

Лабораторна робота #7

Автоматизація тестування вебсайту на прикладі сайту kpi.ua

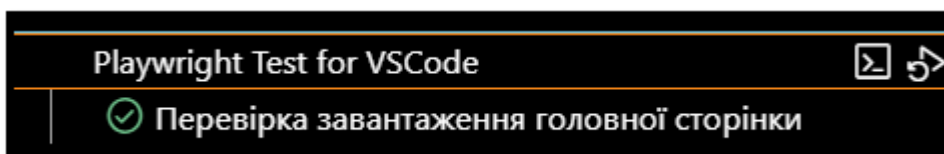
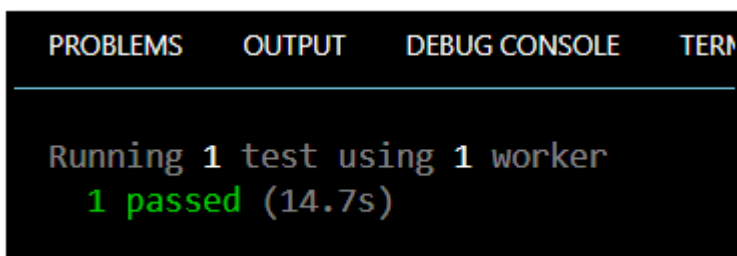
ФБ-24 Сторожев Андрій

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).

```
test('Перевірка завантаження головної сторінки', async ({ page }) => {  
  const response = await page.goto('https://kpi.ua');  
  expect(response.status()).toBe(200);  
});
```

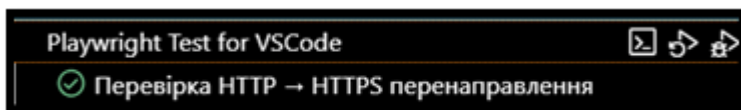


Крок 2: Перевірка коректності HTTP → HTTPS перенаправлення

- Мета: Перевірити, чи перенаправляється запит з <http://kpi.ua> на <https://kpi.ua>.
- Завдання:
 - Відкрити сайт за допомогою HTTP (<http://kpi.ua>).

```
test('Перевірка HTTP → HTTPS перенаправлення', async ({ page }) => {  
  const response = await page.goto('http://kpi.ua');  
  expect(response.status()).toBe(200);  
  expect(page.url()).toBe('https://kpi.ua/');  
});
```

```
Running 1 test using 1 worker  
1 passed (15.2s)
```

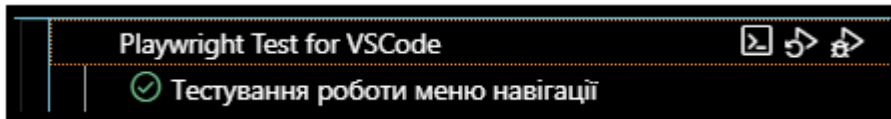


Крок 3: Тестування роботи меню навігації

- 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.

```
test('Тестування роботи меню навігації', async ({ page }) => {  
  await page.goto('https://kpi.ua');  
  const aboutUsLink = await page.locator('a.sf-depth-1.menuparent:has-text("Університет")');  
  await aboutUsLink.waitFor({ state: 'visible' });  
  await aboutUsLink.click();  
  expect(await page.title()).toContain('Про університет');  
});
```

```
Running 1 test using 1 worker  
1 passed (15.2s)
```



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).

```
test('Перевірка внутрішніх посилань', async ({ page, context }) => {
  test.setTimeout(40000); // Збільшення тайм-ауту для тесту
  await page.goto('https://kpi.ua');
  const links = await page.locator('a[href^="/"]');
  const totalLinks = await links.count();
  console.log(`Є ${totalLinks} внутрішніх посилань`);

  for (let i = 0; i < totalLinks; i++) {
    const link = links.nth(i);
    const href = await link.getAttribute('href');

    if (!href || href === '#' || href.startsWith('javascript:')) {
      console.warn('Пропущено некоректне посилання [${i + 1}]: ${href} || немає атрибута href');
      continue;
    }

    const fullUrl = `https://kpi.ua${href}`;
    console.log(`Перевірка (${i + 1}): ${fullUrl}`);

    try {
      const response = await context.request.get(fullUrl, { timeout: 20000 }); // Збільшено тайм-аут
      const status = await response.status(); // Використовуємо await для отримання статусу
      if (status !== 200) {
        console.warn(`Помилка (${i + 1}): ${fullUrl} - Статус: ${status}`);
      } else {
        console.log(`Перевірено (${i + 1}): ${fullUrl}`);
      }
    } catch (error) {
      console.error(`Помилка (${i + 1}): ${fullUrl}\n${error.message}`);
    }

    // Затримка між запитами
    await new Promise(resolve => setTimeout(resolve, 500));
  }

  console.log('Перевірка завершена');
});
```

```
Running 1 test using 1 worker
[chromium] > 4.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/
Посилання працює [4]: https://kpi.ua/
Перевіряємо посилання [5]: https://kpi.ua/en/
Посилання працює [5]: https://kpi.ua/en/
Перевіряємо посилання [6]: https://kpi.ua/news
Посилання працює [6]: https://kpi.ua/news
Перевіряємо посилання [7]: https://kpi.ua/boardlist
```

```
Посилання працює [193]: https://kpi.ua/site
Перевіряємо посилання [194]: https://kpi.ua/sitemap
Посилання працює [194]: https://kpi.ua/sitemap
Перевіряємо посилання [195]: https://kpi.ua/node
Посилання працює [195]: https://kpi.ua/node
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: [chromium] > 4.spec.js (5.2m)
Consider splitting slow test files to speed up parallel execution
1 passed (5.3m)
```

Playwright Test for VSCode

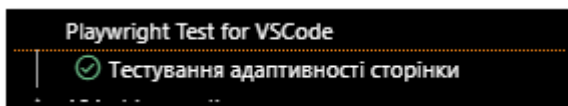
✓ Перевірка внутрішніх посилань

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.

```
test('Тестування адаптивності сторінки', async ({ page }) => {
  await page.goto('https://kpi.ua');
  await page.setViewportSize({ width: 1200, height: 800 });
  expect(await page.isVisible('header')).toBe(true);
  await page.setViewportSize({ width: 375, height: 667 });
  expect(await page.isVisible('header')).toBe(true);
});
```

```
Running 1 test using 1 worker
[chromium] > 5.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 (7.9s)
```



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)

```
test('Перевірка завантаження зовнішніх ресурсів', async ({ page }) => {
  const resources = new Map();
  page.on('request', req => {
    resources.set(req.url(), {
      url: req.url(),
      startTime: Date.now(),
      status: null,
      endTime: null
    });
  });
  page.on('response', res => {
    const resource = resources.get(res.url());
    if (resource) {
      resource.endTime = Date.now();
      resource.status = res.status();
    }
  });
  await page.goto('https://kpi.ua');
  await page.waitForTimeout(5000);

  const results = Array.from(resources.values()).map(r => {
    const loadTime = r.endTime ? r.endTime - r.startTime : null;
    return { ...r, loadTime };
  });

  results.forEach(r => {
    const statusText = r.status ? (r.status < 400 ? 'без помилок' : `Помилка (${r.status})`) : 'Невідомо';
    const loadText = r.loadTime ? (r.loadTime > 3000 ? 'Повільне' : 'Оптимальне') : 'Невідомо';
    console.log(`\nURL: ${r.url}, Час завантаження: ${r.loadTime ? r.loadTime + 'мс' : 'Невідомо'}, Статус: ${r.status ? r.status : 'Невідомо'}, Оцінка: ${loadText}, ${r.loadTime ? r.loadTime : 'Невідомо'} мс`);
  });

  const errors404 = results.filter(r => r.status === 404);
  const errors500 = results.filter(r => r.status === 500);
  const slowResources = results.filter(r => r.loadTime !== null && r.loadTime > 3000);

  expect(errors404.length).toBe(0);
  expect(errors500.length).toBe(0);
  expect(slowResources.length).toBe(0);
});
```

```
Running 1 test using 1 worker
[chromium] > 6.spec.js:3:1 > Тестування часу завантаження сайту
Час завантаження: 2960 мс
Час завантаження в межах норми (менше 3 секунд).
1 passed (5.6s)
```

Playwright Test for VSCode



✓ Тестування часу завантаження сайту

Крок 7: Перевірка наявності всіх основних елементів на головній сторінці

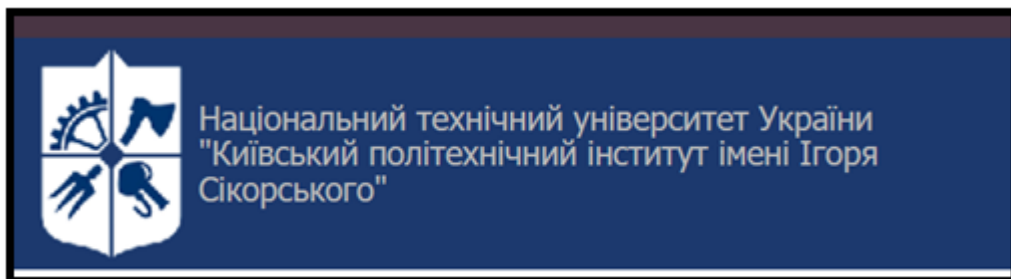
• Мета: Переконаватися, що на головній сторінці присутні всі основні елементи, такі

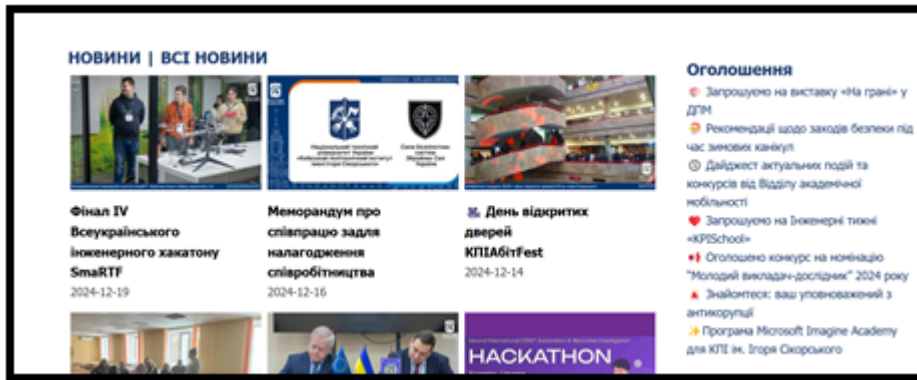
як:

- 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.

Візуально:





```
test('Перевірка наявності основних елементів на головній сторінці', async ({ page }) => {
  await page.goto('https://kpi.ua');

  // Логотип
  const logoSelector = 'img';
  await page.waitForSelector(logoSelector);
  expect(await page.isVisible(logoSelector)).toBe(true);

  // Меню
  const menuSelector = '#superfish-main';
  await page.waitForSelector(menuSelector);
  expect(await page.isVisible(menuSelector)).toBe(true);

  // Блок новин
  const newsBlockSelector = '.field--name-body';
  await page.waitForSelector(newsBlockSelector);
  expect(await page.isVisible(newsBlockSelector)).toBe(true);

  // Пошуковий рядок
  const searchBarSelector = 'a[title="search"]';
  await page.waitForSelector(searchBarSelector);
  expect(await page.isVisible(searchBarSelector)).toBe(true);

  // Перевірка адаптивності
  const devices = [
    { width: 1920, height: 1080 },
    { width: 768, height: 1024 }
  ];

  for (const viewport of devices) {
    await page.setViewportSize(viewport);
    expect(await page.isVisible(logoSelector)).toBe(true);
    expect(await page.isVisible(menuSelector)).toBe(true);
    expect(await page.isVisible(newsBlockSelector)).toBe(true);
    expect(await page.isVisible(searchBarSelector)).toBe(true);
  }
});
```

```
Running 1 test using 1 worker
[chromium] > 7.spec.js:3:1 > Перевірка наявності основних елементів
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 (6.0s)
```

Крок 8: Тестування доступності сайту

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

```
test('Тестування доступності сайту', async ({ page }) => {
  await page.goto('https://kpi.ua');
  const elements = await page.locator('[alt], [role], [aria-label], [aria-labelledby], [aria-hidden], [tabindex], [aria-live], [aria-expanded], [aria-describedby]');
  const elementCount = await elements.count();
  let issues = 0;
  for (let i = 0; i < elementCount; i++) {
    const element = elements.nth(i);
    const alt = await element.getAttribute('alt');
    const role = await element.getAttribute('role');
    const ariaLabel = await element.getAttribute('aria-label');
    if (!alt && !role && !ariaLabel) {
      issues++;
    }
  }
});
```



```
Перевірка доступності для 48 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 тіток університету
- role: немає
- aria-label: немає
- aria-hidden: немає
- tabindex: немає
- aria-live: немає
```

```
Елемент #28:
- alt: немає
- role: navigation
- aria-label: немає
- aria-hidden: немає
- tabindex: немає
- aria-live: немає
Елемент #29:
- alt: немає
- role: navigation
- aria-label: немає
- aria-hidden: немає
- tabindex: немає
- aria-live: немає
```

Крок 9: Перевірка завантаження сторінок іншими мовами

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

```
test('Перевірка тексту на англійській сторінці', async ({ page }) => {
  await page.goto('https://kpi.ua');
  const languageSwitch = page.locator('a[href="/en/"]');
  const isSwitchVisible = await languageSwitch.isVisible();
  expect(isSwitchVisible).toBe(true);

  await languageSwitch.click();
  await page.waitForURL('**/en/', { timeout: 5000 });

  const currentURL = page.url();
  expect(currentURL).toContain('/en/');

  const siteSlogan = page.locator('div.site_slogan a[rel="home"]');
  const sloganText = await siteSlogan.textContent();
  const sloganCleaned = sloganText?.trim();
  expect(sloganCleaned).toContain('National Technical University of Ukraine');
  expect(sloganCleaned).toContain('Igor Sikorsky Kyiv Polytechnic Institute');

  const menu = page.locator('#superfish-main');
  const menuText = await menu.textContent();
  const menuItems = ['University', 'Education', 'Science', 'Admission', 'Student Life'];
  menuItems.forEach(item => {
    expect(menuText).toContain(item);
  });
});
```

```
[chromium] > 9.spec.js:3:1 > Перевірка тексту на англійській сторінці
Перевірка перемикача мови -> видимий
Виконано перемикання на англійську версію
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 (16.5s)
```

Крок 10: Перевірка завантаження зовнішніх ресурсів

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

завантажуються коректно.

- Завдання:

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

```

test('Перевірка завантаження зовнішніх ресурсів', async ({ page }) => {
  const resources = new Map();
  page.on('request', req => {
    resources.set(req.url(), {
      url: req.url(),
      startTime: Date.now(),
      status: null,
      endTime: null
    });
  });
  page.on('response', res => {
    const resource = resources.get(res.url());
    if (resource) {
      resource.endTime = Date.now();
      resource.status = res.status();
    }
  });
  await page.goto('https://kpi.ua');
  await page.waitForTimeout(5000);

  const results = Array.from(resources.values()).map(r => {
    const loadTime = r.endTime ? r.endTime - r.startTime : null;
    return { ...r, loadTime };
  });

  results.forEach(r => {
    const statusText = r.status ? (r.status < 400 ? 'без помилок' : `Помилка (${r.status})`) : 'Невідомо';
    const loadText = r.loadTime ? (r.loadTime > 3000 ? 'повільне' : 'оптимальне') : 'Невідомо';
    console.log(`\nURL: ${r.url}, Час завантаження: ${r.loadTime ? r.loadTime + 'мс' : 'Невідомо'}, Статус: ${r.status ? r.status : 'Невідомо'}, Оцінка: `);
  });

  const errors404 = results.filter(r => r.status === 404);
  const errors500 = results.filter(r => r.status === 500);
  const slowResources = results.filter(r => r.loadTime !== null && r.loadTime > 3000);

```

```

Ресурс: https://kpi.ua/libraries/drupal-superfish/css/superfish.css?soduq8, Час: 11 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/ajax-progress.module.css?soduq8, Час: 4 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/align.module.css?soduq8, Час: 4 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/autocomplete-loading.module.css?soduq8, Час: 4 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/fieldgroup.module.css?soduq8, Час: 5 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/container-inline.module.css?soduq8, Час: 6 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/clearfix.module.css?soduq8, Час: 6 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/details.module.css?soduq8, Час: 7 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/hidden.module.css?soduq8, Час: 7 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/item-list.module.css?soduq8, Час: 16 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/js.module.css?soduq8, Час: 16 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/nowrap.module.css?soduq8, Час: 16 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/position-container.module.css?soduq8, Час: 17 мс, Статус: 200, Оптимально, Без помилок
Ресурс: https://kpi.ua/core/themes/stable9/css/system/components/progress.module.css?soduq8, Час: 17 мс, Статус: 200, Оптимально, Без помилок

```

```

Ресурс: https://www.youtube.com/generate_204?8-s6Ww, Час: 43 мс, Статус: 204, Оптимально, Без помилок
Ресурс: https://www.youtube.com/generate_204?y9s2YA, Час: 44 мс, Статус: 204, Оптимально, Без помилок
Ресурс: https://www.youtube.com/generate_204?j6qi-g, Час: 35 мс, Статус: 204, Оптимально, Без помилок
Ресурс: https://www.youtube.com/generate_204?GPmVIA, Час: 40 мс, Статус: 204, Оптимально, Без помилок
Ресурс: https://www.youtube.com/generate_204?GnEFjg, Час: 42 мс, Статус: 204, Оптимально, Без помилок
Ресурс: https://www.youtube.com/generate_204?QJUPCA, Час: 43 мс, Статус: 204, Оптимально, Без помилок
Ресурс: https://www.youtube.com/generate_204?TBV4nA, Час: 26 мс, Статус: 204, Оптимально, Без помилок
Ресурс: https://www.youtube.com/youtubei/v1/log_event?alt=json, Час: 156 мс, Статус: 200, Оптимально, Без помилок
1 passed (9.6s)

```

Playwright Test for VSCode

✓ Перевірка завантаження зовнішніх ресурсів