

# *"MEAL PLAN ANALYSIS"*

A Project Report Submitted to  
Rajiv Gandhi Proudhyogiki Vishwavidyalaya



Towards Partial Fulfillment for the Award of  
Bachelor of Technology in Computer Science

*Submitted by:*  
ANURAG MAHAJAN  
ANUSHKA BHANPIYA  
ARYAN TAPKIRE  
AVANI JAIN

*Guided by:*  
PROF. RONAK JAIN  
DEPT. OF COMPUTER SCIENCE  
AITR, INDORE



# EXAMINER APPROVAL

The Project entitled "*Meal Plan Analysis*" submitted by Anurag Mahajan, Anushka Bhanpiya, Aryan Tapkire, Avani Jain has been examined and is hereby approved towards partial fulfillment for the award of *Bachelor of Technology degree in Computer Science* discipline, for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner)

Date:

(External Examiner)

Date

# GUIDE RECOMMENDATION

This is to certify that the work embodied in this project entitled "*Meal Plan Analysis*" submitted by Anurag Mahajan, Anushka Bhanpiya, Aryan Tapkire, Avani Jain is a satisfactory account of the bonafide work done under the supervision of PROF.RONAK JAIN, is recommended towards partial fulfillment for the award of the Bachelor of Engineering (Computer Science and Engineering ) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

(Project Guide)

(Project Coordinator)

# STUDENTS UNDERTAKING

This is to certify that a project entitled "*Meal Plan Analysis*" has been developed by us under the supervision of *PROF. RONA K JAIN*. The whole responsibility of work done in this project is ours. The sole intention of this work is only for practical learning and research.

We further declare that to the best of our knowledge, this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work is found then we are liable for explanation to this.

# Acknowledgement

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

There are number of people without whom this projects work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor PROF. RONAK JAIN, Professor, AITR, Indore for his motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to PROF. RONAK JAIN, HOD CSE, AITR Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of CSE Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if we do not acknowledge the support and guidance received from Dr S C Sharma, Director, AITR, Indore whenever needed. We take the opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for the project to achieve our objectives.

# Executive Summary

## *Meal Plan Analysis*

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal(MP), India for partial fulfillment of Bachelor of Engineering in Information Technology branch under the sagacious guidance and vigilant supervision of *PROF.RONAK JAIN*.

In the project, Flask is used , which is an open-source python framework.Libraries such as SQLAlchemy were also used for ease of database access and creation. It is used for creating and hosting lightweight web applications.It is implemented on Werkzeug and Jinja2.The project uses SQLITE database for data storage of users .

**Key words :** Flask,SQLITE ,SQLAlchemy

*"Where the vision is one year,  
cultivate flowers;*

*Where the vision is ten years,  
cultivate trees;*

*Where the vision is eternity,  
cultivate people."* -

*Oriental Saying*

# List of Figures

---

Figure 2-1 : Data Flow Diagram Level 0	6
Figure 2-2: Data Flow Diagram Level 1	7
Figure 2-3 : Data Flow Diagram Level 2	8
Figure 3-1 :Screenshot 1	13
Figure 3-2 : Screenshot 2	13
Figure 3-3 : Screenshot 3	14
Figure 3-4 : Screenshot 4	14
Figure 3-5 : Screenshot 5	15
Figure 3-6 : Screenshot 6	15
Figure 3-7 :Screenshot 7	15
Figure 3-8 :Screenshot 8	16
Figure 3-9 : Screenshot 9	16
Figure 3-10: Screenshot 10	17
Figure 3-11 : Screenshot 11	17



# Table of Contents

---

S.NO	TOPIC	PAGE NO.
1.	Introduction	1-4
1.1	Overview	1
1.2	Background and Motivation	1
1.3	Problem Statement and Objectives	2
1.4	Scope of the Project	2
1.5	Team Organization	2
1.6	Report Structure	3
2.	Proposed System	5-10
2.1	The Proposal	5
2.2	Benefits of the Proposed System	5
2.3	Feasibility Study	5
2.4	Design Representation	6
2.5	Deployment Requirements	9
3.	Implementation	11-17
3.1	Technique and Tools Used	11
3.2	Language Used	12
3.3	Screenshots	13
4.	Conclusion	18-19
4.1	Conclusion	18
4.2	Limitations of the Work	19
	Bibliography	20
	Project Plan	21
	Guide Interaction Sheet	22
	Source code	23-33



# Chapter 1 .Introduction

## Introduction

---

The application in subject, provides a means for keeping a check on a person's daily eating habits by providing calorie intake information as well as providing suggestions of foods a person should be eating in order to maintain a nutritional balance. The application tracks a user's whole day consumption of food and accordingly determines the food suggestions based on his proteins, fats and carbs intake. The application in subject, provides a means for keeping a check on a person's daily eating habits by providing calorie intake information as well as the food suggestions and timely food reminders of what a person can consume as per his current physical needs.

### 1.1 Overview

A diet recall is a dietary assessment tool that consists of a structured interview in which participants are asked to recall all food and drink they have consumed in the previous 24/7 days. The Outcome of the web application is to help users a detailed understanding of their daily diet and the changes that are required to maintain a healthy diet.

### 1.2 Background and Motivation

The main aim of building this project is to provide the public with an application interface that helps them to keep a track of their daily eating and drinking diet components and have a systematically prepared diet with appropriate nutritional information.

The application in subject, provides a means for keeping a check on a person's daily eating habits by providing calorie intake information as well as the food suggestions and timely food reminders of what a person can consume as per his current physical needs.

The application also helps to keep a daily/weekly/monthly track record of user's diet and provides calorie intake information

### 1.3 Problem Statement and Objectives

Web Application to keep track of Quantity/Quality/type of food taken  
Built food analyzer Application that allows user to keep track of their food and also recommend best dietary chart based on their age,gender ,disease(if any) and BMI. Application should have gamification feature which motivates the user to follow the diet chart (Like badges, points etc...)"

### 1.4 Scope of the Project

People are increasingly becoming diet conscious and they require a dietitian to help them keep a track of their daily eating habits. This Web application works as a substitute to that. Users can now themselves find out what is missing from their diets and what they should be eating with the help of this app. The app helps the user to enter the food items that are consumed by him today and obtain calorie intake considering those food items in terms of proteins, fats and carbs. Therefore, considering the total intake, the application is able to analyze that the user lacks in which nutrients and thus is able to provide a healthy list of food items that are rich in those lacking nutrients. Thus this application serves as an alternative to the dietitian by providing necessary food suggestions as per the user's whole day calorie intake and as per his current physical parameters like height, weight, age and gender ; and by providing timely food reminders at breakfast, lunch, evening snacks and dinner –specific timings regarding what a person can have at these respective meals of the day. A person of any age can use this application.

### 1.5 Team Organization

*ANURAG MAHAJAN*

I investigated and found the right technology and studied in deep about it. For the implementation of the project , I also collected the object data .Documentation is also a part of the work done by me in this project.

*ANUSHKA BHANPIYA*

Implementation logic for the project objective and coding of internal functionalities is done by me. Also, worked on Back end design for storing results in database for maintaining logs.

*ARYAN TAPKIRE*

I also worked on the implementation of flask framework and analysis of meals of user in the project. Worked on creating a database for storing results in the database SQLITE.

*AVANI JAIN*

Along with doing preliminary investigation and understanding the limitations of the current system, I studied about the topic and its scope and surveyed various research papers related to the meal analysis and the technology that is to be used.

## 1.6 Report Structure

The project *Meal Plan Analysis* is primarily concerned with the meal analysis and the whole project report is categorized into five chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of the project which is then subsequently ended with report outline.

Chapter 2: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken.

Later it gives details of the different deployment requirements for the developed project.

Chapter 3: Implementation - includes the details of different the Project. It also includes different Technology/ Techniques/ Tools/ Programming Languages used in developing the with along project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter4: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

# Chapter 2 . Proposed System

## Proposed System

---

### 2.1 The Proposal

The proposal is to deploy a system at the entry gate which can analyse meal taken by user and store that information in the database. It can also give suggestions for food items for a healthy lifestyle to user.

### 2.2 Benefits of the Proposed System

The current system had a lot of challenges that are overcome by this system :

- ☐ Economic : The proposed system is economic as there is no cost involved in it.
- ☐ Man Power : It does not require any person(dietician) or their efforts to suggest food to users.
- ☐ 24 x 7 Availability : As it is a web application ,it is always available provided you have internet connection.

Promote Healthy lifestyle: As right food suggestion is just at a click.

### 2.3 Feasibility Study

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

#### 2.3.1 Technical

The framework used in this is Flask, which is a framework Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. The system once set up completely, works automatically without needing any person to operate it. The result gets automatically displayed to user. For making the system technically feasible, there is a requirement of system with high processor for better performance.

### 2.3.2 Economical

Since the system is completely automated, there is a need of continuous network supply for it to operate 24X7. The Flask framework used in the system works great and is in most demand.

### 2.3.3 Operational

The main motto of our system is to reduce the manual efforts of dietician for diet planning. The system is able to do that accurately and efficiently making the system operationally feasible.

## 2.4 Design Representation

### 2.4.1 Data Flow Diagrams

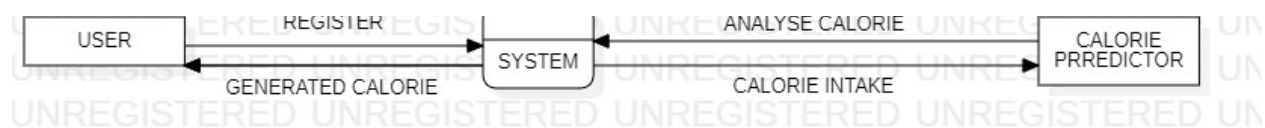


Figure 2-1 Data Flow Diagram Level 0



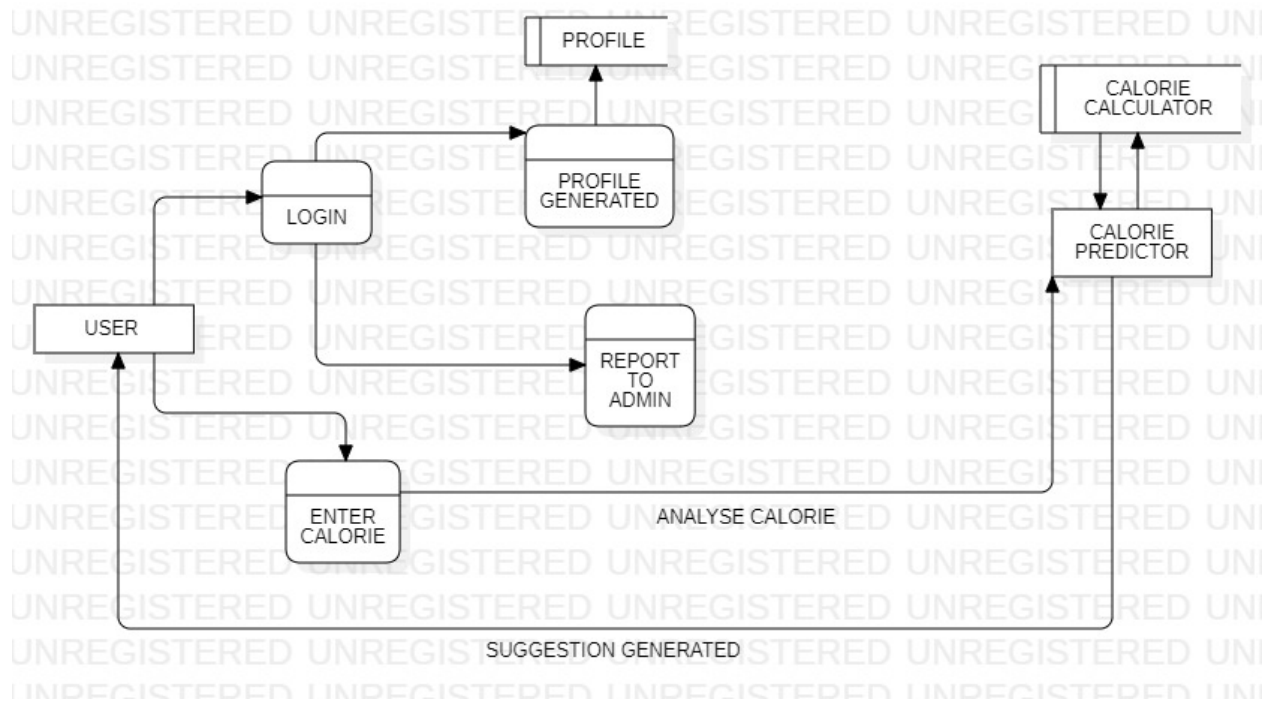


Figure 2-2 Data Flow Diagram Level 1

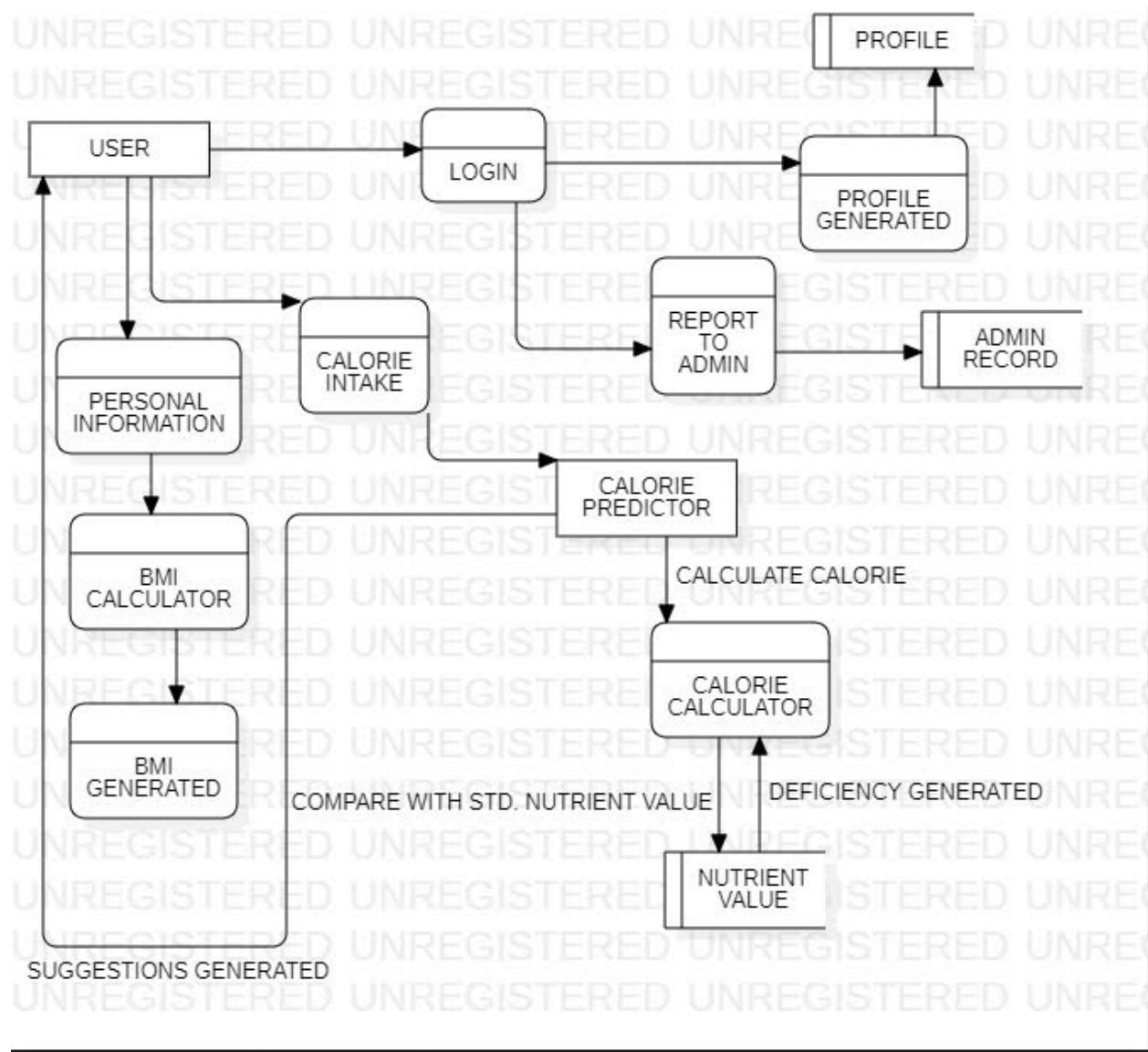


Figure 2-3 Data Flow Diagram Level 2

### 2.4.2 Database Structure

The name of the database created is "User.db" and there is one table in the database named "user" for storing the records.

The "user" table has the following structure :

NAME	DATATYPE	DESCRIPTION
ID	Integer	Unique identification for every user.
USERNAME	Varchar2	Unique name selected by user.
EMAIL	Varchar2	Email address of user
<b>PASSWORD</b>	Varchar2	Password required for authentication of user.
<b>GENDER</b>	Varchar2	Gender of user
<b>DISEASE</b>	Varchar2	Specific disease of user if any
<b>AGE</b>	Integer	Age of user
<b>WEIGHT</b>	Integer	Weight of user
<b>HEIGHT</b>	Integer	Height of user

Table 1 : Database Structure

## 2.5 Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below :

### 2.5.1 Hardware

- ☐ 32-bit, x86 Processing system
- ☐ Windows 7 or later operating system
- ☐ High processing computer system without GPU or with GPU(high performance)

## 2.5.2 Software

- ☐ Python and its supported libraries
- ☐ Flask

If Installing Flask in VSCODE :

1. In VS Code, create a new file in your project folder named app.py using either File > New from the menu, pressing Ctrl+N, or using the new file icon in the Explorer View .
2. In app.py , add code to import Flask and create an instance of the Flask object.

# Chapter 3 .Implementation

## Implementation

---

For the problem of suggesting food items for proper nutrition to every individual ,this application would take some basic input like calorie intake, age, height,bmi and specific ailments if any.

### 3.1 Technique and Tools Used

#### 3.1.1 Flask

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

Applications that use the Flask framework include Pinterest and LinkedIn

#### 3.1.2 SQLITE

SQLite is a database engine written in the C programming language. It is not a standalone app; rather, it is a library that software developers embed in their apps. As such, it belongs to the family of embedded databases. It is the most widely deployed database engine, as it is used by several of the top web browsers, operating systems, mobile phones, and other embedded systems.

Many programming languages have bindings to the SQLite library. It generally follows PostgreSQL syntax, but does not enforce type checking by default. This means that one can, for example, insert a string into a column defined as an integer.

### 3.2 Language Used

Python language is used in the system due to the following Characteristics :

Simple :

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English (but very strict English!). This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the syntax i.e. the language itself.

Free and Open Source :

Python is an example of a FLOSS (Free/Libre and Open Source Software). In simple terms, you can freely distribute copies of this software, read the software's source code, make changes to it, use pieces of it in new free programs, and that you know you can do these things. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and improved by a community who just want to see a better Python.

Object Oriented :

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simple way of doing object-oriented programming, especially, when compared to languages like C++ or Java.

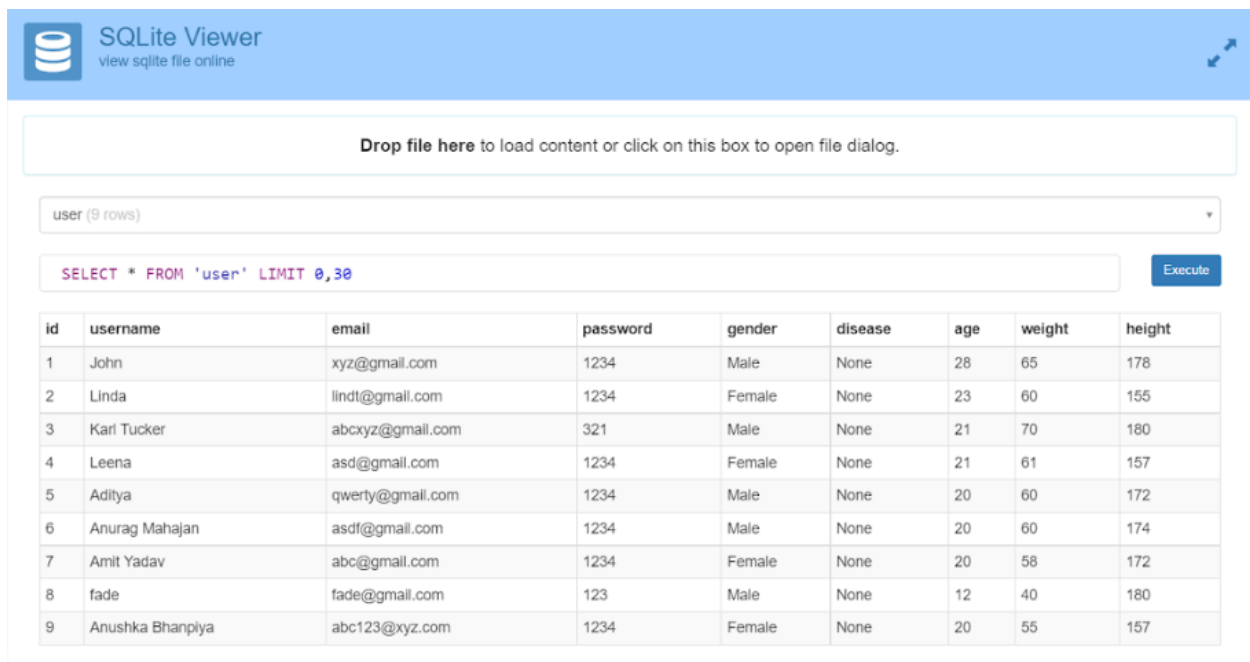
Extensive Libraries :

The Python Standard Library is huge indeed. It can help you do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, ftp, email, XML

XML-RPC, HTML, WAV files, cryptography, GUI(graphical user interfaces) using Tk, and also other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the "batteries included" philosophy of Python.

### 3.3 Screenshots

The Following are the screenshots of the result of the project :



The screenshot shows the SQLite Viewer web application interface. At the top, there's a blue header with the SQLite logo and the text "SQLite Viewer" and "view sqlite file online". Below the header, there's a light blue box with the text "Drop file here to load content or click on this box to open file dialog." Below this, there's a dropdown menu showing "user (9 rows)". Below the dropdown, there's a text input field containing the SQL query "SELECT \* FROM 'user' LIMIT 0,30" and a blue "Execute" button. Below the query input, there's a table with 9 rows and 9 columns. The columns are: id, username, email, password, gender, disease, age, weight, and height. The rows contain user data.

id	username	email	password	gender	disease	age	weight	height
1	John	xyz@gmail.com	1234	Male	None	28	65	178
2	Linda	lindt@gmail.com	1234	Female	None	23	60	155
3	Karl Tucker	abxxyz@gmail.com	321	Male	None	21	70	180
4	Leena	asd@gmail.com	1234	Female	None	21	61	157
5	Aditya	qwerty@gmail.com	1234	Male	None	20	60	172
6	Anurag Mahajan	asdf@gmail.com	1234	Male	None	20	60	174
7	Amit Yadav	abc@gmail.com	1234	Female	None	20	58	172
8	fade	fade@gmail.com	123	Male	None	12	40	180
9	Anushka Bhanpiya	abc123@xyz.com	1234	Female	None	20	55	157

Figure 3-1: Screenshot 1

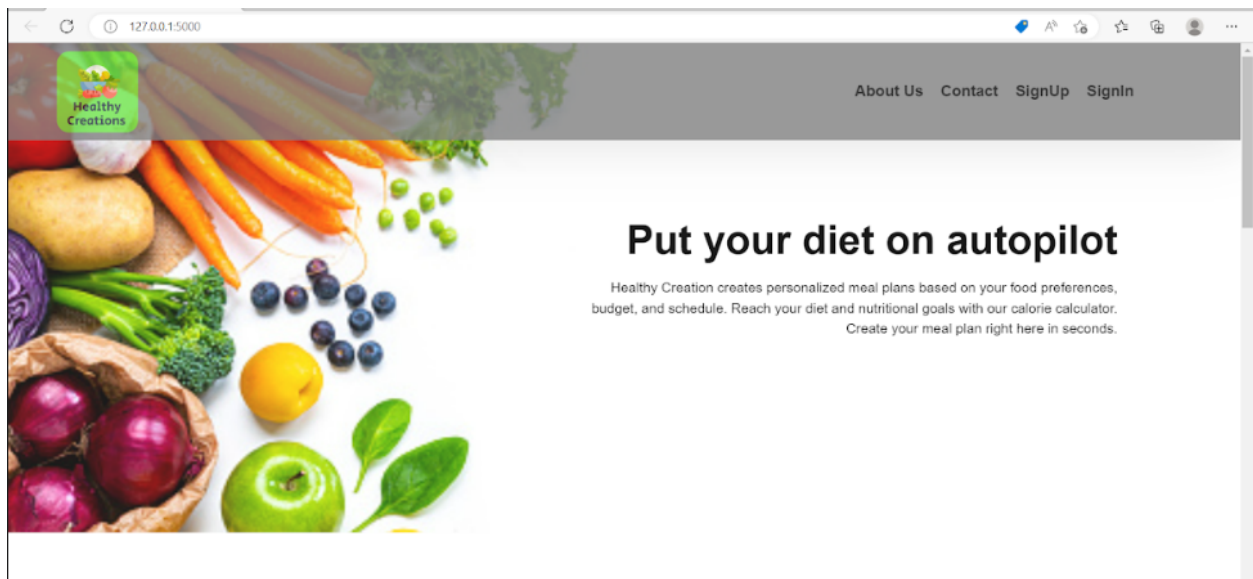


Figure 3-2 : Screenshot 2

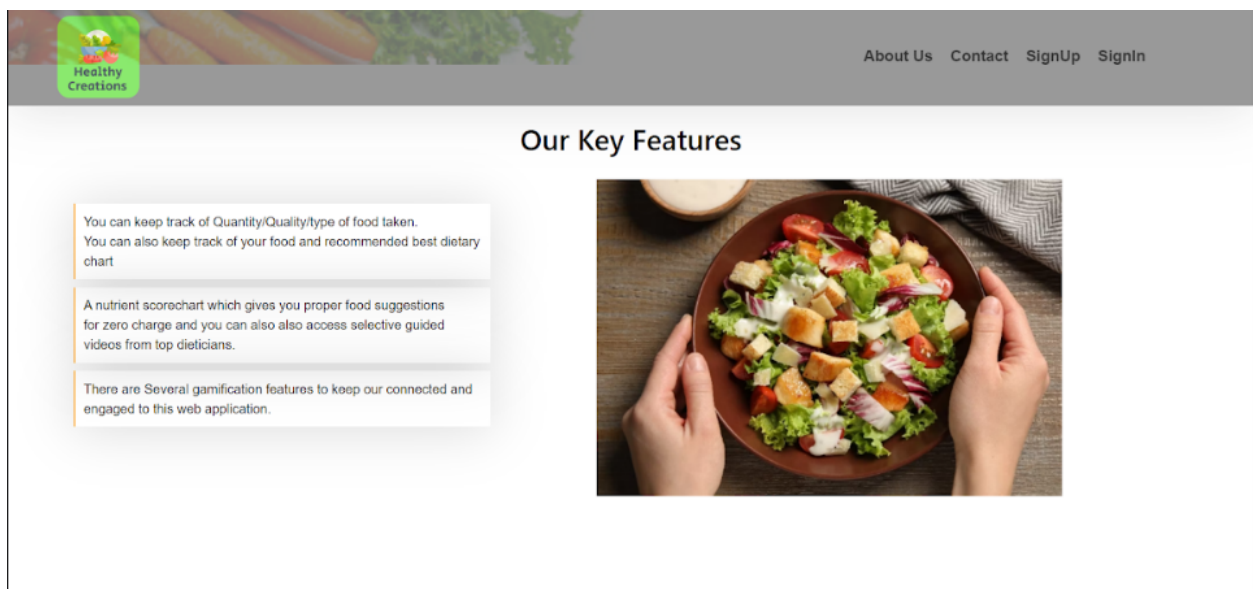


Figure 3-3 : Screenshot 3



## CONTACT US



Address : Acropolis Institute Of Technology And Research, Indore, Madhya Pradesh, India

Figure 3-4: Screenshot 4

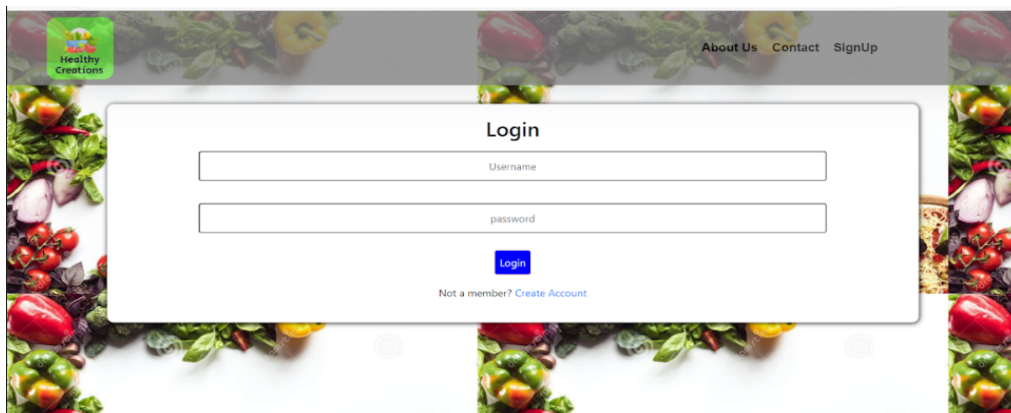


Figure 3-5: Screenshot 5

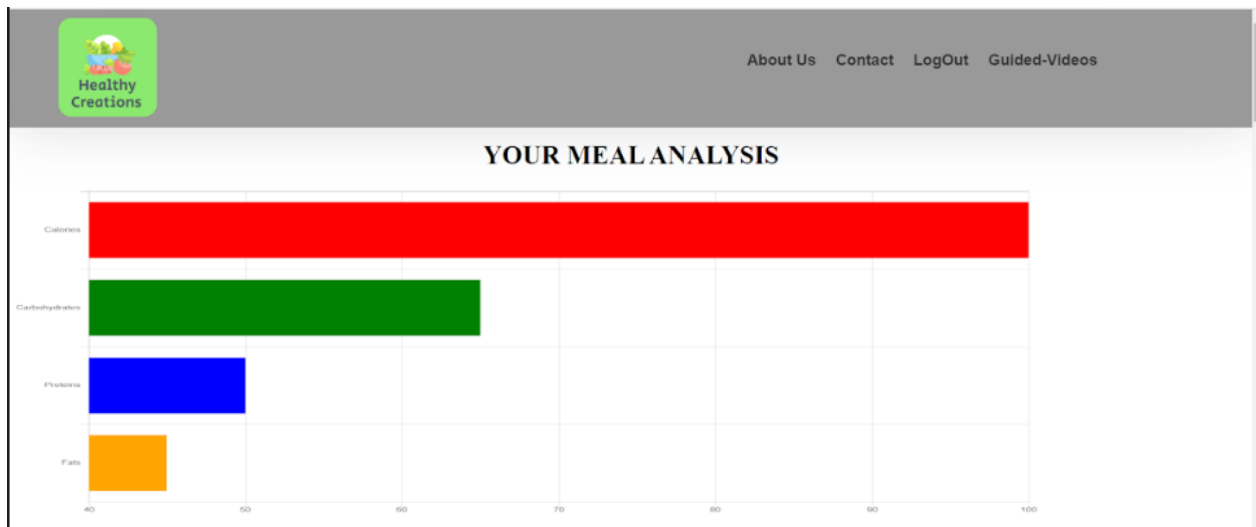
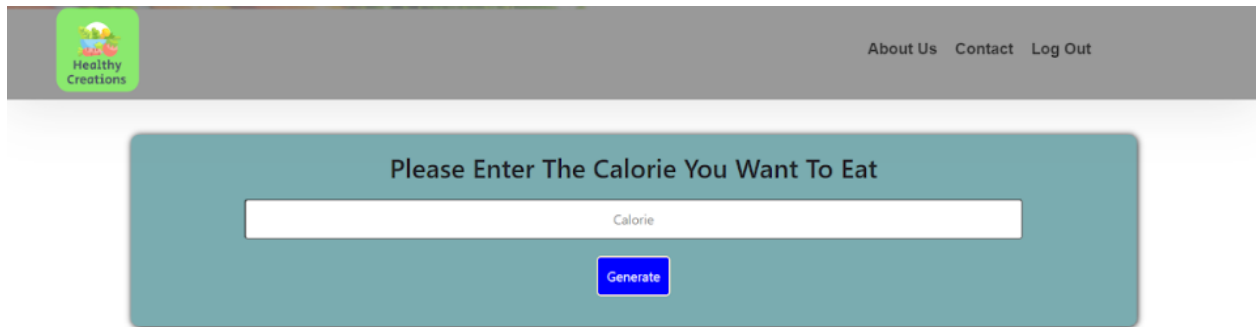


Figure 3-6: Screenshot 6



Healthy Creations

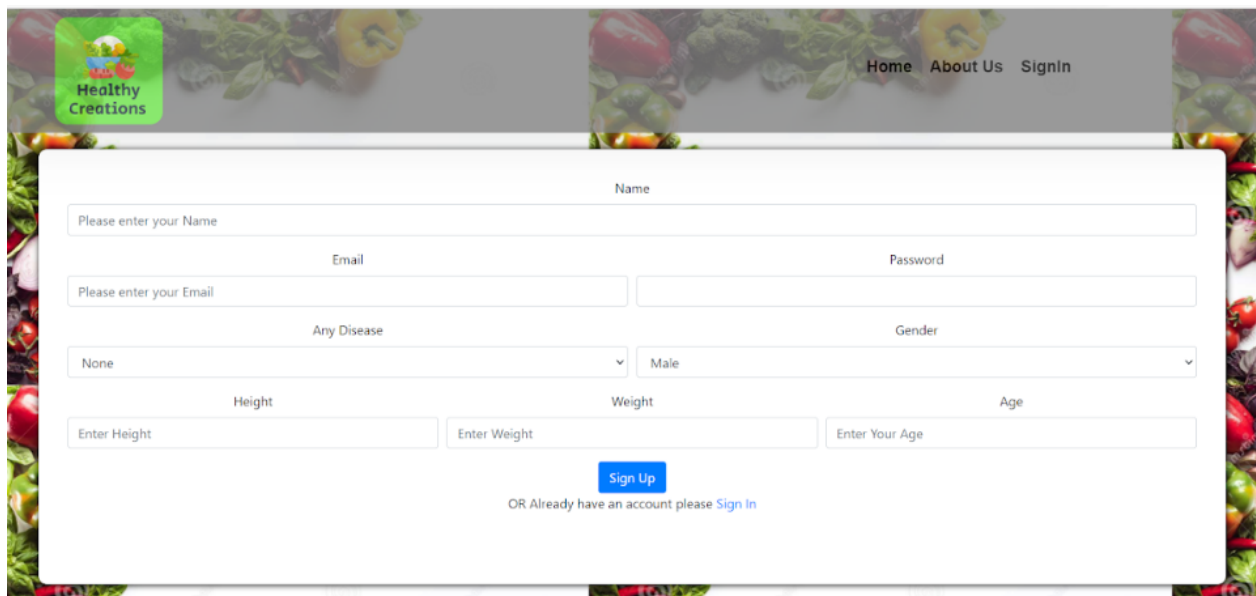
About Us Contact Log Out

Please Enter The Calorie You Want To Eat

Calorie

Generate

Figure 3-7: Screenshot 7



Healthy Creations

Home About Us SignIn

Name

Please enter your Name

Email Password

Please enter your Email

Any Disease Gender

None Male

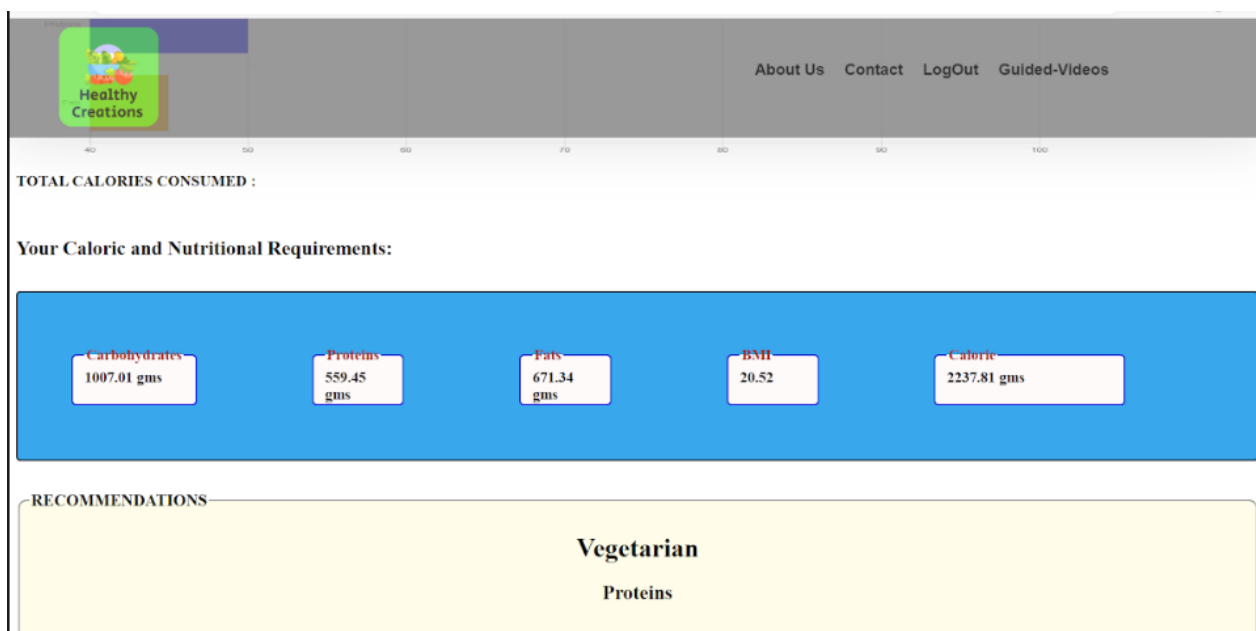
Height Weight Age

Enter Height Enter Weight Enter Your Age

Sign Up

OR Already have an account please [Sign In](#)

Figure 3-8: Screenshot 8



Healthy Creations

About Us Contact LogOut Guided-Videos

40 50 60 70 80 90 100

TOTAL CALORIES CONSUMED :

Your Caloric and Nutritional Requirements:

Carbohydrates 1007.01 gms	Proteins 559.45 gms	Fats 671.34 gms	BMI 20.52	Calorie 2237.81 gms
------------------------------	------------------------	--------------------	--------------	------------------------

RECOMMENDATIONS

Vegetarian  
Proteins

Figure 3-9: Screenshot 9

1007.01 gms

559.45 gms

671.34 gms

20.52

2237.81 gms

[About Us](#)
[Contact](#)
[LogOut](#)
[Guided-Videos](#)

RECOMMENDATIONS

Vegetarian

Proteins

Oats

Amount of proteins per serving : 13 grams

Required amount of given dish : 43.034615384615385 grams

Paneer

Amount of proteins per serving : 14 grams

Required amount of given dish : 39.96071428571429 grams

Chick Peas

Amount of proteins per serving : 18 grams

Required amount of given dish : 31.08055555555556 grams

Lentils

Amount of proteins per serving : 19 grams

Required amount of given dish : 29.444736842105264 grams

Figure 3-10: Screenshot 10

[About Us](#)
[Contact](#)
[Log Out](#)

5 Best Foods for Diabetes Control | Bes...

2 MILLION+ VIEWS

5 BEST FOODS TO CONTROL DIABETES

Watch on YouTube

DIET FOR DIABETES - 5 TRADITIONAL I...

TRADITIONAL INDIAN FOODS FOR DIABETES

Watch on YouTube

DIABETES diet plan (Hindi) || Indian || Ve...

क्या खायें और क्या न खायें ?

Watch on YouTube

Top 6 Carbohydrate Sources | Best and ...

BEST CARB

Watch on YouTube

Figure 3-11: Screenshot 11

18

# Chapter 4 . Conclusion

## Conclusion

---

### 4.1 Conclusion

Thus this application will help to provide users a detailed understanding of their daily diet and the changes that are required to maintain a healthy diet.

It will be easier for users to get a hold of alternatives for their food habits while also maintaining the level of nutrition. There is a need for maintaining track of a person's daily food habits and accordingly formulate a desirable diet to maintain a healthy lifestyle. However, these functionalities are barely available under one roof. This application fulfills that requirement of a user. This application will help a lot of users to maintain a healthy lifestyle as the app will take care of how much the user eats and what he/she should be eating for a balanced food intake. This application will stand as a good commercial app in today's world including a good combination of desirable features and workflow that will provide the best way of maintaining user's health. Also further, the application will be flexible enough to include more add-on features. As this is an academic level project, we are using a limited data and a .net server as an extensive database will not be financially feasible. However, if the project has to be enhanced commercially, we can create a cloud to maintain the extensive data.

## 4.2 Limitations of the Work

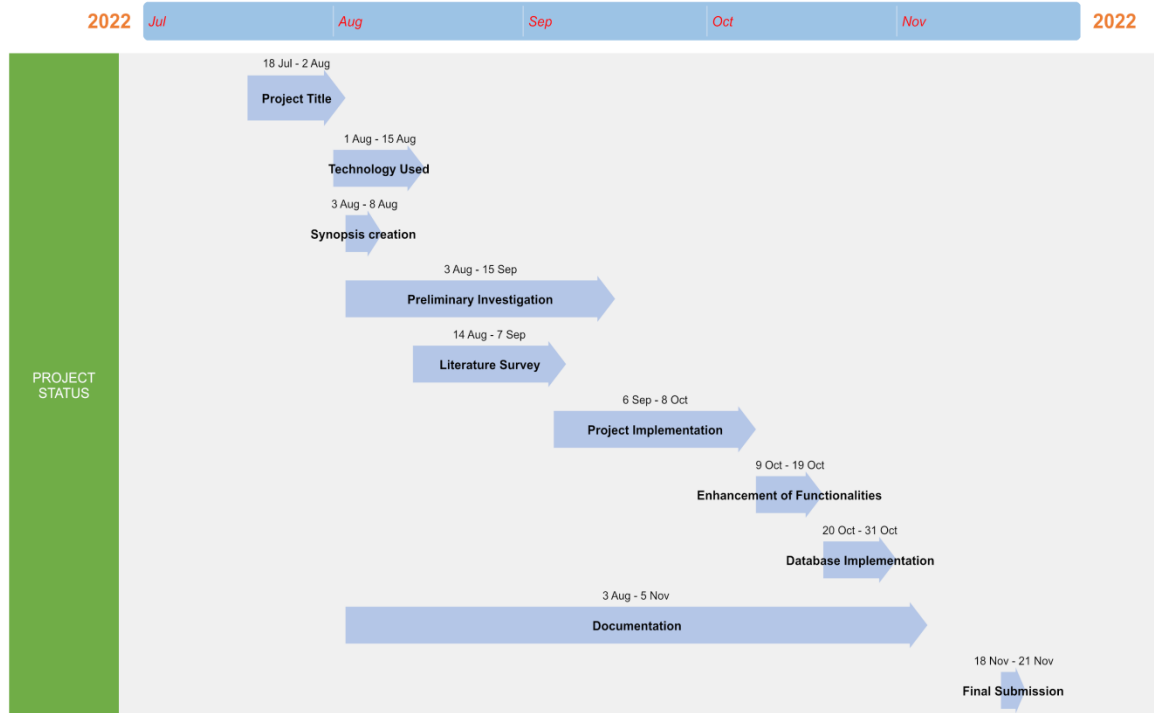
1. It depends on how accurately the user enters calories.
2. It takes into consideration only basic factors and diseases leaving behind more complicated diseases .
3. Quantity of food to be taken is not suggested to the user.

# BIBLIOGRAPHY

- Webster J, Waqanivalu T, Arcand J, et al. Understanding the science that supports **population-wide** salt reduction programs. J Clin Hypertens (Greenwich). **2017;19(6):569-576**. [PMC free article] [PubMed] [Google Scholar]
- World Health . Organization. **Follow-up** to the Political Declaration of the **High-level** Meeting of the General Assembly on the Prevention and control of **Non-communicable** Diseases: **Sixty-sixth** World Health Assembly (WHA66.10). Geneva, Switzerland: World Health Organization, 2013. [Google Scholar]
- TRUE Consortium . Recommended standards for assessing blood pressure in human research where blood pressure or hypertension is a major focus. J Clin Hypertens (Greenwich). **2017;19(2):108-113**. 10.1111/jch.12948. [PMC free article] [PubMed] [CrossRef] [Google Scholar]
- Cogswell ME, Zhang Z, Carriquiry AL, et al. Sodium and potassium intakes among US adults: NHANES 2003–2008. Am J Clin Nutr. **2012;96(3):647-657**. 10.3945/ajcn.112.034413. [PMC free article] [PubMed] [CrossRef] [Google Scholar]
- Baranowski T. **24-Hour** Recall and Diet Record Methods. In: Willett WC, ed.
- Nutritional Epidemiology. Oxford: Oxford University Press; 2013. [Google Scholar]
- Conway JM, Ingwersen LA, Moshfegh AJ. Accuracy of dietary recall using the USDA
- Gemming L, Jiang Y, Swinburn B, et al. **Under-reporting** remains a key limitation of **self-reported** dietary intake: an analysis of the 2008/09 New Zealand Adult
- Nutrition Survey. Eur J Clin Nutr. **2013;68(2):259-264**. [PubMed] [Google Scholar]

# PROJECT PLAN

## Project Status Gantt Chart





# GUIDE INTERACTION SHEET

Date	Discussion	Action Plan
01/08/2022	Discussed about the title of the project.	Meal plan analysis was decided as the title.
08/08/2022	Discussion on the technology to be used.	Flask(Python was finalized)
10/08/2022	Discussion of the creation of synopsis of the project.	Gathering of information for synopsis creation
17/08/2022	Suggestions on how to do a literature survey and preliminary investigation on the topic.	Many research papers were read , understood and their abstract were to be written.
22/08/2022	Discussion on the implementation of the project.	Started using Flask framework
29/08/2022	Discussion on the objective of the project.	Decided to Include the logic of meal analysis.
3/10/2022	For storing the result, database was advised to be added.	SQLite was used for the same purpose.
26/10/2022	Discussion on project documentation	Decided to write the content and integrate it in the proper format of the report.

# Source Code

## 1. app.py

```
from flask import Flask ,render_template,redirect,request,url_for,flash
from flask_sqlalchemy import SQLAlchemy
from flask_login import LoginManager,login_user,UserMixin

app = Flask(__name__)
app.config["SQLALCHEMY_DATABASE_URI"] = "sqlite:///user.db"
app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False
app.config['SECRET_KEY'] = 'thisissecret'
db = SQLAlchemy(app)
login_manager = LoginManager()
login_manager.init_app(app)

class User(UserMixin,db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(25), unique=True, nullable=False)
    email = db.Column(db.String(25), unique=True,nullable=False)
    password = db.Column(db.String(25), nullable=False)
    gender = db.Column(db.String(25), nullable=False)
    disease = db.Column(db.String(25), nullable=False)
    age = db.Column(db.Integer, nullable=False)
    weight = db.Column(db.Integer, nullable=False)
    height = db.Column(db.Integer, nullable=False)

    def __repr__(self):
        return '<User %r>' % self.username

@login_manager.user_loader
def load_user(user_id):
    return User.query.get(int(user_id))

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

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

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

```

@app.route("/Sign_up", methods=['GET','POST'])
def signup():
    if request.method=='POST':
        username = request.form['uname']
        email = request.form['email']
        password = request.form['password']
        gender = request.form['gender']
        disease = request.form['disease']
        age = request.form['age']
        weight = request.form['weight']
        height = request.form['height']
        user =
        User(username=username,email=email,password=password,gender=gender,
        disease=disease,age=age,weight=weight,height=height)
        db.session.add(user)
        db.session.commit()
        flash('Successfully registered','success')
        return redirect('/Sign_in')
    return render_template("Sign_up.html")

```

```

@app.route("/Sign_in", methods=['GET','POST'])
def signin():
    if request.method=='POST':
        username = request.form['uname']
        password = request.form['password']
        user = User.query.filter_by(username=username).first()
        if user and password==user.password:
            login_user(user)
            flash('Successfully logged in','success')
            return redirect(f'/main/{user.id}')
        else:
            flash('Invalid username or password','warning')
            return redirect('/Sign_in')
    return render_template("Sign_in.html")

```

```

@app.route("/main/<int:id>", methods=['GET','POST'])
def main1(id):
    obj = User.query.filter_by(id=id).first()
    # return redirect('/result')
    return render_template("main.html",obj=obj)

```

```

@app.route("/result/<int:id>", methods=['GET','POST'])
def result(id):
    obj = User.query.filter_by(id=id).first()
    ht = obj.height
    wt = obj.weight

```

```

age = obj.age
gender = obj.gender
if(gender == "Male"):
    cal = round((10*wt + 6.25*ht - 5*age + 5)*1.375,2)
else:
    cal = round((10*wt + 6.25*ht - 5*age - 161)*1.375,2)
carb = round(0.45*cal,2)
fat = round(0.30*cal,2)
protein = round(0.25*cal,2)
bmi = round(((wt/(ht*ht))*10000),2)
return
render_template("result.html",cal=cal,carb=carb,fat=fat,protein=protein,bmi=
bmi)

@app.route("/videos")
def video():
    return render_template("video.html")

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

```

## 2. Home.html

```

password = request.form['password']
gender = request.form['gender']
disease = request.form['disease']
age = request.form['age']
weight = request.form['weight']
height = request.form['height']
user =
User(username=username,email=email,password=password,gender
=gender,disease=disease,age=age,weight=weight,height=height)
db.session.add(user)
db.session.commit()
flash('Successfully registered','success')
return redirect('/Sign_in')
return render_template('Sign_up.html')
@app.route('/Sign_in',
methods=['GET','POST'])
def signin():
    if request.method=='POST':
        username = request.form['uname']
        password = request.form['password']
        user = User.query.filter_by(username=username).first()
        if user and password==user.password:
            login_user(user)
            flash('Successfully logged in','success')

```

```

        return redirect(f'#{user.id}#')
    else:
        flash('Invalid username or
password', 'warning')
        return redirect('/Sign_in')
    return render_template('Sign_in.html')
@app.route('/main/<int:id>',
methods=['GET', 'POST'])
def main1(id):
    obj = User.query.filter_by(id=id).first()
    # return redirect('/result')
    return render_template('main.html', obj=obj)
@app.route('/result/<int:id>',
methods=['GET', 'POST'])
def result(id):
    obj = User.query.filter_by(id=id).first()
    ht = obj.height
    wt = obj.weight
    age = obj.age
    gender = obj.gender
    if(gender == 'Male'):
        cal = round((10*wt + 6.25*ht - 5*age + 5)*1.375,2)
    else:
        cal = round((10*wt + 6.25*ht - 5*age - 161)*1.375,2)
    fat = round(0.30*cal,2)
    protein = round(0.25*cal,2)
    bmi = round(((wt/(ht*ht))*10000),2)
    return
    render_template('result.html', cal=cal, carb=carb, fat=fat,
    protein=protein, bmi=bm
i)
@app.route('/videos')
def video():
    return render_template('video.html')
if __name__ == '__main__':
    app.run(debug = True)

```

### 3. Result.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>

```

```

<link rel="stylesheet"
href="/static/result.css">
</head>
<body>
    <nav class="nav nav_top">
        <div class="logo">
            <a href="{url_for('#39;home#39;')}}"></a>
        </div>
        <div class="nav_side">
            <a href="{url_for('#39;about#39;')}}">About
Us</a>
            <!-- <a href="service.html">Services</a> -->
            <a
href="{url_for('#39;contact_us#39;')}}">Contact</a>
            <a href="{url_for('#39;home#39;')}}">LogOut</a>
            <a href="{url_for('#39;video#39;')}}">Guided-
Videos</a>
        </div>

</nav><br><br><br><br><br><br><br><br>
    <div>
        <h1 style="text-align: center;">YOUR MEAL
ANALYSIS</h1>
        <script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.5.0/Chart.min.js
">

        </script>
        <canvas id="myChart" style="max-width:1250px;max-
block-size: 400px;">
            <input type="text">
        </canvas>

    </div>
</script>
<div class="total_calorie">
    <h3>TOTAL CALORIES CONSUMED :</h3>
</div>

</div>
</div>
<br>
<h2>Your Caloric and Nutritional Requirements:</h2><br>

```

```

    <div class="calorie">
    <div class="cal">
        <fieldset class="cal1" id="Carbohydrates"
name="carb"
style="width: 50px;"
        <legend style="color: brown
; "><b>Carbohydrates</b></legend><b>{{carb}}
gms</b>
        </fieldset>
        <fieldset class="cal2" id="protein"
name="protein" style="
width: 80px;"
        <legend style="color:
brown;"><b>Proteins</b></legend><b>{{prote
in}} gms</b>
        </fieldset>
        <fieldset class="cal3" id="fats"
name="fats" style="width: 80px;"
        <legend style="color:
brown;"><b>Fats</b></legend><b>{{fat}}
gms</b>
        </fieldset>
        <fieldset class="cal4" id="BMI"
name="bmi" style="width: 80px;"
        <legend style="color:
brown;"><b>BMI</b></legend><b>{{bmi}}</
b>
        </fieldset><br><br><br><br><br>
    </div>
    <div class="rightbox">
        <fieldset class="cal5" id="Totalcalorie"
name="cal" style="width:
200px;"
        <legend style="color: brown;"><b>
Calorie</b></legend><b>{{cal}} gms</b>
        </fieldset>
    </div>
</div>
<script src="/static/chart.js"></script>
</div>
<br>
<br>
    <fieldset class="fieldset">
        <legend style="font-size:
20px"><b>RECOMMENDATIONS</b></legend>
        <div class="vegetarian">
            <h1>Vegetarian</h1>

```

```

</h2>Proteins</h2>
    <div class="food_item1">
        
        <p>Oats <br> <span>Amount of proteins per serving :
13
grams<br>
        Required amount of given dish : {{protein/13}}
grams</span></p>
    </div>
    <div class="food_item2">
        
        <p>Paneer <br> <span>Amount of proteins per serving
: 14
grams<br>
        Required amount of given dish : {{protein/14}}
grams</span></p>
    </div>
    <div class="food_item3">
        
        <p>Chick Peas <br> <span>Amount of proteins per
serving : 18
grams<br>
        Required amount of given dish : {{protein/18}}
grams</span></p>
    </div>
    <div class="food_item4">
        
        <p>Lentils <br> <span>Amount of proteins per serving
: 19
grams<br>
        Required amount of given dish : {{protein/19}}
grams</span></p>
    </div>
    <br>
    <h2>Fats</h2>
    <div class="food_item1">
        
        <p>Nuts(almonds etc.) <br> <span>Amount of fats per
serving :
54 grams<br>
        Required amount of given dish : {{fat/54}}
grams</span></p>

```



```

    </div>
    <div class="food_item2">
        
        <p>Stir Fried Corn <br><span>Amount of fats per
serving : 1.2
grams<br>
        Required amount of given dish : {{fat/1.2}}
grams</span></p> <br>
    </div>
    <div class="food_item3">
        
        <p>Coconut Milk <br><span>Amount of fats per
serving : 24
grams<br>
        Required amount of given dish : {{fat/24}}
grams</span></p> <br>
    </div>
    <div class="food_item4">
        
        <p>Tofu <br><span>Amount of fats per serving : 4.8
grams<br>
        Required amount of given dish : {{fat/4.8}}
grams</span></p> <br>
    </div>
    <br>
    <h2>Carbohydrates</h2>
    <div class="food_item1">
        
        <p>Bread <br><span>Amount of carbohydrates per
serving : 49
grams<br>
        Required amount of given dish : {{carb/49}}
grams</span></p>
    </div>
    <div class="food_item2">
        
        <p>Pasta <br><span>Amount of carbohydrates per
serving : 25
grams<br>
        Required amount of given dish : {{carb/25}}
grams</span></p>

```

```

</div>
<div class="food_item3">
  
  <p>Potatoes <br><span>Amount of carbohydrates per
serving : 17
grams<br>
  Required amount of given dish : {{carb/17}}
grams</span></p> <br>
</div>
<div class="food_item4">
  
  <p>Rice <br><span>Amount of carbohydrates per
serving : 28
grams<br>
  Required amount of given dish : {{carb/28}}
grams</span></p> <br>
</div>
</div>
<hr style="color: black;">
<div class="non_vegetarian">
  <h1>Non Vegetarian</h1>

  <h2>Proteins</h2>
  <div class="food_item1">
    
    <p>Eggs <br><span>Amount of proteins per serving :
13
grams<br>
    Required amount of given dish : {{protein/13}}
grams</span></p> <br>
    </div>
    <div class="food_item2">
      
      <p>Fish <br><span>Amount of proteins per serving :
22
grams<br>
      Required amount of given dish : {{protein/22}}
grams</span></p> <br>
      </div>
      <div class="food_item3">
        

```

<p>Chicken <br><span>Amount of proteins per  
 serving : 27  
 grams<br>  
 Required amount of given dish : {{protein/27}}  
 grams</span></p><br>  
 </div>  
 <div class="food\_item4">  
   
 <p>Beef <br><span>Amount of proteins per serving :  
 26  
 grams<br>  
 Required amount of given dish : {{protein/26}}  
 grams</span></p><br>  
 </div>  
 <br>  
 <h2>Fats</h2>  
 <div class="food\_item1">  
   
 <p>Lamb <br><span>Amount of fats per serving : 21  
 grams<br>  
 Required amount of given dish : {{fat/21}}  
 grams</span></p><br>  
 </div>  
 <div class="food\_item2">  
   
 <p>Scallops <br><span>Amount of fats per serving :  
 1.8  
 grams<br>  
 Required amount of given dish : {{fat/1.8}}  
 grams</span></p><br>  
 </div>  
 <div class="food\_item3">  
   
 <p>Tuna Fish <br><span>Amount of fats per serving :  
 1.6  
 grams<br>  
 Required amount of given dish : {{fat/1.6}}  
 grams</span></p><br>  
 </div>  
 <div class="food\_item4">  
 

```

        <p>Mutton <br><span>Amount of fats per serving : 21
grams<br>
        Required amount of given dish : {{fat/21}}
grams</span></p> <br>
    </div>
    <br>
    <h2>Carbohydrates</h2>
    <div class="food_item1">
        
        <p>Chicken <br><span>Amount of carbohydrates per
serving : 5
grams<br>
        Required amount of given dish : {{carb/5}}
grams</span></p>
    </div>
    <div class="food_item2">
        
        <p>Steak <br><span>Amount of carbohydrates per
serving : 0.1
grams<br>
        Required amount of given dish : {{carb/0.1}}
grams</span></p> <br>
    </div>
    <div class="food_item3">
        
        <p>Beef <br><span>Amount of carbohydrates per
serving : 0.1
grams<br>
        Required amount of given dish : {{carb/0.1}}
grams</span></p> <br>
    </div>
    <div class="food_item4">
        
        <p>Mutton <br><span>Amount of carbohydrates per
serving : 5
grams<br>
        Required amount of given dish : {{carb/5}}
grams</span></p> <br>
    </div>
</div>
</fieldset>

</body>

```

&lt;script src=&quot;/static/chart.js&quot;&gt;&lt;/script&gt;