## **1. Introduction**

The **GitHub Repository Explorer** is an interactive web application built with **Streamlit**. It allows users to explore and analyze a curated dataset of GitHub repositories, focusing on various metrics like stars, forks, programming languages, and more. Users can filter repositories by topics, programming languages, and activity levels, and visualize these insights through dynamic charts and graphs.

## **2. Project Methodology**

The project follows a systematic approach that includes data collection, cleaning, storage, and visualization, culminating in the creation of an interactive web application.

### ****2.1 Data Collection from GitHub Repositories****

The data was gathered from GitHub using their API:

* **GitHub API Access**:
  + A GitHub Personal Access Token was created to authenticate API requests.
  + Selected trending topics, such as Machine Learning, Data Science, Artificial Intelligence, etc.
* **Data Collection**:
  + For each topic, repositories were fetched using the GitHub API.
  + Extracted key information like:
    - **Repository Name**: The name of the repository.
    - **Owner**: The repository owner's username.
    - **Description**: A brief summary of the repository.
    - **URL**: The direct link to the repository on GitHub.
    - **Programming Language**: The primary language used in the repository.
    - **Creation Date**: The date when the repository was created.
    - **Last Updated Date**: The last time the repository was updated.
    - **Number of Stars**: Stars received, indicating popularity.
    - **Number of Forks**: Number of times the repository was forked.
    - **Number of Open Issues**: Current open issues in the repository.
    - **License Type**: The repository's licensing terms.

### ****2.2 Data Cleaning****

The collected dataset was cleaned to ensure accuracy:

* **Handling Missing Data**:
  + Removed repositories with missing key information.
  + Ensured all numeric values and dates were properly formatted.
* **Data Consistency**:
  + Corrected any inconsistencies in column types (e.g., converting strings to dates).

### ****2.3 Storing Data****

The cleaned data was saved in a CSV file for ease of use, named Non\_Nan\_Repo.csv.

### ****2.4 Building the Streamlit Application****

The Streamlit application was developed to allow users to explore, filter, and visualize the dataset interactively.

* **Loading Data**:
  + The cleaned dataset is loaded into the application for display and analysis.
* **Interactive Filtering**:
  + **Filter by Programming Language**: Users can select repositories written in specific languages.
  + **Filter by Activity Level**: Sliders allow filtering by stars, forks, or open issues.
  + **Search by Topic**: Users can search repositories by entering relevant keywords or repository names.
* **Detailed Visualizations**:
  + **Bar Charts**: Display top repositories by stars, forks, or issues.
  + **Scatter Plots**: Show relationships between stars and forks.
  + **Correlation Heatmaps**: Explore correlations between repository metrics.
  + **Pairplots**: Visualize distributions and relationships between numeric variables.
* **Additional Features**:
  + Users can download the filtered dataset as a CSV file for offline analysis.
  + Interactive charts allow users to zoom in and explore data points in more detail.

## **3. Application Instructions**

### ****3.1 Prerequisites****

Ensure the following are installed:

* **Python**: Version 3.7 or higher.
* **Required Libraries**:

bash

Copy code

pip install streamlit pandas plotly seaborn matplotlib mysql-connector-python

### ****3.2 Running the Application****

1. **Clone the Project Repository**:
   * Download the project files from the GitHub repository.
2. **Run the Streamlit Application**:
   * Navigate to the project directory and run:

bash

Copy code

streamlit run app.py

1. **Access the Application**:
   * Open your browser and visit the URL shown in the terminal, typically http://localhost:8501.

### ****3.3 Using the Application****

* **Apply Filters**:
  + Use the sidebar to filter by programming language, stars, forks, or open issues.
  + Search for repositories by name or keywords in the search box.
* **Visualize Data**:
  + Explore bar charts, scatter plots, and heatmaps.
  + Hover over data points for more details.
* **Download Data**:
  + Download the filtered dataset by clicking the "Download CSV" button.

## **4. Key Findings**

The analysis of the GitHub repository data revealed several key insights:

### ****4.1 Popular Programming Languages****

* **Python** and **JavaScript** dominate the landscape, especially in data science and machine learning repositories.
* These languages are associated with the highest number of stars and forks, reflecting their widespread use and popularity.

### ****4.2 Repository Activity Levels****

* A positive correlation exists between stars and forks, meaning that repositories with more stars tend to attract more contributions.
* Repositories with many open issues suggest active development, though they may also indicate maintenance challenges.

### ****4.3 Licensing Trends****

* A significant number of repositories lack a specified license, which could discourage potential contributors or users due to legal uncertainties.

### ****4.4 Trending Topics****

* Topics like **Machine Learning**, **Deep Learning**, and **Artificial Intelligence** consistently appear at the top, reflecting strong interest in these areas.

## **5. Conclusion**

The **GitHub Repository Explorer** provides users with a robust tool for exploring and analyzing GitHub repositories in the data science field. The combination of interactive filtering and detailed visualizations makes it easier to understand key trends and insights in the open-source ecosystem. The modular design allows for easy updates, making the tool adaptable to future needs.