FitFlex: Your Personal Fitness Companion

# Introduction

Project: FitFlex: Your Personal Fitness Companion

Team Members:

* Team Leader: Anish M
* Team Member 1: Gokul S
* Team Member 2: Kavin J
* Team Member 3: Krishdivakar P

Project Overview

Purpose

The primary goal of FitFlex is to provide a personalized and interactive fitness platform for users to track workouts, follow exercise routines, and stay motivated on their health journey.

Features

* 🧭 Intuitive interface for discovering and saving workouts
* 🔍 Advanced search to find exercises based on type, muscle group, or difficulty
* 📅 Custom workout routine creation and tracking
* 📱 Responsive design optimized for all devices
* ⏱️ Real-time activity management with single routine tracking at a time

Architecture

Component Structure

* App → Root container with routing and layout
* Workouts → Displays and fetches exercise items
* Favorites → Manages saved workouts
* Routines → Handles workout routine creation and tracking

State Management

* Local State: Managed using useState and useEffect for temporary states like search filters and active workout tracking
* Global State: Workout routines and favorites are synchronized with a JSON server via Axios

Routing

Implemented using react-router-dom with routes:

* / → Home / Workouts
* /favorites → Favorite Workouts
* /routines → Workout Routines

Setup Instructions

Prerequisites

* Node.js and npm
* Git
* Code editor (Visual Studio Code recommended)

Installation Steps

1. Clone the repository
2. Navigate into the project directory
3. Run npm install
4. Start the frontend with npm start
5. Start the mock backend with:
6. json-server --watch ./db/db.json

Folder Structure

* Client: Structured into components/, pages/, assets/, utils/
* Utilities: Includes Axios for API requests and helper functions for workout management

Running the Application

* Frontend: Run with npm start
* Backend (Mock Server): Run with json-server --watch ./db/db.json

Component Documentation

Key Components

* App → Main application container managing layout and routing
* Workouts → Lists and filters exercises fetched from the backend
* Favorites → Allows users to save and manage favorite workouts
* Routines → Enables creation and monitoring of personalized workout routines

State Management

* Global State: Workout routines and favorites handled via useState and synced with backend using Axios
* Local State: Filters, selected difficulty level, currently active workout, and UI-related temporary states

User Interface

* Modern, responsive design using Bootstrap and Tailwind CSS
* Sidebar navigation for seamless access to features
* Interactive workout cards with action icons (Add to Routine / Favorite)
* Mobile-friendly layout and touch-friendly components

Stylin

* CSS Frameworks: Bootstrap + TailwindCSS
* Theming: Consistent and reusable style system across components

Testing

Strategy

* Unit and integration tests developed using Jest and React Testing Library

Coverage

* Automated tests cover key components and user interactions to ensure app stability

Screenshots or Demo

* Demo Video: Click Here
* Project Repository: Click Here

Known Issues

* Minor sync delay when rapidly modifying both favorites and routines due to JSON Server limitations

Future Enhancements

* Add user authentication and profile customization
* Integrate a real-time API and database (e.g., Firebase, MongoDB)
* Include offline tracking and reminders
* Incorporate AI-based smart workout suggestions based on user goals