

AUTOMATYZACJA TESTÓW
REJESTRACJI, LOGOWANIA ORAZ
SPRAWDZENIA KOSZYKA NA STRONIE
<https://automationpractice.com/>

Cel testów: Celem testów było przeprowadzenie próby logowania, następnie wylogowania i ponownego zalogowania oraz sprawdzenia, czy po wejściu w koszyk pozostaje on pusty i nie ma możliwości zrobienia zakupów.

Oprogramowanie wymagane do wykonania testów: Testy zostały przeprowadzone przy wykorzystaniu przeglądarki Mozilla Firefox wersja 98.0.2 (64-bit) działającej na systemie Ubuntu 20.04.3 LTS, wersja 64-bit uruchomionym przy pomocy Oracle VirtualBox Wersja 6.1.30 r148432 (Qt5.6.2).

Inne wykorzystane oprogramowanie: Testy zostały napisane przy wykorzystaniu selenium, PyCharm 2022.1.1 (Community Edition), gecodriver oraz faker.

Kod:

W pierwszym etapie konieczny jest import bibliotek wymaganych do przeprowadzenia testów:

```
#import bibliotek
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep
from selenium.webdriver.support.select import Select
from faker import Faker
```

Następnie przy wykorzystaniu Faker'a wygenerowano część danych koniecznych do przeprowadzenia testu rejestracji:

```
# DANE TESTOWE
```

```
fake = Faker()
lastname = fake.last_name_male()
firstname = fake.first_name_male()
nick = fake.user_name()
email = fake.email()
password = fake.password()
```

```
day = str(15)
month = str(3)
year = str(fake.year())
address = "Testerkowa 32 Bydgoszcz"
city = "Bydgoszcz"
postcode = "41026"
phone = str(252689231)
alias = "my alias"
```

Następnie wprowadzono kod, który miał na celu uruchomienie przeglądarki na zmaksymalizowanym oknie i przejście na stronę testową:

#Test rejestracji, logowania i zakupu przy pustym koszyku.

```
class RegistrationTest(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Firefox()
        self.driver.maximize_window()
        self.driver.get("https://automationpractice.com/")
        self.driver.implicitly_wait(10)
```

Dalej zdefiniowano 22 kroki umożliwiające zarejestrowanie oraz wylogowanie użytkownika (1-18), sprawdzenie poprawności logowania (19-20) oraz sprawdzenie czy bez dodawania produktów koszyk jest pusty (21-22).

```
def testRegistration(self):
    # KROKI
    driver = self.driver
    # 1. Kliknij "Sign in"
    driver.find_element(By.CLASS_NAME, "login").click()
    # Wpisz e-mail
    driver.find_element(By.ID, 'email_create').send_keys(email)
    # 3. Kliknij przycisk „Create account”
```

```
driver.find_element(By.ID, "SubmitCreate").click()
```

4. Wybierz płeć

```
driver.find_element(By.ID, "id_gender1").click()
```

5. Wpisz imię

```
driver.find_element(By.ID, "customer_firstname").send_keys(firstname)
```

5. Wpisz nazwisko

```
driver.find_element(By.ID, "customer_lastname").send_keys(lastname)
```

6. Wpisz hasło

```
driver.find_element(By.ID, "passwd").send_keys(password)
```

8. Wybierz datę urodzenia

#dzien

```
Select(driver.find_element(By.ID, "days")).select_by_value(day)
```

#miesiac

```
Select(driver.find_element(By.ID, "months")).select_by_value(month)
```

#rok

```
Select(driver.find_element(By.ID, "years")).select_by_value(year)
```

11. Wpisz adres

```
driver.find_element(By.ID, 'address1').send_keys(address)
```

12. Wpisz miasto

```
driver.find_element(By.ID, 'city').send_keys(city)
```

13. Wpisz kod pocztowy

```
driver.find_element(By.ID, 'postcode').send_keys(postcode)
```

14. Wybierz stan

```
Select(driver.find_element(By.ID, "id_state")).select_by_visible_text("Colorado")
```

15. Wpisz numer telefonu

```
driver.find_element(By.ID, 'phone_mobile').send_keys(phone)
```

16. Wpisz alias adresu

```
driver.find_element(By.ID, 'alias').send_keys(alias)
```

17. Kliknij Register

```
driver.find_element(By.ID, 'submitAccount').click()
```

18. Wyloguj

```

driver.find_element(By.CLASS_NAME, 'logout').click()

# 19. Ponownie kliknij zaloguj
driver.find_element(By.CLASS_NAME, "login").click()

# 20. Poowne logowanie

#e-mail
driver.find_element(By.ID, 'email').send_keys(email)

#haslo
driver.find_element(By.ID, "passwd").send_keys(password)

#kliknij sign in
driver.find_element(By.ID, 'SubmitLogin').click()

# 21. powroc na strone glowna
driver.find_element(By.XPATH, "/html/body/div/div[2]/div/div[1]/a/i").click()

# 22. Wybierz zakladke "koszyk"
driver.find_element(By.XPATH,
"/html/body/div/div[1]/header/div[3]/div/div/div[3]/div/a").click()

#Sprawdzanie powodzenia misji
komunikat = driver.find_element(By.XPATH, "/html/body/div/div[2]/div/div[3]/div/p").text
self.assertEqual("Your shopping cart is empty.", komunikat)

sleep(10)

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

```

Ostatnie linijki kodu są odpowiedzialne za sprawdzenie faktycznego stanu koszyka oraz wyłączenie przeglądarki.

Testy zostały wykonane kilkakrotnie i za każdym razem przyniosły pozytywny rezultat.

Wykonał: Damian Podgrudny