**COMP3011 – Web Services and Web Data**

Jacob Stockwell – Semester 2, 2024

**Coursework 1 Report**

A screen shot of a computer

Description automatically generated**Client Application**The client application is implemented as a simple terminal application. Once run, using ```python client.py```, the program runs on a non-terminating while loop prompting the user to enter commands (see figure 1).

*Figure 1*

When commands are input, they are parsed using the argparse library, allowing for easy separation of switches and command arguments. A singular session is initiated and maintained as the program runs, in order to keep consistency between requests, and allow for login/user authentication.

**Login**

A screen shot of a computer screen

Description automatically generatedLogin with “login <url>”, where url is the full qualified URL for the request. A supernumary “/” will be appended if forgotten, before attempting to access API services. Upon successful login, the session username will be saved, indicating an authenticated session. A 200 status code is checked client-side to ensure that the login was successful, and a Boolean flag set to indicate so, as well as a brief welcome message.. This Boolean flag is used to check whether the client is logged in, allowing for basic client-side checks that reduce unnecessary network traffic; for example, attempting to request a story creation when not logged in. See figure 2 for example output.

*Figure 2*

**Logout**

A close up of a logo

Description automatically generatedA post request is sent to api/logout/. A 200 status code indicates a successful logout, and as such the Boolean flag is set for this. See figure 3 for example output.

*Figure 3*

**Post story**

A screenshot of a computer screen

Description automatically generatedPost information is collected through prompting the user. No client side checks for validity are conducted. The user is prompted for the headline, category, story details and region, and will receive a relevant message if the story is created. See figure 4.

*Figure 4*

A screen shot of a computer

Description automatically generatedIf the details provided are invalid, for example a headline that is too long, or an invalid category, a response message from the server is displayed detailing all of the incorrect information the user provided. See figure 5, where all details have breached the constraints set in the spec.

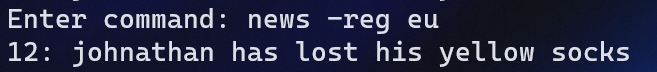
*Figure 5*

**A blue background with white text

Description automatically generatedGet news articles**

The news command fetches all articles stored on the database. The database key of the article is appending to the left of the headline. See figure 6 for an example. Switches can be added to sort by region, date or category. See figure 7 for an example.

*Figure 6*

If no news articles are found for the search criteria, then a relevant output is produced by the server and displayed client side.

*Figure 7*

*Figure 8*

**Delete story**

A blue background with white letters

Description automatically generatedStories are deleted with the “delete <key>” command. Keys can be found next to news article headlines when running the news command (e.g. 12 is the key of the article displayed in figure 7.). See figure 9.

*Figure 9*

A blue screen with white text

Description automatically generatedAttempting to delete a story that does not exist results in a basic error message being displayed (figure 10).

*Figure 10*