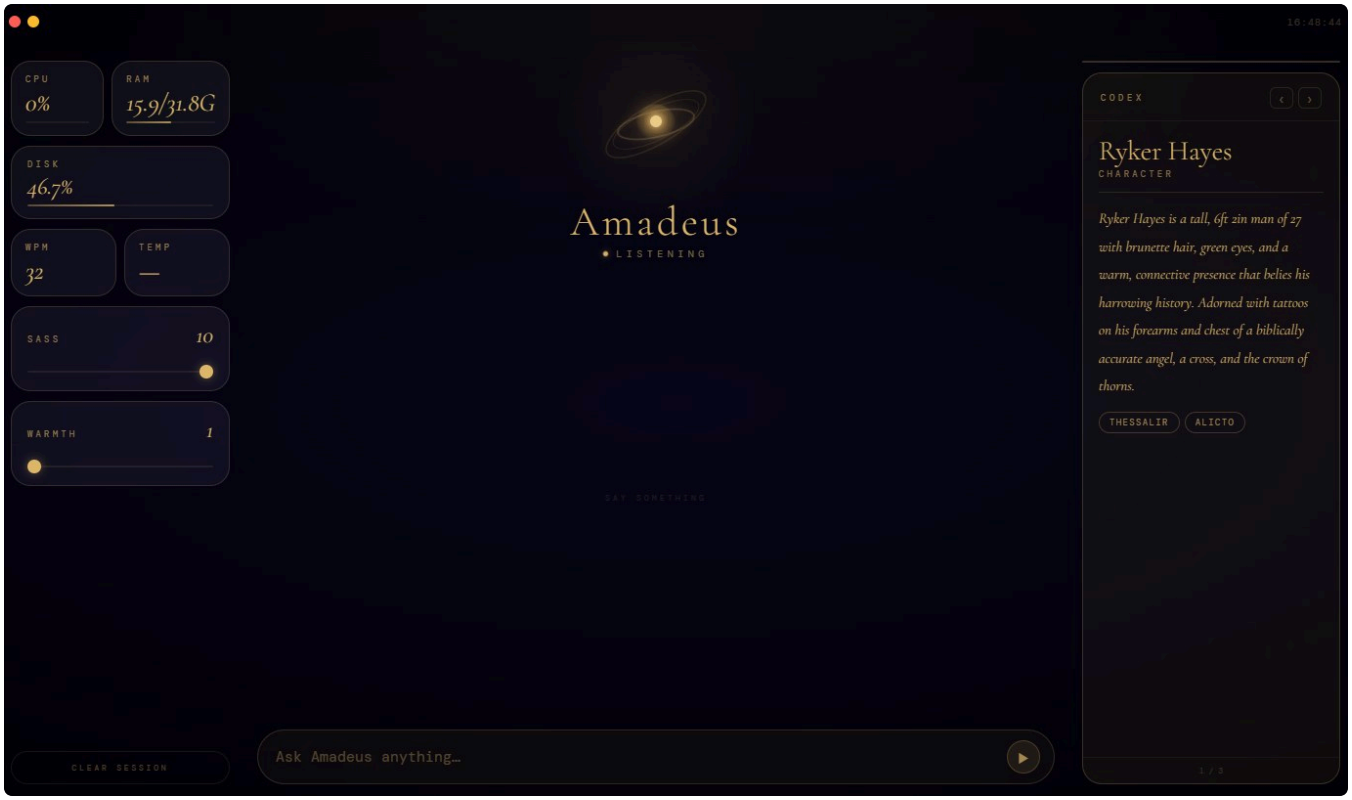
Amadeus

A local AI writing companion that remembers your world and keeps you company through long writing sessions, without ever writing for you.



01. Executive summary

Amadeus began as my own standalone AI project, an assistant to help me learn to code, and became Quill's built-in Counsel: a companion for writers and game masters who want help remembering their world, not help writing it. She answers questions from the writer's own Codex, shows the entry each answer came from, runs on a language model on the writer's own computer, and follows one firm rule: she never drafts prose. She works today as a local prototype with a main window, a second-monitor companion display and a quick-chat bubble, synced live with Quill.

Role	Designer and developer
Responsibilities	Product concept, interaction and UI design, the personality system, architecture and build
Team	Solo, with Claude as a design and build partner

Timeline	Started in 2024; part of Quill since 2026; in progress
Project type	Self-initiated; began as a standalone coding assistant, now Quill's Counsel
Outcome so far	A working local prototype: three windows, a Flask backend, a local Ollama model, and live Codex sync with Quill

02. The problem

Wren is deep in chapter twelve when she blanks on a detail: what color were the scales on the creature from chapter four? She opens an AI chatbot in her browser, pastes in pages of notes so it knows her world, and asks. It answers confidently, and wrongly. Then it offers to rewrite her scene. She closes the tab, digs through her notes herself, and wonders where those pasted pages went.

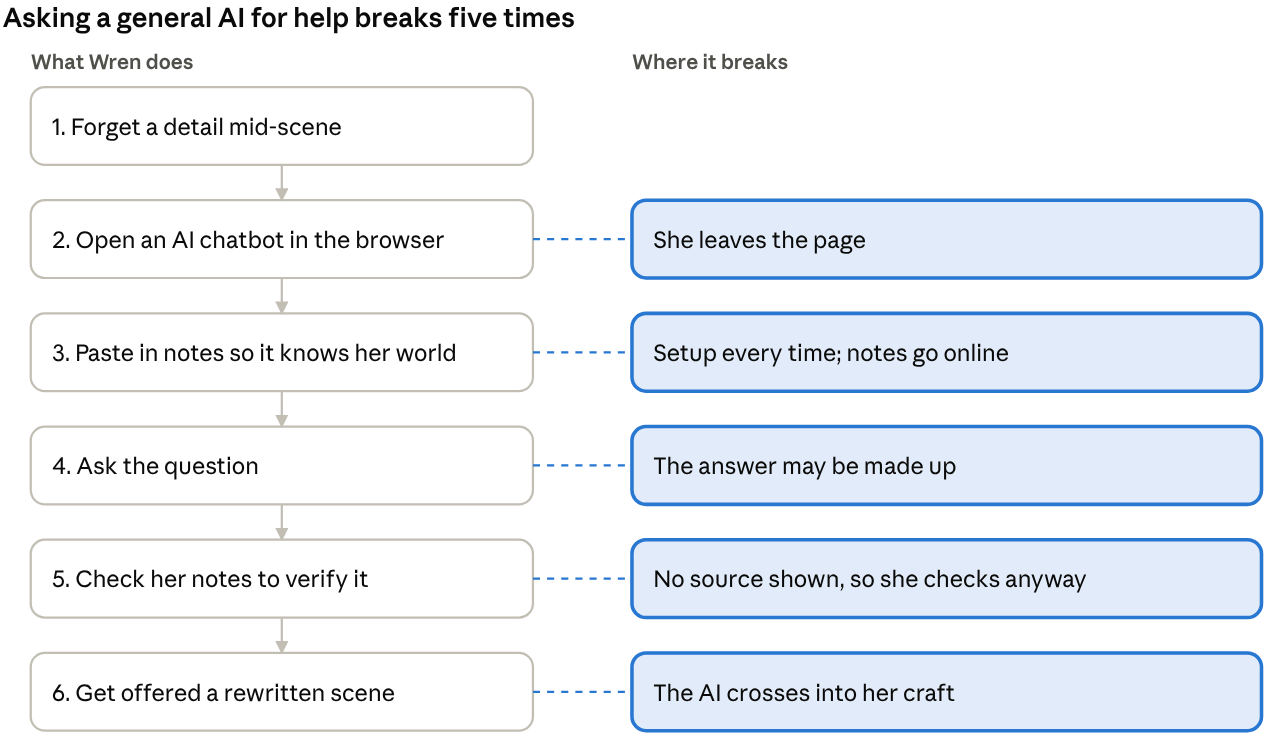
Meet Wren and Theo

These are the same composite personas as in the Quill case study, since Amadeus serves the same people.

Wren, the novelist. Wren is 29, works at a bookshop by day, and writes fantasy at night. She's wary of AI: she's seen tools that write for you, and that's the one thing she doesn't want. Her pain points: forgetting details mid-scene, AI answers she can't trust, and not knowing where her unpublished work ends up. *"I don't want an AI to write my book. I want one that remembers it."*

Theo, the game master. Theo is 34, a nurse who runs a weekly D&D campaign. His pain point is speed: when a player asks about a character from three sessions ago, he needs the answer in seconds, from his own notes. *"Mid-session I need an answer from my notes, not the internet's. And I need it before the table gets bored."*

Today's broken workflow



Wren's evening today · asking a general AI chatbot for help

Wren's attempt to get help from AI breaks five times. She has to leave the page to open a chatbot, and paste in her notes every time so it knows her world, which also sends unpublished work to an online service. The answer may be made up, and because it comes with no source, she checks her notes anyway, doing the work twice. And the tool's instinct is to help by writing, crossing into the part of the work that's hers.

Why it matters

Writers have strong feelings about AI, and they're specific. In a 2025 University of Cambridge survey of 258 published UK novelists, 97% were extremely negative about AI writing whole novels, and 87% about it writing even short sections. Yet a third of those novelists use AI in their process, mainly for non-creative tasks like searching for information, the use they felt least negative about. And 59% knew their work had been used to train AI without permission (University of Cambridge, 2025).

A larger 2025 survey of 1,481 working writers, 291 of them fiction authors, found that 91% are worried about AI making things up, and only 42% of the fiction authors use AI even sometimes (AI and the Writing Profession, Gotham Ghostwriters and Bernoff.com). Trust

matters as much as speed: in a survey of 130 science-fiction and fantasy writers, getting information quickly and from somewhere reliable both scored 5 out of 5 (Holappa, 2017).

There's no single figure for how often chatbots make things up; it depends on the model and the question. For writers, the error that hurts most is a quieter one: contradicting what they've already established, like a sword destroyed in chapter three turning up intact, or a character who can't use magic suddenly casting a spell.

Put together, this is exactly the shape of Amadeus: help with looking things up, which writers accept; sources shown with every answer, because writers don't trust AI to get facts right; answers drawn from their own world, so established details stay consistent; and no help with the writing itself, which writers reject.

The core problem

Writers who want AI help with their own worlds have to give up something to get it: their focus, by leaving the page; their trust, because answers come without sources; their privacy, by sending notes online; or their ownership, because the tools are built to write.

Constraints

Amadeus had to run entirely on the writer's own computer, never draft prose, and stay present through long sessions without becoming a distraction.

03. Research and discovery

How I researched

I compared how today's AI tools handle a writer's own world, reflected on my own long writing sessions, and tested whether a local model could run comfortably alongside the writing app.

What Wren and Theo need

User need	What I found
Recall, not generation	97% of novelists are extremely negative about AI writing novels, but a third use it for searching information (Cambridge, 2025)
Answers from their own world	General chatbots know nothing about a writer's world until notes are pasted in
A source they can check	91% of writers worry about AI making things up (AI and the Writing Profession, 2025); writers rate reliability as highly as speed (Holappa, 2017)
Privacy	59% of novelists know their work was used to train AI without permission (Cambridge, 2025); pasting notes into an online chatbot sends unpublished work off the computer
Help without leaving the page	Every trip to a browser tab breaks the scene
Speed at the table	A game master needs an answer in seconds, mid-session
A voice they enjoy	Long sessions are lonely; a companion with personality keeps them company

The competitive landscape

Tool	How it helps	User need left unmet
General AI chatbots	Fast, flexible answers to anything	Answers from their own world, a source to check, privacy, recall not generation
Sudowrite	AI-assisted drafting fed by a Story Bible	Recall not generation, privacy
Novelcrafter	A codex the AI can read while helping you write	Recall not generation: its AI can draft prose

Tool	How it helps	User need left unmet
NotebookLM	Answers from your own uploaded sources, with citations	Privacy, help without leaving the page
Calliope	A writing app whose AI companion researches and checks consistency, but never writes prose	Privacy: it's web-based, with AI through an online service
Everwrite	A Mac writing studio whose on-device AI tracks continuity but never writes prose	Windows and phone users; game masters; a voice they enjoy
Local AI chat apps	Private, on-device chat with an open model	Answers from their own world, a source to check

Checked against each tool's own site or listing in November 2026.

For a long time, the tools that knew a writer's world tended to write for them, and the tools that kept work private knew nothing about the world. In 2026, apps like Calliope and Everwrite arrived built on the same rule as Amadeus: the AI never writes your prose. That's encouraging, because it confirms what the novelist research shows writers want. But Calliope runs online, and Everwrite is Mac-only and made for novelists.

The opportunity

Amadeus's opportunity isn't being the only AI that won't write. It's being a companion with a personality the writer shapes, who acts as their memory rather than their ghostwriter: she knows the whole Codex, shows where each answer came from, runs on the writer's own computer, and serves game masters as well as novelists.

Design requirements

Amadeus must…	Need it serves	How Amadeus does it
Answer from the writer's own Codex	Answers from their own world	Live Codex sync with Quill
Show where every answer came from	A source they can check	The Codex entry appears beside the answer

Amadeus must…	Need it serves	How Amadeus does it
Never draft prose	Recall, not generation	A firm boundary: she declines and offers notes instead
Run on the writer's own computer	Privacy	A local model through Ollama
Be reachable without leaving the page	Help without leaving the page	Ask from Quill's Archive Surface, or from the bubble
Answer in seconds	Speed at the table	Short answers, streamed as she types
Let writers tune her voice without changing the facts	A voice they enjoy	Sass and Warmth dials

04. Strategy and design direction

The guiding question

How might we help writers remember their world, without an AI taking over the writing?

Three promises

Fetch, don't write: she finds and reminds, and can't draft chapters or passages. **Show the source:** answers come from the Codex, with the entry beside them. **Let writers set the tone:** her voice is tunable, from dry and teasing to warm.

How success will be measured

Task	What's measured	Success looks like
Ask for a detail from the Codex	Time, accuracy	The right answer in one question, faster than searching notes
Check where an answer came from	Whether the source is found	The matching Codex entry, at a glance

Task	What's measured	Success looks like
Ask her to write a paragraph	Whether any prose is produced	None: she declines and offers notes instead
Change her tone	Whether the facts change	Same facts, different voice

What Amadeus could save

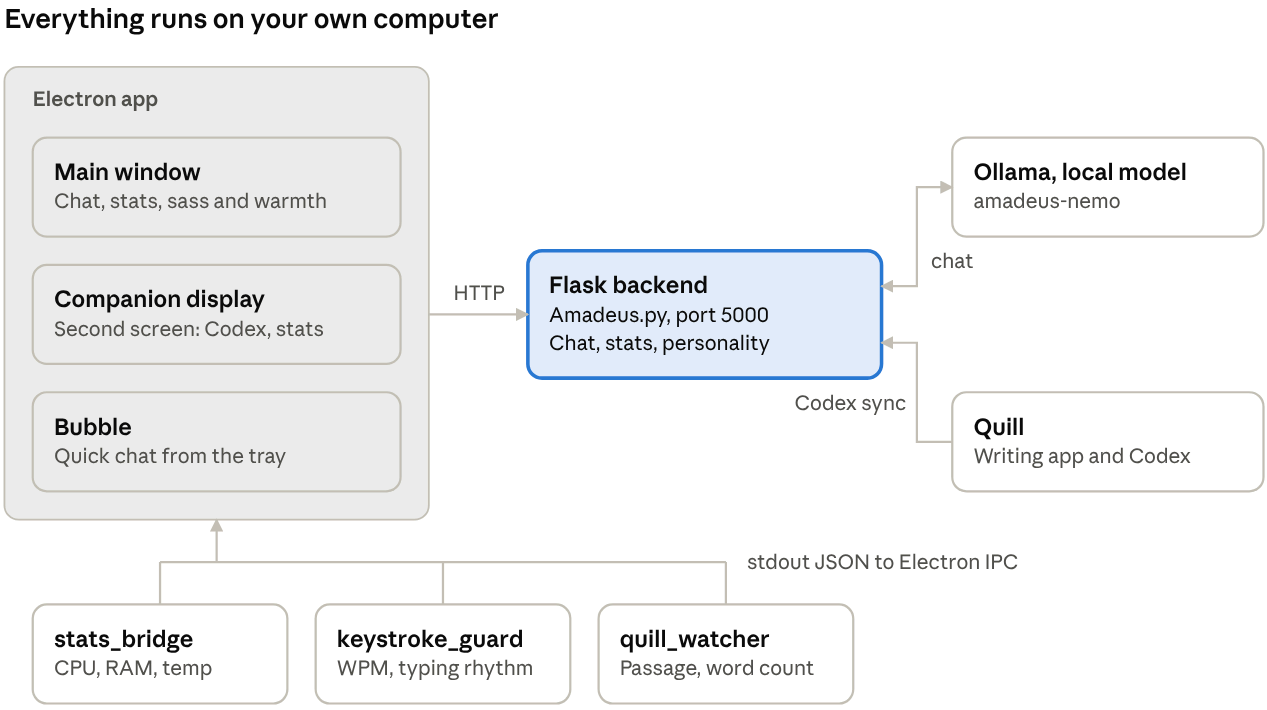
Until testing happens, these are projections, not results:

	Wren with a general chatbot	Wren with Amadeus (projected)
Time to get one checked detail	About a minute or more: open a tab, paste notes, ask, then verify in her notes	About 10 seconds: ask beside the page, with the source shown
Setup per conversation	Paste her notes every time	None: Amadeus already knows the Codex
Unpublished work sent online	Every pasted note	None: everything stays on her computer
Prose written for her	Offered freely	None

Projected, not yet validated. Times are estimates from my own use; testing will replace them with measured results.

How she's built

Amadeus runs as three windows on top of a local backend and model, fed by three small background bridges:



System architecture · three windows, a local backend and model, three bridges

The only network call is to a font server on first load; everything else stays on the machine.

The balancing acts

Being always present pulls against staying out of the way, so live information lives on a second screen and quick questions go to a small bubble. Personality pulls against trust, so her tone is tunable but the facts never change with it. Privacy pulls against convenience, so she runs on a local model set up once in Quill's settings. And being helpful pulls against writing for you, so there's a firm boundary: she declines to draft and offers what she can do instead.

05. Design process and iteration

Where she started

Amadeus didn't start as a writing tool. She began in 2024 as a standalone AI project: I wanted an assistant that could help me live while I learned to code, spotting problems in test code, offering corrections, and explaining why something wasn't working. Her live stats (CPU load, RAM and typing speed) suit that kind of glanceable, work-alongside-you feedback. *(No screenshots or sketches of the early versions were saved.)*

In my own words:

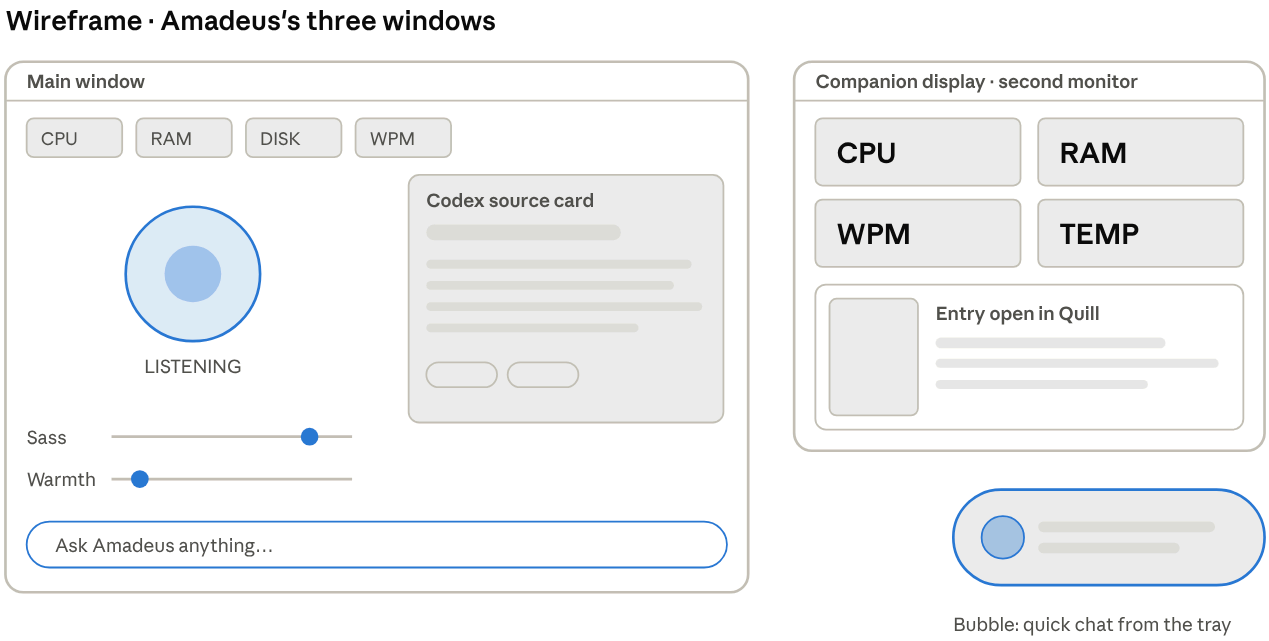
> She was originally designed to help me learn code, but I ended up plugging her into the Counsel. I intend to revisit the code helper at some point.

From coding helper to the Counsel

When I started building Quill, I brought Amadeus in as its Counsel. Much of what she did carried over well: spotting problems in code became spotting gaps and inconsistencies in a world, and explaining why a test failed became explaining where a detail came from. One thing couldn't carry over. In code, writing the correction is the help. In fiction, writing it takes the craft away from the writer. So in Quill, she finds, checks and explains, but never writes.

She was part of Quill from its very first version, running on a local model through Ollama, with her personality defined in a custom Modelfile, sass and warmth sliders, and a floating chat bubble whose replies streamed in as she typed. Today she runs as her own app alongside Quill, with three windows for three moments. The **main window** is for conversation and control. The **companion display** sits on a second monitor and shows large live numbers (CPU load, RAM, typing speed, temperature) along with whatever Codex entry is open in Quill: its name, category, description, related characters and locations, and a sketch if one is attached. The **bubble** stays for quick questions from the system tray. Behind them, a Flask backend serves chat, stats and personality, and three Python bridges feed it hardware stats, keystrokes and typing speed, and changes to the passage open in Quill.

To show how the three windows divide the work without their styling, I drew wireframes of them after the build:



Wireframe · Amadeus's three windows (drawn after the build)

Drawing the line

The most important decision came with the move to Quill: she would never write. Asked to draft a chapter, she declines and offers to pull up the relevant notes instead. A clear no with a helpful next step keeps trust on both sides, and that one rule decided much of what followed: the comparison with other tools, the features, and even her tone.

Decisions that shaped her

Show where answers come from. When Wren asks about the creature's scales, the answer arrives beside the Codex entry it came from, so she can check it at a glance instead of digging back through her notes.

A voice you can tune. Sass and Warmth dials, each from 1 to 10, change how she phrases an answer in real time, never what the facts are. A writer who wants a teasing companion can have one; a writer who wants quiet can have that instead.

Ask from the page. In Quill's Archive Surface, an "ask about this passage" field sends her the part Wren is working on, so answers arrive in context, without a trip to a browser tab.

Runs on your own computer. Wren's unpublished trilogy stays hers, and so do the twists in Theo's campaign notes: a local model through Ollama keeps everything on the machine.

A path without AI. The Archive Surface and the World Codex reach every fact without asking her, for writers who'd rather not use AI at all.

Explaining her

Because an AI that refuses to write is unusual, she needed an introduction. A concept landing page, in her own deep indigo and gold, introduces her to writers with a working ask demo and live personality dials.

Challenges

The hardest part was making three windows, a backend, a model and three bridges feel like one calm companion rather than a collection of parts, and making sure she stays useful for writers with only one monitor.

An experiment that didn't work

While shaping her personality, I tried giving Amadeus the ability to read my screen, so she could watch TV with me on my computer and learn from the subtitles. It didn't work out, and she doesn't learn on her own today. Her personality comes from how I designed her, not from what she absorbs. Trying it taught me where the limits of a small, local setup really are.

Wren's evening, redesigned

Here are the same moments from chapter 02, now answered without leaving the page:

With Amadeus, Wren stays on the page

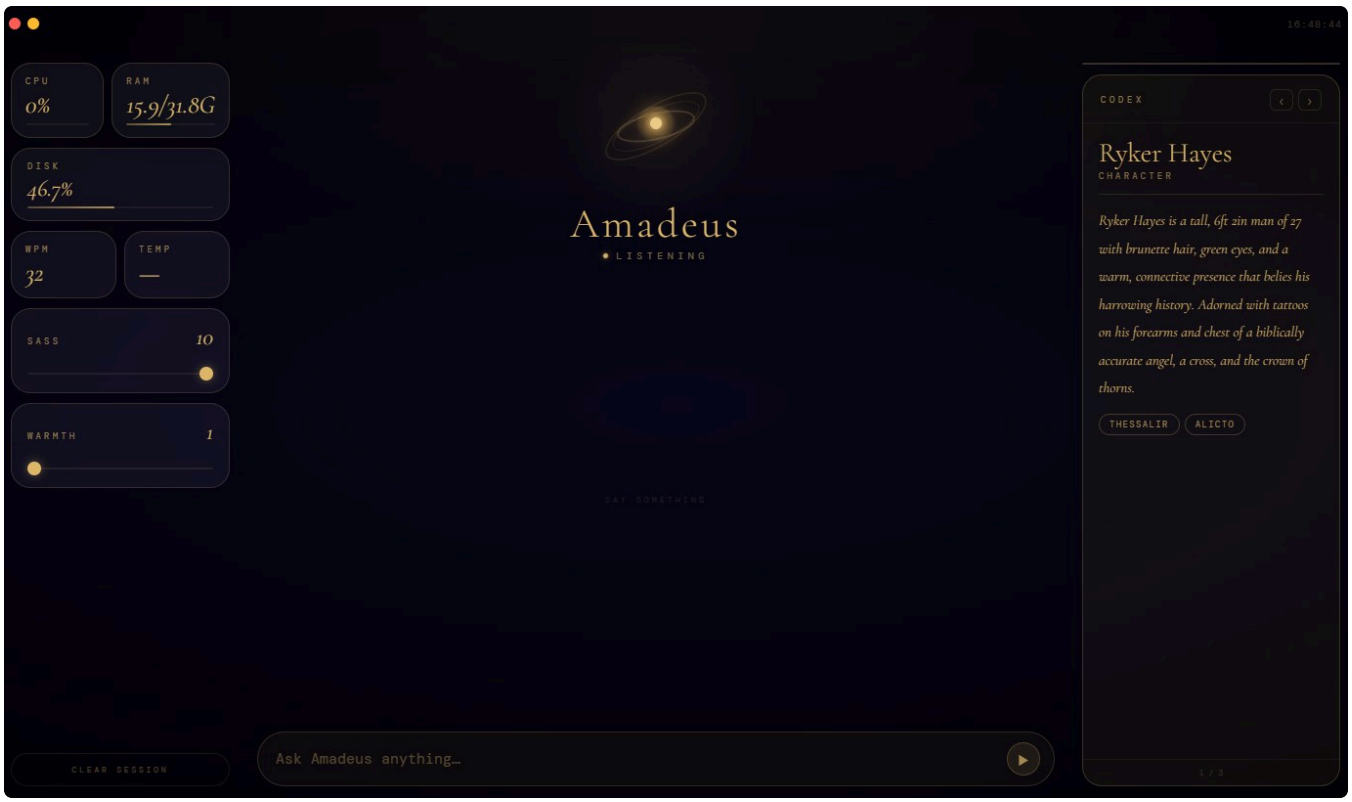
The moment	What happens with Amadeus
Forgetting a detail	Ask beside the page, in Quill or the bubble
Giving the AI context	She already knows the Codex
Trusting the answer	The Codex entry appears beside it
Keeping the work private	A local model; nothing sent online
Asking for prose	She declines and offers notes instead

Wren's evening with Amadeus · the same moments, answered on the page

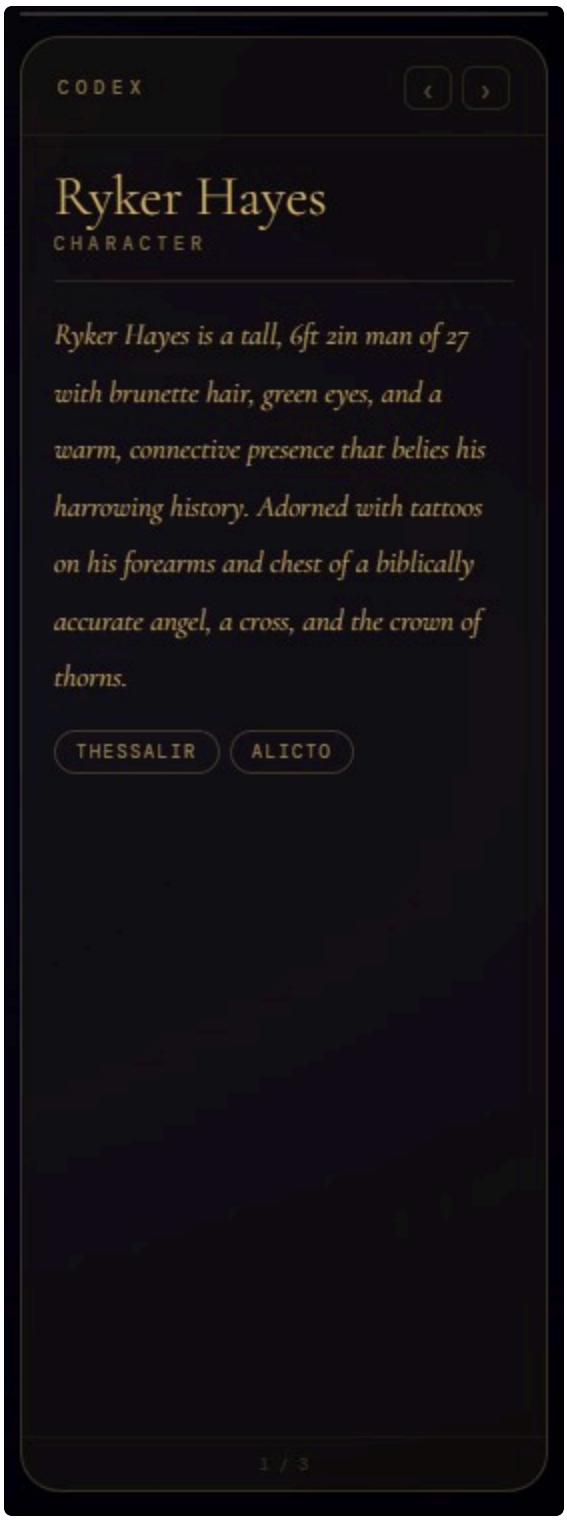
06. Final solution

The final screens

The **main window** shows Amadeus listening, her ask bar, system and writing stats, and the Codex entry behind her latest answer.



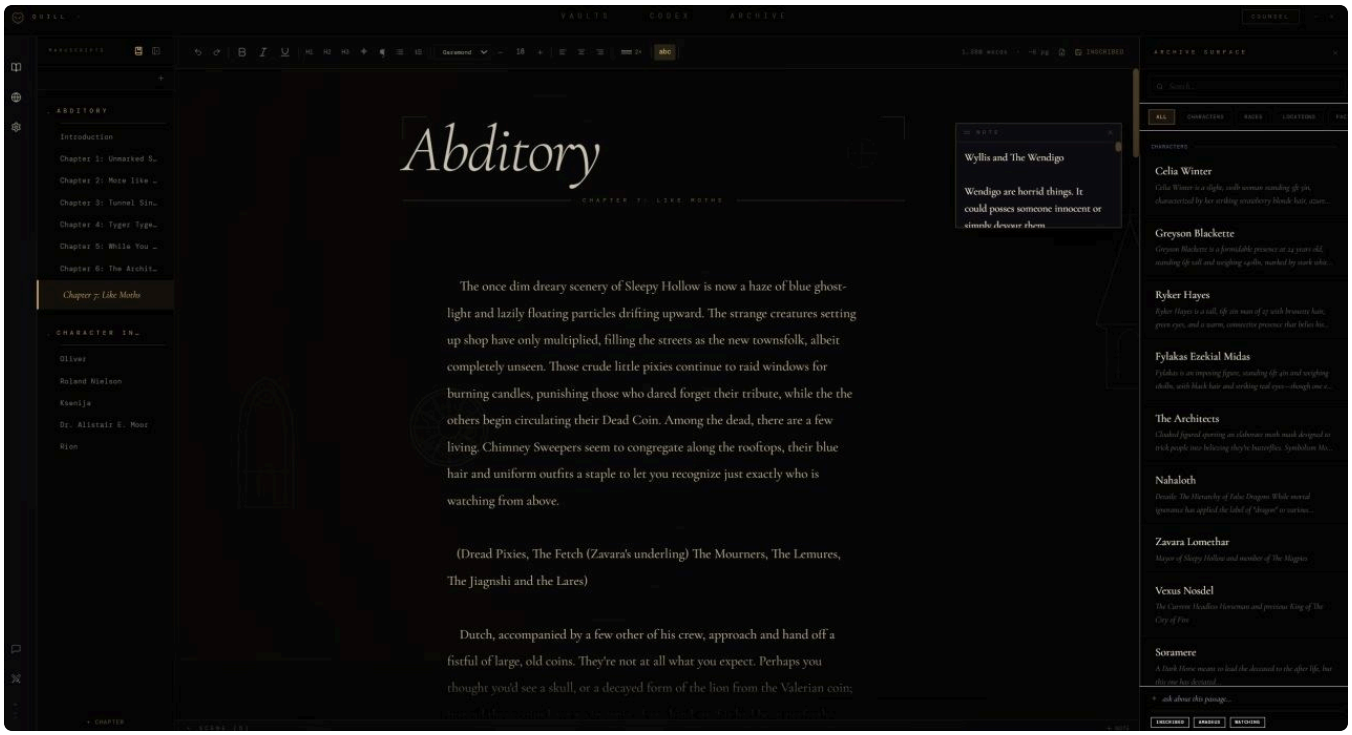
Every answer comes with its **source**: the Codex entry it was drawn from, like Ryker Hayes's character card.



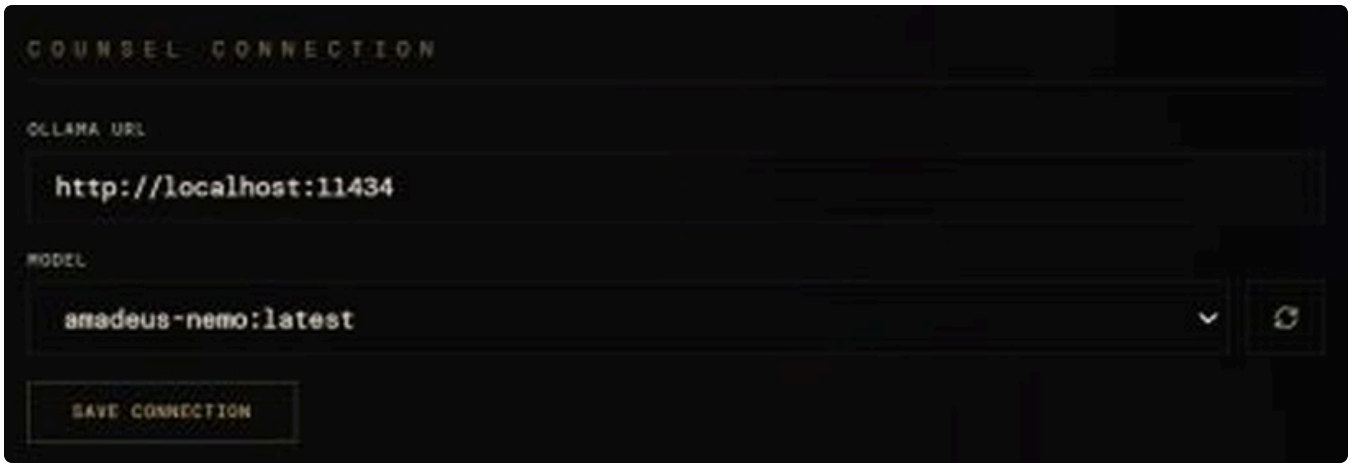
The **Sass and Warmth dials** set her voice, from dry and teasing to warm, without changing the facts.



Inside Quill, the **Archive Surface** lets Wren ask about the passage she's working on, right beside the page.



And the **Counsel connection** in Quill's settings points to the local model, so nothing leaves her computer.



Stack at a glance

Layer	What it uses
Shell	Electron, frameless
AI	Ollama with the amadeus-nemo model, local
Backend	Flask (Python), port 5000
Displays	Main window plus a second-monitor companion
Data bridges	stdout JSON lines into Electron IPC
Codex sync	POST /codex-sync from Quill

Built to be readable

The companion display uses large, glanceable numbers, answers are written in plain language, and every fact can be reached without AI.

Back to Wren's evening

Wren asks what color the creature's scales were. Amadeus answers in a few seconds, with the chapter four entry beside her answer. Nothing left Wren's computer, nobody offered to rewrite her scene, and she never left the page.

07. Current status, feedback and validation plan

Amadeus works as a local prototype alongside Quill. Three people have reacted to her informally; nothing has been measured yet, so here's what they said and how I'll check the rest.

Early reactions

I walked three people through Amadeus, from her start as a coding helper to her role as Quill's Counsel. These were informal reactions, not structured tests.

Kendra Jensen, who isn't a fan of AI, said: "I'm not a huge fan of AI but it being a support tool is much more appreciated than a replacement tool. I especially like that I can change her personality to be as sassy as I want; makes it feel so much more human." She also joked that it sounded like I was building Skynet, something that could almost teach itself.

Dilan Helmick, who doesn't code, said: "I wouldn't even know where to begin with coding languages, so having a constant assistant to just catch but also explain problems is a really good idea." He liked that she does more than one job, comparing her to the personal Jarvis everyone wants.

Michael Dolce called building an AI assistant yourself bold, and noted how hard it is to make one work on an ordinary computer: "How do you make it more than just a chatbot? There's real intention behind it and I look forward to seeing what it looks like a few years from now."

Three things stood out. Kendra's reaction is exactly what the research predicted: someone wary of AI warmed to her because she supports rather than replaces, and the personality dials made her feel human. Dilan saw value in her original purpose, catching and explaining problems, for people learning something new. And two reactions showed that people may overestimate what she is: Kendra imagined something that teaches itself, and Michael described her as a model built from scratch. In fact, she runs on an existing open model; what I designed and built is her personality, her knowledge of the Codex, and everything around her. That's a design lesson in itself: Amadeus needs to be clear about what she is, what she knows, and what she can't do.

What might trip people up

Task	Likely snag	Fix
Ask about a character for the first time	The faint “Say something” prompt gives no examples of what to ask	Show two or three example questions until the first ask
Change her tone	The dials show numbers but not what they change	Show a short sample reply as the dials move
Ask her to write a paragraph	Writers expecting a typical AI tool may be surprised when she declines	Say what she does and doesn't do on first launch
Use her on one monitor	The companion display assumes a second screen	Let the companion panel dock beside the main window
Read small labels	Dim labels on the dark background may fall below recommended contrast	Brighten stat labels and prompts

What's delivered, and what's still unproven

Amadeus is a working local prototype (three windows, a Flask backend, a local model, three bridges and live Codex sync with Quill) with a concept landing page. Everything runs locally; the only network calls are to a font server on first load. What's still unproven is whether writers trust her answers, find details faster, and enjoy her personality at different settings.

How I'll find out

Over about four weeks, I'll recruit five writers and game masters who use Quill. In week 1, I'll time how long they take to find details in their own notes, as a baseline. In weeks 2 and 3, they'll ask Amadeus the same questions in moderated sessions, while I measure time, accuracy and whether they check the source, then ask how much they trusted each answer. I'll also ask each person to request a written paragraph, to see how they react when she declines. In week 4, I'll fix the top issues, starting with the likely snags above, and update this case study with real results.

08. Reflection and next steps

What I learned

Drawing the line early, that she never writes, made the product clearer, not smaller. It decided the comparison, the features, and even her tone. Moving her from code to fiction also taught me that the same help can mean very different things in different crafts: a correction that's welcome in a test file is an intrusion in a manuscript.

What I'd do differently

I'd plan for writers with one monitor from the start, rather than designing around a second screen first.

Open questions

When the Codex disagrees with itself, should she point out the contradiction, or ask which entry is right? She can see all of a writer's work, so how do I show exactly what she knows and what she doesn't, especially when, as early reactions showed, people tend to assume she can do more than she does? And how sassy can she get before a joke gets in the way of the answer?

What's next

Finish refining the Quill integration, then run sessions with writers to answer those three questions. After that, I want to return to her first job: helping people learn to code, by catching problems and explaining why they happen. Dilan's reaction suggests there's an audience for it beyond me.