

# **NUTRITION ASSISTANT APPLICATION USING CLOUD APPLICATION DEVELOPMENT**

***NALAIYA THIRAN PROJECT BASED LEARNING  
ON  
PROFESSIONAL READINESS FOR INNOVATION,  
EMPLOYABILITY AND ENTREPRENEURSHIP***

## **A PROJECT REPORT**

*Submitted by*

**TEAM ID: PNT2022TMID37648**

**DEVANATHAN M(*Team lead*)**

**JAYALAKSHMI M**

**MOHANKUMAR M**

**SURIYA S**

**BACHELOR OF ENGINEERING  
IN  
COMPUTER SCIENCE AND ENGINEERING**

**ADHI COLLEGE OF ENGINEERING AND TECHNOLOGY  
SANKARAPURAM, KANCHIPURAM 631 605**



**ANNA UNIVERISTY :: CHENNAI 600 025**



**NOVEMBER 2022**

**ADHI COLLEGE OF ENGINEERING AND TECHNOLOGY  
SANKARAPURAM, KANCHIPURAM 631 605**



**FACULTY MENTOR:**

**Mrs. RADHIKA R - Asisstant professor**

Department of Computer Science and Engineering  
Adhi College of Engineering and Technology,  
Kanchipuram - 631 605

**INDUSTRY MENTOR:**

**SAI PRIYA**

# **TABLE OF CONTENTS**

## **ABSTRACT**

### **1. INTRODUCTION**

- 1.1 Project Overview
- 1.2 Purpose

### **2. LITERATURE SURVEY**

- 2.1 Existing problem
- 2.2 References
- 2.3 Problem Statement Definition

### **3. IDEATION & PROPOSED SOLUTION**

- 3.1 Empathy Map Canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution fit

### **4. REQUIREMENT ANALYSIS**

- 4.1 Functional requirement
- 4.2 Non-Functional requirements

### **5. PROJECT DESIGN**

- 5.1 Data Flow Diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

### **6. PROJECT PLANNING & SCHEDULING**

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

### **7. CODING & SOLUTIONING**

- 7.1 Feature 1
- 7.2 Feature 2
- 7.3 Database Schema (if Applicable)

## **8. TESTING**

8.1 Test Cases

8.2 User Acceptance Testing

## **9. RESULTS**

9.1 Performance Metrics

## **10.ADVANTAGES & DISADVANTAGES**

## **11.CONCLUSION**

## **12.FUTURE SCOPE**

## **13.APPENDIX**

13.1 Source code

13.2 Github & Project Demo Link

## ABSTRACT

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

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

# 1. INTRODUCTION

## 1.1 Project Overview

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

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

### Work Flow of the Project:

- User interacts with the Web App to Load an image.
- The image is passed to the server application, which uses Clarifai's AI-Driven Food Detection Model Service to analyze the images and Nutrition API to provide nutritional information about the analyzed Image.
- Nutritional information of the analyzed image is returned to the app for display.

## 1.2 Purpose

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

## **2. LITERATURE SURVEY**

### **2.1 Existing problem**

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.

The current nutrition system does not give attention to individuals body condition rather it gives general suggestions for the input given. Hence it may have chances to suggest food items that are allergic to a particular user.

### **2.2 References**

#### **Survey 1:**

Enhancing Cloud and healthy Food Nutrition Information Systems Practice-by Paul, PK and Aithal, PS and Bhuimali A.

**Year:** 2017

**Technology:** Cloud computing and Mobile Computing. Findings:

Among the common mass food information systems are not yet popularized as a domain and thus there are huge potentialities to work on this.

#### **Pros and Cons:**

Regarding manpower development there are a lot of things are pending and possible to work with. Hence cloud will do an attention on skill and manpower development for sophisticated development of food information systems.

**Survey 2:**

Mobile cloud based system recognizing nutrition and freshness of food image-by Kumbhar, Diptee and Patil, Sarita.

**Year:** 2017

**Technology:** Cloud Computing, Image Segmentation. Findings:

Mobile cloud computing (MCC) has been introduced to be a potential paradigm for mobile health services to overcome the interoperability issues over distinctive information formats. In this, we propose a mobile cloud-based food calorie measurement framework.

**Pros and cons:**

- Multiple Platform Support Cost-Efficient
- Connectivity and Performance Issues

**Survey 3:**

Predicting calorific value for mixed food using image processing-by Kohila and Meenakumari

**Year:** 2017

**Technology:** Cloud Computing, Image Segmentation.

**Findings:**

The objective of this paper is to predict and to fix diet control for various diseases by measuring the calorific value to help the patients and nutritionists. The image captured through a mobile phone/tablet camera will provide information concerning the calorie rate of the food.

**Pros and cons:**

- Increased security and reduced cost.
- Limited control and lacks support.

**Survey 4:**

Use of artificial intelligence in precision nutrition and fitness-byde Moraes Lopes, Maria Helena Baena and Ferreira, Danton Diego and Ferreira, Ana Claudia Barbosa Honorio and da Silva, Giuliano Roberto and Caetano, Aletha Silva and Braz.

**Year:** 2020



**Technology:** Artificial Intelligence, Nutritional surveillance.

**Findings:**

Among the available computational tools, artificial intelligence (AI) has gained more and more attention recently, since it is able to learn and model linear and nonlinear relationships between variables by constructing an input-output mapping such that hidden and extremely useful information for decision-making is revealed and interpreted.

**Pros and cons:**

- A large amount of data is collected by these technologies.
- AI is not yet widely used in the areas of nutrition and fitness.

## **2.3 Problem Statement Definition**

### **1. Who are all affected by this issue?**

- People from all age group who are all careless about their health due to their busy schedule and high calorie diet.
- This leads to an unhealthy lifestyle because of their eating habits.
- Thus leads to many health issues like obesity, heart attack, diabetics and rise in cholesterol level.

### **2. What are the boundaries of the problem?**

- Based on the information collected from the user, if the user is diagnosed with diabetes/Heart attack/obesity then the application provides information about diet.
- The application sets some boundaries on the user's food habits to maintain their diet and improve their condition.
- The boundaries are set on the age group of people like elder persons who have some problems with digestion so they will be provided with that information.

### **3.What is the issue?**

- Peoples are struggling to find if the packed food is good for their health or not having conflict with themselves.
- They don't know about the ingredients used in that dish and calories present in them.
- To help them to solve this problem they can take a clear picture of the food and know what are nutrients are present in that food or search for the food recipes which are suitable for them.

### **4.When does the issue occur?**

- When people want to try western culture food habits which are not suitable for our country.
- This issue will occur when people eat unhealthy food like packed or fast food because they are busy with their work and they are not giving importance to their health and food habits.
- Some people like food lovers who want to taste different dishes without knowing its effect, this leads to obesity and other health problems.

### **5.Where does the issue occur?**

- Mostly this issue occurs in developed and developing countries.
- Packed or fast food is convenient and time saving for the people who work in the IT industry.
- Slowly the intake of this food will cause to increase in insulin and cholesterol level which causes diabetes and heart attack.

### **6.Why is it important that we fix the problem?**

- This application is used to control the serious health issues before it becomes fatal.
- It helps users to improve their health and switch to a healthy lifestyle.
- For knowing what are all the ingredients present in the food and their calories present in the food they are consuming is suitable for their body condition.
- It reduces the risk of heart disease, stroke, obesity and any other health problems.



miro

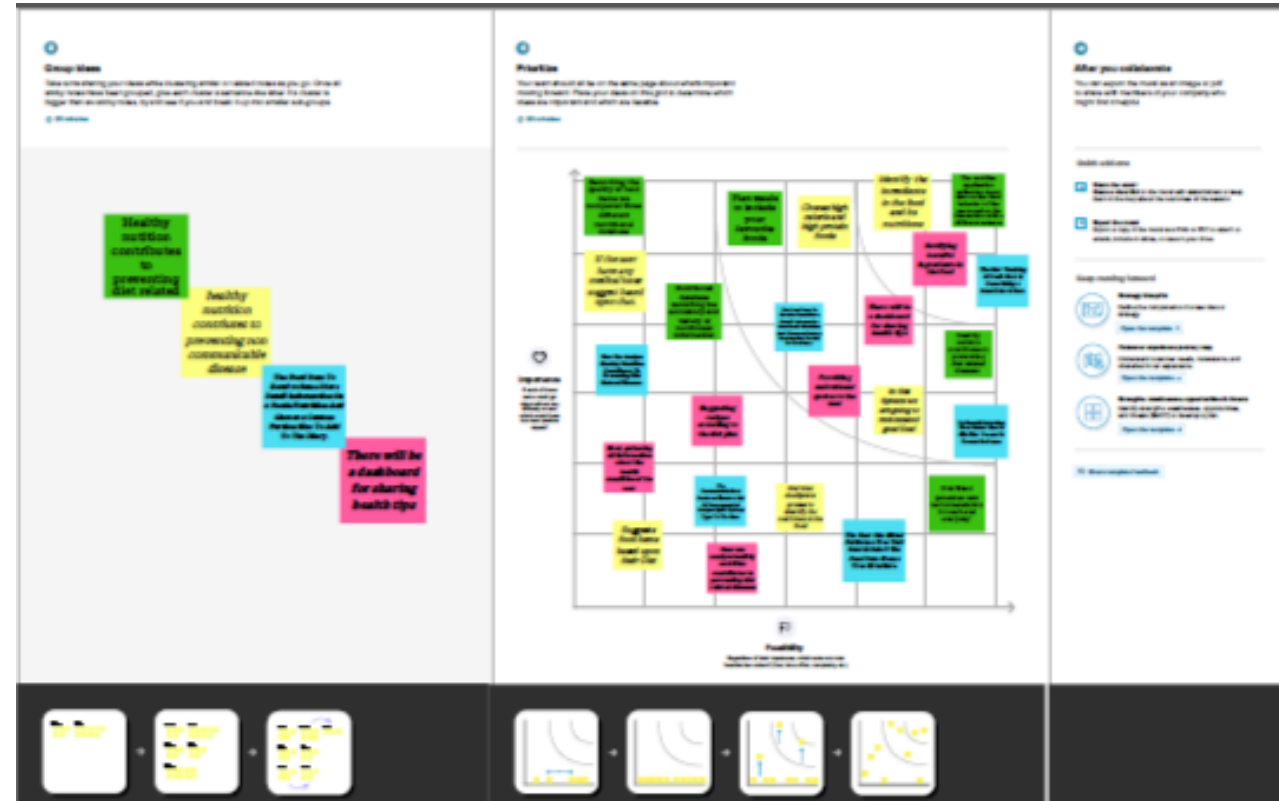
### 3. IDEATION & PROPOSED SOLUTION

#### 3.1 Empathy Map Canvas

An empathy map is used to gain deeper insights on the customer's interaction with the system. It gives an idea on what the user feels and experiences while using the system, what fears the user has regarding the system, etc. It also specifies how supportive the system environment is and what the users are likely to hear from the people around them regarding the usage of the system.



Ideation and Brainstorming are performed to generate ideas and solutions. Brainstorming is a group activity unlike ideation.



### 3.3 Proposed Solution

The project aims at developing an application that helps people to lead a healthy lifestyle by providing information about the ingredients and their nutritional content in the food they are consuming. By this people can avoid various health related issues like obesity, heart attack, diabetics, etc. Monitoring and tracking of goal and diet plans will be provided for the user based on the data collected from them.

S.No:	Parameter	Description
1.	Problem Statement (Problem to be Solved)	<ul style="list-style-type: none"><li>❖ Now a days peoples are not eating healthy foods with respect to their health condition. If it happens continuously means, it will lead to obesity and any other health problems.</li><li>❖ To avoid that the system will detect and recognize the food and evaluating the nutrient values present in the food.</li></ul>
2.	Idea / Solution Description	<ul style="list-style-type: none"><li>❖ To store the food and details of the nutrients present in it.</li><li>❖ Then scan the real time food and retrieve the corresponding food's nutrient values.</li></ul>
3.	Novelty / Uniqueness	<ul style="list-style-type: none"><li>❖ Clustering the peoples based on their BMI value.</li></ul>
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"><li>❖ The application which gives awareness among the people about the obesity and various health problems.</li></ul>
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"><li>❖ In market, this application gives a benefit across the people by health wise and economical wise.</li></ul>
6.	Scalability of the Solution	<ul style="list-style-type: none"><li>❖ The application which creates an impact among the healthy lifestyle.</li></ul>

### 3.4 Problem Solution fit

Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <b>CS</b> All age group people who are careless about their health due to their busy schedule and intake of high-calorie food like fast foods and packed foods.	<b>6. CUSTOMER CONSTRAINTS</b> <b>CC</b> If the image is not clear, the app doesn't provide accurate result. So the customer should provide a clear image for knowing the nutrition content about the food.	<b>5. AVAILABLE SOLUTIONS</b> <b>AS</b> Although the packed food comes with nutrition labels like calorie level and nutrition contents, it's still not very convenient for people to refer to App-based nutrient dashboard systems.	Explore AS, differentiate
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <b>PR</b> The problem of the user are obesity, fear of getting health related issues like heart attack, diabetes, etc... They will get frustrated of not getting immediate result and difficult to do tedious work. Sometimes they feel like lack of confidence due to their appearance.	<b>9. PROBLEM ROOT CAUSE</b> <b>RC</b> It is easy to fall into a trap of eating unhealthy foods which is heavy in calories. Once the nutritional value is replaced by foods high in sugar, bad fats and salt it leads to various health issues so users need to control their daily calorie intake to lead a healthy lifestyle.	<b>7. BEHAVIOUR</b> <b>BE</b> The behavioral changes in users reflect in their day-to-day life such as they will maintain a proper diet and follow the daily routine in eating and intake of healthy food. So, that it helps them to improve their health.	
Focus on J&P, tap into	<b>3. TRIGGERS</b> <b>TR</b> Desire to live a healthy lifestyle. By knowing the success story of people who achieved their goal. By seeing people who are fit and healthy.	<b>10. YOUR SOLUTION</b> <b>SL</b> By taking the picture of the food and uploading it in the app, the user can know what are all the nutrients present in the food. Clarifai's AI-Driven Food Detection Model is used for getting accurate identification of food and APIs to give the nutritional value of the identified food.	<b>8. CHANNELS of BEHAVIOUR</b> <b>CH</b> <b>ONLINE</b> The application provides a user friendly environment that enables users to interact through chatbot to clarify their queries and a dashboard is displayed to know the activities.  <b>OFFLINE</b> Connecting all the users through offline meeting and giving some complimentary gifts. Conducting offline session by nutrition expert.	Focus on PR, tap into BE, understand RC
	<b>4. EMOTIONS: BEFORE / AFTER</b> <b>EM</b> They scared of declining health, so they get motivated towards eating healthy foods and move to healthy lifestyle.			
Identify strong TR & EM				Extract online & offline CH of BE

## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement

#### **Project Description:**

This project is aimed at developing a desktop-based application named Nutrition Assistant Application for estimating food attributes such as ingredients and nutritional value by classifying the input images of food. The Nutrition Assistant Application refers to the system and processes to help the user to analyze the intake of food with the involvement of a technology system. This system can be used to store the details of the user's health, calculating the BMI, classifying the food image to know the nutritional value, update the status of their health condition based on the information provided, and generate health reports weekly or monthly based. This project is categorizing the individual health condition of the user. The Nutrition Assistant Application is important to control their daily calorie intake by eating healthier foods, which is the basic method to avoid obesity. Without proper diet control, this is reflective of the risks to people's health. A good Nutrition Assistant Application will alert the users when it is time to avoid. 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 the food.

#### **Scope:**

- **Maintains good health:** The application can help in guiding them on how to remain healthy and how to take good nutrition. The application will help them without personally going to the doctor. Promote better nutrition in the community by educating about their diet and nutrition.
- **Functional limitation:** The user to be specific can't access the web or admin module, whereas the administrator has all the rights to modify and manage the contents such as news, tips, etc.
- **Improve Usability:** On the part of user's just the internet connection is enough in order to access the news, updates and other contents provided by the admin regarding their health condition.
- **Health Conscious:** This will provide convenience to users who wants to learn about nutrition and other related health topics by using the Nutrition Assistant Application.

## Purpose:

The user continue to demand to know the nutritional value that is in their food. The users learn about the effect of different foods on human health. Evidently, the ultimate aim of this application is to provide the ways in which one can lead a healthy life by maintaining diet. The user can access the nutritional information by taking a photo of the food, uploading a photo from the gallery, or by entering manually.

Nutrition is more than just obtaining nutrients and calories from food. Its more than just eating the healthy stuff. Its more than just following the most recent diet. Nutrition, the food we eat and the way we eat it, is an integral part of life. Nutrition is an experience. It evokes memories, helps us celebrate good times, and is there for us in times of grief. I believe the purpose of nutrition is to nourish the body and soul.

The Nutrition Assistant Application helps the users to eat nutritional rich food which yields to lead a healthy life.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Data collection	Collection of required input data
FR-4	Data analysis	Process the given inputs using CNN and Nutrion API
FR-5	Data processing	Evaluate the data and store it in database and integrate in cloud containers
FR-6	Provide output to user	Display the result to the user



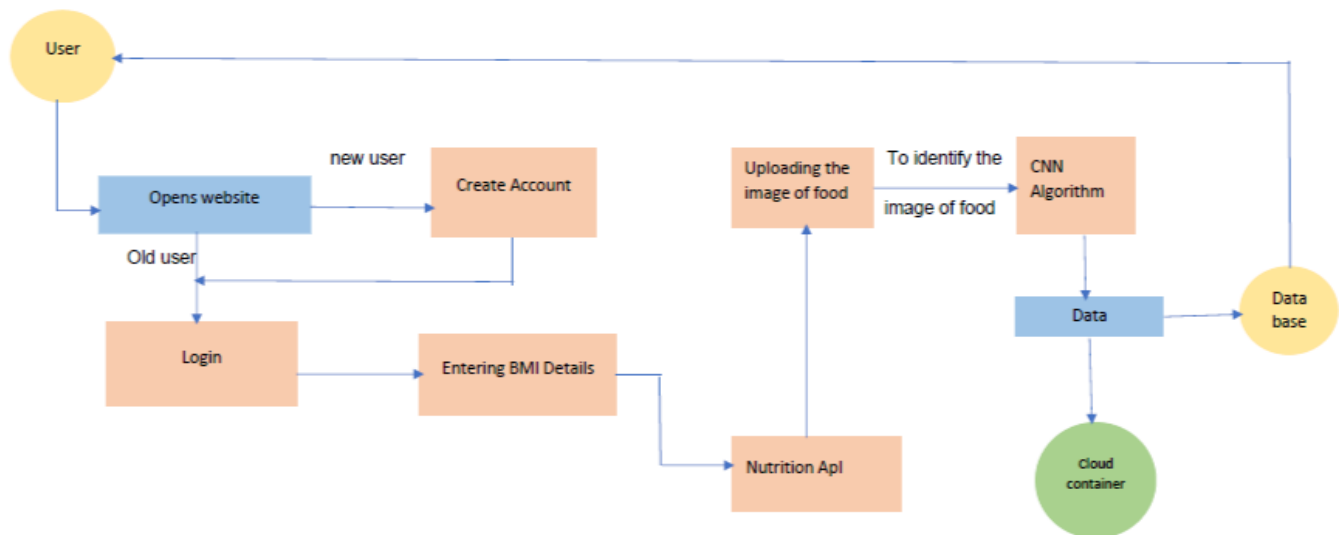
## 4.2 Non-Functional requirements

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

## 5. PROJECT DESIGN

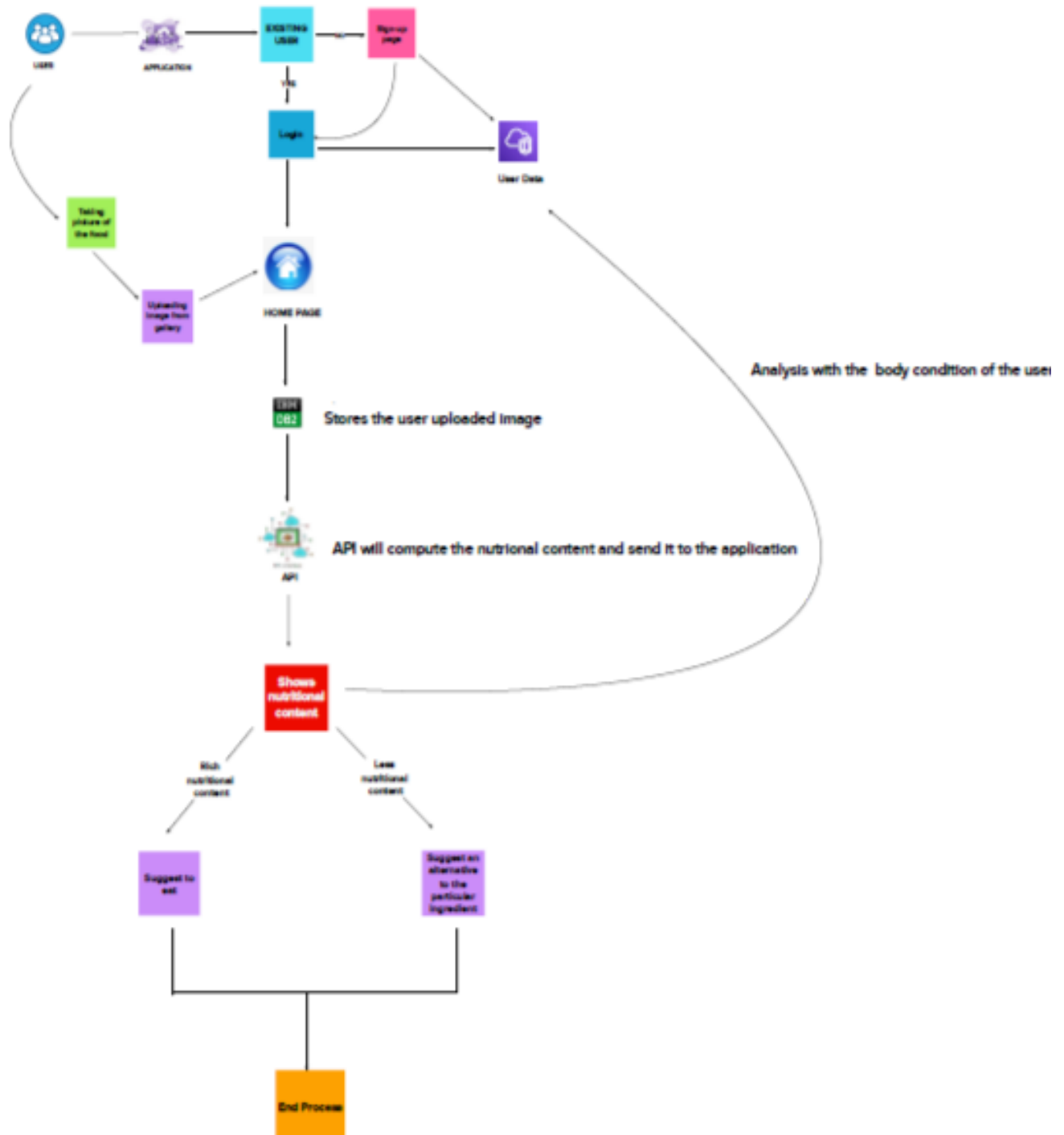
### 5.1 Data Flow Diagrams

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

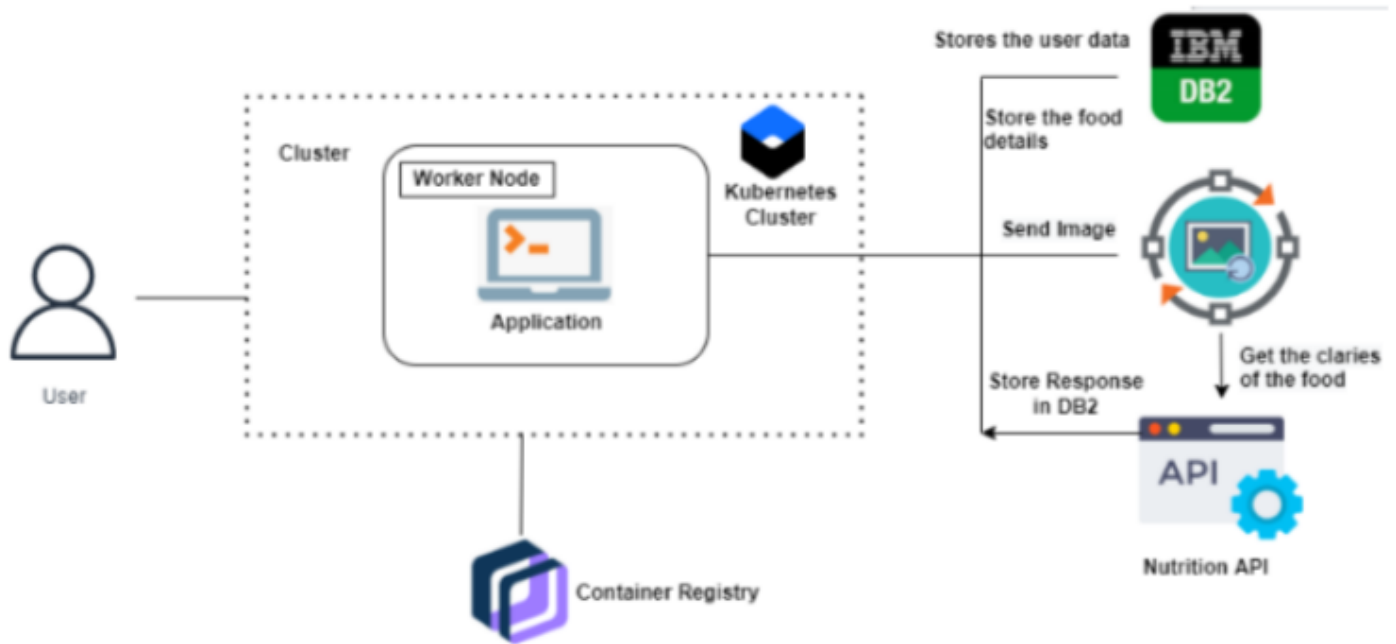


## 5.2 Solution & Technical Architecture

### Solution Architecture



## Technical Architecture



### 5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User(All common people)	User Registration	USN-1	As a user, I can register for the application by entering my name,email, password.	I can access my dashboard.	High	Sprint-1
	Login	USN-2	As a user, I can login to the application using my given credentials.	I can access my dashboard.	High	Sprint-1
	BMI Calculation	USN-3	As a user, I enter my height and weight details.	I can get to know about my BMI	High	Sprint-1
	Uploading the Image	USN-4	As a user, I will upload the image of food that I want to eat.	I can upload the image to decide whether to eat or not.	High	Sprint-1
	Providing output to user	USN-5	As a user, I will get to know the results of the inputs I've given.	I will get to know if I can eat the food or not.	Medium	Sprint-2
Administrator	Data Analysis	USN-6	As an admin, I will develop algorithms and modules to process the data.	I can store the result in database	High	Sprint-1
	Integrating with Cloud	USN-7	As a admin, I integrate the results in cloud containers.	I can deploy the data in cloud.	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
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High
Sprint-2	Profile Update	USN-3	As a user, I have to enter my height, weight and daily activity details.	2	High
Sprint-3	Login	USN-4	As a user, I can login to the application by entering e-mail and password	2	High
Sprint-3		USN-5	As a user, I can reset my password if I forget my password	1	Medium
Sprint-4	Dashboard	USN-6	As a user, I can upload or capture live image of the food.	2	High
Sprint-4	Dashboard	USN-7	As a user, I can know the nutrition content in the food.	1	Medium
Sprint-4		USN-8	As a user, I can track my daily calorie intake.	1	Medium
Sprint -4		USN-9	Maintaining detail for user	1	High

### 6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	7	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	5	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	5	19 Nov 2022

## Velocity:

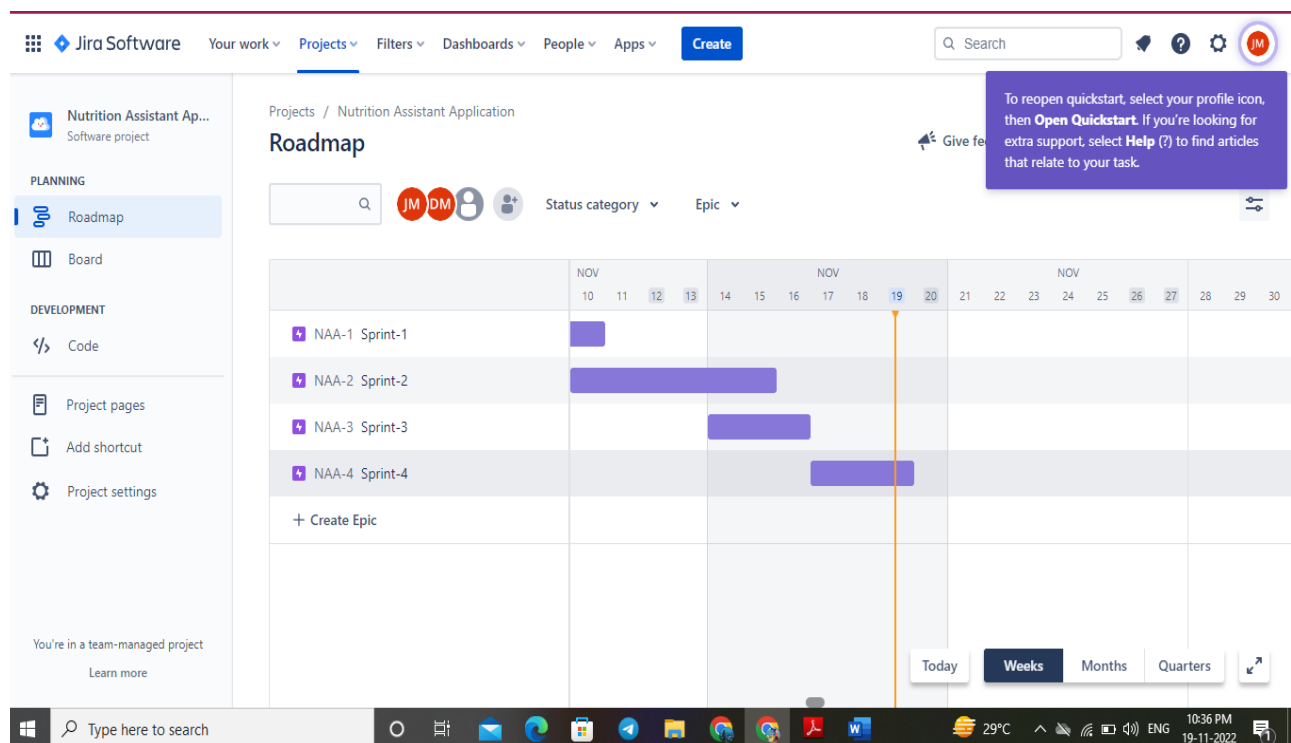
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

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

## Burndown Chart:

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

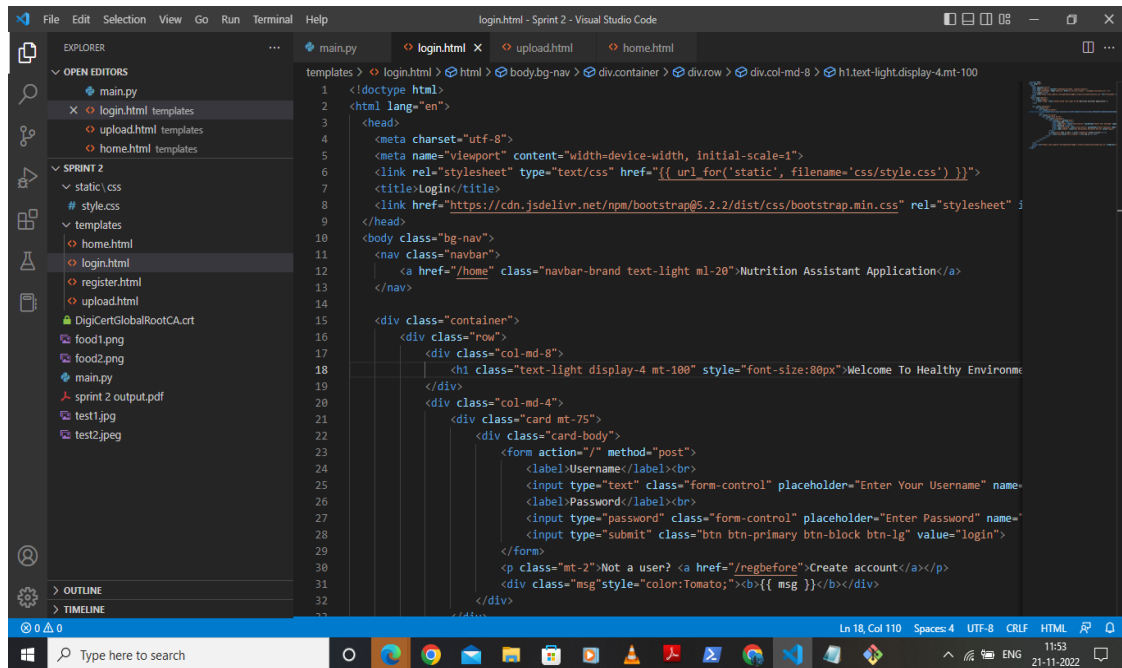




## 7. CODING & SOLUTIONING

### 7.1 Feature

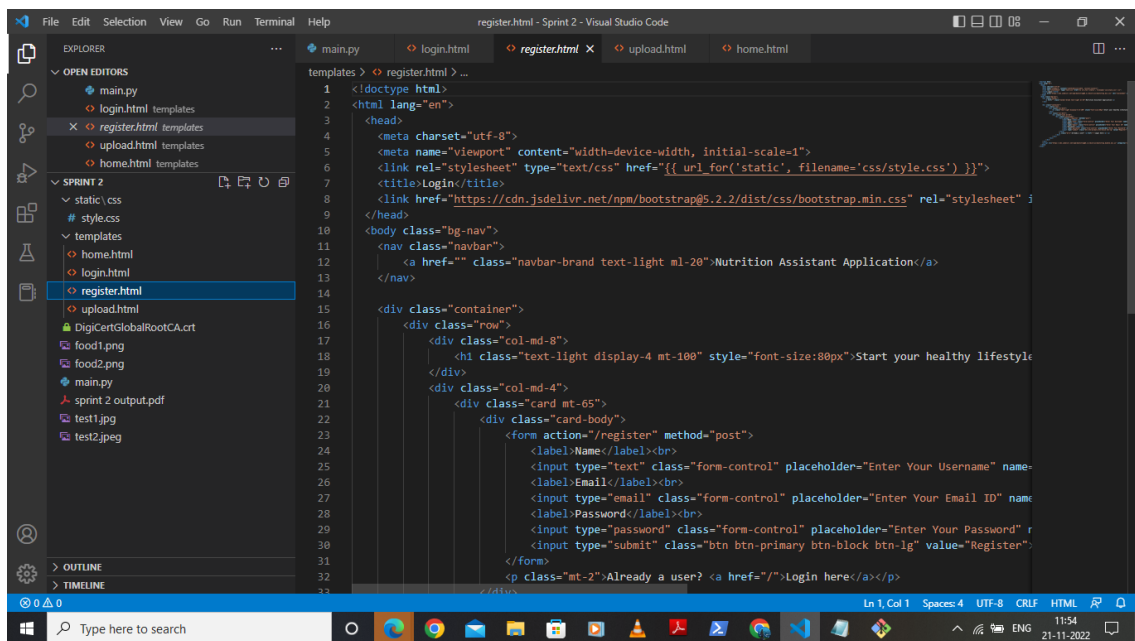
#### Login page



The screenshot shows the Visual Studio Code editor with the 'login.html' file open. The Explorer panel on the left shows the project structure, including 'main.py', 'login.html', 'upload.html', and 'home.html'. The main editor area displays the HTML code for the login page, which includes a Bootstrap 5.2.2 layout with a login form and a 'Welcome To Healthy Environment' message.

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}">
7     <title>Login</title>
8     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">
9   </head>
10  <body class="bg-nav">
11    <nav class="navbar">
12      <a href="/home" class="navbar-brand text-light ml-20">Nutrition Assistant Application</a>
13    </nav>
14
15    <div class="container">
16      <div class="row">
17        <div class="col-md-8">
18          <h1 class="text-light display-4 mt-100" style="font-size:80px">Welcome To Healthy Environment
19        </div>
20        <div class="col-md-4">
21          <div class="card mt-75">
22            <div class="card-body">
23              <form action="/" method="post">
24                <label>Username</label><br>
25                <input type="text" class="form-control" placeholder="Enter Your Username" name="username">
26                <label>Password</label><br>
27                <input type="password" class="form-control" placeholder="Enter Password" name="password">
28                <input type="submit" class="btn btn-primary btn-block btn-lg" value="login">
29              </form>
30              <p class="mt-2">Not a user? <a href="/regbefore">Create account</a></p>
31              <div class="msg" style="color:tomato;"><b>{{ msg }}</b></div>
32            </div>
33          </div>
34        </div>
35      </div>
36    </div>
37  </body>
38 </html>
```

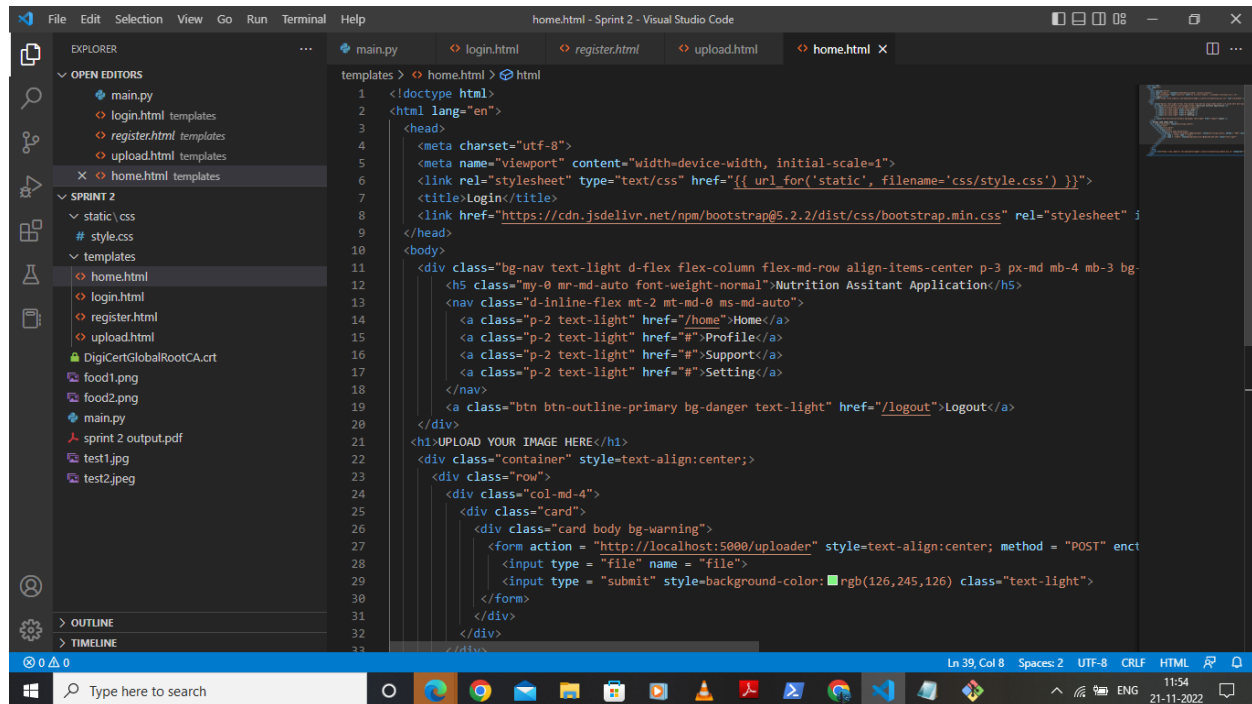
#### Register page



The screenshot shows the Visual Studio Code editor with the 'register.html' file open. The Explorer panel on the left shows the project structure, including 'main.py', 'login.html', 'upload.html', and 'home.html'. The main editor area displays the HTML code for the register page, which includes a Bootstrap 5.2.2 layout with a registration form and a 'Start your healthy lifestyle' message.

```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}">
7     <title>Login</title>
8     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">
9   </head>
10  <body class="bg-nav">
11    <nav class="navbar">
12      <a href="/" class="navbar-brand text-light ml-20">Nutrition Assistant Application</a>
13    </nav>
14
15    <div class="container">
16      <div class="row">
17        <div class="col-md-8">
18          <h1 class="text-light display-4 mt-100" style="font-size:80px">Start your healthy lifestyle
19        </div>
20        <div class="col-md-4">
21          <div class="card mt-65">
22            <div class="card-body">
23              <form action="/register" method="post">
24                <label>Name</label><br>
25                <input type="text" class="form-control" placeholder="Enter Your Username" name="name">
26                <label>Email</label><br>
27                <input type="email" class="form-control" placeholder="Enter Your Email ID" name="email">
28                <label>Password</label><br>
29                <input type="password" class="form-control" placeholder="Enter Your Password" name="password">
30                <input type="submit" class="btn btn-primary btn-block btn-lg" value="Register">
31              </form>
32              <p class="mt-2">Already a user? <a href="/">Login here</a></p>
33            </div>
34          </div>
35        </div>
36      </div>
37    </div>
38  </body>
39 </html>
```

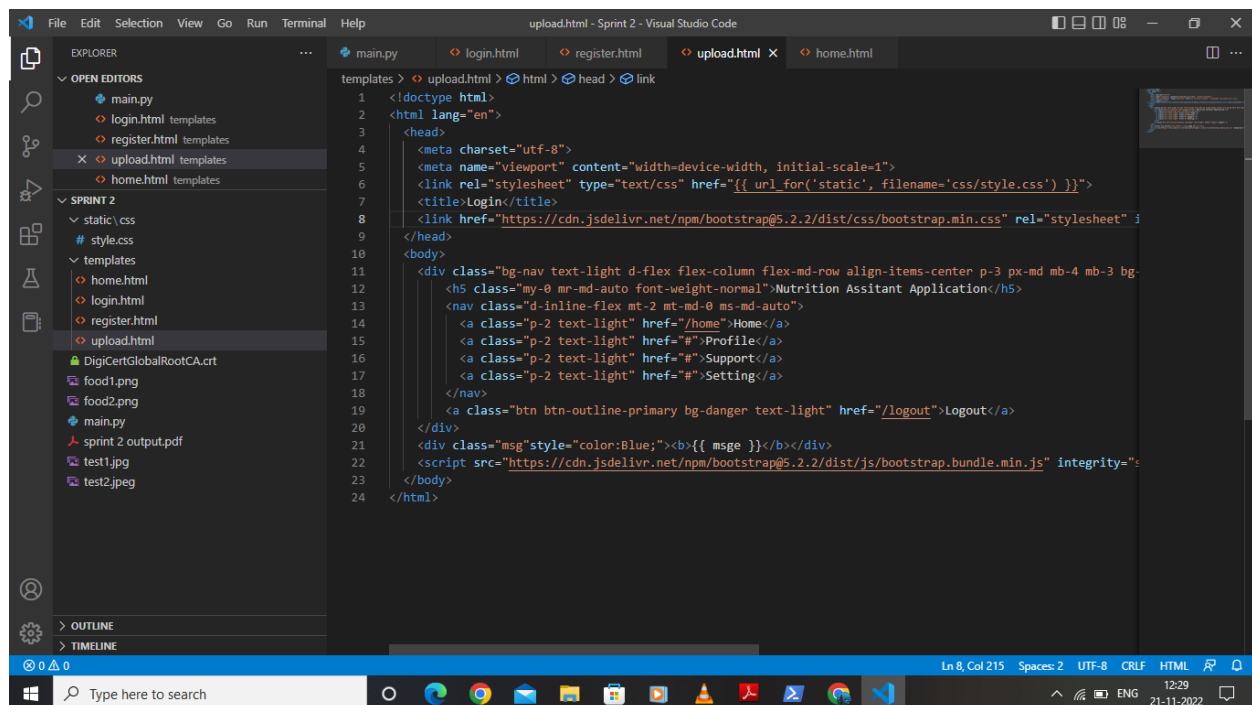
## Home Page



The screenshot shows the Visual Studio Code editor with the 'home.html' file open. The Explorer panel on the left shows the project structure, including 'main.py', 'login.html', 'register.html', 'upload.html', and 'home.html'. The main editor displays the HTML code for the home page, which includes a navigation bar, a login form, and a file upload section.

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}">
7   <title>Login</title>
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">
9 </head>
10 <body>
11   <div class="bg-nav text-light d-flex flex-column flex-md-row align-items-center p-3 px-md-4 mb-3 bg">
12     <h5 class="my-0 mr-md-auto font-weight-normal">Nutrition Assistant Application</h5>
13     <nav class="d-inline-flex mt-2 mt-md-0 ms-md-auto">
14       <a class="p-2 text-light" href="/home">Home</a>
15       <a class="p-2 text-light" href="#">Profile</a>
16       <a class="p-2 text-light" href="#">Support</a>
17       <a class="p-2 text-light" href="#">Setting</a>
18     </nav>
19     <a class="btn btn-outline-primary bg-danger text-light" href="/logout">Logout</a>
20   </div>
21   <h1>UPLOAD YOUR IMAGE HERE</h1>
22   <div class="container" style=text-align:center;>
23     <div class="row">
24       <div class="col-md-4">
25         <div class="card">
26           <div class="card body bg-warning">
27             <form action = "http://localhost:5000/uploaden" style=text-align:center; method = "POST" enc
28               <input type = "file" name = "file">
29               <input type = "submit" style=background-color:rgb(126,245,126) class="text-light">
30             </form>
31           </div>
32         </div>
33       </div>
34     </div>
35   </div>
```

## Upload Page

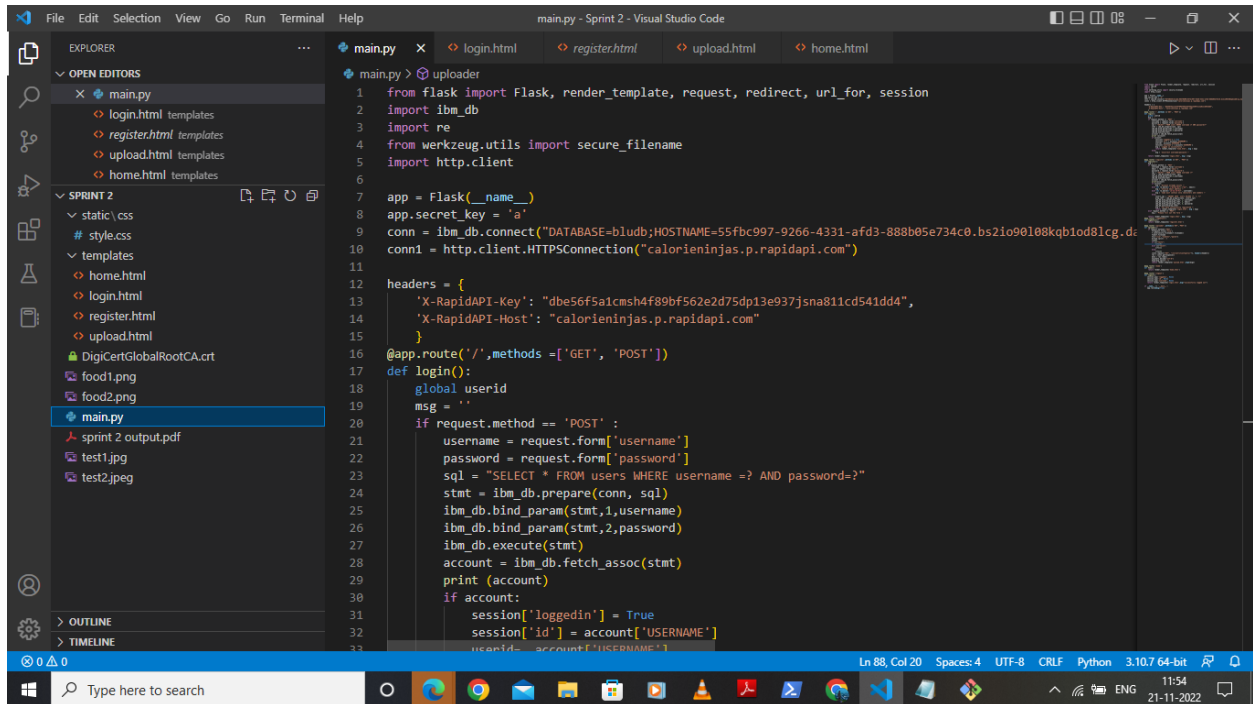


The screenshot shows the Visual Studio Code editor with the 'upload.html' file open. The Explorer panel on the left shows the project structure, including 'main.py', 'login.html', 'register.html', 'upload.html', and 'home.html'. The main editor displays the HTML code for the upload page, which includes a navigation bar, a login form, and a file upload section.

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}">
7   <title>Login</title>
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">
9 </head>
10 <body>
11   <div class="bg-nav text-light d-flex flex-column flex-md-row align-items-center p-3 px-md-4 mb-3 bg">
12     <h5 class="my-0 mr-md-auto font-weight-normal">Nutrition Assistant Application</h5>
13     <nav class="d-inline-flex mt-2 mt-md-0 ms-md-auto">
14       <a class="p-2 text-light" href="/home">Home</a>
15       <a class="p-2 text-light" href="#">Profile</a>
16       <a class="p-2 text-light" href="#">Support</a>
17       <a class="p-2 text-light" href="#">Setting</a>
18     </nav>
19     <a class="btn btn-outline-primary bg-danger text-light" href="/logout">Logout</a>
20   </div>
21   <div class="msg" style="color:Blue;"><b>{{ msge }}</b></div>
22   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="s
23 </body>
24 </html>
```

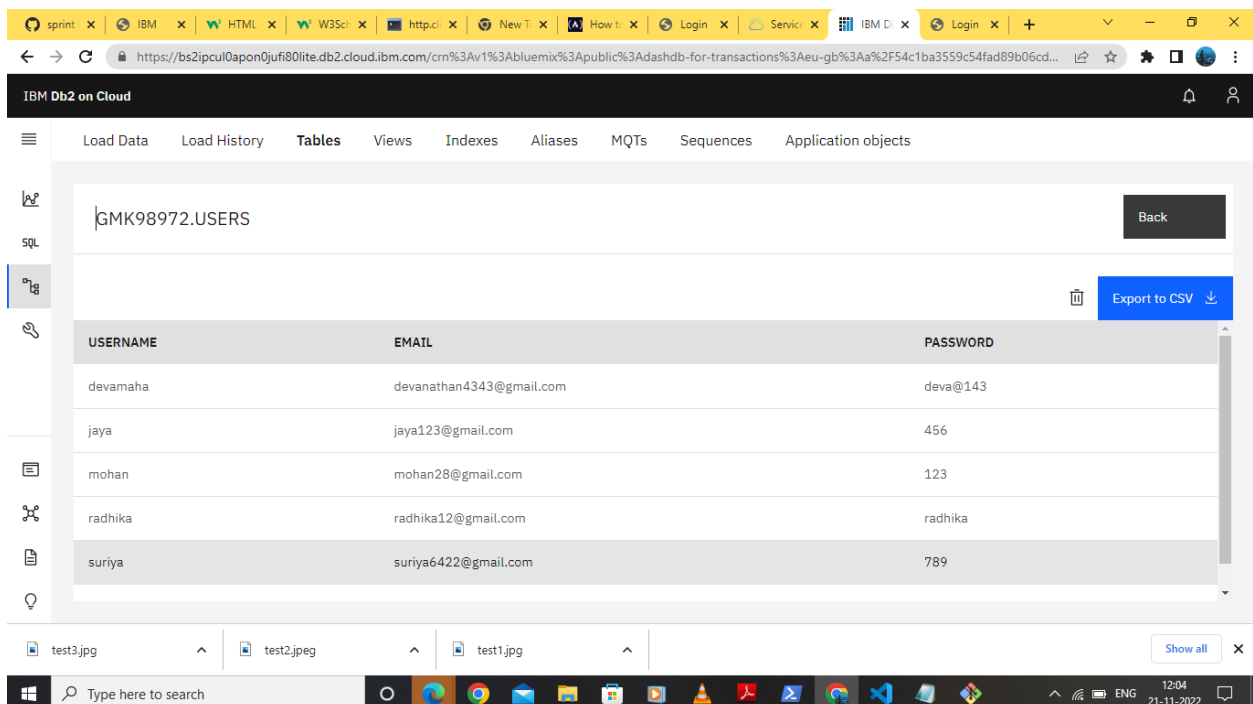


## 7.2 Feature 2



```
1 from flask import Flask, render_template, request, redirect, url_for, session
2 import ibm_db
3 import re
4 from werkzeug.utils import secure_filename
5 import http.client
6
7 app = Flask(__name__)
8 app.secret_key = 'a'
9 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kbiod8l1cg.db2cloud.com;UID=ibmadmin;PWD=ibmadmin")
10 conn1 = http.client.HTTPSConnection("calorieninjas.p.rapidapi.com")
11
12 headers = {
13     'X-RapidAPI-Key': "d5e56f5a1cmsh4f89bf562e2d75dp13e937jsna811cd541dd4",
14     'X-RapidAPI-Host': "calorieninjas.p.rapidapi.com"
15 }
16 @app.route('/', methods=['GET', 'POST'])
17 def login():
18     global userid
19     msg = ''
20     if request.method == 'POST':
21         username = request.form['username']
22         password = request.form['password']
23         sql = "SELECT * FROM users WHERE username =? AND password=?"
24         stmt = ibm_db.prepare(conn, sql)
25         ibm_db.bind_param(stmt, 1, username)
26         ibm_db.bind_param(stmt, 2, password)
27         ibm_db.execute(stmt)
28         account = ibm_db.fetch_assoc(stmt)
29         print (account)
30         if account:
31             session['loggedin'] = True
32             session['id'] = account['USERNAME']
33             return redirect(url_for('home'))
```

## 7.3 Database schema



USERNAME	EMAIL	PASSWORD
devamaha	devanathan4343@gmail.com	deva@143
jaya	jaya123@gmail.com	456
mohan	mohan28@gmail.com	123
radhika	radhika12@gmail.com	radhika
suriya	suriya6422@gmail.com	789

## 8. TESTING

### Purpose of Document

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3

### Defect Analysis

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

## Test Case Analysis

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

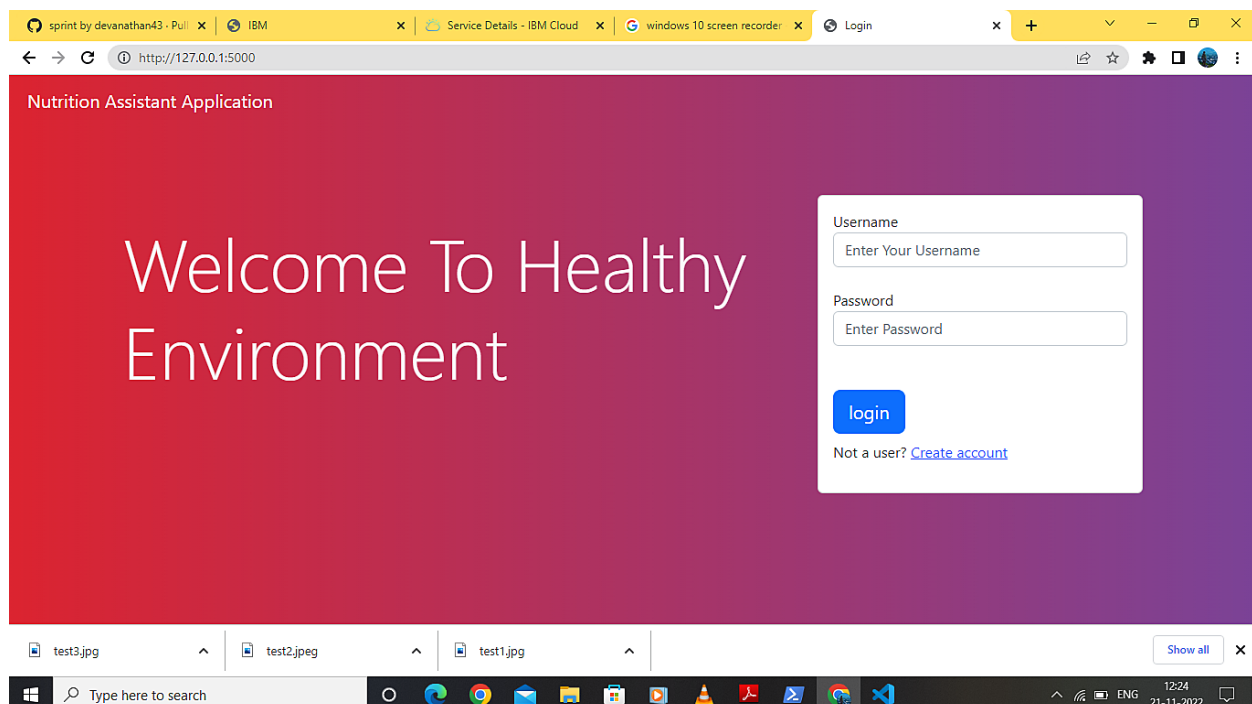
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

## 9. RESULTS

### 9.1 Performance metrics

Metrics are the base line of performance tests. Monitoring the correct parameters will help you to detect the areas that requires more attention and finds ways to improve them.

### Login Page



## Importing an image

sprint by devanathan43 - Pull x IBM x Service Details - IBM Cloud x windows 10 screen recorder x Login x

← → ↻ http://127.0.0.1:5000

Nutrition Assitant Application Home Profile Support Setting Logout

### UPLOAD YOUR IMAGE HERE

Choose File No file chosen Submit



## Output Values

sprint by devanathan43 - Pull x IBM x Service Details - IBM Cloud x windows 10 screen recorder x Login x

← → ↻ http://localhost:5000/uploader

Nutrition Assitant Application Home Profile Support Setting Logout

```
[{"items": [{"sugar_g": 0.6, "fiber_g": 1.8, "serving_size_g": 100.0, "sodium_mg": 0, "name": "noodles", "potassium_mg": 57, "fat_saturated_g": 0.2, "fat_total_g": 0.9, "calories": 161.8, "cholesterol_mg": 0, "protein_g": 5.8, "carbohydrates_total_g": 31.2}]}]
```



## **10. ADVANTAGES AND DISADVANTAGES**

### **10.1 Advantages**

- By using this application we can avoid junk foods.
- We will get nutrition values of various foods.
- Easy and accessible user interface.

### **10.2 Disadvantages**

- Works only with the pretrained images.
- The image to be uploaded should be clear and in well good clarity.
- Requires some time for scanning before eating the food.

## **11. Conclusion**

This Application takes image as input and it analyzes the input image through Clarifai AI driven food detection model. The output of the detection model is sent to the RapidApi-NutritionApi that will give the nutritional values of the corresponding food image. The registered user information are stored in the IBM\_DB2 database.

## **12. Future scope**

In future this application could be trained with large number of food images to detect a wide range of foods and its corresponding nutrients.

## 13. APPENDIX

### 13.1 Source code

#### login.html

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}">
    <title>Login</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">
  </head>
  <body class="bg-nav">
    <nav class="navbar">
      <a href="/home" class="navbar-brand text-light ml-20">Nutrition Assistant Application</a>
    </nav>

    <div class="container">
      <div class="row">
        <div class="col-md-8">
          <h1 class="text-light display-4 mt-100" style="font-size:80px">Start your healthy
lifestyle here</h1>
        </div>
        <div class="col-md-4">
          <div class="card mt-75">
            <div class="card-body">
```

```

        <form action="/" method="post">
            <label>Username</label><br>
            <input type="text" class="form-control" placeholder="Enter Your Username"
name="username"><br>
            <label>Password</label><br>
            <input type="password" class="form-control" placeholder="Enter Password"
name="password"><br><br>
            <input type="submit" class="btn btn-primary btn-block btn-lg" value="login">
        </form>
        <p class="mt-2">Not a user? <a href="/regbefore">Create account</a></p>
        <div class="msg" style="color:Tomato;"><b>{{ msg }}</b></div>
    </div>
</div>
</div>
</div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
</body>
</html>

```

### resigter.html

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}">
    <title>Login</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">
</head>
<body class="bg-nav">
  <nav class="navbar">
    <a href="" class="navbar-brand text-light ml-20">Nutrition Assistant Application</a>
  </nav>

  <div class="container">
    <div class="row">
      <div class="col-md-8">
        <h1 class="text-light display-4 mt-100" style="font-size:80px">Start your healthy
lifestyle here</h1>
      </div>
      <div class="col-md-4">
        <div class="card mt-65">
          <div class="card-body">
            <form action="/register" method="post">
              <label>Name</label><br>
              <input type="text" class="form-control" placeholder="Enter Your Username"
name="username" required><br>
              <label>Email</label><br>
              <input type="email" class="form-control" placeholder="Enter Your Email ID"
name="email" required><br>
              <label>Password</label><br>
              <input type="password" class="form-control" placeholder="Enter Your Password"
name="password" required><br><br>
              <input type="submit" class="btn btn-primary btn-block btn-lg" value="Register">
            </form>
            <p class="mt-2">Already a user? <a href="/">Login here</a></p>
          </div>
        </div>
      </div>
    </div>
  </div>
```



```
        </div>
    </div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
</body>
</html><!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}">
<title>Login</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">
</head>
<body class="bg-nav">
<nav class="navbar">
<a href="" class="navbar-brand text-light ml-20">Nutrition Assistant Application</a>
</nav>
<div class="container">
<div class="row">
<div class="col-md-8">
<h1 class="text-light display-4 mt-100" style="font-size:80px">Start your healthy
lifestyle here</h1>
</div>
<div class="col-md-4">
<div class="card mt-65">
<div class="card-body">
```

```

<form action="/register" method="post">
<label>Name</label><br>
<input type="text" class="form-control" placeholder="Enter Your Username"
name="username" required><br>
<label>Email</label><br>
<input type="email" class="form-control" placeholder="Enter Your Email ID"
name="email" required><br>
<label>Password</label><br>
<input type="password" class="form-control" placeholder="Enter Your Password"
name="password" required><br><br>
<input type="submit" class="btn btn-primary btn-block btn-lg" value="Register">
</form>
<p class="mt-2">Already a user? <a href="/">Login here</a></p>
</div>
</div>
</div>
</div>
</div>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
</body>
</html>

```

## home.html

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}">

```

```

<title>Login</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">
<script>
$(document).ready(function(){
    $('#input[type="file"]').change(function(e){
        var fileName = e.target.files[0].name;
        document.getElementById("demo").innerHTML =
            'The file "' + fileName + '" has been selected.';
    });//from w ww . j a v a 2 s . c o m
    });
</script>
</head>
<body>
<div class="bg-nav text-light d-flex flex-column flex-md-row align-items-center p-3 px-md mb-4
mb-3 bg-white border-bottom">
    <h5 class="my-0 mr-md-auto font-weight-normal">Nutrition Assitant Application</h5>
    <nav class="d-inline-flex mt-2 mt-md-0 ms-md-auto">
        <a class="p-2 text-light" href="/home">Home</a>
        <a class="p-2 text-light" href="#">Profile</a>
        <a class="p-2 text-light" href="#">Support</a>
        <a class="p-2 text-light" href="#">Setting</a>
    </nav>
    <a class="btn btn-outline-primary bg-danger text-light" href="/logout">Logout</a>
</div>
<h1>Upload your image here</h1>
<div class="container" style=text-align:center;>
    <div class="row">
        <div class="col-md-4">
            <div class="card">
                <div class="card body bg-warning">

```

```

        <form action = "http://localhost:5000/uploader" style=text-align:center; method = "POST"
enctype = "multipart/form-data">
        <input type = "file" name = "file">
        <input type = "submit" style=background-color:Green class="text-light">
    </form>
</div>
</div>
</div>
</div>
</div>
</div>
</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>
</body>
</html>

```

### upload.html

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css') }}">
    <title>Login</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">
</head>
<body>

```

```

<div class="bg-nav text-light d-flex flex-column flex-md-row align-items-center p-3 px-md mb-4
mb-3 bg-white border-bottom">
  <h5 class="my-0 mr-md-auto font-weight-normal">Nutrition Assitant Application</h5>
  <nav class="d-inline-flex mt-2 mt-md-0 ms-md-auto">
    <a class="p-2 text-light" href="/home">Home</a>
    <a class="p-2 text-light" href="#">Profile</a>
    <a class="p-2 text-light" href="#">Support</a>
    <a class="p-2 text-light" href="#">Setting</a>
  </nav>
  <a class="btn btn-outline-primary bg-danger text-light" href="/logout">Logout</a>
</div>
<div class="msg"style="color:Tomato;"><b>{{ msge }}</b></div>
<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>
</body>
</html>

```

### style.css

```

..bg-nav{
  background: #12ca3a; /* fallback for old browsers */
  background: -webkit-linear-gradient(to right, #dc2430, #7b4397); /* Chrome 10-25, Safari 5.1-
6 */
  background: linear-gradient(to right, #dc2430, #7b4397); /* W3C, IE 10+/ Edge, Firefox 16+,
Chrome 26+, Opera 12+, Safari 7+ */
}
.mt-100{
  margin-top:100px;
}
.ml-20{
  margin-left:20px;
}

```

```

.mt-75{
    margin-top:75px;
}
.mt-65{
    margin-top:65px;
}
.mt-200{
    margin-top:200px;
}
.button1{
    background-color: #4CAF50;
}
.ml-200{
    margin-top:200px;
}

```

### **main.py**

```

from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
from werkzeug.utils import secure_filename
import http.client

app = Flask(__name__)
app.secret_key = 'a'
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31505;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=jjt02162;PWD=xVTBdOcHSV6GmZpV",",")
conn1 = http.client.HTTPSConnection("calorieninjas.p.rapidapi.com")
headers = {
    'X-RapidAPI-Key': "dbe56f5a1cmsh4f89bf562e2d75dp13e937jsna811cd541dd4",
    'X-RapidAPI-Host': "calorieninjas.p.rapidapi.com"
}

```

```
}
```

```
@app.route('/', methods = ['GET', 'POST'])
def login():
    global userid
    msg = ""
    if request.method == 'POST' :
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM users WHERE username =? AND password=?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print (account)
        if account:
            session['loggedin'] = True
            session['id'] = account['USERNAME']
            userid= account['USERNAME']
            session['username'] = account['USERNAME']
            msg = 'Logged in successfully !'
            return render_template('home.html', msg = msg)
        else:
            msg = 'Incorrect username/password !'

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

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

```

email = request.form['email']
password = request.form['password']
sql = "SELECT * FROM users WHERE username =?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,username)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print(account)
if account:
    msg = 'Account already exists !'
elif not re.match(r'^@]+@[^@]+\.[^@]+', email):
    msg = 'Invalid email address !'
elif not re.match(r'[A-Za-z0-9]+', username):
    msg = 'name must contain only characters and numbers !'
else:
    insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prepare_stmt, 1, username)
    ibm_db.bind_param(prepare_stmt, 2, email)
    ibm_db.bind_param(prepare_stmt, 3, password)
    ibm_db.execute(prepare_stmt)
    msg = 'You have successfully registered !'
    return render_template('login.html', msg = msg)
elif request.method == 'POST':
    msg = 'Please fill out the form !'

    return render_template('login.html', msg = msg)
@app.route('/regbefore')
def regbefore():
    return render_template('register.html')

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

```



```

if request.method=='POST':
    f=request.files['file']
    f.save(secure_filename(f.filename))
    food=f.filename
    f1=["rice","noodles","pasta"]
    d=food.split('.')
    s=d[0]
    if s=="food1":
        s=f1[0]
    elif s=="food2":
        s=f1[1]
    else:
        s=f1[2]
    conn1.request("GET", "/v1/nutrition?query="+s, headers=headers)
    res = conn1.getresponse()
    data = res.read()
    msg=data.decode("utf-8")
    msg1=msg.split(',')
    return render_template('upload.html',msge=msg1)

```

```

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

```

```

@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('username', None)
    return render_template('login.html',msg="successfully logged out")

```

```

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

```

## 13.2 GitHub & Project Demo Link

Github link - <https://github.com/IBM-EPBL/IBM-Project-15937-1659606192>

Video demo link -

<https://drive.google.com/file/d/11zVokPd3XuxWlzu1tT8FYb3sJOjKfHWN/view?usp=drivesdk>