**🚦 Project Title: Smart Fitness Tracker Web App**

**🎯 Real-life Problem:**

People exercising outdoors (like jogging or cycling) often face challenges like:

* Losing track of how far or where they’ve gone.
* Internet disruptions affecting location tracking or activity saving.
* Lack of motivation due to lack of real-time feedback or stats.

**🌟 Solution:**

Build a **Smart Fitness Tracker** web app that:

* Tracks user’s **location** and **distance** in real-time.
* Warns users of **poor internet connectivity**.
* Pauses/resumes tracking if the user is inactive for a while (uses Intersection Observer).
* Shows real-time **route and stats visualization** using the **Canvas API**.

**✅ Web APIs Used:**

1. **Geolocation API**  
   👉 To get the user’s real-time location while jogging or cycling.
2. **Network Information API**  
   👉 Detect internet connection quality and alert the user when it's poor, so data isn’t lost.
3. **Intersection Observer API**  
   👉 Detect when the page/tab is not visible to pause tracking and auto-resume when user returns.
4. **Canvas API** (optional enhancement)  
   👉 To draw the user’s route on a custom map with speed, checkpoints, and distance visuals.

**🧠 How It Works:**

**🗺️ Live Location & Route Tracking**

* Use **Geolocation API** to constantly fetch latitude and longitude.
* Calculate distance using haversine formula.
* Draw user’s route on a canvas map with start/stop points.

**📶 Network Status Monitor**

* Use **Network Information API** to monitor connection type and quality.
* Show a warning like: “⚠️ Poor connection! Data may not be saved.”

**👀 Inactivity Pause/Resume**

* Use **Intersection Observer** to detect if the user switches tabs or minimizes the window.
* If the app is not visible for more than 30 seconds → **pause tracking**.
* Automatically **resume** when user returns.

**🛠️ Tech Stack**

* HTML + CSS + JS
* Optional: PWA for offline tracking
* Optional: LocalStorage to store session data

**📱 Bonus Features (Optional):**

* Canvas-based graphs: speed vs time, elevation (mock data).
* Voice notifications: “You've completed 1km in 7 minutes!”
* Dark mode / Night view for late joggers.

**🧩 Code Structure**

plaintext

CopyEdit

index.html --> UI layout

style.css --> Basic styling

app.js --> All JS logic

|\_\_ trackLocation() --> Geolocation

|\_\_ monitorNetwork() --> Network API

|\_\_ handleVisibility() --> Intersection Observer

|\_\_ drawRouteOnCanvas() --> Canvas API

**🔚 Outcome:**

You’ll create a **smart, responsive, offline-ready** web app that assists people during outdoor workouts, ensuring they’re always tracked accurately and informed about connectivity, even when switching tabs.

Would you like the complete code or a prototype UI for this project?

**You said:**

Assignment - Build any project solving a real-life problem using a minimum 3 of these web APIs https://developer.mozilla.org/en-US/docs/Web/API/Background\_Tasks\_API https://developer.mozilla.org/en-US/docs/Web/API/Canvas\_API https://developer.mozilla.org/en-US/docs/Web/API/Geolocation\_API

**ChatGPT said:**

Great! Here's a **complete assignment idea** using **three required Web APIs**:

**🚗 Project Title: Smart Parking Locator & Time Tracker**

**🎯 Real-life Problem:**

People in cities often:

* Struggle to **remember where they parked**.
* Forget how long the vehicle has been parked.
* Want a quick sketch/visual of the parking spot for future reference.

**🌟 Solution:**

Build a **Smart Parking Web App** that:

1. Uses **Geolocation API** to save your parking location.
2. Uses **Background Tasks API** to run a timer that reminds you after a fixed time (e.g., for paid parking).
3. Uses **Canvas API** to let users **sketch surroundings** (like drawing next to a shop or building) to remember the location better.

**✅ APIs Used:**

| **API** | **Purpose** |
| --- | --- |
| **Geolocation API** | Get and save exact parking location (latitude/longitude). |
| **Background Tasks API** | Run background timer to notify about time elapsed (useful for paid parking or reminders). |
| **Canvas API** | User can draw a map or sketch of the location or surroundings. |

**🖼️ App Features:**

1. **Mark Your Parking Location**
   * On button click, save user’s current location using Geolocation API.
   * Display it on a map or store it for reference.
2. **Start Background Parking Timer**
   * Starts a timer in the background (e.g., 2 hours).
   * Sends a notification when time is about to expire (simulate with logs if notifications are restricted).
3. **Sketch Your Spot**
   * Use Canvas API to allow user to draw hints like “near red gate” or “left of mall”.

**🛠️ Technologies:**

* HTML + CSS + JavaScript
* Uses browser APIs (no backend needed)
* PWA optional (for offline usage)

**🧩 File Structure:**

plaintext

CopyEdit

index.html --> UI layout

styles.css --> Styling

app.js --> JS logic

|\_\_ getLocation() --> Uses Geolocation API

|\_\_ startBackgroundTimer() --> Uses Background Tasks API

|\_\_ setupCanvasSketch() --> Uses Canvas API

a