Практична робота 3

Мета: Створення автоматизованих тестів для веб-додатків за допомогою Selenium. Запуск тестів і аналіз результатів.

Завдання:

- 1. Перевірка заголовку веб-сторінки
- 2. Перевірка полів введення та кнопки "Пошук"
- 3. Навігація та перевірка посилань
- 4. Крос-браузерне тестування з використанням

Selenium Grid

5. Data-Driven Testing з використанням зовнішніх

джерел даних

Завдання 1: Перевірка заголовку веб-сторінки

Опис: Створіть тестовий випадок для перевірки правильності заголовку веб-сторінки.

Кроки:

- Відкрийте браузер та перейдіть на сайт за заданою URL.
- Перевірте, що заголовок сторінки відповідає очікуваному

(наприклад, "Google" для google.com).

• Закрийте браузер.

Підказки:

Використовуйте метод driver.getTitle() для отримання заголовку сторінки.

Завдання 2: Перевірка полів введення та кнопки "Пошук"

Опис: Напишіть тестовий випадок для перевірки роботи форми пошуку на сайті.

Кроки:

- Відкрийте браузер та перейдіть на сайт, де є поле для пошуку (наприклад, Google).
- Введіть текст у поле пошуку (наприклад, "Selenium WebDriver").
- Натисніть на кнопку "Пошук" або використайте клавішу "Enter".
- Перевірте, що результати пошуку відображаються.

Підказки:

Використовуйте методи driver.findElement() для знаходження поля пошуку та кнопки.

Використовуйте метод sendKeys() для введення тексту.

Завдання 3: Навігація та перевірка посилань

Опис: Створіть тест для перевірки того, що посилання на веб-сторінці ведуть на правильні URL-адреси.

Кроки:

- Відкрийте браузер та перейдіть на сторінку, де є декілька посилань.
- Знайдіть і натисніть на перше посилання на сторінці.
- Перевірте, що нова сторінка відкрилася з правильним URL.
- Поверніться на початкову сторінку.

Підказки:

Використовуйте методи driver.navigate().to() для переходу між сторінками та driver.getCurrentUrl() для отримання поточного URL.

Завдання 4: Крос-браузерне тестування з використанням Selenium Grid

Опис: Налаштуйте середовище Selenium Grid для виконання тестів на різних браузерах та операційних системах, забезпечуючи крос-браузерну сумісність веб-додатку.

Кроки:

- Налаштування Selenium Grid:
- Встановіть Selenium Grid Hub на локальній машині або сервері.
- Додайте кілька Node з різними браузерами (Chrome, Firefox, Edge) та операційними системами (Windows, macOS, Linux).
- Розробка тестів:
- Напишіть тести, які можуть виконуватись паралельно на різних браузерах.
- Використовуйте DesiredCapabilities або Options для визначення браузера та платформи в конфігурації тестів.

- Виконання тестів:
- Запустіть тести через Selenium Grid.
- Забезпечте логування та звітність для кожного браузера окремо.
- Аналіз результатів:
- Перевірте, що веб-додаток працює коректно на всіх підтримуваних браузерах.
- Виявте та задокументуйте несумісності або проблеми.

Підказки:

Використовуйте TestNG Parallel Execution для запуску тестів одночасно на різних браузерах.

Завдання 5: Data-Driven Testing з використанням зовнішніх джерел даних

Опис: Розробіть тести, які використовують зовнішні джерела даних (наприклад, CSV, Excel, бази даних) для введення різних комбінацій даних у веб-додаток.

Кроки:

- Підготовка даних:
- Створіть файл CSV або Excel з різними наборами даних для тестування (наприклад, різні комбінації логінів та паролів).
- Імплементація Data-Driven підходу:
- Використовуйте бібліотеки, такі як Apache POI для роботи з Excel або OpenCSV для CSV файлів.
- Реалізуйте механізм читання даних з файлу та передачі їх у тести.
- Написання тестів:

- Створіть тести для функцій, що потребують введення даних (наприклад, форма реєстрації або входу).
- Використовуйте різні набори даних для перевірки коректності обробки вводу.
- Валідація результатів:
- Перевірте, що для кожного набору даних веб-додаток реагує відповідно (успішна реєстрація, повідомлення про помилку тощо).

Підказки:

Використовуйте TestNG DataProvider або JUnit Parameterized Tests для інтеграції з даними.

Забезпечте обробку винятків для некоректних або відсутніх даних.

Допоміжні матеріали

Короткий мануал по налаштуванню:

Крок 1: Встановлення необхідних інструментів

• JDK (Java Development Kit)

Завантажте та встановіть JDK (рекомендовано JDK 8 або новішу версію).

Переконайтеся, що змінна середовища JAVA_HOME налаштована.

Завантажити можна тут: Oracle JDK

• Maven

Maven — це інструмент для управління проектами та залежностями.

Завантажте Maven: Maven Download

Налаштуйте шлях до Maven у змінній середовища M2_HOME.

• IDE (IntelliJ IDEA / Eclipse....)

Завантажте будь-яке зручне середовище розробки, наприклад IntelliJ IDEA або Eclipse, Visual Studio Code, ets.

Крок 2: Створення Maven проекту

- 1. Відкрийте IntelliJ IDEA або Eclipse.
- 2. Створіть новий Maven проект.
- 3. У файлі pom.xml додайте залежність для Selenium WebDriver:

<dependencies>

<!-- Selenium WebDriver -->

<dependency>

<groupId>org.seleniumhq.selenium</proupId>

<artifactId>selenium-java</artifactId>

<version>4.5.0</version> <!-- Оновіть до актуальної версії -->

</dependency>

<!-- TestNG (або JUnit) для тестування -->

<dependency>

- <groupId>org.testng</groupId>
- <artifactId>testng</artifactId>

<version>7.6.1</version>

<scope>test</scope>

</dependency>

</dependencies>

Після цього Maven автоматично завантажить всі необхідні залежності.

Крок 3: Встановлення WebDriver для браузера

Selenium WebDriver взаємодіє з браузерами через драйвери. Для кожного браузера потрібен свій драйвер:

Chrome: Завантажте <u>ChromeDriver</u>

Firefox: Завантажте GeckoDriver

Edge: Завантажте <u>EdgeDriver</u>

Додайте драйвер у системну змінну РАТН, або вкажіть шлях до драйвера в коді:

System.setProperty("webdriver.chrome.driver", "шлях/до/chromedriver");

Крок 4: Написання першого тесту

Створіть новий Java-клас у папці src/test/java, наприклад GoogleSearchTest.java, та додайте наступний код:

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class GoogleSearchTest {

WebDriver driver;

@BeforeClass

public void setUp() {

// Встановіть шлях до ChromeDriver, якщо не додали в РАТН System.setProperty("webdriver.chrome.driver", "шлях/до/chromedriver"); driver = new ChromeDriver(); // Запуск Chrome driver.manage().window().maximize(); // Розгортання вікна браузера

@Test

}

```
public void googleSearchTest() {
```

```
driver.get("https://www.google.com"); // Перехід на сайт Google
```

WebElement searchBox = driver.findElement(By.name("q")); // Пошук елемента поля пошуку

```
searchBox.sendKeys("Selenium WebDriver"); // Введення тексту
```

searchBox.submit(); // Відправлення форми

// Додайте очікування або перевірку результатів, наприклад, заголовку

assert driver.getTitle().contains("Selenium WebDriver");

@AfterClass

```
public void tearDown() {
```

driver.quit(); // Закриття браузера після виконання тесту

Крок 5: Виконання тесту

1. Запустіть тест через IDE або через Maven командою

mvn test

Крок 6: Додаткові налаштування (опціонально)

• **Headless режим**: Використовуйте цей режим для запуску тестів без відображення вікна браузера (актуально для CI/CD):

```
ChromeOptions options = new ChromeOptions();
options.addArguments("--headless");
WebDriver driver = new ChromeDriver(options);
```

• Waits: Додайте очікування для елементів, щоб тести були стабільнішими:

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));

WebElement element = wait.until(ExpectedConditions.visibilityOfElementLocated(By.name("q")));

Корисні ресурси

- Selenium Documentation
- <u>TestNG Documentation</u>

Вимоги до здачі та захисту лабораторної роботи №3:

- Звіт повинен бути оформленим в документ
- Оформлення звіту:
- 1. рамки не є обов'язковими
- 2. документ повинен бути читабельним
- 3. виділяйте відповіді на кожне завдання зі скріншотами та прикладами лістингу коду
- 4. звіт заливати на гугл диск
- 5. Опціонально залити проект з задачами у відкритий доступ (github/gitlab....)

Порівняльна таблиця інструментів для автоматизованого тестування

| | | | | · · · | | |
|-------------------|----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Тип тестування | Підтримувані мови програмування | Підтримувані браузери | Інтеграція з CI/CD | Ліцензія | Основні переваги | |
| UI тестування | Java, C#, Python, Ruby, JavaScript | Chrome, Firefox, Safari, Edge, Opera | Jenkins, Travis CI, GitLab CI | Відкрите (Open Source) | Широка підтримка браузерів та мов програмування, велика спільнота, гнучкість | |
| UI тестування | JavaScript | Chrome, Edge, Electron | Jenkins, CircleCl, GitHub Actions | Відкрите (Free) / Платне | Швидке виконання тестів, інтеграція з сучасними фреймворками, простота налаштування | |
| UI тестування | JavaScript, TypeScript, Python, C# | Chrome, Firefox, Safari, Edge | Jenkins, GitHub Actions, CircleCl | Відкрите (Open Source) | Підтримка сучасних браузерів, можливість тестування мобільних веб-додатків, автоматичне очікування елементів | |
| | Тип тестування UI тестування UI тестування UI тестування | Тип тестуванняПідтримувані мови програмуванняUI тестуванняJava, C#, Python, Ruby, JavaScriptUI тестуванняJavaScriptUI тестуванняJavaScriptUI тестуванняJavaScript | Тип тестуванняПідтримувані мови програмуванняПідтримувані браузериUI тестуванняJava, C#, Python, Ruby, JavaScriptChrome, Firefox, Safari, Edge, OperaUI тестуванняJavaScriptChrome, Firefox, Safari, Edge, OperaUI тестуванняJavaScriptChrome, Edge, ElectronUI тестуванняJavaScript, TypeScript, Python, C#Chrome, Firefox, Safari, Edge | Тип тестуванняПідтримувані мови програмуванняПідтримувані браузериІнтеграція з СІ/СDUI тестуванняJava, C#, Python, Ruby, JavaScriptChrome, Firefox, Safari, Edge, OperaJenkins, Travis CI, GitLab CIUI тестуванняJavaScriptChrome, Edge, ElectronJenkins, CircleCI, GitHub ActionsUI тестуванняJavaScriptChrome, Firefox, Safari, Edge, OperaJenkins, CircleCI, GitHub ActionsUI тестуванняJavaScriptChrome, Edge, ElectronJenkins, CircleCI, GitHub ActionsUI тестуванняJavaScript, TypeScript, Python, C#Chrome, Firefox, Safari, EdgeJenkins, GitHub Actions, CircleCI | Тип тестуванняПідтримувані мови програмуванняПідтримувані браузериІнтеграція з СІ/СDЛіцензіяUI тестуванняJava, C#, Python, Ruby, JavaScriptChrome, Firefox, Safari, Edge, OperaJenkins, Travis CI, GitLab CIВідкрите (Open Source)UI тестуванняJavaScriptChrome, Edge, ElectronJenkins, CircleCI, GitHub ActionsВідкрите (Free) / ПлатнеUI тестуванняJavaScriptChrome, Edge, ElectronJenkins, CircleCI, GitHub ActionsВідкрите (Free) / ПлатнеUI тестуванняJavaScript, TypeScript, Python, C#Chrome, Firefox, Safari, EdgeJenkins, GitHub Actions, CircleCI (Open Source) | Тип тестуванняПідтримувані мови програмуванняПідтримувані браузериІнтеграція з СІ/СDЛіцензіяОсновні перевагиUI тестуванняJava, C#, Python, Ruby, JavaScriptChrome, Firefox, Safari, Edge, OperaJenkins, Travis CI, GitLab CIBigkpure (Open Source)Широка підтримка браузерів та мов програмування, велика спільнота, гнучкістьUI тестуванняJavaScriptChrome, Edge, ElectronJenkins, CircleCI, GitHub ActionsBigkpure (Pree) / ПлатнеШвидке виконання тестів, інтеграція з сучасними фреймворками, простота налаштуванняUI тестуванняJavaScript, TypeScript, Python, C#Chrome, Firefox, Safari, EdgeJenkins, GitHub Actions, CircleCIBigkpure (Open Source)Підтримка сучасних браузерів, можливість тестування мобільних веб-додатків, автоматичне очікування елементів |

- Selenium є найстарішим та найбільш універсальним інструментом з великою спільнотою та підтримкою різних мов і браузерів.
- **Cypress** відомий своєю швидкістю та зручністю для розробників, особливо у проектах на JavaScript, але має обмежену підтримку браузерів порівняно з Selenium.
- **Playwright** пропонує сучасні можливості та підтримку різних мов програмування і браузерів, поєднуючи гнучкість Selenium з деякими зручностями Cypress.

Вибір інструменту залежить від конкретних вимог проекту, технічних навичок команди та специфіки тестування, яке потрібно провести.