

Project Report

1. INTRODUCTION

- a. Project Overview
- b. Purpose

2. LITERATURE SURVEY

- a. Existing problem
- b. References
- c. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- a. Empathy Map Canvas
- b. Ideation & Brainstorming
- c. Proposed Solution
- d. Problem Solution fit

4. REQUIREMENT ANALYSIS

- a. Functional requirement
- b. Non-Functional requirements

5. PROJECT DESIGN

- a. Data Flow Diagrams
- b. Solution & Technical Architecture
- c. User Stories

6. PROJECT PLANNING & SCHEDULING

- a. Sprint Planning & Estimation
- b. Sprint Delivery Schedule

7. CODING & SOLUTIONING

- a. Feature 1
- b. Feature 2

8. TESTING

- a. Test Cases
- b. User Acceptance Testing

9. RESULTS

- a. Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

NUTRITION ASSISTANT

S.NO	REG.NO	NAME	DEPARTMENT	TEAM
1.	312419205030	HARI PRIYA S	IT	Team Lead
2.	312419205045	KEERTHANA R	IT	Team Member1
3.	312419205046	KEERTHANA T	IT	Team Member2
4.	312419205004	ABINAYA S	IT	Team Member3
5	312419205038	JANANI MARTINA G	IT	Team Member4

APPLICATION DONE BY TEAM ID: PNT2022TMID28383

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.

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>

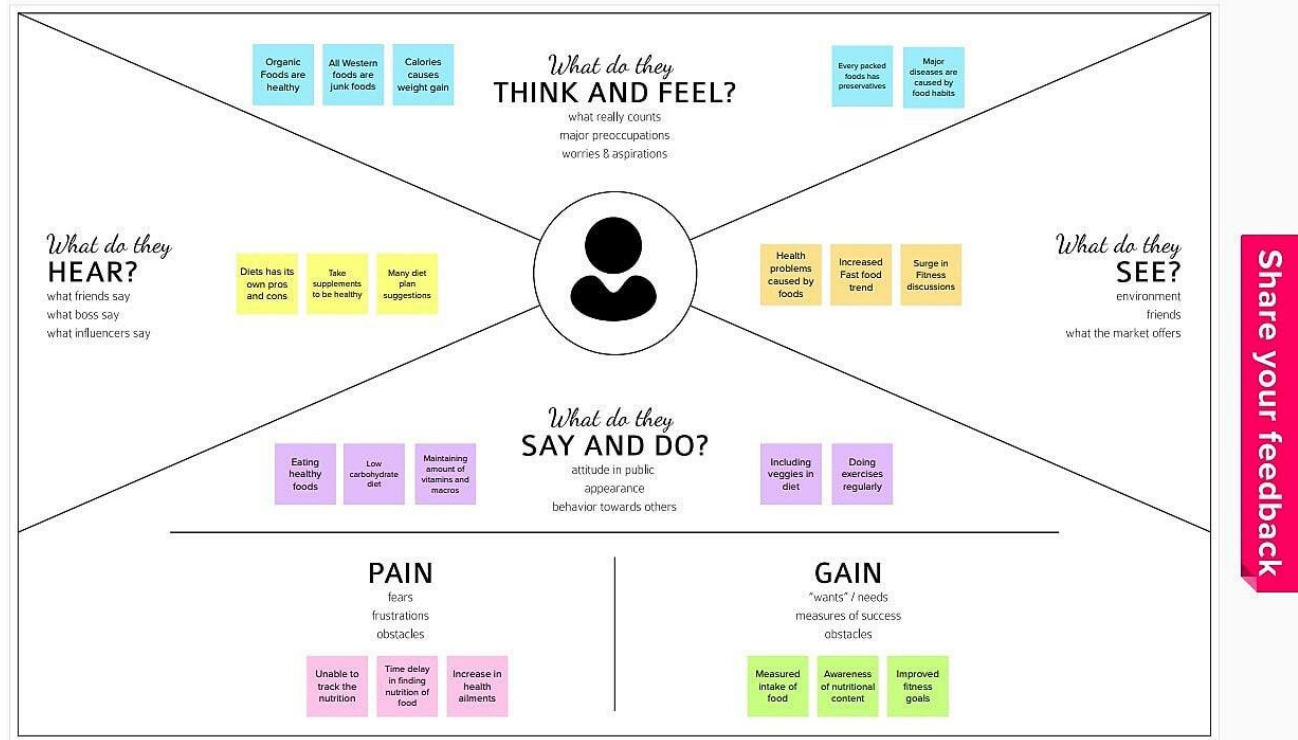
Kubernetes : https://www.youtube.com/embed/d6WC5n9G_sM

3) Problem Statement Definition

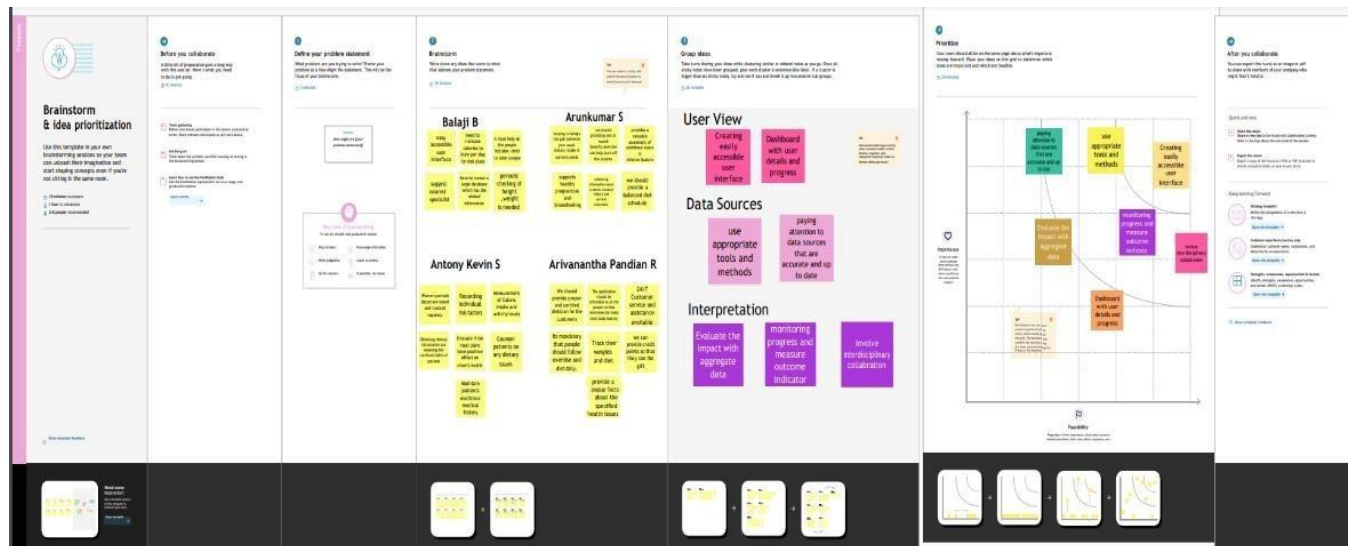
App-based nutrient dashboard systems which can analyze real-time images of a meal and analyze it for nutritional content which can be very handy and improves the dietary habits, and therefore, helps in maintaining a healthy lifestyle.

IDEATION & PROPOSED SOLUTION

1)Empathy Map Canvas



2) Ideation & Brainstorming









3) Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• The solution is a responsive Web application that can be used in any PC devices.• The website provides a user-friendly interface and accepts multiple samples predicting 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	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Getting feedback from the users for enhancement and giving notification on their diet plans and goal tracking.• Nutrition focused food banking & targeted in-depth reporting reviews that

		paid subscriptions the best.
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"> 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 targeted in-depth reporting reviews that paid subscriptions the best.
6.	Scalability of the Solution	<ul style="list-style-type: none"> 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) <small>Who is your customer? i.e. working parents of 0-5 y.o. kids</small>  <p>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.</p>	6. CUSTOMER CONSTRAINTS <small>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.</small>  <p>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.</p>	5. AVAILABLE SOLUTIONS <small>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</small>  <p>1. Personal diet tracking app which helps to maintain diet. 2. Personal nutritionist or trainer to suggest correct schedule according to customer requirement.</p>	Explore AS, differentiate

Focus on J&P, tap into BE, understand RC	2. JOBS-TO-BE-DONE / PROBLEMS <small>Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one - envision different sides</small>  <p>1. To calculate calories and nutrients present. 2. Monitor customers calorie consumption in order to maintain diet</p>	9. PROBLEM ROOT CAUSE <small>What is the real reason that this problem exists? What is the root cause?</small>  <p>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</p>	7. BEHAVIOUR <small>What does your customer do to address this problem and not the job, done?</small>  <p>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</p>	Focus on J&P, tap into BE, understand RC

<p>3. TRIGGERS What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.</p> <p>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</p>	<p>10. YOUR SOLUTION 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.</p> <p>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.,</p>	<p>8. CHANNELS of BEHAVIOUR 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7</p> <p>1. follow people who give healthy and nutritious food recipes. 2. Keep track of fitness freaks in social media and follow their fitness tips</p>
<p>4. EMOTIONS: BEFORE / AFTER 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.</p> <p>They scared of declining health, so they get motivated towards eating healthy foods and move to healthy lifestyle.</p>		<p>8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.</p> <p>Notice people around you who follows healthy habits in both consumption of food and workouts.</p>

REQUIREMENT ANALYSIS

1) Functional requirement

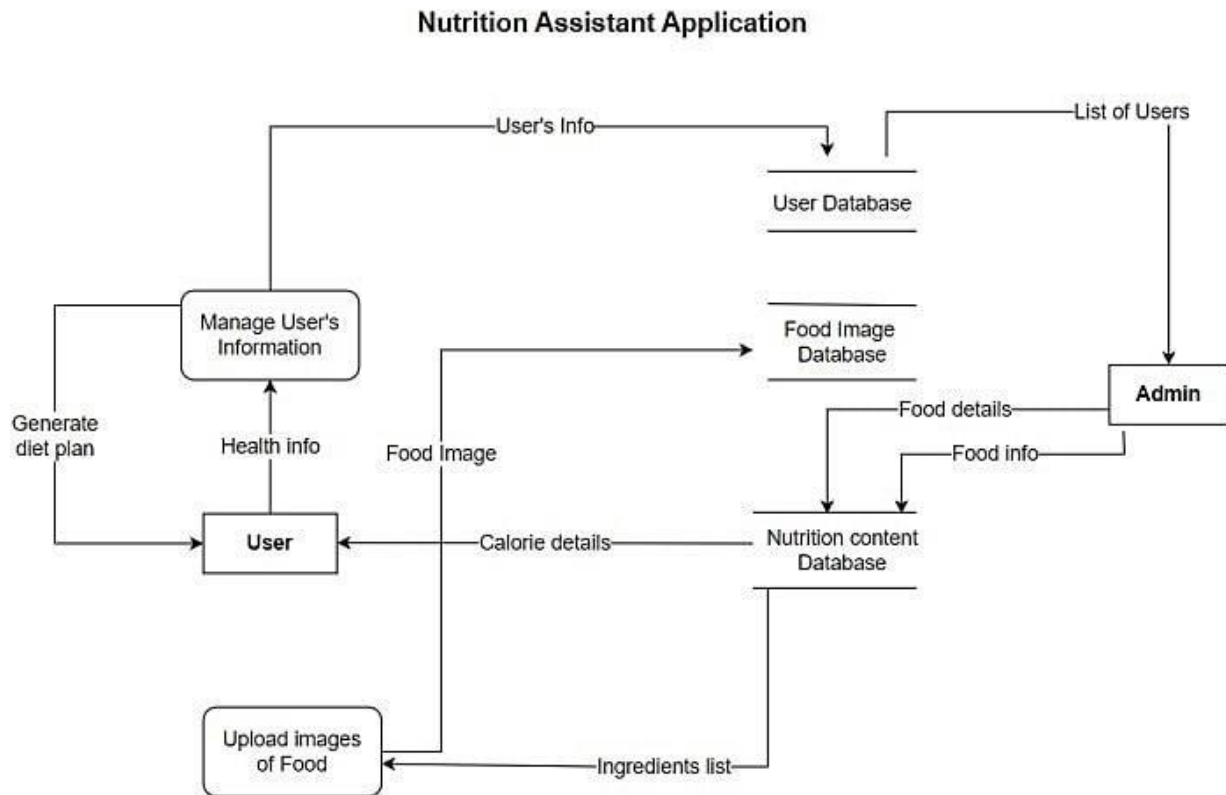
- User Registration
- User Confirmation
- Update Profile
- User Authentication
- Report

2) Non-Functional requirements

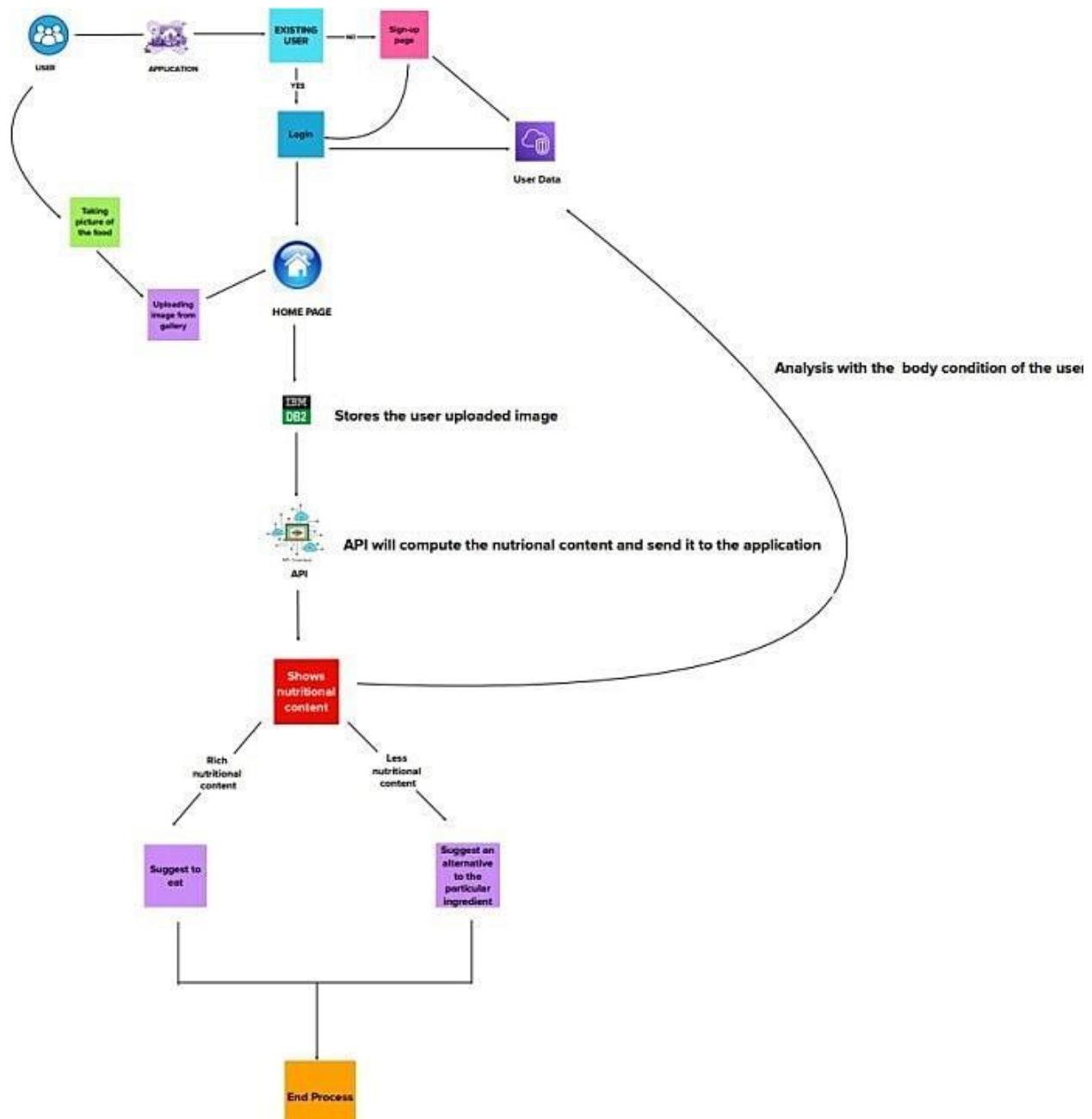
- Usability
- Security
- Reliability
- Performance
- Availability
- Scalability

PROJECT DESIGN

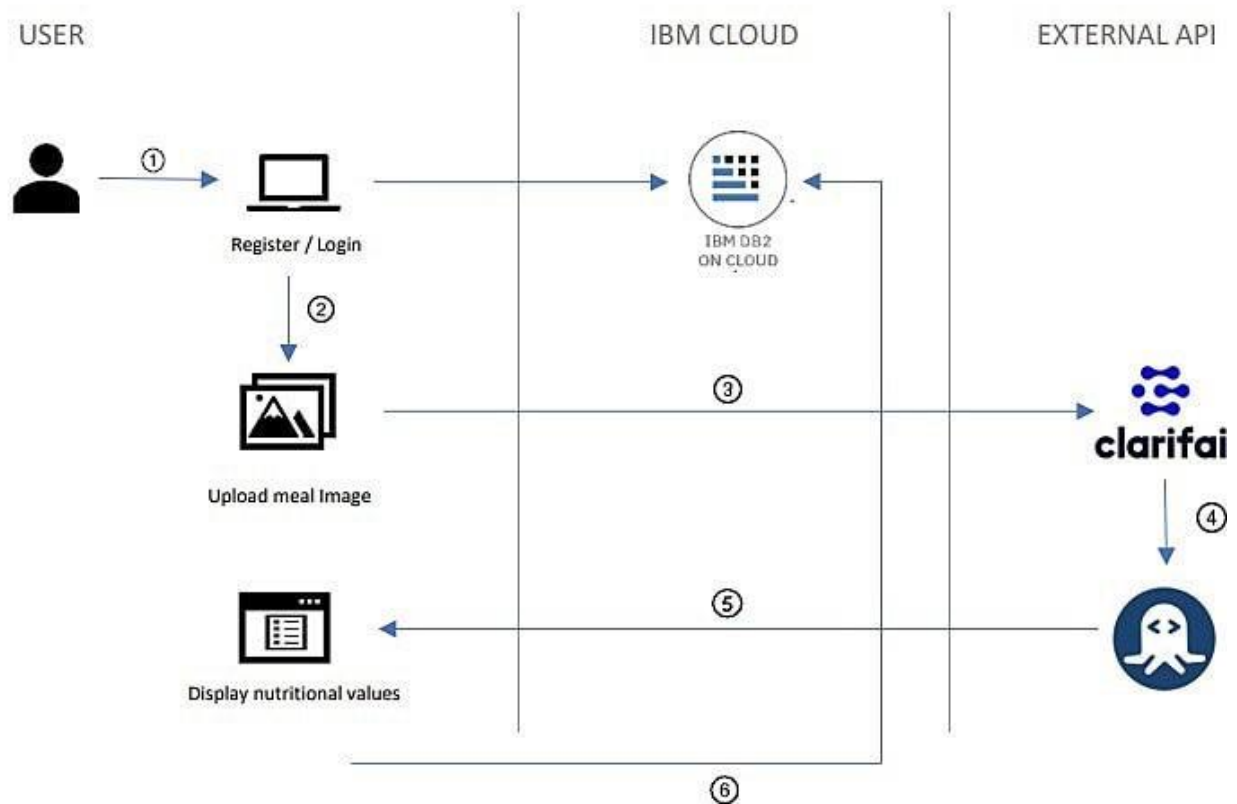
1) Data Flow Diagrams



2) Solution & Technical Architecture



Technical Architecture:



3) 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 and get the nutrition details and recipe for related scanned 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

FOOD SERVICES[Dashboard](#)[About Us](#)[Food Services](#)[Daily Tracker](#)[Personal Diary](#)[Logout](#)




Please Enter the image url



Please upload an image

Choose File | No file chosen



Please enter the food name

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

FOOD ITEM		Ingredients	
ESTIMATE	PERCENTAGE	Nutrition	Nutrition Value



FOOD SERVICES		Dashboard	About Us	Food Services	Daily Tracker	Personal Diary	Logout
FOOD ITEM		Ingredients					
ESTIMATE	PERCENTAGE						
Your entered food is samosa							
		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

```

1 from flask import Flask,render_template,request,redirect,url_for ,session
2 import ibm_db
3 import re
4 import os
5 import math
6 import random
7 import smtplib
8 import requests
9 app=Flask(__name__,template_folder='templates',static_folder='static')
10 app.secret_key='a'
11 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8l1cg.databases.appdomain.cloud;PORT=3
12 print("successfully connected")
13 @app.route('/')
14 def home():
15     return render_template('index.html')
16
17
18 @app.route('/login',methods=['GET','POST'])
19 def login():
20     global userid
21     msg=''
22
23     if request.method=='POST':
24         username=request.form.get('username',False)
25         password=request.form.get('password',False)
26         sql='SELECT * FROM USER WHERE username=? AND password=?'
27         stmt=ibm_db.prepare(conn,sql)
28         ibm_db.bind_param(stmt,1,username)
29         ibm_db.bind_param(stmt,2,password)
30         ibm_db.execute(stmt)
31         account=ibm_db.fetch_assoc(stmt)
32         print(account)
33         if account:
34             session['Logged in']=True
35             session['id']=account['USERNAME']
36             userid=account['USERNAME']
37             session['username']=account['USERNAME']
38             msg='Logged in successfully'

```

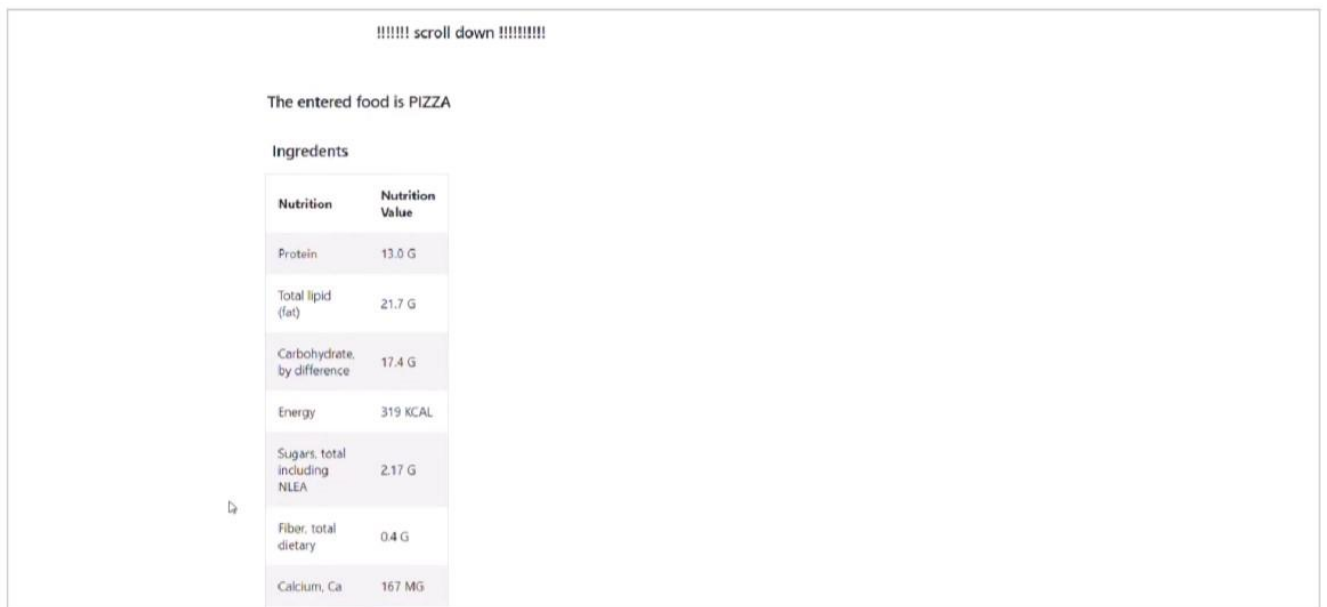
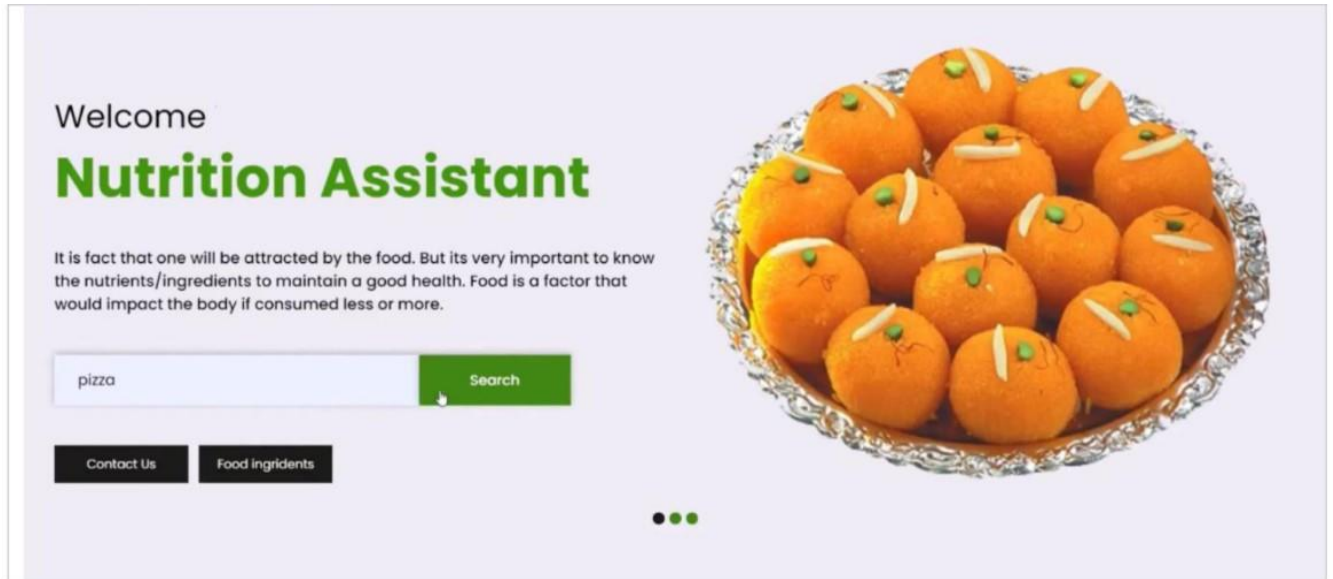
TESTING

1)Test Cases

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

2)User Acceptance Testing

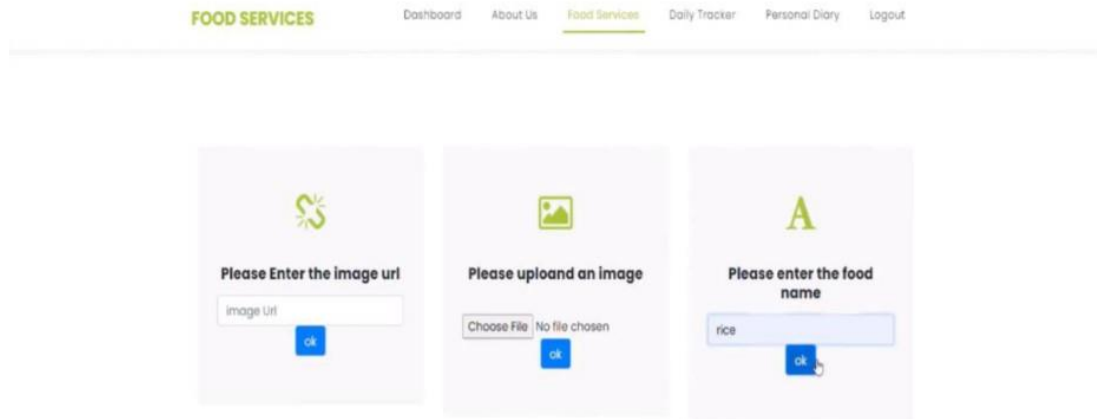
Our project was tested by an end user to verify that it's working correctly.



RESULTS

Performance Metrics

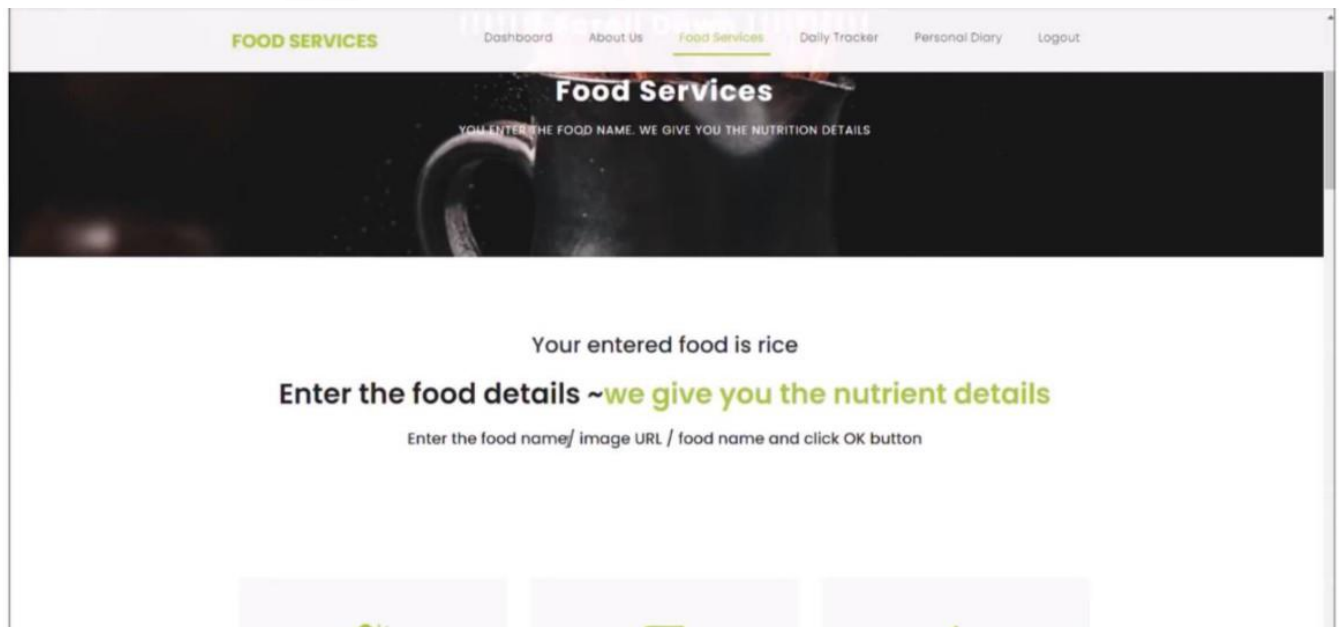
The proposed procedure was implemented and tested set of images. The training database consists of various images of food items. Once a food is recognized the equivalent **Nutri on** is shown on the screen



The image shows three separate input forms for the 'FOOD SERVICES' application. Each form has a title, an icon, a text input field, and an 'OK' button.

- Form 1:** Title 'Please Enter the image url', icon of a leaf, input field 'Image Url', and 'OK' button.
- Form 2:** Title 'Please upload an image', icon of a picture, input field 'Choose File' (showing 'No file chosen'), and 'OK' button.
- Form 3:** Title 'Please enter the food name', icon of a letter 'A', input field 'rice', and 'OK' button.

If the photo is blurry/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.



The image shows the main application interface for 'FOOD SERVICES'. It features a navigation bar with links: Dashboard, About Us, Food Services (active), Daily Tracker, Personal Diary, and Logout. Below the navigation bar is a large banner with the text 'Food Services' and 'YOU ENTER THE FOOD NAME. WE GIVE YOU THE NUTRITION DETAILS'. The main content area displays the result: 'Your entered food is rice' and 'Enter the food details ~we give you the nutrient details'. Below this, there is a sub-header 'Enter the food name/ image URL / food name and click OK button' and three input fields with icons (leaf, picture, and letter A).

FOOD SERVICES		Dashboard	About Us	Food Services	Daily Tracker	Personal Diary	Logout
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	1.22 G				

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.

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.

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

```
        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(prepare_stmt,1,username)
        ibm_db.bind_param(prepare_stmt,2,email)
        ibm_db.bind_param(prepare_stmt,3,password)
        ibm_db.bind_param(prepare_stmt,4,Firstname)
        ibm_db.bind_param(prepare_stmt,5,lastname)
        ibm_db.execute(prepare_stmt)
```

```
@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":request.args['excluded']}
        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-30789-1660189611>