### Frontend Development Documentation: ****FitFlex - Your Personal Fitness Companion****

#### 1. Introduction

* **Project Title:** FitFlex - Your Personal Fitness Companion
* **Team Members:**
  + **NATHIYA C: Coding**
  + **NANDHINI G: Coding Development**
  + **KAVYA R: Document**
  + **JAYANTHI R: Demo Video**

#### 2. Project Overview

* **Purpose:**  
  FitFlex is a personal fitness tracking web application built using React.js. The app allows users to track workouts, monitor progress, set fitness goals, and get personalized workout recommendations based on their activity history and preferences.
* **Features:**
  + User authentication and profile management
  + Workout logging (with type, duration, and intensity)
  + Progress tracking with graphical visualizations
  + Goal setting (e.g., steps, calories, workout frequency)
  + Personalized workout suggestions
  + Integration with fitness APIs for data (e.g., heart rate, calories burned)
  + Responsive design for mobile and desktop devices

#### 3. Architecture

* **Component Structure:**  
  The React app is organized into reusable components, such as:
  + **Header** – Displays the navigation and user info.
  + **WorkoutLog** – Handles workout entries, including type, time, and intensity.
  + **ProgressTracker** – Displays the user's progress through graphs and statistics.
  + **GoalSetting** – Lets users set and edit their fitness goals.

These components interact via props, passing data between parent and child components to update the UI dynamically.

* **State Management:**  
  The app uses **Redux** for global state management, with the following slices:
  + **User** – Stores user info and authentication status.
  + **Workout** – Holds workout data, including past entries and progress.
  + **Goal** – Stores fitness goals and their progress.

Local state within components is managed using React’s **useState** hook.

* **Routing:**  
  React Router is used to manage page navigation. Routes include:
  + /login – Login page
  + /dashboard – User’s main dashboard showing workout logs and progress
  + /settings – Settings for user preferences and goals
  + /profile – User profile page

Each route is linked to a corresponding React component that renders the appropriate view.

#### 4. Setup Instructions

* **Prerequisites:**
  + Node.js (version 14 or above)
  + npm (Node Package Manager)
* **Installation:**
  + Clone the repository:
  + git clone https://github.com/your-username/fitflex.git
  + Navigate into the project directory:
  + cd fitflex
  + Install dependencies:
  + npm install
  + Configure environment variables (e.g., API keys, base URLs) in a .env file.
* **Run the Application:**
  + To start the local development server:
  + npm start
  + Navigate to http://localhost:3000 in your browser to view the app.

#### 5. Folder Structure

* **Client Folder:**
* /client
* ├── /public
* ├── /src
* ├── /components # Reusable React components
* ├── /pages # Page components for different routes
* ├── /assets # Static files like images, icons, etc.
* ├── /redux # Redux state management (slices, actions)
* ├── /utils # Helper functions, custom hooks
* ├── /styles # Global styles (CSS, SCSS)
* **Utilities:**
  + **customHooks.js** – Contains custom hooks like useFetchData for API calls.
  + **utils.js** – Helper functions for data formatting and validation.

#### 6. Running the Application

To start the frontend server locally:

npm start

This will start the app in development mode and can be accessed via http://localhost:3000.

#### 7. Component Documentation

* **Key Components:**
  + **WorkoutLog**
    - **Props:**
      * workoutData: Array of previously logged workouts.
      * onSubmit: Function to handle new workout submission.
    - **Description:** Handles input fields for workout type, duration, and intensity. Submits data to the Redux store.
  + **ProgressTracker**
    - **Props:**
      * progressData: Object containing the user’s workout stats.
    - **Description:** Displays the user’s progress through line charts and bar graphs.
* **Reusable Components:**
  + **Button:**
    - **Props:**
      * label: Text to display on the button.
      * onClick: Callback function for button click event.
    - **Description:** A simple reusable button component styled with different variants (primary, secondary).

#### 8. State Management

* **Global State:**  
  State is managed using Redux for cross-component state. The global state includes user info, workout logs, and fitness goals. This state is accessible across all pages.
* **Local State:**  
  Local state is used for temporary UI data, like form inputs or toggle buttons within individual components.

#### 9. User Interface

**Screenshots:**

* **Login Page:**
* **Dashboard:**
* **Progress Tracker:**

#### 10. Styling

* **CSS Frameworks/Libraries:**
  + **Styled-Components**: Used for scoped and dynamic styling.
  + **CSS Modules**: Applied for component-specific styles.
* **Theming:**
  + The app features a dark and light mode, with users able to toggle the theme in the settings.

#### 11. Testing

* **Testing Strategy:**
  + Unit tests for individual components using **Jest**.
  + Integration tests to check interactions between components with **React Testing Library**.
  + End-to-end tests for overall app functionality using **Cypress**.
* **Code Coverage:**  
  Jest’s built-in code coverage tool is used to ensure that tests cover a significant portion of the codebase.

#### 12. Screenshots or Demo

blob:https://web.whatsapp.com/46206cf4-aa4a-4467-b0dc-72340d65c667

#### 13. Known Issues

* **Bug 1:** Graphical progress tracker sometimes fails to load data after a user logs out and logs back in.
* **Bug 2:** On mobile devices, the "goal setting" page may not display properly on very small screens.

#### 14. Future Enhancements

* **Enhancement 1:** Implement a calendar view for scheduling workouts.
* **Enhancement 2:** Add a social component to share achievements with friends.
* **Enhancement 3:** Improve the workout recommendation algorithm with machine learning to personalize recommendations better.