Executive Summary

This capstone project demonstrates end to end data analytics using the CustomerLoyaltyProgram and survey_data datasets. Data was collected through web scraping and APIs, then cleaned and prepared for analysis. Using Python libraries like Pandas, NumPy, Matplotlib, and Seaborn, exploratory and statistical analyses identified key trends and correlations in customer satisfaction and engagement. Insights were visualized in interactive dashboards using IBM Cognos Analytics and Google Looker Studio. This work showcases practical expertise in data wrangling, visualization, and dashboard design.

Introduction

Organizations increasingly rely on analytics to understand customer behavior and drive decisions. This project applies core data analysis techniques to uncover insights about customer engagement, satisfaction, and loyalty.

The workflow included data collection via APIs and web scraping, followed by data cleaning to handle duplicates and missing values. Exploratory Data Analysis revealed distributions, outliers, and relationships between variables. Key findings were communicated through visualizations including histograms, scatter plots, box plots, and line charts. Finally, interactive dashboards summarized insights for stakeholder storytelling.

Methodology

1) **Data Collection**

The data collection stage focused on gathering real-world information from online sources to simulate how professional data analysts acquire data. Two datasets — **CustomerLoyaltyProgram.csv** and **survey_data.csv** — served as the foundation for analysis. To strengthen practical experience, I practiced API access and web scraping techniques using Python's requests and BeautifulSoup libraries. This step demonstrated how data can be pulled dynamically from the web and transformed into structured formats for analysis

Web Scraping & Data Compilation

As part of the challenge, I automated the extraction of job listings by technology and location, combining multiple sources into a single dataset. The data was structured into a Pandas DataFrame for easy exploration and visualization.

```python
## Challenge Accepted ##
results = []

for tech in technologies:
    tech_name, T_Jobs_count = get_number_of_jobs_T(tech)
    for loc in locations:
        loc_name, L_Jobs_count = jobs_location(loc)
        results.append((tech_name, T_Jobs_count, loc_name, L_Jobs_count))

df = pd.DataFrame(results, columns=["Technology", "Tech_Jobs", "Locations", "Location Jobs"])
df.head(10)
```

57]:

	Technology	Tech_Jobs	Locations	Location Jobs
50	Python	1173	Washington DC	5316.0
155	SQL Server	250	Washington DC	5316.0
288	Java	2609	Washington DC	5316.0
43	JavaScript	355	Washington DC	5316.0
169	PostgreSQL	10	Washington DC	5316.0
253	PostgreSQL	10	Washington DC	5316.0
113	C++	305	Washington DC	5316.0
309	Scala	33	Washington DC	5316.0
162	MySQL Server	0	Washington DC	5316.0
92	MongoDB	174	Washington DC	5316.0
260	MongoDB	174	Washington DC	5316.0

2) Data Wrangling

The data wrangling stage focused on cleaning and preparing the dataset for reliable analysis. I identified and removed duplicate records, handled missing values, and normalized numerical fields to maintain consistency across all observations.

Duplicates were defined in two ways — based on all columns and then on a set of critical columns that described the respondent's demographic and work information.

```python
# Finding duplicates (complete columns)
dup_complete = df.duplicated()
print(dup_complete.sum())   # → 20 duplicates

# Defining critical columns for refined check
critical_cols = [
    "MainBranch","Employment","RemoteWork","Country","OrgSize",
    "Industry","Age","EdLevel","YearsCode","YearsCodePro",
    "WorkExp","DevType","LanguageHaveWorkedWith","CompTotal"
]

# Finding duplicates (based on critical columns)
dup_subset = df.duplicated(subset=critical_cols, keep=False)
print(dup_subset.sum())     # → 3,289 potential duplicates

# Removing duplicates
df_cleaned = df.drop_duplicates(subset=critical_cols, keep="first")

# Result summary
print("Original:", df.shape)
print("After cleaning:", df_cleaned.shape)
```

Output Summary

```yaml
Original: (65457, 114)
After cleaning: (62564, 114)
Number of duplicates removed: 2,893
```

Missing Values & Normalization

```python
# Finding missing values
df_cleaned.isnull().sum()

# Imputing numeric columns with mean
df_cleaned.fillna(df_cleaned.mean(numeric_only=True), inplace=True)

# Imputing categorical columns with mode
df_cleaned.fillna(df_cleaned.mode().iloc[0], inplace=True)

# Normalizing numerical features (Min-Max scaling)
df_cleaned["CompTotal"] = (df_cleaned["CompTotal"] - df_cleaned["CompTotal"].min()) / \
                          (df_cleaned["CompTotal"].max() - df_cleaned["CompTotal"].min())
```

After data wrangling:

- **2,893 duplicate entries** were removed.
- Missing values were imputed for both numeric and categorical fields.
- Key numerical columns were normalized to ensure comparability.

The cleaned dataset was now ready for **Exploratory Data Analysis (EDA)**

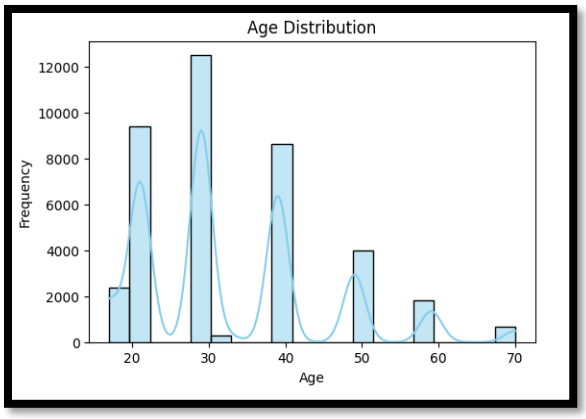
3) Exploratory Data Analysis and visualization

At this stage, the goal was to understand the data's structure, detect outliers, and identify relationships among demographic, experience, and compensation variables.
Exploratory analysis was paired with meaningful visualizations to highlight trends and patterns. The three variables selected — **Country**, **YearsCodePro**, and **CompTotal_normalized** — provided insight into respondents' demographics, experience levels, and compensation differences across regions.

Dataset shape: (39608, 233)

Summary statistics:

	ResponseId	YearsCodePro	CompTotal	WorkExp	JobSatPoints_1
count	39608.000000	39608.000000	39608.000000	14293.00000	14152.000000
mean	32683.969981	9.507120	108388.542820	12.41146	18.068224
std	18777.871385	8.242757	13015.281521	9.54046	25.823815
min	1.000000	0.000000	70000.000000	0.00000	0.000000
25%	16333.750000	5.000000	110000.000000	5.00000	0.000000
50%	33069.500000	7.000000	110000.000000	10.00000	10.000000
75%	48838.000000	12.000000	110000.000000	18.00000	20.000000
max	65437.000000	51.000000	150000.000000	50.00000	100.000000



4) Building A Dashboard

Using IBM Cognos Analytics in this project provided a hands-on experience in transforming cleaned data into interactive and meaningful visual insights. After uploading the *survey_data_updated.csv* dataset, I designed a three-tab dashboard covering Current Technology Usage, Future Technology Trends, and Demographics. Each section was organized into a 2×2 layout of visual panels, showcasing key metrics such as the most used and desired **languages, databases, platforms, and web frameworks**, along with respondent **age, country, and education distribution**. The platform's drag-and-drop interface made it easy to build visualizations like bar charts, pie charts, maps, and hierarchy bubbles, while features such as Show Value Labels and interactive filters enhanced clarity and exploration. Overall, using Cognos allowed me to apply professional BI techniques—connecting data preparation, analysis, and visualization into a cohesive and data-driven storytelling experience.

Panel 1 – Top 10 Languages Have Worked With (Bar Chart)
Most respondents work with *C#* and *HTML/CSS/JavaScript*, often combined with SQL and TypeScript, showing that full-stack web development remains the dominant skill area among developers.

Panel 2 – Top 10 Databases Have Worked With (Column Chart)

PostgreSQL is by far the most commonly used database, followed by *MySQL* and Microsoft SQL Server, indicating a clear preference for open-source relational databases in current professional environments.

Panel 3 – Top 10 Platforms Have Worked With (Word Cloud)
Amazon Web Services (AWS) appears prominently as the most frequently used platform, with *Microsoft Azure* in second place, confirming that cloud computing is central to modern software development workflows.

Panel 4 – Top 10 Web Frameworks Have Worked With (Hierarchy Bubble Chart)
Frameworks like React, Spring Boot, and ASP.NET Core lead developer usage, reflecting a strong mix of frontend flexibility (React) and enterprise-grade backend tools (Spring Boot, ASP.NET).

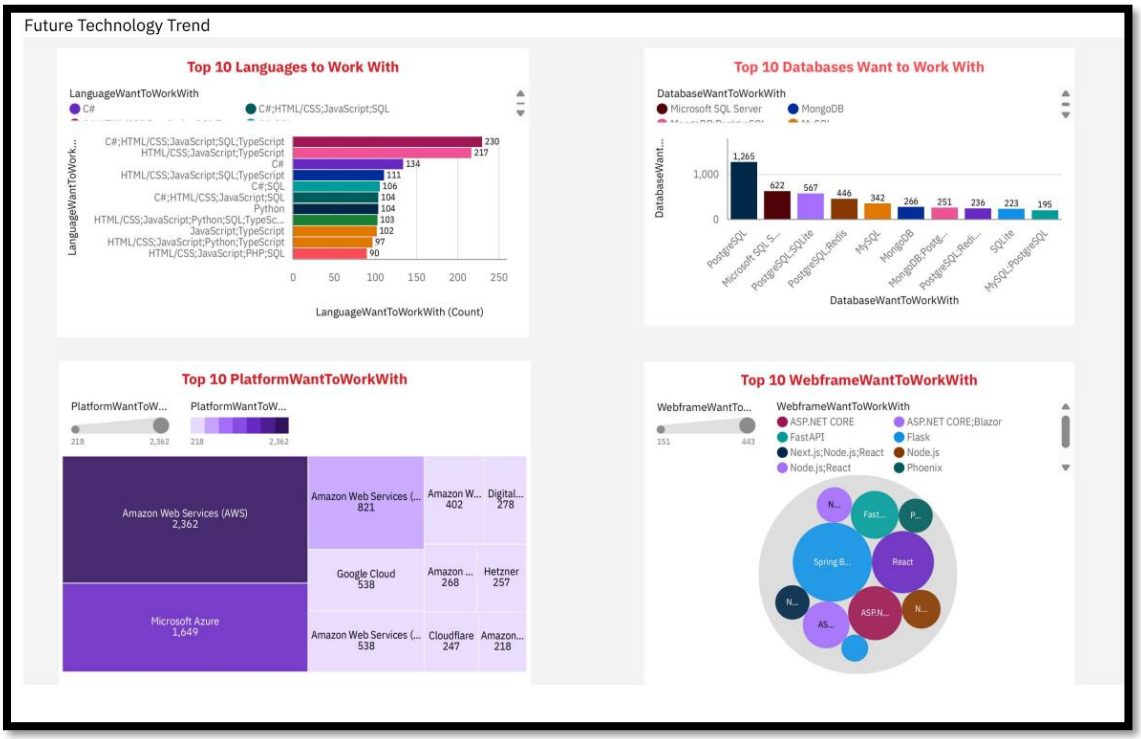
The **Current Technology Usage** dashboard shows that developers primarily work with C# and web languages like HTML, CSS, JavaScript, and SQL. PostgreSQL leads among databases, while AWS and Microsoft Azure dominate cloud platforms. Popular frameworks such as React, Spring Boot, and ASP.NET Core highlight a strong focus on both modern frontend and enterprise backend development.

Panel 1 – Top 10 Languages Want to Work With (Bar Chart)
Developers most want to work with Bash/Shell, HTML/CSS/JavaScript, and Python, showing a continued interest in scripting, web development, and data-related languages.

Panel 2 – Top 10 Databases Want to Work With (Column Chart)
PostgreSQL is again the top choice for future work, reinforcing its reputation as the most preferred open-source database, while Microsoft SQL Server and SQLite follow at a distance.

Panel 3 – Top 10 Platforms Want to Work With (Tree Map)
Amazon Web Services (AWS) dominates developer interest for future projects, with Google Cloud and Microsoft Azure ranking next, confirming a strong shift toward multi-cloud skill development.

Panel 4 – Top 10 Web Frameworks Want to Work With (Hierarchy Bubble Chart)
React, Spring Boot, and Phoenix stand out as the most desired frameworks, reflecting developers' focus on modern, high-performance, and scalable web technologies.



The **Future Technology Trend** dashboard reveals that developers' ambitions closely align with current market trends — emphasizing Python, PostgreSQL, AWS, and React as the leading technologies shaping the next wave of software development.

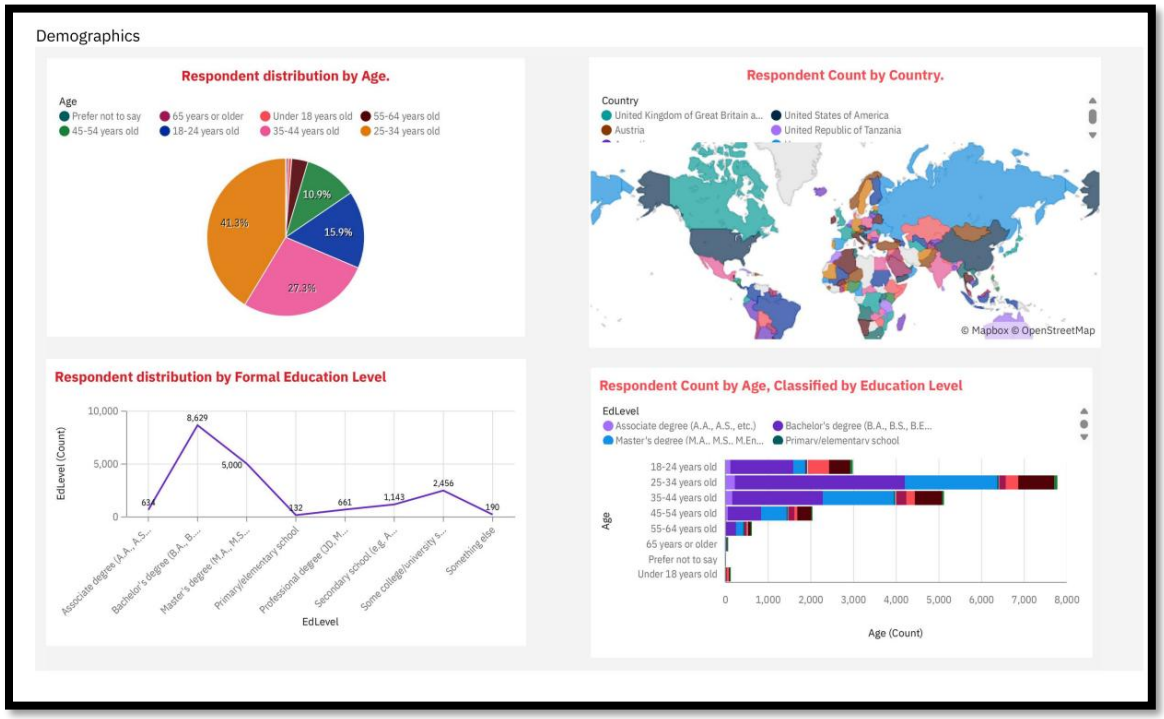
Panel 1 – Respondent Distribution by Age (Pie Chart)

Most respondents are between 25–34 years old (41.3%), followed by 35–44 years old (27.3%), showing that the dataset is dominated by mid-career professionals.

Panel 2 – Respondent Count by Country (Map Chart)
Respondents are globally distributed, with strong representation from the United States, India, and European countries, illustrating the international reach of the developer community.

Panel 3 – Respondent Distribution by Formal Education Level (Line Chart)
A majority of respondents hold a Bachelor's degree (8,629) or Master's degree (5,000), confirming that most professionals in this field have formal higher education backgrounds.

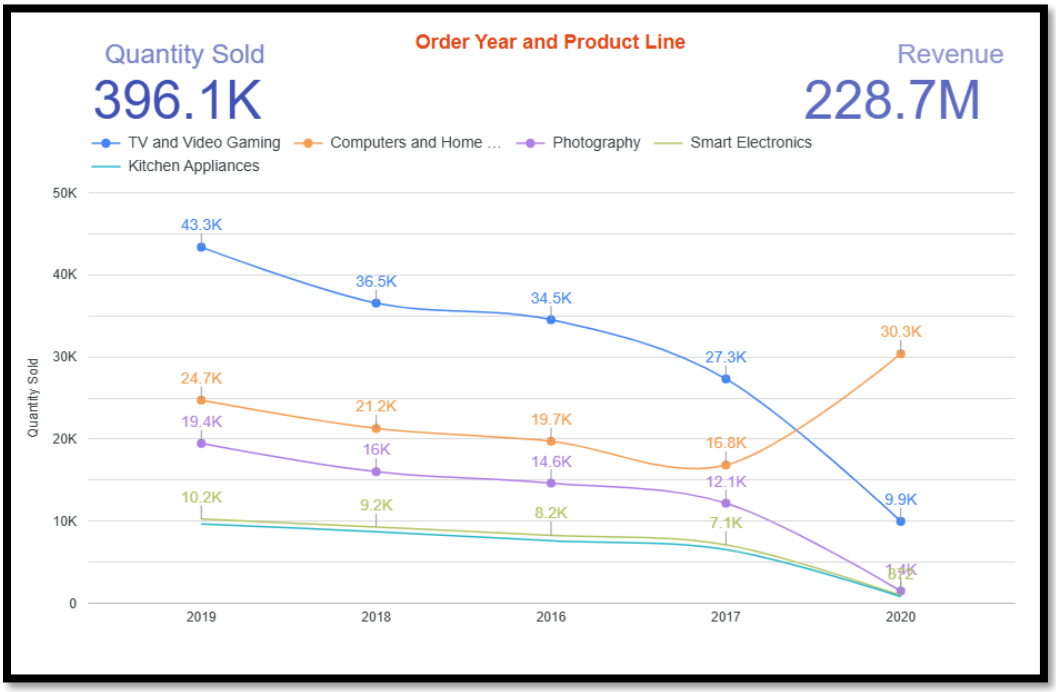
Panel 4 – Respondent Count by Age, Classified by Education Level (Stacked Bar Chart)
The 25–34 age group shows the highest concentration of respondents with Bachelor's and Master's degrees, reinforcing the connection between education and active participation in the tech workforce.



The **Demographics Dashboard** highlights a young, highly educated, and globally diverse developer community.
Most participants are in their mid-career stage (25–44 years old), hold undergraduate or graduate degrees, and come from major technology regions worldwide, reflecting a well-distributed and skilled global talent pool.

Google Looker

Using **Google Looker Studio**, I built an interactive dashboard to visualize sales performance by product line and year, connecting the dataset directly to dynamic visual elements. The dashboard highlights key metrics such as **Quantity Sold** (396.1K) and **Revenue** (228.7M), while the line chart shows trends across product categories like **TV & Video Gaming**, **Computers & Home**, and **Smart Electronics**. Looker Studio's intuitive design tools made it easy to combine metrics, apply filters, and add color-coded visuals that clearly convey sales fluctuations and revenue shifts from 2016 to 2020. Overall, this dashboard demonstrated the platform's strength in turning numerical data into an engaging and easily interpretable story for business decision-making.

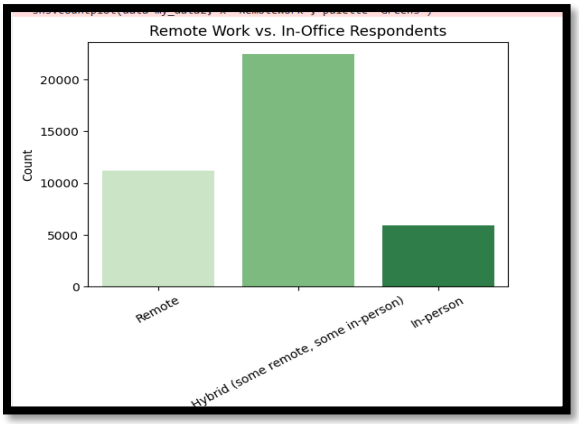


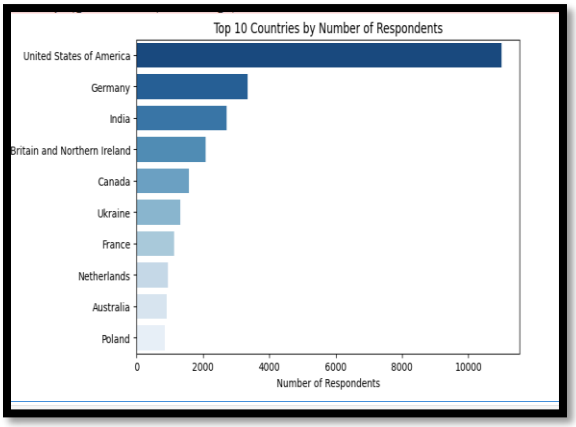
Conclusion

The project successfully applied the end-to-end data analysis process—from data collection and cleaning to transformation, visualization, and insight generation—using the *Stack Overflow survey data* and supplementary job market datasets. Leveraging IBM Cognos Analytics and Google Looker Studio, interactive dashboards were created to explore current technology usage, future developer preferences, and demographic patterns. The analysis revealed strong reliance on web technologies (HTML, CSS, JavaScript, TypeScript), databases such as **PostgreSQL (994)**, and cloud platforms like **AWS (2,301)** and **Azure (1,931)**. Future trends indicate continued

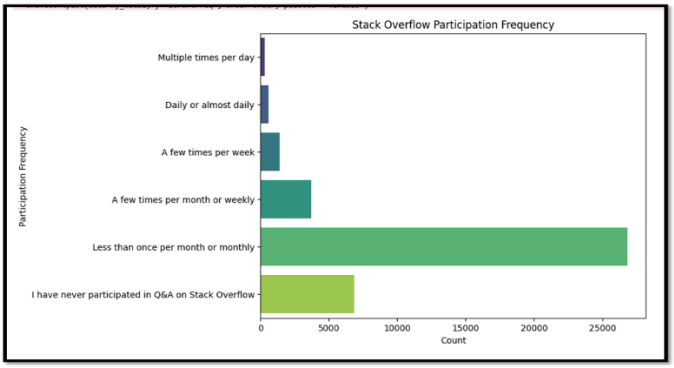
interest in enhancing full-stack skills through modern frameworks such as **React, Spring Boot, and ASP.NET Core**. Demographic insights further showed that most respondents were professionals aged **25–34** with *bachelor's or master's degrees*, reflecting a skilled, globally distributed tech community. Overall, the project demonstrates how business intelligence tools can effectively convert complex datasets into actionable insights, supporting strategic decisions in technology adoption and workforce development.

Appendix

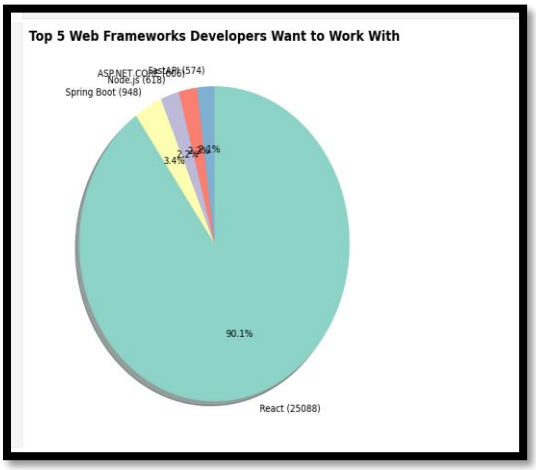


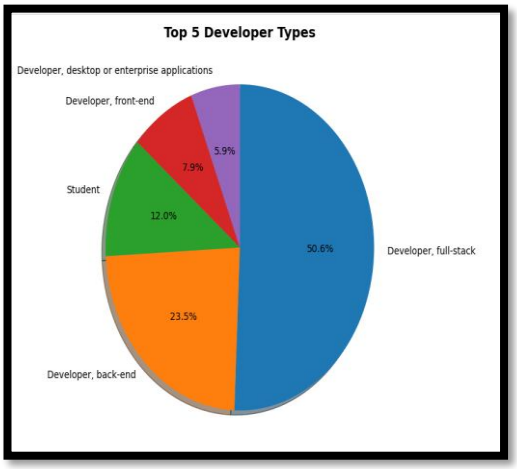


The above graphs reveal that the United States leads in developer representation, followed by Germany, India, and the United Kingdom, showing strong participation from major global tech hubs. Most respondents work in hybrid environments, while fully remote work remains more common than fully in-person roles. These results highlight an increasingly global and flexible workforce, where teams distributed and cross-border collaboration are the new norm.

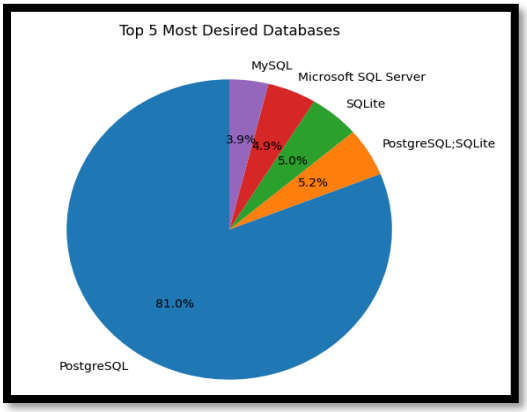


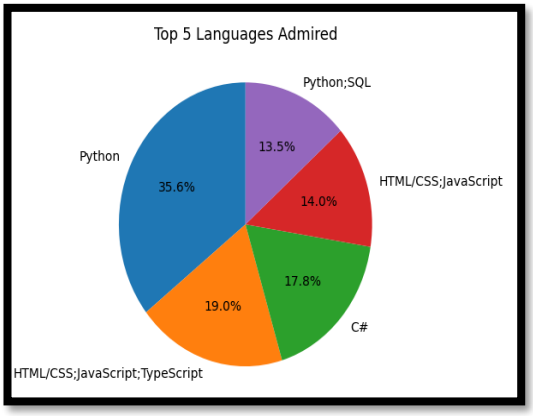
- A significant portion reported *never participating at all*, showing that many developers use the platform more as a reference than as active contributors.





React overwhelmingly dominates future framework preferences, with 90.1% of developers choosing it, far ahead of Spring Boot (3.4%) and Node.js (2.2%).
Over half of respondents identify as ***full-stack developers*** (50.6%), followed by back-end developers (23.5%) and students (12%).





PostgreSQL overwhelmingly leads as the most desired database, capturing **81%** of developer interest, far ahead of SQLite and MySQL.
Python stands out as the most admired language (**35.6%**), followed by HTML/CSS/JavaScript/TypeScript (19%) and **C# (17.8%)**.