NALAIYA THIRAN PROJECT REPORT

NUTRITION ASSISTANT APPLICATION

IBM-Project-166571659619614

TEAM ID: PNT2022TMID39029

Submitted by

K.RAJESH (422119104020)

B.DHIVAGAR (422119104005)

P.A.VASANTH (422119104030)

P.A.YOKESH (422119104031)

K.TAMIL OLI (422119104029)

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

ST.ANNE'S COLLEGE OF ENGINEERING & TECHNOLOGY

ANGUCHETTYPALAYAM, PANRUTI, CUDDALORE.

INDEX

NUTRITION ASSISTANT APPLICATION

1. 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 and provide it's nutritional values. Our method employs "**Clarifai's AI-Driven Food Detection Model**" for accurate food identification and "**Spoonacular Nutrition API**" to give the nutritional value of the identified food. Clarifai AI-Driven Food Detection Model is an API that classifies the ingredients of the meal and provide the name of the meal. That name will be provided as an input to the Spoonacular API which provides the nutritional value of the identified food.

1.2 Purpose

Basically, a diet and nutrition app comes with lots of benefits. It helps users in:

- To keep track of daily intake
- To monitor calories intake
- To provide facility to upload meal image
- To get nutritional value of the uploaded image
- To keep track of BMI

2. LITERATURE SURVEY

2.1 Existing solutions

MY FITNESS PAL [1]

Mobile applications that count calories, such as My Fitness Pal, are frequently employed on a daily basis. Recent research has shown that in undergraduates, calorie tracking is associated with eating disorder pathology. In the current study (N = 105 individuals diagnosed with an eating disorder), we assessed usage of My Fitness Pal to track calories. We also assessed perceptions that My Fitness Pal contributed to eating disorder symptoms and if these perceptions were associated with eating disorder symptoms. We found that a substantial percentage (~ 75%) of participants used My Fitness Pal and that 73% of these users perceived the app as contributing to their eating disorder. Furthermore, we found that these perceptions were correlated with eating disorder symptoms.

This research suggests that My Fitness Pal is widely used in an eating disorder population and is perceived as contributing to eating disorder symptoms. Further research is needed to clarify the role calorie tracking applications play within a sample of individuals with eating disorders. It can be tailored to fit the needs of anyone with specific and/or doctor/dietitian-recommended requirements. MyFitnessPal's community aspect basically consists of a forum, where other fellow users of the app are free to exchange tips and advice, as well as to create relationships through sharing personal experiences or struggles.

MyFitnessPal has more than 350 exercise stored in its database, and it shows how much each person burns during each activity, based on their specific height, weight, and gender. It includes most cardio and strength training workouts, as well as yoga and Pilates. The free application is available for Blackberry, Android, Windows, and the iPhone.

NUTRI ASSISTANT [2]

NutriAssistant dedicates itself to the development of software in the cloud, with connection to web and mobile applications, for Professionals and any other Users. NutriAssistant is a work tool that is designed to be used by a trained nutrition Professionals. That software, NutriAssistant, enables the simplification of complex tasks, such as management and analysis of nutritional information of the Professions' clients. We consider 46 macro and micro nutrients as well as allergies and religious preferences. With NutriAssistant the Customer can plan, analyze and create professional grade weekly meal plans. NutriAssistant can generate meal plans and recipes automatically, based on parameters submitted by the User However, NutriAssistant cannot provide medical or health advice, dietary recommendations, or produce diagnosis for any patients. It is the responsibility of the User that the use of NutriAssistant meet the local dietary, health and other regulations. With NutriAssistant the Customer can plan, analyze and create professional grade weekly meal plans. If the User generates any kind of nutrition-related document in NutriAssistant, including meal plans, recipes, reports, and food analysis. NutriAssistant is not liable for any adverse effects, issues, or complications any patient or client may experience after implementing a meal plan or recipe created by using NutriAssistant in their lifestyle. NutriAssistant will endeavor to keep all nutritional facts, including nutrient data, recipe and diet information as accurate and up-to-date as possible. NutriAssistant will not be responsible for generating any meal plans or recipes that contain allergens the User wishes to avoid. Recipes and Menus that are created or amended by the User are owned by the Company.

PERSONALIZED DIETARY ASSISTANT [3]

As the Internet gains dominance as the primary source of information in the daily life of people, it is naturally among the first places one would start looking for such information, although numerous online sources have been shown to lack accuracy considering dietary guidelines. Nowadays, there are numerous types of diets that aim to improve the quality of life, health and longevity of people. However, these diets typically involve a strictly planned regime, which can be hard to get used to or even to follow

through at all, due to the sudden nature of the change. In this paper, the framework for an Intelligent Space application is proposed that helps its users to achieve a healthier diet in the long term by introducing small, gradual changes into their consumption habits. The application observes the daily nutrition intake of its users, applies data mining in order to learn their personal tastes, and educates them about the effects of their current diet on their health. Then it analyzes the knowledge base to find different food or drink items that align with the perceived preferences, while also add to the balance of the daily nutrition of the users considering their physical properties, activities, and health conditions (e.g. diabetes, celiac disease, food allergies, etc). Finally, the system uses the findings to make suggestions about adding items from the consumption list, or change one item to another.

POPULAR NUTRITION-RELATED MOBILE APPS A FEATURE ASSESSMENT[4]

A key challenge in human nutrition is the assessment of usual food intake. This is of particular interest given recent proposals of eHealth personalized interventions. The adoption of mobile phones has created an opportunity for assessing and improving nutrient intake as they can be used for digitalizing dietary assessments and providing feedback. This study aims to analyze the main features of the most popular nutrition apps and to compare their strategies and technologies for dietary assessment and user feedback. Apps were selected from the Google Play Store for Android and the iTunes App Store for iOS based on popularity as measured by the number of installs and reviews. Results - A total of 13 apps were classified as popular for inclusion in the analysis. Nine apps offered prospective recording of food intake using a food diary feature. Food selection was available via text search or barcode scanner technologies. Portion size selection was only textual (i.e, without images or icons). All nine of these apps were also capable of collecting physical activity (PA) information using self-report, the global positioning system (GPS), or wearable integrations. Their outputs focused predominantly on energy balance between dietary intake and PA. None of these nine apps offered features directly related to diet plans and motivational coaching. In contrast, the remaining four of the 13 apps focused on these opportunities, but without food diaries.

Conclusions: The high number of installs indicates that there is a clear interest and opportunity for diet monitoring and recommendation using mobile apps. All the apps collecting dietary intake used the same nutrition assessment method (ie, food diary record) and technologies for data input (ie, text search and barcode scanner). Emerging technologies, such as image recognition, natural language processing, and artificial intelligence, were not identified.

NUTRITRACK: ANDROID-BASED FOOD RECOGNITION APP FOR NUTRITION AWARENESS [5]

The use of smartphone technology has created new opportunities for people to be aware about health and wellness using diet monitoring applications. Proliferation of such applications have been manifested in the society and that using a smartphone and mobile technology nowadays become universal. One of the emergent concerns of human life is about health and wellness. Undeniably, health and nutrition are one of the valuable aspects of life. Thus, technological innovations to help enhance and even promote health awareness is essential. With the advent of mobile computing, it is much easier to be aware of health information because of its mobility and availability. Much mobile application is being developed to serve as a tool for health monitoring and nutritional guide. Mobile applications have the ability to support health needs like detecting heart rate, classifying food, and many more. Taking advantage of technology, utilization of it hereby addresses certain issue and problems of human life, especially in health. In this study, the researcher's attempts to design and develop an Android-based food recognition application that could be used as a health awareness tool for non-health conscious individual. The application lets the user take the photo of the food and show its nutritional contents. Implementing Mifflin-St Jeor method in determining daily calorie consumption, users shall be aware of their required calorie intake. Moreover, the researchers' have studied its effect on people's health awareness on food nutrition by the randomly selected respondents. Finally, this paper presents an analysis of the impact of the food recognition app to change people's concept of food nutrition.

2.2 References

[1] https://www.myfitnesspal.com/

[2] https://www.nutriassistant.com/

[3] B. Tusor, G. Simon-Nagy, J. T. Tóth and A. R. Várkonyi-Kóczy, "Personalized dietary assistant — An intelligent space application," 2017 IEEE 21st International Conference on Intelligent Engineering Systems (INES), 2017, pp. 000027-000032, doi: 10.1109/INES.2017.8118575.

[4] Zenun Franco, Rodrigo & Fallaize, Rosalind & Lovegrove, Julie & Hwang, Faustina. (2016). Popular Nutrition-Related Mobile Apps: A Feature Assessment. JMIR mhealth and uhealth. 4. 10.2196/mhealth.5846

.

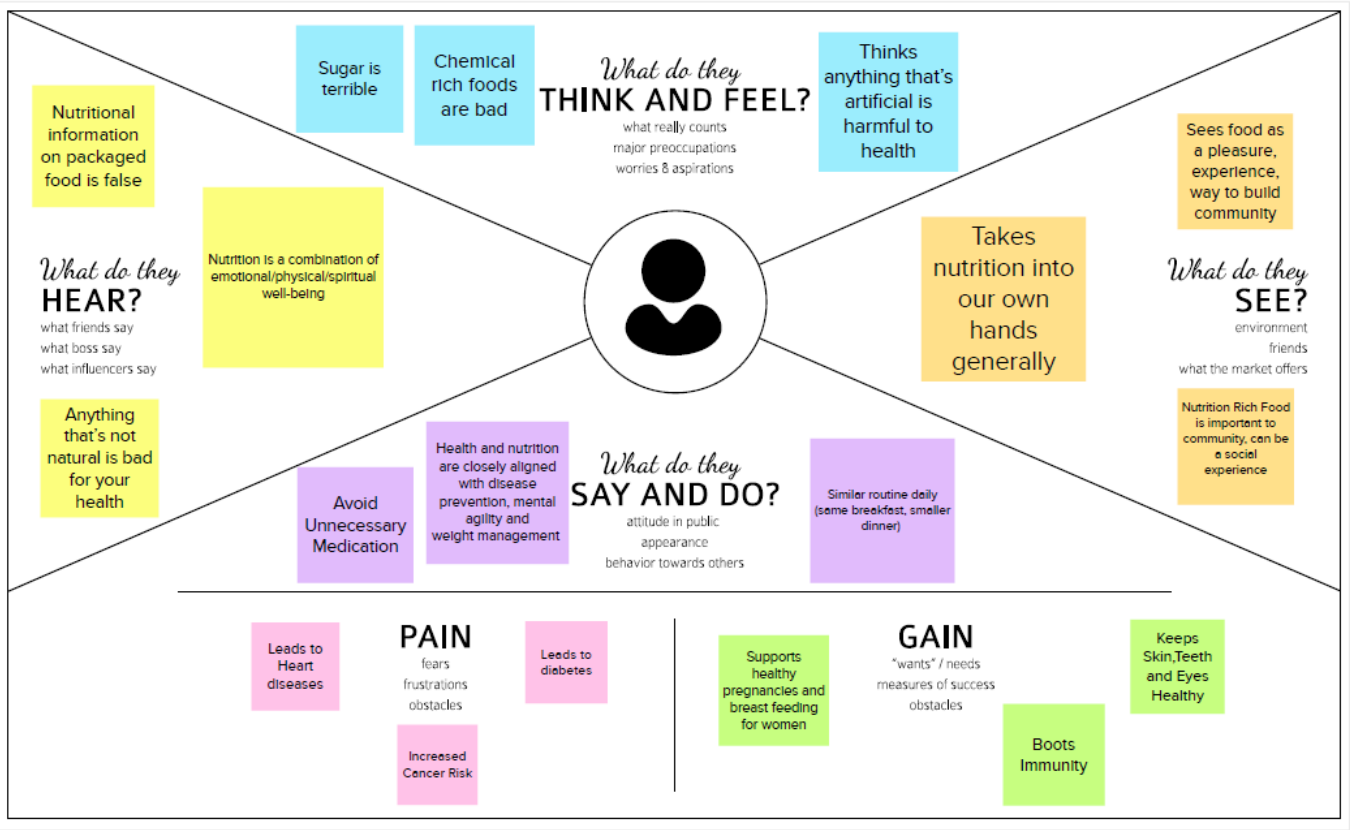
[5] A. B. Ocay, J. M. Fernandez and T. D. Palaoag, "NutriTrack: Android-based food recognition app for nutrition awareness," 2017 3rd IEEE International Conference on Computer and Communications (ICCC), 2017, pp. 2099-2104, doi: 10.1109/CompComm.2017.8322907.

2.3 Problem Statement Definition

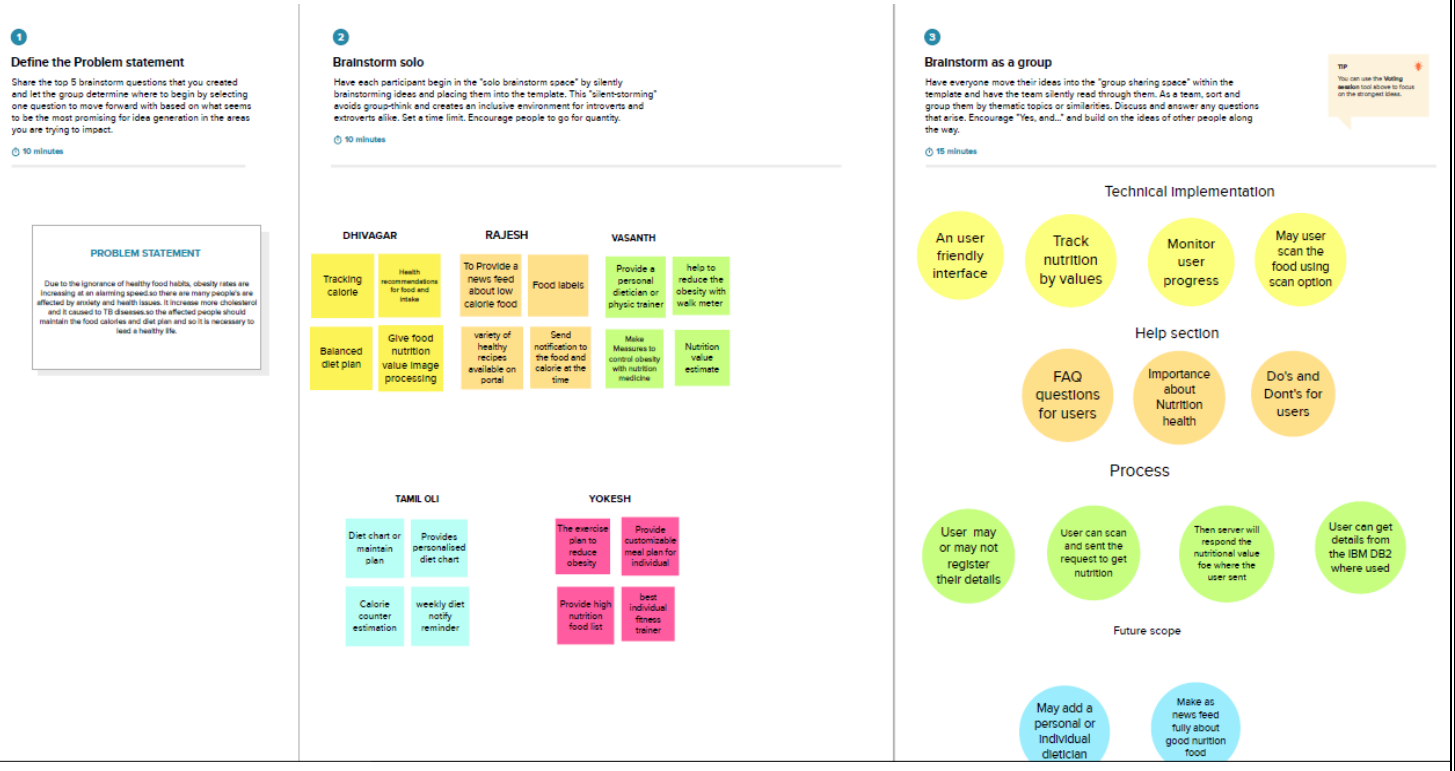
- People need to control their daily calorie intake by eating healthier foods, which is the most basic method to avoid obesity. However, although food packaging comes with nutrition (and calorie) labels, it is still not very convenient for people to refer.
- App-based nutrient dashboard systems which can analyze real time images of meal and analyze it for nutritional content can be very handy and improve the dietary habit.
- To develop a web application that automatically estimates food attributes such as ingredients and nutritional value by classifying the input image of food provided.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

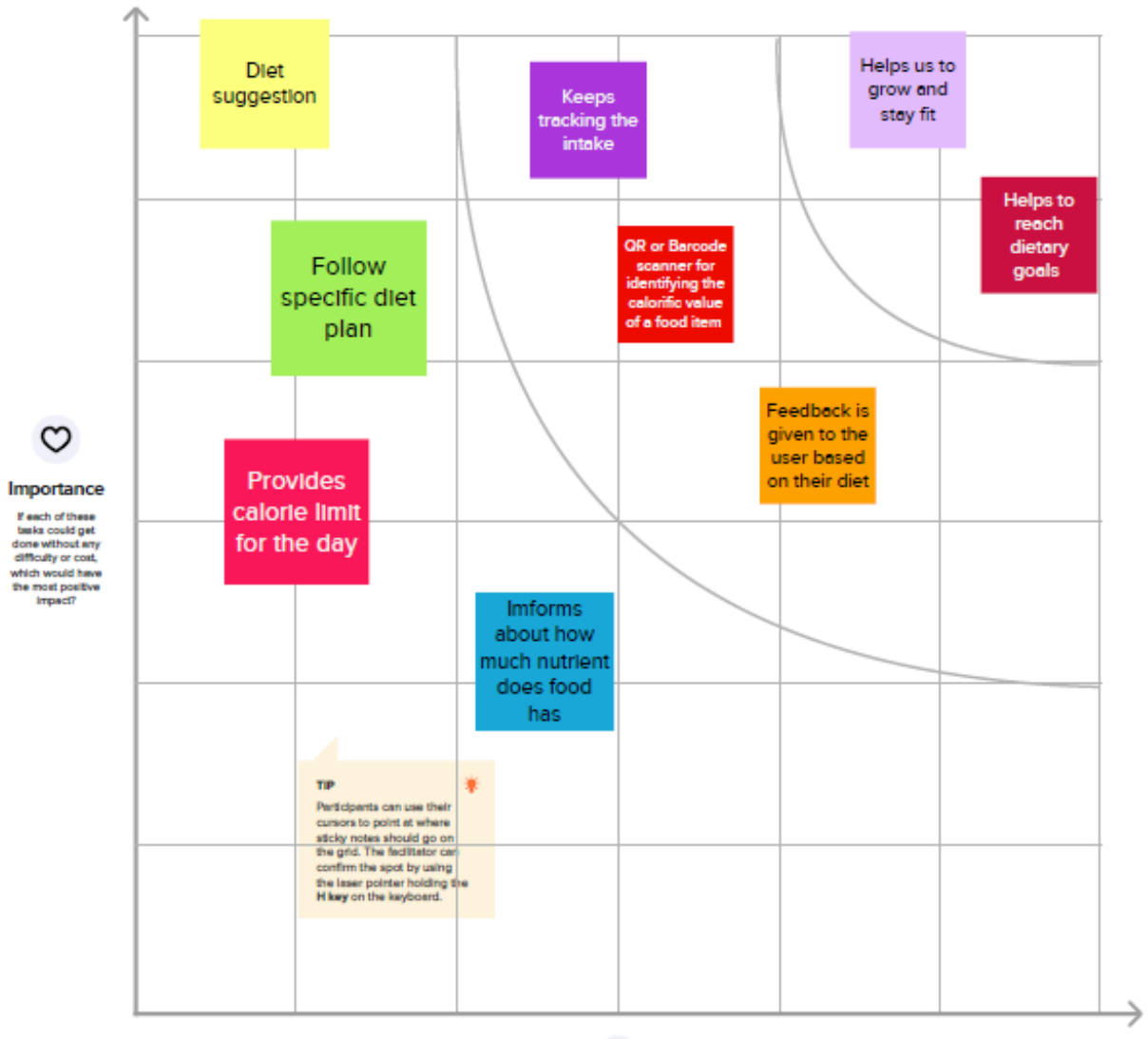


4

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.

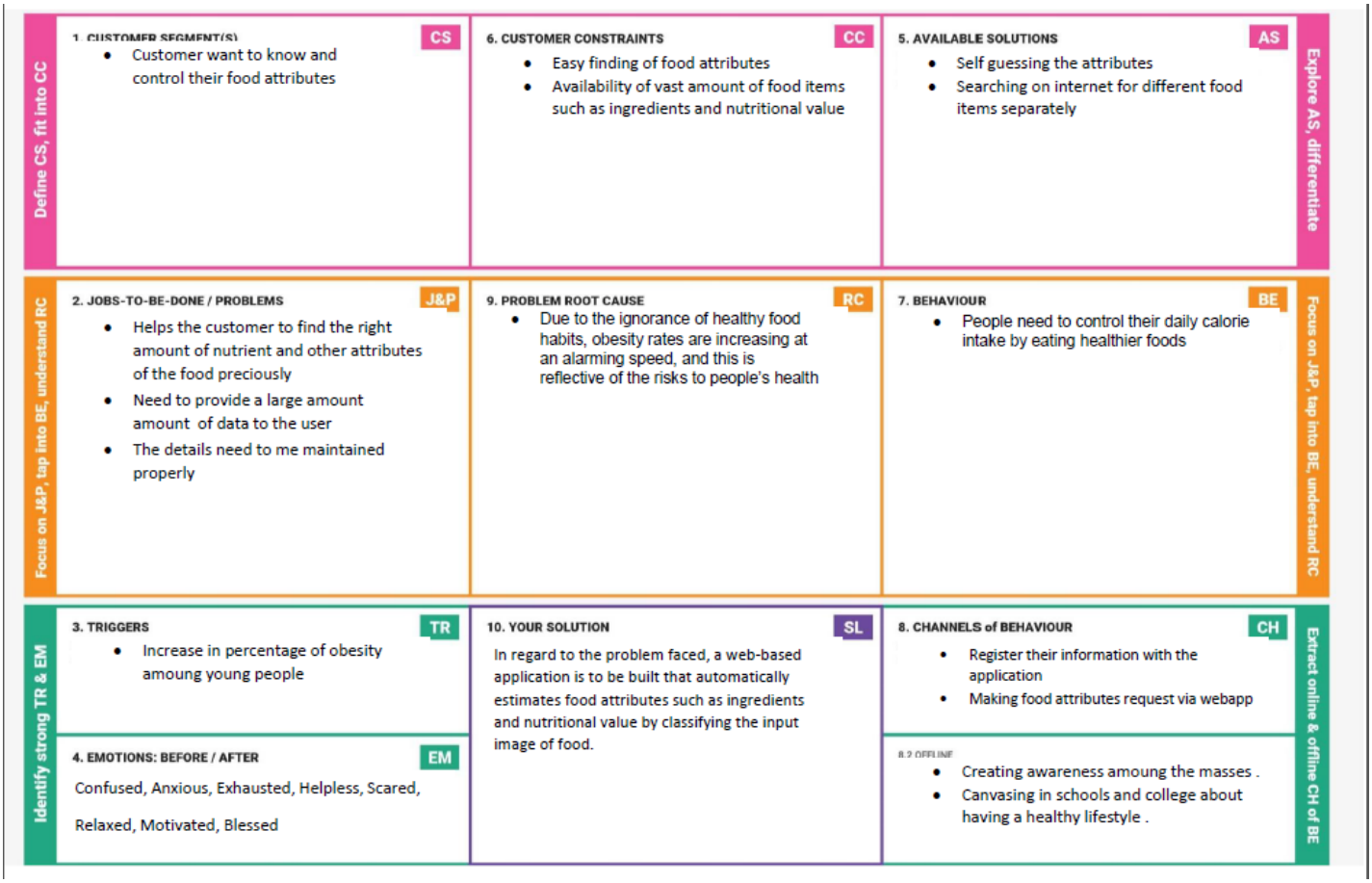
20 minutes



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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.
2.	Idea / Solution description	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
3.	Novelty / Uniqueness	This application can able to perform certain functionality and possess certain feature which are unique. Those are listed below: • The accuracy of provided data in the web application is high • Support to a large number of food variety • User interface Is simple so that the end users can use without any trouble
4.	Social Impact / Customer Satisfaction	By using the application one can easily able to find the nutritional value like macro and micro presents in a food item which make it easy for the user to calculate their required nutrient for daily diet plan. Through this we can build a healthy society of tomorrow.
5.	Business Model (Revenue Model)	We can provide some additional services in order to generate some revenue. Services like giving a diet plan which suits for the user, adding subscription and some other service.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

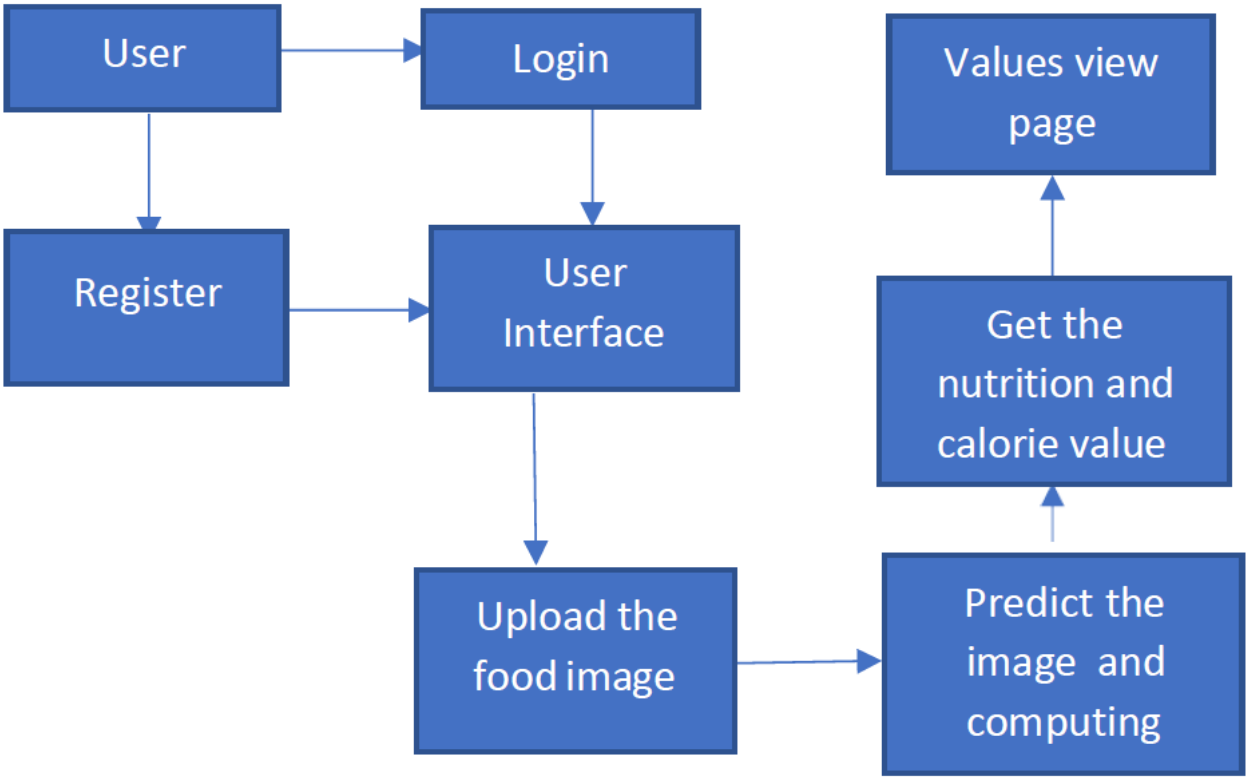
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through email and phone number
FR-2	User Confirmation	Confirmation via Email
FR-3	User Profile Completion	Get personal details like height, weight, etc.
FR-4	Gather meal image	Upload photo Take live photo of the meal
FR-5	Display calorie information	Integrate Clarifai API to get name of the food Integrate Nutrition API (rapid API) to collect calorie information

4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Provide user friendly UI Simple and intuitive design
NFR-2	**Security**	Comprehensive authorization and authentication scheme for each system actor
NFR-3	**Reliability**	The system must perform without failure in 95 percent of use cases
NFR-4	**Performance**	The landing page supporting several users must provide 5 second or less response time
NFR-5	**Availability**	Uninterrupted services must be available all time except the time of server updation.
NFR-6	**Scalability**	Provide horizontal or vertical scaling for higher workloads.

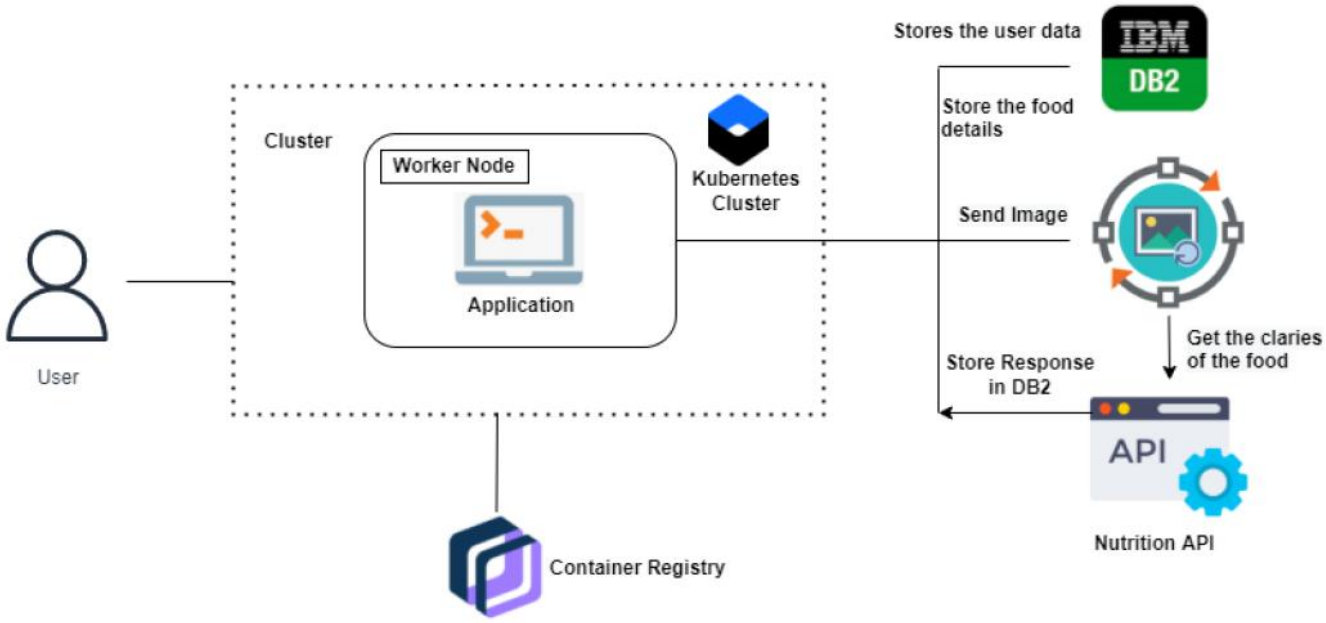
5. PROJECT DESIGN

5.1 Data Flow Diagrams

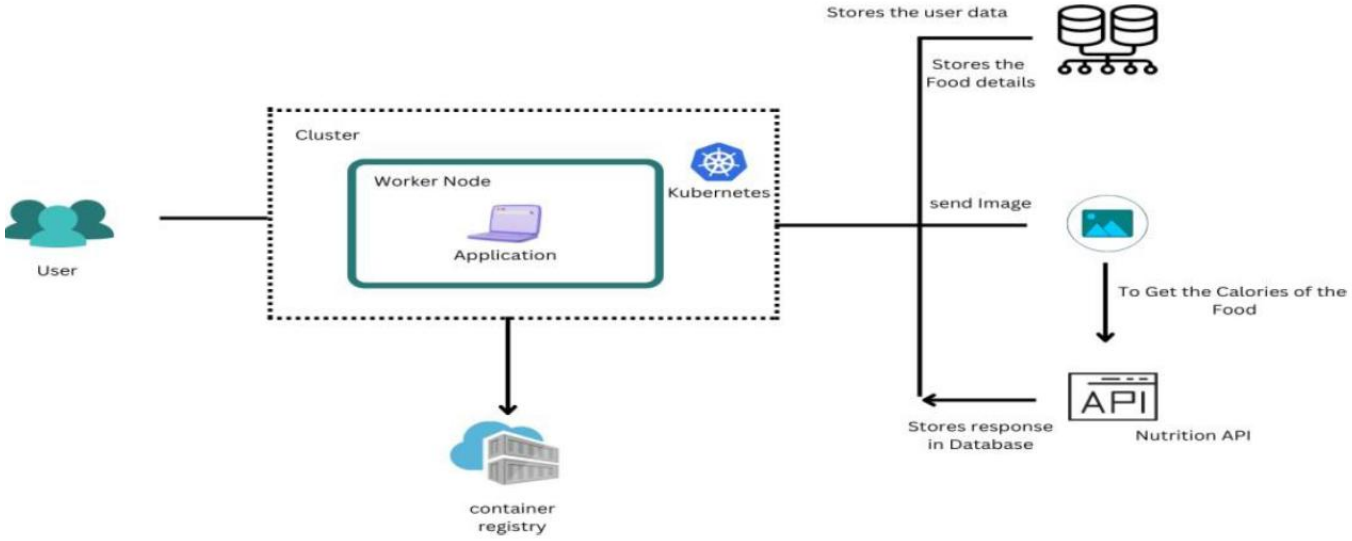


5.2 Solution & Technical Architecture

5.2.1 Solution Architecture



5.2.2 Technical Architecture



5.3 User Stories

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 confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		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	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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		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
	Login	USN-9	As a user, I can login to the application by entering my mail and password		High	Sprint-1
	Dashboard	USN-10	User get into the dashboard and see's the different web pages to compute what the user needs.		High	Sprint-1
Customer Care Executive	FAQ	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 response from the admin.		Medium	Sprint-3
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

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user ,I can register for the application by entering my email, password and conforming my password	10	High	Rajesh K Dhivagar B Vasanth P.A Yokesh P.A Tamiloli K
Sprint-1	Login	USN-2	As a user,I can log into the application by entering email and password	10	High	Rajesh K Dhivagar B Vasanth P.A Yokesh P.A Tamiloli K

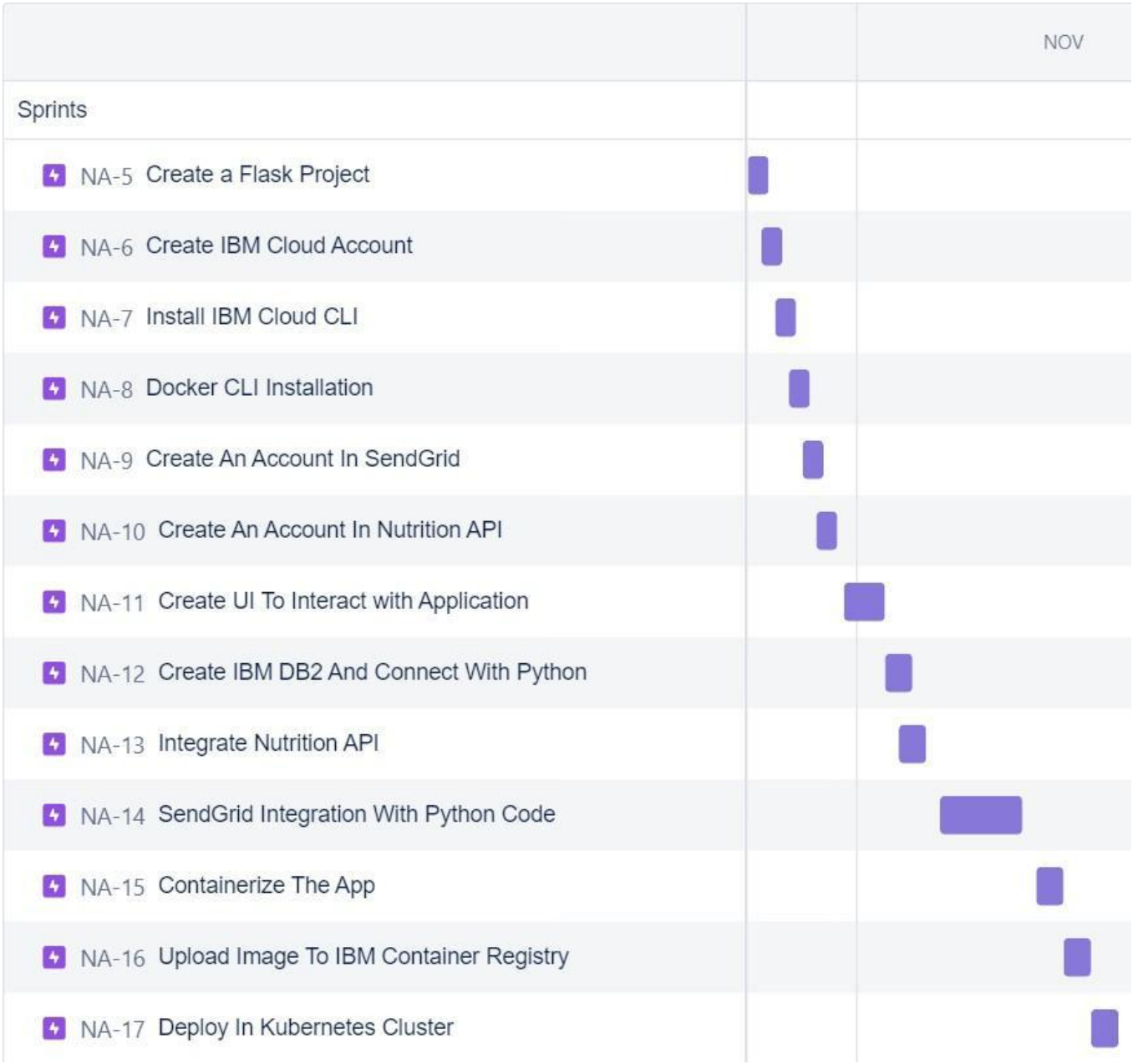
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Implementing Web Application	USN-3	We create a UI to interact with application. Create database system DB2 and connect it with python and integrate with Nutrition API.	20	High	Rajesh K Dhivagar B Vasanth P.A Yokesh P.A Tamiloli K
Sprint-3	Integrating SendGrid Service	USN-4	SendGrid integration with python code for include some RestAPI services for to give a Nutrition and calorie value.	20	High	Rajesh K Dhivagar B Vasanth P.A Yokesh P.A Tamiloli K
Sprint-4	Deployment of App in IBM Cloud	USN-5	In the deploy process, the deployment in Kubernetes cluster is the major task before that we need to containerize the app and upload image to IBM container Registry	20	High	Rajesh K Dhivagar B Vasanth P.A Yokesh P.A Tamiloli K

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

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-4	20	6 Days	14 Nov 2022	19 Nov 2022		19 Nov 2022

6.3 Reports from JIRA



7. CODING & SOLUTIONING

7.1 Feature 1

Home page:

Nutrition Assistant Application is a responsive web application which gives the nutritional value of the uploaded meal image. The home page contains information about Nutrition Assistant App. It also contains navigation bar with links to Register, Login and Support pages.

Home.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link
   rel="stylesheet"
   href="static/home-style.css"
  />
  <title>HOME</title>
  <script src="https://kit.fontawesome.com/a076d05399.js"></script>
 </head>
 <body>
  <header>
   <img
    class="logo"
    src="static/applogo.png"
    alt="logo"
   />
   <nav>
    <input type="checkbox" id="check" />
    <label for="check" class="checkbtn">
     <i class="fas fa-bars"></i>
    </label>
    <label class="nutri">Nutrition Assistant Application</label>
    <ul>
     <li><a href="{{url_for('reg')}}">Register</a></li>
     <li><a href="{{url_for('login')}}">Login</a></li>
     <li><a href="{{url_for('support')}}">Support</a></li>
    </ul>
   </nav>
  </header>
```

```
        <div class="bg-text">
          <p>
            <span>Nutrition Assistant Application</span> is a web app that aims at automatically
      estimating
            the food attributes such as ingredients and nutritional value by
            classifing the input image of the food. The person who wishes to use the
            app must register before using the app. The user can login to their
            account to see thier respective dashboard with the details of height,
            weight, BMI that is automatically generated and the amount of calorie
            that can be taken according to the BMI of the user. The user can upload
            the image of the food that he has taken. The app will then display the
            nutritional value of the food. The user can take a note of it and can
            alter his food habits accoring to his BMI and amount of calories that
            can be intaken. The user can view the history of the food and the
            calorie intake details in the histroy page. The user can make use of the
            support in case of any queries.
          </p>
        </div>
        <section></section>
      </body>
    </html>
```

home-style.css

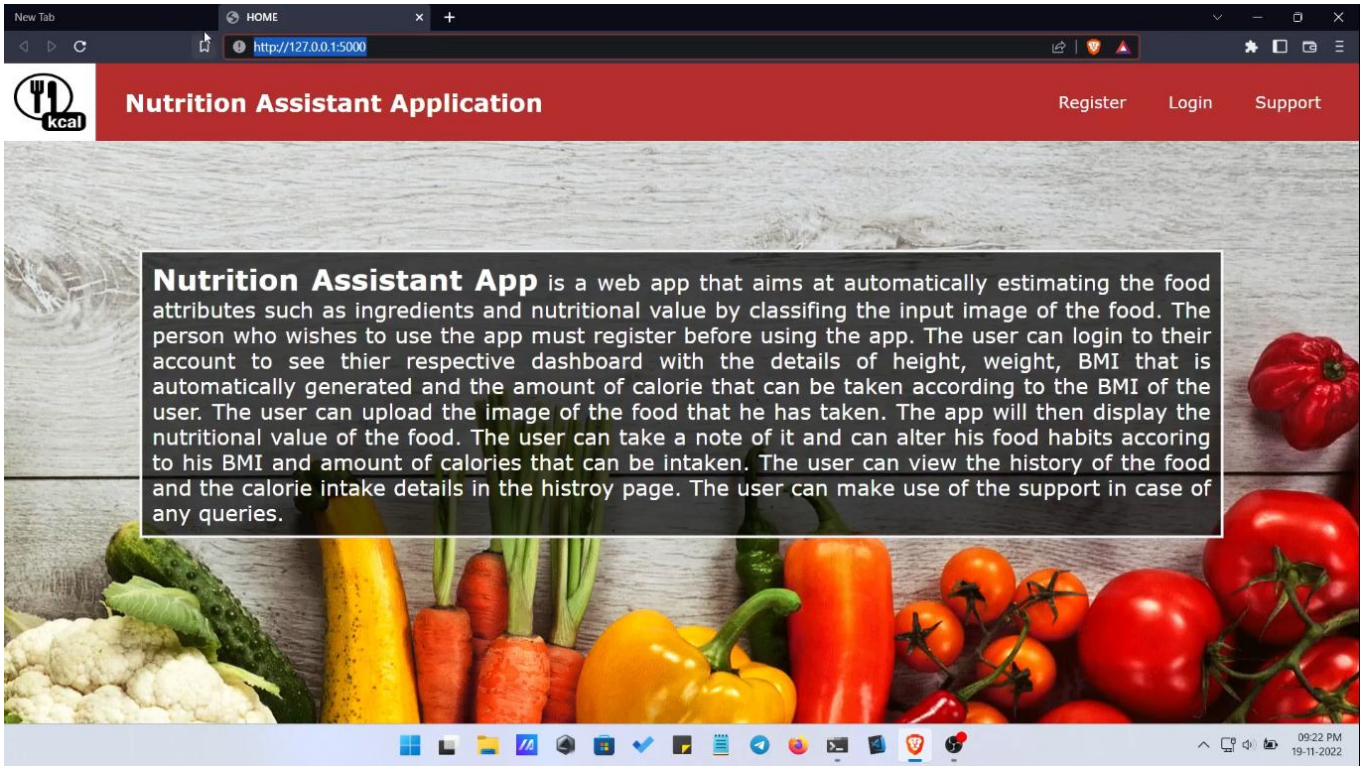
```
* {
  padding: 0;
  margin: 0;
  text-decoration: none;
  list-style: none;
  font-family: verdana;
  box-sizing: border-box;
}
body {
  background-color: # rgb(180, 45, 45);
}
span {
  font-size: 30px;
  text-align: center;
  font-weight: bold;
}
```

```
img.logo {
  height: 60px;
  width: 80px;
  margin: 10px;
}
header {
  display: flex;
  justify-content: space-between;
  align-items: center;
}
nav {
  background-color:  rgb(180, 45, 45);
  height: 80px;
  width: 100%;
}
label.nutri {
  padding: 0 30px;
  color: white;
  font-size: 25px;
  line-height: 80px;
  font-weight: bold;
}
nav ul {
  float: right;
  margin-right: 20px;
}
nav ul li {
  display: inline-block;
  line-height: 80px;
  margin: 0 5px;
}
nav ul li a {
  color: white;
  font-size: 17px;
  border: 1px solid transparent;
  padding: 7px 13px;
```

```
border-radius: 3px;
}
a.active,
a:hover {
  background: #80cc66;
  border: 1px solid white;
  transition: 0.5s;
}
.checkbtn {
  font-size: 30px;
  color: crimson;
  float: right;
  line-height: 80px;
  margin-right: 40px;
  cursor: pointer;
  display: none;
}
#check {
  display: none;
}
.bg-text {
  background-color: rgb(0, 0, 0); /* Fallback color */
  background-color: rgba(0, 0, 0, 0.7); /* Black w/opacity/see-through */
  color: white;
  border: 3px solid #f1f1f1;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  z-index: 2;
  width: 80%;
  padding: 10px;
  text-align: justify;
  font-size: 22px;
}
@media (max-width: 992px) {
  label.nutri {
```

```
    font-size: 30px;
    padding-left: 50px;
   }
   nav ul li a {
    font-size: 16px;
   }
   .bg-text {
    margin-top: 50px;
    top: 50%;
    left: 50%;
   }
}
@media (max-width: 858px) {
   .checkbtn {
    display: block;
   }
   label.nutri {
    display: none;
   }
   .bg-text {
    font-size: 15px;
    top: 50%;
    left: 50%;
   }
   ul {
    position: fixed;
    width: 100%;
    height: 100vh;
    background: #ffee99;
    z-index: 20;
    top: 80px;
    left: -100%;
    text-align: center;
    transition: all 0.5s;
    color:#f1f1f1;
   }
```

```
nav ul li {
    display: block;
    margin: 50px 0;
    line-height: 30px;

  }
  nav ul li a {
    font-size: 20px;
    color: white;
  }
  a.active,
  a:hover {
    background: none;
    border: none;
    color: #f26d1b;
  }
  #check:checked ~ ul {
    left: 0;
  }
}
section {
  width: 100%;
  background: url(mainnutri.png);
  background-position: center;
  background-size: cover;
  height: calc(100vh - 80px);
  background-repeat: no-repeat;
}
```



Register Page:

The register page asks the user details like First name, Last name, Email, Phone number, Password. After clicking on the register button , we will check whether the user has already registered or not. If they have already registered, they will be directed to login page. If the person does not have an account, then they will receive a confirmation email and will be directed to personal details page.

Registration.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Responsive Registration Form</title>
    <meta
      name="viewport"
      content="width=device-width,
      initial-scale=1.0"
    />
    <link
      rel="stylesheet"
      href="static/Registration-style.css"
    />
  </head>
```

```
<body>
  <div class="container">
    <h1 class="form-title">
      <img
        class="logo"
        src="{{url_for('static',filename='NutriAux.png')}}"
        alt="logo"
      /> Registration
    </h1>
    <form action="/register" method="post">
      <div class="main-user-info">
        <div class="user-input-box">
          <label for="firstName">First Name</label>
          <input
            type="text"
            id="firstName"
            name="firstName"
            placeholder="Enter First Name"
            required
          />
        </div>
        <div class="user-input-box">
          <label for="lastName">Last Name</label>
          <input
            type="text"
            id="lastName"
            name="lastName"
            placeholder="Enter Last Name"
            required
          />
        </div>
        <div class="user-input-box">
          <label for="email">Email</label>
          <input
            type="email"
            id="email"
            name="email"
            placeholder="Enter Email"
            required
          />
        </div>
        <div class="user-input-box">
          <label for="phoneNumber">Phone Number</label>
          <input
            type="text"
            id="phoneNumber"
```

```
        name="phoneNumber"
                  placeholder="Enter Phone Number"
                  required
                 />
                </div>
                <div class="user-input-box">
                 <label for="password">Password</label>
                 <input
                  type="password"
                  id="password"
                  name="password"
                  placeholder="Enter Password"
                  pattern="^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{5,}$"
                  required
                 />
                </div>
                <div class="user-input-box">
                 <label for="confirmPassword">Confirm Password</label>
                 <input
                  type="password"
                  id="confirmPassword"
                  name="confirmPassword"
                  placeholder="Confirm Password"
                  pattern="^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{5,}$"
                  required
                 />
                </div>
                <div>
                 <span style="color: white; font-size: 13px"
                  >Password must be minimum Five characters with at least one letter
                  and one number</span
                 >
                </div>
               </div>
               <div class="form-submit-btn">
                <input type="submit" value="Register" />
               </div>
    <div class="go-back-btn">
              <a href="Home.html" class="goback">Go Back</a>
             </div>
            </form>
           </div>
          </body>
</html>
```

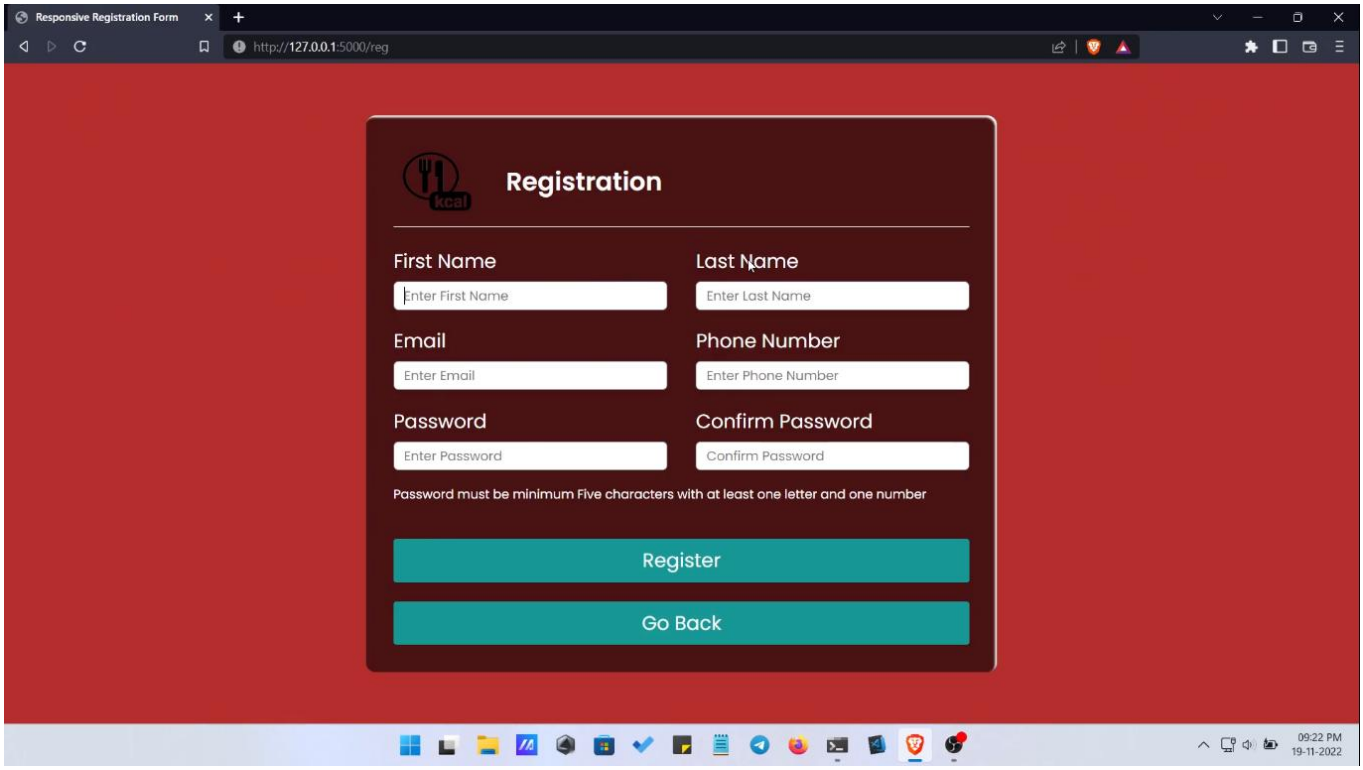
Registration-style.css

```
@import
url("https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600;700&display=
swap");

* {
  padding: 0;
  margin: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}
img.logo {
  height: 60px;
  width: 70px;
  margin: 10px;
}

body {
  display: flex;
  height: 100vh;
  justify-content: center;
  align-items: center;
  background-color: rgb(180, 45, 45);
  background-blend-mode: color-burn;
  background-size: 430px 400px;
  /*background: linear-gradient(to bottom right, #f9d423 0%, #ff4e50 100%);}*/
}
.container {
  width: 100%;
  max-width: 650px;
  background: rgba(
    0,
    0,
    0,
    0.6
  ); /*fdc100 - 253,193,0   06837f - 6,131,127  b4ffff - 180,255,255  02cecb - 2,206,203*/
  padding: 28px;
  margin: 0 28px;
  border-radius: 10px;
  box-shadow: inset -2px 2px 2px white;
}

.form-title {
  font-size: 26px;
  font-weight: 600;
  text-align: center;
  padding-bottom: 6px;
  color: white;
  border-bottom: solid 1px white;
  display: flex;
```

```
align-items: center;
}

.main-user-info {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
  padding: 15px 0;
}

.user-input-box:nth-child(2n) {
  justify-content: end;
}

.user-input-box {
  display: flex;
  flex-wrap: wrap;
  width: 50%;
  padding-bottom: 12px;
}

.user-input-box label {
  width: 95%;
  color: white;
  font-size: 20px;
  font-weight: 400;
  margin: 5px 0;
}

.user-input-box input {
  height: 30px;
  width: 95%;
  border-radius: 5px;
  outline: none;
  border: 1px solid grey;
  padding: 0 10px;
}

.gender-title {
  color: white;
  font-size: 20px;
  font-weight: 400;
}

.gender-category {
  margin: 20px 0;
  color: white;
}
```

```
.gender-category label {
  padding: 0 20px 0 2px;
}

.gender-category label,
.gender-category input,
.form-submit-btn input {
  cursor: pointer;
}

.form-submit-btn {
  margin-top: 20px;
}

.form-submit-btn input {
  display: block;
  width: 100%;
  margin-top: 10px;
  font-size: 20px;
  padding: 7px;
  border: none;
  border-radius: 3px;
  color: rgb(255, 255, 255);
  background:rgba(2, 206, 203, 0.7);
}

.form-submit-btn input:hover {
  background: rgba(2, 206, 203, 0.7);
  color: rgb(255, 255, 255);
}
.go-back-btn {
  margin-top: 20px;
}
.go-back-btn a {
  display: block;
  width: 100%;
  margin-top: 10px;
  font-size: 20px;
  padding: 7px;
  text-decoration: none;
  border: none;
  text-align: center;
  border-radius: 3px;
  color: rgb(255, 255, 255);
  background:rgba(2, 206, 203, 0.7);
}
.go-back-btn a:hover {
  background: rgba(2, 206, 203, 0.7);
  color: rgb(255, 255, 255);
```

```
    cursor: pointer;
  }

  @media (max-width: 600px) {
    .container {
      min-width: 280px;
    }

    .user-input-box {
      margin-bottom: 12px;
      width: 100%;
    }

    .user-input-box:nth-child(2n) {
      justify-content: space-between;
    }

    .gender-category {
      display: flex;
      justify-content: space-between;
      width: 100%;
      font-size: 17px;
    }

    .main-user-info {
      max-height: 380px;
      overflow: auto;
    }

    .main-user-info::-webkit-scrollbar {
      width: 0;
    }
  }
```


Responsive Registration Form
http://127.0.0.1:5000/reg
Registration
First Name
Enter First Name
Last Name
Enter Last Name
Email
Enter Email
Phone Number
Enter Phone Number
Password
Enter Password
Confirm Password
Confirm Password
Password must be minimum Five characters with at least one letter and one number
Register
Go Back
09:22 PM
19-11-2022

Personal Details:

The personal details page asks for age, weight, height, gender, and daily activities of the user. After entering all the details the user has to click the submit button which will direct the user to login page.

personaldetails.html:

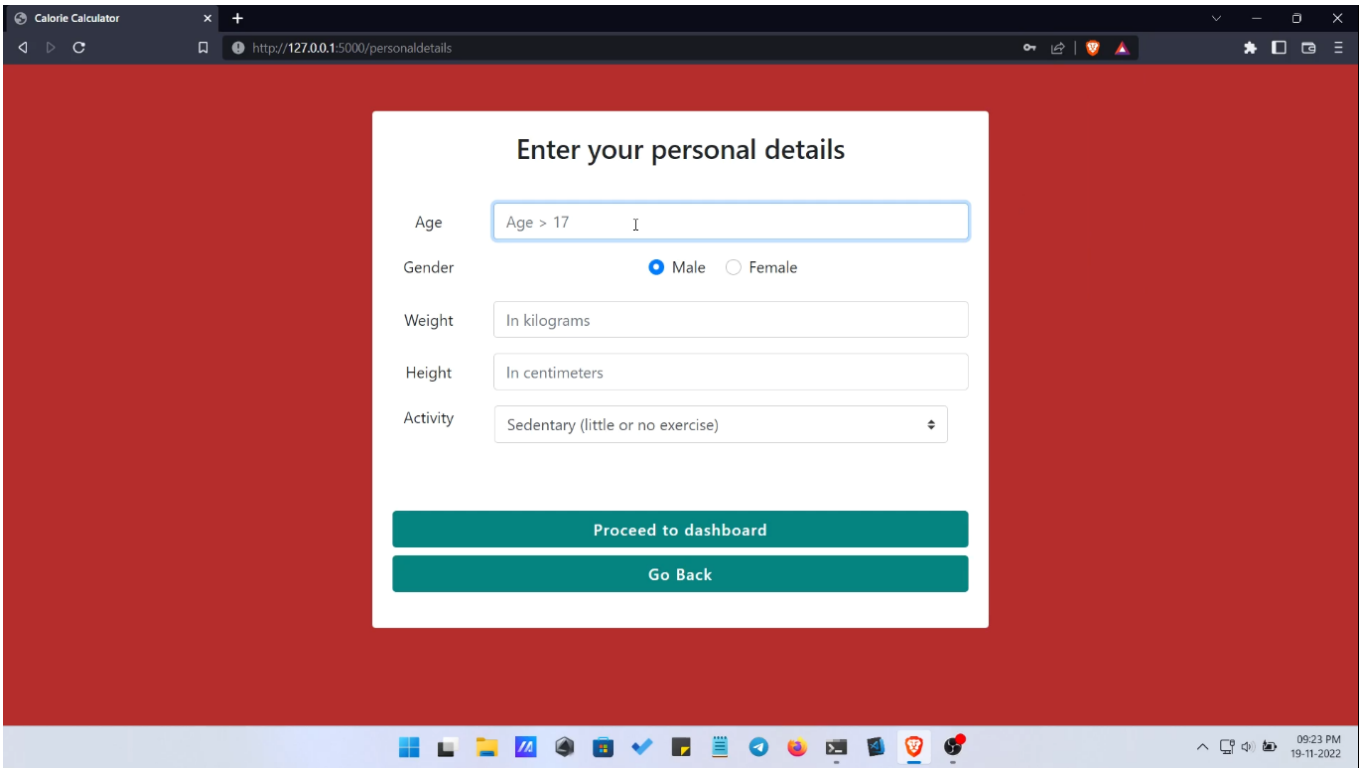
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8" />
    <meta
      name="viewport"
      content="width=device-width, initial-scale=1, shrink-to-fit=no"
    />

    <!-- Bootstrap CSS -->
    <link
      rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
      integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
      crossorigin="anonymous"
    />
    <title>Calorie Calculator</title>

    <style>
      body {
        background-color: rgb(180, 45, 45);
        background-image: url("{{url_for('static',filename='calorie_calc_bg.png')}}");
        background-blend-mode: color-burn;
        background-size: 430px 400px;
      }
      .results {
        display: none;
      }

      .btn {
        background: lighten(#06837f, 3%);
        font-weight: 600;
        letter-spacing: 1px;
      }
    </style>
  </head>
```

```
<body>
   <div class="container">
    <div class="row">
     <div class="col-lg-7 mx-auto">
      <div class="card card-body text-center mt-5">
       <h3 class="heading display-5 pb-3 mb-4">
        Enter your personal details
       </h3>
       <form id="calorie-form" method="post" action="/addpersonaldetails">
        <div class="form-group row">
         <label for="age" class="col-sm-2 col-form-label">Age</label>
         <div class="col-sm-10">
          <input
           type="text"
           class="form-control"
           id="age"
           name="age"
           placeholder="Age > 17"
           pattern="(?:1[01][0-9]|120|1[7-9]|[2-9][0-9])$"
           required
          />
         </div>
        </div>

        <fieldset class="form-group">
         <div class="row">
          <legend class="col-form-label col-sm-2 pt-0">Gender</legend>
          <div class="col-sm-10" id="form-radio">
           <div
            class="custom-control custom-radio custom-control-inline"
           >
            <input
             type="radio"
             id="male"
             name="Gender"
             value="male"
             class="custom-control-input"
             checked="checked"
            />
            <label class="custom-control-label" for="male"
             >Male</label
            >
           </div>
           <div
            class="custom-control custom-radio custom-control-inline">
```

```
<input
                type="radio"
                id="female"
                name="Gender"
                value="female"
                class="custom-control-input"
              />
              <label class="custom-control-label" for="female"
                >Female</label
              >
            </div>
          </div>
        </div>
      </fieldset>

      <div class="form-group row">
        <label for="weight" class="col-sm-2 col-form-label"
          >Weight</label
        >
        <div class="col-sm-10">
          <input
            type="number"
            class="form-control"
            id="weight"
            name="weight"
            placeholder="In kilograms"
            required
          />
        </div>
      </div>

      <div class="form-group row">
        <label for="height" class="col-sm-2 col-form-label"
          >Height</label
        >
        <div class="col-sm-10">
          <input
            type="number"
            class="form-control"
            id="height"
            name="height"
            placeholder="In centimeters"
            required
          />
```

```
</div>
        </div>


 <div class="form-group row">
          <legend class="col-form-label col-sm-2 pt-0">Activity</legend>
          <select
           class="custom-select col-sm-10 col-lg-9 ml-3"
           id="list"
           name="activity"
          >
           <option selected value="1">
            Sedentary (little or no exercise)
           </option>
           <option value="2">
            Lightly active (light exercise/sports 1-3 days/week)
           </option>
           <option value="3">
            Moderately active (moderate exercise/sports 3-5 days/week)
           </option>
           <option value="4">
            Very active (hard exercise/sports 6-7 days a week)
           </option>
           <option value="5">
            Extra active (very hard exercise/sports & physical job or 2x
            training)
           </option>
          </select>
         </div>

         <div class="form-group" style="margin-top: 70px">
          <input
           type="submit"
           value="Proceed to dashboard"
           class="btn btn-block"
           style="background-color: #06837f; color: white"
          />
          <input
           type="button"
           onclick="window.history.back()"
           value="Go Back"
           class="btn btn-block"
           style="background-color: #06837f; color: white"
          />
         </div>
```

```
 </form>
        </div>
       </div>
      </div>
     </div>
     <script
      src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
      integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
      crossorigin="anonymous"
     ></script>
     <script
      src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
      integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
      crossorigin="anonymous"
     ></script>
     <script
      src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
      integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
      crossorigin="anonymous"
     ></script>
    </body>
</html>
```



Login Page:

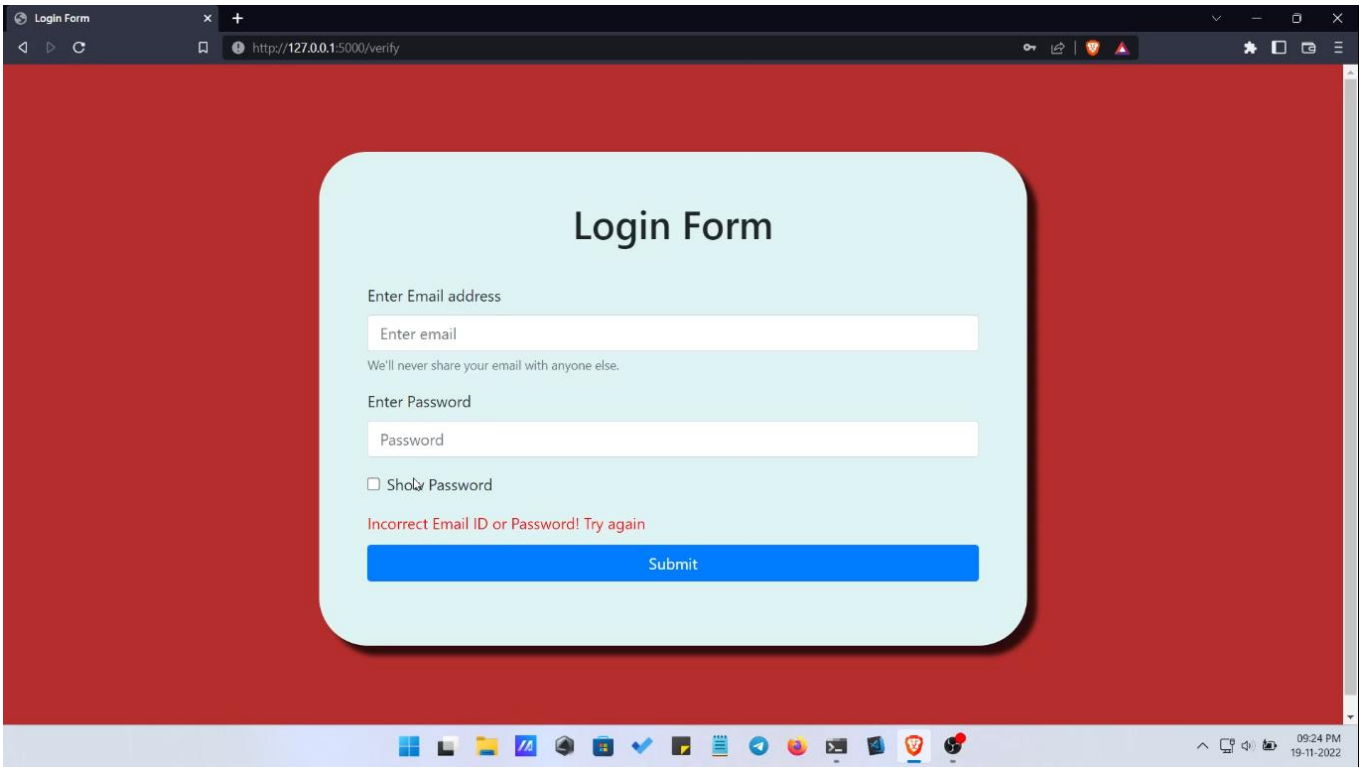
The user will be asked to enter their registered email and password. After entering correct email and password the user have to click the login button. After that the user will be directed to their respective dashboard.

Login.html

```
<! DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Login Form</title>
    <link
      rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
    />
  </head>
  <style>
    body {
      padding-top: 50px;
      background-color: rgb(180, 45, 45)
      background-image: url("{{url_for('static',filename='calorie_calc_bg.png')}}");
      background-blend-mode: color-burn;
      background-size: 430px 400px;
    }
    .login-form {
      background: #dcf1f1;
      margin-top: 40px;
      margin-bottom: 100px;
      padding: 50px;
      border-radius: 50px;
      box-shadow: 10px 10px 5px 0px rgba(0, 0, 0, 0.75);
    }

    .btn-primary {
      width: 100%;
    }
  </style>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-md-8 offset-md-2">
          <div class="login-form">
            <h1 class="text-center" style="padding-bottom: 30px">Login Form</h1>

            <form action="/verify" method="post">
              <div class="form-group">
                <label for="exampleInputEmail1">Enter Email address </label>
                <input
                  type="email"
                  name="email"
                  class="form-control"
```

```
            id="exampleInputEmail1"
            aria-describedby="emailHelp"
            placeholder="Enter email"
            required
          />
          <small id="emailHelp" class="form-text text-muted">
            We'll never share your email with anyone else.
          </small>
        </div>
        <div class="form-group">
          <label for="exampleInputPassword1"> Enter Password </label>
          <input
            type="password"
            class="form-control"
            id="password"
            name="password"
            placeholder="Password"
            pattern="^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{5,}$"
            required
          />
        </div>
        <div class="form-group form-check">
          <input
            type="checkbox"
            class="form-check-input"
            onclick="showpass()"
          />
          <label class="form-check-label" for="exampleCheck1">
            Show Password
          </label>
        </div>
        <span style="color: red">{{message}}</span>
        <button
          type="submit"
          class="btn btn-primary"
          style="margin-top: 10px"
        >
          Submit
        </button>
      </form>
    </div>
  </div>
 </div>
</div>
<script>
 function showpass() {
  var x = document.getElementById("password");
  if (x.type === "password") {
    x.type = "text";
  } else {
    x.type = "password";
  }
 }
</script>
```

```
        <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
          integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
          crossorigin="anonymous"
        ></script>
        <script
          src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
          integrity="sha384-
      UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
          crossorigin="anonymous"
        ></script>
        <script
          src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
          integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
          crossorigin="anonymous"
        ></script>
        <script></script>
      </body>
</html>
```



7.2 Feature

Dashboard page:

The user's personal details such as height, weight, BMI and daily calorie intake will be displayed in their respective dashboard. The user can select either upload image option or track history option. This page will contain a navbar with link to support page and a log out option which will redirect the user to Home page.

dashboard.html

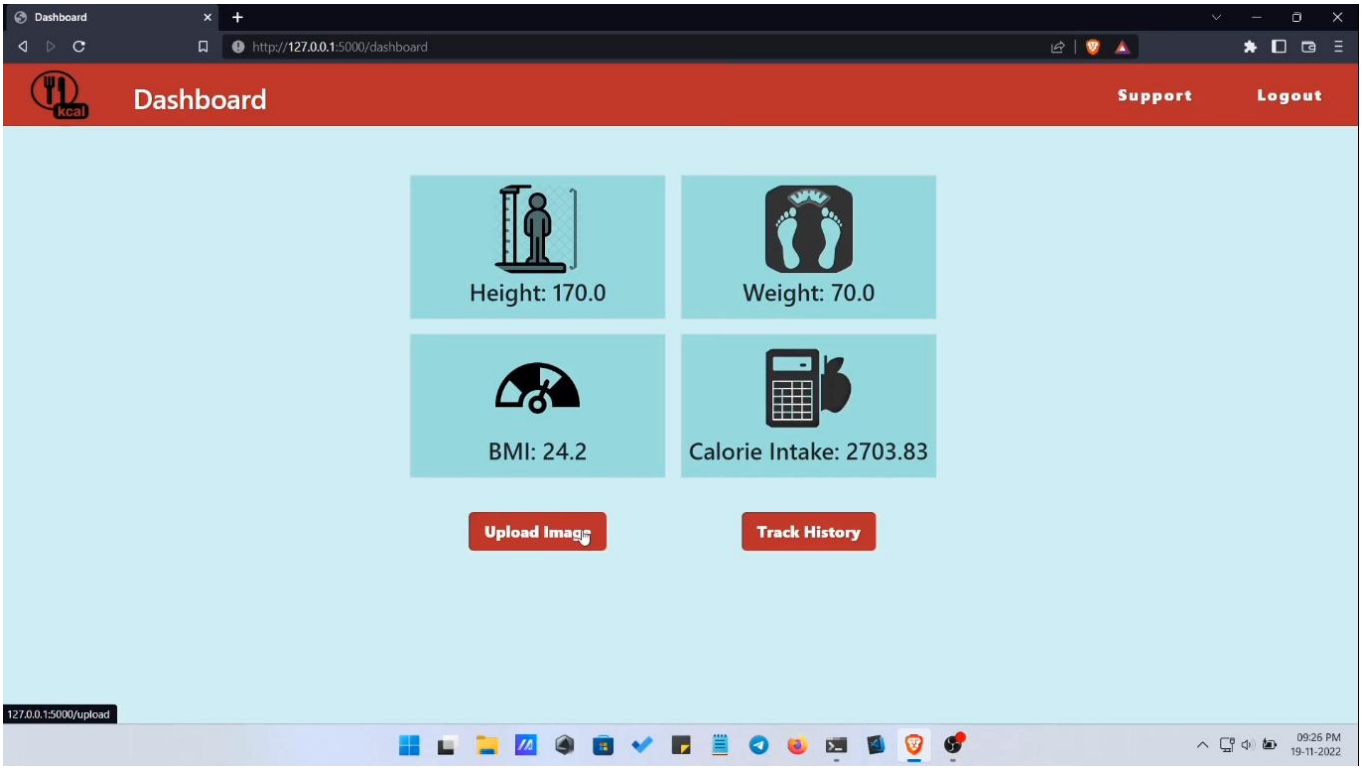
```
<!DOCTYPE html>
<html>
   <head>
      <title>
         Dashboard
      </title>
      <meta name="viewport" content="width=device-width , initial-scale=1.0">
      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
      <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>
      <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
      <link rel="stylesheet" href= "static/dashboard.css">
   </head>
   <body>

      <nav class="navbar navbar-expand-lg ">
         <div class="container-fluid">
           <img class="navbar-brand"  src="static/applogo.png" alt="Nutrition Assistant Application"
height="60px" width="60px" ><h3 style="color: white; margin-left: 30px;margin-top: 15px;">Dashboard</h3>
           <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
             <i class="fa fa-bars" style="font-size:36px"></i>

           </button>
           <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav ml-auto text-right">
              <li class="nav-item">
                <a class="nav-link" href="{{url_for('support')}}">Support</a>
              </li>
              <li class="nav-item ">
                <a class="nav-link" href="{{url_for('home')}}">Logout</a>
              </li>
            </ul>
           </div>
         </div>
      </div>
        </nav>
        <div class="container">
        <div class="row  ">
          <div class="column border box">
            <img src= "static/height.jpg" class="rounded-circle" alt="height" id="pic" >
            <h4>Height: {{height}} </h4>
          </div>
          <div class="column border box">
```

```
<img src="static/weight.png" class="rounded-circle" alt="weight" id="pic" >
          <h4>Weight: {{weight}} </h4>
         </div>
         </div>
         <div class="row my-0 ">
          <div class="column border box">
           <img src="static/bmi.jpg"class="rounded-circle" alt="bmi" id="pic" >
           <h4>BMI: {{BMI}} </h4>
          </div>
          <div class="column border box">
           <img src="static/caloriesintake.png" class="rounded-circle" alt="Calorie" id="pic" >
           <h4>Calorie Intake: {{daily_calorie_intake}} </h4>
          </div>
         </div>
         <div class="row">
         <div class="column">
          <a href="{{url_for('upload')}}" id="bu1" class="btn btn-info btn-lg">Upload Image</a>
         </div>
         <div class="column ">
          <a href="{{url_for('history')}}" id="bu1" class="btn btn-info btn-lg">Track History</a>
         </div>
        </div>
       </div>
       </div>
         </div>
        </div>

   </body>
</html>
```

dashboard.css

```
body{
    background-color:#D4F1F4;

}
.nav-item{
    text-align: end;
}
.navbar-toggler{
    outline: none !important;
    padding: 0 !important;
}
.navbar-toggler i{
    width: 40px;
}
#bu1{
    text-decoration: none;
}
.navbar{
    background-color:rgb(180, 45, 45);
    top: 0;
    width: 100%;
    overflow: hidden;
    height: 65px;
}
.nav-link{
    color: white;
    font-weight: 900;
    margin-left: 50px;
    letter-spacing: 2px;
}
.nav-item a:hover{
    color: #FDC100;
}
* {
    box-sizing: border-box;
  }

  .row {
    display: flex;
    margin-top: 30px;
    margin-bottom: 15px;
  }
  .box{
    background-color: #a0dfdf;
  }
  .container{
    margin: 50px auto;
    width:40%;
    text-align: center;

  }
```

```
        .container h3{
            color: #f15337;
        }
        .container img{
          object-fit:contain;
        }
        .column {
          flex: 33.33%;
          padding: 5px;
          margin-right:15px;
        }
        #pic{
          height: 100px;
          width: 130px;
        }

      #img{
          background-color: #24a0ed;
          color: black;
          padding: 14px 25px;
          text-align: center;
          text-decoration: none;
          display: inline-block;
      }
      #bu1{
          background-color:rgb(180, 45, 45) ;
          color: white;
          font-weight: 800;
          font-size: medium;
          border-style: none;
          text-align:center;

}
```



Upload Image Page:

In this page the user can either choose the meal image from the file or drap and drop the meal image. After uploading the image the user should click the submit button. After clicking the submit button the nutritional value of the meal will be displayed.

upload.html:

```
<!DOCTYPE html>
<html>
   <head>
      <title>Upload image</title>
      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
      <link rel="stylesheet" href="static/upload.css">
   </head>
   <body>
      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
      <form action="/getnutri" method="post" enctype="multipart/form-data">
```

```
   <div class="container uploadOuter ">
         <table class="uploadimage">
            <tr>
         <th><label for="uploadFile" id="file" ><h2>Choose your image to upload</h2><i class="fa
fa-file-photo-o" style="font-size:48px;color:red"></i></i>

         </label></th>
         <th>    </th>
         <th><h3>OR</h3></th>

        <th><span class="dragBox" >
         <h2>Drag and Drop image here</h2><i class="fa fa-file-photo-o" style="font-
size:48px;color:red"></i>
        <input name="file" type="file" onChange="dragNdrop(event)"  ondragover="drag()"
ondrop="drop()" id="uploadFile" required/>
       </span></th>
     </tr>
     </table>

        <button class="btn btn-primary btn-lg" id="b1" onclick="window.history.back()">Go Back
</button>
        <button class="btn btn-primary btn-lg" id="b1" type="submit" >Submit </button>

        </div>
       </form>
         </div>
        </div>
        <div>
         <table class="table table-bordered"><br/><br/>
            <tr>
              <th scope="col" style="background-color:#050a30;color:white;">Calories</th>
              <td style="background-color:white;">{{calories}}</td>
            </tr>
            <tr>
              <th scope="col" style="background-color:#050a30;color:white;">Protein</th>
              <td style="background-color:white;">{{protein}}</td>

            </tr>
            <tr>
              <th scope="col" style="background-color:#050a30;color:white;">Fat</th>
              <td style="background-color:white;">{{fat}}</td>

            </tr>
            <tr>
              <th scope="col" style="background-color:#050a30;color:white;">Carbs</th>
              <td style="background-color:white;">{{carbs}}</td>
            </tr>

         </table>
        </div>
```

```
 <div id="preview" style="height:105px">  </div>

        </div>
        </div>
        <script>
          "use strict";
function dragNdrop(event) {
   var fileName = URL.createObjectURL(event.target.files[0]);
   var preview = document.getElementById("preview");
   var previewImg = document.createElement("img");
   previewImg.setAttribute("src", fileName);
   previewImg.setAttribute("width", "100px");
   previewImg.setAttribute("height", "100px");
   preview.innerHTML = "";
   preview.appendChild(previewImg);
}
function drag() {
   document.getElementById('uploadFile').parentNode.className = 'draging dragBox';
}
function drop() {
   document.getElementById('uploadFile').parentNode.className = 'dragBox';
}
        </script>
   </body>
</html>
```

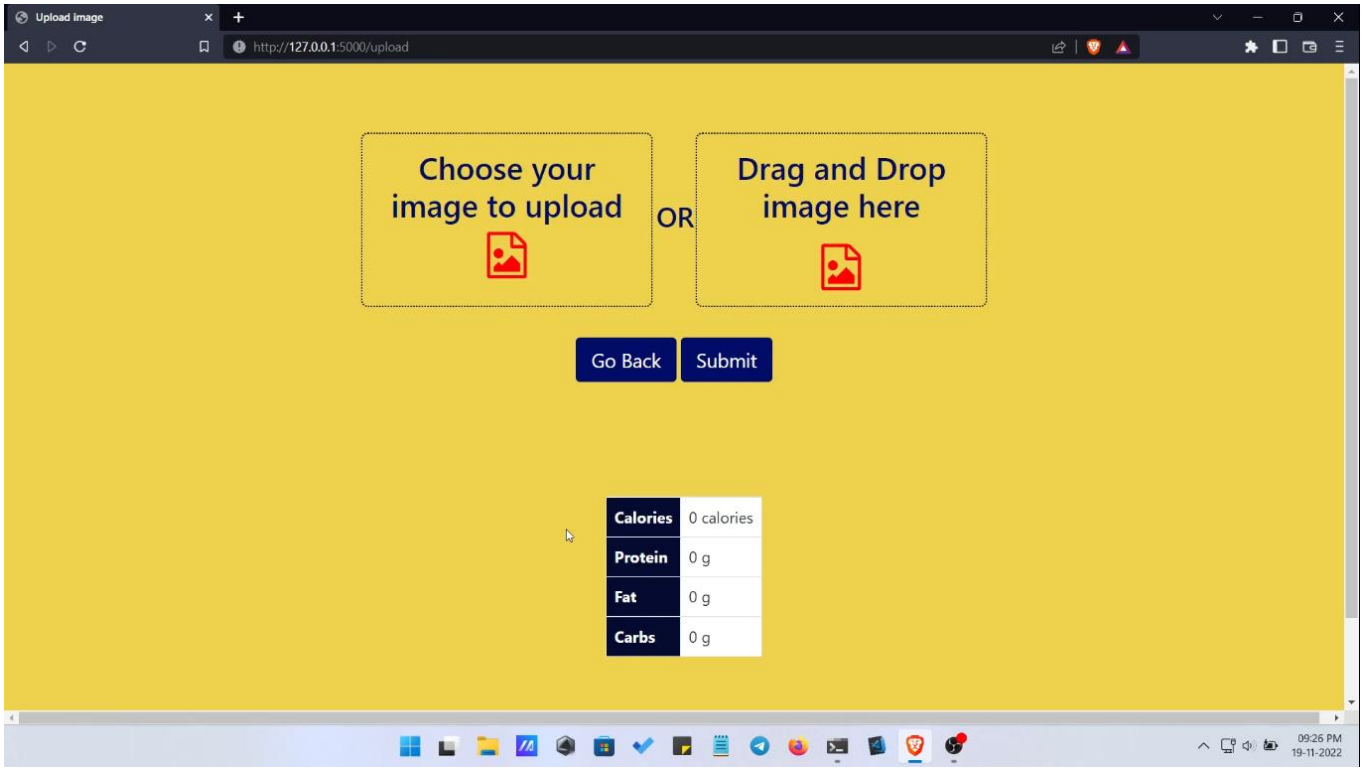
upload.css:

```
body {



  background-color: rgb(236, 207, 76);

  background-blend-mode: color-burn;

  background-size: 430px 400px;

}

.uploadOuter {

  text-align: center;

  padding: 30px;

}
```

```
h2,
h3 {
  color: #000c66;
}
table {
  display: flex;
  flex-direction: row;
  justify-content: center;
  margin: 10px;
  margin-bottom: auto;
  padding: 30px;
}
#file {
  padding: 15px;
  width: 300px;
  height: 180px;
  text-align: center;
  border: 2px dotted #050a30;
  border-radius: 7px;
  cursor: pointer;
  position: relative;
  color: #000c66;
}
.dragBox {
  padding: 15px;
  width: 300px;
  height: 180px;
```

```
  border-radius: 7px;
  margin: 0 auto;
  position: relative;
  text-align: center;
  font-weight: bold;
  line-height: 95px;
  border: 2px dotted #050a30;
  display: inline-block;
  transition: transform 0.3s;
}
input[type="file"] {
  position: absolute;
  height: 100%;
  width: 100%;
  opacity: 0;
  top: 0;
  left: 0;
}
.draging {
  transform: scale(1.1);
}
#preview {
  text-align: center;
}
img {
  max-width: 100%;
}
```

```
#b1 {
  background-color: #000c66;
  color: white;
  border: none;
}
```



Track History Page:

The user will be asked to enter food name and calorie value of the food. After entering, the user must click the add button. After clicking the add button , the user meal history will be added to the table below.

History.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Track history</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
```

```
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous"/>
   <!--Navbar fonts-->
   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-
free@6.2.0/css/fontawesome.min.css" integrity="sha384-
z4tVnCr80ZcL0iufVdGQSUzNvJsKjEtqYZjiQrrYKlpGow+btDHDfQWkFjoaz/Zr"
crossorigin="anonymous">

   <!--Adding icons-->
   <!--Date icon and bootstrap alignment-->
   <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css"/>
   <!--Burger icon-->
   <script src='https://kit.fontawesome.com/a076d05399.js' crossorigin='anonymous'></script>
   <!--Bolt icon-->
   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
   <!--Nav Icon-->
   <script src="https://kit.fontawesome.com/a076d05399.js" crossorigin="anonymous"></script>

   <!--Link to trackhistorystyle.css file-->
   <link rel="stylesheet" href="static/trackhistorystyle.css"/>

  </head>
  <body>
   <!--Navbar-->
   <div class="bg">

     <nav>
      <img src="static/applogo.png" style="width:60px; height:55px;margin-right:15px;margin-
bottom: 17px;margin-left:15px;"/>
     <input type="checkbox" id="check">
     <label for="check" class="checkbtn">
      <i class="fas fa-bars"></i>
     </label>
     <label class="logo"><span style="margin-top:5px;">Track History</span></label>
      <ul>
       <li>
        <button class="navbutton" onclick="window.history.back()" ><a >Go Back</a></button>
       </li>
       <li>
        <a href="#">Upload Image</a>
       </li>
       <li class="nav-item">
        <a href="{{url_for('support')}}">Support</a>
       </li>
       <li class="nav-item">
        <a href="{{url_for('home')}}">Log out</a>
       </li>
      </ul>
```

```
</nav>


   <div class="container" style="margin-top: 50px;">
    <form action='/addhistory' method="POST">
     <div class="row">
      <div class="col-md-3 mt-3">
       <!--component-->
       <span style="font-size:17px;">Date:</span>
        <div class="input-group date" id="datePicker"  >
         <span class="input-group-addon">
          <i class="glyphicon glyphicon-calendar" style="color:rgb(180, 45, 45)"></i>
         </span>
         <input
          type="text"
          class="form-control"
          name="date"
          id="date"
          readonly
          style="background-color:white"
         />
        </div>
       </div>

       <div class="col-md-3 mt-3">
        <span style="font-size:17px;">Enter the food name:</span>

         <div class="input-group date" id="datePicker" >
          <span class="input-group-addon" >
           <i class='fas fa-hamburger ' style="color:rgb(180, 45, 45)"></i>
          </span>
          <input
           type="text"
           class="form-control"
           name="meal_name"
           id="meal_name"
          />
         </div>
        </div>
        <div class="col-md-3 mt-3">
         <span style="font-size:17px;">Enter Calories:</span>
          <div class="input-group date" id="datePicker" >
           <span class="input-group-addon">
            <i class="fa fa-bolt" style=" color:rgb(180, 45, 45)"></i>
           </span>
           <input
            type="text"
            class="form-control"
            name="calories"
            id="calories"
           />
```

```
            </div></div>
          <div class="col-md-3 mt-3" style="margin-left:auto">


<button type="submit" class="btn btn-default" width="50px" style="margin-top:20px ;background-color:white;color: rgb(180, 45, 45);border-color: #0b3c49;" ><b>Add +</b></button>
          </div>

        </div>
        <div class="row">
          <div class="col-md-12">
            <table class="table table-bordered"><br/><br/>
              <thead class="thead" style="background-color:rgb(180, 45, 45);color:white;" >
                <tr>
                  <th scope="col">Date</th>
                  <th scope="col">Food Name</th>
                  <th scope="col">Calories</th>
                </tr>
              </thead>
              <tbody>
                {%set count = namespace(value=0)%}
                {%for i in range(no_of_rows)%}
                  <tr>
                    {%for j in range(3)%}
                      <td>{{history[count.value]}}{%set count.value = count.value + 1%}</td>
                    {%endfor%}
                  </tr>
                {%endfor%}
              </tbody>
            </table>
          </div>
        </div>
      </div>
    </form>
  </div>
 </div>
  <!--Link to trackhistory Javascript-->
  <script src="static/trackhistory.js"></script>

  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js" integrity="sha384-
```

```
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
crossorigin="anonymous"></script>
  </body>
</html>
```

trackhistorystyle.css:

```
* {
  margin: 0;
  padding: 0;
  text-decoration: none;
  list-style: none;
  box-sizing: border-box;
}
body {
  font-family: montserrat;
  background: #dcf1f1;
}
.bg {
  height: 100vh;
  width: 100%;
  background-image: linear-gradient(#dcf1f16a, #dcf1f16a),


}
button.navbutton {
  background: none;
  color: white;
  border: none;
  padding: 0;
  font: inherit;
  cursor: pointer;
  outline: inherit;
}
nav {
  background: rgb(180, 45, 45);
  height: 80px;
```

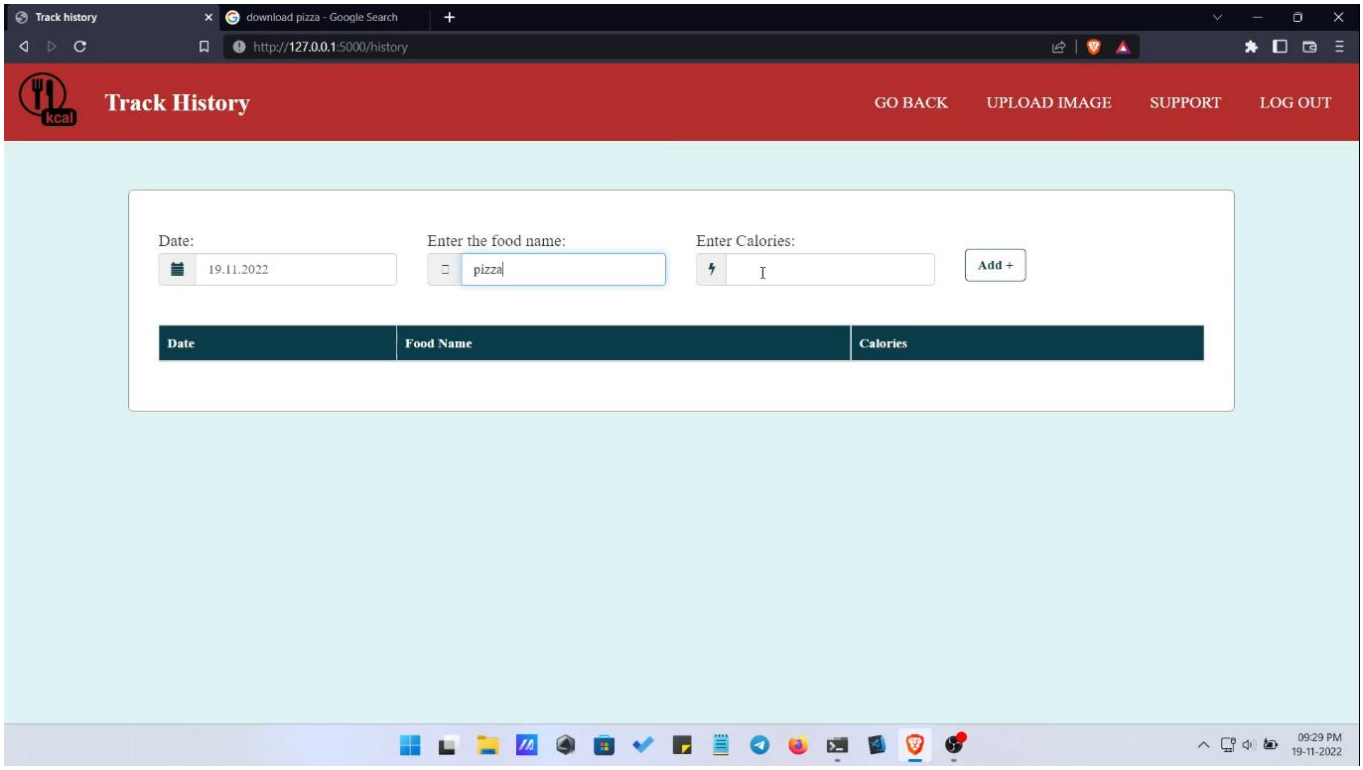
```
width: 100%;
}
label.logo {
  color: white;
  font-size: 25px;
  line-height: 80px;
  padding: 0 10px;
  font-weight: bold;
}
nav ul {
  float: right;
  margin-right: 10px;
}
nav ul li {
  display: inline-block;
  line-height: 80px;
  margin: 0 5px;
}
nav ul li a {
  color: white;
  font-size: 17px;
  padding: 7px 13px;
  border-radius: 3px;
  text-transform: uppercase;
}
a:hover {
  color: white;
  background: #02cecb87;
  text-decoration: none;
}
.checkbtn {
  font-size: 30px;
  color: white;
```

```
  float: right;
  line-height: 80px;
  margin-right: 40px;
  cursor: pointer;
  display: none;
}
#check {
  display: none;
}
@media (max-width: 952px) {
  label.logo {
    font-size: 25px;
  }
  nav ul li a {
    font-size: 16px;
  }
}
@media (max-width: 858px) {
  .checkbtn {
    display: block;
  }
  ul {
    position: fixed;
    width: 100%;
    height: 100vh;
    background: rgb(180, 45, 45);
    top: 80px;
    left: -100%;
    text-align: center;
    z-index: 20;
    transition: all 0.5s;
  }
  nav ul li {
```

```
    display: block;
    margin: 50px;
    line-height: 30px;
  }
  nav ul li a {
    font-size: 20px;
  }
  a:hover {
    background: none;
    color: #90f1bab6;
  }
  button.navbutton:hover {
    color: rgb(180, 45, 45);
  }
  #check:checked ~ ul {
    left: 0;
  }
}
.container {
  background: white;
  margin-top: 30px;
  margin-bottom: 30px;
  border: 1px solid rgba(0, 0, 0, 0.4);
  padding: 30px;
  border-radius: 6px;
}
```

trackhistory.js:

```
var todaydate = new Date();
var day = todaydate.getDate();
var month = todaydate.getMonth() + 1;
var year = todaydate.getFullYear();
var datestring = day + "." + month + "." + year;
document.getElementById("date").value = datestring;
```


Track history
download pizza - Google Search
http://127.0.0.1:5000/history
Track History
GO BACK
UPLOAD IMAGE
SUPPORT
LOG OUT
Date:
19.11.2022
Enter the food name:
pizza
Enter Calories:
Add +
Date
Food Name
Calories
09:29 PM
19-11-2022

Support page:

If the user has any doubts, then he can come to the support page anytime with the help of link provided in the navigation bar.

support.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Support</title>
    <!-- CSS only -->
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
      crossorigin="anonymous"
    />
    <link
      rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css"
    />
    <!-- JavaScript Bundle with Popper Bootstrap -->
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
      crossorigin="anonymous"
    ></script>
    <link
      rel="stylesheet"
      href="static/supportstyle.css"
    />
  </head>
  <body>
    <img
      src="{{url_for('static',filename='NutriAux.png')}}"
      style="width: 60px; height: 60px; margin-left: 15px; margin-top: 5px"
    />
    <h2 class="support-message">
      <b>Answer for your Queries</b>
    </h2>
    <section>
      <div class="container">
        <div class="row">
```

```
<!--FAQ column-->
<div class="col-md-6" style="margin-top: 30px">
  <div class="accordion" id="accordionExample">
    <div class="accordion-item">
      <h2 class="accordion-header" id="headingOne">
        <button
          class="accordion-button collapsed"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#collapseOne"
          aria-expanded="false"
          aria-controls="collapseOne"
          style="
            --bs-accordion-btn-focus-box-shadow: none;
            --bs-accordion-active-color: #000;
            --bs-accordion-active-bg: #fdc100;
            --bs-accordion-border-color: none;
          "
        >
          How can I use this app?
        </button>
      </h2>
      <div
        id="collapseOne"
        class="accordion-collapse collapse"
        aria-labelledby="headingOne"
        data-bs-parent="#accordionExample"
      >
        <div class="accordion-body">
          <ul>
            <li>
              Complete the <code>registration</code> process by
              clicking the register button present in the home page
            </li>
            <li>
              After successful registration, <code>login</code> into
              your account.
            </li>
            <li>
              After login successfully you will be led to your
              <code>dashboard</code>, where you can see your personal
              details like height, weight, BMI and daily calorie
              intake needed (i.e, how much calorie you should consume
              per day). Also you can select various options like
              Upload image, track history
            </li>
          </ul>
        </div>
      </div>
    </div>
```

```
<div class="accordion-item">
  <h2 class="accordion-header" id="headingTwo">
    <button
      class="accordion-button collapsed"
      type="button"
      data-bs-toggle="collapse"
      data-bs-target="#collapseTwo"
      aria-expanded="false"
      aria-controls="collapseTwo"
      style="
        --bs-accordion-btn-focus-box-shadow:: none;
        --bs-accordion-active-color: #000;
        --bs-accordion-active-bg: #fdc100;
        --bs-accordion-border-color: none;
      "
    >
      How can I Register?
    </button>
  </h2>
  <div
    id="collapseTwo"
    class="accordion-collapse collapse"
    aria-labelledby="headingTwo"
    data-bs-parent="#accordionExample"
  >
    <div class="accordion-body">
      Inorder to use our application , registration process is
      mandatory. This is because we have to keep track of your
      personal details like height, weight etc.Click on the
      <code>register</code> button present inside the home page.
      You will be asked to provide your personal details for
      providing a personalised experience. Once you have
      successfully registered you will receive a confirmation mail
      and you will be redirected to the home page. From there you
      can login and enter into your Dashboard
    </div>
  </div>
</div>
<div class="accordion-item">
  <h2 class="accordion-header" id="headingThree">
    <button
      class="accordion-button collapsed"
      type="button"
      data-bs-toggle="collapse"
      data-bs-target="#collapseThree"
      aria-expanded="false"
      aria-controls="collapseThree"
```

```
      style="
        --bs-accordion-btn-focus-box-shadow: none;
        --bs-accordion-active-color: #000;
        --bs-accordion-active-bg: #fdc100;



        --bs-accordion-border-color: none;
      "
    >
      How can I login?
    </button>
  </h2>
  <div
    id="collapseThree"
    class="accordion-collapse collapse"
    aria-labelledby="headingThree"
    data-bs-parent="#accordionExample"
  >
    <div class="accordion-body">
      It is necessary
      <strong>complete your registration process</strong> before
      you login. Once you have succesfully registered, you can
      click on the <code>login</code> button present inside the
      home page. You will be led to the Login page where you will
      be asked to enter your E-Mail ID and password that you
      entered in the registration process. After successful login
      you will be led to your Dashboard.
    </div>
  </div>
</div>
<div class="accordion-item">
  <h2 class="accordion-header" id="headingFour">
    <button
      class="accordion-button collapsed"
      type="button"
      data-bs-toggle="collapse"
      data-bs-target="#collapseFour"
      aria-expanded="false"
      aria-controls="collapseFour"
      style="
        --bs-accordion-btn-focus-box-shadow: none;
        --bs-accordion-active-color: #000;
        --bs-accordion-active-bg: #fdc100;
        --bs-accordion-border-color: none;
      "
    >
      How can I get the nutritional value of the food that I am
      eating now?
    </button>
```

```
  </h2>
  <div
    id="collapseFour"
    class="accordion-collapse collapse"
    aria-labelledby="headingFour"
    data-bs-parent="#accordionExample"
  >
    <div class="accordion-body">
      With the help of our <code>Upload Image</code> feature you


      can upload an image of the meal that you would like to have.
      Once the image has been uploaded you will be provided with
      the nutrional content present inside that particular dish.
      You can find this feature in the <code>menu bar</code>.
    </div>
  </div>
</div>
<div class="accordion-item">
  <h2 class="accordion-header" id="headingFive">
    <button
      class="accordion-button collapsed"
      type="button"
      data-bs-toggle="collapse"
      data-bs-target="#collapseFive"
      aria-expanded="false"
      aria-controls="collapseFive"
      style="
        --bs-accordion-btn-focus-box-shadow: none;
        --bs-accordion-active-color: #000;
        --bs-accordion-active-bg: #fdc100;
        --bs-accordion-border-color: none;
      "
    >
      How can I track my daily calorie intake?
    </button>
  </h2>
  <div
    id="collapseFive"
    class="accordion-collapse collapse"
    aria-labelledby="headingFive"
    data-bs-parent="#accordionExample"
  >
    <div class="accordion-body">
      With the help of our <code>Track History</code> feature you
      can track your daily calorie intake which will help you to
      know your current progress and achieve your fitness goal.
      You can find this feature in your <code>Dashboard</code>.
    </div>
```

```
      </div>
    </div>
    <div class="accordion-item">
      <h2 class="accordion-header" id="headingSix">
        <button
          class="accordion-button collapsed"
          type="button"
          data-bs-toggle="collapse"
          data-bs-target="#collapseSix"
          aria-expanded="false"
          aria-controls="collapseSix"
          style="



--bs-accordion-btn-focus-box-shadow: none;
            --bs-accordion-active-color: #000;
            --bs-accordion-active-bg: #fdc100;
            --bs-accordion-border-color: none;
          "
        >
          How much calorie should I consume per day?
        </button>
      </h2>
      <div
        id="collapseSix"
        class="accordion-collapse collapse"
        aria-labelledby="headingSix"
        data-bs-parent="#accordionExample"
      >
        <div class="accordion-body">
          With the help of the data that you provide during the
          registration process like Height, Weight, Daily activities,
          etc., we will provide you how much calorie you should
          consume daily which will be shown as
          <code>Daily calorie Intake</code> in your dashboard.
        </div>
      </div>
    </div>
  </div>
</div>
<!--FAQ column-->
<div class="col-md-6 text-center">
  <img
    src="static/support.png"
    class="support-desk"
  />
  <br /><br /><br />
  <button onclick="window.history.back()" class="go-back-button">
```

```
            Go back
          </button>
        </div>
      </div>
    </div>
  </section>
 </body>
</html>
```

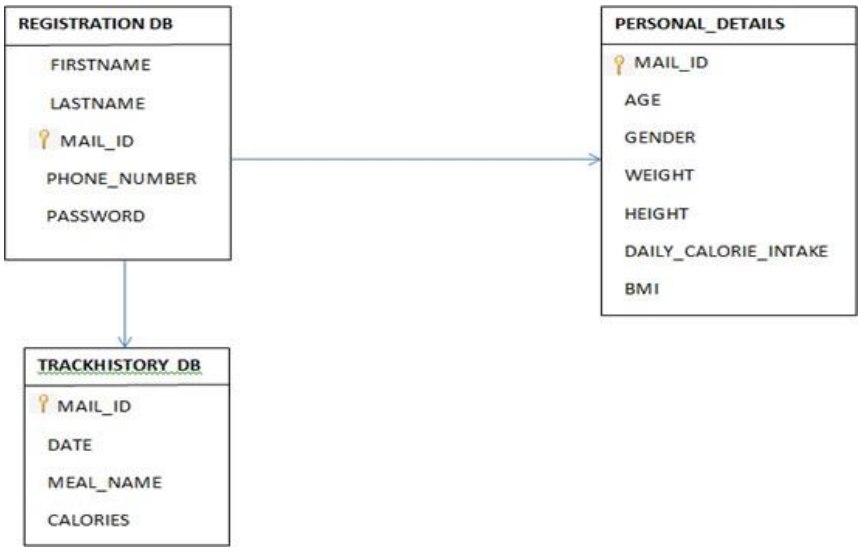
supportstyle.css:

```
 * {
  margin: 0;
  padding: 0;
 }
 body {
  background-color: #fff6bf;
 }
 .support-message {
  text-align: center;
  margin: 20px 0 !important;
  font-family: sans-serif;
  background-color: transparent;
 }
 .accordion-item,
 .accordion-header {
  background-color: transparent !important;
  border: none !important;
  padding: 0 !important;
  margin-bottom: 20px;
 }

 .accordion-button {
  width: 100%;
  height: 60px;
  background-color: rgb(224, 91, 91);
  font-weight: bold;
  text-align: left;
  text-decoration: none;

 }

 .accordion-button:active {
  background-color: rgb(224,91,91);
 }
 .support-desk {
  width: 490px;
  height: 440px;
  margin-top: 20px;
```

```
}
.go-back-button {
  color: white;
}
.go-back-button {
  background: lighten(rgb(224,91,91);, 3%);
  border: 1px solid darken(#06837f, 4%);
  box-shadow: 0px 2px 0 darken(#06837f, 5%), 2px 4px 6px darken(#06837f, 2%);
  font-weight: 900;
  letter-spacing: 1px;
  transition: all 150ms linear;
}

.go-back-button:hover {
  background: darken(#06837f, 1.5%);
  border: 1px solid rgba(#000, 0.05);
  box-shadow: 1px 1px 2px rgba(#fff, 0.2);
  color: lighten(#06837f, 18%);
  text-decoration: none;
  text-shadow: -1px -1px 0 darken(#06837f, 9.5%);
  transition: all 250ms linear;
}

/* CSS */
.go-back-button {
  background-color: #06837f;
  background-image: linear-gradient(#06837f, #329ca0);
  border: 1px solid #15aca7;
  border-radius: 4px;
  box-shadow: rgba(0, 0, 0, 0.12) 0 1px 1px;
  color: #ffffff;
  cursor: pointer;
  font-family: -apple-system, ".SFNSDisplay-Regular", "Helvetica Neue",
    Helvetica, Arial, sans-serif;
  font-size: 15px;
  margin: 0;
  outline: 0;
  padding: 11px 15px 12px;
  text-align: center;
  transition: box-shadow 0.05s ease-in-out, opacity 0.05s ease-in-out;
  user-select: none;
  -webkit-user-select: none;
  touch-action: manipulation;
  width: 30%;
}

.go-back-button:hover {
  box-shadow: rgba(24, 230, 127, 0.995) 0 0 2px inset,
    rgba(5, 197, 235, 0.818) 0 1px 2px;
```

```
    text-decoration: none;
    transition-duration: 0.15s, 0.15s;
  }

  .go-back-button:active {
    box-shadow: rgba(0, 0, 0, 0.15) 0 2px 4px inset, rgba(0, 0, 0, 0.4) 0 1px 1px;
}
```

7.3 Database Schema



8. TESTING

8.1 Test Cases

Sprint 1:

Test case ID	Test Scenario	Expected Result	Status
Home_TC_OO1	Verify user is able to see the Login , register and support button	Login , register and support button is displayed	Pass
Home_TC_OO2	Verify whether register button works	Redirected to registration page	Pass
Home_TC_OO3	Verify whether login button works	Redirected to login page	Pass
Home_TC_OO4	Verify whether support button works	Redirected to support page	Pass
Registration_TC_OO1	Verify the registration credentials vaild or not	Application should show below UI elements: a.First name and last name text box b.email text box - mandatory field c.Phone number textbox - mandatory field with maximum 10 digits allowance d.Password textbox - mandatory field with minimum 5 characters with atleast 1 alphabet and 1 number no special characters allowed e.Confirm password text box - mandatory field f.Register button	Pass
Registration_TC_OO2	Verify whether register button works	Redirects to Personal details page	Pass
Registration_TC_OO3	Verify whether the page will redirect to login page if account already registered	Redirects to Login page	Pass
Profileupdation_TC_OO1	Verify user is able to see profile updation credentials	1.Verify personal details page with below UI elements: a.Age text box - mandatory field b.Height text box - mandatory field c.Weight textbox- mandatory field d.Daily activity dropdown- mandatory field e.Proceed to dashboard button f.Go back button	Pass
Profileupdation_TC_OO2	Verify whether proceed to dashboard button works	Redirects to login page	Pass
Login_TC_OO1	Verify whether user is able to see email and password text box	User should navigate to user account homepage	Pass
Login_TC_OO2	Verify user is able to log into application with Valid credentials	Application redirects to dashboard	Pass
Login_TC_OO3	Verify user is able to log into application with InValid credentials	Application should show 'Incorrect email or password ' validation message.	Pass

Sprint 2:

Test case ID	Test Scenario	Expected Result	Status
Dashboard_TC_OO1	Verify user is able to see their height, weight, bmi and calorie intake	Verify these are available 1.Height 2.Weight 3.BMI 4.Calorie intake 5.Upload image button 6.Track history button	Pass
Dashboard_TC_OO2	Verify whether upload image button works	Redirected to upload image page	Pass
Dashboard_TC_OO3	Verify whether track history button works	Redirected to History page	Pass
Uploadimage_TC_OO1	Check the choose file option available	Able to view the 1.Choose file 2.Submit button 3.Go back button	Pass
Uploadimage_TC_OO2	Verify whether food image can be uploaded	Preview of the image uploaded will be displayed	Pass
Uploadimage_TC_OO3	Verify whether it alerts when no image is uploaded	The alert will show "Please upload the file"	Pass
Trackhistory_TC_OO1	Verify whether history table displayed	1.Date picker text box 2.Food name text box 3.Calorie text box 4.Add button 5.The track history table with date,food name and calorie value	Pass
Trackhistory_TC_OO2	Verify whether add button works	1.Chose Date 2.Enter food 3.Enter Calorie 4.Click add button 5.The data will be added to the table displayed	Pass

Sprint 3:

Test case ID	Test Scenario	Expected Result	Status
Registrationdatabase_TC_OO1	Verify whether registration credentials are added to the cloud database	The given credentials by the user should be same as the credentials stored in ibm cloud database	Pass
Personaldetailsdatabase_TC_OO1	Verify whether personal details credentials are added to the cloud database	The given credentials by the user should be same as the credentials stored in ibm cloud database	Pass
Track_historydatabase_TC_OO1	Verify whether added food details are added to the cloud database	The added food details by the user should be same as the details stored in ibm cloud database	Pass
Trackhistory_TC_OO2	Verify whether add button works	1.Chose Date 2.Enter food 3.Enter Calorie 4.Click add button 5.The data will be added to the table displayed	Pass

Sprint 4:

Test case ID	Test Scenario	Expected Result
Uploadimage_TC_OO1	Verify whether we get the nutrition values for food image uploaded	The nutrition value of the uploaded will be d
Clarify API_TC_OO2	When	

8.2 User Acceptance Testing

Sprint 1:

1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	2	3	2	4	11
Duplicate	2	0	4	0	6
External	1	1	1	1	4
Fixed	9	3	2	11	25
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	14	7	9	16	40

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Home Page	4	0	0	4
Registration Page	3	0	0	3
Profile Updation	2	0	0	2
Login Page	3	0	0	3

Sprint 2:

1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	2	3	2	4	11
Duplicate	1	0	1	0	2
External	1	1	1	1	4
Fixed	7	3	2	10	22
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	11	7	6	15	39

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Dashboard	3	0	0	3
Upload Image	4	0	0	4
TrackHistory Page	4	0	0	4

Sprint 3:

1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	3	3	2	4	12
Duplicate	2	0	3	0	5
External	1	1	1	1	4
Fixed	9	3	2	11	25
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	15	7	8	16	46

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Personal details Database	1	0	0	1
Track History Database	2	0	0	2
Registration Database	1	0	0	1
Track History Page	4	0	0	4

Sprint 4:

1. Purpose of Document

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

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	4	2	4	11
Duplicate	1	0	1	0	2
External	1	0	0	1	2
Fixed	10	3	4	10	27
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	13	7	7	15	42

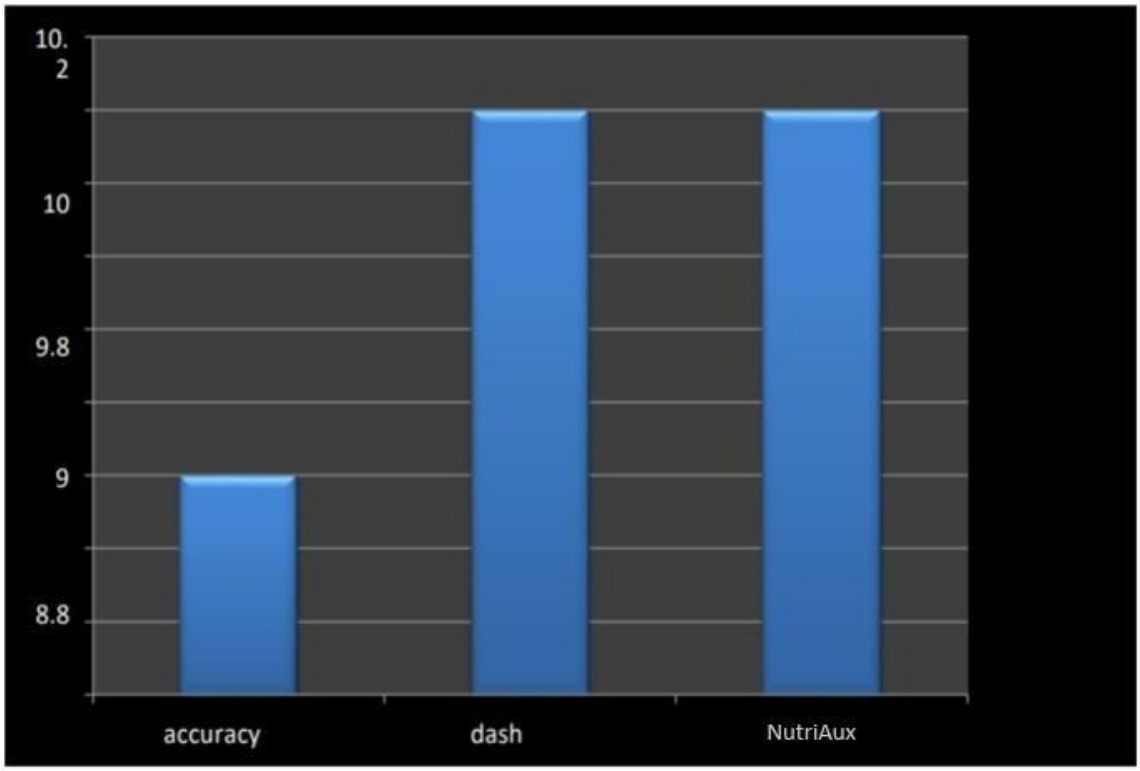
3. Test Case Analysis

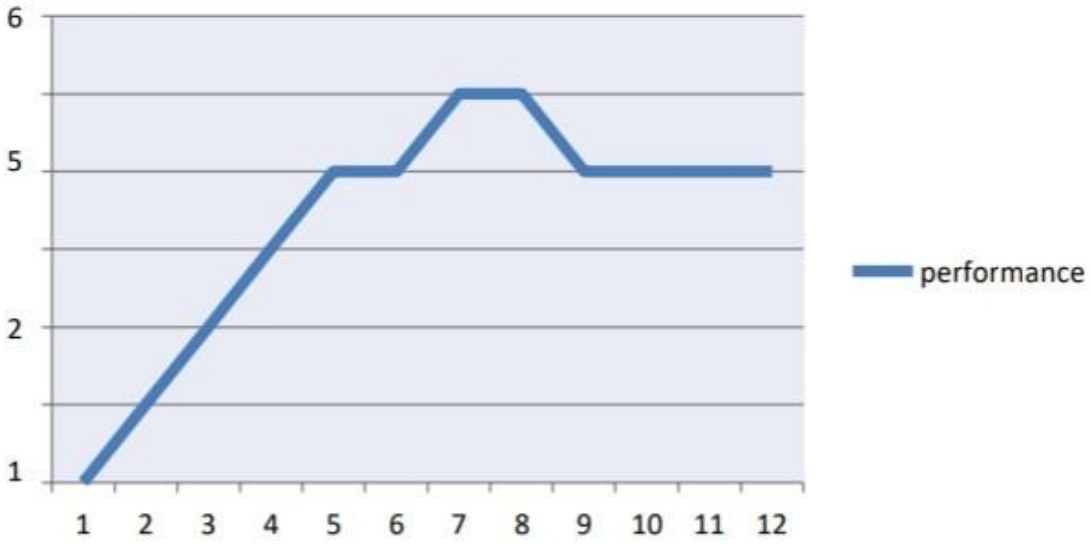
This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Upload Image	3	0	0	3
Clarifai API	1	0	0	1
Spoonacular Nutrition API	1	0	0	1

9. RESULTS

9.1 Performance Metrics





10. ADVANTAGES & DISADVANTAGES

Advantages:

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

Disadvantages:

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

11. CONCLUSION

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

12. FUTURE SCOPE

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

13. APPENDIX

Source Code

app.py :

```
from flask import Flask,render_template,request,url_for,redirect,session
from clarifai_grpc.grpc.api import service_pb2,resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2
import ibm_db
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
import requests
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc

stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())



YOUR_CLARIFAI_API_KEY="064920064d594a4fad0c07262b246a75"
YOUR_APPLICATION_ID="nutrition_assistant_app"



app=Flask(__name__)
app.secret_key='a'
try:
    conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=8e359033-a1c9-4643-82ef-
```

```
8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30120;SECURITY=SSL;S
SLServerCertificate=DigiCertGlobalRootCA.crt;UID=gkr16989;PWD=WvN7xr79Kp6YfdL7","","")
except:
    print("Unable to connect: ",ibm_db.conn_error())



@app.route("/")
def home():
    session['status_msg']=' '

    return render_template('Home.html')

@app.route("/reg")
def reg():
    return render_template('Registration.html')

@app.route("/register",methods=["POST","GET"])
def register():
     if request.method == 'POST' :
        firstName = request.form['firstName']
        lastName = request.form['lastName']

        session['email'] = request.form['email']
        phoneNumber = request.form['phoneNumber']
        password = request.form['password']
        sql = "SELECT * FROM registration WHERE EMAIL_ID=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,session['email'])
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        message =
Mail(from_email='nutriaux04@gmail.com',to_emails=session['email'],subject="NutriAux –
```

```
Registration",html_content='<b>NutriAux welcomes you</b><br/><p>Your account has been registered successfully</p>')
    try:
        #USe the API key given in this link for security purposes - https://docs.google.com/document/d/1xrF_chjAgbNJOCcsrGuVXtKWexPb5Ff5vmLpkscbgbU/edit?usp=sharing
        #sg=SendGridAPIClient('##USE THE API GIVEN IN THE ABOVE LINK##')
        response=sg.send(message)
        print(response.status_code)
        print(response.body)
        print(response.headers)
    except Exception as e:
        print(e)


    if account:


        session['status_msg']= 'Account already exists ! Kindly login'
        return redirect(url_for('login'))
    else :
        insert_sql = "INSERT INTO registration VALUES (?,?,?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, firstName)
        ibm_db.bind_param(prep_stmt, 2, lastName)
        ibm_db.bind_param(prep_stmt, 3, session['email'])
        ibm_db.bind_param(prep_stmt, 4, phoneNumber)
        ibm_db.bind_param(prep_stmt, 5, password)
        ibm_db.execute(prep_stmt)
    print('You have successfully registered !')
    return redirect(url_for('personaldetails'))
```

```
@app.route("/personaldetails")
def personaldetails():
    return render_template("personaldetails.html")

@app.route("/addpersonaldetails",methods=["POST","GET"])
def addpersonaldetails():
    if request.method == 'POST' :
        age=float(request.form.get('age'))
        gender=request.form.get('Gender')
        weight=float(request.form.get('weight'))
        height=float(request.form.get('height'))
        activity=request.form.get('activity')
        print(age,gender,weight,height,activity)
        if(gender == 'male'and activity == "1"):
            totalCalories = 1.2 * (66.5 + (13.75 * weight) + (5.003 * height) - (6.755 * age))
        elif(gender == 'male' and activity == "2"):
            totalCalories = 1.375 * (66.5 + (13.75 * weight) + (5.003 * height) - (6.755 * age))
        elif (gender == 'male' and activity == "3"):
            totalCalories = 1.55 * (66.5 + (13.75 * weight) + (5.003 * height) - (6.755 * age))
        elif(gender == 'male' and activity == "4"):
            totalCalories = 1.725 * (66.5 + (13.75 * weight) + (5.003 * height) - (6.755 * age))
        elif(gender == 'male' and activity == "5"):
            totalCalories = 1.9 * (66.5 + (13.75 * weight) + (5.003 * height) - (6.755 * age))
        elif(gender == 'female' and activity == "1"):
            totalCalories = 1.2 * (655 + (9.563 * weight) + (1.850 * height) - (4.676 * age))
        elif(gender == 'female' and activity == "2"):
            totalCalories = 1.375 * (655 + (9.563 * weight) + (1.850 * height) - (4.676 * age))
        elif(gender == 'female' and activity == "3"):
            totalCalories = 1.55 * (655 + (9.563 * weight) + (1.850 * height) - (4.676 * age))
        elif(gender == 'female' and activity == "4"):
            totalCalories = 1.725* (655 + (9.563 * weight) + (1.850 * height) - (4.676 * age))
        else:
```

```
        totalCalories = 1.9 * (655 + (9.563 * weight) + (1.850 * height) - (4.676 * age))
    print(int(totalCalories))
    BMI = (weight / (height/100)**2 )
    if BMI <= 18.5:
        BMI_message="underweight"
    elif BMI <= 24.9:
        BMI_message="healthy"
    elif BMI <= 29.9:
        BMI_message="overweight"
    else:
        BMI_message="obese"
    print(BMI)
    insert_query="INSERT INTO personal_details VALUES(?,?,?,?,?,?,?)"
    prep_stmt=ibm_db.prepare(conn,insert_query)
    ibm_db.bind_param(prep_stmt,1,session['email'])
    ibm_db.bind_param(prep_stmt,2,str(int(age)))
    ibm_db.bind_param(prep_stmt,3,gender)
    ibm_db.bind_param(prep_stmt,4,str(weight))
    ibm_db.bind_param(prep_stmt,5,str(height))
    ibm_db.bind_param(prep_stmt,6,str(totalCalories))
    ibm_db.bind_param(prep_stmt,7,str(BMI))
    ibm_db.execute(prep_stmt)
  return redirect(url_for('login'))


@app.route("/login")
def login():
  return render_template("login.html",message=session['status_msg'])


@app.route("/verify",methods=["POST","GET"])
def verify():
  session['email'] = request.form.get("email")
```

```
    password = request.form.get("password")
    get_query="SELECT * FROM registration WHERE EMAIL_ID=? AND PASSWORD=?"
    prep=ibm_db.prepare(conn,get_query)
    ibm_db.bind_param(prep,1,session['email'])
    ibm_db.bind_param(prep,2,password)
    result=ibm_db.execute(prep)
    login = ibm_db.fetch_assoc(prep)
    if login:
        get_query="SELECT weight,height,daily_calorie_intake,BMI FROM personal_details WHERE
EMAIL_ID=?"
        prep=ibm_db.prepare(conn,get_query)
        ibm_db.bind_param(prep,1,session['email'])
        result=ibm_db.execute(prep)
        data = ibm_db.fetch_tuple(prep)
        global weight
        weight= data[0]
        global height
        height= data[1]
        global daily_calorie_intake
        daily_calorie_intake=data[2]
        daily_calorie_intake=daily_calorie_intake[0:7]
        global BMI
        BMI=data[3]
        BMI=BMI[0:4]
        return redirect((url_for('dashboard')))
    print("Wrong password" , session['email'],password)
    return render_template("login.html",message="Incorrect Email ID or Password! Try again")

@app.route("/dashboard")
def dashboard():
    return
render_template('dashboard.html',weight=weight,height=height,daily_calorie_intake=daily_calorie_i
ntake,BMI=BMI)
```

```
@app.route("/upload")
def upload():
    return render_template('upload.html',calories="0 calories",fat="0 g",protein="0 g",carbs="0 g")

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

            img = request.files['file']
            print("working")

            path='./static/'+session['email']+'.jpg'
            img.save(path)

            metadata=(('authorization','Key '+YOUR_CLARIFAI_API_KEY),)
            with open(path,"rb") as f:
                file_bytes=f.read()
                request1=service_pb2.PostModelOutputsRequest(
                    model_id='9504135848be0dd2c39bdab0002f78e9',
                    inputs=[
                        resources_pb2.Input(
                            data=resources_pb2.Data(
                                image=resources_pb2.Image(
                                    base64=file_bytes
                                )
                            )
                        )
                    ])
            response =stub.PostModelOutputs(request1, metadata=metadata)
            if response.status.code != status_code_pb2.SUCCESS:
                raise Exception("Request failed, status code: " + str(response.status.code))
```

```
            for concept in response.outputs[0].data.concepts:
                print('%12s: %.2f' % (concept.name, concept.value))
            api_url = 'https://api.spoonacular.com/recipes/guessNutrition?title='
            query = response.outputs[0].data.concepts[0].name
            response = requests.get(api_url + query, headers={'X-Api-Key':
'8f123f2f983b4b69bfe1e4a25f7bfb06'})
            if response.status_code == requests.codes.ok:
                fullresponse=response.json()
                calories=str(fullresponse['calories']['value'])+' '+str(fullresponse['calories']['unit'])
                protein=str(fullresponse['protein']['value'])+' '+str(fullresponse['protein']['unit'])
                fat=str(fullresponse['fat']['value'])+' '+str(fullresponse['fat']['unit'])
                carbs=str(fullresponse['carbs']['value'])+' '+str(fullresponse['carbs']['unit'])
                print(calories,protein,fat,carbs)
                print(type(fullresponse['calories']['value']))
                print(fullresponse)
            else:
                print("Error:", response.status_code, response.text)
            return render_template('upload.html',calories=calories,fat=fat,protein=protein,carbs=carbs)
        except Exception as e:
            print(e)
        return "Error Occured"



@app.route("/history")
def history():
    get_query="SELECT CHAR(DATE_OF_CONSUMPTION,EUR)AS
DATE,MEAL_NAME,CALORIES FROM TRACKHISTORY WHERE EMAIL_ID=?"
    prep=ibm_db.prepare(conn,get_query)
    ibm_db.bind_param(prep,1,session['email'])
    result=ibm_db.execute(prep)
    if result==False:
        print("not working")
    history=[]
```

```
    dictionary = ibm_db.fetch_assoc(prep)
    while dictionary != False:
        history.insert(0,dictionary["DATE"])
        history.insert(1,dictionary["MEAL_NAME"])
        history.insert(2,dictionary["CALORIES"])

        print("The date is : ", dictionary["DATE"])
        print("The name is : ", dictionary["MEAL_NAME"])
        print("The calories is : ", dictionary["CALORIES"])
        dictionary = ibm_db.fetch_assoc(prep)
    print(history)
    no_of_rows=len(history)//3
    print(no_of_rows)
    return render_template("History.html",history=history,no_of_rows=no_of_rows)


@app.route("/addhistory",methods=["POST","GET"])
def addhistory():
    if request.method=='POST':
        date=request.form['date']
        meal_name=request.form['meal_name']
        calories=request.form['calories']
        insert_query="INSERT INTO trackhistory VALUES(?,?,?,?)"
        prep_stmt=ibm_db.prepare(conn,insert_query)
        ibm_db.bind_param(prep_stmt,1,session['email'])
        ibm_db.bind_param(prep_stmt,2,date)
        ibm_db.bind_param(prep_stmt,3,meal_name)
        ibm_db.bind_param(prep_stmt,4,calories)
        ibm_db.execute(prep_stmt)

    return redirect(url_for('history'))
@app.route("/support")
```

```
def support():
    return render_template("support.html")


if __name__ == '__main__':
    app.run(debug=True)
```

app.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-node-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: flasknode
  template:
    metadata:
      labels:
        app: flasknode
    spec:
      containers:
        - name: flasknode
          image: us.icr.io/nutritionapplication/nutriaux
          imagePullPolicy: Always
          ports:
    - containerPort: 5000
```

Docker file:

```
FROM python:3.6
WORKDIR /app
ADD . /app
COPY requirements.txt /app
RUN python3 -m pip install -r requirements.txt
RUN python3 -m pip install ibm_db
EXPOSE 5000
CMD ["python","app.py"]
```

Service.yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: flask-node-deployment
spec:
  ports:
  - port: 5000
    targetPort: 5000
  selector:
app: flasknode
```

GitHub & Project Demo Link

GitHub: https://github.com/IBM-EPBL/IBM-Project-16657-1659619614

Project Demo Link: https://youtu.be/56cue-k84B8