**Project Overview**

The Workout Tracker is a React-based application designed to help users track their workouts, progress, and goals. The application features a responsive UI, dynamic content management, and an interactive chart for visualizing key fitness insights.

**Project Structure**

The project is structured in a modular way, with separate components for different sections and pages of the app. Here's an overview of the key files and directories:

A screenshot of a computer program

Description automatically generated

**Key Features**

* **Routing**: Handled by react-router-dom to navigate between different pages such as Home, Workouts, Progress, and Profile.
* **State Management**: Uses React Hooks (useState, useEffect) to manage local states across various components.
* **Responsive UI**: TailwindCSS is used for styling, ensuring the app is mobile-friendly and responsive across devices.
* **Doughnut Chart**: A custom chart displays key workout tracker insights such as "Completed Workouts", "Active Days", and "Goals Achieved".

**Routing**

* **Home Page** (/): Displays welcome message, a carousel of key features, and a FAQ accordion.
* **Workouts Page** (/workouts): Allows users to view and add workouts.
* **Progress Page** (/progress): Displays progress metrics through a doughnut chart.
* **Profile Page** (/profile): Allows users to view and edit their profile information.

**State Management**

State management is handled using React Hooks like useState and useEffect for managing dynamic data.

**Key Uses of**useState**:**

* Navbar Component: Manages the open/close state of the mobile hamburger menu.
* Workouts Component: Manages the state for the list of workouts and modal visibility for adding new workouts.
* Profile Component: Manages the state for profile information and toggles between viewing and editing modes.

**Key Uses of**useEffect**:**

* Carousel: Automatically slides through images every few seconds.
* Fetching Data: Is used for fetching workout data when the component mounts.

**Components**

1. **Navbar**

**Features:**

* + - Responsive design with a hamburger menu for mobile devices.
    - Uses useState to toggle the visibility of the mobile menu.
    - Implements navigation through react-router-dom.

1. **Workouts**

**Features**:

* + - Allows users to add new workouts through a modal.
    - Dynamically renders a list of workout cards.
    - Manages the workout list and modal visibility.

1. **AddWorkoutModal**

**Features**:

* + - A form that allows users to add new workouts, including multiple exercises.
    - Form validation and data submission handling.

1. **Carousel**

**Features**:

* + - Automatic sliding of images.
    - Manual navigation and dot indicators for image selection.
    - Fully responsive layout.

1. **Accordion**

**Features**:

* + - Expandable/collapsible sections.
    - Toggleable content to show more information.

1. **WorkoutCard**

**Features**:

* + - Displays a card for each workout with details like exercise name, sets, reps, and duration.

1. **AddCalorieModal**

**Features**:

* + - Modal for adding calorie information related to workouts.

1. **DoughnutChart**

**Features**:

* + - Visualizes key insights (e.g., "Completed Workouts", "Active Days") using a customizable doughnut chart powered by ApexCharts.

**Key React Hooks Used**

**useState**

* Manages local component-level states, such as toggling visibility and managing dynamic data.

**useEffect**

* Executes side effects like auto-sliding for the carousel or fetching data on mount.

**Styling**

The app is styled using **TailwindCSS**, which ensures the design is responsive and mobile-friendly. Tailwind provides utility classes for:

* Layout (e.g., grid, flex, p-4)
* Typography (e.g., text-xl, font-bold)
* Spacing (e.g., mt-4, mb-8)
* Responsiveness (e.g., sm:hidden, md:grid-cols-2)