|  |  |
| --- | --- |
| GOVERNMENT ARTS AND SCIENCE COLLEGE  19, POONTHOTTAM STREET  THIRUVOTTIYUR ,CHENNAI-19   |  | | --- | | FitFlex : Your Personal Fitness Companion |   TEAM ID:NM2025TMID37751  TEAM SIZE: 4  Naan-Mudhalvan/Brand Logo.pdf at main · Dhanapals03/Naan-Mudhalvan · GitHub  TEAM LEADER: DINESH J  TEAM MEMBER: DIWAKARAN P  TEAM MEMBER: DINESH S  TEAM MEMBER: GOKULA KRISHNAN V  2025-26 |

**INDEX**

* **Introduction**
* **Project goal**
* **Objectives**
* **Features of SB recipes**
* **Pre-Requisites**
* **Project Structure**
* **Project flow**

**FitFlex : Your Personal Fitness Companion**

**(React Application)**

****

**Introduction:**

In today's fast-paced world, maintaining a healthy lifestyle can be challenging due to time constraints, lack of motivation, and limited access to personalized fitness guidance. **FitFlex: Your Personal Fitness Companion** is a comprehensive solution designed to address these challenges by offering a smart, user-friendly platform that supports individuals on their fitness journey.

**Project Goal:**

* The primary goal of **FitFlex** is to develop a personalized fitness companion application that motivates users to lead a healthier lifestyle by providing customized workout routines, nutrition guidance, and real-time fitness tracking in an intuitive and accessible platform.

**Objectives:**

1. **Personalized Fitness Plans:**  
   To create customized workout routines based on user preferences, fitness level, and goals (e.g., weight loss, muscle gain, general fitness).
2. **Diet and Nutrition Tracking:**  
   To provide meal suggestions and allow users to track their daily caloric intake and nutritional information for balanced health management.
3. **Real-Time Progress Monitoring:**  
   To implement features like step counting, calorie tracking, BMI calculation, and activity logs to monitor user performance over time.
4. **Goal Setting and Motivation:**  
   To enable users to set short-term and long-term fitness goals, and stay motivated through reminders, streaks, and progress badges.
5. **User-Friendly Interface:**  
   To design an intuitive and interactive UI/UX that is accessible to users of all ages and technical backgrounds.
6. **Data Security and Privacy:**  
   To ensure that all personal and health-related data are securely stored and handled in compliance with privacy standards.
7. **Scalability and Integration:**  
   To build a system that can integrate with wearable devices (e.g., smartwatches, fitness bands) and scale with additional features like social sharing, AI-based suggestions, or community challenges.

**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.

**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/>
* 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.

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](http://localhost:3000%20) 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/drive/folders/14f9eBQ5W7VrLdPhP2W6PzOU_HCy8UMex?usp=sharing>

* **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

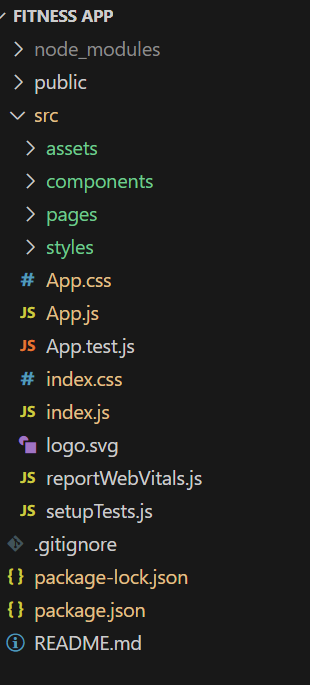
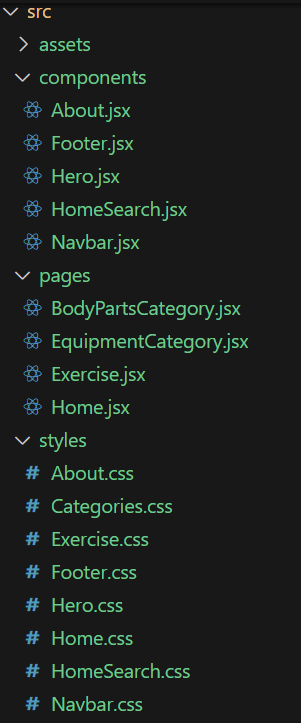
* **Access the App:**

• 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/1dVVEwbZgAltQyv8yXszbQkw98dhnOb9V/view?usp=sharing>

Use the code in:

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

**Milestone 1: Project setup and configuration.**

* **Installation of required tools**:

1. 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>

**Milestone 2: Project Development**

* Setup the Routing paths:

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

Ex:



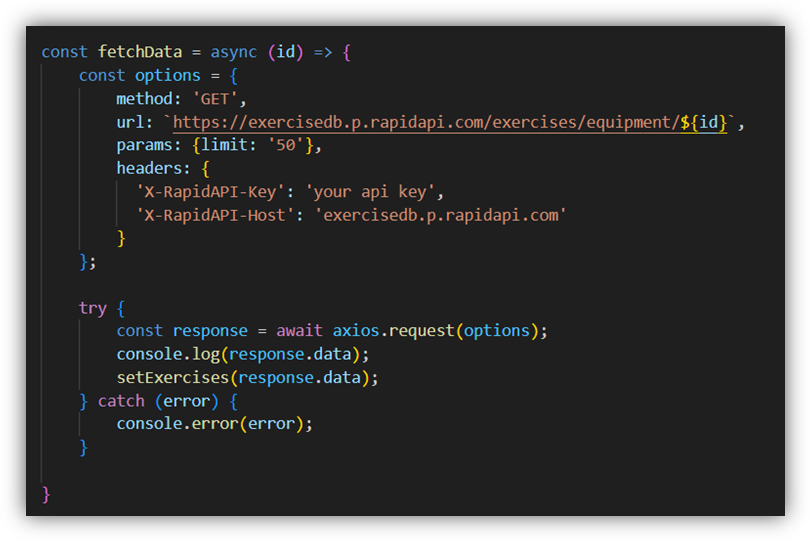
* Develop the Navbar and Hero components
* Code the popular search/categories components and fetch the categories from ***rapid Api***.
* Additionally, we can add the component to subscribe for the newsletter and the footer.
* Now, develop the category page to display various exercises under the category.
* 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**

****

* **Fetching exercises under particular category**

****

* **Fetching Exercise details**

****

* **Fetching related videos from YouTube**

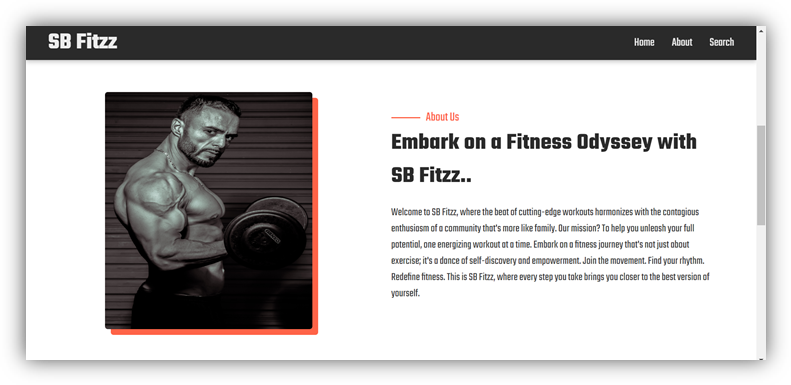
****

**User Interface snips:**

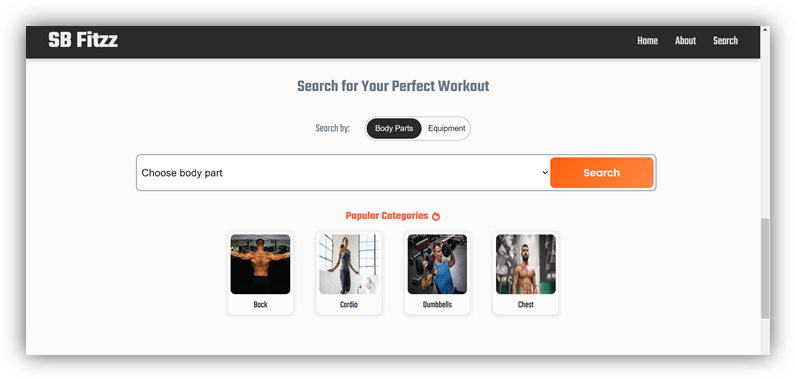
* Hero component

****

* About

****

* Search

****

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