**Project  Documentation**

                                           FitFlex:Your Personal Fitness Companion

1. **Introduction**

* **Project Title:**FitFlex -Your Personal Fitness Companion.
* **Team ID:**NM2025TMIV35159
* **Team Leader :**name & Mail id

                 Shanmuga Priya&priyasamynathanvj07@gmail.com

* **Team Members:**

                Sivapriya & sivapriyasenthil22@gamil.com

                Usha Nandhini &umaausha063@gmail.com

                Yamuna & duraiyamuna66@gmail.com

1. **Project Overview**

* **Purpose:**

     FitFlex is designed to help individuals track their  workouts, diet, and wellness goals in one place. Unlike typical apps, this project focuses on personalized recommendations and a clean, mobile-first UI.

                      1.Weight Management- Lose or maintain body weight.

                      2.Muscle Building – Increase strength and muscle size.

                      3.Endurance Improvement- Boost stamina for longer physical activities.

                      4.Flexibility & Mobility- Enhance range of motion and prevent injuries.

                      5.Cardiovascular Health- Strengthen the heart and lungs.

* **Goals:**

                Encourage users to stay consistent with fitness.

                Provide a centralized platform to log workouts and meals.

                Offer visual progress tracking with charts.

                Build a scalable frontend for future AI/ML integration.

* **Features:**

                Log daily workouts with sets, reps, and time.

                Track nutrition intake and calories burned.

                Graphical charts showing weekly/monthly progress.

                Daily reminders & motivational notifications.

                Customizable UI themes (light/dark mode).

                Responsive design for mobile and desktop.

1. **Architecture**

* **Frontend:**

FitFlex uses a modular, component-based architecture (React/Angular/Vue) to build reusable UI elements like workout cards, progress trackers, and user dashboards. This ensures maintainability and scalability of the frontend code.

* **Backend:**

  The backend uses MongoDB as the primary database, enabling flexible schema designs for storing user profiles, workout logs, and nutrition plans. Mongoose ODM is used to manage data models and ensure validation.

**4.Setup Instructions**

* **Prerequisites:**

              -[Node.js](http://node.js):

[Node.js](http://node.js) is an JavaScript runtime built on chromes v8 engine, used to built fast, scalable server-side and networking applications.

              -Mango dp:

                    Mongo DB is a NoSQL database that stores data in JSON-Like documents, making it flexible and scalable for modern applications..

              -Git:

                    Git is a distributed version control system used to track changes in code and collaborate efficiently with others.

              -[React.js](http://react.js):

[React.js](http://react.js) is a JavaScript library for building user interfaces, especially for single-page applications, using a components-bases architecture.

               -[Express.js](http://express.js):

                     A minimal and flexible [Node.ja](http://node.ja) web applications framework used to build web and API serves.

                     Visual Studio Code :Visual studio code is a free, lightweight, and open-sources.

* **Initialisation Steps:**

                 1.Instal Python on your system.

                 2.Save the FitFlex program py    (e.g.,fitflex.py).

                 3.Open terminal or command prompt

                 4.Run the program using: python fitflix.py

                 5.Start adding your steps,water intake,and workouts.

**5.  Folder Structure**

fitflex/

│── public/

│── src/

│   ├── assets/         # Images, icons

│   ├── components/     # Navbar, Chart, WorkoutForm

│   ├── redux/          # Redux slices & store

│   ├── pages/          # Dashboard, Workouts, Nutrition, Profile, Settings

│   ├── utils/          # Validators, formatters, custom hooks

│   ├── App.js

│   ├── index.js

│── package.json

**6.  Running the Application**

* **Fronted:**

 Running the application frontedWhen a user accesses FitFlex, the frontend (built using React) loads in the browser and dynamically renders components like login, workout dashboard, and progress charts based on user interaction and data fetched from the backend**.**

* **Backend:**

     On startup, the backend connects to MongoDB using Mongoose. It handles CRUD operations—like storing new user sign-ups, updating workout logs, and retrieving personalized fitness data—by interacting with defined schemas and models.

**7.  API Documentation**

* **User:**

                Register:

                    POST/users/register

                    BODY:{username, email, password }

                    RESPONSE:{message, user Id}

                 Login:

                    POST/users/login

                    BODY:{email, password}

                    RESPONSE:{token}

* **Projects:**

                Create Project:

                    POST/projects

                    Headers:Authorization:Bearer<token>

                    Body:{title, description}

                    Response:{project Id, message}

* **Get Projects:**

                   Get/Projects

                   Headers :Authorization :Bearer<token>

                   Response:[{project Id, title, description}]

* **Chats:**

                 Send Message:

                    POST/chats/message

                    Headers:Authori5:Bearer<token>

                    Body:{projectId,message}

                    Response:{messageId, timestamp}

       Get Messages:

           Get/chats/messages?prohecrId=<id>

           Headers:Authorization:Bearer<token>

           Response:[{messageId, sendId,timestamp}]

**8.  Authentication**

* Authentication in the fitness app user JWT tokens to verify users after they login.

    Users send their token with each request to access protected resources securely.

**9.   User Interface**

           Minimal, clean layout.

           Dashboard with charts and quick stats.

           Mobile-first responsive design.

           Dark & Light theme support.

**10.  Testing**

            Testing Strategy

                Unit Tests → Reducer slice· Component Tests → Dashboard, WorkoutTracker.

                Integration Tests → API calls & state updates.

Code Coverage

               Jest + React Testing Library.

               70–80% coverage targeted.

**11.Screenshorts or Demo**



**12.   Known Issues**

               Many devices miscount steps, calories, or struggle with non-walking exercises like cycling, strength, training, or swimming.

              Trackers often need frequent charging and may run out quickly, especially with continuous tracking features.

               Wearables collect sensitive health and location and location data.

               Constant monitoring can lead to stress, obsessive behaviours, overexercise, or feeling anxious---especially if the numbers become more important than enjoyment.

**13.   Future Enhancements**

               AI-based personalized workout suggestions.

              Integration with wearable devices (Fitbit, Mi Band).

              Social community & leaderboard.

              Export progress data

              Enhance social features for user challenges, leaderboard, sharing progress for motivation and accountability.

              Blend fitness with nutrition tracking, mental, wellness(mindfullness), sleep analysis for comprehensive health insights.