Frontend Development with React.js

Project Documentation [ADKRS FITZZ]

Introduction

Project Title: ADKRS FITZZ **Team**
Members:

- Aarthi K– Team Lead, Frontend Developer

- @Mailtoaarthi3@gmail.com

- Divya K – Documentation

- @kdivya2394@gmail.com

- Karlina Priya M- API Integration Specialist

- @vishaliv655@gmail.com

- Rahina rose B – Ideation

- @rahiiiinarose18@gmail.com

- Shirisha N - Demonstration

- @narrasiri355@gmail.com

Project Overview

Purpose:

ADKRS FITZZ Fitness websites serve as valuable platforms that provide users with information, resources, and tools to support their health and wellness goals. They offer workout plans, nutritional advice, fitness tracking features, and expert guidance to help individuals stay motivated and achieve their desired fitness levels. Many fitness websites also include articles, videos, and interactive communities where users can connect with like-minded individuals for support and encouragement. Whether for weight loss, muscle building, or overall well-being, these websites play a crucial role in promoting a healthy and active lifestyle.

Features:

- Customized exercise programs for different fitness levels and goals.

- Fitness logs, weight tracking, and performance analytics

- Demonstrations of exercises and live workout sessions

- Interaction with trainers and fellow fitness enthusiasts for motivation.

Architecture

Component Structure:

- **Navbar Component** – Manages navigation links and user authentication status.

- **Hero Component** – Highlights key features or promotions with engaging visuals.

- **Categories Component** – Displays various fitness categories or programs available.

- **Footer Component** – Contains contact information, social media links, and additional navigation.

- **Context API** – Provides a way to pass data through the component tree without having to pass props down manually at every level.

State Management: The application uses **React Context API** to manage global state, particularly for user preferences (such as dark mode) and API responses.
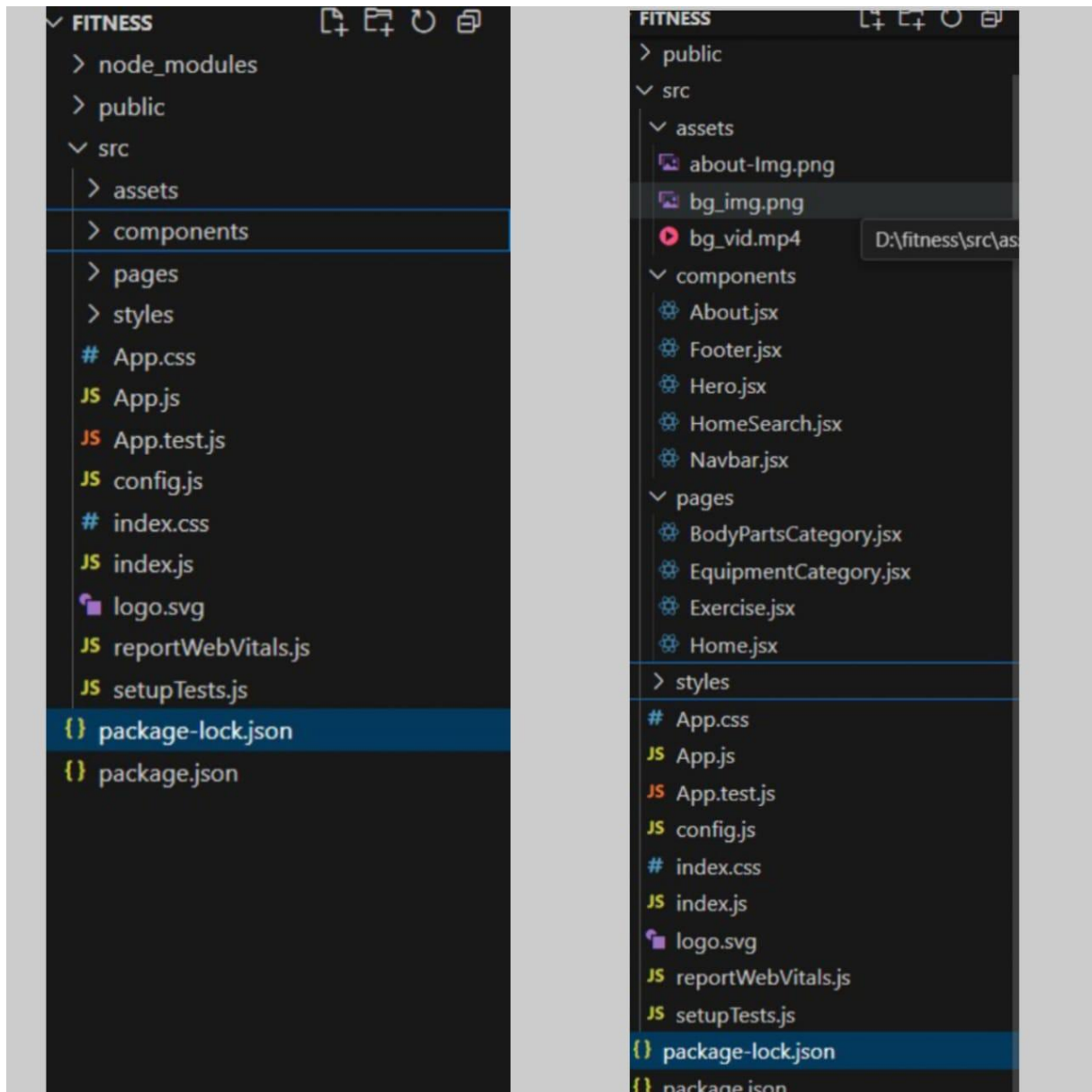

Setup Instructions

Prerequisites:

- Node.js (>=14.x)

- npm or yarn package manager

- Basic knowledge of React.js **Installation:**

```
# Clone the repository git clone

https://github.com/Divyakd12/FitFlex_AarthiK

# Navigate to the project directory

cd ADKRS-

# Install dependencies
npm install
```

Running the Application

To start the frontend server locally, use:

```
npm start
```

This will launch the development server at **http://localhost:3000**.

Component Documentation

Key Components:

- **Navbar:** Displays navigation links and search functionality.

- **Hero Section:** Highlights featured fitness videos.

- **Categories:** Provides category-based filtering.

- **Footer:** Contains site information and social links.

Reusable Components:

- **Button:** A universal button component for various UI interactions.

- **Loader:** Displays a loading while fetching videos.

- **Modal:** Used to show full fitness related videos

State Management

Global State:

- **Workout Plans :** manage user-specific workout routines and progress data globally, allowing for personalized experiences and consistent access.

Local State:

- **Session-Based Activities:** Manage temporary states like current workout sessions or real-time feedback locally to ensure responsive and efficient user experiences.

User Interface

- **Homepage:** Display search option for view more.

- **Category Page:** Shows fitness videos filtered by category.

- **Search Page:** Displays results based on user queries.

- **Article Page:** Opens full fitness video with a link to the original source.

Styling

CSS Frameworks/Libraries:

- Tailwind CSS for rapid UI development.

- React Icons for enhanced UI elements.

Theming:

- Dark mode toggle using Context API.

- Responsive design using CSS media queries.

Testing

Testing Strategy:

- **Unit Testing:** Using Jest for testing individual components.

- **Integration Testing:** React Testing Library for simulating UI interactions.

Code Coverage:

- Ensuring 80%+ test coverage with Jest and React Testing Library.

Screenshots or Demo

—— About Us

Embark on a Fitness Odyssey with SB Fitzz..

Welcome to ADKRS Fitzz, where the beat of cutting-edge workouts harmonizes with the contagious enthusiasm of a community that's more like family. Our mission? To help you unleash your full potential, one energizing workout at a time. Embark on a fitness journey that's not just about exercise; it's a dance of self-discovery and empowerment. Join the movement. Find your rhythm. Redefine fitness. This is SB Fitzz, where every step you take brings you closer to the best version of yourself.

Search for Your Perfect Workout

Search by: Body Parts Equipment

Choose body part   Search

Popular Categories 🔥

Back   Cardio   Dumbbells   Chest

---

category: cardio

jack burpee
cardiovascular system quadriceps hamstrings

mountain climber
cardiovascular system core shoulders

run (equipment)
cardiovascular system quadriceps hamstrings

run
cardiovascular system quadriceps hamstrings
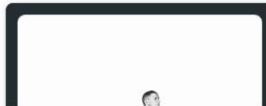
```jsx
<div className={scrollPosition > 100 || window.location.href !== "http://localhost:3000/" ? 'navbar nav-scrolled':`navbar`}>
    <h3 onClick={()=> navigate('/')} > ADKRS FITZZ</h3>

    {url === "http://localhost:3000/" || url === "http://localhost:3000/#about" || url === "http://localhost:3000/#search" ?
     <ul>
        <li onClick={()=> navigate('/')} >Home</li>
        <a href="#about"><li>About</li></a>
        <a href="#search"><li>Search</li></a>
     </ul>

      :

      <ul>
        <li onClick={()=> navigate('/')} >Home</li>
        <li onClick={()=> navigate('/#about')} >About</li>
       <li onClick={()=> navigate('/#search')}>Search</li>
      </ul>
    }
```

```jsx
const fetchData = async (id) => {
    console.log("Fetching data for body part:", id);
    const options = {
        method: 'GET',
        url: `https://exercisedb.p.rapidapi.com/exercises/bodyPart/${id}`,
        params: {limit: '200'},
        headers: {
          'X-RapidAPI-Key': CONFIG.EXERCISE_API_KEY,
          'X-RapidAPI-Host': 'exercisedb.p.rapidapi.com'
        }
    };

    try {
        const response = await axios.request(options);
        console.log("API Response:", response.data);
        setExercises(response.data);
    } catch (error) {
        console.error("API Error:", error);
    }

}
```

```jsx
    <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>
  );
}
```

Known Issues

- Lack of Mobile Responsiveness

- Limited Accessibility Features

Future Enhancements

- AI-Powered Personalization.

- Enhanced Data Analytics.

- Interactive Fitness Journey Mapping.

- Expand API sources for a richer fitness video experience.