

Homework — IT Roles & Open Source

Estimated time: ~60 min · No code you have to run · A browser, a job-board account, and a text editor for the finale

Three exercises to close the course. The first two are *tool tours* — you go look at the actual open-source world and the actual IT job market, with the vocabulary you've built over the last eleven lessons. The third is the one and only **reflection task** of the whole course: a 200-word IT autobiography that asks you to name what surprised you, where you want to land, and what you'll learn next. Save that one. Re-read it in six months.

Before you start

Keep a one-page notebook open (paper or a text file). For the two tool tours, write the short answers under (a)/(b)/(c)/(d) per listing — you're not turning this in, the goal is to *look at the real thing* with the course's vocabulary in your head. For the autobiography, open whatever you actually write in (Notes, Obsidian, Google Doc, paper). There is **no code you need to run**. Free accounts on GitHub and one job board are sufficient as of 2026; the URLs and UI elements named below were correct as of the course's last review, so if a button has moved by the time you read this, pick the nearest equivalent.

The three tasks

Task 1 — Tour the open-source ecosystem (OSS shopping)

Open github.com/trending in your browser. At the top of the page is a **language filter**. Try it three times in a row — switch to **Python**, then **TypeScript**, then **Rust**. For each language, scroll the top 5 trending repos and read each repo's one-line description (the small grey text just under the repo name).

Then open opensource.guide/how-to-contribute/ in a second tab and read the section titled "*Finding a project to contribute to*". It's short — under five minutes.

Finally, pick any one of the trending repos you just looked at, click into it, and find the **Insights → Contributors** page (top-right tab on the repo, then *Contributors* in the left side-bar). Take a screenshot of the chart. Notice the **long-tail distribution**: a few people commit a lot, many people commit a little. That's the shape of almost every successful open-source project — and one of the points the theory page makes about how OSS actually works.

Self-check — answer all three:

- **(a)** Of the three languages (Python, TypeScript, Rust), which one has the most "*AI / LLM / agent*" projects in its top 5 today? (Likely Python — but check. The answer can shift week to week, and noticing the shift is the point.)
- **(b)** Pick one trending repo whose purpose you *completely understood* from the one-line description alone, and one whose purpose you *didn't*. Write down both names. The first one is evidence that the vocabulary from this course is now in your head. The second one is a pointer at where to look next.
- **(c)** For the repo whose purpose you didn't understand, click in and read the README for five minutes. Does it become clear? If not, that's fine — not every project explains itself well, and reading READMEs critically is itself a skill. Write one sentence on what the project does, in your own words.

What you just saw

The trending page is a daily snapshot of what the open-source world is currently excited about. The contributor distribution is the same shape regardless of project size — a small core, a wider ring of regulars, and a long tail of one-time contributors. If you ever want to contribute to OSS yourself, the one-time-contribution slot is exactly where you'd start.

✓ You have now seen what the open-source ecosystem looks like from the inside — not as a slogan, but as a live page of real projects with real contributor graphs.

Task 2 — Tour the IT job market through the lens of this course

Open linkedin.com/jobs (or indeed.com, or welcometothejungle.com if you're in Europe). For each of the following **5 job titles**, search and open *one* real listing — whichever one happens to look representative, you're not job-hunting, you're sampling:

1. **Frontend Engineer**
2. **Site Reliability Engineer (SRE)**
3. **Data Engineer**
4. **AI Engineer**
5. **Security Engineer**

For each of the 5 listings, work through the four short prompts below. You will end up with one short table of notes — that table is your second map of the IT landscape, overlaying the one you started this course with in Lesson 01.

Self-check — answer all four per listing:

- **(a)** Write down the **3 most-named technologies** in the listing's *Requirements* or *Must-haves* section. Names only — no commentary.
- **(b)** For each of those three, write down **which lesson of this course** you saw it in. Examples to calibrate: "React" → L04; "Kubernetes" → L08; "Postgres" → L07; "LLM / RAG / vector database" → L11; "SIEM / EDR / SOC" → L09. If a named technology didn't appear anywhere in this course, write "*new*" next to it — that's normal and useful information.
- **(c)** Find **one bullet point in the listing that you didn't recognize at all**. Copy it verbatim into your notes.
- **(d)** Paste that bullet into Claude or ChatGPT with the prompt: "*Explain this to someone who's just finished a 12-lesson IT survey course.*" Read the explanation. Notice that you can now *understand* the explanation — even though you didn't recognize the bullet cold — because the vocabulary that the explanation uses is the vocabulary you spent twelve lessons building. That compounding is the actual point of this exercise.

One listing per title — resist the rabbit hole

It's tempting to open ten listings per role to "*compare*". Don't. One representative listing per title is enough for the mapping exercise; ten will swamp you without changing the conclusion. If you find yourself wanting to bookmark roles for later, open a separate *future_me* note — but finish the homework first.

✓ You have now read 5 real job listings spanning 5 distinct IT clusters, mapped their requirements back to specific lessons of this course, and used an LLM to translate one unfamiliar bullet per listing. The map of IT in your head is now anchored to actual employer language, not just curriculum.

Task 3 — Write your 200-word IT autobiography (the course finale)

This is the course's only reflection task, and it sits here at the very end on purpose. Open a plain text editor, a Notes app, a Google Doc, or paper — whatever you actually write in. Write approximately **200 words** answering, in this order, the three short paragraphs below.

1. **One paragraph — which 2–3 lessons in this course surprised you?** For example: maybe you had no idea the cloud was that *physical* (L06); maybe you had no idea databases mattered as much as they do (L07); maybe you didn't realize *AI Engineer* and *ML Engineer* are actually different roles (L11 / this lesson); maybe you found yourself reading the security lesson three times (L09). The word "*surprised*" is doing the work here — the goal isn't "*which was your favourite*", it's "*which made you sit up*". Surprise is what your interest looks like in real time.
2. **One paragraph — based on those surprises, where in IT do you most want to land?** Use the role clusters from this lesson's theory: *engineering*, *infrastructure*, *data-and-AI*, *security*, or one of the *non-coder doors* (Product Manager, Designer, QA / SDET, Tech Writer, DevRel, IT Support, Project Manager, Tech Recruiter). Be honest — "*I want to land in a non-coder door*" is a perfectly normal and well-paid answer, and it's the one this course works hardest to make visible.

3. **One paragraph — which sibling course on this platform will you take next, and when?** Pick one from slide 11 of this lesson, or from the theory page's "Reading on" list — the candidates are *Claude & LLMs*, *Building MCP Servers*, *Building Your Own Agents*, *hashcat*, *helm*, and *linux-net*. Give it a date. Put the date in your calendar before you close this page.

Self-check: the 200 words are written; they name at least **2 specific lessons** of this course; they pick at least **1 sibling course** by name; and there is a **concrete date** for when you'll start it (and that date is in your actual calendar, not just on the page). This is the autobiography. *There is no answer key* — reflection tasks deliberately don't have one. Save the file somewhere you'll find it again, and re-read it in 6 months to see which parts came true.

Why we end the course with this

Every other task in every other lesson had a tool tour or a lab to anchor it. This one doesn't, on purpose. Twelve lessons of vocabulary only become *your* vocabulary once you've said, out loud and in writing, which parts of it grabbed you. The 200 words you write today are a small artefact that converts "*a course you took*" into "*a direction you chose*" — which is the whole reason this course exists.

What "done" looks like

You're done with this homework — and with the course — when, in your notebook, you have:

- a screenshot of one trending repo's *Insights* → *Contributors* graph and short notes (a)–(c) per language sampled from Task 1 (OSS tour)
- a small table of 5 job listings with (a)–(d) filled in per listing from Task 2 (role tour)
- a saved file of approximately 200 words from Task 3 — the autobiography — with at least 2 lesson references, 1 named sibling course, and a date on your calendar

Where this goes next

Module 5 ends here, and so does the IT Landscape 2026 course. The skills you just exercised — reading real OSS projects with the right vocabulary, reading real job listings with the right vocabulary, and naming out loud what you actually want next — are the daily skills of *navigating* IT rather than being intimidated by it. The technology will keep moving fast; the habit of looking at the real thing and the habit of choosing your own next step are what stay useful. See you on whichever sibling course you put on your calendar.