AUTHENTICATION-TEST

CODE:

test_authentication.py 8

C: > Users > Aayushi > Desktop > SE > test_authentication.py > TestAuthentication > test_authentication

```python
import pytest
import time
import json
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.support import expected_conditions
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.desired_capabilities import DesiredCapabilities

class TestAuthentication():
    def setup_method(self):
        self.driver = webdriver.Chrome()
        self.vars = {}

    def teardown_method(self):
        self.driver.quit()

    def test_authentication(self):
        self.driver.get("http://localhost:3000/Register")
        self.driver.set_window_size(1296, 688)

        time.sleep(2)  # Wait for 2 seconds

        self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(1)").click()
        self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(1)").send_keys("ansh")
```

```python
time.sleep(1)  # Wait for 1 second

self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(2)").click()
self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(2)").send_keys("aayushicodingworld@gmail.com")

time.sleep(1)  # Wait for 1 second

self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(3)").click()
self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(3)").send_keys("123")

time.sleep(1)  # Wait for 1 second

self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(4)").click()
self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(4)").send_keys("15")

time.sleep(1)  # Wait for 1 second

self.driver.find_element(By.CSS_SELECTOR, ".submit-button").click()

time.sleep(2)  # Wait for 2 seconds

self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(1)").click()
self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(1)").send_keys("aayushicodingworld@gmail.com")
```

```
        time.sleep(1)  # Wait for 1 second

        self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(2)").click()
        self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(2)").send_keys("123")

        time.sleep(1)  # Wait for 1 second

        self.driver.find_element(By.CSS_SELECTOR, ".submit-button").click()

test = TestAuthentication()
test.setup_method()
test.test_authentication()
test.teardown_method()
```

OUTPUT:

- This is the main test case where the authentication process is automated.
- It navigates to the registration page (http://localhost:3000/Register) and waits for 2 seconds to ensure the page is fully loaded.
- It interacts with the registration form by filling in the username, email, password, and age fields.
- After submitting the registration form, it waits for 2 seconds before proceeding to the login process.
- It fills in the email and password fields on the login page and submits the login form

NAVIGATION-TEST

CODE:

test_authentication.py 8 | test_navigation.py 8

C: > Users > Aayushi > Desktop > SE > test_navigation.py > ...

```
1  import pytest ···

11
12 class TestNavigation():
13     def setup_method(self):
14         self.driver = webdriver.Chrome()
15         self.vars = {}
16
17     def teardown_method(self):
18         self.driver.quit()
19
20     def test_navigation(s  Follow link (ctrl + click)
21         self.driver.get("http://localhost:3000/Register")
22         self.driver.set_window_size(1296, 688)
23
24         time.sleep(2)  # Wait for 2 seconds
25
26         self.driver.find_element(By.LINK_TEXT, "Login").click()
27
28         time.sleep(1)  # Wait for 1 second
29
30         self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(1)").click()
31         self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(1)").send_keys("ansh@gmail.com")
32
33         time.sleep(1)  # Wait for 1 second
34
35         self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(2)").click()
36         self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(2)").send_keys("kutti")
```

```
38         time.sleep(1)  # Wait for 1 second
39
40         self.driver.find_element(By.CSS_SELECTOR, ".submit-button").click()
41
42         time.sleep(2)  # Wait for 2 seconds
43
44         self.driver.find_element(By.LINK_TEXT, "Upload").click()
45
46         time.sleep(1)  # Wait for 1 second
47
48         self.driver.find_element(By.LINK_TEXT, "View").click()
49
50 test = TestNavigation()
51 test.setup_method()
52 test.test_navigation()
53 test.teardown_method()
54
```

OUTPUT:

- Navigation to Register Page: The test starts by navigating to the "Register" page of a web application running on localhost:3000.
- Login Link Clicked: After waiting for 2 seconds, the test clicks on a "Login" link to navigate to the login page.
- Entering Credentials: It then interacts with the login form by entering credentials (email and password) into input fields.
- Submit Button Clicked: After entering the credentials, it clicks on the submit button to attempt login.
- Navigation to Upload Page: Upon successful login (assuming no errors are encountered), the test waits for 2 seconds and then clicks on a link labeled "Upload" to navigate to another page.
- Navigation to View Page: Finally, it clicks on a link labeled "View" to navigate to yet another page.

UPLOAD IMAGE-TEST

CODE:

```python
import pytest···

class TestUploadimg:
    def setup_method(self):
        self.driver = webdriver.Chrome()

    def teardown_method(self):
        self.driver.quit()

    def test_uploadimg(self):
        self.driver.get("http://localhost:3000/Register")
        self.driver.set_window_size(1296, 688)

        # Wait for 2 seconds
        time.sleep(2)

        # Navigate to Home page
        self.driver.find_element(By.LINK_TEXT, "Home").click()

        # Wait for 1 second
        time.sleep(1)

        # Navigate to Upload page
        self.driver.find_element(By.LINK_TEXT, "Upload").click()

        # Wait for 1 second
        time.sleep(1)
```

```
# Navigate to Upload Image page
wait = WebDriverWait(self.driver, 10)
upload_image_link = wait.until(EC.element_to_be_clickable((By.LINK_TEXT, "Upload Image")))
upload_image_link.click()

# Wait for 2 seconds
time.sleep(2)

# Upload Image
image_input = self.driver.find_element(By.CSS_SELECTOR, "input:nth-child(1)")
image_input.click()
image_input.send_keys("C:\\fakepath\\IMG-20210413-WA0002 (2).jpg")

# Wait for 1 second
time.sleep(1)

# Enter Image Name
image_name_input = self.driver.find_element(By.CSS_SELECTOR, "input:nth-child(2)")
image_name_input.click()
image_name_input.send_keys("demoimg")

# Wait for 1 second
time.sleep(1)
```

```
        # Submit Image
        submit_button = self.driver.find_element(By.CSS_SELECTOR, "button:nth-child(3)")
        submit_button.click()

        # Wait for 5 seconds for the alert
        time.sleep(5)

        try:
            alert = self.driver.switch_to.alert
            assert alert.text == "Image Uploaded Successfully"
            alert.accept()
        except:
            assert False, "Alert not found or different alert text."

        # Wait for 1 second
        time.sleep(1)

        # Navigate back to Upload page
        self.driver.find_element(By.LINK_TEXT, "Upload").click()

# Execute test
test_uploadimg = TestUploadimg()
test_uploadimg.setup_method()
test_uploadimg.test_uploadimg()
test_uploadimg.teardown_method()
```

OUTPUT:

- It navigates to the "Register" page of a web application running on localhost:3000.
- After waiting for 2 seconds, it navigates to the "Home" page.
- Then, it navigates to the "Upload" page.
- After another 1-second wait, it clicks on the "Upload Image" link.
- It waits for up to 10 seconds for the "Upload Image" link to become clickable.
- It uploads an image file by sending the file path to the file input element.
- It enters an image name into an input field.
- It submits the form by clicking on the submit button.
- It waits for 5 seconds for an alert to appear.
- It verifies that the alert text is "Image Uploaded Successfully" and accepts the alert.
- It then navigates back to the "Upload" page.

UPLOAD PDF-TEST

CODE:

```python
import pytest ···

class TestUploadpdf():
    def setup_method(self):
        self.driver = webdriver.Chrome()
        self.vars = {}

    def teardown_method(self):
        self.driver.quit()

    def test_uploadpdf(self):
        self.driver.get("http://localhost:3000/Register")
        self.driver.set_window_size(1296, 688)

        time.sleep(2)  # Wait for 2 seconds

        self.driver.find_element(By.LINK_TEXT, "Home").click()

        time.sleep(1)  # Wait for 1 second

        self.driver.find_element(By.LINK_TEXT, "Upload").click()

        time.sleep(1)  # Wait for 1 second

        self.driver.find_element(By.LINK_TEXT, "Upload pdf").click()

        time.sleep(1)  # Wait for 1 second
```

```python
        self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(2)").click()
        self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(2)").send_keys("C:\\fakepath\\soln_from_text_midterm.pd

        time.sleep(1)  # Wait for 1 second

        self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(1)").click()
        self.driver.find_element(By.CSS_SELECTOR, ".input-field:nth-child(1)").send_keys("pdf1")

        time.sleep(1)  # Wait for 1 second

        self.driver.find_element(By.CSS_SELECTOR, ".submit-button").click()

        time.sleep(2)  # Wait for 2 seconds

        assert self.driver.switch_to.alert.text == "Pdf Uploaded Successfully"

test = TestUploadpdf()
test.setup_method()
test.test_uploadpdf()
test.teardown_method()
```

OUTPUT:

- Navigation to Register Page: The test starts by navigating to the "Register" page of a web application running on localhost:3000.
- Navigation to Home Page: After waiting for 2 seconds, the test navigates to the "Home" page by clicking on a link labeled "Home".
- Navigation to Upload Page: Then, it clicks on a link labeled "Upload" to navigate to the "Upload" page.
- Navigation to Upload PDF Page: After that, it clicks on a link labeled "Upload PDF" to navigate to the page where the PDF can be uploaded.
- PDF Upload: The test uploads a PDF file by locating the file input element, clicking on it, and sending the file path of the PDF to be uploaded. It also enters a name for the PDF into an input field.
- Submission: After entering the PDF and name, it clicks on a submit button to initiate the upload process.
- Alert Handling: The test checks if an alert with the text "Pdf Uploaded Successfully" appears. If the alert is found and contains the expected text, the test passes.

VIEW IMAGE-TEST

CODE:

```
import pytest···

class TestViewimg():
    def setup_method(self):
        self.driver = webdriver.Chrome()
        self.vars = {}

    def teardown_method(self):
        self.driver.quit()

    def wait_for_window(self, timeout=2):
        time.sleep(round(timeout / 1000, 2))
        wh_now = self.driver.window_handles
        wh_then = self.vars["window_handles"]
        if len(wh_now) > len(wh_then):
            return set(wh_now).difference(set(wh_then)).pop()

    def test_viewimg(self):
        self.driver.get("http://localhost:3000/Register")
        self.driver.set_window_size(1296, 688)

        time.sleep(2)  # Wait for 2 seconds

        self.driver.find_element(By.LINK_TEXT, "Home").click()

        time.sleep(1)  # Wait for 1 second

        self.driver.find_element(By.LINK_TEXT, "View").click()
```

```
        time.sleep(1)  # Wait for 1 second

        self.driver.find_element(By.LINK_TEXT, "View Image").click()

        time.sleep(1)  # Wait for 1 second

        self.driver.execute_script("window.scrollTo(0,0)")

        self.vars["window_handles"] = self.driver.window_handles

        time.sleep(1)  # Wait for 1 second

        self.driver.find_element(By.CSS_SELECTOR, ".image-item:nth-child(9) > .view-image-btn").click()

        time.sleep(2)  # Wait for 2 seconds for new window to open

        self.vars["win922"] = self.wait_for_window(2000)
        self.driver.switch_to.window(self.vars["win922"])

test = TestViewimg()
test.setup_method()
test.test_viewimg()
test.teardown_method()
```

OUTPUT:

- Navigation to Register Page: The test starts by navigating to the "Register" page of a web application running on localhost:3000.
- Navigation to Home Page: After waiting for 2 seconds, the test navigates to the "Home" page by clicking on a link labeled "Home".
- Navigation to View Page: Then, it clicks on a link labeled "View" to navigate to the "View" page.
- Navigation to View Image Page: After that, it clicks on a link labeled "View Image" to navigate to the page where an image can be viewed.
- Scroll to Top: The test executes JavaScript to scroll the window to the top of the page.
- Handling New Window: It waits for the new window to open and switches to it using the window handles.

VIEW PDF-TEST

CODE:

```
> import pytest ···

class TestViewpdf():
    def setup_method(self):
        self.driver = webdriver.Chrome()
        self.vars = {}

    def teardown_method(self):
        self.driver.quit()

    def wait_for_window(self, timeout=2):
        time.sleep(round(timeout / 1000, 2))
        wh_now = self.driver.window_handles
        wh_then = self.vars["window_handles"]
        if len(wh_now) > len(wh_then):
            return set(wh_now).difference(set(wh_then)).pop()

    def test_viewpdf(self):
        self.driver.get("http://localhost:3000/Register")
        self.driver.set_window_size(1296, 688)

        # Navigate to Home page
        self.driver.find_element(By.LINK_TEXT, "Home").click()
        time.sleep(2)  # Wait for 2 seconds

        # Navigate to View page
        self.driver.find_element(By.LINK_TEXT, "View").click()
        time.sleep(2)  # Wait for 2 seconds
```

```
        # Navigate to View pdf page
        self.driver.find_element(By.LINK_TEXT, "View pdf").click()
        time.sleep(2)  # Wait for 2 seconds

        # Store current window handles
        self.vars["window_handles"] = self.driver.window_handles

        # Click on view pdf button
        view_pdf_btn = self.driver.find_element(By.CSS_SELECTOR, ".image-item:nth-child(9) > .view-image-btn")
        view_pdf_btn.click()

        # Wait for 2 seconds
        time.sleep(2)

        # Switch to new window
        self.vars["win3345"] = self.wait_for_window(2000)
        self.driver.switch_to.window(self.vars["win3345"])

# Execute test
test_viewpdf = TestViewpdf()
test_viewpdf.setup_method()
test_viewpdf.test_viewpdf()
test_viewpdf.teardown_method()
```

OUTPUT:

- Navigation to Register Page: The test starts by navigating to the "Register" page of a web application running on localhost:3000.
- Navigation to Home Page: After waiting for 2 seconds, the test navigates to the "Home" page by clicking on a link labeled "Home".
- Navigation to View Page: Then, it clicks on a link labeled "View" to navigate to the "View" page.
- Navigation to View PDF Page: After that, it clicks on a link labeled "View pdf" to navigate to the page where a PDF file can be viewed.
- Handling New Window: The test stores the current window handles and waits for a new window to open after clicking on the "view pdf" button. Once the new window opens, it switches to it using the window handles.