



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
WYDZIAŁ INŻYNIERII METALI I INFORMATYKI PRZEMYSŁOWEJ

KATEDRA INFORMATYKI STOSOWANEJ I MODELOWANIA

Projekt dyplomowy

*Statystyczna analiza ofert sprzedaży nieruchomości z
wykorzystaniem techniki web scraping*

Autor:	<i>Jakub Sosin</i>
Kierunek studiów:	<i>Informatyka stosowana</i>
Opiekun projektu:	<i>dr inż. Krzysztof Bzowski</i>

Kraków, rok 2022

Spis treści

1. Wstęp.....	3
2. Web scraping.....	3
2.1. Metody ekstrakcji danych.....	4
2.2. Legalne formy wykorzystania źródła danych.....	5
3. Przegląd istniejących rozwiązań.....	6
4. Cel pracy.....	10
5. Opis zastosowanych technologii.....	10
6. Projekt i architektura rozwiązania.....	12
6.1. Założenia i wymagania rozwiązania.....	12
6.2. Architektura rozwiązania.....	13
6.3. Serwer.....	14
6.4. Aplikacja Klientka.....	17
7. Implementacja.....	18
7.1. Funkcja pobierająca dane.....	18
7.2. Interfejs użytkownika.....	23
7.3. Powiadomienie o okazji oferty.....	27
7.4. Statystyczna analiza ofert.....	30
7.5. Test interfejsu.....	39
8. Podsumowanie.....	42
9. Bibliografia.....	42

1. Wstęp

W ostatnich latach zauważalny jest szybki trend wzrostowy wartości polskiego rynku nieruchomości. Sektor ten według danych NBP w 2020 roku wart był ok. 4,9 bln złotych, wobec 4,3 bln złotych w roku poprzednim [1]. Rozwój został chwilowo spowolniony wraz z pojawieniem się epidemii koronawirusa, jednak sektor pozostał jednym z ważniejszych gałęzi gospodarki, a inwestowanie w nieruchomości bezpieczną formą alokacji kapitału. Rosnące zainteresowanie nabyciem własnego mieszkania sprzyja rozwojowi serwisów ogłoszeniowych, które pozwalają kupującemu na znalezienie oferty o interesujących dla niego parametrach. Portale te, wykorzystywane zarówno przez zwykłego użytkownika jak i profesjonalnego agenta nieruchomości ograniczają się głównie do wyświetlania ofert. Dla osoby poszukującej nieruchomości, takie rozwiązanie może okazać się niewystarczające. Brakuje bowiem serwisów, które dokonywałyby analizy parametrów ofert i wyświetlały użytkownikowi jej wyniki w formie spersonalizowanych statystyk. Takie rozwiązanie mogłoby przyczynić się do podjęcia szybszej i bardziej opłacalnej decyzji, gdyż osoba poszukująca nieruchomości dostanie narzędzie, dzięki któremu zdobędzie informacje o czynnikach kształtujących cenę, oraz lepiej pozna sytuację panującą na rynku. Próba odpowiedzi na to zapotrzebowanie jest aplikacja wykonana w ramach niniejszego projektu inżynierskiego.

2. Web scraping

Termin web scraping oznacza ogół mechanizmów ekstrakcji danych ze stron internetowych. Podobnie jak przeglądarka, rozwiązanie oparte o web scraping wysyła żądania do wybranego serwera www w celu pobrania danych w formacie HTML (HyperText Markup Language) – dane nie są jednak renderowane graficznie tak jak w tradycyjnej przeglądarce. Zadaniem programu nie jest prezentacja treści, lecz jej przeszukiwanie celem znalezienia pożądaných elementów. Działanie jest skoncentrowane na pobraniu konkretnych danych, które przeważnie zostały zapisane w kodzie HTML i mogą nie być ustrukturyzowane. Program powinien więc przekształcić zebrane informacje do postaci, która będzie możliwa do zapisu w bazie danych, a także ułatwi późniejszą ich analizę. Technika ekstrakcji danych ze stron WWW ma szerokie spektrum zastosowania, może posłużyć między innymi do monitorowania cen produktów, wykrywania czy zmienia się treść wybranej strony, a także w badaniach naukowych, gdzie portale internetowe mogą również posłużyć jako źródło informacji.

2.1. Metody ekstrakcji danych

Do pobrania danych znajdujących się na stronie internetowej, można użyć między innymi metod, których działanie polega na wskazaniu ścieżki do elementu, gdzie została umieszczona poszukiwana zawartość. Do tego typu metod, można zaliczyć selektory CSS, które lokalizują elementy na podstawie ich klasy, identyfikatora oraz tagu. Zapis takiego selektora, może wyglądać następująco: `#offer-24009712 > article > ul > li.email`. Kolejnym sposobem wyszukiwania elementów za pomocą ścieżki są selektory XPath, które mają przeważnie dłuższą i bardziej skomplikowaną budowę niż selektory CSS. Istnieją jednak przypadki, gdzie wykorzystanie XPath może być konieczne. Selektory tego typu wyszukują również element nadrzędny w strukturze strony, podczas gdy selektory CSS oferują jedynie „poruszanie się” po elementach potomnych. Kolejną zaletą XPath, jest także możliwość znalezienia elementu na podstawie fragmentu jego zawartości bądź nazwy atrybutu. Aby wyszukać element po części jego nazwy, należy użyć funkcji *starts-with*, z kolei do znajdowania po fragmencie tekstu wykorzystuje się funkcję *contains*. Przykładowe użycie funkcji:

- `//article[contains(email, "biuro@")]` – selektor znajdzie każdy element typu article, który posiada adres email zawierający fragment „biuro@”,
- `//*[starts-with(name(), 'b')]` – selektor znajdzie elementy o nazwie, która zaczyna się od podanej litery.

Konwencja zapisu selektora XPath jest następująca: `//*[id="offer-24009712"]/article/ul/li[2]`. Kolejną możliwą metodą ekstrakcji są wyrażenia regularne (ang. *Regular expresion*), których podstawowym założeniem jest definicja wzorca, według którego na stronie bądź też dokumencie tekstowym, wyszukiwane są ciągi znaków spełniające określone reguły. Metoda ta, może być zastosowana przy znajdowaniu w tekście elementów, które mają charakterystyczny sposób zapisu, między innymi są to adresy mailowe, numery telefonów bądź też adresy URL stron internetowych. Do odnalezienia ciągu znaków, który jest potencjalnie adresem mailowym, można wykorzystać następujący wzorzec:

`/^\\w+ ([\\.-]?\\w+)*@\\w+ ([\\.-]?\\w+)* (\\.\\w{2,3})+$/` - wyrażenie sprawdzi, czy ciąg znaków spełnia cechy adresu mailowego, takie jak:

- znak @ po pierwszym fragmencie ciągu,

- pierwsza część ciągu może zawierać wyłącznie litery, cyfry, znak podkreślenia oraz kropkę,
- nazwa domeny serwerowej poczty, która musi składać się z co najmniej dwóch znaków,
- ciąg rozpoczęty i zakończony znakiem litery bądź cyfry.

2.2. Legalne formy wykorzystania źródła danych

Pojęcie własności danych zostało określone w artykule 6 ustawy z dnia 27 lipca 2001 roku, wedle którego *„Producentowi bazy danych przysługuje wyłączne i zbywalne prawo pobierania danych i wtórnego ich wykorzystania w całości lub w istotnej części, co do jakości lub ilości.”* [2].

Ustawa przewiduje jednak pewne wyjątki, odstępstwa od prawa *sui generis*, przysługującego twórcy. Na mocy art 8. wolno przechowywać informację z bazy: *”w charakterze ilustracji, w celach dydaktycznych lub badawczych, ze wskazaniem źródła, jeżeli takie korzystanie jest uzasadnione niekomercyjnym celem, dla którego wykorzystano bazę”* [2].

Należy podkreślić, że zapis ten nie gwarantuje jednak użytkownikowi prawa do legalnego wykorzystania źródła danych, jeśli pochodzące z niego informacje, zostały uzyskane przy użyciu techniki web scraping. Ustawodawca wyklucza bowiem możliwość użycia automatycznych form zbierania danych w art. 8 punkcie 2: *„Nie jest dozwolone powtarzające się i systematyczne pobieranie lub wtórne wykorzystanie sprzeczne z normalnym korzystaniem i powodujące nieusprawiedliwione naruszenie słuszych interesów producenta.”* [2]. Dlatego przy rozważaniu, nad wyborem prawnie dozwolonej formy pozyskiwania danych, rozstrzygające mogą okazać się zapisy regulaminu umieszczonego w serwisie, w którym określone są warunki korzystania ze strony, oraz prawa użytkownika.

Przy gromadzeniu danych, poza kwestiami prawnymi należy zwrócić również uwagę, czy twórcy zamieścili w serwisie plik robots.txt, gdzie mogą być zawarte informacje, jakie czynności roboty mogą wykonywać na stronie, a czego nie powinny robić. Na przykład zapis: *User-agent: * allow: /* pozwala robotom na przejście do dowolnej podstrony portalu [3].

Dobłą praktyką, którą warto uwzględnić przy tworzeniu programu jest ustawienie pewnego odstępu czasowego, między którym odbywa się pobieranie danych. Celem takiego działania jest zmniejszenie obciążenia serwera strony, tym samym nie doprowadzenie do uzyskania blokady dostępu, w momencie w którym zostałoby wykryte nadmierne wykorzystywanie jego zasobów [4].

Wszystkie powyższe kwestie wzięto pod uwagę, przy wyborze serwisu ogłoszeniowego, który będzie stanowił najlepsze możliwe źródło informacji. Wybór padł na portal sprzedajemy.pl, gdzie autor

uzyskał zgodę na wykorzystanie danych umieszczonych w ofertach. Dużą zaletą serwisu jest jego integracja z popularnymi aplikacjami do eksportu ogłoszeń, takimi jak asari czy galactica virgo, które zostały stworzone na potrzeby agencji nieruchomości. Portal zawiera ponad 50000 ogłoszeń dotyczących nieruchomości, a znaczna ich część – dziewięćdziesiąt procent, została umieszczona właśnie przez pośredników.

3. Przegląd istniejących rozwiązań

Na rynku dostępnych jest kilka rozwiązań umożliwiających monitorowanie serwisów ogłoszeniowych. Wśród nich można znaleźć między innymi dodatki do przeglądarki, w większości darmowe, które jednak nie oferują zbyt wielu możliwości. Istnieją również serwisy internetowe, posiadające zdecydowanie większą funkcjonalność, ale niestety dostępną za dodatkową opłatą.

Jednym z rozwiązań jest wtyczka Web Scraper - Free Web Scraping [5], dostępna między innymi dla przeglądarki Chrome. Użytkownik wskazuje informacje którą chce pobrać, poprzez kliknięcie przyciskiem myszy w element strony w którym się ona znajduje. Wtyczka ma jednak pewne ograniczenia, w celu zgromadzenia większej ilości danych tego samego typu, wymagana jest już od użytkownika przynajmniej podstawowa wiedzy o sposobie działania selektorów klas, oraz identyfikatorów strony. Kolejnym ograniczeniem występującym w wersji bezpłatnej jest brak możliwości cyklicznego i automatycznego pobierania treści, gdyż uruchomienie programu w tym przypadku jest możliwe jedynie w sposób manualny, poprzez działanie użytkownika. Wymaganą konfigurację do odnalezienia informacji na stronie przedstawiono na rysunku 1, dane zgromadzone w jej wyniku przedstawiono na rysunku 2.

ID	Selector	type	Multiple	Parent selectors	Actions
price	span price	SelectorText	no	element	Element preview Data preview Edit Delete
area	span first	SelectorText	no	element	Element preview Data preview Edit Delete
date	time	SelectorText	no	element	Element preview Data preview Edit Delete
title	title a	SelectorText	no	element	Element preview Data preview Edit Delete
localisation	strong	SelectorText	no	element	Element preview Data preview Edit Delete
image_url	.listingWrp img	SelectorImage	no	element	Element preview Data preview Edit Delete

Add new selector

Rysunek 1. Wybrane elementy ogłoszenia do pobrania przez wtyczkę

Data Preview

635 000 zł	Pow.: 212 m²	Dzisiaj 18.18	Dom na sprzedaż Dąbrowa Tarnowska 212m2	Dąbrowa Tarnowska	https://thumbs.img-sprzedajemy.pl/350x250/cb/39/06/dom-na-sprzedaz-dabrowa-tarnowska-212m2-malopolskie-548629477.jpg
280 000 zł	Pow.: 90 m²	Dzisiaj 18.17	Dom w Białej Podlaskiej przy ulicy Dalekiej	Biała Podlaska	https://thumbs.img-sprzedajemy.pl/350x250/cb/e9/1e/dom-w-bialej-podlaskiej-ulicy-dalekiej-biala-podlaska-544785227.jpg
1 290 000 zł	Pow.: 234 m²	Dzisiaj 18.08	Dom wolnostojący Szczyclice 233.84m2	Szczyglice	https://thumbs.img-sprzedajemy.pl/350x250/c8/76/86/dom-wolnostojacy-szczyglice-23384m2-do-wykonczenia-549127235.jpg
604 172 zł	Pow.: 92 m²	Dzisiaj 18.08	Dom bliźniak Mogilany 91.67m2	Mogilany	https://thumbs.img-sprzedajemy.pl/350x250/c6/2f/75/50/dom-blizniak-mogilany-9167m2-sprzedaz-sprzedaz-549127189.jpg
800 000 zł	Pow.: 160 m²	Dzisiaj 18.07	Ogłoszenie dom wolnostojący 160m2 Chrosna	Chrosna	https://thumbs.img-sprzedajemy.pl/350x250/cb/15/e1e2/ogloszenie-dom-wolnostojacy-160m2-chrosna-549126998.jpg
410 000 zł	Pow.: 150 m²	Dzisiaj 18.05	Dom wolnostojący 150 metrów Kraków	Kraków	https://thumbs.img-sprzedajemy.pl/350x250/c8/7c/b8/c8/dom-wolnostojacy-150-metrow-krakow-549126802.jpg
699 000 zł	Pow.: 219 m²	Dzisiaj 18.00	Dom piętrowy do zamieszkania Pietrzykowice Żywieckie	Pietrzykowice	https://thumbs.img-sprzedajemy.pl/350x250/c7/d1/de/99/dom-pietrowy-do-zamieszkania-pietrzykowice-ceglana-547652799.jpg
140 000 zł	Pow.: 80 m²	Dzisiaj 17.58	sprzedam budynek do zaadaptowania na mieszkanie	Ścinawa Polska	https://thumbs.img-sprzedajemy.pl/350x250/c2/c1/e2/2e/sprzedam-budynek-do-zaadaptowania-na-mieszkanie-scinawa-polska-521019517.jpg
569 000 zł	Pow.: 125 m²	Dzisiaj 17.54	Segment dom szeregowy SKRAJNY - Wzgórze Gołonoskie	Dąbrowa Górnicza	https://thumbs.img-sprzedajemy.pl/350x250/c2/80/4c/segment-dom-szeregowy-skrainy-wzgorze-golonoskie-4552zm2-dabrowa-gornicza-sprzedam-548440361.jpg
249 000 zł	Pow.: 300 m²	Dzisiaj 17.53	Sprzedaz domu Księginice Wielkie 300m2	Księginice Wielkie	https://thumbs.img-sprzedajemy.pl/350x250/c3/17/f0/sprzedaz-domu-ksieginice-wielkie-300m2-543254514.jpg
650 000 zł	Pow.: 126 m²	Dzisiaj 17.49	Dom bliźniak 126m2 Myslowice	Mysłowice	https://thumbs.img-sprzedajemy.pl/350x250/c0/1c/81/14/dom-blizniak-126m2-myslowice-548802435.jpg
440 000 zł	Pow.: 250 m²	Dzisiaj 17.49	Dom wolnostojący na dużej działce Proszowice II	Proszowice	https://thumbs.img-sprzedajemy.pl/350x250/c7/34/4e/43/dom-wolnostojacy-zebocin-250m2-rok-budowy-2001-542372991.jpg

Rysunek 2. Dane które zgromadzono przy użyciu wtyczki

Kolejnym z rozwiązań jest portal lzx.pl [6]. Serwis po wprowadzeniu wymaganych parametrów, przedstawionych na rysunku 3, oferuje monitorowanie stron, dla których twórcy dostosowali działanie programu. Dzięki temu, nie ma konieczności podawania, czy zaznaczania przez użytkownika selektorów, w których znajdują się poszukiwane informacje. Dane są zbierane w sposób cykliczny, a o wykryciu nowych ogłoszeń użytkownik jest powiadamiany poprzez wiadomość mailową. Portal posiada jednak pewne ograniczenia, treści oferty pobieranie są jedynie z pierwszej strony wyników wyszukiwań, przez co stworzenie dużej bazy danych staje się czasochłonne, bądź też niemożliwe, w przypadku chęci znalezienia ogłoszeń o rzadziej spotykanych parametrach. Kolejną wadą serwisu jest brak opcji filtrowania i sortowania znalezionych ofert, jak również eksportu danych, więc możliwości portalu sprowadzają się głównie do wyświetlania znalezionych ogłoszeń. Dane które udało się zgromadzić przy użyciu portalu przedstawiono na rysunku 4.

EDYCJA WYSZUKIWANIA

LINK DO WYNIKÓW WYSZUKIWANIA

<https://sprzedajemy.pl/nieruchomosci/domy>



skanuj







NAZWA WYSZUKIWANIA

nowe wyszukanie

CZĘSTOŚĆ SPRAWDZANIA

skanowanie co 5 minut ▼

Rysunek 3. Utworzenie nowego scrapera w portalu lzx.pl

Tytuł oferty	Filtr	Serwis	Cena	Data znalezienia	
 Dom wolnostojący 380m2 Kraków	Domy w Krakowie	Sprzedajemy.pl	1 500 000 zł	paź 28, 12:42	☆
 Dom bliźniak, Wzgórza Krzesławickie, okazja!	Domy w Krakowie	Sprzedajemy.pl	899 000 zł	paź 28, 12:42	☆
 Dom wolnostojący 192.1m2 Kraków Łuczanowicka	Domy w Krakowie	Sprzedajemy.pl	2 450 000 zł	paź 28, 12:42	☆
 Dom bliźniak 142.76m2 Kraków	Domy w Krakowie	Sprzedajemy.pl	670 000 zł	paź 28, 12:42	☆
 Dom sprzedam 745m2 Kraków Geologów	Domy w Krakowie	Sprzedajemy.pl	1 999 999 zł	paź 28, 12:42	☆
 Sprzedam dom Kraków 136m2	Domy w Krakowie	Sprzedajemy.pl	659 000 zł	paź 28, 12:42	☆
 Dom wolnostojący 745m2 Kraków Geologów	Domy w Krakowie	Sprzedajemy.pl	1 999 999 zł	paź 28, 12:42	☆

Rysunek 4. Widok danych, które zebrano przy użyciu strony lzx.pl

Innym przykładem jest program parsehub.com [7]. Do jego wykorzystania nie jest wymagana szczegółowa wiedza z zakresu budowy stron internetowych. Użytkownik przy pomocy kliknięcia przyciskiem myszy wybiera element, który chce pobrać. Program sam zidentyfikuje pozostałe podobne elementy, więc nie ma potrzeby wskazywania osobno każdego z nich (rysunek 5). Istnieje również możliwość pobrania danych z każdej strony wyników wyszukiwania, wystarczy że zostanie podany element posiadający adres URL następnej strony. Kolejną zaletą programu jest możliwość eksportu

danych w formacie CSV jak również JSON. Jako wadę należy z kolei wskazać konieczność pobrania aplikacji desktopowej. Program w wersji bezpłatnej oferuje możliwość utworzenia 5 projektów – funkcji pobierających danych. Dane pobrane przy użyciu programu przedstawiono na rysunku 6.



Dzisiaj 20:37

Mieszkanie Gorzów Wielkopolski 33m2 3 pokoje

189 000 zł

Pow.: 33 m² | Pokoje: 3 | Rok bud.: 1945

Gorzów Wielkopolski



Dzisiaj 20:37

Mieszkanie Katowice Tyska 45m2 2-pok

1 800 zł

Pow.: 45 m² | Zabudowa: blok | Pokoje: 2

Katowice
Piotrowice



Dzisiaj 20:36

mieszkanie jednopokojowe 36 m2 Nowa Huta

379 000 zł

Pow.: 36 m² | Zabudowa: blok | Pokoje: 1

Kraków
Nowa Huta

Rysunek 5. Aplikacja pozwala na wskazanie na stronie parametrów tego samego typu

offer_name	offer_url	offer_price	offer_location	offer_location_url	offer_area	offer_image
OFERTA NIEAKTUALNA: Loftowy Apartament w sercu Saskiej Kępy	https://sprzedajemy.pl/oferta-nieaktualna-loftowy-apartament-w-sercu-saskiej-kepy-warszawa-2-1b8e55-nr65103245	625 000 zł	Warszawa	https://sprzedajemy.pl/warszawa/nieruchomosci/mieszkania/sprzedaz	Pow.: 37 m ²	https://spruserfiles2.bl.../userpictures/t1u8102604ddb40f...
Grunť 4080m2 Gdańsk Grzybowa	https://sprzedajemy.pl/grunt-4080m2-gdansk-grzybowa-2-1b8e55-nr65256414	1 346 400 zł	Gdańsk	https://sprzedajemy.pl/gdansk/nieruchomosci/grunty-i-dzialki/sprzedaz	Pow.: 4 080 m ²	https://spruserfiles2.bl.../userpictures/t1u9365360d583a18...
Budownictwo wielorodzinne, media w drodze, MPZP	https://sprzedajemy.pl/budownictwo-wielorodzinne-media-w-drozdze-mpzp-mysiadlo-2-1b8e55-nr65135586	1 200 000 zł	Mysiadło	https://sprzedajemy.pl/mysiadlo/nieruchomosci/grunty-i-dzialki/sprzedaz	Pow.: 900 m ²	https://thumbs.img-sprzedajemy.pl/350x250c/92/52/92/budownictwo-wielorodzinne-media-w-drozdze-mpzp-mazowieckie-mysia...
Mieszkanie Warszawa 76.9m2 3 pokojowe	https://sprzedajemy.pl/mieszkanie-warszawa-76-9m2-3-pokojowe-2-1b8e55-nr64842526	1 220 000 zł	Warszawa	https://sprzedajemy.pl/warszawa/mokotow/nieruchomosci/mieszkania/sprzedaz	Pow.: 77 m ²	https://thumbs.img-sprzedajemy.pl/350x250c/c6/02/79/mieszkanie-warszawa-769m2-3-pokojowe-liczba-pokoi-3-549681805.j...

Rysunek 6. Widok danych zebranych przy użyciu programu parsehub.com

4. Cel pracy

Celem pracy było stworzenie aplikacji mającej posłużyć do statystycznej analizy ofert sprzedaży na rynku nieruchomości. Analiza oraz wizualna prezentacja informacji ma się odbyć przy wykorzystaniu wykresów porównawczych. Dane niezbędne do realizacji celu, pobierane są z dedykowanego portalu przy zastosowaniu techniki ekstrakcji danych, a następnie gromadzone w bazie danych. Podjęcie próby stworzenia takiej aplikacji motywowane było brakiem rozwiązania umożliwiającego monitorowanie serwisów ogłoszeniowych, które oferuje równocześnie wszystkie wyżej wymienione funkcjonalności, a przy tym nie wymaga od użytkownika przeprowadzenia skomplikowanej konfiguracji. Na etapie analizy istniejących rozwiązań zauważono, że większość narzędzi posiada następujące ograniczenia:

- Brak automatyzacji procesu pobierania danych. Dane kolekcjonowane są jednorazowo, kolejne ich pobranie wymaga działania użytkownika;
- Monitorowanie ofert jedynie najnowszych, znajdujących się na pierwszej stronie wyników wyszukiwania;
- W przypadku darmowego konta, aplikacje oferują ograniczoną funkcjonalność bądź limity czasowe.

Zebranie wystarczającej ilości informacji, niezbędnych do przeprowadzenia wiarygodnej analizy ofert sprzedaży nieruchomości z wykorzystaniem istniejących rozwiązań jest procesem niemożliwym lub długotrwałym.

5. Opis zastosowanych technologii

Do implantacji projektu wykorzystano następujące technologie: język JavaScript z bibliotekami Node.js, Chart.js i React, HTML, CSS, Bootstrap, Playwright oraz web push.

Javascript - jeden z najpopularniejszych skryptowych języków programowania, wykorzystywany głównie przy tworzeniu stron internetowych, powodujący że stają się one interaktywne. Język ten posiada kilka rozwiązań wykorzystywanych przy budowie aplikacji internetowych, najpopularniejsze z nich to React.js, Angular oraz Vue.js. Celem ich stosowania jest przyspieszenie oraz ułatwienie pracy programisty, między innymi poprzez tworzenie elementów, możliwych do ponownego użycia w innych częściach aplikacji. Javascript dzięki obecności na rynku biblioteki Node.js, może być również wykorzystywany do budowania aplikacji serwerowych [8].

Bootstrap - framework CSS wykorzystywany przy tworzeniu wizualnej części strony. Zawiera gotowe już komponenty HTML i CSS, które można użyć do tworzenia elementów strony takich jak etykiet, przycisków, tabel czy też list [9]. Komponenty można dostosowywać na wiele sposobów, Bootstrap pozwala na zmianę ich kształtu, koloru jak również położenia względem innych elementów. Biblioteka, przyspiesza i ułatwia więc proces tworzenia interfejsu aplikacji internetowych.

React - biblioteka rozwijana przez firmę Facebook, której przeznaczeniem jest między innymi tworzenie interaktywnych stron internetowych. Jedną z zalet biblioteki to możliwość rozdzielenia całego interfejsu użytkownika na mniejsze unikalne komponenty, które mogą być ponownie wykorzystane w innych częściach strony. React umożliwia przekazywanie między nimi danych, które mogą zostać następnie odczytane za pomocą zmiennej *props*. Biblioteka monitoruje zmiany zachodzące na stronie, przy użyciu stanu, który jest zbiorem zmiennych śledzących na bieżąco i przy zmianie których, następuje ponowne renderowanie komponentów [10].

Node.js - środowisko uruchomieniowe, używane do tworzenia aplikacji po stronie serwera. Node.JS, jako pierwszy, dał możliwość uruchomienia języka JavaScript niezależnie od przeglądarki. Do przetwarzania kodu, środowisko posługuje się silnikiem V8, z którego korzysta również m. in. przeglądarka Google Chrome. Silnik kompiluje kod JavaScript do kodu maszynowego, dodatkowo poddając go optymalizacji, co powoduje bardziej wydajne działanie. Node.js obecnie rozwija się bardzo dynamicznie, jest używany przez wiele rozpoznawalnych marek. Wśród nich są między innymi: Netflix, Uber, LinkedIn czy PayPal. Wysokie zainteresowanie tym środowiskiem, może wynikać z faktu iż korzystający z niego programista, zyskuje dostęp do wielu narzędzi deweloperskich, m.in menadżera pakietów NPM [11].

Playwright - biblioteka służąca między innymi do automatycznego testowania działania stron internetowych. Nieskomplikowana obsługa, oraz duże możliwości automatyzacyjne sprawiają, że jest to narzędzie doskonale nadające się do wykorzystania również przy eksploracji danych. Dodatkowo biblioteka pozwala wykonać działania przy użyciu przeglądarki bez interfejsu użytkownika (ang. Headless browser) [12]. Obecnie, wiele stron internetowych jest budowanych jako Single Page Application, gdzie treść jest renderowana dynamicznie przy wykorzystaniu języka JavaScript. Ekstrakcja danych z tego typu stron wymaga użycia przeglądarki z interpretatorem, która wykona kod JavaScript i załaduje pełną zawartość strony. Narzędzia typu axios nie sprawdzają się w tym przypadku, gdyż nie pozwalają uruchomić JavaScript, więc nie analizują kodu strony w taki sam sposób jak robi to przeglądarka. [13].

Chart.js - darmowa biblioteka JavaScript, która używana jest do wizualizacji danych. Zawiera gotowe szablony, dzięki czemu korzystanie z niej nie jest skomplikowane, głównym warunkiem wykorzystania jej elementów, jest wcześniejsze przygotowanie danych w formie wymaganej przez dany typ wykresu. Chart.js umożliwia stworzenie wielu różnego rodzaju wykresów, które są dodatkowo responsywne, oraz pozwalają na interakcję z użytkownikiem. Wśród nich, można wyróżnić między innymi wykresy: słupkowy, liniowy, radarowy czy też kołowy. Zawartość biblioteki niestety nie jest zbyt obszerna, więc znajduje swoje zastosowanie jedynie do podstawowej, niezbyt skomplikowanej formy przedstawiania danych. Brakuje bowiem kilku ważnych w statystyce elementów, jak chociażby opcji wskazania minimum, mediany i maksimum wśród danych występujących na wykresie, automatycznego wyznaczenia linii trendu czy też regresji.

Web push - jedna z form komunikacji z użytkownikiem, przy pomocy której serwer może przekazywać informacje. Powiadomienie ma postać okienka, przeważnie pojawiającego się przez pewien okres czasu na brzegu ekranu. Struktura takiego typu wiadomości może być złożona przykładowo z: nagłówka, kilku linijek treści, zdjęcia, oraz linku do zawartości wskazanej przez nadawcę. Dopuszczalne jest również umieszczenie elementów takich jak przyciski, które po interakcji z użytkownikiem mogą wyzwolić określone akcje. Powiadomienia web push są obecnie obsługiwane przez większość najpopularniejszych przeglądarek, a do ich wysłania nie ma potrzeby zbierania od użytkowników danych typu email czy login, jak może to mieć miejsce w przypadku innych form komunikacji.

6. Projekt oraz architektura rozwiązania

6.1. Założenia oraz wymagania funkcjonalne

Głównymi założeniami co do projektu jest stworzenie mechanizmu pobierania informacji z wybranego serwisu ogłoszeniowego oraz ich zapisu w bazie danych. Kolejnym z podstawowych założeń, to umożliwienie użytkownikowi zdefiniowania parametrów funkcji pobierającej dane, podgląd ofert zgromadzonych w wyniku jej działania, oraz wyświetlenie statystyk i analiz. Poniżej przedstawiono wymagania, które powinno rozwiązanie spełniać.

Lista wymagań funkcjonalnych:

- możliwość utworzenia nowego profilu użytkownika oraz zalogowania się,
- wyświetlenie listy pobranych ogłoszeń,
- wyświetlenie pojedynczego ogłoszenia wraz możliwością pobrania pełnych danych zawartych w jego opisie,
- filtrowanie i sortowanie ogłoszeń,
- pobieranie danych w sposób cykliczny z każdej strony wyników wyszukiwania,
- możliwość zarządzania (dodawania, modyfikacji, usuwania) funkcji pobierającej dane,
- wyświetlenie wyników statystycznej analizy zgromadzonych ogłoszeń w bazie danych, przy użyciu wykresów oraz tabel. Informacje przedstawiane na wykresach powinny zostać objaśnione stosowanym komentarzem.

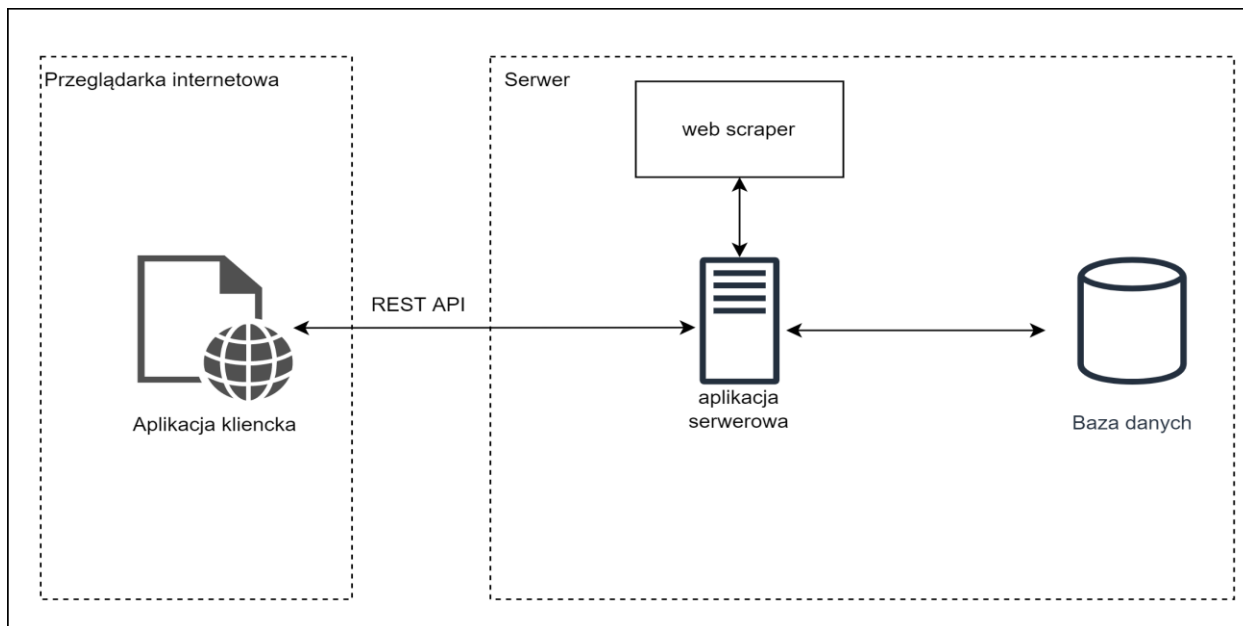
Lista wymagań niefunkcjonalnych:

- aplikacja powinna być dostępna poprzez przeglądarkę internetową, do działania nie jest wymagana instalacja dodatkowych wtyczek;
- pobieranie danych z portalu ogłoszeniowego powinno odbywać się cyklicznie, z każdej strony wyników wyszukiwania. Po zebraniu informacji ze wszystkich ogłoszeń, program powinien ponownie zacząć swoje działanie od strony pierwszej i pobrać dane, które nie zostały jeszcze zapisane;
- utworzenie funkcji pobierającej dane nie wymaga posiadania specjalistycznej wiedzy, użytkownik przy konfiguracji powinien określić jedynie źródło informacji, czyli adres url strony;
- elementy interfejsu graficznego powinny być responsywne, a wykresy na stronie w sposób przejrzysty prezentować dane.

6.2. Architektura rozwiązania

Na rysunku 7 przedstawiono architekturę proponowanego rozwiązania, która jest złożona z dwóch części. Pierwsza część to serwer, który został napisany w języku JavaScript z wykorzystaniem platformy programistycznej Express.js. Pobieranie danych odbywa się przy użyciu biblioteki Playwright. Do przechowywania informacji zastosowano relacyjną bazę danych MariaDB. Druga część, to aplikacja

przeznaczona dla klienta, która została zrealizowana z wykorzystaniem języka JavaScript i biblioteki React. Interfejs użytkownika został utworzony przy użyciu HTML, CSS oraz biblioteki Bootstrap. Do prezentacji danych na stronie w formie wykresów wykorzystano Chart.js, wraz z dodatkowym modulem, który umożliwił utworzenie wykresów pudełkowych.



Rysunek 7. diagram przedstawiający podstawową architekturę rozwiązania

6.3. Serwer

Część serwerowa ma na celu głównie realizację trzech zadań:

- komunikacji z aplikacją klienta przy pomocy API (*Application Programming interface*),
- wymianę informacji z bazą danych,
- przetwarzanie danych zebranych z serwisu ogłoszeniowych oraz pochodzących od klienta.

W celu realizacji pierwszego zadania zdefiniowano punkty końcowe, które mają obsłużyć żądania przychodzące ze strony klienta. Zostały one stworzone zgodnie z zasadami REST (*Representational State Transfer*), dane przekazywane przy użyciu API są formatu JSON (*JavaScript Object Notation*). Punkty końcowe, przy pomocy których klient ma dostęp do zasobów bazy danych, przedstawiono w tabeli 1.

Tabela 1. Punkty końcowe udostępnione klientowi.

Metoda	Ścieżka	Opis punktu końcowego	Oczekiwany kod odpowiedzi	Wyjątki
POST	/api/scrapers	Tworzy nową funkcję pobierającą dane z serwisu ogłoszeniowego	201 (utworzono zasób)	401 (użytkownik nie został uwierzytniony) 422 (przesłano niepoprawne dane) 500 (błąd techniczny serwera)
GET	/api/scrapers	Zwraca listę wszystkich funkcji, utworzonych przez aktualnie zalogowanego użytkownika	200 (operacja powiodła się)	401 (użytkownik nie został uwierzytniony) 404 (funkcje nie istnieją) 500 (błąd techniczny serwera)
GET	/api/scrapers/{id}	Zwraca parametry funkcji o podanym identyfikatorze	200 (operacja powiodła się)	401 (użytkownik nie został uwierzytniony) 404 (funkcje nie istnieją) 500 (błąd techniczny serwera)
PUT	/api/scrapers/{id}	Edycja parametrów funkcji o podanym identyfikatorze	202 (zaktualizowano dane)	401 (użytkownik nie został uwierzytniony) 404 (funkcja nie istnieje) 422 (przesłano niepoprawne dane) 500 (błąd techniczny serwera)
DELETE	/api/scrapers/{id}	Usuwa parametry funkcji o zadaniem identyfikatorze oraz oferty przez nią znalezione	204 (poprawnie wykonano akcje)	401 (użytkownik nie został uwierzytniony) 404 (brak zasobu o podanym id) 500 (błąd techniczny serwera)
GET	/api/offers	Zwraca listę wszystkich ofert	200 (operacja powiodła się)	404 (brak ofert) 500 (błąd techniczny serwera)
GET	/api/offers/{id}	Zwraca parametry oferty o podanym identyfikatorze	200 (operacja powiodła się)	404 (brak zasobu o podanym id) 500 (błąd techniczny serwera)
GET	/api/stats	Zwraca dane potrzebne do utworzenia wykresów i wykonania analiz	200 (operacja powiodła się)	404 (brak zasobów) 500 (błąd techniczny serwera)
GET	/api/stats/bestOffer	Zwraca ofertę o parametrach najbardziej dopasowanych to wymagań użytkownika	200 (operacja powiodła się)	404 (brak zasobów) 500 (błąd techniczny serwera)

GET	/api/rates/{id}	Zwracane są parametry funkcji przypisującej ofertom punkty	200 (operacja powiodła się)	401 (użytkownik nie został uwierzytniony) 404 (funkcje nie istnieje) 500 (błąd techniczny serwera)
POST	/api/rates	Utworzenie nowej funkcji	201 (utworzono zasób)	401 (użytkownik nie został uwierzytniony) 422 (przesłano niepoprawne dane) 500 (błąd techniczny serwera)
PUT	/api/rates/{id}	Edycja parametrów funkcji	202 (zaktualizowano dane)	401 (użytkownik nie został uwierzytniony) 404 (funkcja nie istnieje) 422 (przesłano niepoprawne dane) 500 (błąd techniczny serwera)

Kod aplikacji serwerowej został podzielony na mniejsze moduły:

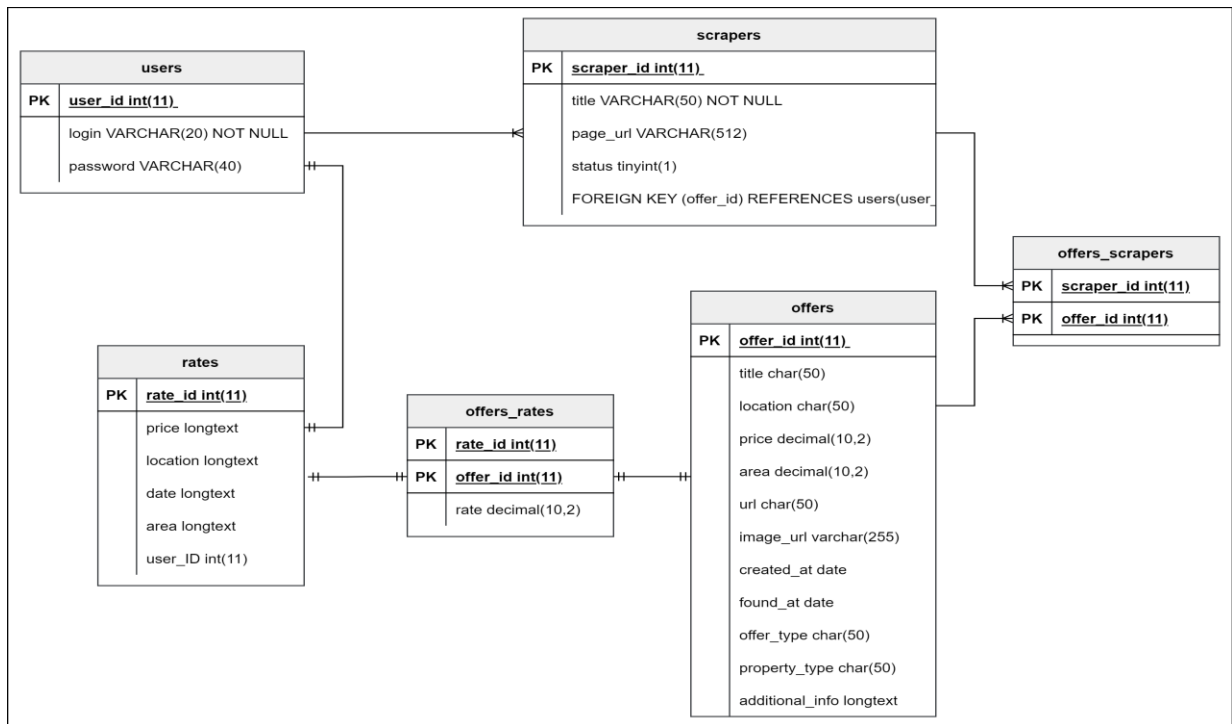
- Routes – odpowiada za ustalenie tras żądań;
- Controllers – odpowiada za obsługę przychodzących żądań ,oraz wysłanie odpowiedzi do klienta;
- Database – znajduje się w nim konfiguracja połączenia z bazą danych;
- Scripts – zawiera funkcje odpowiadające za pobieranie danych z serwisu ogłoszeniowego, ich przetwarzanie oraz zapis w bazie danych;
- Exceptions – odpowiada za obsłużenie wyjątków;
- Middleware – obsługuje proces uwierzytniania użytkowników.

Do zapisu danych wykorzystano relacyjną bazę danych, której schemat przedstawiono na rysunku 8. Struktura bazy jest złożona z sześciu tabel:

- users - w tabeli zapisane są dane użytkowników. Hasła przechowywane są w postaci haszowanej;
- scrapers - w tabeli przechowywane są parametry funkcji pobierających dane;
- offers - tabela przechowuje dane z pobranych ofert;
- rates – zawiera parametry zdefiniowane przez użytkowników, na podstawie których

przyszanawane są ofertom punkty;

- rates_offers – zawiera punkty przypisane ofertom;
- scrapers_offers – zawiera relację między funkcjami pobierającymi dane, a ofertami które udało się dzięki nim pobrać.



Rysunek 8. Schemat bazy danych

6.4. Aplikacja kliencka

Aplikacja kliencka została podzielona na mniejsze komponenty, z których każdy odpowiada za konkretny element widoku:

- Profile – wewnątrz niego umieszczone zostały komponenty odpowiadające za edycję profilu, dodanie lub modyfikacje funkcji pobierających dane i przyporządkowujących ofertom punkty;
- SortFilter – umożliwia wybór opcji według których oferty są sortowane i filtrowane;
- Statistics – w komponencie prezentowane są wyniki analiz statystycznych w postaci tabel i wykresów;
- Menu – komponent znajdujący się w górnej części widoku aplikacji, po zalogowaniu użytkownika, znajdują się w nim zakładki strony głównej, profilu, statystyk oraz przycisk za pomocą którego można się wylogować;

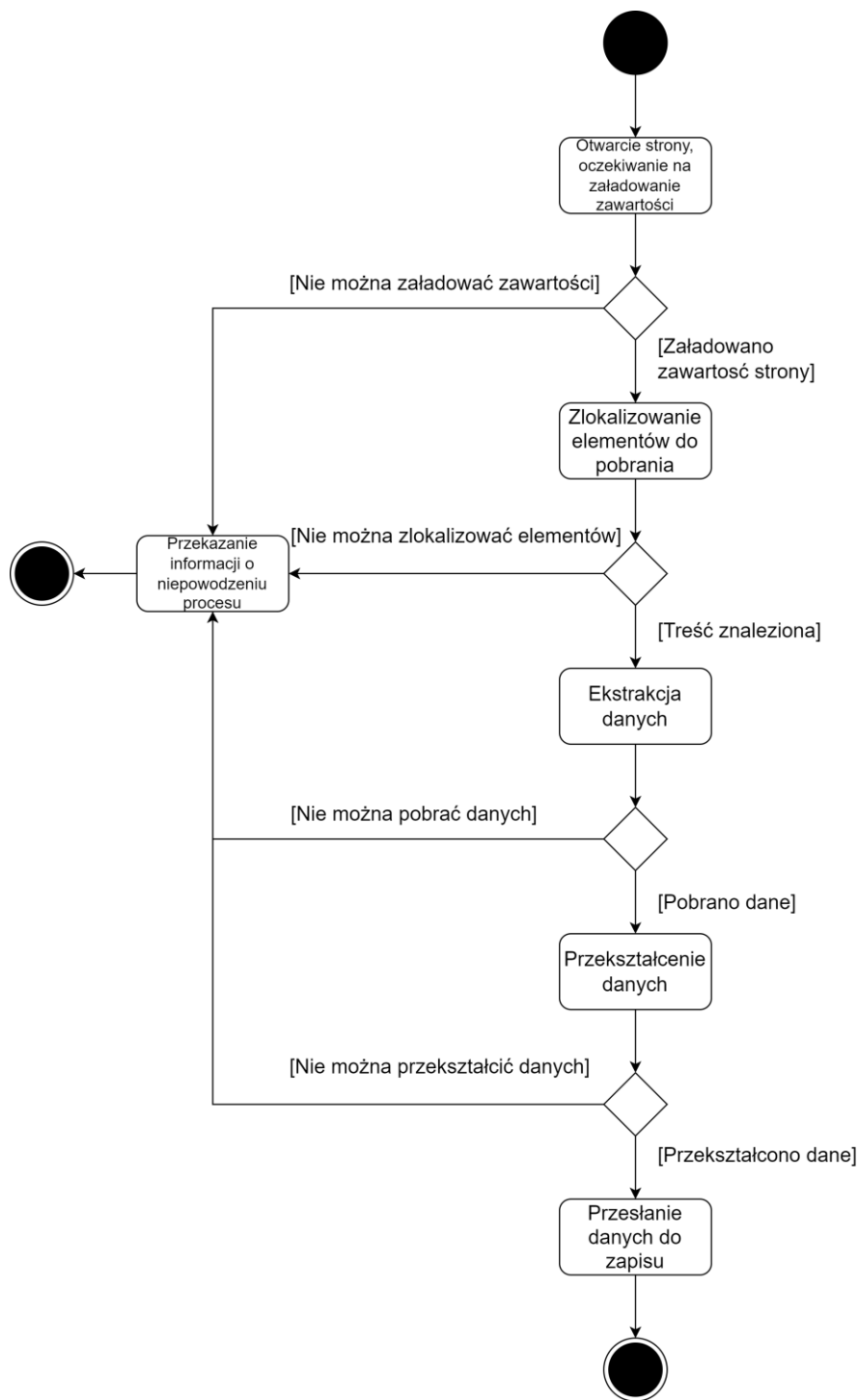
- Offers – komponent odpowiadający za prezentowanie częściowej listy pobranych ofert, zawiera element za pomocą którego można dostosować ilość wyświetlanych danych, oraz przechodzić po kolejnych stronach wyników;
- Headers – w komponencie znajduje się pasek wyszukiwania, przy pomocy którego można wyszukiwać ogłoszenia po tytule;
- ElementsUI – zawiera mniejsze elementy interfejsu użytkownika, takie jak ikona ładowania czy pasek wyszukiwania;
- BestOffer – komponent wyświetlający najlepszą ofertę znaną dla użytkownika, pojawia się po wcześniejszym zalogowaniu i podaniu parametrów funkcji przypisującej punkty.

7. Szczegóły implementacyjne

W rozdziale opisano sposób implementacji funkcjonalności, określonych w wymaganiach funkcjonalnych, które rozwiązanie powinno spełniać. Przedstawiony został również rezultat przeprowadzonych prac oraz działanie interfejsu użytkownika, którego test przeprowadzono na urządzeniu mobilnym.

7.1. Funkcja pobierająca dane

Proces pobierania informacji rozpoczyna się od odczytania z bazy danych parametrów funkcji pobierających dane, które zostały oznaczone przez użytkowników jako aktywne. Funkcja jest uruchamiana w takim samym odstępie czasowym, który został określony za pomocą narzędzia cron. Dane zebrane przy użyciu funkcji, po zakończonym jej działaniu zostają umieszczone w bazie danych. Program gdy pobierze informacje z każdej z podstron wyników wyszukiwań, rozpoczyna swoje działanie od strony pierwszej. Jednak tym razem, pobierane będą wyłącznie nowe oferty, funkcja po wykryciu ogłoszenia znajdującego się już w bazie, zakończy przeszukiwanie i wznowi proces od strony pierwszej. Metoda działania funkcji pobierającej dane, została przedstawiona na rysunku 9.



Rysunek 9. Diagram aktywności funkcji pobierającej dane

Przed rozpoczęciem właściwej części pracy, funkcja oczekuje aż przeglądarka w pełni załaduje zawartość serwisu internetowego. Następnie za pomocą selektora *element > li*, na stronie lokalizowane są wszystkie elementy zawierające treści ogłoszeń. Każdy z tych elementów zawiera informacje, które przedstawiono w tabeli 2.

Tabela 2. Elementy pobrane za pomocą scrapera

Element ogłoszenia	Selektor
Tytuł	<code>.title</code>
Lokalizacja	<code>.address > div > a > strong</code>
Cena	<code>.pricing</code>
Data dodania	<code>.time</code>
Powierzchnia	<code>.offer-list-item-footer > .attributes > .first</code>
Adres url podstrony z pełną treścią ogłoszenia	<code>.title a</code>
Główne zdjęcie zawarte w ogłoszeniu	<code>.img</code>

Pobranie elementów wymienionych w tabeli 2, wymagało użycia funkcji *querySelector*. Jako jej argument należało podać selektor lub grupę selektorów za pomocą których, można zlokalizować pożądany element ogłoszenia. Treść ogłoszenia zawiera również inne dane, których pozyskanie wymagało użycia większej ilości kodu. Informacje, o rodzaju oferty oraz nieruchomości zostały wyodrębnione równocześnie, przy wykorzystaniu jednego selektora *.location*. W wyniku działania funkcji *querySelector* pobrano ciąg znaków którego przykładowa zawartość może być następująca: „https://sprzedajemy.pl/serock/nieruchomosci/domy/sprzedaz”. Do uzyskania danych w docelowej postaci wymagane jest użycie metody *split*, która jako argument przyjmuje znak – separator dzięki któremu ciąg zostanie podzielony na pojedyncze elementy (fragment kodu 1).

```
let category = card.querySelector(".location");
if (category) {
  category = category.href;
  category = category.split("/");
  propertyType = category[category.length - 2];
  offerType = category[category.length - 1];
}
```

Fragment kodu 1. Pobranie informacji o typie oferty oraz nieruchomości

Zgromadzono również unikalny identyfikator oferty, który wykorzystano w bazie danych jako klucz główny tabeli zawierającej dane ogłoszeń. Do pobrania użyto funkcji *querySelector* wraz z atrybutem *href*. Następnie wykorzystując metodę *substr*, uzyskano z adresu 8 ostatnich cyfr, które stanowią właśnie ID oferty (fragment kodu 2).

```
let url = card.querySelector(".title a").href;
let id = url.substr(url.lastIndexOf("r") + 1, url.length);
```

Fragment kodu 2. Pobranie informacji o typie oferty oraz nieruchomości

W zależności od rodzaju nieruchomości, zakres danych zawartych w ogłoszeniu może się różnić. Oferty mieszkań uwzględniają informacje na temat liczby pokoi, rodzaju zabudowy oraz o numerze piętra. W przypadku domu jest to informacja o powierzchni działki oraz rodzaju zabudowy. Wymienione dane pozyskano używając funkcji *querySelectorAll*, która zwraca wszystkie elementy pasujące w tym przypadku do grupy selektorów *offer-list-item-footer > .attributes* (fragment kodu 3).

```
let additionalInfo = card.querySelectorAll(
  ".offer-list-item-footer > .attributes"
);
if (additionalInfo) {
  additionalInfo = Array.from(additionalInfo, (info) => {
    info = info.textContent.replace(/(\r\n|\n|\r|\t)/gm, "");
    info = info
      .split(" ")
      .filter((e) => e)
      .join(" ");

    return info;
  });
}
```

Fragment kodu 3. Pobranie informacji charakterystycznych dla danego typu nieruchomości

Program ma umożliwić przechodzenie funkcji pobierającej dane po każdej stronie wyników wyszukiwania, tak aby zebrać informacje z każdego ogłoszenia. W tym celu, należało pobrać informacje o numerze strony, na której aktualnie znajduje się funkcja, jak również sprawdzić czy istnieje możliwość

przejścia do następnej (fragment kodu 4). Jeśli istnieje taka możliwość, zwracany jest adres URL tej strony, który wykorzystywany jest przy kolejnym uruchomieniu funkcji. W przeciwnym razie następuje powrót funkcji do strony pierwszej. Wykrywany jest również przypadek, gdzie na stronie brakuje elementu zawierającego paginację, co oznacza, że istnieje tylko jedna strona wyników wyszukiwania.

```
let Page;
try {
  Page = await page.$eval(
    ".cntListTools",
    (el, url) => {
      let nextPage = el.querySelector(".next a");
      let lastPage = "";
      let prevPage = el.querySelector(".previous a");
      let pagination = el.querySelectorAll("ul > li > a");
      if (nextPage !== undefined) {
        firstPage = prevPage === undefined ? url : pagination[1].href;
        nextPage = nextPage.href;
        lastPage = pagination[pagination.length - 2].href;
      }

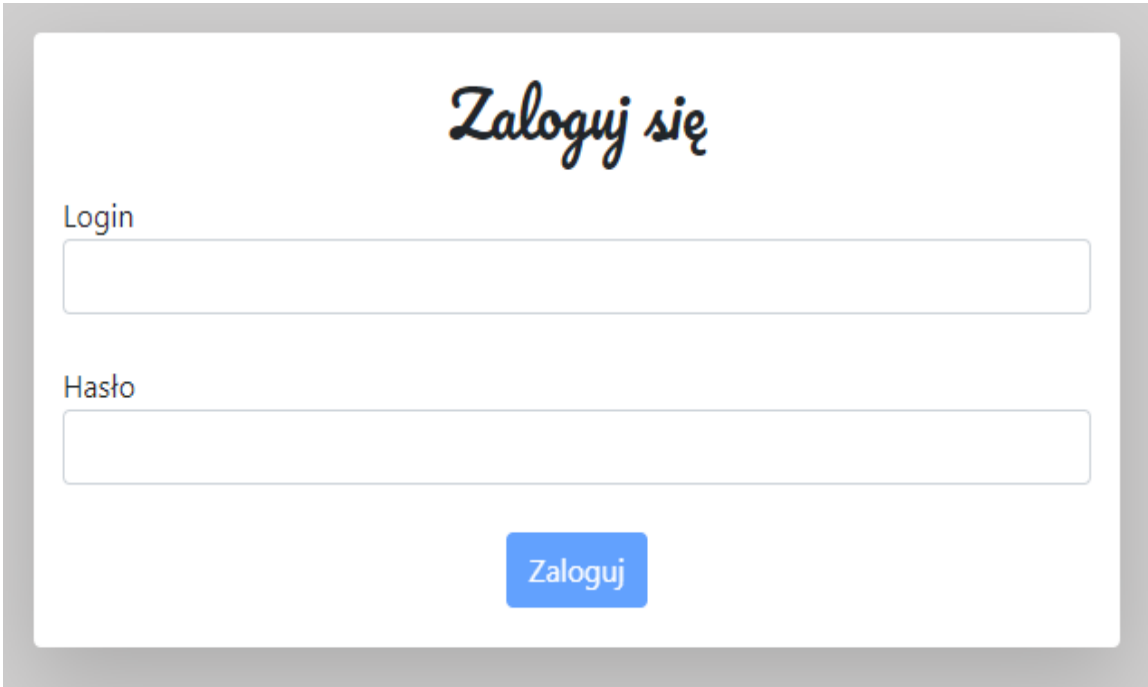
      if (nextPage === undefined) {
        firstPage = prevPage === undefined ? url : pagination[1].href;
        nextPage = pagination[1].href;
        lastPage = url;
      }

      return { nextPage, lastPage, firstPage };
    },
    url
  );
} catch (err) {
  Page = {
    nextPage: url,
    lastPage: url,
    firstPage: url,
  };
}
```

Fragment kodu 4. Pobranie numerów stron

7.2. Interfejs użytkownika

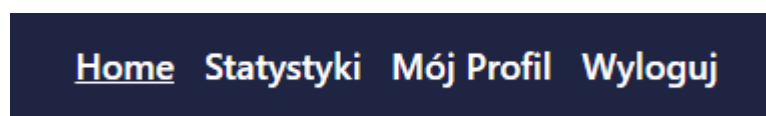
Po wprowadzeniu poprawnych danych w panelu logowania (rysunek 10), użytkownik zyskuje dostęp do wszystkich funkcjonalności portalu. W procesie uwierzytelniania użyto algorytm bcrypt, który wykorzystywany jest do haszowania haseł. Gdy uwierzytelnianie zakończy się pomyślnie, użytkownikowi przesyłany jest Token JWT, który zawiera identyfikator użytkownika. Przy jego użyciu, możliwy jest dostęp do chronionych zasobów, które pokazano w tabeli 1. Token jest ważny przez jedną godzinę, po wygaśnięciu jest on odświeżany, tak aby możliwy był dalszy dostęp bez konieczności ponownego wprowadzania danych. Login, identyfikator oraz tokeny umieszczono w pamięci lokalnej przeglądarki. Po wylogowaniu, następuje usunięcie danych użytkownika z pamięci lokalnej.



The image shows a login panel with a light gray border. At the top, the text "Zaloguj się" is written in a large, dark blue, cursive font. Below this, there are two input fields. The first is labeled "Login" in a small, gray font, and the second is labeled "Hasło" in the same font. Both fields are empty and have a light gray border. Below the input fields, there is a blue button with the text "Zaloguj" in white, sans-serif font.

Rysunek 10. Panel logowania

Na rysunku 11 przedstawiono cztery zakładki umiejscowione w menu na samej górze strony, do których użytkownik ma dostęp po zalogowaniu



Rysunek 11. Widok menu strony po zalogowaniu

Naciśnięcie zakładki *Home* powoduje przekierowanie do strony głównej, gdzie prezentowane są znalezione ogłoszenia. Użytkownik może dostosować wyświetlanie wyników poprzez różne opcje filtrowania i sortowania. Każda pobrana oferta umieszczona jest wewnątrz karty, po kliknięciu przycisku *zobacz* następuje przejście do podstrony z pełnymi informacjami o ogłoszeniu (rysunek 12).

Wybierz z listy

Sortuj po:

dacie

malejąco

Cena od:

do:

Pow. od:

do:

Lokalizacja:

Data od:

dd.mm.rrrr

do:

dd.mm.rrrr

Typ nieruchomości:

wybierz

Typ oferty:

wybierz

Pokaż

Najlepsza oferta

Działka Dom Grębałów Kraków bliźniak 14 a 480 m2 media

Cena: 400000 złotych

Pokaż


Pokaż moje oferty

☐

Ofert na stronie:

5

NA STRONIE: 5 OFERT



Mieszkanie 47m2 2 pokoje Kraków Sołtysowska

Lokalizacja: Kraków

Powierzchnia: 47 m²

Data dodania oferty: 2022-01-10

Data znalezienia oferty: 2022-01-10

Cena: 580000 zł

Zobacz

Rysunek 12. Widok strony głównej

W chwili otwarcia strony z ofertą przedstawiane są jedynie podstawowe informacje, które zawarto w tabeli *Dane oferty*. Więcej szczegółów można poznać po kliknięciu przycisku *pobierz pełną treść oferty*. Uruchamiany jest wtedy funkcja pobierająca wszystkie informacje umieszczone w ogłoszeniu, łącznie z jej tekstowym opisem (rysunek 13).

24

Dom wolnostojący Warszawa 100m2

Pełne informacje o ofercie:

Oferta od	firmy
Rynek	wtórny
Cena za m ²	20000 zł/m ²
Powierzchnia	100 m ²
Powierzchnia działki	1800 m ²
Zabudowa	wolnostojący
Liczba pięter	-
Stan	do remontu
Rok budowy	-
Ogrzewanie	-
Liczba pokoi	2
Materiał budynku	-
Pokrycie dachu	-
Typ	mieszkalny

Dom z działką do wyburzenia lub remontu w Marysinie Wawerskim. 3 pokoje z kuchnią, widna łazienka, parkiet, piwnica, poddasze nieużytkowe. Wysokość 3m. Blisko do Mazowieckiego Parku Krajobrazowego. Kontakt: Marek Skibniewski o@b*****.pl tel. 603 399 226 lub 691 47 88 03 Treść niniejszego ogłoszenia nie stanowi oferty handlowej w rozumieniu Kodeksu Cywilnego. Oferta wysłana z programu IMO dla biur nieruchomości

Dane oferty:

Id	Miasto	Cena	Powierzchnia	Data dodania oferty	Link do oferty	Kategoria	Typ oferty
62373407	Warszawa	2000000	100	2021-12-09	https://sprzedajemy.pl/dom-wolnostojacy-warszawa-100m2-2-1b8e55-nr62373407	domy	sprzedaż

Rysunek 13. Widok strony oferty z pełnym opisem oferty

Naciśnięcie w menu zakładki *Mój profil* powoduje przekierowanie do strony, gdzie użytkownik ma możliwość edycji swojego loginu i hasła, modyfikacji parametrów istniejących funkcji pobierających dane bądź też utworzenie nowych (rysunek 14).

Moje dane

Scraper

Najlepsza oferta

login

kubasn1

password

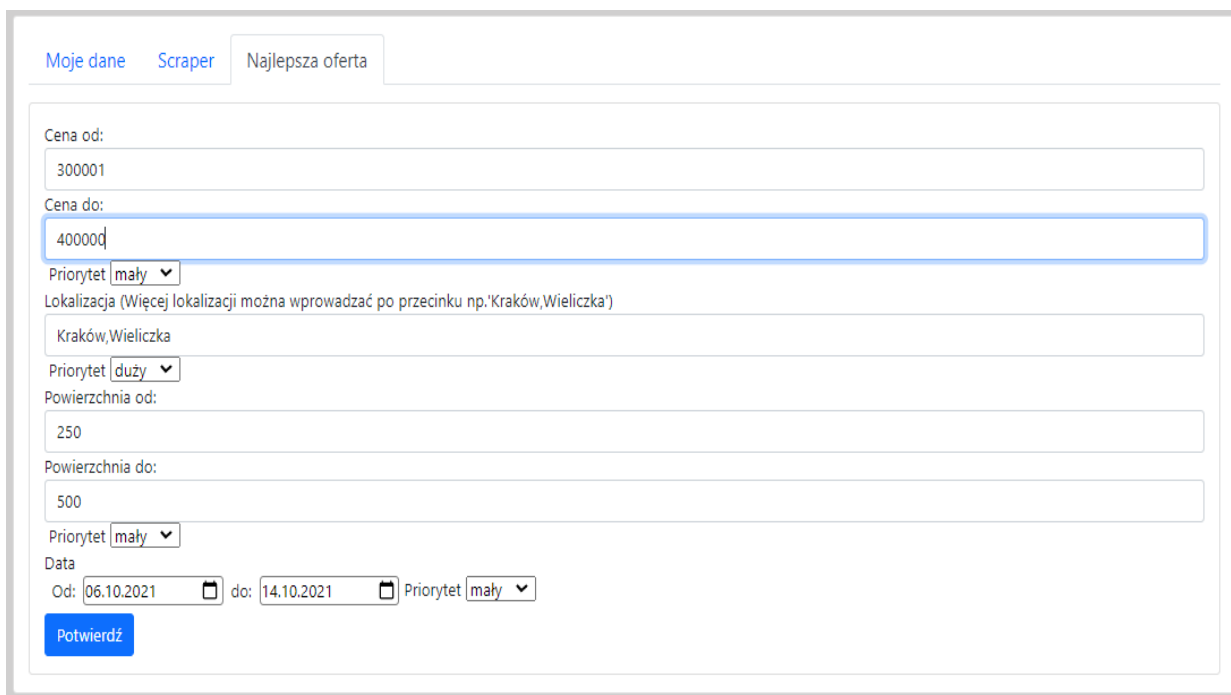
••

Hasło za krótkie min. 5 znaków

Potwierdź

Rysunek 14. Widok zakładki *Mój profil*

Zakładka zawiera również podstronę, gdzie po wypełnieniu formularza można uruchomić funkcję, której zadaniem jest przyporządkowanie każdej ofercie punktów na podstawie podanych parametrów. Dzięki temu, można wyłonić spośród wszystkich ofert tą najbardziej dopasowaną do oczekiwań klienta. Na stronie głównej, po zaznaczeniu odpowiedniej opcji można również wyświetlić oferty posortowane według ilości punktów (rysunek 15).



The screenshot shows a web interface with three tabs: "Moje dane", "Scraper", and "Najlepsza oferta". The "Najlepsza oferta" tab is active. The form contains the following fields and controls:

- "Cena od:" text input with value "300001".
- "Cena do:" text input with value "400000", which is highlighted with a blue border.
- "Priorytet" dropdown menu with "mały" selected.
- "Lokalizacja (Więcej lokalizacji można wprowadzać po przecinku np.'Kraków,Wieliczka')" text input with value "Kraków,Wieliczka".
- "Priorytet" dropdown menu with "duży" selected.
- "Powierzchnia od:" text input with value "250".
- "Powierzchnia do:" text input with value "500".
- "Priorytet" dropdown menu with "mały" selected.
- "Data" section with "Od:" date input (06.10.2021), "do:" date input (14.10.2021), and a "Priorytet" dropdown menu with "mały" selected.
- A blue "Potwierdź" button at the bottom left.

Rysunek 15. Widok strony na której można zdefiniować parametry funkcji przyporządkowującej punkty ogłoszeniom

Do całościowej oceny oferty, brane jest pod uwagę każde z poniższych kryteriów:

- Cena,
- Lokalizacja,
- Powierzchnia,
- Data dodania ogłoszenia.

Dla każdego z parametrów ogłoszenia, funkcja może przyznać maksymalnie 1 punkt. Dla danych typu: cena, data i powierzchnia ocena będzie tym większa, im ich wartości znajdą się bliżej przedziału zadanego przez użytkownika. W przypadku lokalizacji, oferta dostanie punkt tylko, jeśli jest ona zgodna z tą określoną przez użytkownika. Ocena jest pomnażana dodatkowo przez wagę, która jest liczbą całkowitą i mieści się w przedziale liczbowym od 1 do 3. Wybierając większą z nich, użytkownik ma możliwość wskazania parametrów ogłoszenia, których składowe oceny mają mieć większy wpływ na

wynik końcowy. Na rysunku 16, ukazano widok strony głównej, gdzie przedstawiana jest najlepsza oferta znaleziona dla użytkownika, po zaznaczeniu odpowiedniej opcji ogłoszenia mogą być również posortowane według ilości przydzielonych punktów.

The screenshot displays a web interface for finding real estate. At the top, there's a search bar titled "Wybierz z listy" with various filters: "Sortuj po" (sorted by) with a dropdown set to "dopasowaniu" (fitting), "malejąco" (decreasing), "Cena od:" (price from), "do:" (to), "Powierzchnia od:" (area from), "do:" (to), "Lokalizacja" (location), "Data dodania od:" (date added from), "do:" (to), "Typ nieruchomości" (type of property), and "Typ oferty" (type of offer). A blue "Pokaż" (show) button is on the right. Below this is a green banner for the "Najlepsza oferta" (Best offer) titled "Do sprzedaży dom wolnostojący Kraków 150m2" with a price of "Cena: 410000 złotych" and a "Pokaż" button. Further down, there's a section "NA STRONIE: 5 OFERT" (ON PAGE: 5 OFFERS). The first offer is shown with a photo of a house, the title "Do sprzedaży dom wolnostojący Kraków 150m2", price "Cena: 410000 zł", points "Punkty: 5.28", and a "Zobacz" (view) button. At the bottom right, it shows "Data dodania oferty: 2021-10-11" and "Data znalezienia oferty: 2021-10-12".

Rysunek 16. Strona główna z elementem przedstawiającym najlepszą ofertę

7.3. Powiadomienie o okazyjnej ofercie

W trakcie procesu zapisywania danych ogłoszenia, obliczana jest średnia cena nieruchomości zawartych już w bazie, których parametry są zbliżone do tych umieszczonych w przesyłanej ofercie, co zostało przedstawione w fragmencie kodu 5. Jeżeli cena umieszczona w aktualnie zapisywanym ogłoszeniu jest znacząco mniejsza od średniej, program za pomocą powiadomienia poinformuje użytkownika o znalezieniu okazyjnej oferty. Na potrzeby niniejszego projektu założono że znacząca różnica cen to 20 procent.

```

const minArea = parseFloat(offer.area) - parseFloat(offer.area) * 0.2;
const maxArea = parseFloat(offer.area) + parseFloat(offer.area) * 0.2;

if (offer.property_type == "mieszkania") {
  const rooms = JSON.parse(offer.additional_info).rooms;
  sql = [
    `SELECT AVG(price) from offers WHERE location LIKE ?
    and property_type= ? and offer_type = ? and area between ? and ?
    and JSON_EXTRACT(additional_info,'$.rooms')=?`,
    [
      offer.location, offer.property_type, offer.offer_type, minArea,
      maxArea, rooms,
    ],
  ];
}
if (offer.property_type == "domy") {
  if (JSON.parse(offer.additional_info).lotArea > 10) {
    const lotArea = parseFloat(
      JSON.parse(offer.additional_info).lotArea
    );
    const minLotArea = lotArea - lotArea * 0.2;
    const maxLotArea = lotArea + lotArea * 0.2;
    sql = [
      `SELECT AVG(price) from offers WHERE location LIKE ?
      and property_type= ? and offer_type = ? and area between ?
      and ? and JSON_EXTRACT(additional_info,'$.lotArea') between ? and ?`,
      [offer.location, offer.property_type, offer.offer_type, minArea, maxArea,
minLotArea, maxLotArea,
      ],
    ];
  }
}
}

```

Fragment kodu 5. Pobranie średniej ceny nieruchomości o podobnych parametrach

Użytkownik otrzyma powiadomienie o ofercie jedynie, gdy została ona znaleziona przez funkcję pobierającą informacje, którą utworzył. Do wyświetlenia komunikatu wymagana jest włączona przeglądarka, oraz wcześniejsze zalogowanie się na stronie. Ponadto, witryna może potrzebować od użytkownika przyznania zgody na wysyłanie powiadomień, gdyż wiele przeglądarek domyślniej blokuje ten rodzaj wiadomości. Utworzenie wiadomości *web push* przedstawiono we fragmencie kodu 6.

```

self.addListener("push", (e) => {
  localforage.getItem("key").then((id) => {
    const data = e.data.json();
    let info = "";
    if (data.userId == id) {
      self.registration.showNotification(`Znaleziono okazję: ${data.title}`, {
        body: `\\nCena: ${data.price} złotych (-${data.priceDifference}% taniej od
średniej ceny) Lokalizacja: ${data.location}
\\nInformacje: ${data.keyWords} ${data.buildingYearInfo}`,
        image: data.image,
        data: { url: data.url },
      });
    }
  });
});

```

Fragment kodu 6. Treść powiadomienia

Wygląd powiadomienia przedstawiono na rysunku 17. Treść jego zawiera parametry nieruchomości takie jak: cena, lokalizacja oraz rok budowy. Dodatkowo, w treści ogłoszenia wyszukiwane są frazy zdefiniowane w kodzie programu, które mają znaczący wpływ na niższą wartość nieruchomości. Dzięki temu użytkownik, bez konieczności przechodzenia do strony z ofertą, może z góry odrzucić nieruchomość obciążoną wadami bądź brakami, które są przez niego nie do zaakceptowania.



Rysunek 17. Widok powiadomienia prezentującego okazijną ofertę

7.4. Statystyczna analiza ofert

Wyniki analiz statystycznych przedstawiono na stronie w postaci wykresów oraz tabel (rysunki 18 i 19). Dla parametrów takich jak powierzchnia użytkowa nieruchomości oraz cena za jej metr kwadratowy wyliczono wartości: maksimum, minimum, zakres, średnia oraz mediana.

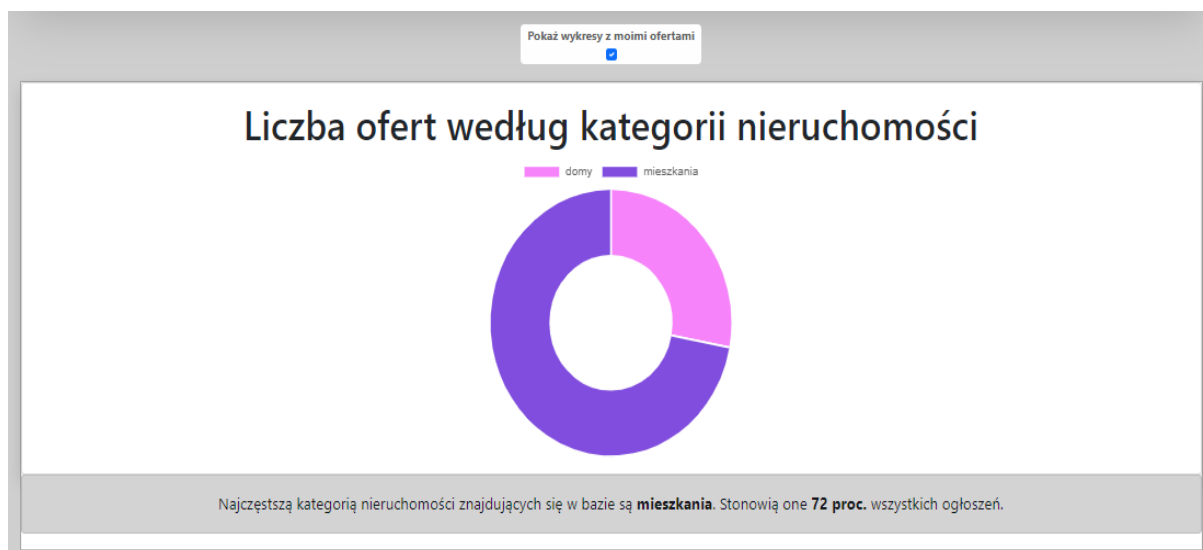
Cena za 1 metr kwadratowy powierzchni użytkowej							
Lokalizacja	Rodzaj nieruchomości	Liczba ofert	Maksimum	Minimum	Zakres	Średnia	Mediana
Warszawa	domy	489	31000	2075.47	28924.53	7747.21	7062.44
Warszawa	mieszkania	1732	29702.97	2173.91	27529.06	13097.84	12500
Kraków	mieszkania	7199	40280	2487.5	37792.5	11210.03	10625
Kraków	domy	377	23636.36	2368.42	21267.94	6855.16	5862.07
Myślenice	domy	27	17914.44	2421.05	15493.39	5644.64	4729.32
Dąbrowa Tarnowska	domy	9	4343.48	2083.33	2260.15	2955.9	2806.6

Rysunek 18. Tabela przedstawiająca analizę ceny za 1 metr kwadratowy powierzchni użytkowej

Powierzchnia użytkowa							
Lokalizacja	Rodzaj nieruchomości	Liczba ofert	Maksimum	Minimum	Zakres	Średnia	Mediana
Warszawa	domy	489	977	102	875	285.39	250
Warszawa	mieszkania	1732	350	21	329	61.99	53
Kraków	mieszkania	7199	618	21	597	57.44	52
Kraków	domy	377	868	106	762	252.1	200
Myślenice	domy	27	460	113	347	217.33	187
Dąbrowa Tarnowska	domy	9	230	114	116	165.67	153

Rysunek 19. Tabela przedstawiająca analizę ceny za 1 metr kwadratowy powierzchni użytkowej

Wykresy mogą przedstawiać dane pobrane ze wszystkich ogłoszeń lub te spersonalizowane dla konkretnego użytkownika. Pod wykresem zamieszczono komentarz, który jest krótkim dopełnieniem przedstawianych informacji. Wartości w nim zawarte, mogą się zmienić w przypadku gdy do bazy zostaną zapisane nowe ogłoszenia.



Rysunek 20. Wykres przedstawiający liczebność poszczególnych kategorii nieruchomości

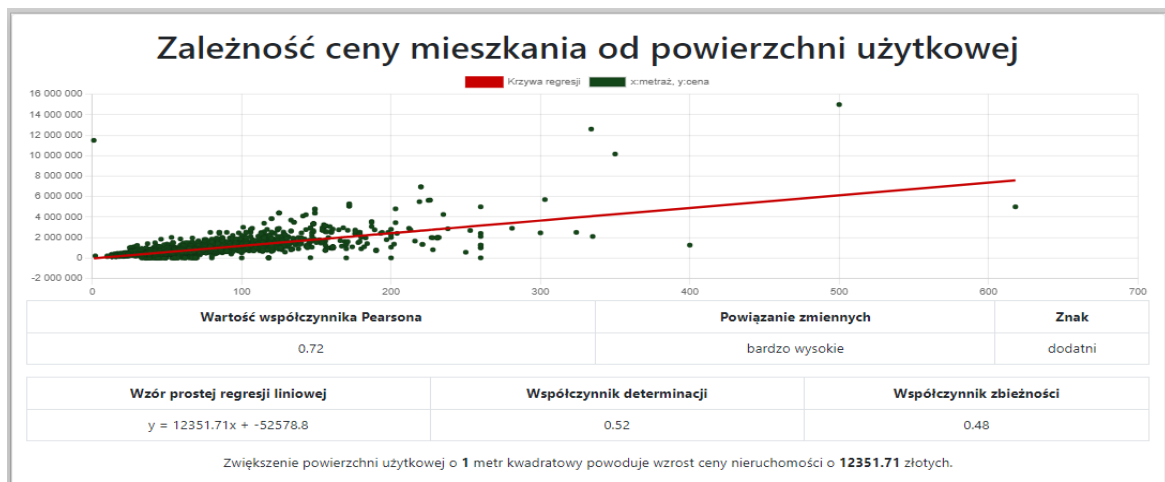
Wykresy podobnego typu jak przedstawiony na rysunku 20, przedstawiające ilościowe rozkład parametrów nieruchomości, zostały utworzone po wcześniejszym przekształceniu danych do postaci wymaganej przez dany typ wykresu. Następnie, dane umieszczono w szablonie dostarczonym przez bibliotekę, określając dodatkowo przy tym typ i tytuł wykresu, oraz etykiety dla danych (fragment kodu 7).

```
<div className={styles.charts}>
  <Chart
    type="doughnut"
    data={chartData.propertyType.data}
    labels={chartData.propertyType.labels}
    title="Liczba ofert według kategorii nieruchomości"
  />
  <ChartsComment data={chartData.propertyType} type="propertyType" />
</div>
```

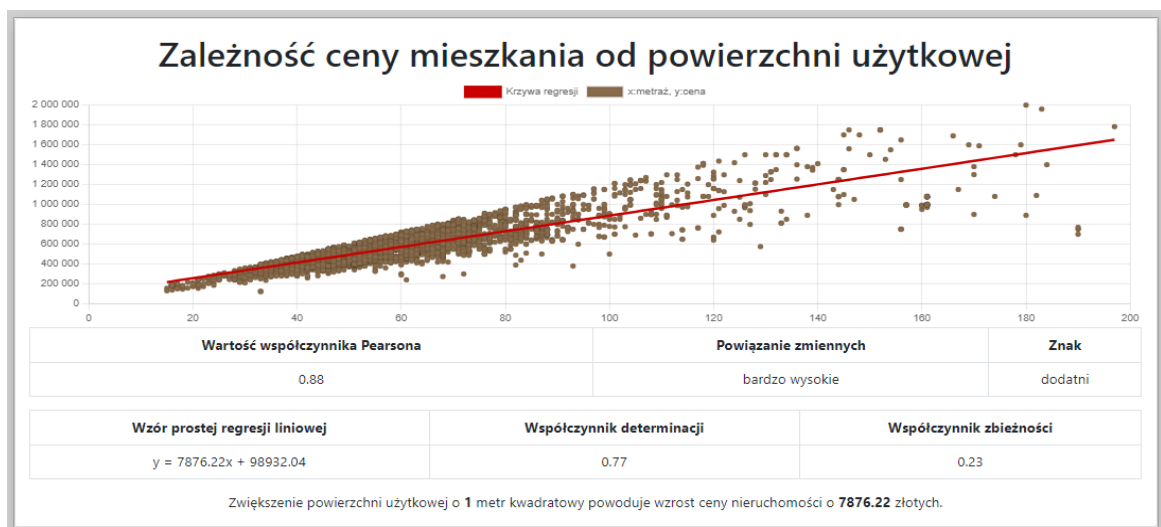
Fragment kodu 7. Metoda tworzenie wykresu przy użyciu biblioteki

Na stronie umieszczono również grupę wykresów bardziej szczegółowych, użytych do przedstawienia zależności, które występują między niektórymi parametrami. Jest to także próba odpowiedzi na pytanie jakie atrybuty nieruchomości mogą mieć wpływ na kształtowanie jej ceny. Dane poddano wcześniejszej selekcji, pominięto oferty zawierające podejrzenie niską cenę oraz powierzchnię, które mogły zostać dodane przez pomyłkę. Odrzucono również oferty, dla których obliczony stosunek

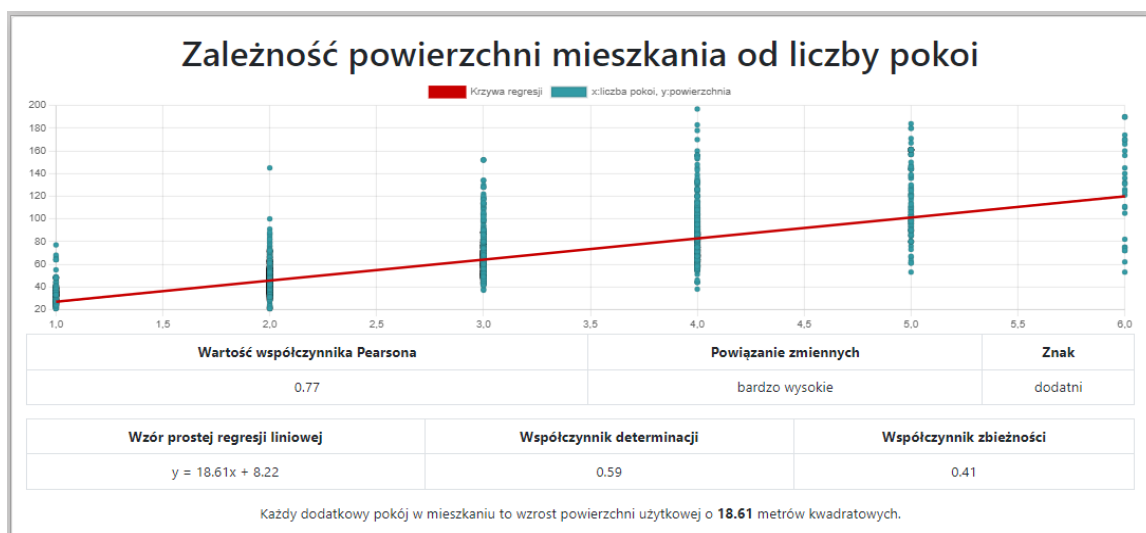
ceny do metra kwadratowego powierzchni, znacznie różnił się od wartości uzyskanych w pozostałych ogłoszeniach. Takie odfiltrowanie danych poprawiło postać regresji, wrażliwą na obserwacje odstające, która mogłaby zostać przez nie naruszona. Na rysunku 21 przedstawiono wykres przed normalizacją, natomiast wykresy po odrzuceniu danych odstających przedstawiane są na rysunkach: 22, 23, 24, 25.



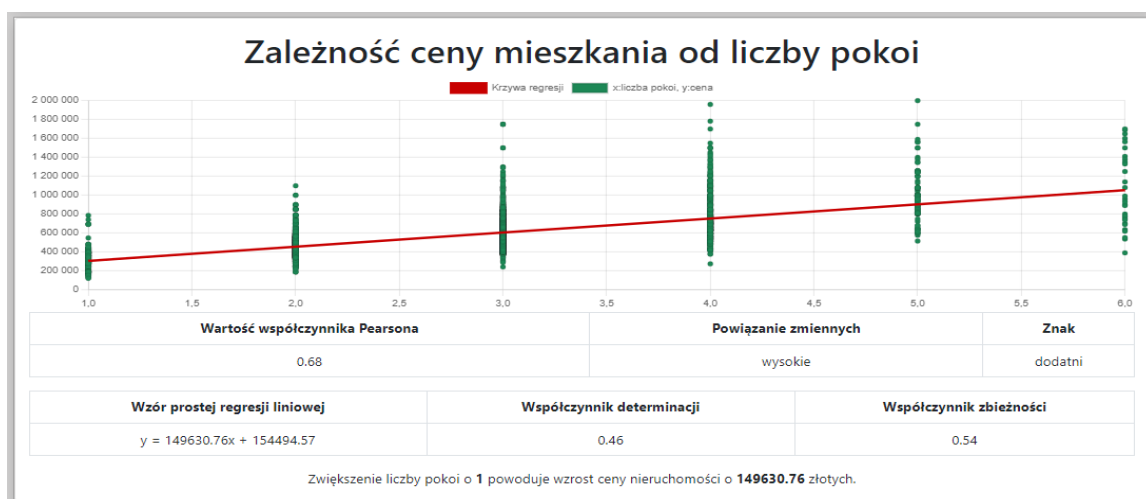
Rysunek 21. Wykres przed normalizacją danych



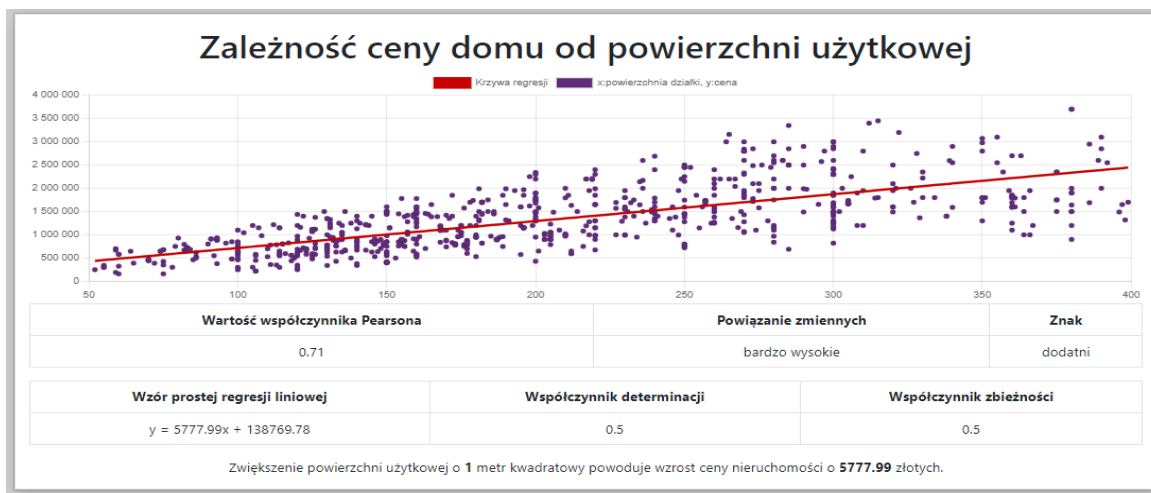
Rysunek 22. Wykres przedstawiający zależność ceny mieszkania od liczby pokoi po normalizacji danych



Rysunek 23. Wykres przedstawiający zależność powierzchni mieszkania od liczby pokoi

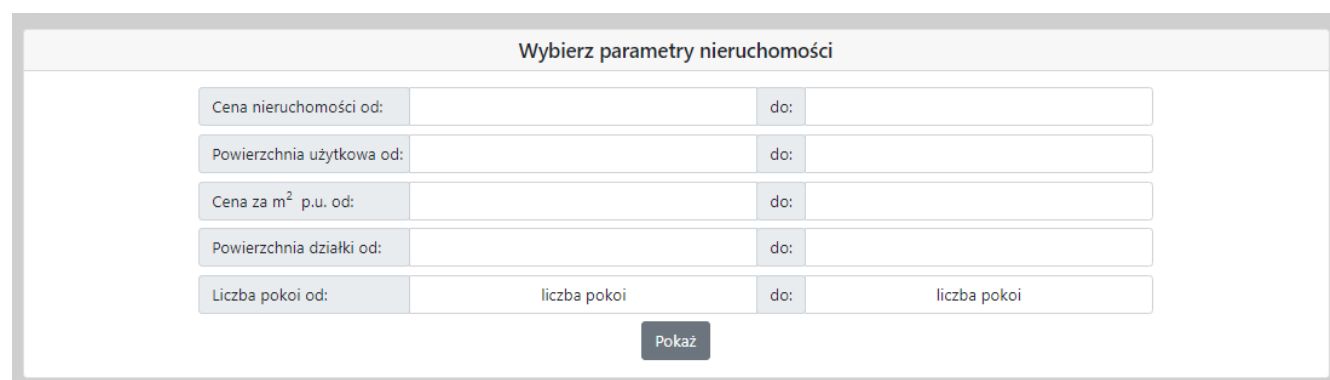


Rysunek 24. Wykres przedstawiający zależność ceny mieszkania od liczby pokoi



Rysunek 25. Wykres przedstawiający zależność ceny domu od powierzchni użytkowej

Oprócz generowania wykresów na podstawie wartości określonych w kodzie programu, istnieje również możliwość selekcji danych używanych do analizy statystycznej, poprzez wprowadzenie przez użytkownika odpowiednich wartości w formularzu, który przedstawiono na rysunku 26.



Wybierz parametry nieruchomości

Cena nieruchomości od:		do:	
Powierzchnia użytkowa od:		do:	
Cena za m ² p.u. od:		do:	
Powierzchnia działki od:		do:	
Liczba pokoi od:	liczba pokoi	do:	liczba pokoi

Pokaż

Rysunek 26. Formularz do wprowadzenia wartości parametrów na podstawie których tworzone są wykresy

Na wykresie oprócz danych naniesiono dodatkowo linię regresji, która ma określić kierunek zmiany wartości jednego parametru względem drugiego. Niestety, biblioteka chart.js nie zawiera gotowego modułu, który umożliwiłby automatyczne generowanie i umieszczenie jej na wykresie. Dlatego stworzono implementację modelu regresji linowej, która dobrze odpowiada badanym relacjom (fragment kodu 8). Funkcja jako argument przyjmuje tablice będącą zbiorem wartości dwóch zmiennych. Na ich podstawie przy użyciu metody najmniejszych kwadratów, wyznaczano wartości regresji, które następnie zostały zwrócone przez funkcję i naniesione na wykres.

```

function linearRegression(data) {
  let sum_x = 0;
  let sum_y = 0;
  let sum_xx = 0;
  let sum_xy = 0;
  let n = 0;

  for (let i = 0; i < data.length; i++) {
    if (data[i][1] !== null) {
      n++;
      sum_x += data[i][0];
      sum_y += data[i][1];
      sum_xx += data[i][0] * data[i][0];
      sum_xy += data[i][0] * data[i][1];
    }
  }

  const rise = n * sum_xy - sum_x * sum_y;
  const run = n * sum_xx - sum_x * sum_x;

  const a = run === 0 ? 0 : round(rise / run, 2);
  const b = round(sum_y / n - (a * sum_x) / n, 2);
  const calcPoints = (point) => [round(point, 2), round(a * point + b, 2)];

  const points = data.map((point) => calcPoints(point[0]));
  return {
    points,
  };
}
export default linearRegression;

```

Fragment kodu 8. Implementacja funkcji obliczającej regresję liniową

Do zbadania czy para zmiennych jest ze sobą powiązana, wykorzystano współczynnik korelacji Pearsona, implementację przedstawiono we fragmencie kodu 9. W celu wyznaczenia wartości współczynnika przygotowano funkcję *pearsonCorrelation* która wykonuje obliczenia zgodnie ze wzorem [14]:

$$r = \frac{n * \sum XY - \sum X - \sum Y}{\sqrt{[n * \sum X^2 - (\sum X)^2]} * \sqrt{[n * \sum Y^2 - (\sum Y)^2]}} \quad (1)$$

gdzie: n – liczebność zbioru, X- wartości pierwszej zmiennej, Y- wartości drugiej zmiennej. Implementację współczynnika korelacji Pearsona przedstawiono we fragmencie kodu 9.

```
const pearsonCorrelaction = (x, y) => {
  let sum_x = 0;
  let sum_y = 0;
  let sum_xy = 0;
  let sum_x2 = 0;
  let sum_y2 = 0;
  const length = Math.min(x.length, y.length);
  for (let i = 0; i < length; i++) {
    const y_i = y[i];
    const x_i = x[i];
    sum_x += x_i;
    sum_y += y_i;
    sum_x2 += x_i * x_i;
    sum_y2 += y_i * y_i;
    sum_xy += x_i * y_i;
  }

  let numerator = length * sum_xy - sum_x * sum_y;
  let denominator =
    Math.sqrt(length * sum_x2 - sum_x * sum_x) *
    Math.sqrt(length * sum_y2 - sum_y * sum_y);
  let r = numerator / denominator;
  return round(r, 2);
};
```

Fragment kodu 9. Implementacja funkcji obliczającej współczynnik korelacji Pearsona

Funkcja zwraca obliczoną wartość, która dodatkowo została zaokrąglona do dwóch miejsc po przecinku. Następnie, umieszczono ją pod wykresem, gdzie w dwóch sąsiednich kolumnach zawarto wynikającą z wartości współczynnika siłę relacji oraz jej znak. Stopień powiązania zmiennych określono na podstawie wartości zawartych w tabeli 3.fragm

Tabela 3. Określenie siły związków korelacyjnych [15]

r	Siła związku
0	brak
0 – 0,1	nikła
0,1 – 0,3	słaba
0,3-0,5	przeciętna
0,5 – 0,7	wysoka
0,7 – 0,9	bardzo wysoka
0,9 - 1	prawie pełna

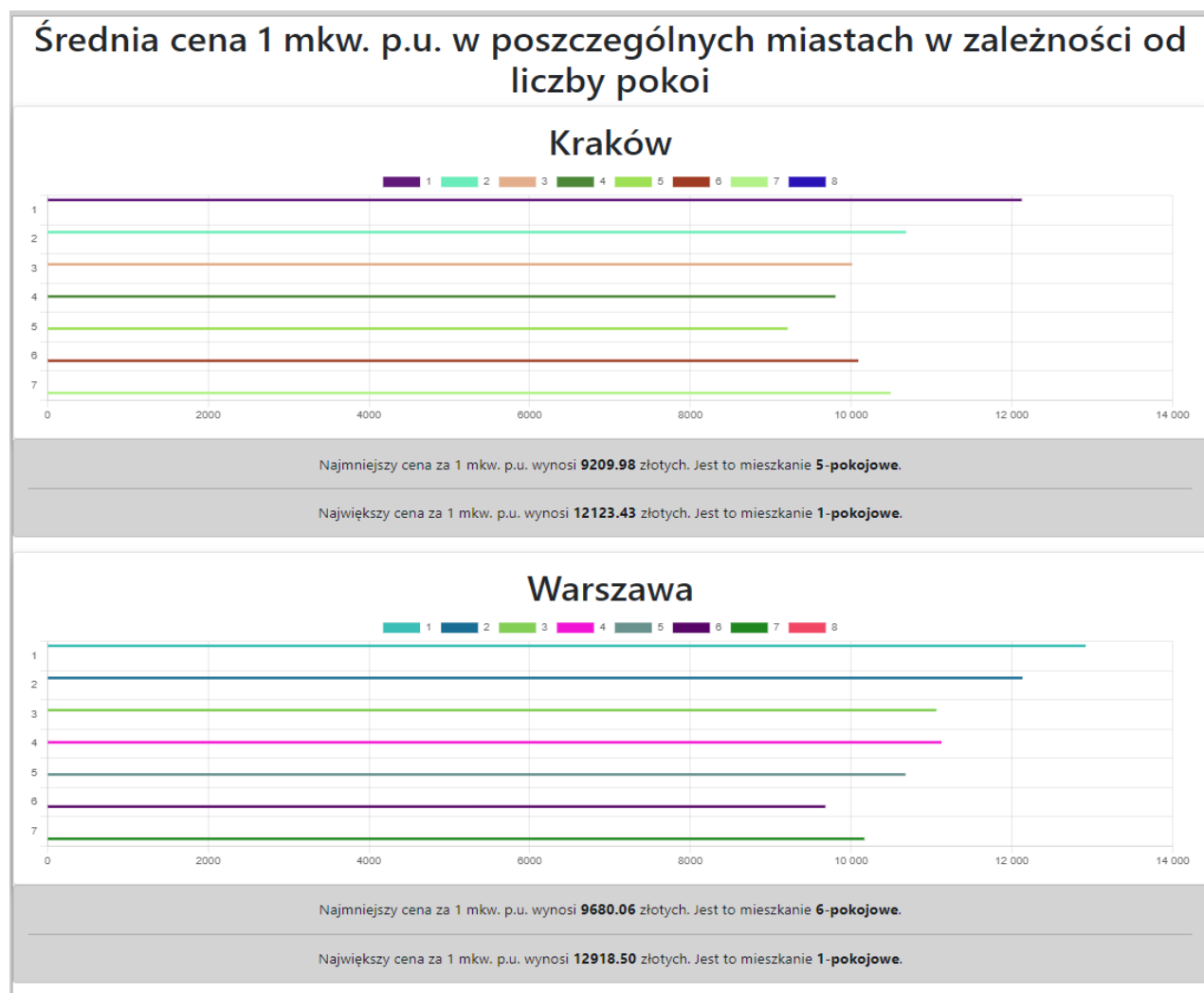
Znak współczynnika określa kierunek korelacji, jeżeli jest on ujemny oznacza to że wraz z spadkiem wartości jednej zmiennej, następuje wzrost wartości drugiej (i odwrotnie). W przypadku znaku dodatniego wartości obu zmiennych równocześnie rosną albo maleją. Po wskazaniu kursorem myszy na jedną z zawartych w tabeli wartości, pojawia się krótki komentarz objaśniający jej znacznie, który przedstawiono na rysunku 27.

Wartość współczynnika Pearsona		Powiązanie zmiennych	Znak
0.71		bardzo wysokie	Wartość współczynnika dodatnia - jeśli wartość jednej zmiennej rośnie, to drugiej również (i odwrotnie)
Wzór prostej regresji liniowej	Współczynnik determinacji	Współczynnik k	
y = 5777.99x + 138769.78	0.5	0.5	

Zwiększenie powierzchni użytkowej o 1 metr kwadratowy powoduje wzrost ceny nieruchomości o **5777.99** złotych.

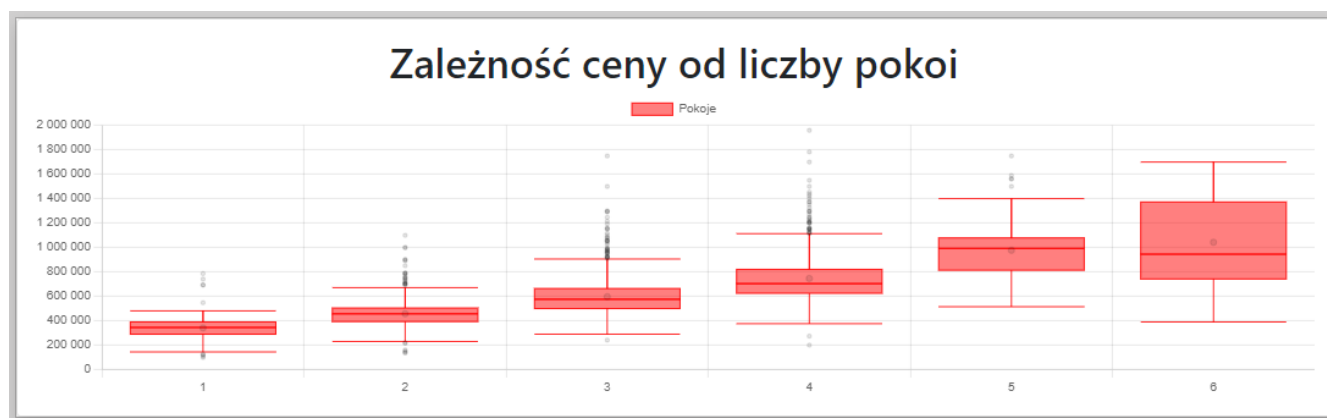
Rysunek 27. Fragment znajdującej się pod wykresami analizy korelacji zmiennych

Na stronie umieszczono także wykresy, przedstawiające cenę za 1 metr kwadratowy powierzchni mieszkania w poszczególnych miastach w zależności od liczby pokoi (rysunek 28). Analiza poniższych wykresów pokazuje, że najwięcej za 1 metr kwadratowy powierzchni przyjdzie zapłacić przy kupnie mieszkania jedno- i dwupokojowego, najmniej w przypadku mieszkań pięcio- lub sześciopokojowych.

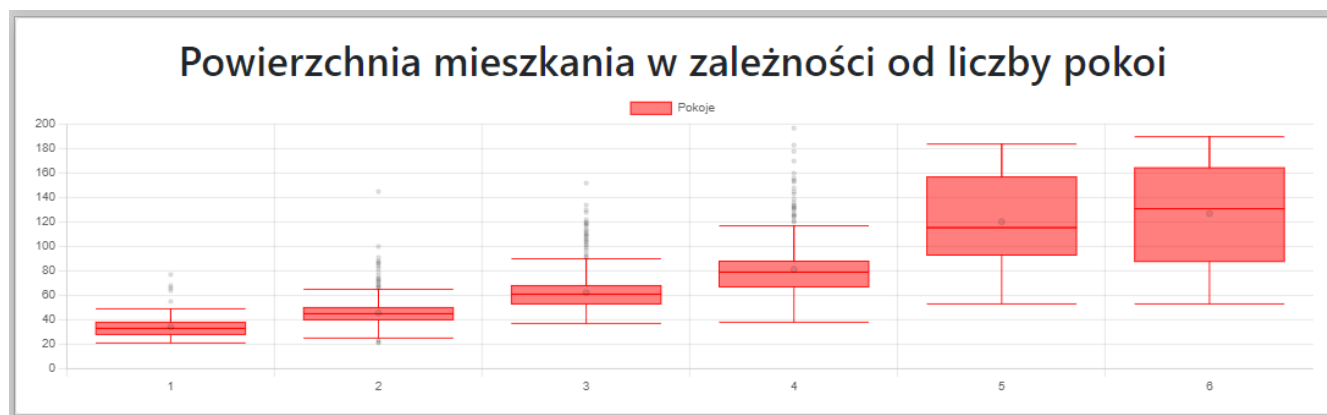


Rysunek 28. Wykres przedstawiający powierzchnię mieszkania w zależności od liczby pokoi

Przy użyciu wykresów pudełkowych, które ukazano na rysunkach 29 i 30, przedstawione zostały zależności między ceną lub powierzchnią mieszkania a liczbą pokoi. Pionowe linie znajdujące się na zewnątrz pudełka przedstawiają zakres ceny bądź powierzchni dla danej liczby pokoi. Dolny koniec linii wskazuje wartość minimalną, a górny maksymalną. Linia pozioma umieszczona wewnątrz pudełka przedstawia natomiast medianę. Analizując kształty wykresów można wyciągnąć wniosek, że wartości ceny mieszkania są bardziej do siebie zbliżone przy mniejszej liczbie pokoi, to samo zachodzi dla powierzchni.



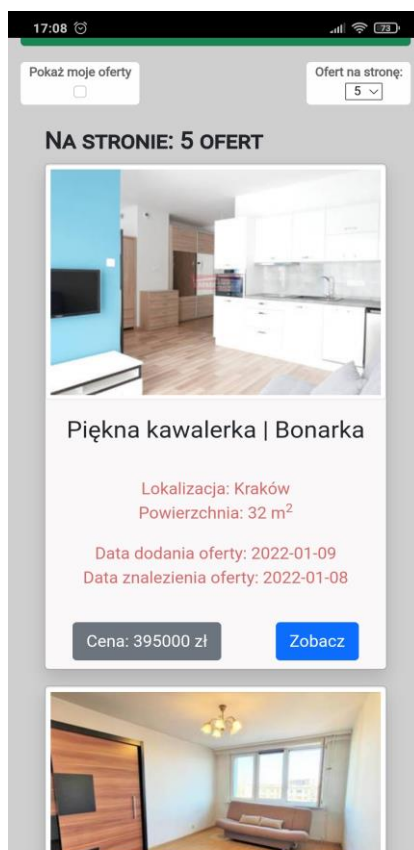
Rysunek 29. Wykres przedstawiający cenę w zależności od liczby pokoi



Rysunek 30. Wykres przedstawiający powierzchnię mieszkania w zależności od liczby pokoi

7.5. Test interfejsu

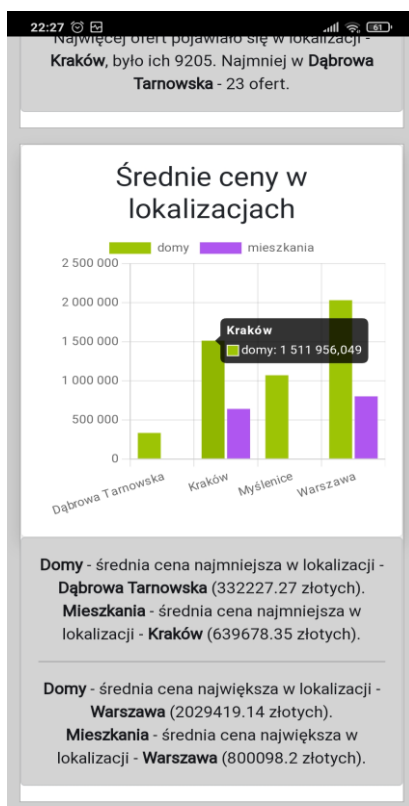
Jednym z wymagań, jakie postawiono przed wizualną częścią tworzonego rozwiązania, było automatyczne dopasowanie się układu i elementów strony do rozmiaru urządzenia, na którym jest wyświetlana. Weryfikację responsywności interfejsu graficznego, przeprowadzono na urządzeniu mobilnym o przekątnej ekranu 6.47" cala. Na rysunkach 31, 32, 33, 34 przedstawiono rezultaty testu.



Rysunek 31. Widok karty oferty na urządzeniu mobilnym



Rysunek 32. Widok panelu logowania na urządzeniu mobilnym



Rysunek 33. Widok jednego z wykresów na urządzeniu mobilnym

16:04

[Home](#) [Statystyki](#) [Rejestruj](#) [Zaloguj](#)

szukaj frazy... [Szukaj](#)

Wybierz z listy

Cena od: do:

Pow. od: do:

Data od: do:

Sortuj po: cenie malejąco

Typ oferty: wybierz

Lokalizacja:

Typ nieruchomości: domy

[Pokaż](#)

Zaloguj się, aby zobaczyć swoje oferty!

Ofert na stronie: 5

NA STRONIE: 5 OFERT

Rysunek 34. Widok strony głównej wraz z formularzem na urządzeniu mobilnym

8. Podsumowanie

Celem pracy było wykonanie analizy statystycznej ofert sprzedaży nieruchomości przy wykorzystaniu techniki web scraping. Do zrealizowania celu, stworzono niezbędne oprogramowanie w postaci serwera, która zbiera informacje z ofert umieszczonych w serwisie ogłoszeniowych, oraz dokonuje ich zapisu w bazie danych. Przygotowano także interfejs użytkownika, za pomocą którego można skonfigurować funkcje pobierające dane, jak również wyświetlić wynik ich działania w formie listy lub wykresów, które przedstawiają statystyczną analizę zebranych ogłoszeń.

Oprogramowanie zostało zrealizowane przy użyciu języka Javascript, bibliotek Node.js oraz Playwright który użyto przy tworzeniu narzędzia pobierające dane. Do implementacji wizualnej części projektu wykorzystano React, HTML, CSS i bibliotekę Chart.js, w celu stworzenia wykresów.

Otrzymaną aplikację wyróżnia wśród innych podobnych rozwiązań możliwość analizy zebranych danych. Nie wszystkie narzędzia były dostosowane również do pobierania informacji z większej ilości stron wyników wyszukiwania niż strona pierwsza, powiadomienia o znalezieniu oferty wyróżniającej się ceną czy też możliwością filtracji i sortowania zgromadzonych danych. Wszystkie analizy statystyczne w zaprezentowanym rozwiązaniu jak i wizualizacja wyników zostały zaimplementowane przez autora.

Rezultat przeprowadzonych prac pozwala na dalszy rozwój projektu, którego kolejnymi etapami mogłoby być zapisywanie w bazie informacji zawartych w pełnej treści ogłoszenia i ich analiza, oraz dostosowanie programu do pobierania danych z większej ilości stron. W efekcie uzyskano by możliwość tworzenia bardziej zaawansowanych analiz.

9. Bibliografia

- [1] Raport o sytuacji na rynkach nieruchomości mieszkaniowych i komercyjnych w Polsce w 2020, dostęp 2021-10-21, https://www.nbp.pl/publikacje/rynek_nieruchomosci/raport_2020.pdf
- [2] Ustawa z dnia 27 lipca 2001 r. o ochronie baz danych, Dziennik Ustaw 2001, nr 128 poz. 1402.
- [3] Tworzenie pliku robots.txt, dostęp 2021-10-24, <https://developers.google.com/search/docs/advanced/robots/create-robots-txt>
- [4] Jacek Maślankowski, Pozyskiwanie i analiza danych na temat ofert pracy z wykorzystaniem big data, Wiadomości statystyczne 64, 2019, str. 60-74.
- [5] Dokumentacja rozszerzenia Web Scraper – Free Web Scraping, dostęp 2021-10-29, <https://webscraper.io/documentation>

- [6] Strona główna portalu lzx, dostęp 2021-10-29, <https://www.lzx.pl>
- [7] Strona główna programu ParseHub, dostęp 2021-11-04, <https://www.parsehub.com>
- [8] JavaScript – przewodniki, dostęp 2021-11-7, <https://developer.mozilla.org/pl/docs/Web/JavaScript>
- [9] Dokumentacja biblioteki Bootstrap, dostęp 2021-11-7, <https://getbootstrap.com/docs/5.1/getting-started/introduction/>
- [10] Dokumentacja biblioteki React, dostęp 2021-11-8, <https://pl.reactjs.org/docs/getting-started.html>
- [11] What do tech giants, like Netflix and PayPal, have in common? Yes, it is JavaScript, dostęp 2021-11-10, <https://codescrum.medium.com/what-do-tech-giants-like-netflix-and-paypal-have-in-common-yes-it-is-javascript-c39cc0dd8387>
- [12] Dokumentacja biblioteki Playwright, dostęp 2021-11-12, <https://playwright.dev/docs/intro>
- [13] Paweł Miech, How does a headless browser help with web scraping and data extraction?, dostęp 2021-12-08, <https://www.zyte.com/blog/how-does-a-headless-browser-help-with-web-scraping-and-data-extraction/>
- [14] Martyna Szczygiał, Korelacja Pearsona, dostęp 2021-12-09, <https://cyrkiel.info/statystyka/korelacja-pearsona/>
- [15] Aleksander Zaigrajew, Korelacja. Współczynniki korelacji, <https://www-users.mat.umk.pl/~alzaig/kor.pdf>