

# FINÁLNÍ PROJEKT

## č.2



# ENGETO

Autor: Marie Toncarová  
Datum: 06.04.2025

# Obsah

ZADÁNÍ.....	3
Testovací scénáře a exekuce testů.....	4
Testovací prostředí a fixtures.....	5
1. Otestování načtení webové stránky.....	6
a. Cíl testu:.....	6
b. Testovací kroky:.....	6
c. Očekávaný výsledek:.....	6
d. Kód Python.....	6
2. Otestování ověření přítomnosti konkrétního elementu na stránce.....	7
a. Cíl testu:.....	7
b. Testovací kroky:.....	7
c. Očekávaný výsledek:.....	7
d. Kód Python.....	7
3. Otestování přesměrování na jinou stránku.....	8
a. Cíl testu:.....	8
b. Testovací kroky:.....	8
c. Očekávaný výsledek:.....	8
d. Kód Python.....	8

# ZADÁNÍ

## **Zadání**

Napište tři automatizované testy pomocí frameworku Playwright.

## **Postup**

1. Vyberte si libovolnou webovou stránku, kterou chcete otestovat. (např. Engeto.cz).
2. Vytvořte soubor Python (.py), ve kterém napíšete tři automatizované testy pro danou stránku.
3. Pro zjednodušení psaní testů importujte do projektu plugin pytest-playwright.

**Pokud si nejste jisti, neváhejte se obrátit na lektora :)**

*P.S. Pokud se někdo cítí jistěji v jiném programovacím jazyce, může ho použít. Jedinou podmínkou je použití frameworku Playwright.*

# Testovací scénáře a exekuce testů

V rámci této práce byly provedeny testy funkčnosti webové stránky <https://engeto.cz/>. Testování probíhalo na operačním systému Windows 11, verze 23H2, za využití Visual Studio Code jako vývojového prostředí. K implementaci testů byl použit programovací jazyk Python, doplněný o knihovny pytest a Playwright pro automatizované testování webových aplikací.

Pro každý testovací scénář byla připravena detailní tabulka, která zahrnuje jednotlivé kroky testování a výsledky jejich provedení. Každý test je v dokumentu podrobně popsán, včetně popisu vykonání a očekávaného chování systému.

# Testovací prostředí a fixtures

Pro účely testování byly vytvořeny tři fixtures pro zajištění správného fungování testů. Tyto fixtures se postarají o otevření prohlížeče, zavření cookies okna a správné připravení testovací stránky.

Fixture 1: browser

Fixture pro spuštění prohlížeče na základě Playwright. Tento fixture inicializuje prohlížeč v režimu `headless=False` (pro zobrazení akce) a s nastavením zpomalení akce (`slow_mo=1000`). Po dokončení testu prohlížeč zavře.

```
import pytest
from playwright.sync_api import sync_playwright
@pytest.fixture()
def browser():
    with sync_playwright() as playwright:
        browser = playwright.chromium.launch(headless=False, slow_mo=1000)
        yield browser
        browser.close()
@pytest.fixture()
def page(browser):
    page = browser.new_page()
    yield page
    page.close()
@pytest.fixture(scope="function")
def close_cookies(page):
    # Otevře stránku a zavře cookies okno
    page.goto("https://engeto.cz/")
    # Pokusíme se najít tlačítko pro zavření cookies okna
    cookies_button = page.locator("#cookiescript_reject")
    # Počkáme, dokud bude cookies okno viditelné
    try:
        cookies_button.wait_for(state="visible", timeout=10000) # Časový
limit 10 sekund
        cookies_button.click()
        # Počkáme, dokud se cookies okno nezmizí
        page.wait_for_selector("#cookiescript_injected", state="detached",
timeout=10000) # Časový limit 10 sekund
    except Exception as e:
        print(f"Cookies okno nešlo zavřít: {e}")
    yield page
```

## 1. Otestování načtení webové stránky

### a. Cíl testu:

Cílem prvního testovacího scénáře je ověřit, že webová stránka <https://engeto.cz/> se načte správně a obsahuje očekávaný text v hlavním nadpisu.

### b. Testovací kroky:

Prohlížeč se spustí a načte stránku <https://engeto.cz/>. Následně se zavře pop-up okno, přičemž uživatel odmítne souhlas s cookies. Poté se ověří, zda je v hlavním nadpisu (element h1) zobrazen text „*Staň se novým IT talentem*“. Pokud text odpovídá očekávané hodnotě, test je považován za úspěšný.

### c. Očekávaný výsledek:

Na stránce by měl být zobrazen nadpis „*Staň se novým IT talentem*“, což potvrzuje, že stránka byla načtena správně.

### d. Kód Python

```
# Test pro ověření, že stránka se správně načte
def test_page_load(close_cookies):
    page = close_cookies # Používáme fixture close_cookies
    # Zkontrolujeme, že text v h1 odpovídá očekávanému textu
    page_title = page.inner_text('body > main > div:nth-child(1) > div >
div > h1')
    # Nahradíme non-breaking spaces za běžné mezery
    page_title = page_title.replace('\xa0', ' ').strip()
    assert page_title == "Staň se novým IT talentem"
```

Test ID	Testovací scénář	Testovací kroky	Očekávaný výsledek	Skutečný výsledek	Výsledek testu
1	Otestování načtení webové stránky	1. Otevřít aplikaci Visual Studio Code 2. napsat kód Python v příslušném souboru, v tomto případě se jmenem: test_engeto.py 3. spustit code Python pomocí python -m pytest test_engeto.py 4. vyčkat na vyhodnocení testu	Test bude vyhodnocen jako úspěšný	Test je vyhodnocen jako úspěšný	Pass

## 2. Otestování ověření přítomnosti konkrétního elementu na stránce

### a. Cíl testu:

Cílem tohoto testovacího scénáře je ověřit, že na webové stránce <https://engeto.cz/> je přítomný navigační prvek s názvem „Kurzy“.

### b. Testovací kroky:

Prohlížeč se spustí a načte stránku <https://engeto.cz/>. Následně se zavře pop-up okno, přičemž uživatel odmítne souhlas s cookies. Poté se ověří, zda zda je v navigačním menu přítomen prvek (tlačítko) s názvem „Kurzy“. Pokud text odpovídá očekávané hodnotě, test je považován za úspěšný.

### c. Očekávaný výsledek:

Na stránce by měl být zobrazen prvek „Kurzy“ v navigačním menu, což potvrzuje, že stránka byla správně načtena a obsahuje očekávaný element.

### d. Kód Python

```
# Test pro ověření přítomnosti konkrétního elementu na stránce
def test_element_present(close_cookies):
    page = close_cookies # Používáme fixture close_cookies
    # Ověříme, že v navigaci webové stránky je prvek s textem "Kurzy"
    kurzy_element = page.locator('#top-menu > li.area-kurzy.menu-
item.menu-item-type-post_type.menu-item-object-page.menu-item-has-
children.children-items-type-row > a')
    assert kurzy_element.inner_text() == "Kurzy"
```

Test ID	Testovací scénář	Testovací kroky	Očekávaný výsledek	Skutečný výsledek	Výsledek testu
2	Existence konkrétního elementu	1. Otevřít aplikaci Visual Studio Code 2. napsat kód Python v příslušném souboru, v tomto případě se jménem: test_engeto.py 3. spustit code Python pomocí python -m pytest test_engeto.py 4. vyčkat na vyhodnocení testu	Test bude vyhodnocen jako úspěšný	Test je vyhodnocen jako úspěšný	Pass

### 3. Otestování přesměrování na jinou stránku

#### a. Cíl testu:

Cílem tohoto testovacího scénáře je ověřit, že při kliknutí na navigační prvek „Kurzy“ na webové stránce <https://engeto.cz/> dojde k přesměrování na správnou stránku, konkrétně na adresu <https://engeto.cz/prehled-kurzu/>.

#### b. Testovací kroky:

Prohlížeč se spustí a načte stránku <https://engeto.cz/>. Následně se zavře pop-up okno, přičemž uživatel odmítne souhlas s cookies. Poté se klikne na navigační prvek s názvem „Kurzy“. Pokud text odpovídá očekávané hodnotě, test je považován za úspěšný. Po kliknutí se počká na přesměrování a ověří se, že stránka byla přesměrována na URL <https://engeto.cz/prehled-kurzu/>. Pokud URL odpovídá očekávané hodnotě, test je považován za úspěšný.

#### c. Očekávaný výsledek:

Po kliknutí na prvek „Kurzy“ dojde k přesměrování na stránku <https://engeto.cz/prehled-kurzu/>, což potvrzuje správnou funkci přesměrování.

#### d. Kód Python

```
# Test pro ověření přesměrování na jinou stránku
def test_redirect(close_cookies):
    page = close_cookies # Používáme fixture close_cookies
    # Klikneme na odkaz "Kurzy"
    page.click('text="Kurzy"')
    # Počkáme na přesměrování na správnou URL
    page.wait_for_url("https://engeto.cz/prehled-kurzu/")
    # Ověříme, že URL je správně přesměrována
    assert page.url == "https://engeto.cz/prehled-kurzu/"
```

Test ID	Testovací scénář	Testovací kroky	Očekávaný výsledek	Skutečný výsledek	Výsledek testu
3	Přesměrování na jiné url	1. Otevřít aplikaci Visual Studio Code	Test bude vyhodnocen jako úspěšný	Test je vyhodnocen jako úspěšný	Pass
		2. napsat kód Python v příslušném souboru, v tomto případě se jménem: test_engeto.py			
		3. spustit code Python pomocí python -m pytest test_engeto.py			
		4. vyčkat na vyhodnocení testu			