CMPUT 291 Mini Project 2 Report

Cynthia Li (sqli)   Michael Kwok (mkwok1)

November 2020

1 Project Overview

This program utilizes an online MongoDB NoSQL database to store data for posts, votes, and tags. The user interacts with the system by selecting options displayed on screen in each state.

Upon start up, the user may choose to enter a numeric user id. If the user id is valid (contains only numbers), a user report is generated which outlines the number of questions and answers the given user has posted, along with their average scores, then displays the total votes the user has made. Otherwise if a user id was not provided or if the user id is invalid, the program proceeds to the main menu.

In the main menu, the user has the option to post a question, search for questions, or exit the program. In posting a question, the user is asked to enter a title, body, and tags for the post. Upon finishing, the user is taken to view the newly posted question with all its newly generated data that is stored in the database. From there, several options are presented to the user: answer the question, list all answers to the question, vote on the question, and return to the main menu. Answering the question will have an extra line appear to prompt the user to write an answer to the question, to which once the user is finished, will take the user to view the answer inidivudally. The answer view shows the original question of which the answer is responding to, along with the answer itself as well as two actions the user can take: vote or return to main menu.

When the user selects list answers from the question screen, a new table will be displayed on a new screen that shows the accepted answer first (if any exists), and then other answers to the question. The user can select any of the answers listed by their number and go to view the answer individually.

When the user selects the vote action on a question or answer in focus, the score of the post wil increment by 1 if the user is logged in and has not previously voted on the post already. Users who did not provide a user id at the login screen are free to vote without constraint on the post.

If the user chooses to search questions, they will be first prompted to enter keywords delimited by spaces. Upon pressing enter, a table of search results will be displayed in pages of 5. The user can select any of the questions by their numbers in the results or select the see more option to display the next page of search results. The user may also choose to return to menu on the search results page.

2 Running Instructions

To install the dependencies of this program, run the following:

```
pip3 install -r requirements.txt
```

To run the program, the following commands can be used:

```
python3.5 phase1.py
python3.5 main.py
```

3 Design

The first phase of the project is written as a simple script as it's main job is to import the data and no processing.

In this project, the user interface and database access functions are split into two files. The main file

The user interface and control flow is handled by a state machine. States are defined when the program starts, with and each state having it's own output for the user interface. The state machine calls into the Database class to perform actions, fitting both the View and Controller in MVC, also allowing for cleaner seperation between parts, allowing simpler bug fixes and ensuring that bugs in one part will not affect the other.

4 Testing Strategy

5 Groupwork Breakdown

The group coordinated through WhatsApp direct message, and code was hosted in a private GitHub repository.

- Cynthia
  - Time estimate: 10 hours
  - Designed user flow
  - Implemented most states
  - Created user interface
  - Co-authored the report document
- Michael
  - Time Estimate: 15 hours
  - Wrote all of `database.py`
  - Unit tests in `tests.py` and corresponding test data
  - Implemented the list answers and some of search states
  - Co-authored the report document