**API Data Fetching, Storage, and Visualization**

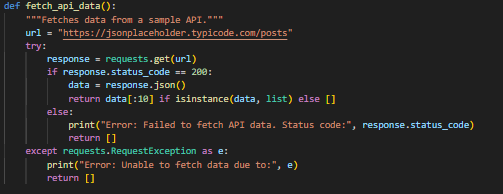
**Name: Samuel Asuoha**

**Student ID: C00305107**

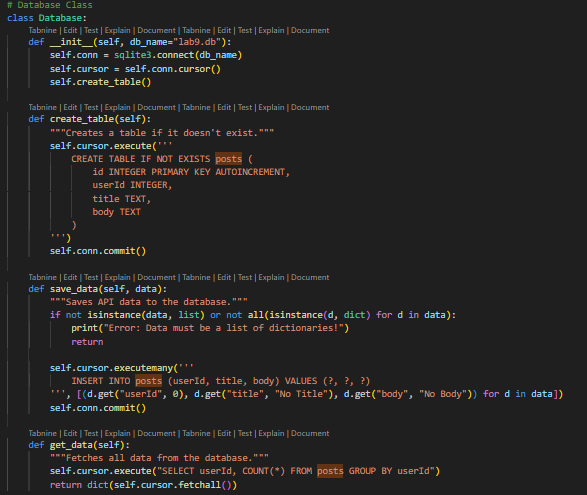
**Summary:** This lab explores the process of fetching data from an API, storing it in a database, and visualizing the results using Python. It integrates SQLite for data management, Matplotlib for data visualization, and a secure login system for user authentication. The goal is to interact with real-world data while ensuring proper error handling and security measures.

**Approach:**

1. **Fetching API Data:**
   * The script retrieves data from ***https://jsonplaceholder.typicode.com/posts*** using the ***requests*** library.
   * The response is processed to ensure it is in the expected JSON format.
   * If valid, the first 10 posts are extracted and prepared for database storage.

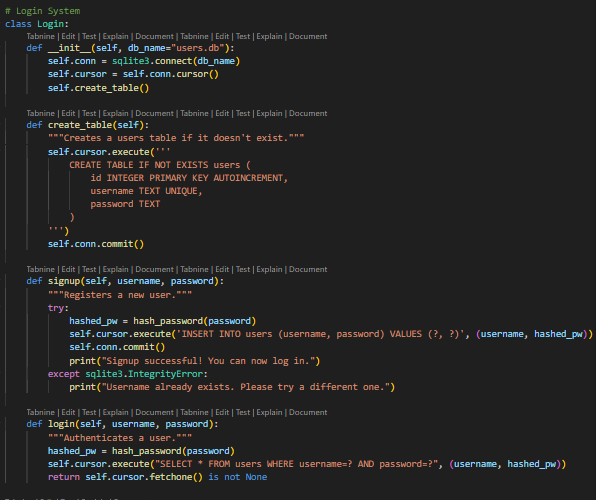


1. **Database Management:**
   * SQLite is used to store user posts in a table named ***posts***.
   * A ***Database*** class handles creating tables, inserting data, and retrieving stored information.
   * Data validation ensures that only structured dictionaries are processed.

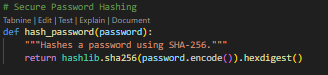


1. **User Authentication:**
   * A ***Login*** class allows users to sign up and log in using a secure system.
   * Passwords are hashed using ***hashlib.sha256(*)** before storage.
   * The authentication process checks user credentials securely before granting access.

**Login class:**

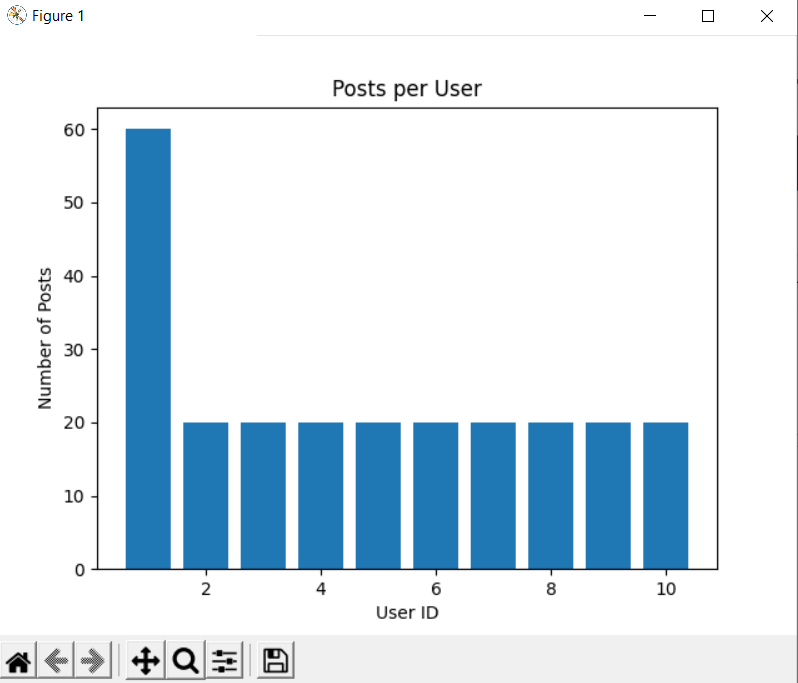


**Hash password function for secure passwords:**

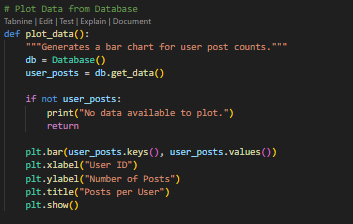


1. **Data Visualization:**
   * The stored data is analyzed to count the number of posts per user.
   * A bar chart is generated using Matplotlib to display post distribution.
   * The script ensures only valid data is processed for visualization.

**Chart appearance:**



**Code:**



**Problems Encountered:**

* **Requests Import Issue:** The ***requests*** library was not recognized, leading to errors when fetching API data. This was resolved (partially) by ensuring it was installed using ***pip install requests****.*
* **Matplotlib Import Issue:** The ***matplotlib.pyplot*** module is underlined in red, indicating an import issue. I tried to get this resolved by verifying the installation with ***pip install matplotlib*** and restarting the IDE. It is still underlined in red but the graph seems to be working.
* **Login Page Interaction:** The login functionality was not interactive in VSCode.I tried to get this resolved by running the script in the terminal instead of the built-in console, but I keep getting ***ModuleNotFoundError: No module named 'requests'***
* **Plot Data Showing Same Values:** The visualization initially displayed uniform values due to incorrect data extraction. This was fixed by adjusting how user posts were counted in the database retrieval function.

**Findings:**

* Implementing proper error handling prevents crashes during API failures.
* Using ***hashlib*** enhances password security by preventing plaintext storage.
* Storing user-generated data in SQLite allows for structured and efficient data retrieval.
* Matplotlib successfully visualizes the distribution of posts per user.
* A functional login system provides user authentication for secure data interaction.

**Conclusion:** This lab successfully demonstrates the process of fetching, storing, and visualizing API data while implementing a secure login system. By incorporating robust data validation and security measures, the project ensures a reliable and user-friendly experience. Future improvements could involve enhanced authentication techniques, such as password salting and token-based authentication, to further strengthen security.

**References:**

* Official Python documentation:<https://docs.python.org/>
* SQLite documentation:<https://www.sqlite.org/docs.html>
* Matplotlib documentation:<https://matplotlib.org/stable/contents.html>