**Frontend Development with React.js**

**Project Documentation**

1. **Introduction**

**Project Title:**

FitFlex – Your Personal Fitness Companion

**Team Members:**

* Kamalesh. K – Lead Developer

Email: kamaleshvishal611@gmail.com

* Aaron. K – Frontend Developer

Email: aaron2315aaron@gmail.com

* Gokul. S – API Integration & Backend Support

Email: gokulssilver@gmail.com

* Kumaran. M – Testing & Debugging Specialist

Email: kumarkumarks848@gmail.com

1. **Project Overview**

**Purpose:** FitFlex is designed to provide a personalized fitness experience with dynamic exercise recommendations, a user-friendly interface, and video tutorials. Users can explore workout categories, search exercises, and receive guided workout plans.

**Features:**

* Search and filter exercises by body part, equipment, and difficulty level.
* Fetch real-time exercise data from RapidAPI.
* View detailed exercise instructions and related YouTube videos.
* Responsive and user-friendly interface with interactive elements.

1. **Architecture**

**Component Structure**

FitFlex follows a modular component-based architecture:

* **Navbar.jsx** – Handles navigation across different pages.
* **Home.jsx** – Displays trending workouts and the search bar.
* **Exercise.jsx** – Fetches and displays detailed exercise data.
* **BodyPartsCategory.jsx & EquipmentCategory.jsx** – Show exercises filtered by body part or equipment.
* **SearchBar.jsx** – Allows users to search exercises dynamically.

**State Management**

* **Local State:** Handled using React’s useState for managing component-specific data.
* **Global State:** Managed using Context API to store exercise data across pages.

**Routing**

* **React Router DOM** is used for navigation.
* Routes include:
  + / → Home Page
  + /bodyPart/:id → Exercises by Body Part
  + /equipment/:id → Exercises by Equipment
  + /exercise/:id → Detailed Exercise Page

1. **Setup Instructions**

**Prerequisites**

* Node.js & npm – [Download](https://nodejs.org/en/download/)
* Git for version control – [Download](https://git-scm.com/downloads)
* Code Editor (VS Code recommended) – [Download](https://code.visualstudio.com/download)

**Installation**

1. **Clone the Repository:**
2. git clone [Kamaleshkabeerdoss/FitnessApp.Using-react-](https://github.com/Kamaleshkabeerdoss/FitnessApp.Using-react-)
3. cd fitness-app
4. **Install Dependencies:**
5. npm install
6. **Start the Application:**
7. npm start
   * The application will run at http://localhost:3000
8. **Folder Structure**

/fitness-app

│── /src

│ ├── /components # Reusable components like Navbar, SearchBar

│ ├── /pages # Pages like Home, ExerciseDetails

│ ├── /styles # CSS stylesheets

│ ├── App.js # Main App component

│ ├── index.js # Entry point

│── /public # Static assets

│── package.json # Dependencies

│── README.md # Documentation

**Utilities**

* **API Requests:** Implemented using Axios for fetching exercise data.
* **Custom Hooks:** Created for handling API calls and state management.

1. **Running the Application**

To start the frontend server locally, use:

npm start

1. **Component Documentation**

**Key Components**

* **Navbar.jsx** – Contains navigation links.
* **Home.jsx** – Displays featured workouts and the search bar.
* **Exercise.jsx** – Fetches and displays exercise details.
* **SearchBar.jsx** – Allows users to search exercises dynamically.

**Reusable Components**

* **ExerciseCard.jsx** – Displays individual exercise cards.
* **CategoryFilter.jsx** – Allows filtering by body part or equipment.

1. **State Management**

**Global State:**

* Uses Context API to store and pass exercise data across components.

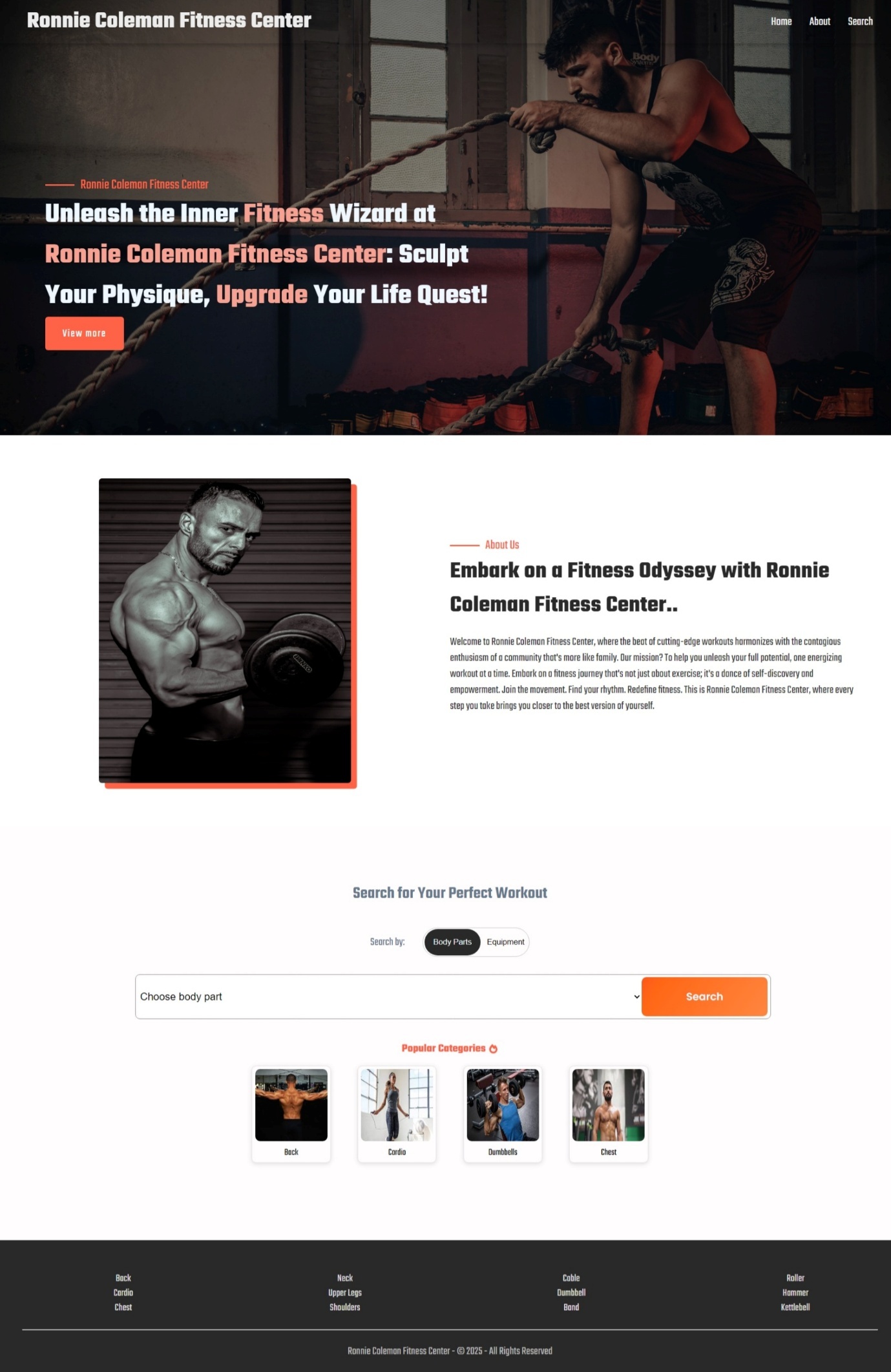
**Local State:**

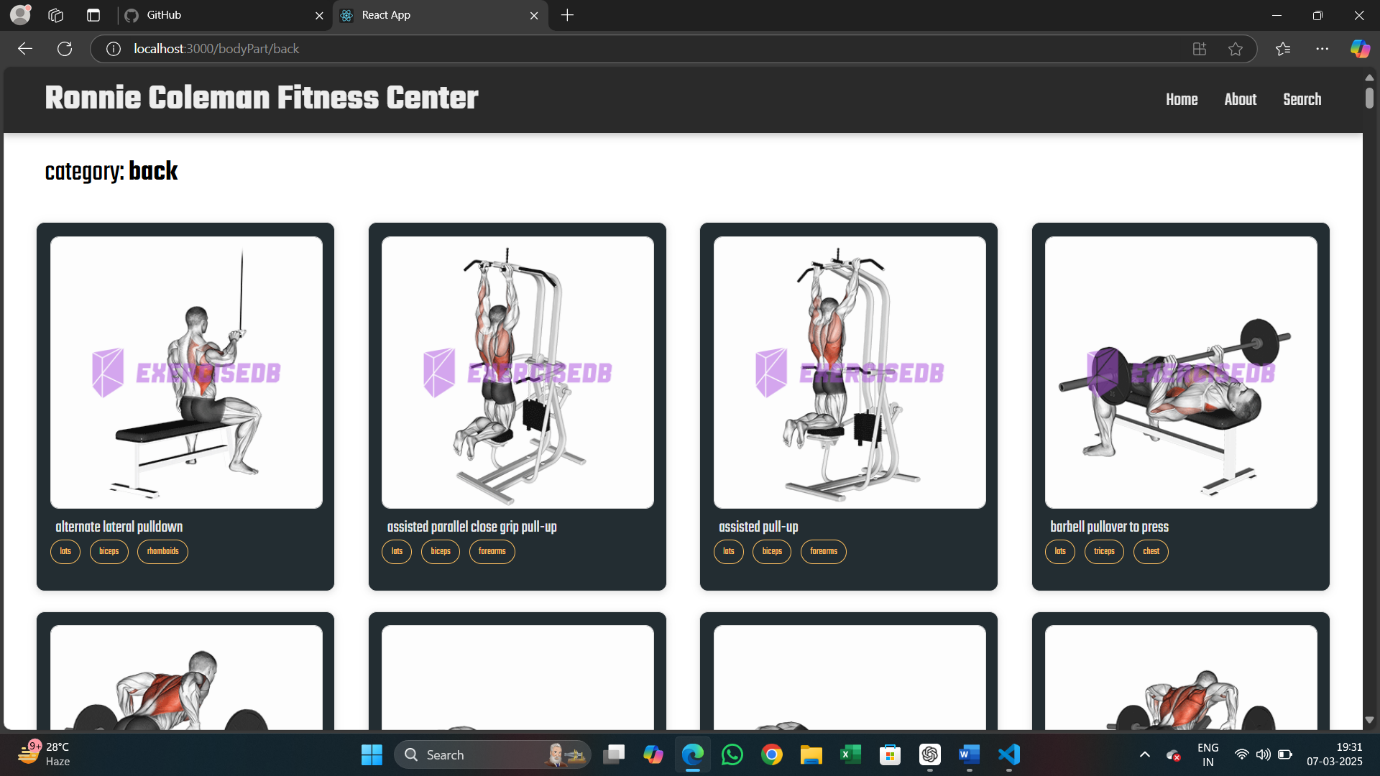
* Managed using useState hooks for component-specific data.

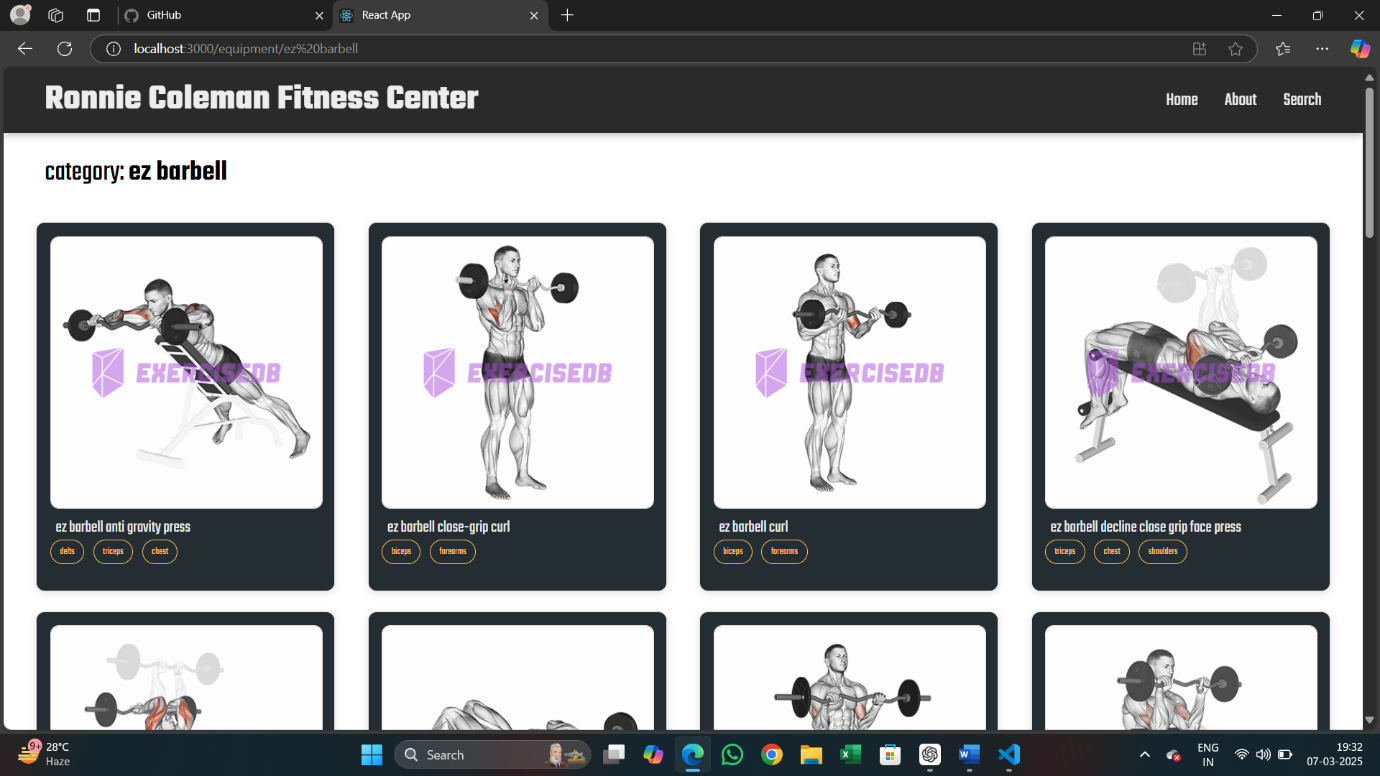
1. **User Interface**

**UI Flow & Screenshots**

1. **Landing Page: Users start at the home page, where they can explore trending workouts, featured exercise plans, and use the search bar for quick navigation. The page includes a call-to-action (CTA) for signing up or logging in to track progress.**
2. **Search Feature: Users can enter keywords to filter exercises by body part, equipment, difficulty level, or workout goals (e.g., weight loss, muscle gain). The search results update dynamically with real-time suggestions.**
3. **Category Selection: Users can browse exercises grouped under body parts (e.g., chest, back, legs) or equipment categories (e.g., dumbbells, resistance bands). A filter option allows users to refine their selection further.**
4. **Exercise Details Page: Clicking an exercise card opens a detailed page that includes:**
   * **Workout steps: Step-by-step instructions with images or GIFs.**
   * **Video Tutorials: Embedded YouTube videos demonstrating proper form.**
   * **Targeted Muscles: Visual representation of muscle engagement.**
   * **Difficulty Level & Tips: Recommendations for beginners, intermediate, or advanced users.**
5. **Navigation: A responsive navbar provides quick access to different sections such as Home, Categories, Workout Plans, and Profile. A floating action button (FAB) allows users to quickly add exercises to their routine.**
6. **Community & Social Sharing: Users can share their workout progress, completed exercises, or personal achievements with friends via social media integration.**
7. **Responsive Design: The UI adapts dynamically across desktop, tablet, and mobile screens, ensuring a seamless experience. Dark mode support enhances user comfort during late-night workouts.**

****

****

****

1. **Styling**

* **CSS Frameworks:** Tailwind CSS & Bootstrap.
* **Theming:** Custom styles with global CSS variables for consistency.

1. **Testing**

**Testing Strategy**

* **Unit Testing:** Using Jest for testing individual components.
* **Integration Testing:** Verifying API calls using Mock APIs.
* **End-to-End Testing:** Using Cypress for UI testing.

**Code Coverage**

* Jest coverage report ensures all major functions are tested.

1. **Demo Video**

**LINK :** <https://drive.google.com/drive/folders/1qcC9CqP87RvbvMHxaHlZiDl7lYLkX0yr?usp=sharing>

1. **Known Issues**

* API rate limits may cause occasional slowdowns.
* Some exercises lack video tutorials due to missing YouTube data.

1. **Future Enhancements**

* **User Authentication**: Allow users to save workouts.
* **Workout Tracking**: Users can log completed workouts.
* **Expanded Database**: More exercises and custom workout plans.

**Conclusion**

FitFlex successfully integrates **React.js, APIs, and modern UI components** to provide an interactive fitness experience. The app efficiently fetches exercise data and presents users with engaging workouts and video tutorials.

**References**

* [React.js Official Docs](https://react.dev/learn/installation)
* [RapidAPI Documentation](https://rapidapi.com/)
* [YouTube API Guide](https://developers.google.com/youtube)