4 Free Claude Plugins Setup Guide

Install Impeccable, 21st MCP, Humanizer, and Claude Code Setup in Claude Code or Codex, with the commands I run and a fix table for the first-run errors.

Why I stopped blaming the model

For months I thought my AI output looked generic because the model was generic. Same fonts, same purple gradient, same card-inside-a-card layout, same captions that read like a press release. Then I noticed the people whose output did not look like that were running the same model as me. The difference was what Claude Code knew before it started working.

Every model trained on the same SaaS templates and the same blog posts. Left alone, it picks the answer that fits the widest range of people, which is another way of saying the most forgettable one. Four free plugins change what it knows going in. Impeccable gives it design rules. 21st MCP gives it more than 10,000 finished UI components to search. Humanizer gives it a list of the writing tells to remove, and learns your voice from a sample. Claude Code Setup scans your project and tells you which skills, hooks, MCP servers, and subagents actually fit, so you start clean.

This guide has every install command for Claude Code and Codex, the commands I run day to day, and the errors I hit setting them up so you do not have to. All four are free and open source, and only one of them (21st MCP) needs an API key.

01

How the four plugins actually work

1. Impeccable is one skill with 23 commands and 61 deterministic detector rules for frontend design. `/impeccable init` writes your product context to `PRODUCT.md`, later commands write the visual system to `DESIGN.md`, and a hook runs the detector every time Claude Code edits a UI file. It started from Anthropic's frontend-design skill and is Apache 2.0 licensed.
2. 21st MCP is a plain HTTP MCP server at https://21st.dev/api/mcp. It replaced the old Magic MCP. From inside your editor you search a catalog of over 10,000 React and Tailwind components, pull the code for one, and adapt it. Hosted AI generation (the `generate` tool) only appears when AI access is enabled on your account.
3. Humanizer is a skill that marks every AI writing tell it finds, strongest first, drafts a rewrite, checks it against the patterns and the original claims, then writes the final version. It never adds facts. Claude Code Setup, by Anthropic, is read-only: it scans the codebase and recommends the top one or two automations in each category (MCP servers, skills, hooks, subagents, slash commands) without touching a file.

---

02 The one thing almost everyone gets wrong

People install Impeccable and Humanizer and expect the output to change on its own. It does not. Both are skills you have to call. Impeccable only touches your design when you run an `/impeccable` command, and Humanizer only rewrites text when you run `/humanizer` or ask for it in plain words. The design hook is the one automatic piece, and it only surfaces findings after a UI file edit. Everything else is on you to invoke.

> **WATCH OUT**
>
> The Magic MCP you may have installed last year is dead. The old magic.21st.dev backend was replaced by the 21st MCP and every old Magic API key was reset. Your old mcp.json entry still works as a proxy, but only with a fresh key from https://21st.dev/mcp. If 21st returns nothing, the key is the first thing to check.

---

03 The four repos and the install commands

> **SOURCE CODE**
>
> Impeccable, design guidance for AI coding agents:
>
> **https://github.com/pbakaus/impeccable**

> **SOURCE CODE**
>
> 21st MCP (the repo is still named magic-mcp):
>
> **https://github.com/21st-dev/magic-mcp**

> **SOURCE CODE**
>
> Humanizer, the AI writing tell remover:
>
> **https://github.com/blader/humanizer**

> **SOURCE CODE**
>
> Claude Code Setup, in Anthropic's official plugin marketplace:
>
> **https://github.com/anthropics/claude-plugins-official/tree/main/plugins/claude-code-setup**

INSTALL-CLAUDE-CODE.SH

```
# 1. Claude Code Setup (inside Claude Code)
/plugin install claude-code-setup@claude-plugins-official

# 2. Impeccable (from your project root, in a terminal)
npx impeccable install
# then inside Claude Code:
/impeccable init

# 3. 21st MCP (get a free key at https://21st.dev/mcp first)
npx @21st-dev/cli@latest init --client claude

# 4. Humanizer (inside Claude Code, needs 2.1.142 or newer)
/plugin marketplace add blader/humanizer
/plugin install humanizer@humanizer
```

INSTALL-CODEX.SH

```
# Impeccable for Codex CLI (from the cloned repo)
git clone https://github.com/pbakaus/impeccable
cp -r impeccable/dist/agents/.agents your-project/
mkdir -p your-project/.codex
cp impeccable/dist/codex/.codex/hooks.json your-project/.codex/hooks.json
# or let the installer detect ~/.codex:
npx impeccable install --providers=codex

# 21st MCP for Codex
npx @21st-dev/cli@latest init --client codex
# or as a plugin:
codex plugin marketplace add 21st-dev/magic-mcp

# Humanizer for Codex and every other agent
npx skills add blader/humanizer --global
```

MCP.JSON (MANUAL 21ST MCP ENTRY)

```
{
  "mcpServers": {
    "21st": {
      "url": "https://21st.dev/api/mcp",
      "headers": { "x-api-key": "YOUR_21ST_API_KEY" }
    }
  }
}
```

SITES SHOWN IN THE VIDEO

- Impeccable docs
  https://impeccable.style
- 21st MCP setup and API keys
  https://21st.dev/mcp
- Humanizer on skills.sh
  https://skills.sh/blader/humanizer

Set it up in Claude Code or Codex

1. Start with Claude Code Setup, because it tells you what the other three should sit next to. Inside Claude Code run `/plugin install claude-code-setup@claude-plugins-official`, then type *recommend automations for this project*. It reads the codebase and hands back one or two picks per category. Nothing is written yet. This plugin is Claude Code only; on Codex, skip it.
2. Install Impeccable from a terminal at the project root with `npx impeccable install`. It lists the harness folders it found (`~/.claude`, `~/.codex`, a project-local `.cursor`) and asks project or global. Pick project for your first run. On Claude Code it also writes the design hook into `.claude/settings.local.json`. Reload Claude Code afterwards or the skill will not appear.
3. Run `/impeccable init` once. It inspects the project, asks only about gaps in product truth (audience, purpose, constraints, voice), and writes `PRODUCT.md`. Commit that file. On Codex the skill is invoked as `$impeccable` from `/skills`, repo-local installs live in `.agents/skills/`, and you need to open `/hooks` and approve the project hook once.
4. For 21st MCP, get a free key at https://21st.dev/mcp first. Then run `npx @21st-dev/cli@latest init --client claude` (or `--client codex`). If you go the plugin route instead, the plugin expects the key in the `API_KEY_21ST` variable. Ask Claude Code to add `export API_KEY_21ST=your_key` to your shell profile, then open a new terminal window so it loads.
5. Install Humanizer. In Claude Code 2.1.142 or newer: `/plugin marketplace add blader/humanizer` then `/plugin install humanizer@humanizer`, and it answers to `/humanizer:humanizer`. On Codex or any other agent: `npx skills add blader/humanizer --global`, and it answers to `/humanizer`. For a manual install, copy the repo into `~/.claude/skills/humanizer/`. The folder name must match the `name:` field in `SKILL.md`, which is `humanizer`.
6. First test. Run `/impeccable audit` on any page and you should see a numbered list of findings from the detector. Ask *search 21st for a pricing table* and you should get back component names with code you can request. Paste one AI-written paragraph into `/humanizer` and you should see three things: the first rewrite, a short critique, then the final version. If all three happen, you are set.

💡 QUICK TIPS

- Add the Impeccable ignore block to `.gitignore` before your first commit. It writes screenshots and cache files under `.impeccable/` that do not belong in the repo, while `PRODUCT.md` and `DESIGN.md` do.

- Node is only needed for the `npx impeccable` installer. The skill and hook run a self-contained binary downloaded once into `~/.impeccable/bin/`.
- If you use one Impeccable command constantly, pin it: `/impeccable pin audit` gives you `/audit` as a standalone shortcut.

04

The commands I actually run

PLAN A SECTION BEFORE BUILDING IT

```
/impeccable shape the [SECTION NAME] for [PAGE], audience is [WHO], goal is [ONE ACTION THEY SHOULD TAKE]
```

PULL A REAL COMPONENT INSTEAD OF INVENTING ONE

```
Search 21st for a [COMPONENT TYPE, e.g. testimonial grid] in React and Tailwind. Show me the top three, then get the code for the one that fits our DESIGN.md and adapt it to our tokens.
```

PRE-SHIP DESIGN PASS

```
/impeccable audit the [PAGE OR COMPONENT]

then

/impeccable polish the [PAGE OR COMPONENT]
```

HUMANIZE COPY IN MY VOICE

```
/humanizer

Here's a sample of my writing for voice matching:
[PASTE 2–3 PARAGRAPHS OF YOUR OWN WRITING]

Now humanize this text:
[PASTE THE AI TEXT]
```

HUMANIZE A WHOLE FILE WITHOUT TOUCHING CODE

```
Humanize the prose in [PATH/TO/FILE.md]. Leave code blocks, frontmatter, and link targets alone.
```

When it breaks

ERROR	FIX
/impeccable is not recognised after install	Reload Claude Code (or restart Codex). Newly installed skills only load on restart. On Codex, open /skills and type $impeccable instead of a slash command.
21st MCP returns nothing or an auth error	Old Magic keys were reset. Generate a fresh key at https://21st.dev/mcp and put it in API_KEY_21ST or the x-api-key header, then reconnect the MCP.
21st returns ai_subscription_required	Hosted generation is off for your account. Do not retry. Use search and get_component, then adapt the code with Claude Code yourself, or enable AI at 21st.dev/mcp and refresh the tool list.
/plugin install humanizer@humanizer fails	The plugin route needs Claude Code 2.1.142 or newer. Update Claude Code, or use npx skills add blader/humanizer --global instead.
Humanizer runs but the text still sounds like a robot	Give it a writing sample. Paste two or three paragraphs of your own writing above the text and it will match your rhythm, word choice, and punctuation.
Codex ignores the Impeccable design hook	Codex tracks trust per hook definition. Open /hooks and approve the project hook; you may need to approve again after npx impeccable update changes .codex/hooks.json.
Claude Code Setup recommended things but nothing changed	That is by design. The skill is read-only. Install the recommended plugins and MCPs yourself from its list.
Git shows a pile of .impeccable/ files	Add the impeccable-ignore block from the README to .gitignore. Keep PRODUCT.md and DESIGN.md tracked; everything under .impeccable/ is ephemeral.