| | |
|---|---|
| **Project Title** | **GitHub Data Dive** |
| **Skills take away From This Project** | **Python, GitHub API, Pandas, SQL, Streamlit, Data Analysis, Data Visualization** |
| **Domain** | **Open Source Software Analytics** |

**Problem Statement:**

In today's rapidly evolving software development landscape, GitHub serves as a pivotal platform for collaboration and innovation in the open-source community. With millions of repositories available, identifying relevant projects, understanding development trends, and leveraging insights can be challenging for developers, researchers, and organizations alike.

This project aims to extract and analyze data from GitHub repositories focused on specific topics,to uncover patterns and trends in repository characteristics, popularity, and technology usage. By leveraging the GitHub API, the project seeks to provide a comprehensive overview of repository dynamics, including metrics like stars, forks, programming languages, and creation dates.

The ultimate goal is to create a user-friendly Streamlit application that visualizes these insights, enabling users to make informed decisions regarding project collaboration, technology adoption, and educational resource identification in the open-source ecosystem.

**Business Use Cases:**

- Developers can find trending repositories for collaboration or inspiration.
- Organizations can analyze the popularity and activity of repositories related to their technologies.
- Educators and researchers can explore open-source projects for teaching or study materials.

**Approach:**

- **Data Extraction:** Utilize the GitHub API to fetch repository data based on 10 currently trending topics in the data world (e.g., machine learning, data visualization, deep learning, natural language processing, etc.). The following data fields will be extracted:
    - **Repository Name**: Name of the repository.
    - **Owner**: Username of the repository owner.
    - **Description**: Brief description of the repository.
    - **URL**: Link to the repository.
    - **Programming Language**: Primary language used in the repository.
    - **Creation Date**: Date when the repository was created.
    - **Last Updated Date**: Date of the last update to the repository.
    - **Number of Stars**: Count of stars received by the repository.
    - **Number of Forks**: Count of times the repository has been forked.
    - **Number of Open Issues**: Count of open issues in the repository.
    - **License Type**: Type of license under which the repository is released.

- **Data Cleaning:** Handle missing values and ensure data consistency by standardizing formats.
- **Data Storage:** Save the cleaned data in a SQL database for efficient access. Table reference as follows

| Column Name | Data Type | Description |
|---|---|---|
| id | INT | Primary Key (Auto-increment) |
| Repository_Name | VARCHAR | Name of the repository. |
| Owner | VARCHAR | Username of the repository owner. |

| | | |
|---|---|---|
| Description | TEXT | Description of the repository. |
| URL | VARCHAR | Link to the repository. |
| Programming_Language | VARCHAR | Primary language used in the repository. |
| Creation_Date | DATETIME | Date when the repository was created. |
| Last_Updated_Date | DATETIME | Date of the last update to the repository. |
| Number_of_Stars | INT | Count of stars received by the repository. |
| Number_of_Forks | INT | Count of times the repository has been forked. |
| Number_of_Open_Issues | INT | Count of open issues in the repository. |
| License_Type | VARCHAR | Type of license under which the repository is released. |

- **Data Analysis:** Analyze the dataset to uncover trends, such as popular programming languages and repository activity.
- **Visualization:** Build an interactive Streamlit application to display insights visually and allow user interaction.
- **Deployment**: Deploy the Streamlit app on Render for easy public access and scalability.
- **Documentation**: Document the project workflow and summarize key findings for future reference.

**Results:**

- A cleaned and structured dataset containing details of repositories related to 10 trending topics in the data world, including metrics like stars, forks, programming languages, and creation dates.
- A fully functional Streamlit web application that allows users to interactively explore the dataset, visualize insights, and filter repositories by topic, language, and activity level.
- Clear documentation outlining the project methodology, instructions for using the Streamlit application, and explanations of key findings.

**Project Evaluation metrics:**

- **Data Accuracy**: Percentage of correctly extracted data fields compared to the actual data on GitHub.
- **Data Completeness**: Proportion of required data fields successfully extracted (e.g., all specified fields like name, owner, stars, etc.).
- **Code Quality**: Adherence to coding standards and best practices (e.g., use of comments, structure, and modularity of code).
- **Functionality of Streamlit App**:
    - Assessment of the Streamlit application's functionality, including:
    - Correctness of visualizations.
    - User interaction capabilities (e.g., filtering, searching).
    - Performance (loading time, responsiveness).
- **Insightfulness of Analysis:**Quality of insights derived from the data analysis (e.g., trends identified, clarity of reports).
- **User Experience**: Feedback from peers or mentors on the usability and design of the Streamlit application.
- **Documentation Quality**: Clarity and completeness of the project documentation, including setup instructions and project methodology.
- **Presentation Skills**: Effectiveness of the presentation or demo of the project, including clarity, engagement, and ability to answer questions.

**Technical Tags:**

Python, GitHub API, Data Extraction, Data Cleaning, Data Analysis, Pandas, Streamlit, Data Visualization, SQL

**Project Guidelines:**

- **Coding Standards:** Follow PEP 8 guidelines for Python code.
- **Version Control:** Use Git for version control and maintain regular commits.
- **Best Practices:** Write modular, reusable code, and include comments and docstrings.

**Timeline:**

Please note that the project submission deadline is **15th October 2024**.

**Submit your completed project through the following link:**
**https://forms.gle/d5URZJZSxWERPCJf9**

## LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

**About Session:** The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.

**Note: This form will Open on Saturday (after 2 PM ) and Sunday Only on Every Week**

**Timing: Monday-Saturday (11:30AM to 1:00PM)**

**Booking link : https://forms.gle/1m2Gsro41fLtZurRA**

## PROJECT DOUBT CLARIFICATION SESSION ( PROJECT AND CLASS DOUBTS)

**About Session:** The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.
**Note: Book the slot at least before 12:00 Pm on the same day**

**Timing: Tuesday, Thursday, Saturday (5:00PM to 7:00PM)**

**Booking link :https://forms.gle/XC553oSbMJ2Gcfug9**