Frontend Development with React.js

Project Documentation

# Introduction

Project Title: FitFlex: Your Personal Fitness Companion

Team Leader:  
-GOKULSHREE D

Team Members:  
- JENNISPETER  
- PRAKASH L  
- YASHWIN

-DEVENDIRAN

# Project Overview

Purpose: FitFlex is a personal fitness companion app designed to help users track workouts, monitor progress, and maintain a healthy lifestyle.

Features:   
- User authentication and profile management  
- Workout tracking and progress visualization  
- Personalized fitness recommendations  
- Nutrition and diet tracking  
- Responsive and user-friendly UI

# Architecture

Component Structure: The app uses modular React components including Navbar, Dashboard, Workout Tracker, Diet Planner, and Profile Manager.

State Management: Managed using React Context API for global state and local state within components.

Routing: React Router is used for navigation across pages.

# Setup Instructions

Prerequisites: Node.js, npm

Installation:  
1. Clone the repository  
2. Run `npm install`  
3. Configure environment variables  
4. Run `npm start` to launch the development server

# Folder Structure

Client:  
- /src/components  
- /src/pages  
- /src/assets  
- /src/context  
- /src/utils

# Running the Application

To run locally: Navigate to the client directory and execute:  
`npm start`

# Component Documentation

Key Components:  
- Navbar: Navigation across the app  
- Dashboard: Displays summary of user fitness stats  
- Workout Tracker: Allows adding and viewing workouts  
- Diet Planner: Tracks daily meals and nutrition  
- Profile Manager: User information and preferences

Reusable Components: Buttons, Modals, Cards, and Input Fields.

# State Management

Global State: Managed with Context API for authentication and user data.  
Local State: Managed within individual components for UI interactions.

# User Interface

The UI is responsive and mobile-friendly with intuitive navigation. Includes dashboards, charts, and interactive forms.

# Styling

CSS Frameworks/Libraries: Tailwind CSS for styling.  
Theming: Custom themes with reusable design tokens.

# Testing

Testing Strategy: Unit tests and integration tests with Jest and React Testing Library.  
Code Coverage: Ensured with Jest coverage tools.

# Screenshots or Demo

Screenshots and demo links will be provided showcasing the application features.

# Known Issues

Currently, the app may face performance issues with large datasets.

# Future Enhancements

Planned features include AI-powered workout recommendations, social features, and wearable device integration.