

SEO and GEO for your website with Claude

The promise: Within a week your site will be readable and understandable to both Google and AI engines (ChatGPT, Claude, Gemini, Perplexity), with checks you can run yourself.

Who this is for

For managers of nonprofits, small businesses and local authorities who have a site on WordPress, Wix, Lovable or similar, and someone who can edit it. You don't need to be a developer. You need Claude or ChatGPT, edit access to the site, and about an hour or two per step. At the end there's a short section for people who work with Claude Code.

The problem in three lines

People already ask chatbots "which organization does X in area Y" and get an answer without ever opening Google. Many sites look great in a browser, but AI engine bots see an almost empty page, because the content loads with JavaScript. And even when they see the text, they have no structured way to understand who you are, what you do and for whom.

SEO means being understood and found on Google. **GEO** (Generative Engine Optimization) means being understood and cited by AI answer engines. Most of the work is shared between the two.

Before you start: what to expect and what not to

Worth knowing up front, so you're not disappointed:

- **FAQ with an expanded result on Google:** since 2023 Google shows this expanded result almost only for government and health sites. For most sites it won't appear.
- **Stars next to an app or software product:** these require a price and real reviews. Inventing ratings is not allowed, and Google penalizes it.
- **So why do it?** Because the markup (Schema) helps Google and AI engines understand the page: who the organization is, what the service is, what the answers to common questions are. That's what raises the chance they'll cite you correctly. Don't promise anyone stars.

One rule runs through every step: **only the truth, and only what appears on the page.** No invented reviews, prices, clients, awards, addresses, founding years or numbers. If a data point is missing, leave it out.

The method in 7 steps

● Step 1: Check what a bot actually sees

What to do: Check whether the page's main text, title and description appear in the HTML the server sends, before JavaScript runs.

How: Two ways:

- **Without a terminal:** In Chrome, right-click the page → "View Page Source". Search (Ctrl+F) for a sentence from the main text. If it isn't there, most AI bots don't see it.
- **With a terminal:** The command `curl -sL https://example.org` downloads the page the way a bot that doesn't run JavaScript sees it. The prompt below explains what to look for in the output.

What should appear: the main text, `<title>`, the description (`meta description`), the canonical link (`canonical`), and every Schema block.

Common mistake: Checking with "Inspect" (developer tools) instead of "View Source". Inspect shows the page after JavaScript has run, meaning what you see, not what the bot sees.

● Step 2: Fix visibility for bots

What to do: If the page came out empty in step 1, make sure the content arrives in the HTML straight from the server (static HTML or server-side rendering).

How:

- **WordPress:** The content is usually already in the HTML. Check that your page-builder plugin doesn't load the text only via JavaScript.
- **Wix:** Sites are rendered on the server. Check with View Source anyway.
- **Lovable and React sites (SPAs):** This is where the problem is most common. Ask Lovable (or whoever builds the site) to put the Schema, title and description into the `index.html` file at build time, or to switch to server-side rendering or prerendering. Schema injected only at runtime (for example via React Helmet) is invisible to most AI bots.

Common mistake: Adding Schema through a plugin that injects it with JavaScript and assuming it works because Google's testing tool sees it. Google runs JavaScript. Most AI bots don't.

● Step 3: A definition sentence and a visible FAQ

What to do: Near the top of the page, add one sentence that says what you are and who you serve, plus a questions-and-answers section that appears on the page itself.

How:

- A definition sentence, for example: "[Name] is a nonprofit that runs [what] for [whom] in [where]." AI engines quote exactly this kind of sentence.

- 5 to 8 questions people actually ask you (by phone, email, WhatsApp), with direct answers of 2 to 4 sentences.
- Concrete facts: service areas, languages, how to get in touch, who it's for and who it isn't for.
- Exactly the same name everywhere: on the site, on LinkedIn, on Facebook, in the official registration.

Common mistake: Writing an FAQ that exists only inside the Schema and isn't shown on the page. That goes against Google's guidelines. The questions and answers in the Schema must be identical to what's written on the page.

● Step 4: Schema (JSON-LD) built from the page text

What to do: Add a `<script type="application/ld+json">` block to the page that describes the organization and the page, built from the same text that's displayed.

How: The types we use in the systems we've built:

- One `Organization` for the whole site: name, site URL, logo, and `sameAs` links to real profiles (LinkedIn, Facebook). A fixed identifier, for example `https://example.org/#organization`.
- `WebSite` for the site.
- By page type: `ProfessionalService` for a service, `SoftwareApplication` or `Product` for a product, `Person` for a person's about page (only facts that appear on the site), `BlogPosting` with author and date for a post, `FAQPage` only where the questions are visible, `BreadcrumbList` for inner pages.
- Connect them all in one `@graph` with `@id` identifiers.
- `inLanguage` matching the language the page is shown in (`en` for English, `he` for Hebrew).

On WordPress, paste the block through a plugin that allows code in the page head, or use an existing SEO plugin and fill in the gaps. Wix has a "Structured Data" setting for each page. On Lovable, ask in the prompt to put the block into `index.html`.

Common mistake: Adding `aggregateRating`, `offers` or `address` "because it'll help". If it isn't real and doesn't appear on the page, it doesn't go in.

● Step 5: robots.txt and sitemap

What to do: Make sure you're not blocking AI engine bots, and that you have a sitemap.

How: Open `https://example.org/robots.txt`. Check that there's no `Disallow: /` under any of these, unless you decided to block them on purpose: `GPTBot`, `OAI-SearchBot`, `ChatGPT-User`, `ClaudeBot`, `Claude-SearchBot`, `PerplexityBot`, `Google-Extended`, `Applebot-Extended`. At the end of the file there should be a line `Sitemap: https://example.org/sitemap.xml`, and the sitemap file should include all public pages.

A minimal example:

User-agent: *

Allow: /

Sitemap: <https://example.org/sitemap.xml>

Common mistake: A theme or security plugin that blocks "unknown bots" and takes out every AI bot along the way. Check the actual file, not the setting in the dashboard.

● Step 6: llms.txt

What to do: Add a short Markdown text file at the root of the site that sums up who you are and which pages matter.

How: The file lives at <https://example.org/llms.txt>. One summary line, followed by a list of key pages with a short description of each. It's cheap and takes ten minutes. It isn't an official standard, and right now there's no guarantee every engine reads it, so it's an optional step and not a replacement for steps 1 to 5.

Common mistake: Writing things in llms.txt that don't appear on the site. The file is a summary, not a separate marketing channel.

● Step 7: Meta tags and a final check

What to do: Make sure every page has a unique title and description, and run the validation tools.

How:

- For every page: a unique `title` and `meta description`, `og:image`, `canonical`, and in the `<html>` tag a `lang` attribute for the page's language (`lang="en"` ; a Hebrew or Arabic site also needs `dir="rtl"`). A multilingual site: `hreflang` .
- After publishing, run the live page through Google's Rich Results Test (search.google.com/test/rich-results) and the Schema Markup Validator (validator.schema.org). Fix every error. Warnings about optional fields don't have to be fixed.
- Repeat step 1 with the live URL.

Common mistake: Checking only the homepage. Every page template (service, post, about) needs its own check.

For a small or local business: three things that make the difference

If you're a business with customers in a specific area (a clinic, an office, a shop, a service provider), three extra steps matter as much as the site:

1. **Google Business Profile.** When someone asks Google or a chatbot for "an accountant in Petah Tikva", the information usually comes from this profile and from Maps, not from your site. Make sure the

name, address, phone, opening hours and category are up to date and exactly match what's on the site.

2. **LocalBusiness Schema.** Instead of a generic Organization, use the type that fits the business (for example `LocalBusiness`, `ProfessionalService`, `Dentist`, `Store`) with address, phone, hours and service area. Only real details that also appear on the page.
3. **Real reviews, in the right place.** Ask happy customers to leave a Google review. Don't write reviews yourself, and don't mark up a rating in Schema that isn't backed by reviews shown on the site.

Bonus: the same business name, the same one-sentence description and the same contact details everywhere (site, Google, Facebook, LinkedIn, directories). AI engines cross-check sources, and consistency is what makes them "believe" you.

Ready-made prompts

1. FAQ from your existing content

Here is the text of the page [URL] on our site: [paste text].

And here are questions people actually ask us: [list, if any].

Write 5 to 8 questions and answers for the page. Each answer 2 to 4 sentences, direct, in plain language and in the tone of the existing text.

Use only information that appears in the text. If a question needs information we don't have, list it separately under "Missing information" and don't answer it.

Also write one definition sentence: who we are, what we do and for whom.

2. JSON-LD from the page text

Build one JSON-LD block (schema.org) for this page, inside `<script type="application/ld+json">`.

Site URL: [https://...]. Page URL: [https://...]. Language: en.

The page text as displayed: [paste]. The questions and answers visible on the page: [paste].

Official profiles: [LinkedIn, Facebook...]. Logo: [URL].

Use an `@graph` with Organization (`@id: <site>/#organization`), WebSite, and the type that fits the page. FAQPage only with the questions I pasted, word for word.

Rules: don't invent reviews, ratings, prices, an address, a founding year or clients. Omit any data point I don't have, and after the code list the "missing data".

Make sure the JSON is valid, and that every `<` inside a string is written as `\u003c`, so it cannot break out of the script tag.

3. What a bot sees (curl output)

I ran `curl -sL` on my page, and here's the output: [paste the output, or the first 300 lines].

Check and report in a table whether these appear: 1) the page's main text (sample sentence: "[sentence from the page]"), 2) title, 3) meta description, 4) canonical, 5) lang (and dir, if relevant) on the html tag, 6) application/ld+json blocks, and how many.

If there's JSON-LD, check that it's valid and that its questions are identical to the questions shown in the text.

If the main text is missing, explain in one sentence why and what the options are for fixing it on [WordPress / Wix / Lovable].

4. llms.txt

Write an llms.txt file for the site [URL], in Markdown format:

First line: # [organization name]. Then a quote line (>) with a one-sentence summary.

Then a "## Key pages" section with a list of links: - [page name](URL): one-sentence description.

The pages and content: [paste a list of pages and a short text from each].

Only information that appears on the site. No superlatives.

5. robots.txt check

Here is my site's robots.txt: [paste].

Does it block any of these bots: GPTBot, OAI-SearchBot, ChatGPT-User, ClaudeBot, Claude-SearchBot, PerplexityBot, Google-Extended, Applebot-Extended? Is there a Sitemap line?

Suggest a corrected version that changes only what's needed, and explain each change in one line.

For people who work with Claude Code

If your site is a repo (Lovable syncs to GitHub, for example), you can let Claude Code do most of the work. This is how we work:

1. **Always a new branch.** Never work on `main`. At the end, open a Pull Request, and someone reviews and merges it.
2. **One source of truth.** The JSON-LD is built from the same data the page displays (the same text file or the same array of questions), so the text and the Schema don't drift apart over time.
3. **Verification before the PR:** build the site, run `curl` on the resulting HTML, extract every `application/ld+json` block and make sure it parses as valid JSON, and that the number of questions and their text match the visible ones.
4. **An automated test:** add a test that renders the page and checks that the JSON-LD exists, is valid and matches the text. That way a future text change doesn't silently break the markup.
5. **In the PR description:** what was added, how to verify after publishing (Rich Results Test and `validator.schema.org` on the live URL), and open questions about missing data.

A starter prompt for Claude Code:

Run an SEO/GEO audit of the site in this repo, without changing any code. Check: whether the HTML the server returns (without JavaScript) includes the text, title, description, canonical and JSON-LD; which schema.org types exist and which are missing; robots.txt against GPTBot, ClaudeBot, PerplexityBot, Google-Extended; sitemap.xml; llms.txt; lang and dir.

Return a prioritized list: what's missing, impact, effort. Then, on a new branch, add only what can be grounded in text that appears on the site. Don't invent reviews, prices or data. Run the build and tests before the PR.

One-page checklist

Visibility to bots

- ☒ View Source or curl shows the main text
- ☒ title, meta description and canonical in the HTML
- ☒ JSON-LD in the HTML, not injected only by JavaScript
- ☒ lang set on the `<html>` tag (plus `dir="rtl"` for right-to-left languages)

Content

- ☒ Definition sentence near the top of the page: who, what, for whom
- ☒ 5 to 8 questions and answers visible on the page
- ☒ The same organization name on every channel

Schema

- ☒ One Organization with @id, logo and sameAs
- ☒ WebSite
- ☒ A fitting type for each page template
- ☒ FAQPage identical word for word to the visible questions
- ☒ No ratings, prices or data that don't appear on the page

Root files

- ☒ robots.txt doesn't block AI bots
- ☒ Sitemap line in robots.txt
- ☒ sitemap.xml with all public pages
- ☒ llms.txt (optional)

Checks

- ☒ Rich Results Test on the live URL
- ☒ validator.schema.org on the live URL
- ☒ Every page template checked, not just the homepage

How you know it's working

- **Right away:** curl or View Source shows the text and the JSON-LD, and both validation tools pass without errors.
 - **After a few weeks:** in Google Search Console, check the search performance report for more impressions on queries about your service. The "Enhancements" reports show whether Google detected your structured data.
 - **AI engines:** once a month, ask ChatGPT, Claude, Gemini and Perplexity the same 5 questions a customer would ask ("who offers X in Y?", "what is [your name]?"). Note whether you were mentioned, whether the description is accurate, and whether your site was cited as a source. Results vary between runs, so look at the trend over several months rather than a single check.
 - **Server logs (if you have access):** whether GPTBot, ClaudeBot and PerplexityBot visit your pages and get a 200 status code.
 - **Traffic:** in your analytics tool, whether visitors arrive from chatgpt.com, perplexity.ai and similar.
-

Want us to do this with you? aviad@gindiinnovation.com · gindiinnovation.com