# GitHub Indian Users Deep Data Analysis

## Introduction:

India has emerged as a global technology powerhouse, with a rapidly growing developer community on GitHub. This report delves into the **contributions, engagement, and trends** of Indian GitHub users, providing key insights into **programming language preferences, repository activity, and collaboration patterns**. By analyzing user behavior, commit frequency, and technological adoption, this study helps **businesses, educators, and policymakers** understand India's evolving developer ecosystem. With data-driven insights, we uncover the impact of Indian developers on the open-source world and highlight opportunities for growth, innovation, and industry alignment.

## Scope of the Project:

This project analyzes GitHub Indian users' contributions, programming trends, and engagement in open-source development. It covers user demographics, top programming languages, repository activity, collaboration patterns, and hiring trends. The insights will help businesses, educators, and policymakers understand India's growing developer ecosystem and predict future trends.

## Project Objectives:

**->Analyze GitHub Indian Users' Activity** – Identify trends in repositories, commits, and contributions.
**->Understand Programming Language Trends** – Determine the most popular languages among Indiandevelopers.
**->Examine User Demographics** – Explore location, education,

and company affiliations.
**->Track Repository Growth & Engagement** – Measure stars, forks, and issues to assess project popularity.
**->Identify Top Contributors & Influencers** – Recognize highly active and impactful developers.
**->Evaluate Hiring & Career Trends** – Analyze companies hiring GitHub contributors and their preferred skills.
**->Provide Actionable Insights** – Help businesses, educators, and policymakers leverage GitHub data for decision-making.

## Metholodogy:

**->Data Collection** – Extract GitHub Indian users' data from relevant sources.
->**Data Cleaning & Preprocessing** – Handle missing values, duplicates, and ensure data consistency.

->**Exploratory Data Analysis (EDA)** – Use statistical and visual techniques to uncover trends and patterns.
-> **Feature Engineering** – Create meaningful metrics for better insights (e.g., activity scores, repo engagement levels).
->**Visualization & Dashboarding** – Develop interactive Power BI dashboards for dynamic insights.
->**Trend & Pattern Analysis** – Identify key trends in programming languages, contributions, and growth.
->**Segmentation & Comparative Study** – Group users based on engagement levels, demographics, and skills.
->**Actionable Insights & Reporting** – Summarize key findings to help businesses, recruiters, and policymakers make informed decisions.

# Challenges and Solutions:

1.**Data Inconsistency & Missing Values**

- **Issue:** Incomplete or inconsistent user information (e.g., missing locations, incorrect timestamps).
- **Solution**: Implement data cleaning techniques such as handling missing values using imputation, removing duplicates, and standardizing formats to ensure data accuracy.

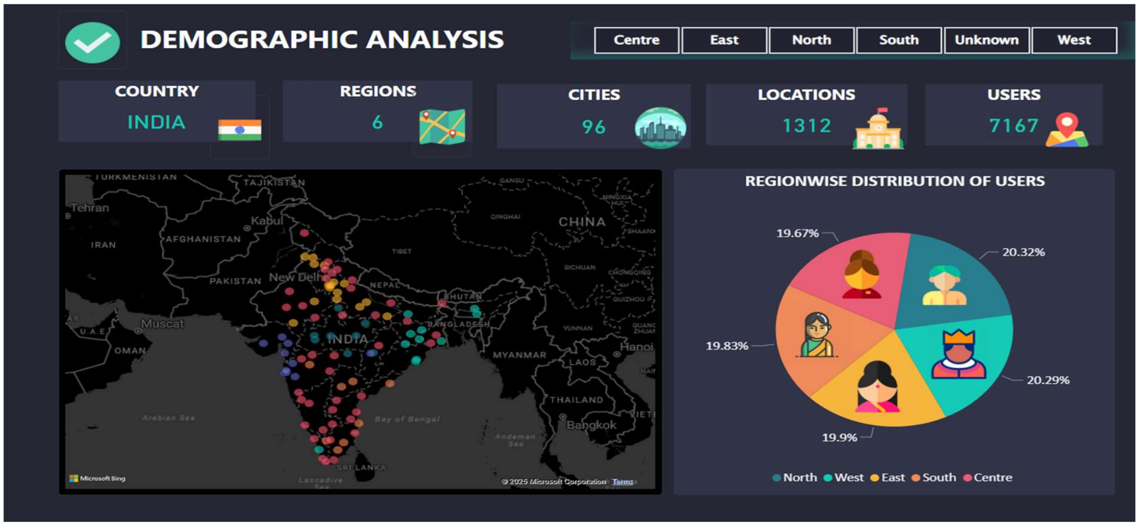
2. **Identifying Genuine User Engagement**

- **Issue**: Distinguishing between active contributors and inactive or bot accounts.
- **Solution**: Use **engagement metrics** like commit frequency, repository stars, and forks to filter out low-activity or non-contributing accounts.

**3**. **Handling Large Data Volumes for Analysis**

- **Issue**: Processing vast amounts of GitHub data efficiently without performance issues.
- **Solution**: Leverage **data aggregation, indexing, and optimized queries** in Power BI and Python (Pandas) to improve performance and visualization speed.

## **Results**:



COUNTRY INDIA

REGIONS 6

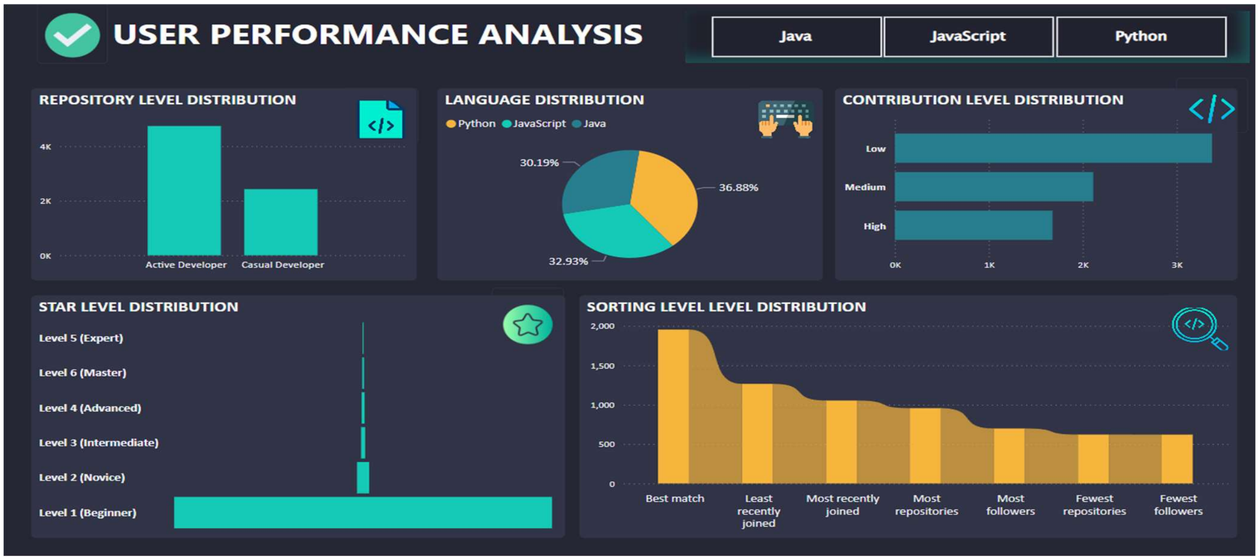
CITIES 96

LOCATIONS 1312

USERS 7167

1. Overall User Statistics:
   - The total number of users is 7,167, spread across various regions in India.
   - There are 1,312 locations and 96 cities covered in the analysis.
2. Regional Distribution:
   - Users are evenly distributed across the regions, with percentages ranging between 19.67% to 20.32%, indicating no significant dominance of any one region.
   - The North region has the highest user share (20.32%), followed closely by Centre (20.29%), while the South has the least (19.67%).
3. Geographical Coverage:
   - A map visualization displays user locations, helping identify areas with high concentrations of users.
   - The widespread distribution of dots suggests that users are present across major urban and semi-urban locations.
4. Region-wise Analysis Capability:
   - The dashboard includes filter options (Centre, East, North, South, West, Unknown) to drill down into region-specific data.
   - This allows targeted demographic analysis and decision-making.
5. User Representation:

- The pie chart visually represents user distribution across regions with icons symbolizing diverse demographics.
- The near-equal distribution of users may indicate a balanced marketing reach or engagement across regions.



1. Repository Level Distribution

   - There are more active developers than casual developers, indicating that a significant portion of users contribute regularly to repositories.

2. Language Distribution

   - The top three programming languages used are:
     - Python (36.88%) – The most widely used language.
     - JavaScript (32.93%) – A close second.
     - Java (30.19%) – Slightly less popular but still significant.
   - This suggests that Python is the preferred language among users, but JavaScript and Java also have strong representation.

3. Contribution Level Distribution

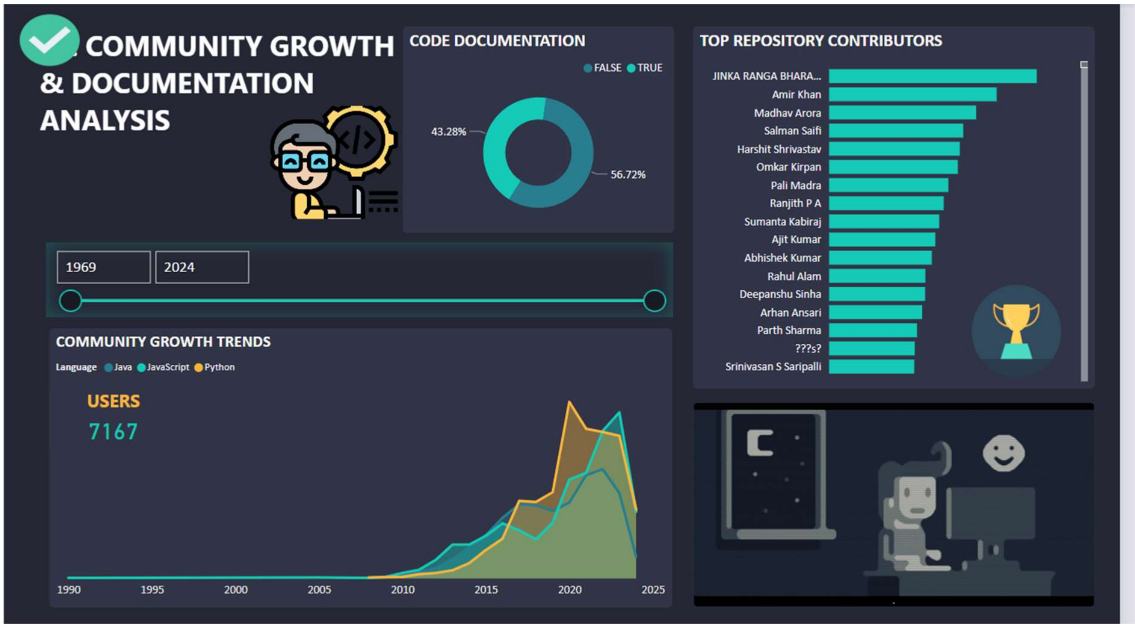
- The majority of users have low contribution levels, followed by medium, and the least number of users have high contribution levels.
- This indicates that while there are many users in the platform, fewer are making significant contributions.

4. Star Level Distribution

- The largest number of users fall under Level 1 (Beginner).
- The higher levels (Expert, Master, and Advanced) have significantly fewer users.
- This suggests that many users are new or less experienced, and there is a need for skill development programs.

5. Sorting Level Distribution

- The most relevant users (Best Match) are the largest category.
- Other sorting criteria like Least Recently Joined, Most Recently Joined, and Most Repositories show decreasing numbers.
- Users with the Fewest Followers and Fewest Repositories are the least represented



1. Community Growth Trends

- The number of users is 7,167, indicating a significant developer community.
- Steady growth from 2005 onwards, with a sharp increase around 2015-2020, followed by a slight decline in recent years.
- Python and JavaScript have shown strong growth, while Java has had a fluctuating presence.

2. Code Documentation Status

- 56.72% of the repositories have proper documentation, while 43.28% lack documentation.
- This suggests that while a majority follow best practices, a significant portion of repositories need better documentation for maintainability and usability.

3. Top Repository Contributors

- Jinka Ranga Bharath is the leading contributor, followed by Amir Khan and Madhav Arora.
- The list highlights key contributors driving repository development.
- There is one unidentified contributor labeled as "???", which may indicate missing data or an anonymous contributor.

4. Growth Over Time

- The growth trajectory indicates an increase in developers over the years, particularly after 2010.
- A slight decline post-2020 may suggest reduced engagement or shifting trends in programming languages.

## Conclusion:

1 . User Performance Analysis: Most users are beginners, with low contribution levels. Active developers outnumber casual ones, and Python, JavaScript, and Java have nearly equal popularity. Sorting preferences favor "Best Match."

2 .Community Growth & Documentation Analysis: The community grew rapidly from 2010-2020 but declined post-2020. Only 56.72% of repositories are well-documented, and contributions are dominated by a few individuals.

3. Contribution & Repository Engagement: Contribution levels are uneven, with a small group of active developers driving engagement. Sorting preferences indicate users prioritize relevance.