NUTRITION ASSISTANT APPLICATION

NALAIYA THIRAN PROJECT REPORT

SUBMITTED BY

TEAM ID	PNT2022TMID06981
BATCH NO	B5-5M1E
TEAM LEADER	ARIHARAN D (GCTC1918103)
TEAM MEMBER	NISHOK R R (GCTC1918128)
TEAM MEMBER	PRADEEPAN S (GCTC1918129)
TEAM MEMBER	GODS GRACESON M U (GCTC1918301)

of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

GOVERNMENT COLLEGE OF TECHNOLOGY

COIMBATORE – 641013.

1. INTRODUCTION

Diet and nutrition 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.The nutrition and diet planner app is becoming popular among users because of its great usability and amazing convenience

1.1 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. So our Nutrition Assistant Application aims at building a web App that automatically estimates food attributes such as ingredients and nutritional value by classifying the input image of food. It helps to plan and prepare nutritious meals for people who need them. It may also be responsible for educating patients about healthy eating habits.

Our method employs Clarifai's AI-Driven Food Detection Model for accurate food identification and Food API's togive the nutritional value of the identified food. User interacts withthe Web App to Load an image.The image is passed to the serverapplication, which uses Clarifai's AI-Driven Food Detection ModelService to analyze the images and Nutrition API to provide nutritional information about the analyzed Image. Nutritional information of the analyzed image is returned to the app for display. A web based tool is being planned for therapeutic nutrition prescriptions in clinical settings. The cloud based system would have the ability to calculate the nutritional requirements and to guide first line nutritional management to patients and clients automatically. Also, it serves as an electronic medical and dieteticrecord, and personalized nutrition consultation approach can be client can converse to his/ her personal dietitian at their own convenient setting.

Work Flow of the Project:

- User interacts with the Web App to Load an image.
- The image is passed to the server application, which uses Clarifai's AI-Driven Food Detection Model Service to analyze the images and Nutrition API to provide nutritional information about the analyzed Image.
- Nutritional information of the analyzed image is returned to the app for display.

1.2 Purpose:

- Providing dieticians with the facility's meal and menu planning.
- Obtaining dietary information and assessing the nutritional habits of patients.
- Recording individual risk factors or dietary restrictions that might impact meal planning.
- Coordinating meal plans with nutritionists and healthcare professionals.
- Performing ongoing nutrition assessments, including the measurement of caloric intake and activity levels.
- Facilitating immediate interventions for signs of malnutrition, allergic reactions, or refusal to eat.
- Assisting in meal distribution, ensuring correctly delivered, and timely served meals.
- Maintaining proper sterilization protocols in the clearing away

and cleaning of plates and utensils.

- Safely discarding leftover portions to prevent the spread of disease.
- Instructing patients and families on nutrition plans and healthy eating habits.

2. LITERATURE SURVEY

Nutrition and clinical dietetic services provide evidence-based support which has become essential for maintaining healthy lifestyle and avoiding malnutrition among population. National health with digital technology integration is gaining importance in the current COVID-19 pandemic scenario. Digital health technologies offer valuable means for community to create and share information about healthcare.

This research intended to study the effects of utilizing games in health e-learning network on teaching third graders in elementary schools about nutrition. The studied groups of this research were 2 classes of 33 third graders; the two classes were separated into experimental and control group.

The experiment was implemented in a four-week duration. The experimental group learned the knowledge of nutrition based on game playing on a national health e-learning network, whereas the control group waslectured with multi-media slide shows.

2.1 Existing Solutions:

1. MyFitnessPal MyFitnessPal is a powerhouse app, with an enormous food database, barcode scanner, recipe importer, restaurant logger, food insights, calorie counter.

2.Yummly Recipes & Cooking Tools Recipes are sorted and organized by cuisine, course, diet, and required ingredients, making it easy to find something that works for you.

3. Lifesum: Healthy Eating When signing up, the app collects information about your height, weight, age, and specific goals to provide a personalized plan based on your needs.

4. Ate Food Journal Advertised as a mindful food journaling application that doesn’t count calories, Ate Food Journal aims to help you understand why you eat certain foods and how they make you feel.

5. MyNet Diary Calorie Counter The app helps you set goals, monitor your weight trends, and track your intake based on the specific diet plan you select. It also offers detailed nutrient information for each ingredient in your food log and a daily analysis to help keep you on track.

2.2 References:

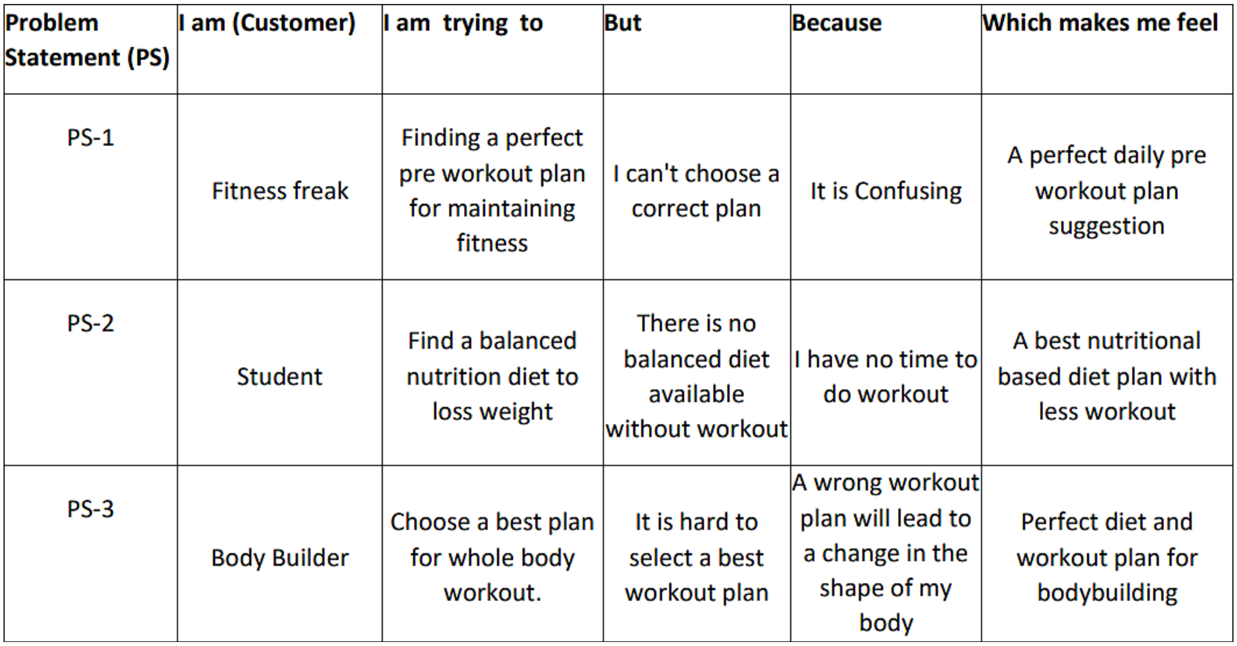
- https://blog.myfitnesspal.com/
- https://www.yummly.com/
- https://lifesum.com/
- https://youate.com/
- https://www.mynetdiary.com/

2.3 Problem Statement Definition:

Due to the ignorance of healthy food habits, obesity rates are increasing at an alarming speed, and this is reflective of the risks to people’s health. People need to control their daily calorie intake by eating healthier foods, which is the most basic method to avoid obesity.

However, although food packaging comes with nutrition (and calorie) labels, it’s still not very convenient for people to refer to App-based nutrient dashboard systems which can analyse real-time images of a meal and analyse it for nutritional content which can be very handy and improves the dietary habits, and therefore, helps in maintaining a healthy lifestyle.

The main objective of this project is to building a web App that automatically estimates food attributes such as ingredients and nutritional value by classifying the input image offood.



Problem Statement (PS)	I am (Customer)	I am trying to	But	Because	Which makes me feel
PS-1	Fitness freak	Finding a perfect pre workout plan for maintaining fitness	I can't choose a correct plan	It is Confusing	A perfect daily pre workout plan suggestion
PS-2	Student	Find a balanced nutrition diet to loss weight	There is no balanced diet available without workout	I have no time to do workout	A best nutritional based diet plan with less workout
PS-3	Body Builder	Choose a best plan for whole body workout.	It is hard to select a best workout plan	A wrong workout plan will lead to a change in the shape of my body	Perfect diet and workout plan for bodybuilding

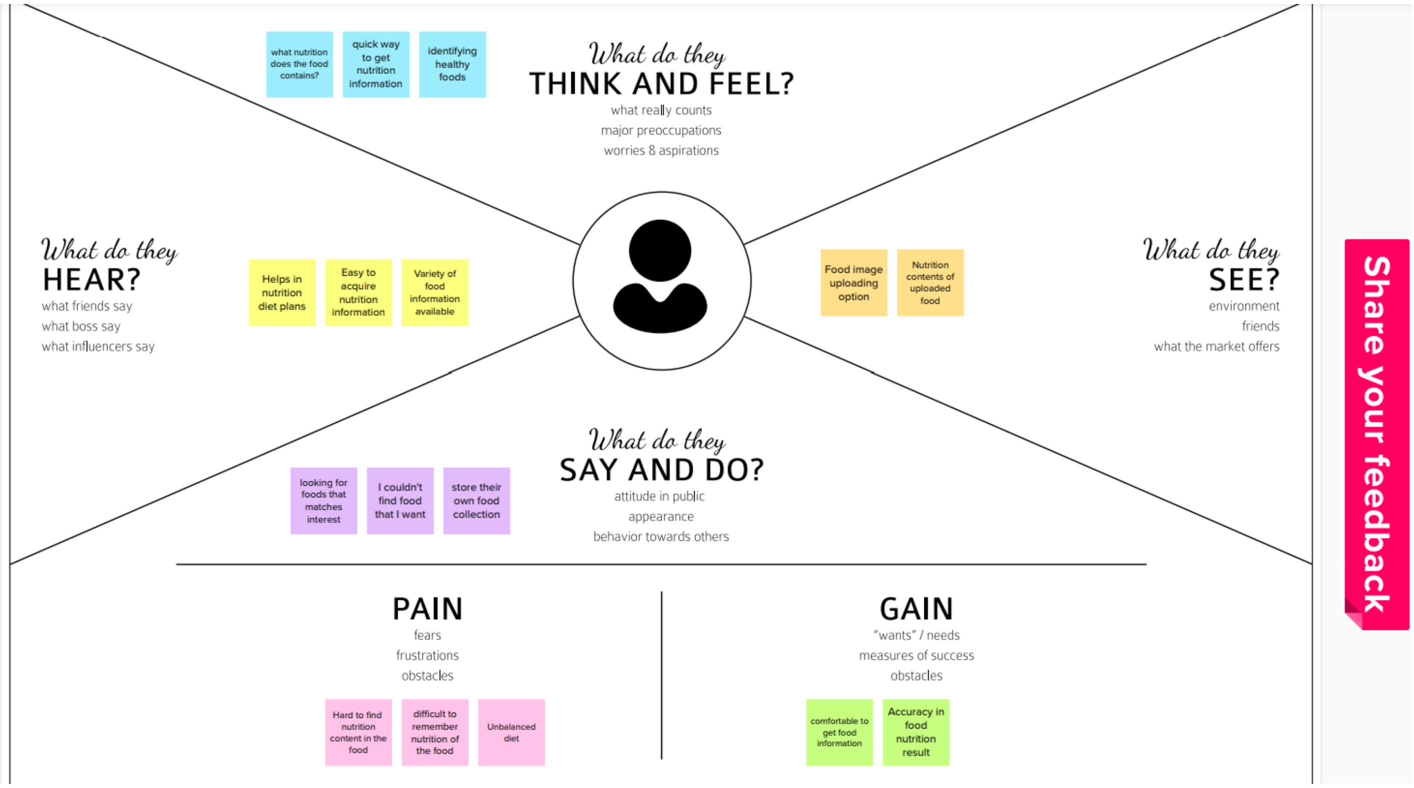
3.IDEATION & PROPOSED SOLUTION

3.1Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes.

It is a useful tool to helps teams better understand their users.
Creating an effective solution requires understanding the true problem and the person who is experiencing it.

The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.


what nutrition does the food contains?
quick way to get nutrition information
identifying healthy foods
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
What do they HEAR?
what friends say
what boss say
what influencers say
Helps in nutrition diet plans
Easy to acquire nutrition information
Variety of food information available
Food image uploading option
Nutrition contents of uploaded food
What do they SEE?
environment
friends
what the market offers
looking for foods that matches interest
I couldn't find food that I want
store their own food collection
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
PAIN
fears
frustrations
obstacles
Hard to find nutrition content in the food
difficult to remember nutrition of the food
Unbalanced diet
GAIN
"wants" / needs
measures of success
obstacles
comfortable to get food information
Accuracy in food nutrition result
Share your feedback

3.2 Ideation & Brainstorming:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the- box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

IDEAS PLANNED:

1) Saving history of each user activity
2) Live photo from camera
3) Activity statistics of user
4) Highlight the nutrients
5) Uploading image from gallery or already stored space
6) Suggestive comments on user
7) Filter in history
8) Show the percentage of nutrients
9) A healthy tip based on user profile
10) Showing food calories depending upon given quantity
11) Identify whether it is food

PRIORITIZED TOP 3 IDEAS BASED ON FEASIBILITY AND IMPORTANCE:

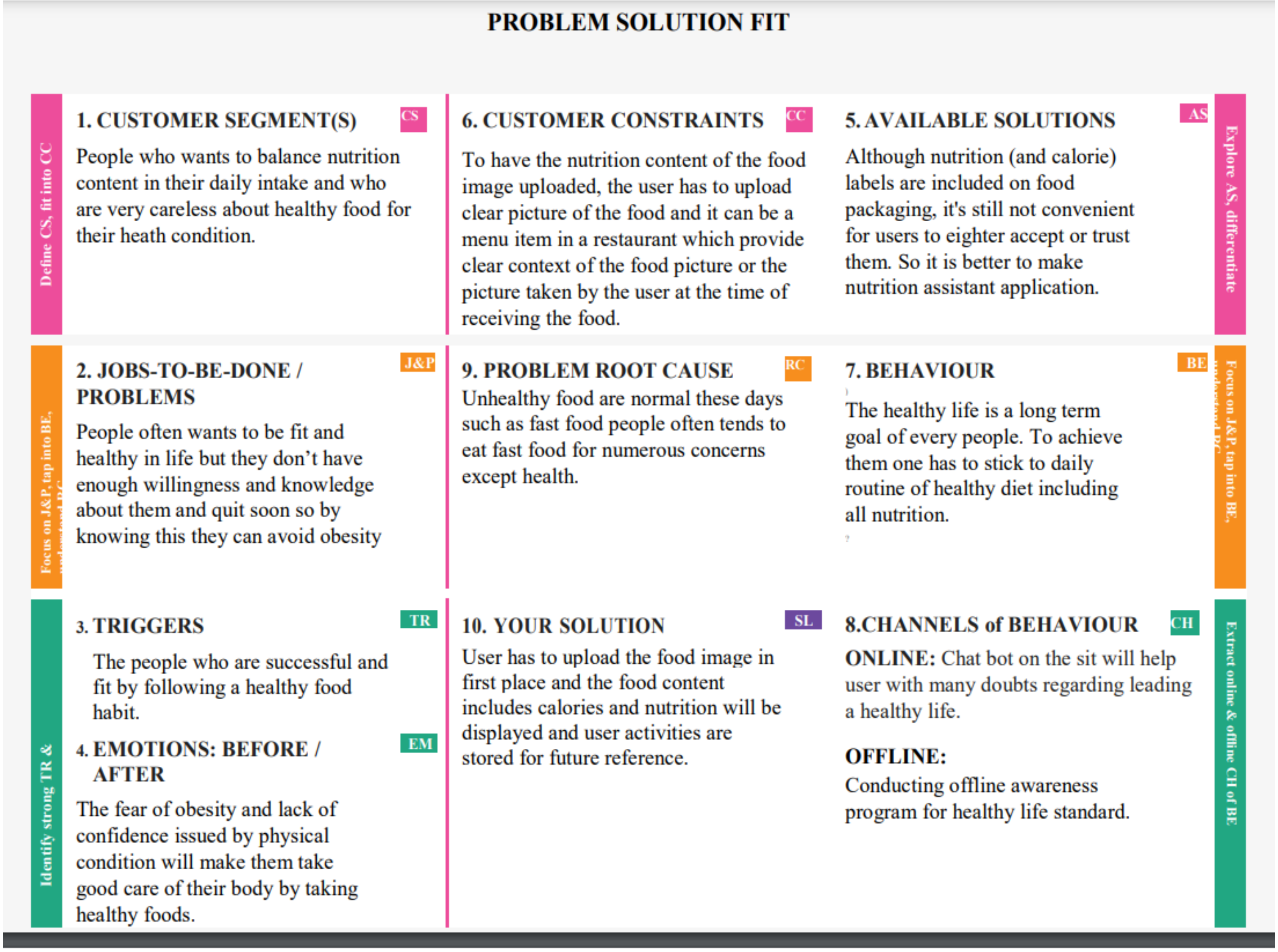
1) Two options for uploading photos of food and describing the food.
2) Save the history of user in IBM DB2 and performing CRUD operation on stored data.
3) Activity statistics of user depending upon their search and activity on the application.

3.3 Proposed Solution:

Sl. No.	Parameter	Description
1	Problem Statement (Problem to be solved)	People seems to be not having enough knowledge about healthy diet and eating unhealthy foods and make them obese and such problem are resolved here.
2	Idea / Solution description	Uploading the real image of food and fetch nutrition and calorie contents of food. Evaluate healthy condition food for user
3	Novelty / Uniqueness	Differentiate people on their health condition and suggest best food for fighting their illness.
4	Social Impact / Customer Satisfaction	This will provide whole some of knowledge about different kind of food taken day to day life.
5	Business Model (Revenue Model)	This application is very successful in market since it is a primary concern in everyone life.
6	Scalability of the Solution	Provides sustainable healthy life development

3.4 Problem Solution fit:

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



PROBLEM SOLUTION FIT

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

People who wants to balance nutrition content in their daily intake and who are very careless about healthy food for their heath condition.

6. CUSTOMER CONSTRAINTS CC

To have the nutrition content of the food image uploaded, the user has to upload clear picture of the food and it can be a menu item in a restaurant which provide clear context of the food picture or the picture taken by the user at the time of receiving the food.

5. AVAILABLE SOLUTIONS AS

Although nutrition (and calorie) labels are included on food packaging, it's still not convenient for users to eighter accept or trust them. So it is better to make nutrition assistant application.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS J&P

People often wants to be fit and healthy in life but they don’t have enough willingness and knowledge about them and quit soon so by knowing this they can avoid obesity

9. PROBLEM ROOT CAUSE RC

Unhealthy food are normal these days such as fast food people often tends to eat fast food for numerous concerns except health.

7. BEHAVIOUR BE

The healthy life is a long term goal of every people. To achieve them one has to stick to daily routine of healthy diet including all nutrition.

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

Identify strong TR &

3. TRIGGERS TR

The people who are successful and fit by following a healthy food habit.

4. EMOTIONS: BEFORE / AFTER EM

The fear of obesity and lack of confidence issued by physical condition will make them take good care of their body by taking healthy foods.

10. YOUR SOLUTION SL

User has to upload the food image in first place and the food content includes calories and nutrition will be displayed and user activities are stored for future reference.

8.CHANNELS of BEHAVIOUR CH

ONLINE: Chat bot on the sit will help user with many doubts regarding leading a healthy life.

OFFLINE:
Conducting offline awareness program for healthy life standard.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 Functional requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration with email and password
FR-2	User Confirmation	Confirmation via Email
FR-3	User Profile Completion	Fetch personal details such as weight, height etc..,
FR-4	Gather meal image	Upload photo from gallery or take photo of the food and upload it on the website
FR-5	Display calorie information	Use Clarifai API to get the food details and display calorie information

4.2 Non-Functional requirements:

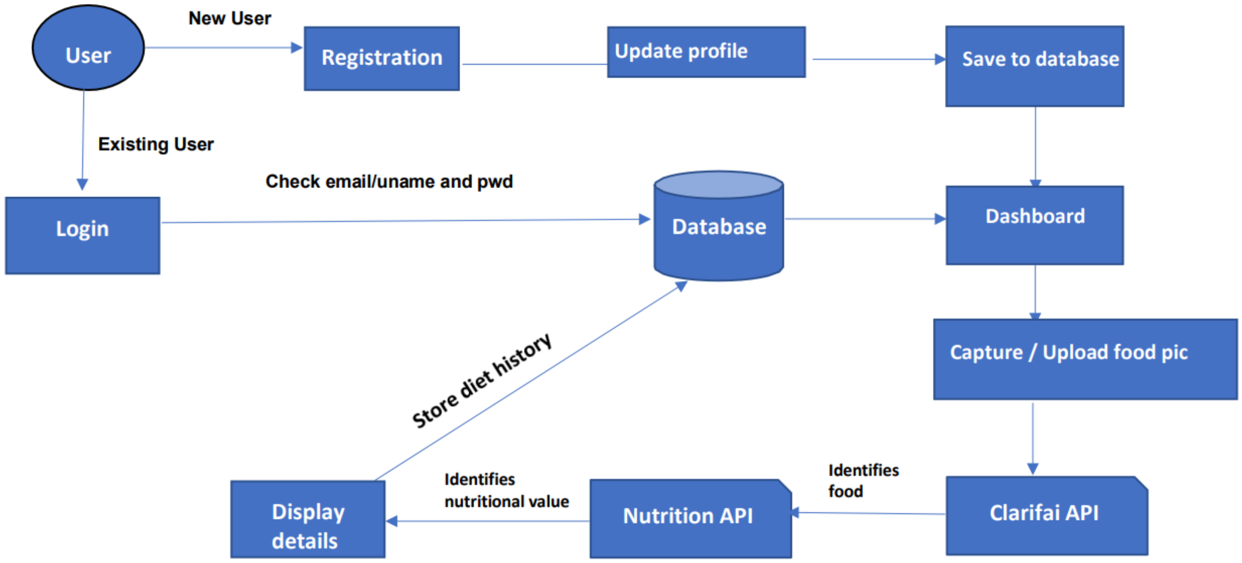
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Better user friendly interface for user excitement of using the application.
NFR-2	Security	Strong authorization and authentication scheme for accessing user information
NFR-3	Reliability	The system ensure the no compromises over user available info on database.
NFR-4	Performance	Responsiveness should be primary concern while many users actively using the application at the same time.
NFR-5	Availability	Service should be available 24/7
NFR-6	Scalability	Provides better scaling for higher workloads

5. PROJECT DESIGN

5.1 Data Flow Diagrams:

A data-flow diagram is a way of representing a flow of data through a process or a system. A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture:

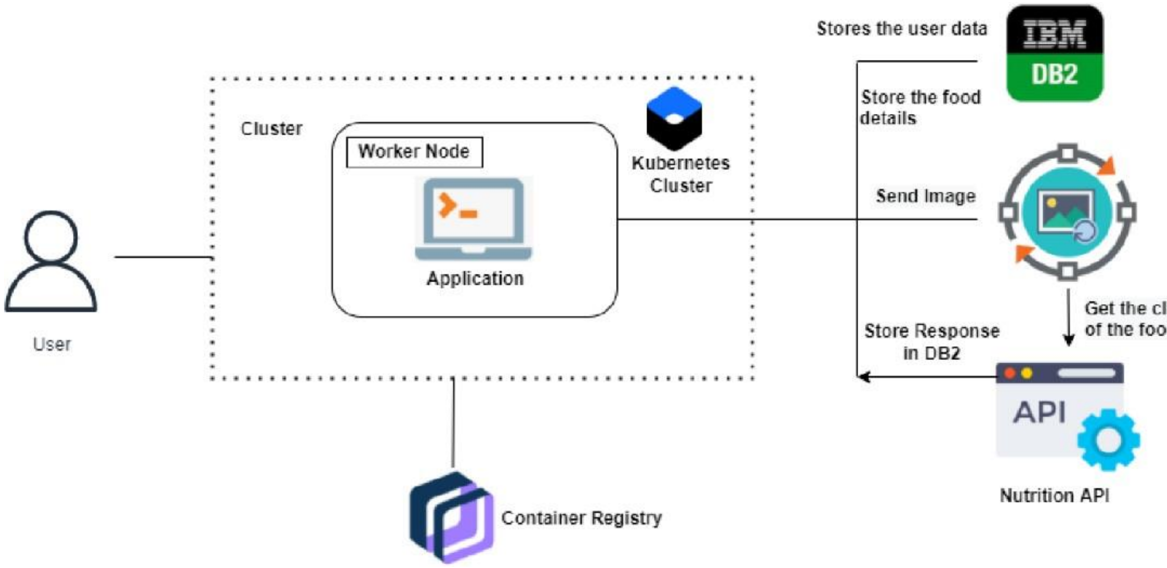
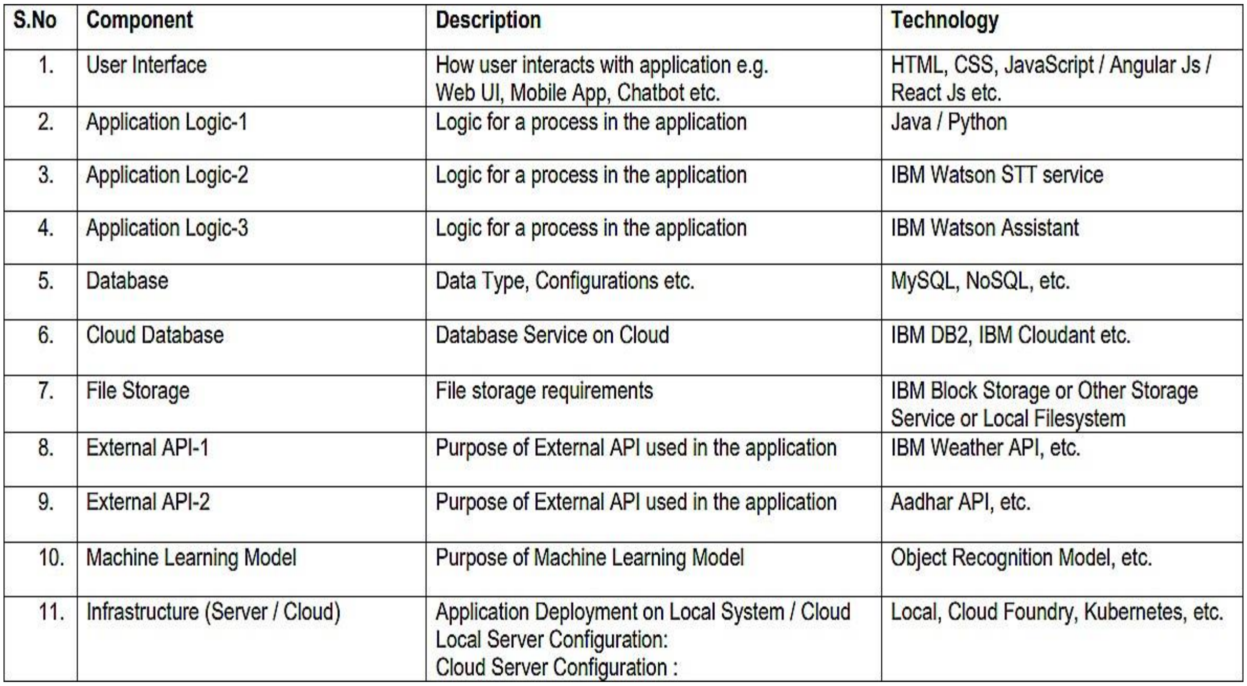
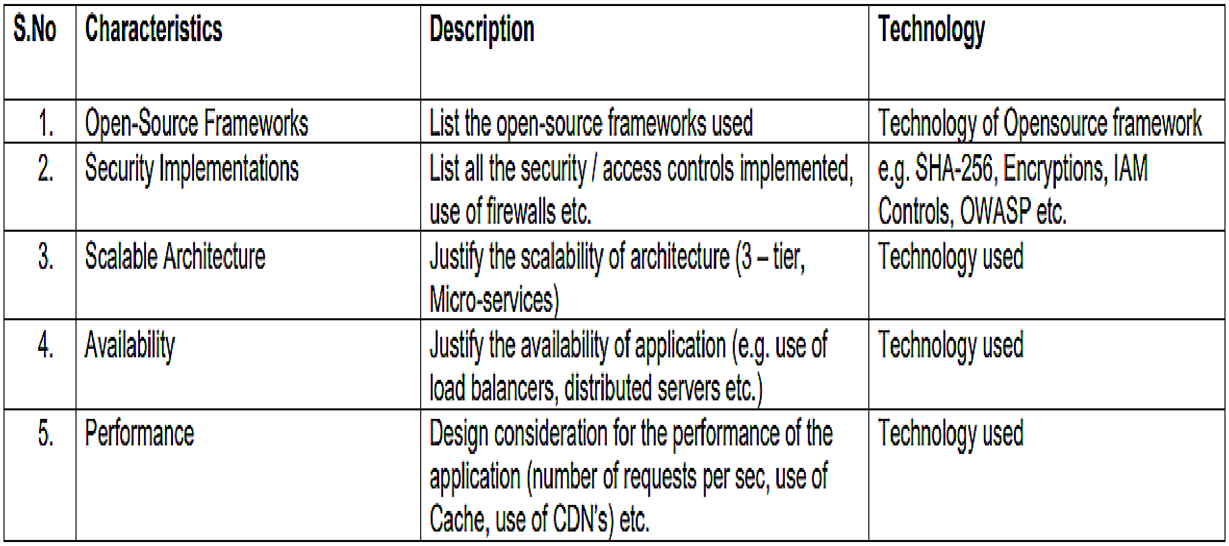


Table-1 : Components & Technologies:



S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

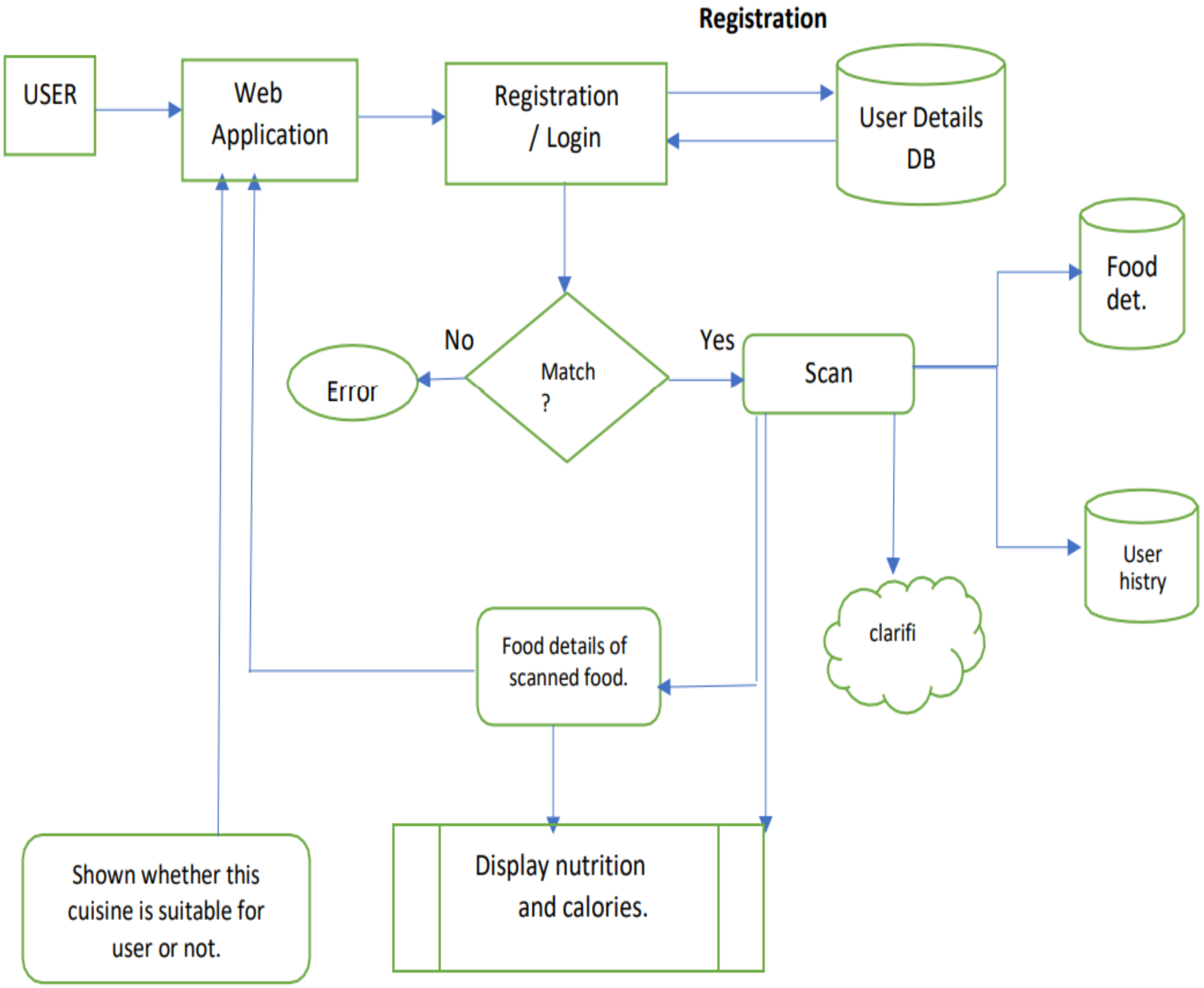


S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

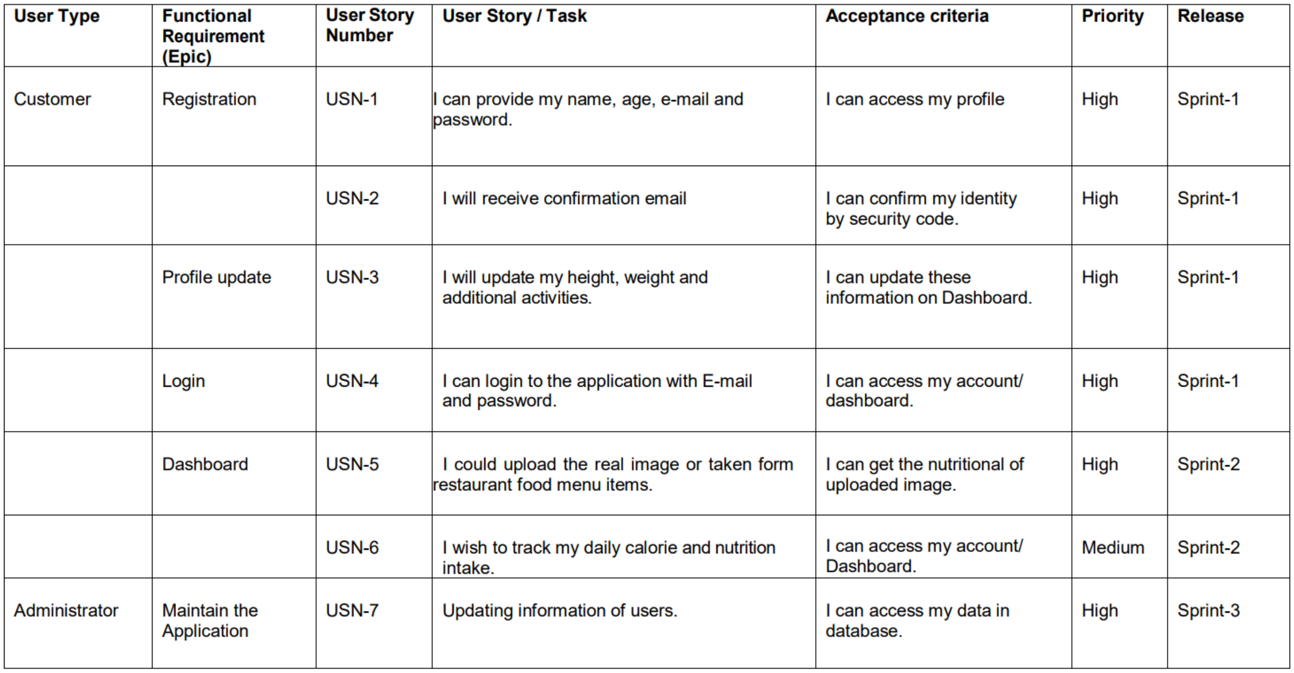
Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

Solution Architecture Diagram for Nutrition Assistant Application :


Registration
USER
Web Application
Registration / Login
User Details DB
Food det.
No
Match ?
Yes
Error
Scan
User histry
clarifi
Food details of scanned food.
Shown whether this cuisine is suitable for user or not.
Display nutrition and calories.

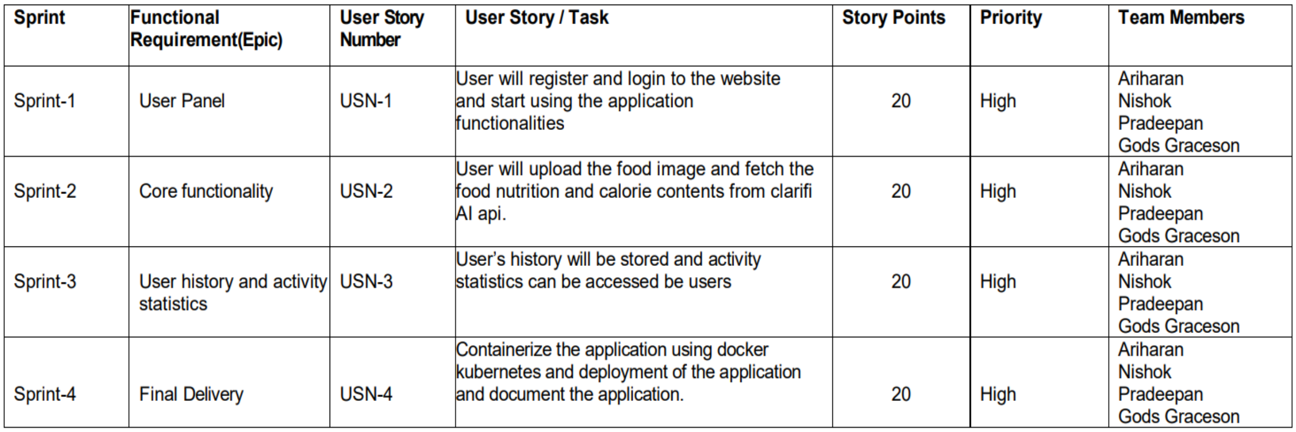
5.3 User Stories:



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	I can provide my name, age, e-mail and password.	I can access my profile	High	Sprint-1
		USN-2	I will receive confirmation email	I can confirm my identity by security code.	High	Sprint-1
	Profile update	USN-3	I will update my height, weight and additional activities.	I can update these information on Dashboard.	High	Sprint-1
	Login	USN-4	I can login to the application with E-mail and password.	I can access my account/ dashboard.	High	Sprint-1
	Dashboard	USN-5	I could upload the real image or taken form restaurant food menu items.	I can get the nutritional of uploaded image.	High	Sprint-2
		USN-6	I wish to track my daily calorie and nutrition intake.	I can access my account/ Dashboard.	Medium	Sprint-2
Administrator	Maintain the Application	USN-7	Updating information of users.	I can access my data in database.	High	Sprint-3

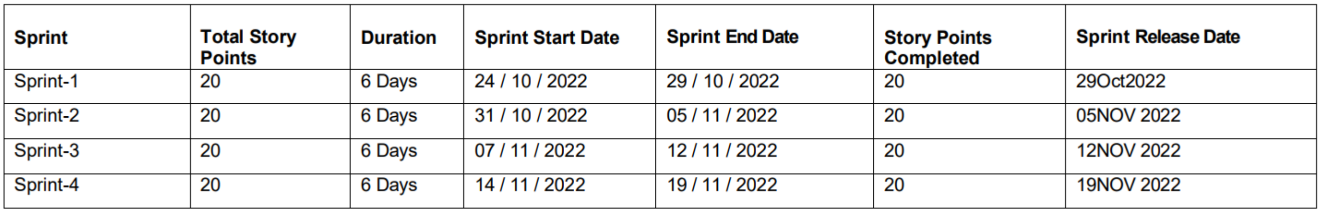
6 PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:



Sprint	Functional Requirement(Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Panel	USN-1	User will register and login to the website and start using the application functionalities	20	High	Ariharan Nishok Pradeepan Gods Graceson
Sprint-2	Core functionality	USN-2	User will upload the food image and fetch the food nutrition and calorie contents from clarifi AI api.	20	High	Ariharan Nishok Pradeepan Gods Graceson
Sprint-3	User history and activity statistics	USN-3	User's history will be stored and activity statistics can be accessed be users	20	High	Ariharan Nishok Pradeepan Gods Graceson
Sprint-4	Final Delivery	USN-4	Containerize the application using docker kubernetes and deployment of the application and document the application.	20	High	Ariharan Nishok Pradeepan Gods Graceson

6.2 Sprint Delivery Schedule:



Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date	Story Points Completed	Sprint Release Date
Sprint-1	20	6 Days	24 / 10 / 2022	29 / 10 / 2022	20	29Oct2022
Sprint-2	20	6 Days	31 / 10 / 2022	05 / 11 / 2022	20	05NOV 2022
Sprint-3	20	6 Days	07 / 11 / 2022	12 / 11 / 2022	20	12NOV 2022
Sprint-4	20	6 Days	14 / 11 / 2022	19 / 11 / 2022	20	19NOV 2022

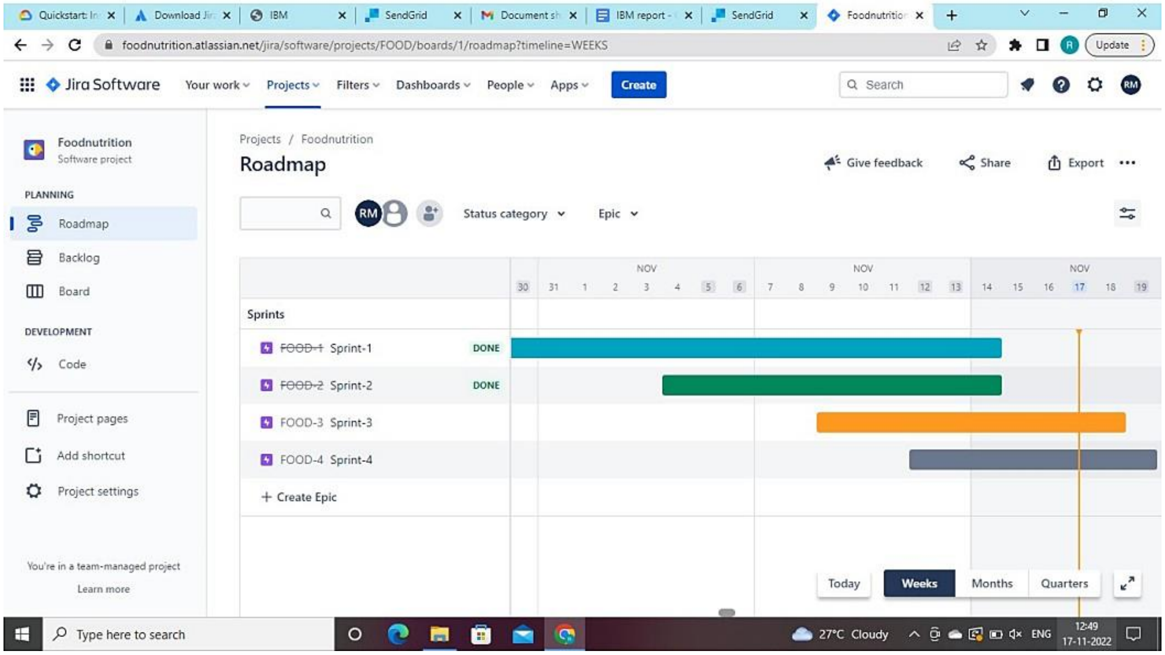
Velocity:

Imagine that we have a 10-day sprint duration and the velocity of the team is 20 (points per sprint). Let us calculate the team’s average velocity(AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3 Reports from JIRA:

JIRA Roadmap



foodnutrition.atlassian.net/jira/software/projects/FOOD/boards/1/roadmap?timeline=WEEKS

Sprints

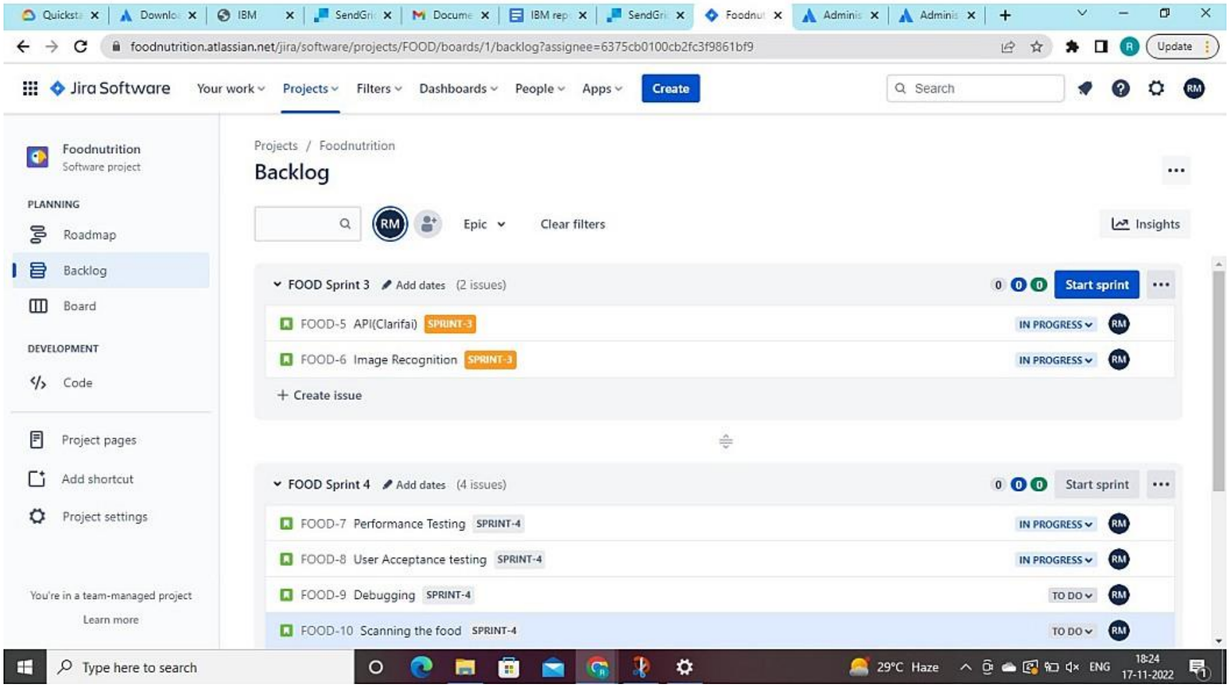
~~FOOD-1~~ Sprint-1 DONE

~~FOOD-2~~ Sprint-2 DONE

FOOD-3 Sprint-3

FOOD-4 Sprint-4

JIRA Backlog



foodnutrition.atlassian.net/jira/software/projects/FOOD/boards/1/backlog?assignee=6375cb0100cb2fc3f9861bf9

Jira Software | Your work | Projects | Filters | Dashboards | People | Apps | Create | Search

Foodnutrition
Software project

PLANNING
- Roadmap
- Backlog
- Board

DEVELOPMENT
- Code
- Project pages
- Add shortcut
- Project settings

You're in a team-managed project
Learn more

Projects / Foodnutrition

Backlog

Epic | Clear filters | Insights

FOOD Sprint 3 · Add dates (2 issues) 0 0 0 Start sprint

FOOD-5	API(Clarifai)	SPRINT-3	IN PROGRESS	RM
FOOD-6	Image Recognition	SPRINT-3	IN PROGRESS	RM

+ Create issue

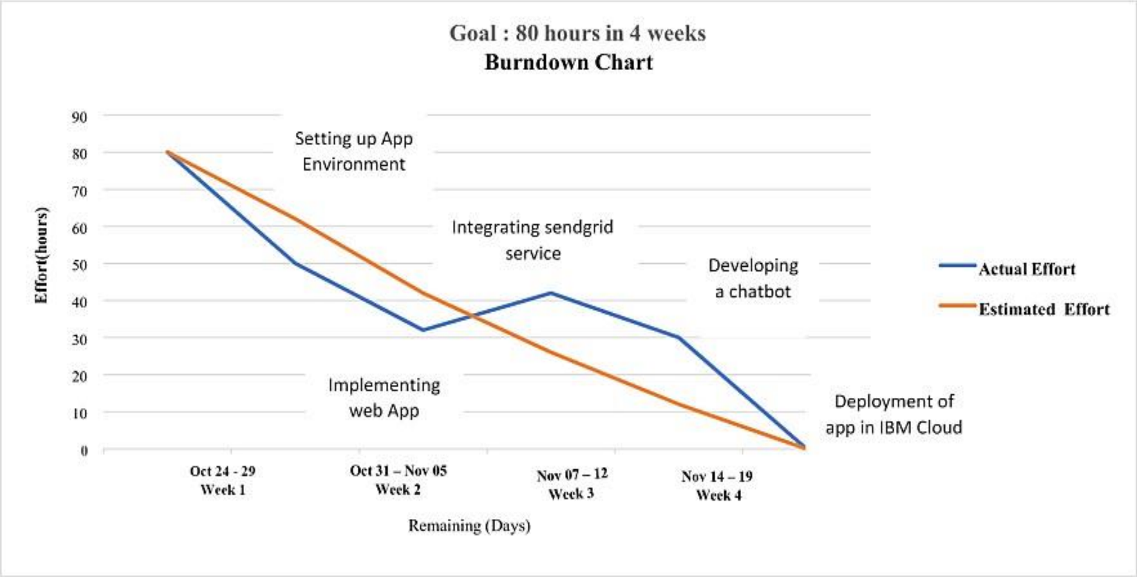
FOOD Sprint 4 · Add dates (4 issues) 0 0 0 Start sprint

FOOD-7	Performance Testing	SPRINT-4	IN PROGRESS	RM
FOOD-8	User Acceptance testing	SPRINT-4	IN PROGRESS	RM
FOOD-9	Debugging	SPRINT-4	TO DO	RM
FOOD-10	Scanning the food	SPRINT-4	TO DO	RM

Type here to search | 29°C Haze | ENG | 18:24 17-11-2022

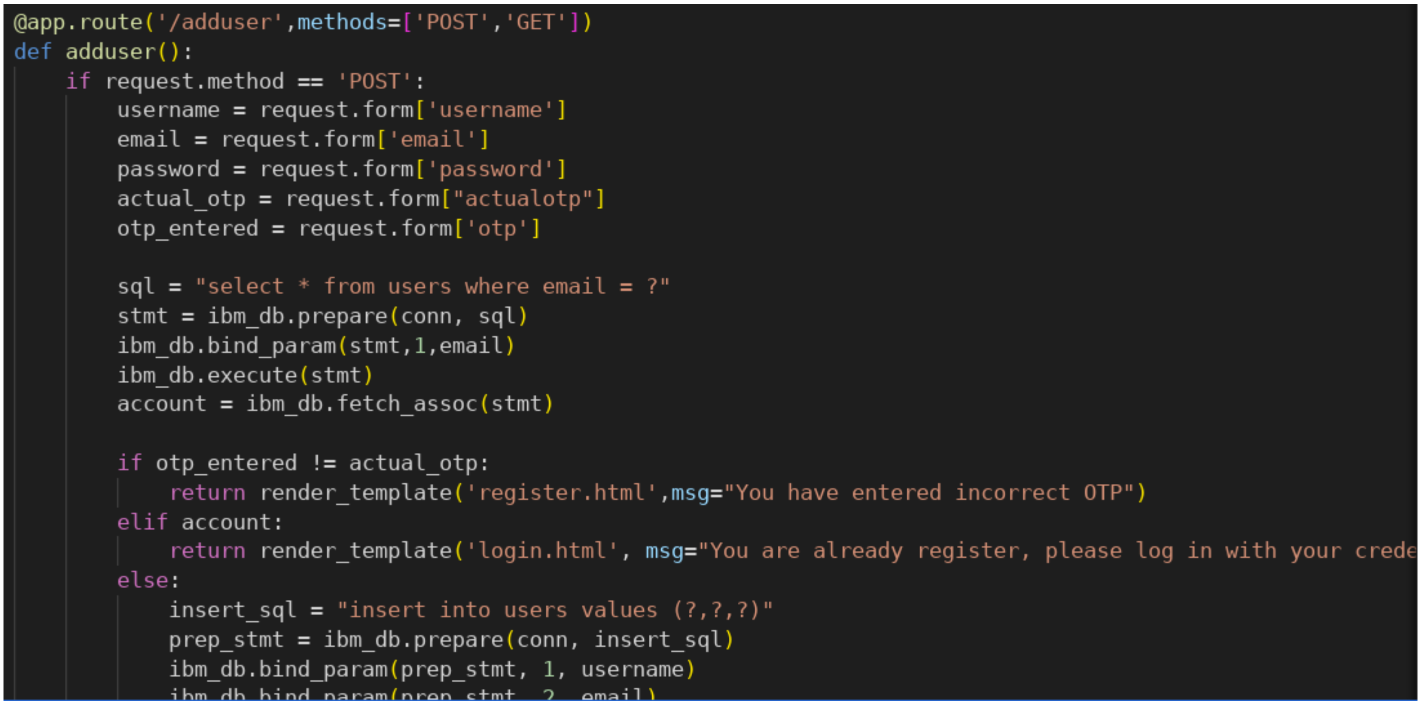
Burndown Chart

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

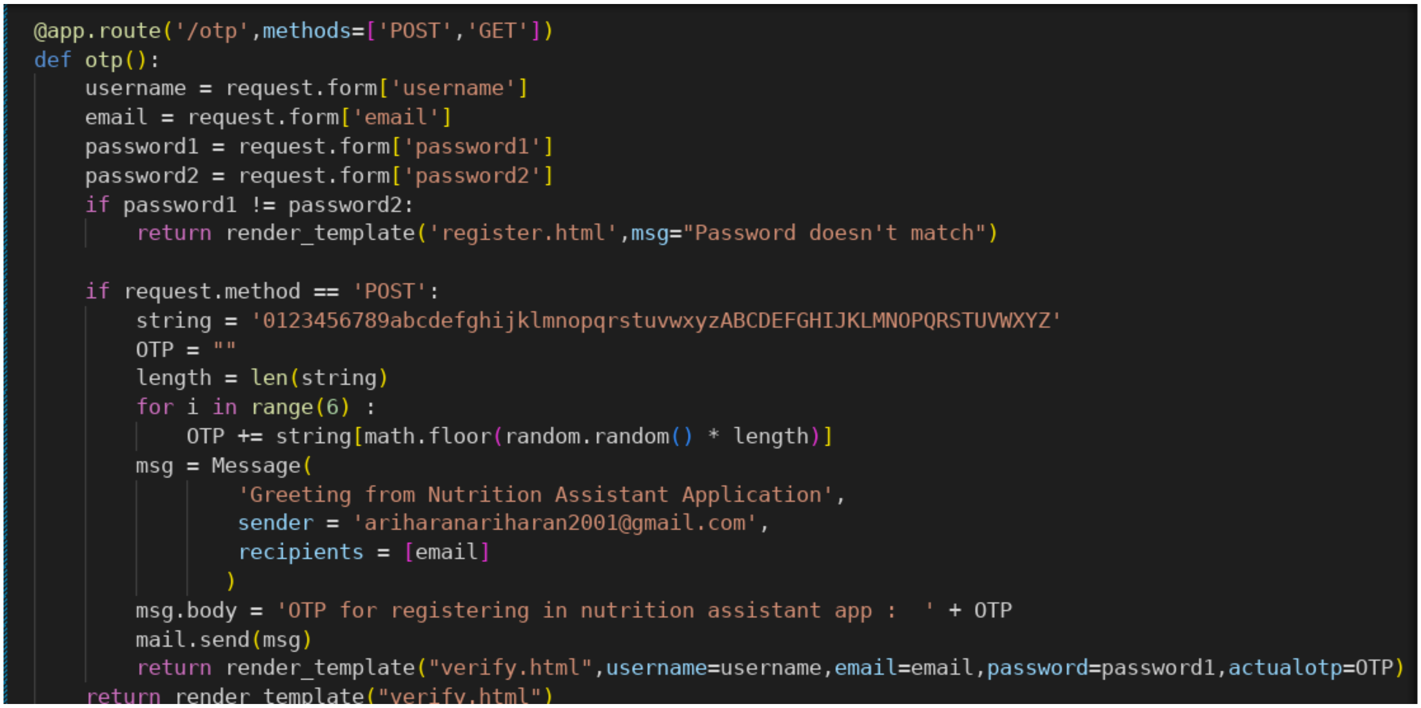


7. CODING & SOLUTIONING

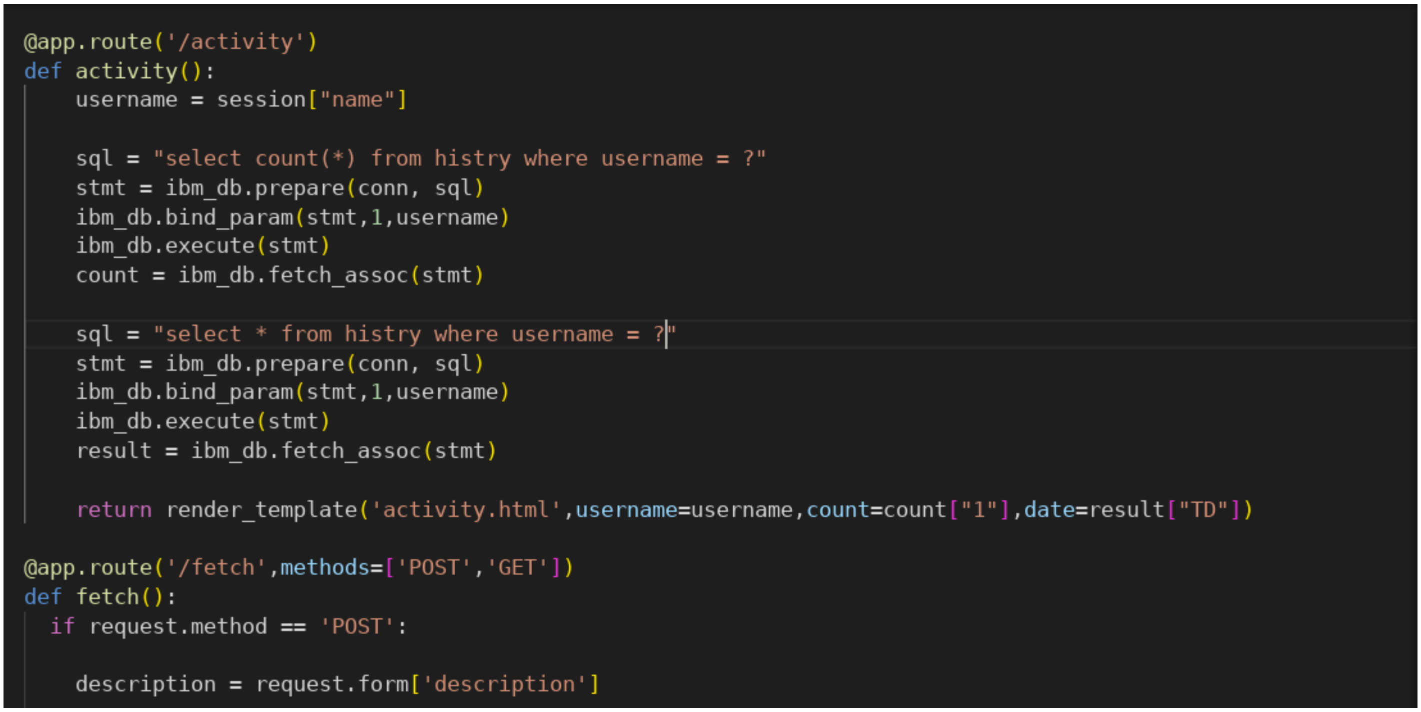
7.1 Feature 1:



```
@app.route('/adduser',methods=['POST','GET'])
def adduser():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        actual_otp = request.form["actualotp"]
        otp_entered = request.form['otp']

        sql = "select * from users where email = ?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if otp_entered != actual_otp:
            return render_template('register.html',msg="You have entered incorrect OTP")
        elif account:
            return render_template('login.html', msg="You are already register, please log in with your crede
        else:
            insert_sql = "insert into users 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)
```



```python
@app.route('/otp',methods=['POST','GET'])
def otp():
    username = request.form['username']
    email = request.form['email']
    password1 = request.form['password1']
    password2 = request.form['password2']
    if password1 != password2:
        return render_template('register.html',msg="Password doesn't match")

    if request.method == 'POST':
        string = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
        OTP = ""
        length = len(string)
        for i in range(6) :
            OTP += string[math.floor(random.random() * length)]
        msg = Message(
                'Greeting from Nutrition Assistant Application',
                sender = 'ariharanariharan2001@gmail.com',
                recipients = [email]
               )
        msg.body = 'OTP for registering in nutrition assistant app :  ' + OTP
        mail.send(msg)
        return render_template("verify.html",username=username,email=email,password=password1,actualotp=OTP)
    return render_template("verify.html")
```

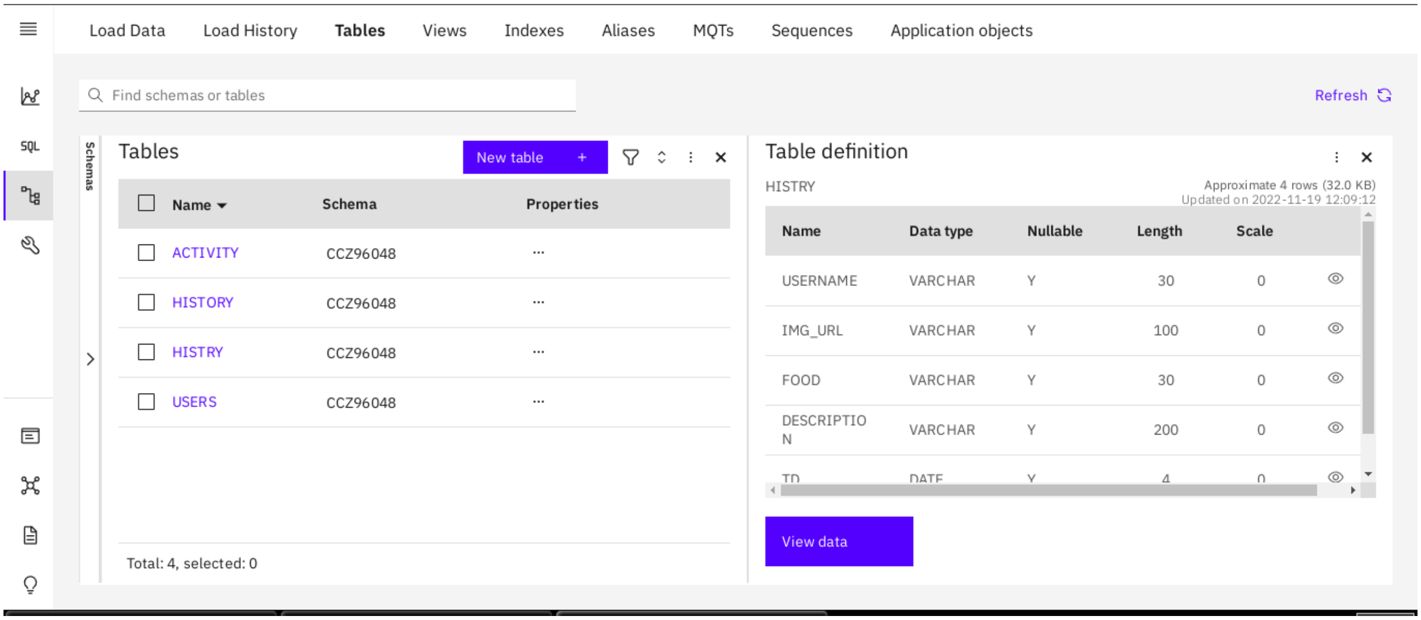


```python
@app.route('/activity')
def activity():
    username = session["name"]

    sql = "select count(*) from histry where username = ?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    count = ibm_db.fetch_assoc(stmt)

    sql = "select * from histry where username = ?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    result = ibm_db.fetch_assoc(stmt)

    return render_template('activity.html',username=username,count=count["1"],date=result["TD"])

@app.route('/fetch',methods=['POST','GET'])
def fetch():
  if request.method == 'POST':

    description = request.form['description']
```

7.2 Database

Schema:IBM Db2 ON

CLOUD:



Tables

☐	Name	Schema	Properties
☐	ACTIVITY	CCZ96048	...
☐	HISTORY	CCZ96048	...
☐	HISTRY	CCZ96048	...
☐	USERS	CCZ96048	...

Total: 4, selected: 0

Table definition

HISTRY

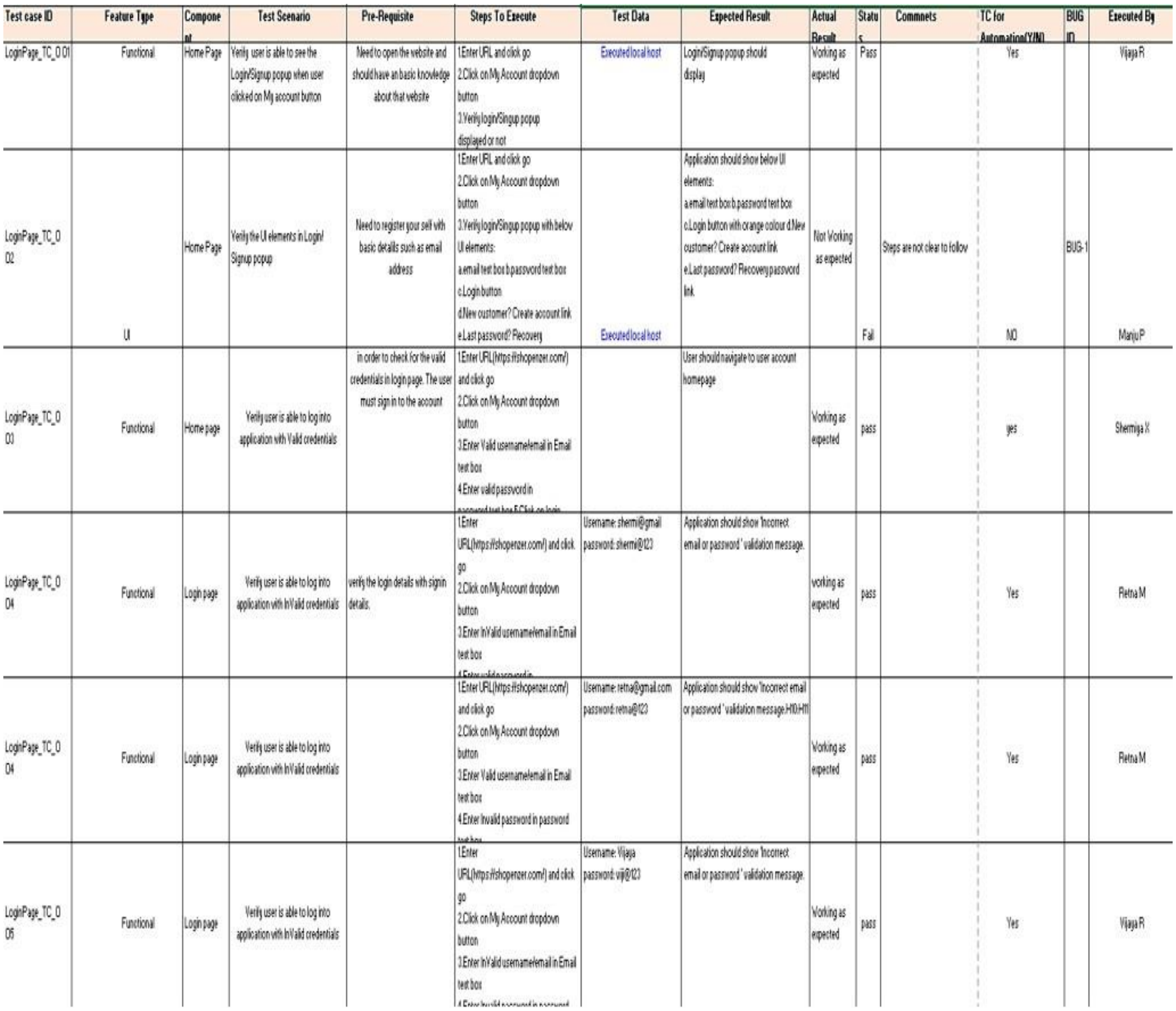
Approximate 4 rows (32.0 KB)
Updated on 2022-11-19 12:09:12

Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	Y	30	0
IMG_URL	VARCHAR	Y	100	0
FOOD	VARCHAR	Y	30	0
DESCRIPTION	VARCHAR	Y	200	0
ID	DATE	Y	4	0

View data

8. TESTING

8.1 Test Cases:



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_0 01	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	Need to open the website and should have an basic knowledge about that website	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	Executed local host	Login/Signup popup should display	Working as expected	Pass		Yes		Vijaya R
LoginPage_TC_0 02	UI	Home Page	Verify the UI elements in Login/ Signup popup	Need to register your self with basic details such as email address	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery	Executed local host	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Not Working as expected	Fail	Steps are not clear to follow	NO	BUG-1	Manju P
LoginPage_TC_0 03	Functional	Home page	Verify user is able to log into application with Valid credentials	in order to check for the valid credentials in login page. The user must sign in to the account	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login		User should navigate to user account homepage	Working as expected	pass		yes		Shermiya X
LoginPage_TC_0 04	Functional	Login page	Verify user is able to log into application with InValid credentials	verify the login details with signin details.	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in	Username: shermi@gmail password: shermi@123	Application should show 'Incorrect email or password ' validation message.	working as expected	pass		Yes		Retna M
LoginPage_TC_0 04	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box	Username: retna@gmail.com password: retna@123	Application should show 'Incorrect email or password ' validation message.H10:H11	Working as expected	pass		Yes		Retna M
LoginPage_TC_0 05	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password	Username: Vijaya password: viji@123	Application should show 'Incorrect email or password ' validation message.	Working as expected	pass		Yes		Vijaya R

8.2 User Acceptance Testing:

UAT Execution & Report Submission

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Smart Fashion Recommender Application project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Model	1	2	1	0	3
Duplicate	1	0	0	0	1
External	2	0	0	1	3
Fixed	7	2	3	0	12
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	1	0	0	1
Totals	11	5	6	2	23

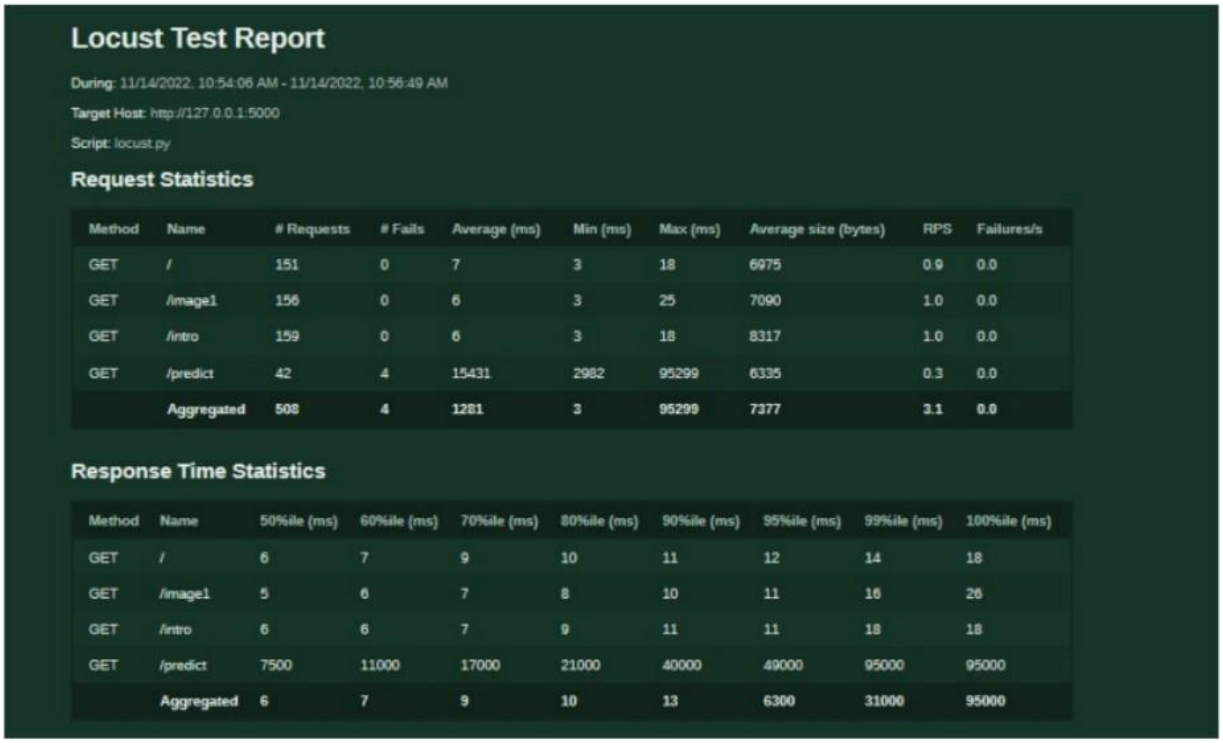
Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Hypothesis Condition	2	0	0	2
Train Test Split	5	2	0	3
Hyper Tuning Parameter Test	4	0	0	4
ConfusionMatrix	1	0	0	1
Logistic Regression	1	0		1
Final Report Output	6	2	0	4
SVM Model	1	0	0	1

9. RESULTS

9.1 Performance Metrics:



Locust Test Report

During: 11/14/2022, 10:54:06 AM - 11/14/2022, 10:56:49 AM

Target Host: http://127.0.0.1:5000

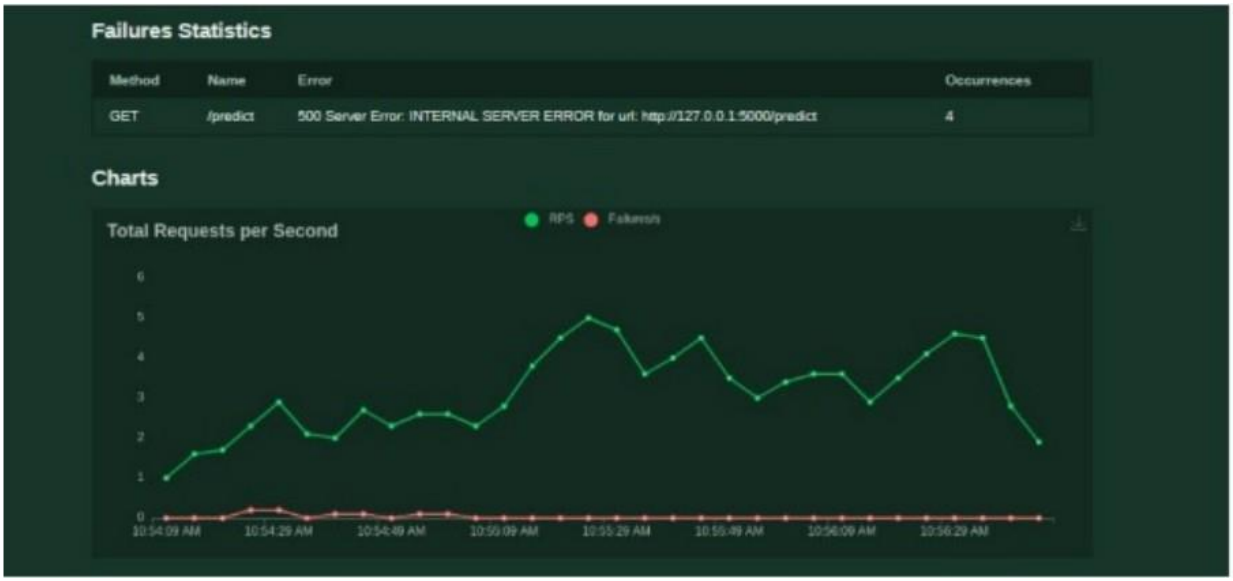
Script: locust.py

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	/	151	0	7	3	18	6975	0.9	0.0
GET	/image1	156	0	6	3	25	7090	1.0	0.0
GET	/intro	159	0	6	3	18	8317	1.0	0.0
GET	/predict	42	4	15431	2982	95299	6335	0.3	0.0
	Aggregated	**508**	**4**	**1281**	**3**	**95299**	**7377**	**3.1**	**0.0**

Response Time Statistics

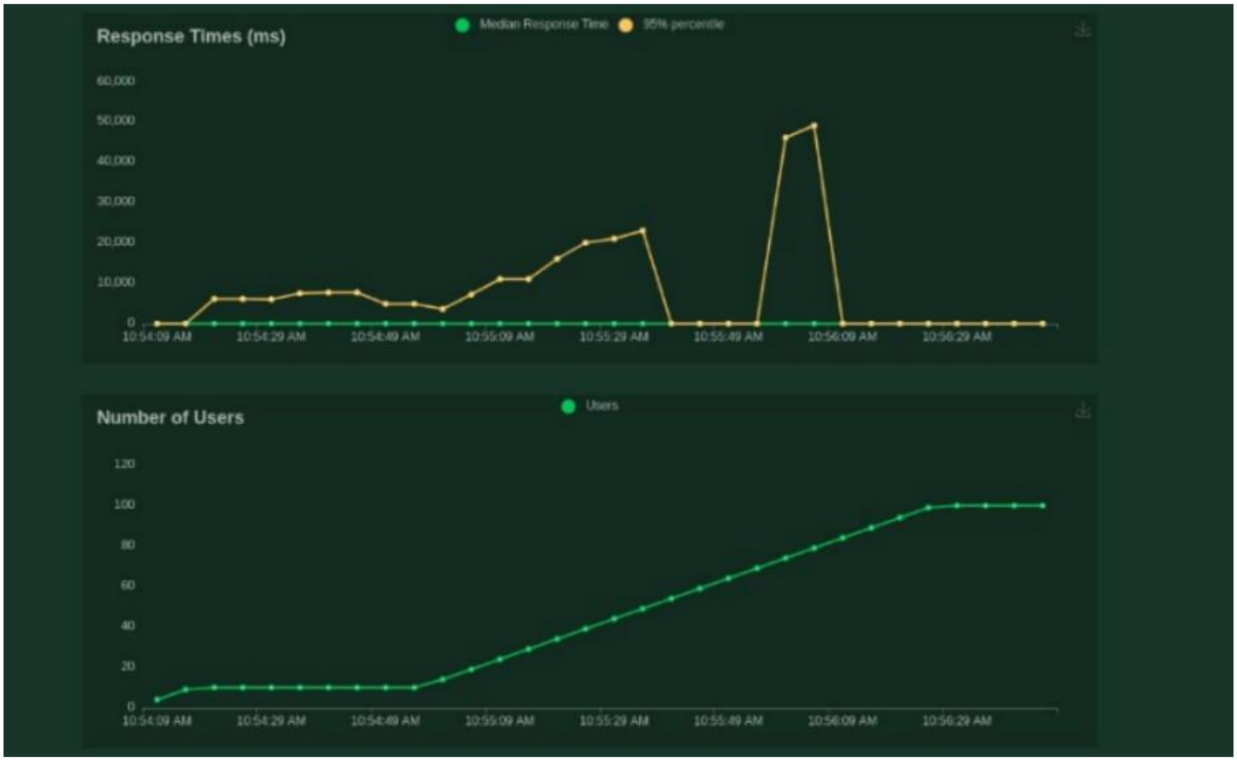
Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	/	6	7	9	10	11	12	14	18
GET	/image1	5	6	7	8	10	11	16	26
GET	/intro	6	6	7	9	11	11	18	18
GET	/predict	7500	11000	17000	21000	40000	49000	95000	95000
	Aggregated	**6**	**7**	**9**	**10**	**13**	**6300**	**31000**	**95000**



Failures Statistics

Method	Name	Error	Occurrences
GET	/predict	500 Server Error: INTERNAL SERVER ERROR for url: http://127.0.0.1:5000/predict	4

Charts



10. ADVANTAGES & DISADVANTAGES

Advantages:

The major advantage of this tool is that they can help us to eat healthier.

- It is also easy to track our progress.
- It provides general awareness of nutrients in food
- Keep you motivated.
- All in one health tool.

Disadvantages:

The tool can be quite expensive as it requires cameras and other expensive devices to capture images and process it.

● This tool may not always be 100% accurate.

● We might avoid certain healthy foods that are difficult to add into the food tracker.

11. CONCLUSION

- In this project we developed a tool which recognizes our health and calorific value.
- It helps us to eat nutritional food. The diet chartwill be provided to individual users based on user's calorific value. It allows the users to upload their food images and give suggestion to that food. It also does not require the user to have any device on them to use it.
- Further this technology can be extended to other industries like it can be used by presenters, by teachers for show images in the classroom, etc.

12. FUTURE SCOPE

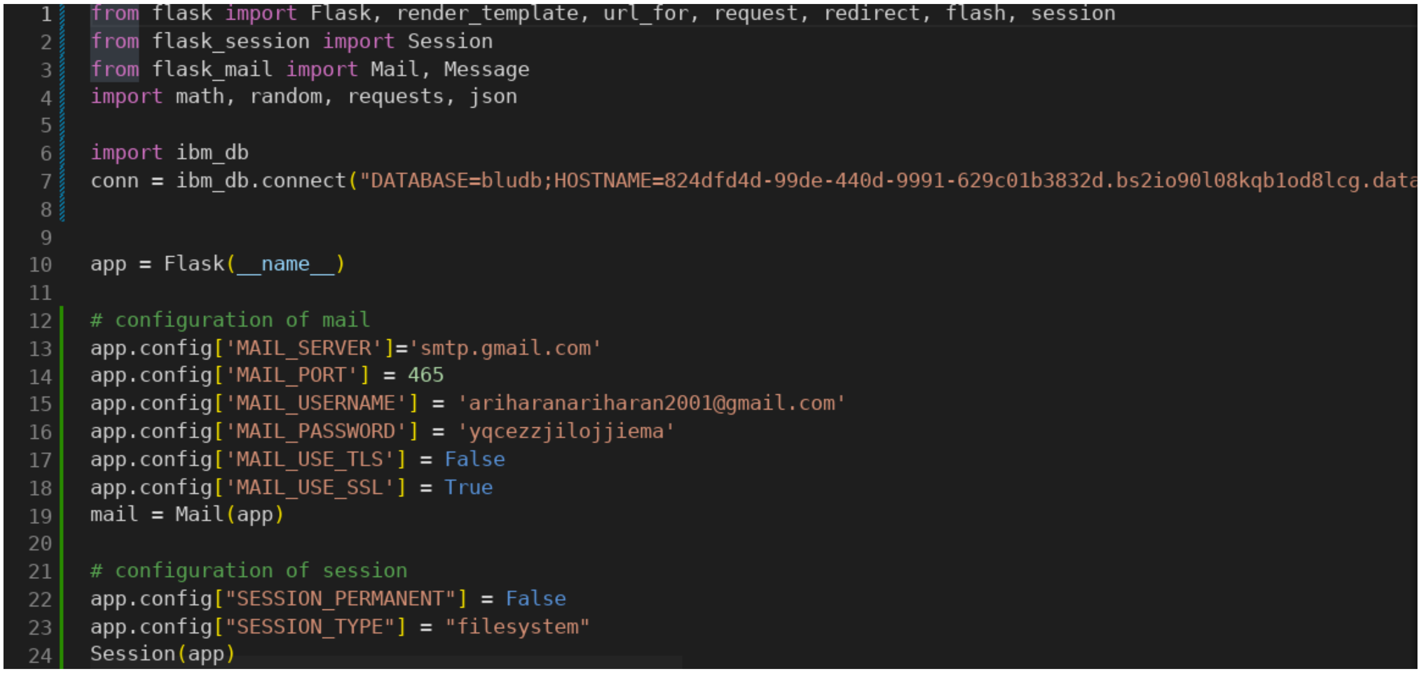
The tool can be made quicker by increasing the recognition speed.

- They can work with a licensed healthcare provider to help individuals with previously diagnosed disease recognize biochemical imbalances and toxicity which lead to poor health.
- Voice commands can also be added to further increase the functionality.

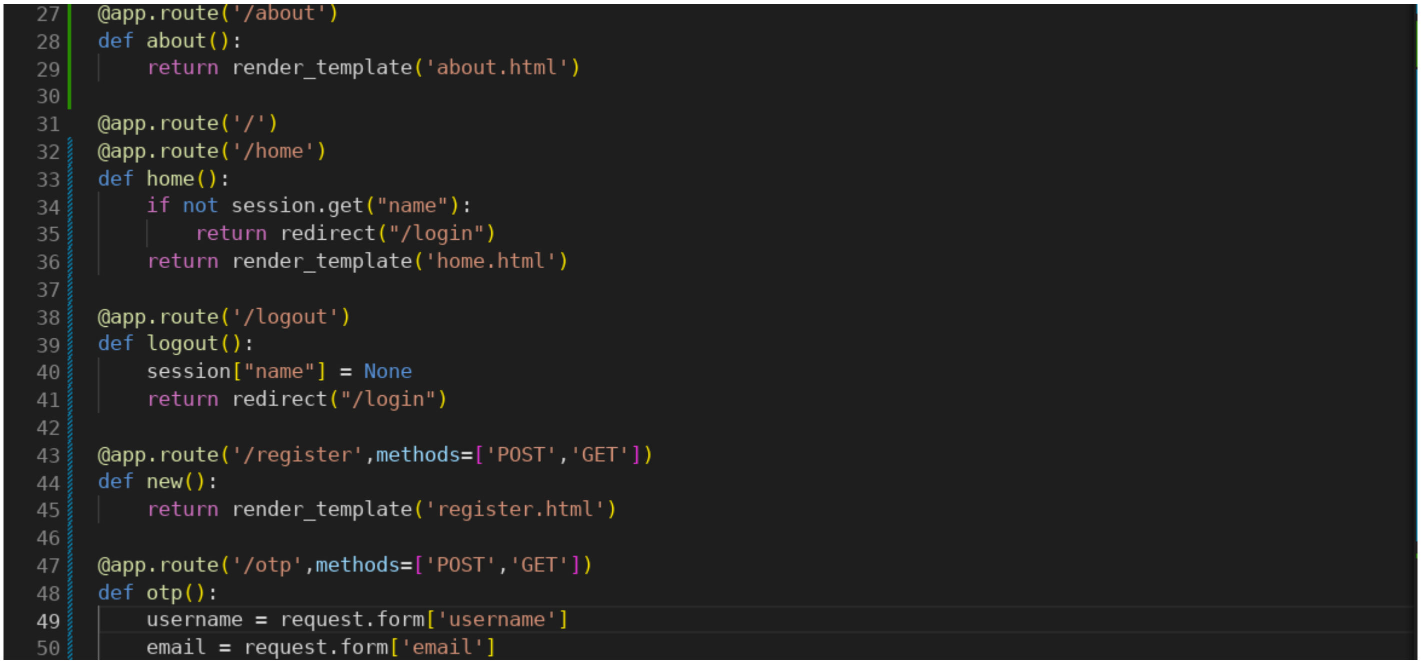
In summary, our study shows different challenges that health-focused nutritional assistance systems face when being used in the long term. Our findings can be used to improve future system regarding their impact in the long-term and to postulate more long-term evaluation of recommender approaches

13. APPENDIX

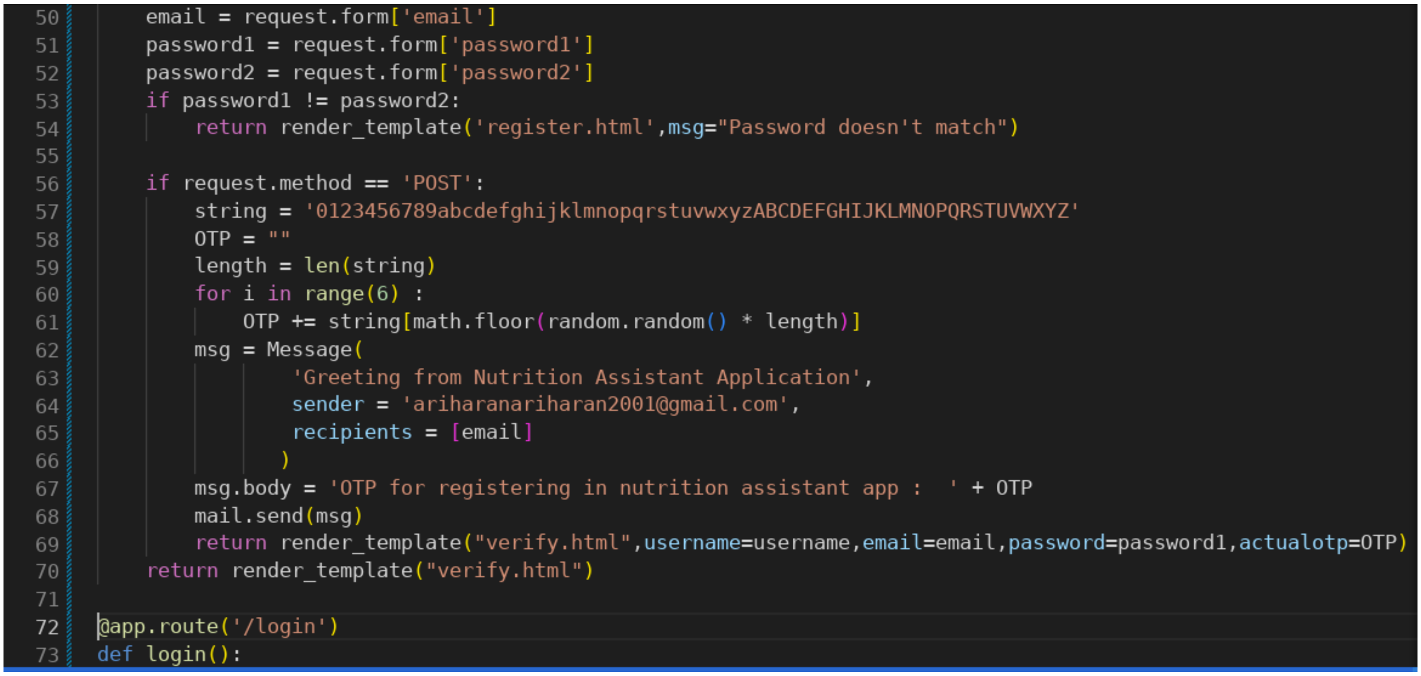
Source Code:



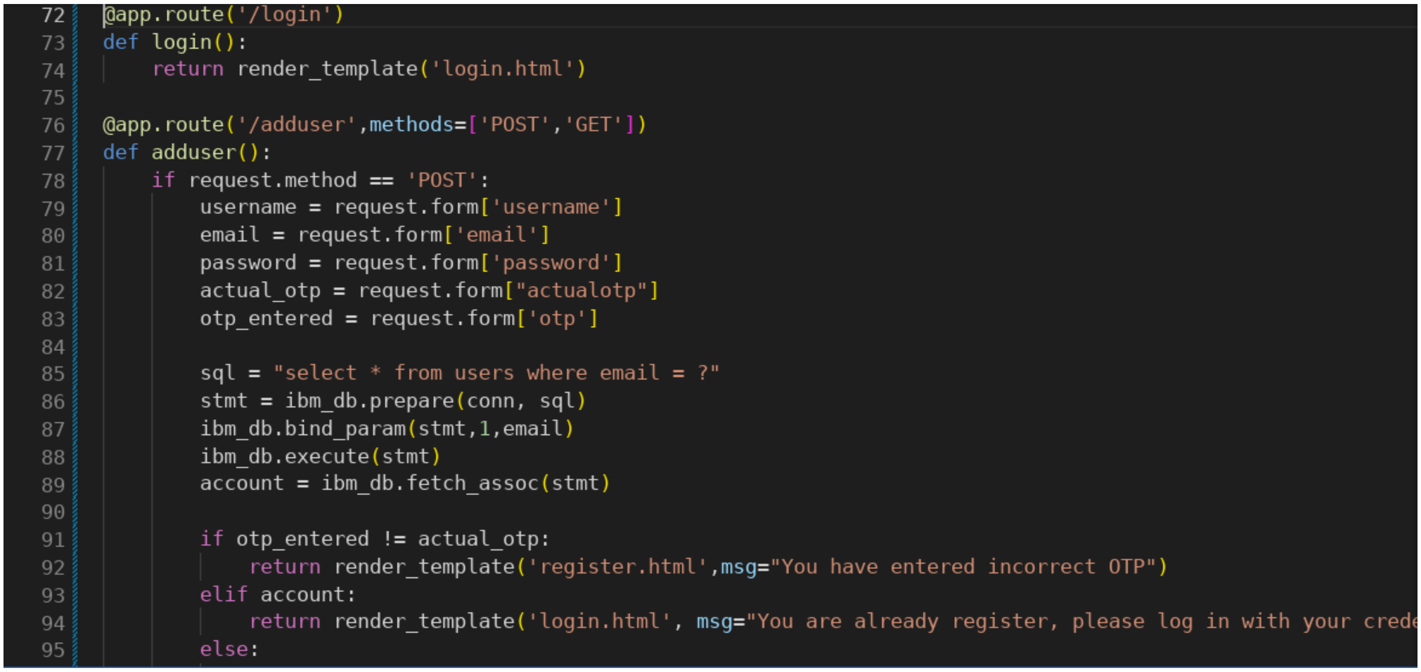
```python
from flask import Flask, render_template, url_for, request, redirect, flash, session
from flask_session import Session
from flask_mail import Mail, Message
import math, random, requests, json

import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.data


app = Flask(__name__)

# configuration of mail
app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'ariharanariharan2001@gmail.com'
app.config['MAIL_PASSWORD'] = 'yqcezzjilojjiema'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)

# configuration of session
app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
Session(app)
```



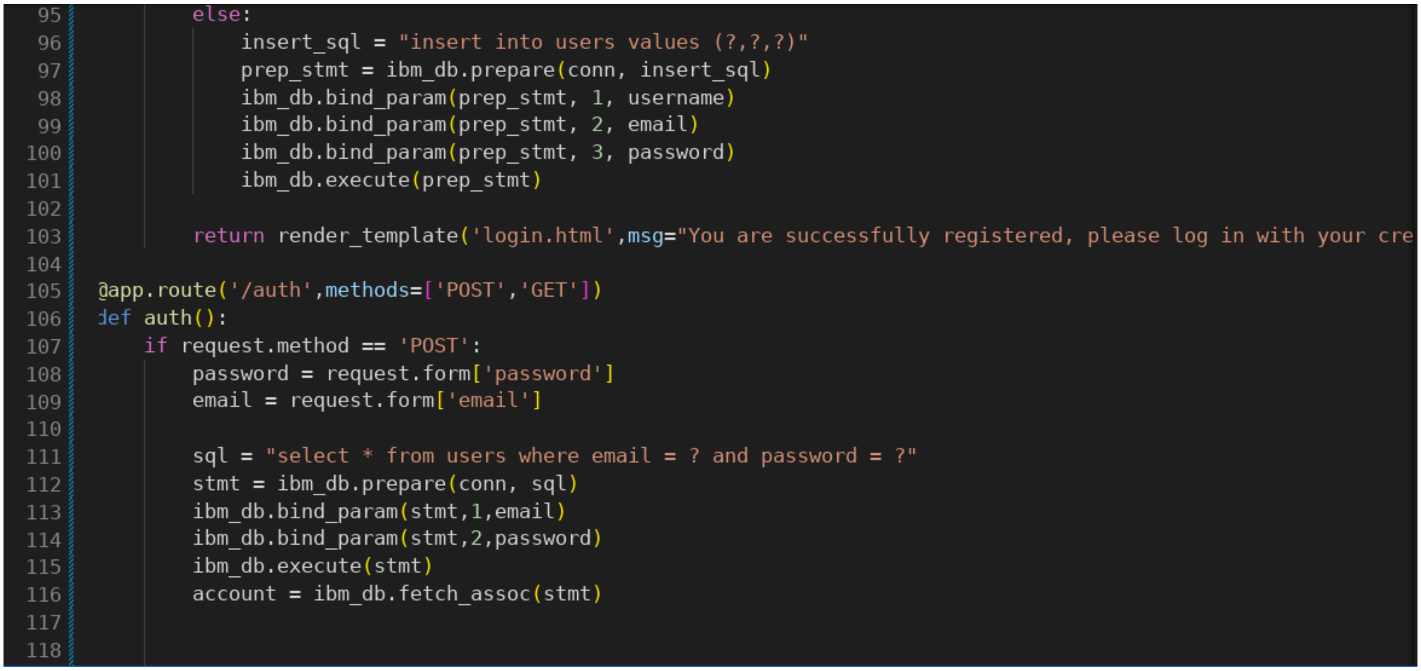
```python
@app.route('/about')
def about():
    return render_template('about.html')

@app.route('/')
@app.route('/home')
def home():
    if not session.get("name"):
        return redirect("/login")
    return render_template('home.html')

@app.route('/logout')
def logout():
    session["name"] = None
    return redirect("/login")

@app.route('/register',methods=['POST','GET'])
def new():
    return render_template('register.html')

@app.route('/otp',methods=['POST','GET'])
def otp():
    username = request.form['username']
    email = request.form['email']
```



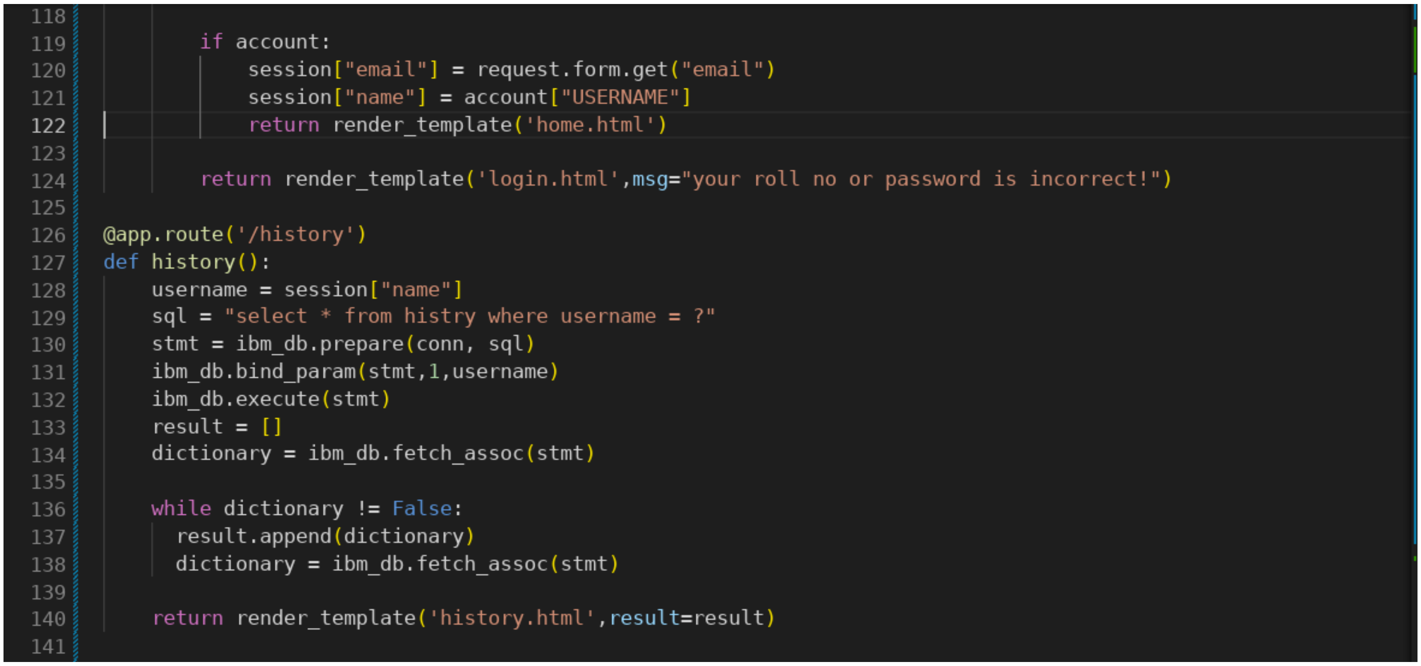
```python
    email = request.form['email']
    password1 = request.form['password1']
    password2 = request.form['password2']
    if password1 != password2:
        return render_template('register.html',msg="Password doesn't match")

    if request.method == 'POST':
        string = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'
        OTP = ""
        length = len(string)
        for i in range(6) :
            OTP += string[math.floor(random.random() * length)]
        msg = Message(
                'Greeting from Nutrition Assistant Application',
                sender = 'ariharanariharan2001@gmail.com',
                recipients = [email]
               )
        msg.body = 'OTP for registering in nutrition assistant app :  ' + OTP
        mail.send(msg)
        return render_template("verify.html",username=username,email=email,password=password1,actualotp=OTP)
    return render_template("verify.html")

@app.route('/login')
def login():
```



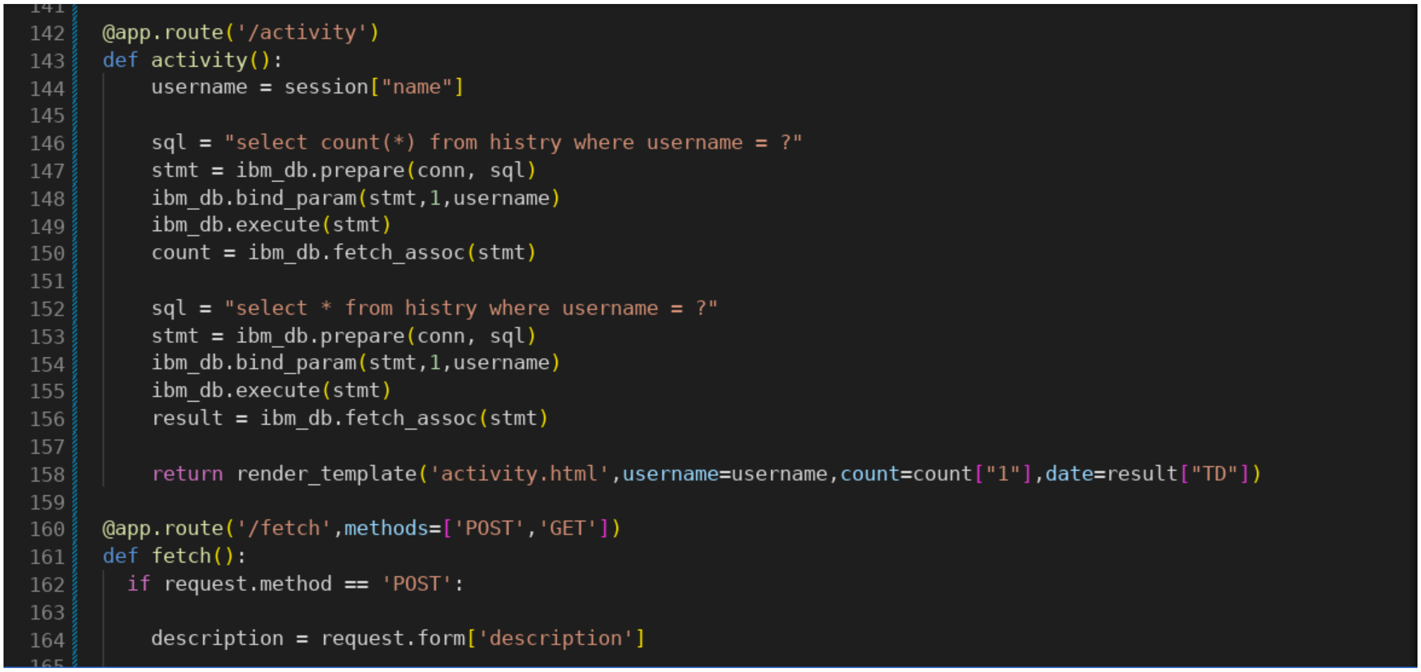
```python
@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/adduser',methods=['POST','GET'])
def adduser():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        actual_otp = request.form["actualotp"]
        otp_entered = request.form['otp']

        sql = "select * from users where email = ?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

        if otp_entered != actual_otp:
            return render_template('register.html',msg="You have entered incorrect OTP")
        elif account:
            return render_template('login.html', msg="You are already register, please log in with your crede
        else:
```



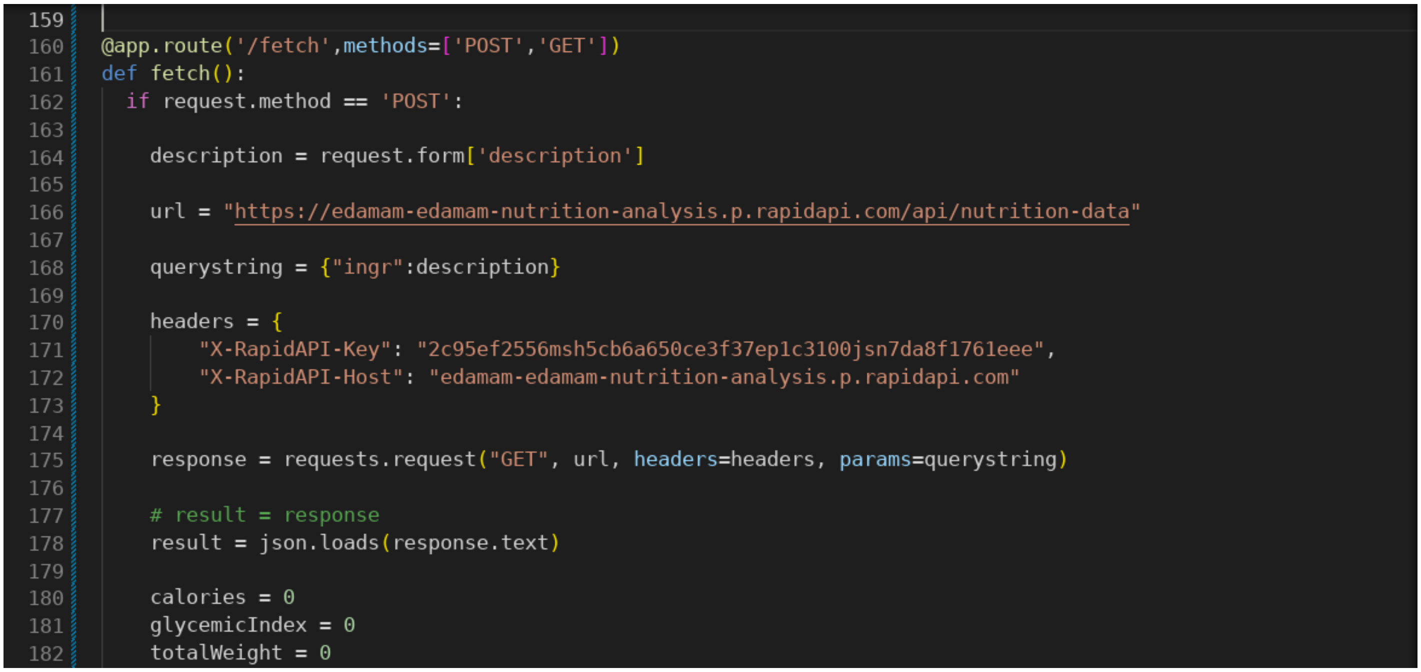
```
        else:
            insert_sql = "insert into users values (?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, password)
            ibm_db.execute(prep_stmt)

        return render_template('login.html',msg="You are successfully registered, please log in with your cre

@app.route('/auth',methods=['POST','GET'])
def auth():
    if request.method == 'POST':
        password = request.form['password']
        email = request.form['email']

        sql = "select * from users where email = ? and password = ?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)

```



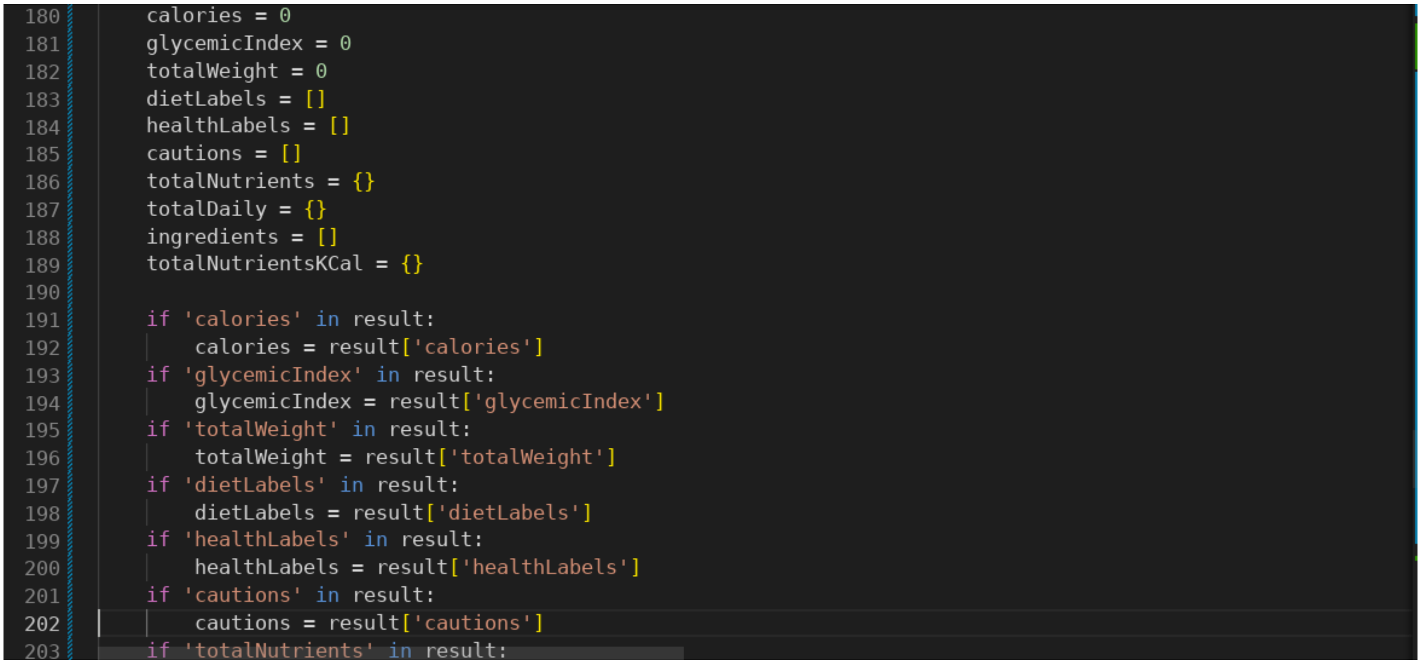
```

        if account:
            session["email"] = request.form.get("email")
            session["name"] = account["USERNAME"]
            return render_template('home.html')

        return render_template('login.html',msg="your roll no or password is incorrect!")

@app.route('/history')
def history():
    username = session["name"]
    sql = "select * from histry where username = ?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    result = []
    dictionary = ibm_db.fetch_assoc(stmt)

    while dictionary != False:
      result.append(dictionary)
      dictionary = ibm_db.fetch_assoc(stmt)

    return render_template('history.html',result=result)

```



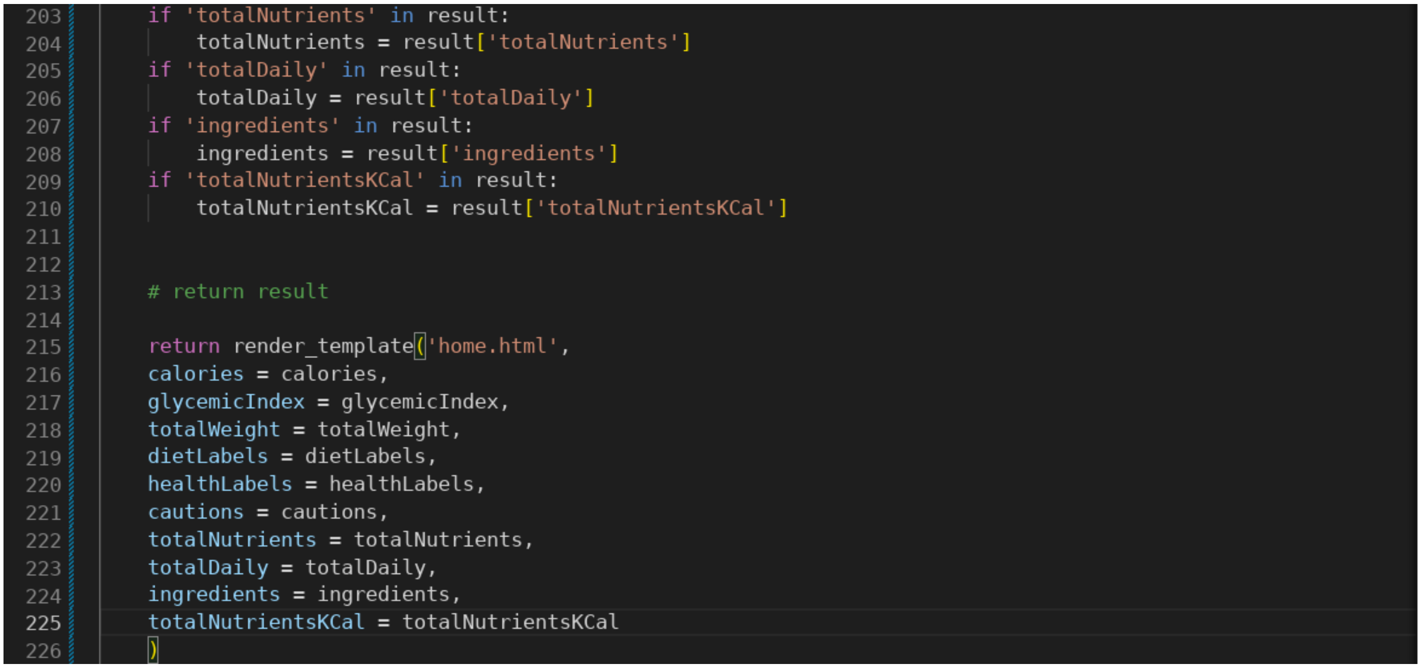
```python
@app.route('/activity')
def activity():
    username = session["name"]

    sql = "select count(*) from histry where username = ?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    count = ibm_db.fetch_assoc(stmt)

    sql = "select * from histry where username = ?"
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    result = ibm_db.fetch_assoc(stmt)

    return render_template('activity.html',username=username,count=count["1"],date=result["TD"])

@app.route('/fetch',methods=['POST','GET'])
def fetch():
  if request.method == 'POST':

    description = request.form['description']
```



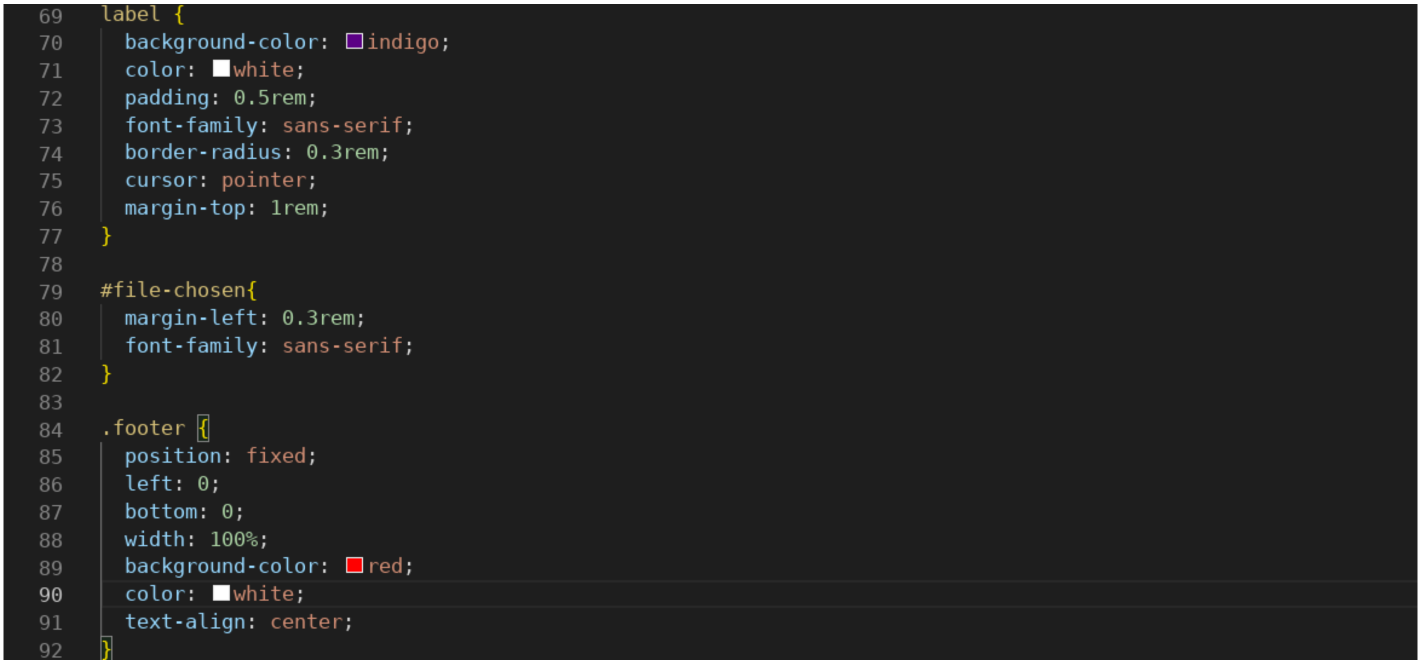
```python

@app.route('/fetch',methods=['POST','GET'])
def fetch():
  if request.method == 'POST':

    description = request.form['description']

    url = "https://edamam-edamam-nutrition-analysis.p.rapidapi.com/api/nutrition-data"

    querystring = {"ingr":description}

    headers = {
        "X-RapidAPI-Key": "2c95ef2556msh5cb6a650ce3f37ep1c3100jsn7da8f1761eee",
        "X-RapidAPI-Host": "edamam-edamam-nutrition-analysis.p.rapidapi.com"
    }

    response = requests.request("GET", url, headers=headers, params=querystring)

    # result = response
    result = json.loads(response.text)

    calories = 0
    glycemicIndex = 0
    totalWeight = 0
```



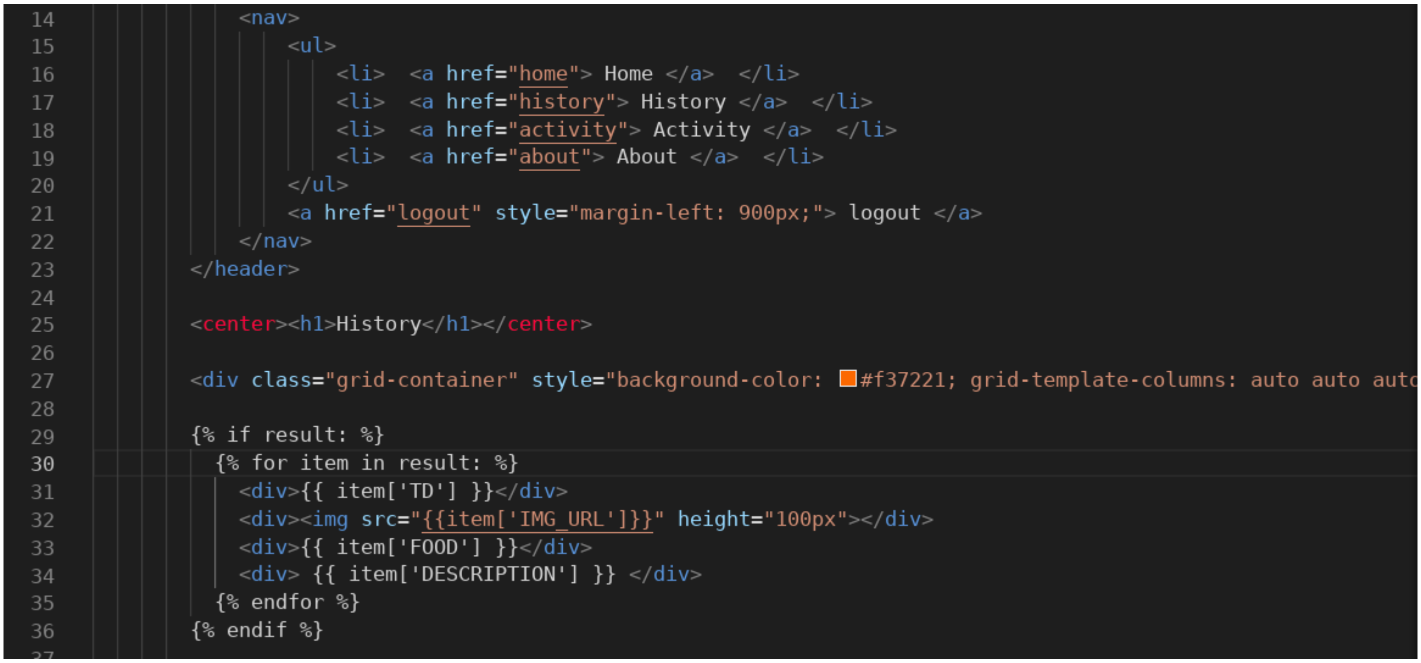
```python
    calories = 0
    glycemicIndex = 0
    totalWeight = 0
    dietLabels = []
    healthLabels = []
    cautions = []
    totalNutrients = {}
    totalDaily = {}
    ingredients = []
    totalNutrientsKCal = {}

    if 'calories' in result:
        calories = result['calories']
    if 'glycemicIndex' in result:
        glycemicIndex = result['glycemicIndex']
    if 'totalWeight' in result:
        totalWeight = result['totalWeight']
    if 'dietLabels' in result:
        dietLabels = result['dietLabels']
    if 'healthLabels' in result:
        healthLabels = result['healthLabels']
    if 'cautions' in result:
        cautions = result['cautions']
    if 'totalNutrients' in result:
```



```python
    if 'totalNutrients' in result:
        totalNutrients = result['totalNutrients']
    if 'totalDaily' in result:
        totalDaily = result['totalDaily']
    if 'ingredients' in result:
        ingredients = result['ingredients']
    if 'totalNutrientsKCal' in result:
        totalNutrientsKCal = result['totalNutrientsKCal']


    # return result

    return render_template('home.html',
    calories = calories,
    glycemicIndex = glycemicIndex,
    totalWeight = totalWeight,
    dietLabels = dietLabels,
    healthLabels = healthLabels,
    cautions = cautions,
    totalNutrients = totalNutrients,
    totalDaily = totalDaily,
    ingredients = ingredients,
    totalNutrientsKCal = totalNutrientsKCal
    )
```

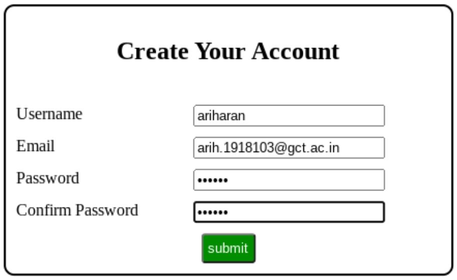


```css
label {
  background-color: indigo;
  color: white;
  padding: 0.5rem;
  font-family: sans-serif;
  border-radius: 0.3rem;
  cursor: pointer;
  margin-top: 1rem;
}

#file-chosen{
  margin-left: 0.3rem;
  font-family: sans-serif;
}

.footer {
  position: fixed;
  left: 0;
  bottom: 0;
  width: 100%;
  background-color: red;
  color: white;
  text-align: center;
}
```



```html
            <nav>
                <ul>
                    <li>  <a href="home"> Home </a>  </li>
                    <li>  <a href="history"> History </a>  </li>
                    <li>  <a href="activity"> Activity </a>  </li>
                    <li>  <a href="about"> About </a>  </li>
                </ul>
                <a href="logout" style="margin-left: 900px;"> logout </a>
            </nav>
        </header>

        <center><h1>History</h1></center>

        <div class="grid-container" style="background-color: #f37221; grid-template-columns: auto auto auto

        {% if result: %}
          {% for item in result: %}
            <div>{{ item['TD'] }}</div>
            <div><img src="{{item['IMG_URL']}}" height="100px"></div>
            <div>{{ item['FOOD'] }}</div>
            <div> {{ item['DESCRIPTION'] }} </div>
          {% endfor %}
        {% endif %}
```

OUTPUT:



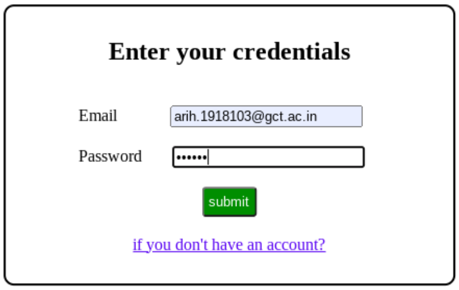
Create Your Account

Username ariharan

Email arih.1918103@gct.ac.in

Password ••••••

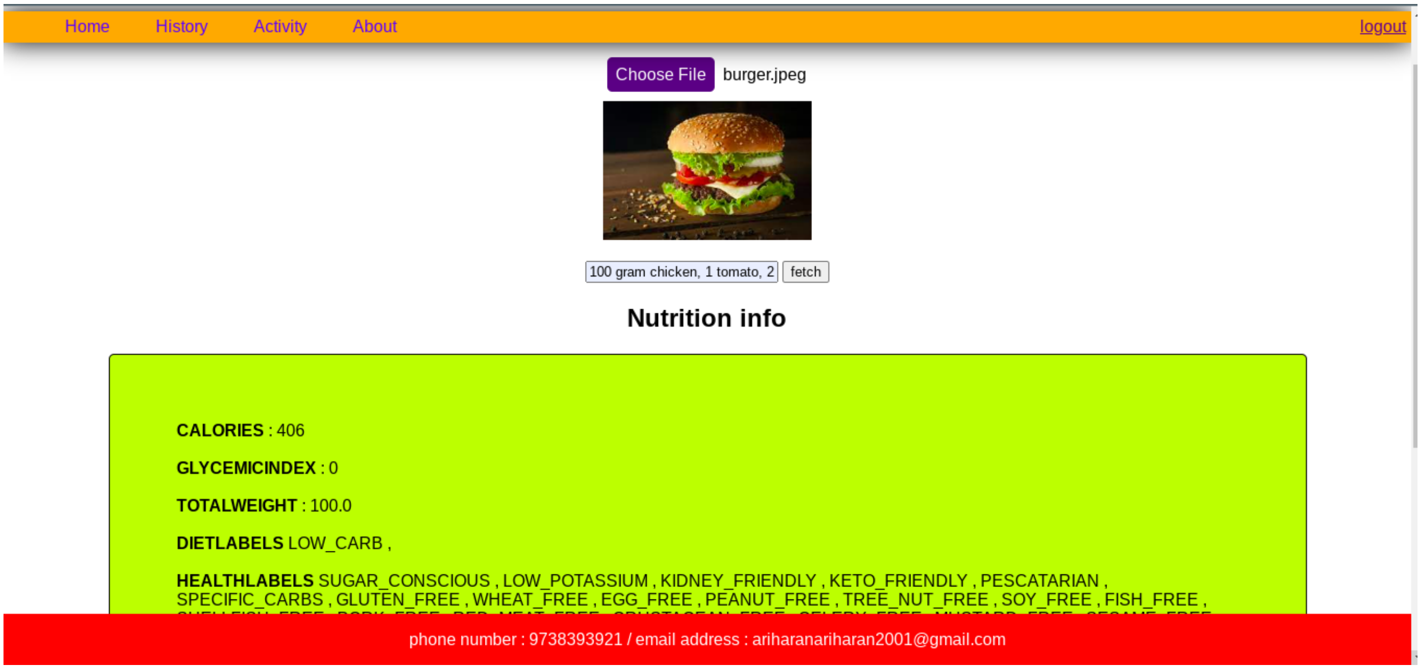
Confirm Password ••••••



Enter your credentials

Email arih.1918103@gct.ac.in

Password ••••••



burger.jpeg

100 gram chicken, 1 tomato, 2 fetch

Nutrition info

CALORIES : 406

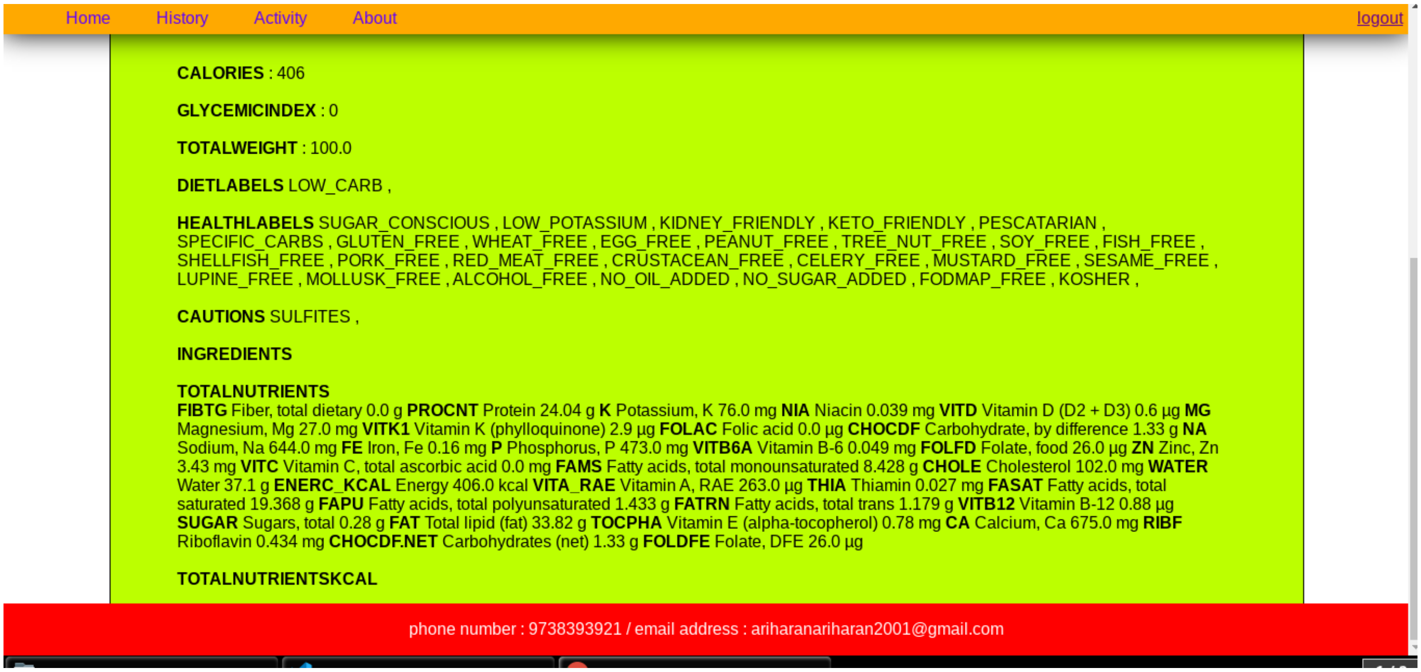
GLYCEMICINDEX : 0

TOTALWEIGHT : 100.0

DIETLABELS LOW_CARB ,

HEALTHLABELS SUGAR_CONSCIOUS , LOW_POTASSIUM , KIDNEY_FRIENDLY , KETO_FRIENDLY , PESCATARIAN , SPECIFIC_CARBS , GLUTEN_FREE , WHEAT_FREE , EGG_FREE , PEANUT_FREE , TREE_NUT_FREE , SOY_FREE , FISH_FREE ,

phone number : 9738393921 / email address : ariharanariharan2001@gmail.com



CALORIES : 406

GLYCEMICINDEX : 0

TOTALWEIGHT : 100.0

DIETLABELS LOW_CARB ,

HEALTHLABELS SUGAR_CONSCIOUS , LOW_POTASSIUM , KIDNEY_FRIENDLY , KETO_FRIENDLY , PESCATARIAN , SPECIFIC_CARBS , GLUTEN_FREE , WHEAT_FREE , EGG_FREE , PEANUT_FREE , TREE_NUT_FREE , SOY_FREE , FISH_FREE , SHELLFISH_FREE , PORK_FREE , RED_MEAT_FREE , CRUSTACEAN_FREE , CELERY_FREE , MUSTARD_FREE , SESAME_FREE , LUPINE_FREE , MOLLUSK_FREE , ALCOHOL_FREE , NO_OIL_ADDED , NO_SUGAR_ADDED , FODMAP_FREE , KOSHER ,

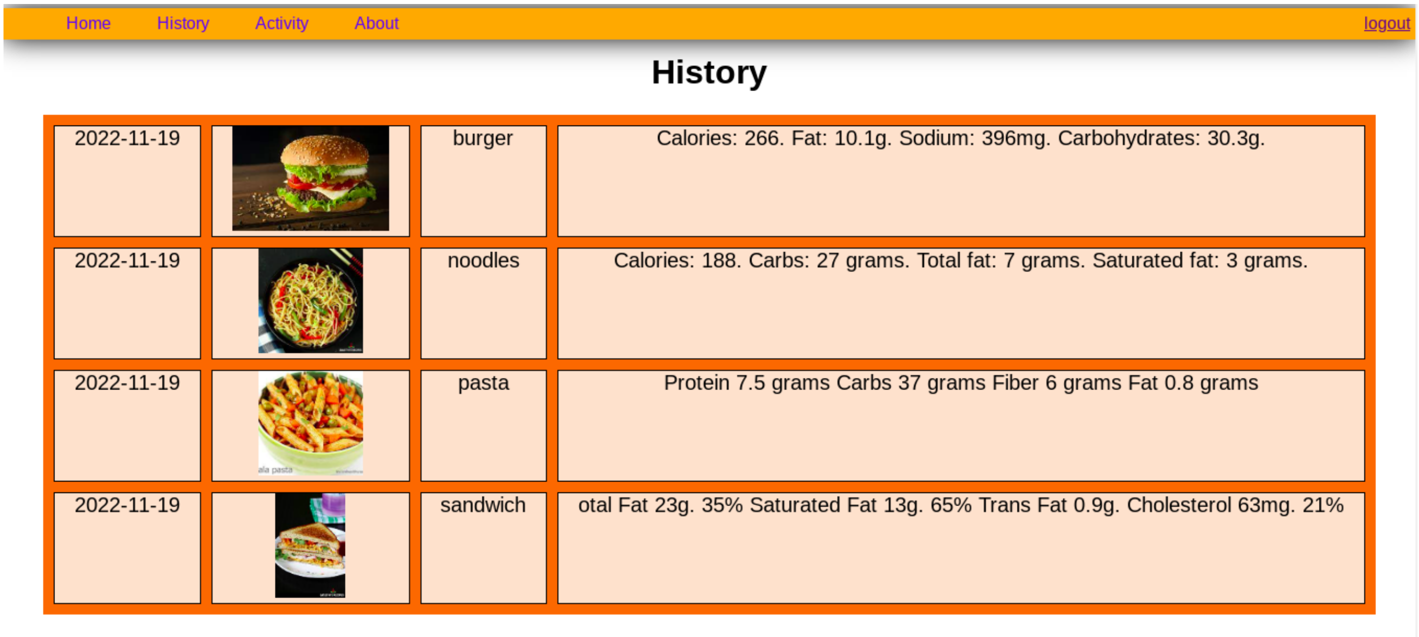
CAUTIONS SULFITES ,

INGREDIENTS

TOTALNUTRIENTS
FIBTG Fiber, total dietary 0.0 g **PROCNT** Protein 24.04 g **K** Potassium, K 76.0 mg **NIA** Niacin 0.039 mg **VITD** Vitamin D (D2 + D3) 0.6 µg **MG** Magnesium, Mg 27.0 mg **VITK1** Vitamin K (phylloquinone) 2.9 µg **FOLAC** Folic acid 0.0 µg **CHOCDF** Carbohydrate, by difference 1.33 g **NA** Sodium, Na 644.0 mg **FE** Iron, Fe 0.16 mg **P** Phosphorus, P 473.0 mg **VITB6A** Vitamin B-6 0.049 mg **FOLFD** Folate, food 26.0 µg **ZN** Zinc, Zn 3.43 mg **VITC** Vitamin C, total ascorbic acid 0.0 mg **FAMS** Fatty acids, total monounsaturated 8.428 g **CHOLE** Cholesterol 102.0 mg **WATER** Water 37.1 g **ENERC_KCAL** Energy 406.0 kcal **VITA_RAE** Vitamin A, RAE 263.0 µg **THIA** Thiamin 0.027 mg **FASAT** Fatty acids, total saturated 19.368 g **FAPU** Fatty acids, total polyunsaturated 1.433 g **FATRN** Fatty acids, total trans 1.179 g **VITB12** Vitamin B-12 0.88 µg **SUGAR** Sugars, total 0.28 g **FAT** Total lipid (fat) 33.82 g **TOCPHA** Vitamin E (alpha-tocopherol) 0.78 mg **CA** Calcium, Ca 675.0 mg **RIBF** Riboflavin 0.434 mg **CHOCDF.NET** Carbohydrates (net) 1.33 g **FOLDFE** Folate, DFE 26.0 µg

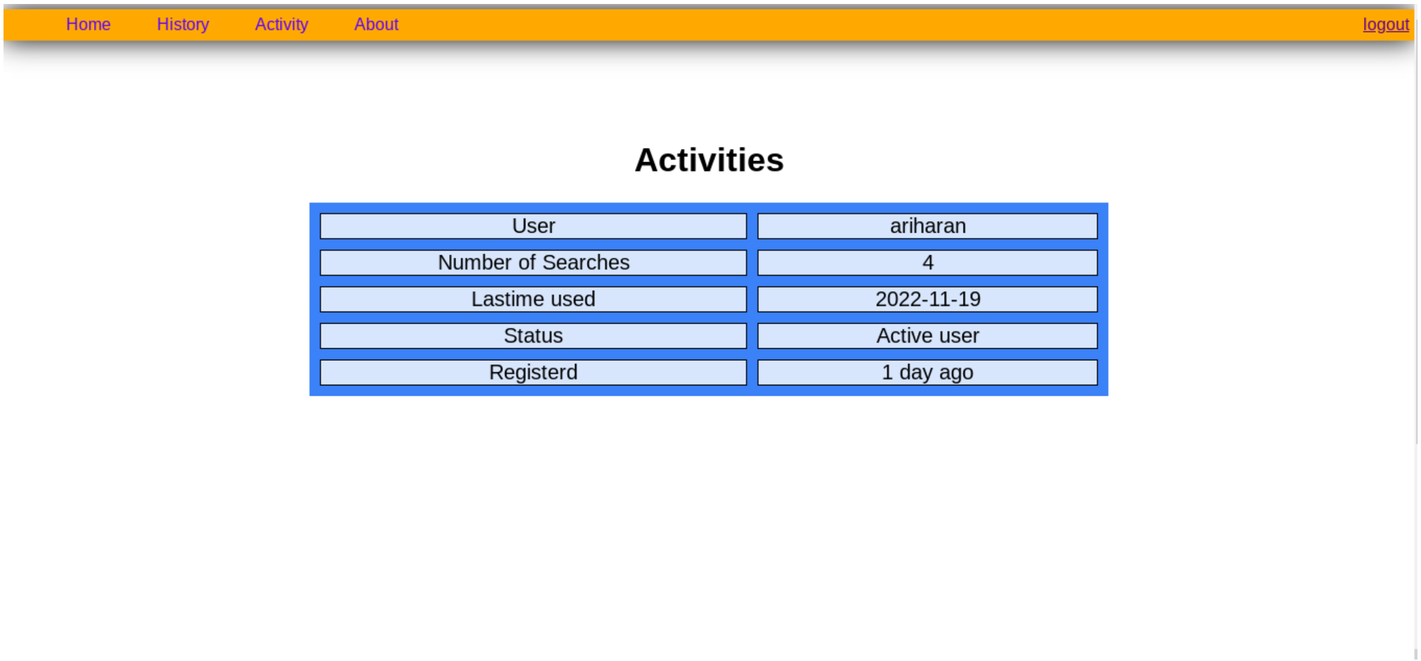
TOTALNUTRIENTSKCAL

phone number : 9738393921 / email address : ariharanariharan2001@gmail.com



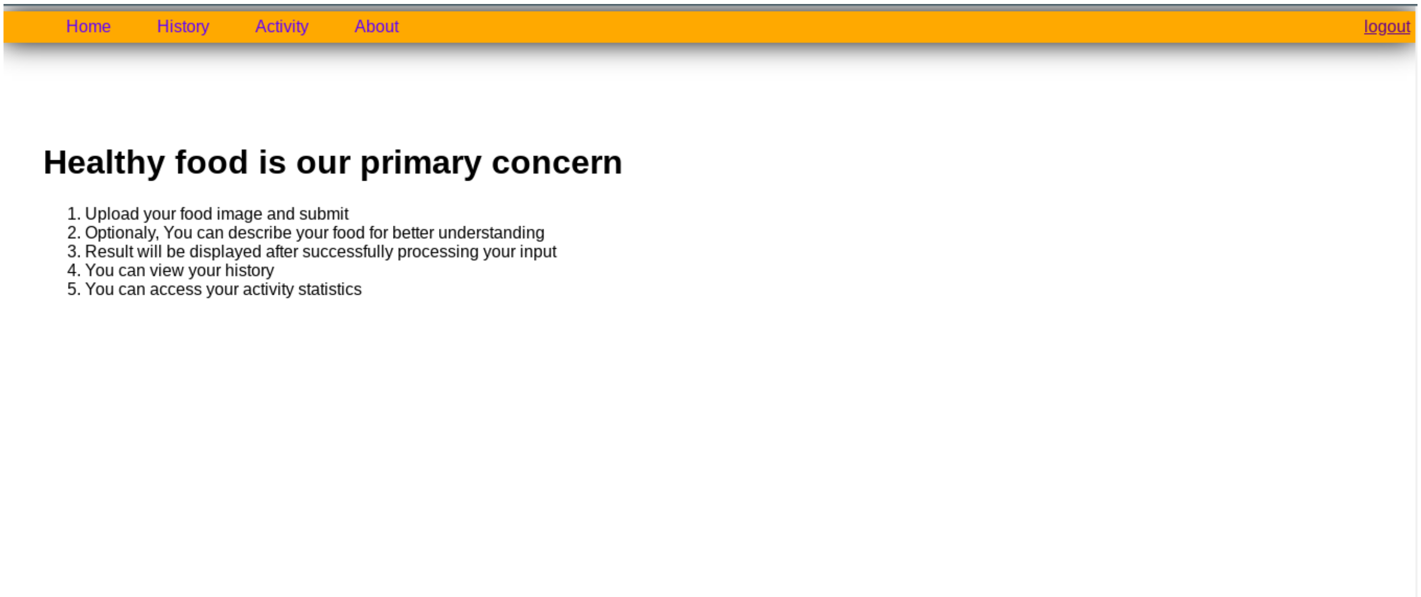
History

2022-11-19		burger	Calories: 266. Fat: 10.1g. Sodium: 396mg. Carbohydrates: 30.3g.
2022-11-19		noodles	Calories: 188. Carbs: 27 grams. Total fat: 7 grams. Saturated fat: 3 grams.
2022-11-19		pasta	Protein 7.5 grams Carbs 37 grams Fiber 6 grams Fat 0.8 grams
2022-11-19		sandwich	otal Fat 23g. 35% Saturated Fat 13g. 65% Trans Fat 0.9g. Cholesterol 63mg. 21%



Activities

User	ariharan
Number of Searches	4
Lastime used	2022-11-19
Status	Active user
Registerd	1 day ago



Healthy food is our primary concern

1. Upload your food image and submit
2. Optionaly, You can describe your food for better understanding
3. Result will be displayed after successfully processing your input
4. You can view your history
5. You can access your activity statistics

Git hub link:

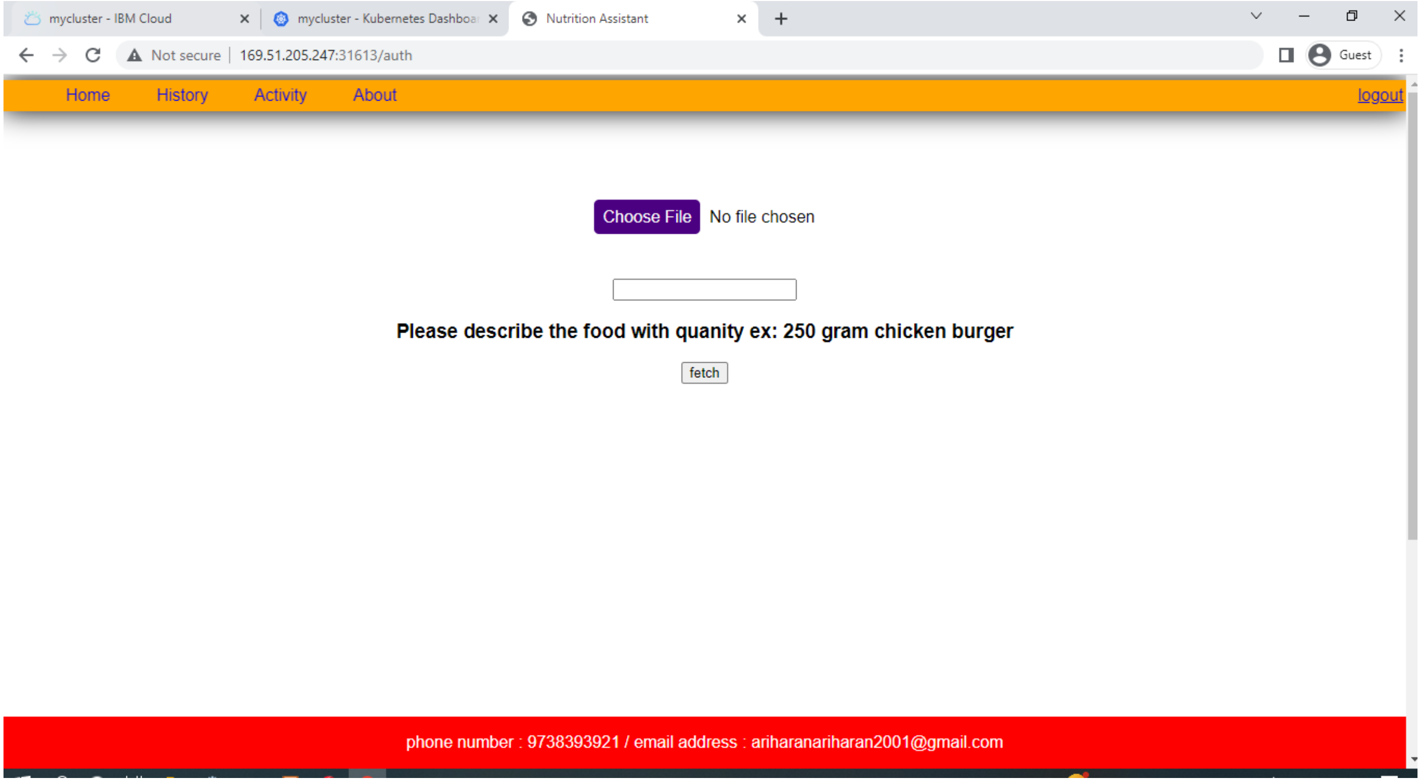
https://github.com/IBM-EPBL/IBM-Project-3284-1658512447

Demo link:

https://drive.google.com/file/d/1Pbd_8zyssV6irhvdxQ4dDgHC1afwy3mD/view?usp=share_link

Public IP:

169.51.205.247:31613



Please describe the food with quanity ex: 250 gram chicken burger

phone number : 9738393921 / email address : ariharanariharan2001@gmail.com