Polymer app

**Pre-MVP:** Supabase (login and database) + GitHub + Vercel setup, then I create the database schema (tables only, no triggers, cost function comes later in MVP coding phase), must be prepared for mobile and web app, must be ready for health feature, and voice feature. Also define the framework, language and libraries for frontend and backend, and the database then host it somewhere (Vercel)

**Build order:**

1. MVP first (you have all APIs ready)
   1. One page where I can register the deal
   2. Once I press register deal it sends a WhatsApp, telegram feature and updates google sheets
   3. View it in a list, on a separate page
2. Health Feature – check everything. And make it so that I can add new stuff there
3. When I click on the list, I should be able to see it visually
4. Upload data
5. Dashboard
6. Chat with Data (15 years of data = big value)
7. Price List
8. Inventory
9. Document Vault
10. Credit Tracking
11. Web Crawlers
12. Alerts

**Language:** TypeScript  
**Framework:** Next.js 14+  
**Database:** PostgreSQL (Supabase)  
**Hosting:** Vercel  
**Auth:** Supabase Auth

**Web server:** Vercel Edge Network (automatic)  
**Application server:** Vercel Serverless Functions (automatic)

--

Good Practices

* 1. Always start with the premvp part, making the codebase, connecting it to github and branch, vercel, supabase for login and database.
  2. Plan everything in advance, all the features, describe what the ultimate version will look like. Mobile and web both or one, all the details
  3. Decide the mvp based on it and plan which feature to add one after another

Project info:

Supabase:

Anon: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InpnbWRjcGFseGZ2aWxoY2tpc3VmIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NTk4NTMyNzMsImV4cCI6MjA3NTQyOTI3M30.z7TBdasOawjyjANF\_YE4QQ3sdhfGlW6TqU3x3e7ps5k

Service\_role: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6InpnbWRjcGFseGZ2aWxoY2tpc3VmIiwicm9sZSI6InNlcnZpY2Vfcm9sZSIsImlhdCI6MTc1OTg1MzI3MywiZXhwIjoyMDc1NDI5MjczfQ.znHcmF6yvOJw9NkUhWyRD6jedPdFqY7V5EAme6WpUbk

Openai API Key: sk-proj-pYvXByNbQiTRssoeZtZ35LZOCBzN-4YwrlSZSMbTHrBEzPWNnI7is9VAoairqO3nc9AQsh5r4LT3BlbkFJ2KmmMmyd\_Ox3TzrnjwJpl4\_5CMNv065VIGuUpq2XdMAL5-FN6JTPRx4jkkxWIlAcIWQAm72wwA

AlphaAdvantage: 9GDI9NIANXJEKI3B

--

Infinite - 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

--

So, now you can make agents and workflows. You can ideally connect it to customized DIY hardware like small tablets and projectors and what not.

What do you want to make.

People are selling agents.

* What can you make with agents – different possibilities – what workflows can you make
* Can you make a digital assistant using OpenAI agents sdk
* How to make sure that Claude Code doesn’t make the same mistake when working with openai agents sdk
* Best way to build complex agents and workflows – use n8n for complex and openai sdk for simpler ones.

LangChain and LangGraph and RAG, and RAG Pipeline