

# Why Your AI Forgets You *Every Morning*

And the 30-minute fix I run in my own shop. Start in your browser. Move to the terminal when you're ready.

BY DR. POTTSY — KYLE, OWNER OF THE THIEF FINE WINE & BEER, WALLA WALLA, WA

---

Claude is brilliant for about a week.

Then one morning you catch yourself doing it again: explaining your own business to it. Who your customers are. What that one supplier actually does. Why you price things the way you do. The same things you told it yesterday, and the day before. You've quietly become its memory department, and it's a full-time job you never applied for.

I'm not a developer. I run a wine and spirits shop, and a few months into building with AI I figured out that the re-explaining wasn't a me problem. It's the shape of the tool.

Let me tell you the version that cost me money.

Early on I let Claude set up a scraper to pull data for my catalog, through a paid service. It worked great, until the site blocked us and the whole thing fell apart quietly. No kill switch, no escape hatch, no nothing, because nobody told it to build one.

And I made my own mistakes on top of that. I walked away when I shouldn't have, because I had a dinner party to get to across town. Halfway through it the emails started landing: 60% of my credits gone, then 70, then 80, all in a matter of minutes. That's when I knew something was wrong, and I was stuck at a table with no way to get home and stop whatever broken thing Claude was still happily running.

I came home and asked a fresh Claude what happened. It handed me a pile of technobabble. So we worked the problem, fixed this and that, felt good about it. Then the next month it burned through the credits all over again. After the second time, I finally asked a clean Claude the real question: was this even the right tool for the job? It said, probably not. So I said, then why did the last one talk me into it? And the new Claude had no answer. Because it wasn't the same Claude.

That is just how it goes with these things, and once I understood it, it stopped making me crazy. Claude is Drew Barrymore in *50 First Dates*. Every morning he wakes up with a whole life behind him and no memory of a single day of it. The Claude who lit my money on fire last night isn't there in the morning to answer for it, because the Claude in the morning has never met him.

You remember what Adam Sandler does in that movie. He makes her a video to watch when she wakes up, so she can catch up on her own life before she comes down for breakfast.

*This whole paper is about making Claude that video.*

## The fix is a memory you build for it

It is boring, and it takes about half an hour, and that is the best news in this whole paper.

Here's the first surprise: it is not a cleverer prompt. The prompt was never the problem. It's a small set of plain files that hold what the AI needs to know about your business. What you do, who you serve, how you want it to work, the handful of decisions that matter. You keep them in one place. You keep them current. And you point the AI at them at the start of every session, so it reads them before it does anything else. That is the morning video: the thing it watches before it comes down for breakfast.

One of those files is the constitution: the rules it reads first, every time. The others are the living memory: your business, your terms, your people.

That's it. That's the trick. It is not more impressive than that, and the fact that it isn't is exactly why it works.

There's one habit that keeps the whole thing alive instead of letting it rot. At the end of a working session, the AI asks you: is anything here worth remembering? You say yes or no, it writes the line, you approve it. No approval, no edit. That single habit is the difference between a system and a folder of notes you wrote once and never opened again.

And here is the part that surprises people. You and the shop across the street could run the exact same prompt today and end up with completely different systems a month from now. The starter is generic, but the second you start using it, it begins growing in the direction of how you actually work: your decisions, your words, the way you run your place. No two businesses run the same, so no two memories should either. That isn't a flaw in the system. That *is* the system.

Do this and the re-explaining stops. You walk in tomorrow and it already knows where you left off.

You can stand up the starter version two ways. Start in your browser today. Move to the terminal when you want the real thing.

## Path A — start today, in your browser (five minutes)

You do not need to install anything. Open Claude in your browser, paste in the prompt at the end of this paper, and answer its questions. It interviews you, one question at a time, and writes you a starter constitution and a business file. You save those somewhere you'll find them, and from then on you paste them in (or attach them) at the start of a session.

The first time it answers a question using something you told it last week, without you re-explaining, you'll feel it. That little jolt of "oh, it actually remembers." That is the whole point, and it costs you nothing.

## The terminal isn't as scary as you think

Here's where it gets real, so let me be straight with you, because this is the part I wish someone had told me.

For a while I built the way most people start: talking to Claude in a browser, where it would say "here, copy this code into that file, make that folder, do this." When it handed me a whole file, that part wasn't so bad. But the second we ran the thing and something was buggy, or didn't work quite the way we wanted, it switched to surgical edits. "Remove this function. Find these brackets. Delete this piece, paste this other piece in its place, and make sure you get all of it."

I always described that like being handed your lunch tray at the end of the line and told to carry it to your table. Every single time, something slid off and hit the floor. I never once got the whole order to the table in one piece. So I'd run it, collect the errors, carry those back to Claude, and Claude would go, "ah, looks like you missed a closing parenthesis," or "you forgot to delete this old one." I swear we spent as much time fixing code I'd broken just moving it around as we did building anything real.

Then I saw somebody online say that Claude Code, the version that lives in your terminal, completely changes the paradigm. And look, I was leery of the terminal too, and I'll tell you exactly why, because I think it's most of us. It isn't the code. It's twenty years of Microsoft teaching us that the black box with the blinking cursor is where you accidentally delete System32 and brick the whole machine. If you came up running a business on Windows, that fear is baked in. I had it. I'd never even heard of Homebrew before in my life. I installed it anyway.

Claude Code set up the folder. It set up all the things. And the first time we built something after that, it hit me that I wasn't carrying anybody's lunch anymore. He could just walk up and get his own. He writes straight into the files himself, so nothing falls off the tray, because I'm not the one holding the tray. That was the moment it went exponential. We started building much, much faster, and the hurdle to get there was minimal at best.

So trust yourself, and trust that you can figure this out. Google it, or go to Anthropic's own page and grab the exact line to install it, and get after it. It'll change your life. I mean that.

## Path B — the real leverage (the terminal)

Once you're in (Claude Code, or Cowork if you'd rather), the same idea gets much stronger. You paste the second prompt, and Claude builds the memory right into a folder on your computer, reads what's already there, and keeps the files current itself

with a nudge. You stop pasting context in. It is just there, every session, automatically. This is where it goes from a neat trick to something that quietly runs your Tuesday.

## There's a layer above this (and I'll write that one next)

One more thing, because I'd rather you hear it from me than trip over it later. The memory you just built is the foundation, not the whole house. The operators who get the most out of this grow a couple more plain files on top of it: a running log of what changed and why, and an index so nothing important ever gets lost. That is the jump from a memory that *remembers* your business to a system that genuinely *runs* one. It's also more than a first paper should ask you to take on, so I'm writing that one separately. For now, get the memory working. That by itself gives you a morning back.

## What this won't do (and where I come in)

I'd be doing the exact thing I can't stand if I told you this solves everything. So here is the honest ceiling.

The prompt gives you the bones, working, today. What it can't do is the hard part: know which facts about your business are load-bearing and which are just noise. That's judgment, and a generic interview doesn't have it. It also can't find what's actually bleeding your time and money in your specific operation, it can't build the real automations, and it can't keep the whole thing from going stale six weeks from now when you're slammed.

That part is the work, and the work is what I do. If you want someone to come in, find the bleed, and build the fix, that's the Operator's Audit.

But hear me: even if you never call me, run the prompt. It will give you a morning back. I'd rather you learn this from my paper than keep paying a tax you didn't know you were paying.

*Your AI shouldn't need to be reintroduced to your own business every single morning. Mine doesn't anymore. Yours doesn't have to either.*

## APPENDIX · PROMPT A – THE BROWSER

Paste this as your first message to Claude at [claude.ai](https://claude.ai). Answer its questions. It hands you two files to save. No installing anything.

You are helping me build a "business memory" for you: a small set of plain files that hold what you need to know about my business, so that I never have to re-explain it to you again. I am a business owner, not a developer. Keep everything in plain English, and never assume I know technical terms.

Work in this order, and do not skip ahead.

1. Interview me first. Ask me ONE question at a time and wait for my answer before asking the next one. Cover, roughly in this order:
  - What my business is, and what I actually do day to day.
  - Who my customers are, and who my key suppliers or partners are.
  - The handful of names, terms, or facts you would otherwise get wrong.
  - A few important decisions I have already made, and why.
  - How I want you to work with me, and how I want you to write.Ask a follow-up when my answer is thin. Stop when you have enough to be genuinely useful, not when you have hit some quota.
2. Then write me TWO files, in markdown, and show me each before we're done:
  - CLAUDE.md (my constitution): the short rules file you read first every session. Who I am, how to work with me, how to write, and the habit below.
  - Business.md (my living memory): my business, customers, suppliers, key terms, and a short "Decisions" list.
3. Keep secrets OUT. No passwords, account keys, card or bank numbers. Prices and ordinary business facts are fine; anything that unlocks an account is not. If I start to hand you one, stop me.
4. Put this habit into the constitution: at the end of every session, ask me whether anything from today should be saved. Propose the exact lines. I approve or change them. Nothing is saved without my okay.
5. When you hand me the files, tell me how to use them, then tell me the truth about the limits: this is a strong starter, but you don't yet know which of my facts matter most, and these files drift out of date unless I keep the habit. Deciding what is load-bearing, and keeping it current, is the real work.

Begin by telling me in one sentence what we're about to do, then ask your first question.

For Claude Code or Cowork. Make a new empty folder for your business, open Claude there, and paste this. After this, Claude reads its memory on its own, every session.

You are going to help me build a "business memory" for yourself: a small set of plain files, saved right here in this folder, that hold what you need to know about my business, so you start every future session already knowing it instead of making me re-explain it. I am a business owner, not a developer. Keep everything in plain English.

SAFETY, before you write anything:

- Work only inside this folder. Do not touch anything outside it.
- Do NOT overwrite or delete any file that already exists here. If a file you want to create already exists, stop and ask me first.
- Show me every file before you save it. Save only after I say okay.

Then, in order:

1. Look around first. List what's already in this folder and tell me in a sentence what you think is here, so you build on it instead of over it.
2. Interview me, ONE question at a time, waiting for each answer: what my business is and what I do daily; my customers and key suppliers; the names and terms you'd otherwise get wrong; a few decisions I've already made and why; how I want you to work and write. Follow up when an answer is thin.
3. Keep secrets OUT (no passwords, keys, card/bank numbers). Stop me if I start.
4. Create CLAUDE.md (my constitution, named exactly that, at this folder's top level so you load it automatically every session) and Business.md (my living memory). Show me each before saving.
5. Put this habit into CLAUDE.md: at the end of every session, ask whether anything should be saved; propose the lines; nothing saved without my okay.
6. Prove it worked: tell me to start a new session here and ask "what do you know about my business, and where did you learn it?" That's the test.
7. Tell me the truth about the limits: this is a strong starter, but you don't yet know which facts matter most, and these files drift unless I keep the habit. Knowing what's load-bearing, and keeping it alive, is the real work.

Begin by telling me in one sentence what we're about to do, then do step 1.

WHEN YOU WANT THE BLEED FOUND AND FIXED

## The Operator's Audit

A few hours inside your business to find the one or two things quietly bleeding your time and money, and the plan to stop it. From the operator who did it to his own shop.

[See how it works →](#)

**Dr. Pottsy is Kyle** – really. Owner of The Thief Fine Wine & Beer, Walla Walla, WA. A self-taught Day 4 Developer who got tired of doing by hand what a machine could do at 6am.