# FitFlex:YourPersonal Fitness Companion(ReactApplication)

**Team ID: SWTID1741426304154548**


**Team Leader:**

**Praveen Raj .S**

Praveenrajpjofficial@gmail.com

**Team Members:**

**Preethi .R**

Preethiramu2204@gmail.com

**Vijaya Kumar .T**

Vijayk91359@gmail.com

**Akash .S**

Akashalash382@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. Effortlesslynavigatethroughawidearrayofexercisecategories with features like dynamic search, bringing you the latest andmosteffectiveworkoutsfrom the fitness world.

Fromthose embarking on their fitness journey toseasonedworkoutaficionados,FitFlex embraces a diverse audience,fosteringadynamiccommunityunitedbyasharedpassionfor a healthy lifestyle. Our vision is to reshape how users interact with fitness, presenting a platform that not onlyprovideseffectiveexerciseroutinesbutalsoencouragescollaboration and sharing within the vibrant fitness community.

Embarkon this fitness adventure with us, whereinnovationseamlesslyintertwineswith established exercise principles. Every tap within FitFlex propels you closer to a realm of diverse workouts and wellness perspectives. Join us and experience the evolutionoffitness engagement, where each feature ismeticulouslycraftedtoofferaglimpseintothefutureof a healthier you.

ElevateyourfitnessexplorationwithFitFlex,whereeveryexercisebecomesagateway to a world of wellness waiting to be discovered and embraced. Trust FitFlex to be your reliablecompanionontheJourneytostayingconnectedwithafitandactivelifestyle.

## ScenariobasedIntro:

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 yourworkouts.Withatap,youopentheapp.Vibrantvisualsfloodthescreen– 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.

## ProjectGoalsandObjectives:

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

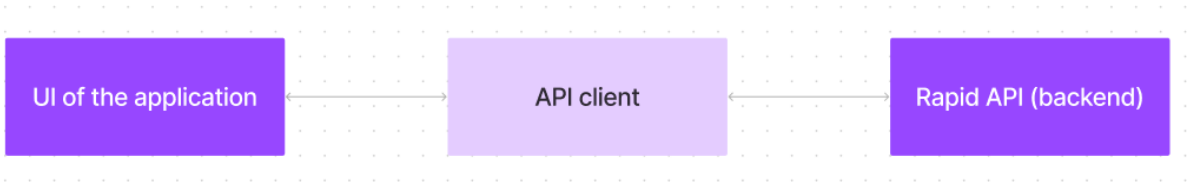
Ourkeyobjectivesareasfollows:

- ✔ **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.

## FeaturesofFitFlex:

- ✔**Exercises from Fitness API:** Access adiversearrayofexercisesfromreputablefitness APIs, coveringabroadspectrumofworkoutcategoriesandcateringtovariousfitness 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.

## TechnicalArchitecture:




FitFlex prioritizes a user-centric approach from the ground up. The engaging user interface (UI), likely built with a framework like React Native, keeps interaction smooth andintuitive. AnAPIclientspecificallydesignedforFitFlexcommunicateswiththebackend,butwitha

twist: it leverages Rapid API. This platform grants access to various external APIs, allowing FitFlex to potentially integrate features like fitness trackers, nutrition data, or workout tracking functionalities without building everything from scratch. This approach ensures a feature-rich experience while focusing development efforts on the core FitFlex functionalities.

## **PRE-REQUISITES**:

HerearethekeyprerequisitesfordevelopingafrontendapplicationusingReact.js:

✔**Node.js and npm**:

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 Node.js and npm on your development machine, as they are required to runJavaScript on the server-side.

- Download:https://nodejs.org/en/download/
- Installationinstructions:https://nodejs.org/en/download/package-manager/

✔**React.js**:

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.

InstallReact.js,aJavaScriptlibraryforbuildinguserinterfaces.

- CreateanewReact app:

```
npx create-react-app my-react-app
```

Replacemy-react-appwithyourpreferredprojectname.

- Navigatetotheprojectdirectory:

```
cd my-react-app
```

- RunningtheReactApp:

  With the React app created, you can now start the development server andsee your React application in action.

- Startthedevelopmentserver:

```
npm start
```

This command launches the development server, and you can access your React app at http://localhost:3000in 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 changesthroughoutthedevelopmentprocess.PlatformslikeGitHuborBitbucketcan host your repository.

- Git:Downloadandinstallationinstructionscanbefoundat: 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.

- VisualStudioCode:Downloadfromhttps://code.visualstudio.com/download
- SublimeText:Downloadfromhttps://www.sublimetext.com/download
- WebStorm:Downloadfromhttps://www.jetbrains.com/webstorm/download

TogettheApplicationprojectfromdrive:

Followbelowsteps:

✔**Get the code:**

- Downloadthecodefromthedrivelinkgivenbelow:

https://drive.google.com/drive/folders/14f9eBQ5W7VrLdPhP2W6PzOU_HCy8UMex?usp=sharing

**InstallDependencies:**

- NavigateintotheclonedrepositorydirectoryandinstalllibrariesIn:

```
cdfitness-app-react
npm install
```

✔**StarttheDevelopment Server**:

- Tostartthedevelopmentserver,executethefollowingcommand:

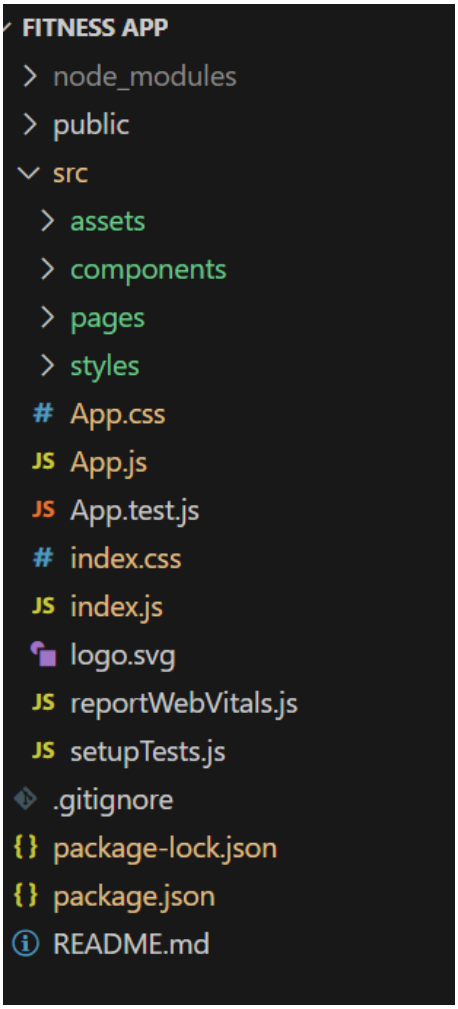
```
npm start
```

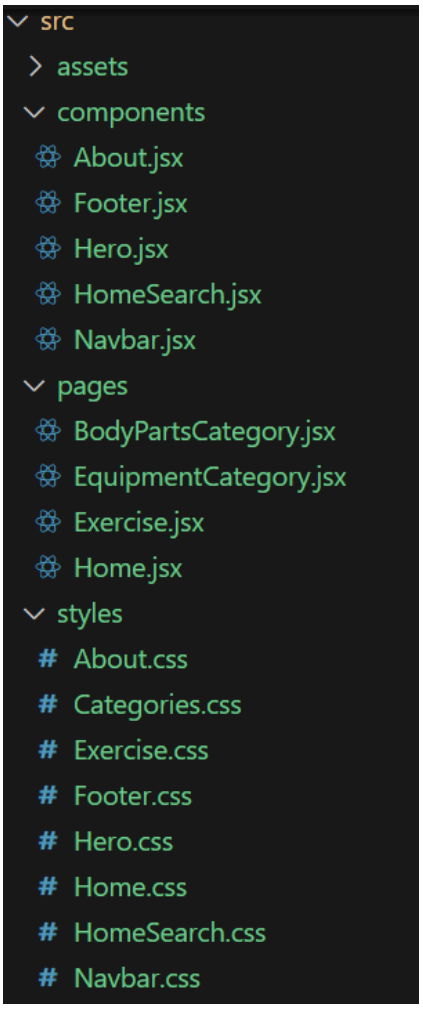
**Access the App:**

- Openyourwebbrowserandnavigatetohttp://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.

## Projectstructure:




In this project, we've split the files into 3 major folders, *Components, Pages and Styles.* Inthe pages folder, we store the files that acts as pages at different URLs in the application. The components folder stores all the files, that returns the small components in the application. All the styling css files will be stored in the styles folder.

## ProjectFlow:

### Projectdemo:

Beforestartingtoworkonthisproject,let'sseethedemo.

Demo link:https://drive.google.com/file/d/1mMqMb41RtroiFbUQ-1ZfeYfWJZ6okSNb/view?usp=sharing

Usethecode in:

https://drive.google.com/drive/folders/14f9eBQ5W7VrLdPhP2W6PzOU_HCy8UMex?usp=sharing

### Milestone1:Projectsetupandconfiguration.

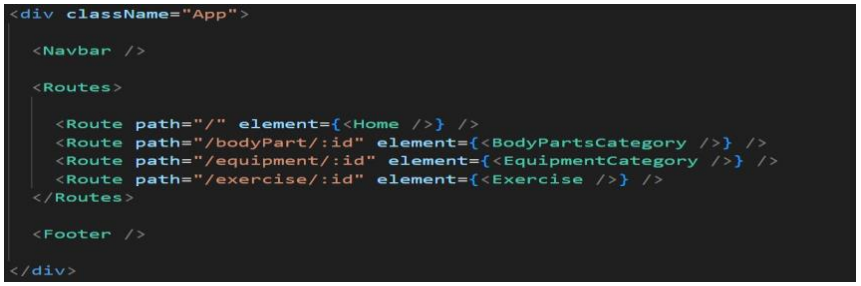
- **Installationofrequiredtools**:
  To build the FitFlex app, we'll need a developer's toolkit. We'll leverage React.js for the interactive interface, React Router Dom for seamless navigation, and Axios to fetch fitness data. To style the app, we'll choose either Bootstrap orTailwindCSSfor pre-built components and a sleek look.
  Opentheprojectfoldertoinstallnecessarytools.Inthisproject,weuse:
    - React Js
    - ReactRouterDom
    - ReactIcons
    - Bootstrap/tailwindcss
    - Axios

- Forfurtherreference,usethefollowingresources
    - https://react.dev/learn/installation
    - https://react-bootstrap-v4.netlify.app/getting-started/introduction/
    - https://axios-http.com/docs/intro
    - https://reactrouter.com/en/main/start/tutorial

### Milestone2:ProjectDevelopment

❖ SetuptheRoutingpaths

Setuptheclearroutingpathstoaccessvariousfilesintheapplication.




- DeveloptheNavbarandHero components
- Code the popular search/categories componentsandfetchthecategoriesfrom***rapid Api***.
- Additionally,wecanaddthecomponenttosubscribeforthenewsletterandthe footer.
- Now,developthecategorypagetodisplayvariousexercisesunderthecategory.
- Finally,codetheexercisepage,wheretheinstructions,otherdetailsalongwith related videos from the YouTube will be displayed.

**ImportantCodesnips:**

? **FetchingavailableEquipmentlist&Bodyparts list**
From the Rapid API hub,wefetchavailableequipmentandlistofbodypartswith an API request.

```
const bodyPartsOptions = {
  method: 'GET',
  url: 'https://exercisedb.p.rapidapi.com/exercises/bodyPartList',
  headers: {
    'X-RapidAPI-Key': 'place your api key',
    'X-RapidAPI-Host': 'exercisedb.p.rapidapi.com'
  }
};

const equipmentOptions = {
  method: 'GET',
  url: 'https://exercisedb.p.rapidapi.com/exercises/equipmentList',
  headers: {
    'X-RapidAPI-Key': 'place your api key',
    '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);
  }
}
```

Here'sabreakdownofthecode:

*Dependencies:*

Thecodeutilizesthefollowinglibraries:

Axios: A popular promise-based HTTP client for JavaScript. You canaddalink to the official documentation for Axios https://axios-http.com/

*API Key:*

Replace 'place your api key' with a placeholder mentioning that the user needs to replace it with their own RapidAPI key. You can mention how to acquire an API key from RapidAPI.

*bodyPartsOptionsandequipmentOptions:*

Thesevariablesholdconfigurationoptionsforfetchingdatafromthe RapidAPI exercise database.

- *method:* The HTTP method used in the request. Inthiscase,it'ssettoGETas the code is fetching data from the API.
- *url:* The URL of the API endpoint to fetch data from. Here, it's set to https://exercisedb.p.rapidapi.com/exercises/bodyPartListforfetchingalistof body parts and https://exercisedb.p.rapidapi.com/exercises/equipmentList for fetching a list of equipment.
- *headers:* This section contains headers required for making the API request. Here it includes the X-RapidAPI-Key header to provide your API key and theX-RapidAPI-Host header specifying the host of the API.

*fetchDatafunction:*

This function is responsible for fetching data from the API. It makes use of async/await syntax to handle asynchronous operations. First it fetches data forbody parts using axios.request(bodyPartsOptions). Then it stores the fetched data in the bodyParts state variable using setBodyParts.

Similarly, it fetchesdataforequipmentusingaxios.request(equipmentOptions).Then itstoresthefetcheddataintheequipmentstatevariableusingsetEquipment.Incase of any errors during the API request, the catch block logs the error to the console using console.error.

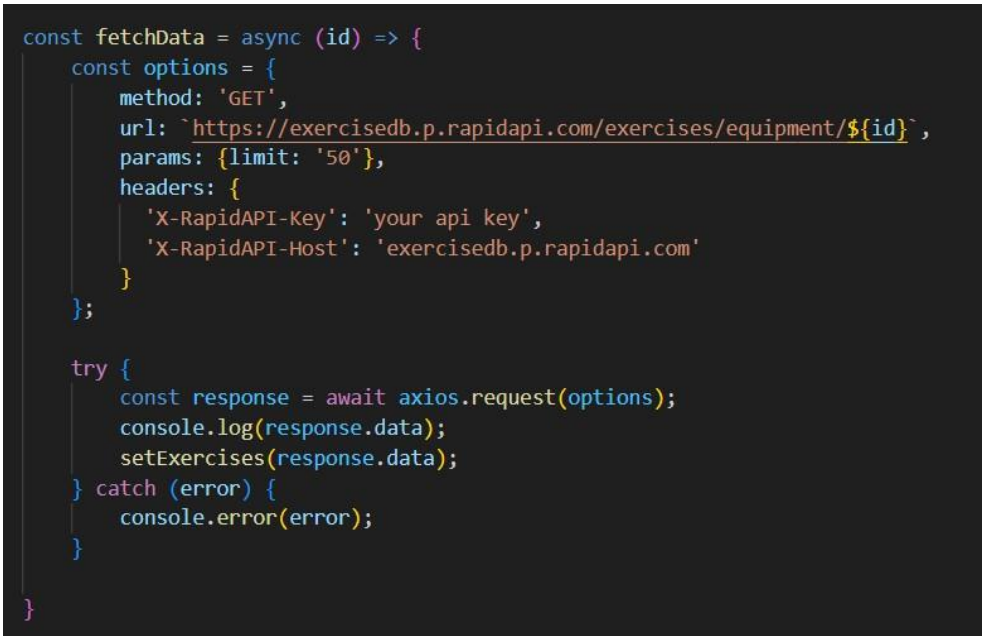
*useEffectHook:*

The useEffect hook is used to call the fetchData function whenever the component mounts. This ensures that the data is fetched as soon as the component loads.

Overall, the code snippet demonstrates how to fetch data from a RapidAPI exercise database using JavaScript's Axios library.

? **Fetchingexercisesunderparticularcategory**

Tofetchtheexercisesunderaparticularcategory,weusethebelowcode.



```
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',
          '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);
    }

}
```

It defines a function called fetchDatathatfetchesdatafromanexercisedatabase API. Here's a breakdown of the code:

*constoptions= {...}:*

This line creates a constant variable named options and assigns it an object literal. The object literal contains properties that configure the API request, including:

- method: Set to 'GET', indicating that the API request is a GET request to retrieve data from the server.
- url: Set to https://exercisedb.p.rapidapi.com/exercises/equipment/${id}, which is the URL of the API endpoint for fetching exercise equipment data. The ${id} placeholder will likely be replaced with a specific equipment ID when the function is called.
- params: An object literal with a property limit: '50'. This specifies that you want to retrieve a maximum of 50 exercise equipment results.
- headers: An object literal containingtwoheadersrequiredformakingtheAPI request:
- 'X-RapidAPI-Key': Your RapidAPI key, which is used for authentication. You should replace 'your api key' withaplaceholderinstructinguserstoreplaceit with their own API key.
- 'X-RapidAPI-Host': The host of theAPI,whichis'exercisedb.p.rapidapi.com'in this case.

*constfetchData=async(id)=>{...}:*

This line defines an asynchronous function named fetchData that takes an id parameter. This id parameter is likely used to specifytheequipmentIDforwhich data needs to be fetched from the API.

*try...catchblock:*

- Thetry...catchblockisusedtohandletheAPI request.
- The tryblockcontainsthecodethatattemptstofetchdatafromtheAPIusing axios.request(options).
- Theawaitkeywordisusedbeforeaxios.request(options)becausethefunction isasynchronousandwaitsfortheAPIrequesttocompletebeforeproceeding.
- IftheAPIrequestissuccessful,theresponsedataisstored in the response constant variable.
- Theconsole.log(response.data)linelogsthefetcheddatatotheconsole.
- The.thenmethod(notshownintheimage)islikelyusedtoprocessthe fetched data after a successful API request.
- The catch block handles anyerrorsthatmightoccurduringtheAPIrequest.If there's an error, it's logged to the console using console.error(error).

? **FetchingExercisedetails**

Now, with the help of the ExerciseID,wefetchthedetailsofaparticularexercise with API request.

```
useEffect(()=>{
    if (id){
        fetchData(id)
    }
},[])

const fetchData = async (id) => {
    const options = {
      method: 'GET',
      url: `https://exercisedb.p.rapidapi.com/exercises/exercise/${id}`,
      headers: {
        'X-RapidAPI-Key': 'ae40549393msh0c35372c617b281p103ddcjsn0f4a9ee43ff0',
        'X-RapidAPI-Host': 'exercisedb.p.rapidapi.com'
      }
    };

    try {
        const response = await axios.request(options);
        console.log(response.data);
        setExercise(response.data);

        fetchRelatedVideos(response.data.name)
    } catch (error) {
        console.error(error);
    }

}
```

The code snippet demonstrates how to fetch exercise data from an exercise database API using JavaScript's fetch API. Here's a breakdown of the code:

*APIEndpointandKey:*

- Replace'https://example.com/exercise'withtheactualURLoftheAPI endpoint you want to use.
- Replace'YOUR_API_KEY'withaplaceholderinstructinguserstoreplaceit with their own API key obtained from the API provider.

*asyncfunction:*

The code defines an asynchronous function named fetchData that likelytakesan id parameter as input. This id parameter might be used to specify the ID of a particular exercise or category of exercises to fetch.

*fetchrequest:*

Inside the fetchData function, the fetch APIisusedtomakeanHTTPGETrequest to the API endpoint. The function creates a fetch request with the following details:

- Method:GET(toretrievedatafromtheserver)
- URL:TheAPIendpointURLwhereexercisedataresides.

*Handling the Response:*

- The then method is used to handle the response from theAPIrequest.Ifthe request is successful (i.e., status code is 200), the response is converted to JSON format using response.json().
- The .then method then likely processes the fetched exercise data, which might involve storing it in a state variable or using it to populate a user interface.

*Error Handling:*

The .catch method is used to handle any errors that might occur during the API request. If there's an error, it's logged to the console using console.error.

**? FetchingrelatedvideosfromYouTube**

Now, with the API, we also fetch the videos related to a particular exercise with code given below.

```
const fetchRelatedVideos = async (name)=>{
  console.log(name)
  const options = {
    method: 'GET',
    url: 'https://youtube-search-and-download.p.rapidapi.com/search',
    params: {
      query: `${name}`,
      hl: 'en',
      upload_date: 't',
      duration: 'l',
      type: 'v',
      sort: 'r'
    },
    headers: {
      'X-RapidAPI-Key': 'ae40549393msh0c35372c617b281p103ddcjsn0f4a9ee43ff0',
      'X-RapidAPI-Host': 'youtube-search-and-download.p.rapidapi.com'
    }
  };

  try {
    const response = await axios.request(options);
    console.log(response.data.contents);
    setRelatedVideos(response.data.contents);
  } catch (error) {
    console.error(error);
  }
}
```

Thecodesnippet shows a function called *fetchRelatedVideos* that fetches data from YouTube using the RapidAPI service. Here's a breakdown of the code:

*fetchRelatedVideosfunction:*

Thisfunctiontakesanameparameterasinput,whichislikelythenameofa video or a search query.

*API configuration:*

Thecodecreatesaconstantvariablenamedoptionsandassignsitanobject literal containing configuration details for the API request:

- method: Set to 'GET', indicating a GET request to retrieve data from the server.
- url: Set to 'https://youtube-search-and-download.p.rapidapi.com/search', which is the base URL of the RapidAPI endpoint for YouTube search.
- params: An object literal containing parameters for the YouTube search query:
- query: Set to \${name}, a template literal that likely gets replaced with the actual name argument passed to the function at runtime. This specifies the search query for YouTube videos.
- Other parameters like hl (language), sort (sorting criteria), and type (video type) are included but their values are not shown in the snippet.

- headers: An object literal containing headers required for making the API request:
- 'X-RapidAPI-Key': Your RapidAPI key, which is used for authentication. You should replace 'YOUR_API_KEY' with a placeholder instructing users to replace it with their own API key.
- 'X-RapidAPI-Host':ThehostoftheAPI,whichis 'youtube-search-and-download.p.rapidapi.com' in this case.

*FetchingData(try...catchblock):*

- Thetry...catchblockisusedtohandletheAPI request.
- The tryblockcontainsthecodethatattemptstofetchdatafromtheAPIusing axios.request(options).
- axios is an external JavaScript library for making HTTP requests. If you don't already use Axios in your project, you'll need to install it using a package manager like npm or yarn.
- The .then method (not shown in the code snippet) is likely used to process the fetched data after a successful API request.
- The catch block handles anyerrorsthatmightoccurduringtheAPIrequest.If there's an error, it's logged to the console using console.error(error).

## ProjectExecution:

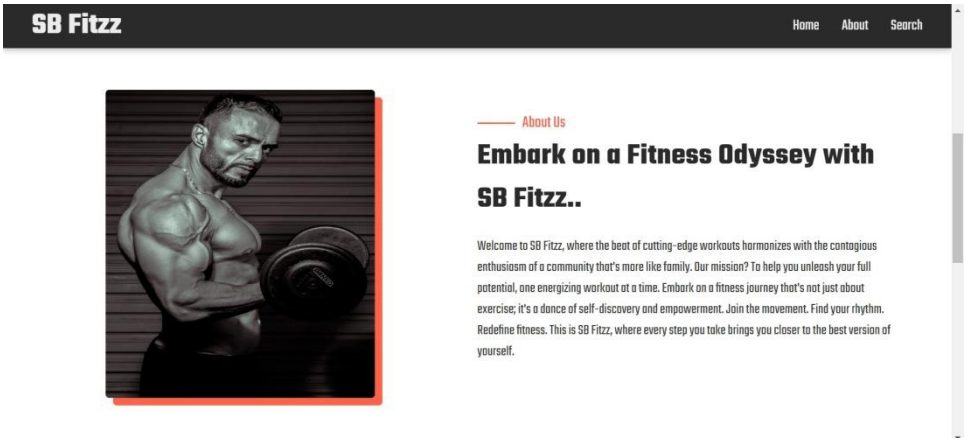
Aftercompletingthecode,runthereactapplicationbyusingthecommand“npm start” or “npm run dev” if you are using vite.js

Herearesomeofthescreenshotsoftheapplication.

? **Herocomponent**
thissectionwouldshowcasetrendingworkoutsorfitnesschallengestograb users' attention.

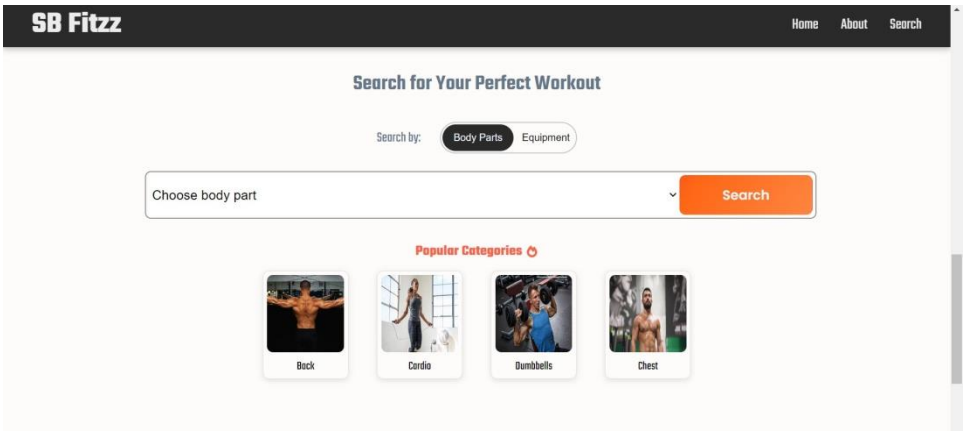
## ?About

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



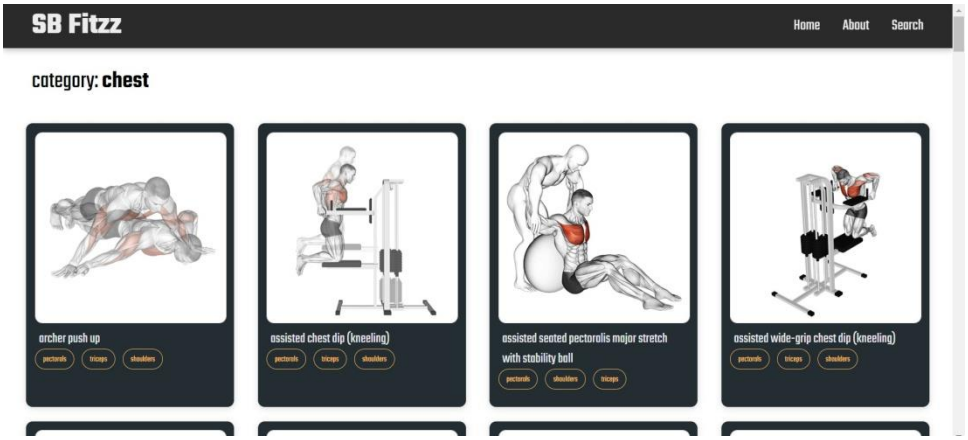

## ?Search

B Fitzz makes finding your perfect workout effortless. Our prominent search bar empowers you to explore exercises by keyword, targeted muscle group, fitness level, equipment needs, or any other relevant criteria you have in mind.Simply type in your search term and let FitFlex guide you to the ideal workout for your goals.



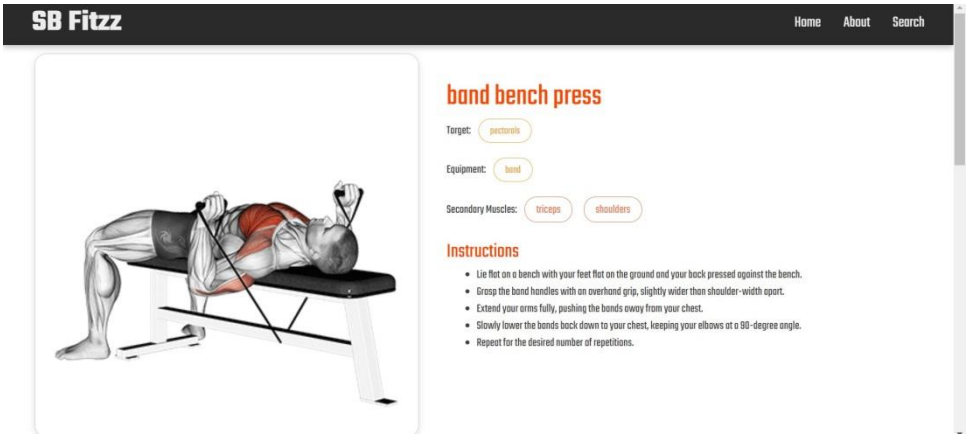

## ?Categorypage

FitFlex would offer a dedicated section for browsing various workout categories. This could be a grid layout with tiles showcasing different exercise types (e.g., cardio, strength training, yoga) with icons or short descriptions for easy identification.




**?Exercise page**

This is where the magic happens!Each exercise page on FitFlex provides a comprehensive overview of the chosen workout.Expect clear and concise instructions, accompanied by high-quality visuals like photos or videosdemonstrating proper form.Additional detailsliketargetedmusclegroups,difficulty level, and equipment requirements (if any) will ensure you have all the information needed for a safe and effective workout.




Demolink:https://drive.google.com/file/d/1mMqMb41RtroiFbUQ-1ZfeYfWJZ6okSNb/view?usp=sharing

***** Happy coding!! *****