

Лабораторна робота #7

Автоматизація тестування вебсайту на прикладі сайту kpi.ua

ФБ-24 Синеуцький Олександр

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).

```
tests > JS step1.spec.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
```

```
PS C:\Users\fidje\OneDrive\Desktop\lab7_kpittest> npx playwright test tests/step1.spec.js

Running 1 test using 1 worker
  1 passed (5.8s)

To open last HTML report run:

npx playwright show-report
```

```
Project: firefox

▼ step1.spec.js

✓ Перевірка завантаження головної сторінки
  step1.spec.js:3
```

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>).
 - Переконавшись, що автоматично відбувається перенаправлення на HTTPS (<https://kpi.ua>).

```
tests > JS step2.spec.js > test("Перевірка HTTP HTTPS перенаправлення") callback
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
7    expect(page.url().replace(/\/$/, ''')).toBe('https://kpi.ua');
8  });
9
```

```
PS C:\Users\fidje\OneDrive\Desktop\lab7_kpittest> npx playwright test tests/step2.spec.js

Running 1 test using 1 worker
1 passed (6.0s)

To open last HTML report run:

npx playwright show-report
```

Project: firefox

▼ step2.spec.js

✓ Перевірка HTTP HTTPS перенаправлення
step2.spec.js:3

Крок 3: Тестування роботи меню навігації

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

```

tests > JS step3.spec.js > ...
1  const { test, expect } = require('@playwright/test');
2
3  test("Тестування роботи меню навігації", async ({ page }) => {
4      await page.goto('https://kpi.ua');
5      const aboutUsLink = page.locator('a[href="/about"]');
6
7      await aboutUsLink.click();
8      expect(await page.title()).toContain('Про університет');
9      const admissionLink = page.locator('a[href="/admission"]');
10     await admissionLink.click();
11
12     const admissionHeader = page.locator('h1');
13     expect(await admissionHeader.textContent()).toContain("Вступ в КПІ ім.Ігоря Сікорського");
14     const educationLink = page.locator('a[href="/education"]');
15     await educationLink.click();
16
17     expect(await page.title()).toContain("Освіта");
18     const scienceLink = page.locator('a[href="/science"]');
19     await scienceLink.click();
20     expect(await page.title()).toContain('Наука та інновації');
21
22     const studentsLink = page.locator('a[href="/kpi_students"]');
23     await studentsLink.click();
24
25     const studentSection = page.locator('h2:has-text("Освітній процес)');
26     expect(await studentSection.isVisible()).toBe(true);
27 });
28

```

```

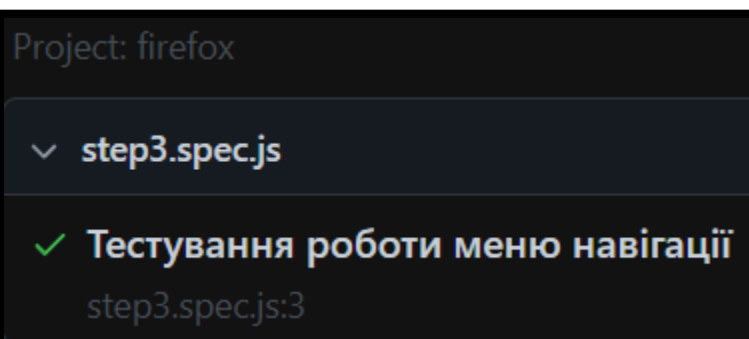
PS C:\Users\fidje\OneDrive\Desktop\lab7_kpittest> npx playwright test tests/step3.spec.js

Running 1 test using 1 worker
  1 passed (16.1s)

To open last HTML report run:

npx playwright show-report

```



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 ОК).

```
tests > JS step4.spec.js > ...
1  const { test, expect } = require('@playwright/test');
2
3  test("Перевірка внутрішніх посилань", async ({ page, context }) => {
4      test.setTimeout(400000);
5      await page.goto('https://kpi.ua');
6      const links = await page.locator('a[href^="/"]');
7      const totalLinks = await links.count();
8      console.log(`Знайдено ${totalLinks} внутрішніх посилань`);
9
10     for (let i = 0; i < totalLinks; i++) {
11         const link = links.nth(i);
12         const href = await link.getAttribute('href');
13
14         if (!href || href === '#' || href.startsWith("javascript")) {
15             console.warn(`Пропущено некоректне посилання [${i + 1}]: ${href || 'немає атрибута href'}`);
16             continue;
17         }
18
19         const fullUrl = `https://kpi.ua${href}`;
20         console.log(`Перевіряємо посилання [${i + 1}]: ${fullUrl}`);
21
22         try {
23             const response = await context.request.get(fullUrl, { timeout: 15000 });
24             const status = response.status();
25             expect(status).toBe(200);
26             console.log(`Посилання працює [${i + 1}]: ${fullUrl}`);
27         } catch (error) {
28             console.error(`Помилка при перевірці посилання [${i + 1}]: ${fullUrl}\n${error.message}`);
29         }
30
31         await new Promise(resolve => setTimeout(resolve, 500));
32     }
33
34     console.log("Перевірка завершена");
35 });
36
```

```
PS C:\Users\fidje\OneDrive\Desktop\lab7_kpittest> npx playwright test tests/step4.spec.js

Running 1 test using 1 worker
[firefox] > step4.spec.js:3:1 > Перевірка внутрішніх посилань
Знайдено 198 внутрішніх посилань
Перевіряємо посилання [1]: https://kpi.ua/
Посилання працює [1]: https://kpi.ua/
Перевіряємо посилання [2]: https://kpi.ua/
Посилання працює [2]: https://kpi.ua/
Перевіряємо посилання [3]: https://kpi.ua/searcher
Посилання працює [3]: https://kpi.ua/searcher
Перевіряємо посилання [4]: https://kpi.ua/
```

```
Перевіряємо посилання [196]: https://kpi.ua/location
Посилання працює [196]: https://kpi.ua/location
Перевіряємо посилання [197]: https://kpi.ua/2022-emergency-procedure
Посилання працює [197]: https://kpi.ua/2022-emergency-procedure
Перевіряємо посилання [198]: https://kpi.ua/files/2024-op.pdf
Посилання працює [198]: https://kpi.ua/files/2024-op.pdf
Перевірка завершена
Slow test file: [firefox] > step4.spec.js (6.0m)
Consider splitting slow test files to speed up parallel execution
1 passed (6.1m)

To open last HTML report run:

npx playwright show-report
```

Project: firefox

✓ step4.spec.js

✓ Перевірка внутрішніх посилань

step4.spec.js:3

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.

```

tests > JS step5.spec.js > ...
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: 1200, height: 800 });
7      console.log('Десктоп: 1200x800');
8
9      expect(await page.isVisible('header')).toBe(true);
10     expect(await page.isVisible('footer')).toBe(true);
11     expect(await page.isVisible('main')).toBe(true);
12
13     console.log('Десктоп: Хедер, футер, основний контент видно.');

```

```

PS C:\Users\fidje\OneDrive\Desktop\lab7_kpitest> npx playwright test tests/step5.spec.js

Running 1 test using 1 worker
[firefox] > step5.spec.js:3:1 > Тестування адаптивності сторінки
Десктоп: 1200x800
Десктоп: Хедер, футер, основний контент видно.
Планшет: 768x1024
Планшет: Хедер, футер, основний контент видно.
Смартфон: 375x667
Смартфон: Хедер, футер, основний контент видно.
1 passed (7.4s)

To open last HTML report run:

npx playwright show-report

```

Project: firefox

▼ step5.spec.js

✓ Тестування адаптивності сторінки
step5.spec.js:3

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)

```
tests > JS step6.spec.js > ...
1  const { test, expect } = require('@playwright/test');
2
3  test('Тестування часу завантаження сайту', async ({ page }) => {
4    const start = Date.now();
5    await page.goto('https://kpi.ua');
6    const loadTime = Date.now() - start;
7
8    console.log(`Час завантаження: ${loadTime} мс`);
9
10   if (loadTime < 3000) {
11     console.log('Час завантаження в межах норми (менше 3 секунд).');
12   } else {
13     console.log('Час завантаження перевищує норму (більше 3 секунд).');
14   }
15 });
16
```

```
PS C:\Users\fidje\OneDrive\Desktop\lab7_kpittest> npx playwright test tests/step6.spec.js
```

```
Running 1 test using 1 worker
```

```
[firefox] > step6.spec.js:3:1 > Тестування часу завантаження сайту
```

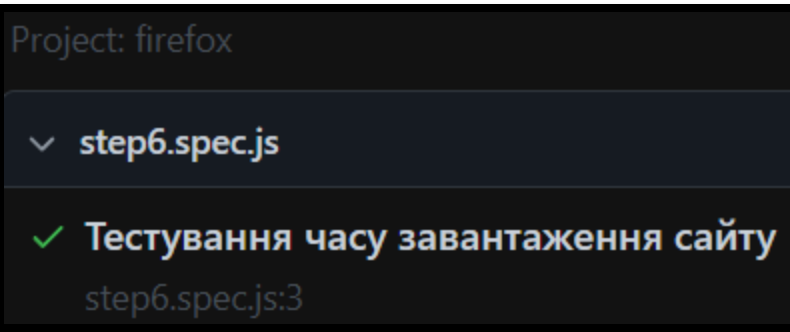
```
Час завантаження: 2232 мс
```

```
Час завантаження в межах норми (менше 3 секунд).
```

```
1 passed (5.7s)
```

```
To open last HTML report run:
```

```
npx playwright show-report
```



Крок 7: Перевірка наявності всіх основних елементів на головній сторінці

- Мета: Переконатися, 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.

```
tests > JS step7.spec.js > ...
  1  const { test, expect } = require('@playwright/test');
  2
  3  test('Перевірка наявності основних елементів на головній сторінці', async ({ page }) => {
  4      await page.goto('https://kpi.ua');
  5
  6      const logoSelector = 'img[alt="Головна"]';
  7      await page.waitForSelector(logoSelector);
  8      expect(await page.isVisible(logoSelector)).toBe(true);
  9
 10     const menuSelector = '#superfish-main';
 11     await page.waitForSelector(menuSelector);
 12     expect(await page.isVisible(menuSelector)).toBe(true);
 13
 14     const newsBlockSelector = '.field--name-body';
 15     await page.waitForSelector(newsBlockSelector);
 16     expect(await page.isVisible(newsBlockSelector)).toBe(true);
 17
 18     const searchBarSelector = 'a[title="search"]';
 19     await page.waitForSelector(searchBarSelector);
 20     expect(await page.isVisible(searchBarSelector)).toBe(true);
 21
 22     const devices = [
 23         { width: 1920, height: 1080 },
 24         { width: 768, height: 1024 },
 25     ];
```



```

26
27     for (const viewport of devices) {
28         console.log(`Перевірка адаптивності для розміру: ${viewport.width}x${viewport.height}...`);
29         await page.setViewportSize(viewport);
30
31         expect(await page.isVisible(logoSelector)).toBe(true);
32         console.log('Логотип присутній.');

```

```
PS C:\Users\fidje\OneDrive\Desktop\lab7_kpittest> npx playwright test tests/step7.spec.js
```

```

Running 1 test using 1 worker
[firefox] > step7.spec.js:3:1 > Перевірка наявності основних елементів на головній сторінці
Перевірка адаптивності для розміру: 1920x1080...
Логотип присутній.
Головне меню присутнє.
Блок новин присутній.
Пошуковий рядок присутній.
Перевірка адаптивності для розміру: 768x1024...
Логотип присутній.
Головне меню присутнє.
Блок новин присутній.
Пошуковий рядок присутній.
Усі перевірки виконані успішно.
  1 passed (7.4s)

```

To open last HTML report run:

```
npx playwright show-report
```

Project: firefox

▼ step7.spec.js

✓ Перевірка наявності основних елементів на головній сторінці
step7.spec.js:3

Крок 8: Тестування доступності сайту

- Мета: Перевірити, чи відповідає сайт основним принципам доступності (accessibility).

- Завдання:

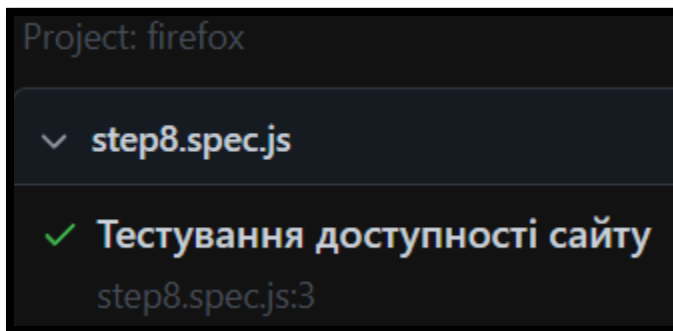
- Переконалися, що кожен елемент має коректні атрибути alt, aria-label
- або аналогічні.
- Використати інструменти тестування доступності

```
tests > JS step8.spec.js > ...
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 elements = await page.locator("[alt], [role], [aria-label], [aria-labelledby], [aria-hidden], [tabindex], [aria-"]
7      const elementCount = await elements.count();
8      console.log(`Перевірка доступності для ${elementCount} елементів із спеціальними атрибутами...`);
9
10     let issues = 0;
11     for (let i = 0; i < elementCount; i++) {
12         const element = elements.nth(i);
13         const role = await element.getAttribute('role');
14         const alt = await element.getAttribute("alt");
15         const ariaLabel = await element.getAttribute('aria-label');
16         const ariaHidden = await element.getAttribute("aria-hidden");
17         const tabindex = await element.getAttribute('tabindex');
18         const ariaLive = await element.getAttribute("aria-live");
19
20         console.log(`Елемент #${i + 1}:`);
21         console.log(` alt: ${alt || "немає"}`);
22         console.log(` role: ${role || "немає"}`);
23         console.log(` aria-label: ${ariaLabel || "немає"}`);
24         console.log(` aria-hidden: ${ariaHidden || "немає"}`);
25         console.log(` tabindex: ${tabindex || "немає"}`);
26         console.log(` aria-live: ${ariaLive || "немає"}`);
27
28         if (!role || !alt || !ariaLabel || !ariaHidden || !tabindex || !ariaLive) {
29             issues++;
30         }
31     }
32
33     if (issues > 8) {
34         console.error(`Виявлено ${issues} потенційно недоступних елементів.`);
35     } else {
36         console.log(`Всі елементи мають атрибути доступності.`);
37     }
38
39     console.log("Перевірка завершена.");
40 });
41
```

```
PS C:\Users\fidje\OneDrive\Desktop\lab7_kpittest> npx playwright test tests/step8.spec.js
```

```
Running 1 test using 1 worker
[firefox] > step8.spec.js:3:1 > Тестування доступності сайту
Перевірка доступності для 46 елементів із спеціальними атрибутами...
Елемент #1:
alt: Головна
role: немає
aria-label: немає
aria-hidden: немає
tabindex: немає
aria-live: немає
Елемент #2:
alt: офіційний тікток університету
role: немає
aria-label: немає
aria-hidden: немає
tabindex: немає
aria-live: немає
```

```
Елемент #46:  
alt: arrow_right sharing button  
role: немає  
aria-label: немає  
aria-hidden: немає  
tabindex: немає  
aria-live: немає  
Виявлено 46 потенційно недоступних елементів.  
Перевірка завершена.  
1 passed (11.3s)  
  
To open last HTML report run:  
  
npx playwright show-report
```



Крок 9: Перевірка завантаження сторінок іншими мовами

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

```

tests > JS step9.spec.js > test("Перевірка тексту на англійській сторінці") callback
1   const { test, expect } = require('@playwright/test');
2
3   test("Перевірка тексту на англійській сторінці", async ({ page }) => {
4       await page.goto('https://kpi.ua');
5
6       const languageSwitch = page.locator('a[href="/en/"]');
7       const isSwitchVisible = await languageSwitch.isVisible();
8       console.log('Перевірка перемикача мови:', isSwitchVisible ? 'видимий' : 'не видимий');
9       expect(isSwitchVisible).toBe(true);
10
11      await languageSwitch.click();
12      console.log('Виконано перемикання на англійську версію');
13
14      try {
15          await page.waitForURL('**/en/**', { timeout: 10000 });
16          console.log('Перехід на англійську версію успішний');
17      } catch (error) {
18          console.error("Не вдалося перейти на англійську версію за 10 секунд.");
19          throw error;
20      }
21
22      const currentURL = page.url();
23      console.log('URL після переключення:', currentURL);
24      expect(currentURL).toContain('/en/');
25
26      const siteSlogan = page.locator('div.site_slogan a[rel="home"]');
27      const sloganText = await siteSlogan.textContent();
28      console.log('Текст слогану:', sloganText?.trim());
29      expect(sloganText).toContain('Igor Sikorsky Kyiv Polytechnic Institute');
30

```

```

30
31      const menu = page.locator('#superfish-main');
32      const menuText = await menu.textContent();
33      const menuItems = ['University', 'Education', 'Science', 'Admission', 'Student Life'];
34
35      console.log('Перевірка пунктів меню:');
36      menuItems.forEach(item => {
37          const contains = menuText.includes(item);
38          console.log(`${item} -> ${contains ? 'присутній' : 'відсутній'}`);
39          expect(menuText).toContain(item);
40      });
41
42      const headers = await page.locator('h1').allTextContents();
43      console.log('Заголовки сторінки:');
44      headers.forEach((header, index) => console.log(`Заголовок #${index + 1}: ${header.trim()}`));
45
46      console.log('Елементи сайту коректно відображаються англійською мовою');
47  });
48

```

```
PS C:\Users\fidje\OneDrive\Desktop\lab7_kpittest> npx playwright test tests/step9.spec.js

Running 1 test using 1 worker
[firefox] > step9.spec.js:3:1 > Перевірка тексту на англійській сторінці
Перевірка перемикача мови: видимий
Виконано перемикання на англійську версію
Перехід на англійську версію успішний
URL після переключення: https://kpi.ua/en/
Текст слогану: National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
Перевірка пунктів меню:
"University" -> присутній
"Education" -> присутній
"Science" -> присутній
"Admission" -> присутній
"Student Life" -> присутній

Заголовки сторінки:
Заголовок #1: Mission:

Елементи сайту коректно відображаються англійською мовою
1 passed (9.4s)

To open last HTML report run:

npx playwright show-report
```

Project: firefox

▼ step9.spec.js

✓ Перевірка тексту на англійській сторінці
step9.spec.js:3

Крок 10*: Перевірка завантаження зовнішніх ресурсів

- Мета: Переконалися, що всі зовнішні ресурси (зображення, CSS, JavaScript)

завантажуються коректно.

- Завдання:

- Відкрити консоль розробника браузера (DevTools) і перевірити вкладку "Network".
- Переконалися, що немає помилок завантаження (404, 500).
- Перевірити, що всі ресурси мають оптимальний час завантаження (без значних затримок, тобто не більше 3 секунд).

```

tests > JS step10.spec.js > ...
1  const { test, expect } = require('@playwright/test');
2
3  test("Перевірка завантаження зовнішніх ресурсів", async ({ page }) => {
4      const resources = new Map();
5
6      page.on('request', req => {
7          resources.set(req.url(), { url: req.url(), startTime: Date.now(), status: null, endTime: null });
8      });
9
10     page.on('response', res => {
11         const resource = resources.get(res.url());
12         if (resource) {
13             resource.endTime = Date.now();
14             resource.status = res.status();
15         }
16     });
17
18     await page.goto('https://kpi.ua');
19     await page.waitForTimeout(5000);
20
21     const results = Array.from(resources.values()).map(r => {
22         const loadTime = r.endTime ? r.endTime - r.startTime : null;
23         return { ...r, loadTime };
24     });
25
26     results.forEach(r => {
27         console.log(`Ресурс: ${r.url}, Час: ${r.loadTime ?? "невідомо"} мс, Статус: ${r.status ?? "невідомо"}, ` +
28             `${r.loadTime && r.loadTime > 3000 ? "Повільне" : "Оптимальне"}, ` +
29             `${r.status && r.status < 400 ? "Без помилок" : "Помилка (" + r.status + ")"}`);
30     });
31
32     const errors404 = results.filter(r => r.status === 404);
33     const errors500 = results.filter(r => r.status === 500);
34     const slowResources = results.filter(r => r.loadTime !== null && r.loadTime > 3000);
35
36     expect(errors404.length).toBe(0);
37     expect(errors500.length).toBe(0);
38     expect(slowResources.length).toBe(0);
39 });
40

```

```
PS C:\Users\fidje\OneDrive\Desktop\lab7_kpittest> npx playwright test tests/step10.spec.js
```

```
Running 1 test using 1 worker
```

```
[firefox] > step10.spec.js:3:1 > Перевірка завантаження зовнішніх ресурсів
```

```
Ресурс: https://kpi.ua/, Час: 189 мс, Статус: 200, Оптимальне, Без помилок
```

```
Ресурс: https://kpi.ua/libraries/drupal-superfish/css/superfish.css?soduq8, Час: 41 мс, Статус: 200, Оптимальне, Без помилок
```

```
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/ajax-progress.module.css?soduq8, Час: 46 мс, Статус: 200, Оп
```

```
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/align.module.css?soduq8, Час: 46 мс, Статус: 200, Оптимальні
```

```
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/autocomplete-loading.module.css?soduq8, Час: 49 мс, Статус:
```

```
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/fieldgroup.module.css?soduq8, Час: 52 мс, Статус: 200, Опти
```

```
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/container-inline.module.css?soduq8, Час: 54 мс, Статус: 200
```

```
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/clearfix.module.css?soduq8, Час: 56 мс, Статус: 200, Оптима
```

```
Ресурс: https://www.youtube.com/generate_204?4dVqFA, Час: 28 мс, Статус: 204, Оптимальне, Без помилок
```

```
Ресурс: https://www.youtube.com/generate_204?KdLU_A, Час: 26 мс, Статус: 204, Оптимальне, Без помилок
```

```
Ресурс: https://www.youtube.com/generate_204?jvA44Q, Час: 26 мс, Статус: 204, Оптимальне, Без помилок
```

```
Ресурс: https://www.youtube.com/generate_204?2fGNYw, Час: 27 мс, Статус: 204, Оптимальне, Без помилок
```

```
Ресурс: https://www.youtube.com/youtubei/v1/log_event?alt=json, Час: 73 мс, Статус: 200, Оптимальне, Без помилок
```

```
1 passed (10.9s)
```

```
To open last HTML report run:
```

```
npx playwright show-report
```

Project: firefox

✓ **step10.spec.js**

✓ **Перевірка завантаження зовнішніх ресурсів**
step10.spec.js:3