Set up a context system for your AI

Three short files and one line of instructions, so you don't explain your work in every new chat.

9 minutes to read · about 20 minutes to set up

> **Works in ChatGPT and Claude.** The screenshots are from ChatGPT. Where the buttons differ, the step has two parts: **In ChatGPT** and **In Claude**. Everything else is the same in both.

The problem

You ask AI to draft a reply to a client. The answer is polite and ready to send. It also promises a day when you're away, for work that isn't free.

AI wasn't careless. It never saw your schedule or your terms. So you start pasting them in: the earlier messages, your prices, your days off. That works. Then you open a new chat for the next task and paste everything again.

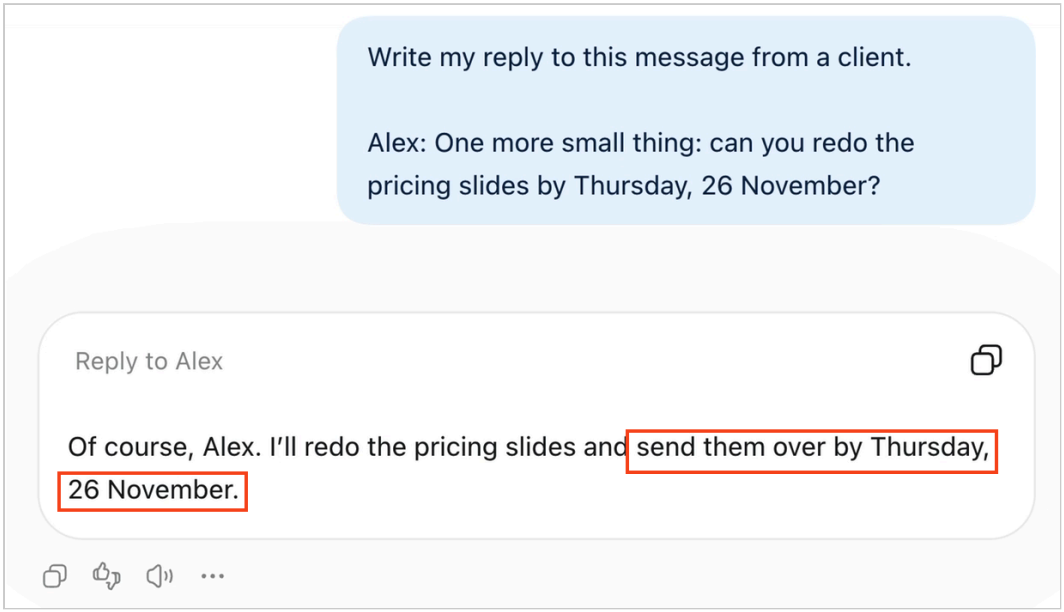
A context system solves this once. Context is the information AI has when it writes an answer. Instead of adding it to every chat, you keep it in one place AI can read: a project.

Here is the example I'll use. It's fictional, so I can show you everything.

> ILLUSTRATIVE EXAMPLE
>
> I'm a freelancer. I make presentations for small companies. A client, Alex, writes:
>
> **"One more small thing: can you redo the pricing slides by Thursday, 26 November?"**

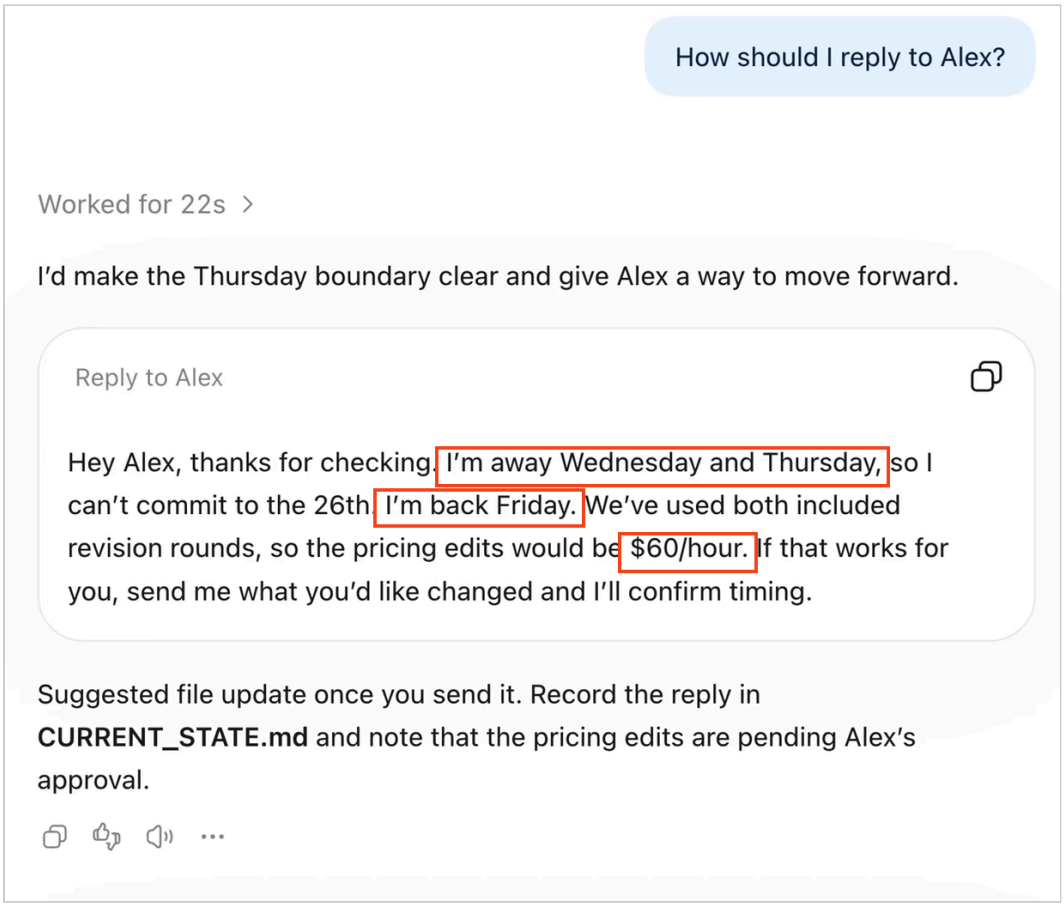
I asked for a reply twice. In the first chat, I pasted Alex's message. In the second, I only asked "How should I reply to Alex?" inside a project that holds my context.

Without context



A confident yes. I'm away that Thursday, and both included rounds of changes are already used.

With a context system



A short question. My days off and my terms are already in the draft.

What you'll set up

1 Three short text files about your work.

2 A project that holds them.

3 One line of instructions that keeps them up to date.

CONTEXT.md
FACTS AND GOALS

Your business
Clients
Prices and terms

CURRENT_STATE.md
WHAT IS HAPPENING NOW

Deadlines
Decisions
Days off

WORKING_RULES.md
HOW AI SHOULD WORK

Your tone
What AI can't promise

ONE LINE IN THE PROJECT INSTRUCTIONS

After each task, suggest updates to my files. Save them when I agree.

Step 1. Write three short files

Each file answers one question.

File	The question it answers	Lines from my example
CONTEXT.md	What should AI know about my work?	I work alone. Two rounds of changes are included. Extra changes cost $60 an hour.
CURRENT_STATE.md	What is happening right now?	Alex's presentation is due Monday, 30 November. Both rounds are used. I'm away on 25 and 26 November.
WORKING_RULES.md	How should AI work with me?	Client replies are short and warm. Never promise a date or a price that isn't in these files.

.md means Markdown: plain text with headings and lists. No coding. Any text editor works.

Three things make these files useful:

- **Keep each file to about one screen.** Short files are easier to check and to keep current.
- **Write real dates.** “Thursday, 26 November”, not “next Thursday”. AI reads the file later, on a different day.
- **Leave out what you wouldn’t put in a shared document.** No passwords, no bank details, no private information about other people.

The fastest way: let AI interview you

Open a new chat, in ChatGPT or in Claude, and paste this:

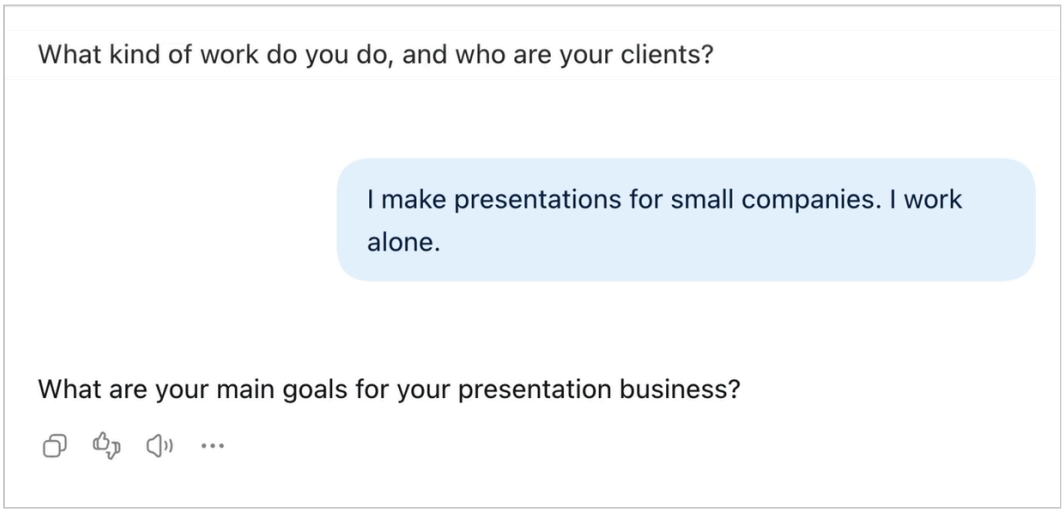
PROMPT

I want to set up three short files so you know my work: CONTEXT.md (facts and goals), CURRENT_STATE.md (what is happening now) and WORKING_RULES.md (how you should work with me).

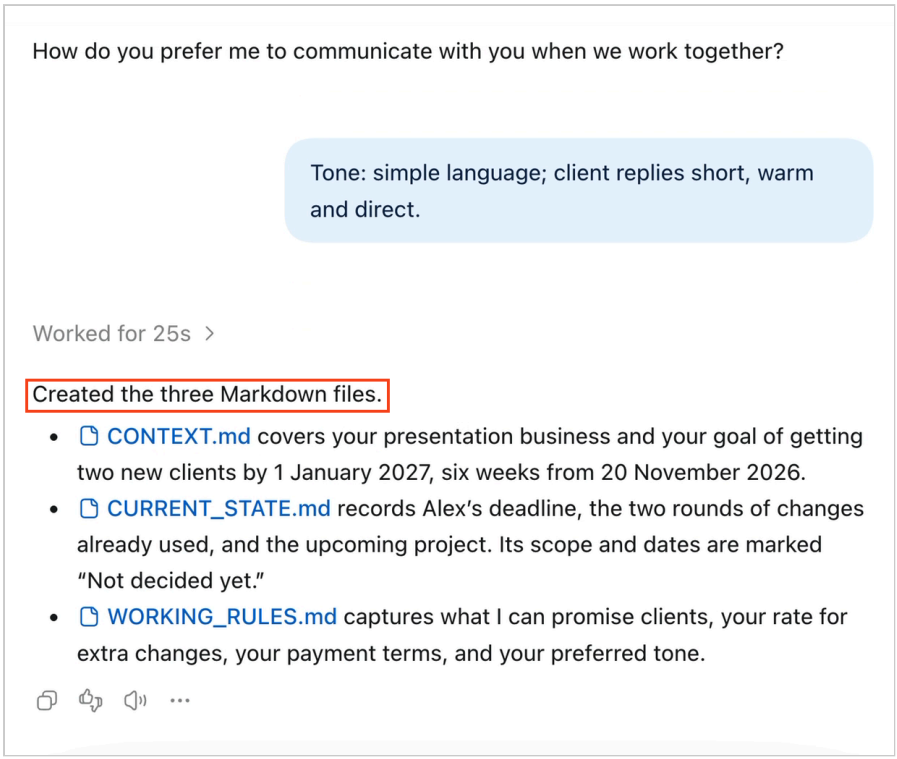
Interview me. Ask one question at a time, no more than 12 questions. Start with what I do and who my clients are.

Then write the three files in Markdown. Use only what I told you. Use real dates. Where something is missing, write "Not decided yet" instead of guessing.

Answer in your own words. You can dictate.



One question at a time. Short answers are enough.



After the last answer, it wrote the three files.

Then read every line. These are your facts, and AI can change them on the way. In my run, I said "two new clients in the next six weeks", and AI turned that into "two new clients by 1 January 2027". The date is correct, but I never gave it. Keep what is right, fix what is wrong, add what is missing.

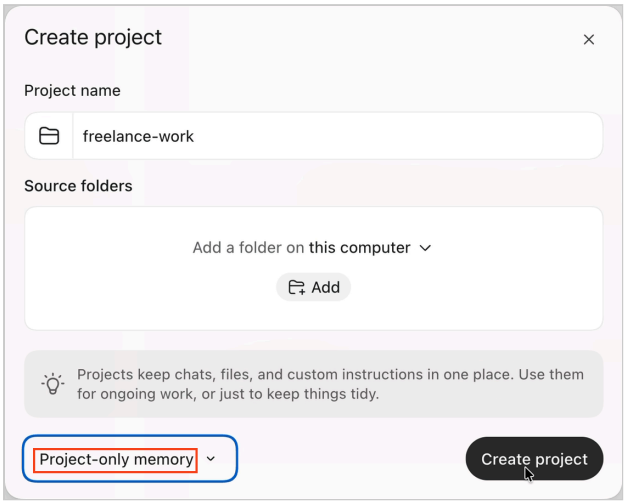
If your AI gives you text instead of files, paste each part into a plain text file and name it **CONTEXT.md**, **CURRENT_STATE.md** or **WORKING_RULES.md**.

Prefer to start from a template? The starter kit has empty versions of all three files and my filled example.

Step 2. Create a project and add the files

In ChatGPT

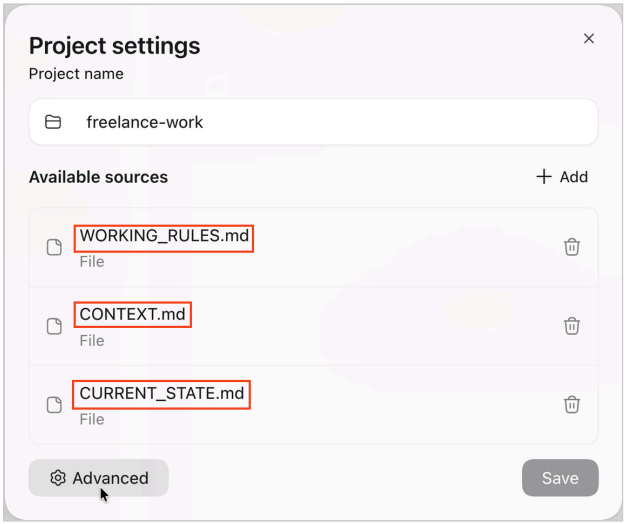
1. Next to **Projects**, click **+**. Give the project a name.
2. In the same window, change **Default memory** to **Project-only memory**. The project then keeps its own memory, hidden from your other chats. You choose this when you create the project. ChatGPT says it can't be changed later.
3. Click **Create project**.



Name the project and choose Project-only memory.

4 Open the project's menu and choose **Project settings**.

5 Next to **Available sources**, click **Add** and select your three files.



One project, three files. Keep this window open for the next step.

In Claude

1 In the left sidebar, open **Projects** and click **+ New Project**. Give it a name.

2 On the project page, find the project knowledge panel on the right. Click **+** and upload your three files.

3 To keep this project separate from your other chats, turn off **Use account memory** under **Memory** on the project page.

Projects are available on the free Claude plan too, up to five projects.

Step 3. Add one line of instructions

This is the line:

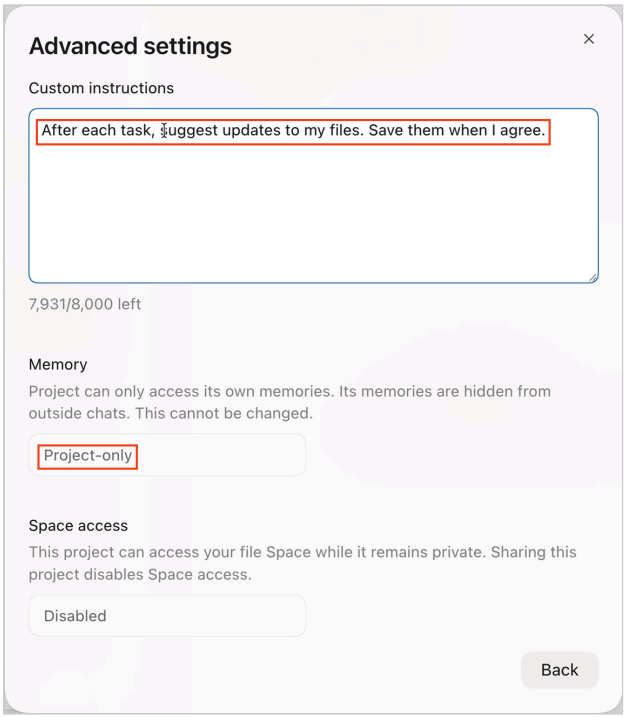
PROJECT INSTRUCTIONS

After each task, suggest updates to my files. Save them when I agree.

Your work changes every week. Without updates, the files get old, and so do the answers. This line makes updating part of every task: AI suggests the change, and you decide.

In ChatGPT

In **Project settings**, click **Advanced**. Paste the line into **Custom instructions**. Click **Back**, then **Save**.



The line in Custom instructions. Memory shows Project-only.

In Claude

On the project page, click **Set project instructions**. Paste the line and click **Save instructions**.

Step 4. Check, then ask the short question

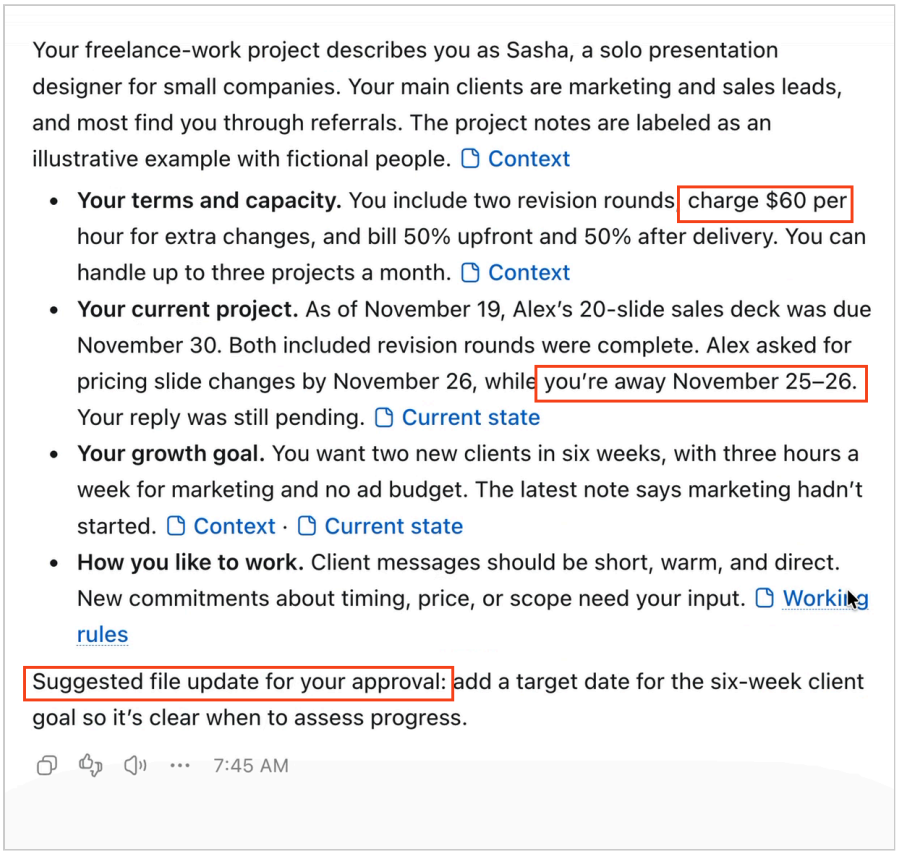
The rest is the same in ChatGPT and Claude.

First, check what AI knows

Start a new chat inside the project and ask:

PROMPT

What do you know about my work?



My terms, my deadlines and my days off, each with a link to the file it came from.

Compare the answer with your files. Check the facts, not the filenames: prices, dates, names. If a fact is wrong or missing, fix the file, not the chat.

Look at the last line in my screenshot. AI already suggests an update and waits for my approval. That is the line from Step 3 at work.

Then ask

Start another new chat in the project:

PROMPT

How should I reply to Alex?

No thread pasted. No terms. No schedule.

The draft is the second screenshot on page 2. It says I'm away Wednesday and Thursday, so I can't commit to the 26th. It says both included rounds are used, so the edits would be $60 an hour. It doesn't offer a new deadline. It asks Alex to send the changes first.

Check the answer before you use it. Two questions are enough:

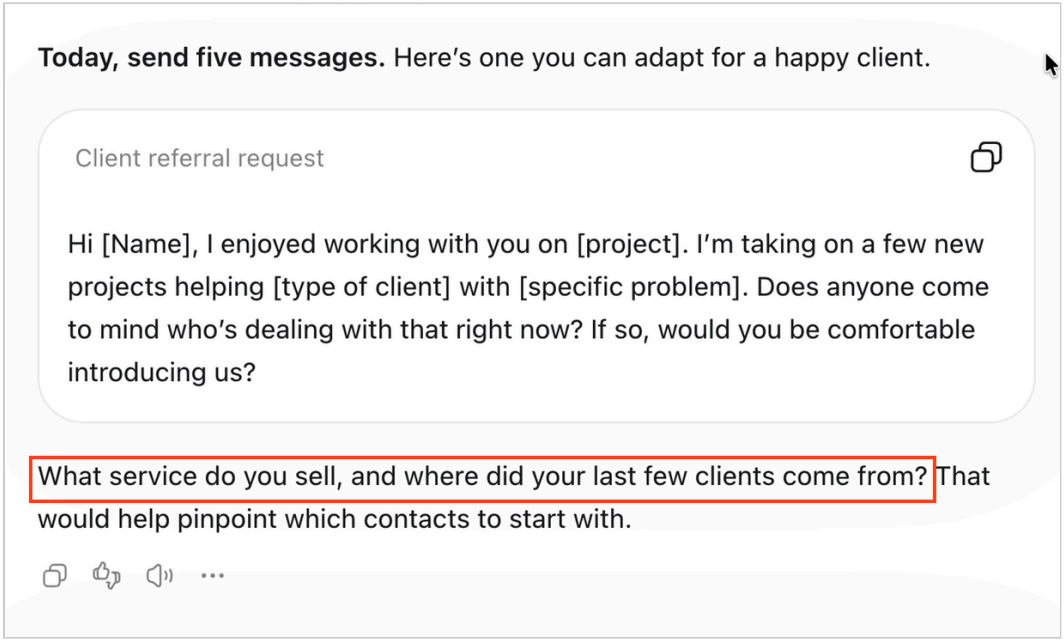
- Does every fact match my files?
- Did it promise a date, a price or an amount of work that isn't there?

A project doesn't make an answer correct. It gives AI what it needs. Checking is still your part.

The same files work for other tasks

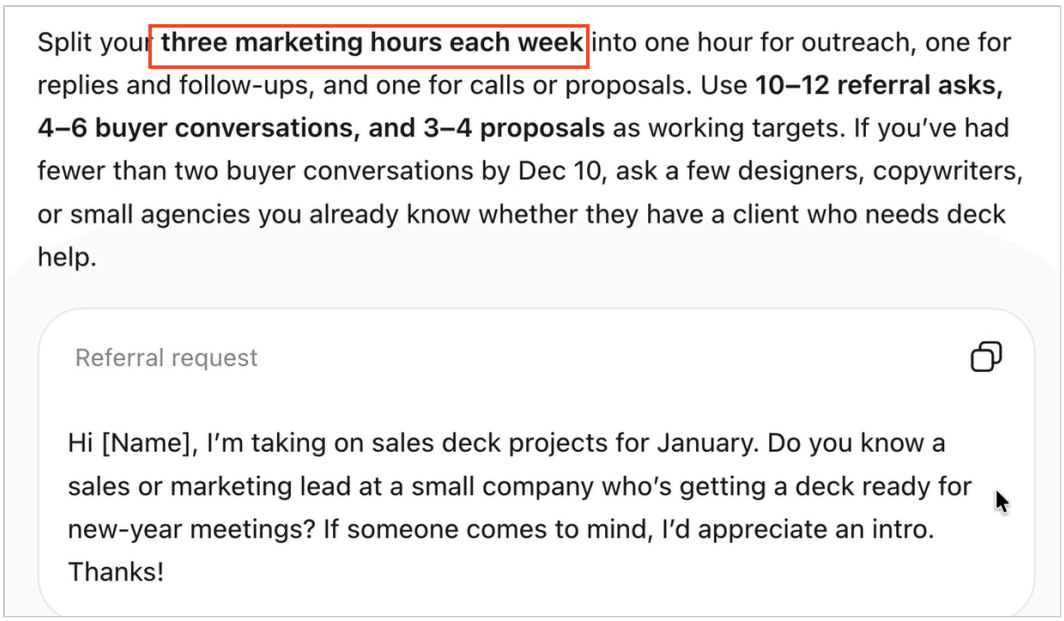
I asked a different question in two places: "How do I get more clients in the next six weeks?"

Outside the project



A general plan: contact 20 people a week. At the end, AI asked what I sell and where my clients come from.

Inside the project



It asked nothing. It split my three marketing hours a week into outreach, follow-ups and calls.

Inside the project, the plan also put the first three messages before my days off and kept Alex's deadline in view.

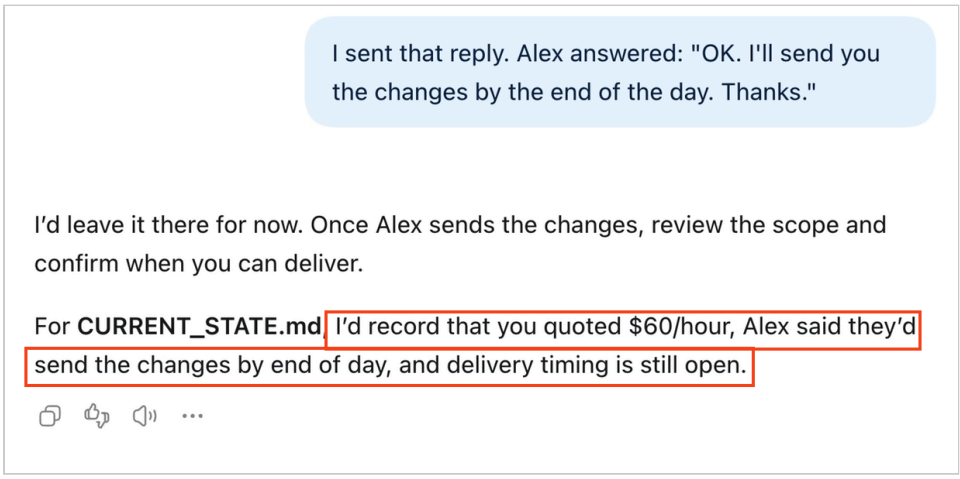
I wrote the context for a client email. It worked for a business question too.

Step 5. Keep it up to date

After a task, tell AI what happened:

PROMPT

I sent that reply. Alex answered: "OK. I'll send you the changes by the end of the day. Thanks."



AI says what it wants to record.

Read what it wants to record. Look for anything written down as agreed that isn't. In my example, Alex asked for Thursday and I never confirmed it. So the right note is "delivery timing is still open", and that is what AI proposed.

If it's right, agree:

PROMPT

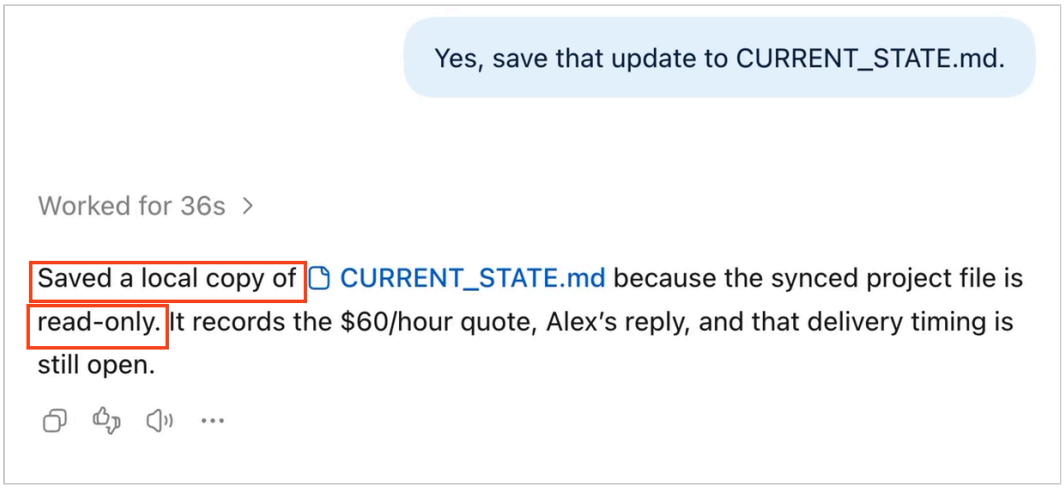
Yes, save that update to CURRENT_STATE.md.

Check where it saved

This part depends on your tool, so look before you trust it.

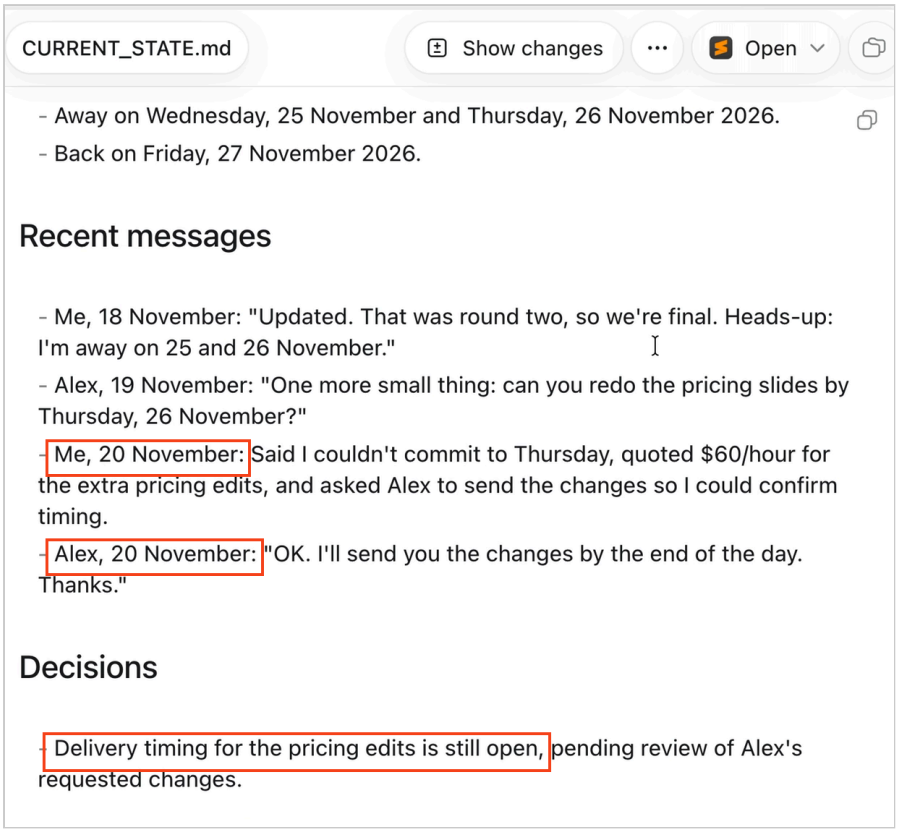
In ChatGPT

In my run, it could not change the file I had added to the project. It saved an updated copy instead and told me so.



A local copy, because the project file is read-only.

Open the copy and read it.



Two new lines under Recent messages, and one decision that is honestly still open.

If you keep the originals on your computer, replace them with the updated copy.

In Claude

Open the file in project knowledge after you agree. If it still shows the old text, ask Claude for the full updated file:

PROMPT

Show me the full updated CURRENT_STATE.md so I can replace the file.

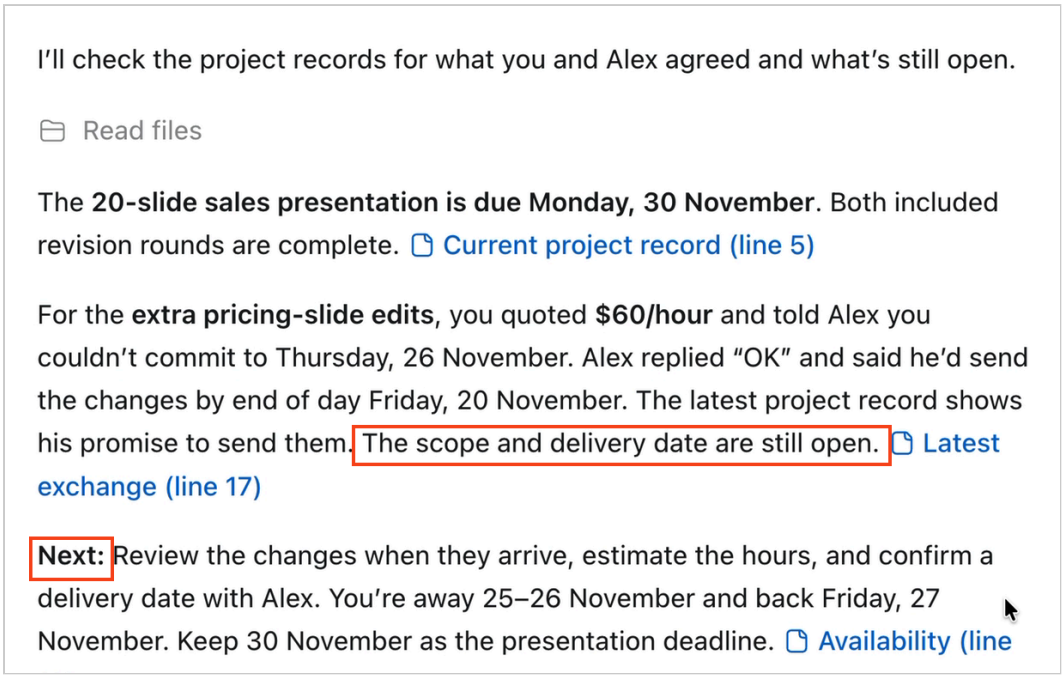
Then remove the old file from project knowledge and add the new one. Claude's help page says chats in a project don't share context unless it is in project knowledge. So the file is the reliable place for it.

Test the handover

In both tools, start a new chat in the project and ask:

PROMPT

What have we agreed with Alex, and what comes next?



A new chat, and it knows where things stand.

If the answer matches the update, your next chat can pick up the work where you left it.

If something goes wrong

What you see	What to do
AI doesn't suggest an update	Ask: "What should change in CURRENT_STATE.md? Show me the exact text before saving."
The answer ignores your files	Add a second line to the project instructions: "Before answering, read CONTEXT.md, CURRENT_STATE.md and WORKING_RULES.md."
AI promises a date or a price	Add this rule to WORKING_RULES.md: "Never promise a date, a price or an amount of work that is not in these files. Ask me."
A new chat doesn't know the update	The update didn't reach the project. Replace the file in the project with the updated version.
The facts are old	Look at the "Last updated" line in CURRENT_STATE.md. Update the file, then ask again.

When to skip this

For a quick question or a task you won't repeat, paste what's needed into the chat and move on.

A context system is worth the 20 minutes when you return to the same work again and again: the same clients, the same project, the same terms.

Checklist

- [ ] Three files, each about one screen, with real dates.
- [ ] A project with the three files added.
- [ ] One line in the project instructions.
- [ ] "What do you know about my work?" returns your facts.
- [ ] One real task done with a short question, and the answer checked.
- [ ] One update agreed, saved and opened.
- [ ] A new chat knows the update.

Your turn

Pick one real task this week. Set up the three files for it. Ask the short question. Check the answer against your facts. Agree one update for next time.

The starter kit comes with this guide: empty templates for the three files, my filled example and all the prompts.

Questions

If something isn't clear, write to me.

help@k-har.com

About the example. Alex and the project are fictional. The screenshots come from one run of my own in the ChatGPT desktop app on 20 November 2026. They are cropped, and the outlines are mine. Nothing else is changed. AI answers vary, so yours will be worded differently. That is why every step ends with a check.

About Claude. I haven't recorded this setup in Claude. The Claude steps follow Claude's own help pages as they were on 20 November 2026. Apps change, so if a button has moved, look for the same three things: the project, its files, its instructions.