**FitFlex – React application documentation**

**1. Introduction**

* **Project Title:** FitFlex(Your Personal Fitness Companion)
* **Team Leader:** Rakesh Kumar K
* **Team Members:**
  + Praveen K
  + Prushothaman R
  + Rohith K
  + Saravanan N

**2. Project Overview**

* **Purpose:**  
  The Fitness Tracker App is designed to help users track their physical activities, workout progress, calorie intake, and health goals. It allows users to log exercises, track progress over time, and maintain motivation through personalized feedback and goals.
* **Features:**
  + **User Authentication:** Users can sign up, log in, and track their data securely.
  + **Exercise Logging:** Users can log various exercises with the date, type, and duration.
  + **Progress Tracking:** Visual charts and graphs showing the user's workout progress.
  + **Goal Setting:** Users can set fitness goals and monitor their progress toward achieving them.
  + **Daily/Weekly Reports:** Users can view reports of their activity over time.
  + **Responsive Design:** Fully responsive design optimized for both desktop and mobile.

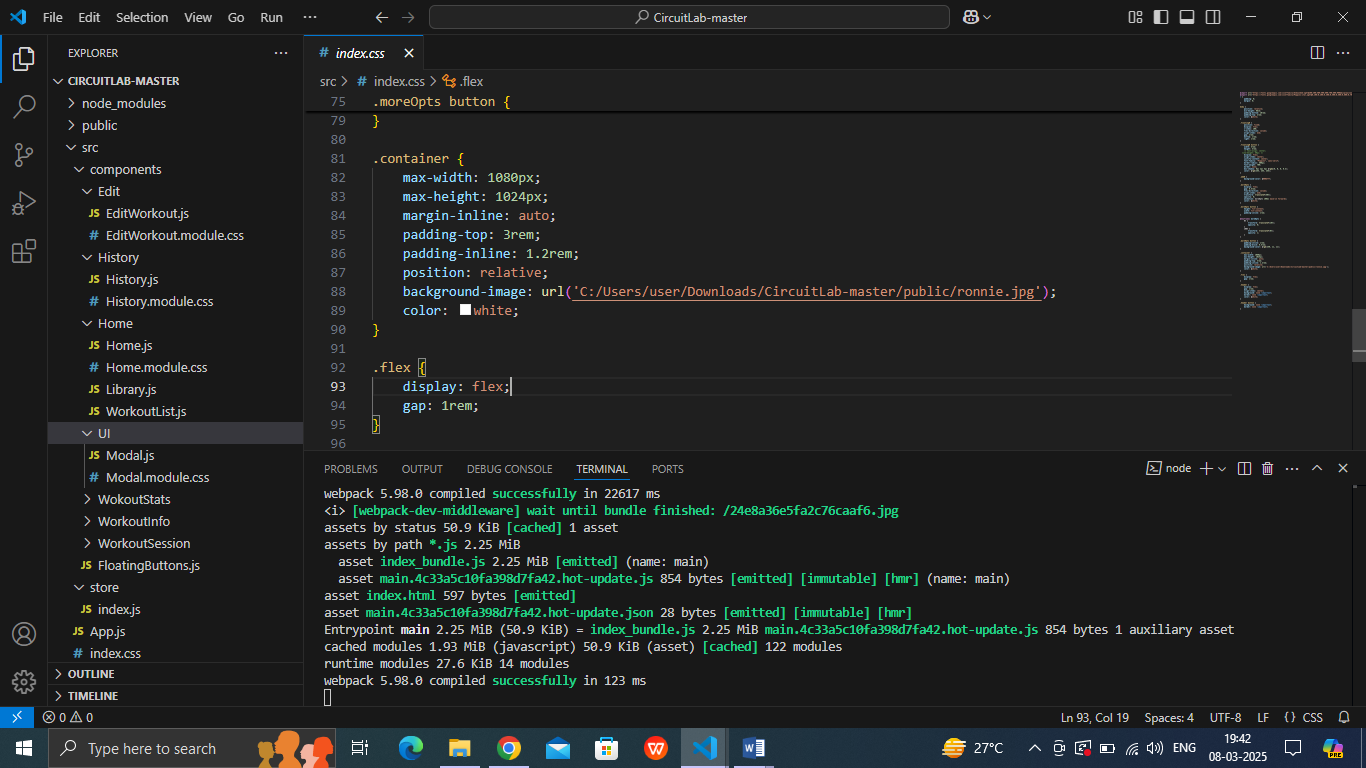
**3. Architecture**

* **Component Structure:**  
  The app is built using ReactJS, with the following major components:
  + **App:** The main wrapper component.
  + **Header:** Displays the navigation and user info.
  + **Dashboard:** Main page displaying user data and progress.
  + **ExerciseLog:** Component for logging exercises.
  + **GoalTracker:** Displays current user goals and progress.
  + **Reports:** Shows daily/weekly reports in graph format.
  + **Footer:** Basic footer with contact info and privacy policy links.
* **State Management:**  
  State management is handled via the **Context API** to provide a global state for user authentication, exercise logs, and goals. The app uses **React Hooks** for managing local component states (e.g., form inputs).
* **Routing:**  
  React Router is used for page navigation. The routing structure includes:
  + / - Home page
  + /workout - Workout page
  + /dashboard - User dashboard
  + /exercise – View exercise
  + /history - User history page

**4. Setup Instruction**

* **Prerequisites:**
  + Node.js
  + npm
  + Git
* **Installation:**
  + Clone the repository:  
    git clone https://github.com/username/fitness-tracker-app.git
  + Navigate to the project directory:  
    cd circuitlab-master
  + Install dependencies:  
    npm install
  + Create an .env file in the root directory and set up any necessary environment variables (e.g., API keys, database connections).
  + Start the development server:  
    npm start

**5. Folder Structure**

* **Client:**  
  The React application is located in the client folder, organized as follows:
* **Utilities:**
  + useFetch.js - Custom hook for fetching data from APIs.
  + calculateCalories.js - Helper function for calculating calories burned based on activities.
  + validateInputs.js - Input validation for forms.

**6. Running the Application**

* **Frontend:**  
  Run the following command in the client directory:  
  npm start

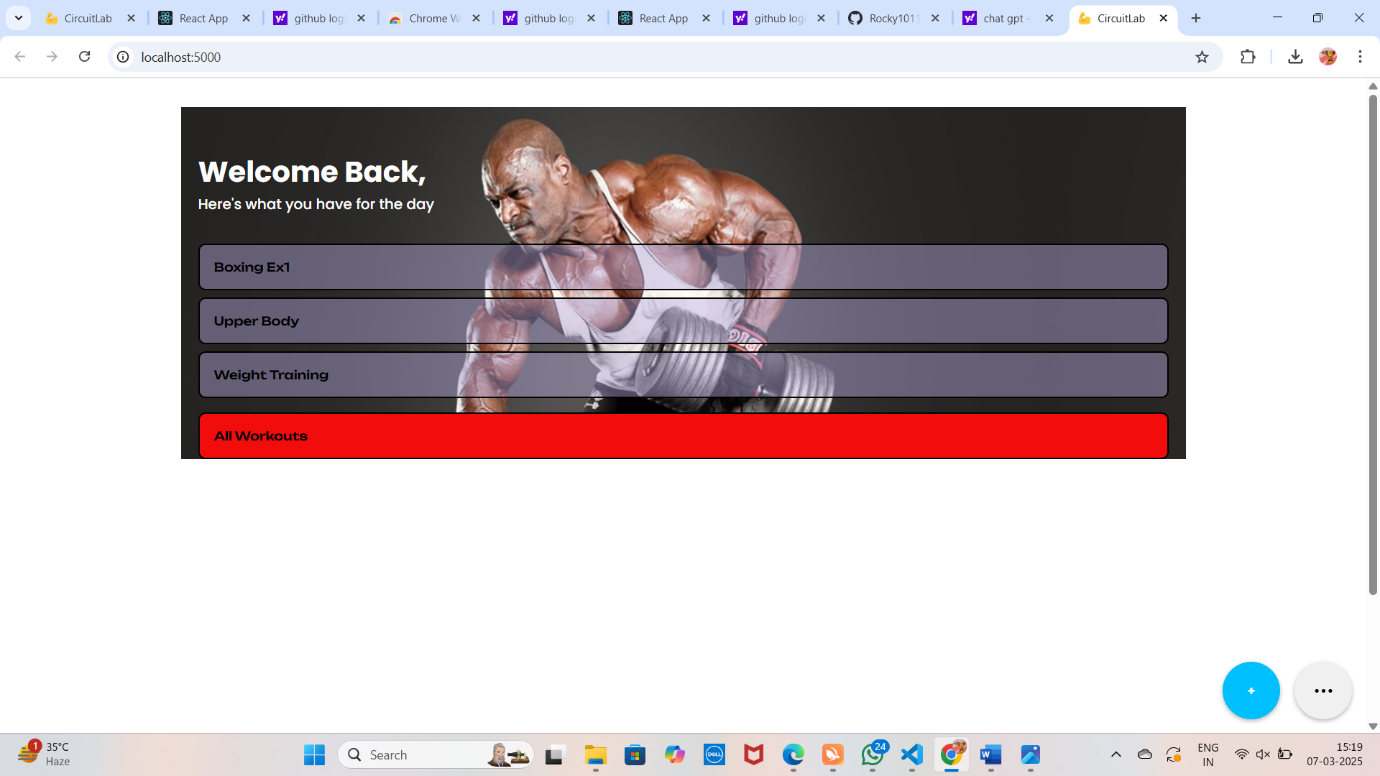
**7. Component Documentation**

* **Key Components:**
  + **ExerciseLog:**
    - * onAddLog (function) - Callback function to handle adding a new exercise log.
      * user (object) - Current user information.
  + **GoalTracker:**
    - * goal (object) - The user's fitness goal (e.g., weight loss, muscle gain).
      * progress (number) - The current progress toward the goal.
* **Reusable Components:**
  + **Button:** A reusable button component used throughout the app.
    - * label (string) - Text to display on the button.
      * onClick (function) - Function to execute when the button is clicked.
  + **Input:** A reusable input field component
    - * type (string) - Type of the input
      * value (string) - Input field value.

**8. State Management**

* **Global State:**  
  The global state is managed using the React Context API. This includes the user's authentication status, exercise logs, and fitness goals.
* **Local State:**  
  Local states are handled using React useState hooks within components.

**9. User Interface**

* **Screenshots:**
  + Dashboard: Displays a summary of the user's progress with exercise logs and goals.
  + Exercise Log Form: Interface for logging new exercises.
  + Goal Setting: Allows users to set personal fitness goals.

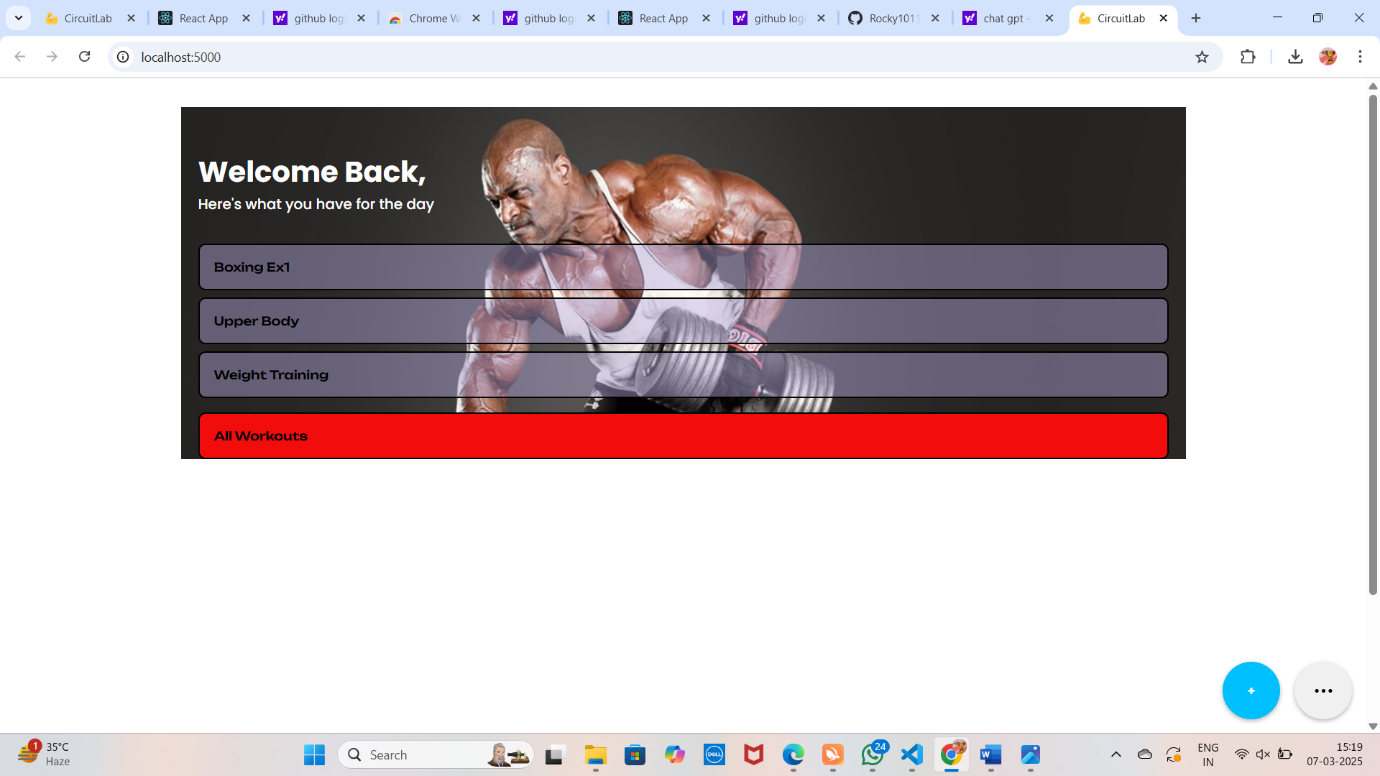
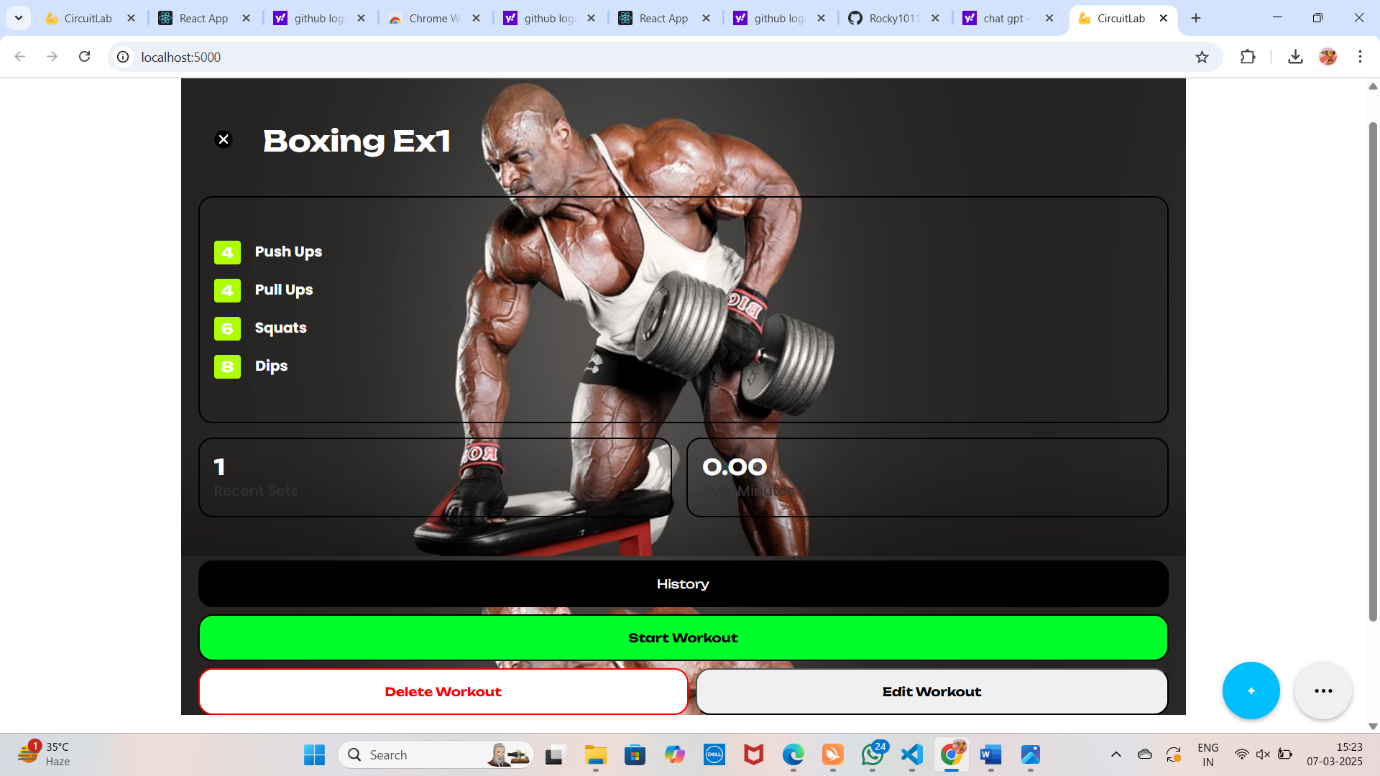
**10. Styling**

* **CSS Frameworks/Libraries:**
  + **Styled-Components** for styling React components.
  + **CSS Grid & Flexbox** for responsive layouts.
  + **Font Awesome** for icons.
* **Theming:**
  + Custom themes for light and dark modes are implemented using styled-components' theming capabilities.

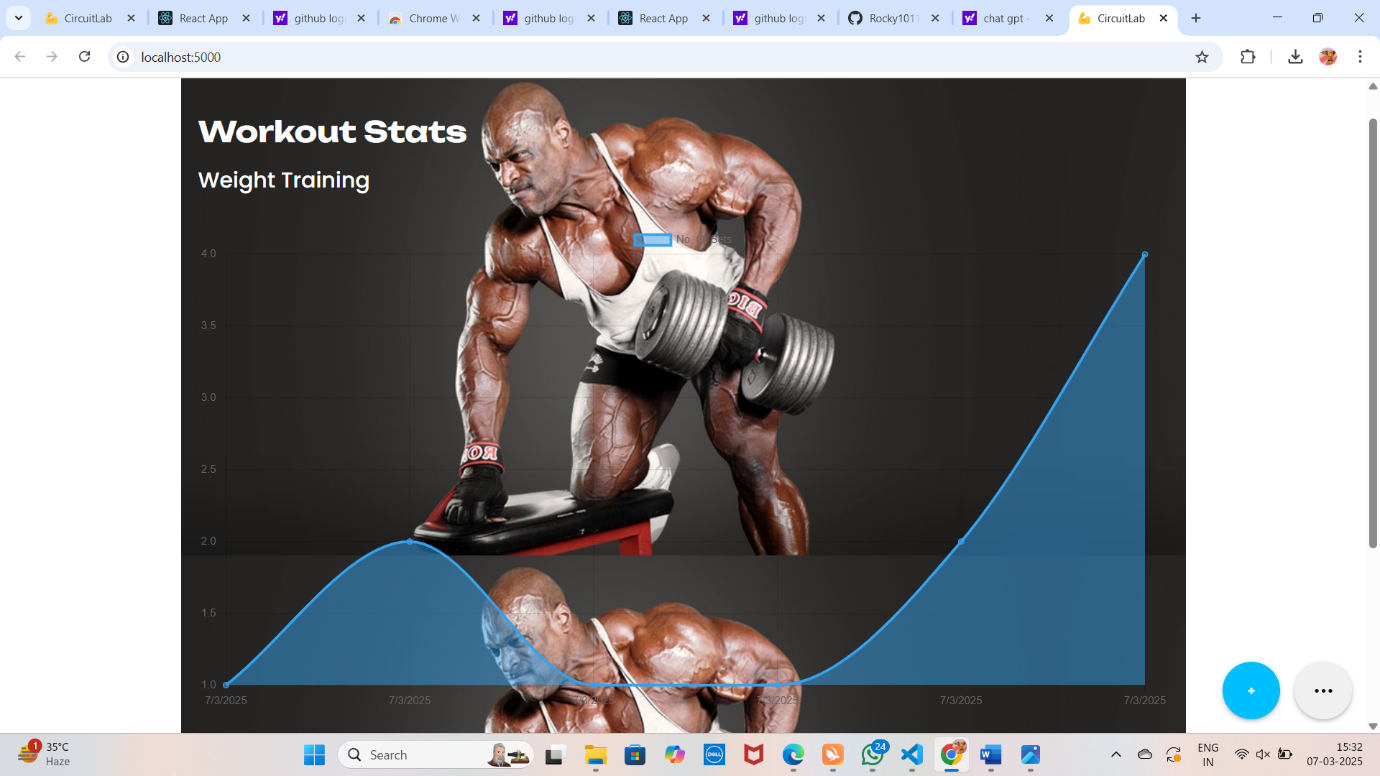
**11. Testing**

* **Testing Strategy:**
  + **Unit Testing:** Using Jest and React Testing Library to test individual components.
  + **Integration Testing:** Ensures that multiple components work together as expected.
  + **End-to-End Testing:** Using Cypress to simulate user behavior and interactions across the app.
* **Code Coverage:**
  + Code coverage is tracked using Jest’s built-in coverage tool to ensure all major components and functions are tested.

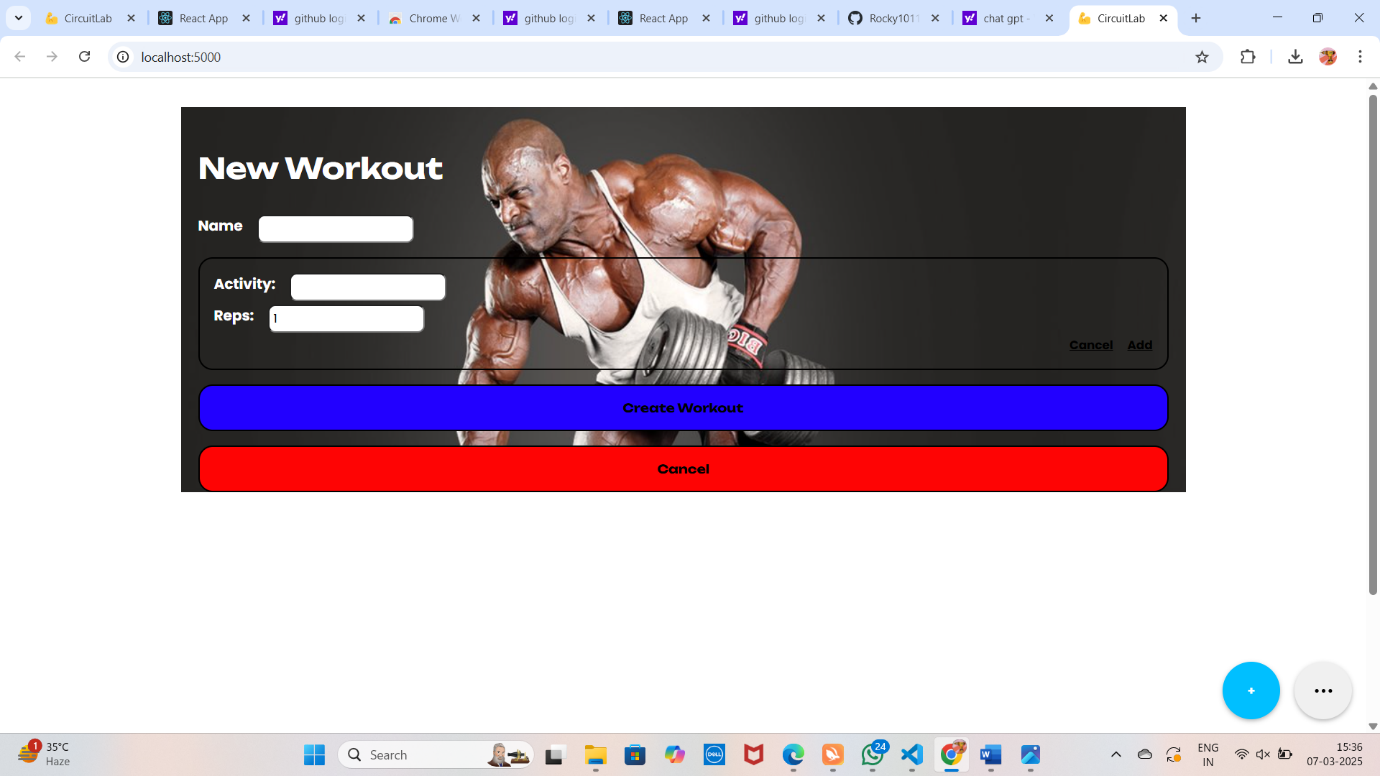
**12. Screenshots or Demo**

* Home
* Category page

History page



Workout create page



**13. Known Issues**

* **Issue 1:** Some mobile devices might not render the charts correctly due to screen size limitations.
* **Issue 2:** Occasionally, the user session may expire without warning when there is network instability.

**14. Future Enhancements**

* **Additional Features:**
  + Add workout video tutorials.
  + Integrate with wearable devices like Fitbit
* **Improvements:**
  + Refine the goal-tracking algorithm to offer more personalized recommendations.
  + Improve the user interface for better accessibility.

Demo link:

<https://drive.google.com/file/d/1mMqMb41RtroiFbUQ-1ZfeYfWJZ6okSNb/view?usp=sharing>