

Project Report

1. Project Goals

The goal of this project was to use real-world data from external APIs to explore relationships and patterns through data storage.

the project planned to:

1. Collect PokéMon data from the PokéAPI, including basic PokéMon attributes and their associated types.
2. Collect repository data from the GitHub Search API, focusing on repository popularity, such as stars and programming language.
3. Store data from both APIs in a single SQLite database with multiple tables.
4. Collect at least 100 records from each API while limiting inserts to a maximum of 25 records per run.
5. Analyze the collected data.
6. Create visualizations that clearly represent the results of these calculations.

2. Achieved Goals

1. Data was collected from two APIs: PokéAPI and GitHub Search API.
2. A total of 100 PokéMon records and 100 GitHub repository records were stored in an SQLite database.
3. PokéAPI data was stored across two related tables (pokemon and pokemon_types) that share an integer key.
4. Data collection scripts were run multiple times, inserting 25 records per run without modifying the source code.
5. SQL SELECT queries were used to calculate:
 - a. The average PokéMon height grouped by PokéMon type.
 - b. The average number of GitHub repository stars grouped by programming language.
6. Calculated results were written to CSV files.
7. Two visualizations were created using Matplotlib to represent the calculated results.

3. Problems Faced

1. Managing API pagination and batching data so that only 25 items were stored per execution without manually changing the code.
2. Avoiding duplicate entries in the database when running data collection scripts multiple times.
3. Handling missing or null values in GitHub data, particularly for programming languages.

Each issue was resolved through debugging, documentation review, and testing.

4. Calculations from the Database

The following calculations were performed using SQL SELECT queries:

Average Pokémon height by Pokémon type:

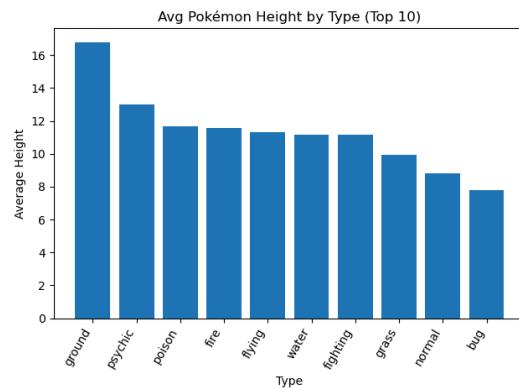
```
type_name,avg_height,count
ground,16.8,10
psychic,13.0,7
poison,11.67741935483871,31
fire,11.55555555555555,9
flying,11.33333333333334,12
water,11.16666666666666,18
fighting,11.16666666666666,6
grass,9.90909090909098,11
normal,8.785714285714286,14
bug,7.8,10
electric,6.0,5
```

Average number of GitHub repository stars by programming language:

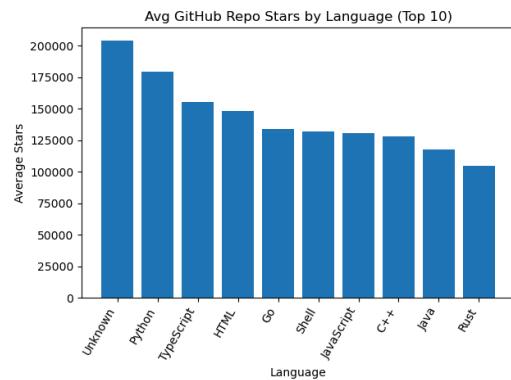
```
language,avg_stars,count
Unknown,204192.35714285713,14
Python,179626.7777777778,18
TypeScript,155172.82352941178,17
HTML,148426.6666666666,3
Go,134082.4,5
Shell,132128.3333333334,3
JavaScript,130678.6666666667,12
C++,128435.8,5
Java,117783.25,4
Rust,104557.25,4
```

5. Visualizations Created

A bar chart showing the average Pokémon height by Pokémon type.



A bar chart showing the average GitHub repository stars by programming language:



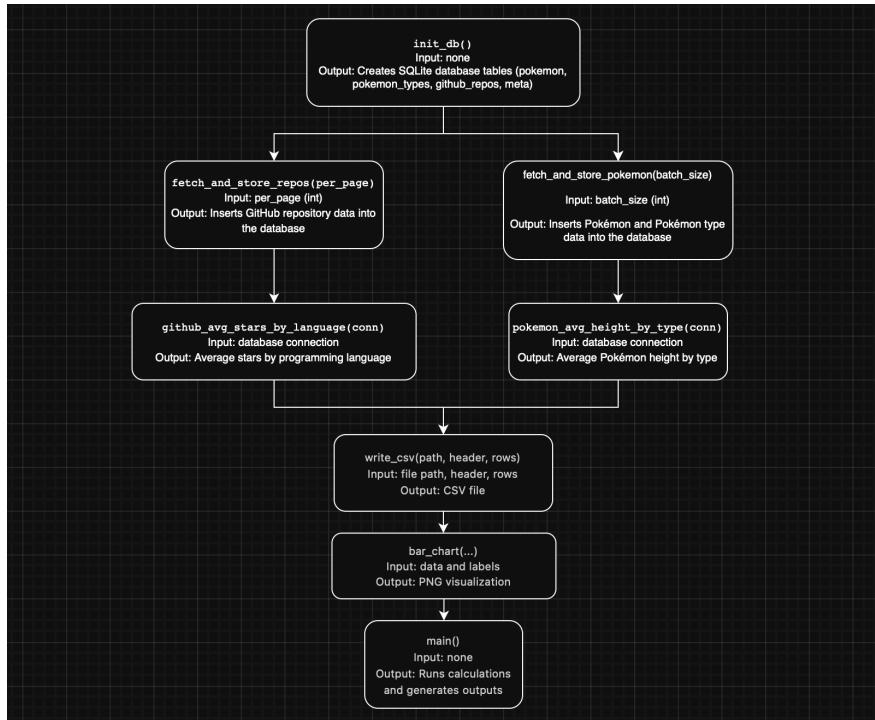
6. Instructions for Running the Code

1. Initialize the database
=> db.py
2. Gather PokéAPI data (run four times):
=> pokemon_collect.py
3. Gather GitHub API data (run four times):
=> github_collect.py
4. Process data, perform calculations, and generate CSV outputs and visualizations:
=> process_and_viz.py
- 5.

(Optional) you can also verify database row counts:

=> check_counts.py

7. Function Diagram (20 points)



All functions were implemented by Zayn Jaber.

8. Resources Used

Date	Issue Description	Location of Resource	Result
12/14/2025	Creating bar charts and saving them as image files	Matplotlib documentation	Yes
12/13/2025	Preventing duplicate entries when inserting data into SQLite	SQLite documentation	Yes
12/10/2025	Needed to understand how to use limit and offset when calling the PokéAPI	PokéAPI documentation	Yes
12/9/2025	Understanding how to search repositories and paginate results with the GitHub API	GitHub REST API documentation	Yes

Github repo: https://github.com/zynjaber/si201_finalproject