

Building Your Dashboard with Claude in VS Code — Step by Step

This is the "real files on your computer" route. It takes a little setup, but you end up with a project folder you can rebuild, version, and update over time — the better fit if you expect to maintain the dashboard.

Before you start — data safety

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

- **Aggregate data only.** Every file in your data folder 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.
- Not sure a file is safe? **Don't include 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. This site gives you three things for building it:

- **The data folder spec** (on the downloads page) — every file, column by column, plus 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.

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. **VS Code** — free from code.visualstudio.com.
2. **The Claude Code extension** — installed from inside VS Code (step 3 below). You'll sign in with a Claude account; this route requires a paid plan — Pro or Max, or a seat on a Team or Enterprise plan if your college has one.
3. The prompt file: `build_var_dashboard_vscode.md` (from this resource site).
4. The design guide: `dashboard_best_practices.md` (also from this site).
5. Your college's aggregated data files in a folder named `dashboard_data` — see step zero above.

6. Optional but recommended: your college seal or logo as an image file.

Set up (one time)

1. Create a new folder on your computer — for example `my-dashboard` — and put these inside it: your `dashboard_data` folder, `dashboard_best_practices.md`, and the prompt file. If you have a seal or logo image, add a folder named `assets` and put it there.
2. Open VS Code → **File** → **Open Folder...** → choose `my-dashboard`.
3. Install the extension: click the **Extensions** icon in the left sidebar, search for **"Claude Code"** (published by Anthropic), click **Install**, then sign in with your Claude account when prompted.

First, make the prompt yours (recommended)

The prompt was written for a college with a complete data folder. Before building, ask Claude to tailor it. In the Claude Code panel, try:

"Read build_var_dashboard_vscode.md. Adapt it for my college: we're Mesa Vista College, and we don't have direct aid data yet. Save the adapted version over the original."

Then build from your adapted prompt below.

Build steps

1. Open the Claude Code panel (the Claude icon in the sidebar).
2. Open the prompt file, copy **everything below the horizontal line**, and paste it into the Claude chat. Send it.
3. Claude will read your files and start building. Along the way it will **ask permission** before running commands or writing files — read each request and approve it. (This is a feature: nothing runs without your OK.)
4. When the build finishes, Claude starts a small local preview server and gives you a link — usually `http://localhost:8000`. Open it in your browser. "Localhost" means the page is only visible on your computer — nothing is on the internet yet.
5. Click through every section and try the year selector.
6. Want changes? Ask in plain English in the Claude panel: *"Make the header darker."* *"Stack those two charts."* Claude edits the files and the same link updates.
7. **Verify before you publish.** AI can make mistakes. Spot-check the numbers on the dashboard against your source files — 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.
8. Your finished dashboard is a folder in your project (Claude will tell you its name — it contains `index.html`). That folder is what you publish next, using the **Netlify hosting guide** — either method works, and the GitHub method pairs especially well with VS Code.

If something goes wrong

- **Claude keeps asking permission for everything.** Normal and by design. Read and approve; it goes quickly.
- **The localhost link doesn't load.** Ask Claude: *"The preview link isn't working — restart the server."* If port 8000 is busy it will pick another and give you the new link.
- **A number looks wrong.** Ask Claude to re-read the source file and re-check that figure. Always trust your source file over the screen.
- **You closed VS Code and the preview died.** That's expected — localhost only runs while the server does. Reopen the folder and ask Claude to serve the site again.
- **You want to start over.** Ask Claude to delete the generated site folder and rebuild from the prompt — your data files are untouched.