AI-POWERED NUTRITION ANALYZER FOR FITNESS ENTHUSIASTS

A PROJECT REPORT

Submitted by

DEEPSHIKA RAGHURMAN
THRISHYA R
SURYA S
VISHNU A
MONISHA V

of

DEPARTMENT OF INFORMATION TECHNOLOGY
MADRAS INSTITUTE OF TECHNOLOGY
ANNA UNIVERSITY

CHENNAI - 600 044

INDEX

CHAPTER NO.	TITLE	PAGE NO.
1	**INTRODUCTION**	
	1.1 Project Overview	4
	1.2 Purpose	4
2	**LITERATURE SURVEY**	
	2.1 Existing Problem	5
	2.2 References	5
	2.3 Problem Statement Definition	6
3	**IDEATION AND PROPOSED SOLUTION**	
	3.1 Empathy Map Canvas	8
	3.2 Ideation and Brainstorming	9
	3.3 Proposed Solution	9
	3.4 Problem Solution Fit	11
4	**REQUIREMENT ANALYSIS**	
	4.1 Functional Requirement	13
	4.2 Non-Functional Requirements	15
5	**PROJECT DESIGN**	
	5.1 Data Flow Diagrams	16
	5.2 Solution and Technical Architecture	16
	5.3 User Stories	18
6	**PROJECT PLANNING AND SCHEDULING**	
	6.1 Sprint Planning and Estimation	19
	6.2 Sprint Delivery Schedule	20

7	**CODING AND SOLUTIONING**	
	7.1 Feature 1	21
	7.2 Feature 2	24
8	**TESTING**	
	8.1 Test Cases	26
	8.2 User Acceptance Testing	27
9	**RESULTS**	
	9.1 Performance Metrics	28
10	**ADVANTAGES AND DISADVANTAGES**	29
11	**CONCLUSION**	30
12	**FUTURE SCOPE**	31
13	**APPENDIX**	
	13.1 Source Code	32
	13.2 GitHub and Project Demo Link	90

CHAPTER 1
INTRODUCTION

1.1 PROJECT OVERVIEW

Food is a necessity for human life and has been addressed in numerous medical conventions. Modern dietary evaluation and nutrition analysis technologies give consumers more possibilities to explore nutrition patterns, comprehend their daily eating habits, and keep up a balanced diet. A crucial component of analytical chemistry that gives knowledge about the chemical make-up and quality of food is nutritional analysis, which is the process of determining the nutritional content of food.

Building a model that can be used to categorise food items is the project's major goal. Here, users can take pictures of various components and send them to a trained model. The model examines the image and determines the ingredient's nutrition.

1.2 PURPOSE

People should pay attention to their dietary intake to ensure that they follow a balanced diet and lead a healthy lifestyle. Paying attention to the food one consumes can help keep a check on health and related health issues. The presence of an application that helps analyze the calorie content of the items one consumes as well as keep track of the total calories consumed per day is of great benefit as it helps follow a rigorous diet. Thus, a nutrition analyzer application is required.

CHAPTER 2
LITERATURE SURVEY

2.1 EXISTING PROBLEM

Several applications exist for fitness enthusiasts. This helps them keep track of the amount of calories they burn and how much workout they clock in per day. There also exist applications that intimate the users about the calorie value of the food they consumer. However, they do not maintain the history of the calories consumed.

2.2 REFERENCES

[1] Rojas-Aranda, J.L., Nunez-Varela, J.I., Cuevas-Tello, J.C., Rangel-Ramirez, G. (2020). Fruit Classification for Retail Stores Using Deep Learning. In: Figueroa Mora, K., Anzurez Marín, J., Cerda, J., Carrasco-Ochoa, J., Martínez-Trinidad, J., Olvera-López, J. (eds) Pattern Recognition. MCPR 2020. Lecture Notes in Computer Science(), vol 12088. Springer, Cham. https://doi.org/10.1007/978-3-030-49076-8_1.

[2] Jana, S., Parekh, R., Sarkar, B. (2020). Automatic Classification of Fruits and Vegetables: A Texture-Based Approach. In: Mandal, J., Mukhopadhyay, S., Dutta, P., Dasgupta, K. (eds) Algorithms in Machine Learning Paradigms. Studies in Computational Intelligence, vol 870. Springer, Singapore. https://doi.org/10.1007/978-981-15-1041-0_5.

[3] C. Liu, X. Wang, J. Ni, Y. Cao and B. Liu, "An Edge Computing Visual System for Vegetable Categorization," 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019, pp. 625-632, doi: 10.1109/ICMLA.2019.00115.

[4] D. G. Savakar and A. K. Talawar, "Fuzzy C-Means Clustering based Identification of Indian Common Non-Leafy Vegetables," 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 2021, pp. 858-863.

[5] G. Zeng, "Fruit and vegetables classification system using image saliency and convolutional neural network," 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC), 2017, pp. 613-617, doi: 10.1109/ITOEC.2017.8122370.

[6] R. S. Chaulagain, S. Pandey, S. R. Basnet and S. Shakya, "Cloud Based Web Scraping for Big Data Applications," 2017 IEEE International Conference on Smart Cloud (SmartCloud), 2017.

[7] K. Jaspin, S. Selvan, J. D. Rose, J. Ebenezer and A. Chockalingam, "Real-Time Surveillance for Identification of Fruits Ripening Stages and Vegetables Maturation Stages with Infection Detection," 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), 2021, pp. 581-586, doi: 10.1109/ISPCC53510.2021.9609441.

[8] Khatun, Mehenag & Nine, Julker & Ali, Md. Forhad & Sarker, Pritom & Turzo, Nakib. (2020). Fruits Classification using Convolutional Neural Network. 5. 1-6.

2.3 PROBLEM STATEMENT DEFINITION

Humans should give equal importance to their health as they do towards other things. But in today’s fast moving lifestyle, people aren’t conscious about their health and follow questionable eating habits. When followed on a regular basis, they can have detrimental effects on the human body and may even be fatal.

Thus it is imperative to maintain good health. A person needs to follow a balanced diet, i.e. consume food containing proteins, vitamins and other vital nutrients that are needed by the human body in suggested proportions on a daily basis.

In this project, a system is developed to identify edible products and discern their nutritional information. The users of this system can capture images of the ingredients that go into their food and be informed of their nutritional composition. This way, fitness enthusiasts will be able to keep track of their calorie intake and people will be able to follow a healthy lifestyle of eating.

The work proposed is a simple application that can recognize the raw food items based on the input image and provide information regarding their nutritional value to the user.

CHAPTER 3
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map is a collaborative tool that teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. Empathy maps should be used throughout any UX process to establish common ground among team members and to understand and prioritize user needs. In user-centered design, empathy maps are best used from the very beginning of the design process.

3.2 IDEATION AND BRAINSTORMING

Ideation essentially refers to the whole creative process of coming up with and communicating new ideas. Ideation is innovative thinking, typically aimed at solving a problem or providing a more efficient means of doing or accomplishing something. Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.

3.3 PROPOSED SOLUTION

S. NO.	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	To develop a system that identifies edible products and discerns their nutritional information for the benefit of fitness enthusiasts
2.	Idea / Solution description	The system developed is an application that scans the surroundings to capture images. The image is analyzed to identify the fruits present in the image using machine learning models. Once the raw food items have been identified,

		their corresponding nutritional values are fetched from a database where the relevant details are stored. The application allows for a user to keep track of the amount of calories they consume in a day versus the total recommended amount for their dietary needs. The data of frequently consumed fruits is stored locally in the database.
3.	Novelty / Uniqueness	The proposed system maintains a personal nutrition calendar for the user and notifies them when they do not meet the requirements of their diet. Further, the app is inbuilt with features that suggest alternative foods, construct a food chart, develop a workout schedule, and recommend recipes that suit the caloric needs of the user. The system also integrates capabilities of identifying spoilt food items and whether fruits have ripened.
4.	Social Impact / Customer Satisfaction	The proposed application is useful for fitness enthusiasts to keep track of their calorie intake and thus maintain their physical state. Even those who are not conscious about their physique may use this application to lead a healthier lifestyle as it helps to keep track of what they eat, suggests healthy alternatives and recipes, as well as workout plans.
5.	Business Model (Revenue Model)	The application can be deployed for access by the general public. The application would draw the attention of several users who are

		determined to lead a healthy lifestyle and wish to undergo a physical transformation. The application could be built in such a way that features are progressively unlocked based on the subscription amount paid by the user starting from the generic nutrition analyzer feature to charting out personal plans for users.
6.	Scalability of the Solution	The proposed application has several features. It can be further enhanced to integrate more features based on feedback from users and ratings.

3.4 PROBLEM SOLUTION FIT

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why.

Purpose:

❑ Solve complex problems in a way that fits the state of your customer

❑ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior

❑ Sharpen your communication and marketing strategy with the right triggers and messaging

❑ Increase touchpoints with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems

❑ Understand the existing situation in order to improve it for your target group

Problem-Solution fit

Purpose / Vision: For developing an AI-Powered Nutrition Analyzer for Fitness Enthusiasts

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) — CS

Who is your customer?

Customers include fitness enthusiasts and those who are conscious of their health. The application's target audience include those who need a means to keep track of their nutritional intake like athletes and sportspersons, people working in the entertainment industry, as well as people who generally wish to live a healthy life.

6. CUSTOMER CONSTRAINTS — CC

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

- Lack of time to fully understand and dedicate towards fitness
- Hesitation towards altering one's diet and lifestyle
- Lack of reliable sources for guidance regarding dietary and fitness planning

5. AVAILABLE SOLUTIONS — AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

- Applications that keep track of workout and calories burnt
- These do not track calorie consumption and thus there is no control of intake which is equally important
- Personal trainers and nutritionists that are expensive

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS — J&P

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

- Monitoring calorie-count of consumed food
- Restricting consumption to recommended level such that a balanced diet is followed
- Maintaining consumption records of user to analyse overall health and diet
- Ensuring that people do not follow misinformation from the internet and end up with severe health issues
- Providing necessary personalised guidance

9. PROBLEM ROOT CAUSE — RC

What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.

- Lack of time to prepare healthy home-cooked meals due to the fast pace of life which leads to increased consumption of unhealthy fast-food
- Increase in cases of obesity and overall lack of healthiness among the general population
- The application is required to provide a means by which people can attempt to live a healthy lifestyle

7. BEHAVIOUR — BE

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

- Become members of physical fitness institutions/gyms or take up a sport
- Cutting down on consumption of carbs, fats, and sugars
- Avoiding junk food and eating outside
- Cooking simple yet nutritional meals at home

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

Identify strong TR & EM

3. TRIGGERS — TR

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

- Peer-pressure, beauty standards and society
- Health issues and illness

4. EMOTIONS: BEFORE / AFTER — EM

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

Before: Insecure, low self-esteem, and unfit
After: Self-confident, happy, and content

10. YOUR SOLUTION — SL

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

- Tracking per day calorie consumption
- Alerts for over-consumption
- Personal calendar to track eating habits
- Suggestions for nutritional food and recipes
- Creating personalized workout plan
- Detection of spoilt ingredients to prevent sickness

8. CHANNELS of BEHAVIOUR — CH

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

- Search about ways to become fit & lead a healthy lifestyle
- Comment on other's fitness on social media

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

- Observe people around them and analyze their level of fitness
- Try to follow fit people's lifestyle
- Comment on other's way of living

Extract online & offline CH of BE

CHAPTER 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	Users can create an account to use the application. This can be done by creating a persona on the application with a username and password or by making use of an existing email ID
FR-2	User Confirmation	Once a user registers onto the application, they receive a confirmation to their email id which they provide for registration. OTP authentication is integrated to ensure identity theft does not occur
FR-3	Calorie Calendar Creation	On creation of a user profile, a calendar is generated in association with the account. This calendar is private to the user and keeps track of the calories consumed in a day and related statistics
FR-4	Image Capturing and Processing	The application allows users to capture images of the ingredients they consume. These are given to the model for predicting their labels, i.e. identify the fruits. Further, the quantity of the fruits should be discerned. The application should be able to work with images of

		low quality and low resolution as well
FR-5	Calorie Value Computation	Once the labels of the ingredients and their quantity have been found, the net calorie value of the meal is calculated by summing up the calories of each ingredient in their respective amounts. The calorie values are fetched from the internet while that of frequently used items are fetched from a database
FR-6	Storage of Data	Data about the user and their log in details are stored in a backend database. Apart from these, calorific information of frequently consumed ingredients are also stored to minimize overhead and complexity
FR-7	Calorie Over-Consumption Notification	When a user exceeds their permissible calorie consumption amount for the day, the application issues a notification for the same. The application then suggests low-calorie diets to ensure minimum over-consumption
FR-8	Diet-Plan Specification	Users can select the kind of diet plan they want to follow with a target in mind such as weight loss, muscle building, etc. The application sources diet plans and food items that supplement their goals from the internet to help them achieve their goal

4.2 NON-FUNCTIONAL REQUIREMENTS

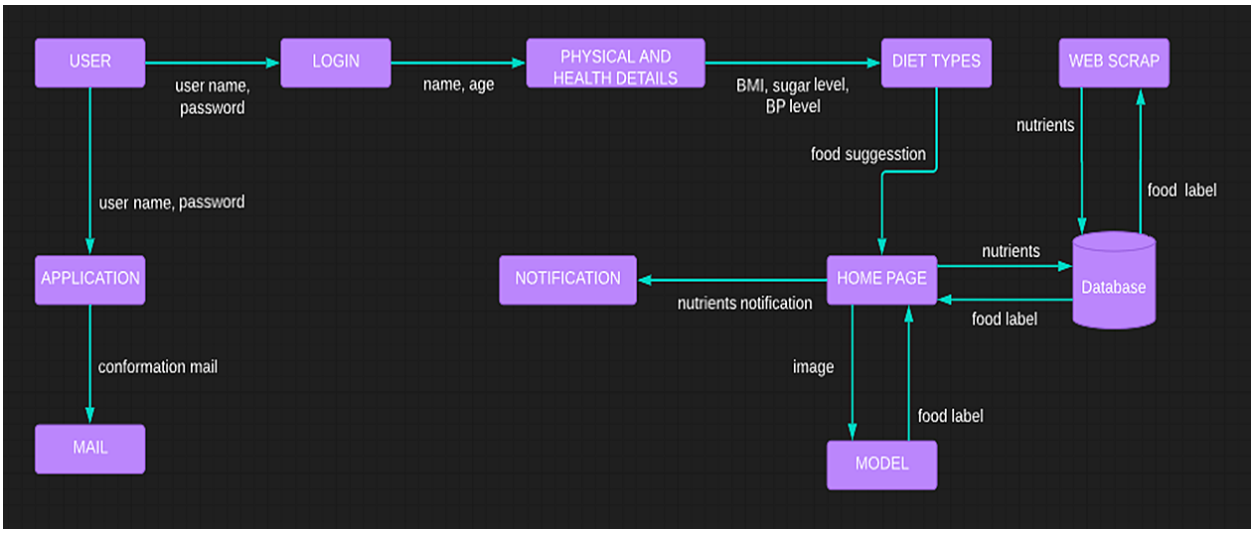
Following are the non-functional requirements of the proposed solution:

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The users should be able to use the application without any difficulties. The interface should be easy to use and understand. The image capture process should be smooth and not tedious
NFR-2	Security	Details of the users and their personal calories calendar should not be disclosed or shared to other users. Privacy of data should be ensured
NFR-3	Reliability	The application should correctly identify the fruits from the captured image and fetch its nutritional value. The count and calculation of the calories should be done accurately
NFR-4	Performance	The application should be built on a highly efficient prediction model such that the results are accurate. It should keep in mind time and space complexity
NFR-5	Availability	The application should be available to its users at all times and should work efficiently. It should not suffer from issues such as application crashes
NFR-6	Scalability	The application should be able to support updates in terms of features and functionality. The system should be built such that it can upgrade using the existing underlying architecture

CHAPTER 5
PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

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



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

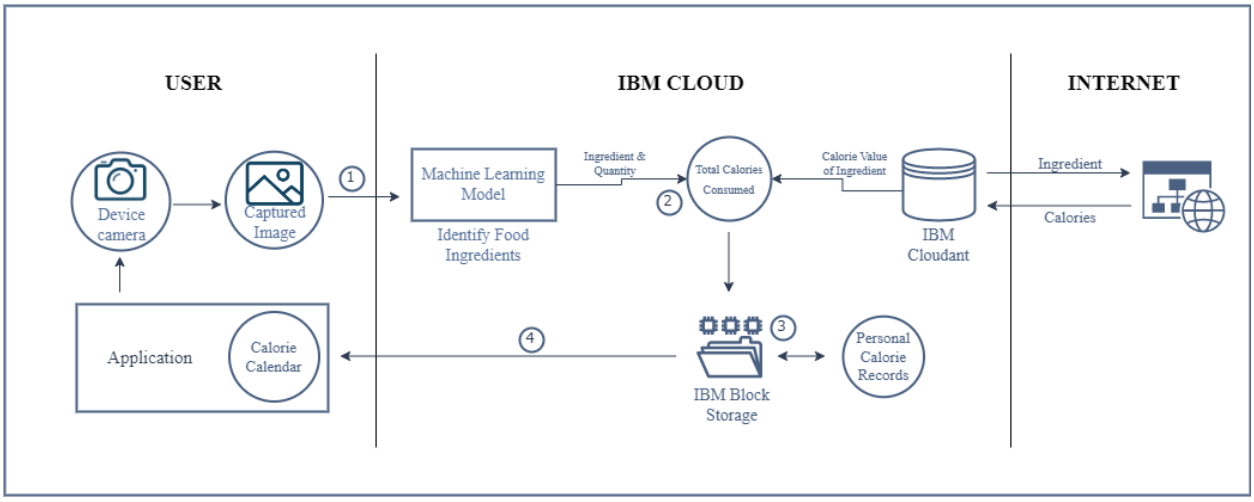
SOLUTION:

Our solution is to build an efficient and intelligent system to analyze the calories in food items by applying a machine learning algorithm which implements classification algorithms to identify the item based on which the calorie is informed.

TECHNICAL ARCHITECTURE:

Technical architecture which is also often referred to as application architecture includes the major components of the system, their relationships, and the contracts that define the interactions between the components. The goal of technical architects is to achieve all the business needs with an application that is optimized for both performance and security.

The Deliverable shall include the architectural diagram as below:



5.3 USER STORIES

User Type	Functional Requirement	User Story Number	User Story/ Task	Acceptance Criteria
User - Fitness Enthusiast	Sign up	USN - 1	As a user, I can create an account	Account is created
	Login	USN - 2	As a user, I can login to my account	Login is validated
	Upload Image	USN - 3	As a user, I can upload an image of what I consume	Image is uploaded
	Calorie Analysis	USN - 4	As a user, I am informed of the calorie value	Calorie content of food is analyzed
	Calorie Calendar	USN - 5	As a user, I can view my calorie consumption history	Calendar is generated with past details

CHAPTER 6
PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	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.	High	Team Lead, Team Member 4
		USN - 2	As a user, I will receive confirmation email once I have registered for the application	High	Team Member 2, Team Member 3
		USN - 3	As a user, I can register for the application through Facebook	Low	Team Member 1, Team Lead
		USN - 4	As a user, I can register for the application through Gmail	Medium	Team Member 3, Team Member 2
Sprint 2	Login	USN - 5	As a user, I can log into the application by entering email & password	High	Team Member 1, Team Lead
	Dashboard	USN - 6	As a user, I can view my profile and update my details	Medium	Team Member 4, Team Lead
		USN - 7	As a user, I can view my personal calorie calendar	High	Team Member 1, Team Member 2
		USN - 8	As a user, I can change my password	High	Team Member 3, Team Member 2
Sprint 3	Image Capturing	USN - 9	As a user, I can capture images of the ingredients I consume	High	Team Member 1, Team Member 4
	Image Processing	USN - 10	In the application, the captured images are processed to label constituent ingredients	High	Team Member 2, Team Member 3
	Data Storage	USN - 11	In the application, the calorie value of different food items are stored using a database	High	Team Leader, Team Member 3
	Calorie Value Computation	USN - 12	As a user, I am informed of the calorie value of the ingredients used	High	Team Leader, Team member 1
Sprint 4	Data Storage	USN - 13	As a user, the details of the calories I've consumed over the course of a day are stored.	High	Team Member 1, Team Lead
	Calorie-Over Consumption Notification	USN - 14	As a user, I am notified if I cross the daily recommended value of calories for a day	High	Team Member 3, Team Member 2
	Diet Plan Specification	USN - 15	As a user, I can specify my target based on which I receive personalized diet plans	High	Team Member 2, Team Member 4

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	19 Nov 2022

CHAPTER 7
CODING AND SOLUTIONING

7.1 FEATURE 1 - INGREDIENT IDENTIFICATION AND CALORIE VALUE ANALYZER

The primary feature of this project is to classify the ingredient present in the images uploaded by the user. Based on the ingredient identified and the quantity of the product, the total calories consumed is calculated.

7.1.1 METHODOLOGY

The dataset used is a collection of images of various food items. The following are the items whose images are present in the dataset:

- 'turnip'
- 'pumpkin'
- 'sweetcorn'
- 'raddish'
- 'ginger'
- 'lemon'
- 'pineapple'
- 'onion'
- 'potato'
- 'spinach'
- 'bean'
- 'jalepeno'
- 'orange'
- 'capsicum'
- 'banana'

- 'peas'
- 'mutton'
- 'carrot'
- 'papaya'
- 'beetroot'
- 'tomato'
- 'mango'
- 'sweetpotato'
- 'garlic'
- 'fish'
- 'eggplant'
- 'cucumber'
- 'chicken'
- 'bitter gourd'
- 'kiwi'
- 'paprika'
- 'bottlegourd'
- 'corn'
- 'grapes'
- 'watermelon'
- 'pomegranate'
- 'broccoli'
- 'pear'
- 'bellpepper'
- 'chillipepper'
- 'egg'
- 'cabbage'
- 'lettuce'
- 'cauliflower'
- 'soy beans'
- 'apple'

The image classification model is built with the MobileNet-V2 algorithm. This is a convolution neural network that is 53 layers deep. This model gave a precision score of 97% on the test set. The constructed machine learning model was used to build the nutrition analyzer application.

The calorie value of each of the above ingredients was stored in a JSON file. Based on the predicted value, the application fetches the appropriate calorie value from the file to display to the user and store in the history of calorie consumption.

```
{
    "turnip" : 27.8,
    "pumpkin" : 27,
    "sweetcorn": 400,
    "raddish" : 22.2,
    "ginger" : 80.0,
    "lemon" : 29.3,
    "pineapple" : 50.0,
    "onion" : 40.0,
    "potato" : 76.9,
    "spinach" : 22.9,
    "bean" : 25.8,
    "jalapeno" : 28,
    "orange" : 47.3,
    "capsicum" : 26.6,
    "banana" : 88.8,
    "peas" : 80.6,
    "mutton" : 294,
    "carrot" : 40.9,
    "papaya" : 43.0,
    "beetroot" : 42.6,
    "tomato" : 18.0,
    "mango" : 60.1,
```

7.2 FEATURE 2 - PERSONAL CALORIE CALENDAR

The application allows the user to upload images of the food ingredients they consume based on which the calorie value is predicted. These values are stored in the backend to make up the history of calorie consumption. The calories consumed in a particular day are added up and stored. This is fetched and displayed in a personal calorie calendar. This also helps users track over-consumption of calories in a day.

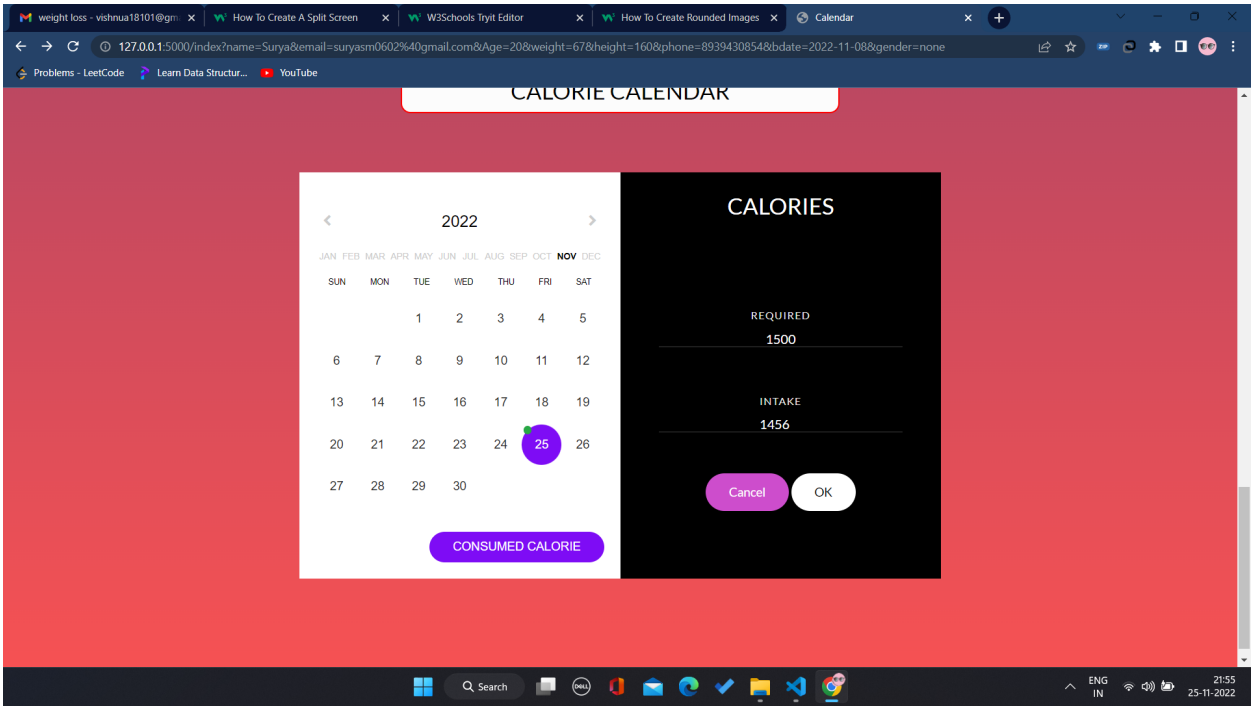


Fig. 7.1 Personal Calorie Calendar

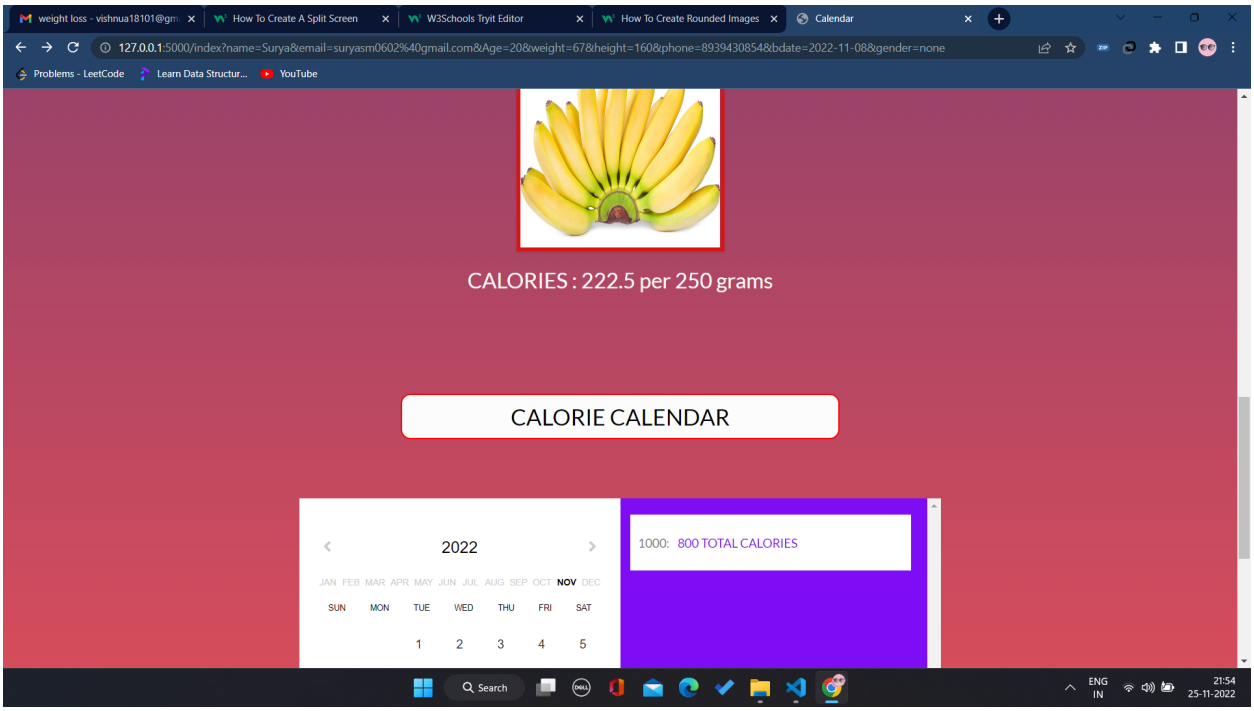


Fig. 7.2 Calorie Analysis for Uploaded Image

CHAPTER 8
TESTING

8.1 TEST CASES

Test Case ID	Feature Type	Component	Test Scenario	Steps to Execute	Test Data	Expected Result	Actual Result	Status
SignUp_TC_01	Functional	Sign Up Page	User can create an account	1. User gives details	Name: Surya Age: 20	Account is created	Work as expected	Pass
SignUp_TC_02	Functional	Sign Up Page	User create account with gmail	1. User uses google account	test@email	Account is created	Work as expected	Pass
Login_TC_01	Functional	Login Page	Returning user can login	1. User gives login details	Name: Surya	Login Success	Work as expected	Pass
Regiser_TC_01	Functional	Register Page	User enters personal details	1. User gives personal details	DOB: 01/01/01	Registration Success	Work as expected	Pass
Dashboard_TC_01	UI	Home Page	User logs into account to view profile	1. Fetch user's details	username: uname	Details fetched	Work as expected	Pass
Dashboard_TC_02	UI	Home Page	User selects to view personal calorie calendar	1. Display calorie calendar	Select calorie calendar tab	Calendar displayed	Work as expected	Pass
CalorieAnalysis_TC_01	Functional	Prediction Page	User uploads an image of a food item	1. Upload image	image of a food item	image uploads	Work as expected	Pass
CalAnalysis	Functional	Prediction Page	Predict the food item	1. Classify the image using	image of food	identifies food	Work as expect	Pass

_TC_02			from image	trained model		item	ed	
CalAnalysis_TC_03	Functional	Prediction Page	Inform calorie value for item identified	1. Classified value of food item	Name of food item	Calorie value of food item	Work as expected	Pass

8.2 USER ACCEPTANCE TESTING

DEFECT ANALYSIS

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

Section	Total Cases	Not Tested	Fail	Pass
Food Item Prediction	10	0	0	10
Calorie Value	10	0	0	10

CHAPTER 9
RESULTS

9.1 PERFORMANCE METRICS

The median efficiency is used to assess each categorization model's effectiveness. The final item will appear in the way it was envisioned. Graphical representations are used to depict information during classification. The percentage of predictions made using the testing dataset is used to gauge accuracy. By dividing the entire number of forecasts even by properly predicted estimates, it is simple to calculate. The difference between actual and anticipated output is used to calculate accuracy.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

Where TP = True Positives, TN = True Negatives, FN = False Negatives and FP = False Positives.

The accuracy for the prediction model built on MobileNet-V2 was 97%.

PRECISION SCORE : 0.9704819987838856
RECALL SCORE : 0.9643605870020965
F1 SCORE : 0.9648662669443673

CHAPTER 10
ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- The developed application can correctly identify the food ingredient that the user has consumed from the image they upload
- The calories consumed can be determined for the food item that is identified
- The calories consumed in a day are stored in the backend to maintain the consumption history of the user
- Users can keep track of the calories they consume and thus watch over their health

DISADVANTAGES

- The application requires huge dataset for a variety of food items

CHAPTER 11
CONCLUSION

A nutrition-analyzer application based on AI has been developed for the benefit of fitness enthusiasts as well as health-conscious people. Users can create an account and maintain a profile for themselves within the application. Users can simply upload images of the food they consume and specify the quantity to the application. The app then identifies the ingredient using a machine learning model. Once the food item has been identifies, the calories of the ingredient are sourced from a database that contains the calorie values of ingredients per 100 g.

Thus, a user will be informed of the amount of calories they have consumed. The application keeps track of the calories consumed within a day by storing and accumulating the details over the period of a day. These data are stored and history of consumption is tracked. The application developed helps people track their dietary intake and pay attention to what they consume. It helps them cut down on food that are causing them harm and also gives a comprehensive outlook to their eating habits that lead to an unhealthy lifestyle.

CHAPTER 12
FUTURE SCOPE

The application has been developed and deployed in the cloud. It has the capability to identify food items and inform their calorie value. This information is displayed to the user as well as stored for future use and to keep track. The application can be improved in many ways. It can include features to analyze the consumption history of the user and generate visual graphs to show progress. It can also suggest diet plans based on the user's goals and tastes. The users can specify their goals based on which a curated set of recipes will be suggested to the user.

CHAPTER 13
APPENDIX

13.1 SOURCE CODE

Registration & Login Form, 'reg_form.html'

```
<!DOCTYPE html>
<html>

   <head>
        <meta charset="ISO-8859-1">
      <title>HOME</title>
      <!-- sign up and login page -->
      <style>
         @charset "ISO-8859-1";

*{
   margin: 0;
   padding: 0;
}

.container,h1
{
   width: 100%;
   height: 100vh;
   font-family: 'Malgun Gothic';
   background: rgb(208, 208, 247);
   /* background-image: url('D:\\IBM\\image.jpg'); */
   background-repeat: no-repeat;
   background-size:1600px 1000px;
   color: #fff;
```

```
    display: flex;
    align-items: center;
    justify-content: center;
}

.card
{
    width: 350px;
    height: 500px;
    box-shadow: 0 0 40px 20px rgba(0, 0, 0, 0.26);
    perspective: 1000px;
}

.inner-box
{
    position:relative;
    width: 100%;
    height: 100%;
    transform-style: preserve-3d;
    transition: transform 1s;
}

.card-front, .card-back
{
    position: absolute;
    width: 100%;
    height: 100%;
    background-position: center;
    background-size: cover;
        background-image: linear-gradient(rgba(0, 0, 100, 0.8),rgba(248, 40, 40, 0.8)),url("E:\Programs\Java Program for Web Developers\Project\workin.jpg");
    padding: 55px;
```

```
    box-sizing: border-box;
    backface-visibility: hidden;
}

.card-back
{
    transform:rotateY(180deg) ;
}

.card h2
{
    font-weight: normal;
    font-size: 24px;
    text-align: center;
    margin-bottom: 20px;
}

.input-box
{
    width: 100%;
    background: transparent;
    border: 1px solid #fff;
    margin: 6px 0;
    height: 32px;
    border-radius: 20px;
    padding: 0 10px;
    box-sizing: border-box;
    outline: none;
    text-align: center;
    color: #fff;
}
```

```
::placeholder
{
   color: #fff;
   font-size: 12px;
}

button
{
   width: 100%;
   background: transparent;
   border: 1px solid #fff;
   margin: 10px 0 ;
   height: 32px;
   font-size: 12px;
   border-radius: 20px;
   padding: 0 10px;
   box-sizing: border-box;
   outline: none;
   color: #fff;
   cursor: pointer;
}

.submit-btn
{
   position: relative;
}

.submit-btn::after
{
   content: '\27a4';
   color: #333;
   line-height: 32px;
```

```
    font-size: 17px;
    height: 32px;
    width: 32px;
    border-radius: 50%;
    background: #fff;
    position: absolute;
    right: -1px;
    top: -1px;
}

span
{
    font-size: 14px;
    margin-left: 10px;
}

card .btn
{
    margin-top: 70px;
}

.card a
{
    color: #fff;
    text-decoration: none;
    display: block;
    text-align: center;
    font-size: 13px;
    margin-top: 8px;
}
        </style>
```

```
</head>

<body>
    <div class="container">
        <div class="card">
            <div class="inner-box" id="card">
                <div class="card-front">
                    <h2>LOGIN</h2>
                    <form action="index.html" method="post">
                        <input type="email" class="input-box" placeholder="Your Email ID"  name="email" required>
                                                <input  type="password"  class="input-box" placeholder="Password" name="password" required>
                                                <button  type="submit"  class="submit-btn" name="login">Login</button>
                        <button type="submit" class="submit-btn" name="login">Login with Google</button>
                        <button type="submit" class="submit-btn" name="login">Login with Facebook</button>
                      <input type="checkbox"><span>Remember Me</span>
                    </form>
                        <button type="button" class="btn" onclick="openRegister()">I'm New Here</button>
                </div>
                <div class="card-back">

                    <h2>SIGN IN</h2>
                    <form action="Register" method="post">
                        <input type="text" class="input-box" placeholder="Your Name" name="username" required>
                        <input type="email" class="input-box" placeholder="Your Email ID" name="email" required>
```

```
                                    <input type="password" class="input-box" placeholder="Password" name="password" required>
                                <input type="number" class="input-box" placeholder="Your Number" name="number" required>
                                    <button type="submit" class="submit-btn" name="sign-in">SIGN IN</button>
                                <input type="checkbox"><span>Remember Me</span>
                            </form>
                            <button type="button" class="btn" onclick="openLogin()">I've an account</button>


                        </div>
                    </div>
                </div>
            </div>

            <script>

                var card=document.getElementById("card");

                function openRegister()
                {
                    card.style.transform="rotateY(-180deg)";
                }

                function openLogin()
                {
                    card.style.transform="rotateY(0deg)";
                }
            </script>
        </body></html>
```

Google Sign Up, 'Google Sign Up.js'

```
import React, { useState, useRef } from "react";
import { useScript } from "./hooks/useScript";
import jwt_decode from "jwt-decode";

const App = () => {
  const googlebuttonref = useRef();
  const [user, setuser] = useState(false);
  const onGoogleSignIn = (user) => {
    let userCred = user.credential;
    let payload = jwt_decode(userCred);
    console.log(payload);
    setuser(payload);
  };
  useScript("https://accounts.google.com/gsi/client", () => {
    window.google.accounts.id.initialize({
      client_id: "1005153530632-
ad4qhfn0oieuchej56k0s4cida2ol92e.apps.googleusercontent.com", // here's your
Google ID
      callback: onGoogleSignIn,
      auto_select: false,
    });
    window.google.accounts.id.renderButton(googlebuttonref.current, {
      size: "medium",
    });
  });
  return (
    <div
      style={{
        display: "flex",
```

```
      justifyContent: "center",
      alignItems: "center",
      height: "100vh",
    }}
  >
    {!user && <div ref={googlebuttonref}></div>}
    {user && (
      <div>
        <h1>{user.name}</h1>
        <img src={user.picture} alt="profile" />
        <p>{user.email}</p>
        <button
          onClick={() => {
            setuser(false);
          }}
        >
          Logout
        </button>
      </div>
    )}    </div>
  );
};

export default App;
```

Google Sign Up, 'Google Sign Up Test.js'

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
```

```
  const linkElement = screen.getByText(/learn react/i);
  expect(linkElement).toBeInTheDocument();
});
```

Profile Display, 'profile.html'

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<style>
body{
    background-image: linear-gradient(rgba(224, 224, 235, 0.8),rgba(69, 69, 74, 0.8));
}

.card {
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
  max-width: 300px;
  margin: auto;
  text-align: center;
  font-family: arial;
}

.title {
  color: grey;
  font-size: 18px;
}

button,button.red {
```

```
  border: none;
  outline: 0;
  display: inline-block;
  padding: 8px;
  color: white;
  background-color: #000;
  text-align: center;
  cursor: pointer;
  width: 100%;
  font-size: 18px;
}

button.red{
    background-color: darkred;
}

a {
  text-decoration: none;
  font-size: 22px;
  color: black;
}

button:hover, a:hover, button.red {
  opacity: 0.7;
}
</style>
</head>
<body>

<h2 style="text-align:center">PROFILE</h2>

<div class="card">
```

```
  <img src="prof.jpg" alt="John" style="width:100%">
  <h1>NAME</h1>
  <p class="title"></p>
  <h3>AGE</h3>

  <div style="margin: 24px 0;">
    <a href="#"><i class="fa fa-whatsapp"></i></a>
    <a href="#"><i class="fa fa-instagram"></i></a>
    <a href="#"><i class="fa fa-facebook"></i></a>
  </div>
  <p><button>Contact</button></p>
  <a href="index.html"><p><button  class="red" >Exit</button></p></a>
</div>

</body>
</html>
```

Personal Calorie Calendar, 'index.html'

```
<!doctype html>
<html lang="en">
<head>
<title>Calendar</title>

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link
```

```
href="https://fonts.googleapis.com/css?family=Lato:300,400,700&display=swap"
rel="stylesheet">
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-
awesome/4.7.0/css/font-awesome.min.css">
<link rel="stylesheet" href="css/style.css">

<style>
.text-{
  border: 2px solid rgb(119, 255, 0);
  border-radius: 12px;
  width: fit-content;
  padding: 10px;
  /* background-color: rgb(224, 227, 227); */
  background-image: linear-gradient(rgba(165, 165, 224, 0.8),rgba(209, 88, 88,
0.8));
}

body{
  /* background-color: #accde0; */
  background-image: linear-gradient(rgba(0, 0, 100, 0.8),rgba(248, 40, 40, 0.8));
}

.w3-container{
  background-color: aliceblue;
}

 h1 {
  /* border: 2px solid red;
  border-radius: 12px; */
  padding: 5px;
  width: fit-content;
  font-family:fantasy;
```

```
  font-size:50px;

  /* background-color: rgb(253, 253, 252); */
}

h2.heading-section{
  text-align: center;
  border: 2px solid red;
  border-radius: 12px;
  padding: 5px;
  /* width: fit-content; */
  background-color: rgb(252, 252, 252);
}
/* h3{
  border: 2px solid red;
  border-radius: 12px;
  padding: 5px;
  width: fit-content;
  background-color: rgb(247, 243, 247);
} */

.mySlides {
    display: none;
  }
img {
      vertical-align: middle;
    }

.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
```

```
}

.prev, .next {
  cursor: pointer;
  position: absolute;
  top: 50%;
  width: auto;
  padding: 16px;
  margin-top: -22px;
  color: rgb(2, 1, 1);
  font-weight: bold;
  font-size: 18px;
  transition: 0.6s ease;
  border-radius: 0 3px 3px 0;
  user-select: none;
}

/* Position the "next button" to the right */
.next {
  right: 0;
  border-radius: 3px 0 0 3px;
}

/* On hover, add a black background color with a little bit see-through */
.prev:hover, .next:hover {
  background-color: rgba(0,0,0,0.8);
}

/* Caption text */
.text {
  color: #f2f2f2;
  font-size: 20px;
```

```
  padding: 8px 12px;
  position: absolute;
  bottom: 8px;
  width: 100%;
  text-align: center;
}

/* Number text (1/3 etc) */
.numbertext {
  color: #f2f2f2;
  font-size: 12px;
  padding: 8px 12px;
  position: absolute;
  top: 0;
}

/* The dots/bullets/indicators */
.dot {
  cursor: pointer;
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}

.active, .dot:hover {
  background-color: #717171;
}
@media only screen and (max-width: 300px) {
```

```
  .prev, .next,.text {font-size: 11px}
}
        </style>

        </head>
        <body>
  <div class="w3-sidebar w3-bar-block w3-border-right" style="display:none"
id="mySidebar">
     <button onclick="w3_close()" class="w3-bar-item w3-large">Close
&amp;times;</button>
     <a href="profile.html" class="w3-bar-item w3-button">PROFILE</a>
     <a href="#" class="w3-bar-item w3-button">SIGN OUT</a>
     <a href="#" class="w3-bar-item w3-button">CHANGE PASSWORD</a>
  </div>

  <div class="w3-teal">
    <button class="w3-button w3-teal w3-xlarge"
onclick="w3_open()">☰</button>
    <div class="w3-container">
     <center><h3>USERNAME</h3></center>
    </div>
  </div>

<br><br>
        <center><h1>CALORIE CALCULATOR</h1></center>
  <br><br>
    <div class="slideshow-container">

       <div class="mySlides">
         <center><img src="apple.jpg" width="450" height="450"></center><br>
       <center><div class="text-"><h4>APPLE : 52/100g</h4></div></center>
       </div>
```

```
<div class="mySlides">
  <center><img src="beef.jpg" width="450" height="450"></center><br>
<center><div class="text-"><h4>BEEF : 250/100g</h4></div></center>
</div>

<div class="mySlides">
  <center><img src="papaya.jpg" width="450" height="450"></center><br>
  <center><div class="text-"><h4>PAPAYA : 32/100g</h4></div></center>
</div>

<div class="mySlides">
  <center><img src="kiwi.jpg" width="450" height="450"></center><br>
  <center><div class="text-"><h4>KIWI : 61/100g</h4></div></center>
</div>

<div class="mySlides">
  <center><img src="lemon.jpg" width="450" height="450"></center><br>
  <center><div class="text-"><h4>LEMON : 29/100g</h4></div></center>
</div>

<div class="mySlides">
  <center><img src="mutton.jpg" width="450" height="450"></center><br>
  <center><div class="text-"><h4>MUTTON :
294/100g</h4></div></center>
</div>

<a class="prev" onclick="plusSlides(-1)">❮</a>
<a class="next" onclick="plusSlides(1)">❯</a>

</div>

<!-- <div style="text-align:center">
```

```
        <span class="dot" onclick="currentSlide(1)"></span>
        <span class="dot" onclick="currentSlide(2)"></span>
        <span class="dot" onclick="currentSlide(3)"></span>
      </div> -->


<script>
 function w3_open() {
  document.getElementById("mySidebar").style.display = "block";
 }

 function w3_close() {
  document.getElementById("mySidebar").style.display = "none";
 }
</script>

<script>
      let slideIndex = 1;
      showSlides(slideIndex);

      function plusSlides(n) {
        showSlides(slideIndex += n);
      }

      function currentSlide(n) {
        showSlides(slideIndex = n);
      }

      function showSlides(n) {
        let i;
        let slides = document.getElementsByClassName("mySlides");
        let dots = document.getElementsByClassName("dot");
        if (n > slides.length) {slideIndex = 1}
```

```
        if (n < 1) {slideIndex = slides.length}
        for (i = 0; i < slides.length; i++) {
          slides[i].style.display = "none";
        }
        for (i = 0; i < dots.length; i++) {
          dots[i].className = dots[i].className.replace(" active", "");
        }
        slides[slideIndex-1].style.display = "block";
        dots[slideIndex-1].className += " active";
      }
</script>

        <section class="ftco-section">
                <div class="container">
                        <div class="row justify-content-center">
                                <div class="col-md-6 text-center mb-5">
                                        <h2 class="heading-section">CALORIE
CALENDAR</h2>
                                </div>
                        </div>
                        <div class="row">
                                <div class="col-md-12">
                                        <div class="content w-100">
                                   <div class="calendar-container">
                                     <div class="calendar">
                                       <div class="year-header">
                                         <span class="left-button fa fa-chevron-left"
id="prev"> </span>
                                         <span class="year" id="label"></span>
                                         <span class="right-button fa fa-chevron-right"
id="next"> </span>
                                       </div>
```

```
    <table class="months-table w-100">
      <tbody>
        <tr class="months-row">
          <td class="month">Jan</td>
          <td class="month">Feb</td>
          <td class="month">Mar</td>
          <td class="month">Apr</td>
          <td class="month">May</td>
          <td class="month">Jun</td>
          <td class="month">Jul</td>
          <td class="month">Aug</td>
          <td class="month">Sep</td>
          <td class="month">Oct</td>
          <td class="month">Nov</td>
          <td class="month">Dec</td>
        </tr>
      </tbody>
    </table>

    <table class="days-table w-100">
      <td class="day">Sun</td>
      <td class="day">Mon</td>
      <td class="day">Tue</td>
      <td class="day">Wed</td>
      <td class="day">Thu</td>
      <td class="day">Fri</td>
      <td class="day">Sat</td>
    </table>
    <div class="frame">
      <table class="dates-table w-100">
<tbody class="tbody">
</tbody>
```

```
                                    </table>
                                </div>
                                <button class="button" id="add-
button">CONSUMED CALORIE</button>
                            </div>
                        </div>
                        <div class="events-container">
                        </div>
                        <div class="dialog" id="dialog">
                            <h2 class="dialog-header">CALORIES</h2>
                            <form class="form" id="form">
                                <div class="form-container" align="center">
                                    <label class="form-label"
id="valueFromMyButton" for="name">REQUIRED</label>
                                    <input class="input" type="text" id="name"
maxlength="36">
                                    <label class="form-label"
id="valueFromMyButton" for="count">INTAKE</label>
                                    <input class="input" type="number" id="count"
min="0" max="1000000" maxlength="7">
                                    <input type="button" value="Cancel"
class="button" id="cancel-button">
                                    <input type="button" value="OK" class="button
button-white" id="ok-button">
                                </div>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
        </section>
```

```
        <script src="js/jquery.min.js"></script>
  <script src="js/popper.js"></script>
  <script src="js/bootstrap.min.js"></script>
  <script src="js/main.js"></script>

        </body>
</html>
```

Registration Form, 'form.html'

```
<!DOCTYPE html>
<html>
  <head>
    <title>Registration Form</title>
    <link href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700"
rel="stylesheet">
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" integrity="sha384-
B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3BlX
eVIU" crossorigin="anonymous">
    <style>
      html, body {
      min-height: 100%;

      }
      body{
        background-image: linear-gradient(rgba(0, 0, 100, 0.8),rgba(248, 40, 40, 0.8));
      }
      body, div, form, input, select, textarea, label {
      padding: 0;
```

```
margin: auto;
outline: none;
font-family: Roboto, Arial, sans-serif;
font-size: 14px;
color: #666;
line-height: 22px;
width: 75%;
}
h1 {
position: absolute;
margin: 0;
font-size: 40px;
color: #fff;
z-index: 2;
line-height: 83px;
}
.testbox {
display: flex;
margin: auto;
height: inherit;
padding: 20px;
}
form {
width: 100%;
padding: 20px;
border-radius: 6px;
background: #fff;
box-shadow: 0 0 8px  #cc7a00;
}
.banner {
position: relative;
height: 100px;
```

```
    background-size: cover;
    display: flex;
    justify-content: center;
    align-items: center;
    text-align: center;
    }
    .banner::after {
    content: "";
    background-color: rgba(0, 0, 0, 0.2);
    position: absolute;
    width: 100%;
    height: 100%;
    }
    input, select, textarea {
    margin-bottom: 10px;
    border: 4px solid rgb(225, 197, 197);
    border-radius: 10px;
    }
    input {
    width: calc(100% - 10px);
    padding: 5px;
    width: 50;
    }
    input[type="date"] {
    padding: 4px 5px;
    }
    textarea {
    width: calc(100% - 12px);
    padding: 5px;
    }
    .item:hover p, .item:hover i, .question:hover p, .question label:hover,
input:hover::placeholder {
```

```
color: #cc7a00;
}
.item input:hover, .item select:hover, .item textarea:hover {
border: 1px solid transparent;
box-shadow: 0 0 3px 0 #cc7a00;
color: #cc7a00;
}
.item {
position: relative;
margin: 10px 0;
}
.item span {
color: red;
}
input[type="date"]::-webkit-inner-spin-button {
display: none;
}
.item i, input[type="date"]::-webkit-calendar-picker-indicator {
position: absolute;
font-size: 20px;
color: #cc7a00;
}
.item i {
right: 1%;
top: 30px;
z-index: 1;
}
[type="date"]::-webkit-calendar-picker-indicator {
right: 1%;
z-index: 2;
opacity: 0;
cursor: pointer;
```

```
}
input[type=radio], input[type=checkbox]  {
display: none;
}
label.radio {
position: relative;
display: inline-block;
margin: 5px 20px 15px 0;
cursor: pointer;
}
.question span {
margin-left: 30px;
}
.question-answer{
  border: 4px solid rgb(225, 197, 197);
  border-radius: 12px;
}

.question-answer label {
display: block;
}
label.radio:before {
content: "";
position: absolute;
left: 0;
width: 17px;
height: 17px;
border-radius: 50%;
border: 2px solid #ccc;
}
input[type=radio]:checked + label:before, label.radio:hover:before {
border: 2px solid #cc7a00;
```

```
}
label.radio:after {
content: "";
position: absolute;
top: 6px;
left: 5px;
width: 8px;
height: 4px;
border: 3px solid #cc7a00;
border-top: none;
border-right: none;
transform: rotate(-45deg);
opacity: 0;
}
input[type=radio]:checked + label:after {
opacity: 1;
}
.btn-block {
margin-top: 10px;
text-align: center;
}
button {
width: 150px;
padding: 10px;
border: none;
border-radius: 5px;
background: #cc7a00;
font-size: 16px;
color: #fff;
cursor: pointer;
}
button:hover {
```

```
    background: #ff9800;
    }
    @media (min-width: 568px) {
    .name-item, .city-item {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-between;
    }
    .name-item input, .name-item div {
    width: calc(50% - 20px);
    }
    .name-item div input {
    width:97%;}
    .name-item div label {
    display:block;
    padding-bottom:5px;
    }
    }
  </style>
</head>
<body>
  <center><div class="testbox">
    <form action="index.html">
      <div class="banner">
        <h1>Registration Form</h1>
      </div>
      <div class="item">
        <label for="name">Name<span>*</span></label>
        <input id="name" type="text" name="name" required/>
      </div>
      <div class="item">
        <label for="email">Email Address<span>*</span></label>
```

```
  <input id="email" type="email" name="email" required/>
</div>
<div class="item">
  <label for="Age">Age<span>*</span></label>
  <input id="Age" type="text" name="Age" required/>
</div>
<div class="item">
  <label for="weight">Weight<span>*</span></label>
  <input id="weight" type="number" name="weight" required/>
</div>
<div class="item">
  <label for="height">Heigth<span>*</span></label>
  <input id="height" type="number" name="zip" required/>
</div>
<div class="item">
  <label for="phone">Phone<span>*</span></label>
  <input id="phone" type="number" name="phone" required/>
</div>
<div class="item">
  <label for="bdate">Date of Birth<span>*</span></label>
  <input id="bdate" type="date" name="bdate" required/>
  <i class="fas fa-calendar-alt"></i>
</div>
<div class="question">
  <label>Gender</label>
  <div class="question-answer">
    <div>
      <input type="radio" value="none" id="radio_1" name="gender"/>
      <label for="radio_1" class="radio"><span>Male</span></label>
    </div>
    <div>
      <input  type="radio" value="none" id="radio_2" name="gender"/>
```

```
            <label for="radio_2" class="radio"><span>Female</span></label>
          </div>
        </div>
      </div>
      <br><br>
      <div class="btn-block">
        <button type="submit">SUBMIT</button>
      </div>
    </form>
  </div></center>
 </body>
</html>
```

Calories.json,

```
{
   "turnip" : 27.8,
   "pumpkin" : 27,
   "sweetcorn": 400,
   "raddish" : 22.2,
   "ginger" : 80.0,
   "lemon" : 29.3,
   "pineapple" : 50.0,
   "onion" : 40.0,
   "potato" : 76.9,
   "spinach" : 22.9,
   "bean" : 25.8,
   "jalapeno" : 28,
   "orange" : 47.3,
   "capsicum" : 26.6,
   "banana" : 88.8,
```

"peas" : 80.6,
"mutton" : 294,
"carrot" : 40.9,
"papaya" : 43.0,
"beetroot" : 42.6,
"tomato" : 18.0,
"mango" : 60.1,
"sweetpotato" : 92.1,
"garlic" : 133,
"fish" : 278,
"eggplant" : 25.1,
"cucumber" : 16,
"chicken" : 218.8,
"bitter gourd" : 34,
"kiwi" : 61.2,
"paprika" : 282,
"bottle gourd" : 15.0,
"corn" : 364,
"grapes" : 68.8,
"watermelon" : 30.0,
"pomegranate" : 82.9,
"broccoli" : 34.0,
"pear" : 56.7,
"bellpepper" : 26.6,
"chilli pepper" : 26.6,
"egg" : 155.0,
"cabbage" : 25.0,
"lettuce" : 15.0,
"cauliflower" : 23.0,
"soy beans" : 173.0,
"apple" : 52

}

Machine Learning Model, 'Fruit_Veg_Classification.ipynb'

```
import numpy as np
import pandas as pd
from pathlib import Path
import os.path
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.preprocessing.image import load_img,img_to_array
train_dir = Path('..content/train')
train_filepaths = list(train_dir.glob(r'**/*.*'))

test_dir = Path('..content/test')
test_filepaths = list(test_dir.glob(r'**/*.*'))

val_dir = Path('D:/fv/fvclass/validation')
val_filepaths = list(test_dir.glob(r'**/*.*'))

def image_processing(filepath):
    """ Create a DataFrame with the filepath and the labels of the pictures
    """

    labels = [str(filepath[i]).split("\\")[-2] \
              for i in range(len(filepath))]

    filepath = pd.Series(filepath, name='Filepath').astype(str)
    labels = pd.Series(labels, name='Label')

    df = pd.concat([filepath, labels], axis=1)
    df = df.sample(frac=1).reset_index(drop = True)
```

```
    return df

train_df = image_processing(train_filepaths)
test_df = image_processing(test_filepaths)
val_df = image_processing(val_filepaths)

print('-- Training set --\n')
print(f'Number of pictures: {train_df.shape[0]}\n')
print(f'Number of different labels: {len(train_df.Label.unique())}\n')
print(f'Labels: {train_df.Label.unique()}')


df_unique = train_df.copy().drop_duplicates(subset=["Label"]).reset_index()

fig, axes = plt.subplots(nrows=5, ncols=9, figsize=(10, 12),
                         subplot_kw={'xticks': [], 'yticks': []})

for i, ax in enumerate(axes.flat):
    ax.imshow(plt.imread(df_unique.Filepath[i]))
    ax.set_title(df_unique.Label[i], fontsize = 12)
plt.tight_layout(pad=0.5)
plt.show()

train_generator = tf.keras.preprocessing.image.ImageDataGenerator(
    preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input
)

test_generator = tf.keras.preprocessing.image.ImageDataGenerator(
    preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input
)
```

```
train_images = train_generator.flow_from_dataframe(
    dataframe=train_df,
    x_col='Filepath',
    y_col='Label',
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=True,
    seed=0,
    rotation_range=30,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest"
)

val_images = train_generator.flow_from_dataframe(
    dataframe=val_df,
    x_col='Filepath',
    y_col='Label',
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=True,
    seed=0,
```

```
    rotation_range=30,
    zoom_range=0.15,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.15,
    horizontal_flip=True,
    fill_mode="nearest"
)

test_images = test_generator.flow_from_dataframe(
    dataframe=test_df,
    x_col='Filepath',
    y_col='Label',
    target_size=(224, 224),
    color_mode='rgb',
    class_mode='categorical',
    batch_size=32,
    shuffle=False
)

pretrained_model = tf.keras.applications.MobileNetV2(
    input_shape=(224, 224, 3),
    include_top=False,
    weights='imagenet',
    pooling='avg'
)
pretrained_model.trainable = False

inputs = pretrained_model.input


x = tf.keras.layers.Dense(128, activation='relu')(pretrained_model.output)
```

```
x = tf.keras.layers.Dense(128, activation='relu')(x)

outputs = tf.keras.layers.Dense(46, activation='softmax')(x)

model = tf.keras.Model(inputs=inputs, outputs=outputs)

model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
history = model.fit(
    train_images,
    validation_data=val_images,
    batch_size = 32,
    epochs=5,
    callbacks=[
        tf.keras.callbacks.EarlyStopping(
            monitor='val_loss',
            patience=2,
            restore_best_weights=True
        )
    ]
)

pred = model.predict(test_images)
pred = np.argmax(pred,axis=1)
labels = (train_images.class_indices)
labels = dict((v,k) for k,v in labels.items())
pred1 = [labels[k] for k in pred]
```

```
def output(location):
    img=load_img(location,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)
    y = " ".join(str(x) for x in y_class)
    y = int(y)
    res = labels[y]
    return res

model.save('FV.h5')

from sklearn.metrics import
confusion_matrix,precision_score,recall_score,f1_score
cm = confusion_matrix(test_images.classes, pred)

ps = precision_score(test_images.classes, pred, average='weighted')
print("PRECISION SCORE : ",ps)
rs = recall_score(test_images.classes, pred, average='weighted')
print("RECALL SCORE : ",rs)
fs = f1_score(test_images.classes, pred, average='weighted')
print("F1 SCORE : ",fs)
```

Confusion Matrix:

```
array([[ 8,  0,  0, ...,  0,  0,  0],
       [ 0,  7,  0, ...,  0,  0,  0],
       [ 0,  0, 10, ...,  0,  0,  0],
       ...,
       [ 0,  0,  0, ..., 10,  0,  0],
       [ 0,  0,  0, ...,  0, 10,  0],
       [ 0,  0,  0, ...,  0,  0, 10]], dtype=int64)
```

Accuracy and Evaluation:

```
PRECISION SCORE :  0.9704819987838856
RECALL SCORE :  0.9643605870020965
F1 SCORE :  0.9648662669443673
```

Flask App, 'app.py'

```
from flask import *

import os

from PIL import Image

from tensorflow.keras.utils import load_img,img_to_array

import numpy as np

from keras.models import load_model

import requests

from bs4 import BeautifulSoup

from werkzeug.utils import secure_filename

app = Flask(__name__)

model = load_model('FV.h5')

labels = {0: 'apple', 1: 'banana', 2: 'bean', 3: 'beetroot', 4: 'bell pepper', 5: 'bitter gourd', 6: 'bottle gourd',7: 'broccoli', 8: 'cabbage', 9: 'capsicum', 10: 'carrot', 11: 'cauliflower', 12: 'chicken', 13: 'chilli pepper', 14: 'corn', 15: 'cucumber', 16: 'egg', 17: 'eggplant', 18: 'fish', 19: 'garlic', 20: 'ginger', 21: 'grapes', 22: 'jalepeno', 23: 'kiwi', 24: 'lemon', 25: 'lettuce', 26: 'mango', 27: 'mutton', 28: 'onion', 29: 'orange', 30: 'papaya',  31: 'paprika', 32: 'pear', 33: 'peas', 34: 'pineapple', 35: 'pomegranate', 36: 'potato', 37: 'pumpkin', 38: 'raddish', 39: 'soy beans', 40: 'spinach', 41:
```

'sweetcorn', 42: 'sweetpotato', 43: 'tomato', 44: 'turnip', 45: 'watermelon'}

fruits = ['Apple','Banana','Bello Pepper','Chilli Pepper','Grapes','Jalepeno','Kiwi','Lemon','Mango','Orange','Paprika','Pear','Pineapple','Pomegranate','Watermelon','Papaya']

vegetables = ['Beetroot','Cabbage','Capsicum','Carrot','Cauliflower','Corn','Cucumber','Eggplant','Ginger','Lettuce','Onion','Peas','Potato','Raddish','Soy Beans','Spinach','Sweetcorn','Sweetpotato','Tomato','Turnip','Bean','Bitter Gourd','Bottle Gourd','Broccoli','Pumpkin']

non_vegetables=['Chicken', 'Egg', 'Fish', 'Mutton']

original = ['adidas','alfaRomeo','Amazon','Apple','audi','bmw','chevrolet','citroen','Coca-Cola','dacia','Facebook','ferrari','fiat','ford','Google','honda','hyundai','jaguar','jeep','McDonald_s','NIKE','puma','starbucks']

fake = ['fake-logo-adidas','fake-logo-apple','fake-logo-mcdonalds','fake-logo-nike','fake-logo-puma','fake-logo-starbucks']

classes = ['adidas','alfaRomeo','Amazon','Apple','audi','bmw','chevrolet','citroen','Coca-Cola','dacia','Facebook','fake-logo-adidas','fake-logo-apple','fake-logo-mcdonalds','fake-logo-nike','fake-logo-puma','fake-logo-starbucks','ferrari','fiat','ford','Google','honda','hyundai','jaguar','jeep','McDonald_s','NIKE','puma','starbucks']

path = 'D:/IBM@2/'

```
def fetch_calories(prediction):
    try:
        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
    except Exception as e:
        print("Can't able to fetch the Calories")
        print(e)


def image_processing(img):
    imgpath = os.path.join(path,img)
    img = Image.open(imgpath).resize((250,250))
    img=load_img(imgpath,target_size=(224,224,3))
    try:
        img=img_to_array(img)
        img=img/255
        img=np.expand_dims(img,[0])
        # log_img=cv2.resize(log_img,(50,50))
        # image = np.array(log_img).flatten()
```

```
        # # data.append(image)
    except Exception as e:
        pass
    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 processed_img(img_path):
#     img = Image.open(img_file).resize((250,250))
#     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()
@app.route('/')
@app.route('/reg_form')
def reg_form():
    return render_template('reg_form.html')

@app.route('/form', methods = ['GET', 'POST'])
def form():
    return render_template('form.html')

@app.route('/index', methods = ['GET', 'POST'])
def index():
    return render_template('index.html')

@app.route('/profile', methods = ['GET', 'POST'])
def profile():
    return render_template('profile.html')
@app.route('/split', methods = ['GET', 'POST'])
```

```
def split():
    return render_template('split.html')


@app.route('/predict', methods=['GET', 'POST'])
def upload():
    multimg = []
    tot = []
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']
        file_path = secure_filename(f.filename)
        value = request.form.get("quantity")
        number = int(value)
        f.save(file_path)
        # Make prediction
        result = image_processing(file_path)
        print(result)
        if result in vegetables:
                print("Category : Vegetables")
        elif result in non_vegetables:
                print("Category : Non-Vegetables")
```

```
        else:
            print("Category : Fruit")
            print("Predicted : "+result)

        spl = fetch_calories(result)
        spl = spl.split()
        val = int(spl[0])
        g1 = val/100

        sum = number * g1

        cal = "CALORIES : " + str(sum) + " per " + str(number) + " grams"
        os.remove(file_path)
        print(cal)
        return cal
    return None

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

Diet Plans, 'split.html'

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  font-family: Georgia, serif;
  color: rgb(13, 12, 12);
  font-size: 20px;
}

.split {
  height: 96%;
  width: 33%;
  position: fixed;
  z-index: 1;
  top: 0;
  overflow-x: scroll;
  padding-top: 20px;
}
```

```
.left {
  left: 0;
  /* background-image: linear-gradient(rgba(38, 38, 222, 0.8),rgba(241, 37, 37, 0.8)); */
  background-color: #FFFFFF;
background-image: linear-gradient(180deg, #FFFFFF 0%, #6284FF 50%, #FF0000 100%);

  border: 5px solid rgb(16, 239, 42);
}

.middle{
    left: 513.5px;
    /* background-image: linear-gradient(rgba(241, 37, 37, 0.8),rgba(38, 38, 222, 0.8)); */
    background-color: #FA8BFF;
background-image: linear-gradient(45deg, #FA8BFF 0%, #2BD2FF 52%, #2BFF88 90%);


   border: 5px solid rgb(16, 239, 42);
```

```
}

.right {
  right: 0;
  /* background-image: linear-gradient(rgba(38, 38, 222, 0.8),rgba(241, 37, 37, 0.8)); */
  background-color: #52ACFF;
background-image: linear-gradient(180deg, #52ACFF 25%, #FFE32C 100%);

  border: 5px solid rgb(16, 239, 42);
}

.centered {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  text-align: center;
}
```

```
.centered img {
  width: 150px;
  border-radius: 50%;
}
</style>
</head>
<body>

<div class="split left" >
  <div class="centered">
    <img src="">
    <br><br><br><br><br><br><br><br> <br><br><br><br><br><br><br><br>
    <h2>WEIGHT GAIN</h2>
    <ul>
      <li>Day 1: 1514 cal</li>
      <ul type="square">
      <li>Breakfast (387)</li>
      <li>A.M. Snack (190)</li>
      <li>Lunch (325)</li>
      <li>P.M. Snack (105)</li>
```

Dinner (507)

Day 2: 1513 cal

<ul type="square">

Breakfast (387)

A.M. Snack (192)

Lunch (344)

P.M. Snack (95)

Dinner (495)

Day 3: 1502 cal

<ul type="square">

Breakfast (387)

A.M. Snack (95)

Lunch (344)

P.M. Snack (201)

Dinner (475)

Day 4: 1524 cal

<ul type="square">

Breakfast (393)

A.M. Snack (78)

Lunch (344)

P.M. Snack (188)

Dinner (521)

Day 5: 1487 cal

<ul type="square">

Breakfast (287)

A.M. Snack (192)

Lunch (344)

P.M. Snack (210)

Dinner (454)

Day 6: 1496 cal

<ul type="square">

Breakfast (393)

A.M. Snack (200)

Lunch (360)

P.M. Snack (78)

```
      <li>Dinner (465)</li>
      </ul>
      <li>Day 7: 1501 cal</li>
      <ul type="square">
      <li>Breakfast (285)</li>
      <li>A.M. Snack (95)</li>
      <li>Lunch (345)</li>
      <li>P.M. Snack (220)</li>
      <li>Dinner (556)</li>
      </ul>
      </ul>
  </div>
</div>


<div class="split middle">
  <div class="centered">
    <img src="" alt=""><br>
    <br><br><br><br><br><br><br><br> <br><br><br><br><br><br><br><br>
```

<h2>WEIGHT LOSS</h2>

Day 1: 1514 cal

<ul type="square">

Breakfast (387)

A.M. Snack (190)

Lunch (325)

P.M. Snack (105)

Dinner (507)

Day 2: 1513 cal

<ul type="square">

Breakfast (387)

A.M. Snack (192)

Lunch (344)

P.M. Snack (95)

Dinner (495)

Day 3: 1502 cal

<ul type="square">

Breakfast (387)

A.M. Snack (95)

Lunch (344)

P.M. Snack (201)

Dinner (475)

Day 4: 1524 cal

<ul type="square">

Breakfast (393)

A.M. Snack (78)

Lunch (344)

P.M. Snack (188)

Dinner (521)

Day 5: 1487 cal

<ul type="square">

Breakfast (287)

A.M. Snack (192)

Lunch (344)

P.M. Snack (210)

```
<li>Dinner (454)</li>
</ul>
<li>Day 6: 1496 cal</li>
<ul type="square">
<li>Breakfast (393)</li>
<li>A.M. Snack (200)</li>
<li>Lunch (360)</li>
<li>P.M. Snack (78)</li>
<li>Dinner (465)</li>
</ul>
<li>Day 7: 1501 cal</li>
<ul type="square">
<li>Breakfast (285)</li>
<li>A.M. Snack (95)</li>
<li>Lunch (345)</li>
<li>P.M. Snack (220)</li>
<li>Dinner (556)</li>
</ul>
</ul>
</div>
```

```
</div>



<div class="split right">
  <div class="centered">
    <img src="" alt="">
    <br><br><br><br><br><br><br><br> <br><br><br><br><br><br><br><br>
    <h2>DIABETICS</h2>
<ul>
<li>Day 1: 1195 cal</li>
<ul type="square">
<li>Breakfast (281)</li>
<li>A.M. Snack (66)</li>
<li>Lunch (325)</li>
<li>P.M. Snack (95)</li>
<li>Dinner (428)</li>
</ul>
<li>Day 2: 1203 cal</li>
<ul type="square">
```

Breakfast (276)

A.M. Snack (77)

Lunch (344)

P.M. Snack (95)

Dinner (411)

Day 3: 1195 cal

<ul type="square">

Breakfast (276)

A.M. Snack (30)

Lunch (344)

P.M. Snack (62)

Dinner (483)

Day 4: 1209 cal

<ul type="square">

Breakfast (276)

A.M. Snack (77)

Lunch (344)

P.M. Snack (62)

Dinner (450)

Day 5: 1110 cal

<ul type="square">

Breakfast (276)

A.M. Snack (30)

Lunch (344)

P.M. Snack (103)

Dinner (457)

Day 6: 1206 cal

<ul type="square">

Breakfast (276)

A.M. Snack (129)

Lunch (275)

P.M. Snack (62)

Dinner (464)

Day 7: 1204 cal

<ul type="square">

```
<li>Breakfast (349)</li>

<li>A.M. Snack (62)</li>

<li>Lunch (254)</li>

<li>P.M. Snack (95)</li>

<li>Dinner (444)</li>

</ul>

</ul>

  </div>

</div>


</body>

</html>
```

13.2 GITHUB AND PROJECT DEMO LINK

GitHub Link:
https://github.com/IBM-EPBL/IBM-Project-2492-1658472818.git

Project Demo Link:
https://drive.google.com/drive/folders/1Sr7FL24jXvHoSzNzgQNIdtK_ZouEaY_w?usp=share_link