Fitness Tracking System

Introduction:

With the increasing focus on health and wellness, there is a growing demand for digital solutions that assist individuals in tracking their fitness journeys. This website aims to provide a user-friendly platform for fitness enthusiasts.

Purpose:

Why this? : The rising occurrence of sedentary habits and health concerns, including obesity and cardiovascular diseases, underscores the critical demand for effective fitness solutions. Numerous individuals find it difficult to sustain motivation and monitor their progress, resulting in frustration and the potential abandonment of their fitness objectives. The Fitness Tracker Website seeks to tackle these issues by offering a holistic platform that assists users in their health endeavours.

Who Is It For?: Individuals new to fitness who need guidance and motivation. Anyone interested in maintaining or improving their health through regular physical activity.

Objective:

The objective of the Fitness Tracking System is to develop a system that enables users to monitor and manage their fitness activities and goals effectively. This project allows users to log various physical activities (such as running, cycling, or swimming), set fitness goals, and track their progress over time. The primary objectives include:

1) **Facilitate Activity Logging:** Allow users to record detailed information about their fitness activities, including type, duration, distance, and calories burned. This helps users maintain a consistent activity history and analyse trends.
2) **Goal Setting and Progress Tracking:** Enable users to set specific fitness goals, such as target distance, duration, or calorie burn, for various activities. The system helps users measure their progress and stay motivated by tracking achievements and completion status for each goal.
3) **Data Analysis for Improved Insights:** Offer insights into users' fitness data by calculating metrics (like total distance, total calories burned, etc.), enabling users to visualize their performance and achieve healthier lifestyles.
4) **Enable Integration with Frontend Applications:** Serve as a robust backend API that can be easily integrated with mobile or web applications, providing a seamless user experience for tracking fitness activities and progress in real time.

Technologies Involved:

- **Programming Language :**C#
- **Database Used :** Microsoft SQL Server
- **Frontend Design :** HTML ,CSS, ReactJs
- **Backend Framework:** ASP.NET Core
- **Integration of DB :** Entity Framework Core
- **Packages :** Nu-Get Packages
- **API :** Swagger

Tools Involved:

- Visual Studio , SQL Sever Management Studio, Visual Studio Code

Architecture Diagram:

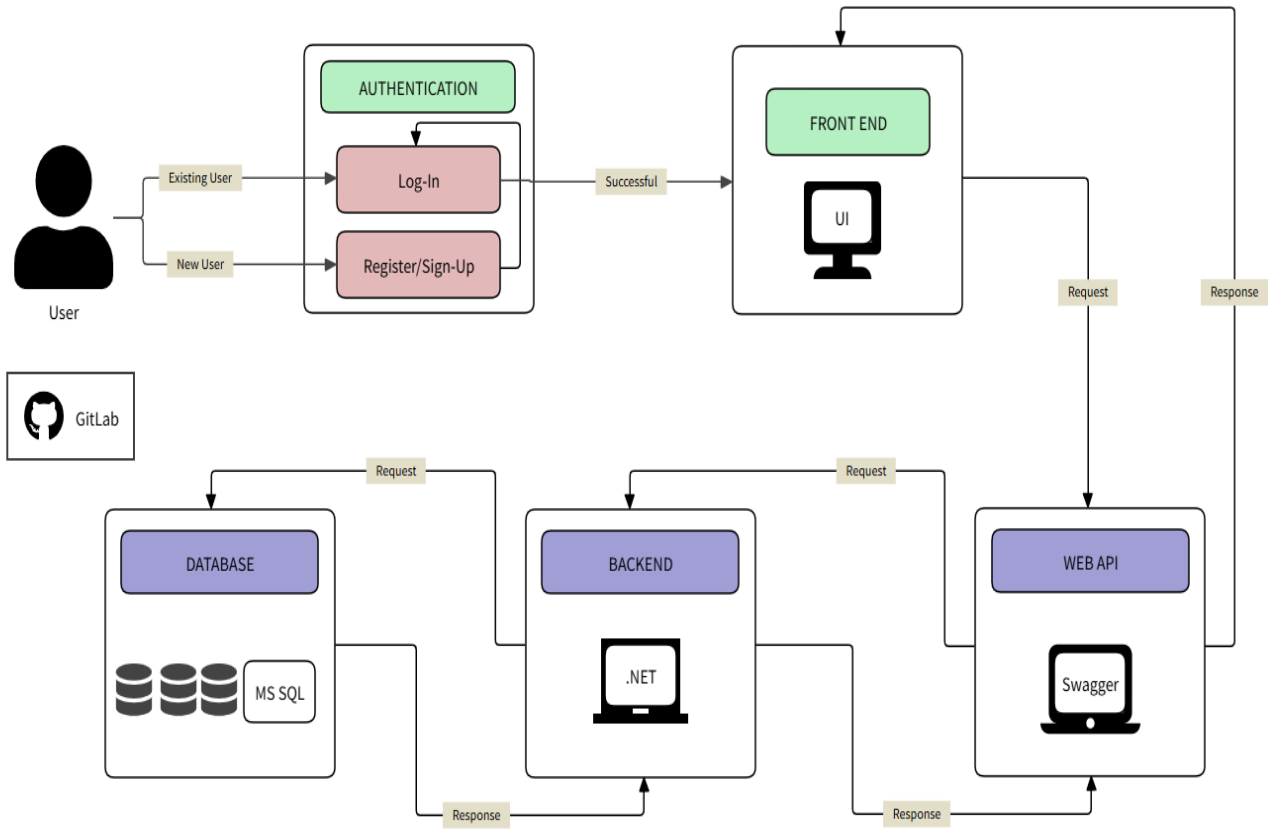

Fig:-Architecture Diagram

Database:

Microsoft SQL Server:

- **Database Design** : The database schema is designed using a code-first approach with Entity Framework Core. The key tables include: UserProfile Table, FitnessGoal Table, Workout Table.
- **Database Interactions**: Entity Framework Core facilitates interaction with the database, allowing for seamless querying, updating, and management of data. The FitnesssTrackerDbContext class manages these operations, ensuring data integrity and consistency across the application.

Backend (Server-Side)

ASP.NET Core:

- **Controllers:** These handle HTTP requests from the frontend, such as user authentication, adding user data, deleting user account, etc. Controllers include: UserController, WorkoutController , FitnessGoalController.

- **Services & Business Logic:** Implements the core business logic, processing data, validating inputs.
- **Repositories:** Repositories act as the data access layer, interfacing with the database to execute queries and manage data persistence. Key repositories include: UserRespository , UserBLRespository WorkoutRepository , WorkoutBLRepository ,FitnessGoalRepository, FitnessGoalBLRepository.

FRONTEND (CLIENT-SIDE)

REACT.JS:

- **Components:** The frontend is built using React.js, with various components dedicated to specific functionalities. These include: Layout and Sidebar.
- **Pages:** Here we are having Login, Register, Dashboard, FitnessGoals, Workout, Statistics and Settings. Each has specific functionalities.
- **API Integration:** Axios is used for HTTP requests to interact with the backend API. This includes submitting data, retrieving back and deleting using required methods for this project.

Implementation:

1. Database -SQLServer:

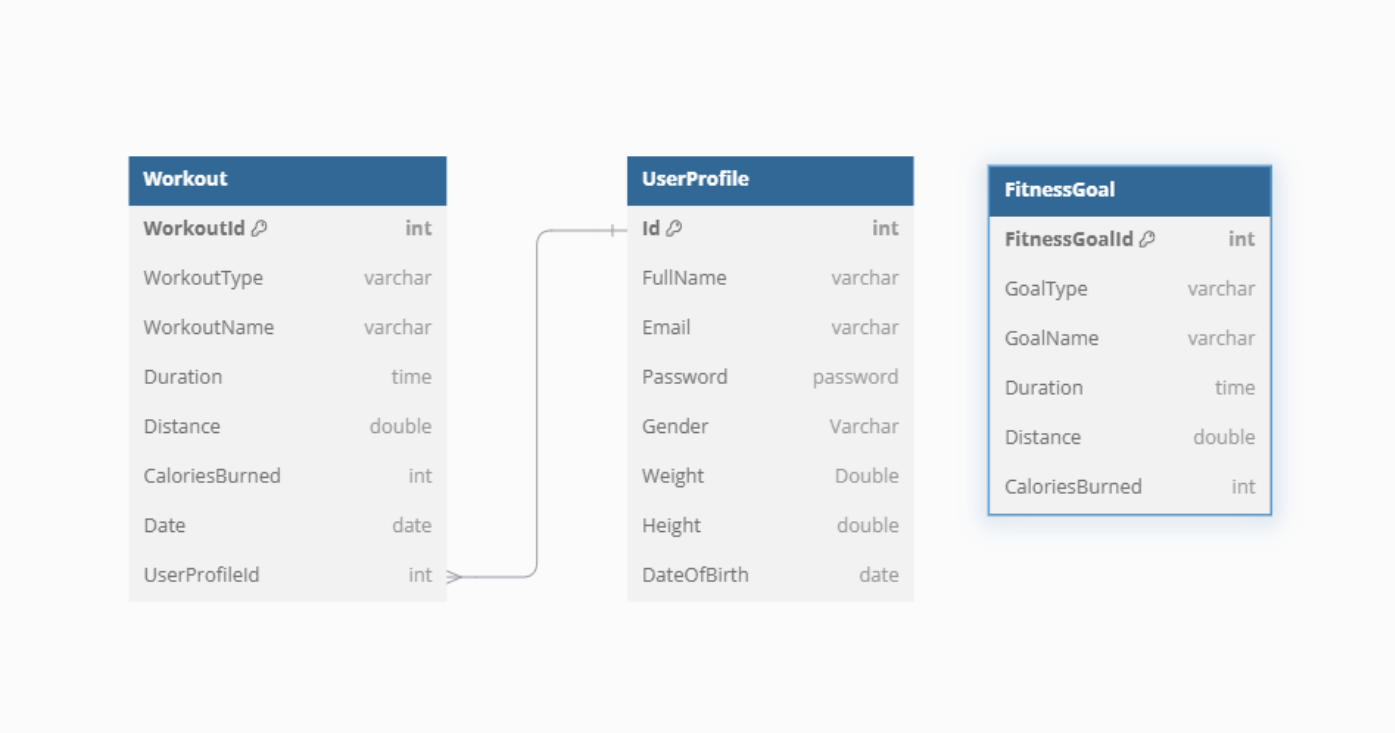

Fig :-Database Diagram

- **UserProfile:**
  - **Id:** Unique Identifier for each User.
  - **FullName :** Full Name of the User.
  - **Email :** E-Mail Id of the particular User.
  - **Password :** Unique Key or String for each user to login.
  - **Gender:** Describes about Gender of the User.
  - **Weight:** Describes about Weight of the User
  - **Height:** Describes about Height of the User
  - **DateOfBirth:** Tells about Date of Birth of the User
- **FitnessGoal:**
  - **FitnessGoalId:** Unique Id of the Particular Activity**.**
  - **GoalType:** Describes about the Type of Activity like Indoor , Outdoor.
  - **GoalName:** Tells About the Name of the Goal.
  - **Duration:** States About the minimum time duration of the Activity.
  - **Distance:** Tells about the distance need to be covered as part of the activity.
  - **CaloriesBurned:** Tells About the Calories Need To burned as part of the Activity.
- **Workout:**
  - **WorkoutId:** Unique Id that differentiates from each other.
  - **WorkoutType:** Describes about the Type of Activity like Indoor , Outdoor.
  - **WorkoutName:** Tells About the Name of the Workout/Activity.
  - **Duration:** States About the time duration of the Activity done by user.
  - **Distance:** Tells about the distance that has covered as part of the workout.
  - **CaloriesBurned:** Tells About the Calories that has burned as part of the workout.
  - **Date:** Indicates the date of the workout that has been performed by user.
  - **UserProfileId:** Indicates the User Id of the User Who has done the workout.

2. Backend– ASP.NET

The backend is implemented using ASP.NET Core Web API with a code-first approach to database management. The backend consists of the following components:

- **Controllers**: Handle HTTP requests from the frontend.
  - UserController manages about user details about login and everything.
  - FitnessGoalController deals with the Goals of the workouts.
  - WorkoutController deals with the workouts of Particular User and weather the Goals has been achieved or not.
- **Models**: Define the Tables used in the application. Models include Workouts , UserProfile and FitnessGoals.

- **Repositories**: Implement data access and business logic. Repositories include UserRepository , FitnessGoalRepository , WorkoutRepository ..etc each of which interacts with the respective database tables.
- **Database Context**: The FitnessTrackerDbContext class manages database interactions using Entity Framework Core.
- **Migrations:** The Migrations folder contains scripts for managing database schema changes.
- **appsettings.json:** This configuration file contains essential settings, including the database connection string, to ensure proper application configuration.

3. FrontEnd-React.js

- **Pages folder:**
  - **Register/Sign-Up Page:** Allows new users to create an account. The form includes fields for Fullname, Email, Password, Gender, Weight, Height and DateOfBirth. Form validation is implemented using Formik to ensure that each field is correctly filled out and that the email and password are unique across the system.
  - **Login Page:** Allows existing users to log in using their email and password. The system checks if the user exists in the database and logs in.
  - **Dashboard/Home Page:** Displays after successful login. It contains a side navbar that contains Dashboard, Fitness Goals, Workout, Statistics and Settings.
- **Forms and Validation:** Validation is performed during both sign-up and login processes.
- **API Integration:** Axios is used to make HTTP requests to the backend API.
- **Routing:** React Router is implemented to manage navigation between different pages, ensuring a smooth transition between the Login, Register, Dashboard, and other pages.
- **UI/UX Design:** Components are styled using CSS or libraries like tailwind to create a user-friendly and visually appealing interface that aligns with the fitness tracking theme.
- **GitLab Version Control:** All frontend code is managed in GitLab, ensuring version tracking, collaboration, and integration with the backend during development and deployment.

4. Version Control

GitLab: Used for source code management, enabling collaboration and version tracking.

5. Testing

- Testing for the Stock Market Data Application is performed using C# with **Moq** for all methods (Get, Post, Put, Delete). xUnit is used for unit testing, and Swagger is utilized to develop and describe RESTful APIs.

- **Unit Testing**: xUnit tests are written for each API endpoint to ensure that they handle both valid and invalid data correctly. Moq is used to mock dependencies, ensuring that the business logic is properly tested in isolation.

Class Diagram:

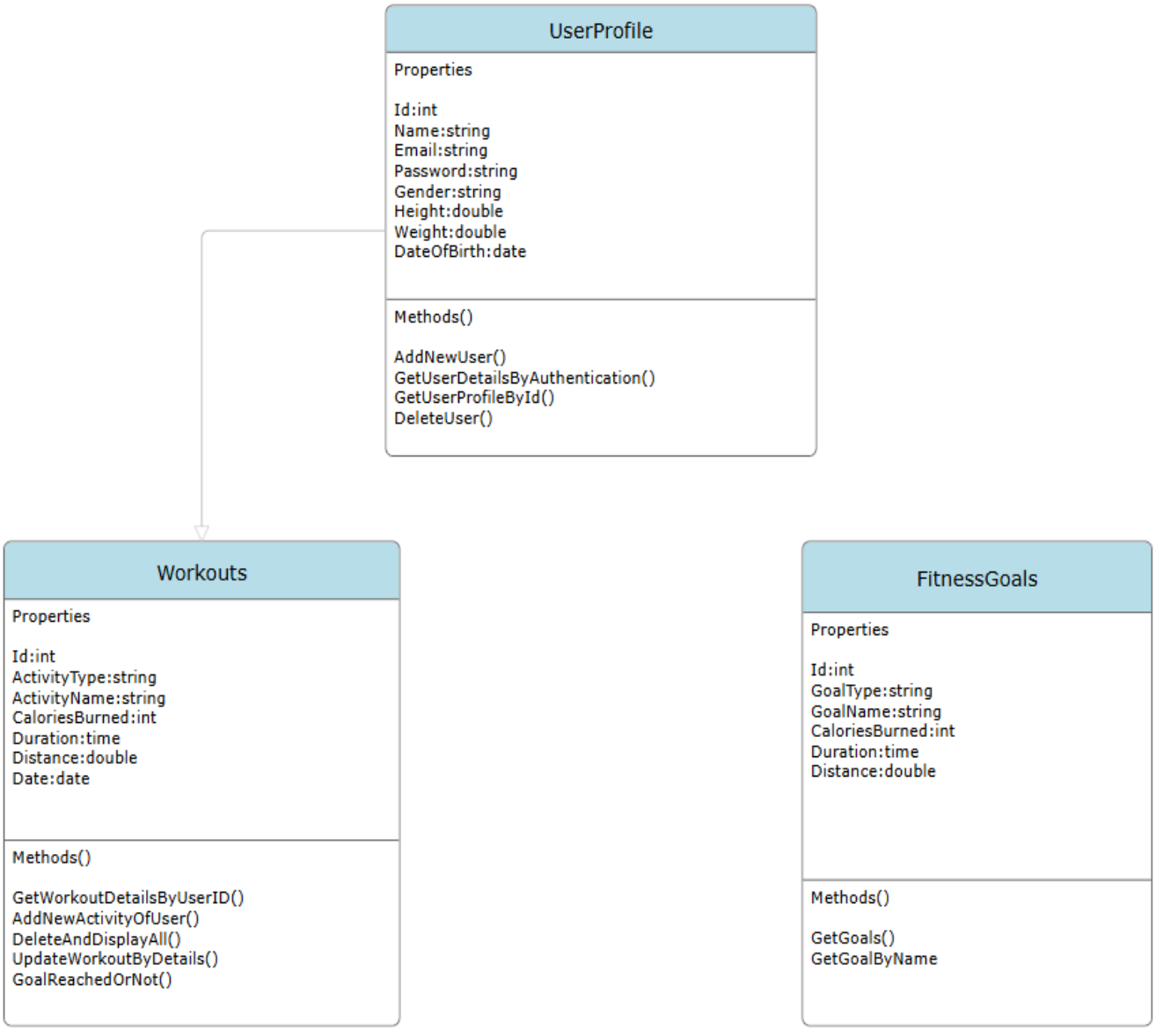

Fig :-Class Diagram

Flowchart Diagram:

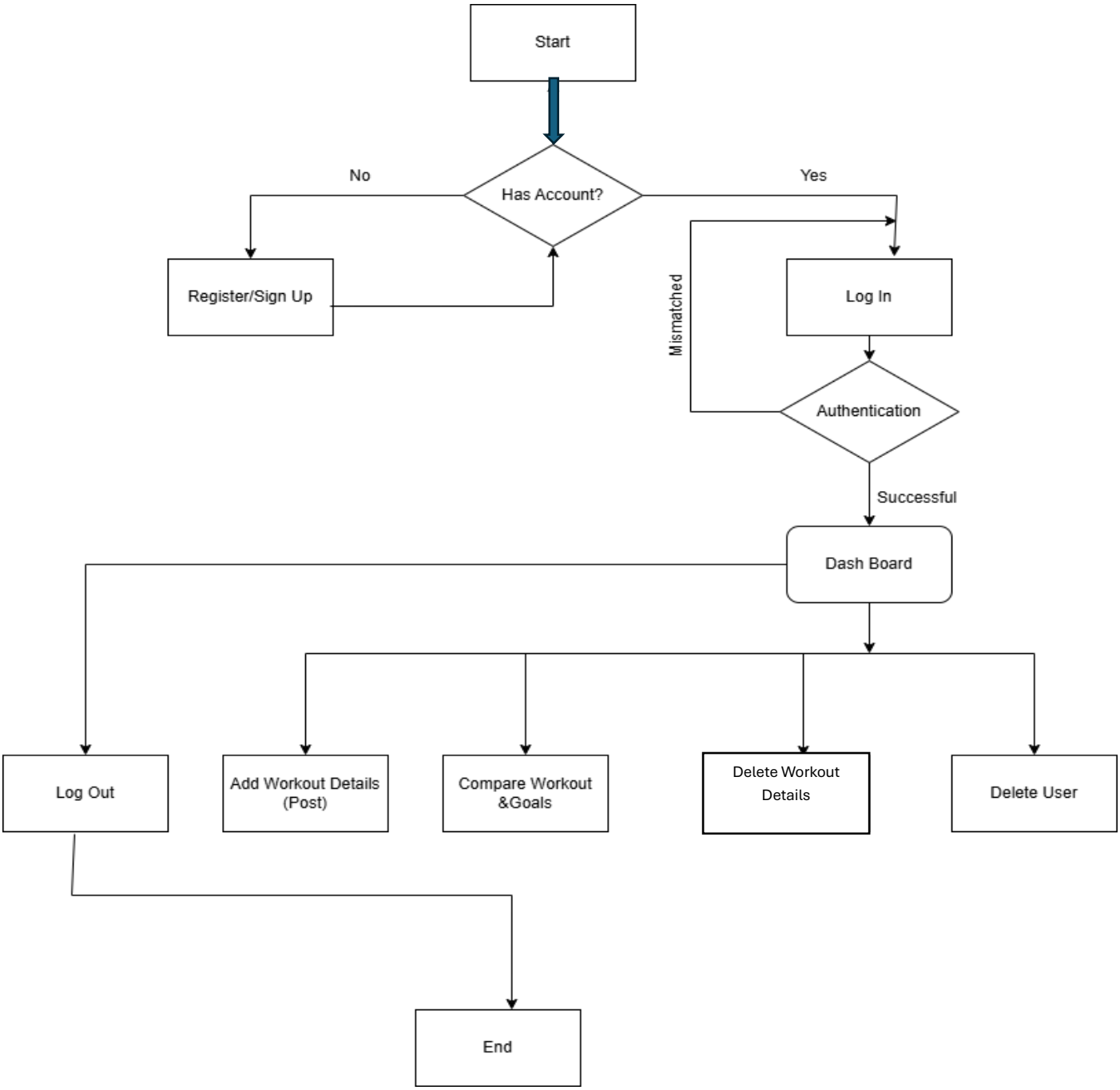

Fig :-FlowChart Diagram

Conclusion:

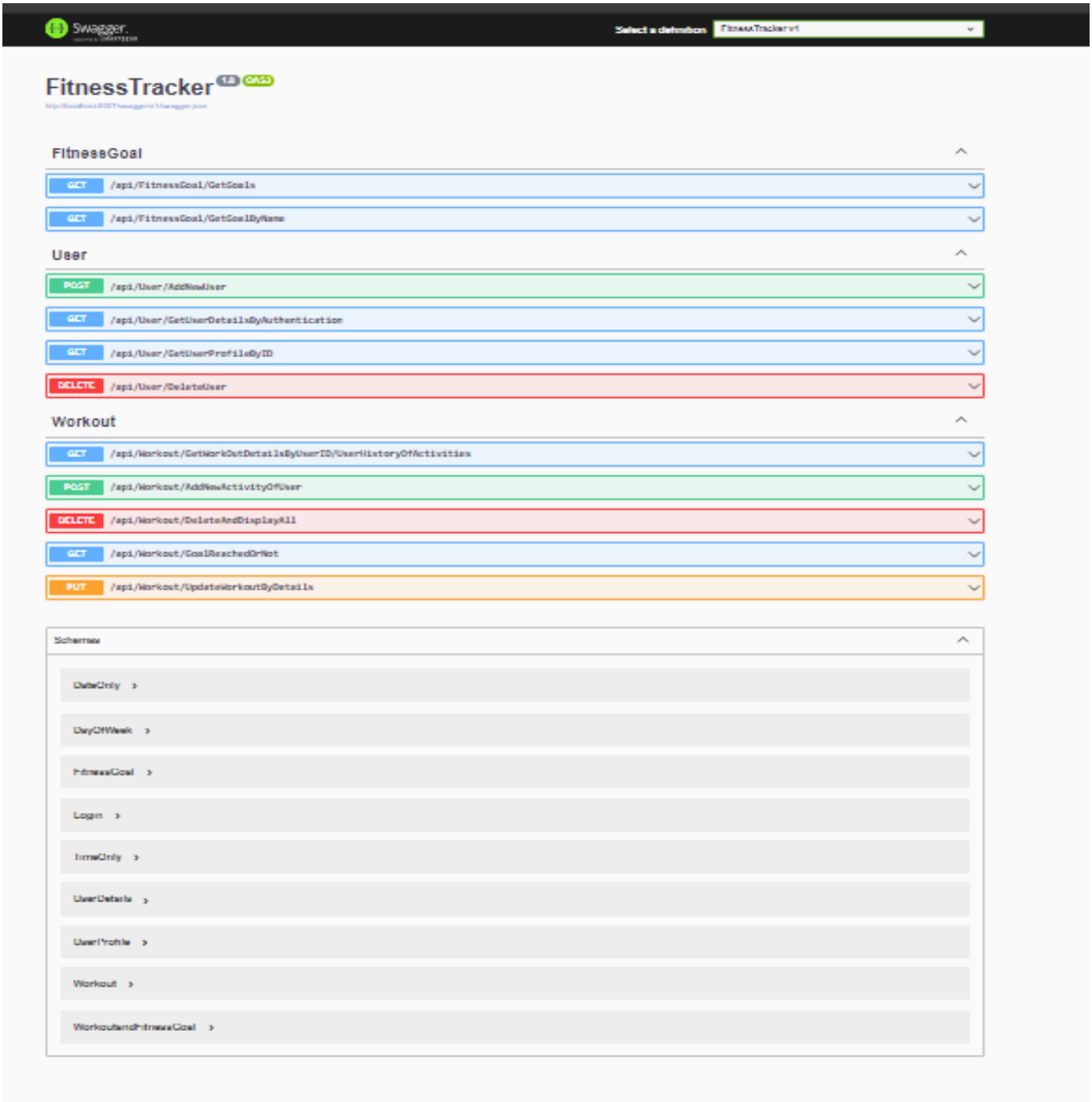

Fig :-Swagger API

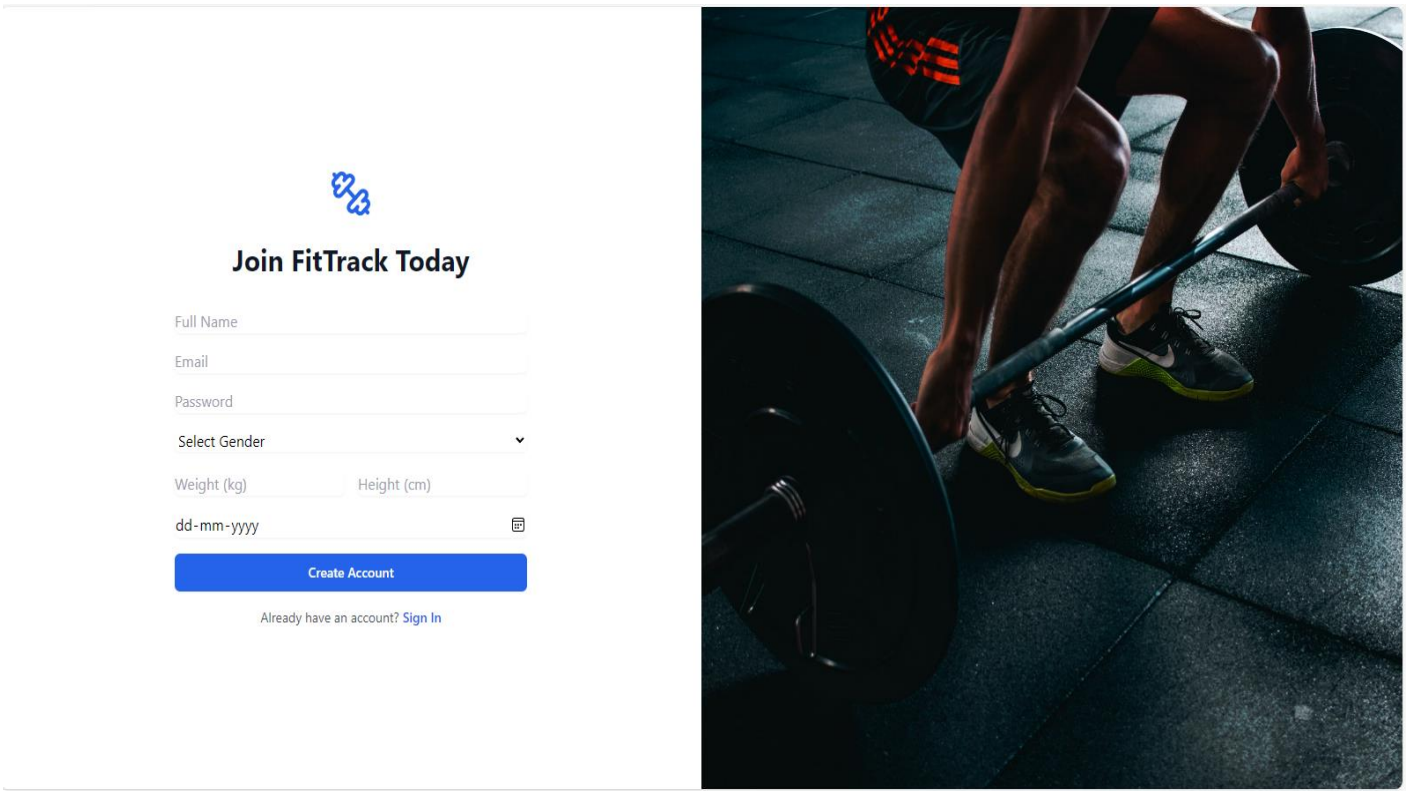

Fig :-SignUp/Register

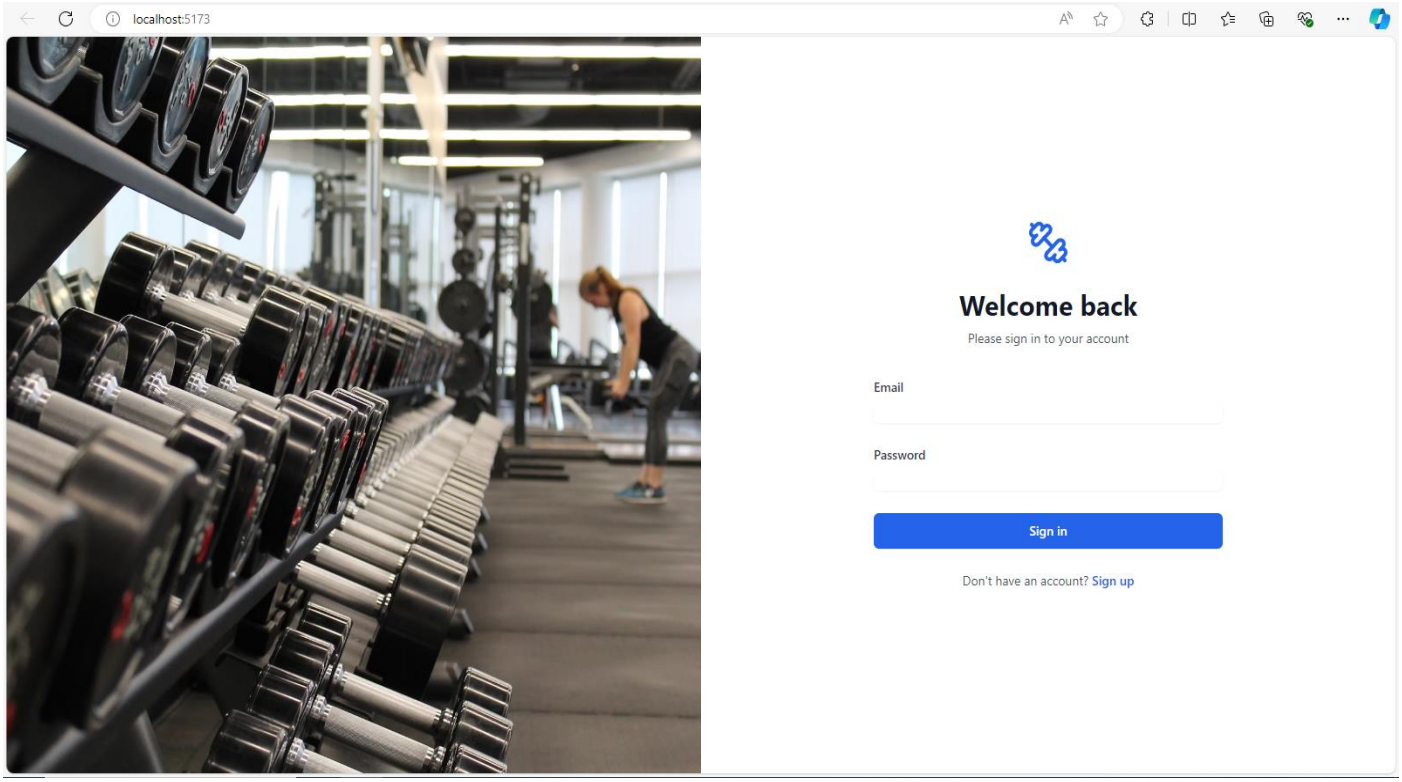

Fig :-LogIn Form

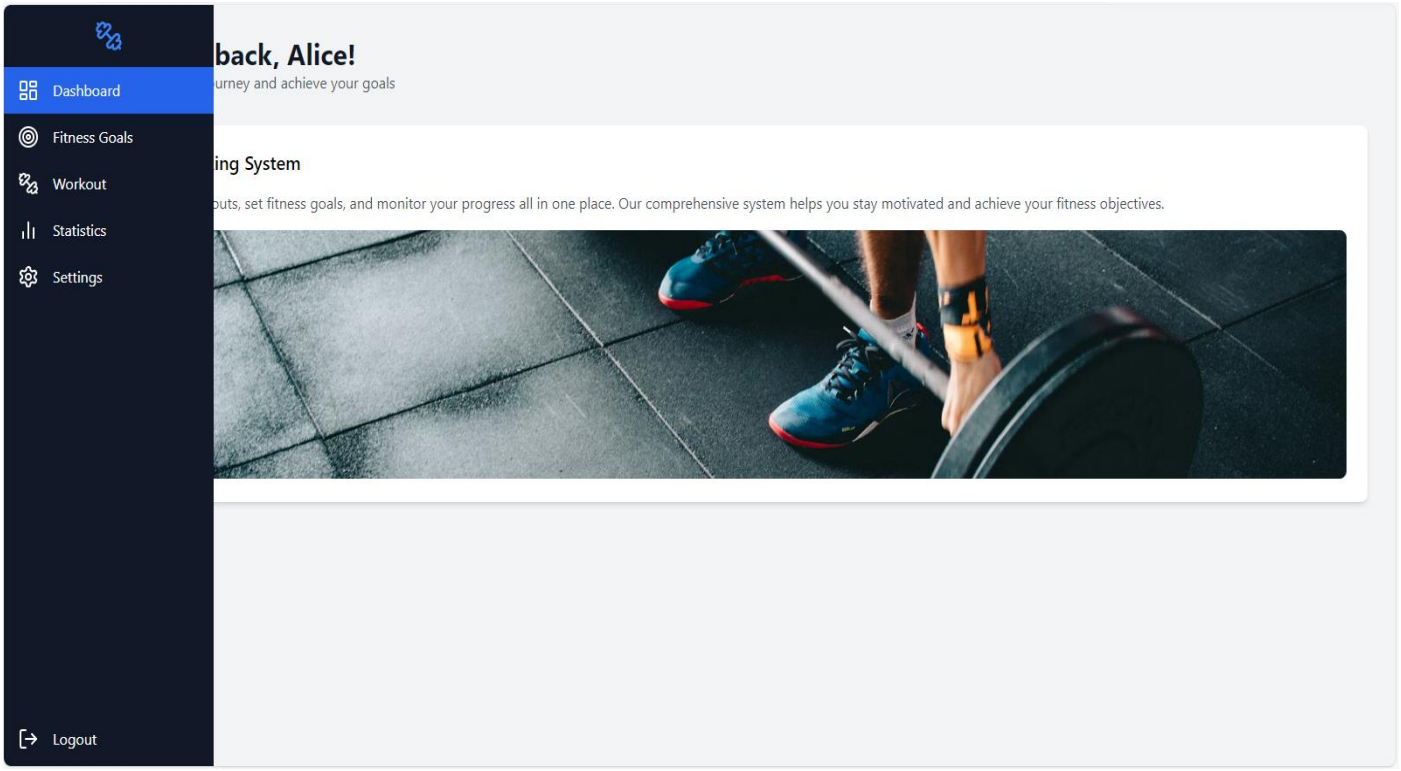

Fig :-HomePage & Dashboard

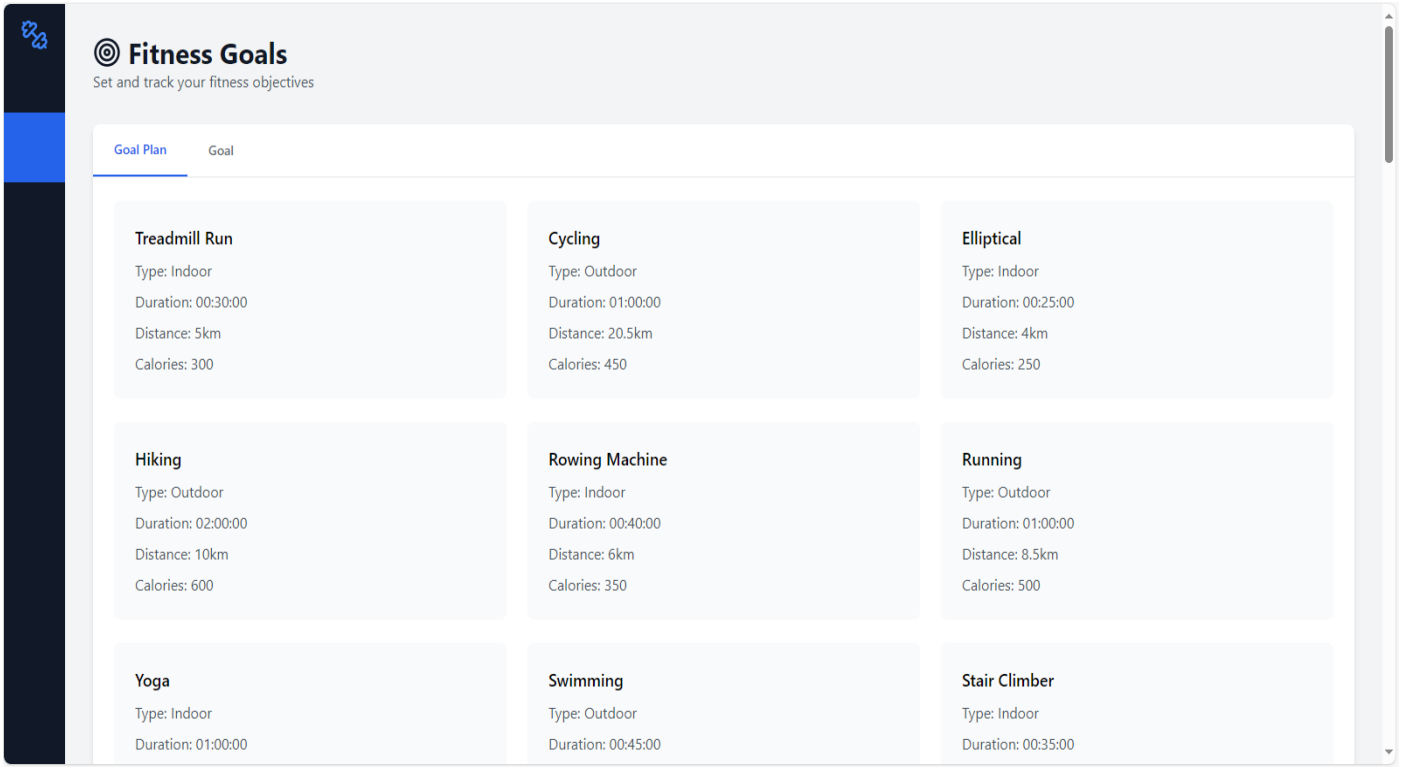

Fig :-All Goals Details

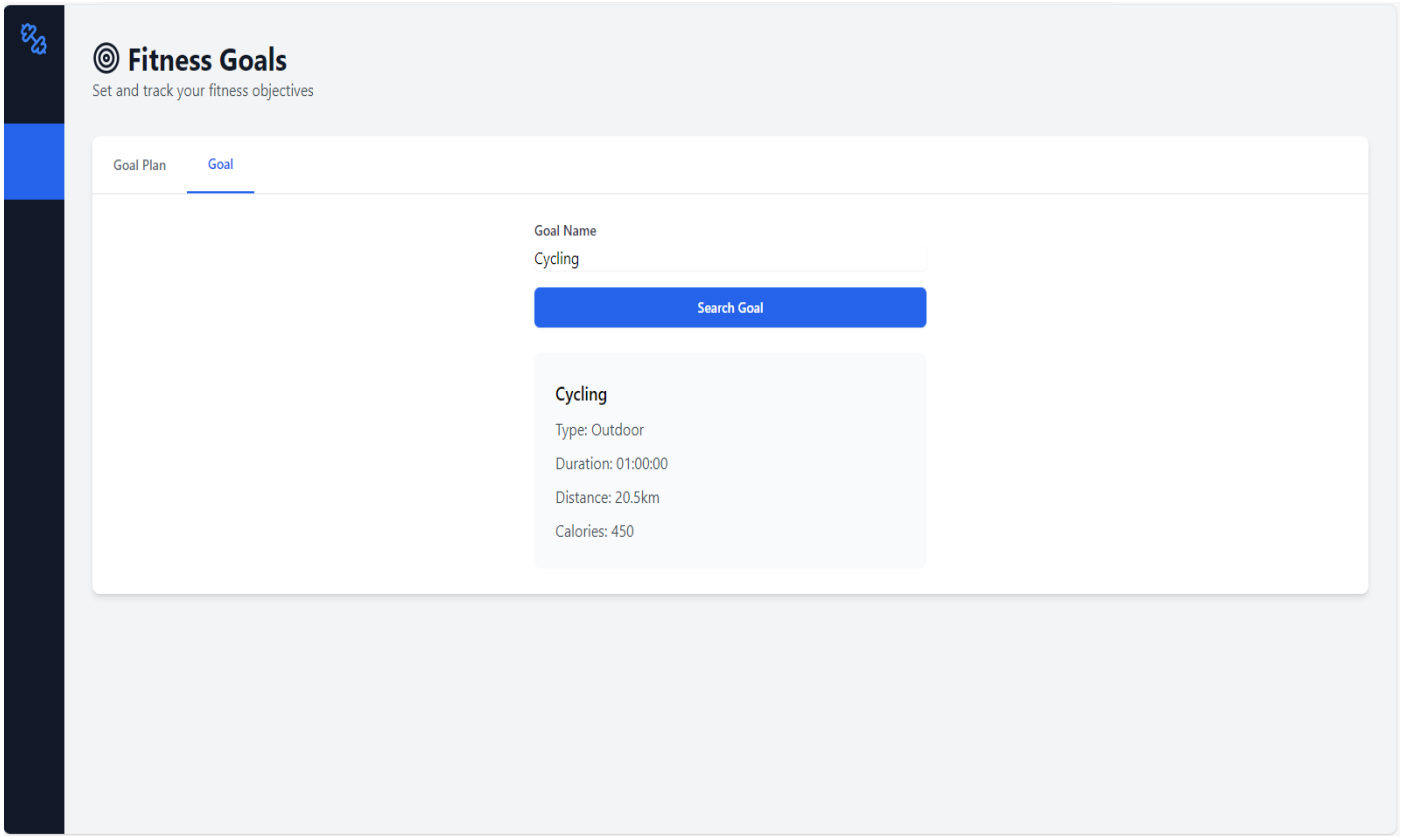

Fig :-Goal Details By Name

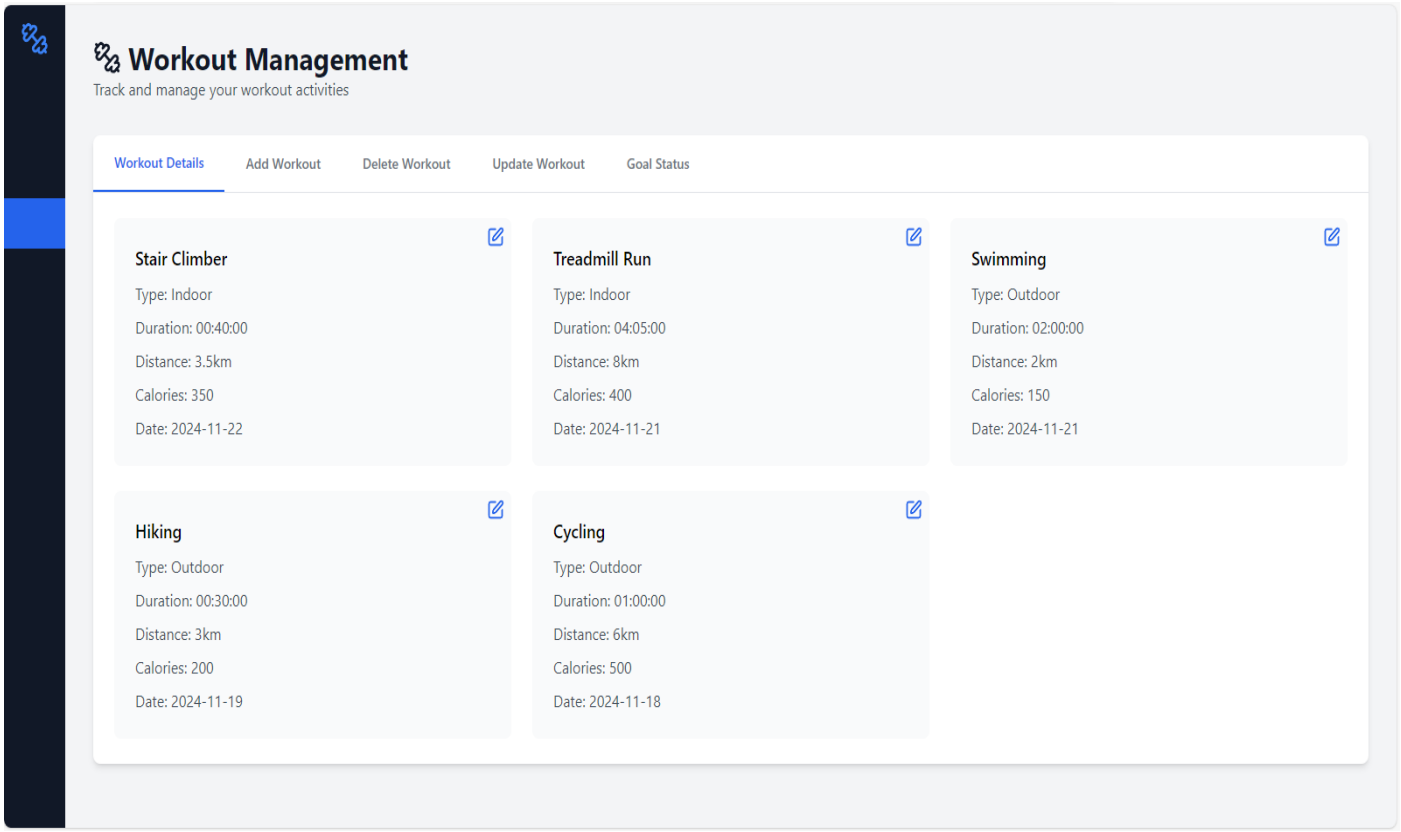

Fig :-Workout History of User

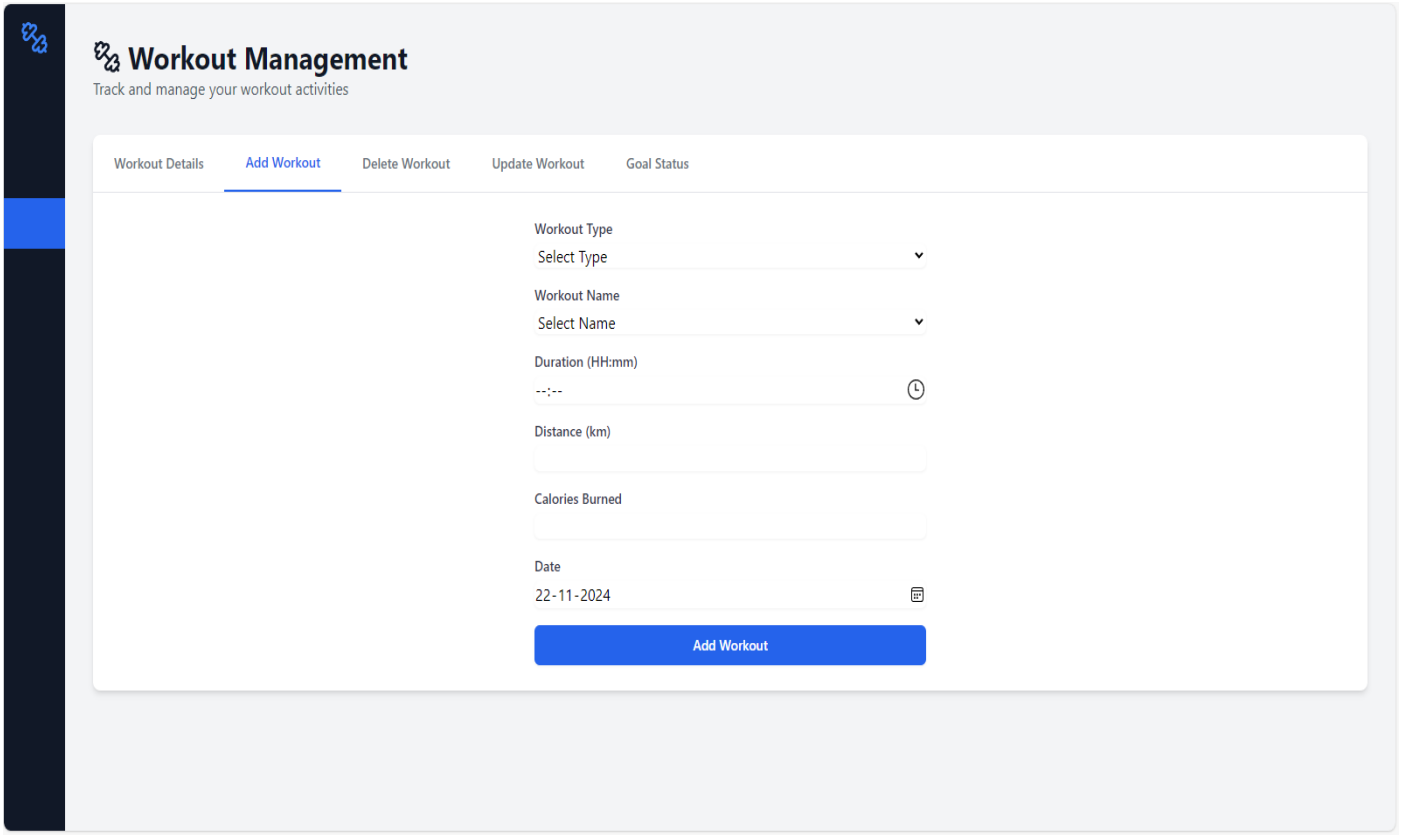

Fig :-Add new Workout of User

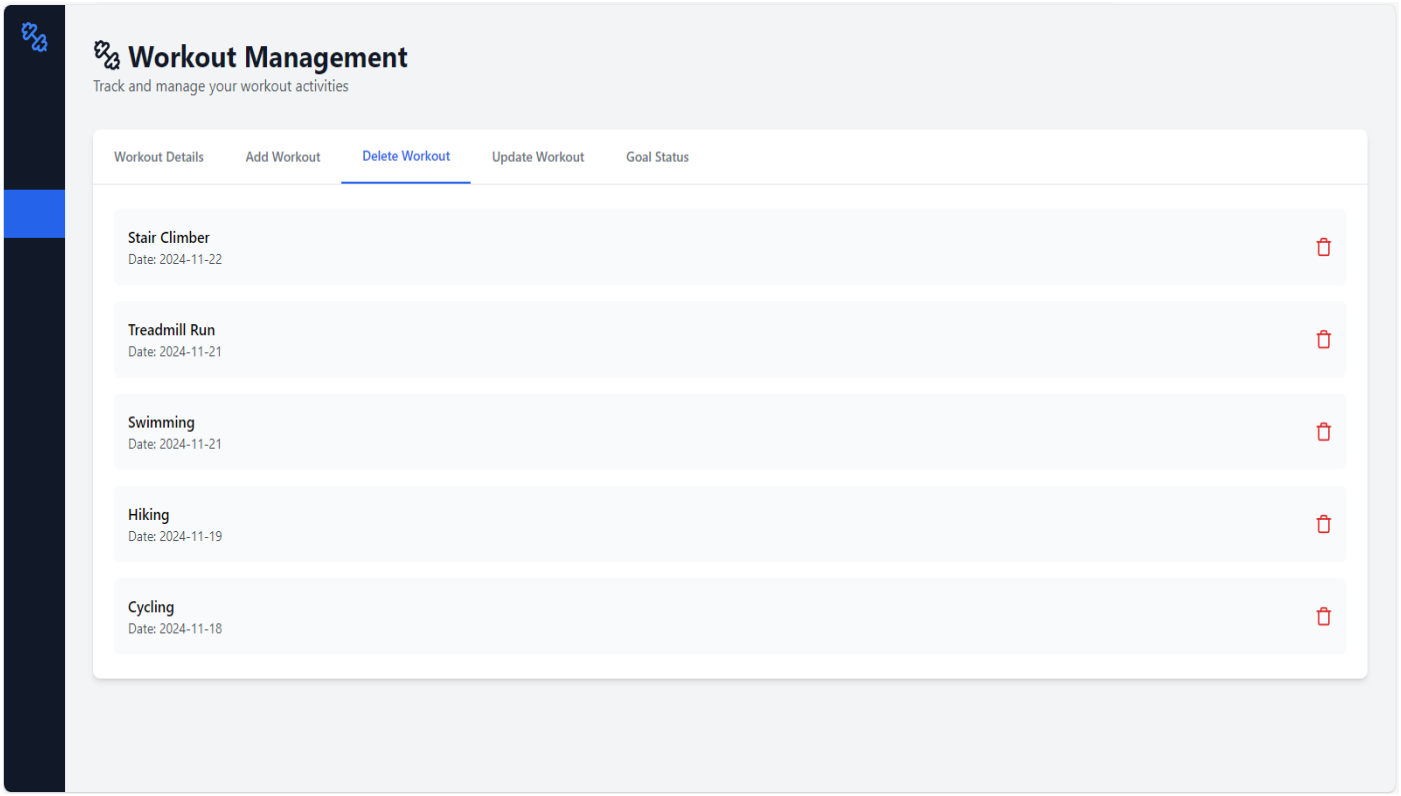

Fig :-Delete a Workout and display Remaining

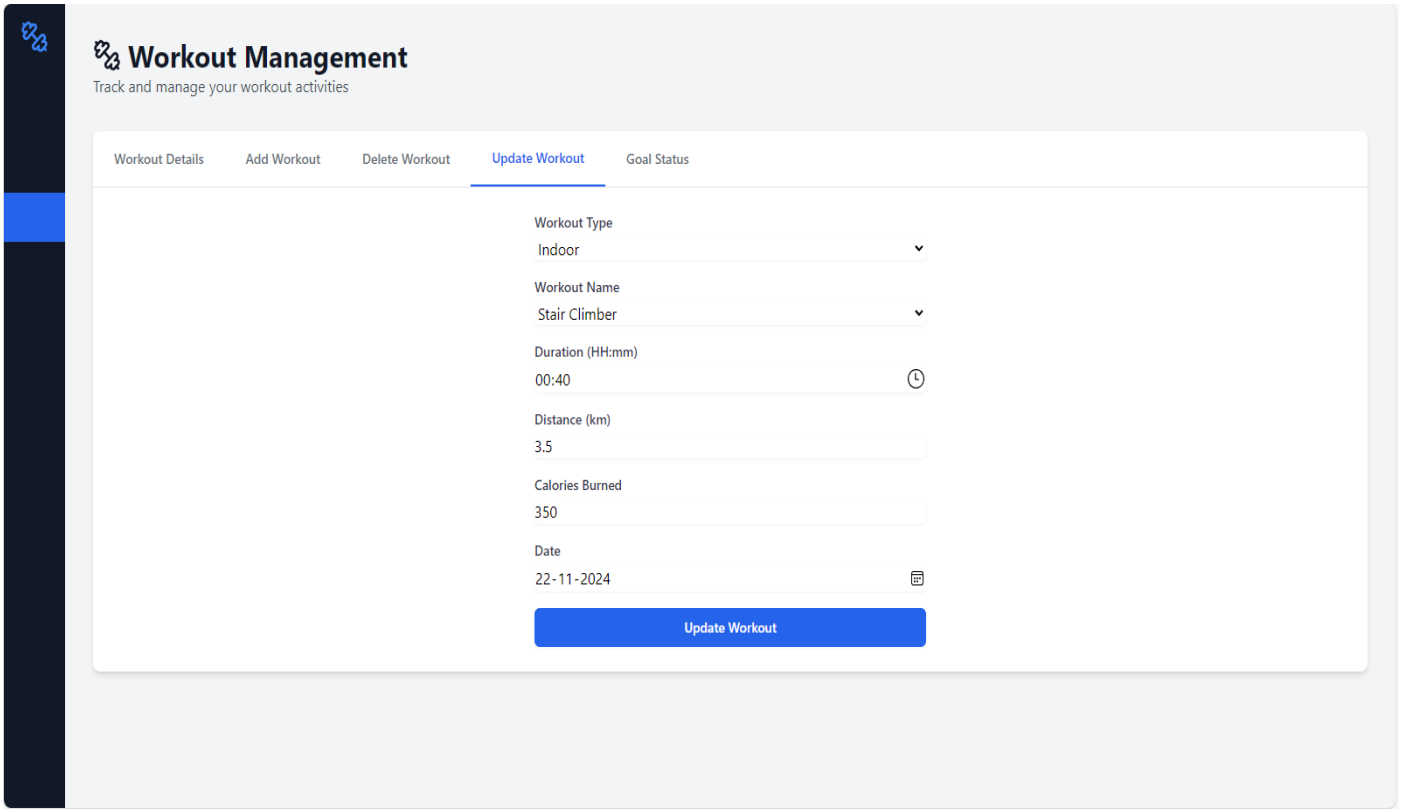

Fig :-Edit Workouts

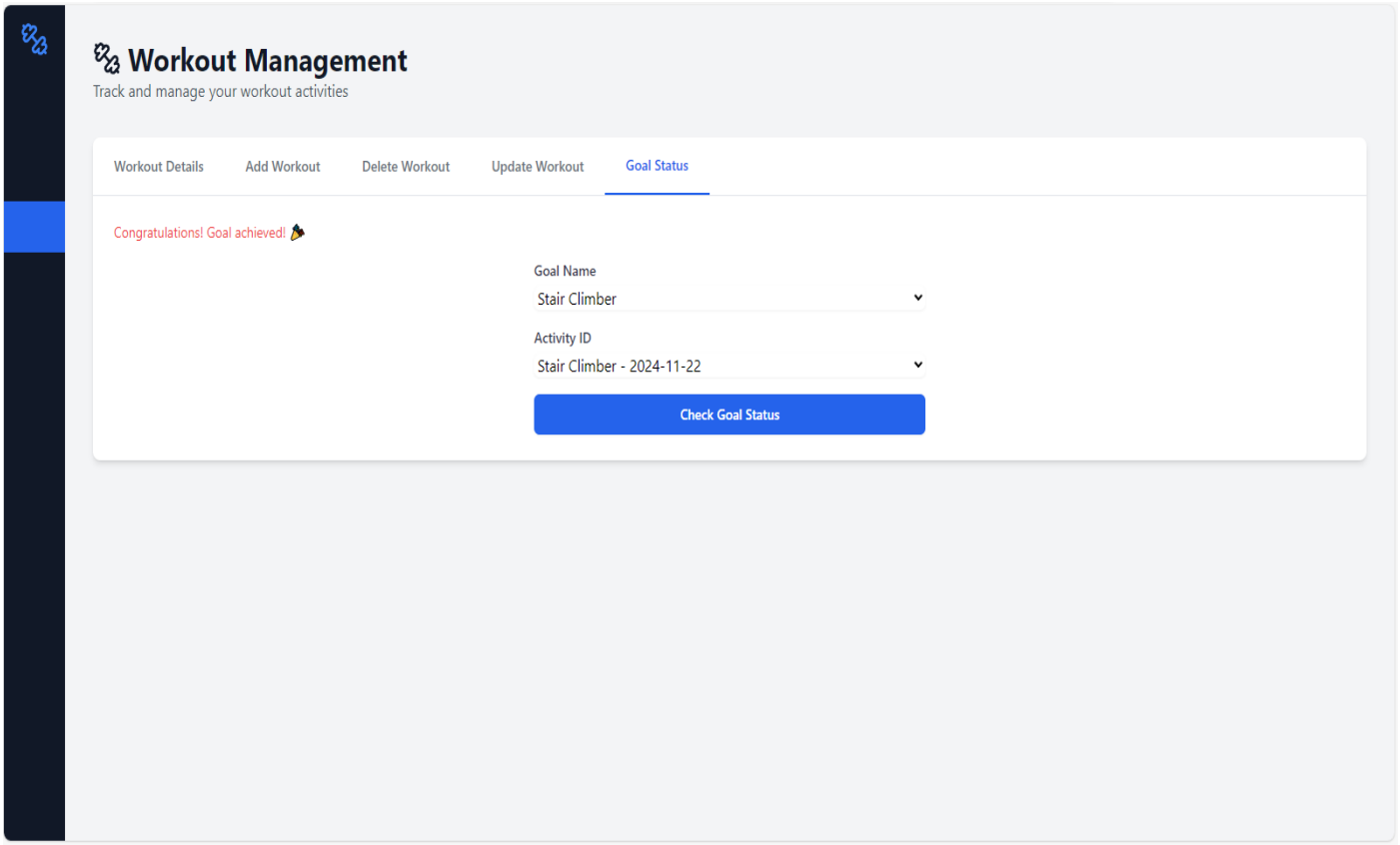

Fig :-Comparsion and Displaying weather Goal Achieved or Not.

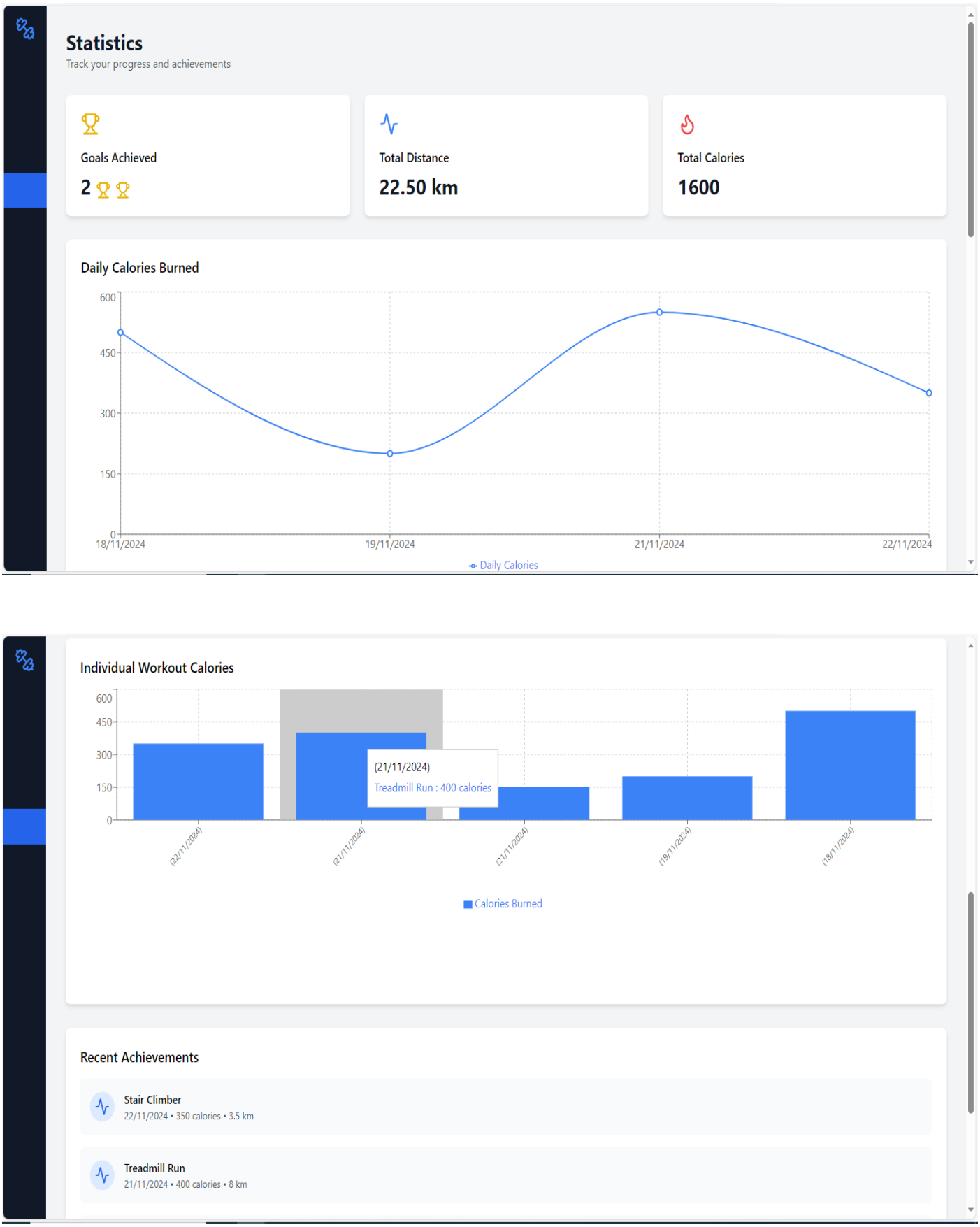

Fig :-Statistics of a particular User

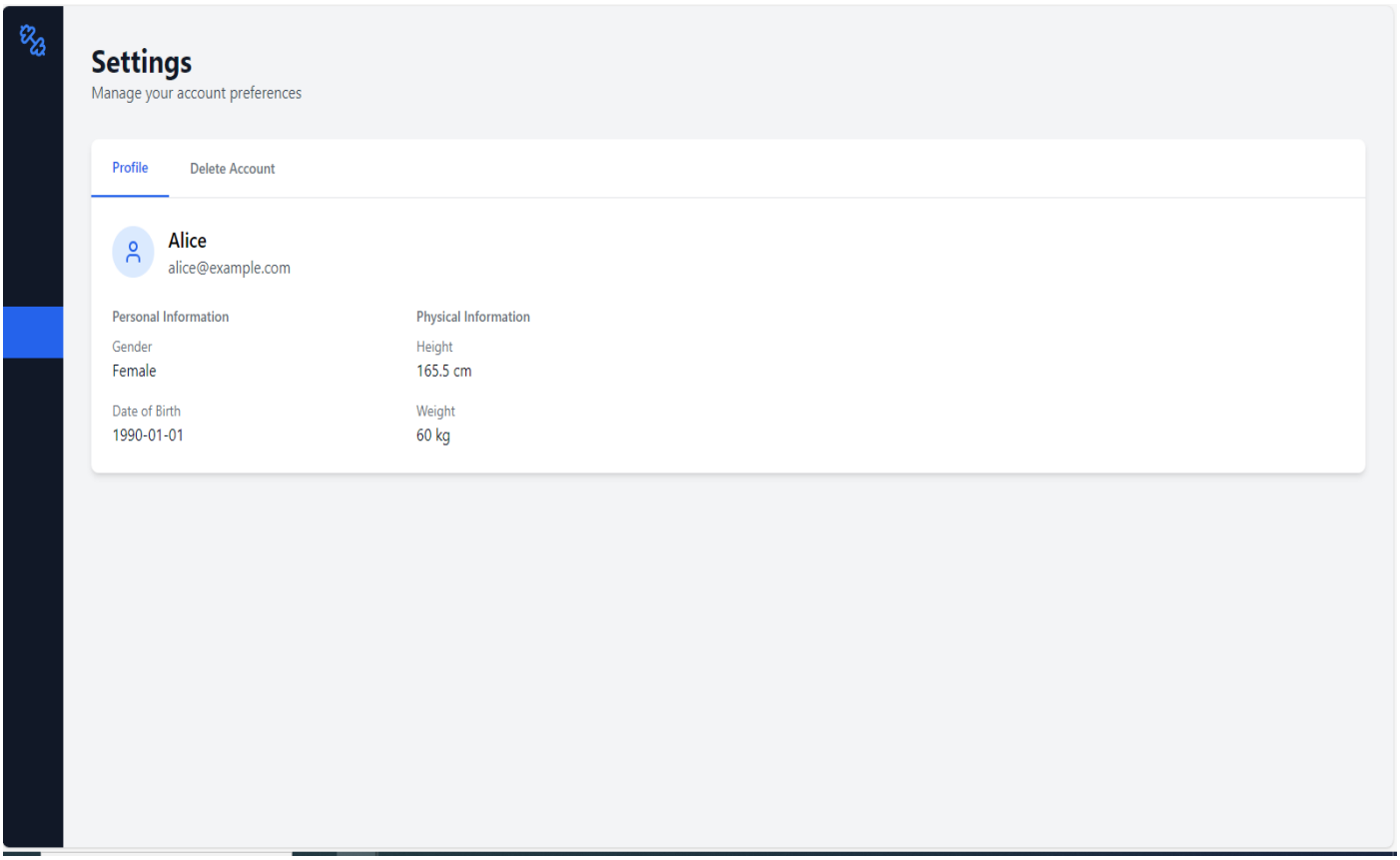

Fig :-User Details -Profile.

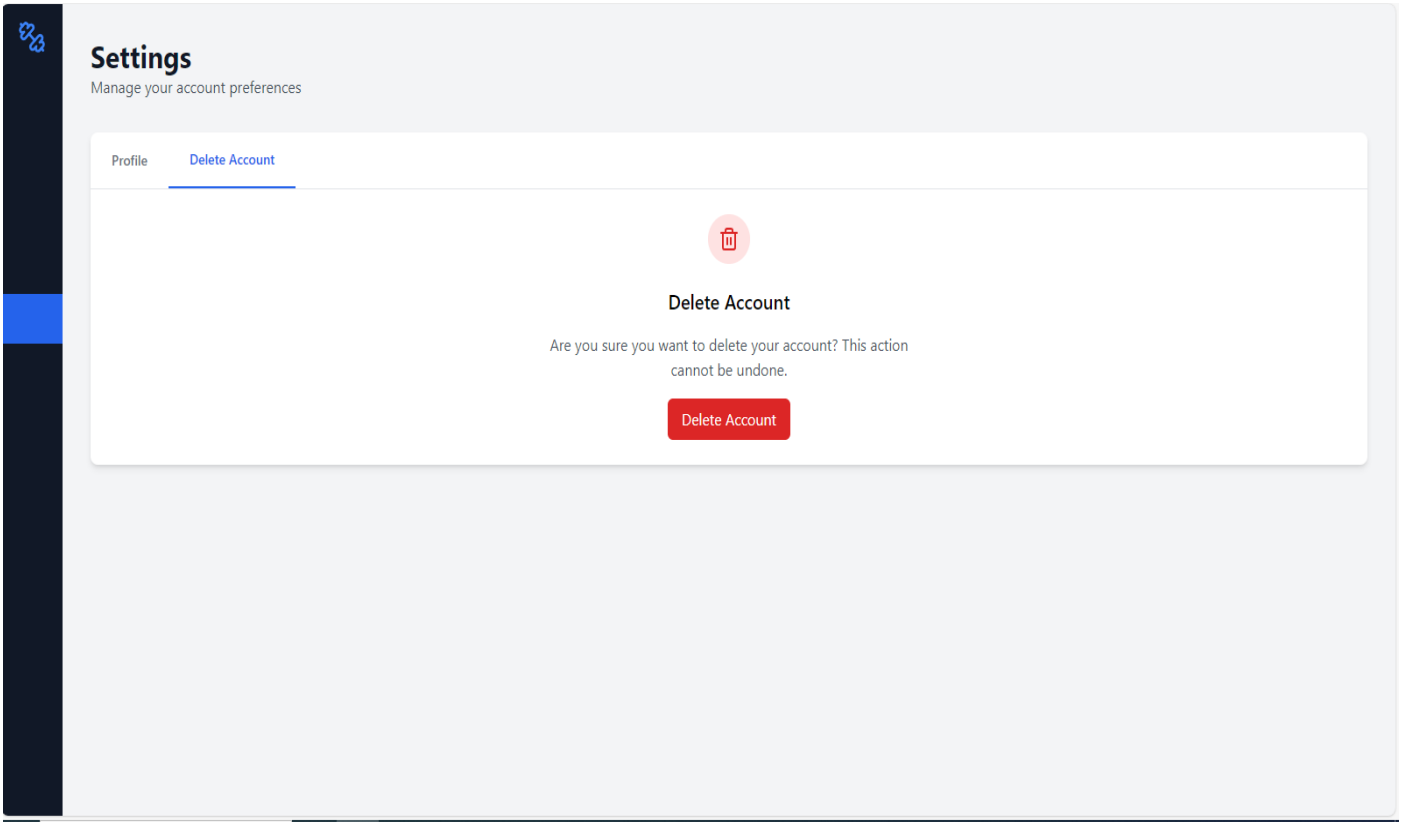

Fig :-Delete Account.

Future Scope :

- Adding GPS tracking Feature for the Outdoor Activities like running , jogging , Cycling etc.
- Adding a Feature like work-reward in order increase the interest of users.
- Adding Nutrition and Diet Plan Based on the Workouts user has chosen with consideration of his/her Age.