TweetBook – Social Media Platform

General Overview:

TweetBook is a CLI-based social media platform written in Python. It allows users to sign up and log into their accounts to view, search, compose and reply to tweets. It also allows users to search for and follow different users. The data is stored in an SQLite database.

User Guide:

Landing Page:

- To login, press ‘L’
- To sign up, press ‘S’
- To exit, press ‘X’

Sign Up Page:

- Provide your personal information as per the prompts.
- Upon successful sign up, you will be directed back to the Landing Page.

Login Page:

- Enter your User ID and Password.
- You can try again if the credentials are wrong.
- After logging in, you will be directed to your Tweets Page.

Tweets Page:

- Displays tweets and retweets of the users you follow.
- You can navigate through the tweets or compose new ones.
- You can select a tweet to view additional statistics or perform an action on that tweet such as retweet or reply.

- You can also view your followers, and search for users or tweets.
- You can also log out from this page back to the Landing Page.

Compose Tweet:

- Allows you to post a tweet of your own or reply to an existing tweet.

Followers Page:

- Allows you to view all your followers.
- You can navigate through your followers list and get a detailed view of them.
- After selecting a follower, you can view their statistics, navigate through their tweets, or choose to follow them.

Search Users:

- Allows you to search for users based on keywords in their name or city.
- You can view that user's details, navigate through their tweets or choose to follow them.

Search Tweets:

- Allows you to search for tweets based on keywords in the text.
- Just like the tweets page, you can select any of the resulting tweets to view additional statistics or perform an action such as retweeting or replying.

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:

- SQLite was used to store all the user data, tweets, followers, and other relevant information.
- Tables used: users, tweets, follows, mentions, hashtags, retweets.

User Authentication:

- User credentials are used for verification against the database during login.

Data Retrieval and Display:

- SQL queries are used to fetch any relevant data which is then displayed using a CLI interface designed and executed by Python.

User Input and Error Handling:

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

SQL Injection Attacks:

- Use of parameterized queries and placeholders (like '?') ensures SQL attacks are countered.

Testing Strategy:

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

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 in a way such that functions that required similar implementation were handled by the same person. More specifically:

- Nathan Thai (~ 9 hours): Landing Page, Sign Up Page, Login Page, Tweet Action, Compose Tweet
- Muhammed Rayyan Rashid (~ 7 hours): Tweets Page, Followers Page, Follower Actions, Search Users
- Kevin Frito: (~ 7 hours) Search Tweets, SQL Injection and Error Checks
- Samrathjit Sandhu (~ 6 hours): Password Invisibility, Code Documentation, SQL Injection and Error Checks