**Naan Mudhalvan-Frontend Development and Database Administration**

**VALLIAMMAL COLLEGE FOR WOMEN**

**(College Code: 1363)**

**Department of Computer Applications**

**Project Title: FitFlex: Your Personal Fitness Companion**

**(React Application)**

**Team ID : 149637**

**Team Leader : TAMILARASI . V**

**Team Size : 4**

**Team Members :1.TAMILARASI . V**

**2.MONIGA . S**

**3.SUREKA . M**

**4.MAGIBA . A**

**GitHub Link (includes coding and documentation):**

**https://github.com/Magiebaabi/Fitflex\_Tamilarasi\_NM.git**

**Google Drive Link(includes project demo video) : https://drive.google.com/drive/folders/1Q1Mt05txjmSJzCXrYEBKYx9hQuLcElf5**

**Frontend development with React.js**

**Project Documentation Format**

1. **Introduction**

* **Project Title :** FitFlex: Your Personal Fitness Companion (React Application)
* **Team Members :1.TAMILARASI . V**(Role: Leading the team and assigning task)

**2.MONIGA . S**(Role: Project video demo-making and assistance)

**3.SUREKA . M**(Role: Documentation and assistance)

**4.MAGIBA . A**(Role: Coding and assistance)

1. **Project Overview**
   * **Purpose**: Briefly describe the purpose and goals of the project.
   * **Features**: Highlight the key features and functionalities of the frontend.

FitFlex

: **Your Personal Fitness**

**Companion**

**(React Application)**

**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 aesthetic.

🌐

**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 – personalizedworkout plans, diverse exercise categories, and a supportive community.

**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.

**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.

**Technical Architecture:**

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 and intuitive.

An API client specifically designed for FitFlex communicates with the backend, but with atwist: it leverages Rapid API.

**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.

● Download: https://nodejs.org/en/download/

● Installation instructions: https://nodejs.org/en/download/package-manager/

✔ **React.js**:

● 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 startThis command launches the development server, and you can access

your React 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 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.

• You should see the application's homepage, indicating that the installation

and setup were successful.

**Project structure:**

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

**Project demo:**

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

Demo

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

aring

Use the code in:

https://drive.google.com/drive/folders/14f9eBQ5W7VrLdPhP2W6PzOU\_HCy8UMex?usp=sharing

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

● **Installation of required tools**:

o React Icons

o Bootstrap/tailwind css

o Axios

● For further reference, use the following resources

o https://react.dev/learn/installation

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

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

o 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.❖ Develop the Navbar and Hero components

**Important Code snips:**

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.

Overall, the code snippet demonstrates how to fetch data from a RapidAPI exercise

database using JavaScript's Axios library.

To fetch the exercises under a particular category, we use the below code.It defines a function called fetchData that fetches data from an exercise database

API. Here's a breakdown of the code:

*const options = {...}:*

● 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 containing two 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 'exercisedb.p.rapidapi.com' in

this case.

*const fetchData = async (id) => {...}:*This line defines an asynchronous function named fetchData that takes an id

parameter. This id parameter is likely used to specify the equipment ID for which

data needs to be fetched from the API.

*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).

● The await keyword is used before axios.request(options) because the function

is asynchronous and waits for the API request to complete before proceeding.

● If the API request is successful, the response data is stored in the response

constant variable.

● The console.log(response.data) line logs the fetched data to the console.

*API Endpoint and Key:*

● Replace 'https://example.com/exercise' with the actual URL of the API

endpoint you want to use.

● Replace 'YOUR\_API\_KEY' with a placeholder instructing users to replace it

with their own API key obtained from the API provider.

*async function:*

The code defines an asynchronous function named fetchData that likely takes an

id parameter as input. This id parameter might be used to specify the ID of a

particular exercise or category of exercises to fetch.

*fetch request:*

● Method: GET (to retrieve data from the server)

● URL: The API endpoint URL where exercise data resides.

*Handling the Response:*

● The then method is used to handle the response from the API request. If the

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 code snippet shows a function called *fetchRelatedVideos* that fetches data

from YouTube using the RapidAPI service. Here's a breakdown of the code:

*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:

*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.

**Hero component**

this section would showcase trending workouts or fitness challenges to grab

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.

**Category page**

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.

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 videos

demonstrating proper form. .

Demo link:

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

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