МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Вінницький національний технічний університет

Лабораторні роботи

з дисципліни: «Тестування програмного забезпечення»

Виконав:

студент групи 1КН-20мс

Кравченко Я.М.

Перевірила:

Польгуль Т.Д.

Вінниця 2021р.

Хід роботи

Selenium - це проект, який призначений головним чином для автоматизації тестування веб-додатків. Проект речі, open-source, тобто з відкритим вихідним кодом. У будь-який момент можна заглянути всередину коду і подивитися, як працює той чи інший метод, а також можна написати свої пропозиції розробникам проекту, і, більш того, зробити pull request з виправленнями і поліпшеннями коду.

Це збірний, зонтичний проект, тобто він складається з безлічі самих різних компонентів, наприклад:

1. Selenium WebDriver
2. Selenium Grid
3. Selenium Server
4. Selenium IDE

Автоматизація UI-тестування (тестування інтерфейсів) дозволяє скоротити час проведення регресійного тестування і отримати більше впевненості в якості продукту перед його релізом в продакшн. Також автоматизовані регресивні тести залишають більше часу, щоб більш повно тестувати нову функціональність в продуктах.

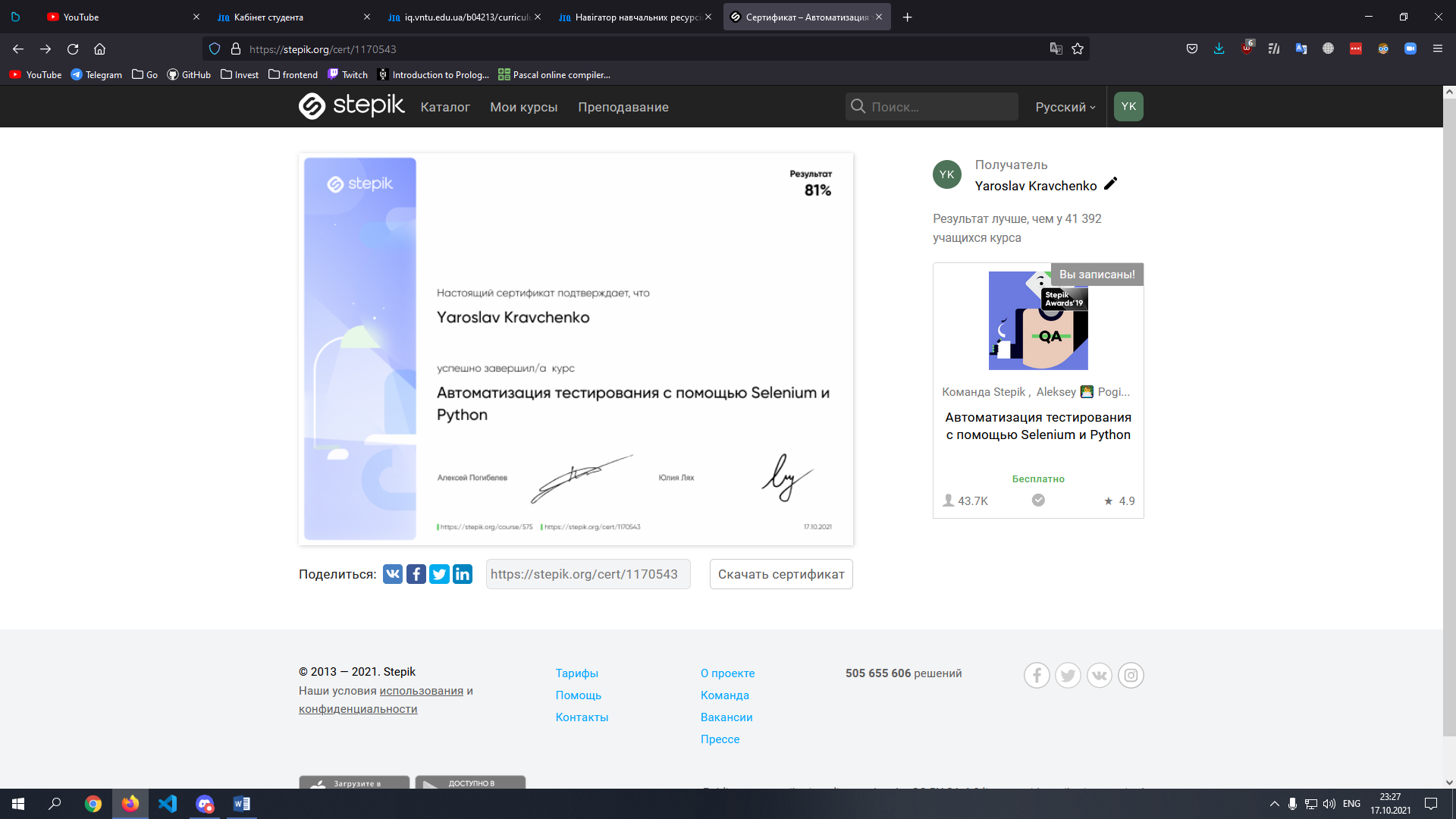


Рисунок 1.1 – Сертифікат про проходження курсу

Приклад тесту, що був створений в процесі проходження курсу (урок 6, крок 3):

import pytest

import time

import math

from selenium import webdriver

@pytest.fixture(scope="function")

def browser():

print("\nstart browser for test..")

browser = webdriver.Chrome()

browser.implicitly\_wait(5)

yield browser

print("\nquit browser..")

browser.quit()

class TestMain:

message = ''

@pytest.mark.parametrize('linkId', ["236895", "236896", "236897", "236898", "236899", "236903", "236904", "236905"])

def test\_send\_answer(self, browser, linkId):

link = f"https://stepik.org/lesson/{linkId}/step/1"

browser.get(link)

text\_area = browser.find\_element\_by\_css\_selector(".ember-text-area ")

text\_area.send\_keys(str(math.log(int(time.time()))))

button = browser.find\_element\_by\_css\_selector(".submit-submission")

button.click()

result = browser.find\_element\_by\_css\_selector(".smart-hints\_\_hint")

self.message += result.text

print(self.message)

assert "Correct!" in result.text

Даний тест створений за допомогою фреймворку PyTest та охоплює теми параметризації тестів та додання fixture. Даний тест виконує такі дії:

1. перехід на сторінку з певним параметризованим значенням;
2. знаходить поле для введення відповіді, відповідь це чисельне значення, що генерується на основі поточного часу;
3. виконує submit введених даних, як тільки дані відправляються з’явиться сповіщення з результатом на вашу відповідь;
4. знаходить блок, що відповідає за сповіщення;
5. додає текст з сповіщення в змінну message, та виводить на екран;
6. проводить перевірку чи сповіщення являється “Correct!”.

Тест буде пройдений якщо сповіщення буде містити текст “Correct!”. Суть даного завдання в тому що, потрібно виділити тести, що мають результат Fail та виділити з них текстову частину. Всі частини складуть певне речення, що і являється відповіддю у завданні з курсу.

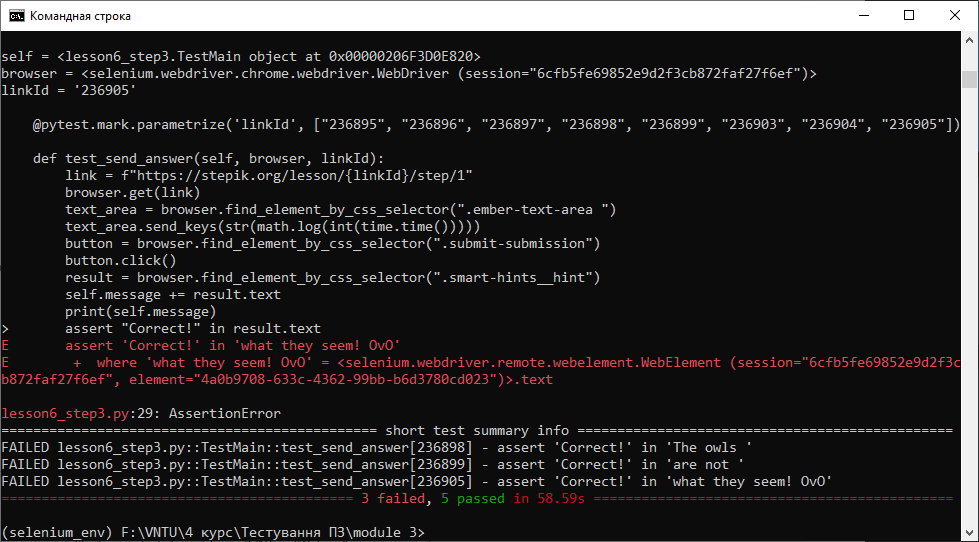


Рисунок 1.2 – Приклад виконання

**Висновок:** Отже, у процесі проходження даного курсу ми ознайомились з можливостями проекту Selenium, а саме з його частиною Selenium WebDriver. Також були розглянуті фреймоврки для автоматизації тестів, а саме: PyTest та unittest.