

TweetBook – Social Media Platform

General Overview:

TweetBook is a CLI-based social media platform written in Python. It allows users to search and list tweets or users and also compose tweets of their own. The data is stored in a MongoDB database.

User Guide:

Landing Page:

- To search for tweets, press 'T'
- To search for users, press 'U'
- To list top 'n' tweets, type 'LT'
- To list top 'n' users, type 'LU'
- To compose a tweet, press 'C'
- To exit, press 'X'

Search Tweets:

- Enter the keyword you'd like to search in the database
- To get more information about the obtained tweets, enter the corresponding tweet number

Search Users:

- Enter the keyword you'd like to search in the database
- To get more information about the obtained users, enter the corresponding user number

List Tweets:

- Enter the number of tweets you want to see
- Enter the criteria by which you'd like to sort the tweets

- To get more information about the obtained tweets, enter the corresponding tweet number

List Users:

- Enter the number of users you want to see
- To get more information about the obtained users, enter the corresponding tweet number

Compose Tweets:

- Type in the content of your Tweet

Detailed Software Design:

Architecture:

- The system is built using a modular design.
- It consists of multiple functions responsible for handling each functionality and page.

Database:

- MongoDB was used to store all the tweets and its subsequent information.
- It makes use of indexes to speed up queries

Data Retrieval and Display:

- MongoDB queries are used to fetch any relevant data which is then displayed using a CLI interface designed and executed by Python.
- We use indexes to speed up the data retrieval in large datasets.

User Input and Error Handling:

- CLI inputs are validated to make sure no errors occur.

Testing Strategy:

We had done a variety of tests during the development process. All these tests were performed manually using different JSON files.

Unit Testing:

After each function has been implemented, we tested it individually to ensure it performed as expected. Functions that involved user inputs were also tested to see whether it broke or failed when incorrect inputs were used.

Integration Testing:

After testing the necessary functions and integrating it with other functions, we ran a few more tests to make sure it behaved as expected with the rest of the program.

Group Work Strategy:

We used GitHub as a version control and code-sharing platform and Discord for communicating. We also used VSCode LiveShare to program collaboratively.

Tasks were allocated so that functions that required similar implementation were handled by the same person. More specifically:

- Nathan Thai (~ 8 hours): List Tweets, Search Users, Compose Tweet
- Muhammed Rayyan Rashid (~ 8 hours): Part 1, Landing Page, Bugs, Documentation
- Kevin Frito (~ 7 hours): Search Tweets, Code Comments
- Samrathjit Sandhu (~ 7 hours): List Users, Code Comments