PROJECT DOCUMENTATION

Fitflex: Your Personal Fitness Companion

1.INTRODUCTION

\* PROJECT TITLE : Fitflex: your personal fitness companion

\* Team ID : NM2025TMID33793

\* Team Leader : NOORUL RAYAN N

& [noorulrayan03@gmail.com](mailto:noorulrayan03@gmail.com)

\* Team Members :

\_ ZULAIGA NAZREEN M

& yusufyusuf678moha@gmail.com

\_ASRA M

& asraaara035@gmail.com

\_AZREENA J

& azreenajafar@gmail.com

### Project Objective

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

### PRE-REQUISITES

Here are the key prerequisites for developing a frontend application using React.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 run JavaScript on the server-side.

* Download: <https://nodejs.org/en/download/>
* Installation instructions: <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.

Install React.js, a JavaScript library for building user interfaces.

* Create a new React app:

npx create-react-app my-react-app

Replace my-react-app with your preferred project name.

* Navigate to the project directory:

cd my-react-app

* Running the React App:

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

* Start the development server:

npm start

This command launches the development server, and you can access your React app at [http://localhost:3000](about:blank) 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](https://www.jetbrains.com/webstorm/download%20)

To get the Application project from drive:

Follow below steps:

* **Get the code:**

• Download the code from the drive link given below:

<https://drive.google.com/file/d/1kBr3-Ze8EMYaqyMmFRa3JLK_zH75deHs/view?usp=drivesdk>

**Install Dependencies:**

• Navigate into the cloned repository directory and install libraries:

cd fitness-app-react

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](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, 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.

PROJECT FLOW

Project demo:

Before starting to work on this project, let’s see the demo.

Demo link:

https://drive.google.com/file/d/18fAM5Ei7SU5FGt2zjeG3Xu20ajY06w0C/view?usp=drivesdk

PROJECT SETUP AND CONFIGURATION:

Installation of required tools:

Open the project folder to install necessary tools

In this project, we use:

React Js

React Router Dom

React Icons

Bootstrap/tailwind css

Axios

For further reference, use the following resources

https://react.dev/learn/installation

https://react-bootstrap-v4.netlify.app/getting-started/introduction/

https://axios-http.com/docs/intro

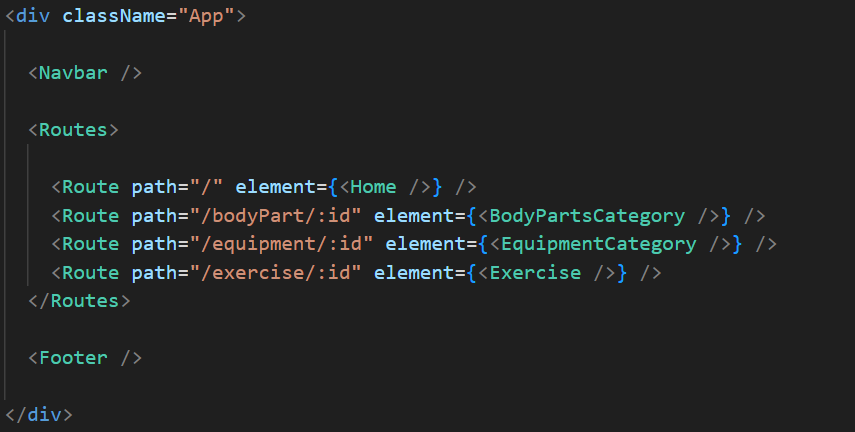
https://reactrouter.com/en/main/start/tutorial

PROJECT DEVELOPMENT:

### Project Development

* Setup the Routing paths

Setup the clear routing paths to access various files in the application.



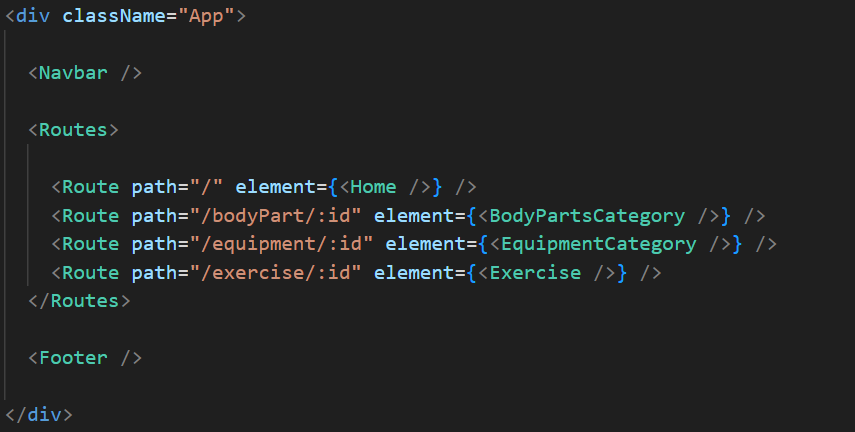
* Develop the Navbar and Hero components
* Code the popular search/categories components and fetch the categories from ***rapid Api***.
* Additionally, we can add the component to subscribe for the newsletter and the footer.
* Now, develop the category page to display various exercises under the category.
* Finally, code the exercise page, where the instructions, other details along with related videos from the YouTube will be displayed.
* [**Guided Project**](https://naanmudhalvan.smartinternz.com/Student/guided_project_info/24220#menu1)
* [**Project Workspace**](https://naanmudhalvan.smartinternz.com/Student/guided_project_workspace/24220)



### Project Development

* Setup the Routing paths

Setup the clear routing paths to access various files in the application.



* Develop the Navbar and Hero components
* Code the popular search/categories components and fetch the categories from ***rapid Api***.
* Additionally, we can add the component to subscribe for the newsletter and the footer.
* Now, develop the category page to display various exercises under the category.
* Finally, code the exercise page, where the instructions, other details along with related videos from the YouTube will be displayed.

**Important Code snips:**

* **Fetching available Equipment list & Body parts list**

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

****

Here's a breakdown of the code:

*Dependencies:*

The code utilizes the following libraries:

Axios: A popular promise-based HTTP client for JavaScript. You can add a link 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.

*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.

**Important Code snips:**

* **Fetching available Equipment list & Body parts list**

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

Important Code snips:

• Fetching Top/Trending news

Develop the Navbar and Hero components

Code the popular categories components and fetch the categories from newsapi.

Also, add the trending news in the home page.

Additionally, we can add the component to subscribe for the newsletter and the footer.

Now, develop the category page to display various news articles under the different categories.