

The One-Prompt Build Kit

The 5 prompts behind a tool an agency would quote at \$100,000, built by one non-developer in 15 hours. Copy, adapt the bracketed parts, ship.

Building a real tool with AI is one strong founding prompt plus a review loop. You supply the expertise and the judgement. The AI supplies the research, the build, and the iteration. These five prompts are the skeleton of that loop. Swap the domain, keep the structure.

The golden rule: specificity beats cleverness. Say *what* you're building, *who* it's for, *how* it should be structured, and *where* the facts come from.

PROMPT 1 · THE FOUNDING PROMPT (VERBATIM)

Start the whole build in one message

USED VERBATIM TO START THE BUILD

I want to build an interactive reference tool that maps the recent legislative reform changes to NSW workers compensation. The tool should display changes clearly against the claim lifecycle, from injury notification through to claim closure, so users can see exactly where each change applies and what it means in practice. Users will include injury managers, RTW coordinators, employers, self-insurers, and claims managers. I want the tool to align with the icare and SIRA audit framework so practitioners can cross-reference their obligations against what auditors will be looking at. Each reform change should show what changed, when it took effect, who it applies to, and what the practical implication is at that stage of the claim. Make it visual, easy to navigate, and interactive. Get the information from current sources. Build it so users can filter by claim stage, stakeholder type, or reform area.

Why it worked: it named the *what* (an interactive tool mapping X to a lifecycle), the *who* (five real user types), the *structure* (the fields and filters), and the *source* ("current sources", which triggered live web research). Not clever. Specific.

YOUR TEMPLATE · FILL THE BRACKETS

I want to build an interactive [tool type] that maps [your subject] against [the structure it lives in]. Users will include [3 to 5 real user types]. Each [item] should show [field 1], [field 2], [field 3], and the practical implication. Align it with [the standard your field is judged against]. Make it visual and interactive, filterable by [dimension 1], [dimension 2], and [dimension 3]. Get the information from current sources.

PROMPT 2 · THE DESIGN BRIEF

Turn a working tool into a professional-looking one

Once the first version works, don't fiddle with colours ad hoc. Commission a brief.

COMMISSION A DESIGN DIRECTION

Act as a senior UX/UI designer reviewing this live tool as a formal audit. Assess visual hierarchy, typography, colour, spacing, and mobile layout against professional standards for a [corporate / consumer / clinical] audience. Recommend a neutral, professional design direction (palette, typography, layout rules) and produce a design brief I can implement in the next session.

In the real build this locked a corporate palette and IBM Plex Sans, and the brief became the spec for every later visual fix.

LEVEL UP THE LOOK · TWO DESIGN SKILLS I RUN

Install these as Claude Code skills so the build doesn't come out looking AI-generic. Run the design-brief prompt above with one of them active.

- **UI/UX Pro Max** — 161 colour palettes, 57 font pairings, layout and UX rules.
github.com/nextlevelbuilder/ui-ux-pro-max-skill
- **Impeccable** (by Paul Bakaus) — a design language plus anti-slop commands for your AI harness.
github.com/pbakaus/impeccable

PROMPT 3 · THE AUDIT (THE MOST VALUABLE ONE)

Make the AI attack its own work

The best prompts weren't build prompts. They were audits. This is where polish comes from.

POINT THE AI AT ITS OWN OUTPUT

Run a formal audit of this tool. Assume it is flawed and find the problems. Check content accuracy, mobile layout, accessibility (ARIA, keyboard nav, colour independence), date handling, and disclaimer adequacy. Return a numbered list of findings (F-01, F-02, ...), each with the issue, why it matters, and the fix. Then implement them.

Produced a 16-point findings list, from a broken footer to an illegible filter pill.

PROMPT 4 · THE RISK REVIEW

Specialist framing catches what generic review misses

REVIEW IN A SPECIFIC EXPERT'S VOICE

Review this tool as a [20-year media and publishing risk specialist / compliance lawyer / accessibility auditor]. Identify every [liability / compliance / safety] gap: missing disclaimers, absent liability limitations, no "official sources prevail" clause, unwarranted accuracy claims. List each gap and draft the exact remediating language.

Surfaced seven disclaimer gaps a generic "is this okay?" would never have found.

PROMPT 5 · THE INTEGRITY CHECK

Ship clean: verify sources and originality

TWO CHECKS BEFORE LAUNCH

Do two checks before this ships. First, a full link audit: test every external link and flag any that are dead or wrong, then replace them. Second, a plagiarism / originality review: compare all cited content against its sources and flag anything reproduced too closely, distinguishing genuine copying from unavoidable statutory language.

Caught one dead link, and confirmed one flagged phrase matched a source only because it was quoting the legislation itself.

The loop, in one line: founding prompt, build, review, one specific fix, repeat. You are the domain expert and the quality reviewer. The AI is the team.

A note on the count: the real build was 57 prompts across three sittings, done in a chat app on a phone. In Claude Code on a laptop it's very likely one session with a fraction of the back-and-forth. Read 57 as the hard-mode number. It still beat a \$100,000 agency quote by two days.



From the post: "A Development Agency Would Quote This at \$100,000. I Built It in 15 Hours."

Share this freely, keep the attribution. Building something on the side? [Subscribe, it's free.](#)