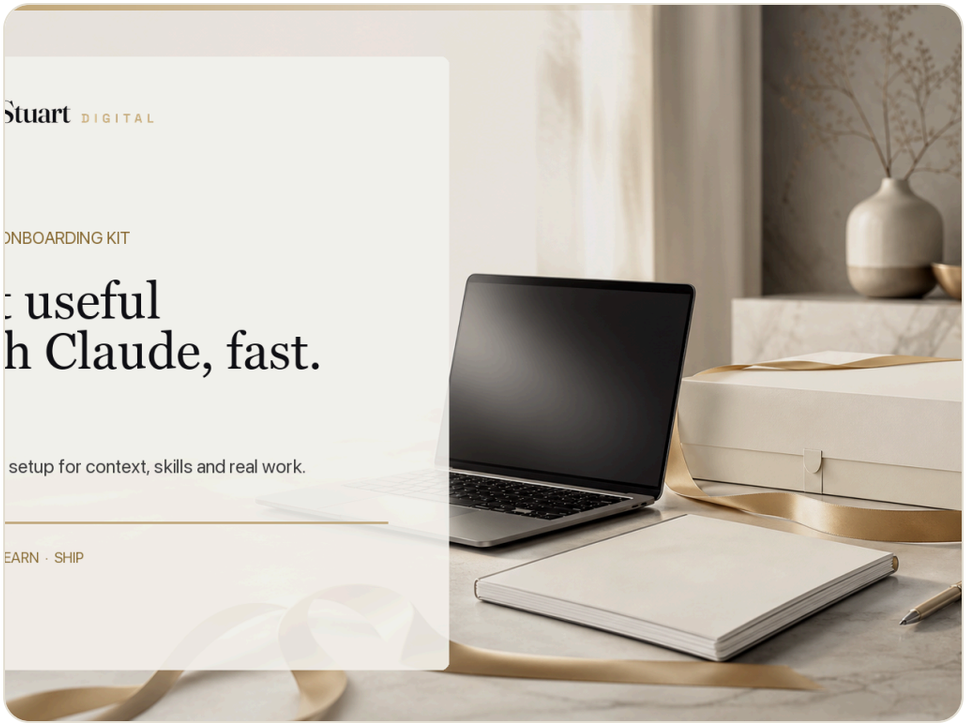
STAGE 5 · ADVANCED

Code.

The first session outside the chat window. One new tool, a plan you approve before anything happens, and a change you can read. You will not write code today — and you don't need to.



WHAT TODAY LOOKS LIKE

The 75 minutes.

01

Launch it

02

The plan and the diff

03

The fuel gauge

04

Inside a real app

This is the first Advanced session — the pair that shows you how real things get built. Same rules as every stage: you do the work, no coffee break, the hands-ons are the break.

WHERE YOU GOT TO

Your work has addresses now.

- **A dashboard changed in front of the room** at the same URL.
- **A questionnaire of your own** went live and started collecting answers.
- **The loop:** say the change, see it live.
- **Tools that run without you** keep the work current.

THE RHYTHM BREAK

One tool. Not five skills.

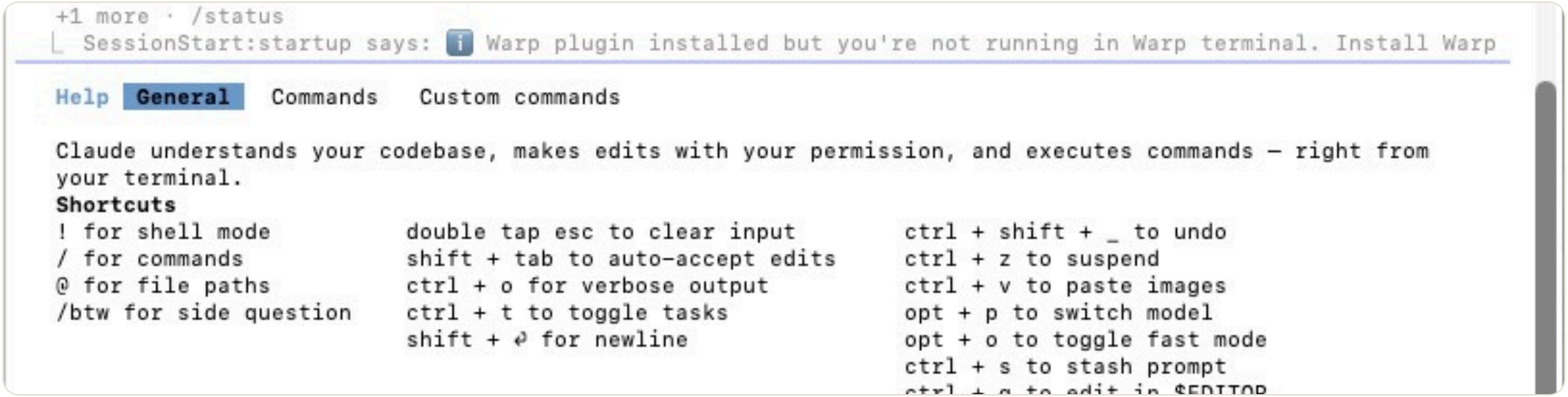
Every stage so far handed you a set of skills. **Stage five doesn't.** It hands you one tool and the habits to use it safely, because this tool is the door every later thing walks through — the dashboards you saw last stage were built with it.

> If you were counting skills: correct, there aren't five today. That's the point.

Out of the window. Into the terminal.

CLAUDE CODE

The same Claude, in a terminal.



Claude Code is Claude with its hands on the files — it reads a whole project, plans a change, makes it, and shows you exactly what it did. The window looks technical. The conversation is the same one you've been having since stage 1: plain English in, work out.

> The terminal is not the skill. Saying what you want clearly is the skill — and you've had four sessions of practice.

8 MINUTES · THE INSTALL CHECK

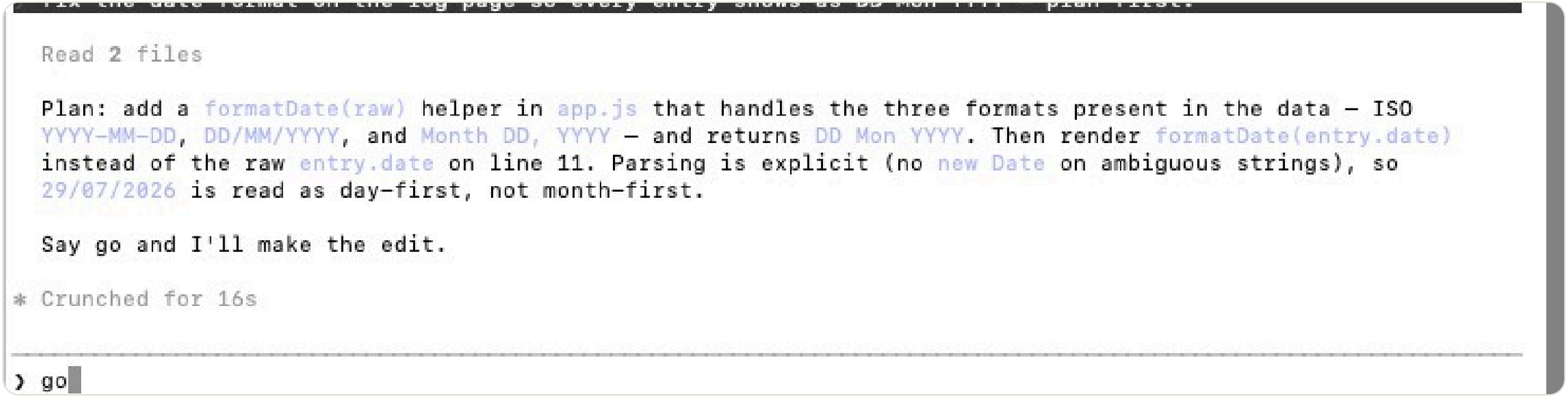
Launch it.

1. **Open the terminal, type `claude`.**
2. **Type `/help`.**
   Look at what it can do; don't memorise it.
3. **Type `/context`.**
   Remember the number you see. It matters in twenty minutes.

> Installed before today per the email? This takes ninety seconds. Didn't? Pair with a neighbour for the session — the install guide in your pack gets you running at home tonight, step by step, pictures first.

THE TRUST BEAT

The plan comes first.

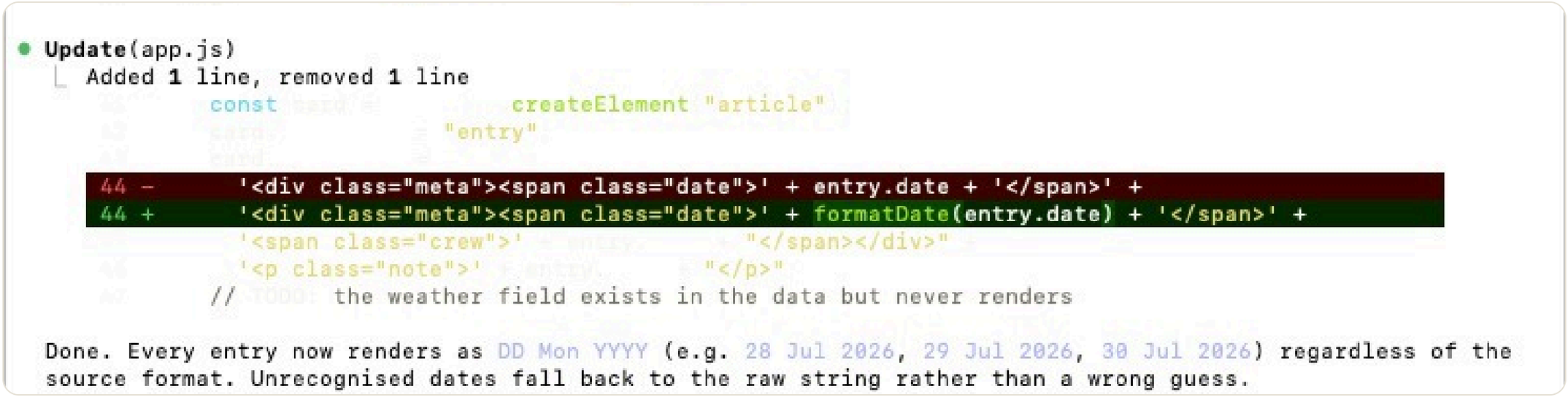


Say `add an export button to the reports page — plan first`. Before a single file changes, Claude Code lays out what it will touch, in what order, and why. You read it and approve it — or say no, and nothing happened.

> This answers “but what if it breaks something?” Nothing happens until you say yes. That's not a setting — it's the way of working you're learning today.

BEFORE AND AFTER

Read the change, then decide.



A diff is before-and-after with the changes marked: red went away, green arrived. This frame shows the edit already applied; your job here is to read what changed, ask if anything looks wrong, and approve it or push back.

> You review changes for a living. This is the same job with better highlighting.

12 MINUTES · THE THROWAWAY REPO

Your turn — and you can't break anything.

1. **In your practice project:** `claude`
2. `look at this project and tell me what it is`
   Watch it read.
3. **Pick a TODO, or say** `fix the date format on the log page — plan first`
4. **Read the plan. Approve it. Read the diff. Approve it.**
5. **Open the page and look at your change.**

> The repo is a throwaway built for this room. Break it with confidence — that's what it's for.

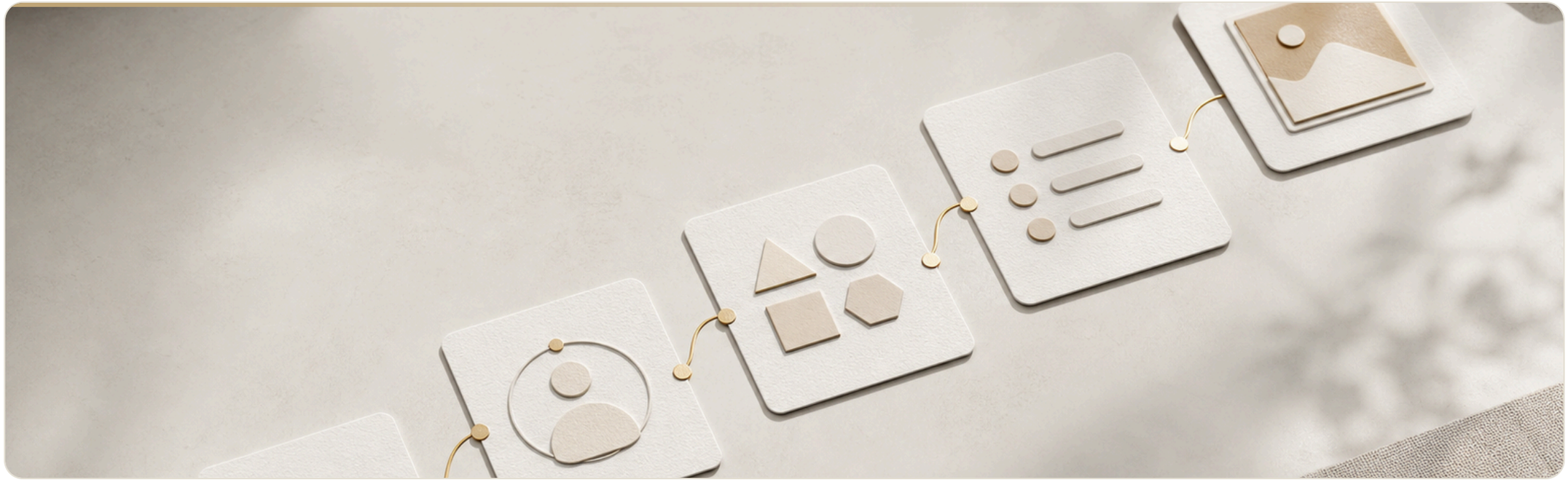
YOU ALREADY KNOW THIS IDEA

CLAUDE.md — your context skill, one level down.

In stage 1 you built a context skill so Claude knows who you are on every task. A project's `CLAUDE.md` is the same idea for a codebase: what this project is, how it's structured, what the rules are. Claude Code reads it every session, automatically.

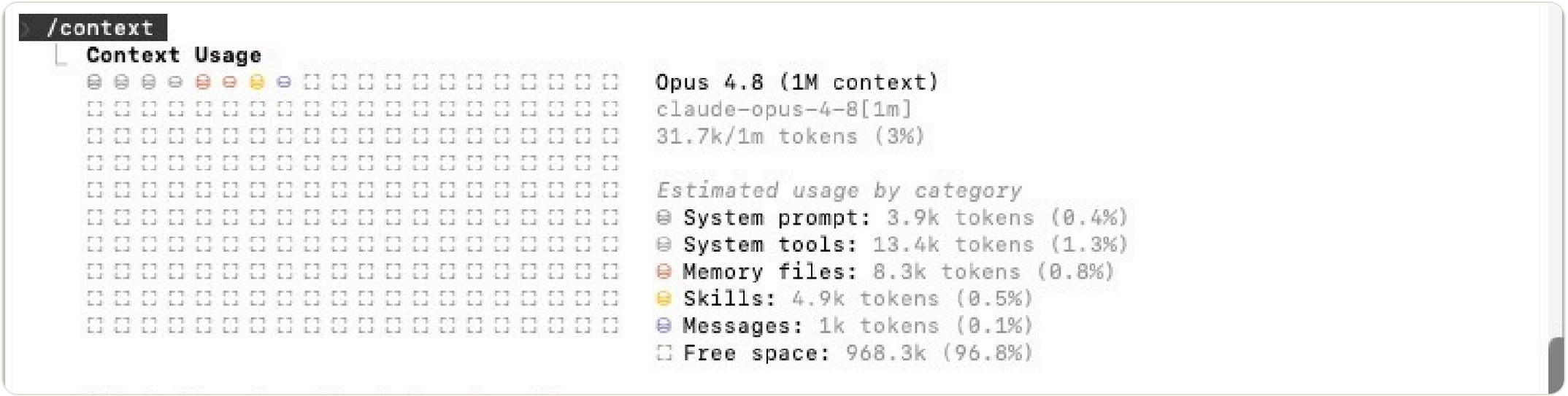
> Same lesson as ever: context in once, quality out every time after.

The fuel gauge.



/CONTEXT

Tens of thousands of tokens went before you typed.



A session starts with a chunk of its memory already spent — the tool's own instructions, your CLAUDE.md, the project map. The rest is your working room, and everything you do eats it. When it runs low, answers get worse before they stop.

> Same physics as “one topic, one chat” — you learned it in the chat window; here there's a gauge for it.

/COMPACT

Clear at task boundaries.

When one job ends and the next begins, `/compact` — Claude keeps a summary of what mattered and clears the rest. New room to think, same knowledge of the work.

> A task boundary is the moment you'd say “right — next”. That's when. Not mid-job.

LONGER JOBS

The handoff: summarise → clear → paste back.

For a job too big for one session: ask for a handoff summary of where things stand, start fresh, paste it back. Thirty seconds of ritual, and the new session picks up mid-stride. It's the same trick your own team uses when a shift changes.

NOT A DEMO REPO

Five minutes inside a shipped app.

1

The folder

Small enough to read. This is a whole shipped product.

2

Its CLAUDE.md

The project telling Claude how to work on it.

3

One question

`where does the renewal reminder get scheduled?`

> Everything you've been taught scales to this. Same tool, same habits, real store, real users.

CONSTRUCTION / CM ROUTE

Point it at a drawing set.

Say `analyse this drawing set`. A folder of construction drawings goes in; out comes a structured index — every sheet, every reference, cross-checks between them — that later questions answer instantly. One pass, then it's a resource.

> This one is a live demo today, not a download — it ships to your pack once its cleanup lands. The tool you installed is the takeaway of this stage.

WHAT YOU UNLOCKED

Six things, one tool.

Claude Code | plan mode | reading a diff | CLAUDE.md | /context + /compact | the handoff

Next: Shipping apps — the machine that builds the machines

> Stage 6 shows you the pipeline that turns this tool into shipped products — and it's the last rung of the ladder.

ONE REAL QUESTION THIS WEEK

Take the guide, keep the repo, and see you at stage 6.

Stage five's takeaway is the tool itself. No zip — your downloads are the guide and the practice project.

Start here · install guide | Practice project | Deck PDF

Scan to add Stuart to your contacts.

`training.stuartdigital.io/card`