### Q1: What is the scope of the project? Why did you choose it?

#### Scope:

The GitHub India Dashboard project aims to analyze GitHub user activity in India using Power BI, SQL, and Python. The project extracts insights about contributions, repositories, followers, programming language trends, and engagement patterns from GitHub profiles and visualizes them in interactive dashboards.

#### Why I Chose It:

I selected this project to:  
 ✔ Gain hands-on experience in data extraction, transformation, and visualization.  
 ✔ Improve SQL querying and Power BI dashboarding skills.  
 ✔ Analyze real-world data to understand trends in the developer community.

### Q2: Can you walk me through the most challenging technical problem you encountered and how you resolved it?

#### Challenge:

The biggest challenge was handling missing data and inconsistencies in the dataset. Some users lacked README files, gender pronouns, or had incorrect joining years.

#### Solution:

✔ Used SQL & Python (Pandas) for data cleaning (e.g., filling missing values with "Unknown" or removing corrupt entries).  
 ✔ Applied DAX functions in Power BI to create derived columns and handle missing values dynamically.

### Q3: What specific technologies did you use, and why did you choose them for this project?

✔ Python (Pandas, NumPy, Matplotlib, Seaborn) – For data cleaning & initial exploratory analysis.  
 ✔ MySQL Workbench – For efficient querying, KPI extraction, and filtering.  
 ✔ Power BI – For interactive data visualization & dashboard development.

🔹 Why these?  
 Each technology was chosen based on its strengths:

* Python for flexibility in data preprocessing.
* SQL for fast querying & relational analysis.
* Power BI for interactive dashboards & real-time insights.

### Q4: Did you encounter any performance issues? How did you identify and address them?

#### Issue:

⚡ Large datasets slowed down Power BI performance, especially when applying complex DAX calculations.

#### Solution:

✔ Optimized SQL queries before importing data into Power BI (e.g., filtered unnecessary columns).  
 ✔ Used Power BI’s DirectQuery mode instead of Import Mode to reduce memory usage.  
 ✔ Created aggregated tables in SQL to avoid redundant calculations in Power BI.

### Q5: Can you explain a situation where you had to make a critical decision that impacted the project's direction?

#### Situation:

Originally, I planned to analyze all global GitHub users, but due to data limitations and performance constraints, I had to narrow the scope to Indian users.

#### Impact of Decision:

✔ Improved query efficiency & dashboard performance.  
 ✔ Allowed for deeper insights into India’s developer ecosystem.  
 ✔ Focused on high-impact KPIs rather than general trends.

### Q7: Were there any trade-offs you had to make while working on the project? How did you prioritize?

#### Trade-offs:

📌 I had to choose between more complex DAX calculations vs. faster dashboard loading times.

#### How I Prioritized:

✔ I pre-processed complex calculations in SQL instead of using Power BI’s DAX to enhance performance.  
 ✔ Limited the number of visuals per dashboard to maintain responsiveness.

### Q8: Describe a piece of feedback you received on your project and how you implemented it.

#### Feedback:

An early version of the dashboard had too many visuals on a single page, making it cluttered.

#### Implementation:

✔ Separated insights into three dedicated dashboards:  
 1 User Contributions & Activity Overview  
 2 Engagement & Popularity Analysis  
 3 Programming Language Trends

✔ Added slicers & filters to allow users to customize the data view.

### Q9: What would you do differently if you were to start this project over again?

🔹 Integrate GitHub’s API to fetch real-time data instead of relying on a static dataset.  
 🔹 Use Azure or AWS services to store & process large-scale GitHub data.  
 🔹 Explore predictive modeling to analyze future GitHub engagement trends.

### Q10: Can you explain the architecture of your project? Why was it designed this way?

🔹 Architecture:

📂 Step 1: Data Collection & Cleaning

* Used Python (Pandas, NumPy) & SQL to clean raw GitHub user data.

📂 Step 2: Data Processing & Querying

* Stored the cleaned dataset in MySQL for efficient querying & aggregations.

📂 Step 3: Power BI Dashboarding

* Connected SQL to Power BI and created interactive dashboards.

✔ Why this design?

* SQL optimized data storage & retrieval.
* Power BI provided real-time visualization.
* Python enabled advanced preprocessing.