

Лабораторна робота #7

Автоматизація тестування вебсайту на прикладі сайту kpi.ua

Мета роботи: Навчитися використовувати засоби автоматизації тестування на прикладі Playwright для автоматизованого тестування вебсайтів, перевірити доступність сторінок, правильність перенаправлення та функціональність ключових елементів на сайті.

Крок 1: Перевірка завантаження головної сторінки

- Мета: Переконаватися, що головна сторінка сайту успішно завантажується без помилок.
- Завдання: Відкрити вебсайт <https://kpi.ua> та перевірити код відповіді сервера (200 OK).
- Приклад:

```
JS test1-hompage.js X
Lab7 > tests > JS test1-hompage.js > ...
1  const { test, expect } = require('@playwright/test');
2
3  test('Перевірка завантаження головної сторінки', async ({ page }) => {
4    const response = await page.goto('https://kpi.ua');
5    expect(response.status()).toBe(200);
6  });
7
```

Результат:

```
✓ 1 tests\test1-hompage.spec.js:3:1 > Перевірка завантаження головної сторінки (4.1s)
```

Крок 2: Перевірка коректності HTTP → HTTPS перенаправлення

- Мета: Перевірити, чи перенаправляється запит з <http://kpi.ua> на <https://kpi.ua>.
- Завдання:
 - Відкрити сайт за допомогою HTTP (<http://kpi.ua>).
 - Переконаватися, що автоматично відбувається перенаправлення на HTTPS (<https://kpi.ua>).
- Приклад:

```
Lab7 > tests > JS test2-redirect.spec.js > ...
1  const { test, expect } = require('@playwright/test');
2
3  test('Перевірка HTTP → HTTPS перенаправлення', async ({ page }) => {
4    const response = await page.goto('http://kpi.ua');
5    expect(response.status()).toBe(200);
6    expect(page.url().replace(/\/$/, '')).toBe('https://kpi.ua');
7  });
8
```

Результат:

```
✓ 3 tests\test2-redirect.spec.js:3:1 > Перевірка HTTP → HTTPS перенаправлення (4.3s)
```

Крок 3: Тестування роботи меню навігації

- Мета: Переконаватися, що всі посилання у головному меню працюють коректно
- Завдання:
 - Перейти до кожного пункту меню
 - Перевірити, що сторінка за відповідним посиланням завантажується без помилок.
- Приклад:

```
Lab7 > tests > JS test3-navigation.spec.js > ...  
1  const { test, expect } = require('@playwright/test');  
2  test('Тестування роботи меню навігації', async ({ page }) => {  
3    await page.goto('https://kpi.ua');  
4    const aboutUsLink = await page.locator('a[href="/about"]');  
5    await aboutUsLink.click();  
6    expect(await page.title()).toContain('Про університет');  
7  });  
8
```

Результат:

```
✓ 4 tests\test3-navigation.spec.js:2:1 > Тестування роботи меню навігації (6.2s)
```

Крок 4: Перевірка внутрішніх посилань

- Мета: Упевнитися, що всі внутрішні посилання на сторінці є працездатними.
- Завдання:
 - Знайти всі внутрішні посилання на головній сторінці.
 - Перевірити, що кожне з них веде до сторінки, яка успішно завантажується (код відповіді 200 OK).
- Приклад:

```

Lab7 > tests > JS test4-internal-links.spec.js > test('Перевірка внутрішніх посилань') callback
1  const { test, expect } = require('@playwright/test');
2  test('Перевірка внутрішніх посилань', async ({ page }) => {
3      await page.goto('https://kpi.ua');
4      const links = await page.locator('a[href^="/"]');
5      for (let i = 0; i < await links.count(); i++) {
6          const link = links.nth(i);
7          const href = await link.getAttribute('href');
8          const response = await page.goto(`https://kpi.ua${href}`);
9          expect(response.status()).toBe(200);
10     }
11 });

```

Результат:

```

2) tests\test4-internal-links.spec.js:2:1 > Перевірка внутрішніх посилань
Test timeout of 30000ms exceeded.

Error: page.goto: net::ERR_ABORTED; maybe frame was detached?
Call log:
- navigating to "https://kpi.ua/en/structure", waiting until "load"

6 |         const link = links.nth(i);
7 |         const href = await link.getAttribute('href');
> 8 |         const response = await page.goto(`https://kpi.ua${href}`);
    |                                     ^
9 |         expect(response.status()).toBe(200);
10 |     }
11 | });
    at D:\Docs\LABS\3_КУРС\ТЗЯПС\Lab7\tests\test4-internal-links.spec.js:8:37

```

Крок 5: Тестування адаптивності сторінки

- Мета: Перевірити, як сайт виглядає на пристроях з різними розмірами екрану.
- Завдання:
 - Емулювати пристрої (смартфон, планшет, десктоп) і перевірити, чи коректно відображаються всі елементи сайту.
- Приклад:

```

Lab7 > tests > JS test5-responsiveness.spec.js > test("Тестування адаптивності сторінки") callback
1  const { test, expect } = require('@playwright/test');
2
3  test('Тестування адаптивності сторінки', async ({ page }) => {
4    await page.goto('https://kpi.ua');
5
6    await page.setViewportSize({ width: 1280, height: 720 });
7    expect(await page.isVisible('header')).toBe(true);
8
9    await page.setViewportSize({ width: 768, height: 1024 });
10   expect(await page.isVisible('header')).toBe(true);
11
12   await page.setViewportSize({ width: 375, height: 667 });
13   expect(await page.isVisible('header')).toBe(true);
14 });
15

```

Результат:

```

✓ 6 tests\test5-responsiveness.spec.js:3:1 > Тестування адаптивності сторінки (5.2s)

```

Крок 6: Тестування часу завантаження сайту

- Мета: Переконатися, що час завантаження сторінки є оптимальним.
- Завдання: Виміряти час завантаження сторінки та порівняти його з рекомендаціями (менше 3 секунд).
- Приклад:

```

Lab7 > tests > JS test6-loading-time.spec.js > test("Тестування часу завантаження сайту") callback
1  const { test, expect } = require('@playwright/test');
2
3  test('Тестування часу завантаження сайту', async ({ page }) => {
4    const startTime = Date.now();
5    await page.goto('https://kpi.ua');
6    const endTime = Date.now();
7
8    const loadTime = endTime - startTime;
9    console.log(`Час завантаження: ${loadTime} мс`);
10   expect(loadTime).toBeLessThan(3000);
11 });

```

Результат:

```

3) tests\test6-loading-time.spec.js:3:1 > Тестування часу завантаження сайту

Error: expect(received).toBeLessThan(expected)

Expected: < 3000
Received:   3846

   8 |     const loadTime = endTime - startTime;
   9 |     console.log(`Час завантаження: ${loadTime} мс`);
>  10 |     expect(loadTime).toBeLessThan(3000);
      |                           ^
   11 |   });
   12 |
      at D:\Docs\LABS\3_КУРС\ТЗЯПС\Lab7\tests\test6-loading-time.spec.js:10:20

```

Можна помітити, що час завантаження сторінки більше 3 секунд.

Крок 7: Перевірка наявності всіх основних елементів на головній сторінці

- Мета: Переконаватися, що на головній сторінці присутні всі основні елементи, такі як:
 - Логотип сайту.
 - Головне меню навігації.
 - Блок новин або оголошень.
 - Пошуковий рядок.
- Завдання: Візуально перевірити, що ці елементи коректно відображаються на сторінці та не зникають при зміні розміру екрана.
- Приклад:

```

Lab7 > tests > JS test7-main-elements.spec.js > test("Перевірка наявності основних елементів на головній сторінці") callback
1  const { test, expect } = require('@playwright/test');
2  test('Перевірка наявності основних елементів на головній сторінці', async ({ page }) => {
3    await page.goto('https://kpi.ua', { waitUntil: 'load' });
4
5    const logo = await page.locator('.site-logo > a:nth-child(1) > img:nth-child(1)');
6    await expect(logo).toBeVisible();
7    const mainMenu = await page.locator('#superfish-main');
8    await expect(mainMenu).toBeVisible();
9    const newsBlock = await page.locator('.nws > a:nth-child(1)');
10   await expect(newsBlock).toBeVisible();
11   const searchBox = await page.locator('.srch > a:nth-child(1)');
12   await expect(searchBox).toBeVisible();
13   const screenSize = [
14     { width: 1200, height: 800 },
15     { width: 375, height: 667 }
16   ];
17
18   for (const size of screenSize) {
19     await page.setViewportSize(size);
20     await expect(logo).toBeVisible();
21     await expect(mainMenu).toBeVisible();
22     await expect(newsBlock).toBeVisible();
23     await expect(searchBox).toBeVisible();
24   }
25 });
26

```

Результат:

```

✓ 8 tests\test7-main-elements.spec.js:3:1 > Перевірка наявності основних елементів на головній сторінці (4.6s)

```

Крок 8: Тестування доступності сайту

- Мета: Перевірити, чи відповідає сайт основним принципам доступності (accessibility).
- Завдання:
 - Переконалися, що кожен елемент має коректні атрибути alt, aria-label або аналогічні.
 - Використати інструменти тестування доступності.
- Приклад:

```
Lab7 > tests > JS test8-accessibility.spec.js > test("Тестування доступності сайту") callback
1  const { test, expect } = require('@playwright/test');
2
3  test('Тестування доступності сайту', async ({ page }) => {
4    await page.goto('https://kpi.ua');
5
6    const images = await page.locator('img');
7    for (let i = 0; i < await images.count(); i++) {
8      const image = images.nth(i);
9      expect(await image.getAttribute('alt')).not.toBeNull();
10   }
11 });
```

Результат:

```
✓ 9 tests\test8-accessibility.spec.js:3:1 > Тестування доступності сайту (4.6s)
```

Крок 9: Перевірка завантаження сторінок іншими мовами

- Мета: Переконалися, що сайт коректно відображається при виборі іншої мови інтерфейсу (наприклад, англійської).
- Завдання:
 - Знайти перемикач мови на сайті.
 - Вибрати англійську мову (або будь-яку іншу доступну).
 - Перевірити, що всі основні елементи сайту (меню, заголовки, кнопки) відображаються обраною мовою.
- Приклад:

```

Lab7 > tests > JS test9-language-switch.spec.js > test(Перевірка зміни мови інтерфейсу на англійську) callback
1  const { test, expect } = require('@playwright/test');
2
3  test('Перевірка зміни мови інтерфейсу на англійську', async ({ page }) => {
4      await page.goto('https://kpi.ua');
5      const languageSwitcher = await page.locator('a[href="/en/"]');
6      await languageSwitcher.click();
7      const links = await page.locator('a');
8      const headers = await page.locator('h1, h2, h3');
9      const buttons = await page.locator('button, input[type="button"], input[type="submit"]');
10     const testItems = [
11         links,
12         headers,
13         buttons
14     ];
15     for (const item of testItems) {
16         const itemTexts = await item.allTextContents();
17         itemTexts.forEach(text => {
18             expect(text).toMatch(/^[A-Za-z\s,.'""-()]+$\/);
19         });
20     }
21 });

```

Результат:

```

4) tests\test9-language-switch.spec.js:3:1 > Перевірка зміни мови інтерфейсу на англійську —
Error: expect(received).toMatch(expected)

Expected pattern: /^[A-Za-z\s,.'""-()]+$\/
Received string: "Історичні фотографії"

    20 |         const itemTexts = await item.allTextContents();
    21 |         itemTexts.forEach(text => {
>   22 |             expect(text).toMatch(/^[A-Za-z\s,.'""-()]+$\/);
        |                               ^
    23 |         });
    24 |     }
    25 | });
    at forEach (D:\Docs\LABS\3_КУРС\ТЗЯПС\Lab7\tests\test9-language-switch.spec.js:22:26)
    at D:\Docs\LABS\3_КУРС\ТЗЯПС\Lab7\tests\test9-language-switch.spec.js:21:19

Slow test file: tests\test4-internal-links.spec.js (30.0s)

```

Крок 10*: Перевірка завантаження зовнішніх ресурсів

- Мета: Переконаватися, що всі зовнішні ресурси (зображення, CSS, JavaScript) завантажуються коректно.
- Завдання:
 - Перевірити, що всі ресурси мають оптимальний час завантаження (без значних затримок, тобто не більше 3 секунд).
- Приклад:

```

Lab7 > tests > JS test10-external-resources.spec.js > test('Перевірка коректного завантаження всіх зовнішніх ресурсів') callback
1  const { test, expect } = require('@playwright/test');
2
3  test('Перевірка коректного завантаження всіх зовнішніх ресурсів', async ({ page }) => {
4    const failedRequests = [];
5    page.on('response', response => {
6      const url = response.url();
7      const status = response.status();
8
9      if (status === 404 || status === 500) {
10       failedRequests.push({ url, status });
11     }
12   });
13   const startTime = Date.now();
14   page.on('requestfinished', (request) => {
15     const duration = Date.now() - startTime;
16     expect(duration).toBeLessThan(3000);
17   });
18
19   await page.goto('https://kpi.ua');
20   expect(failedRequests.length).toBe(0);
21 });
22

```

Результат:

Можна побачити, що не всі ресурси мають оптимальний час завантаження (не більше 3 секунд).

```

2) tests\test10-external-resources.spec.js:3:1 > Перевірка коректного завантаження всіх зовнішніх ресурсів

Error: expect(received).toBeLessThan(expected)

Expected: < 3000
Received: 3037

14 |   page.on('requestfinished', (request) => {
15 |     const duration = Date.now() - startTime;
> 16 |     expect(duration).toBeLessThan(3000);
    |                               ^
17 |   });
18
19 |   await page.goto('https://kpi.ua');
   at Page.<anonymous> (D:\Docs\LABS\3_КУРС\ТЗЯПС\Lab7\tests\test10-external-resources.spec.js:16:22)

Error: page.goto: Test ended.
Call log:
  - navigating to "https://kpi.ua/", waiting until "load"

17 |   });
18
> 19 |   await page.goto('https://kpi.ua');
    |               ^
20 |   expect(failedRequests.length).toBe(0);
21 | });
22
   at D:\Docs\LABS\3_КУРС\ТЗЯПС\Lab7\tests\test10-external-resources.spec.js:19:14

```


Загальний результат тестів

Running 10 tests using 2 workers

```
✓ 1 tests\test1-homepage.spec.js:3:1 > Перевірка завантаження головної сторінки (4.1s)
✗ 2 tests\test10-external-resources.spec.js:3:1 > Перевірка зміни мови інтерфейсу на англійську (6.3s)
✓ 3 tests\test2-redirect.spec.js:3:1 > Перевірка HTTP → HTTPS перенаправлення (4.3s)
✓ 4 tests\test3-navigation.spec.js:2:1 > Тестування роботи меню навігації (6.2s)
✗ 5 tests\test4-internal-links.spec.js:2:1 > Перевірка внутрішніх посилань (30.0s)
✓ 6 tests\test5-responsiveness.spec.js:3:1 > Тестування адаптивності сторінки (5.2s)
✗ 7 tests\test6-loading-time.spec.js:3:1 > Тестування часу завантаження сайту (4.3s)
Час завантаження: 3846 мс
✓ 8 tests\test7-main-elements.spec.js:3:1 > Перевірка наявності основних елементів на головній сторінці (4.6s)
✓ 9 tests\test8-accessibility.spec.js:3:1 > Тестування доступності сайту (4.6s)
✗ 10 tests\test9-language-switch.spec.js:3:1 > Перевірка зміни мови інтерфейсу на англійську (4.6s)
```