PROJECT DOCUMENTATON

INTRODUCTION

Project Title: FITFLEX Your personal fitness companion

TEAM ID: SWTID1741606347148271

TEAM SIZE:5

TEAM MEMBERS	EMAIL ID
Ravinaa K	ravinaaisu@gmail.com
Rendhika Sri A	loshinimarar2003@gmail.com
Selva Suguna C	selvasuguna556@gmail.com
Sneha K	snehavalli0802@gmail.com
Valarkodi K	vvalarkodi@gmail.com

PROJECT OVERVIEW:

The purpose of the Fitflex is it empowers individuals to achieve their health and wellness goals from any location,at any time. Its should platform allows users to access personalized workout plans, nutritional guidance, instructional videos, and more.

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:

a. **User-Friendly Experience:** Develop an intuitive interface that facilitates easy navigation, enabling users to effortlessly discover, save, and share their preferred workout routines.
b. **Comprehensive Exercise Management:** Provide robust features for organizing and managing exercise routines, incorporating advanced search options for a personalized fitness experience.
c. **Technology Stack:** Harness contemporary web development technologies, with a focus on React.js, to ensure an efficient and enjoyable user experience.

Features of SB Fitzz:

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: 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.

Importance of fitflex:

The Fitflex app is designed to train people for fitness anywhere, anytime. This app provides training plans containing short, fitness goal-centric workouts from top international and local trainers.it also includes personalized nutrition plans for its users.

Testing Scope

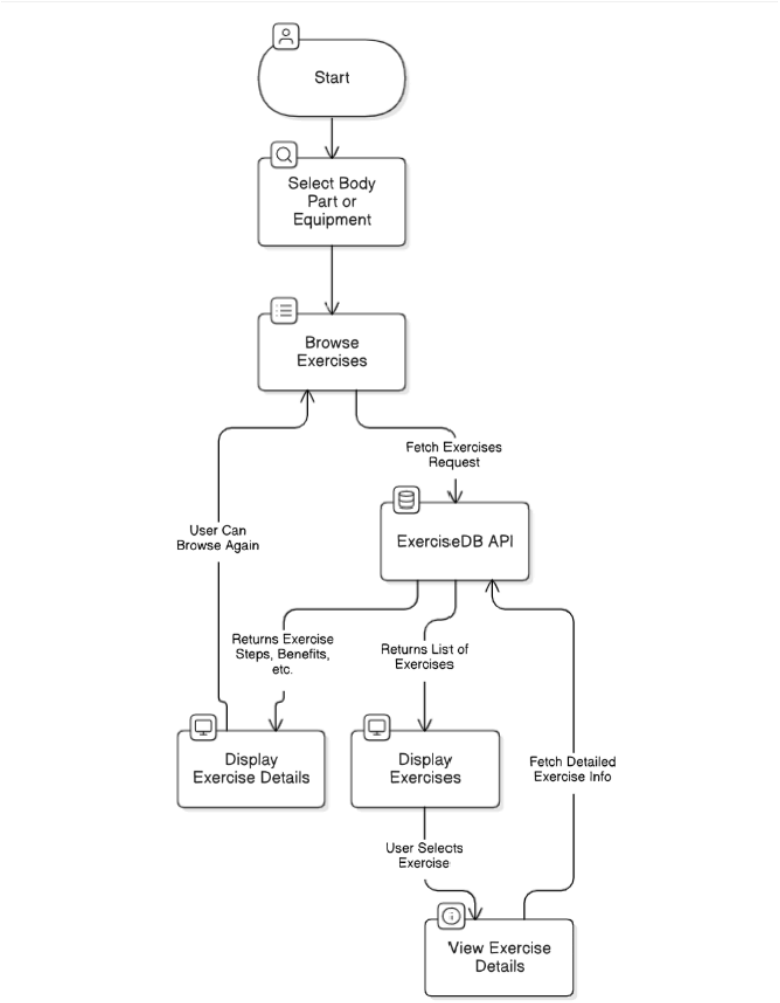
- Features and Functionalities to be Tested
- Home Page & Navigation
- Exercise Search & Discovery
- API Integration for Exercise Data
- Filtering Exercises by Body Part & Equipment
- Viewing Exercise Details
- UI/UX Testing (Responsiveness, Icons, Styling)
- Error Handling & Performance Testing

User Stories or Requirements to be Tested

- Searching & Viewing Exercises
- Filtering Exercises by Body Part & Equipment
- Displaying Exercise Details with Instructions
- Responsive UI across Mobile, Tablet, and Desktop
- Handling API Errors Gracefully

Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.The User selects a body part or equipment.



- The request is sent to Browse Exercises, which fetches relevant data from ExerciseDB API.
- The API returns a list of exercises, which is displayed to the User.
- The User selects a specific exercise, triggering the View Exercise Details process.
- The ExerciseDB API provides detailed exercise information.
- The app displays the details, and the User can either browse more exercises or select another one.

Problem – Solution Fit Overview:

The **Problem-Solution Fit** ensures that the identified problem aligns with the needs of users and that the proposed solution effectively addresses it. This concept helps developers, marketers, and business strategists validate the **necessity and effectiveness** of their solution before further development.

Purpose:

- Address the **lack of a structured and interactive fitness guidance platform** for users who seek customized exercises based on body parts or equipment.
- Provide an intuitive and engaging experience for users to **discover exercises quickly** without the need for manual research.
- Offer seamless navigation and **real-time data retrieval** from **ExerciseDB API** to enhance user experience.
- Improve accessibility and engagement through an **interactive UI, responsive design, and well-structured data flow**.

Problem Statement:

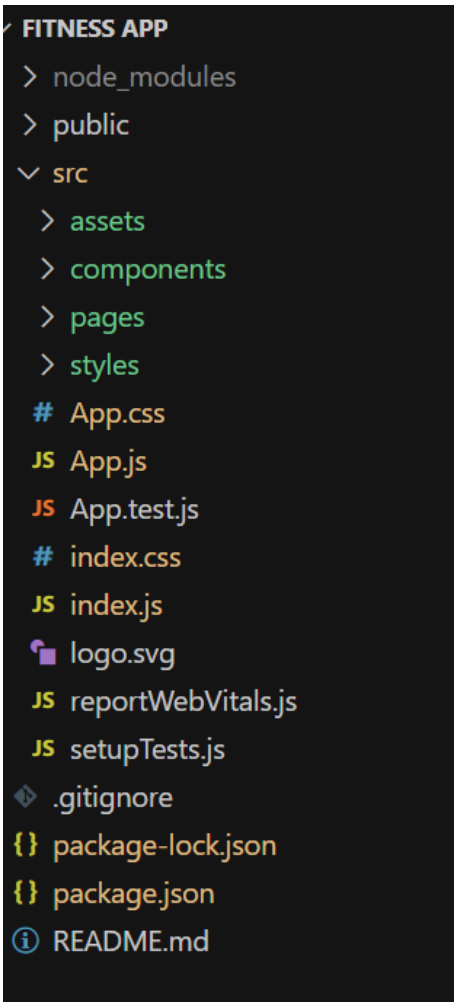
Many users struggle to find **relevant and structured exercise information** online, leading to frustration and inconsistency in their fitness journey. Most available platforms either require paid memberships or provide unstructured exercise listings without filtering options based on equipment or body parts.

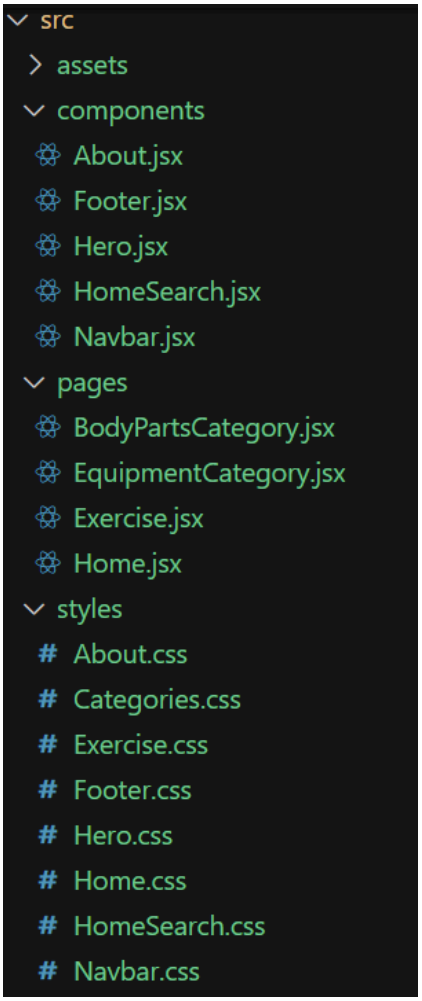
Solution:

- A **React.js-based Fitness Web Application** that provides users with an easy-to-navigate interface to explore exercises by **body parts and equipment**.
- Integration with **ExerciseDB API** ensures users get **up-to-date and detailed exercise information** with images and descriptions.
- **Axios-powered API requests** ensure smooth data retrieval with minimal delays.
- **Categorization and search functionalities** improve accessibility and user engagement.
- A scalable and **responsive UI design** ensures seamless experience across different devices.

PROJECT STRUCTURE:

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.

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

Milestone 2: Project Development

- ❖ Setup the Routing paths

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

Ex:

```
<div className="App">

  <Navbar />

  <Routes>

    <Route path="/" element={<Home />} />
    <Route path="/bodyPart/:id" element={<BodyPartsCategory />} />
    <Route path="/equipment/:id" element={<EquipmentCategory />} />
    <Route path="/exercise/:id" element={<Exercise />} />
  </Routes>

  <Footer />

</div>
```

- ❖ 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.

```
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);
  }
}
```

- **Fetching exercises under particular category**

```
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);
    }

}
```

- **Fetching Exercise details**

```
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);
    }

}
```

- **Fetching related videos from YouTube**

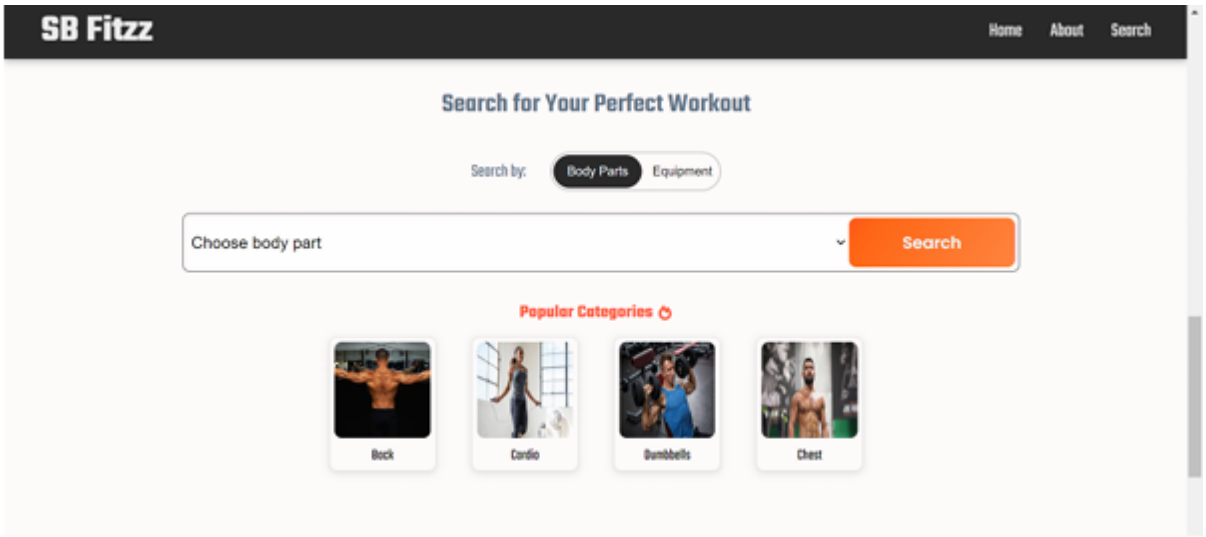
```
const fetchRelatedVideos = async (name)=>{
  console.log(name)
  const options = {
    method: 'GET',
    url: 'https://youtube-search-and-download.p.rapidapi.com/search',
    params: {
      query: `${name}`,
      hl: 'en',
      upload_date: 't',
      duration: 'l',
      type: 'v',
      sort: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

SCREENSHOTS OF PROJECT EXECUTION

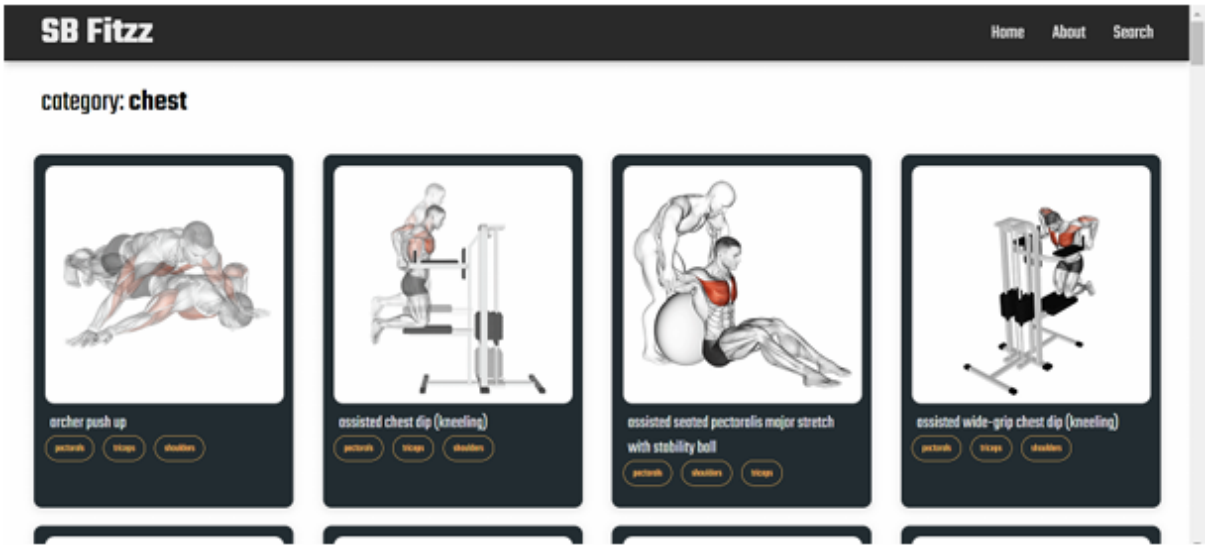
Slide of Search page:

SB 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



Slide of 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.



Demo link:
https://drive.google.com/file/d/1H_mb4hYd0M4Cu8D8eEFiEKByOEP1_Ak6/view?usp=drivesdk

CONCLUSION:

Fitflex is a comprehensive fitness platform that offers personalized plans, nutrional guidance,and community support,addressing the need for Customization, engagement in the fitness market.by leveraging AI for tailored plans and providing a user-friendly interface.