CLOUD APPLICATION DEVELOPMENT

NUTRITION ASSISTANT APPLICATION

PROJECT REPORT

Submitted by:

Ashwin Kumar MM (950019104006)

Ganesan A (950019104012)

Sriram K (950019104045)

Nivendhan C (950019104034)

ANNA UNIVERSITY REGIONAL CAMPUS – TIRUNELVELI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AUGUST 2022 – NOVEMBER 2022

CHAPTER-1

INTRODUCTION

1.1 Project Overview

A Nutrition Assistant is a specialist that uses diagnostic procedures to identify nutrition deficiencies in patients. They work closely with nutritionists and dietitians to improve the well-being of patients through proper nutrition. Nutritionists need to determine their patients' needs through interviewing them and giving them the best meal plans after assessing all risk factors. They must also monitor their progress through follow-ups.

A Nutrition assistant interacts directly with patients to note their habits and lifestyles, enabling them to make informed decisions. They can find work in hospitals, outpatient clinics, rehabilitation centers, schools, health clubs, or assisted living facilities. A successful nutrition assistant should be equipped with nutritional experience, communication skills, and organizational skills.

1.2 Purpose

The purpose of this project to 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.

CHAPTER-2

LITERATURE SURVEY

2.1 Existing Problem

This literature review provides an update on the findings of the research on nutrition content claims which has been published since 2007. The review examines whether consumers may be misled by nutrition content claims, and whether their behaviour may be influenced by them.

● In India, because of unhealthy food, most young people are dying due to obesity, type 2 diabetes, heart disease, high blood pressure, and stroke.

● Nowadays new dietary assessment and nutrition analysis tools are available.

● Nutritional analysis is the process of determining the nutritional content of food. This helps the fitness enthusiast to track and monitor their intake nutrition and calorie intake.
Social Impact.

● People can do weight managements, strengthen their bones and muscles, manage chronic health conditions & disabilities.
Business Model/Impact.

● Social media is the best way to spread the word about our application. And with the influencers we can attract the normal people.

● Clustering and targeting the fitness people with the help of local gyms.

2.2 References

https://www.healthifyme.com/in/

https://analyticsindiamag.com/

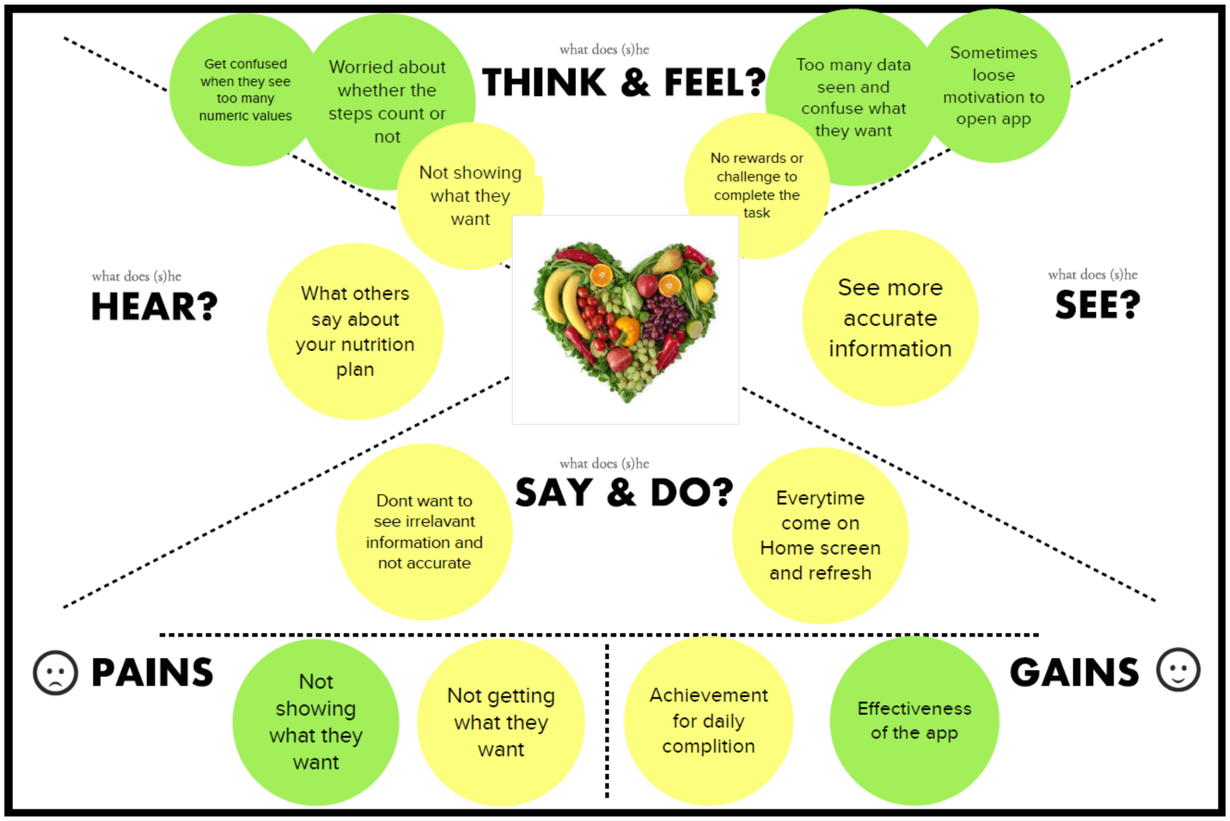
2.3 Problem Statement Definition

A variety of medical problems can affect your appetite. Your illness, medicines or surgery can cause these problems. Many people become frustrated when they know they need to eat to get well but they aren't hungry, or when they gain weight because they are fatigued and unable to exercise. Each of the following sections describes a nutritional problem and suggests possible solutions. Not all solutions will work for everyone.

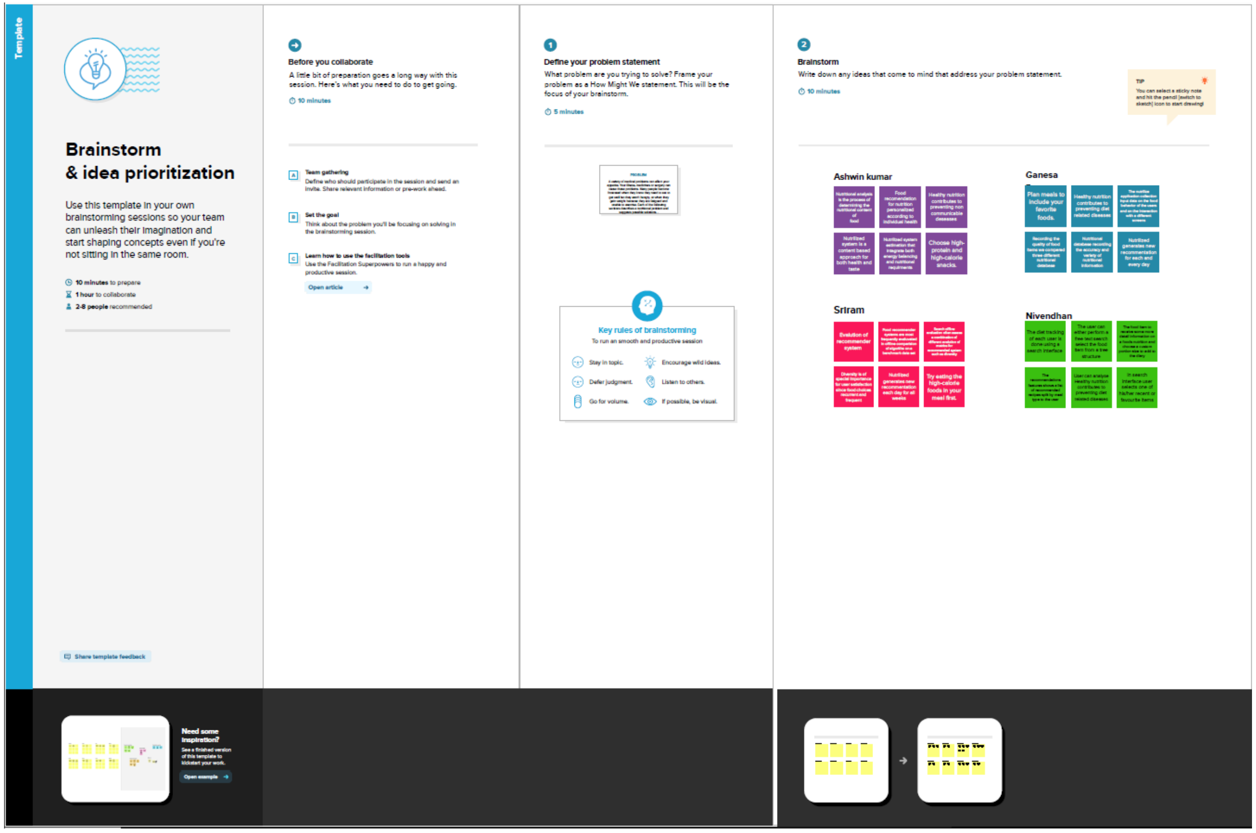
CHAPTER-3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas


what does (s)he
THINK & FEEL?
Get confused when they see too many numeric values
Worried about whether the steps count or not
Not showing what they want
No rewards or challenge to complete the task
Too many data seen and confuse what they want
Sometimes loose motivation to open app
what does (s)he
HEAR?
What others say about your nutrition plan
what does (s)he
SEE?
See more accurate information
what does (s)he
SAY & DO?
Dont want to see irrelavant information and not accurate
Everytime come on Home screen and refresh
PAINS
Not showing what they want
Not getting what they want
Achievement for daily complition
Effectiveness of the app
GAINS

3.2 Ideation & Brainstorming


Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Share template feedback

Need some inspiration? See a finished version of this template to kickstart your work. Open example

Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.
10 minutes

A **Team gathering** Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal** Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools** Use the Facilitation Superpowers to run a happy and productive session.
Open article

1 **Define your problem statement**
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes

Key rules of brainstorming
To run an smooth and productive session
- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2 **Brainstorm**
Write down any ideas that come to mind that address your problem statement.
10 minutes

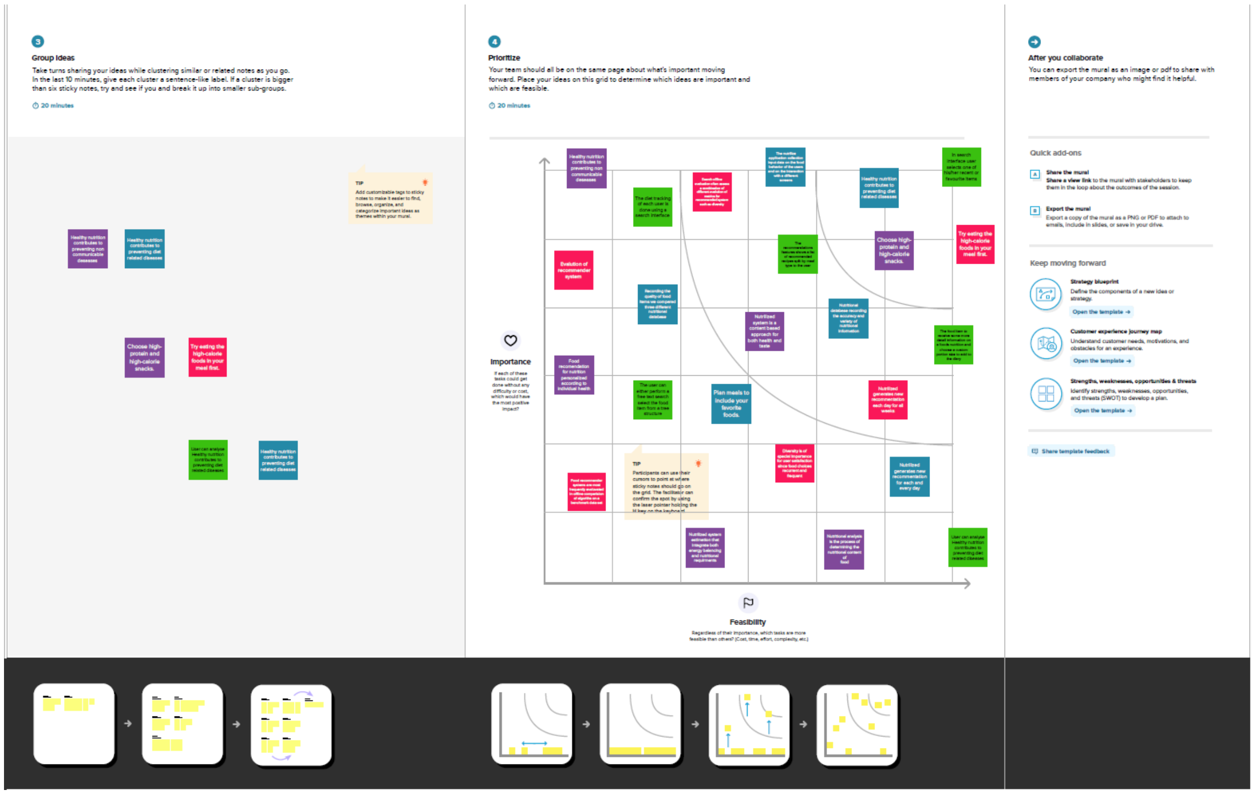
TIP You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Ashwin kumar
- Nutritional analysis is the process of determining the nutritional content of food
- Food recomendation for nutrition personalized according to individual health
- Healthy nutrition contributes to preventing non communicable diseases
- Nutrilized system is a content based approach for both health and taste
- Nutrilized system estimation that integrate both energy balancing and nutritional requirements
- Choose high-protein and high-calorie snacks.

Ganesa
- Plan meals to include your favorite foods.
- Healthy nutrition contributes to preventing diet related diseases
- The nutrition application collection input data on the food behavior of the users and on the interaction with a different screens
- Recording the quality of food items we compared three different nutritional database
- Nutritional database recording the accuracy and variety of nutritional information
- Nutrilized generates new recommentation for each and every day

Sriram
- Evalution of recommender system
- Food recommender systems are most frequently evaluvated in offline comparision of algorithm on a benchmark data set
- Search offline evaluation often assess a combination of different evalution of metrics for recommended system such as diversity
- Diversity is of special importance for user satisfaction since food choices recurrent and frequent
- Nutrilized generates new recommentation each day for all weeks
- Try eating the high-calorie foods in your meal first.

Nivendhan
- The diet tracking of each user is done using a search interface
- The user can either perform a free text search select the food item from a tree structure
- The food item to receive some more detail information on a foods nutrition and choose a custom portion size to add to the diary
- The recommendations features shows a list of recommended recipes split by meal type to the user
- User can analyse Healthy nutrition contributes to preventing diet related diseases
- In search interface user selects one of his/her recent or favourite items


3 **Group ideas**
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes

TIP Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

- Healthy nutrition contributes to preventing non communicable diseases
- Healthy nutrition contributes to preventing diet related diseases
- Choose high-protein and high-calorie snacks.
- Try eating the high-calorie foods in your meal first.
- User can analyse Healthy nutrition contributes to preventing diet related diseases
- Healthy nutrition contributes to preventing diet related diseases

4 **Prioritize**
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes

Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

TIP Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.

After you collaborate
You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural** Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural** Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint Define the components of a new idea or strategy. Open the template →

Customer experience journey map Understand customer needs, motivations, and obstacles for an experience. Open the template →

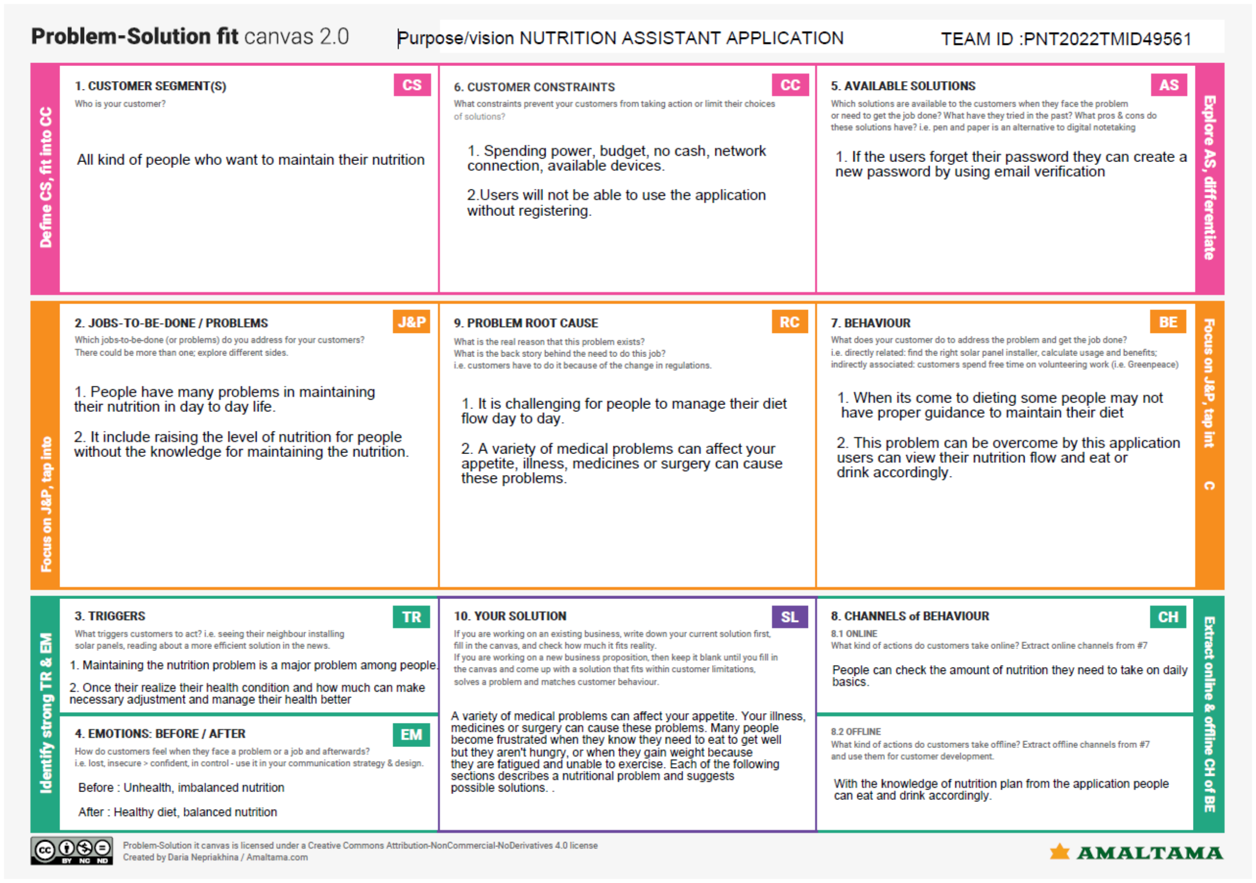
Strengths, weaknesses, opportunities & threats Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan. Open the template →

Share template feedback

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	If your appetite and taste have been affected by Illness, medications or others health issues, you may have trouble eating and getting proper nutrition. These changes can affect your overall health.
2.	Idea / Solution description	Eat smaller meals and snacks more frequently. Eating six or seven times a day might be more easily tolerated than eating the same amount of food in three meals. Avoid non nutrition beverages such as black coffee and tea instead of milk and juices. Try to eat more protein and fiber foods and less simple sugars. Walk in light activity to simulate your appetite.
3.	Novelty / Uniqueness	This application provides link which contains tasty and healthy food recipe.
4.	Social Impact / Customer Satisfaction	It will help people with providing proper nutrition and helps in maintaining a healthy lifestyle.
5.	Business Model (Revenue Model)	Social media is the best way to spread the word about our application. And with the influencers we can attract the normal people. Subscription or the membership will have extra benefits.
6.	Scalability of the Solution	This application can maintain many users and assign a separate assistant for subscribed members.

3.4 Problem Solution Fit



Problem-Solution fit canvas 2.0

Purpose/vision NUTRITION ASSISTANT APPLICATION

TEAM ID :PNT2022TMID49561

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

All kind of people who want to maintain their nutrition

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

1. Spending power, budget, no cash, network connection, available devices.

2.Users will not be able to use the application without registering.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

1. If the users forget their password they can create a new password by using email verification

Explore AS, differentiate

Focus on J&P, tap into

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

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

1. People have many problems in maintaining their nutrition in day to day life.

2. It include raising the level of nutrition for people without the knowledge for maintaining the nutrition.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.

1. It is challenging for people to manage their diet flow day to day.

2. A variety of medical problems can affect your appetite, illness, medicines or surgery can cause these problems.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

1. When its come to dieting some people may not have proper guidance to maintain their diet

2. This problem can be overcome by this application users can view their nutrition flow and eat or drink accordingly.

Focus on J&P, tap int

C

Identify strong TR & EM

3. TRIGGERS TR

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

1. Maintaining the nutrition problem is a major problem among people.

2. Once their realize their health condition and how much can make necessary adjustment and manage their health better

4. EMOTIONS: BEFORE / AFTER EM

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

Before : Unhealth, imbalanced nutrition

After : Healthy diet, balanced nutrition

10. YOUR SOLUTION SL

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

A variety of medical problems can affect your appetite. Your illness, medicines or surgery can cause these problems. Many people become frustrated when they know they need to eat to get well but they aren't hungry, or when they gain weight because they are fatigued and unable to exercise. Each of the following sections describes a nutritional problem and suggests possible solutions. .

8. CHANNELS of BEHAVIOUR CH

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

People can check the amount of nutrition they need to take on daily basics.

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

With the knowledge of nutrition plan from the application people can eat and drink accordingly.

Extract online & offline CH of BE

AMALTAMA

CHAPTER-4
REQUIREMENT ANALYSIS

4.1 Functional Requirement

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Register	Registration through Email
FR-2	E mail Alert	Confirmation via Email
FR-3	Enter OTP	Confirmation OTP via Email
FR-4	User Login	Login through Login Form
FR-5	User Profile	Shown user information
FR-6	Update User Details	Update user details
FR-7	Clarifai-AI	Upload image and it shown nutrition values
FR-8	User logout	User can redirected to home page

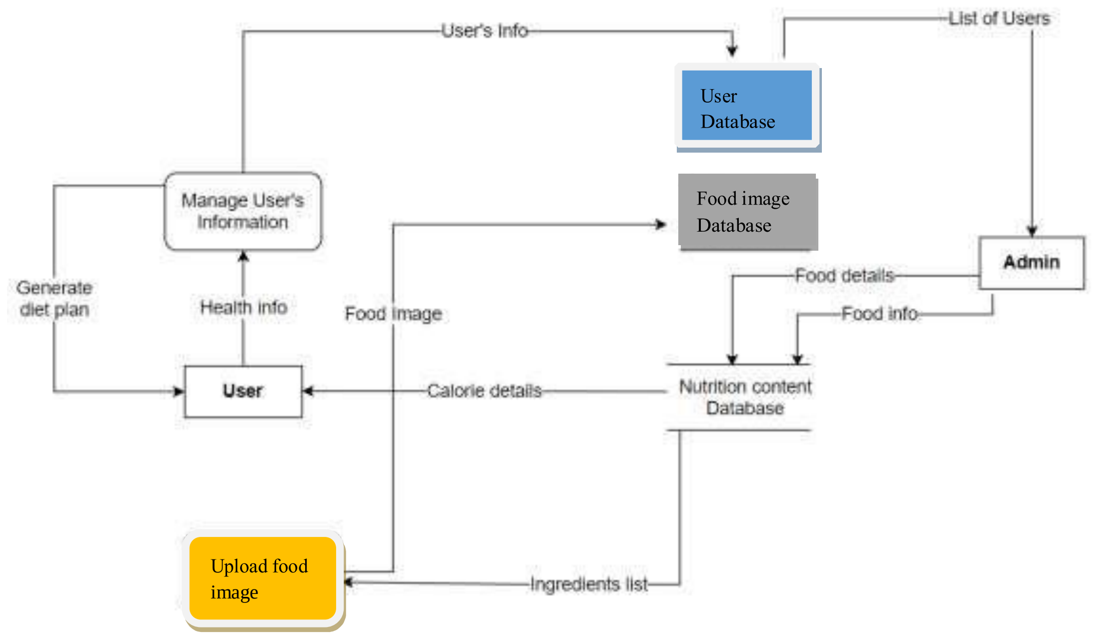
4.2 Non-Functional Requirements

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	User can recognize their nutrition value by their uploaded picture Which helps to understand their nutrition details in easy manner.
NFR-2	**Security**	We only store the information needed to save user. Application also has a security feature that lets users set a password to access their account.
NFR-3	**Reliability**	The database update process can rollback to all related details in case of problem arise in updating
NFR-4	**Performance**	The application can perform well user can experience the fast while using the application
NFR-5	**Availability**	This application could provide better access to improve user
NFR-6	**Scalability**	This application can able to with stand many number of users

CHAPTER-5

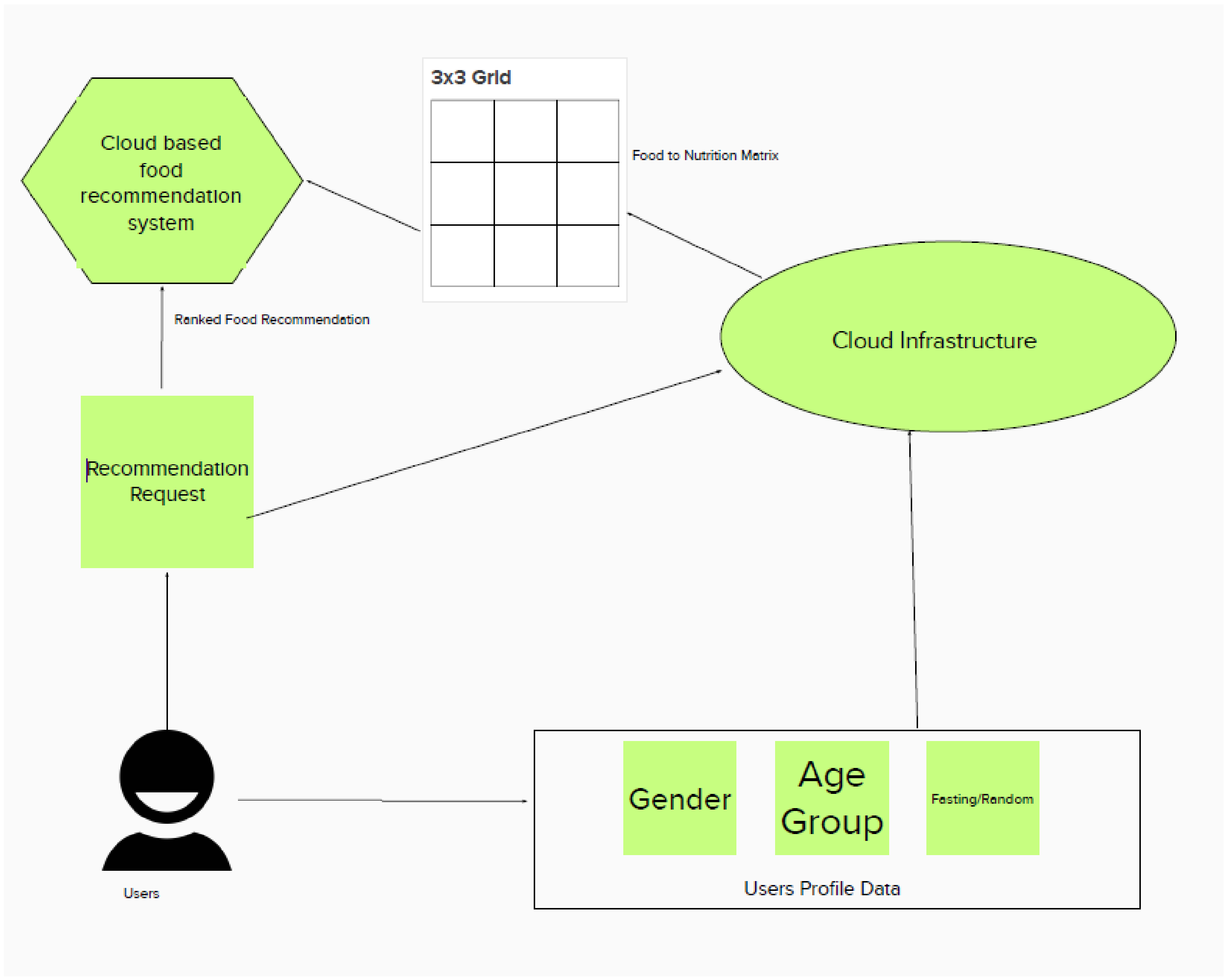
PROJECT DESIGN

5.1 Data Flow Diagram

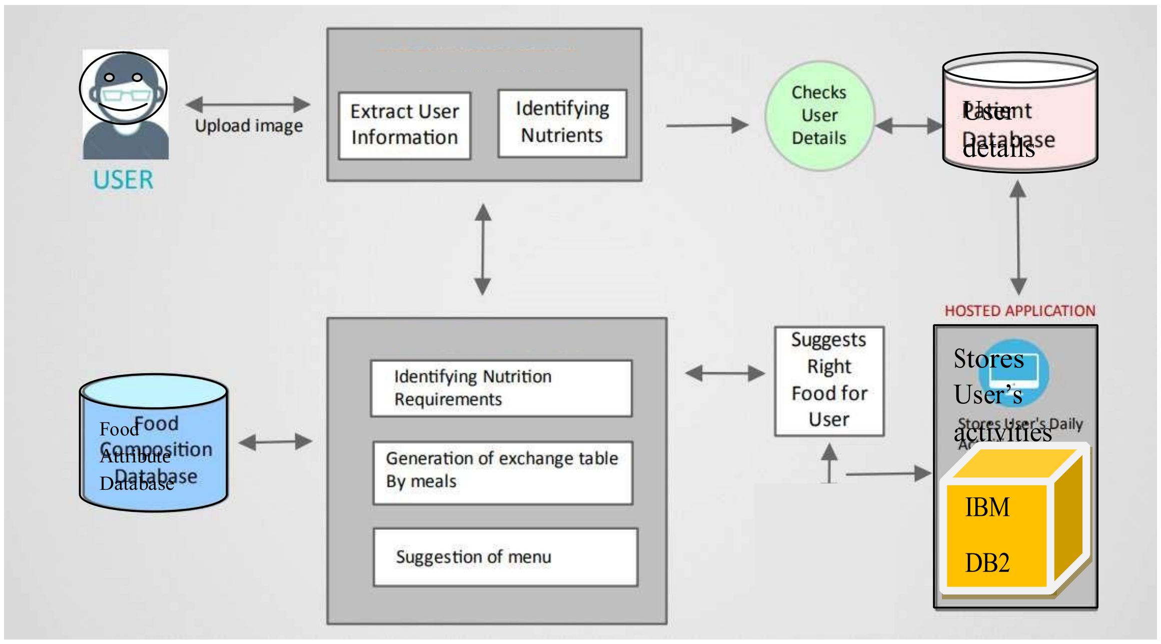


5.2 Solution & Technical Architecture

Solution Architecture



Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	As a user, I can login to the application by entering E-mail and password	I can access my user profile	High	Sprint-3
	Profile Update	USN-4	As a user, I have to enter my height, weight, gender and blood group details	I can access my user update profile	High	Sprint-2
	Clarifai-AI	USN-5	As a user, I can upload or capture live image of the meal	I can Access my nutritional value	High	Sprint-4
	Maintain the application	USN-6	Maintaining detail for user	Admin maintanance	High	Sprint-4

CHAPTER-6
PROJECT PLANNING AND SCHEDULING

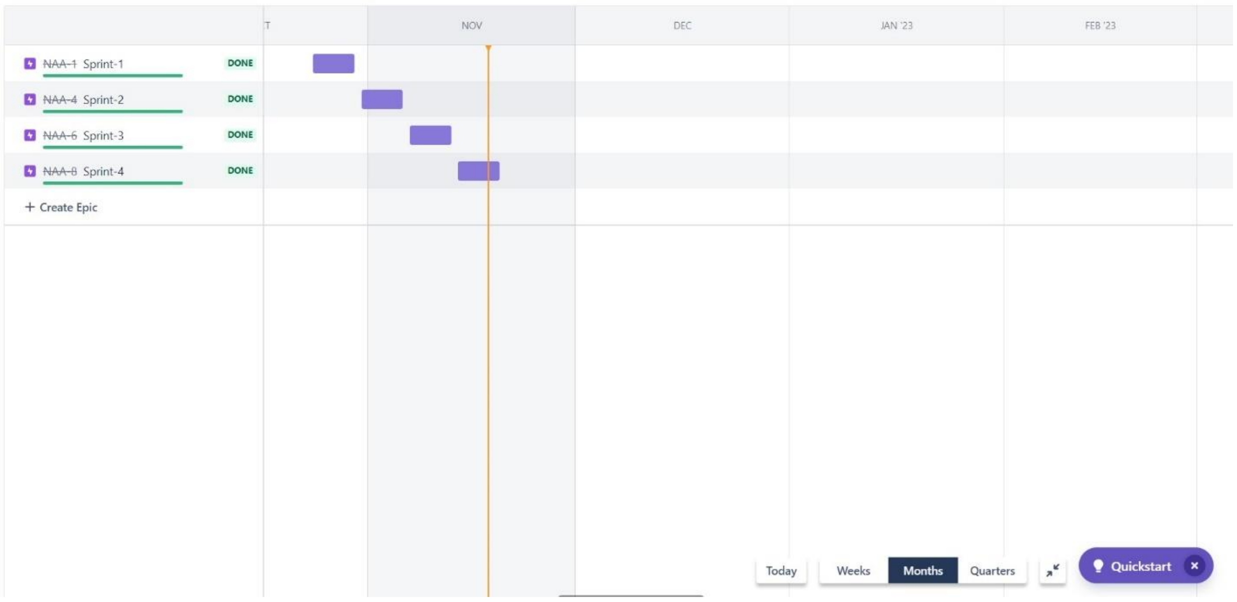
6.1 Sprint Planning and Estimation

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	ASHWIN KUMAR MM
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	ASHWIN KUMAR MM.
Sprint-2	Profile Update	USN-3	As a user, I have to enter my height, weight and daily activity details.	2	high	SRIRAM K
Sprint-3	Login	USN-4	As a user, I can login to the application by entering E-mail and password	2	high	SRIRAM K
Sprint-4	dashboard	USN-5	As a user, I can upload or capture live image of the meal	1	High	NIVENDHAN C
Sprint-4		USN-6	As a user, I can track my daily calorie intake	1	medium	GANESAN A
Sprint-4	Maintain the application	USN-7	Maintaining detail for user	1	high	GANESAN A

6.2 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	7	29 OCT 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	5	05 NOV 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	8	12 NOV 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	5	19 NOV 2022

6.3 Reports From JIRA



CHAPTER-7
CODING & SOLUTIONING

Python code:

nutrition.py

```
from flask import Flask, render_template, request, redirect, url_for, session, flash
import ibm_db
import re
import requests
from random import *
from clarifai_grpc.grpc.api import service_pb2, resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc
from flask_mail import Mail, Message
import os
from flask_mail import Mail, Message
app = Flask(__name__)

mail = Mail(app) # instantiate the mail class
# configuration of mail
app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'nassistant.gans@gmail.com'
app.config['MAIL_PASSWORD'] = 'ddlomuragdcdyojh'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)
otp = randint(000000,999999)

from clarifai_setup import (
    DOG_IMAGE_URL,
    GENERAL_MODEL_ID,
    NON_EXISTING_IMAGE_URL,
    RED_TRUCK_IMAGE_FILE_PATH,
    both_channels,
    metadata,
    raise_on_failure,
    post_model_outputs_and_maybe_allow_retries,
)

def test_predict_image_url():
    stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())

    req = service_pb2.PostModelOutputsRequest(
        model_id=GENERAL_MODEL_ID,
        inputs=[
            resources_pb2.Input(
                data=resources_pb2.Data(image=resources_pb2.Image(url=DOG_IMAG
E_URL))
            )
        ],
    )

        response = post_model_outputs_and_maybe_allow_retries(stub, req,
metadata=metadata())
    print(response)
    raise_on_failure(response)

    assert len(response.outputs[0].data.concepts) > 0

app.secret_key = 'a'

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-
9991-
629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30119;S
ecurity=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=lbs14903;PW
D=1N4walQ5ywwiwP7c;",",")

picsfolder = os.path.join('static','pics')
app.config['UPLOAD_FOLDER']=picsfolder

@app.route('/')

@app.route('/homepage')
def homepage():
    icon = os.path.join(app.config['UPLOAD_FOLDER'],'icon.gif')
    return render_template('homepage.html',user_image=icon)

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

```
def about():
    icon = os.path.join(app.config['UPLOAD_FOLDER'],'icon.gif')
    return render_template('about.html',user_image=icon)



@app.route('/login', methods =['GET', 'POST'])
def login():
    msg=''
    if request.method=='POST' and 'username' in request.form and 'passwords' in request.form:
        username = request.form['username']
        passwords = request.form['passwords']
        stmt = ibm_db.prepare(conn,'SELECT * FROM appuser WHERE username = ? AND passwords = ?')
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,passwords)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        if account:
            session['loggedin'] = True
            session['username'] = account['USERNAME']
            msg='Login successful'
            return redirect(url_for('userprofile'))
        else:
            msg='Incorrect username/password'
    return render_template('login.html',msg=msg)


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

```
def logout():
    if 'id' in session:
        session.pop('id',None)
        session.pop('username',None)
        session.pop('passwords',None)
    return redirect(url_for('homepage'))

@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        fullname = request.form['fullname']
        email = request.form['email']
        passwords = request.form['passwords']
        cpassword = request.form['cpassword']
        stmt = ibm_db.prepare(conn,'SELECT * FROM appuser WHERE username =
?')
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'Username must contain only characters and numbers !'
        elif not username or not passwords or not email:
            msg = 'Please fill out the form !'
        else:
                prep_stmt = ibm_db.prepare(conn,"INSERT INTO appuser(username,
fullname, email, passwords, cpassword) VALUES(?, ?, ?, ?, ?)")
            ibm_db.bind_param(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, fullname)
            ibm_db.bind_param(prep_stmt, 3, email)
            ibm_db.bind_param(prep_stmt, 4, passwords)
            ibm_db.bind_param(prep_stmt, 5, cpassword)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully registered !'
            return render_template('email.html')
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('registration.html', msg = msg)


@app.route('/userprofile', methods =['GET', 'POST'])
def userprofile():
    if 'username' in session:
        username = session['username']
        stmt = ibm_db.prepare(conn, 'SELECT * FROM appuser WHERE username =
?')
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        acc = ibm_db.fetch_tuple(stmt)
        return render_template('userprofile.html',username = acc[1], fullname = acc[2],
email = acc[3],)
```

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


@app.route('/updateprofile', methods =['GET', 'POST'])
def updateprofile():
    msg = ''
    if request.method == 'POST':
            username=request.form["username"]
            height = request.form['height']
            weight = request.form['weight']
            gender = request.form['gender']
            blood = request.form['blood']
               prep_stmt = ibm_db.prepare(conn,"INSERT INTO userdetail(username,
height, weight, gender, blood) VALUES(?, ?, ?, ?, ?)")
            ibm_db.bind_param(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, height)
            ibm_db.bind_param(prep_stmt, 3, weight)
            ibm_db.bind_param(prep_stmt, 4, gender)
            ibm_db.bind_param(prep_stmt, 5, blood)
            ibm_db.execute(prep_stmt)
            return redirect(url_for('detail'))
    return render_template('updateprofile.html')


@app.route('/detail', methods =['GET', 'POST'])
def detail():
    if 'username' in session:
        username = session['username']
        stmt = ibm_db.prepare(conn, 'SELECT * FROM  userdetail WHERE username = ?')
        ibm_db.bind_param(stmt, 1,username)
        ibm_db.execute(stmt)
        acc = ibm_db.fetch_tuple(stmt)
        return render_template('detail.html',height = acc[2], weight = acc[3], gender = acc[4], blood = acc[5])
    return render_template('detail.html')



@app.route('/window', methods=['POST', 'GET'])
def window():

  # Calorie Ninja
    url = "https://calorieninjas.p.rapidapi.com/v1/nutrition"

    headers = {
        "X-RapidAPI-Key": "aa95b88b45mshe4394a422ce8c48p13a698jsn9d8eb019e144",
        "X-RapidAPI-Host": "calorieninjas.p.rapidapi.com"
    }

    if request.method == 'POST':
        foodname = request.form['foodname']

        querystring = {"query": foodname}
        response = requests.request(
            "GET", url, headers=headers, params=querystring)
        return response.text

    return render_template('window.html')

@app.route('/window', methods=['POST', 'GET'])
def clarifai():
    if request.files.get('image'):
        image = request.files['image'].stream.read()
        stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())

        CLARIFAI_API_KEY = "04fe7a95051541789ba44a08eaa5722e"
        APPLICATION_ID = "Nutrition_Assistant1"

        # Authenticate

        # image = '/home/bala/Desktop/Images/foodsample.jpeg'

        metadata = (("authorization", f"Key {CLARIFAI_API_KEY}"),)

        with open(image, "rb") as f:
            file_bytes = f.read()

        request = service_pb2.PostModelOutputsRequest(
            model_id='9504135848be0dd2c39bdab0002f78e9',
            inputs=[
                resources_pb2.Input(
                    data=resources_pb2.Data(
                        image=resources_pb2.Image(
                            base64=file_bytes
                        )
                    )
                )
            ])
        response = stub.PostModelOutputs(request, metadata=metadata)

        if response.status.code != status_code_pb2.SUCCESS:
            raise Exception("Request failed, status code: " +
                            str(response.status.code))

        for concept in response.outputs[0].data.concepts:
            print('%12s: %.2f' % (concept.name, concept.value))

    return render_template('window.html')

@app.route('/verify', methods=['GET', 'POST'])
def verify():
    if request.method == 'POST':
        email1 = request.form['email1']
        sql = "SELECT * FROM email WHERE email1 = ? "
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,email1)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_tuple(stmt)
```

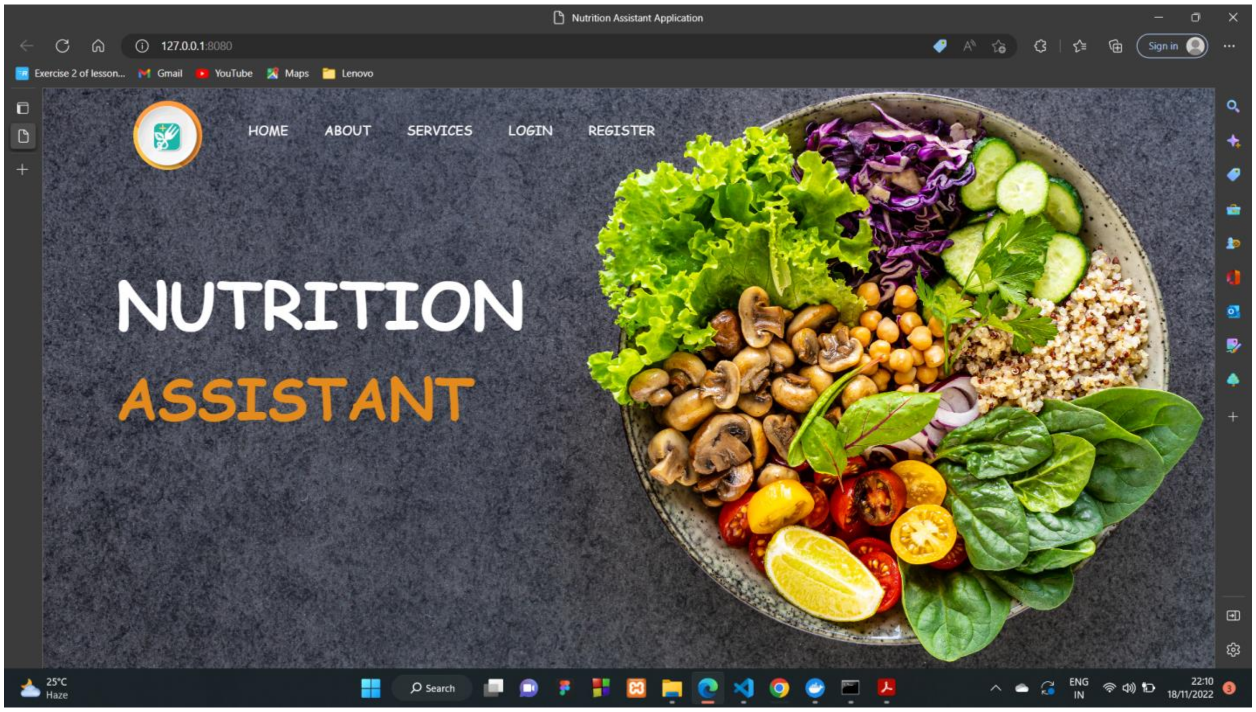
```
        print(account)
        if account:
            msg = 'Account already exists !'
        else:
           insert_sql = "INSERT INTO email(email1) VALUES(?)"
           stmt = ibm_db.prepare(conn,insert_sql)
           ibm_db.bind_param(stmt, 1, email1)
           ibm_db.execute(stmt)
                              msg = Message('NUTRITION ASSISTANT',sender
='nassistant.gans@gmail.com',recipients = [email1])
           msg.body = 'Hello user,THIS IS YOUR ONE TIME PASSWORD'
           msg.body = str(otp)
           mail.send(msg)
           return render_template('verify.html')
    return render_template('email.html')

@app.route('/validate',methods=['GET', 'POST'])
def validate():
 user_otp = request.form['otp']
 if otp == int(user_otp):
     return render_template('login.html')
 return render_template('verify.html')

@app.route('/services')
def services():
    icon = os.path.join(app.config['UPLOAD_FOLDER'],'icon.gif')
    return render_template('services.html',user_image=icon)
```

```
if __name__ == '__main__':
    app.debug = True
    app.run(host='0.0.0.0',port=8080)
```

7.1 Feature 1

homepage.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Nutrition Assistant Application</title>

                                                                                                        <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;700;900&displa
y=swap" rel="stylesheet">

    <link rel="stylesheet" href="/static/homepage.css">

</head>

<body>

    <!-- <img src="{{ user_image }}"> -->

    <header >


    <div class="wrapper">


        <div class="logo">

            <img src="{{ user_image}}" alt="">

        </div>

        <ul class="nav-area">

            <li><a href="{{url_for('homepage')}}">Home</a></li>

            <li><a href="{{url_for('about')}}">About</a></li>a

            <li><a href="{{url_for('services')}}">Services</a></li>

            <li><a href="{{url_for('login')}}">Login</a></li>

            <li><a href="{{url_for('register')}}">register</a></li>


        </ul>
    </div>
<div class="welcome-text">
        <h1>
NUTRITION <br> <span>ASSISTANT</span></h1>

    </div>
</header>

</body>
</html>
```

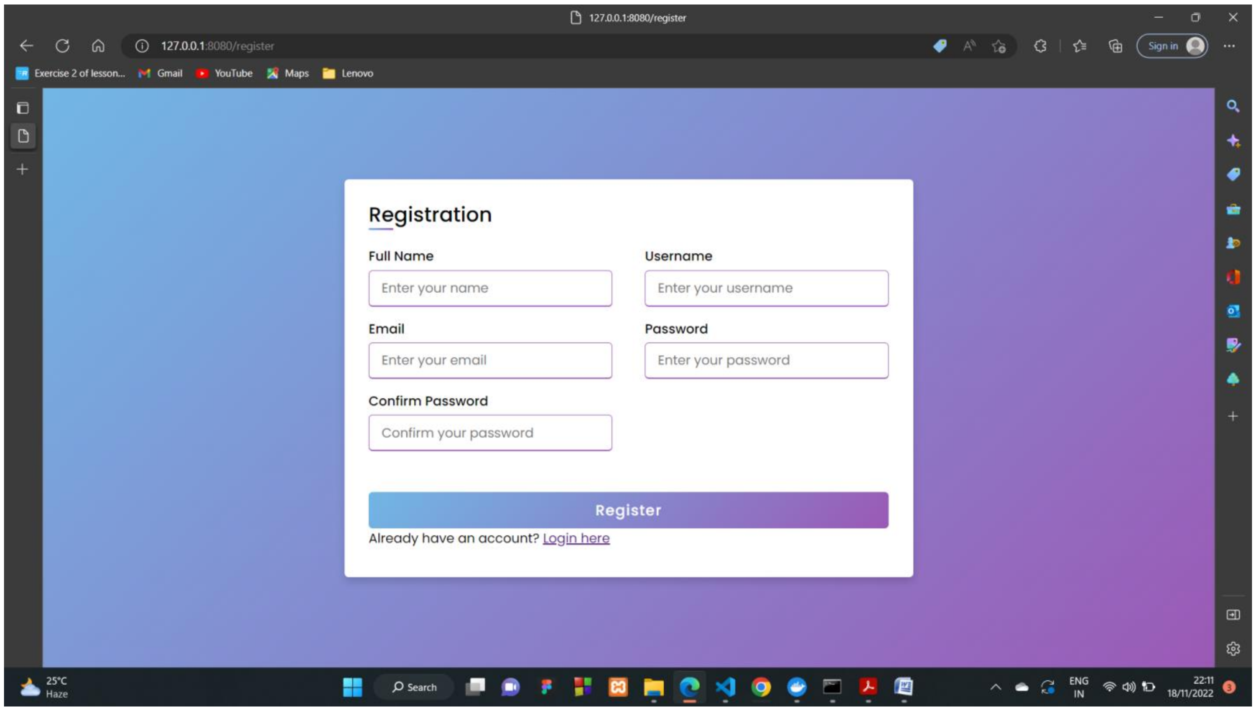


7.2 Feature 2

registration.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
```

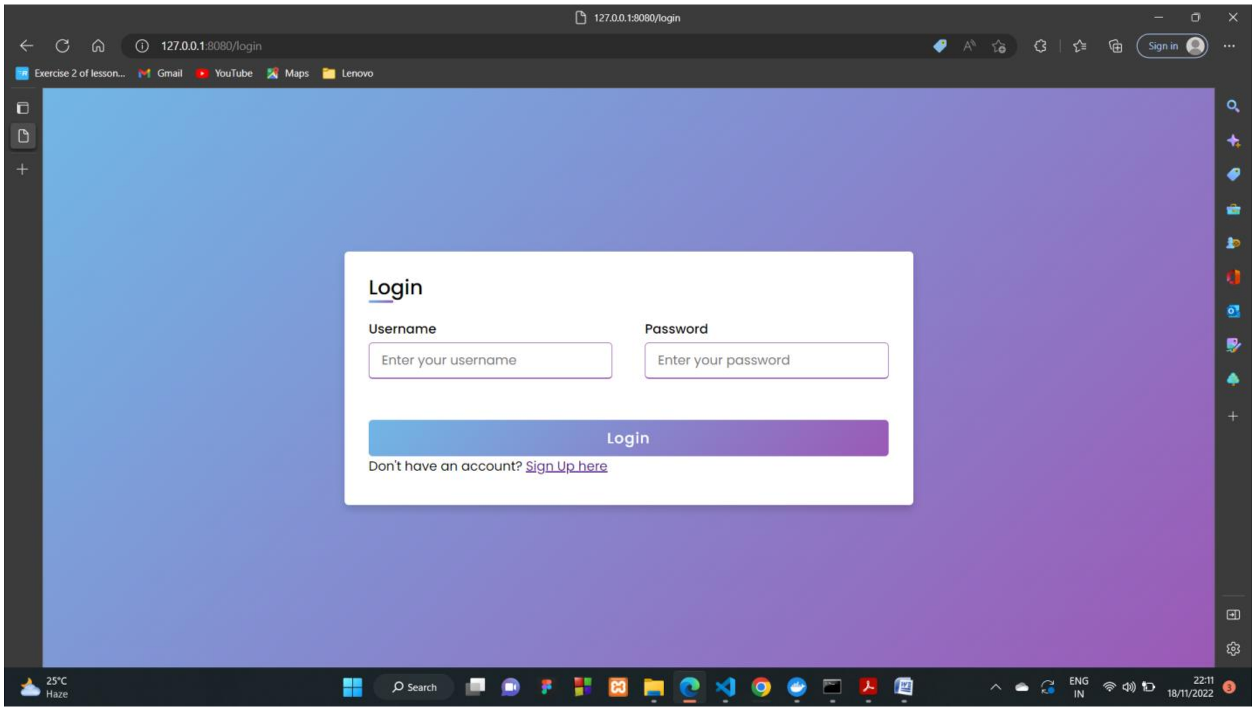
```
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="/static/registration.css">
     <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
<body>
  <div class="container">
    {{msg}}
    <div class="title">Registration</div>
    <div class="content">
      <form action="{{url_for('register')}}" method="POST" class="login-email">
        <div class="user-details">
          <div class="input-box">
            <span class="details">Full Name</span>
            <input type="text" placeholder="Enter your name" name="fullname">
          </div>
          <div class="input-box">
            <span class="details">Username</span>
            <input type="text" placeholder="Enter your username" name="username">
          </div>
          <div class="input-box">
            <span class="details">Email</span>
            <input type="text" placeholder="Enter your email" name="email">
          </div>

          <div class="input-box">
            <span class="details">Password</span>
            <input type="password" placeholder="Enter your password" name="passwords">
          </div>
          <div class="input-box">
            <span class="details">Confirm Password</span>
                          <input type="password" placeholder="Confirm your password" name="cpassword">
          </div>
        </div>
        <div class="button">
          <input type="submit" href="{{url_for('register')}}" value="Register">
                    <p class="bottom">Already have an account? <a class="bottom" href="{{url_for('login')}}"> Login here</a></p>
        </div>
      </form>
    </div>
  </div>

</body>
</html>
```



7.3 Feature 3

login.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="/static/registration.css">
     <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
<body>
  <div class="container">
    {{msg}}
    <div class="title">Login</div>
    <div class="content">
      <form action="{{url_for('login')}}" method="POST" class="login-email">
        <div class="user-details">
          <div class="input-box">
            <span class="details">Username</span>
            <input type="text" placeholder="Enter your username" name="username">
          </div>
          <div class="input-box">
            <span class="details">Password</span>
            <input type="password" placeholder="Enter your password" name="passwords" >
          </div>
        </div>
        <div class="button">
          <input type="submit" value="Login">
                    <p class="bottom">Don't have an account? <a class="bottom"
href="{{url_for('register')}}"> Sign Up here</a></p>
        </div>
      </form>
    </div>
  </div>


</body>
</html>
```



Login

Username

Enter your username

Password

Enter your password

Login

Don't have an account? Sign Up here

7.4 Feature 4

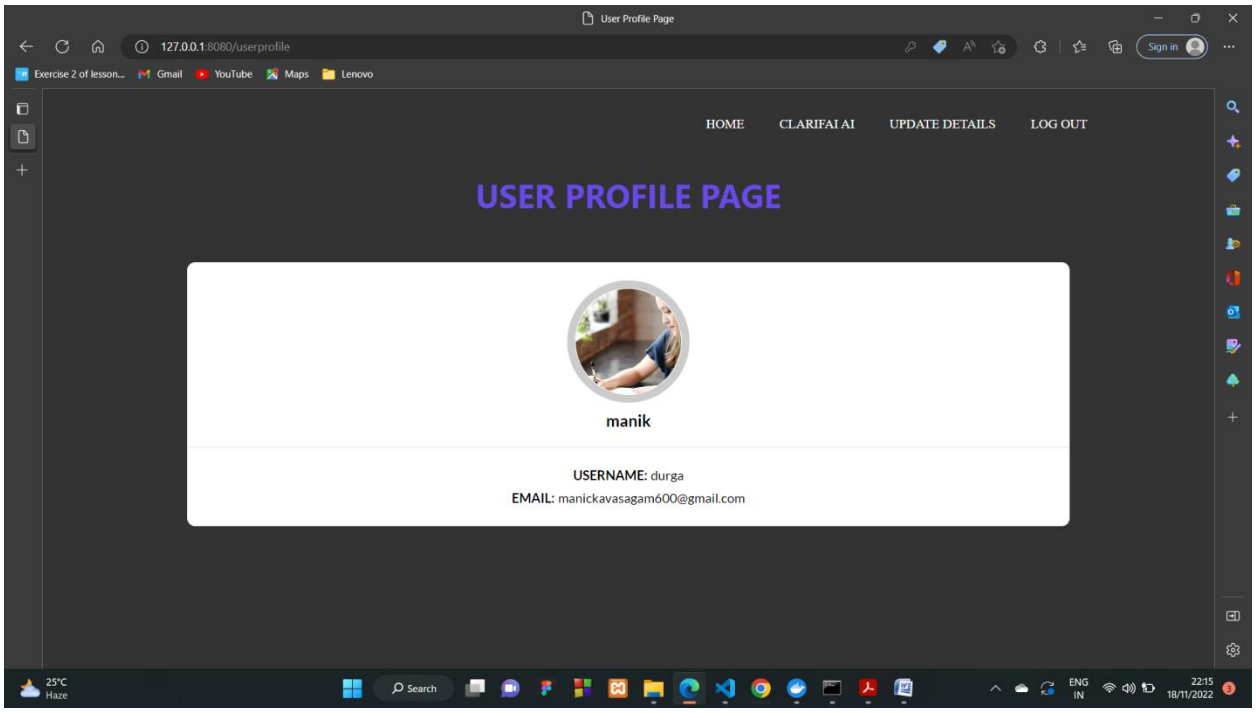
userprofile.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Profile Page</title>
    <meta name="author" content="Codeconvey" />
<link href="https://fonts.googleapis.com/css?family=Lato:300,400,700,900&display=swap" rel="stylesheet"><link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.1.3/css/bootstrap.min.css'>
<link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.12.1/css/all.min.css'>


    <link rel="stylesheet" href="/static/userprofile1.css" />



      <link rel="stylesheet" href="/static/userprofile2.css">
</head>
<body>

  <div class="wrapper">
    <div class="logo">
        <img src="/eatapps-1562790590.gif" alt="">
    </div>
    <ul class="nav-area">
        <li><a href="{{url_for('homepage')}}">Home</a></li>
        <li><a href="{{url_for('window')}}">Clarifai AI</a></li>
        <li><a href="{{url_for('updateprofile')}}">update Details</a></li>
        <li><a href="{{url_for('logout')}}">Log Out</a></li>

    </ul>
</div>

<header class="ScriptHeader">
```

```
    <div class="rt-container">
      <div class="col-rt-12">
          <div class="rt-heading">
              <h1>USER PROFILE PAGE</h1>
            </div>
        </div>
    </div>
</header>

<section>
    <div class="rt-container">
          <div class="col-rt-12">
              <div class="Scriptcontent">



<div class="student-profile py-4">
  <div class="container">
    <div class="row">
      <div class="col-lg-12">
        <div class="card shadow-sm">
          <div class="card-header bg-transparent text-center">
                <img  class="profile_img"  src="https://source.unsplash.com/600x300/?student"
alt="student dp">
            <h3>{{fullname}}</h3>
          </div>
          <div class="card-body text-center">
                   <p  class="mb-0"><strong  class="pr-1">USERNAME:</strong>{{username
}}</p>
            <p class="mb-0"><strong class="pr-1">EMAIL:</strong>{{email}}</p>
        </div>
      </div>
    </div>
    <!-- <div class="col-lg-8">
      <div class="card shadow-sm">
        <div class="card-header bg-transparent border-0">
              <h3 class="mb-0"><i class="far fa-clone pr-1"></i>General Information
&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&n
bsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&n
bsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbs
p&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&
nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&n
bsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
          <a class="bottom" href="{{url_for('updateprofile')}}"> EDIT</a></h3>
        </div>
        <div class="card-body pt-0">
          <table class="table table-bordered">
            <tr>
              <th width="30%">Height</th>
              <td width="2%">:</td>
              <td>{{height}}</td>
            </tr>
            <tr>
              <th width="30%">Weight</th>
              <td width="2%">:</td>
              <td>{{weight}}</td>
            </tr>
            <tr>
              <th width="30%">Gender</th>
```

```
                <td width="2%">:</td>
                <td>{{gender}}</td>
              </tr>
              <tr>
                <th width="30%">Blood</th>
                <td width="2%">:</td>
                <td>{{blood}}</td>
              </tr>
            </table>
          </div>
        </div>



          <div style="height: 26px"></div>
        <div class="card shadow-sm">
          <div class="card-header bg-transparent border-0">
            <h3 class="mb-0"><i class="far fa-clone pr-1"></i>Daily Activity</h3>
          </div>
          <div class="card-body pt-0">
             <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```
        </div>
      </div>
      </div>
</section>
     -->


      <!-- Analytics -->


   </body>
</html>
```



USER PROFILE PAGE

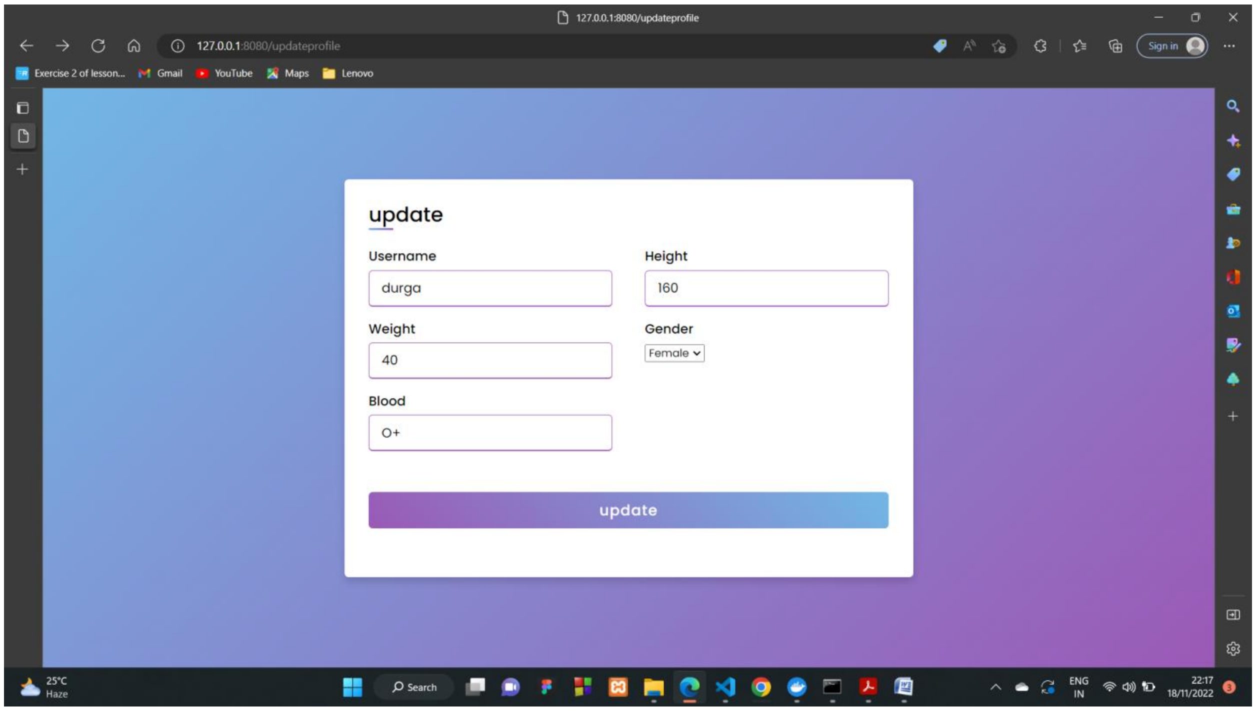
manik

USERNAME: durga
EMAIL: manickavasagam600@gmail.com

7.5 Feature 5

updateprofile.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="/static/registration.css">
     <meta name="viewport" content="width=device-width, initial-scale=1.0">

   </head>
<body>
  <div class="container">
    <div class="title">update</div>
    <div class="content">
      <form action="{{url_for('updateprofile')}}" method="POST" class="login-email">
        <div class="user-details">
          <div class="input-box">
            <span class="details">Username</span>
            <input type="text" placeholder="Enter your height" name="username">
          </div>
          <div class="input-box">
            <span class="details">Height</span>
            <input type="text" placeholder="Enter your height" name="height">
          </div>
          <div class="input-box">
            <span class="details">Weight</span>
            <input type="text" placeholder="Enter your weight" name="weight">
          </div>
          <div class="input-box">
            <span class="details">Gender</span>
            <select name="gender">
              <option value="Male">Male</option>
              <option value="Female">Female</option>
            </select>
          </div>
          <div class="input-box">
            <span class="details">Blood</span>
            <input type="text" placeholder="Enter your Blood group" name="blood">
          </div>



        </div>
        <span>{{msg}}</span>
        <div class="button">
          <input type="submit" href="{{url_for('userprofile')}}" value="update">

        </div>
      </form>
    </div>
  </div>

</body>
</html>
```



update

Username: durga

Height: 160

Weight: 40

Gender: Female

Blood: O+

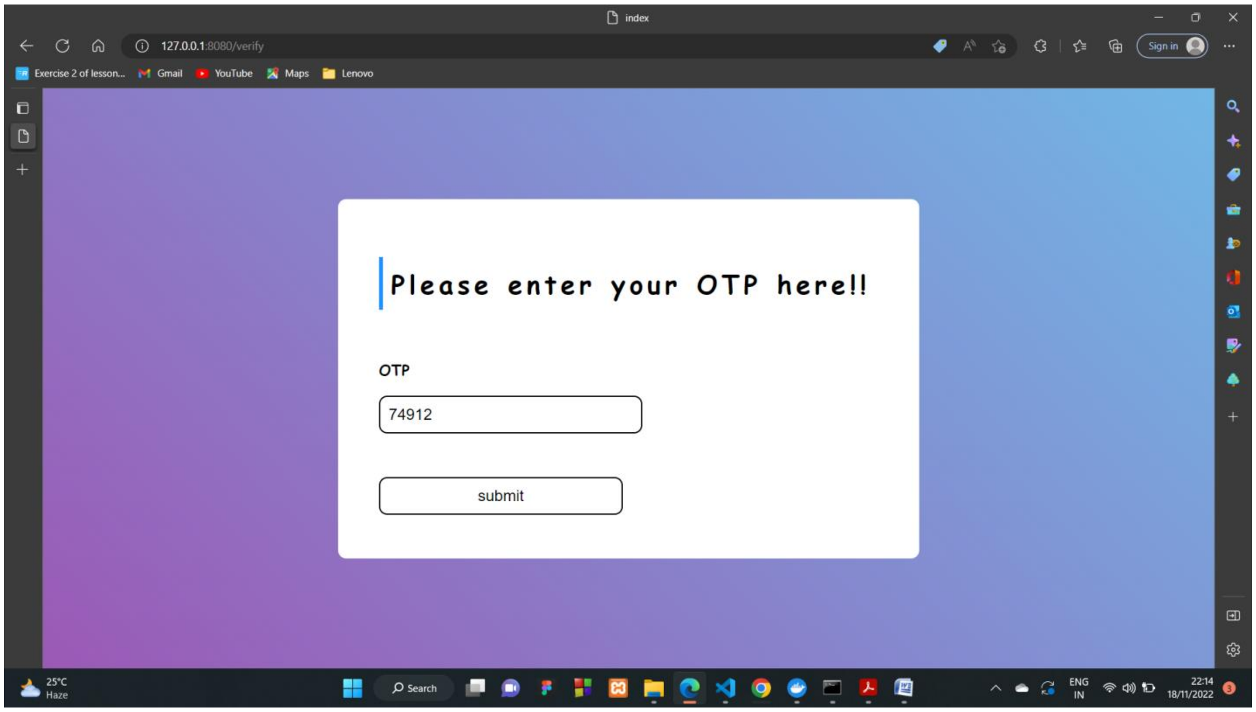
7.6 Feature 6

verify.html

```
<!DOCTYPE html>
 <html>
 <head>
     <title>index</title>
 </head>
 <body>
    <style>
@import url("https://fonts.googleapis.com/css2?family=Sansita+Swashed:wght@600&display=swap");
body {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  background: linear-gradient(45deg, #9b59b6, #71b7e6);
  font-family: cursive;


}
.center {
  position: relative;
  padding: 50px 50px;
  background: #fff;
  border-radius: 10px;
}
.center h1 {
  font-size: 2em;
  border-left: 5px solid dodgerblue;
  padding: 10px;
  color: #000;
  letter-spacing: 5px;
  margin-bottom: 60px;
  font-weight: bold;
  padding-left: 10px;
}
```

```
.center .inputbox {
  position: relative;
  width: 300px;
  height: 50px;
  margin-bottom: 50px;
}
.center .inputbox input {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  border: 2px solid #000;
  outline: none;
  background: none;
  padding: 10px;
  border-radius: 10px;
  font-size: 1.2em;
}
.center .inputbox:last-child {
  margin-bottom: 0;
}
.center .inputbox span {
  position: absolute;
  top: 14px;
  left: 20px;
  font-size: 1em;
  transition: 0.6s;
  font-family: sans-serif;
}
.center .inputbox input:focus ~ span,
.center .inputbox input:valid ~ span {
  transform: translateX(-13px) translateY(-35px);
  font-size: 1em;
}
.center .inputbox [type="button"] {
  width: 50%;
  background: dodgerblue;
  color: #fff;
  border: #fff;
}
.center .inputbox:hover [type="button"] {
  background: linear-gradient(45deg, #71b7e6, #9b59b6);
}
    </style>
<form action = "{{ url_for('validate') }}" method="post">
    <div class="center">
        <h1>Please enter your OTP here!!</h1>

        <h3>OTP</h3>
          <div class="inputbox">

            <input type="text" name="otp">



          </div>
          <div class="inputbox">
```

```
            <input type="submit"  value="submit" value="Continue">
          </div>


      </div>
 </form>
 </body>
 </html>
```



Please enter your OTP here!!

OTP

74912

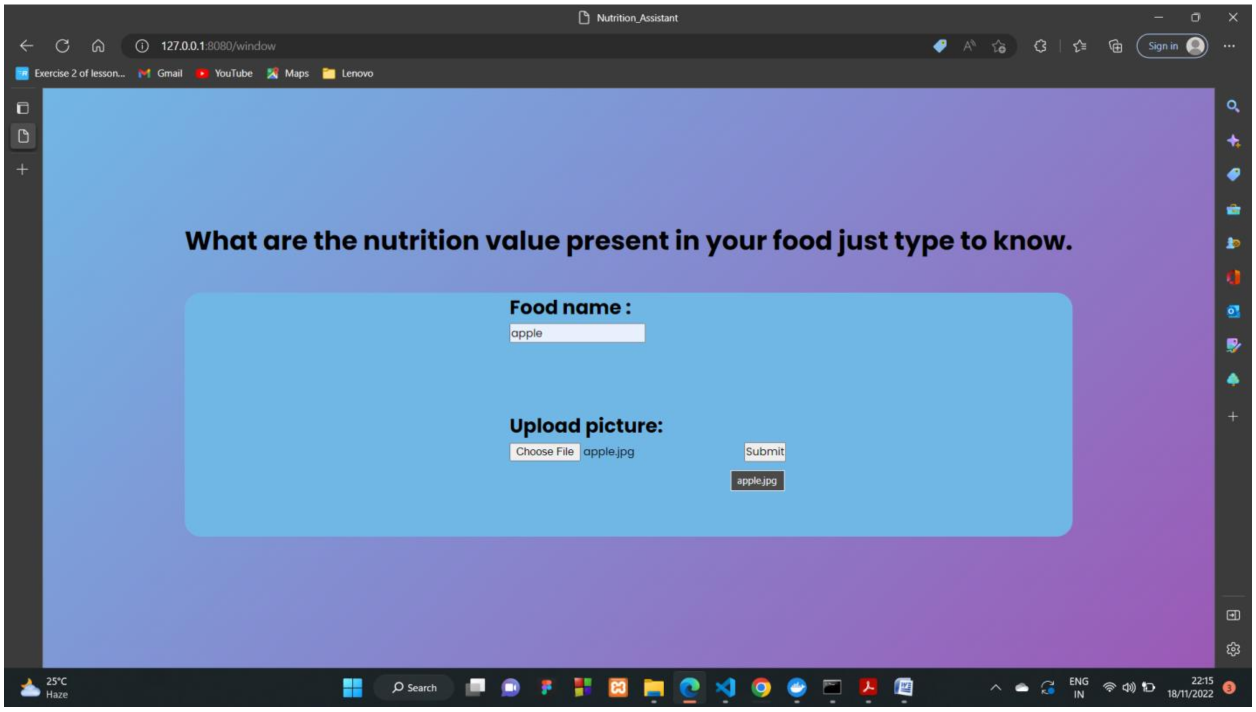
submit

7.7 Feature 7

window.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Nutrition_Assistant</title>
    <link rel='stylesheet' href='https://fonts.googleapis.com/css?family=Rubik:400,700'>
    <link rel="stylesheet" href="/static/window.css">

</head>

<body>
    <script>
        function act(e){
          e.preventDefault()
          console.log(e.target.file.files[0])
          alert("hii")
        }
      </script>
    <div class="windows">
        <form action="{{ url_for('window') }}" method="POST">
            <h1>What are the nutrition value present in your food just type to know.</h1>
           <div class="row">
            <div class="foodname">
                <h2>Food name :</h2>
                <input type="text" class="food-name" name="foodname">
```

```
         </div><br>
         <div class="imagesearch">
           <h2>Upload picture:</h2>
               <input  type="file"  accept="image/*"  class="request-image"  id="image"
name="image">
           <input class="btn btn-outline-primary" type="submit" value="Submit">
         </div>
       </div>


       </form>
     </div>
</body>


</html>
```



What are the nutrition value present in your food just type to know.

Food name :

apple

Upload picture:

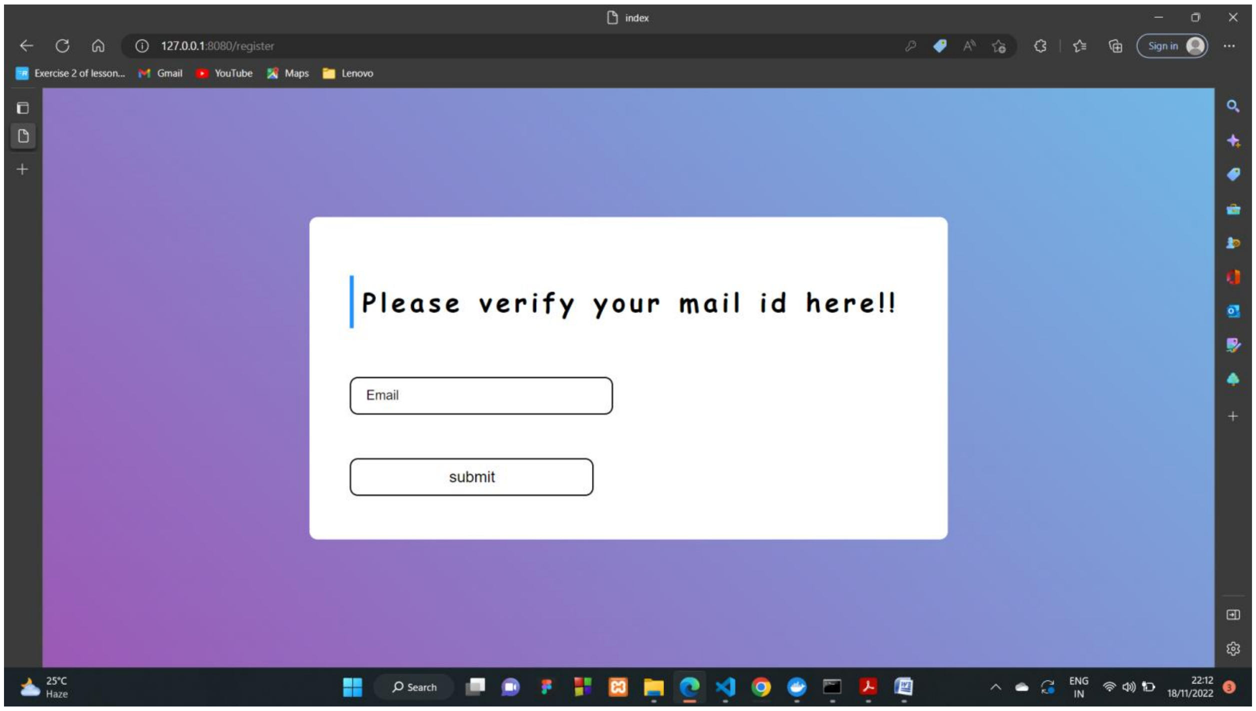
apple.jpg

7.8 Feature 8

email.html

```
<!DOCTYPE html>
 <html>
 <head>
     <title>index</title>
 </head>
 <body>
    <style>
                                                                                              @import
url("https://fonts.googleapis.com/css2?family=Sansita+Swashed:wght@600&display=sw
ap");
body {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  background: linear-gradient(45deg, #9b59b6, #71b7e6);
  font-family: cursive;



}
.center {
  position: relative;
  padding: 50px 50px;
  background: #fff;
  border-radius: 10px;
}
.center h1 {
  font-size: 2em;
  border-left: 5px solid dodgerblue;
  padding: 10px;
  color: #000;
  letter-spacing: 5px;
  margin-bottom: 60px;
  font-weight: bold;
  padding-left: 10px;
}
.center .inputbox {
  position: relative;
  width: 300px;
  height: 50px;
  margin-bottom: 50px;
}
.center .inputbox input {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  border: 2px solid #000;
  outline: none;
  background: none;
  padding: 10px;
  border-radius: 10px;
  font-size: 1.2em;
}
.center .inputbox:last-child {
  margin-bottom: 0;
}
.center .inputbox span {
  position: absolute;
  top: 14px;
  left: 20px;
  font-size: 1em;
  transition: 0.6s;
  font-family: sans-serif;
}
.center .inputbox input:focus ~ span,
.center .inputbox input:valid ~ span {
  transform: translateX(-13px) translateY(-35px);
  font-size: 1em;
}
.center .inputbox [type="button"] {
  width: 50%;
  background: dodgerblue;
  color: #fff;
  border: #fff;
}
.center .inputbox:hover [type="button"] {
  background: linear-gradient(45deg, #71b7e6, #9b59b6);
}
```

```
  </style>
<form action = "{{ url_for('verify') }}" method = "post">
  <div class="center">
      <h1>Please verify your mail id here!!</h1>


        <div class="inputbox">
          <input type="text" required="required" name="email1">
          <span>Email</span>
        </div>
        <div class="inputbox">
          <input type="submit"  value="submit">
        </div>


    </div>
</form>
</body>
</html>
```



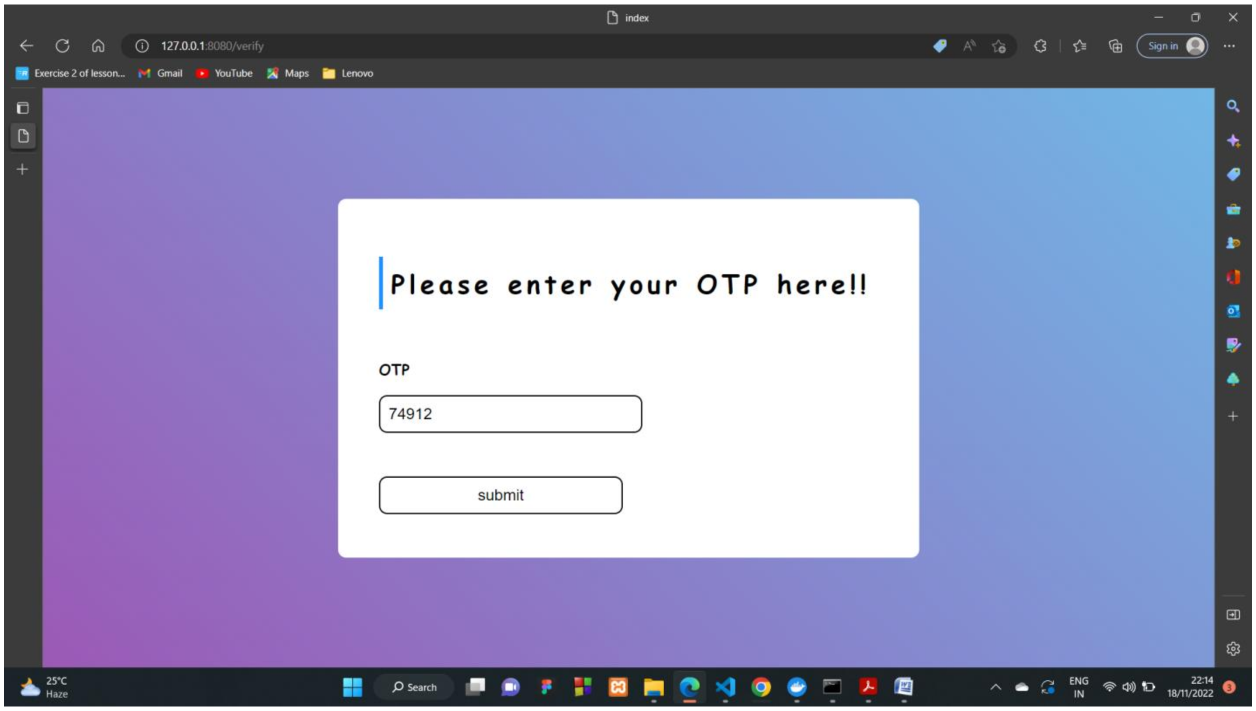
Please verify your mail id here!!

7.9 Feature 9

details.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Profile Page</title>


    <meta name="author" content="Codeconvey" />
                                                                        <link
href="https://fonts.googleapis.com/css?family=Lato:300,400,700,900&display=swap"
rel="stylesheet"><link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.3/css/bootstrap.min.css'>
        <link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.12.1/css/all.min.css'>
```

```
        <link rel="stylesheet" href="/static/userprofile1.css" />




          <link rel="stylesheet" href="/static/userprofile2.css">
    </head>
<body>
    <div class="col-lg-12">
        <div class="card shadow-sm">
          <div class="card-header bg-transparent border-0">
                    <h3 class="mb-0"><i class="far fa-clone pr-1"></i>General Information
&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&n
bsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&n
bsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbs
p&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&
nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&n
bsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
            <a class="bottom" href="{{url_for('userprofile')}}"> Profile</a></h3>
          </div>
          <div class="card-body pt-0">
            <table class="table table-bordered">
              <tr>
                <th width="30%">Height</th>
                <td width="2%">:</td>
                <td>{{height}}</td>
              </tr>
              <tr>
                <th width="30%">Weight</th>
                <td width="2%">:</td>
                <td>{{weight}}</td>
              </tr>
              <tr>
                <th width="30%">Gender</th>
                <td width="2%">:</td>
                <td>{{gender}}</td>
              </tr>
              <tr>
                <th width="30%">Blood</th>
                <td width="2%">:</td>
                <td>{{blood}}</td>
              </tr>
            </table>
          </div>
        </div>
        <!-- button -->



      </div>
    </div>
  </div>
</div>


</body>
</html>
```



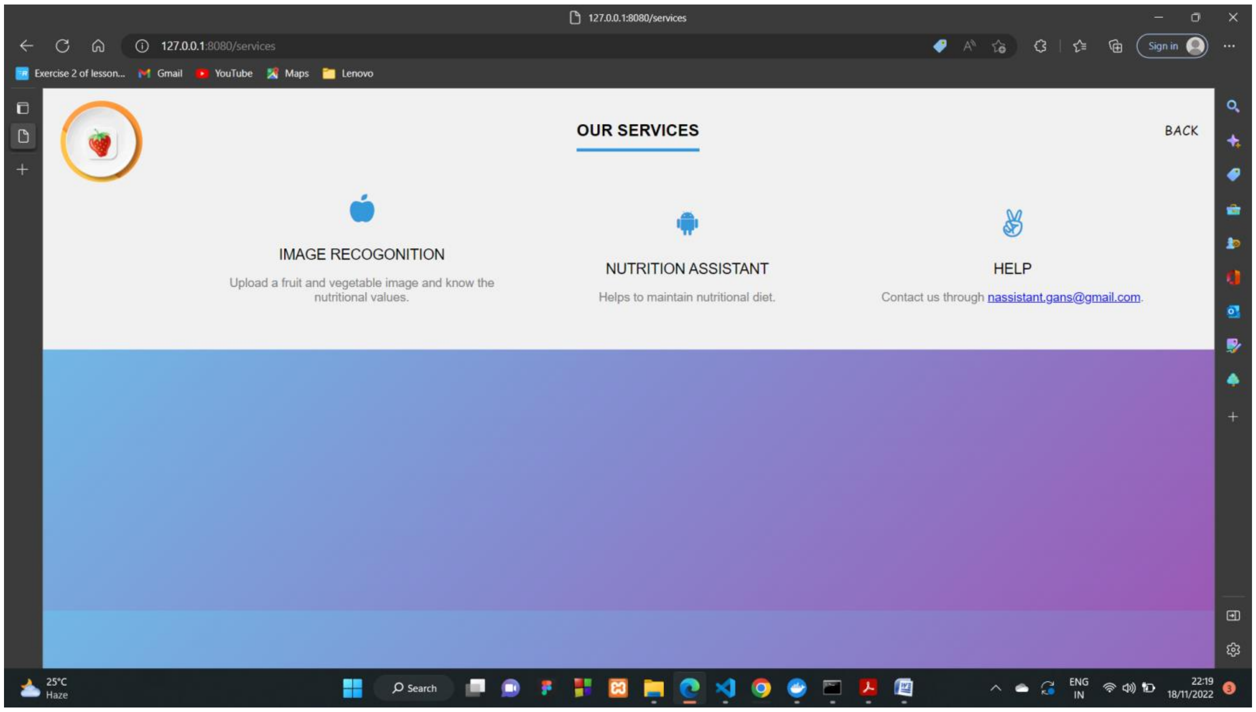
7.10 Feature 10

service.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>
    <link rel="stylesheet" href="/static/services.css">

    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.3.1/css/all.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>

    </style>
  </head>
```

```
<body>
  <div class="wrapper">

    <div class="logo">
        <img src="{{ user_image}}" alt="">
    </div>
    <ul class="nav-area">
        <li><a href="{{url_for('homepage')}}">Back</a></li>



    </ul>
</div>
  <div class="services">
    <h1>Our Services</h1>
    <div class="cen">
      <div class="service">
        <i class="fas fa-apple-alt"></i>
        <h2>Image Recogonition</h2>
        <p>Upload a fruit and vegetable image and know the nutritional values.</p>
      </div>

      <div class="service">
        <i class="fab fa-android"></i>
        <h2>Nutrition Assistant</h2>
        <p>Helps to maintain nutritional diet.</p>
      </div>
```

```
        <div class="service">
          <i class="fab fa-angellist"></i>
          <h2>Help</h2>
                                    <p>Contact us through <a href="mailto:
nassistant.gans@gmail.com">nassistant.gans@gmail.com</a>.</p>
        </div>


      </div>
    </div>


  </body>
</html>
```



OUR SERVICES

BACK

IMAGE RECOGONITION

Upload a fruit and vegetable image and know the nutritional values.

NUTRITION ASSISTANT

Helps to maintain nutritional diet.

HELP

Contact us through nassistant.gans@gmail.com.

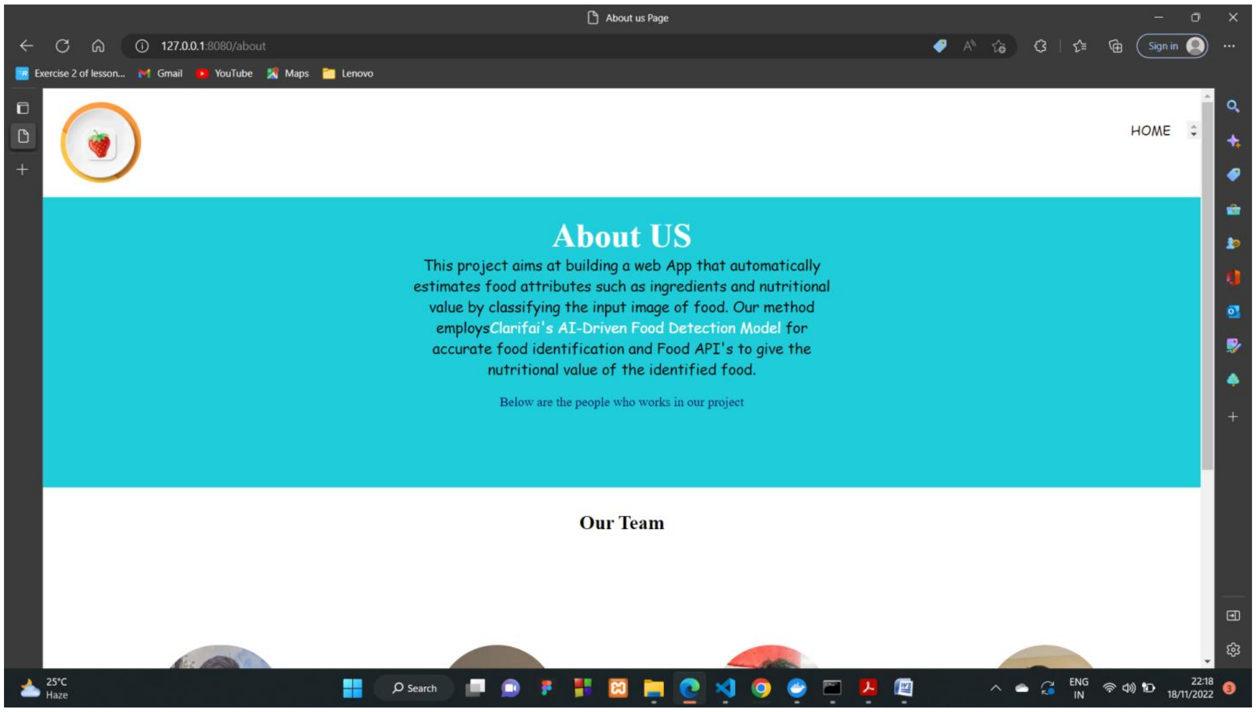
7.11 Feature 11

about.html

```
<!DOCTYPE html>
```

```
<html>

<head>
    <title>About us Page</title>
    <link rel="stylesheet" href="/static/about.css">
    <!-- <link rel="stylesheet" href="homepage.css"> -->
</head>
<body>

    <div class="wrapper">

        <div class="logo">
            <img src="{{ user_image}}" alt="">
        </div>
        <ul class="nav-area">
            <li><a href="{{url_for('homepage')}}">Home</a></li>



        </ul>
    </div>




    <section class="background firstsection">
        <div class="box-main">
            <div class="firstHalf">
                <p class="text-big">About US</p>
                <p class="text-small" style="color: black;">
                        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<span style="color: white;">Clarifai's AI-Driven Food Detection Model</span>  for accurate food identification and Food API's to give the nutritional value of the identified food.
                </p>
                <br>
                <p class="center"><a href="#Order"
                style="text-decoration:none;color:rgb(9, 10, 98);">
                    Below are the people who
                    works in our project</a>
                </p>
            </div>
        </div>
    </section>
    <section class="service">
        <h1 class="h-primary center" style="margin-top:30px;text-align:center;">
            Our Team
        </h1>
    <div id="services">
            <div class="box">
                <img src=
"/static/pics/Ashwin.jpg"
                    alt="picture goes here">

                <p class="center">
                    <a href="#xyz" style="text-decoration:none;color:black;
        font-weight:bold;font-family: 'Langar', cursive;">
```

```
                    ASHWIN KUMAR
                  </a>
                </p>
                <p style="font-family: sans-serif">TEAM LEADER</p>
              </div>
              <div class="box">
                <img src=
"/static/pics/GAN.jpeg"
                  alt="picture goes here">

                <p class="center">
                  <a href="#abc" style="text-decoration:none;color:black;
        font-weight:bold;font-family: 'Langar', cursive;">
                    GANESAN
                  </a>
                </p>

                <p style="font-family: sans-serif ">TEAM MEMBER</p>

              </div>
              <div class="box">
                <img src="/static/pics/SRI.jpeg"
                  alt="picture goes here">

                <p class="center">
                  <a href="#abc" style="text-decoration:none;color:black;
        font-weight:bold;font-family: 'Langar', cursive;">
                    SRIRAM
                    </a>
                </p>

                <p style="font-family: sans-serif ">TEAM MEMBER</p>

            </div>
            <div class="box">
                <img src=
"/static/pics/NIVI.jpg"
                    alt="picture goes here">
                <br>
                <p class="center">
                    <a href="#xyz" style="text-decoration:none;color:black;
        font-weight:bold;font-family: 'Langar', cursive;">
                        NIVENDHAN
                    </a>
                </p>
                <p style="font-family: sans-serif ">TEAM MEMBER</p>



            </div>
        </div>


    </section>


    <footer class="background">
        <p class="text-footer">
            NUTRITION ASSISTANT APPLICATION
        </p>


    </footer>
</body>


</html>
```

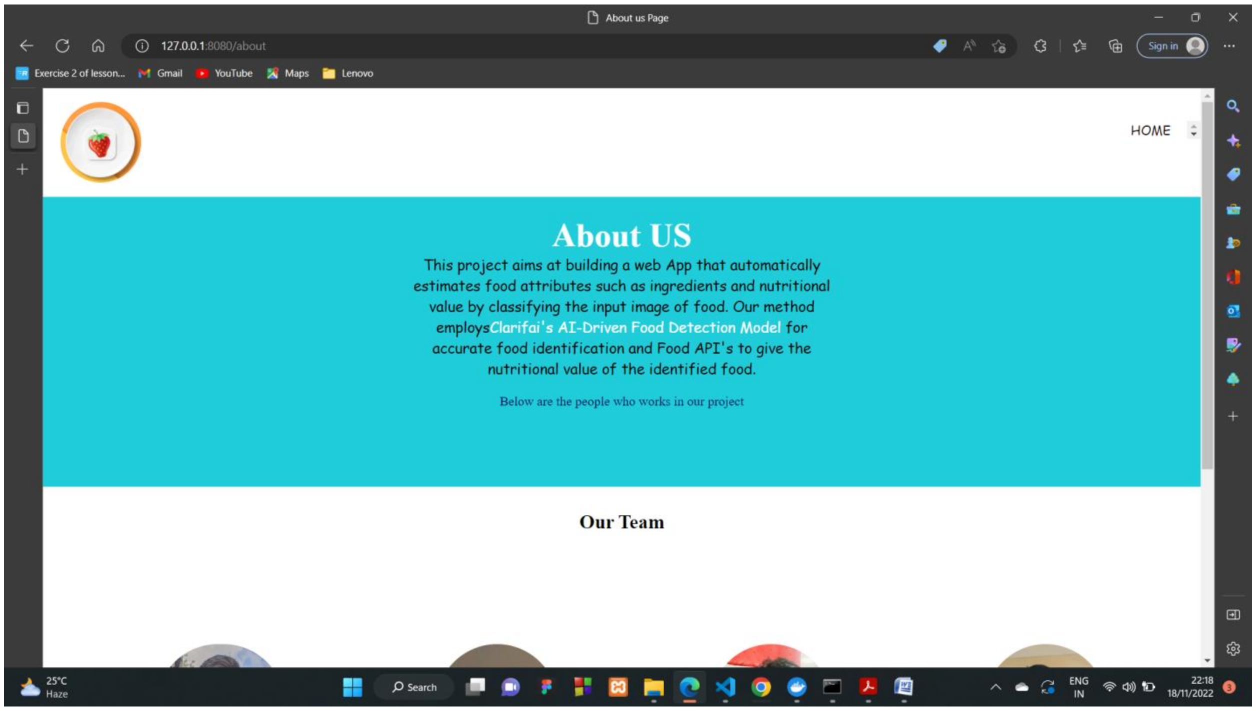


About US

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 employsClarifai's AI-Driven Food Detection Model for accurate food identification and Food API's to give the nutritional value of the identified food.

Below are the people who works in our project

Our Team



About US

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 employsClarifai's AI-Driven Food Detection Model for accurate food identification and Food API's to give the nutritional value of the identified food.

Below are the people who works in our project

Our Team

CHAPTER-8

TESTING

8.1 Test Case

Test case ID	Test Scenario	Expected Result	Status
Home_TC_OO1	Verify user is able to see the Login button	Login button is displayed	Pass
Home_TC_OO2	Verify whether register button works	Redirected to registration page	Pass
Home_TC_OO3	Verify whether login button works	Redirected to login page	Pass
Home_TC_OO4	Verify whether service button works	Redirected to support page	Pass
Registration_TC_OO1	Verify the registration credentials vaild or not	Application should show below UI elements: a.fullname box b.email text box - mandatory field c.Password textbox - mandatory field with minimum 5 characters with atleast 1 alphabet and 1 number no special characters allowed d.Confirm password text box - mandatory field e.Register button	Pass
Registration_TC_OO2	Verify whether register button works	Redirects to Email verification page	Pass
Registration_TC_OO3	Verify whether the page will redirect to login page if account already registered	Redirects to Login page	Pass
Profileupdation_TC_OO1	Verify user is able to see profile updation credentials	1.Verify personal details page with below UI elements: a.Height text box - mandatory field b.Weight textbox- mandatory field c.Gender text box - mandatory field d.Blood text box - mandatory field	Pass
Profileupdation_TC_OO2	Verify whether proceed to Update button works	Redirects to User profile page	Pass
Login_TC_OO1	Verify whether user is able to see email and password text box	User should navigate to user account homepage	Pass
Login_TC_OO2	Verify user is able to log into application with Valid credentials	Application redirects to Userprofile	Pass
Login_TC_OO3	Verify user is able to log into application with InValid credentials	Application should show 'Incorrect email or password ' validation message.	Pass

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. 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 Design	10	4	2	3	18
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	9	2	4	18	35
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	0	1	8
Totals	22	14	11	24	74

3. Test Case Analysis

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

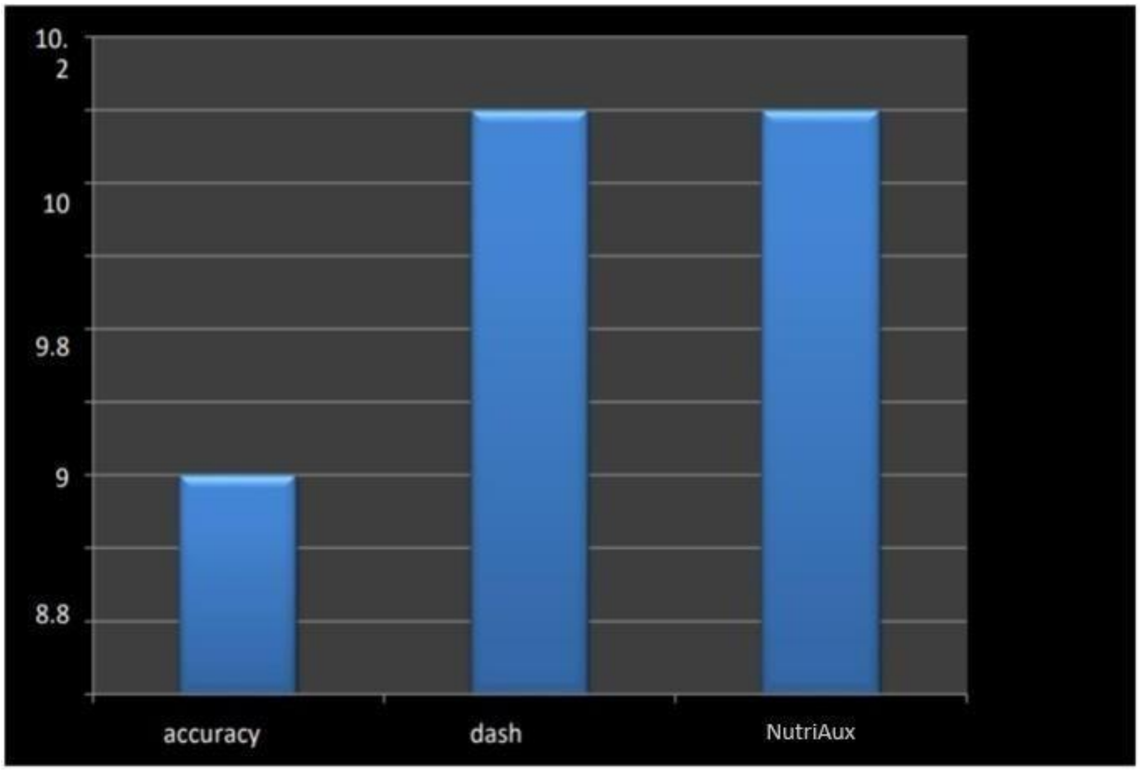
Section	Total Cases	Not Tested	Fail	Pass

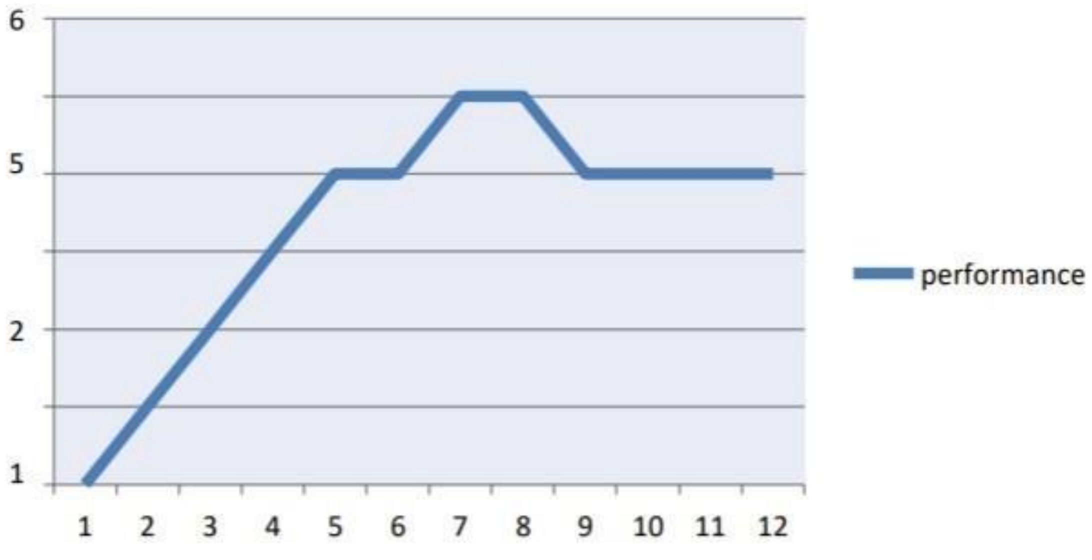
Homepage	5	0	0	5
Login	26	0	0	26
Register	3	0	0	3
Email Verification	3	0	0	3
OTP Verification	9	0	0	9
User Details	5	0	0	5
Clarifai-AI	3	0	0	3

CHAPTER-9

RESULT

9.1 Performance Metrics





CHAPTER-10

ADVANTAGES & DISADVANTAGES

Advantages:

- By using our webapp, the user can know their BMI, which will lead the user to decide whether he has to gain weight or lose weight
- User can know their daily calorie intake, which can help them to know amount of calorie they can consume for that particular day.
- The user can upload the image of the meal which will provide them the nutritional value of that particular meal.
- NutriAux is a user friendly and easy to use application.
- The user can track the daily calorie intake which will help them to know their progress towards their fitness goal.

Disadvantages:

- It requires an active internet connection.
- Not all types of foods can be detected correctly by Clarifai Food Detection Model API. The user cannot update their personal details once it has been registered.

CHAPTER-11

CONCLUSION

Since obesity rate has become a major problem in this decade, the diet management is very important. The information about the nutritional value of the food that has been printed in the food packages are not convenient to keep track of the daily calorie intake. NutriAux helps in finding the nutritional content present in the food with real time image processing using Clarifai Food Detection Model API and Spoonacular Nutrition API. The user can upload his daily meal image and get the nutritional value. They can also track their daily calorie intake

.

CHAPTER-12

FUTURE SCOPE

NutriAux will be upgraded in the following years with the feature of "Profile Updation". The user can update his personal details like height, weight and age which will help them to keep track of the daily calorie intake and the BMI. "Dietary Recommendation" facility and "Water Reminder" facility will also be added in the future.

CHAPTER-13

Source code

homepage.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Nutrition Assistant Application</title>
                                                                                    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;700;900&displa
y=swap" rel="stylesheet">
   <link rel="stylesheet" href="/static/homepage.css">
</head>
<body>
```

```
    <!-- <img src="{{ user_image }}"> -->
    <header >

    <div class="wrapper">

        <div class="logo">
            <img src="{{ user_image}}" alt="">
        </div>
        <ul class="nav-area">
            <li><a href="{{url_for('homepage')}}">Home</a></li>
            <li><a href="{{url_for('about')}}">About</a></li>a
            <li><a href="{{url_for('services')}}">Services</a></li>
            <li><a href="{{url_for('login')}}">Login</a></li>
            <li><a href="{{url_for('register')}}">register</a></li>

        </ul>
    </div>
<div class="welcome-text">
        <h1>
NUTRITION <br> <span>ASSISTANT</span></h1>

    </div>
</header>

</body>
</html>
```

registration.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="/static/registration.css">
     <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
<body>
  <div class="container">
    {{msg}}
    <div class="title">Registration</div>
    <div class="content">
      <form action="{{url_for('register')}}" method="POST" class="login-email">
        <div class="user-details">
          <div class="input-box">
            <span class="details">Full Name</span>
            <input type="text" placeholder="Enter your name" name="fullname">
          </div>
          <div class="input-box">
            <span class="details">Username</span>
            <input type="text" placeholder="Enter your username" name="username">
          </div>
          <div class="input-box">
            <span class="details">Email</span>
            <input type="text" placeholder="Enter your email" name="email">
          </div>
          <div class="input-box">
            <span class="details">Password</span>
            <input type="password" placeholder="Enter your password" name="passwords">
          </div>
          <div class="input-box">
            <span class="details">Confirm Password</span>
            <input type="password" placeholder="Confirm your password" name="cpassword">
          </div>
        </div>
        <div class="button">
          <input type="submit" href="{{url_for('register')}}" value="Register">
          <p class="bottom">Already have an account? <a class="bottom" href="{{url_for('login')}}"> Login here</a></p>
        </div>
      </form>
    </div>
  </div>


</body>
</html>
```

login.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="/static/registration.css">
     <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
<body>
  <div class="container">
    {{msg}}
    <div class="title">Login</div>
    <div class="content">
      <form action="{{url_for('login')}}" method="POST" class="login-email">
        <div class="user-details">
          <div class="input-box">
            <span class="details">Username</span>
            <input type="text" placeholder="Enter your username" name="username">
          </div>
          <div class="input-box">
            <span class="details">Password</span>
            <input type="password" placeholder="Enter your password" name="passwords" >
          </div>
        </div>
        <div class="button">
          <input type="submit" value="Login">
                        <p class="bottom">Don't have an account? <a class="bottom"
href="{{url_for('register')}}"> Sign Up here</a></p>
        </div>
      </form>
    </div>
  </div>

</body>
</html>
```

userprofile.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Profile Page</title>

    <meta name="author" content="Codeconvey" />
    <link href="https://fonts.googleapis.com/css?family=Lato:300,400,700,900&display=swap" rel="stylesheet"><link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.1.3/css/bootstrap.min.css'>
<link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.12.1/css/all.min.css'>

    <link rel="stylesheet" href="/static/userprofile1.css" />



      <link rel="stylesheet" href="/static/userprofile2.css">
</head>
```

```
<body>

  <div class="wrapper">
    <div class="logo">
        <img src="/eatapps-1562790590.gif" alt="">
    </div>
    <ul class="nav-area">
        <li><a href="{{url_for('homepage')}}">Home</a></li>
        <li><a href="{{url_for('window')}}">Clarifai AI</a></li>
        <li><a href="{{url_for('updateprofile')}}">update Details</a></li>
        <li><a href="{{url_for('logout')}}">Log Out</a></li>

    </ul>
</div>

<header class="ScriptHeader">
    <div class="rt-container">
      <div class="col-rt-12">
          <div class="rt-heading">
              <h1>USER PROFILE PAGE</h1>
            </div>
        </div>
    </div>
</header>

<section>
    <div class="rt-container">
          <div class="col-rt-12">
```

```
            <div class="Scriptcontent">



<div class="student-profile py-4">
  <div class="container">
    <div class="row">
      <div class="col-lg-12">
        <div class="card shadow-sm">
          <div class="card-header bg-transparent text-center">
                <img class="profile_img" src="https://source.unsplash.com/600x300/?student"
alt="student dp">
            <h3>{{fullname}}</h3>
          </div>
          <div class="card-body text-center">
                  <p class="mb-0"><strong class="pr-1">USERNAME:</strong>{{username
}}</p>
            <p class="mb-0"><strong class="pr-1">EMAIL:</strong>{{email}}</p>
          </div>
        </div>
      </div>
      <!-- <div class="col-lg-8">
        <div class="card shadow-sm">
          <div class="card-header bg-transparent border-0">
                   <h3 class="mb-0"><i class="far fa-clone pr-1"></i>General Information
&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&n
bsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&n
bsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbs
p&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&
nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&n
bsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
```

```html
    <a class="bottom" href="{{url_for('updateprofile')}}"> EDIT</a></h3>
  </div>
  <div class="card-body pt-0">
    <table class="table table-bordered">
      <tr>
        <th width="30%">Height</th>
        <td width="2%">:</td>
        <td>{{height}}</td>
      </tr>
      <tr>
        <th width="30%">Weight</th>
        <td width="2%">:</td>
        <td>{{weight}}</td>
      </tr>
      <tr>
        <th width="30%">Gender</th>
        <td width="2%">:</td>
        <td>{{gender}}</td>
      </tr>
      <tr>
        <th width="30%">Blood</th>
        <td width="2%">:</td>
        <td>{{blood}}</td>
      </tr>
    </table>
  </div>
</div>
```

```
          <div style="height: 26px"></div>
        <div class="card shadow-sm">
          <div class="card-header bg-transparent border-0">
            <h3 class="mb-0"><i class="far fa-clone pr-1"></i>Daily Activity</h3>
          </div>
          <div class="card-body pt-0">
             <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>



        </div>
    </div>
    </div>
</section>
      -->


    <!-- Analytics -->


  </body>
</html>
```

updateprofile.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="/static/registration.css">
     <meta name="viewport" content="width=device-width, initial-scale=1.0">

   </head>
<body>
  <div class="container">
    <div class="title">update</div>
    <div class="content">
      <form action="{{url_for('updateprofile')}}" method="POST" class="login-email">
        <div class="user-details">
          <div class="input-box">
            <span class="details">Username</span>
            <input type="text" placeholder="Enter your height" name="username">
          </div>
          <div class="input-box">
            <span class="details">Height</span>
            <input type="text" placeholder="Enter your height" name="height">
          </div>
          <div class="input-box">
            <span class="details">Weight</span>
            <input type="text" placeholder="Enter your weight" name="weight">
          </div>
          <div class="input-box">
            <span class="details">Gender</span>
            <select name="gender">
              <option value="Male">Male</option>
              <option value="Female">Female</option>
            </select>
          </div>
          <div class="input-box">
            <span class="details">Blood</span>
            <input type="text" placeholder="Enter your Blood group" name="blood">
          </div>



        </div>
        <span>{{msg}}</span>
        <div class="button">
          <input type="submit" href="{{url_for('userprofile')}}" value="update">

        </div>
      </form>
    </div>
  </div>

</body>
</html>
```

verify.html

```
<!DOCTYPE html>
```

```
 <html>
 <head>
     <title>index</title>
 </head>
 <body>
    <style>
                                                                        @import
url("https://fonts.googleapis.com/css2?family=Sansita+Swashed:wght@600&display=sw
ap");
body {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  background: linear-gradient(45deg, #9b59b6, #71b7e6);
  font-family: cursive;



}
.center {
  position: relative;
  padding: 50px 50px;
  background: #fff;
  border-radius: 10px;
}
.center h1 {
  font-size: 2em;
  border-left: 5px solid dodgerblue;
  padding: 10px;
  color: #000;
  letter-spacing: 5px;
  margin-bottom: 60px;
  font-weight: bold;
  padding-left: 10px;
}
.center .inputbox {
  position: relative;
  width: 300px;
  height: 50px;
  margin-bottom: 50px;
}
.center .inputbox input {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  border: 2px solid #000;
  outline: none;
  background: none;
  padding: 10px;
  border-radius: 10px;
  font-size: 1.2em;
}
.center .inputbox:last-child {
  margin-bottom: 0;
}
.center .inputbox span {
  position: absolute;
  top: 14px;
  left: 20px;
  font-size: 1em;
  transition: 0.6s;
  font-family: sans-serif;
}
.center .inputbox input:focus ~ span,
.center .inputbox input:valid ~ span {
  transform: translateX(-13px) translateY(-35px);
  font-size: 1em;
}
.center .inputbox [type="button"] {
  width: 50%;
  background: dodgerblue;
  color: #fff;
  border: #fff;
}
.center .inputbox:hover [type="button"] {
  background: linear-gradient(45deg, #71b7e6, #9b59b6);
}
   </style>
<form action = "{{ url_for('validate') }}" method="post">
   <div class="center">
      <h1>Please enter your OTP here!!</h1>
```

```
        <h3>OTP</h3>
          <div class="inputbox">

            <input type="text" name="otp">



          </div>
          <div class="inputbox">
            <input type="submit"  value="submit" value="Continue">
          </div>

      </div>
 </form>
 </body>
 </html>
```

window.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Nutrition_Assistant</title>
    <link rel='stylesheet' href='https://fonts.googleapis.com/css?family=Rubik:400,700'>
    <link rel="stylesheet" href="/static/window.css">
```

```
</head>


<body>
    <script>
        function act(e){
          e.preventDefault()
          console.log(e.target.file.files[0])
          alert("hii")
        }
      </script>
    <div class="windows">
        <form action="{{ url_for('window') }}" method="POST">
            <h1>What are the nutrition value present in your food just type to know.</h1>
           <div class="row">
            <div class="foodname">
                <h2>Food name :</h2>
                <input type="text" class="food-name" name="foodname">


            </div><br>
            <div class="imagesearch">
                <h2>Upload picture:</h2>
                        <input  type="file"  accept="image/*"  class="request-image"  id="image"
name="image">
                <input class="btn btn-outline-primary" type="submit" value="Submit">
            </div>
        </div>


        </form>
    </div>
</body>

</html>
```

email.html

```
<!DOCTYPE html>
 <html>
 <head>
     <title>index</title>
 </head>
 <body>
    <style>
                                                                      @import
url("https://fonts.googleapis.com/css2?family=Sansita+Swashed:wght@600&display=sw
ap");
body {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  background: linear-gradient(45deg, #9b59b6, #71b7e6);
  font-family: cursive;



}
```

```
.center {
  position: relative;
  padding: 50px 50px;
  background: #fff;
  border-radius: 10px;
}
.center h1 {
  font-size: 2em;
  border-left: 5px solid dodgerblue;
  padding: 10px;
  color: #000;
  letter-spacing: 5px;
  margin-bottom: 60px;
  font-weight: bold;
  padding-left: 10px;
}
.center .inputbox {
  position: relative;
  width: 300px;
  height: 50px;
  margin-bottom: 50px;
}
.center .inputbox input {
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  border: 2px solid #000;
  outline: none;
  background: none;
  padding: 10px;
  border-radius: 10px;
  font-size: 1.2em;
}
.center .inputbox:last-child {
  margin-bottom: 0;
}
.center .inputbox span {
  position: absolute;
  top: 14px;
  left: 20px;
  font-size: 1em;
  transition: 0.6s;
  font-family: sans-serif;
}
.center .inputbox input:focus ~ span,
.center .inputbox input:valid ~ span {
  transform: translateX(-13px) translateY(-35px);
  font-size: 1em;
}
.center .inputbox [type="button"] {
  width: 50%;
  background: dodgerblue;
  color: #fff;
  border: #fff;
}
```

```
.center .inputbox:hover [type="button"] {
  background: linear-gradient(45deg, #71b7e6, #9b59b6);
}
    </style>
 <form action = "{{ url_for('verify') }}" method = "post">
    <div class="center">
        <h1>Please verify your mail id here!!</h1>

          <div class="inputbox">
            <input type="text" required="required" name="email1">
            <span>Email</span>
          </div>
          <div class="inputbox">
            <input type="submit"  value="submit">
          </div>

      </div>
 </form>
 </body>
 </html>
```

details.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>User Profile Page</title>


        <meta name="author" content="Codeconvey" />
                                                                                                                        <link
href="https://fonts.googleapis.com/css?family=Lato:300,400,700,900&display=swap"
rel="stylesheet"><link rel='stylesheet' href='https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.3/css/bootstrap.min.css'>
                <link      rel='stylesheet'      href='https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.12.1/css/all.min.css'>



        <link rel="stylesheet" href="/static/userprofile1.css" />



            <link rel="stylesheet" href="/static/userprofile2.css">
    </head>
<body>
    <div class="col-lg-12">
        <div class="card shadow-sm">
          <div class="card-header bg-transparent border-0">
                    <h3  class="mb-0"><i  class="far  fa-clone  pr-1"></i>General  Information
&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&n
bsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&n
bsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbs
p&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&
nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&n
bsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
            <a class="bottom" href="{{url_for('userprofile')}}"> Profile</a></h3>
          </div>
```

```
  <div class="card-body pt-0">
    <table class="table table-bordered">
      <tr>
        <th width="30%">Height</th>
        <td width="2%">:</td>
        <td>{{height}}</td>
      </tr>
      <tr>
        <th width="30%">Weight</th>
        <td width="2%">:</td>
        <td>{{weight}}</td>
      </tr>
      <tr>
        <th width="30%">Gender</th>
        <td width="2%">:</td>
        <td>{{gender}}</td>
      </tr>
      <tr>
        <th width="30%">Blood</th>
        <td width="2%">:</td>
        <td>{{blood}}</td>
      </tr>
    </table>
  </div>
</div>
<!-- button -->
```

```
      </div>
    </div>
  </div>
</div>

</body>
</html>
```

service.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>
    <link rel="stylesheet" href="/static/services.css">

    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.3.1/css/all.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>

    </style>
  </head>

  <body>
    <div class="wrapper">

      <div class="logo">
          <img src="{{ user_image}}" alt="">
      </div>
      <ul class="nav-area">
        <li><a href="{{url_for('homepage')}}">Back</a></li>



      </ul>
    </div>
     <div class="services">
      <h1>Our Services</h1>
      <div class="cen">
        <div class="service">
          <i class="fas fa-apple-alt"></i>
          <h2>Image Recogonition</h2>
          <p>Upload a fruit and vegetable image and know the nutritional values.</p>
        </div>

        <div class="service">
          <i class="fab fa-android"></i>
          <h2>Nutrition Assistant</h2>
          <p>Helps to maintain nutritional diet.</p>
        </div>

        <div class="service">
          <i class="fab fa-angellist"></i>
          <h2>Help</h2>
          <p>Contact us through <a href="mailto:
nassistant.gans@gmail.com">nassistant.gans@gmail.com</a>.</p>
        </div>
      </div>
    </div>

  </body>
</html>
```

about.html

```
<!DOCTYPE html>
<html>

<head>
    <title>About us Page</title>
    <link rel="stylesheet" href="/static/about.css">
    <!-- <link rel="stylesheet" href="homepage.css"> -->
</head>
<body>

    <div class="wrapper">

        <div class="logo">
            <img src="{{ user_image}}" alt="">
        </div>
        <ul class="nav-area">
            <li><a href="{{url_for('homepage')}}">Home</a></li>
        </ul>
    </div>




    <section class="background firstsection">
        <div class="box-main">
            <div class="firstHalf">
                <p class="text-big">About US</p>

                <p class="text-small" style="color: black;">
                        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<span style="color: white;">Clarifai's AI-Driven Food
Detection Model</span>  for accurate food identification and Food API's to give the
nutritional value of the identified food.
                </p>
                <br>
                <p class="center"><a href="#Order"
                style="text-decoration:none;color:rgb(9, 10, 98);">
                    Below are the people who
                    works in our project</a>
                </p>
            </div>
        </div>
    </section>
    <section class="service">
        <h1 class="h-primary center" style="margin-top:30px;text-align:center;">
```

```
            Our Team
        </h1>
    <div id="services">
            <div class="box">
                <img src=
"/static/pics/Ashwin.jpg"
                    alt="picture goes here">

                <p class="center">
                    <a href="#xyz" style="text-decoration:none;color:black;
        font-weight:bold;font-family: 'Langar', cursive;">
                        ASHWIN KUMAR
                    </a>
                </p>
                <p style="font-family: sans-serif">TEAM LEADER</p>
            </div>
            <div class="box">
                <img src=
"/static/pics/GAN.jpeg"
                    alt="picture goes here">

                <p class="center">
                    <a href="#abc" style="text-decoration:none;color:black;
        font-weight:bold;font-family: 'Langar', cursive;">
                        GANESAN
                    </a>
                </p>
        <p style="font-family: sans-serif ">TEAM MEMBER</p>

      </div>
      <div class="box">
        <img src="/static/pics/SRI.jpeg"
          alt="picture goes here">

        <p class="center">
          <a href="#abc" style="text-decoration:none;color:black;
    font-weight:bold;font-family: 'Langar', cursive;">
            SRIRAM
          </a>
        </p>

        <p style="font-family: sans-serif ">TEAM MEMBER</p>

      </div>
      <div class="box">
        <img src=
"/static/pics/NIVI.jpg"
          alt="picture goes here">
        <br>
        <p class="center">
          <a href="#xyz" style="text-decoration:none;color:black;
    font-weight:bold;font-family: 'Langar', cursive;">
            NIVENDHAN
          </a>
        </p>
```

```
            <p style="font-family: sans-serif ">TEAM MEMBER</p>



          </div>
        </div>


    </section>


    <footer class="background">
        <p class="text-footer">
            NUTRITION ASSISTANT APPLICATION
        </p>


    </footer>
</body>


</html>
```

Nutrition.py

```
from flask import Flask, render_template, request, redirect, url_for, session, flash
import ibm_db
import re
import requests
from random import *
from clarifai_grpc.grpc.api import service_pb2, resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc
from flask_mail import Mail, Message
import os
from flask_mail import Mail, Message
app = Flask(__name__)

mail = Mail(app) # instantiate the mail class
# configuration of mail
app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'nassistant.gans@gmail.com'
app.config['MAIL_PASSWORD'] = 'ddlomuragdcdyojh'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True
mail = Mail(app)
otp = randint(000000,999999)

from clarifai_setup import (
    DOG_IMAGE_URL,
    GENERAL_MODEL_ID,
    NON_EXISTING_IMAGE_URL,
    RED_TRUCK_IMAGE_FILE_PATH,
    both_channels,
    metadata,
    raise_on_failure,
    post_model_outputs_and_maybe_allow_retries,
)
```

```
def test_predict_image_url():
    stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())

    req = service_pb2.PostModelOutputsRequest(
        model_id=GENERAL_MODEL_ID,
        inputs=[
            resources_pb2.Input(
                data=resources_pb2.Data(image=resources_pb2.Image(url=DOG_IMAG
E_URL))
            )
        ],
    )

            response = post_model_outputs_and_maybe_allow_retries(stub, req,
metadata=metadata())
    print(response)
    raise_on_failure(response)

    assert len(response.outputs[0].data.concepts) > 0

app.secret_key = 'a'

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-
9991-
629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30119;S
ecurity=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=lbs14903;PW
D=1N4walQ5ywwiwP7c;","","")
```

```
picsfolder = os.path.join('static','pics')
app.config['UPLOAD_FOLDER']=picsfolder

@app.route('/')

@app.route('/homepage')
def homepage():
    icon = os.path.join(app.config['UPLOAD_FOLDER'],'icon.gif')
    return render_template('homepage.html',user_image=icon)

@app.route('/about')
def about():
    icon = os.path.join(app.config['UPLOAD_FOLDER'],'icon.gif')
    return render_template('about.html',user_image=icon)


@app.route('/login', methods =['GET', 'POST'])
def login():
    msg=''
     if request.method=='POST' and 'username' in request.form and 'passwords' in
request.form:
        username = request.form['username']
        passwords = request.form['passwords']
        stmt = ibm_db.prepare(conn,'SELECT * FROM appuser WHERE username =
? AND passwords = ?')
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,passwords)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        if account:
            session['loggedin'] = True
            session['username'] = account['USERNAME']
            msg='Login successful'
            return redirect(url_for('userprofile'))
        else:
            msg='Incorrect username/password'
    return render_template('login.html',msg=msg)

@app.route('/logout')
def logout():
    if 'id' in session:
        session.pop('id',None)
        session.pop('username',None)
        session.pop('passwords',None)
    return redirect(url_for('homepage'))

@app.route('/register', methods =['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST':
        username = request.form['username']
        fullname = request.form['fullname']
        email = request.form['email']
        passwords = request.form['passwords']
        cpassword = request.form['cpassword']
        stmt = ibm_db.prepare(conn,'SELECT * FROM appuser WHERE username =
?')
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'Username must contain only characters and numbers !'
        elif not username or not passwords or not email:
            msg = 'Please fill out the form !'
        else:
                prep_stmt = ibm_db.prepare(conn,"INSERT INTO appuser(username,
fullname, email, passwords, cpassword) VALUES(?, ?, ?, ?, ?)")
            ibm_db.bind_param(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, fullname)
            ibm_db.bind_param(prep_stmt, 3, email)
            ibm_db.bind_param(prep_stmt, 4, passwords)
            ibm_db.bind_param(prep_stmt, 5, cpassword)
            ibm_db.execute(prep_stmt)
            msg = 'You have successfully registered !'
            return render_template('email.html')
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('registration.html', msg = msg)
```

```
@app.route('/userprofile', methods =['GET', 'POST'])
def userprofile():
    if 'username' in session:
        username = session['username']
        stmt = ibm_db.prepare(conn, 'SELECT * FROM appuser WHERE username =
?')
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        acc = ibm_db.fetch_tuple(stmt)
        return render_template('userprofile.html',username = acc[1], fullname = acc[2],
email = acc[3],)
    return render_template('userprofile.html')


@app.route('/updateprofile', methods =['GET', 'POST'])
def updateprofile():
    msg = ''
    if request.method == 'POST':
            username=request.form["username"]
            height = request.form['height']
            weight = request.form['weight']
            gender = request.form['gender']
            blood = request.form['blood']
            prep_stmt = ibm_db.prepare(conn,"INSERT INTO userdetail(username,
height, weight, gender, blood) VALUES(?, ?, ?, ?, ?)")
            ibm_db.bind_param(prep_stmt, 1, username)
            ibm_db.bind_param(prep_stmt, 2, height)
            ibm_db.bind_param(prep_stmt, 3, weight)
            ibm_db.bind_param(prep_stmt, 4, gender)
            ibm_db.bind_param(prep_stmt, 5, blood)
            ibm_db.execute(prep_stmt)
            return redirect(url_for('detail'))
    return render_template('updateprofile.html')

@app.route('/detail', methods =['GET', 'POST'])
def detail():
    if 'username' in session:
        username = session['username']
        stmt = ibm_db.prepare(conn, 'SELECT * FROM  userdetail WHERE username
= ?')
        ibm_db.bind_param(stmt, 1,username)
        ibm_db.execute(stmt)
        acc = ibm_db.fetch_tuple(stmt)
        return render_template('detail.html',height = acc[2], weight = acc[3], gender =
acc[4], blood = acc[5])
    return render_template('detail.html')


@app.route('/window', methods=['POST', 'GET'])
def window():

  # Calorie Ninja
    url = "https://calorieninjas.p.rapidapi.com/v1/nutrition"

    headers = {
```

```
                                                                  "X-RapidAPI-Key":
"aa95b88b45mshe4394a422ce8c48p13a698jsn9d8eb019e144",
        "X-RapidAPI-Host": "calorieninjas.p.rapidapi.com"
    }

    if request.method == 'POST':
        foodname = request.form['foodname']

        querystring = {"query": foodname}
        response = requests.request(
            "GET", url, headers=headers, params=querystring)

        return response.text

    return render_template('window.html')

@app.route('/window', methods=['POST', 'GET'])
def clarifai():
    if request.files.get('image'):
        image = request.files['image'].stream.read()
        stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())

        CLARIFAI_API_KEY = "04fe7a95051541789ba44a08eaa5722e"
        APPLICATION_ID = "Nutrition_Assistant1"

        # Authenticate
        # image = '/home/bala/Desktop/Images/foodsample.jpeg'

        metadata = (("authorization", f"Key {CLARIFAI_API_KEY}"),)

        with open(image, "rb") as f:
            file_bytes = f.read()

        request = service_pb2.PostModelOutputsRequest(
            model_id='9504135848be0dd2c39bdab0002f78e9',
            inputs=[
                resources_pb2.Input(
                    data=resources_pb2.Data(
                        image=resources_pb2.Image(
                            base64=file_bytes
                        )
                    )
                )
            ])
        response = stub.PostModelOutputs(request, metadata=metadata)

        if response.status.code != status_code_pb2.SUCCESS:
            raise Exception("Request failed, status code: " +
                            str(response.status.code))

        for concept in response.outputs[0].data.concepts:
            print('%12s: %.2f' % (concept.name, concept.value))
    return render_template('window.html')


@app.route('/verify', methods=['GET', 'POST'])
def verify():
    if request.method == 'POST':
        email1 = request.form['email1']
        sql = "SELECT * FROM email WHERE email1 = ? "
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,email1)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_tuple(stmt)
        print(account)
        if account:
            msg = 'Account already exists !'
        else:
           insert_sql = "INSERT INTO email(email1) VALUES(?)"
           stmt = ibm_db.prepare(conn,insert_sql)
           ibm_db.bind_param(stmt, 1, email1)
           ibm_db.execute(stmt)
                              msg      =      Message('NUTRITION      ASSISTANT',sender
='nassistant.gans@gmail.com',recipients = [email1])
           msg.body = 'Hello user,THIS IS YOUR ONE TIME PASSWORD'
           msg.body = str(otp)
           mail.send(msg)
           return render_template('verify.html')
    return render_template('email.html')
```

```
@app.route('/validate',methods=['GET', 'POST'])
def validate():
 user_otp = request.form['otp']
 if otp == int(user_otp):
     return render_template('login.html')
 return render_template('verify.html')

@app.route('/services')
def services():
    icon = os.path.join(app.config['UPLOAD_FOLDER'],'icon.gif')
    return render_template('services.html',user_image=icon)

if __name__ == '__main__':
    app.debug = True
    app.run(host='0.0.0.0',port=8080)
```

APPENDIX

GitHub Link

GitHub: **https://github.com/IBM-EPBL/IBM-Project-49681-1660834474**

Demo video link: **https://drive.google.com/file/d/1CIWeefg-t5X6exrsP_T0EEyeOEaa10s0/view**