AST-Assignment #4

Prof: Nico Hochgeschwender
T.A.: Samuel Parra: samuel.parra@smail.inf.h-brs.de

June 2020

We have made a lot of progress in the Facebook clone app, but now it's time to introduce it's final feature... Chat rooms! In this assignment, we will refactor the code base of the Facebook app to create a chat room functionalities to the groups. In order to do so we will employ a communication protocol called WebSockets.

1   WebSockets

WebSocket is a communication protocol that provides *full-duplex* communication over a TCP connection. Originally, for a client to receive updates from a server the client had to poll the server several times per minute, which wastes resources. To solve this WebSockets were introduced, they provide a persistent connection between a client and a server that can be use to send messages without needing to open a connection every time.

You can read more about them here: `https://blog.teamtreehouse.com/ an-introduction-to-websockets`. (Keep in mind this blog post is oriented towards web developers)

2   User Case

In order to see what your system should be able to perform, let's consider a use case. Consider three users: Alex, Sue, and Sandy. They are fans of British panel shows: Alex and Sue like "8 out of 10 cats", Sue and Sandy are fans of "Quite Interesting", and the three of them are fans of "Taskmaster". They are members of the Facebook fan groups of the shows they like, and after each episode they like to discuss it using the newest feature: the group chat rooms.

After a new episode, they log into Facebook and select the option "Join a chat room". They are prompted with the choice of which group chat room they would like to join, and after making a selection they join the conversation. Users can only connect to groups they are members to, and can only see the messages if they are inside the chat room. When they join the chat room they get to see the previous 10 messages in the conversation to get some context.

3   Python WebSockets

We will employ the websocket python library. Please take your time to read the documentation: `https://websockets.readthedocs.io/en/stable/intro.html`. You can ignore the sections "Secure example" and "Browser-based example" as they are not relevant for our application. Keep in mind that this library requires Python $\geq$ 3.6.1.

4   Set up your Facebook system

This new feature requires that you, once again, refactor your code. You will be creating a `client` module that interacts with your Facebook system, and a server that communicates with your client. You can find a structure of the server in `server.py` file found in the code folder. All you have to do is fill in the logic.

Your Facebook system must populate the three users and groups mentioned earlier when initiated. Please, use the following data to initialize your system:

username	member of
Alex	8 out of 10 cats, Taskmaster
Sue	8 out of 10 cats, Quite Interesting, Taskmaster
Sandy	Quite Interesting, Taskmaster

Set all the passwords to "password" and use dummy data for any additional data you need to create a User object.

5   assignment submission

- This assignment can be submitted by individuals or pairs only.

- The deadline for this assignment is **06.07.2020**. Please *do not* make more commits on your GitHub repository after the deadline.

- All models should be placed on the `./docs` folder.

- All the code should be place on the `./code` folder.

- Submission is made through GitHub. Please, try to commit your progress constantly as commits after the deadline won't be taken into consideration.

6   Notes

- We expect a command line interface (CLI) to use your program, not a GUI.

- Don't forget to populate the database with the data provided.

- You should be able to explain your code and your design decisions.

- Your code should be well commented.

- Update `requirements.txt` before submitting.

- You can contact the T.A. for any questions you may have with the email address provided at the beginning of this document. Please, write [AST-Assignment #] in the subject and your name(s) in the body of the email.

7 Resources

- Code Style: `https://docs.python-guide.org/writing/style/`