

10 essential skills for Claude (and any programming agent)

This is a comprehensive guide to agent skills that can transform the performance of Claude Code, Cursor, Gemini CLI, and other AI programming assistants in a production environment.

This is a comprehensive guide to agent skills that can transform the performance of Claude Code , Cursor, Gemini CLI, and other AI programming assistants in a production environment.

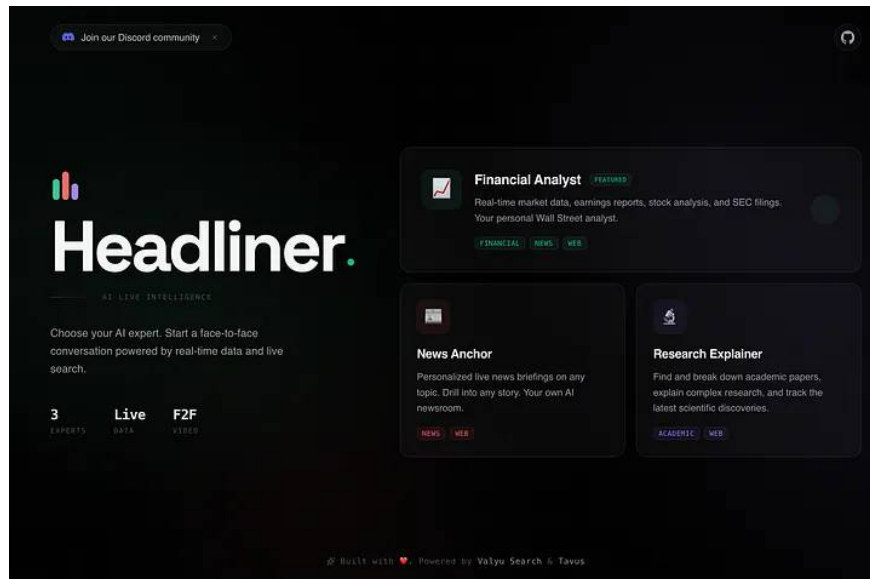
1. Frontend Design

The problem : Ask any LLM expert to build a landing page without instructions, and you'll get the same result almost every time: Inter font, purple gradient on a white background, minimal animation, grid cards. The result isn't wrong, it's just too mediocre.

This is what Anthropic calls 'distributed convergence'. Models are trained based on the statistical center of design decisions, meaning they reproduce the statistical center. Frontend design skills break that pattern.

How it works : Anthropic's official frontend design skills (over 277,000 installations as of March 2026) provide Claude with a design system and philosophy before he even starts writing code. It generates bold aesthetic choices, distinctive typography, purposeful color palettes, and animations that feel more intentional than just decorative.

The difference is striking. Without this skill, Claude would default to a safe, forgettable design. With it, you get components that look like they've been reviewed by a senior designer.



How to install:

```
npx skills add anthropics/claude-code - skill frontend-design
```

Alternatively, you can install it directly through Claude's plugin page. After installation, call it using `/frontend-design` and describe what you want to build.

2. Use a browser.

The problem : Programming agents can't see the web directly. They can write crawlers, but they can't run them. They can describe what a page looks like, but they can't interact with it. If your agent needs to fill out a form, log into a dashboard, extract dynamic content, or verify that a implemented feature actually works from start to finish, you've hit a roadblock. That's the main reason every application/system/infrastructure is being redesigned so that virtual assistants can interact more easily.

The skill browser-use solves this problem by allowing the agent to directly control the browser.

How it works : Skill browser-use connects Claude to a headless browser instance. The virtual assistant can navigate URLs, click on elements, fill out forms, extract content from rendered pages using JavaScript, take screenshots, and interact with complex web frontends, all as part of a natural language workflow.

This is different from using data extraction libraries. The virtual assistant doesn't need to understand the DOM structure beforehand. It navigates the web like a human: See, click, read, act.

How to install :

```
npx skills add https://github.com/browser-use/browser-use --skill browser-use
```

3. Code Review

The problem : Agents write code very quickly. They've become very good at it. However, they're rather poor at evaluating code. If left to them to automate, most programming agents generate code that passes the first read but misses more subtle issues: unnecessary abstract classes, duplicate logic, functions that do too much, inconsistent naming conventions, and a lack of exception handling.

The code works. But it might not maintain long-term stability. Handling complex codebases is what made you a skilled senior engineer in the first place, so you want to ensure all the code the AI agent writes is properly written and abstracted for better maintainability over time!

Code review skills transform quality assessment into a top priority, rather than a last resort.

How it works : Skill code-reviewer performs a structured review process on any code that the agent writes or modifies. It checks:

1. Logic can be simplified or extracted into reusable utilities.
2. These functions violate the principle of sole responsibility.
3. The patterns are inconsistent with the rest of the codebase.
4. Performance inefficiencies (unnecessary re-rendering, N+1 queries, blocking operations)
5. Dead code and unused imports
6. The naming convention doesn't convey intent.

Importantly, it not only detects problems but also fixes them. The review process takes place before the code is displayed to you.

How to install :

```
npx claude-code-templates@latest --skill development/code-reviewer
```

4. Remotion

The problem : Video conveys things that documents can't. But video production requires a completely different workflow, different tools, a different timeline, and a different team. Most developers release features without a demo video because the cost is too high.

How it works : Remotion is a React framework for creating videos programmatically. Instead of a timeline editor, you write components. The animation simply changes state over time. The Remotion agent skill in Claude Code translates natural language into functional Remotion components.

Workflow : describe what you want in the prompt, Claude generates the React/Remotion code, you preview it in Remotion Studio, and export it as an MP4.

```
npx skills add remotion/agent-skills
```

Then in Claude:

```
/remotion T?o video demo s?n ph?m dài 30 giây hi?n th? b?ng ?i?u khi?  
n API c?a chúng tôi v?i bi?u ?? ??ng và hi?u ?ng chuy?n ti?p
```

5. Skill Google Workspace (GWS)

The problem : Google Workspace has over 50 APIs. Gmail, Drive, Calendar, Docs, Sheets, Slides, Chat, Admin—each API has its own client library, OAuth flow, and REST endpoint. Building an agent to interact with Workspace previously required writing a large amount of integration code just to get started.

Google released `gws` in March 2026 and completely changed this.

How it works : `gws` is a command-line interface (CLI) that automatically discovers all Google Workspace APIs via Google's Discovery Service and displays them as a unified interface. It comes with a built-in MCP server; simply run a command and your AI agent will have full access to Workspace.

The numbers are real: gws achieved 4,900 stars on GitHub in its first 3 days. This is not a specialized tool.

```
# Cài ??t npm install -g @googleworkspace/cli # Kh?i ??ng máy ch? MCP v?
i các API ?ã ch?
n gws mcp -s drive,gmail,calendar,sheets npx skills add https://github.com/googl
?n có quy?n truy c?p tr?c ti?p vào các API này
```

6. Valyu: Real-time web search & specialized data access

The problem : Programmers are very good at handling source code. However, they are far less adept at handling real-world data because the real world is locked behind fee walls, proprietary databases, and specialized APIs that general-purpose search engines cannot access.

1. Building a financial research app? You need SEC reports.
2. Building a biomedical tool? You need PubMed and ChEMBL.
3. Building an economic analysis dashboard? You need FRED and BLS.

Without these data sources, programmers would create information that sounds plausible but is actually outdated or fabricated.

How it works : Skill Valyu connects developers to over 36 specialized data sources, providing high-quality document and web search through a single API. A single search command returns results from across the web AND sources such as SEC 10-K reports, PubMed, ChEMBL (2.5 million bioactive compounds), clinical trials, the FRED economic index, patent databases, and academic publishers.

Install Valyu's skills :

```
npx skills add https://github.com/valyuai/skills --skill valyu-best-practices
```

7. Antigravity Awesome Skills

The problem : Every agent skill issue you encounter has already been solved by others. But the solutions are scattered across GitHub repositories, blog posts, and Discord servers. You waste time writing `SKILL.md` files from scratch for things like pull request (PR) creation, debugging strategies, API design, and security testing, while actual tested versions already exist.

How it works : This is a community-maintained library of over 1,234 agent skills designed to work with all major AI programming assistants: Claude Code, Cursor, Gemini CLI, Codex CLI, GitHub Copilot, Antigravity IDE, etc. The skills follow the popular `SKILL.md` format, are organized by category, and can be installed with a single command.

Over 22,000 stars on GitHub. Over 3,800 forks. Updated to March 2026 (v7.3.0). This is the most comprehensive skill collection currently available.

Installation for Claude Code :

```
npx antigravity-awesome-skills - claude
```

8. PlanetScale Database Skills

The problem : Database-related work is where programmers make the worst mistakes. Schema design decisions cause problems six months later. Queries work fine for 100 rows but crash for 100,000 rows. Missing indexes are only discovered in production environments.

Programmers treat databases like any other code. They write something that runs and get on with their work.

PlanetScale's database skills change this by providing developers with in-depth knowledge of MySQL, serverless Postgres, branching workflows, and query performance right from the start.

How it works : PlanetScale operates a serverless MySQL-compatible database platform with a branching model that maps directly to Git: You create a database branch for each skill, merge it when finished, and never directly interfere with the production schema. The PlanetScale skill will guide the agents:

1. The schema design uses PlanetScale's foreign key and branching conventions.
2. Write queries that use indexes correctly (and report errors when they are not usable).
3. Use the `pscale` CLI to create branches, deploy requests, and manage migrations.
4. Treat schema changes as source code. This can include reviewing, reverse engineering, and merging.

Install the CLI and the PlanetScale skill :

```
# Cài ??t CLI pscale brew install planetscale/tap/pscale # Xác th?  
c pscale auth login # Cài ??t skill npx skills add planetscale/agent-skill
```

9. Shannon: Automated AI Penetration Testing Specialist

The problem : Security testing is a step that most development teams often skip, not because they don't care, but because it's expensive, slow, and requires specialized knowledge.

A traditional penetration test costs thousands of dollars and returns a PDF report after two weeks. Manual security reviews detect some vulnerabilities and miss others based on the evaluator's specific expertise. Meanwhile, the source code continues to be developed.

Shannon is an automated penetration testing agent that runs in your local or test environment, performs real-world attacks, and only reports vulnerabilities that it can demonstrate.

How it works : Skill Shannon integrates KeygraphHQ's Shannon, a white-box security testing framework that analyzes source code, maps attack surfaces, and performs realistic attacks on over 50 vulnerability types across 5 OWASP categories.

Notable benchmark results: 96.15% successful exploit rate on the XBOW security benchmark (100/104 attacks). This is not a tool scanning for potential issues, but rather an agent exploiting or not reporting the vulnerability.

Setting:

```
npx skills add unicodeveloper/shannon
```

10. Excalidraw diagramming tool

The problem : Decisions about architecture, system design, and data flow interpretation are all communicated in writing or during whiteboard discussions without anyone taking notes.

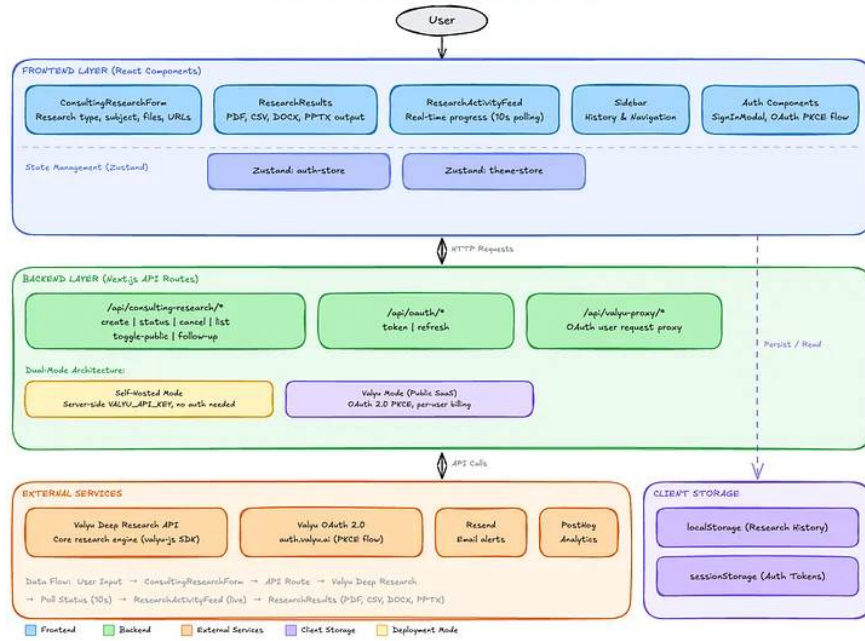
Code comments describe the function of something; diagrams show why it's structured that way. Most tools can describe architecture in text. Almost none can generate a diagram that visually represents the reasoning.

How it works : This skill generates production-quality Excalidraw diagrams from natural language descriptions. But what sets it apart from simpler diagramming tools is the design philosophy built into the skill itself:

1. The diagram represents the argument, not just for display. Each shape and group reflects the concept it represents. Branching structures for one-to-many relationships. Timeline layouts for sequential flows. Converging shapes for synthesis. The tool doesn't default to a uniform card grid, but maps the visual structure to the conceptual structure.
2. Concrete proof. The technical diagram includes actual code snippets and a real JSON payload embedded directly, not placeholder text.
3. Visual self-checking. This skill includes a Playwright-based rendering process. The agent generates an Excalidraw JSON file, renders it to PNG, examines its own output for layout issues (overlapping text, misaligned arrows, disproportionate spacing), and corrects errors before displaying the result. No more faulty diagrams.

ConsultRalph - System Architecture

AI-Powered Consulting Research Platform | Next.js 15 + React 19



How to install :

```
npx skills add https://github.com/coleam00/excalidraw-diagram-skill --skill excalidraw
```

You finished reading the article "**10 essential skills for Claude (and any programming agent)**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.