Fitnessstan

By:

Zain Ul Abideen
35515

Obaid Ullah
35739

Huzaifa Khan
35726

Supervised by:
M. Islam Abbasi (Lecturer)

Faculty of Computing
Riphah International University, Islamabad

Spring/Fall 2024

A Dissertation Submitted To

Faculty of Computing,

Riphah International University, Islamabad

As a Partial Fulfillment of the Requirement for the Award of the Degree of

Bachelors of Science in Computer Science

Faculty of Computing
Riphah International University, Islamabad

Date: 26 December, 2024

Final Approval

This is to certify that we have read the report submitted by ***Zain Ul Abideen (35515)*, *Obaid Ullah (35739), Huzaifa Khan (35726)*** for the partial fulfillment of the requirements for the degree of the Bachelors of Science in Computer Science (BSCS). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelors of Science in Computer Science (BsCs).

Committee:

1

M. Islam Abbasi

(Supervisor)

2

Dr. Musharraf Ahmed

(Head of Department)

Declaration

We hereby declare that this document "**Fitnessstan**" neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers, especially our supervisor **Muhammad Islam Abbasi**. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from anywhere else, we shall stand by the consequences.

____

Zain Ul Abideen

35515

____

Obaid Ullah

35739

____

Huzaifa Khan

35726

Dedication

This work is devoted to Almighty Allah, the Most Merciful and the Most Beneficent, who granted us the knowledge, perseverance, and strength to complete this work. We also wish to extend our deepest gratitude to our parents for their unyielding love, support, and prayers along the journey. We finally dedicate this to our supervisor, “Muhammad Islam Abbasi”, with whose valuable guidance, mentorship, and encouragement we are able to make this project into reality.

Acknowledgement

We are grateful to Almighty Allah, the provider of all knowledge and wisdom, for giving us the strength and perseverance to bring this project to fruition.

We would like to extend our wholehearted gratitude to our supervisor, “Muhammad Islam Abbasi” who guided and encouraged us on this endeavor throughout the course of this project. His mentorship has been the backbone of our learning and growth.

We would also like to acknowledge the encouragement and constructive criticism offered by our faculty members and colleagues who guided us to correct our mistakes. Their encouraging words pushed us beyond our limits of apprehension. Last but not least, we express our profound appreciation to our families and friends for their exemplary support, prayers, and patience that provided us with strength in pursuit of excellence.

____

Zain Ul Abideen

35515

____

Obaid Ullah

35739

____

Huzaifa Khan

35726

Abstract

In today's fast-paced world, people work tirelessly with dedication but often neglect their health and fitness, leading to a steady rise in the number of unhealthy individuals. Addressing this concern, we aim to develop a comprehensive platform, available as both a website and a mobile application, to help users manage their daily exercise and food intake effectively.

Existing apps and websites in Pakistan often fall short in providing personalized food recommendations tailored to users' needs. While some platforms suggest exercise routines, they rarely focus on balanced nutrition or calorie management, which are essential for maintaining overall health. Moreover, these platforms lack customization based on user preferences, leaving a gap in truly effective fitness solutions.

Our solution, **Fitnessstan**, bridges this gap by offering personalized food and exercise recommendations. Using Flutter technology, we developed a user-friendly app that integrates an AI model to suggest appropriate calorie intake and workouts based on individual requirements. For users preferring web access, we also created a responsive website using React.js.

Fitnessstan is designed to empower users with tailored guidance, promoting healthier lifestyles through a combination of technology, innovation, and convenience. With this platform, we strive to make fitness accessible and achievable for everyone.

Table of Content

List of Tables

List of Figure

Chapter 1: Introduction

Chapter 1: **Introduction**

Fitnessstan is a web-based and app-based platform designed to provide a holistic solution for fitness enthusiasts, including individuals with diabetes. In today's world, achieving a healthy lifestyle can be challenging, especially for beginners who lack personal guidance on balancing exercise and nutrition. Neglecting either aspect often leads to unsatisfactory progress or health risks. Existing platforms fail to offer comprehensive and personalized solutions, particularly for users with specific health needs.

Fitnessstan addresses these challenges by integrating artificial intelligence to deliver customized workout and dietary plans tailored to each user's unique health metrics, goals, and preferences. The platform specifically supports diabetic patients by incorporating features that help manage their condition through personalized exercise and nutrition guidance. This seamless integration of fitness and health ensures safe and effective progress for all users.

With features like progress tracking and adaptive recommendations, Fitnessstan ensures long-term results and user satisfaction. Whether its weight loss, muscle gain, or managing conditions like diabetes, Fitnessstan empowers individuals to take control of their health journey while promoting a culture of sustainable fitness and well-being.

1.1 **Goals and Objectives**

1.1.1 **Goals**

1. The platform in which Provide a personalized and user-friendly fitness platform for enthusiasts.
2. To facilitate users achieve and maintain their fitness goals through AI-powered workout and dietary recommendations.

1.1.2 **Objectives**

1. To offer fitness workout plans as per users' fitness levels, preferences, and health metrics.
2. To provide dietary recommendations personalized to the specific needs of diabetic patients and general users, respectively.
3. To enable users to track progress and result analysis in measurable improvements over time.

1.2 **Scope of the Project**

The scopes of the **“Fitnessstan”** are as following

1. Our website will be developed on java stack development and our app will be developed on Flutter.
2. We will develop a website and app that will provide platform for diabetic patient and for those people who loss, gain and maintain their weight.
3. User will upload their information without any hesitation because we are securing his information.
4. User will get personalized diet plan on the basis of his information.
5. Problem will be solved with Machine learning techniques.

The platform would also be accessible, interactive and user-friendly, and simplified for Users.

Chapter 2: Literature Review

Chapter 2: **Literature Review**

2.1 **Introduction**

Fitnessstan is a web-based and app-based platform designed to provide a holistic solution for fitness enthusiasts, including individuals with diabetes. In today’s world, achieving a healthy lifestyle can be challenging, especially for beginners who lack personal guidance on balancing exercise and nutrition. Neglecting either aspect often leads to unsatisfactory progress or health risks. Existing platforms fail to offer comprehensive and personalized solutions, particularly for users with specific health needs.

Fitnessstan addresses these challenges by integrating artificial intelligence to deliver customized workout and dietary plans tailored to each user’s unique health metrics, goals, and preferences. The platform specifically supports diabetic patients by incorporating features that help manage their condition through personalized exercise and nutrition guidance. This seamless integration of fitness and health ensures safe and effective progress for all users.

With features like progress tracking and adaptive recommendations, Fitnessstan ensures long-term results and user satisfaction. Whether its weight loss, muscle gain, or managing conditions like diabetes, Fitnessstan empowers individuals to take control of their health journey while promoting a culture of sustainable fitness and well-being.

2.2 **Background and Problem Elaboration**

While the industry of fitness is rapidly expanding, on many platforms users, especially first-time users, end up stuck in their goals due to the fact that most do not properly guide one both through exercise and nutrition. Most of them only give workout plans without connecting the critical role of nutrition for achieving fitness goals. This leads to less useful results, bad recovery, and potential health risks.

Also, many of these applications do not support different types of constraints, such as diabetes, and provide no full, personalized recommendations. There is also disconnection between guidance provided on exercise and nutrition, making the whole experience disjointed and giving the user no tools needed to be successful in the long term.

Fitnessstan offers a remedy to that problem by introducing AI-based online services combining customized workout routines, suggested diets, and monitoring to create one all-inclusive solution where every user will have the right resources for a holistic fitness journey.

2.3 **Detailed Literature Review**

2.3.1 **Definition**

Fitness refers to a state of health and well-being that allows an individual to perform daily tasks with energy and minimal fatigue. It has physical, mental, and emotional factors to it and, thus, outlines the importance of good regular exercise and lifestyle habits.

A diet, on the other hand, is a short-term measure that incorporates certain food-related restrictions designed to fit the requirements for achieving objectives like weight loss or improvement in the state of being healthy. Many diets are temporary and specific to the client's need.

On the other hand, a nutrition plan is generally a long-term approach or an eating regimen designed to maintain the body's necessary macronutrient and micronutrient inputs by proper consumption to achieve certain health and fitness goals.

2.3.2 Related Research Work 1

Reema Golagana et al [1] proposed a diet recommendation system utilizing machine learning and deep learning techniques to create personalized meal plans based on a user's physical characteristics and health conditions. The system evaluates factors like BMI, age, and gender to determine the optimal daily calorie intake, and it recommends meal plans by considering the nutritional requirements of individuals. Using algorithms such as Random Forest, K-means, and Long Short-Term Memory (LSTM), the system outperforms traditional methods in predicting diet plans. The model is designed to help users lead healthier lives by providing tailored nutrition advice. The research highlights the importance of personalized diet recommendations and the use of advanced machine learning techniques to improve accuracy.

2.3.3 Related Research Work 2

Mr. Abhijeet Pawar et al [2]developed an AI-based fitness trainer app that helps users with their exercise routines at home using image and video processing. The goal of the app is to make fitness more accessible to people who can't afford gym memberships or have limited time. It calculates the user's BMI based on their height and weight and provides personalized workout plans according to different BMI categories, such as underweight, normal, overweight, and obese. The app uses OpenCV and Convolutional Neural Networks (CNN) to track body posture and movements during exercises. It offers features like real-time feedback, performance tracking, and the ability to detect and correct common mistakes to improve workouts. The app includes various exercise types, including cardio, flexibility, and strength training, to offer a complete fitness solution for home users.

2.4 Literature Review Summary Table

No.	Name	Authors	Year	Input	Output	Description
[1]	Diet Recommendation System Using Machine Learning	Reema Golagana, V. Sravani, T. Mohan Reddy, CH Kavitha	2023	User data (BMI, preferences)	Personalized diet plans	Uses Random Forest, K-Means, and LSTM for generating tailored diet recommendations.
[3]	Multi-Choice Diet Recommendation Application for Indian Scenario	Karthika Subbaraj	2024	BMI, TDEE, and Indian food dataset	Calorie-specific meal plans	Utilizes Random Forest for meal classification and KNN for alternative suggestions, achieving.
[4]	A Hybrid Healthy Diet Recommender System	Sara Sweidan, S.S. Askar, Mohamed Abouhawwash, Elsayed Badr	2025	Anthropometric and clinical measurements	Calorie and nutrient estimations	Combines SVR, LR, and DTR models to generate calorie estimates with R=0.985 for obesity treatment.
[5]	Personalized Diet Recommendation System Using Machine Learning	D. Navya Narayana Kumari, T. Praveen Satya, B. Manikanta, et al.	2024	User preferences and health parameters	Meal recommendations and preparation details	Employs Nearest Neighbors with cosine similarity for content-based filtering tailored to user preferences.

[6]	AI Fitness Traine	Ahsan Ashraf, Talha Shahid,	June 2024	Real time tracking	• Customized Workouts • Live Feedback • Motion Tracking • Exercise Tracking Integration with Wearable Devices	Pose Estimation with MediaPipe, and OpenCV for image and video processing, particularly using YOLOv5 for motion tracking and posture analysis.
[2]	FITNESS TRAINER APPLICATION USING ARTIFICIAL INTELLIGENCE	Mr. Rutvik Sonawane, Mr. Vaibhav Adke, Mr. Abhijeet Pawar	2023	Image processing and video processing	Workout plans and coorection of posture	Video Processing for real-time analysis and body posture detection and Convolutional Neural Networks (CNN) for deep learning to analyze user movements and provide feedback

Table 1: Summary of Research Paper

2.5 **Problem Statement**

The fitness industry faces significant challenges, particularly for beginners who are dedicated to workouts but lack awareness of the critical role nutrition plays in achieving their goals. This neglect of proper diet not only slows their progress but also poses health risks, including poor recovery, nutritional deficiencies, and the exacerbation of conditions like obesity and diabetes. Additionally, many individuals struggle with tailored strategies to effectively lose weight, gain muscle, or maintain their current fitness levels. Without a holistic approach that integrates both exercise and personalized nutrition, these issues continue to undermine the overall success and well-being of fitness enthusiasts.

Chapter 3: Requirements and Design

Chapter 3: **Requirements and Design**

In this chapter, we have developed the functional requirements for our actors, i.e., **User** and **Admin**. These requirements are specifically designed for the Fitnessstan platform.

Fitnessstan is both a web-based and app-based platform that offers personalized fitness solutions by integrating AI-powered workout and dietary recommendations. The platform is user-friendly, easy to navigate, and provides an efficient way for users to achieve their fitness goals and for admins to manage the platform's functionality seamlessly.

We created system use cases against each functional requirement and designed use case diagrams and fully dressed use cases for our actors, i.e., **User** and **Admin**.

3.1 **Requirements**

3.1.1 **User/Customer Functionalities:**

ID	**Functionality Description**
1	The user must be able to sign up.
2	The user must be able to login to their account.
3	The user must be able to recover and forget their password
4	Input personal information, like DOB, weight, height, gender, religion, sleep-hour, occupation and exercise level, to compute BMI.
5	Diet recommendations that are specifically tailored to the needs of diabetics.
6	Monitoring of progress - calories burned, workouts performed, and nutritional intake.
7	User will be read the article about features and supplements.
8	User shall be able to give feedback about our recommendation.
9	Weekly updates and alerts on progress and fitness plan.

Table 2: User Functional Requirement

3.1.2 **Admin Functionalities:**

ID	Functionality Description
1	Admin must be able to login to the system
2	Administrators will be able to add users.
3	Admin must be able to delete the users.
4	Admin must be able to Manages user by adding, updating, or deleting accounts.
5	Admin must be reviewed the feedback given by user.
6	Monitor the platform's overall performance and usage statistics.
7	Provide data security and impose privacy policies.
8	Manage system configurations and updates and Database management.
9	Resolve technical issues and provide platform support.

Table 3: Admin Functional Requirement

3.2 **Hardware and Software Requirements**

3.2.1 **Hardware Requirement**

1. **Server:**
   A dedicated server or cloud hosting service to host the website and app, ensuring efficient handling of user requests, AI processing, and data storage.
2. **Storage:**
   Adequate storage capacity to store user profiles, workout and dietary data, progress reports, and AI-generated recommendations.
3. **Processing Power:**
   Sufficient processing power to manage concurrent user requests, perform

AI computations for personalized fitness plans, and handle real-time data tracking and updates.

3.2.2 **Software Requirement**

1. **Operating System:**
   The server should run a compatible operating system such as Linux, windows server or macOS server.

2. **React.js:**
   It is a JavaScript library used for creating responsive and dynamic web interfaces.
3. **Spring Boot and Java:**
   These are used for back-end development to ensure scalable and robust server-side functionality.
4. **Python:**
   It is used for AI/ML tasks such as generating personalized fitness and dietary recommendations.
5. **MongoDB:**
   It is a NoSQL database for managing user data, activity logs, and app-related information.
6. **Bootstrap:**
   Used for creating beautiful and responsive user interfaces.
7. **VS Code:**
   Primary IDE for coding and debugging.
8. **Git:**
   A version control system to collaborate on development and manage source code.

3.3 **Proposed Methodology**

Our solution focuses on providing personalized diet plans by leveraging advanced data clustering techniques and synthetic user data generation. To achieve this, we utilize unsupervised learning to cluster food items based on their nutritional values. This process categorizes foods into 14 distinct clusters, each representing specific nutritional characteristics, such as high protein and low fat, or high fiber and moderate carbohydrates. These clusters provide a structured representation of the nutritional profiles of various foods.

To align the dietary recommendations with individual needs, we create a synthetic dataset that simulates user information, including health metrics, dietary preferences, and fitness goals. This synthetic dataset enables the development and testing of personalized diet plans by mapping user requirements to the appropriate food clusters.

By combining nutritional clustering and synthetic user data, our approach ensures tailored, data-driven diet plans that cater to diverse dietary needs and health objectives, promoting sustainable and effective nutrition management.

3.3.1 Methodology Diagram:

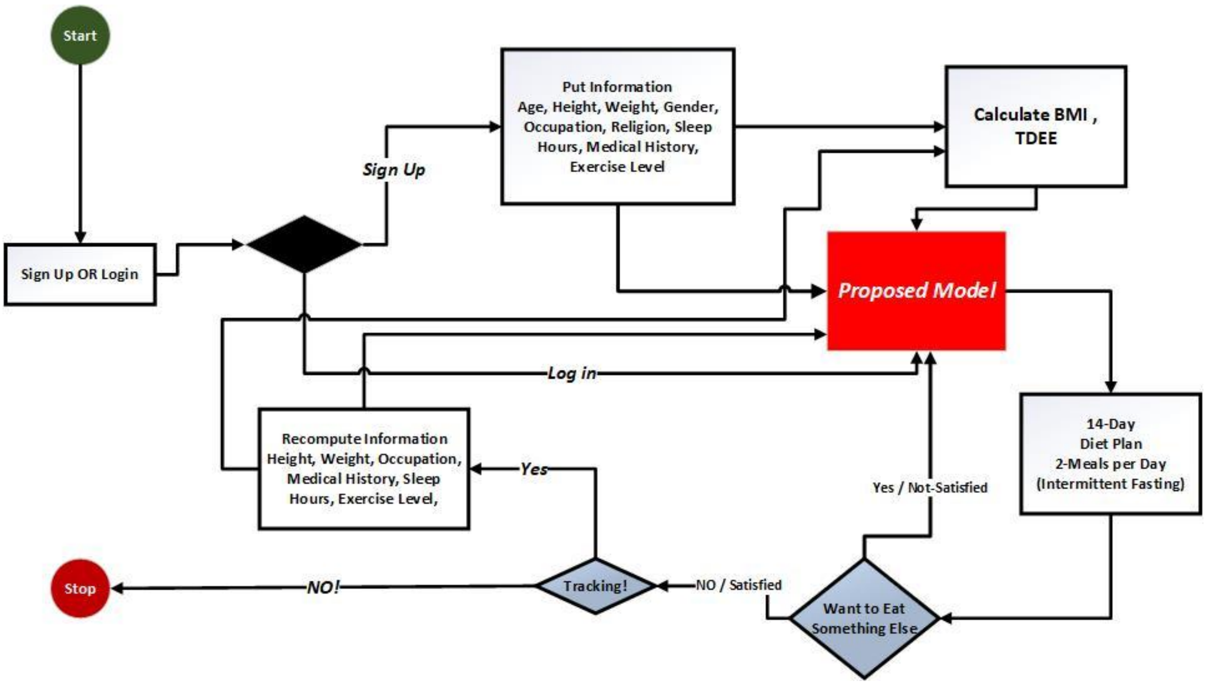

Figure 1: Methodology Diagram

3.3.2 **Proposed Model**

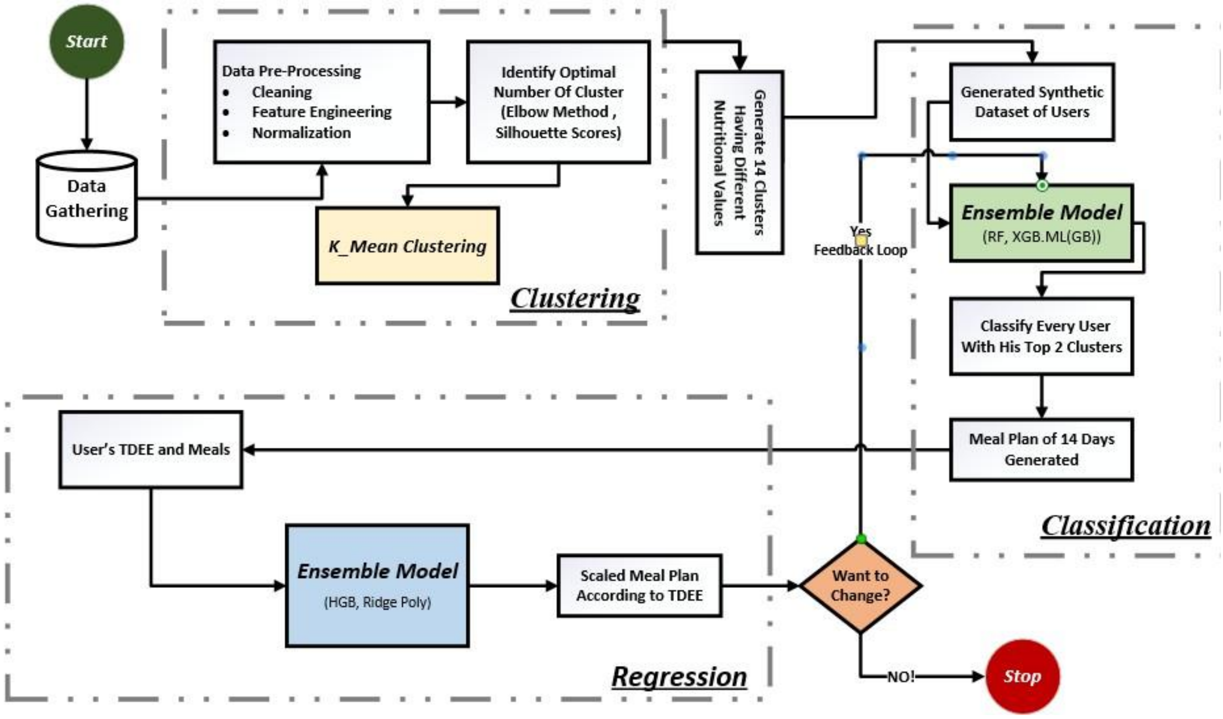
Start
Data Gathering
Data Pre-Processing
• Cleaning
• Feature Engineering
• Normalization
Identify Optimal Number Of Cluster (Elbow Method , Silhouette Scores)
K_Mean Clustering
Clustering
Generate 14 Clusters Having Different Nutritional Values
Generated Synthetic Dataset of Users
Ensemble Model
(RF, XGB.ML(GB))
Classify Every User With His Top 2 Clusters
Meal Plan of 14 Days Generated
Classification
Yes
Feedback Loop
User's TDEE and Meals
Ensemble Model
(HGB, Ridge Poly)
Scaled Meal Plan According to TDEE
Want to Change?
Regression
NO!
Stop

Figure 2: Proposed Model

3.4 System Architecture

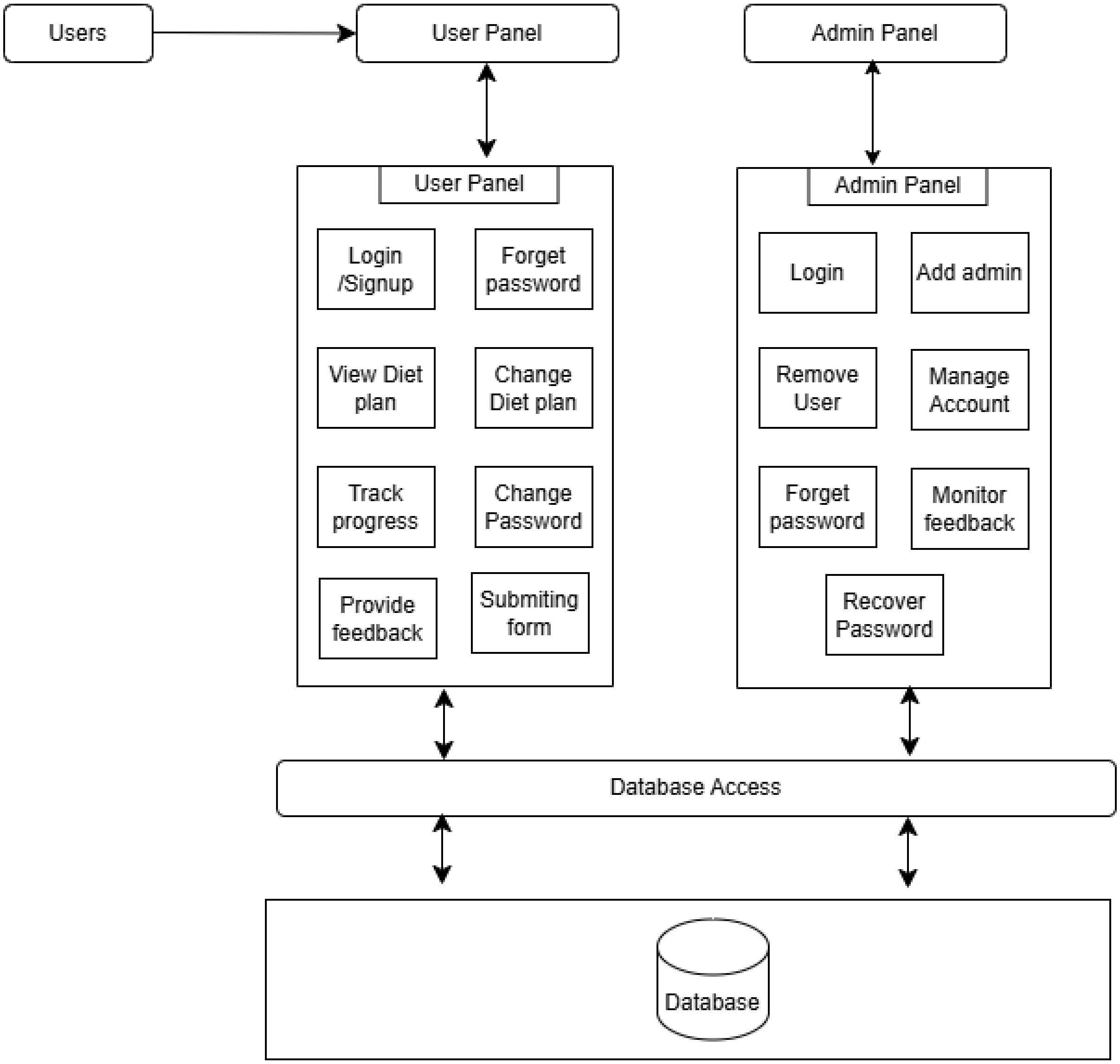

Figure 3: System architecture diagram

3.5 Use Cases

3.5.1 User Use-Case

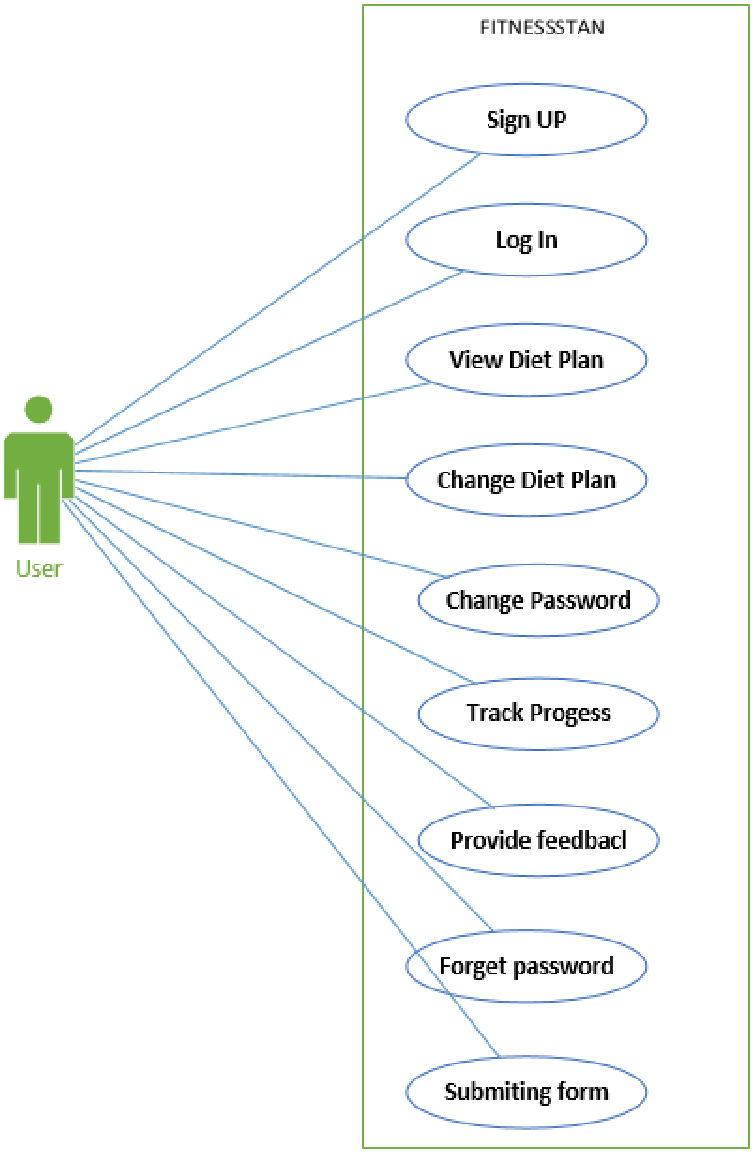

Figure 4: User use case diagram

3.5.2 Admin Use-Case

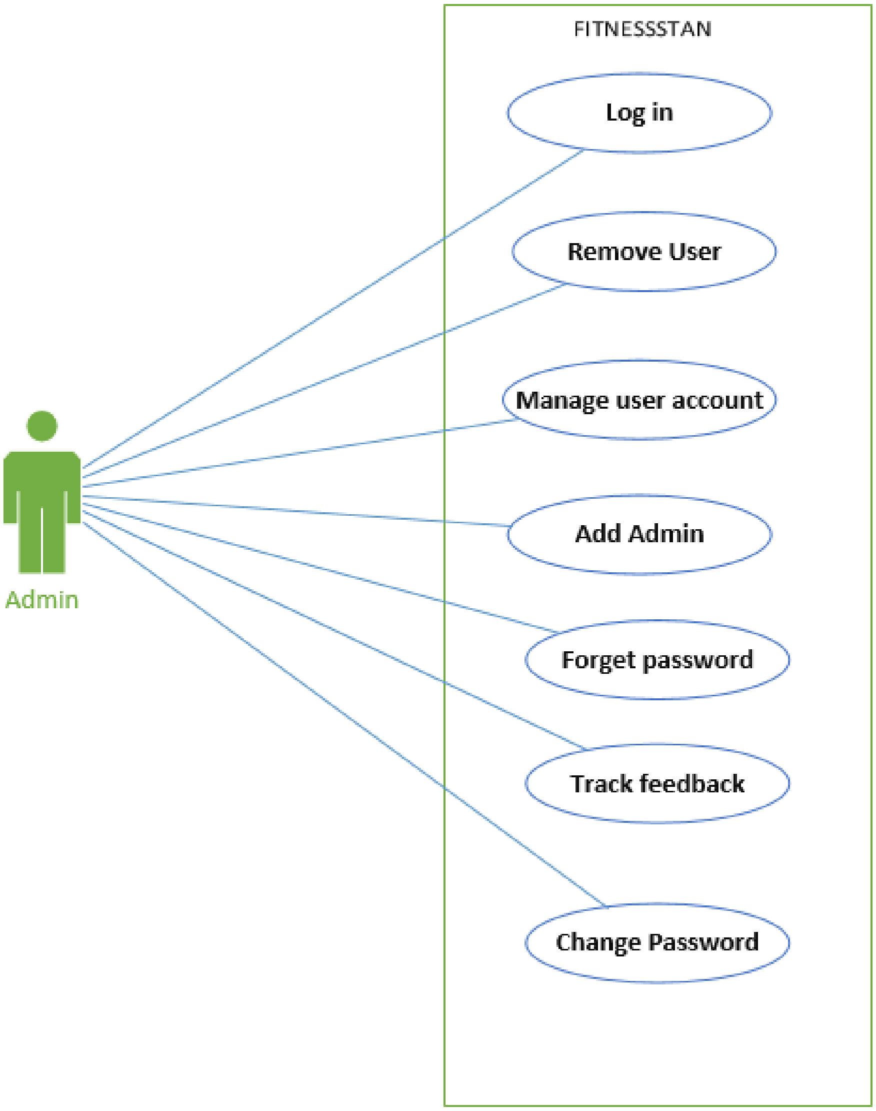

Figure 5: Admin use-case diagram

3.6 **Fully-Dressed Use Cases**

3.6.1 **Login**

Name	Login		
Actors	Admin, User		
Summary	The user must input their email address and password on the login form, following which they will be forwarded to the home page.		
Pre-Conditions	The user must be registered in to the system.		
Post-Conditions	The user will be sent to the system's home page, and his or her session will be preserved.		
Special Requirement	Ensure password is not visible during verification.		
Basic Flow			
Actor Action		System Response	
1	The user navigates to the login page.	2	The login page appears and asking for email address and password.
3	The user provide a valid email and password.	4	The system verify the email and password, establishes a session and redirect to the appropriate dashboard.
Alternative Flow			
A3	The user enters invalid email and password.		The system displays the following error message: incorrect email address and password.

Table 4: Login

3.6.2 **Sign Up**

Name	Sign Up		
Actors	User		
Summary	The user register a new account by entering the required information.		
Pre-Conditions	The user must be registered in to the system.		
Post-Conditions	New account is created and user redirect to the login page for authentication.		
Special Requirement	Validate email format and check the duplication of email		
Basic Flow			
Actor Action		System Response	

1	The user visit the sign-up page.	2	The user is prompted to provide details like their name, email address and password on the sign-up page
3	The user provide their name, email address and password.		
4	The user fills out the sign-up form	5	The system validate the information and send OTP to the user email.
6	The user put the OTP on the sign-up page.	7	The system validate the OTP and generates a new account for user.
		8	The system display a success message, indicating that user account was successfully created.
Alternative Flow			
A3	If the user submit the sign-up form with incomplete or invalid information.		The system display and error message, indicating specific fields that need to be corrected.
A6	If the user enters invalid OTP.		The system display the error message indicating you OTP is incorrect.

Table 5: Sign Up

3.6.3 **Recover password**

Name	Recover password		
Actors	Admin, User		
Summary	The admin and user can recover their password who has forgotten or lost their login credentials.		
Pre-Conditions	The user must have FITNESSSTAN account and user must have access to email address which is associated with his account.		
Post-Conditions	The user password is successfully reset and they can log in to their account using the new password.		
Special Requirement	The reset process must be secure to prevent the unauthorized access.		
Basic Flow			
Actor Action		System Response	
1	The user navigates to the login page.		
2	User click on the "Forgot Password" button.	3	A password recovery form is displayed by the system.
4	In the recovery form user enter the email which is associated with their account.		
5	The user presses the "submit" button	6	The system validate the email address and create the password reset link.

		7	The system send the reset link on the email which is associated with their account.
8	The user open their email and clicks on the password reset link.	9	The system redirect the user to a password reset form.
10	In the prompted field the user enters a new password and verifies it.		
11	The user fill out the password reset form.	12	The new password is validated by the system and added to the user account in database.
		13	When the password is successfully reset, the system display a confirmation message.
Alternative Flow			
A4	If a user provide an incorrect or unregistered email address.		The system prompt the user to input a valid email address which is associated with their account.
			If the password reset link is expired the system prompts the user reset button for again process.

Table 6: Recover password

3.6.4 **View Diet plan**

Name	View diet plan
Actors	User
Summary	The user will view their personalized diet plan which is recommend from ML model on the basis of his input.
Pre-Conditions	The user must be login and navigate to user dashboard and click on the view diet button.
Post-Conditions	The user will view their diet plan of one week.
Special Requirement	None

Basic Flow			
Actor Action		System Response	
1	The user must be login in the system.	2	The System redirect the user on the user dashboard.
3	The user choose an option or view diet on his dashboard.	4	The system will show the diet plan to the User
Alternative Flow			

A3	The user enters invalid email and password.		The system displays the following error message: incorrect email address and password and redirect again on the login page.

Table 7: View Diet plan

3.6.5 **Remove User**

Name	Remove user		
Actors	Admin		
Summary	Admin can remove a user's account permanently from the system.		
Pre-Conditions	The admin must be logged into the system with appropriate permissions and The user account must exist in the database.		
Post-Conditions	The user account will be removed from the system and cannot be recovered.		
Special Requirement	Ensure proper authorization is in place to prevent unauthorized removal of users.		
Basic Flow			
Actor Action		System Response	
1	The admin navigates to the user management section.	2	The system displays a list of all registered users.
3	The admin selects a user and clicks on the "Remove User" button.	4	The system confirms the removal with a confirmation message. Upon confirmation, the user is removed from the system.
Alternative Flow			
A3	The admin attempts to remove a non-existing user. .		The system displays an error message: "User does not exist."

Table 8: Remove User

3.6.6 **Manage User account**

Name	Manage User Account
Actors	Admin
Summary	Admin can update or modify user account details such as email.
Pre-Conditions	The admin must be logged into the system with appropriate permissions and The user account must exist.

Post-Conditions	The changes to the user account are saved and reflected in the system.		
Special Requirement	Ensure proper input validation and role-based access controls.		
Basic Flow			
Actor Action		System Response	
1	The admin navigates to the user management section.	2	The system displays a list of registered users.
3	The admin selects a user and edits the account details	4	The system validates the changes and updates the user account successfully. .
		5	The system send update details notifications to the User email.
Alternative Flow			
A3	The admin enters invalid input (e.g., an email in an incorrect format).		The system displays an error message: "Invalid input. Please correct the errors."

Table 9: Manage User account

3.6.7 **Track Feedback**

Name	Track Feedback		
Actors	Admin		
Summary	Admin can view feedback submitted by users and monitor trends or issues.		
Pre-Conditions	The admin must be logged into the system. Feedback must exist in the database.		
Post-Conditions	Feedback is displayed and may be marked as addressed or unresolved		
Special Requirement	Provide filtering options for easier navigation of feedback		
Basic Flow			
Actor Action		System Response	
1	The admin navigates to the feedback tracking section.	2	The system displays a list of user feedback with relevant details.
3	The admin selects a feedback to view details or mark it as addressed	4	The system updates the feedback status accordingly.
Alternative Flow			
A3	The admin searches for feedback but none exists.		The system displays a message: "No feedback available."

Table 10: Track Feedback

3.6.8 **Manage content**

Name	Manage Content		
Actors	Admin		
Summary	Admin can create, update, or delete content displayed on the platform.		
Pre-Conditions	The admin must be logged into the system.		
Post-Conditions	Content changes are saved and displayed on the platform		
Special Requirement	Ensure preview functionality is available before saving changes.		
Basic Flow			
Actor Action		System Response	
1	The admin navigates to the content management section.	2	The system displays existing content items.
3	The admin creates new content or updates/deletes existing content.	4	The system validates the action and saves the changes.
Alternative Flow			
A3	The admin does not have permission to update/delete the content		The system displays a warning message: "you don't have access to update/ delete the content."

Table 11: Manage content

3.6.9 **Change diet plan**

Name	Change diet plan		
Actors	User		
Summary	User will change his diet plan or specific diet if he/she do not like that diet meal.		
Pre-Conditions	The user must be login and must have already diet plan		
Post-Conditions	Diet meal is change according to user choice.		
Special Requirement	None		
Basic Flow			
Actor Action		System Response	
1	User must be login with correct credentials.	2	The system validate and redirect to the User dashboard.
2	The user navigate to the diet plan section on the dashboard.	3	The system displays his/her existing diet meal and also give option to change diet meal.

4	The user click on the option of change diet meal.	5	The system analyze and give meals option according to user choice.
6	The user select the meal according to his/her choice.	7	The system update the meal in his diet plan of one week.
Alternative Flow			
A1	If the user input the incorrect email and password.		The system displays a error message and ask “enter again email and password”

Table 12: Change diet plan

3.6.10 **Change Password**

Name	Change password		
Actors	user		
Summary	The user can change his/her password.		
Pre-Conditions	The user must be go to change password form		
Post-Conditions	Password is changed		
Special Requirement	None		
Basic Flow			
Actor Action		System Response	
1	Click on the change password button	2	System move to the change password form
3	User Enter the current password		
4	User enter the new password		
5	User enter the new password in the conform password field and click on the change password button	6	System validate the requirement of the password and display password is changed
Alternative Flow			
A3	The user enter the wrong password		The system show current password is incorrect.
A4	The user enter new password < 6 characters		The system show error password must be greater than 6 characters.

Table 13: Change password

3.6.11 **Tracking calories**

<table>
<tr><td>Name</td><td colspan="3">Tracking calories</td></tr>
<tr><td>Actors</td><td colspan="3">user</td></tr>
<tr><td>Summary</td><td colspan="3">The user must track his calories through the system how much he/she intake calories.</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">The admin must be logged into the system.</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">View the calories intake in a day by the user.</td></tr>
<tr><td>Special Requirement</td><td colspan="3">None</td></tr>
<tr><th colspan="4">Basic Flow</th></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1</td><td>The user must be login.</td><td>2</td><td>The system validate and redirect to the user dashboard.</td></tr>
<tr><td>3</td><td>The user move to dashboard and click the “tracking calories” option.</td><td>4</td><td>The system shows the calories tracking meter where his/her calories intake are stored.</td></tr>
<tr><th colspan="4">Alternative Flow</th></tr>
<tr><td>A3</td><td>The user cannot register diet plan.</td><td></td><td>The system display a message must be register for diet plan.</td></tr>
</table>

Table 14: Tracking calories

3.6.12 **Give feedback**

<table>
<tr><td>Name</td><td colspan="3">Give feedback</td></tr>
<tr><td>Actors</td><td colspan="3">User</td></tr>
<tr><td>Summary</td><td colspan="3">The user give feedback about the system progress and diet suggestion or any problem they have.</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">The user must be logged into the system.</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">Feedback is successfully submitted now is visible for everyone.</td></tr>
<tr><td>Special Requirement</td><td colspan="3">None</td></tr>
<tr><th colspan="4">Basic Flow</th></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1</td><td>The User access to the FITNESSSTAN website or app and login to their account.</td><td>2</td><td>The system validate the user credentials and redirect to the user dashboard.</td></tr>
</table>

3	The User have multiple option and choose the “feedback” option.	4	The system redirect user to the feedback page.
5	The user select the option of post feedback.	6	The system present a feedback form to the user.
7	User enters their feedback in the provided text area.		
8	User click on the “Submit” or “post” button to publish the feedback.	9	The system adds the feedback to the feedback section.
		10	The System shows a notification conforming that the feedback was successfully posted.
Alternative Flow			
A7	If the user enters inappropriate or abusive feedback.		Then the system may have implemented AI sentiments to identify and remove the abusive feedback.

Table 15: Give feedback

3.6.13 **Submitting form**

Name	Submitting form		
Actors	User		
Summary	The user submit the form about his details like: weight, height, age etc.		
Pre-Conditions	The user must be logged into the system.		
Post-Conditions	Form is successfully submitted.		
Special Requirement	None		
Basic Flow			
Actor Action		System Response	
1	The User access to the FITNESSSTAN website or app and login to their account.	2	The system validate the user credentials and redirect to the user dashboard.
		3	The system show the form page.
4	The User answers the all questions which is in the form.		
5	The User click the submit button to submit the form	6	The system shows the data is successfully submit.
Alternative Flow			

A2	The user does not give answer of all question.		Then the system shows a red alert to show the user answer that question.

Table 16: Submitting form

3.7 **Database Schema Diagram**

3.7.1 **User Collection**

Field Name	**Type**	**Required?**	**Description**	**Indexes**
_id	ObjectId	Yes	Primary key	_id (default)
username	String	Yes	User's username	
email	String	Yes	User's email address	
password	String	Yes	User's password (hashed)	
roles	Array	Yes	Array of user roles (e.g., ADMIN, USER)	
status	String	Yes	User's status (e.g., PASS, FAIL)	
createdAt	Date	Yes	Timestamp of user account creation	
updatedAt	Date	Yes	Timestamp of last account update	
heightFt	Double	Yes	User's height in feet	
dob	Date	Yes	User's date of birth	
weightKg	Double	Yes	User's weight in kilograms	
gender	String	Yes	User's gender (e.g., Male, Female)	
occupation	String	Yes	User's occupation	

religion	String	Yes	User's religion	
exerciseLevel	String	Yes	User's exercise level (e.g., 6 days a week)	
sleepHours	Double	Yes	Average sleep hours per night	
medicalHistory	Array	Yes	Array of medical history (e.g., None, Diabetic)	
bmi	Double	Yes	User's body mass index	
ree	Double	Yes	Resting energy expenditure (REE)	
tdee	Double	Yes	Total daily energy expenditure (TDEE)	

Table 17: User Collection Schema

3.7.2 **Diet Collection**

Field Name	**Type**	**Required?**	**Description**	**Indexes**
_id	ObjectId	Yes	Primary key	_id (default)
user	ObjectId	Yes	Reference to the user in the users collection	
mealPlan	Object	Yes	Object containing meal plans for each day (1-14)	
startDate	Date	Yes	Diet plan start date	
endDate	Date	Yes	Diet plan end date	

Table 18: Diet Collection Schema

3.7.3 **Exercises Collection**

Field Name	**Type**	**Required?**	**Description**	**Indexes**
_id	ObjectId	Yes	Primary key	_id (default)
name	String	Yes	Exercise name	
muscleGroup	String	Yes	Target muscle group (e.g., Core, Legs)	
gifUrl	String	Yes	Path to the GIF representing the exercise	
description	String	Yes	Short description of the exercise	
equipment	String	Yes	Equipment used for the exercise (e.g., body weight, dumbbells)	

Table 19: Exercise Collection Schema

3.7.4 **Feedback Collection**

Field Name	**Type**	**Required?**	**Description**	**Indexes**
_id	ObjectId	Yes	Primary key	_id (default)
name	String	Yes	Name of the person giving feedback	
email	String	Yes	Email address of the person providing feedback	
feedback	String	Yes	The actual feedback text	
submittedAt	Date	Yes	Timestamp of when feedback was submitted	

Table 20: Feedback Collection Schema

3.7.5 **Workout Plan Collection**

Field Name	**Type**	**Required?**	**Description**	**Indexes**
_id	ObjectId	Yes	Primary key	_id (default)
user	ObjectId	Yes	Reference to the user in the users collection	
planName	String	Yes	Name of the workout plan	
startDate	Date	Yes	Start date of the workout plan	
endDate	Date	Yes	End date of the workout plan	
dayPlans	Array	Yes	Array of day plans, each containing a day number and exercises for that day	
- dayNumber	Integer	Yes	Day number (1-14)	
- exercises	Array	Yes	Array of exercises (references to exercises collection)	

Table 21: Workout Plan Collection schema

Chapter 4: Implementation and Test Cases

Chapter 4: **Implementation and Test Cases**

4.1 **Implementation:**

We have successfully implemented key components of the website, model, and app. The website includes signup and login with role-based authentication (user/admin) and a user dashboard. The model uses clustering for dataset segmentation and classification for insights. The Fitnesstan app, built with Flutter, features a multi-screen interface with navigation, role-based authentication, and a dashboard showcasing workouts, diet plans, and sleep tracking, styled with a cohesive red-black gradient theme and interactive elements.

4.1.1 **Implementation of First Component/Algorithm:**

4.1.1.1 **Web Development:**

Platform: The website is built using React, leveraging its component-based architecture for seamless development and scalability.

Homepage Design:

- **Introduction Section**: The homepage features a visually appealing layout with an introduction to the application accompanied by an aesthetically pleasing image on the right side.
- **Features Section**: Includes an introduction to the application's key features, followed by a button to navigate to the detailed Features Page.
- **Supplements Section**: A similar section introduces the supplements offered, with a button directing users to the dedicated Supplements Page for further details.
- **Admin-Approved Content Section**: A section displays content specifically approved by the admin for visibility on the homepage.
- **Team Collaboration Section**: Showcases the teamwork and collaboration behind the platform, highlighting contributors.

- **Feedback Section**: Provides a dedicated area where users can submit their feedback, enabling interaction and improvement.

Footer: A professionally designed footer credits contributors and acknowledges the work behind the platform, ensuring completeness and branding.

Navigation:

- **Sticky Navbar**: The website features a sticky navigation bar on the front page for improved usability and accessibility.
- **React Router DOM**: Well-defined routes seamlessly connect pages, including the Features Page, Supplement Page, Registration Page, Additional Info Page, and Email Verification Page.

Interactive Buttons:

- **Back to Top Button**: A custom-designed button smoothly transitions the user back to the top of the page.
- **Back to Home Button**: A functional button allows users to return directly to the homepage, enhancing navigation and user experience.

Authentication:

- A robust authentication system is implemented, enabling secure user signup and login functionalities.

Admin Panel:

- **Sidebar Design**: The admin panel replaces the standard navbar with a sidebar layout for better navigation and improved workspace efficiency. The sidebar serves as a quick-access menu, streamlining administrative tasks.
- **Theme**: Styled with a dark grey primary color and strong red accents as the secondary color, ensuring a visually appealing and professional look.
- **Functionality**: The admin panel includes tools and sections for managing website content, user data, and administrative settings.

Visual Theme: The entire website uses a dark grey primary and vibrant red accents for a modern, cohesive, and energetic design theme.

User Engagement: The interface is designed with a focus on usability and interactivity, ensuring an intuitive and smooth user journey across all pages and features.

4.1.1.2 **Algorithm:**

In algorithm we use the ensemble technique in which we use 3 types of algorithm that are k-mean clustering for making clusters and random forest for classification and multiple linear. Regression but know we implement the 2 model k-mean clustering and Random forest algorithm.

- **Data Collection:**

  Data collection began with the food nutrition dataset which is sourced from USDA food database in the dataset there is a lot of food items and their corresponding nutrition values.

- **Data Preprocessing:**

  Preprocessing involved cleaning and normalization the dataset, addressing issues like missing values, normalize the serving size or irrelevant data. This steps helps the model for best results.

- **Model training**:

  We give the clean dataset to the model and we do clustering by using k mean clustering and make 14 cluster and each cluster is labeled for example high protein cluster, high carbs cluster etc. and then we use synthetic data of users to train the model Random forest for classification.

- **Trained Model:**

  The final trained model is now implemented in which take a input from the user like weight, height, dob, religion, occupation, disease and then on the basis of this we calculate the BMI and on the basis of it we classify the user belongs to which cluster.

4.2 Test Case Design and description

4.2.1 Test Data

Test Data	TD-1
Form	Signup Form
Stakeholder	User
Field	Full Name
Technique	Equivalence Class Partitioning (ECP)
Valid	{A…..Z, a……z}
Invalid	{1, 2, 3…..} {@,# and other special characters}

Table 22: Test Data Registration

Test Data	TD-2
Form	Signup Form
Stakeholder	User
Field	Email
Technique	Equivalence Class Partitioning (ECP)
Valid	{a….z, A…..Z,1,2,3,…….@gmail.com}, { a….z, A…..Z,1,2,3….,@students.riphah.edu.pk}
Invalid	user@invalid.com, multiple '@' characters and not ending with numbers like 1,2,3,4.

Table 23: Test Data Registration

Test Data	TD-3
Form	Signup Form
Stakeholder	User
Field	Password
Technique	Equivalence Class Partitioning (ECP)
Valid	Password length>==6 character{ { A…..Z, a…..z}, {1,2,3…9}} Special characters
Invalid	Password length <6

Table 24: Test Data Registration

Test Data	TD-4
Form	Login
Stakeholder	User/Admin
Field	Email
Technique	Equivalence Class Partitioning (ECP)
Valid	{a….z, A…..Z,1,2,3,…….@gmail.com}, { a….z, A…..Z,1,2,3….,@students.riphah.edu.pk}
Invalid	user@invalid.com, multiple '@' characters and not ending with numbers like 1,2,3,4.

Table 25: Test Data Login

Test Data	TD-5
Form	Login
Stakeholder	User
Field	Password
Technique	Equivalence Class Partitioning (ECP)
Valid	Password length>==6 character{ { A…..Z, a…..z}, {1,2,3…9}} Special characters
Invalid	Password length <6

Table 26: Test Data Login

Test Data	TD-6
Form	Additional information
Stakeholder	User
Field	Date of Birth
Technique	Equivalence Class Partitioning (ECP)
Valid	Month/date/year
Invalid	Date/month/year, valid numbers

Table 27: Test Data Additional information

Test Data	TD-6
Form	Feedback
Stakeholder	User
Field	Post feedback
Technique	Equivalence Class Partitioning (ECP)
Valid	{A…..Z, a…..z}, {12, 3…..}
Invalid	Empty input

Table 28: Test Data feedback

4.2.2 **Test Case**

ID	TC-1
Form	Registration
Stakeholder	User
Field	Full name
Date	2/04/25
Input	Obaid Ullah
Partition Tested	{A…..Z,a….z}

Expected Output	OK
Test case Passed	

Table 29: Test Case Registration

ID	**TC-2**
Form	Registration
Stakeholder	User
Field	Full name
Date	2/04/25
Input	1234
Partition Tested	{A…..Z,a….z,_}
Expected Output	Error
Test case Passed	

Table 30: Test Case Registration

ID	**TC-3**
Form	Registration
Stakeholder	User
Field	Email
Date	2/04/25
Input	Obaidullah7601@gmail.com
Partition Tested	Contain ‘@’ along with a, b,….z, A, B,C,….Z 1,2,3,….before’@’.,_ before the letter ‘@’ at the end following those three to four alphabetic letters
Expected Output	OK
Test case Passed	

Table 31: Test Case Registration

ID	**TC-4**
Form	Registration
Stakeholder	User
Field	Email
Date	2/04/25
Input	Obaidullah760gmail.com

Partition Tested	Contain ‘@’ along with a, b,….z, A, B,C,….Z 1,2,3,….before’@’.,_ before the letter ‘@’ at the end following those three to four alphabetic letters
Expected Output	Error
Test case Passed	

Table 32: Test Case Registration

ID	**TC-5**
Form	Registration
Stakeholder	User
Field	Password
Date	2/04/25
Input	Zain_12345
Partition Tested	Password length >=6 character {A……Z,a…..Z} {1,2,3…..9} special characters
Expected Output	OK
Test case Passed	

Table 33: Test Case Registration

ID	**TC-6**
Form	Registration
Stakeholder	User
Field	Password
Date	2/04/25
Input	Zain
Partition Tested	Password length >=6 character {A……Z,a…..Z} {1,2,3…..9} special characters
Expected Output	error
Test case Passed	

Table 34: Test Case Registration

ID	**TC-7**
Form	Login
Stakeholder	User
Field	Email
Date	5/04/25
Input	Obaidullah7601@gmail.com

Partition Tested	Contain ‘@’ along with a, b,….z, A, B,C,….Z 1,2,3,….before’@’.,_ before the letter ‘@’ at the end following those three to four alphabetic letters
Expected Output	OK
Test case Passed	

Table 35: Test case Login

ID	**TC-8**
Form	Login
Stakeholder	User/Admin
Field	Email
Date	5/04/25
Input	Obaidullah7601gmail.com
Partition Tested	Contain ‘@’ along with a, b,….z, A, B,C,….Z 1,2,3,….before’@’.,_ before the letter ‘@’ at the end following those three to four alphabetic letters
Expected Output	Invalid Email
Test case passed	

Table 36: Test Case Login

ID	**TC-9**
Form	Login
Stakeholder	User/Admin
Field	Email
Date	5/04/25
Input	Obaidullah7601@gmail.com
Assume	User does not register
Partition Tested	Contain ‘@’ along with a, b,….z, A, B,C,….Z 1,2,3,….before’@’.,_ before the letter ‘@’ at the end following those three to four alphabetic letters
Expected Output	Error (User Does not Exist)
Test case passed	

Table 37: Test Case Login

ID	**TC-10**
Form	Feed back
Stakeholder	User
Field	Post feedback
Date	9/04/25

Input	I Like this platform and also I feel changes in my health
Partition Tested	{A…..Z, a…..z}, {12, 3…..}
Expected Output	Feedback posted successfully
Test case Passed	

Table 38: Test Case Feedback

ID	**TC-11**
Form	OTP
Stakeholder	User
Field	Enter OTP
Date	14/04/25
Pre-Condition	OTP send on Email
Input	456445
Partition Tested	{1, 2, 3…}
Expected Output	User register successfully
Test case Passed	

Table 39: Test Case OTP

ID	**TC-12**
Form	OTP
Stakeholder	User
Field	Enter OTP
Date	14/04/25
Pre-Condition	OTP send on Email
Assume	User enter invalid OTP
Input	313412
Partition Tested	{1, 2, 3…}
Expected Output	Incorrect OTP
Test case Passed	

Table 40: Test Case OTP

ID	**TC-13**
Form	Additional Form
Stakeholder	User
Field	Enter Information
Date	18/04/25

Input	Height= 5.6 Date of birth=03/28/2003 Occupation= student Gender=Male Muslim no exercise 9 hours
Partition Tested	Built in choose by user
Expected Output	Enter data successfully
Test case Passed	

Table 41: Test case add information

ID	TC-14
Form	Calculating BMI,TDEE and REE
Stakeholder	User
Field	Showing BMI , TDEE and REE
Date	18/04/25
Input	Height= 5.6 Date of birth=03/28/2003 Occupation= student Gender=Male Muslim no exercise 9 hours
Partition Tested	Built in choose by user
Expected Output	BMI = 51, TDEE = and REE=

Test case Passed

Table 42: Test Case calculating

ID	**TC-15**
Form	Calculating BMI,TDEE and REE
Stakeholder	User
Field	Showing BMI , TDEE and REE
Date	18/04/25
Input	Height= 5.6 Date of birth=01/18/2004 Occupation= student Gender=Male Muslim 6 days exercise 8hours
Partition Tested	Built in choose by user
Expected Output	BMI = 22, TDEE = and REE =
Test case Passed	

Table 43: Test Case calculating

ID	**TC-16**
Use case	Making diet plan

Stakeholder	User
Date	23/04/25
Input	Height= 5.6 Date of birth=03/28/2003 Occupation= student Gender=Male Muslim no exercise 9 hours
Partition Tested	Built in choose by user
Expected Output	Model make a 14 days plan successfully
Test case Passed	

Table 44: Test Case making Diet

bo

ID	**TC-17**
Use case	Change meal
Stakeholder	User
Date	26/04/25
Pre-condition	Already diet plan exist
Input	Click on change meal
Expected Output	Model change the meal according to his/her need
Test case Passed	

Table 45: Test Case Change meal

4.3 **Test Metrics**

4.3.1 **Test case Matric**

Metric	Purpose
Number of Test Cases	22
Number of Test Cases Passed	22
Number of Test Cases Failed	0
Test Case Defeat Density	0
Test Case Effectiveness	0*100
Traceability Matrix	Traceability refer to the ability to determine that each feature has a source in requirements and that each requirement has a matching implemented feature.

Table 46: Test Case Matric

Chapter 5:
Experimental Result & Analysis

Chapter 5: **Experimental Result and Analysis**

5.1 **Introduction**

This chapter is all about experiments and their result analysis so we are going to present the experimental setup, results and analysis of our website "Fitnessstan". We aim in this evaluation to validate the fitnessstan effectiveness when connecting a user with the website and its performance of recommendation system the experiments that we are going to discuss are designed for technical and user-centric metrics which will provide a comprehensive overview of fitnessstan capabilities and will also discuss about areas for improvement.

5.2 **Experiments Analysis**

In the Fitnesstan our model is divided into three phases Clustering, Classification and regression.

5.2.1 **Clustering**

5.2.1.1 **Experimental Setup**

In the first phase, we initially use a nutrition dataset from Kaggle, which contains 8,789 instances and 79 features. We then perform data preprocessing, including normalization, feature engineering, and removal of irrelevant features. After preprocessing, the dataset contains 8,789 instances and 25 features.

After preprocessing, we aim to create clusters from the dataset and determine the optimal number of clusters for Clustering. To find the optimal number of clusters, we use two methods: the **Elbow Method** and the **Silhouette Score**.

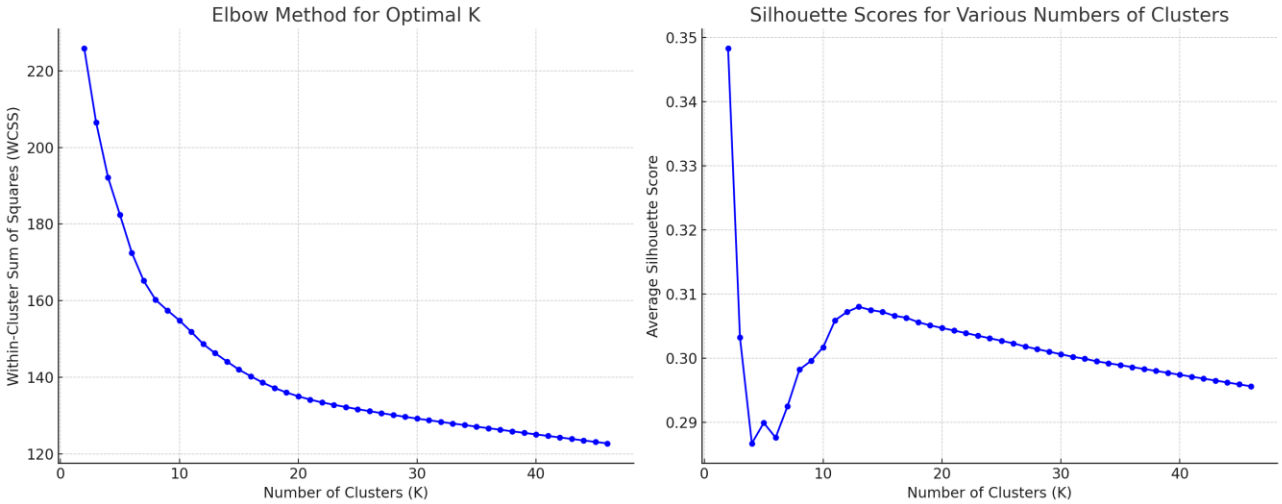

Figure 6: Evaluate number of cluster

Both the **Elbow Method** and **Silhouette Score** indicate that the optimal number of clusters is **K = 14**. The Elbow Method shows a significant reduction in WCSS at K = 14, while the Silhouette Score is highest at this value, suggesting the best cluster separation and cohesion.

5.2.1.2 **Clustering Model Training:**

- **Data Input**: The dataset used in clustering contains 8,789 instances and 25 features after preprocessing.
- **Algorithm Execution**: Once the optimal *K* is determined using the Elbow Method and Silhouette Score, the **K-Means algorithm** is executed on the dataset to perform the clustering.
- **Model Output**: The output of the model will be a set of clusters, each containing instances that are similar to each other based on the feature set used for clustering.

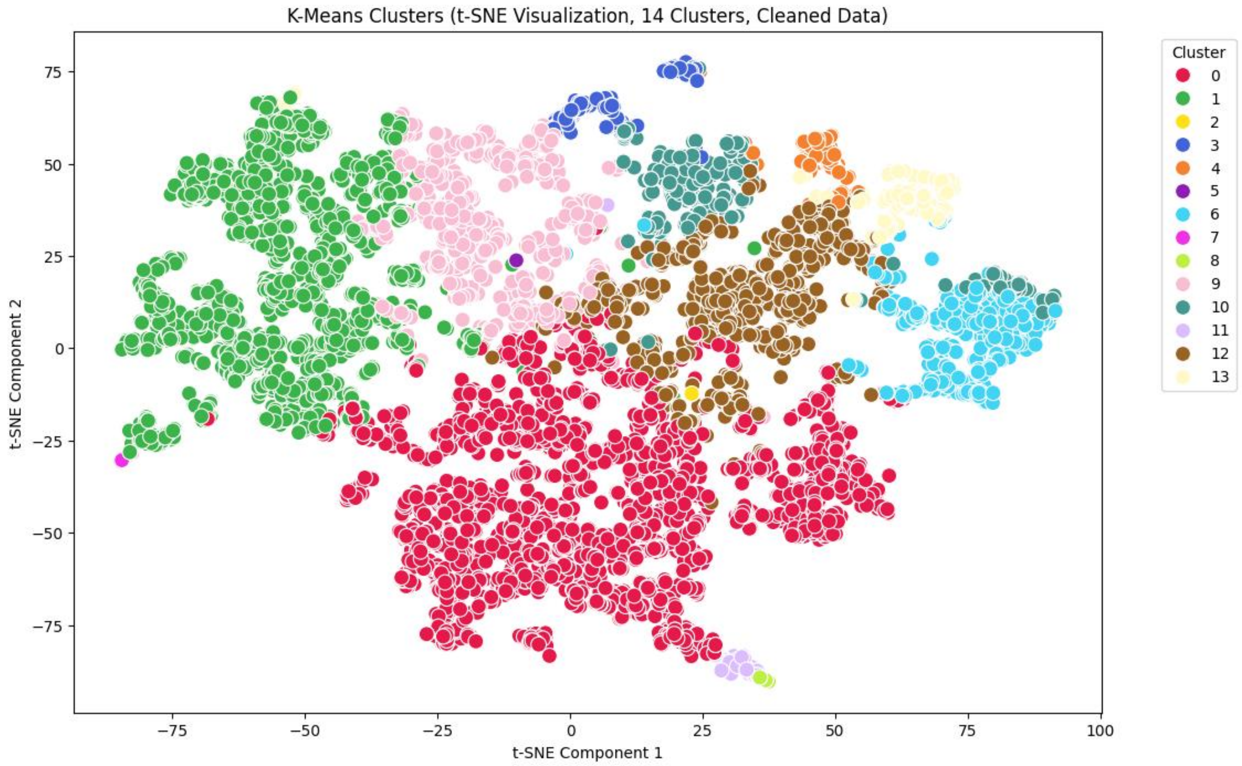

Figure 7: Clustering

This t-SNE visualization shows the K-Means clustering results of the dataset with 14 clusters. Each data point is assigned to one of the 14 clusters, represented by different colours, and the plot projects the high-dimensional data onto two dimensions for easier interpretation. The clusters appear to be well-separated, indicating that the data points within each cluster share similar characteristics, which could be useful for further analysis in the Fitnessstan recommendation system.

5.2.2 **Classification**

5.2.2.1 **Experimental Setup**

In the classification phase first we make a synthetic dataset of different users to train the classification Model. First we make condition or data of different users with the help of expert then we used that dataset and we use CTGAN model which is used for to make synthetic dataset and we train on the dataset and after training the CTGAN model we make a synthetic dataset of 20000 users and after making the dataset we split dataset 80% for training and 20% for testing.

5.2.2.2 Classification Model Training:

After creating the synthetic dataset, we proceeded to train classification models using four different algorithms: Random Forest, XGBoost, AdaBoost, and SVM. Initially, we evaluated these models, and their classification accuracies are shown in the figure.

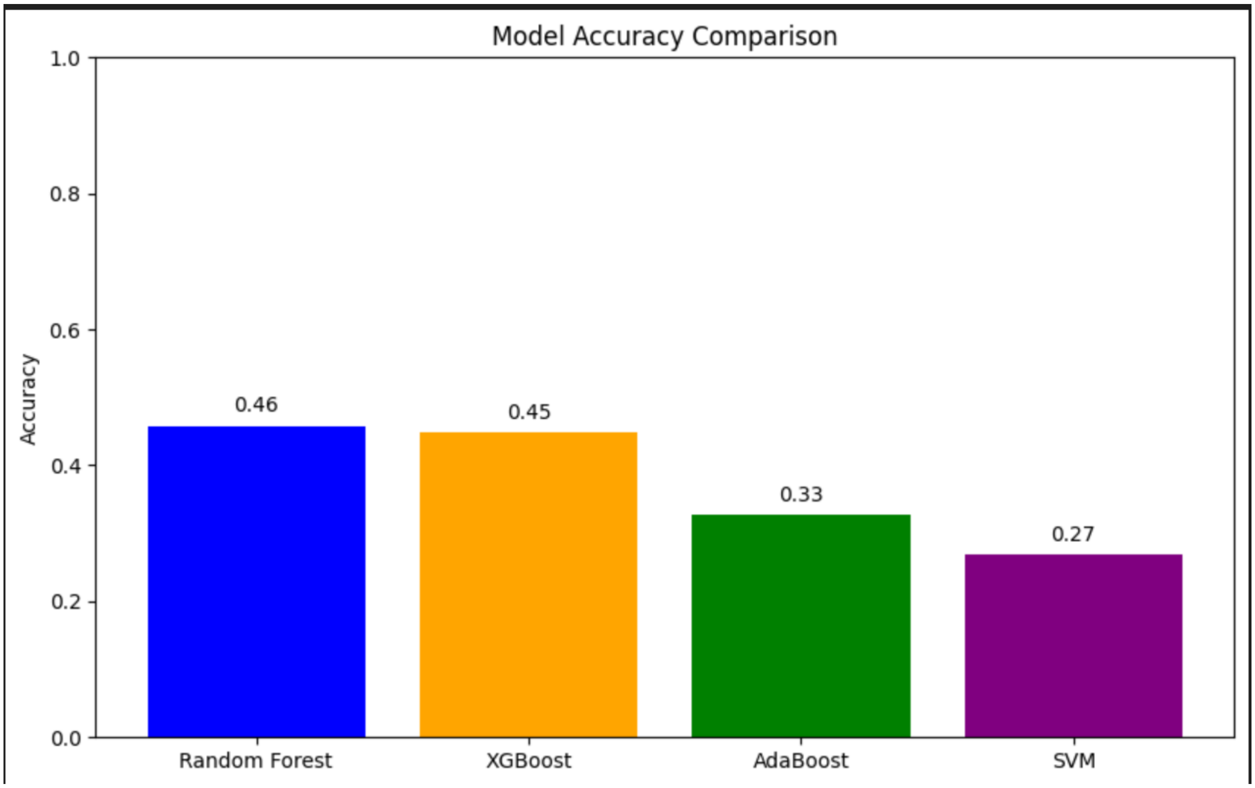

Figure 8: Comparison of Models

After the first phase of training, we selected the top two models with the highest accuracy: Random Forest and XGBoost. In the second phase of classification training, we improved the dataset and performed hyper parameter tuning on both models to enhance classification accuracy. As a result, Random Forest achieved an accuracy of 60%, while XGBoost reached 49%.

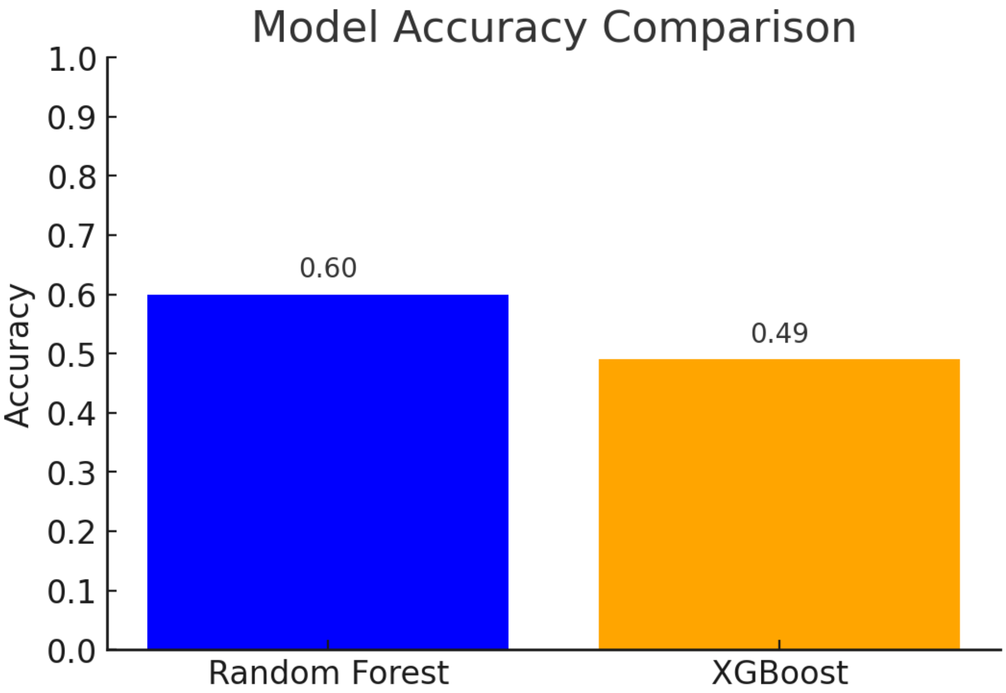

Figure 9: Models accuracy

In the third phase, to further improve classification accuracy, we applied an ensemble technique by combining both Random Forest and XGBoost models. Additionally, we used Gradient Boosting as a meta-learner, along with hyper parameter optimization and k-fold cross-validation to enhance overall performance. As a result, we achieved 72% accuracy for the primary cluster and 76% accuracy for the secondary cluster.

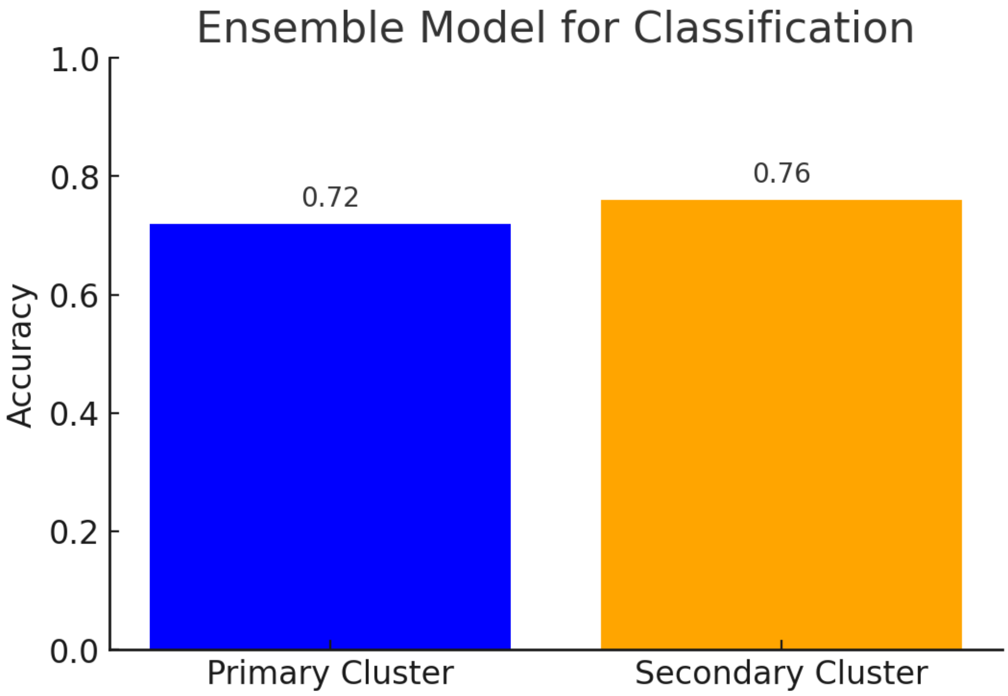

Figure 10: Classification Final Model Accuracy

5.2.3 **Regression**

5.2.3.1 **Experimental Setup**

This experiment start with loading the dataset from CSV file which contained food nutrition data such as protein, carbohydrates, fats and calories etc. The data was cleaned and pre-processed by removing non-numeric values and missing values filled with zeros. The data was split into training 80% and testing 20% to ensure the models could be evaluated on unseen data. These preprocessing steps ensured that the dataset was clean, structured, and ready for model training.

5.2.3.2 **Regression Model training**

In the model training phase we use two different machine learning models that are HistGradientBoostingRegressor (HGB) and ridge regression with polynomial features. To improve the performance, the predictions from both models were combined in an Ensemble Model, where the predictions from both the **HGB** and **Ridge** models were averaged. The Ensemble Model delivered the best results, with the lowest RMSE and MAE, and the highest R² value. Below is the comparison of the results from each model:

Models	**RMSE**	**MAE**	**R²**
HGB	13.80	7.36	0.993
RIDGE	12.12	6.19	0.995
BLEND	11.78	6.10	0.995

Figure 11: Regression model evaluation

5.3 **Conclusion**

After completing the experiments, we obtained promising results across all three phases: clustering, classification, and regression. We then integrated these models to implement the core functionality of FITNESSSTAN, which delivers a personalized 14-day diet plan tailored to the user's preferences. The integration was successful, and the system produced effective results. Now if new users come on the Fitnessstan they put his/her information our model classify according to his/her information and assign the two clusters where they belong and after that our regression model assign the make a diet for 14 days according to his/her TDEE.IF they don’t like meal there is an option to change the meal and our model took the meal from cluster and give to the user.

Chapter No 6

Conclusion and future Direction

Chapter 6: **Conclusion and Future Directions**

6.1 **Introduction**

Fitnessstan is a smart fitness platform that keeps users fit through customized diet plans and workout programs. The platform utilizes artificial intelligence to learn the individual needs and goals of users. The interface of the platform is user-friendly and supports both web and mobile applications. Notably, Fitnessstan is highly suitable for diabetes patients since it offers customized meal plans. Furthermore, the platform offers calorie tracking, user input, and progress tracking. In general, the goal is to make it easy, safe, and effective to handle fitness for everyone. By using technological innovation combined with user input, Fitnessstan provides an improved solution to fitness and health management.

6.2 **Achievements**

Fitnessstan has made significant progress in providing a comprehensive fitness platform for users. The AI-powered system successfully delivers personalized diet and exercise plans, tailored to individual health profiles. The meal recommendation system offers a 14-day plan that adapts to user progress and health metrics. For users with specific needs, such as diabetics, customized nutrition guidance is offered. The platform's calorie tracking and progress monitoring tools ensure users stay informed. Flexibility is incorporated with the Change feature, allowing users to modify meal plans. These achievements collectively enhance the user experience and support the platform's mission to promote healthier lifestyles.

6.3 **Improvements and Future Directions**

Improvements and Future Directions: Fitnessstan will focus on making the platform more user-friendly. This includes enhancing the user interface for a smoother and more intuitive navigation experience, making it easier for users to interact with the app and access their personalized fitness plans. Moving forward, Fitnessstan will implement dockerization to improve deployment, scalability, and ease of maintenance. The platform will also add multi-language support, allowing it to serve a broader, more diverse user base. Additionally, AI-powered workout customization will be introduced, enabling users to create and personalize their own workout routines, further enhancing the app's adaptability and user experience.

6.4 **Conclusion**

A personalized fitness platform called Fitnessstan revolutionizes the way users approach fitness by seamlessly combining AI-powered diet and workout recommendations. The platform offers personalized fitness plans based on individual health metrics, and preferences, ensuring optimal results. By leveraging AI (Machine learning) for diet suggestions and exercise database for workout routines, it provides users with data-driven, effective guidance, particularly for individuals with specific health conditions like diabetes.

Designed for ease of use, Fitnessstan includes features such as progress tracking, personalized updates, and real-time adjustments. Looking ahead, the platform will integrate further enhancements like AI-driven workout customization, multi-language support, and dockerization, ensuring scalability and broad accessibility for users worldwide.

References

[1] R. Golagana, V. Sravani, and T. M. Reddy, “DIET RECOMMENDATION SYSTEM USING MACHINE LEARNING 4 kavitha.chekuri@raghuenggcollege.in*,” *Dogo Rangsang Res. J. UGC Care Gr. I J.*, vol. 13, no. 4, pp. 2347–7180, 2023.

[2] S. G, M. H. H, K. P. T J, and H. R, “Fitness Trainer Application Using Artificial Intelligence,” *Interantional J. Sci. Res. Eng. Manag.*, vol. 07, no. 10, pp. 1–11, 2023, doi: 10.55041/ijsrem26018.

[3] K. Subbaraj, “Multi-Choice Diet Recommendation Application for Indian Scenario Based on Insights from Ensemble Learning Techniques †,” *Eng. Proc.*, vol. 62, no. 1, 2024, doi: 10.3390/engproc2024062013.

[4] S. Sweidan, S. S. Askar, M. Abouhawwash, and E. Badr, “A hybrid healthy diet recommender system based on machine learning techniques,” *Comput. Biol. Med.*, vol. 184, no. May 2024, p. 109389, 2025, doi: 10.1016/j.compbiomed.2024.109389.

[5] P. B.R, D. N. N. Kumari, B. Manikanta, A. P. Chandana, and Y. L. . Aditya, “Personalized Diet Recommendation System Using Machine Learning,” *SSRN Electron. J.*, 2024, doi: 10.2139/ssrn.4877349.

[6] A. Ashraf, E. Sciences, T. Shahid, and E. Sciences, “Abstract :,” no. June, 2024, doi: 10.13140/RG.2.2.29212.30089.