QUASAR Coding Screener — Scrollable Multichannel Plot (EEG + ECG)

Goal

Build a plotting script that loads our CSV and shows a static time-series plot that users can scroll/pan/zoom to explore the data. Tools like Plotly (Python or JS) or similar interactive plotting libraries are ideal because they provide built-in range sliders, panning, and zoom.

Dataset

Download: EEG and ECG data_02_raw.csv

File notes the app should handle

- Lines starting with # are metadata (ignore them).
- The first non-comment row defines the column headers.
- Relevant signals include:
  - **Time** (seconds)
  - **EEG channels (μV):** e.g., Fz, Cz, P3, C3, F3, F4, C4, P4, Fp1, Fp2, T3, T4, T5, T6, O1, O2, F7, F8, A1, A2, Pz
  - **ECG channels (mV):**
    - X1:LEOG → Left ECG
    - X2:REOG → Right ECG
  - **Other columns:**
    - CM → Reference/Common Mode (not an ECG channel; large amplitude, used internally for referencing, include in plot)
    - X3: → not a real channel (ignore it)
    - Trigger, Time_Offset, ADC_Status, ADC_Sequence, Event, Comments (ignore these)

Scaling hint:

- ECG (X1, X2): thousands of μV (~mV).
- CM: also large amplitude but not ECG, just the reference. You may plot it separately if helpful, but treat it differently from the ECG traces.

  EEG channels: tens–hundreds of μV.

Your task

- Load the CSV (ignoring # lines).

- Plot multiple channels vs. Time.
- Make the plot scrollable/zoomable. (E.g., Plotly's range slider, pan/zoom, or equivalent.)
- Decide how to handle scaling so EEG and ECG signals remain interpretable (multiple y-axes, per-trace scaling, etc.).
- Optional: Choose any features you think improve usability — for example: channel selection, unit conversions (µV vs mV), normalization toggles, exporting figures.

Time guidance

- Spend **~2–4 hours** on this project.
- Prioritize features that make the app most usable for a reviewer.

What we'll look for

- **Usability**: easy to scroll, zoom, and inspect EEG vs ECG.
- **Scaling choices**: EEG (µV) remains visible alongside ECG (mV).
- **Code quality**: clear, readable, maintainable.
- **Judgment**: good feature selection given time budget.

Submission

1. **GitHub repo** link.
2. **README** with:
    - How to run (dependencies/commands)
    - Notes on design choices (esp. handling of EEG vs ECG scaling) and use of AI assistance to complete this project.
    - Document features you wanted to add but couldn't due to time in a short "Future Work" note in your README.
3. Screenshots or a short GIF showing scroll/zoom (optional but encouraged).