NUTRITION ASSITANT APPLICATION

1.INTRODUCTION

1.1 PROJECT OVERVIEW

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

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

1.2 PURPOSE

Who does the problem affect?	People want to maintain health and body fit but doesn't have anyone to guide for regular diet
What is the issue?	People who intake poor nutrition can lead to many illnesses and they don't know about the ingredients used in the dish and the calories present in them
When does the issue occur?	The issue occurs when the people does not eat the right amount of nutrition dishes
Where is the issue occurring?	The issue occurs in person's health due to lack of nutrition
Why is it important that we fix the problem?	It is important to fix the problem because it helps users to improve their health and switch to a healthy lifestyle.By solving this issue,people can track their health.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

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.

2.2 REFERENCES

1."**APPLICATION OF ARTIFICIAL INTELLIGENCE ON NUTRITION ASSESSMENT AND MANAGEMENT" -** The application of AI for the provision of food services to hospitalized patients is of immense scope. This review details the various ways through which AI can be applied for nutrition assessment. Even though commercial AI-based nutritional assessment systems are available, many do not evaluate the nutrient intake, and the data available through them were not validated. FatSecret is a available AI-based food and nutrient assessment system that can evaluate the food's calorie content. Also, the major challenge posed by such systems is the availability of locally appropriate data sets. Hence further research and validation are essential in this field.

2.**"PERSONAL HEALTH ASSITANT ON ANDRIOD MOBILE DEVICE"-** this work aims to develop an application on mobile devices that is able to record the daily sleeping, exercise and nutrition information,analyze the collected information in order to provide a notification or an alarm, and present the analyzed results in a simple and easy to understand format. The proposed application can collect data from other application and from the users. A set of simple data analysis methods is performed on the collected data in order to provide a personal health advice based on the user pre-defined preferences.

3."**SMARTPHONE APPLICATION FOR PROMOTING HEALTHY DIET AND NUTRITION"** -There is a need for culturally appropriate, tailored health messages to increase knowledge and awareness of health behaviors such as healthy eating. Smartphone apps are likely to be a useful and low-cost intervention for improving diet and nutrition and addressing obesity in the general population. Participants prefer applications that are quick and easy to administer and those that increase awareness of food intake and weight management.

4."**A FOOD RECOMMENDER SYSTEM CONSIDERING NUTRITIONAL INFORMATION AND USER PREFERENCES"** - this paper presents a general framework for daily meal plan recommendations, incorporating as main feature the simultaneous management of nutritional-aware and preference-aware information, in contrast to the previous works which lack this global viewpoint. The proposal incorporates a pre-filtering stage that uses AHPSort as multi-criteria decision analysis tool for filtering out foods which are not appropriate to the current user characteristics. Furthermore, it incorporates an optimization-based stage for generating a daily meal plan whose goal is the recommendation of food highly preferred by the user, not consumed recently, and satisfying his/her daily nutritional requirements. A case study is developed for testing the performance of the recommender system.

5."**DESIGN AND EVALUATION OF MOBILE-BASED NUTRITION EDUCATION APPLICATION "**-According to the nutritionists and infertility specialists, the contents determined for the nutrition educational application were categorized under three general sections of user's demographic data, educational contents, required capabilities. The users' mean score of the application usability was calculated as 7.44 out of 9 indicating a good level of satisfaction. Conclusions Nutrition education of women with infertility problems can play a significant role in improving their awareness and treatment outcomes. Due to the increasing use of smartphones, designing a mobile-based nutrition educational application can be of great benefit for women with infertility according to the cultural conditions and characteristics of each community.

2.3PROBLEM STATEMENT DEFINITION

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.

3. IDEATION & PROPOSED SOLUTION

3.1EMPATHY MAP

- ➤ The **Says** quadrant contains what the user says out loud in an interview or some other usability study. Ideally, it contains verbatim and direct quotes from research.

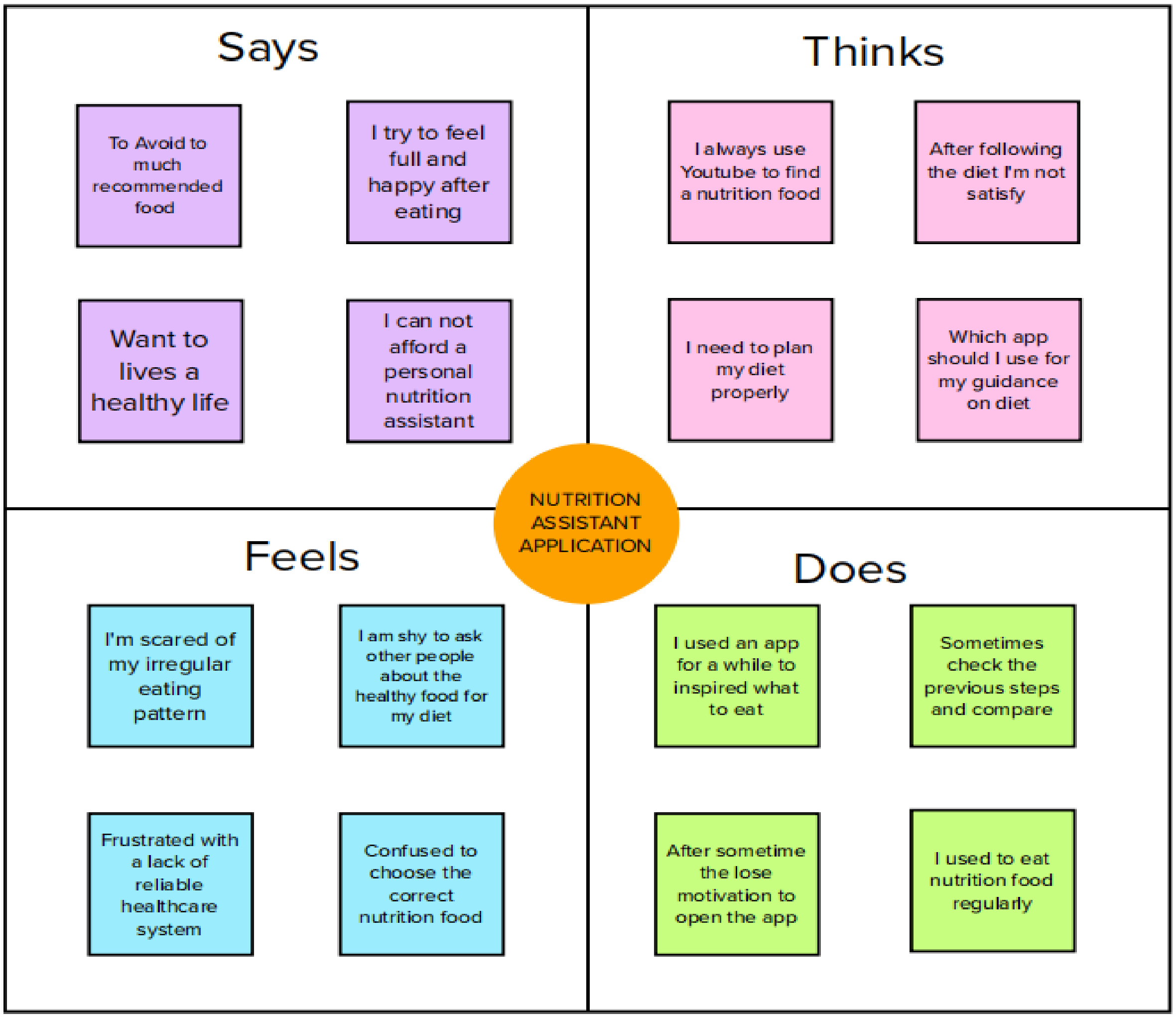
- ➤ The **Thinks** quadrant captures what the user is thinking throughout the experience. Ask yourself (from the qualitative research gathered): what occupies the user's thoughts? What matters to the user? It is possible to have the same content in both Says and Thinks. However, pay special attention to what users think, but may not be willing to vocalize.

- ➤ The **Does** quadrant encloses the actions the user takes. From the research, what does the user physically do? How does the user go about doing it?

- ➤ The **Feels** quadrant is the user's emotional state, often represented as an adjective plus a short sentence for context. Ask yourself: what worries the user? What does the user get excited about? How does the user feel about the experience?

- The empathy map describes what, does, and feels for users of the Waste food management system. The task is for the user as a donor need to register and log in to the page and add details of food to donate and logout.

- The NGO as a volunteer can register and view the donor request list and verify the food and accept or reject the food, If the food is good then accept the request and the volunteer send the pickup time to the donors.



NUTRITION ASSISTANT APPLICATION

Says

To Avoid to much recommended food

I try to feel full and happy after eating

Want to lives a healthy life

I can not afford a personal nutrition assistant

Thinks

I always use Youtube to find a nutrition food

After following the diet I'm not satisfy

I need to plan my diet properly

Which app should I use for my guidance on diet

Feels

I'm scared of my irregular eating pattern

I am shy to ask other people about the healthy food for my diet

Frustrated with a lack of reliable healthcare system

Confused to choose the correct nutrition food

Does

I used an app for a while to inspired what to eat

Sometimes check the previous steps and compare

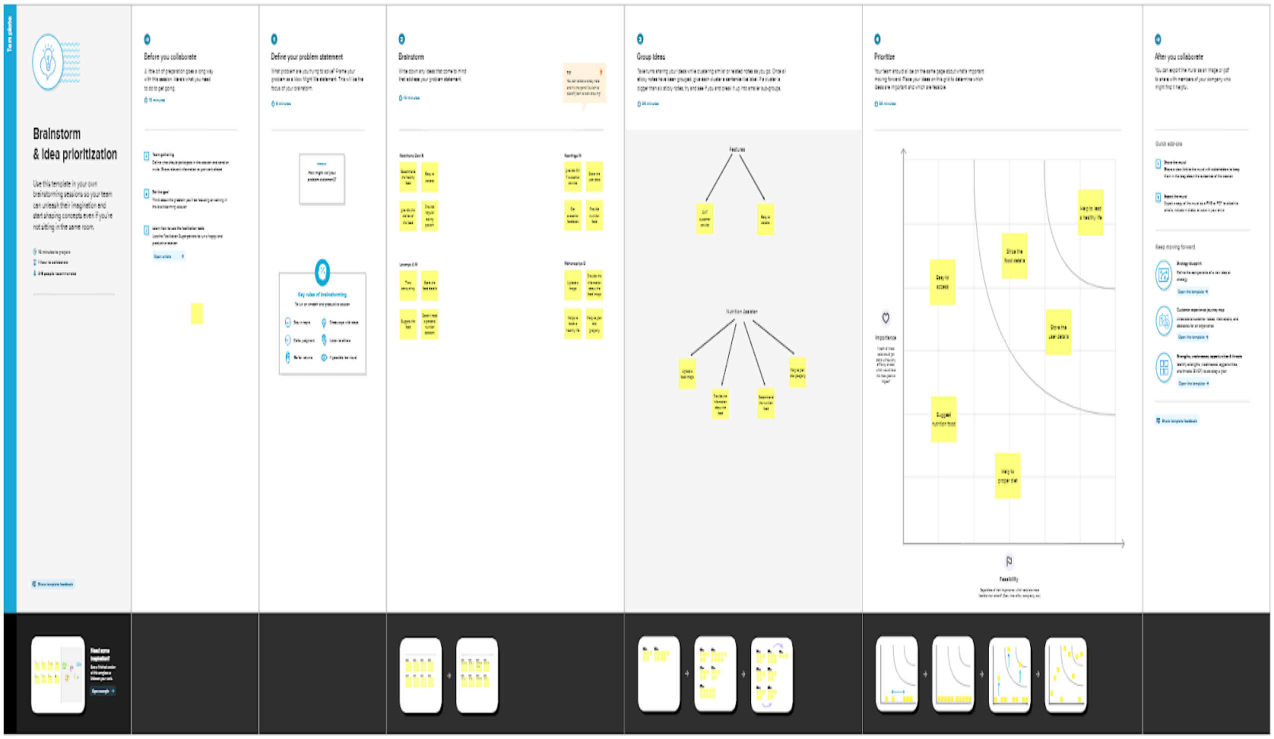
After sometime the lose motivation to open the app

I used to eat nutrition food regularly

3.2 IDEATION & BRAINSTORMING

As this is an open ended problem statement, We can start with initial brainstorming. By this process we'd have an idea of overall app structure & features.

- After gathering all notes and information on wall or soft paper, we would organize a vertical section based on similar functionality and features.
- I've tried to collect all possible information and points related to food processing activities & user's health.



Brainstorm & Idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Before you collaborate

Define your problem statement

How might we [your problem statement]?

Key rules of brainstorming

Brainstorm

Write down any ideas that come to mind that address your problem statement.

Group Ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

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	Clarifai's AI-Driven Food Detection Model is used to get accurate food identification. People can easily track the Nutrition and calories by scanning real-time images of a food and examine its nutritional content which will improve the dietary habits. This app will provide healthy life and also recommended diet plans for all the users.

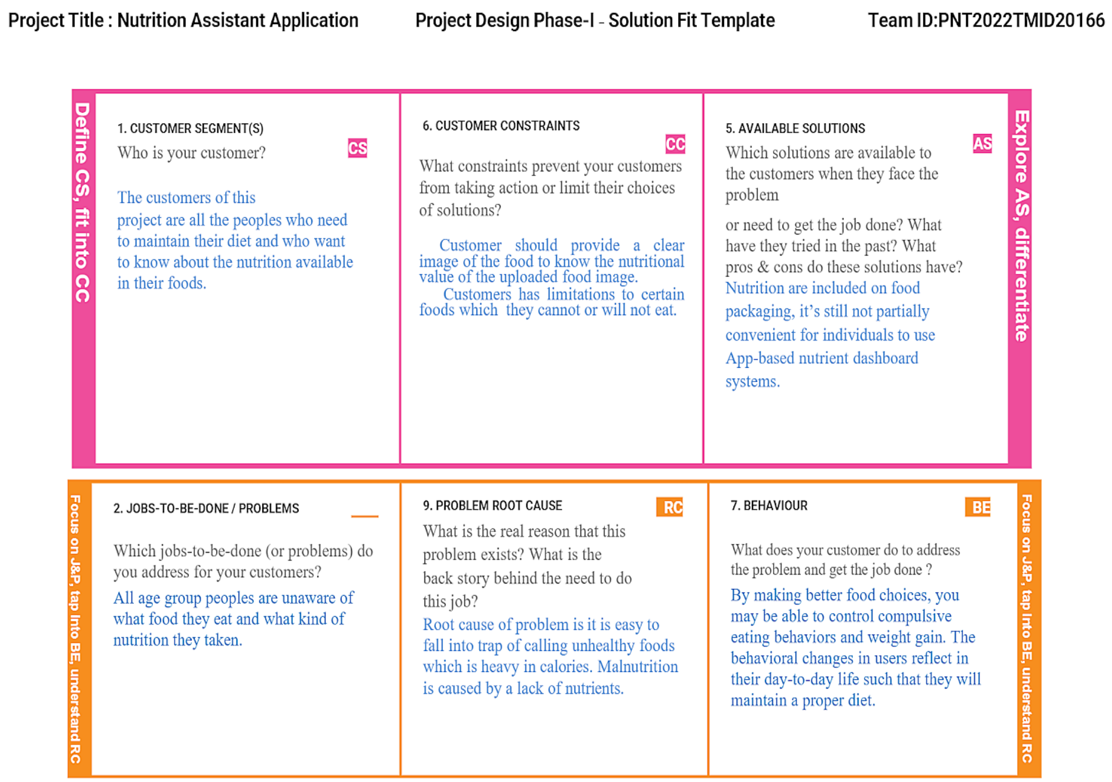
3.	Novelty / Uniqueness	This solution has the uniqueness that we can realize real time images of meals and can easily analyze its nutritional content. A web app that can automatically estimate food attributes such as ingredients and nutrition value by classifying the input image. Providing a user friendly environment. Prove recipes according to their diet.
4.	Social Impact / Custom er Satisfac tion	The Obesity rate will get reduced and people can able to lead a healthy life. It helps achieve and maintain a healthy life and healthy life.
5.	Business Model (Revenue Model)	Social media is the best way to develop this application. This application will increase the confidence among the people.

6.	Scalability of the Solution	1. People can access from anywhere at any time to track the calories and nutrition value that will improve a healthy eating pattern. 2. This App will improve dietary habits and helps in maintaining a healthy weight and healthy lifestyle. 3. Providing regular updates and upgradation. 4.The application is user friendly and interactive. 5.Enables users to access nutritional value of food.

3.4 PROBLEM SOLUTON FIT

To want to lead a healthy life, being aware of success stories of others who succeeded in their endeavors, by observing those who are in good health and shape.

By taking a picture of the food and uploading it to the app, users may learn the nutritional value of the food they are consuming. For precise food recognition and APIs that provide the discovered item’s nutritional value,Clarify uses its AI-Driven food detection model.



Project Title : Nutrition Assistant Application

Project Design Phase-I - Solution Fit Template

Team ID:PNT2022TMID20166

Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

CS

Who is your customer?

The customers of this project are all the peoples who need to maintain their diet and who want to know about the nutrition available in their foods.

6. CUSTOMER CONSTRAINTS

CC

What constraints prevent your customers from taking action or limit their choices of solutions?

Customer should provide a clear image of the food to know the nutritional value of the uploaded food image.

Customers has limitations to certain foods which they cannot or will not eat.

5. AVAILABLE SOLUTIONS

AS

Which solutions are available to the customers when they face the problem

or need to get the job done? What have they tried in the past? What pros & cons do these solutions have?

Nutrition are included on food packaging, it’s still not partially convenient for individuals to use App-based nutrient dashboard systems.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS

Which jobs-to-be-done (or problems) do you address for your customers?

All age group peoples are unaware of what food they eat and what kind of nutrition they taken.

9. PROBLEM ROOT CAUSE

RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?

Root cause of problem is it is easy to fall into trap of calling unhealthy foods which is heavy in calories. Malnutrition is caused by a lack of nutrients.

7. BEHAVIOUR

BE

What does your customer do to address the problem and get the job done ?

By making better food choices, you may be able to control compulsive eating behaviors and weight gain. The behavioral changes in users reflect in their day-to-day life such that they will maintain a proper diet.

Focus on J&P, tap into BE, understand RC

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIRMENT

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 better 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: In 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 persons/users who wants to learn about nutrition and other related health topics by just using the Nutrition Assistant Application

Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail And set a unique Password.
FR-2	User Login	User can login to the Application page whom already registered the registration.
FR-3	User Request	The user sends the request to the server to know about the food nutrition value and calories
FR-4	Server Response	The server sends the response to the user sent requested, that to display the nutrition value of the user sent image.
FR-5	User Details	First the user will send the request, (they took a picture and upload it to the page) then the server responds the user sent request (the server notify the picture and upload the nutrition value of the given picture).

4.2 NON-FUNCTIONAL REQUIREMENTS

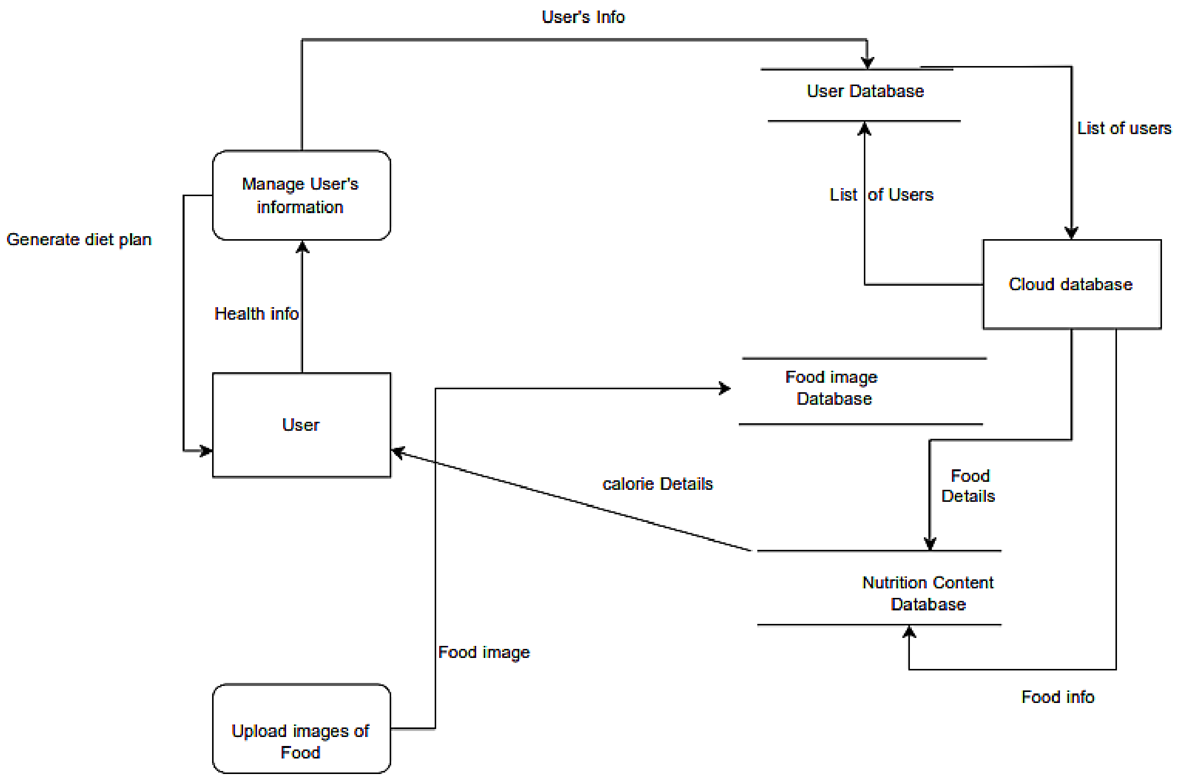
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Functional and Friendly UI. Better visualization tools to help the user to understand the importance of health and calorie details of consuming food.
NFR-2	**Security**	We maintain the normal secured protocol for the security systems. To prevent the user information's.
NFR-3	**Reliability**	Its so reliable to the users because the information we are providing on the web application is totally taken the nutrition consultant and some top references of the nutrition value pages.

NFR-4	**Performance**	Using cloud database for storage.Clarifai's AI-Driven Food Detection Model for accurate food identification and Food API's to give the nutritional value of the identified food.
NFR-5	**Availability**	Web app. Accessible from all devices. Its has any premium plans so its available for every user.
NFR-6	**Scalability**	Its all about the server system side and we are providing a service for a short scale of users. This application will run how long the server systems were going through.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION & TECHNICAL ARCHITECTURE

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic 1	Logic for a process in the application	Java / Python
3.	Application Logic 2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic 3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.

6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Hadar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Open source framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryption, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of applications (e.g. use of load balancers, distributed servers	Technology used

5.3 USER STORIES

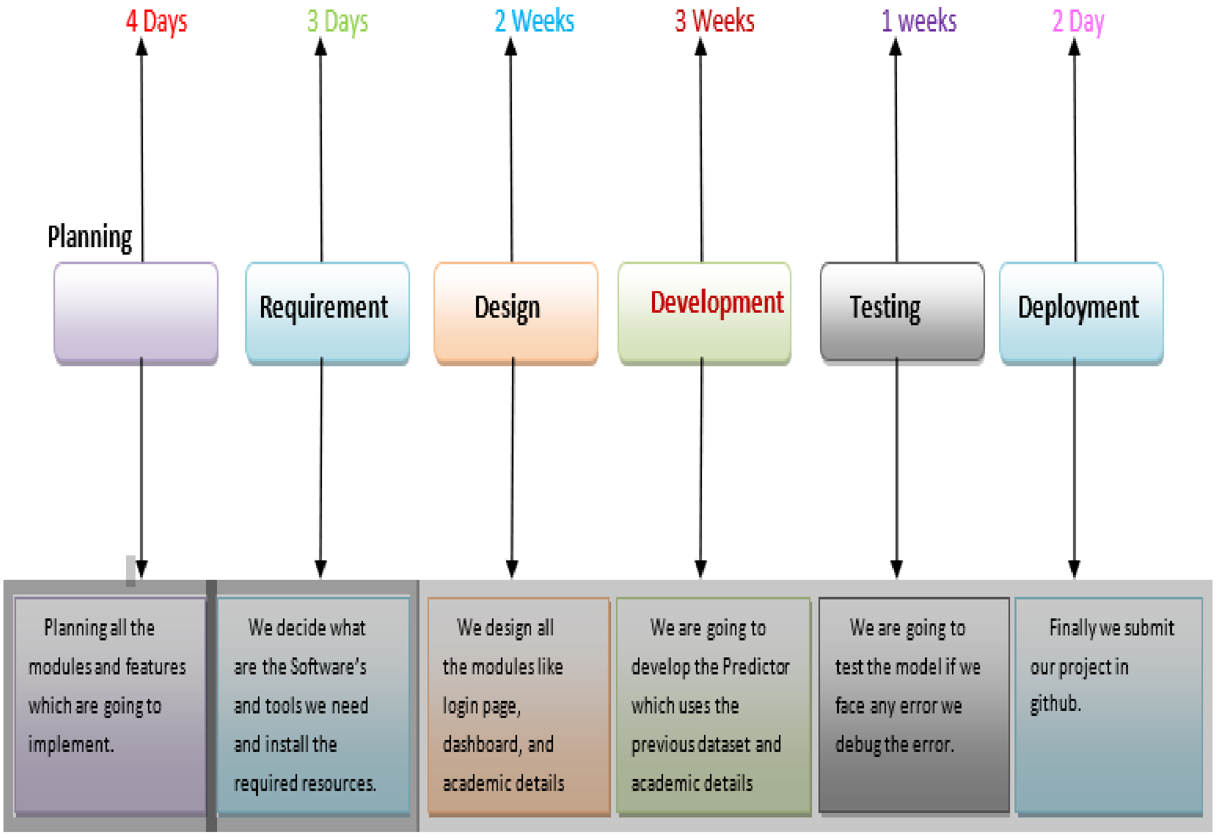
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Release
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Sprint-1
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Sprint-1
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email and password	1	High	Sprint-1
Sprint-2	User details	USN-4	The details of the user and the history of the searched food nutrition data will be available	2	High	Sprint-2
Sprint-3	Upload image of food	USN-5	The user will upload the food image to get nutrition details.	2	Medium	Sprint-3
Sprint-4	Shown the nutrition details and Recipe for	USN-6	The system will scanned the food image and display the nutrition detail and recipe for related scanned	1	High	Sprint-4

6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	2
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	1
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email and password	1	High	3
Sprint-2	User details	USN-4	The details of the user and the history of the searched food nutrition data will be available	2	High	4
Sprint-3	Upload image of food	USN-5	The user will upload the food image to get nutrition details.	2	Medium	3
Sprint-4	Shown the nutrition details and Recipe for	USN-6	The system will scanned the food image and display the nutrition detail and recipe for related scanned	1	High	4

6.2 SPRINT PLANNING AND ESTIMATION



Planning all the modules and features which are going to implement.

We decide what are the Software's and tools we need and install the required resources.

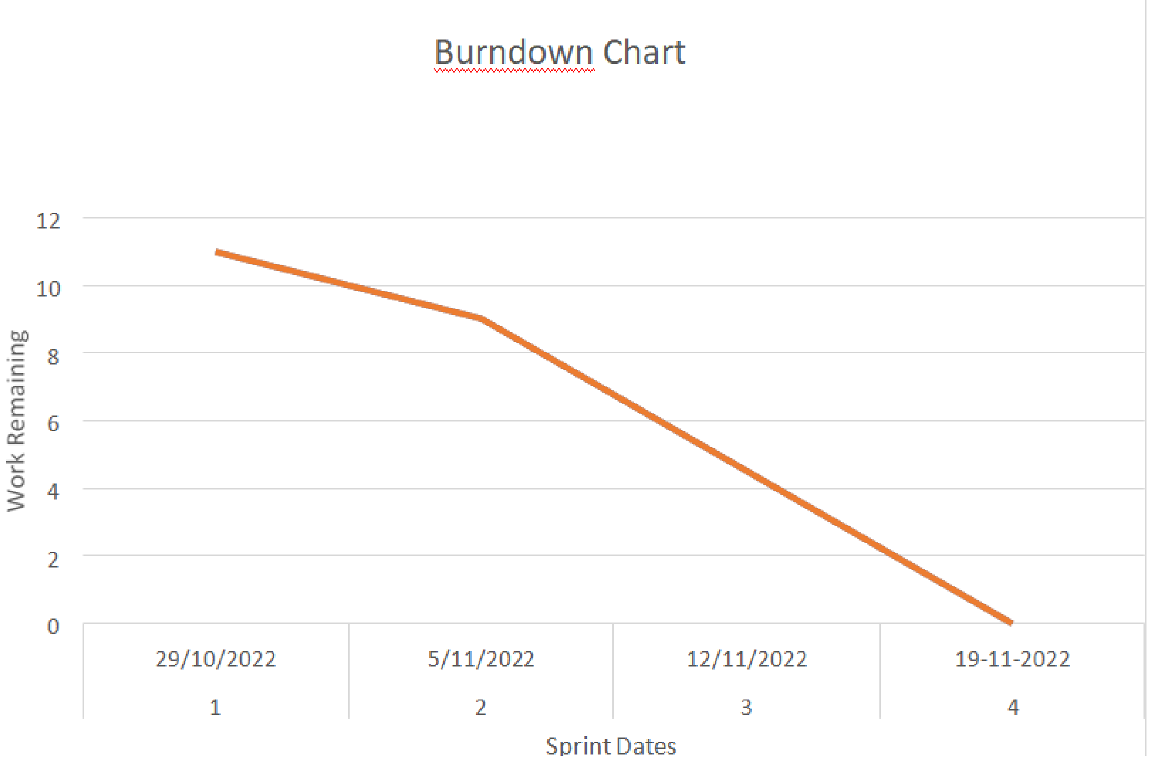
We design all the modules like login page, dashboard, and academic details

We are going to develop the Predictor which uses the previous dataset and academic details

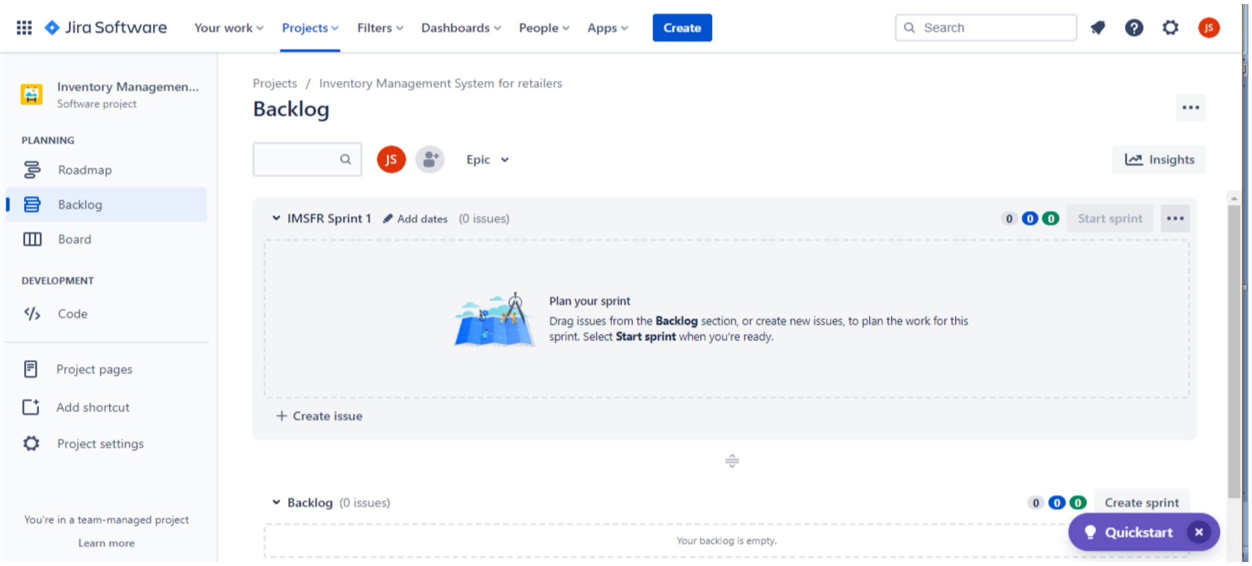
We are going to test the model if we face any error we debug the error.

Finally we submit our project in github.

6.2 Sprint Delivery Schedule



6.3 Reports from JIRA



Plan your sprint

Drag issues from the **Backlog** section, or create new issues, to plan the work for this sprint. Select **Start sprint** when you're ready.

Your backlog is empty.

You're in a team-managed project

Learn more

7. CODING AND SOLUTIONING

7.1 FEATURE 1

register:

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="register.css">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body background="img8.jpg">
<div class="container">
<div class="title">Registration</div>
<div class="content">
<form action="{{url_for('register')}}" method="POST" class="login email">
<div class="user-details">
<div class="input-box">
<span class="details">Full Name</span>
<input type="text" placeholder="Enter your name" name="fullname">
```

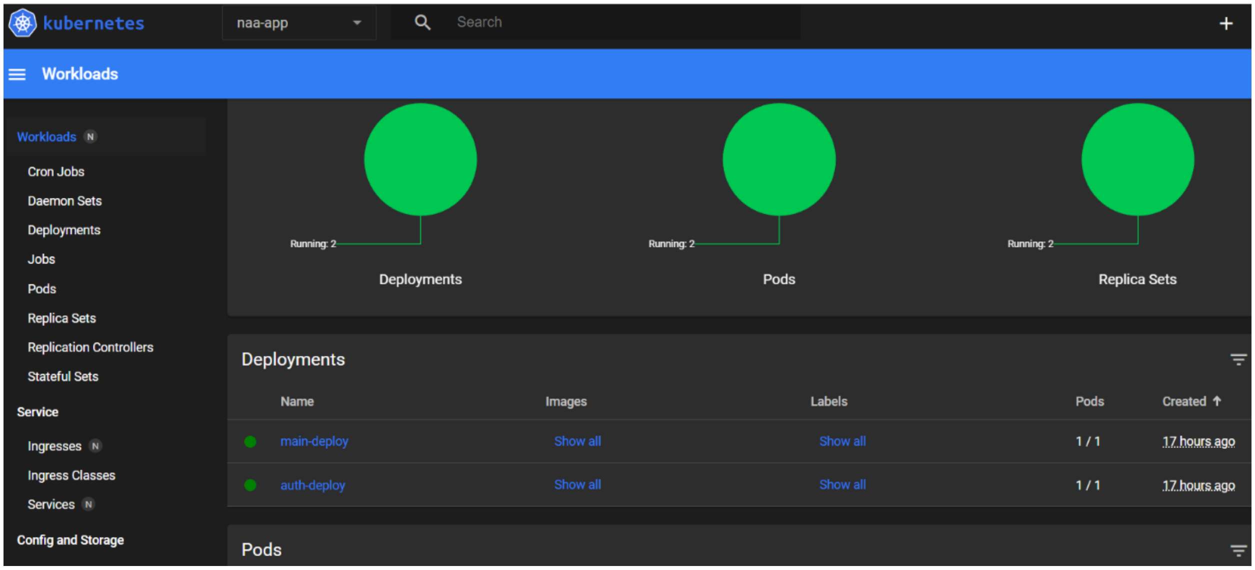
```
</div>
<div class="input-box">
<span class="details">Username</span>
<input type="text" placeholder="Enter your username"
name="username">
</div>
<div class="input-box">
<span class="details">Email</span>
<input type="text" placeholder="Enter your email" name="email">
</div>
<div class="input-box">
<span class="details">Phone Number</span>
<input type="text" placeholder="Enter your number"
name="phonenumber">
</div>
<div class="input-box">
<span class="details">Password</span>
<input type="password" placeholder="Enter your password"
name="passwords">
</div>
<div class="input-box">
<span class="details">Confirm Password</span>
<input type="password" placeholder="Confirm your password"
name="cpassword">
</div>
</div>
<div class="button">
 <a href="login.html"> <center>REGISTER </center></a>
<br><br>
already registered?
 <a href="login.html"> login </a>
</div>
</form>
</div>
</div>
</body>
</html>
```

7.2 FEATURE 2

Login

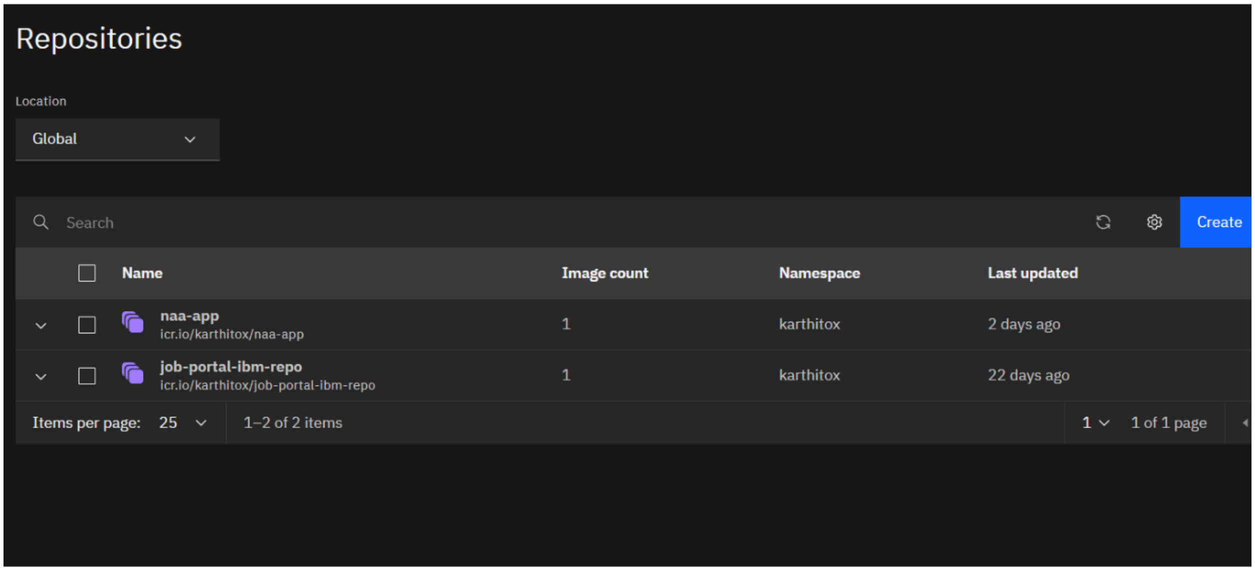
```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="UTF-8">
<link rel="stylesheet" href="register.css">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body background="img8.jpg">
<div class="container">
<div class="title">Login</div>
<div class="content">
<form action="{{url_for('register')}}" method="POST" class="login email">
<div class="user-details">
<div class="input-box">
<span class="details">Username</span>
<input type="text" placeholder="Enter your username"
name="username">
</div>
<br><br>
<div class="input-box">
<span class="details">Password</span>
<input type="password" placeholder="Enter your password"
name="passwords">
</div>
</div>
<div class="button">
<a href="USER DETAILS.html">
<center>SUBMIT</center></a>
<br><br>
not registered?
 <a href="register.html"> register </a>
</div>
</form>
</div>
</div>
</body>
</html>
```

7.3 DATABASE SCHEMA



Deployments

	Name	Images	Labels	Pods	Created ↑
●	main-deploy	Show all	Show all	1 / 1	17 hours ago
●	auth-deploy	Show all	Show all	1 / 1	17 hours ago



	□	Name	Image count	Namespace	Last updated
⌄	□	naa-app icr.io/karthitox/naa-app	1	karthitox	2 days ago
⌄	□	job-portal-ibm-repo icr.io/karthitox/job-portal-ibm-repo	1	karthitox	22 days ago

Items per page: 25 ⌄ 1–2 of 2 items 1 ⌄ 1 of 1 page

8. TESTING

The purpose of software testing is to access or evaluate the capabilities or attributes of a software program's ability to adequately meet the applicable standards and application need. Testing does not ensure quality and the purpose of testing is not to find bugs. Testing can be verification and validation or reliability estimation. The primary objective if testing includes:

- To identifying defects in the application.
- The most important role of testing is simply to provide information to

check the proper working of the application while inserting updating and deleting the entry of the products.

8.1 Type of Testing

We have used one type of testing to ensure the error free features of our software application:

8.2 User Acceptance Testing

This type of testing is the testing of individual software components. It is typically done by the programmer and not by the testers. It requires details information and knowledge about the internal program design and code to perform this. During unit testing, we carried out various testing task such as the reflection of the unit data on database and its interface. Various types of bugs associated with the component were identified and fixed. We use various functional keys to test our software. In our software unit testing concerns the stock units, opening stock units and product units validation as well as the validation of product units.

9. RESULTS

9.1 Performance Metrics

User satisfaction

For the ease of use for the end users, we have developed this website with a modular approach and clean UI. The end user can easily use the website with the modules. Also this website is responsive which makes it easier to use across all the devices.

Average response time

Since the website was made using Flask(Python) and it uses the IBM cloud to get the details as go, The response time is negligible.

Average Request time

The user inputs the data needed to process in the IBM cloud and then the api needs to fetch data, the processing speed at the IBM cloud will determine the request time or waiting time, The request time is low.

Error rate

All the data was taken from IBM datasets and we used comprehensive data pre-processing to avoid and eliminate the errors. This app is almost error free.

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Nutrition assistants help dieticians with providing proper nutrition at healthcare facilities.
- We can aware of nutrition about our in taking foods by our gadgets, there is no need of physical visit to nutrition or doctor consultant
- An advantage is control over what you eat. A nutrition program ensures you are eating what your body needs and limits the amount of unnecessary fat you may eat. A disadvantage is that you are not as free.
- It provides a maintained strategy of healthy eating habits.
- It delivers information on the nutritional value of foods and how balanced and healthy eating habits are important for us.
- It limits the amount of unnecessary food such as fat that people consume a lot.

DISADVANTAGES:

- Sometimes it makes a level of disbalance in the balanced diet of an individual.

- It can improve the level of nutrition among individuals but delivers an inappropriate means of nutritional labeling.

- Sometimes, it is considered one of the major factors of weight gain.

11. CONCLUSION

With people becoming conscious about their diets and fitness goals, there is a wide scope of nutrition assistant application thriving in the app world. Therefore, this time is perfect to create a nutrition assistant app of your own and enter the market with a unique idea in order to lure the audience towards your app. For developing a healthcare app, you must be sure of hiring the best team of experts who have prior experience in the same field and can guide you through the development process.

12. FUTURE SCOPE

This is the , wherein users can get experts to manage their diet and make diet charts for them on a regular basis. These experts not only share personalized diet charts but make it easier for the users by offering them healthy food recipes as well. These health and fitness experts provide their guidance on the basis of the body measurement report and the user’s goals.

Push notifications and alerts are essential to keep a user on track for his desired goal. Alerts remind the users about the activity they had to do at a particular time or a meal they had to take. If the user is looking to skip a meal or exercise, these reminders will spur them on to achieve their desired goals on a daily. Also if a new feature is added or some important information needs to be shared then also these alerts will play their part.

13. APPENDIX

SOURCE CODE

app.py

```
import is, re, string, random, time, date time, requests, sendgrid, random, flask
import ibm_db
from sendgrid.helpers.mail import *
from flask import Flask, request, render_template, flash, redirect, url_for, session
from werkzeug.utils import secure_filename
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2, resources_pb2, service_pb2_grpc
from clarifai_grpc.grpc.api.status import status_code_pb2



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

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



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



YOUR_CLARIFAI_API_KEY = "af4bc9886c744e998ee0e20f104b1518"
YOUR_APPLICATION_ID = "test"
SAMPLE_URL =
"https://res.cloudinary.com/swiggy/image/upload/f_auto,q_auto,fl_lossy/nxmlubuz0b1qixa29gov"
metadata = (("authorization", f"Key {YOUR_CLARIFAI_API_KEY}"),)
channel = ClarifaiChannel.get_grpc_channel()
stub = service_pb2_grpc.V2Stub(channel)
```

```
RAPIDAPI_KEY = "74e62205b6msha6b4e69e0088de5p12c619jsn1ed9cc5e0727"


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


def sendMail(to, title, text):
  sg = sendgrid.SendGridAPIClient(api_key=SENDGRID_API_KEY)
  from_email = Email("nsnandhaa1@gmail.com")
  to_email = To(to)
  subject = title
  content = Content("text/plain", text)
  mail = Mail(from_email, to_email, subject, content)
  response = sg.client.mail.send.post(request_body=mail.get())
  print(response.status_code)
  print(response.body)
  print(response.headers)


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

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

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

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


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

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

    dbcode = verify['CODE']
    if user code == dbcode:
      session['LoggedIn'] = True
      delete_sql = "DELETE FROM verify WHERE CODE=?"
      prep_stmt=ibm_db.prepare(conn, delete_sql)
      ibm_db.bind_param(prep_stmt, 1, db code)
      ibm_db.execute(prep_stmt)
      flash("Email verified. Enter new password")
      return redirect(url_for("changepw"))
    else:
      flash("Error")
      return render_template("confirm-mail")
  return render_template("confirm-mail.html")

@app.route("/change-pw", methods=["GET", "POST"])
def change():
  if flask.request.method == "POST" and session['LoggedIn']:
    data = flask.request.form
    password=data['pw']
    sql = "UPDATE users SET PASSWORD=? WHERE USERID=?"
    prep_stmt=ibm_db.prepare(conn, sql)
    print(password, session['userid'])
    ibm_db.bind_param(prep_stmt, 1, password)
    ibm_db.bind_param(prep_stmt, 2, session['userid'])
    ibm_db.execute(prep_stmt)
    flash("Password changed.")
    return redirect(url_for("login"))
  else:
    flash("verification error")
    redirect(url_for("confirmMail"))
  return render_template("change-pw.html")


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

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

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

    return redirect(url_for("dashboard"))
  return render_template("reg.html")

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

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

    sql = "SELECT * FROM users WHERE username=? AND password=?"
    stmt = ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt, 1, username)
    ibm_db.bind_param(stmt, 2, password)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)
    print(account)
    if account:
      session['LoggedIn'] = True
      session['userid'] = account['USERID']
      session['username'] = account['USERNAME']
      userid = account['USERID']
      flash("logged in")
      return redirect(url_for("dashboard"))
    else:
      flash("error")

  return render_template("login.html")
@app.route("/dashboard", methods=["GET", "POST"])
def dashboard():
  global request
  if flask.request.method == "POST" and session['LoggedIn']:
    if 'file' not in flask.request.files:
      flash('No file part')
```

```
   return redirect(flask.request.url)
  file = flask.request.files['file']
  if file.filename == ":
   flash('No image selected')
   return redirect(flask.request.url)
  if file and allowed_file(file.filename):
   filename = secure_filename(file.filename)
   file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
   flash('Image successfully uploaded')

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

   request = service_pb2.PostModelOutputsRequest(
     model_id="food-item-v1-recognition",
     user_app_id=resources_pb2.UserAppIDSet(app_id=YOUR_APPLICATION_ID),
     inputs=[
      resources_pb2.Input(
       data=resources_pb2.Data(image=resources_pb2.Image(
          base64=file_bytes
        )
       )
      )
     ],
   )
   response = stub.PostModelOutputs(request, metadata=metadata)

   if response.status.code != status_code_pb2.SUCCESS:
     print(response)
     raise Exception(f"Request failed, status code: {response.status}")

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

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

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

```
# headers = {
#   "X-RapidAPI-Key": RAPIDAPI_KEY,
#   "X-RapidAPI-Host": "spoonacular-recipe-food-nutrition-v1.p.rapidapi.com"
# }
# response = requests.request("GET", nutritionApiUrl, headers=headers, params=querystring)
# nutritions = response.text
nutrition = {
  "recipesUsed": 10,
  "calories": {
    "value": 470,
    "unit": "calories",
    "confidenceRange95Percent": {
      "min": 408.93,
      "max": 582.22
    },
    "standardDeviation": 139.8
  },
  "fat": {
    "value": 17,
    "unit": "g",
    "confidenceRange95Percent": {
      "min": 12.81,
      "max": 21.36
    },
    "standardDeviation": 6.9
  },
  "protein": {
    "value": 15,
    "unit": "g",
    "confidenceRange95Percent": {
      "min": 9.06,
      "max": 29.78
    },
    "standardDeviation": 16.71
  },
  "carbs": {
    "value": 65,
    "unit": "g",
    "confidenceRange95Percent": {
      "min": 57.05,
      "max": 77.9
```

```
     },
     "standardDeviation": 16.81
    }
   }
   nutritions.pop('recipesUsed')
   for i in nutrition:
    nutritionValues += f"{i}: {nutritions[i]['value']} {nutritions[i]['unit']},
   sql = "INSERT INTO foods VALUES(?,?,?,?,?)"
   stmt=ibm_db.prepare(conn, sql)
   ibm_db.bind_param(stmt, 1, session['userid'])
   ibm_db.bind_param(stmt, 2, datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S'))
   ibm_db.bind_param(stmt, 3, food name)
   ibm_db.bind_param(stmt, 4, ingredients)
   ibm_db.bind_param(stmt, 5, nutritionValues)
   ibm_db.execute(stmt)
   # os.remove(os.path.join(app.config['UPLOAD_FOLDER'], filename))
   return render_template("dashboard.html",
    filename = filename,
    username = session['username'],
    food name = foodname,
    ingredients = ingredients,
    nutritionValues = nutritionValues,
   )
  else:
   flash('Allowed image formats - png, jpg, jpeg')
   return redirect(flask.request.url)

 elif session['LoggedIn']:
  return render_template("dashboard.html", username=session['username'])
 else:
  return redirect(url_for("login"))
@app.route('/logout', methods=["GET", "POST"])
def logout():
 session.pop('LoggenIn', None)
 session.pop('userid', None)
 session.pop('username', None)
 return render_template("index.html")
@app.route('/display/<filename>', methods=["GET", "POST"])
def display(filename):
 print(filename)
 return redirect(url_for('static', filename='uploads/' + filename), code=301)
```

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

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

Deploy:

----- remove debug

----- install docker

----- install ibmcloud cli

```
docker build -t nut-ass .

docker images

docker run -p 5000:5000 nut-ass

Ibmcloud plugin install container-registry

Ibmcloud plugin install container-service

ibmcloud login

Ibmcloud cr login --client docker

Ibmcloud cr namespace-add nut-space

docker tag nut-ass icr.io/nut-space/nut-ass:latest

docker push icr.io/nut-space/nut-ass:latest

Ibmcloud cr image-list
```

```
ibmcloud ks cluster con-fig --cluster cde0ac7f066vuobgl68g
```

```
kubectl apply -f deployment.yaml
```

```
Ibmcloud cs workers --cluster cde0ac7f066vuobgl68g
```

```
kubectl describe service nut-ass | grep NodePort
```

```
http://<public-ip-address>:<port>/app
```

```
kubectl delete deploy nut-ass
```

```
ibmcloud cs cluster rm -c cde0ac7f066vuobgl68g -f
```

Upload Image:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="static/styles.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
  <title>Nutrition Assistant</title>
</head>
<body style="background-color:pink"><center>
   <div class="row align-items-md-stretch">
      <div class="col-md-6 my-3">
        <div class="h-100 p-5 text-bg-dark rounded-3">
          <h2>Upload food image</h2>
          <form action = "/dashboard" method = "POST" enctype="multipart/form-data">
            <input class="my-3 form-control" type="file" name="file" required/>
            <a href="food details.html"<center>ANALYZE </center></a>
          </form>
        </div>
      </div>
    </div>
  </div>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
crossorigin="anonymous"></script>
</body>
</html>
```

Food Details:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="static/styles.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"
crossorigin="anonymous">
  <title>Nutrition Assistant</title>
</head>
<body background="img1.webp">
<h1><center>FOOD DETAILS</center></h1>
<h3><center><b>FOOD NAME</b>: </center></h3>
<h3><li><b>INGREDIENTS: </b></li></h3>
<h4><li><ol></ol></li></center></h4>
<h4><li><ol></ol></li></h4>
<h4><li><ol></ol></li></h4>
<h4><li><ol></ol></li></h4>
<h4><li><ol></ol></li></h4>
<h4><li><ol></ol></li></h4>
<h4><li><ol></ol></li></h4>
<h4><li><b>NUTRITION VALUE: </b></li></h4>
<h3><li><ol>CALORIES : </ol></li> </h3>
<h4><li><ol>CALORIES FROM FAT :  </ol></li> </h4>
<h4><li><ol>CHOLESTEROL 54mg : </ol></li> </h4>
<h4><li><ol>TOTAL CARBOHYDRATES 18g : </ol></li> </h4>
<h4><li><b>PROTEIN 15g : </b></li> </h4>
<h4><li><ol>VITAMIN A : </ol></li></h4>
<h4><li><ol>VITAMIN C : </ol></li></h4>
<h4><li><ol>CALCIUM : </ol></li></h4>
<h4><li><ol>IRON : </ol></li></h4>
```

GitHub Link:
https://github.com/IBM-EPBL/IBM-Project-24727-1659947924

Demo Video Link:
https://drive.google.com/file/d/1TiXywPoP4KW9DUlXOuWYs7tVUII6j9eC/view?usp=share_link