

09.06.2019

Artur Dylewski

Ivan Novgorodtsev

Przemysław Pernak

## Podstawy teleinformatyki

# **Porównywarka kursów bukmacherskich (webscrapper)**

## 1. Wstęp

Projekt zakłada stworzenie strony internetowej, która będzie przedstawiała kursy polskich bukmacherów na poszczególne mecze piłkarskie. Głównym elementem projektu są jednak moduły pobierające (*scrapujące*) dane ze stron internetowych bukmacherów. Moduły te wyszukują na stronach pożądane dane (ligi, mecze, kursy) i za pomocą stworzonych funkcji zapisuje je do bazy danych. Następnie specjalnie napisany algorytm dokonuje przypisania do jednej drużyny odpowiednich identyfikatorów dla tabel drużyn dla poszczególnych bukmacherów. Ostatecznie kursy te są prezentowane na stronie internetowej.

Do napisania silnika scrapującego, algorytmu przypisującego id do zespołów i backendu strony wykorzystano język Python wraz z bibliotekami i frameworkami:

- requests - służy do wykonywania zapytań do stron internetowych
  - BeautifulSoup - służy do wyciągania danych z plików html
  - selenium - pozwala na automatyzowanie pracy przeglądarki
  - sqlite3 - pozwala na stworzenie bazy danych w systemie SQLite
  - Flask - microframework pozwalający stworzyć serwer
  - difflib - comparing sequences
  - itertools - functions creating iterators for efficient looping
  - re - regular expression operations
  - Bootstrap - biblioteka css rozwijana przez twórców twittera
- 
- requests - najpopularniejsza biblioteka Pythona oferująca wiele zaawansowanych opcji; w naszym projekcie służy jednak jedynie do wysyłania zapytania GET do przeglądarki, jednak z racji swojej niezawodności i tak zdecydowaliśmy się na nią

- beautifulsoup - prosta, a jednocześnie bardzo potężna biblioteka pozwalająca na parsowanie kodu HTML i XML; wykorzystana do pobierania danych o kursach ze stron Fortuny i Forbeta oraz o tabelach ze strony Fortuny, jest niezbędna w projekcie; alternatywą była biblioteka lxml, ale bardziej niż na szybkości zależało nam na prostym poruszaniu po niezbyt uporządkowanych danych
- selenium - framework dostępny na wiele platform, używany przez profesjonalistów do wykonywania zautomatyzowanych testów; z racji na swoją prostotę idealnie nadał się nam do symulacji użytkownika "przeklikującego" poszczególne ligi na stronach, gdzie do wydobycia interesujących danych potrzebne było działanie użytkownika, a także do wydobycia kodu zawartego w JavaScript
- Wybór biblioteki Flask jeśli chodzi o część serwerową był podyktowany faktem, iż jest to bardzo prosta biblioteka idealna dla początkujących, co podkreśla sam autor na swojej stronie głównej.
- DiffLib - moduł dostarczający klasy oraz funkcje umożliwiające porównywanie struktur danych. Może być przykładowo użyty do porównywania plików, generowania różnic pomiędzy danymi w różnym formacie, łącznie z HTMLem, context i unified diffs.
- Itertools - moduł implementujący iteratory umożliwiające szybkie i efektywne operacje na zbiorach danych
- re - moduł dostarczający wyrażenia regularne, z których korzystaliśmy podczas zbierania danych ze stron bukmacherskich
- Bootstrap-popularna i darmowa biblioteka pozwalająca na tworzenie reaktywnych stron internetowych

## 2.      **Podział prac:**

Zakładany podział prac:

IN:

- algorytm przypisujący id tym samym zespołom na różnych stronach

pod różnymi nazwami

AD:

- dokończenie pobierania iForbet
- scraping kolejnej strony

PP:

- wykonanie strony dla prezentacji wyników
- scraping kolejnej strony

Ostateczny podział prac:

IN:

- zescrapowanie tabel ze stron bukmacherów na potrzeby algorytmu
- algorytm przypisujący id tym samym zespołom na różnych stronach pod różnymi nazwami

AD

- dokończenie pobierania iForbet
- zescrapowanie kursów ze stron Milenium i LVBet
- łączenie przez proxy
- obsługa błędów

PP:

- wykonanie strony dla prezentacji wyników
- zescrapowanie kursów ze strony STS

### **3. Funkcjonalności aplikacji**

Funkcjonalności, które przygotowaliśmy dla użytkowników w naszej aplikacji to przede wszystkim dostęp do zescrapowanych danych. Dzięki zakładce “wyszukiwania” na naszej stronie mamy możliwość podejrzenia najlepszych

kursów meczy piłkarskich z danej ligi. Użytkownik dostaje między innymi takie informacje jak: nazwy drużyn rywalizujących, kurs zwycięski dla drużyny, kurs dla remisu oraz kursy na konkretny wynik. Do tego podawane jest źródło jednej z czterech stron internetowych, z której pochodzi wynik. Ponad to użytkownik posiada możliwość szybkiego dostępu do informacji na temat trzech najpopularniejszych lig: "Copa America", "Liga mistrzów" i "Mistrzostwa świata" z każdej strony poprzez panel "szybki dostęp".

W celu zachowania wygody przeszukiwania wyników tam gdzie to konieczne zadbałmy o okienko filtrowania wyników po nazwach "drużyn" i "lig". To ułatwia użytkownikowi znalezienie konkretnej drużyny z pośród naprawdę ich dużej liczby na naszej stronie.

W temacie funkcjonalności nasz projekt mógłby oczywiście być jeszcze bardziej rozbudowany. Zdajemy sobie sprawę, że to co zawiera jest podejściem minimalistycznym. Wynika to z faktu, że główny nacisk kładliśmy na scrapowanie stron co mimo to wywołało niemały problem co opiszemy w 5 punkcie.

Jeśli użytkownik jest bardziej zaawansowany może samodzielnie korzystać z przygotowanych skryptów i korzystać z bazy danych wedle własnego uznania. Funkcje *scrap* umieszczone w modułach odpowiedzialnych za pobieranie danych ze stron bukmacherów pozwalają na uruchamianie tych modułów z poziomu linii komend. Aby aplikacja była w pełni funkcjonalna należy wcześniej uruchomić moduł `main.py`.

#### **4. Architektura rozwiązania**

Fundamentem aplikacji są dane zescrapowane ze stron bukmacherskich.

Aplikacja scrapuje następujące strony:

**Fortuna:**

Moduł do scrapowania strony Fortuny został podzielony na trzy części, w celu umożliwienia wykonywania poszczególnych etapów osobno. W związku z tym użycie funkcji *load\_leagues* jest konieczne tylko przy tworzeniu bazy danych, aby załadować do tabeli *Fortuna\_leagues* linki do wszystkich stron poszczególnych lig. W aktualizowaniu kursów ta funkcja może zostać pominięta, a wywoływana rzadziej (np. raz w ciągu dnia), aby sprawdzić, czy nie pojawiła się oferta z nowych lig.

Funkcja *load\_matches\_odds* zapisuje do bazy, lub aktualizuje w niej kursy poszczególnych meczów (w tabeli *Fortuna\_match\_odds*), korzystając w funkcji stworzony w module *database.py*. Wykorzystywana jest w funkcji *load\_matches* która z kolei zbiera dane z każdej strony zawartej w tabeli *Fortuna\_leagues*

i następnie przekazuje je do wspomnianej funkcji *load\_matches\_odds*.

Wszystkie powyższe funkcje w swoim działaniu wykorzystują bibliotekę *beatifulsoup*.

Funkcja *scrap* jest funkcją pomocniczą, pozwalającą na wykonanie całego poziomu scrapowania z poziomu wiersza poleceń, za pomocą wywołania modułu *Fortuna.py*.

### **Forbet:**

Podział analogiczny jak przy scrapowaniu strony Fortuny. Różnicą jest to, że z racji na brak dostępu do kursów 1X, X i X2 na głównej stronie ligi, pobierane są kursy tylko na zwycięstwo gospodarzy, gości i remis. Przy ujednoliceniu funkcji zapisujących dane do bazy danych wymagało to zastosowania wartości domyślnych dla tych danych, których nie dało się pobrać.

### **LVBet:**

Do scrapowania strony LVBet należało wykorzystać selenium, ponieważ strona do wyświetlania potrzebnych nam danych korzysta ze skryptów

(więcej w punkcie 5.). Pomimo tego udało się podzielić moduł na funkcję, co pozwala na niewywoływanie każdorazowo funkcji *load\_leagues*, której wykonanie trwa stosunkowo długi (z racji właśnie na użycie selenium, które musi zasymulować działanie użytkownika). Funkcja ta wykorzystuje kontener, w którym zawarte są linki do stron poszczególnych krajów, w których odbywają się rozgrywki i z nich pobiera odnośniki do poszczególnych lig i zapisuje je do tabeli *Lvbet\_leagues*. Wykorzystuje także *driver*, który jest niezbędny do działania selenium. Do uzyskania go służy funkcja *getdriver*. *Driver* wykorzystuje również funkcja *load\_matches* która zapisuje do bazy danych zarówno mecze (tabela *Lvbet\_matches*) jak i kursy do nich (tabela *Lvbet\_match\_odds*).

Funkcja *scrap* jest funkcją pomocniczą, której wykonanie spowoduje kolejne uruchomienie wszystkich funkcji i wypełnienie wszystkich tabel powiązanych z Lvbet.

### **Milenium:**

Z racji na fakt, że nawigowanie na stronie odbywa się przez wykonywanie skryptów, a nie przechodzenie pomiędzy podstronami praktycznie całość modułu musiała zostać umieszczona w jednej funkcji *scrap*. Niestety, ostatecznie z racji na zmianę struktury strony (więcej w punkcie 5.) moduł nie działa poprawnie.

### **STS:**

W scrapowaniu strony STS można wyznaczyć 2 główne funkcje takie jak:

*getAllLaguesLinks(allLinkstoLague=[]):*

oraz

*scrapMatches(stsleuge):*

Pierwsza jest odpowiedzialna za pobranie wszystkich linków na stronie STS zawierających kursy piłkarskie. Zwraca ona tablicę linków, które potem są zapisywane do bazy danych.

Druga funkcja zbiera informacje ze strony dzięki wcześniej zescrapowanym linkom

i zapisuje je do bazy danych. Dzięki temu mamy dostęp do wyników kursów ze strony STS.

Wszystkie dane zapisywane są do bazy danych, za pomocą funkcji udostępnianych przez moduł *database.py*. Wśród najważniejszych należy wymienić *create\_league\_table*, *create\_matches\_table*, *create\_match\_odds\_table*, *insert\_league*, *insert\_match*, *insert\_odds*. Odpowiadają one za stworzenie struktury i wypełnienie bazy danych. Dla działania modułów scrapujących ważne są też funkcje *get\_leagues*, *compare\_odds*, *update\_odds*, *is\_match\_in\_db*. W czasie trwania projektu funkcje modułu zostały przemodelowane w taki sposób, aby można je było zaimplementować dla kolejnych modułów scrapujących, które mogą powstać w przyszłości.

Struktura tabel bazy danych dla pojedynczego bukmachera (pozostałe wyglądają analogicznie):



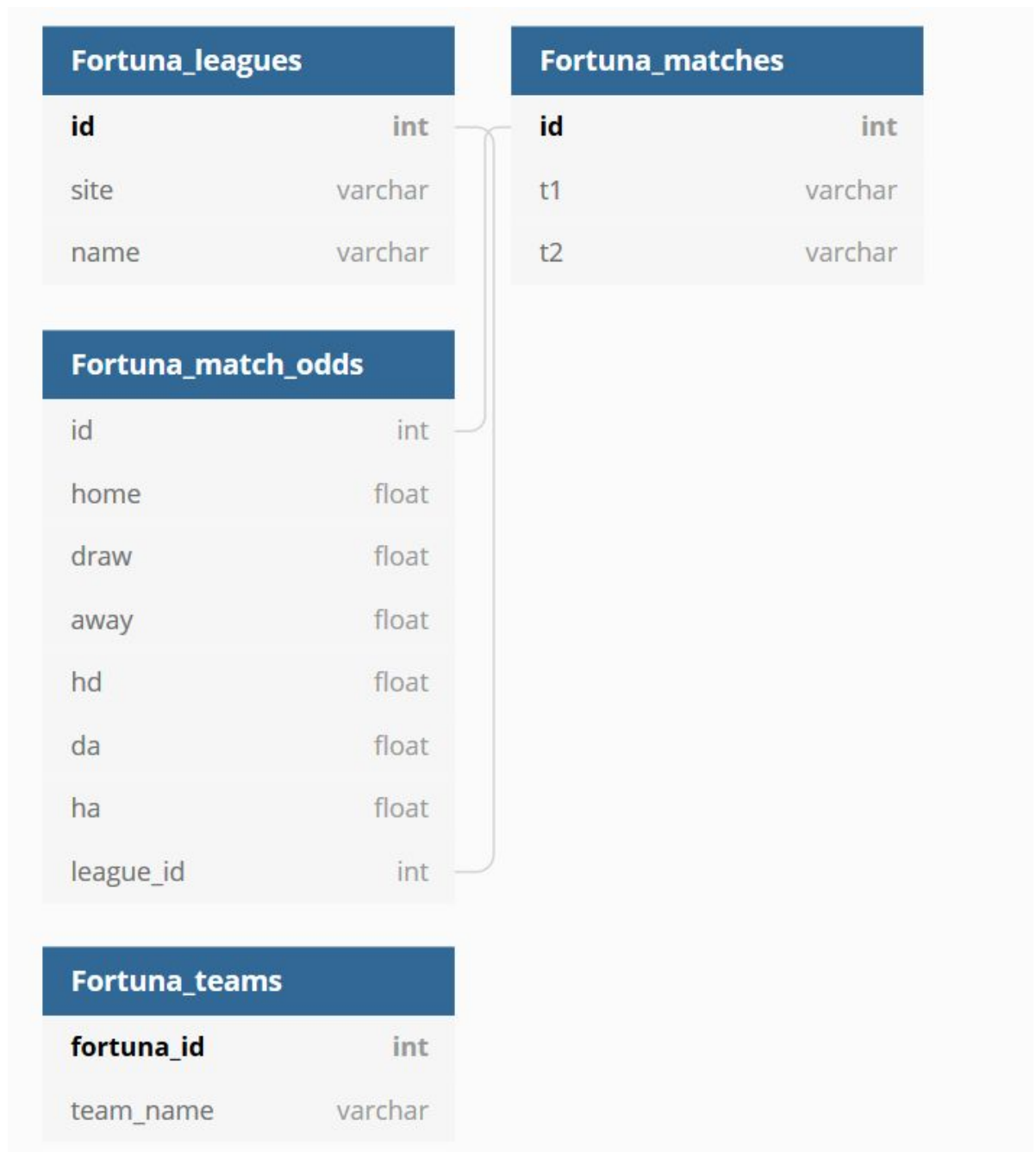


Tabela leagues:

- id - id ligi
- site - odnośnik do ligi (jeśli istnieje)
- name - nazwa ligi

Tabela matches:

- id - id meczu
- t1 - nazwa zespołu gospodarzy
- t2 - nazwa zespołu gości

Tabela match\_odds:




- id - id meczu
- home - kurs na zwycięstwo gospodarzy
- draw - kurs na remis
- away - kurs na zwycięstwo gości
- hd - kurs na zwycięstwo gospodarzy lub remis
- da - kurs na zwycięstwo gości lub remis
- ha - kurs na zwycięstwo gospodarzy lub gości
- league\_id - id ligi

Schemat tabeli tworzonej przez moduł algorytm.py:

Relationship_table	
team_name	varchar
id_fortuna	varchar
id_forbet	varchar

- team\_name - nazwa zespołu, pobierana z tabeli Forbet\_matches
- id\_fortuna - reprezentacja danego zespołu przez id w tabeli Fortuna\_teams
- id\_forbet - reprezentacja danego zespołu przez id w tabeli Fortuna\_teams

Poniżej przykładowe rekordy tabel:

Table:  Forbet\_leagues  


	id	site	name
	Filter	Filter	Filter
1	0	https://www.i...	league 0
2	1	https://www.i...	league 1
3	2	https://www.i...	league 2
4	3	https://www.i...	league 3
5	4	https://www.i...	league 4
6	5	https://www.i...	league 5
7	6	https://www.i...	league 6
8	7	https://www.i...	league 7
9	8	https://www.i...	league 8
10	9	https://www.i...	league 9
11	10	https://www.i...	league 10
12	11	https://www.i...	league 11
13	12	https://www.i...	league 12
14	13	https://www.i...	league 13
15	14	https://www.i...	league 14

Table:  Forbet\_match\_odds

	id	home	draw	away	hd	da	ha	league_id
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	283079391	3.0	3.3	2.49	0.0	0.0	0.0	0
2	283078809	3.2	3.5	2.28	0.0	0.0	0.0	0
3	283090415	2.34	3.5	3.1	0.0	0.0	0.0	0
4	283111763	2.14	3.65	3.35	0.0	0.0	0.0	0
5	283111750	2.47	3.3	3.0	0.0	0.0	0.0	0
6	283111754	2.35	3.45	3.05	0.0	0.0	0.0	0
7	283111757	1.72	3.9	5.0	0.0	0.0	0.0	0
8	283111760	2.43	3.35	3.05	0.0	0.0	0.0	0
9	282667702	2.65	3.3	2.8	0.0	0.0	0.0	0
10	282667675	1.77	3.65	5.0	0.0	0.0	0.0	0
11	282667696	2.8	3.3	2.65	0.0	0.0	0.0	0
12	282667705	1.87	3.7	4.3	0.0	0.0	0.0	0
13	282667708	3.85	3.65	1.97	0.0	0.0	0.0	0
14	282667450	1.87	3.7	4.3	0.0	0.0	0.0	0
15	282667699	2.02	3.6	3.75	0.0	0.0	0.0	0
16	282667693	1.87	3.7	4.3	0.0	0.0	0.0	0
17	285693999	2.15	2.9	3.35	0.0	0.0	0.0	1
18	285694468	1.75	3.65	3.8	0.0	0.0	0.0	1
19	285695670	1.6	3.85	4.45	0.0	0.0	0.0	1

Table: Forbet\_matches

	id	t1	t2
	Filter	Filter	Filter
1	283079391	Arka Gdynia	Jagiellonia Bia...
2	283078809	ŁKS Łódź	Lechia Gdańsk
3	283090415	Raków Często...	Korona Kielce
4	283111763	Wisła Kraków	Śląsk Wrocław
5	283111750	Piast Gliwice	Lech Poznań
6	283111754	Zagłębie Lubin	Cracovia
7	283111757	Legia Warsza...	Pogoń Szczecin
8	283111760	Wisła Płock	Górnik Zabrze
9	282667702	Górnik Zabrze	Zagłębie Lubin
10	282667675	Lech Poznań	Wisła Płock
11	282667696	Śląsk Wrocław	Piast Gliwice
12	282667705	Jagiellonia Bia...	Raków Często...
13	282667708	Korona Kielce	Legia Warsza...
14	282667450	Cracovia	ŁKS Łódź
15	282667699	Lechia Gdańsk	Wisła Kraków
16	282667693	Pogoń Szczecin	Arka Gdynia
17	285693999	Broń Radom	Huragan Morąg
18	285694468	Olimpia Zemb...	Zelazna Białe...

Table:  Forbet\_teams

	forbet_id	team_name
	Filter	Filter
1	1	'Partizani Tira...
2	2	'Teuta Durres'
3	3	'Skenderbeu ...
4	4	'Flamurtari Vl...
5	5	'Kastrioti Kruje'
6	6	'Veleciku Koplik'
7	7	'Apolonia Fier'
8	8	'Turbina Cerrik'
9	9	'Tomori Berat'
10	10	'Vllaznia Shko...
11	11	'Beselidhja Le...
12	12	'Dinamo Tirana'
13	13	'Korabi Peshk...
14	14	'Erzeni Shijak'
15	15	'Bylis Ballsh'
16	16	'Besa Kavaje'
17	17	'USM Alger'
18	18	'JS Kabylie'
19	19	'Paradou AC'
20	20	'JS Saoura'
21	21	'ES Setif'

Table: Relationship\_table

	team_name	id_fortuna	id_forbet
	Filter	Filter	Filter
1	'USM Alger'	71	0
2	'JS Kabylie'	72	1
3	'Paradou AC'	73	2
4	'JS Saoura'	74	3
5	'ES Setif'	75	4
6	'MC Alger'	76	5
7	'CS Constantine'	77	6
8	'CR Belouizdad'	78	7
9	'MC Oran'	79	8
10	'MO Bejaia'	80	9
11	'Olympique M...	81	10
12	'DRB Tadjenant'	82	11
13	'US Biskra'	83	12
14	'NC Magra'	84	13
15	'WA Tlemcen'	85	14

Implementacja serwera i stron internetowych:

Endpointy serwera oraz Query:

- `@app.route('/')`-link strony głównej
- `@app.route('/liga-mistrzow')`-link zwracający jsona dla ligi mistrzów wykorzystany w szybkim dostępie na podobnej zasadzie działają:  
`@app.route('/copaamerica')`,`@app.route('/mistrzostwaeuropy')`
- `@app.route('/szukaj')`-link do strony wyszukiwania po nazwach drużyn
- `@app.route('/szukaj/<name>')`-link zwracający jsona z meczami dla konkretnej drużyny. Przykładowe zapytanie dla strony Fortuna wysyłane do bazy danych w celu znalezienia wyniku:  
`cur.execute("select id from Fortuna_leagues where name=(?)",[name])`- w pierwszym kroku znajdujemy id ligi



```
cur.execute("select t1,t2,home,draw,away,hd,da,ha FROM (select *
FROM Fortuna_match_odds WHERE
Fortuna_match_odds.league_id=(?) )AS F Inner join
Fortuna_matches On F.id = Fortuna_matches.id ",[idligi[0][0]])
```

## 5. Interesujące problemy na jakie się natknęliśmy podczas pracy nad projektem:

Strona Milenium 03.06.2019

Na dwóch powyższych screenach widać typowy problem dla webscrappingu. Po zmianie kodu strony przez jej właściciela do tej pory działający moduł scrapujący dane staje się praktycznie bezużyteczny. Oprócz nazw i właściwości elementów strony zmieniło się też jej działanie, co widać na screenach - przed zmianą wszystkie ligi były dostępne po rozwinięciu zakładki "Piłka nożna", po zmianie po rozwinięciu tej zakładki pojawia się dostęp do poszczególnych krajów i dopiero po rozwinięciu któregoś z nich mamy dostęp do danych rozgrywek. Wymaga to sporych zmian w kodzie scrapującym taką stronę.

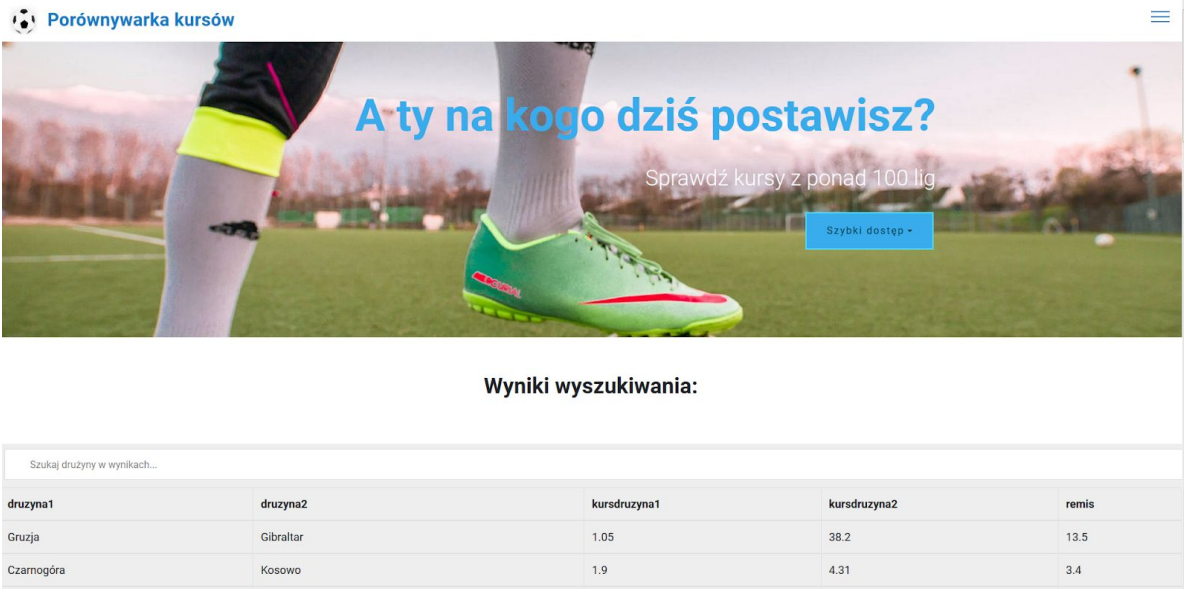
The screenshot shows the forBET website interface. At the top, there's a navigation bar with 'ZAKŁADY BUKMACHERSKIE' and 'ZAKŁADY NA ŻYWO'. Below this, a menu lists various sports: PIŁKA NOŻNA, KOSZYKÓWKA, HOKEJ NA LODZIE, SIATKÓWKA, PIŁKA RĘCZNA, KRYKIET, RUGBY, BADMINTON, and WIĘCEJ. The main content area is titled 'Piłka nożna' and includes a 'MECZE' section for '14.06.2019'. It lists matches from different leagues like 'Świat - Mistrzostwa Świata (K)', 'Chiny - Super League', and 'Brazylia - Série A'. A 'KALENDARZ' (calendar) for June 2019 is also visible on the right side of the match list.

Jednym z kolejnych problemów które napotkaliśmy było użycie Java Scriptu przez podmiot tworzący stronę. Do wyciągnięcia potrzebnych nam informacji było konieczne użycie selenium, które korzystając z webdrivera imituje działanie użytkownika. Jest to związane niestety z większym nakładem czasowym, pamięciowym oraz obliczeniowym.

## 6. Instrukcja użytkowania aplikacji:



1. Użytkownik pod adresem lokalnym 127.0.0.0:5000 ma dostęp do strony głównej gdzie jego oczom ukazuje kilka ostatnich wyników z bazy danych.



Porównywarka kursów

A ty na kogo dziś postawisz?

Sprawdź kursy z ponad 100 lig

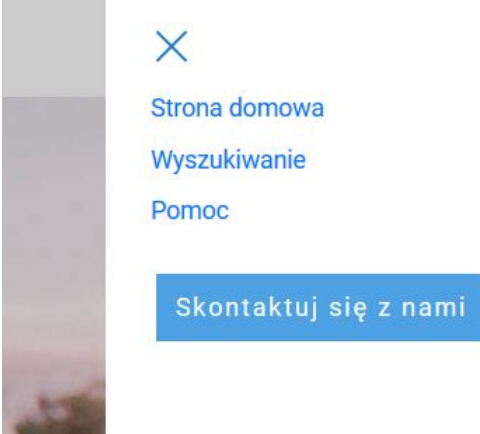
Szybki dostęp

Wyniki wyszukiwania:

Szukaj drużyny w wynikach...				
druzyna1	druzyna2	kursdruzyna1	kursdruzyna2	remis
Gruzja	Gibraltar	1.05	38.2	13.5
Czarnogóra	Kosowo	1.9	4.31	3.4

Widok strony głównej

2. Z rozwijanego paska możemy przejść do wyszukiwarki meczy znajdujące się pod adresem 127.0.0.0:5000/szukaj.



×

Strona domowa

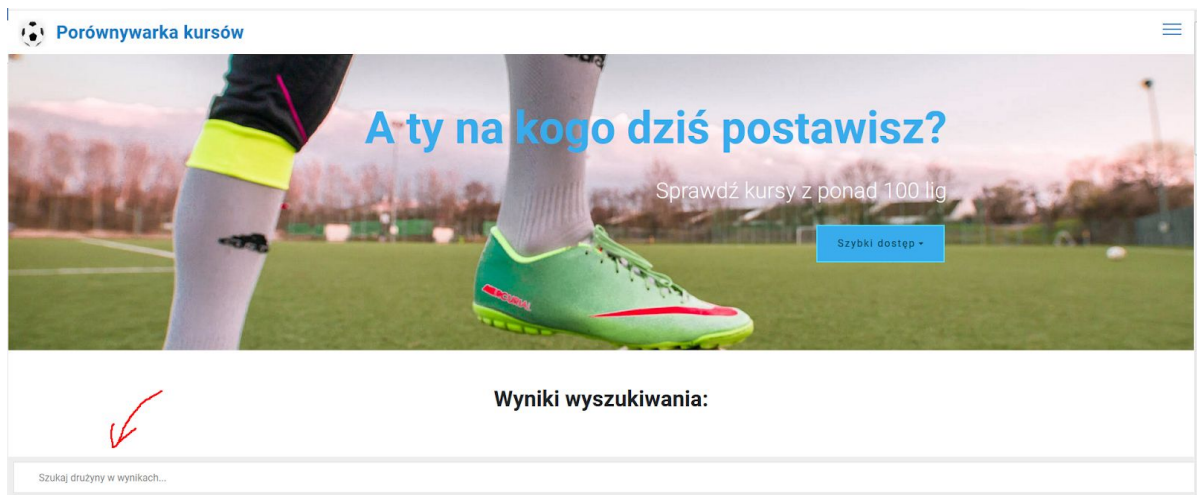
Wyszukiwanie

Pomoc

Skontaktuj się z nami

Widok z paska dostępu

3. Na stronie wyszukiwania wyświetlają nam się w tabeli nazwy drużyn. Przy pomocy okienka wyszukiwania możemy odfiltrować interesujące nas wyniki.



### Okienko wyszukiwanie

4. Po wybraniu konkretnej drużyny ładuje nam się tabela z aktualnymi kursami dla danej drużyny. Mamy podane dane odnośnie: nazw przeciwników, kursu na wygraną, kursu na remis, kursu na konkretny wynik i źródło z jakiej strony pochodzą kursy.

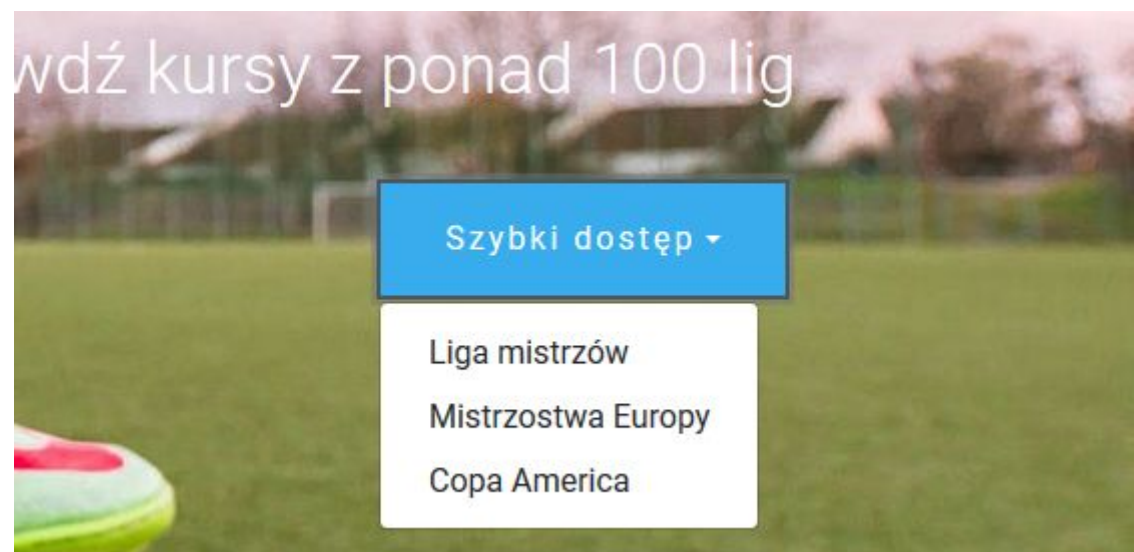
Wyniki wyszukiwania:

Szukaj drużyny w wynikach...

Drużyna 1	Drużyna 2	Kurs na drużynę 1	Remis	Kurs na drużynę 2	jx	jd	id	źródło
Ekwador	Włochy	3.55	3.2	2.15	1.68	1.29	1.34	
Senegal	Kolumbia	3.9	2.8	2.25	1.63	1.25	1.43	
Polska	Tahiti	1.03	18	33	0	11.5	0	
Honduras	Urugwaj	14.5	6.5	1.19	4.5	0	1.09	
Katar	Ukraina	7	4.6	1.41	2.78	1.08	1.17	
Norwegia	N.Zelandia	2.03	3.4	3.55	1.27	1.74	1.29	
USA	Nigeria	2.86	3.4	2.35	1.55	1.39	1.29	
Panama	Francja	11	5.8	1.22	3.8	1.01	1.1	
Portugalia	Argentyna	2.65	3.3	2.5	1.47	1.42	1.29	
RPA	Korea Płd.	2.77	3.2	2.45	1.48	1.39	1.3	

### Widok tabeli

5. Ponad to użytkownik ma dostęp do wyników meczy poprzez opcję szybkiego dostępu gdzie wybierając konkretną ligę uzyska wyniki wszystkich zaplanowanych meczy i kursów.



*Pasek szybkiego dostępu*