NUTRITION ASSISTANT APPLICATION

TEAM ID: PNT2022TMID07438

A PROJECT REPORT

Submitted by

DHILIP KUMAR.V.K

DHIVAGAR.A

ASWATH.M

ARUN KUMAR.S

COMPUTERSCIENCE AND ENGINEERING

P. A. COLLEGE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

Pollachi, Coimbatore Dt. - 642 002

NOVEMBER 2022

P. A. COLLEGE OF ENGINEERING AND TECHNOLOGY
BONAFIDE CERTIFICATE

Certified that this project report **"WEB PHISHING DETECTION USING MACHINE LEARNING"** is the work of **"DHILIP KUMAR.V.K (721719104018), DHIVAGAR.A (721719104020), ASWATH.M(721719104010), ARUN KUMAR.S (721719104007)"** who carried out the project work under our supervision.

SIGNATURE
Dr. D. CHITRA
Professor
HEAD OF THE DEPARTMENT
Computer Science and Engineering
P. A. College of Engineering and Technology

SIGNATURE
FACULTY MENTOR
Dr. M. UMASELVI
Associate Professor
Computer Science and Engineering
P. A. College of Engineering and Technology

SIGNATURE
FACULTY EVALUATOR
Mrs. P. SANGEETHA
Assistant Professor
Computer Science and Engineering
P. A. College of Engineering and Technology

Submitted to the Viva- Voce Examination held on ----------

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INDEX

NUTRINFO - 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 called **NUTRINFO** 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. Thatname 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

(1) Smartphone Applications for Promoting Healthy Diet and Nutrition

A variety of apps relating to diet, nutrition, and weight control are available from major smartphone platforms such as iPhone, Android, Nokia, and BlackBerry. Common techniques include providing feedback, goal-setting for healthy eating, healthy cooking, grocery or restaurant decision making, self-monitoring of energy and nutrient intake, weight tracking, and planning social support and change

Advantages: Smartphone apps are likely to be a useful and low-cost intervention for improving diet and nutrition and addressing obesity in the general population. The accuracy of diet and nutrition measurements obtained using mobile devices has generally been foundto be good.

Disadvantages: In future work, it prospects to improve classification accuracy.

(2) PERSONALIZED DIETARY ASSISTANT

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.

(3) Machine Learning based SVM classifier and LLC

It is a menu-match: restaurant-specific food logging from images. An image recognition framework based on the bag of visual words approach which extracts the base features from the images and then encoded with locality- constrained linear coding (LLC). The extracted features are pooled using max-pooling in a rotationinvariant pooling scheme.A regression based method estimates the calories and along with feature representation mapped the feature space to calories using Support Vector Regression. The approach is limited for discrete serving sizes and custom menu and is also dependent on the GPSS of food consumption. The system lacks user customization and requires cost-sensitive learning to directly minimize calorie estimation errors during the training.

Advantages: Automated computer vision system for logging food and robust calorie estimation.

Disadvantages: It lacks user customization and is GPSS dependent.

(4) Machine Learning based K Means clustering and SVM

It is a method for measuring the calories and nutrition from food images using machine learning techniques. The images got from the mobile device are pre processed followed by the segmentation step to extract the colour and texture features through K Means clustering. The extracted options are used for food classification using Support Vector Machine (SVM). The food portion volume measurement is done by superimposing a grid of squares onto the image segment which matches the irregular shape of the food images easily. The calorie measurement is done based on the food mass and nutritional tables. The system has limited cuisine varieties mixed food images have not been considered.

Advantages:Img2 calories app that determines the calorie intake and estimation.

Disadvantages:It lacks user customization and is GPSS dependent.

2.2 References:

1. Steven S. Coughlin, PhD, Mary Whitehead, MPH, CHES, Joyce Q. Sheats, RN, MPH, Jeff Mastromonico, Dale Hardy, PhD, RD, LD, CDE, CHES, and Selina A. Smith, PhD, MDiv.(2016).Smartphone Applications for Promoting Healthy Diet and Nutrition: A Literature Review.PMC4725321.

2. 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.

2.3 Problem Statement Definition

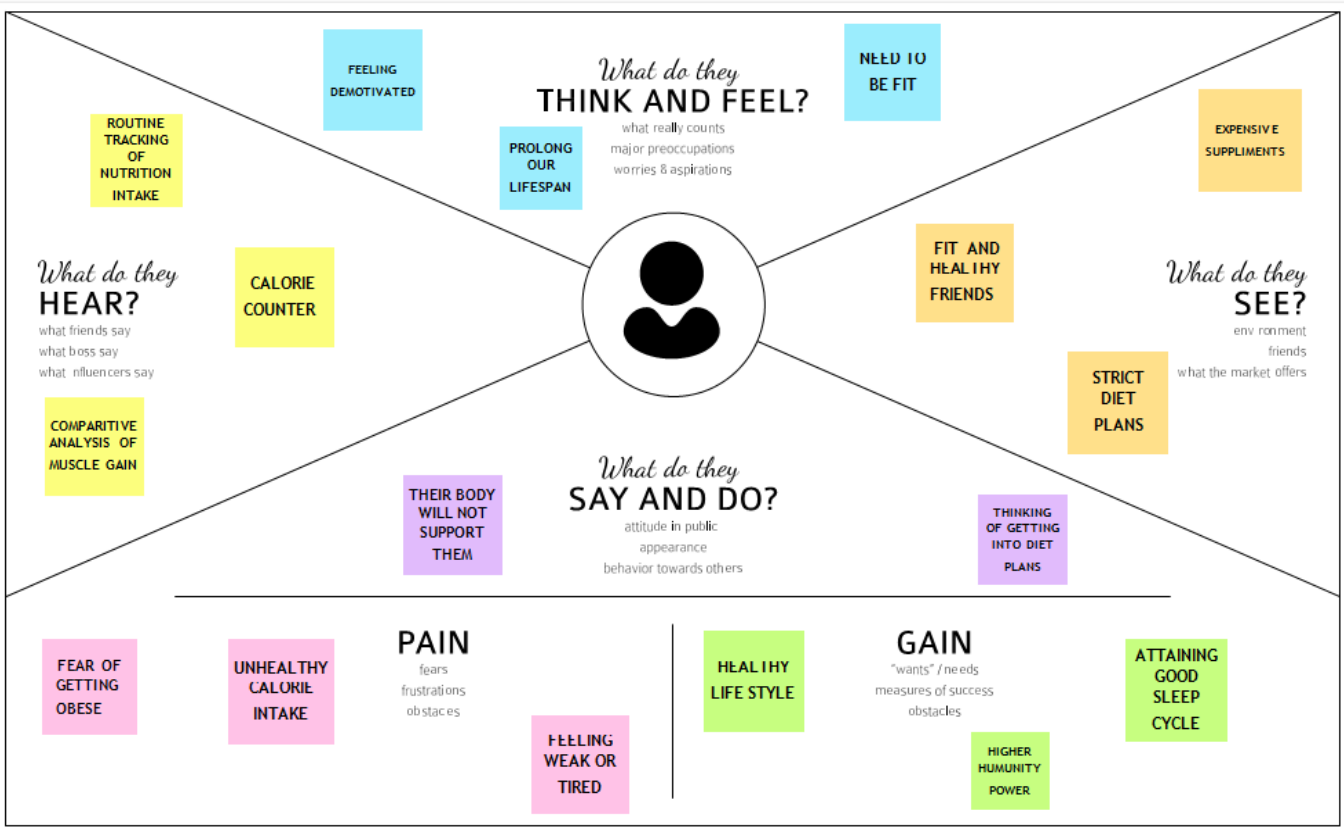
- People need to control their daily calorie intake by eating healthier foods, which is the most basic method to avoid obesity. However, although food packaging comes with nutrition (and calorie) labels, it is still not very convenient for people to refer.

- 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.

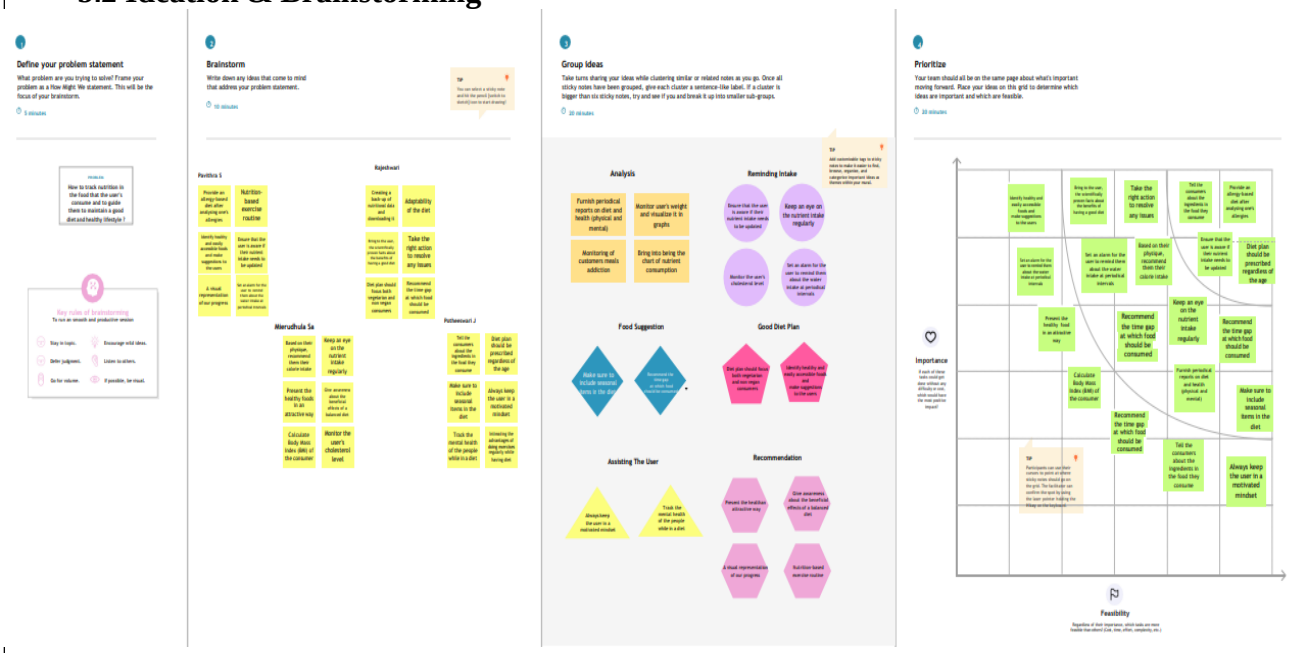
- To develop a web application that automatically estimates food attributes such as ingredients and nutritional value by classifying the input image of food provided.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

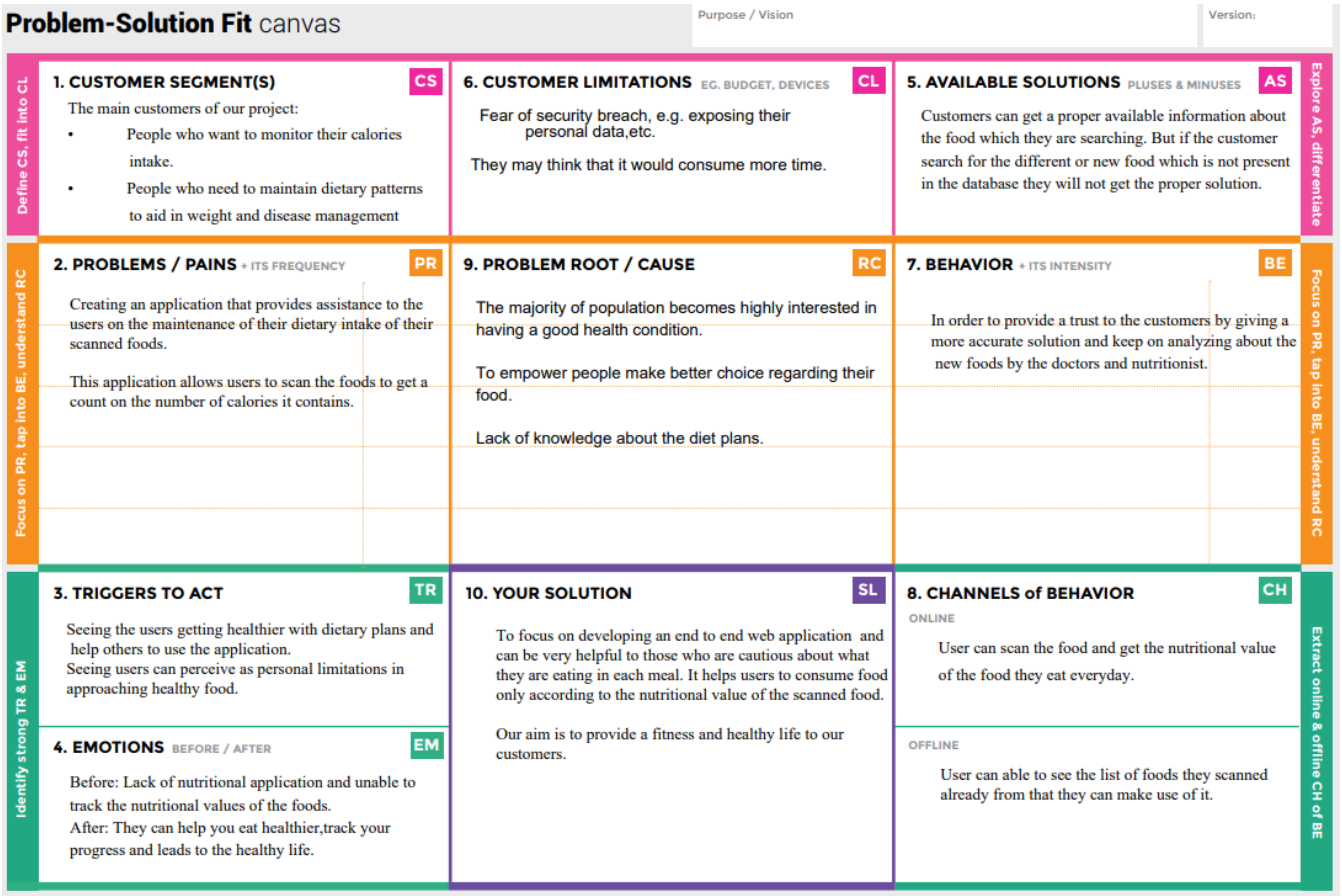


3.3 Proposed Solution

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. • 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
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 and Food API's to give the nutritional value of the identified food
3.	Novelty / Uniqueness	• Like smartwatches counting our steps and physical activities, this app counts the nutrition that one consumes in each meal • This app can be very useful to those who are cautious about what they are eating in each meal
4.	Social Impact / Customer Satisfaction	• The customers using this app can lead a healthier life • Improving the self efficiency of an

		individual • Increasing their desire to set and achieve health diet goals • One to one approach is much more simplified and the client can converse to his/her personal dietician at their own convenient setting
5.	Business Model (Revenue Model)	• Health care policy • Customised nutrition and consumption requirements
6.	Scalability of the Solution	• Easily accessible • Provides good customer relationship

3.4 Problem Solution fit:



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

S.No	IDENTIFIER	REQUIREMENTS
1	Add health information	This application will allow to add health related information of the user.
2	Delete health information	This application will allow to delete the unwanted details about their health.
3	Categories of nutritional food	The categories of food.
4	View of Dashboard	Application will allow user to view the dashboard containing nutrition details.
5	Mail Notification	This application will allow to send mail notification to user when there are any issues regarding their health
6	Tracking System	The health can be tracked with this application.
7	Graph analysis	This application will demonstrate health condition by means of nutritional content
8	Identifying the high calorie food	The high calorie ingredients will be shown via this application.
9	Identifying the low calorie food	The high calorie ingredients will be shown via this application.
10	Passcode	This application has the option to set a passcode to keep their medical reports safe.
11	Add multiple accounts	This application has the option of creating multiple accounts for the users.

12	Selection of health report duration	This application has the ability to select the duration for displaying the health report as weekly or monthly.
13	Update account	This application will allow the user to update their profile.
14	Add account	This application will allow the user to add their profile.
15	Delete account	This application will allow the user to delete their profile.
16	PDF report	This application will generate the pdf report of medical analysis.
17	Pupation of nutritional trends	This application will allow constant review of nutritional trends and pupation.

4.2 Non-Functional requirements

4.2.1 PERFORMANCE REQUIREMENTS

- RESPONSE TIME:
  - ❖ This web application loads within 1-2 seconds, when first loaded and loads within a second when refreshed.
  - ❖ Displaying the nutrient list takes 3-4 seconds when a photo is successfully uploaded as the web app uses two APIs and then sorts the list to display along with a pie chart.
  - ❖ Calculation of BMI value is done within a fraction of seconds.
- OPTIMIZATION :
  - ❖ To decrease the loading time of the web application all the images were compressed.
  - ❖ Optimization of JavaScripts was done very keenly.

4.2.2 SECURITY REQUIREMENTS

- Authentication : The user can only access the features of this web application after

successful authentication through logging in.

● Encryption : The passwords are hashed before being saved into the database with MD5 hashing methods

4.2.3 SOFTWARE QUALITY ATTRIBUTES

● EASE OF USE :

- ❖ Users only have to upload a photo to get all the nutritional content about the food item instantly.
- ❖ A step by step process of doing that is also shown for users ease.

● FLEXIBILITY :

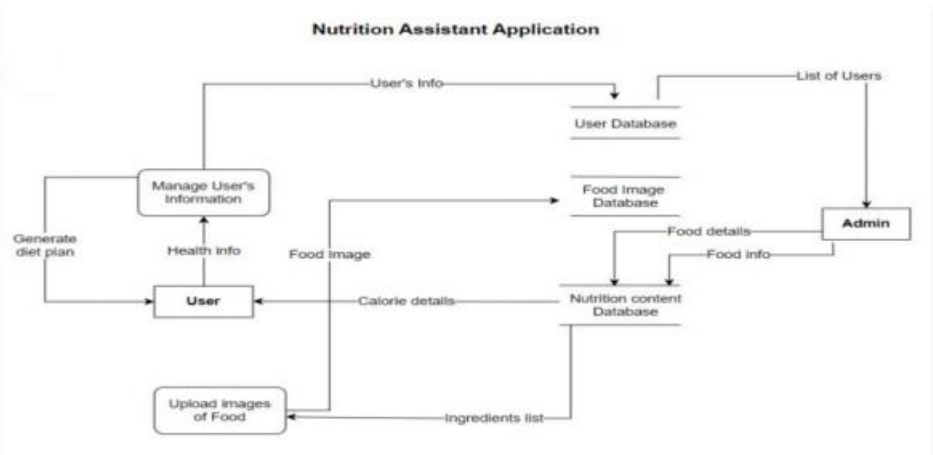
Containerizing a web application makes it very flexible as one has to no longer worry about the prerequisites to run this application on their machine. All that is required is having Docker installed to run the containerized applications.

● CORRECTNESS :

- ❖ The nutrient list and their values are being fetched from a very official site U.S Department of Agriculture, the amount present in the food item specified by their api are totally correct and accurate.
- ❖ While calculating the BMI value of users , measures have been taken to provide accurate results.

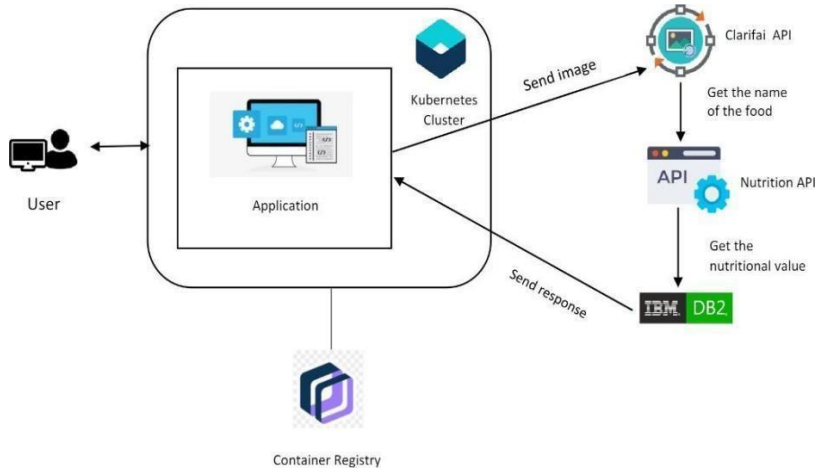
5. PROJECT DESIGN

5.1 Data Flow Diagrams:

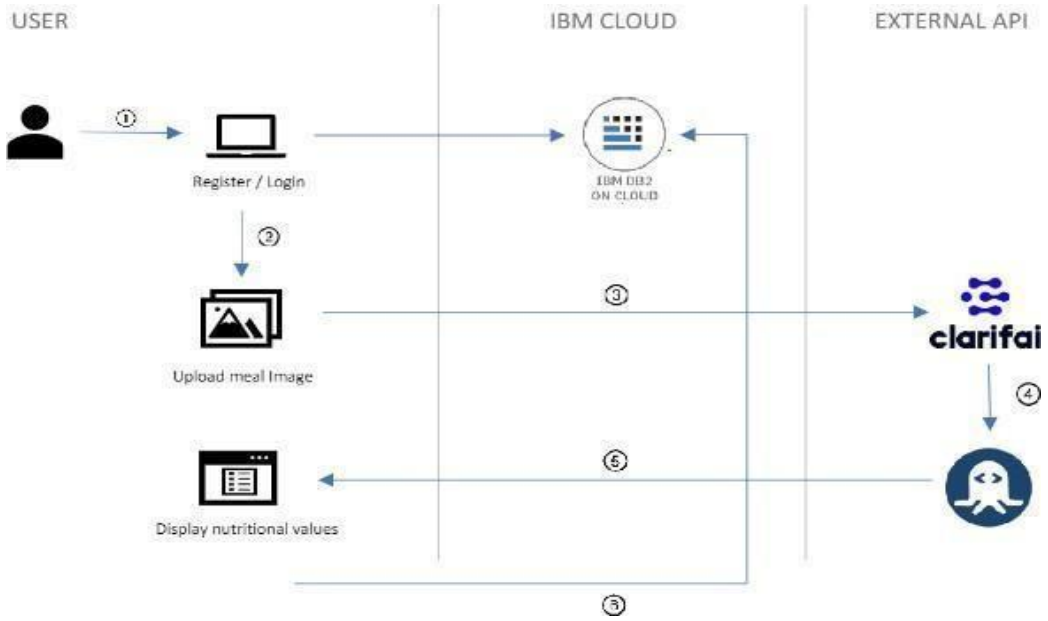


5.2 Solution & Technical Architecture

5.2.1 Solution Architecture



5.2.2 Technical Architecture



5.2 User Stories

SPRINT

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web 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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		USN-3	As a user, I can register for the application through social media	I can register & access the dashboard with Social Media Login	Low	Sprint - 2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail	Medium	Sprint - 1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access the application	High	Sprint - 1
	Dashboard	USN-6	As a user, I can navigate the different features of the application		High	Sprint - 1
	Home Page	USN-7		.	High	
	Upload the food image	USN-8	As a user, I can able to upload food images in	Clarity of the picture	High	Sprint - 3
	View the history of food items searched.	USN-10	As a user, I can able toview the search history.	I can view my history of search	Medium	Sprint - 4
Customer Care Executive		USN- 11	As a customer care executive I can able to solve the issues faced bythe customer	I can provide solution/support at any time	Medium	Sprint - 3
Administrat or	Application	USN-12	As a Administrator I canable to update the application whenever inneed	I can fix the bug which is arises by the customer who use the application	Medium	

C

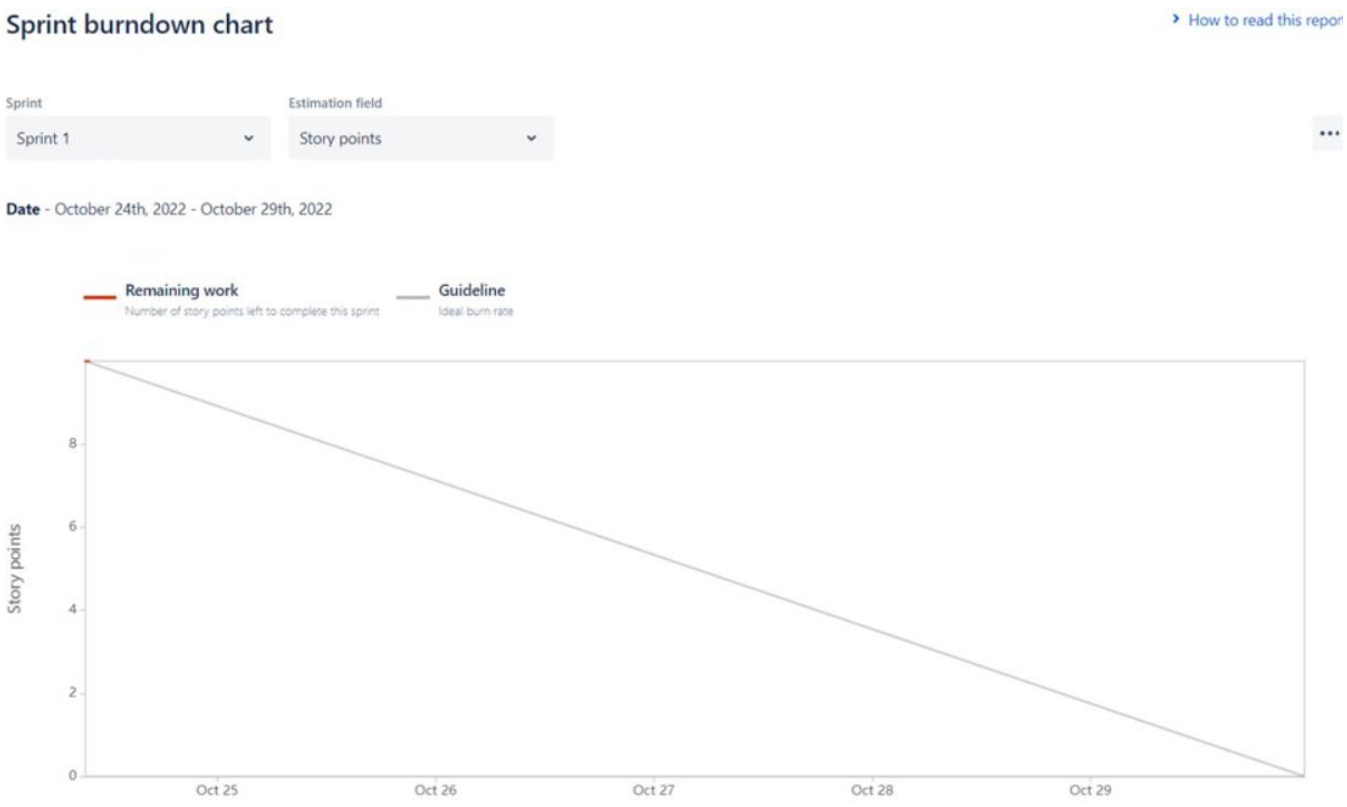
	Add food datato the database	USN-13	As an Administrator I will store the nutrition value and caloriesof the food.	I can store, add and update the food nutritional content and calorie in the database	Medium	Sprint- 3
	Add user datato the database	USN-14	As an Administrator I will storethe user information provided while the user register.	I can store the user data in the database.	Medium	Sprint-3

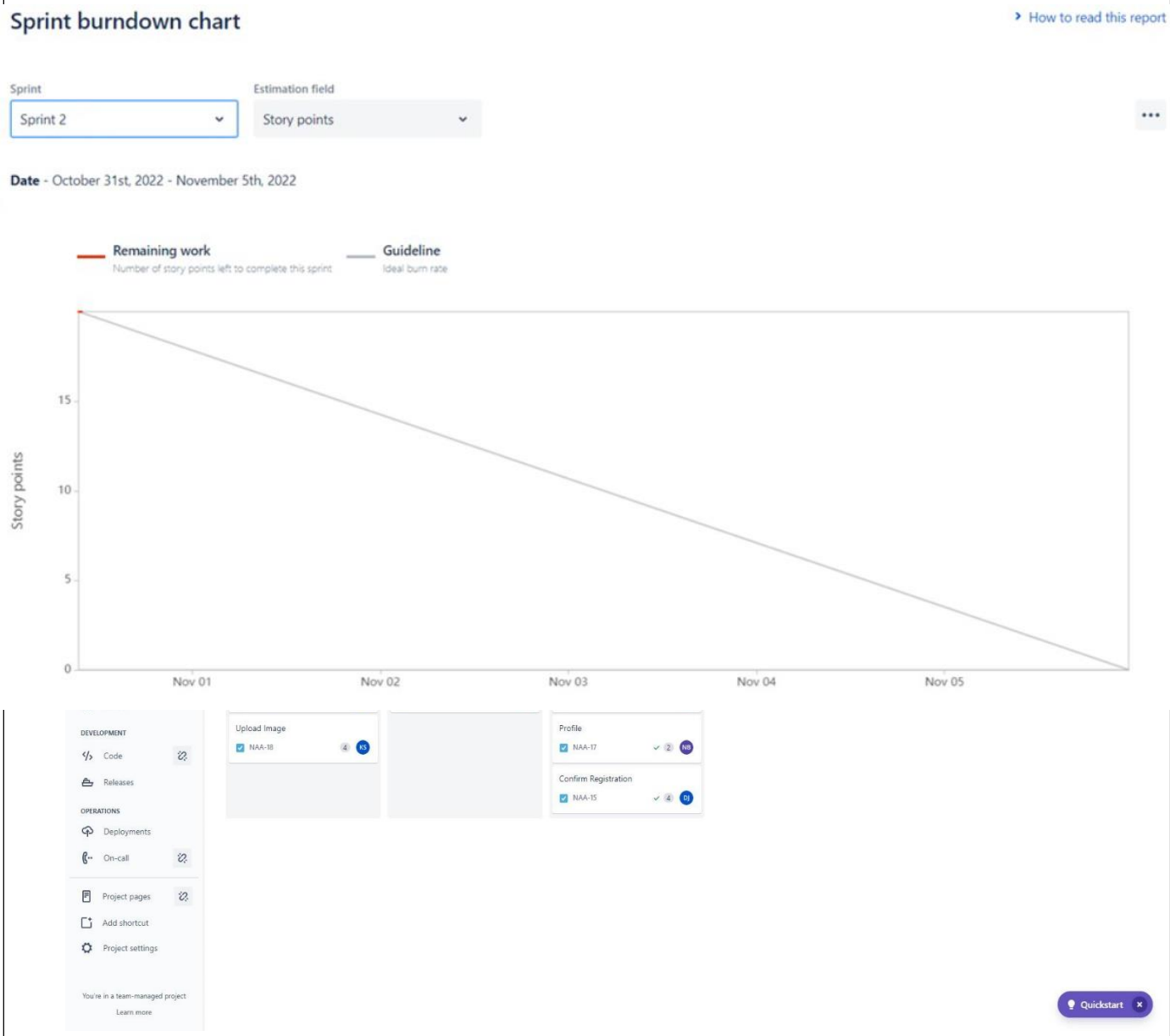
6 PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION:

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

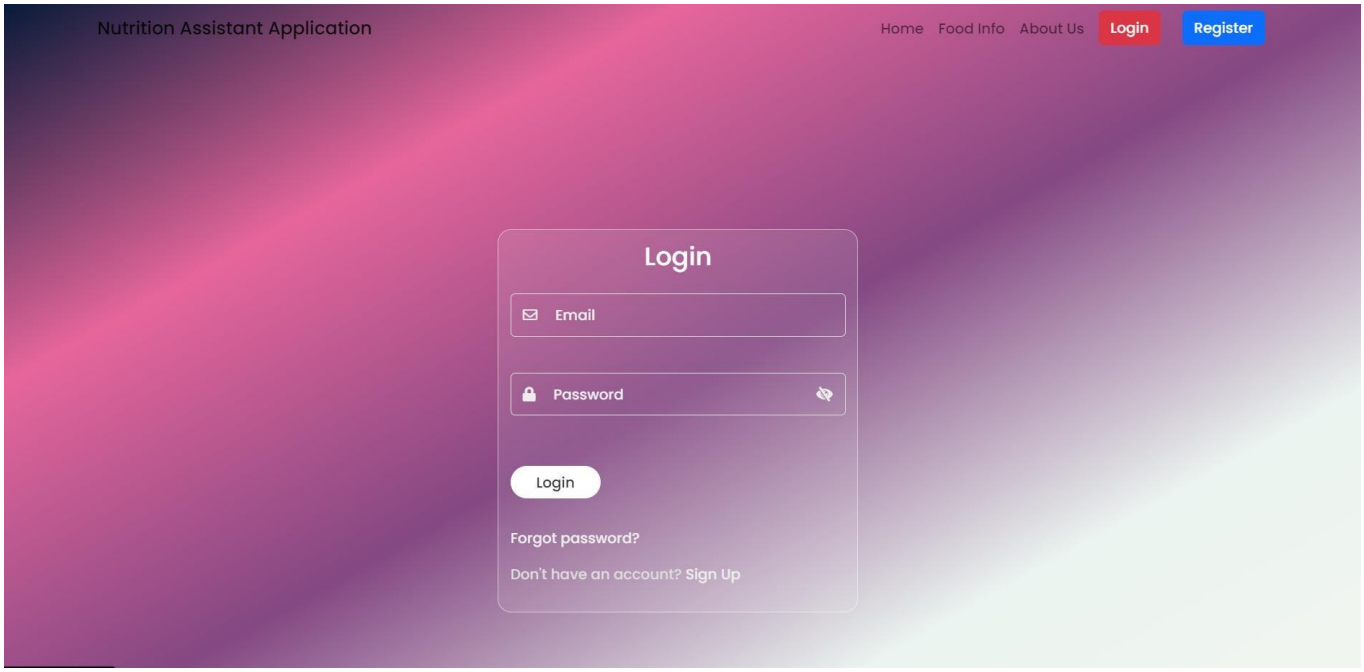
6.1 Reports from JIRA

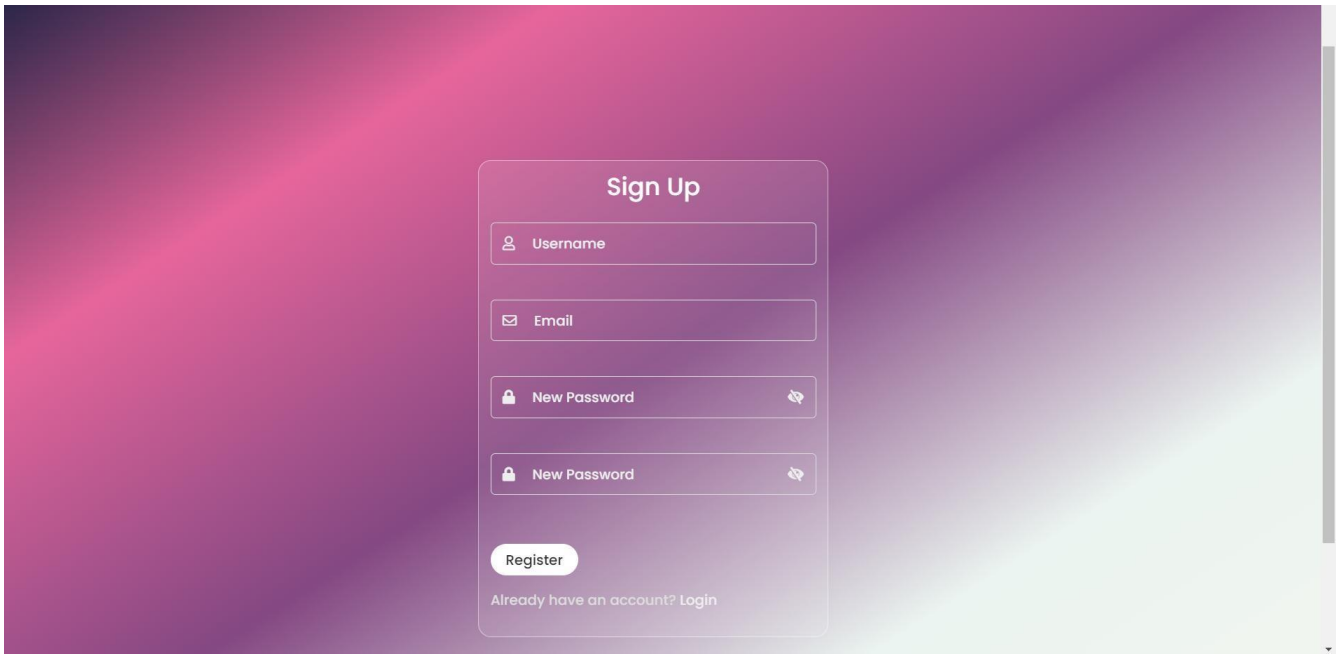


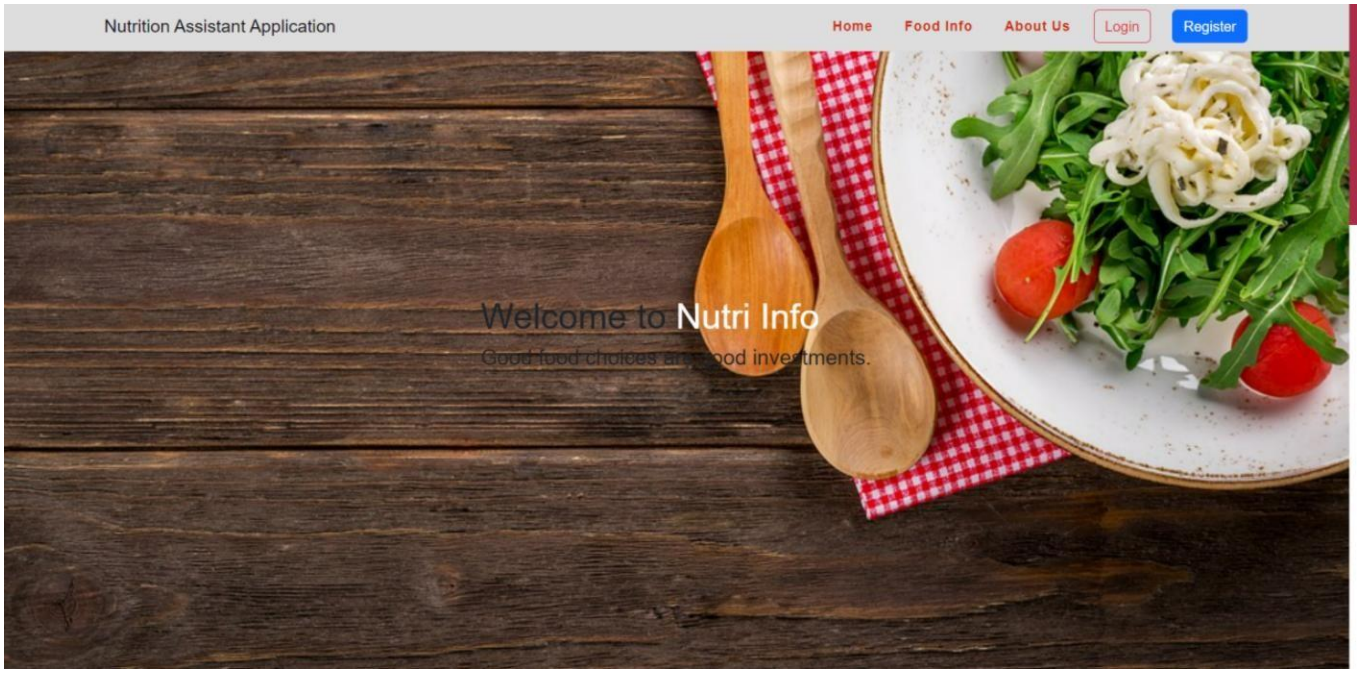

Sprint burndown chart
How to read this report
Sprint
Sprint 2
Estimation field
Story points
Date - October 31st, 2022 - November 5th, 2022
Remaining work
Number of story points left to complete this sprint
Guideline
Ideal burn rate
Story points
15
10
5
0
Nov 01
Nov 02
Nov 03
Nov 04
Nov 05
DEVELOPMENT
Code
Releases
OPERATIONS
Deployments
On-call
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Upload Image
NAA-18
Profile
NAA-17
Confirm Registration
NAA-15
Quickstart

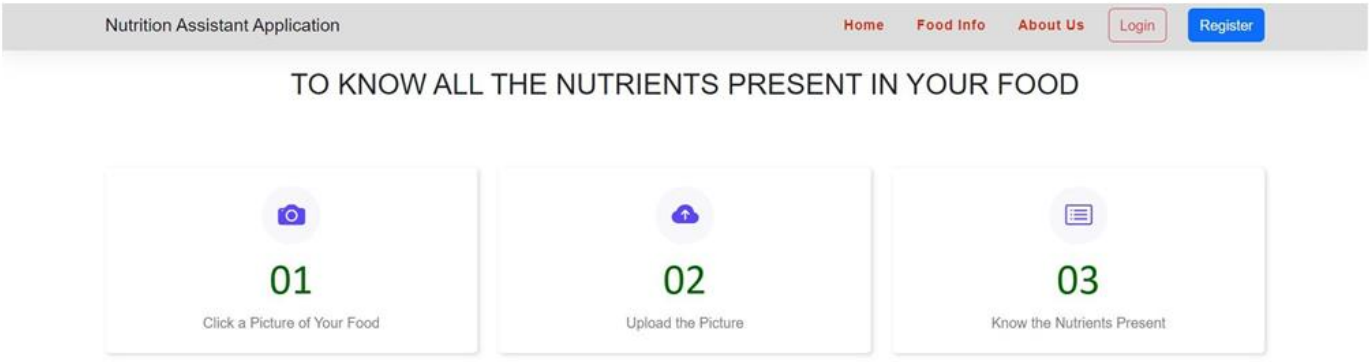
1. CODING & SOLUTIONING

7.1 RESULTS:






Nutrition Assistant Application
Home
Food Info
About Us
Login
Register
Welcome to Nutri Info


Nutrition Assistant Application
Home
Food Info
About Us
Login
Register
TO KNOW ALL THE NUTRIENTS PRESENT IN YOUR FOOD
01
Click a Picture of Your Food
02
Upload the Picture
03
Know the Nutrients Present


Nutri Info

Learn More..

Food item: baked_beans

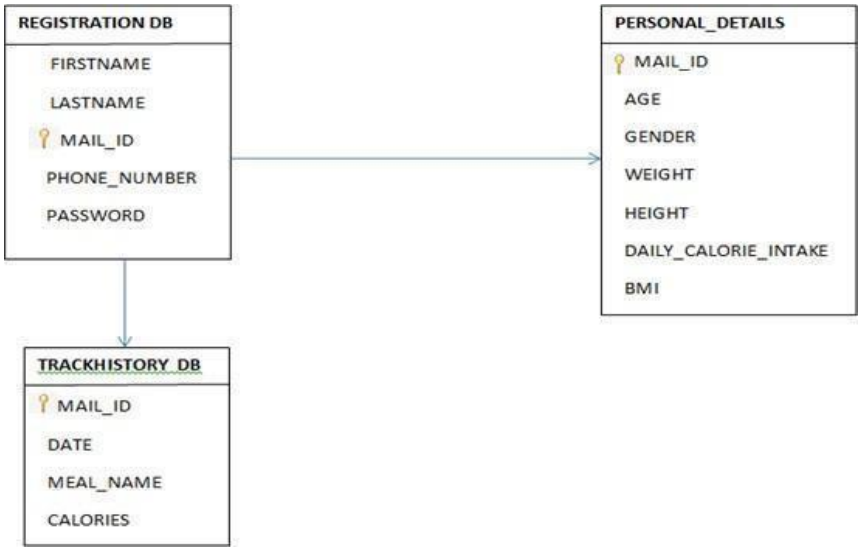
Calories: 349.0 Kcal

Carbs: 51.0 Gm

Fat: 9.0 Gm

Protein: 12.0 Gm

7.2 Database Schema



2. 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_001	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_002	Verify whether upload image button works	Redirected to upload image page	Pass
Dashboard_TC_003	Verify whether track history button works	Redirected to History page	Pass
Uploadimage_TC_001	Check the choose file option available	Able to view the 1.Choose file 2.Submit button 3.Go back button	Pass
Uploadimage_TC_002	Verify whether food image can be uploaded	Preview of the image uploaded will be displayed	Pass
Uploadimage_TC_003	Verify whether it alerts when no image is uploaded	The alert will show "Please upload the file"	Pass
Trackhistory_TC_001	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_002	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	Status
Uploadimage_TC_OO1	Verify whether we get the nutrition values for food image uploaded	The nutrition value of the food image uploaded will be displayed	Pass
Clarify API_TC_OO2	When the user uploads the image the Clarify API will identify the food	The food image should be intendified by calrify api and display	Pass

8.1 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

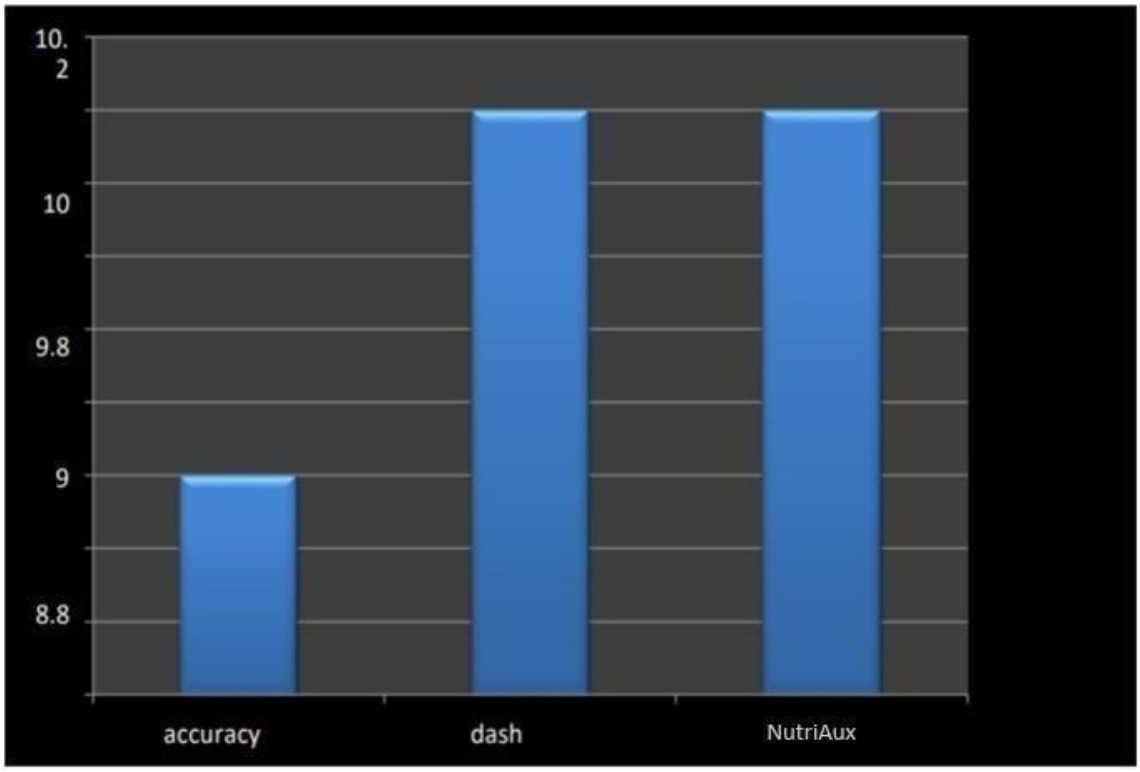
3. Test Case Analysis

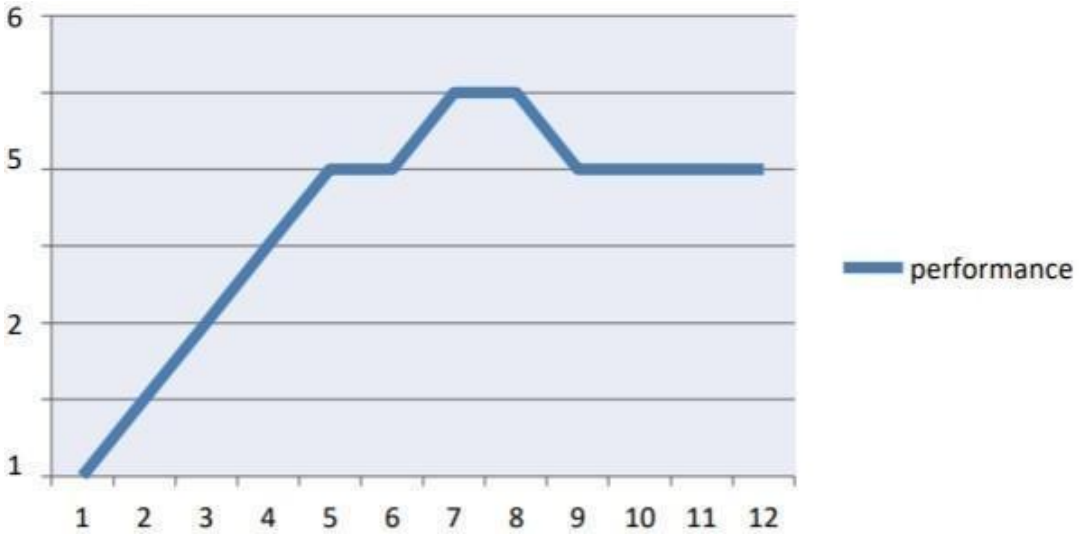
This report shows the number of test cases that have passed, failed, and untested

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

3. RESULTS

9.1 Performance Metrics





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.
- NUTRINFO 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.

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. NUTRINFO 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 thenutritional value. They can also track their daily calorie intake.

FUTURE SCOPE

NUTRINFO 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 the future.

APPENDIX

Source Code

flaskapi.py

home.html

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

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

{% block body %}
<header>
   <nav class="navbar navbar-expand-lg navigation-wrap">
      <div class="container">
         <a class="navbar-brand" href="/">Nutrition Assistant Application</a>
         <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav"
            aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
         </button>
         <div class="collapse navbar-collapse" id="navbarText">

            <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
               <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="/">Home</a>
               </li>
               <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="/foodinfo">Food Info</a>
               </li>
```

```
                {% if session['username'] %}
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="/uploader">Upload
Images</a>
                </li>
                {% endif %}
                <li class="nav-item">
                  <a class="nav-link" aria-current="page" href="/#aboutus">About Us</a>
                </li>
                {% if session['username'] %}
                <li>
                  <a class="btn btn-outline-danger ms-2"
href="{{ url_for('logout') }}">Logout</a>
                </li>
                {% else %}
                  <li>
                    <a class="btn btn-outline-danger ms-2" href="/login"
role="button">Login</a>
                  </li>
                  <li>
                    <a class="btn btn-primary ms-4" href="/signup"
role="button">Register</a>
                  </li>
                {% endif %}
              </ul>
            </div>
          </div>
        </nav>
      </header>
      <section id="home">

        <div class="test">
          <div class="main">

          </div>
          <div class="textbox">
            {% if session['username'] %}
            <h1>Welcome <span style="color:#D12B10;">{{ session['username'] |
title }}</span></h1>
            <!-- <a href="{{ url_for('logout') }}">Logout</a> -->
          {% else %}
            <h1>Welcome to  <span style="color:#D12B10;">Nutrini</span></h1>
          {% endif %}

          <h4>Good food choices are good investments.</h4>
          </div>
        </div>

      </section>
```

```
<section class="feature section-padding" id="feature">
        <div class="container-fluid px-0 top-banner1">
        <div class="container">
                <div class="feature-heading" style="visibility: visible; animation-name: zoom">
                        <h2><span style="color:#D12B10;">Nutrify</span>'s Main Features</h2>
                </div>

                <div class="serv-field row mt-4">
            <div class="col-12 col-md-6 col-lg-4 text-center">
               <div class="serv-box">
                  <div class="icon">
                     <i class="bi bi-stopwatch-fill"></i>
                  </div>
                  <h4>60 seconds</h4>
                  <p>Professional grade weekly meal plan creation in under 60 seconds.</p>
               </div>
            </div>
            <div class="col-12 col-md-6 col-lg-4 text-center">
               <div class="serv-box">
                  <div class="icon">
                     <i class="bi bi-menu-up"></i>
                  </div>
                  <h4>3,000+ variations</h4>
                  <p>Such as smoothies, soups, main dishes and so much more.</p>
               </div>
            </div>
            <div class="col-12 col-md-6 col-lg-4 text-center">
               <div class="serv-box">
                  <div class="icon">
                     <i class="bi bi-cloud-arrow-down-fill"></i>
                  </div>
                  <h4>1-Click Download</h4>
                  <p>Easily access from any device &amp;amp; share meal plans from any device.</p>
               </div>
            </div>
        </div>
        </div>
</div>

</section>
<script type="text/javascript" >
        let nav=document.querySelector(".navigation-wrap");
        window.onscroll = function(){
                if(document.documentElement.scrollTop > 20){
                        nav.classList.add("scroll-on");
```

```
}else{
        nav.classList.remove("scroll-on");
```

```
                }
        }

</script>

<section class="feature section-padding" id="trackfood">
        <div class="container-fluid px-0 top-banner1">
        <div class="container">
                <div class="feature-heading" style="visibility: visible; animation-name: zoom">
                        <h2>TO KNOW ALL THE NUTRIENTS PRESENT IN YOUR FOOD</h2>
                </div>

                <div class="serv-field row mt-4">
            <div class="col-12 col-md-6 col-lg-4 text-center">
               <div class="serv-box">
                  <div class="icon">
                     <i class="bi bi-camera-fill"></i>
                  </div>
                 <h1 style="font-size: xxx-large;
      color: darkgreen;
      font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;">01</h1>
                  <p>Click a Picture of Your Food</p>
               </div>
            </div>
            <div class="col-12 col-md-6 col-lg-4 text-center">
               <div class="serv-box">
                  <div class="icon">
                     <i class="bi bi-cloud-arrow-up-fill"></i>
                  </div>
                 <h1 style="font-size: xxx-large;
      color: darkgreen;
      font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;">02</h1>
                  <p>Upload the Picture</p>
               </div>
            </div>
            <div class="col-12 col-md-6 col-lg-4 text-center">
               <div class="serv-box">
                  <div class="icon">
                     <i class="bi bi-card-list"></i>
                  </div>
                 <h1 style="font-size: xxx-large;
      color: darkgreen;
      font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;">03</h1>
                  <p>Know the Nutrients Present</p>
               </div>
            </div>
         </div>
```



```
            <br>
            <center>
               {% if session['username'] %}
      <a href="/uploader" class="btn btn-outline-primary" role="button" aria-pressed="true">
TRY NOW </a>
      {% endif %}
    </center>
            </div>
</div>

</section>

<footer id="aboutus">
      <div class="container">
         <div class="row pb-4">
            <div class="foot-info col-12 col-md-6 col-lg-9">
               <a class="foot-logo" href="#home">
                  Nutri<span style="color:#D12B10;" >ni</span>
               </a>
               <div class="mail">
                  <i class="fas fa-envelope"></i>
                  <a
href="mailto:dinesh.j.2019.it@rajalakshmi.edu.in">dinesh.j.2019.it@rajalakshmi.edu.in</
a>
               </div>
               <div class="mail">
                 <p>Contact Us: +91 12345 67890</p>
               </div>
            </div>
         </div>
      </footer>
{% endblock %}
```

navbar.css

```
a,
a:hover{
      text-decoration: none;
}

a:hover{
      color: #FF0000;
}

html{
      scroll-behavior: smooth;
}

body{
```

```
        font-family: 'Roboto', sans-serif;
        font-size: 100%;
        font-weight: 400;
    }
```

register.html

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

    {% from "_render_field.html" import render_error_field %}

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

    {% block body %}
      <div class="container h-100">
       <div class="wrapper">
          <div class="card">
             <form class="d-flex flex-column" action="{{ url_for('signup') }}"
    method="POST" novalidate>
                {{ form.csrf_token }}
                <div class="h3 text-center text-white">Sign Up</div>
                <div class="d-flex align-items-center input-field my-3 mb-4">
                   <span class="far fa-user p-2"></span>
                   {{ form.username(class="form-control", placeholder="Username") }}
                </div>
                {{ render_error_field(form.username) }}

                <div class="d-flex align-items-center input-field mb-4">
                   <span class="far fa-envelope p-2"></span>
                   {{ form.email(class="form-control", placeholder="Email") }}
                </div>
                {{ render_error_field(form.email) }}

                <div class="d-flex align-items-center input-field mb-4">
                   <span class="fas fa-lock p-2"></span>
                   {{ form.pass1(class="form-control", placeholder="New Password",
    id='pwd') }}
                   <button type="button" class="btn" onclick="showPassword()">
                      <span class="fas fa-eye-slash"></span>
                   </button>
                </div>
                {{ render_error_field(form.pass1) }}
```

```
                <span class="fas fa-lock p-2"></span>
                {{ form.pass2(class="form-control", placeholder="New Password",
id='pwd1') }}
                <button type="button" class="btn" onclick="Password()">
                  <span class="fas fa-eye-slash"></span>
                </button>
              </div>
              {{ render_error_field(form.pass2) }}

              <div class="my-3">
                <input type="submit" value="Register" class="btn btn-primary">
              </div>

              <div class="mb-3">
                <span class="text-light-white">Already have an account?</span>
                <a href="/login">Login</a>
              </div>

            </form>

          </div>
        </div>
      </div>
      {% endblock %}
```

login.html

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

      {% from "_render_field.html" import render_error_field %}

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


      {% block body %}
      <div class="container h-100">
        <div class="wrapper">
          <div class="card">
            <form class="d-flex flex-column" method="POST" action="{{ url_for('login') }}"
      novalidate >
              {{ form.csrf_token }}
              <div class="h3 text-center text-white">Login</div>
              <div class="d-flex align-items-center input-field my-3 mb-4">
                <span class="far fa-envelope p-2"></span>
                {{ form.email(class="form-control", placeholder="Email") }}
                <!-- <input type="text" placeholder="Email" required class="form-control"> -
```

```
->
                </div>
                {{ render_error_field(form.email) }}
                <div class="d-flex align-items-center input-field mb-4">
                  <span class="fas fa-lock p-2"></span>
                  {{ form.password(class="form-control", id="pwd", placeholder="Password") }}
                  <!-- <input type="password" placeholder="Password" required class="form-control" id="pwd"> -->
                  <button type="button" class="btn" onclick="showPassword()">
                    <span class="fas fa-eye-slash"></span>
                  </button>
                </div>
                {{ render_error_field(form.password) }}
                <div class="my-3">
                  <input type="submit" value="Login" class="btn btn-primary">
                </div>
                <div class="d-sm-flex  align-items-sm-center  justify-content-sm-between  my-3">
                  <div class="mt-sm-0 mt-3"><a href="/forgot">Forgot password?</a></div>
                </div>

                <div class="mb-3">
                  <span class="text-light-white">Don't have an account?</span>
                  <a href="/signup">Sign Up</a>
                </div>
              </form>

            </div>
          </div>
        </div>
        {% endblock %}
```

login_style.css

```
        * {
          margin: 0;
          padding: 0;
          box-sizing: border-box;
          font-family: 'Poppins', sans-serif;
        }

        body {
          background-image: linear-gradient(to right bottom, #051937, #004d7a, #008793, #00bf72, #a8eb12);
          background-repeat: no-repeat;
          height: auto;
        }
```

```
.wrapper {
    max-width: 500px;
    margin-top: 15%;
    margin-left: 32%;
    margin-bottom: 13.5%;
}

.wrapper .card {
    max-width: 400px;
    min-height: 380px;
    margin: 30px;
    background: rgba(255, 255, 255, 0.1);
    overflow: hidden;
    backdrop-filter: blur(10px);
    border: 1px solid rgba(255, 255, 255, 0.5);
    border-radius: 15px;
    cursor: pointer;
    padding: 0.8rem;
}

.wrapper .card a {
    text-decoration: none;
    color: #eee;
}

.wrapper .card a:hover {
    color: #fff;
}

.wrapper .card .input-field {
    border: 1px solid #ddd;
    border-radius:  5px;
    color: #eee;
    padding: 0.3rem;
}

.wrapper .card .input-field input {
    background-color: inherit;
}

.wrapper .card .input-field input.form-control,
.wrapper .card .input-field input.form-control:focus {
    border: none;
    outline: none;
    box-shadow: none;
    color: #eee;
}

.wrapper .card .input-field button.btn {
```

```
    color: #eee;
    padding: 0rem;
    padding-right: 0.5rem;
}

.wrapper .card .input-field button.btn:hover {
    color: #fff;
}

.wrapper .card .input-field button.btn:focus {
    border: none;
    outline: none;
    box-shadow: none;
}

.wrapper .card .input-field input::placeholder {
    color: #eee;
}

.wrapper .card .option {
    display: block;
    position: relative;
    padding-left: 25px;
    cursor: pointer;
    user-select: none
}

.wrapper .card .option span.text-light-white:hover {
    color: #fff;

}

.wrapper .card .option input {
    position: absolute;
    opacity: 0;
    cursor: pointer;
    height: 0;
    width: 0
}

.checkmark {
    position: absolute;
    top: 3px;
    left: 0;
    height: 18px;
    width: 18px;
    background-color: #fff;
    border-radius: 2px
}
```

```
.wrapper .card .btn.btn-primary {
    border-radius: 20px;
    width: 100px;
    background-color: #fff;
    color: #333;
    border: none;
}

.wrapper .card .btn.btn-primary:hover {
    color: #fff;
    background: #333;
}

.wrapper .card .btn.btn-primary:focus {
    border: none;
    box-shadow: none;
}

.wrapper .card .text-light-white {
    color: #ddd;
}

.wrapper .card .line span.connect {
    position: absolute;
    top: -12px;
    left: 33%;
    color: #000;
    padding: 0 0.3rem;
    z-index: 100;
    border-radius: 2px;
    background-color: #fff;
}

.wrapper .card .connections a img {
    width: 40px;
    height: 40px;
    border-radius: 50%;
    object-fit: cover;

}
```

forget_password.html

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

{% from "_render_field.html" import render_error_field %}

{% block head %}
```

```
<title>Reset Password</title>
<link rel="stylesheet" href="{{url_for('static', filename='css/reset_style.css')}}">
{% endblock %}

{% block body %}
<div class="container h-100">
  <div class="wrapper">
    <div class="card">
      <form action="{{ url_for('forgot_password') }}" method="POST" class="d-flex
flex-column" novalidate>
        {{ form.csrf_token }}
        <div class="h3 text-center text-white">Reset Password</div>
        <div class="d-flex align-items-center input-field my-4 mb-4">
          <span class="far fa-envelope p-2"></span>
          {{ form.email(class="form-control", placeholder="Registered Email") }}
        </div>
        {{ render_error_field(form.email) }}
        <div class="my-3">
          <input type="submit" value="Reset" class="btn btn-primary">
        </div>

      </form>

    </div>
  </div>
</div>
{% endblock %}
```

index.html

```
{% extends "base.html" %}
{% block head %}
      <title>Nutrify</title>
  <link rel="stylesheet" type="text/css" href="{{url_for('static',
filename='css/indexstyle.css')}}">
      <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
  <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>
{% endblock %}
{% block body %}
```

```
        {% for row in files %}
          <div class="piccard card">
            <img class="pic" src="{{row[0]}}">
            <a href="/nutritioninfo/{{row[1]}}"><button style="width: 30%;" type="submit"
class="btn btn-outline-primary">Learn More..</button>
            </a>
          </div>

        {% endfor %}
        {%  endblock %}
```

indexstyle.css

```
body, html {
   height: 100%;
  }
a,
a:hover{
        text-decoration: none;
}

a:hover{
        color: #FF0000;
}

html{
        scroll-behavior: smooth;
}

body{
        font-family: 'Roboto', sans-serif;
        font-size: 100%;
        font-weight: 400;
}

::-webkit-scrollbar{
        width: 0.625rem;
}

::-webkit-scrollbar-track{
        background: white;
}

::-webkit-scrollbar-thumb{
        background: #AB2C48;
}

.navigation-wrap{
```

```
background-color:#ddd;
```

```
        position: fixed;
        width: 100%;
        left:0;
        z-index: 1000;
        -webkit-transition:all 0.3s ease-out;
        transition: all 0.3s ease-out;
}

.navigation-wrap .nav-item{
        padding: 0 0.625rem;
        transition: all 200ms linear;
}

.navbar-toggler:focus{
        outline: unset;
        border: unset;
        box-shadow: none;
}

.nav-item .nav-link{
        font-size: 0.9375rem;
        font-weight: 600;
        text-transform: capitalize;
        color: #D12B10;
        letter-spacing: 1px;
}
.nav-item .nav-link a:hover{
        color: #FF0000;
}

.navigation-wrap.scroll-on{
        position: fixed;
        top:0;
        left: 0;
        width: 100%;
        background: white;
        box-shadow: 0 0.125rem 1.75rem 0 rgb(0,0,0,0.09);
        transition: all .15s ease-in-0ut 0s;
}

#home{
   height: 100%;
}

.test{
   height: 100%;
   position: relative;
}
```

```
.top-banner{
        width: 80%;

        padding: 10rem 0 7rem;

}

.main{
  background-image: url('/static/images/testy.jpg');
  height: 100%;
  width: 100%;
  font-size: 70px;
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}

.textbox{
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);

}

.upload{
    padding: 4%;
    margin:7%;

    box-shadow: 10px 10px 10px #ddd;
}

.piccard{
width: 30%;
padding: 2%;
border-radius: 7rem;
}

.pic{
    margin-bottom: 7%;
}

.foodinfomain{

padding: 3%;
}


.foodinfo{
```

```
    text-align: right;
  position: absolute;
  top: 20%;
  left: 25%;
    padding: 2%;
    width:50%;
   border-radius: 2rem;
   box-shadow: 10px 10px 10px #ddd;
   font-size: 20px;


}
.fo{
    width: 130%;
}

.top-banner h1{
        font-size: 48px;
}

.top-banner1{
        width: 100%;
        padding: 9.875rem 0 7.375rem;
}

.feature-heading{
        text-align: center;
}

.serv-field {
    padding-top: 50px;
}
.serv-field .serv-box {
    padding: 20px;
    margin-bottom: 20px;
    border-radius: 5px;
    border: 1px solid transparent;
    box-shadow: 2px 2px 5px rgba(136, 136, 136, 0.3);
    transition:  0.4s ease
}
.serv-field .serv-box:hover {
    box-shadow: rgba(0, 0, 0, 0.4) 0px 30px 90px;
    border-color:#5C46F9;
}
.serv-field .serv-box .icon {
    position: relative;
    width:  65px;
    height: 65px;
    display: flex;
```

```
    align-items: center;
    justify-content: center;
    margin: 0 auto 10px;
    color:#5C46F9;
    border-radius: 50%;
}
.serv-field .serv-box .icon:before {
    content: '';
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%,-50%) scale(1);
    width: 100%;
    height: 100%;
    border-radius: 50%;
    background-color: #F7F7FC;
    z-index: -1;
}
.serv-field .serv-box .icon i {
    font-size: 30px;
}
.serv-field .serv-box h4 {
    font-size: 25px;
    color:  #090719;
}
.serv-field .serv-box p {
    color: #787878;
    margin-bottom: 0px;
}

footer {
    background-color: #222;
    color: #fff;
    padding-top: 50px;
}
footer a {
   color: #fff;
    transition: color 0.4s ease;
    text-decoration: none;
}
footer a:hover{
 color: #D12B10;
}
footer ul {
    padding-left: 0;
}
footer .foot-logo {
    display: inline-block;
    font-size: 35px;
```

```
    font-weight: 700;

    margin-bottom: 15px;
}
footer .foot-info > div {
    display: flex;
    margin-bottom: 15px;
}
footer .foot-info > div i {
    width: 35px;
    font-size: 18px;
}
```

foodinfo.html

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

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

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

        </div>
<div class="col-lg-6">
    <p>Food item: {{files[1]}}</p>
    <hr>
    <p>Calories: {{files[2]}} Kcal</p>
    <hr>
    <p>Carbs: {{files[3]}} Gm</p>
    <hr>
    <p>Fat: {{files[4]}} Gm</p>
    <hr>
    <p>Protein: {{files[5]}} Gm</p>

</div>
    </div>
</div>
```

```
</div>
{% endblock %}
```

upload.html

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

{% block head %}
        <title>Nutrify</title>
   <link rel="stylesheet" type="text/css" href="{{url_for('static',
filename='css/indexstyle.css')}}">
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
   <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>
{% endblock %}


{% block body %}
<div class="row">

        <div class="col-lg-6">
        <div class="upload card">
                <h1>Upload Food Image</h1>
        <form class="uploaderform" action = "/uploader" method = "POST"
        enctype = "multipart/form-data">


          <div class="mb-3">
                  <input  class="form-control" type = "file" name = "file" />

          </div>
          <button type="submit" class="btn btn-primary">Submit</button>
        </form>
</div>
</div>

<div class="col-lg-6">
<img src="/static/images/mainimage.jpg" alt="">
</div>

</div>
```

```
{%  endblock %}
```

base.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">

   <!-- Bootstrap -->
   <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@5.0.2/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM
"
      crossorigin="anonymous"></script>
   <link       href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet">
   <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"></scrip
t>

   <link              rel="stylesheet"              href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.5.0/font/bootstrap-icons.css" />
   <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.3.1/css/all.css"
      integrity="sha384-
mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU"
crossorigin="anonymous">

   <!-- Fonts -->
   <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300&display=swap"
rel="stylesheet">
   <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@500&display=swap"
rel="stylesheet">
   <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.3.1/css/all.css"
      integrity="sha384-
mzrmE5qonljUremFsqc01SB46JvROS7bZs3IO2EmfFsd15uHvIt+Y8vEf7N7fWAU"
crossorigin="anonymous">
   <!-- CSS -->
```

```
    <link rel="stylesheet" href="{{url_for('static', filename='css/navbar.css')}}">
    <!-- <link rel="stylesheet" href="{{url_for('static', filename='css/style.css')}}"> -->

    <!-- JS Script for-->
    <script type="text/javascript">

        function showPassword() {
            var password = document.getElementById('pwd');
            if (password.type === 'password') {
                password.type = "text";
            }
            else {
                password.type = "password";
            }
        }

        function Password() {
            var password = document.getElementById('pwd1');
            if (password.type === 'password') {
                password.type = "text";
            }
            else {
                password.type = "password";
            }
        }

    </script>
        {% block head %} {% endblock %}

</head>


<body>
    {% with messages = get_flashed_messages(category_filter=["error"]) %}
    {% if messages %}
    <div class="alert alert-danger" role="alert">
        <ul class=flashes>
            {% for message in messages %}
            <li>{{ message }}</li>
            {% endfor %}
        </ul>
    </div>
    {% endif %}
    {% endwith %}

    {% with messages = get_flashed_messages(category_filter=["success"]) %}
    {% if messages %}
    <div class="alert alert-success" role="alert">
        <ul class=flashes>
```

```
        {% for message in messages %}
        <li>{{ message }}</li>
        {% endfor %}
      </ul>
    </div>
    {% endif %}
    {% endwith %}

<header>
    <nav class="navbar navbar-expand-lg navigation-wrap">
      <div class="container">
        <a class="navbar-brand" href="/">Nutrition Assistant Application</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav"
          aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarText">

          <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
            <li class="nav-item">
              <a class="nav-link" aria-current="page" href="/">Home</a>
            </li>

            <li class="nav-item">
              <a class="nav-link" aria-current="page" href="/foodinfo">Food Info</a>
            </li>
            {% if session['username'] %}
            <li class="nav-item">
              <a class="nav-link" aria-current="page" href="/uploader">Upload
Images</a>
            </li>
            {% endif %}
            <li class="nav-item">
              <a class="nav-link" aria-current="page" href="/#aboutus">About Us</a>
            </li>
            {% if session['username'] %}
            <li>

              <a class="btn btn-outline-danger ms-2"
href="{{ url_for('logout') }}">Logout</a>
            </li>
            {% else %}
              <li>
                <a class="btn btn-outline-danger ms-2" href="/login"
role="button">Login</a>
              </li>
              <li>
```

```
                        <a class="btn btn-primary ms-4" href="/signup"
role="button">Register</a>
                    </li>
                {% endif %}
            </ul>
        </div>
    </div>
  </nav>
</header>


  {% block body %} {% endblock %}

</body>

</html>
```

_render_field.html

```
{% macro render_error_field(field) %}
<ul>
  {% for error in field.errors %}
    <li class="field_error">{{ error }}</li>
  {% endfor %}
</ul>
{% endmacro %}
```

reset_style.css

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: 'Poppins', sans-serif;
}

body {
  background-image: linear-gradient(to right bottom, #051937, #004d7a, #008793,
#00bf72, #a8eb12);
  background-repeat: no-repeat;
  height: auto;
}

.wrapper {
  max-width: 500px;
  margin-top: 17%;
  margin-left: 32%;
  margin-bottom: 24%;
}
```

```
.wrapper .card {
    max-width: 400px;
    min-height: 200px;
    margin: 30px;
    background: rgba(255, 255, 255, 0.1);
    overflow: hidden;
    backdrop-filter: blur(10px);
    border: 1px solid rgba(255, 255, 255, 0.5);
    border-radius: 15px;
    cursor: pointer;
    padding: 0.8rem;
}

.wrapper .card a {
    text-decoration: none;
    color: #eee;
}

.wrapper .card a:hover {
    color: #fff;
}

.wrapper .card .input-field {
    border: 1px solid #ddd;
    border-radius:  5px;
    color: #eee;
    padding: 0.3rem;
}

.wrapper .card .input-field input {
    background-color: inherit;
}

.wrapper .card .input-field input.form-control,
.wrapper .card .input-field input.form-control:focus {
    border: none;
    outline: none;
    box-shadow: none;
    color: #eee;
}

.wrapper .card .input-field button.btn {
    color: #eee;
    padding: 0rem;
    padding-right: 0.5rem;
}

.wrapper .card .input-field button.btn:hover {
```

```
    color: #fff;
}

.wrapper .card .input-field button.btn:focus {
    border: none;
    outline: none;
    box-shadow: none;
}

.wrapper .card .input-field input::placeholder {
    color: #eee;
}

.wrapper .card .option {
    display: block;
    position: relative;
    padding-left: 25px;
    cursor: pointer;
    user-select: none
}

.wrapper .card .option span.text-light-white:hover {
    color: #fff;

}

.wrapper .card .option input {
    position: absolute;
    opacity: 0;
    cursor: pointer;
    height: 0;
    width: 0
}

.checkmark {
    position: absolute;
    top: 3px;
    left: 0;
    height: 18px;
    width: 18px;
    background-color: #fff;
    border-radius: 2px
}

.wrapper .card .btn.btn-primary {
    border-radius: 20px;
    width: 100px;
    background-color: #fff;
    color: #333;
```

```
    border: none;
}

.wrapper .card .btn.btn-primary:hover {
    color: #fff;
    background: #333;
}

.wrapper .card .btn.btn-primary:focus {
    border: none;
    box-shadow: none;
}

.wrapper .card .text-light-white {
    color: #ddd;
}

.wrapper .card .line span.connect {
    position: absolute;
    top: -12px;
    left: 33%;
    color: #000;
    padding: 0 0.3rem;
    z-index: 100;
    border-radius: 2px;
    background-color: #fff;
}

.wrapper .card .connections a img {
    width: 40px;
    height: 40px;
    border-radius: 50%;
    object-fit: cover;

}

.nav-link{
     font-family: 'Bungee Spice', Georgia;
    font-size: 20px;
    padding-top: 29px;
    color: rgb(127,255,0);
}

.nav-link:hover{
    color: #00FF00!important;
}
```

GitHub Link

GitHub: https://github.com/IBM-EPBL/IBM-Project-53237-1661321923