

Intern Assessment: Data Analytics Dashboard

Create a **data analytics dashboard** that visualizes key insights from a dataset simulating a real-world IoT sensor payload. The dataset (CSV ; separated) includes **timestamped water level measurements** for a well pump. Your goal is to **build a single-page dashboard** using basic web development tools (HTML, CSS, JavaScript), with a focus on **statistics**. You may also use another language, like Python, to analyze the payload, then display it on a webpage. Third-party packages like graphs, gauges are allowed.

We want to see how you can **analyze**, **structure**, and **visualize** meaningful information for an end user. Your final product should reflect not only your coding skills but also your ability to interpret and present data effectively.

MUST READ:

This **voluntary** assessment exists solely to evaluate your skills; the company will not use your submission in any product or for any commercial purpose. **Typical candidates spend 2-4 hours, however, you may spend as little or as much time as you want, and you will not be compensated.** Don't worry if you can't finish every tier—partial completions still give us valuable insight into your abilities and experience level.

What You'll Work With

A CSV file with multiple records per day representing the water level of a well pump. You may simulate a REST API to serve this data using tools like Postman.

Tier 1: Basic Data Visualization

- Parse and display the dataset in a well-organized dashboard layout.
- Dynamically generate **data cards** or components based on the statistics you plan to gather(example: Runtime: 10hrs, Current Water Level: 40ft).
- Show raw values clearly and cleanly (could be a line graph or a trending card that shows each value on the day it was.).

Tier 2: Data-Driven Visual Elements

- Display **basic statistics**, such as:
 - Average, max, and min water levels
 - Pump Runtime (the time from when the water level increased drastically, start to finish)

- Cycles: How many times did the pump run?

Tier 3: Interactivity & API Integration

- Host the dashboard on a free source (e.g., GitHub Pages, Vercel).
- Fetch data from a **REST API endpoint** (you may mock this with Postman).
- Ensure the dashboard responds dynamically to data changes, without manually updating your HTML/JS.

Analytics Bonus Features (Optional)

- Implement **time-series analysis** (e.g., simple trends).
- Show visuals: line charts, bar charts, or gauges.
- Include **anomaly detection** (e.g., unusually high/low values).

Tips

- Keep it to a **single-page app**. No need for navbars or routing.
- Prioritize **clarity, readability, and functionality**.
- Focus first on making the data visible and useful, then enhance with visuals and interaction.
- Seeing the raw values in a graph typically helps visualize data and what statistics can be found in it (I provided an example of what the raw data may look like).

Submission

- Share a **GitHub repository** with your code.
- Optionally include a **hosted live demo** link.

Good luck and have fun!

Raw Data Example:

