Step 1: Install Required Dependencies

To run this system on your local machine, you will need to install the necessary libraries.

i. **pandas**: For data manipulation and cleaning.
ii. **numpy**: For numerical operations.
iii. **Jupyter Notebook**: For running and interacting with the system easily.

To install these dependencies, use the following command in your terminal or command prompt:

pip install pandas numpy jupyter

Step 2: Download the Code

Clone or download the repository containing the system. This repository should include:

1. The dataset file (csv or tsv).
2. The Python scripts for data ingestion and querying.
3. Instructions for how to query the system.

For now, assume you have the following files:

- data.csv: Contains the dataset.
- system.ipynb: Contains the Jupyter Notebook code for data ingestion and querying.

Step 3: Running the System in Jupyter Notebook

1. Open a terminal, navigate to the folder where the code is saved, and launch Jupyter Notebook.
2. This will open Jupyter in your web browser. Find the system.ipynb file and open it. It contains all the necessary code to load the data, clean it, and query it using the Pandas library.

Step 4: Data Ingestion and Cleaning

Once the system is running, the first step is to ingest and clean the data. The following steps are executed automatically as part of the notebook:

1. **Data Loading**: The data is loaded into a Pandas DataFrame. This allows for efficient manipulation and querying.

```
import pandas as pd

# Load the dataset

df = pd.read_csv('data.csv')
```

2. **Data Cleaning**: Before running any queries, the data is cleaned. This step handles missing values, converts columns to appropriate data types, and prepares the data for analysis. For example:
    - Converting retweeted counts and follower counts to numeric values.
    - Handling missing data in hashtags and mentioned_handles.

```
# Clean the data
```

```
df['retweeted_follower_count'] = pd.to_numeric(df['retweeted_follower_count'], errors='coerce').fillna(0)

df['retweeted'] = pd.to_numeric(df['retweeted'], errors='coerce').fillna(0)

df['hashtags'] = df['hashtags'].apply(lambda x: x if isinstance(x, list) else [])
```

Step 5: Running Queries (Using Custom Functions)

Once the data is loaded and cleaned, you can start running custom queries. For example, the tweets_per_day(term) function from Part 2 can be used to count the number of tweets for a specific term on each day.

To run this in the Jupyter notebook, simply call the function:

```
term = 'Britney'

print(tweets_per_day(term))
```

You can also compute other metrics like the average likes (from the formula we discussed earlier):

```
print(average_likes(term))
```

Step 6: Querying the System

You can create and run more complex queries to gain insights from the dataset. Here are some examples:

1. **Query 1: Total Tweets Mentioning a Term**

   Use the tweets_per_day function to count the number of tweets for any term, such as "Britney":

   ```
   tweets_per_day('Britney')
   ```

2. **Query 2: Calculate Average Likes**

   Using the average_likes function, you can calculate an estimate of tweet popularity based on retweets, hashtags, and mentions:

   ```
   average_likes('Britney').
   ```

3. **Query 3: Most Retweeted Tweets**

   You can sort the data by retweeted_follower_count to find the most influential retweeted tweets:

   ```
   df.sort_values(by='retweeted_follower_count', ascending=False).head()
   ```