

Build Your Own JARVIS Voice Assistant

The full setup for a free, open source voice assistant that runs on your Claude Code subscription, no API keys required.

Why I built this

Most voice assistant projects want an OpenAI key, an ElevenLabs key, and a separate cloud bill before you say a single word to them. JARVIS skips all of that. It runs on the Claude Code subscription you already pay for, opens in a normal Chrome tab, and answers with a real Iron Man style holographic interface while it works.

Say "Hey Jarvis" and it searches the web, generates an image, takes a screenshot of your phone, or opens your GitHub notifications, depending on what you have connected. No dashboard, no monthly SaaS fee stacked on top of Claude Code.

The repo (adewaskar/jarvis on GitHub, 143 stars, MIT licensed as of this writing) puts the requirements in one line: a Claude Code subscription, plus two free things every computer already has, Node.js and Chrome. That is the whole list. An ElevenLabs key is a good add-on for a better voice, not a requirement.

The first boot is the part that sells it. Power-up plays a four-beat Iron Man style startup: an INITIATING SYSTEM status bar with a progress log, then concentric reticle rings resolving into J.A.R.V.I.S, then a suit schematic, then the triangular arc reactor lighting up with a startup sound under it. It is a genuinely fun thing to watch happen in a browser tab you built yourself.

01

How it actually works

1. The browser tab is the face and the voice. It shows the reactor UI (built with React, Three.js, and custom GLSL shaders), listens for the wake word with a local voice activity detector, and speaks the replies out loud.
2. A small Node process called the bridge is the brain. It runs the Claude Agent SDK, which spawns the real claude CLI as a child process, so the model answering you is Claude Code itself, running headless and authenticated off your existing login.
3. The two sides talk over a WebSocket on ws://localhost:8787. The bridge can spawn local MCP servers a browser tab never could, things like playwright for browser control, android for your phone, or elevenlabs for a better voice.

- Detection is local: an energy-based voice activity detector decides when you're speaking. It's instant and it's what makes barge-in work, speak while JARVIS is talking and he stops.
- Transcription has two tiers, chosen automatically at boot: ElevenLabs Scribe if a key is present, otherwise the browser's own SpeechRecognition, guarded by a heartbeat so it recovers when Chrome throttles it.
- Speaking uses the ElevenLabs voice when a key is present and the browser's built-in speechSynthesis otherwise. If a cloud call fails, it falls back to the browser voice automatically.

02

The one thing almost everyone gets wrong

It has to be a real Chrome or Edge window, not a preview pane.

WATCH OUT

Embedded preview panes, including the one inside code editors and inside Claude Code itself, block microphone access. The page loads and looks completely normal, and JARVIS just never hears you. Open <http://localhost:5173> in an actual browser window, not an inline preview.

THE SECOND GOTCHA

JARVIS reads MCP servers from your local `~/.claude.json` file. If you added Gmail, Calendar, or another connector through your claude.ai account instead, it isn't stored on disk, so the bridge can't see it and JARVIS will act like it doesn't exist. Anything you want JARVIS to reach has to be a server entry in `~/.claude.json`, the same way elevenlabs or playwright get added.

The permission gate isn't the only safeguard. The bridge's WebSocket only accepts local dev origins by default, its `/file`, `/img`, and `/media` endpoints validate the path and refuse private or loopback addresses so a remote image request can't be turned into a probe of your own network, and any HTML the model authors for the heads-up display gets sanitized with DOMPurify and a strict content security policy before it renders. All of it lives in `bridge/server.mjs`, so it's readable in one file.

03

The files

SOURCE CODE

The full repo, MIT licensed:

<https://github.com/adewaskar/jarvis>

PACKAGE.JSON (SCRIPTS)

```
"scripts": {
  "start": "node scripts/start.mjs",
  "dev": "vite",
  "bridge": "node bridge/server.mjs",
  "bridge:writes": "JARVIS_ALLOW_WRITES=1 node bridge/server.mjs",
  "setup": "node scripts/setup.mjs"
}
```

.ENV.EXAMPLE (THE ONE LINE WORTH ADDING)

```
# Optional. Gives JARVIS a better voice and Scribe transcription.
# Without it, everything still works on the browser's own speech.
ELEVENLABS_API_KEY=

# Also optional. Which model the bridge asks Claude Code to run.
JARVIS_MODEL=claude-opus-5
JARVIS_EFFORT=medium
```

- Web and search: exa, serper, serpapi
- Images and video: higgsfield, openrouter-image, palmier-pro
- Voice: elevenlabs
- Your phone: android
- The browser: playwright
- The UI itself, driven by voice through MCP tools the bridge exposes: ui_theme (accent and background colors), ui_reactor (shape, spin, intensity), ui_orbit (put an image in orbit), ui_chrome (show or hide rails and transcript), ui_effect (glitch, pulse, scan, shake, flash)

Set it up in Claude Code or Codex

1. Install Claude Code if you haven't: `npm install -g @anthropic-ai/claude-code`, then run `claude` once and log in. JARVIS reuses that login, no API key needed.
2. Install Node.js 20 or newer from nodejs.org, if you don't have it already.
3. Clone the repo: `git clone https://github.com/adewaskar/jarvis.git`, then `cd jarvis`.
4. Run `npm run setup`. It checks Node, Chrome, and your Claude Code login in plain language and tells you what's missing.
5. Optional: if you have an ElevenLabs API key and want the better voice, ask Claude to add `ELEVENLABS_API_KEY` to your shell profile, then open a new terminal so it's picked up. Otherwise skip this, JARVIS still works fully on the browser's own speech.
6. Run `npm install`, then `npm start`. This launches the bridge and the face together in one command.
7. Open `http://localhost:5173` in a real Chrome or Edge window, click INITIALISE, allow the microphone, and say "Hey Jarvis." A successful first run plays a four-beat startup sequence and the reactor lights up when it's listening.

💡 QUICK TIPS

- Prefer two terminals? Run `npm run bridge` in one and `npm run dev` in the other, then open `http://localhost:5173` yourself.
- If you get 'no such file or directory' on the clone step, you're not in the parent folder you meant to clone into. `cd` there first.
- The tool gate starts read-only on purpose. Search, generation, and lookups run freely, but anything effectful (sending, tapping, deleting, installing, paying) is denied until you start the bridge with `npm run bridge:writes`. Read `decideTool()` in `bridge/server.mjs` before you flip that on.
- Once it's running, a few keys are worth knowing: Space talks without the wake word, V cycles the browser voice, Escape stands JARVIS down, D opens the live diagnostics panel, and T runs a one-line audio self-test. Just speaking mid-reply interrupts him, that's the barge-in the local voice detector is built for.

04

The commands I actually run

DAILY BRIEF

Hey Jarvis, what's happening in AI this week?

GENERATE AN IMAGE

Hey Jarvis, generate an image of [SUBJECT].

CHECK MY PHONE

Hey Jarvis, take a screenshot of my phone.

RESTYLE THE INTERFACE

Hey Jarvis, make it [COLOR], hide the systems list, and put that last render in orbit.

TURN ON EFFECTFUL TOOLS (RUN THIS IN YOUR TERMINAL, NOT SPOKEN)

```
npm run bridge:writes
```

When it breaks

ERROR	FIX
No voice at all, JARVIS never responds	Confirm you're in a real Chrome or Edge window, not an embedded preview pane, and that you allowed the microphone when asked.
Bridge not reachable	Check that npm run bridge is still running in its terminal, and that nothing else is holding port 8787.
Can't tell if it's hearing you	Press D for the diagnostics panel, or T for a one-line audio self-test.
JARVIS answers but won't send, tap, delete, install, or pay for anything	That's the default read-only gate (decideTool()) in bridge/server.mjs). Restart the bridge with npm run bridge:writes to allow effectful tools.
JARVIS can't read your Gmail or Calendar	It only sees MCP servers listed in your local ~/.claude.json, not connectors added through your claude.ai account. Add a local MCP server for that if you want it.
Voice or transcription didn't upgrade after adding an ElevenLabs key	Restart the bridge so it re-reads the environment; /health only picks up the key at boot.

Ready to go deeper?

This playbook is the short version. Inside the hub you get the full build walkthroughs, the prompt library, and the systems I use to ship this in a weekend.

Join the Build →

<https://hub.digicuratoragency.com/join>