

Prompt — Build a VAR Dashboard (Claude on the web)

How to use this at **claude.ai** — no installs, nothing on your computer:

⚠ **Before you upload anything: aggregate data only.** This builds a public-facing dashboard, so nothing you add to the project may contain personally identifiable information. Never upload student-level records — no names, student IDs, birthdates, or any file with one row per student. Counts based on fewer than 10 students should already be suppressed in your files. If you're not sure a file is safe, don't upload it.

1. **Make it yours (recommended).** Before building, paste this prompt into a chat and ask Claude to adapt it to your college — your name, your colors, and whichever data files you actually have. Build from the adapted version.
2. **Create a project.** At **claude.ai**, open **Projects** in the sidebar and click **+ New Project**. Name it something like "VAR dashboard." (A project keeps your files loaded for every chat, so you never re-upload them.)
3. **Add your files to the project's knowledge area** (the + button on the project page):
 4. your college's aggregated data files — everything in your `dashboard_data` folder (the README, the .csv files, and the .json files);
 5. `dashboard_best_practices.md` (from this resource site);
 6. this prompt file itself;
 7. a brand style guide, if you have one as a PDF. **Not your logo:** project files are read as text only, so an image placed here is invisible to Claude. The logo goes in the next step.
8. **Start a new chat inside the project**, attach your college seal or logo image to the message, and name your college and brand colors, for 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 image.)
9. The dashboard appears as a **live preview in a panel beside the chat** — no local server needed. If Claude stops partway through building, just reply **"continue."** Ask for changes right in the chat; the preview updates in place.
10. When you're happy: use the artifact's **Download** button, save the file as `index.html`, and drag it onto Netlify. (Clicking **Publish** instead creates a public **claude.ai** link — the same aggregate-only rule applies.)

Build a public-facing **Vision Aligned Reporting (VAR) dashboard** for the college whose data is in this project's files. VAR is the California Community Colleges' framework for reporting student services, student outcomes, and equity against the system's Vision 2030 goals.

Before writing any code, read `dashboard_best_practices.md` in the project files and follow it strictly — it is the design authority for this project. Then read `README.md`, `metric_definitions.json`, and

`data_notes.json` to understand every data file and its caveats. The dashboard presents only what is in the project's data files — nothing else, and nothing from files the sections below don't name.

Identify the college from the project files (`data_notes.json` names it) or from the user's message. If you have neither, ask before building.

What to build

One **plain HTML artifact** — a single self-contained page, not a React app:

- Vanilla HTML/CSS/JS only. No CDNs, no external fonts, no chart libraries — hand-rolled SVG/CSS charts. The downloaded file must work on its own, opened from a file or dropped onto Netlify.
- **Inline all the data** as JavaScript constants at the top of the page, transcribed exactly from the project's CSV/JSON files. Parse the files carefully and double-check totals — never estimate, round, or invent a number.
- **Keep the code compact** so the whole page fits in one artifact: write a few small chart-rendering functions and loop over the data; do not hand-write every SVG element. If you run out of room, pause and continue in your next message — never silently drop a section.

Four sections behind a centered pill toggle, plus a footer on every view:

1. **Vision 2030: At a Glance** — KPI tiles for the four Vision 2030 outcomes: awards conferred, transfers to four-year, median units at completion (vs. the 60-unit minimum), and workforce — which has no local data and is an honest labeled placeholder pointing to the state's published figure. Timely completion (within 3 years) rides along as a sub-stat. Trends vs. prior year.
2. **Completion & Momentum** — awards by type and year, median units trend with the 60-unit reference line, transfers to four-year trend. The college's VAR narrative (from `narrative.json`) sits beside the charts it talks about — narrative next to numbers is the point of VAR, not an appendix.
3. **Student Services** — the heart of VAR: students served (headline number + by program), service contacts by category, direct aid. Contacts, dollars, and hours are different units — separate charts, labeled, never summed.
4. **Equity Gap Analysis** — outcome rates by student group shown as **lollipop charts**: each group a dot on a stem, measured against a vertical line marking the collegewide rate, switchable across the metrics in `equity_outcomes.csv`; and the participation panel from `equity_participation_share.csv` — per program, each group's share of students served vs. its share of the college, with the proportionality index and its plain-language verdict shown as a badge.

Footer (all views): four closed dropdowns — Methodology, Data Protection, Sources, Glossary of Terms — built from `data_notes.json` and `metric_definitions.json`, including any data-quality indicators and disclaimers those files carry.

A single academic-year selector scopes the tiles and per-year charts; trend charts always show all years with the selected year marked; equity pools cohorts and says so.

Hard rules

- **This dashboard will be public. If any project file looks like student-level data — one row per student, or names, IDs, birthdates, or other personally identifiable information — stop immediately, tell the user which file, and ask them to re-aggregate. Do not build from it.**
- Aggregates only, exactly as shipped — never re-derive or un-suppress.
- Suppressed cells (<10) render visibly (hatched, with an explanation on hover) — never silently dropped. Blank means "not yet observable," never 0.
- Branding: use the brand colors the user gave; if none were given, take them from the seal or logo image attached in the chat (or a style guide in the project files). Pick the darkest for chrome/headings and a warm one as the accent, then derive and validate chart colors from them per the best-practices doc. You cannot copy an attached image into the page, so build a simple circular monogram for the header that echoes the seal's colors — and tell the user they can swap in their real logo after downloading. No colors and no image? Use a restrained two-hue institutional palette and say so in your summary.
- Every value must be reachable without hovering (labels or a "view as table" option), and the page must hold up at phone width — the artifact preview panel is narrow, so it will be seen that way first.

Finish

Before calling it done, review your own output section by section — label collisions, lopsided cards, masked-cell rendering, an early data year — and fix what you find; the preview updates in place. Then, in a few plain sentences: summarize what you built and any data quirks you hit, and remind the user they can view it in the preview panel, download it as `index.html` to drag onto Netlify, or ask you for changes right here in the chat.