Academia International College

Tribhuvan University

Institute of Science and Technology

Final-Defense Report On

"Diet Recommendation System – Smart Diet"

Submitted to

Institute of Science and Technology, Tribhuvan University

In Partial Fulfillment of the Requirement for the Bachelor Degree in Computer Science and Information Technology

Submitted by

Ashmita Thapa (20906/075)

Neha Lama Tamang (20913/075)

Rinu Maharjan (20919/075)

On The Supervision of

Ghanashyam BK

10th June, 2023

ACKNOWLEDGEMENT

We owe our most profound appreciation to the efforts of Academia International College for providing us the necessary information and infrastructure regarding the project and for creating friendly environment for the successful completion of the project. It gives us immense pleasure to express our deepest sense of gratitude and thanks to our respected and esteemed supervisor **Mr. Ghanshyam BK** who gave us guidance and constant supervision in achieving goal as well as his encouragement to maintain our progress in track. We are highly indebted to our esteemed project coordinator **Mrs. Alka Anny Jha** for her constant support, guidance, valuable suggestions and constructive comments. Our thanks and appreciations also go to our colleagues for the help and inspiration during the project and people are willingly helped us with their abilities.

Thanking You,

Ashmita Thapa (20906/075)

Neha Lama Tamang (20913/075)

Rinu Maharjan (20919/075)

ABSTRACT

In the present context, access to a balanced diet is a major problem in many parts of the world. With unhealthy choices of processed food and less nutritious foods like fast food, junk food has led to many health problems. People are forgetting the importance of balanced diet. “Smart Diet” provides a digital platform for personalized and effective diet plan that can help to improve the nutritional status and health of people. It helps users to adopt healthy behaviors, such as eating a balanced diet, which can help improve overall health and achieve specific health goals. It creates a personalized diet plan by taking into account the user's age, gender, activity, nutritional values and no. of meals, as well as their preferences. “Smart Diet” provides users with a convenient and accessible resource for managing their diet and potentially saving their time and effort in seeking out information on their own.

Keywords: *Diet, User, BMI, Calories, Nutrition, Personalized, Data, Python, Streamlit, FastAPI, Bootstrap, CSS, HTML*

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Acronyms	Abbreviations
BMI	Body Mass Index
BMR	Basal Metabolic Rate
DFD	Data Flow Diagram
DSQ	Dietary Screener Questionnaire
KNN	K-Nearest Neighbor
RDEE	Resting Daily Energy Expenditure
SOTA	State of the Art
SI	System of Units
SRS	Software Requirements Specification
UI	User Interface
USC	Customary System
WBS	Work Breakdown Structure
WHO	World Health Organization

1. Introduction

1.1. Introduction

"Smart Diet" is a web application developed to help people lead healthier lifestyles. It provides comprehensive information about each meal and allows users to personalize their own diet plan, all presented through an easy-to-use interface. People who are concerned with maintaining their general health and fitness are the target audience for the application. When designing the plan, a variety of elements are taken into account, including age, gender, physical activity, meals per day and nutritional values.

For the body to receive all the nutrients it needs to function at its best, a well-balanced diet is essential. Applications that recommend foods based on content are very helpful for promoting a healthy diet. To provide individualized recommendations, these applications make use of nutrition and food ingredient data. One benefit of content-based strategies is that they take into account each person's dietary needs and preferences when offering suggestions. The personalized recommendations can improve users' overall health and generate recipes in addition to assisting them in making better food decisions. Smart Diet is design for revolutionized your cooking experience and inspire taste buds. We understand that everyone has different preferences when it comes to their meals, whether it is breakfast, lunch, dinner, or even snacks. Smart Diet allows you to select your desired meal type and generate according to match you are craving at any time.

1.2. Problem Statement

People all over the world are becoming more concerned with their health and way of life in the current global context. Exercise and avoiding greasy, spicy, and junk food alone are insufficient; a balanced diet is just as crucial. A balanced diet may help one live a healthy life depending on variables like size, weight, and age. An active lifestyle and a healthy diet can reduce the risk of chronic diseases. A state-of-the-art (SOTA) project on food and diet recommendations led to the notion of developing a content-based recommendation system based on machine learning.

The goal of this project is to create a sustainable, adaptable diet plan that takes each user's unique needs and preferences into account. The diet program will be simple to understand and adhere to, with options for suggestions based on variables like BMI, calorie intake, nutritional values, and ingredients. The result will be a thorough diet program that is customized to the user's particular requirements and preferences and that can be easily incorporated into daily life.

1.3. Objectives

The project aims to provide a user-centered and efficient diet plan that improves people's health, lowers the health burden associated with poor diets, and encourages a healthier lifestyle. Main objectives of a Smart Diet consist of:

- Achieving specific health goal
- Creating an app that recommends tailored diet.

1.4. Scope

The "Smart Diet" application's range of coverage includes the following:

- Offer personalized diet plans that take into account each person's needs.
- Offer ideas for nutritious meals to eat every day.
- Take into account a variety of elements, including nutrient content, ingredients, and physical activity.

1.5. Development Methodology

Because of the project's adoption of the Incremental model and the sizeable amount of data and information it contains, the Smart Diet application is deemed appropriate for it. A systematic validation of the system and evaluation of its development are made possible by the incremental approach. The project is divided into the following stages:

i. **First increment:** In this stage, we brainstormed and outlined regarding the topic of our project and how our project will work.

ii. **Second increment:** Information is gathered at this stage about the person's physical characteristics, including height, weight, age, gender, levels of physical activity, and necessary dietary nutritional values. Additionally, information is gathered to help the recommendation application be trained.

iii. **Third increment:** At this stage, the data model is trained along with the design and development of the application's frontend and backend.

iv. **Fourth increment:** The objective of this stage is to evaluate the application's capacity to suggest diet meal plans based on the supplied data and deploy application in docker.

1.6. Report Organization

The project documentation is divided into chapters that contain the following detailed information:

- Chapter 1 introduces the project by describing its history, problem statement, goals, scope, and constraints.
- A background investigation and literature review make up Chapter 2.
- The topics of requirement analysis and feasibility study are covered in Chapter 3, along with system requirements, a project overview, and the employed algorithms.
- Chapter 4 examines system design, including data dictionary and process design.
- The system's implementation and testing are the main topics of Chapter 5, which also covers the methodology, platforms, development tools, implementation of the algorithm, and system testing as a whole.
- Chapter 6 wraps up the project and provides suggestions.

2. Background Study & Literature Review

2.1. Background Study

Based on user needs research and current dietitian consultation flows, the Diet Recommendation System was created. It facilitates meal planning, grocery shopping, and cooking routines and is available as a web-based application as well as Android and iOS versions. For personalized meal suggestions, users can choose foods from their refrigerator or pantry. The tracking of and adherence to nutritional objectives is made simpler by these applications. Users can create a meal plan quickly and easily using tools like a calorie counter, weekly meal plans, and grocery lists, helping them to meet their dietary and nutritional objectives.

2.1.1. Implemented Algorithm

2.1.1.1. Content Based Filtering

Content Based Recommendation methods have several advantages than other methods. First, provide highly personalized recommendations to individual users based on their preferences and behavior. Additionally, recommendations are transparent to users, making it easy for them to understand why a particular item is recommended.

2.1.1.2. Model Development

To develop our recommender, we used the **Nearest Neighbor algorithm**, an unsupervised learning algorithm capable of neighbor searching. The algorithm provides a unified interface to three different nearest neighbor algorithms: BallTree, KDTree, and a brute force algorithm based on routines in the sklearn.metrics.pairwise module. In our recommendation program, we specifically use the brute force algorithm with cosine similarity because it is efficient on small datasets.

cos(theta) = (A * B) / (||A|| * ||B||)

The cosine similarity formula is used to calculate the similarity of items based on their nutritional value and composition. This formula compares the dot product of two elements to the product of their magnitude, yielding a value between -1 and 1. [1]

Nearest Neighbor Algorithm

Step1: Data loading

Step2: Set k's value to zero.

Step3: Repeat from 1 to all of the training data points in order to obtain the predicted class.

Step4: The separation between each row of training data and the test data should be calculated. Since it is the most often used technique, we will utilize Cosine as our distance metric in this case.

Step5: Based on the calculated distances' values, rank the distances in ascending order.

Step6: the first k rows of the sorted array.

Step7: Identify the row's most prevalent class.

Step8: Send the anticipated class back. [1]

Body Mass Index (BMI)

A common method for determining whether a person is a healthy weight for their height is to use the Body Mass Index (BMI). But it's important to remember that the BMI is not a perfect indicator and shouldn't be used as the only factor in determining someone's health status. Along with BMI, other elements like muscle mass, body shape, and fat distribution can have an effect on health. [2]

The equations used for calculating BMI in the International System of Units (SI) .

SI/ Metric UNITS: $BMI = \frac{mass\ (kg)}{height^2\ (m)}$

Calorie Calculator

To estimate a person's daily calorie needs, the calorie calculator applies various equations. In order to improve its accuracy, the Harris-Benedict Equation—which was originally created to calculate basal metabolic rate (BMR)—was revised in 1984. The Mifflin-St Jeor Equation, which was introduced in 1990, is thought to be more accurate than the updated Harris-Benedict Equation for calculating BMR. Lean body mass is taken into account in the Katch-McArdle Formula when determining resting daily energy expenditure (RDEE). The Katch-McArdle Formula may offer greater precision for people with lower body fat percentages who are aware of their body fat percentage, even though the Mifflin-St Jeor Equation is typically thought to be the most accurate for estimating BMR. The Calorie Calculator combines all three equations to produce an estimated daily caloric requirement average.

Mifflin-St Jeor Equation:

- For men: BMR = 10W + 6.25H - 5A + 5
- For women: BMR = 10W + 6.25H - 5A – 161 [3]

Macro Calculator

Users can get a thorough week-by-week diet plan by providing key details like their current level of exercise or ideal weight. Several variables, such as height, weight, age, gender, and level of physical activity, affect how many calories should be consumed per meal. Given that they affect the number of calories required to maintain a given weight, these variables must be taken into account when determining the recommended daily calorie intake. Generally speaking, three to five meals should be eaten each day. Once the daily calorie intake is known, it can be divided into three, four, or five meals using a meal calculator. [4]

2.2. Literature Review

The use of web-based applications for diet planning is the focus of this literature review. There is a dearth of research on the effectiveness of mobile apps in promoting weight loss and healthy eating habits, despite the fact that they have become a popular tool for people

to track their diet and exercise. This review of the literature's existing research on the use of web-based diet planning applications aims to evaluate that research and to find any gaps that the proposed diet planner app might fill.

Several studies have looked into the use of mobile apps for tracking and planning diets. O'Donovan and colleagues conducted a thorough review. (2015) discovered that mobile apps can effectively encourage weight loss and wholesome eating. [5] Chen et al.'s systematic review of another study. (2017) discovered that mobile apps that offer tailored support and feedback are more successful at encouraging weight loss and healthy eating practices. [6]

Nevertheless, some studies have produced contradictory results. "Neve et al.". (2019) conducted research showing that weight loss can be aided by using mobile applications that provide calorie counting and food tracking features. However, their research also showed that apps that offered recipe and meal planning suggestions did not produce the same outcomes. [7]. In addition, Kim et al. Those with a history of disordered eating were not found to benefit from mobile applications in 2018, according to research. The dietary screener questionnaire (DSQ) and body mass index (BMI) will be used to measure changes in eating habits and evaluate weight loss, which will be the main focus of the study. [8]

According to the literature review, it was found that a web-based application can be useful for a comprehensive nutrition plan. However, little is still known about the extent to which web-based apps are effective in promoting healthy eating habits. Further research is needed to determine the ideal functionality and design for web applications that promote healthy eating habits. This will help fill existing gaps in the literature and provide a more thorough understanding of the topic.

2.3. Study of Existing Systems

“Diet Pal” was developed as a web-based system to expand its reach, especially in rural areas. The primary scripting language used by DietPal is Active Server Pages (ASP) along with other scripting languages such as VBScript and JavaScript. During the development, the lessons learned from the research of user needs and the practice of consulting

nutritionists were taken into account. There are two ways to automatically calculate the energy needs of each patient.

- Predictive equation to estimate energy requirement using the following formula:
  Energy requirement = BMR × stress factor × activity factor
- Quick method, based on the following formula:
  Energy requirement = weight (kg) × quick method factor (kcal/kg). [9]

"Mealime" is an application available on the web, Android and iOS platforms. Mealime's core concept is about simplifying meal planning, grocery shopping and cooking routines. It offers a variety of customization options that allow users to choose from 8 diet types (such as ketogenic or pescatarian), eliminate 12 allergies, exclude 124 usually odd ingredients, and apply custom filters to customize the menu exactly to their liking. The app offers smart recipes that are both nutritious and delicious, created by a team of skilled chefs, making cooking quick and easy. In addition, Mealime uses grocery information to optimize recipes, reduce food waste and deliver convenient meal kits without high costs. [10]

"**Eat This Much**" generates a personalized meal plan based on a person's food preferences, budget and schedule. This web app aims to simplify the decision-making process about what to eat, which can often be overcomplicated by rich, imaginary diets. Users can register and provide detailed information about themselves, their food preferences and dietary goals. Eat This Much generates a personalized meal plan that allows you to rearrange foods and meals according to a person's appetite. The app even offers recipes with step-by-step directions and ingredient lists. The entire process is automated, giving users the flexibility to add items that may not have originally been on the menu. [11]

3. System Analysis

3.1. Requirement Analysis

Finding and comprehending the tasks required to satisfy the requirements or conditions of a new or modified product or project is the focus of requirement analysis. Analysis, documentation, validation, and management of the software or system requirements are all steps in this process, which takes into account the potentially conflicting requirements of various stakeholders.

3.1.1. Functional Requirements

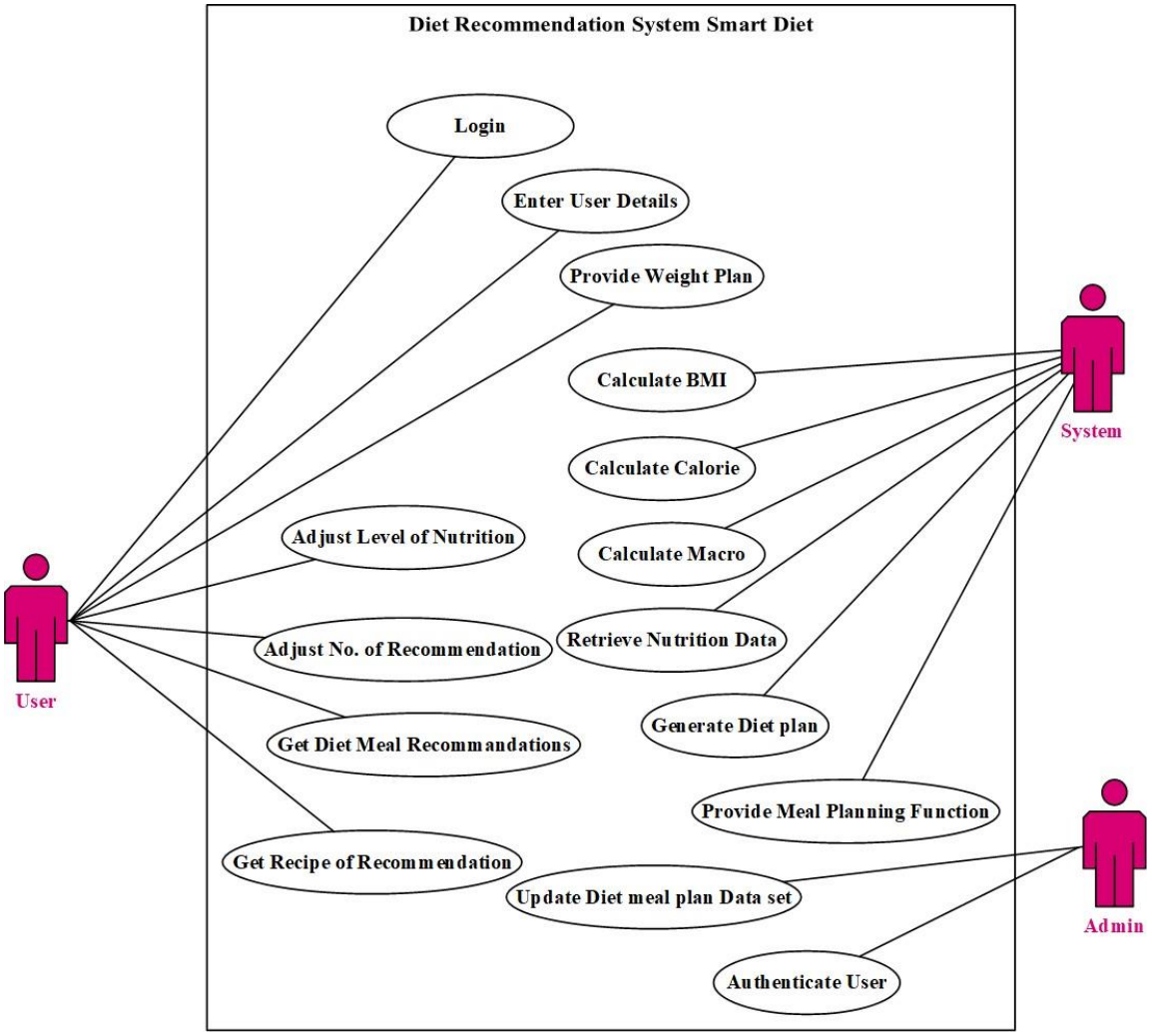


Figure 3. 1: Use Case diagram of Smart Diet- Diet Recommendation system

Functional requirements for a diet recommendation system would describe the specific characteristics and capabilities that the system needs to perform its functions effectively. The functional requirements for a diet recommendation system are:

i. **User login:-** The system should allow users to log in to the system using a unique username and password.
ii. **Personalization:-** The system should be able to learn about the individual interests and preferences of each user based on their past reading history and the news articles they have liked or shared.
iii. **Recommendation generation:-** The system should be able to generate personalized recommendations for each user based on their interests and dietary preferences. The recommendations should be relevant and diverse.
iv. **Diet categorization:-** The system should be able to categorize diet based on their recipes nutritional characteristics matter, to improve the relevance of the recommendations.

3.1.2. Non-Functional Requirements

i. **Performance:** The system should be able to generate diet plans in real-time without experiencing significant delays, and it should be capable of handling a sizable dataset that is continuously being updated.
ii. **Reliability:** The system should be dependable and consistently accessible to users, ensuring that it runs consistently without frequent interruptions or downtime.
iii. **Security:** The system should place a high priority on user privacy and data security, putting in place safeguards to thwart hacking attempts and other security risks.
iv. **Usability:** The program ought to have a user-friendly interface that is simple to operate. To ensure a positive user experience, it should be simple and intuitive for users to interact with.
v. **Maintainability:** The system ought to be made easy to maintain and update in the future. It should have clear documentation and a modular design, which makes it easier to change or expand its functionality as needed.

3.2. Feasibility Analysis

A feasibility study examines the likelihood of a project's success while taking operational, technological, and other pertinent aspects into account. A feasibility study enables project managers to examine both the potential risks and benefits before devoting a significant amount of time and resources to a project, as opposed to moving forward with it blindly and hoping for positive results.

3.2.1. Technical Feasibility

The technical viability of Smart Diet refers to how well the system or technology can be used and supported in a particular setting. This project's specific objectives and requirements will determine whether it is feasible or not. The project's viability will also depend on the availability of data and resources.

Hardware Requirements (Minimum)

- Processor: Intel Pentium IV 2.0 GHz and above
- RAM: 512 MB and above
- Hard disk: 80 GB and above
- Monitor: CTR or LED or LCD
- Keyboard: Normal or Multimedia

Software Requirements

- Design: HTML, CSS, Bootstrap
- Language: Python
- Frontend: Streamlit
- Back End: FastAPI
- Deployment: Docker
- Operating System: Windows 8, Windows 10 or Windows 11

- Browser: Google Chrome, Microsoft Edge, Mozilla Firefox, etc.

In general, this project requires the gathering of food-related data, tools for analyzing and algorithms for formulating personalized meal plans. To complete these components, a variety of technologies, including machine learning, web development, can be used.

3.2.2. Operational Feasibility

How well a proposed project fits with the organization's overarching goals and operations is referred to as operational feasibility. The operational viability of Smart Diet depends on a number of variables, such as the accessibility of resources and support, the product's market demand, and the project's long-term viability.

i. **Resources and personnel:** The project team must determine the resources and personnel required to operate the system and ensure that it has enough support to function.
ii. **User acceptance testing:** It is essential for ensuring that the system is user-friendly and satisfies user needs.
iii. **Maintenance and support:** It's important to set up a plan for keeping the system up and running, including regular backups and troubleshooting. Users should receive assistance from adequate technical support.

3.2.3. Economic Feasibility

The system's overall costs must be evaluated in order to determine its economic viability. The viability of a technology or system from a financial standpoint is determined by an economic feasibility analysis. The costs and benefits associated with development, operation, and maintenance can be used to assess the economic viability of a diet recommendation web application. Our project is fairly simple, and neither its development nor upkeep are expensive. There are several ways the web application can make money:

i. **Offering a freemium model:** Basic features are accessible for free and advanced features require a subscription fee.

ii. **Advertising:** Making money through advertisements by letting advertisers place their ads inside the application and getting paid for ad clicks or views.

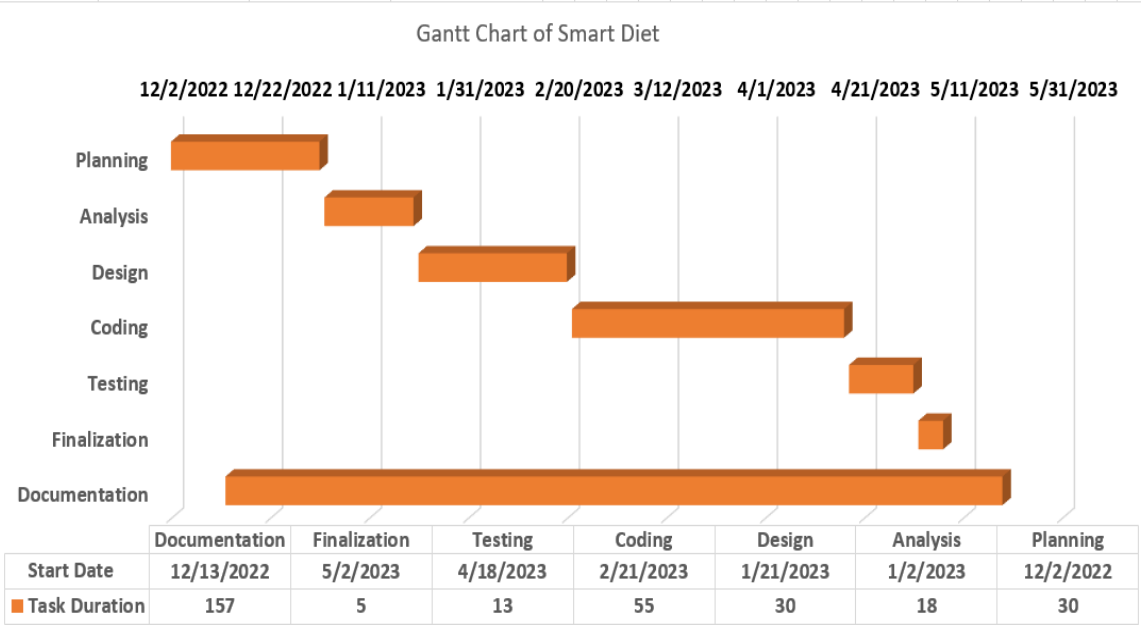
iii. **Affiliate marketing:** Working with food and nutrition businesses to promote their goods in exchange for a cut of any sales generated by the app.

iv. **Collaboration with sponsors:** who are eager to promote wholesome eating practices and who are willing to pay to have their goods or services advertised inside the application.

3.2.4. Data Collection

Any project endeavor needs to include a data collection phase. To gather thorough information, we meticulously planned and allotted resources, established project requirements, performed a literature review, and created a schedule. Online research papers provided us with pertinent information for this study. In order to gather the necessary data, we reviewed previous research on online diet recommendation resources and looked into related projects. As a result, Kaggle provided us with a dataset. Data with over 500,000 recipes and 1,400,000 reviews from food.com.

3.2.5. Schedule



	Documentation	Finalization	Testing	Coding	Design	Analysis	Planning
Start Date	12/13/2022	5/2/2023	4/18/2023	2/21/2023	1/21/2023	1/2/2023	12/2/2022
Task Duration	157	5	13	55	30	18	30

Figure 3. 2: Gantt Chart of Project Schedule

3.3. Work Breakdown Structure

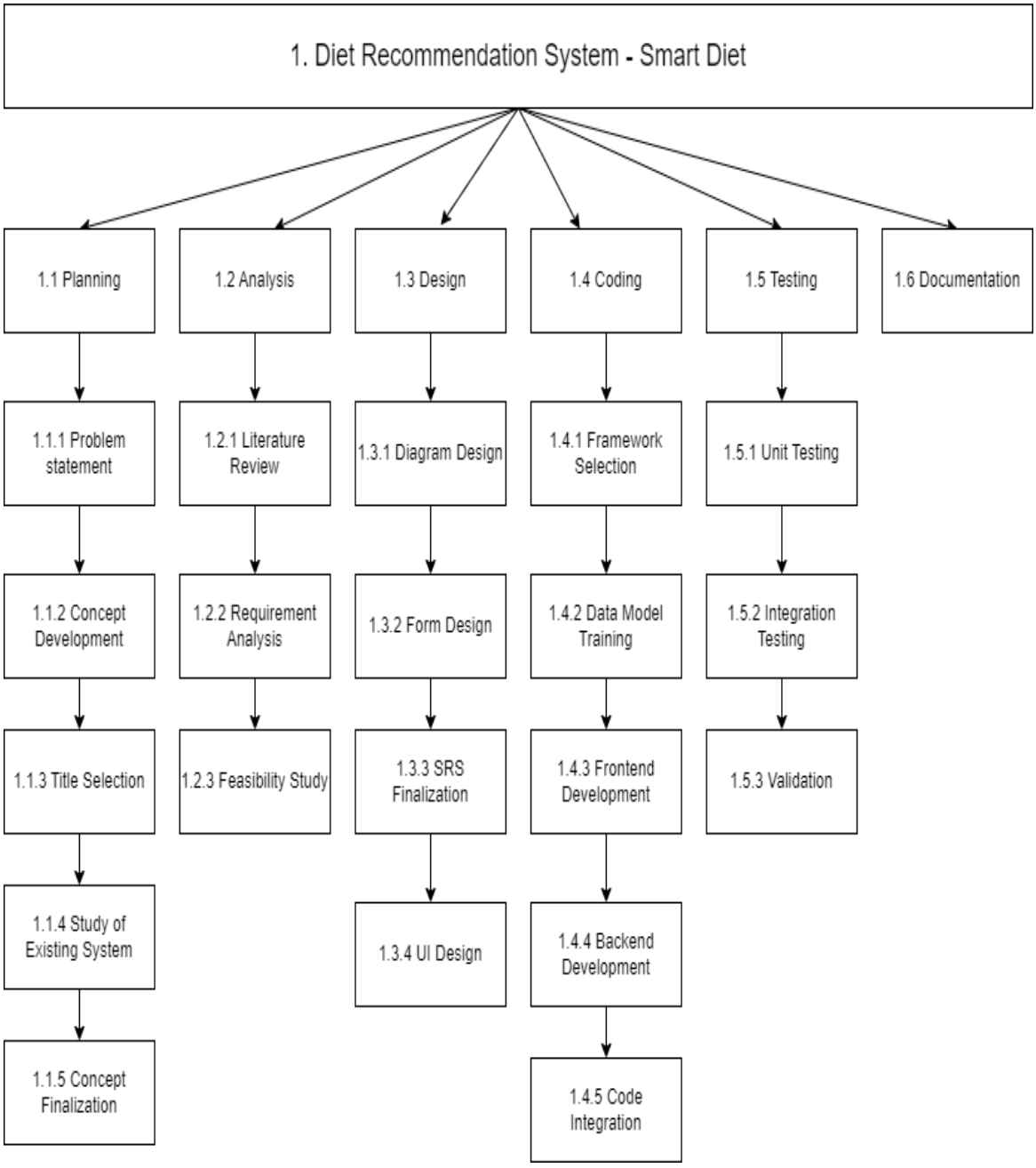


Figure 3. 3: WBS of Smart Diet - Diet Recommendation System

3.4. DFD Diagram

3.4.1. Context Diagram (DFD level 0 diagram)

The highest-level diagram that shows the whole system as a single process is a level 0 DFD (Data Flow Diagram). Without diving into the specifics of internal procedures, it demonstrates how data enters and exits the system.

In this DFD, we have following entity:

User: It is responsible for providing user details and engaging with the recommended diet plan.

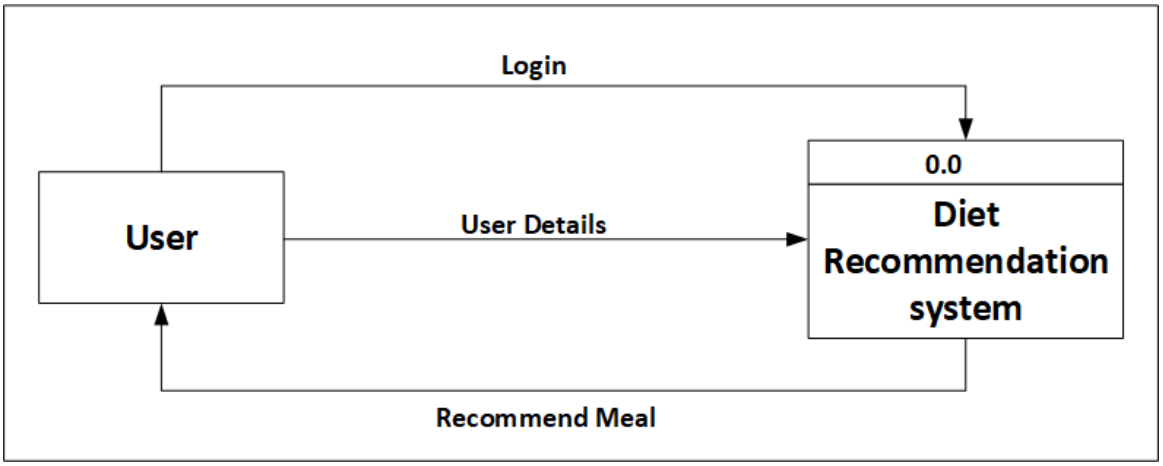


Figure 3. 4: Context diagram of Smart Diet - Diet Recommendation System

3.4.2. DFD level 1 diagram

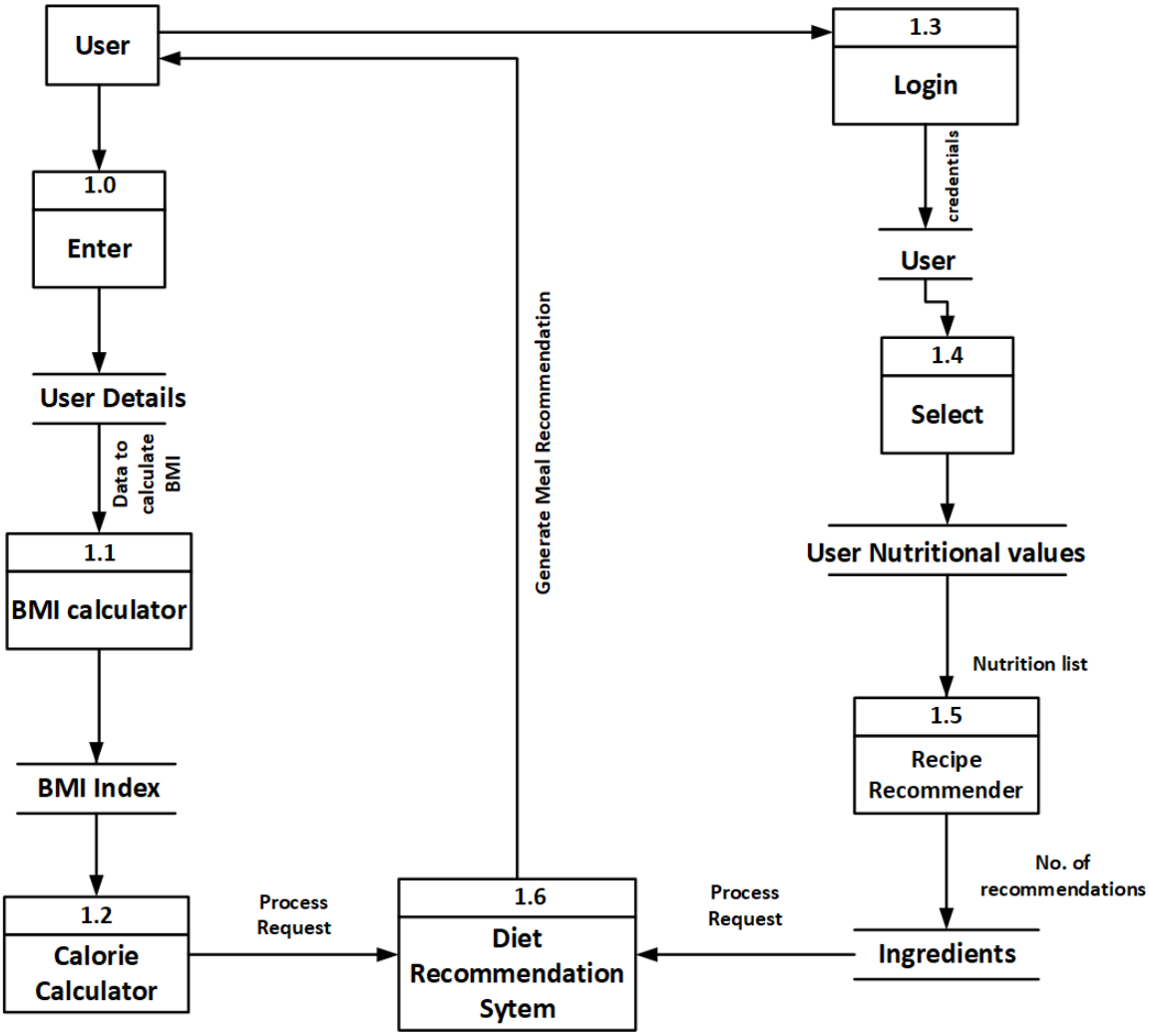


Figure 3. 5: DFD Level 1 diagram of Smart Diet - Diet Recommendation System

3.4.3. DFD level 2 diagram

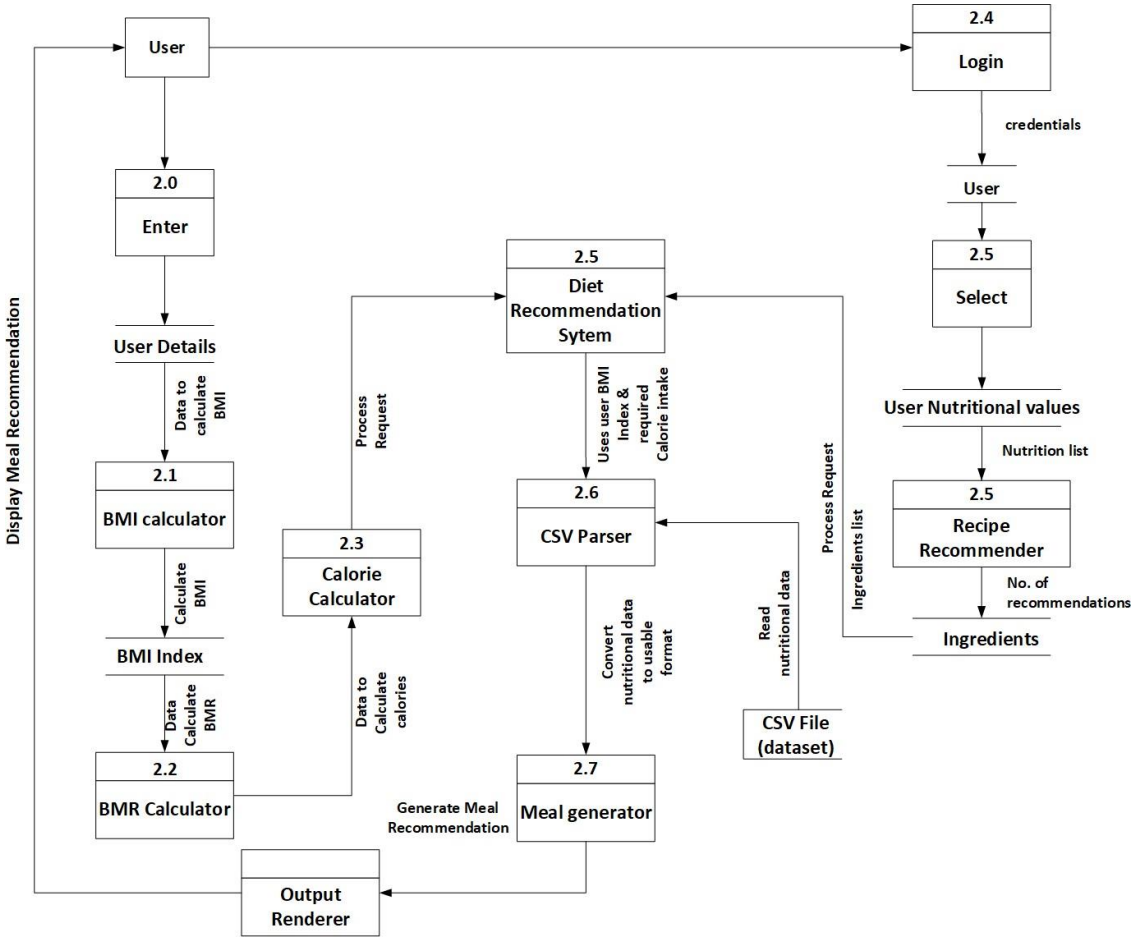


Figure 3. 6: DFD Level 2 of Smart Diet - Diet Recommendation System

3.5. Class Diagram

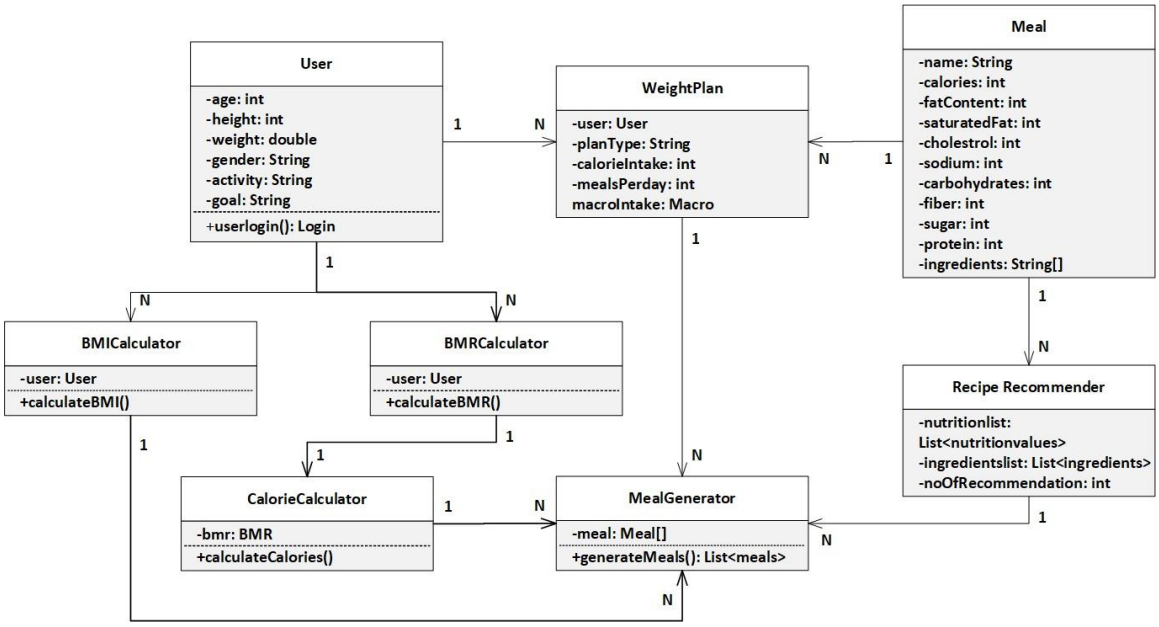


Figure 3. 7: Class Diagram of Smart Diet - Diet Recommendation System

3.6. State and Sequence Diagrams

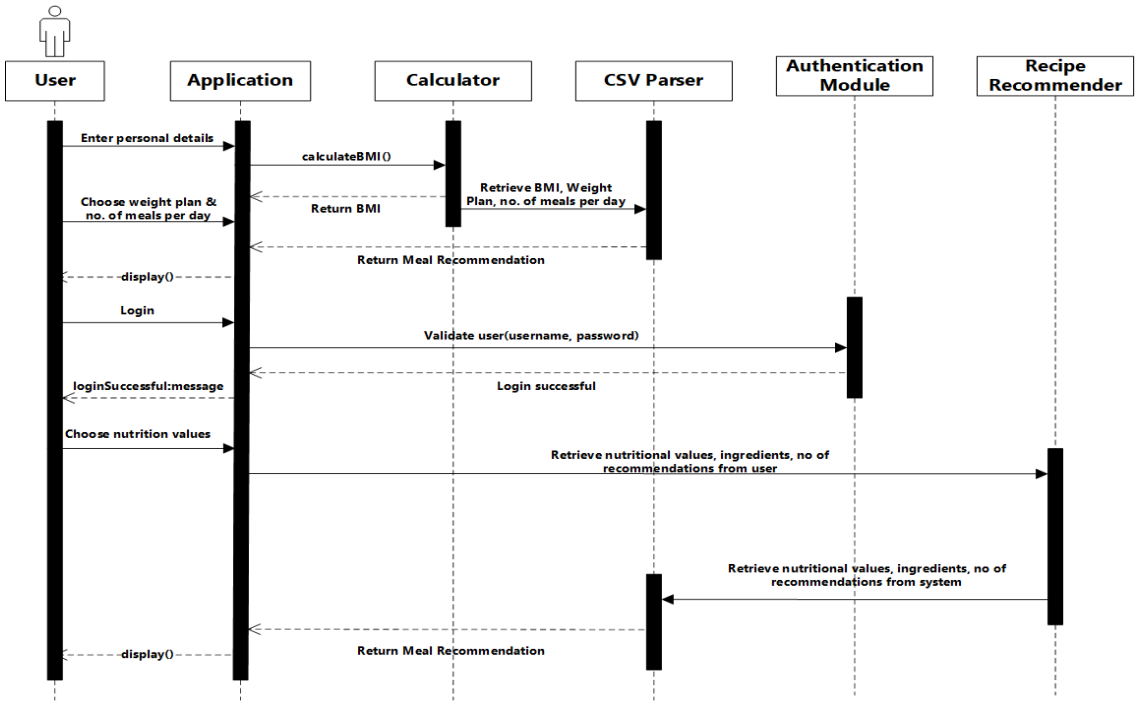


Figure 3. 8: Sequence Diagram of Smart Diet - Diet Recommendation System

3.7. Activity Diagrams

This is the basic activity diagram to get the BMI-based recommendations.

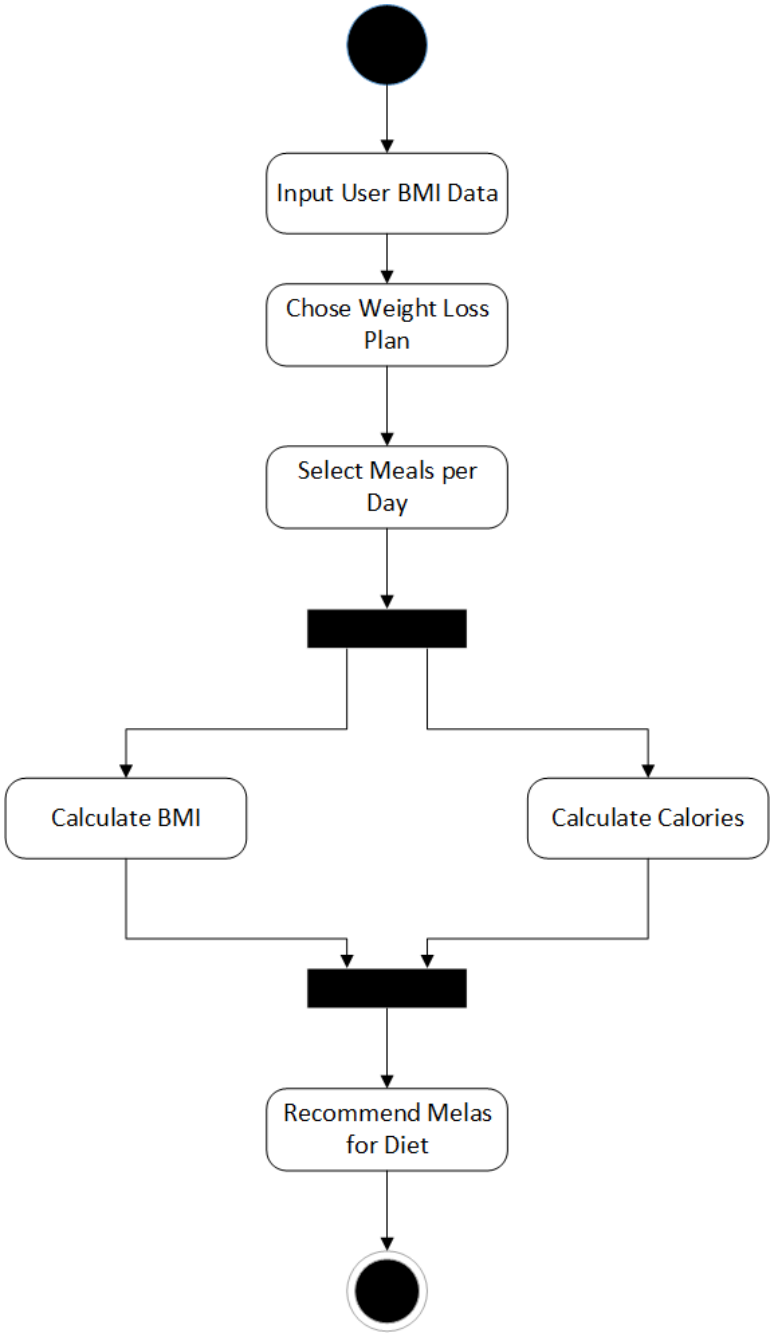


Figure 3. 9: Activity Diagram for BMI Based recommendation of Smart Diet - Diet Recommendation System

Likewise, this is the basic activity diagram to get the Nutrition values-based recommendations.

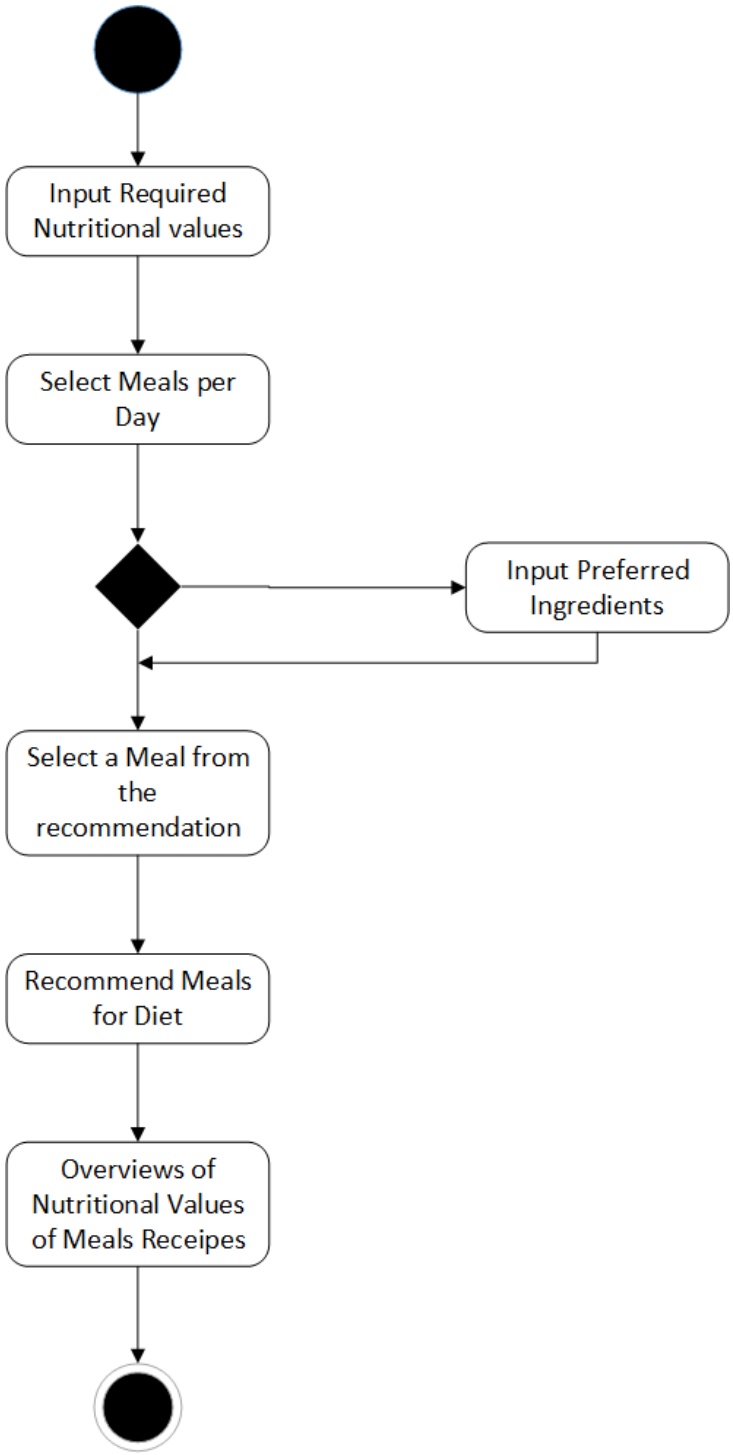


Figure 3. 10: Activity Diagram for Nutrition Based Recommendation of Smart Diet - Diet Recommendation System

4. System Design

System design is an important part of software project development and management, where the software architecture, modules, and components required for a successful software product are planned and organized. This design process is crucial in ensuring that the software development process is successful, and the software product meets all the desired specifications and requirements.

4.1. Form Design

To create a data dictionary, we need to identify the relevant variables and their respective data types, constraints, and descriptions. Here is an example of a data dictionary for the code provided:

4.1.1. Personalized Recommendation

Table 4. 1 : User Authentication

Field Name	Data Type	Constraints	Description
names	List	Cannot be empty or null	List of user names for authentication
usernames	List	Cannot be empty or null	List of usernames for authentication
passwords	List	Cannot be empty or null	List of passwords
hashed passwords	Dict	Must be dictionary containing the hashed password	Dictionary containing hashed passwords
file_path	Path	Must be a valid file path	File path for storing hashed passwords

Table 4. 2: Recommendation

Field name	Data Type	Constraints	Description
nutrition_list	List	Cannot be empty or null	List of nutrition values
nb_recommendations	Integer	Must be a positive integer	Number of recommendations to generate
Ingredient _txt	String	Cannot be empty or null	Ingredients specified for recommendations

Table 4. 3 : Display

Field name	Data Type	Constraints	Description
nutritions_values	List	Cannot be empty or null	List of nutrition values to display

Table 4. 4: Generated Recommendation

Field name	Data Type	Constraints	Description
generated	Boolean	N/A	flag indicating if recommendations were generated
recommendations	List or None	N/A if not found, otherwise list	List of recommended recipes or none if not found

Table 4. 5: Recipe

Field name	Data Type	Constraints	Description
Name	String	Cannot be empty or null	The name of the recommended recipe
ImageLink	String	Cannot be empty or null	The URL of the image associated with the recipe

RecipeIngredientParts	List	Cannot be empty or null	List of ingredients for the recipe
RecipeInstructions	List	Cannot be empty or null	List of instructions for preparing the recipe
CookTime	Integer	Must be a positive integer or zero (0)	Cooking time for the recipe (in minutes)
PrepTime	Integer	Must be a positive integer or zero (0)	Preparation time for the recipe (in minutes)
TotalTime	Integer	Must be a positive integer or zero (0)	Total time for cooking and preparation (minutes)
Calories	Float	Must be a non-negative float	Calorie content of the recipe
FatContent	Float	Must be a non-negative float	Fat content of the recipe
SaturatedFatContent	Float	Must be a non-negative float	Saturated fat content of the recipe
CholesterolContent	Float	Must be a non-negative float	Cholesterol content after recipe
SodiumContent	Float	Must be a non-negative float	Sodium content of the recipe
CarbohydrateConten	Float	Must be a non-negative float	Carbohydrate content of the recipe
FiberContent	Float	Must be a non-negative float	Fat content of the recipe
SugarContent	Float	Must be a non-negative float	Sugar content of the recipe

ProteinContent	Float	Must be a non-negative float	Protein content of the recipe

4.1.2. Contact

Table 4. 6: Contact Information

Field name	Data Type	Constraints	Description
name	String	required	The name of the contact
email	String	required	The email address of contact
message	String	N/A	The message from the contact

4.1.3. Generate Recommendations

Table 4. 7: Generator

Field name	Data Type	Constraints	Description
nutrition_input	List	Cannot be empty or null	List of nutrition inputs
ingredients	List	Cannot be empty or null	List of ingredients
params	Dict	N/A	Dictionary of parameters
request	Dict	N/A	Dictionary represent the request
response	Response	N/A	Response from the backend server

4.1.3.1. Automatic Diet Recommendation

Table 4. 8: Person

Field name	Data Type	Constraints	Description
age	Integer	Must be a positive interger	Age of the person
height	Integer	Must be a positive interger	Height of the person in centimeters
weight	Integer	Must be a positive interger	Weight of the person in kilograms
gender	String	Must be either “Male” or “Female”	Gender of the person (Male or Female)
activity	String	N/A	Level of physical activity
meals_calories_perc	Dict	Cannot be null or empty	Dictionary containing meal names and their respective calorie percentages
weight_loss	Float	N/A	Weight loss plan

Table 4. 9: Display

Field name	Data Type	Constraints	Description
plans	List	Cannot be null or empty	List of weight loss plans

Table 4. 10: Recommendation

Field name	Data Type	Constraints	Description
nutritions_values	list	Cannot be null or empty	List of nutrition values

4.2. Process design

4.2.1. Flow Chart

This is the flowchart of the system.

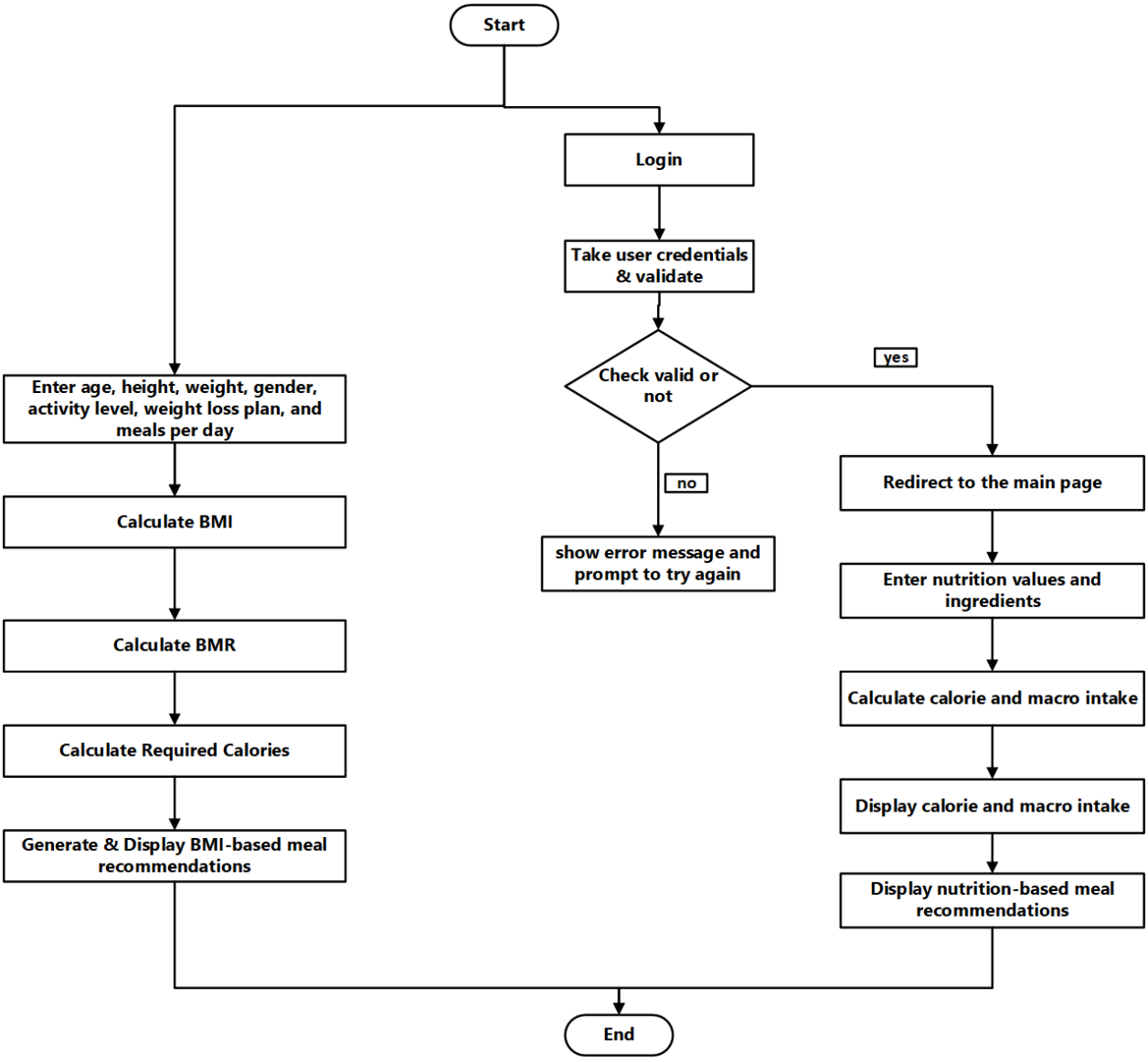


Figure 4. 1: Flow Chart of the whole System of Smart Diet

This is the flowchart of the basic diet recommendations without user login.

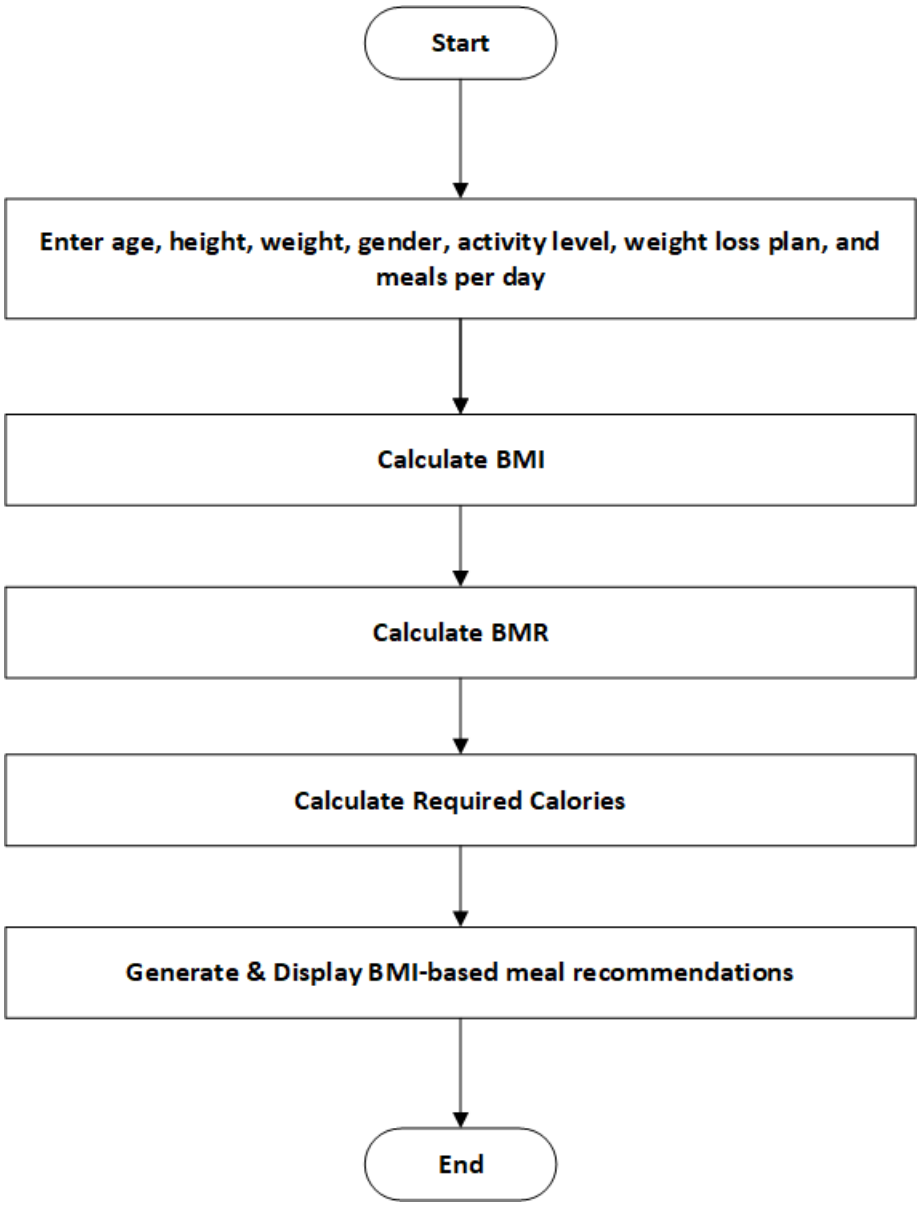


Figure 4. 2: Flow Chart of the Automatic BMI Based Recommendation

This is the flowchart of the basic diet recommendations with user login.

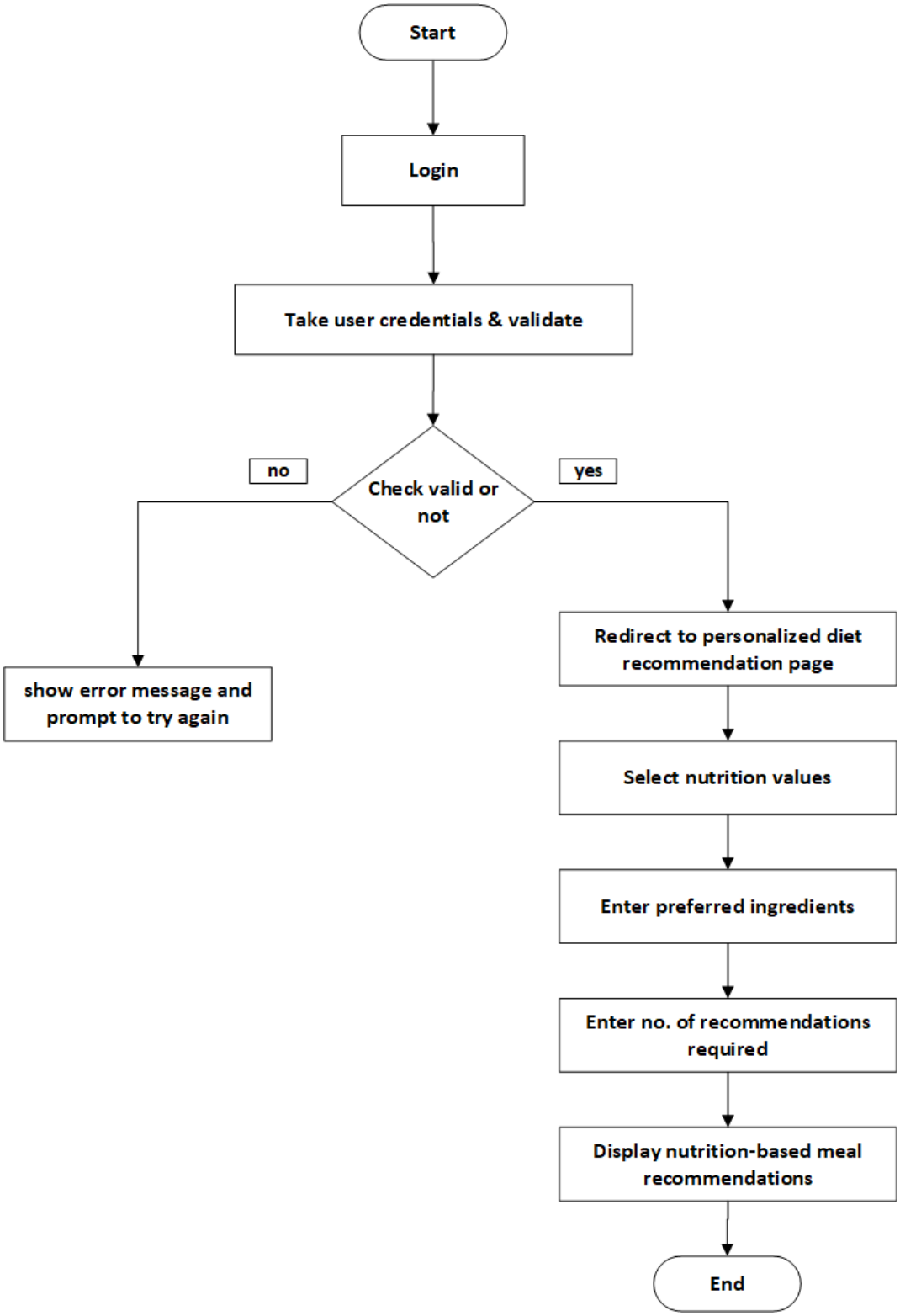


Figure 4. 3: Flow Chart of the Personalized Nutrition Based Recommendation

4.2.2. Component Diagrams

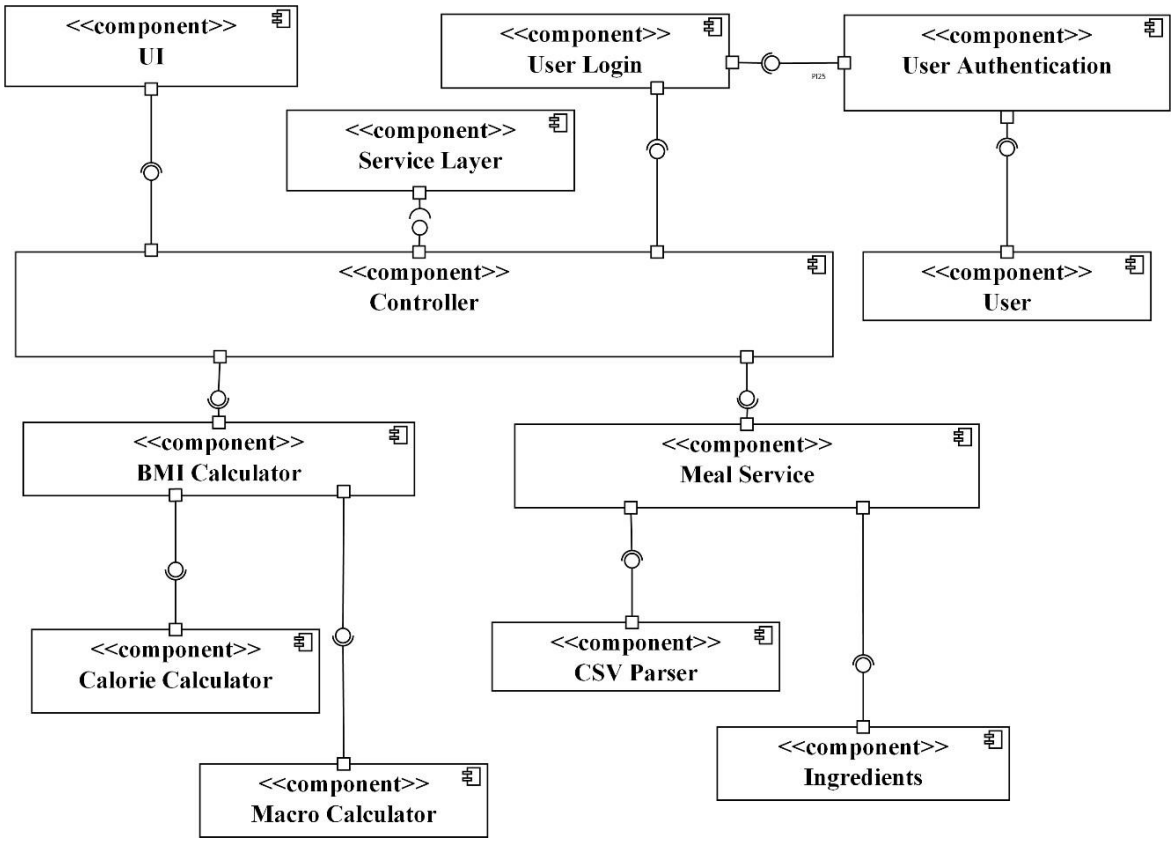


Figure 4. 4: Component Diagram of Smart Diet - Diet Recommendation System

5. Implementation and Testing

5.1. Implementation

Implementation is the process of turning the design of a software system into a functional software product. It involves writing code, integrating different software components, and testing the system to ensure that it meets the desired requirements and specifications.

5.1.1. Tools Used

The tools we have used in our project are listed below:

Table 5. 1: Tools used

Tools	Description
Draw.io, Microsoft Visio	Used for drawing UML models
GitHub	Version control of code
Docker	Used for build, deploy, run and manage containers
Visual Studio Code	Integrated development environment
DataSet	Kaggle

The following are the programming language and libraries used in our system.

Table 5. 2: Programming Language and Libraries Used

Language/Libraries	Description
Python	Programming Language used for model development
Streamlit	Python library used for frontend development
FastAPI	Web framework used for backend development
HTML/CSS	Programming Language used for UI design
Sklearn	Open-source data analysis library
Numpy	python library used for working with Numerical value (arrays)
Pandas	Python library used for working with datasets.

Matplotlib	Graph plotting python library used for data visualizations
TensorFlow	Python library used for functions and classes
Beautifulsoup4	This library is widely used for web scraping and parsing HTML or XML documents.
streamlit-echarts	Library used for displaying charts from echarts json option as python dict

5.1.2. Implementation Details of Modules

Using the tools and libraries mentioned above we implemented Content based Algorithm

Step 1: Build Content based Model

The content-based filtering algorithm is used in the `recommend` function. Here's the breakdown of the algorithms used:

1. Content-based Filtering Algorithm:

   a. Content-based filtering is used to recommend recipes based on the input nutrition and ingredient data.
   b. The `recommend` function takes the `dataframe`, `_input`, `ingredients`, and `params` as input.
   c. It first calls the `extract_data` function to extract the relevant data from the `dataframe` based on the provided `ingredients`.
   d. If the extracted data has enough records (`shape[0] >= params['n_neighbors']`), it proceeds with the recommendation process.
   e. It then calls the `scaling` function to standardize the numeric features of the extracted data.
   f. Next, it uses the `nn_predictor` function to build a Nearest Neighbors model using cosine similarity as the distance metric.
   g. The `build_pipeline` function is called to create a pipeline that includes standard scaling and the Nearest Neighbors model.

h. Finally, the `apply_pipeline` function is used to apply the pipeline and retrieve the recommended recipes based on the input `_input`.

2. Nearest Neighbors Algorithm:

a. The Nearest Neighbors algorithm is used in the `nn_predictor` function.
b. It creates a Nearest Neighbors model using the `NearestNeighbors` class from the scikit-learn library.
c. The model is trained on the preprocessed data obtained from the `scaling` function.
d. The cosine similarity metric is used to measure the distance between data points.
e. The `recommend` function sets the value of `params['n_neighbors']` to determine the number of nearest neighbors to consider for recommendation.

Step 2: Train Function

In the code provided, the training and testing functions are used as follows:

1. Training the model:

- The `scaling` function is used to preprocess the data and obtain the standardized data (`prep_data`) and the scaler object (`scaler`).

- `nn_predictor` function is used to create and train the Nearest Neighbors model (`neigh`) using the preprocessed data.

2. Building the pipeline:

- The `FunctionTransformer` is used to create a transformer from the `neigh.kneighbors` function, with specified keyword arguments (`params`).

- The pipeline is created with two steps: standard scaling (`std_scaler`) and the Nearest Neighbors transformer (`NN`).

Step 3: Test Function

1. Extracting data for testing:

- The `extract_data` function is used to filter the data based on ingredient and maximum nutritional values, and return the extracted data for testing.

2. Applying the pipeline for testing:

- The `apply_pipeline` function is used to transform the testing input `_input` using the pipeline. It returns the corresponding row(s) from the `extracted_data` based on the transformed input.

5.1.2.1. Description

BMI formula: - Below are the equations used for calculating BMI in the International System of Units (SI).

SI/ Metric UNITS: $BMI = \frac{mass\ (kg)}{height^2\ (m)}$

BMI table for adults: - This is the WHO recommended body weight based on BMI values for adults. It is used for both men and women, age 20 or older.

Table 5. 3: BMI Table for Categorization

Classification	**BMI range -kg/m2**
Severe Thinness	<16
Moderate Thinness	16-17
Mild Thinness	17-18.5
Normal	18.5-25
Overweight	25-30
Obese Class I	30-35
Obese Class II	35-40
Obese Class III	>40

BMI chart for adults: - This is a graph of BMI categories based on the WHO data. The dashed lines represent subdivisions within a major categorization. [2]

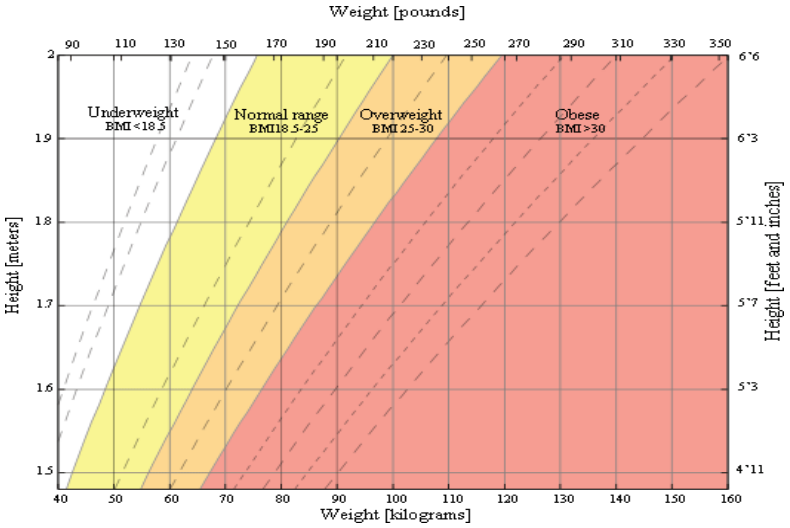


Figure 5. 1: BMI Chart

Calorie Calculator: - The Calorie Calculator combines all three equations i.e., Mifflin-St Jeor Equation, Revised Harris Benedict Equation and Katch Mc-Ardle Formula to calculate an estimated average of daily calorie needs. The three equations are as follows: -

- **Mifflin-St Jeor Equation:**
    - For men: BMR = 10W + 6.25H - 5A + 5
    - For women: BMR = 10W + 6.25H - 5A - 161

Content Recommendation

For model development, the recommendation engine we developed was built using the Nearest Neighbors algorithm, which is an unsupervised learning algorithm that can perform neighbor searches. The algorithm provides a uniform interface to three different nearest neighbor algorithms: BallTree, KDTree, and a brute-force algorithm based on routines in the sklearn.metrics.pairwise module. For our recommendation engine, we used the brute-force algorithm with cosine similarity because it is efficient for small datasets. The cosine similarity formula for comparing the similarity between two vectors, such as the nutrition values of two meals, is:

cos(theta) = (A * B) / (||A|| * ||B||)

OR simply,

cosine_similarity(A, B) = dot_product(A, B) / (norm(A) * norm(B))

where: dot_product(A, B) is the dot product between vectors A and B.

norm(A) is the Euclidean norm (magnitude) of vector A.

In this case, A and B represent the nutrition values of two different meals. The cosine similarity score ranges from -1 (totally dissimilar) to 1 (totally similar), with 0 indicating no similarity. [1]

5.2. Testing

For testing the project, we will be going through the series of test cases to check the modules of our project. After the successful

5.2.1. Test Cases for Unit Testing

Table 5. 4: Table of Unit Testing cases

Test case ID	Test Case	Expected Result	Actual Result
T1	Page loading	The home page should load correctly and in a reasonable amount of time.	Home Page Loaded successfully.
T2	Layout	The layout of the home page should be consistent and visually appealing.	Layout was consistent and visually appealing.
T3	Navigation	The navigation bar should be easy to understand and use. It	Navigation Bar Worked properly.

		should route to the correct pages.	
T4	Links	All the links should be correct. They should route to intended correct destinations.	All links worked properly.
T5	Images	All the images used in the homepage should be displayed correctly.	All images were displayed.

5.2.2. Test Cases for System Testing

Table 5. 5: Table of System Testing cases

Test case ID	Test Case	Expected Result	Actual Result
T6	Browser compatibility	The application should work in different browsers like Microsoft Edge, Google Chrome, Firefox, etc.	Worked successfully in Microsoft Edge, Google Chrome and Firefox.
T7	Calculate BMI	The application should calculate the BMI index using user's age, height, weight and activity level.	BMI Index was calculated successfully.
T8	Calculate Calorie	The application should calculate required calorie based on the BMI Index.	Calorie amount calculated successfully.
T9	Generate meals based on BMI	The application should generate meals recommendation on the basis of the BMI Index.	BMI based diet recommended successfully.

T10	Generate meals based on nutrition values	The application should generate meals recommendation on the basis of the amount of nutrition values provided by the user.	Nutrition Values based diet recommended successfully.
T11	Generate meals based on ingredients	The application should generate meals recommendation on the basis of the different ingredients the user wants in their meal.	Ingredients based diet recommended successfully.
T12	Generate meals for meal planning	The application should generate meals recommendation based on the basis the user plan their meal i.e., 3 meals or 4 meals or 5 meals per day.	Meals for the meal planning was generated successfully.
T13	Security	The application should ensure the security of user data.	The application is secure.

5.3. Result Analysis

The result of the testing of for project Smart Diet was analyzed and validated that it meets our objectives and requirements. Any errors found during the unit testing were debugged before conducting the integration and system testing. The results from the test case showed that the Smart Diet system is functioning as intended and is reliable and secure.

6. Conclusion and Future Recommendations

6.1. Conclusion

In conclusion, the development of a diet recommendation system can be extremely beneficial for individuals seeking to improve their health and wellness through diet. By leveraging data on individual preferences and health goals, the system can provide personalized meal plans and nutritional guidance to help users make informed choices about their diets. However, it is important to note that such a system should not replace the advice of a qualified healthcare professional, and users should always consult with a medical professional before making significant changes to their diets.

6.2. Future Recommendations

There is a scope for the further development in our project to a great extent. A number of features like can be added to this system in future. Another feature we wished to implement was providing recommendations based on the allegories. System may keep track of their diet plan for certain period of plan so that different diet plans for can be provided to the user. Furthermore, we could collaborate with healthcare professionals such as registered dieticians, pediatricians to review the recommendations and provide feedback. These features could have been implemented if time, skills and budget did not limit our team.

6.3. Limitation

Due to the lack of time, skills and manpower, our application has the following limitations.

i. The application can only provide generic meal recommendations based on the user's input. It does not have the ability to factor in allergies, or dietary restrictions, which can limit its effectiveness and accuracy.
ii. Since the application relies on pre-existing data from a CSV file, it cannot provide real-time updates on the nutritional value of food items or ingredients
iii. The lack of tracking functionality means that users cannot monitor their progress over time or make changes to their meal plan based on their results.

References

[1] H. a. K. L. Ghanim, "Development of Recommendation Systems Based on Collaborative Filtering Using KNN and SVD," 06 2020. [Online].

[2] "BMI Calculator," [Online]. Available: https://www.calculator.net/bmi-calculator.html.

[3] "Calorie Calculator," [Online]. Available: https://www.calculator.net/calorie-calculator.html.

[4] "Meal Calorie Calculator," [Online]. Available: https://www.omnicalculator.com/health/meal-calorie.

[5] P. M. M. +. P. a. D. M. P. M. R. Leanne Loranger, "Clinician's Commentary on O'Donovan et al.," *Physiother Can.,* vol. 69(3), 2015.

[6] H. Q. L. YanMeng, "Efficacy of low carbohydrate diet for type 2 diabetes mellitus management: A systematic review and meta-analysis of randomized controlled trials," *Diabetes Research and Clinical Practice,* vol. 131, 2017.

[7] E. F. S. M. J. F. K. N. Corinna Hawkes, "Child-centered food systems: Reorienting food systems towards healthy diets for children," *Global Food Security,* vol. 27, 2019.

[8] Y. J. Youngyo Kim, "A modified Mediterranean diet score is inversely associated with metabolic syndrome in Korean adults," *Eur J Clin Nutr,* vol. 72(12), 2018.

[9] P. S. N. A. M. S. S. P. H. A.-H. M. N. K. B. M. Y. P. R. G. B. N. M.-Y. P. N. S. S. M. a. Z. A.-M. M. Shahrul A Noah, "DietPal: A Web-Based Dietary Menu-Generating and Management System," *J Med Internet Res,* vol. 6(1), 2004.

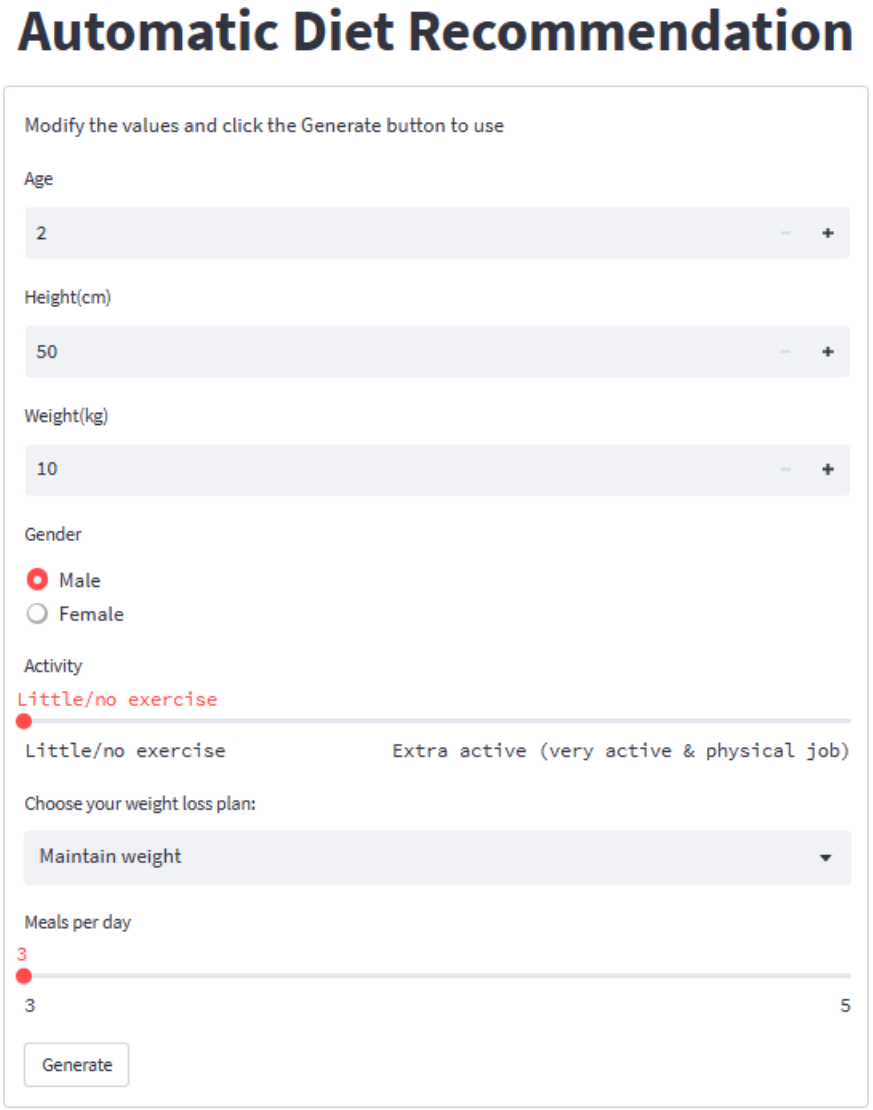
[10] "Mealime Support Docs," Mealime, [Online]. Available: https://support.mealime.com/.

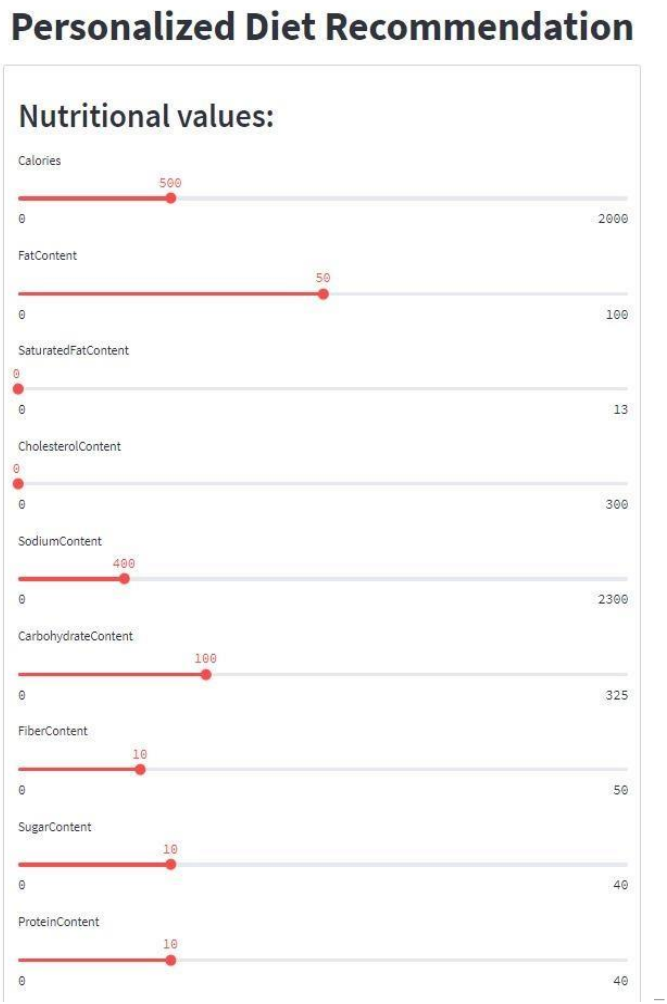
[11] "How it works - Eat This Much," Eat This Much Inc., [Online]. Available: https://www.eatthismuch.com/how-it-works/.

[12] ,. A. D. B. S. Suresh Chandra Devkota, "Nutritional Deficiencies: Major Public Health Problem in Nepal," *American Journal of Public Health Research,* vol. 3, 2015.

Appendix

Interface of Smart Diet




Personalized Diet Recommendation
Nutritional values:
Calories
500
0
2000
FatContent
50
0
100
SaturatedFatContent
0
0
13
CholesterolContent
0
0
300
SodiumContent
400
0
2300
CarbohydrateContent
100
0
325
FiberContent
10
0
50
SugarContent
10
0
40
ProteinContent
10
0
40
Recommendation options (OPTIONAL):
Number of recommendations
5
5
20
Specify ingredients to include in the recommendations separated by ";" :
Ingredient1;Ingredient2;...
Example: Milk;eggs;butter;chicken...
Generate

BMI CALCULATOR

Body Mass Index (BMI)

20.81 kg/m²

Normal

Healthy BMI range: 18.5 kg/m² - 25 kg/m².

CALORIES CALCULATOR

The results show a number of daily calorie estimates that can be used as a guideline for how many calories to consume each day to maintain, lose, or gain weight at a chosen rate.

Maintain weight	Mild weight loss	Weight loss	Extreme weight loss
1626 Calories/day	1464 Calories/day	1301 Calories/day	976 Calories/day
↓ -0 kg/week	↓ -0.25 kg/week	↓ -0.5 kg/week	↓ -1 kg/week

DIET RECOMMENDATOR

Recommended recipes:

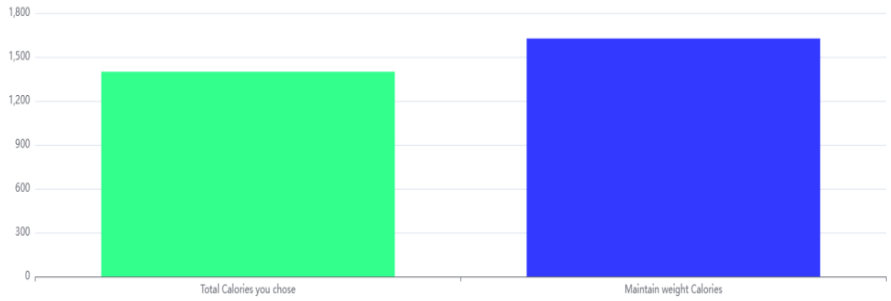
BREAKFAST	LUNCH	DINNER
Angel Hair Pasta With Sauteed Cherry Tomatoes, Lemon and Tuna	Chicken Cashew Pasta	Venison Black Bean Chili
Wi Gal's Chicken Rice Stove Top Dish Made Healthier	Summer Salmon Pasta	Grilled Tuna With White Bean and Charred Onion Salad
Pasta With Tuna and Red Peppers	Sea Bass on a Bed of Swiss Chard and Browned Rosemary Potatoes	Modern Venison Roast

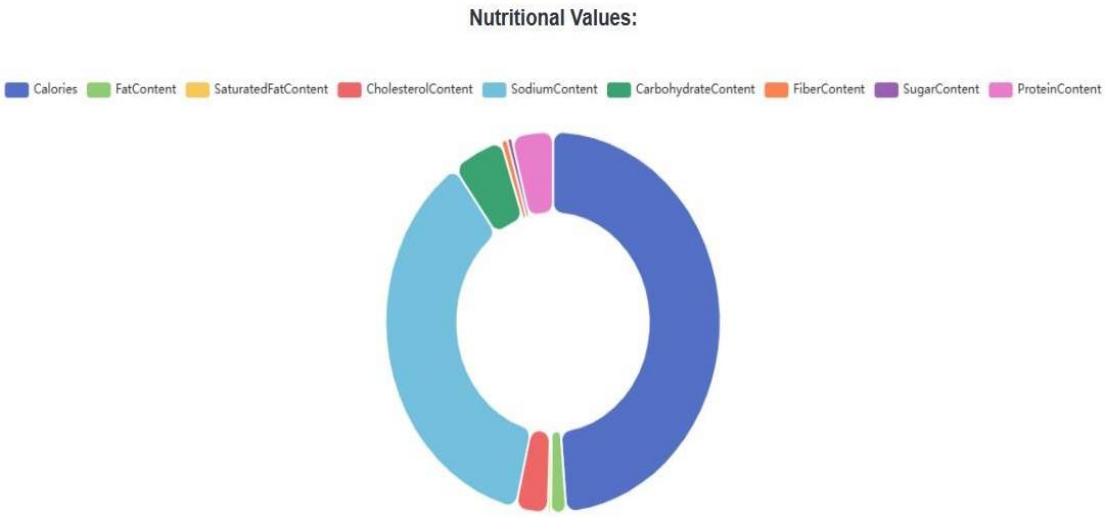
✅ Recommendation Generated Successfully !

Choose your meal composition:

Choose your breakfast:	Choose your launch:	Choose your dinner:
Angel Hair Pasta With Sauteed Cherry Tomatoes, Lemon and Tuna	Chicken Cashew Pasta	Venison Black Bean Chili

Total Calories in Recipes vs Maintain weight Calories:

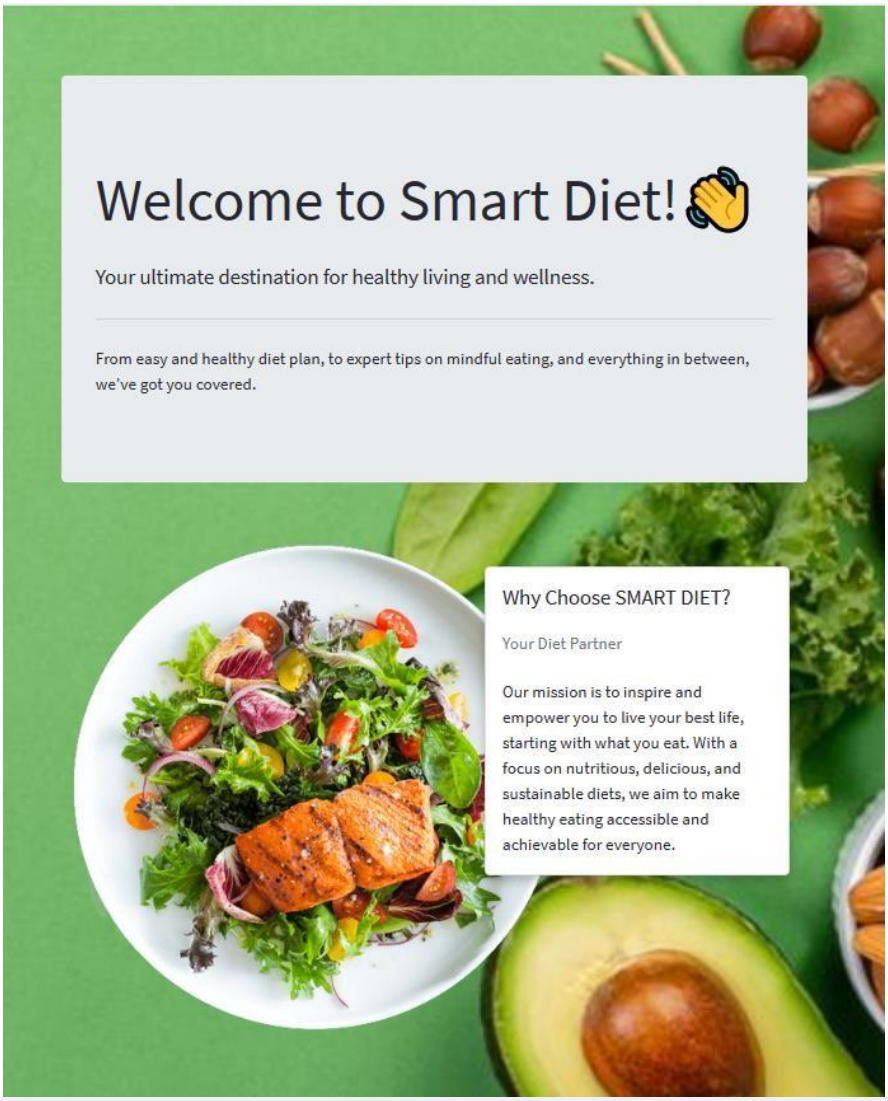


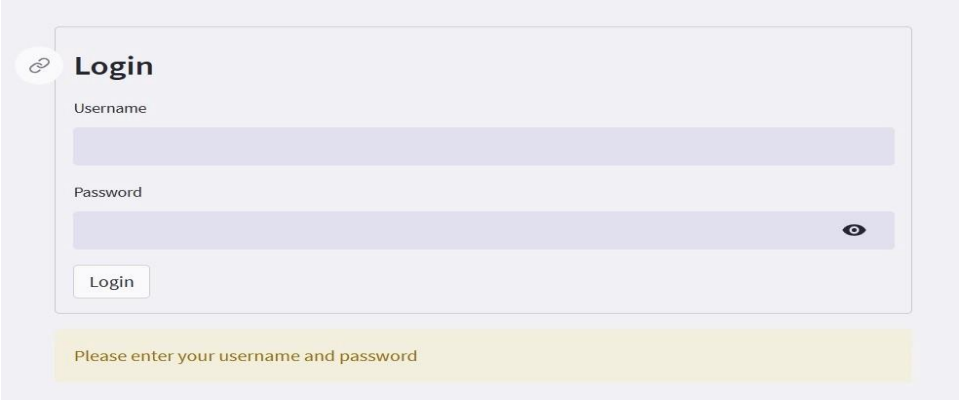


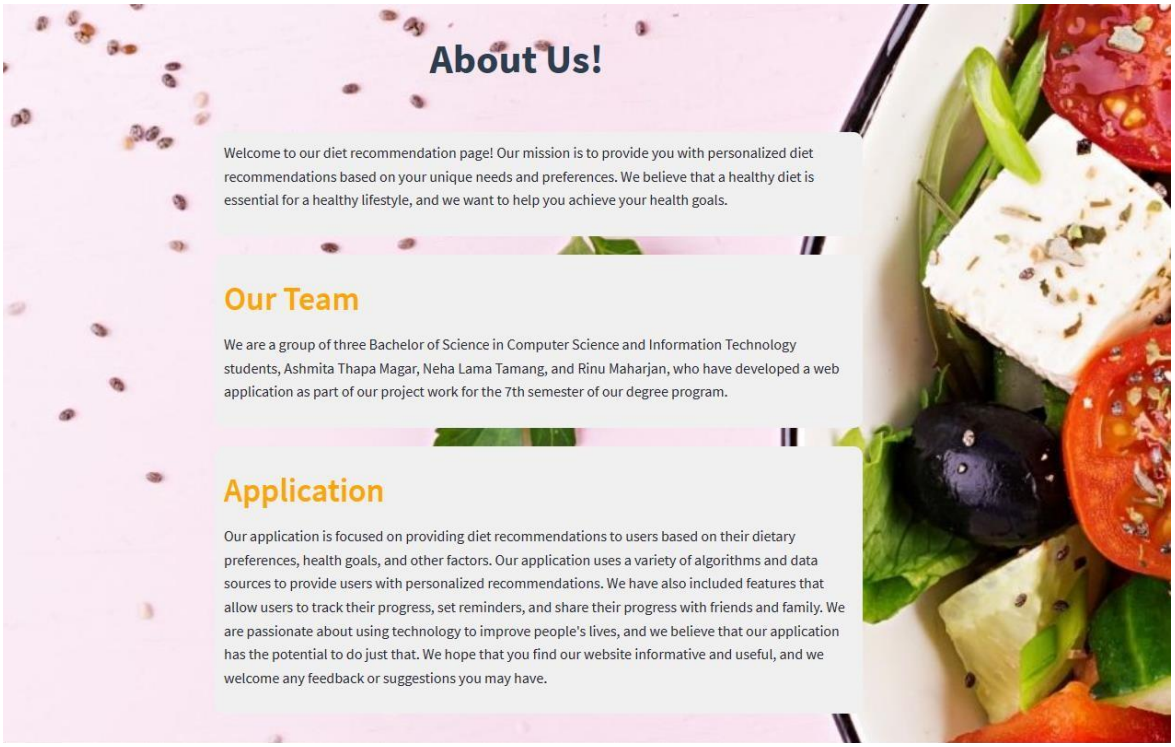
Recommended recipes:

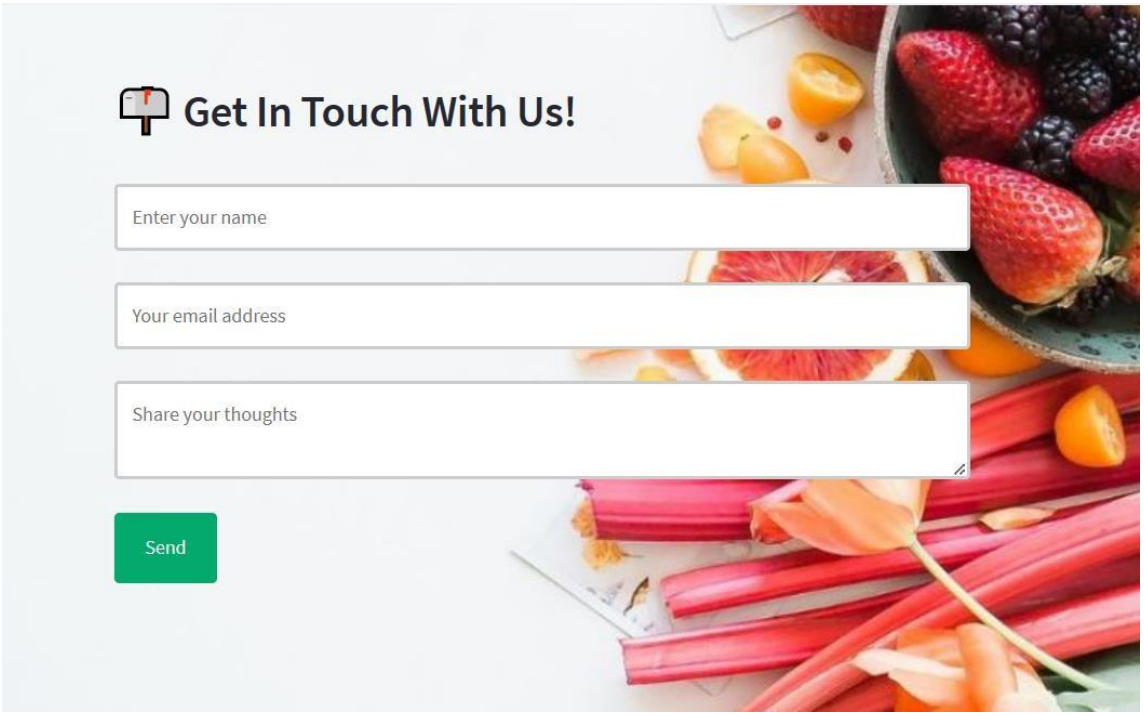
German Potato Salad (Vegetarian) | Seafood Phyllo Parcels | Mediterranean Quinoa Salad | Patatas Brava | Harvest Turkey, Cranberry and Brown Rice Salad

Overview:


Welcome to Smart Diet!
Your ultimate destination for healthy living and wellness.
From easy and healthy diet plan, to expert tips on mindful eating, and everything in between, we've got you covered.
Why Choose SMART DIET?
Your Diet Partner
Our mission is to inspire and empower you to live your best life, starting with what you eat. With a focus on nutritious, delicious, and sustainable diets, we aim to make healthy eating accessible and achievable for everyone.


Login
Username
Password
Login
Please enter your username and password


About Us!
Welcome to our diet recommendation page! Our mission is to provide you with personalized diet recommendations based on your unique needs and preferences. We believe that a healthy diet is essential for a healthy lifestyle, and we want to help you achieve your health goals.
Our Team
We are a group of three Bachelor of Science in Computer Science and Information Technology students, Ashmita Thapa Magar, Neha Lama Tamang, and Rinu Maharjan, who have developed a web application as part of our project work for the 7th semester of our degree program.
Application
Our application is focused on providing diet recommendations to users based on their dietary preferences, health goals, and other factors. Our application uses a variety of algorithms and data sources to provide users with personalized recommendations. We have also included features that allow users to track their progress, set reminders, and share their progress with friends and family. We are passionate about using technology to improve people's lives, and we believe that our application has the potential to do just that. We hope that you find our website informative and useful, and we welcome any feedback or suggestions you may have.


Get In Touch With Us!
Enter your name
Your email address
Share your thoughts
Send