

Project Documentation

Introduction

Project Title: FitFlex

Team Lead:

Faikha Thabasum. S Email: s.faikhathabasum@gmail.com

Team Members:

Sushvamitha. Y Email: sushmasaran672@gmail.com

Bairavi. S Email: bairavis765@gmail.com

Jeevitha. M Email: mjeevitha0105@gmail.com

Hemasri. R Email: ravihemasri32@gmail.com

Project Overview

Purpose:

FitFlex is a personalized fitness and wellness application designed to provide users with workout recommendations, exercise tutorials, and progress tracking. The goal is to encourage users to maintain a healthy lifestyle by integrating fitness routines into their daily schedules.

Features:

- User login and profile management.
- Personalized workout recommendations.
- Exercise library with descriptions and images.
- Daily and weekly fitness goals tracking.

- BMI calculator for quick health insights.
- Responsive design for mobile and desktop

Architecture

Component Structure:

- **App.js** : Root component, manages routing and global layout.
- **Header & Footer** : Navigation and app branding.
- **Dashboard** : Displays user summary, goals, and recommended workouts.
- **WorkoutList** : Lists all available exercises.
- **WorkoutDetail** : Shows details of a specific exercise.
- **ProgressTracker** : Tracks progress, goals, and completed sessions.
- **BMI Calculator** : Calculates BMI based on user input.
- **Profile** : Manages user details and settings.

State Management:

- **Local State** : Managed using React useState and useEffect for small component-level states (e.g., input forms, toggles).
- **Global State** : Implemented using Context API for authentication and user profile data.
- **Async State** : Handled with Redux Toolkit for workouts, progress tracking, and fetching exercise data.
- **Persistence** : User data stored in browser's localStorage for quick retrieval.

Routing:

Implemented using React Router v6.

Routes include:

- /login — User authentication

- /dashboard — Main dashboard
- /workouts — Workout library
- /workouts/:id — Workout detail page
- /progress — Track goals and performance
- /profile — User profile settings

Setup Instructions

Prerequisites:

- Node.js (v16 or above)
- npm or yarn package manager

Installation:

```
git clone https://github.com/username/fitflex.git
```

```
cd fitflex
```

```
npm install
```

```
npm start
```

Folder Structure

```
fitflex/
```

```
| — public/
```

```
|   | — index.html
```

```
|   | — assets/ (images, icons, etc.)
```

```
|
```

```
| — src/
```

```
|   | — components/
```

```
|   |   | — Header.js
```

```
| | └─ Footer.js
| | └─ BMI.js
| | └─ ProgressTracker.js
| |
| └─ pages/
| | └─ Dashboard.js
| | └─ Workouts.js
| | └─ WorkoutDetail.js
| | └─ Profile.js
| |
| └─ context/
| | └─ AuthContext.js
| |
| └─ redux/
| | └─ store.js
| | └─ workoutSlice.js
| |
| └─ styles/
| | └─ main.css
| |
| └─ App.js
|
```

└─ package.json

Running the Application

npm start

This will start the app on <http://localhost:3000/>.

Component Documentation

Key Components:

- Dashboard.js — Displays summary, BMI status, and workout suggestions.
- WorkoutList.js — Fetches and displays exercises with images.
- WorkoutDetail.js — Shows tutorial, benefits, and reps.
- ProgressTracker.js — Visual progress chart of completed vs. pending tasks.
- Profile.js — Handles user data editing.

Reusable Components:

- Button.js — Custom button component with styling.
- Card.js — Reusable card for workouts and stats.
- InputField.js — Common input field for forms

State Managemen

Global State (Context API):

- Authentication (login/logout).
- User profile.

Redux Toolkit:

- Workout data storage.
- Progress tracking.

Local State:

- Form inputs (login form, BMI form).
- Theme toggle (light/dark mode).

User Interface

- Modern dashboard view for workouts and goals.
- Interactive cards for exercises.
- Charts and graphs for progress tracking.
- Mobile-first design for accessibility.

Styling:

- CSS Frameworks: Tailwind CSS.
- Custom Styling: CSS modules for component-level styles.
- Theming: Supports light and dark mode.

Testing

- Unit Testing: Jest for functions and reducers.
- Component Testing: React Testing Library.
- Integration Testing: Ensures correct API and Redux interaction.
- Code Coverage Tool: Istanbul.

Screenshots or Demo

(Add screenshots of dashboard, workout pages, and progress tracker here.)

Known Issues

- Requires stable internet connection for API-based workout data.
- Dark mode not fully styled for all components

Future Enhancements

- AI-powered workout recommendation system.
- Voice-assisted exercise guidance.

- Gamification features (reward points, badges).
- Integration with wearable fitness devices.