

RWD JAKO NARZĘDZIE OPTIMALIZACJI STRON INTERNETOWYCH

Halina Ochim¹, Beata Pańczyk²

¹Asseco Business Solutions S.A, Lublin, ²Politechnika Lubelska, Wydział Elektrotechniki i Informatyki, Instytut Informatyki

Streszczenie. Na podstawie przeprowadzonej analizy aspektów projektowania i tworzenia responsywnych stron internetowych, niniejszy artykuł poświęcono zagadnieniu optymalizacji portalu internetowego z wykorzystaniem RWD (ang. Responsive Web Design). W artykule przedstawiono identyfikację czynników wpływających na wydajność projektów responsywnych oraz wyniki pomiaru wydajności testowego portalu przed i po optymalizacji.

Słowa kluczowe: strony responsywne, RWD, aplikacje internetowe

WEB PAGES OPTIMIZING USING RWD

Abstract. Based on the analysis of aspects of designing and creating responsive web pages, this article presents a web portal optimizing using RWD (Responsive Web Design). The paper presents the identification of factors affecting the responsive projects' performance and performance measurement of the portal before and after optimization.

Keywords: responsive web design, RWD, web applications

Wstęp

Postęp technologiczny ostatnich kilku lat sprawił, że na rynku pojawiła się cała gama nowych urządzeń mobilnych, przystosowanych do przeglądania stron internetowych. O ile nie ma problemu z technicznym wyświetleniem strony na ekranie każdego urządzenia, to jednak pojawia się problem z wykorzystaniem funkcjonalności strony na urządzeniach o małych wymiarach ekranu. Wynika to z faktu, iż większość stron była projektowana dla stałej szerokości okna przeglądarki, przystosowanej dla monitorów o dużej rozdzielczości. Wyświetlanie takich stron na ekranach urządzeń mobilnych wywołuje efekt miniaturyzacji elementów i konieczność ich powiększania przez użytkownika. Problem ten rozwiązuje responsywne podejście do projektowania stron internetowych [1, 3, 7, 8]. Ma ono na celu zapewnienie użytkownikowi jak najlepszego odbioru strony, niezależnie od rozmiaru używanej przez niego przeglądarki, ekranu czy typu urządzenia. Użytkownik przeglądający stronę responsywną na smartfonie, tablecie czy dużym monitorze powinien mieć łatwy dostęp do wszystkich funkcjonalności strony oraz zamieszczonych na niej odnośników. Prezentowane teksty i grafiki powinny być czytelne, w pełni widoczne oraz dopasowane do aktualnych rozmiarów okna przeglądarki. W projekcie responsywnym rozwijana jest tylko jedna wersja strony HTML, której układ dopasowuje się do aktualnych rozmiarów okna przeglądarki użytkownika, dzięki zastosowaniu odpowiednich arkuszy CSS [4, 5].

W aspekcie responsywnym podstawą stało się zaprojektowanie w pełni funkcjonalnej strony w pierwszej kolejności na urządzenia mobilne (ang. Mobile-first), a dopiero później tworzenie wersji desktopowej strony. Takie podejście wymaga już na początku projektu rozwiązania kwestii dotyczących niezbędnych funkcjonalności oraz zawartości strony.

1. Responsywny portal

Budowa w pełni funkcjonalnej, responsywnej strony internetowej od zera jest bardzo trudnym i czasochłonnym zadaniem. Projekty responsywne wymagają rozwiązania wielu technicznych kwestii, w szczególności ustalenia odpowiedniej liczby kolumn, obliczenia wartości procentowej dla każdej kolumny, określenia odpowiednich punktów granicznych dla poszczególnych typów urządzeń, zadbanie o poprawność wyświetlania danego elementu w różnych przeglądarkach itp. Dobrym rozwiązaniem jest skorzystanie z funkcjonalności oferowanych przez responsywne platformy programistyczne [3], ukierunkowane na interfejs strony oraz interakcję z użytkownikiem. Wyróżniane są dwie kategorie takich platform responsywnych – systemy płynnych siatek oraz bardziej

rozwinęte platformy oferujące dodatkowo predefiniowane arkusze stylów CSS. Arkusze te udostępniają stylizację tekstów oraz podstawowych elementów interfejsu użytkownika. Przykładem takiego rozwiązania jest Bootstrap [9], którego wersja 3.1 została wykorzystana do przygotowania optymalnego interfejsu dla analizowanego portalu [6]. Pierwsza wersja portalu fikcyjnej firmy faktoringowej została stworzona na potrzeby projektu inżynierskiego. W niniejszym artykule przedstawiono zoptymalizowaną, pod kątem responsywności, nową wersję portalu. Funkcjonalność systemu nie uległa zmianie. Wygląd fragmentu strony głównej przed i po optymalizacji prezentują rysunki 1 i 2.



Rys. 1. Wygląd strony głównej przed optymalizacją portalu



Rys. 2. Wygląd elementu z nawigacją po optymalizacji na komputerze i smartfonie

2. Poprawa wydajności responsywnych stron internetowych

Ważnym zagadnieniem związanym bezpośrednio z optymalizacją stron responsywnych jest poprawa wydajności i szybkości ich pobierania.

2.1. Minimalizacja liczby żądań HTTP

Za pomocą protokołu HTTP przesyłane są żądania udostępnienia dokumentów WWW, informacje dotyczące wybranego przez użytkownika odnośnika na stronie czy też dane z pól formularzy. Pobranie każdego dokumentu podczas generowania strony (np. pliku graficznego, arkusza styli, pliku czcionek, pliku z kodem JavaScript) wymaga każdorazowo przesłania odrębnego żądania HTTP. Każde takie zapytanie trwa określony czas (w zależności od szybkości łącza od ułamka sekundy do kilku sekund) i wpływa na czas ładowania całej strony [7]. Podstawą staje się zatem minimalizacja liczby takich żądań.

2.2. Konsolidacja i minimalizacja zasobów

Istotnym czynnikiem poprawiającym wydajność witryny jest minimalizacja liczby żądań HTTP poprzez konsolidację i łączenie pobieranych w ramach strony plików (np. arkuszy CSS, skryptów JavaScript) w jeden plik. W przypadku używania różnych plików danego rodzaju, na etapie tworzenia witryny, proces scalania plików można wykonać na końcu, za pomocą dedykowanych narzędzi do łączenia plików [2]. Dobrym rozwiązaniem jest również minimalizacja kodu w dołączanych do strony plikach poprzez eliminację niepotrzebnych znaków (nadmiarowych spacji, wcięć w akapitach, znaków końca linii). Zastosowanie takiego rozwiązania dla kodu z przykładu 1, przedstawia przykład 2.

Przykład 1. Kod przed minimalizacją znaków

```
p { font-family: Georgia, serif;
    font-size: 1.1em;
}
li { font-family: Helvetica, sans-serif;
    font-size: 1.2em;
}
```

Przykład 2. Kod po minimalizacji liczby znaków

```
p{font-family:Georgia,serif;font-size:1.1em;}li{fontfamily:
Helvetica,sans-serif;font-size:1.2em;}
```

2.3. Technika Image Sprites

Techniką optymalizacji plików graficznych jest Image Sprites. Polega ona na redukcji liczby używanych zasobów graficznych na stronie (a tym samym liczby plików koniecznych do pobrania przy wyświetlaniu strony), poprzez ich konsolidację w jeden duży plik graficzny. Największe zastosowanie ma to dla obrazów wzajemnie ze sobą powiązanych lub wyświetlanych na stronie blisko siebie. Może być to zestaw ikon używany w listach lub przy nawigacji. Wybór odpowiedniego fragmentu obrazu do wyświetlenia następuje z wykorzystaniem informacji o jego rozmiarach oraz marginesach odcinających pozostałą część obrazu [11]. Przykład zastosowania tej techniki poprzez wybór pierwszej ikony ze skonsolidowanego obrazu (rys. 3) prezentuje przykład 3 [10].



Rys 3. Przykład skonsolidowanego obrazu w ramach techniki Image Sprites [10]

Przykład 3. Zastosowanie techniki Image Sprites [22]

```
#home { width: 46px; height: 44px;
        background: url(img_navsprites.gif) 0 0; }
```

2.4. Warunkowe wczytywanie zawartości

Założeniem projektowania responsywnego jest udostępnienie użytkownikom wszystkich funkcjonalności strony, niezależnie od urządzenia, z którego korzystają. Nie oznacza to jednak, że zawartość strony oraz poszczególne opcje muszą być obecne dokładnie w tym samym miejscu i na tej samej stronie. Przykładem takiej sytuacji jest prezentacja panelu bocznego, z dodatkową zawartością przy dużych szerokościach okna przeglądarki, który nie będzie wyświetlany przy przeglądaniu strony na ekranach telefonów czy tabletów. Użytkownik ma jednak dostęp do treści panelu w ramach innego rozwiązania. Taka technika jest nazywana warunkowym wczytywaniem zawartości i jest związana z użyciem zapytania JavaScript o rozmiary okna przeglądarki [7] (Przykład 4).

Przykład 4. Zapytanie JavaScript o szerokość okna przeglądarki

```
<script>
    if (document.documentElement.clientWidth > 640) {
        //skrypt z funkcjonalnością
    } </script>
```

W przypadku braku obsługi JavaScript przez przeglądarkę, kod skryptu zostanie pominięty, jednak z perspektywy użytkownika nie będzie to zauważalne - zostanie zaprezentowana domyślna zawartość. Istotnym aspektem tego rozwiązania jest fakt, iż kod JavaScript jest przetworzony tylko raz, przy pierwszym wyświetleniu strony. Modyfikacja rozmiarów okna przeglądarki lub przejście z układu pionowego na poziomy nie powoduje zmian w obsłudze kodu skryptu (inaczej niż ma to miejsce w przypadku zapytań o media w arkuszach CSS) [7].

2.5. Lokalizacja zasobów strony na różnych serwerach

Kolejną techniką poprawiającą wydajność strony WWW jest lokalizacja zasobów, z których ona korzysta, na różnych serwerach (ang. Domain Sharding). Ma to na celu minimalizację czasu potrzebnego do wyświetlenia strony w przeglądarce. W sytuacji, gdy strona internetowa zawiera wiele odwołań do plików zewnętrznych, przeglądarka generuje żądania do serwera WWW celem ich pobrania. W przypadku, gdy zapytania odnoszą się do pojedynczego serwera, są one generowane kolejno tzn. dane zapytanie zostanie wygenerowane po zakończeniu poprzedniego. Może to istotnie wydłużyć czas wyświetlenia strony. Rozwiązaniem tego problem jest technika Domain Sharding, polegająca na umieszczeniu zewnętrznych plików (np. grafik, mediów) na różnych serwerach. Pozwoli to przeglądarce na równoległe wygenerowanie zapytań do różnych serwerów, celem pobrania zawartości plików. Jeśli takie rozwiązanie jest możliwe to przyspiesza znacznie proces wygenerowania strony.

2.6. Sieć dostarczania treści

Do optymalizacji czasu ładowania strony internetowej może być wykorzystana sieć dostarczania treści (ang. Content Delivery Network - CDN). Rozwiązanie to, podczas generowania strony, bierze pod uwagę czynnik lokalizacji geograficznej użytkownika oraz serwerów WWW. Wybiera najbliższy użytkownikowi serwer do pobrania wskazanych elementów strony. Dodatkowym atutem takiego rozwiązania jest zabezpieczenie się przed sytuacją niedostępności jednego z serwerów w grupie (np. w wyniku awarii czy problemów z łącznością). W takiej sytuacji żądania przeglądarek są obsługiwane przez pozostałe serwery w sieci [7].

3. Wybrane elementy optymalizacji witryny

Pierwsza wersja testowanego portalu została zaprojektowana dla minimalnej szerokości okna przeglądarki 960 px (rys. 1). Powodowało to efekt miniaturyzacji elementów na tabletach i smartfonach. Nowa wersja witryny wykorzystuje klasy i skrypty platformy programistycznej Bootstrap oraz korzysta z przedstawionych w rozdziale 2 metod optymalizacji.

3.1. Baner portalu

Na każdej stronie portalu pojawia się baner, który prezentuje zmieniające się obrazy z podpisami (rys. 1, 2). Pierwsze rozwiązanie bazowało na plikach graficznych o stałych rozmiarach 960 px na 200 px. Napisy były naniesione bezpośrednio na pliki graficzne a za przeglądanie obrazów odpowiadała funkcja JS. Zmiana rozmiarów okna nie powodowała zmiany wielkości wyświetlanego banera. Implementację tego rozwiązania prezentuje przykład 5.

Przykład 5. Fragment kodu HTML i JS dla banera przed optymalizacją

```
<!-- baner -->
<div id="banner">
<div class="slajdy">
<a href="index.php"> </a>
<a href="index.php"> </a>
...
<a href="index.php"> </a>
</div><!-- slajdy w banerze -->
</div> <!-- baner -->

/* funkcja js dla slajdów w banerze */
$(document).ready(function(){
$($('.slajdy').innerfade({ speed: 'slow', timeout: 10000,
type: 'random', containerheight: '200px' });});
```

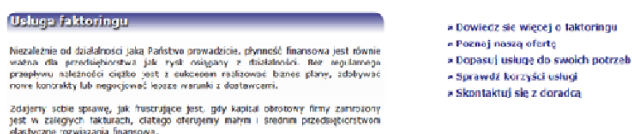
W ramach modyfikacji, dotychczasowe rozwiązanie zastąpiono komponentem typu karuzela. Wyświetlenie tego komponentu dostosowane jest do rozmiarów okna użytkownika w zakresie prezentowanego obszaru obrazu i czytelności podpisu. Dostępna jest również funkcjonalność przewijania obrazów oraz możliwość wskazania konkretnego obrazu do odświetlenia (czego nie zapewniało dotychczasowe rozwiązanie). Implementację nowego rozwiązania prezentuje przykład 6. Funkcja JavaScript do obsługi komponentu jest natywnie zaimplementowana i dostarczana razem z platformą Bootstrap.

Przykład 6. Fragment kodu HTML i JS dla banera po optymalizacji

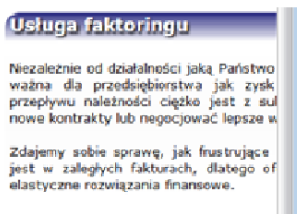
```
<!-- Karuzela -->
<header id="myCarousel" class="carousel slide">
<ol class="carousel-indicators">
<li data-target="#myCarousel" data-slide-to="0" class="active"></li>
<li data-target="#myCarousel" data-slide-to="1"></li>
...
<li data-target="#myCarousel" data-slide-to="7"></li>
</ol>
<!-- Kontener slajdów -->
<div class="carousel-inner">
<div class="item active">
<div class="fill" style="background-image:url('images/baner/b1.jpg');">
</div>
<div class="carousel-caption"> <h2>Profesjonalizm</h2></div>
</div>
...
<div class="item">
<div class="fill" style="background-image:url('images/baner/b7.jpg');">
</div>
<div class="carousel-caption"><h2>Rozliczenia online</h2></div>
</div></div>
<!-- Przewijanie elementów w banerze -->
<a class="left carousel-control" href="#myCarousel" data-
slide="prev"><span class="icon-prev"></span></a>
<a class="right carousel-control" href="#myCarousel" data-
slide="next">
<span class="icon-next"></span></a>
```

3.2. Strona główna

Zawartość strony głównej portalu zawiera opis usługi faktoringu oraz linki do najważniejszych podstron witryny. Pierwotne dopasowanie do stałej szerokości 960 px powoduje, iż przy mniejszej szerokości okna przeglądarki treść strony nie jest w pełni widoczna a odnośniki pozostają ukryte (rys. 4, 5). W ramach optymalizacji wyświetlania strony głównej wykorzystano podział przestrzeni strony na kolumny i wiersze korzystając z siatki platformy Bootstrap (przykład 7, rys. 6). Zawartość strony podzielona została na dwa wiersze – pierwszy dedykowany na nagłówek strony, drugi na opis usługi oraz linki do poszczególnych podstron. Drugi wiersz z kolei podzielono na dwie kolumny o zróżnicowanej szerokości, dedykowane odpowiednio na opis usługi oraz odnośniki do podstron. Jednocześnie w ramach kolumny z opisem usługi zaimplementowano dodatkowy wiersz podzielony na dwie kolumny ze skróconym opisem rodzajów faktoringu oraz odnośnikami do dedykowanych podstron w postaci paneli. Układ elementów na stronie głównej przed i po optymalizacji prezentują rysunki 4, 5 i 6.



Rys. 4. Wygląd fragmentu strony głównej portalu przed modyfikacją dla standardowej szerokości okna przeglądarki



Rys. 5. Wygląd fragmentu strony głównej portalu przed modyfikacją dla małej szerokości okna przeglądarki



Rys. 6. Wygląd fragmentu strony głównej portalu po modyfikacji dla małej szerokości okna przeglądarki

Przykład 7. Fragment kodu HTML strony głównej po optymalizacji

```

<div class="row">
<div class="col-lg-12">
<h1 class="page-header"> Usługa faktoringu </h1></div>
</div>
<div class="row">
<div class="col-md-8">
<h3><small>Niezależnie od działalności ...</small></h3>
<div class="row">
<div class="col-lg-12">
<h2 class="page-header"><small>Rodzaje Faktoringu</small></h2>
</div>
<div class="col-md-6">
<div class="panel panel-default text-center">
<div class="panel-heading">
<h4>Faktoring klasyczny</h4>
</div>
<div class="panel-body">
<p>Usługa finansowa polegająca na ...</p>
<a href="faktoring.html" class="btn btn-primary">Dowiedz się
więcej</a> </div>
</div></div>
<div class="col-md-6">

```

```

<div class="panel panel-default text-center">
<div class="panel-heading"><h4>Faktoring odwrotny</h4></div>
<div class="panel-body">
<p>Usługa finansowa polegająca na ...</p>
<a href="faktoring.html" class="btn btn-primary">Dowiedz się
więcej</a>
</div>
</div></div></div></div>
<div class="col-md-4">
<a class="btn btn-lg btn-default btn-block" href="uslugi.html"> Poznaj
naszą ofertę</a>
...
</div>
</div>

```

3.3. Podstrona z prezentacją oferty firmy

Prezentacja oferowanych przez firmę usług na portalu cechowała się koniecznością wyboru rodzaju oferowanej usługi z tabeli. Wybór taki przekierowywał użytkownika na dedykowaną stronę z szerszym opisem. Z uwagi na określenie stałej szerokości dla obu tych elementów, zmiana rozmiarów okna przeglądarki nie powodowała zmiany ich wymiarów. Modyfikacja portalu w zakresie oferty firmy dotyczyła dwóch aspektów – nadano temu elementowi responsywność oraz wyeliminowano konieczność przejścia do innej strony przy przeglądaniu oferowanych usług. W nowym rozwiązaniu zastosowano komponent typu zakładka (ang. *tab*), który umożliwia podgląd opisu po zaznaczeniu wybranej zakładki. W zależności od rozmiarów okna przeglądarki komponent jest prezentowany: w formie poziomego wiersza podzielonego na bloki zakładek (dla standardowych wielkości okna przeglądarki) lub w postaci pionowej listy z elementami wyboru zakładek w formie przycisków (dla małych szerokości okna przeglądarki – rys. 7).



Rys. 7. Wygląd fragmentu strony z ofertą firmy przed i po modyfikacji dla małej szerokości okna przeglądarki

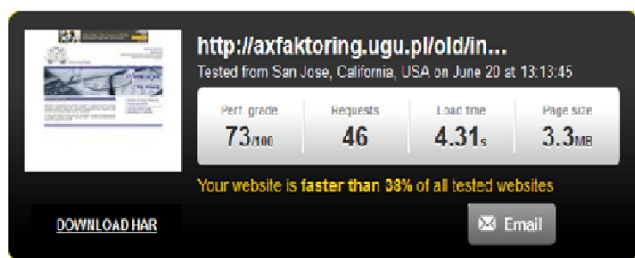
4. Pomiar i testowanie wydajności strony

Kluczowym elementem procesu optymalizacji projektów responsywnych jest testowanie stron oraz analiza zmian w ich wydajności realizowana za pomocą dedykowanych narzędzi. W celu przeprowadzenia takiej analizy została wybrana strona główna portalu przed i po optymalizacji.

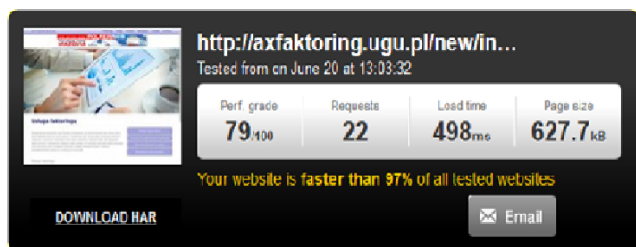
4.1. Pingdom Website Speed Test

Jednym z narzędzi do testowania wydajności stron internetowych jest Pingdom Website Speed Test (dostępny na

stronie <http://tools.pingdom.com/fpt/>). Podstawowe informacje dotyczące analizowanej strony, ogólna ocena jej wydajności, liczba żądań HTTP, czas ładowania oraz rozmiar strony przedstawiają rysunki 8 i 9. Na ich podstawie widać, iż w istotny sposób zmniejszył się czas ładowania strony (z 4.31s do niecałej 0.49s). Zmniejszeniu uległa też liczba żądań HTTP oraz rozmiar elementów strony. Osiągnięto to poprzez zmniejszenie liczby obrazów, ładowanych do banera portalu oraz mniejszą liczbę plików Java Script. W przypadku strony po optymalizacji pobierany jest tylko jeden obraz do banera. W przypadku obu stron przetwarzane są żądania związane z pobraniem arkuszy CSS oraz skryptów JavaScript.



Rys. 7. Ocena strony głównej przed optymalizacją wg Pingdom Website Speed Test



Rys. 8. Ocena strony głównej po optymalizacji wg Pingdom Website Speed Test

4.2. WebPageTest

Kolejne popularne narzędzie analizy wydajności stron WWW to WebPageTest (<http://www.webpagetest.org/>). Jego dodatkowym atutem jest możliwość dokonania pomiarów przy wykorzystaniu różnych lokalizacji geograficznych oraz wersji przeglądarek internetowych.

Do testowania strony głównej portalu została wybrana lokalizacja z możliwością przeprowadzenia testów na urządzeniu mobilnym Nexus 7 z systemem operacyjnym Android 4.4 i przeglądarką Chrome. Podstawowe wskaźniki oceny wydajności strony wskazały istotne zmniejszenie czasu ładowania strony głównej portalu po optymalizacji (z 10.5s do 3.4s). Znacznie zmniejszyła się też ilość połączeń HTTP oraz rozmiar elementów strony. Porównanie analizowanych parametrów prezentuje rysunek 9 i 10.

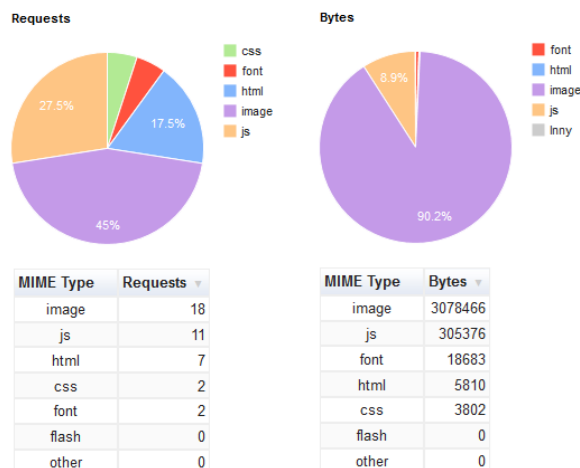
Narzędzie to umożliwia również analizę poszczególnych komponentów strony, ze wskazaniem ich rozmiaru oraz koniecznej liczby połączeń. Rysunek 11 prezentuje analizę typu komponentów strony głównej portalu przed optymalizacją, natomiast rysunek 12 przedstawia te parametry dla strony głównej portalu po optymalizacji. Porównując obie analizy można zauważyć, iż zmniejszeniu uległa liczba żądanych plików (w szczególności plików graficznych, skryptów JavaScript oraz arkuszy CSS), jak również ich rozmiar.

| | Load Time | First Byte | Start Render | Speed Index | Document Complete | | | Fully Loaded | | | |
|-------------|-----------|------------|--------------|-------------|-------------------|----------|----------|--------------|----------|----------|------------|
| | | | | | Time | Requests | Bytes In | Time | Requests | Bytes In | Cost |
| First View | 10.568s | 0.384s | 1.792s | 6543 | 10.568s | 37 | 3,332 KB | 11.875s | 41 | 3,333 KB | \$\$\$\$\$ |
| Repeat View | 2.862s | 0.259s | 1.217s | 2666 | 2.862s | 14 | 77 KB | 4.221s | 18 | 78 KB | |

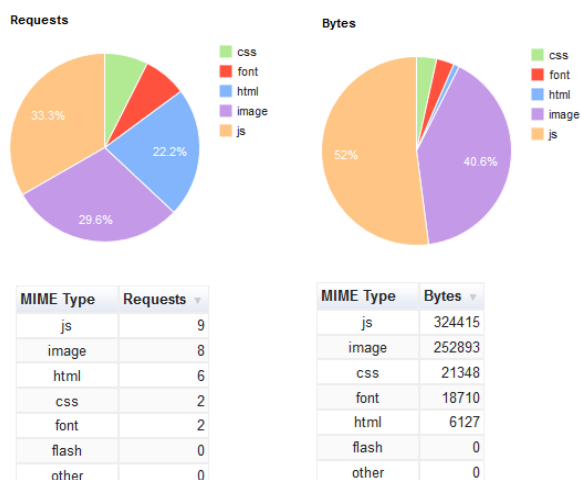
Rys. 9. Ocena strony głównej portalu przed optymalizacją wg WebPageTest

| | Load Time | First Byte | Start Render | Speed Index | Document Complete | | | Fully Loaded | | | |
|-------------|-----------|------------|--------------|-------------|-------------------|----------|----------|--------------|----------|----------|---------|
| | | | | | Time | Requests | Bytes In | Time | Requests | Bytes In | Cost |
| First View | 3.438s | 0.379s | 1.366s | 1904 | 3.438s | 24 | 609 KB | 4.878s | 28 | 614 KB | \$\$--- |
| Repeat View | 2.309s | 0.899s | 0.519s | 797 | 2.309s | 12 | 74 KB | 3.566s | 15 | 74 KB | |

Rys. 10. Ocena strony głównej portalu po optymalizacji wg WebPageTest



Rys. 11. Analiza elementów strony przed optymalizacją wg WebPageTest



Rys. 12. Analiza elementów strony po optymalizacji wg WebPageTest

4.3. MobiTest

MobiTest (<http://mobitest.akamai.com>) jest kolejnym narzędziem, które zostało wykorzystane do weryfikacji wydajności strony internetowej. Dodatkową zaletą jest tu możliwość symulacji, podczas ładowania strony, fizycznego urządzenia mobilnego. Wynikiem końcowym jest informacja o czasie wyświetlenia strony oraz jej rozmiarze. Podobnie jak w przypadku poprzednich narzędzi, przedstawiane są dane o połączeniach HTTP. Analiza wydajności strony głównej optymalizowanego portalu została przeprowadzona za pomocą MobiTest dla urządzenia mobilnego iPad3 z systemem operacyjnym iOS7. Wyniki pomiaru wydajności strony głównej wskazały czas generowania na poziomie 4,20 s przed optymalizacją oraz 3,81 s po optymalizacji. Rozmiar strony głównej przed optymalizacją wyniósł 3297,62 kb, natomiast po optymalizacji 574,07 kb [6].

5. Wnioski

Optymalizacja portalu została wykonana z użyciem platformy Twitter Bootstrap. Wcześniejszy projekt witryny stał się w pełni responsywny i łatwo modyfikowalny. Jego funkcjonalności zostały w pełni zachowane. Portal uzyskał spójny, estetyczny wygląd, intuicyjną nawigację oraz przejrzystość prezentowanych treści zarówno na ekranach urządzeń mobilnych, jak i na monitorach komputerów stacjonarnych. Wyniki pomiarów wydajności zmodyfikowanego portalu jednoznacznie świadczą na korzyść stosowania projektowania responsywnego. Ważna jest nie tylko kwestia poprawy użyteczności interfejsu użytkownika, ale także istotne zwiększenie szybkości pobierania responsywnych stron internetowych.

Literatura

- [1] Beaird J., George J.: Niezawodne zasady web designu. Projektowanie spektakularnych witryn internetowych, Helion, 2015.
- [2] Crespo G.: Responsive Web Design with jQuery, Helion 2014.
- [3] Firdaus T.: Responsive Web Design. Nowoczesne strony WWW na przykładach, Helion, 2014.
- [4] Frain B.: Responsive Web Design. Projektowanie elastycznych witryn w HTML5 i CSS3, Helion 2013.
- [5] Gaston P.: The book of CSS3: A Developer's Guide to the Future of Web Design., No Starch Press 2011.
- [6] Ochim H.: Optymalizacja stron internetowych pod kątem projektowania responsywnego, praca magisterska, Politechnika Lubelska, 2015.
- [7] Peterson C.: Responsywne strony WWW dla każdego, Helion, 2015.
- [8] Sharkie C., Fisher A.: Responsywne strony WWW. Technologia na start!, Helion, 2014.
- [9] <http://getbootstrap.com/getting-started/>, [1.11.2016].
- [10] http://www.w3schools.com/css/css_image_sprites.asp, [1.11.2016].

Mgr inż. Halina Ochim
e-mail: halina.ochim@gmail.com

Absolwentka jednolitych studiów magisterskich o kierunku Zarządzanie i marketing na Wydziale Ekonomicznym Uniwersytetu Marii Curie-Skłodowskiej w Lublinie oraz studiów inżynierskich (I stopnia) i magisterskich (II stopnia) o kierunku Informatyka na Wydziale Elektrotechniki i Informatyki Politechniki Lubelskiej. Menedżer Projektu w Asseco Business Solutions S.A. uczestniczący we wszystkich etapach procesu wdrożenia systemów faktoringowych. Aktywnie zaangażowana w rozwój nowych rozwiązań dla branży faktoringowej.



Dr Beata Pańczyk
e-mail: b.panczyk@pollub.pl

Ukończyła studia matematyczne na UMCS w Lublinie. W latach 1989–2011 pracownik naukowy (asystent, adiunkt) w Instytucie Informatyki Politechniki Lubelskiej. Tytuł doktora uzyskała w roku 1996 na Wydziale Elektrycznym PL. Temat rozprawy doktorskiej: Konstrukcja obrazu rozkładu właściwości fizycznych obiektu metodą Impedancyjnej Tomografii Komputerowej. Od roku 2011 pracuje na stanowisku starszego wykładowcy. Obszar zainteresowań dydaktycznych i naukowych to metody numeryczne i języki programowania.



otrzymano/received: 17.03.2016

przyjęto do druku/accepted: 30.10.2016