Project Report

NUTRITION ASSISTANT APPLICATION

S.NO	REG.NO	NAME	DEPARTMENT	TEAM
1.	312419104009	AKSHARA	CSE	Team Lead
2.	312419104023	BEAUTLIN S	CSE	Team Member1
3.	312419104021	ASWINI	CSE	Team Member2
4.	312419104003	AISWARYA	CSE	Team Member3

DONE BY

TEAM ID: PNT2022TMID28040

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 metto 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 tobe 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- DrivenFood 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 typesand cuisines, convert ingredient amounts, or even compute an entire meal plan.

LITERATURE SURVEY

1) Existing problem

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.

2) References

Flask : https://www.youtube.com/embed/uxZuFm5tmhM

Send-Grid : https://sendgrid.com/

Rapid API : https://rapidapi.com/hub

Docker :https://www.youtube.com/embed/pTFZFxd4hOI

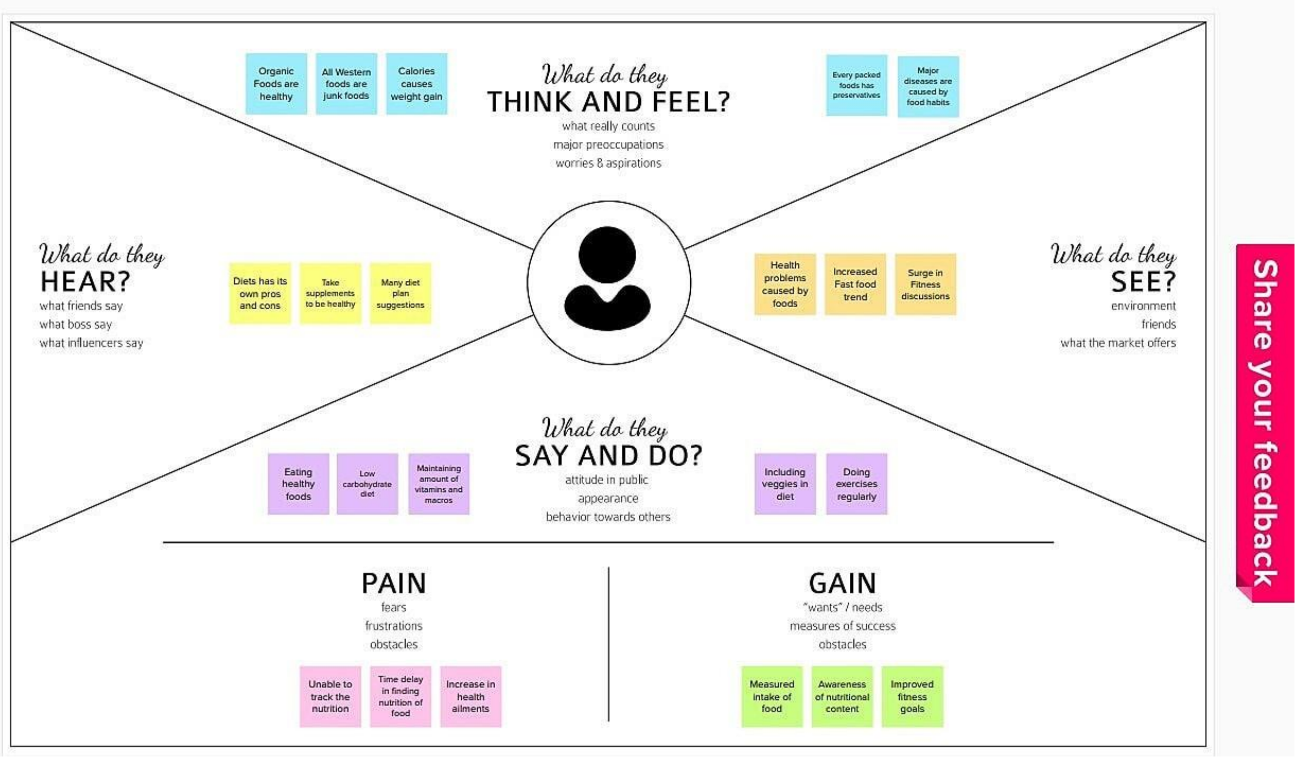
Kubernetes : https://www.youtube.com/embed/d6WC5n9G_sM

3) Problem Statement Definition

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.

IDEATION & PROPOSED SOLUTION

1) Empathy Map Canvas



What do they THINK AND FEEL?

what really counts
major preoccupations
worries & aspirations

Organic Foods are healthy

All Western foods are junk foods

Calories causes weight gain

Every packed foods has preservatives

Major diseases are caused by food habits

What do they HEAR?

what friends say
what boss say
what influencers say

Diets has its own pros and cons

Take supplements to be healthy

Many diet plan suggestions

What do they SEE?

environment
friends
what the market offers

Health problems caused by foods

Increased Fast food trend

Surge in Fitness discussions

What do they SAY AND DO?

attitude in public
appearance
behavior towards others

Eating healthy foods

Low carbohydrate diet

Maintaining amount of vitamins and macros

Including veggies in diet

Doing exercises regularly

PAIN

fears
frustrations
obstacles

Unable to track the nutrition

Time delay in finding nutrition of food

Increase in health ailments

GAIN

"wants" / needs
measures of success
obstacles

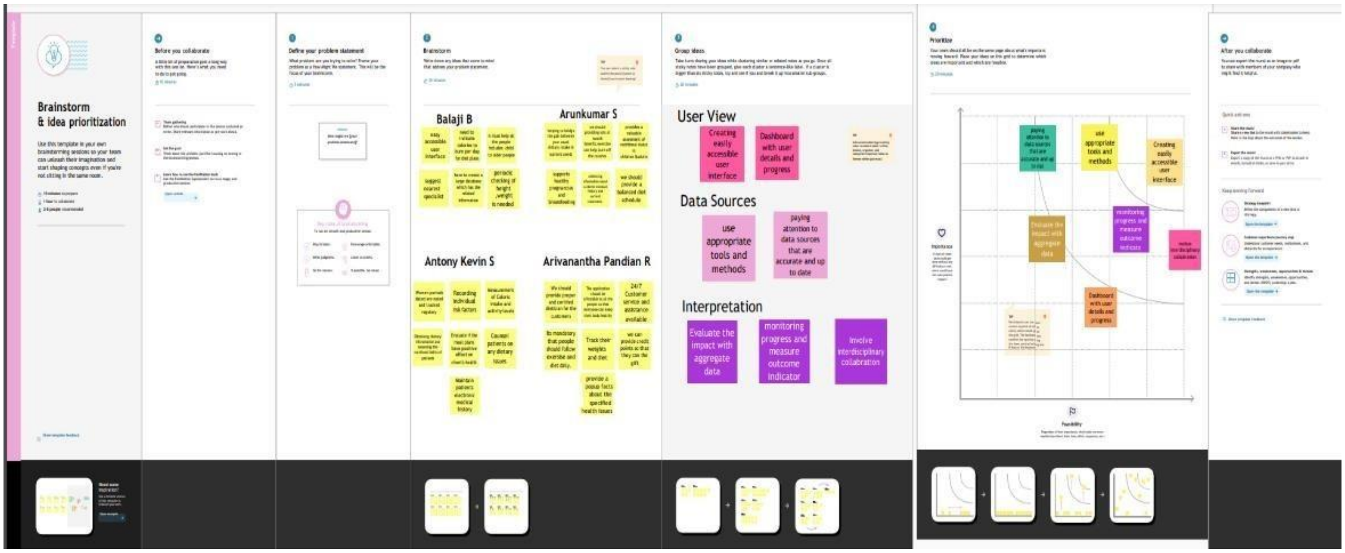
Measured intake of food

Awareness of nutritional content

Improved fitness goals

Share your feedback

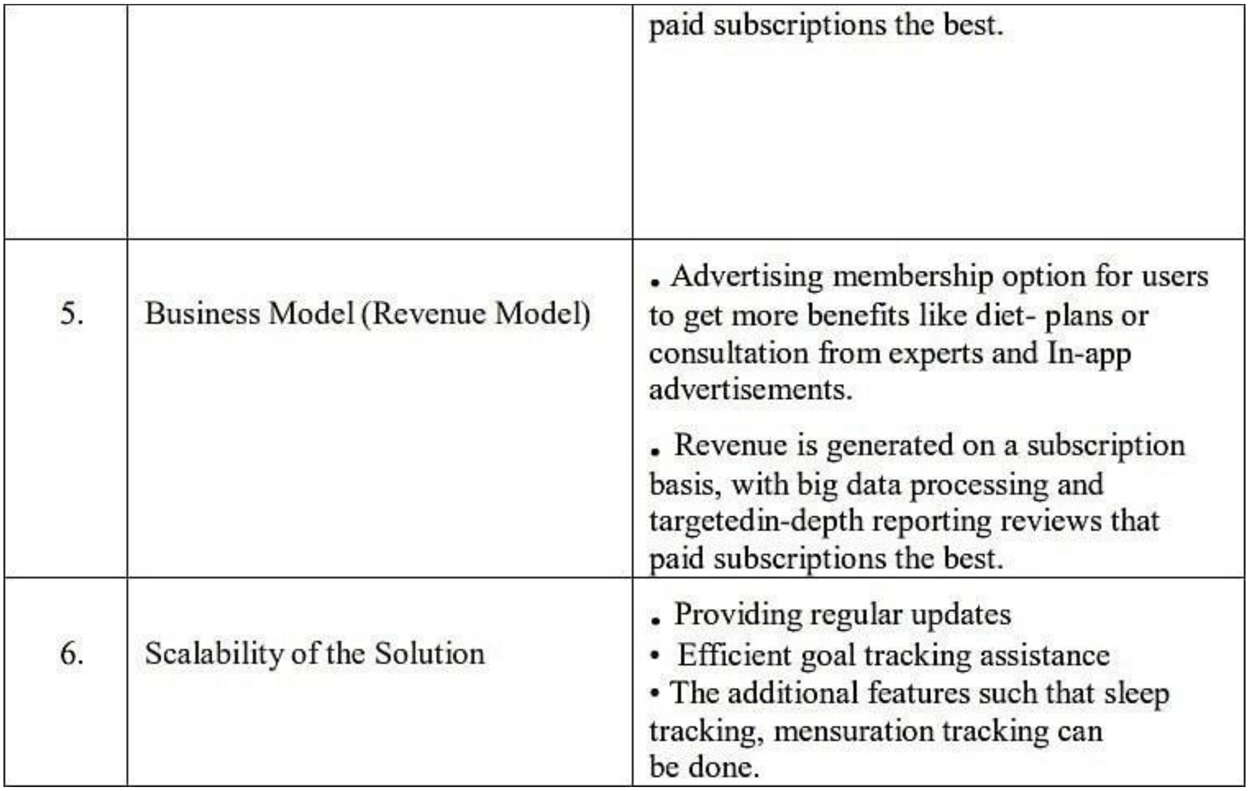
2) Ideation & Brainstorming



Brainstorm & idea prioritization
Before you collaborate
Define your problem statement
Brainstorm
Balaji B
Arunkumar S
Antony Kevin S
Arivanantha Pandian R
Group ideas
User View
Creating easily accessible user interface
Dashboard with user details and progress
Data Sources
use appropriate tools and methods
paying attention to data sources that are accurate and up to date
Interpretation
Evaluate the impact with aggregate data
monitoring progress and measure outcome indicator
Involve interdisciplinary collabration
Prioritize
After you collaborate

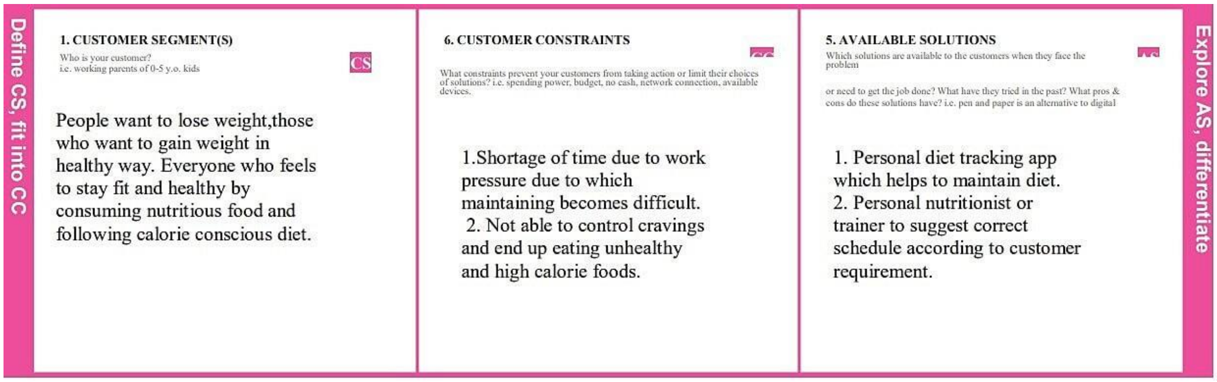
3) Proposed Solution

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.

4) Problem Solution fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

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

CS

People want to lose weight,those who want to gain weight in healthy way. Everyone who feels to stay fit and healthy by consuming nutritious food and following calorie conscious diet.

6. CUSTOMER CONSTRAINTS

CC

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

1.Shortage of time due to work pressure due to which maintaining becomes difficult.
2. Not able to control cravings and end up eating unhealthy and high calorie foods.

5. AVAILABLE SOLUTIONS

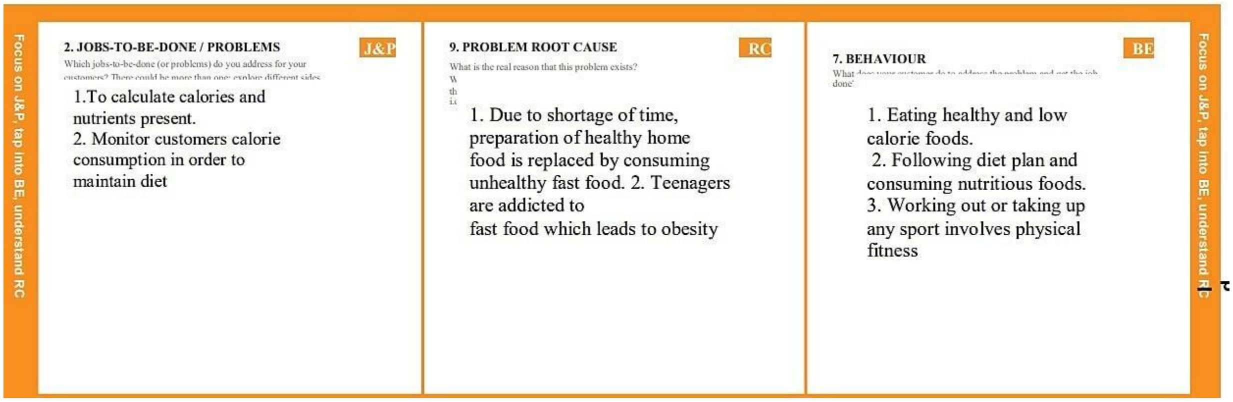
AS

Which solutions are available to the customers when they face the problem

or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital

1. Personal diet tracking app which helps to maintain diet.
2. Personal nutritionist or trainer to suggest correct schedule according to customer requirement.

Explore AS, differentiate



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

2. JOBS-TO-BE-DONE / PROBLEMS

J&P

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

1.To calculate calories and nutrients present.
2. Monitor customers calorie consumption in order to maintain diet

9. PROBLEM ROOT CAUSE

RC

What is the real reason that this problem exists?

1. Due to shortage of time, preparation of healthy home food is replaced by consuming unhealthy fast food. 2. Teenagers are addicted to fast food which leads to obesity

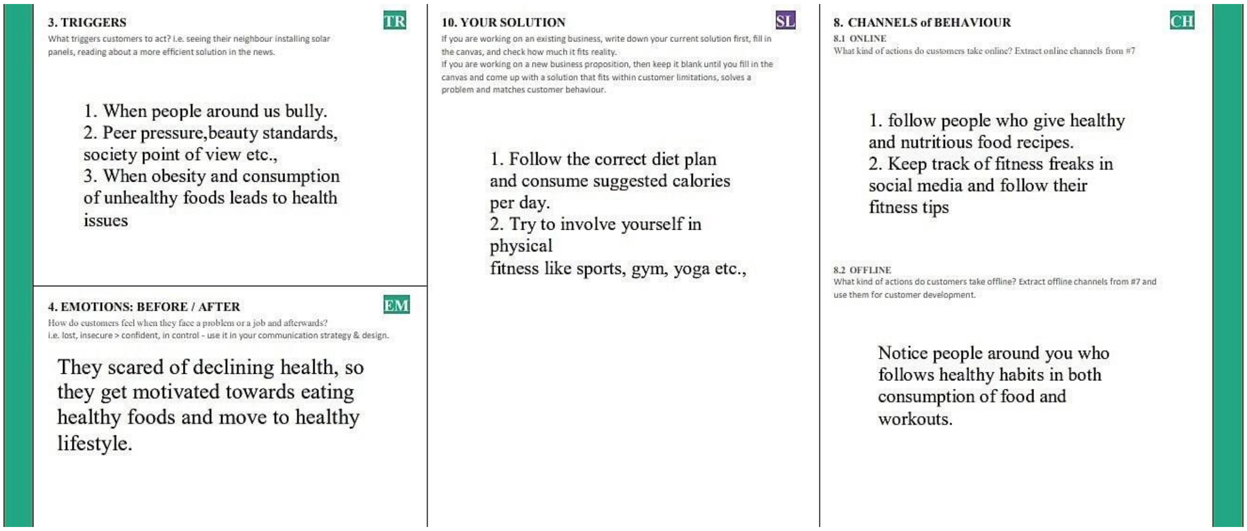
7. BEHAVIOUR

BE

What does your customer do to address the problem and get the job done?

1. Eating healthy and low calorie foods.
2. Following diet plan and consuming nutritious foods.
3. Working out or taking up any sport involves physical fitness

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



3. TRIGGERS TR

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

1. When people around us bully.
2. Peer pressure,beauty standards, society point of view etc.,
3. When obesity and consumption of unhealthy foods leads to health issues

4. EMOTIONS: BEFORE / AFTER EM

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

They scared of declining health, so they get motivated towards eating healthy foods and move to healthy lifestyle.

10. YOUR SOLUTION SL

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

1. Follow the correct diet plan and consume suggested calories per day.
2. Try to involve yourself in physical
fitness like sports, gym, yoga etc.,

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

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

1. follow people who give healthy and nutritious food recipes.
2. Keep track of fitness freaks in social media and follow their fitness tips

8.2 OFFLINE

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

Notice people around you who follows healthy habits in both consumption of food and workouts.

REQUIREMENT ANALYSIS

1) Functional requirement

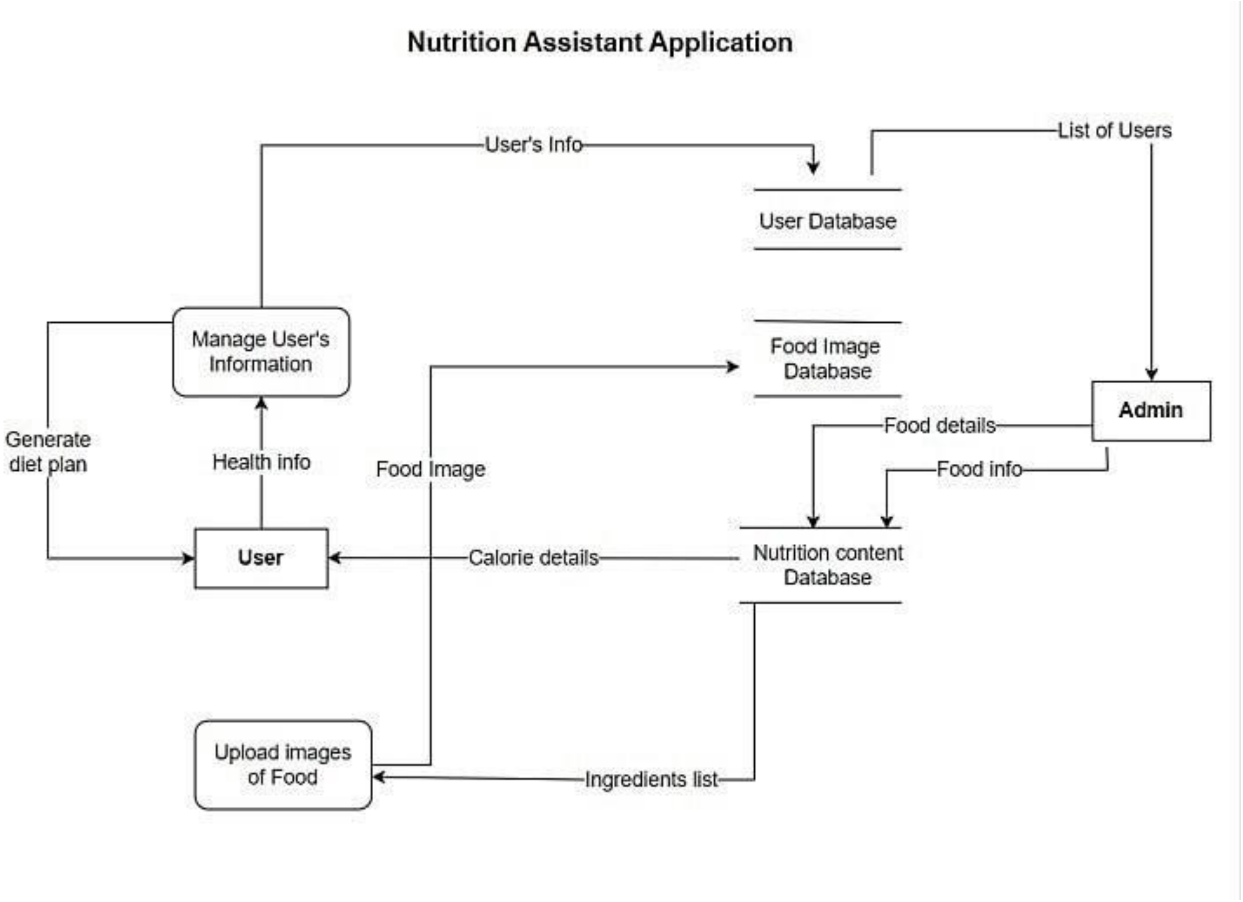
- User Registration
- User Confirmation
- Update Profile
- User Authentication
- Report

2) Non-Functional requirements

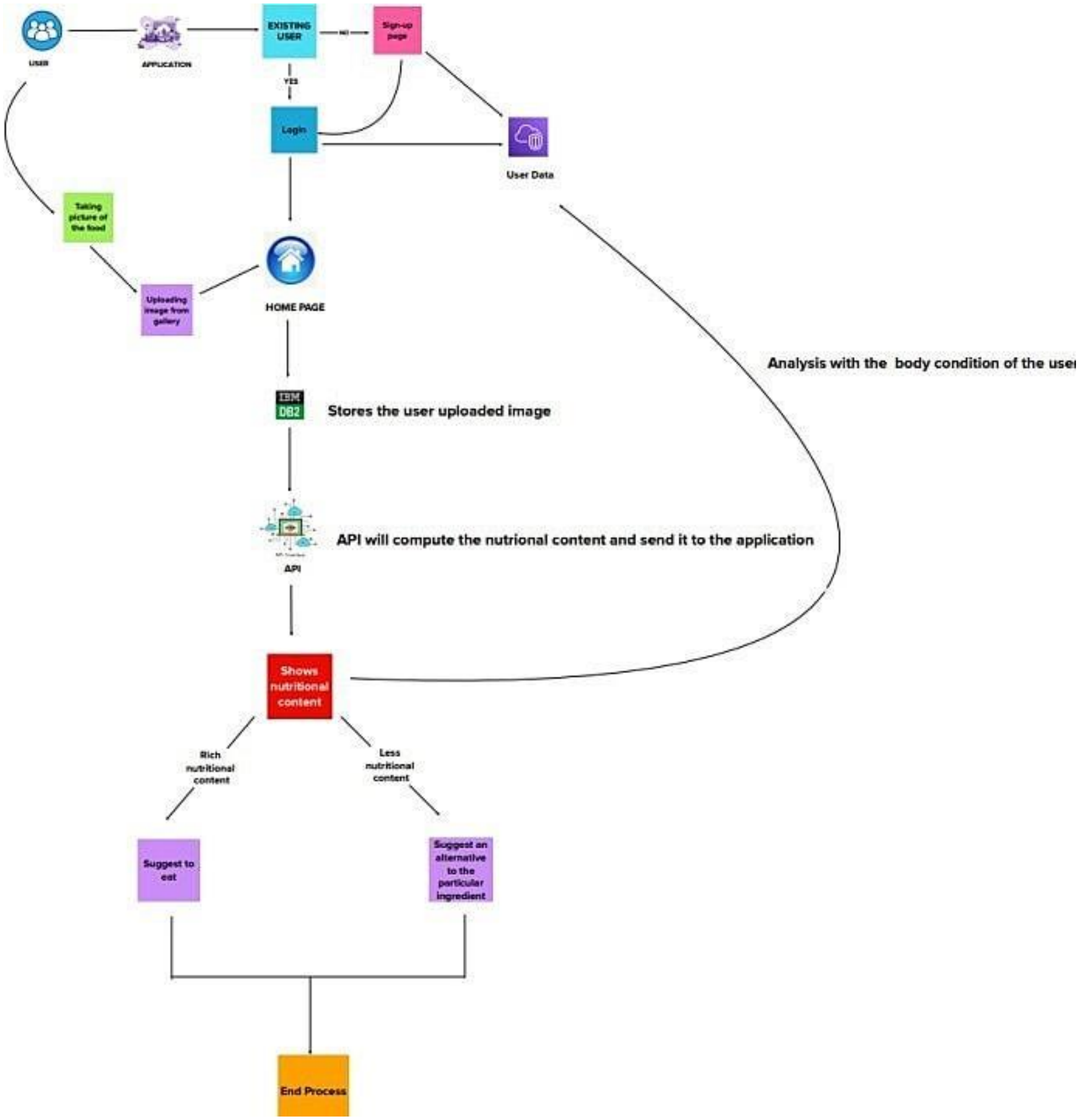
- Usability
- Security
- Reliability
- Performance
- Availability
- Scalability

PROJECT DESIGN

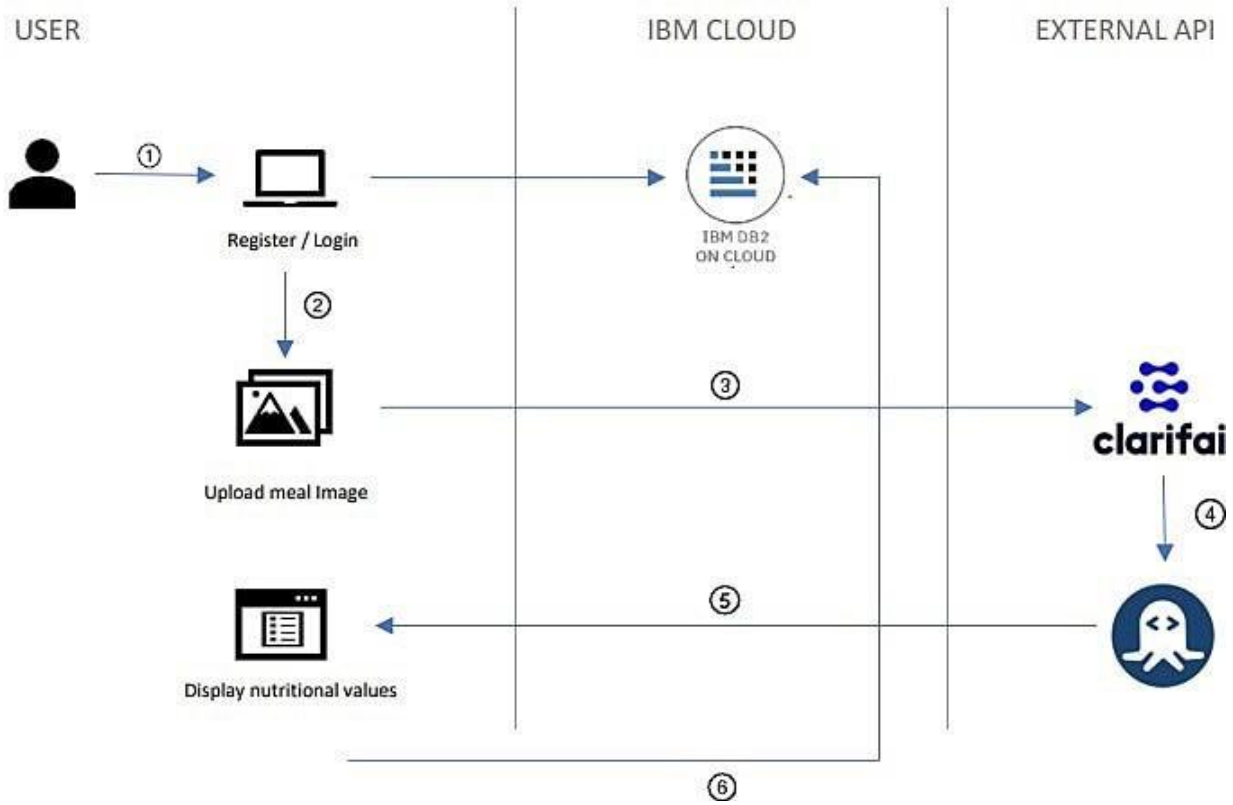
1) Data Flow Diagrams



2) Solution & Technical Architecture


USER
APPLICATION
EXISTING USER
NO
Sign-up page
YES
Login
User Data
Taking picture of the food
Uploading image from gallery
HOME PAGE
IBM DB2
Stores the user uploaded image
Analysis with the body condition of the use
API
API will compute the nutrional content and send it to the application
Shows nutritional content
Rich nutritional content
Less nutritional content
Suggest to eat
Suggest an alternative to the particular ingredient
End Process

Technical Architecture:

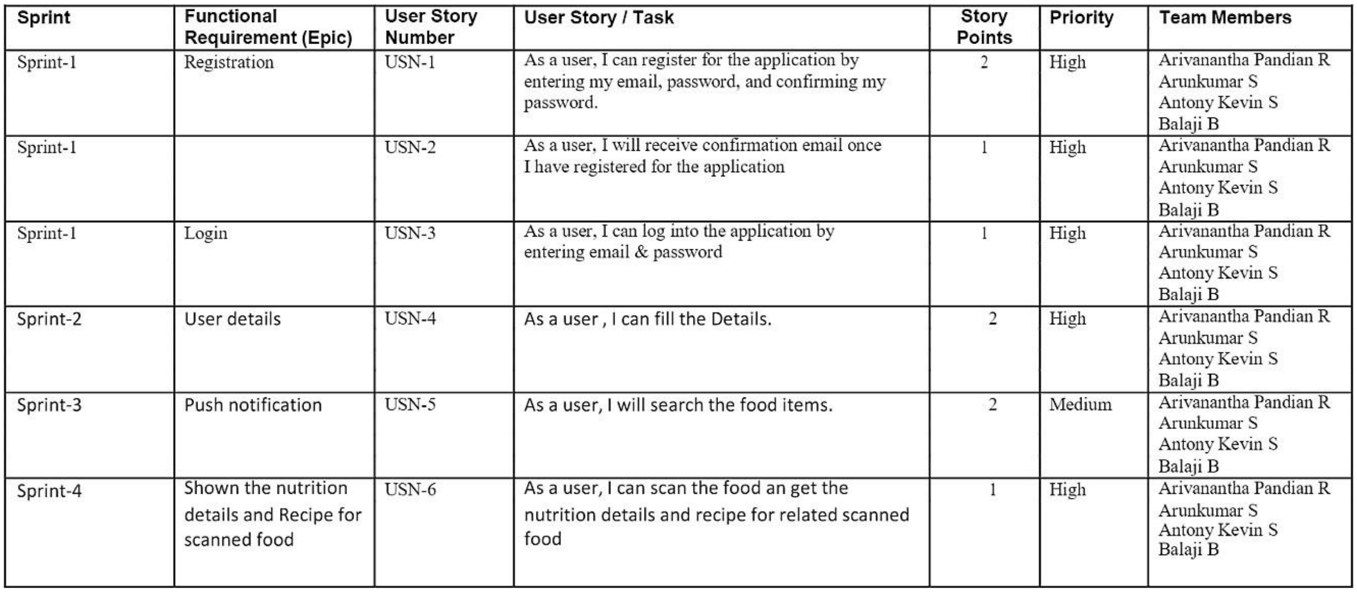


3) User Stories

- As a user, I can register for the application by entering my email, password, andConfirming my password
- As a user, I will receive confirmation email once I have registered for theapplication
- 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 relatedscanned food.

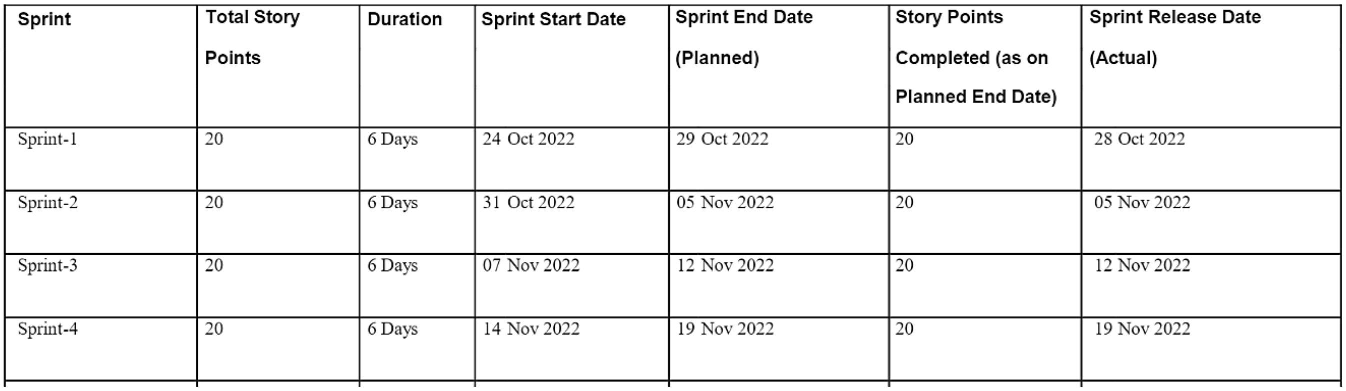
PROJECT PLANNING & SCHEDULING

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

2) Sprint Delivery Schedule

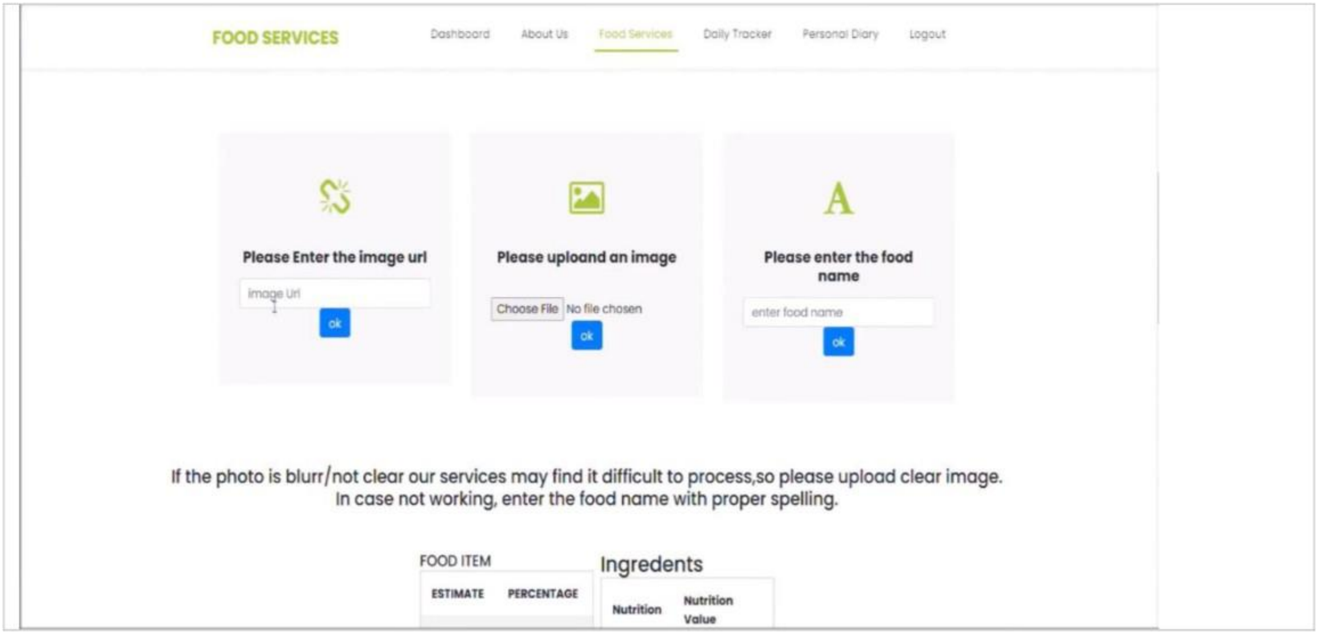


Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	28 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

CODING & SOLUTIONING

1) Feature 1

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



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

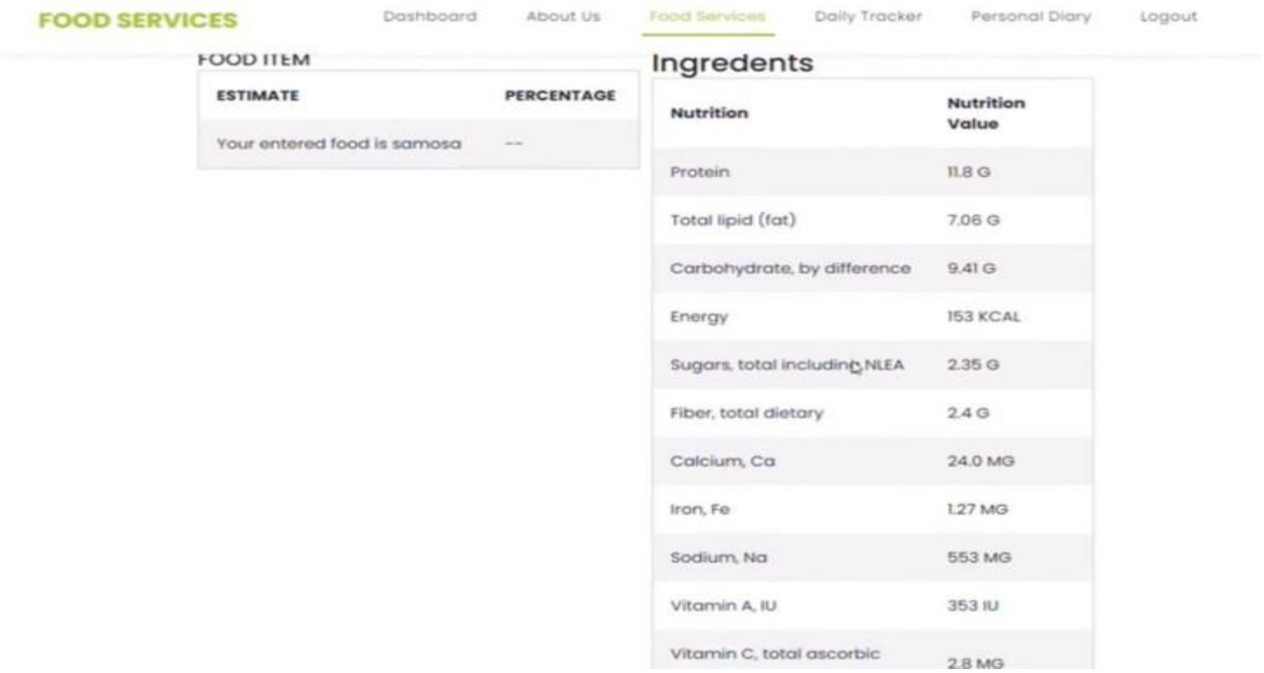
If the photo is blurr/not clear our services may find it difficult to process,so please upload clear image.
In case not working, enter the food name with proper spelling.

FOOD ITEM

ESTIMATE	PERCENTAGE

Ingredents

Nutrition	Nutrition Value



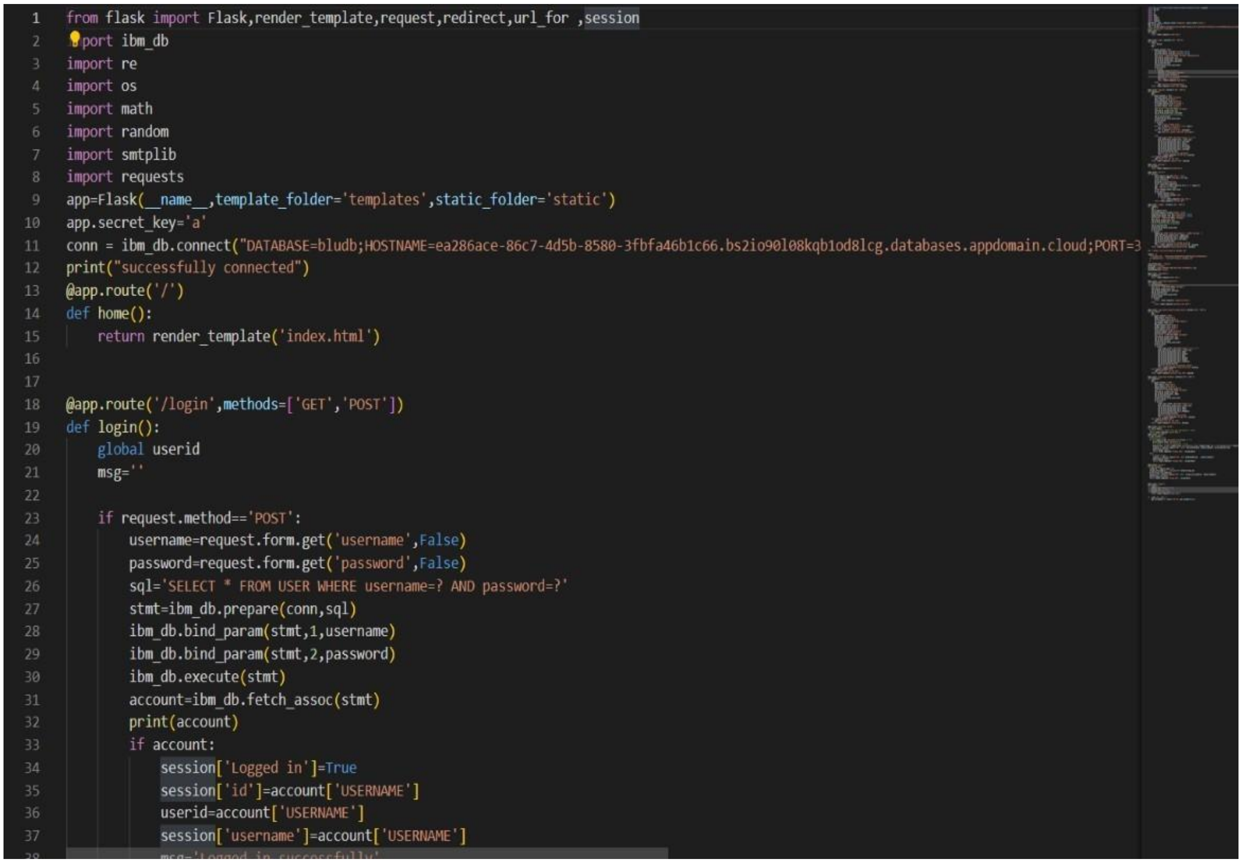
FOOD ITEM

ESTIMATE	PERCENTAGE
Your entered food is samosa	--

Ingredents

Nutrition	Nutrition Value
Protein	11.8 G
Total lipid (fat)	7.06 G
Carbohydrate, by difference	9.41 G
Energy	153 KCAL
Sugars, total including NLEA	2.35 G
Fiber, total dietary	2.4 G
Calcium, Ca	24.0 MG
Iron, Fe	1.27 MG
Sodium, Na	553 MG
Vitamin A, IU	353 IU
Vitamin C, total ascorbic	2.8 MG

2) Feature 2



```
from flask import Flask,render_template,request,redirect,url_for ,session
💡port 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'
```

TESTING

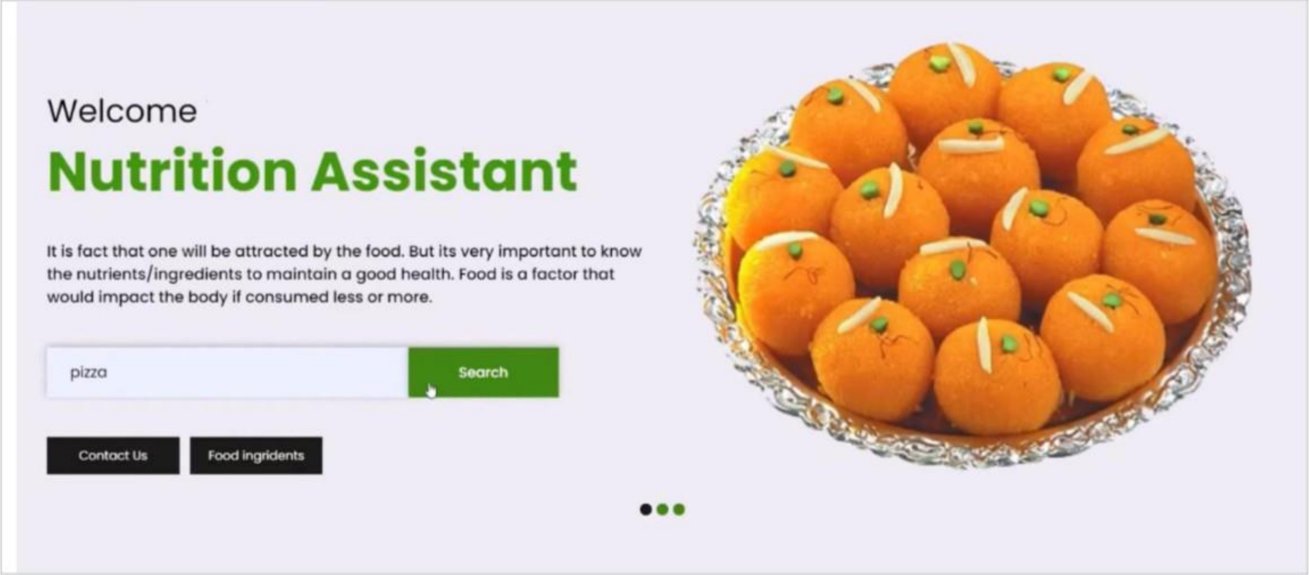
1) Test Cases

i. Our code was tested on various food to check whether it gives the correctoutput

ii. To satisfy the customer’s expectations we tested it fully.

2) User Acceptance Testing

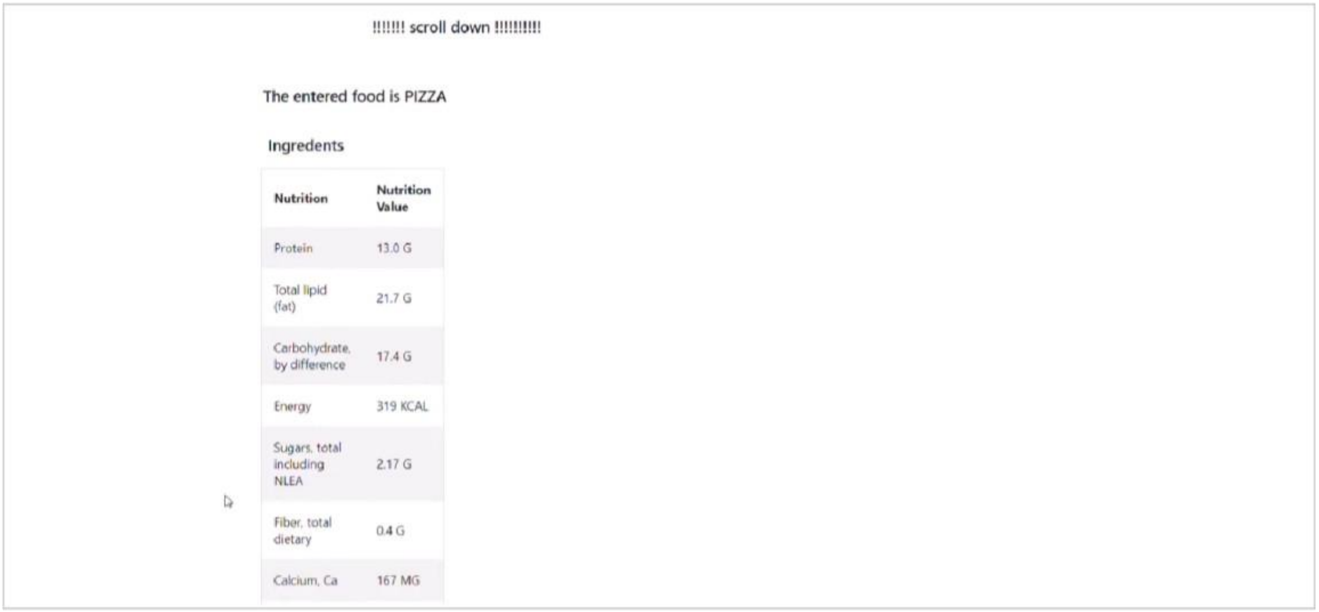
Our project was tested by an end user to verify that it’s working correctly.



Welcome

Nutrition Assistant

It is fact that one will be attracted by the food. But its very important to know the nutrients/ingredients to maintain a good health. Food is a factor that would impact the body if consumed less or more.



!!!!!!! scroll down !!!!!!!!!!

The entered food is PIZZA

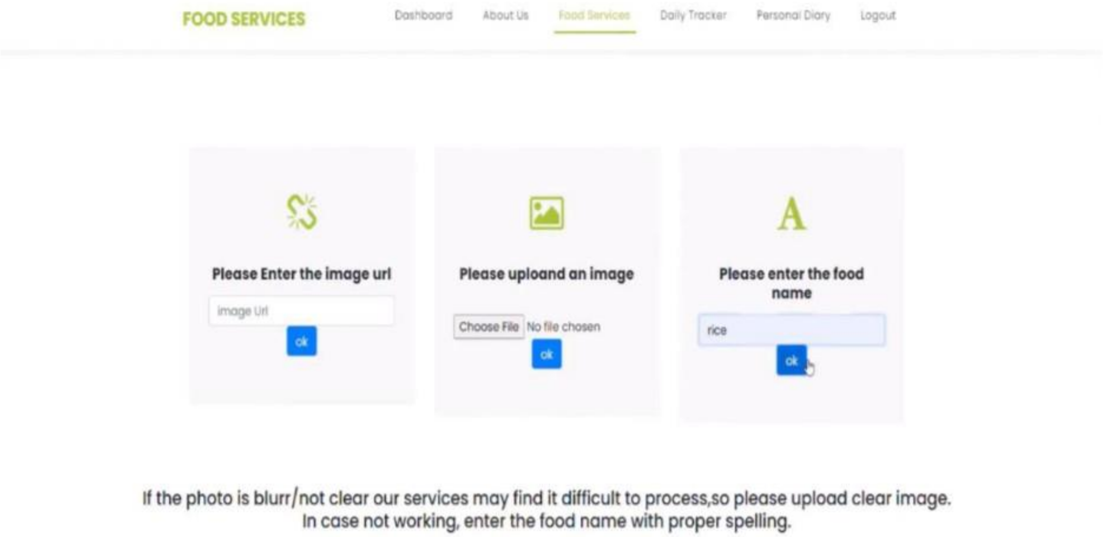
Ingredents

Nutrition	Nutrition Value
Protein	13.0 G
Total lipid (fat)	21.7 G
Carbohydrate, by difference	17.4 G
Energy	319 KCAL
Sugars, total including NLEA	2.17 G
Fiber, total dietary	0.4 G
Calcium, Ca	167 MG

<u>RESULTS</u>

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



Please Enter the image url

image Url

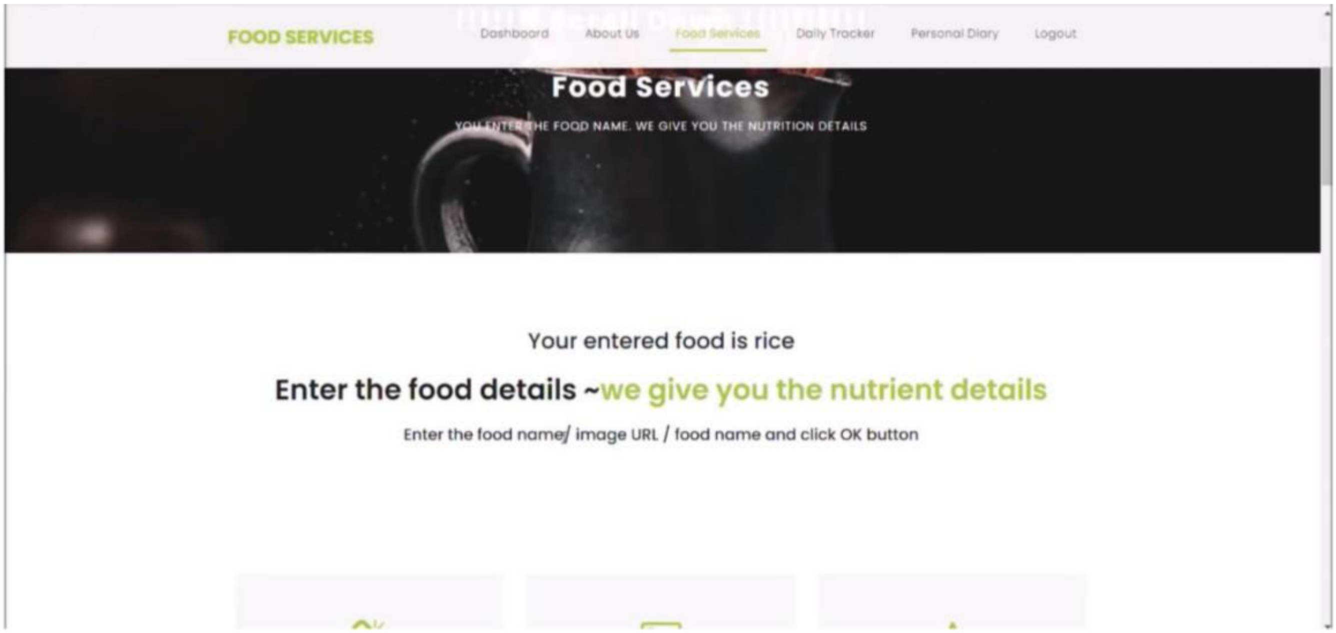
Please uploand an image

No file chosen

Please enter the food name

rice

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.



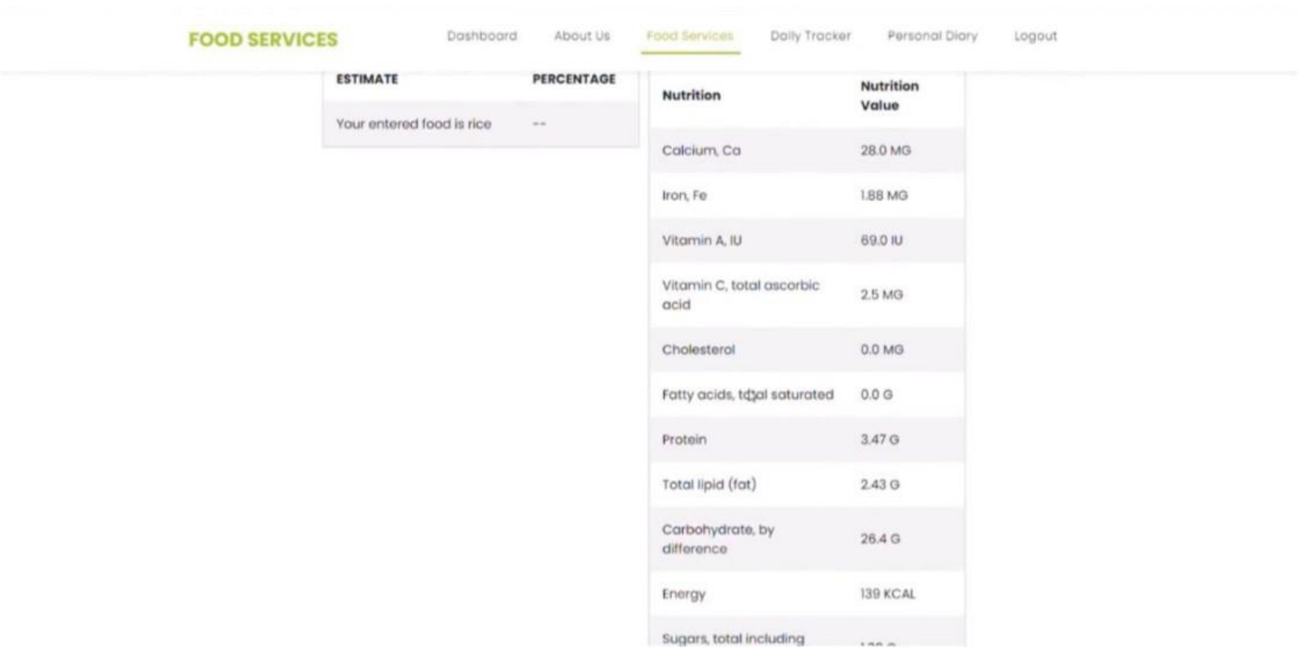
Food Services

YOU ENTER THE FOOD NAME. WE GIVE YOU THE NUTRITION DETAILS

Your entered food is rice

Enter the food details ~we give you the nutrient details

Enter the food name/ image URL / food name and click OK button



ESTIMATE	PERCENTAGE
Your entered food is rice	--

Nutrition	Nutrition Value
Calcium, Ca	28.0 MG
Iron, Fe	1.88 MG
Vitamin A, IU	69.0 IU
Vitamin C, total ascorbic acid	2.5 MG
Cholesterol	0.0 MG
Fatty acids, total saturated	0.0 G
Protein	3.47 G
Total lipid (fat)	2.43 G
Carbohydrate, by difference	26.4 G
Energy	139 KCAL
Sugars, total including	[illegible]

ADVANTAGES

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

CONCLUSION

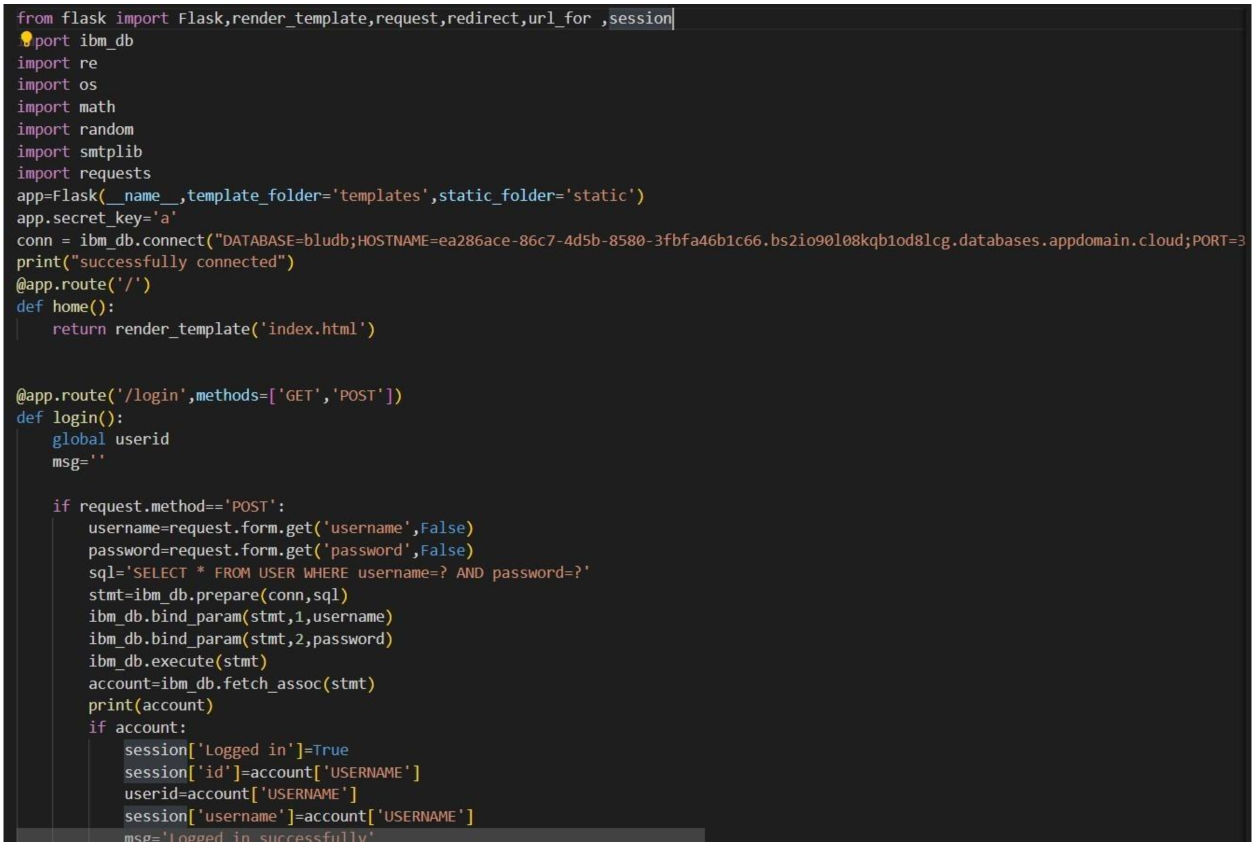
In conclusion, many people have become aware of their health. Moreover, they arealso informed how to live a healthy lifestyle. Most of the research related to these themes aims to identify changes in healthy lifestyle behavior with web applications that are considered effective in dietary self-monitoring.

FUTURE SCOPE

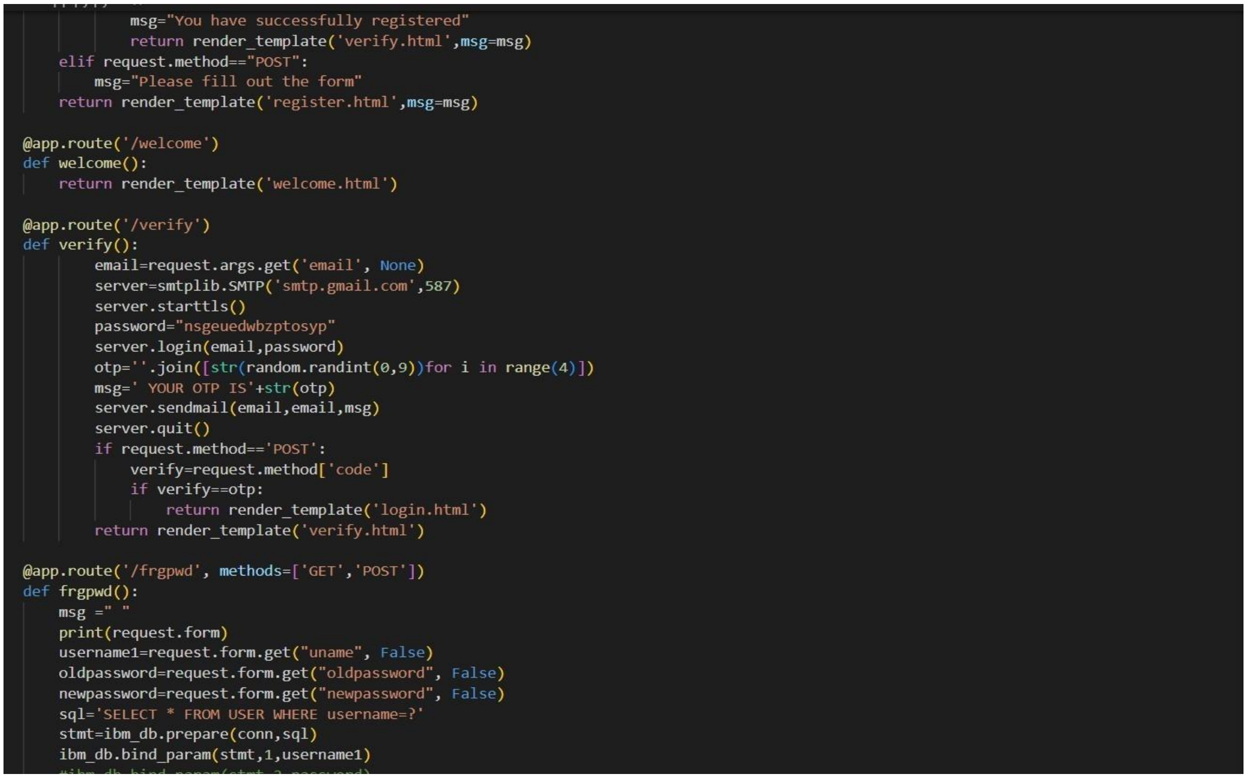
Nutrition assistants help dietitians with providing proper nutrition at healthcare facilities. They determine patients' nutritional needs, assess risk factors, and plan meals and menus.They also ensure proper sterilization of plates and utensils.

APPENDIX

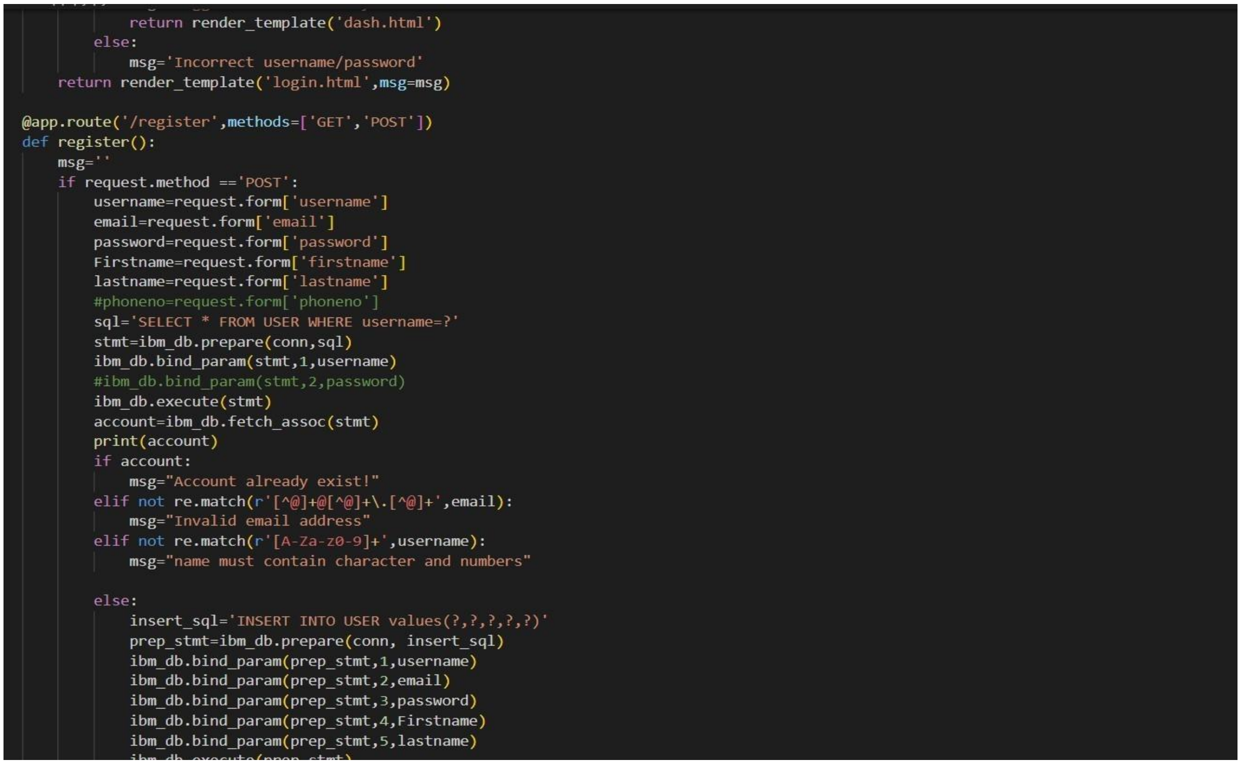
1) Source Code



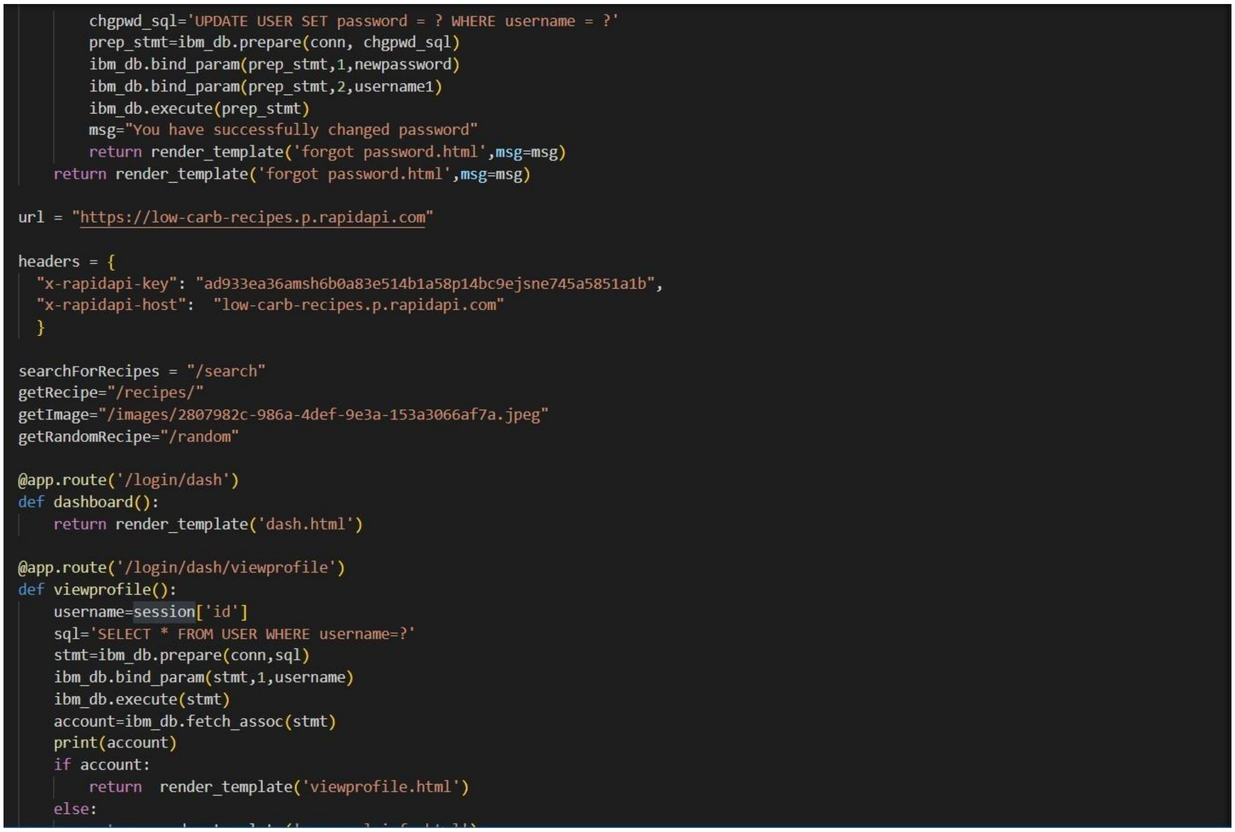
```
from flask import Flask,render_template,request,redirect,url_for ,session
💡port ibm_db
import re
import os
import math
import random
import smtplib
import requests
app=Flask(__name__,template_folder='templates',static_folder='static')
app.secret_key='a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.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'
```



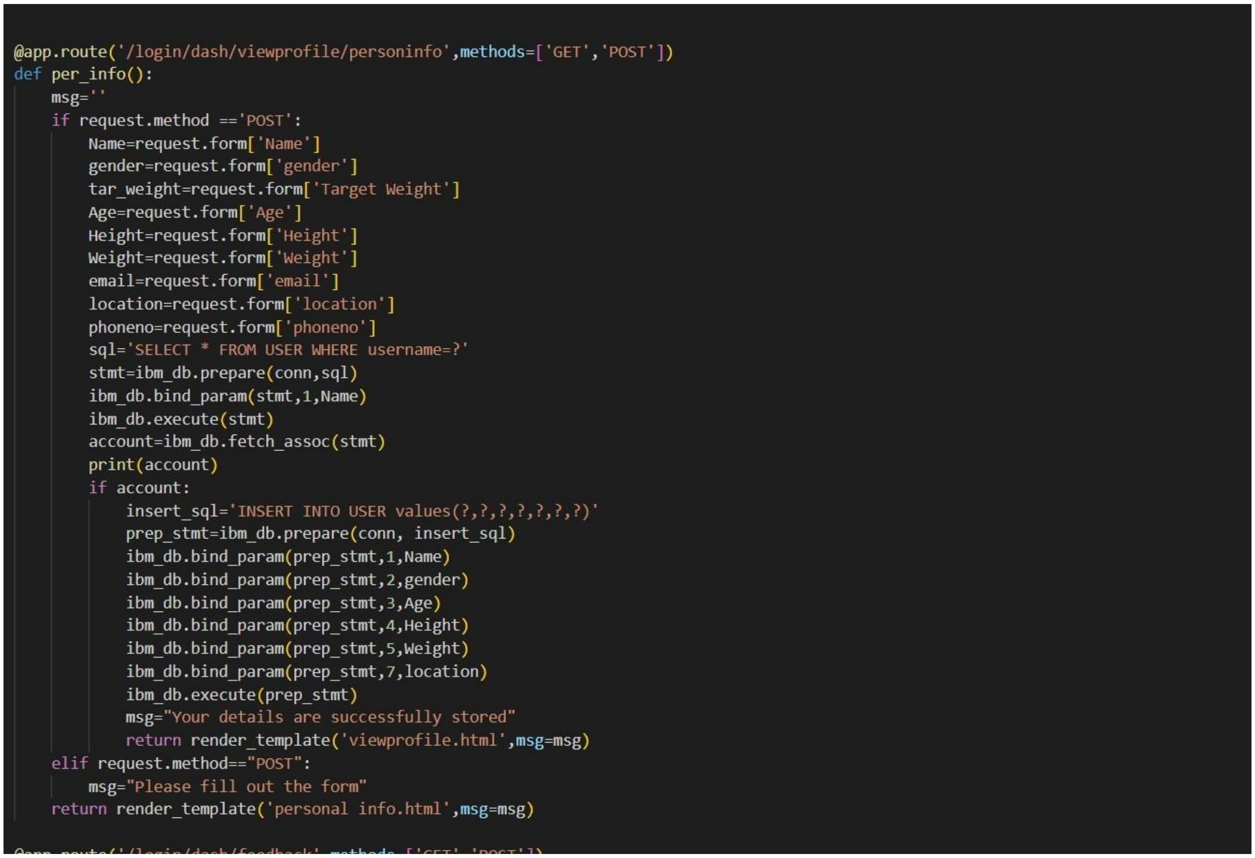
```
            msg="You have successfully registered"
            return render_template('verify.html',msg=msg)
    elif request.method=="POST":
        msg="Please fill out the form"
    return render_template('register.html',msg=msg)

@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)
    #ibm_db.bind_param(stmt,2,password)
```



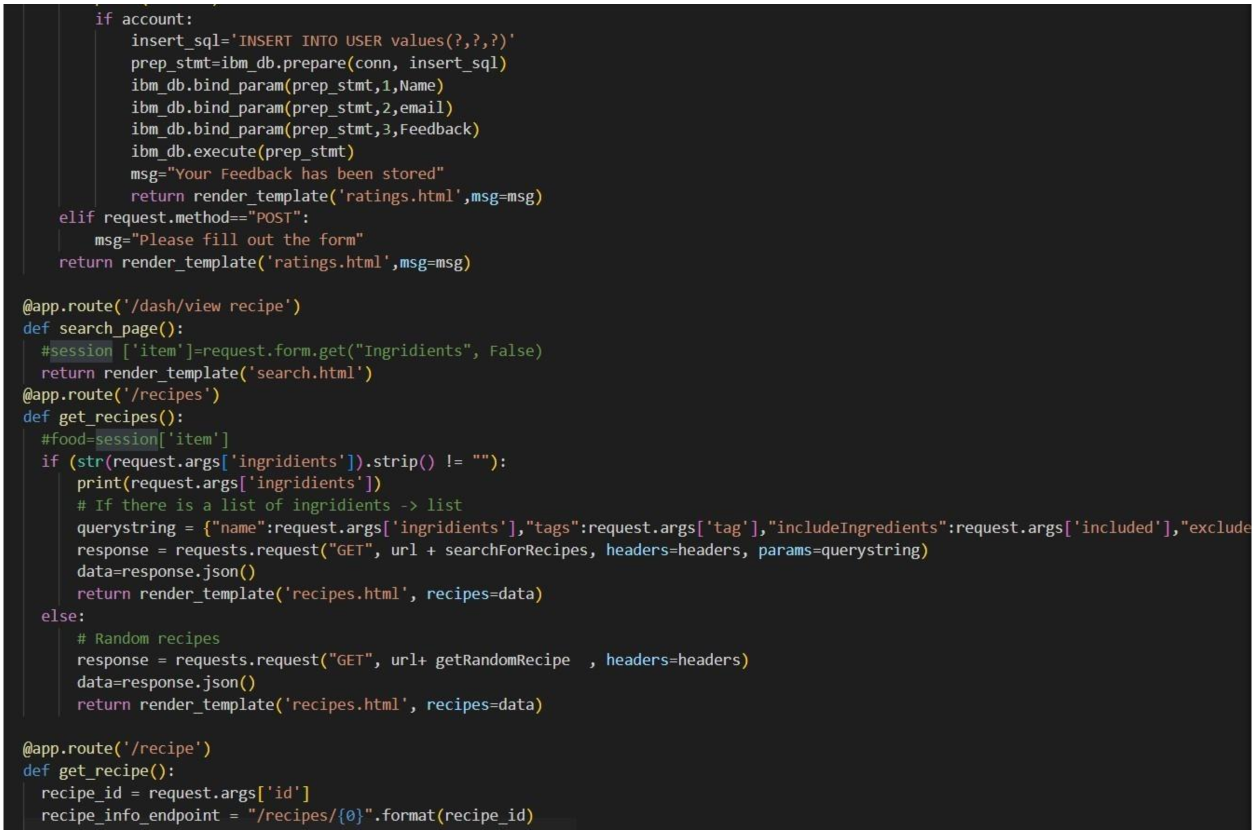
```
            return render_template('dash.html')
        else:
            msg='Incorrect username/password'
    return render_template('login.html',msg=msg)

@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)
```



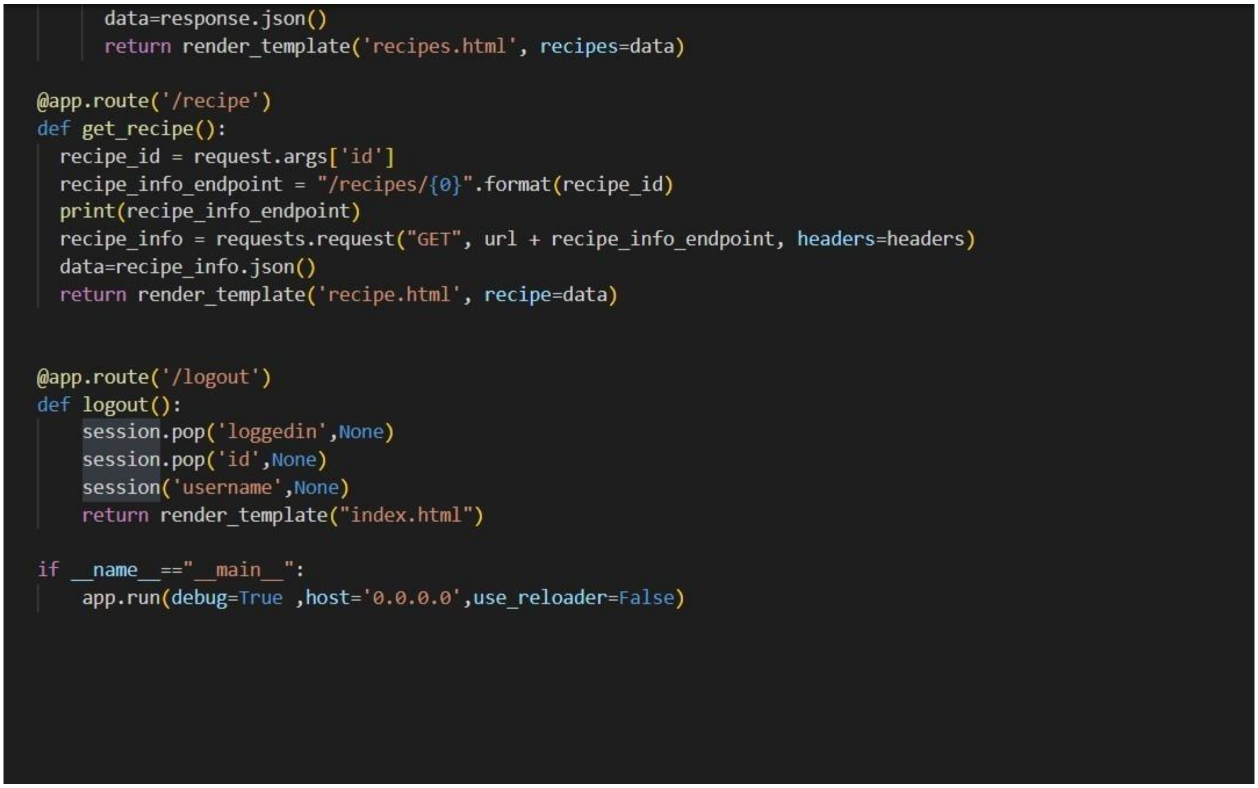
```
        chgpwd_sql='UPDATE USER SET password = ? WHERE username = ?'
        prep_stmt=ibm_db.prepare(conn, chgpwd_sql)
        ibm_db.bind_param(prep_stmt,1,newpassword)
        ibm_db.bind_param(prep_stmt,2,username1)
        ibm_db.execute(prep_stmt)
        msg="You have successfully changed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

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



```
            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)
```

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

3) Project Demo Link

https://drive.google.com/file/d/1KwGtsi6pW8tJZ7LO6HmtzIxHNqbm5ZGB/view?usp=share_link