NUTRITION ASSISTANT APPLICATION

PTN2022TMID27311

PROJECT REPORT

1 INTRODUCTION

1.1 PROJECT OVERVIEW

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 gives accurate food identification and Food API's to give the nutritional value of the identified food.

1.2 PURPOSE

To help to maintain diet balance for users 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 LITERATURE SURVEY

2.1 EXISTING PROBLEM

Unhealthy diets have been identified as the important causing factor of such diseases. In this context, personalized nutrition emerges as a new research field for providing tailored food intake advice to individuals according to their physical, physiological data, and further personal information. Specifically, in the last few years, several types of research have proposed computational models for personalized food recommendation using nutritional knowledge and user data.

2.2REFERENCES

Nutritional_biomarkers_and_machine_learning_for_personalized_nutrition_applications_and_health_optimization

https://www.researchgate.net/publication/360530930

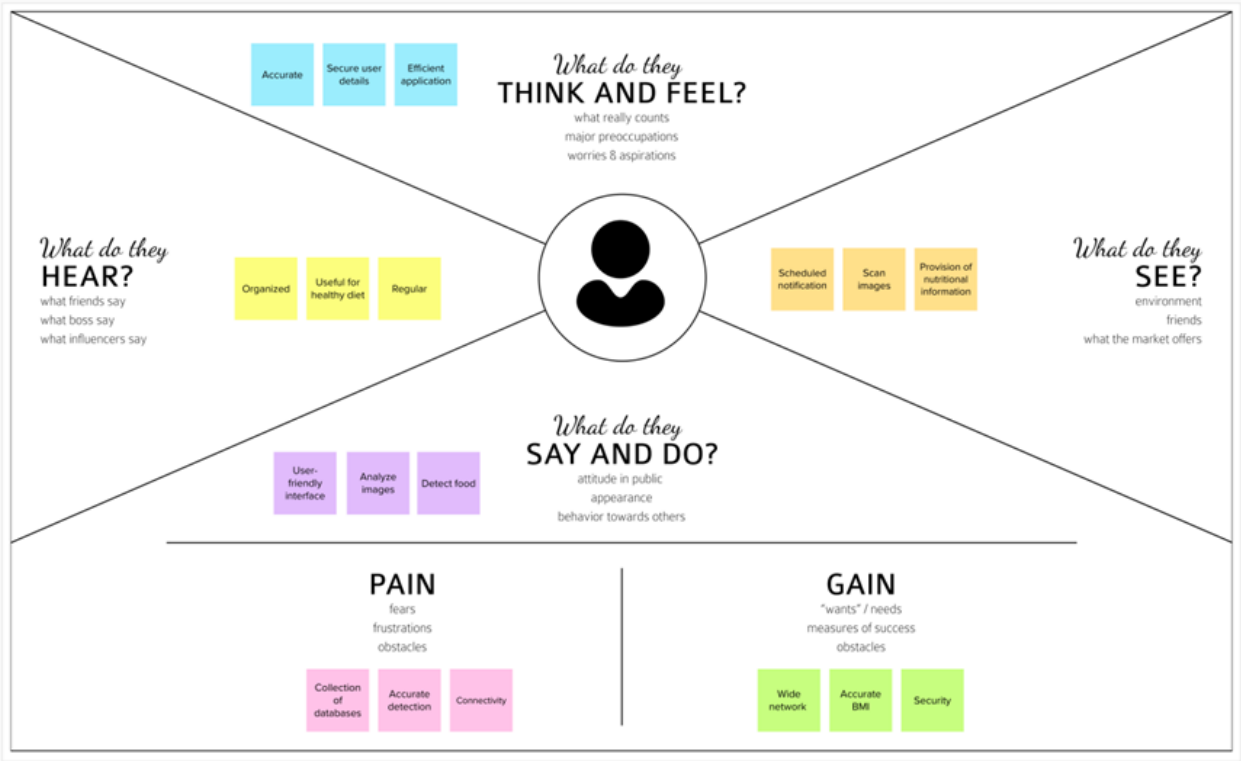
https://www.researchgate.net/publication/364203081

2.3 PROBLEM STATEMENT

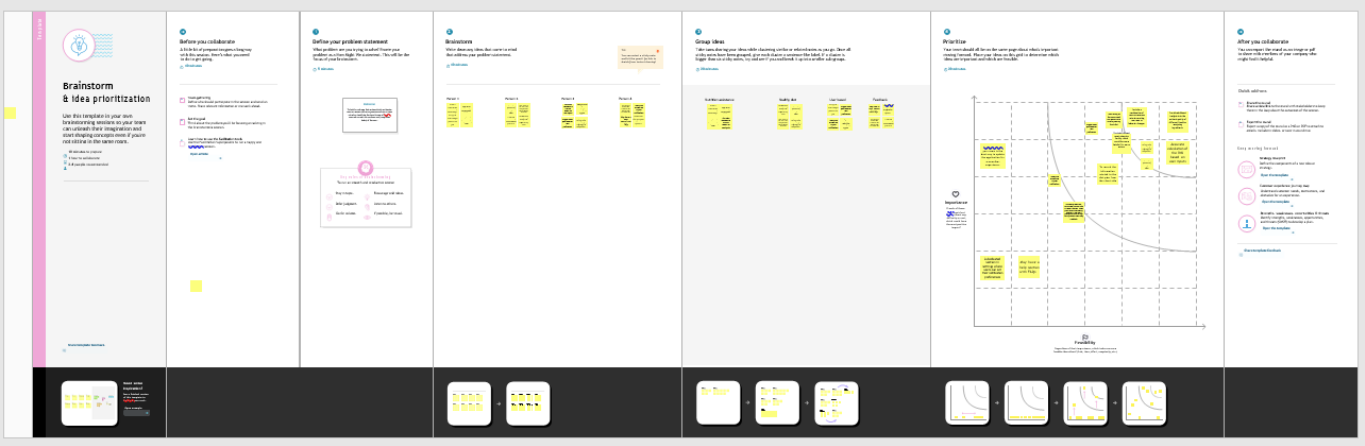
To build a web App that automatically estimates food attributes such as ingredients and nutritional value by classifying the input image of food. Our method gives accurate food identification and Food API's to give the nutritional value of the identified food.

3 IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING

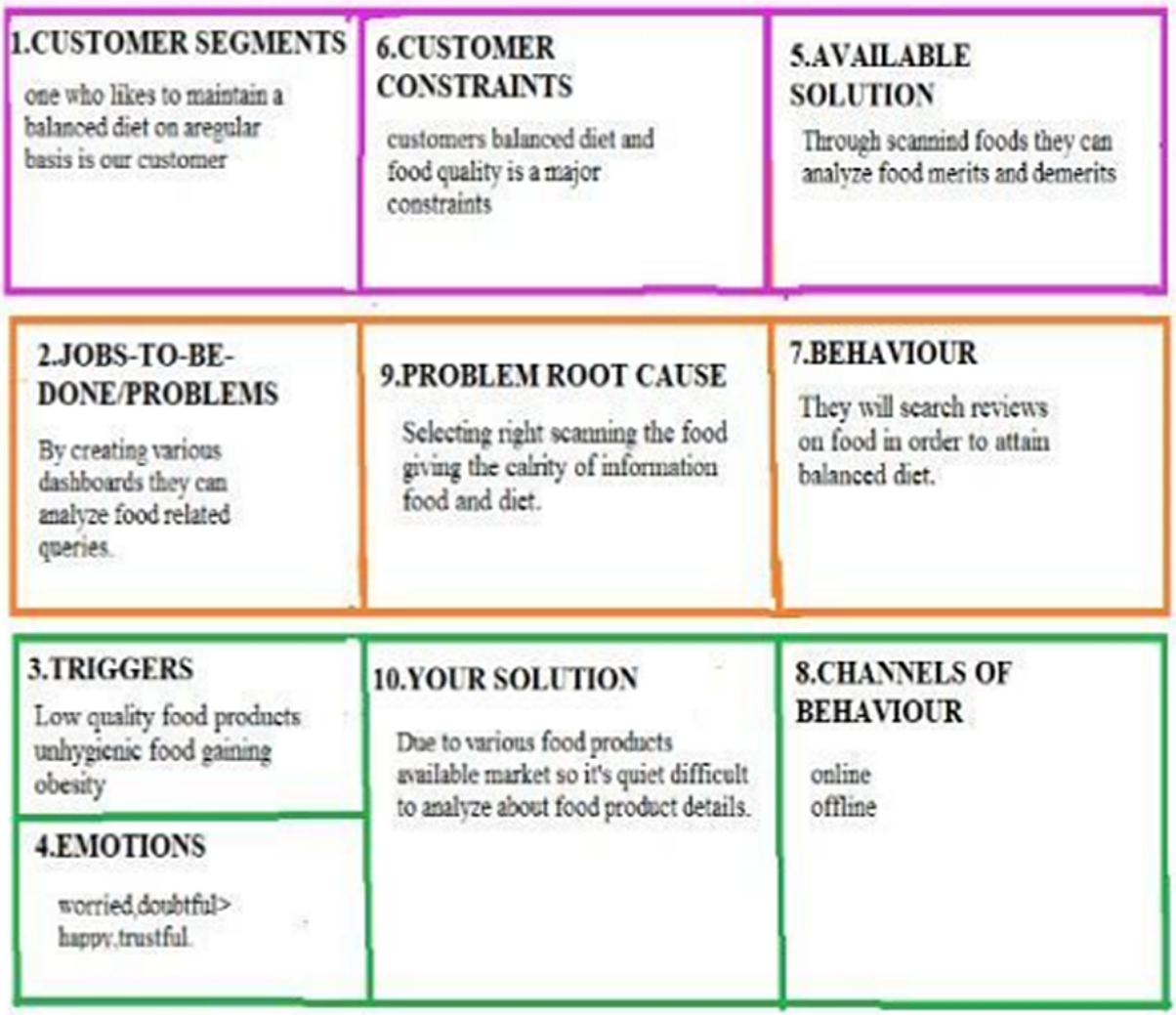


3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To provide nutrition assistance to the user by displaying nutrients of the scanned food.
2.	Idea / Solution description	To display nutrients of the scanned food. To recommend food based on the BMI calculation.
3.	Novelty / Uniqueness	This project is unique for its multi-functionality. In this project, various functions are combined as a single app.
4.	Social Impact / Customer Satisfaction	Our system offers automated personalized visual feedback and recommendations based on individual dietary behavior, phenotype, and preferences.

5.	Business Model (Revenue Model)	It includes cost of equipment, services and fee paid to technology providers by initial development fund and local partners' fund.
6.	Scalability of the Solution	The database base can be updated accordingly. The input details can be changed by the authorized user anytime.

3.4 PROBLEM SOLUTION FIT



4 REQUIREMENT ANALYSIS

4.1 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
FR-2	User Confirmation	Confirmation via Email
FR-3	Food Detection	Scanning by Clarifai's AI-Driven Food Detection Model
FR-4	Nutrients Display	Display nutrients through IBM Cloud
FR-5	User BMI Calculation	Calculating Body Mass Index accurately

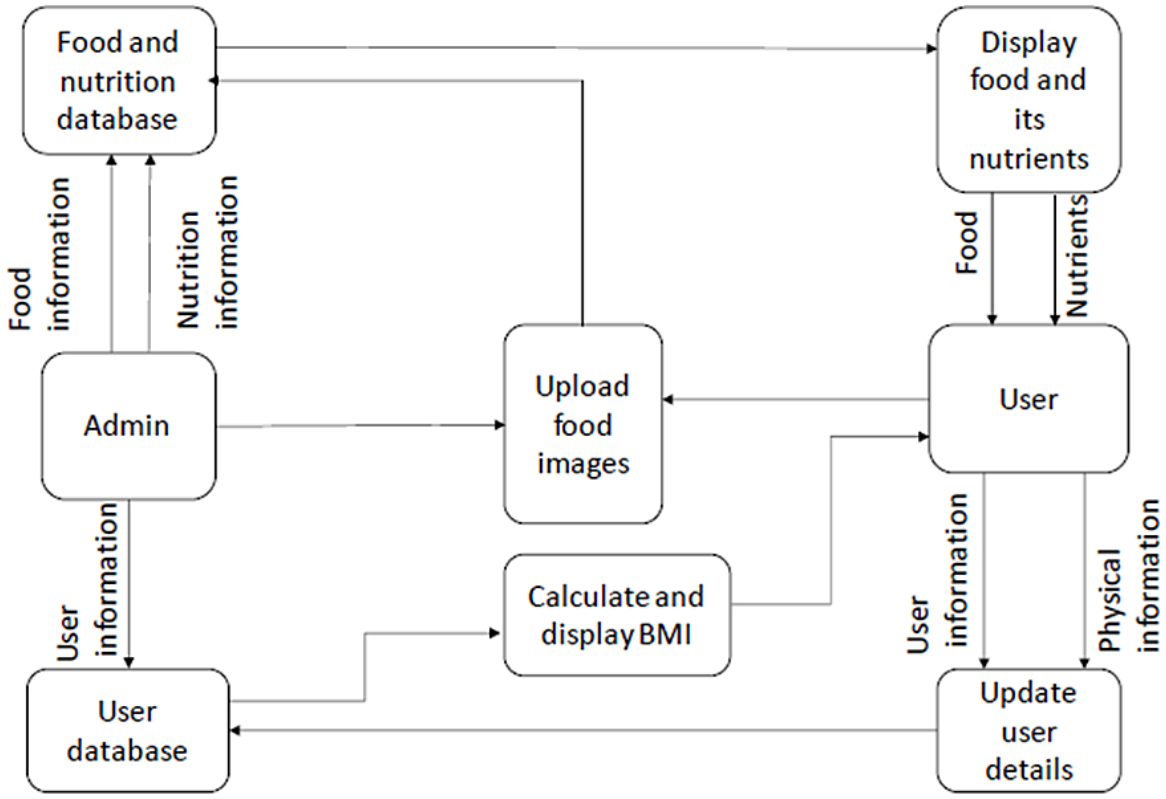
4.2 NON-FUNCTIONAL REQUIREMENTS:

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

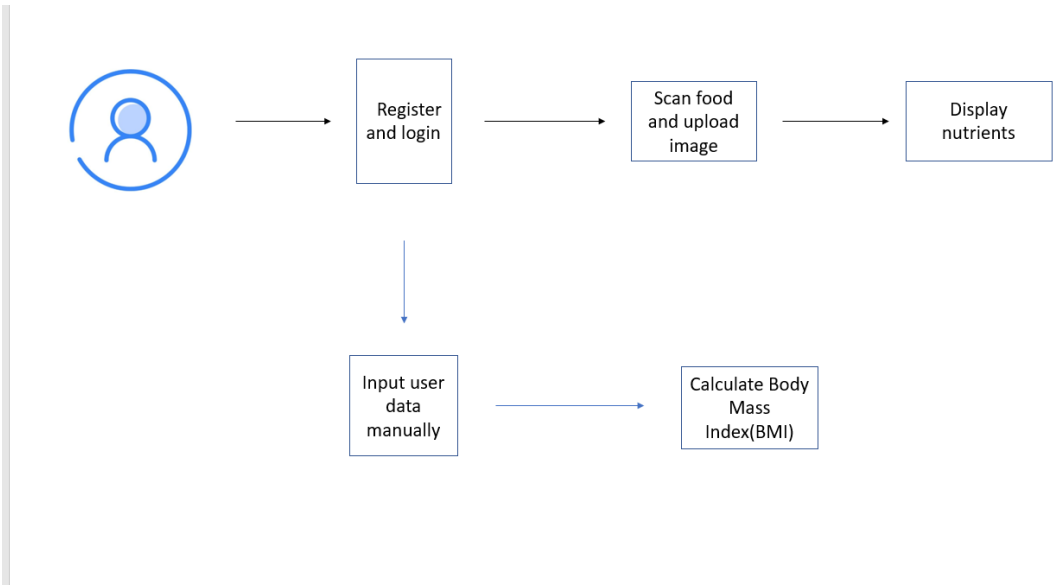
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Usage of Python as a programming language Flask as a Python framework Docker as a software platform IBM cloud as a cloud storage IBM DB2 as a database support
NFR-2	**Security**	Maintain user credentials such as username and user details in a secure manner.
NFR-3	**Reliability**	Reliable as it includes accurate BMI calculation and efficient nutrient display.
NFR-4	**Performance**	Provision of relevant scanning of food and best diet plan which makes the user follow a healthy diet.
NFR-5	**Availability**	Easily accessible as the user requires only a smart-phone with a good network connection.
NFR-6	**Scalability**	The database base can be updated accordingly. The input details can be changed by the authorized user anytime.

5 PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 USER STORIES

Functional Requirement (Epic)	User Story Number	User Story / Task
User Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.
User Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application
Food Detection	USN-3	As a user, I will upload the food image and the food will be detected
Nutrition Display	USN-4	As a user, I can view the nutrition contents of the detected food
User BMI Calculation	USN-5	As a user, I can view my calculated Body Mass Index

6 PROJECT PLANNING & SCHEDULING

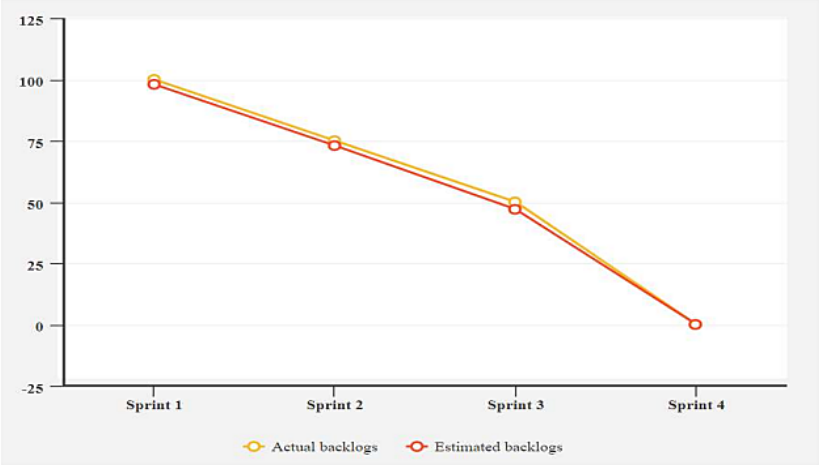
6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	R.Varsha , S.Sankari , T.Ooviya , M.Umadevi.
Sprint-1	User Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	R.Varsha , S.Sankari , T.Ooviya , M.Umadevi.
Sprint-2	Food Detection	USN-3	As a user, I will upload the food image and the food will be detected	2	High	R.Varsha , S.Sankari , T.Ooviya , M.Umadevi.
Sprint-3	Nutrition Display	USN-4	As a user, I can view the nutrition contents of the detected food	2	High	R.Varsha , S.Sankari , T.Ooviya , M.Umadevi.
Sprint-4	User BMI Calculation	USN-5	As a user, I can view my calculated Body Mass Index	1	Low	R.Varsha , S.Sankari , T.Ooviya , M.Umadevi.

6.2 SPRINT DELIVERY SCHEDULE

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

6.3 REPORTS FROM JIRA



7 CODING & SOLUTIONING

7.1 SPRINT 1

LOGIN.HTML

```
<!DOCTYPE html>
                    <html>
                      <head>
                        <title> Login Page</title>
                        <style>
                          body{
                            background-image: url("image.jpg");
                            background-repeat: no-repeat;
                            background-image: cover;

                          }
                        </style>
                      </head>
                      <body>
```

```
                    <form action="/Login" method="POST">
                    <center> <h1> Login </h1>
                    <label>Username:</label>
                    <input type="text" placeholder="Enter Username" name="username" required>
                    <br></br>
                    <label>Password:</label>
                    <input type="password" placeholder="Enter password " name="password" required>
                    <br></br>
                    <button type="submit"> Login</button>
                    <br></br>
                    </center>
                    </form>

                  </body>
                </html>
```

REGISTER.HTML

```
 !DOCTYPE html>
                  <html>
                    <head>
                      <title>Register page</title>
                      <style>
                      body{
                        background-image: url("Naa.jpg");
                        background-repeat:no-repeat;
                        background-size: 100% 100%;

                      }
                      </style>
                    </head>
                    <body>
                      <form action="/Register" method="POST">
                      <center> <h1>Singup</h1>
                      <label><b>username:</b></label>
                      <input type="text" placeholder="Enter Username" name="username" required>
                      <br></br>
                      <label><b>password:</b></label>
                      <input type="password" placeholder="Enter password " name="password" required>
                      <br></br>
                      <label><b>phonenumber:</b></label>
                      <input type="phonenumber" placeholder="Enter phone number " name="phonenumber" required>
                      <br></br>
```

```
    <label><b>emailid:</b></label>
    <input type="emailid" placeholder="Enter email id" name="emailid" required>
    <br></br>
    <button type="submit"> submit</button>
    <br></br>
    </center>
    </form>

  </body>
</html>
```

DASH.HTML

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: "Lato", sans-serif;
   background-image: url("image.jpg");
   background-repeat: no-repeat;
   background-image: cover;
}

.sidenav {
  height: 100%;
  width: 0;
  position: fixed;
  z-index: 1;
  top: 0;
  left: 0;
  background-color: #111;
  overflow-x: hidden;
  transition: 0.5s;
  padding-top: 60px;
}

.sidenav a {
  padding: 8px 8px 8px 32px;
  text-decoration: none;
  font-size: 25px;
  color: #818181;
  display: block;
```

```
  transition: 0.3s;
}

.sidenav a:hover {
  color: #f1f1f1;
}

.sidenav .closebtn {
  position: absolute;
  top: 0;
  right: 25px;
  font-size: 36px;
  margin-left: 50px;
}

@media screen and (max-height: 450px) {
  .sidenav {padding-top: 15px;}
  .sidenav a {font-size: 18px;}
}
</style>
</head>
<body>
<form action="/dash" method="POST">
<div id="mySidenav" class="sidenav">
  <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>
  <a href="Login.html">Home</a>
  <a href="Register.html">Register</a>
  <a href="upload.html">Upload Image</a>
  <a href="#">Food Items</a>
  <a href="BMI_Calculation.html">BMI Calculation</a>
  <a href="ref.html">Logout</a>
</div>
</form>


<span style="font-size:30px;cursor:pointer" onclick="openNav()">&#9776; Menubar</span>

<script>
function openNav() {
  document.getElementById("mySidenav").style.width = "250px";
}

function closeNav() {
```

```
  document.getElementById("mySidenav").style.width = "0";
}
</script>

</body>
</html>
```

7.2 SPRINT 2

UPOLOAD.HTML

```
<!DOCTYPE html>
<html>
<head>
   <title>select the file</title>
   <style>
      body{
         background-color: #ffb6c1;
      }
   </style>
</head>
<body>
<form action="/upload" method="POST">
<center>
<label for="myfile">Select a file:</label>
<input type="file" id="myfile" name="myfile" /> <br/><br/>
<input type="submit" value="submit" />
</center>



<P><b>This page helpful for check the nutrition value</b></P>
</form>
</body>
</html>
```

7.3 SPRINT 3

```
import streamlit as st
from PIL import Image
from keras_preprocessing.image import img_to_array
import numpy as np
from keras.models import load_model
import requests
from bs4 import BeautifulSoup
```

```
model = load_model('FV.h5')
labels = {0: 'apple', 1: 'banana', 2: 'beetroot', 3: 'bell pepper', 4: 'cabbage', 5: 'capsicum', 6: 'carrot', 7: 'cauliflower', 8: 'chilli pepper', 9: 'corn', 10: 'cucumber', 11: 'eggplant', 12: 'garlic', 13: 'ginger', 14: 'grapes', 15: 'jalepeno', 16: 'kiwi', 17: 'lemon', 18: 'lettuce',
        19: 'mango', 20: 'onion', 21: 'orange', 22: 'paprika', 23: 'pear', 24: 'peas', 25: 'pineapple', 26: 'pomegranate', 27: 'potato', 28: 'raddish', 29: 'soy beans', 30: 'spinach', 31: 'sweetcorn', 32: 'sweetpotato', 33: 'tomato', 34: 'turnip', 35: 'watermelon'}

fruits = ['Apple','Banana','Bello Pepper','Chilli Pepper','Grapes','Jalepeno','Kiwi','Lemon','Mango','Orange','Paprika','Pear','Pineapple','Pomegranate','Watermelon']
vegetables = ['Beetroot','Cabbage','Capsicum','Carrot','Cauliflower','Corn','Cucumber','Eggplant','Ginger','Lettuce','Onion','Peas','Potato','Raddish','Soy Beans','Spinach','Sweetcorn','Sweetpotato','Tomato','Turnip']

def fetch_calories(prediction):
        url = 'https://www.google.com/search?&q=calories in ' + prediction
        req = requests.get(url).text
        scrap = BeautifulSoup(req, 'html.parser')
        calories = scrap.find("div", class_="BNeawe iBp4i AP7Wnd").text
        return calories

def processed_img(img_path):
    img=load_img(img_path,target_size=(224,224,3))
    img=img_to_array(img)
    img=img/255
    img=np.expand_dims(img,[0])
    answer=model.predict(img)
    y_class = answer.argmax(axis=-1)
    print(y_class)
    y = " ".join(str(x) for x in y_class)
    y = int(y)
    res = labels[y]
    print(res)
    return res.capitalize()

def run():
    st.title("Fruits-Classification")
    img_file = st.file_uploader("Choose an Image", type=["jpg", "png"])
    if img_file is not None:
        img = Image.open(img_file).resize((250,250))
        st.image(img,use_column_width=False)
        save_image_path = './upload_images/'+img_file.name
        with open(save_image_path, "wb") as f:
            f.write(img_file.getbuffer())

        # if st.button("Predict"):
        if img_file is not None:
            result= processed_img(save_image_path)
            print(result)
```

```
        if result in vegetables:
            st.info('**Category : Vegetables**')
        else:
            st.info('**Category : Fruit**')
        st.success("**Predicted : "+result+'**')
        cal = fetch_calories(result)
        if cal:
            st.warning('**'+cal+'(100 grams)**')


run()
```

7.4 SPRINT 4

BMI CALCULATION.HTML

```
<!DOCTYPE html>
                        <html lang="en">
                        <head>
                            <meta charset="UTF-8">
                            <meta http-equiv="X-UA-Compatible" content="IE=edge">
                            <meta name="viewport" content="width=device-width, initial-scale=1.0">
                            <title>Document</title>
                            <style>
                                body{
                                  background-image: url("image.jpg");
                                  background-repeat: no-repeat;
                                  background-image: cover;
                                }
                            </style>
                        </head>
                        <body>
                            <form action="/BMI_Calculation" method="POST">
                            <div class="calculator-container">
                                <center>
                                <h1>BMI CALCULATOR</h1>
                                <label>Height:</label>
                                <input class="Height-input-field" type="text">
                                <br></br>
                                <label>Weight:</label>
                                <input class="Weight-input-field" type="text"><br>
                                <button class="calculate"> Calculate</button>
                            </div>
                            <h3 class="result"></h3>
                            <p class="result-statement"></p>
                            <script src="script.js"></script>
```

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

8 TESTING

8.1 TESTCASES

				Date	29 Nov 22							
				Team ID	PNT2022TMID27311							
				Project Name	Project - Nutrition Assistant Appli							
				Maximum Marks	4 marks							
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_OO 1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Signup popup displayed or not	[illegible]	Login/Signup popup should display	Working as expected	Pass				S.Sankari
LoginPage_TC_OO 2	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Signup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Lost password? Recovery password link	[illegible]	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Lost password? Recovery password link	Working as expected	Fail	Steps are not clear to follow		BUG-1234	R.Varsha
LoginPage_TC_OO 3	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter URL(https://shopnuser.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in email text box 4.Enter valid password in password text box 5.Click on login button	User name: [illegible]@gmail.com password: Testing123	User should navigate to user account homepage	Working as expected	Pass				M.Umadevi
LoginPage_TC_OO 4	Functional	Login page	Verify user is able to log into application with Initial credentials		1.Enter URL(https://shopnuser.com/) and click go 2.Click on My Account dropdown button 3.Enter Initial username/email in email text box 4.Enter valid password in password text box 5.Click on login button	User name: abcabc@gmail password: Testing123	Application should show "Incorrect email or password" validation message.	Working as expected	Fail	Need to verify			T.Divya
LoginPage_TC_OO 4	Functional	Login page	Verify user is able to log into application with Initial credentials		1.Enter URL(https://shopnuser.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in email text box 4.Enter Invalid password in password text box 5.Click on login button	User name: [illegible]@gmail.com password: [illegible]	Application should show "Incorrect email or password" validation message.	Working as expected	Pass				R.Varsha
LoginPage_TC_OO 5	Functional	Login page	Verify user is able to log into application with Initial credentials		1.Enter URL(https://shopnuser.com/) and click go 2.Click on My Account dropdown button 3.Enter Initial username/email in email text box 4.Enter Invalid password in password text box 5.Click on login button	User name: Divya password: [illegible]	Application should show "Incorrect email or password" validation message.	Working as expected	Pass				S.Sankari

8.2 USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issuesof the Nutrition Assistant Application project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report showsthe number of resolved or closed bugs at each severity level, and how they were resolved

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

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9 RESULTS

9.1 PERFORMANCE METRICS

10 ADVANTAGES AND DISADVANTAGES

Advantages:

The application reduces manual work.
The user would be able to maintain healthy diet.
The application saves time.

Disadvantages:

It requires manual entry of user details.
The application requires camera with more clarity.

11 CONCLUSION

The development of Nutrition Assistant Application was a good learning experience.Developing this application has given clear idea and knowledge about the cloud application development. This help in familiarising with Cloud and Docker concepts.This gave us insights into real-time software engineering.

12 FUTURE SCOPE

The scope of a Nutrition Assistant Application can cover many needs including prediction of nutrition in a mixed food items.The application may include meal planning according to the user location.The accuracy of the prediction and the variety of input comparison may be incrreased.

13 APPENDIX

```
<!DOCTYPE html>
<html>
  <head>
    <title> Login Page</title>
    <style>
      body{
        background-image: url("image.jpg");
        background-repeat: no-repeat;
        background-image: cover;

      }
    </style>
  </head>
  <body>
    <form action="/Login" method="POST">
    <center> <h1> Login </h1>
    <label>Username:</label>
    <input type="text" placeholder="Enter Username" name="username" required>
    <br></br>
    <label>Password:</label>
    <input type="password" placeholder="Enter password " name="password" required>
    <br></br>
    <button type="submit"> Login</button>
    <br></br>
```

```
      </center>
      </form>

   </body>
</html>
<!DOCTYPE html>
<html>
   <head>
      <title>Register page</title>
      <style>
      body{
         background-image: url("Naa.jpg");
         background-repeat:no-repeat;
         background-size: 100% 100%;

      }
      </style>
   </head>
   <body>
      <form action="/Register" method="POST">
      <center> <h1>Singup</h1>
      <label><b>username:</b></label>
      <input type="text" placeholder="Enter Username" name="username" required>
      <br></br>
      <label><b>password:</b></label>
      <input type="password" placeholder="Enter password " name="password" required>
      <br></br>
      <label><b>phonenumber:</b></label>
      <input type="phonenumber" placeholder="Enter phonenumber " name="phonenumber" required>
      <br></br>
      <label><b>emailid:</b></label>
      <input type="emailid" placeholder="Enter emailid" name="emailid" required>
      <br></br>
      <button type="submit"> submit</button>
      <br></br>
      </center>
      </form>

   </body>
</html>
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: "Lato", sans-serif;
   background-image: url("image.jpg");
   background-repeat: no-repeat;
   background-image: cover;
```

```
}

.sidenav {
  height: 100%;
  width: 0;
  position: fixed;
  z-index: 1;
  top: 0;
  left: 0;
  background-color: #111;
  overflow-x: hidden;
  transition: 0.5s;
  padding-top: 60px;
}

.sidenav a {
  padding: 8px 8px 8px 32px;
  text-decoration: none;
  font-size: 25px;
  color: #818181;
  display: block;
  transition: 0.3s;
}

.sidenav a:hover {
  color: #f1f1f1;
}

.sidenav .closebtn {
  position: absolute;
  top: 0;
  right: 25px;
  font-size: 36px;
  margin-left: 50px;
}

@media screen and (max-height: 450px) {
  .sidenav {padding-top: 15px;}
  .sidenav a {font-size: 18px;}
}
</style>
</head>
<body>
<form action="/dash" method="POST">
<div id="mySidenav" class="sidenav">
  <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&amp;times;</a>
  <a href="Login.html">Home</a>
  <a href="Register.html">Register</a>
  <a href="upload.html">Upload Image</a>
  <a href="#">Food Items</a>
```

```
  <a href="BMI_Calculation.html">BMI Calculation</a>
  <a href="ref.html">Logout</a>
</div>
</form>


<span style="font-size:30px;cursor:pointer" onclick="openNav()">&#9776; Menubar</span>

<script>
function openNav() {
  document.getElementById("mySidenav").style.width = "250px";
}

function closeNav() {
  document.getElementById("mySidenav").style.width = "0";
}
</script>

</body>
</html>
<!DOCTYPE html>
<html>
<head>
    <title>select the file</title>
    <style>
        body{
            background-color: #ffb6c1;
        }
    </style>
</head>
<body>
<form action="/upload" method="POST">
<center>
<label for="myfile">Select a file:</label>
<input type="file" id="myfile" name="myfile" /> <br/><br/>
<input type="submit" value="submit" />
</center>



<P><b>This page helpful for check the nutrition value</b></P>
</form>
</body>
</html>

from tkinter.tix import Meter

import ibm_db
from flask import Flask, redirect, render_template, request, session, url_for
```

```
app=Flask(__name__)

conn=ibm_db.connect('DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-
10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;SECURITY=SSL;SSLServerCertificate
=certi.crt;UID=spy48271;PWD=80QbK1bDsTDY3NOO;',",")

@app.route("/")
def index():
    return render_template("dash.html")



@app.route("/dash",methods=["GET","POST"])
def dash():
    return render_template("Register.html")

@app.route("/Register",methods=["GET","POST"])
def Register():
    if request.method=="POST":
        username=request.form['username']
        password=request.form['password']
        phonenumber=request.form['phonenumber']
        emailid=request.form['emailid']
        sql="Insert INTO REGISTER VALUES(?,?,?,?)"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.bind_param(stmt,3,phonenumber)
        ibm_db.bind_param(stmt,4,emailid)
        ibm_db.execute(stmt)
        return render_template("Login.html")


@app.route("/Login",methods=["GET","POST"])
def Login():
    if request.method=="POST":
        Username=request.form['username']
        Password=request.form['password']
        sql="Insert INTO LOGIN VALUES(?,?)"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,Username)
        ibm_db.bind_param(stmt,2,Password)
        ibm_db.execute(stmt)
        return render_template("upload.html")

@app.route("/upload",methods=["GET","POST"])
def upload():
    if request.method=="POST":
         myfile=request.form['myfile']
         sql="Insert INTO UPLOAD VALUES(?)"
```

```
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,myfile)
        ibm_db.execute(stmt)
        return render_template("BMI_Calculation.html")



@app.route("/BMI_Calculation",methods=["GET","POST"])
def BMI_Calculation():
    if request.method=="POST":
        Height=request.form['Height']
        Weight=request.form['Weight']
        sql="Insert INTO BMI_CALCULATION VALUES(?,?)"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,Height)
        ibm_db.bind_param(stmt,2,Weight)
        ibm_db.execute(stmt)
        return render_template("ref.html")


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



import streamlit as st
from PIL import Image
from keras_preprocessing.image import img_to_array
import numpy as np
from keras.models import load_model
import requests
from bs4 import BeautifulSoup

model = load_model('FV.h5')
labels = {0: 'apple', 1: 'banana', 2: 'beetroot', 3: 'bell pepper', 4: 'cabbage', 5: 'capsicum', 6: 'carrot', 7: 'cauliflower', 8: 'chilli
pepper', 9: 'corn', 10: 'cucumber', 11: 'eggplant', 12: 'garlic', 13: 'ginger', 14: 'grapes', 15: 'jalepeno', 16: 'kiwi', 17: 'lemon',
18: 'lettuce',
         19: 'mango', 20: 'onion', 21: 'orange', 22: 'paprika', 23: 'pear', 24: 'peas', 25: 'pineapple', 26: 'pomegranate', 27:
'potato', 28: 'raddish', 29: 'soy beans', 30: 'spinach', 31: 'sweetcorn', 32: 'sweetpotato', 33: 'tomato', 34: 'turnip', 35:
'watermelon'}

fruits = ['Apple','Banana','Bello Pepper','Chilli
Pepper','Grapes','Jalepeno','Kiwi','Lemon','Mango','Orange','Paprika','Pear','Pineapple','Pomegranate','Watermelon']
vegetables =
['Beetroot','Cabbage','Capsicum','Carrot','Cauliflower','Corn','Cucumber','Eggplant','Ginger','Lettuce','Onion','Peas','Potato','Ra
ddish','Soy Beans','Spinach','Sweetcorn','Sweetpotato','Tomato','Turnip']

def fetch_calories(prediction):
        url = 'https://www.google.com/search?&q=calories in ' + prediction
        req = requests.get(url).text
```

```
        scrap = BeautifulSoup(req, 'html.parser')
        calories = scrap.find("div", class_="BNeawe iBp4i AP7Wnd").text
        return calories

def processed_img(img_path):
    img=load_img(img_path,target_size=(224,224,3))
    img=img_to_array(img)
    img=img/255
    img=np.expand_dims(img,[0])
    answer=model.predict(img)
    y_class = answer.argmax(axis=-1)
    print(y_class)
    y = " ".join(str(x) for x in y_class)
    y = int(y)
    res = labels[y]
    print(res)
    return res.capitalize()

def run():
    st.title("Fruits-Classification")
    img_file = st.file_uploader("Choose an Image", type=["jpg", "png"])
    if img_file is not None:
        img = Image.open(img_file).resize((250,250))
        st.image(img,use_column_width=False)
        save_image_path = './upload_images/'+img_file.name
        with open(save_image_path, "wb") as f:
            f.write(img_file.getbuffer())

        # if st.button("Predict"):
        if img_file is not None:
            result= processed_img(save_image_path)
            print(result)
            if result in vegetables:
                st.info('**Category : Vegetables**')
            else:
                st.info('**Category : Fruit**')
            st.success("**Predicted : "+result+'**')
            cal = fetch_calories(result)
            if cal:
                st.warning('**'+cal+'(100 grams)**')


run()
import requests
from bs4 import BeautifulSoup

# def get_weather(place):
url='https://www.google.com/search?&q=calories in '+'dal makhani'
req=requests.get(url).text
print(req)
```

```
scrap=Beautifulsoup(req,'html.parser')
tmp = scrap.find("div", class_= "BNeawe iBp4i AP7Wnd").text
print(tmp)
<!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">
   <title>Document</title>
   <style>
      body{
        background-image: url("image.jpg");
        background-repeat: no-repeat;
        background-image: cover;
      }
   </style>
</head>
<body>
   <form action="/BMI_Calculation" method="POST">
   <div class="calculator-container">
      <center>
      <h1>BMI CALCULATOR</h1>
      <label>Height:</label>
      <input class="Height-input-field" type="text">
      <br></br>
      <label>Weight:</label>
      <input class="Weight-input-field" type="text"><br>
      <button class="calculate"> Calculate</button>
   </div>
   <h3 class="result"></h3>
   <p class="result-statement"></p>
   <script src="script.js"></script>
    </center>
   </form>
</body>
</html>
```

GITHUB LINK
https://github.com/IBM-EPBL/IBM-Project-15641-1659602222

PROJECT DEMO LINK
https://www.kapwing.com/videos/6377d4e47ac50400178c157e