PROJEKT ZALICZENIOWY
SELENIUM WEBDRIVER

Aleksandra Kacperczyk

Tester oprogramowania aplikacji mobilnych i serwerowych

2018/2019

Grupa 1

Home page
https://www.phptravels.net

Signup page
https://www.phptravels.net/register

Login page
https://www.phptravels.net/login

Account page
https://www.phptravels.net/account

Tours page
https://www.phptravels.net/tours

Struktura projektu:

projekt zaliczeniowy/

- requirements.txt
- README.md
- wrong_email.csv

- pages/

  - base_page.py
  - home_page.py
  - login_page.py
  - signup_page.py
  - account_page.py
  - tours_page.py

- locators/

  - home_page_locators.py
  - login_page_locators.py
  - sigup_page_locators.py
  - account_page_locators.py

- tests/

  - test_login.py
  - test_signup.py
  - test_account.py
  - test_tours.py

Przypadki testowe

Plik: test_login.py

ID: 001
Tytuł: Logowanie zarejestrowanego użytkownika poprawnymi danymi do logowania
Środowisko: macOS Mojave 10.14.4, Google Chrome 74.0.3729.108
Warunki wstępne: Użytkownik nie jest zalogowany. Przeglądarka otwarta na Home page (**https://www.phptravels.net/**)
Dane wejściowe:

Dane do logowania:
Email: user@phptravels.com
Hasło: demouser

Kroki:

1. Kliknij "**MY ACCOUNT**".
2. Wybierz opcję "**Login**" z listy.
3. Wprowadź poprawny adres e-mail.
4. Wprowadź poprawne hasło.
5. Naciśnij '**LOGIN**'.

Oczekiwany rezultat: Użytkownik został poprawnie zalogowany. Następuje automatyczne przekierowanie na stronę **https://www.phptravels.net/account/.**

ID: 002
Tytuł: Logowanie zarejestrowanego użytkownika z użyciem niepoprawnego hasła
Środowisko: macOS Mojave 10.14.4, Google Chrome 74.0.3729.108
Warunki wstępne: Użytkownik nie jest zalogowany. Przeglądarka otwarta na Home page (**https://www.phptravels.net/**)
Dane wejściowe:

Dane do logowania:
Email: user@phptravels.com
Hasło: demouser123

Kroki:

1. Kliknij "MY ACCOUNT".
2. Wybierz opcję "Login" z listy.
3. Wprowadź poprawny adres e-mail.
4. Wprowadź poprawne hasło.
5. Naciśnij '**LOGIN**'.

Oczekiwany rezultat: Użytkownik nie został zalogowany. Wyświetla się komunikat: "**Invalid Email or Password**'.

Plik: test_signup.py

ID: 003
Tytuł: Rejestracja nowego użytkownika z użyciem niepoprawnego adresu email
Środowisko: macOS Mojave 10.14.4, Google Chrome 74.0.3729.108
Warunki wstępne: Użytkownik nie ma konta na portalu **phptravels.net**. Przeglądarka otwarta na Home page ((**https://www.phptravels.net/**)
Dane wejściowe: Dane z pliku wrong_email.csv
Kroki:

1. Kliknij 'MY ACCOUNT'
2. Wybierz opcję 'Sign up' z listy.
   (te dwa kroki robią się za jednym razem)
3. Kliknij przycisk 'SIGN UP'
4. Wprowadź imię (*First Name*)
5. Wprowadź nazwisko (*Last Name*)
6. Wprowadź numer komórkowy (*Mobile Number*)
7. Wprowadź niepoprawny adres email
8. Wprowadź hasło (*Password*)
9. Wprowadź powtórnie hasło (*Confirm Password*)

Oczekiwany rezultat: Konto nie zostaje utworzone. Wyświetlany jest komunikat: "The Email field must contain a valid email address."

Plik: test_account.py

ID: 004
Tytuł: Zmiana hasła na Account Page w zakładce My Profile
Środowisko: macOS Mojave 10.14.4, Google Chrome 74.0.3729.108
Warunki wstępne: Użytkownik ma konto na portalu **phptravels.net**. Użytkownik nie jest zalogowany. Przeglądarka otwarta na Home page (**https://www.phptravels.net**)
Dane wejściowe:

Dane do logowania:
Email: user@phptravels.com
Hasło: demouser

Nowe hasło: demouser123

Kroki:

1. Zaloguj się (Test case 001)
2. Kliknij "My Profile".
3. Wpisz nowe hasło.
4. Potwierdź nowe hasło wpisując je ponownie.
5. Kliknij "SUBMIT".

Oczekiwany rezultat: Wyświetla się komunikat "Profile Updated Successfully".

Plik: test_tours.py

ID: 005
Tytuł: Wyszukanie wycieczki bez wybrania lokalizacji
Środowisko: macOS Mojave 10.14.4, Google Chrome 74.0.3729.108
Warunki wstępne: Użytkownik nie jest zalogowany. Przeglądarka otwarta na Home page ((**https://www.phptravels.net/**)
Dane wejściowe:

Miasto: London
Data odlotu: 12/06/2019
Liczba osób: 3
Typ wycieczki: Family

Kroki:

1. Zmień tryb wyszukiwania na loty klikając ‘TOURS”.
2. Wybierz datę lotu z kalendarza.
3. Naciśnij na cyfrę 1 w polu wyboru liczby osób
4. Wybierz liczbę osób.
5. Wybierz typ wycieczki.
6. Naciśnij ‘SEARCH’.

Oczekiwany rezultat: Użytkownik zostaje przeniesiony na stronę z wynikami wyszukiwania. Wyświetlony jest komunikat “No Results Found”.

Kod

account_page_locators.py

```
from selenium.webdriver.common.by import By
class AccoutPageLocators():

    MY_PROFILE = (By.PARTIAL_LINK_TEXT, 'My Profile')

    NEW_PASSWORD = (By.NAME, "password")

    CONFIRM_NEW_PASSWORD = (By.NAME, 'confirmpassword')

    SUBMIT_BUTTON = (By.XPATH, '//button[@class="btn btn-action btn-block
updateprofile"]')

    COOKIE_BUTTON = (By.ID, 'cookyGotItBtn')
```

home_page_locators.py

```
from selenium.webdriver.common.by import By

class HomePageLocators():
    MY_ACCOUNT_BTN = (By.XPATH, '//a[@href="javascript:void(0);"]')
    LOGIN_BTN = (By.XPATH, '//a[@href="https://www.phptravels.net/login"]')

    FLIGHTS_BTN = (By.XPATH, '//span[text()="Flights   "]')

    TOURS_BTN = (By.XPATH, '//span[text()="Tours     "]')

    CITY_BTN = (By.XPATH, '//input[@class="hotelsearch locationlisttours"]')

    DATE_BTN = (By.NAME, 'date')

    DATE_ARROW = (By.XPATH, '/html/body/div[12]/div[1]/table/thead/tr[1]/th[3]')

    JUNE_12 = (By.XPATH, '/html/body/div[12]/div[1]/table/tbody/tr[3]/td[4]')

    ADULTS = (By.XPATH, '//select[@name="adults"]')

    ADULTS_NUM = (By.XPATH, '//option[@value="3"]')

    TOUR_TYPE = (By.ID, 'tourtype')

    TYPE_FAMILY = (By.XPATH, '//option[@value="196"]')

    SEARCH_BTN = (By.XPATH, '//*[@id="tours"]/form/div[5]/button')
```

login_page_locators.py

```
from selenium.webdriver.common.by import By

class LoginPageLocators():

    EMAIL_FIELD = (By.XPATH, '//input[@placeholder="Email"]')

    PASSWORD_FIELD = (By.NAME, 'password')

    LOGIN_BUTTON = (By.XPATH, '//button[text()="Login"]')

    SIGN_UP_BTN = (By.XPATH, '//a[@href="https://www.phptravels.net/register"]')
```

signup_page_locators.py

```
from selenium.webdriver.common.by import By

class SignupPageLocators():

    FIRST_NAME_FIELD = (By.NAME, 'firstname')
    #//input[@name="firstname"]

    LAST_NAME_FIELD = (By.NAME, 'lastname')

    MOBILE_NUMBER_FIELD = (By.XPATH, '//input[@placeholder="Mobile Number"]')

    EMAIL_FIELD = (By.XPATH, '//input[@placeholder="Email"]')

    PASSWORD_FIELD = (By.XPATH, '//input[@placeholder="Password"]')

    CONFIRM_PASSWORD_FIELD = (By.XPATH, '//input[@name="confirmpassword"]')

    SIGNUP_BUTTON = (By.XPATH, '//button[text()=" Sign Up"]')
```

account_page.py

```
from pages.base_page import BasePage
from locators.account_page_locators import AccoutPageLocators
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.action_chains import ActionChains
import time


class AccountPage(BasePage):

    def _verify_page(self):
        assert "My Account" in self.driver.title

    def click_my_profile(self):
        self.driver.execute_script("""document.getElementsByTagName("a")
[62].click();""")
        WebDriverWait(self.driver,
10).until(EC.presence_of_element_located((By.NAME, 'password')))

    def enter_new_password(self, password):
        el = self.driver.find_element(*AccoutPageLocators.NEW_PASSWORD)
        el.click()
        el.send_keys(password)

    def confirm_new_password(self, confirmed_password):
        el = self.driver.find_element(*AccoutPageLocators.CONFIRM_NEW_PASSWORD)
        el.click()
        el.send_keys(confirmed_password)

    def click_submit(self):

        cookie_btn = self.driver.find_element(*AccoutPageLocators.COOKIE_BUTTON)
        cookie_btn.click()
        #time.sleep(3)
        el = self.driver.find_element(*AccoutPageLocators.SUBMIT_BUTTON)
        actions = ActionChains(self.driver)
        actions.move_to_element(el)
        actions.perform()
        el.click()
        #self.driver.implicitly_wait(3)
```

```
        WebDriverWait(self.driver,
10).until(EC.presence_of_element_located((By.XPATH, '//div[@class="alert alert-
success"]')))
```

base_page.py

```
class BasePage():

    def __init__(self, driver):
        self.driver = driver
        self._verify_page()

    def _verify_page(self):
        return
```

home_page.py

```
from pages.base_page import BasePage
from locators.home_page_locators import HomePageLocators
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import time

class HomePage(BasePage):

    def _verify_page(self):
        assert "PHPTRAVELS | Travel Technology Partner" in self.driver.title
        '''
        WebDriverWait(self.driver, 15).until(
            EC.presence_of_element_located(HomePageLocators.MY_ACCOUNT_BTN)
        )
        WebDriverWait(self.driver, 15).until(
            EC.element_to_be_clickable(HomePageLocators.MY_ACCOUNT_BTN)
        )
        '''


    def click_my_account_btn(self):
        #el = self.driver.find_element(*HomePageLocators.MY_ACCOUNT_BTN)
        #el.click()
        self.driver.execute_script("""document.getElementsByTagName("a")
[3].click();""")
        time.sleep(5)

    def click_login(self):
        el = self.driver.find_element(*HomePageLocators.LOGIN_BTN)
        el.click()

    def click_flights(self):
        self.driver.find_element(*HomePageLocators.FLIGHTS_BTN).click()


    def click_tours(self):
        self.driver.find_element(*HomePageLocators.TOURS_BTN).click()
        #WebDriverWait(self.driver,
15).until(EC.presence_of_element_located((By.XPATH, '//input[@class="hotelsearch
locationlisttours"]')))
        self.driver.implicitly_wait(5)

    def enter_date(self, date):
        self.driver.find_element(*HomePageLocators.DATE_BTN).send_keys(date)

    def select_date(self):
```

```
        self.driver.find_element(*HomePageLocators.DATE_BTN).click()
        self.driver.find_element(*HomePageLocators.DATE_ARROW).click()
        self.driver.find_element(*HomePageLocators.JUNE_12).click()


    def select_adults_number(self):
        self.driver.find_element(*HomePageLocators.ADULTS).click()
        self.driver.find_element(*HomePageLocators.ADULTS_NUM).click()

    def select_tour_type(self):
        self.driver.find_element(*HomePageLocators.TOUR_TYPE).click()
        self.driver.find_element(*HomePageLocators.TYPE_FAMILY).click()


    def click_search(self):
        self.driver.find_element(*HomePageLocators.SEARCH_BTN).click()
```

login_page.py

```
from pages.base_page import BasePage
from locators.login_page_locators import LoginPageLocators
import time
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

class LoginPage(BasePage):

    def _verify_page(self):
        assert "Login" in self.driver.title

    def submit_email(self, email):
        el = self.driver.find_element(*LoginPageLocators.EMAIL_FIELD)
        el.click()
        el.send_keys(email)

    def submit_password(self, password):
        el = self.driver.find_element(*LoginPageLocators.PASSWORD_FIELD)
        el.send_keys(password)

    def click_login_btn(self):
        self.driver.find_element(*LoginPageLocators.LOGIN_BUTTON).click()
        time.sleep(5)


    def click_sign_up_btn(self):
        self.driver.execute_script("""document.getElementsByTagName("a")
[61].click();""")
        WebDriverWait(self.driver, 15).until(EC.title_is('Register'))
```

signup_page.py

```
from pages.base_page import BasePage
import time
from locators.signup_page_locators import SignupPageLocators
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
```

```
class SignupPage(BasePage):

    def _verify_page(self):
        assert "Register" in self.driver.title

    def submit_first_name(self, first_name):
        el = self.driver.find_element(*SignupPageLocators.FIRST_NAME_FIELD)
        el.click()
        el.send_keys(first_name)

    def submit_last_name(self, last_name):
        el = self.driver.find_element(*SignupPageLocators.LAST_NAME_FIELD)
        el.click()
        el.send_keys(last_name)

    def submit_mobile_number(self, mobile_number):
        el = self.driver.find_element(*SignupPageLocators.MOBILE_NUMBER_FIELD)
        el.click()
        el.send_keys(mobile_number)

    def submit_email(self, email):
        el = self.driver.find_element(*SignupPageLocators.EMAIL_FIELD)
        el.click()
        el.send_keys(email)

    def submit_password(self, password):
        el = self.driver.find_element(*SignupPageLocators.PASSWORD_FIELD)
        el.click
        el.send_keys(password)

    def confirm_password(self, password_confirmed):
        el =
self.driver.find_element(*SignupPageLocators.CONFIRM_PASSWORD_FIELD)
        el.send_keys(password_confirmed)

    def confirm_signup(self):
        self.driver.find_element(*SignupPageLocators.SIGNUP_BUTTON).click()
        #WebDriverWait(self.driver, 15).until(EC.visibility_of('//
div[@class="alert alert-danger"]'))
        self.driver.implicitly_wait(5)
```

tours_page.py

```
from pages.base_page import BasePage

class ToursPage(BasePage):

    def _verify_page(self):
        assert "Search Results" in self.driver.title
```

test_account.py

```
import unittest
from selenium import webdriver
import time
from pages.home_page import HomePage
from pages.login_page import LoginPage
from pages.account_page import AccountPage

class TravelAccount(unittest.TestCase):
```

```
    def setUp(self):

        # Warunki wstępne: Przeglądarka otwarta na stronie https://
www.phptravels.net/account

        self.driver = webdriver.Chrome()
        self.driver.maximize_window()
        self.driver.get("https://www.phptravels.net")

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

    def test_password_change(self):

        # 1. Logowanie
        home_page = HomePage(self.driver)
        home_page.click_my_account_btn()

        # 2. Wprowadź poprawny adres email
        login_page = LoginPage(self.driver)
        login_page.submit_email('user@phptravels.com')

        # 3. Wprowadź poprawne hasło
        login_page.submit_password('demouser')

        # 4. Kliknij 'LOGIN'
        login_page.click_login_btn()

        account_page = AccountPage(self.driver)

        # 5. Kliknij 'My Profile'
        account_page.click_my_profile()

        # 6. Wprowadz nowe haslo
        account_page.enter_new_password('demouser123')

        # 7. Wprowadz powtornie nowe haslo
        account_page.confirm_new_password('demouser123')

        # 8. Kliknij SUBMIT
        account_page.click_submit()

        # Oczekiwany rezultat: Wyświetla się komunikat "Profile Updated
Successfully"
        success_notices = self.driver.find_elements_by_xpath('//
div[@class="alert alert-success"]')
        visible_success_notices = []
        for success in success_notices:
            if success.is_displayed():
                visible_success_notices.append(success)
        self.assertEqual(len(visible_success_notices), 1)
        self.assertEqual(visible_success_notices[0].text, "Profile Updated
Successfully.")
```

test_login.py

```
import unittest
from selenium import webdriver
import time
from pages.home_page import HomePage
from pages.login_page import LoginPage
from pages.account_page import AccountPage


class TravelLogin(unittest.TestCase):
```

```
    def setUp(self):

        # Warunki wstępne: Przeglądarka otwarta na stronie https://
www.phptravels.net/

        self.driver = webdriver.Chrome()
        self.driver.maximize_window()
        self.driver.get("https://www.phptravels.net/")

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

    def test_login_correct_credents(self):

        # Scenariusz: Logowanie zarejestrowanego użytkownika poprawnymi danymi
do logowania

        # 1. Kliknij "MY ACCOUNT"
        # ta metoda klika My Account i Login za jednym razem
        home_page = HomePage(self.driver)
        home_page.click_my_account_btn()

        '''
        # 2. Kliknij "Login"
        main_page.click_login()
        time.sleep(3)
        '''
        # 3. Wprowadź poprawny adres email
        login_page = LoginPage(self.driver)
        login_page.submit_email('user@phptravels.com')

        # 4. Wprowadź poprawne hasło
        login_page.submit_password('demouser')

        # 5. Kliknij 'LOGIN'
        login_page.click_login_btn()

        #Oczekiwany rezulat: Użytkownik jest poprawnie zalogowany. Następuje
automatyczne przeniesienie na stronę https://www.phptravels.net/account/

        account_page = AccountPage(self.driver)

    def test_login_wrong_password(self):
        # Scenariusz: Logowanie zarejestrowanego użytkownika z użyciem
niepoprawnego hasła

        # 1 & 2. Kliknij "MY ACCOUNT"
        main_page = HomePage(self.driver)
        main_page.click_my_account_btn()

        # 3. Wprowadź poprawny adres email
        login_page = LoginPage(self.driver)
        login_page.submit_email('user@phptravels.com')

        # 4. Wprowadź niepoprawne hasło
        login_page.submit_password('zlehaslo123')

        # 5. Kliknij 'LOGIN'
        login_page.click_login_btn()


        # Oczekiwany rezultat: Wyświetla się komunikat "Invalid Email or
Password"
```

```
        error_notices = self.driver.find_elements_by_xpath('//div[@class="alert
alert-danger"]')
        visible_error_notices = []
        for error in error_notices:
            if error.is_displayed():
                visible_error_notices.append(error)
        self.assertEqual(len(visible_error_notices), 1)
        self.assertEqual(visible_error_notices[0].text, "Invalid Email or
Password")

if __name__ ==  '__main__':
    unittest.main(verbosity=2)
```

test_login.py

```
import unittest
from selenium import webdriver
import time
from pages.home_page import HomePage
from pages.signup_page import SignupPage
from pages.login_page import LoginPage
from locators.signup_page_locators import SignupPageLocators
from selenium.webdriver.common.keys import Keys
from ddt import ddt, data, unpack
import csv

@ddt
class TravelSignUp(unittest.TestCase):

    # Scenariusz: Rejestracja nowego użytkownika z użyciem niepoprawnego adresu
email

    def setUp(self):

        # Warunki wstępne: Przeglądarka otwarta na stronie https://
www.phptravels.net/

        self.driver = webdriver.Chrome()
        self.driver.maximize_window()
        self.driver.get("https://www.phptravels.net/")

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

    def get_data(file_name):
        rows = []
        data_file = open(file_name, 'rt')
        reader = csv.reader(data_file)
        next(reader, None)
        for row in reader:
            rows.append(row)
        return rows


    @data(*get_data("wrong_email.csv"))
    @unpack
    def test_signup_invalid_email(self, first_name, last_name, mobile_num,
wrong_email, password, password_confirmed):

        # 1 & 2 Kliknij "MY ACCOUNT" i "Login"
```

```
        home_page = HomePage(self.driver)
        home_page.click_my_account_btn()

        # 3. Kliknij przycisk ‘SIGN UP’
        login_page = LoginPage(self.driver)
        login_page.click_sign_up_btn()

        signup_page = SignupPage(self.driver)

        # 4. Wprowadź imię
        signup_page.submit_first_name(first_name)

        # 5. Wprowadź nazwisko
        signup_page.submit_last_name(last_name)

        # 6. Wprowadź numer komórkowy
        signup_page.submit_mobile_number(mobile_num)

        # 7. Wprowadź niepoprawny adres email
        signup_page.submit_email(wrong_email)

        # 8. Wprowadz hasło
        signup_page.submit_password(password)

        # 9. Wprowadź powtórnie hasło
        signup_page.confirm_password(password_confirmed)

        # 10. Potwierdz rejestrację klikając 'SIGN UP'
        signup_page.confirm_signup()

        # Oczekiwany rezultat: Konto nie zostaje utworzone. Wyświetlany jest
komunikat: “The Email field must contain a valid email address. ”

        error_notices = self.driver.find_elements_by_xpath('//div[@class="alert
alert-danger"]')
        visible_error_notices = []
        for error in error_notices:
            if error.is_displayed():
                visible_error_notices.append(error)
        self.assertEqual(len(visible_error_notices), 1)
        self.assertEqual(visible_error_notices[0].text, "The Email field must
contain a valid email address.")


if __name__ ==  '__main__':
    unittest.main(verbosity=2)
```

test_tours.py

```
import unittest
from selenium import webdriver
from pages.home_page import HomePage
from pages.signup_page import SignupPage
from pages.login_page import LoginPage
from pages.tours_page import ToursPage
import time

class TravelTours(unittest.TestCase):
# Scenariusz: Wyszukanie wycieczki bez podania lokalizacji

    def setUp(self):

        # Warunki wstępne: Przeglądarka otwarta na stronie https://
www.phptravels.net/

        self.driver = webdriver.Chrome()
        self.driver.maximize_window()
        self.driver.get("https://www.phptravels.net/")

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

    def test_tour(self):

        tours_page = HomePage(self.driver)

        # 1. Kliknij 'TOURS'
        tours_page.click_tours()

        # 2.Wpisz date
        #tours_page.enter_date('18/06/2019')
        tours_page.select_date()

        # 3. Wpisz liczbe osob
        tours_page.select_adults_number()

        # 4. Wybierz rodzaj wycieczki
        tours_page.select_tour_type()

        # 5. Kliknij 'SEARCH'
        tours_page.click_search()
        time.sleep(5)

        # Oczekiwany rezultat: Uzytkownik jest przenoszony na stronę z wynikami.
Wyświetla się komunikat "No Results Found"
        tours_page = ToursPage(self.driver)
        time.sleep(3)

        assert "No Results Found" in self.driver.page_source
```