****

**PROJECT REPORT**

**FitFlex: Your Personal Fitness Companion**

|  |  |
| --- | --- |
| YEAR  **:** | **2024 – 2025** |
| COLLEGE NAME  **:** | **K.C.S KASI NADAR COLLEGE OF ARTS & SCIENCE** |
| CODE  **:** | UNM203 |
| DEPARTMENT  **:** | **COMPUTER SCIENCE** |
| PROGRAM  **:** | **B.C.A** |
| SEMESTER  **:** | **VI** |
| PROJECT SUBMITTED TO**:** | UNIVERSITY OF MADRAS / NAAN MUDALVAN |
| Course Name **:** | **Front End Development and Database Administration** |

**TEAM LEADER: AJAY C EMAIL:** rcajay95@gmail.com

**MEMBERS:**

|  |  |  |
| --- | --- | --- |
| 1. | KISHANTH A | **EMAIL:** kishanth7209@gmail.com |
| 2. | IMRAN ULLAH R | **EMAIL:** imranik200505@gmail.com |
| 3. | KARTHIKEYAN P | **EMAIL:** **k**k9734969@gmail.com |
| 4. | PRABAKAR K | **EMAIL:** prabakark779@gmail.com |

**GUIDED BY: MRS.R.PADMADEVI**

**SPOC NAME: Dr.K. LALITHAKAMESWARI**

**FitFlex: Your Personal Fitness Companion**

**(React Application)**

**Introduction:**

🏋️‍♂️ Welcome to the forefront of fitness exploration with SB Fitzz! 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, SB Fitzz is set to redefine the entire fitness discovery and exercise experience.

💪 Crafted with a commitment to user-friendly aesthetics, SB Fitzz 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. Whether you're looking for high-intensity interval training, strength-based workouts, flexibility routines, or guided meditation exercises, SB Fitzz ensures that you have access to the most relevant and up-to-date fitness content at your fingertips.

🌐 From those embarking on their fitness journey to seasoned workout aficionados, SB Fitzz embraces a diverse audience, fostering a dynamic community united by a shared passion for a healthy lifestyle. The app serves as a bridge between knowledge and action, making it easier than ever to find exercises that match your goals, preferences, and available equipment. Users can interact with fellow fitness enthusiasts, share their progress, and exchange tips and tricks to stay motivated on their journey.

🔥 Embark on this fitness adventure with us, where innovation seamlessly intertwines with established exercise principles. Every tap within SB Fitzz propels you closer to a realm of diverse workouts and wellness perspectives. The app is designed to be an all-in-one fitness solution, incorporating cutting-edge technology to provide a seamless and engaging experience. With personalized recommendations, progress tracking, and the ability to customize workouts based on individual needs, SB Fitzz ensures that fitness remains accessible, enjoyable, and sustainable for everyone.

💥 Experience a new level of fitness engagement with intelligent workout suggestions tailored to your body’s needs. Whether you prefer bodyweight exercises, gym-based routines, or at-home workouts, SB Fitzz adapts to your environment and goals. Stay on track with integrated goal-setting features, reminders, and performance insights that help you measure your progress effectively.

⚡ Push your limits with a growing library of exercises sourced from expert trainers and fitness professionals. Explore visual workout guides, detailed descriptions, and instructional videos that ensure proper form and technique. With SB Fitzz, every workout is an opportunity to challenge yourself, learn something new, and move one step closer to your fitness goals.

🏋️‍♀️ Elevate your fitness exploration with SB Fitzz, where every exercise becomes a gateway to a world of wellness waiting to be discovered and embraced. Trust SB Fitzz to be your reliable companion on the journey to staying connected with a fit and active lifestyle. No matter your experience level, fitness preference, or training style, SB Fitzz is here to support you every step of the way. Join the movement today and redefine what’s possible in your fitness journey! 🚀

**Project Goals and Objectives:**

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

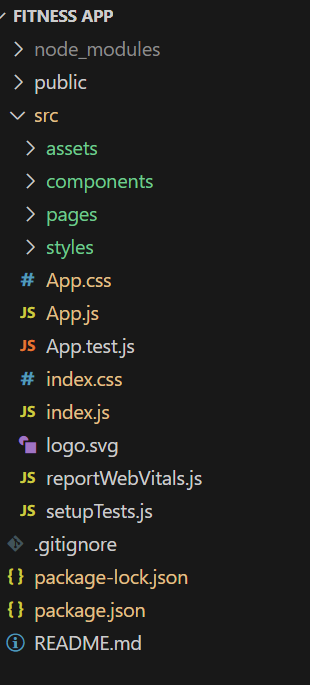
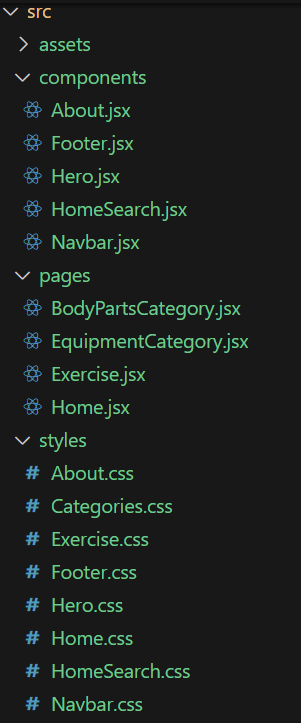
Our key objectives are as follows:

* **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 SB Recipess:**

* **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.

**Project structure:**

In this project, we’ve split the files into 3 major folders, *Components, Pages and Styles.* In the 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.

*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.

*bodyPartsOptions and equipmentOptions:*

These variables hold configuration options for fetching data from the RapidAPI exercise database.

* *method:* The HTTP method used in the request. In this case, it's set to GET as 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/bodyPartList for fetching a list of 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 the X-RapidAPI-Host header specifying the host of the API.

*fetchData function:*

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 for body parts using axios.request(bodyPartsOptions). Then it stores the fetched data in the bodyParts state variable using setBodyParts.

Similarly, it fetches data for equipment using axios.request(equipmentOptions). Then it stores the fetched data in the equipment state variable using setEquipment. In case of any errors during the API request, the catch block logs the error to the console using console.error.

*useEffect Hook:*

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.

* **Fetching exercises under particular category**

To fetch the exercises under a particular category, we use the below code.

****

* **Fetching Exercise details**

Now, with the help of the Exercise ID, we fetch the details of a particular exercise with API request

****

* **Fetching related videos from YouTube**

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

****

*fetchRelatedVideos function:*

This function takes a name parameter as input, which is likely the name of a video or a search query.

*API configuration:*

The code creates a constant variable named options and assigns it an object 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': The host of the API, which is 'youtube-search-and-download.p.rapidapi.com' in this case.

*Fetching Data (try...catch block):*

* The try...catch block is used to handle the API request.
* The try block contains the code that attempts to fetch data from the API using 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 any errors that might occur during the API request. If there's an error, it's logged to the console using console.error(error).

**Project Execution:**

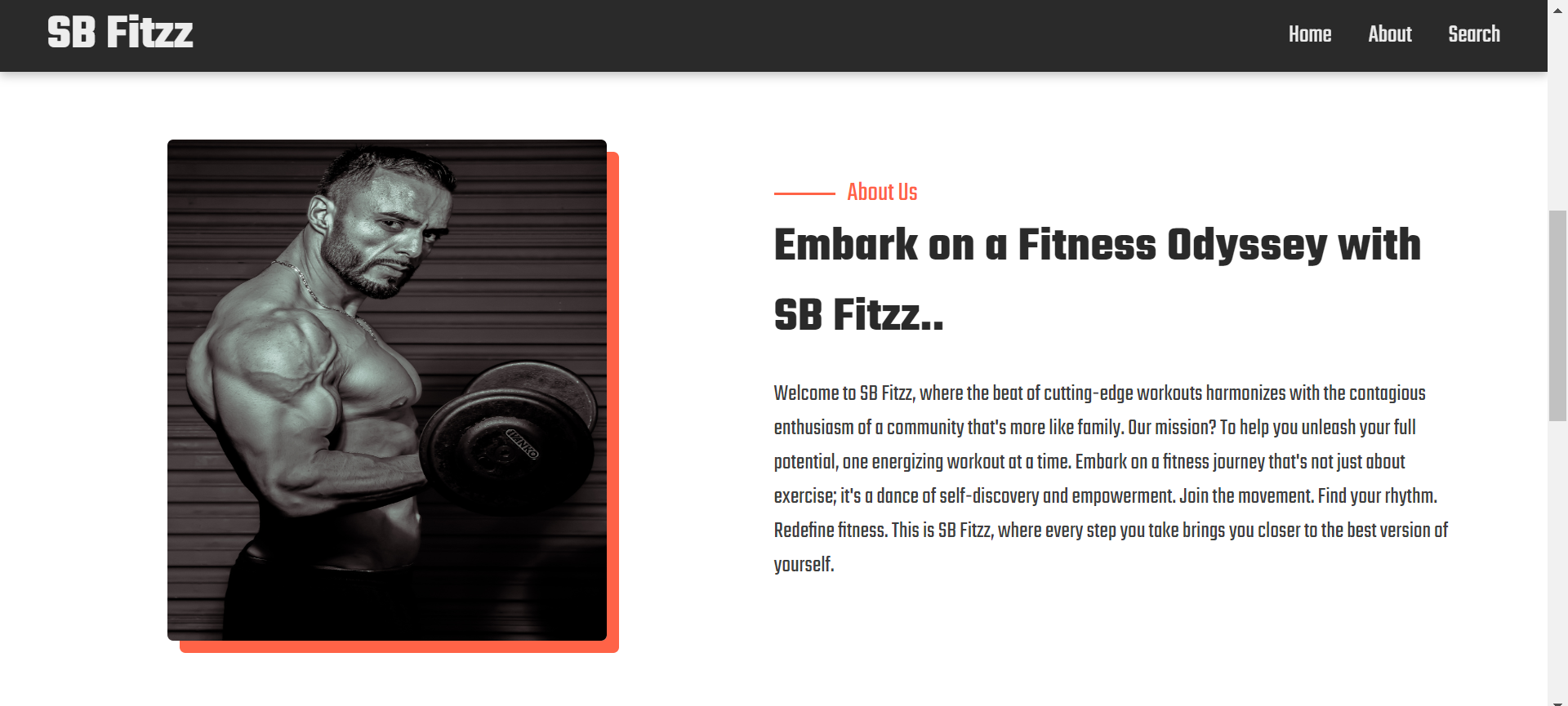
After completing the code, run the react application by using the command “npm start” or “npm run dev” if you are using vite.js Here are some of the screenshots of the application.

**User Interface snips:**

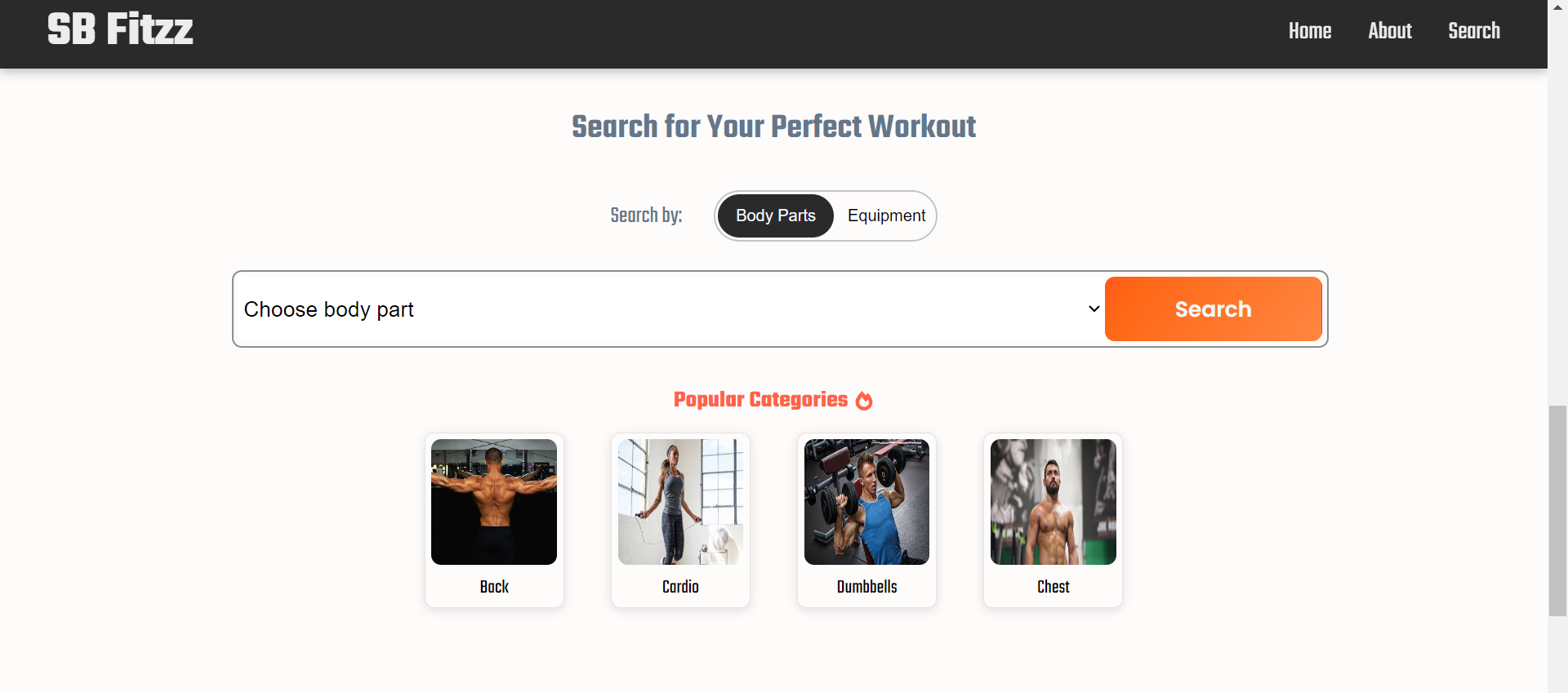
* Hero component

****

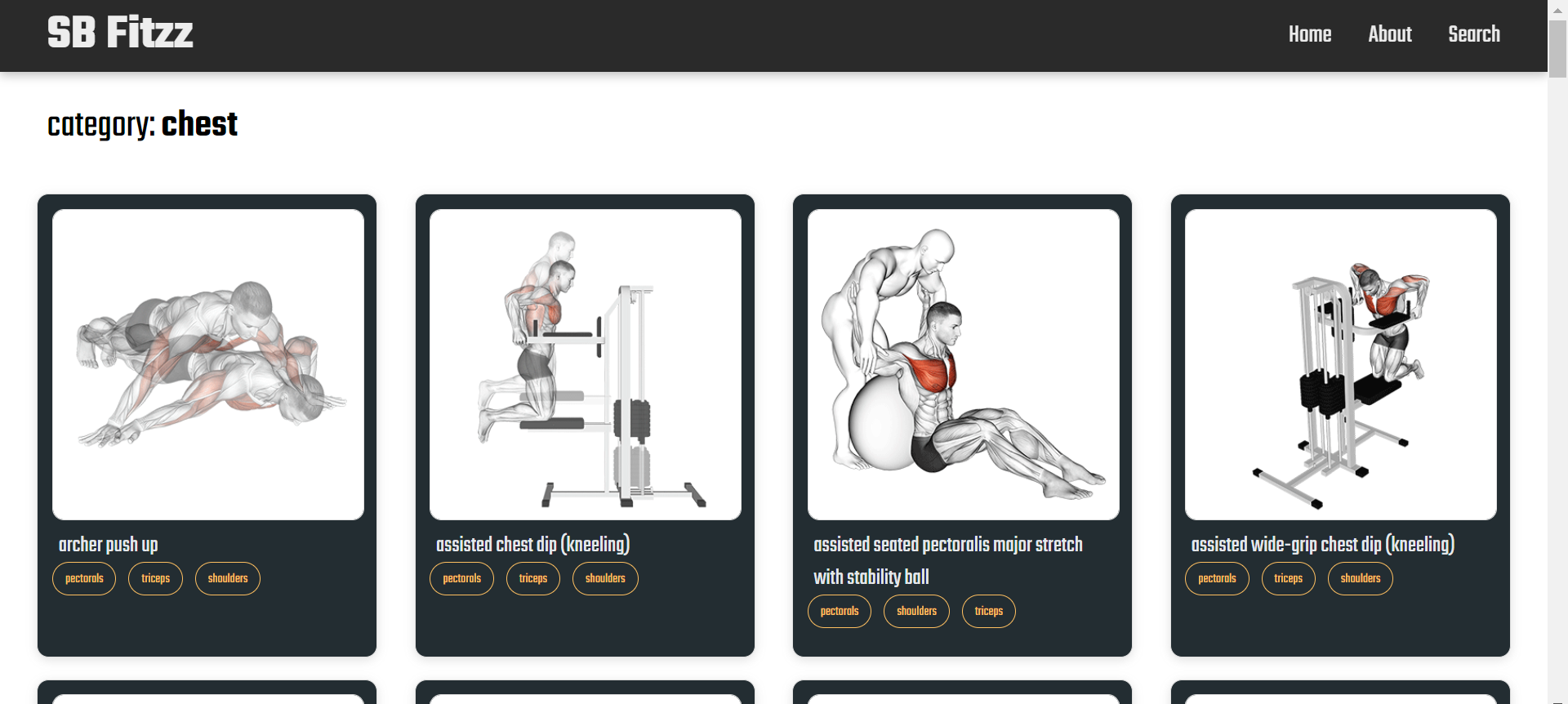
* About

****

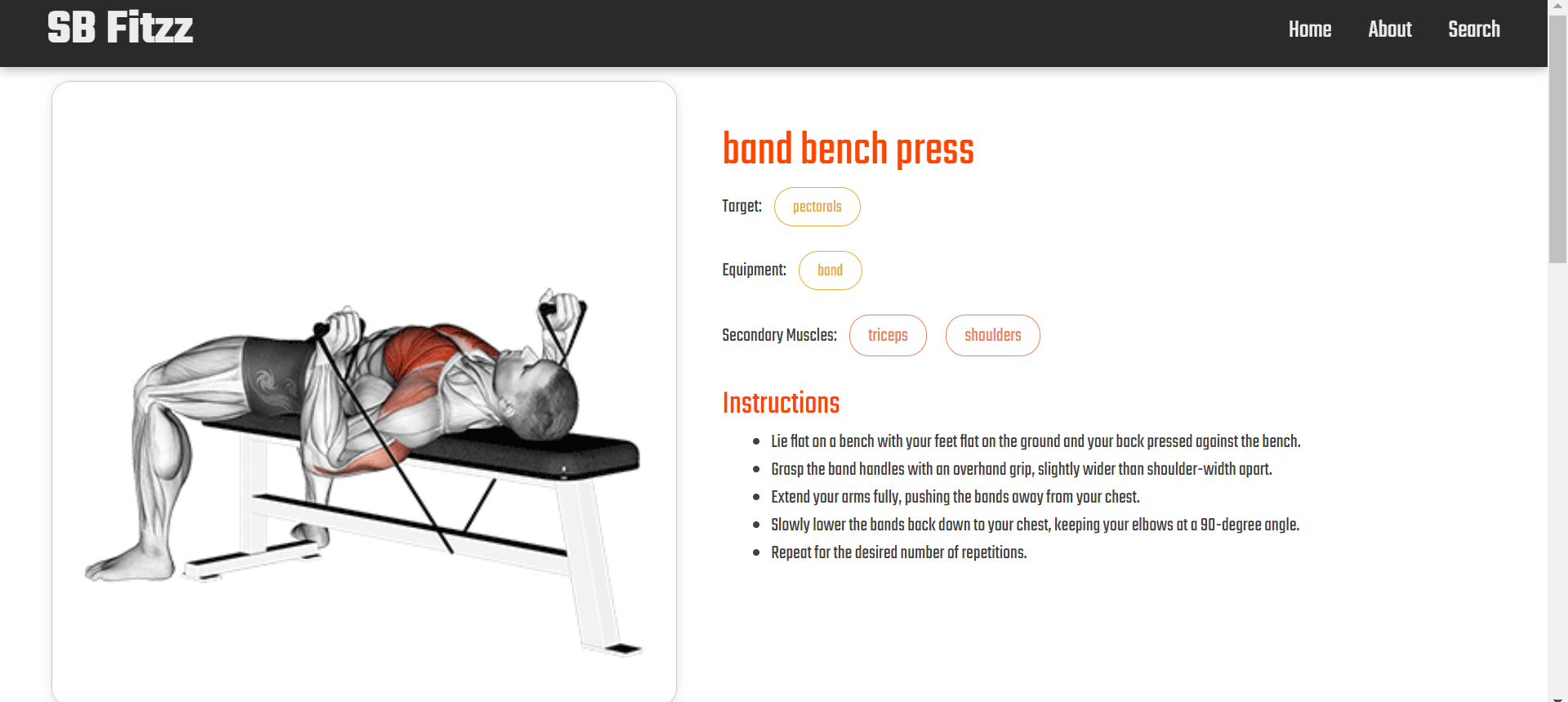
* Search

****

* Category page

****

* Exercise page



**Project Demo link:**

https://drive.google.com/drive/folders/1cTIeMa\_j4CufC1Ir5aTe5hkV2mS3XNk0