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

***Project Documentation for FitFlex***

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

**Imthiyas S** (**Team Leader**) [Email Id[: imthiyas1625@gmail.com](:%20imthiyas1625@gmail.com)]



**Divakar B** [Email Id: divakarbaskaran21@gmail.com]

**Dhanush G.K**  [Email Id: ddhhaannuusshh@gmail.com]

**Kenin Luther king**  [Email Id: keninluther@gmail.com]

**Kavin K** [Email Id: kavinkaran1607@gmail.com]

1. **Project Overview** 
   * **Purpose**:   
      FitFlex is a cutting-edge fitness app designed to revolutionize how you engage with exercise routines. With an intuitive interface and diverse workout categories, it caters to both beginners and seasoned athletes. The app fosters a dynamic community, encouraging collaboration and sharing of effective exercises. FitFlex is your ultimate companion on the journey to a healthier, more active lifestyle.
   * **Features**:
     + Access a wide range of exercises from trusted fitness APIs.
     + Explore workout routines visually with curated image galleries.
     + Enjoy a clean, intuitive, and user-friendly interface.
     + Utilize an advanced search feature to find specific exercises and plans.
     + Discover new fitness challenges to match various goals and preferences.

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.
    - **Search Bar**: Provides search functionality for workouts, exercises.
    - **Home Page**: Displays featured exercises, recommended workouts.
    - **API Communication**: Handles the fetching and processing of data from external fitness APIs.
* **API Response Handling**: Ensures smooth handling of response data,like workouts and fitness tracking
  + **State Management**:

The application uses **Redux** for global state management, ensuring efficient data flow and consistency across the app. Redux is used to manage and share various fitness-related data such as user workouts, fitness progress, and API response data.

* + **Routing**:

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

* /:Home page
* /Search:Search page
* /exercise/:id:Exercise Detail Page
* /login:Login Page

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

o Node.js (v16 or higher) o npm (v8 or higher) o Git

* + **Installation**:
    1. Clone the repository: git clone https://github.com/Imthiyas1625/fitnessapp.git
    2. Navigate to the client directory: cd rhythmic-tunes/client
    3. Install dependencies: npm install
    4. Configure environment variables: Create a .env file in the client directory and add the necessary variables (e.g., API keys).
    5. Start the development server: npm start

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

* **src/components: # Reusable components (Body, Exercise, etc.)**
* **src/pages: # Page components (HomePage, IndexPage, etc.)**
* **src/assets: # Images, icons, and Exercise Detail files**
* **src/redux: # Redux store, actions, and reducers**
* **src/utils: # Utility functions and helpers**
* **App.js: # Main application component**
* **index.js: # Entry point**
  + **Utilities**:
    - **api.js**: Handles API requests to the backend.
    - **fetch.js**: Manages user authentication and token storage.

1. **Running the Application**

**Frontend**:

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

npm start

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

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

o **Header**: Displays the navigation bar and search bar.

▪ Props: navigation and search (for navigation links).

o **Home page**: To display featured exercises and recommended workout plans for the user.

▪ A list or grid of featured exercise(Displays workout plans).

o **API Communication**: To fetch data from external fitness APIs for exercises

▪API Client Makes HTTP requests to the fitness API to get data like exercises, workout plans, or statistics

* + **Reusable Components**:

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

▪ Props: text, onClick, disabled.

o **Input**: A reusable input field for forms and search. ▪ Props: type, placeholder, value, onChange.

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

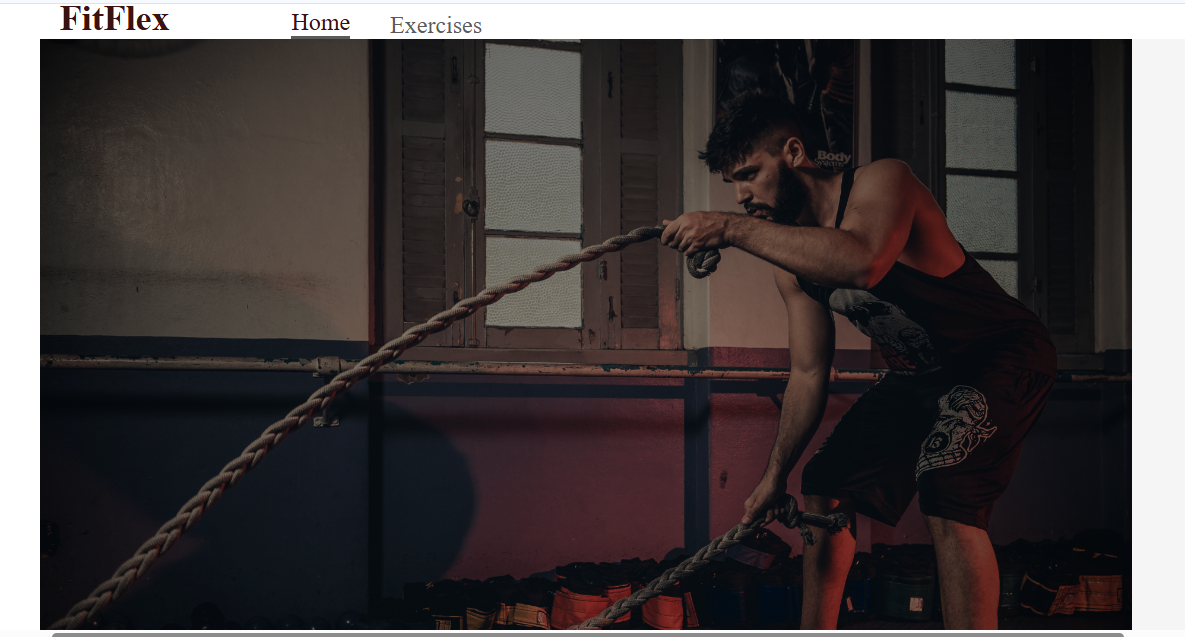
The Redux store manages the following global states:

* + - **user:** Current authenticated user.
    - **selectedBodyPart:** The body part selected by the user (e.g., Chest, Back, Legs) for exercises.
    - **exercises:** A list of exercises filtered by the selected body part or search criteria.
    - **searchResults:** The results from the search functionality (e.g., workout names, exercises).
  + **Local State**:

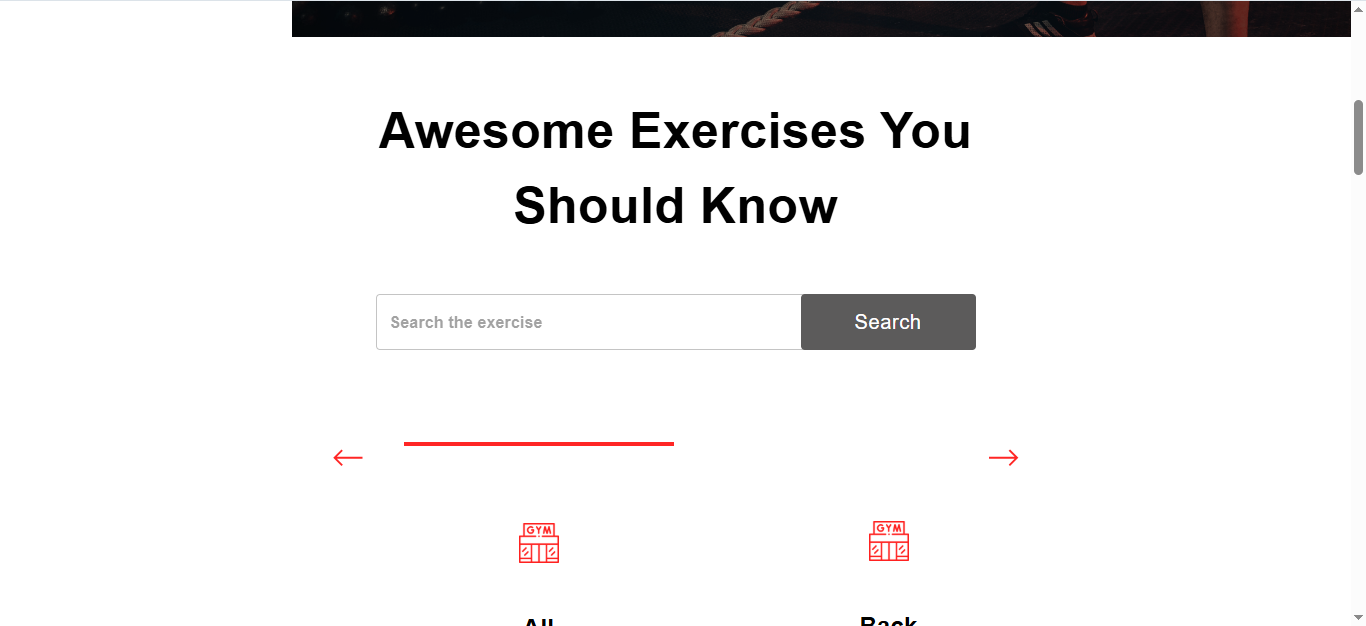
Local state is managed using React's useState hook within components. For example, the SearchPage component manages the search query input locally.

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

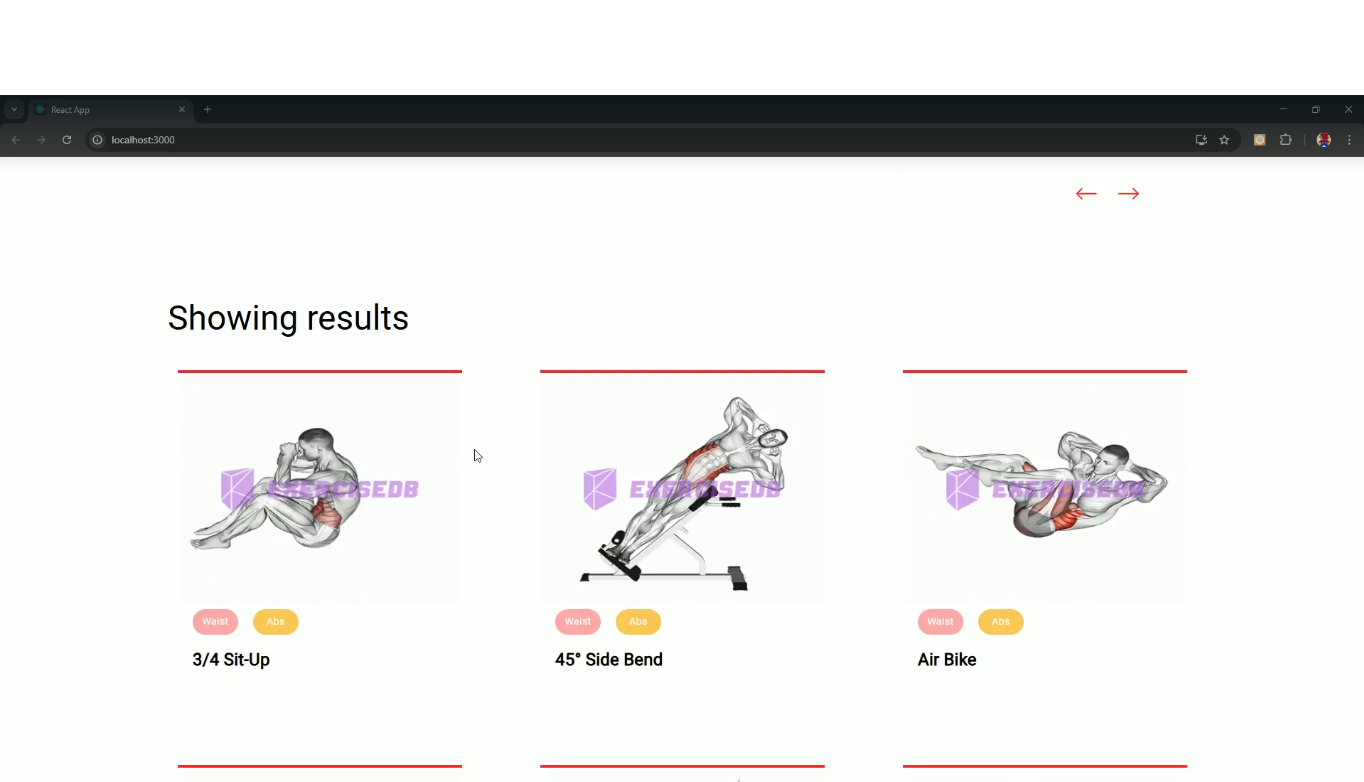
o **Home Page:** Display featured exercises and recommended exercises.



* + - **Search Page:** Allows users to search for exercise to the user.



* + - **Exercise Page:** Displays different type of exercise and their instructions and management.



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**:

o 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/1BgPufFmxaNM9sJSbkD2pbe02tqevg0f5/view?usp=drive_link>
   * **Screenshots:** See section 9 for UI screenshots.

1. **Known Issues** 
   * **Issue 1**: The search functionality is slow with large datasets.

1. **Future Enhancements** 
   * **Future Features**:
     + Add support for user profiles and social sharing.
     + Add animations 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.