# Diet Recommendation

**Session: 2022-2026**

**Submitted by:**

| | |
|---|---|
| Saba Shahdin | 2022-CS-112 |
| Afeera Fatima | 2022-CS-151 |

**Submitted to:**

Dr. Samyan Wahla

**Course:**

CS-371 Artificial Intelligence

**Department of Computer Science**

**University Of Engineering And Technology,**

**Lahore, Pakistan**

## Abstract

This report presents the design, development, and implementation of a personalized Diet Recommendation System that utilizes machine learning algorithms to offer customized diet plans based on individual health data and lifestyle preferences. The system takes into account various factors such as body mass, activity levels, age, height, weight, and food allergies to generate diet suggestions tailored to users' specific needs. By collecting and analyzing user input, the application provides personalized meal plans that promote balanced nutrition and encourage healthier eating habits.

In addition to diet recommendations, the system offers features for users to track their daily meals, monitor their progress, and adjust their diet plans based on changing health goals. The user-friendly interface ensures easy navigation through steps such as profile creation, activity level selection, allergy input, and confirmation of dietary preferences.

The implementation of machine learning models allows for dynamic and adaptable suggestions, improving as more user data is collected over time. This report covers the system architecture, the algorithms used to generate diet plans, and the overall user experience. It also highlights the testing phase and evaluation metrics to assess the system's accuracy and effectiveness in meeting dietary goals.

Finally, the report outlines potential future enhancements, including the integration of more advanced machine learning techniques and expansion of the system's features to accommodate a wider range of dietary requirements, aiming to further personalize and improve the user experience.

# Contents

## List of Figures

## List of Tables

# 1 Introduction

## 1.1 Background

In recent years, there has been a growing awareness of the importance of maintaining a healthy lifestyle, with a significant focus on diet and nutrition. With the rise of sedentary lifestyles, poor eating habits, and the increasing prevalence of lifestyle, diseases and specific mood swings people are more conscious of the need to adopt healthy eating practices. However, finding personalized and effective diet plans can be a challenge due to the vast amount of dietary information available and the individual variability in nutritional needs.

A diet recommendation system addresses this issue by providing tailored meal plans based on a user's specific health data, preferences, and goals. Such systems leverage advances in technology, particularly machine learning, to analyze individual health factors and suggest diets that optimize nutritional intake. This can help users make informed decisions about their food choices, manage their health better, and achieve their wellness goals.

In today's fast-paced world, where individuals often struggle to plan their meals or stick to healthy eating habits, a diet recommendation system offers a practical solution. By automating the process of meal planning and considering various factors such as age, weight, activity level, and user food alergies, this system can play a pivotal role in promoting long-term health and well-being. The importance of such a system is particularly evident in the context of increasing reliance on technology for health management and the growing demand for personalized health solutions.

## 1.2 Problem Statement

In today's health-conscious society, many individuals struggle to manage their diet due to food allergies (e.g., milk, meat, gluten) and their physique that require specific dietary adjustments, such as maintaining a healthy BMI. Existing diet recommendation systems often overlook these factors, providing generic meal plans that may not account for allergens or the user's BMI class. This project aims to address this gap by developing a personalized diet recommendation system that considers both food allergies and health metrics, such as BMI and activity levels, ensuring safe and tailored meal plans for users.

## 1.3 Objectives

The primary objectives of this project are:

- To develop a machine learning-based diet recommendation application that offers personalized meal plans based on user preferences e.g. weight, age and height, and food allergies (e.g., milk, meat, gluten).
- To create a user-friendly mobile app that allows users to receive dietary suggestions tailored to their nutritional needs while ensuring allergen-free meal options.

- To implement a machine learning model that accurately predicts and suggests dietary plans based on factors such as age, gender, weight, activity level, and allergies to specific foods.
- To evaluate the performance of the system in terms of accuracy, user satisfaction, and its ability to promote healthier eating habits while ensuring allergen safety.

## 1.4 Scope of the Report

This report covers the design, implementation, and evaluation of a diet recommendation system that takes food allergies into account, including allergens such as milk, meat, gluten, and others. The following key areas will be discussed:

- **System Design and Architecture:** The overall design of the diet recommendation app, including the architecture, user flow, and technology stack used.
- **Machine Learning Approach:** The selection, training, and evaluation of the machine learning model used to generate personalized diet recommendations considering user preferences while avoiding allergens.
- **App Development:** The process of developing both the front-end and back-end of the application, including UI/UX design, database design, and API integration.
- **Implementation:** The specific features implemented in the app, including allergen detection and meal recommendations, challenges faced during development, and solutions to overcome them.
- **Results and Evaluation:** The performance of the app in terms of user engagement, satisfaction, and its ability to meet personalized dietary needs while avoiding allergens.

The report concludes with recommendations for future enhancements and potential areas for further research in the field of allergen-aware diet recommendation systems.

## 2 Literature Review

Personalized diet recommendation systems have gained significant attention in recent years due to the increasing need for tailored health solutions. Traditional diet plans, while effective to some extent, often fail to account for individual differences such as food preferences, allergies, and health conditions. This has led to the development of more advanced systems that leverage machine learning algorithms to provide customized meal plans. These systems aim to optimize nutrient intake, reduce the risk of adverse health effects, and improve overall well-being by considering a variety of personal factors.

One of the key areas in personalized diet recommendation systems is the inclusion of food allergies. Many individuals suffer from allergies to common foods like milk, meat, gluten, making it essential for diet planning systems to factor in these restrictions. Without this consideration, users may inadvertently consume foods that trigger allergic reactions, leading to serious health risks. The integration of food allergies into diet recommendation systems ensures a safer and more effective meal plan for users.

Another important aspect is the incorporation of health metrics such as BMI (Body Mass Index), which plays a crucial role in determining an individual's nutritional needs. By analyzing BMI alongside other health data, personalized systems can better understand a user's health status and provide diet recommendations that align with their goals, whether it's weight loss, weight maintenance, or improving overall health.

Despite the progress made in developing these systems, several challenges remain. The complexity of accurately predicting a user's dietary needs, the vast array of food items and allergens, and the need for real-time dietary updates are some of the hurdles that need to be addressed. Additionally, existing systems may not always be accessible to a wide range of users, particularly those with unique dietary restrictions or limited access to technology.

This project addresses these gaps by creating a diet recommendation system that not only considers food allergies but also utilizes machine learning algorithms to personalize diet plans based on an individual's health metrics, preferences, and lifestyle. By integrating these features, the system aims to provide a more holistic and user-friendly approach to nutrition management, ensuring that users receive accurate, safe, and effective diet recommendations tailored to their specific needs.

## 3 Methodology

The system was designed to provide personalized diet recommendations based on user information, physical activity level, and food preferences. The methodology can be broken down into the following steps:

## 3.1 Data Collection

The system collects essential user details such as age, weight, height, gender, and physical activity level through an interactive user interface. The physical activity level is categorized into three groups: active, sedentary, or light. In addition, the system allows users to specify any food allergies, ensuring the meal suggestions are safe for consumption.

## 3.2 System Design

The system consists of a frontend interface developed using `Flutter` , which communicates with a `Node.js` backend. The backend processes the user input, performs calculations, and provides the necessary diet recommendations. The data is stored in a `MySQL` database.

## 3.3 BMI and Nutrient Calculations

The Body Mass Index (BMI) is calculated using the standard formula:

$$BMI = \frac{\text{Weight (kg)}}{\text{Height (m)}^2}$$

Daily calorie needs and macronutrient distribution (carbohydrates, protein, fats) are calculated using the **Mifflin-St Jeor** equation. The BMR is adjusted based on the user's activity level to determine the appropriate calorie intake for maintaining, losing, or gaining weight.

## 3.4 Machine Learning Model

A machine learning model was trained using user data to predict body mass. The model was selected based on its accuracy in predicting the BMI class and was evaluated using accuracy , precision , F1-Score and recall.

## 3.5 Diet Recommendation

The system uses a recommendation algorithm to suggest three meal options for each of the four meal types. These suggestions are based on the user's macronutrient requirements and meal preferences. Additionally, the system removes any allergens specified by the user from the meal options.

## 3.6 Meal Customization

Users can update their meal preferences, dietary restrictions, and daily meal choices through the system's interface. This allows the system to provide customized meal recommendations and adapt to changing dietary needs.

### 3.7 Technologies and Tools

The backend of the system was developed using `Node.js` and `Express`, while the frontend was implemented using `Flutter`. A `MySQL` database was used to store user and meal data. Machine learning models were trained using `Python` and `Scikit-learn`, and the system utilizes REST APIs for communication between the frontend and backend.

| **Component** | **Technology Used** |
|---|---|
| **Backend** | Node.js, Express |
| **Frontend** | Flutter |
| **Database** | MySQL |
| **Machine Learning** | Python, Scikit-learn |
| **Communication** | REST APIs |

Table 1: Technology Stack

## 4 Machine Learning Model

### 4.1 Data Collection

- The data was collected form the open source as well as through AP. For the prediction of **BMI Category** i.e. Normal Weight , Underweight , Overweight , Obese Class 1 , Obese Class 2 , Obese Class 3 . This data has the input features of Age , weight , height and BMI.
- For the recommendation of food we use the **Spoonacular** API to fetch the data of Asian Cuisine. Our app focus on Asian cuisines only for now. It gets the food Item their calories , micronutrient and the ingredients to give user centric results

### 4.2 Model Selection

- The machine learning model of **K Nearest Neighbor** is used to train our model which predicts the BMI class according to user input. First the model is trained on 800 users and then it is test on unknown new user and this will predict the BMI class. The model has an accuracy of 0.92 which give the insights that the model si working fine.
- To recommend food on the basis of suer preference we use the AI based based recommendation technique of **Content Based Filtering** which works on recommending the food based on the calculated cosine similarity score index which helps to get the results more closer to the user preferences.
- We use simple Natural Language technique to get the insights form the food allergies input which is given in comma separated form. The model analyze the user input it also check for the type mistakes using fuzzy library and also the phrase matcher to extract the food with the ingredient so that they don’t recommend to the user.

## 4.3 Prediction of the Body Mass

The primary objective of this project was to predict a model to classify categories of BMI (body mass index) based on characteristics such as age, height, weight and BMI. The data set was processed and modeled using K-Nearest Neighbors (KNN), while handling class imbalance using SMOTE (Synthetic Minority Oversampling Technique). Evaluation metrics and visualizations were also included to measure the performance of the model.

### 4.3.1 Dataset

We get the data set from the open source of different users age , weight , height and bmi as input features and based on this our model predicts the bmi class in which the user lies. The output feature is divided into 6 classes.

- Normal Weight
- Underweight
- Overweight
- Obese Class 1
- Obese Class 2
- Obese Class 3

### 4.3.2 Tools and Frameworks

we used the python machine learning models to train our system and to predict BMI. The data set classifies the user into six classes.K Nearest neighbors helps to train the model in order to predict the BMI class which give the accuracy of 0.92.

### 4.3.3 Implementation

The implementation starts by preprocessing the data in order to remove the missing values which affect the model prediction. Then we perform **Standardization** technique in order to make all the features on one scale. If one feature has highest range of values it become more prominent in the prediction. When we deal with the categorical data we encode it to the numerical data so we used **One Hot Encoding**.

### 4.3.4 Class Imbalance

Class imbalance occurs when the distribution of target classes in a dataset one or more classes have far fewer instances than others. This imbalance often leads to machine learning models being biased towards the majority class, as they tend to predict the more frequent classes more accurately while ignoring the minority classes.The target variable, BMI Class, represents categories such as Underweight, Normal, Overweight, and Obese.These classes are likely to be unevenly distributed because, in real-world populations Most individuals tend to fall in the

Normal or Overweight categories.Fewer people may be classified as Underweight or Obese, depending on regional health statistics and sample demographics.

### 4.3.5 Reason for Class Imbalance

- The BMI class distribution in the dataset likely reflects real-world health data, where the majority of individuals are in the "Normal" or "Overweight" categories. The "Underweight" and "Obese" categories often represent smaller proportions of the population.
- Sample Collection Bias: If the dataset was collected from a population with specific demographic characteristics (e.g., age group, region, or health status), the distribution may inherently favor some classes over others.
- Health Trends:In many regions, health trends like increased rates of overweight or obesity due to sedentary lifestyles and unhealthy diets can result in an over representation of these categories compared to "Underweight."
- Dataset Size:Smaller datasets are more prone to class imbalance because they may not capture the full diversity of the population.

### 4.3.6 Impact of Class Imbalance on the Model

- Bias Toward Majority Classes: Machine learning models, including KNN, tend to favor majority classes. This is because the model optimizes for overall accuracy, often at the expense of minority classes.
- Poor Prediction for Minority Classes: Without addressing the imbalance, predictions for "Underweight" and "Obese" classes may be inaccurate or non-existent.

### 4.3.7 Evaluation Metrics

## Accuracy

### Formula for Accuracy

The formula for Accuracy for each class is:

$$\text{Accuracy}_{\text{Class}} = \frac{\text{True Positives (TP)}}{\text{Total Instances for Class}}$$

Where:

- True Positives (TP): The number of correct predictions for the class.
- Total Instances for Class: The total number of true instances (support) of the class in the dataset.

### Class-Wise Accuracy Calculations

For your dataset, the Accuracy for each class can be calculated as follows:

| Class | True Positives (TP) | Total Instances (Support) | Accuracy |
|---|---|---|---|
| Normal Weight | 100 | 101 | 0.9901 |
| Obese Class 1 | 1 | 6 | 0.1667 |
| Obese Class 2 | 13 | 13 | 1.0000 |
| Obese Class 3 | 16 | 19 | 0.8421 |
| Overweight | 52 | 57 | 0.9123 |
| Underweight | 25 | 27 | 0.9259 |

Table 2: Class-wise Accuracy Calculations

### Interpretation

The accuracy for each class measures the fraction of correctly classified instances for that specific class. Here's the breakdown of the calculated values:

- **Normal Weight**: 0.9901 (99.01% of instances correctly classified as "Normal Weight").
- **Obese Class 1**: 0.1667 (16.67% of instances correctly classified as "Obese Class 1").
- **Obese Class 2**: 1.0000 (100% of instances correctly classified as "Obese Class 2").
- **Obese Class 3**: 0.8421 (84.21% of instances correctly classified as "Obese Class 3").
- **Overweight**: 0.9123 (91.23% of instances correctly classified as "Overweight").
- **Underweight**: 0.9259 (92.59% of instances correctly classified as "Underweight").

## Precision Calculation

### Formula for Precision

The formula for Precision for each class is:

$$\text{Precision}_{\text{Class}} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

Where:

- True Positives (TP): The number of correct predictions for the class.
- False Positives (FP): Instances predicted as the class but actually belonging to other classes.

## Class-Wise Precision Calculations

For your dataset, the Precision for each class can be calculated as follows:

| Class | True Positives (TP) | False Positives (FP) | Precision |
|---|---|---|---|
| Normal Weight | 100 | 8 | 0.93 |
| Obese Class 1 | 1 | 0 | 1.00 |
| Obese Class 2 | 13 | 7 | 0.65 |
| Obese Class 3 | 16 | 0 | 1.00 |
| Overweight | 52 | 2 | 0.96 |
| Underweight | 25 | 0 | 1.00 |

Table 3: Class-wise Precision Calculations

## Interpretation

Precision measures the proportion of correct predictions out of all predictions made for a specific class. Here's the breakdown of the calculated values:

- **Normal Weight**: 0.93 (93% of instances predicted as "Normal Weight" were correct).
- **Obese Class 1**: 1.00 (100% of instances predicted as "Obese Class 1" were correct).
- **Obese Class 2**: 0.65 (65% of instances predicted as "Obese Class 2" were correct).
- **Obese Class 3**: 1.00 (100% of instances predicted as "Obese Class 3" were correct).
- **Overweight**: 0.96 (96% of instances predicted as "Overweight" were correct).
- **Underweight**: 1.00 (100% of instances predicted as "Underweight" were correct).

## Recall Calculation

### Formula for Recall

The formula for Recall for each class is:

$$\text{Recall}_{\text{Class}} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

Where:

- True Positives (TP): The number of correct predictions for the class.
- False Negatives (FN): Instances of the class incorrectly classified as other classes.

### Class-Wise Recall Calculations

For your dataset, the Recall for each class can be calculated as follows:

| **Class** | **True Positives (TP)** | **False Negatives (FN)** | **Recall** |
|---|---|---|---|
| Normal Weight | 100 | 1 | 0.99 |
| Obese Class 1 | 1 | 5 | 0.17 |
| Obese Class 2 | 13 | 0 | 1.00 |
| Obese Class 3 | 16 | 3 | 0.84 |
| Overweight | 52 | 5 | 0.91 |
| Underweight | 25 | 2 | 0.93 |

Table 4: Class-wise Recall Calculations

### Interpretation

Recall measures the ability of the classifier to correctly identify all instances of a specific class. Here's the breakdown of the calculated values:

- **Normal Weight**: 0.99 (99% of all "Normal Weight" instances were correctly identified).
- **Obese Class 1**: 0.17 (17% of all "Obese Class 1" instances were correctly identified).
- **Obese Class 2**: 1.00 (100% of all "Obese Class 2" instances were correctly identified).
- **Obese Class 3**: 0.84 (84% of all "Obese Class 3" instances were correctly identified).
- **Overweight**: 0.91 (91% of all "Overweight" instances were correctly identified).
- **Underweight**: 0.93 (93% of all "Underweight" instances were correctly identified).

## F1 Score Calculation

### Formula for F1 Score

The formula for the F1 Score for each class is:

$$\text{F1 Score}_{\text{Class}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:

- Precision: As defined earlier.
- Recall: As defined earlier.

The F1 Score is the harmonic mean of precision and recall, providing a balance between the two metrics.

### Class-Wise F1 Score Calculations

For your dataset, the F1 Score for each class can be calculated as follows:

| **Class** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|
| Normal Weight | 0.93 | 0.99 | 00.96 |
| Obese Class 1 | 1.00 | 0.17 | 0.29 |
| Obese Class 2 | 0.65 | 1.00 | 0.79 |
| Obese Class 3 | 1.00 | 0.84 | 0.91 |
| Overweight | 0.96 | 0.91 | 0.94 |
| Underweight | 1.00 | 0.93 | 0.96 |

Table 5: Class-wise F1 Score Calculations

### Complete Report Summary

| **Class** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **Support** |
|---|---|---|---|---|---|
| Normal Weight | 0.9901 | 0.93 | 0.99 | 0.96 | 101 |
| Obese Class 1 | 0.1667 | 1.00 | 0.17 | 0.29 | 6 |
| Obese Class 2 | 1.0000 | 0.65 | 1.00 | 0.79 | 13 |
| Obese Class 3 | 0.8421 | 1.00 | 0.84 | 0.91 | 19 |
| Overweight | 0.9123 | 0.96 | 0.91 | 0.94 | 57 |
| Underweight | 0.9259 | 1.00 | 0.93 | 0.96 | 27 |

Table 6: Model Evaluation: Accuracy, Precision, Recall, F1 Score, and Support

### Interpretation

The F1 Score provides a balance between precision and recall. Here's the breakdown of the F1 Score for each class:

- **Normal Weight**: 0.96 (The F1 score is high, indicating a good balance between precision and recall).
- **Obese Class 1**: 0.29 (A lower F1 score, reflecting a poor balance due to low recall despite perfect precision).
- **Obese Class 2**: 0.79 (A relatively good F1 score, with a moderate precision and perfect recall).
- **Obese Class 3**: 0.91 (A high F1 score, indicating a good balance of precision and recall).
- **Overweight**: 0.94 (A very high F1 score, indicating strong performance across both metrics).
- **Underweight**: 0.96 (A high F1 score, indicating a good balance between precision and recall).

## Confusion Matrix

| **Actual \Predicted** | **Normal Weight** | **Obese Class 1** | **Obese Class 2** | **Obese Class 3** | **Overweight** | **Underweight** |
|---|---|---|---|---|---|---|
| **Normal Weight** | 100 | 0 | 0 | 0 | 1 | 0 |
| **Obese Class 1** | 1 | 4 | 0 | 0 | 1 | 0 |
| **Obese Class 2** | 0 | 0 | 13 | 0 | 0 | 0 |
| **Obese Class 3** | 0 | 0 | 0 | 16 | 0 | 3 |
| **Overweight** | 0 | 0 | 0 | 0 | 52 | 5 |
| **Underweight** | 0 | 0 | 0 | 0 | 0 | 25 |

Table 7: Confusion Matrix for K-Nearest Neighbors Classification with imbalance

### 4.3.8 Handling Class Imbalance: SMOTE

To mitigate the impact of class imbalance in the BMI dataset, SMOTE (Synthetic Minority Oversampling Technique) was applied: SMOTE generates synthetic data points for minority classes by interpolating between existing samples. This balances the distribution of target classes in the training dataset, ensuring the model learns equally from all categories.

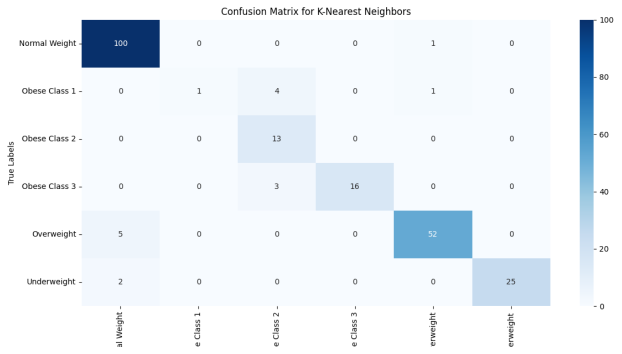


Figure 1: Confusion Matrix for K Nearest Neighbor with imbalance

## Classification Report

| **Class** | **Precision** | **Recall** | **F1-Score** | **Support** | **Accuracy** |
|---|---|---|---|---|---|
| Normal Weight | 0.98 | 0.94 | 0.96 | 101 | 0.94 |
| Obese Class 1 | 1.00 | 0.50 | 0.67 | 6 | 0.50 |
| Obese Class 2 | 0.72 | 1.00 | 0.84 | 13 | 1.00 |
| Obese Class 3 | 1.00 | 0.84 | 0.91 | 19 | 0.84 |
| Overweight | 0.93 | 0.98 | 0.96 | 57 | 0.98 |
| Underweight | 0.90 | 0.96 | 0.93 | 27 | 0.96 |

Table 8: Classification Report after removing imbalance

## Confusion Matrix for K-Nearest Neighbors

| **Actual \Predicted** | **Normal Weight** | **Obese Class 1** | **Obese Class 2** | **Obese Class 3** | **Overweight** | **Underweight** |
|---|---|---|---|---|---|---|
| **Normal Weight** | 100 | 0 | 0 | 0 | 1 | 0 |
| **Obese Class 1** | 1 | 4 | 0 | 0 | 1 | 0 |
| **Obese Class 2** | 0 | 0 | 13 | 0 | 0 | 0 |
| **Obese Class 3** | 0 | 0 | 0 | 16 | 0 | 3 |
| **Overweight** | 0 | 0 | 0 | 0 | 52 | 5 |
| **Underweight** | 0 | 0 | 0 | 0 | 0 | 25 |

Table 9: Confusion Matrix for K-Nearest Neighbors Classification after removing imbalance

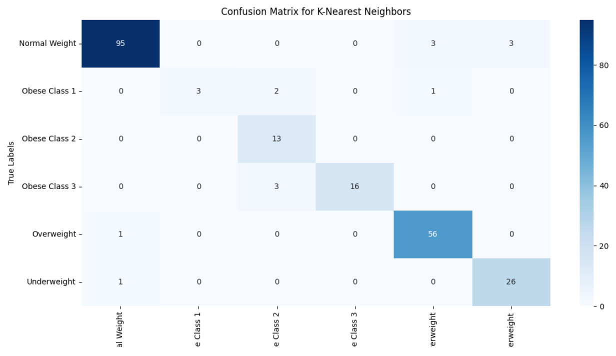


Figure 2: Confusion Matrix for K Nearest Neighbor after applying SMOTE

## 4.4 Content Based Filtering Recommendation

Content-based food recommendation system is implemented to recommend food items based on user preferences for nutrients (Calories, Fats, Proteins, Carbohydrates). The system processes a dataset of food items, each with nutritional information, and provides personalized food recommendations to users for different meal categories like Breakfast, Lunch, Dinner, Snacks, and Side Dishes based on user calories need and food allergies.

### 4.4.1 Dataset Loading

The dataset used for this recommendation system is generated using **spoonacular** API and most preferably gathered of Asian cuisine, which contains nutritional information for various food items. The dataset is loaded and processed by checking for null values and setting the `Food Items` column as the index. If any null values are found, they are handled accordingly.

### 4.4.2 Normalization

Data normalization is performed to standardize the values for each nutrient: Calories, Fats, Proteins, and Carbohydrates. This normalization ensures that each nutrient contributes equally to the cosine similarity calculation, preventing any single nutrient from dominating the recommendation process.

### 4.4.3 Meal Categorization

The dataset is divided into meal categories: Breakfast, Lunch, Dinner and Snack.

### 4.4.4 Recommendation Algorithm

- The user provides their preferred nutritional values (Calories, Fats, Proteins, Carbohydrates) and the food allergies for the recommendation system to use.
- For each meal type, the system calculates the **cosine similarity** between the user's nutritional preferences and the available food items.

### 4.4.5 Recommendation Output

The system outputs the top recommended food items for each meal type based on the **highest cosine similarity scores**. These recommended items are accompanied by their ingredients, providing users with information about the food they are considering.

### 4.4.6 Selection Process

For each meal category, the user is shown the 3 recommended dishes based on their preferences. The user selects one dish from these recommendations, ensuring that the selected dish aligns with their nutritional needs.

### 4.4.7 Nutrient Tracking

Once the user has selected dishes for each meal category, the system calculates the total nutritional values (Calories, Fats, Proteins, Carbohydrates) based on the selected dishes. The total nutrient intake is then compared with the user's original preferences to evaluate the alignment.

### 4.4.8 Nutrient Comparison

The system compares the total nutrients of the selected dishes with the user's preferences and provides feedback. It checks if the total calories exceed the user's preferred limit, and warns the user if any other nutrients (Fats, Proteins, Carbohydrates) exceed the preferred limits. The system provides warnings for any nutrient that exceeds the user's defined threshold, ensuring that the selected food items align with their dietary goals.

## 4.5 Allergens Detection

The Allergy Detection helps user in selecting food items based on their nutritional preferences while avoiding ingredients they are allergic to. The system is built using Python, leveraging libraries such as spaCy, fuzzywuzzy, and csv. This utilizes fuzzy string matching to detect and correct possible misspellings or variations in allergen names, ensuring accurate allergy filtering.

### 4.5.1 Text Processing and Normalization

The user inputs their allergies as a comma-separated list of terms. The system normalizes these terms by stripping extra spaces and converting them to lowercase.

### 4.5.2 Fuzzy Matching

The system uses fuzzy string matching to handle variations in how allergies might be spelled by users. Although this is not strictly NLP in the traditional sense (e.g., it doesn't involve syntax, semantics, or grammar), it does rely on string comparison algorithms.

#### 4.5.3 Handling Synonyms and Misspellings

The system can recognize potential misspellings and synonyms in the allergy terms provided by the user. For example, "mik" and "milk" might be matched using fuzzy logic, helping the system understand terms that are phonetically similar or commonly abbreviated.

#### 4.5.4 Text Matching (Allergy Detection)

The system attempts to match user-provided terms to a list of predefined allergens using fuzzy matching techniques. This is a form of text-based information retrieval, which is often considered a part of NLP. Although the system doesn't employ complex language models, it still processes and interprets the meaning of the user's input based on similarity to a list of known allergens.

#### 4.5.5 NLP Elements Involved

- **Tokenization**: Splitting the allergy input into individual terms (tokens) for processing.
- **Normalization**: Converting text to a standard form (e.g., lowercase) and removing extraneous spaces.
- **Fuzzy Matching**: Finding approximate matches of text based on similarity metrics, which is a technique often used in NLP for handling variations in text.

#### 4.5.6 Limitations

While the system works to some extent, it is not a fully developed solution and still has limitations. The fuzzy matching technique can sometimes produce false positives or negatives, especially if the input terms are highly ambiguous or too far off from the predefined allergens. Further improvements in handling complex synonyms, spelling variations, and context would be necessary to create a more robust allergen detection system.

# 5 System Design and Architecture

## 5.1 System Overview

The system is the first step towards the development of the solution that assist nutritionist and common man to generate the diet plan according to their preferences. The system gets user input and generate in accordance to user It starts by calculating the caloric need and then recommending food , it allow user to change their preferences at any time

## 5.2 System Architecture

### 5.2.1 Front-End

The front-end is the user-facing part of the application, built with Flutter. It is responsible for collecting user input, displaying results, and managing navigation.

- **Technology Stack:** Flutter (cross-platform for Android, iOS, Web, Desktop).
- **Responsibilities:**
  - Collect user input (e.g., height, weight, age , physical activity , gender).
  - Display visual feedback (e.g., BMI, Calories Needed along with breakdown of proteins , fats and carbohydrates food recommendations).
  - Communicate with the back-end via REST APIs.

### 5.2.2 Back-End

The back-end is responsible for business logic, data validation, and integration with databases and machine learning models.

- **Technology Stack:** Node.js, Python (Flask).
- **Responsibilities:**
  - API Gateway for communication with the front-end.
  - Data validation and processing (e.g., calculating BMI , calories and macro nutrients).
  - Integration with machine learning models for predictive analytics.

### 5.2.3 Database

The database stores user data, preferences, and potentially training data for machine learning.

- **Technology Stack:** MySQL.
- **Responsibilities:**
  - Store user information (e.g., name, age, height, weight).
  - Save application state (e.g., user progress, preferences).

### 5.2.4 Machine Learning Models

The machine learning model adds predictive or analytical capabilities to the system.

- **Technology Stack:** Python (Scikit-learn).
- **Responsibilities:**

  - Predict health metrics (e.g., BMI category) using K Nearest Neighbors.
  - Recommend food using content based filtering according to user preferences.

### 5.2.5 Data Flow Example

- The user enters personal details, height, and weight , age , gender , physical activity and food allergies through the Flutter app.
- The front-end sends the collected data to the back-end via a REST API.
- The back-end validates the data, calculates basic metrics (e.g., BMI) and store it in the datasbe.
- The machine learning model processes the data stored in database fetched via API and returns a prediction (e.g., BMI category , meal plan recommendation).
- User data and results are saved in the database for future use.
- The back-end sends the processed results back to the front-end, which displays them to the user.

## 5.3 Database Design

For the app the database is designed to store the suer information so that user don't have to input its preference every time . he just login and get the meal for the day in one click and can update only one the thing he needs to. It also have the facility to store user meals to keep track of its daily diet and calories needed.

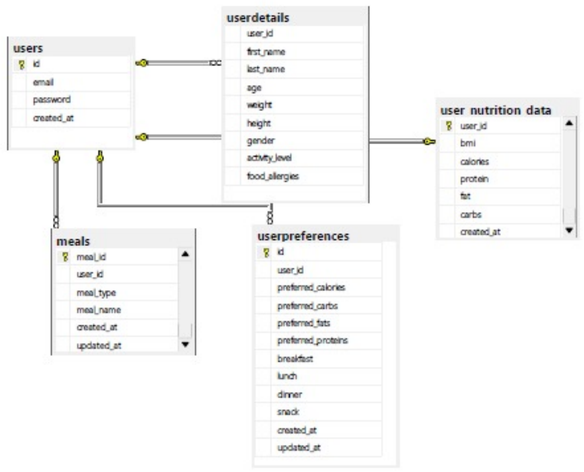


Figure 3: Database Design for the Ai based Diet Recommendation system

## 5.4 User Interface Design

### Explore Now

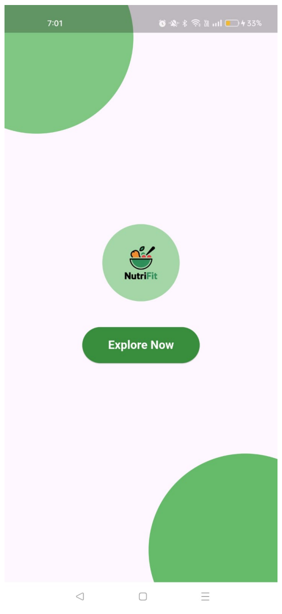


Figure 4: Welcome page allowing users to explore the app features.

### Sign Up Page

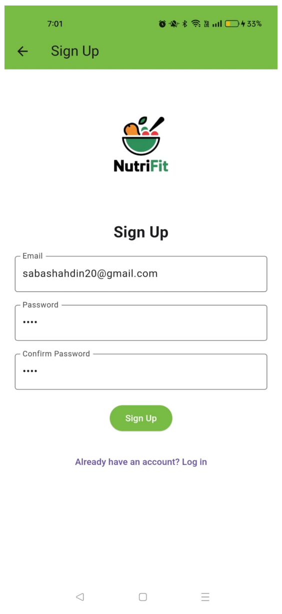
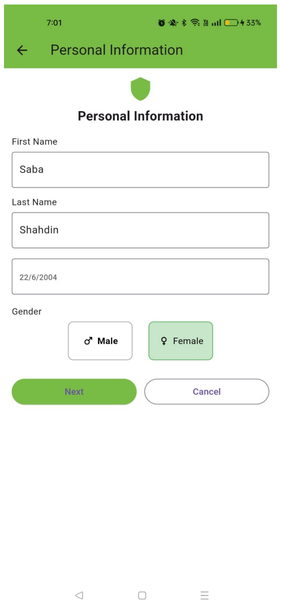


Figure 5: Sign up page for creating a new account if the user doesn't already have one.

## Personal Information Page


7:01

Personal Information

Personal Information

First Name

Saba

Last Name

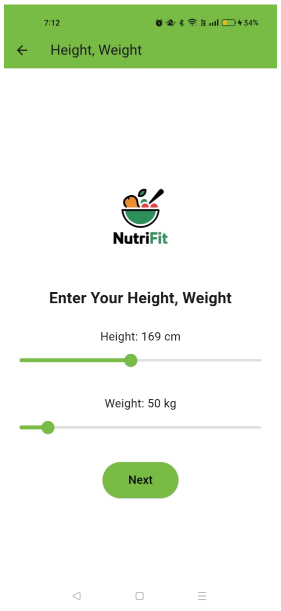
Shahdin

22/6/2004

Gender

Figure 6: Form to collect personal information such as name, age, and gender.

## Height and Weight Page


7:12

Height, Weight

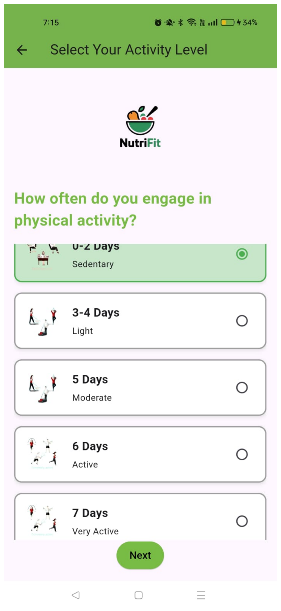
Enter Your Height, Weight

Height: 169 cm

Weight: 50 kg

Figure 7: User input form for entering height and weight to calculate body metrics.

## Activity Level Selection


7:15

Select Your Activity Level

NutriFit

How often do you engage in physical activity?

[✓ 0-2 Days Sedentary]

3-4 Days Light

5 Days Moderate

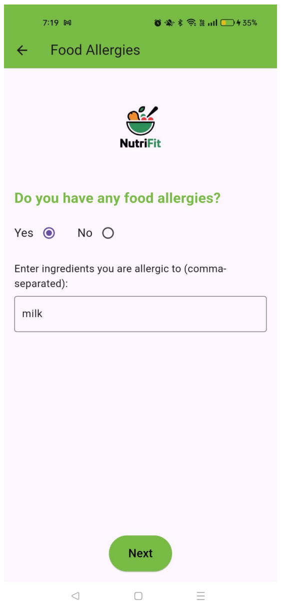
6 Days Active

7 Days Very Active

Next

Figure 8: Page to select the user's daily activity level for more accurate recommendations.

## Allergy Information Page


7:19

Food Allergies

NutriFit

Do you have any food allergies?

[✓ Yes] No

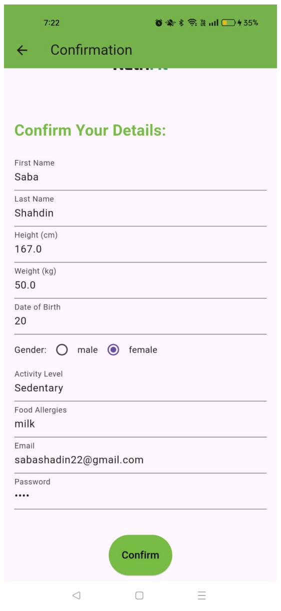
Enter ingredients you are allergic to (comma-separated):

milk

Next

Figure 9: Field to input any food allergies to customize diet recommendations.

## Details Confirmation Page


Confirmation

Confirm Your Details:

First Name
Saba

Last Name
Shahdin

Height (cm)
167.0

Weight (kg)
50.0

Date of Birth
20

Gender: male [✓ female]

Activity Level
Sedentary

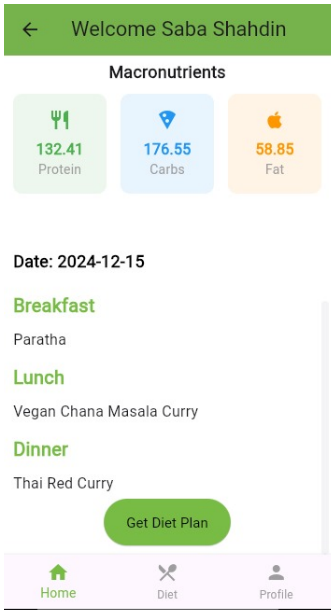
Food Allergies
milk

Email
sabashadin22@gmail.com

Password
••••

Figure 10: Summary of user-provided details for confirmation before proceeding.

## Home Page


Welcome Saba Shahdin

Macronutrients

132.41 Protein

176.55 Carbs

58.85 Fat

Date: 2024-12-15

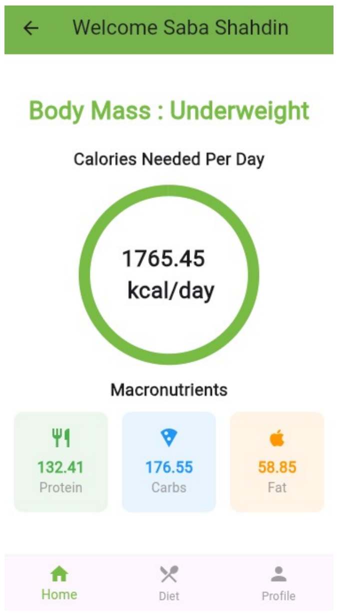
Breakfast
Paratha

Lunch
Vegan Chana Masala Curry

Dinner
Thai Red Curry

Figure 11: Home page displaying an overview of features and quick access options.

## Body Mass Classification



Welcome Saba Shahdin

Body Mass : Underweight

Calories Needed Per Day

1765.45 kcal/day

Macronutrients

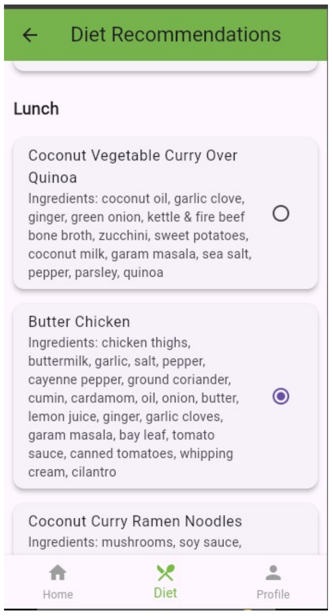
132.41 Protein

176.55 Carbs

58.85 Fat

Figure 12: Page displaying calculated body mass index (BMI) classification.

## Diet Recommendation Page



Diet Recommendations

Lunch

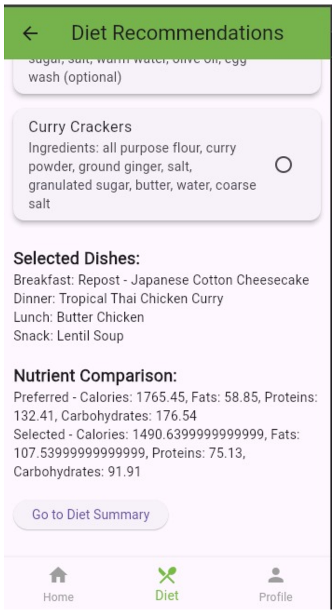
Coconut Vegetable Curry Over Quinoa
Ingredients: coconut oil, garlic clove, ginger, green onion, kettle & fire beef bone broth, zucchini, sweet potatoes, coconut milk, garam masala, sea salt, pepper, parsley, quinoa ○

Butter Chicken
Ingredients: chicken thighs, buttermilk, garlic, salt, pepper, cayenne pepper, ground coriander, cumin, cardamom, oil, onion, butter, lemon juice, ginger, garlic cloves, garam masala, bay leaf, tomato sauce, canned tomatoes, whipping cream, cilantro [✓]

Coconut Curry Ramen Noodles
Ingredients: mushrooms, soy sauce,

Figure 13: Personalized diet recommendations based on user data and preferences.

### Nutrient Comparison Page



sugar, salt, warm water, olive oil, egg wash (optional)

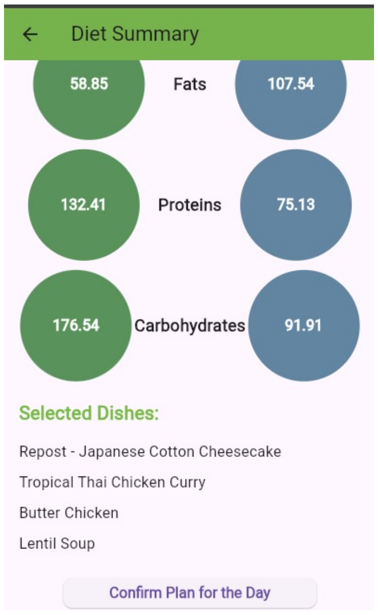
Curry Crackers
Ingredients: all purpose flour, curry powder, ground ginger, salt, granulated sugar, butter, water, coarse salt

Selected Dishes:
Breakfast: Repost - Japanese Cotton Cheesecake
Dinner: Tropical Thai Chicken Curry
Lunch: Butter Chicken
Snack: Lentil Soup

Nutrient Comparison:
Preferred - Calories: 1765.45, Fats: 58.85, Proteins: 132.41, Carbohydrates: 176.54
Selected - Calories: 1490.6399999999999, Fats: 107.53999999999999, Proteins: 75.13, Carbohydrates: 91.91

Figure 14: Comparison of nutrients in selected dishes for informed decision-making.

### Diet Summary Page



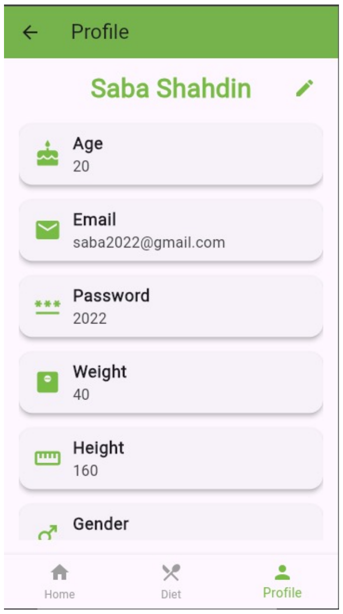
| | | |
|---|---|---|
| 58.85 | Fats | 107.54 |
| 132.41 | Proteins | 75.13 |
| 176.54 | Carbohydrates | 91.91 |

Selected Dishes:

Repost - Japanese Cotton Cheesecake
Tropical Thai Chicken Curry
Butter Chicken
Lentil Soup

Figure 15: Detailed summary of the recommended diet plan, including meal timings.

### User's Profile Overview



Saba Shahdin

Age
20

Email
saba2022@gmail.com

Password
2022

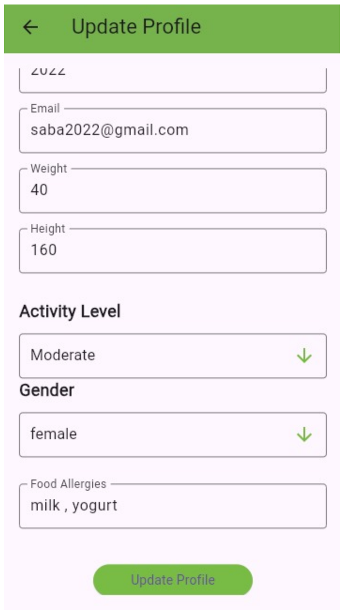
Weight
40

Height
160

Gender

Figure 16: Overview of the user's profile, displaying personal details.

### Edit User Profile



2022

Email
saba2022@gmail.com

Weight
40

Height
160

Activity Level
Moderate

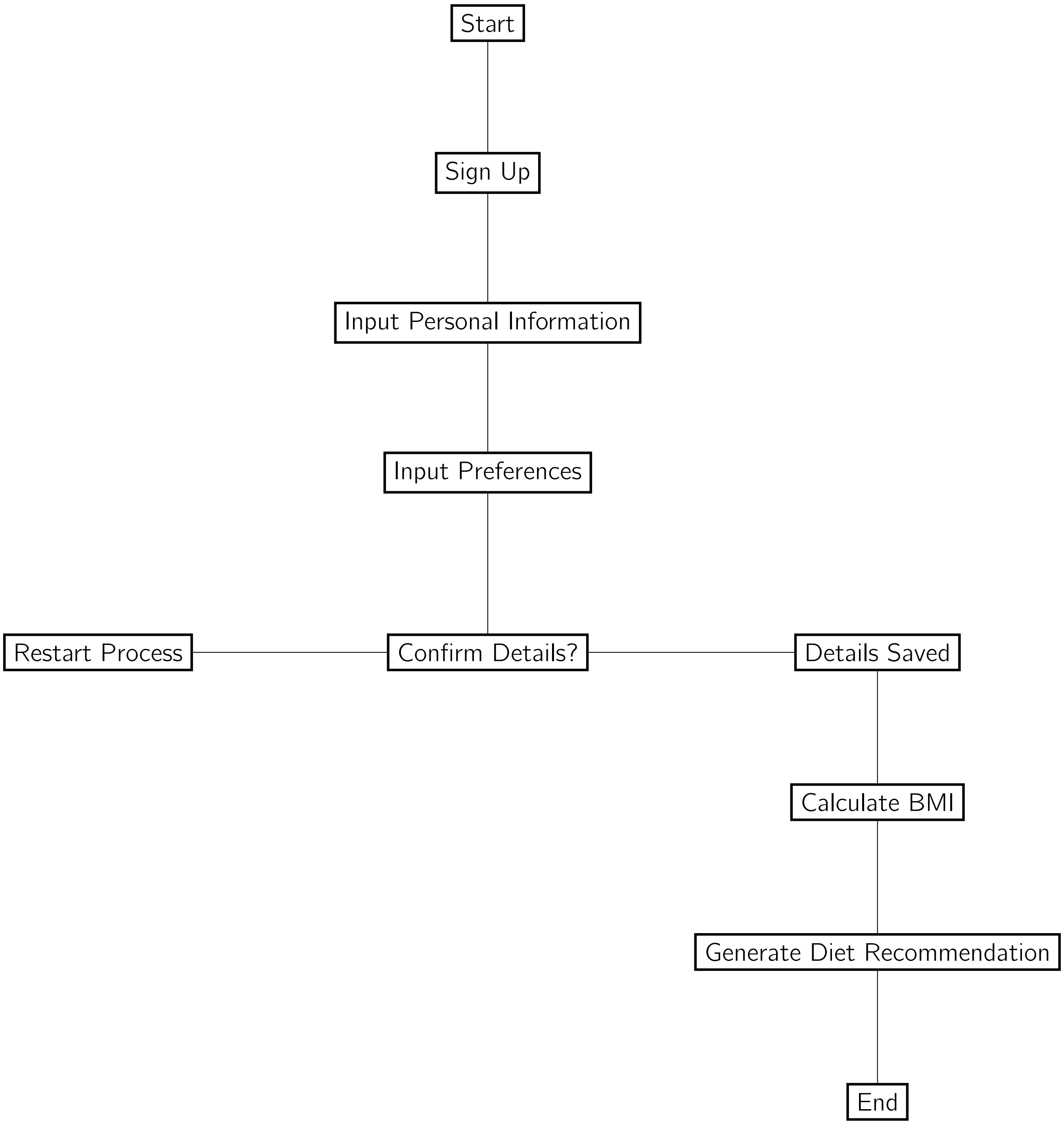
Gender
female

Food Allergies
milk , yogurt

Figure 17: Interface for editing user details.

## 5.5 User Flow

The following diagram illustrates the step-by-step interaction of user with diet recommendation system, beginning from registration process to receiving personalized diet recommendations.



# 6 Implementation

## 6.1 User Data

The system collects essential user details to provide personalized recommendations, including:

- **Personal Information**: Age, weight, height, and gender.
- **Physical Activity Level**: The system considers the user's physical activity level, categorized as active, sedentary, or light.

Using the collected data, the system performs the following steps:

- **Body Mass Index (BMI) Calculation**: The BMI is calculated using the standard formula based on the user's weight and height.
- **Calorie and Macronutrient Calculation**: The system calculates the daily calorie needs and macronutrients (carbohydrates, protein, fats) based on the user's input and activity level.
- **Machine Learning Model for Body Mass Prediction**: A machine learning model is employed to predict the user's body mass and potential health risks.

## 6.2 Diet Recommendation

The system provides diet suggestions for the day, covering all four meal types: breakfast, lunch, dinner, and a snack.

- For each meal, the system suggests three options. The user can choose one option per meal type.
- **Allergen-Free Recommendations**: The system takes into account any allergens specified by the user and removes ingredients that the user is allergic to.
- After the user selects their meal options for the day, the system calculates the total nutrient values for the day, including calories, protein, fats, and carbohydrates.

## 6.3 Meal Customization

Users have the flexibility to update their meal choices for the day based on preferences or new dietary requirements.

- The system allows users to set and update preferences for future meal suggestions, ensuring that the recommendations align with their dietary goals and restrictions.

## 7 Challenges Faced

The major challenge in developing this system is the collection f data of the food in the Asia and their nutrient levels . Te other thing is the combination of artificial intelligence e along with recommendation approach to give more defined results. We face challenge in reading and decrypting the allergies written in human language comma separated on which we still working but until no it is working to the extent which aligns to our sytem.

## 8 Conclusion

### 8.1 Summary

In this project, a personalized diet recommendation system was developed to help users manage their nutritional needs based on their physical characteristics, activity levels, and dietary preferences. The system collects essential user data such as age, weight, height, gender, and activity level, and calculates BMI, daily calorie requirements, and macronutrient distribution. Using these inputs, the system suggests meal options for the day, taking into account user preferences and allergen restrictions. A machine learning model was integrated to predict the user's body mass and the system allows users to customize their meal plans for flexibility and long-term use. The system's functionality was evaluated through rigorous testing, ensuring its accuracy in calorie and nutrient calculation, as well as user satisfaction in meal recommendations.

### 8.2 Key Findings

Throughout the development process, key findings included the importance of personalization for user engagement, as allowing users to customize their diet plans led to higher satisfaction. Accurate nutrient calculation built trust by providing reliable calorie and macro nutrient information. The allergen-free recommendations made the system more accessible to users with food sensitivities. The machine learning model provided valuable health insights, though its accuracy can improve with more data. Lastly, the system's flexibility in updating meal plans and preferences ensured long-term usage and adaptability to users' evolving needs.

### 8.3 Future Work

Though this symmetry is developed to assist the nutritionist and suer in gaining the health diet in their daily life but this is our first step in this domain. This app only recommend food on user physical conditions and preferences of food but in future we plan to incorporate the health metrics to give more professional diet plan for the user.Moreover we used advanced Natural Language technique to develop a Chat bot which helps user to gain more efficient diet plan according to their need.