PROJECT REPORT

NUTRITION ASSISTANT APPLICATION

Team ID: PNT2022TMID02638 **Batch:** B3-3M5E

TEAM LEADER:

Name: SARVESH K	**Register Number:** 2116191001085

TEAM MEMBERS:

Name: SHEIK SAMEERA SABRIN W	**Register Number:** 2116191001089
Name: SUSHMITHA S	**Register Number:**2116191001103
Name: THAANYA S	**Register Number:**2116191001105

CONTENTS

1. INTRODUCTION

1.1 Project Overview

The project titled “Nutrition Assistant Application”, aims to provide a platform for users to make better nutritional choices and lead healthier lives. This is achieved by creating a convenient and easy-to-use application where users can upload images of the food they eat on a day-to-day basis and know about the nutritional value of the food. This can be especially useful for people with health concerns and who need to incorporate more of a certain nutrient in their food. It’s an easy way to educate people about the different nutrients in their foods as well.

1.2 Purpose

With the perpetual health craze, people in the technology era need an apt tool for them to understand nutrition in a convenient way. That’s where our project comes in handy. In this fast-paced world, people don’t have the time to focus on health and on-the-go meals or fast food is rising in popularity. This results in poor health among youth which can impact our future generation. The need of the hour is to make nutrition accessible and help people understand which nutrients are present in which foods.

2. LITERATURE SURVEY

2.1 Existing Problem

It's hard for people to eat a balanced diet, consisting of all the nutrients necessary for good health. Sometimes they end up overconsuming one type of food which makes them miss out on other nutrients.

2.2 References

1. **Development of a cloud-based solution for effective nutrition intervention in the management of lifestyle diseases.**

<u>Authors:</u> Manju P George, C. A. Kalpana <u>Year:</u> 2020

This paper proposes a system that aims to bridge the gap between clinical nutrition and the common man. For the purpose of prescribing therapeutic nutrition in clinical settings, a web-based application is being developed. The cloud-based solution would be able to figure out the nutritional needs and automatically direct first-line nutritional treatment to patients and clients. Additionally, it functions as an electronic medical and dietetic record, allowing for the planning of a customised nutrition counselling approach around the client's hectic schedule. One method is much simpler, and the client can speak with his or her personal nutritionist in a setting that suits them.

2. Cloud-Based Meta learning System for Predictive Modeling of Biomedical Data

Authors: Milan VukiTeviT, Sandro RadovanoviT, Miloš MilovanoviT, and Miroslav MinoviT
Year: 2013

This research presented a cloud-based infrastructure for biomedical big data storage, processing, and predictive modelling. The meta-learning system is added to the existing service-based cloud architecture as a knowledge service that is data and model driven. We supported community-based data and algorithm collecting as part of the suggested architecture because it is a crucial prerequisite for the high quality of meta-learning. Through a platform for the development and execution of distributed data mining processes and algorithms, this research field can advance and gain new value. Finally, we provide data- and model-driven decision help for choosing the optimal biomedical data processing techniques.

3. DeepFood: Automatic Multi-Class Classification of Food Ingredients Using Deep Learning

Authors: Lili Pan, Samira Pouyanfar, Hao Chen, Jiaohua Qin Year: 2017

This study suggests the DeepFood framework, which combines various deep feature sets, a number of feature selections, and an improved classifier known as SMO to automatically multi-class categorise food items using deep learning. The architecture is made to categorise small to medium-sized datasets, which is a highly common and essential task in practical applications.

4. Study for Food Recognition System Using Deep Learning

Authors: Nareen O. M.Salim, Subhi R. M. Zeebaree, Mohammed A.M.Sadeeq A.H Radie
Year: 2013

This paper reviewed a significant number of recent articles on the APP on the deep learning of foodstuffs, and it went into detail about each article's structure, training methodology, and final assessment results of the deep learning for processing the food picture, spectrum, text, and other details. In terms of effectiveness, we compared deep learning to other widely used methodologies and found that, in these evaluated studies, deep learning produces superior results to other approaches. This essay discussed crucial Food Recognition. According to the literature study, food recognition is aided by a number of active mechanisms. The researchers have successfully used a variety of strategies and algorithms to accomplish this goal.

5. **Deep feature extraction technique based on Conv1D and LSTM network for food image recognition**

Authors: **Sirawan Phiphitphatphaisit, Olarik Surinta** **Year: 2021**

The ResNet50+Conv1D-LSTM network was suggested in this study for precise food image identification. The reliable spatial features were first extracted. Second, the Conv1D network linked with the long short-term memory (LSTM) network, known as Conv1D-LSTM, employed robust characteristics as input data. The Conv1D-LSTM network's main job was to extract a temporal characteristic. The output of the Conv1D-LSTM was then converted into a probability distribution using the softmax algorithm.

6. **Automatic Fruits Detection Using Artificial Intelligence**

Authors: Tejswini Balpande, Nikita Dhothkar, Heena Satpute, Namrata Durbude
Year: 2020

In this study, we employ AI, which uses a camera to automatically determine the fruit's quality. The image can be scanned using a camera, after which image processing is carried out to identify the fruits' calories and other characteristics such as form, size, colour, and texture. In this study, image enhancement is the primary goal of image processing in order to reduce undesired noise and provide a better image. The camera's image is enhanced using image processing, which also identifies fruit characteristics like size, colour, and calories.

7. **Enhancing Cloud and Big Data Systems for healthy Food and Nutrition Information Systems Practice: A Conceptual Study**

Authors: P.K. Paul1, P.S. Aithal2, A. Bhuimali3 Year: 2019

This essay clarified the fundamentals of cloud computing, such as its fundamental attributes and functionalities. It also aided in our understanding of the primary difficulties associated with cloud computing and related technologies in the context of poor nations.

8. **Mobile cloud-based system recognizing nutrition and freshness of food image**

Authors: Diptee Kumbhar, Sarita Patil Year:
2017

The framework presented in this research offers clients practical and clever methods that let them keep tabs on their calorie consumption and measure their food intake. Our system's food recognition method makes use of a cloud computing environment with classifier machine learning and a Naive Bayes training mechanism. Using image processing techniques, this device also verifies the fruit's freshness. The precision of the procedure used to measure calorie consumption is improved by this technology.

9. Dietary Nutrition Cloud Platform Technology Based on Big Data

Author: Muhammad Jmail
Year: 2021

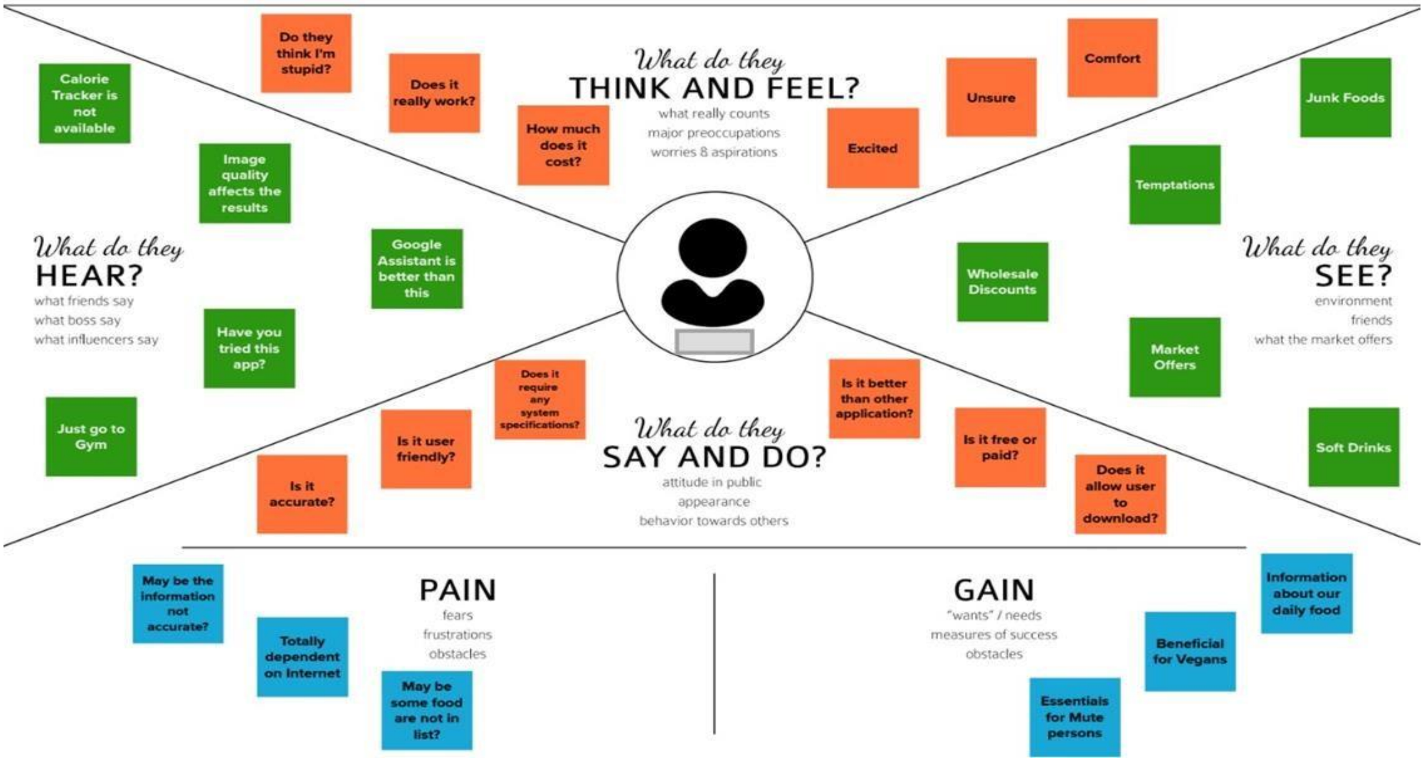
In order to identify the relationship between dietary intake and disease, this paper analyses the characteristics of the food nutrition cloud platform, disease, and nutrition intake in great detail. It also identifies the drawbacks of the traditional association rule algorithm in the dietary nutrition cloud platform and suggests an improved immune algorithm based on clustering. The method can speed up association rule searches and can instantly locate the desired number of frequent item sets.

2.3 Problem Statement Definition

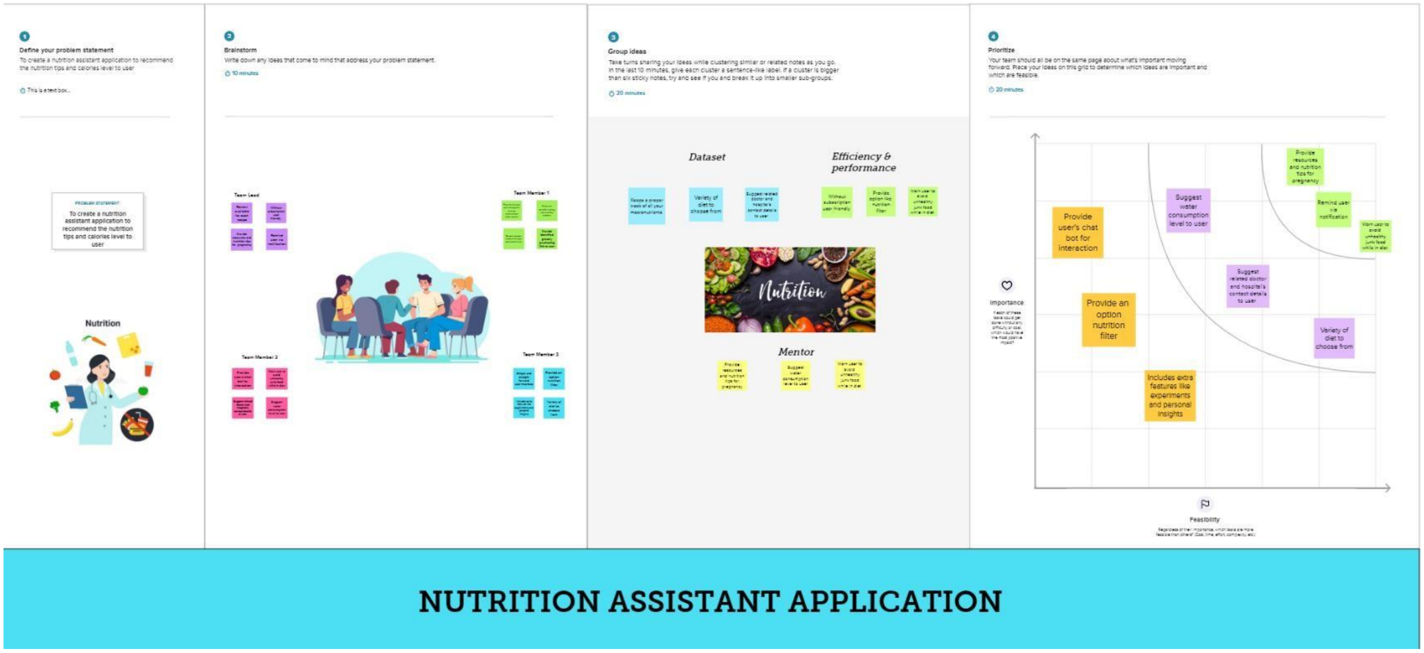
Obesity rates are rising alarmingly quickly as a result of people's lack of knowledge about appropriate eating practices, which reflects the hazards to their health. The simplest way to prevent obesity is for people to limit their daily calorie consumption by eating healthier meals. It's still not very convenient for people to use app-based nutrient dashboard systems, even though food packaging includes nutrition (and calorie) labels. These systems can analyse real-time images of a meal and analyse it for nutritional content, which can be very handy and improve dietary habits and subsequently help with maintaining a healthy lifestyle.

3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming



1 Define your problem statement

To create a nutrition assistant application to recommend the nutrition tips and calories level to user

This is a text box...

PROBLEM STATEMENT

To create a nutrition assistant application to recommend the nutrition tips and calories level to user

2 Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

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

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.

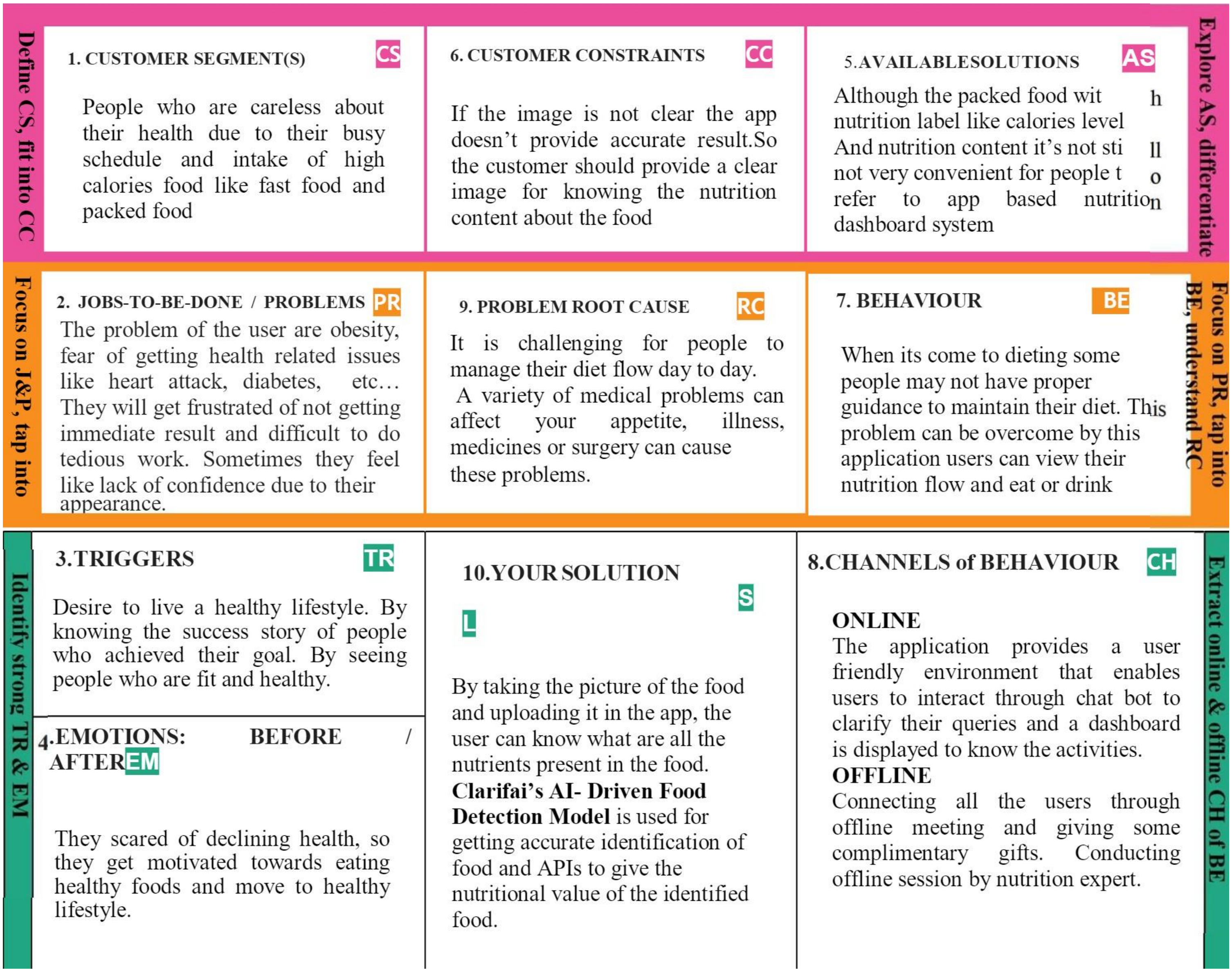
20 minutes

Nutrition
Team Lead
Team Member 1
Team Member 2
Team Member 3
Dataset
Keeps a proper track of all your macronutrients
Variety of diet to choose from
Suggest related doctor and hospital's contact details to user
Efficiency & performance
Without subscription user friendly
Provide option like nutrition filter
Warn user to avoid unhealthy junk food while in diet
Nutrition
Mentor
Provide resources and nutrition tips for pregnancy
Suggest water consumption level to user
Warn user to avoid unhealthy junk food while in diet
Importance
Provide resources and nutrition tips for pregnancy
Suggest water consumption level to user
Provide user's chat bot for interaction
Remind user via notification
Warn user to avoid unhealthy junk food while in diet
Suggest related doctor and hospital's contact details to user
Provide an option nutrition filter
Variety of diet to choose from
Includes extra features like experiments and personal insights
Feasibility
NUTRITION ASSISTANT APPLICATION

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Rate of Obesity are increasing at an high speed,due to the ignorance of the proper Nutrition foods, and this leads to risks in people's health. People need to control their daily calorie intake by eating healthier foods, which is the most basic method to avoid obesity.However,some food packaging has an added nutrition and calorie values,but it's not very comfortable to refer.
2.	Idea / Solution description	The solution is user can know the nutritional content of the food they are intaking,by taking picture of the food and upload it in the app.It is used for get accurate food identification and APIs to give the nutritional value of the identified food.
3.	Novelty / Uniqueness	Provides a user-friendly environment.provides recipes according to their diet.Provides different ways to access the nutritional information about the food by taking the snap of the food,upload in the gallery and entering manually.
4.	Social Impact / Customer Satisfaction	Getting feedbacks from the users for enhancement and giving notification on their diet plans and goal tracking.
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 membership will have extra benefits.
6.	Scalability of the Solution	People can access it from anywhere at anytime to track the calories and nutrition value that will improve a healthy eating pattern.This app will improves the dietary habits and helps in maintaining healthy weight and healthy lifestyle.

3.4 Problem Solution Fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

People who are careless about their health due to their busy schedule and intake of high calories food like fast food and packed food

6. CUSTOMER CONSTRAINTS CC

If the image is not clear the app doesn't provide accurate result.So the customer should provide a clear image for knowing the nutrition content about the food

5.AVAILABLESOLUTIONS AS

Although the packed food with nutrition label like calories level And nutrition content it's not still not very convenient for people to refer to app based nutrition dashboard system

Explore AS, differentiate

Focus on J&P, tap into

2. JOBS-TO-BE-DONE / PROBLEMS PR

The problem of the user are obesity, fear of getting health related issues like heart attack, diabetes, etc… They will get frustrated of not getting immediate result and difficult to do tedious work. Sometimes they feel like lack of confidence due to their appearance.

9. PROBLEM ROOT CAUSE RC

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

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

7. BEHAVIOUR BE

When its come to dieting some people may not have proper guidance to maintain their diet. This problem can be overcome by this application users can view their nutrition flow and eat or drink

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

3.TRIGGERS TR

Desire to live a healthy lifestyle. By knowing the success story of people who achieved their goal. By seeing people who are fit and healthy.

4.EMOTIONS: BEFORE / AFTER EM

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

10.YOUR SOLUTION S L

By taking the picture of the food and uploading it in the app, the user can know what are all the nutrients present in the food. **Clarifai's AI- Driven Food Detection Model** is used for getting accurate identification of food and APIs to give the nutritional value of the identified food.

8.CHANNELS of BEHAVIOUR CH

ONLINE

The application provides a user friendly environment that enables users to interact through chat bot to clarify their queries and a dashboard is displayed to know the activities.

OFFLINE

Connecting all the users through offline meeting and giving some complimentary gifts. Conducting offline session by nutrition expert.

Extract online & offline CH of BE

4 REQUIREMENT ANALYSIS

4.1 Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Data collection	Collection of required input data

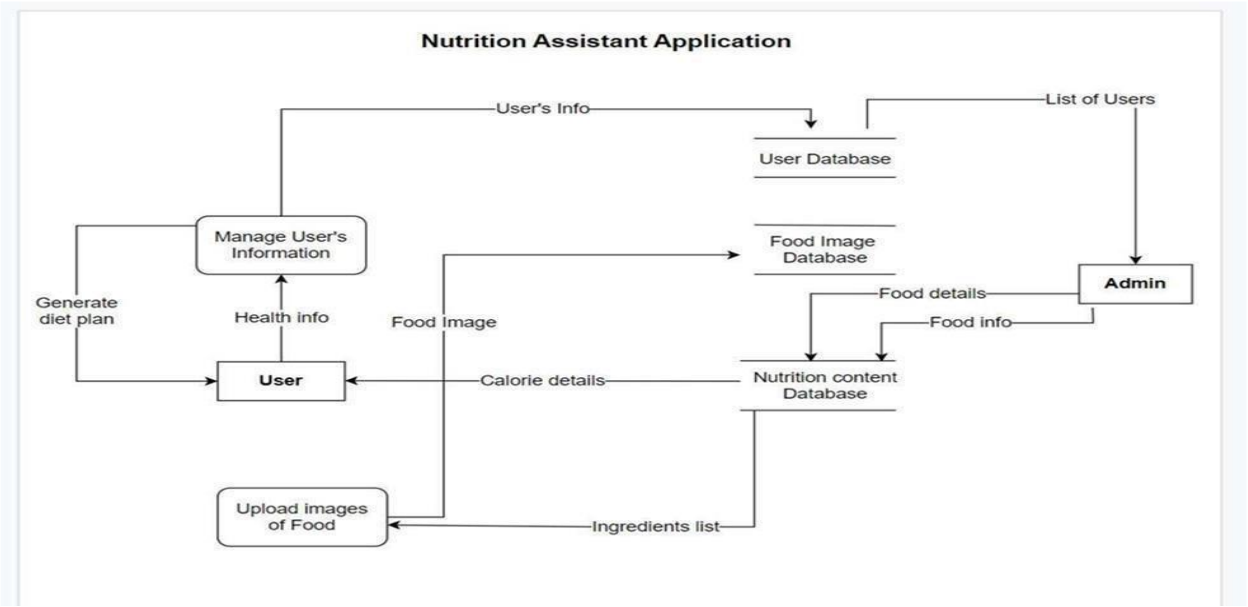
FR-4	Data analysis	Process the given inputs using CNN and Nutrion API
FR-5	Data processing	Evaluate the data and store it in database and integrate in cloud containers
FR-6	Provide output to user	Display the result to the user

4.2 Non-Functional Requirements

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	User-friendly and overall satisfaction of the user while using the website
NFR-2	**Security**	The website provides proper authentication and verification
NFR-3	**Reliability**	The site always provides reliable outputs and lacks failures
NFR-4	**Performance**	Provides 100% efficiency of the output
NFR-5	**Availability**	The product is readily available for all kinds of users when needed
NFR-6	**Scalability**	Effective in obtaining good accuracies

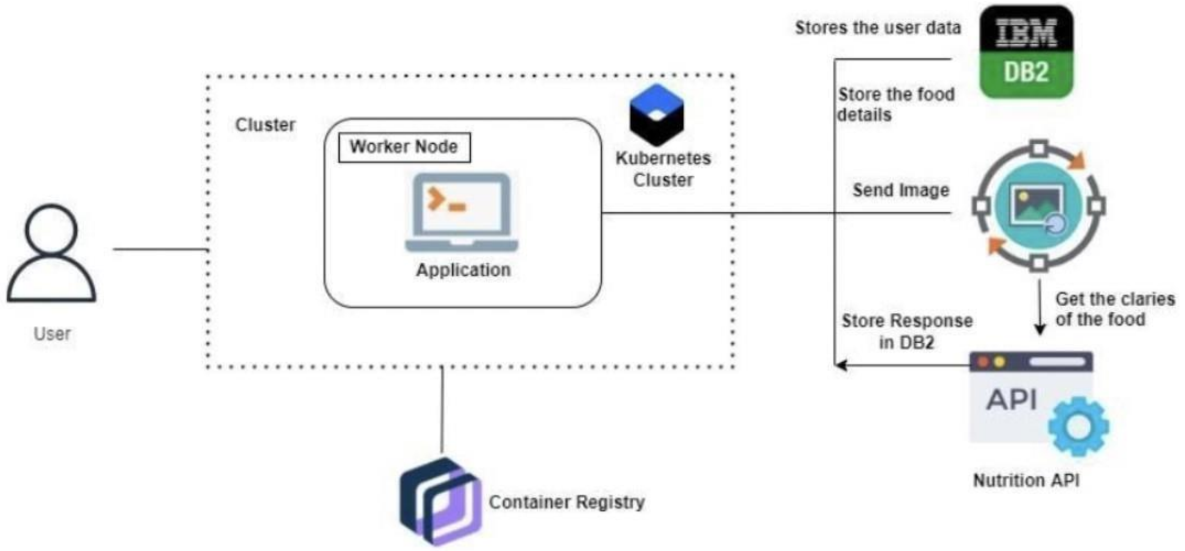
5 PROJECT DESIGN

5.1 Data Flow Diagram

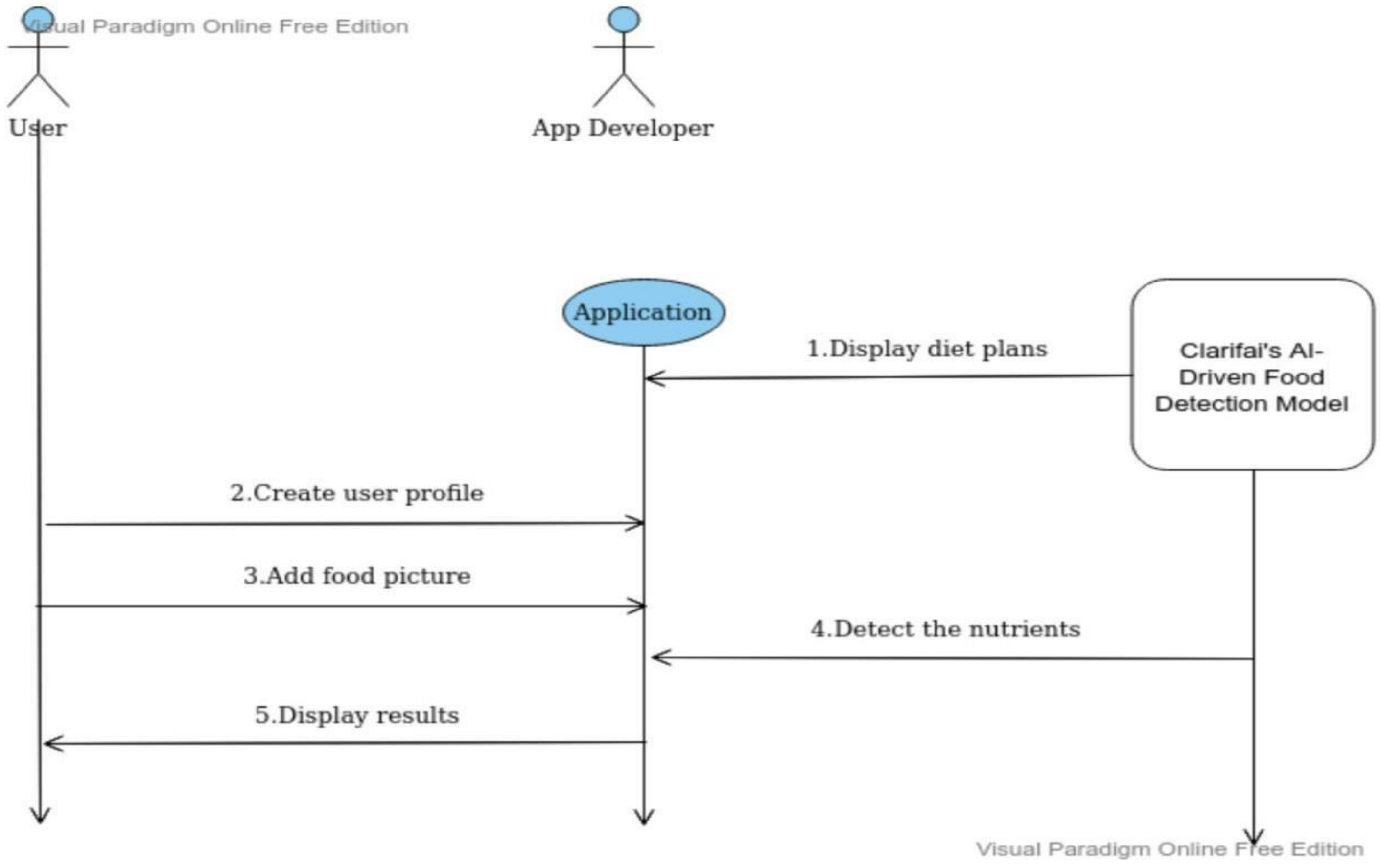


5.2 Solution and Technical Architecture

Technical Architecture



Solution Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile use)	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 / dashboard	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 log into the application by entering email & password	I can access the Dashboard and the application.	High	Sprint-1
Customer (Webuser)	Upload Photo	USN-4	As a user , I can upload the food photo.	I can get the nutrition details.	High	Sprint-1
Administrator	User details	USN-5	As a user , I can fill the Details.	I can get whether the scanned food is suitable or not.	High	Sprint-2
	Push notification	USN-6	As a user, I will search the food items.	I can get the notification, related to my search.	High	Sprint-1
	Shown the nutrition details	USN-7	As a user, I can scan the food.	I can get the nutrition details of the scanned food.	High	Sprint-1

6 PROJECT PLANNING AND EXECUTION

6.1 Sprint Planning and Scheduling

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	Sarvesh K Thaanya S Sushmitha S Sheik Sameera Sabrin W
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Sarvesh K Thaanya S Sushmitha S Sheik Sameera Sabrin W
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	1	High	Sarvesh K Thaanya S Sushmitha S Sheik Sameera Sabrin W
Sprint-2	User details	USN-4	As a user , I can fill the Details.	2	High	Sarvesh K Thaanya S Sushmitha S Sheik Sameera Sabrin W
Sprint-3	Push notification	USN-5	As a user, I will search the food items.	2	Medium	Sarvesh K Thaanya S Sushmitha S Sheik Sameera Sabrin W
Sprint-4	Shown the nutrition details and the recipe for scanned food		As a user, I can scan the food and get the details and Recipe for related food.			Sarvesh K Sushmitha S Thaanya S Sheik Sameera Sabrin w

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	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

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

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

Average Velocity = Story Points per Day

Average Velocity = Story Points per Day

Sprint Duration = Number of (Duration) days per Sprint

Velocity= points per sprint

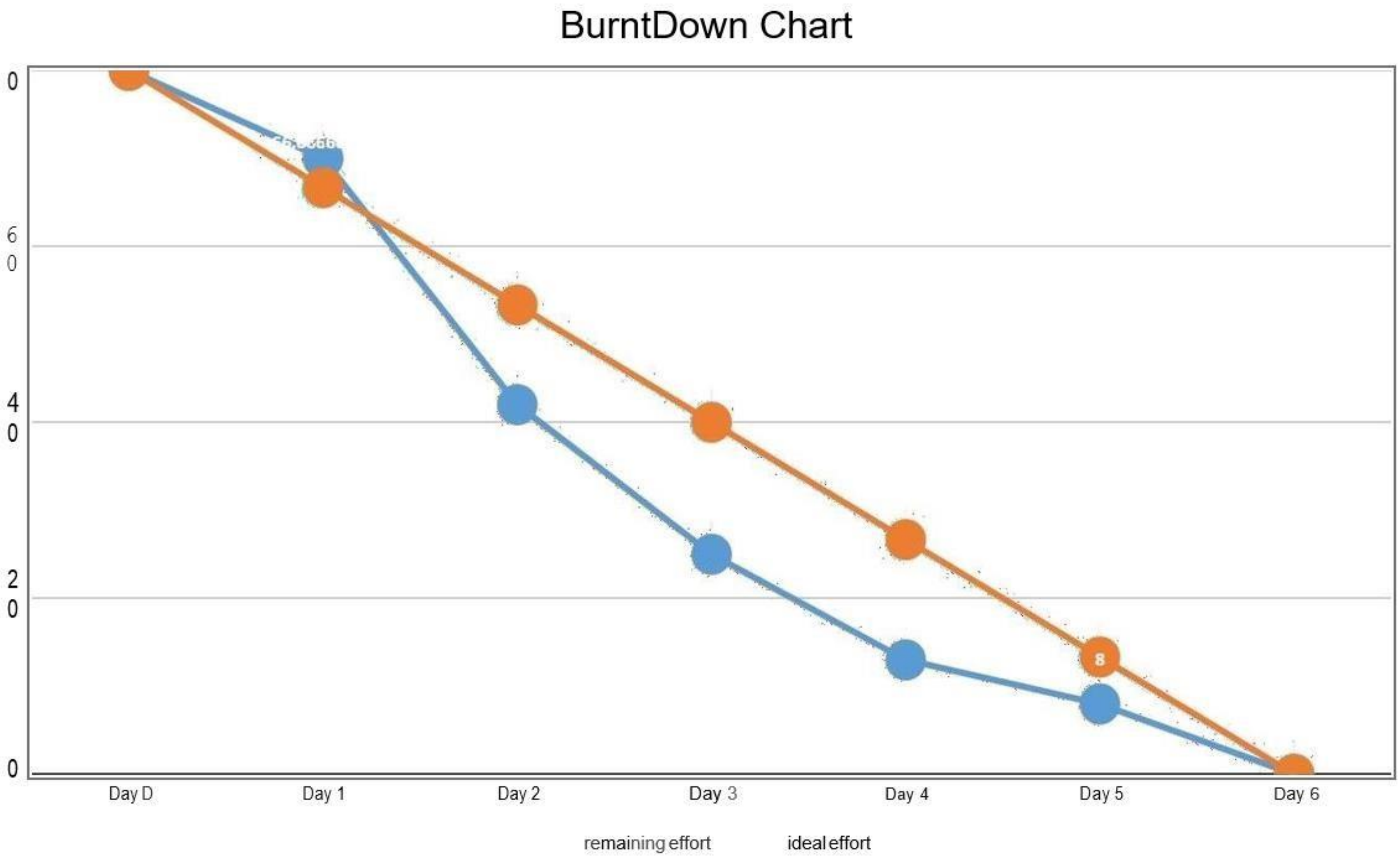
Therefore, the **AVERAGE VELOCITY IS 4 POINTS PER SPRINT**

Burndown Chart:

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

	Initial Estimate	24-Oct	25-Oct	26-Oct	27-Oct	28-Oct	29-Oct
Sprint number	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
Sprint-1	20	0	10	5	3	1	1
Sprint-2	20	2	10	4	1	1	2
Sprint-3	20	5	5	5	5	0	0
Sprint-4	20	3	3	3	3	3	5
remaining effort	80	70	42	25	13	8	0
ideal effort	80	66.66666667	53.33333333	40	26.66666667	13.33333333	0

BURNDOWN CHART



3.3.1 CODING & SOLUTIONING

7.1 Feature 1

Python Flask

Python Flask is used to develop chatbot applications using python. Flask is mainly used to render and integrate the nutrition assistant application in the browser by providing API. By running the python application, the suitable server domain link is obtained and run in the browser.

HTML

The HTML and CSS is used to design the overall nutrition assistant application's U**I.** HTML is used to add UI components and CSS is used to add style to those components.

Build PYTHON FLASK Code:

app.py

```
import ibm_db as db
from flask import Flask, render_template, request, redirect, session,
abort import os import pathlib import requests
from dotenv import load_dotenv from
sendgrid import SendGridAPIClient from
sendgrid.helpers.mail import Mail from
google.oauth2 import id_token from
google_auth_oauthlib.flow import Flow
from pip._vendor import cachecontrol
import google.auth.transport.requests

# Configure Flask app app =
Flask(__name__) SECRET_KEY =
os.urandom(32)
app.config['SECRET_KEY'] =
SECRET_KEY

# Load .env file
load_dotenv()


connection_string = "DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-
a1fc-
1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=3
1249;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=pbg6
0128;PWD=KwrdTHmsmD5GqtJn",'',''.format(DATABASE_NAME, HOSTNAME,
PORT_NUMBER, USERNAME, PASSWORD)
conn = db.connect(connection_string, "", "")

# Frequently used variables
SIGN_UP_PAGE_URL = '/'
LOG_IN_PAGE_URL = '/login'
HOME_PAGE_URL = '/home'
GOOGLE_LOGIN_PAGE_URL = '/google_login'
PROFILE_PAGE_URL = '/profile'
CHANGE_PASSWORD_URL = '/changepwd'
```

```
# Google Auth Configuration
os.environ["OAUTHLIB_INSECURE_TRANSPORT"] = "1"

client_secrets_file = os.path.join(pathlib.Path(__file__).parent,
"client_secret.json")

flow = Flow.from_client_secrets_file(
client_secrets_file=client_secrets_file,
   scopes=["https://www.googleapis.com/auth/userinfo.profile",
"https://www.googleapis.com/auth/userinfo.email", "openid"],
redirect_uri="http://127.0.0.1:5000/callback"
)
# Helper Function to execute SQL queries def
execute_sql(statement, **params):     global
conn
   stmt = db.prepare(conn, statement)

   param_id = 1     for key, val in
params.items():
      db.bind_param(stmt, param_id, val)
      param_id += 1

   result = ''
try:
      db.execute(stmt)        result
= db.fetch_assoc(stmt)     except:
pass

   return result

# Creates user table if not exists
create_table = "CREATE TABLE IF NOT EXISTS user(email varchar(30), username
varchar(30), password varchar(30))"
execute_sql(statement=create_table)

# Helper function to send confirmation mail on sign
in def send_confirmation_mail(user, email):
message = Mail(
      from_email="nutrition@gmail.com",
      to_emails=email,
    subject="YAYY!! Your Account was created successfully!",
html_content= "<strong>Account Created with username
{0}</strong>".format(user)
  )

try:
    sg = SendGridAPIClient(os.environ.get('xxxxxxxxxxxxxxxxx'))
response = sg.send(message)        print(response.status_code)
print(response.body)        print(response.headers)     except
Exception as e:        print(e)

# Sign up page
@app.route(SIGN_UP_PAGE_URL, methods=['GET',
'POST']) def signup():     msg = ''

  if session.get('user'):
    return redirect(HOME_PAGE_URL)

  if request.method == 'POST':
user = request.form['user']
email = request.form['email']
    password = request.form['password']

    duplicate_check = "SELECT * FROM user WHERE username=?"
account = execute_sql(statement=duplicate_check, user=user)

    if account:
      msg = "There is already an account with this username!"
else:
      insert_query = "INSERT INTO user values(?, ?, ?)"
execute_sql(statement=insert_query, email=email, user=user, password=password)

      send_confirmation_mail(user, email)
return redirect(LOG_IN_PAGE_URL)
  return render_template('signup.html', msg=msg)

# Login page
@app.route(LOG_IN_PAGE_URL, methods=['GET',
'POST']) def login():     msg = ''
```

```
    if session.get('user'):
        return redirect(HOME_PAGE_URL)

    if request.method == "POST":

        user = request.form['user']
password = request.form['password']

        duplicate_check = "SELECT * FROM user WHERE username=?"
account = execute_sql(statement=duplicate_check, user=user)

        print(account)        if account and
account['PASSWORD'] == password:
session['user'] = user            return
redirect(HOME_PAGE_URL)        elif account and
account['PASSWORD'] != password:
            msg = 'Invalid Password!'
else:
            msg = "Invalid Username!"

    return render_template('login.html', msg=msg)

# Login using Gmail
@app.route(GOOGLE_LOGIN_PAGE_URL , methods=['GET','POST']) def
google_login():
    authorization_url, state =
flow.authorization_url()     session["state"] = state
return redirect(authorization_url)

# Configuring user credentials after gmail login
@app.route("/callback") def
callback():
    flow.fetch_token(authorization_response=request.url)

    if session["state"] != request.args["state"]:
abort(500)  # State does not match!

    credentials = flow.credentials
request_session = requests.session()
    cached_session = cachecontrol.CacheControl(request_session)
    token_request = google.auth.transport.requests.Request(session=cached_session)

    id_info = id_token.verify_oauth2_token(
id_token=credentials._id_token,
request=token_request,
audience=GOOGLE_CLIENT_ID,
        clock_skew_in_seconds=10
    )

    session["user"] = id_info.get("email")
session["google_id"] = id_info.get("sub")
session["name"] = id_info.get("name")
    return redirect(HOME_PAGE_URL)

# Home page
@app.route(HOME_PAGE_URL, methods=['GET', 'POST'])
def homepage():     if not session.get('user'):
        return redirect(LOG_IN_PAGE_URL)

    msg = ''     if request.method
== 'POST':        if
request.form['food']:
            msg = 'Image Uploaded Successfully!'
else:
            msg = "Image wasn't uploaded, Try again!"

    return render_template('homepage.html', user=session.get('user'),
msg=msg)

# Profile page
@app.route(PROFILE_PAGE_URL, methods=['GET',
'POST']) def profile():     if not session.get('user'):
        return redirect(LOG_IN_PAGE_URL)

    sqlst = "select email from user where username=?"
    user = session.get('user')
    email = execute_sql(statement=sqlst, user=user)

    return render_template('profile.html', user=user, email=email['EMAIL'])
```

```
#change password
@app.route(CHANGE_PASSWORD_URL, methods=['GET',
'POST']) def changepwd():     if not session.get('user'):
    return redirect(LOG_IN_PAGE_URL)

  msg = ''     user = ''     email = ''     if
request.method == 'POST':         user
= session.get('user')         oldpass =
request.form['oldpass']
    newpass = request.form['newpass']

    sqlst = 'SELECT password from user where username = ?'
    dbpass = execute_sql(statement = sqlst , username = user)['PASSWORD']
sqlst = 'SELECT email from user where username = ?'
    email = execute_sql(statement = sqlst ,username = user)['EMAIL']

    if dbpass == oldpass:
      sqlst = 'UPDATE user SET password = ? where username = ?'
execute_sql(statement = sqlst , password = newpass , username = user)
msg = 'Updated Successfully!'         else:
      msg = 'Old Password Incorrect!'

    return render_template('profile.html', user=user, email=email, msg=msg)

  return render_template('passwordChange.html')


# Logout user
@app.route('/logout') def
logout():
  session['user'] = ''
  return redirect(LOG_IN_PAGE_URL)

# Delete user account
@app.route('/delete') def
delete():     if not
session.get('user'):
    return redirect(LOG_IN_PAGE_URL)

  user = session['user']
```

```
   delete_query = "DELETE FROM user WHERE username=?"
execute_sql(statement=delete_query, user=user)

   session.clear()
   return redirect(SIGN_UP_PAGE_URL)

# Run the application if
__name__ == '__main__':
   app.run(debug=True)
```

Register.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>Sign Up</title>
       <link rel="preconnect" href="https://fonts.googleapis.com">
       <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
      <link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100;0,200;0,300;0,400;0
,500;0,600;0,700;0,800;1,400;1,600&display=swap" rel="stylesheet">
     <link
       rel="stylesheet"
       href="stylesheet.css"
     />
    </head>
    <body>
    <div class="main">

    <div id="continer">
     <a href="/home.html"><span id="logao"><p id="logo-i">BMSZ</p></span></a>
    </div>
    <div id="continer">

      <div id="form">
        <h2 id="heading">REGISTER</h2>
<form method="POST" action="">
        <div> <input type="text" placeholder="Username" name="user" required/> </div>
        <div> <input type="text" placeholder="Email" name="email" required/> </div>
        <div> <input type="password" placeholder="Password" name="password" required/>
</div>
        <input type="submit" value="Sign Up" />
        </form>
        <span id="text">Already Have an account  &nbsp;<a href="/login.html"><span id="texta">
Log  In</span></a></span>
      </div>
    </div>



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

Home.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>Home</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100;0,200;0,300;0,400;0
,500;0,600;0,700;0,800;1,400;1,600&display=swap" rel="stylesheet">
   <link
     rel="stylesheet"
href="stylesheet.css"
    />
    </head>
    <body>
   <div class="main">

    <div id="continer">
        <a href="/home.html"><span id="logao"><p id="logo-i">BMSZ</p></span></a>
     </div>

     <div id="continer-text">
        <h2>Nutrition Assistant Application</h2>
        <p>In addition to providing essential nutrients and micronutrients to the organism,
food plays a crucial role in the immune and metabolic regulation of the organism. The
nutritional quality of food has always been an important indicator in the evaluation of food,
and the nutrition of food is very important, and many food-related nutrients are closely related
to our health. However, the way to understand the nutritional quality of food while ensuring
the efficiency and integrity of food has been a hot topic of research.</p>        </div>

     <div id="continer">
        <div id="continer-img">

<spanid="textbtn"><ahref="/upload_image.html">UPLOADIMAGENOW</a></span>
        </div>
     </div>

     <div id="continer">

       <div id="cointiner-inline">
        <a href="/login.html" id="list">Log In</a>
        <a      href="/signup.html"      id="list">Register</a>
</div>
      </div>
      </body>
      </html>
```

7.2 Feature 2

Upload Image

This page allows users to upload images of food in order to get the nutrition information of it.

```
     <!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>Upload Image</title>
        <link rel="preconnect" href="https://fonts.googleapis.com">
       <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100;0,200;0,300;0,400;0
,500;0,600;0,700;0,800;1,400;1,600&display=swap" rel="stylesheet">
   <link
     rel="stylesheet"
     href="stylesheet.css"
   />

     <script>
   function readText (form) {

     food =form.inputbox.value;
     if (food === "apple" ){

       document.getElementById('apple').style.cssText = 'visibility: visible';
document.getElementById('banana').style.cssText = 'visibility: hidden';
document.getElementById('orange').style.cssText = 'visibility: hidden';
document.getElementById('potato').style.cssText = 'visibility: hidden';
document.getElementById('grape').style.cssText = 'visibility: hidden';


     }else if (food === "banana"){

       document.getElementById('apple').style.cssText = 'visibility: hidden';
document.getElementById('banana').style.cssText = 'visibility: visible';
document.getElementById('orange').style.cssText = 'visibility: hidden';
document.getElementById('potato').style.cssText = 'visibility: hidden';
document.getElementById('grape').style.cssText = 'visibility: hidden';

     }else if (food === "orange"){

       document.getElementById('apple').style.cssText = 'visibility: hidden';
document.getElementById('banana').style.cssText = 'visibility: hidden';
document.getElementById('orange').style.cssText = 'visibility: visible';
document.getElementById('potato').style.cssText = 'visibility: hidden';
document.getElementById('grape').style.cssText = 'visibility: hidden';

     }else if (food === "potato"){

       document.getElementById('apple').style.cssText = 'visibility: hidden';
document.getElementById('banana').style.cssText = 'visibility: hidden';
document.getElementById('orange').style.cssText = 'visibility: hidden';
document.getElementById('potato').style.cssText = 'visibility: visible';
document.getElementById('grape').style.cssText = 'visibility: hidden';

     }else if (food === "grape"){

       document.getElementById('apple').style.cssText = 'visibility: hidden';
document.getElementById('banana').style.cssText = 'visibility: hidden';
document.getElementById('orange').style.cssText = 'visibility: hidden';
document.getElementById('potato').style.cssText = 'visibility: hidden';
document.getElementById('grape').style.cssText = 'visibility: visible';

     }

  }
</script>


  </head>
  <body>
    <div class="main">

      <div id="continer">
        <a href="/home.html"><span id="logao"><p id="logo-i">BMSZ</p></span></a>        </div>


      <div id="continer">

          <div id="form-search">
              <FORM NAME="myform" ACTION="" METHOD="GET">
                  <div id="cointiner-inline">
                          <INPUT TYPE="text" placeholder="Enter Product" id="search-form"
NAME="inputbox" VALUE="">
```

```
            <INPUT TYPE="button" class="button" NAME="button1" Value="Search"
onClick="readText(this.form)">
          </div>


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

    <div id="continer">
      <div class="hidden product" id="apple">
        <h3>APPLE</h3>
        <table>
          <tr>
           <td>Calories</td>
              <td>Water</td>
           <td>Protein</td>
          </tr>
          <tr>
            <td>52</td>
            <td>86%</td>
            <td>0.3 grams</td>
           </tr>
        </table>
      </div>

      <div class="hidden product" id="banana">
        <h3>BANANA</h3>
        <table>
          <tr>
           <td>Calories</td>
           <td>Water</td>
              <td>Protein</td>
          </tr>
          <tr>
            <td>110</td>
            <td>20%</td>
            <td>1 grams</td>
           </tr>
</table>
      </div>

      <div class="hidden product" id="orange">
        <h3>ORANGE</h3>
        <table>
          <tr>
           <td>Calories</td>
              <td>Water</td>
           <td>Protein</td>
          </tr>
          <tr>
            <td>110</td>
            <td>20%</td>
            <td>1 grams</td>
           </tr>
</table>
      </div>

      <div class="hidden product" id="potato">
        <h3>POTATO</h3>
        <table>
          <tr>
           <td>Calories</td>
           <td>Water</td>
              <td>Protein</td>
          </tr>
          <tr>
            <td>110</td>
            <td>20%</td>
            <td>1 grams</td>
           </tr>
        </table>
      </div>

       <div class="hidden product" id="grape">
        <h3>GRAPE</h3>
        <table>
```

```
          <tr>
           <td>Calories</td>
           <td>Water</td>
           <td>Protein</td>
          </tr>
          <tr>
            <td>110</td>
            <td>20%</td>
            <td>1 grams</td>
           </tr>
        </table>
      </div>


    </div>


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

7.1 Feature 3

View History of Items

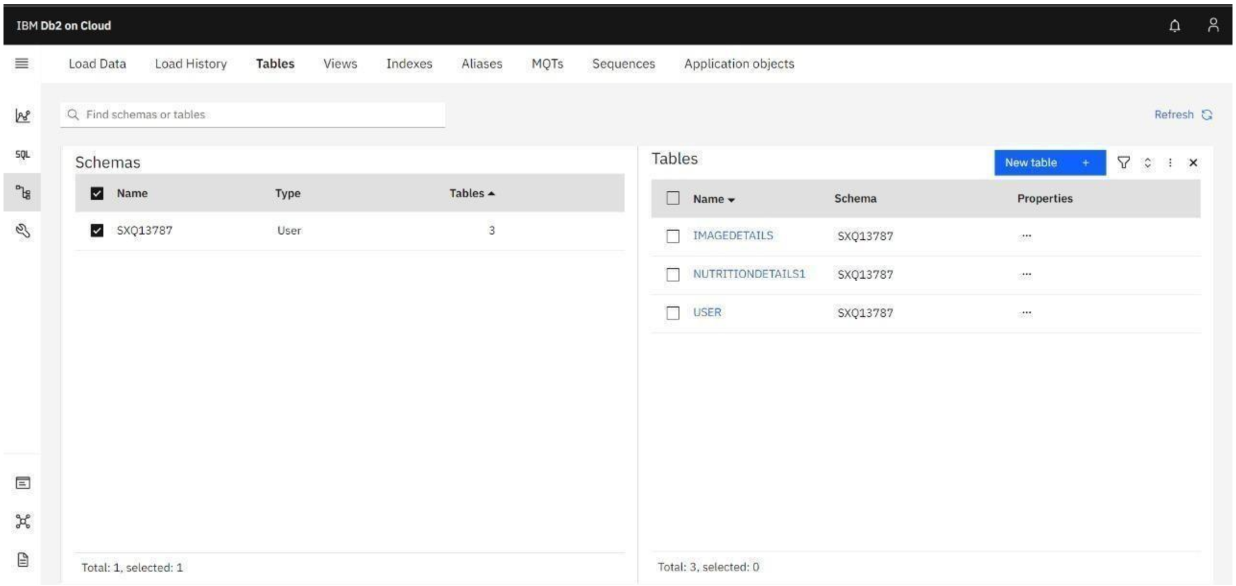
The users can view the nutritional information of all the past foods they have uploaded pictures of

FOODINFO.HTML

```
{% extends "base.html" %}

{% block head %}
      <title>Nutrify</title>
   <link rel="stylesheet" type="text/css" href="{{url_for('static',
   filename='css/indexstyle.css')}}">
{% endblock %}

{% block body %}
<br>
<br>
<br>

<div class="foodinfomain">
```

```
 div class="foodinfo card">
   <div class="row">
       <div class="col-lg-6">
          <img class="fo" src="{{files[8]}}" alt="">

       </div>
<div class="col-lg-6">

   <p>Food item: {{files[1]}}</p>
   <hr>
   <p>Calories: {{files[2]}} Kcal</p>
   <hr>
   <p>Carbs: {{files[3]}} Gm</p>
   <hr>
   <p>Fat: {{files[4]}} Gm</p>
   <hr>
   <p>Protein: {{files[5]}} Gm</p>

</div>
   </div>

</div>
</div>

{% endblock %}
```

7.5 Database Schemas



Schemas

☑ Name	Type	Tables ▲
☑ SXQ13787	User	3

Total: 1, selected: 1

Tables

□ Name ▾	Schema	Properties
□ IMAGEDETAILS	SXQ13787	...
□ NUTRITIONDETAILS1	SXQ13787	...
□ USER	SXQ13787	...

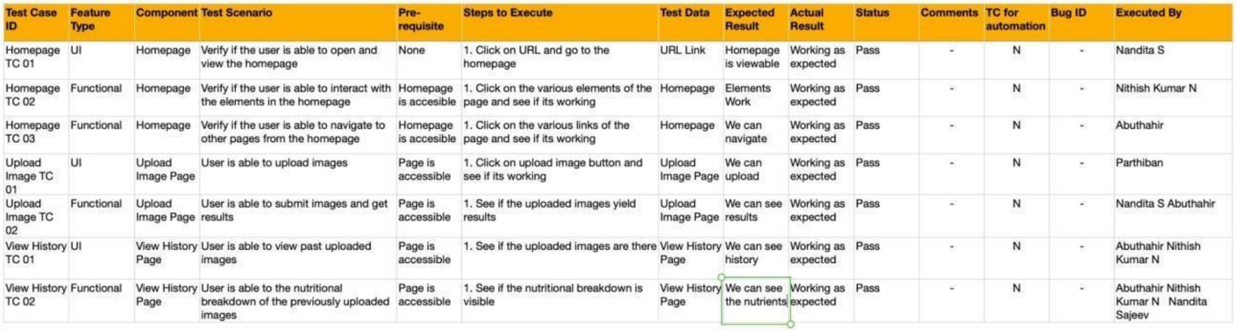
Total: 3, selected: 0

8 TESTING

8.1 Test Cases

	Test Scenarios
1	Verify if the user is able to open and view the homepage
2	Verify if the user is able to interact with the elements in the homepage
3	Verify if the user is able to navigate to the other pages of the application from the homepage
	Upload Image Page Actions
1	User is able to upload image
2	User is able to submit the image and obtain results
	View History of Items Related Actions
1	User is able to view all past uploaded images
2	User is able to see the nutritional breakdown of the previously uploaded images
	User is able to log in and sign up
1	User is able to create an account and log in

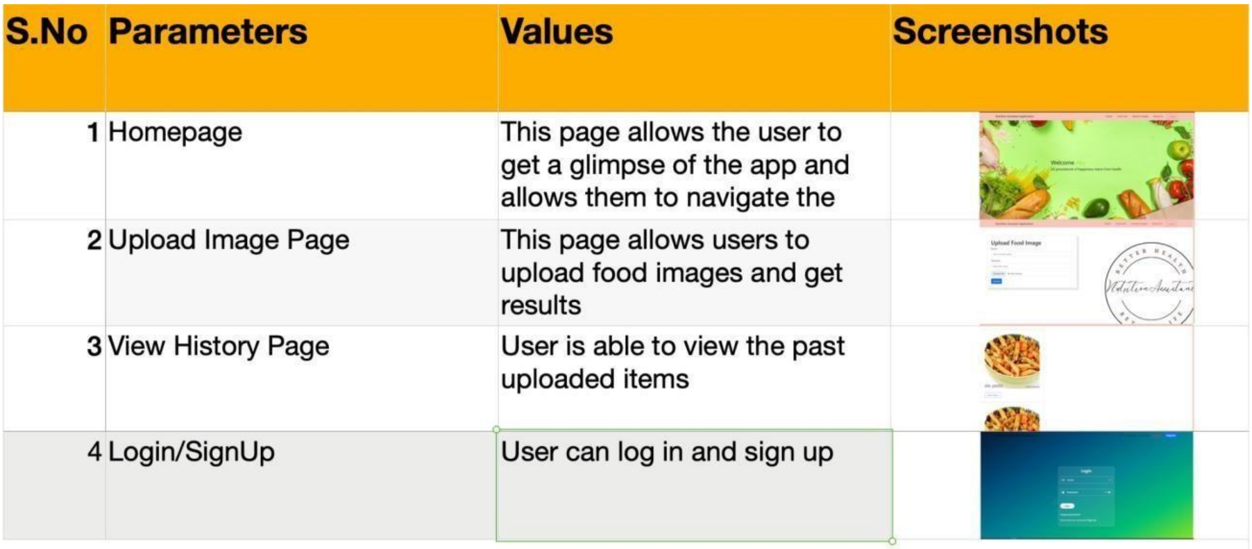
8.2 User Acceptance Testing



Test Case ID	Feature Type	Component	Test Scenario	Pre-requisite	Steps to Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for automation	Bug ID	Executed By
Homepage TC 01	UI	Homepage	Verify if the user is able to open and view the homepage	None	1. Click on URL and go to the homepage	URL Link	Homepage is viewable	Working as expected	Pass	-	N	-	Nandita S
Homepage TC 02	Functional	Homepage	Verify if the user is able to interact with the elements in the homepage	Homepage is accesible	1. Click on the various elements of the page and see if its working	Homepage	Elements Work	Working as expected	Pass	-	N	-	Nithish Kumar N
Homepage TC 03	Functional	Homepage	Verify if the user is able to navigate to other pages from the homepage	Homepage is accesible	1. Click on the various links of the page and see if its working	Homepage	We can navigate	Working as expected	Pass	-	N	-	Abuthahir
Upload Image TC 01	UI	Upload Image Page	User is able to upload images	Page is accessible	1. Click on upload image button and see if its working	Upload Image Page	We can upload	Working as expected	Pass	-	N	-	Parthiban
Upload Image TC 02	Functional	Upload Image Page	User is able to submit images and get results	Page is accessible	1. See if the uploaded images yield results	Upload Image Page	We can see results	Working as expected	Pass	-	N	-	Nandita S Abuthahir
View History TC 01	UI	View History Page	User is able to view past uploaded images	Page is accessible	1. See if the uploaded images are there	View History Page	We can see history	Working as expected	Pass	-	N	-	Abuthahir Nithish Kumar N
View History TC 02	Functional	View History Page	User is able to the nutritional breakdown of the previously uploaded images	Page is accessible	1. See if the nutritional breakdown is visible	View History Page	We can see the nutrients	Working as expected	Pass	-	N	-	Abuthahir Nithish Kumar N Nandita Sajeev

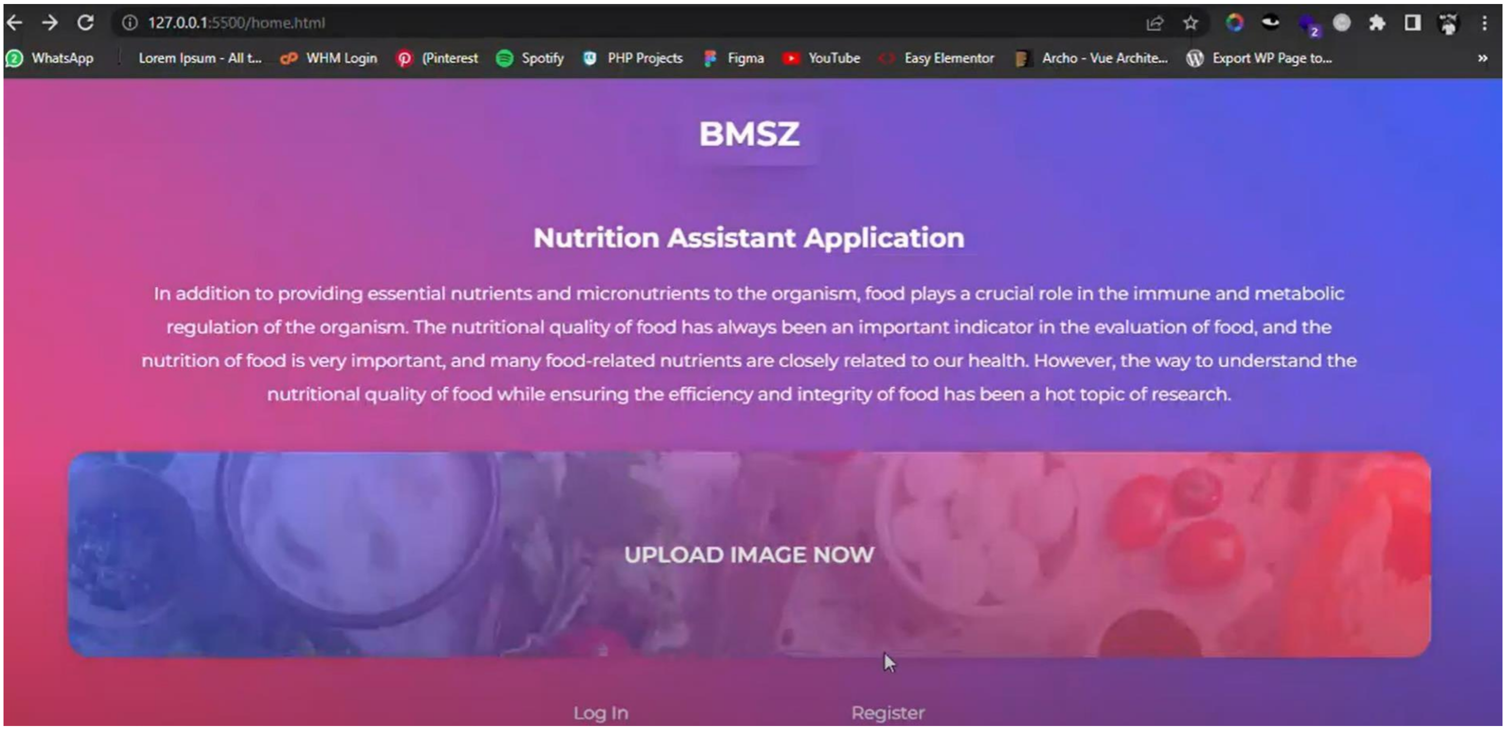
3.3.2 RESULTS

9.1 Performance Metrics



S.No	Parameters	Values	Screenshots
1	Homepage	This page allows the user to get a glimpse of the app and allows them to navigate the	
2	Upload Image Page	This page allows users to upload food images and get results	
3	View History Page	User is able to view the past uploaded items	
4	Login/SignUp	User can log in and sign up	

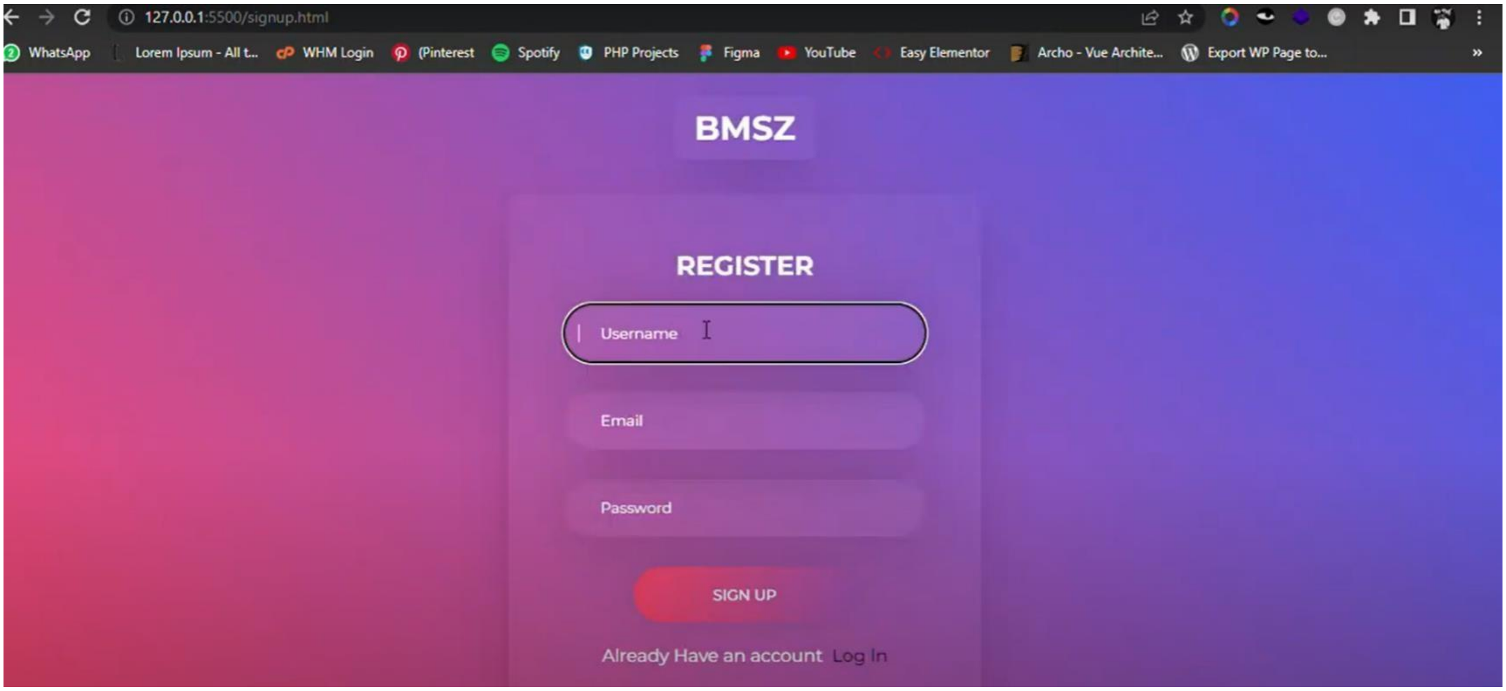
9.2 Screenshots of UI

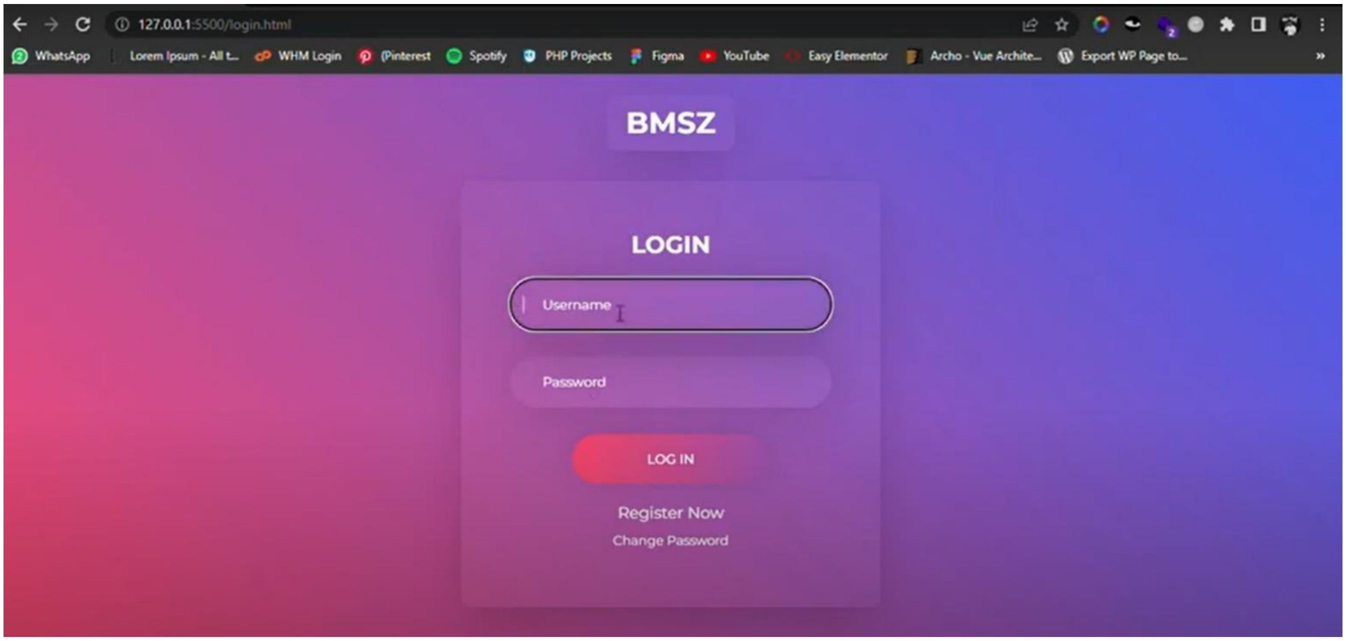


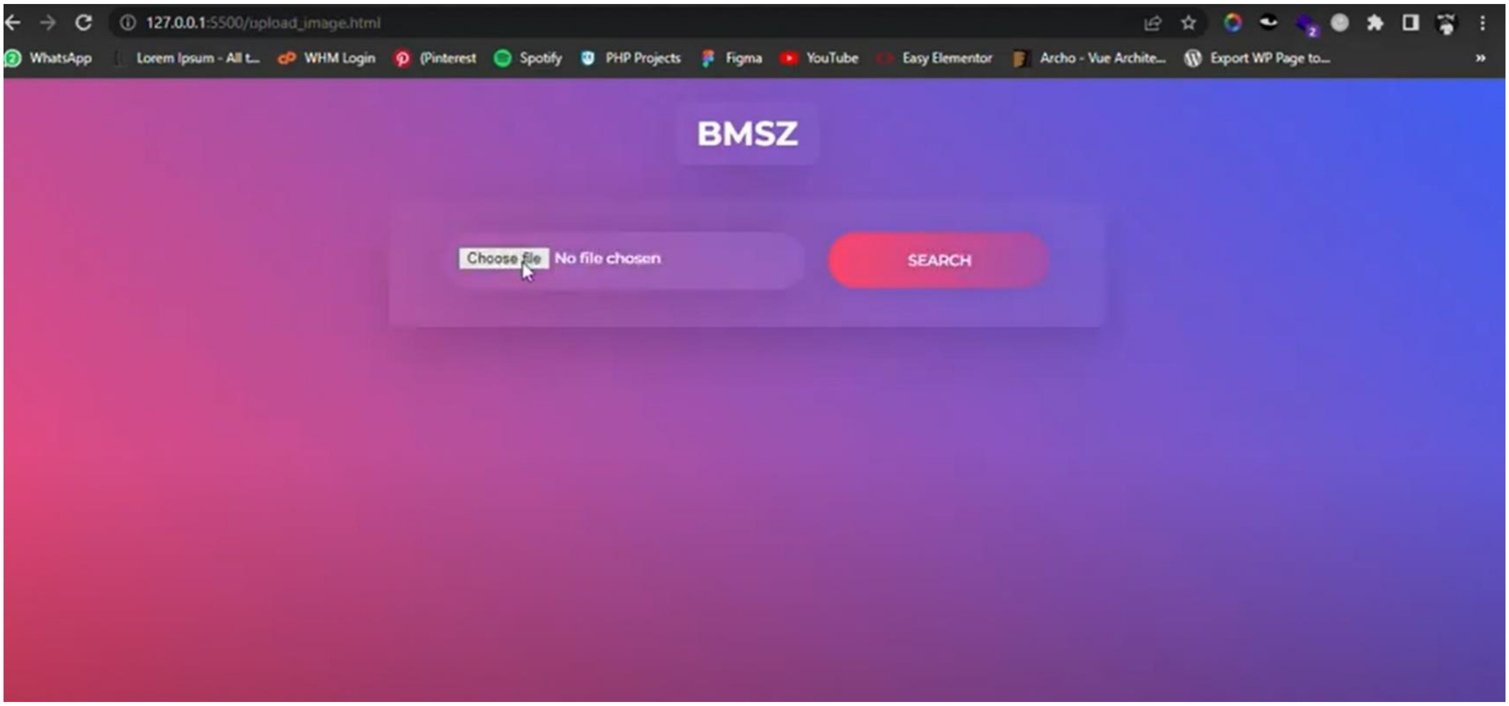
BMSZ

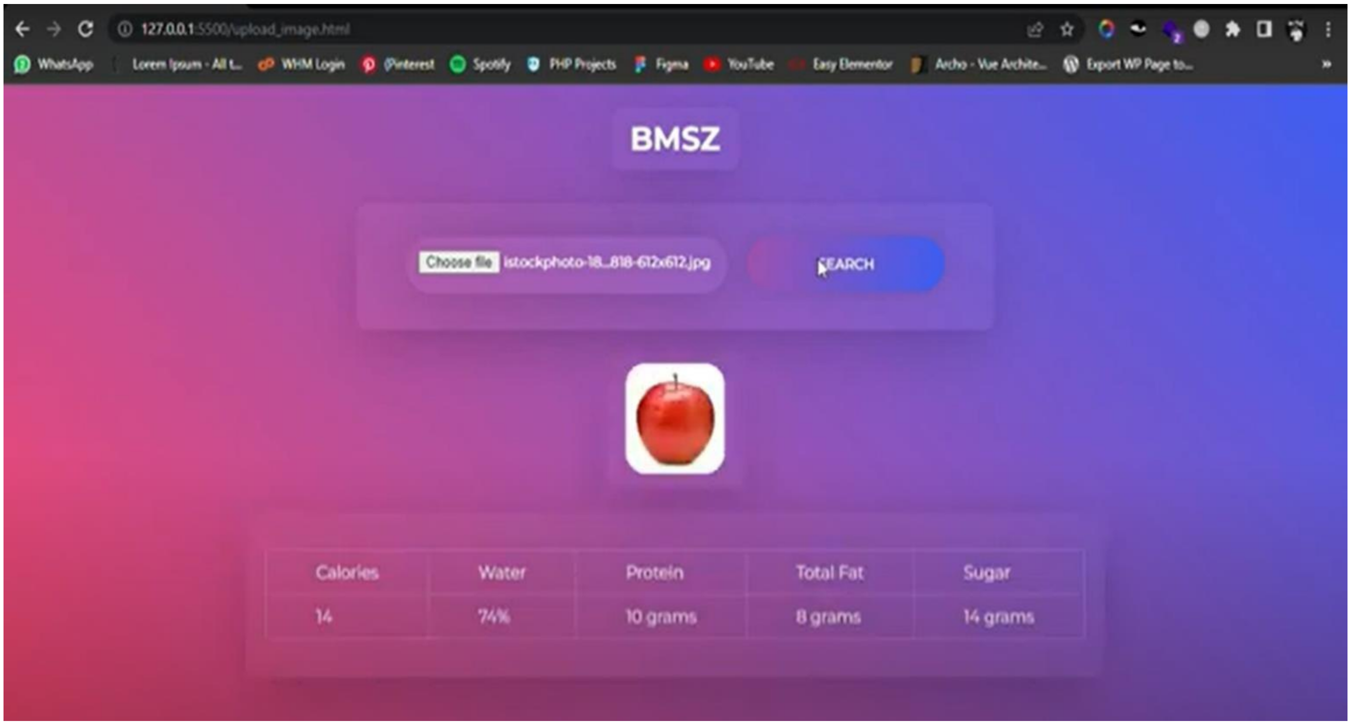
Nutrition Assistant Application

In addition to providing essential nutrients and micronutrients to the organism, food plays a crucial role in the immune and metabolic regulation of the organism. The nutritional quality of food has always been an important indicator in the evaluation of food, and the nutrition of food is very important, and many food-related nutrients are closely related to our health. However, the way to understand the nutritional quality of food while ensuring the efficiency and integrity of food has been a hot topic of research.


127.0.0.1:5500/signup.html
WhatsApp
Lorem Ipsum - All t...
WHM Login
(Pinterest
Spotify
PHP Projects
Figma
YouTube
Easy Elementor
Archo - Vue Archite...
Export WP Page to...
BMSZ
REGISTER
Username
Email
Password
SIGN UP
Already Have an account Log In


127.0.0.1:5500/login.html
WhatsApp
Lorem Ipsum - All t...
WHM Login
(Pinterest
Spotify
PHP Projects
Figma
YouTube
Easy Elementor
Archo - Vue Archite...
Export WP Page to...
BMSZ
LOGIN
Username
Password
LOG IN
Register Now
Change Password





Calories	Water	Protein	Total Fat	Sugar
14	74%	10 grams	8 grams	14 grams

10 ADVANTAGES AND DISADVANTAGES

Advantages:

1. The user is now able to track his daily calorie intake.
2. He/she can now take effective measures to maintain a healthy body weight.
3. It delivers information on the nutritional value of food and how it should be maintained on a daily basis.

Disadvantages:

1. It cannot be used without an Internet Connection.
2. Usage of 3rd party API may cause a time delay.

11. CONCLUSION

The nutrition assistant application using cloud computing is able to get images from the users and analyze them and show the nutritional breakdown of the food item. It is able to do this in an efficient and cost-effective way. This application allows people to get to know the nutrients of foods at any time which makes it more convenient for the users. This can be scaled to include APIs that have a larger variety of foods to have it cater to larger audiences of different backgrounds and ethnicities.

12. FUTURE SCOPE

The application can be improved to cater to more people

1. ADDING GRAPHICAL DATA ON THE FOODS CONSUMES

Adding a pie chart or a breakdown of what nutritional components are being consumed can give more insight into the food habits of a user. This can help the user make changes and increase or decrease their consumption of a particular nutrient or food.

2. CREATING A PERSONALISED FOOD RECOMMENDATION SYSTEM

Based on the previously uploaded images we can provide recommendations for the kinds of foods to eat to have a balanced diet.

13 APPENDIX

13.1 Source Code:

Home.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>Home</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,70
0;0,800;1,400;1,600&display=swap" rel="stylesheet">
    <link
      rel="stylesheet"
      href="stylesheet.css"
    />
  </head>
  <body>
    <div class="main">

        <div id="continer">
            <a href="/home.html"><span id="logao"><p id="logo-i">BMSZ</p></span></a>
        </div>

        <div id="continer-text">
            <h2>Nutrition Assistant Application</h2>
            <p>In addition to providing essential nutrients and micronutrients  to the organism, food plays a crucial role in
the immune and metabolic regulation of the organism. The nutritional quality of food has always been an important
indicator in the evaluation of food, and the nutrition of food is very important, and many food-related nutrients are
closely related to our health. However, the way to understand the nutritional quality of food while ensuring the
efficiency and integrity of food has been a hot topic of research.</p>
        </div>

        <div id="continer">
            <div id="continer-img">
                <span id="text-btn"><a href="/upload_image.html">UPLOAD IMAGE NOW</a></span>
            </div>
        </div>

        <div id="continer">

         <div id="cointiner-inline">
      <a href="/login.html" id="list">Log In</a>
      <a href="/signup.html" id="list">Register</a>
     </div>

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

Register.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>Sign Up</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;1,400;1,600&display=swap" rel="stylesheet">
  <link
   rel="stylesheet"
   href="stylesheet.css"
  />
 </head>
 <body>
  <div class="main">

    <div id="continer">
     <a href="/home.html"><span id="logao"><p id="logo-i">BMSZ</p></span></a>
    </div>


    <div id="continer">

      <div id="form">
       <h2 id="heading">REGISTER</h2>
<form method="POST" action="">
        <div> <input type="text" placeholder="Username" name="user" required/> </div>
        <div> <input type="text" placeholder="Email" name="email" required/> </div>
        <div> <input type="password" placeholder="Password" name="password" required/> </div>
<input type="submit" value="Sign Up" />
        </form>
        <span id="text">Already Have an account  &nbsp;<a href="/login.html"><span id="text-a"> Log
In</span></a></span>
      </div>
    </div>




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

Login.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>Log In</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,70
0;0,800;1,400;1,600&display=swap" rel="stylesheet">
  <link
   rel="stylesheet"
href="stylesheet.css"
  />
 </head>
 <body>
  <div class="main">

    <div id="continer">
     <a href="/home.html"><span id="logao"><p id="logo-i">BMSZ</p></span></a>
    </div>


    <div id="continer">

      <div id="form">
       <h2 id="heading">LOGIN</h2>
        <form method="POST" action="">
          <div> <input type="text" placeholder="Username" name="user" required /></div>
          <div> <input type="password" placeholder="Password" name="password" required/></div>
<input type="submit" value="Log In" />
        </form>
```

```
                <span id="text"><a href="/signup.html">Register Now</a></span>
                <span id="text" class="s-text"><a href="/change_password.html">Change Password</a></span>
            </div>
        </div>



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

Change_password.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>Sign Up</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;1,400;1,600&display=swap" rel="stylesheet">
    <link
      rel="stylesheet"
      href="stylesheet.css"
    />
  </head>
  <body>
    <div class="main">

        <div id="continer">
          <a href="/home.html"><span id="logao"><p id="logo-i">BMSZ</p></span></a>
        </div>


        <div id="continer">

            <div id="form">
              <h2 id="heading">Change Password</h2>
              <form method="POST" action="">
                <input type="text" placeholder="Old Password" name="old_password" required />
                <input type="password" placeholder="New Password" name="new_password" required />
<input type="submit" value="Change" />
              </form>
            <span id="text"><a href="/login.html">Log In</a></span>
          </div>
        </div>




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

Image.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>Upload Image</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,70
0;0,800;1,400;1,600&display=swap" rel="stylesheet">
    <link
      rel="stylesheet"
      href="stylesheet.css"
    />

<script>
    function readText (form) {

      food =form.inputbox.value;
      if (food === "apple" ){

        document.getElementById('apple').style.cssText = 'visibility: visible';
document.getElementById('banana').style.cssText = 'visibility: hidden';
document.getElementById('orange').style.cssText = 'visibility: hidden';
document.getElementById('potato').style.cssText = 'visibility: hidden';
document.getElementById('grape').style.cssText = 'visibility: hidden';


      }else if (food === "banana"){

        document.getElementById('apple').style.cssText = 'visibility: hidden';
document.getElementById('banana').style.cssText = 'visibility: visible';
document.getElementById('orange').style.cssText = 'visibility: hidden';
document.getElementById('potato').style.cssText = 'visibility: hidden';
document.getElementById('grape').style.cssText = 'visibility: hidden';

      }else if (food === "orange"){

        document.getElementById('apple').style.cssText = 'visibility: hidden';
document.getElementById('banana').style.cssText = 'visibility: hidden';
document.getElementById('orange').style.cssText = 'visibility: visible';
document.getElementById('potato').style.cssText = 'visibility: hidden';
document.getElementById('grape').style.cssText = 'visibility: hidden';       }else if (food === "potato"){

        document.getElementById('apple').style.cssText = 'visibility: hidden';
document.getElementById('banana').style.cssText = 'visibility: hidden';
document.getElementById('orange').style.cssText = 'visibility: hidden';
document.getElementById('potato').style.cssText = 'visibility: visible';
document.getElementById('grape').style.cssText = 'visibility: hidden';

      }else if (food === "grape"){

        document.getElementById('apple').style.cssText = 'visibility: hidden';
document.getElementById('banana').style.cssText = 'visibility: hidden';
document.getElementById('orange').style.cssText = 'visibility: hidden';
document.getElementById('potato').style.cssText = 'visibility: hidden';
document.getElementById('grape').style.cssText = 'visibility: visible';

      }

  }
</script>


  </head>
  <body>
    <div class="main">

        <div id="continer">
          <a href="/home.html"><span id="logao"><p id="logo-i">BMSZ</p></span></a>
        </div>


        <div id="continer">

            <div id="form-search">
                <FORM NAME="myform" ACTION="" METHOD="GET">
                    <div id="cointiner-inline">
                        <INPUT TYPE="text" placeholder="Enter Product" id="search-form" NAME="inputbox" VALUE="">
            <INPUT TYPE="button" class="button" NAME="button1" Value="Search"
onClick="readText(this.form)">
          </div>


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

    <div id="continer">
      <div class="hidden product" id="apple">
        <h3>APPLE</h3>
        <table>
          <tr>
           <td>Calories</td>
           <td>Water</td>
           <td>Protein</td>
          </tr>
          <tr>
            <td>52</td>
            <td>86%</td>
            <td>0.3 grams</td>
           </tr>
</table>
      </div>

      <div class="hidden product" id="banana">
        <h3>BANANA</h3>
        <table>
          <tr>
           <td>Calories</td>
           <td>Water</td>
           <td>Protein</td>
          </tr>
          <tr>
            <td>110</td>
            <td>20%</td>
            <td>1 grams</td>
           </tr>
</table>
      </div>

      <div class="hidden product" id="orange">
        <h3>ORANGE</h3>
        <table>
          <tr>
```

```
           <td>Calories</td>
           <td>Water</td>
           <td>Protein</td>
          </tr>
          <tr>
            <td>110</td>
            <td>20%</td>
            <td>1 grams</td>
           </tr>
</table>
      </div>

      <div class="hidden product" id="potato">        <h3>POTATO</h3>
        <table>
          <tr>
           <td>Calories</td>
           <td>Water</td>
           <td>Protein</td>
          </tr>
          <tr>
            <td>110</td>
            <td>20%</td>
            <td>1 grams</td>
           </tr>
</table>
      </div>

      <div class="hidden product" id="grape">
        <h3>GRAPE</h3>
        <table>
          <tr>
           <td>Calories</td>
           <td>Water</td>
           <td>Protein</td>
          </tr>
          <tr>
            <td>110</td>
            <td>20%</td>
            <td>1 grams</td>
           </tr>
</table>
      </div>


    </div>
```

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

Stylesheet.css:

```
       body{
margin: 0%;
   font-family: 'Montserrat', sans-serif;
}

.main {     height:
100vh;
   background: linear-gradient(45deg, #FC466B, #3F5EFB); }

#continer{
   display: flex;
  justify-content: center;

}

#logo-i{     background-color:
#ffffff08;     backdrop-filter:
blur(20px);
   color: #ffffff;     font-
weight: 700;     font-size:
30px;     margin: 0 auto;
padding: 10px 20px;
margin-top: 20px;
border-radius: 10px;
   box-shadow: rgba(0, 0, 0, 0.2) 0px 18px 50px -10px;
}

#form{
   background-color: #ffffff08;
backdrop-filter: blur(20px);
padding: 50px;     margin-top:
30px;     border-radius: 10px;
   box-shadow: rgba(0, 0, 0, 0.2) 0px 18px 50px -10px;
}

#form-search{     background-
color: #ffffff08;     backdrop-
filter: blur(20px);     padding:
30px 50px 15px 50px;     margin-
top: 30px;     border-radius:
10px;
   box-shadow: rgba(0, 0, 0, 0.2) 0px 18px 50px -10px;
}

input {
    color: #ffffff;
    background: transparent;
width: 300px;     height:
25px;     padding: 1em;
margin-bottom: 2em;
border: none;     border-
radius: 5000px;
backdrop-filter: blur(5px);
box-shadow: 4px 4px 60px
rgba(0,0,0,0.2);     font-
family: Montserrat, sans-
serif;     font-weight: 500;
    transition: all 0.2s ease-in-out;
}

#search-form{     margin-
right: 20px;
    margin-bottom: 0px;
}
.image-input{
    color: #ffffff;
     display: block;
}
::placeholder {
padding: 20px;
    font-family: Montserrat, sans-serif;     font-
weight: 400;
    color: #fff;
  }
  input[type=submit] {

    margin: auto;

    background: linear-gradient(95deg, #FC466B, #3F5EFB);
text-align: center;         text-transform: uppercase;
transition: 0.5s;         background-size: 200% auto;
        color: white;
        box-shadow: 0 0 20px rgba(0, 0, 0, 0.123);
border-radius: 50px;         display: block;
height: 50px;         width: 200px;
        margin-bottom: 20px

}
```

```
input[type=submit]:hover {
       background-position: right center; /* change the direction of the change here */
color: #fff;
       text-decoration: none;
       ;
}

#heading{     margin:
auto;     text-align:
center;     margin-
bottom: 20px;
   color: #ffffff;
}
#text{     display: flex;
justify-content: center;
   color: #ffffff;
   margin: 10px;
}
a{
   color: #ffffff;     text-
decoration: none;
   text-align: center;
}
#text-a{
   color: #000000;
}
.s-text{
   font-size: 13px;
}

#continer-text{
width: 1100px;
margin: auto;
   color: #ffffff;     margin-
top: 50px;     text-align:
center;
   line-height: 30px;
}

#continer-img{
   background:linear-gradient(90deg, #3f5efbb9, #fc466ad2), url(/nutritionphilosophy_detailfeature.png);
background-size:cover;     margin: auto;     text-align: center;     padding: 80px 500px;     margin-top: 20px;
margin-bottom: 20px;     border-radius: 20px;
   box-shadow: 0 0 20px rgba(0, 0, 0, 0.123);
   transition-duration: 0.5s;     font-
size: 20px;
   font-weight: 600;
```

```
}

#continer-img:hover{
   background-position: right center;
}

#cointiner-inline{
   display: flex;
}

#list{
   flex: 1;     text-align:
center;     margin:
20px 100px;
}

.product{     background-
color: #ffffff08;     backdrop-
filter: blur(20px);
   color: #ffffff;
margin: 0 auto;
padding: 10px 20px;
margin-top: 80px;
border-radius: 10px;
   box-shadow: rgba(0, 0, 0, 0.2) 0px 18px 50px -10px;
   text-align: center;
   position: absolute;
}

table {
   border-collapse: collapse;
width: 100%;
   margin-bottom: 25px;
  }

td{
   border: 1px solid #ffffff38;     text-
align: left;
   padding: 10px 50px;
  }

  .hidden { visibility: hidden; }

  .button{
margin: auto;
```

```
    background: linear-gradient(95deg, #FC466B, #3F5EFB);
text-align: center;         text-transform: uppercase;
transition: 0.5s;
        background-size: 200% auto;
        color: white;
        box-shadow: 0 0 20px rgba(0, 0, 0, 0.123);
border-radius: 50px;        display: block;
height: 50px;        width: 200px;
        margin-bottom: 20px
  }

  .button:hover{
    background-position: right center; /* change the direction of the change here */
color: #fff;
        text-decoration: none;
        ;
  }
```

13.1 GitHub and Project Demo Link

Github: IBM-EPBL/IBM-Project-21085-1659772002: Nutrition assistant Application (github.com)

Demo video:

https://drive.google.com/file/d/1HptDD7ufgZ7tHxHPO6u4X8GMWfdOpsmH/view?usp=sharing