GYM FITNESS TRACKER

C-Programming Project

Submitted By : *MAYANK TOMAR*

SAP ID : 590025685

Submitted To: *Dr. Srinivasan Ramachandran*

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Dr. Srinivasan Ramachandran** for giving me the opportunity to work on this C programming project. This project helped me understand many practical concepts which were earlier only theoretical to me.
I also thank my classmates and friends who motivated me to complete this project successfully.

ABSTRACT

This project, titled *Gym Fitness Tracker*, is a simple yet creative attempt to understand how the C programming language can be used to solve real-life problems.
The program allows the user to:

- Enter gym member details
- Calculate BMI
- Identify fitness level
- Give a workout recommendation
- Store the complete data in a text file
- View all saved members anytime

This project helped me learn and apply key C concepts such as:

- Structures
- Functions
- File Handling
- Conditional Logic
- Loops
- Modular Programming

Overall, this project acts as a beginner-friendly example of how programming can assist in the field of health and fitness.

INTRODUCTION

Fitness has become extremely important in today's lifestyle. People want to know if they are fit, underweight, overweight, or obese. BMI (Body Mass Index) is one of the most common ways to check this.
So I decided to create a simple program that:

1. Takes basic user details
2. Calculates BMI
3. Gives clear output
4. Suggests a gym routine
5. Saves everything for future viewing

PROBLEM DEFINITION

The problem was to build a simple system using C programming that can:

- Read user details
- Calculate BMI from weight and height
- Decide the fitness category
- Suggest an appropriate gym workout
- Store the information
- Display previous records upon request

Another aim was to make the program menu-driven so it feels like a small application rather than a plain code.

OBJECTIVES OF THE PROJECT

✓ To understand how structures store different types of data
✓ To learn the importance of modular programming
✓ To practice functions and reusability
✓ To understand how file operations permanently save data
✓ To connect theory with a real-world problem
✓ To improve logic-building and problem-solving skills

SCOPE OF PROJECT

This project can be used by:

- Gyms to store member health data
- Trainers to quickly calculate BMI
- Students to understand programming concepts
- Beginners who wish to track basic fitness levels

SYSTEM REQUIREMENTS

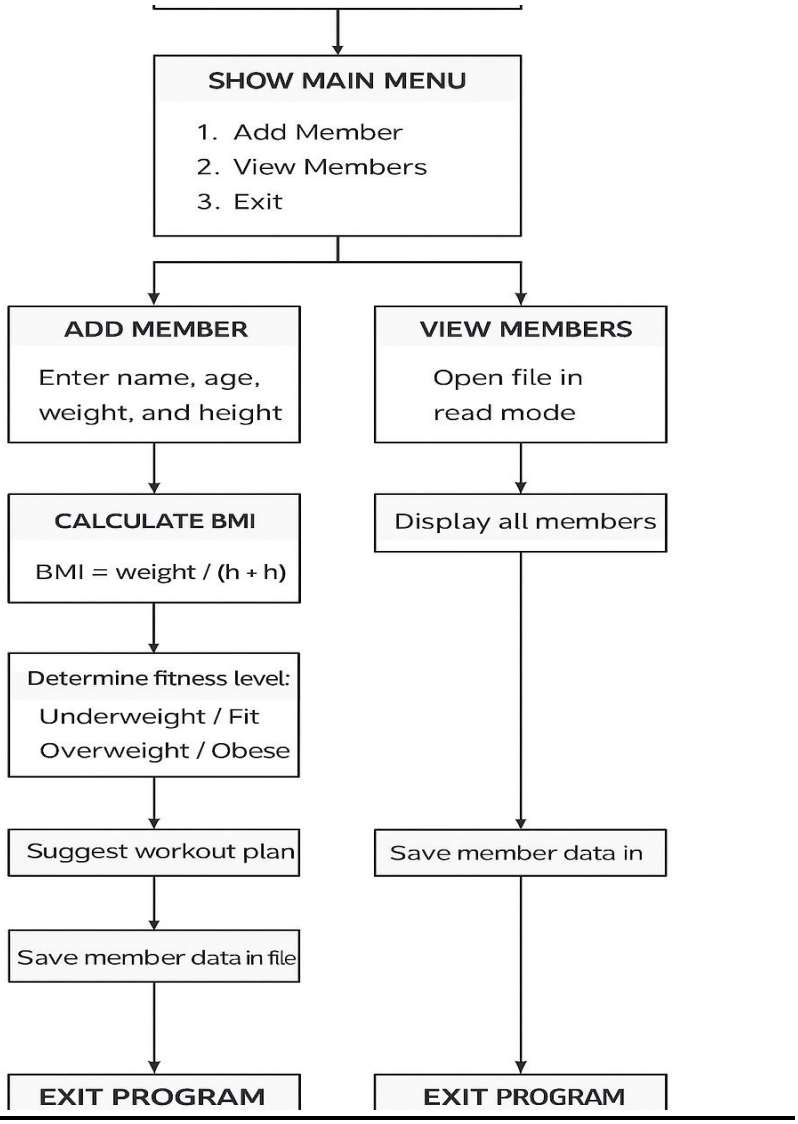
Hardware
• Any basic computer/laptop
• 2GB RAM or above

Software
• Windows / Linux
• GCC Compiler
• VS Code

ALGORITHM

1. Start
2. Display the main menu
3. Ask user to choose an option
4. If user chooses “Add Member”:
    - Read member name
    - Read age
    - Read weight
    - Read height
    - Calculate BMI using BMI = weight / (height * height)
    - Display BMI result
    - Display fitness category
    - Suggest gym workout
    - Save the data into a file
5. If user chooses “View Members”:
    - Open file
    - Read all records
    - Display each record
6. If user chooses Exit
    - Stop the program
7. Repeat menu until Exit

FLOWCHART


SHOW MAIN MENU
1. Add Member
2. View Members
3. Exit
ADD MEMBER
Enter name, age, weight, and height
VIEW MEMBERS
Open file in read mode
CALCULATE BMI
BMI = weight / (h + h)
Display all members
Determine fitness level:
Underweight / Fit
Overweight / Obese
Suggest workout plan
Save member data in
Save member data in file
EXIT PROGRAM
EXIT PROGRAM

IMPLEMENTATION DETAILS

main.c

This file displays the menu:

- Add Member
- View Members
- Exit

It calls functions from gym.c when needed.

gym.h

Contains:

- Structure
- Function declarations

Structure:

```
struct Member {
   char name[50];
   int age;
   float weight, height;
};
```

gym.c

Contains:

- Function to calculate BMI
- Function to display category
- Function to give workout suggestions
- Function to save data
- Function to view saved members

This separation of files makes the project more readable and professional.

WORKOUT RECOMMENDATION LOGIC

To make the project more realistic, I added gym routines:

If Underweight:
- High-calorie diet
- Strength training
- Light cardio

If Normal:
- Balanced workout
- Treadmill
- Chest press
- Core exercises

If Overweight:
- More cardio
- Cycling
- Running
- Weight control exercises

If Obese:
- Light walking
- Very light cardio
- Slow movement exercises

TESTING & RESULTS (DETAILED)

✓ Test Case 1

Input:
- Name: Rohan
- Age: 21
- Weight: 58
- Height: 1.72

Output:
- BMI = 19.6
- Category: Fit
- Workout: Mixed routine

✓ Test Case 2

Input:
• Name: Aman
• Age: 19
• Weight: 48
• Height: 1.65

Output:
• BMI = 17.6
• Category: Underweight
• Workout: Strength training

```
mtr@DESKTOP-6GLDC6G:~/Workspace/Gym-Fitness-Tracker$ gcc src/main.c src/gym.c -Iinclude -o gym
mtr@DESKTOP-6GLDC6G:~/Workspace/Gym-Fitness-Tracker$ ./gym

===== GYM FITNESS TRACKER =====
1. Add Gym Member
2. View Member Records
3. Exit
Choose option: 1
Enter name: Aman
Enter age: 19
Enter weight (kg): 48
Enter height (meters): 1.65

BMI = 17.63
Fitness Level : Weak (Need weight gain)

--- Recommended Gym Routine ---
Beginner Bulk Plan:
- Light dumbbells
- Pushups, squats
- Protein diet

===== GYM FITNESS TRACKER =====
1. Add Gym Member
2. View Member Records
3. Exit
Choose option:
```

✓ Test Case 3

Input:
• Name: Meena
• Age: 20
• Weight: 75
• Height: 1.58

Output:
• BMI = 30.0
• Category: Obese
• Workout: Light walking

- SAVED GYM MEMBER AND THEIR DATA IN gym_members.txt

```
--- Saved Gym Members ---
Name : Rohan
Age : 21
Weight(in Kg) : 58.00
Height(in meters) : 1.72
BMI=19.61


Name : Meena
Age : 27
Weight(in Kg) : 78.00
Height(in meters) : 1.62
BMI=29.72


Name : Dharya
Age : 21
Weight(in Kg) : 52.00
Height(in meters) : 1.71
BMI=17.78


Name : Aman
Age : 19
Weight(in Kg) : 48.00
Height(in meters) : 1.65
BMI=17.63
```

ADVANTAGES OF THIS PROJECT

✓ Very easy to use
✓ Helps students learn file handling
✓ Real-life application example
✓ Lightweight program
✓ Beginner-friendly logic

LIMITATIONS

✕ Does not store data in a database
✕ No graphical interface
✕ Limited workout suggestions
✕ Cannot handle multiple trainers/users

FUTURE ENHANCEMENTS

In the future this project can be expanded with:

- Weekly fitness progress
- Graph-based BMI visualization
- Diet recommendation system
- Water intake reminder
- Login system for multiple users

CONCLUSION

The Gym Fitness Tracker project helped me apply all basic concepts of C programming in a practical way. I learned how to handle user input, store data, calculate results, and display meaningful output. This project improved my understanding of:

- Logic building
- Modular programming
- File handling
- Real World Problem solving

This project is simple but very useful for learning. It gives a strong foundation for future programming projects.