NUTRITION ASSISTANT APPLICATON

IBM - NALAIYA THIRAN PROJECT BASED LEARNING

ON

PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

A PROJECT REPORT

INTERNAL MENTOR: Dr. P. K. POONGUZHALI

INDUSTRY MENTOR: PRIYA DHARSHINI

SUBHIKSHA S	19106118
RIFHATH RIZAN	19106096
SARVESH V	19106101
VIJAYSHREERAM S R	19106131

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

HINDUSTHAN COLLEGE OF ENGINEERING AND TECHOLOGY

Approved by AICTE, New Delhi, Accredited with ‘A’ Grade by NAAC

(An Autonomous Institution, Affiliated to Anna University, Chennai)

COIMBATORE – 641 032

ABSTRACT

Many people have become aware of their health. Moreover, they are also informed how to live a healthy lifestyle. Many studies discuss dietary self-monitoring, and the research on this topic increased in 2017 and started to include mobile applications. Based on the analysis that was carried out using VOS viewer and NVivo, the results of clusters and trending topics from the study were obtained. The researchers chose two cluster themes related to this research: attitudes to improved dietary behavior and mobile health applications.

Most of the research related to these themes aims to identify changes in healthy lifestyle behavior with mobile applications that are considered effective in dietary self-monitoring. Furthermore, recent studies have shown that, in the face of this research, researchers experience found self-monitoring dietary and physical activity, healthy diet and lifestyle, and mobile application of dietary self-monitoring. Based on the results of this research, the authors recommend, for future research, the development of a nutrition mobile application that helps people self-monitor their diet based on their lifestyle behaviour.

TABLE OF CONTENTS

NUTRITION ASSISTANT APPLICATION

1. **INTRODUCTION**

PROJECT OVERVIEW:

Many people today are conscious of their health and the value of leading healthy lives. Nutritional education is crucial for encouraging healthy eating practices since it guarantees that nutrient needs are met to prevent malnutrition. People's dietary behaviours and food preferences are likely to improve by being exposed to education-based interventions, such as increased nutrition knowledge. People who are aware of the connection between certain health problems and inadequate nutrition are better able to monitor and control their weight through diet.

The analysis of typical food intake is a significant difficulty in human nutrition. This is of particular significance in light of current recommendations for eHealth-tailored therapies. Because mobile phones may be used for digitizing dietary measurements and delivering feedback, they have presented an opportunity for measuring and increasing nutrient consumption. Hundreds of nutrition-related smartphone applications have been released and downloaded by millions of users in recent years

Because of the advancement of technology, our age has become sedentary. The cost of app development technology has nearly reduced the amount of physical effort, which is the underlying cause of many difficulties. These statistics suggest that we are living in terrible times, and the Health & Fitness agenda is in serious need of a boost. The Nutrition Assistant app is a lifesaver for such people. They help consumers maintain a healthy diet and keep a close eye on their calorie consumption.

2. PURPOSE:

A nutrition assistant application is a type of nutrition tracking app that assists users in losing weight, becoming healthier, and becoming stronger. Nutrition apps include calorie counters, diet trackers, nutrition planner apps, and

marketplace platforms that connect users and nutrition coaches.

There are also apps built for certain niches, such as app-based food diaries, prenatal nutrition apps, bodybuilder nutrition apps, vegan nutrition apps, diet-tracking apps, health activity tracker apps, and so on. Our goal is to examine the key elements of the most popular nutrition apps and compare their dietary assessment and user feedback methodologies and technology.

2. LITERATURE SURVEY

2.1. EXISTING PROBLEMS:

Non Communicable illnesses, such as diabetes and cardiovascular disease, are responsible for about two-thirds of all fatalities worldwide. The fundamental advice for combating this epidemic revolves upon lifestyle modifications, namely supporting healthy foods, physical activity (PA), and reducing cigarette and alcohol intake. Nutritional intervention requires accurate food intake data. Food intake data collection methods may be characterized in several ways. Retrospective approaches, such as the 24-hour meal recall and the food frequency questionnaire (FFQ), need memory for remembering of items eaten based on the time of collection.

In contrast, prospective techniques necessitate diet reporting as intake happens, effectively functioning as food diaries. Prospective approaches are often used in clinical nutrition between 4 and 7 days. The approaches might also be classified as quantitative daily consumption or food frequencies. The first group concentrates on precisely tracking detailed food consumption over a period of many days. The latter evaluates normal consumption habits over time. These approaches have typically been supplied using paper and pen, however there is a burden associated with this system for both patients and health providers. Patients favor digitization of food diaries since it saves time and resource.

2. REFERENCES:

[1] Caitlyn G. Edwards, PhD, Pejman Sajjadi, PhD Alex Fatemi, MS Erica N. Krieger, BS Alexander Klippel, PhD Travis D. Masterson, PhD
URL: https://doi.org/10.1016/j.jneb.2022.02.014

[2] Alita Rushton BHlthSc (Nutrition),Anna Edwards MDietSt, APD,Judith Bauer PhD,Jack J. Bell PhD (2021) Dietitian assistant opportunities within the nutrition care process for patients with or at risk of malnutrition: a systematic review URL: https://doi.org/10.1111/1747-0080.12651

[3] Carvalho, M., Kotian, P., George, H., Pawade, D., Dalvi, A., Siddavatam, I. (2021). Implementation of Smart Diet Assistance Application. In: Mahapatra, R.P., Panigrahi, B.K., Kaushik, B.K., Roy, S. (eds) Proceedings of 6th International Conference on Recent Trends in Computing. Lecture Notes in Networks and Systems, vol 177. Springer, Singapore. URL: https://doi.org/10.1007/978-981-33-4501-0_30

[4] Harsh Athavale, Pradnya Vaity, Fauziya Khan, Prof. Atul Shintre (2022) International Journal of Research Publication and Reviews URL: https://ijrpr.com/uploads/V3ISSUE4/IJRPR3571.pdf

[5] Fiteni, D. B. (2021). Application of hybrid learning interventions in advancing food and nutrition pedagogy in UAE and beyond through Culinary Science to sustain human health and wellbeing. International Journal of Home Economics, 14(1), 16–38.
URL: https://search.informit.org/doi/10.3316/informit.990845530758092

[6] Salma Alabdulwahed, Natalia Galán-López, Tom Hill, Lewis J. James, Bryna Catherine Rose Chrismas, Sebastien Racinais, Trent Stellingwerff, Diogo V. Leal, Matheus Hausen, Karim Chamari, Hugh H.K. Fullagar, Christopher Esh, and Lee Taylor (2022) Heat Adaptation and Nutrition Practices: Athlete and Practitioner Knowledge and Use.
URL: https://doi.org/10.1123/ijspp.2021-0462

2.3. PROBLEM STATEMENT DEFINITION:

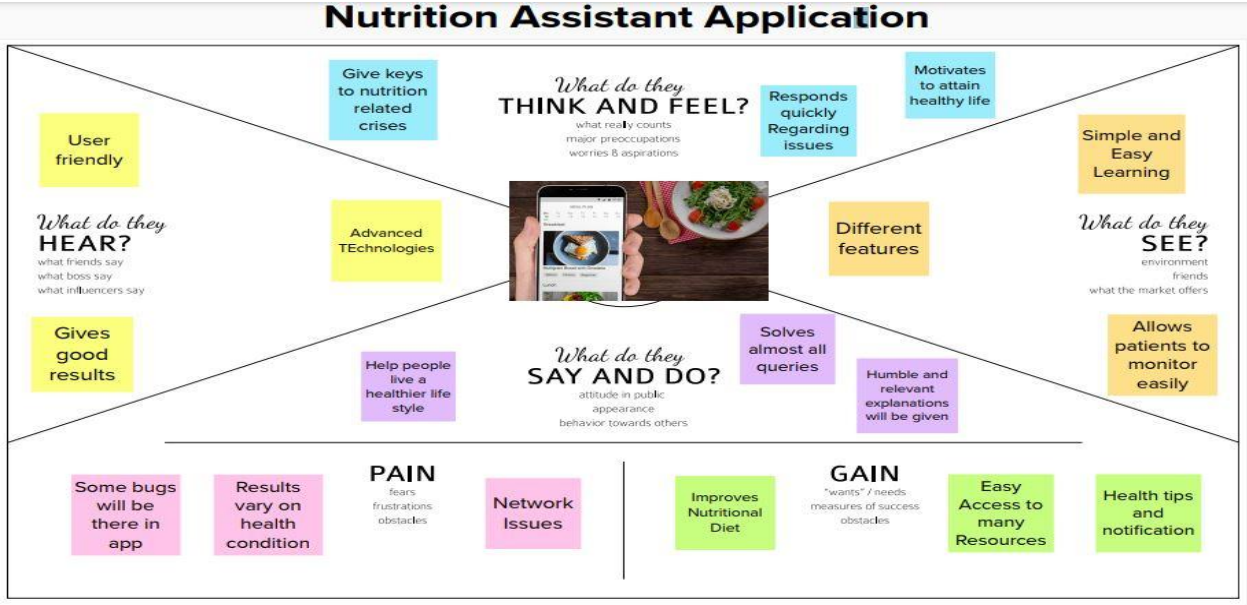
- This Nutrition assistant app is based on nutrients and calories of the food will help people with providing proper nutrition and helps in maintaining a healthy lifestyle.

- Instead of using many different apps to keep touch with people this one software handle everything such as meal planning diet analysis communication between client and nutritionists, workout plans, questionnaires and nutrition coaching for clients.

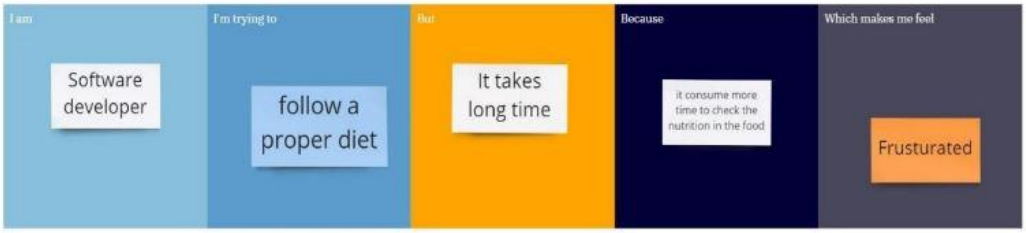
- Further this will help you to track their progress keep a food journal track their water intake.

3.IDEATION AND PROPOSED SOLUTION

3.1. Empathy Map Canvas:

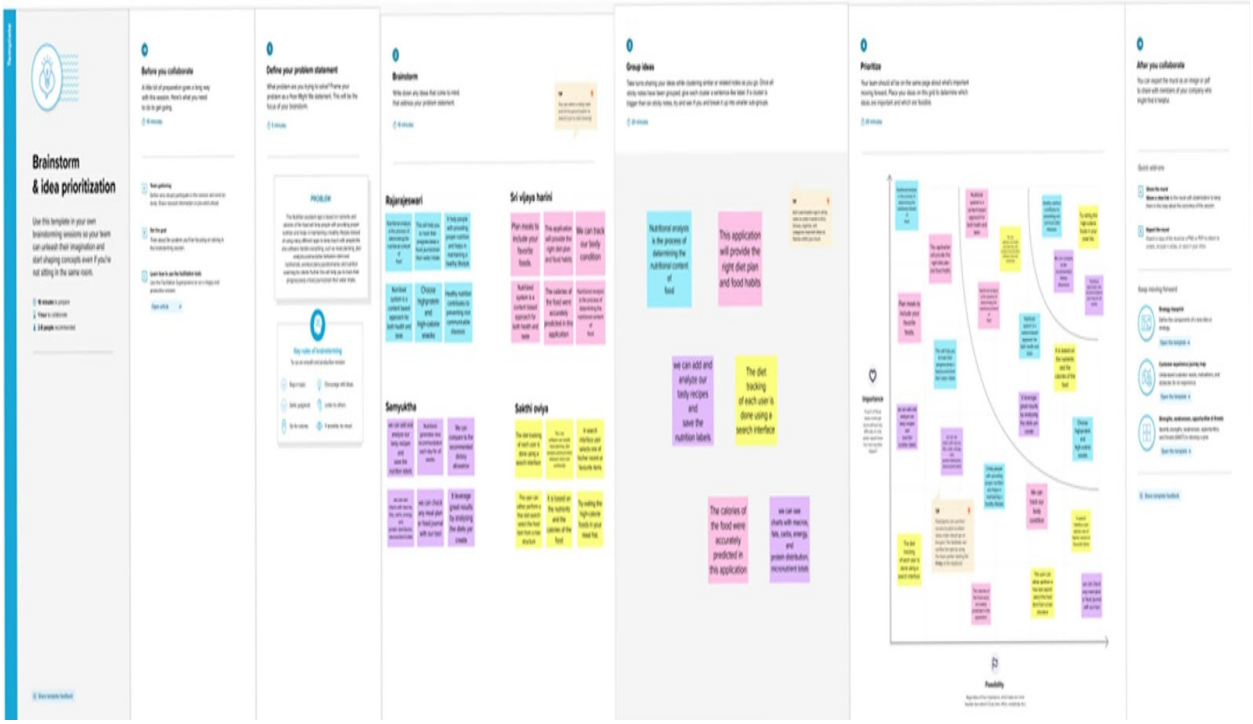


1.Ideation :



Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Entrepreneur	follow up a proper diet	I couldn't	of my cravings for fast food	sad
PS-2	Working Woman	follow a proper diet	I Can't	Because changes in working Time	stress

3.2.2. Brainstorming:

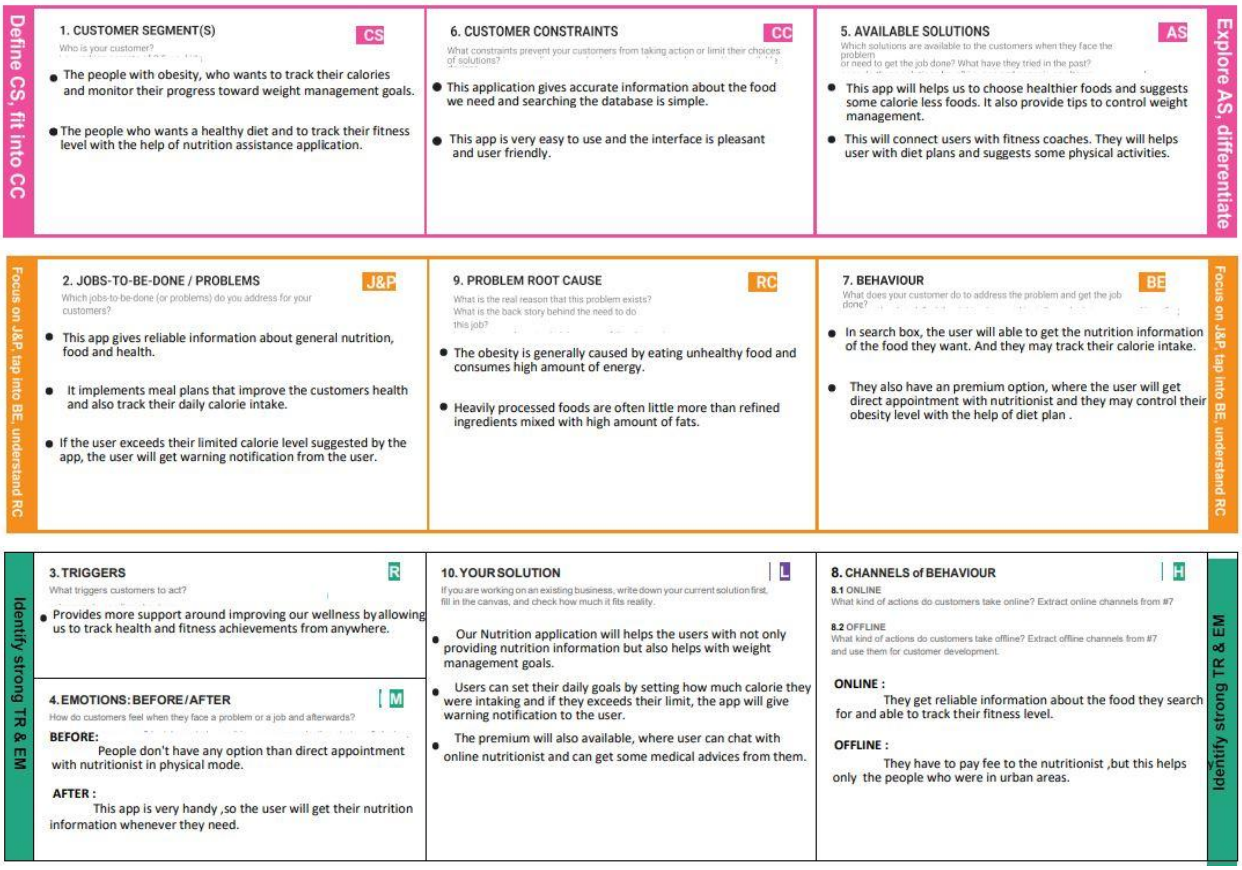


3.3. Proposed Solution:

S.No.	Parameter	Description
1.	ProblemStatement(Problem to be solved)	● This Nutrition assistant app is based on nutrients and calories of the food will help people with providing proper nutrition and helps in maintaining a healthy lifestyle. ● Instead of using many different apps to keep touch with people this one software handle everything such as meal planning diet analysis communication between client and nutritionists, workout plans, questionnaires and nutrition coaching for clients. ● Further this will help you to track their progress keep a food journal track their water intake.
2.	Idea/Solutiondescription	● By creating an application , we can recommended diet plans for the users and measures sugar level.
3.	Novelty/Uniqueness	● I can realize real time images of meal and analyze it for nutritional content can be handy and improve dietary habit..

4.	SocialImpact/CustomerSatisfaction	• It helps to maintain with providing proper nutrition and healthy lifestyle for normal people
5.	BusinessModel(RevenueModel)	• Social Media is to best way to Develop our application.
6.	ScalabilityoftheSolution	• Good Relationship • Easily access to the Application. • Different diet charts can be planned f or different aspecs of people.

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.
FR-2	User Confirmatin	Confirmation via OTP.
FR-3	Uploading Image	The system should able to get the image from the user.
FR-4	Identification of image	The system should able to identify the image of the food given using model. .
FR-5	Obtain the ingredients	The system must able to obtain the ingredients of the given food image.

FR-6	Display the nutritional value	The system must able to display the nutritional value of the food with the help of nutritional Application.

4.2 Non-functional Requirements:

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

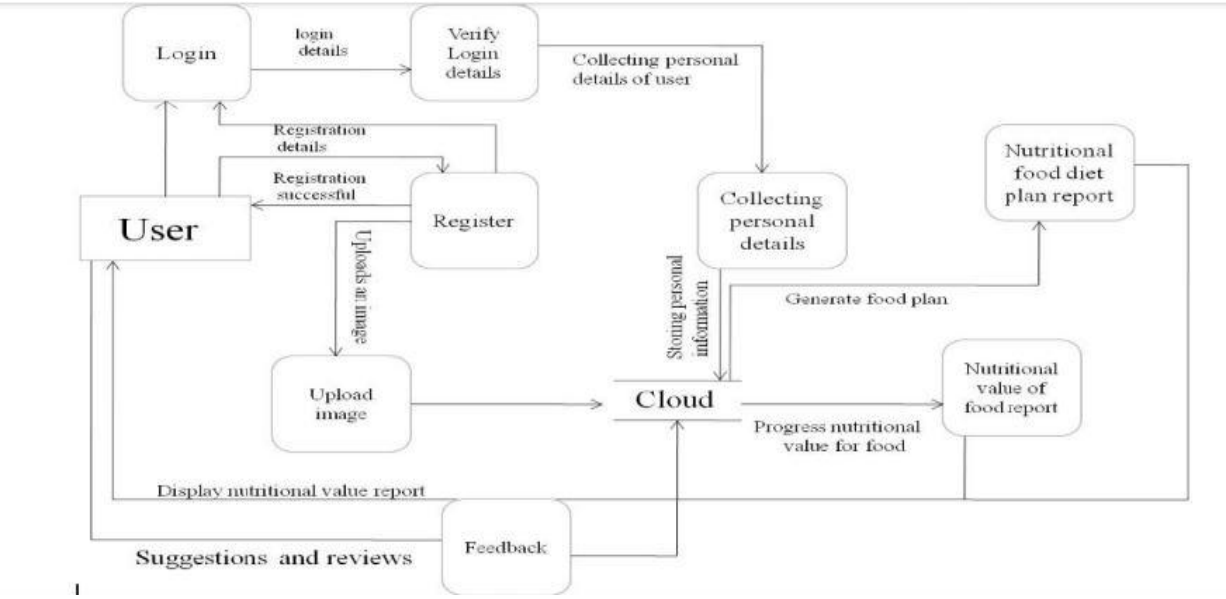
FR No	Non-FunctionalRequirement	Description
NFR-1	Usability	Only registered user is allowed to using the application.
NFR-2	Security	Authentication of user is done for security purpose.
NFR-3	Reliability	The user gets the standardized nutritional value of the food items.
NFR-4	Performance	User satisfaction is ensured by getting their feedback .
NFR-5	Availability	This application can be used by the user when they are in online Mode.

NFR-6	Scalability	This application can be used in all operating system and it can handle quite large Quantity of users too.

5. PROJECT DESIGN

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.



1. .2. Technology Stack:

Table-1: Components & Technologies:

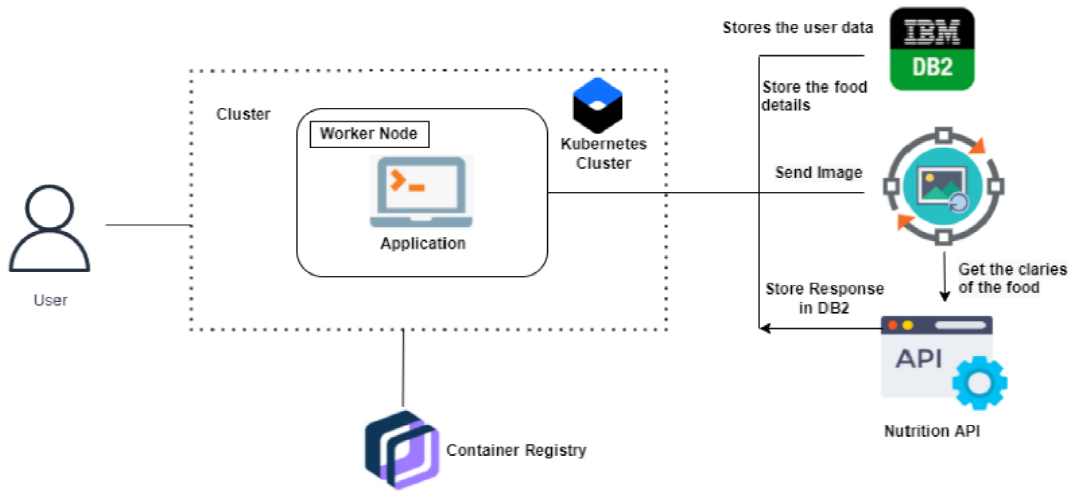
S.No	Component	Description	Technology
1	User Interface	user interacts with application	HTML, CSS, JavaScript, React Js etc.
2	Database	Data Type, Configurations etc.	MySQL,javascript,python ,flask
3	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
4	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
5	External API-1	To predict the image that user will upload in the upload image page	Clarifai’s AI-driven Food detection Model API
6	External API-2	Food API’s for to the nutritional value for the identified food	Food API
7	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, Docker.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	open-source frameworks used	SendGrid, Python
2	Security Implementations	Request authentication using encryption	.Encryptions
3	Scalable Architecture	The scalability of architecture consists of 3 tiers	Web Server – HTML, CSS, JavaScript Application Server – Python Flask Database Server – IBM Cloud
4	Availability	Availability is increased by loads balancers in cloud VPS	working to reduce the severity and likelihood of problems, closely monitoring applications and infrastructure, keeping technical debt in check, automating recovering mechanisms, and regularly putting those recovery mechanisms to the test.
5	Performance	The application is expected to handle up to 4000 predictions per second	Optimize image sizes, use a content delivery network, use website caching and adopt cloud based website

			monitoring

2. SOLUTION AND TECHNICAL ARCHITECTURE



3. **USER STORIES:**

User Type	Functional requirement(Epic)	User Story number	User Story/Task	Acceptance criteria	Priority	Release

Customer(Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint1
	Login	USN-3	As a user, I can log into the application by entering email & password	I can login when password and email are correct	High	Sprint1

	Collecting personal details	USN-4	As a user,I can provide a personal information for processing	I can enter the personal details	Medium	Sprint 1
	Upload image	USN-5	As a user,I can upload an image for the processi ng of food.	I can upload a food image.	High	Sprint1
	Feedback	USN-6	As a user,I can give feedback	I can give feedback about the application	Low	Sprint1
Cloud	Nutritional value of report	USN-7	In cloud the food image is processed and provides the nutritional value of food.	It gives the nutritional value of food.	High	Sprint2

	Nutritional food diet plan report	USN-8	In cloud the food diet plan based on nutritional value is generated based on the personal information provided by the user.	It provides the diet nutritional plan	Medium	Sprint2

6. PROJECT PLANNING & SCHEDULING

1. Sprint Planning and Estimation:\

S. NO	MILESTONES	ACTIVITIES	DATE
1.	Preparation Phase	Pre-requisites	24Aug2022
		Prior Knowledge	25Aug2022
		Project Structure	23Aug2022
		Project Flow	23Aug2022
		Project Objectives	22Aug2022
		Registrations	26Aug2022
		Environment Set-up	27Aug2022

2.	Ideation Phase	Literature Survey	29Aug2022–03Sept2022
		Empathy Map	5Sept 2022-7Sept 2022
		ProblemStatement	8Sept 2022-10Sept 2022
		Ideation	12Sept 2022–16Sept
3.	Project Design Phase - 1	Proposed Solution	19 Sept 2022 – 23 Sept 2022
		Problem Solution Fit	24 Sept 2022 – 26 Sept 2022
		Solution Architecture	27 Sept 2022 – 30 Sept 2022
4.	Project Design Phase - 2	Customer Journey Map	03 Oct 2022 – 08 Oct 2022
		Requirement Analysis	09 Oct 2022 – 11 Oct 2022
		Data Flow Diagrams	11 Oct 2022 – 14 Oct 2022
		Technology	15 Oct 2022 -

		Architecture	16 Oct 2022
5.	Project Planning Phase	Milestones & Tasks	17 Oct 2022 – 18 Oct 2022
		Sprint Schedules	19 Oct 2022 – 22 Oct 2022
6.	Project Development Phase	Sprint - 1	26 Oct 2022 – 31 Oct 2022
		Sprint – 2	01 Nov 2022 – 07 Nov 2022
		Sprint – 3	08 Nov 2022 – 13 Nov 2022
		Sprint – 4	15 Nov 2022 – 20 Nov 2022

1.Sprint Delivery Schedule:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	8	High	Subhiksha, Rifhath
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	8	High	Sarvesh, Vijayshreeram
Sprint-1	Validating user	USN-3	Checking whether new user or existing user of the application	4	Medium	Subhiksha, Rifhath
Sprint-2	Add nutrtion diet	USN-4	As a user, I can add the day-to-day nutrtion dietto the application	8	High	Sarvesh, Vijayshreeram

Sprint-2	Edit and Delete nutrtion diet	USN-5	As a user, I can edit and delete the previously created nutrtion diet	8	High	Subhiksha, Rifhath
Sprint-2	Creating time- based filters in history.	USN-6	As a user, I can see the time-based history of nutrtion diet.	4	Medium	Sarvesh, Vijayshreeram
Sprint-3	ntegrating with pie charts for analysis	USN-7	As a user, I can view diagrammatic representation of nutrtion diet	8	High	Subhiksha, Rifhath
Sprint-3	Enabling limit feature	USN-8	As a user, I can set monthly limit to nutrtion diet	4	Medium	Sarvesh, Vijayshreeram
Sprint-3	Sending Email Alerts	USN-9	As a user, I will receive a mail if I cross a limit	8	High	Subhiksha, Rifhath
Sprint-4	Testing	USN-9	Testing the application with various tools	10	High	Sarvesh, Vijayshreeram
Sprint-4	Deployment	USN-9	Deployment of the application	10	High	Subhiksha, Rifhath

6.3. Reports from JIRA:

PROJECT TRACKER:

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	24 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	31 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	07 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

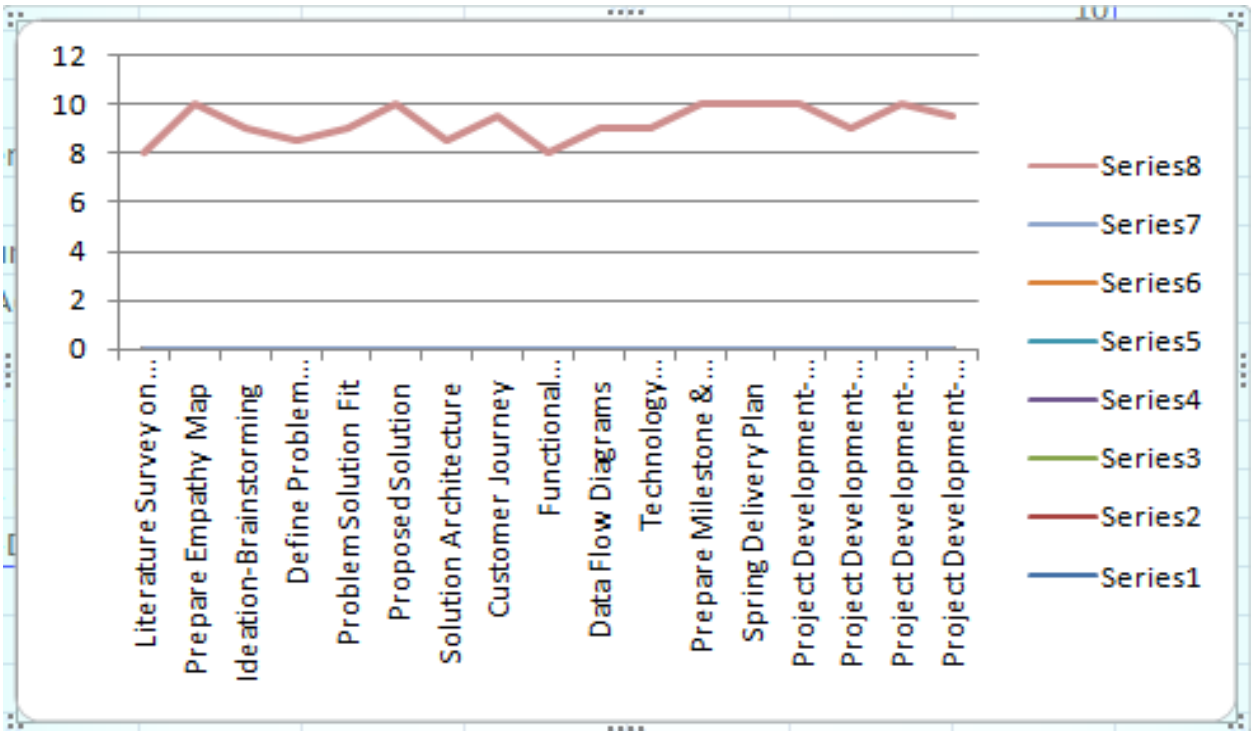
Velocity:

Imagine we have a10 days print duration ,and the velocity of the team is 20 (points per sprint) .Let's calculate the team average velocity (AV) per iteration unit(story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time However, burndown charts can be applied to any project containing measurable progress overtime.



Reference:

https://www.atlassian.com/agile/project-management

https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software

https://www.atlassian.com/agile/tutorials/epics

https://www.atlassian.com/agile/tutorials/sprints

https://www.atlassian.com/agile/project-management/estimation

https://www.atlassian.com/agile/tutorials/burndown-charts

7. CODING AND SOLUTIONING

Feature 1:Login

Step 1: Create a login page using the python functionality and save as Login.py

```
from tkinter import *
from tkinter import messagebox
import ast

root=Tk()
root.title('Login Page')
root.geometry('1540x1080')
root.configure(bg='#fff')
root.resizable(True,True)

def signin():
    username=user.get()
    password=code.get()

    file=open('datasheet.txt', 'r')
    d=file.read()
    r=ast.literal_eval(d)
    file.close()

    #print(r.keys())
    #print(r.values())

    if username in r.keys() and password==r[username]:
        screen=Toplevel(root)
        screen.title("Nutrition Assistant")
        screen.geometry('1540x1080')
        screen.config(bg="white")
```

```
            Label(screen,text='Welcome to the Nutrition
            Assistant!',bg='white',font=('Cascadia Mono
            SemiBold',50,'bold')).pack(expand=True)
            screen.mainloop()

        elif username!='admin' and password!='1234':
             messagebox.showerror("Invalid", "Invalid Credentials")

        elif username!='admin':
            messagebox.showerror("Invalid", "Invalid UserName")
        elif password!='1234':
            messagebox.showerror("Invalid", "Invalid PassWord")
        else:
            messagebox.showerror("Invalid", "Invalid Credentials")

    #####----------------------------------------------------call sign up page-----------
-----------------------------------
    def signup():
        window=Toplevel(root)

    #-------------------------------------------------------code of signup.py-------------
-----------------------------------
    #-----------------------------------------------------------------------------------------
-----------------------------
        window.title("Registration Page")
        window.geometry('1540x1080')
        window.configure(bg='#fff')
        window.resizable(True,True)

        def signup():
            username=user.get()
            password=code.get()
            confirmpassword=c_code.get()
```

```
    if password==confirmpassword:
        try:
            file=open('datasheet.txt', 'r+')
            d=file.read()
            r=ast.literal_eval(d)

            dict2={username:password}
            r.update(dict2)
            file.truncate(0)
            file.close()

            file=open('datasheet.txt','w')
            w=file.write(str(r))

            messagebox.showinfo('Sign Up','Successfully Signed up')
            window.destroy()

        except:
            file=open('datasheet.txt','w')
            pp=str({'Username':'password'})
            file.write(pp)
            file.close()

    else:
        messagebox.showerror('Invalid','Both Password should match')

def Login():
    window.destroy()


img=PhotoImage(file='restaurant.png')
Label(window,image=img,border=0,bg='white').place(x=50,y=60)

frame=Frame(window,width=450,height=680,bg='white')
frame.place(x=970,y=50)
```

```
        heading = Label(frame,text="Sign Up", fg="#57a1f8", bg='white',
        font=('Cascadia Mono SemiBold',30,'bold'))
        heading.place(x=155,y=5)

    #-----------------------------------------Username---------------------------------
-----------------------------------

        def on_enter(e):
            user.delete(0,'end')
        def on_leave(e):
            if user.get()==":
                user.insert(0,'User Name')

        user =
        Entry(frame,width=28,fg='black',border=0,bg='white',font=('Times New
        Roman',15))
        user.place (x=88,y=80)
        user.insert(0, 'User Name')
        user.bind("<FocusIn>", on_enter)
        user.bind("<FocusOut>", on_leave)

        Frame(frame,width=295,height=1,bg='black').place(x=85,y=105)

    #-----------------------------------------------------password-------------------
-----------------------------------

        def on_enter(e):
            code.delete(0,'end')
        def on_leave(e):
            if code.get()==":
                code.insert(0,'Password')

        code =
```

```
        Entry(frame,width=28,fg='black',border=0,bg='white',font=('Times New
        Roman',15))
        code.place (x=88,y=160)
        code.insert(0, 'Password')
        code.bind("<FocusIn>", on_enter)
        code.bind("<FocusOut>", on_leave)

        Frame(frame,width=295,height=1,bg='black').place(x=85,y=185)

    #-----------------------------------Confirm password-------------------------
-----------------------------------

        def on_enter(e):
            c_code.delete(0,'end')
        def on_leave(e):
            if c_code.get()=='':
                c_code.insert(0,'Confirm Password')

        c_code =
        Entry(frame,width=28,fg='black',border=0,bg='white',font=('Times New
        Roman',15))
        c_code.place (x=88,y=240)
        c_code.insert(0, 'Confirm Password')
        c_code.bind("<FocusIn>", on_enter)
        c_code.bind("<FocusOut>", on_leave)

        Frame(frame,width=295,height=1,bg='black').place(x=85,y=265)

    #-----------------------------------------signin Button-------------------------
-----------------------------------------

        Button(frame,width=50,pady=10,text='Sign
        Up',bg='#57a1f8',fg='white',border=0,command=signup).place(x=60,y=
        300)
```

```
        label=Label(frame,text='I already have an account
        ?',fg='black',bg='white',font=('Times New Roman',13))
        label.place(x=100,y=360)


    signin=Button(frame,width=5,text='Login',border=0,bg='white',cursor='ha
    nd2',fg='#47a1f8',command=Login)
        signin.place(x=290,y=360)



        window.mainloop()
    #-------------------------------------------------------------------------------------------
-----------------------------------
    #-------------------------------------------------------------------------------------------
-----------------------------------
    img=PhotoImage(file='restaurant.png')
    Label(root,image=img,border=0,bg='white').place(x=50,y=60)

    frame=Frame(root,width=450,height=680,bg='white')
    frame.place(x=970,y=50)

    heading = Label(frame,text="Login", fg="#57a1f8", bg='white',
    font=('Cascadia Mono SemiBold',30,'bold'))
    heading.place(x=155,y=5)

    #----------------------------------------Username---------------------------------
-----------------------------------

    def on_enter(e):
        user.delete(0,'end')

    def on_leave(e):
        if user.get()=='':
            user.insert(0,'User Name')
```

```
        user = Entry(frame,width=28,fg='black',border=0,bg='white',font=('Times
        New Roman',15))
        user.place (x=88,y=80)
        user.insert(0, 'User Name')
        user.bind("<FocusIn>", on_enter)
        user.bind("<FocusOut>", on_leave)

        Frame(frame,width=295,height=1,bg='black').place(x=85,y=105)

        #----------------------------------------------------password----------------------
---------------------------------

        def on_enter(e):
            code.delete(0,'end')

        def on_leave(e):
            if code.get()=='':
                code.insert(0,'Password')

        code = Entry(frame,width=28,fg='black',border=0,bg='white',font=('Times
        New Roman',15))
        code.place (x=88,y=160)
        code.insert(0, 'Password')
        code.bind("<FocusIn>", on_enter)
        code.bind("<FocusOut>", on_leave)

        Frame(frame,width=295,height=1,bg='black').place(x=85,y=185)

        #----------------------------------------signin Button----------------------------
----------------------------------------

        Button(frame,width=50,pady=10,text=' Login
        ',bg='#57a1f8',fg='white',border=0, command=signin).place(x=60,y=300)
        label=Label(frame,text="Don't have an account
```

```
?",fg='black',bg='white',font=('Times New Roman',13))
label.place(x=100,y=360)

signup=Button(frame,width=5,text='Sign
Up',border=0,bg='white',cursor='hand2',fg='#47a1f8', command=signup)
signup.place(x=290,y=360)



root.mainloop()
```

Feature 2:Signup

Step 2: Create a signup page.

```
import code
from re import L
from tkinter import *
from tkinter import messagebox
import ast

window=Tk()
window.title("Registration Page")
window.geometry('1540x1080')
window.configure(bg='#fff')
window.resizable(True,True)

def signup():
    username=user.get()
    password=code.get()
    confirmpassword=c_code.get()

    if password==confirmpassword:
        try:
```

```
            file=open('datasheet.txt', 'r+')
            d=file.read()
            r=ast.literal_eval(d)

            dict2={username:password}
            r.update(dict2)
            file.truncate(0)
            file.close()

            file=open('datasheet.txt','w')
            w=file.write(str(r))
            messagebox.showinfo('Sign Up','Successfully Signed up')

        except:
            file=open('datasheet.txt','w')
            pp=str({'Username':'password'})
            file.write(pp)
            file.close()

    else:
        messagebox.showerror('Invalid','Both Password should match')

def Login():
    window.destroy()

img=PhotoImage(file='restaurant.png')
Label(window,image=img,border=0,bg='white').place(x=50,y=60)

frame=Frame(window,width=450,height=680,bg='white')
frame.place(x=970,y=50)

heading = Label(frame,text="Sign Up", fg="#57a1f8", bg='white',
font=('Cascadia Mono SemiBold',30,'bold'))
heading.place(x=155,y=5)
```

```
    #-----------------------------------------Username-----------------------------------
-----------------------------------

    def on_enter(e):
       user.delete(0,'end')
    def on_leave(e):
       if user.get()=='':
          user.insert(0,'User Name')

    user = Entry(frame,width=28,fg='black',border=0,bg='white',font=('Times
    New Roman',15))
    user.place (x=88,y=80)
    user.insert(0, 'User Name')
    user.bind("<FocusIn>", on_enter)
    user.bind("<FocusOut>", on_leave)

    Frame(frame,width=295,height=1,bg='black').place(x=85,y=105)

    #-----------------------------------------------------password----------------------
-----------------------------------
    def on_enter(e):
       code.delete(0,'end')
    def on_leave(e):
       if code.get()=='':
          code.insert(0,'Password')

    code = Entry(frame,width=28,fg='black',border=0,bg='white',font=('Times
    New Roman',15))
    code.place (x=88,y=160)
    code.insert(0, 'Password')
    code.bind("<FocusIn>", on_enter)
    code.bind("<FocusOut>", on_leave)

    Frame(frame,width=295,height=1,bg='black').place(x=85,y=185)
```

```
    #-----------------------------------Confirm password-------------------------
---------------------------------

    def on_enter(e):
        c_code.delete(0,'end')
    def on_leave(e):
        if c_code.get()=='':
            c_code.insert(0,'Confirm Password')

    c_code =
    Entry(frame,width=28,fg='black',border=0,bg='white',font=('Times New
    Roman',15))
    c_code.place (x=88,y=240)
    c_code.insert(0, 'Confirm Password')
    c_code.bind("<FocusIn>", on_enter)
    c_code.bind("<FocusOut>", on_leave)

    Frame(frame,width=295,height=1,bg='black').place(x=85,y=265)

    #-----------------------------------------signin Button-------------------------
---------------------------------------

    Button(frame,width=50,pady=10,text='Sign
    Up',bg='#57a1f8',fg='white',border=0,command=signup).place(x=60,y=30
    0)
    label=Label(frame,text='I already have an account
    ?',fg='black',bg='white',font=('Times New Roman',13))
    label.place(x=100,y=360)

    signin=Button(frame,width=5,text='Login',border=0,bg='white',cursor='ha
    nd2',fg='#47a1f8',command=Login)
    signin.place(x=290,y=360)
    window.mainloop()
```

DATA BASE SCHEMA:

Create a Data set and save with .csv extension.

Food	Measure	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs	Category
Cows' milk	1 qt.	976	660	32	40	36	0	48	Dairy products
Milk skim	1 qt.	984	360	36	t	t	0	52	Dairy products
Buttermilk	1 cup	246	127	9	5	4	0	13	Dairy products
Evaporated, undiluted	1 cup	252	345	16	20	18	0	24	Dairy products
Fortified milk	6 cups	1,419	1,373	89	42	23	1.4	119	Dairy products
Powdered milk	1 cup	103	515	27	28	24	0	39	Dairy products
skim, instant	1 1/3 cups	85	290	30	t	t	0	42	Dairy products
skim, non-instant	2/3 cup	85	290	30	t	t	1	42	Dairy products
Goats' milk	1 cup	244	165	8	10	8	0	11	Dairy products
(1/2 cup ice cream)	2 cups	540	690	24	24	22	0	70	Dairy products
Cocoa	1 cup	252	235	8	11	10	0	26	Dairy products
skim. milk	1 cup	250	128	18	4	3	1	13	Dairy products
(cornstarch)	1 cup	248	275	9	10	9	0	40	Dairy products
Custard	1 cup	248	285	13	14	11	0	28	Dairy products
Ice cream	1 cup	188	300	6	18	16	0	29	Dairy products
Ice milk	1 cup	190	275	9	10	9	0	32	Dairy products
Cream or half-and-half	1/2 cup	120	170	4	15	13	0	5	Dairy products
or whipping	1/2 cup	119	430	2	44	27	1	3	Dairy products
Cheese	1 cup	225	240	30	11	10	0	6	Dairy products
uncreamed	1 cup	225	195	38	t	t	0	6	Dairy products
Cheddar	1-in. cube	17	70	4	6	5	0	t	Dairy products
Cheddar, grated cup	1/2 cup	56	226	14	19	17	0	1	Dairy products
Cream cheese	1 oz.	28	105	2	11	10	0	1	Dairy products
Processed cheese	1 oz.	28	105	7	9	8	0	t	Dairy products
Roquefort type	1 oz.	28	105	6	9	8	0	t	Dairy products
Swiss	1 oz.	28	105	7	8	7	0	t	Dairy products
Eggs raw	2	100	150	12	12	10	0	t	Dairy products
Eggs Scrambled or fried	2	128	220	13	16	14	0	1	Dairy products
Yolks	2	34	120	6	10	8	0	t	Fats, Oils, Shortenings
Butter	1T.	14	100	t	11	10	0	t	Fats, Oils, Shortenings
Butter	1/2 cup	112	113	114	115	116	117	118	Fats, Oils, Shortenings
Butter	1/4 lb.	112	113	114	115	116	117	118	Fats, Oils, Shortenings
Hydrogenated cooking fat	1/2 cup	100	665	0	100	88	0	0	Fats, Oils, Shortenings
Lard	1/2 cup	110	992	0	110	92	0	0	Fats, Oils, Shortenings
Margarine	1/2 cup	112	806	t	91	76	0	t	Fats, Oils, Shortenings
Margarine, 2 pat or	1 T.	14	100	t	11	9	0	t	Fats, Oils, Shortenings
Mayonnaise	1 T.	15	110	t	12	5	0	t	Fats, Oils, Shortenings
Corn oil	1 T.	14	125	0	14	5	0	0	Fats, Oils, Shortenings
Olive oil	1T.	14	125	0	14	3	0	0	Fats, Oils, Shortenings
Safflower seed oil	1 T.	14	125	0	14	3	0	0	Fats, Oils, Shortenings
French dressing	1 T.	15	60	t	6	2	0	2	Fats, Oils, Shortenings
Thousand Island sauce	1 T.	15	75	t	8	3	0	1	Fats, Oils, Shortenings
Salt pork	2 oz.	60	470	3	55		0	0	Meat, Poultry
Bacon	2 slices	16	95	4	8	7	0	1	Meat, Poultry
Beef	3 oz.	85	245	23	16	15	0	0	Meat, Poultry
Hamburger	3 oz.	85	245	21	17	15	0	0	Meat, Poultry
Ground lean	3 oz.	85	185	24	10	9	0	0	Meat, Poultry
Roast beef	3 oz.	85	390	16	36	35	0	0	Meat, Poultry
Steak	3 oz.	85	330	20	27	25	0	0	Meat, Poultry

Testing:

Test cases:

1.Login button click with wrong credentils entered.

2.Signup with already registerd email ID.

3.Signup with wrong form data enterd.

4.Entering home page with logged out session.

5.Clicking Homepage buttons with logged out session.

6.Invalid data provided in change password page and request for change in paassword.

USER ACCEPTANCE TESTING:

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Executed By
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	Login page	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	login.py	Login/Signup popup should display	Working as expected	Pass	Rifhath rizan Subhiksha Vijayshreeram Sarvesh

LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup	Signup page	1.Enter URL and click go 2.Click on My Account dropdown 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 e.Last password? Recovery password link	signup.py	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New user	Working as expected	Pass	Rifhath rizan Subhiksha Vijayshreeram Sarvesh

RESULTS:

PERFORMANCE METRICS

1.Planned value: Rs.4000

2.Actual value:Rs.1300

3.Hours worked:45 hours

4.Stick to timelines: 100%

5.Stay within budget: 100%

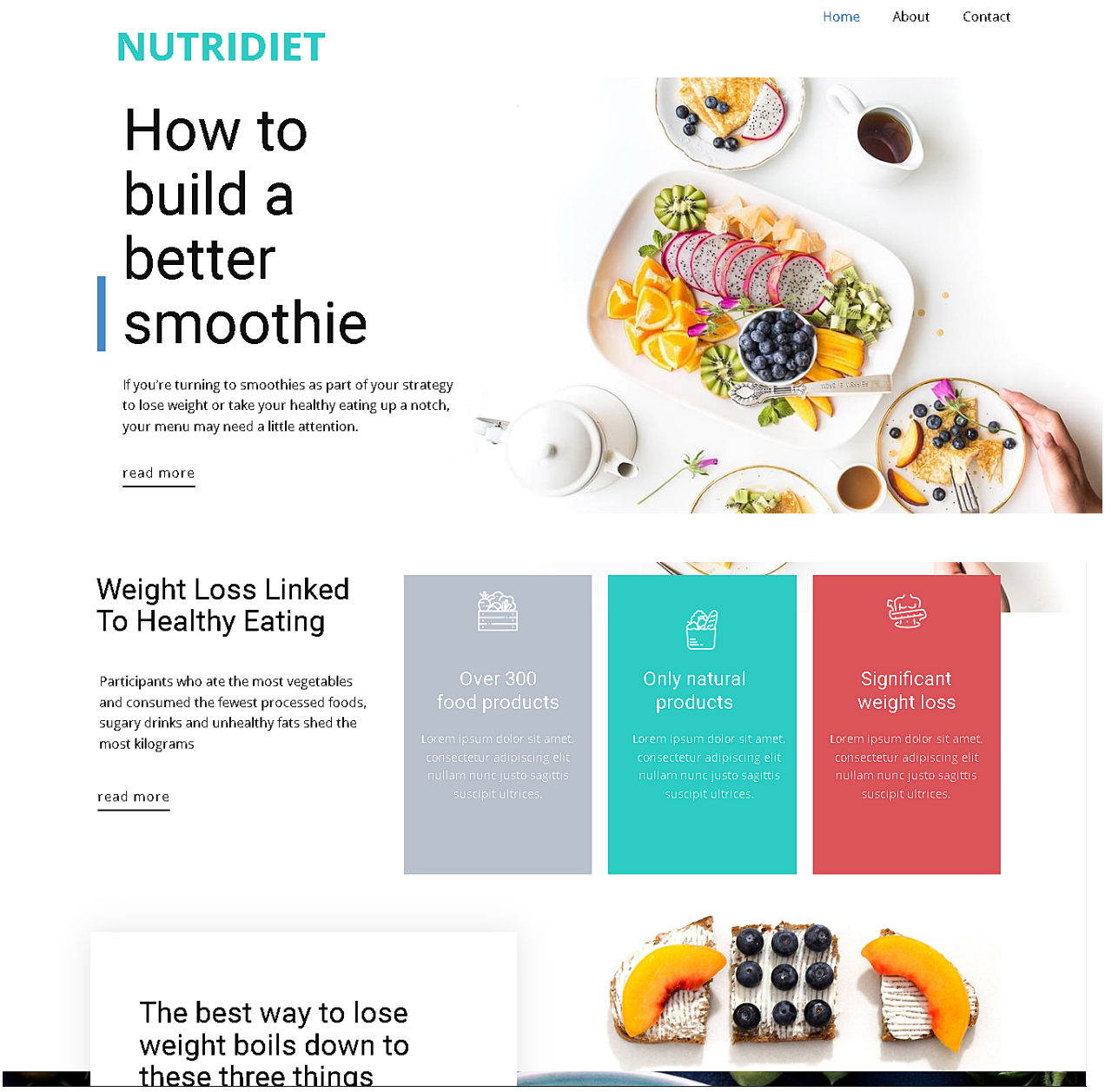
6.Consistency of the product: 75%

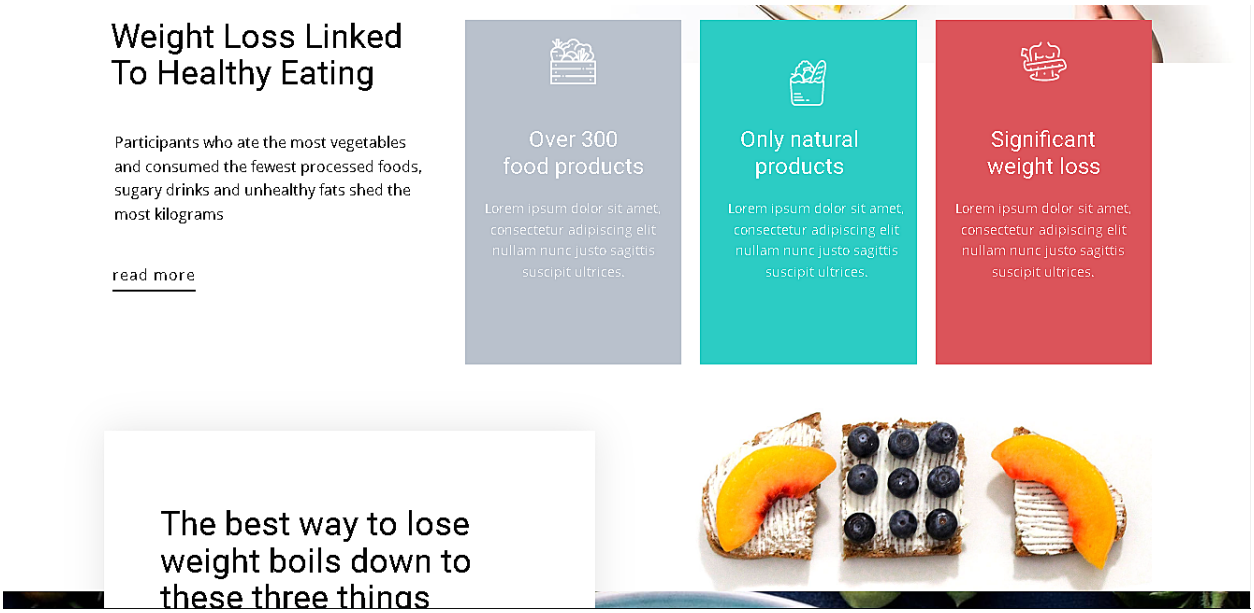
7.Efficiency of the product: 80%

8.Quality of the product: 80%

OUTPUT:

OVERVIEW OF THE APPLICATION ON A WEBPAGE





ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

1.Low cost.

2.Simple UI.

3.Faster response due to single page web page.

4.Capability of adding many features with ease and less cost.

DISADVANTAGES:

1.Lack of efficiency.Efficiency of the product needs to be proved.

2.Consistency of the product is not 100%.

3.Not a compact sized product.Size needs to be decreased.

CONCLUSION

In conclusion, many people have become aware of their health. Moreover, they are also informed how to live a healthy lifestyle. Health education is very important as it improves the health standards of the country which is highly populated where this application speaks sound. It further helps in preventing diseases and making people more aware of their health conditions. Most importantly, it not only focuses on physical health But it also helps in mental health, gives stability where it is much need one in current world's scenario. Current trends are about dietary self-monitoring based on mobile applications where in these mobile applications can upgrade people’s lifestyles.

Our generation has grown lazy as a result of technological growth. The cost of app development technology has almost entirely eliminated the physical labour, which is the root of many problems. These data indicate that we are in a dire situation, and the health and fitness agenda urgently needs a boost. For these

folks, the Nutrition Assistant app is a lifesaver. They aid users in maintaining a healthy diet and closely monitoring their calorie intake.

FUTURE SCOPE

1.This application can be used anywhere at anytime.

2.Results will be faster.

APPENDIX:

Source Code:
https://github.com/IBM-EPBL/IBM-Project-34028-1660230699/tree/main/Final%20Deliverables

GITHUB LINK:
https://github.com/IBM-EPBL/IBM-Project-34028-1660230699

PROJECT DEMO LINK: https://youtu.be/Rcb4Ig7YbPA