Website RAG Chatbot

2026-07-09

The chatbot embedded on this website. It reads my whole portfolio and FAQ, answers the exact question a visitor came for, and cites the page each answer came from so they can click through.

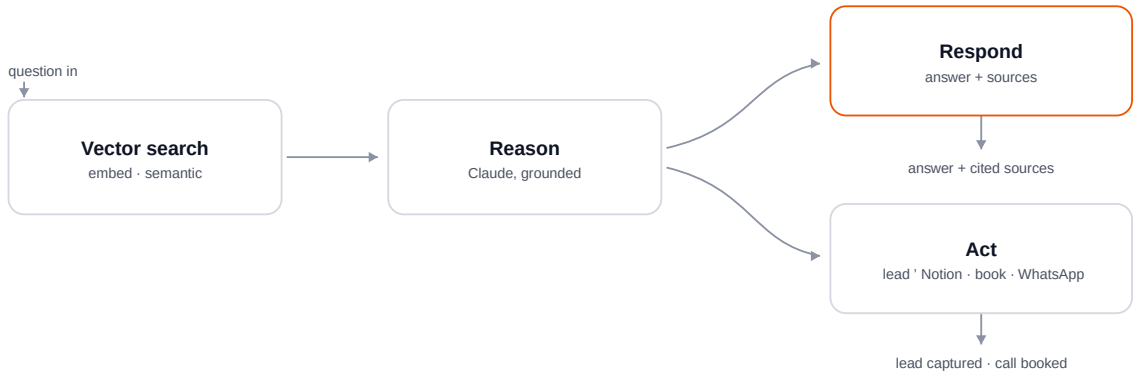
Claude API · Gemini embeddings · pgvector · RAG evals · Next.js

Problem

Most people do not read everything on a website. They skim, and they leave. So a visitor can bounce before they ever reach the products, the services, or the one detail that would have convinced them. Whether you sell physical products or services, that reading gap is quietly costing you conversions. I wanted a way to hand every visitor the exact answer they came for, instantly, without asking them to dig through pages, and to prove it by linking to where each answer came from.

Approach

I built a retrieval-augmented chatbot and embedded it on the site. First I mapped my own content, every service, every portfolio piece, my process and FAQ, into a curated knowledge base of short documents, and embedded each one into a vector database. When a visitor asks something, the system embeds the question, runs a semantic search over those vectors, and Claude answers grounded only in the documents it retrieves, so it never invents. Every grounded answer surfaces its sources as clickable links, so the visitor can jump straight to the page it came from. The embedding step is free (Gemini) and the vector store is pluggable: it runs in memory on this site, and swaps to a pgvector (Postgres) backend when the corpus grows. Because I come from statistics, I did not want to guess at quality, so I built an evaluation harness that scores retrieval (recall and precision at k) and answer faithfulness against a golden set. On top of answering, I gave it tools it can decide to use: capture a lead straight into my Notion CRM, offer my booking link, or hand the visitor to WhatsApp with the message already written.



Result

It runs in production on this site for a fraction of a cent per conversation and no fixed infrastructure. Visitors get instant, grounded, sourced answers in three languages, at any hour, without reading a single page in full. It reasons about fit too: ask whether I could handle a senior AI engineering role and it answers with the relevant evidence. It quietly captures leads while I sleep, and it doubles as the clearest proof of the work: when someone asks whether I can build a chatbot, the honest answer is that they are already talking to one.

What I'd do differently

Wire the eval harness into CI so every change is scored before it ships, add a reranking pass over the retrieved documents, and log every conversation from day one so I can see what visitors actually ask and tighten the weak answers.

Repository: **https://github.com/hermann-ndamen/website-rag-chatbot**

Interactive version and demo: **https://www.hermannndamen.com/portfolio/ai-engineer/website-rag-chatbot**