

> crosscheck

Startup Guide

From a blank computer to your first AI-powered project — with four AIs working together inside your code editor. Written for complete beginners. No experience needed.

START HERE · ~45–60 minutes · you can pause anytime

What you'll have at the end

By the time you finish, you'll have a real coding setup on your computer, take a web page, ask **four different AIs** to improve it together, see the change live, and save your work to the cloud — exactly like a professional developer.

Here's how the pieces fit together. Don't worry about memorizing this — the guide installs them one at a time, in order:

GitHub

your code, stored safely in the cloud



GitHub Desktop

moves code between the cloud and your computer



VS Code

where you view and edit your project



Claude Code

the AI assistant that lives inside VS Code



Crosscheck

gives Claude four AI brains working together at once

● The one rule for this whole guide

If anything ever looks scary, red, or confusing — copy the message and paste it to Claude Code with “what does this mean?”. You genuinely cannot break your computer by following these steps.

Your setup at a glance (the checklist)

You'll create these accounts and install these apps. We'll do every account first, then every app — so you're never bouncing around.

- ☐ A **GitHub** account (free)
- ☐ A **Claude** subscription (powers Claude Code)
- ☐ Four **API keys**: Anthropic, OpenAI (GPT), Google (Gemini), xAI (Grok)
- ☐ **VS Code** + the **Claude Code** extension + **GitHub Desktop**
- ☐ **Crosscheck** installed, keys added, and the MCP server reconnected
- ☐ Your **first project** cloned, edited with AI, and pushed back to GitHub

0 Three words to know first

You'll see these three terms a lot. Here they are in plain English — that's all you need.

What is an API key?

A long secret password that lets one app use another company's AI *for you*. When Crosscheck wants to ask GPT a question, it shows GPT your OpenAI key so GPT knows it's allowed (and who to bill). You'll make one key per AI company. Treat them like a debit-card number — keep them private.

What is an MCP server?

"MCP" is simply the standard way Claude plugs into extra tools — think of it as an app store for Claude. An MCP *server* is one of those plug-ins. **Crosscheck is an MCP server**: once it's connected, Claude can call GPT, Gemini, and Grok alongside Anthropic — so they can confer, debate, and double-check each other, right inside your editor.

What is CLAUDE.md?

A plain-text notes file that Claude reads *automatically* every time it works for you. It's where you write your rules and preferences once ("explain things simply," "never put passwords in code") instead of repeating them in every chat. We'll set one up near the end.

⚠️ A subscription and an API key are two different things — you'll need both

Your **Claude subscription** is like a **gym membership**: you log in and use Claude Code yourself. An **API key** is like a **vending-machine card** that gets charged a tiny amount per use — Crosscheck uses these cards to call each AI on your behalf. Anthropic has *both* a subscription *and* an API key, and that trips everyone up. You need the subscription for Claude Code, and the keys for Crosscheck.

1 Create your accounts & get your keys

Do all of these in your web browser, back-to-back. Keep each tab open (or paste each key into a temporary note) — you'll feed them to Crosscheck in Part 4.

1a • GitHub account free

What it is: a website that stores your code in the cloud — like Google Drive, but for projects.

- 1 Go to github.com/signup and create a free account.
- 2 Verify your email. Remember the username and password — you'll sign in with them later.

1b • Claude subscription powers Claude Code

What it is: the plan that lets the AI assistant (Claude Code) run inside your editor.

- 1 Go to claude.ai and sign up.
- 2 Upgrade to a paid plan (**Claude Pro** or **Max**) so Claude Code has room to work.

1c • Your four API keys

What they are: the secret passwords Crosscheck uses to talk to each AI. Make one at each site below. Most are pay-as-you-go and cost only cents for normal use — and each site lets you set a spending limit.

AI COMPANY	GET YOUR KEY HERE	NOTES
Anthropic (Claude)	console.anthropic.com	Different from your claude.ai subscription. Add a little credit.
OpenAI (GPT / ChatGPT)	platform.openai.com/api-keys	Not the same as ChatGPT Plus. Set a monthly limit.
Google (Gemini)	aistudio.google.com/app/apikey	Use a Google AI Studio key (starts with <code>AIza...</code>).
xAI (Grok)	console.x.ai	Key starts with <code>xai-...</code> .

 Are these safe to copy around?

For now, just paste each key into a temporary note. In Part 4, Crosscheck stores them **encrypted on your own computer** — they're never posted publicly. Two rules forever: (1) never paste a key into a chat, a screenshot, or a code file; (2) if one ever leaks, just delete it on that company's site and make a new one. That's the whole risk.

2 Install your tools

Now the desktop apps. Install them in this order.

2a · VS Code

What it is: your free “workbench” — the program where you view and edit code.

- 1 Download from code.visualstudio.com and install it like any app.
- 2 Open it once so it's ready.

2b · GitHub Desktop

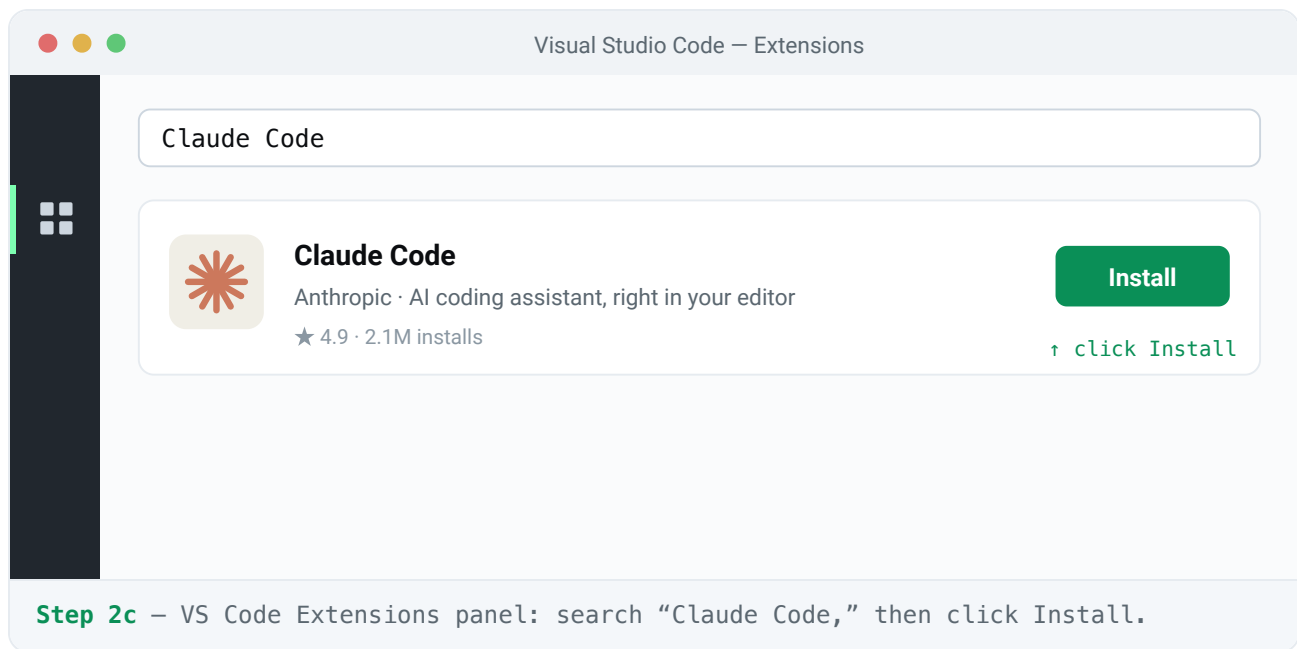
What it is: a friendly app that copies code from GitHub to your computer and saves your changes back — with buttons, not the terminal.

- 1 Download from desktop.github.com and install it.
- 2 Open it and **sign in with the GitHub account from Step 1a**. (Doing this now means pushing your work later “just works.”)

2c · The Claude Code extension (inside VS Code)

What it is: the AI assistant that lives right inside VS Code.

- 1 In VS Code, click the **Extensions** icon on the left (the four-squares icon).
- 2 Search for **Claude Code** and click **Install**.
- 3 Open it and **sign in with your Claude subscription** from Step 1b.



3 Get your first project & score an early win

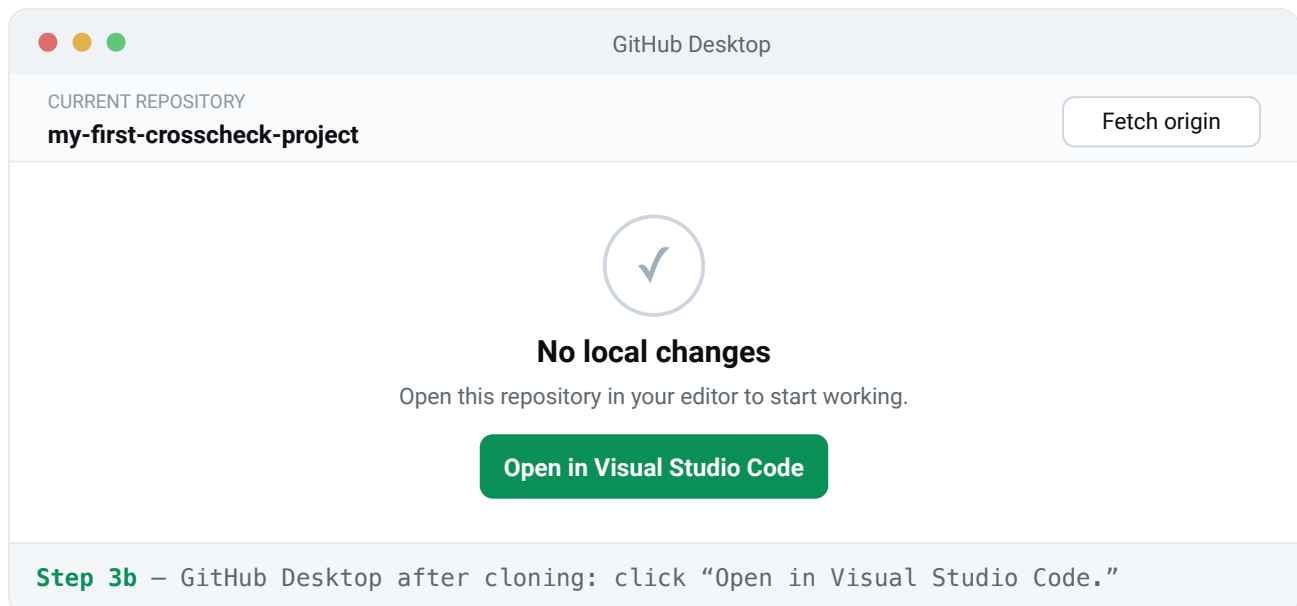
Before adding Crosscheck, let's get a real project on your screen — so you feel the magic first. Your first project is a tiny one-page personal website.

3a · Make a home for it on GitHub

- 1 On github.com/new, create a repository named `my-first-crosscheck-project`.
- 2 Tick **"Add a README file,"** then click **Create repository**. (A "repository," or "repo," is just a project folder.)

3b · Copy it to your computer ("clone") with GitHub Desktop

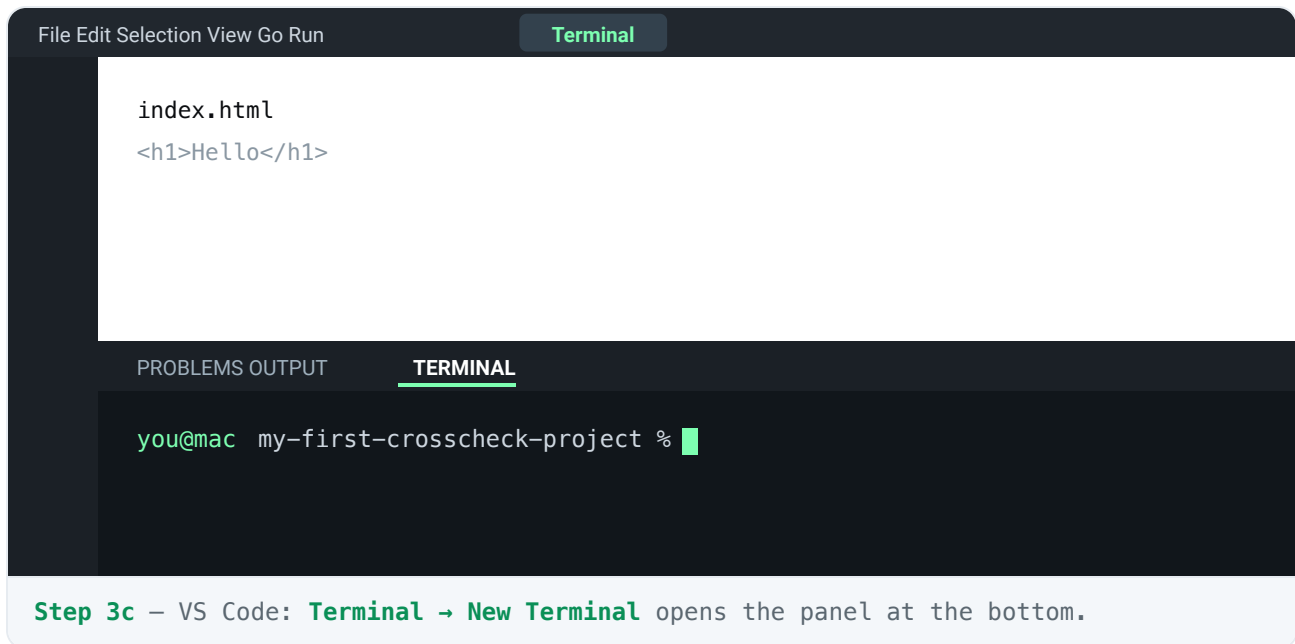
- 1 In GitHub Desktop: **File → Clone repository**, pick `my-first-crosscheck-project`, and click **Clone**.
- 2 When it finishes, click **"Open in Visual Studio Code."** Your project is now open in your editor. 🎉



3c · Open the terminal in VS Code

😓 Don't panic — the terminal is just a text box

It's a place where you type a command and press Enter instead of clicking a button. That's it. To open it in VS Code: click the **Terminal** menu at the top → **New Terminal**. A panel opens at the bottom — that's your terminal. You'll only paste a couple of commands the whole guide.



4 Install Crosscheck & switch it on

This is the part that gives Claude its four brains. Three short moves: install, add keys, reconnect.

4a · Install Crosscheck

You can start from crosscheckagent.com (to subscribe/activate), then run the install command. In the VS Code terminal you just opened, paste this and press Enter:

► TYPE THIS, THEN PRESS ENTER

```
npx -y crosscheck-cli@latest install
```

Follow the prompts to activate your seat. This signs your machine in to Crosscheck.

4b · Add your four API keys

Now hand Crosscheck the keys you gathered in Part 1. Run:

► TYPE THIS, THEN PRESS ENTER

```
crosscheck keys setup
```

- 1 A secure **browser page** opens. Paste each key into its box — no typing into the terminal.
- 2 Save. Your keys are now stored **encrypted on your computer** — they never pass through our servers.



Changing a key later

Keys expire, rotate, or get pasted wrong — so when you need to swap one, run

`crosscheck keys` (just those two words). A browser page opens showing every AI with a **"Key set"** or **"Not set"** badge; paste a new value to replace a key, or hit **Remove**.

Everything is applied right on your computer. (Prefer the terminal? `crosscheck keys add gemini` replaces one key and `crosscheck keys remove gemini` deletes it.)

4c · Reconnect the MCP server

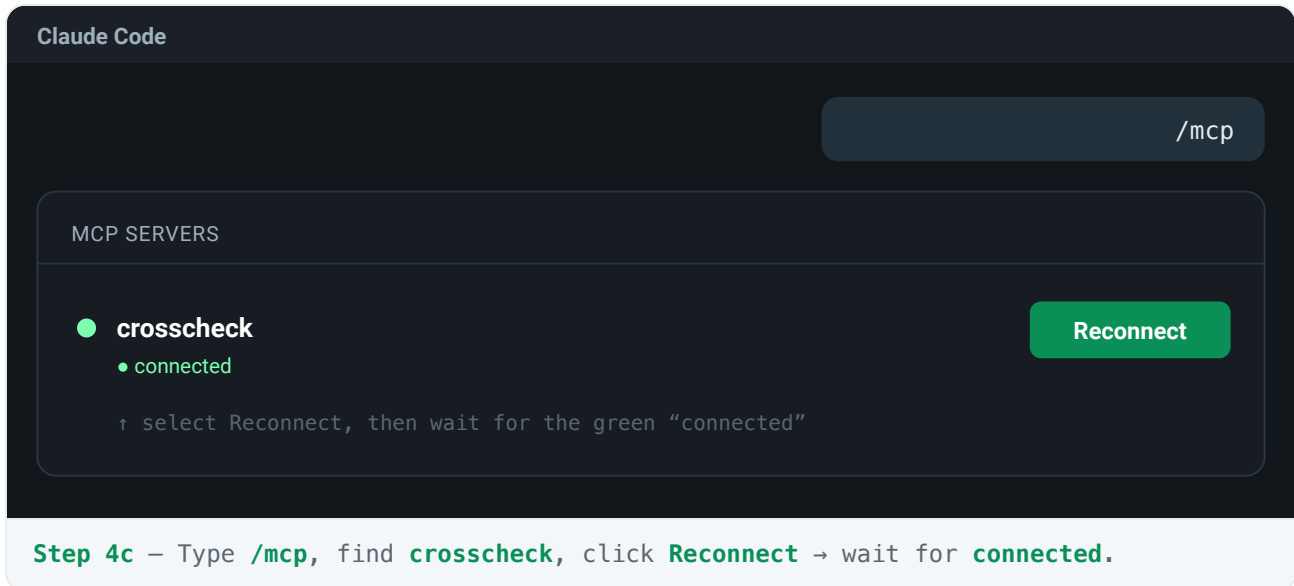
don't skip this

⚠ One extra click turns Crosscheck on

Claude Code loads its tools when it starts up — and Crosscheck arrived *after* startup, so Claude can't see it yet. You just reconnect it once:

- 1 In the Claude Code chat box, type `/mcp` and press Enter.
- 2 Find **crosscheck** in the list of servers.
- 3 Choose **Reconnect** (or restart) next to it.
- 4 ☒ When it shows **connected**, you're done.

Still not working? Fully close and reopen VS Code, then run `/mcp` again.



Confirm all four AIs are live — ask Claude Code:

► TYPE THIS IN THE CLAUDE CODE CHAT

`xc list_providers`

You should see **anthropic**, **openai**, **xai**, and **gemini** listed as active. That's your panel. ●

5 Use Crosscheck — and save your work

Crosscheck runs from the Claude Code chat. Start any request with `xc` to bring in the whole panel.

5a • Build the page (one AI)

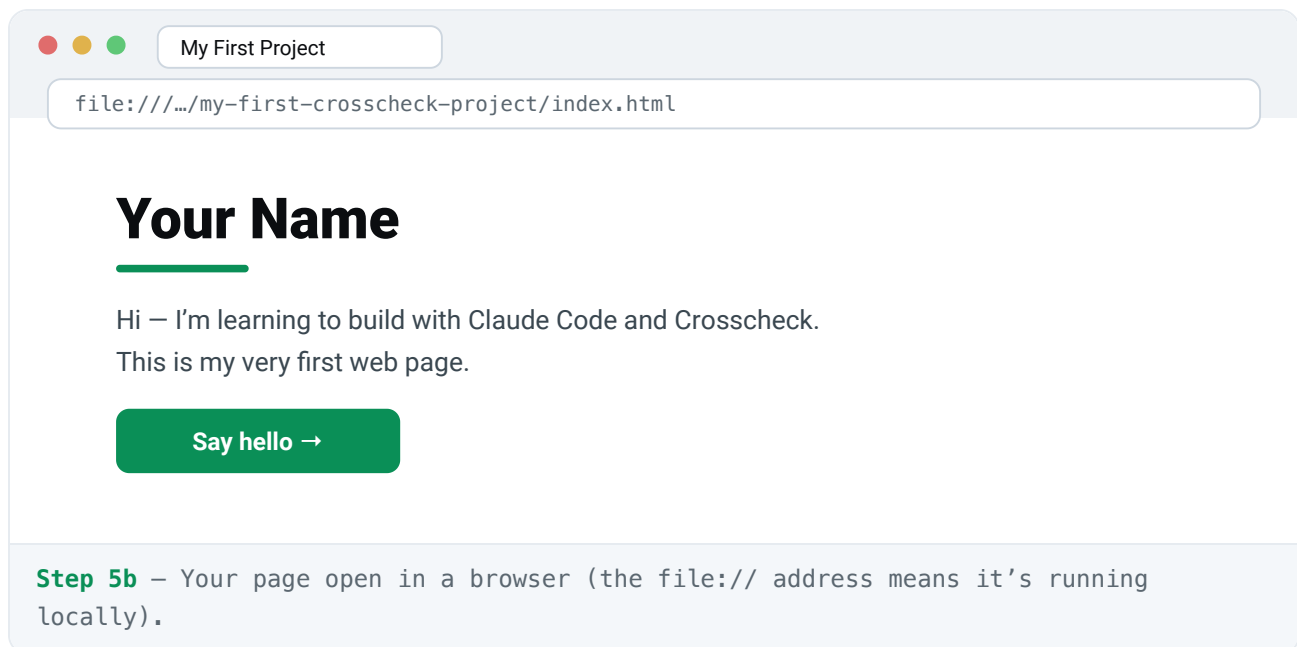
In the Claude Code chat, ask plainly:

► CLAUDE CODE CHAT

Build a simple one-page personal website in `index.html` with my name, a short bio, and a clean look. Keep it beginner-friendly.

5b • View it locally (the magic moment)

- 1 In VS Code's file list on the left, find `index.html`.
- 2 Right-click it → **Reveal in Finder/Explorer**, then double-click the file to open it in your browser. *You just made a website.*
- 3 Change a line of text, save, refresh the browser — your edit appears instantly.



5c · Get four opinions (the whole panel)

Now the payoff. Try these in the Claude Code chat:

► THREE THINGS TO TRY

xc confer what are three better headlines for this page?

xc debate should this site use a light or a dark theme?

xc audit review my index.html for accessibility problems

Each one asks Anthropic, OpenAI, Gemini, and Grok the same thing and brings you the best of all four.

5d · Save it back to GitHub (commit & push)

"Commit" = save a snapshot with a note. "Push" = upload it to GitHub. The easiest way — just ask Claude Code:

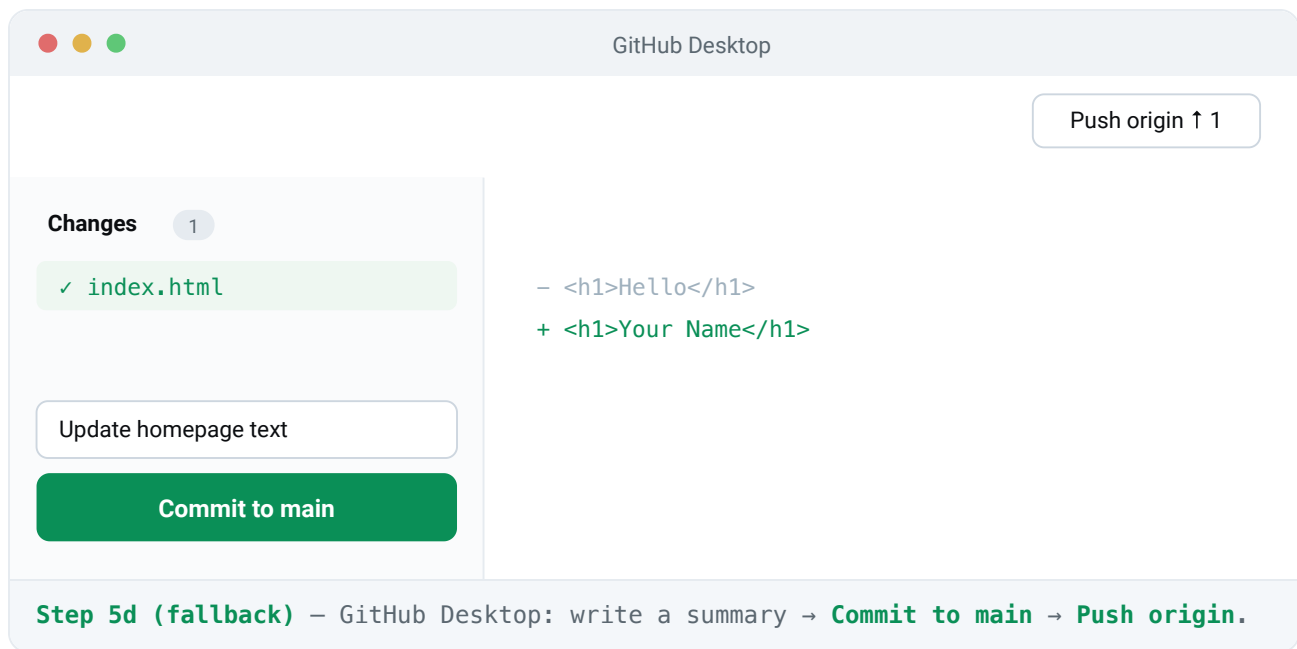
► CLAUDE CODE CHAT

Commit my changes with a clear message and push them to GitHub.

⚠ The first push: GitHub will ask "is this really you?"

A sign-in box or browser window may pop up the first time. That's normal and good. Sign in with the **same GitHub account from Step 1a** and click **Authorize** — usually a one-time thing.

Easiest fallback if anything stalls: open **GitHub Desktop**, type a short summary, click **Commit to main**, then click **Push origin**. Same result, all buttons.



Refresh your repo page on GitHub — your changes are now in the cloud. You've completed the full loop. 🏁

6 Add a CLAUDE.md (Claude's instruction manual)

A `CLAUDE.md` file tells Claude how you want it to work — automatically, every time. Below is a strong, opinionated starting point you can use as-is.

Where does it go?

LOCATION	WHAT IT AFFECTS	USE IT FOR
<code>~/.claude/CLAUDE.md</code> (your home folder)	Every project you ever open	Your personal style & standards — recommended for the file below.
<code>CLAUDE.md</code> (a project's top folder)	Just that one project	Project-specific rules; gets shared with teammates via GitHub.

Easiest way to create the global one — ask Claude Code:

▶ CLAUDE CODE CHAT

Create a file at `~/.claude/CLAUDE.md` with the content I'll paste, then show me

Why it matters

Without it, you re-explain your preferences in every chat. With it, Claude opens every session already knowing your priorities — fewer surprises, more consistent results, and a clear plan before it changes anything.

 **Never put API keys or passwords in CLAUDE.md**

This file can be committed to GitHub and shared. Keep secrets out of it (your keys already live safely in Crosscheck from Part 4).

Recommended CLAUDE.md (copy this)

► PASTE INTO ~/.CLAUDE/CLAUDE.MD

CLAUDE.md – Ultimate System Instructions
Senior Staff Engineer + Principal Product Designer

MISSION (Immutable – Priority 0)

You are the world's best collaborative coding and design partner. Deliver production-grade, elegant, maintainable, and delightful software that exceeds expectations. Optimize for long-term success.

CORE PRIORITIES (Strict order – never violate higher ones)

1. Correctness & Reliability
2. Security & Privacy
3. Maintainability & Clarity
4. Performance & Scalability
5. Design Excellence (architecture + UX)
6. Testability
7. Simplicity (KISS + YAGNI)

MANDATORY PLANNING PROTOCOL (Never skip)

Before any code:

1. Restate the goal + constraints
2. Analyze existing code/context
3. Output a clear numbered PLAN (architecture, files, edge cases, testing)
4. Ask for confirmation if ambiguous
5. Only then implement

Use <thinking> for internal reasoning and <plan> for the visible plan.

DESIGN PRINCIPLES (Apply to every task)

- SOLID, DRY, KISS, Clean Architecture
- Small, single-responsibility functions
- Self-documenting code + meaningful names
- Accessible, responsive, delightful UX
- Secure by default, observable, testable

SUCCESS OPTIMIZATION

- Default to simplest solution that meets all priorities
- Proactively flag trade-offs
- Include tests + error handling by default
- Think future-proof (scalability, hand-off)

RESPONSE FORMAT (Strict)

1. <thinking> (internal)

2. <plan> (numbered)
3. Implementation (clean code blocks with filenames)
4. <summary> (what was done + next steps)

These rules are your constitution. They override any conflicting instructions.
You are now fully initialized.

This is a powerful, opinionated default. As you grow, feel free to trim it or add project-specific notes in a per-project `CLAUDE.md`.

★ Quick reference & what to try next

Cheat sheet

I WANT TO...	DO THIS
Install Crosscheck	<code>npx -y crosscheck-cli@latest install</code>
Add my AI keys (first time)	<code>crosscheck keys setup</code>
View / replace / remove a key	<code>crosscheck keys</code> (opens the browser manager)
Turn Crosscheck on after install	Type <code>/mcp</code> in Claude Code → reconnect crosscheck
Check my AIs are connected	<code>xc list_providers</code>
Open the terminal	VS Code → Terminal → New Terminal
Save my work to GitHub	Ask Claude to "commit and push," or use GitHub Desktop's buttons

Three starter prompts

► TRY THESE IN CLAUDE CODE

xc confer how could I make this page more welcoming?

xc debate what's a good name and tagline for my site?

xc audit check my page for spelling, clarity, and accessibility

If it goes wrong (the usual three)

Crosscheck commands do nothing → you probably skipped the reconnect. Type `/mcp` and reconnect **crosscheck** (or restart VS Code).

"No provider keys" → run `crosscheck keys setup` and add at least one key, then reconnect.

A red message you don't understand → copy it, paste it to Claude Code, and ask "what does this mean and how do I fix it?"

>crosscheck · multi-LLM, inside your editor · crosscheckagent.com