

Vora IQ

The AI operating system for founders

The Chat-to-Content Framework

The complete guide to turning AI conversations into a full content engine, SEO infrastructure, and auto-publishing pipeline.

Tools used: Claude, Cursor, Vercel, Next.js, GitHub, Google Search Console

Time: One session. Results: 6 weeks of autopilot content.

What you need before you start

Claude (claude.ai)

Your content strategist. Use it to audit your SEO, search your past conversations, map articles to keywords, and generate the articles themselves. Claude's memory and past chat search are what make the extraction process possible.

Cursor (cursor.com)

Your code editor. Use it to implement the blog infrastructure, components, and routes in your Next.js project. Cursor's AI features help you wire up files faster, but you're still making the architectural decisions.

Vercel (vercel.com)

Your hosting and deployment platform. Handles auto-deploys from GitHub, cron jobs for scheduled publishing, and environment variables for secrets. If your landing page is already on Vercel, you're set.

Next.js

Your framework. The blog uses file-based routing, MDX for rich content, gray-matter for frontmatter parsing, and next-mdx-remote for rendering. Static generation means fast pages and good SEO.

GitHub

Your repo. Push all articles at once with future publish dates. Vercel auto-deploys on push to main. The cron job handles the rest.

Google Search Console

Your SEO dashboard. Submit your sitemap, request indexing, and monitor which keywords you're appearing for. Free and essential.

Medium (medium.com)

Your backlink source. Medium has a domain authority of 95+. Every link from a Medium article to your site carries real SEO weight. Update existing articles with links back to your domain.

PHASE 01

Mine your chat history

30 minutes

Open Claude and search your past conversations

Claude can search through your conversation history. Use keywords related to your product, your opinions, your decisions, and your journey. Every strong conversation is a potential article.

Tag each conversation by theme

Create a simple list. For each conversation, write the theme: Founder story, Product decision, Industry hot take, Framework, Honest failure. Each theme becomes a content pillar.

"That conversation where I explained why I left my job = Founder story pillar"

Select the 10-15 strongest

Look for: strong opinions, specific stories, frameworks you explained, decisions you defended. If you argued for something passionately, it's an article. Cut anything generic.

How I did it

I asked Claude to search my past conversations

Claude pulled up conversations about positioning Vora IQ, my YC application, my LinkedIn posts, agent architecture decisions, career history, and industry opinions. Each search returned the raw material for 2-3 articles.

PHASE 02

Map conversations to SEO targets

30 minutes

Find the keyword for each article

Ask yourself: what would someone Google to find this? Use Google autocomplete to validate demand. Type the first few words and see what Google suggests.

"How I built AI agents" maps to keyword: "AI business agents"

Assign a content pillar

Founder story, Education, Product deep dive, Industry hot take, Category creation, Build in public. Each pillar attracts a different audience and ranks for different keywords.

Write the hook for each article

One sentence that stops the scroll. Pull it from the strongest moment in the original conversation. If it made you feel something when you first said it, it'll work as a hook.

How I did it

I created a publishing calendar

12 articles mapped to 6 weeks. Monday and Wednesday publishes. Each article assigned a keyword, pillar, hook, and outline. Claude generated a full content map document I could reference throughout.

PHASE 03

Generate the articles

2-3 hours

Feed each conversation back to Claude with structure

Prompt: "Here's a conversation I had about [topic]. Turn this into a 1,500-word blog article targeting keyword [X]. Keep my voice. Don't invent facts. Include internal links to [your pages]."

Weave in backlinks to your other properties

Link to your Medium articles, your other sites, your social posts. Every cross-reference creates a web that Google reads as topical authority.

Review for accuracy, not perfection

The AI organized your thinking. Check that it didn't invent experiences or overstate claims. The voice should sound like you on your best day, not a marketing team.

How I did it

Claude generated all 12 articles in one session

Each article referenced the original conversation, included 2+ internal links to voraiq.com pages, linked to relevant Medium articles, and ended with a consistent CTA and sign-off line.

I saved each article as a .md file with frontmatter

Each file has YAML frontmatter with title, description, publishDate, author, pillar, keyword, tags, and draft status. The publishDate is the key: it controls when the article goes live.

```
---
title: "Why I Left Meta to Build an AI Operating System"
description: "The real story behind Vora IQ..."
publishDate: "2026-03-31"
author: "Khalel Dumaz"
pillar: "Founder story"
keyword: "AI for entrepreneurs"
tags: ["founder story", "AI", "Meta"]
draft: false
---
```

PHASE 04

Build the blog infrastructure

2-3 hours

Step 1: Install dependencies

```
npm install next-mdx-remote gray-matter
```

Step 2: Create the blog reader (src/lib/blog.ts)

This file reads .md and .mdx files from content/blog/

It parses frontmatter with gray-matter, filters out drafts and future-dated posts, and sorts by date. The key function is getPublishedPosts() which only returns articles where publishDate is today or earlier.

Step 3: Create the routes

app/blog/page.tsx

The blog listing page. Shows all published posts sorted by date with pillar labels, dates, and tag pills.

app/blog/[slug]/page.tsx

The individual post page. Renders MDX content with custom styled components, generates metadata for SEO, includes breadcrumbs, author info, a bottom CTA, and related posts.

app/api/revalidate/route.ts

The cron endpoint. Vercel hits this on schedule to rebuild the site. It checks for a Bearer token (your CRON_SECRET) to prevent unauthorized triggers.

app/rss.xml/route.ts

RSS feed for your blog. Useful for syndication and as a signal to search engines that your content updates regularly.

Step 4: Create blog components

BlogMarkdown.tsx

The MDX renderer. Uses next-mdx-remote with custom components for callouts, images with captions, inline CTAs, agent cards, viability score previews, and quiz embeds. All styled in your brand colors.

blog-cta.tsx

The bottom-of-post CTA component. Appears on every article with a link to try Vora IQ free.

Step 5: Configure Vercel

Add CRON_SECRET to environment variables

In Vercel dashboard: Settings > Environment Variables > Add. Key: CRON_SECRET. Value: any random string. Select All Environments.

Add cron config to vercel.json

This tells Vercel to hit your revalidation endpoint every Monday and Wednesday at 9 AM EST.

```
{
  "crons": [
    {
      "path": "/api/revalidate",
      "schedule": "0 14 * * 1,3"
    }
  ]
}
```

Step 6: Push articles to content/blog/

Drop all .md files into content/blog/ and push to main

All articles have future publishDate values. They won't appear on the site until their date arrives and the cron job triggers a rebuild. Push once, auto-publishes for 6 weeks.

PHASE 05

Deploy the SEO foundation

1 hour

What I deployed in one session

robots.txt

Tells search engines what to crawl. Place at your domain root. References your sitemap.

```
User-agent: *  
Allow: /  
Disallow: /api/  
Sitemap: https://www.yourdomain.com/sitemap.xml
```

sitemap.xml

Lists every indexable page with priority and update frequency. Submit to Google Search Console after deploying.

Meta tags for every page

Title, description, Open Graph, and Twitter cards. Each page gets unique, keyword-targeted metadata. Use your framework's metadata API.

Structured data (JSON-LD)

Organization schema on all pages. SoftwareApplication schema on the homepage. FAQ schema on any page with FAQ content. This is what generates rich snippets in Google.

Image alt text

Every image on every page needs descriptive alt text. This is both an accessibility requirement and an SEO signal.

Verify everything

Google Rich Results Test

Go to search.google.com/test/rich-results and enter your URL. Verify your FAQ, Organization, and SoftwareApplication schemas are valid.

Google Search Console

Submit your sitemap. Request indexing for your homepage and key pages. Set up email alerts for crawl errors.

PHASE 06

Build the backlink web

30 minutes + ongoing

Update your Medium articles

Add links to your domain in every existing Medium article. Add an author bio at the bottom with a link to your site. Medium's DA 95 makes these high-value backlinks.

Cross-link your properties

Your blog links to Medium. Medium links to your site. Your other sites (guides, tools, resources) link to both. Google reads interconnected, topically related domains as authority.

Submit to startup directories

Product Hunt, BetaList, SaaSHub, Indie Hackers, AlternativeTo, There's An AI For That, Futurepedia. Budget 15 minutes per submission. Each one is a backlink.

Social distribution per article

Same day each article publishes: LinkedIn thought-leadership post (not a link drop), Threads/X short version with link. The social posts drive immediate traffic while SEO builds long-term.

WHAT YOU WALK AWAY WITH

The output.

- > 6+ weeks of auto-publishing content
 - > Full SEO infrastructure (meta, schema, sitemap)
 - > A multi-domain backlink network
 - > Social distribution pipeline per article
 - > All sourced from thinking you already did
-

TRY VORA IQ FREE

Your idea deserves more than a **guess.**

Vora IQ validates your idea, builds your roadmap, and helps you execute with 13 AI agents. Start free.

voraiq.com

"I built a real AI co-founder. Now I'm shipping the product."

Khalel Dumaz | Co-Founder & Design Technologist, Vora IQ

voraiq.com | thevibecodebible.com