

How I Built My One-Page Website

From a written outline to a live site on my own domain, using Claude

When my role at Arctic Shores was coming to an end, I wanted one page I could send to people in my network. Something that clearly explained what I do, what I'm looking for and, most importantly, how they could help. A CV doesn't do that. So I built simonhawtin.com.

I didn't write any code. I wrote the content in plain English, then built, edited and published the whole thing with Claude. This guide walks you through the process I followed, from a rough outline to a live site on my own domain, so you can build your own version.

What I used

Tool	What it did	Cost
Claude Pro	My Claude subscription. It gives you more usage than the free plan, plus Cowork and the connectors to other tools.	\$20/month, or \$17/month billed annually*
Claude Cowork	Claude inside the desktop app. It can work with files on my computer, pull from my Google Drive and publish pages, so it did the building for me.	Included with Claude
Netlify	Hosts the website and puts it on the internet under my own domain.	Free plan*
My domain	simonhawtin.com. I already owned it (it's managed at GoDaddy).	What I already paid for it
Google Drive	Where my master CV lives. Claude read it through the Google Drive connector.	Free

* Prices as listed on claude.com/pricing and netlify.com/pricing in September 2026. Check them before you sign up, as they change.

First: what Netlify actually is

A website is just a set of files: the page itself, images and anything you link to, like a PDF. For people to see it, those files need to sit on a computer that is always on and connected to the internet. That's called hosting, and it's what Netlify does.

You give Netlify your site's files, and it gives you a web address and serves your page to anyone who visits, quickly, from anywhere in the world. It also handles the padlock (HTTPS) that tells browsers your site is safe.

Why I chose it

- **It's free for a site like this.** The free plan includes custom domains and HTTPS.
- **No code or technical setup.** You can literally drag a folder onto a web page to publish it.
- **Every version is saved.** If an update goes wrong, you click an older version and it's live again in seconds.
- **Claude can connect to it.** Once linked, I just tell Claude to push a change live.

Step 1: Write the outline yourself

This is the part I'd never hand to AI. Before opening Claude I wrote down, in my own words, what the page needed to have on it:

- A short summary of me, written by me
- A few highlighted achievements and a brief tour of my experience
- What I'm looking for: the role, the working pattern and the type of employer
- How people can help, with a message they can copy and send to their contacts
- A booking link and my LinkedIn
- My pro-bono offer for early-stage SaaS founders

I wrote my summary and my experience in full, in my own voice. Where I didn't have the detail to hand, I left placeholders like "achievements here were x, y and z" for Claude to fill in.

Why this matters

If you let AI write the whole thing, it sounds like everyone else's. Your outline is the brief. It decides what the page is for, and your own words are what make it sound like you.

Step 2: Give Claude your raw material

I opened Claude Cowork, pasted in my outline and asked it to help me build a one-page website. I told it I wanted to iterate, so the first version didn't need to be perfect. Then I gave it what it needed:

- **My master CV.** Claude found it in my Google Drive and picked out achievements to fill my placeholders. Because my master CV flags which figures I've confirmed, Claude only used numbers I can stand behind.
- **My links.** My LinkedIn and my Google Calendar booking page.
- **Files.** A PDF guide I wanted to link to, and company logos from my Downloads folder. Claude can only look in folders you connect in the desktop app, and it asks first.

Step 3: Let Claude ask you questions first

I asked Claude to question me before building, and it was worth it. It asked what the one outcome of the page should be, who would read it first, where to find my achievements and whether I wanted a web page, a

PDF or both. My answers (warm intros from my network, one page that also prints to PDF) shaped everything that followed.

Step 4: Get a first version, fast

Within one reply, Claude built a first draft and published it as a private preview page that only I could see. It was simple and clean, it worked on my phone, and it had clear placeholders for anything missing, such as my booking link. It also gave me a short list of places where my CVs disagreed with each other on dates or figures, so I could decide which version to use.

Treat it as a draft

The first version is there to react to, not to publish. Seeing something real on screen makes it far easier to know what you want to change.

Step 5: Iterate by commenting on the page

This was the best part. Instead of describing changes in chat, I highlighted text on the preview page and left a comment, like you would in a Google Doc. Claude made the change, updated the page and replied in the comment thread to confirm what it did. Some of mine:

- "Swap these around so 75 NPS comes first"
- "It was 2019, so change this and the date below"
- "Add all the logos across my career and make it a slider that keeps ticking across"
- "Swap this for a marketing evidence point"

When I wanted a bigger change, I asked Claude to mock it up as a separate page first. That way I could compare the new idea with the live version before committing. I also asked other people what they thought. My wife's feedback that it was too wordy led to fold-away sections, and later to the biggest improvement of all.

The change that made the biggest difference

My first version read like a digital CV. It explained who I am, but it didn't tell people what I wanted them to do. So the page now opens with one question: *why are you here?* Each answer leads to its own path. Helpers get three simple steps, hiring managers get my experience and a booking link, and founders get my pro-bono offer. Work out what you want each visitor to do, then design the page around that.

Tips for iterating

- One change per comment. It keeps each edit clean and easy to check.
- Check it on your phone. Most people will open your link on a phone.
- Ask Claude for its opinion, not just changes. It pushed back on a few of my ideas and suggested better ones.

Step 6: Publish it on your own domain with Netlify

Once I was happy, I asked Claude to get it onto `simonhawtin.com`. It checked where my domain was managed, spotted that an old site was already running on it, and packaged my site into a folder ready to upload. Here's what I then did:

1. **Upload the site.** Go to `app.netlify.com/drop`, drag your site folder onto the page and sign up when asked. Netlify gives you a test address ending in `.netlify.app`.
2. **Add your domain in Netlify.** Open **Domain management**, choose **Add a domain** and enter it with `www`, for example `www.yourname.com`.
3. **Point your domain at Netlify.** Log in to wherever you bought your domain (mine is GoDaddy), open its DNS settings and change two records:
 - The **A** record for `@` becomes `75.2.60.5`
 - The **CNAME** record for `www` becomes your `.netlify.app` address
4. **Leave everything else alone.** In particular, don't touch the MX records. They run your email.
5. **Wait.** DNS changes can take up to 24 hours to spread, though mine worked within minutes. Netlify then sets up HTTPS on its own.

Don't panic at the warning

When I first visited my site, Chrome showed "Your connection is not private". That's normal. I'd simply loaded the page a minute before Netlify had finished setting up the secure certificate. Wait 10 to 15 minutes and try an Incognito window.

These are Netlify's own instructions for domains managed elsewhere, from their "Configure external DNS" guide. Check it for the latest values before you change anything.

Step 7: Connect Claude to Netlify

Dragging a new folder onto Netlify after every edit got old quickly. So I connected Netlify to Claude. Claude has a Netlify connector: you click connect, sign in to Netlify and switch it on for your chat.

Now my process is simple. I comment on the preview page, Claude makes the change, and when I say "publish this", Claude pushes it straight to `simonhawtin.com` and checks it loaded correctly.

You can always go back

Netlify keeps every version you publish. If you don't like an update, open your site's **Deploys** tab, click an earlier version and choose **Publish deploy**. Or just ask Claude to do it.

The finishing touches

- **A favicon.** The little icon in the browser tab. Claude designed four options in my site's colours and I picked the SH monogram.
- **Real logos.** Claude pulled what it could from my Downloads folder and from the companies' own websites. I supplied the ones it couldn't reach.

- **PDF-friendly.** Printing the page or saving it as a PDF gives a clean copy with the buttons hidden, so it doubles as a one-pager attachment.

The two principles behind all of it

1. **Your words, Claude's hands.** I decided what the page said and why. Claude did the building, the design and the technical work I'd never have done myself.
2. **Ship rough, then improve.** The first version went up in minutes. Everything good about the final site came from reacting to what was in front of me, one comment at a time.

Take a look at the finished site at simonhawtin.com. Happy to walk you through any of this live if it helps.