AUTOMATED FITNESS TRACKER

Major project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

RUDRESH SHARMA (181372)

KUNAL KANSAL (181357)

UNDER THE SUPERVISION OF

DR. RAJINDER SANDHU

Assistant Professor (SG)

to

Department of Computer Science and Information Technology

Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

DECLARATION

I hereby declare that, this project has been done by me under the supervision of (Dr. Rajinder Sandhu, Assistant Professor (SG)), Jaypee University of Information Technology. I also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:

Dr. Rajinder Sandhu

Assistant Professor (SG)

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology

Submitted by:

Rudresh Sharma

(181372)

Kunal Kansal

(181357)

Computer Science & Engineering Department

Jaypee University of Information Technology

CERTIFICATE

This is to certify that the work which is being presented in the project report titled "Automated Fitness Tracker" in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by "Rudresh Sharma (181372), Kunal Kansal (181357)" during the period from January 2022 to May 2022 under the supervision of "Dr. Rajinder Sandhu", Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Rudresh Sharma
(181372)

Kunal Kansal
(181357)

The above statement made is correct to the best of my knowledge.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Dr. Rajinder Sandhu)
Assistant Professor (SG)
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat, 173234

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

I really grateful and wish my profound my indebtedness to Supervisor Dr. Rajinder Sandhu, Assistant Professor (SG), Department of CSE Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of my supervisor in the field of "Data Science" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to Dr. Rajinder Sandhu, Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straight forwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Rudresh Sharma, Kunal Kansal

ABSTRACT

Due to COVID 19 pandemic everyone is so concerned about the fact “Health is wealth” as a person’s health is their only valuable asset. Everyone has started doing physical, mental and breathing exercises to be fit but one’s goal towards fitness starts with what they eat. Most of them doesn’t know about keeping a track of the food they are eating and the workouts they are doing to burn calories. As, they are not able to distinguish between healthy and unhealthy food items they are consuming and are not able to see results of their hard work and patience. So, we decided to make this project on “AUTOMATED FITNESS TRACKER” for such people to keep a track of the food they are consuming and the exercises they are performing. For this they have to know the proper amount of protein, fibre, carbohydrates and fats they are eating in their daily meals and the calories exhausted in doing a particular workout. As a proper balance of these substances in one’s body and some good daily workout plans results in good health and is able to fight diseases very efficiently.

In our project we’ve made a sincere attempt to make a calculator for counting our daily calories and show user, the amount of carbohydrates, fats, proteins and fibre they have taken in their meals. User can track out calories burned in their workout plans and are also able to make changes and add new workout and diet plans. Our project is based on the python development environment and code is easy and clear to be understood by anyone. We have selected and refined a huge dataset of food items and workout plans using various data refining and data cleaning processes by using Pandas. We also provided the admin to modify the dataset so that new food items and workouts can be constantly added in the dataset as per the user requirement. Our focus has been constantly on keeping the functioning of the application very simple and accurate so that anyone can use our application easily without any challenge. Various python modules have been used to support the backend of the application and is merged with HTML, CSS and Django for its frontend designing and web implementation.

Table of Contents

Chapter Number **Page No.**

Chapter 1 INTRODUCTION 1
Introduction 1
1.2 Objective of Automated Fitness Tracker 2
1.3 Motivation of Automated Fitness Tracker 2
1.4 Language Used 3
1.5 Technical Requirements 4
1.6 Deliverables of Automated Fitness Tracker 4
Chapter 2 AUTOMATED FITNESS TRACKER SDLC 6
2.1 Feasibility Study on Automated Fitness Tracker 6
2.2 Requirements on Automated Fitness Tracker 6
2.2.1 Functional Requirements: 6
2.2.2 Non-Functional Requirements: 7
2.3 Use Case Diagram of Automated Fitness Tracker 7
2.4 DFD Diagram of Automated Fitness Tracker 9
2.5 State Transition Diagram of Automated Fitness Tracker 11
Chapter 3 LITERATURE SURVEY 12
3.1 HealthifyMe Android Application 12
3.2 JEFIT Android Application: 13
3.3 The Smart Kitchen Concept 14
3.4 CalculNutri Project 16
3.5 The Caloric Calculator 17
Chapter 4 IMPLEMENTATION OF THE MINOR PROJECT 19
4.1 Data Set Used in the Automated Fitness Tracker 19
4.1.1 Data Cleaning 20
4.2 Data Set Features 22
4.2.1 Types of Data Set 22
4.2.2 Number of Attributes, Field, Description of dataset 23
4.3 Design of the problem statement 25
4.4 Algorithm/ Pseudo code of the Automated Fitness Tracker 25
4.4.1 Algorithm: 25
4.4.2 Pseudo Code: 25
4.5 Flow Graph of the Automated Fitness Tracker 26

4.6 About Chatbot 27
4.6.1 Libraries used in our chatbot 27
4.6.2 Working of the chatbot 28
4.6.3 Algorithm/ Pseudo code for the Chatbot 29
4.7 Screen shots of the various stages of Automated Fitness Tracker 29
4.7.1 Data Cleaning and increasing accuracy: 29
4.7.3 Training Model: 33
4.7.4 Deploying the real-time model: 34
4.7.5 Running Application snippets: 36
4.8 Major Modules used in this Automated Fitness Tracker 40
Chapter 5 CONCLUSIONS 41
5.1 Discussions on the results achieved 41
5.1.1 Login Page: 41
5.1.2 Web Application: 42
5.2 Machine learning algorithm used in Chatbot: 43
5.3 Application of Automated Fitness Tracker 44
5.4 Limitations of Automated Fitness Tracker 45
5.5 Future of Automated Fitness Tracker 45
REFERENCES 46
1. Websites: 46
2) Journals: 47
3) Articles: 47
4) Datasets: 48
APPENDICES 49

Table of figures:

Figure Name **Page No.**

Figure 1-1 Python 3
Figure 1-2 Health tracker display 5
Figure 2-1Web Application Use Case Diagram of User 7
Figure 2-2 Web Application Use Case Diagram of Admin 8
Figure 2-3 Data Use Case Diagram of automated calorie calculation 8
Figure 2-4 DFD Diagram for automated calorie calculation 10
Figure 2-5 State Transition Diagram for automated calorie calculation 11
Figure 3-1 HealthifyMe app photo 12
Figure 3-2 JEFIT Application 13
Figure 3-3 JEFIT interface 13
Figure 3-4 Smart kitchen 14
Figure 3-5 Nutritional score table 16
Figure 3-6 Nutrition analysis 16
Figure -4-1 Database in CSV format (1) 19
Figure 4-2 Database in CSV format (2) 20
Figure 4-3 Exercise dataset in csv format 20
Figure 4-4 Processing backend view 21
Figure 4-5 Cleaned Dataset 21
Figure 4-6 Cleaned exercise dataset 22
Figure 4-7 JSON data format 23
Figure 4-8 Algorithm of the given project 25
Figure 4-9 Flow graph of the project problem 26
Figure 4-10 Chatbot hello 27
Figure 4-11 Algorithm for chatbot 29
Figure 4-12 Data Cleaning and using Fuzzy Wuzzy Code (1) 30
Figure 4-13 Data Cleaning and using Fuzzy Wuzzy Code (2) 30
Figure 4-14 Output of the existing data set 31
Figure 4-15 Backend output of FuzzyWuzzy module 31
Figure 4-16 Data representation of exercise dataset 31
Figure 4-17 Graphical Data Representation 32
Figure 4-18 Login page design in HTML 32

Figure 4-19 Registering account design in HTML 32
Figure 4-20 Creating account 33
Figure 4-21 Figure 4 20 Login page display 33
Figure 4-22 Figure 4 19 Sign in page display 33
Figure 4-23 Training models backend 33
Figure 4-24 py.manage file 34
Figure 4-25 models.py file 34
Figure 4-26 decoators.py file 35
Figure 4-27 view.py file 35
Figure 4-28 PowerShell working 36
Figure 4-29 Main page (i) 36
Figure 4-30 Main page (ii) 36
Figure 4-31 Nutrition Table (i) 37
Figure 4-32 Nutrition Table (ii) 37
Figure 4-33 Addition of meals 38
Figure 4-34 Chatbot 38
Figure 4-35 Figure 4 34 Calorie Burned Tracker 39
Figure 4-36 Searching food item 39
Figure 4-37 Workout tracker interface 39
Figure 5-1 Sign up 41
Figure 5-2 Login 41
Figure 5-3 Create a new account 41
Figure 5-4 Running Application 42
Figure 5-5 Various food items 43
Figure 6-1 decorators.py file 49
Figure 6-2 urls.py file 50
Figure 6-3 forms.py file 51
Figure 6-4 manage.py file 51

Tables:

Table Name **Page No.**

Table 1 Key for DFD diagram given above 10
Table 2 Data set attributes 23
Table 3 Exercises targeting particular muscle 24
Table 4 Modules used 40

Chapter 1 INTRODUCTION

Introduction

In this chapter we will learn about the objective of our major project and the environment used to develop our project. Here the basic frame of the project will be discussed.

As, we all our aware of the fact that our country is going under a serious pandemic and in this situation, we are constantly being told by our administration and health workers to have a strong immunity and be healthy. In this hour of need when all of the platforms are going online everyone has started to work upon their health and fitness. They need to track the constant activity of their meals and exercises so some free and accurate fitness calculators should be provided to our people so that they can track down their meals and activities and move a step closer towards their fitness. The calorie tracker must be able to tell the user about every macro nutrient (e.g., proteins, carbohydrates, fats, fibre, etc.) in their meal and then let the user decide that whether it's healthy or not

There also occurs a main feature function nowadays "Chatbot", many here ask the question that what is a chatbot and what are its various functionalities, we can say that a chatbot is a robotic version of human helper as when we go in a shop there our various workers helping us in getting our desired stuff similarly, in a computer application a chatbot is the one who helps you in running the application, talks about the variety of features and guides you anywhere you're stuck. A chatbot is created to reduce the wastage of human energy in some basic and common queries which a lot of people have, e.g., in a shopping application query related to delivery and return etc.

The Fitness Tracker tracks your all-daily activities and meals which shows us that how much calories we have taken and exhausted in a day. The calorie measures should be correct and precise in order to be a better substitute than other applications in the market. We tend to incorporate our fitness tracker with functionalities like sleep, water intake, stress and heartbeat measures.

So, in this project we have tried to give our most so that this can be make possible and everyone should be able to track down their daily intake amount and quality of meals and the calories burned during a day. We have used various data cleaning and refining processes for our data according to our needs. Python pandas, NumPy, SciPy and other libraries are used to

create the data frames and to improve the accuracy of the data set. Python FuzzyWuzzy library is used to improve the search results and to store the desired result. We have used python module Keras [6] and TensorFlow [7] for testing and training our chatbot and nltk [5] for making it more realistic. PythonSQL is used to manage our database and html.py, css.py and Django are used to design, style and run the application over the server.

1.2 Objective of Automated Fitness Tracker

Automated Fitness Tracker is a utility used to calculate energy and tells the quantity of carbs, proteins, fibres and fat in their meals and it also tells us the number of calories burned. It facilitates the customers to track down their food and to discover what they're ingesting and what kind of food they're ingesting, i.e., it offers the excellent and exact amount evaluation of our food. It tells about the daily eating habits of an individual and guides them toward a healthful lifestyle. This accurately describes the number of calories one burns during a workout. This application is able to make the quantity analysis and to check the grade of the food items in our meal. And by the addition of chatbot it makes people more attracted towards this and they find it easier and more useful than the other application which have paid subscriptions.

The knowledge of workouts and diet plans are only limited to the gym trainers, dieticians and professional athletes and coaches but not to the public so here we decided to share this knowledge to everyone whether they are professional or not. This is a field where nowadays most of the youth and as well as elders are also attracted to move towards a healthy lifestyle. By this application everyone will be able to get to know about the distribution of the macro nutrients and the muscle tear down and recovery process to prevent any injury and bad eating habits.

1.3 Motivation of Automated Fitness Tracker

There are already many prebuilt health calculators in the market and they are achieving a great success but many of them have inaccurate results and the one’s having accurate results and other premium functionalities have paid subscriptions. We want to provide our users with a free health calculator with all the latest functionalities which give accurate results and is easy to access so that everyone can have proper training and diet plans, and be fit and healthy without paying unnecessarily for subscriptions to the other applications. Also, everyone

should be focused on their health so that we can move towards a healthy India which will make our county grow in every sector as it says a healthy country is a developed country.

This project is completed with full devotion so that in this distress situation of COVID 19 everyone should take care of their health and be able to seek the correct food items in their diet plan which will help them build immunity and make their body able to fight with this disease. Many families have been ruined by this disease so we tried to contribute a little for us all to keep a track of our daily habits and to figure out the faults in our daily diet and workout plans. This situation made us more motivated towards our goal to provide everyone with this project as soon as possible. We wish that our project may help everyone who desires to use such an application and provide them with good knowledge about the food items we should consume and the workouts we can plan for being strong and healthy enough in this distress situation.

1.4 Language Used

Many languages had been used in the designing of this project undertaking beneath the python work environment.

Figure 1-1 Python

Python work environment: It is a virtual environment used to create python projects by providing them with an isolated unique space for working.

Python: It is a HLL (High Level Language) used for object-oriented programming. It is one of the vastest languages aside from JAVA and have feature extraordinarily libraries and modules which makes its code simple and clean to be comprehensible even by a layman [1].

Rest of the languages used on this undertaking are:

i. PythonSQL (Structured Query Language): SQL language for working our data set;
ii. JSON: Retrieving data for the chatbot [9];
iii. Hypertext Markup Language (HTML): Used for frontend designing;
iv. Cascading Style Sheets (CSS): Used for styling the frontend;
v. Django web framework: Helps in strolling the network application;
vi. Bootstrap: It creates the web user interface.

1.5 Technical Requirements

To run this project, one should have a machine specification of:

i. Python latest version – PyCharm 1.19 -> PyCharm 2.31
ii. Django latest modules – Django-Filters
iii. PC of RAM – 6GB
iv. Window PowerShell ISE

Whereas, there is no need of the technical hardware requirements as it is a software-based project.

1.6 Deliverables of Automated Fitness Tracker

The project delivers an application using a web interface to reflect its output. It tells us about our daily calorie limit, total calories consumed and calories left. Search bar for searching different items and the total number of items consumed in a day. It also depicts the number of calories burned by performing some specific workouts so that the user can track both the calorie intake and calories exhausted during a day which helps in the calculation of weight loss or gain.

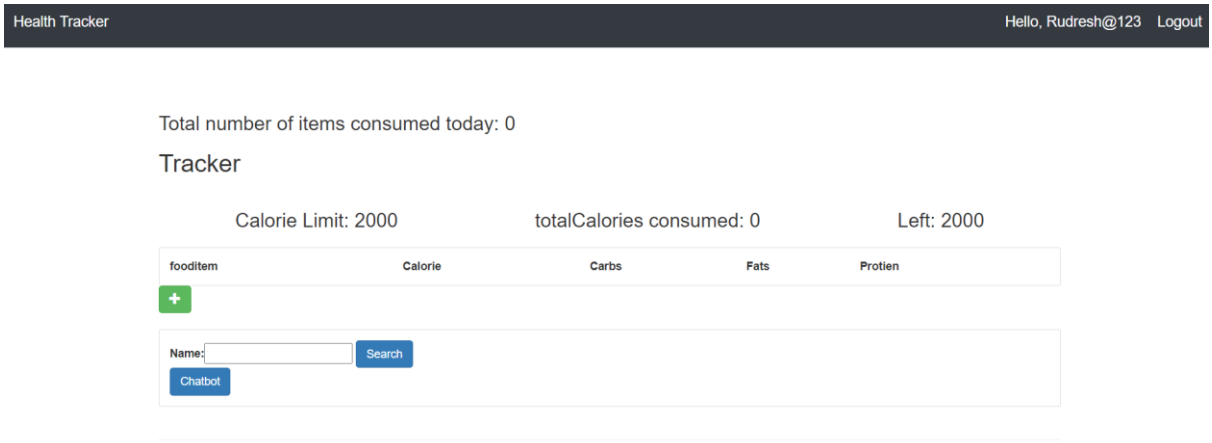


Figure 1-2 Health tracker display

Chapter 2 AUTOMATED FITNESS TRACKER SDLC

2.1 Feasibility Study on Automated Fitness Tracker

There has not been just an idea of the project but some researches have also been done to assess the success rate of this project. First of all, a basic assessment was taken to understand the need of this project with respect to stakeholders. As, there are other advanced calculators present too, so we need to know where does our prototype stands among them. But this project has the potential to grow more than them in a short span of time as we have already known on which points to focus and what improvements they need. The main objective is to provide those upgraded functionalities to the user without paying for the huge subscriptions.

The market for this kind of product is going quite vast just because everyone is moving towards healthy lifestyles and for that they need such applications which doesn't cost them much and provide with the accurate facts and results.

2.2 Requirements on Automated Fitness Tracker

2.2.1 Functional Requirements:

Here we use PyCharm community to run and edit our project requirements. Many python libraries and modules are combined to provide a solid backend. The functions and classes are simple to read and understand to work upon, moreover python provides us with very easy functionalities for object-oriented programming (where we can convert our raw code to implement real world models) for creating real-time working model [1].

The frontend is provided by html, CSS and Django written inside the PyCharm community itself and thus, helping all the code to be written and debugged under a single platform and not perplexing the user. Html provides with the designing of our web application whereas CSS helps in styling our application and at last Django provides with the run-server for the application [2].

Dataset is stored in the form of SQLlite3 and is provided with very vast items or in simple words is a very large data set to work upon. Data cleaning and modifying is done to remove out all the invalid data in order to increase the accuracy of the results.

The backend has been handled by various python modules such as NumPy, nltk, pandas, etc. for creating a strong base of the project which we are deploying.

2.2.2 Non-Functional Requirements:

When we come to see where we can run our project, then it appears to be more satisfying that no specific paid hardware and software requirements are required and the project can simply run on your personal computers, tablets and laptops without any fuss. we just need to have a PyCharm software and any windows above 7, no additional system requirements are necessary.

Our project can be linked with the hardware such as smart watches and smart bands to track the activity of our project but this is beyond the scope of this current project.

2.3 Use Case Diagram of Automated Fitness Tracker

Use cases are used to depict the application in the form of diagrams to be able to understand the working of project very easily and effectively.

1. **Use Case for User:**

   This use case diagram is depicting the user privileges in our project. As we can see there are various functionalities available for the user and can be used very easily.

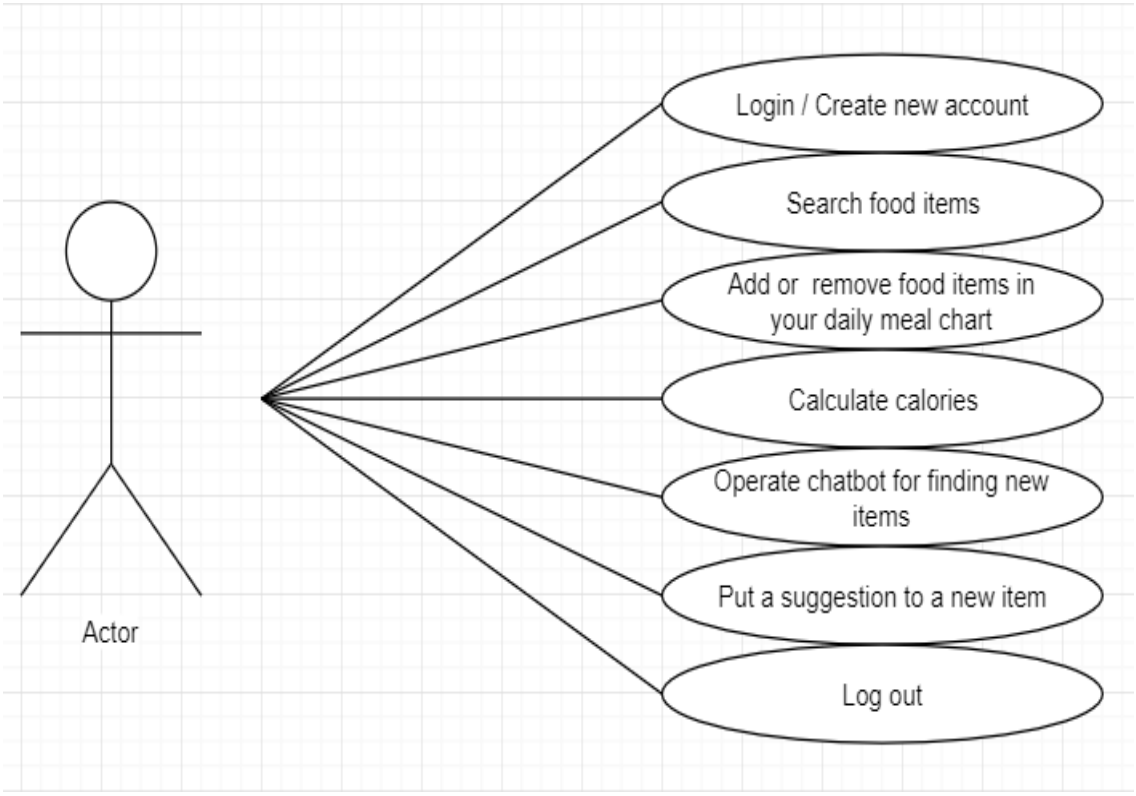


Figure 2-1Web Application Use Case Diagram of User

2. Use Case for Admin:

This use case diagram is depicting the admin privileges in our project. There are many functionalities for the admin and these cannot be handled carelessly.

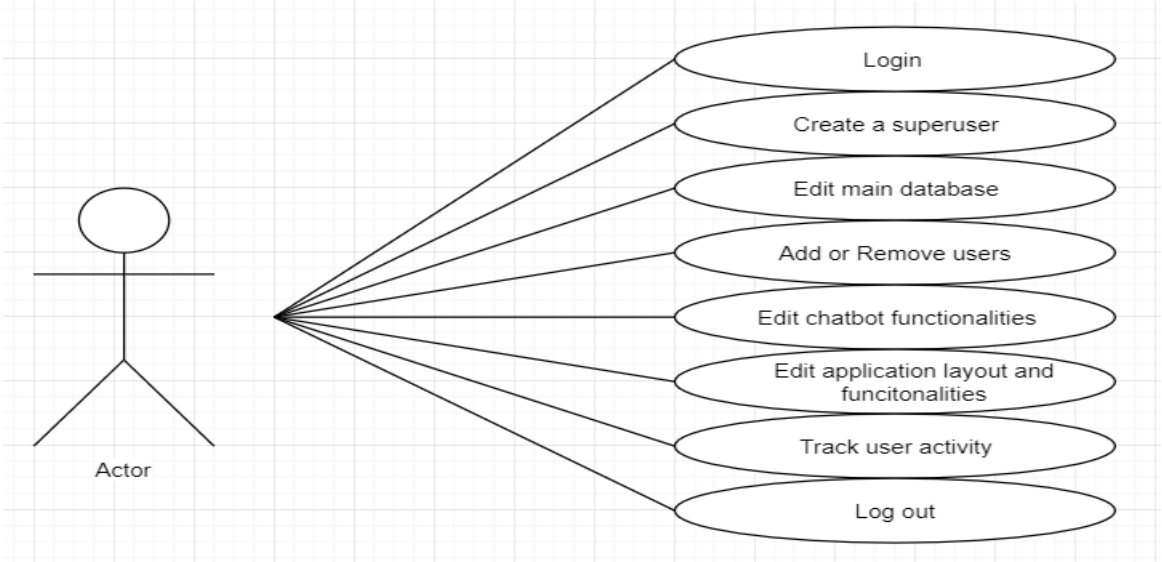


Figure 2-2 Web Application Use Case Diagram of Admin

3. Data set use case diagram:

This use case diagram is for the data set and training of our model. This is depicting the true process of our data refining and training of our model. The diagram depicts the flow of the data cleaning and model training.

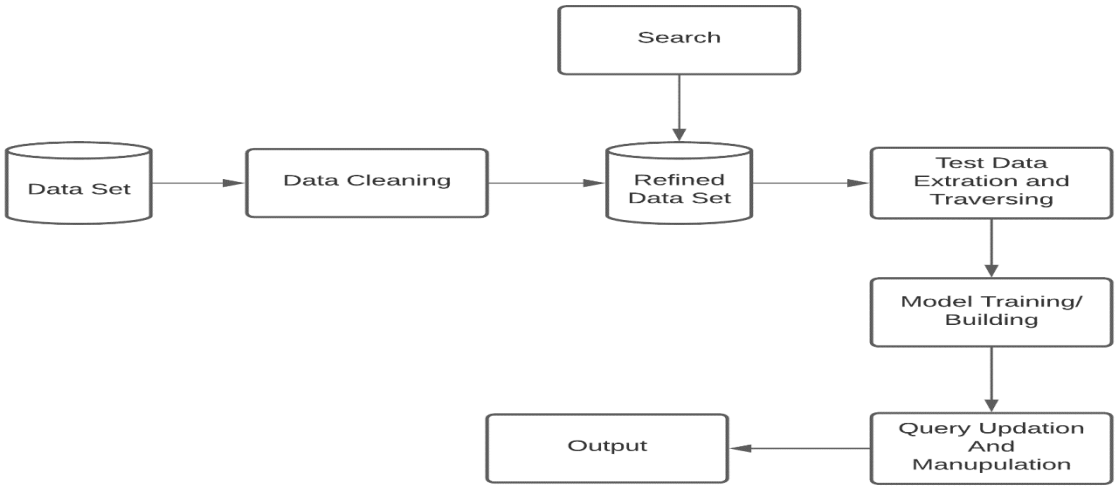


Figure 2-3 Data Use Case Diagram of automated calorie calculation

- Firstly, the data set should be cleaned and a refined data set should be formed
- Searching happens now in the refined data set.
- Then comes the test data extraction and traversing.
- The next step is to train the model which is fulfilled easily by the Django framework.
- Then queries are updated and data is manipulated
- And at last, the result is shown.

2.4 DFD Diagram of Automated Fitness Tracker

DFD (Data Flow Diagrams) are used to depict the flow of the data in a given project i.e., to check how the data will be handled under the project and what the flaws should be.

In the figure given below we learn that:

i. We have two login interfaces one for the user and other for the admin.
ii. And there is another option for users to create new accounts
iii. The password checking process takes place and then the next panel is opened if password is correct and if incorrect, they are reverted back to the login page to try again.
iv. After that admin and user functionalities are shown in which they can edit their database, operate the chatbot, calculate calories and perform other functions.
v. After that result is shown on the screen and then we can log out.

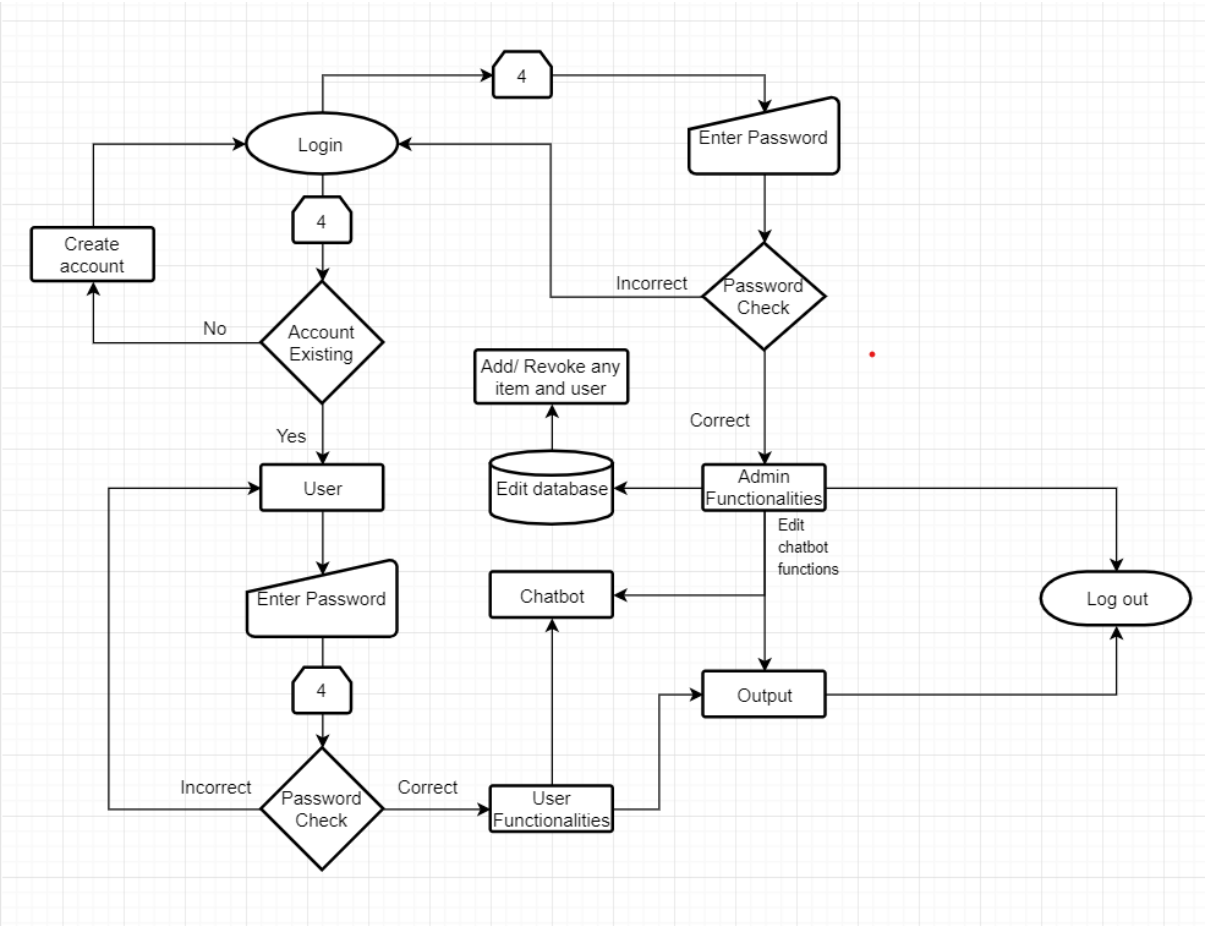


Figure 2-4 DFD Diagram for automated calorie calculation

Table 1 Key for DFD diagram given above

SNo.	Shape	Operation
1.		It depicts the starting point.
2.		It tells us about the various processes.
3.		It is used to check the conditions.
4.		These shows the state transitions.
5.		It depicts the termination.
6.		It tells about the loop limit.
7.		It tells us about the database.
8.		It tells us about the input data.

2.5 State Transition Diagram of Automated Fitness Tracker

There are eight states in our program as depicted below, one start state q0, six intermediate states (q1, q2, q3, q4, q5, q6, q7, q8) and at last one final state qf.

q0-> Initial stage is the start state;

q1-> Login state; q2-> Admin privileges; q3-> Edit database; q4->chatbot

functionalities; q5-> User privileges; q6-> Calorie calculation; q7-> Chatbot experience; q8->Output;

qf-> Final state is the end state.

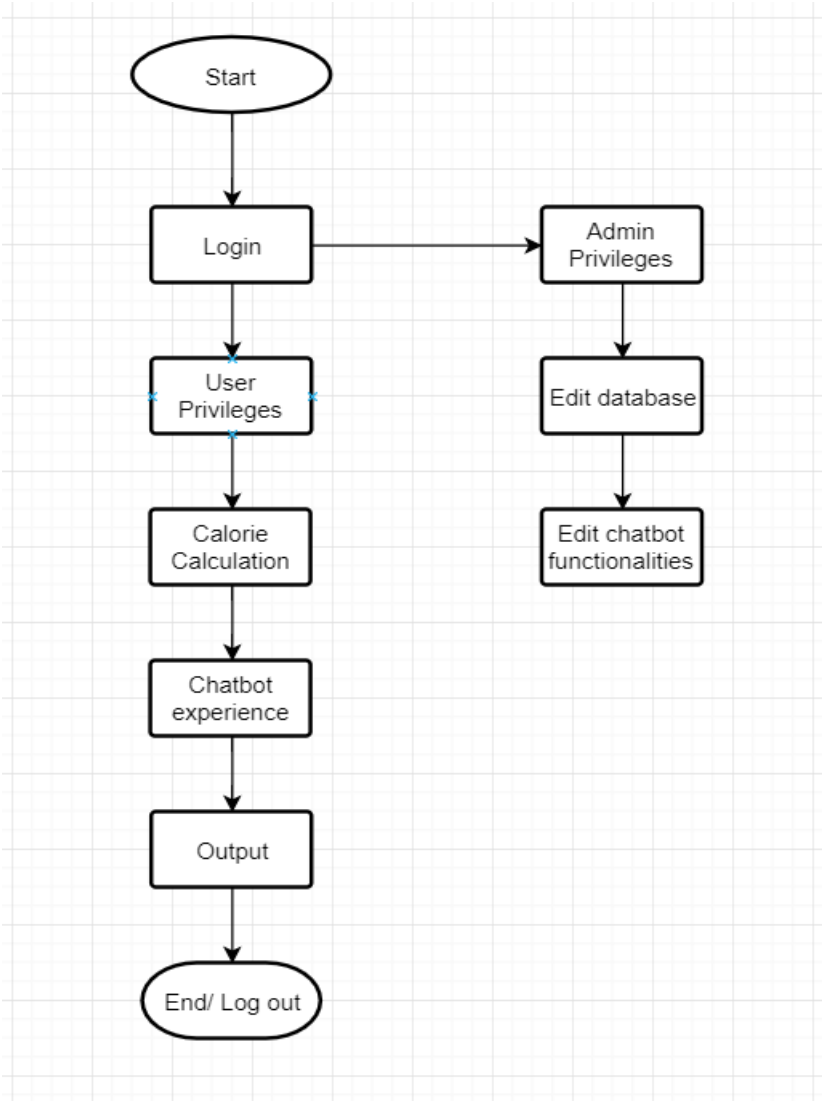


Figure 2-5 State Transition Diagram for automated calorie calculation

Chapter 3 LITERATURE SURVEY

3.1 HealthifyMe Android Application

Author: CEO Tushar Vashisht, Mathew Cherian, Sachin Shenoy

Publication: [HealthifyMe]

Figure 3-1 HealthifyMe app photo

About:

- HealthifyMe is a health calculator for calculating calories and it also provides personal trainers and coaches for dieting and workout plans. It is developed for Android and iOS platforms.
- These premium features of personal trainers and plans are only for the members with its subscription. It also provides various paid therapies by the world's best therapists and trainers,
- It also provides with an AI- driven nutritionist Ria for assistance.
- It syncs our footsteps and calorie expenditure via fitness bands and phone sensors also.
- It also provides us with medals and other functionalities if we achieve some milestones for boosting the confidence of an individual and making him strive harder for the more success.
- Nowadays it is also providing tracking of sleep, water, oxygen level, heartbeat, etc. using your mobile phones fingerprint sensors and various health bands.

- It also tracks workouts like running, swimming, bicycling, strength training and durability training exercises, etc...

3.2 JEFIT Android Application:

Author: CEO Ying Len

Publication: JEFIT app

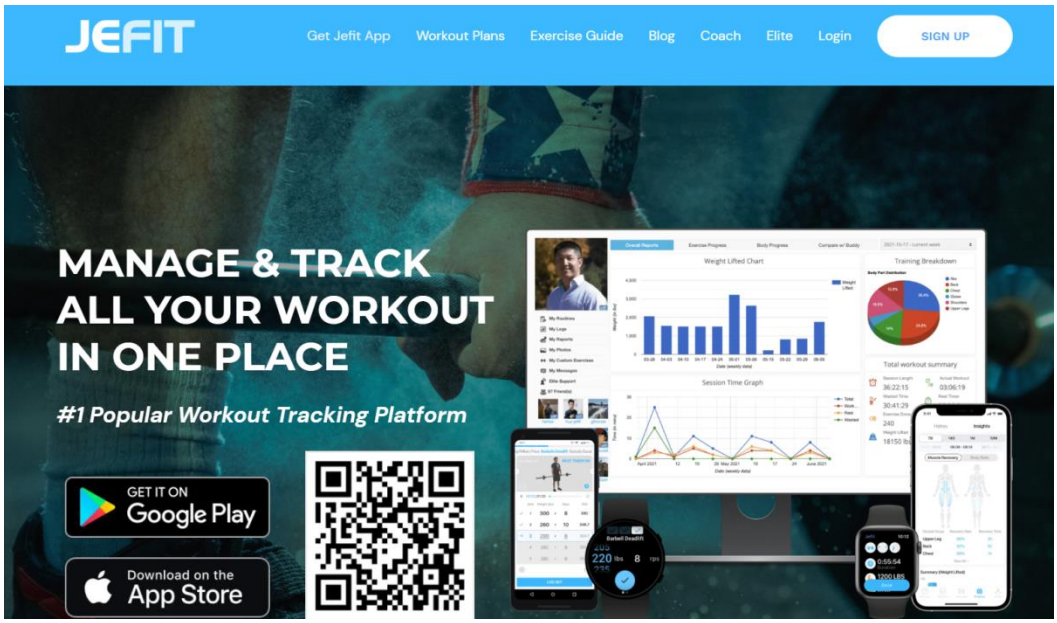


Figure 3-2 JEFIT Application

Figure 3-3 JEFIT interface

About:

- It helps one to track out their workouts easily and effectively.
- Here we can track our workout with the best options like the amount of time a workout is performed, the number of sets and repetition in a particular workout, etc…
- It gives a full graph analysis of our daily, weekly and monthly workouts which tells us the number of calories burned, our highest weight repetition and lowest weight repetition the amount of time in performing a particular exercise.
- It also has videos attached with particular exercises which helps us in performing an exercise accurately without any injury.
- Its visual representations help user identify the muscle breakdown and growth in a particular week.
- It also depicts the total amount of time required for the recovery of a particular muscle.

3.3 The Smart Kitchen Concept

Author: P. Chi, J. Chen, H. Chu and J. Lo

Publication: Lecture Notes in Computer Science, vol 5033. Springer, Berlin, Heidelberg

Figure 3-4 Smart kitchen

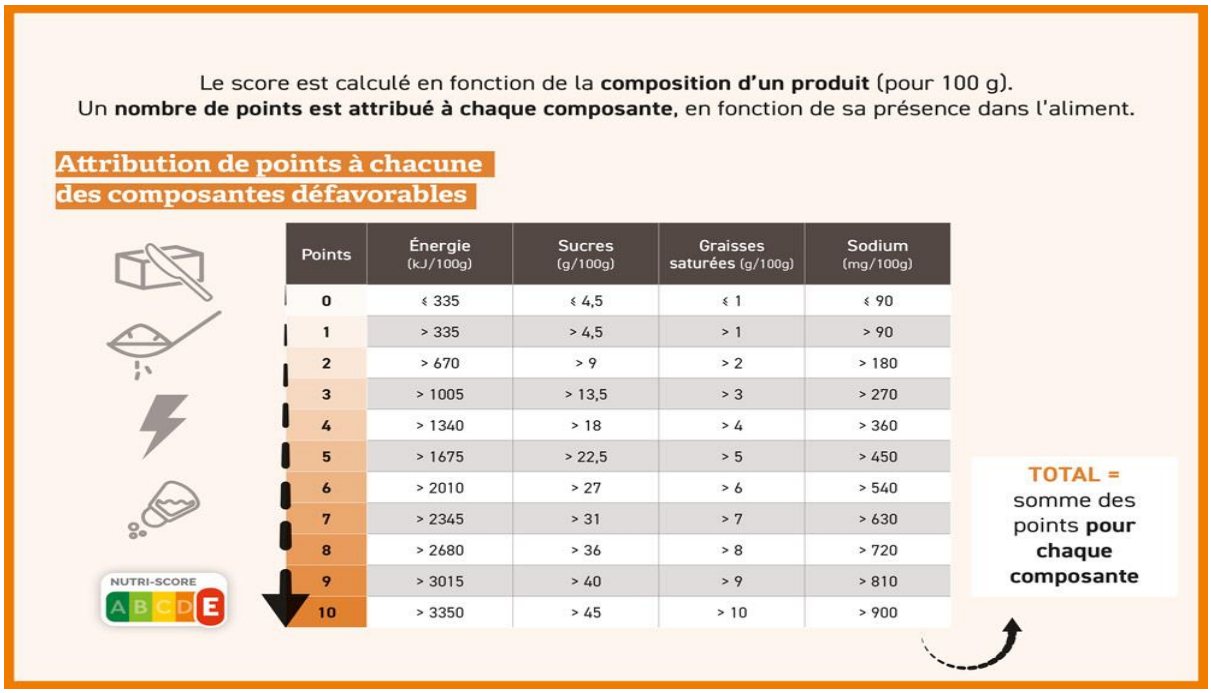
About:

- It was created as a concept of health calculators but they tried to do some invention getting them a huge success. This concept was one of a exciting features in Japan, China, USA and many more developed countries as it helped almost everyone in finding the proper amount of nutrition in their diets, even those who didn't knew cooking and ate ready to eat food only are now cooking their diet themselves with the help of this smart kitchen concept.
- The nutrition in daily dietary is targeted here. The simpler concept was to find out the healthy way in cooking good meals. They focused on AI technologies, sensors and other equipment's in order to find out the nutritional value of the item just by weighing and screening it.
- It is focused on the nutritional value of the ingredients used in the meal. It focuses in providing more healthy fats, proteins and fibres with an adequate amount of carbohydrates to support the energy expenditure.
- The development of Ubi Comp technology to find out the proper nutritional values of the food items and accordingly calculate the calorie of each different meals we prepare instead of following the traditional way of common caloric parameters on the internet.
- Kitchen sensors are used to track the number of calories in food ingredients, and then providing real-time feedback to users on these values through an awareness display by which they can find out that how much amount is required for their desired calorie meal.
- It is most suitable for the sportspersons and gym lovers who love to cook their meal themselves as they don't need to worry about proteins, fibers, sodium, good and bad fats, carbohydrates and other nutritional values in their meal to have great growth.
- This also focus on people struggling in weight loss, obesity, diabetes, thyroid and other health related issues as they can now have the proper nutritional values in their diet which will help them in fighting against such diseases.
- The result of this study suggests that bringing calorie awareness can be an effective means in helping family cooks maintain the healthy level of calories in their prepared meals.

3.4 CalculNutri Project

Author: Thierry Maillard (IT) and Perrine Maillard (Dietician)

Publication: Journal of Food and Search 7(1):33-40



Points	Énergie (kJ/100g)	Sucres (g/100g)	Graisses saturées (g/100g)	Sodium (mg/100g)
0	≤ 335	≤ 4,5	≤ 1	≤ 90
1	> 335	> 4,5	> 1	> 90
2	> 670	> 9	> 2	> 180
3	> 1005	> 13,5	> 3	> 270
4	> 1340	> 18	> 4	> 360
5	> 1675	> 22,5	> 5	> 450
6	> 2010	> 27	> 6	> 540
7	> 2345	> 31	> 7	> 630
8	> 2680	> 36	> 8	> 720
9	> 3015	> 40	> 9	> 810
10	> 3350	> 45	> 10	> 900

Figure 3-5 Nutritional score table

Figure 3-6 Nutrition analysis

About:

- CalculNutri is a food Calculator based on official food composition tables: CIQUAL 2017 and USDA SR 28.
- It targets the users which want to make their own recipe's and those who want to found out the nutritional value of their food.
- Using a simple and easy graphical user interface it can be used by highly qualified dieticians as well as by small children above 12 years of age.
- Custom tables and specific unit conversions are used to get through new portions.
- It is able to track the nutritional value of food given to patients.

3.5 The Caloric Calculator

Author: Y. Claire Wang, Amber Hasiao, C. Tracy Orleans, Steven L. Gortmaker

Publication: American Journal of Preventive Medicine

Population	Average weight (kg)	Schofield equation (BMR=)[a]	Intervention	Inputs			Caloric effect (kcal/day)[c]
				Δ METs	Duration (minutes/day)	School-based?[b]	
Boys, age in years							
2–5	18	22.706 kg + 504.3	Add 30 minutes/day of walking	2.3	30	No	44
6–11	34	22.706 kg + 504.3	Add 30 minutes/day of jogging	7	30	No	186
12–14	59	17.686 kg + 658.2	Add 15 minutes/day of PE	2.6	15	Yes	23
15–18	77	17.686 kg + 658.2	Implement SPARK	3.5	30	Yes	73

About:

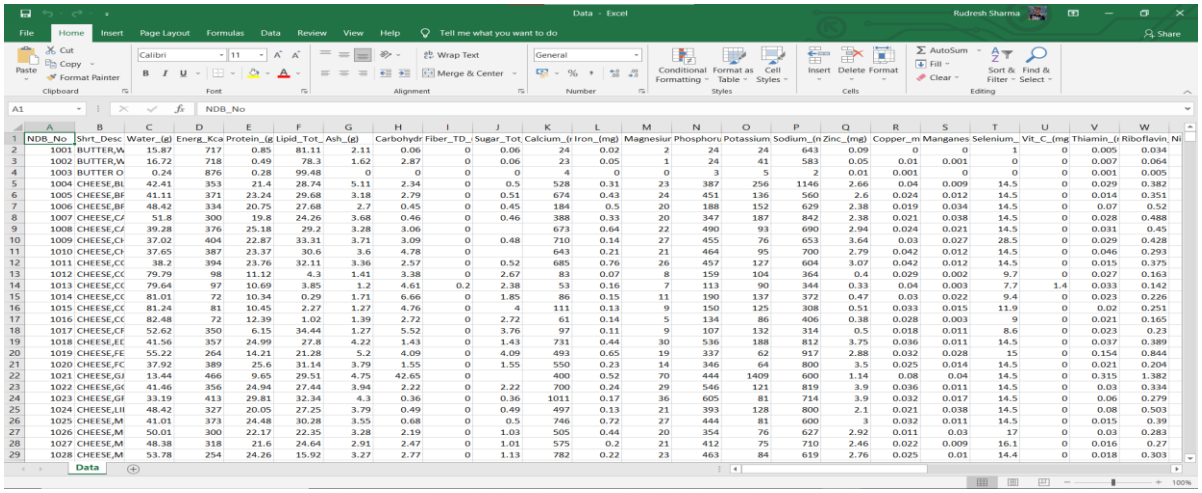
- Obesity in small children is increasing more often as their calorie intake is more as compared to their calorie expenditure. Also, the quality of calories they intake is not balanced as they take more fats, carbohydrates and focus less on the other nutritional components of food. Binge eating is also a major problem among youngsters that should be treated well and before an outburst of obese epidemic.

- In this report the authors are quite determined in finding an Average Caloric Impact (ACI), which would serve as a common parameter to map out the calories needs and intake among youngsters.
- The literature was reviewed from 1996 to 2012 and various parameters were found out about the young health such as their diets, energy expenditure, activity levels and weight.
- The ACIs were calculated based on the intensity, frequency duration and average body weight of a group of a particular age and gender from the 2009-2010 National Health and Nutrition Examination Survey.
- The Caloric Calculator was later on invented to be able to estimate the ACIs of specific target groups by taking care of the energy gap reductions required to reach the Healthy Populations childhood obesity.
- The Caloric Calculator was successful as it used the most common metric unit ACI and it emerged as a guiding pathway for the modernized calculators.

Chapter 4 IMPLEMENTATION OF THE MINOR PROJECT

4.1 Data Set Used in the Automated Fitness Tracker

- Data Set used in this project is taken from the World Food Programme (WFP), it is a vast dataset of nutrition composition of various food items. This dataset initially comprised of many nutrition components which can be merged with major nutrients like carbohydrates, proteins, sugars, fats and fibres for calories calculation. No doubt all nutrient components are responsible to provide a healthy lifestyle but when it comes to burn and count the calories, then only some of these major nutrient components are required.
- We also converted most of this information in the JSON format for the use of our chatbot.
- Dataset used for exercises have been taken from the Manitoba government physical health education and it comprises of 650 of total exercises. This data is refined and cleaned by using various python tools and packages for being able to be used by us for our calorie burned tracker.



NDB_No	Shrt_Desc	Water_(g)	Energ_Kca	Protein_(g	Lipid_Tot_	Ash_(g)	Carbohydr	Fiber_TD_	Sugar_Tot	Calcium_(r	Iron_(mg)	Magnesiur	Phosphoru	Potassium	Sodium_(n	Zinc_(mg)	Copper_m	Manganes	Selenium_	Vit_C_(mg	Thiamin_(r	Riboflavin_Ni
1001	BUTTER,W	15.87	717	0.85	81.11	2.11	0.06	0	0.06	24	0.02	2	24	24	643	0.09	0	0	1	0	0.005	0.034
1002	BUTTER,W	16.72	718	0.49	78.3	1.62	2.87	0	0.06	23	0.05	1	24	41	583	0.05	0.01	0.001	0	0	0.007	0.064
1003	BUTTER O	0.24	876	0.28	99.48	0	0	0	0	4	0	0	3	5	2	0.01	0.001	0	0	0	0.001	0.005
1004	CHEESE,BL	42.41	353	21.4	28.74	5.11	2.34	0	0.5	528	0.31	23	387	256	1146	2.66	0.04	0.009	14.5	0	0.029	0.382
1005	CHEESE,BF	41.11	371	23.24	29.68	3.18	2.79	0	0.51	674	0.43	24	451	136	560	2.6	0.024	0.012	14.5	0	0.014	0.351
1006	CHEESE,BF	48.42	334	20.75	27.68	2.7	0.45	0	0.45	184	0.5	20	188	152	629	2.38	0.019	0.034	14.5	0	0.07	0.52
1007	CHEESE,CA	51.8	300	19.8	24.26	3.68	0.46	0	0.46	388	0.33	20	347	187	842	2.38	0.021	0.038	14.5	0	0.028	0.488
1008	CHEESE,CA	39.28	376	25.18	29.2	3.28	3.06	0		673	0.64	22	490	93	690	2.94	0.024	0.021	14.5	0	0.031	0.45
1009	CHEESE,CH	37.02	404	22.87	33.31	3.71	3.09	0	0.48	710	0.14	27	455	76	653	3.64	0.03	0.027	28.5	0	0.029	0.428
1010	CHEESE,CH	37.65	387	23.37	30.6	3.6	4.78	0		643	0.21	21	464	95	700	2.79	0.042	0.012	14.5	0	0.046	0.293
1011	CHEESE,CC	38.2	394	23.76	32.11	3.36	2.57	0	0.52	685	0.76	26	457	127	604	3.07	0.042	0.012	14.5	0	0.015	0.375
1012	CHEESE,CC	79.79	98	11.12	4.3	1.41	3.38	0	2.67	83	0.07	8	159	104	364	0.4	0.029	0.002	9.7	0	0.027	0.163
1013	CHEESE,CC	79.64	97	10.69	3.85	1.2	4.61	0.2	2.38	53	0.16	7	113	90	344	0.33	0.04	0.003	7.7	1.4	0.033	0.142
1014	CHEESE,CC	81.01	72	10.34	0.29	1.71	6.66	0	1.85	86	0.15	11	190	137	372	0.47	0.03	0.022	9.4	0	0.023	0.226
1015	CHEESE,CC	81.24	81	10.45	2.27	1.27	4.76	0	4	111	0.13	9	150	125	308	0.51	0.033	0.015	11.9	0	0.02	0.251
1016	CHEESE,CC	82.48	72	12.39	1.02	1.39	2.72	0	2.72	61	0.14	5	134	86	406	0.38	0.028	0.003	9	0	0.021	0.165
1017	CHEESE,CF	52.62	350	6.15	34.44	1.27	5.52	0	3.76	97	0.11	9	107	132	314	0.5	0.018	0.011	8.6	0	0.023	0.23
1018	CHEESE,ED	41.56	357	24.99	27.8	4.22	1.43	0	1.43	731	0.44	30	536	188	812	3.75	0.036	0.011	14.5	0	0.037	0.389
1019	CHEESE,FE	55.22	264	14.21	21.28	5.2	4.09	0	4.09	493	0.65	19	337	62	917	2.88	0.032	0.028	15	0	0.154	0.844
1020	CHEESE,FC	37.92	389	25.6	31.14	3.79	1.55	0	1.55	550	0.23	14	346	64	800	3.5	0.025	0.014	14.5	0	0.021	0.204
1021	CHEESE,GI	13.44	466	9.65	29.51	4.75	42.65	0		400	0.52	70	444	1409	600	1.14	0.08	0.04	14.5	0	0.315	1.382
1022	CHEESE,GC	41.46	356	24.94	27.44	3.94	2.22	0	2.22	700	0.24	29	546	121	819	3.9	0.036	0.011	14.5	0	0.03	0.334
1023	CHEESE,GF	33.19	413	29.81	32.34	4.3	0.36	0	0.36	1011	0.17	36	605	81	714	3.9	0.032	0.017	14.5	0	0.06	0.279
1024	CHEESE,LI	48.42	327	20.05	27.25	3.79	0.49	0	0.49	497	0.13	21	393	128	800	2.1	0.021	0.038	14.5	0	0.08	0.503
1025	CHEESE,M	41.01	373	24.48	30.28	3.55	0.68	0	0.5	746	0.72	27	444	81	600	3	0.032	0.011	14.5	0	0.015	0.39
1026	CHEESE,M	50.01	300	22.17	22.35	3.28	2.19	0	1.03	505	0.44	20	354	76	627	2.92	0.011	0.03	17	0	0.03	0.283
1027	CHEESE,M	48.38	318	21.6	24.64	2.91	2.47	0	1.01	575	0.2	21	412	75	710	2.46	0.022	0.009	16.1	0	0.016	0.27
1028	CHEESE,M	53.78	254	24.26	15.92	3.27	2.77	0	1.13	782	0.22	23	463	84	619	2.76	0.025	0.01	14.4	0	0.018	0.303

Figure -4-1 Database in CSV format (1)

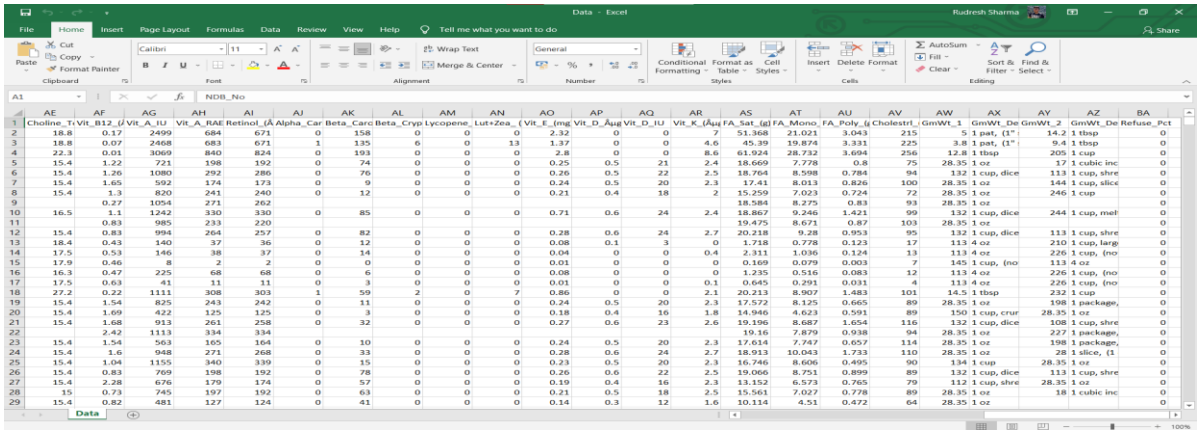


Figure 4-2 Database in CSV format (2)

Muscle Group	Exercise	Level	U/L/C	P / P	Modality	Joint
Abdominals - Lower	Full Reverse Crunch	Advanced	Core	Push	FW	M
Abdominals - Lower	Incline Hip Thrust	Advanced	Core	Push	FW	M
Abdominals - Lower	Incline Reverse Crunch	Advanced	Core	Push	FW	M
Abdominals - Lower	Lying Hip Thrust	Advanced	Core	Push	FW	M
Abdominals - Lower	Reverse Crunch	Beginner	Core	Push	FW	M
Abdominals - Lower	Reverse Medicine Ball Crunch	Advanced	Core	Push	FW	M
Abdominals - Obliques	Alternating Heel Touch	Beginner	Core	Push	FW	M
Abdominals - Obliques	Bent-Knee Medicine Ball Hip Rotation	Advanced	Core	Push	FW	M
Abdominals - Obliques	Cable Chop	Advanced	Core	Push	C	M
Abdominals - Obliques	Cross Crunch	Beginner	Core	Push	FW	M
Abdominals - Obliques	Cross Crunch w/ Medicine Ball	Advanced	Core	Push	FW	M
Abdominals - Obliques	Decline Cross Sit-Up	Advanced	Core	Push	FW	M
Abdominals - Obliques	Decline Sit-Up w/ Twist	Advanced	Core	Push	FW	M
Abdominals - Obliques	Reverse Cable Chop	Advanced	Core	Push	C	M
Abdominals - Obliques	Seated Medicine Ball Twist	Advanced	Core	Push	FW	M
Abdominals - Obliques	Trunk Rotator	Advanced	Core	Push	FW	M
Abdominals - Total	Front Plank (from knees)	Beginner	Core	Static	FW	M
Abdominals - Total	Front Plank (from toes)	Intermediate	Core	Static	FW	M
Abdominals - Total	Front Plank (tripod - 1 arm and 2 legs or 2 legs and 1 arm)	Advanced	Core	Static	FW	M
Abdominals - Total	Side-Plank (from knees)	Intermediate	Core	Static	FW	M
Abdominals - Total	Side-Plank (hip lift)	Beginner	Core	Static	FW	M
Abdominals - Total	Side-Plank (from toes)	Advanced	Core	Static	FW	M
Abdominals - Total	Ab Cycle	Advanced	Core	Push	FW	M
Abdominals - Total	Kneeling Ab Rollout	Intermediate	Core	Push	FW	M
Abdominals - Total	Medicine Ball V-Up	Advanced	Core	Push	FW	M
Abdominals - Total	V-Up	Beginner	Core	Push	FW	M
Abdominals - Total	Weighted V-Up	Intermediate	Core	Push	FW	M
Abdominals - Upper	Bent Knee Sit-Up	Beginner	Core	Push	FW	M

Figure 4-3 Exercise dataset in csv format

4.1.1 Data Cleaning

i. **Removing Useless Irrelevant Data:** This is the first approach of moving towards a refined data set. The removal of irrelevant values such as other minor nutrient components (like chlorine, phosphorous, potassium, etc.) which are not required in counting calories. And the removal of joints, modality which were not necessary in the scope of this current project were removed.

ii. **Removing Duplicating Values:** The duplicating values in our dataset has been removed by using pandas [3] module drop_duplicates () and other functionalities. The result is a more refined dataset.

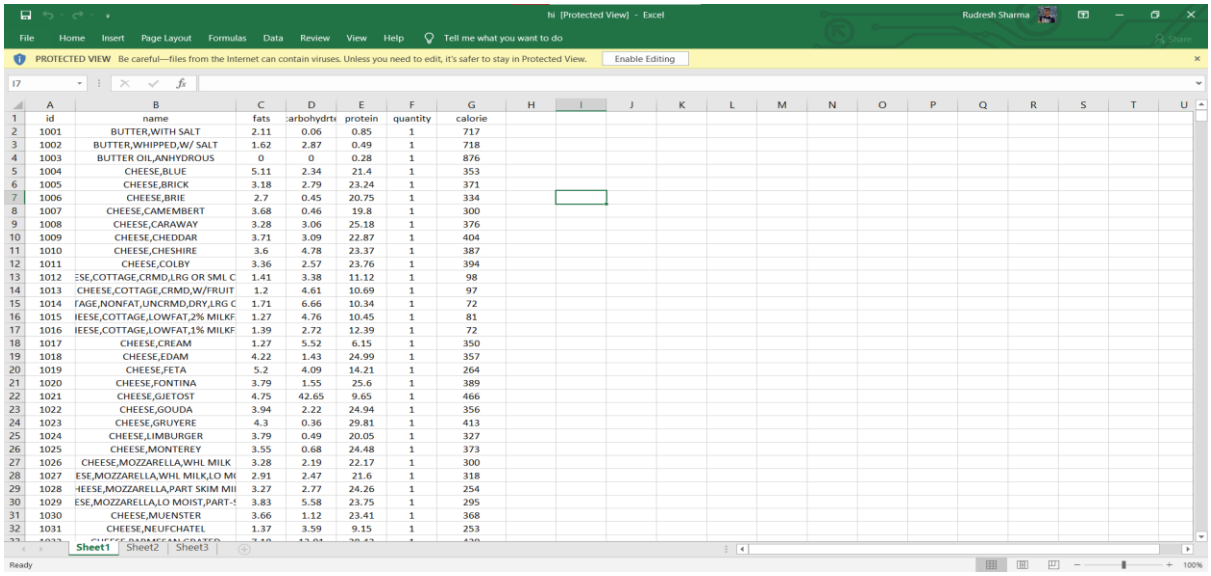
iii. **Avoiding Typos and Converting Data Types:** These can be cleared using various algorithms and functionalities. Uniformity in the data types is on focus here and to do the same data types should remain same and specific data types shouldn't be changed.

iv. **Editing Missing Values:** All the missing values are changed and converted here. It has been done manually and by removing null values using various functionalities and many other algorithms. Also, the calorie burned were also precisely added in the file which has been checked from the JEFIT android application database.

```
      NDB_No                 Shrt_Desc  ...  Carbohydrt_(g)  Calcium_(mg)
0       1001         BUTTER,WITH SALT  ...            0.06          24.0
1       1002  BUTTER,WHIPPED,W/ SALT   ...            2.87          23.0
2       1003    BUTTER OIL,ANHYDROUS   ...            0.00           4.0
3       1004              CHEESE,BLUE  ...            2.34         528.0
4       1005             CHEESE,BRICK  ...            2.79         674.0
...      ...                       ...  ...             ...           ...
8785   83110          MACKEREL,SALTED  ...            0.00          66.0
8786   90240  SCALLOP,(BAY&SEA),CKD,STMD  ...          5.41          10.0
8787   90480              SYRUP,CANE   ...           73.14          13.0
8788   90560              SNAIL,RAW    ...            2.00          10.0
8789   93600          TURTLE,GREEN,RAW  ...            0.00         118.0

[8790 rows x 7 columns]
```

Figure 4-4 Processing backend view

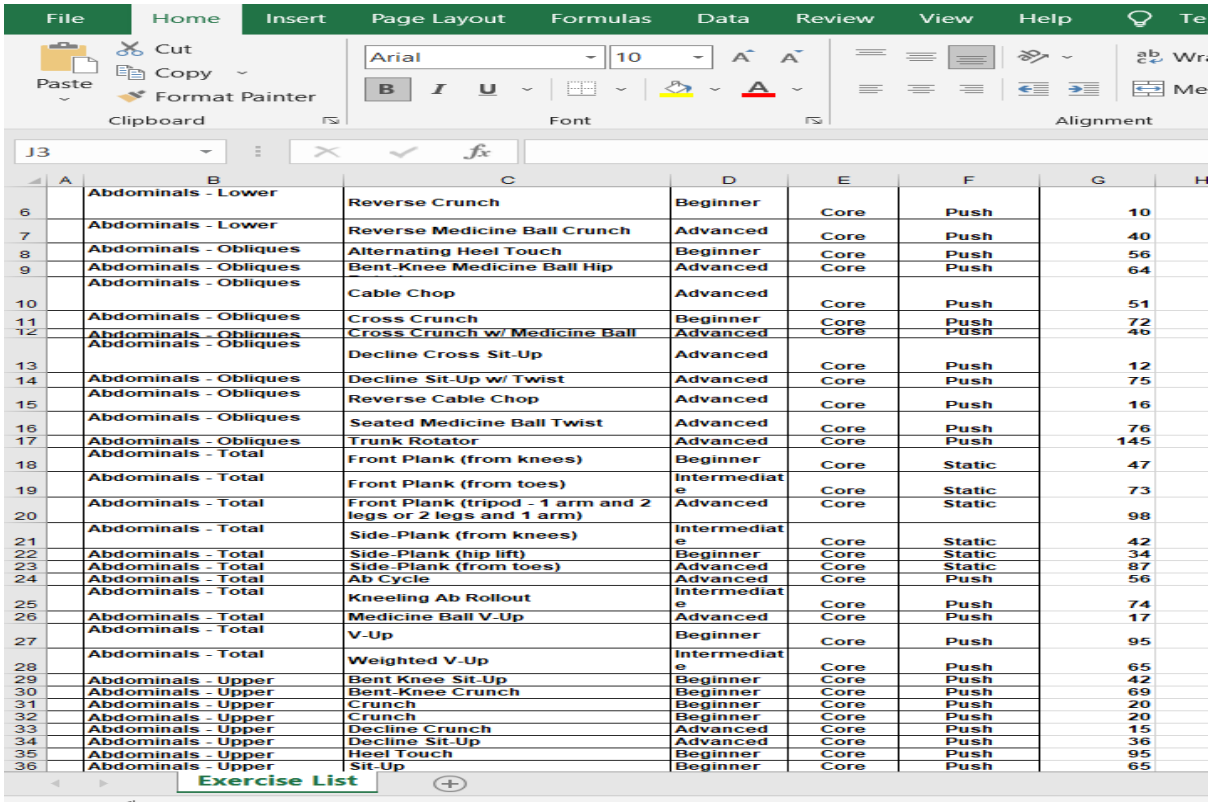
This is the newly refined and cleaned dataset of our food items which is used in the current project.



id	name	fats	carbohydrt	protein	quantity	calorie
1001	BUTTER,WITH SALT	2.11	0.06	0.85	1	717
1002	BUTTER,WHIPPED,W/ SALT	1.62	2.87	0.49	1	718
1003	BUTTER OIL,ANHYDROUS	0	0	0.28	1	876
1004	CHEESE,BLUE	5.11	2.34	21.4	1	353
1005	CHEESE,BRICK	3.18	2.79	23.24	1	371
1006	CHEESE,BRIE	2.7	0.45	20.75	1	334
1007	CHEESE,CAMEMBERT	3.68	0.46	19.8	1	300
1008	CHEESE,CARAWAY	3.28	3.06	25.18	1	376
1009	CHEESE,CHEDDAR	3.71	3.09	22.87	1	404
1010	CHEESE,CHESHIRE	3.6	4.78	23.37	1	387
1011	CHEESE,COLBY	3.36	2.57	23.76	1	394
1012	ESE,COTTAGE,CRMD,LRG OR SML C	1.41	3.38	11.12	1	98
1013	CHEESE,COTTAGE,CRMD,W/FRUIT	1.2	4.61	10.69	1	97
1014	TAGE,NONFAT,UNCRMD,DRY,LRG C	1.71	6.66	10.34	1	72
1015	IEESE,COTTAGE,LOWFAT,2% MILKF	1.27	4.76	10.45	1	81
1016	IEESE,COTTAGE,LOWFAT,1% MILKF	1.39	2.72	12.39	1	72
1017	CHEESE,CREAM	1.27	5.52	6.15	1	350
1018	CHEESE,EDAM	4.22	1.43	24.99	1	357
1019	CHEESE,FETA	5.2	4.09	14.21	1	264
1020	CHEESE,FONTINA	3.79	1.55	25.6	1	389
1021	CHEESE,GJETOST	4.75	42.65	9.65	1	466
1022	CHEESE,GOUDA	3.94	2.22	24.94	1	356
1023	CHEESE,GRUYERE	4.3	0.36	29.81	1	413
1024	CHEESE,LIMBURGER	3.79	0.49	20.05	1	327
1025	CHEESE,MONTEREY	3.55	0.68	24.48	1	373
1026	CHEESE,MOZZARELLA,WHL MILK	3.28	2.19	22.17	1	300
1027	ESE,MOZZARELLA,WHL MILK,LO M	2.91	2.47	21.6	1	318
1028	HEESE,MOZZARELLA,PART SKIM MI	3.27	2.77	24.26	1	254
1029	ESE,MOZZARELLA,LO MOIST,PART-	3.83	5.58	23.75	1	295
1030	CHEESE,MUENSTER	3.66	1.12	23.41	1	368
1031	CHEESE,NEUFCHATEL	1.37	3.59	9.15	1	253

Figure 4-5 Cleaned Dataset

This is the newly refined and cleaned dataset of exercises which is used for the current project.



	B	C	D	E	F	G
6	Abdominals - Lower	Reverse Crunch	Beginner	Core	Push	10
7	Abdominals - Lower	Reverse Medicine Ball Crunch	Advanced	Core	Push	40
8	Abdominals - Obliques	Alternating Heel Touch	Beginner	Core	Push	56
9	Abdominals - Obliques	Bent-Knee Medicine Ball Hip	Advanced	Core	Push	64
10	Abdominals - Obliques	Cable Chop	Advanced	Core	Push	51
11	Abdominals - Obliques	Cross Crunch	Beginner	Core	Push	72
12	Abdominals - Obliques	Cross Crunch w/ Medicine Ball	Advanced	Core	Push	46
13	Abdominals - Obliques	Decline Cross Sit-Up	Advanced	Core	Push	12
14	Abdominals - Obliques	Decline Sit-Up w/ Twist	Advanced	Core	Push	75
15	Abdominals - Obliques	Reverse Cable Chop	Advanced	Core	Push	16
16	Abdominals - Obliques	Seated Medicine Ball Twist	Advanced	Core	Push	76
17	Abdominals - Obliques	Trunk Rotator	Advanced	Core	Push	145
18	Abdominals - Total	Front Plank (from knees)	Beginner	Core	Static	47
19	Abdominals - Total	Front Plank (from toes)	Intermediate	Core	Static	73
20	Abdominals - Total	Front Plank (tripod - 1 arm and 2 legs or 2 legs and 1 arm)	Advanced	Core	Static	98
21	Abdominals - Total	Side-Plank (from knees)	Intermediate	Core	Static	42
22	Abdominals - Total	Side-Plank (hip lift)	Beginner	Core	Static	34
23	Abdominals - Total	Side-Plank (from toes)	Advanced	Core	Static	87
24	Abdominals - Total	Ab Cycle	Advanced	Core	Push	56
25	Abdominals - Total	Kneeling Ab Rollout	Intermediate	Core	Push	74
26	Abdominals - Total	Medicine Ball V-Up	Advanced	Core	Push	17
27	Abdominals - Total	V-Up	Beginner	Core	Push	95
28	Abdominals - Total	Weighted V-Up	Intermediate	Core	Push	65
29	Abdominals - Upper	Bent Knee Sit-Up	Beginner	Core	Push	42
30	Abdominals - Upper	Bent-Knee Crunch	Beginner	Core	Push	69
31	Abdominals - Upper	Crunch	Beginner	Core	Push	20
32	Abdominals - Upper	Crunch	Beginner	Core	Push	20
33	Abdominals - Upper	Decline Crunch	Advanced	Core	Push	15
34	Abdominals - Upper	Decline Sit-Up	Advanced	Core	Push	36
35	Abdominals - Upper	Heel Touch	Beginner	Core	Push	95
36	Abdominals - Upper	Sit-Up	Beginner	Core	Push	65

Figure 4-6 Cleaned exercise dataset

4.2 Data Set Features

This data set is used for calculating the original major nutrient value of an edible item through which we are able to find about the calories (in Kcal).

4.2.1 Types of Data Set

Here there are different datasets for different functions:

i. Dataset of food items: Main dataset used for calculation and display
ii. Dataset of exercises: Dataset used for calculating calories burned and display
iii. Tables of customer_ids: Store ids of customers
iv. Tables of superuser_ids: Store ids of admins
v. JSON file data of chatbot

Figure 4-7 JSON data format

4.2.2 Number of Attributes, Field, Description of dataset

There are several datasets in our project but for:

I. the chatbot and calorie calculation the dataset attributes used are provided by 7 columns starting with:
   - **i.** Id number;
   - **ii.** Name of the item;
   - **iii.** Amount of fat in the given item;
   - **iv.** Amount of carbohydrate in the given item;
   - **v.** Amount of protein in the given item;
   - **vi.** Amount of fibre in the given item;
   - **vii.** And at last, the calculated calories of the given item.

Table 2 Data set attributes

SNo.	Name	Fats	Carbohydrates	Protein	Fibres	Calories
1.	Cheese	12.0	30.0	10.0	20.0	150
2.	Pork	45.0	150.0	25.2	32.4	220
3.	Bread	15.0	70.0	2.5	30.4	70
4.	Butter	55.0	120.54	1.5	4.9	140
5.	Milk	25.0	100.89	5.6	0.5	90
6.	Chocolate	59.6	147.7	6.1	8.2	150
7.	Rice	32.4	97.6	3.5	20.6	125

II. The workout tracker and calorie burned calculator provided by 7 columns starting with:

i. Id number;
ii. Muscle Group;
iii. Exercise;
iv. Level;
v. U/L/C;
vi. Push/Pull;
vii. Calories.

Table 3 Exercises targeting particular muscle

SNo.	Muscle Group	Exercise	Level	U/L/C	Push/Pull	Calories
1.	Abdominals-Lower	Full Reverse Crunch	Advanced	Core	Push	45
2.	Abdominals - Obliques	Decline Sit-Up w/ Twist	Advanced	Core	Push	35
3.	Back - Latissimus Dorsi	Assisted Pull-Up	Beginner	Upper	Pull	26
4.	Back - Lat.Dorsi/Rhomboids	Bent-Over Dumbbell Row	Intermediate	Upper	Pull	56
5.	Biceps	Alternating Dumbbell Curl	Advanced	Upper	Pull	60
6.	Chest - Pectoralis	Alternating Dumbbell Bench Press	Intermediate	Upper	Push	70
7.	Legs - Quadriceps	Lateral Barbell Step-Up	Advanced	Lower	Push	65
8.	Shoulders-Delts/Traps	Pec Deck Real Delt Extensions	Beginner	Upper	Push	46
9.	Triceps	Triceps Pushdown	Advanced	Upper	Push	44

4.3 Design of the problem statement

The idea is to create a fully develop application to calculate calories eaten and burned with a built-in chatbot but for now a highly functional prototype is ready to be deployed. The main focus is to check the speed and accuracy of the calculator and the chatbot. It is created in such a way that we can keep a track of our every meal and also calculate the number of calories burned in a day and to find the details of our every meal using the chatbot. It is designed in such a way that it should be capable of adjusting and training itself to the likes of our user.

4.4 Algorithm/ Pseudo code of the Automated Fitness Tracker

4.4.1 Algorithm:

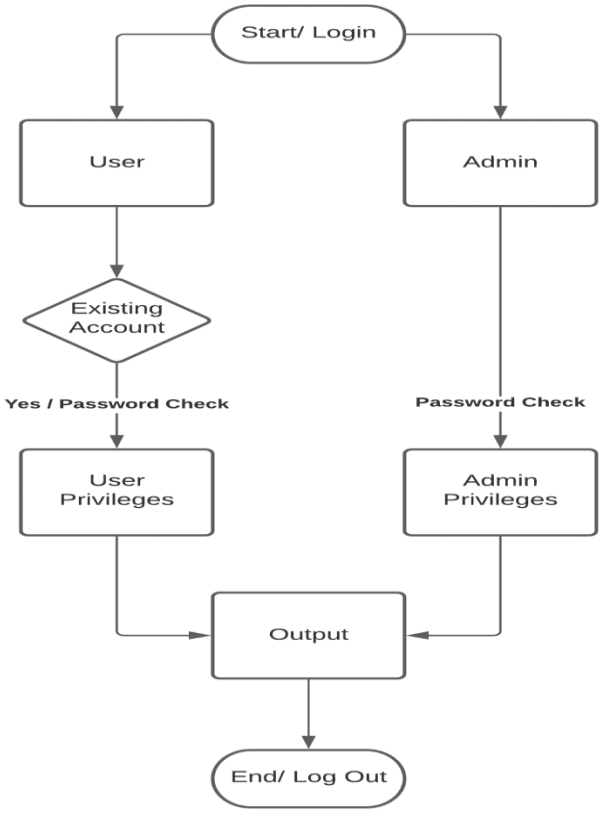


Figure 4-8 Algorithm of the given project

4.4.2 Pseudo Code:

- The user/ admin will login or sign in the application.
- Or they can create a new account.
- Passwords will be checked and then the application will be proceeded further.
- Then the user privileges and admin privileges can be provoked.

- The output will be shown.
- Then they can exit the application.

4.5 Flow Graph of the Automated Fitness Tracker

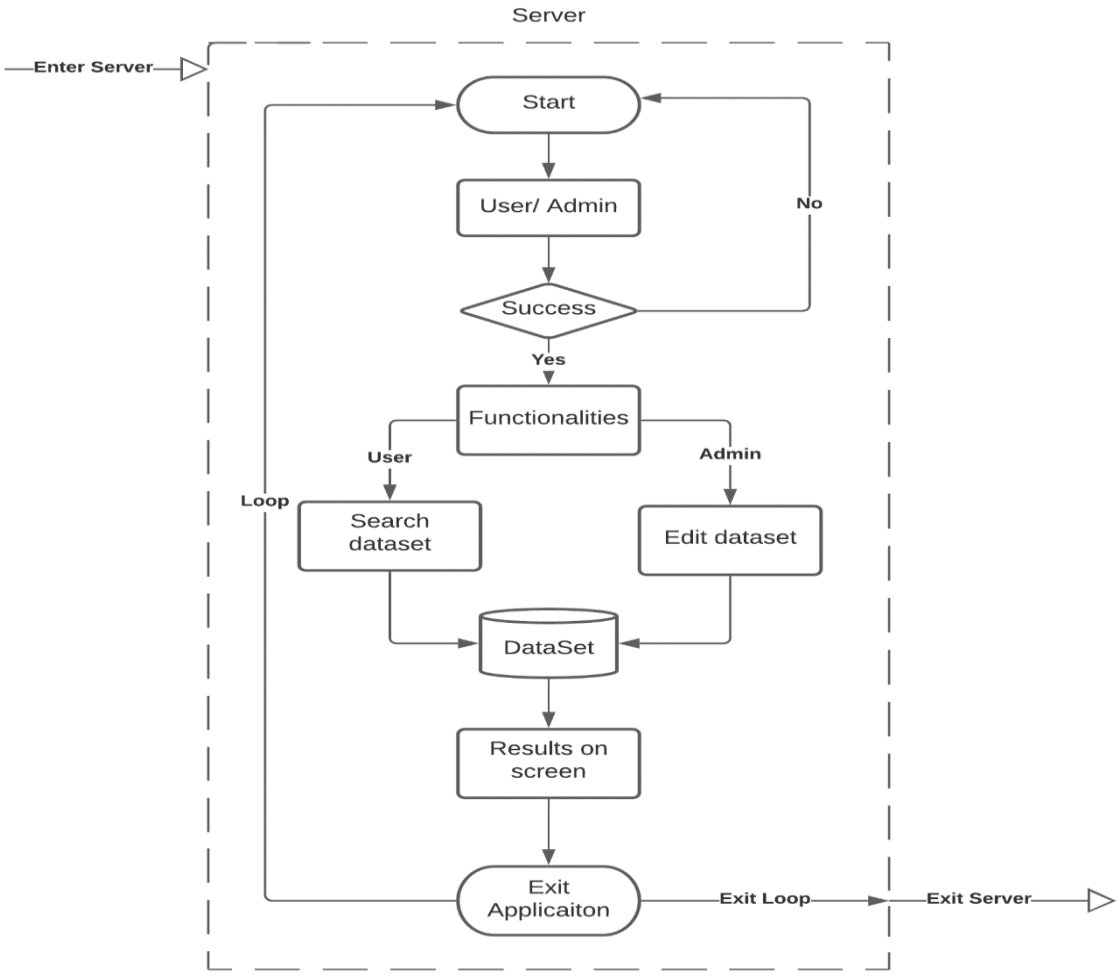


Figure 4-9 Flow graph of the project problem

- We enter the server created through the Django module of the python and then we use that to create and run our application on the server provided by the Django [2].
- The user log themselves in or create a new account and then they use the applications functionalities independently and based on what they can access.
- In the end the results are displayed on the output screen and then user exits the application, at this point he can either use the application again or can exit the application server.

4.6 About Chatbot

A chatbot is an application used to conduct a communication with a client via an online platform in order to provide him with the help he seeks and sort out the other functionalities for him similarly to a live human agent.

The idea to deploy chatbot into our application came up with the modernization of the society and the increase in new technologies in the market. So, to keep our project in competition with the other projects we have to deploy such technologies in our project. We have kept in mind the compatibility of our project in with upcoming technologies so that our project doesn't fade away. With the help of chatbot now it is simpler and easier to find out the flaws and suggestions for our applications.

Figure 4-10 Chatbot hello

4.6.1 Libraries used in our chatbot

1. **NLTK:** It is called Natural Language toolkit, it us used to process Natural language and give us the desired output. It helps in preprocessing the natural language data into useful terms for the chatbot.

2. **Keras:** It evaluated our model and helped in developing the training and testing models of our project using tensor flow applications.
3. **Pickle:** The pickle library in python is used to convert the python object into stream of bytes and store it in file. It is used for storing the multiple python objects.
4. NumPy: It is a python library which is used to perform operation on various entities like n- dimensional arrays and matrices. It is used to perform multiple numerical and other tests over our models.
5. **JSON:** JSON library in python is used to connect are code to JSON file which helps fetch, manipulate data from the file. In JSON format we have converted our data so that the chatbot can convert the data more accurately and quickly. It is easy to locate data while it is in the JSON format instead of looking into a key and then retrieving data from there.
6. **Tkinter:** Tkinter is a GUI for python projects. It is used to create a dialog box for chatbot that display the conversation between user and the chatbot.

4.6.2 Working of the chatbot

1. To open **chatbot** firstly click on the **green chatbot** button this is below search button.
2. Now separate Tkinter window appears that is split into two parts
    a. Upper part of dialog box is used to display our message and how chatbot responds to it.
    b. The lower part is used by user to write his/her message along with send button.
3. Now the user types his/her message i.e., name of food item and clicks send button
4. Now the **chatbot** takes users request it searches users requests through tags in **JSON** file and gives out the appropriate response i.e., Nutritional values of food item written by user

4.6.3 Algorithm/ Pseudo code for the Chatbot

- **Algorithm:**

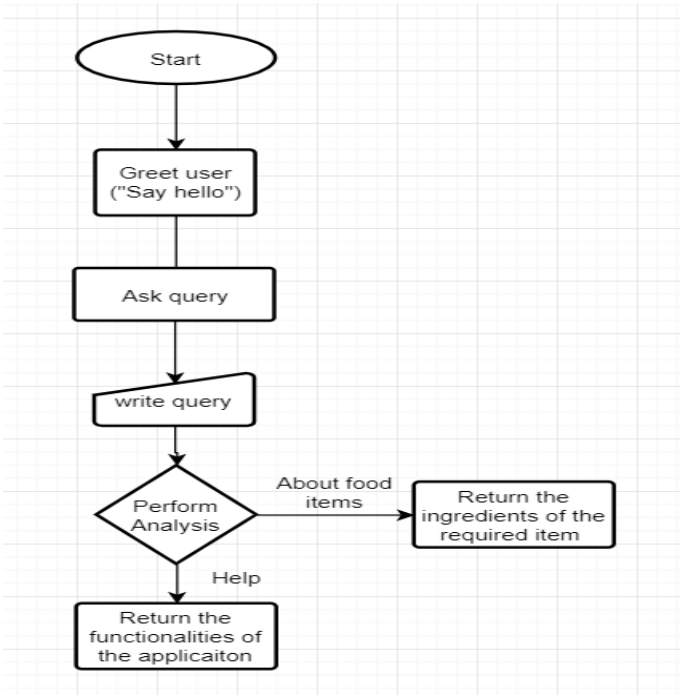


Figure 4-11 Algorithm for chatbot

- **Pseudo Code:**

8. Greet the user.
9. Read the query of user and accordingly perform analysis.
10. If he wants to know about the functionalities tell him about all of them.
11. If he wants to know about the ingredients of a food item read it carefully and perform analysis.
12. Return result

4.7 Screen shots of the various stages of Automated Fitness Tracker

4.7.1 Data Cleaning and increasing accuracy:

Here we've used python pandas and NumPy module for cleaning the data by using various algorithms and functionalities to remove null values and other irrelevant data [5].

Data cleaning process is a huge process where we have to find every flaw in our data and clean it with the help of various methods provided by the python environment and

the pandas module. After cleaning we have to reduce the redundancy to increase the processing speed of our data.

```
import pandas as pd
import numpy as np
import os
from fuzzywuzzy import process, fuzz
ramen = pd.read_csv('Data12.csv')
df = pd.read_csv('Data12.csv')
ramen.head()
for col in ramen[['Shrt_Desc']]:
    ramen[col] = ramen[col].str.strip()
    print('Number of unique values in ' + str(col) +': ' + str(ramen[col].nunique()))
unique_brand = ramen['Shrt_Desc'].unique().tolist()
sorted(unique_brand)[:20]
list=[]
dict1={}
count = []
cnt=0
first_time = 1
filename = "logs.csv"
f = open(filename, "w")
f.close()
while(1):
  inp= input("Enter item: ")
  dict1.update({ inp: cnt})
  x=process.extract(inp, unique_brand, scorer=fuzz.token_sort_ratio)
  print(x)
  a=x[0][0]
  print("--------")
  print(a)
  print("--------")
```

Figure 4-12 Data Cleaning and using Fuzzy Wuzzy Code (1)

For increasing accuracy, we use the FuzzyWuzzy module which helps us in increasing the searching speed and provide us with more accurate results with the help of its various functionalities [4].

```
  a=x[0][0]
  print("--------")
  print(a)
  print("--------")
  #list12=df[['Shrt_Desc']].values.tolist()
  #p#rint(type(List12))
  #if(a in df[df['Shrt_Desc']]):
  abc = (df[df['Shrt_Desc']== a])
  if(first_time == 1):
    abc.to_csv('logs.csv', mode='a', header=True)
    first_time = 0
  else:
    abc.to_csv('logs.csv', mode='a', header=False)
  list.append(abc)
  count.append(abc)


  print('Add more values:')
  x = input("y/n: ")
  if (x == 'y'):
    continue

  elif (x == 'n'):
    break
#df3 = pd.DataFrame(abc)

print(abc)
#print(list)
#print(dict)
##print(a)

#print(rows)
```

Figure 4-13 Data Cleaning and using Fuzzy Wuzzy Code (2)

❖ Output of the above provided snippets:

```
      NDB_No                 Shrt_Desc  ...  Carbohydrt_(g)  Calcium_(mg)
0       1001          BUTTER,WITH SALT  ...            0.06          24.0
1       1002    BUTTER,WHIPPED,W/ SALT  ...            2.87          23.0
2       1003      BUTTER OIL,ANHYDROUS  ...            0.00           4.0
3       1004             CHEESE,BLUE  ...            2.34         528.0
4       1005            CHEESE,BRICK  ...            2.79         674.0
...      ...                      ...  ...             ...           ...
8785   83110          MACKEREL,SALTED  ...            0.00          66.0
8786   90240  SCALLOP,(BAY&SEA),CKD,STMD  ...            5.41          10.0
8787   90480               SYRUP,CANE  ...           73.14          13.0
8788   90560              SNAIL,RAW  ...            2.00          10.0
8789   93600          TURTLE,GREEN,RAW  ...            0.00         118.0

[8790 rows x 7 columns]
```

Figure 4-14 Output of the existing data set

```
Number of unique values in Shrt_Desc: 8787
Enter item: CHEESE
[('CHEESE,BLUE', 71), ('CHEESE,BRIE', 71), ('CHEESE,EDAM', 71), ('CHEESE,FETA', 71), ('CHEESE,BRICK', 67)]
--------
CHEESE,BLUE
--------
Add more values:
y/n: y
Enter item: Sandwitch
[('ICE CRM SNDWCH', 52), ('CANDIES,HEATH BITES', 50), ('BUTTER,WITH SALT', 48), ('APPLES,RAW,WITH SKIN', 48), ('CANDIES,HARD', 48)]
--------
ICE CRM SNDWCH
--------
Add more values:
y/n: n
     NDB_No       Shrt_Desc  ...  Carbohydrt_(g)  Calcium_(mg)
202    1238  ICE CRM SNDWCH  ...           37.14          86.0

[1 rows x 7 columns]
```

Figure 4-15 Backend output of FuzzyWuzzy module

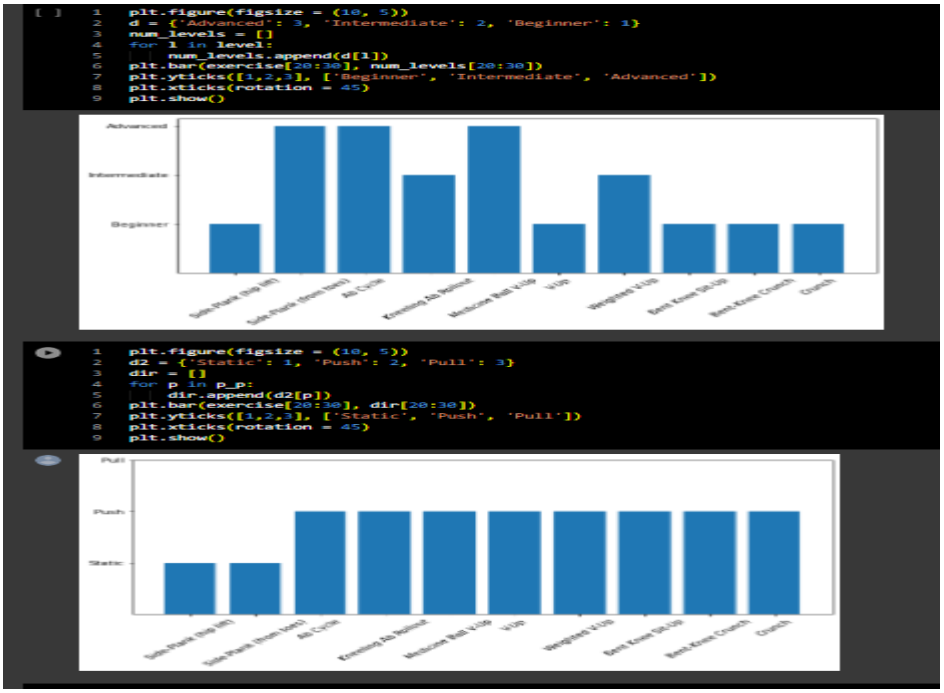
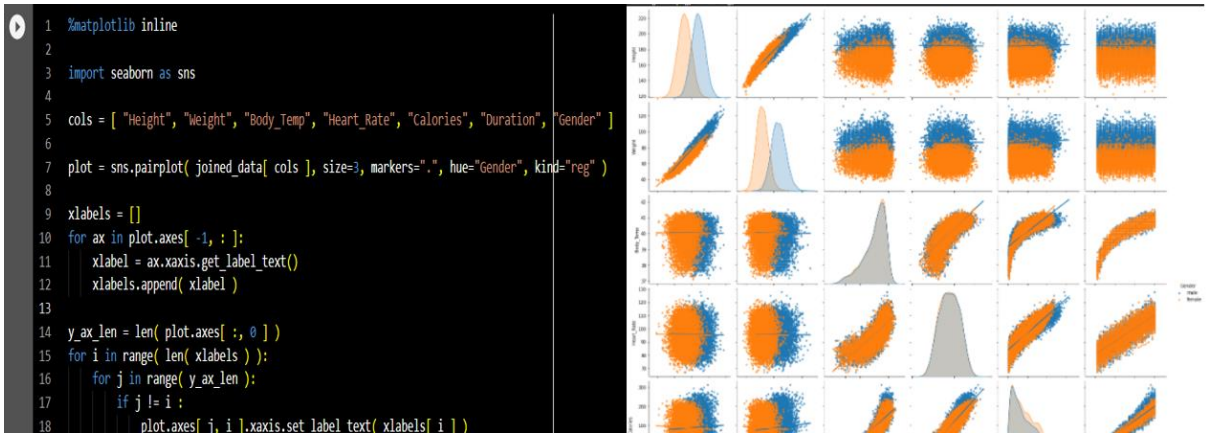


Figure 4-16 Data representation of exercise dataset

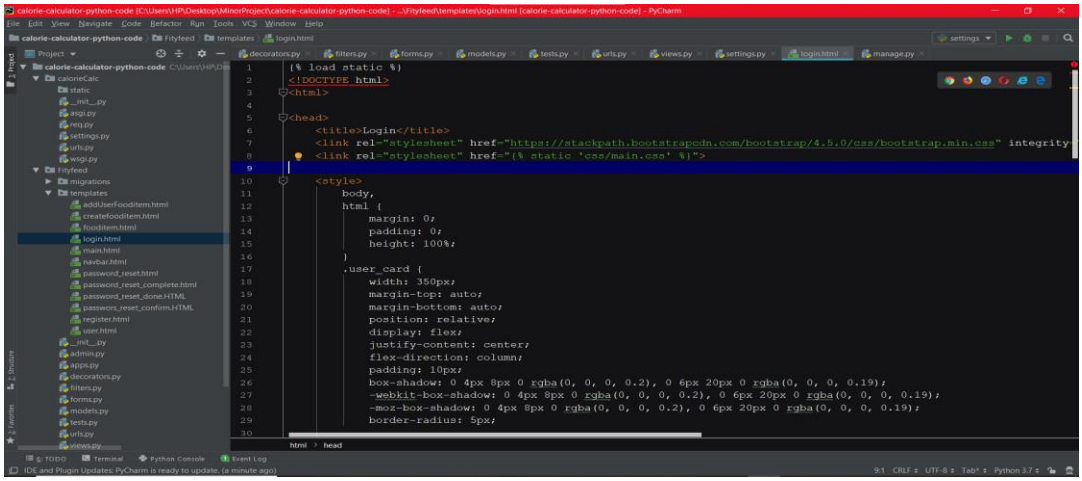


```
%matplotlib inline

import seaborn as sns

cols = [ "Height", "Weight", "Body_Temp", "Heart_Rate", "Calories", "Duration", "Gender" ]

plot = sns.pairplot( joined_data[ cols ], size=3, markers=".", hue="Gender", kind="reg" )

xlabels = []
for ax in plot.axes[ -1, : ]:
    xlabel = ax.xaxis.get_label_text()
    xlabels.append( xlabel )

y_ax_len = len( plot.axes[ :, 0 ] )
for i in range( len( xlabels ) ):
    for j in range( y_ax_len ):
        if j != i :
            plot.axes[ j, i ].xaxis.set_label_text( xlabels[ i ] )
```

Figure 4-17 Graphical Data Representation

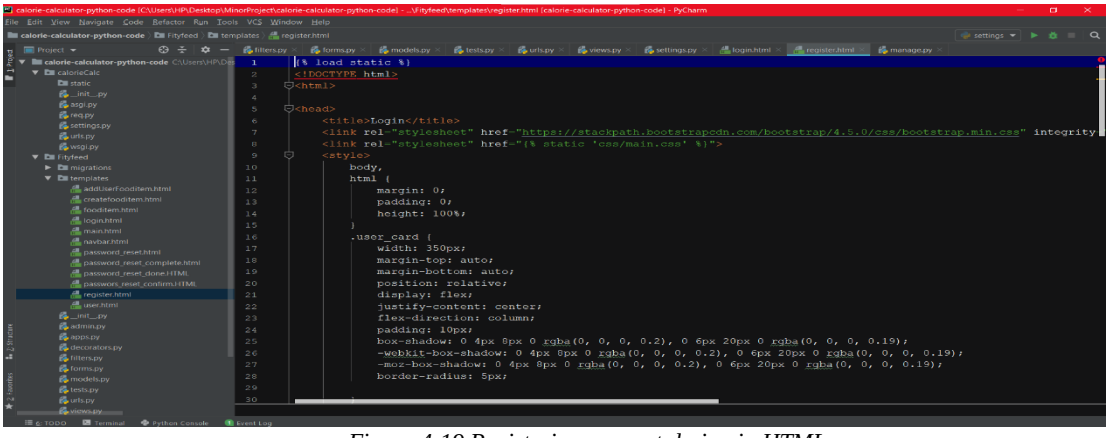
4.7.2 Designing of the web application:

It is done under various steps by creating various designs individually for new pages for example: login page, signup page, main home screen, search screen, etc. under the folder name templates in the python project environment. CSS and Django bootstrap is used in the beautifying and styling of the html pages [2].



```
{% load static %}
<!DOCTYPE html>
<html>

<head>
    <title>Login</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity
    <link rel="stylesheet" href="{% static 'css/main.css' %}">

    <style>
        body,
        html {
            margin: 0;
            padding: 0;
            height: 100%;
        }
        .user_card {
            width: 350px;
            margin-top: auto;
            margin-bottom: auto;
            position: relative;
            display: flex;
            justify-content: center;
            flex-direction: column;
            padding: 10px;
            box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
            -webkit-box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
            -moz-box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
            border-radius: 5px;
```

Figure 4-18 Login page design in HTML



```
{% load static %}
<!DOCTYPE html>
<html>

<head>
    <title>Login</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" integrity
    <link rel="stylesheet" href="{% static 'css/main.css' %}">
    <style>
        body,
        html {
            margin: 0;
            padding: 0;
            height: 100%;
        }
        .user_card {
            width: 350px;
            margin-top: auto;
            margin-bottom: auto;
            position: relative;
            display: flex;
            justify-content: center;
            flex-direction: column;
            padding: 10px;
            box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
            -webkit-box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
            -moz-box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
            border-radius: 5px;
```

Figure 4-19 Registering account design in HTML

❖ **Output of the above provided snippet**:

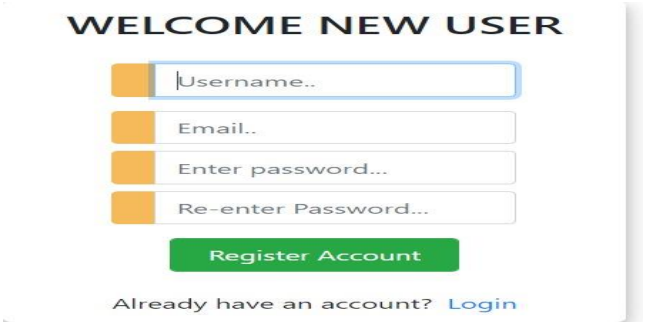


Figure 4-22 Figure 4 19 Sign in page display

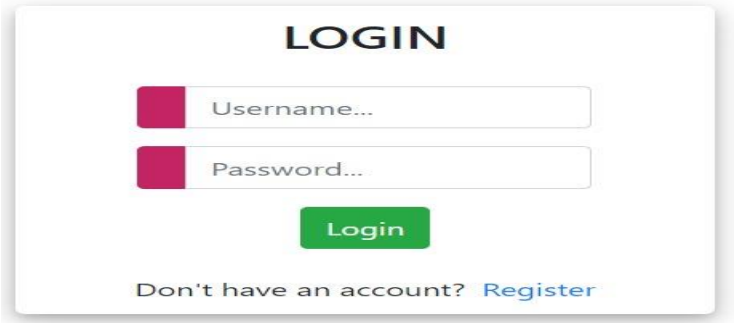


Figure 4-21 Figure 4 20 Login page display

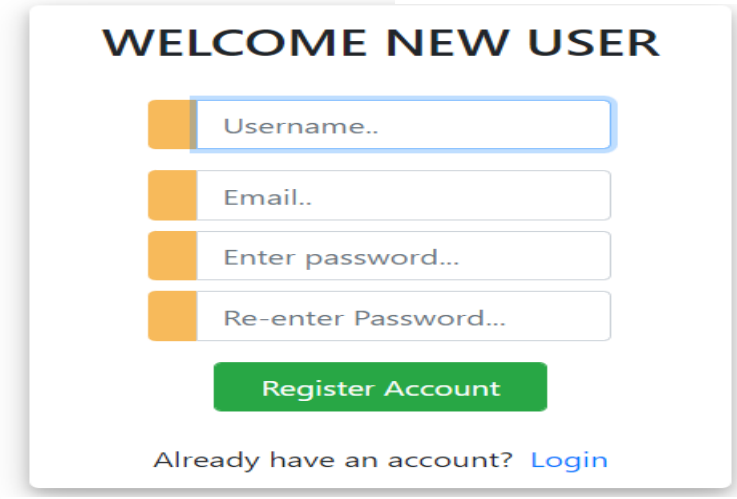
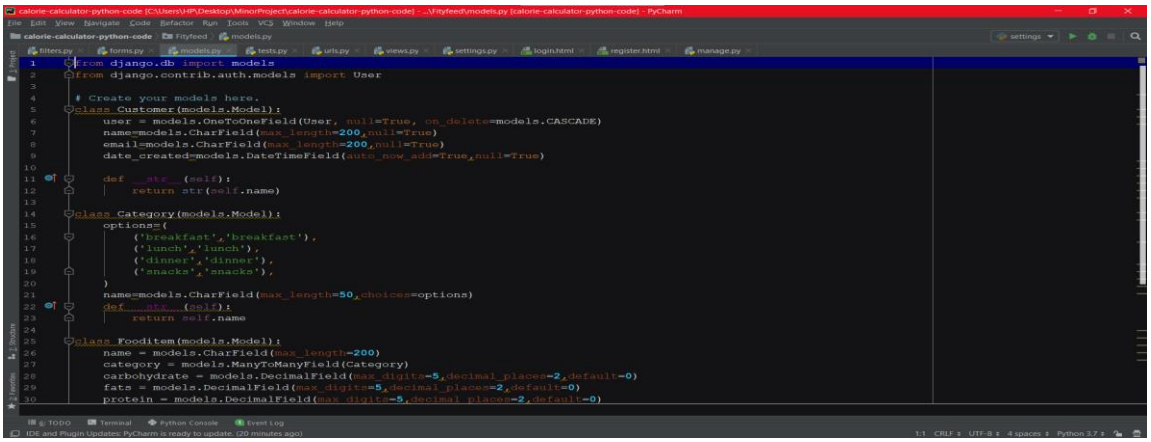


Figure 4-20 Creating account

4.7.3 Training Model:

The training of the web model is put on work here with the help of the Django which makes it very easier because of its flexible toolkits which helps in training the real-time models very efficiently [2].



```
from django.db import models
from django.contrib.auth.models import User

# Create your models here.
class Customer(models.Model):
    user = models.OneToOneField(User, null=True, on_delete=models.CASCADE)
    name=models.CharField(max_length=200,null=True)
    email=models.CharField(max_length=200,null=True)
    date_created=models.DateTimeField(auto_now_add=True,null=True)

    def __str__(self):
        return str(self.name)

class Category(models.Model):
    options=(
        ('breakfast','breakfast'),
        ('lunch','lunch'),
        ('dinner','dinner'),
        ('snacks','snacks'),
    )
    name=models.CharField(max_length=50,choices=options)
    def __str__(self):
        return self.name

class FoodItem(models.Model):
    name = models.CharField(max_length=200)
    category = models.ManyToManyField(Category)
    carbohydrate = models.DecimalField(max_digits=5,decimal_places=2,default=0)
    fats = models.DecimalField(max_digits=5,decimal_places=2,default=0)
    protein = models.DecimalField(max_digits=5,decimal_places=2,default=0)
```

Figure 4-23 Training models backend

4.7.4 Deploying the real-time model:

By starting the manage.py file on windows PowerShell we will be able to see all the functionalities of our project and by running the py.runserver command the server will

```python
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

from django.core.management.commands import runserver


def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'calorieCalc.settings')
    import django
    django.setup()

    try:
        from django.core.management import execute_from_command_line
        runserver.default_port = "8000"

    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()

```

Figure 4-24 py.manage file

load up on which the application will run and then we can use our application [1].

```python
from django.db import models
from django.contrib.auth.models import User

# Create your models here.
class Customer(models.Model):
    user = models.OneToOneField(User, null=True, on_delete=models.CASCADE)
    name=models.CharField(max_length=200,null=True)
    email=models.CharField(max_length=200,null=True)
    date_created=models.DateTimeField(auto_now_add=True,null=True)

    def __str__(self):
        return str(self.name)

class Category(models.Model):
    options=(
        ('breakfast','breakfast'),
        ('lunch','lunch'),
        ('dinner','dinner'),
        ('snacks','snacks'),
    )
    name=models.CharField(max_length=50,choices=options)
    def __str__(self):
        return self.name

class Fooditem(models.Model):
    name = models.CharField(max_length=200)
    category = models.ManyToManyField(Category)
    carbohydrate = models.DecimalField(max_digits=5,decimal_places=2,default=0)
    fats = models.DecimalField(max_digits=5,decimal_places=2,default=0)
    protein = models.DecimalField(max_digits=5,decimal_places=2,default=0)
```

Figure 4-25 models.py file

models.py | apps.py | decorators.py

```python
from django.http import HttpResponse
from django.shortcuts import redirect

def unauthorized_user(view_func):
    def wrapper_func(request,*args,**kwargs):
        if request.user.is_authenticated:
            return redirect('home')
        else:
            return view_func(request,*args,**kwargs)
    return wrapper_func

def allowed_users(allowed_roles=[]):
    def decorator(view_func):
        def wrapper_func(request,*args,**kwargs):
            group=None
            if request.user.groups.exists():
                group=request.user.groups.all()[0].name
            if group in allowed_roles:
                return view_func(request,*args,**kwargs)
            else:
                return HttpResponse("<h1>You are not allowed to access this page</h1>")
        return wrapper_func
    return decorator

def admin_only(view_func):
    def wrapper_func(request,*args,**kwargs):
        group=None
        if request.user.groups.exists():
            group=request.user.groups.all()[0].name
        if group=='user':
```

You are using the Django
PyCharm Professional Edit

Figure 4-26 decoators.py file

models.py | apps.py | decorators.py | admin.py | tests.py | views.py

```python
from django.shortcuts import render,redirect
from django.http import HttpResponse
from django.contrib.auth.models import User
from .models import *
from .forms import *
from django.contrib import messages
from django.contrib.auth import authenticate,login,logout
from django.contrib.auth.decorators import login_required
from .decorators import *
from django.contrib.auth.models import Group
from .filters import fooditemFilter
import os
# Create your views here.

@login_required(login_url='login')
@admin_only
def home(request):
    breakfast=Category.objects.filter(name='breakfast')[0].fooditem_set.all()[:5]
    lunch=Category.objects.filter(name='lunch')[0].fooditem_set.all()[:5]
    dinner=Category.objects.filter(name='dinner')[0].fooditem_set.all()[:5]
    snacks=Category.objects.filter(name='snacks')[0].fooditem_set.all()[:5]
    customers=Customer.objects.all()
    context={'breakfast':breakfast,
             'lunch':lunch,
             'dinner':dinner,
             'snacks':snacks,
             'customers':customers,
            }
    return render(request,'main.html',context)
```

You are using the Djang
PyCharm Professional Ed

Figure 4-27 view.py file

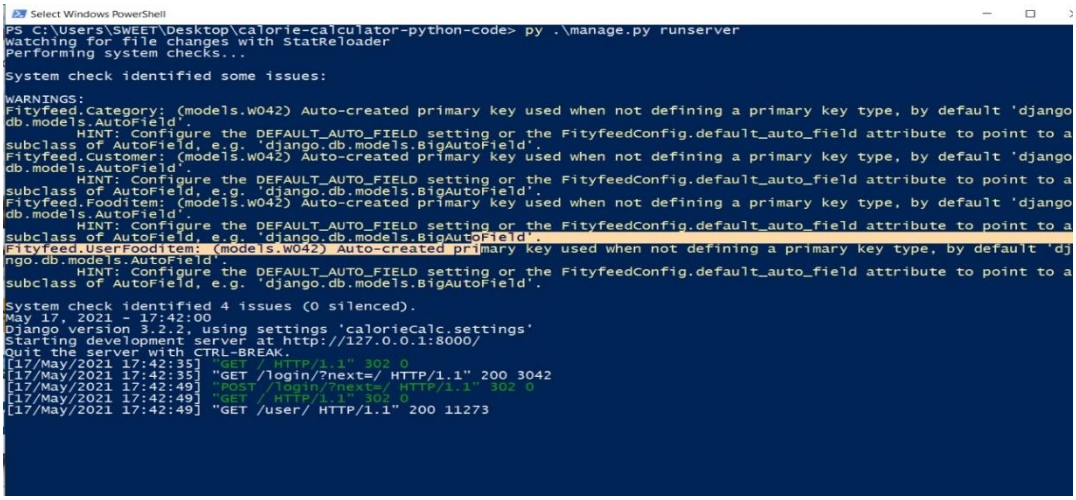


Figure 4-28 PowerShell working

4.7.5 Running Application snippets:

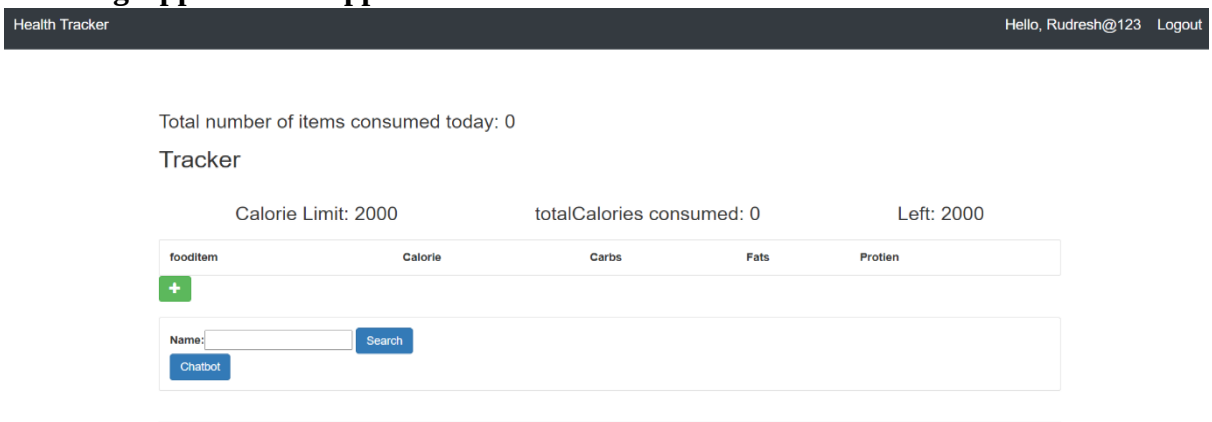


Figure 4-29 Main page (i)

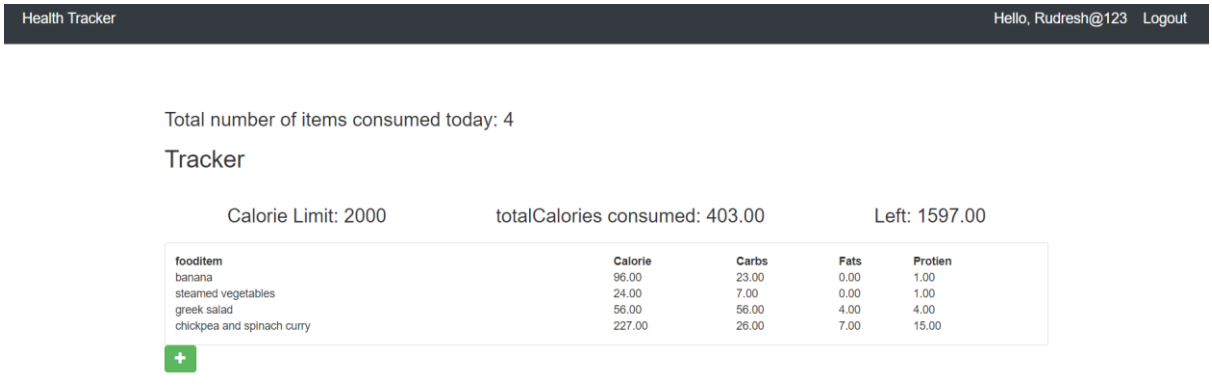


Figure 4-30 Main page (ii)

Available items:

fooditem	Calorie	Carbs	Fats	Protien
banana	96.00	23.00	0.00	1.00
apple	56.00	14.00	0.00	0.00
sprouts	152.00	13.00	8.00	7.00
protein shake	262.00	12.00	6.00	40.00
green tea	0.00	0.00	0.00	0.00
black coffee	0.00	0.00	0.00	0.00
coffee	107.00	16.00	3.00	4.00
oatmeal	346.00	60.00	6.00	13.00
honey green tea	12.00	3.00	0.00	0.00
nuts	105.00	3.00	9.00	3.00
greek yogurt	130.00	5.00	10.00	5.00
egg	69.00	0.00	5.00	6.00

Figure 4-31 Nutrition Table (i)

Pasta	359.00	72.00	2.00	12.00
Mexican rice	160.00	30.00	2.00	3.00
Sandwiches	720.00	77.00	25.00	40.00
breakfast sandwiches	200.00	26.00	6.00	12.00
Ham and cheese baguette	538.00	38.00	34.00	20.00
cupcakes	201.00	29.00	9.00	1.00
potato chps	150.00	14.00	10.00	1.00
Baked sweet potatoes	170.00	43.00	0.00	2.00
Ice cream	195.00	21.00	11.00	3.00
Banana ice cream	278.00	60.00	2.00	5.00
chocolate cupcakes	142.00	29.00	2.00	2.00

This is designed by Rudresh & Kunal

Figure 4-32 Nutrition Table (ii)

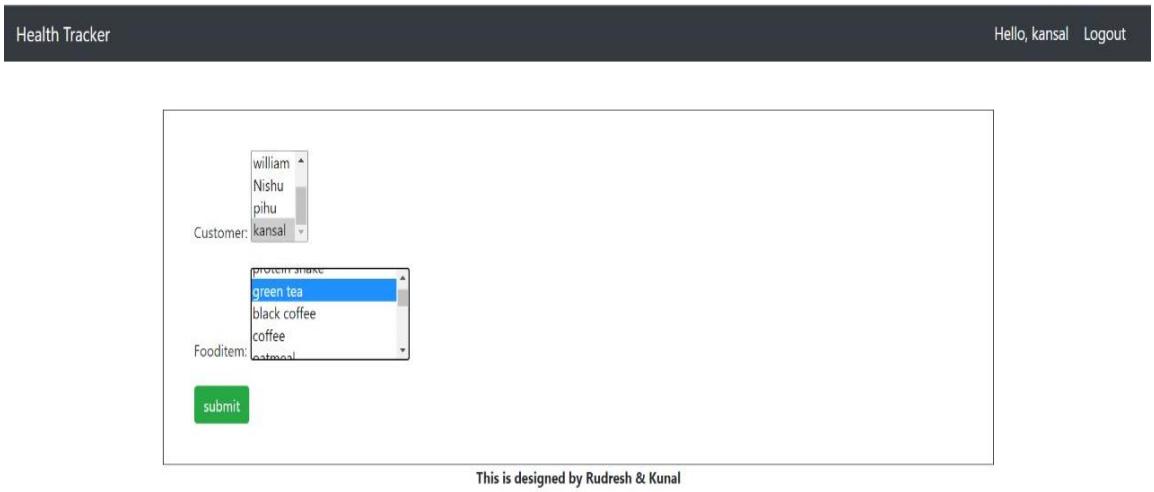


Figure 4-33 Addition of meals

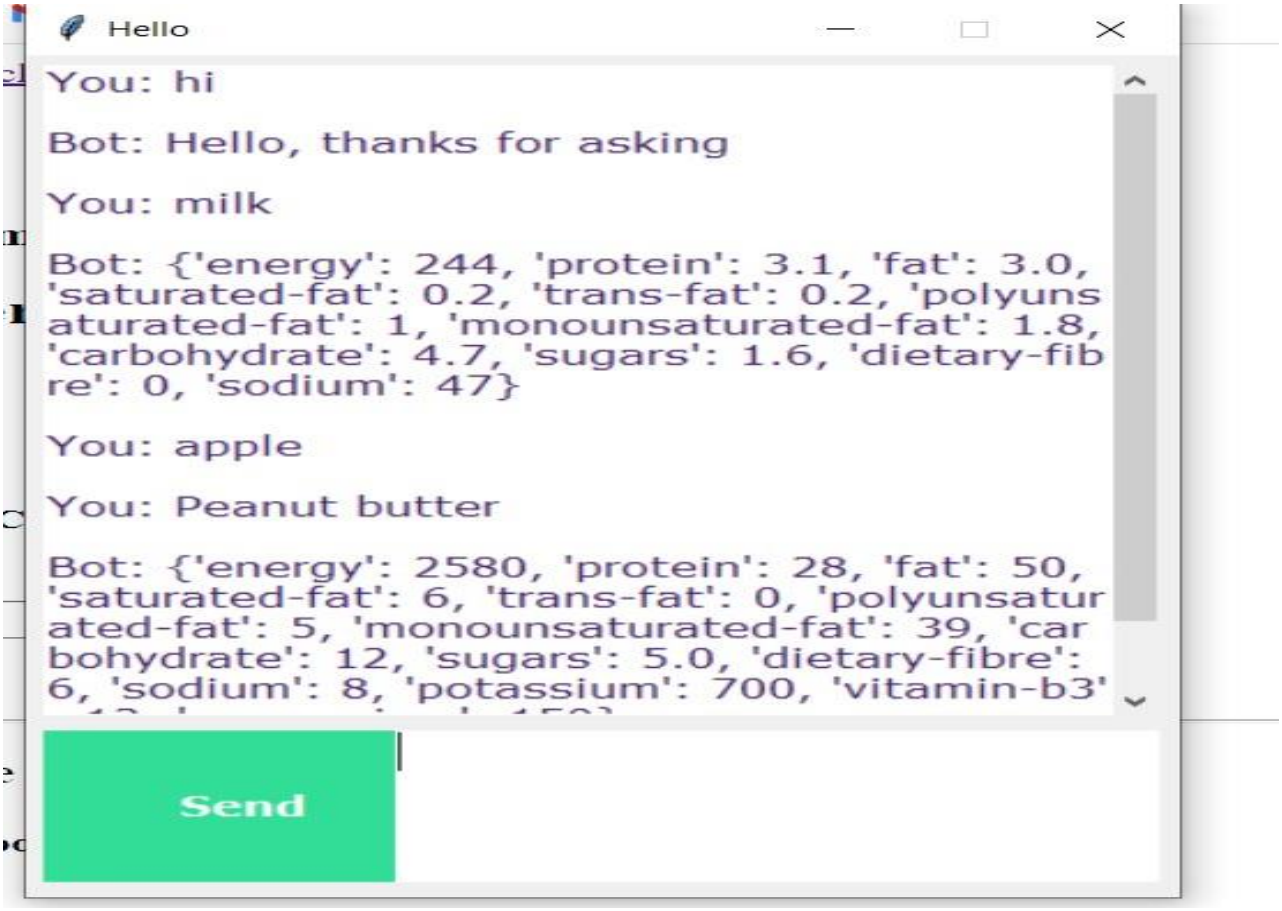


Figure 4-34 Chatbot

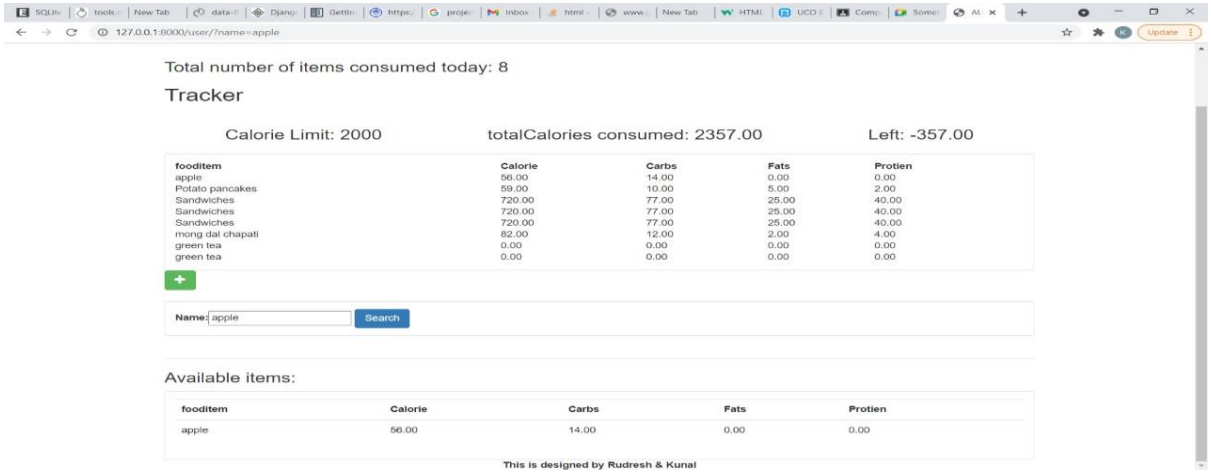


Figure 4-36 Searching food item

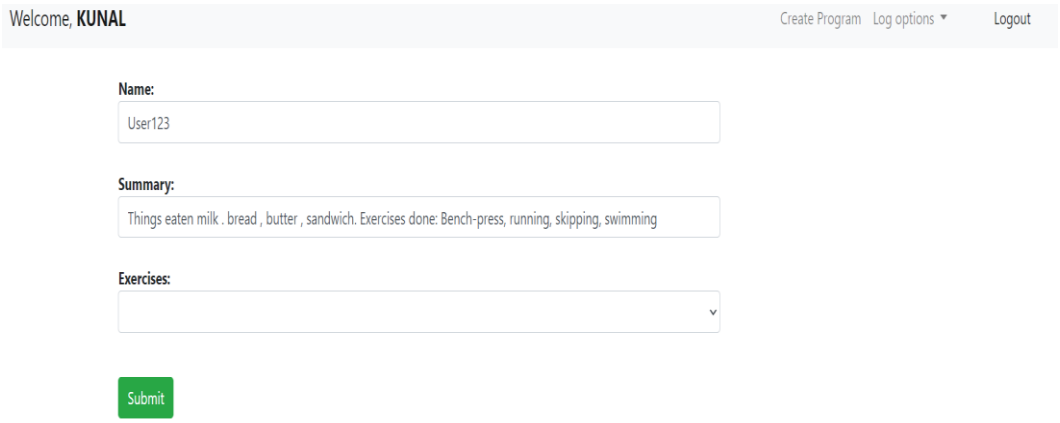


Figure 4-35 Figure 4 34 Calorie Burned Tracker

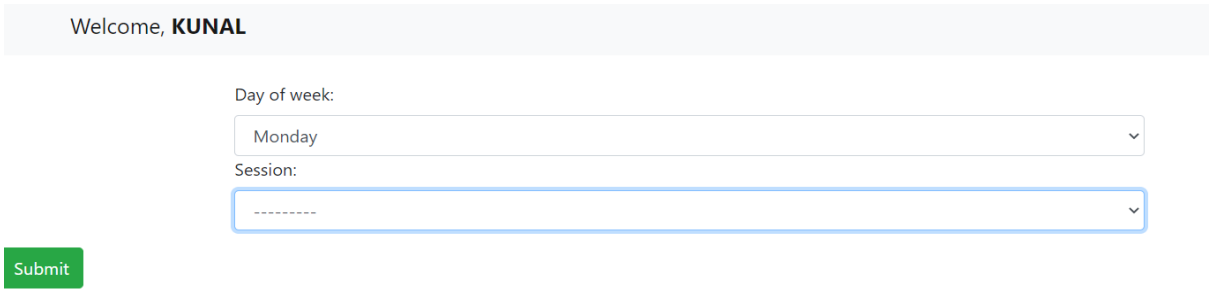


Figure 4-37 Workout tracker interface

4.8 Major Modules used in this Automated Fitness Tracker

Table 4 Modules used

SNo	Name of the module	Definition
1.	**Pandas**	This python module is very efficient module for data science as it helps in creating, editing and deleting huge databases very easily. Data frames are created so that the whole database isn't altered under the processing. Its vast functionalities and database editing powers are very helpful in refining and cleaning the data. It helps in cleaning a huge set of databases in no time [3].
2.	**NumPy**	It is a python module used to do mathematical calculation and helps in removing irrelevant data with its vast set of functionalities. It helps in easy manipulation of the data set [5].
3.	**FuzzyWuzzy:**	This python module helps in speedy diagnose of the dataset and provide with more efficient results [4].
4.	**Django Framework:**	It is a high-level web framework provided to create online web applications without any fuss and provides us with excellent python functionalities [2].
5.	**NLTK** .	It is called Natural Language toolkit. It helps applications to work with natural processing languages. It uses the concept of tokenization (in which we divide our data into tokens and then get information about it), classification, etc. [6].
6.	**Keras**	It is used for developing and evaluating deep learning models and performs as an interface for artificial neural networks [7].
7.	**TensorFlow**	It is widely used in artificial intelligence and deep learning concepts as it performs classification, perception, understanding, discovering, prediction and creation. It performs fast numerical computations [8].
8.	**Pickle**	The pickle library in python is used to convert the python object into stream of bytes and store it in file. It is used for storing the multiple python objects [9].
9.	**JSON**	JSON library in python is used to connect are code to JSON file which helps fetch, manipulate data from the file. It holds the power of conversion of objects into strings. It is very much similar to dictionary data type but it browses data more quickly and conventionally [10].
10.	**Tkinter**	Tkinter provides a wide variety of functionalities for developing Graphical User Interface for python projects. It is very simple and faster way to create a GUI in python [11].
11.	**Random**	It is used to generate random seeds which our used in our train and test models in order to gather more information and improve the accuracy of our resulting models [12].

Chapter 5 CONCLUSIONS

5.1 Discussions on the results achieved

5.1.1 Login Page:

This is our login page; in this you need to enter your details such as username and password if you don't have one you need to click on below register button to open the new user registration page in this you need to enter following details:

1. Your unique username
2. And then you need to enter your email id
3. And at last, enter your 8-digit password
4. Then click on register now button to create your account'
5. Now you will receive a message on your screen that you have successfully created your account.
6. Again, you will see a login page now enter your username and password and you will be logged into our web application.

Figure 5-1 Login

Figure 5-2 Sign up

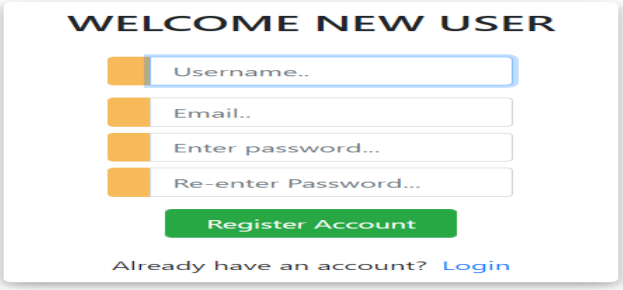


Figure 5-3 Create a new account

5.1.2 Web Application:

1. Now once you have logged in you will see our web application been launched.
2. Now you will see various list of food items below like apple,
3. Mango, Banana and various other things along with that you will find the fat, calories, proteins and carbohydrate composition on right side of name of food items
4. Above the food composition table, you will find the search button through which you can search your food item and select the item. Now you will see once you have selected required food item you will see that the selected food item is displayed in the tracker now as an average human consumes 2000 calories so are tracker keeps the count of calories you have consumed and the item you have eaten in a day which helps one to follow a healthy routine by keeping track of fats, proteins and carbohydrate that one consumes in a day.
5. Now there this a add items button below the tracker table that helps you to add more items in your daily calorie calculator tracker and automatically updates the total calorie consumption.
6. A green chatbot button will open the chatbot window through which we can start our conversation with the chatbot. Here we can find out how to use our application and can ask about all the ingredients of any particular items instead of searching it in the whole database. Also, it tells us whether the item is healthy or not.

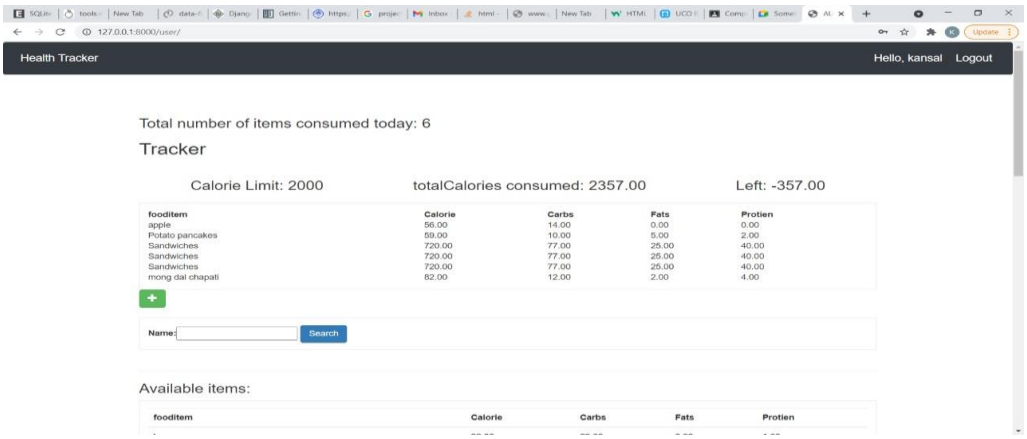
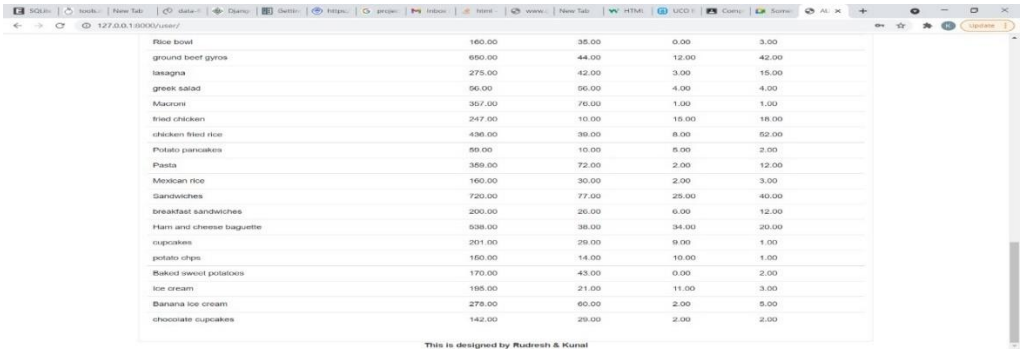


Figure 5-4 Running Application



Rice bowl	160.00	35.00	0.00	3.00
ground beef gyros	650.00	44.00	12.00	42.00
lasagna	275.00	42.00	3.00	15.00
greek salad	56.00	56.00	4.00	4.00
Macroni	367.00	76.00	1.00	1.00
fried chicken	247.00	10.00	15.00	15.00
chicken fried rice	436.00	39.00	8.00	52.00
Potato pancakes	59.00	10.00	5.00	2.00
Pasta	359.00	72.00	2.00	12.00
Mexican rice	160.00	30.00	2.00	3.00
Sandwiches	720.00	77.00	25.00	40.00
breakfast sandwiches	200.00	26.00	6.00	12.00
Ham and cheese baguette	538.00	38.00	34.00	20.00
cupcakes	201.00	29.00	9.00	1.00
potato chips	160.00	14.00	10.00	1.00
Baked sweet potatoes	170.00	43.00	0.00	2.00
Ice cream	165.00	21.00	11.00	3.00
Banana ice cream	278.00	60.00	2.00	5.00
chocolate cupcakes	142.00	29.00	2.00	2.00

This is designed by Rudresh & Kunal

Figure 5-5 Various food items

7. When we go to track our calories, we have to enter the following details:
    - Name of user
    - Summary of what user has done in his entire day like type exercises user has done, type diet user has taken etc.
    - Third option is exercises user has done in his entire day.
    - We have to select the date and then select the workout plan accordingly.

5.2 Machine learning algorithm used in Chatbot:

By creating our chatbot we got know a lot about the supervised and unsupervised machine learning algorithms. The learning process of a chatbot is based on learning a lot by Natural Language Processing (NLP) of the input we give to it.

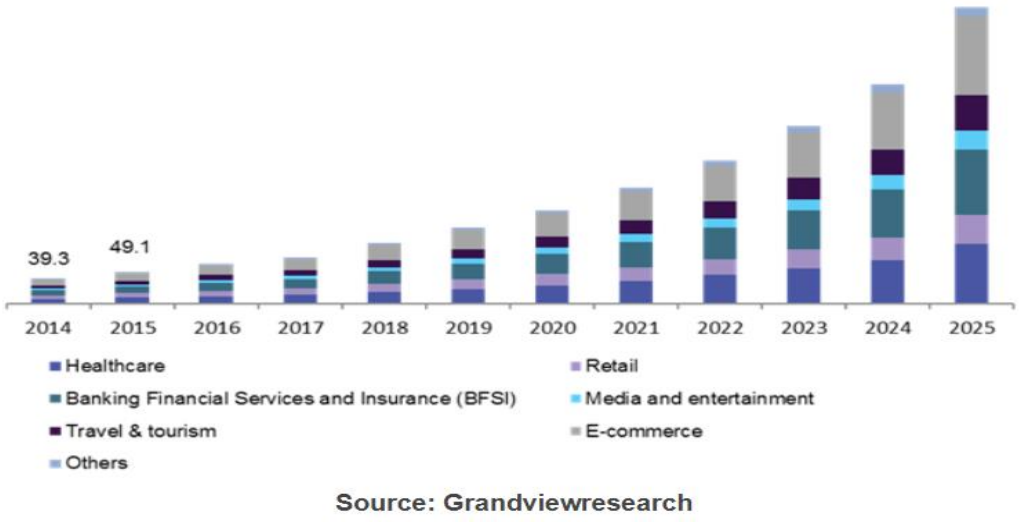


Figure 5-6 graph of chatbot application use in various sectors over time

Chatbot is the new future of our society as it will surely takes upon the unnecessary work burden over the helping desks of many MNCs and many online shopping applications like flipkart, amazon, etc. but it is also being used in various other applications such as health calculators, social media platforms, Spotify, android and iOS systems etc.

Thus, the use of these robotic helping hands in the market is increasing nowadays with an annual growth rate of 24.3% according to the Grand View Research report. This also means that the updated and more modern chatbots are much in need which can be made from artificial intelligence and by applying machine learning algorithms over the training models of such chatbots to provide us with some superb models.

Chatbot are made from a more sophisticated branch of machine learning which is based on artificial neural networks, which replicates working of the human brain. We use many types of architectures here like Deep Neural Network, Deep belief Network and Recurrent.

5.3 Application of Automated Fitness Tracker

1. Now during this time of Covid crisis as everyone is concerned about their health calorie calculator can come in very handy as everyone can keep track of their proteins, fats etc.
2. Now as many food apps are now concerned about their customers health, they can implement a health tracker so that as person can keep track of what person orders.
3. Calorie calculator comes handy for a gym person so that he knows how much calories he needs to consume and how much he needs to burn.
4. Calorie calculator can come in use for a ill person as he can know the items to eat with good health benefits.
5. Calorie calculator users can check their food consumption according to their taste, age etc. and keep a daily monitor of it.
6. It helps in finding workouts of professional athletes, coaches and trainers without paying for any subscription.
7. User gets perfect workout plans and are able to perform such exercises without any injury.

5.4 Limitations of Automated Fitness Tracker

1. Calorie calculator doesn't work on cooked food and that is major drawback of calorie calculator for e.g., everyone has their own style and ingredients of making sandwich which results in different composition of fats, proteins, carbohydrates etc.
2. Calorie calculator being a web application one needs to have a proper internet connection to access the calorie calculator.
3. As the fat, protein of various things depends on the quantity and quality of product used which can lead to more far approximations so that's the major drawback of calorie calculator.
4. As, it is under development process so there are many bugs which are needed to be fixed causing disrupt in smooth running of the application.

5.5 Future of Automated Fitness Tracker

1. Calorie calculator can be improvised with a SMS alarm system which can give a person alert if he exceeds his daily calorie limit.
2. Calorie calculator can have a different gym section in which a gym going person can maintain his logs and can find out how much progress they are making.
3. Calorie calculator can be real time health tracker which can be upgraded so that person can input his health conditions and our app can suggest him items that can help him improve his health.
4. Calorie calculator can be modified with a calorie burn feature that suggest exercises according to calories person want to burn.

REFERENCES

1. Websites:

[1] S. Yegulalp, "What is Python? Powerful, intuitive programming", *InfoWorld*, 2022. [Online]. Available: https://www.infoworld.com/article/3204016/what-is-python-powerful-intuitive-programming.html. [Accessed: 15- May- 2022].

[2] "Django introduction - Learn web development | MDN", *Developer.mozilla.org*, 2022. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction. [Accessed: 15- May- 2022].

[3] "A Brief Introduction to Pandas", *mygreatlearning.com*, 2022. [Online]. Available: https://www.mygreatlearning.com/blog/python-pandas-tutorial/. [Accessed: 15- May- 2022].

[4] "Fuzzy String Matching in Python", *Medium*, 2022. [Online]. Available: https://towardsdatascience.com/fuzzy-string-matching-in-python-68f240d910fe. [Accessed: 15- May- 2022].

[5] "*Array programming with NumPy*", Harris, C. R., 2020. [Online]. Available: https://www.nature.com/articles/s41586-020-2649-2. [Accessed: 15- May- 2022].

[6] "*Introduction to Deep Learning with Keras - Towards Data Science*", Tanner, G., 2019 [Online]. Available: https://towardsdatascience.com/introduction-to-deep-learning-with-keras-17c09e4f0eb2. [Accessed: 15- May- 2022].

[7] "*What is TensorFlow? The machine learning library explained*", Yegulalp, S., 2019[Online]. Available: https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html. [Accessed: 15- May- 2022].

[8] **"*Understanding Python Pickling with example", GeeksforGeeks, 2019[Online]*. Available: https://www.geeksforgeeks.org/understanding-python-pickling-example/. [Accessed: 15- May- 2022].**

[9] **"*NLTK: A Beginners Hands-on Guide to Natural Language Processing",* Analytics Vidhya M, S., 2021[Online]. Available: https://www.analyticsvidhya.com/blog/2021/07/nltk-a-beginners-hands-on-guide-to-natural-language-processing/. [Accessed: 15- May- 2022].**

2) Journals:

[1] R. Bierbrier, V. Lo and R. Wu, "Evaluation of the Accuracy of Smartphone Medical Calculation Apps", *Journal of Medical Internet Research*, vol. 16, no. 2, p. e32, 2014. Available: 10.2196/jmir.3062.

[2] S. Dorman, "Web-Based Health Calculators", *Journal of School Health*, vol. 71, no. 10, pp. 500-501, 2001. Available: 10.1111/j.1746-1561. 2001.tb07291. x.

3) Articles:

[1] "*How to handle JSON in Python? - Towards Data Science",* Prabhu, T. N., 2020 [Online]. Available: https://towardsdatascience.com/how-to-handle-json-in-python-d877125df39b. [Accessed: 15- May- 2022].

[2] "*Python GUI – tkinter",* GeeksforGeeks, 2020[Online]. Available: https://www.geeksforgeeks.org/python-gui-tkinter/. [Accessed: 15- May- 2022].

[3] *"Python Random Module",* GeeksforGeeks, 2021[Online]. Available: https://www.geeksforgeeks.org/python-random-module/. [Accessed: 15- May- 2022].

[4] "Chatbot Market Size Worth $2,485.7 Million By 2028", *Grandviewresearch.com*, 2022. [Online]. Available: https://www.grandviewresearch.com/press-release/global-chatbot-market. [Accessed: 15- May- 2022].

4) Datasets:

[1] "World Food Programme", *wfp.org*, 2022. [Online]. Available: https://data.humdata.org/organization/wfp. [Accessed: 15- May- 2022].

[2] "Physical Education/Health Education | Manitoba Education and Early Childhood Learning", *Edu.gov.mb.ca*, 2022. [Online]. Available: https://www.edu.gov.mb.ca/k12/cur/physhlth/. [Accessed: 15- May- 2022].

APPENDICES

SOURCE CODE SNIPPETS:

urls.py 5 decorators.py 2

E: > calorie-calculator-python-code > Fityfeed > decorators.py > ...

```python
from django.http import HttpResponse
from django.shortcuts import redirect

def unauthorized_user(view_func):
    def wrapper_func(request,*args,**kwargs):
        if request.user.is_authenticated:
            return redirect('home')
        else:
            return view_func(request,*args,**kwargs)
    return wrapper_func

def allowed_users(allowed_roles=[]):
    def decorator(view_func):
        def wrapper_func(request,*args,**kwargs):
            group=None
            if request.user.groups.exists():
                group=request.user.groups.all()[0].name
            if group in allowed_roles:
                return view_func(request,*args,**kwargs)
            else:
                return HttpResponse("<h1>You are not allowed to access this page</h1>")
        return wrapper_func
    return decorator

def admin_only(view_func):
    def wrapper_func(request,*args,**kwargs):
        group=None
        if request.user.groups.exists():
            group=request.user.groups.all()[0].name
        if group=='user':
            return redirect('userPage')
        if group=='admin':
            return view_func(request,*args,**kwargs)
    return wrapper_func

```

Figure 0-1 decorators.py file

E: > calorie-calculator-python-code > Fityfeed > urls.py > ...

```python
from django.urls import path,include
from . import views
from django.contrib.auth import views as auth_views

urlpatterns = [
    path('',views.home,name='home'),
    path('user/',views.userPage,name='userPage'), #userPage
    path('product/',views.fooditem,name='fooditem'),
    path('createfooditem/',views.createfooditem,name='createfooditem'),
    path('register/',views.registerPage,name='register'),
    path('login/',views.loginPage,name='login'),
    path('logout/',views.logoutUser,name='logout'),
    path('addFooditem/',views.addFooditem,name='addFooditem'),
    path('openchatbot/',views.openchatbot,name='openchatbot'),
     path('reset_password/',
         auth_views.PasswordResetView.as_view(),
         name="reset_password"),

    path('reset_password_sent/',
         auth_views.PasswordResetDoneView.as_view(),
         name="password_reset_done"),

    path('reset/<uidb64>/<token>/',
         auth_views.PasswordResetConfirmView.as_view(),
         name="password_reset_confirm"),

    path('reset_password_complete/',
         auth_views.PasswordResetCompleteView.as_view(),
         name="password_reset_complete"),
]
```

Figure 0-2 urls.py file

E: > calorie-calculator-python-code > Fityfeed > forms.py > ...

```python
from django.forms import ModelForm
from .models import *
from django.contrib.auth.forms import UserCreationForm

class fooditemForm(ModelForm):
    class Meta:
        model=Fooditem
        fields="__all__"

class addUserFooditem(ModelForm):
    class Meta:
        model=UserFooditem
        fields="__all__"

class createUserForm(UserCreationForm):
    class Meta:
        model=User
        fields=['username','email','password1','password2']
```

Figure 0-3 forms.py file

```python
def userPage(request):
    user=request.user
    cust=user.customer
    fooditems=Fooditem.objects.filter()
    myfilter = fooditemFilter(request.GET,queryset=fooditems)
    fooditems=myfilter.qs
    total=UserFooditem.objects.all()
    myfooditems=total.filter(customer=cust)
    cnt=myfooditems.count()
    querysetFood=[]
    for food in myfooditems:
        querysetFood.append(food.fooditem.all())
    finalFoodItems=[]
    for items in querysetFood:
        for food_items in items:
            finalFoodItems.append(food_items)
    totalCalories=0
    for foods in finalFoodItems:
        totalCalories+=foods.calorie
    CalorieLeft=2000-totalCalories
    context={'CalorieLeft':CalorieLeft,'totalCalories':totalCalories,'cnt':cnt,'foodlist':finalFoodItems,'fooditem':fooditems,'myfilter':myfilter}
    return render(request,'user.html',context)
```

Figure 0-4 manage.py file