

Claude Design = Instant \$10k Websites

This guide summarizes the key concepts, workflows, and tips for using Claude Design to build, iterate on, and deploy polished, interactive websites, from brand brief to live URL, without needing to be a designer or developer.

By: [Nate Herk](#)

I. What is Claude Design?

Claude Design is Anthropic's new AI-powered design product (launched April 17, 2026 by Anthropic Labs at claude.ai/design) that lets you generate polished visual work like websites, prototypes, slide decks, landing pages, one-pagers, and marketing collateral through conversational prompts and fine-grained editing controls.

Key facts to know:

- **Built on Claude Opus 4.7**, Anthropic's most capable vision model, with roughly 3x the image resolution of prior versions and stronger taste/creative judgment.
 - **Research preview**, so expect bugs, evolving UI, and occasional quirks.
 - **Paid plans only**. Available on Pro, Max, Team, and Enterprise. Not available on free plans. Enterprise/Team admins must manually enable it (off by default).
 - **Two ways to start**: transform an existing codebase (drop in a GitHub repo or folder) OR build from scratch with a prompt + sketch.
 - **Positioned against Figma and Canva**, but designed to complement Canva specifically (direct export integration available).
-

II. The Full Website-Build Workflow (Step by Step)

Step 1: Brand & Idea Generation in Claude Chat

Before touching Claude Design, use regular Claude chat to develop the concept:

1. Ask Claude to invent the product/service, positioning, voice, visual identity, and site spec.
2. Example from the video: a wind-down drink brand called "**LOL**" (magnesium glycinate / sleepy-time vibe).
3. Follow up with a prompt asking for:
 - An **image prompt** for the hero background
 - A matching **video prompt** to animate that image

Step 2: Generate the Hero Background Video

- Use **Kie.ai** (or your preferred image/video generator).
- **Nano Banana 2** generates the hero image at 16:9 ratio (dark ceramic mug with steaming liquid, left side empty for text).
- **Kling 2.0** (Cance 2.0) takes the image as **both first and last frame**, then you paste Claude's video prompt.
- **Key tip:** keep the camera stationary. You want a looping, subtle animation (steam rising), not a panning shot.

Step 3: Open Claude Design

- Click **New Prototype** and pick **High-Fidelity** (or wireframe first to save tokens, see Section VI).
- Skip the design system for now if you're creating a brand-new brand.
- Name the project (e.g., "LOL Website") and hit **Create**.

Step 4: Sketch + Upload Assets

Claude Design spins up a Lovable-style canvas. On the left, you can:

- **Draw a rough sketch** of the hero section with boxes labeled "hero background video," "hero text," "hero subtext," "navbar," "logo," and "shop."
- **Drag in the MP4** of your background video (≤ 30 MB cap, roughly ≤ 20 seconds).
- **Paste in the brand spec** from Claude Chat (it auto-converts to an artifact).

Step 5: Write the Kickoff Prompt

Tell Claude Design:

- What the sketch represents (hero section only).
- That the MP4 should loop as the hero background.
- Where the hero text lives (left side).
- The 20+ lines of brand identity you just pasted.
- To write all the copy for every section now ("we can tweak later").

Claude Design (using Opus 4.7) then builds a to-do list, creates a design system, picks a color palette, and starts generating. You can **watch its thinking live**.

Pro tip: If it goes down the wrong path, stop it immediately. Letting it finish wastes session limit.

III. Iterating on the Canvas (The Token-Saving Gold)

Once the first version renders, Claude Design offers four in-canvas editing tools that burn **far fewer tokens** than re-prompting. This is where Claude Design beats Cloud Code and Lovable for visual work.

Tool	What It Does	When to Use
Comment	Click any element, type a note, send to Claude. Claude knows exactly which element you meant.	Targeted changes ("make this button gold")
Inline Edit	Double-click text to edit directly in the canvas, including resizing on the fly.	Quick copy fixes, removing AI-ish punctuation, font-size tweaks
Draw	Circle or scribble directly over an area, then leave a comment ("don't like how this ends abruptly").	Describing issues in a specific region
Tweaks Panel (Sliders)	Claude generates custom sliders and toggles so you can change palette, accent hue, font, headline size, hero layout, video dim, section rhythm, and card style, all without sending a prompt.	Exploration + creative discovery

Magic unlock prompt: "This is a really great iteration. I want to see what else is possible. Add a bunch of things that I can tweak, different colors, sliders, panels, and give me a bunch of things to play with."

Then Claude Design adds a tweaks panel filled with controls. Experimenting there is essentially free; actual prompts are what burn your limit.

IV. Getting Design Inspiration: motions.ai

- URL: **motions.ai** (one-time \$99 paid plan for lifetime access, most content is free).
- Browse scroll journeys, 3D card sites, and animated backgrounds.
- Hit **Copy** on any concept, paste it into Claude Design, and tell Claude to swap in your own brand video, copy, and colors.
- This is exactly how Nate transformed his AI Automation Society site into its current scroll-journey layout in about 20 minutes.

You can also download standalone animated backgrounds from motions.ai and drop them into any section of your site.

V. Thinking About Your Site in Sections

Break the build into sections instead of trying to design the whole thing at once:

1. **Hero section** (built first)
2. **Section 1**: dynamic background (e.g., shooting stars)
3. **Section 2**: feature or problem/solution
4. **Section 3**: product video (e.g., someone stirring coffee)
5. ... and so on

For each new section:

- Ask Claude Chat what type of element, video, or image would fit.
 - Have it generate the image and video prompts.
 - Build the assets in Kie, then bring them back into Claude Design.
 - Iterate on that section only, then move to the next.
-

VI. Deployment Pipeline: Claude Design, GitHub, Vercel

Once the site is ready inside Claude Design, ship it live:

1. Export from Claude Design

- Click **Share, Download as ZIP** (most reliable).
- Alternative: "**Hand off to Claude Code**" gives you a paste-in command, but was returning 404s at the time of filming.

2. Move into Claude Code

- Extract the ZIP into a named desktop folder (e.g., `LOL-website`).
- Open the folder in **VS Code** with Claude Code.
- Project will contain: background video, sketch (scraps folder), uploads, and the HTML file.

3. Push to GitHub

- Tell Claude Code: *"Push this to a private GitHub repository and call it whatever you want."*
- First time? Create a free github.com account and authorize Claude Code when it prompts.

4. Test on Localhost

- **Always** test before deploying: *"Make sure the website works as expected and open it up in a local host for me to view."*
- Remember: a `localhost:3000` URL is **only viewable on your machine**, so don't share it.

5. Deploy to Vercel

- Sign up for Vercel with the same GitHub account.
- Click **Add New, Project, Import** your repo, then **Deploy**.
- About 60 seconds later you get a live `yoursite.vercel.app` URL.

6. Fix 404 Errors

If Vercel shows 404 Not Found, the repo likely doesn't have an `index.html` at the root. Tell Claude Code:

"I just pushed to GitHub, synced to Vercel, and got a 404. What does this mean? How do I fix it?"

Claude Code will typically rename your main HTML file to `index.html` and push the fix. Vercel auto-redeploys in about 30 seconds.

7. The "Practice vs. Real" Environment

- **Local (Claude Code)** is your practice environment. Only you see changes.
- **GitHub push** promotes changes to production.
- **Vercel** auto-deploys every time GitHub updates.

Clean separation between drafts and the live site.

VII. Don't Forget Mobile Optimization

Claude Design and Claude Code **do not auto-optimize for mobile**. Before going live:

1. Open the site in your local host.
2. Press **F12** and click the mobile view toggle in DevTools.
3. If it looks broken (hero text too big, video misplaced, content squished), tell Claude Code:

"Optimize this for mobile without affecting the desktop view. Move the video below the text, reduce font size, etc."

VIII. The Two-Meter Usage System

Claude Design has its **own weekly quota**, separate from regular Claude chat and Claude Code. Regular Claude keeps working after Design maxes out. Track both in **Settings, then Usage**.

Plan Burn Rates (Approximate)

Plan	Rough Capacity
Pro (\$20/mo)	~1 big website + design system and you're basically done for the week
Max 5x (\$100/mo)	~4 to 5 serious prompt sessions before hitting the limit
Max 20x (\$200/mo)	A few video projects plus a few websites before getting close to the limit
Team / Enterprise	Admin-controlled, custom pricing

Nate burned through his own 20x quota and spent **\$200+ in extra usage** just exploring.

IX. What Drains Your Limit Fastest (and How to Fix It)

- **Defaulting to Opus 4.7 on every prompt.** Switch to **Sonnet 4.6** for drafts, exploration, and small changes. Use Opus for planning/design systems and major structural work only.
- **Skipping wireframes.** Do a wireframe first, confirm direction, then convert to high-fidelity (less iteration equals fewer tokens).
- **Design systems without a URL.** Paste your live site URL so Claude scrapes brand automatically instead of ingesting 10 to 15 minutes of setup prompts.
- **Rewriting whole prompts for small edits.** Use the in-canvas Comment, Draw, Inline Edit, and Tweaks tools instead.
- **Super long threads.** Context gets polluted, tokens get un-cached and reprocessed. Export the project, start a fresh session, and keep iterating on the already-built foundation. (`/clear` is also available but unconfirmed whether it fully clears context.)

X. Best Practices to Stretch Limits

The ZIP Export Hack

- Export your **design system** as a ZIP (contains a **SKILL .md** and **README .md**).
- Drop it into **Claude Code** and tell Claude Code to use **SKILL .md** as its skill.
- Keep generating on-brand designs using your **Claude Code** allowance instead of Design allowance.

Model by Stage

- **Opus 4.7** for design system generation and final high-fidelity polish.
- **Sonnet 4.6** for wireframes, exploration, slide iteration, and copy edits (still excellent if prompts are specific).

Prompting Rules That Save Tokens

- **One visual dimension per prompt.** Don't combine typography, color, and layout in a single mega-prompt. Claude will only execute 1 or 2 of them well and ignore the rest.
 - **Use negatives.** Tell it what you *don't* want ("no Inter, no Roboto, no YC gradients, no weird punctuation") so you don't have to correct later.
 - **Reference real products.** "*Linear 2023 with higher density*" beats "*make it clean.*"
 - **Point-and-comment** instead of full prompt rewrites.
 - **Ask for prototypes, flows, and 10-second demos** for better value per token than static screens.
 - **Give maximum context upfront:** sketches, brand specs, URLs, brand assets. Context on the first try equals fewer iterations.
-

XI. File Uploads & Integrations

What Claude Design Accepts

- **Codebase:** website URL, public GitHub repo, or local folder
- **Design files:** Figma **.fig** exports, custom fonts (**.woff**, **.ttf**), logos (PNG, SVG)
- **Documents:** DOCX, PPTX, XLSX
- **Media:** Images and videos up to ~30 MB (~20 seconds tested as working)
- **Web capture tool:** grab elements directly from live websites

Export Options

- Internal organization URL (shareable link)
- Canva (direct integration, designs become fully editable)
- PDF
- PPTX (PowerPoint)
- Standalone HTML
- Handoff bundle for Claude Code (packages design and intent together)

XII. Known Limitations (Research Preview)

- **No drag-and-drop repositioning.** Edit through the inspector and comments only.
 - **Browser only.** No desktop app or terminal integration at launch.
 - **Comment persistence is buggy.** Inline comments sometimes disappear before Claude reads them. Workaround: paste the comment text into chat.
 - **Compact view save errors.** Switch to full view and retry if saves fail.
 - **Large repos cause lag.** Link specific subdirectories, not entire monorepos.
 - **Multi-screen consistency drops after about 4 screens** unless you re-anchor the design system.
 - **Collaboration is basic.** Not yet fully multiplayer like Figma.
 - **"Container soup" default look.** Pills, cards, serif font, and a blinking status dot appear repeatedly unless you aggressively override with your own references and tokens.
 - **Best with clean codebases.** Messy source code produces messy output.
-

XIII. When Claude Design Runs Out, Keep Building in Claude Code

If you burn through your Claude Design weekly quota mid-project:

1. **Export** your Claude Design project as a ZIP.
 2. **Open it in Claude Code** (which has its own separate usage pool).
 3. **Keep iterating** on the same project with new sections, mobile optimization, and bug fixes.
 4. **Wait for the Design quota to reset**, then bring the updated project back into Claude Design for more visual and canvas work.
-

XIV. Recap: The Minimum Viable Workflow

1. **Claude Chat** to invent brand, positioning, copy spec, and image and video prompts.
2. **Kie.ai** (Nano Banana 2 + Kling) to generate the hero image and looping background video.
3. **Claude Design** to sketch the hero, upload assets, paste the brand spec, and let Claude build.
4. **Iterate with Comments, Inline Edits, Draw, and the Tweaks Panel** instead of fresh prompts.
5. **Motions.ai** to grab scroll-journey and animated-background inspiration when stuck.
6. **Build section by section:** hero, then section 1, then section 2, and so on.
7. **Export as ZIP, open in Claude Code, push to GitHub, deploy to Vercel.**
8. **Test on localhost and mobile view** before pointing anyone at the live URL.

9. **Preserve your quota:** model by stage, one dimension per prompt, negatives, and references to real products.
-

Want to connect with others building and monetizing AI automation?

[Become an AIS Plus Member](#)