

Building Your Dashboard with Claude on the Web — Step by Step

This is the zero-install route: everything happens in your browser at **claude.ai**. Best if you don't have (or don't want) developer tools on your computer.

Before you start — data safety

This dashboard will be **public**. That means:

- **Aggregate data only.** Every file you upload must contain counts, rates, and totals — never individual student records.
- **No personally identifiable information.** No names, student IDs, birthdates, or any file with one row per student.
- **Small counts stay hidden.** Any number based on fewer than 10 students should already be suppressed in your files before you upload.
- Not sure a file is safe? **Don't upload it.** Ask your institutional research office first.

Step zero — your data folder

Everything below assumes you have a `dashboard_data` folder holding your college's numbers in a specific set of files. Building that folder is its own small task, and this site gives you three things for it:

- **The data folder spec** (on the downloads page) — describes every file, column by column, and the rules the numbers must follow.
- **The sample data download** — a complete, filled-in example using synthetic data for a fictional college. Copy its shapes, replace its numbers.
- **Claude itself** — give it the spec plus your college's already-aggregated reports and ask it to help reshape them into these files.

One convention matters enough to repeat: suppressed counts are written in the files as the literal text `<10` . A blank cell means "not yet observable" — never "suppressed," and never zero.

What you'll need

1. A free account at claude.ai. (Free accounts can create up to five projects, which is plenty. A paid plan is still recommended — it allows longer working sessions and access to the most capable models.)
2. The prompt file: `build_var_dashboard.md` (the **Claude on the web** version, from this resource site).
3. The design guide: `dashboard_best_practices.md` (also from this site).
4. Your `dashboard_data` folder — see step zero above.
5. Your college seal or logo as an image file. **You'll attach this in the chat, not the project** — `claude.ai` projects only read text from project files, so an image placed there is invisible to Claude.

First, make the prompt yours (recommended)

The prompt was written for a college with a complete data folder. Your college isn't that college — so before building, have Claude tailor it. Start a chat, paste in the prompt file, and say something like:

"Adapt this dashboard prompt for my college: we're Mesa Vista College, and we don't have direct aid data or a narrative file yet. Keep everything else the same."

Save the result as your own `build_var_dashboard.md`. That's the version you'll use below.

Build steps

1. Go to claude.ai and sign in. In the sidebar, open **Projects** and click **+ New Project**. Name it something like "VAR dashboard." A project keeps your files loaded for every chat, so you upload everything once.
2. On the project page, add your files to the project's knowledge area (the **+** button): your `dashboard_data` files, the design guide, and the prompt file itself. A brand style guide PDF can go here too. (Remember: not the logo — images don't work as project files.)
3. Start a **new chat inside the project**, **attach your seal or logo image to the message**, and send one short line naming your college and its brand colors — hex codes if you know them. Example: *"Read build_var_dashboard.md in this project's files and build the dashboard it describes. Mesa Vista College — navy #1F3A5F, gold #E8B33D. Our logo is attached."* (No hex codes? Claude will take the colors from the attached logo.)
4. Claude reads the project files and starts building. The dashboard appears as a **live preview in a panel beside the chat** — no local server, nothing to install.
5. If Claude stops partway through, just reply **"continue."** It will pick up where it left off.
6. When the build finishes, click through every section in the preview (the pill buttons at the top switch sections). Try the year selector.
7. Want changes? Just ask in plain English, right in the chat: *"Make the header darker."* *"The equity labels overlap on the second chart — fix that."* The preview updates in place.
8. **Verify before you publish.** AI can make mistakes. Spot-check the numbers on the dashboard against your source files — the headline totals, one or two chart values per section, and the equity rates. If anything is off, tell Claude and have it correct the data.
9. When you're happy, click the artifact's **Download** button and save the file as `index.html`.
10. To put it on the web, follow the **Netlify hosting guide** — the drag-and-drop method is the natural fit for this route.

Tip: Claude's **Publish** button also creates a shareable link hosted on claude.ai. That's fine for showing a colleague a draft, but the same rule applies — publish nothing that isn't aggregate-only.

If something goes wrong

- **The build stopped mid-way.** Reply "continue." This is normal for a page this size.
- **The preview looks blank or stale.** Close and reopen the artifact, or ask Claude to show it again.

- **A number looks wrong.** Ask Claude to re-read the relevant file and re-check that figure against it. Always trust your source file over the screen.
- **The colors are wrong.** Make sure the logo was attached to a chat message (not added to the project files), or just tell Claude the hex codes directly.
- **The chat has gotten very long and slow.** Start a fresh chat inside the same project — your files are already loaded. Send the same kick-off message (re-attach the logo), plus a line describing the changes you had already settled on.
- **Claude built something fancy you didn't ask for** (extra charts, a different layout). Say so plainly:
"Remove that section — build only what the prompt describes."