App Idea: Train with Gov

Brief: Personalized training web application for my clients.

Features:

* Daily Workout Sheduler
* Exercise Tracker
* Motivational Messages
* Personalized Daily dietary advice

chosen features:

**Web Application Structure**

**1. Landing Page (Home)**

* **Purpose**: Introduce users to TrainWithGov and its features.
* **Content**:
  + Welcome message with a brief introduction.
  + Quick overview of the main features: Daily Workout Scheduler, Exercise Tracker, Motivational Messages, and Dietary Advice.
  + Call-to-action buttons (e.g., "Get Started" or "Join Now").
  + Motivational quote displayed here as a highlight.

**2. Daily Workout Scheduler**

* **Page Flow**:
  + User selects the workout day from a calendar or a dropdown.
  + Based on the selected day, the app pulls a predefined workout routine (using the ExerciseDB API) or lets the user customize the routine.
  + Display exercises for the day with details such as sets, reps, and rest times.
  + Include an option for users to log completed exercises (which can link to the Exercise Tracker).
* **UI Elements**:
  + Calendar or dropdown for selecting workout days.
  + List of exercises for the day with “Mark Complete” buttons.
  + Integration with ExerciseDB API for exercise suggestions.

**3. Exercise Tracker**

* **Page Flow**:
  + Shows the user’s progress for the week or month, with completed exercises and missed workouts.
  + Users can input data like repetitions, sets, or weight lifted, and the app calculates their total progress.
  + Display personal bests or streaks to keep the user motivated.
* **UI Elements**:
  + Progress bar showing completion for each day.
  + Input fields for tracking reps/sets and a history of past workouts.
  + Motivational badges or achievements for consistency.

**4. Motivational Messages**

* **Page Flow**:
  + Daily motivational messages displayed on the home page or after completing a workout.
  + The message can be random or linked to the user’s progress (e.g., after completing a tough workout, they get a motivational boost).
* **UI Elements**:
  + Message display area.
  + Button for “More Motivation,” where users can cycle through quotes or advice.

**5. Personalized Daily Dietary Advice**

* **Page Flow**:
  + Users input their fitness goals (e.g., weight loss, muscle gain, maintenance) and dietary preferences (e.g., vegetarian, keto).
  + The app generates daily meal suggestions based on the input, pulling from a diet API or using predefined meal plans.
  + Offer tips on portion control, hydration, and nutrient intake.
* **UI Elements**:
  + Dropdowns or radio buttons for selecting dietary preferences and fitness goals.
  + A display section for daily meal recommendations.
  + Option for users to log their meals or mark them as “Complete.”

**Application Flow**

1. **User Onboarding**:
   * New users are prompted to set their fitness goals and dietary preferences.
   * Users are directed to the daily workout scheduler to plan their week.
2. **Daily Routine**:
   * Users check their schedule and complete the daily workout.
   * Post-workout, users log their exercises in the tracker.
   * They receive a motivational message once their workout is logged.
3. **Meal Plan**:
   * Users view their personalized dietary advice based on their fitness goals and preferences.
   * Optionally, they can log meals for the day.
4. **Progress Review**:
   * At the end of the week, users can review their exercise tracker to see their progress and consistency.
   * Display a summary of completed workouts, streaks, and dietary compliance.

Pages:

1. HOME PAGE

• Hero Section:

A welcome message and motivational quote.

A prominent call-to-action form (e.g., name, email, age, gender, dietary goals (e.g. Weight gain/ loss, and fitness goals).

• About Section:

A personal introduction about you (Governor) and what TrainWithGov is about.

A fitness philosophy or personal message that explains your approach to fitness and how this app will help clients stay on track.

Js logic:

**Before:** 1. Check if there’s user information in the local storage. IF yes, take the user to the scheduler page.

Before the page loads, fetch motivation for today.

When the user submits the form, validate it, save the info in the localstorage and redirect the user to the scheduler page

1. Daily Workout Scheduler(scheduler.html):   
   • User Details:

The user's name and fitness goal (captured from the call-to-action form) with motivational quote added to it in the top section.

• Daily Workout Section:

Displays today’s day, with a brief on what body part to worked on for that day.

A calendar or dropdown to select and to change the workout day for a different content.

List of daily exercises displayed based on the selected day (version 1). Each exercise can be clicked on and it shows a modal which shows pictures or videos of workout, its description and steps on how to do it.

In future versions, allow users to customize or plan their workout routines.

• Progress Tracker:

A small section showing the user’s progress for the day (e.g., completed exercises).

Js logic:   
-When the page loads of even before the page loads Check if there’s user details in the local storage and populate the user fields. Else redirect the user to the homepage.

-if there is user information present, go ahead and get today’s date and day, based on the day, fetch the workouts for that day based on the body parts targeted for the day, and also fetch the motivation message for the day.

-When the dropdown day is changed, fetch new data and update the results container with the day and the workout plan for that day.

-When an exercise is clicked, display a modal showing details of that exercise and steps to perform.

1. Personalized Daily Dietary Advice (dietary-advice.html):  
   **Meal Suggestions**:  
   -Before the page loads Check if there’s user details in the local storage and populate the user fields. Else redirect the user to the homepage.

-Provide daily meal recommendations based on user dietary goals (weight loss, weight gain, maintenance).

- Consider using an API (like a dietary or nutrition API from RapidAPI) to generate personalized meal plans.