МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Івано-Франківський національний технічний університет
нафти і газу

Кафедра інженерії програмного забезпечення

ЛАБОРАТОРНА РОБОТА № 6

з дисципліни

Основи автоматизованого тестування

Виконав: ст. гр. ІП-23-1К
Гнатюк Д. М.

Івано-Франківськ

2025

Тема: Написання тестів для користувацьких сценаріїв

Мета: Навчитися створювати тести для перевірки користувацьких сценаріїв, розібратися в існуючих інструментах для створення тестів та навчитися їх обирати.

Github: https://github.com/DementialDima/Lab1-AutomatedTesting

Було створено новий проєкт на базі Puppeteer та Jest для E2E тестування користувацьких сценаріїв на сайті https://magento.softwaretestingboard.com.

Було реалізовано три користувацькі сценарії:

1. Реєстрація нового користувача — перевіряється можливість успішної реєстрації заповненням усіх обов'язкових полів і отриманням повідомлення про успішну реєстрацію.

2. Пошук товару — перевіряється можливість ввести запит у пошуковий рядок, натиснути Enter та побачити список товарів.

3. Додавання товару до корзини — перевіряється вибір товару, його розміру, кольору, додавання до кошика та підтвердження дії через повідомлення.

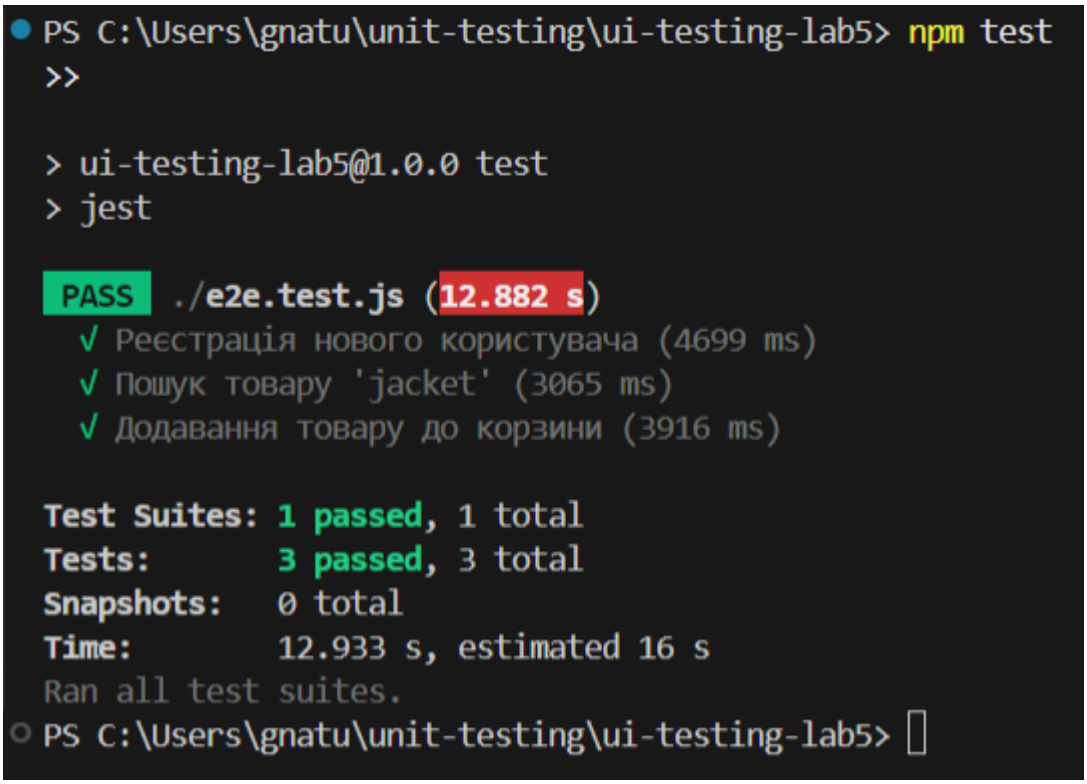
Для уникнення помилок, пов'язаних з таймаутами, для окремих сценаріїв було збільшено час очікування виконання тесту до 15 секунд.

Код тестів:

```
test("Реєстрація нового користувача", async () => {
                                                                    await page.goto("https://magento.softwaretestingboard.com/customer/account/create/");

  await page.type("#firstname", "Іван");
  await page.type("#lastname", "Тестовий");

  const email = `test${Date.now()}@example.com`;
  await page.type("#email_address", email);
  await page.type("#password", "Test1234!");
  await page.type("#password-confirmation", "Test1234!");

  await page.click("button[title='Create an Account']");
  await page.waitForSelector(".message-success, .page-title");

  const success = await page.$eval(".page-title", el => el.textContent.trim());
  expect(success).toMatch(/My Account/i);
}, 15000);

test("Пошук товару 'jacket'", async () => {
  await page.goto("https://magento.softwaretestingboard.com/");

  await page.click(".search");
  await page.type("#search", "jacket");
  await page.keyboard.press("Enter");

  await page.waitForSelector(".product-item");

  const results = await page.$$eval(".product-item", items => items.length);
  expect(results).toBeGreaterThan(0);
});
```

```
    test("Додавання товару до корзини", async () => {
       await  page.goto("https://magento.softwaretestingboard.com/men/tops-men/jackets-
men.html");

      await page.waitForSelector(".product-item");
      await page.click(".product-item .product-item-link");

      await page.waitForSelector("#option-label-size-143-item-166");
      await page.click("#option-label-size-143-item-166"); // Розмір
      await page.click("#option-label-color-93-item-50"); // Колір

      await page.click("#product-addtocart-button");
      await page.waitForSelector(".message-success");

      const cartText = await page.$eval(".message-success", el => el.textContent);
      expect(cartText).toMatch(/You added .* to your shopping cart/);
    }, 15000);
```

Результат виконанння тестів:



Висновок: У ході виконання лабораторної роботи було створено наскрізні тести користувацьких сценаріїв, що охоплюють основні функції вебдодатку з точки зору кінцевого користувача. Було закріплено навички роботи з Puppeteer для імітації дій користувача та Jest як фреймворку для запуску та структурування тестів. Набуті знання дозволяють перевіряти цілісність вебдодатку при зміні або розширенні функціоналу.

Відповіді на питання

1. В чому полягає чуть e2e тестування?

E2E (end-to-end) тестування — це тип тестування, при якому перевіряється повна взаємодія всіх компонентів додатку від початку до кінця. Його мета — переконатися, що всі частини системи працюють разом так, як очікується з точки зору користувача.

2. Чим відрізняється e2e від UI тестування?

UI тестування перевіряє окремі елементи інтерфейсу (наприклад, кнопки, поля вводу).

E2E тестування охоплює повний користувацький сценарій (наприклад, від реєстрації до покупки), включаючи не лише UI, а й інтеграцію з бекендом, БД тощо.

3. Які інструменти для написання e2e тестів вам відомі?

- Puppeteer
- Playwright
- Cypress
- Selenium
- WebdriverIO
- TestCafe

4. Що таке e2e сценарій?

E2E сценарій — це послідовність дій, яка моделює поведінку користувача в реальній ситуації. Наприклад: «Користувач заходить на сайт, реєструється, шукає товар, додає його в корзину та оформлює покупку».

5. З чого можуть складатися e2e сценарії?

- Навігація між сторінками
- Взаємодія з формами (введення даних, кліки)
- Очікування та перевірка результатів (повідомлення, DOM зміни)
- Емуляція складної послідовності дій (логін → дія → лог-аут)

6. Для чого існує e2e тестування?

Щоб перевірити, що:

- Додаток працює правильно як система в цілому
- Не порушено зв’язки між компонентами після змін
- Користувач зможе пройти повний бізнес-сценарій без помилок

7. Охарактеризуйте e2e тестування спираючись на піраміду тестування.

У тестовій піраміді:

- На основі — модульні тести (найбільше, швидкі та дешеві)
- Посередині — інтеграційні тести
- На вершині — E2E тести (найменше, найдорожчі й повільні, але перевіряють найважливіше — повну роботу програми в очах користувача)

Тому E2E тести пишуться обмежено, але стратегічно важливо.