A general overview of the system with a small user guide

For phase1 the system begins by asking the user for the port number they wish to connect to then the database is built from the json files.
Once this is over the user can proceed to the code for phase2.

In phase2:
The system begins by asking the user for the port number they wish to connect to.
The system begins by asking the user if they wish to provide a userid.
The user must type y or n to select an action.
For every invalid user input throughout the program, the program displays an error message and prompts the user to enter a valid input.

If the user selects **y**:

The user is prompted to enter a uid.

Then the user report is displayed.

Then the menu options are displayed.

The user then selects which task they want to perform. The task menu contains 6 options, one for each task and one to exit the program.

If the user selects **n:**

Then the menu options are displayed.

The user then selects which task they want to perform. The task menu contains 6 options, one for each task and one to exit the program.

If the user selects **exit** at any point in the program then a goodbye message is displayed and the program is terminated.

Details of the design of the software

Both Phase1 and Phase2 begin with all necessary import statements and setting up the connection to mongodb and the database.

The project was approached using functional programming.

For phase1 there are 3 functions, one to remove all html tags, the other to insert the data from json files to the database and this includes the terms array too and the last function is to build the index on the terms array in the posts collection.

For phase2 there is a function for every task and search (Q2) uses a helper function too. Furthermore there is a function login to ask the user if they wish to provide a userid then there is a function report which displays the report if the user provided a userid and there is a function menu which displays all menu options and calls the respective functions to perform the selected task.

Each function has a block comment which describes how the function does its task and the parameters used and the return value. There are inline comments too to explain specific lines of code.

Testing Strategy

We used the json files provided and in phase1 we built the database. This database was used for all the testing for phase2. Initially we tested on our local computers on pycharm and used Compass to check if the correct changes are made in the database. Then once we had optimized phase1 we did the testing on the lab computer.

After finishing our individual questions, we all tested our questions on our own.

In every meeting we held, we tested the whole code from scratch and fixed mistakes and debugged errors. We analyzed and tested each function and checked if it is making the necessary changes in the database. We also checked for invalid inputs.

During the meetings, the testing and debugging was done by Tanyaradzwa(gozhora) and he shared his screen and we all used our combined ideas and efforts to fix the bugs and mistakes we had.

We ensured that phase1 is done under 5minutes by optimizing our code for instance by reducing time complexity from O(n^3) to O(n^2) and by replacing the set operation union with concatenation since it is faster.

Group Work Breakdown

The platform for discussion and sharing the code was a discord group which only had 3 of us group members and the platform to hold meetings was google meet using our ualberta emails.

Phase1 was done by Akrash(akrash) which took approximately 3hours and initially the runtime was about 15minutes then the optimization was done later by Tanyaradzwa(gozhora) and Robert(rjoseph1).

Q1 was done by Robert(rjoseph1) which took approximately 1hour.

Q3 was done by Akrash(akrash) which took approximately 1hour.

Q2 was done by Tanyaradzwa(gozhora) which took approximately 5hours.

Then we held a meeting where function to test our codes. The meeting took approximately 3hours.

Since Q2 seemed bulky we decided to have one person do Q2 then each of the other two members to do the remaining questions (i.e Q4 and Q5) then the person who did Q2 can do some part of the report(for the users with a userid).

Q4 was done by Robert(rjoseph1) which took approximately 2hours.

Q5 was done by Akrash(akrash) which took approximately 2hours.

The report q1 and q2 were done by Tanyaradzwa(gozhora) which took approximately 1hour.

The report q3 was done by Akrash(akrash) which took approximately 30 minutes.

Then we held a meeting to test our codes. The meeting took approximately 3hours.

Then we all worked on optimizing the code individually and Tanyaradzwa(gozhora) managed to bring down the runtime from 15 minutes to less than 5 minutes. It took him approximately 4 hours since he changed the code significantly. Then Robert(rjoseph1) further optimized the code by making small smart edits (e.g using concatenation instead of union and a few other changes) which took him 30minutes.

Then we held a meeting and tested all our questions from scratch and did a lot of debugging and error fixing. The meeting took approximately 4hours. We had a 20 minutes break in the meeting. We further optimized phase1 in the meeting.

The report.pdf was made by Akrash(akrash) which took approximately 1hour.