

## Wydział Informatyki i Telekomunikacji

Języki skryptowe

## Projekt zaliczeniowy

## Jakub Krupiński 255356

1. Anons

Wykonany przeze mnie program jest stroną internetową stworzoną przy pomocy biblioteki ‘Django’ i stanowi alternatywę dla innych, popularnych od dawna stron, takich jak w Polsce ‘filmweb.pl’, czy też globalnie znanego – ‘IMDb.com’. Mogłaby ona stanowić ciekawą alternatywę dla tych użytkowników, którzy są ciekawi kina i chcieliby korzystać z szerokiej i zebranej w jednej lokalizacji bazy filmów, jako że działające obecnie na rynku serwisy posiadają bardzo zdecentralizowane listy danych, przez co szukanie odpowiednich pozycji potrafi być uciążliwie, jako że trzeba przechodzić między wieloma stronami internetowymi i ich zakładkami. Mój serwis jest również w pełni darmowy – nie posiada żadnych reklam, w przeciwieństwie do popularnych serwisów, które są nimi przepełnione, w dodatku często nie dając możliwości posiadania aktywnego AdBlocka, co bez wątpienia przeszkadza wielu osobom. Aby korzystać z serwisu konieczne jest posiadanie na swoim komputerze zainstalowanej przeglądarki, stabilnego wydania Pythona (najlepiej 3.9 lub wyżej) oraz zainstalowanie odpowiednich bibliotek w katalogu, w którym znajduje się kod źródłowy.

1. Opis najważniejszych funkcji

Program zapewnia użytkownikowi dostęp do baz filmów ze stron ‘imdb.com’ (najwyżej oceniane 250 filmów) oraz ‘filmweb.com’ (najpopularniejsze pozycje w bazie). Zdecydowałem się zaimplementować przykład nieco innej strony, niż w przypadku ‘imdb.com’ – co prawda serwis Filmweb również zapewnia ranking najwyżej ocenianych filmów, jednak uznałem, że ciekawsze będzie zestawienie ze sobą najwyżej ocenianych filmów razem z tymi najpopularniejszymi – będzie to oznaczało szerszą i bardziej zdywersyfikowaną bazę danych, która zadowoli każdego kinomaniaka. Funkcje, które zapewnia mój serwis, to wspomniane już zbieranie danych z sieci, dodawanie pobranych informacji do bazy danych na serwerze ‘Django’, czyszczenie tej samej bazy, filtrowanie po tytułach listy filmów, które znajdują się obecnie w serwisie oraz wyświetlanie szczegółowych danych dla każdej pozycji w spisie. Bezwarunkową zaletę mojego produktu względem jego konkurencji na rynku stanowi na pewno skonsolidowanie danych z wielu różnych stron w jednym miejscu, co ułatwia przeglądanie zawartości serwisu i znacznie dywersyfikuje jego zawartość; prosty, ale i treściwy interfejs, który zawiera wszystkie niezbędne funkcjonalności nie sprawiając przy tym żadnych problemów w obsłudze, niezależnie od wieku oraz obycia z technologią jego użytkownika oraz brak rozpraszających, nachalnych reklam, które potrafią znacznie pogorszyć doświadczenie korzystania z niejednego dostępnego obecnie na rynku serwisu.

1. Dokumentacja programowa
2. Stosowane narzędzia:

* Python
* HTML
* JSON

oraz biblioteki:

* Scrapy
* BeautifulSoup
* Django

1. Przewodnik po klasach

* MovieAdmin
* WebappConfig
* Data
* MovieListView
* MovieItem

1. Ciekawe fragmenty kodu

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 1. Wykonany przy użyciu biblioteki scrapy pająk pobierający dane z domeny ‘imdb.com’:



Rysunek 2. Automatyzacja terminalowych komend uruchamiających ściąganie danych z sieci:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 3: Wykonany przy pomocy biblioteki BeautifulSoup pająk ściągające dane z domeny 'fimweb.pl', korzystając z pierwszych 30 zakładek

1. Dokumentacja użytkownika

Instrukcja uruchomienia programu:

* znajdując się w głównym katalogu projektu możliwe jest uruchomienie programu przez wpisanie w terminalu komendy „python main.py”
* z lokalizacji głównego katalogu projektu możliwe jest uruchomienie programu przez wejście do folderu „dist” i uruchomienie pliku „main.exe”

W obu przypadkach na stronę główną można wejść linkiem: http://127.0.0.1:8000/

Obraz zawierający tekst

Opis wygenerowany automatycznie

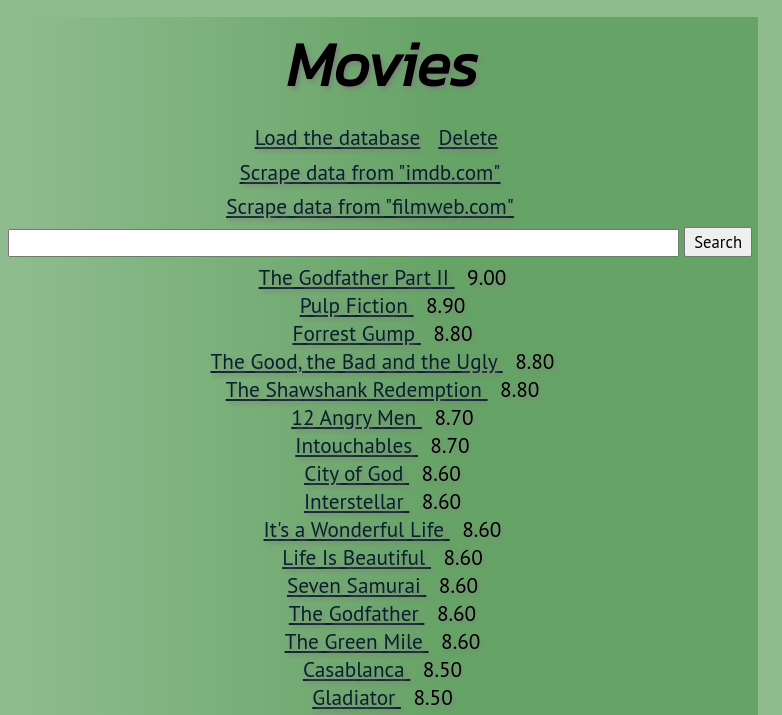
Rysunek 4: Widok głównej strony serwisu, gdy baza danych jest pusta



Rysunek 5: Widok wyświetlany użytkownikowi po tym, jak z sukcesem udało mu się pobrać dane z sieci



Rysunek 6: Widok wyświetlany użytkownikowi po tym, jak z sukcesem udało mu się załadować pobrane z sieci elementy do bazy danych

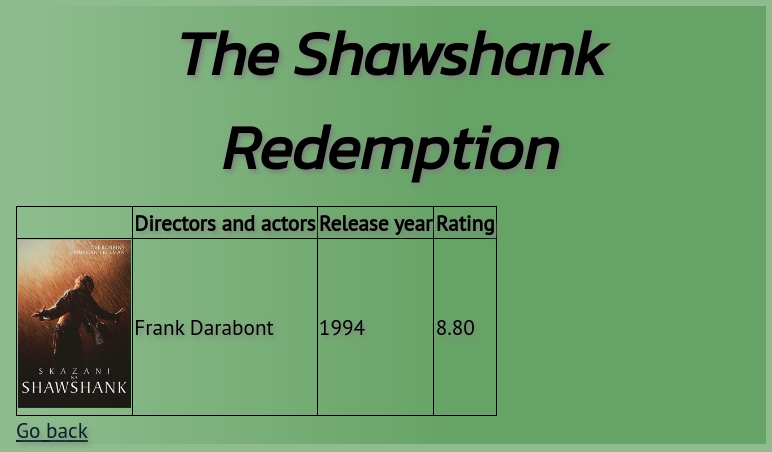


Rysunek 7: Widok początku głównej strony serwisu, gdy baza danych jest niepusta

Obraz zawierający tekst

Opis wygenerowany automatycznie

Rysunek 8: Widok wyświetlany użytkownikowi podczas korzystania z funkcjonalności filtrowania elementów

Rysunek 9: Widok wyświetlany użytkownikowi po udanym wyczyszczeniu zawartości bazy danych



Rysunek 10: Widok wyświetlany użytkownikowi po udanym wyczyszczeniu zawartości bazy danych