NUTRITION ASSISTANT APPLICATION

A PROJECT REPORT

Submitted by

SUMITH SJ (95001910644)

PITCHUMANI A (950019106302)

MOHAMED HAMTHAAN S (950019106703)

MOHAMED ABDULLAH ASHIQUE M (950019106704)

TEAM ID: PNT2022TMID49642

In partial fulfilment for the award of the degree of

DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

ANNA UNIVERSITY REGIONAL CAMPUS

TIRUNELVELI

S.NO **TABLE OF CONTENT**

1.INTRODUCTION

1.1 PROJECT OVERVIEW:

Good health can be achieved by maintaining good behaviours such as a good night sleep, enough exercise and good nutrition. However, the competitive environment nowadays prevents such good behaviours.

In today world due to the ignorance of healthy food habits obesity rates are increasing at an alarming speed, and this is reflective of the risks to people's health. People need to control their daily calorie intake by eating healthier foods, which is the most basic method to avoid obesity. So the people want to know the nutritional value of the food. However, although food packaging comes with nutrition (and calorie) labels, it's still not very convenient for people.

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.

2.**LITERATURE SURVEY**

2.1. EXISTING PROBLEM

SL. NO	**TITLE**	**YEAR OF THE PAPER**	**AUTHOR**	**METHODOLOGY USED**	**MERITS**	**DEMERITS**
1	The use of Smartphone health apps and other mobile health (m health) technologies	2021	J., Fliers, J., Bouman, A., Hanning, R., Allman-Farinelli, M.	To record nutrition information	The analyzed resultsin a simple and easy to understand format.	Only uploaded image can be analyzed
2	Innovative approaches to estimate individual usual dietary intake inlarge-scale epidemiological studies.	2020	Conrad J, Nöthlings U.	Innovative mobile phone–based tools may be superior	To conventionaltools in large- scale setups	Only supporti vein mobile based
3	An Application of the Principles of Minimalism to theDesign of Human Computer Interfaces	2020	J.T. Hackos	The concept of the user interface is based on the Minimalism	The interface is focusing on a simple andclean design	Use only fewer items on the screen

4	An Algorithm to Generate a Diet Plan to Meet Specific Nutritional Requirements	2019	Elsweiler, D., Harvey, M., Ludwig, B.	Computational Nutrition Algorithm	Healthy lifestyle can prevent obesity	Only prevent obesity

2.2. **REFERENCES:**

1. https://onlinelibrary.wiley.com/doi/abs/10.1111/jhn.12446
2. https://journals.lww.com/co-clinicalnutrition/Abstract/2018/09000/New_approaches_in_assessing_food_intake_in.6.aspx
3. https://ieeexplore.ieee.org/abstract/document/6281913
4. https://dl.acm.org/doi/abs/10.1145/2792838.2799665

2.3 .**PROBLEM STATEMENT DEFINITION:**

PROBLEM

People are suffering with obesity and many other various Health problems such as diabetes, thyroid etc .which may be due to deficiency innutrients.

REASONS FOR PROBLEM

Nowadays junk food has become an inevitable part of people's lives. Even though it doesn'tcause much effect with minimalintake, an excessive consumption of the same might lead to various health disorders. Exercise would help to keep these ill effects at bay, but the work culture these days has limited such habits too.

ISSUES

A lot of people suffer with diabetes, thyroid, etc. These people may have a chance of living their life without being sick by following a healthy food regime. There are over thousands of people, mainly kids, suffering due to obesity these days. In today's world exercise has become so sporadic and not everyone is keen nor have the time to pursue the same. So the only way for them to stay healthyis by having a healthydiet. When all men and womenare hale, hearty and healthyour society would achieve great heights and success. An ecosystem filled with sick and unhealthypeople is bound to have a downfall.So it is very much important to ensurethe wellness of our species.

IMPORTANCE OF FIXING THE PROBLEM

A hale, hearty and healthy society is never meant to face a downfall. So it is crucial to make sure all human beings are in good shape. On using our app, the customer would be facilitated to have a note of their calorie consumption and hence, do not go overboard with the junk. When a lot of people get to know about the true intentions of the app, they recommend the same to their peers and family, thus, resulting in an expansion of customerbase.

TECHNOLOGY IN NUTRITION ASSESSMENT

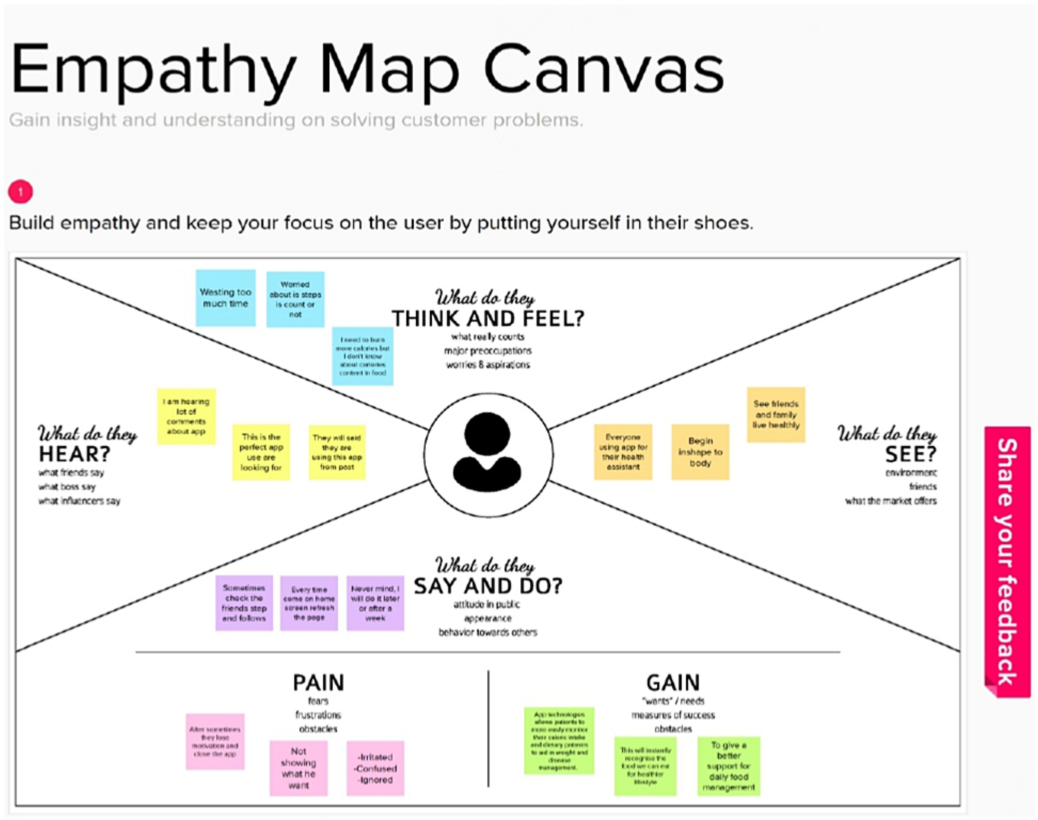
In today's life, people have started emphasizing a healthier lifestyle due to self-awareness rather than societal stereotypes. Majority of people want to start eating foods with more nutritional value but are stuck in a pit of"where to start". The solution to this problem is to build a nutrition analysis system using Artificial Intelligence and Machine Learningthat aims at providing nutritional data of any food item with a decent accuracy.

Fitness apps are blooming in today's technology market. Along with workout recommendations, these apps also help users to connect to nutritionists all around the world who work on suggesting a proper diet schedulefor the users. The hurdle these nutritionists face is that there are a huge number of food items and it is impossible for one to know the details of all such food items. This plays an important part in their job and the goal is to build an application using Artificial Intelligence and Machine Learning that could assess food items and providethe amount of nutrients it contains.

3. **IDEATION & PROPSED SOLUTION**

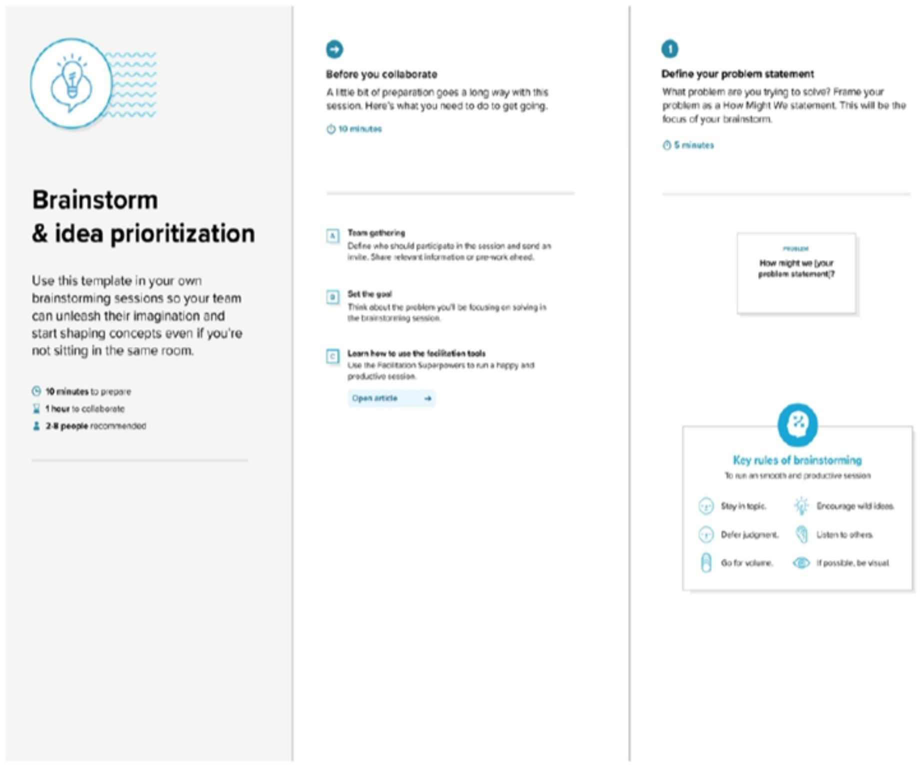
3.1EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors' and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.


Empathy Map Canvas
Gain insight and understanding on solving customer problems.
1
Build empathy and keep your focus on the user by putting yourself in their shoes.
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Westing too much time
Worried about is steps is count or not
I need to burn more calories but I don't know about calories content in food
What do they HEAR?
what friends say
what boss say
what influencers say
I am hearing lot of comments about app
This is the perfect app use are looking for
They will said they are using this app from past
What do they SEE?
environment
friends
what the market offers
Everyone using app for their health assistant
Begin inshape to body
See friends and family live healthly
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Sometimes check the friends step and follows
Every time come on home screen refresh the page
Never mind, I will do it later or after a week
PAIN
fears
frustrations
obstacles
After sometimes they lose motivation and close the app
Not showing what he want
-Irritated
-Confused
-Ignored
GAIN
"wants" / needs
measures of success
obstacles
App technologies allows patients to more easily monitor their caloric intake and dietary patterns to aid in weight and disease management.
This will instantly recognise the food we can eat for healthier lifestyle
To give a better support for daily food management
Share your feedback

3.2 BRAINSTORMING:

Step-1: Team Gathering, Collaboration and Select the Problem Statement



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

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.

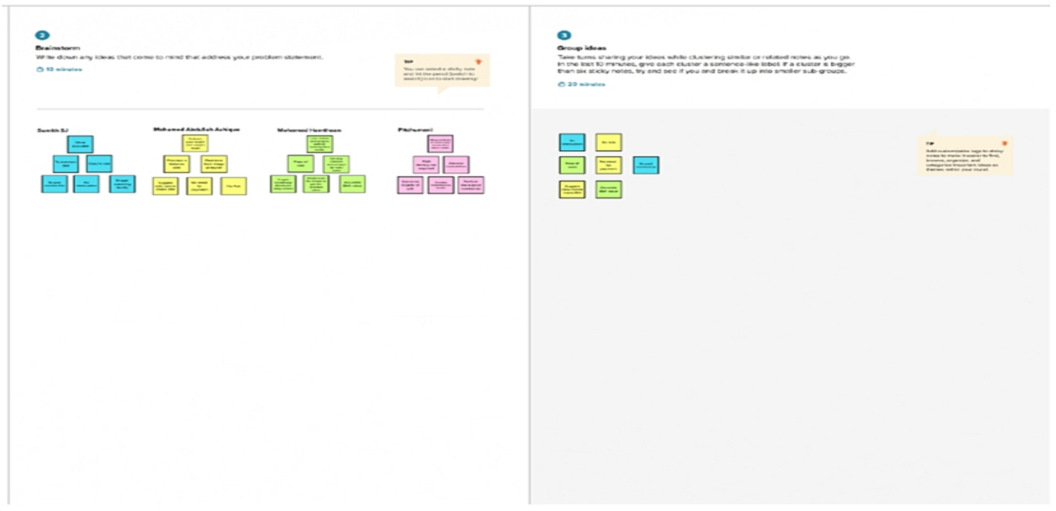
5 minutes

PROBLEM
How might we [your problem statement]?

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.

Step-2: Brainstorm, Idea Listing and Grouping



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!

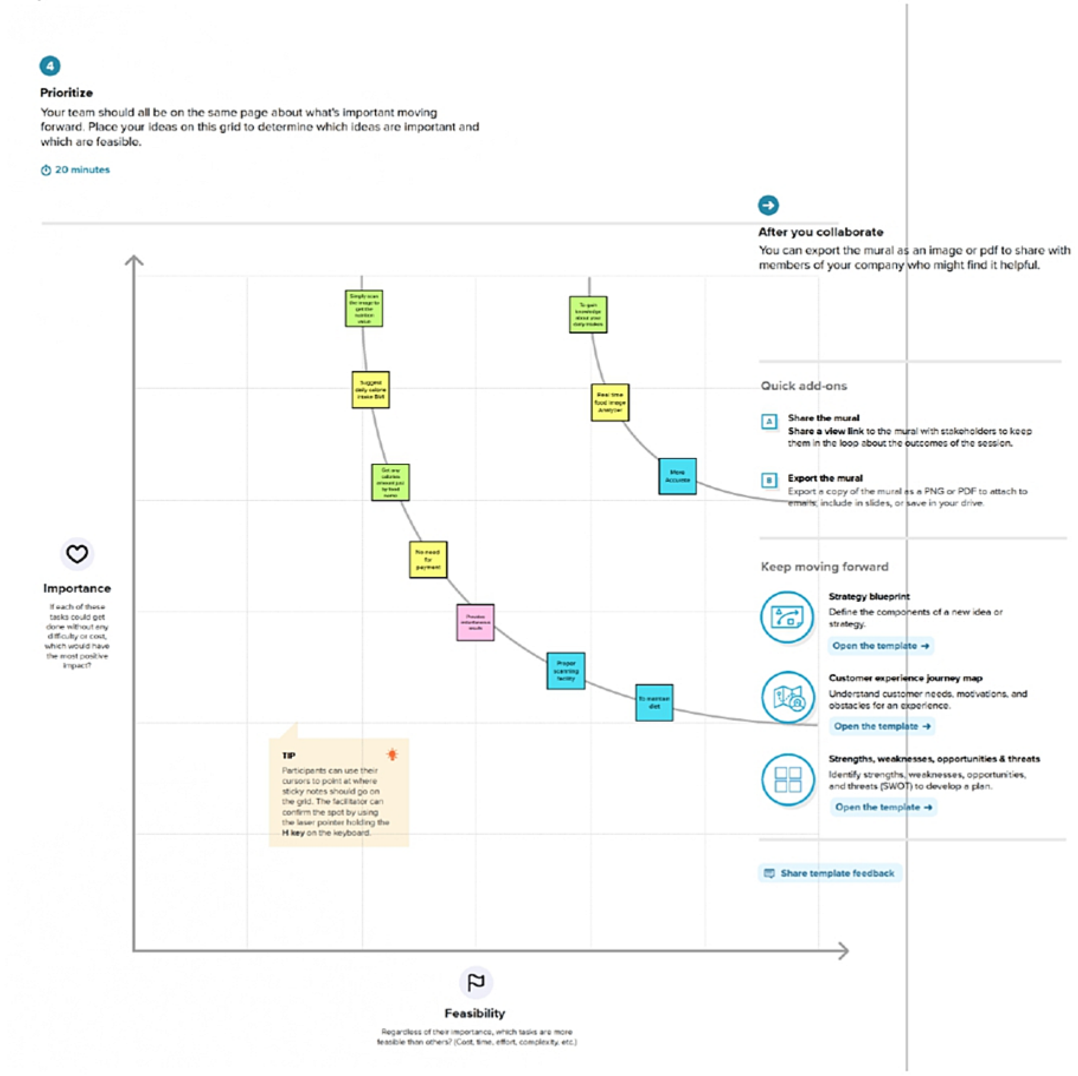
3 **Group ideas**

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.

Step-3: Idea Prioritization



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

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

Share the mural

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

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.

Customer experience journey map

Understand customer needs, motivations, and obstacles for an experience.

Strengths, weaknesses, opportunities & threats

Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.

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.

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.)

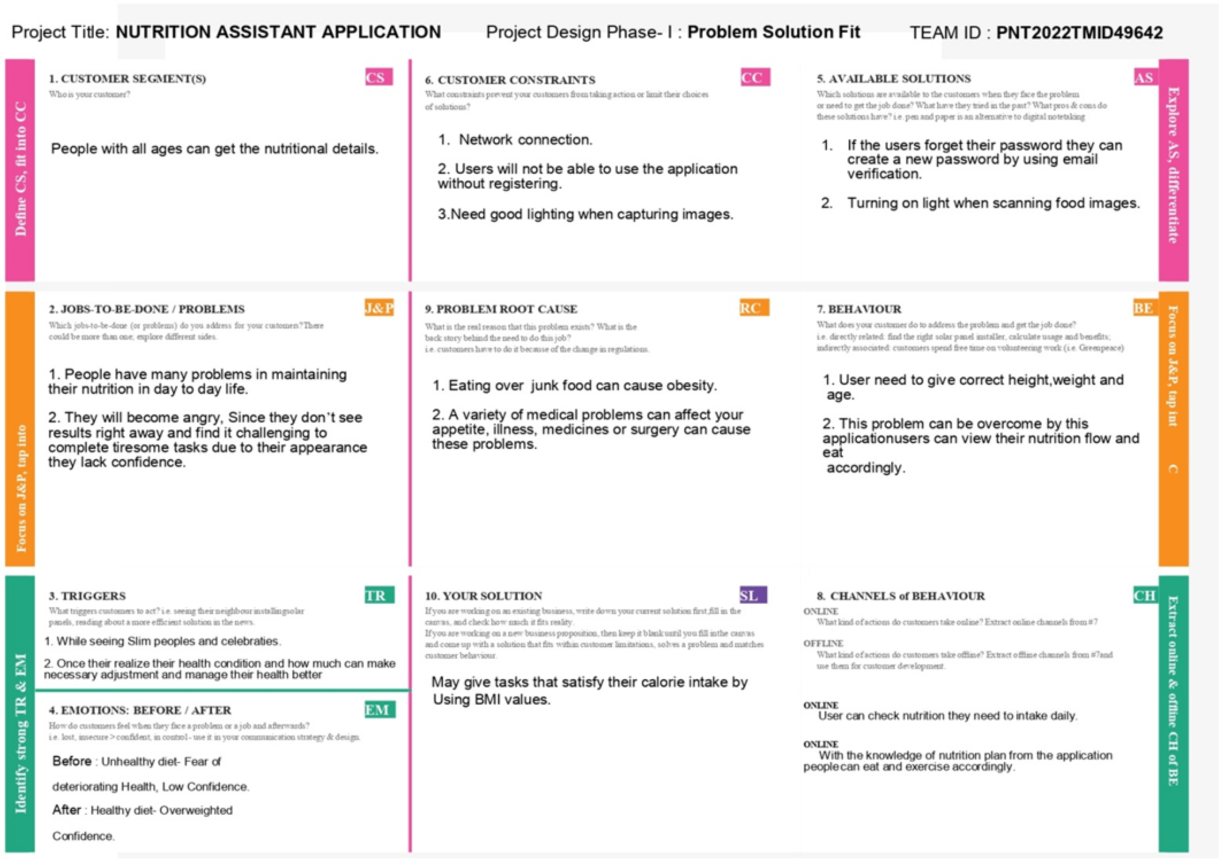
Simply scan the image to get the nutrition value
Suggest daily calorie intake BMI
Get any calories amount just by food name
No need for payment
Provides instantaneous results
Proper scanning facility
To maintain diet
To gain knowledge about your daily intakes
Real time food image Analyzer
More Accurate
Open the template →
Open the template →
Open the template →
Share template feedback

3.3. PROPSED SOLUTION

S.No.	Parameter	Description
1	Problem Statement (Problem to besolved)	People with healthy eating patterns live longer andare at lowerrisk for serioushealth problems. But in today's world due to eating of junk foods in modern lifestyle obesity rate increases rapidly .People should avoid these foods and eat healthy foods to avoidobesity and health problems.
2	Idea / Solution description	● By scanning food, we can provide approximate calorievalues. User can know theirdaily food intakecalorie. ● By usingBMI calculated valuefrom user we can suggest their daily calorie intake.
3	Novelty / Uniqueness	Byusing chat bot user can get calorie of thefood.
4	Social Impact/ Customer Satisfaction	The application whichgives awareness to the peopleabout the obesityand various health problems.

5	Business Model(Revenue Model)	1. To implement Subscription based payment for VIP users. 2. To collaborate with food delivery apps and user can order foods.
6	Scalability of the Solution	Bysuggest their dailycalorie intake theycanmaintain theirBMI

3.4 Problem Solution fit



Project Title: **NUTRITION ASSISTANT APPLICATION** Project Design Phase- I : **Problem Solution Fit** TEAM ID : **PNT2022TMID49642**

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS
Who is your customer?

People with all ages can get the nutritional details.

6. CUSTOMER CONSTRAINTS CC
What constraints prevent your customers from taking action or limit their choices of solutions?

1. Network connection.
2. Users will not be able to use the application without registering.
3.Need good lighting when capturing images.

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.
2. Turning on light when scanning food images.

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. They will become angry, Since they don't see results right away and find it challenging to complete tiresome tasks due to their appearance they lack confidence.

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. Eating over junk food can cause obesity.
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. User need to give correct height,weight and age.
2. This problem can be overcome by this applicationusers can view their nutrition flow and eat
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 installingsolar panels, reading about a more efficient solution in the news.

1. While seeing Slim peoples and celebraties.
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 : Unhealthy diet- Fear of deteriorating Health, Low Confidence.
After : Healthy diet- Overweighted Confidence.

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 inthe canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

May give tasks that satisfy their calorie intake by Using BMI values.

8. CHANNELS of BEHAVIOUR CH
ONLINE
What kind of actions do customers take online? Extract online channels from #7
OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7and use them for customer development.

ONLINE
User can check nutrition they need to intake daily.

ONLINE
With the knowledge of nutrition plan from the application peoplecan eat and exercise accordingly.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	UserRegistration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	UserConfirmation	Confirmation viaEmail Confirmation via OTP
FR-3	UserLogin	Usercan login withregistered email
FR-4	UserRequest	The User can send request to web server to know about nutrition valuesand calories
FR-5	Server Response	The serverrespond to the user with nutrition valueof thegiven image by the user
FR-6	User – Server Interaction	The user send the request with a given image then server respond with the nutrition value

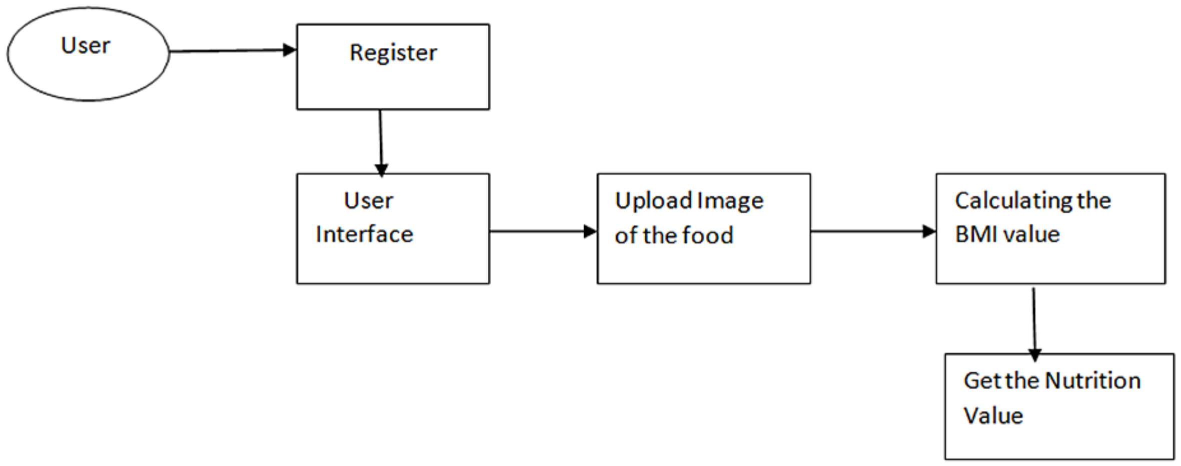
4.2.NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The User Usethis application with internet .Thisapplication is used to calculate the bmi value.
NFR-2	**Security**	Thisapplication maintains the secured protocol security system.
NFR-3	**Reliability**	This Application is so reliable because the Information we are providing are from theProfessional Nutrition Consultant.
NFR-4	**Performance**	ThePerformance of thedepends upon the network
NFR-5	**Availability**	It is Available to Everyone who have smartphones, laptops with goodinternet service.
NFR-6	**Scalability**	Itsall about the serversystem side andwe are providing a service foras short scaleof users.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enter and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture

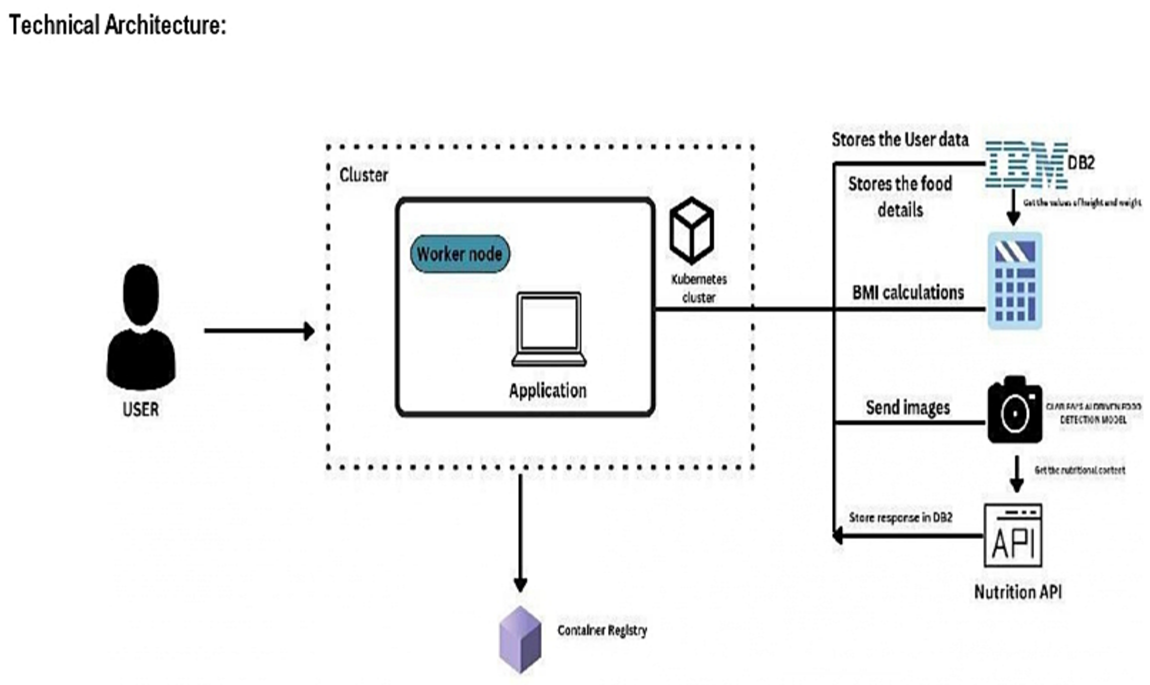


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User interacts with application usingLogin andForm.	HTML-5,CSS, Python FLASK
2.	Registration	User register in the application to verify the user.	Python FLASK, HTML-5, CSS, IBM DB2.
3.	BMI Calculation	Calculate BMI valueby using user'sheight and weight.	Python FLASK, IBM DB2.
4.	Image Analyzer	Analyze real timeimages scanned by the user.	Clarifai's AI drivenfood detection model
5.	Cloud Database	Database Service on cloud	IBM DB2
6.	Kubernetes cluster	Run containerized application	IBM kubernetes

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Docker is used foropen source framework.	Docker
2.	Scalable Architecture	It connected with scalable architecture.	IBM DB2
3.	Availability	This application is anytime accessible.	Python FLASK

4.	Performance	Record resource request and save registered information. Availability of application.	IBM DB2

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobileuser)	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 emailonce I have registered for the application	I can receive confirmationemail &click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access thedashboard with Facebook Login	Low	Sprint-2

		USN-4	As a user, I can register for the application through Gmail	I can access my dashboard	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application byentering email & password	I can access my accountby logging in into my account	High	Sprint-1
	Dashboard					
Customer (Web user)	Sign Up	USN-6	As a user ,I can sign up using my email, password	I can get access to dashboard	High	Sprint-1
	Login	USN-7	As a user, I can login to the application byentering the mailand password	I can access my dashboard by logging intomy account	High	Sprint-1
Administrator	RegisterPage	USN-8	If the user is new to the web application ,admin asks the user to sign up	With the entered details ofuser ,it will get them to dashboard	High	Sprint -1

6.1 **PROJECT PLANNING & 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 theapplication by entering my email, password, and confirmingmy password.	2	High	Pitchumani A, Sumith SJ
Sprint-2	Login	USN-2	As a user, I can log into the application by entering email& password	1	High	Mohamed Hamthaan S, Mohamed Abdullah Ashique M
Sprint-3	Mail	USN-3	As a user, I will receive confirmation email onceIhave registered forthe applicati on	2	High	Pitchumani A, SumithSJ
Sprint-4	Dashboard	USN-4	As a user, I can accessmy details, BMIvalue,cal orie count, scanning real time images, etc.,	2	High	Pitchumani A , Mohamed Hamthaan S, Mohamed Abdullah Ashique M,

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

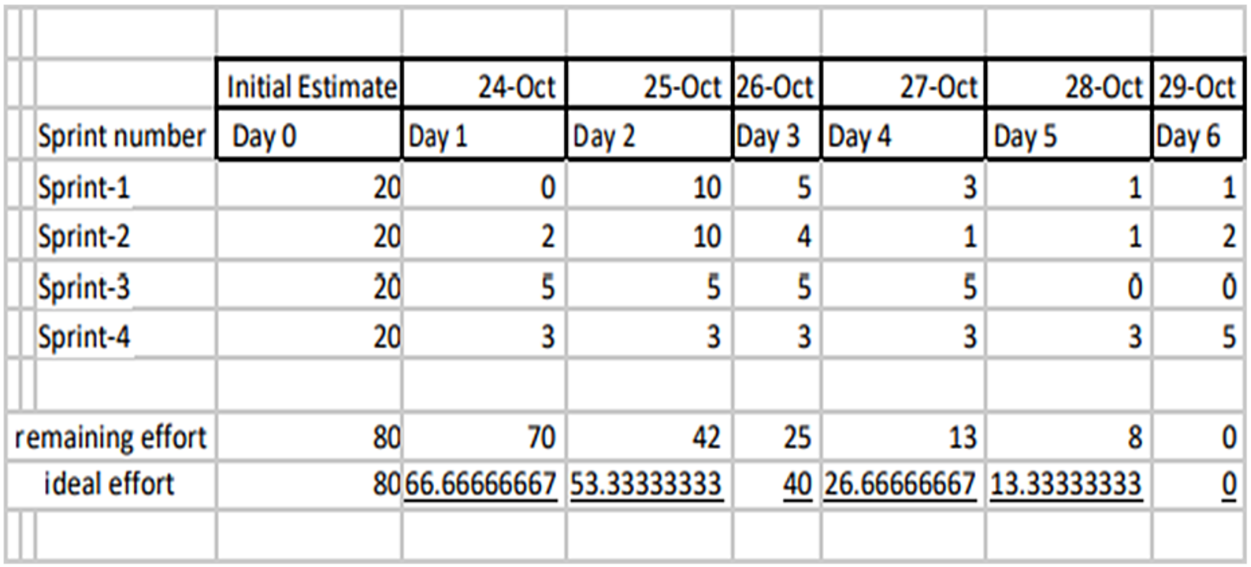
AV=sprint duration/velocity=20/10=2

AverageVelocity = Story Points per Day Sprint Duration = Number of (Duration) days per SprintVelocity = Points per Sprint 20 AV= ≈ 4 6

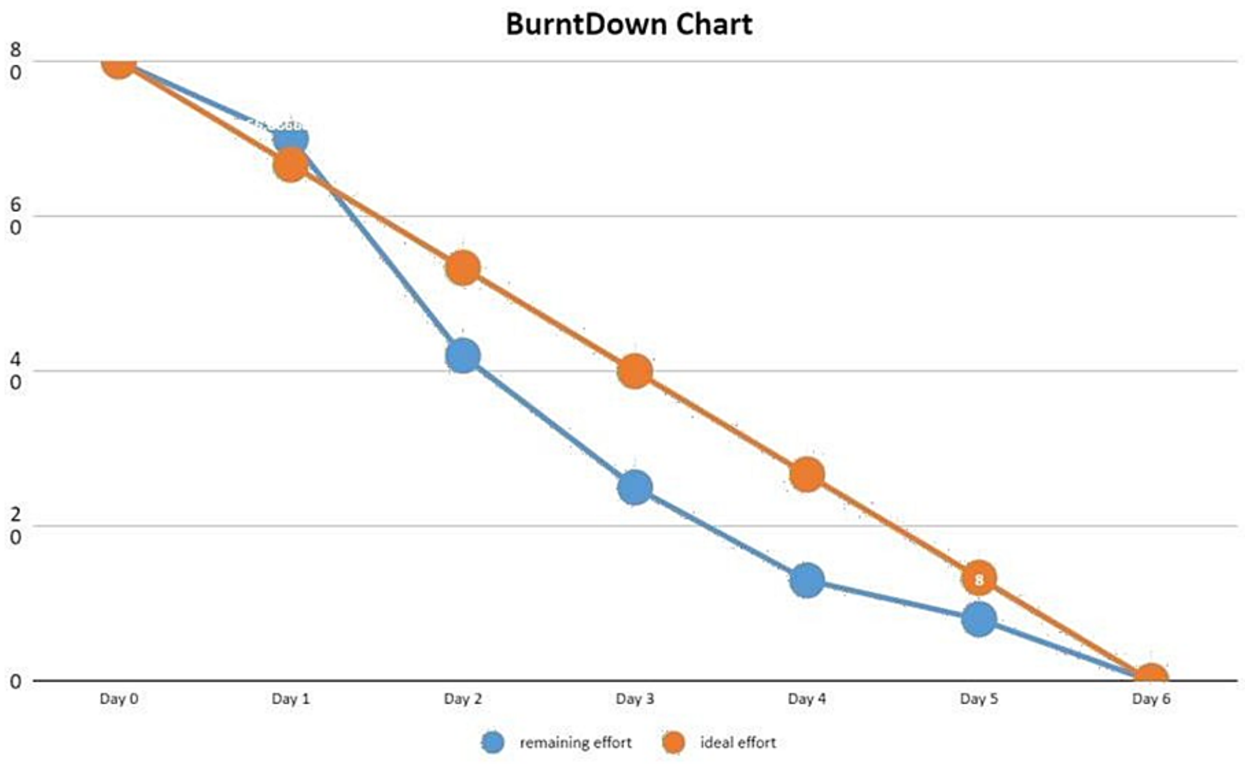
Therefore, the AVERAGE VELOCITYIS 4 POINTS PER SPRINT

Burndown Chart:

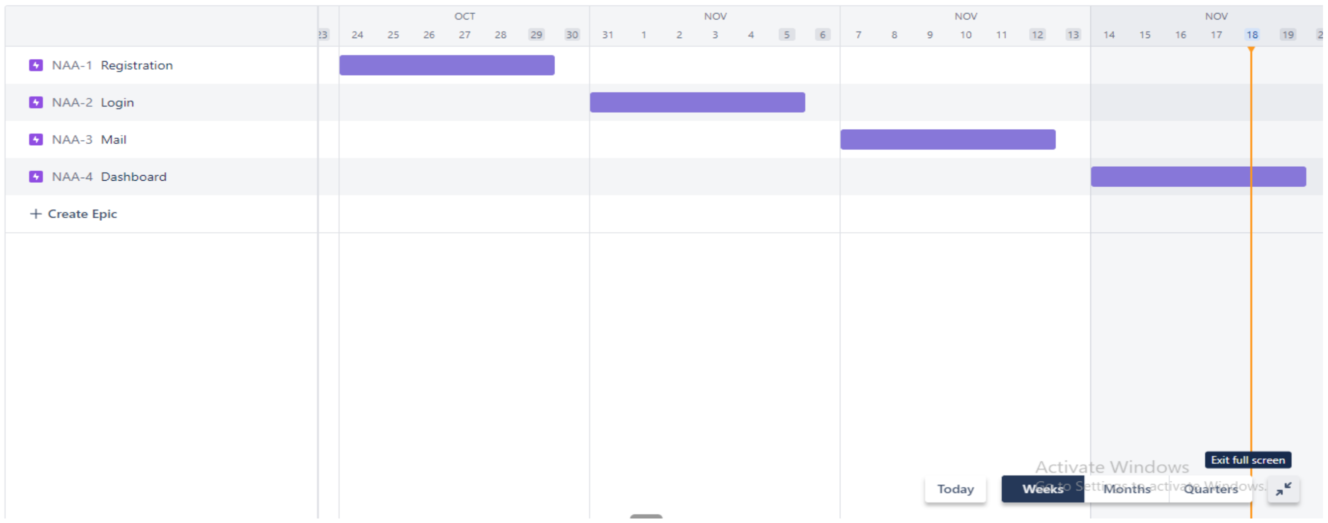
A burn down chart is a graphical representation of work left to do versus time. It is oftenused in agilesoftware develop



	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



6.3 REPORTS FROM JIRA



7.CODING & SOLUTIONING (Explain the features addedin the project along with code)

7.1Feature 1

Register.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="register.css">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body background="nutrition bg.png">
<div class="container">
<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">Phone Number</span>
<input type="text"
placeholder="Enter your number"
name="phonenumber">
```

```
</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">
 <a href="login.html"> <center>REGISTER </center></a>
<br><br>
already registered?
 <a href="login.html">login </a>
</div>
</form>
</div>
</div>
</body>
</html>
```

Login.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="register.css">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body background="nutrition bg.png">
<div class="container">
<div class="title">Login</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">Username</span>
<input type="text" placeholder="Enter
your username"name="username">
</div>

<br><br>
<div class="input-box">
<span class="details">Password</span>
<input type="password" placeholder="Enter
your password"name="passwords">
</div>
</div>
<div class="button">
<a href="USER DETAILS.html">
<center>SUBMIT</center></a>
<br><br>
not registered?

 <a href="register.html"> register </a>
</div>
</form>
</div>
</div>
</body>
</html>
```

USER DETAILS.html

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

```
 <div class="container">
 <div class="title">USERDETAILS</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">HEIGHT</span>
 <input type="text" placeholder="Enter your Height" name="fullname">
 </div>
 <div class="input-box">
 <span class="details">WEIGHT</span>
 <input type="text" placeholder="Enter your Weight" name="fullname">
 </div>
 <div class="input-box">
 <span class="details">BLOOD PRESSURE </span>
 <input type="text" placeholder="Enter your B.P mmHg value" name="fullname">
 </div>
 <div class="input-box">
 <span class="details">DIABETICS </span>
 <input type="text" placeholder="Enter your Diabetics mg/dl value" name="fullname">
 </div>
 <div class="input-box">
 <span class="details">AGE</span>
 <input type="text" placeholder="Enter your Age" name="fullname">

 </div>
 </div>
 <div class="button">
 <a href="dashboard.html">
 <center>SUBMIT</center></a>
</div>
 </form>
```

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

7.2.Feature 2

Dashboard.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">
  <link rel="stylesheet" href="static/styles.css">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
  <title>Nutrition Assistant</title>
</head>

    <div class="row align-items-md-stretch">
      <div class="col-md-6 my-3">
        <div class="h-100 p-5 text-bg-dark rounded-3">
          <h2>Upload food image</h2>
          <form action = "/dashboard" method= "POST" enctype="multipart/form-data">
            <input class="my-3 form-control" type="file" name="file" required/>
            <a href="food details.html"<center>ANALYZE </center></a>



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

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw
3" crossorigin="anonymous"></script>
</body>
</html>
```

clarifai_setup

```
import os
import time
from flask import Flask
from typing import Tuple

from grpc._channel import _Rendezvous

from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2, service_pb2_grpc
from clarifai_grpc.grpc.api.status import status_code_pb2
from clarifai_grpc.grpc.api.status.status_pb2 import Status




DOG_IMAGE_URL = "https://samples.clarifai.com/dog2.jpeg"
TRUCK_IMAGE_URL = "https://s3.amazonaws.com/samples.clarifai.com/red-
truck.png"
TRAVEL_IMAGE_URL = "https://samples.clarifai.com/travel.jpg"
NON_EXISTING_IMAGE_URL = "http://example.com/non-existing.jpg"
RED_TRUCK_IMAGE_FILE_PATH = os.path.dirname(__file__) + "/assets/red-
truck.png"

BEER_VIDEO_URL = "https://samples.clarifai.com/beer.mp4"
CONAN_GIF_VIDEO_URL =
"https://samples.clarifai.com/3o6gb3kkXfLvdKEZs4.gif"
TOY_VIDEO_FILE_PATH = os.path.dirname(__file__) + "/assets/toy.mp4"

GENERAL_MODEL_ID = "aaa03c23b3724a16a56b629203edc62c"


def get_status_message(status: Status):
    message = f"{status.code} {status.description}"
    if status.details:
        return f"{message} {status.details}"
    else:
        return message


def metadata(pat=False):
    if pat:
        return (("authorization", "Key %s" %
os.environ.get("CLARIFAI_PAT_KEY")),)
    else:
        return (('authorization', 'Key
ebecf4a92224420ea072cbbe9880c8ca'),)


def both_channels(func):
    """
    A decorator that runs the test first using the gRPC channel and then
using the JSON channel.
    :param func: The test function.
    :return: A function wrapper.
    """

    def func_wrapper():
        channel = ClarifaiChannel.get_grpc_channel()
        func(channel)

        channel = ClarifaiChannel.get_json_channel()
        func(channel)

    return func_wrapper


def wait_for_inputs_upload(stub, metadata, input_ids):
    for input_id in input_ids:
        while True:
            get_input_response = stub.GetInput(
                service_pb2.GetInputRequest(input_id=input_id),
metadata=metadata
            )
            raise_on_failure(get_input_response)
            if get_input_response.input.status.code ==
status_code_pb2.INPUT_DOWNLOAD_SUCCESS:
                break
            elif get_input_response.input.status.code in (
                status_code_pb2.INPUT_DOWNLOAD_PENDING,
                status_code_pb2.INPUT_DOWNLOAD_IN_PROGRESS,
            ):
                time.sleep(1)
            else:
                error_message =
get_status_message(get_input_response.status)
                raise Exception(
                    f"Expected inputs to upload, but got {error_message}.
"
                    f"Full response: {get_input_response}"
                )
    # At this point, all inputs have been downloaded successfully.


def wait_for_model_trained(stub, metadata, model_id, model_version_id,
user_app_id=None):
    while True:
        response = stub.GetModelVersion(
            service_pb2.GetModelVersionRequest(
                user_app_id=user_app_id, model_id=model_id,
version_id=model_version_id
            ),
            metadata=metadata,
        )
        raise_on_failure(response)
        if response.model_version.status.code ==
status_code_pb2.MODEL_TRAINED:
            break
        elif response.model_version.status.code in (
            status_code_pb2.MODEL_QUEUED_FOR_TRAINING,
            status_code_pb2.MODEL_TRAINING,
        ):
            time.sleep(1)
        else:
            message = get_status_message(response.model_version.status)
            raise Exception(
                f"Expected model to be trained, but got model status:
{message}. Full response: {response}"
            )
    # At this point, the model has successfully finished training.


def wait_for_model_evaluated(stub, metadata, model_id, model_version_id):
    while True:
        response = stub.GetModelVersionMetrics(
            service_pb2.GetModelVersionMetricsRequest(
                model_id=model_id, version_id=model_version_id
            ),
            metadata=metadata,
        )
        raise_on_failure(response)
        if response.model_version.metrics.status.code ==
status_code_pb2.MODEL_EVALUATED:
            break
        elif response.model_version.metrics.status.code in (
            status_code_pb2.MODEL_NOT_EVALUATED,
            status_code_pb2.MODEL_QUEUED_FOR_EVALUATION,
            status_code_pb2.MODEL_EVALUATING,
        ):
            time.sleep(1)
        else:
            error_message = get_status_message(response.status)
            raise Exception(
                f"Expected model to evaluate, but got {error_message}.
Full response: {response}"
            )
    # At this point, the model has successfully finished evaluation.


def raise_on_failure(response, custom_message=""):
    if response.status.code != status_code_pb2.SUCCESS:
        error_message = get_status_message(response.status)
        if custom_message:
            if not str.isspace(custom_message[-1]):
                custom_message += " "
        raise Exception(
            custom_message
            + f"Received failure response `{error_message}`. Whole
response object: {response}"
        )
```

```
def post_model_outputs_and_maybe_allow_retries(
    stub: service_pb2_grpc.V2Stub,
    request: service_pb2.PostModelOutputsRequest,
    metadata: Tuple,
):
    return _retry_on_504_on_non_prod(lambda:
stub.PostModelOutputs(request, metadata=metadata))


def _retry_on_504_on_non_prod(func):
    """
    On non-prod, it's possible that PostModelOutputs will return a
temporary 504 response.
    We don't care about those as long as, after a few seconds, the
response is a success.
    """
    MAX_ATTEMPTS = 15
    for i in range(1, MAX_ATTEMPTS + 1):
        try:
            response = func()
            if (
                len(response.outputs) > 0
                and response.outputs[0].status.code !=
status_code_pb2.RPC_REQUEST_TIMEOUT
            ):  # will want to retry
                break
        except _Rendezvous as e:
            grpc_base = os.environ.get("CLARIFAI_GRPC_BASE")
            if not grpc_base or grpc_base == "api.clarifai.com":
                raise e

            if "status: 504" not in e._state.details and "10020 Failure"
not in e._state.details:
                raise e

            if i == MAX_ATTEMPTS:
                raise e

            print(f"Received 504, doing retry #{i}")
            time.sleep(1)
    return response
```

8.TESTING

8.1.Test Cases

Test case ID	Feature Type	Component	TestScenario	pre-reuisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	TC for Automation(Y/N)	BUG ID
Login Page_TC_OO1	Functional	Home Page	Verif y user is able to see the Logi n/ Signup popu p when userclicked onMy account button		1.Enter URL andclick go 2.Click onMy Account drop down button3 .Verify login/ Singup popup displayd or not	login.html	Login/ Signup popups hould display	Working as expected	pass		
login page_TC_OO2	U1	Home page	verify the u1 elements in login\signup	signip page	1.Enter URL and click go 2.Click onMy Account dropdown button 3.Verify login/Sing up	register .html	Application should show below UI elements: a.emailtext box b.password text box	Working as expected	pass		

					popup with belowUI elements: a.email text box b.password text box c.Login buttond. New customer?		c.Login button with orange colour d.New customer?				

8.User Acceptance Testing

8.1.Purpose of Document

The purposeof this documentis to briefly explain the test coverageand openissuesof the project at the time of therelease to User Acceptance Testing (UAT).

8.2. Defect Analysis

This reportshows the number of resolvedor closed bugs at each severity level,and how they were resolved.

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won’t fix	0	5	2	1	8
Totals	24	14	13	26	77

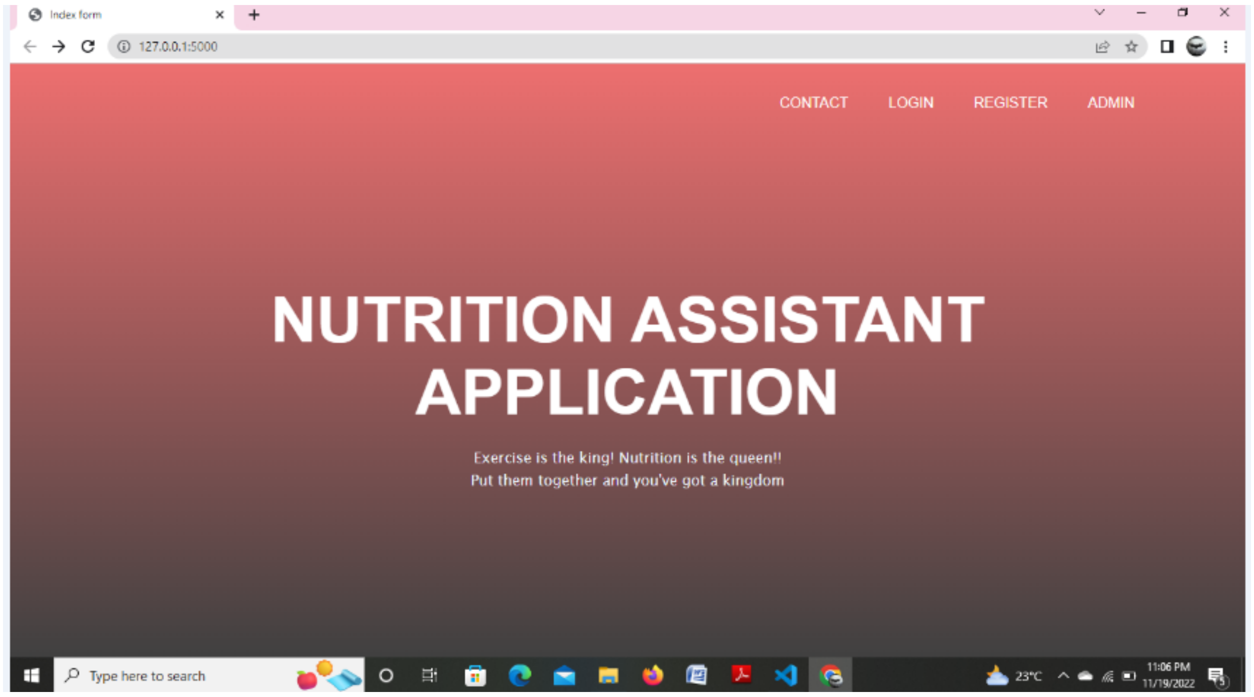
8.3. Test Case Analysis

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

Section	Total cases	Not tested	Fail	Pass
Print engine	7	0	0	7
Client application	51	0	0	51
Security	2	0	0	2
Outsource shipping	3	0	0	3
Exception reporting	9	0	0	9
Final report output	4	0	0	4
Version control	2	0	0	2

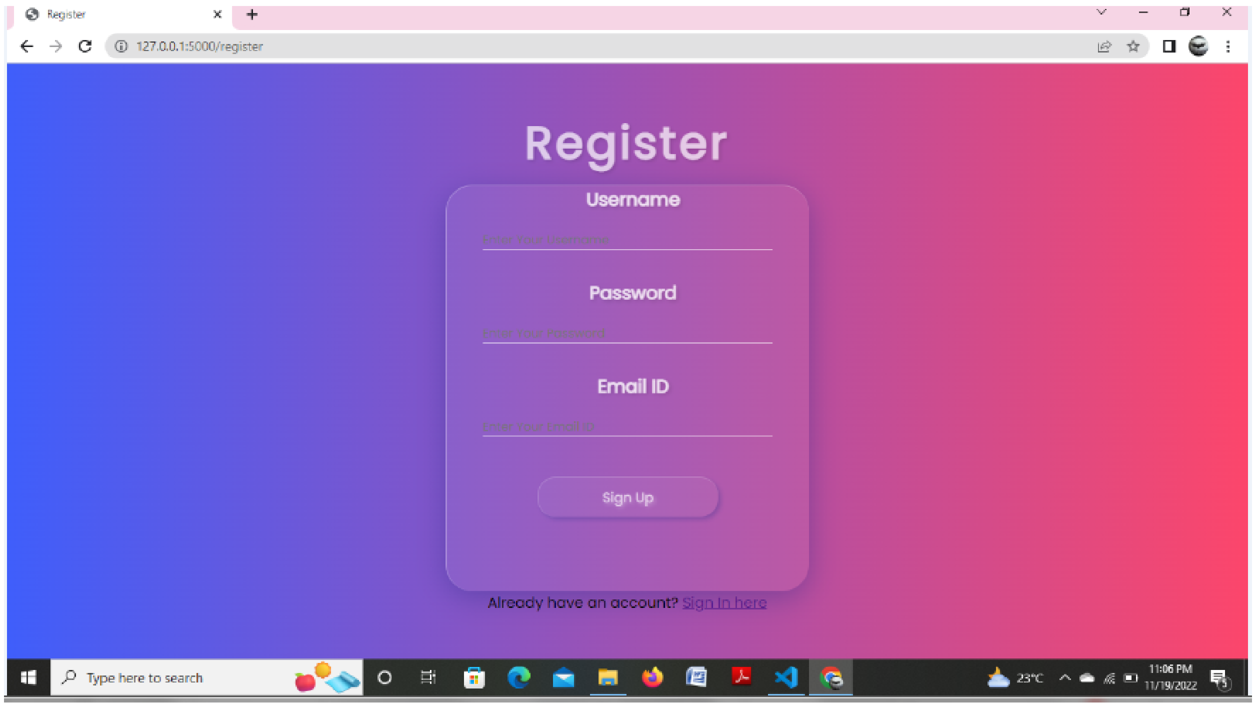
9.RESULTS

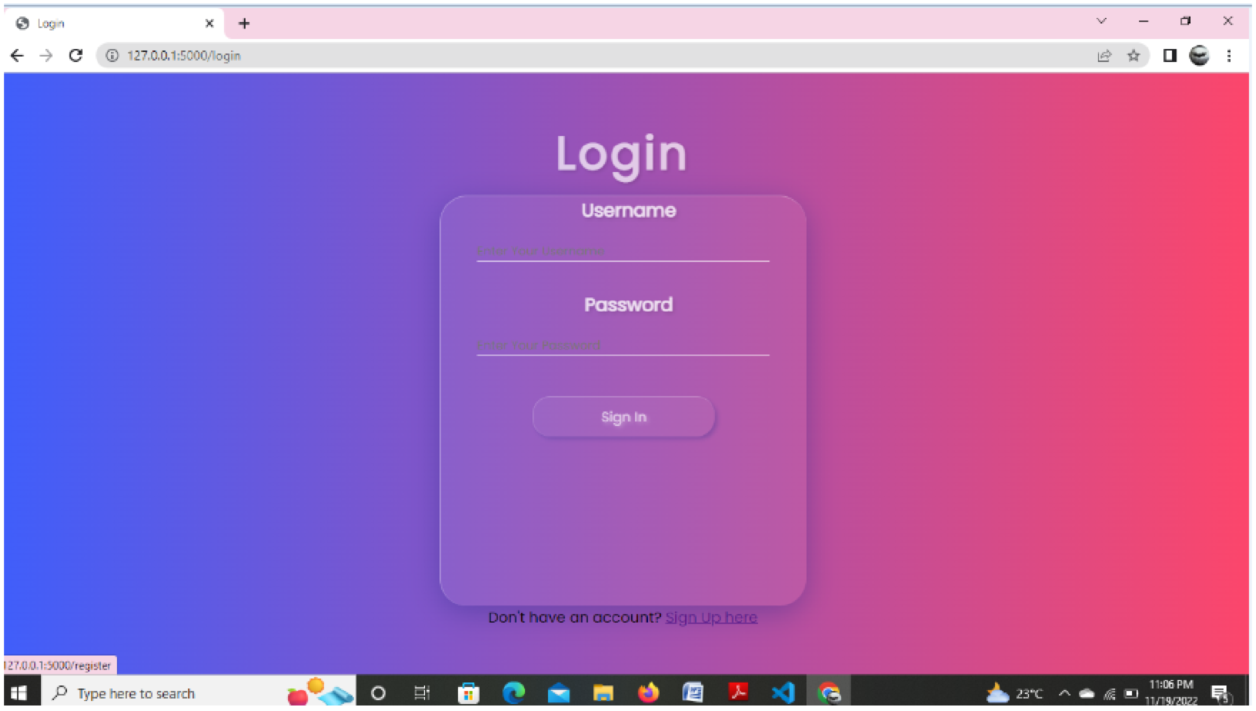
9.1Performance Metrics

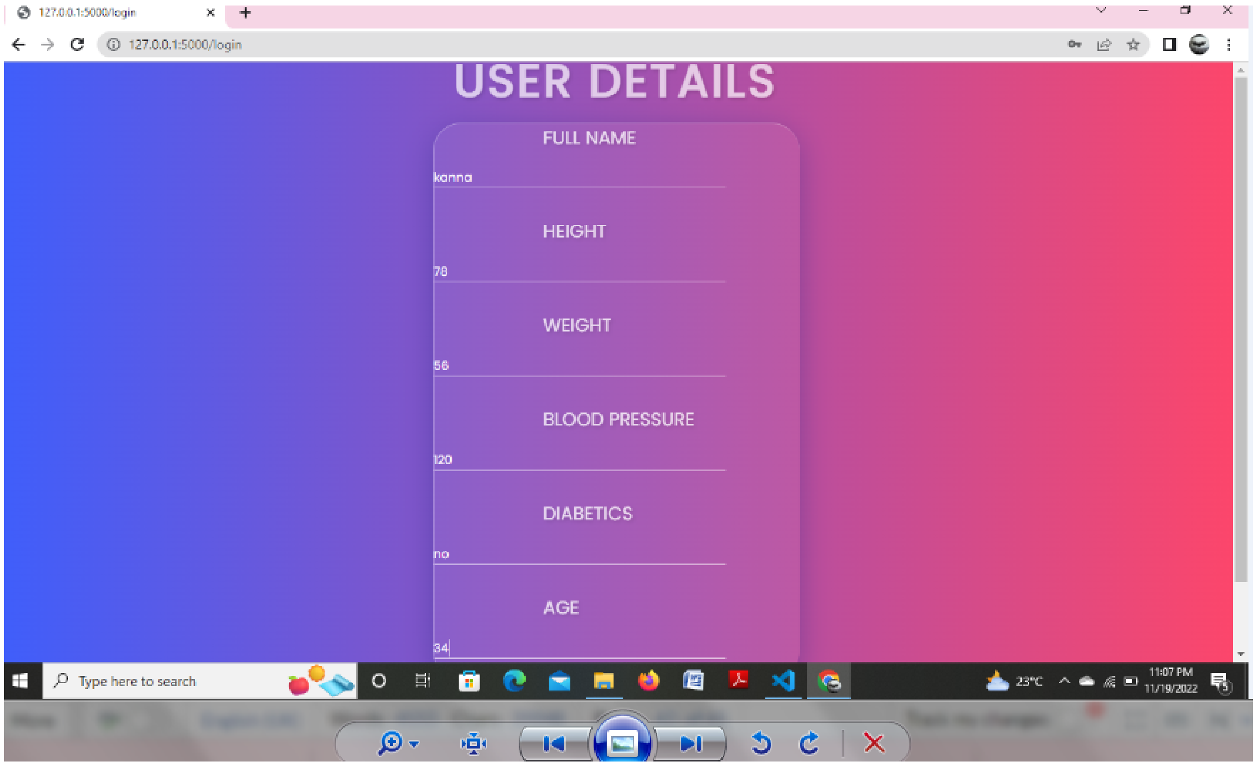


NUTRITION ASSISTANT APPLICATION

Exercise is the king! Nutrition is the queen!!
Put them together and you've got a kingdom


Register
127.0.0.1:5000/register
Register
Username
Password
Email ID
Sign Up
Already have an account? Sign In here
Type here to search
23°C
11:06 PM
11/19/2022


Login
127.0.0.1:5000/login
Login
Username
Password
Sign In
Don't have an account? Sign Up here
127.0.0.1:5000/register
Type here to search
23°C
11:06 PM
11/19/2022



127.0.0.1:5000/login

USER DETAILS

FULL NAME
kanna

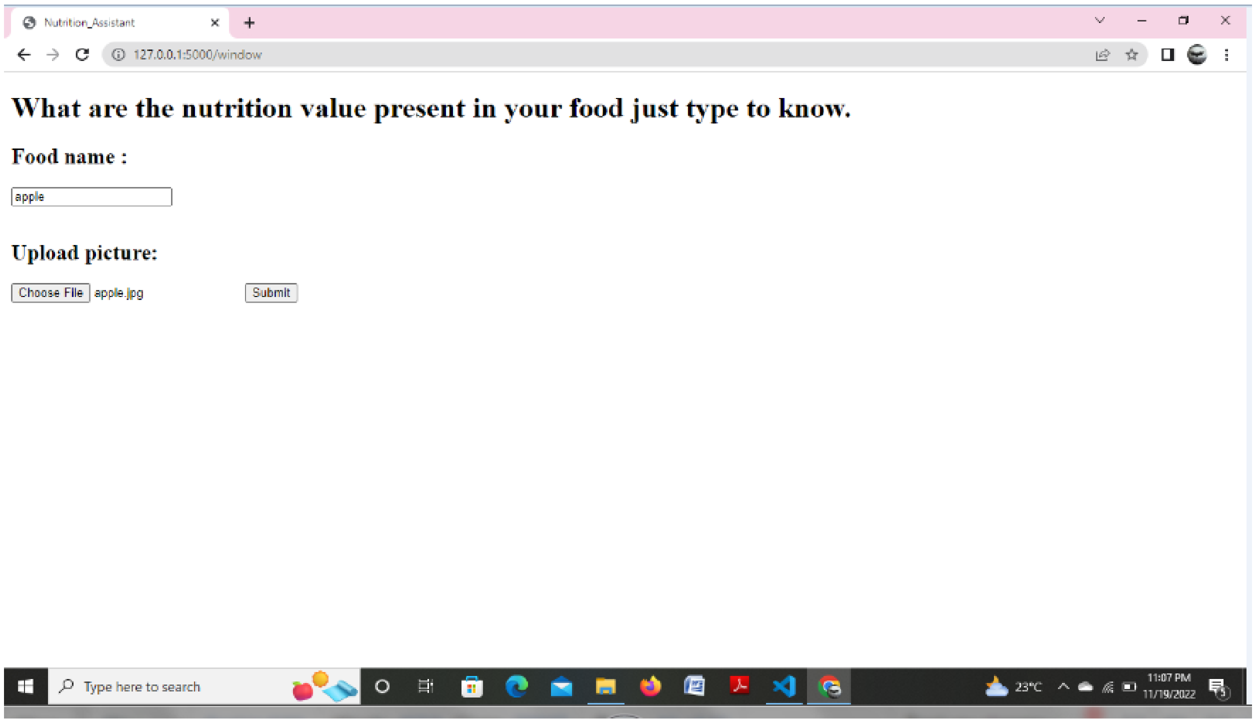
HEIGHT
78

WEIGHT
56

BLOOD PRESSURE
120

DIABETICS
no

AGE
34



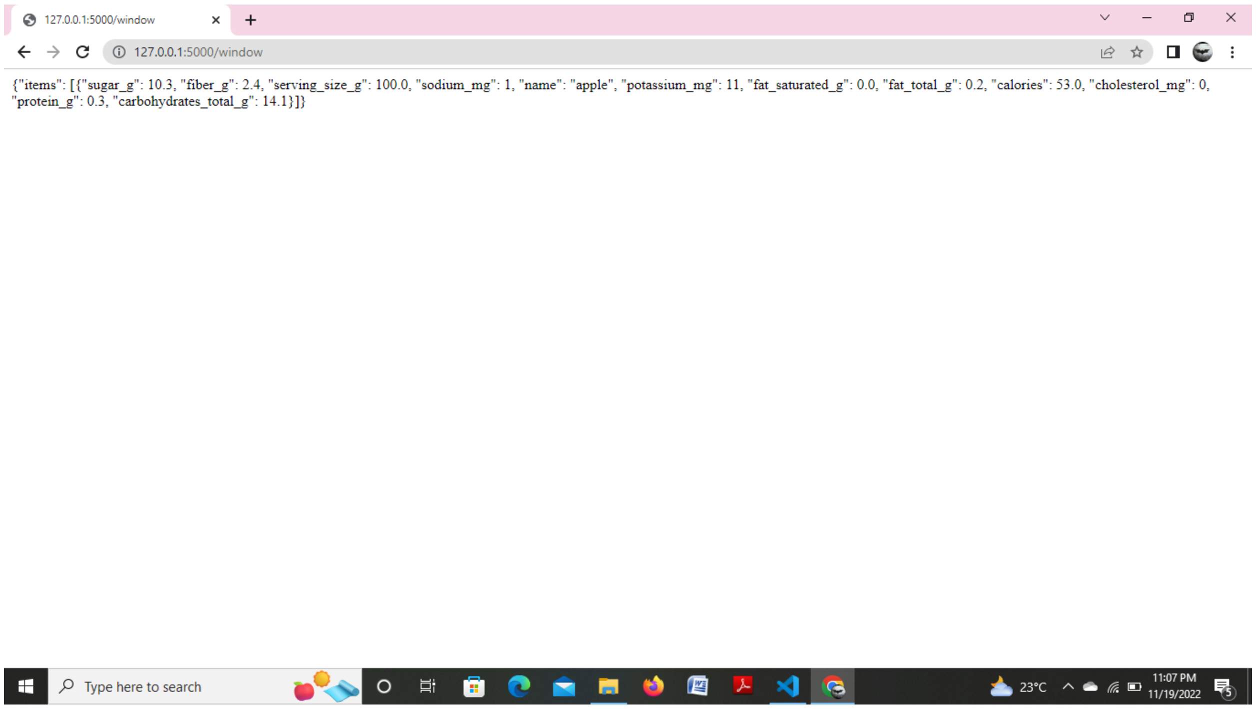
Nutrition_Assistant

127.0.0.1:5000/window

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

Food name :
apple

Upload picture:
Choose File apple.jpg Submit



```
{"items": [{"sugar_g": 10.3, "fiber_g": 2.4, "serving_size_g": 100.0, "sodium_mg": 1, "name": "apple", "potassium_mg": 11, "fat_saturated_g": 0.0, "fat_total_g": 0.2, "calories": 53.0, "cholesterol_mg": 0,
"protein_g": 0.3, "carbohydrates_total_g": 14.1}]}
```

10. ADVANTAGES & DISADVANTAGES

The advantages of nutritionprograms are as follows:

1. It gives a maintained strategy of healthy eating habits.
2. It delivers information on the nutritional value of foods and how balanced and healthy eatinghabits are important for us.
3. It reduces theamount of unnecessary food such as fat that people consume a lot.
4. Increase health literacy.

The disadvantages of nutrition programs are as follows:

1. Sometimes it makes a level of disbalance in the balanceddiet of an individual.
2. Sometimes, it is considered one of the major factorsof weight gain.

11.CONCLUSION

In this study, we conducted a critical review of mobile apps from three popular app stores. Our search results identified a total of 473 related apps, from which we selected and evaluated 80 apps using our modified app rating tool. We devised this app rating tool specifically for analyzing food consumption tracking and recommendation apps by adopting and extending existing mobile app rating scales.Using this rating tool, we evaluated the selected 80 apps and analysed and identified their design faults. According to our evaluation, most of the existing mobile apps in the app stores do not meet the essential requirements for correctly tracking food consumption and recommendations.
Also, there has been much researchon food recommendations but thisfeatureis absent in most of the evaluatedapps, that is why this feature needs to be included in futureapps. These apps suggest diet plans, recommend foods to users, and estimate nutrient values, so an expertdietitian or nutritionist should be involvedin their development. Also, enrichment of thedatabaseis required as nowadays multiple food datasetsare available. Softwarequalities (aesthetics,generalfeatures, performance, and usability) also play a vital role in commercial apps and thus developers

Need to consider these matters. Nonetheless, the analysis provided here covers a variety of general quality features and specific functional features that can be used in food consumption tracking and recommendation apps to provide consumers with a realistic and evidence-based experience. Studies show how people use smart phones to improve their fitness and obesity literacy, as well as the overall statusof the commercial product market for food consumption trackingand recommendation apps.

This study will open the door to future researchers who focus on the implementation, effectiveness and performance measurement of food computing apps.

12.FUTURE SCOPE

Nutrition plays a pivotal role in leading a healthy life. It is a vital element required in every stage of life. Nutritious food intake and metabolism of nutrients are associated with the decreased risk of both infectious and non-communicable diseases. Nutritious diet is a major determinant of future health – physical, mentaland social health, not merelyan absence of disease.

13.APPENDIX

Source Code

Source Code

```
import os, re, string, random, time, datetime, requests,
sendgrid, random, flaskimport ibm_db
from sendgrid.helpers.mail import *
from flask import Flask, request, render_template, flash, redirect,
url_for,sessionfrom werkzeug.utils import secure_filename
from clarifai_grpc.channel.clarifai_channel importClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2, resources_pb2,
service_pb2_grpcfrom clarifai_grpc.grpc.api.status import
status_code_pb2


###############################################################
#######################


UPLOAD_FOLDER = 'static/uploads'
ALLOWED_EXTENSIONS =
set(['png', 'jpg','jpeg'])
SENDGRID_API_KEY = "SG.HwfSJ6D4Tba6O-h7fL1JlA.z2_qdNI-
iXOhrhdzsx05PiEPj3bbNKXF_Rms0eRis4c"


app = Flask(_name_
) app.secret_key =
"bimbilikibilapi"
app.config['UPLOAD_FOLDER'] =
    UPLOAD_FOLDER
    app.config['MAX_CONTENT_LENGTH'] = 16
    * 1024 * 1024


    conn    =    ibm_db.connect("DATABASE=bludb;HOSTNAME=b1bc1829-
6f45-4cd4-bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;Se
curity=SSL;P
ROTOCOL=TCPIP;UID=pzt20234;PWD=r7CB0AmR1QtOHfR4;","","")
#;SSLServerCertificate=DigiCertGlobalRootCA.crt


    YOUR_CLARIFAI_API_KEY =
    "af4bc9886c744e998ee0e20f104b1518"YOUR_APPLICA
    TION_ID = "test"
    SAMPLE_URL                                                         =
"https://res.cloudinary.com/swiggy/image/upload/f_auto,q_auto,fl_lossy/nxmlubuz
0b1qixa29gov "
    metadata = (("authorization", f"Key
    {YOUR_CLARIFAI_API_KEY}"),)channel =
    ClarifaiChannel.get_grpc_channel()
    stub = service_pb2_grpc.V2Stub(channel)


    RAPIDAPI_KEY = "74e62205b6msha6b4e69e0088de5p12c619jsn1ed9cc5e0727"

    def allowed_file(filename):
      return '.' in filenameand \
        filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
```

```
def sendMail(to, title, text):
  sg =
  sendgrid.SendGridAPIClient(api_key=SENDGRID_
  API_KEY)from_email =
  Email("nsnandhaa1@gmail.com")
  to_
  em
  ail
  =
  To
  (t
  o)
  su
  bje
  ct
  =
  tit
  le
  content = Content("text/plain", text)
  mail = Mail(from_email, to_email,subject,
  content) response =
  sg.client.mail.send.post(request_body=mail.g
  et())print(response.status_code)
  print(res
  ponse.b
  ody)
  print(res
  ponse.he
  aders)
```

```
@app.route("/forgot-pw",
methods=["GET", "POST"])def
forgotpw():
  if flask.request.method == "POST":
    data = flask.request.form
    username=data['username']
    code = ''.join(random.choices(string.ascii_letters, k=6))


    sql= "SELECT * FROM users
    WHERE username=?"
    stmt=ibm_db.prepare(conn,sql)ibm_d
    b.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    account=ibm_db.fetch_assoc(stmt)
    print(account)
    session['userid'] = account['USERID']

    insert_sql = "INSERT INTO VERIFY VALUES(?,?)"
    prep_stmt=ibm_db.prepare(conn,
    insert_sql)
    ibm_db.bind_param(prep_stmt, 1,
    account['USERID'])
    ibm_db.bind_param(prep
    _stmt, 2, code)
    ibm_db.execute(prep_st
    mt)


    sendMail(account['EMAIL'], "Verification Code", code)
    flash("We have sent a code to your registered email. please check spam
    folderalso.")return redirect(url_for("confirmMail"))
  flash("We will send you a confirmation code to your
  registered email")return render_template("forgot-
  pw.html")



@app.route("/confirm-mail",
methods=["GET", "POST"])def
confirmMail():
  session['LoggedIn'] = False
  if flask.request.method == "POST":
    data =
    flask.requ
    est.form
    usercode=
    data['code
    ']

    sql= "SELECT * FROM verify
    WHERE userid=?"
    stmt=ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,session['
    userid']) ibm_db.execute(stmt)
    verify=ibm_db.fetch_assoc(stmt)
    print(verify)


    dbcode =
    verify['C
    ODE']if
    usercode
    ==
    dbcode:
      session['LoggedIn'] = True
      delete_sql = "DELETE FROM verify
      WHERE CODE=?"
      prep_stmt=ibm_db.prepare(conn,
      delete_sql) ibm_db.bind_param(prep_stmt,
      1, dbcode) ibm_db.execute(prep_stmt)

      flash("Email verified. Enter
      new password")return
      redirect(url_for("changepw"
      ))
    else:
      flash("Error")
      return
  render_template("confirm-
  mail") return
  render_template("confirm-
  mail.html")

@app.route("/change-pw",
methods=["GET", "POST"])def
changepw():
  if flask.request.method == "POST" and
    session['LoggedIn']:data =
    flask.request.form
    password=data['pw']
    sql = "UPDATE users SET PASSWORD=? WHERE USERID=?"
    prep_stmt=ibm_db.prepare(co
    nn, sql) print(password,
    session['userid'])
    ibm_db.bind_param(prep_stm
    t, 1, password)
    ibm_db.bind_param(prep_stmt, 2,
    session['userid'])
```

```
    ibm_db.execute(prep_stmt)
    flash("Password
  changed.") return
  redirect(url_for("l
  ogin")) else:
    flash("verification error")
    redirect(url_for("confirmMail"))
  return render_template("change-pw.html")



@app.route("/register",
methods=["GET", "POST"])def
reg():
  if flask.request.method == "POST":


    data =
    flask.request.fo
    rm
    email=data['emai
    l']
    username=data['
    username']
    password=data['
    pw']


    sql= "SELECT * FROM users
    WHERE username=?"
    stmt=ibm_db.prepare(conn,sql)ibm_d
    b.bind_param(stmt,1,username)
    ibm_db.execute(stmt)
    account=ibm_db.fetch_assoc(stmt)
    pri
    nt(
    acc
    ou
    nt)
    if
    acc
    ou
    nt:
      flash("Account already exists!")
    elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
      flash("invalid email address")
    elif not re.match(r'[A-Za-z0-9]+', username):
      flash("name must contain only characters
    and numbers")else:
      insert_sql = "INSERT INTO users
      VALUES(?,?,?,?)"
      prep_stmt=ibm_db.prepare(conn,
      insert_sql)
      ibm_db.bind_param(prep_stmt, 1,
      username)
      ibm_db.bind_param(prep_stmt, 2, email)
      ibm_db.bind_param(prep_stmt, 3, password)
      ibm_db.bind_param(prep_stmt, 4, ''.join(random.choices(string.ascii_letters,
      k=16)))ibm_db.execute(prep_stmt)
      flash("logged in")


    return
  redirect(url_for("dashb
  oard"))
  returnrender_template(
  "reg.html")
```

```
@app.route("/login",
methods=["GET", "POST"])def
login():
  if flask.request.method == "POST":


    data = flask.request.form
    username=data['username']
    password=data['pw']



    sql = "SELECT * FROM users WHERE username=?
    AND password=?"stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt, 1, username)
    ibm_db.bind_param(stm
    t, 2, password)
    ibm_db.execute(stmt)
    account =
    ibm_db.fetch_assoc(
    stmt)print(account)
    if account:
      session['LoggedIn'] =
      True session['userid'] =
      account['USERID']
      session['username'] =
      account['USERNAME']
      userid= account['USERID']
      flash("logged in")
      return
    redirect(url_for("dashb
    oard"))else:
```

```
    flash("error")

  return render_template("login.html")



@app.route("/dashboard",
methods=["GET", "POST"])def
dashboard():
  global request
  if flask.request.method == "POST" and
    session['LoggedIn']:if 'file' not in
    flask.request.files:
      flash('No file part')
      return
    redirect(flask.requ
    est.url)file =
    flask.request.files[
    'file']
    if file.filename == '':
      flash('No image
      selected')
      returnredirect(fla
      sk.request.url)
    if file and
      allowed_file(file.filename)
      :filename =
      secure_filename(file.filena
      me)

      file.save(os.path.join(app.config['UPLOAD_FOLDER'],
      filename))flash('Image successfully uploaded')
```

```
with open(os.path.join(app.config['UPLOAD_FOLDER'], filename),
  "rb") as f:file_bytes = f.read()


request =
    service_pb2.PostModelOutputs
    Request(model_id="food-item-
    v1-recognition",
    user_app_id=resources_pb2.UserAppIDSet(app_id=YOUR_APPLICATI
    ON_ID), 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:
    print(response)
    raise Exception(f"Request failed,status code:

{response.status}")foodname=

response.outputs[0].data.concepts[0].name

ingredients = ''
```

```
for concept in response.outputs[0].data.concepts:
  ingredients += f"{concept.name}: {round(concept.value, 2)}, "


nutritionValues = ''
#  nutritionApiUrl  =  "https://spoonacular-recipe-
food-nutrition-v1.p.rapidapi.com/recipes/guessNutrition"
# querystring = {"title":foodname}

# headers = {
#  "X-RapidAPI-Key": RAPIDAPI_KEY,
#  "X-RapidAPI-Host": "spoonacular-recipe-food-nutrition-
v1.p.rapidapi.com"# }
#  response  =  requests.request("GET",  nutritionApiUrl,
headers=headers,params=querystring)
# nutritions
=
response.te
xtnutritions
= {
"recipesUse
d": 10,
"calories": {
  "v
  alu
  e":
  47
  0,
  "u
  nit
  ":
  "c
  alo
  rie
  s",
  "confidenceRang
    e95Percent":
    {"min": 408.93,
    "max": 582.22
  },
  "standardDeviation": 139.8
},
"fat": {
  "va
  lue
  ":
  17,
  "u
  nit
  ":
  "g
  ",
  "confidenceRang
    e95Percent":
    {"min": 12.81,
    "max": 21.36
  },
  "standardDeviation": 6.9
},
"protein": {
  "va
  lue
  ":
  15,
    "u
    nit
    ":
    "g
    ",
    "confidenceRang
      e95Percent":
      {"min": 9.06,
      "max": 29.78
    },

    "standardDeviation": 16.71
  },
  "carbs": {
    "va
    lue
    ":
    65,
    "u
    nit
    ":
    "g
    ",
    "confidenceRang
      e95Percent":
      {"min": 57.05,
      "max": 77.9
    },
    "standardDeviation": 16.81
  }
}
nutritions.pop('r
)

        ecipesUsed')for
        i in nutritions:
          nutritionValues += f"{i}: {nutritions[i]['value']} {nutritions[i]['unit']}, "



        sql = "INSERT INTO foods VALUES(?,?,?,?,?)"
        stmt=ibm_db.prepare(conn,
        sql) ibm_db.bind_param(stmt,
        1, session['userid'])
        ibm_db.bind_param(stmt,     2,      datetime.datetime.now().strftime('%Y-%m-%d
%H:%M:%S'))
        ibm_db.bind_param(stmt, 3, foodname)
        ibm_db.bind_param(stmt, 4, ingredients)
        ibm_db.bind_param(stmt, 5,
        nutritionValues)
        ibm_db.execute(stmt)

        #
        os.remove(os.path.join(app.config['UPLOAD_FOLDER
        '], filename))returnrender_template("dashboard.html",
          filename = filename,
          username =
          session['username'
          ], foodname =
          foodname,
          ingredients =
          ingredients,
          nutritionValues =
          nutritionValues,

        )
```

```
    else:
      flash('Allowed image formats -
      png, jpg, jpeg')
      returnredirect(flask.request.url)

  elif session['LoggedIn']:
    return render_template("dashboard.html",
  username=session['username'])else:
    return redirect(url_for("login"))



@app.route('/logout',
methods=["GET", "POST"])def
logout():
  session.pop('Lo
  ggenIn', None)
  session.pop('us
  erid', None)
  session.pop('us
  ername', None)
  return render_template("index.html")



@app.route('/display/<filename>',
methods=["GET", "POST"])def
display(filename):
  print(filename)
  return redirect(url_for('static', filename='uploads/' + filename), code=301)
```

```
@app.route('/app',
methods=["GET", "POST"])def
other():
  return render_template("index.html")



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


if__name__== "__main__":
  app.run(host ='0.0.0.0', port = 5000)
```

Github link: https://github.com/IBM-EPBL/IBM-Project-48453-1660807413

Demo video : https://www.youtube.com/embed/zwbkvYjTNZE