

**Państwowa Wyższa Szkoła Zawodowa**

**im. Jakuba z Paradyża**

**w Gorzowie Wielkopolskim**

Wydział techniczny

Kierunek studiów: informatyka

##### Kołbuc Andrzej

Nr albumu: 22009

**Praca dyplomowa inżynierska**

**Temat**

**Bootstrap. Resposywne strony internetowe na przykładzie portalu o drukarkach 3D**

Promotor:

Dr. Piotr Rafał Bubacz

Gorzów Wielkopolski, 2015

SPIS TREŚCI

[1. Wstęp 3](#_Toc451712456)

[1.1 Opis pracy 4](#_Toc451712457)

[1.2 Użyte technologie i narzędzia programistyczne 5](#_Toc451712458)

[2. Teoria 6](#_Toc451712459)

[2.1 Responsywność stron internetowych 7](#_Toc451712460)

[2.2 Porównanie frameworków do tworzenia responsywnych stron WWW 9](#_Toc451712461)

[2.3 System Bootstrap 10](#_Toc451712462)

[2.4 Wymagania i możliwości Bootstrapa 12](#_Toc451712463)

[2.4.1 Siatkowy układ Bootstrapa 13](#_Toc451712464)

[2.4.2 Komponenty Bootstrapa 16](#_Toc451712465)

[2.4.3 Rozszerzenia Bootstrapa 19](#_Toc451712466)

[2.4.4 Modyfikowanie i konfiguracja systemu 20](#_Toc451712467)

[2.4.5 Optymalizowanie kodu 21](#_Toc451712468)

[3. Część praktyczna 22](#_Toc451712469)

[3.1 Rozszerzenie funkcjonalności Bootstrapa 23](#_Toc451712470)

[3.2 Budowa struktury strony 24](#_Toc451712471)

[3.2.1 Obsługa starych przeglądarek internetowych 25](#_Toc451712472)

[3.3 Testowanie wersji mobilnej 26](#_Toc451712473)

[4. Zakończenie 27](#_Toc451712474)

[5. Bibliografia 28](#_Toc451712475)

# Wstęp

W dzisiejszych czasach strony internetowe wykorzystywane jako wizytówka firmy, sklep online, portale informacyjne czy aplikacje internetowe. Użytkownicy takich serwisów coraz częściej odwiedzają strony WWW za pomocą urządzeń mobilnych. Stawia to przed projektantami i programistami szereg wyzwań. Każdy użytkownik powinien mieć możliwość przeglądać strony Internetowe szybko i wygodnie. Problem ten rozwiązuje projektowanie stron według zasad RWD. Responsive Web Design oznacza sposób tworzenia komponentów witryny tak aby każda osoba odwiedzająca stronę mogła szybko i sprawnie odszukać potrzebnych informacji.

Rozdzielczości są zależne od producenta i wykorzystanych technologii. Mogą mieć za równo wartość 320px na 480px (np. dla telefonu Goclever quantum 4[[1]](#footnote-2)) jak również 1440 px na 2560px (dla telefonu Samsung Galaxy S7[[2]](#footnote-3)). Grafik jak i programista musi brać pod uwagę te różnice i dostosować witrynę pod wszelkie urządzenia. W tym celu można wykorzystać gotowe rozwiązania pomocne w utrzymaniu standardu RWD. Wiele lat doświadczenia różnych programistów pozwoliło na zbudowanie różnych systemów które dopasowane są pod wymagania i możliwości urządzenia użytkownika. W pracy opisane zostaną rożne szkielety języka CSS które określają standard wykonania storn internetowych. Dostosowanie się do tych wytycznych unikniemy wielu często powtarzanych błędów które utrudniają użytkownikom przeglądanie stron WWW. Jednocześnie systemy te są na tyle elastyczne, że pozwalają programiście wprowadzać własne udogodnienia i poprawki. Można zmienić układ siatki trzymającej treść strony, tak aby dopasować jej wygląd pod własne wymagania. W pracy przedstawione zostaną modyfikacje jednego z frameworków[[3]](#footnote-4) CSS, które usprawniają tworzenie stron dla smartfonów dostępnych na rynku. Opisany w pracy system Bootstrap 3 zostanie porównany z podobnymi konstrukcjami.

## 1.1 Opis pracy

Praca ma na celu przedstawić problem responsywnych stron internetowych i sposobu ich tworzenia wykorzystując gotowe narzędzia. Strony które są dopasowane do różnych mobilnych urządzeń są już standardem w projektowaniu użytecznych witryn. W pracy porównane zostaną frameworki języka CSS do tworzenia responsywnych stron internetowych. Porównane zostaną takie systemy jak Bootstrap 3, Foundation 6 oraz Skeleton 2. Praca poświęcona jest frameworkowi Bootstrap 3. Opisane będą wszystkie komponenty i funkcje systemu które pomocne są w tworzeniu strony. Pomimo określonych standardów, programista może do woli zmieniać wygląd wszystkich elementów pisząc własne style w języku CSS.

Różnorodność przeglądarek i urządzeń mobilnych wymusza na programiście tworzenie elastycznych stron, które będą dobrze wyglądać w każdym przypadku. Praca opisuje sposób tworzenia stron www według standardu systemu Bootstrap który jest oparty o licencje MIT. Jego otwarte źródło umożliwia swobodną modyfikacje kodu pod własne potrzeby a także wprowadzania poprawek które mogą być wdrożone w oficjalną wersję systemu. W pracy przedstawiony zostanie praktyczny projekt, który wykorzystuje framework Bootstrap 3 aby utrzymać stronę w standardzie RWD (tj. Responsive Web Design). Dodatkowo napisany zostanie kod CSS który zwiększa funkcjonalność systemu, który aktualnie nie daje wiele możliwości w tworzeniu wyglądu strony dla małych urządzeń mobilnych wciąż często używanych przez użytkowników. Na koniec opisany zostanie prosty sposób konfiguracji systemu poprzez formularz dostępny na stronie Bootstrapa, oraz możliwości optymalizacji kodu źródłowego aby był bardziej wydajny. Przedstawione zostaną dodatkowe skrypty języka Javascript które pozwolą obsłużyć starsze i nie wystarczająco rozwinięte przez producentów przeglądarki internetowe które słabo radzą sobie z nowymi komendami języka CSS3 i HTML5. W pracy opisane zostaną zmiany które wprowadzi w niedalekiej przyszłości nowa czwarta wersja Bootstrapa która będzie kompatybilna wstecz z aktualną wersją systemu.

## 1.2 Użyte technologie i narzędzia programistyczne

Praca przedstawia responsywną stronę internetową która wykorzystuje języki takie jak HTML5, CSS3, PHP5. Dodatkowo witryna wykorzystuje bazę danych MySQL. Do utrzymania standardu RWD, całość kodu oparta jest o system Bootstrap 3 z drobnymi modyfikacjami. Jako CMS wykorzystany został system Gleez oparty o framework Kohana który obsługuje wzorzec HMVC dający wiele możliwości rozbudowy.

Do zarządzania strukturą projektu wykorzystano oprogramowanie Netbeans. Do testowania użyto przeglądarek internetowych dostępnych na rynku takich jak: Chrome, Mozilla Firefox, Internet Explorer, Microsoft Edge, Safari. Dodatkowe rozszerzenia do przeglądarek umożliwiają symulacje urządzeń mobilnych i testowanie różnych rozdzielczości ekranu. Dodatkowe testy zostaną przeprowadzone na urządzeniach mobilnych takich jak Tablet Galaxy Tab 2 oraz Alcatel Pop C5. Testowanie na różnych urządzeniach i przeglądarkach ma na celu odnaleźć wszelkie błędy wyświetlania się strony WWW. Najmniejsze błędy mogą być uciążliwe dla użytkowników i spowoduje że strona nie będzie spełniać standardu RWD.

Do sprawdzenia poprawności wyświetlania się wersji mobilnej, dostępne są zewnętrzne skrypty. Google udostępnia oprogramowanie do testowania zgodności z urządzeniami mobilnymi. Poprawny wynik tego testu ma wpływ na wynik wyszukiwania w tej wyszukiwarce.

# Teoria

W części teoretycznej pracy, przedstawione zostaną systemy, które pomagają w utrzymaniu standardu tworzenia responsywnych stron internetowych. Porównane zostaną trzy popularne frameworki udostępniane na darmowych licencjach które pozwalają na modyfikacje kodu źródłowego i implementacje w różnych systemach zarządzania treścią.

Bardziej szczegółowo zostanie opisany system Bootstrap 3. Przedstawione zostaną wszystkie możliwości i komponenty które pomocne są w tworzeniu przyjaznej dla użytkownika strony internetowej. Praca obejmuje problem optymalizacji kodu źródłowego. Zoptymalizowana strona która wyświetla się użytkownikowi w kila sekund jest priorytetem dla programisty zajmującego się tworzeniem kodu. Do tego wykorzystywane są liczne skrypty dostarczane wielu programistów które sprawdzają stronę pod kątem ilości pobieranych danych czy jakości kodu źródłowego.

Część teoretyczna jest wprowadzeniem w problem tworzenia responsywnych stron internetowych. W dzisiejszych czasach, w których wiele użytkowników korzysta z Internetu za pomocą urządzeń mobilnych, strony internetowe muszą być dostosowane do różnorodności rozdzielczości ekranów.

## 2.1 Responsywność stron internetowych

Responsywne strony internetowe to strony które są w stanie dopasować się do monitora o każdej rozdzielczości w taki sposób aby strona wyświetlała się w sposób poprawny i czytelny dla użytkownika. Witryna dopasowuje się do rozdzielczości ekranu. Strony takie mają płynne rozmiary i nie wymagają aby użytkownik przybliżał czy przesuwał stronę po swoim monitorze aby doczytać niewidoczne treści.

Dawniej strony były mniej wyszukane, miały mniej interaktywnych elementów i „wodotrysków”. Tworzono osobną stronę internetową dopasowaną do małych rozdzielczości telefonów komórkowych działającą niezależnie od wersji komputerowych. Było to problematyczne gdyż wymagało aktualizowania dwóch witryn, tworzenia osobnej subdomeny i pozycjonowania następnej strony. Obecnie witryny tworzone są w taki sposób aby poprawnie wyświetlały się na każdym urządzeniu bez konieczności tworzenia drugiej strony internetowej. Takie projektowanie kodu pozwala oszczędzić na drugiej domenie i pieniądzach wydanych na pozycjonowanie strony. Aktualnie Google wprowadziło algorytm który obliczając parametry wpływające na pozycje strony w wynikach wyszukiwania uwzględnia fakt czy strona dobrze wyświetla się na urządzeniach mobilnych. Poprawne wyświetlanie się na wszystkich rozdzielczościach jest bardzo ważne. Na rynku jest wiele różnych urządzeń o różnych wymiarach ekranów, więc warto aby każdy użytkownik mógł za pomocą swojego telefonu czy tabletu korzystać z naszej witryny.

W tworzeniu responsywnych stron pomagają szkielety takie jak Bootstrap czy Foundation, można także korzystając z media queries wprowadzonego w CSS3 zaprojektować responsywność samodzielnie. Znajomość i projektowanie stron w oparciu o znane i rozwijane systemy szkieletowe pozwala nam tworzyć nowoczesne witryny które na pewno będą poprawnie wyświetlać się na każdym monitorze. W obecnych czasach tworzenie jednej witryny z szkieletem dopasowującym się do monitorów jest standardem. Ma to również wpływ na pozycje strony w wyszukiwarkach.

Niektóre jeszcze z używanych przeglądarek internetowych jak Internet Explorer 8, czyli najnowsza wersja dostępna na systemy Windows XP , oraz inne stare przeglądarki nie obsługują arkuszy stylów CSS3 wprowadzające elementy media które określają style jakie mają ładować się na konkretnych rozdzielczościach oraz nowości z HTML5. W tych przypadkach należy zastosować odpowiednie komponenty JavaScript dodające te funkcjonalności. W przeciwnym wypadku te przeglądarki nie będą poprawnie wyświetlać responsywnych stron internetowych. Do swojej poprawnej pracy Bootstrap potrzebuje także biblioteki jQuery, warto zastosować najnowszą wersję biblioteki pobraną z oficjalnej strony.

## 2.2 Porównanie frameworków do tworzenia responsywnych stron WWW

Bootstrap

Foundation

Skeleton

## System Bootstrap

Bootstrap, jest szkieletem programistycznym składającym się z rozwiązań CSS i JavaScript służącym do szybkiej budowy responsywnych stron internetowych. Przez swoją budowę w łatwy sposób można tworzyć oraz edytować kod HTML posługując się odpowiednimi klasami Bootstrapa.

Framework Bootstrap powstał w 2011 roku. Tworzony przez programistów pracujących przy portalu Twitter. Szybko jednak projektanci stwierdzili, że jest w tym systemie większy potencjał więc system został opublikowany w repozytorium GitHub aby programiści z całego świata mogli udoskonalić Bootstrapa.

W roku 2011 powstał Bootstrap w wersji 1.0.0. Składał się wówczas jedynie z komponentów HTML i CSS bez rozszerzeń Javascript, aż do wersji 1.3.0 która wprowadzała skrypty i zgodność z Internet Explorerem 7. Kolejna wersja została w większości napisana od początku. Wprowadzono elastyczną obsługę urządzeń mobilnych i różnych rozdzielczości ekranów monitorów. Wprowadzono wiele nowych klas i funkcjonalności CSS oraz rozszerzeń JavaScript. Największą rewolucje wprowadziła wersja 3, która aktualizowana jest do dnia dzisiejszego. Bootstrap 3 został napisany od nowa i nie jest kompatybilny z wcześniejszymi wersjami. Wprowadzono szereg nowych klas które pozwalają elastycznie sterować sposobem wyświetlania treści na różnych rozdzielczościach urządzeń. W tej wersji Bootstrap został odwrócony, wszystkie style są na początku określone dla urządzeń mobilnych, korzystając z właściwości kaskadowych arkuszy stylów dopasowanie się stylów do większych rozdzielczości ekranów.

Obecnie wiele motywów do popularnych systemów jak Wordpress czy Joomla wykorzystuje szkielet Bootstrapa. Zgodność ze standardem pozwala programistom szybko i sprawnie edytować czy implementować zaprojektowane pod szkielet Bootstrapa motywy stron internetowych. Popularność tego systemu pozwala w razie problemów znaleźć w Internecie potrzebne wsparcie w razie problemów z szkieletem Bootstrapa.

Bootstrap składa się z arkuszy stylów CSS, odpowiednich komponentów JavaScript które obsługują zdarzenia wywoływane przez użytkowników. System umożliwia także szybką konfiguracje stylów poprzez metajęzyki Sass lub Less. Za pomocą tych technologii można szybko dostosować szkielet Bootstrapa i domyślne wartości do swoich potrzeb. Umiejętnie posługując się tymi językami można zoptymalizować system do aktualnego projektu.

Podstawowa zaleta Bootstrapa to możliwość konfiguracji układu siatki w zależności od rozdzielczości ekranu za pomocą przypisania znacznikom HTML odpowiednich klas. Skrypty pozwalają budować zwijane menu wyświetlane tylko na ekranach o małych rozdzielczościach i inne interaktywne elementy stron internetowych.

Mimo że Bootstrap nie jest jedyną platformą do tworzenia responsywnych stron, jest najbardziej popularny i rozwijany w ramach wolnej licencji. Każdy programista może udoskonalić główną wersję Bootstrapa lub za pomocą formularza na oficjalnej stronie dostosować system do swoich potrzeb tak również będzie w mojej pracy, gdyż utworzona przeze mnie strona w całości opiera się o szkielet Bootstrapa.

## Wymagania i możliwości Bootstrapa

/\* wymagania dot przeglądarek

możliowości i ograniczenia gridu

\*/

# Siatkowy układ Bootstrapa

Bootstrap bazuje na strukturze siatki, umożliwia on poprawne ułożenie treści strony. Za pomocą odpowiednich klas przypisanych do elementów HTML tworzy się układ rzędów i kolumn. System Bootstrapa dzieli ekran na dwanaście równych kolumn w każdym rzędzie zamkniętym w kontener który zmienia swoją szerokość w zależności od rozdzielczości ekranu. Klasa „container” odpowiadająca za tworzenie kontenera treści powinna grupować elementy klasy „row”, czyli wiersze siatki. Każdy wiersz można podzielić na 12 kolumn za pomocą klas „col-xx-y”, gdzie „xx” odpowiada nazwie zakresu rozdzielczości, a „y” odpowiada ilości zajmowanych kolumn siatki. Zapis „col-lg-5” odpowiada podziałowi siatki na 5 z 12 dostępnych kolumn w układzie dla dużych ekranów powyżej 1200px szerokości. Kolumny mają określoną szerokość za pomocą wartości procentowych odnoszących się do szerokości kontenera dzieląc 100% ekranu na 12 zakresów. Bootstrap pozwala na definiowanie osobnych wyglądów układu siatki w czterech zakresach:

* col-xs – dla ekranów o szerokości poniżej 768px
* col-sm – dla ekranów o szerokości większej niż 768px
* col-md – dla ekranów o szerokości większej niż 992px
* col-lg – dla ekranów o szerokości większej niż 1200px

W zależności od tych szerokości zmienia się także wymiar kontenera. Szerokość klasy „container” w zależności od monitora zmienia się następująco:

* 100% zawartości – dla ekranów o szerokości poniżej 768px
* 750px – dla ekranów o szerokości większej niż 768px
* 970px – dla ekranów o szerokości większej niż 992px
* 1170px – dla ekranów o szerokości większej niż 1200px

Dostępna jest także klasa „container-fluid” która zawsze dopasowuje układ do szerokości przeglądarki. Można stosować ją zamiennie z klasą „container”.

Każdy wiersz można opisać dla różnych rozdzielczości osobno, w ten sposób definiując konkretny wygląd siatki układu. Przykładowo poniższy kod określa układ sześciu kolumn w jednym wierszu w przypadku ekranów powyżej 1200px, układ dwa wiersze z trzema kolumnami dla ekranów do rozdzielczości 992px, układ trzy wiersze z dwoma kolumnami dla rozdzielczości do 768px zaś dla rozdzielczości poniżej 768px układ zmienia się w jednokolumnowy z sześcioma wierszami.

<div class=”container”>

<div class=”row”>

<div class=”col-lg-2 col-md-3 col-sm-6 col-xs-12”></div>

<div class=”col-lg-2 col-md-3 col-sm-6 col-xs-12”></div>

<div class=”col-lg-2 col-md-3 col-sm-6 col-xs-12”></div>

<div class=”col-lg-2 col-md-3 col-sm-6 col-xs-12”></div>

<div class=”col-lg-2 col-md-3 col-sm-6 col-xs-12”></div>

<div class=”col-lg-2 col-md-3 col-sm-6 col-xs-12”></div>

</div>

</div>

Takie możliwości manipulacji układu dają pewność że na każdym ekranie treści będą wyświetlać się w sposób poprawny.

Za pomocą klasy „col-xx-offset-y” gdzie „xx” oznacza zakres rozdzielczości, „y” oznacza ilość kolumn można dodać elementowi kolumny przesunięcie o określoną ilość kolumn od lewej strony ekranu. Można w ten sposób utworzyć wiersz z treścią zawartą w elemencie zajmującym szerokość dwóch kolumn, lecz aby element był wyśrodkowany należy dodać przesunięcie równe pięciu kolumnom.

<div class=”container”>

<div class=”row”>

<div class=”col-md-2 col-md-offset-5”></div>

</div>

</div>

Dodatkowe opcje umożliwiające modyfikowanie układu siatki dodaje klasa „col-xx-pull-y” oraz „col-xx-push-y” gdzie „xx” oznacza zakres rozdzielczości, „y” oznacza ilość kolumn za pomocą których ręcznie można przesuwać kolumny. Pozwala to na przemieszczenie kolumn, zamienienie ich miejscem dla konkretnych rozdzielczości. Na poniższym przykładzie:

<div class=”container”>

<div class=”row”>

<div class=”col-md-4 col-md-push-8”></div>

<div class=”col-md-8 col-md-pull-4”></div>

</div>

</div>

Treść pierwszej kolumny która powinna zajmować cztery kolumny za pomocą klasy „col-\*\*-push-\*” zostaje przesunięta na miejsce drugiego elementu zajmującego osiem kolumn siatki.

Należy pamiętać że siatka Bootstrapa jest poprawna gdy kolumny są zawarte w wierszach, zaś wiersze umiejscowione są w kontenerach. Jest to ważne gdyż każda kolumna posiada 15px dopełnienia z lewej i prawej strony, zaś wiersze redukują zewnętrzne dopełniania za pomocą ujemnych marginesów tej samej wartości aby dopasować się do szerokości kontenera.

# Komponenty Bootstrapa

System Bootstrap za pomocą wbudowanych klas CSS umożliwia szybkie tworzenie elementów HTML takich jak nagłówki, listy, menu nawigacyjne, elementy formularzy, etykiety i wiele innych. Dodanie odpowiedniej klasy do elementu pozwala na dodanie właściwości CSS aby dany znacznik od razu wyglądał przyjaźnie. W razie potrzeb można wszystkie style elementów nadpisywać własnymi stylami.

Nagłówki stron od znacznika h1 do h6 w stylach Bootstrapa mają określone już wymiary, grubość czcionki, marginesy tak aby bez potrzeby pisania dodatkowych styli CSS wyglądały poprawnie. Element <h1> utworzony wewnątrz klasy „page-header” tworzy nam nagłówek strony który nie wymaga dodatkowych styli CSS.

Kontenery <div> z klasą „panel” pozwalają tworzyć na stronie elementy do wyświetlania treści wyróżnione w ramkach. Można dopisać dodatkową klasę określającą dodatkowe opcje kolorystyczne panelu:

* panel-danger – kolor czerwony
* panel-info – kolor błękitny
* panel-warning – kolor żółty
* panel-primary – kolor niebieski
* panel-success – kolor zielony

Wewnątrz panelu należy utworzyć dodatkowe obiekty: klasa „panel-heading” zawiera tekst nagłówka panelu, klasa „panel-body” określa treść wewnątrz panelu, klasa „panel-footer” grupuje elementy znajdujące się w stopce przykładowo przyciski.

Elementy HTML takie jak button lub input typu submit, czyli przyciski w Bootstrapie także mają swoje style. Dodanie klasy „btn” definiuje element wyglądający jak przycisk. Dodatkowe klasy pozwalają określenie kolorystyki przycisku:

* btn-default – dodaje kolor biały
* btn-primary – dodaje kolor ciemnoniebieski
* btn-danger – dodaje kolor czerwony
* btn-success – dodaje kolor zielony
* btn-info – dodaje kolor błękitny
* btn-warning – dodaje kolor pomarańczowy

Można również dodać klasę określającą wielkość przycisku:

* btn-lg – dla dużych przycisków
* btn-sm – dla małych przycisków
* btn-xs – dla bardzo małych przycisków

Dodatkowe klasy które można dodać do przycisków to:

* btn-block – dopasowuje przycisk do szerokości siatki
* active – dodaje wygląd informujący o kliknięciu przycisku
* disabled – dodaje wygląd informujący o wyłączeniu przycisku

Listy uporządkowane bądź nieuporządkowane pomocne są gdy chcemy wyświetlić przydatne zasoby bądź menu strony. Dodanie do listy klasy „list-group” zaś do elementów podrzędnych klas „list-group-item” tworzy nam komponent który poprawnie będzie wyświetlany na każdym monitorze. Do elementów listy można dodać element <span> z klasą „badge” który tworzy miejsce na wyświetlenie dodatkowej informacji jak ilość produktów przypadku listy produktów. Listy mogą posiadać dodatkowe klasy określające ich wygląd:

* list-group-item-success - dodaje kolor zielony
* list-group-item-danger - dodaje kolor czerwony
* list-group-item-warning – dodaje kolor żółty
* list-group-item-info – dodaje kolor błękitny

Bootstrap pozwala robić funkcjonalne menu nawigacyjne. Za pomocą odpowiednich klas można stworzyć menu w układzie pionowym lub poziomym, listę zakładek lub przycisków które w rozdzielczościach dla małych ekranów potrafią zmienić się w ukrytą listę która pokazuje się po kliknięciu w przycisk menu nawigacyjnego. Można w ten sposób zaoszczędzić wiele czasu na określanie styli menu dla każdej rozdzielczości i skryptów do obsługi ukrywania się listy dla urządzeń mobilnych. Lista elementów wymaga dodania klasy „nav” dodatkowo można określić wygląd za pomocą klas:

* nav-tabs – ustawia wygląd zakładek
* nav-pills – ustawia wygląd pigułek

Pasek nawigacji który tworzymy dodając klasę „navbar”. Wewnątrz tego elementu należy utworzyć element „container”. Dodatkowe elementy które należy dodać to komponent klasy „navbar-header” która odpowiada za nagłówek nawigacji, głównie zawiera logo oraz ukryty przycisk do obsługi ukrytego menu na urządzeniach mobilnych. Element <button> który widoczny jest jedynie w przypadku gdy menu zmienia swoją formę w ukrytą, musi zawierać dodatkowe atrybuty HTML5. Atrybut „data-toggle=collapse” określa, że element będzie brał udział w zwijaniu się nawigacji, zaś atrybut „data-target=id” określa identyfikator listy zawierającej odnośniki menu. Drugi element nawigacji to lista odnośników. Dla obsługi ukrywania listy na małych ekranach należy dodać klasę „colapse” oraz „navbar-collapse”. Lista powinna zawierać identyfikator identyczny z atrybutem „data-target” przycisku nawigacji. Tak utworzona struktura tworzy nam w pełni funkcjonalne menu które jest responsywne, gdyż w urządzeniach gdzie menu nie mieści się jak na dużym ekranie, lista ukrywa się a pojawia się przycisk który pozwala rozwinąć listę odnośników. W innym wypadku lista mogłaby wyświetlić się w sposób niepoprawny i niemożna by było na małych rozdzielczościach korzystać z menu nawigacyjnego.

Formularze

# Rozszerzenia Bootstrapa

/\* opis opcjonalnych javascriptów \*/

# Modyfikowanie i konfiguracja systemu

/\* opis możliwości konfiguratora \*/

# Optymalizowanie kodu

Bootstrap mimo iż pracuje nad nim wiele osób udostępniając nowe poprawki, nie jest w stu procentach uniwersalny. Zdarzają się przypadki projektów graficznych w których trzeba nagiąć zasady oryginalnego Boostrapa. W takim wypadku można przy użyciu generatora na oficjalnej stronie pobrać wersje dopasowaną do naszych potrzeb. Można w ten sposób zmienić odstępy, marginesy czy ilość kolumn. W ten sposób można stworzyć własny układ strony. Powoduje to jednak problemy w momencie aktualizacji głównej wersji systemu. Nie wystarczy wtedy pobrać z oficjalnego portalu najnowszej wersji - trzeba wygenerować Bootstrapa podając nasze zmienione parametry układu siatki.

Czasami jednak trzeba dopisać własne zmiany, modyfikując przy tym style które nadaje Bootstrap. Deklarując drugi plik CSS pod stylami Bootstrapa możemy do woli nadpisywać kod korzystając z zasady kaskadowości. Wtedy aktualizacja systemu do nowej wersji przebiega bezinwazyjnie.

Bootstrap ma jednak kilka wad. Na rynku urządzeń mobilnych jest wiele telefonów i tabletów które przyjmują różne szerokości ekranu w zakresie 320px do 992px. Powoduje to zmiany układu siatki w punkcie 768px. Poniżej tej szerokości ekranu mamy do wykorzystania już tylko jeden zestaw klas. Zakres klasy "col-xs" dotyczy urządzeń poniżej 768px szerokości. Główna klasa "container" otrzymuje wtedy szerokość 100% ekranu. Odpowiednie modyfikacje pozwolą nam dodać pośredni zakres układu siatki, który pozwoli zmienić wygląd strony dla standardowych smartphonów. Daje to większą kontrolę nad stroną która będzie bardziej dopasowana do wymagań tabletów i telefonów dotykowych. Dodatkowy zakres klas "col-tm" dodany dla rozdzielczości poniżej 768px do 481px pozwoli modyfikować układ siatki dla smartphonów o wysokiej rozdzielczości i tabletów. Wymagane jest dopisanie obsługi klas które istnieją w pozostałych klasach układu: "col-tm-", "col-tm-offset-", "col-tm-push-", "col-tm-pull-", "visible-tm-", "hidden-tm-", oraz modyfikacje klas "container", "visible-", "hidden-" opisujących pozostałe zakresy rozdzielczości. Takie rozwiązanie jest uniwersalne, aktualizacja głównego pliku CSS z Bootstrapem nie spowoduje żadnych konfliktów.

Istniejący już zakres "col-xs" będzie obowiązywał w rozdzielczościach poniżej 480px włącznie. Dodatkowy przeskok w układzie siatki pozwoli na zróżnicowanie wyglądu strony, lepiej dopasowując elementy do wymagań rozdzielczości monitorów.

# Część praktyczna

/\* opis strony \*/

## 3.1 Rozszerzenie funkcjonalności Bootstrapa

W dotychczasowej wersji Bootstrap zmusza nas do czterech zakresów różnych układów siatki. Klasa "col-xs" która definiuje wygląd dla urządzeń poniżej 768px szerokości zmusza nas do stworzenia układu który dobrze będzie wyglądał na małym smartfonie o szerokości ekranu 320px jak i tablecie o wyświetlaczu do 768px. Uzyskanie dodatkowego zakresu układu siatki w zakresie 481px do 767px pozwoli nam zdefiniować inny wygląd strony dla urządzeń mieszczących się w tym zakresie. W tym celu, wykorzystując kaskadowość arkuszy stylów utworzono kod który ogranicza zakres klasy "col-xs" do 480px włącznie, zaś nowy zakres "col-tm" obejmuje szerokości 481px do 767px. Taki kod musi być dołączony w pliku CSS poniżej deklaracją arkuszu stylów Bootstrapa. Poniżej znajduje się kod rozszerzający funkcjonalność systemu Bootstrap:

@media (min-width: 481px) and (max-width:767px) {

.col-tm-1, .col-tm-2, .col-tm-3, .col-tm-4, .col-tm-5, .col-tm-6, .col-tm-7,

.col-tm-8, .col-tm-9, .col-tm-10, .col-tm-11, .col-tm-12 {

float: left;

}

.col-tm-12 {

width: 100%;

}

.col-tm-11 {

width: 91.66666667%;

}

.col-tm-10 {

width: 83.33333333%;

}

.col-tm-9 {

width: 75%;

}

.col-tm-8 {

width: 66.66666667%;

}

.col-tm-7 {

width: 58.33333333%;

}

.col-tm-6 {

width: 50%;

}

.col-tm-5 {

width: 41.66666667%;

}

.col-tm-4 {

width: 33.33333333%;

}

.col-tm-3 {

width: 25%;

}

.col-tm-2 {

width: 16.66666667%;

}

.col-tm-1 {

width: 8.33333333%;

}

.col-tm-pull-12 {

right: 100%;

}

.col-tm-pull-11 {

right: 91.66666667%;

}

.col-tm-pull-10 {

right: 83.33333333%;

}

.col-tm-pull-9 {

right: 75%;

}

.col-tm-pull-8 {

right: 66.66666667%;

}

.col-tm-pull-7 {

right: 58.33333333%;

}

.col-tm-pull-6 {

right: 50%;

}

.col-tm-pull-5 {

right: 41.66666667%;

}

.col-tm-pull-4 {

right: 33.33333333%;

}

.col-tm-pull-3 {

right: 25%;

}

.col-tm-pull-2 {

right: 16.66666667%;

}

.col-tm-pull-1 {

right: 8.33333333%;

}

.col-tm-pull-0 {

right: auto;

}

.col-tm-push-12 {

left: 100%;

}

.col-tm-push-11 {

left: 91.66666667%;

}

.col-tm-push-10 {

left: 83.33333333%;

}

.col-tm-push-9 {

left: 75%;

}

.col-tm-push-8 {

left: 66.66666667%;

}

.col-tm-push-7 {

left: 58.33333333%;

}

.col-tm-push-6 {

left: 50%;

}

.col-tm-push-5 {

left: 41.66666667%;

}

.col-tm-push-4 {

left: 33.33333333%;

}

.col-tm-push-3 {

left: 25%;

}

.col-tm-push-2 {

left: 16.66666667%;

}

.col-tm-push-1 {

left: 8.33333333%;

}

.col-tm-push-0 {

left: auto;

}

.col-tm-offset-12 {

margin-left: 100%;

}

.col-tm-offset-11 {

margin-left: 91.66666667%;

}

.col-tm-offset-10 {

margin-left: 83.33333333%;

}

.col-tm-offset-9 {

margin-left: 75%;

}

.col-tm-offset-8 {

margin-left: 66.66666667%;

}

.col-tm-offset-7 {

margin-left: 58.33333333%;

}

.col-tm-offset-6 {

margin-left: 50%;

}

.col-tm-offset-5 {

margin-left: 41.66666667%;

}

.col-tm-offset-4 {

margin-left: 33.33333333%;

}

.col-tm-offset-3 {

margin-left: 25%;

}

.col-tm-offset-2 {

margin-left: 16.66666667%;

}

.col-tm-offset-1 {

margin-left: 8.33333333%;

}

.col-tm-offset-0 {

margin-left: 0;

}

.visible-tm {

display: block !important;

}

table.visible-tm {

display: table !important;

}

tr.visible-tm {

display: table-row !important;

}

th.visible-tm,

td.visible-tm {

display: table-cell !important;

}

.visible-tm-block {

display: block !important;

}

.visible-tm-inline {

display: inline !important;

}

.visible-tm-inline-block {

display: inline-block !important;

}

.hidden-tm {

display: none !important;

}

.hidden-xs{

display:block !important;

}

.visible-xs, table.visible-xs, tr.visible-xs, th.visible-xs, td.visible-xs,

.visible-xs-block, .visible-xs-inline, .visible-xs-inline-block {

display: none !important;

}

}

## 3.2 Budowa struktury strony

/\* opis layoutu i rozwiązań\*/

### 3.2.1 Obsługa starych przeglądarek internetowych

/\* dodanie javascriptów respond i html5shiv\*/

## 3.3 Testowanie wersji mobilnej

/\* testy w narzędziach do mobile + screeny\*/

# Zakończenie

# Bibliografia

1. http://www.mgsm.pl/pl/katalog/goclever/quantum4/Goclever-Quantum-4.html [↑](#footnote-ref-2)
2. http://www.mgsm.pl/pl/katalog/samsung/galaxys7/Samsung-Galaxy-S7.html [↑](#footnote-ref-3)
3. Framework – szkielet budowy aplikacji w danym języku programowania, określa metody i mechanizm działania programu. [↑](#footnote-ref-4)