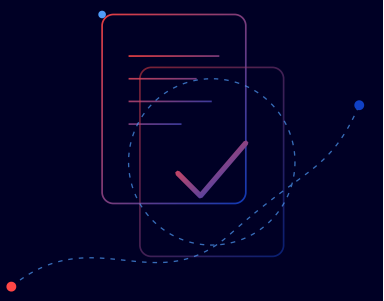
Setup guide · Open source on Claude Code

Set up AI Job Search on Claude Code

Find roles, score fit, tailor your CV and cover letter, and prep every interview round, from a private copy on your own machine.



5 tools to install · **6** slash commands · **1** private repo · **MIT** free and open source

01 Install the prerequisites

`Claude Code`	Signed in with a Claude Pro, Max or Team plan, or API credits
`Python 3.10+`	Runs the helper tools in the repo
`Bun`	Runs any job board CLIs you add
`LaTeX`	TeX Live, MacTeX or MiKTeX with `lualatex` and `xelatex`
`gh`	GitHub CLI, signed in

02 Make a private copy

```
gh repo clone ai-job-search-dev/ai-job-search
cd ai-job-search
git remote rename origin upstream
gh repo create ai-job-search --private \
  --source=. --remote=origin --push
```

Never fork for your own search. A fork of a public repo is always public, and setup writes your details into tracked files.

03 Build your profile

```
claude      # opens Claude Code in the repo
/setup      # run inside Claude Code
```

Drop your CV PDF, LinkedIn export and past applications into `documents/`, paste one CV in chat, or let it interview you. Rerun any time you add material.

Check LaTeX is ready, a PDF means yes:

```
cd cv && lualatex main_example.tex
```

04 Add job boards (optional)

Without a board, `/scrape` uses web search. To add one, copy its folder from **community-portals** into `.agents/skills/`, review the code, then run:

```
python3 tools/install_portal_clis.py
```

No board for your market? `/add-portal` builds one from its URL.

05 Run the workflow

`/scrape`	Searches your boards for roles that match your profile and removes duplicates
`/rank`	Scores every new posting against your profile and returns a shortlist with honest gaps
`/apply <url>`	Checks fit, drafts the CV and cover letter, gets a second agent's review, then compiles and checks the PDFs. Paste the job text if the page sits behind a login
`/interview`	Builds a prep pack for the next round, maps likely questions to your STAR stories and runs a mock
`/outcome`	Logs the result, archives what you sent and drafts follow ups for applications that went quiet

! Use it well

- Feed it detail. Projects, tools and measurable results per role make every application specific.
- It never invents experience. Gaps stay visible, so prepare for them.
- Read each CV and letter before you send it. You stay the one who applies.
- Postings are untrusted input. The integrity gate flags hidden instructions.