

THE FREE PLAYBOOK

# The AI SEO Playbook

Five systems that put my business on page one of Google and on autopilot across every platform. The exact stack I used to go from invisible to ranked — with zero coding experience and zero ad spend.

SYSTEM 1 · llms.txt

SYSTEM 2 · MARKDOWN MIRRORS

SYSTEM 3 · SITEMAPS + GSC

SYSTEM 4 · YOUTUBE AUTOPILOT

SYSTEM 5 · FULL WEBSITE BUILDER

Built by @brycenwood.ai · April 2026

WHAT'S INSIDE · TAP TO JUMP

## Five Systems. One Playbook.

### 1 The AI Recommendation System (llms.txt)

Make ChatGPT, Claude, and Perplexity actively recommend your business when customers ask. The exact file and structure I used to get my shop quoted by name in AI answers. →

### 2 Markdown Mirrors

Give AI a clean, plain-text version of every page on your site so they can quote you faster, more accurately, and more often. The companion to llms.txt that almost nobody is doing yet. →

### 3 Sitemaps + Google Search Console

Tell Google exactly which pages matter, then see your actual ranking data — which keywords you rank for, what position you're in, and where the quick wins are. →

### 4 YouTube Autopilot

Auto-cross-post every Instagram Reel to YouTube Shorts with AI-optimized titles, descriptions, and tags built for YouTube's search algorithm. One prompt builds the entire system. →

### 5 Full Website Builder Prompt

The exact prompt I used to build a 40-page website in one afternoon. Swap in your business info, paste it into Claude Code, and get a complete site with SEO, schema, blog posts, and AI visibility built in. →

#### HOW TO USE THIS PLAYBOOK

Each system has step-by-step prompts you paste directly into Claude Code. You can build all three in a single afternoon. If you get stuck on any step or want the ready-made scripts, I share those inside my free community at [brycenwood.com/community](https://brycenwood.com/community) — but the playbook alone is enough to ship everything.

# The AI Recommendation System

Make ChatGPT, Claude, and Perplexity recommend your business by name.

---

## INTRO

What's up. I'm Brycen. I own a vehicle wrap company in Utah called Summit Wraps and Graphics. A couple months ago I typed "who does the best vehicle wraps in Utah" into ChatGPT and it recommended my shop. I didn't pay for it. I didn't run an ad. I just made my website readable to AI.

Here's the thing nobody is talking about yet: AI chat is replacing Google for a lot of searches. When someone asks ChatGPT for a plumber, an HVAC company, a contractor, a mechanic, that AI has to pull from somewhere. If your business shows up in those answers, you get the lead. If it doesn't, you don't exist.

This guide shows you exactly what I did, step by step, prompt by prompt. Copy it, swap in your business, drop it on your site. That's it.

---

## WHAT YOU'LL BUILD

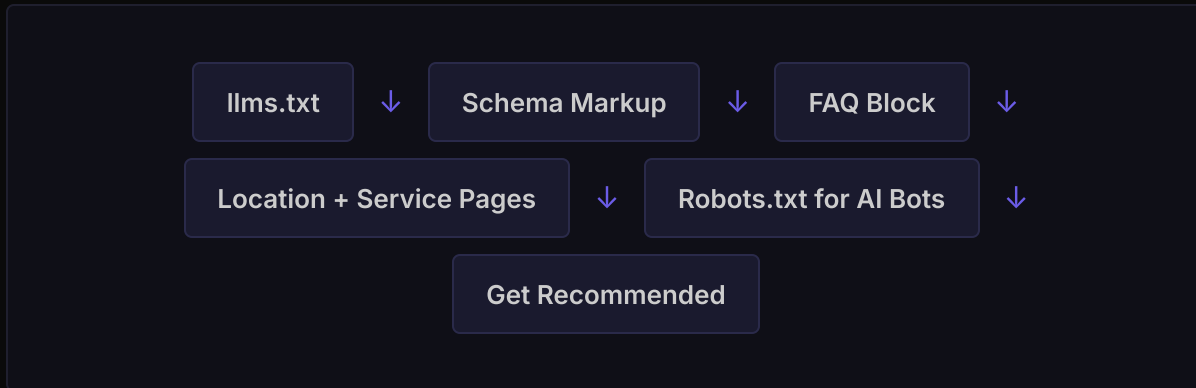
### OVERVIEW

## What You're Building

By the end of this guide, your website will have everything AI models need to recommend your business:

1. **An llms.txt file** — a plain text file that tells AI exactly what you do, where, and for how much
2. **Structured data (schema)** — machine-readable business info AI and Google both love

3. **A strong FAQ section** — direct Q&A pairs that AI quotes back to users
4. **Location + industry pages** — one page per city and per service so AI has something specific to cite
5. **AI crawler access** — robots.txt set up so GPTBot, ClaudeBot, and PerplexityBot can actually read your site



#### WHY THIS WORKS

AI models don't "search" the way Google does. They pull from pages that are clear, structured, and easy to summarize. Most business websites are bloated with marketing fluff and pop-ups. If you give AI a clean, direct source of truth, it will quote you over the noise.

## PREREQUISITES

### BEFORE YOU START

## What You Need

- **Claude Code** installed on your computer (the CLI tool, not claude.ai)
- **Access to your website files** — whatever host you use (GitHub, Netlify, Squarespace, WordPress, Webflow, etc.). You need to be able to upload a new text file to the root of your site.
- **Basic info about your business** — services, pricing ranges, locations, hours, what makes you different. Have this in your head or in a notes doc before you start.
- **15–30 minutes** for the llms.txt + schema. A few hours if you want to do the full location/service pages.

#### HOW MUCH DOES THIS COST?

Zero extra dollars. The whole system is plain text files you drop on the site you already own. If you're on Claude Code's \$20/month subscription you're already paying for it.

## STEP 1

### STEP 1

## Create Your llms.txt File

This is the main file. It's a plain text document that sits at `yourwebsite.com/llms.txt` and tells AI exactly what your business is. Think of it like a one-page pitch written for a machine.

### PASTE THIS INTO CLAUDE CODE

```
I need you to create an llms.txt file for my business. This is a plain text file that lives at the root of my website. It tells AI chat models (ChatGPT, Claude, Perplexity) exactly what my business does, so they can recommend me when someone asks about my industry.
```

```
Here are my business details:
```

```
Business Name: [YOUR BUSINESS NAME]
What I do: [ONE-SENTENCE DESCRIPTION]
Founded by / owners: [YOUR NAME]
Locations: [CITY, STATE – list all locations]
Phone: [YOUR PHONE]
Website: [YOUR URL]
Hours: [YOUR HOURS]
Service area: [LIST CITIES/COUNTIES YOU SERVE]
```

```
Services and pricing:
– [SERVICE 1]: [PRICE RANGE]
– [SERVICE 2]: [PRICE RANGE]
– [SERVICE 3]: [PRICE RANGE]
(add as many as you offer)
```

```
What makes me different: [2-3 SENTENCES ABOUT YOUR EDGE – materials, warranty, in-house vs. outsourced, experience, etc.]
```

```
Ideal customers / industries served: [WHO YOU WANT TO WORK WITH]
```

Key facts and credentials: [YEARS IN BUSINESS, # OF JOBS DONE, REVIEW COUNT, CERTIFICATIONS, WARRANTY]

Now write the llms.txt file following this structure:

# [Business Name]

## About

(short paragraph)

## Services and Pricing

(bulleted list with real numbers – AI loves specific pricing)

## Locations

(bulleted list)

## Contact

(phone, website, social, hours)

## Service Area

(comma-separated list of cities)

## Key Facts

(bulleted list of credibility points)

## What Makes Us Different

(short paragraph)

## Frequently Asked Questions

(5–8 real Q&A pairs that a customer would actually ask – include pricing, timing, warranty, common concerns)

Make it scannable. Use short sentences. Include specific numbers wherever possible. No marketing fluff. Save it as llms.txt in my website's root folder.

### WHY PRICING MATTERS

AI models get asked "how much does X cost" constantly. If your llms.txt has real pricing ranges, you will get quoted. If it says "call for pricing" you will not. Give the AI something to say.

# Upload It to Your Website

The file has to live at `yourwebsite.com/llms.txt` — not in a subfolder, not in a blog post, not behind a login. Root of the site. Test it in your browser after uploading by typing the URL directly.

## PASTE THIS INTO CLAUDE CODE

```
I have a website hosted on [YOUR HOST –  
GitHub/Netlify/Squarespace/WordPress/Webflow/etc.]
```

```
Walk me through exactly how to upload the llms.txt file we just  
created to the root of my site. Give me step-by-step instructions for  
my specific host. After I upload it, tell me how to confirm it's live  
by visiting the URL directly.
```

## COMMON GOTCHA

Some hosts (looking at you, Squarespace) make it hard to drop plain text files at the root. If yours is one of them, ask Claude Code for the workaround. There's always one — redirect, custom route, or switching to a different static host.

## STEP 3

### STEP 3

## Add Schema Markup (Structured Data)

Schema is code AI and Google can read to understand exactly what your business is. It lives inside your website HTML, invisible to visitors, but very visible to machines.

## PASTE THIS INTO CLAUDE CODE

```
I want to add JSON-LD schema markup to my homepage so AI and search  
engines understand my business better.
```

```
Business details:
```

```
– Name: [YOUR BUSINESS]
```

- Type: [e.g., LocalBusiness, AutoRepair, Plumber, HVACBusiness, Restaurant]
- Description: [ONE LINE]
- Address: [FULL STREET ADDRESS]
- Phone: [YOUR PHONE]
- Website: [YOUR URL]
- Hours: [YOUR HOURS]
- Price range: [\$, \$\$, \$\$\$]
- Services: [LIST]
- Review rating and count: [IF YOU HAVE GOOGLE REVIEWS]
- Lat/long: [IF YOU KNOW – otherwise skip]

Generate JSON-LD structured data using schema.org types:

1. LocalBusiness (or the most specific type for my industry)
2. FAQPage – with 5-8 common customer questions and answers
3. Service – one entry per main service I offer

Give me the code to paste inside the <head> section of my website.  
Also explain what each block does so I can check it's accurate.

After you paste it in, test the schema at [validator.schema.org](https://validator.schema.org) and [search.google.com/test/rich-results](https://search.google.com/test/rich-results). Both free. They'll tell you if anything is broken.

## STEP 4

### STEP 4

## Open the Door for AI Crawlers

AI bots like GPTBot (ChatGPT), ClaudeBot (Claude), and PerplexityBot have to be allowed to crawl your site. A lot of templates block them by default. You want them in.

### PASTE THIS INTO CLAUDE CODE

Check the robots.txt file on my website at [YOUR URL]/robots.txt. I want to make sure the following AI crawlers are explicitly ALLOWED (not blocked):

- GPTBot (OpenAI / ChatGPT)
- ClaudeBot (Anthropic)
- PerplexityBot (Perplexity)

- Google-Extended (Google's AI training bot)
- CCBot (Common Crawl - a lot of models train on this)

If any are blocked, give me the updated robots.txt file I should upload. Also add a line pointing to my sitemap.xml so everyone can find my pages.

#### COUNTERINTUITIVE BUT TRUE

Every "SEO best practice" article from 2023 told you to block AI bots to "protect your content." Don't. You want AI to read your site. The whole point is to get quoted in AI answers. Let them in.

## STEP 5

### STEP 5

## Add a Real FAQ Section to Your Homepage

AI pulls quotes directly from FAQ blocks. They're structured, they're direct, and they match how people ask questions. If your homepage has a good FAQ, AI will quote you word-for-word in its answers.

#### PASTE THIS INTO CLAUDE CODE

Write an FAQ section for my homepage. It should have 8-10 real questions a potential customer would actually type into ChatGPT or Google.

My business: [DESCRIBE IT]

My services: [LIST]

My pricing: [REAL RANGES]

My area: [CITY/STATE]

Rules:

- Every question should start with how, what, where, when, why, or do
- Every answer should be 2-4 sentences max, direct, include specific numbers when possible
- Include at least one pricing question, one "how long does it take" question, one "do you serve my area" question, and one "what makes you different" question
- Write the answers so they can be quoted directly by AI - short,

factual, confident

- No marketing words: avoid "premier", "best-in-class", "industry-leading", "cutting-edge"

Then wrap the whole thing in FAQPage schema markup so AI and Google both index it properly.

## STEP 6

### STEP 6

## Build Location + Service Pages (Optional but Powerful)

This is the step that took me from being mentioned to being the top recommendation. Create one page per city you serve and one page per core service. Each page is a clean, specific, AI-readable target.

### PASTE THIS INTO CLAUDE CODE

I want to build location pages and service pages for my website that rank in both Google and AI chat results.

My service area: [LIST OF CITIES YOU SERVE]

My core services: [LIST OF 3-6 MAIN SERVICES]

Generate two sets of pages:

LOCATION PAGES – one per city. URL format: /locations/[city-name]/

Each page should include:

- H1 with "[Service] in [City], [State]"
- Intro paragraph mentioning the city by name and my business by name
- 400-600 words covering: why local matters, services offered in that city, specific neighborhoods or landmarks, pricing, contact info
- Embedded FAQ (3-5 questions) specific to that city
- Schema markup for LocalBusiness with the city's info
- Internal links back to the homepage and related service pages

SERVICE PAGES – one per core service. URL format: /services/[service-name]/

Each page should include:

- H1 with the service name
- 600-800 words covering: what the service is, who it's for, what it costs, how long it takes, what's included, the process step by step

- A "who this is for" section
- FAQ schema block
- Internal links back to the homepage and related services

Write real, useful content. Not keyword-stuffed garbage. Assume a real customer will read these. Start with 3 location pages and 2 service pages. We'll expand from there.

#### THE MULTIPLIER EFFECT

One llms.txt gets you in the door. Twenty location and service pages get you quoted. Every page is another chance for an AI to pull from your site instead of a competitor's. This is what took my traffic from ~7/day to ~150/day in two weeks.

## STEP 7

### STEP 7

## Test It

Wait 3-7 days after everything is live (AI models need time to recrawl). Then test:

1. Open ChatGPT. Ask "who does the best [your service] in [your city]?"
2. Ask Perplexity the same question.
3. Ask Claude the same question.
4. Ask Google's AI Overview (appears at the top of normal Google searches) the same question.

If you show up in 1-2 of them early, you're on the path. If you show up in all 4, you've built a moat. Keep adding pages, keep updating your llms.txt as you add services, and check back every couple weeks.

#### PASTE THIS INTO CLAUDE CODE

Help me audit how well my AI SEO is working. Check [YOUR URL] and give me a report on:

1. Is my llms.txt live and does it have everything important?
2. Is my schema markup valid?
3. Is robots.txt allowing AI bots?
4. Does my homepage have a real FAQ section with schema?

- 5. How many location pages do I have? Are they thin or real content?
- 6. What's missing that top AI-recommended sites in my industry have?

Give me a prioritized list of fixes, most impactful first.

## RESULTS

### MY NUMBERS

## What Happened for Summit Wraps

- **Old site:** ~7 real visitors per day
- **New site (two weeks after launch):** ~100-150 real visitors per day
- **Traffic increase:** ~1,400%
- **ChatGPT result:** actively recommends Summit Wraps when asked about vehicle wraps in Utah
- **Google Search Console:** ranking page 1 for "car wraps near me" within weeks
- **Cost:** \$0 in ads, \$0 to an agency, ~1 week of building with Claude Code
- **Coding experience used:** zero

### REAL TALK

This isn't a hack and it's not going to stay this easy. Right now, almost nobody is doing this. That's the opportunity. In 12 months every decent website will have an llms.txt and schema and location pages. Do it now, while the door is still wide open.

## TIPS

### TIPS

## Things I Learned Building This

1. **Specificity beats polish.** Real prices, real cities, real numbers. AI will quote specifics. It will ignore vague marketing copy.
2. **Update llms.txt every time your business changes.** New service? New price? New location? Update the file. It's literally a text doc.
3. **Don't block AI bots, even "to save on bandwidth."** You want the exposure more than you want the server cycles.

4. **One clean page beats five bloated ones.** A 500-word location page with real info outperforms a 3,000-word SEO monster stuffed with keywords.
  5. **AI cares about FAQs more than paragraphs.** Questions and answers get quoted. Paragraphs get ignored. Add FAQs everywhere.
  6. **Check your Google Search Console.** You'll start seeing impressions for queries you never targeted. That's AI-adjacent traffic showing up.
  7. **Ask AI about yourself regularly.** "Best [service] in [city]" — do this weekly. It's free feedback on what's working.
-

# Markdown Mirrors

Give AI a clean version of every page on your site to read.

OK so this is the second piece of the AI visibility puzzle. The `llms.txt` file you built in System 2 tells AI models *what* your business is. But when AI tries to read individual pages on your site to quote you, it has to wrestle through navigation, popups, scripts, cookie banners, and all the chrome a normal website has for humans. Markdown mirrors solve that.

A markdown mirror is a clean, plain-text version of every page on your website that lives at a predictable URL — just add `/index.md` to any page. Anthropic's docs site does this. Stripe's docs site does this. Almost no small businesses do this yet. That's the opportunity.

## OVERVIEW

### What You're Building

One Python script that walks your website folder and, for every HTML page, generates a sibling `index.md` file containing only the actual content. Plus a small Netlify config tweak so those `.md` files render as plain text in browsers (instead of forcing a download), and an update to your `llms.txt` so AI knows the mirrors exist.

Walk Site Folder



Strip Chrome



Convert to Markdown



Save index.md



Serve as text/plain



Tell AI About It

#### WHY THIS STACKS WITH LLMS.TXT

llms.txt is the pitch — the elevator overview for your whole business. Markdown mirrors are the per-page detail. When AI cites you, it can quote from the right page directly. You want both. The two together turn your site into a fully AI-readable surface.

#### BEFORE YOU START

## What You Need

- **Claude Code** installed (the CLI tool, not claude.ai)
- **A static website** hosted on Netlify, Vercel, GitHub Pages, Cloudflare Pages, or any host that lets you upload a config file. If you're on Squarespace or Wix, you'll need to ask Claude Code for the workaround for your specific host.
- **Your website code in a folder** you can hand to Claude Code. If your site is in a GitHub repo, that's perfect.
- **Python 3 installed** with the `beautifulsoup4` and `markdownify` packages (Claude Code will install these for you if you ask).

#### TIME + COST

End to end this took me about an hour, including testing. Cost: \$0 ongoing. The script is one Python file you run whenever your site updates. You can wire it into a pre-commit hook later if you want it fully automated.

#### STEP 1

## Build the Generator Script

Open Claude Code in your website project folder. Paste this prompt. It will write a Python script that walks every `index.html` file in your site, strips the chrome, and writes a clean markdown version next to it.

#### PASTE THIS INTO CLAUDE CODE

```
I want to build markdown mirrors for every page on my website. The goal is to give AI tools (ChatGPT, Claude, Perplexity) a clean version
```

of each page they can read without wrestling with HTML, scripts, or chrome.

Write me a Python script that:

1. Walks my website folder and finds every index.html file (skip 404s and any /thanks/ pages that are noindex)
2. For each page, parses the HTML with BeautifulSoup
3. Strips out: nav, footer, scripts, styles, noscripts, chat widgets, GHL/HubSpot widgets, iframes, and any element with a class matching nav, footer, cta-split, or starting with ghl
4. Drops empty div/span wrappers that have no text content
5. Converts the remaining body HTML to clean markdown using the markdownify package
6. Cleans the markdown output: collapses 3+ blank lines, strips standalone "01" "02" step numbers, removes bullet separator characters, removes empty image alt tags
7. Writes the result to {page\_dir}/index.md with frontmatter at the top:

```
---  
title: [page title from <title>]  
description: [meta description]  
url: [canonical URL]  
last_updated: [today's date]  
---
```

8. Prints a summary at the end showing how many pages were generated

Save it as generate\_markdown\_mirrors.py in my scripts folder. Make it re-runnable so I can rerun it whenever my site updates. After you write it, run it once and show me the output.

### WHY FRONTMATTER MATTERS

The frontmatter at the top of each markdown file gives AI extra context — title, description, the canonical URL. When AI quotes the page, it knows where the content came from. Don't skip it.

### SHORTCUT

If you'd rather skip building the script yourself, the working Python generator is ready to copy inside the free community at [brycenwood.com/community](https://brycenwood.com/community). Paste it in, run it, done.

## STEP 2

# Configure Your Host to Serve .md as Plain Text

By default, most web hosts treat .md files as a download — when you visit one in a browser, it downloads instead of displaying. That's bad for two reasons: humans can't preview them, and some AI fetchers stumble on the content type. You want them to render inline as plain text.

### PASTE THIS INTO CLAUDE CODE

```
I'm hosted on [YOUR HOST – Netlify / Vercel / Cloudflare Pages / GitHub Pages / Squarespace / etc].
```

```
I just generated markdown mirror files at {page}/index.md across my site. I need every .md file to be served with Content-Type: text/plain so they render in the browser instead of triggering a download.
```

```
Show me exactly which config file I need to edit on my host and the exact lines to add. Then explain how to test it after deploy.
```

```
For Netlify, the answer is the _headers file. For Vercel, it's vercel.json. For Cloudflare Pages, it's _headers also. Walk me through whichever one applies to me.
```

### FOR NETLIFY USERS SPECIFICALLY

```
Add this to your _headers file:
```

```
/*.md
  Content-Type: text/plain; charset=utf-8
  Cache-Control: public, max-age=3600
  X-Robots-Tag: index, follow
```

## STEP 3

# Tell AI the Mirrors Exist

You already have an llms.txt file from System 2. Add a section to it that lists every markdown mirror URL so AI fetchers know where to find them.

#### PASTE THIS INTO CLAUDE CODE

Open my llms.txt file. Add a new section called "Markdown Mirrors" right above the "What Makes Us Different" section. Inside that new section, list every index.md URL on my site. The format should be:

```
## Markdown Mirrors (Clean AI-Readable Versions)
```

```
Every page on this site has a plain markdown mirror. Add /index.md to any URL to get the clean content without navigation, scripts, or layout chrome.
```

```
- https://yourdomain.com/index.md  
- https://yourdomain.com/about/index.md  
- https://yourdomain.com/services/index.md  
(etc, one per page)
```

```
Generate the full list automatically by reading my website folder. Skip any noindex pages (thanks pages, 404s).
```

#### STEP 4

## Test It

After you push the changes and your site redeploys, run through this test:

1. **Open one of your mirror URLs in a browser** — for example `yourdomain.com/business-wraps/index.md`. It should render as clean plain text, not download.
2. **Curl the URL from terminal** — `curl -I https://yourdomain.com/business-wraps/index.md`. The Content-Type header should say `text/plain`.
3. **Paste a mirror URL into ChatGPT** — ask "what does this business do?" ChatGPT will fetch the URL and return a clean summary. If it nails it, your mirrors work.
4. **Paste the same URL into Claude or Perplexity** — same test. Verify all three platforms can read it cleanly.

#### IF THE TEST FAILS

If the .md still downloads instead of displaying, the Content-Type header didn't take. Check that `_headers` is in your site's root and the format is exact (the indentation matters on Netlify). Redeploy and test again.

## STEP 5

# Keep It Updated

Whenever you update a page on your site, you'll want to regenerate that page's markdown mirror so AI is reading the latest version. Two ways to handle this:

## Option A — Manual rerun

Just run `python3 generate_markdown_mirrors.py` whenever you update content. Takes about 2 seconds. Commit + push. Done.

## Option B — Automated via pre-commit hook

### PASTE THIS INTO CLAUDE CODE

I want my markdown mirrors to regenerate automatically whenever I commit changes to my website. Set up a pre-commit hook that:

1. Runs `generate_markdown_mirrors.py` every time I git commit
2. Adds any newly generated/updated `.md` files to the commit automatically
3. Doesn't fail the commit if the script has any non-fatal warnings

Walk me through the `.git/hooks/pre-commit` setup for my repo. Make it so I never have to think about it again.

## MY RESULTS

# What Happened After I Shipped This

- **27 markdown mirrors** generated across my site in about 2 seconds
- **Total time to build + ship:** ~1 hour from "what's a markdown mirror" to live in production
- **Cost:** \$0 (just the Claude Code subscription I already had)
- **ChatGPT, Claude, and Perplexity** can now read each page in one clean fetch with no HTML noise
- **The video about it** hit my biggest reach yet within hours of posting

## REAL TALK

This is one of those "almost nobody is doing this yet" plays. In 12 months it'll be standard. Right now you can ship it in an hour and be early. That's the whole game with AI visibility — be early, be specific, give the machines something clean to quote.

## TIPS

### Things I Learned Building This

1. **Keep the markdown clean.** If your generator leaves visual chrome like "01" "02" or bullet separator characters in the output, AI will get tripped up. Strip aggressively.
2. **Frontmatter helps AI.** The title/description/URL block at the top of each .md file gives the AI context. Don't skip it.
3. **Don't change the URL pattern later.** Once you decide on /page/index.md or /page.md, stick with it. Changing breaks any AI fetcher that already learned where to look.
4. **Update llms.txt every time you add a new page.** Otherwise AI won't know the new mirror exists. Or wire that into your generator script.
5. **Test with all three: ChatGPT, Claude, Perplexity.** They have different fetchers and sometimes one will work while another won't. You want all three reading you.
6. **Don't gate the .md files with login or paywall.** The whole point is for AI to read them freely. If you need gating, gate the HTML version, not the markdown.

# Sitemaps + Google Search Console

Tell Google which pages matter. Then see your actual ranking data.

This is the third piece. `llms.txt` tells AI what your business is. Markdown mirrors let AI read your pages clean. A sitemap tells *Google* which pages exist, which ones matter most, and when they were last updated. Without it, Google is guessing. With it, you're handing Google a map of your entire site with priorities marked.

Then Google Search Console lets you see exactly what's happening — which keywords you're ranking for, what position you're in, and where to focus next. Most small businesses never set this up. That's the gap.

## OVERVIEW

### What You're Building

An XML sitemap that lists every important page on your site with priority scores and update frequencies. Plus a Google Search Console setup so you can see your actual ranking data — which keywords are bringing people to your site, what position you're in, and how your click-through rate compares.

Generate Sitemap



Set Priorities



Deploy



Submit to Google



Monitor in GSC

llms.txt is for AI chatbots. Markdown mirrors are for AI page readers. The sitemap is for Google's crawler. Together, these three files cover every way a machine discovers and reads your business online. You're not choosing between traditional SEO and AI visibility — you're doing both with three text files.

## BEFORE YOU START

# What You Need

- **Claude Code** installed (the CLI tool, not claude.ai)
- **Your website code in a folder** that Claude Code can access
- **A Google account** for Search Console (free)
- **Access to your domain's DNS** (Cloudflare, GoDaddy, Namecheap, etc.) — needed for one verification step

## TIME + COST

Sitemap generation: 5 minutes. Google Search Console setup: 10 minutes. Total: 15 minutes. Cost: \$0. You'll start seeing ranking data within 2-3 days.

## STEP 1

# Generate Your Sitemap

Open Claude Code in your website project folder. Paste this prompt. It will create an XML sitemap that lists every page on your site with the right priorities.

## PASTE THIS INTO CLAUDE CODE

I need an XML sitemap for my website. Generate a sitemap.xml file that:

1. Lists every important page on my site (skip 404 pages, thank-you pages, and any noindex pages)
2. Sets the homepage priority to 1.0
3. Sets main service pages to 0.9
4. Sets other pages (about, contact, gallery) to 0.8
5. Sets blog posts or secondary pages to 0.7
6. Uses today's date for lastmod on all pages

7. Sets `changefreq` to "weekly" for the homepage and "monthly" for everything else

8. Follows the standard XML sitemap protocol at [sitemaps.org](https://sitemaps.org)

Save it as `sitemap.xml` in the root of my website folder. After you create it, show me the full file so I can review it before deploying.

#### WHY PRIORITIES MATTER

The priority score (0.0 to 1.0) tells Google which pages you consider most important. Your homepage and main service pages should be highest. This doesn't guarantee Google will rank them higher, but it tells the crawler where to spend its time. Most auto-generated sitemaps set everything to 0.5 — that's like telling Google "nothing matters more than anything else."

## STEP 2

### Deploy and Verify

Push your `sitemap.xml` to your live site. Then verify it's accessible.

#### PASTE THIS INTO CLAUDE CODE

My `sitemap.xml` is deployed. Help me verify it's working:

1. Fetch `https://[MY-DOMAIN]/sitemap.xml` and confirm it returns valid XML
2. Check that every URL in the sitemap actually resolves (no 404s)
3. Count the total pages listed
4. Show me any issues

Also check if I have a `robots.txt` file. If I do, make sure it includes a `Sitemap:` line pointing to my sitemap. If I don't have a `robots.txt`, create one that:

- Allows all crawlers
- Explicitly allows AI crawlers (GPTBot, ClaudeBot, PerplexityBot, Google-Extended)
- Points to my sitemap URL

## STEP 3

# Set Up Google Search Console

This is how you see your actual Google ranking data. Most business owners have never looked at this — they just guess whether SEO is working. GSC shows you exactly.

1. **Go to** `search.google.com/search-console`
2. **Click "Add property"** and choose "Domain" (not "URL prefix")
3. **Enter your domain** (e.g. `yourbusiness.com` — no `https`, no `www`)
4. **Google will give you a TXT record** to add to your DNS. Copy it.
5. **Go to your DNS provider** (Cloudflare, GoDaddy, etc.) and add the TXT record
6. **Go back to GSC** and click "Verify." It may take a few minutes for DNS to propagate.
7. **Once verified, go to Sitemaps** in the left sidebar and submit your sitemap URL

## IF YOU GET STUCK ON DNS VERIFICATION

This is the one step that trips people up. If you're not sure where your DNS is hosted, ask Claude Code: "Where is my domain's DNS managed?" and give it your domain name. It can usually figure it out. Or just search "[your registrar] add TXT record" — every provider has a guide.

## STEP 4

### Read Your Data

After 2-3 days, GSC will start showing data. Here's what to look for:

1. **Go to Performance** in the left sidebar
2. **Check all four boxes** at the top: Total clicks, Total impressions, Average CTR, Average position
3. **Look at the Queries table** — these are the keywords people are searching when they find you
4. **Sort by Impressions** — high impressions + low clicks = your biggest opportunity (people see you but don't click)
5. **Sort by Position** — anything between position 5-15 is a "quick win" because you're close to page one

#### THE QUICK WIN FORMULA

Find keywords where you rank position 5-15 with high impressions. These are keywords where Google already thinks you're relevant — you just need a small push to get onto page one. That push is usually: better meta title, better meta description, and a clean page structure. Ask Claude Code to optimize those specific pages for those specific keywords.

#### IF YOU WANT HELP READING YOUR GSC DATA

Post a screenshot of your Performance tab in the free community ([brycenwood.com/community](https://brycenwood.com/community)) and I'll tell you exactly which keywords to target first. Seeing the data is easy — knowing what to do with it is where people get stuck.

#### MY RESULTS

## What Happened After I Shipped This

- **Position 6 for "car wraps near me"** — the most valuable keyword in my industry, on page one of Google
- **Position 1 for "car wrap places near me" and "car wrap services"**
- **131 clicks from Google** in the first month of tracking
- **334 total keywords** my site now ranks for (most I didn't even know about)
- **Total time to set up:** 15 minutes for the sitemap + GSC, zero dollars

#### REAL TALK

The sitemap alone didn't do this. The llms.txt + markdown mirrors + sitemap together is what moved the needle. Google is starting to reward sites that are clean, structured, and AI-readable. If you've already built Systems 1 and 2 from this playbook, the sitemap is the last piece that ties it all together for Google's crawler. Three files. That's the whole system.

#### TIPS

## Things I Learned

1. **Update your sitemap when you add pages.** If you add a new service page or blog post, regenerate the sitemap. Otherwise Google won't know about it for weeks.
2. **Don't obsess over position daily.** Rankings fluctuate. Check weekly, not hourly. The trend line matters more than any single day.
3. **GSC data is delayed 2-3 days.** What you see today is from 2-3 days ago. Don't panic if numbers look weird — wait for the data to catch up.
4. **Your meta title and description matter more than you think.** GSC will show you CTR (click-through rate). If you have high impressions but low CTR, your title and description aren't compelling enough. Ask Claude Code to rewrite them.
5. **The robots.txt + sitemap combo is standard practice.** But adding llms.txt and markdown mirrors on top of it is what puts you ahead. Most businesses stop at the sitemap. You're not stopping there.

# YouTube Autopilot

Auto-cross-post every video to YouTube with AI-optimized metadata. One prompt builds it.

---

This is the system that changed how I think about content. You're already filming videos. You're already posting on Instagram and TikTok. But if you're not on YouTube, you're leaving the biggest search engine in the world on the table.

The problem is nobody wants to manually upload to another platform every day. And if you just paste your Instagram caption onto YouTube, it dies. YouTube's algorithm is search-based. Instagram's is engagement-based. Same caption doesn't work on both.

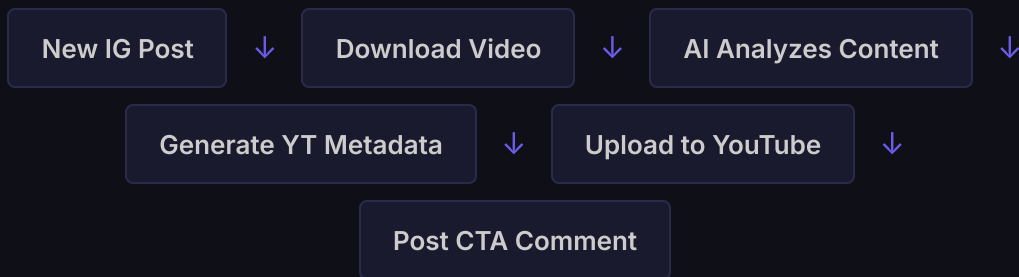
This system solves both problems. It watches your Instagram, grabs every new video, uses AI to generate a completely new title, description, and tags optimized for YouTube search, and uploads it as a Short. Automatically. You set it up once and never think about YouTube again.

---

## OVERVIEW

### What You're Building

Two Python scripts that work together. The first watches your Instagram via the official API, downloads new videos, and generates YouTube-optimized metadata using Claude. The second uploads them to your YouTube channel with the optimized titles, descriptions, and tags, and drops a CTA comment on every video.



#### WHY THIS IS DIFFERENT FROM REPURPOSE.IO OR MANUAL CROSS-POSTING

Apps like Repurpose.io just mirror your video with the same caption. That's not cross-posting, that's copy-pasting. YouTube's algorithm needs searchable titles, keyword-rich descriptions, and relevant tags to surface your content. This system generates all of that automatically using AI that actually analyzes your video content and researches what's ranking on YouTube for that topic.

#### BEFORE YOU START

## What You Need

- **Claude Code** installed (the CLI tool)
- **An Instagram business or creator account** with access to the Instagram Graph API (Meta developer account, free)
- **A YouTube channel** (free, takes 2 minutes to create if you don't have one)
- **A Google Cloud project** with the YouTube Data API enabled (free tier covers everything)
- **An Anthropic API key** for Claude (usage-based, costs less than \$0.01 per video)

#### TIME + COST

Setup: about 1-2 hours including API credentials. Ongoing cost: under \$1/month for the AI metadata generation. The YouTube API and Instagram API are both free. Once it's running, it takes zero time because it's fully automated.

#### STEP 1

# Set Up YouTube API Access

You need a Google Cloud project with the YouTube Data API enabled. This gives your script permission to upload videos and post comments on your channel.

## PASTE THIS INTO CLAUDE CODE

I need to set up YouTube API access so I can upload videos programmatically. Walk me through:

1. Creating a Google Cloud project (or using an existing one)
2. Enabling the YouTube Data API v3
3. Creating OAuth 2.0 credentials (Desktop app type)
4. Downloading the credentials.json file
5. Where to save it in my project

I want to upload YouTube Shorts and post comments on my own channel. I don't need to access anyone else's channel.

## THE OAUTH DANCE

The first time you run the upload script, it will open a browser window asking you to log into Google and authorize the app. Pick the correct YouTube channel on the consent screen. After that, it saves a token and never asks again (it auto-refreshes). If you have multiple YouTube channels, make sure you select the right one.

---

## STEP 2

# Set Up Instagram Graph API Access

Your script needs to read your Instagram posts to know when you've posted new videos. This uses Meta's official Graph API.

## PASTE THIS INTO CLAUDE CODE

I need to set up Instagram Graph API access to read my posts programmatically. Walk me through:

1. Creating a Meta developer account and app (if I don't have one)
2. Adding Instagram Graph API to the app
3. Generating a long-lived access token for my Instagram account

4. Testing the token to make sure it can read my media

My Instagram handle is [YOUR HANDLE]. I only need to read my own posts, not manage other accounts.

Save the config to a JSON file with the access token and my Instagram user ID.

### STEP 3

## Build the Cross-Post System

This is the main prompt. It builds the entire system: the Instagram watcher, the AI metadata generator, and the YouTube uploader. One prompt, two scripts, fully automated.

#### PASTE THIS INTO CLAUDE CODE

I want to build a system that automatically cross-posts my Instagram Reels to YouTube Shorts with AI-optimized metadata. Build me two Python scripts:

#### SCRIPT 1: Instagram to Queue

- Check my Instagram account for new video posts using the Graph API
- Download any new videos it finds
- For each video, read my Instagram caption and use the Anthropic API (Claude) to generate:
  - A YouTube-optimized title (searchable, keyword-rich, under 80 characters, front-load the main keyword)
  - A YouTube description rewritten from scratch for search (NOT a copy of the IG caption). Include relevant keywords, a hook, key takeaways, and a CTA section at the bottom with links to my playbook and social profiles
  - 15-20 YouTube tags mixing exact-match search phrases, related topics, and broad category tags
- Save everything to a JSON queue file
- Track which posts have already been processed so it skips duplicates on re-run

#### SCRIPT 2: Queue to YouTube

- Read the JSON queue
- Upload each video as a YouTube Short (public, category: Education)
- Post a pinned comment with a CTA on each upload
- Handle YouTube quota limits gracefully (pause if exceeded)
- Save the YouTube video ID back to the queue so we know what's

uploaded

Both scripts should be re-runnable and safe to call from a cron job. Add a `--dry-run` flag to both. The Instagram script should have a `--backfill` flag to reprocess everything.

My Anthropic API key is in my `.env` file as `ANTHROPIC_API_KEY`. My Instagram config is at `_config/instagram.json`. My YouTube credentials are at `credentials.json` with token at `_config/youtube_token.json`.

### WANT THE WORKING SCRIPTS?

The exact Python scripts I use are available inside the free community at [brycenwood.com/community](https://brycenwood.com/community). Paste the prompt above to build your own version, or grab the battle-tested ones from the community.

## STEP 4

# Automate It

The scripts work manually, but the whole point is to never think about YouTube again. Set up a cron job (Linux) or launchd agent (Mac) to run them on a schedule.

### PASTE THIS INTO CLAUDE CODE

I want to automate my YouTube cross-posting system so it runs every 6 hours without me doing anything. Set up:

1. A scheduled task that runs the Instagram-to-queue script every 6 hours
2. Immediately after, runs the YouTube uploader script
3. Logs output to a file so I can check what happened
4. Doesn't run if the previous run is still going

I'm on [Mac/Linux]. Use launchd for Mac or cron for Linux. Make sure it runs even if I'm not logged in.

## MY RESULTS

# What Happened After I Shipped This

- **5,000+ views on YouTube** across 12 videos without ever opening YouTube to upload a single one
- **Every 6 hours** the system checks for new Instagram posts and uploads them automatically
- **Each video gets a unique YouTube-optimized title** instead of a pasted Instagram caption
- **Total time spent on YouTube after setup:** zero
- **Setup time:** about 2 hours including API credentials
- **Ongoing cost:** under \$1/month for the AI metadata generation

## REAL TALK

The biggest unlock here isn't the automation. It's that YouTube is a search engine, not a feed. Your Instagram videos have a 24-48 hour shelf life. YouTube Shorts can get discovered months later through search. Every video you post is a long-term asset on YouTube and a short-term play on Instagram. This system makes sure you get both without any extra work.

## TIPS

### Things I Learned

1. **YouTube titles and Instagram captions are different languages.** Instagram captions are conversational ("Comment MIRROR and I'll DM you"). YouTube titles are searchable ("How to Make Your Website Readable by AI"). The AI rewrite handles this but understand why it matters.
2. **YouTube tags have a 500-character total limit.** Don't generate 30 long-tail tags. 15-20 concise tags is the sweet spot.
3. **The YouTube Data API has a daily quota.** You get 10,000 units/day for free. Each upload costs ~1,600 units. That's about 6 uploads per day, which is more than enough for cross-posting.
4. **Post a CTA comment on every video.** YouTube comments with links actually work because YouTube doesn't suppress them like Instagram does. Use this for your playbook, community, or website link.

5. **Don't delete and re-upload.** If you want to update metadata on an existing video, use the API to update in place. This preserves your view count, comments, and watch history.
6. **YouTube Shorts get discovered through search for months.** Unlike Instagram where a post dies in 48 hours, YouTube Shorts surface in search results indefinitely. Every video is a long-term asset.

# Full Website Builder Prompt

Build a complete 40-page website in one afternoon. One prompt.  
Zero code.

---

## THE PROBLEM

### Most AI-built websites are invisible

Anyone can ask AI to build a website. You'll get a homepage and a contact form and it'll sit on the internet and do nothing. No SEO. No schema. No sitemap. No AI visibility. Google doesn't know it exists. ChatGPT can't read it. It's a digital business card that nobody sees.

I spent months figuring out what actually makes an AI-built website show up in Google, get recommended by ChatGPT, and bring in leads on its own. Then I put all of it into one prompt.

## THE PROMPT

### Paste this into Claude Code

This prompt is the bones. It tells Claude Code exactly what to build and how to structure it so your site actually gets found. But the more real information you feed it about your business, the better the results. Don't just fill in the brackets and hit enter -- give it everything. Your services, your story, your customers, what makes you different, your pricing, your location, examples of your work. The more context Claude Code has, the more specific and useful every page will be.

Replace the bracketed fields below to get started. Then keep the conversation going -- tell it more about your business and watch the site get better with every detail you add.

#### PASTE THIS INTO CLAUDE CODE

I need you to build me a complete website for my business.

Business name: [YOUR BUSINESS NAME]

What we do: [ONE SENTENCE – what you sell or offer]

Who we serve: [YOUR IDEAL CUSTOMER]

Location: [CITY, STATE or "online"]

What makes us different: [1-2 SENTENCES]

Build me:

- Homepage with hero section, services overview, testimonials section, and contact form
- Services page with individual service detail pages
- About page with our story
- Contact page with a working form
- Blog with 5 SEO-optimized articles targeting what our customers actually search for
- Case study page with 1-2 examples of our work
- Pricing page (if applicable)
- FAQ page with schema markup
- Privacy policy and terms of service
- llms.txt so AI tools like ChatGPT, Claude, and Perplexity can read and recommend our site
- Markdown mirrors (index.md) for every page so AI reads clean content
- XML sitemap
- Schema markup (JSON-LD) on every page – LocalBusiness, Service, FAQPage, Article, BreadcrumbList
- Meta titles and descriptions optimized for clicks on every page
- Mobile responsive, fast loading, modern design
- robots.txt that allows all AI crawlers

Deploy-ready. No frameworks. Pure HTML and CSS. Every page should be its own folder with an index.html inside it.

#### WHAT THIS BUILDS

## Here's what you get from one prompt

This isn't just a website. It's a website that's built to be found. And the more you tell Claude Code about your business after the initial build, the deeper it goes -- rewriting blog posts with your actual expertise, building FAQ pages from real customer questions, creating case studies from your real projects.

- **30-40+ pages** depending on your services and blog topics

- **SEO on every page** - meta titles, descriptions, heading structure, internal linking
- **Schema markup** - Google Rich Results for your business, services, FAQ, and articles
- **AI visibility** - llms.txt + markdown mirrors so ChatGPT and Perplexity can read and recommend you
- **XML sitemap** - tells Google exactly which pages to index
- **Blog posts** - targeting real keywords your customers search for
- **Mobile responsive** - looks good on every device
- **No monthly fees** - deploy free on Cloudflare Pages, Netlify, or GitHub Pages

## DEPLOY IT

# Get it live in 5 minutes

Once Claude Code builds your site, you need to put it on the internet. All of these are free:

## Option A: Cloudflare Pages (recommended)

### PASTE THIS INTO CLAUDE CODE

Help me deploy this website to Cloudflare Pages. Walk me through:

1. Creating a Cloudflare account (if I don't have one)
2. Connecting my GitHub repo
3. Setting up the build
4. Connecting my custom domain

## Option B: Netlify

### PASTE THIS INTO CLAUDE CODE

Help me deploy this website to Netlify. Walk me through connecting my GitHub repo and setting up my custom domain.

Both options are free, auto-deploy when you push changes, and include SSL (the lock icon in the browser).

## MAKE IT YOURS

# Customize after the first build

The prompt gives you the structure, but the magic happens when you keep feeding Claude Code more about your business. The first build gets you 90% of the way there. Then you refine by telling it what to change, what to add, what to go deeper on:

## PASTE THIS INTO CLAUDE CODE

```
I want to customize my website. Here's what I want to change:  
- [DESCRIBE WHAT YOU WANT DIFFERENT - colors, layout, new pages,  
different copy, etc.]
```

```
Keep everything else the same. Don't break the SEO, schema, or AI  
visibility files.
```

That last line is important. It tells Claude Code to preserve all the invisible infrastructure (schema, sitemap, llms.txt, markdown mirrors) while changing the visual design. Most people rebuild from scratch when they want changes. You just describe what you want different.

## THE SECRET TO A GREAT AI-BUILT WEBSITE

The prompt is the skeleton. Your business knowledge is the muscle. The more you tell Claude Code -- your actual customer stories, your real process, the questions people always ask you, why you started the business -- the more it sounds like YOU wrote every page, not an AI. Don't hold back. Dump everything you know into the conversation.

## REAL RESULT

I used this exact approach to build brycenwood.com - 40+ pages, deployed in one afternoon, ranking on Google within days. Blog posts, case studies, industry pages, guides, pricing, contact form, full SEO, AI visibility. Zero code written. Zero dollars spent on a developer.

# That's the Whole System.

Five systems. Zero ad spend. Zero coding experience. This is the exact stack that took my business from invisible to ranked on page one of Google, recommended by AI, auto-cross-posted to YouTube, and built on a 40-page website that cost nothing.



## System 1: lms.txt

AI chatbots now know what your business does, where you are, and who you serve. They recommend you by name.



## System 2: Markdown Mirrors

AI can read every page on your site cleanly. No HTML wrestling, no missed content, no hallucinated details.



## System 3: Sitemaps + GSC

Google knows exactly which pages matter. You can see your actual rankings, keywords, and where the quick wins are.



## System 4: YouTube Autopilot

Every video auto-cross-posts to YouTube with AI-optimized metadata. Search-friendly titles, keyword-rich descriptions, zero manual work.



## System 5: Full Website Builder

A complete 40-page website with SEO, schema, blog posts, and AI visibility. One prompt. One afternoon. Zero dollars.

**Want help implementing this?**

I run a free community where I share the ready-made scripts, answer questions, and do live builds. Post your site, get feedback, watch me build these systems for real businesses in real time. No upsell, no trial — just builders helping builders.

[Join the Free Community](#)

## Want me to build this for your business?

I'm taking on a small number of custom builds where I interview your business and build the entire system for you. Same tools, same stack — done in 2 weeks.

[DM @brycenwood.ai](#)

*No coding experience was used in the making of this playbook. Or the systems it describes.*

---

### Built by Brycen Wood

[IG: @brycenwood.ai](#)

[TikTok: @brycenwood1](#)

[Web: summitwrapsandgraphics.com](#)

I document everything I build with Claude Code as a business owner with zero coding experience. Follow along for more systems, tools, and mistakes.

[Community: brycenwood.com/community](#)

If you build any of these, post it in the community or DM me. I want to see them work for you.

