A Project Phase-II Report on

NUTRITIONAL FOOD RECOMMENDATION SYSTEM

By

D. Mrunalini Reddy 18251A0578

P. Sri Varshika 18251A0580

G. Anusha 18251A05A1

J. Ushasini 18251A05A2

Under the guidance of

Mr. Garigipati Naga Babu

Assistant Professor

Department of Computer Science and Engineering

G. NARAYANAMMA INSTITUTE OF TECHNOLOGY & SCIENCE

(Autonomous) (For Women)

Shaikpet, Hyderabad 500104.

Affiliated to

Jawaharlal Nehru Technological University Hyderabad

Hyderabad - 500085

June, 2022

A Project Phase-II Report on

NUTRITIONAL FOOD RECOMMENDATION SYSTEM

Submitted to the Department of Computer Science & Engineering, GNITS in the partial fulfillment of the academic requirement for the award of B.Tech (CSE) under JNTUH

By

D. Mrunalini Reddy	**18251A0578**
P. Sri Varshika	**18251A0580**
G. Anusha	**18251A05A1**
J. Ushasini	**18251A05A2**

Under the guidance of

Mr. Garigipati Naga Babu

Assistant Professor

Department of Computer Science and Engineering

G. NARAYANAMMA INSTITUTE OF TECHNOLOGY & SCIENCE

(Autonomous) (For Women)

Shaikpet, Hyderabad- 500104.

Affiliated to

Jawaharlal Nehru Technological University Hyderabad

Hyderabad - 500085

June, 2022

G. Narayanamma Institute of Technology & Science
(Autonomous) **(For Women)**
Shaikpet, Hyderabad – 500 104.

Department of Computer Science & Engineering

Certificate

This is to certify that the Project phase-II report on **"Nutritional Food Recommendation System"** is a bonafide work carried out by **D. Mrunalini Reddy (18251A0578), P. Sri Varshika (18251A0580), G. Anusha (18251A05A1), J. Ushasini (18251A05A2)** in the partial fulfillment for the award of B.Tech degree in Computer Science & Engineering, G. Narayanamma Institute of Technology & Science (Autonomous), Hyderabad affiliated to Jawaharlal Nehru Technological University, Hyderabad under our guidance and supervision.

The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Internal Guide	**Head of the Department**
Mr. Garigipati Naga Babu	**Dr. M. Seetha**
Assistant Professor	**Professor and Head**
Department of CSE	**Department of CSE**

External Examiner

Acknowledgements

We would like to express our sincere thanks to **Dr K Ramesh Reddy**, Principal, GNITS, for providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. M Seetha,** Professor & Head, Department of CSE, GNITS for all the timely support and valuable suggestions during the period of our project.

We are extremely thankful to **Dr. N. Kalyani,** Professor, overall project coordinator, Department of CSE, GNITS for all the valuable suggestions and guidance during the period of our project.

We are extremely thankful to **Dr. O. Obulesu**, Associate Professor, **Dr.G. Malini Devi**, Associate Professor, **Mrs.K. Sneha Reddy** Assistant Professor, Department of CSE, GNITS, project coordinators for their encouragement and support throughout the project.

We are extremely thankful and indebted to our internal guide, **Mr. Garigipati Naga Babu,** Assistant professor, Department of CSE**,** GNITS for his constant guidance, encouragement and moral support throughout the project.

Finally, we are extremely thankful to all the faculty members and staff of CSE Department who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

D. Mrunalini Reddy 18251A0578
P. Sri Varshika 18251A0580
G. Anusha 18251A05A1
J. Ushasini 18251A05A2

ABSTRACT

The patient's nutrition is a key factor in their treatment process in the healthcare industry. Every user has their own specific nutritional needs and requirements. An appropriate nutrition policy can therefore help the patient's recovery process and alleviate possible symptoms. Food recommender systems are platforms that offer personalized suggestions of recipes to users. However, there is a lack of usage of recipe recommendation systems in the healthcare sector. The present project aims to develop a platform for an intelligent planning of the user's meals, based on their clinical conditions.

In this context, this project has emerged, consisting in the development and exploration of a new platform for meal planning, incorporating machine learning algorithms for decision support in the recommendation system. Based on the clinical parameters of the patient this project aims at generating accurate diet plans using several methodologies that are currently available and viable.

SYSTEM REQUIREMENTS:

Software Requirements:

Platform - Windows 10

Programming Language - Python

Development Tool - Jupyter Notebook

Hardware Requirements:

Name of the Processor - Windows 10 intel® core™ i5

RAM Capacity - 8GB

Hard Disk Capacity -1TB

Table of Contents

List of figures

List of Tables

1.INTRODUCTION

1.1 MOTIVATION

A healthy and balanced diet is a key element in everyone's life. A balanced diet is one which contains sufficient amounts of all the nutrients namely carbohydrates, fats, proteins, vitamins, minerals, sugar etc. which are necessary for keeping a person in perfect health. 'Dosha', a Sanskrit word which means the root of a body helps us to infer that if the Dosha of our body i.e., the root causes are disturbed then it may result in a number of diseases. Thus, maintaining a healthy diet is essential for everyone.

One of the major factors for a healthy life is daily diet and food, specifically, for the people suffering from some minor or major diseases. eHealth initiatives and research efforts aim to offer various pervasive applications for novice end users to improve their health. Various studies depict that inappropriate and inadequate intake of diet is the major reasons of various health issues and diseases. A study conducted by World Health Organization (WHO) estimates that around 30% of the total population of the world is suffering from various diseases, and 60% deaths each year in children are related to malnutrition. Another study by WHO reports that inadequate and imbalanced intake of food causes around 9% of heart attack deaths, about 11% of ischemic heart disease deaths, and 14% of gastrointestinal cancer deaths worldwide. Moreover, around 0.25 billion children are suffering from Vitamin-A deficiency, 0.2 billion people are suffering from iron deficiency (anemia), and 0.7 billion people are suffering from iodine deficiency.

The World Health Organization estimates that non-communicable diseases such as cardiovascular diseases, cancer, chronic respiratory diseases and diabetes, are responsible for 63% of all deaths worldwide. Furthermore, it also points out that such diseases are preventable through effective interventions that tackle shared risk factors such as the unhealthy diets. In this context, personalized nutrition can benefits consumers to adhere to a healthy, pleasurable, and nutritional diet when it is closely associated to individual parameters such as the physical and psychological characteristics including health status, the consumer's needs and preferences, behavior, lifestyle.

Personalized nutrition can be used for different target groups from healthy people to patients such as malnourished people, vulnerable groups, people with allergies or non-communicable diseases, including cancer. Personalized nutrition has been formally defined as the healthy eating advice, tailored to suit an individual based on personal health status, lifestyle, nutrients intake. Several computational solutions have been proposed with the goal of healthy eating advice. The menu planning problem has been focused since more than 50 years ago. However, recently it was and still is an open and very active research problem, focused on adding personalization capabilities to the menu generation frameworks.

A recommender system, or a recommendation system (sometimes replacing “system” with a synonym such as platform or engine), is a subclass of information filtering system that seeks to predict the “rating” or “preference” a user would give to an item. They are primarily used in commercial applications. The recommendation process has basically three stages that are Information Collection Phase, Learning Phase and Recommendation Phase. The information is firstly collected about a particular problem and the various solutions related to that problem are categorized. After the collection of information Learning Phase comes in which various conclusions are made out of that information which is gathered and in last phase i.e., Recommendation Phase an output is given in which various recommendations are made. In our system since it is a diet recommendation system so the recommendations will be about the diet plan like what all things you should eat, what is your BMI (Body Mass Index) which states whether you are healthy, overweight, or under-weight.

It is remarkable that the food Recommendation Systems (RSs) are relatively a recent domain whose state of the art has been analyzed in pointing out that its research challenges are related to the collection of user information, the gathering of nutritional information from foods and recipes, and the changing of eating behaviors. In our proposal it focuses on the development of a food recommendation model that integrates both nutritional and user preferences-related information, the use of feedback-based user profiling methods, in the food recommendation domain.

The proposed model is a diet recommendation system using machine learning algorithms which will suggest food items to the user from a predefined dataset depending upon user preferences for healthy food or food for weight loss or food for weight gain.

1.2 LITERATUTE SURVEY

Considering the busy life of people in the present world, it results in situations when people tend to skip proper meals or have food at irregular intervals. In such a scenario, it becomes essential for someone to provide them with a proper diet plan which needs to be strictly adhered to. For the same, people have personal dieticians which provide them with a proper food chart regularly to keep them fit. However, it is not possible for humans to handle many cases and analyze all the elements which are needed to be considered for providing a healthy and balanced diet. Keeping this in mind, an idea struck among the researchers to develops certain systems which may implement machine learning algorithms to gather the user data such as age, weight, height, diseases etc. and analyze them to provide a menu of food items to the user for the proper meals at the right time. Studying such systems and considering the importance of it, the idea is carried on with diet recommendation as the project.

A research cluster focused on building complex information models as base for the personalized services is available. These research works have been focused on the adaptive delivery of healthy diet plans to improve the quality of life of both healthy subjects and patients with diet-related chronic diseases [9], [11], [22]. With this purpose in mind, they have used flow charts supported by user answers to dynamic medical questionnaires [11], social semantic mobile framework to generate healthcare-related recommendation [9], as well as the use of ontologies for managing recipes, menus, and medical prescriptions [6].

Further key research works focused on extensive nutritional information modeling were developed by Espin et al. [22] focusing on helping elderly users to draw up their own healthy diet plans, and by Cioara et al. [20], where dietary knowledge is defined by nutritionists and encoded as a nutrition care process ontology. Eventually, Taweel [1] presents a distributed system that enables home care management in the context of self-feeding and malnutrition prevention, where bio-inspired algorithms are used in Food Menu Plans Generation and Diet-aware Food Ordering.

A second research cluster that tends to work over already available nutritional information sources, and is then focused on nutritional information processing, instead of prioritizing the data modeling task is also present. Some of these works face the nutritional recommendation as an optimization problem related to the healthy menu generation. In this way, the menu planning problems has been treated as an optimization scenario since more than 50 years ago [12]. However, in the last few years, there are still several research groups that use this approach as a mainstream solution, taking as base different optimization approaches such genetic algorithms [14], ant colony optimization [10], or a bacterial foraging optimization approach [2].

Beyond these approaches, there are other proposals in the nutritional information processing research cluster that do not consider optimization approaches because are based on some kind of ad-hoc heuristic for healthy menu generation. Here, there have been some researches focused on restaurant menu recommendation such as Ntalaperas et al. [7], focused on ranking dishes based on medical conditions, users' settings and preferences based on past rankings, but specifically focused on a restaurant menu. In a different direction, we detect a small group of research works focused on processing multimodal data, such as Nag et al. [16] propose a live personalized nutrition recommendation engine that uses multimodal contextual data including GPS location, barometer, and pedometer output to calculate a live estimate of the user's daily nutritional requirements, that are then used to rank the meals based on how well they fulfill the individual's nutritional needs.

In this direction, Ge et al. [15] propose a food recommender system developed on a mobile platform, which not only offers recipe recommendations that suit the user's preference but is also able to take the user's health into account, supported by wearable technologies. At last, it was also identified a research work focused on visual features of foods [13] for modeling individuals' nutritional

expectations, dietary restrictions, and fine-grained food preferences, but assuming a basic strategy to rank the nutritional appropriateness.

Eventually, Ribeiro et al. [8] create a content-based recommender system that manages a personalized weekly meal plan by calculating of nutritional requirements, following static criteria, such as separation of meat and fish, limitation in the repetition of foods, and other similar ones. Beyond these two clusters, Tran et al. [21], and Elsweiler et al. [4] recently analyzed the existing state-of-the-art in food recommender systems and discuss research challenges related to the development of future food recommendation technologies.

Thus, concluding that the current research challenges are related to the collection of user information, the gathering of nutritional information from food and recipes, the changing of eating behaviors, and the generating of bundle recommendations.

The previous analysis leads to the following conclusions:

i. Globally, the incorporation of nutritional concepts and principles in the computational models is not deep.
ii. Several works are not directly focused on the personalized nutrition aim, and only manage it as a component of larger health and wellbeing-related platforms.
iii. There are few works focused on the processing of the users' preferences, which is a key element in any personalization scenario.
iv. Furthermore, there are too few works (only three) managing both nutritional-aware and preference-aware information. However, in the three cases the preference gathering is focused on explicit user questions, and are not focused on a long-term user modeling. Two of them (Ntalaperas et al [7] and Ribeiro et al [8]) are ongoing research, and Yang et al. [13] although manage both kind of information, mostly support their research on exploiting visual food features.

The previous analysis evidences the necessity of a new food recommendation approach which integrates both nutritional and preference-based information. This is the goal of the current research. The proposed model is a diet recommendation system using machine learning algorithms which will suggest food items to the user from a predefined dataset depending upon user preferences for healthy food or food for weight loss or food for weight gain which integrates both nutritional and preference-based information.

1.3 A FOOD RECOMMENDER SYSTEM CONSIDERING NUTRITIONAL INFORMATION AND USER PREFERENCES

Good nutrition is an important part of leading a healthy lifestyle. Combined with physical activity, your diet can help you to reach and maintain a healthy weight, reduce your risk of chronic diseases (like heart disease and cancer), and promote your overall health. A balanced diet is one that gives your body the nutrients it needs to function correctly. The number of calories in a food is a measurement of the amount of energy stored in that food. Your body uses calories from food for walking, thinking, breathing, and other important functions.

The average person needs to eat about 2,000 calories every day to maintain their weight. However, a person's specific daily calorie intake can vary depending on their age, gender, and physical activity level. Men generally need more calories than women, and people who exercise need more calories than people who don't.

Making decisions about what to eat is a major problem in our everyday lives due to a wide variety of ingredients, culinary styles, ethnicities, cultures, and personal tastes. Choosing the right dish at the right time seems to be a very difficult task. Today, many diseases that were previously thought as hereditary are now shown to be seen connected to biological dis-function related to nutrition. Although being healthy and eating better is something the vast majority of the population want, doing so usually requires great effort and organization.

There is an existing system for the Nutritional food Recommendation System model which tends to generate daily food recommendations for users. The existing system is a model which generate the daily food recommendations for users based on their nutritional necessities which include how much calories they want to intake, etc. and also considers previous food preferences so that the recommended food plan is most suitable for the user according to their preferences and choices.

1.4 CHALLENGES IN THE EXISTING SYSTEM

Even though there is an existing system it has some limitations which need to be addressed so that the user will have a more options and choices to make and get a more suitable food plan according to their convenience.

Some limitations include:

i. The existing system does not classify the food or does not show any classification among the cuisines and the type of food. The user can have cuisine preferences like north Indian, south

Indian etc. or can have food type preference like Vegetarian, Non-Vegetarian, Eggetarian and this classification among food and cuisine is not present in the existing system.

ii. The existing system does not provide the recipe of the recommended foods and it is a drawback since the user may or may not know how to prepare the food item. So, the recipe or instructions to prepare the food item should be provided along with the names of the food items and this feature is not available in the existing system.

iii. The existing system does not take into consideration the allergies of the users. It is a drawback since the user may have allergies and if they are not considered and the food items are suggested then it will be difficult for the user. So, the allergic items are considered and the food items not containing thar particular allergic item is only suggested which is not available in the existing system.

1.5 NUTRITIONAL FOOD RECOMMENDATION SYSTEM

The World Health Organization identifies the overall increasing of non-communicable diseases as a major issue, such as premature heart diseases, diabetes, and cancer. Unhealthy diets have been identified as the important causing factor of such diseases. In this context, personalized nutrition emerges as a new research field for providing food intake advices to individuals according to their physical and personal information.

In this modern world various people suffer from different types of diseases and illnesses. It is generally very difficult to suggest a diet as quickly as possible. Most people have a dire need to Lose Weight, Gain Weight or stay Healthy. Time has also become a possible constraint. The project makes use of a dataset which contains various nutrients in the correct amount. In the wake of the situation, this project tries to recommend diet to people.

The main focus of this work is to provide dietary assistance to different people based on their clinical conditions. In this context, whereas a one-size fits-all approach may fail, personalized nutrition can benefits consumers to adhere to a healthy, pleasurable, and nutritional diet when it is closely associated to individual parameters such as health status, the consumer's needs and preferences, height, weight, age, gender etc. Personalized nutrition can be used for different target groups from healthy people to malnourished people. Regarding the use of nutritional principles this work focuses on building a nutritional recommender system that considers user inputs in the above-mentioned fields and generates a diet plan for each meal of the day.

In our proposal, the recommendation system takes user input in terms of their height, weight, gender, age, a yes/no option for diabetes and blood pressure, a preference box for them to choose South Indian food or North Indian food and an option to choose one among weight loss, weight gain or simply healthy diet required by them. Considering these inputs the system generates multiple options for breakfast, lunch and dinner along with necessary instructions to prepare those particular food items for the ease of the user. The user has the freedom to select from multiple options for a particular meal and prepare it easily with the available instructions instead of spending time in searching for the recipe.

The proposed system generates diet plans that are suitable for weight loss, weight gain and mere healthy food based on the user's preference. Few features included in the proposed system are the classification among cuisines, classification among the type of food items, generating recipes for the food items and considering users clinical conditions such as diabetes and blood pressure.

In this work, the aim is to make recipe recommendations that are both healthy and tasty. A recipe's healthiness can be determined by its amount of healthy vs unhealthy nutrients (such as proteins vs sodium). A list of nutritional values pertaining to the food items is maintained according to which the recipes are recommended to the user based on their inputs. Thus, this project proposes an integration of machine learning algorithms to recommend the right food at the right time and with the right nutrition, calories, fat etc.

1.5.1 Advantages of the proposed system:

i. The proposed system shows a classification among food items that are veg, non veg and egg. This way the users can simply select the kind of food that they prefer instead of searching in the given list.

ii. The proposed system shows a classification among food pertaining to different cuisines namely South Indian and North Indian food. This gives the users a choice to select the kind of cuisine they'd like to try depending on their mood or simply based on the region they belong to.

iii. The major advantage of the proposed system is that it generates a set of instructions to prepare a particular recipe which is an added advantage to the user in order to prepare the required meal with ease in less amount of time possible.

iv. The proposed system considers the allergic items of the user and suggests the items which does not contain the allergic items so it will be easy for the user to consider the food recommendations.

1.6 OBJECTIVES

Wide variety of ingredients, cultures and personal tastes makes decision about what to eat a great problem. Many diseases that were previously thought as hereditary are now seen to be connected to biological dysfunction related to nutrition. Being healthy and eating better is something the vast majority of the population wants and doing so usually requires great effort. However, it is not possible for humans to handle many cases and analyze all the elements which are needed to be considered for providing a healthy and balanced diet. Keeping this in mind, this project develops certain system which implements machine learning algorithms to gather the user data such as age, weight, height, diseases etc. and analyze them to provide a menu of food items to the user for the proper meals at the right time.

The working prototype accomplishes a Personalized Diet Recommendation System with integration of Machine Learning Algorithms to recommend the right food at right time and with the right nutrition, calories, fat etc.

The major objectives for this work are as follows:

i. To provide meal recommendations for each course of the day by considering the user's clinical parameters and the purpose of the diet.
ii. To generate the instructions for the corresponding recipes for the ease of the user.
iii. To store the user's ratings and reviews regarding their progress and interaction with the application in order to improve the results for the upcoming users.
iv. To provide an interactive graphical user interface in order to consider the user details.

1.7 METHODOLOGY

The system uses machine learning to make recommendations of the food recipes to the user and Python as the programming language since Python has been accepted widely as a language for experimenting in the machine learning area. Machine learning uses historical data and information to gain experiences and generate a trained model by training it with the data. This model then makes output recommendations.

Food recommendation system is completely dependent upon user's health factors such as diabetics, blood pressure etc. and physical factors like height, weight and gender. These factors play a major role in recommending proper food recipes to the user. Also, considering user preferences make the recommendations suitable for the user. Considering all the factors together makes the recommendation user preferable. Hence, using Machine Learning, we build a system that uses machine

learning techniques to make recommendations of the food recipes by studying the health factors, physical factors and user preferences. The figure 1.1 describes the steps involved in the process of recommendation system and the steps are:

Taking user details as input:

The details regarding user's physical factors, health factors and preferences are taken as input and used to recommend the food items to the user.

Clustering:

The details given by the user are taken into consideration and based on the purpose of the user to use the recommendation system, the data is made into clusters based on the meal for which the food items are consumed using k-means clustering.

Classification:

The labels of the clusters formed and the nutrients appropriate for the purpose for which the user is using the recommendation system is given as the input to the random forest classifier to train the model for giving the outputs. The classifier forms many decision trees and takes an average of all and gives the output.

Figure 1.1 Methodology

1.7.1 Data collection:

The data regarding various food items and their recipes along with their nutrients which include calories, fats, proteins, calcium, potassium, sugars etc. are collected and considered as a dataset. The food items collected are segregated based on the breakfast, lunch, dinner categories i.e., the meal for which they are consumed. The nutrients are useful while considering the purpose of the user to use the recommendation system. The details entered by the user are given as the input to the system.

1.7.2 K-Means Clustering:

k-means clustering is used to make clusters. K-means algorithm is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. The details given by the user are taken into consideration and based on the purpose of the user to use the recommendation system, the data is made into clusters based on the meal for which the food items are consumed using k-means clustering. The labels of the clusters are taken and stored and are used for the next step of the process which includes training of the machine learning model.

1.7.3 Random Forest Classifier:

Random Forest Classifier is used for the training of the model and get the output which are the recommendations for the user. Random Forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. The labels of the clusters which are stored and the nutrients appropriate for the purpose for which the user is using the recommendation system is given as the input to the random forest classifier to train the model for giving the outputs.

1.7.4 Recommendations as output:

The Random Forest Classifier gets the outputs from the decision trees formed which are suitable for the details given by the user and displays those food items and their recipes as output in the user interface which is the medium through which user can interact with the recommendation system.

1.8 HARDWARE AND SOFTWARE REQUIREMENTS

1.8.1 Software Requirements:

Platform - Windows 10

Programming Language - Python

Development Tool - PyCharm, SQLyog

Windows 10 is the platform where the project is being executed. Python is the language used to write the code and it is used because it is very much compatible with the Machine Learning Algorithms. PyCharm is the tool which is used to develop all the files required for the project which include backend and frontend files. SQLyog is the tool which is used for the database storage and connectivity.

1.8.2 Hardware Requirements:

Name of the Processor - Windows 10 intel® core™ i5

RAM Capacity - 8GB

Hard Disk Capacity - 1TB

1.9 ORGANIZATION OF THE PROJECT

The project is organized into the following chapters. Chapter 1 is focused on the outline of the importance of nutritional food in the daily routine of a person. The background and significance of the project are addressed. The existing systems are considered, that are related to the theme of the project along with their advantages and disadvantages. A literature review of the previous studies carried out. Next the challenges in existing system are listed out. The proposed model is explained in detail. Next the objectives of the proposed system are defined. The methodology charted in the project is delineated. Next the Software and Hardware requirements are mentioned followed by organization of project. Chapter 2 gives a detailed report about the project, starting with the architecture, dataflow of the proposed system followed by UML Diagrams. Later the algorithms used are mentioned. Chapter 3 describes the implementation of modules, tools used along with requirement analysis and code implementation. Chapter 4 deals with the results and discussions. The screenshots, graphs, and tables of results are listed here. Chapter 5 notes the future extensions of the project and concludes it.

2.PROPOSED SYSTEM

2.1 ARCHITECTURE/DATA FLOW DIAGRAM

The architecture defines the various layers involved in the machine learning cycle and involves the major steps being carried out throughout. It shows the general structure and the associations between each element that is involved.

The figure 2.1 shows the architectural diagram for the working prototype of the Diet Recommendation System. The code is written in Python which is used to implement various Clustering and Classification algorithms in Machine Learning for predicting a proper diet to the user.

For training of the system, the initial process involves the segregation of food items depending upon the meal for which they are consumed i.e., Breakfast, Lunch and Dinner. The clustering of various nutrients depending upon which are essential for the weight-loss, weight-gain and healthy is performed.

After the clustering is performed, using Random Forest classifier, the nearest food items are predicted which best suits for the appropriate diet. As part of user interface, the inputs needed from the user are Age, Height, Weight, Cuisine, type of food required, if the user has diabetes or blood pressure and for what purpose the diet is required, allergic food items if the user has any.

Depending upon it, from the appropriate clustering, specific food items are classified and recommended to the user. The K – Means Clustering Algorithm takes into consideration various attributes and forms a cluster. The new attribute values being taken into account fall into one of the clusters and then the classification algorithms are used to predict the food item list based on the type of inputs given by the user.

The input is taken from the GUI. The main graphical user interface will consist of various input fields taking into consideration the user preferences for food like vegetarian/ non– vegetarian food items at Lunch/ Dinner/ Breakfast along with other details like an individual's height, weight and age.

The Machine Learning model is connected to the GUI using flask. Additionally, the user's feedback regarding the recommendation system is captured and stored in database which is linked to the GUI.

The Diet Recommendation System will recommend the list of desired foods based on the fact that whether the user is expecting a weight gain/ weight loss/ healthy diet and gives a list of suitable items for breakfast, lunch and dinner along with instructions to prepare each food item.

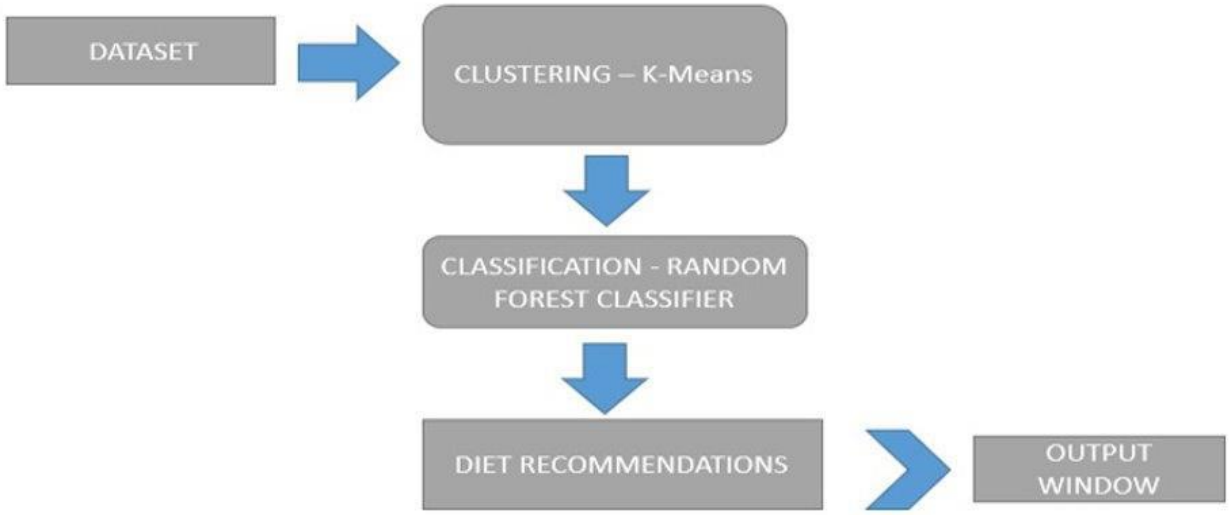


Figure 2.1 Architecture

The figure 2.2 shows the common workflow carried out in the project for better understanding of the algorithms used and the process. The dataset is initially processed by applying K-Means clustering algorithm. It is then sent to the random forest classifier to generate the majority food items depending on the user's conditions. Once that is done the recommendations are generated which are displayed to the user through a GUI.

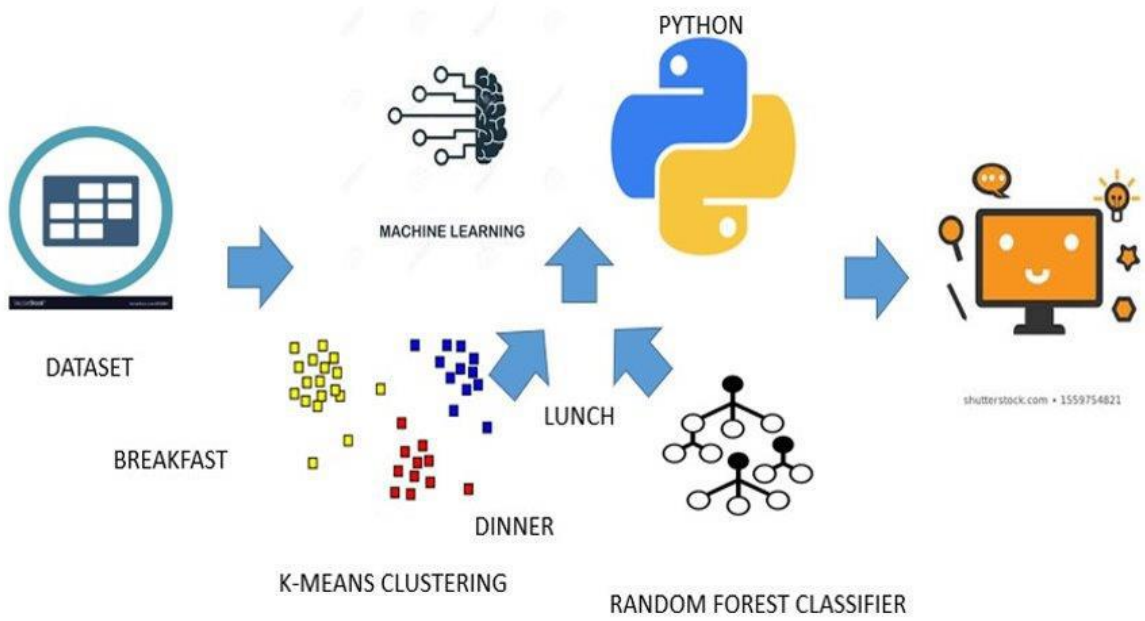


Figure 2.2 Workflow

2.2 UML DIAGRAMS

2.2.1 Overall Use case Diagram:

When a user logs into the application and enters his/her clinical parameters such as diabetes, BP issues and the type of diet required, the model tends to cluster the data using K-Means clustering algorithm based on the nutrients required for the user's diet. Once the clustering is done in order to choose the best possible option Random Forest is applied to select the recipe that has majority votes among the available options. The recommendations are thus generated a displayed to the user through flask framework. The user is also prompted to give feedback which is stored in the database and is displayed to other users when they login to the application.

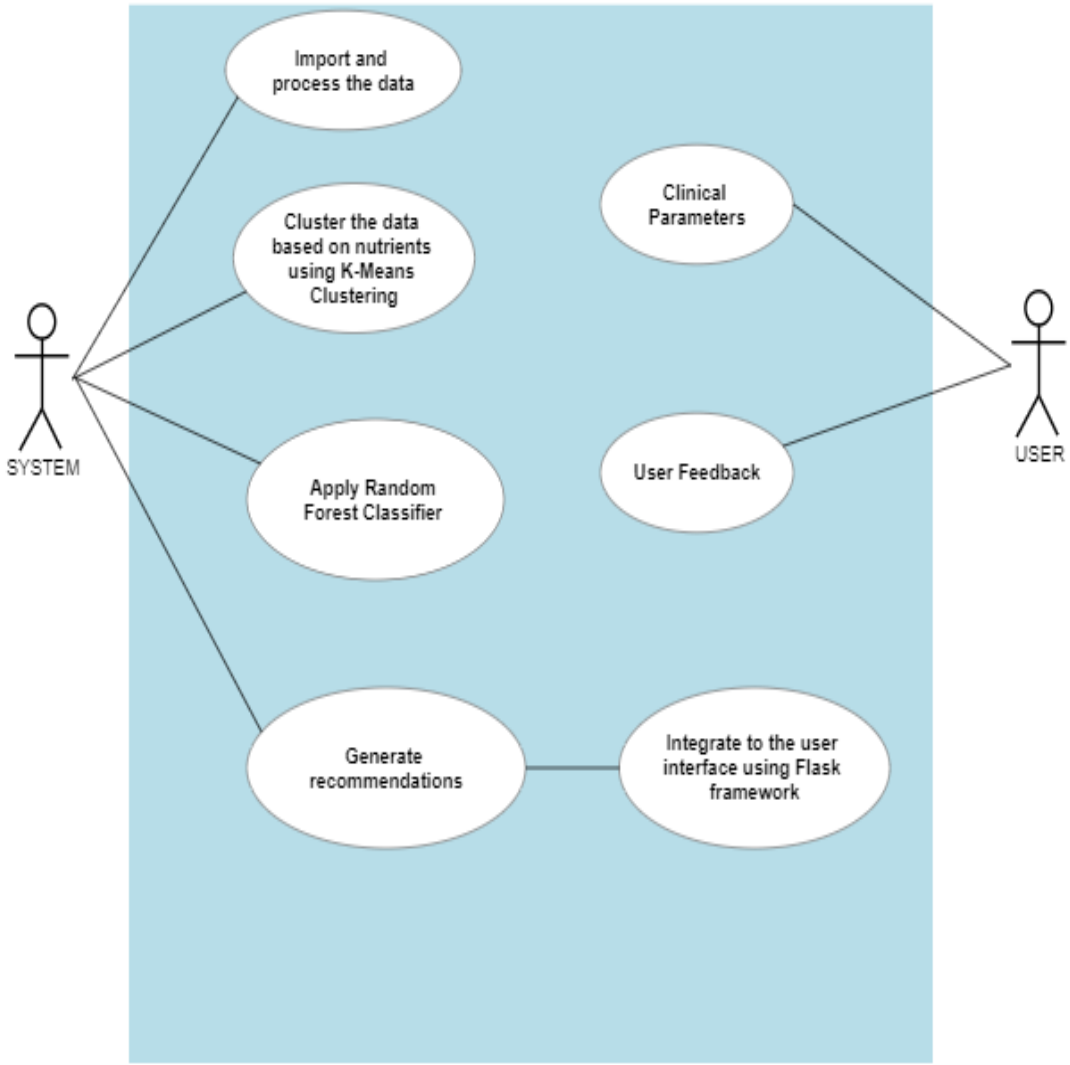


Figure 2.3 Use case Diagram of proposed system

2.2.2 Activity Diagram:

An activity diagram is a behavioral diagram, it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. This Activity diagram displays the behavior of the Nutritional Food Recommendation system from the admin to the user interaction.

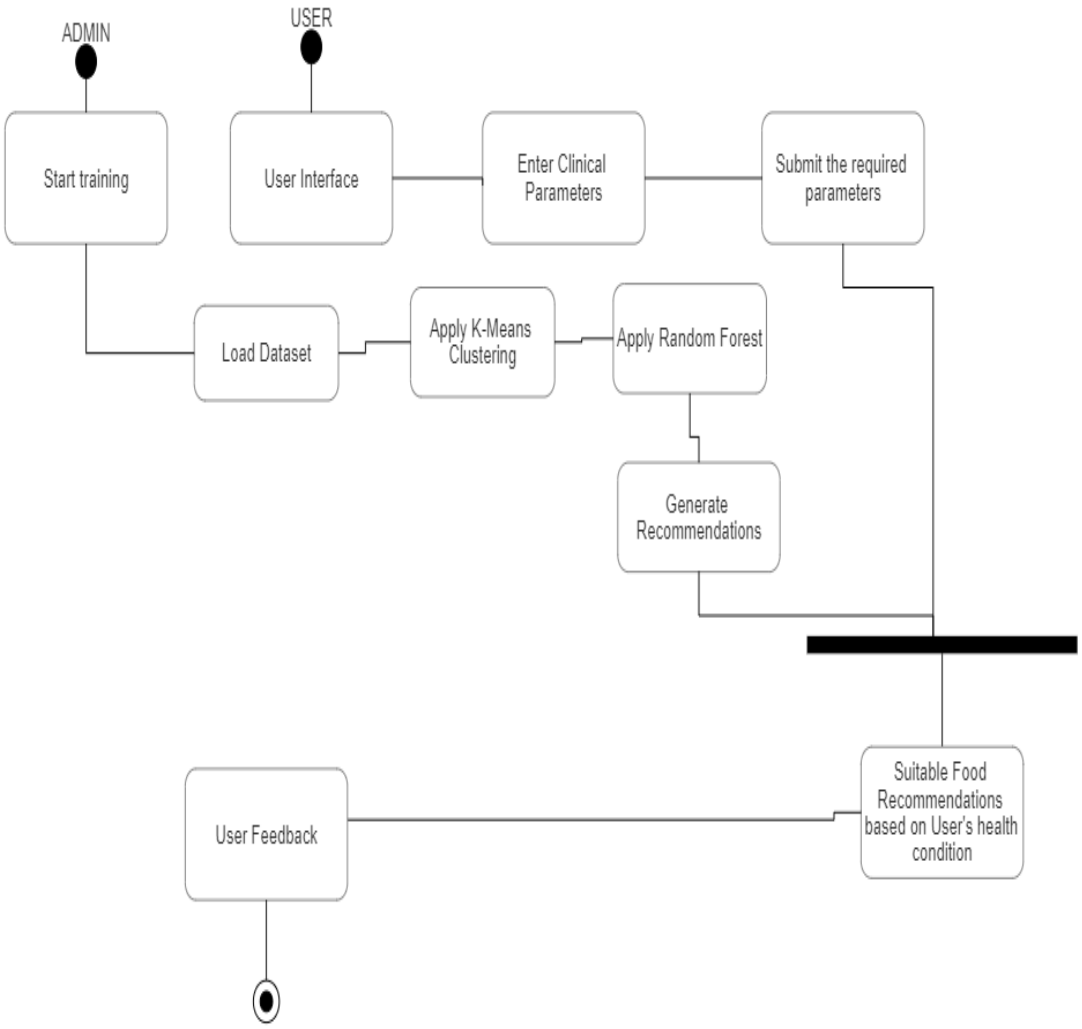


Figure 2.4 Activity Diagram of proposed system

2.3 K – MEANS CLUSTERING

K-means algorithm is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group.

It tries to make the intra- cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (Arithmetic mean of all the data points that belong to that cluster) is at the minimum.

The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

Advantages of k-means:

i. Relatively simple to implement.

ii. Scales to large data sets.

iii. Guarantees convergence.

Disadvantages of k-means:

i. Choosing k manually.

ii. Being dependent on initial variables.

iii. Clustering outliers.

```
## K-Means Based  Breakfast Food
Datacalorie = breakfastfoodseparatedIDdata[0:,1:len(breakfastfoodseparatedIDdata)]
#print("DC=", Datacalorie)
#print("DClwen=", len(Datacalorie))

X = np.array(Datacalorie)
#print("x=", X)
kmeans = KMeans(n_clusters=3, random_state=0).fit(X)
y_kmeans = kmeans.predict(X)
#print("y_kmeans=",y_kmeans)
'''for i in range(len(y_kmeans)):
    print("xxx=", X[y_kmeans[i]])'''
```

Fig 2.5 Implementation of K-means clustering

2.4 RANDOM FOREST CLASSIFICATION

Random Forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting.

It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

Advantages of Random Forest:

i. It takes less training time as compared to other algorithms.

ii. It predicts output with high accuracy, even for the large dataset it runs efficiently.

iii. It can also maintain accuracy when a large proportion of data is missing.

Disadvantages of Random Forest:

i. It tends to overfit for some datasets.

ii. More difficult to interpret than decision trees.

```
# randomforest
for jj in range(len(weightlosscat)):
    valloc = list(weightlosscat[jj])
    #valloc.append(agecl)
    #valloc.append(clbmi)

    X_test[jj] = np.array(valloc) * ti

X_train = weightlossfin  # Features
y_train = yt  # Labels

#print("x_train=",X_train)

#print("y_train=", y_train)
#print("y_trainlen=", len(y_train))

#print("X_test=", X_test)

# Create a Gaussian Classifier
clf = RandomForestClassifier(n_estimators=100)

# Train the model using the training sets y_pred=clf.predict(X_test)
clf.fit(X_train, y_train)

# print (X_test[1])
#X_test2 = X_test
y_pred = clf.predict(X_test)
#print("y_pred=",y_pred)
#print("y_predlen=", len(y_pred))
#print("ids=",breakfastfoodseparatedID)
#print("idslen=", len(breakfastfoodseparatedID))
```

Fig.2.6 Implementation of Random Forest classifier

3.IMPLEMENTATION AND TESTING

3.1 MODULE DESCRIPTION

Breakfast Module includes the retrieval of breakfast data and clustering of various nutrients depending upon which are essential for weight_loss, weight_gain, healthy is performed.

After clustering is performed, using random forest classifier, the nearest food items are predicted which best suited for appropriate diet.

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.ensemble import RandomForestClassifier
import sys
from DBConnection import DBConnection
def Diet_Control_Breakfast(age,gender,height,weight,cuisine,catgry,sugar,bp,wl,cols,sno,allergitic_items):
    try:


        food_dataset = pd.read_csv('food3_2.csv', encoding="ISO-8859-1")
        rec_food = food_dataset
        Breakfastdata = food_dataset['Breakfast']
        BreakfastdataNumpy = Breakfastdata.to_numpy()
        Food_itemsdata = food_dataset['Food_items']

        breakfastfoodseparated = []
        breakfastfoodseparatedID = []

        for i in range(len(Breakfastdata)):
            if BreakfastdataNumpy[i] == 1:
                breakfastfoodseparated.append(Food_itemsdata[i])
                breakfastfoodseparatedID.append(i)

        # retrieving Breafast data rows by loc method
        breakfastfoodseparatedIDdata = food_dataset.iloc[breakfastfoodseparatedID]
        # print("bk=",breakfastfoodseparatedIDdata)
        # print("breakfastfoodseparatedID=",(breakfastfoodseparatedID))
        fddata = breakfastfoodseparatedIDdata
        breakfastfoodseparatedIDdata = breakfastfoodseparatedIDdata.T
        val = list(np.arange(8, 19))
        nutritions = [0] + val
        # print(nutritions)
        breakfastfoodseparatedIDdata = breakfastfoodseparatedIDdata.iloc[nutritions]
        breakfastfoodseparatedIDdata = breakfastfoodseparatedIDdata.T
```

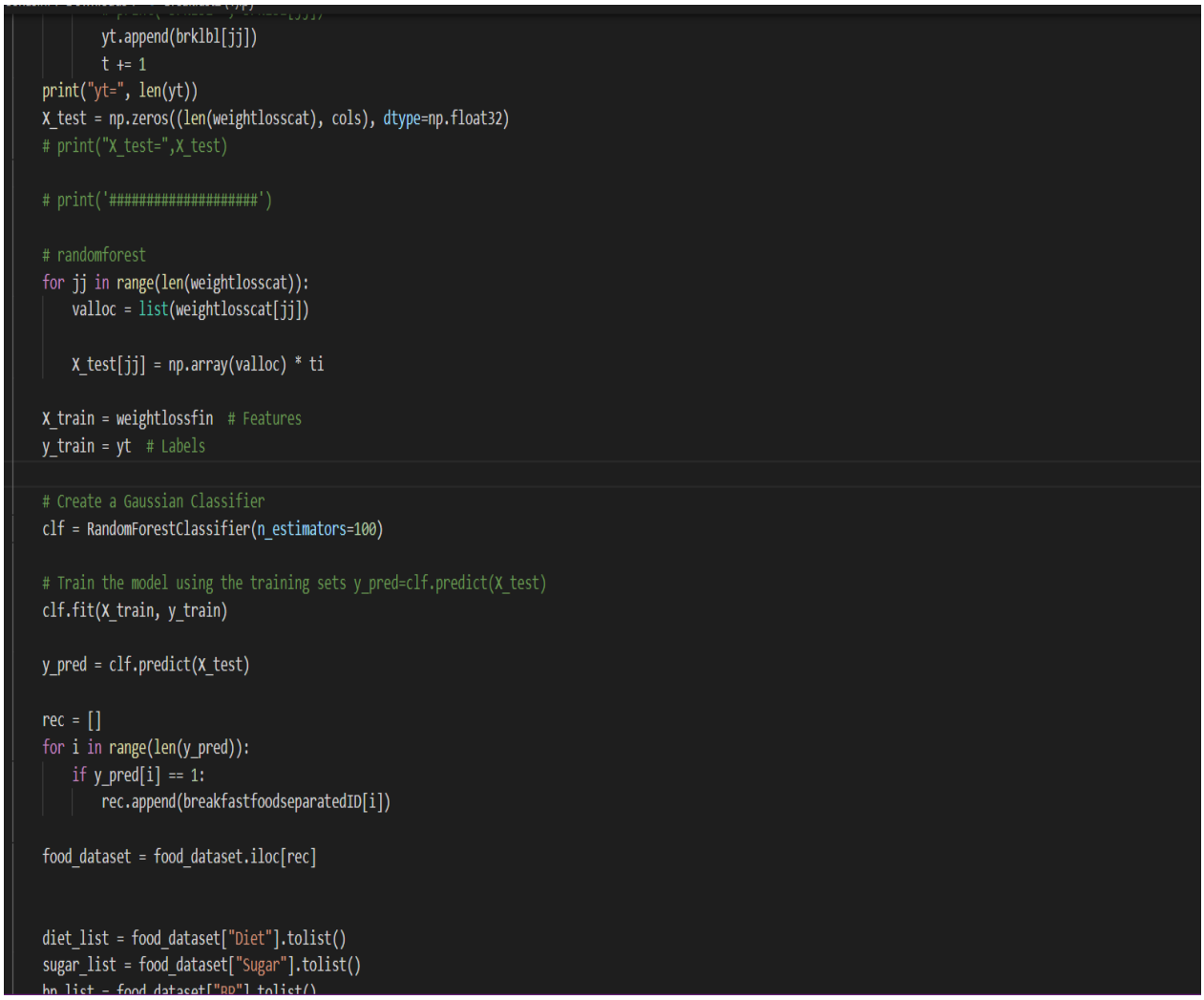
Figure 3.1 Breakfast module image 1

```
    # calculating BMI
    age = int(age)
    gen = int(gender)
    diet = catgry
    cuisine = cuisine
    sugar = sugar  # 1-yes, 0-no
    bp = bp  # 1-yes, 0-no
    weight = int(weight)
    height = int(height)

    bmi = weight / ((height / 100) ** 2)
    agewiseinp = 0
    agecl=0

    # Age Classs [1,2,3]
    for lp in range(0, 80, 20):
        # print("lp=",lp)
        test_list = np.arange(lp, lp + 20)
        # print("test_list=", test_list)
        for i in test_list:
            if (i == age):
                tr = round(lp / 20)
                agecl = round(lp / 20)
                # print("tr=",tr)
                # print("agecl=",agecl)

    print("Your body mass index is: ", bmi)
    # BMI class
    if (bmi < 16):
        # print("Acoording to your BMI, you are Severely Underweight")
        clbmi = 4
    elif (bmi >= 16 and bmi < 18.5):
        # print("Acoording to your BMI, you are Underweight")
        clbmi = 3
    elif (bmi >= 18.5 and bmi < 25):
        # print("Acoording to your BMI, you are Healthy")
```

Figure 3.2 Breakfast module image 2

```
        clbmi = 2
    elif (bmi >= 25 and bmi < 30):
        # print("Acoording to your BMI, you are Overweight")
        clbmi = 1
    elif (bmi >= 30):
        # print("Acoording to your BMI, you are Severely Overweight")
        clbmi = 0

    # converting into numpy array
    breakfastfoodseparatedIDdata = breakfastfoodseparatedIDdata.to_numpy()
    # print("bstnumpi=",breakfastfoodseparatedIDdata)

    ti = (clbmi + agecl + gen) / 2
    # print("ti=", ti)

    ## K-Means Based  Breakfast Food
    Datacalorie = breakfastfoodseparatedIDdata[0:, 1:len(breakfastfoodseparatedIDdata)]
    # print("DC=", Datacalorie)
    # print("DClwen=", len(Datacalorie))

    X = np.array(Datacalorie)
    # print("x=", X)
    kmeans = KMeans(n_clusters=3, random_state=0).fit(X)
    y_kmeans = kmeans.predict(X)
    # retrieving the labels for breakfast food
    brklbl = kmeans.labels_
    # print("brklbl=", brklbl)

    inp = []
    ## Reading of the Dataet
    datafin_nutr = pd.read_csv('nutrition_distriution2.csv')
    ## train set
    datafin_nutr = datafin_nutr.T
    # print("datafin_nutr=",datafin_nutr)
```

Figure 3.3 Breakfast module image 3



```
            yt.append(brklbl[jj])
            t += 1
    print("yt=", len(yt))
    X_test = np.zeros((len(weightlosscat), cols), dtype=np.float32)
    # print("X_test=",X_test)

    # print('###################')

    # randomforest
    for jj in range(len(weightlosscat)):
        valloc = list(weightlosscat[jj])

        X_test[jj] = np.array(valloc) * ti

    X_train = weightlossfin  # Features
    y_train = yt  # Labels

    # Create a Gaussian Classifier
    clf = RandomForestClassifier(n_estimators=100)

    # Train the model using the training sets y_pred=clf.predict(X_test)
    clf.fit(X_train, y_train)

    y_pred = clf.predict(X_test)

    rec = []
    for i in range(len(y_pred)):
        if y_pred[i] == 1:
            rec.append(breakfastfoodseparatedID[i])

    food_dataset = food_dataset.iloc[rec]


    diet_list = food_dataset["Diet"].tolist()
    sugar_list = food_dataset["Sugar"].tolist()
    bp_list = food_dataset["BP"].tolist()
```

Figure 3.4 Breakfast module image 4

Lunch and Dinner Modules also include the retrieval of lunch data and dinner data and clustering of various nutrients depending upon which are essential for weight_loss, weight_gain, healthy is performed.

After clustering is performed, using random forest classifier, the nearest food items are predicted which best suited for appropriate diet.

As a part of user interface, the inputs needed from the user are taken and depending on the user preferences, from the appropriate clustering, specific food items are classified and recommended to the user.

Database Connectivity module establishes the connection between the python application and mysql database

```
import mysql.connector
class DBConnection:
    @staticmethod
    def getConnection():
        database = mysql.connector.connect(host="localhost", user="root", passwd="root", db='food_recommendation')
        return database
if __name__=="__main__":
    print(DBConnection.getConnection())
```

Figure 3.5 Database Connectivity

Flask Module includes the submodules like templates, static and others. The static submodule includes the images, style files. The template submodule includes the files called index, user_signin, user_home, user_signup, food_recommendation, feedback.

```
from flask import Flask, render_template, request,flash
import pandas as pd
from flask import Response
import csv
from flask import session
from DBConnection import DBConnection
from Breakfast2 import Diet_Control_Breakfast
from Lunch2 import Diet_Control_Lunch
from Dinner2 import Diet_Control_Dinner
import sys
app = Flask(__name__)
app.secret_key = "abc"



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


@app.route('/signin')
def signin():
    return render_template('user_signin.html')


@app.route('/user_home')
def user_home():
    return render_template('user_home.html')


@app.route('/signup')
def signup():
    return render_template('user_signup.html')
```

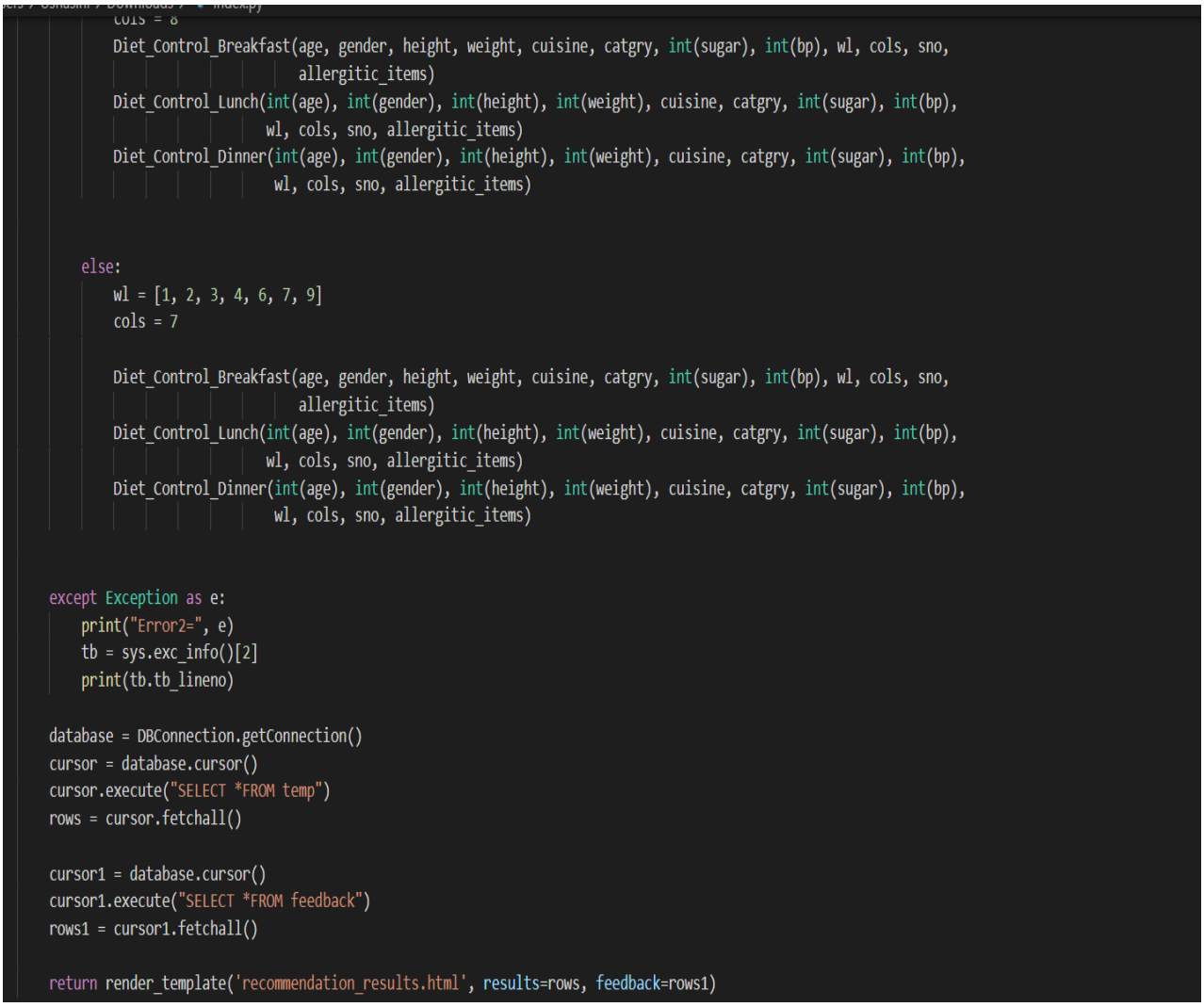
Figure 3.6 Flask file image 1

```python
@app.route('/nutritional_FRS')
def nutritional_FRS():
    return render_template('food_recommendation.html')


@app.route('/feedback')
def feedback():
    return render_template('feedback.html')



@app.route('/save_feedback',methods =["GET", "POST"])
def save_feedback():
    try:
        feedback = request.form.get('fb')
        uid=session['uid']

        database = DBConnection.getConnection()
        cursor = database.cursor()

        sql = "insert into feedback values(%s,%s)"
        values = (uid, feedback)
        cursor.execute(sql, values)
        database.commit()

    except Exception as e:
        print(e)
    return render_template('feedback.html',messages="done")



@app.route('/registering',methods =["GET", "POST"])
def registering():
    try:
```

Figure 3.7 Flask file image 2

```python
    try:
        name = request.form.get('name')
        uid = request.form.get('uid')
        pwd = request.form.get('pwd')
        email = request.form.get('email')
        mno = request.form.get('mno')

        database = DBConnection.getConnection()
        cursor = database.cursor()
        sql = "select count(*) from register where userid='" + uid + "'"
        cursor.execute(sql)
        res = cursor.fetchone()[0]
        if res > 0:

            return render_template("user_signup.html", messages="User Id already exists..!")

        else:
            sql = "insert into register values(%s,%s,%s,%s,%s)"
            values = (name, uid, pwd, email, mno)
            cursor.execute(sql, values)
            database.commit()

        return render_template("user_signin.html", messages="Registered Successfully..! Login Here.")
    except Exception as e:
        print(e)
    return render_template('user_signin.html')


@app.route('/food_recommends',methods =["GET", "POST"])
def food_recommends():
    try:
        age = request.form.get('age')
        gender = request.form.get('gender')
        height = request.form.get('height')
        weight = request.form.get('weight')
        cuisine = request.form.get('cuisine')
```

Figure 3.8 Flask file image 3



```
            cols = 8
            Diet_Control_Breakfast(age, gender, height, weight, cuisine, catgry, int(sugar), int(bp), wl, cols, sno,
                                   allergitic_items)
            Diet_Control_Lunch(int(age), int(gender), int(height), int(weight), cuisine, catgry, int(sugar), int(bp),
                               wl, cols, sno, allergitic_items)
            Diet_Control_Dinner(int(age), int(gender), int(height), int(weight), cuisine, catgry, int(sugar), int(bp),
                                wl, cols, sno, allergitic_items)


        else:
            wl = [1, 2, 3, 4, 6, 7, 9]
            cols = 7

            Diet_Control_Breakfast(age, gender, height, weight, cuisine, catgry, int(sugar), int(bp), wl, cols, sno,
                                   allergitic_items)
            Diet_Control_Lunch(int(age), int(gender), int(height), int(weight), cuisine, catgry, int(sugar), int(bp),
                               wl, cols, sno, allergitic_items)
            Diet_Control_Dinner(int(age), int(gender), int(height), int(weight), cuisine, catgry, int(sugar), int(bp),
                                wl, cols, sno, allergitic_items)


    except Exception as e:
        print("Error2=", e)
        tb = sys.exc_info()[2]
        print(tb.tb_lineno)

    database = DBConnection.getConnection()
    cursor = database.cursor()
    cursor.execute("SELECT *FROM temp")
    rows = cursor.fetchall()

    cursor1 = database.cursor()
    cursor1.execute("SELECT *FROM feedback")
    rows1 = cursor1.fetchall()

    return render_template('recommendation_results.html', results=rows, feedback=rows1)
```

Figure 3.9 Flask file image 4

3.2 DESCRIPTION OF TECHNOLOGIES/TOOLS USED

3.2.1 PyCharm:

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

PyCharm offers some of the best features to its users and developers in the following aspects −

i. Code completion and inspection

ii. Advanced debugging

iii. Support for web programming and frameworks such as Django and Flask

Features of PyCharm

Besides, a developer will find PyCharm comfortable to work with because of the features mentioned below –

i. Code Completion

PyCharm enables smoother code completion whether it is for built in or for an external package.

ii. SQLAlchemy as Debugger

You can set a breakpoint, pause in the debugger and can see the SQL representation of the user expression for SQL Language code.

iii. Git Visualization in Editor

When coding in Python, queries are normal for a developer. You can check the last commit easily in PyCharm as it has the blue sections that can define the difference between the last commit and the current one.

iv. Code Coverage in Editor

You can run .py files outside PyCharm Editor as well marking it as code coverage details elsewhere in the project tree, in the summary section etc.

v. Package Management

All the installed packages are displayed with proper visual representation. This includes list of installed packages and the ability to search and add new packages.

vi. Local History

Local History is always keeping track of the changes in a way that complements like Git. Local history in PyCharm gives complete details of what is needed to rollback and what is to be added.

vii. Refactoring

Refactoring is the process of renaming one or more files at a time and PyCharm includes various shortcuts for a smooth refactoring process.

3.2.2 SQLyog:

SQLyog is a content management tool for Windows that lets you visually compare databases and create schemas. SQLyog gives you a visual overview of all your database information, including schemas and related formulas. You can lay out your database content in easy-to-read table schemas. Plus, with its rapid checksum algorithm, SQLyog also lets you periodically replicate data. All of this is done via its built-in Database Synchronization Wizard, which smoothly guides you through the process.

Every aspect of SQLyog is designed to ease your workflow. It features a wide variety of built-in 'wizards,' including ones to facilitate automation of installation and registration, HTTP/HTTPS tunneling connectivity, and the compression of scheduled backups. You can also use SQLyog to manage database content via a master table lookup, a favorites manager, and a multicolored tabbed interface for easier readability. Many of the software's capabilities are designed to improve the overall user experience by making everything straightforward and intuitive than other MySQL tools.

SQLyog Ultimate enables database developers, administrators, and architects to visually compare, optimize, and document schemas. SQLyog Ultimate includes a power tool to automate and schedule the synchronization of data between two MySQL hosts.

3.2.3 Machine learning:

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. the learning system of a machine learning algorithm into three main parts.

i. A Decision Process: In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labelled or unlabeled, your algorithm will produce an estimate about a pattern in the data.
ii. An Error Function: An error function serves to evaluate the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model.
iii. A Model Optimization Process: If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this evaluate and optimize process, updating weights autonomously until a threshold of accuracy has been met.

Machine learning classifiers fall into three primary categories.

1. Supervised machine learning:

   Supervised learning, also known as supervised machine learning, is defined by its use of labeled datasets to train algorithms that to classify data or predict outcomes accurately. As input data is fed into the model; it adjusts its weights until the model has been fitted appropriately.

   This occurs as part of the cross-validation process to ensure that the model avoids overfitting or underfitting. Supervised learning helps organizations solve for a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox. Some methods used in supervised learning include neural networks, naïve bayes, linear regression, logistic regression, random forest, support vector machine (SVM), and more.

2. Unsupervised machine learning:

   Unsupervised learning, also known as unsupervised machine learning, uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, image and pattern recognition. It's also used to reduce the number of features in a model through the process of dimensionality reduction; principal component analysis (PCA) and singular value decomposition (SVD) are two common approaches for this. Other algorithms used in unsupervised learning include neural networks, k-means clustering, probabilistic clustering methods, and more.

3. Semi-supervised learning:

   Semi-supervised learning offers a happy medium between supervised and unsupervised learning. During training, it uses a smaller labeled data set to guide classification and feature extraction from a larger, unlabeled data set. Semi-supervised learning can solve the problem of having not enough labeled data (or not being able to afford to label enough data) to train a supervised learning algorithm.

3.2.4 Flask:

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. Flask is a web framework, it's a Python module that lets you develop web applications easily. It's having a small and easy-to-extend core: it's a microframework that doesn't include an ORM (Object Relational Manager) or such features. It does have many cool features like url routing, template engine. It is a WSGI web app framework.

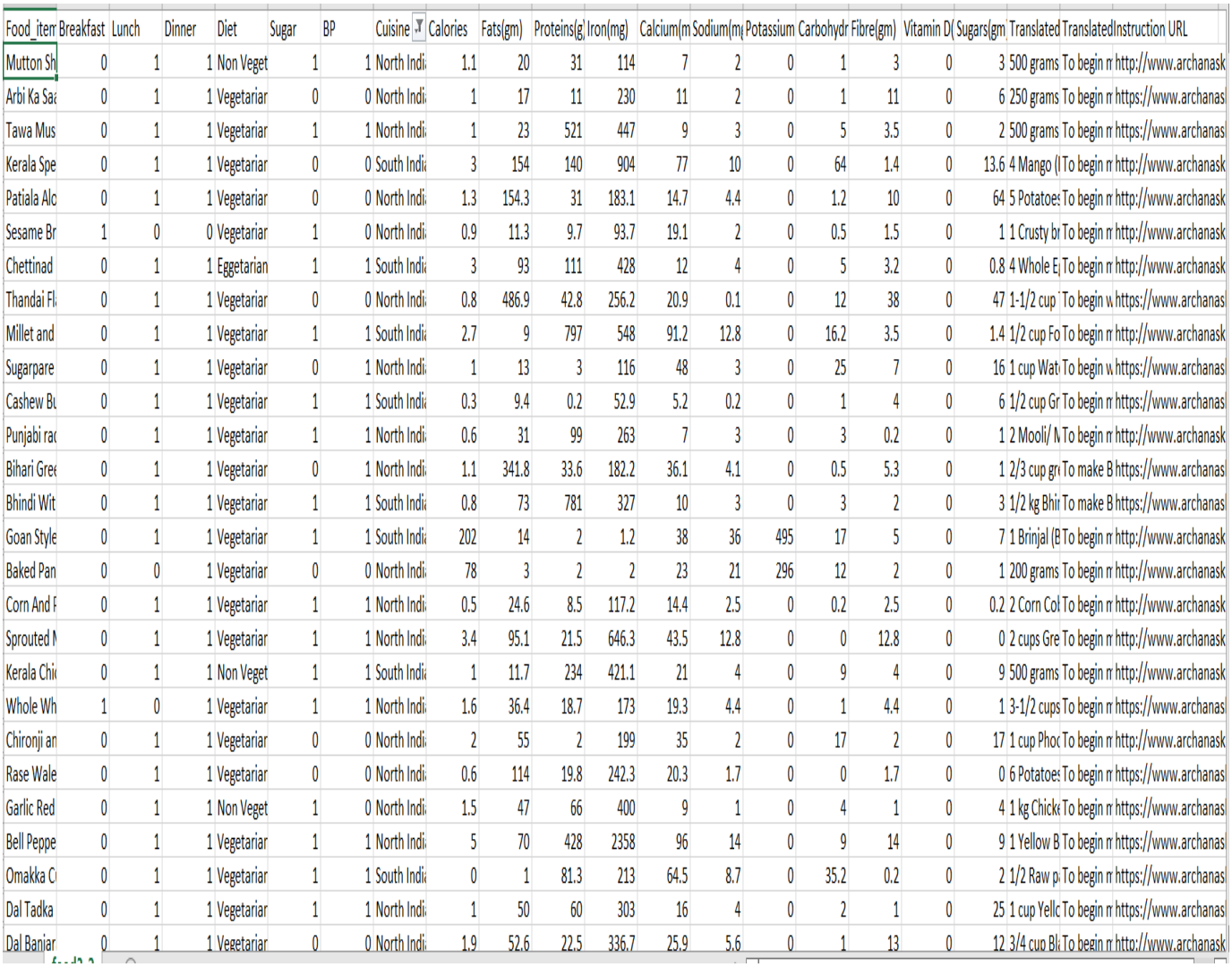
The Flask application is started by calling the `run()` function. The method should be restarted manually for any change in the code. To overcome this, the debug support is enabled so as to track any error. The Variables in the flask is used to build a URL dynamically by adding the variable parts to the rule parameter. Flask support various HTTP protocols for data retrieval from the specified URL, these can be defined as: -

Table 1 HTTP protocols

Method	Description
GET	This is used to send the data in a without encryption of the form to the server.
HEAD	provides response body to the form
POST	Sends the form data to server. Data received by POST method is not cached by server.
PUT	Replaces current representation of target resource with URL.
DELETE	Deletes the target resource of a given URL

3.3 DATASET DESCRIPTION

Dataset includes 2500 data samples. It contains 20 attributes. The attributes are Food item, Breakfast, Lunch, Dinner, Diet, Sugar, BP, Cuisine, Calories, Fats, Proteins, Iron, Calcium, Sodium, Potassium, Carbohydrates, Fibre, Vitamin D, Sugars, Instructions. Calories, Fats, Proteins, Iron, Calcium, Sodium, Potassium, Carbohydrates, Fibre, Vitamin D, Sugars and BP, sugar attributes are considered in generating the recommendations to the users based on their preferences. Diet consists of Vegetarian, Eggetarian and Non vegetarian. BP and Sugar are labelled as 1-Yes and 0-No. Cuisine is categorized into North Indian and South Indian.



Food_item	Breakfast	Lunch	Dinner	Diet	Sugar	BP	Cuisine	Calories	Fats(gm)	Proteins(g	Iron(mg)	Calcium(m	Sodium(m	Potassium	Carbohydr	Fibre(gm)	Vitamin D(	Sugars(gm	Translated	TranslatedInstruction	URL
Mutton Sh	0	1	1	Non Veget	1	1	North Indi	1.1	20	31	114	7	2	0	1	3	0	3	500 grams	To begin n	http://www.archanask
Arbi Ka Sa	0	1	1	Vegetariar	0	0	North Indi	1	17	11	230	11	2	0	1	11	0	6	250 grams	To begin n	https://www.archanas
Tawa Mus	0	1	1	Vegetariar	1	1	North Indi	1	23	521	447	9	3	0	5	3.5	0	2	500 grams	To begin n	https://www.archanas
Kerala Spe	0	1	1	Vegetariar	0	0	South Indi	3	154	140	904	77	10	0	64	1.4	0	13.6	4 Mango (I	To begin n	http://www.archanask
Patiala Alc	0	1	1	Vegetariar	0	0	North Indi	1.3	154.3	31	183.1	14.7	4.4	0	1.2	10	0	64	5 Potatoes	To begin n	http://www.archanask
Sesame Br	1	0	0	Vegetariar	1	0	North Indi	0.9	11.3	9.7	93.7	19.1	2	0	0.5	1.5	0	1	1 Crusty br	To begin n	http://www.archanask
Chettinad	0	1	1	Eggetarian	1	1	South Indi	3	93	111	428	12	4	0	5	3.2	0	0.8	4 Whole E	To begin n	http://www.archanask
Thandai Fl	0	1	1	Vegetariar	0	0	North Indi	0.8	486.9	42.8	256.2	20.9	0.1	0	12	38	0	47	1-1/2 cup	To begin w	https://www.archanas
Millet and	0	1	1	Vegetariar	1	1	South Indi	2.7	9	797	548	91.2	12.8	0	16.2	3.5	0	1.4	1/2 cup Fo	To begin n	http://www.archanask
Sugarpare	0	1	1	Vegetariar	0	1	North Indi	1	13	3	116	48	3	0	25	7	0	16	1 cup Wat	To begin w	https://www.archanas
Cashew Bu	0	1	1	Vegetariar	1	1	South Indi	0.3	9.4	0.2	52.9	5.2	0.2	0	1	4	0	6	1/2 cup Gr	To begin n	https://www.archanas
Punjabi ra	0	1	1	Vegetariar	1	1	North Indi	0.6	31	99	263	7	3	0	3	0.2	0	1	2 Mooli/ M	To begin n	http://www.archanask
Bihari Gre	0	1	1	Vegetariar	0	1	North Indi	1.1	341.8	33.6	182.2	36.1	4.1	0	0.5	5.3	0	1	2/3 cup gr	To make B	https://www.archanas
Bhindi Wit	0	1	1	Vegetariar	1	1	South Indi	0.8	73	781	327	10	3	0	3	2	0	3	1/2 kg Bhi	To make B	https://www.archanas
Goan Style	0	1	1	Vegetariar	1	1	South Indi	202	14	2	1.2	38	36	495	17	5	0	7	1 Brinjal (B	To begin n	http://www.archanask
Baked Pan	0	0	1	Vegetariar	0	0	North Indi	78	3	2	2	23	21	296	12	2	0	1	200 grams	To begin n	http://www.archanask
Corn And F	0	1	1	Vegetariar	1	1	North Indi	0.5	24.6	8.5	117.2	14.4	2.5	0	0.2	2.5	0	0.2	2 Corn Col	To begin n	http://www.archanask
Sprouted M	0	1	1	Vegetariar	1	1	North Indi	3.4	95.1	21.5	646.3	43.5	12.8	0	0	12.8	0	0	2 cups Gre	To begin n	http://www.archanask
Kerala Chi	0	1	1	Non Veget	1	1	South Indi	1	11.7	234	421.1	21	4	0	9	4	0	9	500 grams	To begin n	http://www.archanask
Whole Wh	1	0	1	Vegetariar	1	1	North Indi	1.6	36.4	18.7	173	19.3	4.4	0	1	4.4	0	1	3-1/2 cups	To begin n	https://www.archanas
Chironji ar	0	1	1	Vegetariar	0	0	North Indi	2	55	2	199	35	2	0	17	2	0	17	1 cup Phoo	To begin n	http://www.archanask
Rase Wale	0	1	1	Vegetariar	0	0	North Indi	0.6	114	19.8	242.3	20.3	1.7	0	0	1.7	0	0	6 Potatoes	To begin n	https://www.archanas
Garlic Red	0	1	1	Non Veget	1	0	North Indi	1.5	47	66	400	9	1	0	4	1	0	4	1 kg Chick	To begin n	https://www.archanas
Bell Peppe	0	1	1	Vegetariar	1	1	North Indi	5	70	428	2358	96	14	0	9	14	0	9	1 Yellow B	To begin n	https://www.archanas
Omakka C	0	1	1	Vegetariar	1	1	South Indi	0	1	81.3	213	64.5	8.7	0	35.2	0.2	0	2	1/2 Raw p	To begin n	https://www.archanas
Dal Tadka	0	1	1	Vegetariar	1	1	North Indi	1	50	60	303	16	4	0	2	1	0	25	1 cup Yellc	To begin n	https://www.archanas
Dal Baniar	0	1	1	Vegetariar	0	0	North Indi	1.9	52.6	22.5	336.7	25.9	5.6	0	1	13	0	12	3/4 cup Bl	To begin n	http://www.archanask

Figure 3.10 Dataset

3.4 TEST CASE DESIGN

Test case design refers to how to set-up the test cases. It is important that tests are designed well, or it could fail to identify bugs and defects in the software during testing. There are many different test case design techniques used to test the functionality and various features of the software.

The test case is defined as a group of conditions under which a tester determines whether a software application is working as per the customer's requirements or not. Test case designing includes preconditions, case name, input conditions, and expected result. A test case is a first level action and derived from test scenarios. Some of the test case designs include:

Title 1: User Registration to the website

Description: User should be able to successfully submit their details which will be stored in database and used to login with same credentials to their account on the website.

Preconditions: User must register and give their correct details

Assumptions: They are using a supported device or browser to submit the details

Test Steps:

1. Open website
2. Enter the username, password, email id and contact number in the appropriate fields following the format specified.
3. Click “submit”

Expected Result: The user should submit the details successfully which will be stored in database.

Title 2: User Login to the website

Description: User should be able to successfully log in to their account on the website/app

Preconditions: User must already be registered and use their correct login details

Assumptions: They are using a supported device or browser to log in

Test Steps:

1. Open website
2. Enter the username, password in the appropriate fields
3. Click “login”

Expected Result: The user should log in successfully.

4.RESULTS ANALYSIS

4.1 DESCRIPTION ON RESULTS

The aim of this project is to develop a web application which can give proper nutritious diet so that the user can follow a proper diet to maintain their health or to improve it.

As we know many factors influence an individual's health, such as physical exercise, sleep, nutrition, heredity and pollution. Being nutrition one of the biggest modifiable factors in our lives, small changes can have a big impact. With the exponential increase in the number of available food options, it is not possible to take them all into account anymore. The only way is to consider user preferences, maximize the number of healthy compounds and minimize the unhealthy ones in food, is using recommendation systems.

So, the result is creating a food recommendation application by taking user details like their age, weight, height, sugar, BP, allergic food item and providing a diet based on the info the user gave. Also, the user can give their feedbacks which can be useful for the users to evaluate the recommendations and the application.

4.2 DESCRIPTION OF SCREENSHOTS, GRAPHS AND TABLES

GUI of the website consists of Sign-In, Sign-Up and home page, where the user logins to their account using their username and password from Sign-In page, if it already exists in Database the user can directly login into the account and use the features.

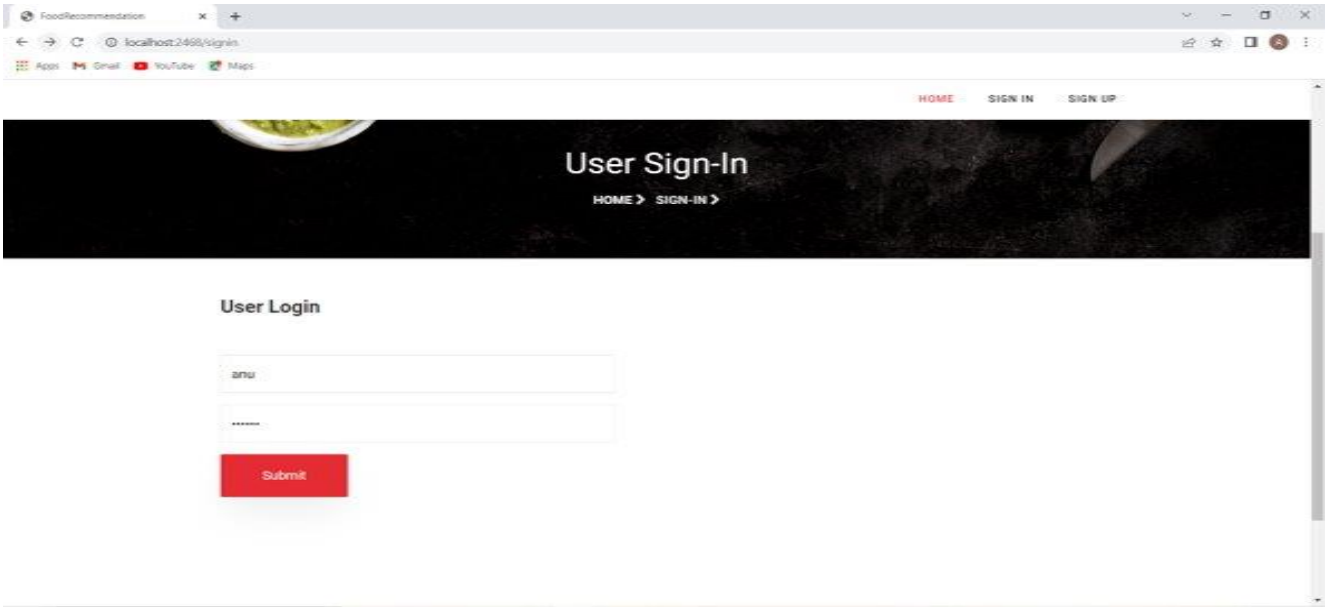


Figure 4.1 Sign in Page

If account doesn’t exist then the user has to Sign-up using name, username, password, email, phone number so that they can use the features of the website,

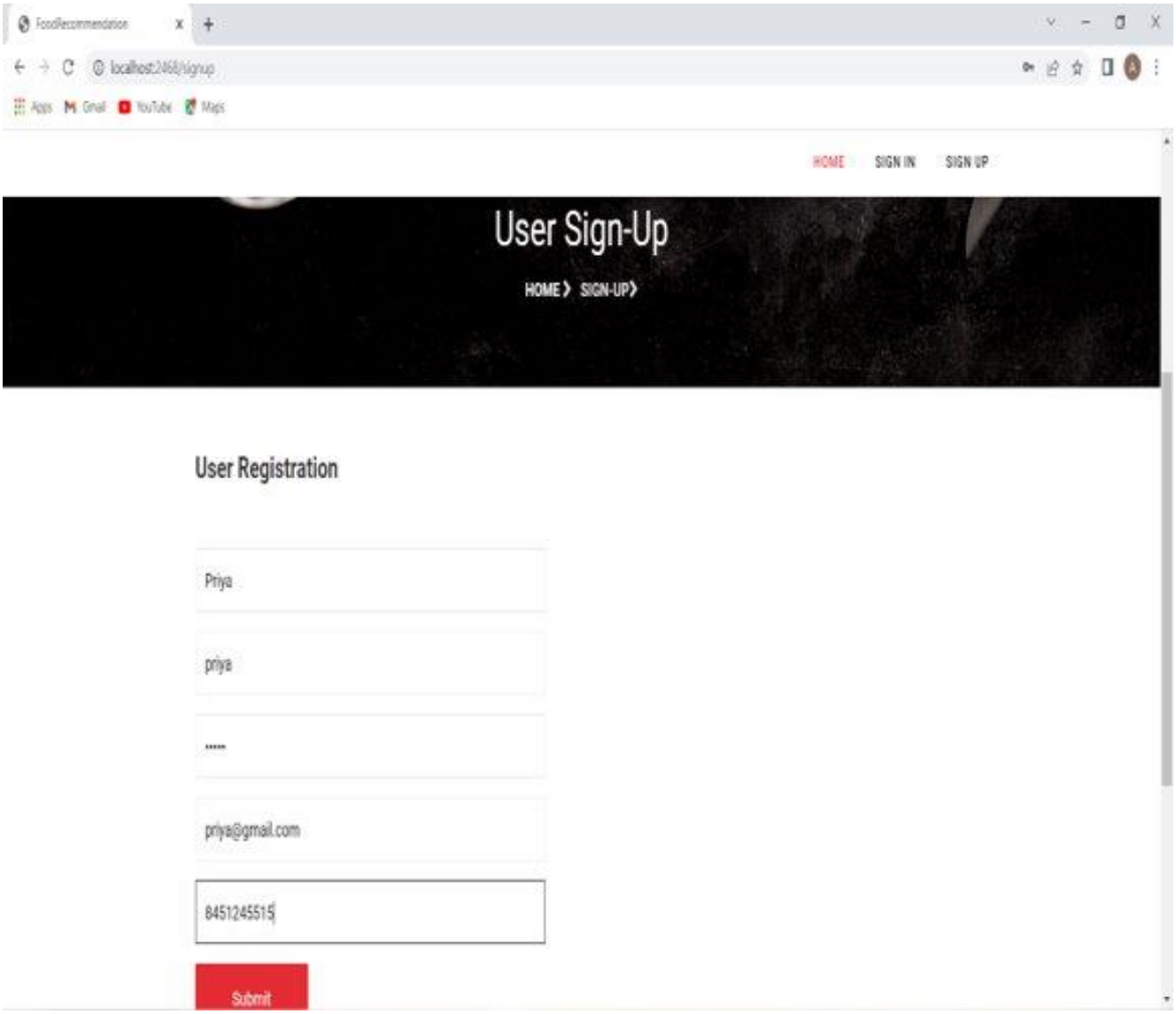


Figure 4.2 Sign Up Page

After logging in to their account, there is a home page where the user has to give details for the diet to be provided. Details like age, weight, gender, Diet Category, sugar, BP, allergic to any food item.

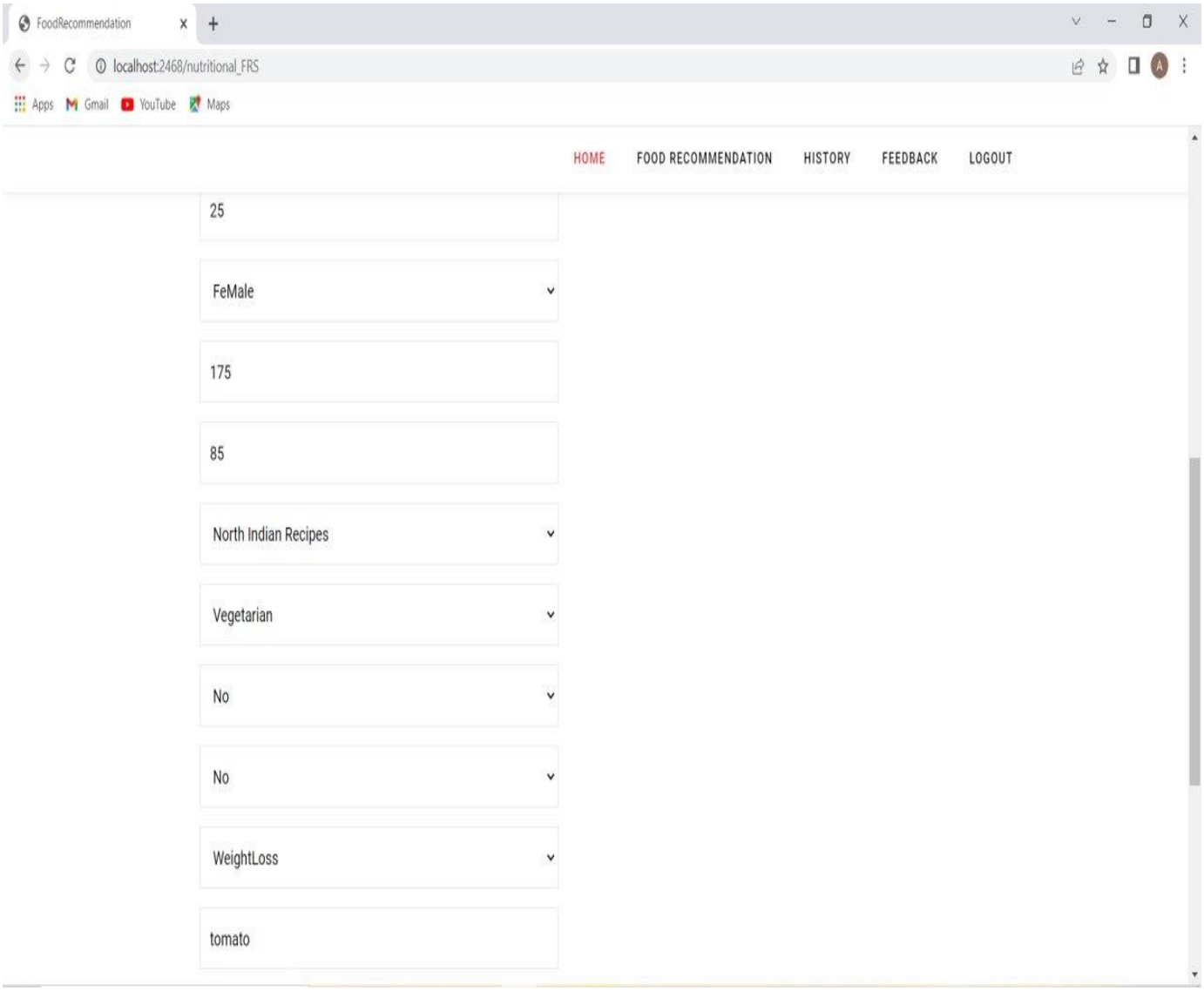
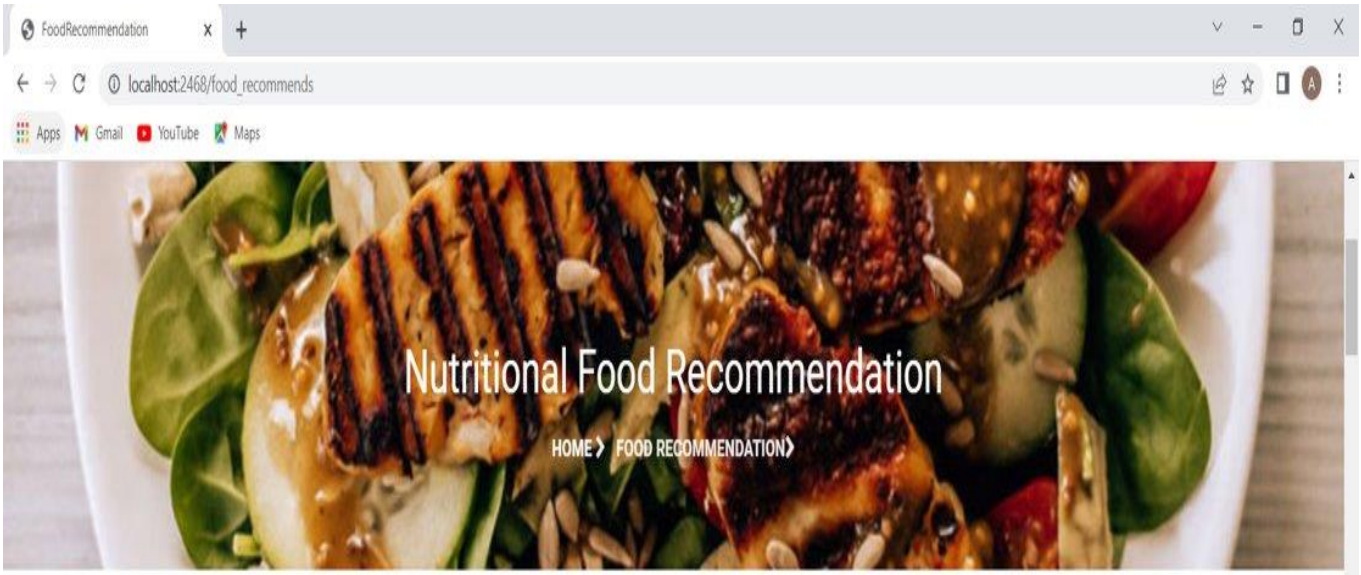


Figure 4.3 Home Page

After entering details, the diet along with recipe instructions is provided in food recommendations tab, for all times of the day i.e., breakfast, lunch and dinner.



Category	Recipe Name	INSTRUCTIONS
Breakfast	Kesar Chai Recipe - A Delicious Saffron Flavored Indian Tea	b'To begin making the Kesar Chai, add water into a saucepan, along with the saffron strands and allow it to come to a brisk boil.Once it begins to boil add the tea leaves. Allow the tea leaves to simmer in the water for about a minute and turn off the heat.Once you add the tea leaves, allow it to brew just a little to ensure the tea stays light and does not turn bitter.Stir in the milk into the chai and allow it to rest for a few seconds.Strain the Kesar chai into the cups and serve. Add a few more saffron strands to the top while serving, this will give a great look and also add to the taste and flavor of the chai.Serve the Kesar chai Recipe along withSamosaorVegetable Bajji or Pakorafor an evening snack.'
Breakfast	Dahi Bhalla Papdi Chaat Recipe	b'To begin making the Dahi Bhalla Papdi Chaat, you can either use the puris or papdi. We will be making a plate at a time.In a flat plate or a bowl arrange the papdis or puris. Add the dahi bhalla over the papdi or into the puri. If your bhallas are large, then break them into small pieces and add them. Pour some of the dahi as well.Next drizzle the date chutney and the green chutneys around each of the papdi/ puris. Sprinkle chaat masala powder and the cumin powder. Next sprinkle the grated carrots, pomegranates and the crisp boondi/sev.Finally sprinkle the onions and the chopped coriander leaves. The chaat is now ready to be served. Proceed the same way with the remaining Bhallas and Papdi.Serve theDahi Bhalla Papdi (Puri) Chaat Recipe as anAppetizeror as asTea Time Snackto fill your hunger craving.'

Figure 4.4 Recommendations Page

The user's feedbacks are displayed at the end of the page, which can be useful for the users to know other's genuine opinion about the application.

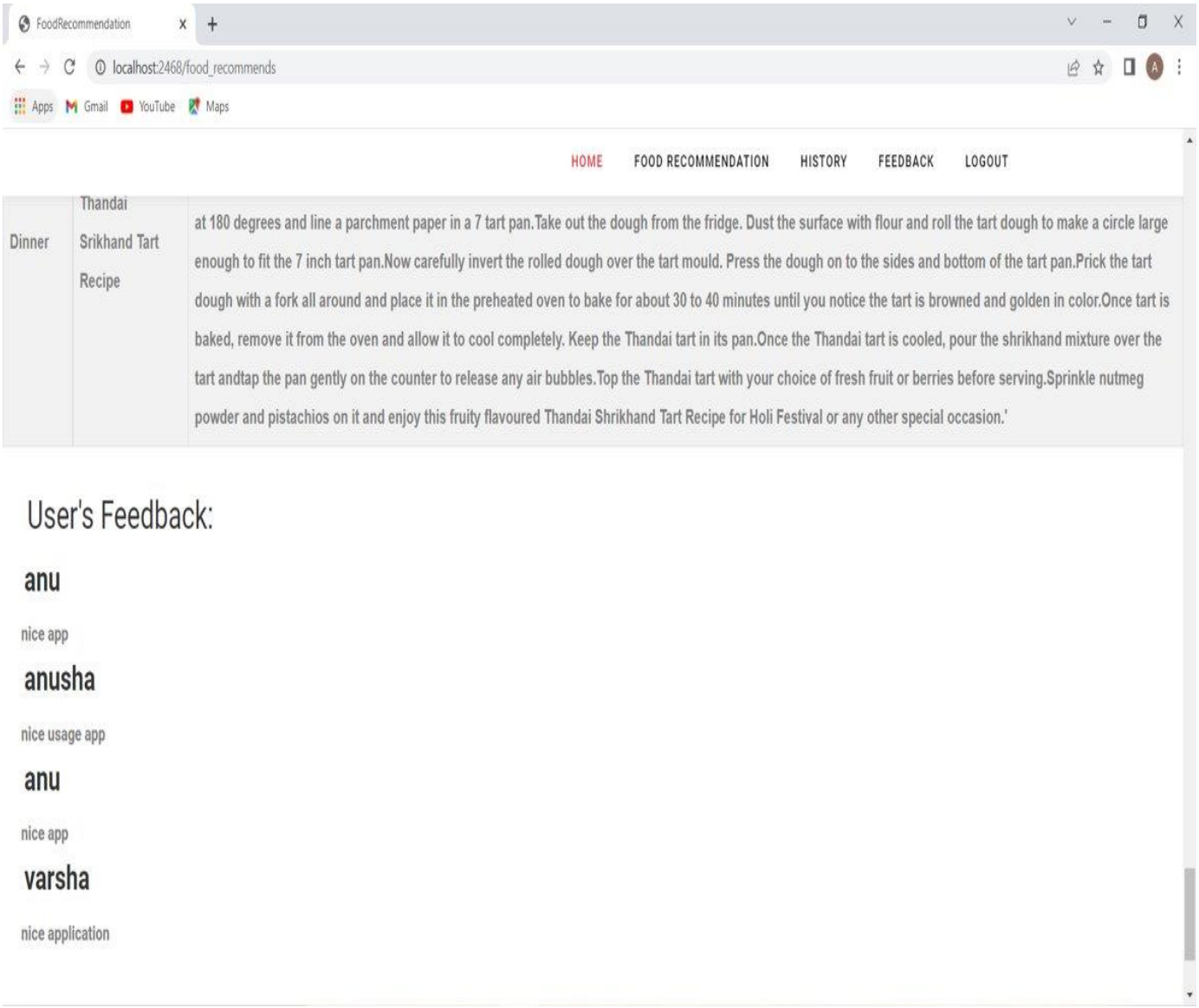
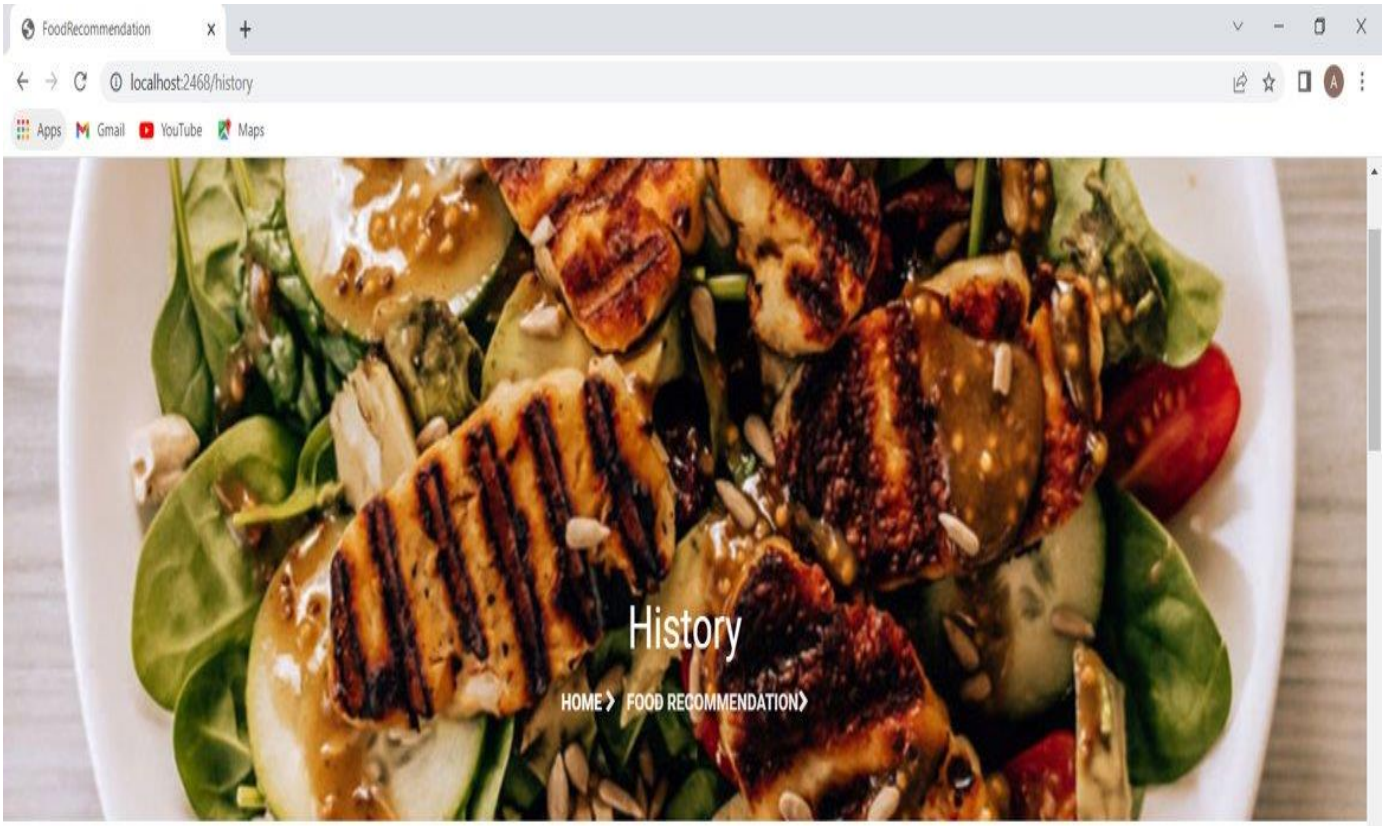


Figure 4.5 Users Feedbacks

In History tab, all the previous data given is stored of the user, so that user can go back and view the diets easily.



Age	Gender	Height	Weight	Cuisine	category	Sugar	BP	Diet_catgry	FoodRecommends
25	FeMale	180	70	South Indian Recipes	Vegetarian	No	No	WeightLoss	View
15	Male	185	95	South Indian Recipes	Vegetarian	Yes	Yes	WeightLoss	View
25	FeMale	180	85	South Indian Recipes	Vegetarian	No	No	WeightLoss	View
25	FeMale	180	85	North Indian Recipes	Vegetarian	No	No	WeightLoss	View

Figure 4.6 History Page

In Feedback Page, the user can give feedback which is stored in the backend feedback table.

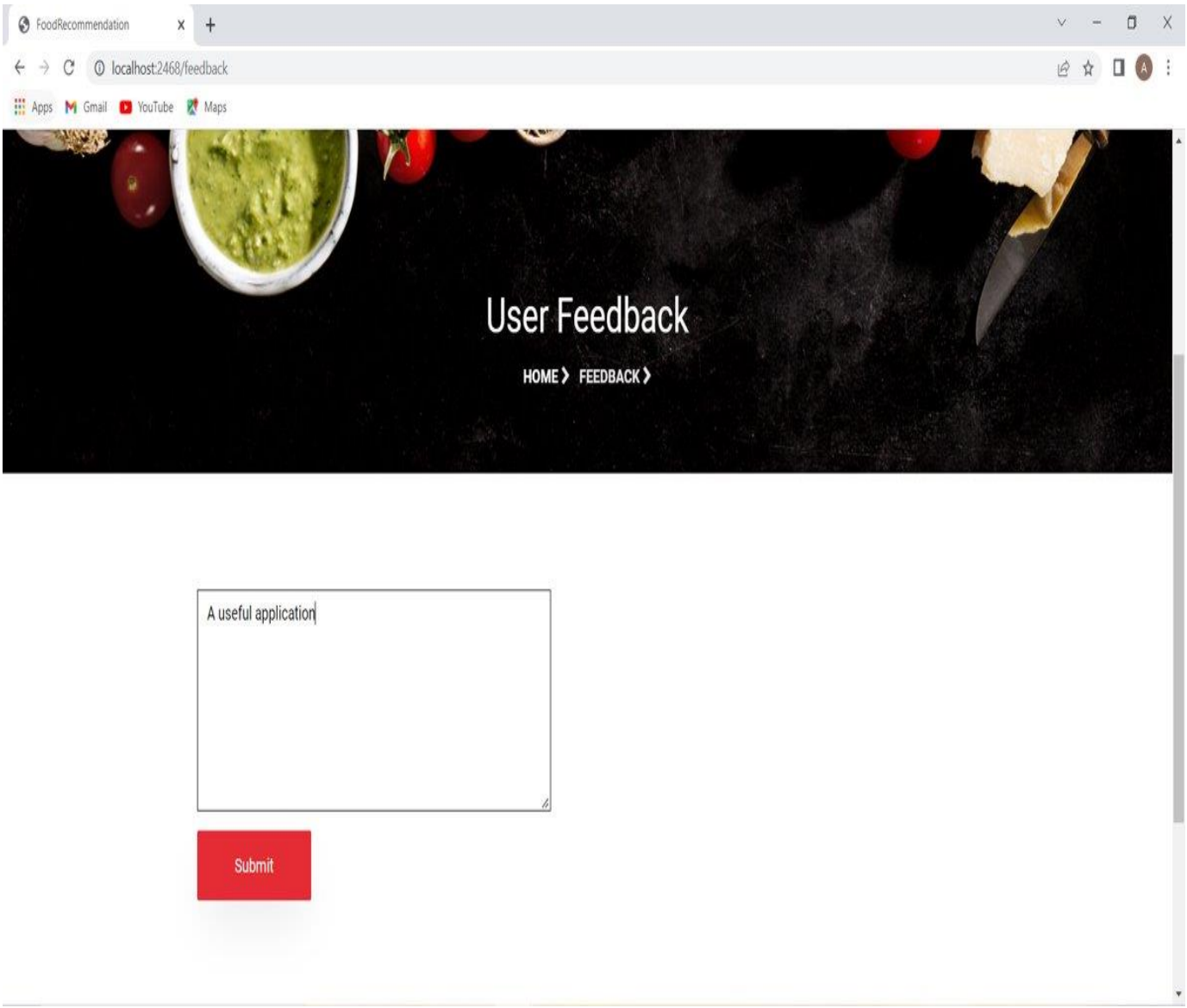


Figure 4.7 Feedback Page

At Backend, Food recommendations is the database name. It has multiple tables for feedback, food recommends, history, register. Where each table stores the data the is either given by user or the that is processed to give output.

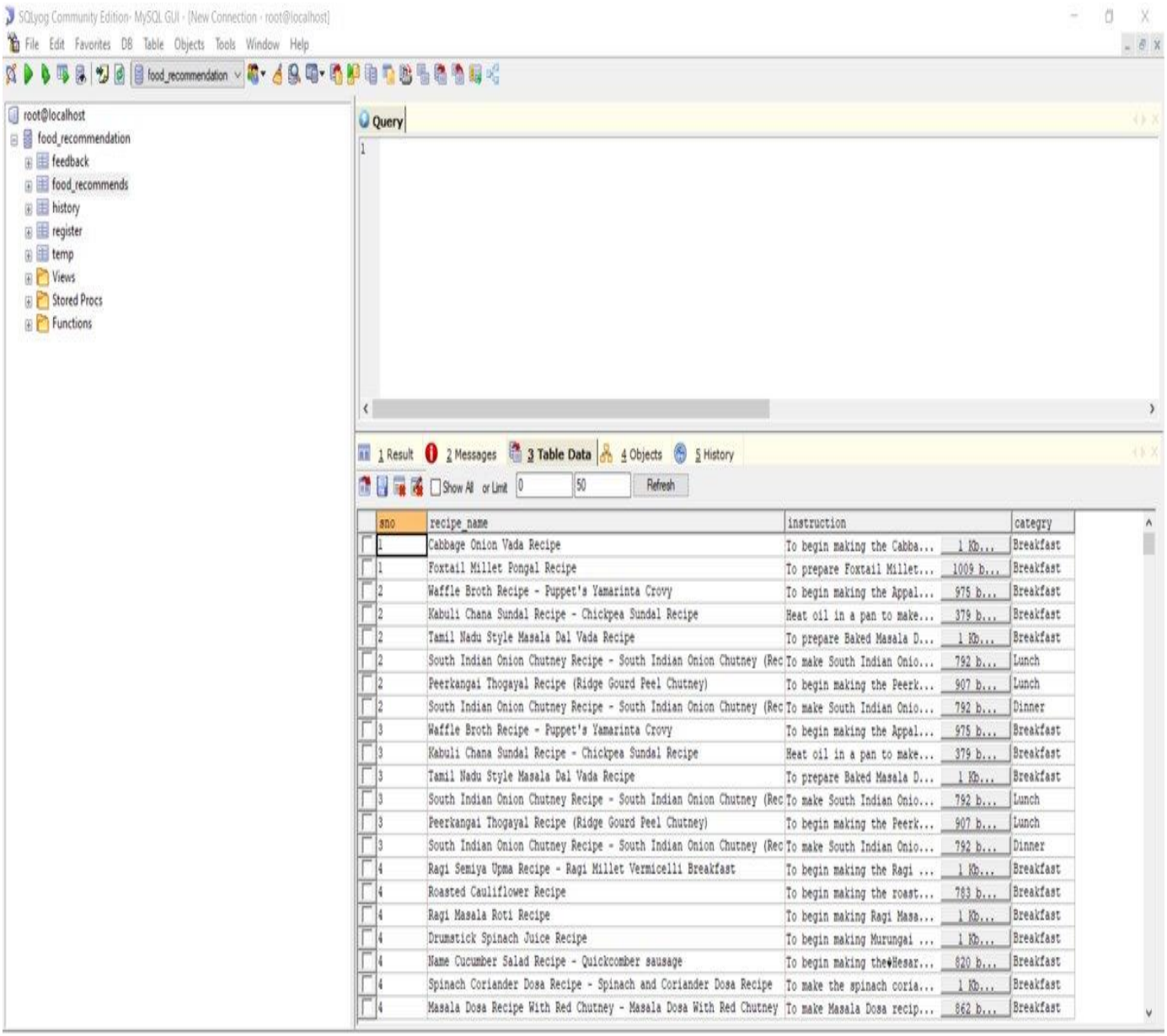


Figure 4.8 Database

5.CONCLUSIONS AND FUTURE WORK

Many factors influence an individual's health, such as physical exercise, sleep, nutrition, heredity and pollution. Being nutrition one of the biggest modifiable factors in our lives, small changes can have a big impact. With the exponential increase in the number of available food options, it is not possible to take them all into account anymore. The only way is to consider user preferences, maximize the number of healthy compounds and minimize the unhealthy ones in food, is using recommendation systems.

In this modern world various people suffer from different types of diseases and illnesses. It is generally very difficult to suggest a diet as quickly as possible. Most people have a dire need to Lose Weight, Gain Weight or stay Healthy. Time has also become a possible constraint. The project makes use of a dataset which contains various nutrients in the correct amount. In the wake of the situation, we have tried to develop a ML program that recommends diet to the people. The items recommended are limited to three categories: Weight Loss, Weight Gain and Healthy category. We have used Machine Learning Algorithms named K-Means Clustering for clustering the data and Random Forest Classification to classify according to the categories listed.

A graphical user interface which is used to predict the food items is to be developed, the Diet Recommendation System which uses user inputs from a Graphical User Interface including age, height, weight, vegetarian or non-vegetarian food etc. and selecting the above three categories. The working prototype of the Diet Recommendation System lists a set of food items as per the user inputs.

The future scope of the working prototype is to develop a mobile application which is compatible to use in the mobiles and also improving the recommendation system into a more precise one by including some more clinical parameters like cholesterol, muscle strength, etc., which also play a important role in the health condition of a person. By considering these parameters the system will become more accurate in recommending food items in regard with their health.

REFERENCES

[1] A. Taweel, L. Barakat, S. Miles, T. Cioara, I. Anghel, A.-R. H. Tawil, and I. Salomie, ‘‘A service-based system for malnutrition prevention and selfmanagement,’’ Comput. Standards Inter., vol. 48, pp. 225–233, Nov. 2016.

[2] B. Hernández-Ocaña, O. Chávez-Bosquez, J. Hernández-Torruco, J. Canul-Reich, and P. PozosParra, ‘‘Bacterial foraging optimization algorithm for menu planning,’’ IEEE Access, vol. 6, pp. 8619– 8629, 2018.

[3] B. Yatsalo and L. Martínez, ‘‘Fuzzy rank acceptability analysis: A confidence measure of ranking fuzzy numbers,’’ IEEE Trans. Fuzzy Syst., vol. 26, no. 6, pp. 3579–3593, Dec. 2018.

[4] C. Trattner and D. Elsweiler, ‘‘Food recommender systems: Important contributions, challenges and future research directions,’’ Nov. 2017, arXiv:1711.02760. [Online]. Available: https://arxiv.org/abs/1711.02760

[5] C. Zopounidis and M. Doumpos, ‘‘Multicriteria classification and sorting methods: A literature review,’’ Eur. J. Oper. Res., vol. 138, no. 2, pp. 229–246, 2002.

[6] D. Bianchini, V. De Antonellis, N. De Franceschi, and M. Melchiori, ‘‘PREFer: A prescriptionbased food recommender system,’’ Comput. Standards Inter., vol. 54, pp. 64–75, Nov. 2017.

[7] D. Ntalaperas, E. Bothos, K. Perakis, B. Magoutas, and G. Mentzas, ‘‘DISYS: An intelligent system for personalized nutritional recommendations in restaurants,’’ in Proc. 19th Panhellenic Conf. Inform., 2015, pp. 382–387.

[8] D. Ribeiro, J. Machado, M. J. M. Vasconcelos, E. Vieira, and A. C. de Barros, ‘‘Souschef: Mobile meal recommender system for older adults,’’ in Proc. 3rd Int. Conf. Inf. Commun. Technol. Ageing Well EHealth, 2017, pp. 36–45.

[9] F. Mata, M. Torres-Ruiz, R. Zagal, G. Guzman, M. Moreno-Ibarra, and R. Quintero, ‘‘A cross-domain framework for designing healthcare mobile applications mining social networks to generate recommendations of training and nutrition planning,’’ Telematics Informat., vol. 35, no. 4, pp. 837–853, 2018.

[10] F. Rehman, O. Khalid, N. ul Haq, A. ur Rehman Khan, K. Bilal, and S. A. Madani, ‘‘Diet-right: A smart food recommendation system,’’ KSII Trans. Internet Inf. Syst., vol. 11, no. 6, pp. 2910–2925, 2017.

[11] G. Agapito, M. Simeoni, B. Calabrese, I. Caré, T. Lamprinoudi, P. H. Guzzi, A. Pujia, G. Fuiano, and M. Cannataro, ‘‘DIETOS: A dietary recommender system for chronic diseases monitoring and management,’’ Comput. Methods Programs Biomed., vol. 153, pp. 93–104, Jan. 2018.

[12] J. L. Balintfy, ‘‘Menu planning by computer,’’ Commun. ACM, vol. 7, no. 4, pp. 255–259, 1964.

[13] L. Yang, C.-K. Hsieh, H. Yang, J. P. Pollak, N. Dell, S. Belongie, C. Cole, and D. Estrin, ‘‘Yum-Me: A personalized nutrient-based meal recommender system,’’ ACM Trans. Inf. Syst., vol. 36, no. 1, p. 7, 2017.

[14] M. F. Syahputra, V. Felicia, R. F. Rahmat, and R. Budiarto, ‘‘Scheduling diet for diabetes mellitus patients using genetic algorithm,’’ J. Phys., Conf. Ser., vol. 801, no. 1, pp. 1–9, 2017.

[15] M. Ge, F. Ricci, and D. Massimo, ‘‘Health-aware food recommender system,’’ in Proc. ACM Int. Conf. Recommender Syst., 2015, pp. 333–334.

[16] N. Nag, V. Pandey, and R. Jain, ‘‘Live personalized nutrition recommendation engine,’’ in Proc. 2nd Int. Workshop Multimedia Pers. Health Health Care, 2017, pp. 61–68.

[17] R. Yera and L. Martínez, ‘‘A recommendation approach for programming online judges supported by data preprocessing techniques,’’ Appl. Intell., vol. 47, no. 2, pp. 277–290, 2017.

[18] R. Yera and L. Martínez, ‘‘Fuzzy tools in recommender systems: A survey,’’ Int. J. Comput. Intell. Syst., vol. 10, no. 1, pp. 776–803, 2017.

[19] R. Y. Toledo, Y. C. Mota, and L. Martínez, ‘‘Correcting noisy ratings in collaborative recommender systems,’’ Knowl.-Based Syst., vol. 76, pp. 96–108, Mar. 2015.

[20] T. Cioara, I. Anghel, I. Salomie, L. Barakat, S. Miles, D. Reidlinger, A. Taweel, C. Dobre, and F. Pop, ‘‘Expert system for nutrition care process of older adults,’’ Future Gener. Comput. Syst., vol. 80, pp. 368–383, Mar. 2016.

[21] T. N. Trang-Tran, M. Atas, A. Felfernig, and M. Stettinger, ‘‘An overview of recommender systems in the healthy food domain,’’ J. Intell. Inf. Syst., vol. 50, pp. 501–526, Jun. 2018.

[22] V. Espín, M. V. Hurtado, and M. Noguera, ‘‘Nutrition for elder care: A nutritional semantic recommender system for the elderly,’’ Expert Syst., vol. 33, no. 2, pp. 201–210, 2016.

[23] Y. Zuo, M. Gong, J. Zeng, L. Ma, and L. Jiao, ‘‘Personalized recommendation based on evolutionary multi-objective optimization [research frontier],’’ IEEE Comput. Intell. Mag., vol. 10, no. 1, pp. 52–62, Feb. 2015.

GLOSSARY

Decision Tree: A decision tree is a flowchart-like structure in which each internal node represents a test on a feature.

Flask: Flask is a Python-based microframework used for developing small scale websites.

GUI: A graphical user interface (GUI) is an interface through which a user interacts with electronic devices such as computers and smartphones.

K-means Clustering: K-Means Clustering is an Unsupervised Learning algorithm. It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs to only one group that has similar properties.

Machine learning: Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

Random Forest Classifier: Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

SQLyog: SQLyog is a content management tool for Windows that lets you visually compare databases and create schemas.

APPENDIX

BMI - Body mass index. A measure that relates body weight to height. BMI is sometimes used to measure total body fat and whether a person is a healthy weight. Excess body fat is linked to an increased risk of some diseases including heart disease and some cancers.

Clustering - Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

Classifier - A classifier in machine learning is an algorithm that automatically orders or categorizes data into one or more of a set of classes.

Food Recommendation System - Food recommendation systems (RSs) are software systems that make personalized recommendations form a large range of different options and thus provide a promising solution for information overload and unhealthy food decisions.