

BUILD A LIVE DASHBOARD WITH AI

A Starter Prompt + Example Workflow

Teresa Thompson | ProfessorTeresaThompson.com

OVERVIEW

I needed one place to house all the disparate files and information my volunteer organization had. Being a visual person, I thought a dashboard would be an easy way to have everything in one place - announcements, files, quick links, etc. I worked with ChatGPT to create this -bringing it in early to explain what I wanted, how it should look, and eventually growing it to include live data from our other systems. It turned out to be a cool and helpful resource for everyone in the organization. So I thought I'd share the process in case it might help other clubs, organization, and committees.

But it was a slow, iterative process. So grab your caffeine and patience. Let's get started.

What I Built

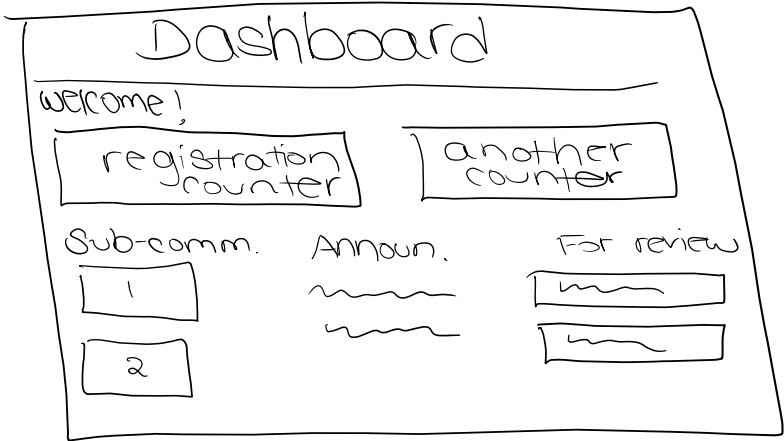
My example: a private TACTYC Board Dashboard using Google Sheets as the editable data layer, Google Apps Script to create the dashboard/web app, and Google Sites as the board-facing home. I later connected it with WildApricot, our registration system, through its API so selected conference metrics could update automatically.

Example workflow

1. Data Google Sheets + WildApricot	2. Logic Google Apps Script	3. Dashboard Apps Script web app	4. Home Google Sites
---	---------------------------------------	--	--------------------------------

What This Looked Like for Me

Start with the architecture. I began with by brainstorming what I visually wanted the dashboard to look like (see my doodle below).



Then shared this doodle, along with the data I wanted to convey -shareable files, announcements, live registration counter, etc. with ChatGPT. Then that LLM guided me through setting up the Google Sheets, connecting the Google Apps Script to that Google Sheet, and providing the html code for the Apps Script. And it walked me through any steps to format the Google Sites. Note, I did use Canva (a separate design system) for the graphics of the Dashboard -because I'm fancy like that. You obviously can pick from the pre-made Google Dashboard headers rather than design your own (unless you are fancy too:)

Add live data only after the basic dashboard works. Later, I connected WildApricot to Apps Script so conference metrics could be pulled into the dashboard automatically. But I started with just the informational pieces first -announcements, quick links, etc.

Iterate. I tested the code, reported what happened, shared errors or screenshots, clarified what I wanted, and had ChatGPT revise the code and help with any troubleshooting.

Starter Prompt

Copy, paste, and customize the [bracketed sections].

I want you to help me build a live organizational dashboard.

I am not an experienced programmer, so act as both my technical guide and coding assistant. Walk me through the project one step at a time rather than giving me everything at once. Explain what I need to do in plain language, provide complete code when code is required, and tell me exactly where to paste it.

MY GOAL: I want to create a private dashboard for my organization that displays useful information from our membership/event-management system. The finished dashboard should be visually polished, easy for nontechnical users to understand, and capable of being embedded in a Google Site.

DATA SOURCE: My organization uses [NAME OF SYSTEM]. It has an API that I would like to use so selected dashboard information updates automatically.

GOOGLE TOOLS AVAILABLE: I can use Google Sheets, Google Apps Script, and Google Sites.

WHAT I WANT THE DASHBOARD TO DISPLAY: [LIST THE INFORMATION YOU WANT TO SEE - for example: event registrations, sponsor counts, membership statistics, important links, announcements, committee information, etc.]

DESIGN: Organization: [NAME] | Primary color: [HEX CODE] | Secondary color: [HEX CODE] | Other design preferences: [DESCRIBE]

IMPORTANT REQUIREMENTS:

- Keep API credentials, tokens, client secrets, and other sensitive information secure and out of any HTML that users can view.
- Tell me when something should be stored in Google Apps Script Properties or another secure location.
- Before building the full dashboard, help me test the API connection with a very small request so we know the connection works.
- Once the connection works, help me identify the exact API endpoints, event IDs, registration types, or other identifiers needed for the information I want to display.
- Build and test one component at a time.
- When something doesn't work, help me troubleshoot the error before moving to the next step.
- When code changes, give me the complete replacement code rather than assuming I know which lines to modify.
- Explain when I need to redeploy the web app and when I can simply update the underlying data.
- Ultimately, help me publish the dashboard as a Google Apps Script web app and embed it into Google Sites.

Before we start coding, ask me the questions you need answered about my data source, the information I want displayed, and how I want the dashboard to look. Then recommend the simplest architecture for accomplishing this.

THE LESSON I WOULD SHARE WITH FACULTY

The first prompt didn't build my dashboard. The conversation built my dashboard. A strong starting prompt gives AI useful context and guardrails, but the real work comes from testing, asking follow-up questions, troubleshooting, and refining.

Tips Before You Start

- Start with the smallest version that would be useful.
- Do not paste passwords, API secrets, access tokens, or other sensitive credentials into a shared prompt or public resource.
- When an error appears, copy the exact error message back into the conversation and troubleshoot it before moving on.
- Keep the finished dashboard private if it contains internal organizational information.

My Finished Product



Welcome to the TACTYC Board Dashboard

One place for board meetings, committee work, upcoming events, and shared resources.

