

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №4

із дисципліни «Автоматизоване тестування програмного забезпечення»

на тему

Тестування веб-застосунку за допомогою Python та Selenium WebDriver

Виконав:

студент групи КМ-82

Бубела Д. В.

Керівник:

асистент

Громова В. В.

ЗМІСТ

Мета роботи	3
1 Постановка задачі	5
2 Основна частина.....	8
3 Розробка тест-кейсів	9
Висновки	12
Перелік посилань	13
Додаток А Текст модулів, що реалізують автоматичне тестування	14
Додаток Б Скріншоти результатів виконання автотестів.....	15
Додаток В Відповіді на контрольні запитання	17

МЕТА РОБОТИ

Ознайомитися з «Selenium WebDriver». Протестувати локально веб-застосунок простої структури відповідно до специфікації. Тестування здійснювати в python за допомогою модуля unittest з застосуванням «Selenium WebDriver»

Специфікація

Веб-застосунок призначений для отримання коментарів-відповідей на запитання, складається з однієї веб-сторінки і має наступну функціональність:

1. Довільний користувач може переглянути питання, яке розташоване в лівій верхній частині сторінки, та існуючі коментарі до цього питання.

2. **Sign Up**: новий користувач може зареєструватися, увівши свої значення в поля Display Name, Email, Password та натиснувши кнопку **Sign Up**. При виконання умови, що всі поля заповнені і значення Email – унікальне серед всіх e-mail на backend, реєстрація буде виконана успішно, користувач автоматично входить в свій обліковий запис, і на сторінці з'являється кнопка **Log Out**.

3. Користувач може вийти зі свого облікового запису, натиснувши кнопку **Log Out**.

4. **Log In**: зареєстрований користувач може увійти в свій обліковий запис. Для цього необхідно ввести правильні e-mail та пароль в поля Email та Password , після чого натиснути кнопку **Log In**.

5. Користувач, що увійшов в свій обліковий запис, може додавати коментарі до питання. Для цього користувачу необхідно написати текст коментаря в спеціальному редакторі, що знаходиться на сторінці, та натиснути кнопку New comment.

6. До натискання кнопки **New comment** користувач може форматувати текст коментаря за допомогою кнопок редактора – виділити коментар або його частину **жирним** (кнопка **B**), *курсивом* (кнопка *I*), підкресленим (кнопка U) або ~~закресленим~~ (кнопка ~~Θ~~).

7. Користувач, що увійшов в свій обліковий запис, може видаляти створений ним коментар, натиснувши кнопку **Remove**, розташовану справа від потрібного коментаря.

8. За замовчуванням на сторінці присутні коментарі “Test comment 1”, “Test comment 2”, створені користувачами Alice A. та Bob B., які мають наступні Email та Password: alice_2002@gmail.com, ‘aaa’ bob_2001@gmail.com, ‘bbb’.

9. Для спрощення задачі веб-застосунок не зберігає дані постійно. При зупинці backend всі дані будуть втрачені. Тестові користувачі та коментарі створюються автоматично при кожному запуску backend.

1 ПОСТАНОВКА ЗАДАЧІ

Необхідні для встановлення пакети в пайтон: tornado, selenium. Selenium WebDriver відповідно до браузера встановлюється в систему.

Завдання

Створити тестовий модуль — скрипт на мові Python, який в автоматичному режимі запускає веб-сервер та виконує наступні дії:

а) Перевіряє, що анонімний користувач бачить на сторінці:

1) Заголовок питання, який співпадає з очікуваним заголовком.

2) Текст питання (переконатися, що блок під заголовком є непустим).

3) Коментарі до питання, які існують за замовчуванням (переконатися, що співпадають тексти та відповідні автори коментарів).

4) Кнопки Sign Up, Log In.

5) Поля для введення Display Name, Email та Password та їх підписи.

б) Перевіряє, що новий користувач може зареєструватись, увівши значення Display Name, Email та Password:

1) Протестувати, що у випадку успішної реєстрації користувач автоматично входить у свій обліковий запис, на сторінці з'являється його Display Name та кнопка Log Out.

2) Протестувати, що натискання на кнопку Log Out призводить до виходу користувача зі свого облікового запису

3) Протестувати, що при неунікальному значенні Email на backend створення нового користувача не відбувається, веб-застосунок не переходить в стан зареєстрованого користувача.

в) Перевіряє, що зареєстрований користувач, увівши правильні значення

Email та Password та натиснувши на кнопку Log In, входить до свого облікового запису. Протестувати, що при введенні некоректного Email або Password — користувач не входить до свого облікового запису.

г) Перевіряє, що якщо користувач знаходиться у своєму обліковому запису, то на сторінці є Редактор для введення нового коментаря та кнопка New comment. Виконати JS script, що робить border елемента з id = “editor section” червоного кольору. Зробити скріншоти веб-сторінки засобами Selenium WebDriver.

д) Перевіряє, що користувач, який знаходиться у своєму обліковому запису, може ввести новий коментар:

1) Після введення тексту коментаря та натискання на кнопку New comment створений коментар відображається у списку всіх коментарів.

2) Текст коментаря, відформатований в Редакторі за допомогою кнопок B, I, U, O відображається у коментарі так, як очікується.

3) Поруч зі створеним коментарем присутнє ім'я користувача Display Name та кнопка Remove.

е) Перевіряє, що користувач, який знаходиться у своєму обліковому запису, може ввести ще один коментар:

1) Переконатися, що вдалося створити ще один коментар.

2) Переконатися, що коментар знаходиться після попереднього коментаря, створеного даним користувачем.

ж) Перевіряє, що користувач, який знаходиться у своєму обліковому запису, може видалити створений ним коментар:

1) Переконатися, що видаляється потрібний коментар.

2) Переконатися, що усі інші коментарі, створені даним користувачем та іншими користувачами, залишилися на місці.

з) Перевіряє, що одночасно у своїх облікових записах можуть бути активними два різних користувача:

1) Увійти в браузері driver1 в обліковий запис користувача user1 та створити коментар comment1.

2) В автоматичному режимі запустити ще один браузер driver2, виконати Log In від імені користувача user2 та створити коментар comment2. Протестувати, що в браузері driver2 присутні обидва коментаря — comment1 та comment2.

3) В браузері driver1 від імені користувача user1 додати коментар comment3. Протестувати, що в браузері driver1, присутні три коментаря — comment1, comment2 та comment3.

4) В браузері driver2 видалити коментар comment2. Протестувати, що в браузері driver2 присутні два коментаря — comment1 та comment3.

5) Закрити обидва браузера, після чого знову запустити браузер і протестувати, що анонімний користувач бачить два коментаря — comment1 та comment3.

2 ОСНОВНА ЧАСТИНА

В обох завданнях уся робота здійснюється в python з бібліотекою для тестувань unittest. Лабораторна робота проводиться в браузері «Mozilla Firefox 83.0». Для здійснення приймального тестування використовувався модуль Popen з бібліотеки subprocess. Він потрібен для створення нового процесу від запущеного скрипта, що забезпечував би роботу веб сервера, який тестується. Метод setUp створює процес “бекенду” і ініціалізує веб-драйвер. Метод tearDown завершує роботу веб-драйвера та сервера, що тестується. Ці методи викликаються безпосередньо перед та після виконання кожного з тестів. Під час тестування в консоль виводяться помилки сервера, але так має бути. Unittest перенаправляє всі вихідні потоки підпроцесу у вивід тої програми, яка запущена для тестування. Для коректного збереження скріншоту в завданні г) необхідно додати таймаут `time.sleep(2)`.

3 РОЗРОБКА ТЕСТ-КЕЙСІВ

Завдання

ВИСНОВКИ

Ми навчилися використовувати Selenium WebDriver для приймального тестування веб-додатків. У ході роботи було розроблено 8 тест-кейсів для повного покриття специфікації програми. Також попрактикувались в створенні автоматичних тестів з допомогою unittest та виклику окремих інструкцій до та після виконання окремих тестів. Переглянули способи використання конструкцій `__enter__` та `__exit__` що було необхідно для розуміння роботи викликів вебдрайверу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Python Software Foundation. subprocess — Subprocess management [Електронний ресурс] / Python Software Foundation.. — 2020. — Режим доступу до ресурсу: <https://docs.python.org/3/library/subprocess.html>..
2. Selenium IDE API Reference [Електронний ресурс]. — 2019. — Режим доступу до ресурсу: <https://www.selenium.dev/selenium-ide/docs/en/api/commands>.
3. MDN contributors. XSLT/XPath Reference [Електронний ресурс] / MDN contributors. — 2020. — Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/XPath>.
4. Python Software Foundation. unittest — Unit testing framework [Електронний ресурс] / Python Software Foundation.. — 2020. — Режим доступу до ресурсу: <https://docs.python.org/3/library/unittest.html>.
5. FriendFeed. Structure of a Tornado web application [Електронний ресурс] / FriendFeed // 6.1.0. — 2019. — Режим доступу до ресурсу: <https://www.tornadoweb.org/en/stable/guide/structure.html>.

Додаток А

Текст модулів, що реалізують автоматичне тестування

Додаток Б

Скріншоти результатів виконання автотестів

```
(venv-gmbot) ~/K/A/L/M/source (main|+1) $ python test.py -v
test_1_anonymous (__main__.AppTest)
1. Перевіряє, що анонімний користувач бачить на сторінці ... ok
test_2_signup (__main__.AppTest)
2. Перевіряє, що новий користувач може зареєструватись ... ERROR:tornado.acce
ok
test_3_log_in (__main__.AppTest)
3. Перевіряє, що зареєстрований користувач входить до свого облікового запису
WARNING:tornado.access:403 POST /api/v1/user/login (127.0.0.1) 0.45ms
ok
test_4_editor (__main__.AppTest)
4. Перевіряє, що на сторінці є Редактор та кнопка New comment ... ok
test_5_comment (__main__.AppTest)
5. Перевіряє, що користувач, може ввести новий коментар ... ok
test_6_comment (__main__.AppTest)
6. Перевіряє, що користувач, може ввести ще один коментар ... ok
test_7_remove (__main__.AppTest)
7. Перевіряє, що користувач може видалити створений ним коментар ... ok
test_8_multiplayer (__main__.AppTest)
Перевіряє, що можуть бути активними два різних користувача ... ok

-----
Ran 8 tests in 43.614s

OK
```

Рисунок Б.1 – Скріншот тестування test_1.py

Додаток В

Відповіді на контрольні запитання

1) Що таке Selenium? Яке його призначення? Чому серія програмних продуктів Selenium вивчається в курсі «Автоматизоване тестування програмного забезпечення»?

Selenium — це безкоштовна система автоматизованого тестування, що використовується для перевірки веб-додатків у різних браузерах та платформах. Вона дозволяє використовувати кілька мов програмування, такі як Java, C#, Python тощо для створення тестових скриптів, так званих «Selenium Test Scripts». Програмне забезпечення Selenium — це набір програмного забезпечення, кожна частина відповідає різним потребам в тестуванні.

2) Що таке Selenium IDE? Які його основні переваги та особливості застосування?

Інтегроване середовище розробки селену (IDE) - це найпростіший фреймворк у наборі Selenium. Це плагін для Firefox та Chrome, за допомогою якого розробник створює Selenium Test Scripts.

Просте у використанні та установці. Може експортувати тести у формати, що використовуються в Selenium RC та WebDriver. Вбудований модуль довідки та результатів тестування, забезпечує підтримку розширень.

Розроблений лише для Mozilla Firefox та Google Chrome. Створений тільки для прототипування тестів. Для використання умовних операторів та циклів часто необхідно використовувати результати роботи javascript. Працює повільніше в порівнянні з Selenium RC та WebDriver.

3) Які основні команди виконуються в Selenium IDE?

Основні команди, що зокрема використовувались в лабораторній роботі: open, execute script, click, store text, store attribute, store xpath count, while end, close. Документацію до них можна переглянути на офіційному сайті [2].

4) Як відбувається процес відлагодження в Selenium IDE? Опишіть його основні етапи і особливості.

Зневаджувач в дозволяє ставити точки зупину в скрипті, починати з обраного місця, ставити виконання на паузу (та продовжувати), рухатися покроково, регулювати швидкість виконання скрипта.

Для тестування певного кроку — на нього, ставиться точка зупину. В такому випадку тестування автоматично стане на паузу. Під час паузи слід змінити неробочий елемент, або, якщо це дослідження його роботи — скористатися можливостями браузера для перегляду його розміщення в коді сторінки і роботи. Після здійснення зневадження окремого кроку можна або продовжити виконання роботи скрипт в звичайному режимі, або перейти до наступного кроку і продовжити зневадження.

5) Що таке локатор? Які типи локаторів вам відомі?

Локаатор — використовується для вибору конкретного елемента або елементів як вхідних даних до інших команд у Selenium, він вибирає певну множину елементів з DOM дерева сторінки. В Selenium IDE існують такі типи локаторів:

- ID
- Name
- Link
- XPath
- CSS

6) Які існують способи одержання значень локаторів?

Команди `store [attribute|json|text|title|value|window handle|xpath count]` дозволяють записати отримане значення в змінну, назва якої знаходиться в полі `Target`.

7) Яке призначення команд `assert` та `verify` в Selenium IDE?

Обидві команди перевіряють чи відповідає змінна очікуваному значенню. В разі негативного результату `assert` зупиняє роботу скрипта, а `verify` — ні.

8) Яке призначення команди `wait for` в Selenium IDE?

Команда `wait for` буде чекати допоки не виконається умова для вибраного елемента.

9) Як можна зберігати і використовувати змінні в Selenium IDE?

Функції, що зазначені трьома пунктами вище дозволяють зберегти змінні. Для їх використання — в полі `Value` записується `${ назва_змінної}`

10) Опишіть використання `TestCase` та `TestSuite`?

`Test` — послідовність команд, що використовується для автоматичного тестування веб-додатку.

`Test Suite` — група тестів. За замовчуванням всі тести додаються в `Default Suite`.

Групи тестів зазвичай використовуються для розділення тестування різних частин веб-додатків.