Pulse AI (WhatsApp AI Agent) — Where AI Meets a Real Customer

How I leverage AI as a builder: a production AI agent handling real customer conversations, with the safety and reliability layer that production actually requires

Aman Patel | whatsapp-ai-agent-inky.vercel.app | github.com/Aman241104/whatsapp-ai-agent

Pulse AI is a multi-tenant WhatsApp Business SaaS: an AI agent that replies to a business's real customers over the actual WhatsApp Business Cloud API — instantly, in-conversation, 24/7 — backed by an admin dashboard a non-technical business owner can run without writing a line of code. The AI isn't a feature here; it *is* the product.

Next.js 16 · Supabase (Postgres, RLS, Realtime) · Groq (Llama 3.1) with automatic fallback provider · Meta Cloud API v19 (WhatsApp/Messenger/Instagram) · Vitest

1. The part that actually matters in production: not letting the AI lie

Every reply the agent drafts is grounded strictly in that business's own knowledge base — no invented prices or policies. Before anything reaches the customer, a **second AI pass reviews the draft** specifically to catch fabricated prices or promises the first pass slipped in. If the primary provider (Groq) hits a rate limit, a secondary model takes over invisibly — the customer never sees a dropped reply.

> The design principle: an AI agent talking to real customers about real money needs a verification step and a failure mode, not just a good prompt. Grounding + review-before-send + provider fallback are the three things that turn "an LLM demo" into something you'd actually let a business run unsupervised.

2. Knowing when *not* to be the AI

The agent detects customer frustration and "talk to a real person" requests — without relying on a magic keyword — and hands off to a human with full conversation context, so the team isn't starting from zero. Bookings happen directly inside the WhatsApp thread (availability checking, conflict prevention, reminders, one-word cancellation), so the AI is doing real transactional work, not just answering FAQs.

3. AI Copilot with a confirm step before anything irreversible

The admin dashboard has its own natural-language Copilot — *"add Rahul as a contact with tag VIP," "send a broadcast about our Diwali offer to all wholesale contacts"* — that executes the request, but inserts an explicit confirm step before anything irreversible (like a broadcast to an entire contact list) actually fires. Same principle as the customer-facing side: let AI act, but put a checkpoint in front of the actions that are expensive to get wrong.

4. Why this is the story I'd tell

- **AI as the product, not a bolt-on.** Multi-tenant, RLS-isolated, live in production — this isn't a prototype, it's a system a real business could run their customer support on tonight.
- **Reliability engineering around the model, not just the model.** Grounding, a second-pass reviewer, and provider fallback exist because "the LLM said something wrong" is a production incident, not an edge case.
- **AI given real leverage, with a leash.** Both the customer-facing agent and the internal Copilot can take real actions (bookings, broadcasts, contact edits) — but every irreversible one has a human checkpoint or a review pass in front of it.

Live and in active production use. Full architecture (webhook signing, tenant resolution, knowledge-base retrieval) is documented in the repo's README and AGENTS.md.