Міністерство освіти і науки України  
Івано-Франківський національний технічний університет нафти і газу

Кафедра ІПЗ

Лабораторна робота №7

Тема: “Побудова базового автоматизаційного рішення”

Виконав:

ст. групи ІП-21-1

Гаврилюк Андрій

Перевірив:

Храбатин Р. І.

Івано-Франківськ

2025

**Мета роботи:** зрозуміти та навчитися будувати прості базові автоматизовані рішення

**Завдання**

**Завдання перше:**

Відносно обраного в попередній роботі сайту розробити автоматизаційне рішення яке включатиме у себе unit, ui, api, та е2е тестування.

Структура рішення може бути довільна, але разом з тим логічна. Стосовно тестів: е2е можна скопіювати з попередньої роботи, оскільки це найгроміздкіша частина і на її розробку пішло б багато часу. Для решти тестів оберіть одну сторінку і пропрацюйте над її тестуванням.

Щодо обсягу кожного з видів – хоча б по пʼять тестів для кожного типу. Для функцій, які ви розробите в межах оптимізації тестів створіть компонентні тести.

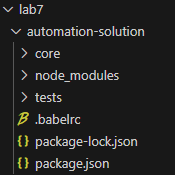
При роботі послуговуйтесь підходами KISS та DRY.

**Завдання друге:**

Завантажити код на власний/навчальний репозиторій GitHub, або інший схожий ресурс та поділитися посиланням.

**Хід роботи**

Для тестування обрано сайт <https://demowebshop.tricentis.com/>, тести типу e2e взяті з попередньої роботи. Структура проекту виглядає наступним чином

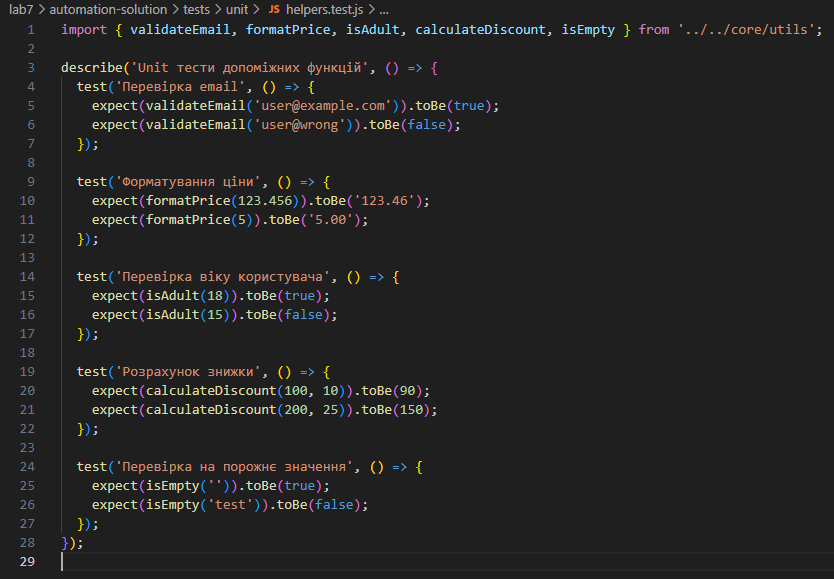
****

Папка **core** — це зручне місце для зберігання загальних утиліт або допоміжних функцій, які можна повторно використовувати в різних частинах свого проекту.

Папка **tests** – це папка з самими тестами, серед яких api, e2e, ui та unit.

**Написання тестів**

**Unit тести:**



Опис тестів:

**validateEmail(email)**

Мета: перевірити, чи рядок відповідає формату email.

Тестові приклади:

* validateEmail("user@example.com") → true
* validateEmail("invalid@") → false

Реалізація: використовує регулярний вираз /\S+@\S+\.\S+/

**formatPrice(price)**

Мета: форматувати число у вигляді ціни з двома знаками після коми.

Тестові приклади:

* formatPrice(15) → "15.00"
* formatPrice(3.7) → "3.70"

**isAdult(age)**

Мета: перевірити, чи людина є повнолітньою (18+).

Тестові приклади:

* isAdult(20) → true
* isAdult(17) → false

**calculateDiscount(price, percent)**

Мета: обчислити суму зі знижкою у відсотках.

Тестові приклади:

* calculateDiscount(100, 10) → 90
* calculateDiscount(200, 25) → 150

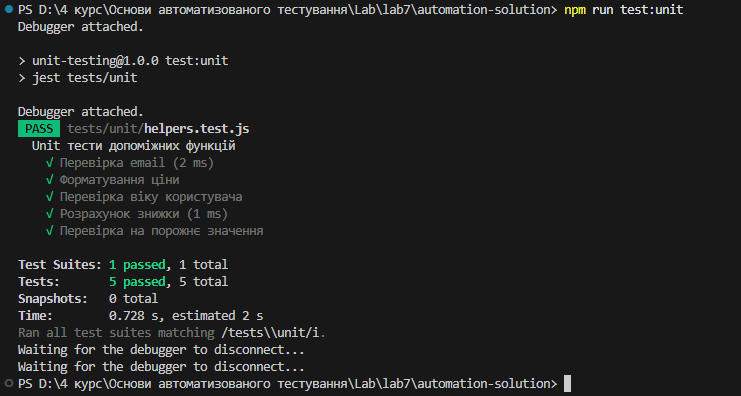
**isEmpty(str)**

Мета: перевірити, чи рядок порожній або складається тільки з пробілів.

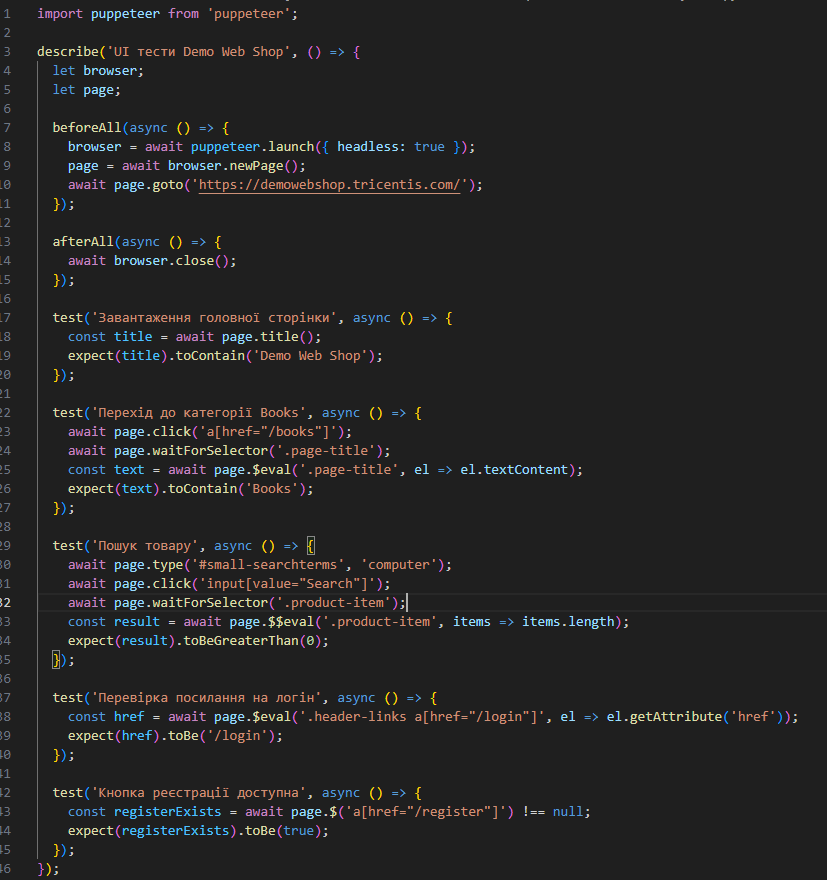
Тестові приклади:

* isEmpty("") → true
* isEmpty(" ") → true
* isEmpty("text") → false

**Виконання тестів:**

****

**UI тести**

****

Опис тестів:

**Тест: Відкриття головної сторінки**

Мета: переконатись, що головна сторінка сайту працює.

Очікування: статус-код 200, заголовок містить "Demo Web Shop"

Важливо: дозволяє перевірити доступність сайту перед глибшими діями.

**Тест: Перехід до категорії "Books"**

Мета: перевірити навігацію по сайту.

Очікування: завантаження сторінки /books, наявність тексту "Books".

**Тест: Пошук товару**

Мета: перевірити пошук по сайту.

Кроки: ввести "computer" у поле пошуку, натиснути кнопку.

Очікування: відкриється сторінка з результатами, що містять "computer".

**Тест: Посилання на логін**

Мета: перевірити, що неавторизовані користувачі бачать посилання на вхід.

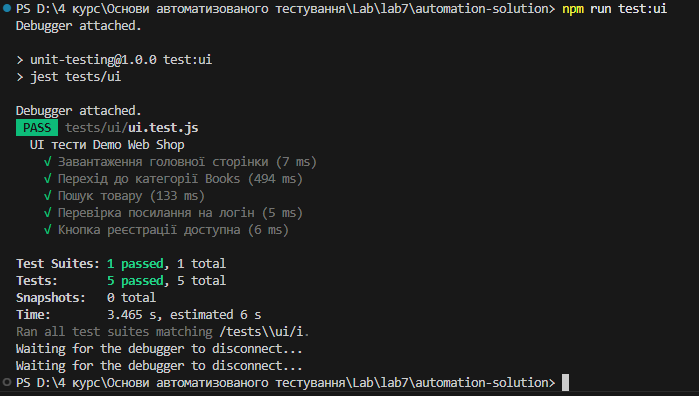
Очікування: у DOM присутній елемент з посиланням /login.

**Тест: Кнопка реєстрації**

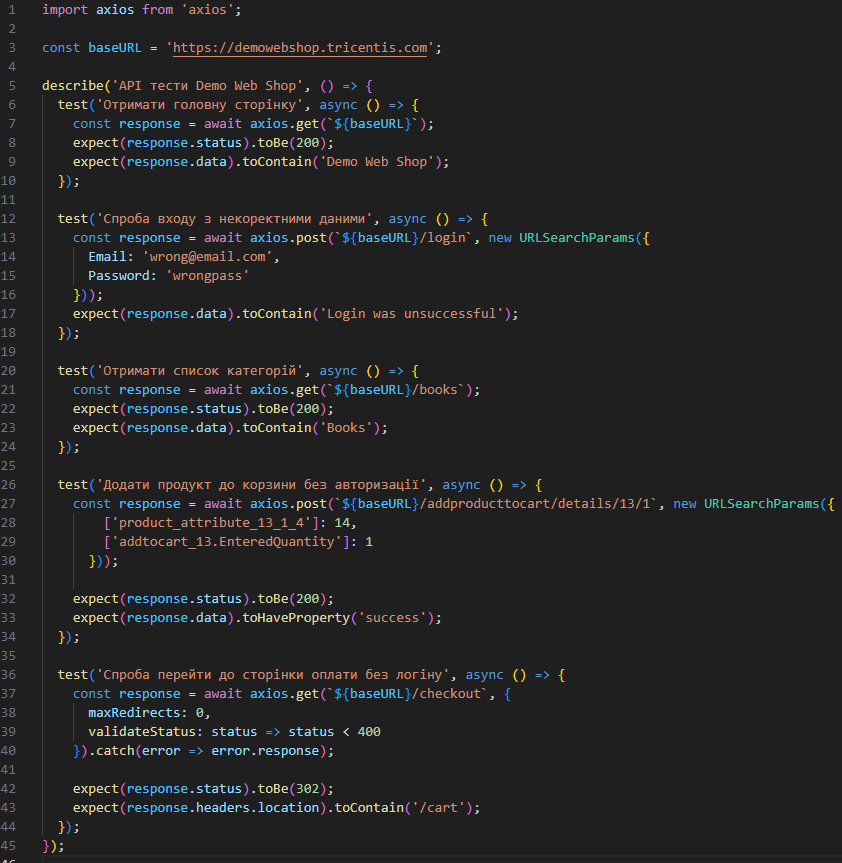
Мета: перевірити доступність кнопки реєстрації.

Очікування: кнопка активна (не disabled) і має правильний текст.

**Виконання тестів**

****

**API тести**

****

Опис тестів:

**GET / — Головна сторінка**

Мета: перевірити, чи сайт доступний.

Очікування: status 200, data містить Demo Web Shop.

**POST /login — Вхід з некоректними даними**

Мета: перевірити захист від неправильних логінів.

Тіло запиту: email і password (неправильні).

Очікування: data містить "Login was unsuccessful".

**GET /books — Сторінка категорії**

Мета: перевірити, чи існує категорія Books.

Очікування: status 200, data містить "Books".

**POST /addproducttocart/details/13/1 — Додати товар до корзини**

Мета: перевірити, чи можна додати товар без авторизації.

Тіло запиту: дані варіантів товару та кількість.

Очікування: data.success → true (або присутній об'єкт success).

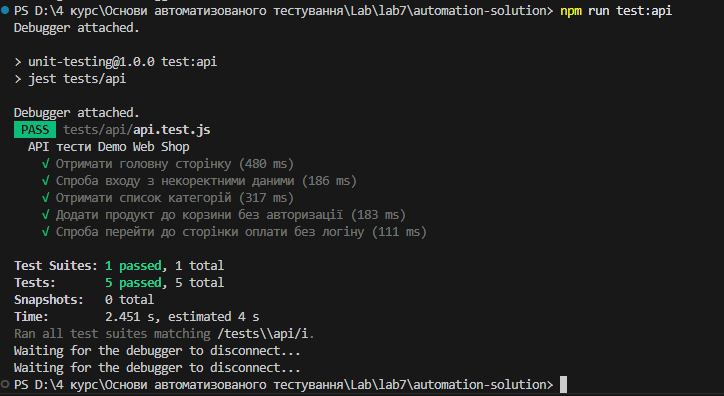
**GET /checkout — Перехід до оплати без логіну**

Мета: перевірити, що користувача не пускає на сторінку оформлення без входу.

Очікування: редірект на /cart.

Реальність: редірект на /cart.

**Виконання тестів**

****

**E2E тести**



Опис тестів:

**1. Вхід користувача**

**Мета:** перевірити, чи може користувач успішно увійти в систему з правильними обліковими даними.

**Кроки:**

1. Переходимо на сторінку входу (/login).
2. Вводимо email qwe@example.com і пароль password123.
3. Натискаємо кнопку входу.
4. Очікуємо завершення завантаження сторінки.
5. Перевіряємо, що з'явилось посилання на **вихід із акаунту** (/logout), тобто користувач увійшов успішно.

**Очікуваний результат:** на сторінці присутній елемент a[href="/logout"].

**2. Додавання подарункової картки у кошик з заповненням форми**

**Мета:** перевірити, що подарункову картку можна додати в кошик після заповнення полів форми.

**Кроки:**

1. Відкриваємо головну сторінку магазину.
2. Натискаємо на посилання подарункової картки (/25-virtual-gift-card).
3. Заповнюємо обов'язкові поля: ім’я та email одержувача.
4. Натискаємо кнопку “Додати в кошик”.
5. Чекаємо відповідь від сервера (/addproducttocart).
6. Очікуємо оновлення індикатора кошика (елемент .cart-qty).
7. Зчитуємо значення індикатора та перевіряємо, що там є 1.

**Очікуваний результат:** у кошику відображається одна одиниця товару.

**3. Пошук товару**

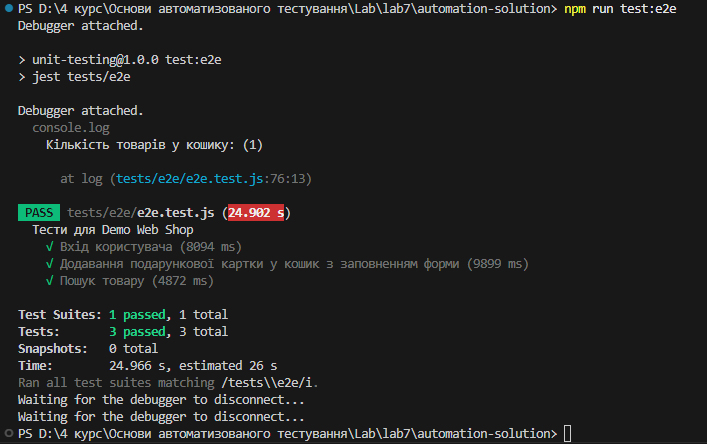
**Мета:** перевірити, що функція пошуку на сайті працює коректно та повертає результати.

**Кроки:**

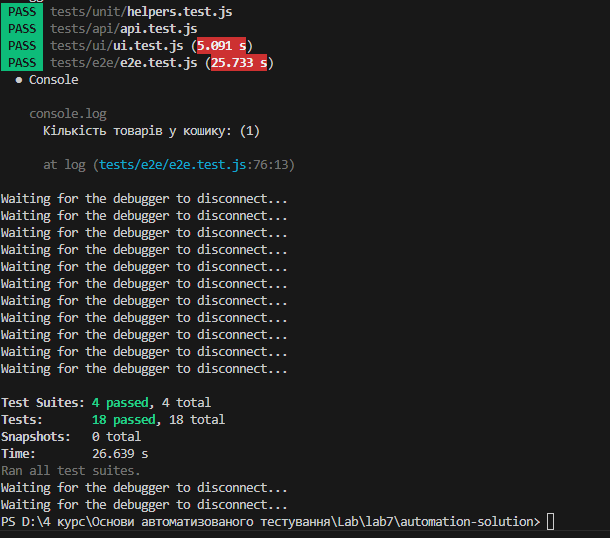
1. Відкриваємо головну сторінку.
2. Вводимо слово “Laptop” у поле пошуку.
3. Натискаємо кнопку пошуку.
4. Очікуємо, поки з’являться результати (.product-item).
5. Рахуємо кількість знайдених товарів.

**Очікуваний результат:** знайдено **хоча б один** товар (кількість елементів > 0).

**Виконання тестів**

****

**Виконання всіх тестів одночасно:**

****

Посилання на github: <https://github.com/B0neS69/LAB_Test>

**Висновок:** У цій лабораторній роботі були створені автоматизовані тести для перевірки API Demo Web Shop. Було протестовано завантаження сторінок, спробу входу з некоректними даними, додавання товару до кошика та обробку редиректу при переході до оплати без авторизації, із використанням таких видів тестування: UNIT, UI, API та E2E.