Group One - Milestone 5

Members:
Adam Rosa, Austin Rowen, Bing Mitchell, Claire Olmstead, Danny Niquette, Jared Gorthy
Team Name: Group One Is The Best
Project Name: Paradise News

User Acceptance Tests:

1. Login/logout
2. Likes/Dislikes
3. Search Bar

Feature 1:

We can test this feature by first creating a test user. We can then use Django to see if a this test user is logged in, since the user has not logged in yet, the is_active() attribute of test user should be False. Next we will have the test user log in and we will check the is_active() attribute again through Django and expect to see a True. Finally, we will repeat the process with logout and hope to see a False.

Feature 2:

We choose the most recent article in some given page, let's say Business. We will check our database for the number of likes and dislikes on this specific post. Next, we will log in as our test user and like this post. Then we will check the database again and see if the post attribute "likes" has been incremented. If it has, then the like functionality works. The same user will then dislike the post and we'll check the database. In this case, we expect a decrement of the "likes" attribute and an increment of the "dislikes" attribute for this specific post. Finally, we will try to dislike the post once more, making this the second time the user has disliked it. The correct function would not let the user dislike the same post twice, so we will check for this in the database.

Feature 3:

To check the search bar, we can query our database using the LIKE clause to find all posts related to a certain word. The query can order the result set by date and then run a second query and order them by number of likes. Now we go to the home page and search the same keyword. We expect that if we order by date, the search bar results will match the SQL query results. The same should be true for the order by likes result set. If there are correct, then we have passed this test. Another test case can check the details of the searching. We can create a fake article which has our keyword only in the description and not in the title. If we search for this article, it should still be part of our result set because we are searching both the

title and description of the articles. The same process can be done with a test article with the keyword only in the title and not in the description.

Automated Test Cases:

Selenium is used to automate web browser interaction, so scripts can automatically perform the same interactions that a user would perform. We can run test and see if they match the expected output. A link to the home page can be found here: https://www.seleniumhq.org

An example of the code used to run a test can be seen below:

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from time import sleep

driver = webdriver.Chrome("/Users/Dannyniquette/Documents/chromedriver")
driver.get("https://anotherdaywithparadise.herokuapp.com/accounts/signin/")
driver.switch_to_window(driver.current_window_handle)
sleep(4)
username = driver.find_element_by_xpath('//*[@id="id_username"]')
user = "Dniquette"
for i in user:
        username.send_keys(i)
        sleep(0.3)

pword = driver.find_element_by_xpath('//*[@id="id_password"]')
p = "alanparadise"
for j in p:
        pword.send_keys(j)
        sleep(.3)

submit = driver.find_element_by_xpath("/html/body/form/input[4]").click()

business = driver.find_element_by_xpath('//*[@id="home-page"]/div/a[4]').click()

like = driver.find_element_by_xpath('//*[@id="business-page"]/main[2]/article/div/a[1]').click()
```

This gives us the following output, passing all the tests, as expected!

```
Dannys-MacBook-Pro-4:softwareProject Dannyniquette$ python StackTest.py
Beginning Testing Protocol:
Opening WebBrowser
Entering Username
Entering Password
Submitting Credentials
Navigating to Business page
Checking link validity
All tests passed
```

We used these tests like these to test functionality of may parts of our website. They helped us test full functionality and were very effective!