General System Overview

Welcome to our Twitter Replicator Project made using Python and MongoDB!! Choose an option to get started!

1. Tweet Search:

Think of keywords as your magic words. Type them into the search bar and watch the tweets that contain these words appear before your eyes. Each tweet will tell you its own story, revealing its id, date, content, and the username of its author.
Want to know more about a tweet? Just select it and you can check all its other aspects, including the url, replyCount, retweetCount, likeCount, and quoteCount.

2. User Search:

Looking for someone? Enter their name into the search bar and all users with matching usernames will appear.
Each user will introduce themselves by their username, displayname, and location.
Want to know more about a user? Just select them and all their details will be revealed.

3. Top Tweets:

Want to know what's trending? Enter a number 'n' and select a field (retweetCount, likeCount, quoteCount) to see the top 'n' tweets based on the selected field.
Each tweet will present its id, date, content, and the username of the person who posted it.
Want to know more about a tweet? Just select it and all its details will be revealed.

4. Top Followed Users:

Ever wondered who's the most popular? Enter a number 'n' and you'll see the top 'n' users ranked by followersCount.
Each user will step forward and present their username, displayname, and followersCount.
Want to know more about a user? Just select them and all their details will be revealed.

5. Compose a Tweet:

Enter your tweet content and press enter. Your tweet will be added to the database with the date field set to the system. All other fields will be set to null or zero.
Now everyone can see what you have said.

6. Exit Program:

Sad to see you go! Start up the program again at any time to interact with others. Bye!!

Algorithm Details

Testing Strategy

The test begins with Phase 1, where the dataset of 100 entries is loaded into the database. The time it takes to load the data and create necessary indexes was measured, being 46s being less than 3 min as specified. This being after indexing the data, correctness of the database or collection and the efficiency of the code are verified.

Next, we moved to Phase 2, which was divided into several parts. The first part was the search for tweets. This was tested with one keyword, multiple keywords, and case variations of keywords. It was verified that all tweets matching the keywords were returned and that for each matching tweet, the id, date, content, and the username of its author were shown. A tweet was selected to verify that all fields including the url, replyCount, retweetCount, likeCount, and quoteCount were shown.

The second part was the search for users. This was tested with a keyword and case variations of the keyword. It was verified that all authors matching the keyword were returned and that for each user, the username, displayname, and location were displayed. It was also verified that listed users did not have the same id or the same username. A user was selected to verify that all fields of the user were shown.

The third part was listing the top users. This was tested with different values of ‘n’. It was verified that the top ‘n’ users based on followersCount were listed and that for each user, the username, displayname, and followersCount were listed. A user was selected to verify that all fields of the user were shown.

The fourth part was listing the top tweets. This was tested with different values of ‘n’ and different fields (retweetCount, likeCount, quoteCount). It was verified that the top ‘n’ tweets based on the selected field were listed and that for each tweet, the tweet id, date, content, and the username of the person who posted it were displayed. A tweet was selected to verify that all fields were shown.

The final part was composing a tweet. A tweet was composed by entering tweet content. It was verified that the tweet was inserted into the database with the date field set to the system date and username set to “291user”. All other fields were set to null or zero.

This test plan ensures that all functionalities of the database are tested thoroughly according to the provided rubric.