CAD - NUTRITION ASSISTANT APPLICATION

TEAM ID-PNT2022TMID29412

(BATCH-B7-1A3E)

PROJECT REPORT
Submitted by

S.Aiswarya roy [510419104004]
T.Aishwarya [510419104003]
S.Amirtha Devi [510419104007]
S.Dheekshana [510419104023]
M.Gayathri [510419104027]

BACHELOR OF ENGINEERING

COMPUTER SCIENCE AND ENGINEERING
ARUNAI ENGINEERING COLLEGE,TIRUVANNAMALAI

ANNA UNIVERSITY - CHENNAI 6000025

INDEX

1. **INTRODUCTION**
    1. Project Overview
    2. Purpose
2. **LITERATURE SURVEY**
    1. Existing problem
    2. References
    3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    1. Empathy Map Canvas
    2. Ideation & Brainstorming
    3. Proposed Solution
    4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    1. Functional requirement
    2. Non-Functional requirements
5. **PROJECT DESIGN**
    1. Data Flow Diagrams
    2. Solution & Technical Architecture
    3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    1. Sprint Planning & Estimation
    2. Sprint Delivery Schedule
    3. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    1. Feature 1
    2. Feature 2
    3. Database Schema (if Applicable)
8. **TESTING**
    1. Test Cases

2. User Acceptance Testing

9. RESULTS

1. Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code GitHub & Project Demo Link

INTRODUCTION:

1.1 PROJECT OVERVIEW:

Due to the ignorance of healthy food habits, obesity rates are increasing at an alarming speed, and this is reflective of the risks to people's health. People need to control their daily calorie intake by eating healthier foods, which is the most basic method to avoid obesity. However, although food packaging comes with nutrition (and calorie) labels, it's still not very convenient for people to refer to App-based nutrient dashboard systems which can analyze real-time images of a meal and analyze it for nutritional content which can be very handy and improves the dietary habits, and therefore, helps in maintaining a healthy lifestyle.

This project aims at building a web App that automatically estimates food attributes such as ingredients and nutritional value by classifying the input image of food. Our method employs Clarifai's AI-Driven Food Detection Model for accurate food identification and Food API's to give the nutritional value of the identified food.

1.2 PURPOSE:

Nutrition assistant application help dieticians with providing proper nutrition at healthcare facilities. They determine patients' nutritional needs, assess risk factors, and plan meals and menus. They also ensure proper sterilization of plates and utensils.

Providing dieticians with the facility's meal and menu planning.Obtaining dietary information and assessing the nutritional habits of patients.Recording individual risk factors or dietary restrictions that might impact meal planning.Coordinating meal plans with nutritionists and healthcare professionals.

2. LITERATURE SURVEY:

2.1 Existing problem:

LICATION OF ARTIFICIAL INTELLIGENCE ON NUTRITION ASSESSMENT AND MANAGEMENT
Published year: May 2021
Author: Dr. Kavita Sudersanadas
Journal Name: EUROPEAN JOURNAL OF PHARMACEUTICAL AND MEDICAL RESEARCH

Summary:
The application of AI for the provision of food services to hospitalized patients is of immense scope. This review details the various ways through which AI can be applied for the nutrition assessment. Even though commercial AI-based nutritional assessment systems are available, many do not evaluate the nutrient intake, and the data available through them were not validated. Fat Secret is a commercially available AI-based food and nutrient assessment system that can evaluate the food's calorie content. Also, the major challenge posed by such systems is the availability of locally appropriate data sets. Hence further research and validation are essential in this field. AI-based nutrient intake assessment system is of immense value to obtain and assess food intake data in isolation wards and for the follow-up without contact. Methodology used: Artificial Intelligence.

Virtual Nutritionist using AI Publication year: June 2019 Author: Siddarthan Chitra Suseendran, Nanda Kishore B, Josephus Andrew, M.S. Rajya Shree Journal Name: International Journal of Engineering and Advanced Technology (IJEAT). Summary: In this way, a requirement for a full help for furnishing them with solid nourishment is a fundamental focus to reach. In this paper, we propose a model for a sustenance master framework which point is to give its clients the nourishment skill. It creates solid dinners for people in various ages as indicated by various criteria including their development stage, sexual orientation, and their wellbeing status. An application is created and a few contextual investigations are connected to show how the proposed model can be connected for deciding one's nourishment utilizing Artificial Intelligence (Machine and deep learning). Few key advantages are: Customized diet for any lifestyle and age along with various types of diets to choose from which acknowledges your pre medical conditions with appropriate macronutrient ratio split that ensures micronutrient supplement suggestions based on the foods you consume. Methodology used: Artificial Intelligence.

2.2 Reference:

1: Application of artificial Intelligence on nutrition assesment and management
Published year: May 2021
Author: Dr. Kavita Sudersanadas
Journal Name: European journal of pharamaceutical and medical research .

2: Smartphone Applications for Promoting Healthy Diet and Nutrition: A Literature Review

Published date: January 2016

Author: Steven S. Coughlin, Mary Whitehead, Joyce Q. Sheats, Jeff Mastromonico, Dale Hardy.

Journal Name: Jacobs J Food Nutra

3: Intelligent SVM Based Food Intake Measurement System

Published date: 15-17 July 2013

Author: Parisa Pouladzadeh ,Shervin Shi Mohammadi ,Tarik Arici

Journal Name: 2013 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA) .

2.3 Problem Statement Definition :

What does the problem effect?	If we eat mostly junk foods over many weeks, months, or years, there can be several long-term impacts on health. Frequent consumption of junk foods can also increase the risk of diseases such as hypertension and stroke. Other long-term effects of eating too much junk food include tooth decay and constipation.
What is the issue?	Major nutritional problems include: 1) Maternal nutritional anemia. 2) Protein energy malnutrition. 3) Vitamin A deficiency. 4) Lactation failure. 5) Addiction to milk feeding.
When does the issue occur?	Eating junk food on a regular basis can lead to an increased risk of obesity and chronic diseases like cardiovascular disease, type 2 diabetes, non-alcoholic fatty liver disease and some cancers.
Why is it important that we fix the problem?	If your appetite and taste have been affected by illness, medications or other health issues, you may have trouble eating and getting proper nutrition. These changes can affect your overall health.

3. IDEATION & PROPOSED SOLUTION :

3.1 Empathy Map Canvas :

3.2 Ideation & Brainstorming:

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!

Masimo

Nabora

Lisa

Jimin

3

Group Ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, 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.

PHYSICAL HEALTH

CONSULT A DIETITION

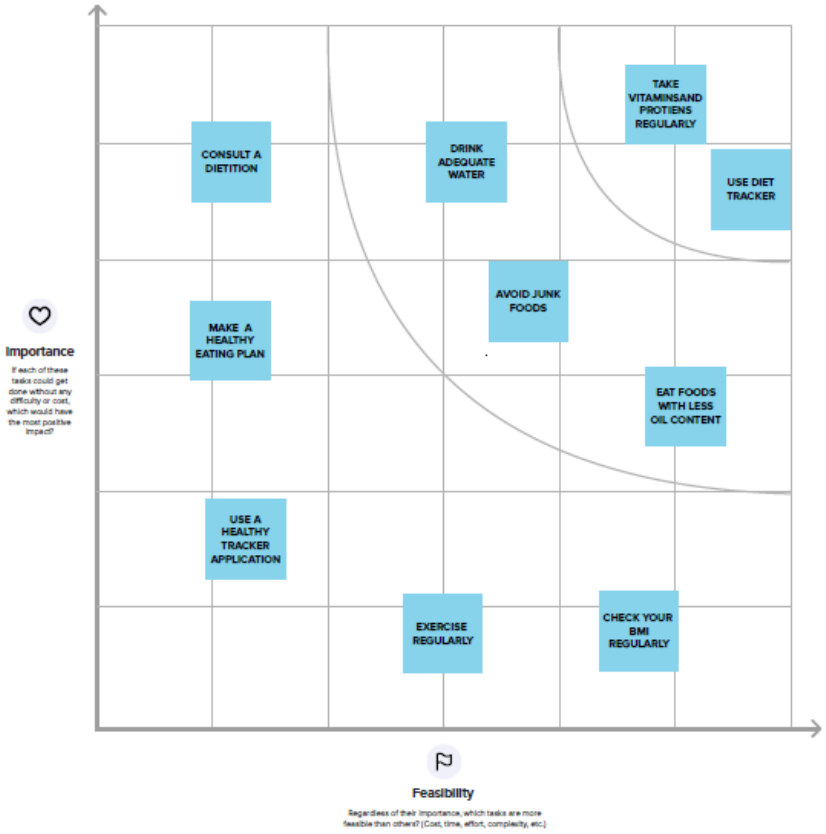
FOOD HABITS

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

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

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

Keep moving forward

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

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

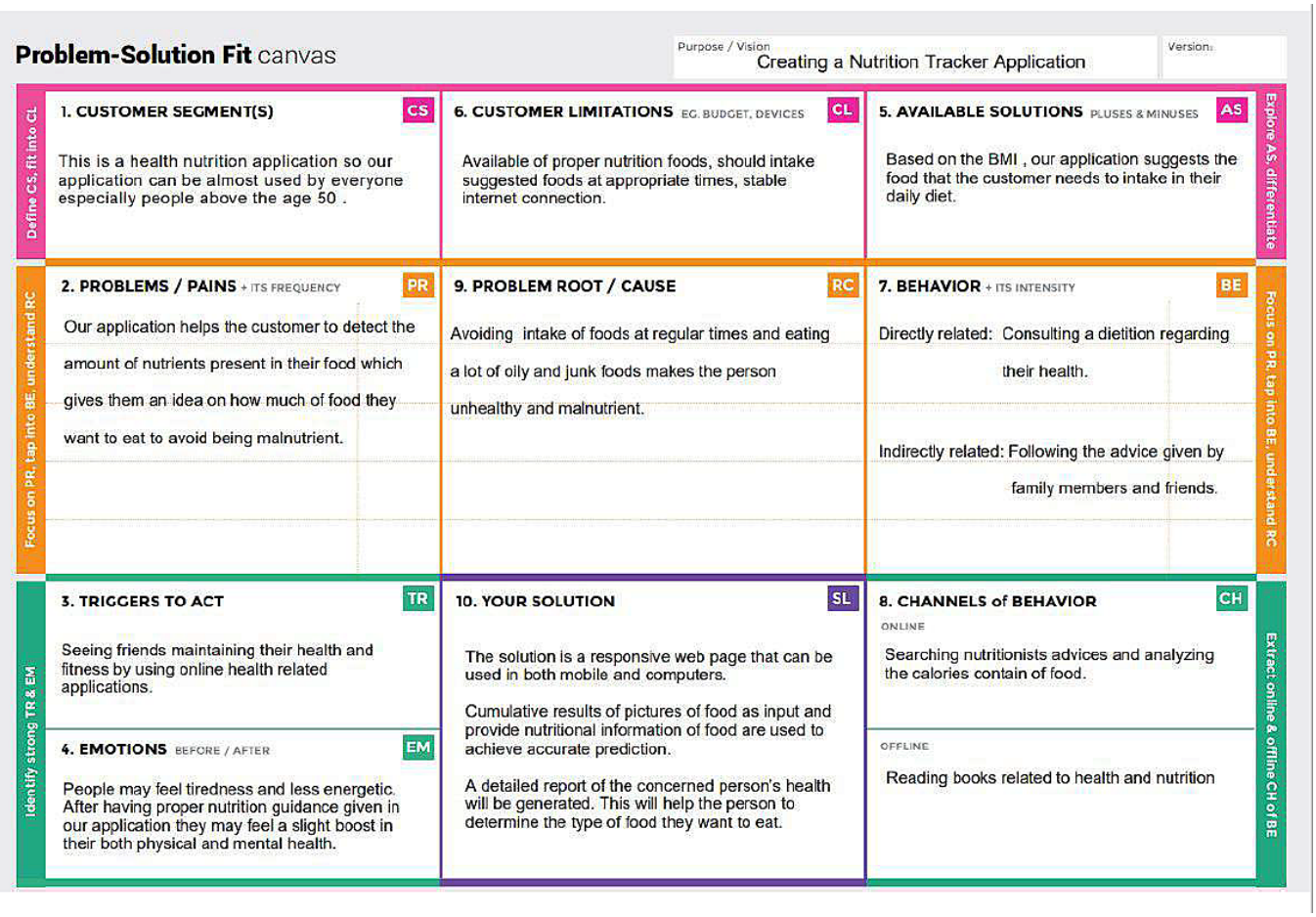
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3 Proposed Solution:

1. Problem Statement (Problem to be solved)	This project aims in building awebapplication that automatically estimatesfood contents like ingredients andnutritional value by classifying theinputimage of the given food. Our methodusesConvolutional Neural Networks(CNN) foraccurate foodidentification and rapid.com'sFood API to give the nutritional valueof theidentified food.
2. Idea / Solution description	The solution is a responsive web page that can be used in both mobile and computers. Cumulative results of pictures of food as input and provide nutritional information of food are used to achieve accurateprediction. A detailed report of theconcerned person's health will begenerated. This will help the persontodetermine the type of food they want to eat.
3.Novelty / Uniqueness	● Our method uses Convolutional Neural Networks(CNN) to accurately identifyfoods. ● Rapidapi.com's food API reports thenutritional value of identified foods. ● Checks the nutritional value of thefood ● Water monitoring. ● Suggests the type of food they want toeat. ● Regular tracking of food intake
4.Scalability of the Solution	Furthermore, features can be extendedinour application. Additional features suchassleep tracking, water tracking, foodintakecanbe measured.

3.4 Problem solution fit :



4.REQUIREMENT ANALYSIS:

4.1 Functional requirement :

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	● Registration through Mobile Number. ● Registration through Gmail.

FR-2	User Confirmation	● Confirmation via OTP through Email. ● Confirmation via OTP through Mobile Number. ● Confirmation via Retyping the password
FR-3	User Login	● Login with username and password.
FR-4	Food Detection	● Scanning the food by using Clarifai's AIDriven Food Detection Model.
FR-5	Display	● The system display the nutritional value of the food with the help of nutritional AP.
FR-6	Review	● User feedback and Rating
FR-7	Updates	● The latest updates will be display by Virtual Assistant so the user can easily familiarize themselves with the new services and policies. ● Notification message is send to user for new updates

4.2 Non-Functional requirements :

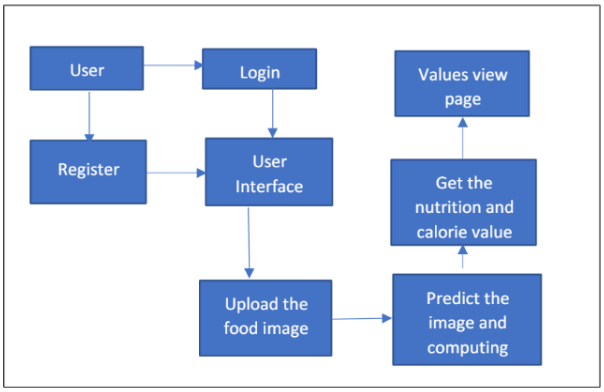
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	Usability	● It is a user-friendly application which tracks calories and nutritional value by just scanning the

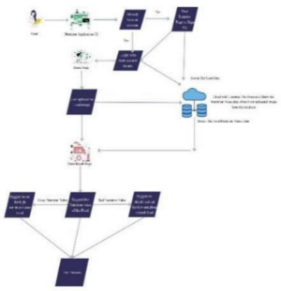
		image of the food. ● Only registered user is able to track the calories
NFR-2	Security	● Authentication and two step verification is done for security. ● Security for each and every user will be the first priority.
NFR-3	Availability	● This application is easily accessible because user needs only a smartphone with a good network connection. ● We can use this application anytime at anywhere. ● These apps offer diet and fitness tracking
NFR - 4	Performance	● Provision of best diet plan which makes the user to maintain a healthy weight. ● User satisfaction is ensured by getting their feedback.
NFR -5	Scalability	● This app can be updated in future as per the users feedback
NFR-6	Reliability	● The user gets the standard nutritional value of the given food. ● Provides consistent updates as per the customers feedback

5. PROJECT DESIGN:

5.1 Data Flow Diagrams:

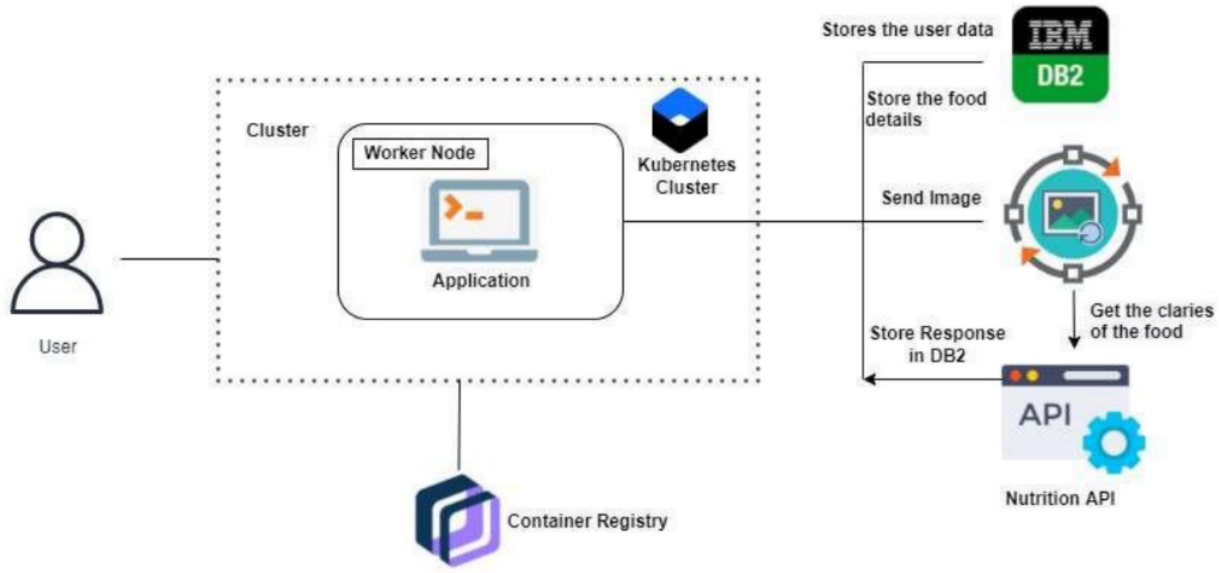
Example: (Simplified)





5.2 Solution & Technical Architecture :

Technical Architecture:



5.3 User Stories:

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and	I can access my account / dashboard	High	Sprint-1

			confirming my password.			
		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
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	User get into the dashboard and see's the different web pages to compute what the user needs		High	Sprint-1

Customer (Web user)	**Registration Prepare the milestones & activity list of the project.**	**USN-7**	**As a user, I can register the form with username, Emil-id and password.**	**I can register and able to access the account**	**High**	**Sprint-1**
		USN-8	**As a user, I can register with my google mail-id and password**	**I can register & get an access to use the dashboard.**	**High**	**Sprint-1**
		USN-9	As a user, I can login to the application by entering my mail and password		High	Sprint-1
		USN-10	User get into the dashboard and see's the different web pages to compute what the user needs.		High	Sprint-1
		USN-11	As a user you'll ask query or doubt about the application to the admin side. As per protocols the user will get the		Medium	Sprint-3

			response from the admin.			
Administrator Register & login page		USN-12				
	Register page	USN-12(I)	If the user is new to the application admin here to ask the user to sign up first or to fill the register the form from the user to get the user details.	If every thing is acceptable the user will access the Dashboard	High	Sprint-1
	Login page	USN-12(II)	If the user already registered the admin will get the data and user will login to application by entering email and password where the data are already stored in the database	User get access to use the Dashboard	High	Sprint-1
	Database process					
	Add food data and	USN-13	(1) Admin will store the		High	Sprint-2

	user data to the database		food nutrition value and calories value of the primary taken foods and fast foods. (2) Admin will customize a code to store the user data to the database from the registration page			
	User Interface					
	Upload the food image and get the prediction	USN-14	Here the user will upload the picture from the files to web page upload the picture, for get to know about the nutrition value. here the computation process as to predict the food image and	If the picture is clear, able to predict and goes to the next stage	High	Sprint-3

			to get the food values from the database.			
	Get the calories and nutrition value form the food item	USN -15	Admin will compute the process over cloud to get the correct food value for the predicted image that user uploaded		High	Sprint -4

6.PROJECT PLANNING & SCHEDULING:

6.1 Sprint Planning & Estimation:

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers,research publications etc.	3 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	10 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	10 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	24 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	24 SEPTEMBER 2022

Solution Architecture	Prepare solution architecture document.	01 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application.	01 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	18 OCTOBER 2022
Technology Architecture	Architecture diagram	23 OCTOBER 2022
Prepare Milestone & Activity List		10 NOVEMBER2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	IN PROGRESS..

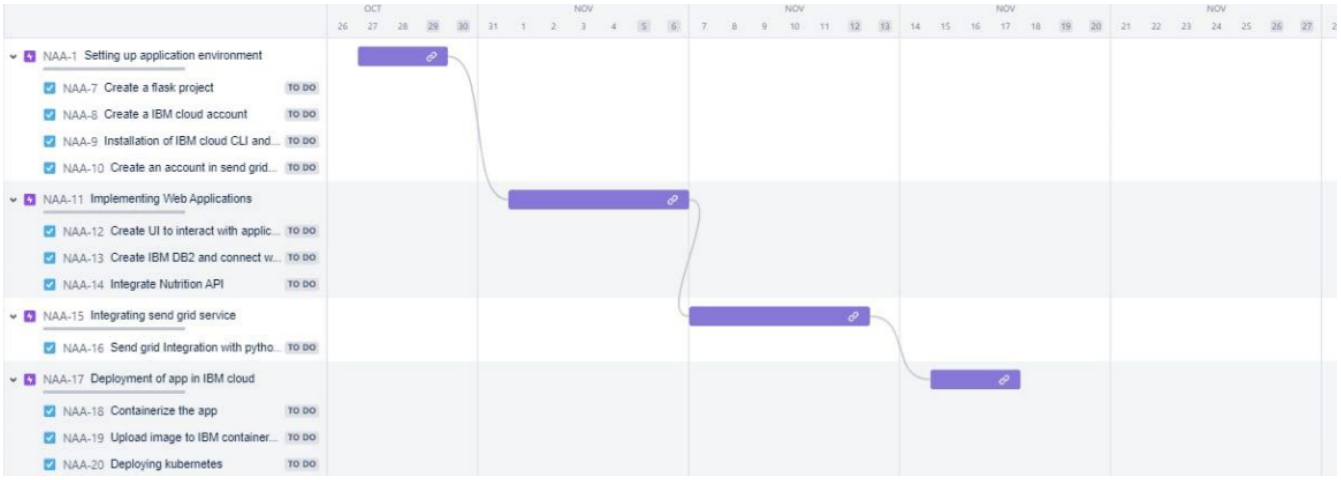
6.2 Sprint Delivery Schedule:

Sprint	Functional Requirement	User Story Number	User story/Task	Story Points	Priority	Team Members
Sprint 1	Create flask project	USN-1	We installed flask and created sample flask project.	3	Medium	S. Aiswaryaroy T. Aishwarya
Sprint 1	Create IBM Cloud account	USN-2	We sign up for an IBM cloud account	7	High	T. Aishwarya S. Aiswaryaroy S. Dheeksha na S. Amirthadevi M. Gayathri
Sprint 1	Install IBM cloud CLI and Docker CLI	USN-3	We installed IBM cloud CLI and installed	5	High	S. Dheeksha na S. Amirthadevi

	installation		Docker CLI			
Sprint 1	Create UI to interact with application	USN-4	We created account in send grid and in nutrition API	5	High	S. Aiswaryaroy M. Gayathri
Sprint 2	Create IBM DB2 and connect with python	USN-5	Create Registration page, login page, Upload image page, prediction image page for food items and view history of items	8	High	S. Aiswaryaroy T. Aishwarya
Sprint 2	Integrate Nutrition API	USN-6	Create the IBM DB2 service in IBM cloud and connect the python code with DB.	6	High	S. Dheeksha na S. Amirthadevi
Sprint 2	Send grid integration with python code	USN-7	Integrate the nutrition API to the flask with API call.	6	High	S. Aiswaryaroy S. Dheeksha na M. Gayathri
Sprint 3	Containeri ze the app	USN-8	To send emails from the application we need to integrate the send grid service.	20	High	S. Aiswaryaroy S. Dheeksha na T. Aishwarya S. Amirthadevi

						M. Gayathri
Sprint 4	Upload image to IBM container registry	USN-9	We created the Docker image for flask app	9	High	S. Dheeksha na T. Aishwarya
Sprint 4	Upload image to IBM container registry	USN-10	We uploaded the image IBM container registry	6	High	S. Aishwaryar oy S. Amirthadevi
Sprint 4	Deploy in kubernetes	USN-11	The uploaded on IBM container registry deployed the image to IBM Kubernetes Cluster.	5	Medium	S. Aiswaryaroy T. Aishwarya S. Dheeksha na

6.3 Reports from JIRA:



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

7.1 Feature 1:

LOGIN.HTML:

```
<html>
<head>
    <link rel="stylesheet" href="style.css">
</head>
<body>
<div class="container">
<div class="card">
<div class="inner-box" id="card">
<div class="card-front">
    <h2>LOGIN</h2>
<form>
    <input type="email" class="input-box" placeholder="your Email Id " required>
    <input type="password" class="input-box" placeholder="password" required>
    <button type="submit" class="submit-btn">submit</button>
    <input type="checkbox"><span>Remember Me</span>
</form>
<button type="button" class="btn" onclick="openSignup()">I'm New Here</button>
<a href="">Forget Password</a>
    </div>
    <div class="card-back">
        <h2>SIGNUP</h2>
<form>
    <input type="text" class="input-box" placeholder="Your Name" required>
    <input type="email" class="input-box" placeholder="your Email Id " required>
    <input type="password" class="input-box" placeholder="password" required>
    <button type="submit" class="submit-btn">submit</button>
    <input type="checkbox"><span>Remember Me</span>
```

```
    </form>
    <button type="button" class="btn" onclick="openLogin()">I've an account</button>
    <a href="">Forget Password</a>
        </div>
        </div>
    </div>
    </div>
    <script>
        var card = document.getElementById("card");
        function openSignup(){
            card.style.transform = "rotateY(-180deg)";
        }
            function openLogin(){
            card.style.transform = "rotateY(0deg)";
        }
    </script>
    </body>
</html>
```

STYLE.CSS:

```
*{
    margin: 0;
    padding: 0;

}
.container{
    width: 100%;
    height: 100vh;
    font-family: sans-serif;
    background-image:("https://img.freepik.com/free-photo/healthy-lunch-go-packed-lunch-box_1220-4541.jpg?w=826&t=st=1668843867~exp=166884
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-size: 100% 100%
    color: #fff;
    display: flex;
    align-items: center;
    justify-content: center;
}
.card{
    width: 350px;
    height: 500px;
    box-shadow: 0 0 40px 20px rgba(0,0,0,0.26);
    perspective: 1000px;
}
.inner-box{
    position: relative;
    width: 100%;
    height: 100%;
    transform-style: preserve-3d;
```

```
}
.card-front,.card-back{
    position: absolute;
    width: 100%;
    height: 100%;
    background-position: center;
    background-size: cover;
    background-image: linear-gradient(rgba(0,0,100,0.8),rgba(0,0,100,0.8)),url(background.png);
    padding: 55px;
    box-sizing: border-box;
    backface-visibility: hidden;
}
.card-back{
    transform: rotateY(180deg);
}
.card h2{
    font-weight: normal;
    font-size: 24px;
    text-align: center;
    margin-bottom: 20px;
}
.input-box{
    width: 100%;
    background: transparent;
    border: 1px solid #fff;
    margin: 6px 0;
    height: 32px;
    border-radius: 20px;
    padding: 0 10px;
    box-sizing: border-box;
    outline: none;
    text-align: center;
    color:  #fff;
}
:placeholder{
    color: #fff;
    font-size: 12px;
}
utton{
    width: 100%;
    background: transparent;
    border: 1px solid #fff;
    margin: 35px 0 10px;
    height: 32px;
    font-size: 12px;
    border-radius: 20px;
    padding: 0 10px;
    box-sizing: border-box;
    outline: none;
    color: #fff;
    cursor: pointer;
}
submit-btn{
    position: relative;
}
submit-btn::after{
```

```
    line-height: 32px;
    font-size: 17px;
    height: 32px;
    width: 32px;
    border-radius: 50%;
    background: #fff;
    position: absolute;
    right: -1px;
    top: -1px;
}
span{
    font-size: 13px;
    margin-left: 10px;
}
.card ,btn{
    margin-top: 70px;
}
.card a{
    color: #fff;
    text-decoration: none;
    display: block;
    text-align: center;
    font-size: 13px;
    margin-top: 8px;
}
```

REGISTRATION. HTML:

```
<!DOCTYPE html>
<html>
<head>
<title>User Details</title>
<link rel="stylesheet" href="profilestyle.css">
</head>
<body>
  <div class="container" style="color:black">
<div class="card">
    <h1><center>PROFILE</center></h1>
<label for="Username"><b>Username:</b></label>
    <input type="text" class="input-box" placeholder="Please enter your name" name="username" id="username" required>
<br>
    <label for="height"><b>Height:</b></label>
    <input type="number"  class="input-box"  placeholder="Please enter your Height in cm" name="height" id="height" required>
<br>

    <label for="weight"><b>Weight:</b></label>
    <input type="number"  class="input-box" placeholder="Please enter your Weight in kg" name="weight" id="weight" required>
```

```
20     <br>
21   <label for="Age"><b>Age:</b></label>
22     <input type="number"  class="input-box" placeholder="Please enter
   your Age" name="weight" id="weight" required>
23     <br>
24  <label for="illness"><b>Illness:</b></label>
25     <input type="text"  class="input-box" placeholder="Please enter
   your body illness" name="illness" id="illness" required>
26 <br>
27    <label for="gender"><b>Gender:</b></label><br>
28 <input type="radio" class="choice-box" id="male" name="gender"
   value="Male">
29 <label for="male">Male</label><br>
30 <input type="radio"  class="choice-box" id="female" name="gender"
   value="Female">
31 <label for="female">Female</label><br>
32 <input type="radio"  class="choice-box" id="others" name="gender"
   value="Others">
33 <label for="others">Others</label>
34 <br>
35
36 <label for="allergy"><b>Allergies(in case):</b></label>
37     <input type="text"  class="input-box" placeholder="Please enter
   your  allergy " name="allergy" id="allergy" required>
38 <br>
39     <center><button type="submit"
   class="submitbtn">Submit</button></center>
40   </div>
41   </div>
42 </body>
43 </html>
```

PROFILE STYLE.CSS:

```css
*{
    margin:0;
    padding:0;
    }
    .container{
    width:100%;
    height:100vh;
    font-family:sans-serif;
    background: rgba(187,187,245);
    color:#fff;
    display:flex;
    align-items:center;
    justify-content:center;
    }
    .card{
    width:350px;
    height:500px;
    box-shadow: 0 0 40px 20px rgba(0,0,0,0.26);
    background:rgba(187,187,245);
    }
    .input-box{
    width:95%;
    border:1px solid #fff;
    margin:6px 0;
    height:32px;
    border-radius:20px;
    padding:0 10px;
    box-sizing:border-box;
    outline:none;
    text-align:center;
    }
```

7.2 Feature 2 :

HOME PAGE.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>Document</title>
    <style>
        .row {
            background-color: gray;
        }

        h2 {
            background-color: #8f405c;
            height:80px;
            font-size: 75px;
        }

        input{

            background-color: #420e58;
            border: none;
            color: white;
            padding: 15px 32px;
            text-align: center;
            text-decoration: none;
            display: inline-block;
            font-size: 16px;
            border-radius: 12px;
            margin-left: 60px;
           margin-top: -30px;
        }
        img{
```

```
            border: 2px solid red;
            margin-top: -500px;
            margin-left: 10px;
        }
        .about{
            width: 300px;
           border: 15px solid green;
           padding: 50px;
           margin: 20px;
           margin-left:700px;

        }
    </style>
</head>

<body>
    <div class="container">
        <div class="row">
            <div class="col-lg-8  offset-lg-2">

                <h2 class="mt-5">Nutritions Assistant</h2>
                <div id="myform">
                    <form method="post" action="{{ url_for('tasks')
}}">
                        <input type="submit" value="Stop/Start"
name="stop" />
                        <input type="submit" value="Capture"
name="click" />
                        <input type="submit" value="Detect"
name="detect" />
                    </form>
                </div>
                <div class="about">By identifying the supplied food
image, this project attempts to create a web application that
automatically calculates food qualities like ingredients and
nutritional value. For precise food recognition and to determine the
nutritional value of the recognized item, our solution uses a food
detection model and food <APIs class="br">
                    <ol>
                        <li>First click on stop and start for starting
```

```
    </li>
66                          <li>then click on capture to take a image
    which nutrients you want to know </li>
67                          <li>click on detect button for see the
    nutrients list</li>
68                          <br>
69                          <br>
70                          <li>The health issues most of the time depends
    on our diet and nutrition</li>
71                        </ol>
72                  </APIs></div>
73                  <img src="{{ url_for('video_feed') }}" height="50%">
74
75
76              </div>
77          </div>
78      </div>
79 </body>
80
81 </html>
```

8.TESTING:

8.1 Test Cases :

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
Uploadimage_TC_001	Functional	Upload Image page	Verify whether we get the nutrition values for food image uploaded		1.After uploading the image 2.Click in Submit button		The nutrition value of the food image uploaded will be displayed	Working as expected	Pass				Aishwarya T
Clarify API_TC_002	Functional	Upload Image	When the user uploads the image the Clarify API will identify the food		1.After uploading the image 2.Click in Submit button		The food image should be interdified by clarify api and display	Working as expected	Pass				S.AiswaryaRoy
Personaldetailsdatabase_TC_002	Functional	Personal details	Verify whether personal details credentials are added to the cloud database		1.Verify personal details page with below UI elements: a.Enter Age b.Enter Height c.Enter Weight d.Choose daily activity dropdown e.Click on proceed to dashboard button		The given credentials by the user should be same as the credentials stored in ibm cloud database	Working as expected	Pass				S.Dheekshana
Track_historydatabase_TC_003	Functional	Track history	Verify whether added food details are added to the cloud database		1.Date picker text box 2.Food name text box 3.Calorie text box 4.Add button		The added food details by the user should be same as the details stored in ibm cloud database	Working as expected	Pass				S.AmirthaDevi

8.2 User Acceptance Testing:

Purpose of Document:

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

Defect Analysis :

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

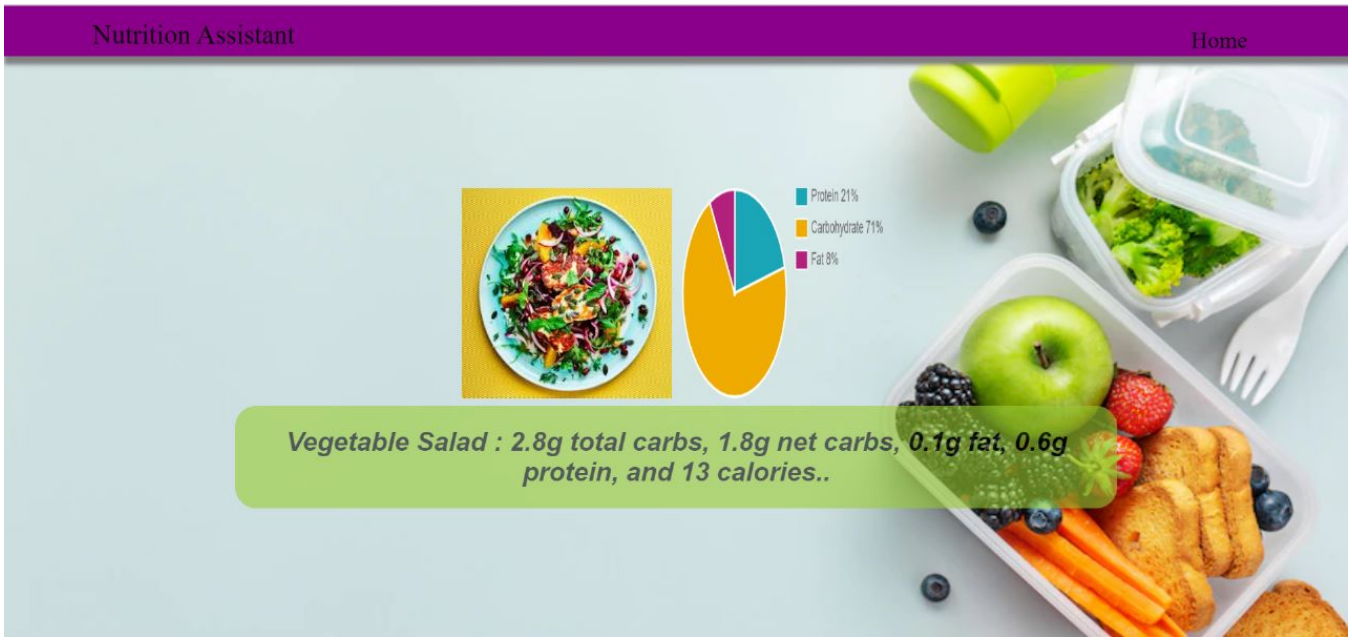
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	`1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

Test Case Analysis:

This report 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:



10.ADVANTAGES & DISADVANTAGES:

Advantages:-

1. This app is user friendly.
2. Its only required the image of the food .
3. To know the different type of nutrients present in food .
4. And also know that how much composition of the nutrients are present.
5. Output of the screen is easy understandable.

Disadvantage:-

1. This device is not able to predict the multiple image as input.
2. The internet is only necessary for opening the web application(After converting the mobile app internet is not necessary for opening).

11.CONCLUSION:

During this assignment we were able to take a closer look at our daily eating habits.From here we can now improve our application so that we can help clients to eat andgrow healthier as a person and athlete.I can truly say that I learnt a lot from this assignment. I was able to point out changes I needed to make and how to move forward and make it work in my life.I am now more educated on the powers of food and how they control our body. I hopethat people will use our application to lead a healthy life. When choosing the right foodsfor yourself you should be focused on what is the healthiest choice. Eating healthy andfeeling good go hand in hand, eating better will automatically give you a better functioning body. Eating healthy means eating a variety of foods that meet your daily requirements.We would recommend our application to anyone who is interested in eating healthy. Not only is it easy to use, but it is a great way to evaluate what you are eating andunderstand the vitamins and minerals that you need.

12.FUTURE SCOPE:

This application will also assist you determine the quantity and degree of flavour of the food.Future goals include increasing the accuracy of our machine learning model and expanding the types of food categories so that we can better meet user needs.We are also increasing dataset of categories of images and nutrition to better efficiency to get output.Our research essentially identifies simply the nutrients, but our team members raise the bar for our project so that we also understand the ingredients and the amount of nutrients in a particular cuisine.

13.APPENDIX:

```
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
```

```
12 import google.auth.transport.requests
13
14 # Configure Flask app
15 app = Flask(_name_)
16 SECRET_KEY = os.urandom(32)
17 app.config['SECRET_KEY'] = SECRET_KEY
18
19 # Load .env file
20 load_dotenv()
21
22
23 connection_string = "DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-
   10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304
   ;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=wpq726
   44;PWD=CTsaqSf91OIzLniF",'','''.format(DATABASE_NAME, HOSTNAME,
   PORT_NUMBER, USERNAME, PASSWORD)
24 conn = db.connect(connection_string, "", "")
25
26 # Frequently used variables
27 SIGN_UP_PAGE_URL = '/'
28 LOG_IN_PAGE_URL = '/login'
29 HOME_PAGE_URL = '/home'
30 GOOGLE_LOGIN_PAGE_URL = '/google_login'
31 PROFILE_PAGE_URL = '/profile'
32 CHANGE_PASSWORD_URL = '/changepwd'
33
34 # Google Auth Configuration
35 os.environ["OAUTHLIB_INSECURE_TRANSPORT"] = "1"
36
37 client_secrets_file = os.path.join(pathlib.Path(_file_).parent,
   "client_secret.json")
38
39 flow = Flow.from_client_secrets_file(
40     client_secrets_file=client_secrets_file,
41     scopes=["https://www.googleapis.com/auth/userinfo.profile",
   "https://www.googleapis.com/auth/userinfo.email", "openid"],
42     redirect_uri="http://127.0.0.1:5000/callback"
43 )
44
45 # Helper Function to execute SQL queries
```

```
46 def execute_sql(statement, **params):
47     global conn
48     stmt = db.prepare(conn, statement)
49     param_id = 1
50     for key, val in params.items():
51         db.bind_param(stmt, param_id, val)
52         param_id += 1
53     result = ''
54     try:
55         db.execute(stmt)
56         result = db.fetch_assoc(stmt)
57     except:
58         pass
59     return result
60 
61 # Creates user table if not exists
62 create_table = "CREATE TABLE IF NOT EXISTS user(email varchar(30),
   username varchar(30), password varchar(30))"
63 execute_sql(statement=create_table)
64 
65 # Helper function to send confirmation mail on sign in
66 def send_confirmation_mail(user, email):
67     message = Mail(
68         from_email="nutrition@gmail.com",
69         to_emails=email,
70         subject="YAYY!! Your Account was created successfully!",
71         html_content= "<strong>Account Created with username
   {0}</strong>".format(user)
72     )
73     try:
74         sg = SendGridAPIClient(os.environ.get('xxxxxxxxxxxxxxxxx'))
75         response = sg.send(message)
76         print(response.status_code)
77         print(response.body)
78         print(response.headers)
79     except Exception as e:
80         print(e)
81 
82 # Sign up page
83 @app.route(SIGN_UP_PAGE_URL, methods=['GET', 'POST'])
```

```
84 def signup():
85     msg = ''
86     if session.get('user'):
87         return redirect(HOME_PAGE_URL)
88 
89     if request.method == 'POST':
90         user = request.form['user']
91         email = request.form['email']
92         password = request.form['password']
93         duplicate_check = "SELECT * FROM user WHERE username=?"
94         account = execute_sql(statement=duplicate_check, user=user)
95         if account:
96             msg = "There is already an account with this username!"
97         else:
98             insert_query = "INSERT INTO user values(?, ?, ?)"
99             execute_sql(statement=insert_query, email=email,
   user=user, password=password)
100 
101                     send_confirmation_mail(user, email)
102                     return redirect(LOG_IN_PAGE_URL)
103             return render_template('Registrationform.html', msg=msg)
104 
105         # Login page
106         @app.route(LOG_IN_PAGE_URL, methods=['GET', 'POST'])
107         def login():
108             msg = ''
109             if session.get('user'):
110                 return redirect(HOME_PAGE_URL)
111 
112             if request.method == "POST":
113 
114                 user = request.form['user']
115                 password = request.form['password']
116                 duplicate_check = "SELECT * FROM user WHERE
   username=?"
117                 account = execute_sql(statement=duplicate_check,
   user=user)
118 
119                 print(account)
120                 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)
```

```
159
160         # Home page
161         @app.route(HOME_PAGE_URL, methods=['GET', 'POST'])
162         def homepage():
163             if not session.get('user'):
164                 return redirect(LOG_IN_PAGE_URL)
165 
166             msg = ''
167             if request.method == 'POST':
168                 if request.form['food']:
169                     msg = 'Image Uploaded Successfully!'
170                 else:
171                     msg = "Image wasn't uploaded, Try again!"
172             return render_template('front.html',
    user=session.get('user'), msg=msg)
173 
174         # Profile page
175         @app.route(PROFILE_PAGE_URL, methods=['GET', 'POST'])
176         def profile():
177             if not session.get('user'):
178                 return redirect(LOG_IN_PAGE_URL)
179             sqlst = "select email from user where username=?"
180             user = session.get('user')
181             email = execute_sql(statement=sqlst, user=user)
182             return render_template('front.html', user=user,
    email=email['EMAIL'])
183 
184         #change password
185         @app.route(CHANGE_PASSWORD_URL, methods=['GET', 'POST'])
186         def changepwd():
187             if not session.get('user'):
188                 return redirect(LOG_IN_PAGE_URL)
189 
190             msg = ''
191             user = ''
192             email = ''
193             if request.method == 'POST':
194                 user = session.get('user')
195                 oldpass = request.form['oldpass']
196                 newpass = request.form['newpass']
```

```
197
198                 sqlst = 'SELECT password from user where username = ?'
199                 dbpass = execute_sql(statement = sqlst , username =
   user)['PASSWORD']
200                 sqlst = 'SELECT email from user where username = ?'
201                 email = execute_sql(statement = sqlst ,username =
   user)['EMAIL']
202
203                 if dbpass == oldpass:
204                     sqlst = 'UPDATE user SET password = ? where
   username = ?'
205                     execute_sql(statement = sqlst , password = newpass
   , username = user)
206                     msg = 'Updated Successfully!'
207                 else:
208                     msg = 'Old Password Incorrect!'
209                 return render_template('front.html', user=user,
   email=email, msg=msg)
210
211             return render_template('passwordChange.html')
212
213
214         # Logout user
215         @app.route('/logout')
216         def logout():
217             session['user'] = ''
218             return redirect(LOG_IN_PAGE_URL)
219
220         # Delete user account
221         @app.route('/delete')
222         def delete():
223             if not session.get('user'):
224                 return redirect(LOG_IN_PAGE_URL)
225
226             user = session['user']
227             delete_query = "DELETE FROM user WHERE username=?"
228             execute_sql(statement=delete_query, user=user)
229             session.clear()
230             return redirect(SIGN_UP_PAGE_URL)
231
```

```
232         # Run the application
233         if _name_ == '_main_':
234             app.run(debug=True)
```

GitHub & Project Demo Link:

https://youtu.be/DFZs_sJrxfk