



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA
z siedzibą w Rzeszowie**

*Projekt
Automatyczne testy funkcjonalne strony wsiz.edu.pl*

*Prowadzący:
dr inż. Leszek Puzio*

*Autor:
Piotr Kujda 64874*

*Kierunek:
Informatyka Niestacjonarne*

Rzeszów, 2025 r.

Spis treści

| | |
|--|-----------|
| 1. Wstęp..... | 3 |
| 1.1 Cel projektu | 3 |
| 1.2 Zakres projektu..... | 3 |
| 1.3 Środowisko i narzędzia..... | 4 |
| 1.4 Identyfikacja elementów do testowania | 5 |
| 2. Struktura projektu..... | 6 |
| 3. Opis przypadków testowych..... | 7 |
| 3.1 Tabela przypadków testowych | 7 |
| 3.2. test_homepage_basic.py | 8 |
| 3.3. test_language_switch.py..... | 12 |
| 3.4. test_logo_redirect.py | 15 |
| 3.5. test_search.py | 17 |
| 3.6. test_top_bar_links.py..... | 19 |
| 3.7. Uruchamianie testów | 23 |
| 4. Propozycja usprawnień i naprawy napotkanych błędów | 24 |
| 5. Podsumowanie i wnioski..... | 25 |

1. Wstęp

1.1 Cel projektu

Celem projektu było przygotowanie i uruchomienie zestawu automatycznych testów interfejsu użytkownika dla strony internetowej Wyższej Szkoły Informatyki i Zarządzania w Rzeszowie dostępnej pod adresem <https://wsiz.edu.pl>. Testy zostały wykonane z wykorzystaniem biblioteki Selenium WebDriver w języku Python oraz frameworka pytest, co pozwoliło zautomatyzować czynności, które standardowo wykonywane byłyby manualnie w przeglądarce.

Projekt skupiał się na funkcjach widocznych i dostępnych dla użytkownika końcowego, bez ingerencji w zaplecze serwisu czy API. W szczególności sprawdzono działanie głównej nawigacji, linków kierujących do zewnętrznych usług uczelni, wyszukiwarki, przełącznika wersji językowych oraz odnośnika w postaci logo prowadzącego do strony głównej. Testy odwzorowują rzeczywiste zachowania użytkownika, takie jak kliknięcia kursorem, zmiana karty, przewijanie ekranu czy wpisywanie tekstu w pola formularzy.

W efekcie powstał kompletny zestaw skryptów, który umożliwia powtarzaną i szybką weryfikację kluczowych elementów strony. Zautomatyzowanie tych czynności pozwala łatwiej wykrywać błędy po zmianach na stronie oraz oszczędza czas w porównaniu z testowaniem manualnym.

1.2 Zakres projektu

Projekt obejmował przygotowanie i uruchomienie automatycznych testów interfejsu strony głównej WSiZ oraz wybranych funkcjonalności dostępnych z poziomu użytkownika. Zakres został celowo ograniczony do elementów widocznych i możliwych do przetestowania bez bezpośredniego dostępu do zaplecza serwisu.

W projekcie uwzględniono:

- Weryfikację ładowania strony głównej
Sprawdzenie, czy serwis otwiera się poprawnie i czy kluczowe elementy, takie jak nagłówek i tytuł strony, są dostępne.
- Test działania menu głównego
Automatyczne przejście do sekcji takich jak „Uczelnia”, „Dla studenta”, „Dla kandydata”, „Dla biznesu”, „Nauka i Badania” oraz „Dla otoczenia” wraz z potwierdzeniem, że wyświetlany jest odpowiedni content.
- Test przełącznika języków
Automatyczne przejście ze strony głównej do wersji EN, UA i RU, a następnie powrót do polskiej wersji.
- Test przekierowania po kliknięciu logo
Sprawdzenie, czy kliknięcie w logo uczelni przenosi użytkownika z dowolnej podstrony z powrotem na stronę główną.

- Test działania wyszukiwarki
Wpisanie frazy w pole wyszukiwania oraz potwierdzenie wyświetlenia wyników zgodnych z zapytaniem.
- Test linków górnego paska nawigacyjnego
Sprawdzenie odnośników kierujących do zewnętrznych usług uczelni, takich jak Wirtualna Uczelnia, e-Learning, Poczta, e-Uslugi czy System Wydruków, z uwzględnieniem obsługi otwierania nowych kart.

1.3 Środowisko i narzędzia

Testy zostały wykonane w środowisku opartym na języku Python z wykorzystaniem biblioteki Selenium WebDriver, pozwalającej na automatyzację interakcji z przeglądarką. Projekt został uruchomiony lokalnie, bez wykorzystania zewnętrznych narzędzi CI/CD.

Do realizacji projektu wykorzystano następujące narzędzia i komponenty:

- Język programowania: Python 3.14
Główny język wykorzystywany do pisania skryptów testujących.
- Framework testowy: PyTest
Umożliwił organizację testów w postaci osobnych modułów oraz uruchamianie ich zbiorczo.
- Biblioteki:
 - selenium – obsługa przeglądarki, wyszukiwanie elementów, wykonywanie akcji użytkownika,
 - pytest – framework testowy, uruchamianie i raportowanie testów.
- Automatyzacja przeglądarki: Selenium WebDriver (Chrome)
Przeglądarka była uruchamiana lokalnie, w trybie graficznym, bez trybu headless, aby móc obserwować przebieg testów.
- IDE: JetBrains PyCharm
Środowisko wykorzystywane do pisania kodu, zarządzania strukturą projektu oraz wirtualnym środowiskiem.
- Wirtualne środowisko: venv
Oddzieliło zależności projektu od reszty systemu.
- System operacyjny: Windows 10
Środowisko wykonywania testów.

Dodatkowo konfiguracja testów zakłada:

- uruchamianie każdego testu w nowej instancji przeglądarki (czysty stan sesji),
- ręczne zatwierdzanie baneru dotyczącego cookies przed wykonaniem akcji testowych,
- brak modyfikacji stanu użytkownika.

1.4 Identyfikacja elementów do testowania

W projekcie skupiono się na testowaniu podstawowych funkcji dostępnych z poziomu strony głównej oraz kluczowych elementów interfejsu użytkownika. Wybrane scenariusze testowe obejmują elementy istotne z perspektywy pierwszego kontaktu użytkownika z witryną oraz nawigacji po najważniejszych sekcjach.

Testowane zostały następujące obszary:

a) Menu główne (nawigacja góra)

Każdy główny element menu został przetestowany pod kątem poprawnego przekierowania do odpowiedniej podstrony. Testy obejmowały kliknięcia w:

- Uczelnia
- Nauka i Badania
- Dla Studenta
- Dla Kandydata
- Dla Biznesu
- Dla Otoczenia

Celem było sprawdzenie, czy po kliknięciu link otwiera się poprawna podstrona, a adres URL ulega zmianie.

b) Górnego menu serwisowe (panel narzędziowy)

Przetestowano odnośniki kierujące do zewnętrznych systemów uczelnianych, w tym:

- Wirtualna Uczelnia
- e-Learning
- Poczta
- e-Uslugi
- Biblioteka
- System Wydruków
- e-Praktyki

Część z nich otwiera się w nowej karcie, co zostało uwzględnione w testach.

c) Wyszukiwarka strony

Przetestowano działanie ikony lupy, otwieranie pola wyszukiwania oraz poprawne wyświetlanie wyników po wpisaniu frazy "informatyka".

d) Przełączanie wersji językowych strony

Sprawdzono poprawność przejścia z wersji polskiej do wersji:

- angielskiej (en.uitm.edu.eu),
- ukraińskiej (ua.uitm.edu.eu),
- rosyjskiej (ru.uitm.edu.eu),
a następnie powrót do wersji polskiej poprzez kliknięcie ikony flagi.

e) Logo strony

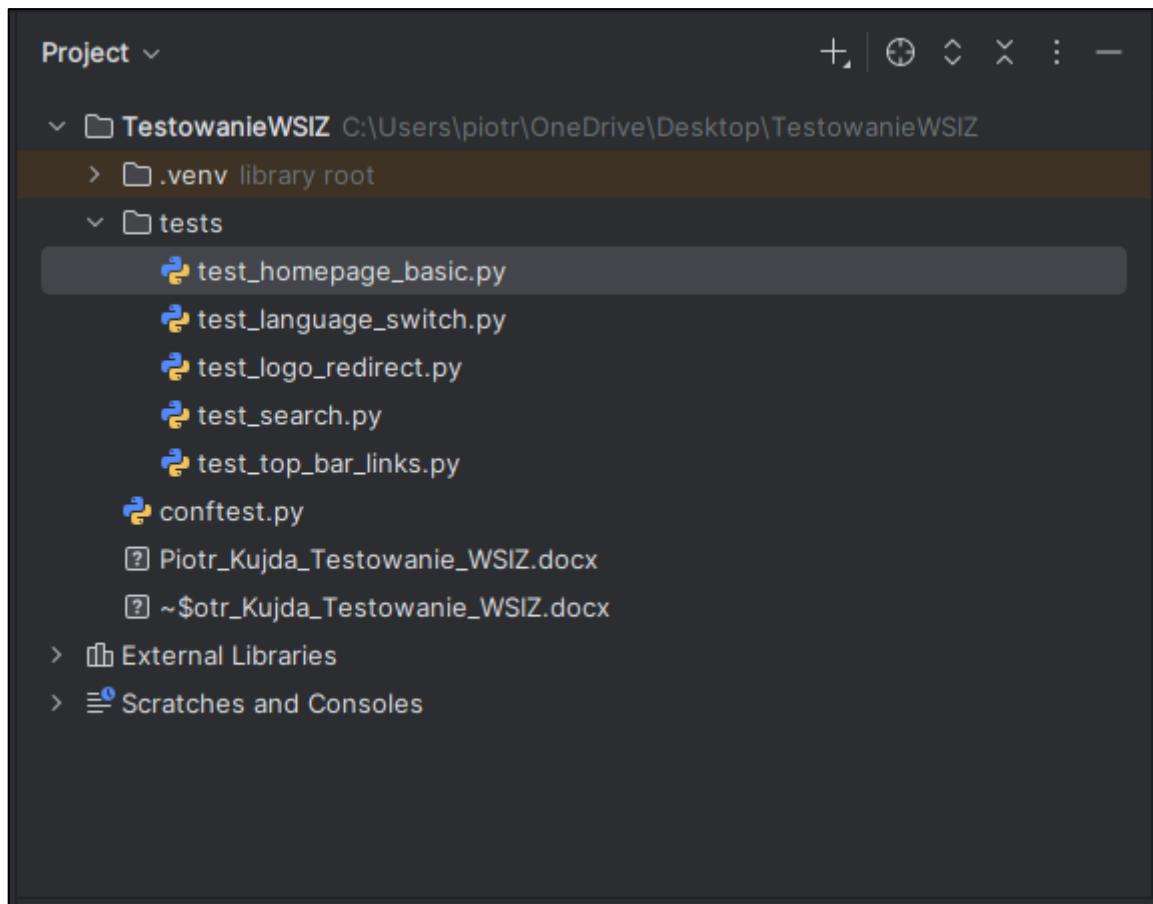
Przetestowano poprawność działania linku powrotnego do strony głównej poprzez kliknięcie w logo uczelni z poziomu podstrony.

2. Struktura projektu

Struktura katalogów (istotne elementy):

- conftest.py – plik z konfiguracją pytest, zawierający wspólny fixture wsiz_home tworzony dla każdego testu:
 - uruchamia nową instancję Chrome,
 - przechodzi na stronę <https://wsiz.edu.pl/>,
 - maksymalizuje okno,
 - po zakończeniu testu zamyka przeglądarkę.
- katalog tests/:
 - test_homepage_basic.py – testy nawigacji w głównym czerwonym menu,
 - test_language_switch.py – test przełączania wersji językowej PL ↔ EN/UA/RU,
 - test_logo_redirect.py – test przekierowania z podstrony na stronę główną po kliknięciu logo,
 - test_search.py – test pola wyszukiwania (ikona lupy),
 - test_top_bar_links.py – testy linków w górnym (czarnym) pasku: Wirtualna Uczelnia, e-Learning, Poczta, e-Uslugi, Biblioteka, System wydruków, e-Praktyki.

- Każdy plik testowy korzysta z tego samego fixture'a wsiz_home, dzięki czemu każdy test działa w osobnej, czystej sesji przeglądarki.



Rysunek 1. Schemat struktury projektu.

3. Opis przypadków testowych

3.1 Tabela przypadków testowych

| ID | Nazwa testu | Kroki | Oczekiwany rezultat |
|-------|-----------------------------|---------------------------|--|
| TC-01 | Wyświetlanie strony głównej | Wejść na wsiz.edu.pl | Strona ładuje się poprawnie, nagłówek widoczny |
| TC-02 | Nawigacja → Uczelnia | Kliknąć „Uczelnia” w menu | Ładuje się strona z treścią o uczelni |
| TC-03 | Nawigacja → Nauka i Badania | Kliknąć „Nauka i Badania” | Strona z treścią o nauce/badaniach |
| TC-04 | Nawigacja → Dla studenta | Kliknąć „Dla studenta” | Strona dla studentów |
| TC-05 | Nawigacja → Dla kandydata | Kliknąć „Dla kandydata” | Strona dla kandydatów |

| | | | |
|-------|-----------------------------|--------------------------------------|---|
| TC-06 | Nawigacja → Dla biznesu | Kliknąć „Dla biznesu” | Strona dla biznesu |
| TC-07 | Nawigacja → Dla otoczenia | Kliknąć „Dla otoczenia” | Strona dla otoczenia |
| TC-08 | Link → Wirtualna Uczelnia | Kliknąć link w górnym pasku | Przekierowanie do <i>my.wsiz.edu.pl</i> |
| TC-09 | Link → e-Learning | Kliknąć link | Przekierowanie do <i>moodle.wsiz.edu.pl</i> |
| TC-10 | Link → Poczta | Kliknąć link | Przekierowanie do portalu Microsoft (Outlook) |
| TC-11 | Link → e-Uslugi | Kliknąć link | Otwiera <i>euslugi.wsiz.pl</i> w nowej karcie |
| TC-12 | Link → Biblioteka | Kliknąć link | Otwiera stronę biblioteki |
| TC-13 | Link → System Wydruków | Kliknąć link | Otwiera <i>print.wsiz.rzeszow.pl</i> w nowej karcie |
| TC-14 | Link → e-Praktyki | Kliknąć link | Przekierowanie do <i>portal.wsiz.edu.pl</i> |
| TC-15 | Wyszukiwarka — wyszukiwanie | Kliknąć lupę → wpisać frazę → ENTER | Wyświetlone wyniki z frazą „informatyka” |
| TC-16 | Zmiana języka PL → EN → PL | Wybrać EN, potem PL | Przejście do wersji EN, potem powrót do wsiz.edu.pl |
| TC-17 | Zmiana języka PL → UA → PL | Wybrać UA, potem PL | Przejście do UA i powrót do PL |
| TC-18 | Zmiana języka PL → RU → PL | Wybrać RU, potem PL | Przejście do RU i powrót do PL |
| TC-19 | Kliknięcie logo uczelni | Wejść w „Dla studenta”, kliknąć logo | Powrót na stronę główną |

3.2. test_homepage_basic.py

Cel: Sprawdzenie, czy strona główna działa oraz czy linki w głównym menu przenoszą użytkownika do odpowiednich sekcji serwisu.

1. test_homepage_loads

- Wejście na stronę główną.

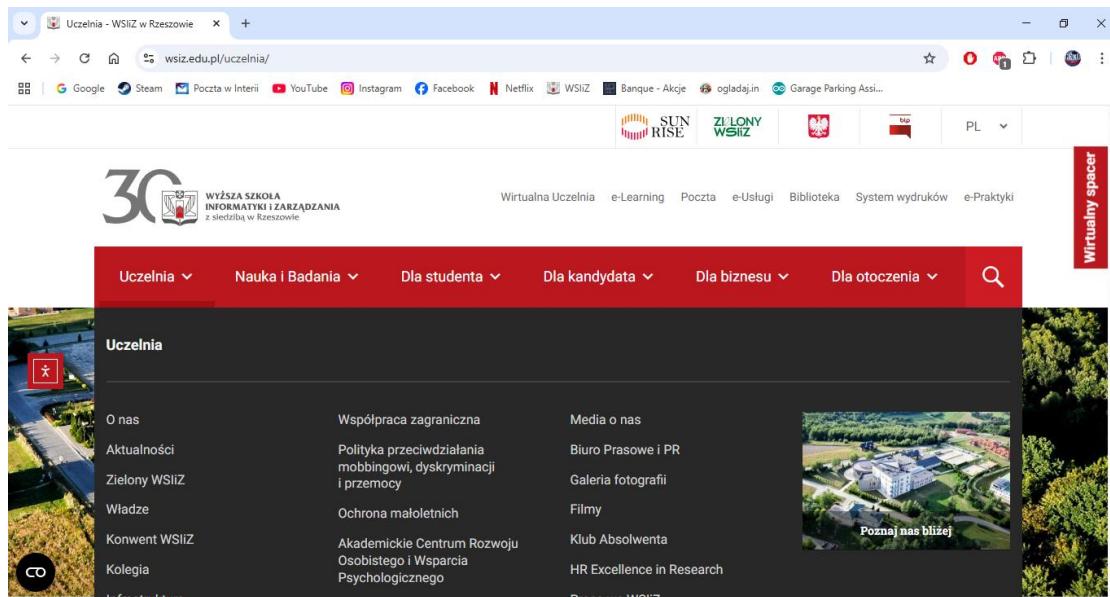


Rysunek 2. Strona główna Uczelni.

- Oczekiwanie na pojawienie się elementu header.
- Sprawdzenie, czy tytuł strony zawiera nazwę uczelni („WSIiZ” lub „Wyższa Szkoła Informatyki...”).

2. test_navigacja_uczelnia

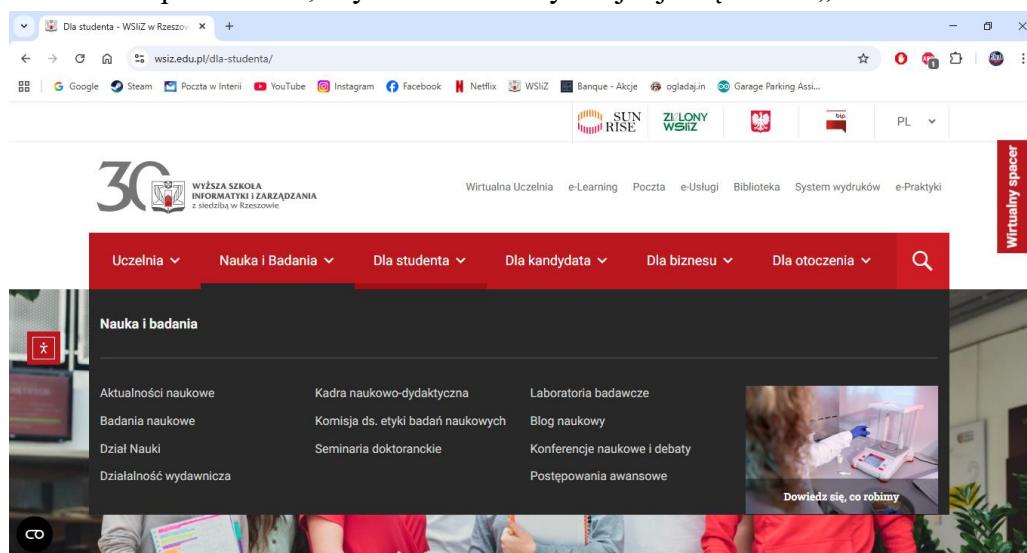
- Kliknięcie pozycji menu „Uczelnia” (po XPATH z mappingu MENU_XPATHS).
- Oczekiwanie na zmianę adresu URL.
- Sprawdzenie, czy w treści strony znajduje się słowo „uczelnia”.



Rysunek 3. Strona główna z zakładką “Uczelnia”.

3. test_navigacja_nauka_i_badania

- Kliknięcie „Nauka i Badania”.
- Oczekiwanie na zmianę URL.
- Sprawdzenie, czy w treści strony znajduje się słowo „nauka”.



Rysunek 4. Strona główna z zakładką ”Nauka I Badania”.

4. test_nawigacja_dla_studenta

- Kliknięcie „Dla studenta”.
- Sprawdzenie, czy w treści pojawia się słowo „student”.

The screenshot shows the official website of the University of Computer Science and Management in Rzeszów (WSiZ). The top navigation bar includes links for SUNRISE, ZIELONY WSiZ, the Polish flag, and language selection (PL). Below the header, there's a main menu with categories like Uczelnia, Nauka i Badania, and a prominent red 'Dla studenta' dropdown. A sub-menu for 'Dla studenta' is open, showing options such as 'Aktualności studenckie', 'Centrum Dydaktyczne Języków Obcych', 'Wykłady dla chętnych', 'Vademecum', 'Osiągnięcia Studentów i Absolwentów', 'Erasmus+', 'Samorząd Studencki WSiZ', and 'Zajęcia prowadzone z wykorzystaniem metod i technik kształcenia na odległość'. The background features a photograph of students holding books and a red hoodie with '90' on it.

Rysunek 5. Strona główna z zakładką "Dla Studenta".

5. test_nawigacja_dla_kandydata

- Kliknięcie „Dla kandydata”.
- Sprawdzenie, czy w treści znajduje się słowo „kandydat”.

This screenshot shows the same WSiZ website as above, but with the 'Dla kandydata' tab highlighted in the main navigation menu. The sub-menu for 'Dla kandydata' is expanded, listing items like 'Oferta edukacyjna', 'Studia I stopnia i jednolite magisterskie', 'Studia II stopnia', 'Studia anglojęzyczne', 'Studia podyplomowe', 'Seminaria doktoranckie', 'Zasady rekrutacji', 'System Elektronicznej Rekrutacji', 'Sprawdź wyniki rekrutacji', 'Rekrutacja krok po kroku', 'Czesne', 'Stypendia - studuj za darmo', 'Akademik', 'Informacje praktyczne', 'Informacje dla kandydata', 'Masz powody, żeby wybrać WSiZ!', 'Studia w Rzeszowie', 'Certyfikaty międzynarodowe', 'Blog Rekrutacyjny', and 'Projekt Matura'. A small image of three students sitting together is visible on the right side of the page.

Rysunek 6. Strona główna z zakładką "Dla kandydata".

6. test_nawigacja_dla_biznesu

- Kliknięcie „Dla biznesu”.
- Sprawdzenie, czy w treści występuje słowo „biznes”.

The screenshot shows a browser window with the URL wsiz.edu.pl/dla-biznesu/. The page header includes the WSiZ logo, navigation links like 'Uczelnia', 'Nauka i Badania', 'Dla studenta', 'Dla kandydata', 'Dla biznesu' (which is highlighted in red), 'Dla otoczenia', and a search bar. A sidebar on the right is titled 'Wirtualny spacer'. The main content area features a heading 'Dla biznesu' with a small icon. Below it are three boxes containing text and logos: 'Instytut Badań i Analiz Finansowych', 'Kampus WSiZ w Kielnarowej', and 'Enterprise Europe Network'. Another box for 'Centrum Studiów Podyplomowych' and 'Centrum Egzaminacyjne Dimaq' is partially visible. At the bottom, there's a photo of a man and some descriptive text about business needs and partnerships.

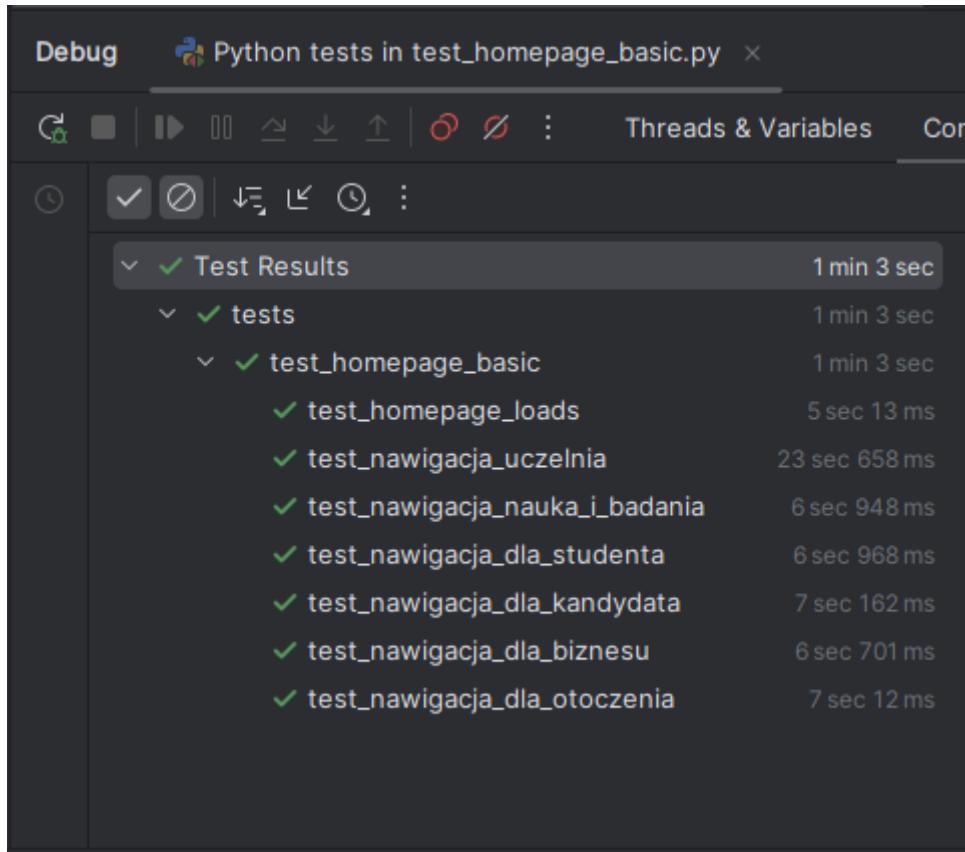
Rysunek 7. Strona główna z zakładką "Dla biznesu".

7. test_nawigacja_dla_otoczenia

- Kliknięcie „Dla otoczenia”.
- Sprawdzenie, czy w treści występuje słowo „otoczenia”.

The screenshot shows a browser window with the URL wsiz.edu.pl/dla-otoczenia/. The page structure is identical to Figure 7, with the 'Dla otoczenia' link highlighted in red in the top navigation bar. The main content area features a heading 'Dla otoczenia' with a small icon. Below it are several boxes listing various programs and initiatives: 'Publiczne Akademickie Liceum Ogólnokształcące z oddziałami dwujęzycznymi', 'Akademia 50+', 'Adaptacja kulturowa', 'Wielkie Pytania w Nauce', 'W labiryntach świata', 'Przybij piątkę nauce', 'Centrum Kultury i Języka Polskiego', 'Centrum Egzaminacyjne CAMBRIDGE English Qualifications', 'EUROPE DIRECT', 'Tour de WSiZ', 'Rzeszowska Akademia Inspiracji', 'Świetlica „Przyjdź i odpocznij”', 'Konkurs - Młodzi dla zdrowia', 'Konkurs - Młodzi dla społeczeństwa', 'Konkurs - Młodzi dla planety', and a note about the university not being a traditional educational institution. There's also a photo of a man at the bottom.

Rysunek 8. Strona główna z zakładką "Dla otoczenia".



Rysunek 9. Końcowe wyniki przeprowadzonych testów.

3.3. test_language_switch.py

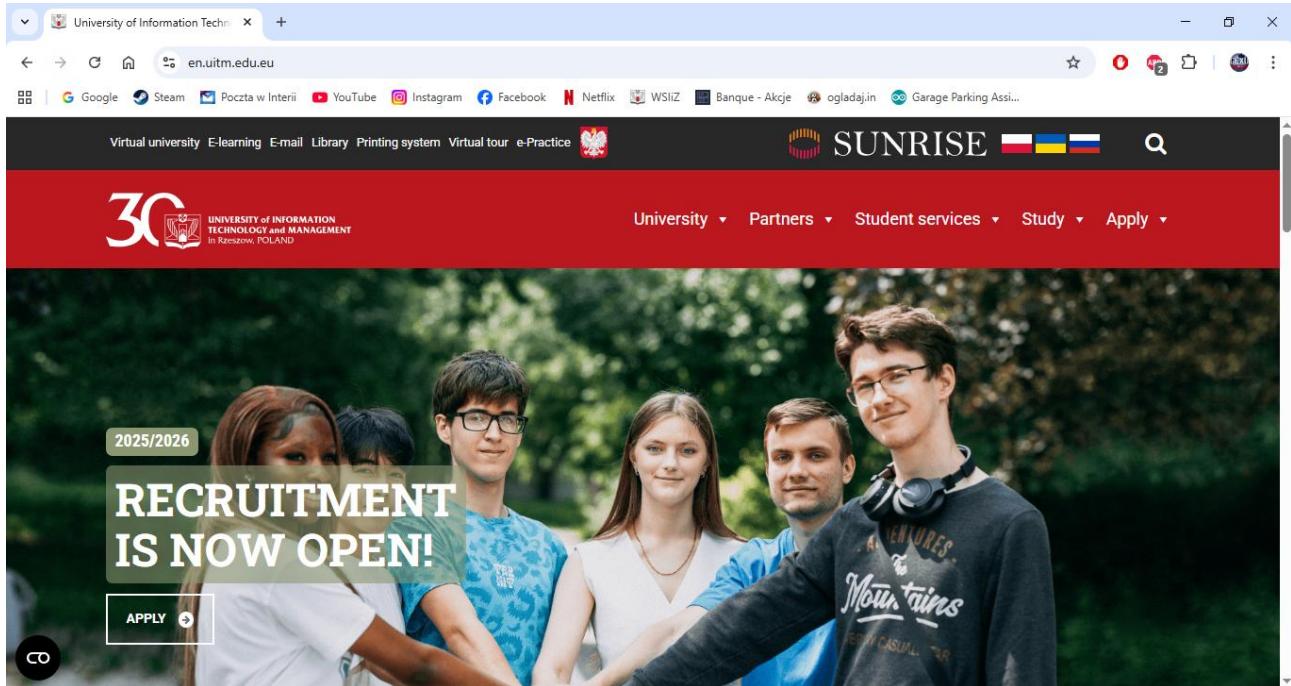
Cel: Weryfikacja działania przełącznika wersji językowych PL ↔ EN/UA/RU.

Dla każdego z trzech testów:

1. Początkowo użytkownik jest na stronie wsiz.edu.pl (wersja PL).
2. Skrypt znajduje element <select id="dynamic_select">.
3. Po atrybucie value wybierana jest odpowiednia wersja językowa:
 - en.uitm.edu.eu,
 - ua.uitm.edu.eu,
 - ru.uitm.edu.eu.
4. Oczekuje się przełączenia domeny (np. adres zawiera en.uitm.edu.eu).
5. Następnie znajdowana jest ikona polskiej flagi (po fragmencie ścieżki grafiki flaga_polska) i klikana, co powinno przywrócić adres w domenie wsiz.edu.pl.

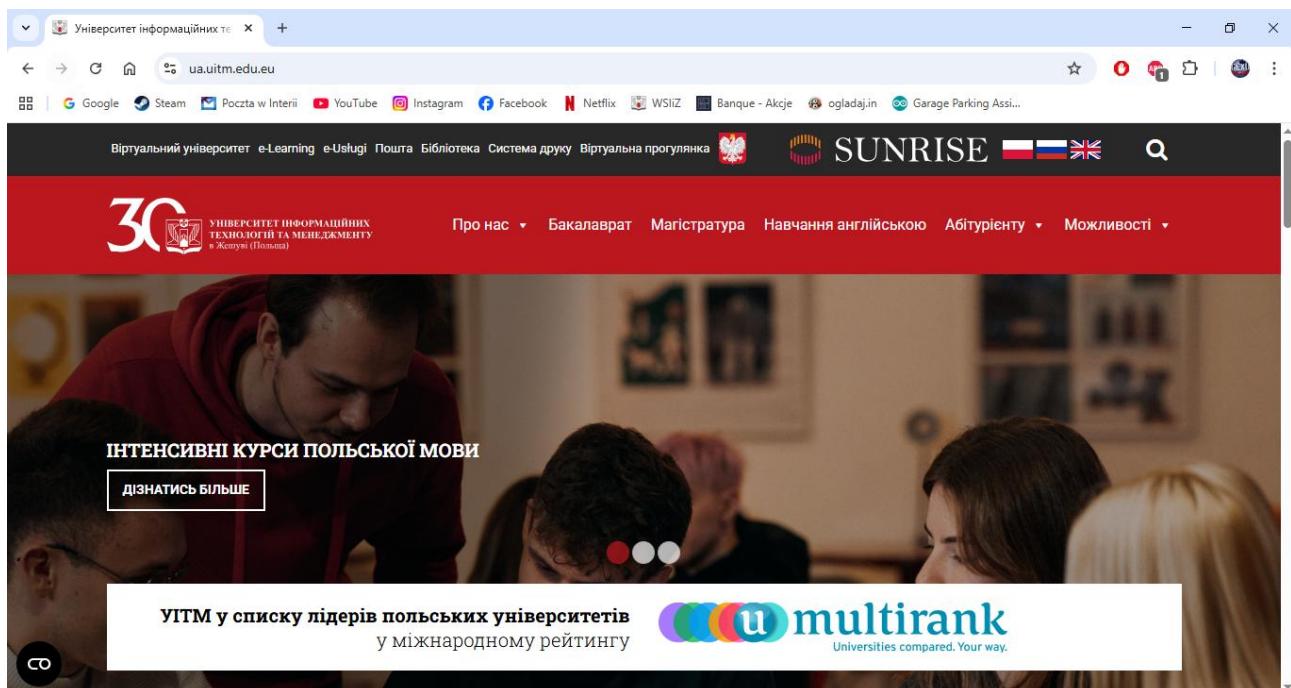
Testy:

- test_language_switch_pl_en_pl



Rysunek 10. Główna strona Uczelni w języku angielskim.

- test_language_switch_pl_ua_pl



Rysunek 11. Główna strona Uczelni w języku ukraińskim.

- test_language_switch_pl_ru_pl



Rysunek 12. Główna strona Uczelni w języku rosyjskim.

Każdy test sprawdza:

- czy po wyborze języka adres zawiera odpowiednią subdomenę,
- czy po kliknięciu polskiej flagi następuje powrót do wsiz.edu.pl.

| Test Case | Time |
|-------------------------------|---------------|
| test_language_switch_pl_en_pl | 17 sec 744 ms |
| test_language_switch_pl_ua_pl | 23 sec 733 ms |
| test_language_switch_pl_ru_pl | 18 sec 584 ms |

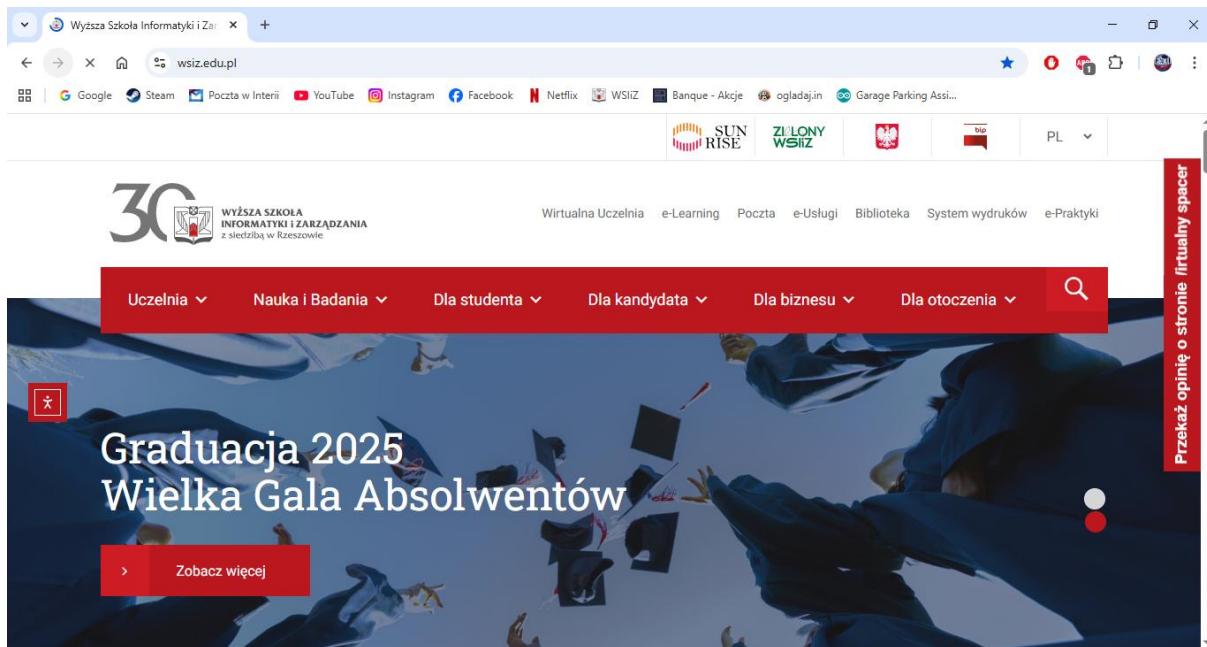
Rysunek 13. Końcowe wyniki przeprowadzonych testów.

3.4. test_logo_redirect.py

Cel: Sprawdzenie, czy kliknięcie logo uczelni przenosi z podstrony z powrotem na stronę główną.

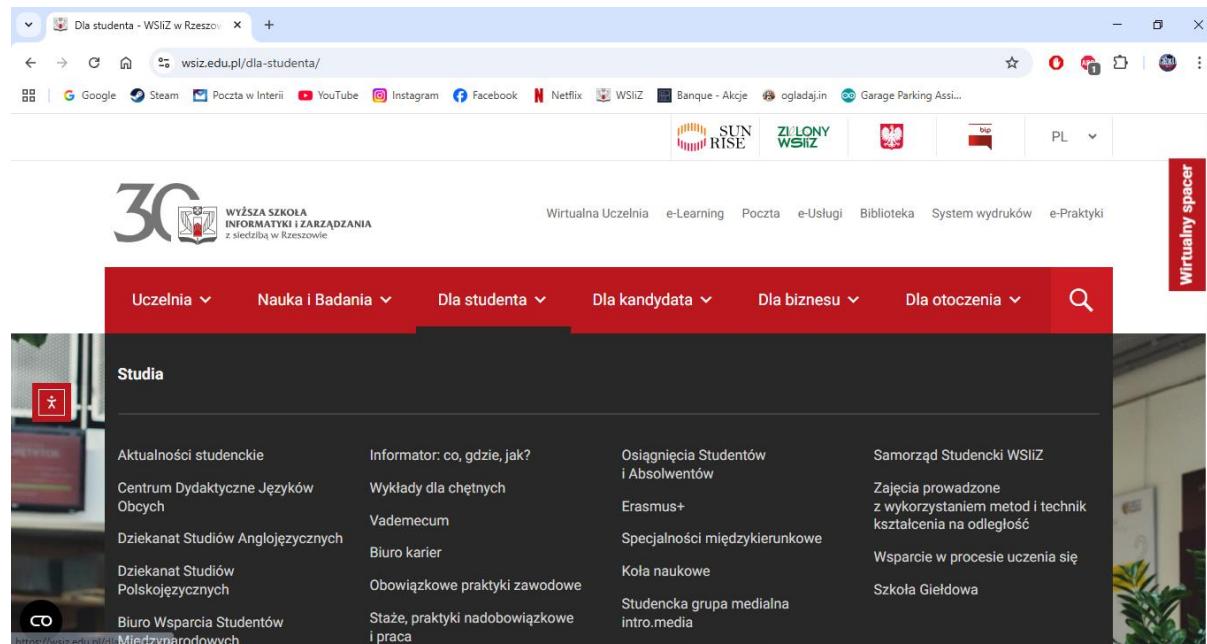
Scenariusz:

1. Wejście na stronę główną.



Rysunek 14. Główna strona Uczelni.

2. Kliknięcie pozycji menu „Dla studenta” (po linku tekstowym).



Rysunek 15. Główna strona z zakładką “Dla studenta”.

3. Czekanie aż adres URL będzie zawierał wsiz.edu.pl/dla-studenta.
4. Krótkie oczekanie (5 sekund) na podstronie.

5. Wyszukanie elementu logo – link o XPATH `//*[@id='mega-menu-item-52539']/a`.
6. Przewinięcie do logo i kliknięcie go przy pomocy JavaScript
(`driver.execute_script("arguments[0].click();", logo_link)`).



Rysunek 16. Logo Uczelni.

7. Oczekiwanie, aż adres będzie zaczynał się od `https://wsiz.edu.pl`.
8. Asercja, że rzeczywiście nastąpił powrót na stronę główną.

A screenshot of a web browser showing the homepage of the University of Computer Science and Management (WSIZ). The address bar shows 'wsiz.edu.pl'. The page features the university's logo and name at the top left. A navigation bar with links like 'Uczelnia', 'Nauka i Badania', 'Dla studenta', 'Dla kandydata', 'Dla biznesu', 'Dla otoczenia', and a search icon. The main content area has a dark blue background with a photo of graduates throwing caps in the air. A red button at the bottom left says 'Zobacz więcej'. On the right side, there is a vertical red sidebar with the text 'Przekaż opinię o stronie / virtualny spacer'.

Rysunek 17. Główna strona Uczelni.

A screenshot of the PyCharm IDE showing the results of a Python test run. The title bar says 'Run Python tests in test_logo_redirect.py'. The results pane shows a tree view of tests: 'Test Results' > 'tests' > 'test_logo_redirect' > 'test_logo_redirects_to_home'. All tests are marked with a green checkmark and have a duration of '11 sec 811 ms'.

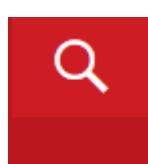
Rysunek 18. Końcowe wyniki przeprowadzonych testów.
16

3.5. test_search.py

Cel: Sprawdzenie działania wyszukiwarki (ikona lupy w górnym menu).

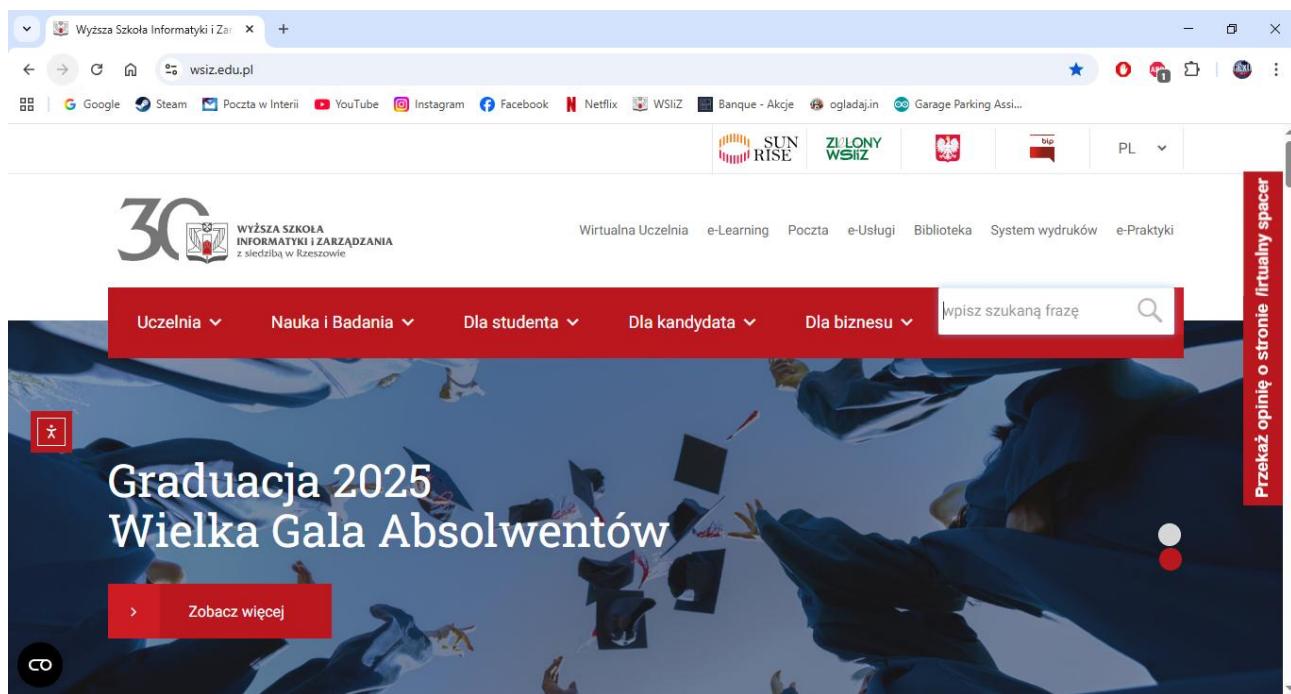
Scenariusz:

1. Po załadowaniu strony krótkie oczekiwanie (`time.sleep(3)`).
2. Kliknięcie ikony lupy po XPATH:
 - o `SEARCH_TOGGLE_XPATH = "//*[@@id='mega-menu-item-52922']/div/form/span".`



Rysunek 19. Ikona lupy.

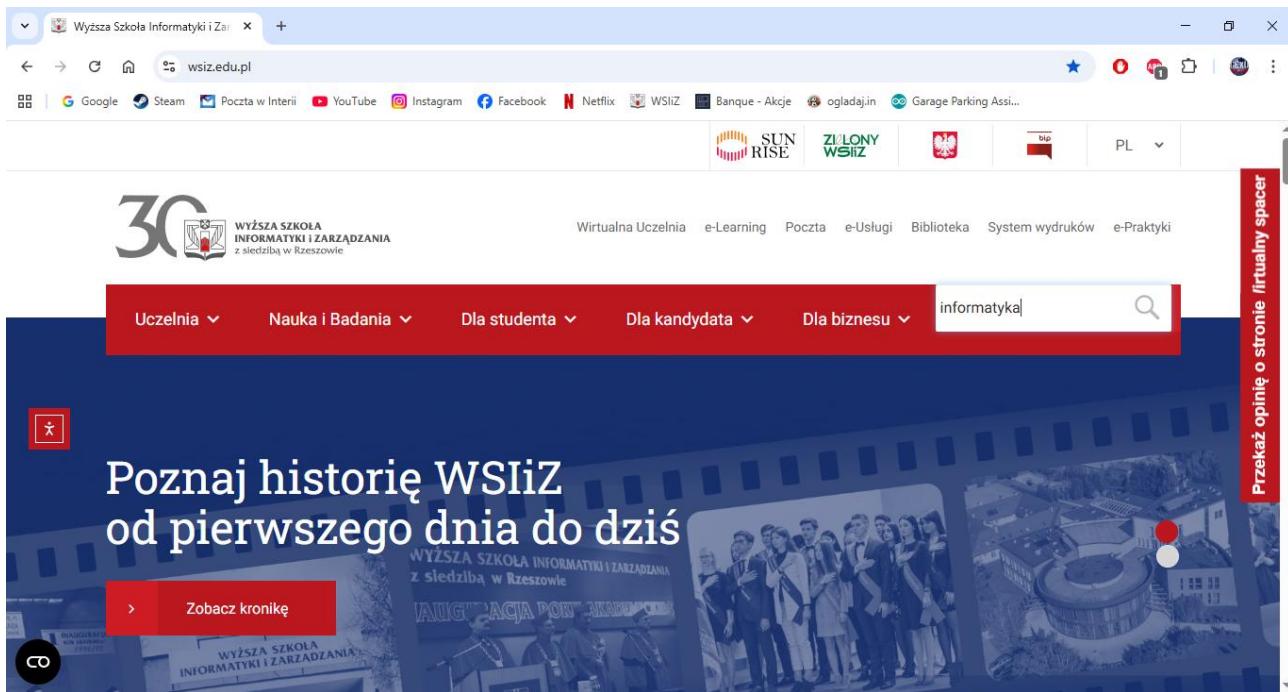
3. Oczekивание (w miarę możliwości) na ustawienie atrybutu aria-expanded="true" – informacja, że pole wyszukiwania zostało rozwinięte.
4. Pobranie pola tekstowego po identyfikatorze:
 - o `SEARCH_INPUT_ID = "mega-search-52922".`



Rysunek 20. Strona główna Uczelni z możliwością wpisania frazy.

5. Ustawienie focusa na polu, wyczyszczenie zawartości.

6. Wpisanie frazy „informatyka” oraz naciśnięcie ENTER.

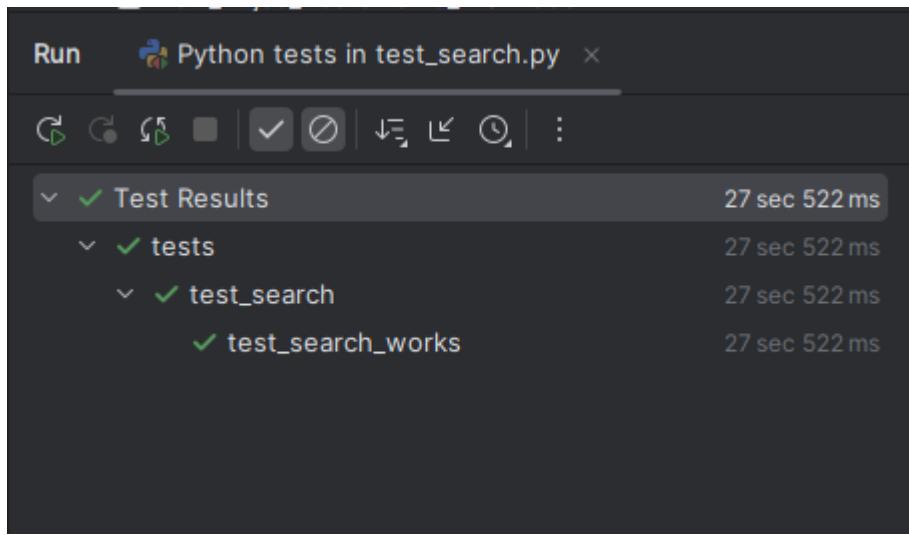


Rysunek 21. Główna strona Uczelni z wpisaną frazą “informatyka”.

- 7.** Oczekiwanie, aż załadują się wyniki (sprawdzany fragment ?s= w URL lub obecność frazy w źródle strony).
- 8.** Asercja, że na stronie wyników występuje tekst „informatyka”.



Rysunek 22. Wyniki wyszukiwania dla wpisanej frazy.



Rysunek 23. Końcowe wyniki przeprowadzonego testu

3.6. test_top_bar_links.py

Cel: Weryfikacja poprawności odnośników w górnym (czarnym) pasku narzędziowym.

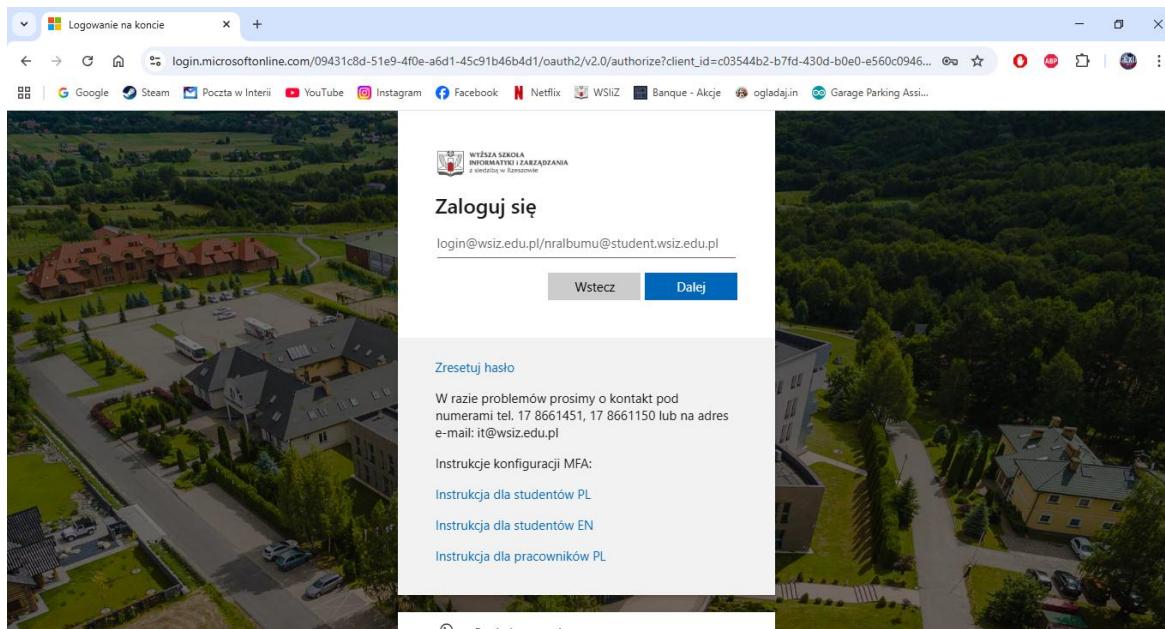
Dla wszystkich testów używane są osobne funkcje pomocnicze:

- `_click_link_same_tab_xpath(...)` – kliknięcie linku, który otwiera się w tej samej karcie (sprawdzana zmiana URL i obecność oczekiwanej części URL),
 - `_click_link_new_tab_xpath(...)` – kliknięcie linku otwieranego w nowej karcie (sprawdzenie pojawienia się nowej karty, przełączenie na nią, weryfikacja URL).

Testowane linki:

1. test_top_link_wirtualna_uczelnia

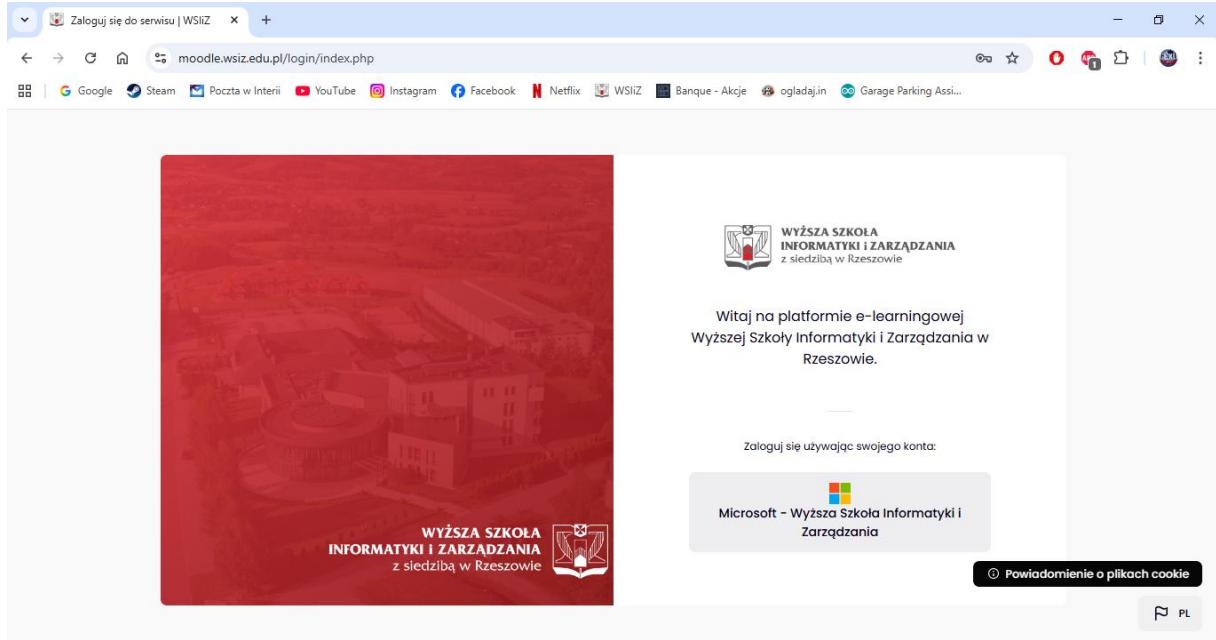
- XPATH: //*[@id='mega-menu-item-52540']/a
 - Oczekiwane fragmenty URL: my.wsiz.edu.pl.



Rysunek 24. Główna strona "Wirtualna Uczelnia".

2. test_top_link_e_learning

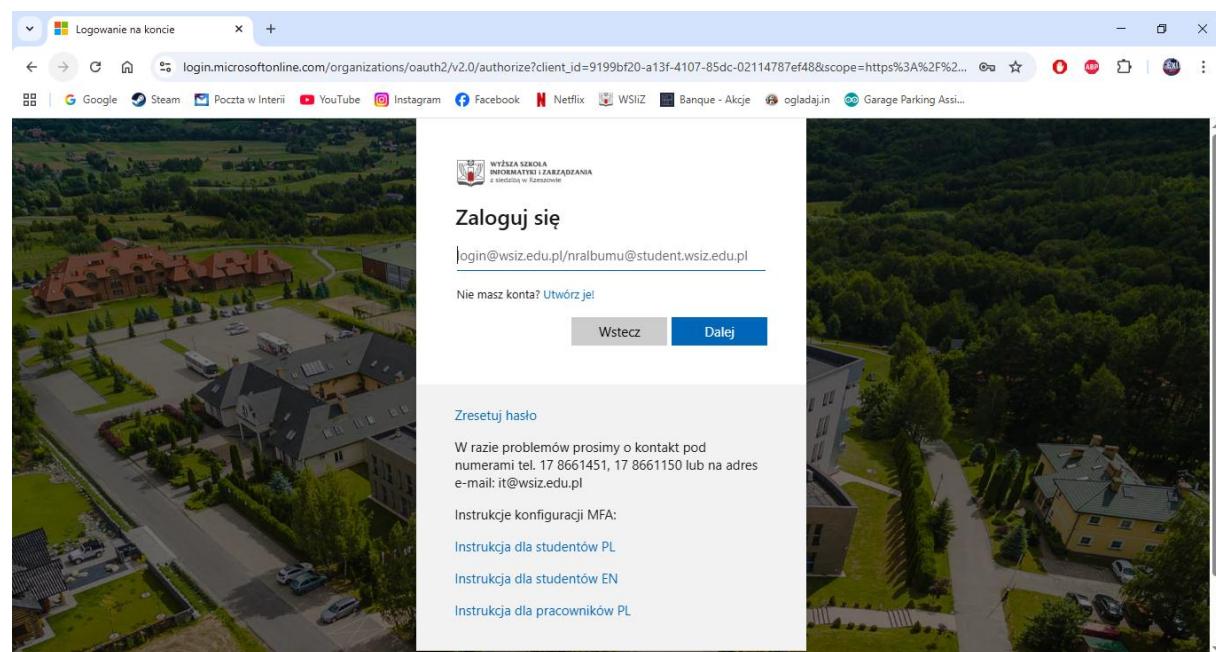
- XPATH: `//*[@@id='mega-menu-item-52541']/a`
- Oczekiwane: `moodle.wsiz.edu.pl`.



Rysunek 25. Główna strona “e-Learning”.

3. test_top_link_poczta

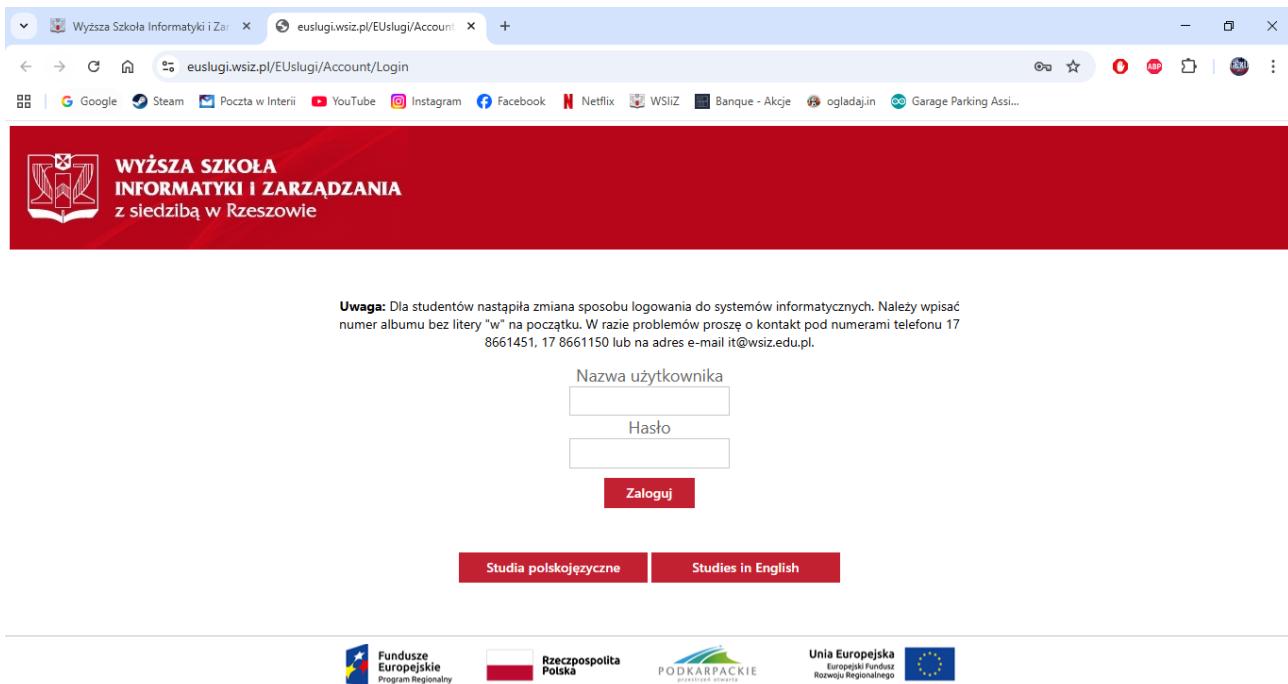
- XPATH: `//*[@@id='mega-menu-item-52542']/a`
- Oczekiwane: m.in. `poczta.wsiz.edu.pl`, `outlook.office365.com`, `office.com`, `microsoftonline.com`.



Rysunek 26. Główna strona “Poczta”.

4. test_top_link_e_uslugi (*nowa karta*)

- XPATH: //*[@id='mega-menu-item-52543']/a
- Oczekiwane: euslugi.wsiz.pl.



Rysunek 27. Główna strona “e-Uslugi”.

5. test_top_link_biblioteka

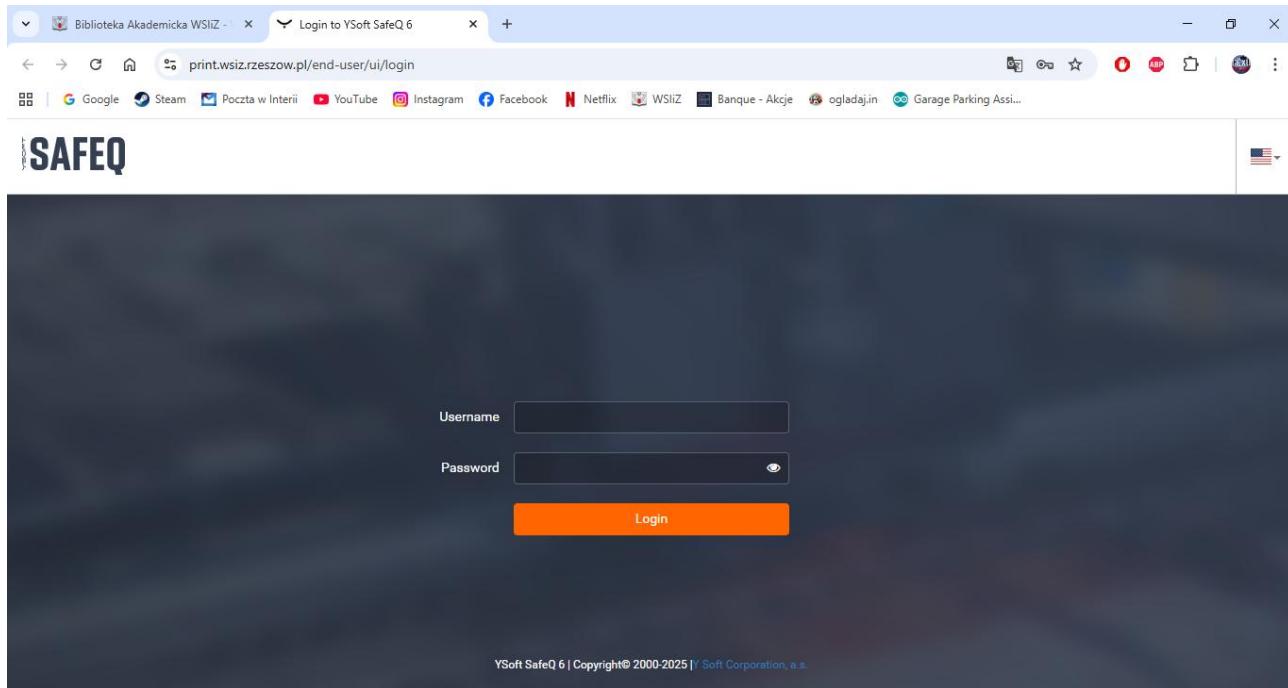
- XPATH: //*[@id='mega-menu-item-52544']/a
- Oczekiwane: wsiz.edu.pl/uczelnia/jednostki-wsiiz/biblioteka-wsiiz lub skrócona forma biblioteka-wsiiz.



Rysunek 28. Główna strona “Biblioteka”.

6. test_top_link_system_wydrukow (nowa karta)

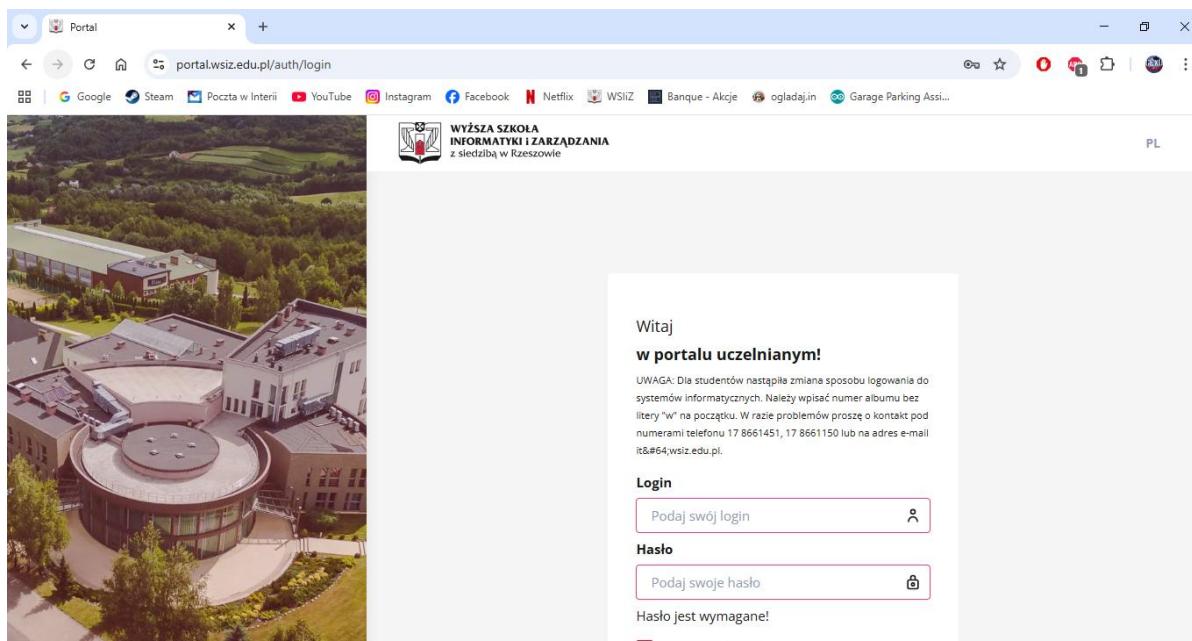
- XPATH: //*[@id='mega-menu-item-52545']/a
- Oczekiwane: print.wsiz.rzeszow.pl.



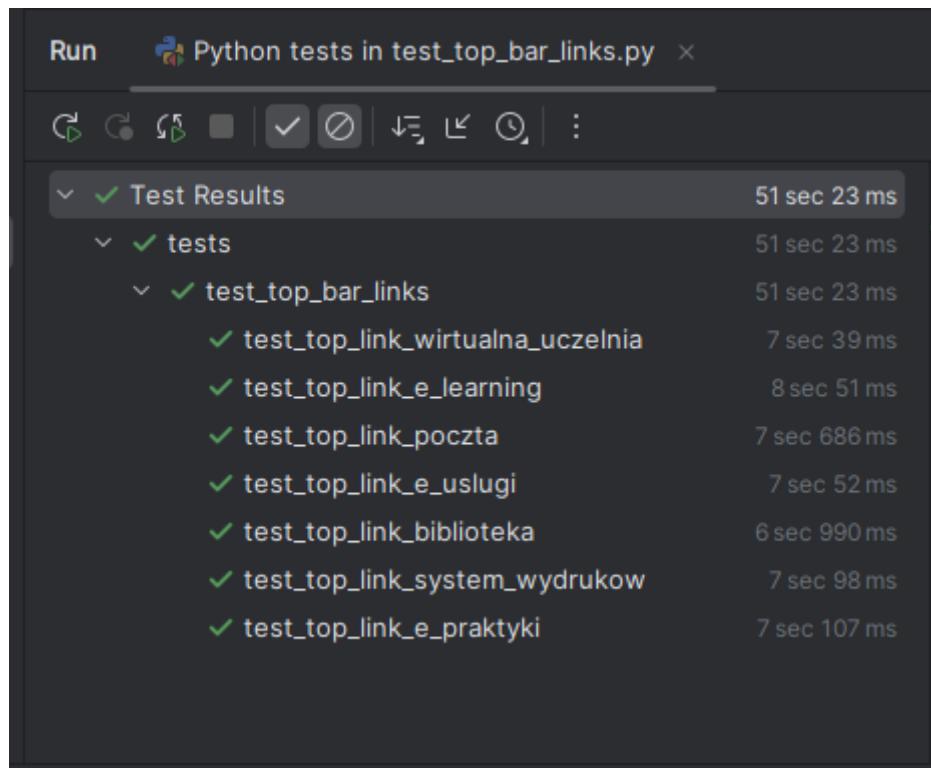
Rysunek 29. Główna strona “System wydruków”.

7. test_top_link_e_praktyki

- XPATH: //*[@id='mega-menu-item-99388']/a
- Oczekiwane: portal.wsiz.edu.pl.



Rysunek 30. Główna strona “e-Praktyki”.



Rysunek 31. Końcowe wyniki przeprowadzonych testów.

3.7. Uruchamianie testów

Wszystkie testy uruchamiane są komendą:

```
pytest -v
```

Wynik końcowy z konsoli: 19 passed in 341.27s (0:05:41).

```
=====
(.venv) PS C:\Users\piotr\OneDrive\Desktop\TestowanieWSIZ> pytest -v
===== test session starts =====
platform win32 -- Python 3.14.0, pytest-9.0.1, pluggy-1.6.0 -- C:\Users\piotr\OneDrive\Desktop\TestowanieWSIZ\.venv\Scripts\python.exe
cachedir: .pytest_cache
rootdir: C:\Users\piotr\OneDrive\Desktop\TestowanieWSIZ
collected 19 items

tests/test_homepage_basic.py::test_homepage_loads PASSED [  5%]
tests/test_homepage_basic.py::test_nawigacja_uczelnia PASSED [ 10%]
tests/test_homepage_basic.py::test_nawigacja_nauka_i_badania PASSED [ 15%]
tests/test_homepage_basic.py::test_nawigacja_dla_studenta PASSED [ 21%]
tests/test_homepage_basic.py::test_nawigacja_dla_kandydata PASSED [ 26%]
	tests/test_homepage_basic.py::test_nawigacja_dla_biznesu PASSED [ 31%]
tests/test_homepage_basic.py::test_nawigacja_dla_otoczenia PASSED [ 36%]
D tests/test_language_switch.py::test_language_switch_pl_en_pl PASSED [ 42%]
tests/test_language_switch.py::test_language_switch_pl_va_pl PASSED [ 47%]
P tests/test_language_switch.py::test_language_switch_pl_ru_pl PASSED [ 52%]
tests/test_logo_redirect.py::test_logo_redirects_to_home PASSED [ 57%]
tests/test_search.py::test_search_works PASSED [ 63%]
H tests/test_top_bar_links.py::test_top_link_wirtualna_uczelnia PASSED [ 68%]
tests/test_top_bar_links.py::test_top_link_e_learning PASSED [ 73%]
D tests/test_top_bar_links.py::test_top_link_poczta PASSED [ 78%]
tests/test_top_bar_links.py::test_top_link_e_uslugi PASSED [ 84%]
B tests/test_top_bar_links.py::test_top_link_biblioteka PASSED [ 89%]
tests/test_top_bar_links.py::test_top_link_system_wydrukow PASSED [ 94%]
I tests/test_top_bar_links.py::test_top_link_e_praktyki PASSED [100%]

===== 19 passed in 341.27s (0:05:41) =====
(.venv) PS C:\Users\piotr\OneDrive\Desktop\TestowanieWSIZ>

```

Rysunek 32. Końcowe wyniki wszystkich przeprowadzonych testów.

Oznacza, że wszystkie przygotowane przypadki testowe zakończyły się powodzeniem.

4. Propozycja usprawnień i naprawy napotkanych błędów

Podeczas realizacji testów pojawiło się kilka problemów związanych głównie z dynamicznym zachowaniem strony oraz sposobem działania elementów JavaScript. Poniżej przedstawiono obserwacje oraz możliwe usprawnienia.

- Dynamiczne przeładowania strony po jej otwarciu

Strona główna czasami automatycznie przewija się i przeładowuje tuż po wejściu. Powodowało to losowe błędy, szczególnie w testach wyszukiwarki i nawigacji. Ostatecznie rozwiązano to poprzez dodanie krótkich opóźnień oraz przewijania do góry strony.

Możliwe usprawnienie:

- wdrożenie stabilnego oczekiwania opartego na `document.readyState == 'complete'`
 - opóźnione inicjalizowanie elementów dynamicznych po stronie front-endu
 - uproszczenie animacji startowych
- Elementy zasłaniane przez inne komponenty (baner cookies, sticky menu, animacje)

Kliknięcia czasem nie działały, ponieważ element był widoczny, ale przykryty niewidoczną warstwą (np. animacja menu lub niewidzialna nakładka). Rozwiązano to użyciem `execute_script("arguments[0].click()", element)` oraz przewijaniem widoku.

Możliwe usprawnienie:

- usunięcie nakładek pełnoekranowych albo ich opóźnienie
 - zmiana `z-index` elementów, które nie powinny blokować UI
 - zapewnienie możliwości kliknięcia przez standardową metodę Selenium
- Niestabilne zachowanie wyszukiwarki
- Pole wyszukiwania sporadycznie nie przyjmowało wpisanego tekstu, szczególnie tuż po kliknięciu w ikonę lupy. Powodem była asynchroniczna inicjalizacja elementu oraz animacja otwierania.
- Możliwe usprawnienie:
- wymuszenie focusu na elemencie po stronie frontu (`.focus()`)
 - opóźnione wyświetlanie inputu dopiero gdy JS zakończy inicjalizację
 - dodanie stałego identyfikatora lub flagi gotowości
- Testy oparte na lokalizatorach XPATH

Część testów oparta była o XPATH-y powiązane z pełną strukturą HTML. Działało to poprawnie, ale było bardziej podatne na zmiany layoutu niż lokalizowanie po tekście linku lub atrybutach.

Mögliche Optimierung:

- Anwendung von By.LINK_TEXT oder By.CSS_SELECTOR dort, wo es möglich ist
- Hinzufügen stabiler Attributtypen für Testzwecke wie data-testid
- Probleme mit Sprachtests während der Aktualisierung von Domänen

Bei Änderung der Sprachversion traten Fehler auf, die durch das Aufladen und Umleiten zwischen den Domänen wsiz.edu.pl → *.uitm.edu.eu entstanden.

Mögliche Optimierung:

- Hinzufügen eines globalen wait_for_url_change() statt harter Zeitverzögerungen
- Verwenden von WebDriverWait mit einer Bedingung auf die Domäne anstelle einer String-Match

5. Zusammenfassung und Fazit

Projekt hatte das Ziel, die Entwicklung, Implementierung und Analyse automatisierter funktionaler Tests für die Internetseite *wsiz.edu.pl* mithilfe des Python-Selenium-Bibliotheksbasis und des Testframeworks Pytest zu realisieren. In der Arbeit wurde ein vollständiger Testset erstellt, der die Wiederholungsfähigkeit von Szenarien des Benutzers ausnutzt – von der grundlegenden Ladezeit der Seite bis zur Navigation im Menü, Interaktion mit Suchmaschinen, dem Dienstleistungsangebot und inneren Diensten, sowie dem Wechsel zwischen Sprachversionen und dem Zurückkehren zur Hauptseite auf Logoebene. Ganzheitlich wurden 19 Tests erstellt, die alle mit „PASSED“ abgeschlossen wurden.

Die Realisierung des Projekts ermöglichte die Analyse des Verhaltens der Seite unter verschiedenen Automatisierungsszenarien. Die Webseite nutzt ein ausgereiftes Menüsystem, dynamische Ladevorgänge von Elementen und verschachtelte Schichten (z.B. Banner, Cookies, Animationen, Übergänge über Ketten), was zu erheblichen Schwierigkeiten in der Stabilität des Klicks auf bestimmte Elemente führte. Ein Problem stellten zufällige Neuladungen der Seite direkt nach dem Eintreten dar, ebenso wie Situationen, in denen Elemente im DOM sichtbar waren, obwohl sie nicht physisch „klickbar“ waren, aufgrund von überlappenden Schichten. Um die Stabilität zu gewährleisten, wurden folgende Methoden eingesetzt:

- Klicken über JavaScript anstelle der standardären Methode .click(),
- Durchscrollen des Bildschirms bis zu den klickbaren Elementen,
- Erwartung der tatsächlichen Änderung des URLs statt nur der Sichtbarkeit des Elements im DOM,
- Sekundenlange Pausen für manuelle Bestätigung des Cookie-Banners.

Ein Fazit aus der Beobachtung ist der Tatsache, dass die Automatisierung von Webanwendungen nicht nur die Kenntnis von Selenium erfordert, sondern auch die Anpassung der Teststrategie an die spezifischen Vorgänge der Seite. Der Service *wsiz.edu.pl* verzerrt die Verwendung von Attributtypen für Testzwecke (wie data-testid), was die eindeutige Selektion von Elementen erschwert. Ein Teil der Links öffnet sich in einer neuen Karte, was eine zusätzliche Herausforderung darstellt.

dodatkowej obsługi przełączania kontekstu WebDrivera. Sam baner cookie nie zawsze wczytuje się w tym samym momencie, co również wpływa na deterministyczność testów.

Rezultaty testów wskazują, że kluczowe funkcje strony działają poprawnie, jednak pewne obszary mogłyby zostać zoptymalizowane pod kątem UX i automatyzacji. Szczególnie przydatne mogłyby być:

- stabilne identyfikatory elementów niezależne od warstwy wizualnej strony,
- wyeliminowanie niepotrzebnych przeładowań po otwarciu witryny,
- uwzględnienie oficjalnego API lub endpointów dla wyszukiwarki,
- bardziej przewidywalne zachowanie przy zmianie języka.

Podsumowując, postawione cele projektu zostały osiągnięte, a wykonane testy stanowią solidną podstawę do dalszego rozwoju automatyzacji. Projekt dostarczył także praktycznego doświadczenia z obsługą Selenium i realnymi problemami testowania współczesnych, dynamicznych interfejsów webowych. W przyszłości zestaw testów może stać się częścią ciągłego nadzoru nad dostępnością i poprawnością działania witryny, co przełożyłoby się na szybsze wykrywanie błędów i poprawę jakości usług oferowanych użytkownikom strony.