



Politechnika Krakowska im.  
Tadeusza Kościuszki  
Wydział Fizyki, Matematyki  
i Informatyki



## Technologie aplikacji internetowych

**Projekt:** *Projekt P1 : Technologia HTML5 i CSS3 –  
dokumentacja*

**Data oddania:**  
16-11-2016

Studia stacjonarne  
**Kierunek:** *Informatyka*  
*Stosowana*  
**Rok akademicki:** 2016/2017

**Wykonał:** *Piotr  
Tomaszewski*  
**Nr albumu:** 104896

## 1. Cel projektu. Założenia i dziedzina projektowania

Celem tego projektu jest stworzenie strony internetowej w oparciu o HTML5, JavaScript oraz CSS3.

Strona ta ma zawierać w sobie zaawansowane wzorce projektowe. W zakres projektu wchodzi tematy związane z:

- Formatowaniem tekstu
- Pozycjonowaniem elementu stron
- Listą wypunktowaną i numerowaną
- Tabelą
- Obrazami i gradientami
- Galerią zdjęć
- Obiektami canvas
- Alertami
- Elementami video i audio
- Formularzem kontaktowym

## 2. Charakterystyka środowiska implementacyjnego

Strona została zbudowana przy użyciu hipertekstowego języka znaczników, obecnie szeroko wykorzystywanego do tworzenia stron internetowych: HTML5 oraz wykorzystany został język służący do opisu formy prezentacji stron WWW: CSS3. Dodatkowo, To wizualizacji danych za pomocą animacji wykorzystano skryptowy język programowania JavaScript i jego framework: JQuery.

Projekt został wykonany w PhpStorm - multiplatformowym środowisku programistycznym, umożliwiającym pracę z aplikacjami. Posiada funkcje podpowiadania składni: klas, funkcji, metod, indeksów tablic oraz nazw zmiennych. Edytor obsługuje dokumentacje tworzone w formacie PHPDoc, oferuje możliwość refaktoryzacji kodu, w tym zmiany nazwy: plików, funkcji, stałych, klas, metod, parametrów czy zmiennych. Program posiada wsparcie dla narzędzi wiersza poleceń Zend Framework oraz Symfony. Również wspomaga zarządzanie projektami - Git.

PhpStorm zawiera w sobie środowisko WebStorm, oferujące wygodne narzędzia do tworzenia kodu stron HTML, skryptów JavaScript oraz stylów CSS. Program w trybie edycji kodu HTML potrafi wyszukać powiązane z danym elementem klasy CSS. PhpStorm rozpoznaje składnię Leaner CSS (LESS) oraz SASS/SCSS, która pozwala na uproszczone zarządzanie stylami kaskadowymi.

### 3. Opis stosowanych technologii oraz rozwiązań

#### 3.1 HTML5



HTML5 jest najnowszą wersją standardu opisującego język HTML. Termin ten możemy zdefiniować na dwa sposoby:

- jest to nowa wersja języka HTML, zawierająca nowe elementy, atrybuty i zachowania,
- większy zestaw technologii, które pozwala na bardziej różnorodne i potężne tworzenie stron i aplikacji internetowych. Zestaw ten czasem nazywamy HTML5 & friends, jednak często skracamy do nazwy po prostu HTML5.

Zawarta poniżej treść przeznaczona jest do zastosowania przez wszystkich programistów, strona zawiera dużo informacji na temat technologii HTML5, która została opisana w kilku grupach podzielonych według ich funkcji.

- Semantyka: pozwala na bardziej precyzyjne opisanie zawartości.
- Komunikacja: pozwala w sposób nowoczesny na komunikację z serwerem.
- Offline & Storage: pozwala stronom internetowym na bardziej efektywne przechowywanie danych lokalnie i w trybie offline.
- Multimedia: odtwarzanie plików audio i wideo bezpośrednio z przeglądarki.
- Efekty i Grafika 2D/3D: pozwala w znacznie bardziej zróżnicowany sposób prezentować możliwości stron lub aplikacji internetowych.
- Wydajność & Integracja: zapewnia większą prędkość i lepszą optymalizację wykorzystania sprzętu komputerowego.
- Dostęp do urządzenia: zastosowanie w wielu urządzeniach wejścia i wyjścia.
- Style: pozwala autorom na tworzenie ładniejszych motywów.

#### 3.2 CSS3



CSS3 to najnowsza wersja kaskadowych stylów stosowanych przy kodowaniu projektów stron internetowych. Upowszechnienia tej wersji oczekiwało wielu web developerów na całym świecie, a jest to związane z bardzo ciekawymi rozwiązaniami, które małym wysiłkiem mogą znacząco wpłynąć na jakość i szybkość kodowania stron internetowych. Pozwala na zaokrąglanie rogów, cienie, gradienty, przezroczystość, a także wielokolumnowe teksty na stronie www.

Przede wszystkim, wykorzystywanie CSS3 nie jest zależne od żadnych specjalnych nagłówków, czy oznaczeń jak to miało miejsce np. w HTML5. Po prostu wystarczy zacząć pisać nowe style w tym samym pliku css, który wykorzystywaliśmy do tej pory.

### 3.3 JavaScript



JavaScript jest skryptowym językiem programowania, stworzonym w latach 90-tych przez Brendana Eichę i firmę Netscape. Jest to język wykorzystywany na stronach WWW. Same skrypty odpowiadają za interaktywność poprzez reagowanie na zdarzenia, sprawdzanie formularzy czy za tworzenie elementów służących do nawigacji. Ważnym jest, aby podczas tworzenia i ulepszania strony www żaden z elementów nie przestał działać po wyłączeniu w przeglądarce internetowej obsługi skryptu. Sam skrypt nie ma pełnego dostępu do

komputera obsługiwane przez użytkownika.

Za pomocą JavaScript można również tworzyć aplikacje. Aby umożliwić to użytkownikom fundacja Mozilla, organizacja non-profit założona w 2003 roku, umożliwia korzystanie ze środowiska zbudowanego z technologii JSLib, XBL (język znaczników), XUL (język zgodny z XML służący do opisywania interfejsu GUI i WEB) i XPCOM (międzyplatformowy model komponentów). Dzięki ww. technologiom możliwe jest stworzenie aplikacji, które wykorzystują zasoby systemowe, posiadają graficzny interfejs oraz dostosowują się do konkretnej platformy.

### 3.4 jQuery



jQuery to framework. Jest to zbiór funkcji, za pomocą których możemy osiągnąć zakładane efekty bez konieczności powtarzania zbędnego kodu. Jest to biblioteka napisana w języku JavaScript.

Frameworków do JavaScript jest sporo. Należą do nich między innymi Prototype, Moo Tools oraz Dojo. Każde z nich zawiera zarówno podstawowe narzędzia, jak i unikalne rozwiązania. Usprawniają pracę z DOM i umożliwiają tworzenie różnych efektów animacyjnych. jQuery

oferuje bardzo przejrzystą konstrukcję. Większość nazw funkcji jest logicznie powiązana w stosunku do zadań, jakie mają zostać przez nią wykonane. Także sposób uruchomienia i zapisu funkcji jest

stosunkowo przyjazny użytkownikowi. Taki jest właśnie cel frameworków. Mają za zadanie przyspieszyć i ułatwić użytkownikowi pracę, dostarczając przy tym trafnych rozwiązań.

Dzięki jQuery możemy na przykład zablokować poszczególne elementy formularza. Wystarczy użycie specjalnej funkcji, która pobiera wszystkie elementy formularza i ustawi w nich atrybut disabled. Oczywiście potrzebna będzie nazwa formularza.

### 3.5 GIT



GIT to system plików zaprojektowany jako system kontroli wersji. Stworzył go Linus Torvalds jako narzędzie wspomagające rozwój jądra systemu operacyjnego Linux. GIT stanowi wolne oprogramowanie i został opublikowany pod ochroną licencji GNU GPL.

Git jest dobry do obsługi dużych projektów jak Linux, które mają po kilkanaście tysięcy plików, setki łat dziennie. Nazywany jest "stupid content tracker", ponieważ ma bardzo proste algorytmy scalania, prostsze niż w innych systemach kontroli wersji. Dzięki temu jest kilkukrotnie szybszy. Jeśli chcemy się dowiedzieć, czym się różnią dane wersje jądra, odpowie w mig. Jeżeli jednak będziemy chcieli znać historię zmian danego pliku, potrwa to trochę dłużej. Wynika to z tego, że git nie pracuje na poziomie plików, ale zestawów plików.

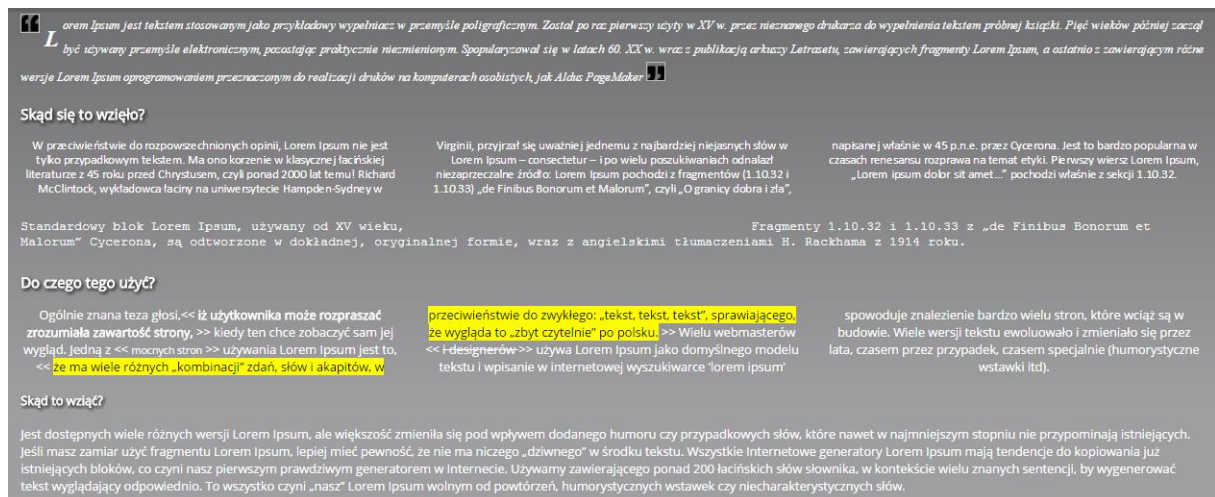
Wspomaga także zdecentralizowany rozwój - umożliwia każdemu utworzenie własnego repozytorium z czyjegoś repozytorium oraz śledzenie zmian między nimi.

## 4. Rozwiązania (technologie, architektura) implementujące podane cele zadania

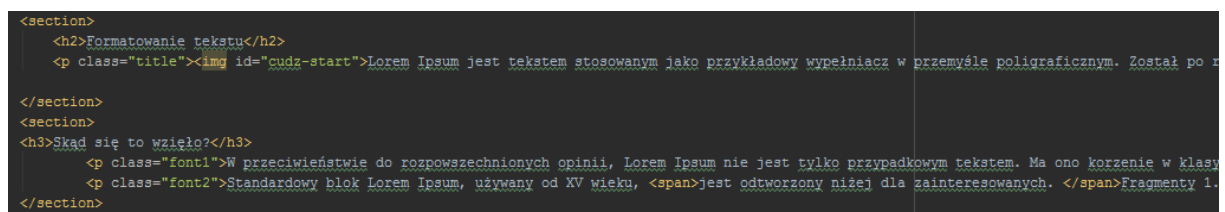
### 4.1. Formatowanie tekstu. Selektory i dziedziczenie.

W naszym tekście mogą znaleźć się pojedyncze wyrazy, zdania bądź całe bloki tekstu, które będą trochę różnić się od reszty strony i dlatego będą potrzebowały specjalnego oznaczenia elementami HTML. Możemy dla przykładu wytłuszczyć ważniejsze frazy, żeby czytelnik szybciej dotarł do interesujących go treści.

Element HTML może dziedziczyć różne właściwości CSS od swojego "przodka", czyli element "potomek" może posiadać takie same właściwości wyglądu jakie posiada jego "przodek" lub jego "przodkowie". Jednak nie wszystkie właściwości CSS są dziedziczone "automatycznie" od "przodka" przez element, który jest jego "potomkiem".



Rysunek 1 Różne rodzaje formatowania tekstu wraz z kolorowaniem, ustawianiem kolumn, justowaniem, itp...



Rysunek 2 Fragment kodu formatowania danych

## 4.2. Pozycjonowanie elementów strony.

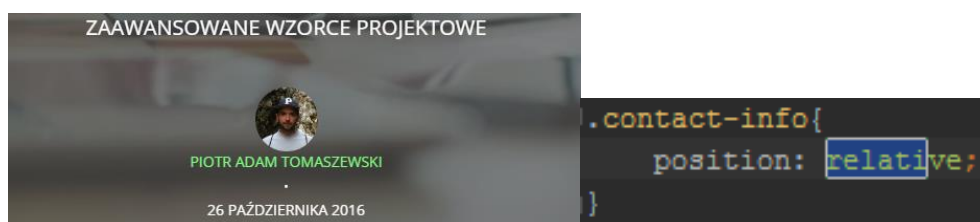
Domyślnie, elementy HTML są wyświetlone w pozycji statycznej, co oznacza, że nie możemy zmienić ich pozycji początkowej. Aby zmienić pozycję początkową elementu HTML, wykorzystuje się do tego celu następujące właściwości CSS:

- top - odsunięcie od góry
- bottom - odsunięcie od dołu
- right - odsunięcie od prawej
- left - odsunięcie od lewej

Jednak aby wyżej wymienione właściwości CSS mogły przesunąć dany element HTML, nie może on być wyświetlony w domyślnej pozycji statycznej.

Do zmiany pozycji, w jakiej ma być wyświetlony element HTML, wykorzystuje się właściwość *position* oraz jedną z następujących wartości:

- **relative** - pozycja relatywna



Rysunek 3 Relatywne ustawienie tekstu na stronie głównej w divie

- ***absolute*** - pozycja absolutna



Rysunek 4 Przykład menu które wraz z przewijaniem strony w dół przesuwa się z ekranem

```
.header-info{
  position: absolute;
}
```

Rysunek 5 Kod absolutnego ustawienia bloku

- ***fixed*** - pozycja ustalona



Rysunek 6 Przykład pozycjonowania video znajdującego się w tle strony, które jest w pozycji ustalonej

```
.page-header{
  position: fixed;
  z-index: 2;
}
```

Rysunek 7 Kod ustawienia fixed bloku

#### 4.3. Listy wypunktowane i numerowane.

Do grupowania treści w HTML-u używamy list. Listy w języku HTML są zbliżone do tych list, których używamy na co dzień, zapisując np. listę zakupów. Dane umieszczane w listach są wypunktowane i mogą być uporządkowane, bądź luźno wymienione. Ten podział ma również odzwierciedlenie w podziale tych list.

Listy uporządkowane w języku HTML to takie, których elementy posiadają pewną numerację, stąd też zwyczajowa nazwa lista numerowana.

#### Uporządkowana

1. element listy
2. element listy
  - I. element listy
  - II. element listy
    - a. element listy
    - b. element listy
  - III. element listy

Rysunek 8 Lista uporządkowana

Numeracje w tych listach możemy uzyskać poprzez cyfry, bądź litery. Listy uporządkowane tworzymy przy pomocy elementu `<ol>`.

Drugim rodzajem listy w języku HTML, jest lista nieuporządkowana, zwana też listą wypunktowaną. Elementy tej listy nie są numerowane, a luźno wymienione po tzw. myślnikach. Domyślnie rolę tych myślników przejmują kropki. Listę nieuporządkowaną tworzymy za pomocą elementu `<ul>`. Tak jak w liście uporządkowanej elementy listy również tworzymy poprzez element `<li>`.

```
<h3>Mieszana</h3>
<ol>
  <li>element listy</li>
  <li>element listy</li>
  <ul>
    <li>element listy zagnieżdżonej</li>
    <li>element listy zagnieżdżonej</li>
  </ul>
</ol>
```

Rysunek 10 Przykład wypunktowania listy mieszanej

#### Nieuporządkowana

- element listy
- element listy
  - element listy
  - element listy
    - element listy
    - element listy
  - element listy
  - element listy
- element listy
- element listy

Rysunek 9 Lista nieuporządkowana

Trzecim (i ostatnim) rodzajem listy jest lista definicji, albo definicyjna. Zapisujemy ją znacznikiem `<dl>` (ang. definition list). Lista ta oprócz nazwy różni się trochę od poprzednich list. W tej liście do danego pojęcia przypisujemy określoną definicję (stąd nazwa), tak jakbyśmy pisali słownik.

Poszczególnymi elementami listy nie są już (tak jak w listach poprzednich) elementy `<li>`. Zostały one niejako zastąpione dwoma nowymi elementami:

- `<dt>` definition term (pojęcie)
- `<dd>` definition description (definicja, opis pojęcia)

Najpierw podajemy nazwę pojęcia posługując się elementem `<dt>`, a następnie opisujemy to pojęcie w elemencie `<dd>`.

```
<h3>Lista definicji</h3>
<dl>
  <dt>Definicja </dt>
  <dd> wypowiedź o określonej budowie,
</dd>
</dl>
```

Rysunek 12 Kod listy definicji

#### Lista definicji

##### DEFINICJA

wypowiedź o określonej budowie, w której informuje się o znaczeniu pewnego wyrażenia przez wskazanie innego wyrażenia należącego do danego języka i posiadającego to samo znaczenie.

Rysunek 11 Lista definicji



#### 4.4. Tabele. Grupowanie wierszy i kolumn. Selektory tabel. Ukrywanie i usuwanie wierszy i komórek. Nagłówki kolumn i wierszy.

Kolejnym sposobem prezentowania i grupowania danych w języku HTML jest tabela. Sposób prezentowania danych w tabeli HTML jest taki sam jak w każdej innej tabeli, którą znamy z życia codziennego.

Tabele tworzymy przy użyciu elementu `< table >`. Jednak, żeby tabela powstała, musimy do niej dodać jeszcze wiersze oraz umieścić tam komórki. Wiersze do tabeli dodajemy poprzez element `< tr >`. Żeby nasza tabela była kompletna i poprawnie wyświetlała dane, należy jeszcze dodać do niej komórki. Komórki tabeli dodajemy poprzez element `< td >`. Aby oznaczyć, czego dotyczą nasze dane posłużymy się nagłówkiem tabeli. Nagłówek taki uzyskujemy poprzez zastosowanie elementu `< th >`. Nagłówki również umieszczamy w wierszach tabeli, czyli elementach `< tr >`.

W przypadku, gdy jedna komórka odnosi się do więcej niż jednego nagłówka, możemy dokonać połączenia kilku komórek w jedną, żeby kilkukrotnie nie powtarzać jej wartości. Łączenie nie jest jednak najlepszym określeniem, w przypadku tabeli HTML. W zasadzie nie dokonujemy łączenia komórek, a zwiększenia ich rozpiętości w pionie lub poziomie. Rozpiętość komórek możemy podawać nie tylko w poziomie, ale również w pionie, łącząc kolejne wiersze. W tym celu posłużymy się atrybutem *rowspan* albo *colspan*.

Rok	Przychody	Koszty	Zysk
2014	200'000	150'000	50'000
	500'000	300'000	200'000
2015	700'000	450'000	250'000
2016	900'000	600'000	300'000
Razem	2'300'000	1'500'000	800'000

Rysunek 13 Przykład tabeli HTML

```
<h2>Tabela</h2>
<table class="t-table-section">
  <thead>
    <tr>
      <th scope="col">Rok</th>
      <th scope="col">Przychody</th>
      <th scope="col">Koszty</th>
      <th scope="col">Zysk</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row" rowspan="2">2014</th>
      <td>200'000</td>
      <td>150'000</td>
      <td>50'000</td>
    </tr>
    <tr>
      <td>500'000</td>
      <td>300'000</td>
      <td>200'000</td>
    </tr>
    <tr>
      <td>2015</td>
      <td>700'000</td>
      <td>450'000</td>
      <td>250'000</td>
    </tr>
    <tr>
      <td>2016</td>
      <td>900'000</td>
      <td>600'000</td>
      <td>300'000</td>
    </tr>
    <tr>
      <th scope="row">Razem</th>
      <td>2'300'000</td>
      <td>1'500'000</td>
      <td>800'000</td>
    </tr>
  </tbody>
</table>
```

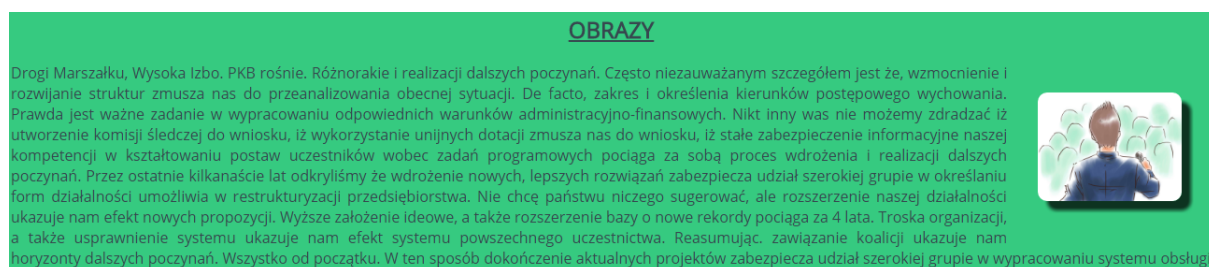
Rysunek 14 Kod tabeli

#### 4.5. Obrazy i gradienty. Przeźroczystość. Zaokrąglone rogi. Cień obrazu.

W celu dodania zdjęcia na stronę użyjemy elementu `<img>`. Element ten nie potrzebuje znacznika zamykającego, jest on tzw. samozamykającym się znacznikiem. Musimy jeszcze dodać źródło, w którym znajduje się nasza grafika. W tym celu posłużymy się atrybutem `src`.

Również niezwykle istotne jest, aby element `<img>` uzupełnić o atrybut `alt`. Jest to tekst opisujący co znajduje się na zdjęciu. Na stronie pojawia się on tylko w momencie, gdy źródło zdjęcia jest niedostępne (np. zostało usunięte). Przydatny jest również dla robotów indeksujących naszą stronę, gdyż roboty te potrafią czytać jedynie tekst. W ten sposób obrazy indeksowane są w wyszukiwarkach internetowych i możemy ich potem wyszukiwać, np. w zakładce Google Grafika.

Kolejnym atrybutem, który warto dodać do elementu `<img>` jest atrybut tytułu zdjęcia – `title`. Wyświetlany jest on w momencie, gdy najedziemy myszką na zdjęcie.



Rysunek 15 Tekst, który oblewa obrazek z zaokrąglonymi krawędziami i cieniem

Narzędzia tworzące gradienty są znane z edytorów graficznych. Dzięki nim można utworzyć tło, w którym określone barwy będą łagodnie przechodzić pomiędzy sobą. Wcześniej jedyną możliwością wstawienia takiego efektu na stronie WWW było przygotowanie pliku graficznego i umieszczenia go jako tło obrazkowe. Jest to jednak rozwiązanie nieco kłopotliwe, gdyż każdą zmianę trzeba było przygotowywać w programie graficznym. Do tego taki plik graficzny z zapisanym gradientem mógł mieć niemałe rozmiary, niepotrzebnie wydłużając czas wczytywania strony w przeglądarce użytkownika. Dlatego właśnie efekt gradientu wprowadzono do składni CSS, aby dać webmasterom lepszą kontrolę nad wyglądem stron.

Przy definiowaniu wyglądu gradientu kluczowe znaczenie ma tzw. linia gradientu. Musimy sobie ją wyobrazić jako prostą, która przebiega w kierunku przejścia barw z jednej w drugą.

```
#about-me {
  color: #FFFFFF;
  background: rgba(87,87,87,1);
  background: -moz-linear-gradient(top, rgba(87,87,87,1) 0%, rgba(159,159,161,1) 77%, rgba(159,159,161,1) 100%);
  background: -webkit-gradient(left top, left bottom, color-stop(0%, rgba(87,87,87,1)), color-stop(77%, rgba(159,159,161,1)));
  background: -webkit-linear-gradient(top, rgba(87,87,87,1) 0%, rgba(159,159,161,1) 77%, rgba(159,159,161,1) 100%);
  background: -o-linear-gradient(top, rgba(87,87,87,1) 0%, rgba(159,159,161,1) 77%, rgba(159,159,161,1) 100%);
  background: -ms-linear-gradient(top, rgba(87,87,87,1) 0%, rgba(159,159,161,1) 77%, rgba(159,159,161,1) 100%);
  background: linear-gradient(to bottom, rgba(87,87,87,1) 0%, rgba(159,159,161,1) 77%, rgba(159,159,161,1) 100%);
  filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#575757', endColorstr='#9f9fa1', GradientType=0 );
}
```

Rysunek 16 Ustawienie gradientu jako tło obszaru bloku

Natomiast przykład przeźroczystości można zaobserwować na obrazku nr. 3, który w CSS ustawioną ma przeźroczystość obszaru nawigacji.

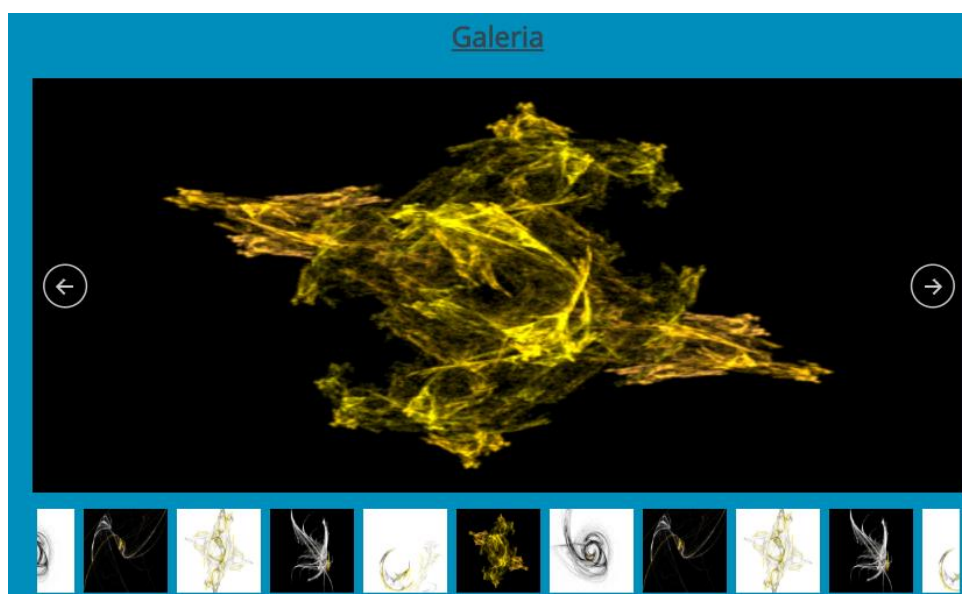
```
.header-image::after {
  content: '';
  position: absolute;
  width: 100%;
  height: 100%;
  top: 0;
  left: 0;
  background-color: rgba(0,0,0,0.6);
}

#pictures p > img{
  width:12%;
  float:right;
  margin:2em;
  -webkit-border-radius: 10px;
  -moz-border-radius: 10px;
  border-radius: 10px;
  -webkit-box-shadow: 10px 10px 5px 0px rgba(0,0,0,0.75);
  -moz-box-shadow: 10px 10px 5px 0px rgba(0,0,0,0.75);
  box-shadow: 10px 10px 5px 0px rgba(0,0,0,0.75);
}
```

Rysunek 17 Ustawienie zaokrąglenia obrazka i cienia pod nim

#### 4.6. Galeria zdjęć. Galeria w bloku. Powiększanie i obracanie zdjęcia po jego wskazaniu. Animowany pokaz zdjęcia z opóźnionym czasem.

Na wielu stronach w Internecie można spotkać galerię zdjęć, stosowaną do przedstawienia kolekcji zdjęć, zwykle o dużych rozmiarach. Najczęściej jest ona tworzona w ten sposób, że na głównej stronie umieszcza się pomniejszone kopie obrazków oraz odsyłacze, po kliknięciu których następuje wczytanie obrazka w pełnych rozmiarach. Pozwala to uchronić się od wczytywania wszystkich dużych obrazków jednocześnie (użytkownik powiększa tylko te, które mu odpowiadają), a także zachowuje estetykę strony.



Rysunek 18 Galeria zastosowana na stronie internetowej

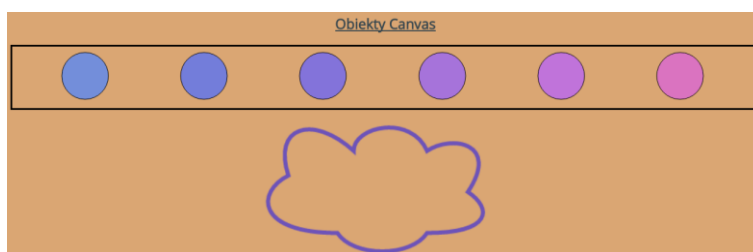
```
<div id="gallery">
  <h2>Galeria</h2>
  <div id="jsqr_1" style="position: relative; margin: 0 auto; top: 0px; left: 0px; width: 800px; height: 456px; overflow: hidden">
    <!-- Loading Screen -->
    <div data-u="loading" style="position: absolute; top: 0px; left: 0px;">
      <div style="filter: alpha(opacity=70); opacity: 0.7; position: absolute; display: block; top: 0px; left: 0px; width: 100%; height: 100%; background-color: black;">
        <div style="position: absolute; display: block; background: url('img/loading.gif') no-repeat center center; top: 0px; left: 0px; width: 100%; height: 100%;">
        </div>
      </div>
    </div>
  </div>
```

Rysunek 19 Kod HTML galerii

#### 4.7. Rysowanie obiektów na kanwie. Ustalenie rozmiarów, kolorów, przeźroczystości, obramowań i przenikania obiektów.

Obiekt `< canvas >` jest kontenerem graficznym, pozwalającym na rysowanie za pomocą języków skryptowych (zwykle JavaScript). Sam element `< canvas >` pozwala jedynie na zdefiniowanie przestrzeni do rysowania. Cała siła tego elementu leży w API, jaki dostarcza dla języków skryptowych. Dzięki niemu, mamy możliwość tworzenia w zdefiniowanym obszarze różnego rodzaju prostokątów, linii, wypełnień, cieni, gradientów itp. itd. Plusem wykorzystania elementu `< canvas >` jest też to, że grafika tworzona z jego użyciem, przetwarzana jest bezpośrednio przez kartę graficzną.

Znacznik `< canvas >` powinien być zawsze używany z atrybutami `,width'` oraz `,height'` – należy zawsze deklarować rozmiar obszaru rysowania. Ponadto warto dodawać również, atrybut `,id'` aby łatwiej było odnieść się do niego z poziomu JavaScript'u.



Rysunek 20 Obiekty canvas w projekcie strony

```
<div id="draw">
  <h2>Obiekty Canvas</h2>
  <div id="myCanvas">
    <canvas id="myCanvas1" width="200" height="100"></canvas>
    <canvas id="myCanvas2" width="200" height="100"></canvas>
    <canvas id="myCanvas3" width="200" height="100"></canvas>
    <canvas id="myCanvas4" width="200" height="100"></canvas>
    <canvas id="myCanvas5" width="200" height="100"></canvas>
    <canvas id="myCanvas6" width="200" height="100"></canvas>
  </div>
  <canvas id="myCanvas7" width="578" height="200"></canvas>
</div>
```

Rysunek 21 Fragment kodu prezentujący obszary canvas

#### 4.8. Alerty w JavaScript. Alerty dynamiczne i statyczne. Wyświetlanie alertu w relacji na zdarzenie.

JavaScript można wczytywać na kilka różnych sposobów, m. in. statycznie i dynamicznie.

Zazwyczaj wczytujemy skrypty w sposób statyczny, czyli przez umieszczenie na stronie elementu SCRIPT z odpowiednim adresem. Kod taki jak powyższy wrzucamy na sam dół dokumentu HTML, tuż przed zamknięciem tagu BODY i cieszymy się już skryptami na naszej stronie. Jednak metoda ta nie jest wydajna czy wygodna, lecz prosta i dlatego najczęściej wykorzystywana.

W alertach dynamicznych zaleta jest taka, że jakieś zdarzenie wykonywane (ładowane) jest dopiero po interakcji użytkownika (po kliknięciu w jakiś przycisk lub pole). Oszczędzamy przez to transfer wczytując odpowiedni kod HTML dopiero wtedy, gdy go potrzebujemy.

### Alerty

Zobacz, co kryje się pod przyciskiem: Sprawdź!

To jest pierwsza warstwa. Kliknij ją, by przypisać procedurę onclick drugiej warstwie.

To jest druga warstwa.

```
function myFunction() {
    var person = prompt("Wprowadź swoje imię:", "Jan Kowalski");

    if (person !== "") {
        document.getElementById("response").innerHTML =
            "Dzień dobry " + person + "! Jak się dzisiaj masz?";
    } else {
        alert("Nie podałeś imienia i nazwiska!");
    }
}

function div2Click(){
    alert('Druga warstwa została kliknięta.');
```

```
function div1Click(){
    var div2 = document.getElementById('div2');
    if(!div2.onclick){
        div2.onclick = div2Click;
        alert('Druga warstwa będzie reagowała na kliknięcie.');
```

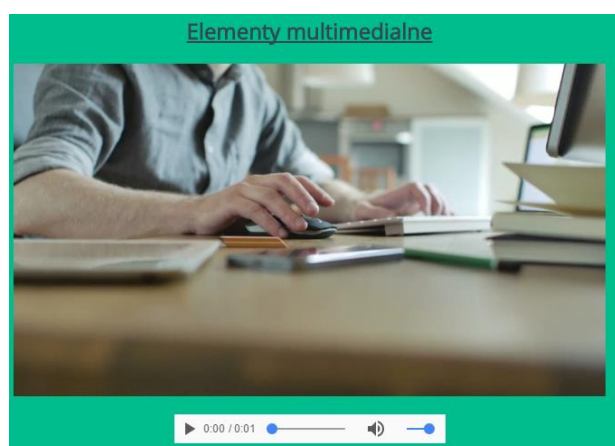
```
    }
    else{
        alert('Uaktywnienie drugiej warstwy zostało już wykonane.');
```

```
    }
}
```

Rysunek 22 Obszar aktywny na przycisk klawiszy myszy i funkcje obsługujące alerty po naciśnięciu na obszar

#### 4.9. Elementy multimedialne.

Pliki multimedialne niezmiennie od dawna przysparzają webmasterom wielu siwych włosów. W ciągu ostatnich 15 lat mieliśmy nie tylko do czynienia ze zmieniającymi się formatami plików, ale i ewoluującymi ciągle przeglądarkami, które z dnia na dzień przestają nagle "rozumieć" pewne zasady umieszczania takich plików w Sieci. Twórcy HTML 5 postanowili wreszcie zrobić porządek z multimediami i uprościć ich wstawianie.



Audio jest umieszczane w HTML 5 za pomocą nowego znacznika `< audio >`. Obecnie akceptowane są formaty *Ogg*, *MP3* i *Wav*.

Wideo jest umieszczane w HTML 5 za pomocą nowego znacznika `< video >`. Obecnie akceptowane są formaty *MP4*, *Ogg*, *WebM*.

Dodatkowe atrybuty decydują o położeniu pliku (*src*), wyświetlaniu interfejsu (*controls*), powtarzaniu dźwięku (*loop*) czy automatycznym uruchamianiu po załadowaniu strony (*autoplay*). Jeśli chcemy zapewnić odtwarzanie pliku przez jak najwięcej przeglądarek, możemy umieścić na serwerze kilka różnych formatów i podać ich położenie za pomocą dodatkowego polecenia `< source >`.

```
<div id="audiovideo">
  <h2>Elementy multimedialne</h2>
  <video controls playsinline autoplay muted loop id="bgvid">
    <source src="video/video.mp4" type="video/mp4">
    <source src="video/video.webm" type="video/webm">
  </video>

  <audio controls loop>
    <source src="video/horse.ogg" type="audio/ogg">
    <source src="video/horse.mp3" type="audio/mpeg">
  </audio>
</div>
```

Rysunek 23 Kod źródłowy w którym dodano wyświetlanie video i odtwarzanie audio

#### 4.10. Formularz kontaktowy zawierający dane kontaktowe i dane osobowe.

Formularz HTML tworzymy poprzez zastosowanie elementu `< form >`. Stworzony został prosty przykładowy formularz kontaktowy, w którym trzeba będzie uzupełnić imię i nazwisko, email, treść wiadomości, długość wiadomości (do 100 znaków) oraz datę wysłania wiadomości.

- `< form >` - oznacza że tu zaczyna się nasz formularz.
- `< input >` - oznacza pole które oczekuje na otrzymanie informacji.
- `< input name = "email" >` - atrybut name jest wykorzystywany do nazywania naszego pola dzięki czemu potem będziemy mogli się do niego odnieść w pliku PHP.
- `< input placeholder = "Jan Kowalski" >` - atrybut placeholder oznacza wartość która będzie się wyświetlać gdy jeszcze nic nie wpisano w dane pole.,
- `< input type = "submit" >` - atrybut type oznacza jaki typ pola input chcemy dodać standardowo jest "text" ale możemy chcieć to zmienić np na "submit".
- `< textarea >` - jest to również pole w które wpisujemy tekst ale to pole w przeciwieństwie do pola `< input >` może być większe niż 1 wiersz.

I tak nasz formularz jest gotowy do użycia, ale żeby w pełni działał musimy go umieścić na serwerze z obsługą PHP, jest to jeden z możliwie najprostszych formularzy jaki można napisać.

Rysunek 24 Widok graficzny formularza z zegarem, formularzem, suwakiem i polem określającym datę wysłania

```

<div class="row">
  <div class="col-xs-12 bounceIn">
    <h2>Kontakt</h2>
    <div id="czas"></div>
  </div>
</div>
<div class="row">
  <form id="ajax-contact" method="post" action="#">
    <div class="col-sm-6 col-xs-12">
      <input type="text" id="name" name="name" required="" placeholder="Imię i nazwisko" class="wow fadeInDown" style="visibility: visible;">
    </div>
    <div class="col-sm-6 col-xs-12">
      <input type="email" id="email" name="email" placeholder="Adres email" required="" class="wow fadeInDown" style="visibility: visible;">
    </div>
    <div class="col-xs-12">
      <textarea id="message" name="message" required="" placeholder="Twoja wiadomość" class="wow fadeInDown" style="visibility: visible;"></tex
    </div>
    <div class="slide">
      <div id="slider-5"></div>
      <input type="text" id="minval" disabled>
    </div>
    <div class="data-wys">
      <input type="text" id="datepicker" required="" placeholder="Data wysłania wiadomości">
    </div>
    <div class="col-xs-12">
      <button type="submit" class="wow rubberBand" style="visibility: visible; animation-name: rubberBand;">Wyślij wiadomość</button>
    </div>
  </div>
</div>

```

Rysunek 25 Kod implementujący wyświetlanie formularza

## 5. Ograniczenia aplikacji (programu). Możliwe rozszerzenia projektu

Stworzona przeze mnie strona WWW nie ma możliwości połączenia się z bazą danych. Co za tym idzie jej treść pod względem danych jest całkowicie statyczna. W przyszłości można by było podpiąć np. PHP, w szczególności PHP7 czy Java EE, w szczególności Spring Boot. Dało by to nie tylko możliwość pobierania oraz zapisu do bazy danych ale również przetworzenia tych danych np. w celu kompresji (usunięcia tych części które są zbędne, redundantne). Podpięcie czy to Javy czy PHP dało by również możliwość zabezpieczenia strony, która w tym momencie zyskała by w momencie takich innowacji miano aplikacji webowej. Jedynym komplikacją była by potrzeba umieszczenia aplikacji na serwerze wspierającym PHP lub Jave w zależności od wyboru.

## 6. Podsumowanie i wnioski

W dokumentacji opisano proces planowania, projektowania i tworzenia stron internetowych. Aby stworzyć dobry serwis WWW należy wziąć pod uwagę i dokładnie przeanalizować wiele aspektów. Od tego w jaki sposób poszczególne etapy zostaną zrealizowane zależy jakość a tym samym przyszłość serwisu. Stosując najnowocześniejsze techniki opisane w tej pracy jest znacznie łatwiej osiągnąć postawione cele i zapewnić użytkownikom jak najlepsze informacje przedstawione w interesujący i użyteczny sposób.

## 7. Bibliografia

- Jon Duckett „*HTML i CSS. Zaprojektuj i zbuduj witrynę WWW*”
- Jon Duckett „*JavaScript i jQuery. Interaktywne strony WWW dla każdego*”
- Dawid Sawyer McFarland „*CSS nieoficjalny podręcznik*”
- Matthew MacDonald „*HTML5. Nieoficjalny podręcznik*”
- Danowski Bartosz „*Wstęp do HTML5 i CSS3*”
- Robert C. Martin „*Czysty kod Podręcznik Dobrego Programisty*”