NUTRITION ASSISTANT APPLICATION

A NAALAIYA THIRAN PROJECT REPORT

Submitted by

DHIVYA J	(610519104019)
AMSAVENI S M	(610519104003)
GAYATHRI S	(610519104024)
JEEVITHA P	(610519104042)

TEAM NO: PNT2022TMID29771

TABLE OF CONTENT

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.

1.2 PURPOSE

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

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

With the rapid development of smart computing and Internet of Things (IoT), now we have a huge amount of data from social networks and mobile networks everyday. People keep uploading, sharing and recording what they do everyday in case of missing the chance of using them to improve our daily life. Food images, recipes and food diaries become the most popular information to be shared, we can learn the implication to build an automatic nutrition analysi system by taking the advantage of such large-scale datasets. With the help of food recognition and analysis systems, users are able to record their daily meals and assess dietary habits, as well as promote their health

2.2 REFERENCES

REFERENCE PAPER:

A survey on nutrition monitoring and dietary management system

June 2019 **Authors:** Kamaks9hi Priyaa Prakash Dr L Arockiam

DESCRIPTION:

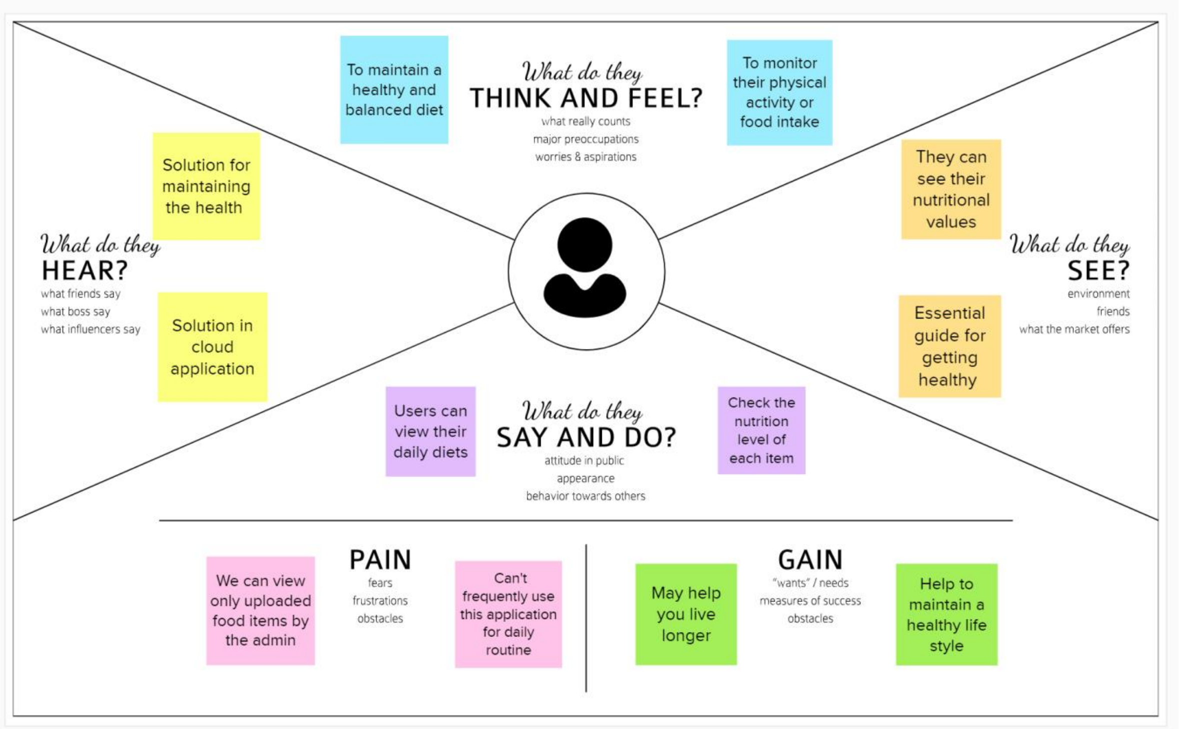
A well balanced diet with an estimated nutrient intake is vital for infants and children which reduces the risks of deadly diseases namely cancer, diabetes, obesity and cardiovascular diseases. Unlike adults, infants require some assistance in their food intake. The survey provides valuable insights about the various advancements of IoT in the healthcare industry and the need for nutrition and dietary monitoring. A varied number of nutrition monitoring systems for the estimation and prediction of calories have been developed using various machine learning techniques and also with advanced deep learning based techniques. A comparative view of the previous works of researchers in the recent times has been provided.

2.3 PROBLEM STATEMENT DEFINITION

This is based on the propose of a deep learning based system for food item detection and analyze the nutrition components of each meal image. Our model consists of three main steps. We first extract the regions of interests (ROIs) by applying the Region Proposal Network derived from the Faster R-CNN model. The ROIs would help to separate the food items from the background, and improve the detection model efficiency. The second step is to apply a well designed Convolutional Neural Network (CNN) on selected RoIs and classify them into different food item categories. Meanwhile, a regression module is also used to locate the food coordinates in the image. The final step is to use modern technology-based dietary assessment tools for food nutrition analysis and generate a health report for users based on their meal images.

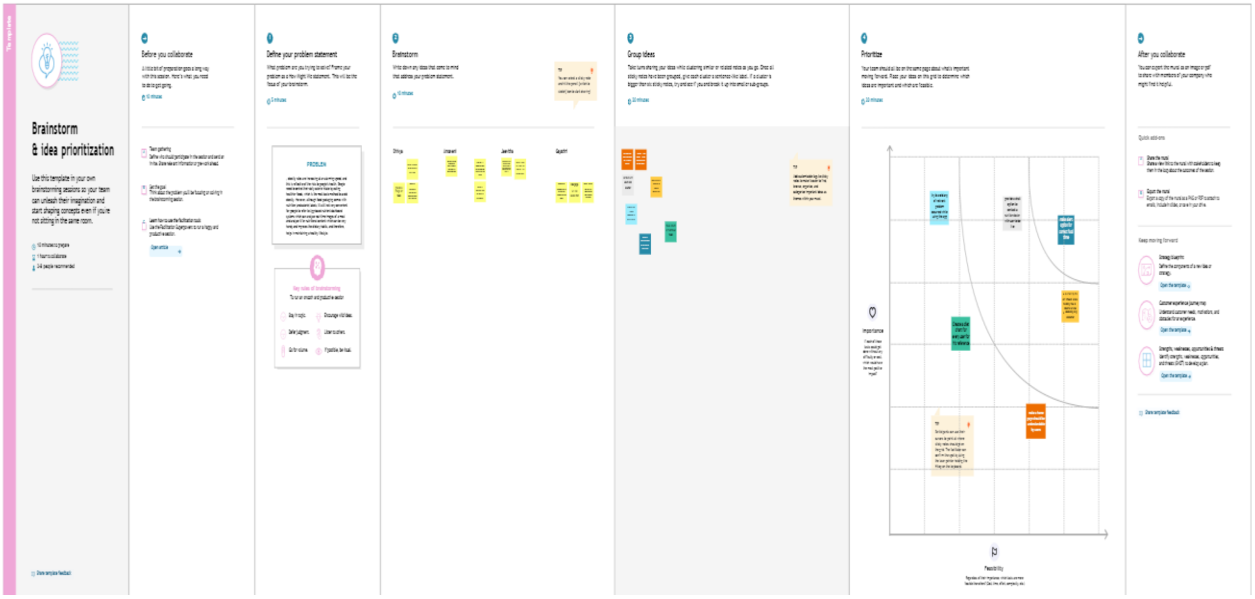
3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
To maintain a healthy and balanced diet
To monitor their physical activity or food intake
What do they
HEAR?
what friends say
what boss say
what influencers say
Solution for maintaining the health
Solution in cloud application
What do they
SEE?
environment
friends
what the market offers
They can see their nutritional values
Essential guide for getting healthy
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Users can view their daily diets
Check the nutrition level of each item
PAIN
fears
frustrations
obstacles
We can view only uploaded food items by the admin
Can't frequently use this application for daily routine
GAIN
"wants" / needs
measures of success
obstacles
May help you live longer
Help to maintain a healthy life style

3.2 Ideation & Brainstorming



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

PROBLEM

Key rules of brainstorming

Brainstorm

Group ideas

Prioritize

Importance

Feasibility

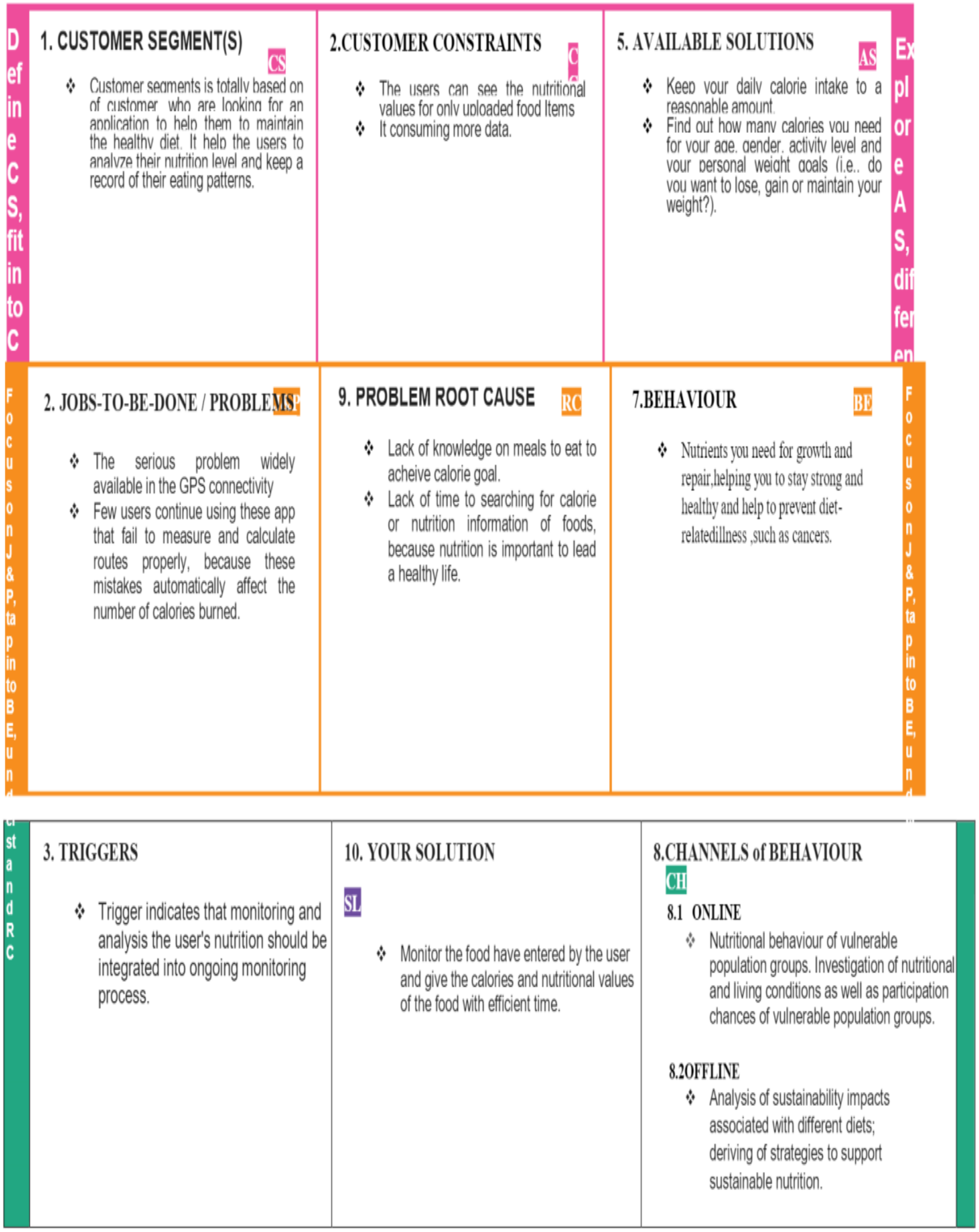
After you collaborate

3.3 Proposed Solution

S. No	Parameter	Description
1.	Problem Statement (Problem to be solved)	Health maintaining. Nutrition enhancement.
2.	Idea / Solution description	In this platform, there are numerous categories are available and we can get the nutritional value by uploading the images. The image is passed to the server application, which uses Clarifai's AI-Driven Food Detection Model Service to

		analyze the images. The nutritional value and the food details are stored in the database. Using Nutrition API to provide nutritional information about the analyzed Image.
3.	Novelty / Uniqueness	In this application , not only we get the nutritional value,the user can also get the detailed information of the images.
4.	Social Impact / Customer Satisfaction	By using this application people can reduce their obesity rate, diseases level and maintain the health in good condition .
5.	Business Model (Revenue Model)	Social media is the best way to spread the word about our application. And with the influencers we can attract the normal people. Revenue can be generated by giving ads in this application and from affiliate commission.
6.	Scalability of the Solution	Through this the user can get the correct nutritional value and lead a healthly life.

3.4 Problem Solution fit



Define CS, fit into C

1. CUSTOMER SEGMENT(S) CS

- ❖ Customer segments is totally based on of customer who are looking for an application to help them to maintain the healthy diet. It help the users to analyze their nutrition level and keep a record of their eating patterns.

2.CUSTOMER CONSTRAINTS CC

- ❖ The users can see the nutritional values for only uploaded food Items
- ❖ It consuming more data.

5. AVAILABLE SOLUTIONS AS

- ❖ Keep your daily calorie intake to a reasonable amount.
- ❖ Find out how many calories you need for your age, gender, activity level and your personal weight goals (i.e., do you want to lose, gain or maintain your weight?).

Explore AS, differen

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

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

- ❖ The serious problem widely available in the GPS connectivity
- ❖ Few users continue using these app that fail to measure and calculate routes properly, because these mistakes automatically affect the number of calories burned.

9. PROBLEM ROOT CAUSE RC

- ❖ Lack of knowledge on meals to eat to acheive calorie goal.
- ❖ Lack of time to searching for calorie or nutrition information of foods, because nutrition is important to lead a healthy life.

7.BEHAVIOUR BE

- ❖ Nutrients you need for growth and repair,helping you to stay strong and healthy and help to prevent diet-relatedillness ,such as cancers.

Focus on J&P, tap into BE, und

3. TRIGGERS

- ❖ Trigger indicates that monitoring and analysis the user's nutrition should be integrated into ongoing monitoring process.

10. YOUR SOLUTION

SL

- ❖ Monitor the food have entered by the user and give the calories and nutritional values of the food with efficient time.

8.CHANNELS of BEHAVIOUR

CH

8.1 ONLINE

- ❖ Nutritional behaviour of vulnerable population groups. Investigation of nutritional and living conditions as well as participation chances of vulnerable population groups.

8.2OFFLINE

- ❖ Analysis of sustainability impacts associated with different diets; deriving of strategies to support sustainable nutrition.

4.REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Registration	This information includes age, body weight, height, foodallergies, weight loss goals, desired weight, preferred diet, and many more. Moreover, there must be a sign-up option via email, Facebook, Twitter, or manually.
FR-2	Dashboard and food logging	After registration, the first thing your user should encounter is a screen or dashboard where the user canfind all the information such as food intake, his/her progress in achieving fitness goals, and various nutritional and healthy eating tips.
FR-3	Push notifications	Push Notification is an important element of any mobileapp as it is well-known for customer retention.You can use Push Notification to remind your user of their daily workout sessions or meals. Additionally, you can use it to motivate them to push more. Push Notification is likea double-edged sword.
FR-4	Diet plan or user's goals	Once your user specifies the goal like desired weight goal, body type, food habits, and preferred food items,your app must suggest them with a proper diet accordingly.on demand diet and nutrition app diet planor users goals .
FR-5	Barcode scanner	Embedding a Barcode Scanner would be a great featurefor your app as it would assist your customers in shopping.It would provide important data such as calories and ingredients of the items with the help ofyour phone.
FR-6	Help	You must have help sections which must include all theFAQs along with a tutorial video which would act as a user manual. You can add a chat service too in case, a user still has some questions.

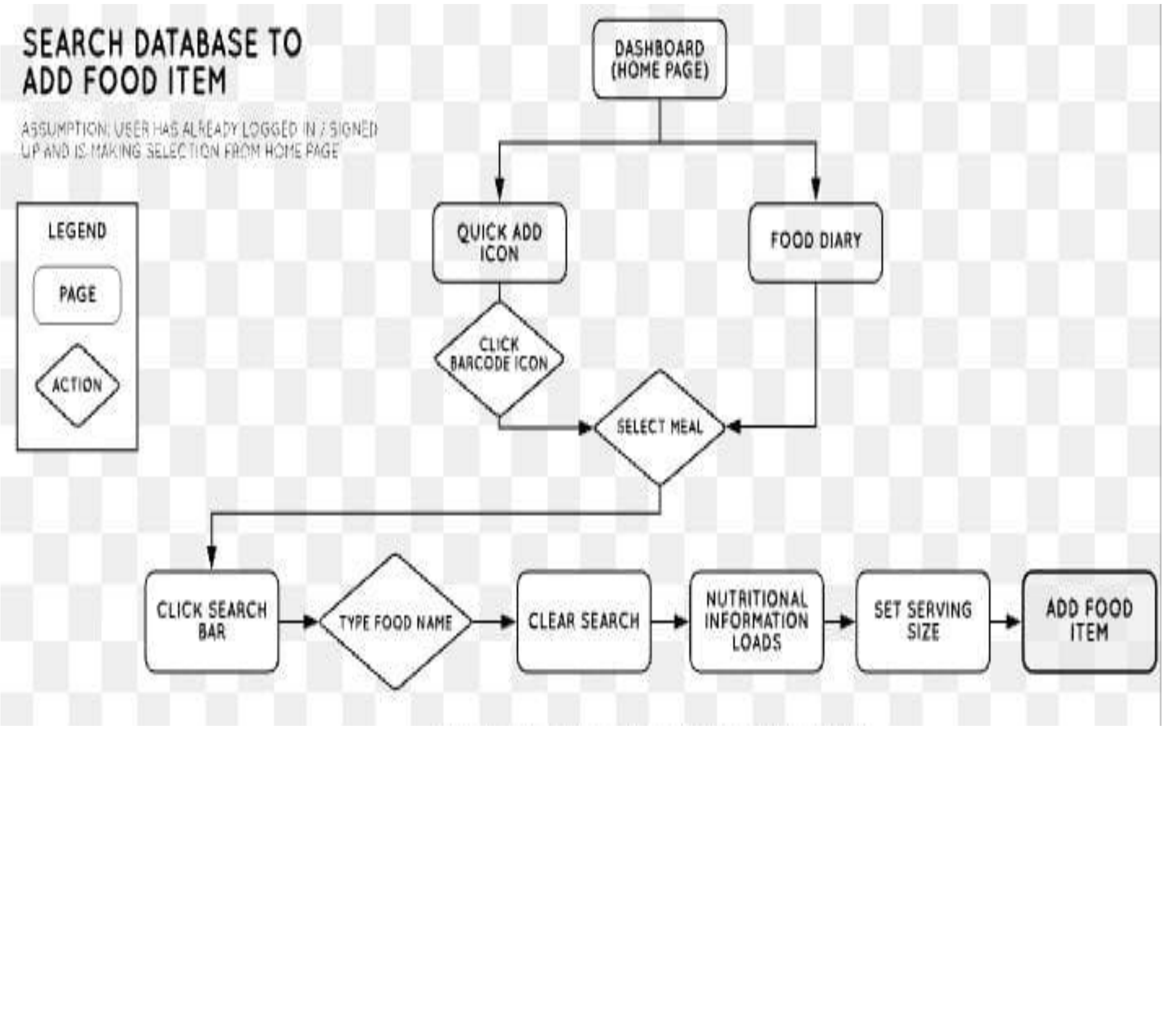
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Nutrition assistant app provides vital nutrients for survival, and helps the body function and stay healthy. Food is comprised of macronutrients including protein, carbohydrate and fat that not only offer calories to fuel the body and give it energy but play specific roles in maintaining health.
NFR-2	**Security**	The nutritional app has a security "exists when all people at all times have both physical and economic access to sufficient, safe and nutritious food that meets their dietary needs for an active and healthy life".
NFR-3	**Reliability**	Nutrition assistant application are effective in changing eating behavior and diet-related health risk factors. However, while they may curb growing overweight and obesity rates, widespread adoption is yet to be achieved.
NFR-4	**Performance**	The right nutrition combined with specific nutrient timing is crucial for every patient to enhance the recovery process.The app will provide step-by-step guidelines for how and which nutrients are necessary for specific patient should fuel their body for optimizing their performance.
NFR-5	**Availability**	The Nutrition assistant application had the greatest number of features in the dietary intake category. Additional dietary intake features were those most likely obtained through a subscription purchase. Behavior change content was absent from this app.
NFR-6	**Scalability**	The nutrition assistant application should possess enough data consistency to handle a growing number of users.Nutrition assistant application's scalability is directly linked to application's architecture

5. PROJECT DESIGN

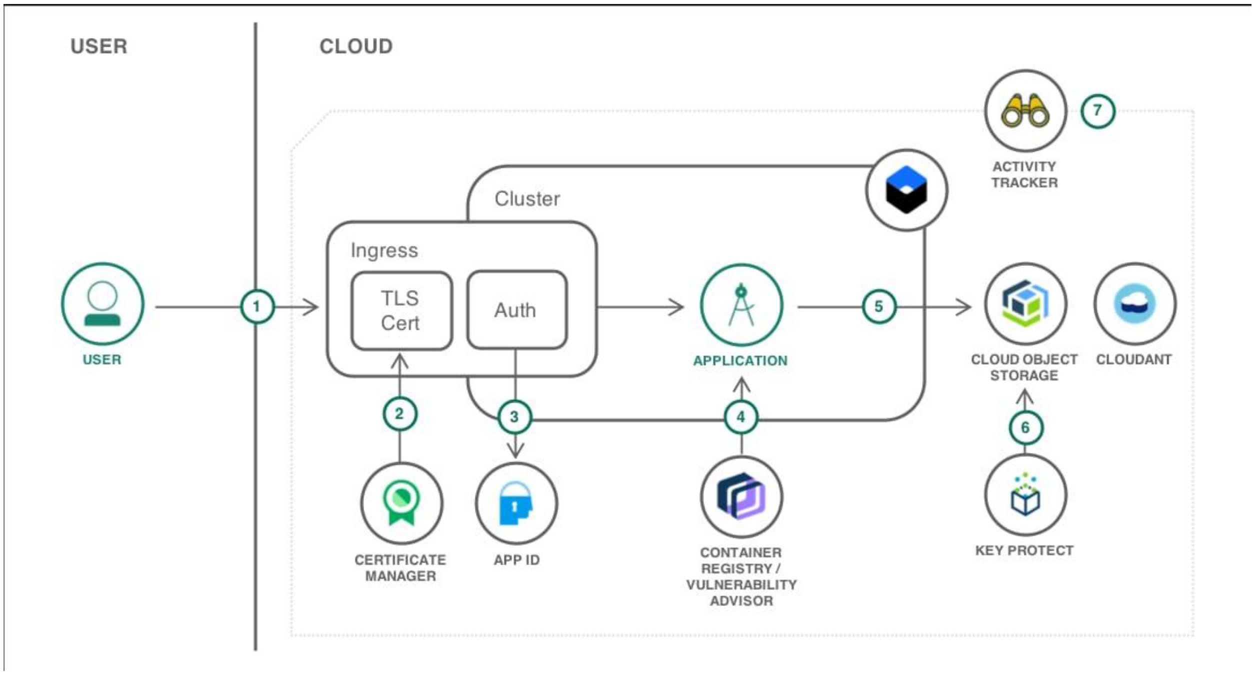
5.1 Data Flow Diagrams

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



5.2 Solution & Technical Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. A key challenge in human nutrition is the assessment of usual food intake. This is of particular interest given recent proposals of eHealth personalized interventions. The adoption of mobile phones has created an opportunity for assessing and improving nutrient intake as they can be used for digitalizing dietary assessments and providing feedback. In the last few years, hundreds of nutrition-related mobile apps have been launched and installed by millions of users.Smartphone applications are increasingly being used to support nutrition improvement. The source of the information on the website should be appropriately referenced and verifiable.A Nutrition Assistant makes sure that patients in a healthcare unit are fed according to their dietary needs.They can know the specialization of the food and the assistant can prefers which food items is necessary for the user based on their health conditions.



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	In the User Signup and Login form, the user has been allowed to simply enter the email id and password to signup their details. Once the details entered, the user account will get created.	I can access the patient record or datas has been seen in dashboard.	High	Sprint-1
		USN-2	The registered user can get log into the main form by just entering the registered email id and password.	Once email and password is entered,the entire data of user has been restored.	High	Sprint-1
		USN-3	The registered users can input the daily breakfast, lunch, dinner and snacks intaken during the whole day in this form.	I can register & access the nutritional values to maintain the healthy diet.	High	Sprint-2
		USN-4	These details will evaluated with the RDA and nutritive value calculation.		Medium	Sprint-1
	Login	USN-5	While entering the following criteria as email and password, the		High	Sprint-1

			user data has been stored in the database.			
	Dashboard		In dashboard,there is a search button,which is used to search the food details.			
Customer (Web user)			The foods searched by the user had been stored.			
Customer Care Executive			Have datas in graph modes.			
Administrator			Access the datas in cloud easily.			

5.4 Technology stack

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
2.	Application Logic-1	Logic for a process in the application	Python
3.	Cloud Database	Database Service on Cloud	IBM DB2

4.	File Storage	File storage requirements	IBM Object Storage
5.	Infrastructure (Server / Cloud)	Application Deployment on Cloud Cloud Server Configuration : Db2/python	Kubernetes

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask	Technology of Open source framework
2.	Security Implementations	Password Hashing	SHA-256 Crypt
3.	Scalable Architecture	Supports higher workloads	Python flask
4.	Availability	Available 24/7	IBM Cloud
5.	Performance	500 requests per day	Rapid API

6.PROJECT PLANNING & SCHEDULING

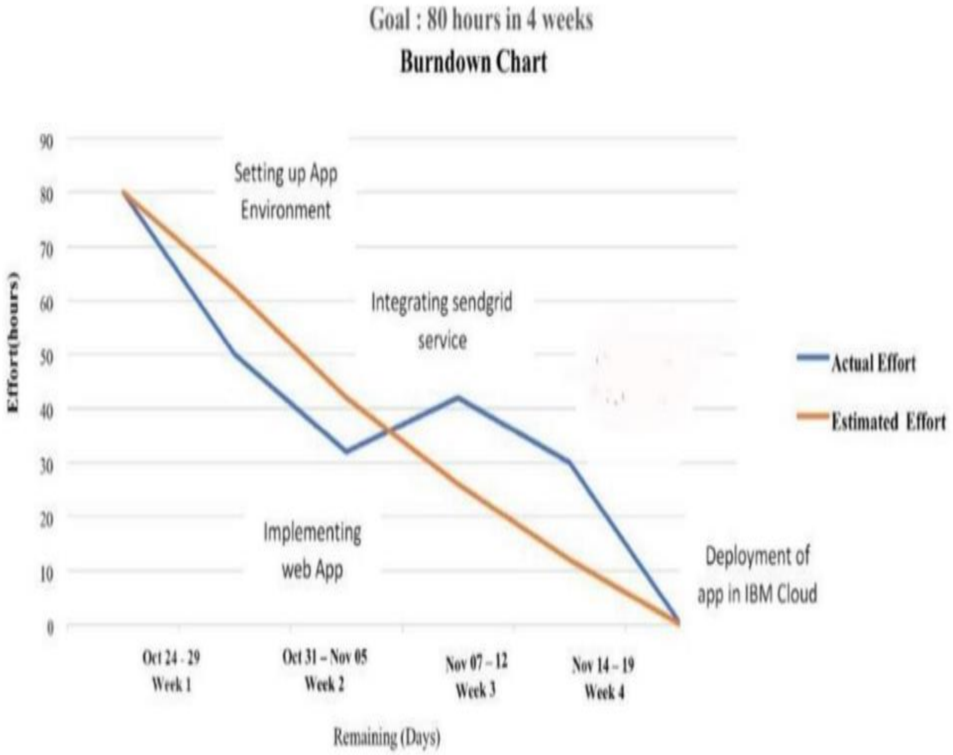
6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	UserStory Number	UserStory/Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user,I can register for the application by entering my username,email, password	2	High	Dhivya J Amsaveni S M Gayathri S Jeevitha P
Sprint-1		USN-2	As a user,I will receive the OTP once I have registered for the application.	1	High	Dhivya J Amsaveni S M Gayathri S Jeevitha P
Sprint-1		USN-3	As a user,I can enter the valid OTP,then only I can l can view the login page.	2	Medium	Dhivya J Amsaveni S M Gayathri S Jeevitha P
Sprint-2	Login	USN-4	As a user,I can login to the application by Entering my username & password.	1	High	Dhivya J Amsaveni S M Gayathri S Jeevitha P
Sprint-2	Database	USN-5	As a admin,I can store the user information in the database.	1	High	Dhivya J Amsaveni S M Gayathri S Jeevitha P
Sprint-3	Accessing the application	USN-7	As a user , I can upload the image of food and identify it’s nutritional value with the help of application.	2	High	Dhivya J Amsaveni S M Gayathri S Jeevitha P
Sprint-3	Tracking the calories	USN-8	As a user , I can easily track my calories.	2	Medium	Dhivya J Amsaveni S M Gayathri S Jeevitha P
Sprint-4	Provide nutritional information	USN-11	As a admin, I can provide nutritional information about the analyzed image.	2	High	Dhivya J Amsaveni S M Gayathri S Jeevitha P

6.2 Sprint Delivery Schedule

Sprint	**Total Story Point**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned EndDate)**	**Sprint Release Date (Actual)**
Sprint-1	20	6Days	24Oct2022	29Oct2022	20	29Oct2022
Sprint-2	20	6Days	31Oct2022	05Nov2022	20	05Nov2022
Sprint-3	20	6Days	07Nov2022	12Nov2022	20	12Nov2022
Sprint-4	20	6Days	14Nov2022	19Nov2022	20	19Nov2022

6.3 Reports from JIRA



7.CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

- IBM cloud
- HTML
- Python-flask
- Kubernetes
- Docker
- IBM DB2
- IBM container registery

7.2 Feature 2

- Login
- Dashboard
- Upload image
- Nutrition value

8.TESTING

8.1 Test Cases

- Login button click with wrong credentials entered.
- Signup with already registered mail ID.
- Signup with wrong form data entered.
- Entering home page with logged out session.
- Clicking home page buttons with logged out session

8.2 User Acceptance Testing

S.NO	TEST CASE	REQURIED OUTPUT	RESULT	STATUS
1	Login button click with wrong credentials entered.	Wrong credentials entered notification	Wrong credentials entered notification	ACCEPTED
2	Signup with already registered mail ID.	Email already registered notification	Email already registered notification	ACCEPTED
3	Signup with wrong form data entered.	Wrong credentials entered notification	Wrong credentials entered notification	ACCEPTED
4	Entering home page with logged out session.	Take user to login page	Take user to login page	ACCEPTED
5	Clicking home page buttons with logged out session	Take user to login page	Take user to login page	ACCEPTED

9.RESULTS

9.1Performance Metrics

- Hours worked : 50 hours
- Stick to Timelines : 100%
- Consistency of the product : 75%
- Efficiency of the product : 80%
- Quality of the product : 80%

10.ADVANTAGES & DISADVANTAGES

Advantages:

- Low cost.
- Simple UI.
- Faster response due to single page web page.
- Capability of adding many features with ease and less cost.
- User can use at any where at any time it leads to save the time.

Disadvantages:

- Lack of efficiency and the product needs to be improved.
- Consistency of the product is not 100%.
- Not a compact sized product. Size needs to be decreased.
- May be network issue in some times.

11.CONCLUSION

The main motive of this application to become user friendly. A food item can upload in that page it can recognise what it is and it can analysed by clarifai food detecting API. The diversity of real food photos is higher than the lab trained model. An ingredient based recognition is a promising way of tracking the free style and homemade food recognition problems in which training data is sparse and not representative. Moreover, the proposed photo based portion selection method is shown to be more accurate and engages the users better than the existing methods

12.FUTURE SCOPE

In future there is a development of web application as compared with this and we'll be adding more features which will benefit the users. The ui of the web application will be improved. Scaling the project for more use cases and customers. Implementing distributed computing for efficient processing. Making encryption standard for cloud storage.

13.APPENDIX

13.1 Source Code

app.py

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

userid = 'dhivya'
apikey = ''
appid = 'dhivya'
model = 'food-item-recognition'
hostname = ""
uid = ""
pwd = ""
```

```
database = ""

port = ""

app = Flask(_name_)

app.secret_key = "Dhivya"

conn=ibm_db.connect(f"DATABASE={database};HOSTNAME={hostname};PORT={port};SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID={uid};PWD={pwd};", "", "")

@app.route("/")

def main():

    return render_template("index.html")

@app.route("/dash")

def dash():

    return render_template("dash.html")

@app.route("/upload")

def upload():

    return render_template("upload.html")

    @app.route("/uploader", methods = ["POST","GET"])

def uploader():

    if request.method == "POST":

        f = request.files["file"]

        image = f.read()

    channel = ClarifaiChannel.get_grpc_channel()

        stub = service_pb2_grpc.V2Stub(channel)
```

```
metadata = (('authorization', 'Key ' + apikey),)
userDataObject = resources_pb2.UserAppIDSet(user_id=userid, app_id=appid)
post_model_outputs_response = stub.PostModelOutputs(
    service_pb2.PostModelOutputsRequest(
        user_app_id=userDataObject,
        model_id=model,
        inputs=[
        resources_pb2.Input(
            data=resources_pb2.Data(
                image=resources_pb2.Image(
                    base64=image
                )
            )
        )
    ]
    ),
    metadata=metadata
)
ifpost_model_outputs_response.status.code!= status_code_pb2.SUCCESS:
    print(post_model_outputs_response.status)
    itemmsg = "Upload a valid image"
    return render_template("upload.html",itemmsg=itemmsg)
```

```
      output = post_model_outputs_response.outputs[0]
   item = output.data.concepts[0].name
        print(item)
        url="https://food-nutritioninformation.p.rapidapi.com/foods/search"
querystring = {"query":f"{item}","pageSize":"1","pageNumber":"1"}
 headers = {
            "X-RapidAPI-Key": "",
            "X-RapidAPI-Host":"food-nutrition-information.p.rapidapi.com"
        }
response = requests.request("GET", url, headers=headers, params=querystring)
 r = response.text
        jsondata = json.loads(r)
        lst = []
        for i in range(len(jsondata['foods'][0]['foodNutrients'])):
            print(jsondata['foods'][0]['foodNutrients'][i]['nutrientName'])
          lst.append(jsondata['foods'][0]['foodNutrients'][i]['nutrientName'])
print(lst)
        lstlen = len(lst)
        return render_template("value.html",item=item,lstlen=lstlen,lst=lst)
    itemmsg = "Upload a valid image"
    return render_template("upload.html",itemmsg=itemmsg)
    if _name_ == "_main_":
        app.run(use_reloader=True,debug=True)
```

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

@app.route("/signinvalid",methods = ["POST","GET"])
def signinv():
    if request.method == "POST":
        global username
        global password
        username = request.form.get("usernamel")
        password = request.form.get("passwdl")
        msgl = ""
        stringl = ""
        sqll = "SELECT * FROM REGISTER WHERE USERNAME =? AND PASSWORD =?"
        stmtl = ibm_db.prepare(conn, sqll)
        ibm_db.bind_param(stmtl,1,username)
        ibm_db.bind_param(stmtl,2,password)
        ibm_db.execute(stmtl)
        accountl = ibm_db.fetch_assoc(stmtl)
        print(accountl)
        if accountl:
            session["username"] = accountl["USERNAME"]
            stringl = f"{username} login success"
            return render_template("dash.html",stringl=stringl)
        else:
            msgl = "Incorrect Username and Password"
    return render_template("signin.html",msgl=msgl)
@app.route("/signup")
def signup():
    return render_template("signup.html")
@app.route("/signupvalid",methods = ["POST","GET"])
def signupv():
    if request.method == "POST":
        global mail
        global user
        global passwd
        mail = request.form.get("emailaddress")
        user = request.form.get("username")
        passwd = request.form.get("passwd")
        msg = ""
        userstatus = ""
        mailstatus = ""
```

```
passwdstatus = ""


sql = "SELECT * FROM REGISTER WHERE USERNAME =?"
stmt = ibm_db.prepare(conn, sql)
ibm_db.bind_param(stmt,1,user)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print(account)
if account:
    msg = "Account already exists"
elif not re.match(r'[A-Za-z0-9]+', user):
    userstatus = "Please enter valid username"
elif not re.match(r'[^@]+@[^@]+\.[^@]+', mail):
    mailstatus = "Please enter valid email"
elif (passwd==""):
    passwdstatus = "Please enter valid password"
else:
    sentotp(user,mail)
    return render_template("validate.html")
return render_template("signup.html",mailstatus=mailstatus,userstatus=userstatus,passwdstatus=passwdstatus,msg=msg)
```

```
@app.route("/checkotp",methods = ["POST","GET"])

def checkotp():

    if request.method == "POST":

        rotp = request.form.get("otp")

        if (str(rotp)==str(sotp)):

            sql1="INSERT INTO REGISTER(USERNAME,PASSWORD,MAIL)
VALUES(?,?,?)"

            stmt1 = ibm_db.prepare(conn, sql1)

                        ibm_db.bind_param(stmt1,1,user)

            ibm_db.bind_param(stmt1,2,passwd)

            ibm_db.bind_param(stmt1,3,mail)

            ibm_db.execute(stmt1)

result = "Account Created Succesfully"

            return render_template("result.html",result=result)

        else:

            status = "Please enter valid OTP"

            return render_template("validate.html",status=status)

def sentotp(user,mail):

    global sotp

    sotp = random.randint(1000,9999)

    message = Mail(

    from_email='dhivyavpy@gmail.com',

    to_emails=mail,
    subject='Otp verification',
    html_content=f'Hello {user} This is OTP - {sotp}')
    sg = SendGridAPIClient("")
    response = sg.send(message)
@app.route("/logout")
def logout():
    session.pop("username", None)
    return render_template("/index.html")
if _name_ == "_main_":
    app.run(debug=True)
```

base.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- CSS only -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
```

```
rel="stylesheet" integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">

  <title>Nutrition Assistant Application</title>

</head>

<body background="https://w.wallha.com/ws/1/VwLlk1Xb.jpg">

<nav class="navbar navbar-expand-lg bg-light">

  <div class="container-fluid">

    <a class="navbar-brand" href="/">Nutrition Assistant Application</a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">

      <span class="navbar-toggler-icon"></span>

    </button>

    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">

      <div class="navbar-nav">

        <a class="nav-link active" aria-current="page" href="/">Home</a>

        <a class="nav-link active" aria-current="page" href="/signup">Sign up</a>

        <a class="nav-link active" aria-current="page" href="/signin">Sign in</a>

      </div>

    </div>

  </div>

</nav>

{% block content %}
{% endblock %}

<!-- JavaScript Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8" crossorigin="anonymous"></script>

</body>

</html>
```

index.html

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

{% block title %}Index Page{% endblock %}

{% block content %}

<!DOCTYPE html>

<html>

<head>

        <meta charset="utf-8">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <title>Nutrition Assistant Application</title>

</head>

<body>

<center>

        <h2>WELCOME</h2>

        <br><br><br>
        <h3><i>Stay Healthy &nbsp;&nbsp;Stay Happy!!!!</i>
        <br><br><br>
        <img src="https://cdn.pixabay.com/photo/2017/10/09/19/29/eat-2834549__340.jpg">
        </center>

</body>
</html>
{% endblock %}
```

signup

```
{% extends "base.html" %}
{% block title %}Signup Page{% endblock %}
{% block content %}
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Sign up</title>
</head>
<center>
<body>
    <h1>Sign Up</h1>
    <form action="{{ url_for("signupv")}}" method="POST">
        <label>Username:</label>
        <input type="text" name="username">
        <span style="color:red">{{ userstatus }}</span>
        <br><br>
        <label>E-mail:</label>
        <input type="email" name="emailaddress">
        <span style="color:red">{{ mailstatus }}</span>
        <br><br>
        <label>Create Password:</label>
        <input type="password" name="passwd">
        <span style="color:red">{{ passwdstatus }}</span>
        <br><br>
       <input type="submit" name="signup">
        <br><br>
        <span style="color:red">{{ msg }}</span>



    </form>
</body>
</center>
</html>
```

```
{% endblock %}
```

Signin.html

```
{% extends "base.html" %}
{% block title %}Signin Page{% endblock %}
{% block content %}
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Sign in</title>
</head>
<center>
<body>
    <h1>Sign In</h1>

    <form action="{{ url_for("signinv")}}" method="POST">
        <label>Username:</label>
        <input type="text" name="usernamel">
        <span style="color:red">{{ userstatusl }}</span>
        <br><br>
        <label>Password:</label>
        <input type="password" name="passwdl">
        <span style="color:red">{{ passwdstatusl }}</span>
        <br><br>
 <input type="submit" name="signin">
        <br><br>
        <span style="color:red">{{ msgl }}</span>
</form>
</body>
</center>
 </html>
{% endblock %}
```

result.html

```
{% extends "base.html" %}
{% block title %}Index Page{% endblock %}
{% block content %}
<!DOCTYPE html>
<html>
<head>
          <meta charset="utf-8">
          <meta name="viewport" content="width=device-width, initial-scale=1">
          <title>Account</title>
</head>
```

```
<center>
<body>
<h1>{{ result }}</h1>
</body>
</center>
</html>
{% endblock %}
```

validate.html

```
{% extends "base.html" %}
{% block title %}Validate Page{% endblock %}
{% block content %}
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Validation Page</title>
</head>
<center>
<body>

<h1>OTP is sent to your mail</h1>
    <form action="{{ url_for("checkotp")}}" method="POST">
```

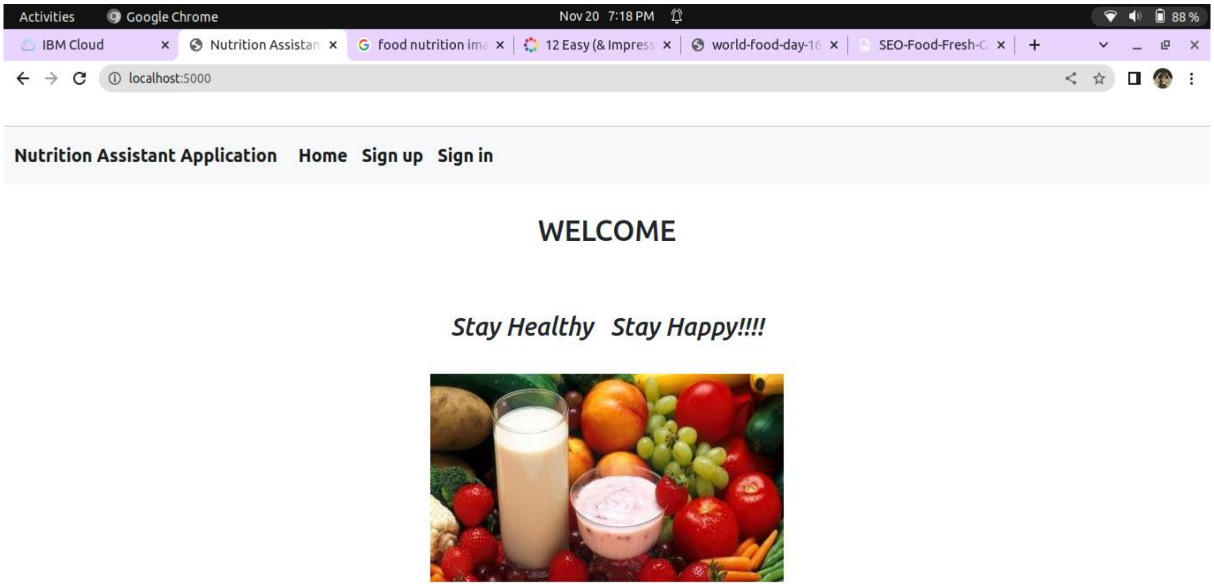
```
        <label>Enter OTP:</label>
        <input type="number" id="otp" name="otp">
        <span style="color:red">{{ status }}</span>
        <br><br>

        <input type="submit" name="signup">
        <br><br>
    </form>

</body>
</center>
</html>
{% endblock %}
```

upload.html

```
{% extends "basel.html" %}
{% block title %}Dashboard{% endblock %}
{% block content %}
<html>
<body background="https://w.wallha.com/ws/1/VwLlk1Xb.jpg">
<center>
<body >
<form action="upload.php" method="post" enctype="multipart/form-data"><br><br>
```

```
<center>

<h1 style="font-size: 35px;"> Select Image to Upload</h1><br><br>

</center>

<input style= "font-size: 25px;"  type="file" name="fileToUpload"
id="fileToUpload">

<input style= "font-size: 25px;" type="submit" value="Upload Image"
name="submit"></br>

</form> </center></body>

</html>

{% endblock %}
```

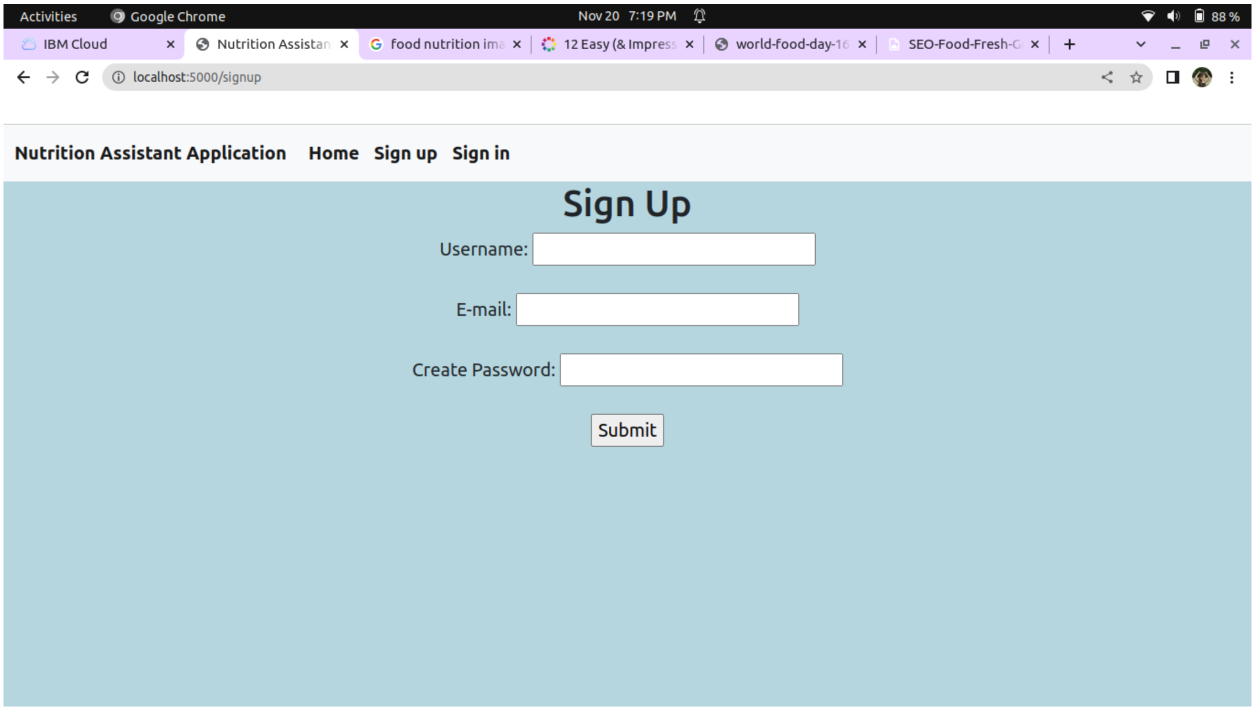
13.2 Screenshots



Nutrition Assistant Application **Home** **Sign up** **Sign in**

WELCOME

Stay Healthy Stay Happy!!!!

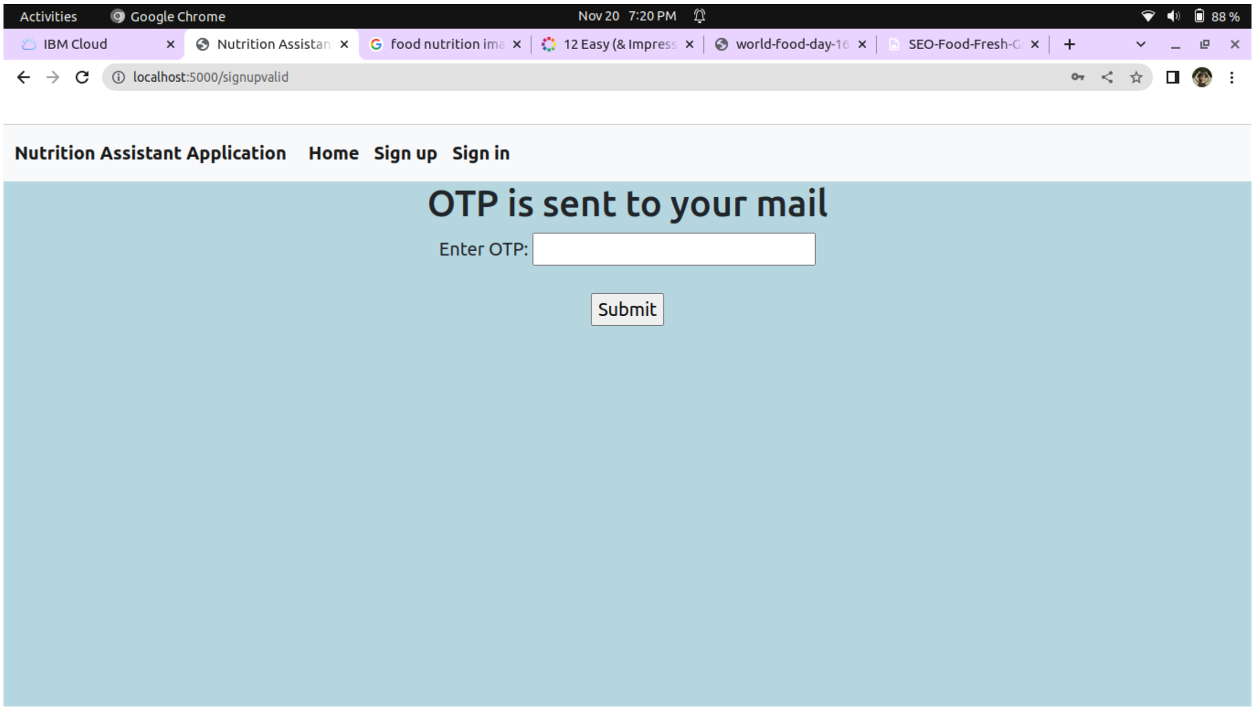


Sign Up

Username:

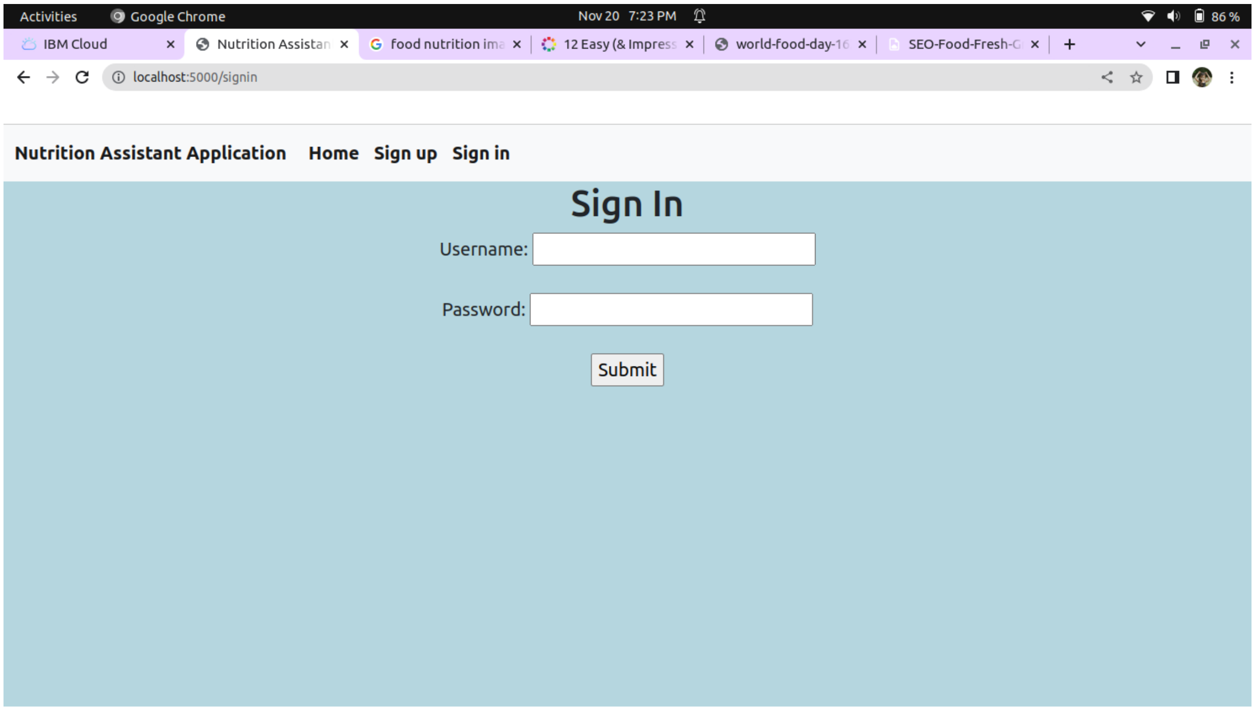
E-mail:

Create Password:



OTP is sent to your mail

Enter OTP:

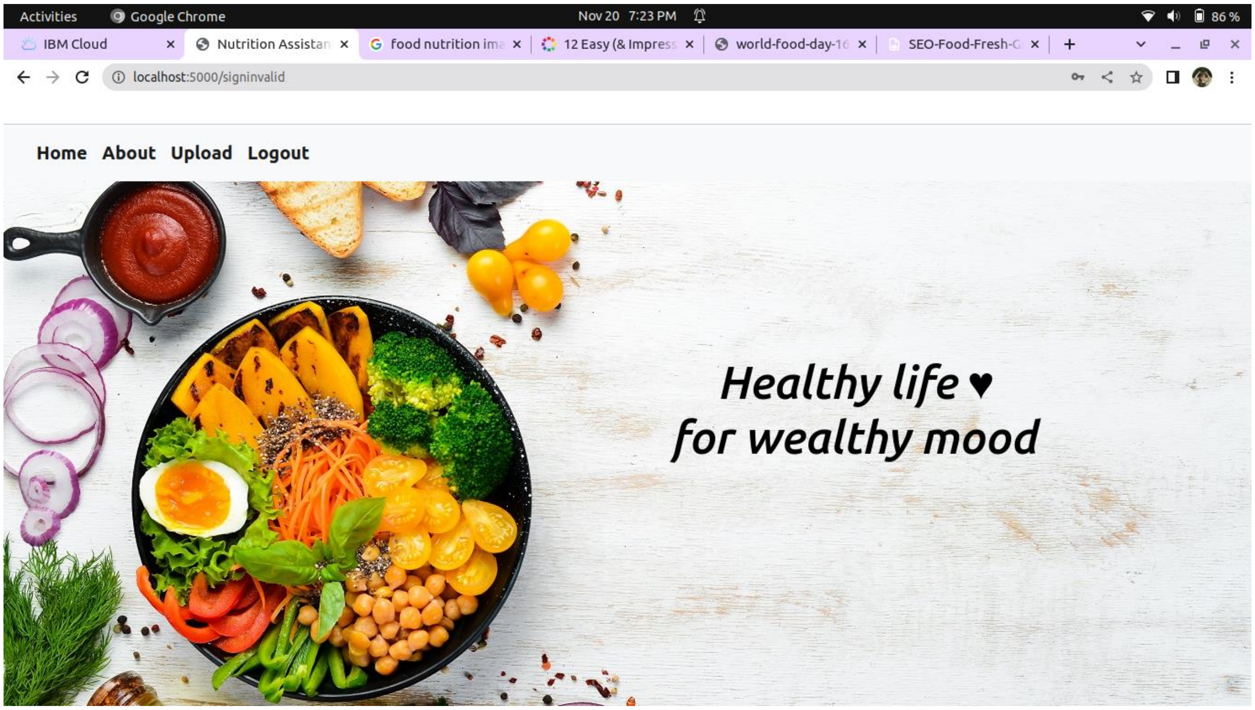


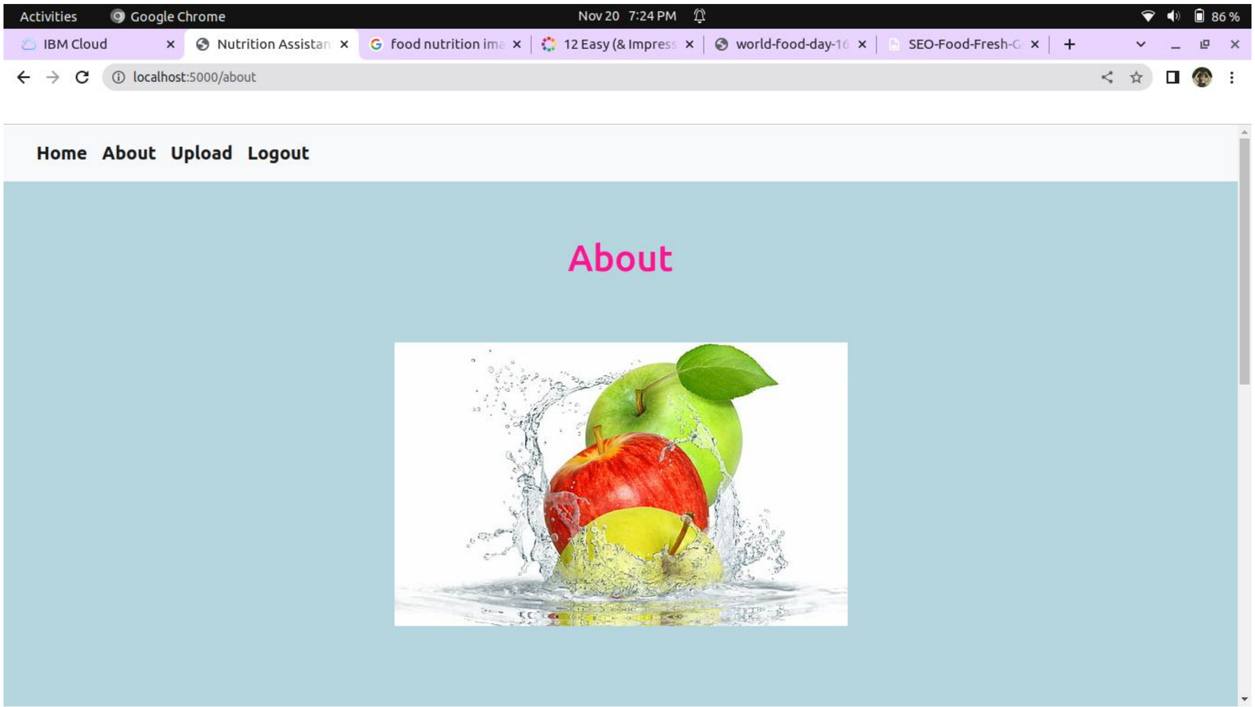
Sign In

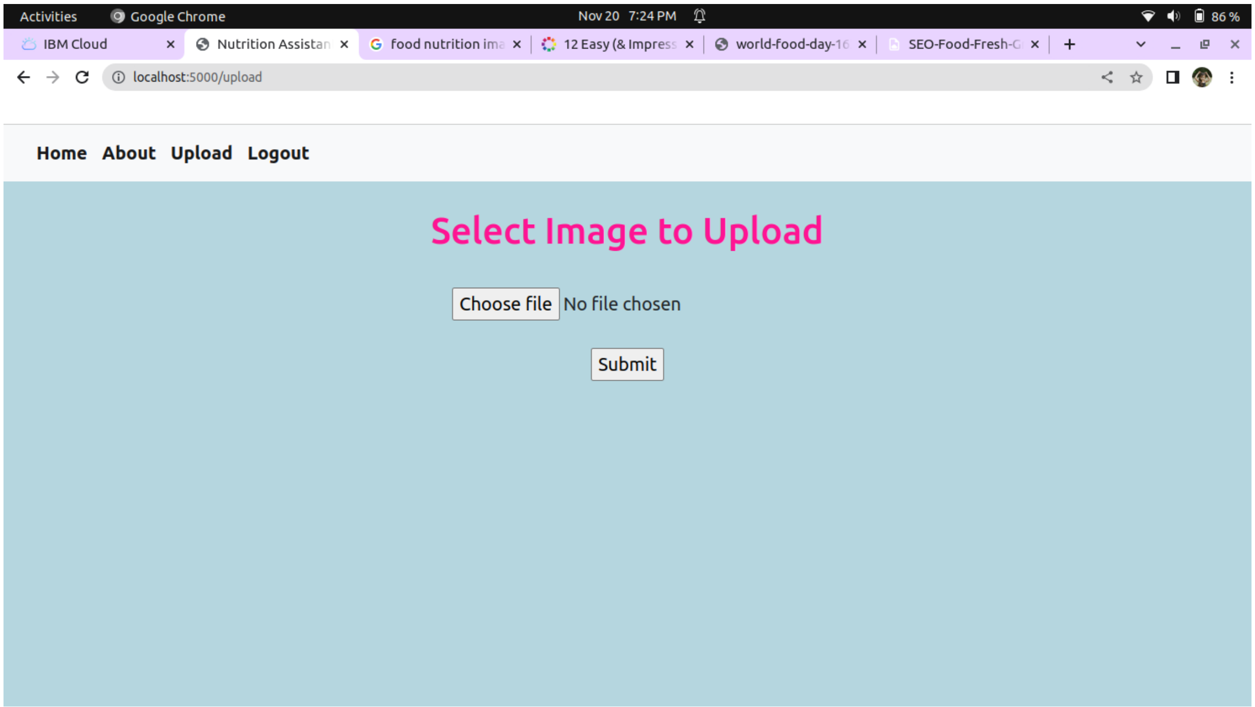
Username:

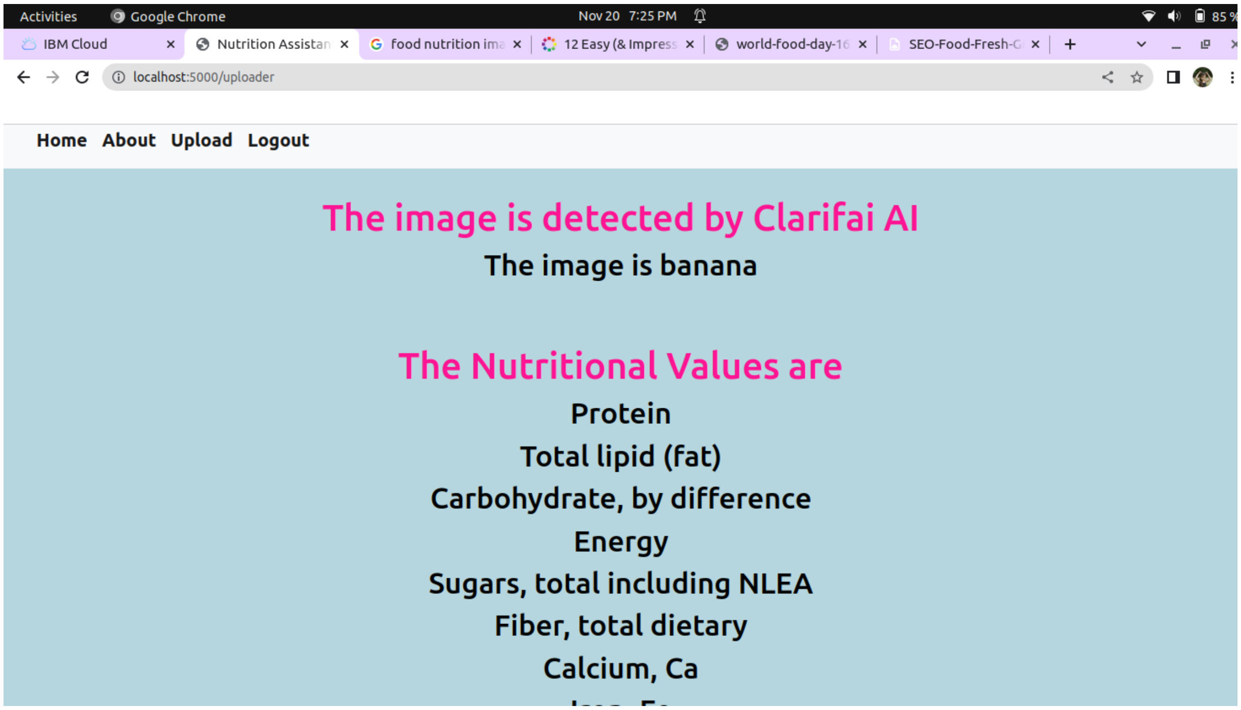
Password:

Submit




Activities
Google Chrome
Nov 20 7:24 PM
86 %
IBM Cloud
Nutrition Assistan
food nutrition ima
12 Easy (& Impress
world-food-day-16
SEO-Food-Fresh-G
localhost:5000/about
Home
About
Upload
Logout
About


Activities
Google Chrome
Nov 20 7:24 PM
86 %
IBM Cloud
Nutrition Assistan
food nutrition ima
12 Easy (& Impress
world-food-day-16
SEO-Food-Fresh-G
localhost:5000/upload
Home
About
Upload
Logout
Select Image to Upload
Choose file
No file chosen
Submit



The image is detected by Clarifai AI

The image is banana

The Nutritional Values are

Protein

Total lipid (fat)

Carbohydrate, by difference

Energy

Sugars, total including NLEA

Fiber, total dietary

Calcium, Ca

13.3 GitHub Link

https://github.com/IBM-EPBL/IBM-Project-36421-1660294982.git