**FRONTEND DEVELOPMENT WITH REACT.JS**

1. **INTRODUCTION**

**Project Title:** Fit Flex Fitness App

**Team Members:**

Santhosh K (Team Leader) - [santhoshk-cswithai@srmasc.ac.in](mailto:santhoshk-cswithai@srmasc.ac.in)

Harish Karthikeyan T - [vetrivelm-cswithai@srmasc.ac.in](mailto:vetrivelm-cswithai@srmasc.ac.in)

Vetrivel M - [prabhakaranb-cswithai@srmasc.ac.in](mailto:prabhakaranb-cswithai@srmasc.ac.in)

Premkumar A - [premkumara-cswithai@srmasc.ac.in](mailto:premkumara-cswithai@srmasc.ac.in)

Prabhakaran B - [t.harishkarthikeyan-cswithai@srmasc.ac.in](mailto:t.harishkarthikeyan-cswithai@srmasc.ac.in)

**2. PROJECT OVERVIEW**

**Purpose:**  
 The **Fit Flex Fitness App** is an interactive platform designed to help users explore various fitness exercises and workouts. The app focuses on providing simple and effective ways for users to discover exercises based on body parts or equipment. Each workout is paired with a corresponding **YouTube tutorial video** to ensure users can follow along with proper form and technique.

The app is built using **React.js** and integrates data from **Express DB -> Fitness API** to dynamically fetch and display exercises. It aims to deliver a smooth and responsive experience, allowing users to search, view, and learn workouts without unnecessary complexity.

**Features:**

1. User-friendly interface for easy navigation and Responsive design.
2. Categorized workouts by body parts and equipment
3. Real-time exercise data fetched from Express DB API
4. Integrated YouTube videos for guided workout tutorials
5. **ARCHITECTURE**

**Component Structure:**

1. **Navbar.jsx:** Handles navigation across Home, Exercises, and Categories.
2. **Footer.jsx:** Displays contact information and social links.
3. **Hero.jsx:** Highlights featured exercises or motivational banners.
4. **HomeSearch.jsx:** Provides a search bar for users to find workouts quickly.
5. **BodyPartsCategory.jsx:** Lists exercises based on targeted body parts.
6. **EquipmentCategory.jsx:** Displays workouts using specific equipment.
7. **Exercise.jsx:** Shows detailed information about selected exercises and embedded YouTube videos.
8. **Home.jsx:** Main homepage component displaying workout categories.

**State Management:**

* **Local State:** Utilizes React's useState for managing component-specific data like search inputs and toggle states.
* **Global State:** (Planned) Future integration of Context API for consistent state flow across components.

**Routing:**

* **react-router-dom** handles client-side routing, enabling smooth navigation without page reloads.

1. **SETUP INSTRUCTIONS**

**Prerequisites:**

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

**Installation:**

1. Clone the repository: git clone [repository-url]
2. Navigate to the project directory: cd fitflex-fitness-app
3. Install all dependencies: npm install
4. Run the development server: npm start
5. **FOLDER STRUCTURE**

**Client:**

/src

├── assets/ # Images, fonts, and static files

├── components/ # Reusable UI components (Navbar, Footer, etc.)

├── pages/ # Page components (Home, Exercise, Categories)

├── styles/ # Component-specific CSS

├── App.js # Root component containing routes

├── index.js # Renders the app into the DOM

**Utilities:**

* **api.js:** Contains fetch requests to the ExpressDb API.
* **helpers.js:** (Future scope) For common utility functions.

1. **RUNNING THE APPLICATION**

To start the frontend server, follow these steps:

1. Open the terminal.
2. Navigate to the project directory:

bash

Copyedit

cd your-project-folder

1. Install dependencies (if not already installed):

bash

Copyedit

npm install

1. Start the development server:

bash

Copyedit

npm start

1. Access the application:  
   Once the server starts, the application will usually be available at:

arduino

CopyEdit

http://localhost:3000

1. Stop the server:  
   To stop the running server, use:

mathematica

CopyEdit

Ctrl + C

1. **COMPONENT DOCUMENTATION**

**Key Components:**

1. **Navbar.jsx:** Navigates between Home, Categories, and Exercise pages.
2. **Footer.jsx:** Displays app information and social media links.
3. **Hero.jsx:** Promotes key exercises and encourages user engagement.
4. **HomeSearch.jsx:** Filters workouts by body parts and equipment.

**Reusable Components:**

1. **Button.jsx:** Customizable buttons for user interactions.
2. **Card.jsx:** Displays individual exercise details in a uniform style.
3. **STATE MANAGEMENT**

**Local State**

Local state is used to manage data and UI behaviour within a specific component. It is confined to the component where it is declared and cannot be accessed directly by other components.

**Use cases:**

* **Search input handling:** Storing and updating the value entered by the user in a search bar.
* **Modal visibility:** Tracking whether a modal (popup) is open or closed.
* **Exercise filters:** Storing user-selected filter options to show relevant exercises.
* **Global State:** (To be added) Plans for Context API to manage shared data between components.

**Global State**

Global state is used when data needs to be shared and accessed by multiple components, ensuring consistency across the application.

**Implementation plans:**

* **Context API:** Will be used to manage shared data such as user authentication status, selected filters, and theme preferences.

1. **USER INTERFACE**

**Screenshots:**



A screenshot of a computer

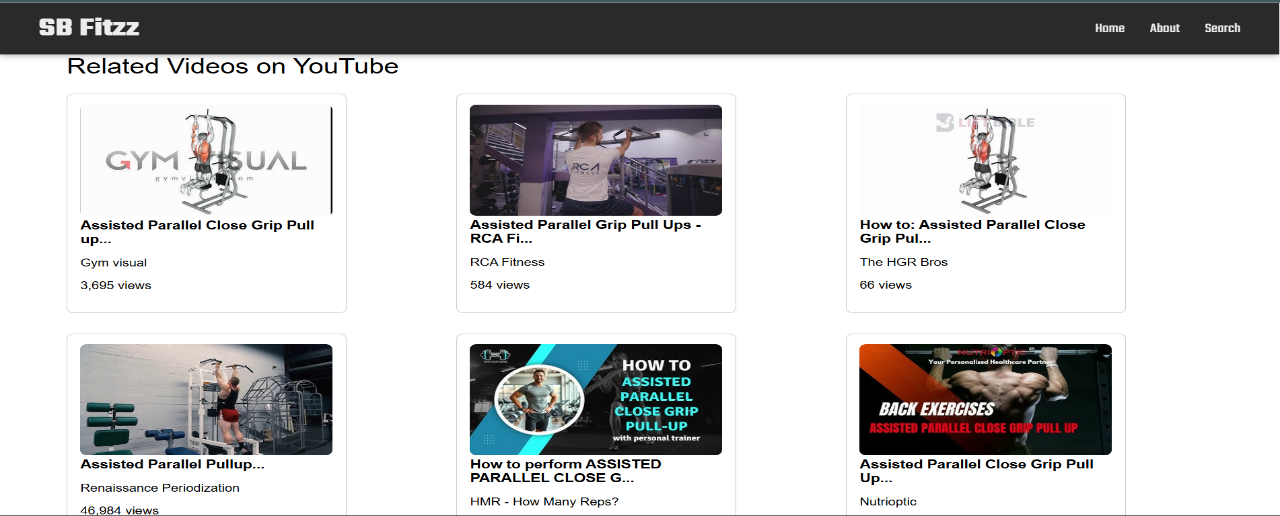
AI-generated content may be incorrect.

Screenshot of a screenshot of a cellphone

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



Screens screenshot of a cellphone

AI-generated content may be incorrect.

A screenshot of a cellphone

AI-generated content may be incorrect.

Screenshot of a screenshot of a cellphone

AI-generated content may be incorrect.

**10. STYLING**

**CSS Frameworks:**

* Plain CSS with modularized styles in the styles folder.

**Theming:**

* Currently supports light mode, future plans for dark mode.

**11. TESTING**

**Testing Strategy:**

* (Planned) Unit tests using **Jest**.
* Component testing with **React Testing Library**.

**Code Coverage:**

* (Future Scope) Integration of Istanbul for visualizing test coverage.

**12. DEMO**

**Demo Link:**

**https://drive.google.com/file/d/1ll7Tl\_z\_kpa1Jq1WzY8GcnO5Oeq7wr2d/view?usp=drive\_link**

**13. KNOWN ISSUES**

1. Exercise filters need further refinement for seamless user experience.
2. No global state management currently implemented.
3. Lack of user authentication or personalized content.

**14. Future Enhancements**

The following features are planned for future iterations of the Fit Flex Fitness App:

1. **Workout List Display:** Enhance the display of exercises by adding filters for muscle groups and equipment, allowing users to easily browse workouts.
2. **YouTube Video Integration:** Improve the integration of YouTube workout videos by embedding customizable video players within the app to offer seamless exercise guidance.