Global Developer Survey Insights: Current Usage, Future Trends & Demographics

Priyadarshini Gupta

5th September, 2025

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

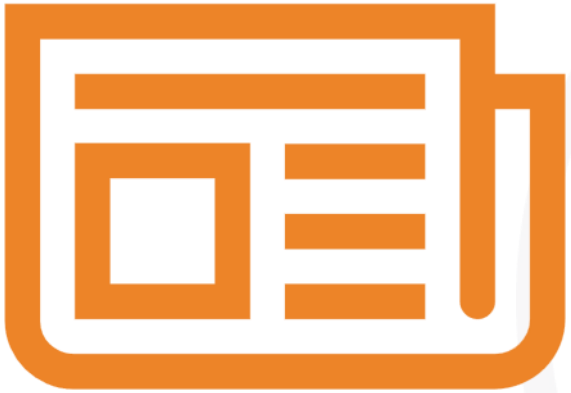
EXECUTIVE SUMMARY

- The dashboard analyzes survey data to reveal **current technology usage, emerging trends, and respondent demographics** across programming languages, databases, platforms, and web frameworks.
- **JavaScript, Python, and SQL** dominate current language use, while **MySQL** and **PostgreSQL** lead database adoption.
- Framework preferences highlight **Node.js, React, and ASP.NET Core** as key tools, with notable adoption of **Next.js** and **Express**.
- Future interest shifts toward **Next.js, FastAPI, and Svelte**, indicating a move toward **modern, lightweight, and performance-oriented stacks**.
- Demographics show a respondent base largely in the **25–34 age range**, with a high proportion holding **bachelor's or higher degrees** and representation from **multiple countries**.
- Insights suggest that **developers intend to deepen expertise in familiar technologies while exploring innovative frameworks**, guiding organizations toward **strategic upskilling and tool adoption**.

INTRODUCTION

- **Purpose**
  - Turn survey data into clear insights on **tech use, trends and demographics**
  - Support **planning, skills focus and decision-making** with visuals
- **Value**
  - Converts raw data into **quick, actionable insights**
  - Guides **curriculum, recruitment & tool adoption**
  - Shows **hands-on skills** in cleaning & visualizing data
  - Highlights **current vs. future technology shifts**
- **Target Audience**
  - **Faculty/evaluators** reviewing dashboard work
  - **Organizations** planning **training and hiring**
  - **Developers and tech leads** tracking stack trends

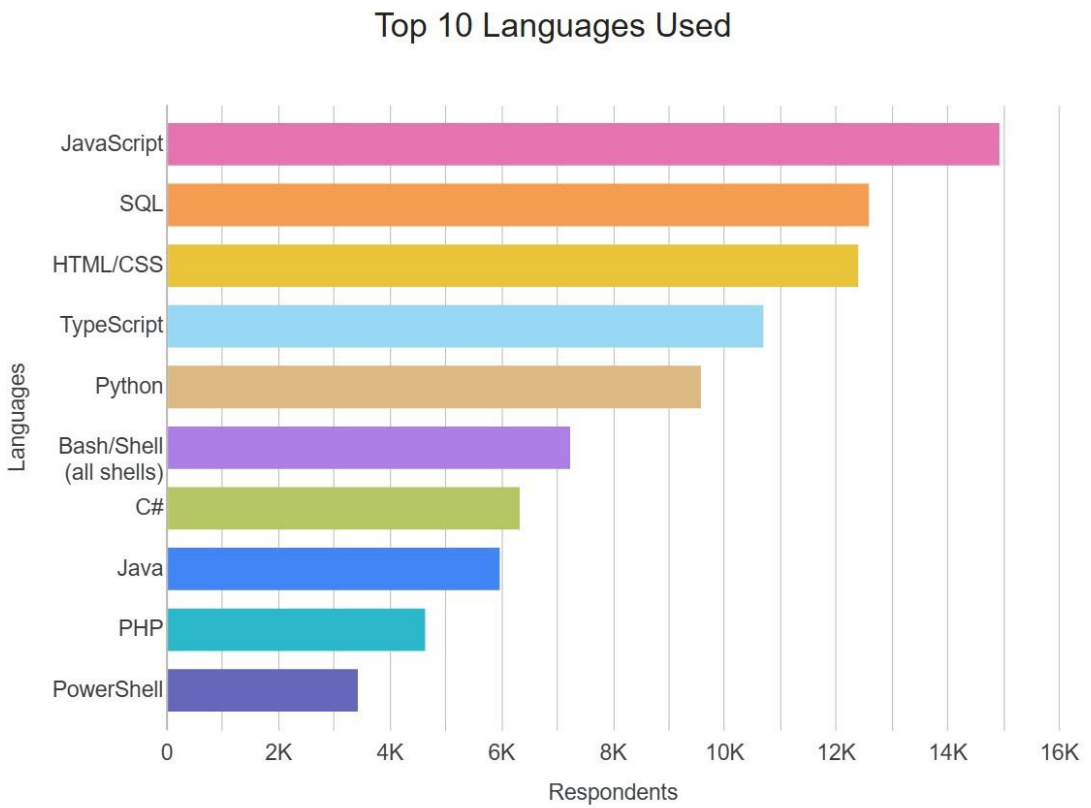
METHODOLOGY



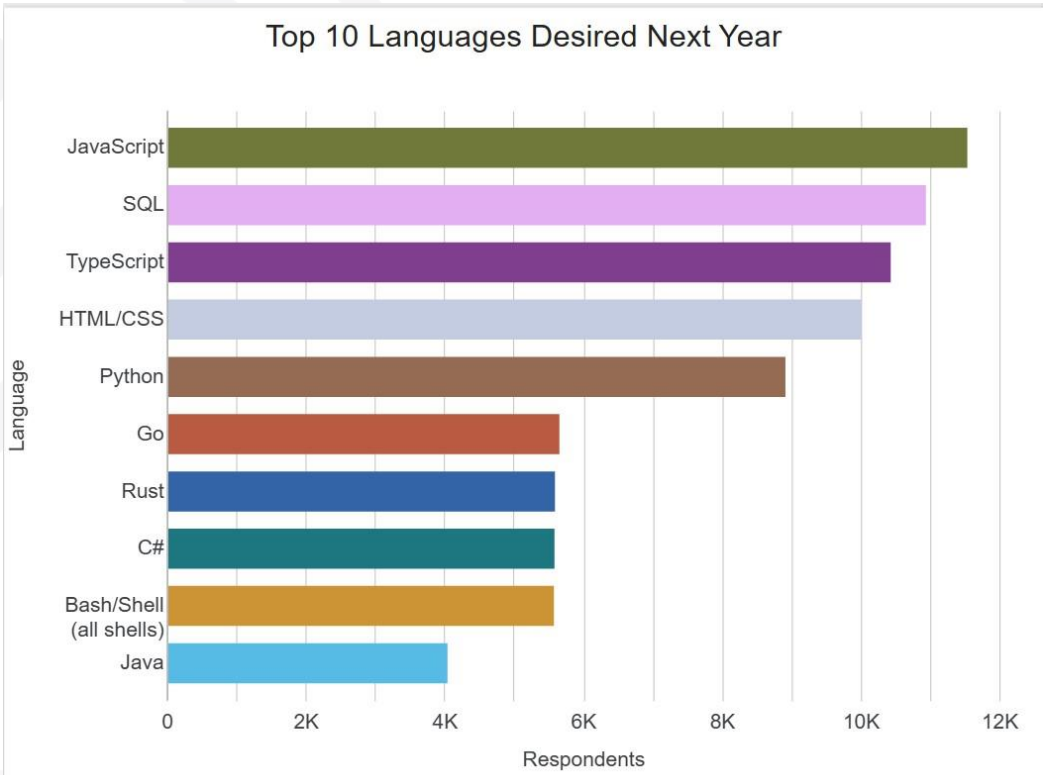
- **Data Source:** Global developer survey dataset (CSV/Excel responses)
- **Collection:** Anonymous self-reported responses on tech usage, preferences & demographics
- **Wrangling:** Removed blanks/duplicates, normalized headers, split multi-choice fields, kept top-10 per category
- **Tool:** Built interactive dashboards in **Google Looker Studio**
- **Layout: 2×2 panel** for clear theme-based visuals
- **Visuals:** Stacked bars/columns, pie, map, tree map, word cloud, scatter bubble, line-bar
- **Validation:** Cross-checked cleaned data with raw files for consistency

PROGRAMMING LANGUAGE TRENDS

Current Year



Next Year



PROGRAMMING LANGUAGE TRENDS - FINDINGS & IMPLICATIONS

Findings

Stable leaders: JavaScript (#1) and SQL (#2) hold the top spots in both current and desired lists → universal front-end + data backbone.

TypeScript momentum: climbs from #4 → #3 (desired), edging past HTML/CSS → stronger push toward typed JS in production apps.

Python steady: holds #5 in both lists → sustained demand across data, scripting, and backend.

New entrants (future): Go (#6) and Rust (#7) appear only in "Desired Next Year" → rising interest in high-performance, cloud-native, systems work.

Deprioritized: PHP (#9 current) and PowerShell (#10 current) drop out of the desired top 10.

Implications

Upskilling focus: Prioritize JavaScript + TypeScript, SQL, and Python; add Go/Rust for future-proofing in performance and cloud services.

Hiring strategy: Expect tighter markets for TypeScript, Go, Rust; start pipeline building and internal training now.

Curriculum & training: Shift hours from PHP/PowerShell toward TypeScript, modern JS tooling, and Go/Rust fundamentals; keep SQL core.

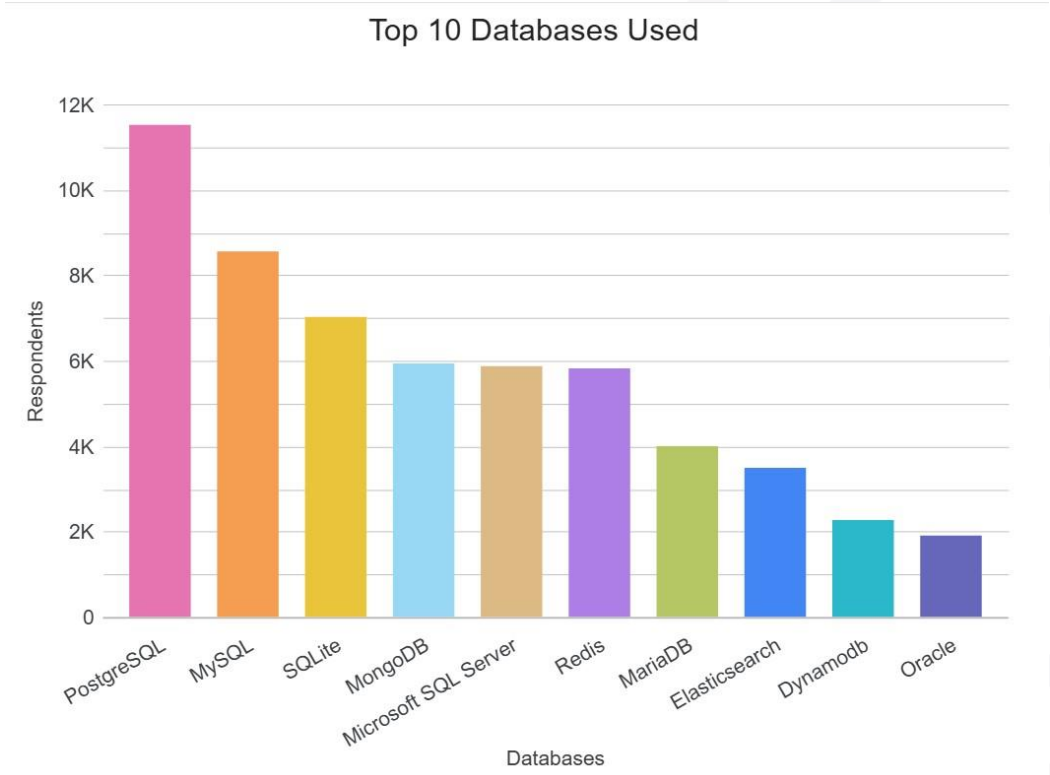
Stack modernization: Invest in typed front-ends (TS) and cloud-native backends (Go/Rust) while maintaining support for existing Java/C# estates.

Team capability mix: Keep Bash/Shell for ops hygiene, but emphasize Python/Go for automation and platform tooling.

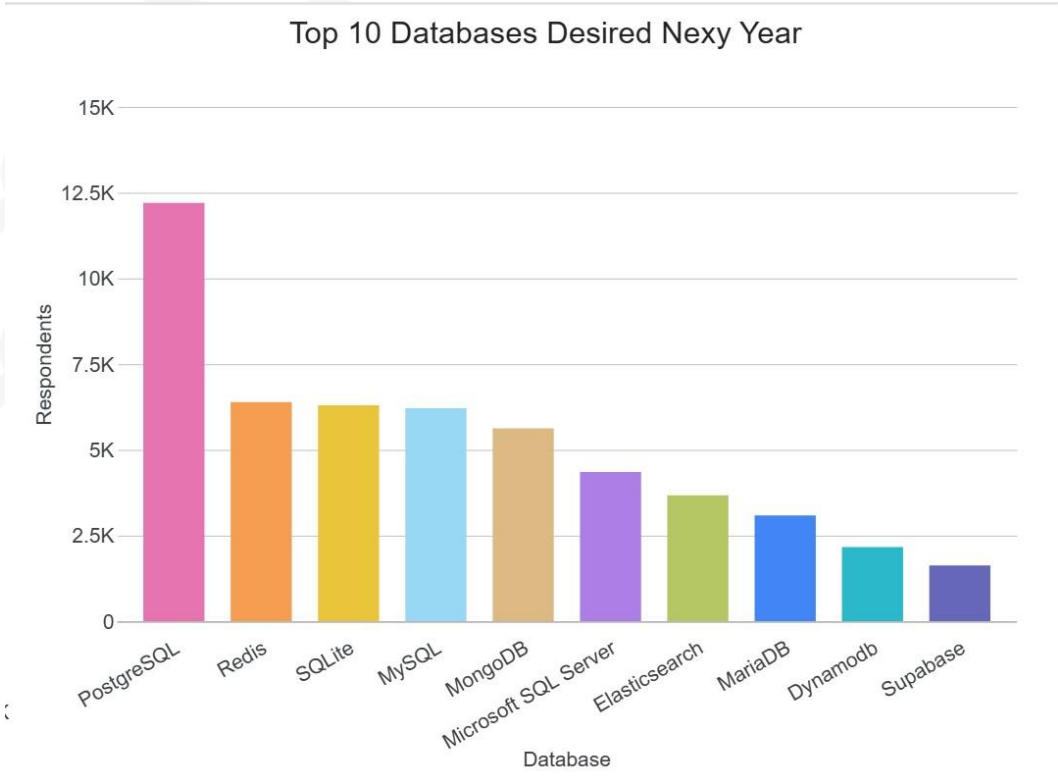
Roadmap signal: The "desired" list suggests the next wave of projects will favor typed JS + performant services—align tech choices and L&D budgets accordingly.

DATABASE TRENDS

Current Year



Next Year



DATABASE TRENDS - FINDINGS & IMPLICATIONS

Findings

PostgreSQL dominance: holds #1 today and next year → trusted relational core for modern stacks.

Redis surge: leaps from #6 → #2 desired → clear move toward low-latency caching & speed-first architectures.

SQLite consistency: remains #3 → lightweight, embedded storage stays essential for edge, mobile, and prototyping.

Managed Postgres rises: Supabase debuts at #10 desired → demand for developer-friendly, hosted Postgres services.

Search & analytics layer: steady interest in Elasticsearch → teams value integrated search/observability next to core data.

Legacy fade: Oracle drops out → cost & flexibility push workloads toward open-source, cloud-ready DBs.

Implications

Upskilling: Prioritize PostgreSQL, Redis, SQLite, and Supabase for future-ready roles.

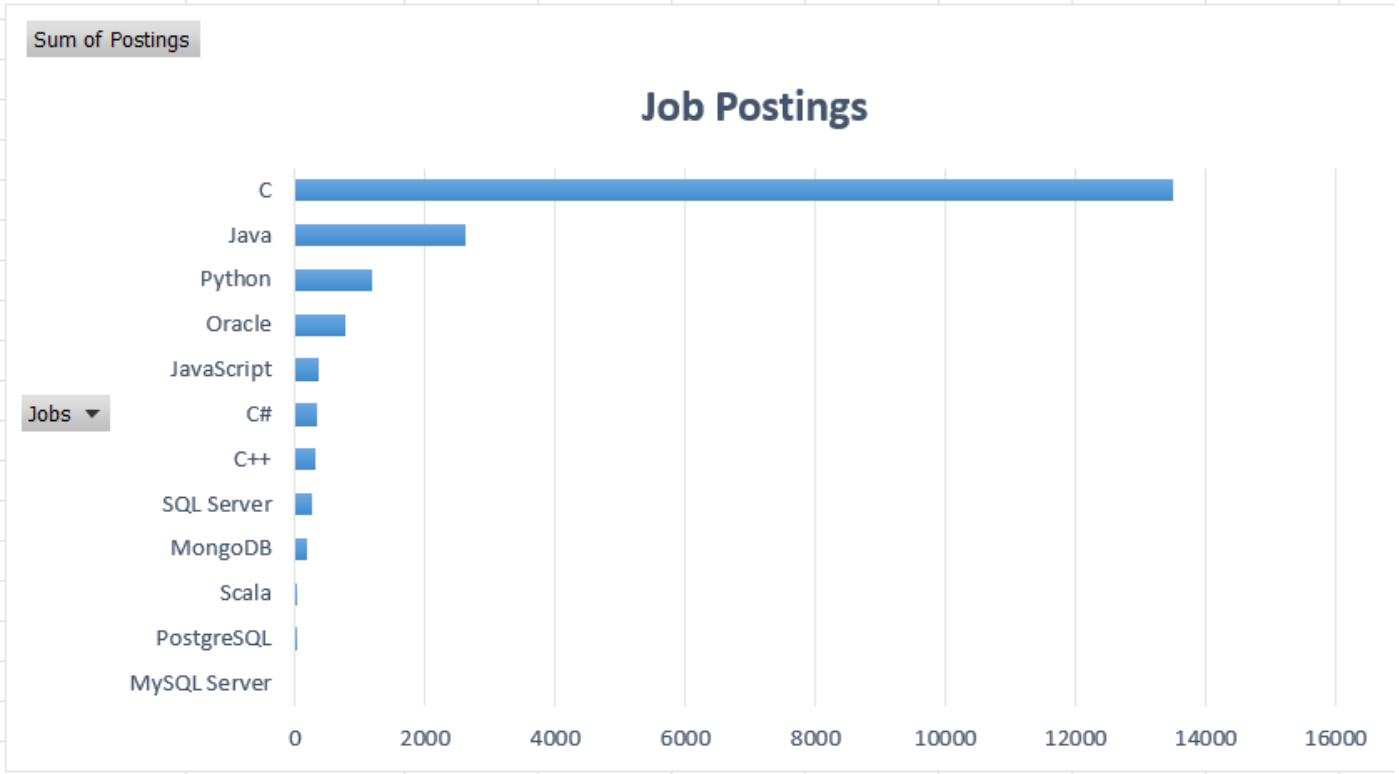
Hiring focus: Build pipelines for Postgres + caching expertise, anticipate demand in Redis/Elasticsearch.

Curriculum & training: Reduce Oracle/legacy SQL exposure; add Postgres extensions, caching patterns, managed DB ops.

Architecture shift: Expect stacks built on a relational core plus caching/search layers for scalability and speed.

JOB POSTINGS

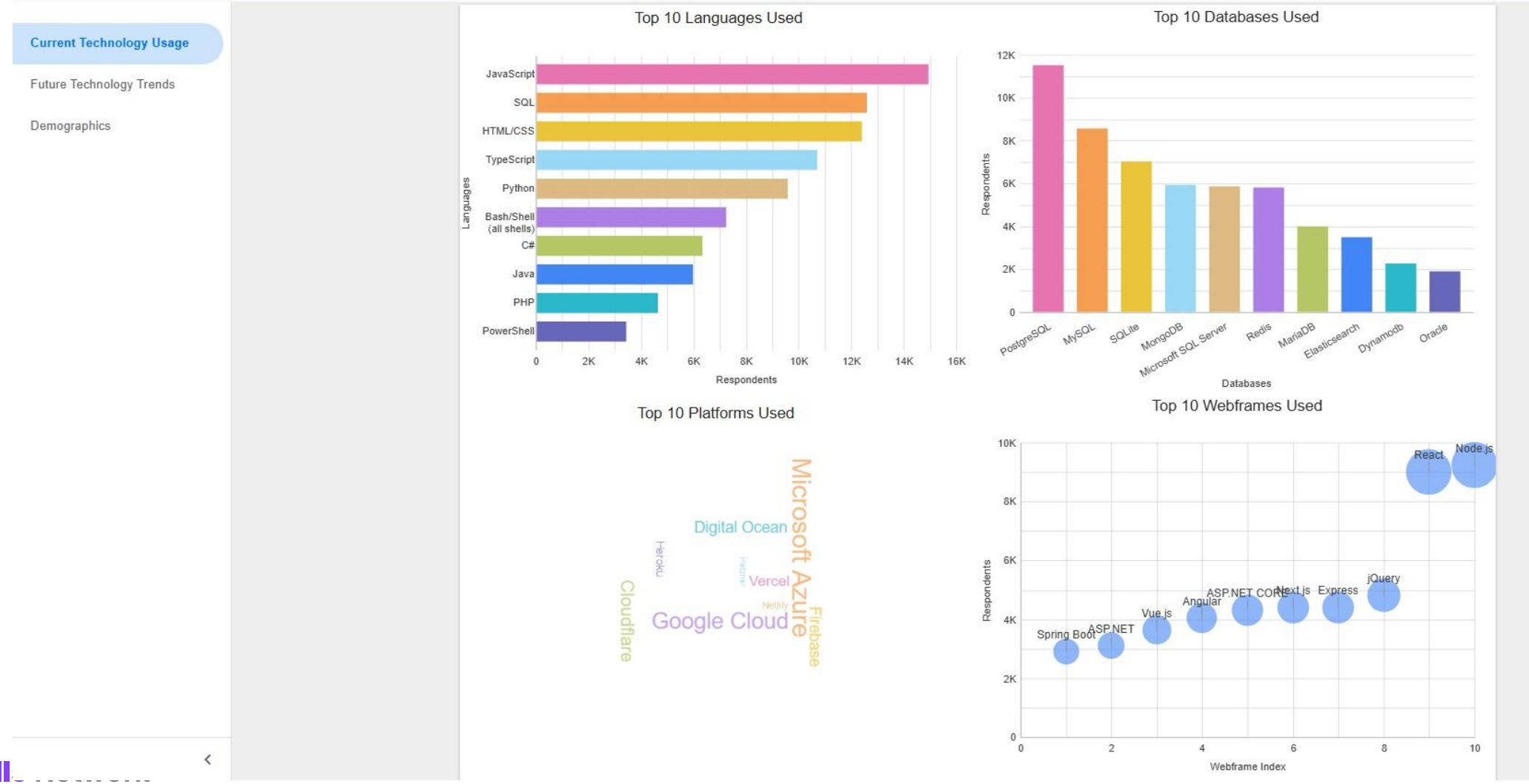
In Module 1 you have collected the job posting data using Job API in a file named “job-postings.xlsx”. Present that data using a bar chart here. Order the bar chart in the descending order of the number of job postings.



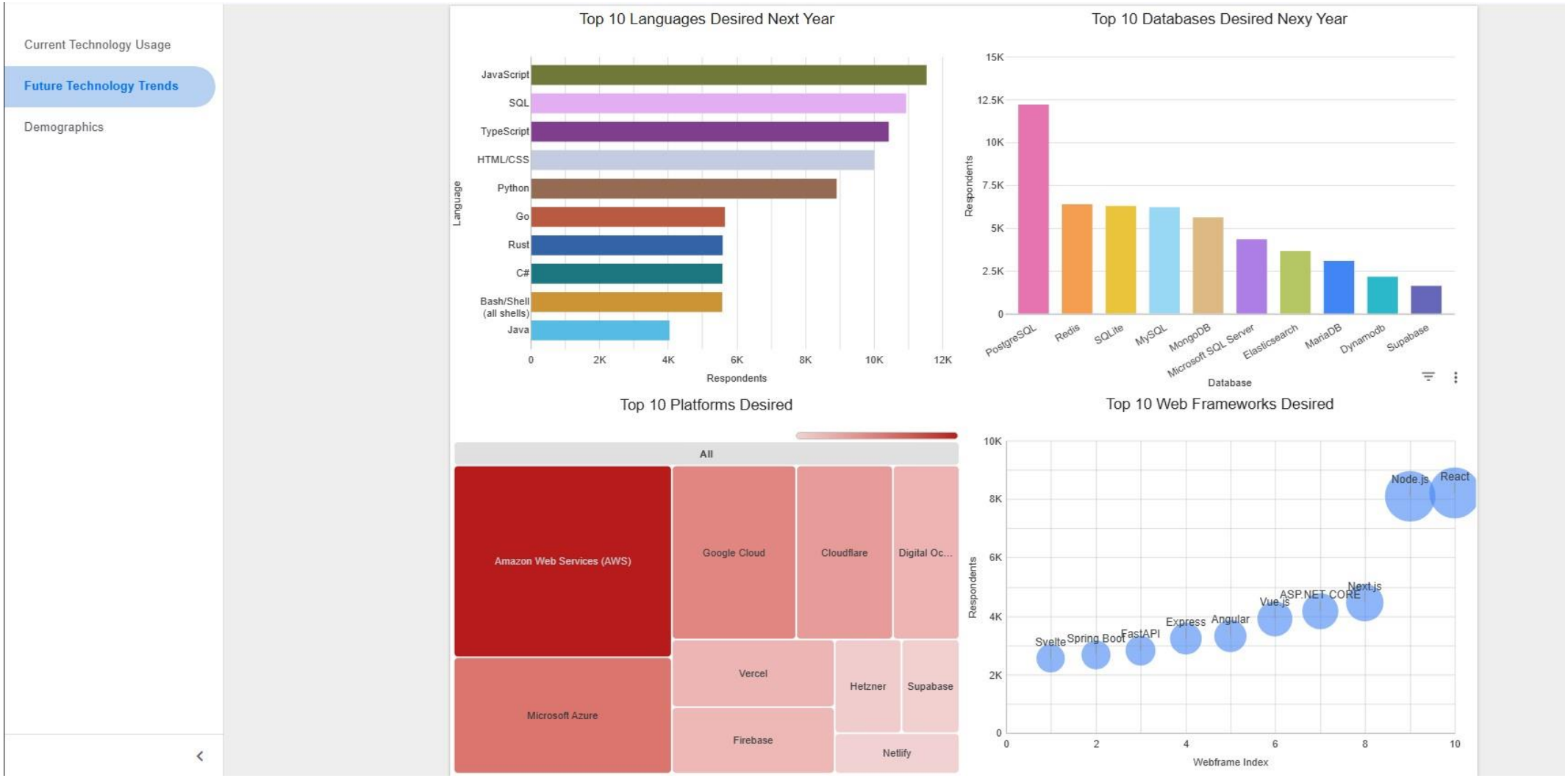
DASHBOARD

https://github.com/priya-pg/IBM-Capstone-Project-/blob/main/Survey_Results_Dashboard_by_Priyadarshini_Gupta.pdf

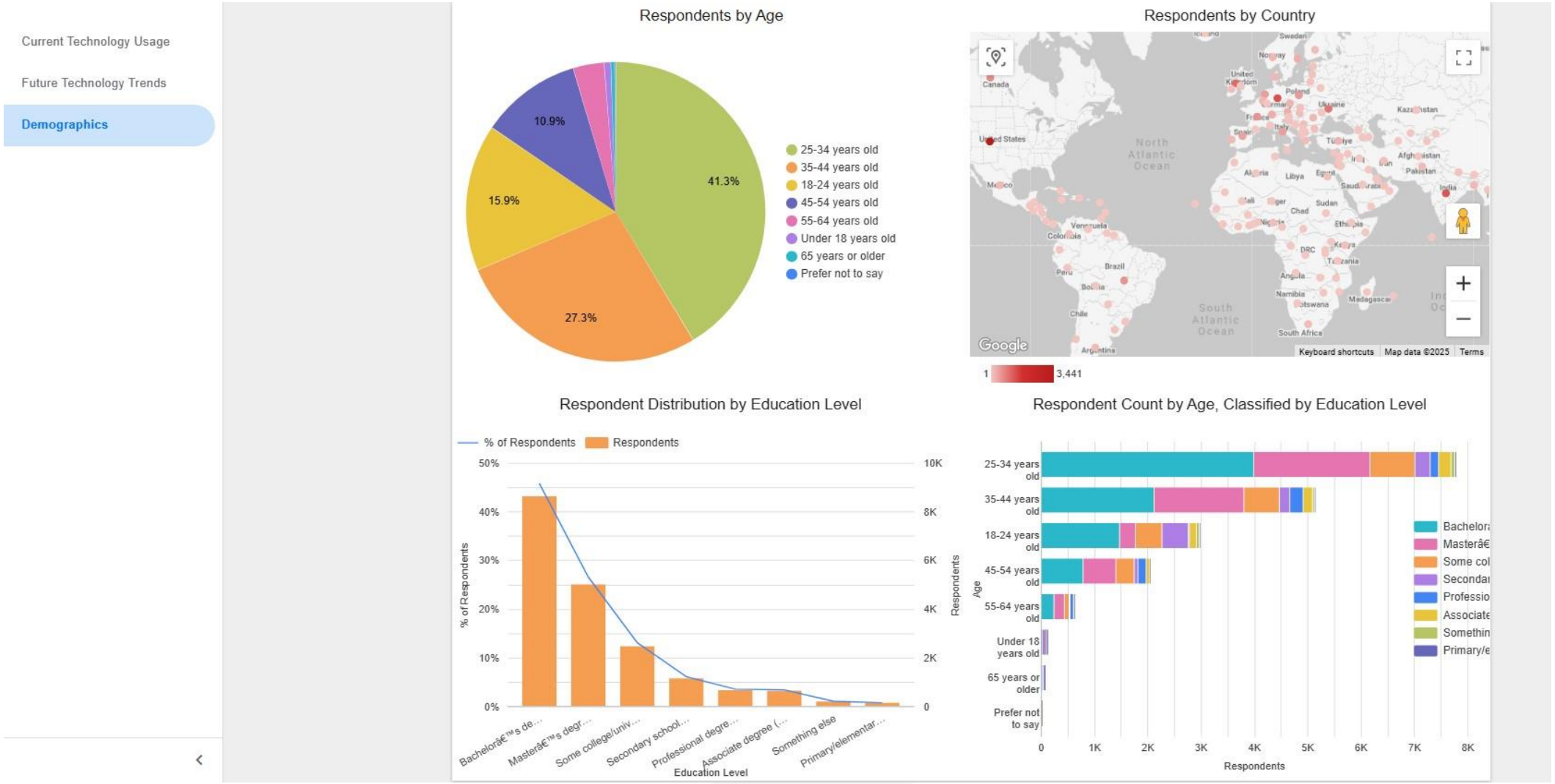
DASHBOARD TAB 1



DASHBOARD TAB 2



DASHBOARD TAB 3



OVERALL FINDINGS & IMPLICATIONS

Findings

JavaScript, SQL, React, PostgreSQL dominate current usage.

Rising interest in Go, Rust, Redis, Svelte, FastAPI.

Cloud platforms (AWS, Google Cloud, Azure) are key focus areas.

Respondents mostly aged 25–44, highly educated, globally spread.

Implications

Balance established tools with adoption of emerging tech.

Upskilling in modern languages & frameworks is critical.

Cloud demand is accelerating—AWS leads future interest.

Strong global talent pipeline ensures adaptability.

CONCLUSION

- Core technologies remain strong but next wave of adoption is forming.
- Cloud-first, modern, lightweight tools will drive future growth.
- Companies must invest in both stability and innovation to stay competitive.

APPENDIX

- To easily build the dashboard from a raw messy and very long data, I created 12 separated tables for each visual for easy readability and visual making.
- Examples:

Age	EdLevel	Count
18-24 years old	Associate degree (A.A., A.S., etc.	132
18-24 years old	Bachelorâ€™s degree (B.A., B.S.,	1468
18-24 years old	Masterâ€™s degree (M.A., M.S., M	299
18-24 years old	Primary/elementary school	25
18-24 years old	Professional degree (JD, MD, Ph.D	20
18-24 years old	Secondary school (e.g. American	495
18-24 years old	Some college/university study wit	496
18-24 years old	Something else	53
25-34 years old	Associate degree (A.A., A.S., etc.	231
25-34 years old	Bachelorâ€™s degree (B.A., B.S.,	3985
25-34 years old	Masterâ€™s degree (M.A., M.S., M	2178
25-34 years old	Primary/elementary school	30
25-34 years old	Professional degree (JD, MD, Ph.D	157
25-34 years old	Secondary school (e.g. American	288
25-34 years old	Some college/university study wit	843
25-34 years old	Something else	76
35-44 years old	Associate degree (A.A., A.S., etc.	176

Database	Count
PostgreSQL	11514
MySQL	8556
SQLite	7021
MongoDB	5930
Microsoft SQL Server	5870
Redis	5814
MariaDB	3994
Elasticsearch	3491
Dynamodb	2268
Oracle	1907

Language	Count
JavaScript	14943
SQL	12602
HTML/CSS	12410
TypeScript	10709
Python	9590
Bash/Shell (all shells)	7244
C#	6340
Java	5982
PHP	4644
PowerShell	3438

P.S. These three are only example purposes, I have made 9 others like this for each of the corresponding visuals in the dashboard

POPULAR LANGUAGES

In Module 1 you have collected the job postings data using web scraping in a file named “popular-languages.csv”. Present that data using a bar chart here. Order the bar chart in the descending order of salary.