Stack Overflow Developer Survey:
Current Usage, Future Trends & Demographics

IBM Data Analyst Capstone • 2025-08-29

Prepared by:Fatemeh Kermani

Outline

- Executive Summary
- Introduction (Purpose, Audience, Value)
- Methodology (Data Sources & Wrangling)
- Programming Languages - Current Top 10
- Programming Languages - Next-Year Trends
- Databases - Current Top 10
- Databases - Future Demand
- Job Postings(job-postings.xlsx)
- Dashboard - Current Technology Usage
- Dashboard - Future Technology Trends
- Dashboard - Demographics
- Insights from Dashboards
- Overall Findings & Implications
- Conclusion
- Appendix

Executive Summary

- **Programming languages (current):** JavaScript/TypeScript lead everyday use in your chart, with C# and Python forming the next tier.
- **Programming languages (next year):** Intent clusters around Python and modern typed JS (TypeScript); these show the clearest “Want > Have” signal for upskilling.
- **Databases (current):** PostgreSQL and MySQL dominate, followed by Microsoft SQL Server and MongoDB; SQLite and MariaDB are also visible, with Redis appears in the long tail.
- **Databases (next year):** Interest continues to favor PostgreSQL/MongoDB and moves toward managed/cloud-friendly options; hands-on experience lags stated intent.
- **So what:** Prioritize a short learning sprint on **Python** and **TypeScript**, and build a small project on **PostgreSQL/MongoDB** to close the gap quickly.

Introduction

- **Purpose:** Present clear signals from the Stack Overflow Developer Survey to guide what to learn, what to build, and where to focus hiring.
- **Scope:** Current usage vs. next-year intent for programming languages and databases, plus job-market signals
- **Audience:** Students and stakeholders in analytics/engineering/product who make training, tooling, and recruiting decisions
- **Value:** Turn charts into actions-prioritize skills with the **largest Want-Have gaps** and align projects with roles that show the most postings
- **Limitations (brief):** Self-reported survey data; sample bias; next-year “intent” is not guaranteed adoption

Methodology

- **Data sources:** Stack Overflow Developer Survey (course dataset) and Job-posting.xlsx for role counts.
- **Cleaning:** Dropped blanks and duplicates, fixed label inconsistencies (e.g., “MS SQL Server” => “Microsoft SQL Server”), trimmed extra spaces.
- **Multi-select columns:** Turned the multi-choice fields (LanguageHaveWorkedWith, LanguageWantToWorkWith, DatabaseHaveWorkedWith, DatabaseWantToWorkWith) into one selection per row so we could count them properly.
- **How we ranked Top 10:** Counted responses, converted to shares (% of respondents), sorted descending, and kept the top ten for each chart.
- **Gap metric:** For each technology we computed Want − Have to spot the biggest upskilling opportunities
- **Job postings chart:** Read Job-posting.xlsx and plotted roles in descending order of postings.
- **Tools:** Prepped data in Python (Pandas/Matplotlib); built dashboards in Cognos/Looker Studio with three tabs (Current, Future, Demographics)
- **Consistency & fairness:** We kept the same filters, names, and sorting across all charts so the comparisons stay fair and easy to read
- **Limitations:** This is self-reported survey data from a sample of the community; what people plan to use next year doesn’t always turn into real adoption

Programming Languages (Current Top 10)

- **JavaScript** and **TypeScript** rank #1 and #2 by current use.
- **C#** and **Python** make up the next tier in everyday development.
- The rest of the top ten sit in a smaller long tail.
- **Takeaway:** Keep JS/TS as core skills and maintain working proficiency in C# and Python for backend/data work.

Programming Languages (Next-Year Trends)

- **Top intent:** Python and TypeScript attract the most “want to use next year.’’
- **Momentum:** Python’s interest grows across roles; TypeScript keeps rising with modern JS stacks.
- **Steady vs softening:** JavaScript stays high and steady; C# sits a bit behind the top cluster for forward intent.
- **Skill gaps (Want – Have):** Biggest gaps show up for Python and TypeScript-clear targets for upskilling.
- **Action:** Run a short 2–3 week sprint: one small Python project (data/automation) and one TypeScript project (frontend or API). Add both to the portfolio

Databases (Current Top 10)

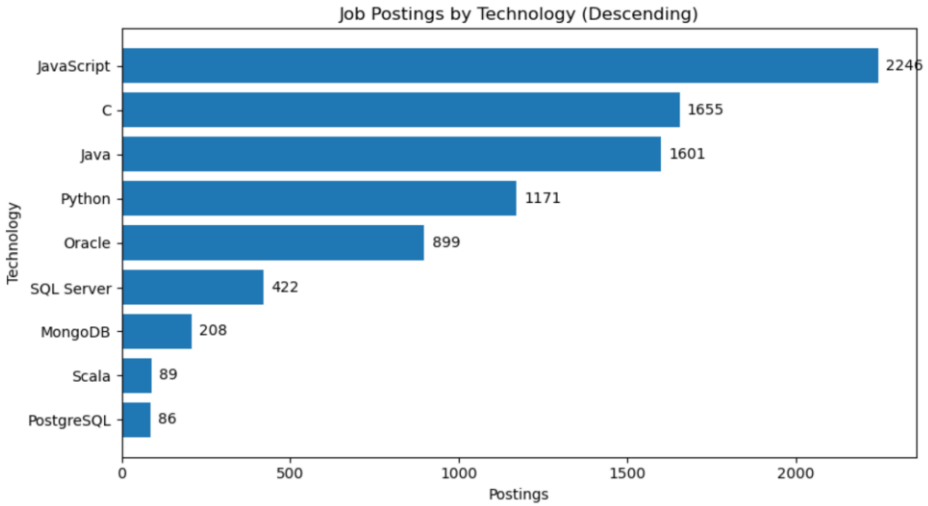
- **PostgreSQL** and **MySQL** lead current use.
- **Microsoft SQL Server** and **MongoDB** make up the next cluster.
- **SQLite** and **MariaDB** show up for lightweight/embedded or legacy needs; **Redis** appears in the long tail.
- **Takeaway:** Standardize on **PostgreSQL/MySQL** for most workloads; keep playbooks for **SQL Server** and **MongoDB** where they fit best

Databases (Future Demand)

- **Top intent:** PostgreSQL and MongoDB attract the strongest “want to use next year.
- **Momentum:** Interest shifts away from purely on-prem toward managed/cloud-friendly databases
- **Skill gaps (Want – Have):** Biggest gaps sit with PostgreSQL and MongoDB-prime targets for hands-on practice.
- **Action:** Spin up a small sandbox-one relational project on PostgreSQL and one document-store project on MongoDB

Job Postings by Technology (Descending)

- **Top technologies:** JavaScript (2,246), C (1,655), Java (1,601), Python (1,171).
- **Distribution:** Sharp drop after the top group; the rest form a long tail (Oracle ~899, SQL Server ~422, MongoDB ~208, Scala ~89, PostgreSQL ~86).
- **Takeaway:** Showcase JavaScript plus one backend language (C/Java/Python) in projects, and include a database example that mirrors demand.



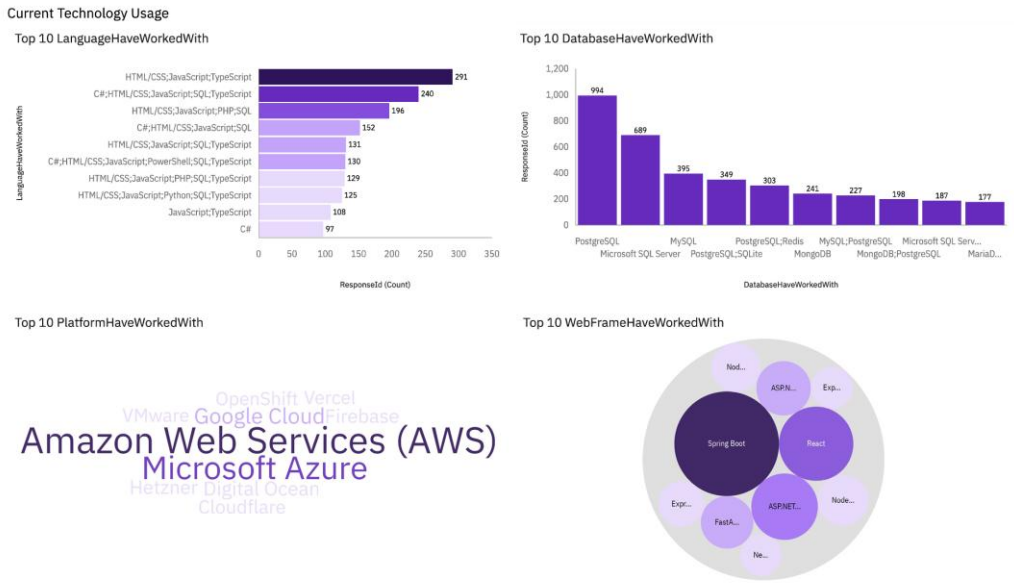
Methods ↔ Findings Cross-Reference

- **Multi-select → Rankings:** Splitting the multi-choice fields let us build true Top-10 lists in Slides 6–9.
- **Cleaning → Fair comparisons:** Label fixes and consistent casing/filters keep the charts apples-to-apples.
- **Gap metric → Actions:** The Want − Have metric directly drove the actions on Slides 7 and 9.
- **Job postings tie-in:** The postings chart supports the earlier language signals (JavaScript, Java, Python).
- **Dashboards ↔ Slides:** The three dashboard tabs (Slides 12–14) use the same logic, so screenshots line up with the results.

See S5 for methods; see S6–10 for results

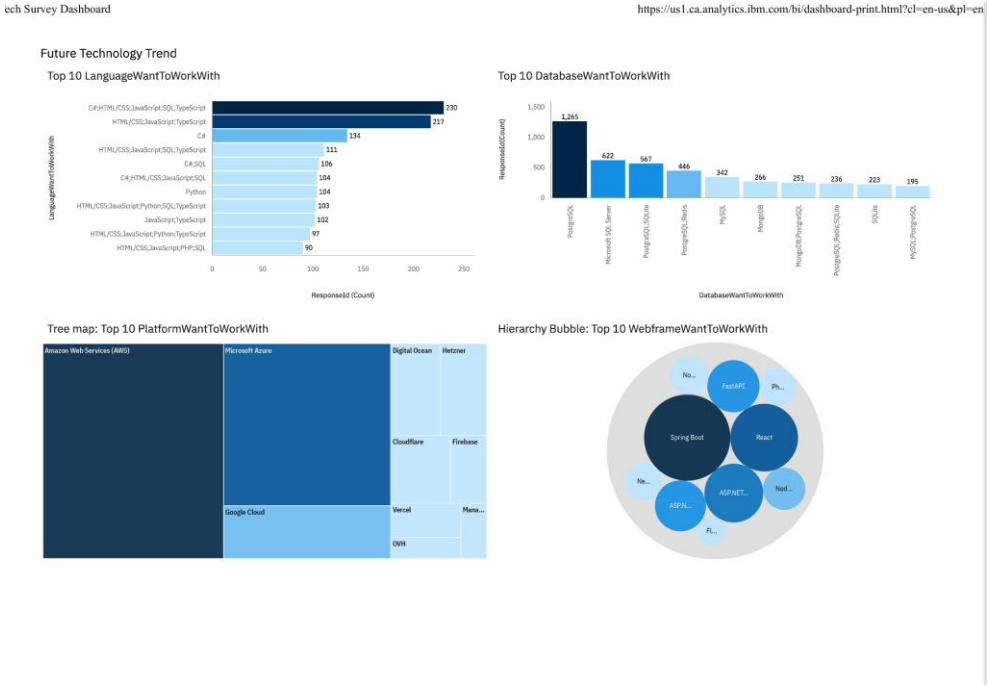
Dashboard: Current Technology Usage

- **What's shown:** Current usage across languages, databases, platforms, and web frameworks (Top-10 in each chart).
- **Languages:** JavaScript + TypeScript + HTML/CSS appear in most stacks; C# and Python are also common.
- **Databases:** PostgreSQL, SQL Server, and MySQL are the main workhorses; MongoDB shows up alongside them.
- **Platforms:** AWS is the default choice, with Azure next; Google Cloud is present but smaller.
- **Web frameworks:** React and Spring Boot stand out in the framework chart



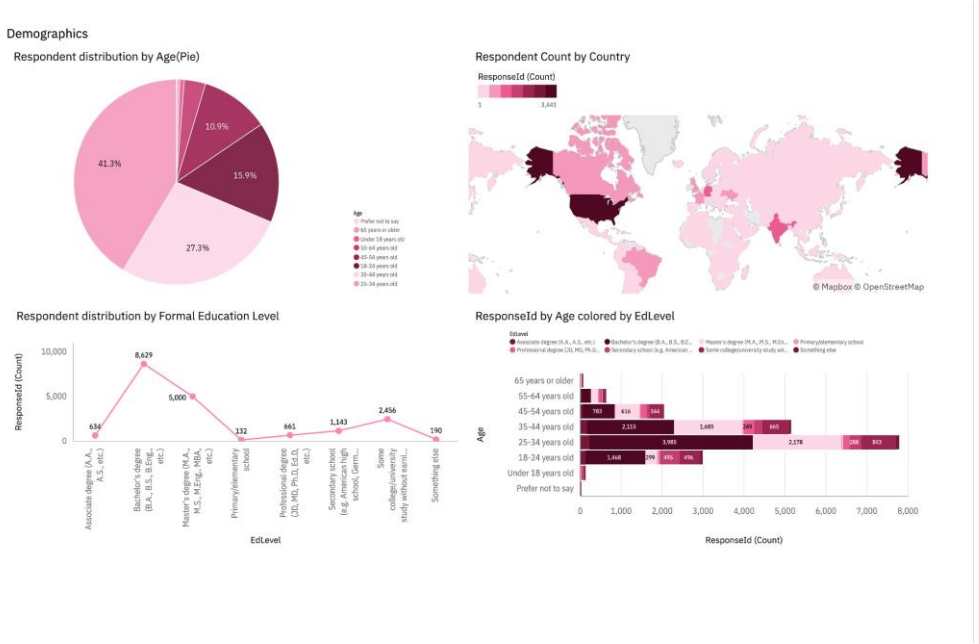
Future Technology Trends

- **What's shown:** Next-year intent across languages, databases, platforms, and web frameworks (Top-10 each).
- **Languages:** JS/TypeScript stacks and **C#** sit at the top; **Python** is a clear riser
- **Databases: PostgreSQL** is #1 by a wide margin, with **Microsoft SQL Server** next; **MySQL** and **MongoDB** follow.
- **Platforms: AWS** leads, **Azure** is second; **Google Cloud** is visible, with smaller interest in DigitalOcean, Vercel, Cloudflare, OVH, Hetzner.
- **Web frameworks: Spring Boot** and **React** stand out; **FastAPI**, **ASP.NET**, and **Node** show up as secondary choices.
- **Skill gaps → action:** Upskill on **TypeScript/JS + Python** and get hands-on with **PostgreSQL/MongoDB**; plan one small project for each



Dashboard: Demographics

- **What's shown:** Age bands, country map, education levels, and the age × education cross-tab.
- **Age:** Responses cluster around early-career bands; there's a taper into +35.
- **Geography:** A few countries account for most responses; regional mix can shape which tools get adopted first.
- **Education:** Bachelor's is most common; Master's is sizable; self-taught/bootcamps also appear.
- **Cross-cut:** Younger groups skew Bachelor's/undergrad, while graduate degrees rise in older bands



Insights from Dashboards

- **Now vs next:** Today's stack is JS/TS + relational DBs; next-year intent adds momentum for **Python** and **PostgreSQL/MongoDB.**
- **Skill gaps worth closing:** The biggest Want – Have gaps sit on **Python**, **TypeScript**, and **PostgreSQL** => fastest ROI for upskilling.
- **Framework reality check: React** and **Spring Boot** anchor most projects; **FastAPI/ASP.NET** look like smart second bets.
- **Cloud pattern: AWS** is the default, **Azure** second; having "AWS + one other" in the toolkit covers most cases.
- **Demographics lens:** A young, US-leaning sample helps explain the strong JS/React and AWS presence; results may weight early-career preferences.
- **So what:** Prioritize one portfolio project in **Python + PostgreSQL** and one in **TypeScript/React**; deploy on **AWS** to match market signals

Overall Findings & Implications

- **Core today:** JS/TS + HTML/CSS dominate; PostgreSQL/MySQL/SQL Server are the workhorse DBs; React/Spring Boot show up often; AWS leads cloud use.
- **Where interest is heading: Python** and **TypeScript** gain momentum; **PostgreSQL/MongoDB** trend up; cloud-first/managed tools grow.
- **Skill gaps:** Largest Want − Have on **Python**, **TypeScript**, and **PostgreSQL** =>fastest wins from focused upskilling.
- **Hiring signal:** Postings favor JS plus one backend (Java/Python/C) with SQL and a major cloud (AWS/Azure).
- **Implications:** Short term-ship projects in **Python + PostgreSQL** and **TypeScript/React** on **AWS**. Medium term-document when to choose **PostgreSQL vs MongoDB**. Long ter--track intent vs adoption in the next survey.
- **Limits & risk:** Self-reported data; intent is not adoption; de-risk with small pilots and measure usage

Conclusion

- **What we know now:** JS/TS lead current use; Python is rising; PostgreSQL/MySQL/SQL Server are core; AWS leads cloud use.
- **What to do next:** Ship two small portfolio projects: (1) Python + PostgreSQL, (2) TypeScript + React on AWS.
- **Hiring signal:** Most postings ask for JS plus one backend language, SQL, and a major cloud.
- **Risks and limits:** The survey is self-reported and intent is not adoption. Start with small pilots and measure actual usage.
- **Closing line:** These choices match both the survey signals and the job market right now.

Appendix

- **Extra charts:** Full Top 10 tables not shown in the main slides (languages, databases, platforms, frameworks).
- **Job postings data:** Table with Technology and Postings used for Slide 10.
- **Data dictionary:** Brief notes for LanguageHaveWorkedWith, LanguageWantToWorkWith, DatabaseHaveWorkedWith, DatabaseWantToWorkWith.
- **Method notes:** Filters applied, label cleanup rules, and the Want - Have formula.
- **Reproducibility:** Tool versions (Python, Pandas, Matplotlib) and the dashboard tool; date of data pull.