Selenium Webdriver test case automation

Author:
Robert Jaszewski

I Test case description.

Test case: Placing order with valid information
ID: 004.
Title: Placing order with valid information
Environment: Chrome version 62.0.3202.94, Ubuntu 20-04 LTS.

Test data:
email = „automationtester223@gmail.com”
lastName = „Smith”
firstName = „John”
password = „Tester77”
quantity = ‘2’

Preconditions:
1. Browser is open
2. Main page is open
3. User is not logged in
4. Default address of the chosen account is set up

Steps:
1. Click on „Sign In”
2. Enter email
3. Enter password
4. Click on „Sign in”
5. Choose Jackets from „Men” dropdown menu
6. Click on „Lando Gym Jacket”
7. Click on „L” size
8. Select green color
9. Enter quantitty in QTY field
10. Click on „Add to cart” button
11. Click on cart icon in the right upper corner
12. Click on „Proceed to Checkout” button
13. Click on „Table Rate” shipping method
14. Click on „Next” button
15. Click on „Place Order” button

Expected results:
1. The user receives the information: "There is 1 error 1. You must register at least one phone number."
2. The user receives the information: "Your order number is: "
3. The user receives the information: "We’ll email you an order confirmatuon with details and tracking info”

Final conditions:
1. Order is completed
2. All required information is displayed

II. Selenium webdriver test case automation

```
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains
from time import sleep

#TEST DATA
email = "automationtester223@gmail.com"
lastName = "Smith"
firstName = "John"
password = "Tester77"
quantity = '2'

class PlacingOrder(unittest.TestCase):
    def setUp(self):
        # PRECONDITIONS
        # 1. Browser is open
        self.driver = webdriver.Chrome()
        self.driver.maximize_window()
        # 2. Main page is open
        self.driver.get('https://magento.softwaretestingboard.com/')
        # 3. User is not logged in
        self.driver.implicitly_wait(5)
        # 4. Default address of the chosen account is set up
    def testPLacingOrderWithValidInformation(self):
        # STEPS
        # 1. Click on "Sign In"
        self.driver.find_element(By.LINK_TEXT, 'Sign In').click()
        # 2. Enter Email
        self.driver.find_element(By.ID, 'email').send_keys(email)
        # 3. Enter Password
        self.driver.find_element(By.ID, 'pass').send_keys(password)
        # 4. Click on "Sign In"
        self.driver.find_element(By.ID, 'send2').click()
        # 5. Choose Jackets from "Men" dropdown menu
        action = ActionChains(self.driver)
        action.move_to_element(self.driver.find_element(By.ID, 'ui-id-5'))
        action.move_to_element(self.driver.find_element(By.ID, 'ui-id-17'))
        action.move_to_element(self.driver.find_element(By.ID, 'ui-id-19')).click()
        action.perform()
        # 6. Click on "Lando Gym Jacket"
        self.driver.find_element(By.LINK_TEXT, 'Lando Gym Jacket').click()
        # 7. Click on "L" size
        self.driver.find_element(By.CSS_SELECTOR, 'div[option-label="L"]').click()
        # 8. Select green color
        self.driver.find_element(By.CSS_SELECTOR, 'div[option-label="Green"]').click()
        # 9. Enter quantity in QTY field.
        self.driver.find_element(By.ID, 'qty').clear()
        self.driver.find_element(By.ID, 'qty').send_keys(quantity)
        # 10. Click on "Add to Cart" button
        self.driver.find_element(By.ID, 'product-addtocart-button').click()
        sleep(3)
        # 11.Click on cart icon in the right upper corner
        self.driver.find_element(By.CSS_SELECTOR, 'div[data-block="minicart"]').click()
        # 12.Click on "Proceed to Checkout" button
        self.driver.find_element(By.ID, 'top-cart-btn-checkout').click()
        sleep(1)
        # 13. Click on "Table Rate" shipping method
        self.driver.find_element(By.CSS_SELECTOR, 'input[value="tablerate_bestway"]').click()
        # 14. Click on "Next" button
```

```
        self.driver.find_element(By.CSS_SELECTOR, 'button[data-role="opc-continue"]').click()
        sleep(3)
        # 15. Click on "Place Order" button
        self.driver.find_element(By.CSS_SELECTOR, 'button[title="Place Order"]').click()
        sleep(3)

        # EXPECTED RESULTS
        # 1. The user receives the information: "Thank you for your purchase!"
        # Looking for text "Thank you for your purchase!"
        infoPurchase1 = self.driver.find_element(By.CSS_SELECTOR, 'span[data-ui-id="page-title-
wrapper"]')
        self.assertTrue(infoPurchase1.is_displayed())
        print("Assertion 1 is correct. 'Thank you for your purchase' text is displayed")
        # 2. The user receives the information: "Your order number is:"
        # Looking for text "Your order number is"
        infoPurchase2 = self.driver.find_element(By.XPATH, '//p[contains( text(), "Your order
number is: ")]')
        orderNumber = self.driver.find_element(By.XPATH,
'//a[@class="order-number"]//strong').text
        print("Assertion 2 is correct. 'Your order number is: {}' text is
displayed".format(orderNumber))
        # 3. The user receives the information: "We'll email you an order confirmation with details
and tracking info."
        # Looking for text "We'll email you an order confirmation with details and tracking info."
        infoPurchase3 = self.driver.find_element(By.XPATH, '//p[contains( text(), "We\'ll email you
an order confirmation with details and tracking info")]')
        self.assertTrue(infoPurchase2.is_displayed())
        print("Assertion 3 is correct. 'We\'ll email you an order confirmation with details and
tracking info.' text is displayed")

    def tearDown(self):
        # End of test
        self.driver.quit()
        # FINAL CONDITIONS
        # 1. Order is completed
        # 2. All required information is displayed
```

```
Process finished with exit code 0
Assertion 1 is correct. 'Thank you for your purchase' text is displayed
Assertion 2 is correct. 'Your order number is: 000017674' text is displayed
Assertion 3 is correct. 'We'll email you an order confirmation with details and tracking info.' text is displayed


Ran 1 test in 21.503s

OK
```

III. Final remarks:

Test case automation (functional test) successful. The test may be sensitive to changing the structure of the website due to the need to use long path in XPATH and CSS locators. For this reason, it is suggested to periodically monitor the course of the test and update the script if it is necessary.