Frontend Development with React.js

# Project Documentation

## 1. Introduction

Project Title: FitFlex – Your Personal Fitness Companion

Team Members:

- Team Leader: Mohammed Muzamil M

- Team Member: Manojraghuram T

## 2. Project Overview

Purpose: FitFlex is a fitness-focused web application designed to help users manage and improve their health by providing a personal fitness companion. The goal of the project is to deliver an intuitive frontend interface where users can access workouts, track their fitness journey, and stay motivated.

Features:

- Interactive fitness dashboard

- User-friendly design with easy navigation

- React-based frontend for fast and responsive UI

- Local development server for real-time updates (http://localhost:3000)

## 3. Architecture

Component Structure:

- Header / Navbar: Navigation across sections

- Home / Dashboard: Displays fitness overview

- Workout Components: Fitness routines and workout details

- Footer: App-related information

State Management: Local state management using React’s useState and useEffect hooks.

Routing: Implemented using React Router to navigate between different pages (e.g., Home, Workouts, About).

## 4. Setup Instructions

Prerequisites: Node.js (LTS version) – https://nodejs.org/en/download/

Installation Steps:

1. Download Node.js and install it.

2. Open PowerShell (Admin) and run: set-executionPolicy unrestricted (then press Y).

3. Download the project folder from SmartInternz portal (Code Drive link).

4. Extract the ZIP file.

5. Open VS Code and open the extracted project folder.

6. In VS Code terminal, run: npm install

7. After dependencies install, run: npm start

8. The application will run at: http://localhost:3000

## 5. Folder Structure

Client:

- components/: Reusable UI components (Navbar, Footer, etc.)

- pages/: Different application pages (Home, Workouts, About)

- assets/: Images, icons, and stylesheets

- App.js: Main React component

- index.js: Entry point

Utilities: Helper functions and custom hooks for state management.

## 6. Running the Application

Frontend: npm start

## 7. Component Documentation

Key Components:

- Navbar: Provides navigation links

- Dashboard: Displays user’s fitness information

- WorkoutList: Lists available workout routines

- Footer: App info and links

Reusable Components: Buttons, cards, and input fields styled for multiple use cases.

## 8. State Management

Global State: Not used (no Redux/Context in PDF). Could be added in future.

Local State: Managed using useState hooks inside components (e.g., toggling menus, handling form inputs).

## 9. User Interface

Modern, responsive UI with navigation bar, pages, and fitness-related visuals.

Screenshots can be added here after running the project.

## 10. Styling

CSS Frameworks/Libraries: Basic CSS (and possibly Bootstrap/Material UI depending on code).

Theming: Custom styles applied per component for consistency.

## 11. Testing

Testing Strategy: Manual testing of frontend components and flows.

Tools: (Future Scope) Jest & React Testing Library for unit and integration testing.

## 12. Screenshots or Demo

After running npm start, capture and insert screenshots of:

- Home page

- Dashboard/Workout list

- Any forms or user interactions

## 13. Known Issues

No backend integration (only frontend).

Limited state management (scales poorly for large apps).

## 14. Future Enhancements

Add authentication and user login.

Integrate APIs for live fitness tracking.

Implement Redux/Context API for global state management.

Add animations and advanced UI/UX improvements.

Mobile responsive optimizations.