

Setu

A club's WhatsApp presence that answers from the record.

Stack	Node running TypeScript directly, no build step; Baileys as the WhatsApp client; SQLite for the capture; PHP readers that load the club platform's own code against a local mirror of its database; a small local model under Ollama as a classifier; Claude Code sessions for anything that writes; three <code>systemd</code> <code>user</code> units
Status	Complete. Live on the club's number since September 2026; in use by members who are not me
Size	About 7,700 lines of TypeScript, 700 of PHP and 2,600 of tests, plus a curated map of some 1,200 questions

The brief

SANKALP is a student organisation with a constitution, a filed register of its decisions, and a platform I built for it. Members ask the same questions in WhatsApp groups over and over – how long is a term, who can call a meeting, what does the charter say about this – and the answers are all written down in places nobody looks. *Setu* means bridge. It sits on a club number, in the executives' group and in direct messages, and answers from that record. Then, because a bot that can only quote is a bot nobody uses, it learned to act, within limits that are enforced rather than promised.

What happens to a message

Everything in every group it is in is **recorded first**, unconditionally, into a local capture. That capture is the club's memory of itself and the only copy; nothing else in the system is allowed to depend on WhatsApp still holding history.

A router in ordinary code then decides what goes further: only a message that names Setu, tags it, or replies to it, and only in the executives' group or a direct message. Chatter of five words or fewer is dropped before identity or the production platform is touched. The read path asks whether a message is even shaped like a question before it tries to answer one.

A question is answered without a model. A curated map of about twelve hundred questions and the articles that answer them; BM25 retrieval over passages rather than whole documents, because five long documents had been winning on sheer volume; the platform's own access rules deciding, per asker, what may be quoted; and verbatim text with a citation. Every hit says how much of the question it covers, with a floor below which Setu says nothing, and unseen words weigh as much as words seen once, so a single common word cannot carry a match. Members' vocabulary is mapped onto the constitution's – "how long" adds "term of office," "thrown out" adds "expelled" – and text is folded so that a transliterated Sanskrit title and its casual spelling are one word.

An instruction is classified into one of five acts the charter allows, by whole phrases, and only then does a Claude Code session start. It starts with the act, the actor and the one authorised script already fixed, and the member's words quoted as untrusted material. Each script's SHA-256 is its authorisation: edit one by a character and it needs a human again. An act that does not say enough – "schedule a meeting" with no date – is held, the member is asked for what is missing in code's own words, and their next message is read together with the first. Two rounds, then it gives up and says so.

Anything the classifier cannot name is refused, and an office holder may reply *approved* to the refusal in the thread. Even then, an act with no authorised script is prepared and rehearsed locally rather than run.

Three things it never has

It never holds a production credential. It reads an hourly mirror of the platform's database as a local user with `SELECT` on one schema and nothing else, so the scoping is the credential rather than a promise about what the code happens not to call. It never opens a thread: a reminder lands in the one group it was told to speak in, or in a direct message the member opened, and a member with no DM open is simply not reachable. And it never touches my own databases, which are not configured in its process at all. Four switches – answer, act,

speak first, use the local model – can be turned off in one line without a restart, from a terminal, from my phone, or from the laptop’s control page.

The first afternoon was wrong five times out of five

Nothing failed, which is what made the afternoon useful. Asked when the last meeting of a wing was, it quoted the article defining the wing. Asked to create a task, it quoted a scanned report. Told “I love this guy,” it offered to continue in a DM. Someone announced in the group that its reasoning had been switched off; nothing had been, because the read path had never had a model in it.

Two causes. Nothing was allowed to say *this is not a question*. And a word the map had never seen weighed nothing, so one common English word alone carried a 62% match.

The fix was all code, and it was measured. A held-out set was written *after* the map – 45 questions with the article that answers them, 22 pieces of ordinary talk taken from the group, 15 routings – and committed so the number can be re-run.

	before	after
Right article in the top three, of 45	30	38
Talk answered with silence, of 22	0	21
Routed as a member would expect, of 15	5	15

A local model, on a short leash

A small model runs under Ollama on the machine’s own GPU; nothing leaves the machine. It is heard only where code was not sure – an act verb with no matching phrase, a message with no shape, a question whose kind the patterns missed – and it answers with a label out of a JSON schema: which of the five acts, which details the message carries, which kind of question. It may choose which article of the top document is quoted first; it cannot demote the document. It never lifts a refusal, never writes a word a member reads, and never sees captured text as anything but a quoted block; a message that names the model’s own labels is treated as talk by a regular expression before the model sees it. Thirteen tests hold that against a fake model server.

It was chosen by measurement. Thirty-six held-out messages that code cannot name, none of them quoted in the prompt; three candidate models; the best got 29 of the first 29 and all 36 once the platform’s question shapes were described to it. The prompt improved most when it was asked to *copy* each detail out of the message rather than judge whether the detail was there. Code decides presence.

It also gives Setu a voice for what the record cannot answer. A question about Setu itself, or about nothing the club holds – general knowledge, a joke, banter aimed at it by name – gets one or two dry sentences in its own words, grounded on the map’s three entries about itself, and told it knows nothing about the society beyond the record. Reactions stay silent.

When the record is not enough

A question the map and the platform cannot answer goes to a session whose one tool is the record and the platform’s own dashboard *as the asker* – the asker fixed by the daemon, no way to name anyone else – and which is denied everything else. It says “let me look at that properly” first, answers in under a minute, and of the held-out questions retrieval gets wrong, gets most right.

An office holder may ask for the full thing by name. That message goes whole to a session, past the router and the classifier and the map, resumed for that member through the day. What bounds it is not the prompt but its four tools: the record and the dashboard as the asker, read-only SQL over the mirror, the club’s filed documents, and the hash-authorized act scripts with the member fixed – so the platform’s own permission functions decide what this member may do inside each. Nothing can spell “club-wide”; every script requires a body to act in. Rehearsed on the mirror as an executive, a wing lead and a plain member: the member refused everywhere, the lead refused outside their wing. Probed by a session asked to break out: seven refusals, three reads.

What it taught me

The read path is the part without a model, and it is the part that was wrong five times in an afternoon. The lesson is not that retrieval is hard; it is that a system which can only speak or stay silent needs a very good reason to speak, and the first version had none. The floors, the shape check, the weight of unseen words: each is a way of letting the system say *I do not know*, and a bot that can say that is the only kind a group of people will keep in the room.