

Практична робота 6

Тема: Розробка автоматизованих сценаріїв тестування за допомогою Playwright

Дисципліна: Тестування, верифікація та валідація ПЗ

Мета роботи: Освоїти основний функціонал інструменту Playwright для автоматизації тестування вебдодатків. Навчитися ефективно використовувати локатори, виконувати дії над елементами, застосовувати різні типи перевірок (assertions) та використовувати інструменти налагодження (Trace Viewer, UI Mode).

Необхідне забезпечення:

- Node.js (LTS версія)
- IDE (Visual Studio Code рекомендовано + розширення "Playwright Test for VSCode")
- Встановлений Playwright у проєкті

Студентам пропонується повторити:

- Різницю між синхронним та асинхронним кодом у JS/TS (`await/async`).
- Типи селекторів: Text, CSS, XPath, та рекомендовані Playwright локатори (Role, Label, TestId).
- Життєвий цикл тесту (`test`, `describe`, `beforeEach`).

Як об'єкт тестування використовуйте демонстраційний ресурс:

- <https://st-crm-test-rx6au.ondigitalocean.app/>
- Доступ: login: admin@test.com , pass: 1234

Блок 1: Налаштування та Базові сутності (Tasks 1-4)

Завдання 1. Конфігурація та Login State

- **Контекст:** CRM вимагає авторизації для кожної дії. Щоб не логінитися в кожному тесті, налаштуйте збереження стану.
- **Дія:**
 1. Налаштувати `playwright.config.ts` для запуску в Chromium та Firefox.
 2. Створити `global-setup.ts` або використати `test.use({ storageState: 'auth.json' })`, щоб авторизуватися один раз і зберегти куки/токени.
 3. Всі наступні тести мають використовувати цей стан.

Завдання 2. Створення Компанії (Complex Forms)

- **Дія:** Перейти в розділ "Companies" -> Натиснути "Create".
- **Сценарій:** Заповнити форму створення нової компанії.
 - Ввести унікальну назву (використати `Date.now()` або бібліотеку `faker` для унікальності, напр. "Horn & Hooves " + timestamp).
 - Вибрати **Company Type** з випадаючого списку (Select/Dropdown).
 - Заповнити контактні дані.
 - Зберегти.

Завдання 3. Перевірка даних (Assertions)

Дія: Після створення компанії (із Завдання 2) система зазвичай перенаправляє на сторінку списку або детального перегляду.

Перевірка:

1. Перевірити, що URL змінився (наприклад, містить ID компанії).
2. Знайти створену компанію в таблиці або на сторінці перегляду.
3. Перевірити, що назва компанії відповідає тій, що була введена
(`expect(locator).toHaveText(...)`).

Завдання 4. Створення Користувача (Contacts) з прив'язкою

Дія: Перейти в розділ "Contacts" (або Users).

Сценарій: Створити нового контакту.

Блок 2: Робота з даними та фільтрми (Tasks 5-8)

Завдання 5. Фільтрація та Пошук

- **Дія:** Перейти до загального списку компаній.
- **Сценарій:**
 1. Ввести в поле пошуку назву існуючої компанії.
 2. Дочекатися оновлення таблиці.
 3. Перевірити, що в таблиці відображається тільки один рядок (або відповідні рядки).
 4. Очистити фільтр і перевірити, що список знову повний.

Завдання 6. Редагування та UI Mode (Trace)

- **Дія:** Відкрити картку будь-якої компанії.
- **Сценарій:** Змінити назву компанії або статус. Зберегти.
- **Налагодження:** Запустити цей тест із прапором `--ui` (UI Mode) і простежити покроково ("Time travel"), як змінюється DOM під час збереження.

Завдання 7. Видалення в Кошик (Trash)

Дія: Знайти компанію в списку.

Сценарій:

1. Натиснути кнопку видалення (Delete/Archive).
2. Підтвердити дію у спливаючому вікні (якщо є).
3. Перевірити, що компанія зникла із загального списку ("Active").

Завдання 8. Робота з "Трашами" (Trash/Restore)

Дія: Перейти в розділ "Trash" (Кошик/Архів).

Сценарій:

- Знайти компанію, видалену в Завданні 7.
- Натиснути "Restore" (Відновити).
- Повернутися в список "Companies" і переконатися, що компанія знову з'явилася там.
- *Це перевіряє повний цикл: Create -> Delete -> Restore.*

Блок 3: Активність та Складні елементи (Tasks 9-11)

Завдання 9. Перевірка "Activity Log" (Історія змін)

Контекст: CRM зазвичай записує, хто і коли змінив сутність.

Сценарій: Знайти таб або блок "Activity".

Перевірка: Знайти запис про створення або редагування. Використати `locator.filter({ hasText: ... })` для пошуку конкретної події в історії.

Завдання 10. Емуляція мобільного відображення

Дія: Перевірити таблицю компаній на емуляторі iPad (`use: { ...devices['iPhone 12'] }`).

Сценарій: на довільний вибір (<5 кейсів).

Завдання 11. Сортування таблиці (JS Parsing)

Сценарій:

1. Зібрати дані з колонки в масив.
2. Перевірити програмно, чи дані відсортовані (за датою або алфавітом).

Блок 4: Архітектура та Мережа (Tasks 12-14)

Завдання 12. Interception (Прискорення тестів)

- **Сценарій:** CRM може підвантажувати важкі файли.
- **Дія:** Використати `page.route` для блокування завантаження .
- **Мета:** Переконалися, що функціонал створення компанії працює навіть якщо картинки "відпали".

Завдання 13. API Testing (Hybrid Approach)

- **Контекст:** Створення компанії через UI — це довго.
- **Дія:**
 - Використати `request.post('/api/companies', ...)` для швидкого створення компанії через API (потрібно підглянути endpoint у вкладці Network браузера).
 - В UI тесті просто відкрити сторінку цієї компанії за її ID.
 - *Це демонструє вміння комбінувати API та UI тести.*

Завдання 14. Page Object Model (POM) для CRM

- **Завдання:** Створити структуру класів для CRM.
- **Файли:**
 - `pages/LoginPage.ts`: методи `login(user, pass)`.
 - `pages/CompaniesPage.ts`: методи `createCompany(data)`, `searchCompany(name)`, `deleteCompany(name)`.
- **Рефакторинг:** Переписати тест створення та видалення компанії (Завдання 2 та 7), використовуючи ці об'єкти.