danimart.ai

FREE GUIDE · CLAUDE CODE

Websites that don't look AI-generated

The three-step setup from the video, plus which model to use, a prompt library and a checklist before going live. About 15 minutes to set up, no design background needed.

01 Install Claude Code

02 Add the design skill

03 Connect 21st.dev

Claude and Claude Code are trademarks of Anthropic. Not affiliated with Anthropic or 21st.dev.

STEP 01

Install Claude Code

Download the Claude app at claude.com/download, sign in and open the **Code** tab. Claude Code needs a paid Claude plan.

> Prefer the terminal? Claude Code also runs from the command line. Either works for this guide.

STEP 02

Add the design skill

The skill is Anthropic's **frontend-design**. It teaches Claude to choose a clear visual direction, real typography and layouts that don't look like a template.

Page: github.com/anthropics/skills · skills/frontend-design

Easiest way: paste this into Claude Code.

```
Install the frontend-design skill from
https://github.com/anthropics/skills/tree/main/skills/frontend-design
into ~/.claude/skills/frontend-design and confirm it loaded.
```

Or by hand: download the folder and copy it to `~/.claude/skills/frontend-design/`. Restart Claude Code.

STEP 03

Connect 21st.dev

21st.dev is a library of interactive UI components: heroes, pricing tables, buttons and more. Connecting it lets Claude use them in your site.

- Go to 21st.dev/mcp and create your API key.
- In your terminal, run:

```
npx @21st-dev/cli@latest init --client claude
```

Or ask Claude Code to add an HTTP MCP server named `21st` with the URL `https://21st.dev/api/mcp` and your key in the `x-api-key` header. Restart Claude Code and you're done.

YOUR FIRST SITE

The prompt to start with

Copy it and fill in the brackets.

```
Use the frontend-design skill and 21st.dev components.
Build a one-page website for [your business: what you do and for whom].
Style: [2 or 3 words, e.g. calm and premium / bold and technical].
Sections: hero, what I offer, how it works, proof, contact.
Start with the hero only and show it to me before building the rest.
```

Tips that make the difference

- **One section at a time.** Approve the hero before asking for the rest.
- **Give a reference.** Paste the link of a site you like and say what you like about it.
- **Use your real content.** Placeholder text makes any design look generic.
- **Check it on your phone.** Ask Claude to review the mobile version and fix anything that overflows.
- **Be specific when correcting.** “Bigger headline, less space under the logo” works better than “make it nicer”.

WHICH MODEL

Use the right model for each job

Claude Code lets you switch models at any time with the `/model` command. Picking well saves usage and gets better results.

Model	Use it for
Claude Opus	The first visual direction, full pages from scratch, redesigns and anything where taste matters. Slower and uses more of your plan.
Claude Sonnet	Day-to-day building: new sections, changes, fixing layout on mobile. The best balance for most of the work.
Claude Haiku	Small, fast edits: changing a text, a colour or a link.

A simple routine: Opus to decide the look and build the hero, Sonnet for the rest of the page, Haiku for final text tweaks. If you hit your usage limit, switch down a model instead of stopping.

Recommendations

- **Work in a project folder.** Open Claude Code inside a new folder for each site so files stay together.
- **Write a short brief first.** Who the site is for, what you sell, one action you want visitors to take.
- **Ask for options before building.** Three visual directions described in words are cheaper than three full pages.
- **Keep one style.** Once you like a direction, ask Claude to save colours, fonts and spacing in a style file and reuse it.
- **Review in the browser.** Ask Claude to start a local preview and check desktop and phone sizes.
- **Don't paste secrets.** API keys go in environment variables or the MCP setup, never in the page code.

PROMPT LIBRARY

Prompts to get the most out of it

Copy, fill in the brackets and paste into Claude Code.

1. Choose a visual direction

```
Using the frontend-design skill, propose 3 visual directions for a website for [business]. For each: mood in 3 words, colour palette, fonts and one reference. Don't write code yet.
```

2. Redesign an existing site

```
Here is my current site: [URL]. Keep the content, redesign it with the frontend-design skill so it looks custom-made, not like a template. Start with the hero and show me.
```

3. A landing page for one offer

```
Build a landing page for [offer] aimed at [who]. One goal: [book a call / download / buy]. Sections: hero, problem, solution, how it works, proof, FAQ, final call to action.
```

4. Use a 21st.dev component

```
Search 21st.dev for a [pricing table / testimonial section / animated hero] that fits our style and adapt it to my content.
```

5. Mobile pass

```
Review every section at 390px wide. Fix anything that overflows, text that is too small and buttons that are hard to tap. List what you changed.
```

6. Final quality check

```
Check the page before publishing: broken links, image sizes, loading speed, colour contrast and missing alt text. Fix what you can and tell me the rest.
```

7. Put it online

```
Explain the simplest way to publish this site for free with my own domain, step by step, and do the parts you can do from here.
```

BEFORE YOU GO LIVE

Checklist

- One clear action on the page, and the button says exactly what happens next.
- Your real texts, photos and logo. No placeholder text left.
- Looks right on a phone, a laptop and a large screen.
- Every link and form tested by you.
- Fast to load: compressed images, no huge videos in the first screen.
- Readable: enough contrast and text size.
- Page title and description set for Google and for link previews.

Common mistakes

- **Asking for “a nice website”.** Vague prompts give generic results. Say who it’s for and how it should feel.
- **Building everything at once.** Long single prompts are harder to correct. Go section by section.
- **Mixing too many components.** Pick a few 21st.dev pieces that match, not one of each.
- **Skipping mobile.** Most visitors arrive from a phone.

Want more setups like this? Join the danimart.ai Telegram group for daily tips, new skills and every new post: t.me/+V8Z1lbLHZgVjYzdk