# Developer Technology Preferences and Usage Patterns

**A Comprehensive Data Analysis Presentation**

***Understanding Current Usage and Future Trends in Software Development***

**Presented by:** Serge Caraus
**Date:** June 24, 2025
**Analysis based on:** Stack overflow Survey Data

# Presentation Outline

- **Key Sections Overview**

1. **Executive Summary** - Core findings at a glance
2. **Introduction** - Purpose, audience, and project value
3. **Methodology** - Data sources and analysis approach
4. **Programming Languages Trends** (Slides 6-7)
5. **Database Technology Trends** (Slides 8-9)
6. **Platform & Framework Analysis** (Slide 10)
7. **Interactive Dashboards** (Slides 11-13)
8. **Dashboard Insights** - Key takeaways from visualizations
9. **Overall Findings & Implications** - Strategic insights
10. **Conclusions** - Final recommendations

# Executive Summary

- **Key Findings Snapshot**
- **Technology Leaders:**
- **JavaScript** dominates both current usage (#1) and future preferences (#1)
- **PostgreSQL** leads database adoption in both categories
- **AWS** maintains cloud platform supremacy

- **Emerging Trends:**
- **Rust** enters desired languages list (not in current top 10)
- **GO** jumps from #10 to #6 in desired languages
- **Supabase** appears in desired platforms, signaling modern dev tool adoption

- **Survey Demographics:**
- **Primary Age Group:** 25-34 years (career-peak developers)
- **Global Reach:** USA, India, Germany, UK leading participation
- **Education:** Bachelor's and Master's degree holders predominant

# Introduction

**•Primary Objectives:**

•Shows which technologies developers are using right now

•Finds new trends and what technologies developers want to learn next

•Gives practical advice for learning new skills and making technology choices

**•Target Audience:**

•Software developers planning their career path

•Tech managers making decisions about what tools to use

•Schools and universities creating programming courses

•People who hire developers and need to know what skills matter

**Value Propositions:**

•Real data showing which technologies are trending up or down

•Comparison between what developers use now versus what they want to use

•Worldwide view of what developers prefer across different countries

•Smart advice for choosing which technologies to invest time and money in

# Methodology

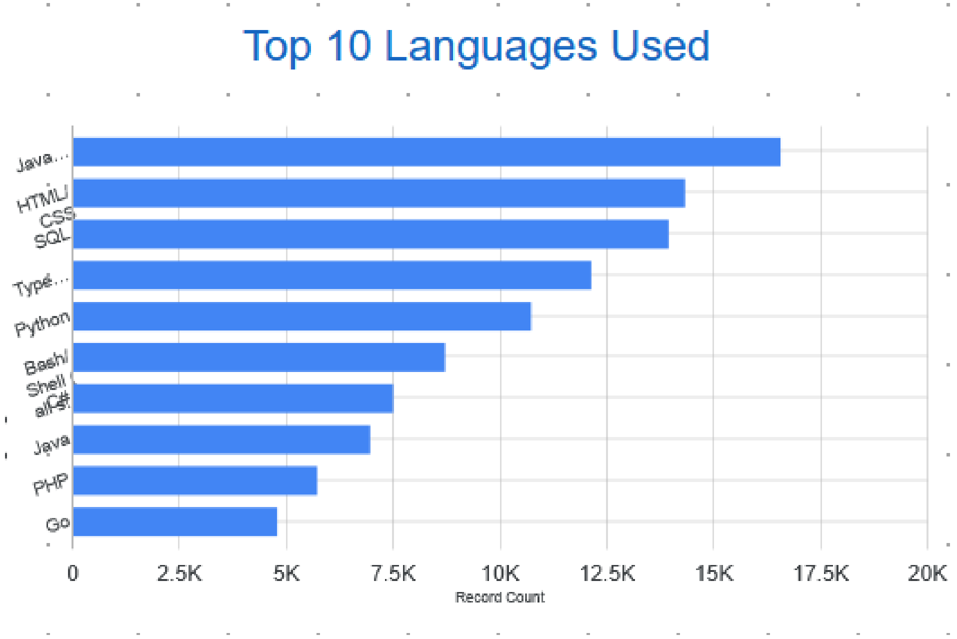
- **Data Sources:**
- Global developer survey covering 4 technology categories
- Rankings for top 10 technologies in each category
- Demographic data (age, location, education)
- Both current usage and future preference metrics
- **Collection Methods:**
- Structured survey methodology
- Global participant recruitment
- Multi-category technology assessment
- Demographic segmentation analysis

**Key Data Wrangling Steps:**

•**Removing NaN values:** Eliminated null/missing values from survey responses

•**Transforming string comma-separated values:** Converted comma-separated technology lists into unique values in individual columns.

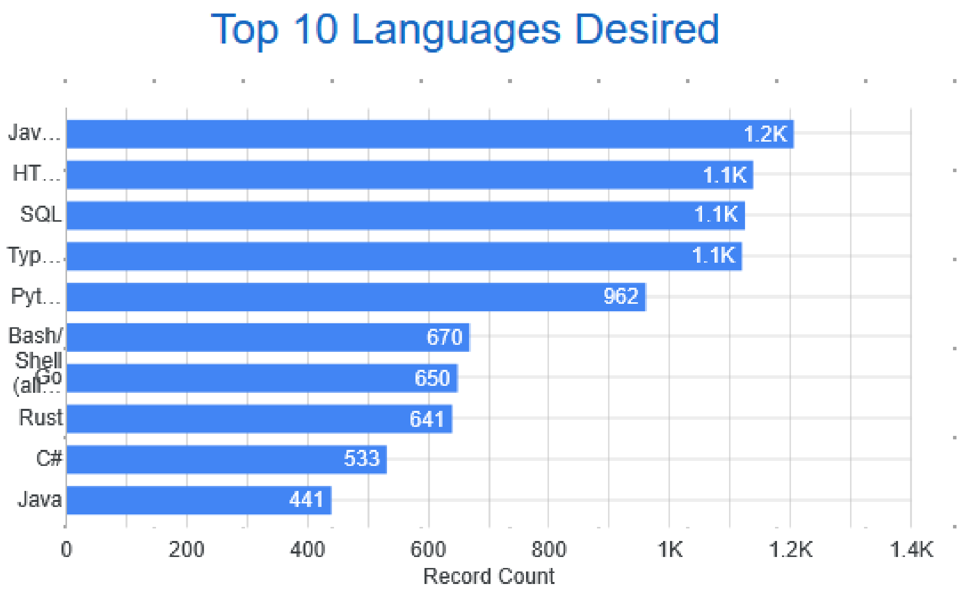
# Programming Languages - Current Usage



**Key Observations:**

- **Web Technologies Dominate:** JavaScript, HTML/CSS, TypeScript represent 3 of top 4
- **Data-Centric Languages Strong:** SQL and Python in top 5
- **Enterprise Languages Present:** C# and Java maintaining relevance

# Programming Languages - Future Trends



**Trend Analysis:**

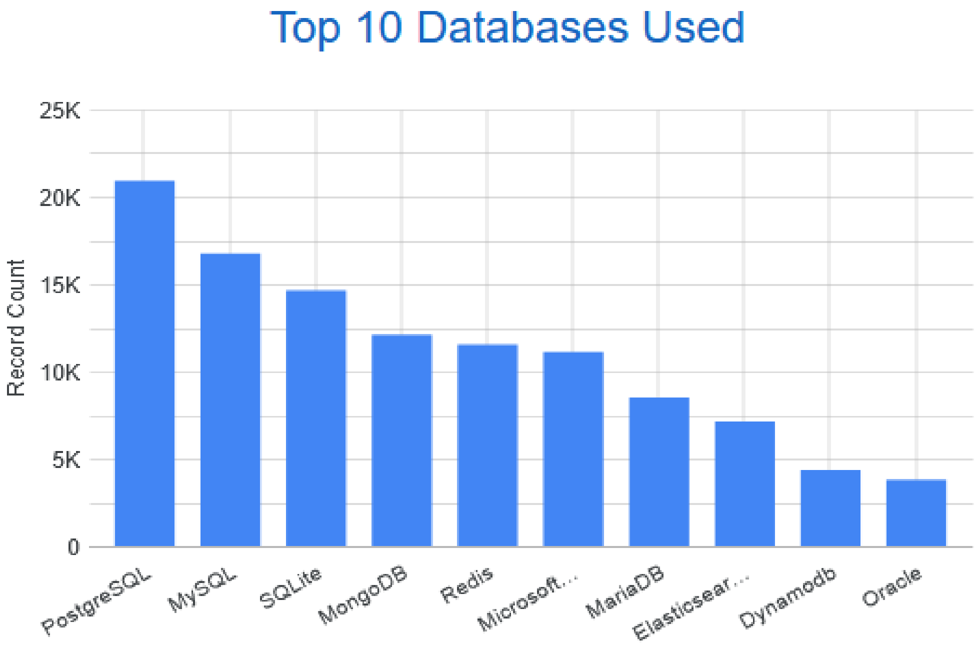
•**Rising Stars:** GO (+4 positions), Rust (new entry) indicate systems programming growth

•**Strengthening:** SQL and TypeScript gaining developer interest

•**Stabilizing:** Traditional enterprise languages (C#, Java) seeing slight decline

•**Innovation Signal:** Rust's emergence suggests safety-focused programming demand

# Database Technologies - Current Usage



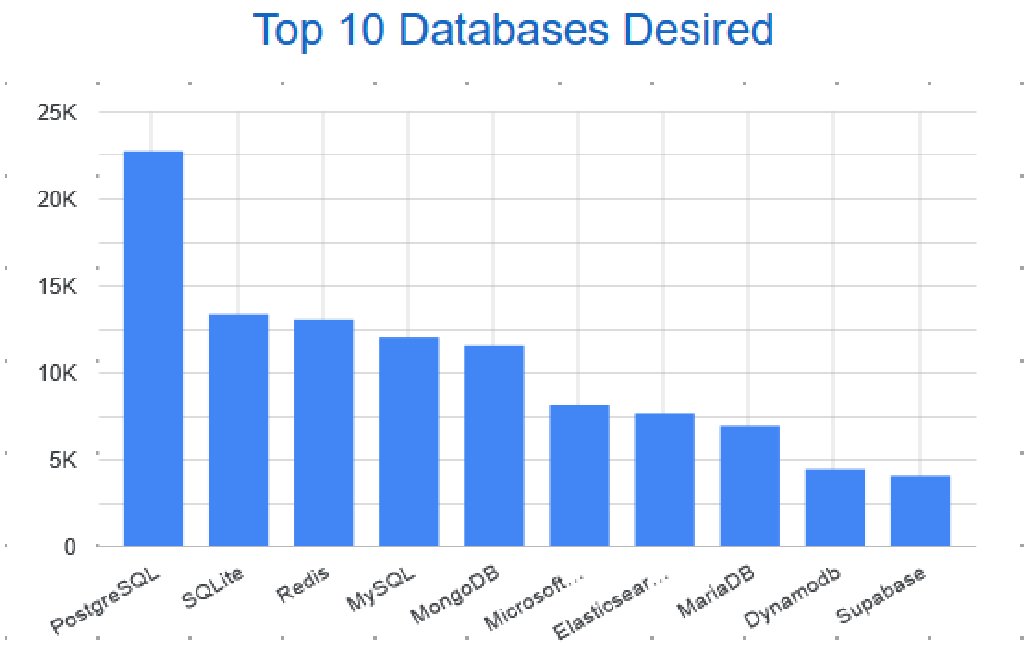
**Current Landscape Insights:**

•**Relational Dominance:** Top 3 are all relational databases

•**NoSQL Presence:** MongoDB, Redis, DynamoDB showing strong adoption

•**Cloud Integration:** DynamoDB reflects cloud-native database trends

# Database Technologies - Future Demand



**Future Trends Analysis:**

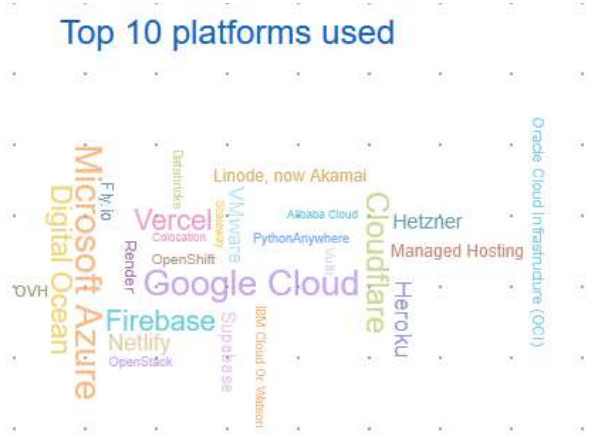
• **PostgreSQL Supremacy:** Maintains top position, indicating developer satisfaction

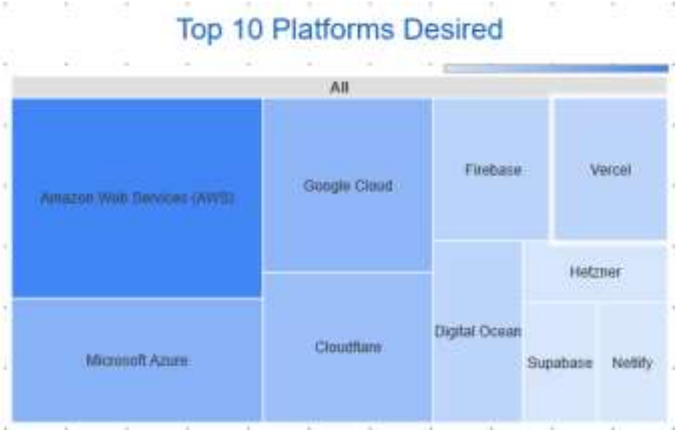
•**Performance Focus:** Redis climbing (+2) suggests speed/caching importance

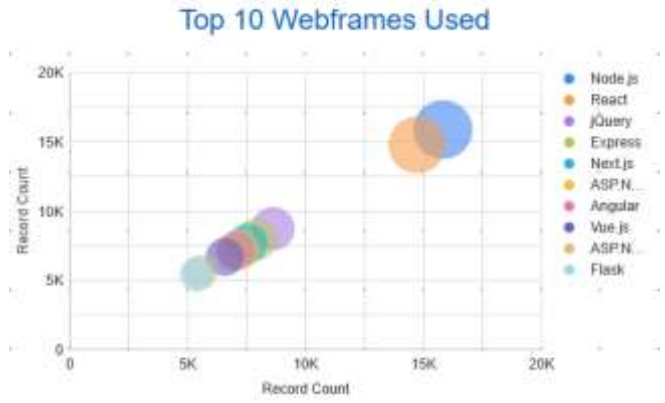
•**Modern Platforms:** Supabase entry signals developer-friendly database services demand

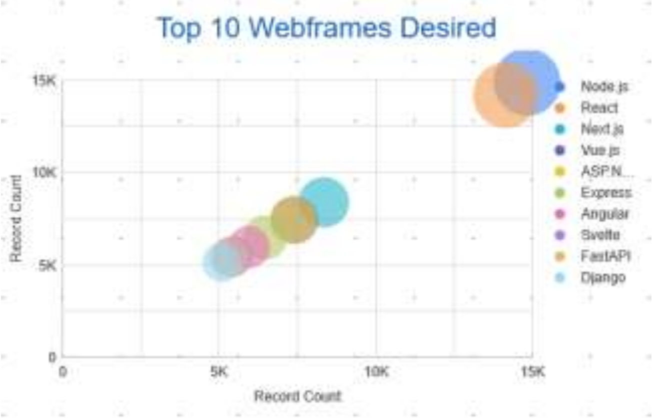
•**Traditional Shift:** MySQL and MongoDB slight decline may indicate feature/usability gaps

# Platforms & Frameworks Overview
# Cloud Platforms & Web Frameworks Landscape







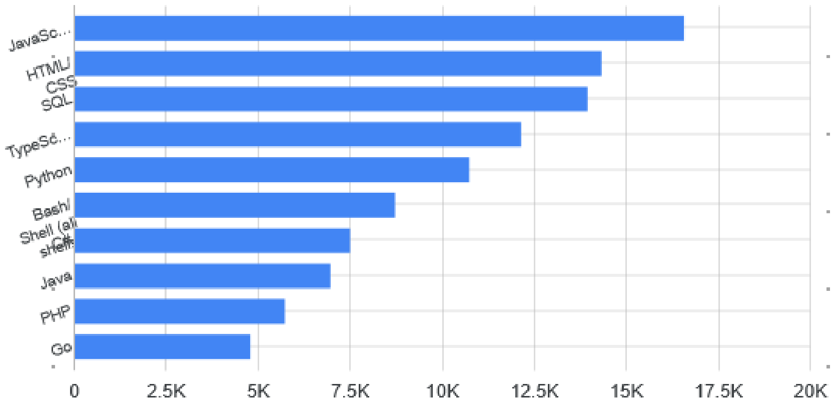


**Platform & Framework Trend Analysis:**

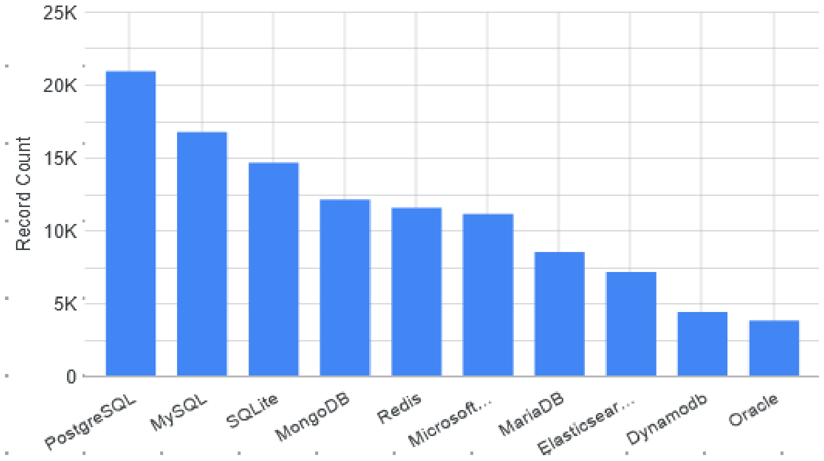
1. Big Cloud Companies Stay on Top: AWS, Azure, Google Cloud keep their leading positions
2. New Options Are Appearing: Hetzner (popular in Europe), Supabase (easy backend service) gaining attention
3. Modern Website Tools Growing: Next.js (+2), Vue.js (+4) show developers want better frontend experiences
4. More Choices Available: New frameworks (Svelte, FastAPI, Django) give developers different options to pick from
5. Python Getting Popular for Websites: FastAPI and Django success shows Python becoming stronger for web development

# Current Technology Usage

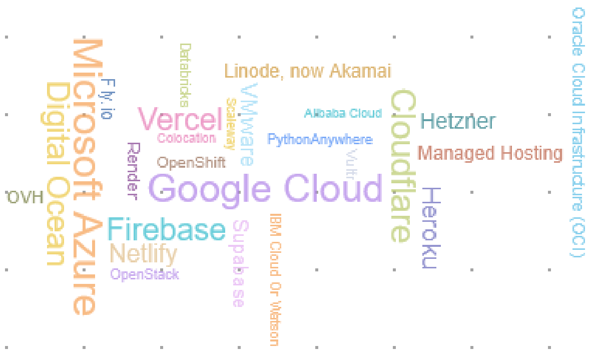
Top 10 Languages Used



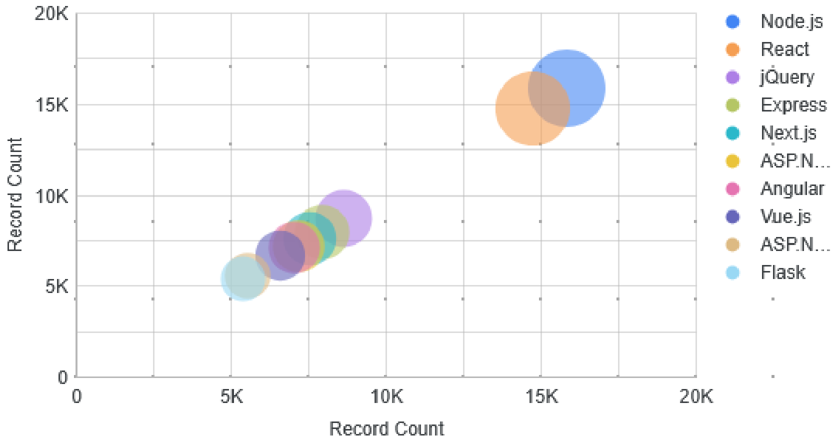
Top 10 Databases Used



Top 10 platforms used

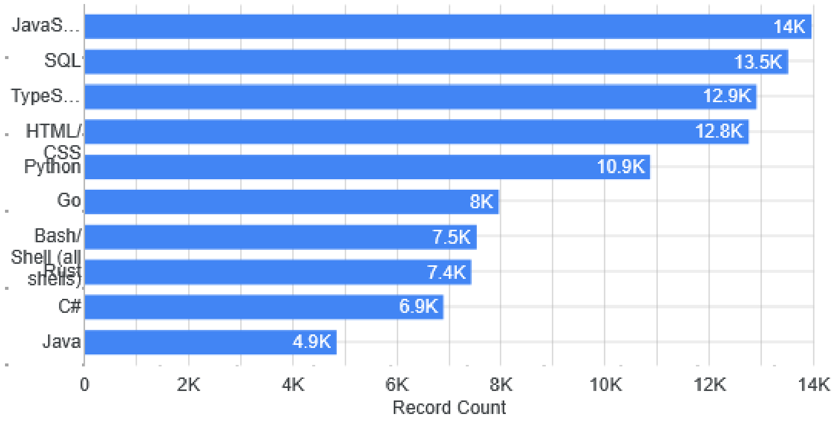


Top 10 Webframes Used

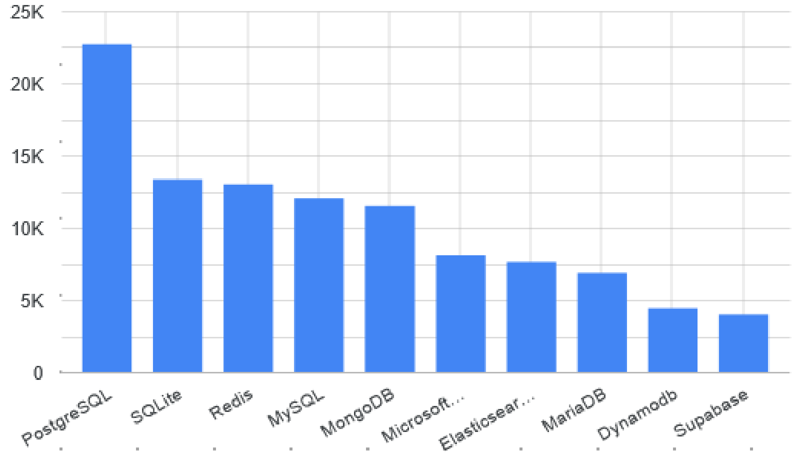


# Future Technology Trends

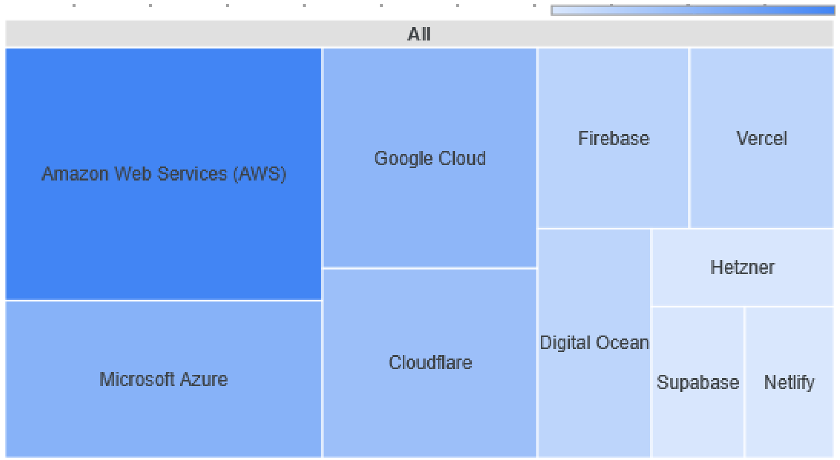
## Top 10 Languages Desired



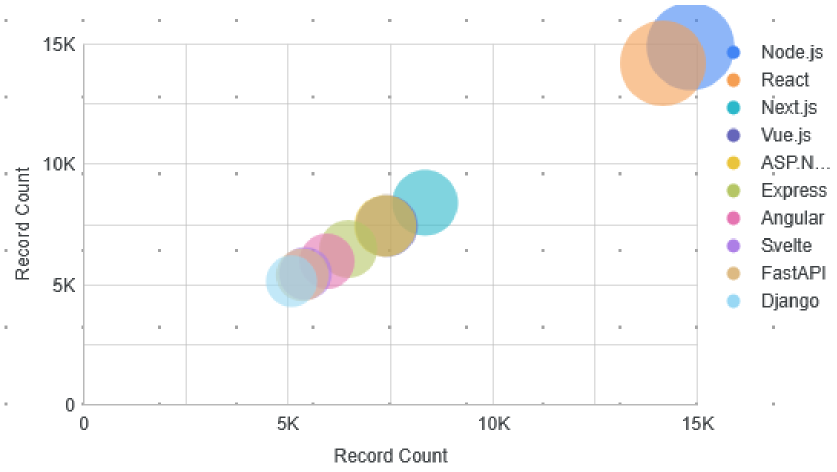
## Top 10 Databases Desired



## Top 10 Platforms Desired

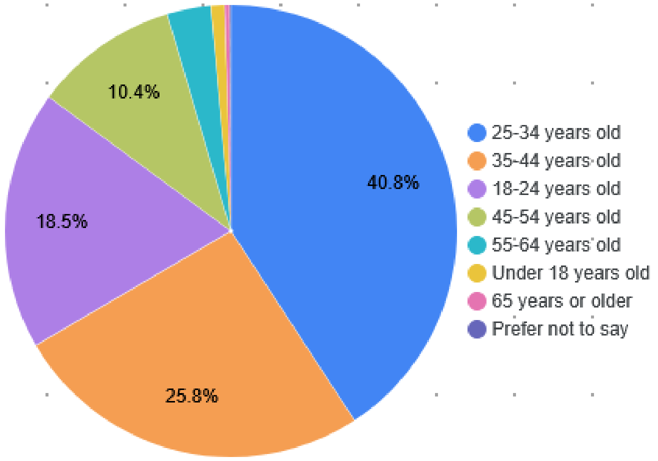


## Top 10 Webframes Desired

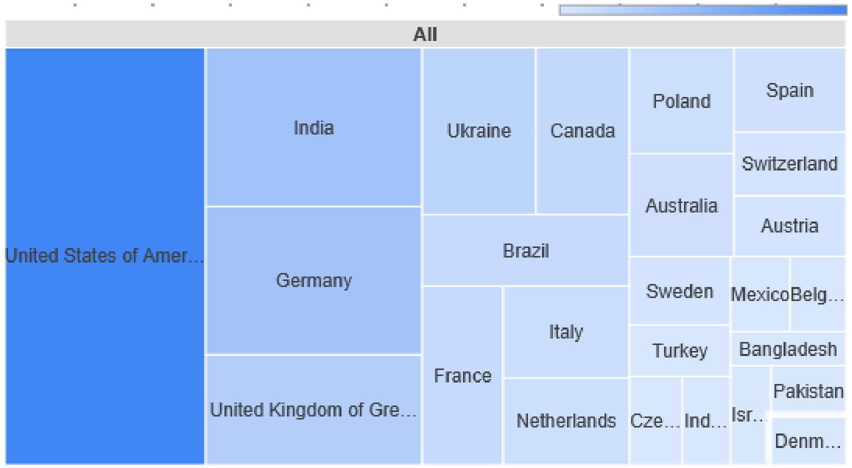


# Demographics

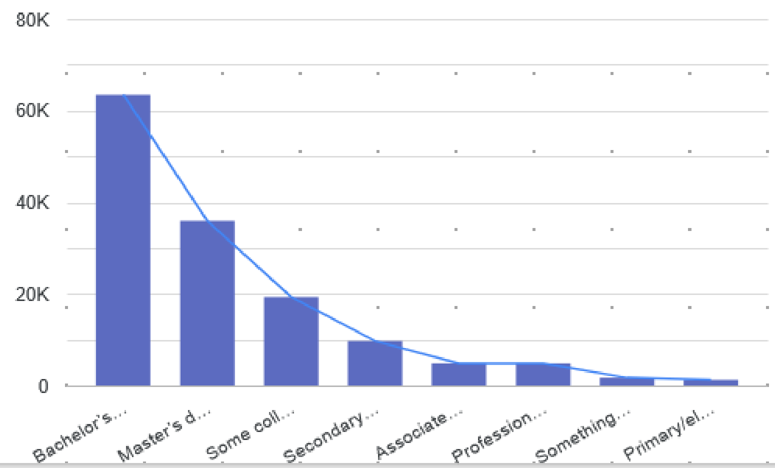
## Respondents by Age



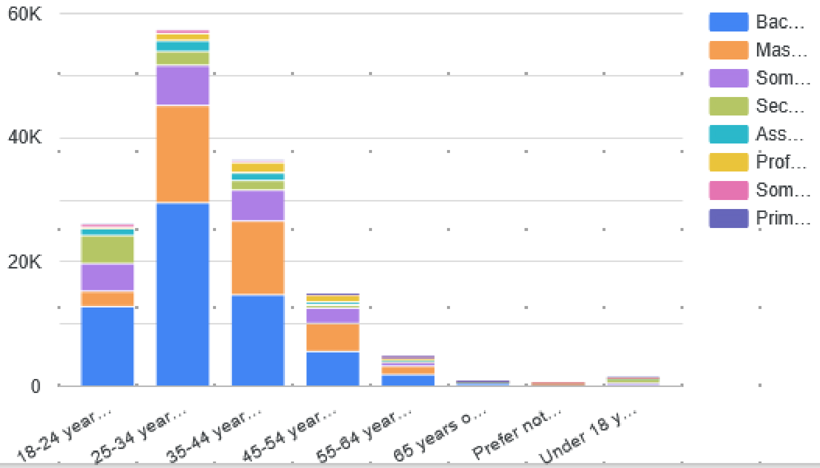
## Respondent Count by Country



## Respondent Distribution by Education Level



## Respondent Count by Age, Classified by Education Level



# Key Insights Derived from Visualizations

- **JavaScript Stays Popular:** Keeps 33-35% of developer preference, showing it's not going anywhere

- **Fast Tools Are Winning:** Redis and GO moving up because developers want speed and efficiency

- **Bug-Prevention Focus:** TypeScript and Rust growing because preventing errors is now essential

- **What's Happening in the Market:**
- **Big Names Stay Strong:** AWS, PostgreSQL, Node.js keep their leadership positions
- **New Ideas Start Small:** Fresh technologies (Rust, Supabase) show up in "want to try" lists first
- **Everything Goes to Cloud:** Platform choices show everyone's building apps that work across multiple servers

- **Who's Making These Decisions:**
- **Experienced Developers Lead:** 25-34 age group (42%) are the main decision makers
- **Worldwide Input:** Responses from multiple countries make these trends globally relevant
- **Smart Choices:** High education levels (63% Bachelor's+) mean these are well-informed decisions

- **How to Predict the Future:**
- **Wish vs Reality Shows Trends:** Gaps between current/desired usage predict what's coming next
- **New "Want to Try" Lists:** Technologies appearing in desired categories signal future growth
- **Consistent Winners Are Safe:** Top performers staying steady suggest reliable long-term choices

# Overall Findings and Implications

- **MAJOR FINDINGS:**
- **Technology Stability Leaders:**
- **JavaScript** ecosystem dominance continues (35% usage, sustained #1 ranking)
- **PostgreSQL** database supremacy confirmed (30% future preference)
- **AWS** cloud platform leadership unchallenged (40% current usage)
- **Emerging Growth Signals:**
- **Rust:** Developers want safer programming languages. Rust prevents common bugs and security problems that crash programs, making software more reliable and secure than older languages like C++.
- **Go:** Moved up 4 spots because it's perfect for cloud apps. Go runs the tools that manage containers and websites, starts fast, uses little memory, and handles many users simultaneously.
- **Supabase:** New easy-to-use backend service. Instead of building databases and user login systems from scratch, developers get everything ready-made with familiar SQL database plus modern features like real-time updates.

**STRATEGIC IMPLICATIONS:**

**For Individual Developers:**
**Core Skills:** Keep learning JavaScript, SQL databases, and cloud platforms like AWS. These are the foundation skills that most jobs require.
**Growth Areas:** Start learning Go and Rust now. These languages are becoming more popular and will create better job opportunities in the future.
**Safe Bets:** PostgreSQL database and AWS cloud skills will always be in demand. Companies rely on these technologies and need people who know them.

**For Organizations:**
**Technology Stack:** Combine JavaScript for apps, PostgreSQL for data, and AWS for hosting. This trio works well together and has proven successful for many companies.
**Innovation Investment:** Test Rust for building fast, secure system software. Try Go for creating small services that can scale up easily when you get more users.
**Platform Strategy:** Use major cloud providers for reliability, but also experiment with newer, simpler alternatives that might work better for your specific needs.

**For Industry:**
**Market Maturation:** Programming tools that prevent bugs before they happen are becoming mandatory. Companies expect this level of quality in professional software development now.
**Performance Focus:** Technology decisions are increasingly based on speed and resource efficiency rather than just having lots of features or being trendy.
**Developer Experience:** Success of tools like Supabase shows that making software easy and enjoyable for developers to use is now a critical business advantage.

# Conclusions

- **1. The Big Technologies Stay on Top:** The main programming languages and tools (JavaScript, SQL databases, major cloud services) have become the standard and aren't being replaced anytime soon.
- **2. New Tech Starts as "Wishlist Items:** New technologies first become popular in "wish lists" and surveys before developers actually start using them in real projects at work.
- **3. Simple is Better Than Complex:** Developers want fewer tools that do more things well, instead of having to juggle many different specialized tools that don't work together.
- **4. Fast, Safe, and Easy Wins:** Future technology choices will be based on three things: how fast they run, how safe they are from bugs, and how easy they make the developer's job.