

Lesson 01 — The IT Map

A tour of what IT people work on — and why nobody can do all of it.

IT Landscape

Module 1

Why you're here

This course is built for three audiences:

- **Career switchers** — you're thinking about moving into tech and want a map before the deep dive.
- **Non-tech managers** — you work next to engineers and want to follow the conversation.
- **Motivated relatives of techies** — you want to understand what they actually do all day.

Goal: leave this course knowing which area you'd want to learn more about.

IT is a city

Think of IT as a city.

- **Frontend** = storefronts
- **Backend** = warehouses
- **Networking** = roads
- **Databases** = libraries
- **Security** = police
- **Cloud** = the utility grid

Every business runs on the whole city. Most engineers live in one or two neighborhoods.

The 7 neighborhoods

1. **Frontend** — what the user clicks: web pages, mobile apps.
 2. **Backend & APIs** — the brain behind the storefront.
 3. **Databases & Data** — where state lives.
 4. **Networking & The Internet** — how anything talks to anything else.
 5. **DevOps & Cloud** — how code gets to real users, 24×7.
 6. **Security** — keeps everyone else's mistakes off the front page.
 7. **AI / ML** — the newest neighborhood, re-shaping the rest.
-

What "writing software" actually looks like

A typical engineer's day:

- Reading existing code
- Writing new code (a smaller share than you'd think)
- Reviewing teammates' code
- Fixing bugs
- Designing how things connect
- Meetings about all of the above

Not "typing furiously into a black screen."

The tool landscape

Names you'll meet in any modern team:

- **GitHub** — where the code lives.
- **VS Code** — the editor most people open.
- **Slack** — where the team talks.
- **Jira** — where the work is tracked.
- **AWS Console** — where the cloud bill lives.
- **ChatGPT / Claude** — the AI assistant in every workflow.
- **Docker** — the box code ships in.
- **Terraform** — infrastructure described as text.

Most IT workers see most of these **every week**.

Languages aren't religions

Python · JavaScript · Go · TypeScript · Rust · SQL

A tool, not a tribe. Most engineers use 2–4 languages daily and pick up new ones over a career.

What an "engineer" actually does

~5%

Writing brand-new code.

~95%

Reading code, debugging, reviewing pull requests, designing how the pieces fit, and talking to other humans.

Roles you'll hear named

- **Software Engineer** — the generalist who ships features.
- **DevOps / SRE** — keeps the production system alive.
- **Data Engineer** — builds the pipelines that move data.
- **Data Scientist** — turns data into answers.
- **ML Engineer** — puts models into production.
- **Security Engineer** — keeps everyone honest.
- **Product Manager** — decides what gets built.
- **UX Designer** — decides how it feels to use.

Lesson 14 expands on every one of these.

The 2026 reality

- **Every team uses cloud.** Owning your own servers is now the exception.
 - **Every team has AI assistants in the loop.** Code, docs, reviews, customer support — touched by an LLM.
 - **"Full-stack solo" is back.** AI handles a lot of glue work, so one person can ship more on their own than five years ago.
-

How to read this course

- **12 lessons**, ~45 min each.
 - **Slides** = the talk.
 - **Theory** = the read.
 - **Homework** = the tool-tour.
 - **No code.** The goal is vocabulary + a mental model, not job-readiness on day one.
-

Recap

- IT is many fields, not one.
- Engineers spend most of their time reading and designing — not typing.
- You don't need to learn it all to be useful.

Coming up: Lesson 02 — what actually happens when you visit a URL.

