing regular updates • Efficient goal tracking assistance • The additional features such that sleep tracking, mensuration tracking can be done.

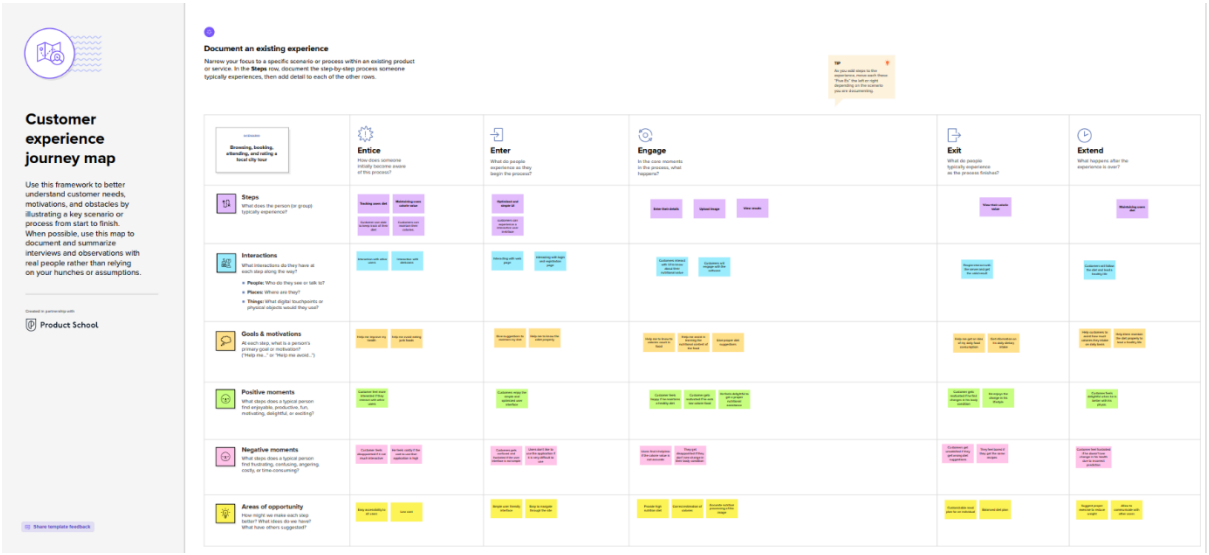
6.3 SOLUTION ARCHITECTURE

Due to the ignorance of healthy food habits, obesity rates are increasing at an alarming speed, and this is reflective of the risks to people's health. People need to control their daily calorie intake by eating healthier foods, which is the most basic method to avoid obesity. However, although food packaging comes with nutrition (and calorie) labels, it's still not very convenient for people to refer to App-based nutrient dashboard systems which can analyze real-time images of a meal and analyze it for nutritional content which can be very handy and improves the dietary habits, and therefore, helps in maintaining a healthy lifestyle.

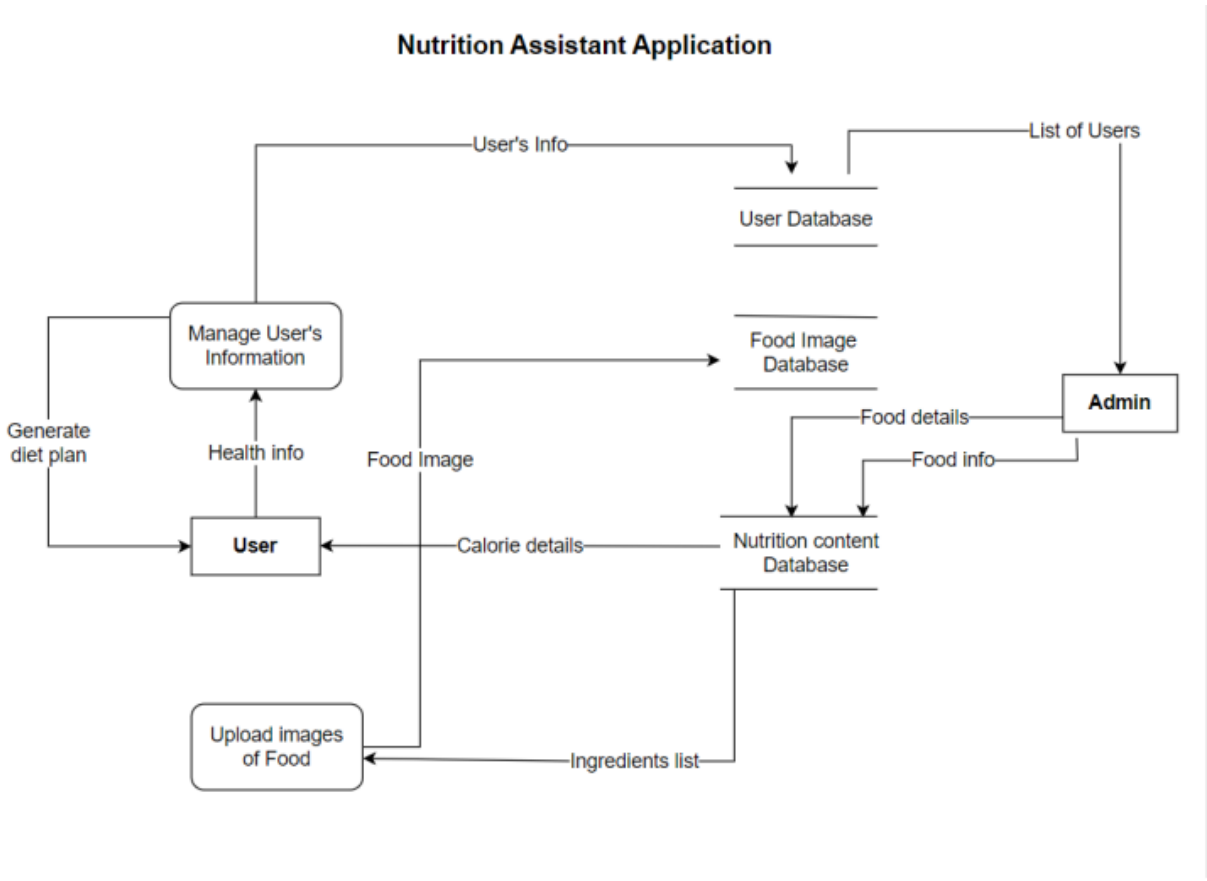
SOLUTION ARCHITECTURE DIAGRAM



6.4 CUSTOMER JOURNEY



6.5 DATA FLOW DIAGRAM



6.6 SOLUTION REQUIRMENTS

Project description:

This project is aimed at developing a desktop-based application named Nutrition Assistant Application for estimates food attributes such as ingredients and nutritional value by classifying the input images of food. The Nutrition Assistant Application refers to the system and processes to help the user to analyze the intake of food with the involvement of a Technology system. This system can be used to store the details of the user's health, calculating the BMI, Classifying the food image to know the nutritional value, update the status of their health condition based on the information provided, and generate health reports weekly or monthly based. This project is categorizing individual health condition of the user. The Nutrition Assistant Application is important to control their daily calorie intake by eating healthier foods, which is the most basic method to avoid obesity. Without proper diet control, and this is reflective of the risks to people's health. A good Nutrition Assistant Application will alert the users when it is time to avoid. This project aims at building a web App that automatically estimates food attributes such as ingredients and nutritional value by classifying the input image of food.

Scope:

Maintains good health: The application can help in guiding them on how to remain healthy and how to take good nutrition. The application will help them without personally going to the doctor. Promote better nutrition in the community by educating about better diet and nutrition.

Functional limitation: The user to be specific can't access the web or admin module, whereas the administrator has all the rights to modify and manage the contents such as news, tips, etc

Improve Usability: In the part of user's just the internet connection is enough in order to access the news, updates and other contents provided by the admin regarding their health condition.

Health conscious: This will provide convenience to persons/users who wants to learn about nutrition and other related health topics by just using the Nutrition Assistant Application

Purpose:

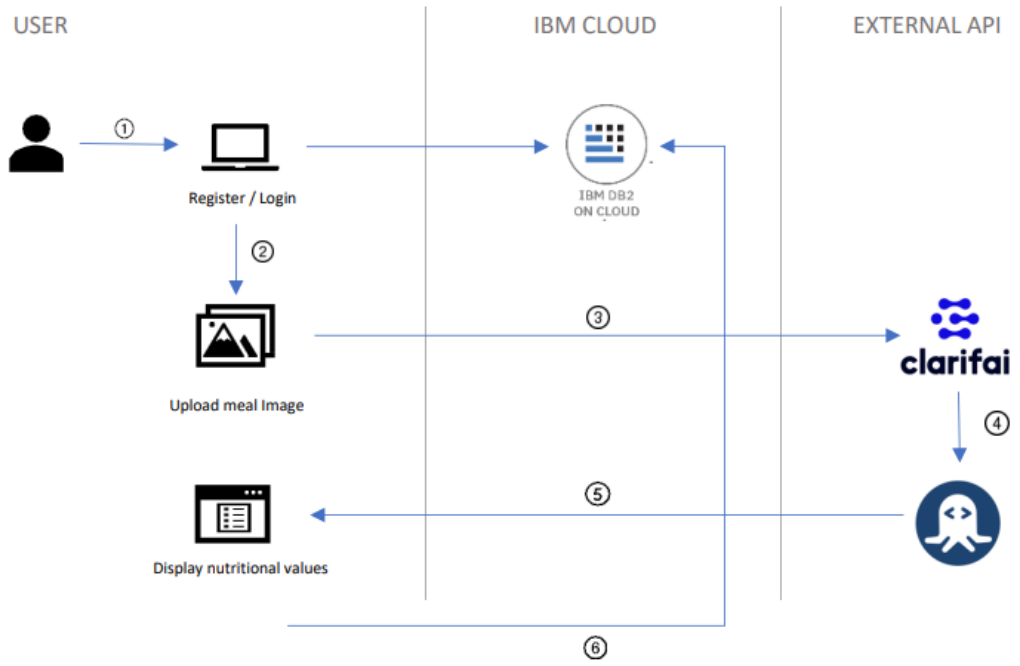
The users continue to demand to know the nutritional value that is in their food. The users learn about the effect of different foods on human health. Evidently, the ultimate aim of this application is to provide the ways in which one can lead a healthy life by maintaining his/her diet. The user can access the nutritional information by taking a photo of the food, uploading a photo from the gallery, or by entering manually. Nutrition is more than just obtaining nutrients and calories from food. It's more than just eating the healthy stuff. It's more than just following the most recent fad diet. Nutrition, the food we eat and the way we eat it, is an integral part of life. Nutrition is an experience. It evokes memories, helps us celebrate good times, and is there for us in times of grief. I believe the purpose of nutrition is to nourish the body and soul. The Nutrition Assistant Application helps the users to eat nutritional rich food which yield to lead a healthy life.

IDENTIFIER	REQUIREMENTS
1. Add health information	This application will allow to add health related information of the user.
2. Delete health information	This application will allow to delete the unwanted details about their health.
3. Categories of nutritional food	The categories of food.
4. View of Dashboard	Application will allow user to view the dashboard containing nutrition details.
5. Mail Notification	This application will allow to send mail notification to user when there are any issues regarding their health
6. Tracking System	The health can be tracked with this application.
7. Graph analysis	This application will demonstrate health condition by means of nutritional content
8. Identifying the high calorie food	The high calorie ingredients will be shown via this application.
9. Identifying the low calorie food	The high calorie ingredients will be shown via this application.
10. Passcode	This application has the option to set a passcode to keep their medical reports safe.
12. Add multiple accounts	This application has the option of creating multiple accounts for the users.
13. Selection of health report duration	This application has the ability to select the duration for displaying the health report as weekly or monthly.

14. Update account	This application will allow the user to update their profile.
15. Add account	This application will allow the user to add their profile.
16. Delete account	This application will allow the user to delete their profile.
17. PDF report	This application will generate the pdf report of medical analysis.
18. Pupation of nutritional trends	This application will allow constant review of nutritional trends and pupation.

6.7 TECHNOLOGY ARCHITECTURE

Technical Architecture:



Guidelines:

1. To use the app the user must register / login.
2. After successful registration/login, the user can upload the meal image.
3. Using Clarifai AI- Driven API the name of the meal will be identified.
4. The identified name will be sent to Nutrition API using Flask.
5. Using Nutrition API, the nutritional value of the meal will be obtained and displayed in the UI using Flask.
6. The diet history will be added to the database to track their daily calorie intake.

Table 1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User interacts with application Web UI	HTML, CSS, JavaScript
2.	Application Logic-1	Connection with Database and external API's	Python Flask
3.	Application Logic-2	Integration of chatbot with application	IBM Watson Assistant
4.	Database	Data Type, Configurations etc.	MySQL
5.	Cloud Database	Database Service on Cloud – used to store user details for registration and login, and track diet history	IBM DB2
6.	External API-1	This API is used to find the name of the food, for which the image has been uploaded	Clarifai AI-Driven API
7.	External API-2	This API is used to find the recipe and Nutritional value present inside the food	Nutrition API (Rapid API)
8.	Infrastructure	Application Deployment to provide good performance and scalability	Kubernetes

Table 2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask is used for connecting database and external API's.	Python flask
2.	Security Implementations	Security is provided for accessing the database.	SSH
3.	Scalable Architecture	Presentation tier: User Interface to login and upload meal image Application tier: Nutrition API, Clarifai API Database tier: IBM cloud DB2	HTML, CSS, JavaScript, Flask, Kubernetes, IBM DB2
4.	Availability	Clustering improves availability. This can be achieved with the help of Kubernetes cluster.	Kubernetes
5.	Performance	By using cache and adding master nodes we can improve performance of the application	Kubernetes

6.8 USER STORIES

- ⇨ As a user, I can register for the application by entering my email, password, and Confirming my password
- ⇨ As a user, I will receive confirmation email once I have registered for the application
- ⇨ As a user, I can log into the application by entering email & password
- ⇨ As a user, I can fill the details.
- ⇨ As a user, I will search the food items.
- ⇨ As a user, I can scan the food an get the nutrition details and recipe for related scanned food.

7. PROJECT PLANNING AND SECHDULING

7.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Arivanantha Pandian R Arunkumar S Antony Kevin S Balaji B
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Arivanantha Pandian R Arunkumar S Antony Kevin S Balaji B
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	1	High	Arivanantha Pandian R Arunkumar S Antony Kevin S Balaji B
Sprint-2	User details	USN-4	As a user , I can fill the Details.	2	High	Arivanantha Pandian R Arunkumar S Antony Kevin S Balaji B
Sprint-3	Push notification	USN-5	As a user, I will search the food items.	2	Medium	Arivanantha Pandian R Arunkumar S Antony Kevin S Balaji B
Sprint-4	Shown the nutrition details and Recipe for scanned food	USN-6	As a user, I can scan the food an get the nutrition details and recipe for related scanned food	1	High	Arivanantha Pandian R Arunkumar S Antony Kevin S Balaji B

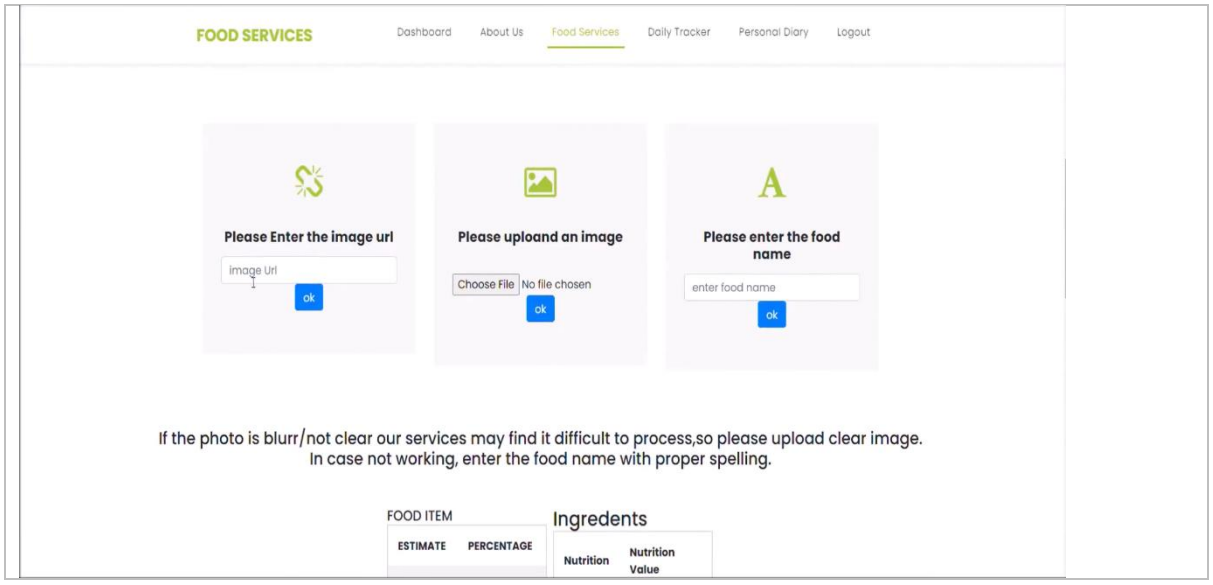
7.2 Sprint Delivery Schedule

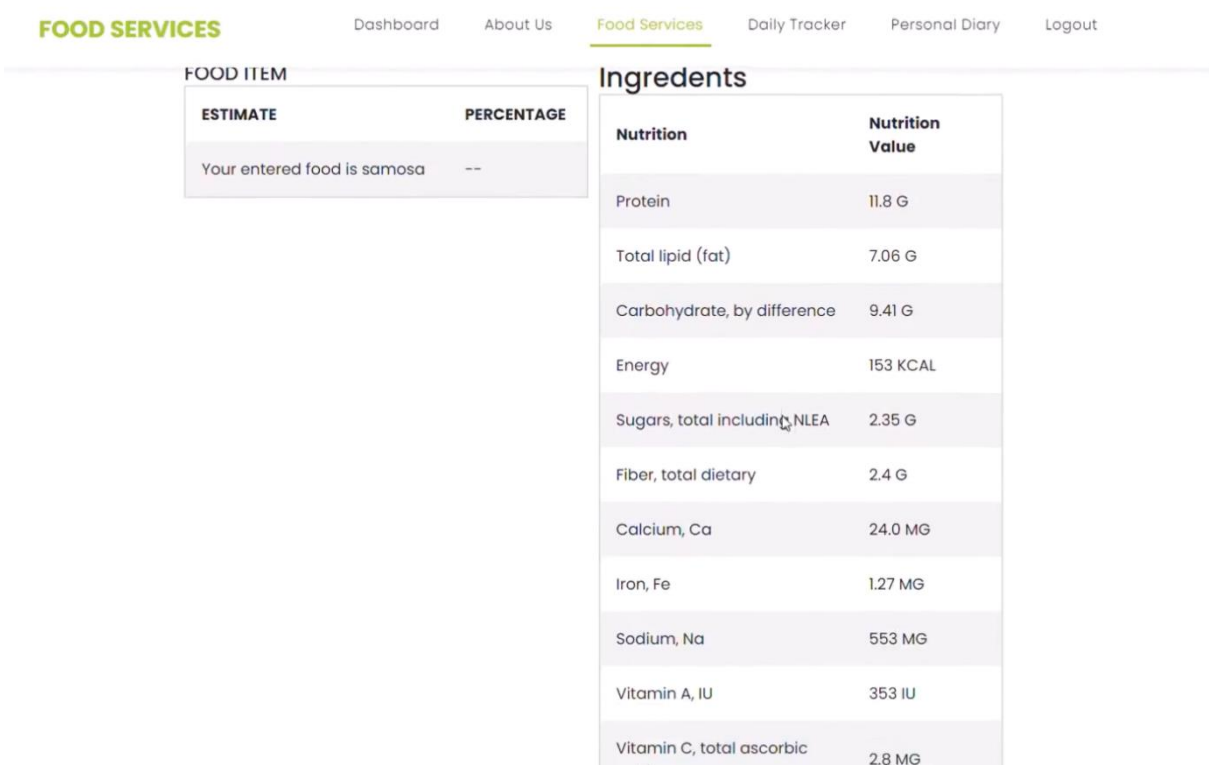
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	28 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

8. CODING & SOLUTIONING

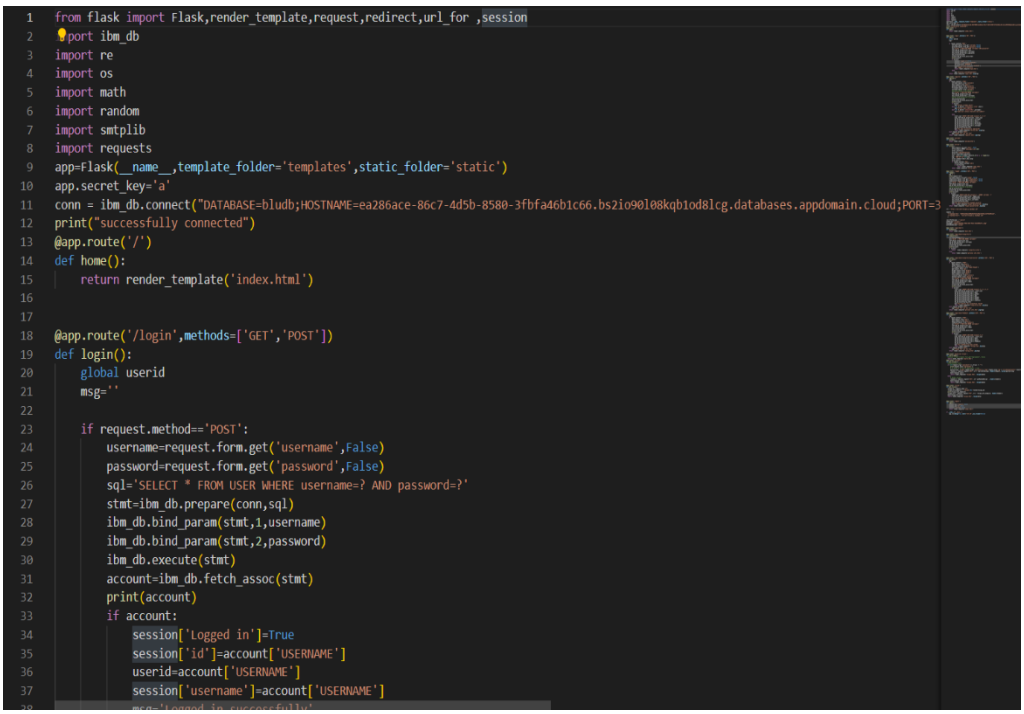
8.1 FEATURE 1

The user can upload any food image Nutrients present in the uploaded image will be displayed


FOOD SERVICES
Dashboard
About Us
Food Services
Daily Tracker
Personal Diary
Logout
Please Enter the image url
image Url
ok
Please uploand an image
Choose File
No file chosen
ok
Please enter the food name
enter food name
ok
If the photo is blurr/not clear our services may find it difficult to process,so please upload clear image.
In case not working, enter the food name with proper spelling.
FOOD ITEM
ESTIMATE
PERCENTAGE
Ingredents
Nutrition
Nutrition Value



8.2 FEATURE 2



```python
from flask import Flask,render_template,request,redirect,url_for ,session
import ibm_db
import re
import os
import math
import random
import smtplib
import requests
app=Flask(__name__,template_folder='templates',static_folder='static')
app.secret_key='a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=3
print("successfully connected")
@app.route('/')
def home():
    return render_template('index.html')


@app.route('/login',methods=['GET','POST'])
def login():
    global userid
    msg=''

    if request.method=='POST':
        username=request.form.get('username',False)
        password=request.form.get('password',False)
        sql='SELECT * FROM USER WHERE username=? AND password=?'
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Logged in']=True
            session['id']=account['USERNAME']
            userid=account['USERNAME']
            session['username']=account['USERNAME']
```

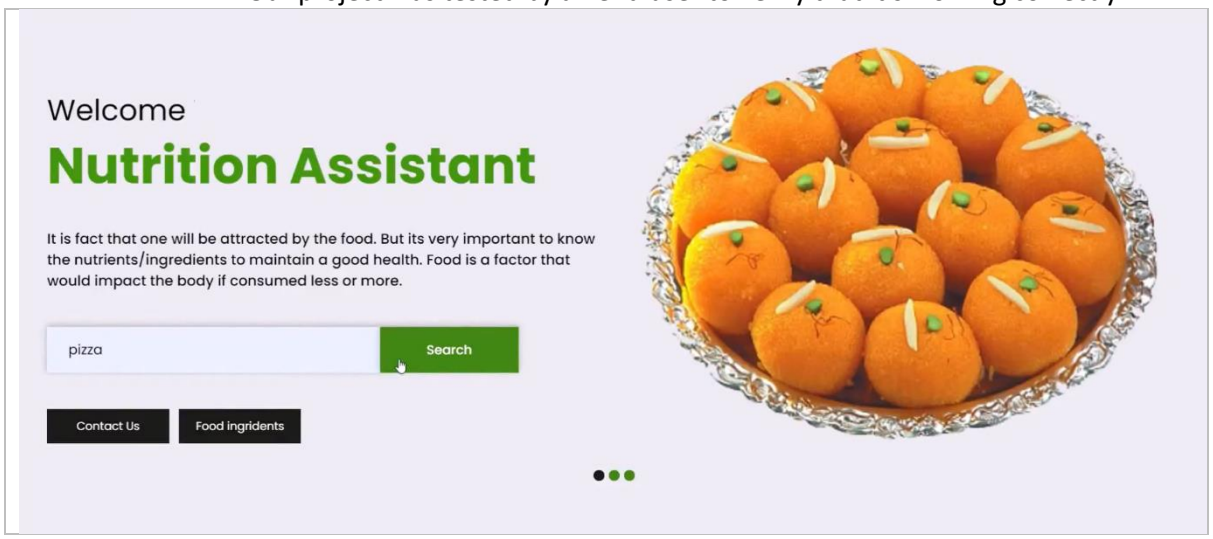
9.TESTING

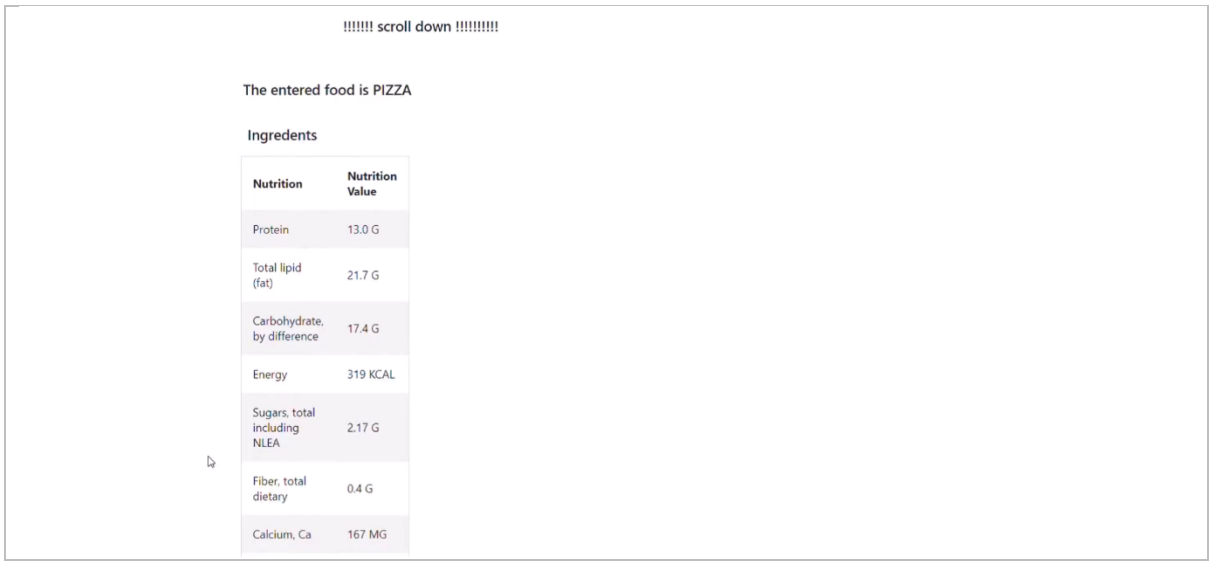
9.1 TEST CASES

1. Our code was tested on various food to check whether it gives the correct output
2. To satisfy the customer's expectations we tested it fully.

9.2 USER ACCEPTANCE TESTING

Our project was tested by an end user to verify that it's working correctly.

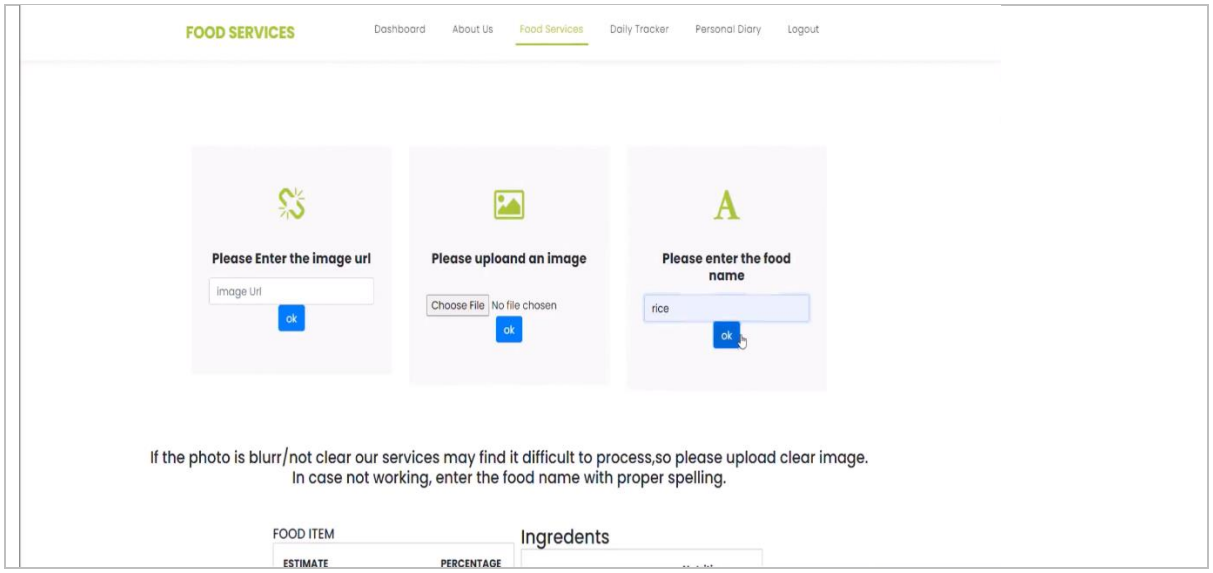


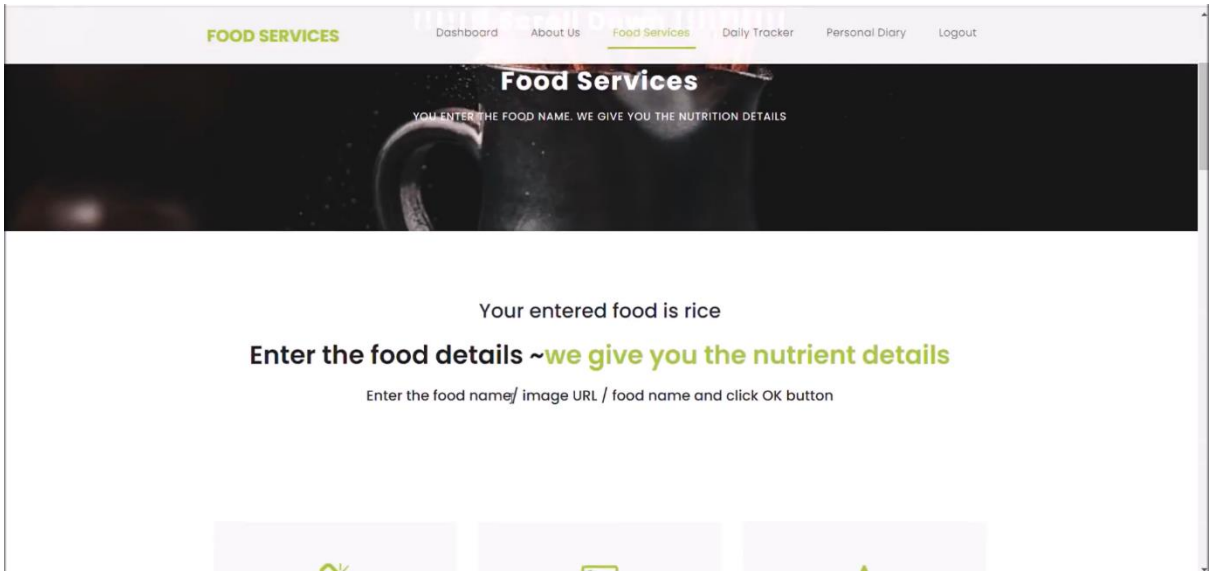


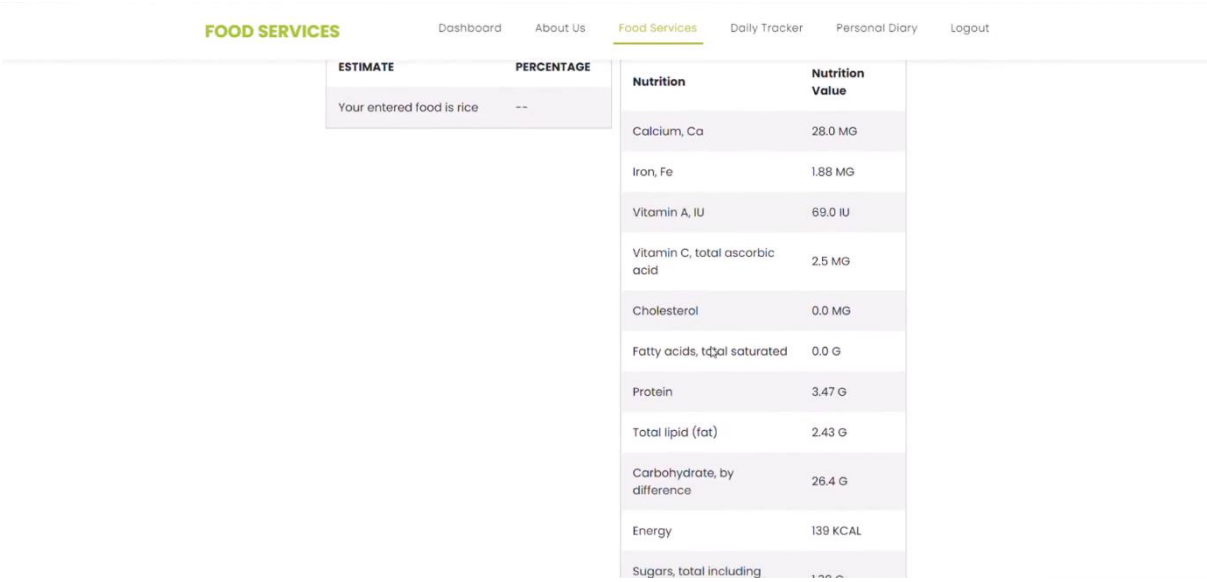
10. RESULT

10.1 PERFORMANCE METRICS

The proposed procedure was implemented and tested set of images. The training database consists of various images of food items. Once a food is recognized the equivalent **Nutrition** in shown on the screen.







11. ADVANTAGES:

- It provides a maintained strategy of healthy eating habits.
- It delivers information on the nutritional value of foods and how balanced and healthy eating habits are important for us.
- It limits the amount of unnecessary food such as fat that people consume a lot.

12. CONCLUSION

In conclusion, many people have become aware of their health. Moreover, they are also informed how to live a healthy lifestyle. Most of the research related to these themes aims to identify changes in healthy lifestyle behavior with web applications that are considered effective in dietary self-monitoring.

13. DESCRIPTION:

Nutrition assistants help dieticians with providing proper nutrition at healthcare facilities. They determine patients' nutritional needs, assess risk factors, and plan meals and menus. They also ensure proper sterilization of plates and utensils.

14. APPENDIX

Source Code:

```
from flask import Flask,render_template,request,redirect,url_for ,session
import ibm_db
import re
import os
import math
import random
import smtplib
import requests
app=Flask(__name__,template_folder='templates',static_folder='static')
app.secret_key='a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=3
print("successfully connected")
@app.route('/')
def home():
    return render_template('index.html')


@app.route('/login',methods=['GET','POST'])
def login():
    global userid
    msg=''

    if request.method=='POST':
        username=request.form.get('username',False)
        password=request.form.get('password',False)
        sql='SELECT * FROM USER WHERE username=? AND password=?'
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Logged in']=True
            session['id']=account['USERNAME']
            userid=account['USERNAME']
            session['username']=account['USERNAME']
            msg='Logged in successfully'
```

```
            return render_template('dash.html')
        else:
            msg='Incorrect username/password'
    return render_template('login.html',msg=msg)

@app.route('/register',methods=['GET','POST'])
def register():
    msg=''
    if request.method =='POST':
        username=request.form['username']
        email=request.form['email']
        password=request.form['password']
        Firstname=request.form['firstname']
        lastname=request.form['lastname']
        #phoneno=request.form['phoneno']
        sql='SELECT * FROM USER WHERE username=?'
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        #ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg="Account already exist!"
        elif not re.match(r'[^@]+@[^@]+\.[^@]+',email):
            msg="Invalid email address"
        elif not re.match(r'[A-Za-z0-9]+',username):
            msg="name must contain character and numbers"

        else:
            insert_sql='INSERT INTO USER values(?,?,?,?,?)'
            prep_stmt=ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt,1,username)
            ibm_db.bind_param(prep_stmt,2,email)
            ibm_db.bind_param(prep_stmt,3,password)
            ibm_db.bind_param(prep_stmt,4,Firstname)
            ibm_db.bind_param(prep_stmt,5,lastname)
            ibm_db.execute(prep_stmt)
```

```
            msg="You have successfully registered"
            return render_template('verify.html',msg=msg)
    elif request.method=="POST":
        msg="Please fill out the form"
    return render_template('register.html',msg=msg)

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

@app.route('/verify')
def verify():
        email=request.args.get('email', None)
        server=smtplib.SMTP('smtp.gmail.com',587)
        server.starttls()
        password="nsgeuedwbzptosyp"
        server.login(email,password)
        otp=''.join([str(random.randint(0,9))for i in range(4)])
        msg=' YOUR OTP IS'+str(otp)
        server.sendmail(email,email,msg)
        server.quit()
        if request.method=='POST':
            verify=request.method['code']
            if verify==otp:
                return render_template('login.html')
        return render_template('verify.html')

@app.route('/frgpwd', methods=['GET','POST'])
def frgpwd():
    msg =" "
    print(request.form)
    username1=request.form.get("uname", False)
    oldpassword=request.form.get("oldpassword", False)
    newpassword=request.form.get("newpassword", False)
    sql='SELECT * FROM USER WHERE username=?'
    stmt=ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,username1)
```

```
        chgpwd_sql='UPDATE USER SET password = ? WHERE username = ?'
        prep_stmt=ibm_db.prepare(conn, chgpwd_sql)
        ibm_db.bind_param(prep_stmt,1,newpassword)
        ibm_db.bind_param(prep_stmt,2,username1)
        ibm_db.execute(prep_stmt)
        msg="You have successfully changed password"
        return render_template('forgot password.html',msg=msg)
    return render_template('forgot password.html',msg=msg)

url = "https://low-carb-recipes.p.rapidapi.com"

headers = {
  "x-rapidapi-key": "ad933ea36amsh6b0a83e514b1a58p14bc9ejsne745a5851a1b",
  "x-rapidapi-host":  "low-carb-recipes.p.rapidapi.com"
  }

searchForRecipes = "/search"
getRecipe="/recipes/"
getImage="/images/2807982c-986a-4def-9e3a-153a3066af7a.jpeg"
getRandomRecipe="/random"

@app.route('/login/dash')
def dashboard():
    return render_template('dash.html')

@app.route('/login/dash/viewprofile')
def viewprofile():
    username=session['id']
    sql='SELECT * FROM USER WHERE username=?'
    stmt=ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    account=ibm_db.fetch_assoc(stmt)
    print(account)
    if account:
        return  render_template('viewprofile.html')
    else:
```

```
@app.route('/login/dash/viewprofile/personinfo',methods=['GET','POST'])
def per_info():
    msg=''
    if request.method =='POST':
        Name=request.form['Name']
        gender=request.form['gender']
        tar_weight=request.form['Target Weight']
        Age=request.form['Age']
        Height=request.form['Height']
        Weight=request.form['Weight']
        email=request.form['email']
        location=request.form['location']
        phoneno=request.form['phoneno']
        sql='SELECT * FROM USER WHERE username=?'
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,Name)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            insert_sql='INSERT INTO USER values(?,?,?,?,?,?,?)'
            prep_stmt=ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt,1,Name)
            ibm_db.bind_param(prep_stmt,2,gender)
            ibm_db.bind_param(prep_stmt,3,Age)
            ibm_db.bind_param(prep_stmt,4,Height)
            ibm_db.bind_param(prep_stmt,5,Weight)
            ibm_db.bind_param(prep_stmt,7,location)
            ibm_db.execute(prep_stmt)
            msg="Your details are successfully stored"
            return render_template('viewprofile.html',msg=msg)
    elif request.method=="POST":
        msg="Please fill out the form"
    return render_template('personal info.html',msg=msg)
```

```
        if account:
            insert_sql='INSERT INTO USER values(?,?,?)'
            prep_stmt=ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt,1,Name)
            ibm_db.bind_param(prep_stmt,2,email)
            ibm_db.bind_param(prep_stmt,3,Feedback)
            ibm_db.execute(prep_stmt)
            msg="Your Feedback has been stored"
            return render_template('ratings.html',msg=msg)
    elif request.method=="POST":
        msg="Please fill out the form"
    return render_template('ratings.html',msg=msg)

@app.route('/dash/view recipe')
def search_page():
  #session ['item']=request.form.get("Ingridients", False)
  return render_template('search.html')
@app.route('/recipes')
def get_recipes():
  #food=session['item']
  if (str(request.args['ingridients']).strip() != ""):
      print(request.args['ingridients'])
      # If there is a list of ingridients -> list
      querystring = {"name":request.args['ingridients'],"tags":request.args['tag'],"includeIngredients":request.args['included'],"exclude
      response = requests.request("GET", url + searchForRecipes, headers=headers, params=querystring)
      data=response.json()
      return render_template('recipes.html', recipes=data)
  else:
      # Random recipes
      response = requests.request("GET", url+ getRandomRecipe  , headers=headers)
      data=response.json()
      return render_template('recipes.html', recipes=data)

@app.route('/recipe')
def get_recipe():
  recipe_id = request.args['id']
  recipe_info_endpoint = "/recipes/{0}".format(recipe_id)
```

```
        data=response.json()
        return render_template('recipes.html', recipes=data)

@app.route('/recipe')
def get_recipe():
  recipe_id = request.args['id']
  recipe_info_endpoint = "/recipes/{0}".format(recipe_id)
  print(recipe_info_endpoint)
  recipe_info = requests.request("GET", url + recipe_info_endpoint, headers=headers)
  data=recipe_info.json()
  return render_template('recipe.html', recipe=data)


@app.route('/logout')
def logout():
    session.pop('loggedin',None)
    session.pop('id',None)
    session('username',None)
    return render_template("index.html")

if __name__=="__main__":
    app.run(debug=True ,host='0.0.0.0',use_reloader=False)
```

GitHub: https://github.com/IBM-EPBL/IBM-Project-3292-1658513627

PROJECT DEMONSTRATION LINK:
https://drive.google.com/file/d/1rhuXkmsZ7v6bEWRJbTTmHRHo0WHg-H26/view?usp=share_link