System Usage Instructions for VS Code:

1. **Install Python:** Ensure Python is installed on your machine. You can download it from python.org. Make sure to install at least Python 3.7+.
2. **Install VS Code:** If you don't have VS Code installed , download it from Visual Studio Code website. This will serve as your code editor and development environment.
3. **Set Up Python Environment in VS Code:**
    - Open VS Code and install the Python extension. This can be found by searching for "Python" in the Extensions marketplace inside VS Code.
    - After installing the extension, restart VS Code.
4. **Install Required Dependencies:**
    - Open a terminal window in VS Code.
    - Install the required dependencies by running:

pip install pandas sqlite3

5. This will install pandas for handling CSV/TSV files and sqlite3 for working with the SQLite database.
6. **Place the CSV File:** Make sure you have your CSV file (like assessment25.csv) in the project directory, and update the path in your Python script accordingly. Your line should look like:

python

df = pd.read_csv("C:\\Users\\dell\\Downloads\\assessment25.csv")

Adjust this line if the file is in a different location.

7. **Run the System:**
    - Open the Python script (e.g., main.py) in VS Code.
    - Run the script by clicking the green play button in the top-right corner of the editor or by typing python main.py in the terminal.
    - The program will ask you to input a search term (e.g., "music"). After entering the term, it will display the search results in the terminal.

Design Choices Explanation:

1. **SQLite Database for Efficiency:** SQLite is lightweight, easy to set up, and does not require any server configuration. It efficiently stores and queries data even for large datasets. We've created a local database (tweets_data.db) to store the tweets for faster querying.
2. **Pandas for Data Processing:** pandas is used to handle the ingestion of large CSV/TSV files because it allows efficient data manipulation. Once the file is loaded, relevant data columns are extracted and pushed into the SQLite database.

3. **Simple Query Interface:** The search interface is terminal-based, allowing users to enter any term to filter the tweets. It's straightforward for querying the dataset and outputting relevant information like tweets per day, unique users, average likes, and more.
4. **Terminal-based Interaction:** Since this is a simple project, there is no need for a GUI or complex web interface. Users can interact with the program directly through the terminal, making the system lightweight and easy to use on any machine with Python.

Example of Expected Output:

Enter the term to search for: music

--- Search Results for Term: 'music' ---

1. Tweets per Day:

   2023-04-15: 120 tweets

   2023-04-16: 150 tweets

   ...

2. Unique Users: 2112

3. Average Likes: 161.19

4. Places (Place IDs):

   3b77caf94bfc81fe: 1 tweet

   ecbe2aea853af44e: 2 tweets

5. Times of Day (Hour):

   10: 56 tweets

   11: 72 tweets

6. Most Active User: musicfan with 91 tweet