## **Project Documentation**

**Fitflex - Fitness Tracking Application**

**1. Introduction**

**Project Titlle- FitFlex – Fitness Tracking Application**

**TEAM ID: NM2025ID**

**TEAM LEADER:**

**NAME: KARAN C** --CODE

**MAIL ID: mcakaranmca@gmail.com**

**TEAM MEMBRS**

1. **NAME:  RAGHUL K--DOCUMENTATION**

**MAIL ID:raghulkumaravel2006@gmail.com**

1. **NAME:  MANIKANDAN R**

**MAIL ID:mmani249880@gmail.com**

1. **NAME:  DHANUSH K**

**MAIL ID: mathu3309@gmail.com**

1. **NAME: AATHI** **S**

**MAIL ID:aathiaathi88798@gmail.com**

**2. Project Overview**

**Purpose**

FitFlex is a **fitness tracking application** designed to help users monitor their workouts, nutrition, and progress. The main goal is to provide a **simple yet powerful platform** for maintaining a healthy lifestyle.

**Goals**

* Allow users to **log workouts** (e.g., cardio, strength, yoga).
* Track **daily nutrition** and calorie intake.
* Provide a **dashboard** with progress insights.
* Offer an easy-to-use **mobile-friendly UI** for consistency.

**Features**

* Add, edit, and delete workout sessions.
* Track food and calories.
* Dashboard with charts to monitor fitness goals.
* User profile and personalized fitness plans.

**3.Architecture**

**Component Structure**

* **App.js** – Main entry point, routes setup.
* **Navbar.js** – Navigation bar across all pages.
* **Dashboard.js** – Displays statistics, progress, and insights.
* **WorkoutList.js** – Shows logged workouts.
* **WorkoutForm.js** – Add or edit workout details.
* **NutritionTracker.js** – Logs food and calories.
* **Profile.js** – Displays user info and settings.
* **Charts.js** – Reusable chart components for analytics.

**State Management**

* **Context API + useReducer** for global state (workouts, nutrition, user profile).
* **useState** for local form handling and UI interactions.
* Actions: ADD\_WORKOUT, UPDATE\_WORKOUT, DELETE\_WORKOUT, LOG\_NUTRITION.

**Routing**

* / → Dashboard
* /workouts → Workout List
* /add-workout → Add Workout
* /nutrition → Nutrition Tracker
* /profile → User Profile

**4.Setup Instructions**

**Prerequisites**

* Node.js (v16+)
* npm or yarn
* Git

**Installation**

# Clone repository

git clone https://github.com/your-username/fitflex.git

# Move into project folder

cd fitflex

# Install dependencies

npm install

# Start development server

npm start

**5. Folder Structure**

fitflex/

│── public/

│── src/

│   ├── assets/          # Images, icons

│   ├── components/      # Reusable components (Navbar, Charts, Forms)

│   ├── context/         # Global state management (WorkoutContext, NutritionContext)

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

│   ├── utils/           # Helper functions, custom hooks

│   ├── App.js           # Main app

│   ├── index.js         # Entry point

│── package.json

**6.Running the Application**

# Run the frontend

npm start

**7.API Documentation**

**Key Components**

* **WorkoutForm** – Add/edit workout sessions. Props: onSubmit, initialData.
* **WorkoutList** – Lists workouts with options to edit/delete.
* **NutritionTracker** – Logs food, calories, and nutrition details.
* **Charts** – Displays progress graphs (uses Recharts library).
* **Reusable Components**
* **Navbar** – Navigation across all pages.
* **Button** – Styled reusable button.
* **InputField** – Reusable form input component.

**State Management**

**Global State**

* **WorkoutContext**: Stores all workout logs.
* **NutritionContext**: Stores daily food and calorie data.
* **ProfileContext**: Stores user details.

**Local State**

* Form data handling with useState.
* Modal visibility and temporary UI state.

**8.Authentication Features:**

* Email/Password login & signup
* Password encryption using industry-standard hashing
* Session/token-based authentication (e.g., JWT)
* Error handling for invalid credentials
* User-friendly password reset flow

**9.User Interface**

* Dashboard with charts (progress tracker).
* Workout logging form.
* Nutrition logging page.
* Profile section.

**Styling**

* Framework: **Tailwind CSS**.
* Used utility classes for responsiveness.
* Light/Dark mode toggle (theming).

**10.Testing**

**Testing Strategy**

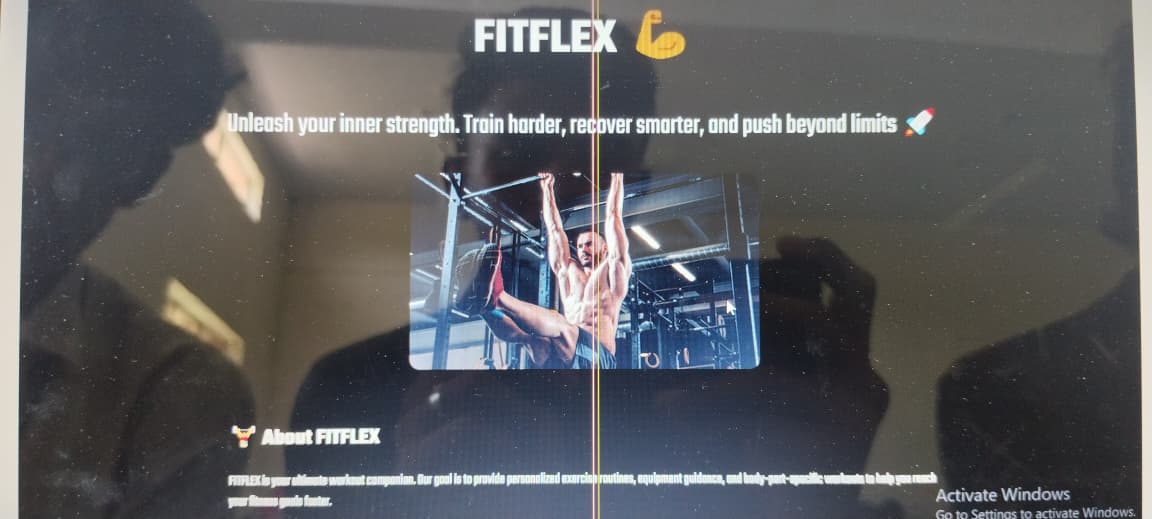
* **Unit Tests** – Reducers and utility functions.
* **Component Tests** – React Testing Library.
* **Integration Tests** – Form submission, state updates.

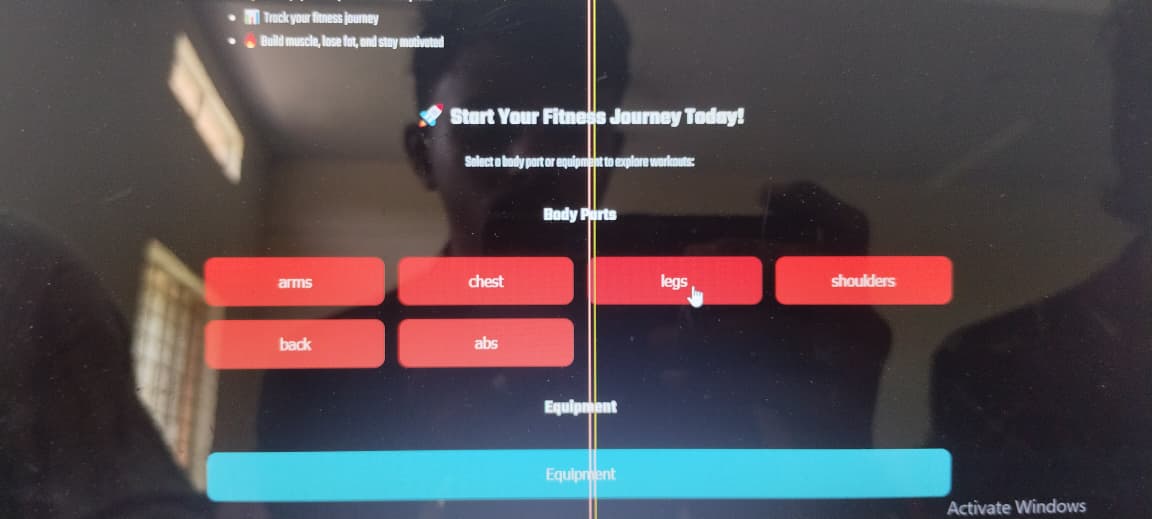
**Code Coverage**

* Reducer actions tested.
* Component rendering tested.

**11.Screenshots or Demo**

(Add screenshots of Dashboard, Workout List, Nutrition Tracker, Profile, or attach demo link.)





**12.Known Issues**

* Chart animations may lag on low-end devices.
* No offline support currently.

**13.Future Enhancements**

* AI-based fitness suggestions.
* Integration with fitness wearables (Fitbit, Apple Watch).
* Offline support with local storage.
* Multi-user login and social features.