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

Кафедра ІПЗ

Лабораторна робота №8

Тема: “Огляд сучасного популярного автоматизованого інструментарію”

Виконав:

ст. групи ІП-21-1

Гаврилюк Андрій

Перевірив:

Храбатин Р. І.

Івано-Франківськ

2025

**Мета роботи:** дослідити та порівняти існуючі сучасні інструменти для створення автоматизованих рішень на основі JavaScript.

**Завдання**

**Завдання перше:**

Згідно варіанту провести невеличке дослідження можливостей заданих інструментів. Дослідження повинно включати наступні пункти: сильні сторони, слабкі сторони, типові задачі які інструмент вирішує без проблем (основне призначення). Порівняння оформити у вигляді таблички:

|  |  |  |  |
| --- | --- | --- | --- |
| **Інструмент** | **Сильна сторона** | **Слабка сторона** | **Типові задачі** |

**Завдання друге:**

Створити три UI тести використовуючи кожен з інструментів. За основу тестування можете брати попередні ресурси з якими ви вже добре знайомі. Проведіть аналіз вашого досвіду роботи з цими інструментами.

**Завдання третє:**

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

**Хід роботи**

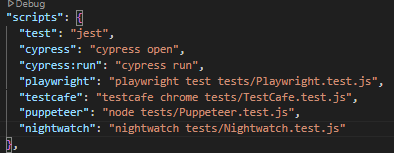
Для тестування обрано сайт <https://demowebshop.tricentis.com/>,

|  |  |  |
| --- | --- | --- |
| Cypress & Playwright | TestCafe & Testim.io | Puppeteer & Nightwatch.js |

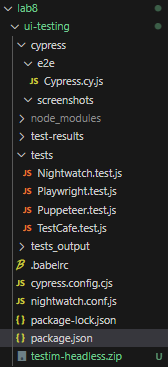
Для того, щоб написати тести використовуючи різні бібліотеки, нам необхідно їх встановити:

|  |  |
| --- | --- |
| **Інструмент** | **Команда встановлення** |
| Cypress | npm install --save-dev cypress |
| Playwright | npm install --save-dev @playwright/test + npx playwright install |
| TestCafe | npm install --save-dev testcafe |
| Puppeteer | npm install --save-dev puppeteer |
| Nightwatch | npm install --save-dev nightwatch |

Також для запуску необхідно внести зміни у файл package.json:

****

Структура проекту виглядає наступним чином:

****

**Аналіз досвіду використання інструментів:**

1. Cypress  
Дуже інтуїтивно зрозумілий інструмент. Легко писати тести. Все працює швидко і наочно. Найзручніше дебажити через вбудовану консоль у браузері.

2. Playwright  
Відмінно підходить для кросбраузерного тестування. Трохи складніше за Cypress у налаштуванні, але дає більше контролю над процесом тестування.

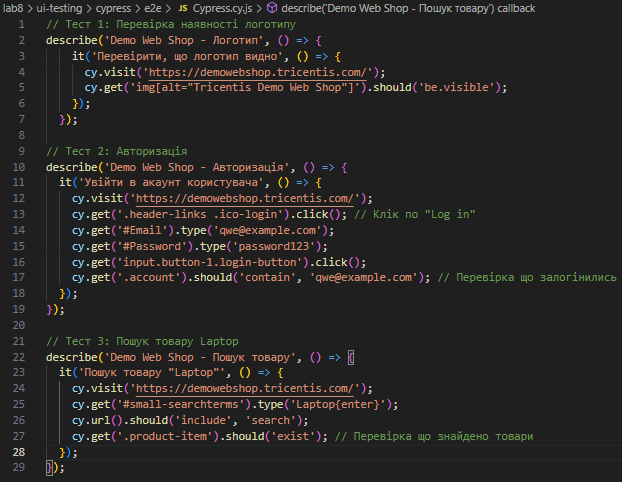
3. TestCafe  
Простий у використанні інструмент. Не потребує WebDriver. Однак, можливості кастомізації тестів трохи обмежені у порівнянні з Playwright.

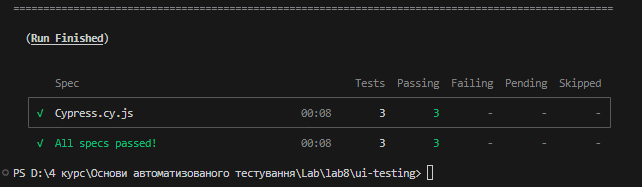
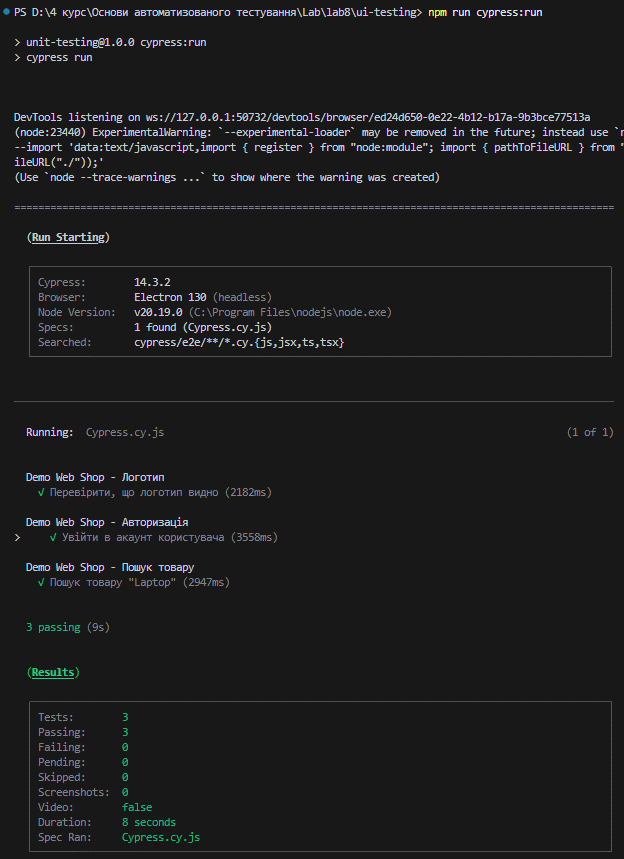
4. Testim  
Інструмент з фокусом на автоматизацію без коду або з мінімальним кодом. Зручний для швидких перевірок, але менш гнучкий при написанні складних сценаріїв.

5. Puppeteer  
Низькорівневий контроль над браузером. Дає дуже точну симуляцію дій користувача. Потрібно більше коду писати вручну, але можна реалізувати майже будь-яку логіку.

6. Nightwatch.js  
Добрий варіант для інтеграції в CI/CD процеси. Спочатку конфігурація здається складною, але потім працювати досить зручно, особливо для великих проектів.

**Cypress тести**

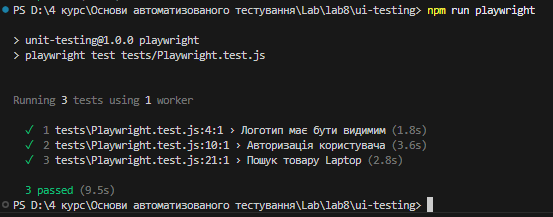


Результат виконання

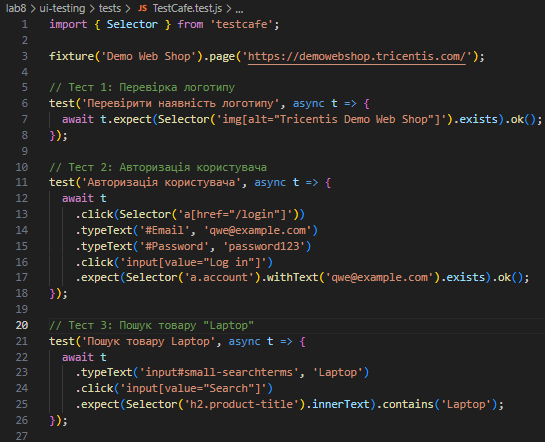
**Playwright тести**

****

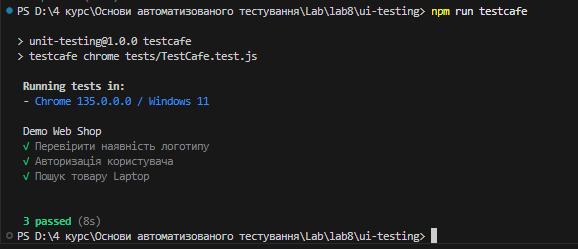
Результат виконання

****

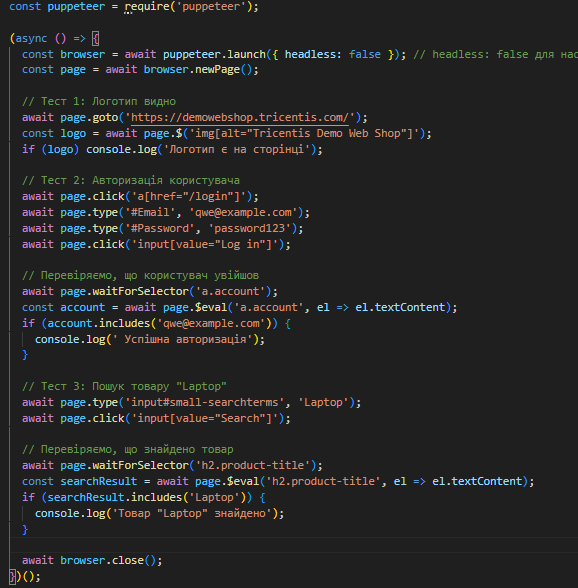
**TetsCafe тести**

****

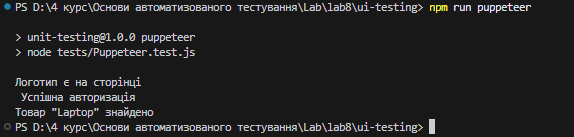
Результат виконання

****

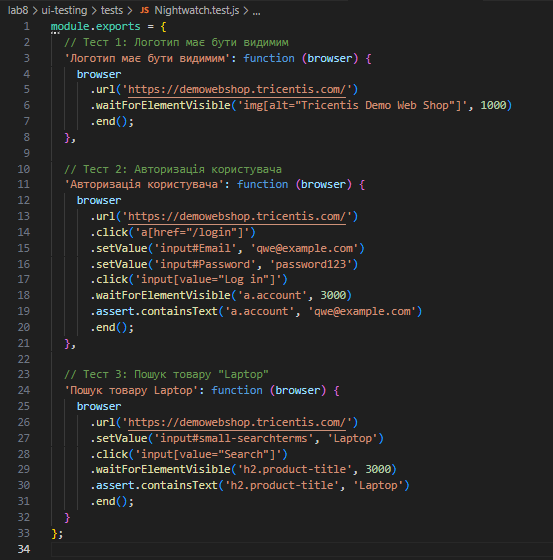
**Puppeteer тести**

****

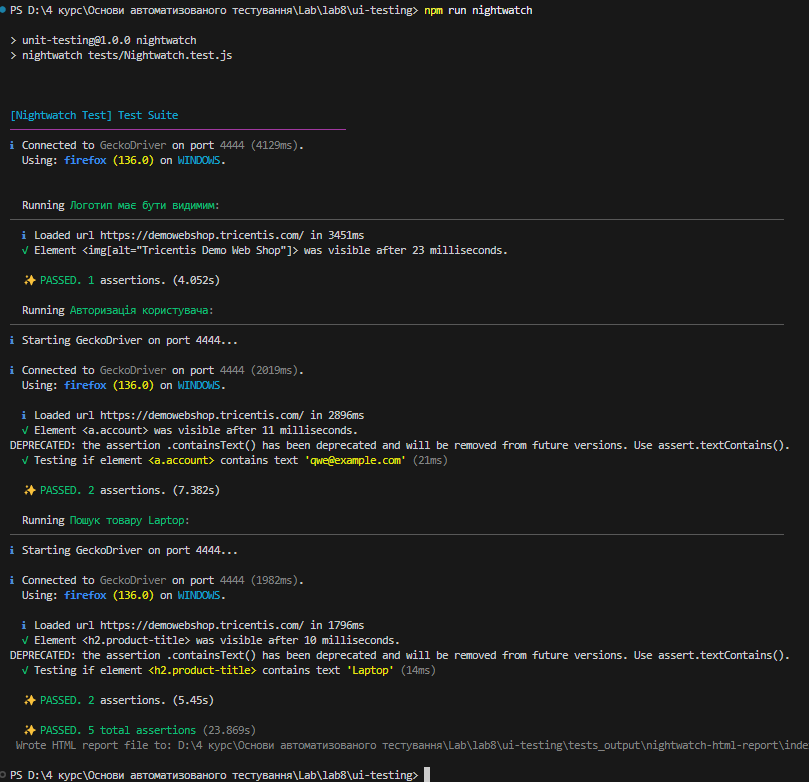
Результат виконання

****

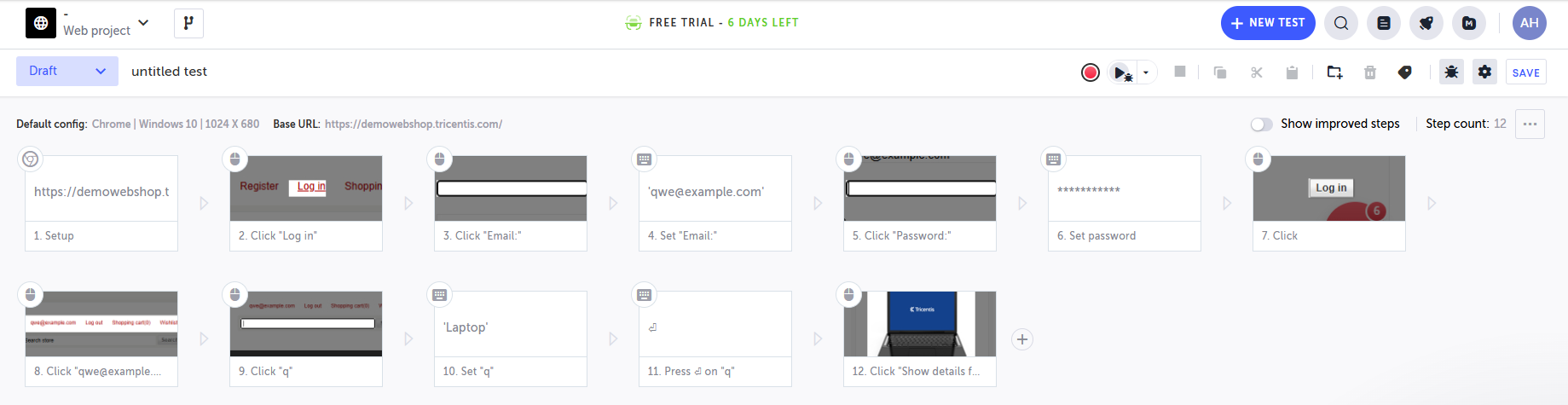
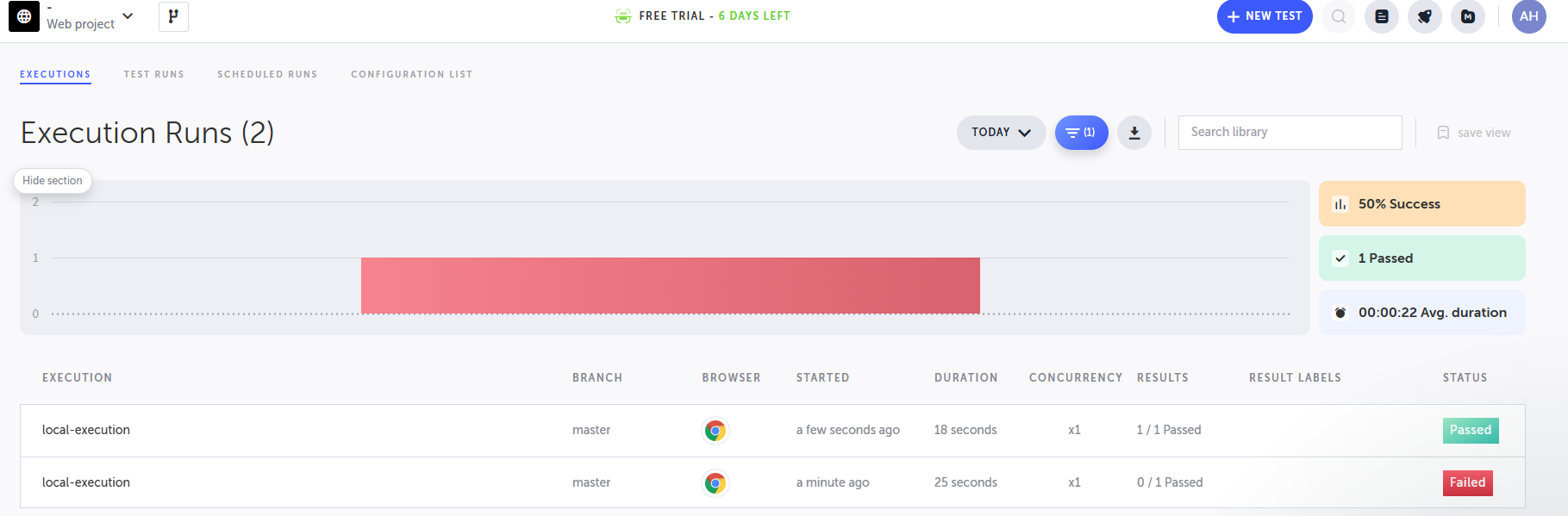
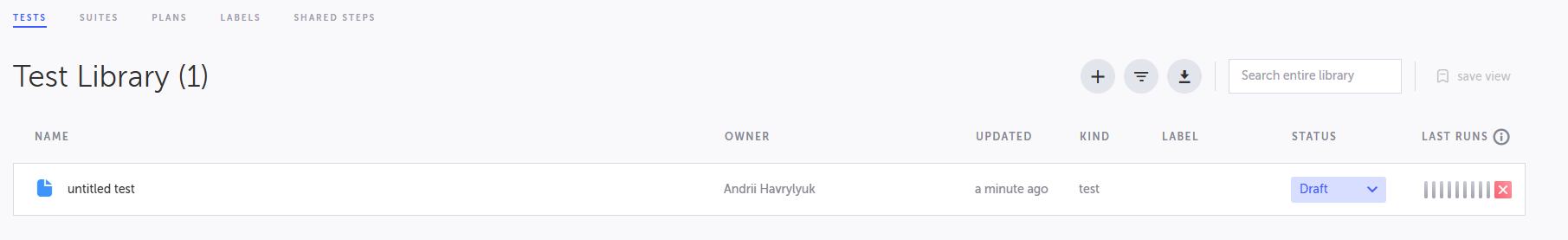
**Nightwatch тести**

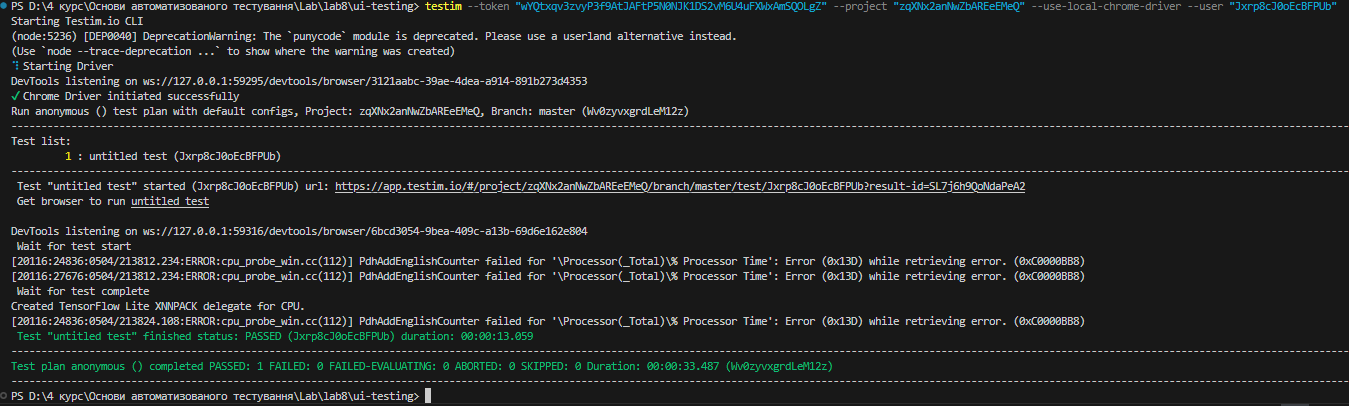
****

Результат виконання

****

**Testim тести**

****



testim --token "token" --project "projectid" --use-local-chrome-driver --user "id"

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.