**Frontend Development with React.js**

**Project Documentation**

**1. Introduction**

* **Project Title**: FitFlex
* **Team Members**: Vicky R(Leader) email:vicky20041605@gmail.com, Vijayram J(Member) email:vijayram02062004@gmail.com, Mukesh P(Member) email:mukesh233106@gmail.com, Velmurugan E(Member) email:pradeshvel10@gmail.com.

**2. Project Overview**

* **Purpose**:  
  FitFlex is a fitness app designed to enhance workout experiences through an intuitive interface, dynamic search, and a comprehensive exercise library. It aims to provide an engaging platform for users to explore fitness routines, track progress, and stay motivated.
* **Features**:
  + Access to a wide variety of exercises from a fitness API
  + Visual exercise exploration with images and videos
  + Advanced search functionality
  + Intuitive and user-friendly UI

**3. Architecture**

* **Component Structure**:
  + The app is structured into **Pages, Components, and Styles** folders.
  + Pages handle different views like home, categories, and exercise details.
  + Components contain reusable UI elements such as Navbar, Search, and Exercise Cards.
* **State Management**:
  + The application uses **React Hooks** for local state management.
  + API responses are stored using useState and managed with useEffect.
* **Routing**:
  + Implemented with react-router-dom to navigate between pages.

**4. Setup Instructions**

* **Prerequisites**:
  + Node.js
  + npm or yarn
  + Git
* **Installation**:
  + Clone the repository:
  + git clone <repo\_url>
  + Navigate into the directory:
  + cd fitness-app-react
  + Install dependencies:
  + npm install
  + Start the development server:
  + npm start

**5. Folder Structure**

* **Client Folder**:
  + /components: Contains reusable UI components like Navbar, Search, Exercise Cards.
  + /pages: Contains major pages such as Home, Category, Exercise Details.
  + /styles: Holds CSS or styling frameworks like TailwindCSS or Bootstrap.
* **Utilities**:
  + Custom hooks for API handling.
  + Helper functions for filtering and processing API data.

**6. Running the Application**

* **To start the frontend server**:
* npm start
* Open http://localhost:3000 in the browser.

**7. Component Documentation**

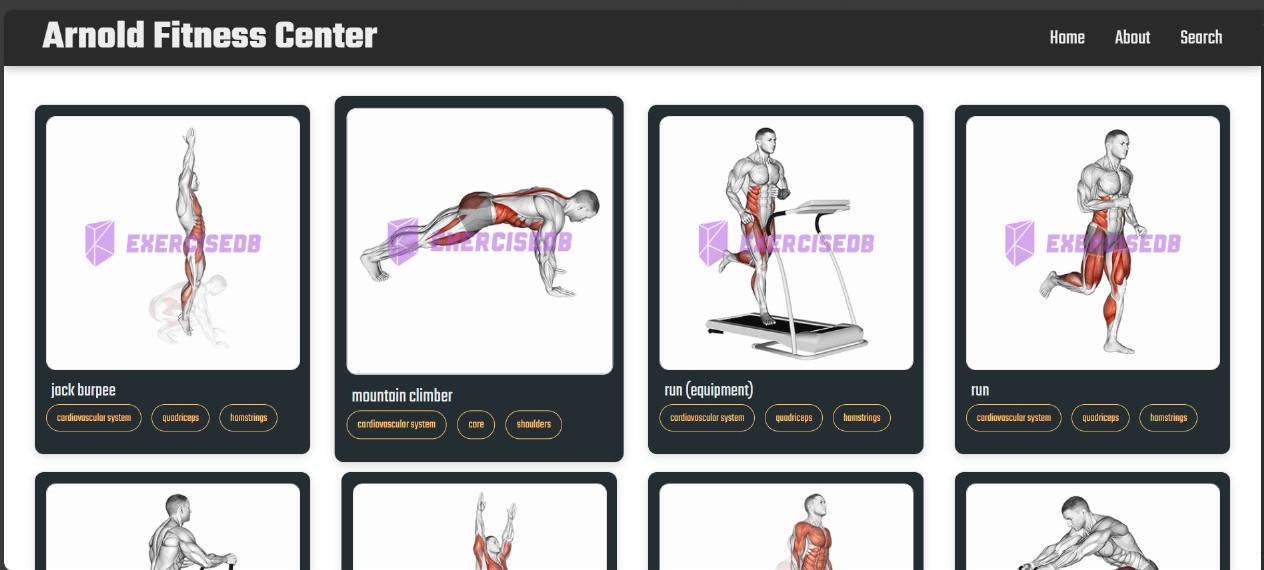
* **Key Components**:
  + **Navbar**: Navigation bar for switching between different sections.
  + **SearchBar**: Allows users to search for exercises.
  + **ExerciseCard**: Displays exercise details with images and descriptions.
* **Reusable Components**:
  + **Button**: Styled button for consistency across UI.
  + **Loader**: Loading spinner for API calls.

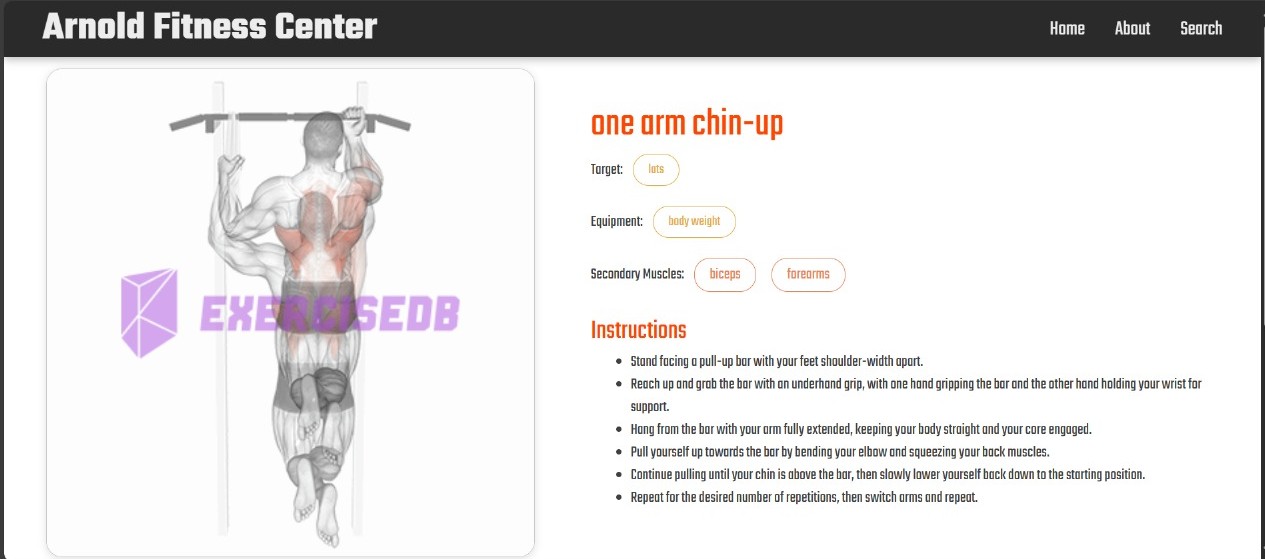
**8. State Management**

* **Global State**:
  + Not using a dedicated state management library, relying on React Hooks instead.
* **Local State**:
  + useState for managing API data and search inputs.
  + useEffect for fetching data from the API.

**9. User Interface**

****

****

****

**10. Styling**

* **CSS Frameworks/Libraries**:
  + Bootstrap or Tailwind CSS for styling components.
* **Theming**:
  + Custom theming with consistent colors and typography.

**11. Testing**

* **Testing Strategy**:
  + Unit tests for individual components using Jest.
  + Integration tests with React Testing Library.
* **Code Coverage**:
  + Ensured by writing tests for key UI elements and API calls.

**12. Screenshots or Demo**

* **Live Demo Link**[: https://drive.google.com/file/d/19QH8ZzL-\_yOdyzRLI36ur8cdjlXKJf5j/view?usp=drive\_link](:%20https:/drive.google.com/file/d/19QH8ZzL-_yOdyzRLI36ur8cdjlXKJf5j/view?usp=drive_link)

**13. Known Issues**

* API rate limits can affect data retrieval.
* Some exercises may not have complete details or images.

**14. Future Enhancements**

* **User authentication** for personalized workout tracking.
* **Workout planner** to create custom exercise routines.
* **Dark mode** for better user experience.