IBM NALAIYATHIRAN PROJECT REPORT

NUTRITION ASSISTANT APPLICATION

Team Id	**PNT2022TMID02079**
Project Name	**Nutrition Assistant Application**
Team Members	**- Abhijeet Venugopalan(2116190701002)** **- Aravindh R(2116190701021)** **- Abhijith PV (2116190701003)** **- Arumugam Madan(2116190701022)**

Table Of Contents

1.INTRODUCTION

1.1 Project Overview

The goal of this project is to create the desktop-based Nutrition Assistant Application, which classifies input food photographs to estimate food features like ingredients and nutritional value. The term "Nutrition Assistant Application" refers to a technology-based system and set of procedures that assist users in analysing their dietary intake. The user's health information can be stored in this system, which can also calculate BMI, classify food images to determine their nutritional worth, update the user's health condition based on the information provided, and produce health reports on a weekly or monthly basis. This initiative is classifying each user's unique health situation. The most fundamental way to prevent obesity is to limit daily caloric intake by choosing healthier foods, hence the Nutrition Assistant Application is crucial. Without appropriate nutrition regulation, there are threats to people's health. When it is time to avoid, an excellent nutrition assistant app will let the users know. By identifying the supplied food image, this project attempts to create a web application that automatically calculates food qualities like ingredients and nutritional value.

Wondershare PDFelement

Scope:

- ❒ **Maintains good health**: The application can help in guiding them on how to remain healthy and how to take good nutrition. The application will help them without personally going to the doctor. Promote better nutrition in the community by educating about 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.

1.2 Purpose

Users still insist on learning how much nutrition their food contains. Users get knowledge about how various diets affect human health. This application's major goal is obviously to show users how to manage their diets while living healthy lives. By snapping a photo of the item, uploading a photo from the gallery, or manually entering the data, the user can obtain the nutritional information.

More than merely consuming calories and nutrients from food is nutrition. It involves more than just consuming nutritious foods. It involves more than just sticking to the newest diet trend. Life would not be possible without nutrition, which includes the food we eat and how we eat it. Food is a sensory experience.

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

Wondershare PDFelement

2.LITERATURE SURVEY

2.1 EXIXTING PROBLEM

The user who want to maintain a healthy and fit body but have no one to guide them on their dieting

The user wants to develop a deep learning model that basically help athlete, body builders or other game players to keep up with their health and fitness by suggesting them with proper nutrition plan

The user wants an AI software to maintain healthy diet rather than having a physical consultant

People who are obese and overweight are more likely to have high-risk factors for heart disease, diabetes, hypertension. The goal of the application is to create a healthy lifestyle for its user.

User has obesity who needs to follow diet to improve his health without the feeling that he's following the diet.

2.2 REFERENCE

S. NO	TITLE	PROPOSED WORK	SOFTWARE USED	TECHNOLOGY	ADVANTAGES/ DISADVANTAGES
1	Personalized Dietary Assistant-an intelligent space application	Observes the daily consumption habits of users& apply datamining to learn their taste.	Distributed intelligent networked devices (DINDS)	Data mining Artificial intelligence	Complex to calculate the entire nutritionalvalue.

2	Profile Based System for Nutritional Information Management	Nutritional control by identifying the person's Shopping profile & usesthe information to suggest the recommended food.	Mobile Application server	Information technology	An automated way to assist users to control food intake.
3	Intelligent Diabetes Assistant	The system collects & processes thedata more efficiently for the care team.	Intelligent Diabetes Assistant (IDA)	Machine Learning	Quick evaluationof patient health.
4	Alexa, what should I Eat? A personalized Virtual nutrition coach for Native American Diabetes Patient	to design an Amazon skill to extend the capability of Amazon Alexa to support diabetes forNA users.	Amazon Alexa software Development kit	Artificial Intelligence	Easy to acceptedby the target audience withthe help of speech- recognition
5	Plan-Cook-Eat: A Meal planner app with optimal macronutrien t distributionof calories based on total daily energy expenditure.	"Plane-cook - eat" A progressive web application that generates meal plan complaints tothe necessary macronutrient distribution of daily calories based on individuals' total daily energy expenditure (TDEE).	AQEL- A nutrition app quality evaluation tool. Data Sources: Food DB, Philippine Food Composition Table, My Food Data, USDA food composition data	Artificial Intelligence, Data analysis.	Generation of meal plan with optimal macronutrient distribution of daily calories.

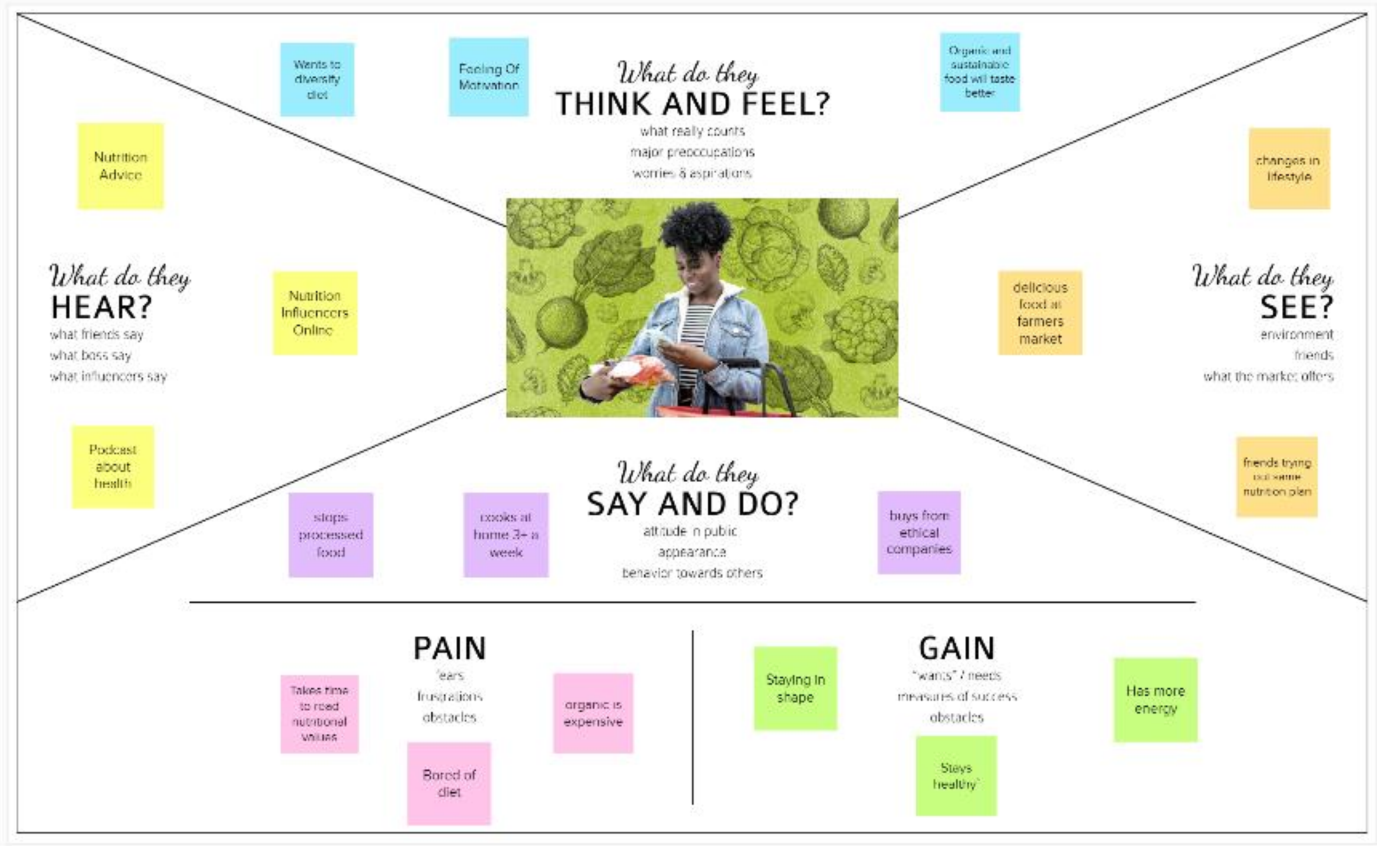
2.3 Problem Statement Definition

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. 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.

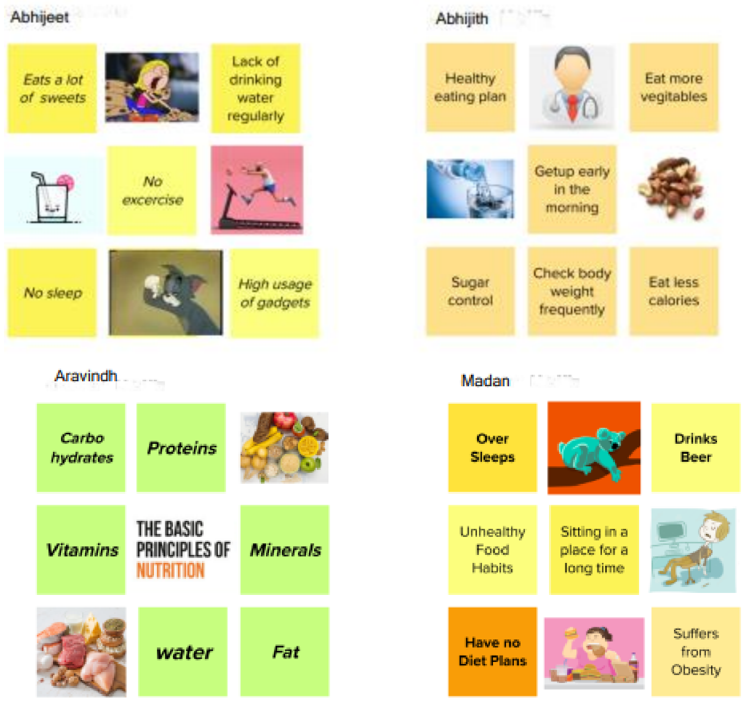
.

3.IDEATION AND PROPOSED SYSTEM

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming

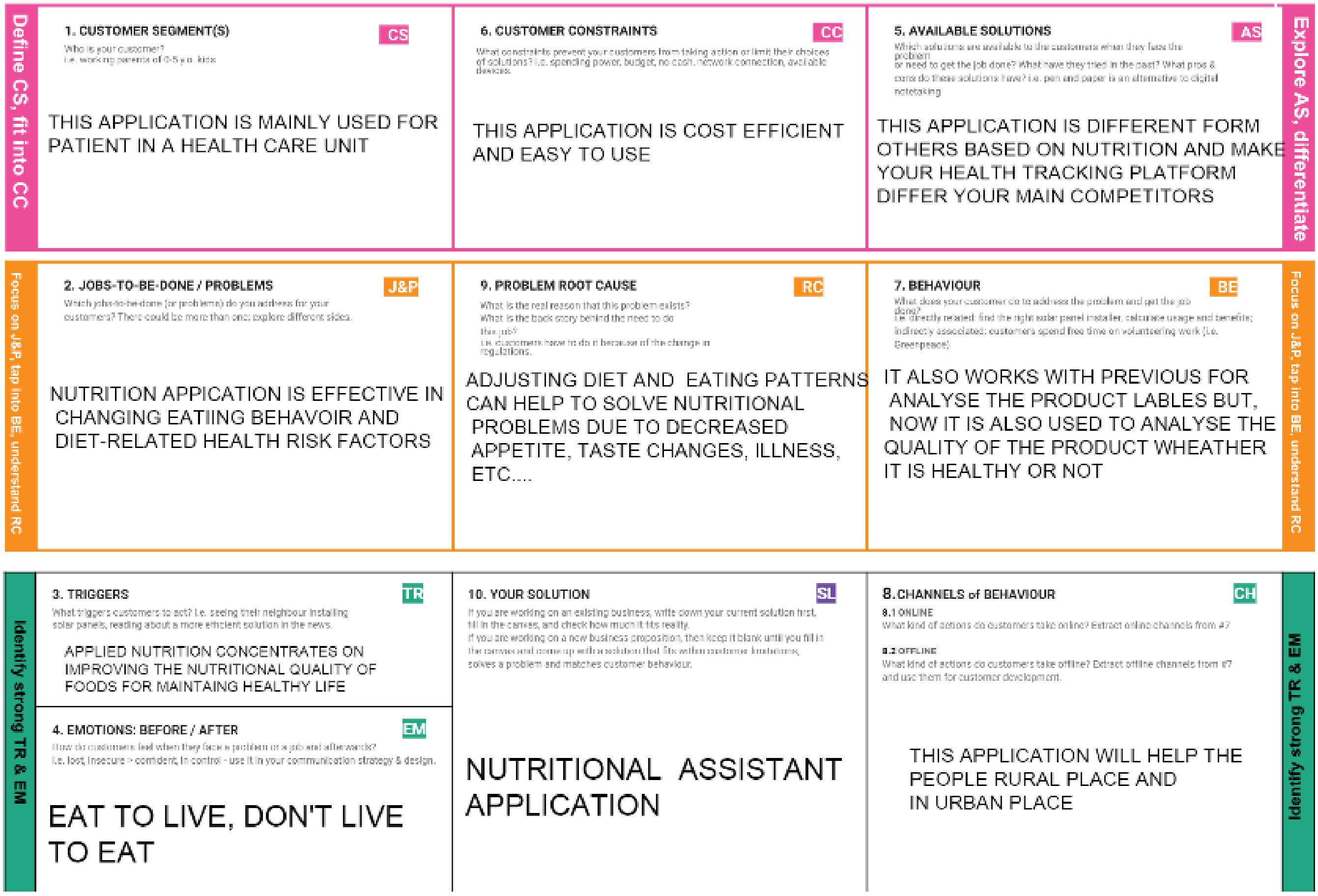


3.3 Proposed Solution

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	✓ Lack Of Basic Infrastructure. ✓ Insufficient Facilities of Doctor Consultancy. ✓ We can’t Buy The Medicines Using This Assistant Application.
1.	Idea / Solution description	✓ Prevent Avoidable Losses. ✓ Limited Fast foods. ✓ High Fibre Foods. ✓ Healthy Source Of Proteins, Seafood and Nuts.
2.	Novelty / Uniqueness	✓ Plant Based Foods. ✓ Healthy, Safe, Living conditions. ✓ Using global food matters Database. ✓ Patients to More Easily Monitor their caloric intake and dietary Pattern to aid in weight and disease Management.

3.	Social Impact / Customer Satisfaction	✓ Subsidies And Piece Promotions on Healthy Food. ✓ Nutrition Focussed Food Banking. ✓ Corporate Social Responsibility initiatives to Increase dignified Access to Healthy food. ✓ Targeted Food Assistance Programs.
4.	Business Model (Revenue Model)	✓ Collaboration Other consultancies. ✓ Customized Nutrition and consumption Requirements. ✓ Health-Trade-Policy.
5.	Scalability of the Solution	✓ Easily Access the Application. ✓ Good Relationship.

3.4Problem Statement Fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

THIS APPLICATION IS MAINLY USED FOR PATIENT IN A HEALTH CARE UNIT

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

THIS APPLICATION IS COST EFFICIENT AND EASY TO USE

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? i.e. pen and paper is an alternative to digital notetaking

THIS APPLICATION IS DIFFERENT FORM OTHERS BASED ON NUTRITION AND MAKE YOUR HEALTH TRACKING PLATFORM DIFFER YOUR MAIN COMPETITORS

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

NUTRITION APPICATION IS EFFECTIVE IN CHANGING EATIING BEHAVOIR AND DIET-RELATED HEALTH RISK FACTORS

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?
i.e. customers have to do it because of the change in regulations.

ADJUSTING DIET AND EATING PATTERNS CAN HELP TO SOLVE NUTRITIONAL PROBLEMS DUE TO DECREASED APPETITE, TASTE CHANGES, ILLNESS, ETC....

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

IT ALSO WORKS WITH PREVIOUS FOR ANALYSE THE PRODUCT LABLES BUT, NOW IT IS ALSO USED TO ANALYSE THE QUALITY OF THE PRODUCT WHEATHER IT IS HEALTHY OR NOT

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

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

APPLIED NUTRITION CONCENTRATES ON IMPROVING THE NUTRITIONAL QUALITY OF FOODS FOR MAINTAING HEALTHY LIFE

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

EAT TO LIVE, DON'T LIVE TO EAT

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

NUTRITIONAL ASSISTANT APPLICATION

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

THIS APPLICATION WILL HELP THE PEOPLE RURAL PLACE AND IN URBAN PLACE

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

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 demonstratehealth 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.
12. Add multiple accounts	This application has the option of creating multiple accounts for the users.
13. Selection of health report duration	This application has the ability to select the duration for displaying the health report as weekly or monthly.

14. Update account	This application will allow the user to update their profile.
15. Add account	This application will allow the user to add their profile.
16. Delete account	This application will allow the user to delete their profile.
17. PDF report	This application will generate the pdf report of medical analysis.
18. Pupation of nutritional trends	This application will allow constant review of nutritional trends and pupation.

4.2 Non-functional Requirements:

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

Security: User's information and their nutrition content are secured.

Performance: The prediction process begins when the image is uploaded. It performs

According to the sources of the image and provides the specific nutritional information.

Reliability: The contents provided are based on the nutritional value based on the image

The image upload consistently performs well.

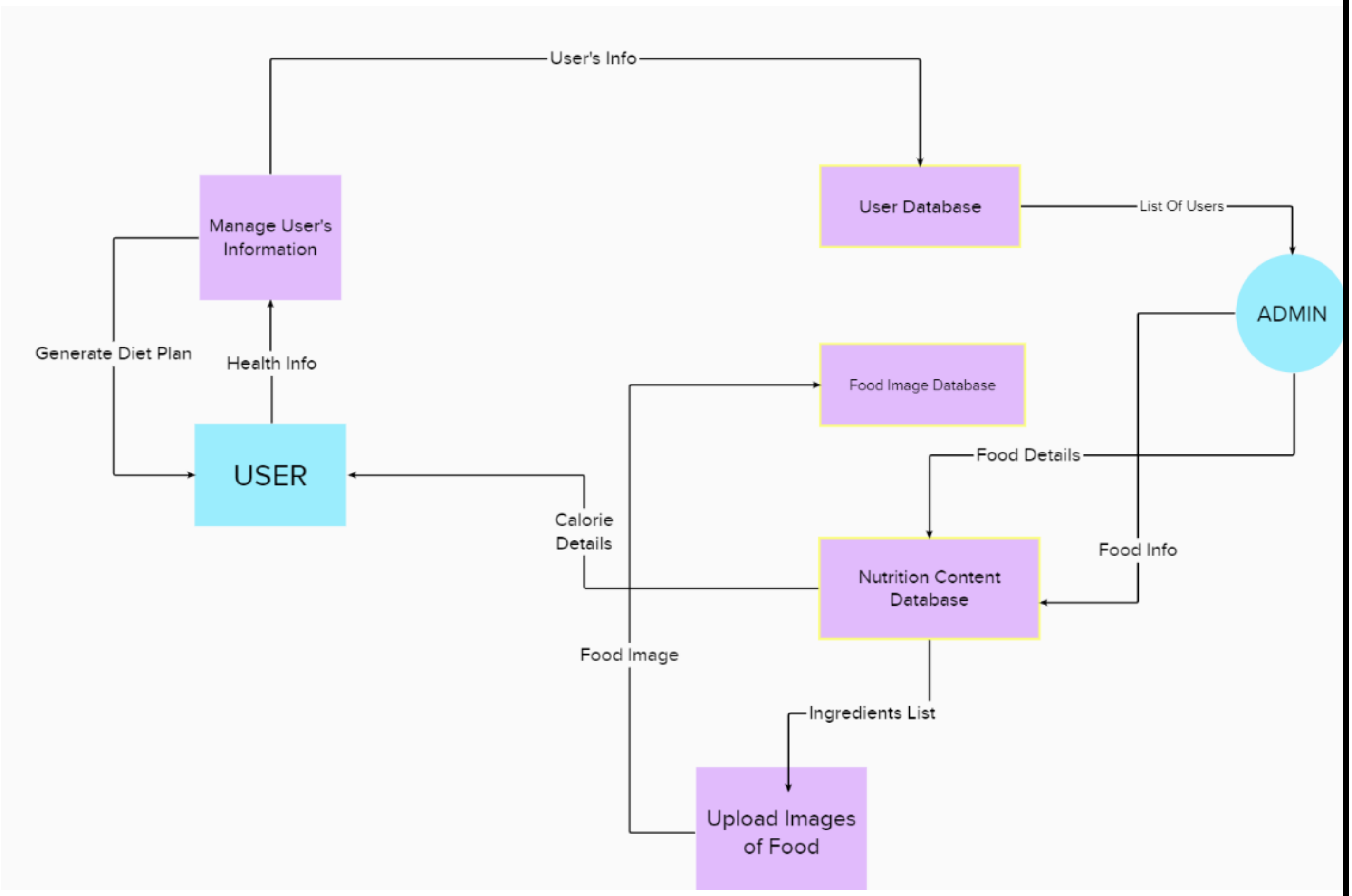
Availability: The image uploaded is used for prediction of nutritional value.

Scalability: Increasing the accuracy of nutritional value prediction of the image uploaded.

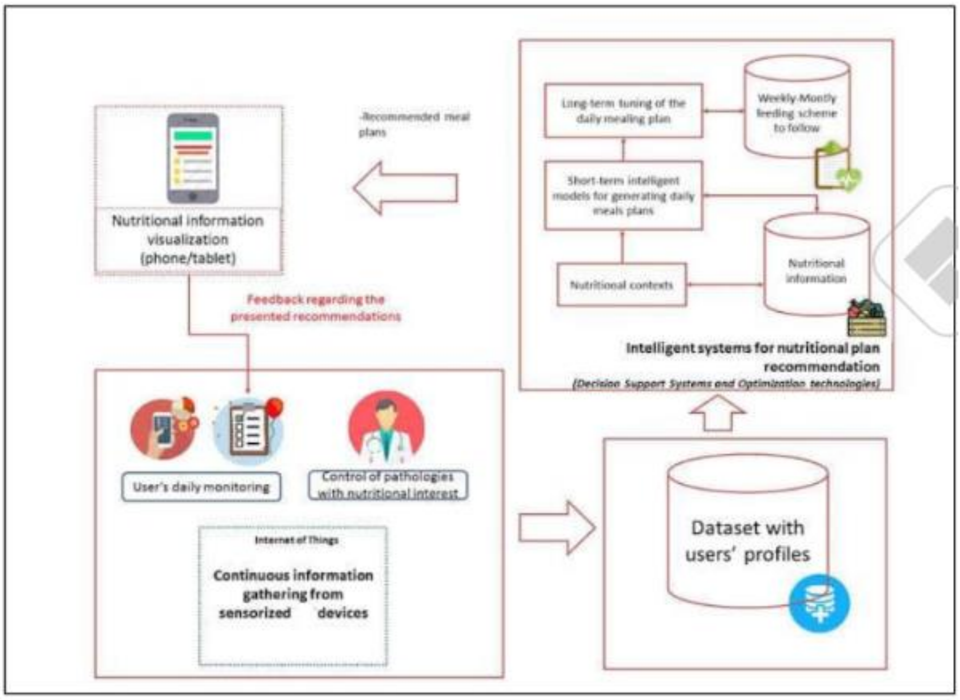
Usability: Nutritional contents are provided based on the image uploaded by clarifai's AI

5.PROJECT DESIGN

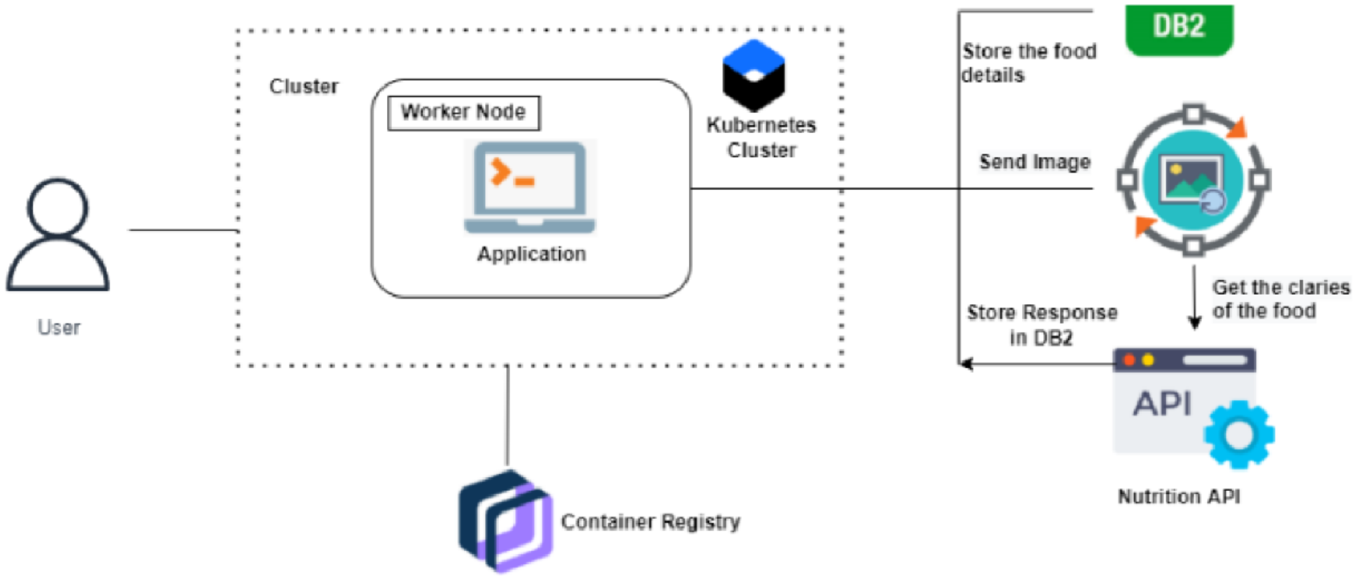
5.1 Data Flow Diagram:



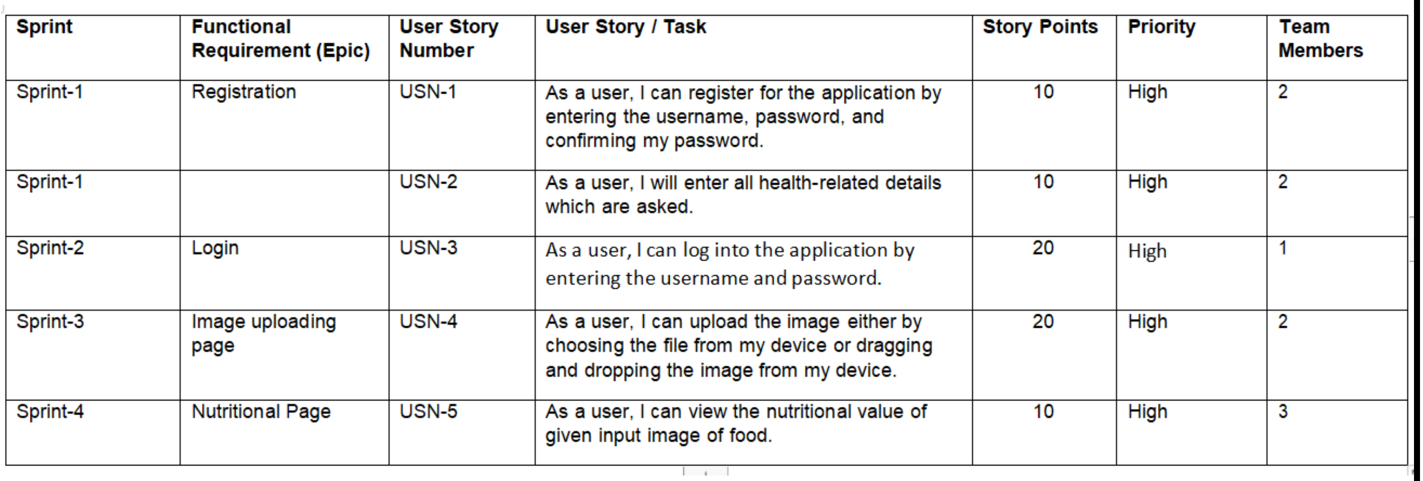
5.2 Solution and Technical Architecture:



Nutritional information visualization (phone/tablet)
-Recommended meal plans
Long-term tuning of the daily mealing plan
Weekly-Montly feeding scheme to follow
Short-term intelligent models for generating daily meals plans
Nutritional contexts
Nutritional information
Intelligent systems for nutritional plan recommendation
(Decision Support Systems and Optimization technologies)
Feedback regarding the presented recommendations
User's daily monitoring
Control of pathologies with nutritional interest
Internet of Things
Continuous information gathering from sensorized devices
Dataset with users' profiles



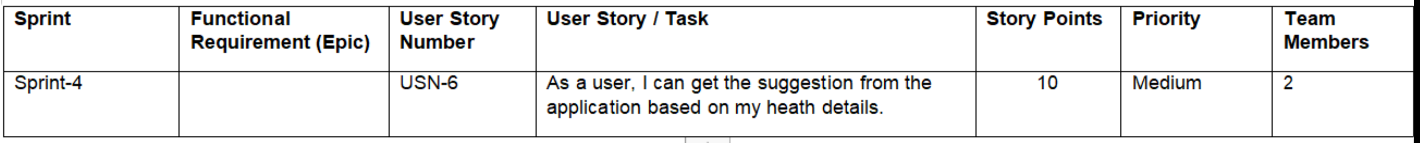
5.3 User Stories:



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 the username, password, and confirming my password.	10	High	2
Sprint-1		USN-2	As a user, I will enter all health-related details which are asked.	10	High	2
Sprint-2	Login	USN-3	As a user, I can log into the application by entering the username and password.	20	High	1
Sprint-3	Image uploading page	USN-4	As a user, I can upload the image either by choosing the file from my device or dragging and dropping the image from my device.	20	High	2
Sprint-4	Nutritional Page	USN-5	As a user, I can view the nutritional value of given input image of food.	10	High	3

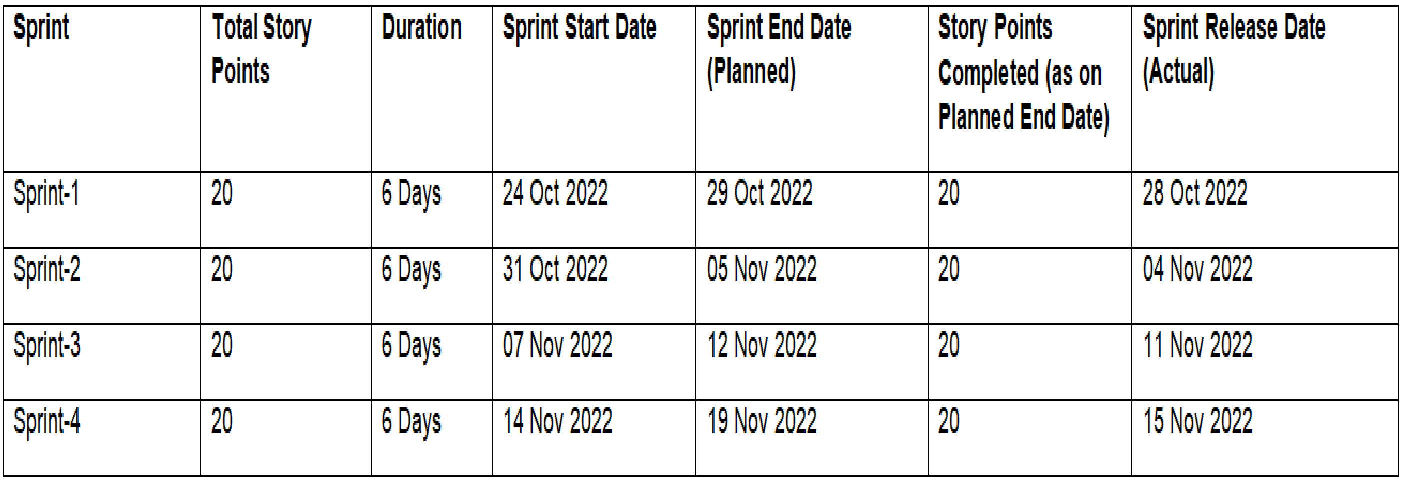
6.PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning & Estimation



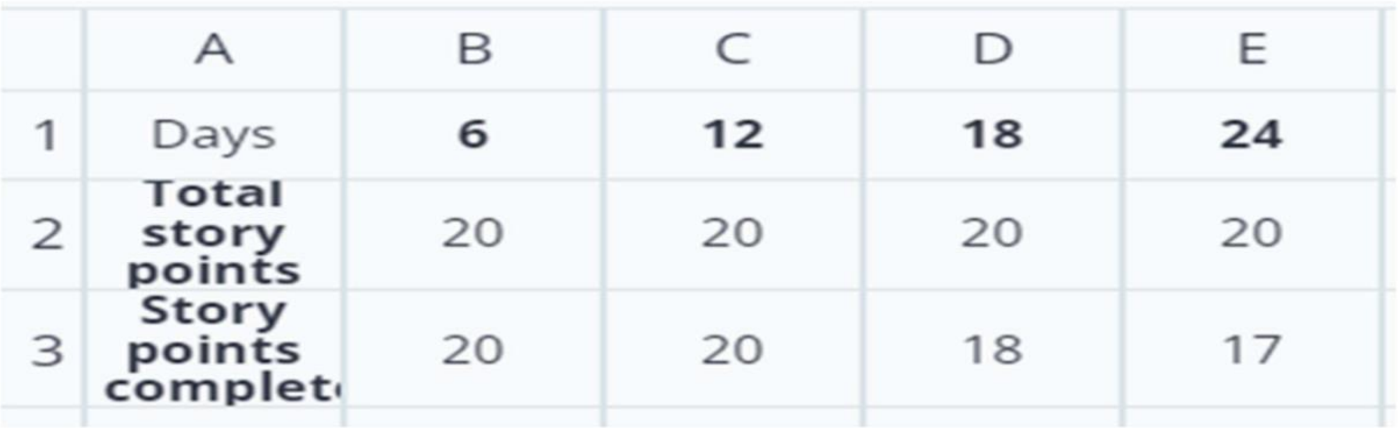
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4		USN-6	As a user, I can get the suggestion from the application based on my heath details.	10	Medium	2

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring to technical papers, research publications etc.	1 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	25 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	8 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	30 SEPTEMBER 2022



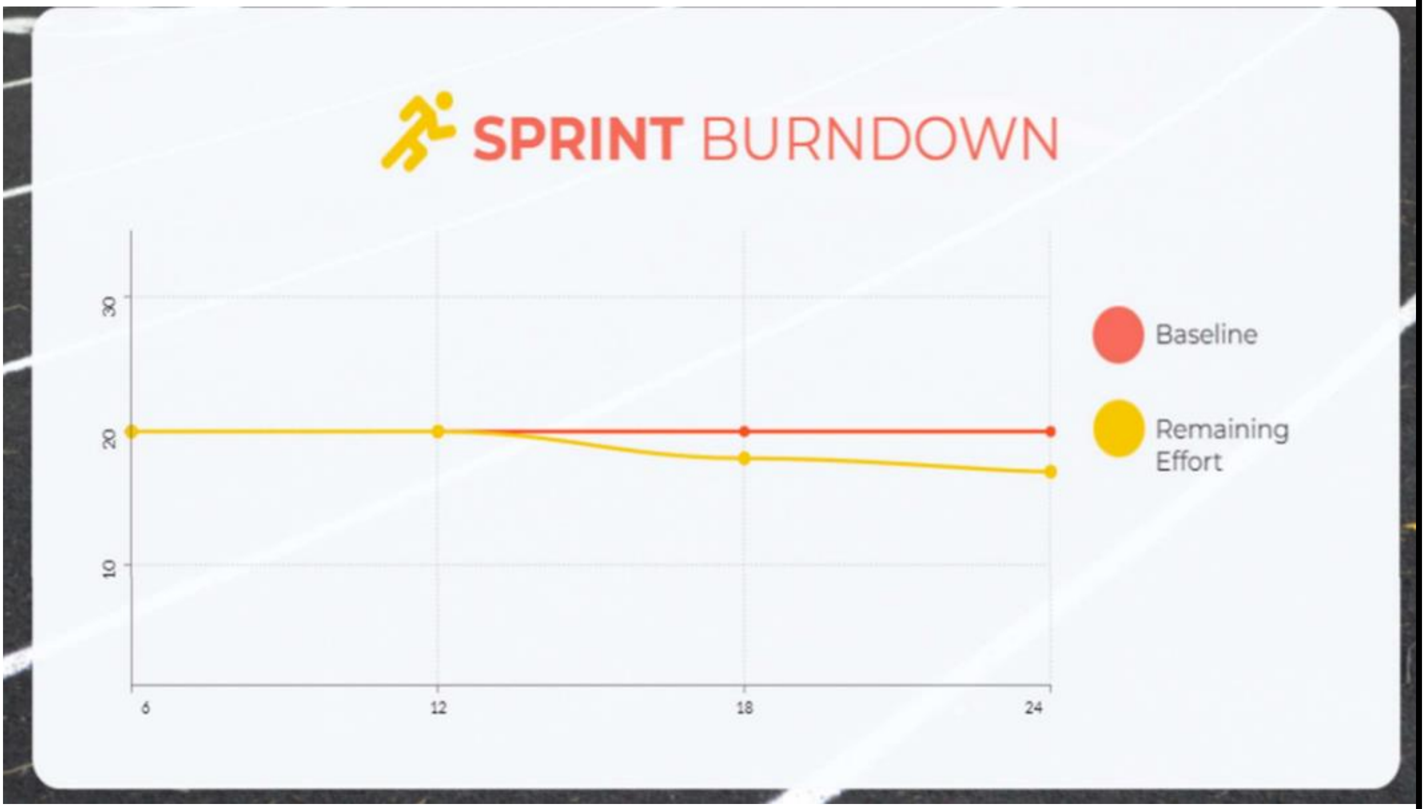
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	28 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	04 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	11 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	15 Nov 2022

6.2 Sprint delivery schedule



	A	B	C	D	E
1	Days	**6**	**12**	**18**	**24**
2	**Total story points**	20	20	20	20
3	**Story points complet**	20	20	18	17

6.3 Reports from JIRA



7.CODING & SOLUTIONING

7.1 Feature 1:

Python

- Python is a widely-used, interpreted, object-oriented, and high-level programming language with dynamic semantics, used for generalpurpose programming. It’s everywhere, and people use numerous Python-powered devices on a daily basis, whether they realize it or not.
- Python was created by Guido van Rossum, and first released on February 20, 1991.
- Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and Unix shell and other scripting languages.
- Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL)
- It is easy to learn – the time needed to learn Python is shorter than for many other languages; this means that it’s possible to start the actual programming fast
- It is easy to use for writing new software – it’s often possible to write code faster when using Python.
- It is easy to obtain, install and deploy – Python is free, open and multiplatform; not all languages can boast that.
- Programming skills prepare you for careers in almost any industry and are required if you want to continue to more advanced and higherpaying software development and engineering roles.
- Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

7.2 Feature 2:

Flask

- **Flask** is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.
- It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.
- Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.
- Applications that use the Flask framework include Pinterest and LinkedIn.

7.3 Database Scheme

IBM Db2

- DB2 is a database product from IBM.
- It is a Relational Database Management System (RDBMS). DB2 is designed to store, analyze and retrieve the data efficiently.
- DB2 product is extended with the support of Object-Oriented features and non-relational structures with XML.
- Provide a massively parallel processing (MPP) architecture Exploits Hive, HBase and Apache Spark concurrently for best-in-class analytic capabilities.
- Provides low latency support for ad-hoc and complex queries, high performance, and federation capabilities Understands dialects from other vendors and various products from Oracle, IBM® Db2® and IBM Netezza® Enables advanced row and column security

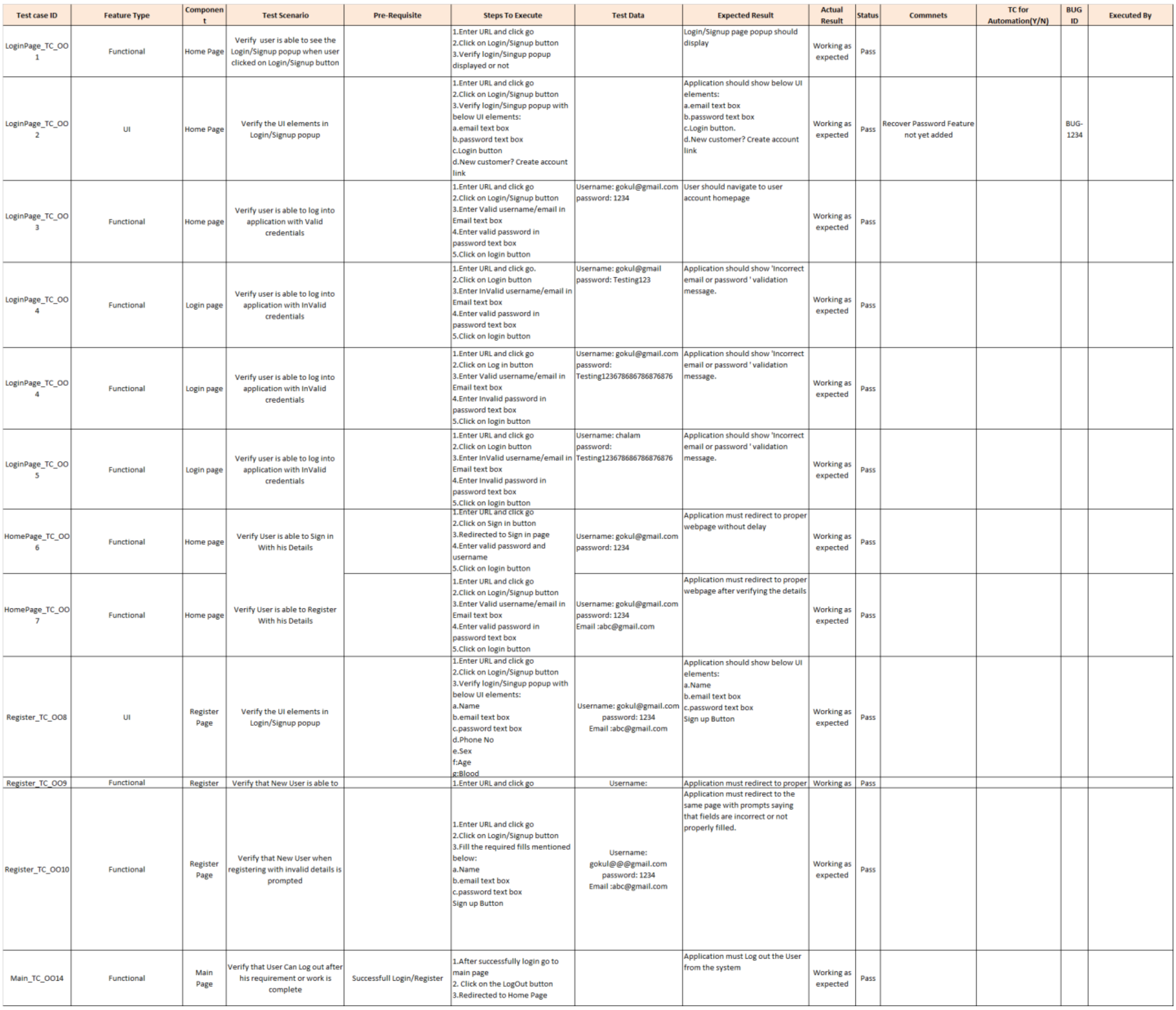
Kubernetes

- **Kubernetes** is also known as **'k8s'.**
- **Kubernetes** is an extensible, portable, and open-source platform designed by **Google** in **2014**.
- It is mainly used to automate the deployment, scaling, and operations of the container-based applications across the cluster of nodes.
- Kubernetes helps to manage containerised applications in various types of physical, virtual, and cloud environments.
- Google Kubernetes is a highly flexible container tool to consistentlydeliver complex applications running on clusters of hundreds to thousands of individual servers
- Kubernetes is the Linux kernel which is used for distributed systems.
- It helps you to be abstract the underlying hardware of the nodes(servers) and offers a consistent interface for applications that consume the shared pool of resources.

8.TESTING

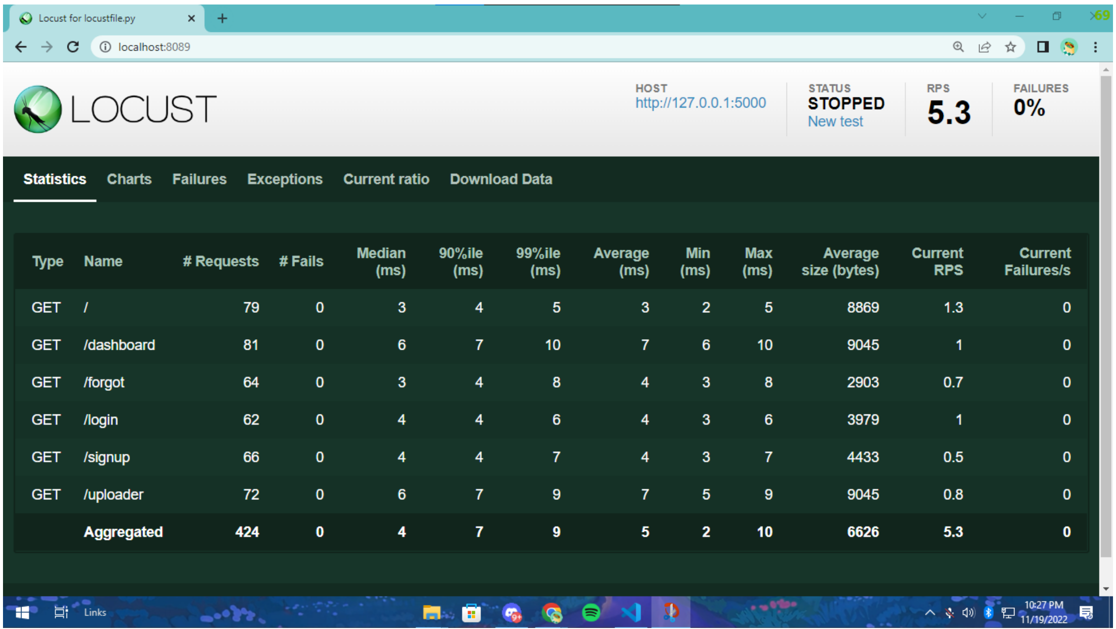
8.1 Test case

- It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectation and does not fail in an unacceptable manner.
- There are various types of test. Each test type addresses a specific testing requirement



Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on Login/Signup button		1.Enter URL and click go 2.Click on Login/Signup button 3.Verify login/Singup popup displayed or not		Login/Signup page popup should display	Working as expected	Pass				
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on Login/Signup button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link		Application should show below UI elements: a.email text box b.password text box c.Login button. d.New customer? Create account link	Working as expected	Pass	Recover Password Feature not yet added		BUG-1234	
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter URL and click go 2.Click on Login/Signup button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: gokul@gmail.com password: 1234	User should navigate to user account homepage	Working as expected	Pass				
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL and click go. 2.Click on Login button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: gokul@gmail password: Testing123	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL and click go 2.Click on Log in button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: gokul@gmail.com password: Testing123678686786876876	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				
LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL and click go 2.Click on Login button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: chalam password: Testing123678686786876876	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				
HomePage_TC_OO6	Functional	Home page	Verify User is able to Sign in With his Details		1.Enter URL and click go 2.Click on Sign in button 3.Redirected to Sign in page 4.Enter valid password and username 5.Click on login button	Username: gokul@gmail.com password: 1234	Application must redirect to proper webpage without delay	Working as expected	Pass				
HomePage_TC_OO7	Functional	Home page	Verify User is able to Register With his Details		1.Enter URL and click go 2.Click on Login/Signup button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: gokul@gmail.com password: 1234 Email :abc@gmail.com	Application must redirect to proper webpage after verifying the details	Working as expected	Pass				
Register_TC_OO8	UI	Register Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on Login/Signup button 3.Verify login/Singup popup with below UI elements: a.Name b.email text box c.password text box d.Phone No e.Sex f:Age g:Blood	Username: gokul@gmail.com password: 1234 Email :abc@gmail.com	Application should show below UI elements: a.Name b.email text box c.password text box Sign up Button	Working as expected	Pass				
Register_TC_OO9	Functional	Register	Verify that New User is able to		1.Enter URL and click go	Username:	Application must redirect to proper	Working as	Pass				
Register_TC_OO10	Functional	Register Page	Verify that New User when registering with invalid details is prompted		1.Enter URL and click go 2.Click on Login/Signup button 3.Fill the required fills mentioned below: a.Name b.email text box c.password text box Sign up Button	Username: gokul@@@gmail.com password: 1234 Email :abc@gmail.com	Application must redirect to the same page with prompts saying that fields are incorrect or not properly filled.	Working as expected	Pass				
Main_TC_OO14	Functional	Main Page	Verify that User Can Log out after his requirement or work is complete	Successfull Login/Register	1.After successfully login go to main page 2. Click on the LogOut button 3.Redirected to Home Page		Application must Log out the User from the system	Working as expected	Pass				

8.2 User Acceptance Testing



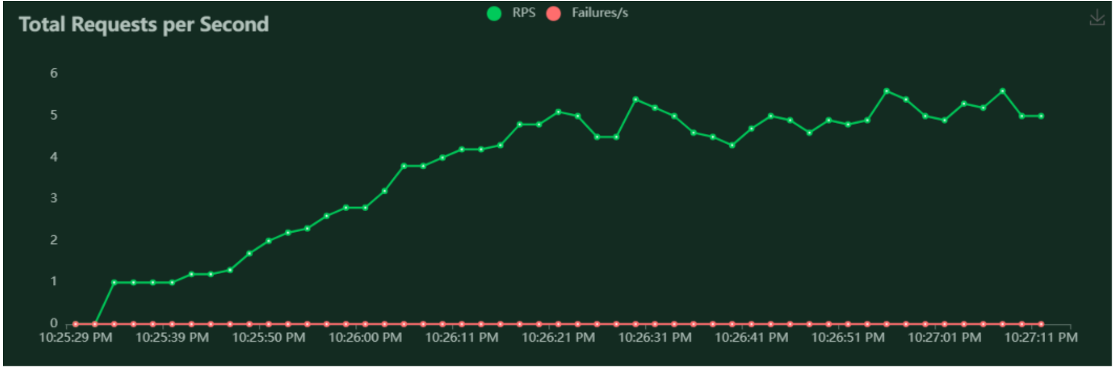
HOST http://127.0.0.1:5000

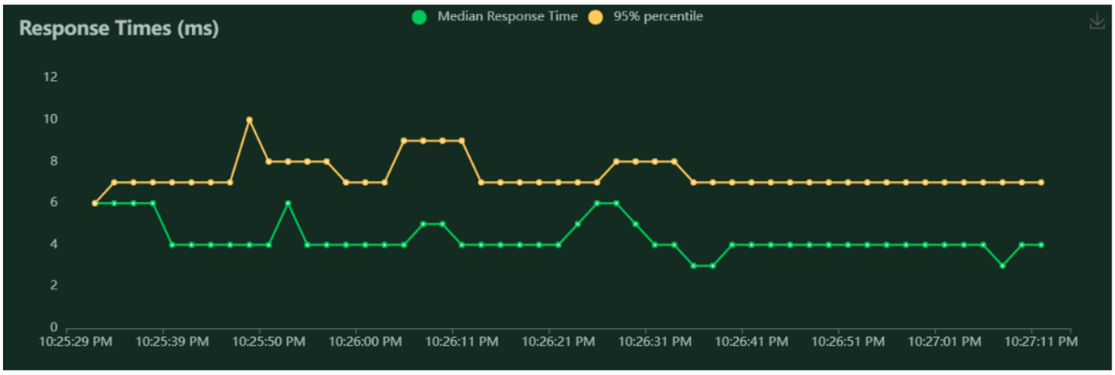
STATUS STOPPED New test

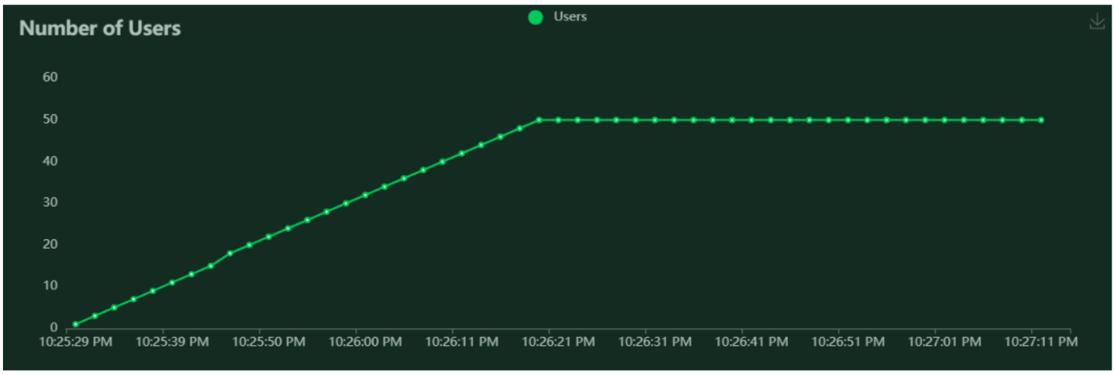
RPS 5.3

FAILURES 0%

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/	79	0	3	4	5	3	2	5	8869	1.3	0
GET	/dashboard	81	0	6	7	10	7	6	10	9045	1	0
GET	/forgot	64	0	3	4	8	4	3	8	2903	0.7	0
GET	/login	62	0	4	4	6	4	3	6	3979	1	0
GET	/signup	66	0	4	4	7	4	3	7	4433	0.5	0
GET	/uploader	72	0	6	7	9	7	5	9	9045	0.8	0
	Aggregated	**424**	**0**	**4**	**7**	**9**	**5**	**2**	**10**	**6626**	**5.3**	**0**







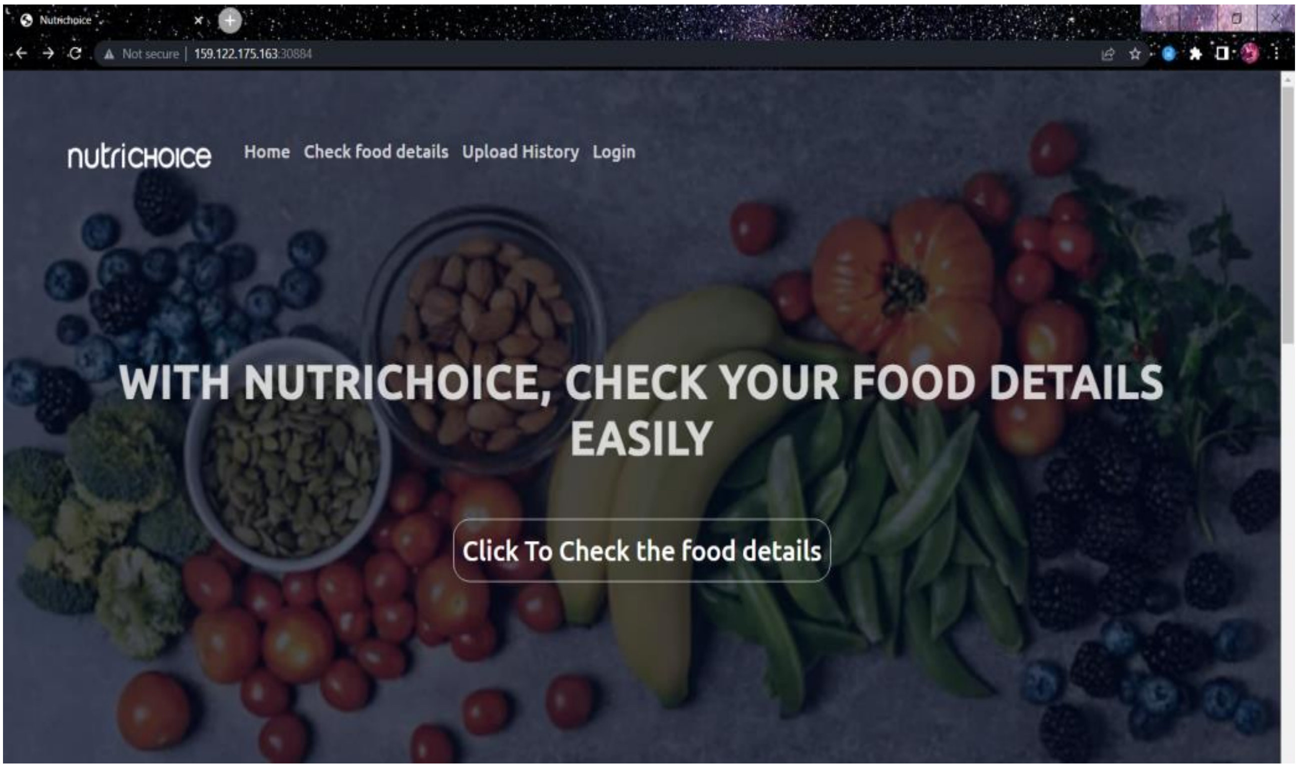
9.RESULTS

9.1 Performance Metrics

>Project metrics are used to track the progress and performance of a project.
>Monitoring parts of a project like productivity, scheduling, and scope make it easier for team leaders to see what's on track.
>As a project evolves, managers need access to changing deadlines or budgets to meet their client's expectations

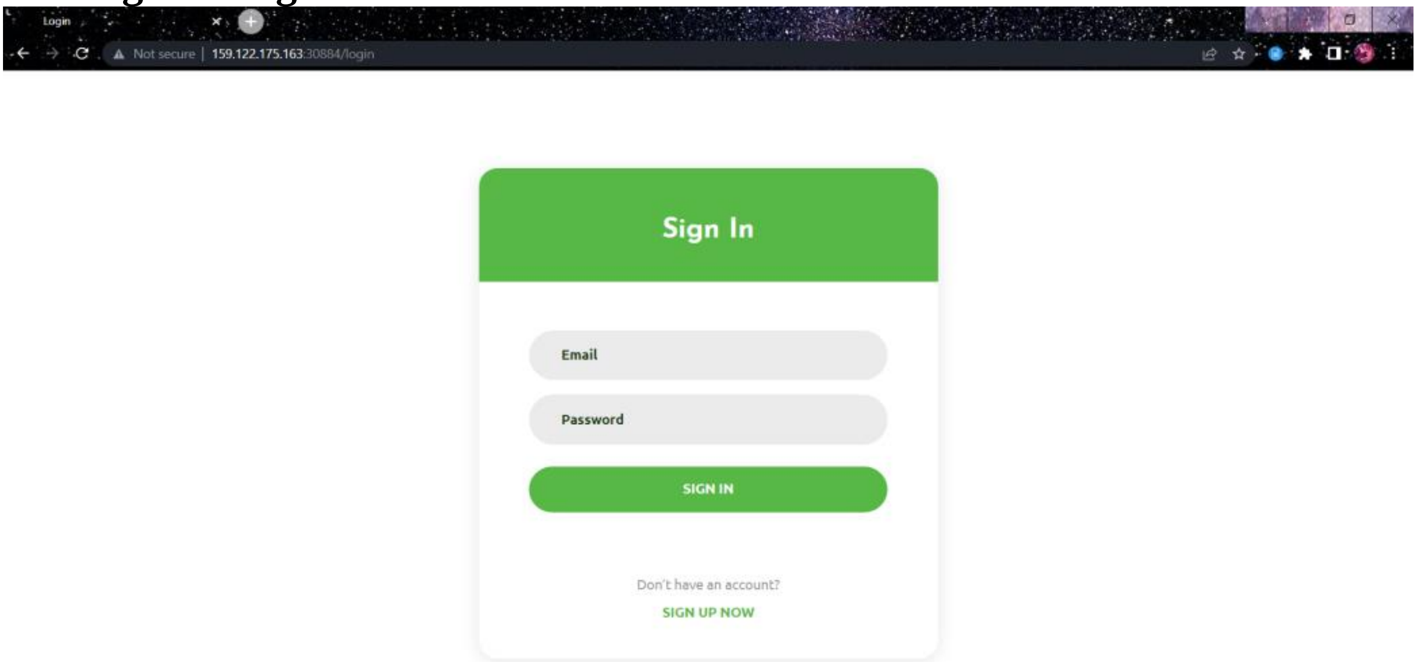
OUTPUT SCREENS

HOME PAGE


Nutrichoice
Not secure | 159.122.175.163:30884
nutriCHOICE
Home
Check food details
Upload History
Login
Click To Check the food details

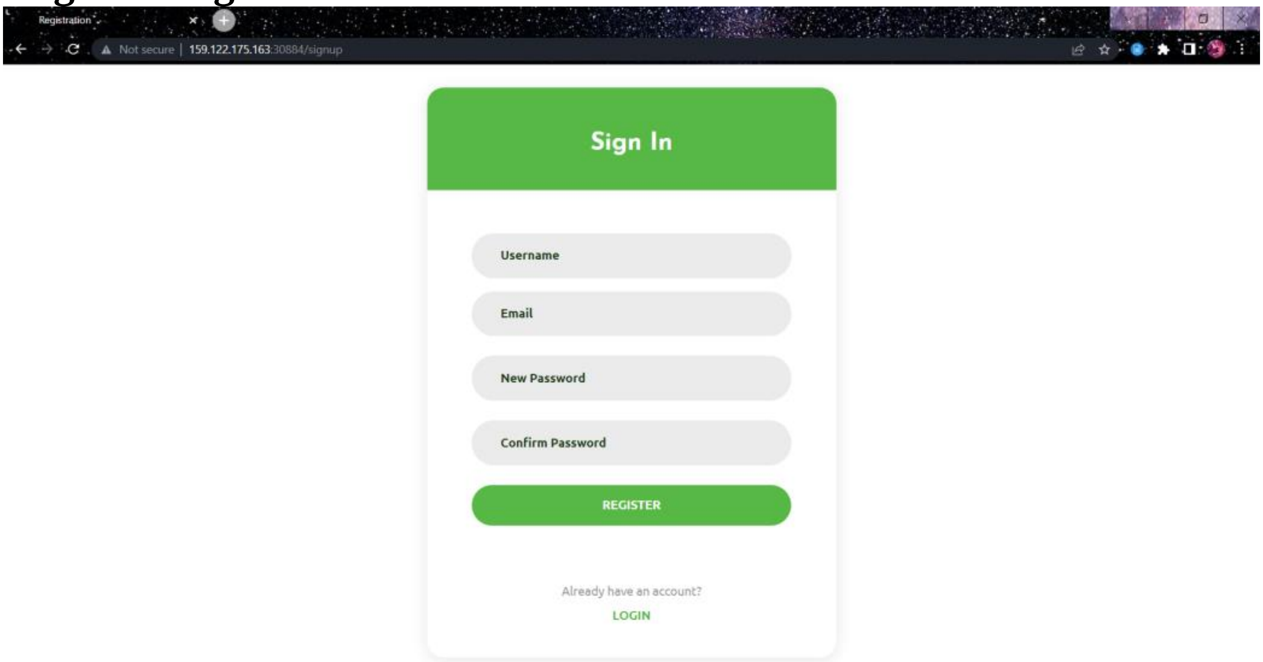
WITH NUTRICHOICE, CHECK YOUR FOOD DETAILS EASILY

Sign In Page



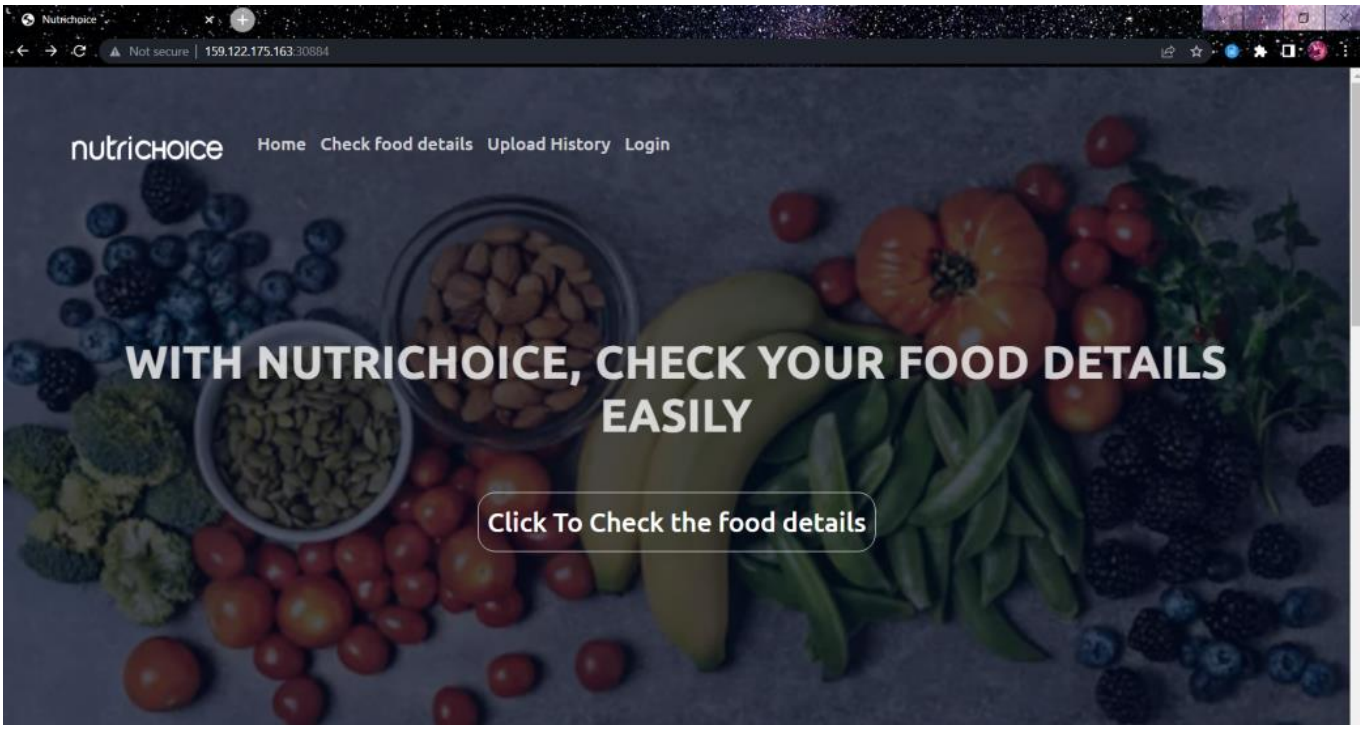
Don't have an account?

Register Page



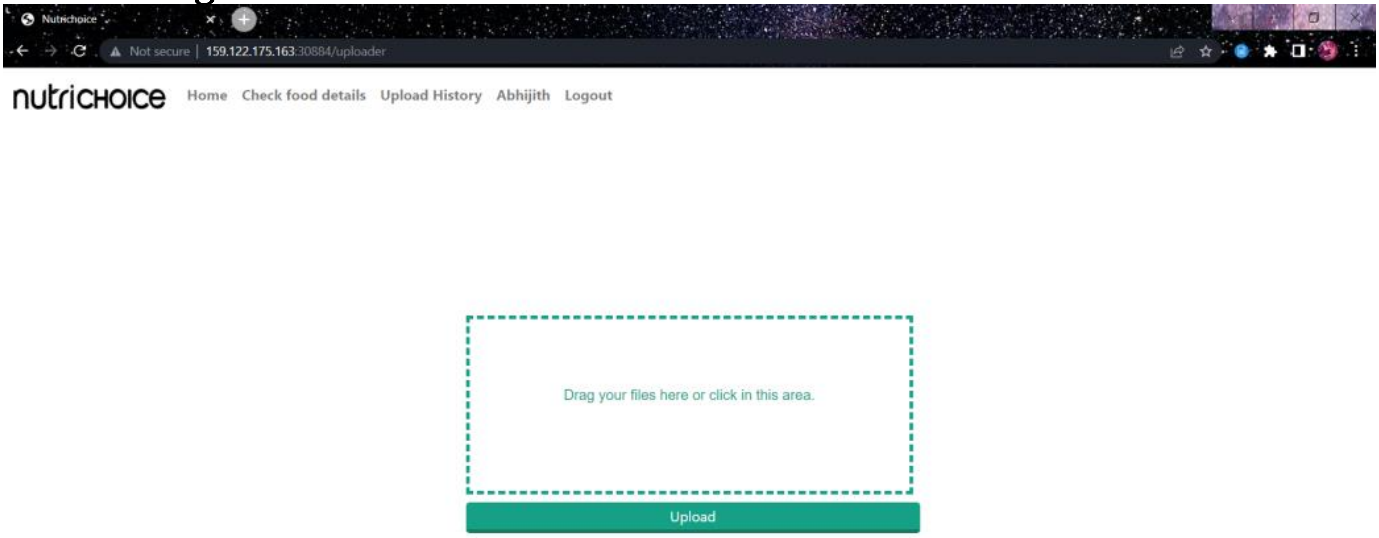
Already have an account?

Dash board


Nutrichoice
Not secure | 159.122.175.163:30884
nutriCHOICE
Home
Check food details
Upload History
Login
Click To Check the food details

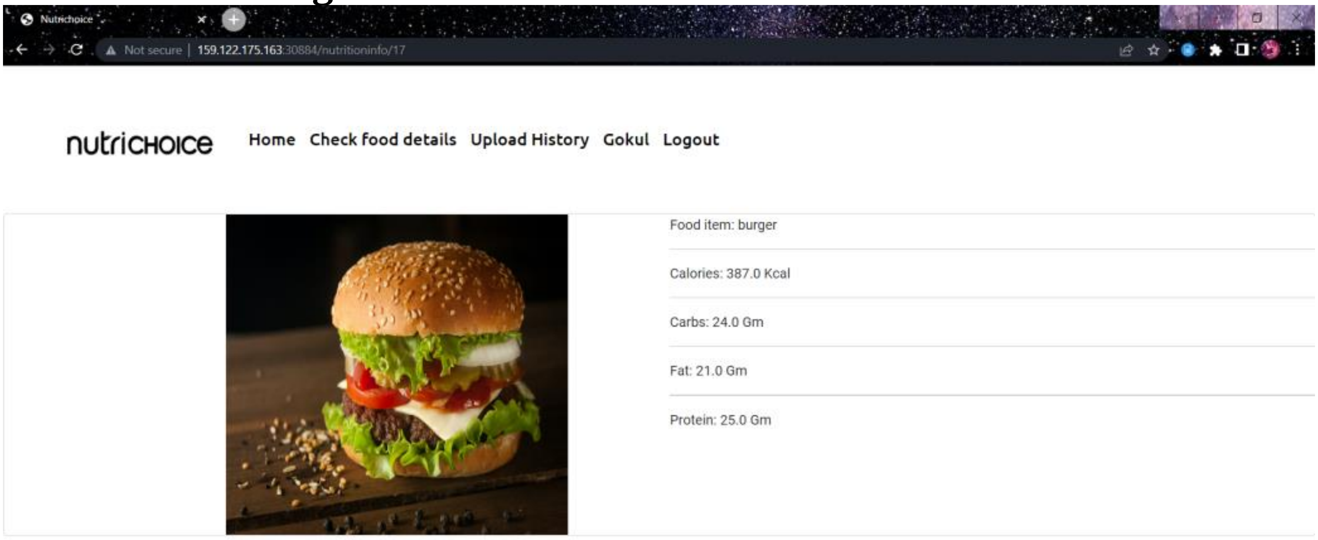
WITH NUTRICHOICE, CHECK YOUR FOOD DETAILS EASILY

Service Page



Drag your files here or click in this area.

Food Details Page



Food item: burger

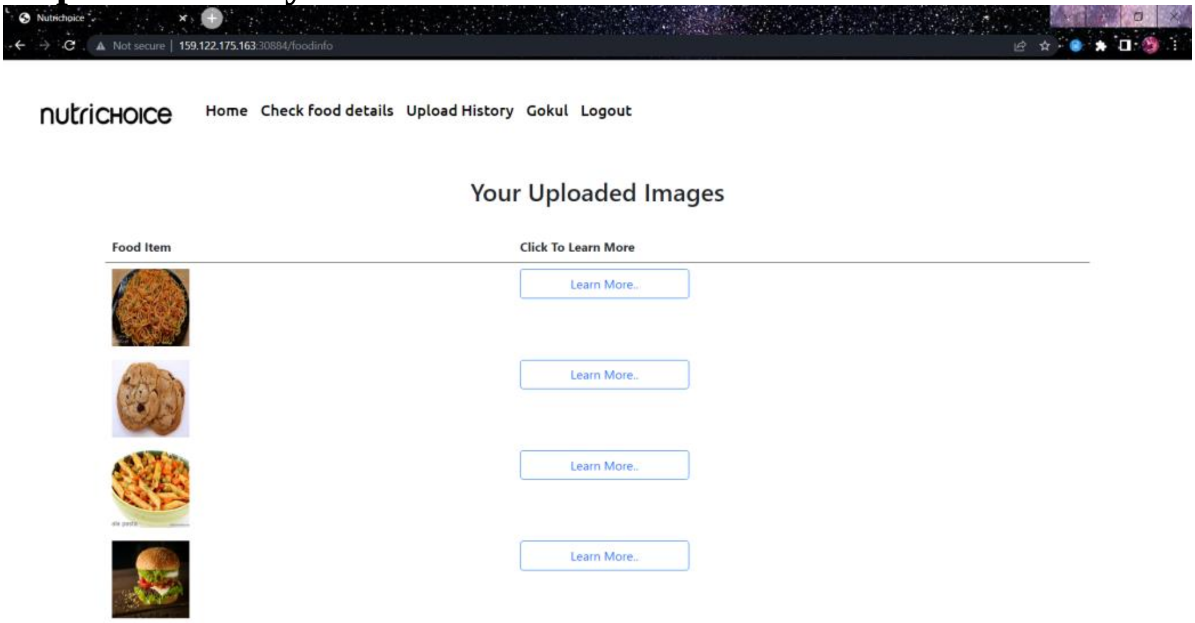
Calories: 387.0 Kcal

Carbs: 24.0 Gm

Fat: 21.0 Gm

Protein: 25.0 Gm

Upload History



Your Uploaded Images

Food Item	Click To Learn More
	Learn More..
	Learn More..
	Learn More..
	Learn More..

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Low Energy Consumption.
- Works Under Low Data Connection.
- User Friendly Web Application.
- Data Privacy.
- Easy to Understand.

DISADVANTAGES:

- It Cannot be Used Without Internet Connection.
- Usage of 3rd party API may cause the time delay.

11.CONCLUSION

- The efficient way of tracking a user’s diet and knowing the nutrient intake of any food is implemented using the nutrition assistant website that is hosted on IBM Cloud platform.

- To ensure the smooth functioning of the web site operation. I have hosted the website in IBM Db2 & Kubernetes Cluster to make sure the operations are running successfully Cloud lambda function is used and to deploy the application IBM Db2 service is used.

12. FUTURE SCOPE

- Upgrading the UI that is more user-friendly which will help many users to access the website and also ensures to keep track of their nutrition value in their diet.

Using elastic load balancer, it helps to handle multiple requests at the same time which will maintain the uptime of the website with negligible

13.APPENDIXES

13.1 Sample Source Code

```
from flask import Flask, render_template, redirect, url_for, request, flash, escape, session
from flask_wtf import FlaskForm
from wtforms import StringField, PasswordField, EmailField
from wtforms.validators import InputRequired, Length, Email, EqualTo
import openapi_client
from com.spoonacular import misc_api
import ibm_boto3
from ibm_botocore.client import Config, ClientError
import os
import ibm_db
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail

import os
from dotenv import load_dotenv

load_dotenv()

COS_ENDPOINT="https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID= os.getenv("COSAPI")
COS_INSTANCE_CRN= os.getenv("COSINSTANCE")

# Create resource https://s3.ap.cloud-object-storage.appdomain.cloud
cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_INSTANCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)


DB_HOSTNAME = os.getenv("DB_HOSTNAME")
DB_PORT = os.getenv("DB_PORT")
DB_USERNAME = os.getenv("DB_USERNAME")
DB_PASS = os.getenv("DB_PASS")
conn =
ibm_db.connect(f"DATABASE=bludb;HOSTNAME={DB_HOSTNAME};PORT={DB_PORT};SECURITY=SSL;SSLServer
Certificate=DigiCertGlobalRootCA.crt;UID={DB_USERNAME};PWD={DB_PASS}",'','')

app = Flask(__name__)
app.config['SECRET_KEY'] = os.getenv("SECERT_KEY")

class LoginForm(FlaskForm):
    email = EmailField("email", validators=[InputRequired("Email is required"), Email()])
    password = PasswordField("password", validators=[InputRequired("Password is required")])

class RegisterForm(FlaskForm):
    username = StringField("username", validators=[InputRequired("Username is required")])
    email = EmailField("email", validators=[InputRequired("Email is required"), Email()])
    pass1 = PasswordField("pass1", validators=[InputRequired("Password is required"),
EqualTo('pass2', message="Passwords must match"), Length(min=4, max=30, message="Length must
be between 4 and 30")])
    pass2 = PasswordField("pass2")
```

```
class ForgetPassword(FlaskForm):
    email = EmailField("email", validators=[InputRequired("Email is required"), Email()])

def SendEmail(user_email, user_name):
    FROM_EMAIL =  "aravindh.r.2019.cse@rajalakshmi.edu.in"
    TEMPLATE_ID = "d-6b7136a3a23441738d003ad8783631ca"
    key = os.getenv("SENDGRID_KEY")

    message = Mail(from_email = FROM_EMAIL,
    to_emails= user_email)

    message.dynamic_template_data = {
        'name': user_name
    }

    message.template_id = TEMPLATE_ID

    try:
        sg = SendGridAPIClient(key)
        response = sg.send(message)
        code, body, headers = response.status_code, response.body, response.headers
        print(f"Response code: {code}")
        print(f"Response headers: {headers}")
        print(f"Response body: {body}")
        print("Dynamic Messages Sent!")
    except Exception as e:
        print(e)

@app.route("/")
def home():
    if 'username' in session:
        username = session['username']
    else:
        username = request.cookies.get('username')

    return render_template("home.html", username=username)

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

    if request.method=='POST' and form.validate_on_submit():
        email = request.form['email']
        password = request.form['password']
        print(email)
        sql = f"SELECT * FROM USER_DB WHERE EMAILID='{escape(email)}'"
        stmt = ibm_db.exec_immediate(conn, sql)
        dic = ibm_db.fetch_both(stmt)
        print(dic)
        if not dic or password != dic['PASSWORD']:

            flash("Incorrect email or password", "error")
            return redirect(url_for('login'))
        session['username'] =  dic['USERNAME']
        return redirect(url_for('home'))
    else:
        return render_template("login.html", form=form)
```

```
@app.route('/signup', methods=['GET', 'POST'])
def signup():
    form = RegisterForm()

    if request.method=='POST' and form.validate_on_submit():
        username = request.form['username']
        email = request.form['email']
        pass1 = request.form['pass1']
        sql = f"SELECT * FROM USER_DB WHERE EMAILID='{escape(email)}'"
        stmt = ibm_db.exec_immediate(conn, sql)
        dic = ibm_db.fetch_both(stmt)
        if dic:
            flash("User with the email already exist", "error")
            return redirect(url_for('signup'))
        sql = "INSERT INTO USER_DB VALUES (?, ?, ?)"
        prep_stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(prep_stmt, 1, username)
        ibm_db.bind_param(prep_stmt, 2, pass1)
        ibm_db.bind_param(prep_stmt, 3, email)
        ibm_db.execute(prep_stmt)
        SendEmail(email, username)
        flash("Registration Successful", "success")
        response = redirect(url_for('login'))
        return response
    else:
        return render_template("register.html", form=form)

@app.route('/forgot', methods=['GET', 'POST'])
def forgot_password():
    form = ForgetPassword()
    if request.method=='POST' and form.validate_on_submit():
        email = request.form['email']
        sql = f"SELECT * FROM USER_DB ERE EMAILID='{escape(email)}'"
        stmt = ibm_db.exec_immediate(conn, sql)
        dic = ibm_db.fetch_both(stmt)
        if dic:
            flash("Email has been sent if user exist", "success")
            return redirect(url_for('forgot_password'))

        return render_template("forgot_password.html", form=form)

    return render_template("forgot_password.html", form=form)

@app.route('/pictures')
def index():
    files = get_bucket_contents('cally2920')
    return render_template('index.html', files = files)


@app.route('/uploader', methods = ['GET', 'POST'])
def upload():
   if request.method == 'POST':
       bucket=os.getenv("BUCKETNAME")
       f = request.files['file']
       filename = f.filename
       file_path = os.path.join('static/uploads', filename)
       imagelink="https://cally2920.s3.jp-tok.cloud-object-
storage.appdomain.cloud/"+filename
```

```
       print(imagelink)
       f.save(file_path)
       uname = session['username']
       multi_part_upload(bucket,filename,file_path)
       sql = f"SELECT * FROM imagedetails WHERE IMAGE_LINK='{escape(imagelink)}'"
       stmt = ibm_db.exec_immediate(conn, sql)
       dic = ibm_db.fetch_both(stmt)
       if dic:
            flash("Image already exist", "error")
            return redirect(url_for('upload'))
       sql = f"INSERT INTO imagedetails(IMAGE_LINK,USERNAME) VALUES(?,?)"
       prep_stmt = ibm_db.prepare(conn, sql)
       ibm_db.bind_param(prep_stmt,1,imagelink)
       ibm_db.bind_param(prep_stmt,2,uname)
       ibm_db.execute(prep_stmt)

       sql = f"SELECT ID FROM imagedetails WHERE image_link='{escape(imagelink)}'"
       stmt = ibm_db.exec_immediate(conn, sql)
       image_id = ibm_db.fetch_both(stmt)
       nutitionapi(imagelink,image_id)
       return redirect("/foodinfo", code=307)

   if request.method == 'GET':
        username = session['username']
        return render_template('upload.html', username=username)

def get_item(bucket_name, item_name):
    print("Retrieving item from bucket: {0}, key: {1}".format(bucket_name, item_name))
    try:
        file = cos.Object(bucket_name, item_name).get()

        print("File Contents: {0}".format(file["Body"].read()))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve file contents: {0}".format(e))


def get_bucket_contents(bucket_name):
    print("Retrieving bucket contents from: {0}".format(bucket_name))
    try:
        files = cos.Bucket(bucket_name).objects.all()
        files_names = []
        for file in files:
            files_names.append(file.key)
            print("Item: {0} ({1} bytes).".format(file.key, file.size))
        return files_names
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve bucket contents: {0}".format(e))

def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name,
bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5
        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15

        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )

        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )


        print("Transfer for {0} Complete!\n".format(item_name))

    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))

def nutitionapi(imagelink,image_id):
    configuration = openapi_client.Configuration(
        host = "https://api.spoonacular.com"
    )
    configuration.api_key['apiKeyScheme'] = os.getenv("NUTRITIONAPI")

    with openapi_client.ApiClient(configuration) as api_client:
        api_instance = misc_api.MiscApi(api_client)
        image_url =imagelink
    try:
        api_response = api_instance.image_analysis_by_url(image_url)
        print(api_response)
        y =api_response
        cal= y["nutrition"]["calories"]["value"]
        print(cal)
        Carb= y["nutrition"]["carbs"]["value"]
        fat= y["nutrition"]["fat"]["value"]
        protein= y["nutrition"]["protein"]["value"]
        name=y["category"]["name"]
        image=image_id["ID"]
        sql = f"INSERT INTO nutritiondetails1(calories,carbs,fat,protein,ref_id,name)
VALUES('{escape(cal)}','{escape(Carb)}','{escape(fat)}','{escape(protein)}','{escape(image)}
','{escape(name)}')"

        prep_stmt = ibm_db.prepare(conn, sql)

        ibm_db.execute(prep_stmt)

    except openapi_client.ApiException as e:
        print("Exception when calling MiscApi->image_analysis_by_url: %s\n" % e)

@app.route('/logout', methods=['GET', 'POST'])
```

```
def logout():
    session.pop('username')
    return redirect(url_for('home'))

@app.route('/foodinfo', methods = ['GET', 'POST'])
def test():
    username = session['username']
    sql = f"SELECT * FROM imagedetails WHERE USERNAME='{escape(username)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    pic = ibm_db.fetch_both(stmt)
    pics=[]
    while pic != False:
     x=[pic["IMAGE_LINK"],pic["ID"]]
     pics.append(x)
     print(pic)
     pic = ibm_db.fetch_both(stmt)
     print(pics)
    return render_template('index.html', files = pics, username=username)

@app.route('/nutritioninfo/<id>', methods = ['GET', 'POST'])
def test1(id):
    print(id)
    sql = f"SELECT * FROM nutritiondetails1,imagedetails where 
nutritiondetails1.ref_id=imagedetails.id and ref_id='{escape(id)}'"
    stmt = ibm_db.exec_immediate(conn, sql)
    pic = ibm_db.fetch_both(stmt)
    username = session['username']
    print(pic)
    return render_template('foodinfo.html', files = pic, username=username)

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

13.2 GITHUB:

https://github.com/IBM-EPBL/IBM-Project-14141-1659542942

DEMO VIDEO:

https://drive.google.com/file/d/1BSQdqu6WbMZK1Osq66Samt5LAQd_E3oc/view?usp=share_link

LIVE APP:

http://159.122.175.163:30884/