

Testy UI

Celem zajęć będzie stworzenie frameworku do testów sprawdzających UI naszej aplikacji.

Do testów użyjemy narzędzia Microsoftu: [Playwrighta](#)



Inne narzędzia do e2e testów front-endu: Cypress, Selenium, webdrive.io

Projekt i instalacja playwright

Komendy do wykonania:

```
npm init
npm init playwright@latest
npx playwright test --headed --project=chromium
npx playwright show-report
```

Do naszych testów użyjemy aplikacji z przedmiotu *Front-end development*.

Zadanie 1

Napisz prosty test który sprawdzi sprawdzi funkcjonalność wyszukiwania Pokemonów.

Test powinien:

1. przejść na stronę z wyszukiwaniem
2. sprawdzić, że na stronie znajduje się 200 pokemonów
3. wyszukać po frazie "pid" i sprawdzić, że zostały wyświetlone 4 pokemony
4. wyszukać po frazie "pikachu" i sprawdzi, że został wyświetlony tylko jeden pokemon

Podpowiedzi:

- Przerobimy już przygotowany przykładowy test playwrighta
- aby znaleźć przycisk z tekstem możemy użyć: `await page.getByText TEKST).click()`
- aby wypełnić pole z id możemy użyć: `await page.locator('#ID').fill(WARTOSC)`
- aby sprawdzić czy na stronie istnieje dokładnie 10 obiektów z klasą: `await expect(page.locator('.className')).toHaveCount(10)`

Zadanie 2

Wykonaj refaktor testu do [POM](#)

Zadanie 3

Napisz nowy test który sprawdzi czy funkcjonalność dodawania do ulubionych działa.

Test Powinien:

1. otworzyć stronę wyszukiwania pokemonów
2. kliknie w "charizarda"
3. doda pokemona do ulubionych
4. przejdzie do ulubionych
5. sprawdzi że Charizard został dodany
6. usunie Charizarda z listy ulubionych
7. sprawdzi że lista ulubionych jest pusta

Podpowiedzi:

- stwórz nowy **page object model** - *favorite-page.ts*
- dodaj w modelu listę z obiektami `favoriteList` oraz `removeButton`
- dodaj w modelu metodę do sprawdzania ilość ulubionych pokemonów
- korzystając z modelu napisz test