

EQUIPPD · FREE RESOURCE

The Tech Career Map

Thirty ways into the industry, and which ones involve writing code.

A plain-language reference to the roles that actually exist,
what they pay, and how people get in.

Built for the Equippd community · equippd.io
Free to use, share and print.

“Working in tech” is not one job

Most people picture a single role — software engineer — and then decide the whole industry is or is not for them based on how they feel about that one job. That is a costly mistake in both directions. Plenty of people who would have thrived in tech ruled it out because they did not enjoy their first programming class. Others grind at engineering when the role they actually wanted was three doors down.

This is a map of what is out there. The coding column is the one most people need.

How to read this

“Heavy” means writing code is the core of the job. “Some” means scripting, queries or configuration — real technical skill, but not software engineering. “Little” means the job is technical in domain but not in daily output. All three categories contain well-paid careers.

Building software

Role	What the job actually is	Coding
Frontend Engineer	The part of the product people see and click. UI, state, accessibility, performance.	Heavy
Backend Engineer	Servers, APIs, business logic, and the data underneath it all.	Heavy
Full-Stack Engineer	Both of the above, usually at smaller companies where you cannot specialise.	Heavy
Mobile Engineer	Native iOS or Android apps, or cross-platform via React Native or Flutter.	Heavy
Game Developer	Engines, gameplay systems, graphics, tools. Notoriously demanding hours.	Heavy
Embedded Engineer	Software that runs on hardware — cars, medical devices, appliances.	Heavy
Data Engineer	The pipelines and infrastructure that make data usable by everyone else.	Heavy
ML Engineer	Taking models and making them run reliably in production at scale.	Heavy

Running and securing systems

Role	What the job actually is	Coding
DevOps Engineer	Build, test and release automation. The engineer that other engineers depend on.	Heavy
Site Reliability Engineer	Keeping large systems up, fast and observable. Automation-first operations.	Heavy
Cloud Engineer / Architect	Designing what runs where in AWS, Azure or GCP, and what it costs.	Some
Security Engineer	Building defences, hardening systems, responding to incidents.	Some
Penetration Tester	Attacking systems with permission, to find what a real attacker would.	Some
Security Analyst (SOC)	Monitoring alerts, triaging incidents, hunting threats. Common entry point.	Little
Network Engineer	Designing and running the network the whole business sits on.	Little
Systems Administrator	Servers, identity, backups, the things everyone notices only when broken.	Some
Database Administrator	Performance, availability and integrity of the data store.	Some

Data and analysis

Role	What the job actually is	Coding
Data Analyst	Answering business questions with data. SQL is the core skill, not Python.	Some
Data Scientist	Statistical modelling and experimentation. Heavier maths than analysis.	Some
Business Intelligence Developer	Dashboards and reporting layers the whole company runs on.	Some
Analytics Engineer	The bridge between raw pipelines and analyst-ready tables. Growing fast.	Some

Product, design and quality

Role	What the job actually is	Coding
Product Manager	Deciding what gets built and why. Judgement, communication, prioritisation.	Little
Technical Program Manager	Coordinating complex delivery across multiple teams.	Little
UX Designer	How the product works — flows, structure, usability research.	Little
UI / Product Designer	How it looks and feels. Visual systems, components, interaction.	Little
UX Researcher	Talking to users properly and turning that into decisions.	Little
QA Engineer	Finding what is broken before customers do. A real on-ramp into engineering.	Some
SDET / Automation Engineer	Writing the automated test suites. QA that pays like engineering.	Heavy

Technical, but not building the product

Role	What the job actually is	Coding
Solutions Engineer	Technical half of the sales conversation. Demos, integrations, objections.	Some
Customer / Implementation Engineer	Getting customers live and unblocking them. Deeply technical support.	Some
Technical Writer	Documentation, API references, guides. Rare skill, well paid, widely needed.	Little
Developer Advocate	Teaching, speaking, writing and building demos for a developer audience.	Some
IT Support / Helpdesk	The most common genuine entry point into the industry. Undervalued.	Little
Technical Recruiter	Finding and assessing engineers. Needs enough fluency to hold the conversation.	Little

Choosing between them

Ignore titles for a moment and answer these instead. They predict fit far better than any job description.

- **Do you want to build the thing, or decide what the thing should be?** Building points at engineering; deciding points at product.
- **Do you prefer a problem that is solved and closed, or one you keep improving?** Closed points at project-based work; ongoing points at operations, SRE, or platform.
- **Do you get energy from people or drained by them?** Solutions engineering, advocacy and product are relentlessly social. Backend and data are far less so.
- **Do you like being the person who finds what is wrong?** That is security and QA, and both are chronically short of good people.
- **Do you want to explain things?** Technical writing and developer advocacy pay well and have far less competition than engineering.

The most useful thing on this page

Almost every role here can be reached from almost every other one. IT support to security analyst is a well-worn path. QA to SDET to engineer is another. Analyst to analytics engineer is another. You are not choosing a life sentence — you are choosing a starting door, and the first one only has to be open.

If a role here interests you, go read three real job postings for it before anything else. Job descriptions tell you the actual required skills far more honestly than any careers guide, including this one.