**FITFLEX: YOUR PERSONAL FITNESS COMPANION**

**Project Title: FITFLEX**

**Team Members**:

**Kirthiga shree k** (**Team Leader**) [Email Id: [iaskirthigashree@gmail.com](mailto:iaskirthigashree@gmail.com) ]

 **Jayashree K** [Email Id: [k.jayashree0706@gmail.com](mailto:k.jayashree0706@gmail.com) ]

**Lavanya J** [Email Id: [lavanyajayaraman33@gmail.com](mailto:lavanyajayaraman33@gmail.com) ]

**INTRODUCTION**

FitFlex is a revolutionary fitness app designed to transform your workout experience. It offers an intuitive interface, dynamic search, and a vast library of exercises for all fitness levels. Join FitFlex to embark on a personalized fitness journey and achieve your wellness goals.

**DESCRIPTION**

"Welcome to the forefront of fitness exploration with FitFlex! Our innovative fitness app is meticulously designed to revolutionize the way you engage with exercise routines, catering to the diverse interests of both fitness enthusiasts and seasoned workout professionals. With a focus on an intuitive user interface and a comprehensive feature set, FitFlex is set to redefine the entire fitness discovery and exercise experience.

Crafted with a commitment to user-friendly aesthetics, FitFlex immerses users in an unparalleled fitness journey. Effortlessly navigate through a wide array of exercise categories with features like dynamic search, bringing you the latest and most effective workouts from the fitness world.

From those embarking on their fitness journey to seasoned workout aficionados, FitFlex embraces a diverse audience, fostering a dynamic community united by a shared passion for a healthy lifestyle. Our vision is to reshape how users interact with fitness, presenting a platform that not only provides effective exercise routines but also encourages collaboration and sharing within the vibrant fitness community.

Embark on this fitness adventure with us, where innovation seamlessly intertwines with established exercise principles. Every tap within FitFlex propels you closer to a realm of diverse workouts and wellness perspectives. Join us and experience the evolution of fitness engagement, where each feature is meticulously crafted to offer a glimpse into the future of a healthier you."

"Elevate your fitness exploration with FitFlex, where every exercise becomes a gateway to a world of wellness waiting to be discovered and embraced. Trust FitFlex to be your reliable companion on the journey to staying connected with a fit and active lifestyle."

**SCENARIO-BASED INTRO**

You lace up your sneakers, determined to get serious about your fitness. But where do you start? Suddenly, you remember FitFlex, the innovative app that promised to revolutionize your workouts. With a tap, you open the app. Vibrant visuals flood the screen—personalized workout plans, diverse exercise categories, and a supportive community. This isn't your typical fitness app. FitFlex feels...different. Intrigued, you select a workout and get ready to experience the future of fitness.

**PROJECT GOALS AND OBJECTIVES**

The overarching aim of FitFlex is to offer an accessible platform tailored for individuals passionate about fitness, exercise, and holistic well-being.

**Our key objectives are as follows**:

* **User-Friendly Experience:** Develop an intuitive interface that facilitates easy navigation, enabling users to effortlessly discover, save, and share their preferred workout routines.
* **Comprehensive Exercise Management:** Provide robust features for organizing and managing exercise routines, incorporating advanced search options for a personalized fitness experience.
* **Technology Stack:** Harness contemporary web development technologies, with a focus on React.js, to ensure an efficient and enjoyable user experience.

**FEATURES OF FITFLEX**

* **Exercises from Fitness API:** Access a diverse array of exercises from reputable fitness APIs, covering a broad spectrum of workout categories and catering to various fitness goals.
* **Visual Exercise Exploration:** Engage with workout routines through curated image galleries, allowing users to explore different exercise categories and discover new fitness challenges visually.
* **Intuitive and User-Friendly Design:** Navigate the app seamlessly with a clean, modern interface designed for optimal user experience and clear exercise selection.
* **Advanced Search Feature:** Easily find specific exercises or workout plans through a powerful search feature, enhancing the app's usability for users with varied fitness preferences.

**TECHNICAL ARCHITECTURE**

FitFlex prioritizes a user-centric approach from the ground up. The engaging user interface (UI), built with React.js, keeps interaction smooth and intuitive. An API client, implemented using Axios, communicates with the backend, leveraging RapidAPI. This platform grants access to the ExerciseDB API, allowing FitFlex to integrate features like exercise names, types, instructions, and GIFs without building everything from scratch. This approach ensures a feature-rich experience while focusing development efforts on the core FitFlex functionalities.

**PRE-REQUISITES**

Here are the key prerequisites for developing a frontend application using React.js:

**NODE.JS AND NPM**

* **Purpose:** Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications.
* **Install:** Download from <https://nodejs.org/en/download/>.
* **Installation Instructions:** Follow the guide at <https://nodejs.org/en/download/package-manager/>.

**REACT.JS**

* **Purpose:** React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.
* **Setup:**
  + Create a new React app:

npx create-react-app fitflex

Replace fitflex with your preferred project name.

* + Navigate to the project directory:

cd fitflex

* + Start the development server:

npm start

* + Access the app at <http://localhost:3000> in your web browser.
* **HTML, CSS, and JavaScript:** Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.
* **Version Control:** Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.
  + **Git:** Download and installation instructions can be found at <https://git-scm.com/downloads>.
* **Development Environment:** Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.
  + **Visual Studio Code:** Download from <https://code.visualstudio.com/download>.
  + **Sublime Text:** Download from <https://www.sublimetext.com/download>.
  + **WebStorm:** Download from <https://www.jetbrains.com/webstorm/download>.

**TO GET THE APPLICATION PROJECT FROM DRIVE**

**Follow these steps:**

* **Get the Code:**
  + Download the code from the drive link (placeholder, to be provided): <https://drive.google.com/drive/folders/14f9eBO5WTVrd4PhP2W6PzOUHCV8UMex?usp=sharing>.
* **Install Dependencies:**
  + Navigate into the cloned repository directory and install libraries:

cd fitflex

npm install

* **Start the Development Server:**
  + To start the development server, execute the following command:

npm start

* **Access the App:**
  + Open your web browser and navigate to <http://localhost:3000>.
  + You should see the application's homepage, indicating that the installation and setup were successful.

You have successfully installed and set up the application on your local machine. You can now proceed with further customization, development, and testing as needed.

**PROJECT STRUCTURE**

In this project, the files are organized into key directories based on their purpose:

* **public/**:
  + index.html: Entry point with meta tags and Cloudflare script for security.
  + manifest.json: Configures PWA settings with icons and theme colors.
  + robots.txt: Allows all web crawlers to index the site.
* **src/**:
  + App.js: Main component handling exercise fetching and rendering.
  + App.css: Custom styles for the app’s layout and components.
  + index.js: Renders the app into the DOM.
  + index.css: Global styles with Poppins font settings.
  + reportWebVitals.js: Performance monitoring with web-vitals.
  + setupTests.js: Configures Jest for testing.
  + App.test.js: Unit test file (currently outdated, referencing "learn react").

**PROJECT FLOW**

PROJECT DEMO

* **Demo Link:** <https://drive.google.com/file/d/1Jlo8jbbzmw5Q3OMC8XYqQ51ptzuroUBJ/view?usp=drivesdk>

**MILESTONE 1: PROJECT SETUP AND CONFIGURATION**

* **Installation of Required Tools:**
  + To build the FitFlex app, we’ll need a developer’s toolkit. We’ll leverage React.js for the interactive interface, Axios to fetch fitness data, and custom CSS with the Poppins font for styling.
  + Open the project folder to install necessary tools. In this project, we use:
    - React.js
    - Axios
  + For further reference, use the following resources:
    - <https://react.dev/learn/installation>
    - <https://axios-http.com/docs/intro>

**MILESTONE 2: PROJECT DEVELOPMENT**

* **Setup the Routing Paths:** [Placeholder for future implementation with React Router Dom if needed.]
* **Develop the Navbar and Hero Components:** Create a header with the "FitFlex Train Hard" logo (with "Train Hard" in red) and a hero section with "Transform Your Body" and a background image.
* **Code the Popular Search/Categories Components:** Fetch exercises from the ExerciseDB API and implement real-time search functionality.
* **Develop the Category Page:** Display exercises under specific categories (e.g., by target muscle or equipment).
* **Code the Exercise Page:** Show detailed instructions, GIFs, and related videos (future enhancement with YouTube API).
* **Develop the Newsletter Subscription and Footer Components:** [Placeholder for future implementation.]

**IMPORTANT CODE SNIPPETS**

**FETCHING AVAILABLE EQUIPMENT LIST & BODY PARTS LIST**

From the RapidAPI hub, we fetch available equipment and list of body parts with an API request.

const bodyPartsOptions = {

method: 'GET',

url: 'https://exercisedb.p.rapidapi.com/exercises/bodyPartList',

headers: {

'X-RapidAPI-Key': 'your-api-key-here',

'X-RapidAPI-Host': 'exercisedb.p.rapidapi.com',

},

};

const equipmentOptions = {

method: 'GET',

url: 'https://exercisedb.p.rapidapi.com/exercises/equipmentList',

headers: {

'X-RapidAPI-Key': 'your-api-key-here',

'X-RapidAPI-Host': 'exercisedb.p.rapidapi.com',

},

};

useEffect(() => {

fetchData();

}, []);

const fetchData = async () => {

try {

const bodyPartsData = await axios.request(bodyPartsOptions);

setBodyParts(bodyPartsData.data);

const equipmentData = await axios.request(equipmentOptions);

setEquipment(equipmentData.data);

} catch (error) {

console.error(error);

}

};

* **Dependencies:** The code utilizes Axios (a popular promise-based HTTP client for JavaScript). Install it with npm install axios.
* **API Key:** Replace 'your-api-key-here' with your RapidAPI key. Obtain it from [RapidAPI](https://rapidapi.com/).
* **bodyPartsOptions and equipmentOptions:** These variables configure API requests to fetch body parts and equipment lists.
* **Breakdown:**
  + **method:** Set to 'GET' for retrieving data.
  + **url:** Specifies the API endpoints (bodyPartList and equipmentList).
  + **headers:** Includes the RapidAPI key and host for authentication.
  + **fetchData function:** Uses async/await to fetch and store data in state variables (setBodyParts, setEquipment).
  + **useEffect Hook:** Ensures data is fetched when the component mounts.

**FETCHING EXERCISES UNDER A PARTICULAR CATEGORY**

To fetch exercises under a particular category, use the following code:

const fetchData = async (id) => {

const options = {

method: 'GET',

url: `https://exercisedb.p.rapidapi.com/exercises/equipment/${id}`,

params: { limit: '50' },

headers: {

'X-RapidAPI-Key': 'your-api-key-here',

'X-RapidAPI-Host': 'exercisedb.p.rapidapi.com',

},

};

try {

const response = await axios.request(options);

console.log(response.data);

setExercises(response.data);

} catch (error) {

console.error(error);

}

};

* **Breakdown:**
  + **options:** Configures the API request with a dynamic URL using the id parameter and a limit of 50 results.
  + **fetchData function:** Fetches exercise data for a specific equipment ID and updates the exercises state.
  + **try...catch block:** Handles successful responses and logs errors.
* **Note:** Replace 'your-api-key-here' with your RapidAPI key. For security, store it in a .env file (e.g., REACT\_APP\_API\_KEY) and access it via process.env.REACT\_APP\_API\_KEY.

**FETCHING EXERCISE DETAILS**

Fetch details of a particular exercise using the exercise ID:

const fetchData = async (id) => {

const response = await fetch(`https://exercisedb.p.rapidapi.com/exercises/exercise/${id}`, {

method: 'GET',

headers: {

'X-RapidAPI-Key': 'your-api-key-here',

'X-RapidAPI-Host': 'exercisedb.p.rapidapi.com',

},

});

const data = await response.json();

setExerciseDetails(data);

console.log(data);

};

* **Breakdown:**
  + **fetch request:** Makes a GET request to the exercise endpoint with the id.
  + **Handling the Response:** Converts the response to JSON and updates the exerciseDetails state.
  + **Error Handling:** Logs errors to the console.

**FETCHING RELATED VIDEOS FROM YOUTUBE**

Fetch videos related to a particular exercise:

const fetchRelatedVideos = async (name) => {

const options = {

method: 'GET',

url: 'https://youtube-search-and-download.p.rapidapi.com/search',

params: {

query: `${name}`,

hl: 'en',

sort: 'relevance',

type: 'video',

},

headers: {

'X-RapidAPI-Key': 'your-api-key-here',

'X-RapidAPI-Host': 'youtube-search-and-download.p.rapidapi.com',

},

};

try {

const response = await axios.request(options);

setRelatedVideos(response.data);

} catch (error) {

console.error(error);

}

};

* **Breakdown:**
  + **options:** Configures the YouTube search API with a dynamic query (name) and parameters.
  + **fetchData function:** Fetches and stores video data in the relatedVideos state.

**PROJECT EXECUTION**

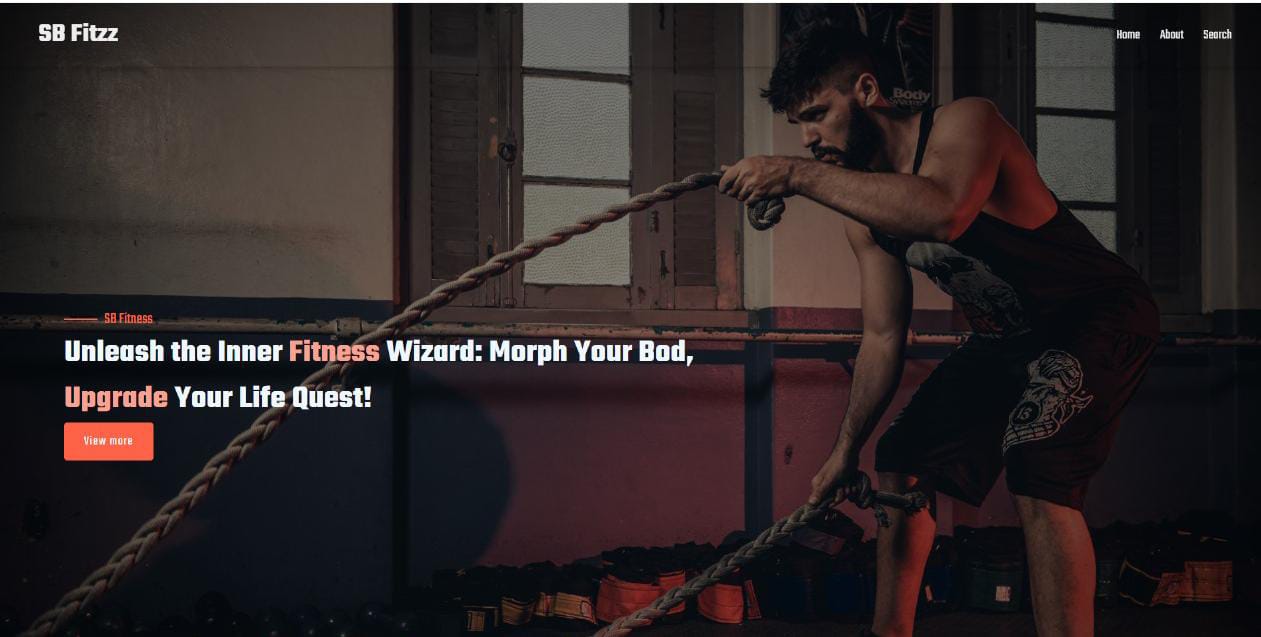
After completing the code, run the React application using:

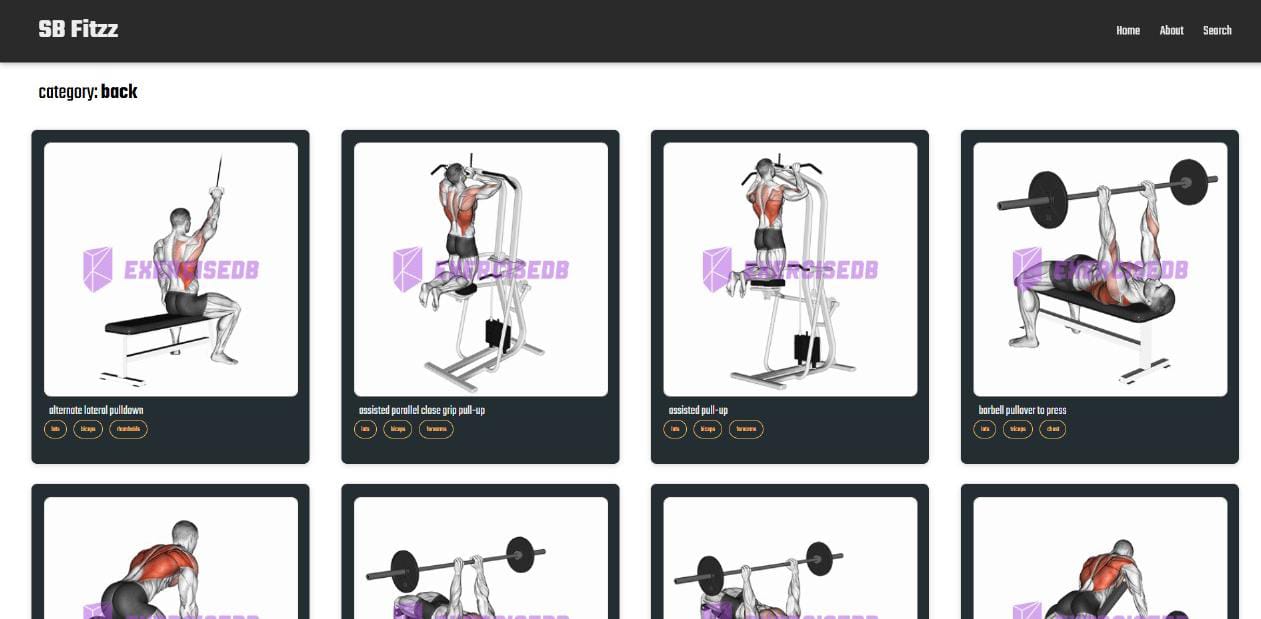
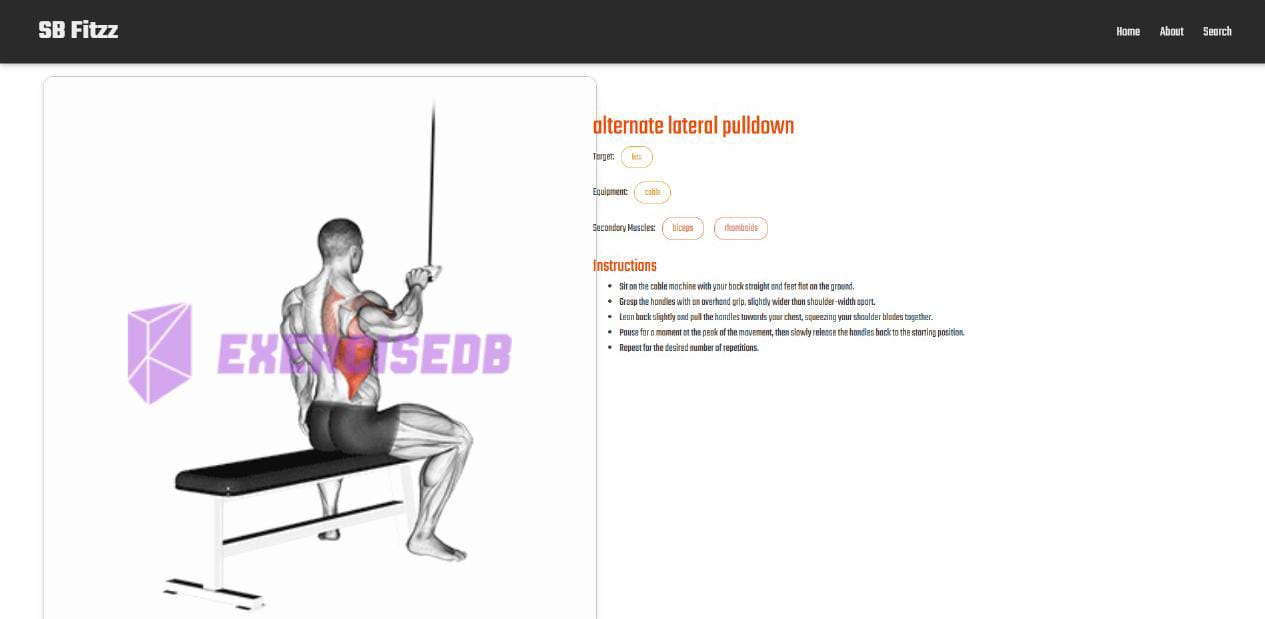
npm start

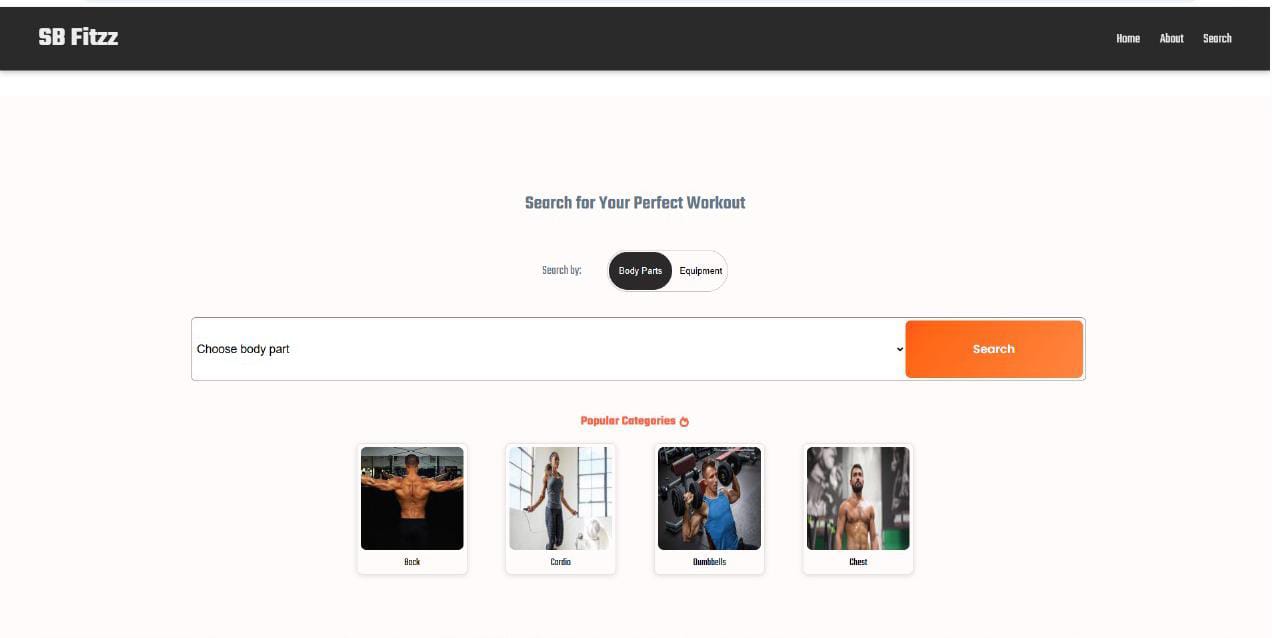
npm run dev

**SCREENSHOTS OF THE APPLICATION**

* **Hero Component:** Showcases trending workouts or fitness challenges to grab users' attention, featuring "Transform Your Body" with a background image.







**Future Enhancements:**

**ABOUT**

FitFlex isn’t just another fitness app. We’re meticulously designed to transform your workout experience, no matter your fitness background or goals.

**CONCLUSION**

FitFlex represents a significant step forward in personal fitness technology, blending a user-friendly interface with powerful API integration to deliver a transformative workout experience. By leveraging React.js and the ExerciseDB API, FitFlex empowers users of all levels to explore, customize, and enjoy their fitness journeys. As we look to the future, FitFlex is poised for growth with potential enhancements like advanced routing, community features, and multimedia integration, promising to remain a trusted companion in the pursuit of a healthier lifestyle. Thank you for joining us on this innovative fitness adventure!