NUTRITION ASSISTANCE APPLICATION

PROJECT REPORT

Submitted by

Team ID: PNT2022TMID07098

SAFEENA THALATH (130719104065)
SIVAMALINI D (130719104076)

SUBHIKSHA I (130719104082)

SIVASANKARI R(130719104079)

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE ENGINEERING

JERUSALEM COLLEGE OF ENGINEERING

(An Autonomous Institution, Affiliated to Anna University, Chennai)

NBA & NAAC (A GRADE) ACCREDITED INSTITUTION Velachery main road, Narayanapuram, Pallikaranai, Chennai - 600100

1. INTRODUCTION

a. Project Overview

Nutrify app is a type of nutrition tracking app that helps users lose weight, be healthy, and get stronger. There are different nutrition apps, including a calorie counter, diet trackers, nutrition planner apps, and marketplace platforms that connect users and nutrition coaches.

Apart from this, there are also apps designed for niches: app-based food diaries, pregnancy nutrition apps, bodybuilding nutrition applications, vegan nutrition apps, diet-tracking apps, health activity tracker apps, etc.

b. Purpose

To keep track of daily intake. To monitor calories intake and consumed. To provide you with guidance on healthy and nutritious food. To create a personalized meal plan.

One of the most basic functions of nutrify app is to guide its users towards a healthy diet and assist them to achieve their health goals. So, once your user specifies the goal like desired weight goal, body type, food habits, and preferred food items, your app must suggest them with a proper diet accordingly.

2. LITERATURE SURVEY

2.1Existing problem

With diet and nutrition analyzer apps, you can analyze your client’s current goal of diet and provide them with effective feedback on what to improve. It is possible to create personalized meal plans and healthy recipes and generate nutrient analysis reports by using barcodes and other relevant nutrition tracking tools compared to the users' nutritional requirements.

2.2References

1.UK (United Kingdom) Department of Health. Dietary reference values for food energy and nutrients in the United Kingdom (Report on health and social subjects; 41). London, UK: Her Majesty's Stationery Office; 1991. [March 4, 2018]. https://www.nutrition.org.uk/attachments/article/234/Nutrition%20Requirements_Revised%20Oct%2020

16.pdf.

2.“Leading health and fitness,” April 2017 [Online]. Available: https://www.statista.com/statstics/650748/health-fitness-app-usage-usa

3.Mukasine Angelique (2014). “Ontology-Based Personalized System to Support Patients at Home.” Research paper [Online]. Available:https://brage.bibsys.no/xmlui/bitstream/handle/11250/221227/IKT-590%20Spring%20Matster's%20thesis%20Angelique%20MUKASINE.pdf?sequence=1

4. Nutritionix API, June 2107 [Online]. Available :https://www.nutritionix.com/business/api

5. Run Android Application, April 2017 [Online] Available: https://developer.android.com/training/basics/firstapp/running-app.html

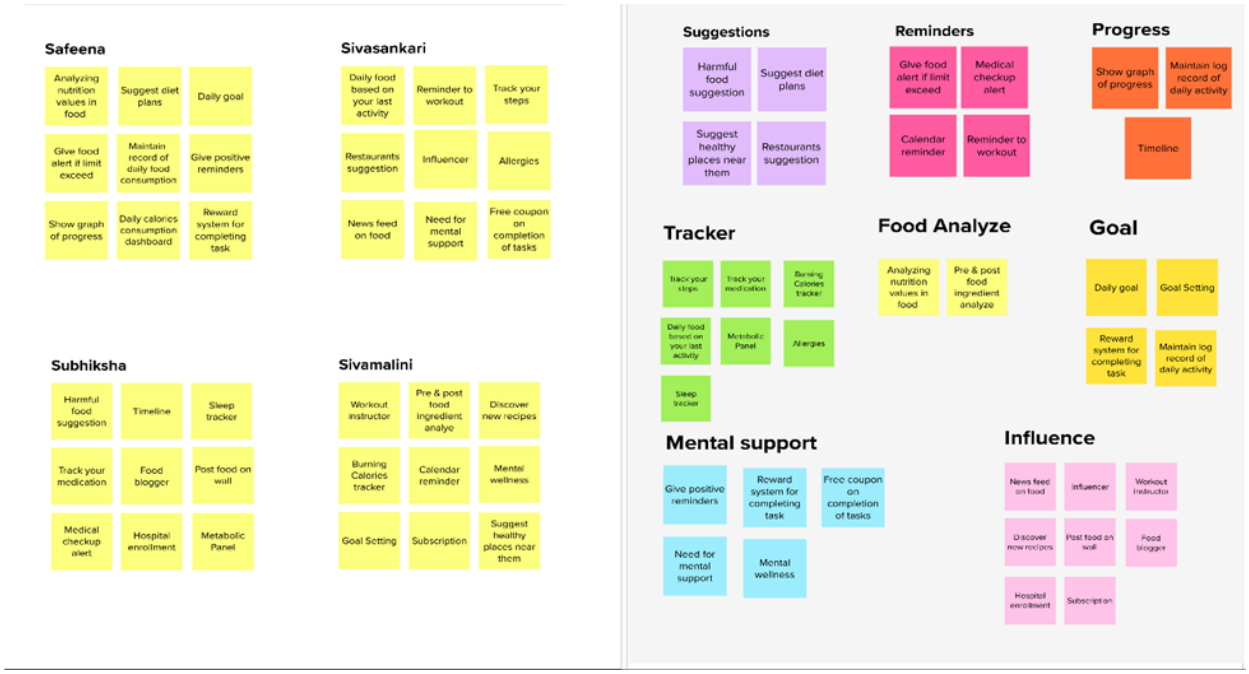
2.3 Problem Statement Definition

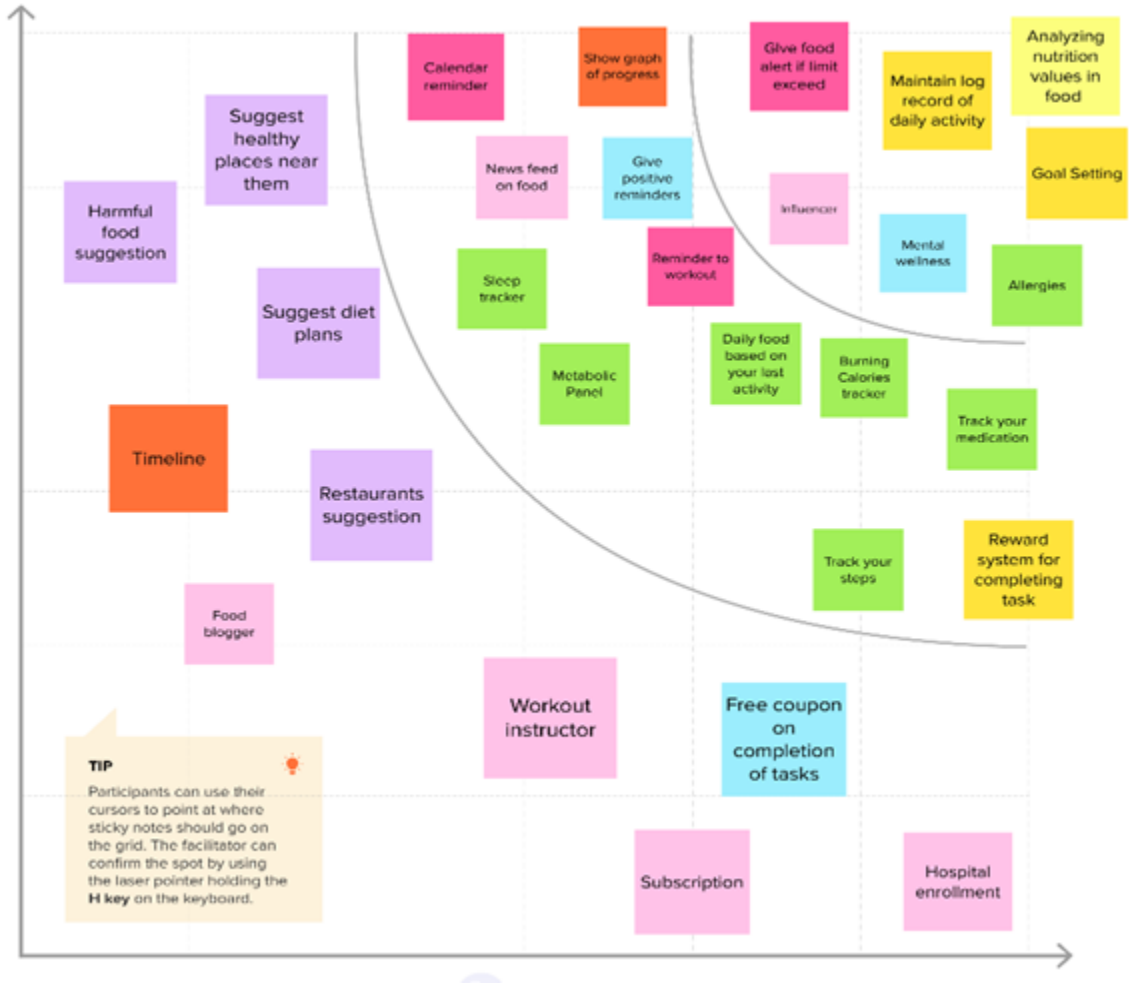
A program is needed to help users who want to manage their weight and calculate calories based on their daily activity and diet. Our project has the privilege of developing this program that can help the average person calculate their calorie intake and how much weight they are expected to lose or gain. People every day are concerned about their health. Often times, reading the nutritional label of the food being consumed isn’t enough to determine if a calorie balance goal is being met. It is also difficult to calculate what is expended based on the activities performed each day. Even now, there are those who are willing to record such information on paper to do the math themselves. This can prove to be quite trivial and tedious, especially in our modern technological world. Others will often be deterred from calculating calorie balance because of this as well.

3. IDEATION & PROPOSED SOLUTION

a. Empathy Map Canvas

b. **Ideation & Proposed solution**





c. Proposed Solution

People have a hard time maintaining their calorie intake and other nutrients which makes people ignore their healthy habits and cause obesity.

Create app-based nutrient dashboard systems that can analyze real-time images of a meal and analyze it for nutritional content.

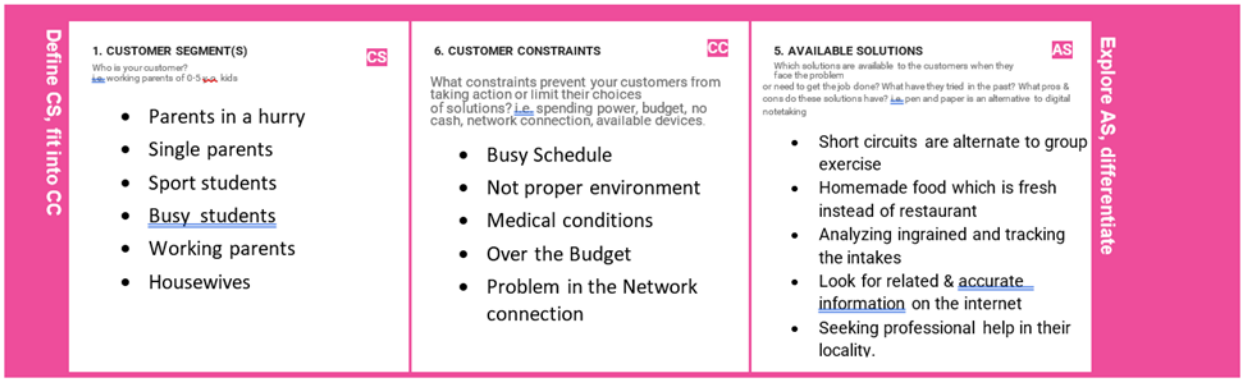
The proposed solution provides users a service that helps prepare a diet plan that fits their tight schedule and pre-plan their food schedule with food within their reach.

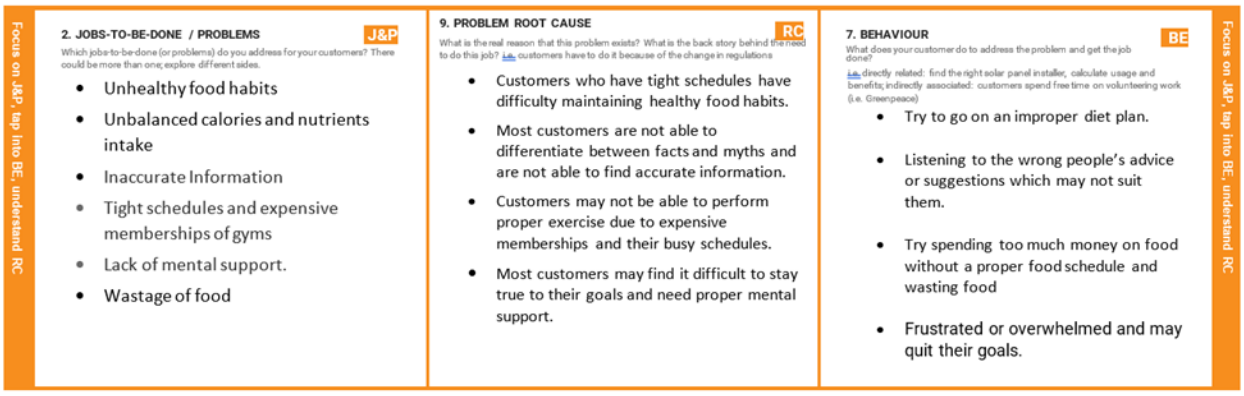
It helps people control their daily calorie intake by eating healthier foods which help to avoid obesity and obtain a healthy lifestyle.

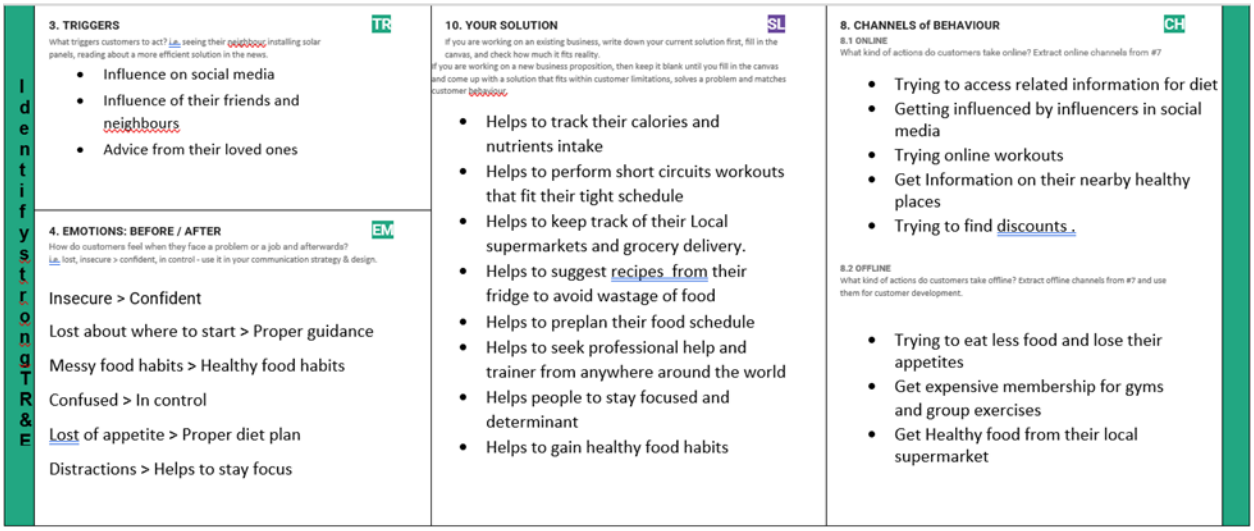
The project team shall fill in the following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	People have a hard time maintaining their calorie intake and other nutrients which makes people ignore their healthy habits and cause obesity.
2.	Idea / Solution description	Create app-based nutrient dashboard systems that can analyze real-time images of a meal and analyze it for nutritional content.
3.	Novelty / Uniqueness	The proposed solution provides users a service that helps prepare a diet plan that fits their tight schedule and pre-plan their food schedule with food within their reach.
4.	Social Impact / Customer Satisfaction	It helps people control their daily calorie intake by eating healthier foods which help to avoid obesity and obtain a healthy lifestyle.
5.	Business Model (Revenue Model)	Freemium business model
6.	Scalability of the Solution	For scalability, the cloud is used which can handle any number of users and provide on-demand service and it helps different types of customers.

d. Problem Solutionfit







4. REQUIREMENT ANALYSIS

a. **Functional requirement**

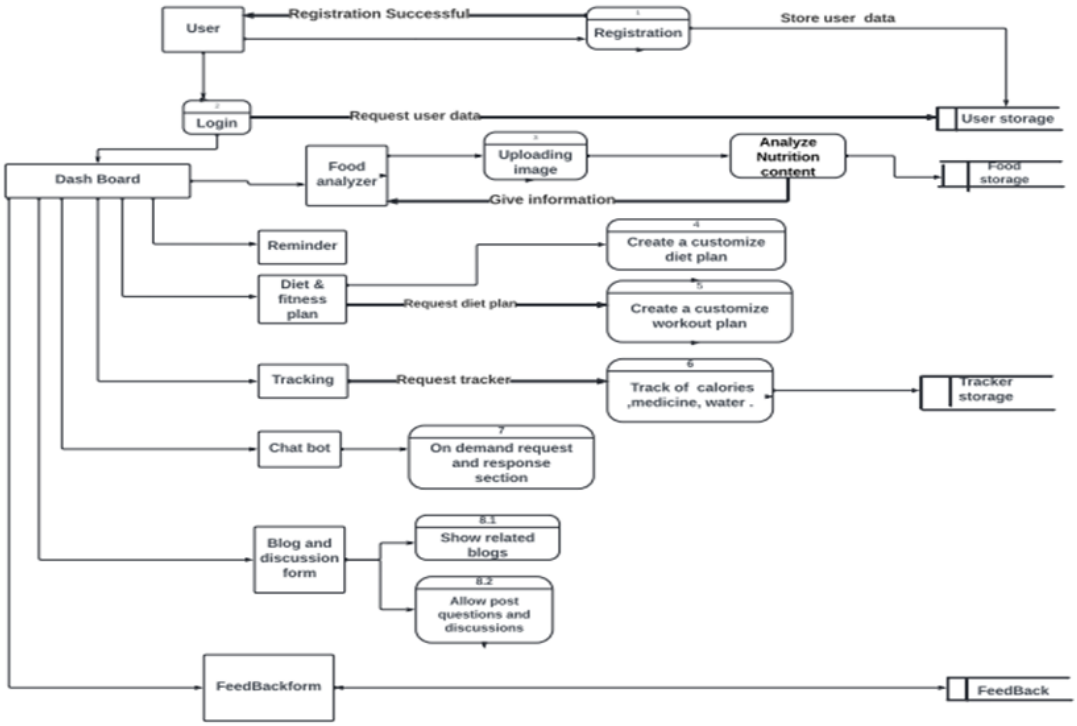
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User personal details	Enter personal details through a form
FR-4	Upload Food Image	Using Clarifai's AI food detection model analyses the food calories
FR-5	Alert Message	Remind it that specified Nutrition exceeds
FR-5	Diet plan & routine schedule	Customer choose their goal and create a diet plan for them

b. **Non-Functional requirements**

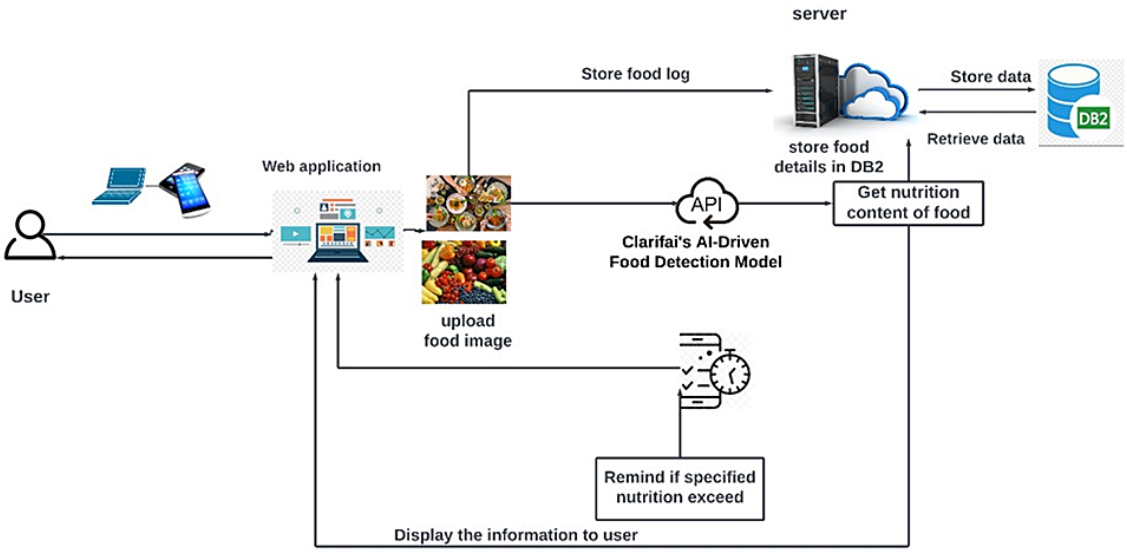
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Using a multi-method approach involving protocol analysis, interviews, and a system usability scale (SUS) was adopted
NFR-2	**Security**	The user's personal details are secured in the cloud.
NFR-3	**Reliability**	The application is verified through verified customers and only authenticate information will be displayed
NFR-4	**Performance**	A service that helps prepare a diet plan that fits their tight schedule and pre-plan their food schedule with food within their reach
NFR-5	**Availability**	The application service is available to all customers and provide support.
NFR-6	**Scalability**	The cloud is used which can handle any number of users and provide on-demand service and it helps different types of customers.

5. PROJECT DESIGN

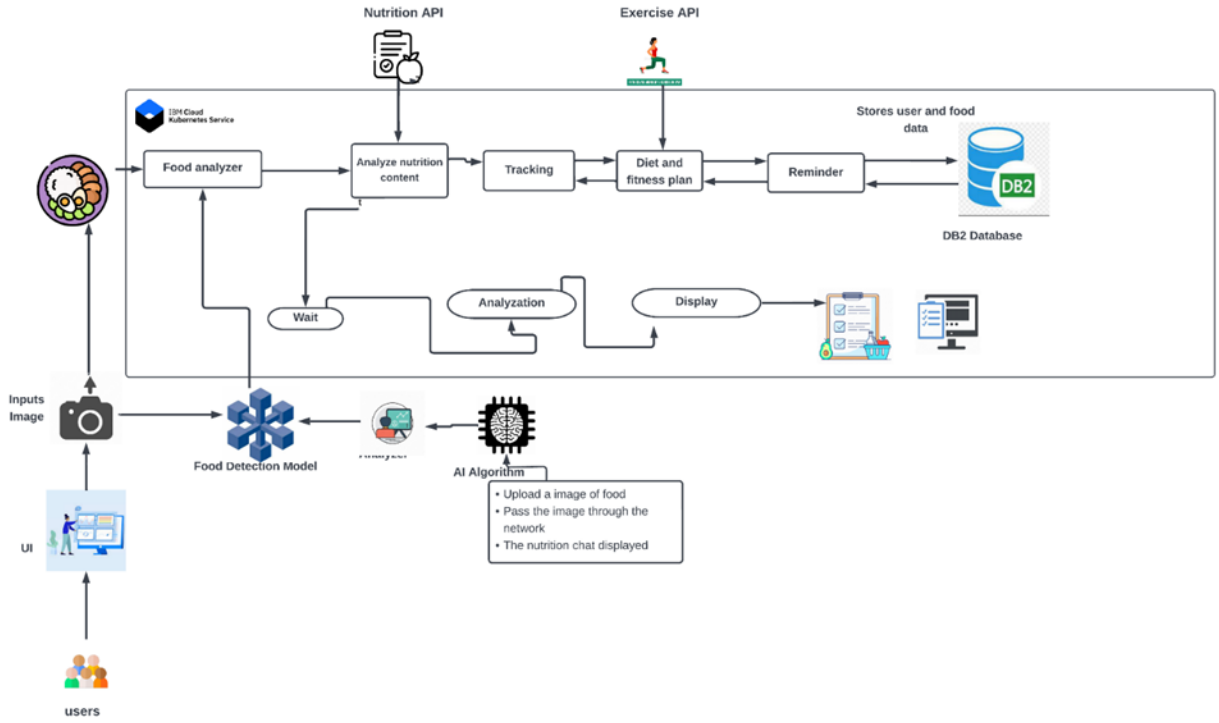
a. **DataFlowDiagram**



b. **Solution Architecture**



5.3 Technical architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email and confirming my password.	I can access my account/dashboard	High	Sprint-1
		USN-2	As a user, I will receive a confirmation message once I have registered for the application.	I can receive a confirmation message.	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail.	I can automatically move on to the dashboard	Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering my email & password.	I can access the dashboard	High	Sprint-1

		USN-5	As a user, I can reset my password if I have forgotten my password	I can reset my password again.	Medium	Sprint-1
	My Account	USN-6	As a user, I can view my personal information	I can view my details	High	Sprint-1
		USN-7	As a user, I can edit my profile photo	I can view my profile photo	Medium	Sprint-1
		USN-8	As a user, I can edit my email and it will updated.	I can change my email address	Low	Sprint-1
		USN-9	As a user, I can log out of the application from my account	I can exit my account	Medium	Sprint-1
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, and password, and confirming my password.	I can access my account/dashboard	High	Sprint-1
		USN-2	As a user, I will receive a confirmation message once I have registered for the application	I can receive a confirmation message.	High	Sprint-1
		USN-3	As a user, I can register for the application through Gmail.	I can automatically move on to the dashboard	Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering my email & password.	I can access the dashboard	High	Sprint-1
		USN-5	As a user, I can reset my password if I have forgotten my password	I can reset my password again	Medium	Sprint-1
	My Account	USN-6	As a user, I can view my personal information	I can view my details	High	Sprint-2
		USN-7	As a user, I can edit my profile photo	I can view my profile photo	Medium	Sprint-2
		USN-8	As a user, I can edit my email and it will updated.	I can change my email address	Low	Sprint-2
		USN-9	As a user, I can log out of the application from my account	I can exit my account	Medium	Sprint-1
Customer Care	Dashboard	USN-1	As a user, I want to upload an image to get a food	I can get my nutritious chart	High	Sprint-2

Executive			analyzing.			
		USN-2	As a user, I can enter the receipe name .	I can get my receipe details	High	Sprint-3
		USN-3	As a user, I can get list of receipes based on my preference.	I can get my receipe by my own preference.	High	Sprint-3
		USN-1	As a user ,I can upload images of the food.	I can click the upload button and upload image.	High	Sprint-4
		USN-2	As a user I can get the nutrition details of the receipe.	I can get my receipe data.	High	Sprint -4
Administrator		USN-3	As an admin, I have a feedback form from my customer	I can have my ups and down in my review session	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

a. Sprint Planning & Estimation

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Team members
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email and confirming my password.	High	Safeenathalath
		USN-2	As a user, I will receive a confirmation message once I have registered for the application	High	Sivasankari
		USN-3	As a user, I can register for the application through Gmail.	Medium	Safeenathalath
	Login	USN-4	As a user, I can log into the application by entering my email &	High	Sivamalini

			password.		
		USN-5	As a user, I can reset my password if I have forgotten my password	Medium	Subhiksha
	My Account	USN-6	As a user, I can view my personal information	High	Subhiksha
		USN-7	As a user, I can edit my profile photo	Medium	Sivamalini
		USN-8	As a user, I can edit my email and it will updated.	Low	Sivasankari
		USN-9	As a user, I can log out of the application from my account	Medium	Sivamalini
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, and password, and confirming my password.	High	Safeenathalath
		USN-2	As a user, I will receive a confirmation message once I have registered for the application	High	Sivasankari
		USN-3	As a user, I can register for the application through Gmail.	Medium	Safeenathalath
	Login	USN-4	As a user, I can log into the application by entering my email & password.	High	Sivamalini
		USN-5	As a user, I can reset my password if I have forgotten my password	Medium	Subhiksha
	My Account	USN-6	As a user, I can view my personal information	High	Subhiksha
		USN-7	As a user, I can edit my profile photo	Medium	Siva Malini

		USN-8	As a user, I can edit my email and it will updated.	Low	Sivasankari
		USN-9	As a user, I can log out of the application from my account	Medium	Sivamalini
Customer Care Executive	Dashboard	USN-1	As a user, I want to upload an image to get a food analyzing	High	Safeenathalth
		USN-2	As a user, I can enter the receipe name .	High	Safeenathalath
		USN-3	As a user, I can get list of receipes based on my preference.	High	Sivasankari
		USN-1	As a user ,I can upload images of the food.	Medium	Subhiksha
Administrator		USN-2	As an admin, I have a feedback form from my customer	High	Sivasankari
		USN-2	As a user I can get the nutrition details of the receipe.	Medium	Subhiksha

b. **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	29 Oct 2022
Sprint-2	34	6 Days	31 Oct 2022	05 Nov 2022	30	05 Nov 2022
Sprint-3	60	6 Days	07 Nov 2022	12 Nov 2022	60	12 Nov 2022
Sprint-4	40	6 Days	14 Nov 2022	19 Nov 2022	40	19 Nov 2022

7. CODING & SOLUTIONING

a. Food Analyzer (Feature 1)

Food analyzer is an important feature that help people understand their daily nutritents intake .It is impotant to a healthy lifestyle. This feature used to analyze the food items in the picture using clarify food detection model and on entering the items and hitting the submit button it will call the CalorieNinjas API to calculate the calories and other micronutrients. The food detection model can now automatically recognize more than a thousand different foods in images all the way down to the ingredient level. The names are stored in a list and sent requests to the respective API endpoints.

The CaloriesNinja API will Natural language API to extract nutrition data from any text.This API extract nutrition data from any text or image containing text (such as menus and recipes) with natural language engine and database of over 100,000 foods and drinks This help us to understand different micronutrients present in the different items.

b. View Recipes (Feature 2)

Cooking is a passionate job that allows people to turn an ordinary meal into a tempting and mesmerizing one and there are a number of people who take up this art as a hobby or make it their profession. Especially during the initial Covid19 phase when the entire world was locked up in their homes, the one thing they genuinely bonded over was cooking and this is why for the first two months there were a plethora was food pictures being shared by people who loved to spend the lockdown exploring their culinary skills.

This application presents a variety of different recipes according to user preferences using Low Carb Recipes API which is a highly flexible search over thousands of low-carb/keto recipes with rich nutrient information. This API also provides calories of nutrients of each recipe so that user can have a better understanding of their nutrients intakes

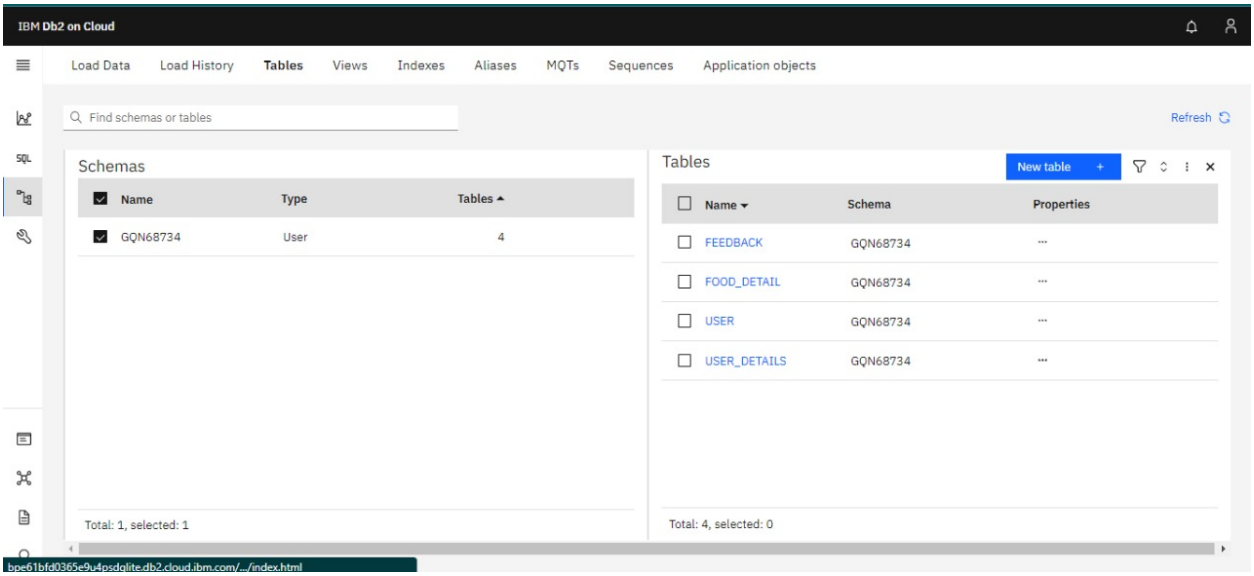
c. Feedback Form (Feature 3)

The feedback form is important to understand users so user-based feedback can improve the development process, allows to create better applications and help you grow. In addition to helping you determine your priorities, user-based feedback can help you with design and thus

improve the final product. Improving the usability of your software is essential to long-term success, and it should never be an afterthought. The feedback is collected through a form which is stored in the database

d. Database Schema

The database used for this application is IBM cloud where the necessary information is stored in the database.This The database schema here is GQN68734 which it consists of 4 tables namely USER, where the registered details are stored and USER_DETAILS where the user's personal information is stored and thirdly we have food details where the details about the food are stored and finally have feedback table where the feedback of the user is stored so administer can view the feedback and make the necessary improvements in the application.



8. ADVANTAGES & DISADVANTAGES

Advantages:

1. Interface is extremely friendly and easy to use. – Helps count calories while tracking weight loss.
2. Analyzes a person's age and weight with how they want to improve their health.
3. You can upload food pictures and type food without having to type in the nutritional values.
4. Provides a daily report of caloric intake.
5. Has a feature where users can calculate their food nutrients by taking a picture of their dish.

Disadvantages:

1. You have to manually input your daily food intake once a weight loss plan has been created

2. It doesn’t track your intake of minerals and vitamins.

3. Food database is not that large. It is missing a lot of the popular food brands, which means users have to manually type the nutritional values in.”

9. CONCLUSION

In the journey of developing this nutrify application, I got an opportunity to learn the detail process of developing Android applications with the help of material design concepts to built beautiful yet elegant user interface. In this process I also able to learn how to develop and use a RESTful API from scratch with the help of Python. Moreover, I even get a chance to learn how to use different Android libraries such as MPAndroidChart, Zxing, FitChart and so on. I also learn about the various food ontology APIs on how to use them and even by developing the client-server application helped me to understand the communication calls between them. FUTURE SCOPE

In the future scope, the application can enhance its functionality by adding image recognization which can be used to analyze the food image and produce the result with the nutrition values contained in that particular food item. A Google map can be added to track the distance covered by the user using the Activity Tracker to provide a more visual representation of the activity to the user. The activity tracker can also be updated using the Google fit API for the more accurate result.

10. APPENDIX

a. Source Code:

The following code is the uses 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
import json
app=Flask(__name__,template_folder='templates',static_folder='static')
app.secret_key='a'
conn = ibm_db.connect("Database=bludb;Hostname=1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;Port=32286;Security=SSL;SSLServe
```

```
rCertificate=DigiCertGlobalRootCA.crt;UID=gqn68734;PWD=IJvrQIkrmldUdQzP","")
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 redirect(url_for('dashboard'))
        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.get('username',False)
        email=request.form.get('email',False)
        password=request.form.get('password',False)
        Firstname=request.form.get('firstname',False)
        lastname=request.form.get('lastname',False)
        #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)
        session['mail']=email
        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('register.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=session['mail']
        server=smtplib.SMTP('smtp.gmail.com',587)
        server.starttls()
        password="nsgeuedwbzptosyp"
        server.login("active.1468@gmail.com",password)
        otp="".join([str(random.randint(0,9))for i in range(4)])
        msg=' YOUR OTP IS'+str(otp)
        server.sendmail("active.1468@gmail.com",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)
    ibm_db.execute(stmt)
    account=ibm_db.fetch_assoc(stmt)
    print(account)
    if account:
        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/"
getRandomRecipe="/random"

@app.route('/dash')
def dashboard():
```

```
    return render_template('dash.html')

@app.route('/dash/viewprofile')
def viewprofile():
    username=session['id']
    sql='SELECT * FROM USER_DETAILS WHERE name=?'
    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',account=account)
    else:
        return render_template('peronal info.html')


@app.route('/personinfo',methods=['GET','POST'])
def per_info():
    msg=''
    if request.method =='POST':
        Name=request.form.get('name',False)
        gender=request.form.get('gender',False)
        tar_weight=request.form.get('Target Weight',False)
        Age=request.form.get('Age',False)
        Height=request.form.get('Height',False)
        Weight=request.form.get('Weight',False)
        health=request.form.get('Health',False)
        location=request.form.get('Location',False)
        phoneno=request.form.get('Phone Number',False)

        insert_sql='INSERT INTO USER_DETAILS values(?,?,?,?,?,?,?,?,?)'
        prep_stmt=ibm_db.prepare(conn, insert_sql)
```

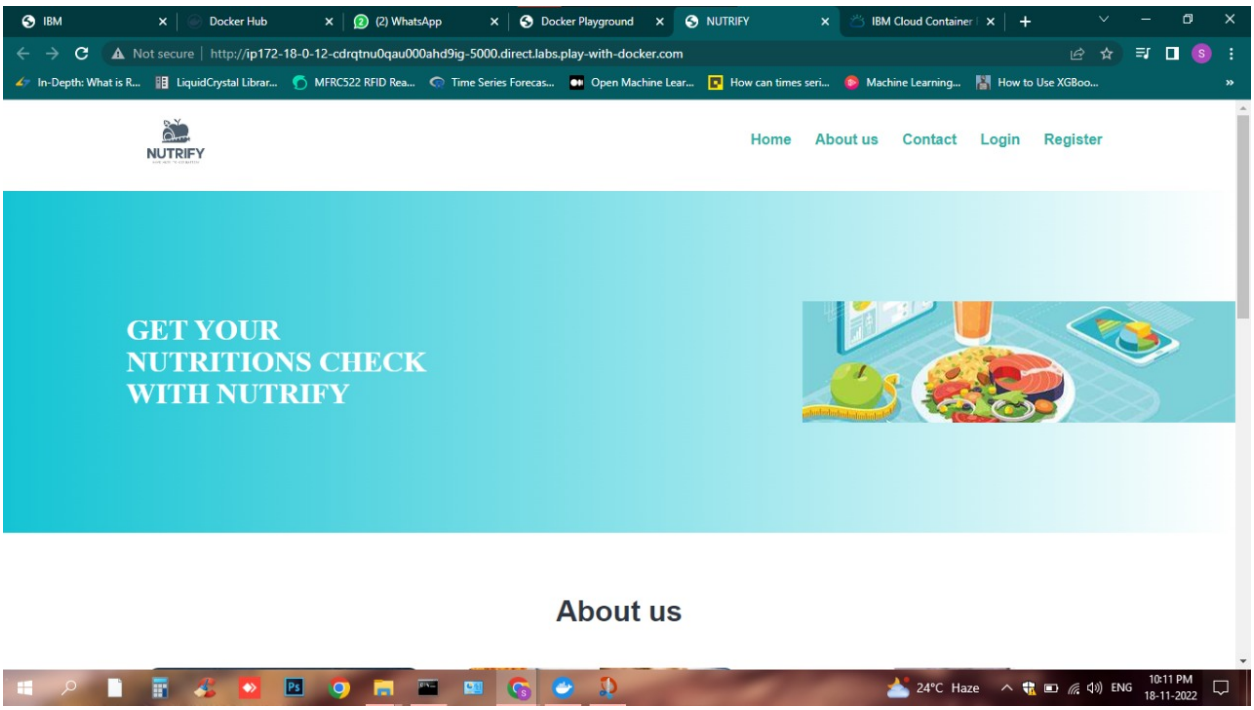
```
        ibm_db.bind_param(prep_stmt,1,Name)
        ibm_db.bind_param(prep_stmt,2,gender)
        ibm_db.bind_param(prep_stmt,3,Age)
        ibm_db.bind_param(prep_stmt,4,Height)
        ibm_db.bind_param(prep_stmt,5,Weight)
        ibm_db.bind_param(prep_stmt,6,tar_weight)
        ibm_db.bind_param(prep_stmt,7,health)
        ibm_db.bind_param(prep_stmt,8,location)
        ibm_db.bind_param(prep_stmt,9,phoneno)
        ibm_db.execute(prep_stmt)
        msg="Your details are successfully saved"
        return redirect(url_for('viewprofile',msg=msg))
    else:
        msg="Please fill out the form"
    return render_template('peronal info.html',msg=msg)


@app.route('/feedback',methods=['GET','POST'])
def feedback():
    msg=''
    if request.method =='POST':
        Name=request.form['name']
        email=request.form['email']
        Feedback=request.form['feedback']
        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 FEEDBACK values(?,?,?)'
            prep_stmt=ibm_db.prepare(conn, insert_sql)
```

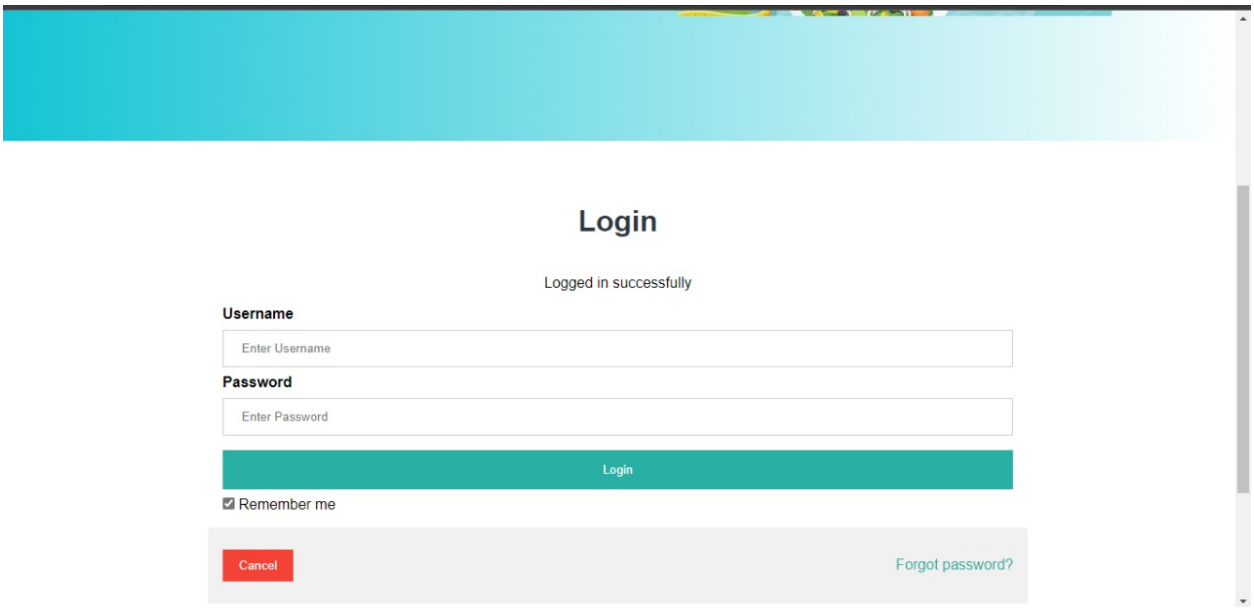
```
            ibm_db.bind_param(prep_stmt,1,Name)
            ibm_db.bind_param(prep_stmt,2,email)
            ibm_db.bind_param(prep_stmt,3,Feedback)
            ibm_db.execute(prep_stmt)
            msg="Your Feedback has been stored"
            return render_template('ratings.html',msg=msg)
    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'],"
excludeIngredients":request.args['excluded'],"maxPrepareTime":"30","maxCookTime":"20","maxCalories
":"500","maxNetCarbs":"5","maxSugar":"3","maxAddedSugar":"0","limit":"10"}
      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)
```

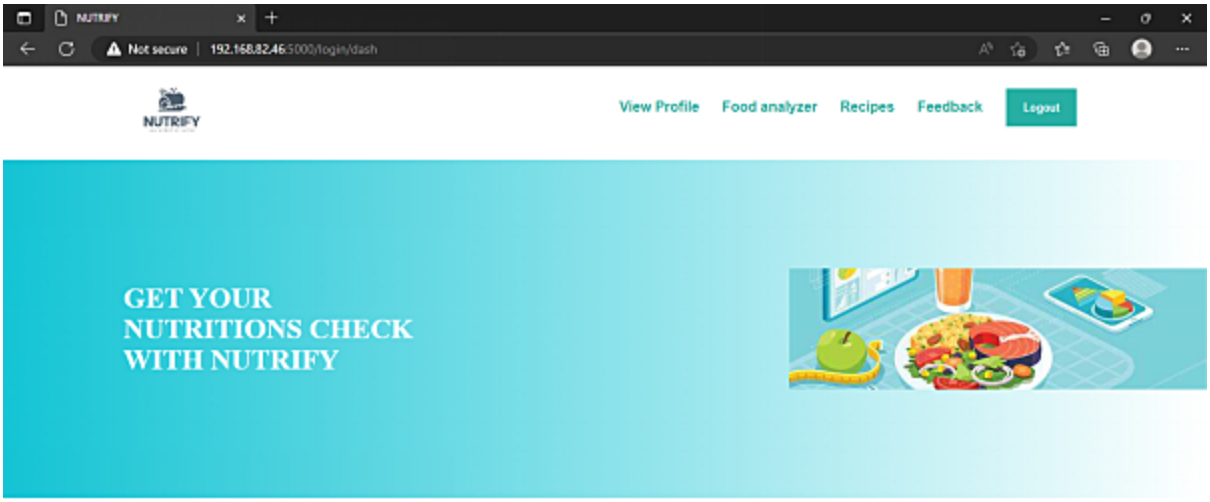
```
@app.route('/recipe')
def get_recipe():
  recipe_id = request.args['id']
  recipe_info_endpoint = "/recipes/{0}".format(recipe_id)
  print(recipe_info_endpoint)
  recipe_info = requests.request("GET", url + recipe_info_endpoint, headers=headers)
  data=recipe_info.json()
  return render_template('recipe.html',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

@app.route('/food', methods=['POST','GET'])
def get_nutri():
    msg=''
    url = "https://nutrition-by-api-ninjas.p.rapidapi.com/v1/nutrition"

    headers = {
        	"X-RapidAPI-Key": "ad933ea36amsh6b0a83e514b1a58p14bc9ejsne745a5851a1b",
        	"X-RapidAPI-Host": "nutrition-by-api-ninjas.p.rapidapi.com"
    }
    if  request.method =='POST':
        food=request.form.get('food')
        if (str(food).strip() != ""):

            querystring = {"query":food}
            response = requests.request("GET", url, headers=headers, params=querystring)
            data=response.json()
        #data=json.load(response)
        #datas=data.json()
            print(data)
            return render_template("foodanalyzer.html", result=data)
    return render_template("foodanalyzer.html")


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

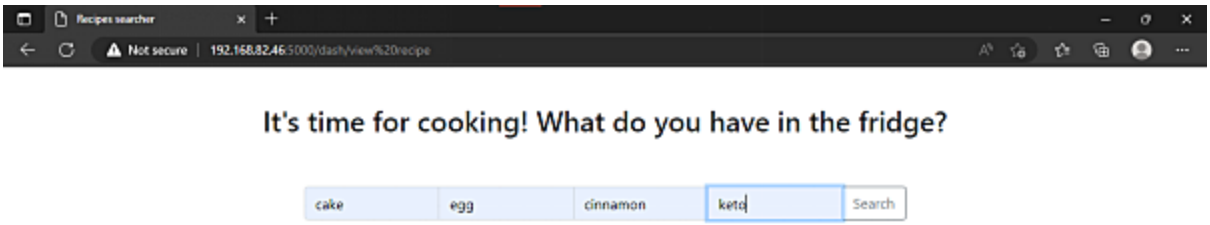
```
def logout():
    session.pop('loggedin',None)
    session.pop('id',None)
    return render_template("index.html")

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

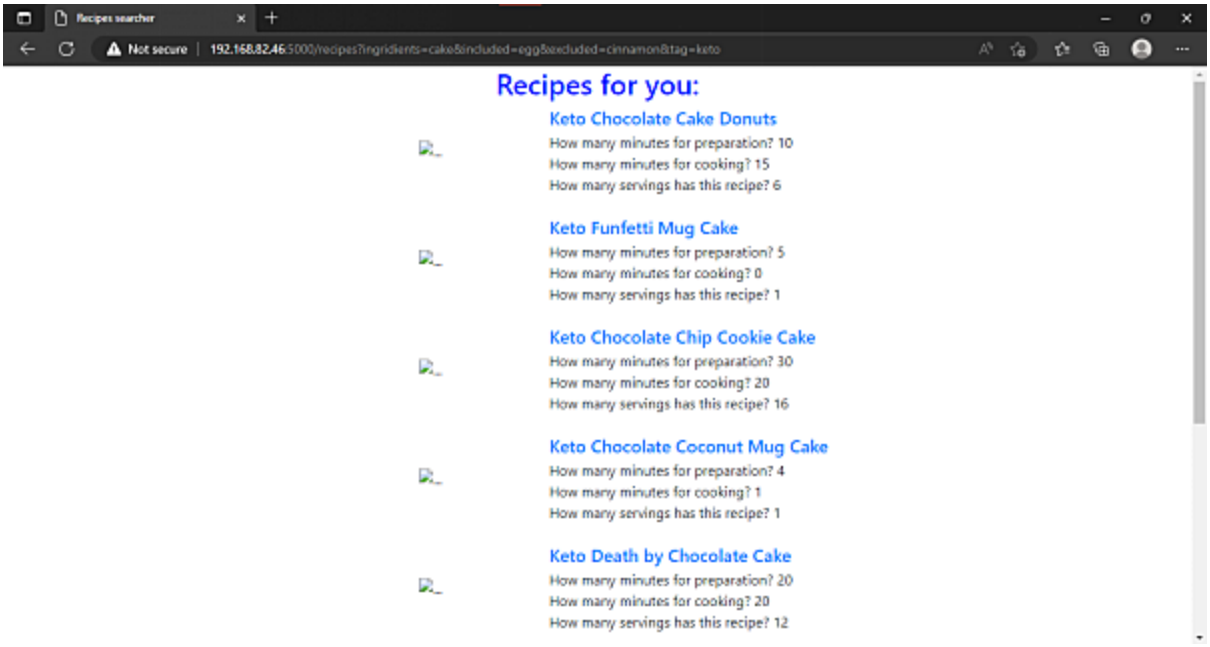
OUTPUT:

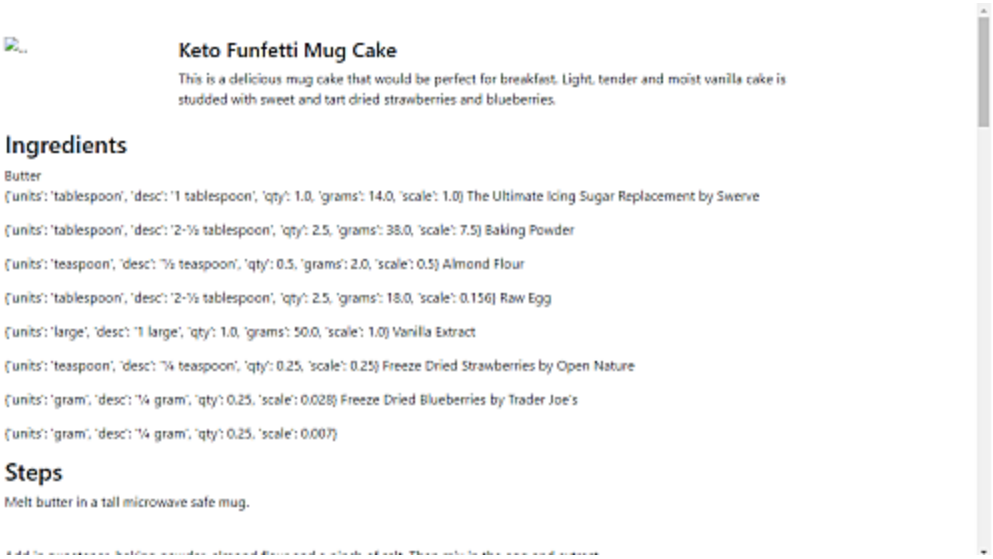


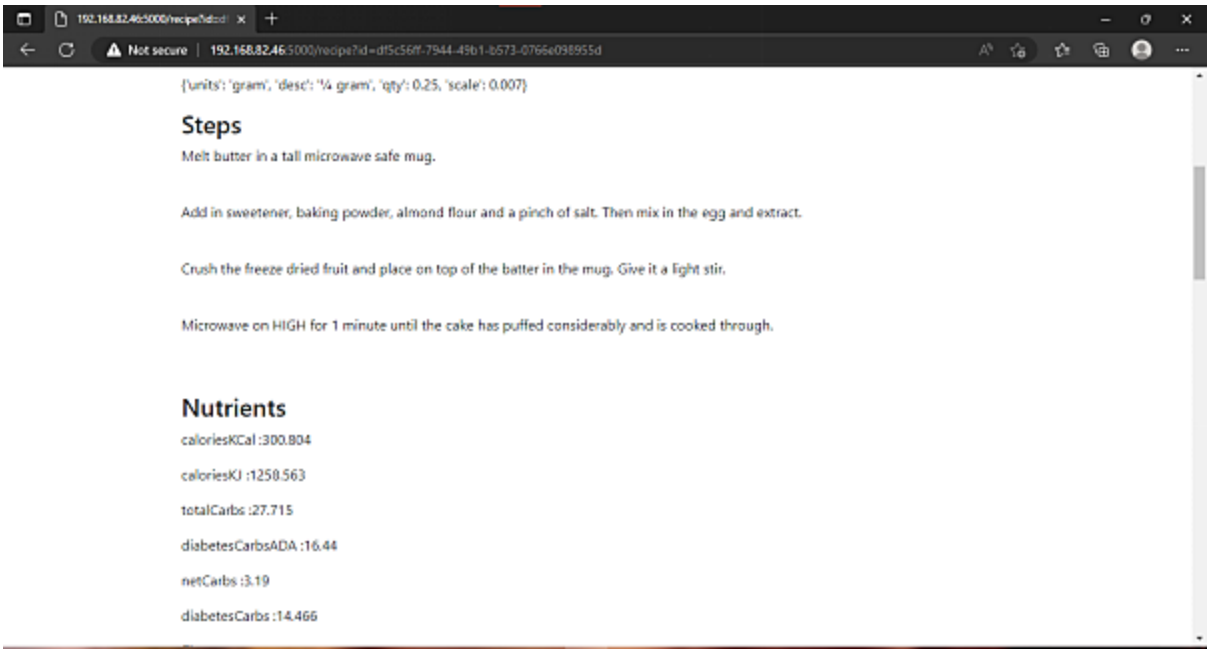

Login
Logged in successfully
Username
Enter Username
Password
Enter Password
Login
Remember me
Cancel
Forgot password?

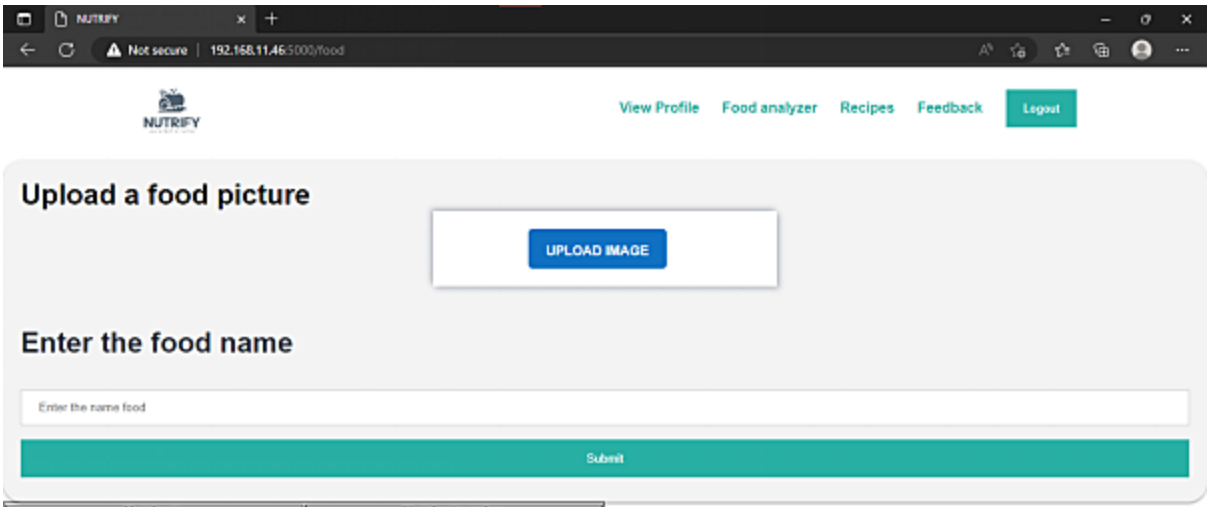

NUTRIFY
Not secure | 192.168.82.46:5000/login/dash
NUTRIFY
View Profile
Food analyzer
Recipes
Feedback
Logout
GET YOUR
NUTRITIONS CHECK
WITH NUTRIFY


Recipes searcher
Not secure | 192.168.82.46:5000/dash/view%20recipe
It's time for cooking! What do you have in the fridge?
cake
egg
cinnamon
keto
Search

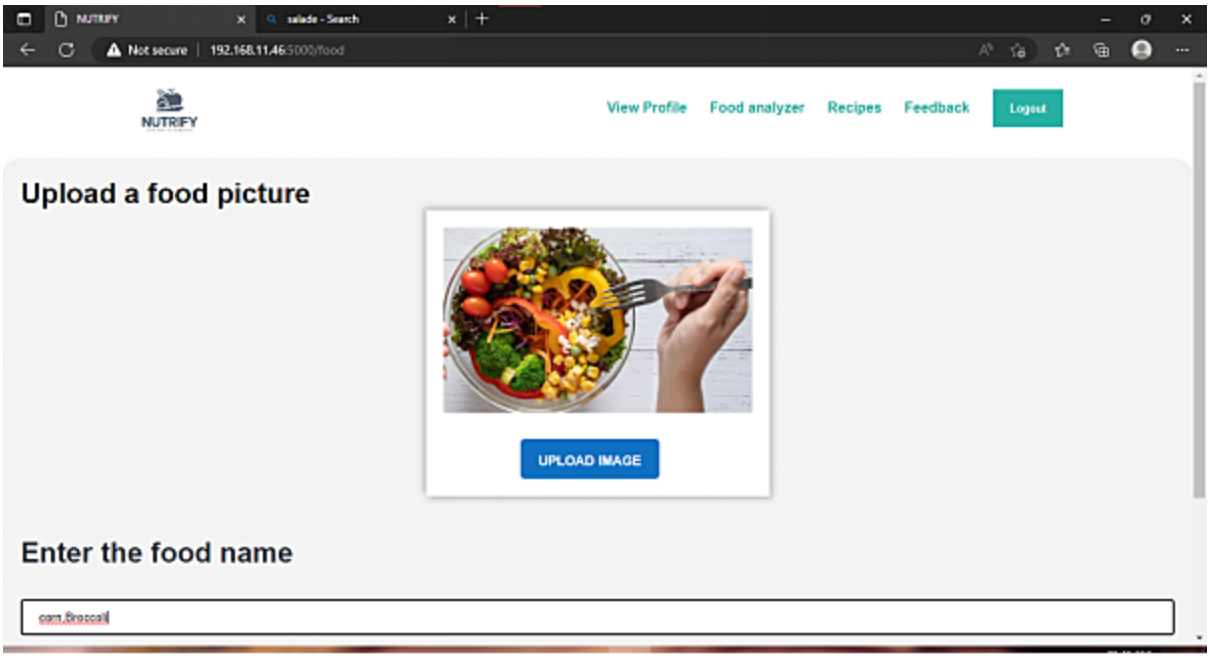

Recipes searcher
Not secure | 192.168.82.46:5000/recipes?ingridients=cake&included=egg&excluded=cinnamon&tag=keto
Recipes for you:
Keto Chocolate Cake Donuts
How many minutes for preparation? 10
How many minutes for cooking? 15
How many servings has this recipe? 6
Keto Funfetti Mug Cake
How many minutes for preparation? 5
How many minutes for cooking? 0
How many servings has this recipe? 1
Keto Chocolate Chip Cookie Cake
How many minutes for preparation? 30
How many minutes for cooking? 20
How many servings has this recipe? 16
Keto Chocolate Coconut Mug Cake
How many minutes for preparation? 4
How many minutes for cooking? 1
How many servings has this recipe? 1
Keto Death by Chocolate Cake
How many minutes for preparation? 20
How many minutes for cooking? 20
How many servings has this recipe? 12

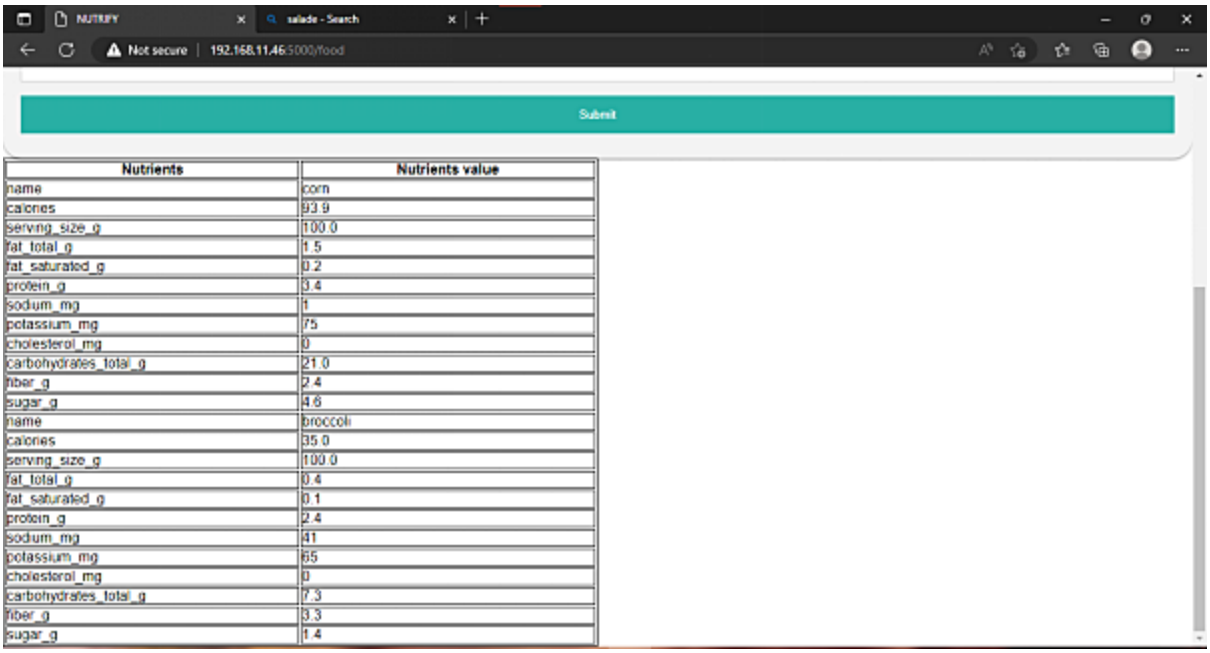

Keto Funfetti Mug Cake
This is a delicious mug cake that would be perfect for breakfast. Light, tender and moist vanilla cake is studded with sweet and tart dried strawberries and blueberries.
Ingredients
Butter
{'units': 'tablespoon', 'desc': '1 tablespoon', 'qty': 1.0, 'grams': 14.0, 'scale': 1.0} The Ultimate Icing Sugar Replacement by Swerve
{'units': 'tablespoon', 'desc': '2-⅓ tablespoon', 'qty': 2.5, 'grams': 38.0, 'scale': 7.5} Baking Powder
{'units': 'teaspoon', 'desc': '½ teaspoon', 'qty': 0.5, 'grams': 2.0, 'scale': 0.5} Almond Flour
{'units': 'tablespoon', 'desc': '2-⅓ tablespoon', 'qty': 2.5, 'grams': 18.0, 'scale': 0.156} Raw Egg
{'units': 'large', 'desc': '1 large', 'qty': 1.0, 'grams': 50.0, 'scale': 1.0} Vanilla Extract
{'units': 'teaspoon', 'desc': '¼ teaspoon', 'qty': 0.25, 'scale': 0.25} Freeze Dried Strawberries by Open Nature
{'units': 'gram', 'desc': '¼ gram', 'qty': 0.25, 'scale': 0.028} Freeze Dried Blueberries by Trader Joe's
{'units': 'gram', 'desc': '¼ gram', 'qty': 0.25, 'scale': 0.007}
Steps
Melt butter in a tall microwave safe mug.



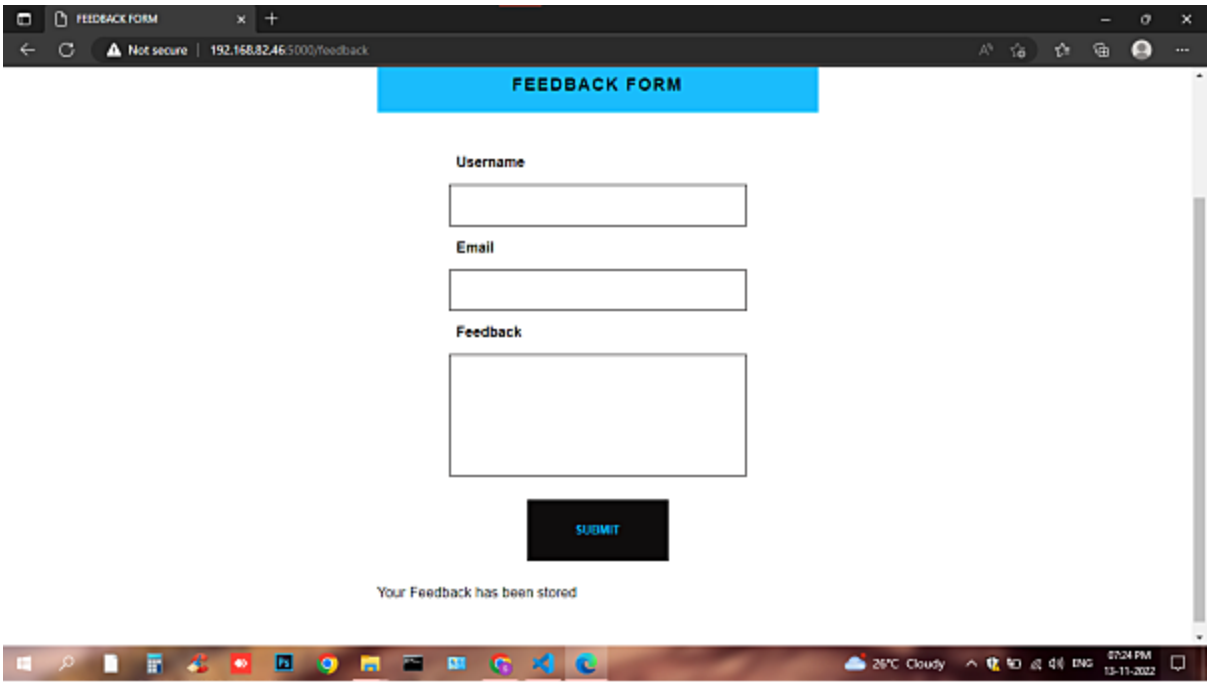


Nutrients	Nutrients value
	9
	7
	1
	0
	7





Nutrients	Nutrients value
name	corn
calories	93.9
serving_size_g	100.0
fat_total_g	1.5
fat_saturated_g	0.2
protein_g	3.4
sodium_mg	1
potassium_mg	75
cholesterol_mg	0
carbohydrates_total_g	21.0
fiber_g	2.4
sugar_g	4.6
name	broccoli
calories	35.0
serving_size_g	100.0
fat_total_g	0.4
fat_saturated_g	0.1
protein_g	2.4
sodium_mg	41
potassium_mg	65
cholesterol_mg	0
carbohydrates_total_g	7.3
fiber_g	3.3
sugar_g	1.4



11.2 GitHub & Project Demo Link

Github link: https://github.com/IBM-EPBL/IBM-Project-25896-1659976830

Project Demo Link:

https://drive.google.com/file/d/189t4pX3H5GAnTyZETJL19dCLdPMyiHbc/view?usp=drivesdk

Please copy and paste the link in browser to see the project demo video.