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

***Project Documentation for Fitflex***

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

**Team Leader:** BHARATH M - [m.bharathcs22krmmc@gmail.com](mailto:m.bharathcs22krmmc@gmail.com)

**Team Member:** AUGUSTIN JABAKUMAR A - [augustinjabakumarcs22.krmmc@gmail.com](mailto:augustinjabakumarcs22.krmmc@gmail.com)

**Team Member:** ASHOK M S - [msashokcs22.krmmc@gmail.com](mailto:msashokcs22.krmmc@gmail.com)

**Team Member:** CHANDRU J - [jchandrucs22.krmmc@gmail.com](mailto:jchandrucs22.krmmc@gmail.com)

**Team Member:** DEEPAK R - [rdeepakcs22.krmmc@gmail.com](mailto:rdeepakcs22.krmmc@gmail.com)



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

FitFlex is a comprehensive bodybuilding program designed to help individuals build muscle, increase strength, and enhance overall fitness. The program combines expert-designed workout routines, personalized nutrition plans, and advanced recovery techniques to maximize gains efficiently.

* + **Features**:
    - **Structured Workout Plans**: Progressive resistance training with hypertrophy-focused routines.
    - **Personalized Nutrition**: Macronutrient-based meal plans tailored to muscle growth.
    - **Tracking & Progress**: Strength and physique tracking through structured milestones.
    - **Recovery & Optimization**: Sleep, mobility, and supplementation strategies for peak performance..

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

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

* + - **Workout Plans** (Customized for Beginner, Intermediate, Advanced)
    - **Exercise Library** (Proper form, video tutorials, modifications)
    - **Progressive Overload Tracking** (Weight, sets, reps tracking)
    - **Periodization Model** (Strength, Hypertrophy, Cutting, Maintenance phases).
  + **State Management**:

A fitness app needs an efficient **state management strategy** to handle user data, workouts, progress tracking, nutrition, and real-time updates. Below is a structured approach to **state management** for a FitFlex fitness app.

* + **Routing**:

In a fitness app, routing plays a key role in **navigating between screens**, handling authentication, and managing deep links for features like workout sessions, progress tracking, and community challenges.

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/Bharath1619/Fitflex.git
    2. Navigate to the client directory: cd Fitflex/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**:

o  **src/components:** # Reusable components (weight, height, etc.) o  **src/pages:** # Page components (HomePage, SearchPage, etc.) o **src/assets:** # Images, icons, and other static files o **src/redux:** # Redux store, actions, and reducers o **src/utils:** # Utility functions and helpers o **App.js:** # Main application component o **index.js:** # Entry point

* + **Utilities**:
    - **api.js**: Handles API requests to the backend.
    - **auth.js**: Manages user authentication and token storage.
    - **hooks**: Custom hook for managing the fitflex exercise.

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**:

🔹 **User Profile** → Name, age, weight, height, fitness goals.  
🔹 **Authentication** → Login, Signup, Social OAuth (Google, Apple, Facebook).  
🔹 **Subscription Management** → Free & premium plans, billing integration.  
🔹 **Settings & Preferences** → Dark mode, notifications, measurement units.

* + **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.

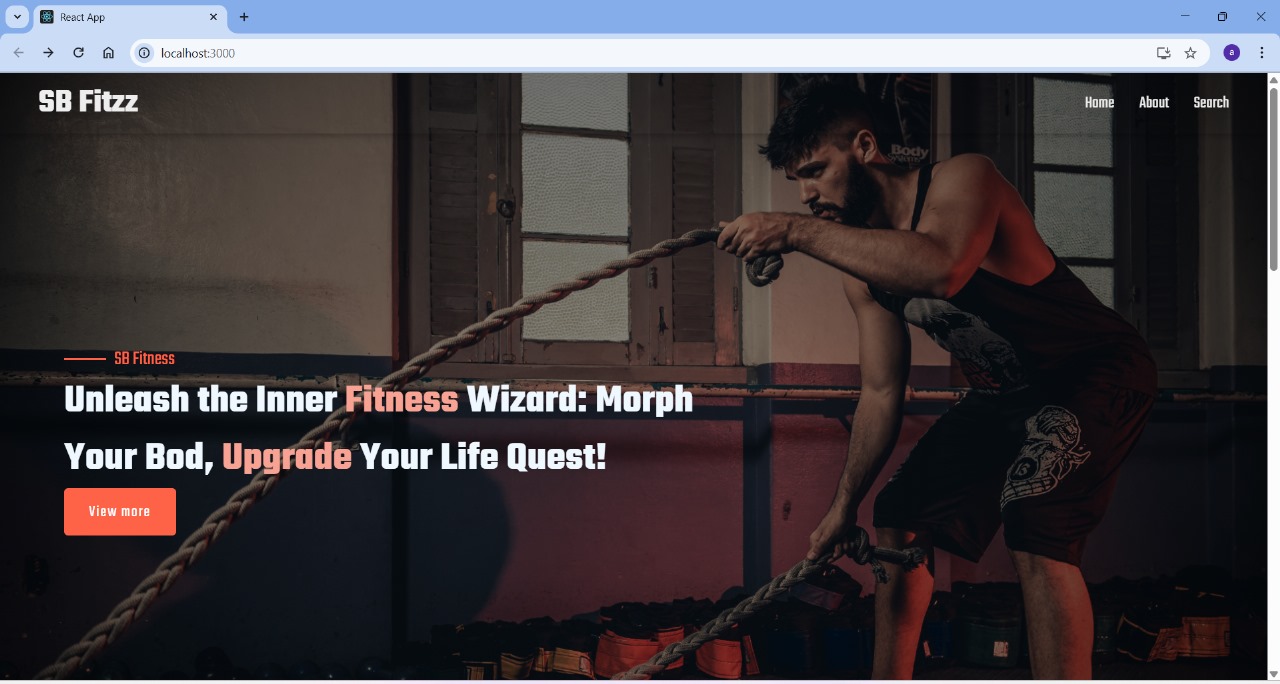
1. **State Management** 
   * **Global State**:

The Redux store manages the following global states:

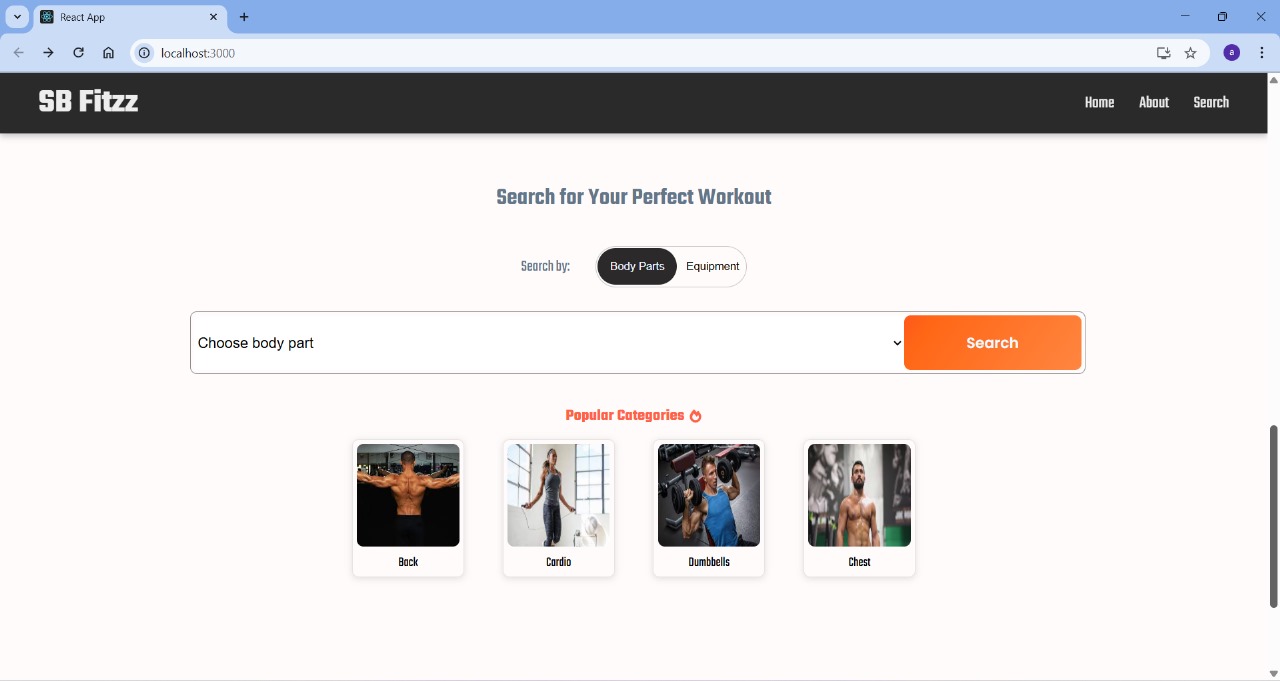
* + - **user:** Current authenticated user.
    - **Exercise:** Dummbbells ,Chest,Cardio
    - **searchResults:** Results from the search functionality.
  + **Local State**:

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

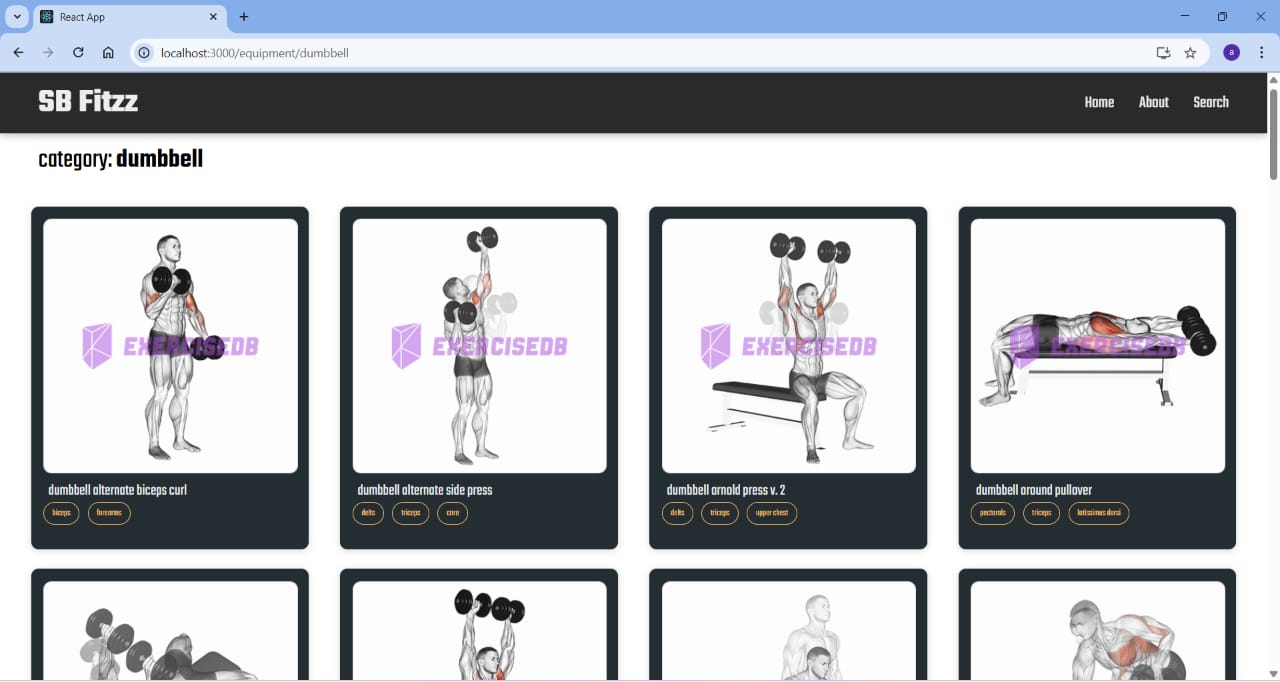
1. **User Interface** 
   * **Screenshots** o **Home Page:**



* + - **Search Page:** Allows users to search for Dummbbells ,Chest, and Cardio.



* + - **Dummbbells:** Displays user-created Dummbbells Exercise Management.



* + **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/17t_KQszALGdGVblHBtZA9N3glE9EWl2u/view?usp=drivesdk>
   * **Screenshots:** See section 9 for UI screenshots.

1. **Known Issues** 
   * **Issue 1**: Building a **fitness app** comes with challenges in **user engagement, data accuracy, security, and scalability**. Below are some **common issues** along with possible solutions.
   * **Issue 2**: The search functionality is slow with large datasets.

1. **Future Enhancements** 
   * **Future Features**:
     + Add support for user profiles and social sharing. o Implement a recommendation engine for personalized music suggestions.
     + 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.