

COSC 4P02 – Edit Test Document:**Completed Automated Tests:**

```

edit_button_test.py::test_editButtonsPresent
DevTools listening on ws://127.0.0.1:56357/devtools/browser/e2905060-29f2-4fde-ab90-e2c314cf2930
PASSED [ 20%]
edit_button_test.py::test_editButtonsFunctionality PASSED [ 40%]
edit_button_test.py::test_cancelButtonFunctionality PASSED [ 60%]
edit_button_test.py::test_editCancelMultiplePosts PASSED [ 80%]
edit_button_test.py::test_blankEdit PASSED [100%]
===== 5 passed in 17.15s =====

```

Test Execution Commands:

`python -m pytest edit_button_test.py -v`

`python -m pytest edit_button_test.py -v-s` (this will display the messages and content printed in the tests when run)

`python -m pytest edit_button_test.py -k "test_editButtonsPresent" -s -v`

`python -m pytest edit_button_test.py -k "test_editButtonsFunctionality" -s -v`

`python -m pytest edit_button_test.py -k "test_cancelButtonFunctionality" -s -v`

`python -m pytest edit_button_test.py -k "test_editCancelMultiplePosts" -s -v`

`python -m pytest edit_button_test.py -k "test_blankEdit" -s -v`

Method Test Cases and Descriptions:**test_editButtonsPresent:**

Test Case 1: Testing the presence of the edit button(s). This will confirm that the button(s) are present on the page. This method will check if the edit button(s) are present on the page. It uses the Selenium WebDriver (predefined as a fixture above) to navigate to our URL and check for the presence of the edit button(s) by their class names.

Expected Result: Pass. The edit button(s) should be present on the page.

test_editButtonsFunctionality:

Test Case 2: Testing the functionality of the edit button(s). This will confirm that the button(s) work as intended. This method will check if the edit button(s) work properly. It uses the Selenium WebDriver (predefined as a fixture above) to navigate to our URL and check that the edit button(s) function as expected when clicked. It will click the first edit button, edit the post content, and then save the changes.

Expected Result: Pass. The edit button(s) should work properly.

test_cancelButtonFunctionality:

Test Case 3: Testing the functionality of the cancel button. This will confirm that the cancel button works as intended. This method will check if the cancel button works properly. It uses the Selenium WebDriver (predefined as a fixture above) to navigate to our URL and check that the cancel button functions as expected when clicked. It will click the first edit button, edit the post content, and then cancel the changes.

Expected Result: Pass. The cancel button should work properly. The content should not be changed after canceling.

test_editCancelMultiplePosts:

Test Case 4: Testing the functionality of editing multiple posts. This will confirm that only the edited post is updated, while the other remains unchanged. This method will check if the edit button(s) work properly when editing multiple posts. It uses the Selenium WebDriver (predefined as a fixture above) to navigate to our URL and check that the edit button(s) function as expected when clicked. It will click the first edit button, edit the post content, and then save the changes. It will also click the second edit button but not make any changes to the post content.

Expected Result: Pass. The edited post should be updated, while the other post should remain unchanged.

test_blankEdit:

Test Case 5: Testing the functionality of the save button when the text area is blank. This will confirm that the button does not save when the text area is blank. This method will check if the save button works properly when the text area is blank. It uses the Selenium WebDriver (predefined as a fixture above) to navigate to our URL and check that the save button functions as expected when clicked. It will click the first edit button, clear the post content, and then save the changes.

Expected Result: Pass. The save button should not save the changes when the textarea is blank.