Frontend Development with React.js

Project Documentation format

FitFlex - React Frontend Documentation

1. Introduction

Project Title: FitFlex

Team leader

R.Sandhiya

Team Members:

- M.Sandhya

- Nivitha.T

--Sangeetha.H

## 2. Project Overview

Purpose:

FitFlex is a modern fitness application that helps users track workouts, set goals, and monitor progress. The frontend provides an intuitive interface for seamless user interactions.

Features:

- User authentication (Login/Signup)

- Dashboard with workout tracking

- Exercise catalog with filtering options

- Personalized fitness plans

- Progress tracking with charts

- Dark/Light mode toggle

-

3. Architecture

Component Structure:

- App.js – Root component managing routes and state
- Pages/ – Contains main pages (Home, Dashboard, Profile)
- Components/ – Reusable UI components (Navbar, Button, Card)

State Management:

- *Global State:* Managed using *Context API* for authentication and user data
- *Local State:* Managed via React's useState within individual components

Routing

- Implemented with *React Router v6*
- Routes defined in App.js and protected using private route wrappers

4. Setup Instructions

Prerequisites:

- Node.js (v16+)
- npm or yarn

Installation:

sh

```
git clone https://github.com/your-repo/fitflex.git
cd fitflex
npm install
```

Environment Variables:

Create a .env file in the root and add:

```
REACT_APP_API_URL=https://your-api-url.com
```

5. Folder Structure

```
/fitflex
|── /src
|   |── /components
|   |── /pages
|   |── /assets
|   |── /context
|   |── /hooks
|   |── /styles
|   |── App.js
|   |── index.js
|── package.json
|── .env
|── README.md
```

Utilities:

- hooks/ – Custom hooks for fetching data, authentication, etc.
- context/ – Global state management using Context API

6. Running the Application

sh

```
npm start  # Runs on http://localhost:3000
```

7. Component Documentation

Key Components:

- Navbar.js – Handles navigation
- WorkoutCard.js – Displays individual workout details
- Dashboard.js – Main user interface for tracking fitness progress

Reusable Components:

- Button.js – Custom button component with variants
- Modal.js – Generic modal component

8. State Management

Global State:

- Context API manages authentication and user settings

Local State:

- Used within components for toggling UI elements (e.g., modal visibility)

9. User Interface

Screenshots showcasing:

- Home page
- Workout dashboard
- Profile settings

10. Styling

CSS Frameworks/Libraries:

- *Tailwind CSS* – For responsive design
- *Styled Components* – For dynamic styling

Theming:

- Supports dark and light mode

11. Testing

Testing Strategy:

- *Jest & React Testing Library* for unit and integration tests

Code Coverage:

- Ensured with Jest and coverage reports

12. Screenshots or demo link

https://drive.google.com/file/d/1-8FhtZU_khPSTdbbp0USRKWh8KhZTEbg/view?usp=drivesdk

13. Known Issues

- Mobile responsiveness needs improvement on some screens
- Workout progress charts need optimization

## 14. Future Enhancements

- Add AI-based personalized workout suggestions
- Implement push notifications for reminders
- Enhance animations and transitions