Zadanie 1.

Przetestowanie czy strona startowa localhost otwiera się prawidłowo.

W tym celu powinieneś:

1. Założyć folder TestDjango
2. Uruchomić środowisko wirtualne
3. Zainstalować selenium, pytest
4. Stworzyć projekt w Django (startproject)
5. Stworzyć plik test\_runbrowser.py

W pliku chcemy umieścić kod, który sprawdzi czy prawidłowo otwiera się strona.

Najpierw powinniśmy zaimportować webdriver z selenium. Następnie do zmiennej browser przypisujemy uruchomienie przeglądarki Firefox (webdriver.Firefox()). Kolejnym krokiem jest użycie funkcji get, w której umieścimy nasz adres (localhost). I na końcu sprawdzenie

assert ‘Django’ in browser.title

1. Uruchamiamy serwer
2. Uruchamiamy skrypt

Uwaga do następnego zadania

W Django w celu stworzenia testów jednostkowych importujemy TestCase z django.test, natomiast sama struktura pozostaje analogiczna jak w przypadku wcześniejszych ćwiczeń tzn.

**from** django.test **import** TestCase

**class** **ExampleTest**(TestCase):

**def** **test\_dodawanie**(self):

self.assertEqual(1 + 1, 3)

Zadanie 2.

Przetestowanie linków Djnago

W tym celu powinieneś:

1. Stworzyć aplikację Django (startapp)
2. W stworzonym katalogu utworzyć plik tests.py
   1. W pliku importujemy TestCase, resolve oraz home\_page z “nazwa katalogu z aplikacją”.views
   2. Tworzymy klasę HomePageTest
   3. W klasie powinna zostać zdefiniowana funkcja test\_url\_resolve, która zmiennej found przypisze resolve(‘/’) i nastąpi porównanie found.func z home\_page

Dla przypomnienia

Resolve to funkcja, której Django używa wewnętrznie do rozwiązywania adresów URL i znajdowania funkcji widoku, do której powinny one być mapowane. Sprawdzamy, czy rozwiązanie, gdy zostanie wywołane z „/”, katalogiem głównym witryny, znajdzie funkcję o nazwie home\_page

1. W pliku views wpiujemy home\_page = None
2. Dodanie w pliku (w projekcie) urls.py ścieżki gdzie będzie przyporządkowanie home\_page oraz nazwa ‘home’)
3. Uruchomienie testu jednoskowego za pomocą python manage.py test
4. W przypadku wystąpienia błędu należy poprawić pliku views.py

Zadanie 3 Napisanie testu dla widoku

W tym celu powinieneś:

1. Stworzyć obiekt HttpRequest, który zobaczy Django, gdy przeglądarka użytkownika zapyta o stronę.
2. Przekazanie go do naszego widoku home\_page, który daje nam odpowiedź.
3. Następnie wyodrębnienie odpowiedzi. Są to nieprzetworzone bajty, jedynki i zera, które byłyby przesyłane przewodem do przeglądarki użytkownika. Wywoływany jest .decode (), aby przekształcić je w ciąg kodu HTML, który jest wysyłany do użytkownika.

html = response.content.decode('utf8')

1. Plus zaimplementowanie testów
2. self.assertTrue(html.startswith('<html>'))
3. self.assertIn('<title>To-Do lists</title>', html)
4. self.assertTrue(html.endswith('</html>'))
5. Uruchomienie testu.
6. Poprawienie odpowiedniego pliku, a dokładnie jednej funkcji