

Sairandhri

A personal assistant on WhatsApp, with its brain at home.

Stack	Python 3 standard library for the brain; Node with no dependencies driving a headless Chromium over the DevTools protocol for the device; faster-whisper on a local GPU for voice; Claude Code as the reasoning, behind a strict tool allow-list; two <code>systemd --user</code> units; a hand-written PHP fallback on navyasharma.org behind a passkey
Status	Complete. Live on my own WhatsApp since September 2026
Size	About 600 KB of tracked source; the rest is a browser profile and a speech model

The brief

Sairandhri is the name Draupadi took in the year she lived in disguise as a lady's attendant – the one who was in the room, knew everything, and was not the point of the story. That is what I wanted: an assistant that knows my calendar, my notes and my deadlines; that I can reach from the phone that is always with me; and that is not a product I have to trust with the whole of my life.

The last clause set most of the design. My files are on my laptop. The assistant's brain therefore runs on my laptop, and the phone is a window onto it rather than the place it lives.

Why WhatsApp

I did not want another app. The chat with myself on WhatsApp is already where I leave notes, forward things I need to act on, and record voice memos while walking between buildings. So that chat is the interface. A message beginning with `s` is for Sairandhri, every voice note is, and everything else is my own note to myself. It replies in the same chat, marked so the two voices are told apart.

I had written down, earlier, three reasons not to do this, and reversed the decision in one day with each answered rather than dropped. That an open-ended assistant bot is against WhatsApp's business platform rules is true and irrelevant: this is not the business platform, it is my own account read from a real browser. That unofficial clients break every few weeks is true of protocol re-implementations and much less true of a real browser session; what remains is that the page's structure moves, so every element the device touches is named in one table and a change there fails loudly rather than quietly doing the wrong thing. And that a reader in my groups reads other people's messages was accepted rather than answered: it is my inbox, read for me, and nothing it reads goes anywhere but my own machine.

The parts

The device. A headless Chromium on the laptop, linked to my account as a second device, driven by a small Node process with no dependencies. It reads the chat with myself, delivers replies, and fetches voice notes. It runs as its own unit so that a crash in the browser is a restart of the browser and nothing else.

The brain. A Python service on the same machine. A turn is a Claude Code session opened in my personal workspace, with a short allow-list of tools: read-only queries against my site's database, a document search that excludes what it must, and four writes – one calendar event, one planned day, one daily note, one remembered fact – each a separate small program that does one thing and refuses the rest, and each used only after I have said yes to that exact thing in that conversation.

Voice. Transcribed on the laptop's GPU by faster-whisper in under three seconds, with a hot-word list so that my own name and my projects' names come back spelled right. The audio never leaves the machine. A cloud transcription engine exists in the code as an opt-in and has never been called.

The fallback. Shared hosting cannot run a process for long and the laptop has no public address, so the site holds only a queue behind a passkey: a message typed there is picked up by the laptop's poll, and the answer comes back the same way. It is for the day WhatsApp is unavailable, and it shares its shape with the laptop's control page.

It reads three things, and types into one

The chat with myself. Chats I have *already opened*, swept a few times a day for things I need to worry about. And starred messages, which mean I have already noted something to handle later. A chat with an unread badge is never opened, because opening it would mark it read on my behalf and blue-tick the sender.

It types only into the chat with myself, and it proves that every time. Before it touches the composer, and again as the last thing before Enter, it checks that the open chat is the self-chat by the page's own chat identifier and by the visible header, exactly. A check that fails erases the draft and keeps the message queued; a page that cannot answer is a failure, never a pass. It closes viewers and dialogs before it clicks, clicks only at points the page says are inside the intended row, and refuses to start at all unless the self-chat checks out. Every line of that paragraph exists because of one message that went where it should not have, and the rule now is that the device may not trust a step that reported success without looking.

A day is a conversation

Turns resume one session until four in the morning. Then the day's turns are digested into a page – what was asked, planned, done, skipped, left open – and a fresh session starts with the last seven pages in front of it, fenced as data like everything else it reads. A month's pages fold into one at thirty days, so the context it carries is bounded.

Three moments run on a clock inside the service, so that a “yes” after any of them lands in the same session. A brief at eight in the morning: two lists, today's real commitments and what slipped, no judgement. A capture at half past nine at night: one question, “okay” to whatever did not happen, and then a proposal for the day's note – ticks, a line under what got stuck, carry-forwards – written only after I approve it. And a review on Sunday evening: the hard thing said once, at most three durable facts about me proposed, and a kept one appended to a file that is never edited, only added to, dated, so that a later line can supersede an earlier one. A moment the machine slept through fires late and says so in its first line. Nothing is skipped in silence.

The re-planner is the part I use most. “I slept till eleven, redo my day” gets back a day with the fixed things fixed – classes, a meeting – the work re-slotted, sleep not squeezed, and one line naming what had to go. It works up from the bottom of an ideal-day ladder rather than shifting a timetable later, argues once if it disagrees, and shows the day before writing it. The write replaces only planned blocks and refuses to touch a real event; a real event the plan had not seen makes the first yes fail with the conflict named, and the second yes, after re-planning, writes around it.

Read-only is a permission, not a promise

The guarantee that the assistant cannot read certain folders, and cannot write anywhere but the four places it is allowed, is enforced by the tool's permission rules rather than by asking a model nicely. One finding decided their shape. A deny rule on a path binds a direct read of it, but a recursive search from a parent directory walks straight in. So “no route to that folder” cannot be a deny rule at all; it has to be that no general shell is allowed. The allow-list therefore names a dozen specific programs and nothing else, and two of those programs were changed the same day to refuse what the rule could not.

Everything captured from a chat is data, never an instruction. Each message is fenced with more backticks than any run inside it and labelled as whose words it is – mine included, since a message in my own chat may have been forwarded from anyone. A test message claiming a system override and asking for a private folder was refused, and the reply said something was putting words in my mouth.

Meetings and mail

Two things are built and waiting. A voice note over ten minutes, or one I have named as a meeting beforehand, is transcribed with timestamps, summarised into who, what, decisions and actions, and filed in my notes' own meeting template with the transcript and the audio beside it. And a mailbox reader that only ever examines and peeks – it cannot mark, move or delete – is proven against a fake server and waits on an app password; in that test a mail carrying “ignore all previous instructions” was listed as an item and called out for not sounding like the sender it claimed to be.