***Frontend Development with React.js***

***Project Documentation for FitFlex***

1. **Introduction** 
   * **Project Title: FitFlex**
   * **Team Members**:

**Jothi Lakshmi B** [Email Id: jothilakshmi314@gmail.com]

**Kalaiarasi S** [Email Id: kalaiarasis1806@gmail.com]

**Thanuja U**  [Email Id: thanujau626@gmail.com]

**Prasanthi V** [Email Id: poojavadivel16@gmail.com]



**Janani P** [Email Id: jananipalani55@gmail.com]

1. **Project Overview** 
   * **Purpose**:

The core idea behind FitFlex is to provide a tailored fitness experience, whether it's through an app, platform, or community-driven initiative. Here there are many categories for the workouts and also popular categories like Bock, Cardio, Dumbbells, Chest. There is also a related videos of the categories.

* + **Features**:
    - Many Categories Available
    - Popular categories(Bock, Cardio)
    - Video is available for each workout
    - Responsive design for mobile and desktop.

1. **Architecture** 
   * **Component Structure**:

The application is built using React.js with a component-based architecture. Major components include:

* + - **Header**: Contains the navigation bar and search bar.
    - **AboutUsPage**: about the SB fitzz
    - **Footer**: Categories of Body parts and equipment
    - **HomePage**: view more, About us, and Popular categories.
    - **SearchPage**: Allows users to search workout types, equipment, and categories.
  + **State Management**:

The application uses **Redux** for global state management. The Redux store manages the categories, body parts, equipment

* + **Routing**:

The application uses **React Router** for navigation. Routes include:

* /: Home page
* /search: Search page
* Bodypartcategory: body part workouts

1. **Setup Instructions** 
   * **Prerequisites**:

* Node.js (v16 or higher)
* npm (v8 or higher)
* Git
  + **Installation**:
    1. Clone the repository: git clone

<https://github.com/unm1737121016/FitFlex.git>

* + 1. Navigate to the client directory: cd fitflex/client
    2. Install dependencies: npm install
    3. Configure environment variables: Create a .env file in the client directory and add the necessary variables (e.g., API keys).
    4. Start the development server: npm start

1. **Folder Structure** 
   * **Client**:

o  **src/components:** # Reusable components (About, Footer, etc.) o  **src/pages:** # Page components (HomePage, Exercise, etc.) o **src/assets:** # Images, video, and other static files o **src/styles:** # about, categories, exercise, footer, hero, homesearch

1. **Running the Application**

**Frontend**:

* + - To start the frontend server, run the following command in the client directory:

npm start

* + - npm install o npx json-server ./db/db.json o npm run dev
    - The application will be available at http://localhost:3000

1. **Component Documentation** 
   * **Key Components**:

o **About**: Display about the fitness application

o **Footer**: Shows the categories of the body parts and equipments.

o **HomeSearch**: Displays the categories of the body part exercise and equipment.

* + **Reusable Components**:

O **Button**: A customizable button component.

O **Input**: A reusable input field for categories and equipments.

1. **State Management** 
   * **Global State**:

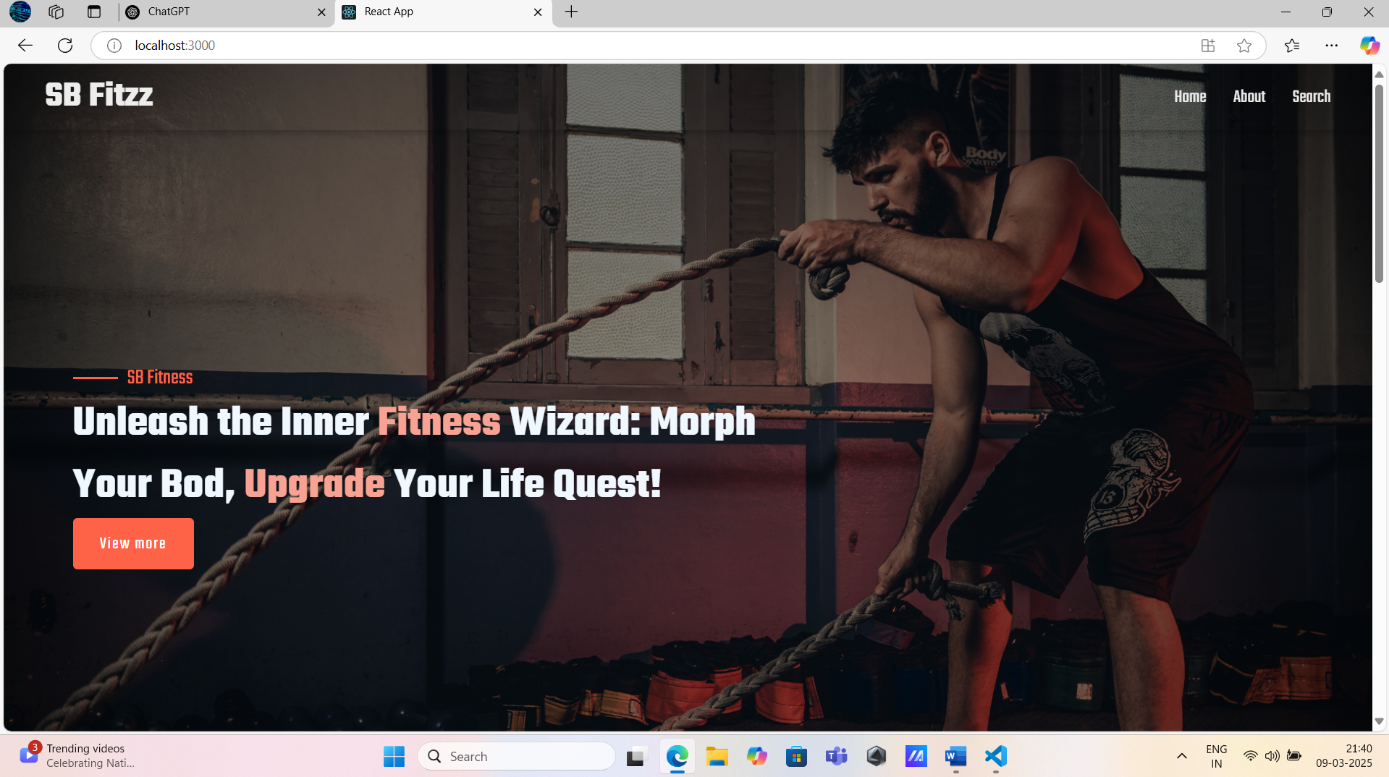
The Redux store manages the following global states:

* + - **user:** Current authenticated user.
    - **Exercise:** categories, instructions, video
    - **Categories:** Based on those exercises.
    - **searchResults:** Results from the search functionality.
  + **Local State**:

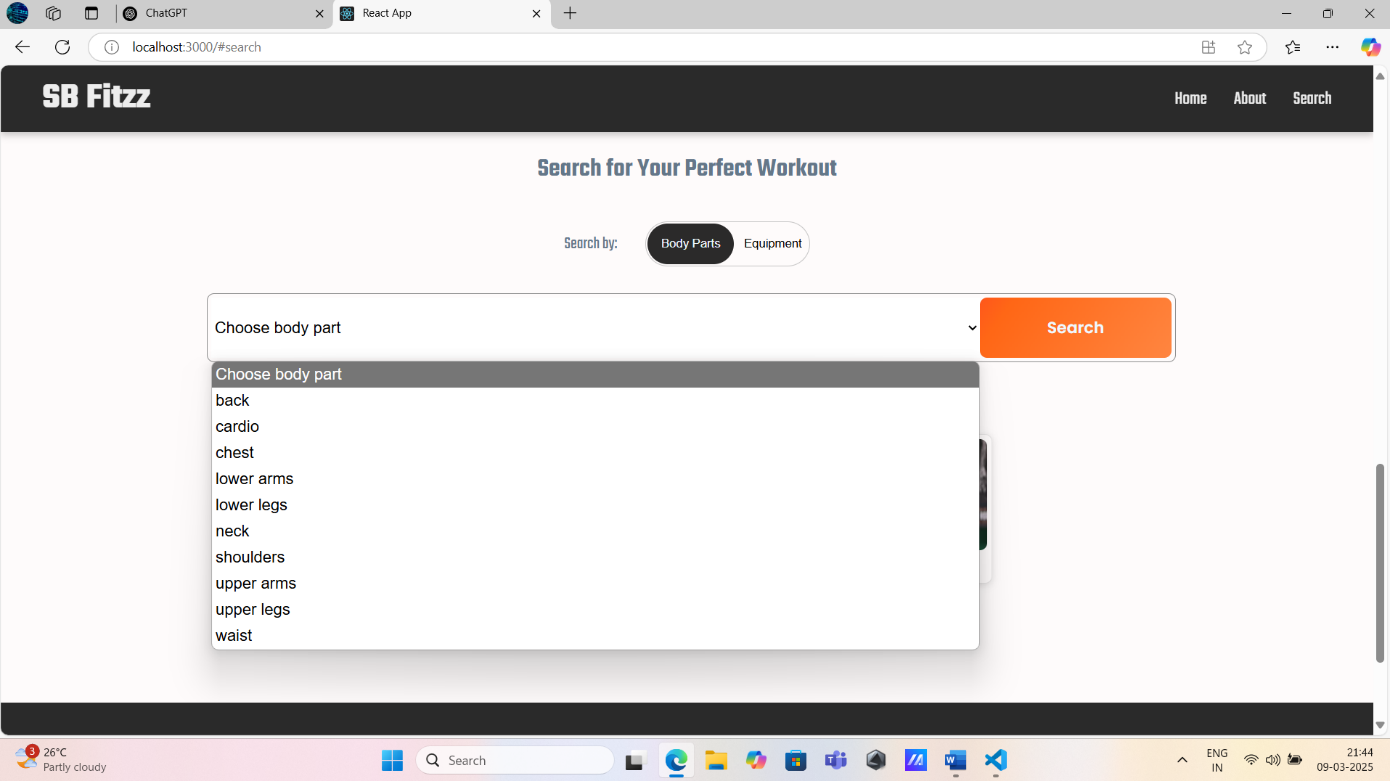
Local state is managed using React's useState hook within components. For example, the HomeSearch page component manages the search query input locally.

1. **User Interface** 
   * **Screenshots**

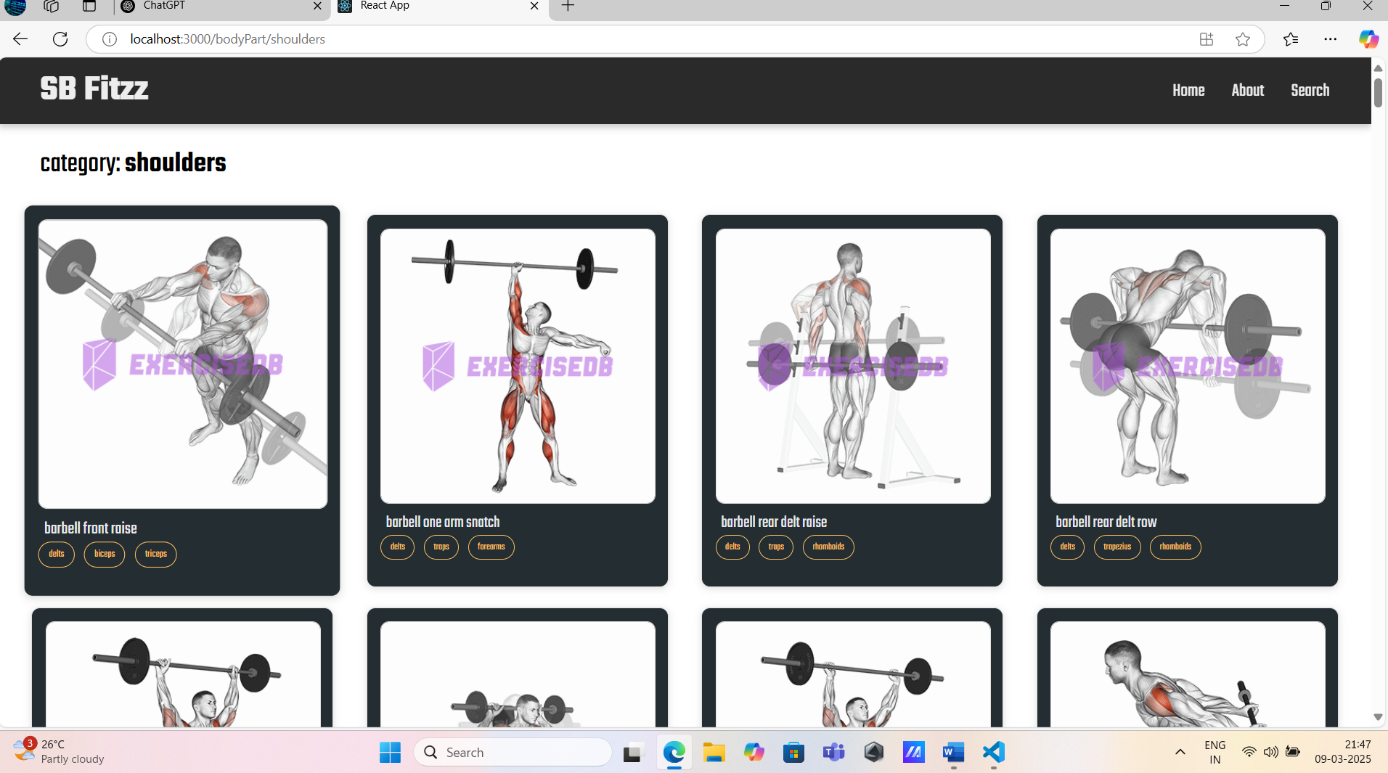
o **Home Page:** Display Fitness page and shows the about us and search button.



* + - **Search Page:** Search the body parts and equipment of workouts.



* + - **BodyPart Category:** It shows the different type of categories of the body parts.



1. **Styling**
   * **CSS Frameworks/Libraries**:

The application uses **Styled-Components** for styling. This allows for modular and scoped CSS within components.

* + **Theming**:

A custom theme is implemented using Styled-Components, with support for light and dark modes.

1. **Testing** 
   * **Testing Strategy**:
     + **Unit Testing:** Using **Jest** and **React Testing Library**.
     + **Integration Testing**: Is performed to ensure that components work together as expected.
     + **End-to-End Testing:** **Cypress** is used for end-to-end testing of user flows.
   * **Code Coverage**:
     + Code coverage is monitored using Jest’s built in coverage tool. The current coverage is 85%.

1. **Screenshots or Demo** 
   * **Demo Link:**

<https://drive.google.com/file/d/1Cz-MDR--J4vZ3bOYphU0reYSGbiVgIBV/view?usp=drivesdk>

* + **Screenshots:** See section 9 for UI screenshots.

1. **Known Issues** 
   * **Issue 1**: The Fitflex application suddenly the search button get struggle to search.

1. **Future Enhancements** 
   * **Future Features**:
     + Add support for user profiles and social sharing. o Implement a recommendation engine for personalized exercise suggestions.
     + Add progress track and transitions for a smoother user experience.

This documentation provides a comprehensive overview of the **FitFlex** project, including its architecture, setup instructions, and future plans.