

Uniwersytet Ekonomiczny w Katowicach  
Wydział Informatyki i Komunikacji  
Kierunek: *Informatyka i Ekonometria*

**Patryk Nowak**

***Języki, narzędzia i technologie wykorzystywane w  
procesie tworzenia stron internetowych***

***Languages, tools and technologies used in the process of  
creating websites***

Praca licencjacka  
napisana w Katedrze Informatyki  
pod kierunkiem dr Artura Strzeleckiego

*Pracę przyjmuję i wnioskuję o jej dopuszczenie  
do dalszych etapów postępowania egzaminacyjnego*

.....  
(data)

.....  
(podpis promotora pracy licencjackiej)

**KATOWICE 2018**

Katowice, dnia .....

.....  
Imię i nazwisko

.....  
Wydział

.....  
Kierunek

### **OŚWIADCZENIE**

Świadom(a) odpowiedzialności prawnej oświadczam, że złożona praca licencjacka/magisterska pt.:..... została napisana przeze mnie samodzielnie.

Równocześnie oświadczam, że praca ta nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz. U. 1994, nr 24, poz. 83) oraz dóbr osobistych chronionych prawem.

Ponadto praca nie zawiera informacji i danych uzyskanych w sposób nielegalny i nie była wcześniej przedmiotem innych procedur związanych z uzyskaniem dyplomów lub tytułów zawodowych uczelni wyższej.

Wyrażam zgodę na przetwarzanie moich danych osobowych oraz nieodpłatne udostępnienie mojej pracy w celu oceny samodzielności jej przygotowania przez system elektronicznego porównywania tekstów oraz przechowywania jej w bazie danych tego systemu.

Oświadczam także, że wersja pracy znajdująca się na przedłożonej przez mnie płycie CD jest zgodna z wydrukiem komputerowym pracy.

.....  
(podpis składającego oświadczenie)

## Spis treści

Wstęp .....	5
1. Języki wykorzystywane przy tworzeniu stron internetowych .....	6
1.1. HTML .....	6
1.1.1. Znaczniki .....	6
1.1.3. Kod HTML .....	7
1.1.4. Pliki HTML.....	7
1.1.5. Historia.....	7
1.1.6. HTML5 .....	8
1.2. Kaskadowe Arkusze Stylów .....	9
1.2.1. Identyfikatory i klasy .....	9
1.2.2. Łączenie CSS z HTML.....	10
1.2.3. Historia.....	11
1.2.4. Prefiksy przeglądarek .....	11
1.2.5. Gotowe rozwiązania CSS .....	12
1.2.6. Kaskada Stylów i dziedziczenie .....	13
1.3. JavaScript.....	13
1.3.1. Standardy JavaScript.....	14
1.3.2. Działanie skryptów .....	15
1.3.3. Przeciążenie strony .....	15
1.3.4. Biblioteki Javascript .....	18
1.3.5. JSON.....	22
1.4. PHP .....	22
2. Narzędzia i systemy wspierające tworzenie stron internetowych .....	25
2.1. Hosting i domena .....	25
2.1.1. Darmowy hosting.....	25

2.1.2. Płatny hosting .....	26
2.1.3. Najważniejsze cechy.....	27
2.1.4. Pozostałe parametry .....	30
2.1.5. Typy hostingów .....	32
2.2. Web cache.....	35
2.2.1. Content Delivery Network.....	37
2.3. Secure Socket Shell .....	40
2.4. Systemy zarządzania treścią .....	40
3. Tworzenie strony internetowej .....	44
3.1. Projekt graficzny strony.....	44
3.2. Kod strony – wersja desktop.....	45
3.3. Wersja mobilna.....	55
3.4. Hosting.....	58
Podsumowanie.....	60
Bibliografia.....	61
Spis ilustracji i tabel.....	62

## Wstęp

Strony internetowe są nierozłączną częścią Internetu, to na nich opiera się struktura sieci jaka znajoma jest milionom ludzi. Każdego dnia w obiegu pojawiają się nowe witryny, a temat tworzenia i zarządzania nimi wzbudza zainteresowanie kolejnych osób. Szybko rozwijająca się technologia pozwala na budowanie co raz to bardziej zaawansowanych technicznie serwisów, które oferują co raz to większą funkcjonalność.

Nowoczesne serwisy internetowe tworzone są z wykorzystaniem wielu języków. Istnieje wiele możliwości, jednak najczęściej wybór pada na języki HTML, CSS, PHP i JavaScript, gdyż wzajemnie się uzupełniając pozwalają one Web Developerom na tworzenie zaawansowanych witryn. Z tematyką witryn internetowych związane są również narzędzia niezbędne do uruchomienia strony, oraz te wspomagające jej działanie.

Celem tej pracy jest omówienie technologii związanych z tworzeniem stron internetowych, oraz praktyczna prezentacja procesu budowania i publikowania strony www w sieci.

W pierwszym rozdziale pracy, autor przedstawi cztery najpopularniejsze języki wykorzystywane przez Web Developerów w celu pisania witryn. Omówione zostaną ich właściwości, oraz zagadnienia z nimi powiązane. W rozdziale drugim zawarte zostaną informacje o niektórych rozwiązaniach wspierających tworzenie, optymalizację, oraz zarządzanie serwisami internetowymi. Przeprowadzona zostanie również analiza opcji hostingowych. Po przeglądzie dostępnych rozwiązań w rozdziale trzecim przedstawiony zostanie proces tworzenia strony internetowej, wraz z etapami na jakie się składa. Autor pracy podzielił proces tworzenia stron internetowych, na projektowanie graficzne, budowanie struktury strony, określenie właściwości graficznych dokumentu HTML za pomocą języka CSS, oraz usprawnieniu działania witryny z użyciem JavaScript. Następnie gotowe już rozwiązanie zostanie zaimportowane na serwer hostingowy.

Wynikiem tej pracy, wraz z końcem trzeciego rozdziału będzie działająca strona internetowa opublikowana w sieci na wybranej uprzednio domenie. Strona graficzna witryny opierała będzie się o typ serwisów zwany One Page. Ten rodzaj serwisów charakteryzuje się małym obciążeniem serwera, oraz wielką przejrzystością.

# 1. Języki wykorzystywane przy tworzeniu stron internetowych

Rozdział pierwszy pracy poświęcony będzie omówieniu najpopularniejszych języków wykorzystywanych przy tworzeniu nowoczesnych serwisów internetowych. Padnie odpowiedź na pytanie czym owe języki są, wraz z krótkim przedstawieniem ich genezy. Mimo iż wybór jest ogromny, to praca dotyczyć będzie tych najbardziej popularnych rozwiązań, które w roku 2018 podbijając rynek stały się najczęstszym wyborem wśród osób zajmujących się tworzeniem serwisów internetowych utrzymując na nim zdecydowaną pozycję lidera. Mowa tu o czterech filarach jakimi są HTML, CSS, PHP, oraz język skryptowy JavaScript.

## 1.1. HTML

HTML to język wykorzystywany do tworzenia stron internetowych. Pełna angielska nazwa to Hypertext Markup Language, co w wolnym tłumaczeniu oznacza hipertekstowy język znaczników. Umożliwia on pisanie tekstu, który zostanie zinterpretowany przez przeglądarki i po przetworzeniu wyświetlony w postaci witryny internetowej. W sieci istnieje bardzo szeroki wybór przeglądarek internetowych, których zadaniem jest interpretacja i wyświetlanie kodu HTML. Język HTML odpowiada w szczególności za tworzenie swoistego szkieletu stron internetowych. Witryny te po uprzednim połączeniu z innymi, później opisanymi technologiami pozwalają na budowanie bardziej zaawansowanych projektów. Istnieje oczywiście możliwość ograniczenia się do samego Hipertekstowego Języka Znaczników, jednakże biorąc pod uwagę zdobytą na przestrzeni lat wiedzę w tej dziedzinie jest to kiepski pomysł. Warunkowane jest to faktem, iż strony takie nie są atrakcyjne w oczach użytkownika, oraz brakuje im wymaganej funkcjonalności.

### 1.1.1. Znaczniki

Znaczniki, inaczej Tagi są podstawowym elementem HTML, bo to na nich opiera się cała struktura języka. Zawarty pomiędzy nimi tekst jest interpretowany i wyświetlany przez przeglądarkę internetową. Większość znaczników oprócz otwarcia musi zostać też zamknięta. Różnicą pomiędzy znacznikiem otwarcia i zamknięcia jest taka, że ten drugi posiada poprzedzający go znak *slash*. W praktyce wygląda to tak **<p> przykładowy tekst </p>**. HTML oferuje szeroki wybór znaczników - wygląd których można określać poprzez sprecyzowane parametry zawarte w kodzie. Dla szczegółowej edycji wyglądu zalecane jest

jednak użycie Kaskadowych Arkuszy Stylów, które uzupełniają HTML o możliwości dokładnej edycji warstwy graficznej.

### **1.1.3. Kod HTML**

Istnieje jeden warunek, który musi zostać spełniony podczas pisania kodu. Dotyczy on programu używanego do tworzenia stron. Kod nie może być tekstem sformatowanym. W tym celu należy używać edytorów tekstu, które w żaden sposób nie wpływają na format. Edytory takie jak Microsoft Word odpadają już na wstępie. Do dyspozycji Web Developerów jest wbudowany w system Notatnik, lub WordPad. Istnieją też edytory tekstu tworzone wyłącznie w celu uproszczenia i umilenia kodowania, które wyspecjalizowane są w tej dziedzinie. Jednym z najbardziej znanych i najczęściej używanych jest prosty i przejrzysty **Notepad++**. Jako, że technologia oferuje wiele możliwości, to do dyspozycji programistów istnieją też bardziej zaawansowane edytory. Przy wszystkich opcjach jakie oferuje Internet, to jak dostosowane jest stanowisko zależy tylko od potrzeb konkretnej osoby.

### **1.1.4. Pliki HTML**

Tak jak ma to miejsce z wszystkimi plikami - kod HTML, by zadziałał musi być zawarty w pliku o odpowiednim rozszerzeniu. Język ten w działaniu jest bardzo uniwersalny i można odczytać, oraz edytować go w każdym systemie desktopowym. Jedynym wymaganiem oprócz niesformatowanego kodu jest zapisanie go w pliku **.htm**, **.html**, bądź **.php**. Od tego momentu każda przeglądarka będzie w stanie go zinterpretować i wyświetlić.

### **1.1.5. Historia**

Korzenie HTML sięgają roku 1980, kiedy to fizyk Tim Berners-Lee stworzył prototypowy system mający na celu ułatwienie pisania i prezentacji dokumentów o nazwie ENQUIRE. W roku 1991 ze współpracy tej samej osoby z inżynierem oprogramowania Robertem Cailliau powstała pierwsza publiczna wersja HTML, nazwana wtedy HTML Tags. Był to wielki początek języka, który jest znany i używany na całym świecie. Wersja ta nie była jednak bardzo rozbudowana, z racji, że zawierała tylko 22 znaczniki, które na przestrzeni lat wraz z rozwojem były zmieniane, lub wycofywane z użytku. Prace nad doskonaleniem trwały kolejne lata, a twórcy przeglądarek poprzez liczne eksperymenty tworzyli kolejne szkice języka. Ostatecznie w roku 1995 zaprezentowana została oficjalna specyfikacja nosząca nazwę HTML 2.0. Od tego czasu rozwój był dynamiczny, a sam język rozwijał się w zaskakująco szybkim tempie. Szybko wzrastało też zainteresowanie. Od tego momentu

bardzo często pojawiały się poprawki, co ostatecznie doprowadziło do opublikowania w dniu 15 maja 2000r. wersji HTML 4.01 jako międzynarodowego standardu ISO/IEC. Jednak prace nie zatrzymały się po wydaniu oficjalnego standardu. Mimo, że wersja 4.01 utrzymywała się na rynku przez długie lata, to próby udoskonalenia trwały cały czas doprowadzając do wydania szkicu wersji 5 na początku roku 2008. Najnowszą modyfikację czekała jeszcze długa droga do uznania jej za oficjalną rekomendację. Doczekała się tego dopiero pod koniec roku 2014.

### 1.1.6. HTML5

HTML 5.1 stał się rekomendacją i to na niej opiera się większość Web Developerów. Od pojawienia się tego wydania, a więc od roku 2014 - budowa serwisów internetowych opiera się w dużym stopniu na wprowadzonych w 5.1 znacznikach. Wersja ta różni się od poprzedników między innymi naciskiem jaki kładziony jest na przejrzystość kodu, co poprawione zostało poprzez wprowadzenie tagów. Dokument dokładniej niż w poprzednich wersjach HTML dzielony jest na opisujące go sekcje, co poprawia jasność kodu. Po wprowadzeniu kilku nowości nie ma potrzeby ciągłego stosowania znaczników **<div>**. Dodane zostały między innymi:

- <header>** - służy jako nagłówek strony internetowej. Zawiera najczęściej nawigację witryny, logo, lub tytuł,
- <nav>** - znacznik, który zawierać powinien główną nawigację,
- <main>** - główna część strony internetowej, zawiera najważniejszą treść,
- <section>** - sekcja dokumentu grupująca konkretne treści,
- <article>** - samodzielny fragment treści strony WWW, np. wpis na blogu,
- <aside>** - element z dodatkową treścią powiązaną z zawartością całej strony internetowej, najczęściej wykorzystywany przy tworzeniu sidebar'a,
- <footer>** - najczęściej umieszczona na końcu sekcja zawierająca informacje o zawartości strony.

Mimo iż są to najczęściej wskazywane udogodnienia wprowadzone w nowej wersji, to nie są one jedynymi nowościami. Zgodnie z zamiarem zmniejszają one znaczenie istniejącego już wiele lat znacznika **<div>**, który był powszechnie wykorzystywany przy dzieleniu stron internetowych na konkretne sekcje. Rzadziej używany jest również element **<span>**, który stracił na rzecz markerów takich jak **<mark>**, czy **<output>**. Poprawa jasności w kodzie nie była jedyną zmianą wniesioną wraz z tą wersją. Poprawiona została również obsługa błędów.



Niepoprawnie napisane fragmenty kodu wyświetlane są w ten sam sposób niezależnie od przeglądarki internetowej. W poprzednich odsłonach HTML brak jednolitości doprowadzał do błędów w wyświetlaniu stron przy użyciu niektórych przeglądarek.

Pomimo naprawienia interpretacji błędów, to wraz z nowymi technologiami pojawiają się kolejne problemy. Nowe znaczniki wprowadzone przez HTML5 wywołują problem przy używaniu starszych narzędzi. Większość starych przeglądarek radzi sobie z wyświetlaniem podmieniając nowe znaczniki tagiem `<span>`. W tym przypadku jednak style CSS nie są właściwie zastosowane przy odpowiednich fragmentach witryny, co wprowadza spory nieporządek.

Można temu zaradzić poprzez użycie specjalnie przygotowanych skryptów JavaScript. Po zastosowaniu kodu wypełnienia starsze wersje przeglądarek będą działały tak jak ich nowsze odpowiedniki. Sam skrypt dostępny jest za darmo w sieci, w celu zastosowania nie jest wymagana więc znajomość JavaScript.<sup>1</sup>

## 1.2. Kaskadowe Arkusze Stylów

Kaskadowe Arkusze Stylów to język służący do opisu sposobu graficznego przedstawienia strony. CSS pozwala na określenie reguł według, których wyświetlone przez przeglądarkę zostaną poszczególne znaczniki HTML. Arkusze Stylów pozwalają na ustalenie takich dyrektyw jak kolor tekstu, czy też ułożenie elementu względem krawędzi przeglądarki.

Bardzo ważnym pojęciem związanym z CSS jest **reguła**, czyli każdorazowe zastosowanie stylu. Reguły składają się z dwóch sekcji - selektora, oraz deklaracji. Pierwsze z nich odpowiedzialne jest za określenie konkretnego elementu HTML, którego wygląd na stronie ma zostać zdefiniowany. Za selektorem znajduje się deklaracja, w której określonej **właściwości** nadaje się konkretną **wartość**. Ta z kolei definiuje sposób w jaki dany element zostanie wyświetlony w przeglądarce. Kaskadowe Arkusze Stylów pozwalają na wprowadzenie kilku deklaracji w jednej regule tworząc tym samym **grupę deklaracji**.

### 1.2.1. Identyfikatory i klasy

Pojęcie, oraz wykorzystanie identyfikatorów i klas jest głęboko zakorzenione w serwisach internetowych. Na potrzeby zmiany stylu pojedynczego elementu strony niektóre znaczniki HTML wymagają sprecyzowanego selektora. Selektor ten stworzyć można przypisując do

---

<sup>1</sup> Clarissa Peterson, Responsywne strony WWW dla każdego, Wydawnictwo Helion, 2015, str. 54

danego elementu klasę, lub identyfikator, które funkcjonują podobnie z różnicą, że klasę można przypisać dowolnej liczbie tagów, a identyfikator tylko jednemu.

Użyte nazwy nie mają żadnego znaczenia dla przeglądarek, dlatego przy tworzeniu klas i identyfikatorów warto używać nazw opisowych. W tym wypadku liczy się tylko i wyłącznie przejrzystość kodu, która ma ułatwiać pracę Web Developerowi.<sup>2</sup> Istnieją oczywiście niezobowiązujące zasady mające na celu ujednoczenie kodów różnych stron, które używane są przez wielu programistów na całym świecie. Stosowanie się do owych reguł nie jest jednak obowiązkowe.

### 1.2.2. Łączenie CSS z HTML

Istnieje kilka możliwości zastosowania stylów na stronie - można wbudować je w kod strony, lub dołączyć je w osobnym pliku .css.

Najprostszą metodą definiowania styli jest zmiana wartości bezpośrednio przy znaczniku HTML. Pozwala to na precyzyjne zmiany wyglądu w odniesieniu do jednego konkretnego elementu. W tym celu w tagu należy wstawić atrybut `style=".."` podając w cudzysłowie style jakie mają zostać zastosowane. Należy w miarę możliwości unikać wstawiania stylów bezpośrednio w kod witryny z racji na nieporządek jakie wywołują one w kodzie. Metoda ta przydatna jest jednak w kilku przypadkach, jednym z nich jest testowanie potencjalnych zmian wprowadzanych na stronie.<sup>3</sup>

Oprócz wyżej omówionej metody jest możliwość wprowadzania zmian graficznych bezpośrednio w sekcji `<head>` strony. W celu modyfikacji właściwości danego elementu wewnątrz znaczników `<style>` należy umieścić odpowiednie reguły CSS. Pomimo faktu, iż sposób ten jest bardziej popularny od poprzednika, to nadal nie jest najlepszą możliwością. Modyfikowanie stylu w nagłówku ogranicza wszelkie wprowadzone zmiany tylko i wyłącznie do pliku HTML, w którym zmiana została zdefiniowana. W przypadku serwisów składających się z kilku podstron zmuszałoby to do uzupełniania wszystkich segmentów strony o te same, długie kody CSS. Dlatego niepodważalnie najlepszą opcją na połączenie szkieletu WWW ze stworzonymi dla niego stylami jest dołączenie zewnętrznego pliku

---

<sup>2</sup> Clarissa Peterson, Responsywne strony WWW dla każdego, Wydawnictwo Helion, 2015, str. 62

<sup>3</sup> Clarissa Peterson, Responsywne strony WWW dla każdego, Wydawnictwo Helion, 2015, str. 68

arkusza stylów. Nie tylko umożliwia to trzymanie layoutu graficznego dla wszystkich podstron w jednym pliku, ale nie wpływa również negatywnie na porządek i prostotę kodu.

Każda strona prędzej czy później ulegnie jakimś zmianom. W przypadku używania osobnego pliku ze stylami wielkim atutem jest przejrzystość ułatwiająca znalezienie potrzebnego fragmentu kodu, oraz jego edycja bez tworzenia zbędnego bałaganu w kodzie HTML. Stworzenie dobrze działającego layoutu strony zajmuje porównywalnie z innymi czynnościami dużo czasu. Toteż trzymanie tych informacji w osobnym pliku zabezpiecza je lepiej przed niechcianymi i przypadkowymi zmianami mogącymi wyniknąć z drobnych edycji strony WWW. Plik arkusza zewnętrznego można połączyć z plikiem HTML za pomocą komendy `<link>` zawartej w nagłówku strony.

### **1.2.3. Historia**

W miarę rozrostu HTML i przebijania się go jako języka do tworzenia struktur serwisów internetowych wzrastała potrzeba na powstanie dodatkowych możliwości związanych ze zmianami szaty graficznej dokumentów. Sprawę w swoje ręce wzięli ludzie odpowiadający za przeglądarki internetowe. Powstało wiele nowych znaczników, które wprowadzały na rynek ogromne zamieszanie. W efekcie każda strona musiała zostać tworzona w kilku różniących się wersjach, tak aby każda inna przeglądarka mogła wczytać ją we właściwy sposób. Nie dawało to jednak żadnej gwarancji, że każda osoba zobaczy tak samo wyglądającą stronę. Z propozycją ujednoczenia wyszedł Håkon Wium Lie. W 1994 roku zaproponował on działające dla wszystkich tak samo CHSS. Następnie w roku 1996 wydana została pierwsza wersja Kaskadowych Arkuszy Stylów. Od tego czasu CSS rozwijał się na równi z językiem HTML. W wersji trzeciej wprowadzone zostały bardzo duże zmiany, które zmieniły patrzenie na sposób tworzenia serwisów internetowych. CSS3 oprócz nowych właściwości elementów wprowadził zapytania o media umożliwiające dostosowanie wyświetlania strony pod urządzenie z jakiego korzysta osoba odwiedzająca.

### **1.2.4. Prefiksy przeglądarek**

Podczas gdy niektóre części CSS3 są w fazie opracowywania i testowania przez twórców przeglądarek, to mogą pojawić się nieścisłości w wyświetlaniu stron na różnych przeglądarkach w zależności od sposobu w jaki zostaną one zinterpretowane. W celu ujednoczenia sposobu w jaki niedopracowany element zostanie wyświetlony zostały wprowadzone **prefiksy przeglądarek** pozwalające Web Developerom na lepszą kontrolę. Polega to na określeniu sposobu wyświetlania na czterech głównych silnikach

wizualizacyjnych, z których korzystają najbardziej popularne przeglądarki internetowe. Są to:<sup>4</sup>

- **WebKit** - Safari, oraz Google Chrome
- **Gecko** - Mozilla Firefox
- **Trident** - Internet Explorer
- **Presto** - Opera

Każdy z nich może przetworzyć stronę w inny sposób dlatego wymagane jest zastosowanie wszystkich czterech prefiksów, które dla wyżej wymienionych silników to kolejno:

- -webkit-
- -moz-
- -ms-
- -o-

### 1.2.5. Gotowe rozwiązania CSS

Wraz z rozwojem technologii Web Developerzy zaczęli dążyć do uproszczenia codziennej pracy. Zaczęły więc powstawać Frameworki, czyli predefiniowane style, które po zaimplementowaniu odpowiedniej biblioteki pozwalają na zmianę wyglądu witryny bazując na przygotowanych już stylach. Z kolei te bardziej zaawansowane oprócz rozwinięcia od strony graficznej oferują również przeróżne skrypty ubogacające funkcjonalność serwisu. Jak to ma miejsce ze wszystkim w Internecie - istnieje wiele Frameworków. Nie obeszło by się jednak bez wspomnienia o dwóch, które znacząco pochłaniają konkurencję - czyli **Bootstrap**, oraz **Foundation**. Oba oparte są na systemie siatki, która pozwala na łatwe, logiczne i estetyczne projektowanie. Bardzo dużo zaawansowanych Web Developerów porzuciło już pisanie własnych stylów od podstaw, a swoje strony tworzą przy pomocy wszechobecnie dostępnych Frameworków. Platformy CSS zawierają style dotyczące wszystkich potencjalnych elementów WWW. Często jest to powodem niepotrzebnego obciążenia strony. Usunięcie nieużywanych stylów z arkusza może pozytywnie wpłynąć na czas ładowania serwisu.

---

<sup>4</sup> Clarissa Peterson, Responsywne strony WWW dla każdego, Wydawnictwo Helion, 2015, str. 64-67

### 1.2.6. Kaskada Stylów i dziedziczenie

Kaskadowość stylów określa pierwszeństwo w oddziaływaniu na te same elementy strony stylów z różnych źródeł.<sup>5</sup> Każdy element WWW posiada zestaw właściwości określany przez style. Często zdarza się, że jeden element przypisane ma kilka różnych wartości. Kaskadowość określa, która z nich jest najważniejsza i powinna zostać zaaplikowana podczas ładowania witryny. Jest to więc swego rodzaju hierarchia reguł.

Kolejność reguł według priorytetów:<sup>6</sup>

1. Reguły oznaczone jako ważne.
2. Reguły stylów wstawianych
3. Reguły zawierające identyfikatory
4. Reguły zawierające klasy, atrybuty i pseudoklasy
5. Reguły zawierające elementy i pseudoelementy
6. Reguły odziedziczone
7. Wartości domyślne

Kolejność przedstawiona powyżej jest często określana jako **precyzja CSS**. Nazwę zawdzięcza temu, że zaczyna się od najbardziej szczegółowej reguły i zmierza w kierunku tej najbardziej ogólnej.

Pojęcie dziedziczenia w CSS można wyjaśnić dosyć krótko. Otóż jeżeli w elemencie nie jest zastosowany styl to przeglądarka szuka elementu nadrzędnego i to jego styl zostaje pobrany dla danego znacznika. Nie wszystkie właściwości są jednak dziedziczne - mowa tu np. o ramkach, które nie przekazują wartości na elementy podrzędne. W przypadku gdy dwa style są według hierarchii równe - porównywana jest liczba atrybutów w selektorze. W tym wypadku większa liczba oznacza większy priorytet. Jeżeli jednak oba style mają równą liczbę atrybutów, to ten znajdujący się w pliku niżej będzie tym, który zostanie zastosowany.

### 1.3. JavaScript

Lata doświadczenia w tworzeniu witryn doprowadziły do nowego, powszechnie przyjętego punktu patrzenia na ogólne zamierzenia i powód tworzenia stron internetowych. Oprócz dominacji minimalizmu w dziedzinie graficznej, bardzo ważne stały się skrypty, które we współdziałaniu z kodem HTML wydeptują nowe ścieżki funkcjonalności serwisów.

---

<sup>5</sup>Kurs HTML [online] [ <http://www.kurshtml.edu.pl>] dostęp:15.04.2018r.

<sup>6</sup> Clarissa Peterson, Responsywne strony WWW dla każdego, Wydawnictwo Helion, 2015, str. 69

JavaScript to skryptowy język programowania. Pomimo, że jest on wykorzystywany przez wiele środowisk, to najbardziej kojarzony jest ze stronami internetowymi. Język ten umożliwia tworzenie zagnieżdżonych bezpośrednio w kodzie HTML krótkich programów. Działanie stworzonych za pomocą kodu JavaScript programów opiera się głównie na zdarzeniach wywołanych najczęściej przez użytkownika, np. kliknięcie myszą, zmiana rozmiaru okna czy przesunięcie elementu witryny. Owe zdarzenia wywołują ustalone wcześniej reakcje. Może to być między innymi pojawienie się komunikatu, lub zmiana graficzna wyświetlanej treści. Wykorzystanie języka skryptowego otwiera nowe drogi dla Web Developerów. Zakres możliwości jest szeroki, począwszy od prostego zegara na sprawdzenie godziny, przez tworzenie dynamicznego menu, aż po skrypty bezpośrednio angażujące gościa w interakcję. To ostatnie zdecydowanie zmieniło sposób funkcjonowania nowoczesnych witryn i nakreśliło nowe normy. W roku 2018 trudno już spotkać serwis, który nie korzystałby z JavaScriptu. Dlatego każdy twórca nowoczesnych stron internetowych musi więc oprócz HTML-a i stylów CSS, znać również i ten język, a przynajmniej jego podstawy.<sup>7</sup>

### **1.3.1. Standardy JavaScript**

Javascript wprowadza pewne zamieszanie związane z nazewnictwem oraz wersją języka. Można spotkać się z różnymi wersjami nazwy, takimi jak JavaScript, JScript i ECMAScript. Dzieje się tak dlatego, iż język rozwijany był początkowo wyłącznie dla przeglądarki Netscape Navigator. Jednakże stał on się na tyle popularny, że został standardowym językiem skryptowym wykorzystywanym przy tworzeniu stron internetowych. Microsoft, który rozwijając przeglądarkę Internet Explorer chciał uniknąć problemów licencyjnych zaczął prace nad własną implementacją figurującą pod nazwą JScript. Przez kolejne lata rozwoju tworzone były różniące się od siebie wersje języka, które doprowadziły do konieczności stworzenia wspólnego standardu. Ujednoczeniem zajęła się firma ECMA<sup>8</sup>, która stworzyła ECMAScript określający standard języka, a JavaScript i JScript to zaledwie implementacje owego standardu wykorzystywane w konkretnych przeglądarkach. Najczęściej jednak usłyszeć można o JavaScriptcie, który jest przyjęty jako najbardziej popularna i potoczna nazwa.<sup>9</sup>

---

<sup>7</sup> Marcin Lis, JavaScript. Ćwiczenia praktyczne. Wydanie III, Wydawnictwo Helion, 2013, str. 7-8

<sup>8</sup> ECMA – European Computer Manufacturers Association

<sup>9</sup> Marcin Lis, JavaScript. Ćwiczenia praktyczne. Wydanie III, Wydawnictwo Helion, 2013, str. 8

### 1.3.2. Działanie skryptów

Kod Javascript może być zaimplementowany zarówno bezpośrednio w kodzie HTML przy pomocy znaczników `<script>`, jak i z pliku zewnętrznego o rozszerzeniu `.js`. Skrypty mogą zostać wczytane w nagłówku kodu, lub między znacznikami `<body>`, wpływa to na sposób w jaki zachowa się przeglądarka podczas wyświetlania użytkownikowi witryny.

Przeglądarki po załadowaniu dokumentu HTML rozpoczynają analizę od jego początku wykonując przy tym każdą linię po kolei, ważne jest więc umiejscowienie kodu odpowiedzialnego za wczytanie ECMAScript. Nie miałyby to większego znaczenia, gdyby nie fakt, że podczas wykonywania skryptu wszystkie inne funkcje wczytywania zostają zatrzymane. Należy być uważnym przy umieszczaniu Javascriptu z racji, iż może to poważnie opóźnić wczytywanie się strony WWW. Najczęściej jednakże za wykonanie kodu odpowiedzialne są trochę wyżej opisane zdarzenia. Dlatego wykonanie skryptu jest niemożliwe przed załadowaniem przez przeglądarkę kluczowych elementów.

Niektóre przeglądarki mogą nie obsługiwać JavaScriptu, a zdecydowana większość dostępnych przeglądarek posiada możliwość wyłączenia obsługi języków skryptowych, co może to doprowadzić do niechcianych problemów z działaniem stron internetowych. W celu obsłużenia takich wyjątków istnieje możliwość zastosowania znaczników `<noscript>` pozwalających na wyświetlenie stosownej informacji.<sup>10</sup>

### 1.3.3. Przeciążenie strony

Każdy Web Developer powinien pamiętać o negatywnych skutkach używania skryptów, ponieważ nadużycie może prowadzić do bardzo dużych obciążeń i poważnego spowolnienia działania strony, co z kolei zniechęca użytkownika z korzystania z witryny. Istnieje kilka sposobów na załagodzenie tego problemu. Pierwszą z nich jest minimalistyczne podejście do tworzenia stron. Należy zastanowić się, czy dany skrypt jest potrzebny do prawidłowego działania. Oczywiście nie zawsze istnieje możliwość ograniczenia użycia JavaScript. W przypadku większych serwisów jest to bardzo często niemożliwe, co prowadzi do innych metod ograniczania obciążenia.

Ważne jest przemyślane usytuowanie skryptów w kodzie HTML. Z racji ułatwień związanych z edycją i większym porządkiem w plikach kody Javascript najczęściej zapisywane są w plikach zewnętrznych i wczytywane do dokumentu z zewnątrz. Metoda ta

---

<sup>10</sup> Marcin Lis, JavaScript. Ćwiczenia praktyczne. Wydanie III, Wydawnictwo Helion, 2013, str. 19

zmusza przeglądarkę do wczytywania dodatkowych plików, co spowalnia proces wczytywania witryny. Umieszczenie mniejszych skryptów bezpośrednio w kodzie HTML nie prowadzi do dużych problemów z przejrzystością, a zdecydowanie może skrócić czas oczekiwania gościa na pojawienie się w pełni załadowanej strony. W niektórych przypadkach można mówić wręcz o poprawieniu czystości związanej z posiadaniem kodu w jednym miejscu zamiast rozrzuconego po kilku plikach .js. Negatywnym skutkiem jest tutaj z kolei uniemożliwienie wykorzystania pamięci podręcznej – kod Javascript będzie musiał zostać wczytany za każdym razem wraz z kodem HTML i CSS. Dlatego wpisywanie większych skryptów w dokument HTML jest raczej kiepskim pomysłem i powinno się ograniczać jedynie do tych, których załadowanie przez przeglądarkę nie zajmie dużo czasu.<sup>11</sup>

Wszystkie skrypty, które mogą zostać wykonane dopiero po pojawieniu się w przeglądarce w pełni załadowanej strony można umieścić bezpośrednio przed znacznikiem zamykającym `</body>`. Odroczy to ich ładowanie aż do momentu pełnego wczytania strony, nie spowalniając tym samym tego procesu. Nieco bardziej skomplikowane jest użycie atrybutu **defer** elementu `<script>`. Użycie tego atrybutu w sekcji `<head>` w odnośniku do pliku zewnętrznego .js wysyła przeglądarce informacje, iż dany skrypt załadowany ma zostać po napotkaniu owego elementu, ale wykonany dopiero po wczytaniu całej strony WWW. Atrybut ten odnosi się tylko do skryptów zewnętrznych i zazwyczaj prostszym i bardziej przejrzystym sposobem jest jednak przeniesienie skryptów na koniec dokumentu HTML.

Wraz z HTML5 wszedł w życie atrybut **async** odpowiadający za ładowanie asynchroniczne, który również pozwala na lepszą kontrolę nad skryptami. Umieszczenie go w elemencie `<script>` pozwala przeglądarce na kontynuowanie wyświetlania strony. Ładowanie Javascriptu zacznie się dopiero po napotkaniu pierwszego odwołania do niego w sekcji `<head>`, lub `<body>`. Główną różnicą jaką powoduje **async** jest to, że przeglądarka nie wstrzyma ładowania i wyświetlania pozostałych elementów strony podczas ładowania skryptu. Zostanie on załadowany w tym samym czasie co reszta zasobów i wykonany, gdy zajdzie taka potrzeba. Atrybut ten można użyć jedynie w odwołaniu do skryptów zewnętrznych. W przypadku, kiedy odwołanie do skryptów znajduje się w nagłówku dokumentu, to w większości przypadków powinno się stosować atrybut **async**. Wyjątkiem jest sytuacja, w której wykonane musi zostać kilka skryptów i ważna jest kolejność w jakiej zostanie to wykonane.

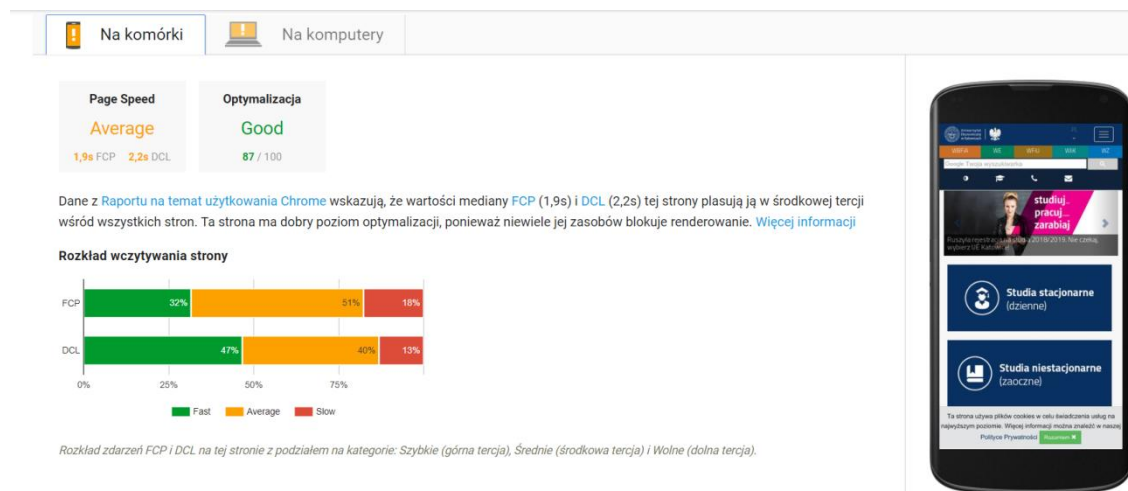
---

<sup>11</sup> Clarissa Peterson, Responsywne strony WWW dla każdego, Wydawnictwo Helion, 2015, str. 314-315



Rozwój HTML i CSS i wprowadzenie nowszych wersji tych języków pozwoliło na zastępowanie części skryptów kodem tych wyżej wymienionych. Największym plusem jest tutaj możliwość ograniczenia problemu związanego z nadmiernym obciążaniem strony WWW, co przekłada się na czas w jakim się ona załaduje. Nie jest to jednak sztywna zasada. Pomimo, że w większości przypadków jest to lepsze rozwiązanie, to czasem działa to również w drugą stronę i w efekcie Javascript jest bardziej dogodnym rozwiązaniem. Daje to jednak alternatywę podczas tworzenia stron. Oprócz eliminacji nadmiernego obciążenia – wykorzystanie kombinacji HTML i CSS pomaga w utrzymaniu porządku w plikach.<sup>12</sup> Jest to bardzo ważne rozwiązanie, które wykorzystywane jest przy tworzeniu dużych serwisów przez najlepszych Web Developerów. Należy pamiętać, że optymalizacja strony, jest również częścią tworzenia.

W sieci istnieje wiele narzędzi pozwalających na sprawdzenie poziomu optymalizacji. Wiele z nich wyszukuje również potencjalne błędy i możliwości poprawy. Jednym z takich narzędzi jest **PageSpeed Insight** oferujący darmową analizę wybranej witryny internetowej.



Rysunek 1 PageSpeed Insights

Źródło: PageSpeed Insights, [<https://developers.google.com/speed/pagespeed/insights>], dostęp: 04.04.2018

Rysunek 1 przedstawia narzędzie PageSpeed Insights wyświetlające analizę strony **ue.katowice.pl** w wersji mobilnej. Na górze widać ogólną ocenę strony, dając dosyć jasny pogląd bez zagłębiania się w szczegóły. Trochę niżej wyświetlone są dokładniejsze dane przedstawiające procentowy rozkład wczytywania strony, czyli dla jakiej części użytkowników strona wczytywała się następująco: **szybko, przeciętnie, wolno**. Analiza

<sup>12</sup> Clarissa Peterson, Responsywne strony WWW dla każdego, Wydawnictwo Helion, 2015, str. 314-316

opiera się na dwóch wartościach - FCP<sup>13</sup> i DCL<sup>14</sup>. Jak widać na obrazie – strona otrzymała ocenę dobrą co sugeruje, że optymalizacja strony jest na wysokim poziomie.

#### Statystyki strony

Według oszacowania PSI ta strona wymaga 2 przebiegów w obie strony blokujących renderowanie i około 55 zasobów (2,6 MB), by została załadowana. Strona, która ma wartość mediany, wymaga 4 przebiegów w obie strony blokujących renderowanie i około 75 zasobów (1 MB), by została załadowana. Strony wymagające mniejszej liczby bajtów i przebiegów w obie strony są szybsze.

#### Wskazówki optymalizacyjne

Skróć czas odpowiedzi serwera  
▶ Pokaż, jak to naprawić

Wyeliminuj blokujący renderowanie kod JavaScript i CSS z części strony widocznej na ekranie  
▶ Pokaż, jak to naprawić

Wykorzystaj pamięć podręczną przeglądarki  
▶ Pokaż, jak to naprawić

Zmniejsz JavaScript  
▶ Pokaż, jak to naprawić

Zoptymalizuj obrazy  
▶ Pokaż, jak to naprawić

#### Już zastosowane rozwiązania optymalizujące

▶ Pokaż szczegóły

### Rysunek 2 PageSpeed Insights - szczegóły

Źródło: PageSpeed Insights, [<https://developers.google.com/speed/pagespeed/insights>], dostęp: 05.04.2018

Oprócz wartości zgeneralizowanych przedstawionych na Rysunku 1 – PageSpeed Insights zwraca również dane bardziej szczegółowe, wraz ze wskazówkami optymalizacyjnymi dla Web Developerów. Wskazówki podzielone są na grupy i oprócz samego nakreślenia problemu, po rozwinięciu zobaczyć można listę konkretnych elementów nadających się do ewentualnej poprawy.

### 1.3.4. Biblioteki Javascript

Wraz z ciągłym rozwojem i latami królowania Javascriptu na rynku zaczęły powstawać biblioteki mające na celu ułatwienie pracy twórcom. Powszechnie wiadomo, że dużo łatwiej jest wykorzystać istniejący już fragment kodu niż poświęcać czas na pisanie go od podstaw. W tym celu tworzone są biblioteki Javascript, które tak jak w przypadku, np. CSS oferują gotowe rozwiązania w postaci funkcji, które możemy zastosować na stronie WWW. Nie tylko przyspiesza to proces tworzenia, lecz również w swoisty sposób ujednocila te same rozwiązania dla wielu serwisów ułatwiając tym samym pracę osobom, które z kodem źródłowym strony mają do czynienia po raz pierwszy. Do dyspozycji Web Developerów stworzonych zostało wiele bibliotek, jednakże nie wszystkie wykorzystywane są na tę samą

<sup>13</sup> First Contentful Paint – czas po jakim użytkownik otrzyma pierwszą wizualną odpowiedź

<sup>14</sup> DOM Content Loaded – czas po jakim dokument HTML został w pełni wczytany

skalę i do tych samych celów. Naturalnie kilka z nich wyprzedza konkurencję zarówno w funkcjonalności jak i popularności. Oto przykłady popularnych bibliotek:

- jQuery
- Lodash
- Underscore.js
- AngularJS
- Backbone.js
- Raphaël
- React.js
- Modernizr
- Polymer

Wybór jest ogromny, a to tylko kilka przykładów dostępnych bibliotek. Każda z nich oczywiście wykorzystywana jest na inną skalę i do innych zadań, nie mniej jednak wszystkie oferują twórcom interesujące rozwiązania, które wykorzystać można w różnoraki sposób. Biblioteki można podzielić na kilka typów.

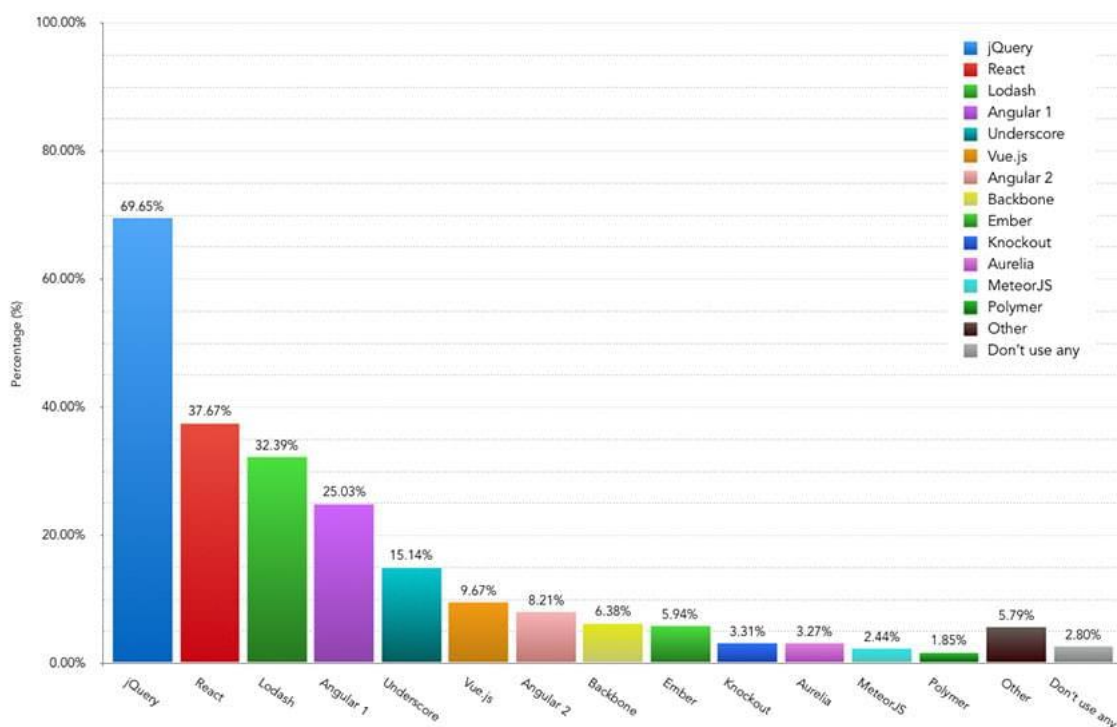
- Biblioteki JS wspierające przeglądarki
- Biblioteki JS architektury aplikacji
- Mini biblioteki i mikrobiblioteki

Z całą pewnością należy wyróżnić **jQuery**, która na rok 2018 jest największą i najczęściej wybieraną biblioteką JavaScript. jQuery zalicza się do grupy bibliotek wspierających korzystanie z funkcjonalności przeglądarek pozwalając na uzyskanie zamierzonego efektu mniejszą ilością kodu i dbając przy tym o kompatybilność między przeglądarkami. Biblioteka ta znacznie ułatwia korzystanie z JavaScriptu kosztem niewielkiego spadku wydajności strony, nie jest to jednak różnica na tyle duża, by zrezygnować z dogodności jakie oferuje jQuery.

**AngularJS** jest otwartym frameworkiem wspieranym przez Google. Zaliczany do drugiej grupy, czyli bibliotek architektury aplikacji jest odpowiedzialny za tworzenie i rozwój aplikacji internetowych. Główną cechą tych frameworków jest ułatwienie zarządzania długimi kodami aplikacji. W przypadku AngularJS jest to wdrożenie wzorca Model-View-Controller odpowiedzialnego za poprawienie organizacji struktury aplikacji posiadających Graficzny Interfejs Użytkownika.

**Raphaël** to przedstawiciel tych najmniejszych bibliotek. Jest to mały framework JavaScript wydany w 2008 roku, służący do rysowania grafiki wektorowej SVG<sup>15</sup>. Biblioteka wspiera przeglądarki takie jak Google Chrome, Firefox, Safari, Opera i Internet Explorer. Pomimo rozmiaru biblioteki jest ona całkiem popularna i wykorzystywana na stronach wielkich koncernów.

Badania Front-End Tooling Trends przeprowadzane przez człowieka o imieniu Ashley Nolan potwierdzają opisaną powyżej teorię. W wyniku ankiety przeprowadzonej wśród 5254 Developerów wyniki na rok 2017 dosyć jasno wskazywały przodujące Frameworki. Ankieta została udostępniona globalnie, tylko w języku angielskim, dlatego wziąć pod uwagę należy fakt, że skierowana była do osób porozumiewających się w owym języku na wystarczająco dobrym poziomie, a wyniki mogą różnić się względem krajów nieangielskojęzycznych. W ankiecie udział brać mogły osoby zarówno zaczynające przygodę z językiem JavaScript, jak i ci bardziej doświadczeni. Jednakże tylko 5% osób przyznało się do posiadania niespełna rocznego stażu.



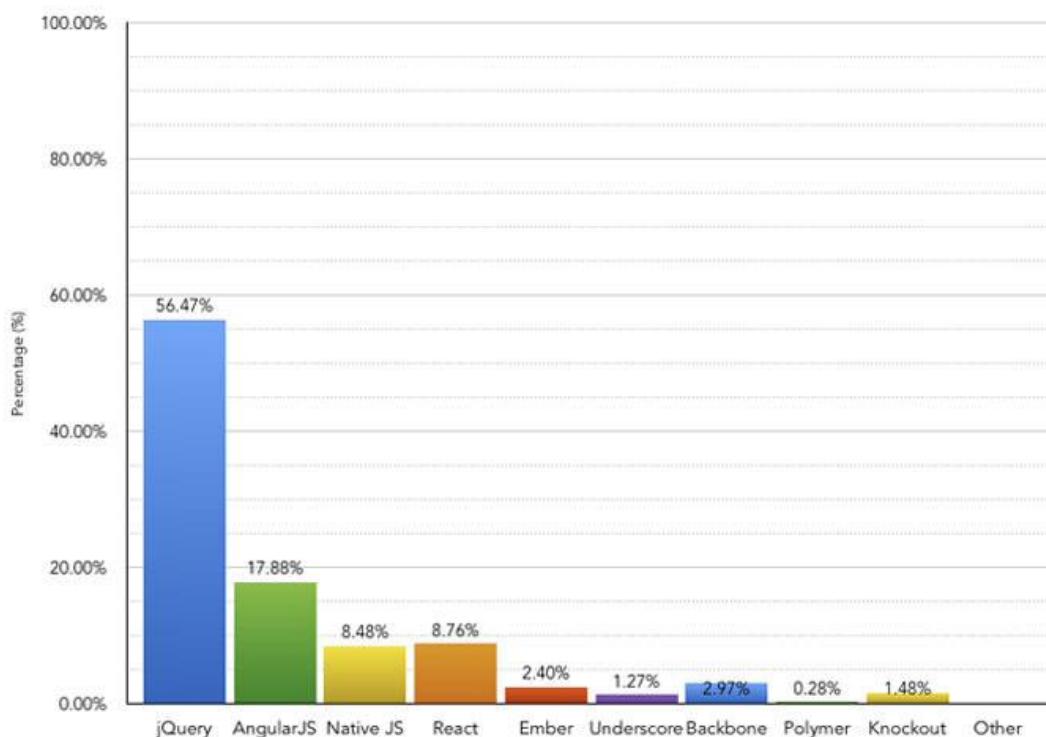
Rysunek 3 Front-End Tooling Trends 2017 - Najczęściej wykorzystywane Frameworki JS

Źródło: [sitepoint.com](https://www.sitepoint.com/front-end-tooling-trends-2017/), [https://www.sitepoint.com/front-end-tooling-trends-2017/], dostęp: 19.04.2018

<sup>15</sup> Scalable Vector Graphics

Web Developerzy zapytani o biblioteki JS jakie wykorzystywali w 2017 roku wskazywali kilka szczególnych nazw. Aż 70% badanych osób wskazało, że używało w ostatnim czasie jQuery czyniąc go najczęściej wybieranym frameworkiem wśród Web Developerów – dodatkowo około 99% badanych potwierdziło, iż miało w swojej karierze choć raz styczność z tą biblioteką. Trochę mniej, bo niespełna 38% wykorzystywało w swoich ostatnich projektach React.js. Częstym wyborem były również Lodash, Angular, czy Underscore.js. Mniej niż 3% ankietowanych nie wspomagało się żadnym frameworkiem. Na liście najczęściej wykorzystywanych nie ma natomiast nazw mniejszych, lecz nadal popularnych bibliotek takich jak np. Raphaël. Spowodowane jest to faktem, iż pomimo popularności i bycia oczywistym wyborem w danych sytuacjach – nie są to rozwiązania dla zastosowań ogólnych, takich jak np. zwiększenie przejrzystości kodu.

Względem roku 2016 nie zanotowano wielkich zmian. Widać jedynie delikatnie rosnące zainteresowanie nieco nowszymi bibliotekami kosztem spadającego zainteresowania jQuery. Jednak różnice względem roku 2015 są ogromne.



**Rysunek 4 Front-End Tooling Trends 2015 - Najczęściej wykorzystywane Frameworki JS**

Źródło: ashleynolan, [<https://ashleynolan.co.uk/blog/frontend-tooling-survey-2015-results/>], dostęp: 19.04.2018

W 2015 roku Frameworki JS wykorzystywane były dużo rzadziej niż w przypadku kolejnych lat. Oprócz dobrego wyniku jQuery i przebijających się AngularJS, Native JS, oraz

React.js – frameworki nie były tak popularne jak, np. rok później. Pomimo szybkiego rozwoju - developerzy ograniczali się głównie do kilku sprawdzonych nazw. W samej ankiecie wzięło początkowo udział 648 osób, a na przestrzeni roku liczba wzrosła ostatecznie do 2028. Pokazuje to rosnące zainteresowanie bibliotekami JS, co w roku 2016 widoczne już było na opublikowanych wynikach – które zbliżone były do tych z 2017.

### **1.3.5.JSON**

JavaScript Object Notation, w skrócie JSON to format tekstowy bazujący na podzbiorze języka JavaScript służący do przekazywania danych do aplikacji opartych o AJAX<sup>16</sup>. Jest jednakże całkowicie niezależny od języków programowania. Używa konwencji wykorzystywanych przez języki takie jak, np. C++, C#, Java, czy właśnie Javascript.

Wraz ze wszystkimi zmianami i udogodnieniami jakie dotykały środowisko witryn internetowych, developerzy zmuszeni byli znaleźć rozwiązanie na odciążenie stron podczas interakcji z użytkownikiem. Wczesne lata 2000 przyniosły duże zmiany, dotychczas wszelkie interakcje użytkownika z serwerem wymuszały na przeglądarce pełne przeładowanie dokumentu HTML. Jako, że technologia w jakiej witryny WWW były tworzone w szybkim tempie szła do przodu, a ponowne wczytanie strony zaczynało być czymś niechcianym, programiści byli zmuszeni wymyślić inny sposób na wczytywanie danych. W ten sposób powstał format XML, który jednakże okazał się problematyczny w obsłudze za pomocą JavaScript, a ostatecznie na jego podobieństwo stworzono format JSON. Ten współpracował już z językiem JavaScript bez zarzutów i był co raz częstszym wyborem wśród Web Developerów.

### **1.4. PHP**

PHP to Open Source'owy język skryptowy zapoczątkowany w 1994 roku przez jednego człowieka - Rasmusa Lerdorfa. Sam język od czasu powstania uległ trzem poważniejszym modyfikacjom, które zmieniły go w produkt wykorzystywany przez developerów na całym świecie. Skrypty stworzone za pomocą PHP są wykorzystywane po stronie serwera WWW. W grudniu 2017 roku aż 80% stron dostępnych w sieci używa PHP, jednak pomimo dostępnego już PHP7 – około 90% tych stron nadal działa na starszej wersji, PHP5. Język wykorzystywany jest nie tylko do tworzenia stron internetowych, za jego pomocą można

---

<sup>16</sup> asynchroniczny JavaScript i XML

pisać również programy pracujące w trybie graficznym. Korzystanie w PHP ma wiele zalet, najczęściej mówi się jednak o tych:<sup>17</sup>

- wysoka wydajność,
- niski koszt,
- łatwość nauki,
- integracja z bazami danych,
- dostępność kodu źródłowego,
- wbudowane biblioteki,
- duża ilość dostępnych frameworków,
- dostępność wsparcia.

Jeden serwer umożliwia obsługę dużej ilości odwiedzin, co czyni PHP bardzo wydajnym wyborem. Z kolei w przypadku większych stron WWW i większej ilości wykorzystywanych serwerów, ograniczenia praktycznie nie istnieją.

Z powodu łatwości w integracji z bazami danych jest on bardzo często łączony z MySQL. PHP pozwala na bezpośrednią obsługę połączeń z bazami danych. Właściwość ta dotyczy dużej ilości systemów, m.in. PostgreSQL, mSQL, dbm, filePro, Oracle, czy Informix. Z tego powodu jest to częsty wybór wśród programistów, którzy opierają ich projekty o bazy danych.

Oprócz dostępu do wsparcia technicznego i dodatkowego oprogramowania, duże zainteresowanie językiem doprowadziło do powstania sporej społeczności programistów tworzących w PHP. W roku 2017 przeszukując sieć trudno nie natknąć się na liczne poradniki, czy gotowe rozwiązania stworzone za pomocą tego języka. Bardzo dużym atutem jest również możliwość modyfikacji kodu źródłowego.

Kod PHP osadza się w dokumencie HTML za pomocą jednego z czterech znaczników:

- `<? ?>`
- `<?php ?>`,
- `<script language="php"></script>`
- `<% %>`

---

<sup>17</sup> L. Welling, PHP i MySQL. Tworzenie stron WWW. Vademecum profesjonalisty Wydanie III, Wyd. Helion, str. 25-26

Plik zapisany musi być pod rozszerzeniem .php. Nie stwarza to żadnego problemu podczas wyświetlania kodu HTML, ma to jednak krytyczne znaczenie dla PHP. Przeglądarka po otrzymaniu pliku zapisanego jako .php, będzie wiedziała, że oprócz standardowego wyświetlenia kodu HTML, plik musi zostać dodatkowo przetworzony przez parser PHP.

W pierwszym rozdziale autor pracy przedstawił cztery najpopularniejsze języki wykorzystywane w procesie tworzenia nowoczesnych stron internetowych. Dodatkowo zaprezentowane zostały gotowe, tworzone przez społeczność biblioteki. Pomimo, że strony tworzone są za pomocą opisanych w tej części języków, to do prawidłowego działania strony potrzebna jest również integracja z innymi technologiami, które omówione zostaną w rozdziale drugim.



## **2. Narzędzia i systemy wspierające tworzenie stron internetowych**

Rozdział drugi poświęcony będzie omówieniu dostępnych i popularnych narzędzi i systemów wspomagających tworzenie witryn WWW. Mowa tu głównie o Systemach Zarządzania Treścią, przydatnych narzędziach i wtyczkach pozwalających Web Developerom na wydajniejszą pracę i lepsze kontrolowanie stworzonych stron. Oprócz CMS, bardzo ważnym elementem jest wybór odpowiedniego hostingu i domeny, na których oparty będzie cały projekt. Przedstawione i zaprezentowane zostaną różne dostępne rozwiązania.

### **2.1. Hosting i domena**

Bardzo ważnym i podstawowym elementem tworzenia witryny internetowej jest wybór hostingu, na którym będzie stała strona WWW. W sieci dostępnych jest dużo różnych hostingów i w zależności od indywidualnych potrzeb należy zdecydować się na jeden z nich. Pierwsze pytanie jakie zadać musi sobie każdy administrator strony, to czy chce bazować na serwerze darmowym, czy płatnym. Ten wybór niesie za sobą wiele konsekwencji i korzyści, lecz istotnie wpływa na działanie witryny.

Hosting polega na udostępnianiu miejsca na serwerze dla stron internetowych. Serwer jest nieustannie podłączony do Internetu, a umieszczone na nim witryny WWW są cały czas dostępne w sieci. Na terenie całego świata działają firmy zajmujące się hostingiem, przez co użytkownicy nie są zmuszeni udostępniać stron internetowych ze swoich urządzeń.

#### **2.1.1. Darmowy hosting**

Największym i jedynym plusem darmowego hostingu jest to, że nie trzeba za niego płacić. Nadaje się on więc do wszelkiego rodzaju nauki, testowania, a niektóre zaczynające dopiero firmy decydują się na wybór tej opcji z racji braku dodatkowych opłat. Wybór ten przynosi jednak konsekwencje, najczęściej są to:

- duże obciążenia podczas administrowania stroną,
- wolno działająca strona,
- niestabilnie działająca strona
- mocno ograniczone miejsce na serwerze
- prawdopodobieństwo usunięcia strony
- słaba nazwa domeny
- ograniczony dostęp do FTP

- ograniczony transfer
- reklamy

Pomimo, że wybór darmowych serwerów nie jest aż tak duży, to można wybierać wśród kilku przodujących w Polsce serwisów. Darmowe hostingi bardzo często oferują możliwość wykupienia subskrypcji dającej dodatkowe przywileje i udogodnienia. Przed tą decyzją należy jednak się zastanowić, czy lepszym rozwiązaniem nie okazałoby się wykupienie płatnego serwera. Oprócz polskich hostingów istnieją oczywiście zagraniczne alternatywy. Często ich oferta okazuje się bogatsza, a nawet oferują dodatkowe usługi, które nie są obsługiwane w przypadku polskich odpowiedników.

### **2.1.2. Płatny hosting**

Profesjonalne i duże strony WWW opierają się jednak głównie o płatne serwery. Ceny wykupienia hostingu nie są wysokie, a jakość usług przez nie oferowanych jest znacznie lepsza niż w przypadku postawienia strony używając darmowej opcji. Odwrotnie względem tych, za które płacić nie trzeba, płatne hostingi charakteryzują się dużą ilością pozytywnych cech. Tak jak w przypadku minusów związanych z serwerami darmowymi – można wymienić kilka szczególnych cech pozytywnych.

- stabilnie działająca strona,
- duża ilość miejsca na serwerze,
- nieograniczony, lub duży transfer,
- dogodny dostęp do serwera FTP,
- brak reklam,
- domena .pl,
- duży zakres usług
- sprawnie działająca pomoc techniczna

Wybór wśród płatnych hostingów jest dużo większy niż w przypadku darmowej alternatywy. Ceny różnią się w zależności od konkretnej firmy hostingowej i oferowanych usług. Jakość serwera może różnić się w zależności od wyboru, jednakże każdy z nich oferuje stabilne środowisko, które bez wątpienia zdaje lepiej egzamin niż to otrzymane podczas używania darmowego serwera. W samej Polsce wybór jest dosyć spory, a oczywiście tak jak w poprzednim przypadku, można dodatkowo przejrzeć i wykorzystać usługi zza granicy. A to kilka przykładowych polskich hostingów:

- seohost.pl,
- dhosting.pl,
- home.pl,
- ehost.pl,
- hekkko.pl,
- kei.pl,
- nazwa.pl,
- ovh.pl

### 2.1.3. Najważniejsze cechy

Przy wyborze hostingu należy zwrócić uwagę na parametry i usługi przez niego oferowane. Część z nich ma większy i bardziej bezpośredni wpływ na prosperowanie strony WWW niż pozostałe. Zastanowić się trzeba, które z danych parametrów są kluczowe. Funkcjonalność jaką spełniać ma witryna określa część najważniejszych cech jakie powinno brać się pod uwagę przy wyborze hostingu. Dla sklepu internetowego będą to inne parametry, niż dla strony czysto informacyjnej, lub bloga. Druga część jest bardziej uniwersalna i niezależnie od celu jaki spełniać ma strona jest dla wszystkich rodzajów tak samo istotna. Warto spojrzeć również na to czy oferta uwzględnia ewentualną możliwość poszerzenia usług, ponieważ bardzo często wraz ze wzrostem popularności strony internetowej pojawia się potrzeba zwiększenia możliwości uzgodnionych początkowo z firmą hostingową. Pomimo, że bardzo rzadko zdarza się, by firma nie oferowała takiej możliwości, to ważne jest by mieć pewność, że w przyszłości nie będzie z tym problemu.

**Pojemność konta** jest bardzo ważną cechą. Wszystkie treści, w tym zdjęcia jakie pojawiają się na stronie WWW przetrzymywane są na serwerze. Przy wyborze serwera z małą ilością miejsca na dysku, ilość materiałów jakie mogą pojawić się na stronie jest znacznie ograniczona. Mała pojemność na dysku dotyka szczególnie wszelakiego rodzaju blogów, czy stron informacyjnych, na których każdego dnia pojawiają się nowe posty, wraz ze zdjęciami. Po czasie prowadzenia takiej strony przeznaczone na dysku miejsce może być zapełnione. Oprócz ograniczenia pojemności, firmy hostingowe nakładają czasami ograniczenia związane z **maksymalnym rozmiarem plików**, nie pozwalając tym samym na umieszczanie na stronie plików przekraczających konkretną wartość. Większość obrazów jakie pojawiają się na stronach i tak przed wgraniem zostaje zoptymalizowana, to

w niektórych przypadkach zbyt ograniczony rozmiar plików może okazać się poważnym problemem.

**Limit transferu**, czyli bandwidth jest kolejną cechą, która ma ogromne znaczenie przy wyborze odpowiedniej firmy hostingowej. Limit transferu odnosi się do liczby danych jakie mogą zostać przesłane pomiędzy użytkownikami, a stroną internetową. Licznik resetowany jest najczęściej na początku każdego miesiąca i przez okres około 30 dni nalicza wszystkie przesłane dane. W przypadku większych serwisów z dużą liczbą odwiedzin i materiałów, zbyt niski limit może doprowadzić to znaczącego obniżenia wydajności, a nawet do wyłączenia witryny. Pozytywnym scenariuszem w tym wypadku jest dodatkowe naliczanie danych, które pozostają następnie do uregulowania. O tyle co w przypadku nowo powstałych witryn z niską ilością odwiedzin przekroczenie limitu nie jest zagrożeniem, to administratorzy tych nieco bardziej popularne muszą uwzględnić ten parametr przy wyborze oferty.

**Czas działania**, czyli uptime to jeden z najważniejszych parametrów. W sieci standardem jest przyjęcie, że strona będzie zawsze online. Pomimo, że głównym celem serwera jest utrzymywanie strony we właśnie takim stanie, a jednym z podstawowych założeń jest utrzymywanie czasu działania na poziomie 100%, czyli 24 godziny na dobę, to należy pamiętać, że serwer to też maszyna. Problemy mogą pojawić się nawet przy wyspecjalizowanym sprzęcie, utrzymanie maksymalnego czasu działania jest więc niemożliwe. Jako nieoficjalną granicę przyjęty został próg 99,5%. W przypadku Internetu wszystkie poślizgnięcia związane ze zbyt długim czasem offline mogą mieć katastrofalne efekty. Tak prezentuje się kilka obliczeń z tym związanych.<sup>18</sup>

- 99% - offline – 3 dni 16 godzin/rocznie (7 godzin 15 minut/miesięcznie)
- 99,5% – offline - 1 dzień 20 godzin/rocznie (3 godziny 40 minut/miesięcznie)
- 99,9% - offline – 8 godzin 45 minut/ rocznie (43 minuty/miesięcznie)

**Hardware** - sprzęt używany przez serwer mają wpływ na szybkość działania strony WWW. Kluczowym elementem jest tutaj pamięć RAM, procesor, oraz dysk serwera. Pamięć RAM i CPU odpowiedzialne są za płynność działania witryny i obsługę ruchu. Dlatego mocniejszy sprzęt pozwoli na lepszą obsługę większych serwisów z dużym ruchem. Dysk również wpływa na szybkość przetwarzania danych, w tym przypadku najbardziej liczy się

---

<sup>18</sup> Uptime Calculator [online] [ <https://hosting.review/uptime-calculator/>] dostęp: 29.05.2018r.

rodzaj dysku jaki używa serwer, z racji iż SSD<sup>19</sup> jest w stanie przetwarzać informacje znacznie szybciej od klasycznego dysku HDD<sup>20</sup>. Informacje odnośnie komponentów serwera są niestety bardzo rzadko podawane razem z pozostałymi parametrami. Jest to jednak ważny parametr i jeśli istnieje możliwość to warto brać go pod uwagę.

**Bazy danych** - bardzo ważnym elementem nowoczesnych stron internetowych jest baza danych. Współdziałanie witryn z nimi pozwala na rozwinięcie funkcjonalności poprzez zbieranie niezbędnych do funkcjonowania danych. Jest to szczególnie ważny aspekt podczas tworzenia między innymi sklepów internetowych. Na nich opierane są bazy produktów, dane klientów, czy też dokonywane transakcje. Możliwość zastosowania ich znacznie rozwija funkcjonalność potencjalnego serwisu. Najczęściej wybieraną opcją wśród baz danych są bazy MySQL, które cechują się wysoką szybkością działania. Dodatkowym atutem tego wyboru jest to, iż bazy te są one dostępne za darmo, a prawie każdy panel kontrolny umożliwia tworzenie i zarządzanie MySQL. Istnieją też inne alternatywy, wśród nich najczęściej wybór pada na:

- Access - ten typ rekomendowany jest głównie dla bardzo małych systemów. W porównaniu do konkurencji jest to raczej wolno działający typ baz danych.
- MsSQL - baza danych oferowana przez Microsoft. Szybko działająca alternatywa, która jednakże wiąże się z kosztami. Z tego tytułu wybierana jest głównie w przypadku dużych i popularnych stron internetowych.

Wybierając hosting najważniejszymi punktami związanymi z tematem baz danych, na które należy zwrócić uwagę są:

- Dopuszczalna wielkość. Czy proponowana maksymalna waga bazy wystarczy, by pokryć potrzeby związane z witryną WWW.
- Jaka jest maksymalna dopuszczalna ilość baz danych i czy istnieje możliwość poszerzenia tej liczby.
- Atutem jest dostęp do graficznego narzędzia zarządzania bazami danych.

**Software** - obsługa języków programowania jest ważnym elementem w przypadku serwisów internetowych. Problem ten dotyczy szczególnie tych bardziej rozwiniętych, jednakże nawet w przypadku braku planów związanych z ich wykorzystywaniem warto

---

<sup>19</sup> Solid-State Drive

<sup>20</sup> Hard Disk Drive

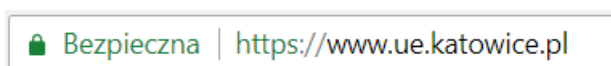
zwrócić uwagę na ten parametr. Pomimo, że w roku 2018 praktycznie każda firma hostingowa oferuje już w zakresie usług ich działanie, to należy upewnić się, iż w ofercie znajdują się języki niezbędne do poprawnego działania witryny. Najwięcej problemów związanych z obsługą języków pojawia się w przypadku ich wersji.

**Backup** – wszystko co znajduje się w sieci, a szczególnie złożone struktury narażone są na uszkodzenia i stratę danych. W przypadku stron internetowych, które przetrzymywane są na serwerach bardzo ważne jest utrzymanie bezpieczeństwa przyjmowanych informacji. Oferty co raz częściej obejmują tworzenie i przetrzymywanie kopii zapasowej. Najczęściej jest to okres 3 dni, do tygodnia. Dlatego w przypadku awarii strony nie ma większego problemu z odzyskaniem wszystkich danych. W przypadku serwisów obsługujących duży ruch prowadzenie kopii zapasowej jest kluczowym parametrem przy wyborze oferty hostingowej.<sup>21</sup>

#### 2.1.4. Pozostałe parametry

Oprócz parametrów kluczowych do poprawnego i bezpiecznego działania stron są też cechy, które mogą mieć wielkie znaczenie w konkretnych przypadkach, lecz nie wpływają bezpośrednio na działanie zainstalowanych na serwerze witryn.

**Domena** – bardzo ważnym, lecz nie wpływającym na działanie strony jest odpowiednia nazwa domeny. Zarejestrowanie domeny pozwala na osiągnięcie atrakcyjnego linku. To między innymi za jej pomocą użytkownicy mogą połączyć się z witryną WWW i bez wątplenia jest elementem, który często zapada w pamięć klientów.



Rysunek 5 Nazwa domeny

Źródło: Opracowanie własne

Przyjazna i zrozumiała nazwa jest dużo łatwiejsza do zapamiętania i wprowadzenia niż adres IP, który w przypadku braku domeny znalazłby się w polu adresowym.

Pomimo, że nie ma to znaczenia dla konkretnej strony internetowej, to z punktu widzenia administratora ważna jest maksymalna możliwa liczba zarejestrowanych domen. Ma to znaczenie w przypadku, gdy osoba wykupująca miejsce na serwerze planuje w przyszłości

---

<sup>21</sup> Web Guide [online] [http://www.web-guide.co.il/pages/hosting-parameters.php?lang=en#Databases-MySQL-Access-MsSQL/] dostęp: 02.06.2018r.

prowadzić większą ilość serwisów sieciowych. W tym przypadku jednak ważne jest dostosowanie pozostałych parametrów pod ewentualne potrzeby.

**Konto FTP** – w celu zarządzania plikami na serwerze potrzebny jest dostęp do serwera. Najbardziej popularnym wyborem jest protokół komunikacyjny FTP. W przypadku stron, przy których pracuje większa liczba ludzi ważne jest upewnienie się, iż wybrana oferta umożliwi jednoczesny dostęp do serwera większej ilości pracowników.

**Email** - w roku 2018 email jest istotnym elementem w działaniu przedsiębiorstwa. Adres email powiązany z domeną strony komunikuje, iż użytkownik ma do czynienia z poważną firmą, dodatkowo przyczynia się to do budowania świadomości marki.

- Maksymalna liczba kont email powiązanych z domeną.
- Maksymalna wielkość kont email.
- Łatwość tworzenia nowych kont

To trzy najważniejsze czynniki związane z adresami email jakie mają znaczenie przy wyborze serwera hostingowego. Oprócz konkretnych wartości związanych z liczbą i wielkością kont, ważna jest dostępność graficznego panelu umożliwiającego tworzenie i zarządzanie kontami. W ten sposób administrator ma pełny dostęp do tych zasobów bez potrzeby kontaktowania się z firmą zajmującą się udostępnianiem strony internetowej.<sup>22</sup>

**Instalator CMS** – Podczas tworzenia i zarządzania stronami internetowymi bardzo często wybór pada na systemy zarządzania treścią, które pozwalają na łatwiejszy dostęp do poszczególnych fragmentów strony nawet bez znajomości języków w jakich strona została napisana. Firmy hostingowe często umożliwiają sprawną instalację wybranego systemu poszerzając ofertę o instalator CMS. W przypadku zamiaru korzystania, z któregoś z dostępnych systemów, jest to coś na co warto zwrócić uwagę.

**Lokalizacja serwera** – jest to również ważny parametr podczas wyboru usług hostingowych. Miejsce w jakim znajduje się serwer ma wpływ na opóźnienie podczas ładowania witryny WWW u osoby odwiedzającej. Dlatego też ważna jest ocena miejsca z jakiego łączyć ze stroną będzie się główna część użytkowników. W roku 2018 problem ze zbyt długim czasem oczekiwania na odpowiedź serwera związaną z długim czasem podróży pakietów nie jest już aż tak odczuwalny. Rozwój technologii przyniósł już rozwiązania

---

<sup>22</sup> Web Guide [online] [<http://www.web-guide.co.il/pages/hosting-parameters.php?lang=en#Databases-MySQL-Access-MsSQL/>] dostęp: 02.06.2018r.

pomagające załagodzić czasy oczekiwania. Jednakże w celu zminimalizowania opóźnienia mimo wszystko warto mieć na lokalizację na uwadze.

### 2.1.5. Typy hostingów

Usługa hostingowa to nic innego jak wyspecjalizowany komputer przechowujący pliki strony internetowej i zajmujący się obsługą ządań osób odwiedzających. Wybrać jednak można w jaki sposób ma owy serwer działać bazując na tym w jaki sposób zasoby są rozlokowane u dostawcy. Bardzo ważne jest rozróżnienie typów hostingu, ponieważ każdy z nich posiada swoje założenia i spełnia inne wymagania klienta.<sup>23</sup>

	<b>Współdzielony</b>	<b>VPS</b>	<b>Chmura</b>	<b>Dedykowany</b>
<b>Rozlokowanie zasobów</b>	Serwer podzielony na kilku klientów	Wydzielone miejsce na serwerze	Zasoby podzielone na kilka serwerów	Cały serwer dla klienta
<b>Stopień kontroli</b>	Predefiniowane cechy w panelu kontrolnym	Prawie jak własny serwer	Pełna kontrola nad ustawieniami	Pełna kontrola nad serwerem
<b>Klientów na serwer</b>	Dużo	Kilku	Kilku	Jeden
<b>Łatwość obsługi</b>	Łatwo	Łatwo/Średnio	Średnio/Trudno	Trudno
<b>Skalowalność</b>	Średnia/Niska	Wysoka/Średnia	Wysoka	Niska
<b>Grupa cenowa</b>	Tanio	Tanio	Umiarkowanie	Drogo

Rysunek 6 Tabela: Typy hostingu

Źródło: satoristudio, [https://satoristudio.net/choosing-hosting/dostęp: 2018-06-02]

**Hosting współdzielony** – najbardziej popularny i najczęściej wykorzystywany rodzaj hostingu. Każda maszyna fizyczna odpowiedzialna jest za przechowywanie i obsługę stron wielu klientów w tym samym czasie. Wszystko rozgrywa się na zasadzie rywalizacji dziesiątek stron o zasoby serwera. W tym o pojemność przechowywania na dysku, moc

<sup>23</sup> Understanding Web Hosting: How It Works and How to Choose One [online] [https://satoristudio.net/choosing-hosting/] dostęp: 02.06.2018



przetwarzania i o transfer. Prowadzi to do tego, że hosting współdzielony w porównaniu do innych technologii jest raczej słaby i mało elastyczny. Najczęściej jednak jest to w zupełności wystarczająca i z całą pewnością najtańsza opcja.

Kolejną atrakcyjną cechą oprócz niskiej ceny jest łatwość obsługi. Bardzo często można spotkać się tu z panelami kontrolnymi zawierającymi dodatkowe predefiniowane cechy pozwalającymi na łatwe i szybkie skonfigurowanie hostingu pod własne potrzeby. Wszystko ma jednak swoje wady i w tym wypadku są to głównie:

- Relatywnie długi czas ładowania względem alternatyw.
- Serwer może zostać przeciążony przez dzielące go strony.
- Spadek wydajności wraz ze wzrostem popularności strony.
- Ograniczone opcje customizacji.

**VPS** – Virtual Private Hosting – pod kątem alokacji zasobów działa podobnie do hostingu dzielonego. Każdy klient otrzymuje miejsce na serwerze, który obsługuje większą liczbę stron. Jednak w przypadku VPS każdy użytkownik otrzymuje swoje wydzielone miejsce o stałych parametrach bez względu czy są one wykorzystywane w pełni, czy też nie. Ten typ hostingu charakteryzuje się nieco większym zaawansowaniem w konfiguracji. Dzieje się tak dlatego, że osoba otrzymuje swoją część serwera, która zachowuje się zblizenie do faktycznej maszyny hostingowej, otwierając przy tym bardziej specjalistyczne możliwości zmiany ustawień. O ile może sprawiać to problemy osobom, które nie znają się na konfiguracji serwerów, to należy mieć na uwadze, że to nadal wirtualna przestrzeń, która mimo wszystko zachowywać będzie się inaczej niż fizyczny serwer dedykowany.

**Managed VPS Hosting** – to częściowo administrowana przez firmę hostingową alternatywa standardowego VPS skierowana do osób, które chcą skupić się na innych aspektach prowadzenia biznesu, niż zarządzanie serwerem. Jest to również rozwiązanie dla osób mających trudności w konfiguracji. W przypadku tego rodzaju wszystkie czasochłonne i trudne prace są wykonywane przez ludzi z serwisu hostingowego. Opcja ta przybliża administrację do poziomu zbliżonego do tego znanego z hostingu dzielonego, pozostawiając przy tym luksus posiadania wydzielonych zasobów.<sup>24</sup>

---

<sup>24</sup> Paul Mahony, Managed VPS service: Is it like having your own server administrator? [online] [<https://hosting.review/tips-tools/managed-vps-service/>] dostęp: 03.06.2018

**Hosting w chmurze** – Cloud hosting– to sieć serwerów leżących w różnych lokalizacjach, które dzielą dane i zasoby pomiędzy sobą używając zaszyfrowanego połączenia. Pozwala to na zredukowanie czasu ładowania strony po stronie użytkowników i poprawienie wydajności. Wraz ze wzrostem potrzeb, klient posiada możliwość łatwego rozwoju potrzebnych parametrów, zwiększając przy tym wydajność w kilku kliknięciach. Jest to bardzo szybko rozwijający się typ hostingu. Hosting w chmurze bazując na kilku różnych serwerach pozwala na przyjmowanie dużej ilości ruchu nie spowalniając przy tym strony internetowej. Dzieje się tak ponieważ żądania osób odwiedzających dzielą się na co raz to więcej serwerów, nie obciążając tym samym głównej jednostki.

„Gwarantowany wysoki czas działania” to podstawa prowadzenia biznesu w sieci. W przypadku korzystania z jednego serwera wszystkie problemy związane ze stabilnością są w stanie wyłączyć kompletnie stronę WWW. W przypadku korzystania z kilku maszyn, awaria na jednym z serwerów nie jest w stanie wyłączyć kompletnie strony internetowej. To samo dotyczy wszystkich innych trudności z jakimi mogą borykać się firmy hostingowe. Hakerstwo, przeciążenie, czy problemy z zasilaniem nie stanowią zagrożenia ponieważ dane przechowywane są jednocześnie na wielu rozproszonych serwerach. Cloud hosting nie wiąże się z żadnym opóźnieniem podczas ładowania stron. Całością zarządzać można z poziomu interfejsu opartego na sieci, lub za pomocą API. Należy jednak pamiętać, iż pliki nie znajdują się w jednym miejscu, z tego tytułu też tracimy niektóre możliwości konfiguracji, które standardowe są dla fizycznych serwerów.<sup>25</sup>

**Hosting dedykowany** – Przeciwnością dla tego wyżej opisanego jest serwer dedykowany. Typ ten polega na rezerwacji całego serwera hostingowego pod potrzeby jednego klienta. W ten sposób osoba zarządzająca stroną na pełny dostęp do konfiguracji maszyny. Wszystkie zasoby oferowane przez sprzęt są do dyspozycji osoby, która wykupiła do niej dostęp. Jest to najbardziej zaawansowana z opcji, jako że w odróżnieniu od pozostałych alternatyw konfiguracja obejmuje całą jednostkę, dając przy tym największą swobodę administratorowi strony. W związku z trudnościami jakie może powodować konfiguracja, wiele serwisów hostingowych tak samo jak w przypadku VPS oferuje usługi zarządzania serwerem. Jest to najdroższa alternatywa z bardzo małą możliwością poszerzenia możliwości. Najczęściej

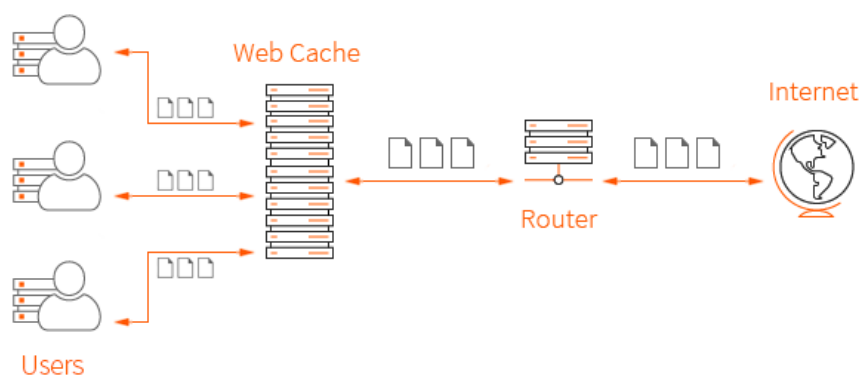
---

<sup>25</sup> Paul Mahony, What Is Cloud Hosting: The Benefits and Drawbacks [online] [<https://hosting.review/web-hosting/what-is-cloud-hosting/>] dostęp: 03.06.2018

wykorzystywana przez ogromne i popularne serwisy internetowe, które są w stanie wykorzystać perspektywę jakiej niesie ze sobą serwer dedykowany.<sup>26</sup>

## 2.2. Web cache

Użytkownicy o podobnych zainteresowaniach bardzo często wczytują i pobierają z sieci te same strony i materiały. Web cache to sprzęt, lub oprogramowanie, które zajmuje się tymczasowym przechowywaniem często wczytywanej zawartości.



Rysunek 7 Web cache

Źródło: maxcdn, [<https://www.maxcdn.com/one/visual-glossary/web-cache/>] dostęp: 03.06.2018

Bez działającego serwera Web Cache, za każdym razem gdy ktoś wyśle żądanie, to musi one zostać przetworzone przez serwer, na którym znajduje się owa witryna (Internet na rysunku nr 6). W przypadku, gdy dużo osób chce wczytać te same pliki może doprowadzić to do wydłużonego czasu oczekiwania, a nawet do przeciążenia serwera. Serwer Cache zajmuje się przejmowaniem tego ruchu i dostarczaniem uprzednio zapisanych plików bezpośrednio do użytkowników bez potrzeby kontaktowania się z głównym serwerem. Istnieje wiele rodzajów Web Cache, które można podzielić ogólnie na prywatny i dzielony. Pierwszy z nich zajmuje się obsługą jednego użytkownika, drugi zbiera dane, które udostępniane są następnie do większej ilości osób. Konkretniej Web Cache rozbić na poniższe kategorie.

- Browser Cache
- Proxy Cache
- Gateway Cache

---

<sup>26</sup> Understanding Web Hosting: How It Works and How to Choose One [online] [<https://satoristudio.net/choosing-hosting/>] dostęp: 02.06.2018

**Browser Cache** - Pamięć podręczna przeglądarki to miejsce na komputerze użytkownika, w którym przetrzymywane są tymczasowo pliki pobrane w celu wyświetlenia strony internetowej, dlatego też pamięć ta zalicza się do grupy Web Cache prywatnych. W chwili ponownego odwiedzenia strony przeglądarka sprawdza czy któryś z plików został zaktualizowany i pobiera jedynie te, które zostały zmienione i te których nie ma jeszcze w pamięci podręcznej. Dodatkowo pliki te wykorzystywane są m.in. podczas nawigacji wstecz/naprzód, zapisywania, lub przeglądania kodu źródłowego witryny. W roku 2018 pamięć podręczna jest już dostępna i szeroko wykorzystywana w przypadku większości przeglądarek internetowych.<sup>27</sup>

**Proxy Cache** – Pamięć podręczna proxy polega na tym, że serwer cache zachowuje się jako pośrednik pomiędzy użytkownikiem, a serwerem źródłowym strony. Serwer proxy przez, który przepływa żądanie osoby odwiedzającej stronę sprawdza, czy posiada najnowszą kopię pliku, o który pyta osoba. W przypadku, gdy plik znajduje się w pamięci podręcznej serwera proxy, użytkownik zostaje obsłużony bez potrzeby przesyłania zapytania do oryginalnego serwera, z którego udostępniana jest strona.



Rysunek 8 Proxy Cache

Źródło: maxcdn, [<https://www.maxcdn.com/one/visual-glossary/proxy-caching/>] dostęp: 03.06.2018

**Gateway Cache** – Dodatkowo serwer proxy wykorzystywany jest często w celu zwiększenia wydajności Web serwera, lub aplikacji. Mowa w tym wypadku o pamięci podręcznej bramy, czasem zwanej również odwrotnym serwerem proxy, który w przypadku stron internetowych znajduje się pomiędzy strefą Internetu, a serwerem Web. Głównym zadaniem tego rozwiązania jest przejmowanie ruchu bezpośrednio przed aplikacją, lub serwerem, na którym znajdują się pliki. Zabieg ten jest to często wykorzystywany w przypadku popularnych stron, które cieszą się ogromnym zainteresowaniem przekładającym się na duży ruch.

Największe zalety wykorzystywania pamięci podręcznej w dziedzinie stron internetowych to:

---

<sup>27</sup> Tina Sieber, Everything You Need to Know About the Browser Cache [online] [<https://www.makeuseof.com/tag/browser-cache-makeuseof-explains/>] dostęp: 03.06.2018

Step	Webpage Response (ms)
1 Get Without Cache	3,013
2 Get With Cache	1,557

Rysunek 9 Web Cache - Webpage response

Źródło: catchpoint, [https://blog.catchpoint.com/2017/04/25/impact-of-web-caching/] dostęp: 03.06.2018

- **Skrócony czas oczekiwania** – czas wczytywania strony jest kluczowym elementem, wykorzystywanie serwerów cache pozwala na jego zredukowanie. Rysunek 8 przedstawia przykładowy wpływ wykorzystania pamięci podręcznej na szybkość wczytywania strony po stronie klienta.
- **Dostępność treści** – kolejnym kluczowym elementem jest dostępność treści witryny, która z kolei może być ograniczona m.in. przez awarie serwera, lub problemy z łączem. W takich sytuacjach wykorzystywanie pamięci podręcznej może uratować sytuację przesyłając użytkownikowi przechowywane w pamięci kopie plików strony.
- **Oszczędzanie transmisji danych** – przez Internet przepływa dziennie ogromna ilość danych, a serwery zmuszane są do obsługi wielkiego ruchu. W efekcie doprowadzić może to do zużycia dużej części ustalonego limitu transmisji. Wykorzystując cache można zauważyć niemały spadek w ilości danych jakie pobierane są z serwera źródłowego. Spora część ruchu obsługiwana jest za pomocą plików przechowywanych w pamięci podręcznej. Rysunek nr 9 przedstawia różnicę w pobranych zasobach bez używania cache, a następnie z jego wykorzystaniem. Dysproporcja jest bardzo duża.<sup>28</sup>

Step	# Wire Requests	Total Downloaded Bytes
1 Get Without Cache	35	382,287
2 Get With Cache	3	30,550

Rysunek 10 Web Cache - bandwidth

Źródło: catchpoint, [https://blog.catchpoint.com/2017/04/25/impact-of-web-caching/] dostęp: 03.06.2018

### 2.2.1. Content Delivery Network

Content Delivery Network to sieć rozproszonych serwerów Web Cache używana do zwiększenia szybkości przesyłania danych, oraz zmniejszania obciążenia serwera

<sup>28</sup> Alva Yashawi, The Impact of Web Caching [online] [http://blog.catchpoint.com/2017/04/25/impact-of-web-caching/] dostęp: 03.06.2018

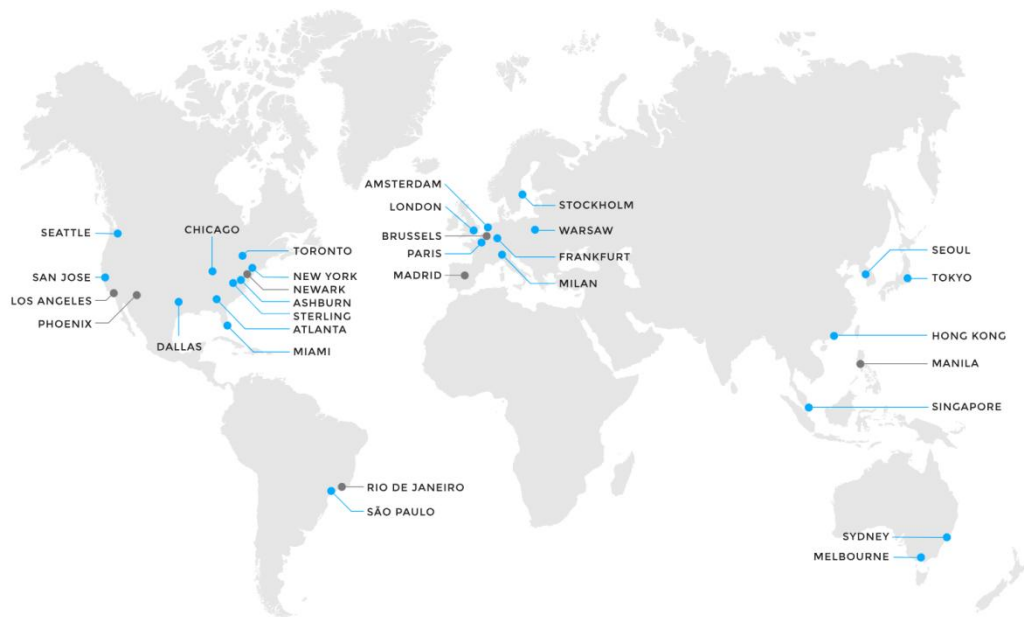
źródłowego. Serwery wymiennie dostarczają pliki strony w zależności od położenia geograficznego osoby wysyłającej zapytanie. Cechą tej sieci jest przede wszystkim ogromna skala, rozproszenie i wysoka wydajność. W roku 2018 korzystanie z możliwości jakie oferuje CDN jest już wśród firm hostingowych powszechne.

Głównymi elementami sieci są serwer źródłowy, na którym znajduje się strona www, rozproszone serwery proxy oraz tzw. system routingu odpowiedzialny za dostarczanie treści do klientów. Content Delivery Network można podzielić na CDN Cache i CDN odpowiedzialny za strumieniowanie danych. Pierwsze z nich zajmuje się pobieraniem m.in. elementów stron internetowych, takich jak pliki HTML, CSS, JavaScript, czy plików do pobrania. To drugie z kolei zoptymalizowane jest pod wszelkiego rodzaju streaming. W obu przypadkach działa to niemalże tak samo, jako że serwery proxy znajdujące się w różnych lokalizacjach pobierają i replikują dane ze serwera źródłowego. Następnie element sieci najbliższy użytkownikowi zajmuje się jego obsługą.<sup>29</sup>

Optymalizacja z użyciem tej technologii pozwala zazwyczaj na obniżenie kosztów usług hostingowych, co przekłada się na niższą cenę dla osoby wykupującej miejsce na serwerze. Szybkość wczytywania stron internetowych jest dzięki temu znacznie przyspieszona, a zwiększone natężenie ruchu jest znacznie lepiej znoszone przez serwer źródłowy. Jednakże oprócz osób odwiedzających strony różnicę odczuwają również roboty wyszukiwarek, które dzięki sprawniejszemu poruszaniu się po serwisie będą w stanie szybciej indeksować odwiedzane witryny.

---

<sup>29</sup> Content Delivery Network w praktyce [online] [<https://www.forbes.pl/technologie/content-delivery-network-w-praktyce/bpblc5e>] dostęp: 04.06.2018



**Rysunek 11 Content Delivery Network**

*Źródło: stackpath, [https://blog.stackpath.com/glossary/cdn/] dostęp: 03.06.2018*

Rysunek 10 przedstawia rozproszenie lokalizacji serwerów proxy związanych z siecią CDN dla jednej z wiodących firm. Niebieskie punkty oznaczają miejsca, w których serwery zostały już zainstalowane i w pełni działają. Z kolei szare punkty oznaczają lokacje, w których prace instalacyjne dopiero trwają. Jak widać sieć jest już szeroko rozproszona i działa w wielu krajach. Content Delivery Network, dzięki temu jak wiele ma do zaoferowania, rozwija się w bardzo szybkim tempie, a co raz więcej firm interesuje się tym tematem.

- CloudFlare
- Amazon
- HP Cloud Services
- Rackspace
- Windows Azure

Korporacje wyżej wymienione to największe firmy w dziedzinie sieci CDN, przy czym z tych wymienionych to CloudFlare w dużym stopniu przewyższa konkurencję. Zaproponowane przez nich rozwiązanie jest bardzo często wybieraną infrastrukturą, z której korzystają firmy hostingowe na całym świecie, w tym w Polsce.

## 2.3. Secure Socket Shell

SSH to protokół komunikacyjny stosowany w sieciach TCP/IP służący do nawiązywania terminalowych połączeń z systemami. Secure Shell wyróżnia się silnym algorytmem uwierzytelniania użytkowników, oraz szyfrowanym połączeniem zapewniającym bezpieczny przesył danych pomiędzy dwoma urządzeniami. Protokół SSH działa na architekturze klient-serwer na domyślnym porcie 22 i jest szeroko wykorzystywany przez administratorów stron internetowych. Używany głównie w celu zarządzania systemami oraz aplikacjami poprzez zdalne połączenie z powłoką systemową. Podczas połączenia z serwerem umożliwia ona m.in. następujące operacje:<sup>30</sup>

- zarządzanie serwerem,
- modyfikacja uprawnień do plików,
- tworzenie i przenoszenie plików,
- tworzenie i przenoszenie katalogów,
- archiwizowanie.

Serwisy hostingowe oferują dostęp do powłoki systemowej głównie w przypadku wykupienia subskrypcji na hosting typu VPS, Cloud, lub serwer dedykowany. Jest to jednak narzędzie wymagające zaawansowanej wiedzy na temat zarządzania serwerami.

Połączenie przy użyciu protokołu SSH chroni użytkowników przed następującymi atakami:<sup>31</sup>

- podsłuch i zafalszowanie sesji,
- podszywanie się pod serwer DNS,
- podszywanie się pod zaufany system,
- podszywanie się pod zdalny serwer docelowy,
- przechwycenie haseł.

## 2.4. Systemy zarządzania treścią

Content Management System to oprogramowanie pozwalające na szybkie i łatwe zarządzanie serwisem internetowym, bez potrzeby posiadania wiedzy programistycznej. Systemy zarządzania treścią pozwalają m.in. na szybkie dodawanie i edytowanie wpisów, zarządzanie wyglądem strony, czy moderację komentarzy. Możliwości różnią się

---

<sup>30</sup> Co to jest SSH i do czego można go wykorzystać? [online] [<https://panel.kylos.pl/knowledgebase/75/Co-to-jest-SSH-i-do-czego-mozna-go-wykorzystac.html>] dostęp: 04.06.2018

<sup>31</sup> Rafał Janus, SSH (Secure Shell) – podstawy bezpiecznej komunikacji [online] [<http://itfocus.pl/dzial-it/sieci/ssh-secure-shell-podstawy-bezpiecznej-komunikacji/>] dostęp: 04.06.2018



w zależności od wybranego systemu i choć oferują one duże ułatwienie w zarządzaniu stronami, to nie zawsze opieranie strony o CMS jest dobrym wyborem. Dotyczy to szczególnie stron statycznych, czyli tzw. wizytówek, które nie wymagają wprowadzania ciągłych zmian. Dzieje się tak ponieważ systemy zarządzania treścią obciążają serwer, dlatego w przypadku braku potrzeby działania na CMS, często lepszym rozwiązaniem jest tworzenie strony bez jego wykorzystania.

Systemy zarządzania treścią można podzielić na trzy ogólne kategorie.

- Licencjonowane CMS
- Dedykowane CMS
- Open Source

Pierwsze z nich to gotowe rozwiązania, za których korzystanie trzeba zapłacić twórcy. Jest to bardzo podobne rozwiązanie do niżej opisanych systemów open source. Różnica polega głównie na tym, że w przypadku tych płatnych można liczyć na korzyści związane z bezpośrednim wsparciem technicznym autora, lub zwiększonym bezpieczeństwem. Są to często rozbudowane i specjalistyczne systemy, które można zoptymalizować pod swoje potrzeby.

Dedykowane CMS to rozwiązania, które oferują największą swobodę personalizacji. Tworzone bezpośrednio pod potrzeby klienta nie mają praktycznie żadnych ograniczeń związanych z funkcjonalnością. Charakteryzują się one wysokim bezpieczeństwem i łatwością rozbudowy. Jest to jednak najdroższa alternatywa, która prawie zawsze w przypadku ewentualnego rozwoju wiązać będzie się z dodatkowymi niemałymi kosztami.

CMS opierane na licencji Open Source są zdecydowanie najbardziej popularną z wymienionych opcji. Ich największym atutem oprócz braku ceny są częste aktualizacje i wiele udoskonaleń proponowanych przez samych użytkowników, którzy często mają możliwość tworzenia swoich wtyczek. Niestety wielka popularność i łatwy dostęp do kodu niesie ze sobą poważne zagrożenia. Systemy takie są szczególnie podatne na ataki hakerskie. Pomimo, że twórcy wprowadzają częste aktualizacje poprawiające bezpieczeństwo, to darmowe systemy zarządzania treścią mimo to pozostają w centrum zainteresowania hakerów.

Istnieje wiele systemów CMS, część z nich dostosowana jest pod różniące się potrzeby klientów. Systemy wyspecjalizowane w ecommerce są naturalnym wyborem podczas tworzenia sklepu internetowego. Duża grupa specjalistycznych CMS z racji na zaawansowane

możliwości pozostaje płatna, można jednak wyróżnić kilka bardzo dobrych i popularnych rozwiązań. Wśród płatnych systemów największą popularnością cieszą się:<sup>32</sup>

- Shopify
- Sky-shop
- Shoper
- IAI-Shop
- Magento Commerce

W przypadku rozwiązań Open Source przodują:<sup>33</sup>

- PrestaShop
- Magento Community Edition
- OpenCart
- WooCommerce

Są to systemy rozwinięte i dobrze zoptymalizowane pod kątem prowadzenia sklepów internetowych. Według badań przeprowadzonych w 2016 roku, aż 84% sklepów internetowych decyduje się na wykorzystanie gotowych już rozwiązań. Tylko 16% stron ecommerce wybrało stworzenie systemu dedykowanego pod konkretne potrzeby ich działalności. Jest to spowodowane wysoką wydajnością wyżej wymienionych skryptów, a z drugiej strony wysokimi kosztami związanymi z tworzeniem rozwiązania od podstaw.

Sklepy to jednak względnie mała część Internetu. Systemy zarządzania treścią są również pomocne w przypadku tworzenia i zarządzania m.in. blogami, stronami z aktualnościami, oraz nawet w niektórych przypadkach stronami informacyjnymi, czyli tzw. wizytówkami. Na przeciw związanym z tym wymaganiom wychodzą systemy CMS, które służą nieco bardziej ogólnym zastosowaniom niż w przypadku tych wyspecjalizowanych. Głównymi graczami tutaj są:

- WordPress
- Joomla!
- Drupal
- Typo3

---

<sup>32</sup> NAJPOPULARNIEJSZE PLATFORMY E-COMMERCE – KTÓRĄ WYBRAĆ? [online] [<https://www.czasnaecommerce.pl/najpopularniejsze-platformy-e-commerce-ktora-wybrac/>] dostęp: 05.06.2018

<sup>33</sup> Paweł Skrzypczak, Jaką platformę ecommerce wybrać? [online] [<https://grafmag.pl/artykuly/jaka-platforme-e-commerce-wybrac>] dostęp: 05.06.2018

Charakteryzują się one różnym poziomem rozbudowania i w zależności od wyboru można spodziewać się innych funkcji. Dodatkowo ich obsługa może wahać się od łatwych w użyciu do tych wymagających wiedzy programistycznej. Oprócz tych najpopularniejszych można wybrać mniej znaną alternatywę. W dziedzinie CMS wybór jest szeroki i każdy jest w stanie znaleźć coś pod swoje potrzeby. W przeciwnym razie są jeszcze rozwiązania dedykowane.

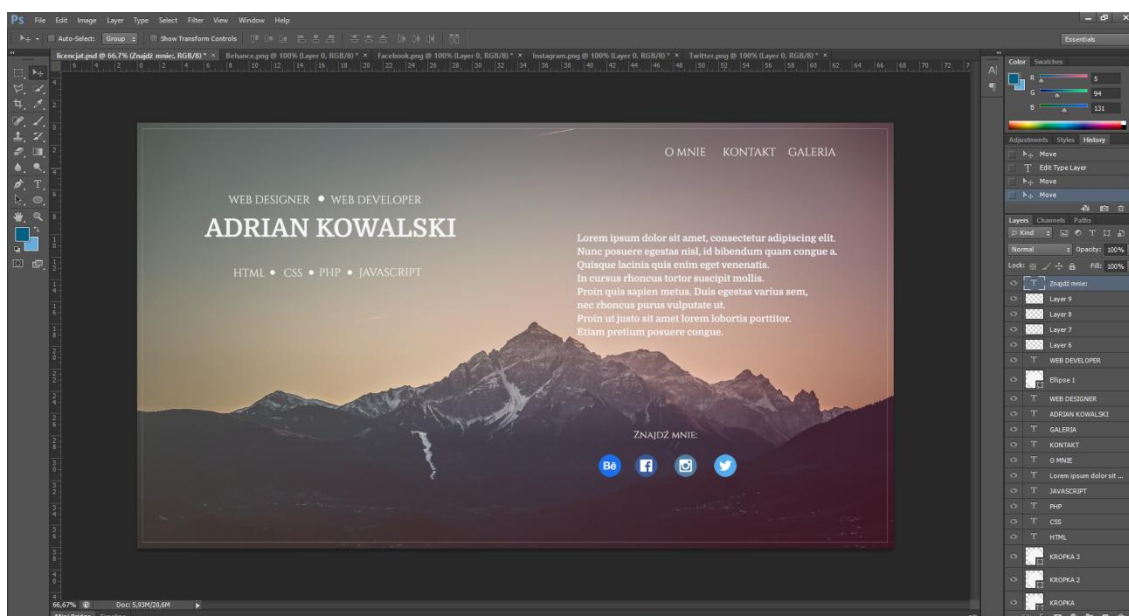
W tym rozdziale autor poruszył tematykę wybranych narzędzi i rozwiązań wspierających proces tworzenia i zarządzania stronami internetowymi. Zaprezentowane i krótko omówione zostały różne alternatywy dla poszczególnych zagadnień. W kolejnym rozdziale zaprezentowany zostanie proces tworzenia strony WWW.

### 3. Tworzenie strony internetowej

W poprzednim rozdziale autor omówił czynniki technologiczne wpływające na działanie strony internetowej. W trzecim rozdziale z kolei przedstawiony zostanie proces tworzenia strony - wizytówki, wraz z opublikowaniem jej w sieci. Wskazane i krótko omówione zostaną użyte do tego celu narzędzia. Dodatkowo zaprezentowany zostanie proces umieszczania owej witryny na serwerze. W celu jak najmniejszego obciążenia serwera – zgodnie z założeniami przedstawionymi w rozdziale drugim – strona nie jest oparta o system zarządzania treścią, przez co wszystkie dane wprowadzane są z poziomu dokumentu HTML.

#### 3.1. Projekt graficzny strony

Tworzenie strony internetowej jest złożonym procesem i zaczyna się najczęściej od ustalenia ogólnego zarysu graficznego. Już pierwszy etap przedsięwzięcia jest kluczowy dla efektu końcowego. Część tą można pominąć jednak prowadzi to najczęściej do znacznie wydłużonego czasu kodowania. Odniesienie do wcześniej zaplanowanej całości nie wymusza na Web Developerze improwizacji. Do tego celu wykorzystywane są różne programy graficzne w zależności od potrzeb i preferencji.



Rysunek 12 Projekt graficzny strony

Źródło. Opracowanie własne

Na potrzeby stworzenia projektu graficznego witryny autor wykorzystał program Adobe Photoshop CS6. Jako, że zamierzeniem było stworzenie witryny One Page zaprojektowana została tylko jedna strona w stanie takim jak trafi na nią użytkownik. W tego typu witrynach

kluczowym elementem jest wybranie odpowiedniego, przejrzystego tła i kolorystyki. W tym celu warto skorzystać z dostępnych w sieci banków zdjęć. Wiele z nich oferuje możliwość darmowego pobrania grafik o wysokiej rozdzielczości. Autor pracy zdecydował się na skorzystanie z grafiki przedstawiającej góry. Ważnym kryterium wyboru była odpowiednia ilość przejrzystej przestrzeni, na której w widoczny sposób umieścić można tekst. Następnie zaprojektowane i rozlokowane zostały menu, oraz bloki z treścią. Z tak przygotowanym zarysem graficznym autor przeszedł do tworzenia szkieletu witryny.

### 3.2. Kod strony – wersja desktop

Brackets to zaawansowany edytor tekstu pozwalający na łatwe zarządzanie kodem. Program jest darmowy, a jego wielkim atutem jest szeroki wybór wtyczek tworzonych przez społeczność. W celu przyspieszenia procesu tworzenia kodu zastosowana została wtyczka Emmet, która oprócz wprowadzania dodatkowych skrótów wspomagających pisanie, pomaga w odnajdywaniu się w strukturze dokumentu HTML. Z tak przystosowanym narzędziem autor przystąpił do tworzenia szkieletu HTML.

```
1 <!DOCTYPE html>
2 <html lang="pl">
3 <head>
4   <meta name="viewport" content="width=device-width, initial-scale=1.0">
5   <meta charset="UTF-8">
6   <title>Adrian Kowalski – Web Developer</title>
7   <meta name="description" content="Adrian Kowalski – Web Designer & Web Developer. Profesjonalne
8     strony internetowe na zlecenie. Strony prywatne i firmowe na terenie całej Polski." />
9   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
10  <link rel="stylesheet" href="css/style.css">
11
12 </head>
```

Rysunek 13 Dokument HTML - nagłówek

*Źródło. Opracowanie własne*

Pierwszym etapem było stworzenie nagłówka witryny. Zawarte zostały tu informacje o obsłudze urządzeń mobilnych, oraz kodowaniu. Dodatkowo ustalone zostały tytuł i krótki opis strony. W nagłówku została również zaimportowana biblioteka jQuery, która wykorzystywana będzie w dalszej części dokumentu i jest kluczowa dla działania niektórych elementów. Na tym etapie stworzony został również plik Kaskadowych Arkuszy Stylów, do którego odwołanie znajduje się również na rysunku nr 12. Następnie zbudowany został szkielet strony uzupełniony o konkretne informacje jakie widnieć będą na gotowej już stronie. Fragment kodu przedstawiający strukturę widoczny jest na następnej stronie.

```

15 <div class="container" id="container">
16   <div class="ramka">
17     </div>
18   <header>
19     <nav>
20       <ul>
21         <li><a href="#" onclick="omnie()">O MNIE</a></li>
22         <li><a href="#" onclick="kontakt()">KONTAKT</a></li>
23         <li><a href="#" onclick="openModal()">GALERIA</a></li>
24       </ul>
25     </nav>
26   </header>
27   <div class="main">
28     <div class="spec">
29       <div class="row1">
30         <p>WEB DESIGNER</p>
31         <p class="ddot"></p>
32         <p>WEB DEVELOPER</p>
33       </div>
34       <div class="row2"><p>Adrian Kowalski</p></div>
35       <div class="row3">
36         <p>HTML</p>
37         <p class="sdot"></p>
38         <p>CSS</p>
39         <p class="sdot"></p>
40         <p>PHP</p>
41         <p class="sdot"></p>
42         <p>JAVASCRIPT</p>
43       </div>
44     </div>
45     <div class="omnie" id="omnie">
46       <p>Nazywam się Adrian Kowalski i od 4 lat działam w branży informatycznej. Głównie
47       zajmuję się tworzeniem stron internetowych opartych o języki HTML, CSS, PHP i
48       JAVASCRIPT. Wykonuję zlecenia z całej Polski, a na co dzień stacjonuję w Katowicach.
49       Jestem osobą młodą i energiczną, dlatego też w moich projektach staram się zawrzeć jak
50       najwięcej świeżości i dążę do tego, by wykonane były one zgodnie z nowoczesnymi
51       trendami dominującymi na rynku.</p>
52     </div>
53     <div class="kontakt" id="kontakt">
54       <p>WEBSERVICES@KOWALSKI.PL</p>
55       <p>TEL. 660 400 400</p>
56       <p>Katowice, 40-871</p>
57     </div>
58     <div id="social">
59       <p>Znajdź mnie:</p>
60       <ul>
61         <li><a href="#"></a></li>
62         <li><a href="#"></a></li>
63         <li><a href="#"></a></li>
64         <li><a href="#"></a></li>
65       </ul>
66     </div>
67   </div>
68 </div>

```

Rysunek 14 Dokument HTML - struktura

Źródło. Opracowanie własne

Jednym z kluczowych elementów stron internetowych jest dobrze zorganizowane menu. W przypadku tej konkretnej strony zbudowane jest ono korzystając ze struktury listy i składa się z 3 elementów. Linki zawarte w menu nie służą jako odnośniki przenoszące na inne podstrony, lecz wywołują działanie skryptu, który szczegółowo przedstawiony zostanie w dalszej części tego rozdziału. Następnie stworzone i uzupełnione zostały bloki treści. W przypadku znacznika `<div>` o identyfikatorze `social`, treść zawiera obrazy, które pobrane zostały z darmowego banku ikon<sup>34</sup>. Są to ikony przedstawiające następujące media społecznościowe:

<sup>34</sup> <https://www.flaticon.com> dostęp: 2018-06-07

- Behance
- Facebook
- Instagram
- Twitter

Służą one jako linki do profili na odpowiadających mediach społecznościowych. Blok z ikonami tak samo jak menu oparty jest o strukturę listy, która ułatwia zarządzanie powiązаныmi ze sobą elementami tworzącymi nierozłącznie jeden blok. Oprócz treści stworzony został również pusty tag <div> o klasie ramka, który skonfigurowany został, by wyświetlać białą ramkę przy krawędzi okna strony, co widoczne jest na rysunku nr 14. Obok przypisania wartości obramowaniu ten fragment CSS odpowiedzialny jest również za ustanowienie tła, wyzerowanie marginesów w głównym „pojemniku” <div>, oraz konfigurację akapitów tekstowych.

```
1 body{
2   background-image: url(../obrazy/background.jpg);
3   background-position: center;
4   background-size: cover;
5   background-repeat: no-repeat;
6   background-attachment: fixed;
7   font-family: Cinzel;
8 }
9 .ramka{
10  border: solid 1px #f6f6f6;
11  position: absolute;
12  top: 10px;
13  left: 10px;
14  bottom: 10px;
15  right: 10px;
16  opacity: 0.3;
17  border-radius: 2px;
18  z-index: -1;
19 }
20 p{
21  color: #f6f6f6;
22  font-size: 14px;
23 }
24 .container{
25  padding: 0;
26  margin: 0;
27 }
```

Rysunek 15 Kaskadowe Arkusze Stylów cz.1

Źródło. Opracowanie Własne

Na rysunku nr 15 przedstawione są wartości przypisane nawigacji, która znajduje się w bloku header. Menu przeniesione jest na prawą stronę witryny i skonfigurowane tak, by linki wyświetlały się jak normalny tekst – bez dekoracji. Dodatkowo najechanie kursorem na link powoduje rozpoczęcie animacji podkreślenia. Zostało to wykonane przy użyciu pseudo klasy **after**, która tworzy element pod danym fragmentem nawigacji i nadaje mu wartości, w tym wypełnienie kolorem. Element stopniowo rozszerza się nadając efekt podkreślenia „od lewej do prawej”.

```
29 ▼ header{
30     height: 100px;
31     padding-right: 10px;
32 }
33 ▼ nav{
34     margin: 10px 15% 15px 10px;
35     float:right;
36     clear: both;
37 }
38 ▼ nav li{
39     display: inline-block;
40     padding: 0 10px;
41 }
42 ▼ nav a{
43     text-decoration: none;
44     font-size: 17px;
45     color:#f6f6f6;
46 }
47 ▼ nav li:after{
48     content: '';
49     display: block;
50     width: 0;
51     height: 1px;
52     background: #f6f6f6;
53     transition: width .4s;
54 }
55 ▼ nav li:hover::after{
56
57     width: 100%;
58 }
```

Rysunek 16 Kaskadowe Arkusze Stylów cz.2

*Źródło. Opracowanie własne*

Po usytuowaniu i nadaniu właściwości nawigacji, następuje proces kodowania graficznej strony bloków treści. Pierwszym z nich jest segment zawierający imię i nazwisko osoby, której witryna dotyczy. Wraz z pozostałymi informacjami został on przeniesiony na ustaloną w projekcie lokację. Komponent został podzielony na 3 rzędy, którym w kolejności przypisane zostały właściwości. Użycie tagu <p>, wraz z odpowiednią konfiguracją w CSS pozwoliło na stworzenie kropek dzielących poszczególne człony.



```

60 ▾ .spec{
61     display: inline-block;
62     width: auto;
63     margin:0;
64     padding: 15px 0 0 6%;
65     float:left;
66 }
67 ▾ .row1{
68     display: flex;
69     justify-content: center;
70     margin-bottom: 0.5em;
71 }
72 }
73 ▾ .spec p{
74     text-align: center;
75     margin: 0 5px;
76     font-size: 22px;
77 }
78 ▾ .ddot{
79     height: 10px;
80     width: 10px;
81     background-color: #f6f6f6;
82     border-radius: 50%;
83     display: inline-block;
84     margin-top: 9px!important;
85 }
86 ▾ .row2 p{
87     font-size: 48px;
88     font-family: Domine;
89     font-weight: bold;
90     text-transform: uppercase;
91 }
92 ▾ .row3{
93     margin-top: 1.75em;
94     display: flex;
95     justify-content: center;
96 }
97 ▾ .sdot{
98     height: 8px;
99     width: 8px;
100    background-color: #f6f6f6;
101    border-radius: 50%;
102    display: inline-block;|
103    margin-top: 10px!important;
104 }

```

Rysunek 17 Kaskadowe Arkusze Stylów cz.3

*Źródło. Opracowanie własne*

Kolejnym etapem było opisanie właściwości bloków zawierających opis „o mnie”, oraz informacje kontaktowe. Oba elementy umiejscowione są w tym samym miejscu na stronie. Kluczową różnicą pomiędzy nimi jest reguła display, która ustawiona jest tak, by fragment zawierający dane kontaktowe nie wyświetlał się po wczytaniu strony. W ten sposób na ekranie użytkownika znajduje się tylko opis. Za pomocą skryptu JavaScript połączonego z nawigacją strony, elementy te wyświetlają się zamiennie w zależności od interakcji użytkownika z witryną. Rozwiązanie to umożliwi umieszczenie większej ilości informacji na stronie internetowej, bez potrzeby rozbijania jej na podstrony. Kod CSS z tym fragmentem jest przedstawiony na następnej stronie.

```

106 ▾ #omnie{
107     text-align: justify;
108     font-family: Domine;
109     float:right;
110     width: 500px;
111     margin: 60px 15% 0 0;
112     display: block;
113 }
114 ▾ #omnie p{
115     margin: 0;
116     padding: 0;
117     font-weight: bold;
118     line-height: 24px;
119 }
120
121 ▾ #kontakt{
122     float:right;
123     width: auto;
124     margin: 50px 22% 40px 0;
125     display: none;
126 }
127 }
128 ▾ #kontakt p{
129
130     margin: 0;
131     padding: 0;
132     line-height: 30px;
133     font-size: 18px;
134 }

```

Rysunek 18 Kaskadowe Arkusze Stylów cz.4

*Źródło. Opracowanie własne*

Ostatnim segmentem widocznym na stronie są odnośniki do mediów społecznościowych. Opisane zostały tu właściwości opisujące położenie elementu, oraz wielkość ikon. Tak samo jak w przypadku nawigacji dodana została tu animacja podkreślenia przy wskazaniu kursorem. W tym wypadku podkreślenie jest krótsze, bo sięga od lewej strony tylko do 1/5 szerokości ikony. Kod przedstawiony jest poniżej.

```

136 ▾ #social{
137     margin: 0;
138     padding: 0;
139     position: absolute;
140     bottom: 240px;
141     right:21%;
142     height: 80px;
143     text-align: center;
144 }
145 ▾ #social p{
146     font-size: 16px;
147 }
148 ▾ li{
149     list-style: none;
150     display: inline-block;
151     padding: 0 10px;
152 }
153 ▾ li img{
154     width: 60px;
155     height: 60px;
156 }
157 ▾ li:after{
158     content: '';
159     display: block;
160     width: 0;
161     height: 1px;
162     background: #f6f6f6;
163     transition: width .4s;
164 }
165 ▾ li:hover::after{
166
167     width: 20%;
168 }

```

Rysunek 19 Kaskadowe arkusze stylów cz.5

*Źródło. Opracowanie własne*

Galeria w stworzonym portfolio jest wyświetlana jako nakładka na stronę główną nie wymagając przy tym wczytywania dodatkowych podstron. Poniżej widać stworzoną do tego celu strukturę, która w połączeniu z JavaScript i CSS pozwala na otwarcie dodatkowego okna w obrębie witryny. Zbudowane zostały cztery slajdy zawierające obrazy, oraz przyciski odpowiedzialne za obsługę galerii, które przy interakcji wywołują powiązany z nimi fragment kodu JavaScript.

```
65 <div id="myModal" class="modal">
66   <span class="close cursor" onclick="closeModal()">&times;</span>
67   <div class="modal-content">
68
69     <div class="mySlides">
70       <div class="numbertext">1 / 4</div>
71       
72     </div>
73
74     <div class="mySlides">
75       <div class="numbertext">2 / 4</div>
76       
77     </div>
78
79     <div class="mySlides">
80       <div class="numbertext">3 / 4</div>
81       
82     </div>
83
84     <div class="mySlides">
85       <div class="numbertext">4 / 4</div>
86       
87     </div>
88
89     <a class="prev" onclick="plusSlides(-1)">&#10094;</a>
90     <a class="next" onclick="plusSlides(1)">&#10095;</a>
91
92     <div class="caption-container">
93       <p id="caption"></p>
94     </div>
95   </div>
96 </div>
```

Rysunek 20 Dokument HTML - Galeria

*Źródło. Opracowanie własne*

Poniższy kod Javascript zawarty jest w dokumencie HTML. Jako, że skrypt jest krótki to nie stanowi to problemu związanego z wydłużonym wczytywaniem strony. Dodatkowo zaoszczędza to przeglądarce potrzeby wczytywania dodatkowych plików. OpenModal() i CloseModal() to kluczowe funkcje odpowiedzialne za pojawianie się i znikanie galerii w zależności od woli użytkownika. Funkcje niżej zajmują się obsługą slajdów – m.in. nadaniu reakcji strzałkom służącym do przewijania obrazów. Oprócz skryptu powiązanego z galerią, na rysunku widoczne są również skrypty, które przypisane są do pozostałych dwóch elementów nawigacji. Zajmują się one wyświetlaniem i ukrywaniem segmentu „o mnie”, oraz „kontakt”, poprzez zmianę wartości reguły display w powiązonym pliku CSS.

```

<script>
function openModal() {
document.getElementById('myModal').style.display = "block";
}

function closeModal() {
document.getElementById('myModal').style.display = "none";
}

var slideIndex = 1;
showSlides(slideIndex);

function plusSlides(n) {
showSlides(slideIndex += n);
}

function currentSlide(n) {
showSlides(slideIndex = n);
}

function showSlides(n) {
var i;
var slides = document.getElementsByClassName("mySlides");
var dots = document.getElementsByClassName("demo");
var captionText = document.getElementById("caption");
if (n > slides.length) {slideIndex = 1}
if (n < 1) {slideIndex = slides.length}
for (i = 0; i < slides.length; i++) {
slides[i].style.display = "none";
}
for (i = 0; i < dots.length; i++) {
dots[i].className = dots[i].className.replace(" active", "");
}
slides[slideIndex-1].style.display = "block";
dots[slideIndex-1].className += " active";
captionText.innerHTML = dots[slideIndex-1].alt;
}
</script>
<script>
function omnie(){
document.getElementById('omnie').style.display = "block";
document.getElementById('kontakt').style.display = "none";
}

function kontakt() {
document.getElementById('omnie').style.display = "none";
document.getElementById('kontakt').style.display = "block";
}
</script>

```

Rysunek 21 Javascript

*Źródło. Opracowanie własne*

Ostatnim elementem potrzebnym do poprawnego działania realizacji jest fragment kodu CSS, który określa właściwości i stan początkowy galerii. Na rysunku nr 21 widać przypisanie maksymalnej wartości pobieranym obrazom. Użycie wartości maksymalnych nie rozciąga obrazów, a pozwala im się przystosować nie przekraczając przy tym ustalonej granicy. W ten sposób zachowana jest ich czytelność. Kluczową wartością w przypadku klasy modal jest z-index, który wypycha galerię na pierwszy plan umożliwiając użytkownikowi swobodną interakcję.

```

170 ▾ .modal img{
171     max-width: 700px;
172     max-height: 700px;
173 }
174 ▾ .modal{
175     display: none;
176     position: fixed;
177     z-index: 1;
178     padding-top: 100px;
179     left: 0;
180     top: 0;
181     background-color: rgba(0, 0, 0, 0.7);
182     width: 100%;
183     height: 100%;
184     overflow: auto;
185 }
186 ▾ .modal-content{
187     width: 700px;
188     height: 700px;
189     margin:auto;
190 }
191 ▾ .close{
192
193     color: white;
194     position: absolute;
195     top: 25px;
196     right: 100px;
197     font-size: 100px;
198 }
199 ▾ .close:hover{
200     cursor: pointer;
201     color: #999;
202     text-decoration: none;
203 }

```

Rysunek 22 Galeria – CSS cz.1

*Źródło. Opracowanie własne*

W drugiej części kodu odnoszącej się do segmentu galerii określone zostały wartości przycisków naprzód i wstecz, wraz z reakcją na najechanie kursorem. Dodatkowo przypisana została wartość dla koloru tekstu wyświetlającego numer slajdu.

```

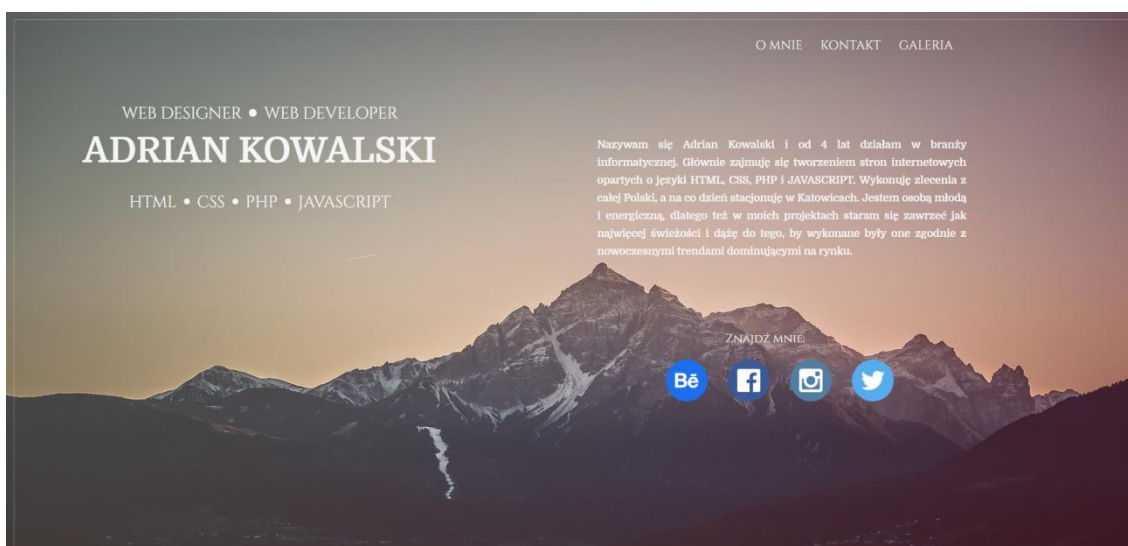
205 .prev
206 ▾ {
207     position: absolute;
208     top: 45%;
209     left: 3%;
210     color: white !important;
211     background-color: rgba(0, 0, 0, 0) !important;
212     box-shadow: none !important;
213     padding: 0.05em 0.35em !important;
214     transition: 0.6s ease;
215     font-size: 36px !important;
216     border-radius: 15px !important;
217 }
218 ▾ .next{
219     position: absolute;
220     top: 45%;
221     right: 3%;
222     color: white !important;
223     background-color: rgba(0, 0, 0, 0) !important;
224     box-shadow: none !important;
225     padding: 0.05em 0.35em !important;
226     transition: 0.6s ease;
227     font-size: 36px !important;
228     border-radius: 15px !important;
229 }
230 .prev:hover,
231 ▾ .next:hover{
232     cursor: pointer;
233     color: black !important;
234 }
235 ▾ .numbertext{
236     color: white;
237 }

```

Rysunek 23 Galeria - CSS cz.2

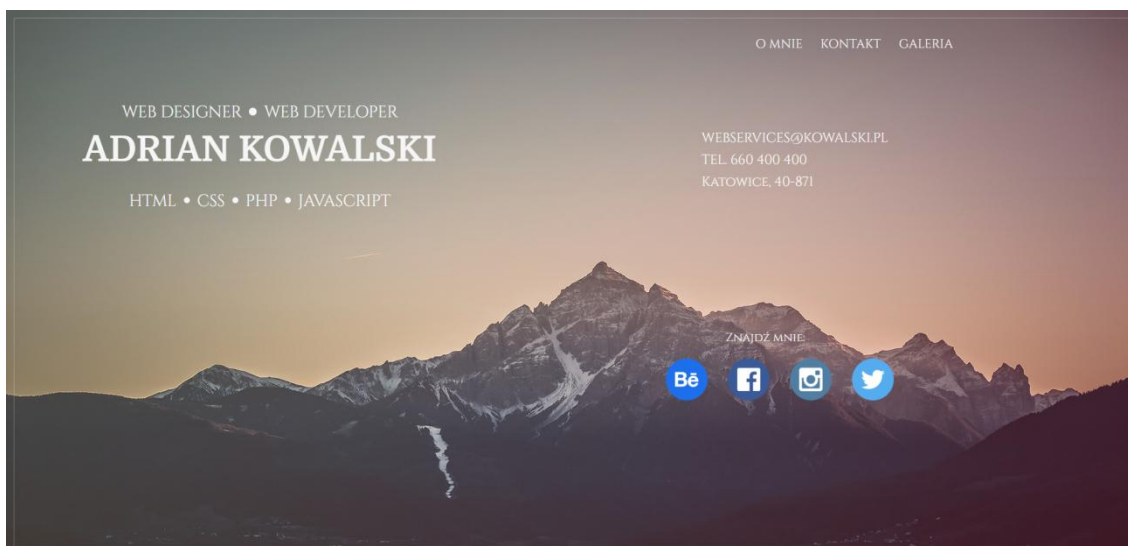
*Źródło. Opracowanie własne*

Gotowa strona zaprezentowana jest na rysunkach 23, 24 i 25. Pomimo nawigacji wykonanej na styl rozbudowanych stron – witryna oparta jest tylko i wyłącznie o stronę indeksową. Podczas przeglądania strony nie występuje wczytywanie, co sprawia, iż witryna działa bardzo szybko. Strona internetowa wykonana jest w stylu minimalistycznym, co w 2018 roku jest jednym z dominujących poglądów na tworzenie serwisów sieciowych. Zamiana tekstu z wykorzystaniem skryptów zobrazowana jest na rysunkach 23 i 24. Pierwszy z nich przedstawia stan początkowy strony, a drugi z kolei efekt przejścia przez menu do zakładki kontakt.



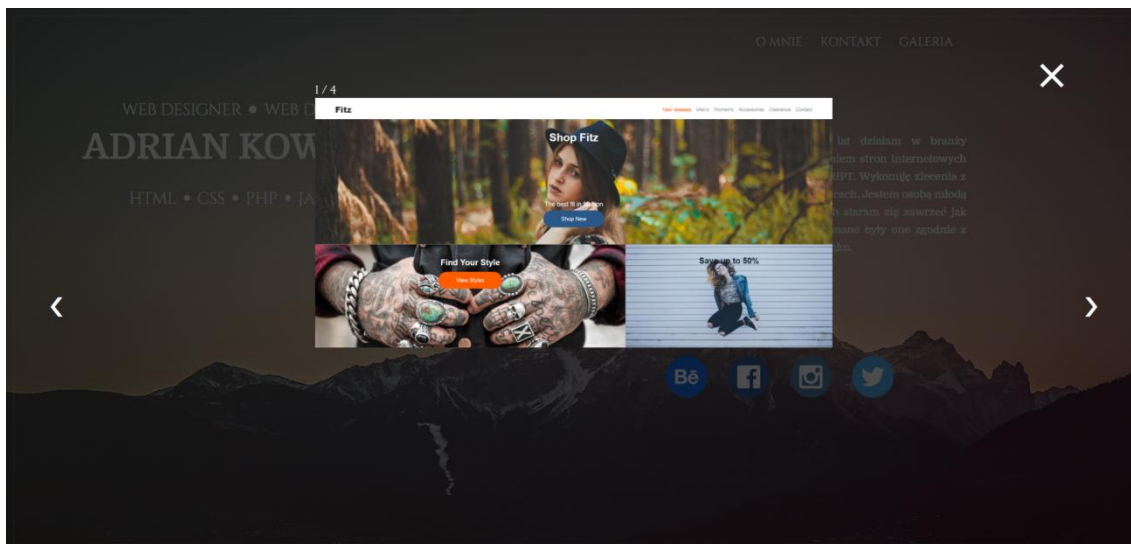
Rysunek 24 Wersja Desktop - O mnie

Źródło. Opracowanie własne



Rysunek 25 Wersja desktop - Kontakt

Źródło. Opracowanie własne



Rysunek 26 Wersja desktop - Galeria

Źródło. Opracowanie własne

### 3.3. Wersja mobilna

W dobie telefonów komórkowych istotne przy tworzeniu strony internetowej jest pamiętanie o wersji mobilnej. Tak też w przypadku tej witryny autor uwzględnił ten nieoficjalny wymóg i przystosował stronę do działania na urządzeniach mobilnych. W kodzie CSS dopisane zostały właściwości jakie przeglądarka ma zastosować w przypadku urządzeń o mniejszym wyświetlaczu.

```
@media screen and (max-width: 500px){
  header{
    height: 100px;
    padding: 0;
  }
  nav{
    margin: 10px 15px 10px 15px;
    float:left;
  }
  nav li{
    padding: 0 7px;
  }
  nav a{
    font-size: 13px;
  }
}
```

Rysunek 27 Wersja mobilna - CSS cz.1

Źródło. Opracowanie własne

Rysunek nr 26 przedstawia fragment kodu, w którym zawarte są instrukcje dla urządzeń mobilnych. W przypadku urządzenia posiadającego rozdzielczość ekranu mniejszą niż 500 pikseli – wszystkie właściwości, które zostały określone w ramach tego zapytania zastąpią te określone dla wersji komputerowej i zostaną użyte przez urządzenie spełniające warunek. W tym przypadku jest to zoptymalizowanie nawigacji i przeniesienie jej na lewą stronę okna.

Poprawione zostało również położenie i wielkość pozostałych elementów strony. W przypadku małych wyświetlaczy wszystkie elementy zostały ułożone tak, by znajdowały się na środku ekranu. Dodatkowo oprócz widocznych na rysunku 27 zmian, poprawione dla urządzeń mobilnych zostało również położenie danych kontaktowych.

```
.spec{
  width: auto;
  margin: 5px 10% 0 10%;
  padding: 5px;
  float: none;
}
.spec p{
  text-align: center;
  margin: 0 5px;
  font-size: 16px;
}
.ddot{
  height: 5px;
  width: 5px;
  margin-top: 8px!important;
}
.row2 p{
  font-size: 36px;
}
.row3{
  margin-top: 1.25em;
}
.sdot{
  height: 5px;
  width: 5px;
  margin-top: 8px!important;
}
#omnie{
  text-align: justify;
  margin-top: 25px;
  padding-top: 5px;
  border-top: solid 1px #f6f6f6
}
#social{
  position: relative;
  bottom: 10px;
  left: 15%;
  right: 15%;
  top: 20px;
}
#social p{
  display: none;
}
li{
  padding: 0 7px;
}
li img{
  width: 40px;
  height: 40px;
}
```

Rysunek 28 Wersja mobilna - CSS cz.2

*Źródło. Opracowanie własne*

Optymalizacji pod kątem niskich rozdzielczości wymagała również galeria, która bazując na właściwościach określonych dla strony w wersji desktopowej wyświetlała zbyt duże zdjęcia. Dodatkowo dla urządzeń mobilnych obramowanie strony jest wyłączone. Przystosowany kod CSS przedstawiony jest na rysunku 28 i 29.



```

@media screen and (max-width: 768px){
  .ramka{
    display: none;
  }
  .modal{
    display: none;
    position: fixed;
    z-index: 1;
    padding-top: 50px;
    left: 0;
    top: 0;
    background-color: rgba(0, 0, 0, 0.7);
    width: 100%;
    height: 100%;
    overflow: auto;
  }
  .modal img{
    max-width: 300px;
    max-height: 300px;
  }
  .modal-content{
    width: 300px;
    height: 300px;
    margin:auto;
  }
  .close{
    color: white;
    position: absolute;
    top: 10px;
    right: 50px;
    font-size: 35px;
  }
  .close:hover{
    cursor: pointer;
    color: #999;
    text-decoration: none;
  }
}

```

Rysunek 29 Wersja mobilna - CSS cz.3

*Źródło. Opracowanie własne*

```

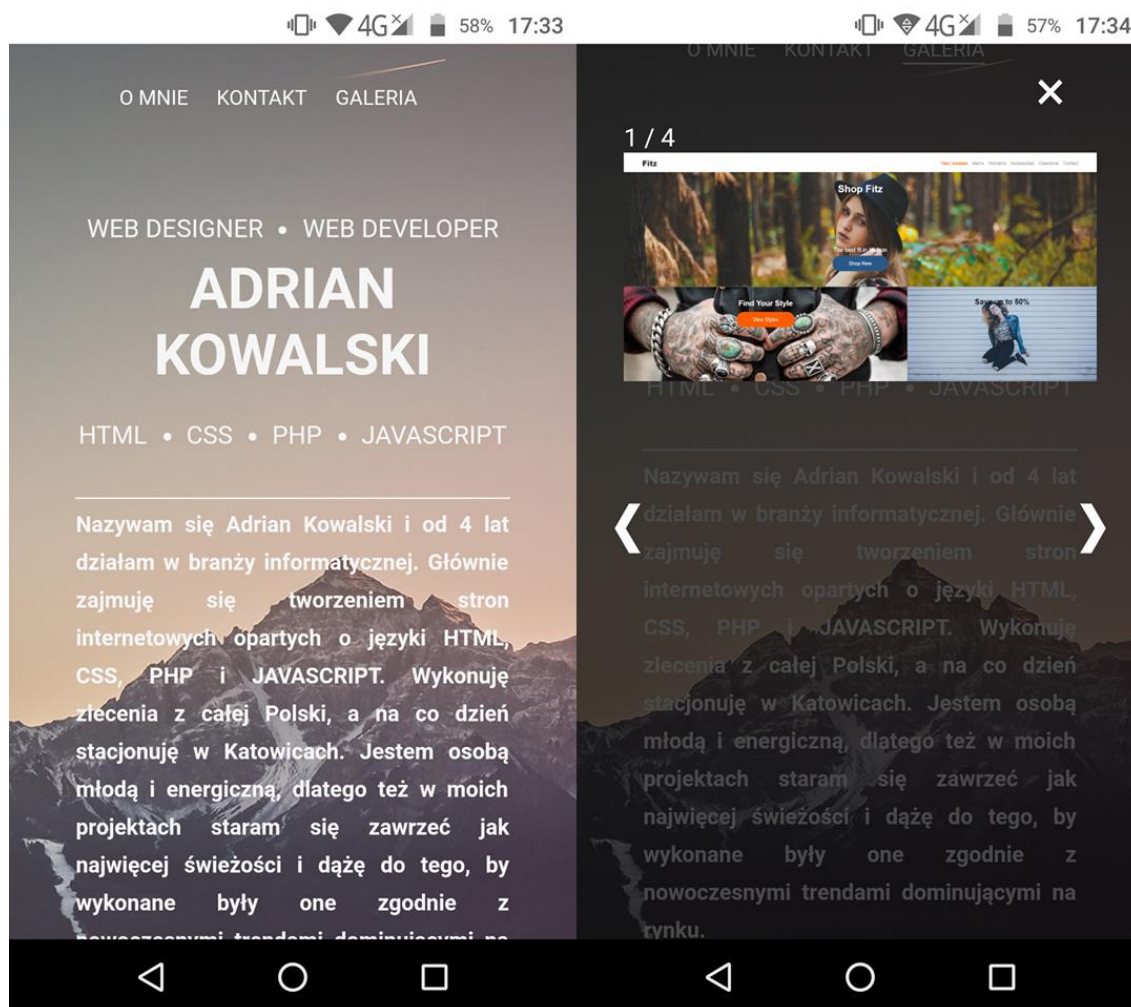
.prev
{
  position: absolute;
  top:45%;
  left: 3%;
  color:white!important;
  background-color: rgba(0, 0, 0, 0)!important;
  box-shadow: none!important;
  padding: 0.05em 0.35em!important;
  transition: 0.6s ease;
  font-size: 36px!important;
  border-radius: 15px!important;
}
.next{
  position: absolute;
  top:45%;
  right: 3%;
  color:white!important;
  background-color: rgba(0, 0, 0, 0)!important;
  box-shadow: none!important;
  padding: 0.05em 0.35em!important;
  transition: 0.6s ease;
  font-size: 36px!important;
  border-radius: 15px!important;
}
.prev:hover,
.next:hover{
  cursor: pointer;
  color:black!important;
}
.numbertext{
  color:white;
}

```

Rysunek 30 Wersja Mobilna - CSS cz.4

*Źródło. Opracowanie własne*

Po zastosowaniu wszystkich potrzebnych do poprawnego działania zmian, strona wyświetla się na urządzeniach mobilnych we właściwy sposób.



Rysunek 31 Wersja mobilna

Źródło. Opracowanie własne

### 3.4. Hosting

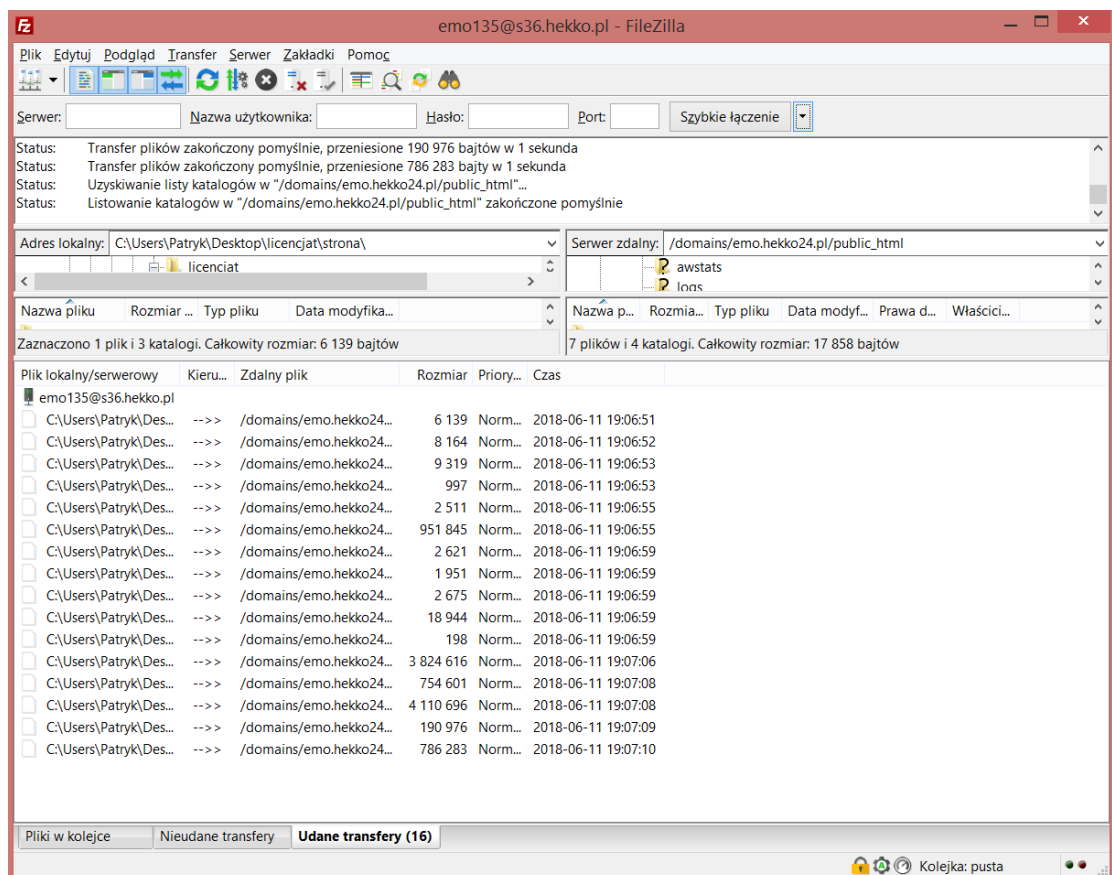
Kluczowym elementem przy publikowaniu serwisu internetowego jest odpowiedni dobór hostingu. Witryny One Page służące jako wizytówka w sieci nie wymagają zaawansowanej konfiguracji i dużych zasobów serwera, dlatego najlepszym wyborem w tym wypadku okazał się hosting współdzielony. Zważając na niskie wymagania transferowe strony, w zupełności wystarczająca okazała się oferta firmy Hekko, której parametry zaprezentowane są na rysunku 31.

	Zaj.	Max
Pojemność (MB)	11.7	15360
Transfer (GB)	0.0893	150.00
E-mail'e [+]	1	nielimitowane
Domeny [+]	1	nielimitowane
Konta FTP [+]	1	nielimitowane
Bazy danych [+]	0	nielimitowane
Ilość plików	169	1000000
E-mail wysłanych dzisiaj	0	5000

Rysunek 32 Domena - parametry

Źródło. Opracowanie własne

Następnie za pomocą otrzymanych danych autor pracy zalogował się na serwerze z wykorzystaniem klienta FTP FileZilla, który umożliwił przesłanie całej struktury strony na serwer. Od momentu, w którym plik index.html znalazł się na maszynie hostującej, strona jest dostępna w sieci i możliwe jest z nią połączenie. Na rysunku nr 32 przedstawiony jest zakończony pomyślnie transfer plików na serwer.



Rysunek 33 FileZilla - przesyłanie plików

Źródło. Opracowanie własne.

## Podsumowanie

Tematyka pracy poświęcona była tworzeniu stron internetowych opartych o cztery najpopularniejsze, przedstawione w pierwszym rozdziale pracy języki. Przedstawione zostało również działanie i alternatywy hostingowe, które są niezbędnym elementem działania strony internetowej w sieci. Dodatkowo omówione zostały narzędzia i rozwiązania wspierające budowę, oraz zarządzanie stronami. W rozdziale drugim zawarty został opis kluczowych technologii wykorzystywanych do zmniejszania obciążenia stron, co przekłada się w znaczący sposób m.in. na ich stabilność, odbiór przez użytkownika, oraz pozycję w wyszukiwarce.

Dziedzina serwisów internetowych jest bardzo rozwiniętym działem informatyki, dlatego też wybór potencjalnych technologii i rozwiązań wspierających dany projekt jest bardzo szeroki. Niemniej jednak ich wykorzystanie w przypadku zaawansowanych serwisów internetowych jest kluczowe zarówno dla sprawnego działania witryny jak i dla lepszego zorganizowania pracy osoby nią zarządzającej.

W rozdziale trzecim przedstawiony został proces tworzenia prostej strony internetowej typu One Page. Omówione zostały etapy przedsięwzięcia, w tym Web Design, budowanie szkieletu strony w oparciu o język HTML, nadawanie właściwości graficznych w plikach CSS, oraz usprawnieniu działania witryny z wykorzystaniem skryptów JavaScript. Oprócz standardowej wersji na urządzenia o wielkiej rozdzielczości, witryna została przystosowana również do działania na urządzeniach mobilnych. Ostatecznie strona przeniesiona została na serwer hostingowy i opublikowana w Internecie.

Proces tworzenia to tylko jeden z wielu etapów prac jakie wykonywane są w związku ze stronami www. Pomimo, że jest to etap relatywnie krótki, to właśnie na nim opierają się w dużym stopniu pozostałe działania. Praca nie wyczerpuje więc w pełni tematyki witryn internetowych. W przyszłych badaniach warto zagłębić się m.in. w zagadnienia szablonów stron internetowych, optymalizacji serwisów pod kątem pozycjonowania w wyszukiwarkach, oraz zarządzaniu stronami w oparciu o poszczególne systemy CMS.

# Bibliografia

## Literatura

1. Clarissa Peterson, Responsywne strony WWW dla każdego, Wydawnictwo Helion, 2015, str. 54, 62, 64-69, 314-316
2. L. Welling, PHP i MySQL. Tworzenie stron WWW. Vademecum profesjonalisty Wydanie III, Wyd. Helion, str. 25
3. Marcin Lis, JavaScript. Ćwiczenia praktyczne. Wydanie III, Wydawnictwo Helion, 2013, str. 7-8, 19

## Źródła internetowe

1. Alva Yashawi, The Impact of Web Caching [online] [https://blog.catchpoint.com/2017/04/25/impact-of-web-caching/] dostęp: 03.06.2018
2. Co to jest SSH i do czego można go wykorzystać? [online] [https://panel.kylos.pl/knowledgebase/75/Co-to-jest-SSH-i-do-czego-mozna-go-wykorzystac.html] dostęp: 04.06.2018
3. Content Delivery Network w praktyce [online] [https://www.forbes.pl/technologie/content-delivery-network-w-praktyce/bpblc5e] dostęp: 04.06.2018
4. Kurs HTML [online] [ http://www.kurshtml.edu.pl] dostęp:15.04.2018r.
5. NAJPOPULARNIEJSZE PLATFORMY E-COMMERCE – KTÓRĄ WYBRAĆ? [online] [https://www.czasnaecommerce.pl/najpopularniejsze-platformy-e-commerce-ktora-wybrac/] dostęp: 05.06.2018
6. Paul Mahony, Managed VPS service: Is it like having your own server administrator? [online] [https://hosting.review/tips-tools/managed-vps-service/] dostęp: 03.06.2018
7. Paul Mahony, What Is Cloud Hosting: The Benefits and Drawbacks [online] [https://hosting.review/web-hosting/what-is-cloud-hosting/] dostęp: 03.06.2018
8. Paweł Skrzypczak, Jaką platformę ecommerce wybrać? [online] [https://grafmag.pl/artykuly/jaka-platforme-ecommerce-wybrac] dostęp: 05.06.2018
9. Rafał Janus, SSH (Secure Shell) – podstawy bezpiecznej komunikacji [online] [http://itfocus.pl/dzial-it/sieci/ssh-secure-shell-podstawy-bezpiecznej-komunikacji/] dostęp: 04.06.2018
10. Tina Sieber, Everything You Need to Know About the Browser Cache [online] [https://www.makeuseof.com/tag/browser-cache-makeuseof-explains/] dostęp: 03.06.2018
11. Understanding Web Hosting: How It Works and How to Choose One [online] [https://satoristudio.net/choosing-hosting/] dostęp: 02.06.2018
12. Uptime Calculator [online] [ https://hosting.review/uptime-calculator/] dostęp: 29.05.2018r.
13. Web Guide [online] [http://www.web-guide.co.il/pages/hosting-parameters.php?lang=en#Databases-MySQL-Access-MsSQL/] dostęp: 02.06.2018r.

## Spis ilustracji i tabel

Rysunek 1 PageSpeed Insights .....	17
Rysunek 2 PageSpeed Insights - szczegóły .....	18
Rysunek 3 Front-End Tooling Trends 2017 - Najczęściej wykorzystywane Frameworki JS .....	20
Rysunek 4 Front-End Tooling Trends 2015 - Najczęściej wykorzystywane Frameworki JS .....	21
Rysunek 5 Nazwa domeny .....	30
Rysunek 6 Tabela: Typy hostingu .....	32
Rysunek 7 Web cache.....	35
Rysunek 8 Proxy Cache.....	36
Rysunek 9 Web Cache - Webpage response .....	37
Rysunek 10 Web Cache - bandwidth.....	37
Rysunek 11 Content Delivery Network.....	39
Rysunek 12 Projekt graficzny strony.....	44
Rysunek 13 Dokument HTML - nagłówek .....	45
Rysunek 14 Dokument HTML - struktura.....	46
Rysunek 15 Kaskadowe Arkusze Stylów cz.1 .....	47
Rysunek 16 Kaskadowe Arkusze Stylów cz.2 .....	48
Rysunek 17 Kaskadowe Arkusze Stylów cz.3 .....	49
Rysunek 18 Kaskadowe Arkusze Stylów cz.4 .....	50
Rysunek 19 Kaskadowe arkusze stylów cz.5 .....	50
Rysunek 20 Dokument HTML - Galeria.....	51
Rysunek 21 Javascript .....	52
Rysunek 22 Galeria – CSS cz.1 .....	53
Rysunek 23 Galeria - CSS cz.2 .....	53
Rysunek 24 Wersja Desktop - O mnie .....	54
Rysunek 25 Wersja desktop - Kontakt .....	54
Rysunek 26 Wersja desktop - Galeria .....	55
Rysunek 27 Wersja mobilna - CSS cz.1 .....	55
Rysunek 28 Wersja mobilna - CSS cz.2.....	56
Rysunek 29 Wersja mobilna - CSS cz.3.....	57
Rysunek 30 Wersja Mobilna - CSS cz.4 .....	57
Rysunek 31 Wersja mobilna.....	58
Rysunek 32 Domena - parametry .....	59
Rysunek 33 FileZilla - przesyłanie plików .....	59