

Podstawy Teleinformatyki WebScraper / Metawyszukiwarka

Paweł Soja

Numer indeksu: 122031

pawel.soja@student.put.poznan.pl

Krzysztof Łuczak

Numer indeksu: 122008

krzysztof.t.luczak@student.put.poznan.pl

Dawid Wiktorski

Numer indeksu: 122056

dawid.wiktorski@student.put.poznan.pl

Spis treści

1	Opis i uzasadnienie wyboru tematu	3
1.1	Ogólny proces zbierania i przetwarzania danych	3
1.2	Uzasadnienie wyboru tematu	3
2	Organizacja pracy	4
2.1	Harmonogram prac	4
2.2	Podział prac pomiędzy członków zespołu	5
2.3	Środowisko pracy	5
3	Wymagania	6
3.1	Opis funkcjonalności	6
3.2	Wymagania funkcjonalne	6
3.3	Wymagania pozafunkcjonalne	7
4	Wybrane technologie i uzasadnienie	9
4.1	Biblioteka BeautifulSoup	9
5	Architektura rozwiązania	10
6	Interesujące problemy i ich rozwiązania	12
6.1	Moduł scrapujący sekurak.pl	12
6.2	Moduł scrapujący altcontroldelete.pl	12
7	Opis stron internetowych, z których zbierane są informacje	13
7.1	sekurak.pl	13
7.2	dobreprogramy.pl/Blog.html	13
7.3	niebezpiecznik.pl	13
7.4	pclab.pl/news.html	14
7.5	altcontroldelete.pl	14
8	Instrukcja użytkowania aplikacji	15
8.1	Strona główna dla użytkownika niezalogowanego	15
8.2	Strona główna dla użytkownika zalogowanego	15
8.3	Źródła #Tagi dla użytkownika niezalogowanego	17
8.4	Źródła #Tagi dla użytkownika zalogowanego	18
8.5	Wyszukiwarka	19
8.6	Profile	21
8.7	Ustawienia	23
8.8	Perspektywy rozwoju	24
9	Interesujące fragmenty implementacji	25

Spis rysunków

1	Schemat bazy danych	10
2	Strona główna dla użytkownika niezalogowanego	15
3	Strona główna dla użytkownika zalogowanego	16
4	Źródła #Tagi dla użytkownika niezalogowanego	17
5	Źródła #Tagi dla użytkownika zalogowanego	18
6	Wyszukiwarka - panel	19
7	Wyszukiwarka - wyniki	20
8	Profile - dodawanie	21
9	Profile - przegląd	22
10	Ustawienia	23

Spis tabel

1	Harmonogram prac	4
2	Podział prac	5
3	Funkcjonalności	6
4	Opis bazy danych	10
5	Parametry artykułów - sekurak.pl	13
6	Parametry artykułów - dobreprogramy.pl	13
7	Parametry artykułów - niebezpiecznik.pl	13
8	Parametry artykułów - pclab.pl/news.html	14
9	Parametry artykułów - altcontroldelete.pl	14

Spis listingów

1	Scraper pclab	25
2	Scraper pclab cd.	26

1 Opis i uzasadnienie wyboru tematu

Celem projektu jest zbudowanie platformy do zbierania i prezentowania danych z różnych stron internetowych. Platforma składa się z serwisu internetowego prezentującego dane użytkownikom zalogowanym oraz z aplikacji zbierających te dane.

Na potrzeby tego projektu i dokumentacji utworzone zostało pojęcie 'scrapowanie', które oznaczać będzie zbieranie danych ze stron internetowych poprzez odwiedzenie jej i zapisanie wybranych informacji do bazy danych.

1.1 Ogólny proces zbierania i przetwarzania danych

Informacje zbierane są przez tzw. scrapery, a następnie zapisywane w bazie danych. Scraper zbiera tylko te dane, które zostaną ustalone przez programistę. Dalsze filtrowanie odbywa się na poziomie aplikacji internetowej w oparciu o profil użytkownika lub podane parametry. Wyszukiwanie wykonywane jest w bazie danych. Dane zapisywane w bazie usuwane są po ustalonym czasie. Dlatego też aplikacja umożliwia filtrowanie wstecz, ale tylko do pewnej granicy. Dane zapisane w bazie można określić jako 'newsy'. Jeżeli informacja zostaje usunięta to znaczy, że jest już nieaktualna. Dzięki takiemu systemowi gromadzenia danych, aplikacja jest w stanie serwować użytkownikom najnowsze materiały, przy jednoczesnym zachowaniu wydajności filtrowania różnych źródeł. Z założenia użytkownik regularnie korzysta z aplikacji.

1.2 Uzasadnienie wyboru tematu

Temat wybraliśmy, ponieważ interesuje nas dziedzina przetwarzania danych. Chcielibyśmy poznać technologie scrapowania, parsowania stron internetowych oraz język Python, framework Django i technologie front-endowe tj. HTML5, Javascript. Jednocześnie nie znaleźliśmy zadowalającego nas serwisu, który udostępniałby takie usługi, dlatego sami zdecydowaliśmy zrobić swój.

2 Organizacja pracy

Przy pracy nad projektem, korzystano z repozytorium GitHub. Link do repozytorium:
[WebSraper/Metawyszukiwarka](#)

2.1 Harmonogram prac

Orientacyjny harmonogram prac został przedstawiony w tablicy 1. Wyszczególniono zadania oraz osobę/osoby zajmujące się danym fragmentem projektu.

Tablica 1: Harmonogram prac

Lp.	Opis	Miesiąc
1.	Wybór technologii, modułów, podział pracy	Marzec
2.	Wstępna dokumentacja, planowanie serwisu	Kwiecień
3.	Zapoznanie z technologią, testy bibliotek	Kwiecień
4.	Baza danych, pierwszy moduł, interfejs	Maj
5.	Kolejne moduły	Maj
6.	Testowanie serwisu, poprawki	Czerwiec
7.	Zakończenie prac nad serwisem	Czerwiec

2.2 Podział prac pomiędzy członków zespołu

W tablicy 2 przedstawiono podział prac pomiędzy członków zespołu.

Tablica 2: Podział prac

Lp.	Opis	Osoby
1.	Baza danych	Wszyscy
2.	Projekt interfejsu	Paweł Soja
3.	Front-end serwisu	Paweł Soja
4.	Back-end serwisu	Krzysztof Łuczak, Dawid Wiktorski
5.	Moduł I	Krzysztof Łuczak
6.	Moduł II	Dawid Wiktorski
7.	Moduł III	Dawid Wiktorski
8.	Moduł IV	Dawid Wiktorski
9.	Moduł V	Dawid Wiktorski
10.	Testowanie	Wszyscy

2.3 Środowisko pracy

- IDE PyCharm,
- TeXstudio,
- przeglądarki internetowe: Google Chrome oraz Mozilla Firefox.

3 Wymagania

3.1 Opis funkcjonalności

Aktorzy systemu:

- użytkownik
 - użytkownik zalogowany - posiada prawa do użytkowania serwisu,
 - użytkownik niezalogowany - może dokonać rejestracji,
 - administrator - zarządza serwisem,
- aplikacja internetowa - prezentuje dane,
- moduł zbierający dane (scraper) - zbiera i przetwarza dane.

3.2 Wymagania funkcjonalne

Wymagania funkcjonalne systemu scharakteryzowano w tablicy 3.

Tablica 3: Funkcjonalności

Funkcja	Opis	Aktorzy
Przeglądanie strony głównej	Możliwość przeglądania strony głównej serwisu.	Użytkownicy
Logowanie	Możliwość logowania się do serwisu.	Użytkownik niezalogowany
Wylogowanie	Możliwość wylogowania się z serwisu.	Użytkownik zalogowany, administrator
Rejestracja	Możliwość zarejestrowania konta w serwisie.	Użytkownik niezalogowany
Potwierdzenie rejestracji linkiem aktywacyjnym.	Po założeniu konta wysyłany jest link aktywacyjny na adres e-mail podany podczas rejestracji. Dopóki konto nie zostanie aktywowane poprzez kliknięcie w link, nie będzie można zalogować się.	Użytkownik niezalogowany

Tablica 3 – *Kontynuacja*

Funkcja	Opis	Aktorzy
Zmiana hasła do konta	Możliwość zmiany hasła do aktywnego konta.	Użytkownik zalogowany, administrator
Potwierdzenie zmiany hasła linkiem aktywacyjnym.	Po zmianie hasła, konto jest deaktywowane i wysyłany jest link aktywacyjny na adres e-mail podany podczas rejestracji. Dopóki konto nie zostanie aktywowane poprzez kliknięcie w link, nie będzie można zalogować się.	Użytkownik niezalogowany
Zmiana adresu e-mail konta	Możliwość zmiany adresu e-mail konta.	Użytkownik zalogowany, administrator
Potwierdzenie zmiany adresu e-mail linkiem aktywacyjnym.	Po zmianie hasła, konto jest deaktywowane i wysyłany jest link aktywacyjny na nowy adres e-mail. Dopóki konto nie zostanie aktywowane poprzez kliknięcie w link, nie będzie można zalogować się.	Użytkownik niezalogowany
Możliwość wyszukiwania artykułów na podstawie źródeł, tagów i przedziału czasowego	Możliwość ustalenia źródeł, tagów i wybrania przedziału czasowego, w celu przefiltrowania interesujących informacji.	Użytkownik zalogowany, administrator
Dodanie profilu z konfiguracją źródeł i tagów	Możliwość wybrania tagów i źródeł, na podstawie których filtrowane będą informacje.	Użytkownik zalogowany, administrator
Zbieranie danych ze strony i parsowanie ich	Scraper zbiera dane ze strony i parsuje je. Jeden scraper zbiera dane z jednej strony.	Scraper
Zapis artykułów do bazy danych	Po zebraniu i przetworzeniu artykułów przez scraper, następuje zapis do bazy danych z uwzględnieniem narzuconego formatu.	Scraper

3.3 Wymagania pozafunkcjonalne

- zainstalowany interpreter języka Python w wersji 3.5 lub wyższej,
- zainstalowany framework Django w wersji 1.10.6,
- zainstalowana biblioteka BeautifulSoup w wersji 4.5.3,
- język interfejsu użytkownika: polski,

- bezpieczne przechowywanie haseł w formie zahaszowanej
- budowa modułowa aplikacji, umożliwiająca dodawanie nowych scraperów.

4 Wybrane technologie i uzasadnienie

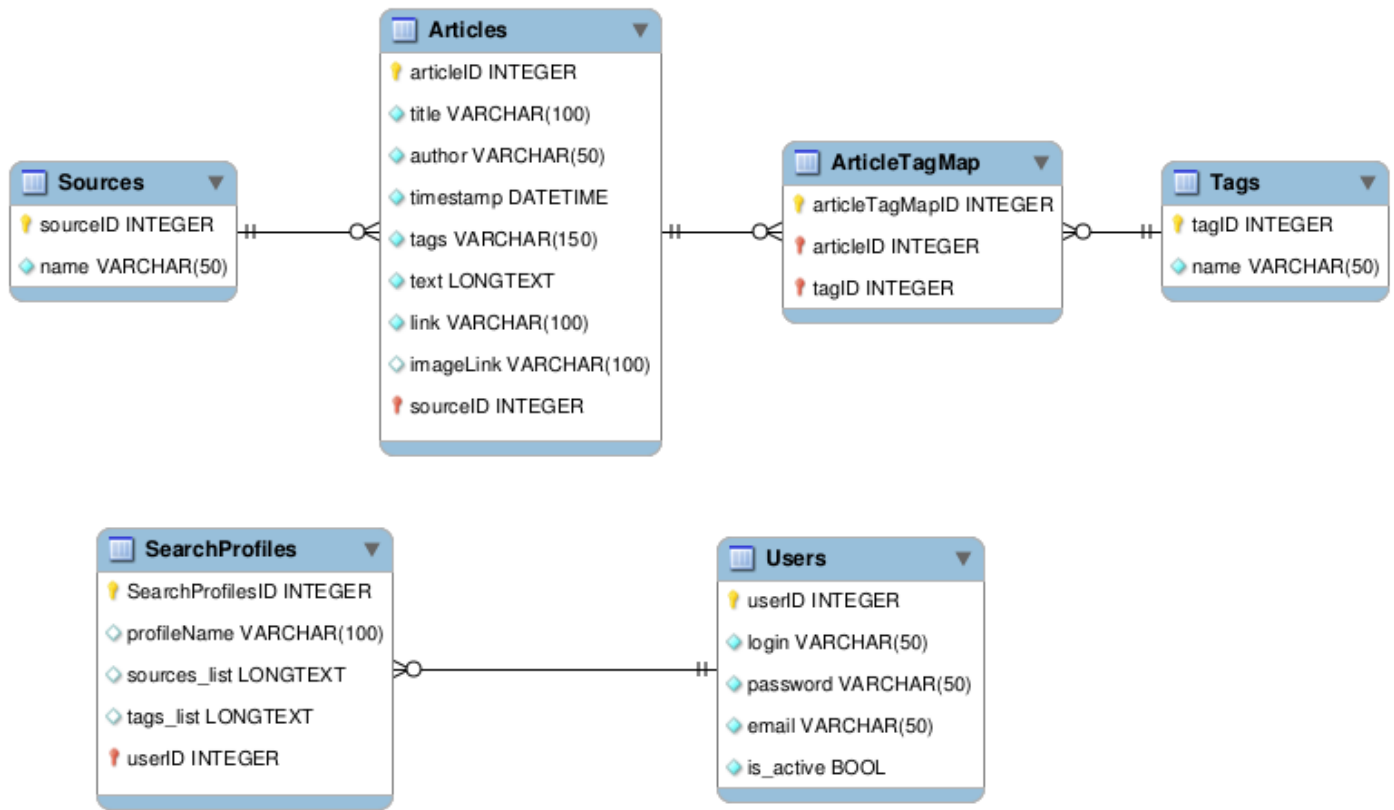
- Back-end - Python, Django
 - stosunkowo krótki czas tworzenia aplikacji przy jednoczesnym zachowaniu:
 - * pełnej funkcjonalności,
 - * stabilności,
 - * wydajności
- scraping - Python, BeautifulSoup
 - przejrzysta dokumentacja,
 - dobra wydajność,
 - duże możliwości konfiguracyjne,
 - prosta obsługa.
- Front-end - HTML5, Javascript
 - uniwersalna technologia, która jest wspierana przez wszystkie popularne przeglądarki internetowe
- Baza danych - SQLite
 - łatwa integracja z językiem Python,
 - dobra we wstępnej fazie projektu,
 - nie wymaga instalacji zewnętrznych bibliotek, programów do obsługi.

4.1 Biblioteka BeautifulSoup

Bibliotek do wyciągania danych ze stron internetowych jest wiele, są to między innymi: Scrapy, BeautifulSoup, Urllib2, MarkupSafe oraz feedparser. W projekcie wykorzystaliśmy bibliotekę BeautifulSoup, która przeznaczona jest dla języka Python. Umożliwia wyciąganie danych z plików HTML oraz XML. BeautifulSoup w wersji 4 współpracuje z takimi parserami jak: Python `html.parser`, `lxml HTML parser`, `lxml XML parser` i `htm5lib`. Po przeprowadzonych testach, `lxml HTML parser` okazał się najszybszym oraz najbardziej niezawodnym parserem spośród wyżej wymienionych. Dużą zaletą biblioteki BeautifulSoup jest łatwa implementacja w architekturze modułowej.

5 Architektura rozwiązania

W projekcie została wykorzystana relacyjna baza danych SQLite. Na rysunku 1 przedstawiono schemat relacyjny bazy danych.



Rysunek 1: Schemat bazy danych

W tablicy 4 scharakteryzowano bazę danych.

Tablica 4: Opis bazy danych

Tabela	Opis
Articles	Zawiera wszystkie sparsowane strony.
Tags	Zawiera wszystkie dostępne tagi. Dodanie nowego taga odbywa się automatycznie, gdy scraper podczas parsowania wykryje, że danego taga jeszcze nie ma w bazie.
ArticleTagMap	Łączy daną stronę z odpowiednim tagiem.

Tablica 4 – *Kontynuacja*

Tabela	Opis
Sources	Zawiera wszystkie dostępne źródła, czyli strony internetowe, z których zbieramy dane. Dodanie odbywa się ręcznie. Administrator musi napisać moduł dla danej strony.
Users	Zawiera wszystkich użytkowników serwisu.
SearchProfiles	Zawiera listę źródeł i tagów przypisanych do użytkownika.

6 Interesujące problemy i ich rozwiązania

Podczas implementacji modułów, w każdym z nich, spotkano się z problemem wyciągania potrzebnych danych ze stron internetowych. Przy parsowaniu stron, najczęściej nieznanym elementem w artykule okazał się link do obrazka. Rozwiązaniem problemu było najpierw ustawienie domyślnego obrazka dla danego modułu oraz wykorzystanie go, gdy parser nie poradził sobie ze znalezieniem obrazka w artykule.

6.1 Moduł scrapujący sekurak.pl

Serwis sekurak.pl zawiera dwa rodzaje artykułów, które pobieramy. Pierwszy rodzaj znajduje się w kategorii "teksty", a drugi "w biegu". Scraper został podzielony na dwa wątki, każdy dla jednej z tych kategorii. To spowodowało przyspieszenie procesu zbierania o około 50%. Algorytm sekwencyjny dla jednej strony artykułów wykonywał się około 24 s. Algorytm wielowątkowy około 14 s.

6.2 Moduł scrapujący altcontroldelete.pl

W serwisie altcontroldelete.pl data publikacji artykułu może być podana w języku polskim lub angielskim. Rozwiązaniem problemu było utworzenie dwóch słowników, które w odpowiedni sposób zinterpretują podaną datę.

7 Opis stron internetowych, z których zbierane są informacje

Strony internetowe, z których zbierane są dane to:

- www.sekurak.pl,
- www.dobreprogramy.pl/Blog.html
- www.niebezpiecznik.pl
- www.pclab.pl/news.html
- www.altcontroldelete.pl

Poniżej w podrozdziałach przedstawiono dane, które są zbierane z każdej ze stron internetowych.

7.1 [sekurak.pl](http://www.sekurak.pl)

W tablicy 5 pokazano dane, które są zbierane ze strony [sekurak.pl](http://www.sekurak.pl)

Tablica 5: Parametry artykułów - [sekurak.pl](http://www.sekurak.pl)

Tytuł	Data opublikowania	Tagi	Obrazek	Fragment tekstu	Link
-------	--------------------	------	---------	-----------------	------

7.2 [dobreprogramy.pl/Blog.html](http://www.dobreprogramy.pl/Blog.html)

W tablicy 6 pokazano dane, które są zbierane ze strony [dobreprogramy.pl/Blog.html](http://www.dobreprogramy.pl/Blog.html)

Tablica 6: Parametry artykułów - [dobreprogramy.pl](http://www.dobreprogramy.pl)

Tytuł	Data opublikowania	Tagi	Autor	Fragment tekstu	Link
-------	--------------------	------	-------	-----------------	------

7.3 [niebezpiecznik.pl](http://www.niebezpiecznik.pl)

W tablicy 7 pokazano dane, które są zbierane ze strony [niebezpiecznik.pl](http://www.niebezpiecznik.pl)

Tablica 7: Parametry artykułów - [niebezpiecznik.pl](http://www.niebezpiecznik.pl)

Tytuł	Data opublikowania	Tagi	Autor	Obrazek	Fragment tekstu	Link
-------	--------------------	------	-------	---------	-----------------	------

7.4 pclab.pl/news.html

W tablicy 8 pokazano dane, które są zbierane ze strony pclab.pl/news.html

Tablica 8: Parametry artykułów - pclab.pl/news.html

Tytuł	Data opublikowania	Tagi	Autor	Obrazek	Fragment tekstu	Link
-------	--------------------	------	-------	---------	-----------------	------

7.5 altcontroldelete.pl

W tablicy 9 pokazano dane, które są zbierane ze strony altcontroldelete.pl

Tablica 9: Parametry artykułów - altcontroldelete.pl

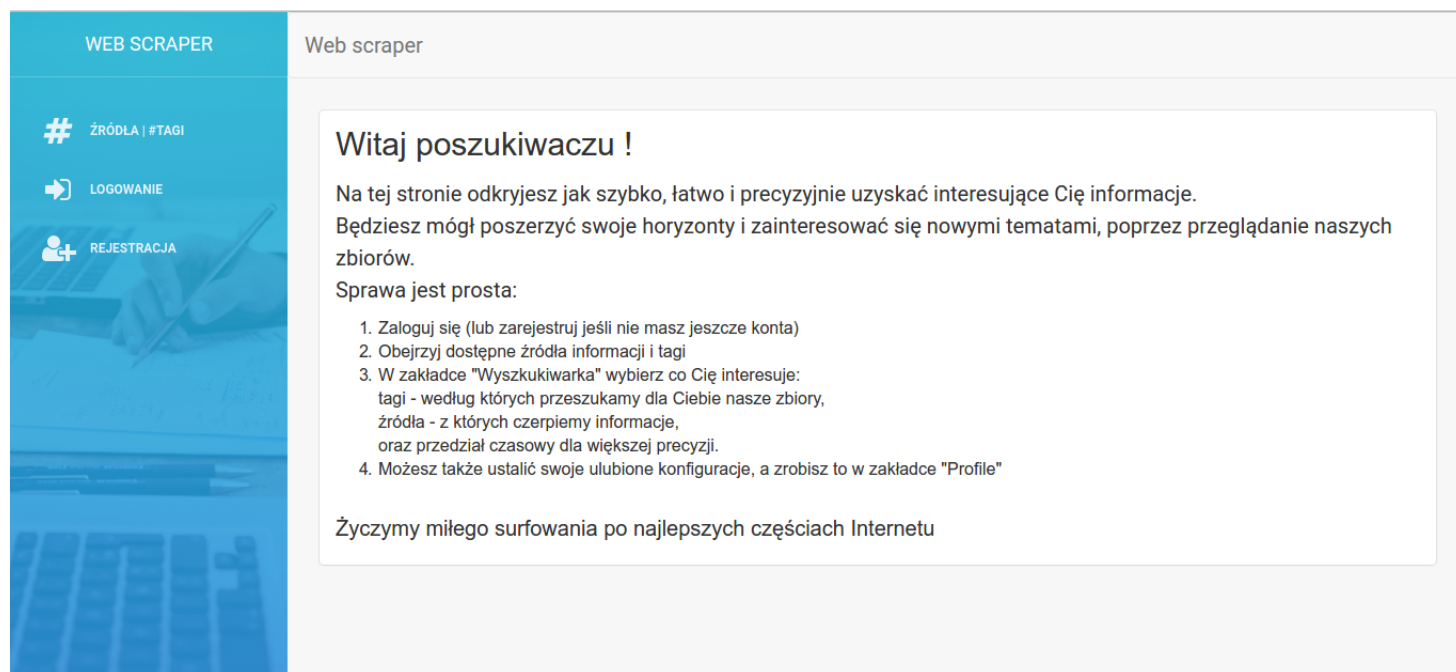
Tytuł	Data opublikowania	Tagi	Autor	Obrazek	Fragment tekstu	Link
-------	--------------------	------	-------	---------	-----------------	------

8 Instrukcja użytkowania aplikacji

W tym rozdziale przedstawiono interfejs strony internetowej wraz z instrukcją użytkowania.

8.1 Strona główna dla użytkownika niezalogowanego

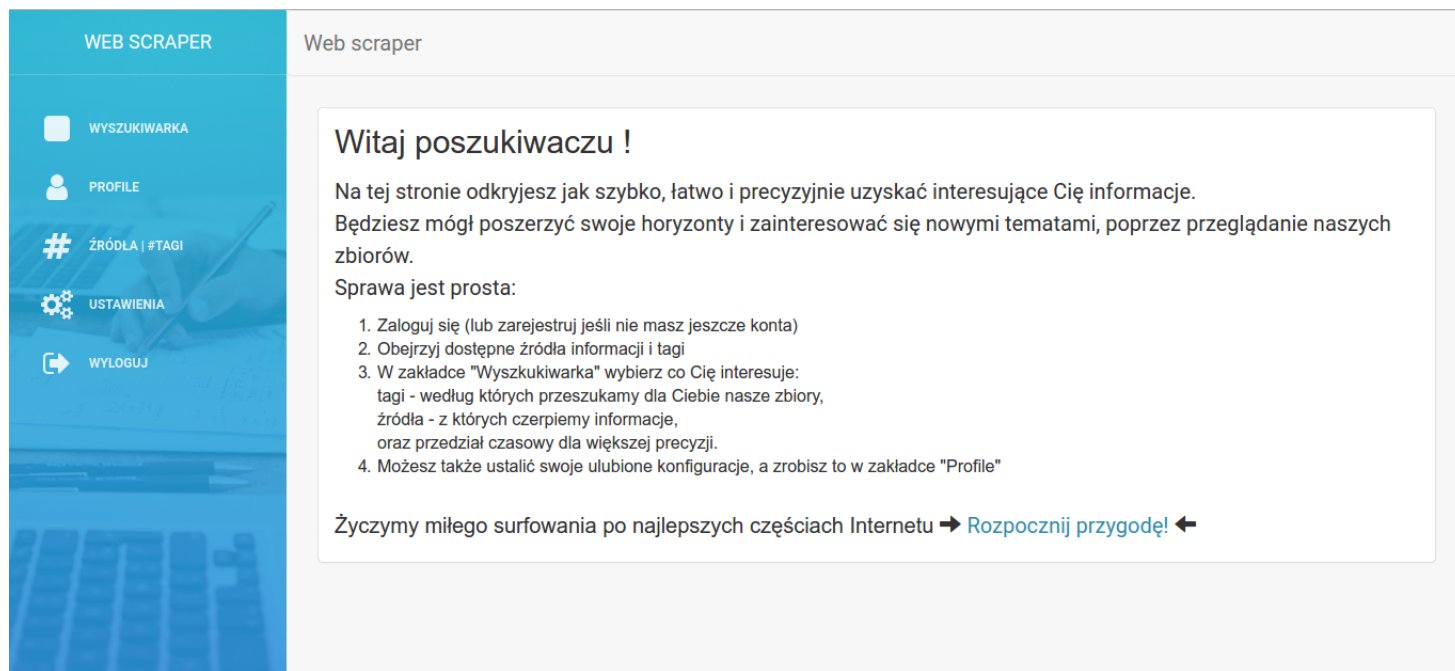
Po wejściu na stronę główną, użytkownikowi ukazuje się krótki opis funkcjonalności serwisu. Z tego miejsca użytkownik może przejść do kolejnej strony ze źródłami i tagami. Również ma możliwość zalogowania się lub założenia konta. Gdy, użytkownik poda złe dane w procesie logowania lub rejestracji, na stronie głównej pojawiają się odpowiednie komunikaty. Na rysunku 2 ukazano stronę główną.



Rysunek 2: Strona główna dla użytkownika niezalogowanego

8.2 Strona główna dla użytkownika zalogowanego

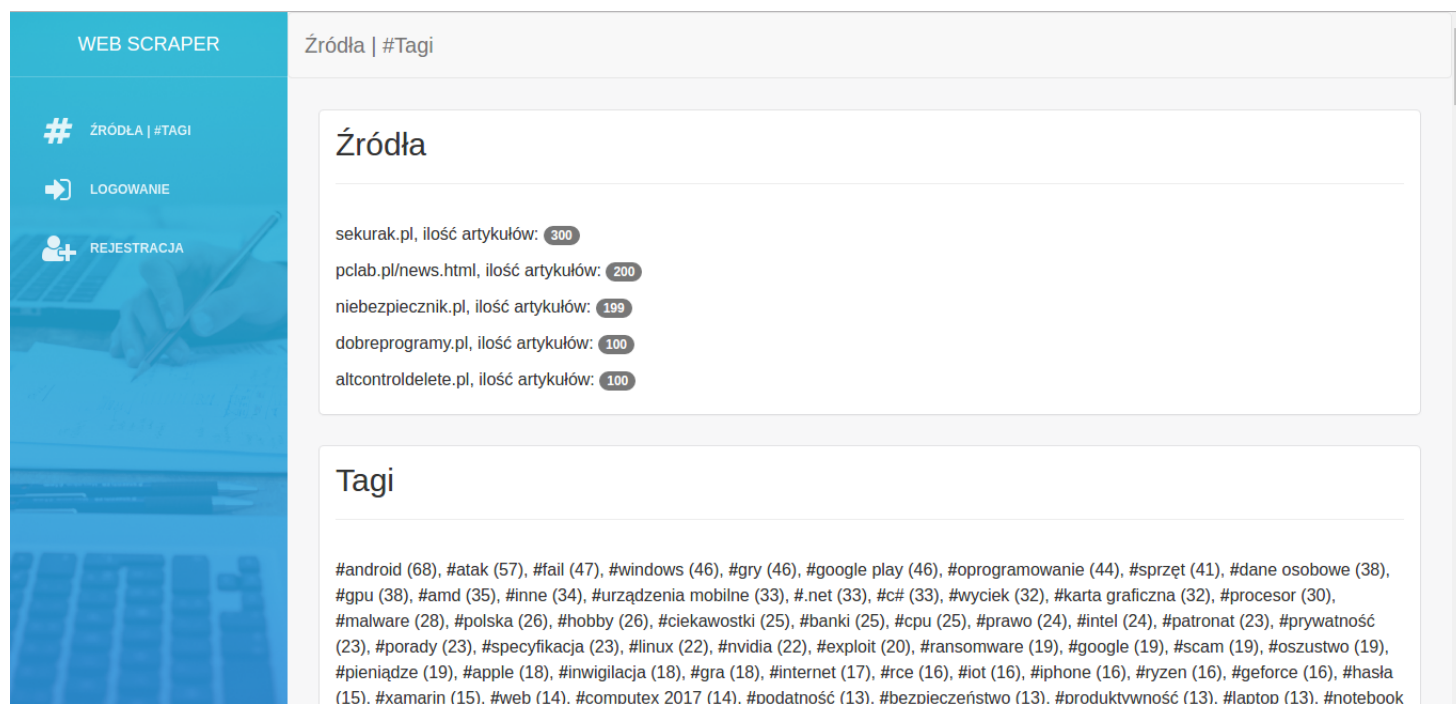
Wygląda podobnie do strony dla niezalogowanych użytkowników, ale dodatkowo po lewej stronie widać menu oraz w opisie znajdziemy link kierujący do wyszukiwarki.



Rysunek 3: Strona główna dla użytkownika zalogowanego

8.3 Źródła | #Tagi dla użytkownika niezalogowanego

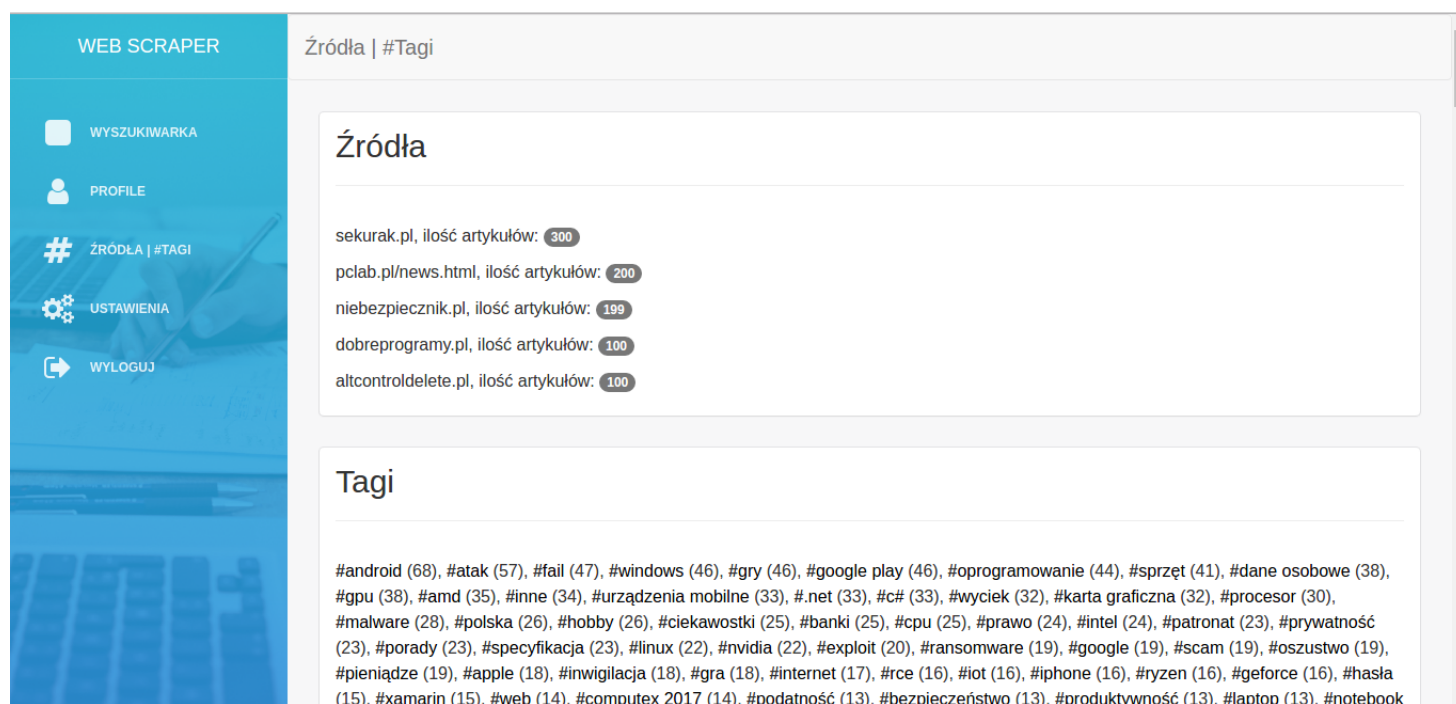
Na stronie związanej ze źródłami i tagami, użytkownik niezalogowany ma możliwość przejrzania z jakich stron internetowych zbierane są dane. Przy każdym źródle pokazana jest liczba artykułów, które obecnie przechowywane są w bazie danych. Poniżej źródeł, wymienione są tagi, które również przechowywane są w bazie danych. Obok każdego z tagów występuje liczba, która wskazuje, ile jest artykułów związanych z danym tagiem. Na rysunku 3 ukazano stronę ze źródłami i tagami.



Rysunek 4: Źródła | #Tagi dla użytkownika niezalogowanego

8.4 Źródła | #Tagi dla użytkownika zalogowanego

Użytkownik zalogowany na stronie ze źródłami i tagami ma możliwość nie tylko przejrzenia z jakich stron internetowych zbierane są dane i jakie tagi występują w bazie danych. Strona ta została rozszerzona o dodatkowe funkcjonalności. Użytkownik po kliknięciu na dany tag, otrzymuje listę powiązanych artykułów z wybranym tagiem. Dodatkowo, po kliknięciu na dany artykuł, użytkownik przechodzi bezpośrednio do strony internetowej z artykułem. Na rysunku 6 ukazano stronę ze źródłami i tagami dla użytkownika zalogowanego, po wybraniu tagu oprogramowanie.



Rysunek 5: Źródła | #Tagi dla użytkownika zalogowanego

8.5 Wyszukiwarka

Ta zakładka oferuje główną funkcjonalność serwisu. W pierwszym polu wybieramy tagi (maksymalnie 7) - dzięki technologii JavaScript pole umożliwia aktywne proponowanie tagów. Nie istnieje możliwość wyboru tagu, którego nie ma w bazie danych.

WEB SCRAPER

Wyszukiwarka

Wyszukiwarka

TAGI (MAX 7):

ŹRÓDŁA ("WSZYSTKIE"):

wszystkie x

Od: dd.mm.rrrr Do: dd.mm.rrrr

Szukaj

Rysunek 6: Wyszukiwarka - panel

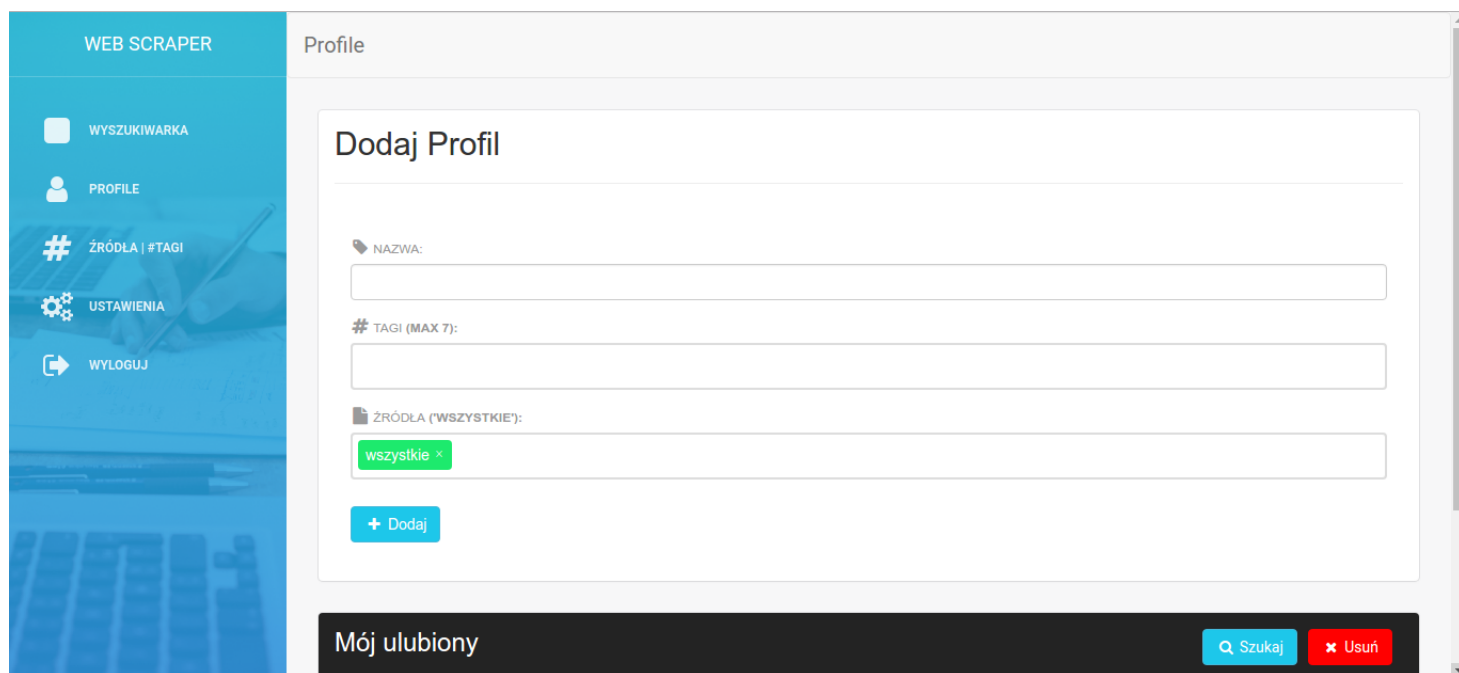
Poniżej panelu wyszukiwania pojawia się wybrana przez nas konfiguracja, a jeszcze niżej wyniki wyszukiwania. Widzimy tytuł artykułu, obrazek i fragment tekstu artykułu, tagi, źródło, autora oraz datę publikacji.

The screenshot displays a web scraper application interface. On the left is a blue sidebar with the title 'WEB SCRAPER' and several menu items: 'WYSZUKIWARKA' (Search), 'PROFILE', 'ŹRÓDŁA | #TAGI' (Sources | Tags), 'USTAWIENIA' (Settings), and 'WYLOGUJ' (Logout). The main content area has a dark header with filters: 'Wybrane tagi: windows | linux | programowanie |' and 'Wybrane źródła: wszystkie |'. Below this, a time range is set from '11 maja 2017' to '01 czerwca 2017', and it shows 'Znaleziono: 28 artykułów'. The search results section features an article titled 'Benq W2000 — krótki przegląd projektora multimedialnego' with a blue 'Artykuł' button. The article text is in Chinese, starting with '克羅斯多佛' and discussing the Benq W2000 projector. At the bottom of the article preview, it lists tags: '# Tagi: windows | oprogramowanie | internet |', source: 'Źródło: dobreprogramy.pl', author: 'Autor: Shaki81', and date: 'Data: 31 maja 2017'.

Rysunek 7: Wyszukiwarka - wyniki

8.6 Profile

Zakładka "Profile" umożliwia stworzenie dowolnej ilości profili wyszukiwania oraz bezpośrednie przejście do wyszukiwania poprzez naciśnięcie przycisku Szukaj".



WEB SCRAPER

Profile

Dodaj Profil

NAZWA:

TAGI (MAX 7):

ŹRÓDŁA (WSZYSTKIE):

wszystkie x

+ Dodaj

Mój ulubiony

Szukaj

Usuń

Rysunek 8: Profile - dodawanie

Dodano profil o nazwie "Mój ulubiony"

The screenshot displays the 'WEB SCRAPER' application interface. On the left is a blue sidebar with navigation options: 'WYSZUKIWARKA', 'PROFILE', 'ŹRÓDŁA | #TAGI', 'USTAWIENIA', and 'WYLOGUJ'. The main area is divided into two sections. The top section is a form for creating or editing a profile, with fields for 'NAZWA:', '# TAGI (MAX 7):', and 'ŹRÓDŁA ("WSZYSTKIE"):', each followed by a text input box. The 'ŹRÓDŁA' field contains a green pill labeled 'wszystkie'. A blue '+ Dodaj' button is at the bottom of the form. The bottom section, titled 'Mój ulubiony', has a dark background and contains the text 'Źródła: wszystkie' and '# Tagi: allegro | wykop | hackme | .net | chrome | spoofing | kryptografia'. In the top right of this section are two buttons: a blue 'Szukaj' button and a red 'Usuń' button.

Rysunek 9: Profile - przegląd

8.7 Ustawienia

W ustawieniach, użytkownik ma możliwość zmiany hasła, zmiany adresu e-mail oraz usunięcia swojego konta. Na tej stronie, pojawiają się również komunikaty związane z powyżej wymienionymi akcjami, które użytkownik jest w stanie zrobić. Gdy użytkownik zmieni swój adres e-mail, automatycznie zostanie wylogowany i przekierowany do strony głównej. Na rysunku 7 ukazano stronę z ustawieniami.

WEB SCRAPER

Ustawienia

Ustawienia

AKTUALNE HASŁO:

NOWE HASŁO:

NOWE HASŁO:

Zapisz

E-MAIL:

wszyscy@student.put.poznan.pl

Zapisz

Usuń konto

Rysunek 10: Ustawienia

8.8 Perspektywy rozwoju

Aplikacja została zbudowana w oparciu o moduły scrapujące oraz część odpowiedzialną za komunikację z bazą danych i prezentację danych. Istnieje możliwość dołączania modułów scrapujących w celu rozszerzenia oferowanych informacji. W ostatecznej produkcyjnej wersji trzeba pomyśleć o zmianie systemu zarządzania bazą danych z SQLite na inną, skalowalną i dostosowaną do przetwarzania dużej ilości danych, np. PostgreSQL, MySQL.

9 Interesujące fragmenty implementacji

W tej części dokumentacji zostaną przedstawione najciekawsze fragmenty kodu. Poniższy listing pokazuje w jaki sposób został zaimplementowany algortm, który zbiera dane ze strony [www.pclab.pl/news.html](http://pclab.pl/news.html). Na początku w zmiennej `webpage` zapisywana jest cała strona internetowa, następnie za pomocą odpowiedniej metody w BeautifulSoup zostaje sparsowana.

Pierwsze szukane elementy na stronie noszą nazwę `element`. W elementach tych, umieszczone są artykuły. W kolejnych liniach kodu, w każdym z wyszukanych elementów, znajdują się kolejne niezbędne dane, między innymi: link do artykułu, autor i tagi. W kolejnej pętli ustalany jest odpowiedni format zapisywania tagów, aby tagi oddzielane były tylko znakiem przecinka.

```
1  for page in range(1, pages + 1):
2      webpage = requests.get("http://pclab.pl/news-" + str(page) + ".html")
3      soup = BeautifulSoup(webpage.content, 'lxml')
4
5      print("Page:" + str(webpage.url))
6
7      pclab = soup.find_all(class_="element")
8      for i in pclab:
9          try:
10             link = "http://pclab.pl" + i.find(class_="title").a['href']
11             text = i.find(class_="text").p.text
12             title = i.find(class_="title").a.text
13             temp_info = i.find(class_="info").text
14             author, date, c, d = temp_info.split("|")
15             date = date[1:-1]
16             date = pclab_date2_python_date(date)
17             image_link = "http://pclab.pl" + i.find(class_="text").a.img['src']
18             tags = i.find(class_="tags").text
19             a, tags = tags.split("Tagi:")
20         except:
21             print("Error:", link)
22             continue
23
24         tags_list = []
25         for tmp in tags.split(","):
26             tmp = tmp.replace("\n", "")
27             if tmp[0] == " ":
28                 tmp = tmp[1:]
29             if tmp[-1] == " ":
30                 tmp = tmp[:-1]
31
32             tags_list.append(tmp)
33
```

Listing 1: Scraper pclab

Na Listingu 2. następuje zebranie wcześniejszych danych do słownika `one_article` oraz dodanie artykułu do kolejki `ARTICLES`. Następnie artykuły pobierane są w aplikacji Django i zapisywane do bazy danych.

```
1  one_article = {  
2      "title": title , "date": date , "author": author ,  
3      "link": link , "tags": tags_list , "text": text ,  
4      "image_link": image_link  
5  }  
6  ARTICLES.put(one_article)  
7
```

Listing 2: Scraper pclab cd.