

Akademia Górniczo-Hutnicza im. St. Staszica

CSS

**Serwis Font Awesome, rozwijane
menu**

Tomasz Bartuś

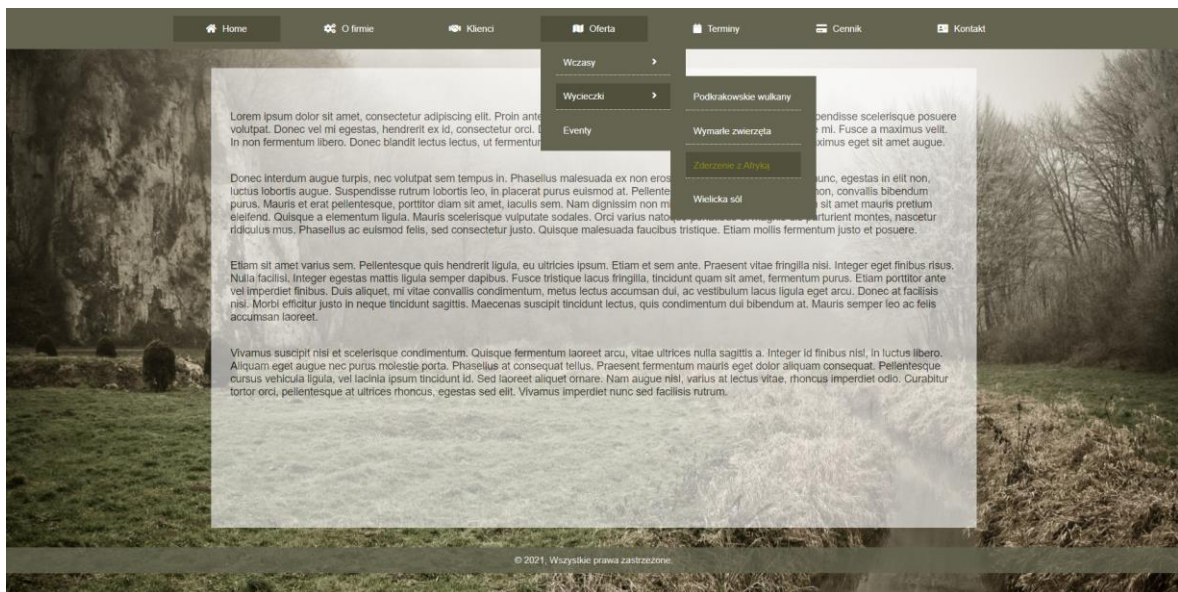
Wersja: 19.04.2023



2023

Wprowadzenie

W ramach kolejnego ćwiczenia utworzymy szablon prostej strony internetowej opierającą się o kaskadowe arkusze styli (css) i model blokowy (pudełkowy). Tym razem poza przypomnieniem sobie sposobu tworzenia struktury modelu blokowego, zajmiemy się także utworzeniem 3-poziomowego rozwijanego menu oraz zapoznamy się z możliwością skorzystania z serwisu internetowego udostępniającego darmowe ikony. Strona www będzie prezentowała ofertę małej firmy turystycznej funkcjonującej na rynku wczasów zagranicznych, eventów i wycieczek geoturystycznych. Strona będzie miała wygląd przedstawiony na [Ryc. 1](#).



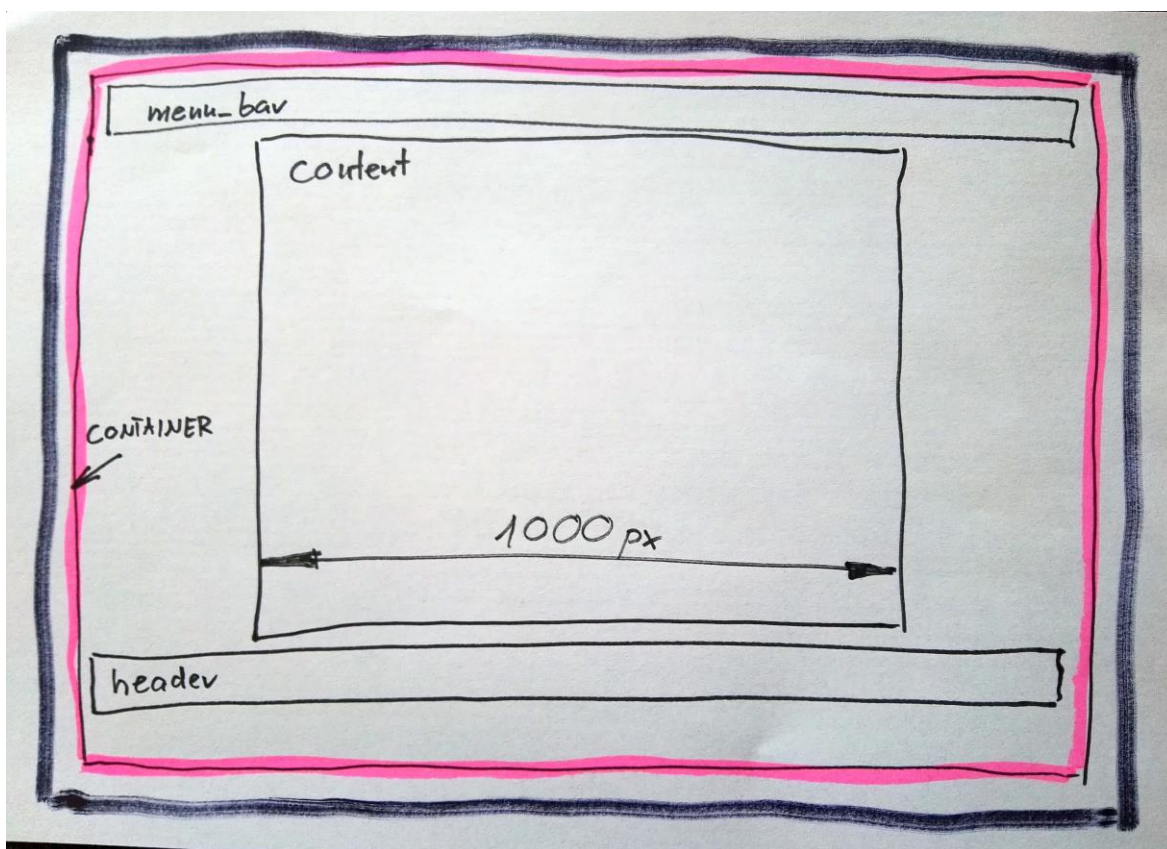
Ryc. 1. Wygląd projektowanej strony www

Założenia projektu strony www

Jak widać na [Ryc. 1](#) tym razem strona składa się z czterech bloków w kształtach prostokątów. W tle całości strony umieścimy klimatyczną

fotografię z Ojcowca. Strona jest wyśrodkowana, a po jej lewej i prawej stronie mamy marginesy dające wrażenie przestrzeni.

Schematycznie rozrysujmy sobie układ projektu strony www (Ryc. 2). Na schemacie zaznaczono proponowane nazwy identyfikatorów elementów blokowych `<div></div>` oraz szerokość bloku `content`.











Ryc. 2. Schemat układu bloków projektu strony www wraz z wymiarowaniem



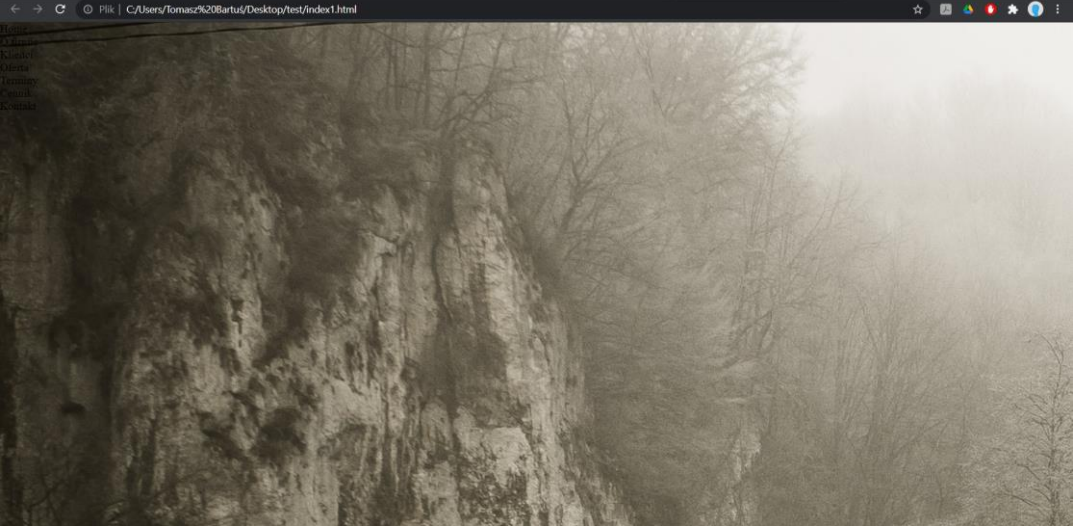

Ćwiczenie

Przejdźmy do wykonania zaprojektowanej strony.

1. W wybranej lokalizacji utwórz folder ćwiczeniowy `\css_04`.

	<p>2. Jeśli go nie posiadasz utwórz w programie Notepad++ szablon dokumentów html.</p> <pre data-bbox="209 432 1410 925"><!DOCTYPE HTML> <html lang="pl"> <head> <meta charset="utf-8"> <title></title> <link rel="stylesheet" type="text/css" href="style.css"> </head> <body> </body> </html></pre>
	<p>3. Zachowaj plik pod nazwą <code>index.html</code> w folderze ćwiczeniowym <code>\css_04</code>.</p>
	<p>4. W folderze ćwiczeniowym <code>\css_04</code> utwórz w programie Notepad++ pusty plik o nazwie <code>style.css</code>.</p>
	<p>5. Do sekcji <code><head></head></code> dodaj tytuł strony oraz metainformacje słów kluczowych oraz opisu strony.</p> <pre data-bbox="209 1391 1410 1653"><title>Firma turystyczna</title> <meta name="description" content="Strona firmy turystycznej. Oferta wyjazdów zagranicznych, eventów, wycieczek"> <meta name="keywords" content="Kraków, turystyka, geoturystyka, wczasy, wycieczki, eventy"></pre>
	<p>6. Wewnątrz tagu <code><body></body></code> utwórz element blokowy <code><div></code> o identyfikatorze <code>id="container"</code>. Tradycyjnie będzie on zawierał wszystkie bloki niższego rzędu.</p> <pre data-bbox="209 1883 1410 2018"><body> <div id="container"></pre>

	<pre></div> </body></pre>
	<p>7. Wewnątrz utworzonego bloku kontenera utwórz blok o identyfikatorze <code>id="menu_bar"</code>. Umieścimy w nim całe menu rozwijane.</p> <pre><div id="container"> <div id="menu_bar"> </div> </div></pre>
	<p>8. Wewnątrz bloku o identyfikatorze <code>menu_bar</code> tworzymy listę nieuporządkowaną. Utworzy ona najwyższy poziom rozwijanego menu.</p> <pre><div id="container"> <div id="menu_bar"> Home O firmie Klienci Oferta Terminy Cennik Kontakt </div> </div></pre>
	<p>9. Do utworzonego wcześniej pustego pliku <code>style.css</code> dodajemy pierwszą deklarację CSS dotyczącą wszystkich elementów HTML. To tzw. selektor uniwersalny. Tworzymy go za pomocą znaku gwiazdki (*).</p> <pre>*{ padding: 0; margin: 0;</pre>

	}
	<p>10. W tle strony internetowej pojawi się klimatyczne zdjęcie. Dodajmy je do projektu w pliku <code>style.css</code>. Zastosowany plik <code>DSC_0075.jpg</code> znajdziecie Państwo w załączniku do ćwiczenia.</p> <pre>body{ background-image: url(./images/DSC_0075.jpg); }</pre>
	<p>11. Wyświetl stronę w przeglądarce www (Ryc. 3).</p>  <p>Ryc. 3. Wygląd strony www po dodaniu elementów menu najwyższego poziomu oraz tła strony</p>
	<p>12. Elementy menu, które mają być docelowo poziome ułożyły się pionowo w prawym narożu strony. Nie dziwi nas to ponieważ póki co jest to zwykła lista nieuporządkowana. Przez ciemne tło obrazka napisy przyszłego menu są na razie słabo widoczne. Zdjęcie jest duże i nie zostało dostosowane do wielkości okna przeglądarki. Widzimy tylko jego niewielki fragment (prawe górne naroże). Dodajemy do deklaracji CSS <code><body></body></code> właściwość, która dostosuje rozmiar obrazu tła, aby przykrył on całe okno przeglądarki.</p> <p>Deklaracja <code>background-size: cover;</code> spowoduje dynamiczne</p>

dostosowywanie wielkości obrazka do wielkości bloku (w tym przypadku okna przeglądarki).

```
body{
  background-image: url(./images/DSC_0075.jpg);
  background-size: cover;
}
```



13. Do deklaracji `<body></body>` dodajemy definicję typowego kroju czcionki. Zastosujemy ogólną deklarację nie precyzującą konkretnej nazwy czcionki. Strona ma zostać wyświetlona za pomocą domyślnej czcionki bezszeryfowej: `font-family: sans-serif;`.

```
body{
  background-image: url(./images/DSC_0075.jpg);
  background-size: cover;
  font-family: sans-serif;
}
```

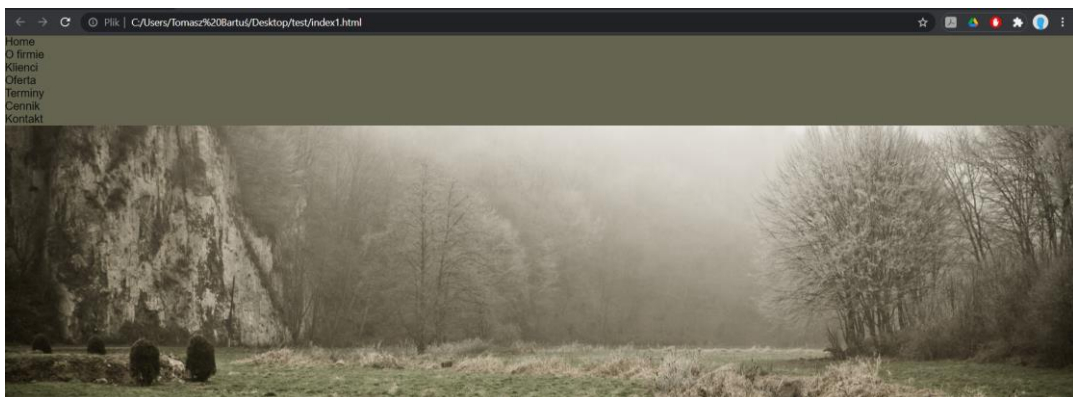


14. Do pliku `style.css` dodajemy deklarację identyfikatora bloku o identyfikatorze `id="menu_bar"`, a do niego wpisujemy kolor tła bloku w formacie `rgb` (r-red, g-green, b-blue).






```
#menu_bar{
  background: rgb(100,100,80);
}
```








15. Odśwież stronę w przeglądarce `www` ([Ryc. 4](#)).



Ryc. 4. Wygląd strony `www` po modyfikacji koloru tła bloku `menu_bar`

	<p>16. Wycentrujemy napisy menu względem bloku <code>menu_bar</code>.</p> <pre>#menu_bar{ background: rgb(100,100,80); text-align: center; }</pre>
	<p>17. Odśwież stronę w przeglądarce www (Ryc. 5).</p>  <p>Ryc. 5. Wygląd strony www po wycentrowaniu treści bloku <code>menu_bar</code></p>
	<p>18. Dla wszystkich elementów <code></code> występujących wewnątrz bloku o identyfikatorze <code>menu_bar</code> utworzymy właściwość <code>display: inline-flex;</code>. Za pomocą właściwości <code>display</code> dla każdego typu elementu HTML możemy określić sposób jego wyświetlania. Ma to wpływ na sposób układania się elementu HTML względem sąsiadujących elementów HTML lub tekstu. Typami elementu HTML, na które najczęściej natrafiamy są: <code>block</code>, <code>inline</code> oraz <code>inline-block</code>. Elementy HTML typu <code>inline-flex</code> po wyświetleniu w oknie przeglądarki internetowej układają się w poziomie jeden obok drugiego w jednym wierszu.</p> <pre>#menu_bar ul{ display: inline-flex; }</pre>
	<p>19. Odśwież stronę w przeglądarce www (Ryc. 6).</p>

	 <p>Ryc. 6. Wygląd strony www po ułożeniu elementów <code></code> bloku <code>menu_bar</code> jeden obok drugiego</p>
	<p>20. Elementy listy nieuporządkowanej przeszkadzają w czytelności tekstu. Pozbądźmy się kropek obok elementów listy <code></code>.</p> <pre data-bbox="209 779 1407 952">#menu_bar ul{ display: inline-flex; list-style: none; }</pre>
	<p>21. Odśwież stronę w przeglądarce www (Ryc. 7).</p>  <p>Ryc. 7. Wygląd strony www po usunięciu znaczników listy nieuporządkowanej</p>
	<p>22. Wszystkie teksty elementów listy nieuporządkowanej są ze sobą „sklejone”. Rozdzielmy je przez zdefiniowanie szerokości elementów listy. Utwórzmy definicję szerokości elementów listy <code></code> wewnątrz <code></code> wewnątrz bloku o identyfikatorze <code>menu_bar</code>.</p> <pre data-bbox="209 1592 1407 1713">#menu_bar ul li{ width: 120px; }</pre>
	<p>23. Odśwież stronę w przeglądarce www (Ryc. 8).</p>  <p>Ryc. 8. Wygląd strony www po zdefiniowaniu szerokości elementów listy</p>

nieuporządkowanej

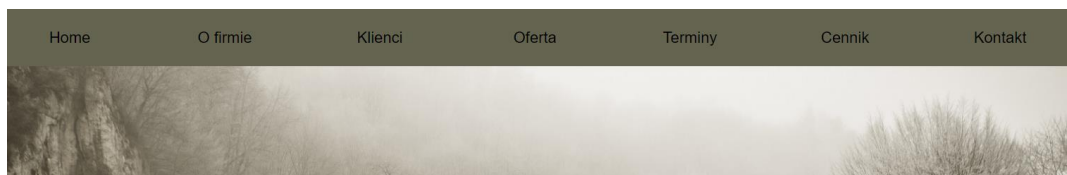


24. Elementy listy nieuporządkowanej przylegają do górnej krawędzi przeglądarki www (*de facto* do bloku `container`) ale także do dolnej krawędzi bloku `menu_bar`. Odsuńmy je i dajmy im pewną przestrzeń. Do utworzonej deklaracji elementów listy wewnątrz wewnątrz bloku o identyfikatorze `menu_bar` dodajmy marginesy wewnętrzny (`padding`) i zewnętrzny (`margin`).

```
#menu_bar ul li{
  width: 120px;
  margin: 10px;
  padding: 10px;
}
```



25. Odśwież stronę w przeglądarce www ([Ryc. 9](#)).



**Ryc. 9. Wygląd strony www po zdefiniowaniu marginesów wewnętrznych i zewnętrznych elementów **



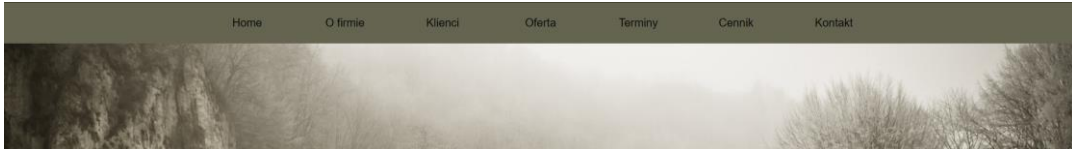
26. Napisy przyszłych linków menu pierwszego poziomu są bardzo rozstrzelone względem siebie. Powodem jest dodawanie do każdego elementu i paddingów i marginów, które powiększają odstępy. Aby elementy nieco zbliżyć do siebie zmienimy domyślne wymiarowanie z typowego dla przeglądarek typu Mozilla (obliczającego parametry `width` i `height` bloków bez uwzględnienia borderów, paddingów i marginów) na typowego dla Internet Explorera (obliczającego parametry `width` i `height` bloków z uwzględnieniem borderów i paddingów ale nie marginów). Dodamy deklarację: `box-sizing: border-box;`.

```
#menu_bar ul li{
```

```
width: 120px;
margin: 10px;
padding: 10px;
box-sizing: border-box;
}
```



27. Odśwież stronę w przeglądarce www (Ryc. 10).



Ryc. 10. Wygląd strony www modyfikacji odstępów pomiędzy napisami bloku menu_bar

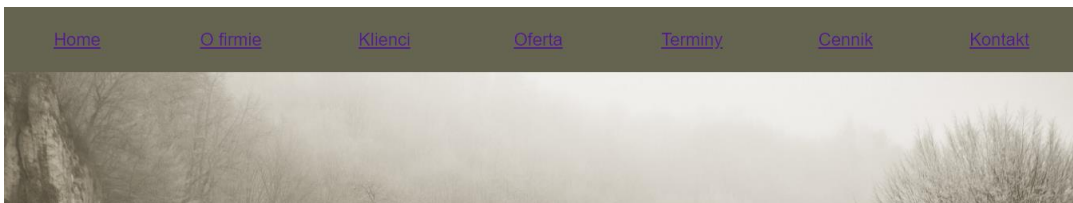


28. Przejdźmy do pliku `index.html` i dodajmy do każdego napisu menu tagi linków `Home`.

```
<div id="menu_bar">
  <ul>
    <li><a href="" title="Strona główna">Home</a></li>
    <li><a href="" title="O firmie">O firmie</a></li>
    <li><a href="" title="Klienci">Klienci</a></li>
    <li><a href="" title="Oferta">Oferta</a></li>
    <li><a href="" title="Terminy">Terminy</a></li>
    <li><a href="" title="Cennik">Cennik</a></li>
    <li><a href="" title="Kontakt">Kontakt</a></li>
  </ul>
</div>
```



29. Odśwież stronę w przeglądarce www (Ryc. 11).



Ryc. 11. Wygląd strony www po dodaniu do bloku menu_bar linków

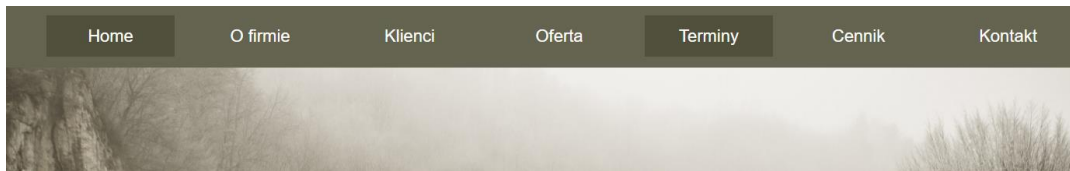
	<p>30. W pliku <code>style.css</code> utwórzmy deklarację dla wszystkich elementów <code><a></code> pojawiających się wewnątrz elementów <code></code>, wewnątrz elementów <code></code> i wewnątrz bloku o identyfikatorze <code>menu_bar</code>:</p> <pre>#menu_bar ul li a{ text-decoration: none; color: #fff; }</pre>
	<p>31. Odśwież stronę w przeglądarce www (Ryc. 12).</p>  <p>Ryc. 12. Wygląd strony www po zdefiniowaniu wyglądu linków menu</p>
	<p>32. Chcielibyśmy wizualnie wyróżnić element menu <code>HOME</code> od innych elementów menu. W tym celu dla tego jednego elementu zdefiniujemy specjalną klasę <code><li class="active">HOME</code>.</p> <pre><div id="menu_bar"> <li class="active">Home O firmie Klienci Oferta Terminy Cennik Kontakt </div></pre>
	<p>33. W pliku <code>style.css</code> zdefiniujemy wygląd przycisku HOME. Nadamy mu styl taki jaki będą miały wszystkie przyciski rozwijanego menu w momencie kiedy najedziemy na nie kursorem. Za jednym razem zdefiniujemy więc reguły dla:</p>

- przycisku HOME w stanie bez najechania na niego kursorem (klasa `.active`),
- Pozostałych przycisków po najechaniu na nie kursorem myszy (pseudoklasa `#menu_bar ul li:hover`).

```
.active, #menu_bar ul li:hover{  
    background: rgb(80,80,60);  
}
```



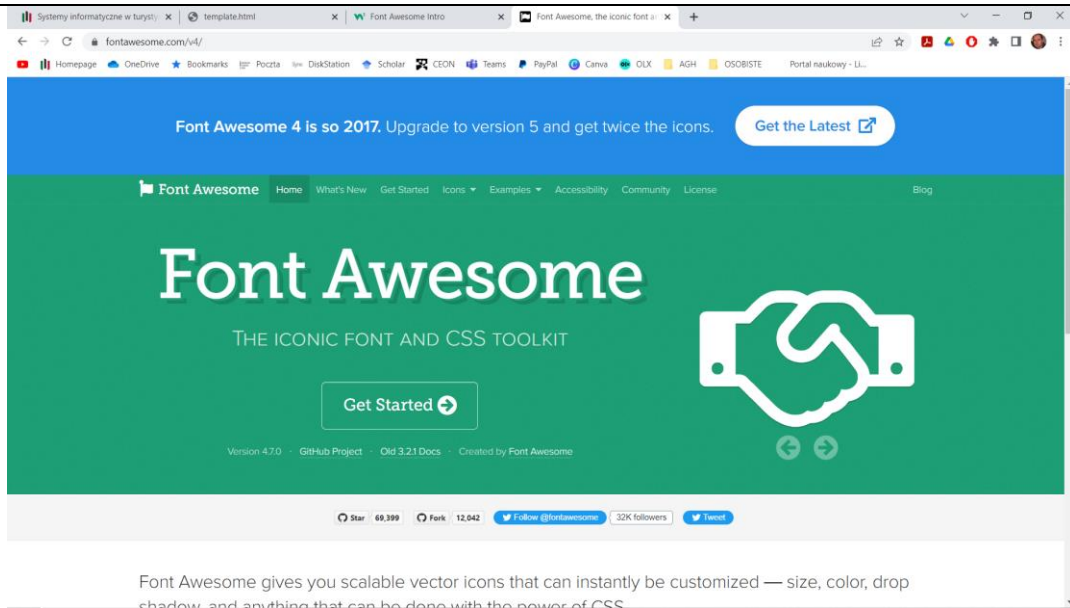
34. Odśwież stronę w przeglądarce www ([Ryc. 13](#)).



Ryc. 13. Wygląd strony www po zdefiniowaniu wyglądu elementu HOME o klasie `active` i pozostałych elementów w stanie po najechaniu na nie kursorem myszy (`li:hover`)



35. Dodamy teraz do każdego elementu menu ikonki. Skorzystamy z usługi serwisu **Font Awesome** udostępniającego online tysiące darmowych ikon. Otwórz stronę serwisu [Font Awesome](#) (ver. 4.7). Jeżeli chcemy skorzystać z najnowszej wersji portalu, darmowe korzystanie z ikon wymaga założenia konta i zalogowania. My jednak zostaniemy przy wersji nie wymagającej zakładania konta.



Ryc. 14. Strona Font Awesome



36. Do sekcji `<head></head>` pliku `index.html` skopiuj następujący

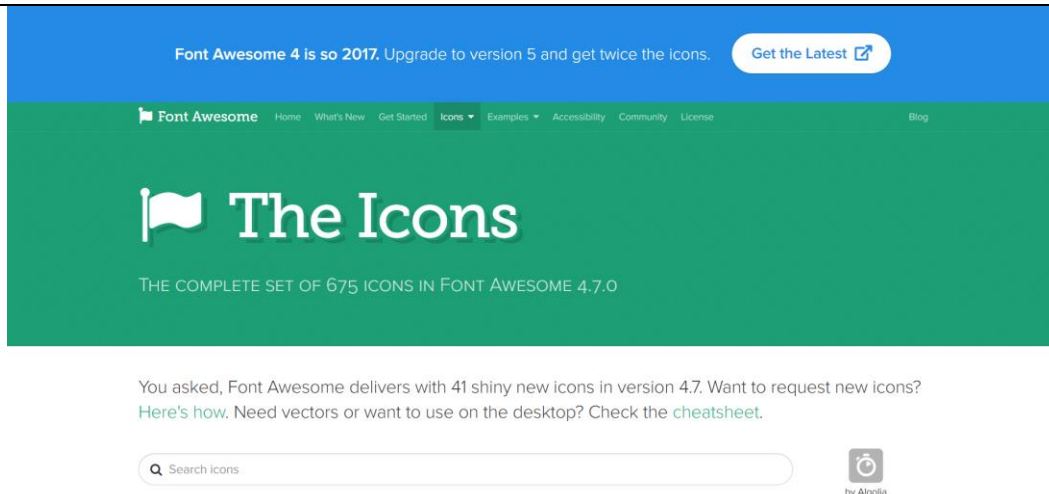
kod: `<link`

```
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.css" rel="stylesheet"
type='text/css'>
```

```
<head>
  <title>Firma turystyczna</title>
  <meta name="description" content="Strona firmy turystycznej. Oferta
wyjazdów zagranicznych, eventów, wycieczek">
  <meta name="keywords" content="Kraków, turystyka, geoturystyka,
wczasy, wycieczki, eventy">
  <meta charset="UTF-8">
  <link rel="stylesheet" type="text/css" href="./style1.css">
  <link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.css">
</head>
```



37. Kliknij w portalu Font Awesome na zakładkę **Icons** (Ryc. 15).



Ryc. 15. Wyszukiwarka ikon portalu Font Awesome




38. W polu formularza wyszukiwarki wpisz frazę „home”. Kliknij na ikonę domku. Z ramki widocznej na [Ryc. 16](#) skopiuj kod HTML udostępniający ikonę `<i class="fa fa-home" aria-hidden="true"></i>` (lub podobny). Zauważ, że ikony mają przypisane dwie klasy:

- 1) `fa` – służącą do symbolizacji wszystkich ikon w projekcie na raz,
- 2) `fa-home` – klasa do indywidualnego zarządzania poszczególnymi ikonami (tu – do stylizacji ikony `home`).



After you get **up and running**, you can place Font Awesome icons just about anywhere with the `<i>` tag:

 fa-home

```
<i class="fa fa-home" aria-hidden="true"></i>
```

Note: to improve **web accessibility**, we recommend using `aria-hidden="true"` to hide icons used purely for decoration.

Ryc. 16. Ikona Home w portalu Font Awesome ver. 4.7

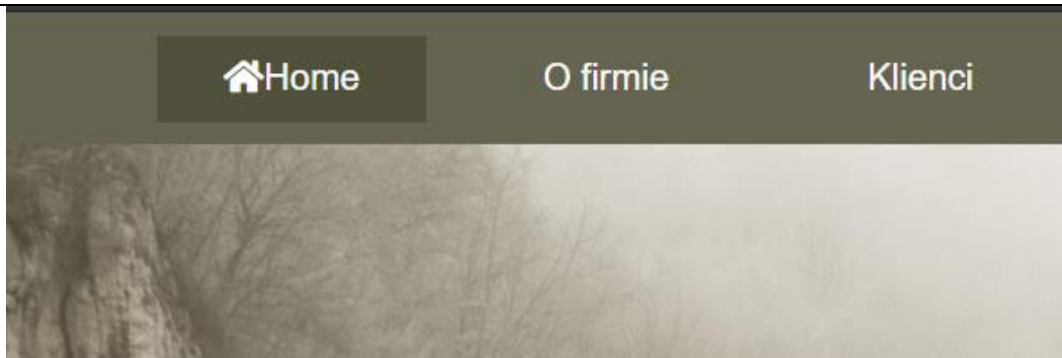


39. Przejdź do edycji pliku `index.html`. Wklej skopiowany kod udostępniający ikonę do wnętrza linku ``.

```
<div id="menu_bar">
  <ul>
    <li class="active"><a href="" title="Strona główna"><i class="fa fa-home" aria-hidden="true" ></i> Home</a> </li>
    <li><a href="" title="O firmie">O firmie</a></li>
    <li><a href="" title="Klienci">Klienci</a></li>
    <li><a href="" title="Oferta">Oferta</a></li>
    <li><a href="" title="Terminy">Terminy</a></li>
    <li><a href="" title="Cennik">Cennik</a></li>
    <li><a href="" title="Kontakt">Kontakt</a></li>
  </ul>
</div>
```



40. Odśwież stronę w przeglądarce www ([Ryc. 17](#)).



Ryc. 17. Fragment projektowanej strony z wklejoną dynamiczną ikoną z serwisu Font Awesome



41. Wyszukaj ikony odpowiednie dla pozostałych linków:

- O firmie
- Klienci
- Oferta
- Terminy
- Cennik
- Kontakt

Ze swojej strony proponuję ikony:

- O firmie -> `<i class="fa fa-cogs"></i>`
- Klienci -> `<i class="fa fa-handshake-o"></i>`
- Oferta -> `<i class="fa fa-map"></i>`
- Terminy -> `<i class="fa fa-calendar"></i>`
- Cennik -> `<i class="fa fa-credit-card"></i>`
- Kontakt -> `<i class="fa fa-address-card"></i>`



42. Odśwież stronę w przeglądarce www ([Ryc. 18](#)).



Ryc. 18. Widok linków uzupełnionych ikonami Font Awesome

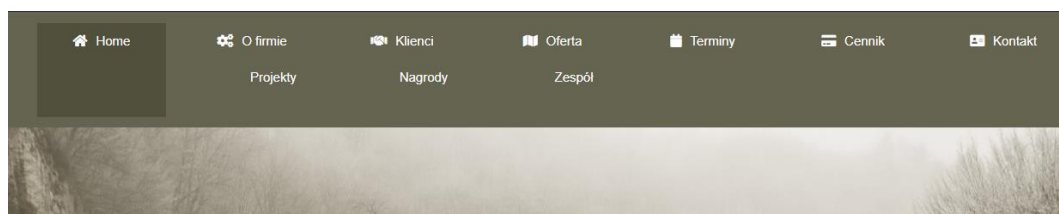
	<p>43. Ikony brzydko przylegają do napisów linków. Musimy odsunąć je od siebie. W pliku <code>style.css</code> utwórz deklarację dotyczącą wszystkich ikon (<code>.fa</code>) znajdujących się wewnątrz bloku o identyfikatorze <code>#menu-bar</code>:</p> <pre>#menu_bar .fa { margin-right: 8px; }</pre>
	<p>44. Odśwież stronę w przeglądarce www (Ryc. 19).</p>  <p>Ryc. 19. Widok linków po modyfikacji prawego marginesu wokół ikon</p>
	<p>45. Czcionki menu są za duże. Dodajmy do sekcji <code><body></body></code> ustawienie wielkości czcionek na 12 pikseli:</p> <pre>body{ background-image: url(./images/DSC_0075.jpg); background-size: cover; font-family: sans-serif; font-size: 12px; }</pre>
	<p>46. Odśwież stronę w przeglądarce www (Ryc. 20).</p>  <p>Ryc. 20. Widok linków po modyfikacji wielkości czcionki</p>
	<p>47. Przejdziemy teraz do zdefiniowania dla linków „O firmie” oraz „Oferta” elementów submenu pierwszego poziomu. Przejdźmy do edycji dokumentu <code>index.html</code> i do wnętrza elementu listy „O firmie” dodajmy element blokowy o klasie <code>submenu-1</code> oraz kolejną listę</p>

nieuporządkowaną:

```
<li><a href="" title="O firmie"><i class="fa fa-cogs"></i>O firmie</a>
  <div class="submenu-1">
    <ul>
      <li><a href="" title="Projekty">Projekty</a></li>
      <li><a href="" title="Nagrody">Nagrody</a></li>
      <li><a href="" title="Zespół">Zespół</a></li>
    </ul>
  </div>
</li>
```



48. Odśwież stronę w przeglądarce www ([Ryc. 21](#)).



Ryc. 21. Menu po dodaniu do kodu HTML elementów klasy `submenu-1`



49. Elementy `submenu-1` pojawiły się na stronie www. Są widoczne permanentnie, niezależnie od tego czy najedziemy kursorem na przycisk linku „O firmie” czy też nie. Zajmiemy się teraz schowaniem tych elementów, aby w przyszłości móc je ukazać na skutek najechania kursorem na przycisk „O firmie”. Dodajmy do pliku `style.css` zapis:

```
.submenu-1{
  display: none;
}
```



50. Dzięki dodaniu zapisu z punktu 49 elementy submenu zostały ukryte, a strona ponownie wygląda jak na [Ryc. 20](#).



51. Teraz musimy pokazać elementy `submenu-1` w momencie najechania kursorem myszy na link „O firmie” lub najechania kursorem na dowolny element będący wewnątrz bloku `submenu-1`.

Utwórzmy deklarację dla:

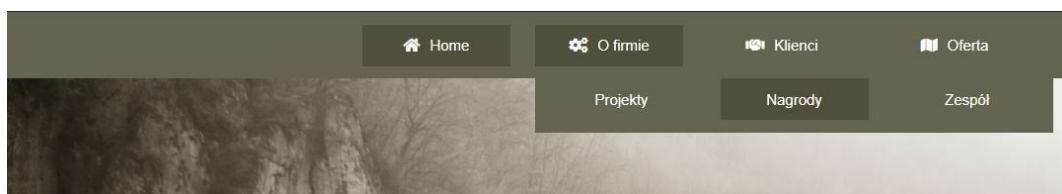
- pseudoklasy `li:hover` wewnątrz `ul` i wewnątrz bloku o identyfikatorze `menu_bar`,
- klasy `submenu-1`:

```
#menu_bar ul li:hover .submenu-1{  
  display: block;  
  position: absolute;  
  background: rgb(100,100,80);  
  margin-top: 10px;  
  margin-left: -10px;  
}
```

Zastosowana właściwość `display: block;` powoduje, że element generuje obiekt, który ma się zachowywać jak element blokowy (układanie każdego elementu klasy `submenu-1` pod rodzicem). Z kolei `position: absolute;` ([pozycjonowanie bezwzględne](#)) powoduje, że element strony można umieścić w dowolnym miejscu na stronie, a jego współrzędne są określane poprzez zdefiniowanie odległości od lewej i górnej krawędzi okna przeglądarki (gdy element przodka ma pozycjonowanie typu `static`) bądź do zewnętrznych krawędzi marginesów wewnętrznych najbliższego elementu rodzica posiadającego pozycję każdą inną tylko nie `static`.



52. Odśwież stronę w przeglądarce [www](#) ([Ryc. 22](#)). Widok przedstawia sytuację po najechaniu kursorem na przycisk „Nagrody”.



Ryc. 22. Menu po zdefiniowaniu wyświetlania elementu klasy `submenu-1`



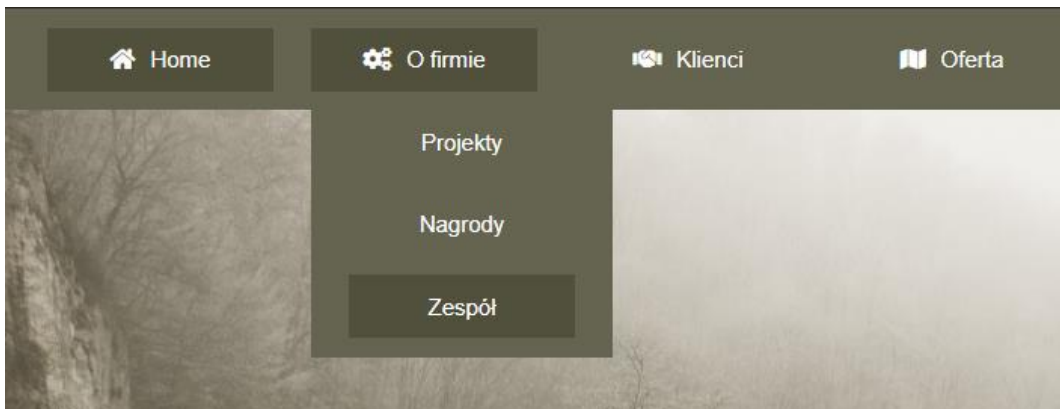
53. Ułożymy teraz elementy `submenu-1` pionowo – jeden pod drugim. Utwórzmy deklarację elementu `` wewnątrz obiektu klasy `submenu-1`.

```
#menu_bar ul li:hover .submenu-1 ul{
  display: block;
  margin: 10px;
}
```

Zastosowana właściwość `display: block;` tym razem spowoduje, że elementy listy nieuporządkowanej `submenu-1` zostaną potraktowane jak typowe elementy blokowe i w związku z tym zostaną ułożone pionowo jeden pod drugim.



54. Odśwież stronę w przeglądarce [www](#) ([Ryc. 23](#)). Widok przedstawia sytuację po najechaniu kursorem na przycisk „Zespół”.



Ryc. 23. Menu po zdefiniowaniu sposobu wyświetlania elementu `` wewnątrz klasy `submenu-1`



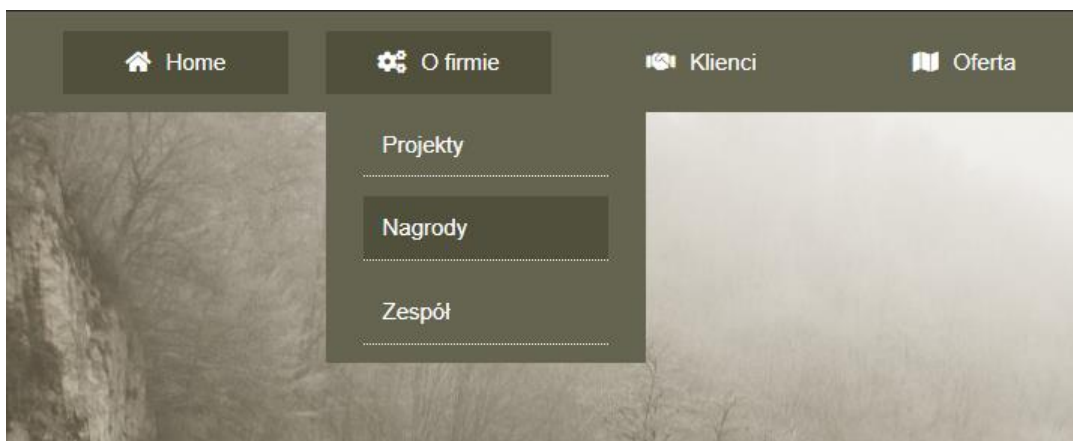
55. Powiększmy nieco bloki podmenu i stwórzmy kropkowane linie oddzielające poszczególne przyciski. Dzięki temu menu stanie się bardziej czytelne.

```
#menu_bar ul li:hover .submenu-1 ul li{
  width: 130px;
  padding: 10px;
  border-bottom: 1px dotted #fff;
}
```

```
text-align: left;
}
```



56. Odśwież stronę w przeglądarce www (Ryc. 24). Widok przedstawia sytuację po najechaniu kursorem na przycisk „Nagrody”.



Ryc. 24. Menu po zmianie szerokości elementów `` submenu-1 oraz po dodaniu linii kropkowych

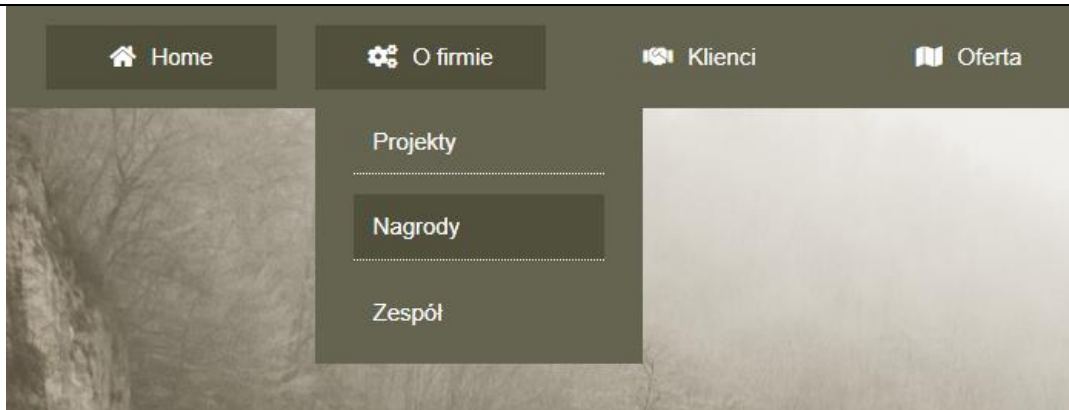


57. Linia kropkowa pod ostatnim elementem submenu nie jest potrzebna i nie wygląda dobrze. Usuńmy ją spod ostatniego przycisku. Utwórzmy pseudoklasę `last-child` ostatniego elementu `` submenu-1:

```
#menu_bar ul li:hover .submenu-1 ul li:last-child{
  border-bottom: none;
}
```



58. Odśwież stronę w przeglądarce www (Ryc. 25). Widok przedstawia sytuację po najechaniu kursorem na przycisk „Nagrody”.



Ryc. 25. Menu po usunięciu pod ostatnim elementem `` submenu-1 linii kropkowej

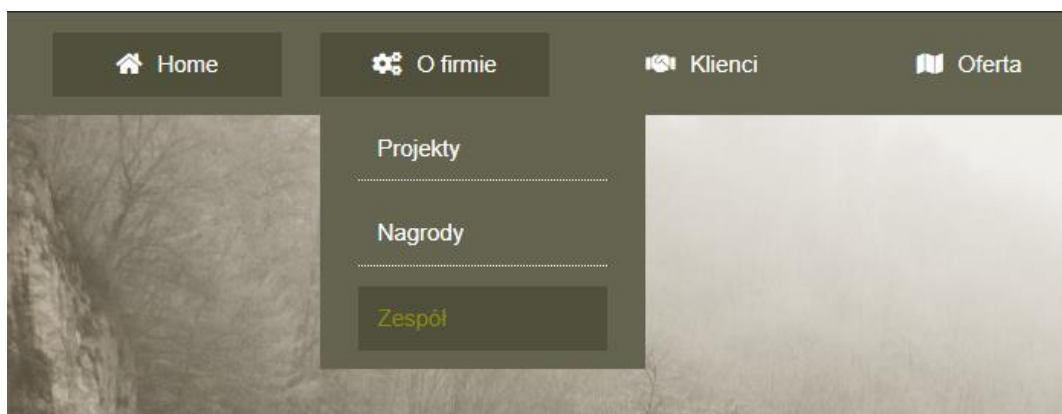


59. Dobrze wyglądałoby gdyby po najechaniu na odpowiedni napis przycisku `submenu-1` zmieniał się jego kolor. Wystylizujmy to. W tym celu dodajmy nagłówek deklaracji pseudoklasy `a:hover`: wewnątrz elementów `` `submenu-1`.

```
#menu_bar ul li:hover .submenu-1 ul li a:hover{
  color: rgb(140,140,10);
}
```



60. Odśwież stronę w przeglądarce [www](#) (Ryc. 26). Widok przedstawia sytuację po najechaniu kursorem na napis przycisku „Zespół”.



Ryc. 26. Menu po zmianie koloru tekstu przycisków po ich najechaniu kursorem myszy



61. Skopiujmy kod HTML całego utworzonego podmenu przycisku „Oferta” firmie”:

```
<div class="submenu-1">
  <ul>
    <li><a href="" title="Projekty">Projekty</a></li>
    <li><a href="" title="Nagrody">Nagrody</a></li>
    <li><a href="" title="Zespół">Zespół</a></li>
  </ul>
</div>
```

Wklejmy go jako podmenu przycisku „Oferta”:

```
<li><a href="" title="Oferta"><i class="fa fa-map"></i>Oferta</a>
  <div class="submenu-1">
    <ul>
      <li><a href="" title="Projekty">Projekty</a></li>
      <li><a href="" title="Nagrody">Nagrody</a></li>
      <li><a href="" title="Zespół">Zespół</a></li>
    </ul>
  </div>
</li>
```



62. Zmieńmy napisy przycisków podmenu przycisku „Oferta” na:

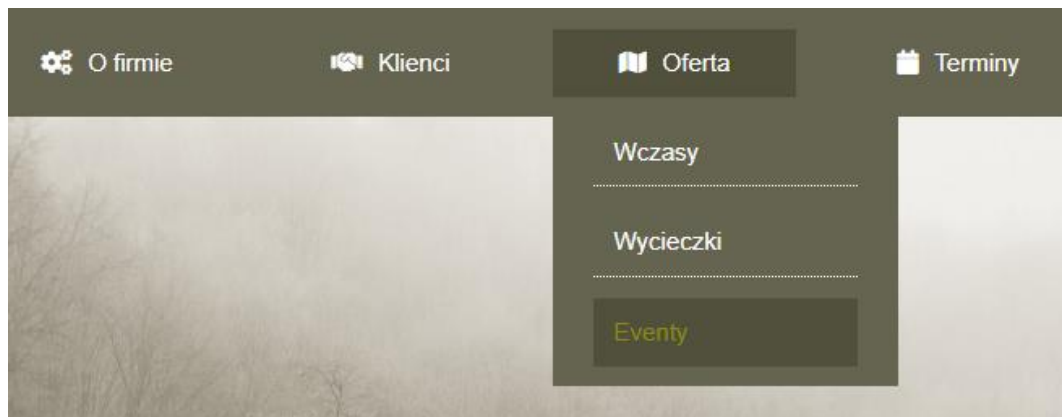
- Projekty -> Wczasy
- Nagrody-> Wycieczki
- Zespół -> Eventy

```
<li><a href="" title="Oferta"><i class="fa fa-map"></i>Oferta</a>
  <div class="submenu-1">
    <ul>
      <li><a href="" title="Wczasy">Wczasy</a></li>
      <li><a href="" title="Wycieczki">Wycieczki</a></li>
      <li><a href="" title="Eventy">Eventy</a></li>
    </ul>
  </div>
</li>
```

UWAGA! – nie zapomnij podmienić także tytuły linków.



63. Odśwież stronę w przeglądarce www (Ryc. 27). Widok przedstawia sytuację po najechaniu kursorem na przycisk „Eventy”.



Ryc. 27. Menu po dodaniu submenu-1 przycisku „Oferta”



64. W kolejnym etapie prac nad stroną, do linków „Wczasy” oraz „Wycieczki” submenu-1 dodamy submenu 2-go poziomu. Zanim to jednak zrobimy dobrym pomysłem byłoby poinformowanie internautów, że te dwa przyciski z submenu oferty firmy mają jakieś dodatkowe linki. Zaproponujemy to dodając po prawej stronie napisów linków „Wczasy” oraz „Wycieczki” ikonki ze strzałeczką.

Skopiujemy dowolną deklarację ikonek (np. ikonki Home) i wklejmy kod na prawo od każdego tekstu wskazanych linków submenu-1.

```
<li><a href="" title="Oferta"><i class="fa fa-map"></i>Oferta</a>
  <div class="submenu-1">
    <ul>
      <li><a href="" title="Projekty">Wczasy<i class="fa fa-home"></i></a></li>
      <li><a href="" title="Nagrody">Wycieczki<i class="fa fa-home"></i></a></li>
      <li><a href="" title="Zespół">Eventy</a></li>
    </ul>
  </div>
</li>
```

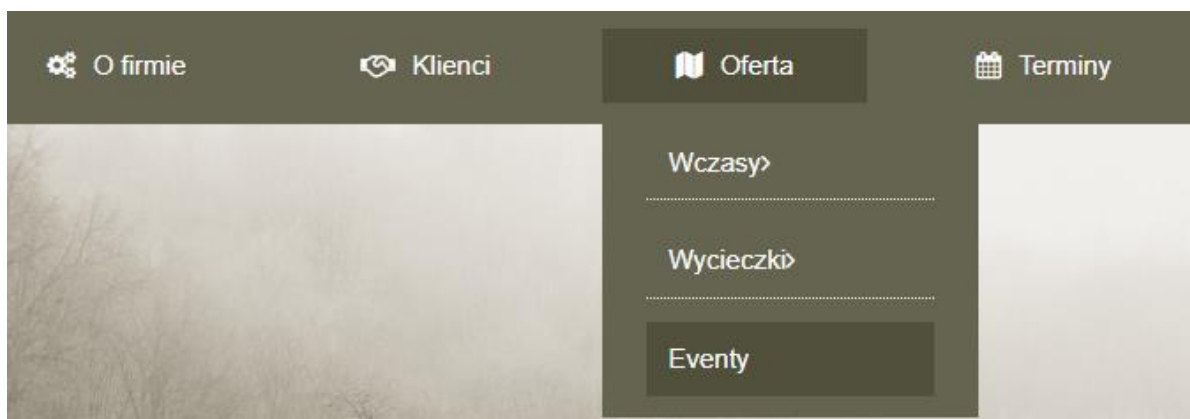


65. Zmieńmy teraz klasę dodanych ikon na `fa fa-angle-right`.

```
<li><a href="" title="Oferta"><i class="fa fa-map"></i>Oferta</a>
  <div class="submenu-1">
    <ul>
      <li><a href="" title="Projekty">Wczasy<i class="fa fa-angle-
right"></i></a></li>
      <li><a href="" title="Nagrody">Wycieczki<i class="fa fa-angle-
right"></i></a></li>
      <li><a href="" title="Zespół">Eventy</a></li>
    </ul>
  </div>
</li>
```



66. Odśwież stronę w przeglądarce www ([Ryc. 28](#)). Widok przedstawia sytuację po najechaniu kursorem na przycisk „Eventy”.


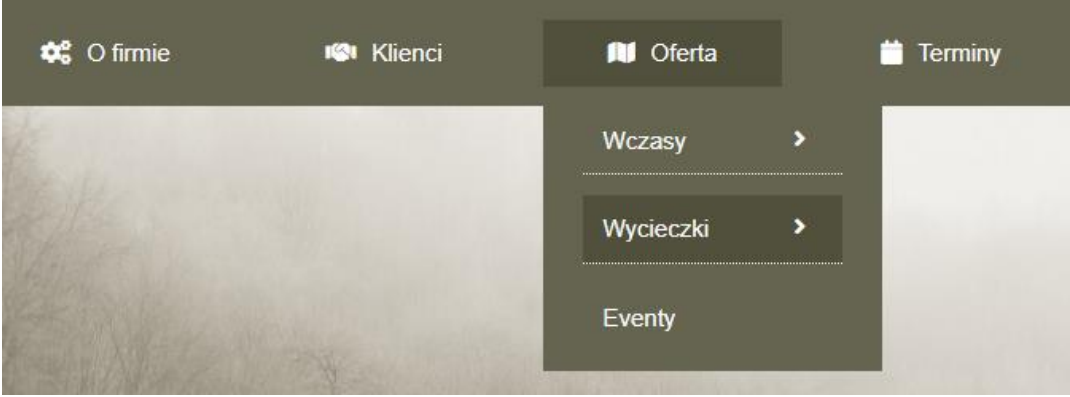




Ryc. 28. Wygląd elementów `submenu-1` po dodaniu z prawej strony napisów „Wczasy” oraz „Wycieczki” strzałeczek sugerujących istnienie podmenu 2-go poziomu

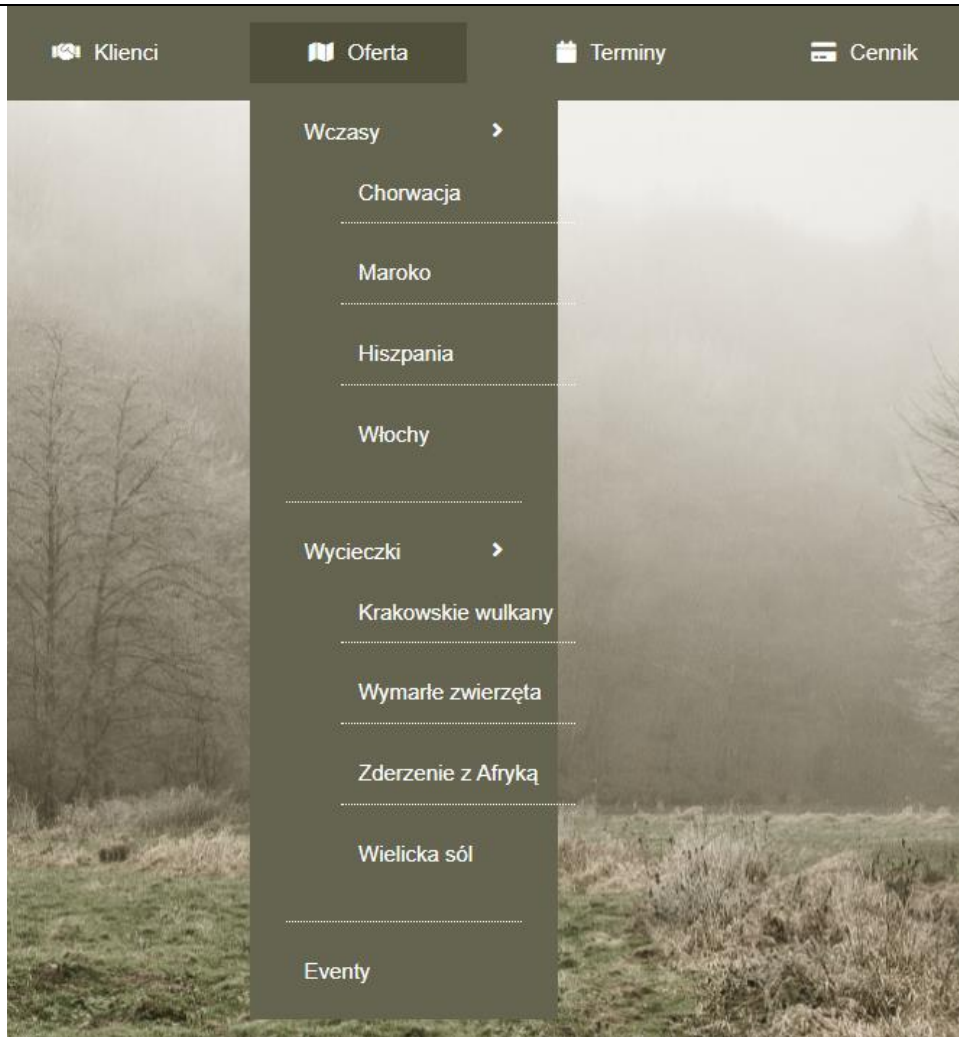


67. Strzałeczki sugerujące istnienie podmenu 2-go poziomu obok przycisków „Wczasy” oraz „Wycieczki” przylegają do samych napisów. Zajmijmy się naprawieniem tych detali. W pliku `style.css` dodajmy deklarację klasy `.fa-angle-right`.

```
.fa-angle-right{
  float: right;
```

	<p>68. Odśwież stronę w przeglądarce www (Ryc. 29). Widok przedstawia sytuację po najechaniu kursorem na przycisk „Wycieczki”.</p>  <p>Ryc. 29. Wygląd submenu-1 po dodaniu z prawej strony napisów „Wczasy” oraz „Wycieczki” strzałeczek sugerujących istnienie podmenu 2-go poziomu</p>
	<p>69. W pliku <code>index.html</code> utwórzmy teraz submenu 2-go poziomu. Elementom listy <code></code> przycisków „Wczasy” i „Wycieczki” nadajmy klasę <code>class="hover-me"</code>:</p> <pre data-bbox="209 1227 1410 1503"> <li class="hover-me">Wczasy<i class="fa fa-angle-right"></i> <li class="hover-me">Wycieczki<i class="fa fa-angle-right"></i> Eventy </pre>
	<p>70. Wewnątrz elementu listy „Wczasy” utwórzmy kod HTML submenu 2-go poziomu:</p> <pre data-bbox="209 1659 1410 2002"><li class="hover-me">Wczasy<i class="fa fa-angle-right"></i> <div class="submenu-2"> Chorwacja Maroko Hiszpania Włochy </div> </pre>

	<pre> </div> </pre>
	<p>71. Podobnie stwórzmy podmenu dla elementu listy „Wycieczki”:</p> <pre> <li class="hover-me">Wycieczki<i class="fa fa-angle-right"></i> <div class="submenu-2"> Krakowskie wulkany Wymarłe zwierzęta Zderzenie z Afryką Wielicka sól </div> </pre>
	<p>72. Odśwież stronę w przeglądarce www (Ryc. 30). Widok przedstawia sytuację po najechaniu kursorem na przycisk „Oferta”.</p>



Ryc. 30. Wygląd submenu-1 po dodaniu submenu 2-go poziomu



73. Jak widać, najechanie kursorem myszy na przycisk „Oferta” powoduje wyświetlenie submenu-1 oraz submenu-2. Musimy submenu-2 ukryć. Powinno się ono pojawić dopiero po naciśnięciu przycisków „Wczasy” lub „Wycieczki”. Ukryjmy elementy klasy submenu-2.

```
.submenu-2{
  display: none;
}
```

W wyniku dodania deklaracji wygląd menu rozwijanego wrócił do postaci znanej z [Ryc. 29](#).

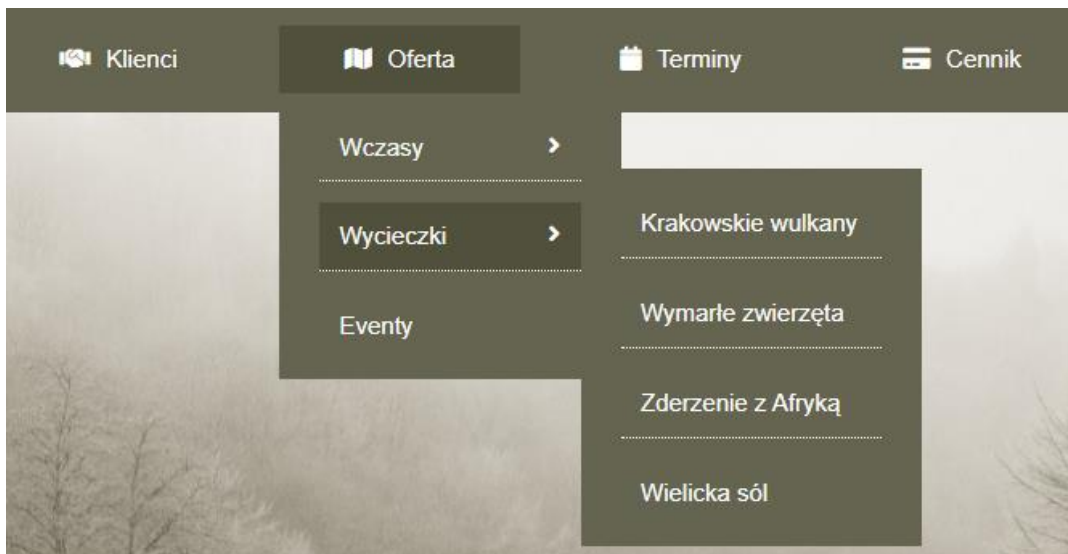


74. Zajmijmy się teraz pojawianiem się submenu-2 w momencie najechania na przyciski „Wczasy” lub „Wycieczki”. Utwórzmy deklarację:

```
.hover-me:hover .submenu-2{
  position: absolute;
  display: block;
  margin-top: -40px;
  margin-left: 120px;
  background: rgb(100,100,80);
}
```



75. Odśwież stronę w przeglądarce www (Ryc. 31). Widok przedstawia sytuację po najechaniu kursorem na przycisk „Wycieczki”.



Ryc. 31. Wygląd submenu-2 po najechaniu kursorem na przycisk „Wycieczki”



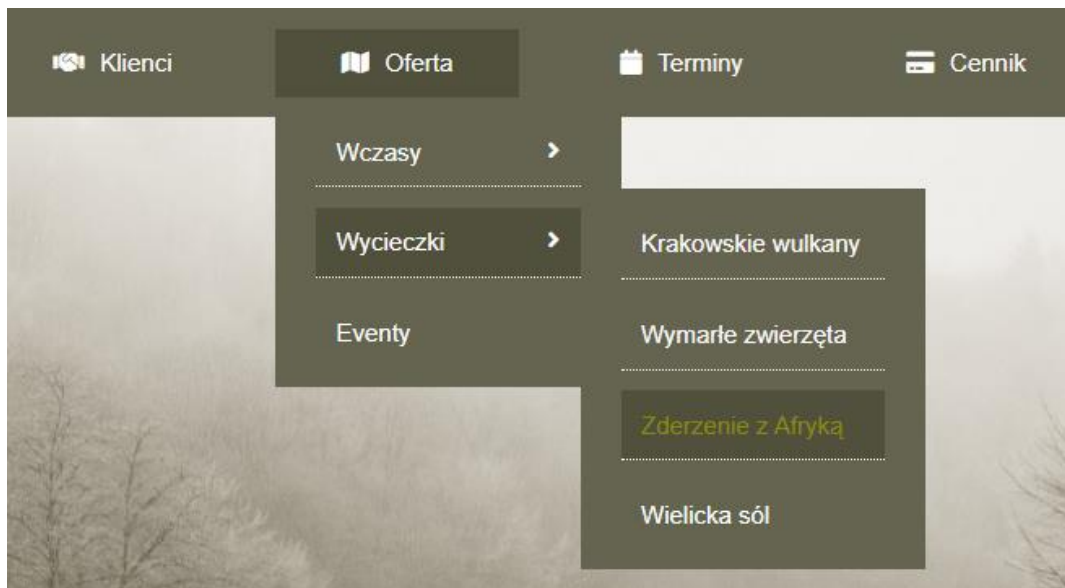
76. Menu 2-go poziomu dość dobrze wpasowuje się w nasz projekt. Jedyne co powinniśmy zrobić to przesunąć je nieco w dół. Zmieńmy jego położenie modyfikując `margin-top` na `-33`.

```
.hover-me:hover .submenu-2{
  position: absolute;
  display: block;
```

```
margin-top: -33px;
margin-left: 120px;
background: rgb(100,100,80);
}
```



77. Odśwież stronę w przeglądarce www (Ryc. 32). Widok przedstawia sytuację po najechaniu kursorem na przycisk „Zderzenie z Afryką”.



Ryc. 32. Wygląd submenu-2 po modyfikacji jego położenia



78. Menu mamy ukończone. Teraz możemy zająć się pozostałymi częściami strony. Na początek dodajmy blok o identyfikatorze `content` z pięcioma akapitami [lorem ipsum](#) w paragrafach.

```
<div id="content">
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam condimentum erat volutpat, tincidunt tellus nec, porttitor magna. Etiam vitae ornare ex, eget auctor mi. Nullam ullamcorper sem vitae quam posuere sollicitudin. Aliquam vitae dui rutrum, placerat velit sed, condimentum dui. Integer accumsan accumsan cursus. Sed turpis metus, auctor at risus vel, porta varius justo. Nulla tellus quam, finibus vel lacus vitae, consectetur viverra magna. In vel velit laoreet nulla gravida vehicula. Praesent quis hendrerit orci. Sed ullamcorper, nisi sed sollicitudin faucibus, eros nunc accumsan velit, vel euismod magna nunc a nulla. Etiam libero lorem, gravida ac tortor in, vestibulum porta turpis.</p>
```

```
<p>Proin auctor leo ac velit tincidunt, ac condimentum nibh posuere. Praesent id congue ex. Vestibulum vestibulum, tortor sed consequat
```

```
ultricies, neque mi euismod tellus, sed vestibulum ipsum dui posuere est.
Donec tincidunt viverra dui a iaculis. Maecenas tempor tortor a purus
interdum vestibulum. Nunc accumsan dictum maximus. Nullam quis purus
suscipit, mattis ligula lacinia, malesuada libero. Duis nec mi id felis
tristique lacinia eget vel diam. Nulla tincidunt a lectus ut sagittis.
Nullam lacinia lectus id elit facilisis, ac cursus dui consequat. In
facilisis tellus non ante facilisis maximus. Nunc est ligula, viverra ut
arcu eget, gravida volutpat nunc. Fusce pulvinar, nibh nec fermentum
laoreet, erat tellus tempus elit, in lacinia urna ligula vel enim.
Vestibulum vel arcu eros. Sed euismod ullamcorper justo id laoreet.
Pellentesque nec aliquet eros.</p>
```

```
<p>Pellentesque quam massa, maximus eu cursus nec, iaculis vel nibh.
Sed ullamcorper leo sed ullamcorper pellentesque. Morbi semper auctor
elit, eget feugiat mi consequat sit amet. Ut convallis erat neque, in
eleifend nulla auctor vel. Vestibulum ante ipsum primis in faucibus orci
luctus et ultrices posuere cubilia curae; Quisque ornare felis at aliquam
maximus. Quisque malesuada, nibh at egestas pulvinar, ante ante pharetra
dui, in placerat purus arcu vitae justo. Aliquam rhoncus lectus quis felis
porttitor porttitor. Curabitur nunc ipsum, malesuada id augue in, dictum
feugiat magna. In euismod mi sit amet risus consectetur semper. Phasellus
semper erat nec risus congue convallis. Curabitur vel libero odio.</p>
```

```
<p>Praesent id odio massa. Vivamus lobortis tellus augue. Donec metus
est, dapibus at ultricies vitae, lobortis et felis. Nulla magna tellus,
dapibus convallis nunc et, aliquet ultricies neque. Sed posuere, mauris
sed convallis ultricies, nunc nulla porta urna, et convallis augue libero
id sapien. Quisque id ligula id sem rutrum euismod. Sed placerat eros eu
erat blandit, et malesuada ligula auctor. Pellentesque ut lacinia elit, a
euismod nulla. Ut id condimentum tellus, blandit rhoncus massa. Cras ex
diam, iaculis quis mauris non, auctor imperdiet risus. Curabitur eu varius
ante. Mauris eleifend urna lorem, a ultrices est ultricies ut. Phasellus
congue odio sem, porta condimentum orci laoreet vitae. Donec vitae nulla
ornare, luctus risus vel, mattis metus. Nulla accumsan tristique
ultricies. Vestibulum vel orci id urna vulputate maximus.</p>
```

```
<p>Quisque bibendum commodo velit, non venenatis erat aliquet quis. Ut
nisl tellus, lobortis non sodales ac, maximus nec ipsum. In vestibulum
egestas dolor. In egestas ut diam ut pulvinar. Suspendisse scelerisque
nulla dolor, ornare ultrices lorem vehicula pulvinar. Nunc sodales iaculis
fermentum. Donec libero tortor, pharetra in pharetra ut, congue quis nunc.
Suspendisse vitae tortor quam. Morbi ac ante elit. Mauris ut eleifend
quam. Phasellus malesuada eros nec faucibus sagittis. Nunc ac orci at
velit lacinia ultricies ut id odio. Maecenas tristique mollis elit, sit
amet rhoncus mauris pharetra euismod. Nunc tempor pellentesque urna, et
aliquam tortor pellentesque id. Vestibulum ante ipsum primis in faucibus
orci luctus et ultrices posuere cubilia curae;</p>
```

```
</div>
```



79. Odśwież stronę w przeglądarce www ([Ryc. 33](#)).



Ryc. 33. Widok strony po dodaniu treści bloku content

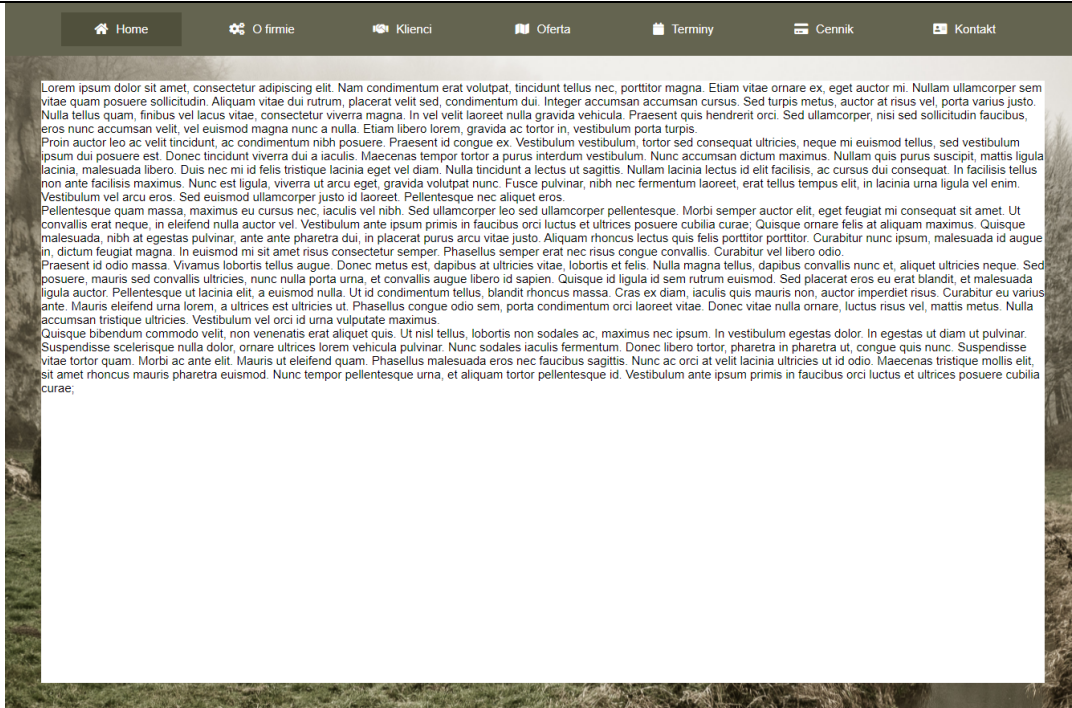


80. Wystylizujemy wygląd bloku content. Zgodnie z Ryc. 2 pudełko powinno mieć szerokość 1000 pikseli i powinno być wyśrodkowane względem okna przeglądarki. Dodatkowo kolor tła ustawmy na biały, oddzielmy blok content od bloku menu_bar o 25 pikseli i dodajmy minimalną wysokość 600 pikseli.

```
#content{
  width: 1000px;
  margin-left: auto;
  margin-right:auto;
  background-color: white;
  margin-top: 25px;
  min-height: 600px;
}
```



81. Odśwież stronę w przeglądarce www (Ryc. 34).



Ryc. 34. Widok strony po zmianie wyglądu bloku content

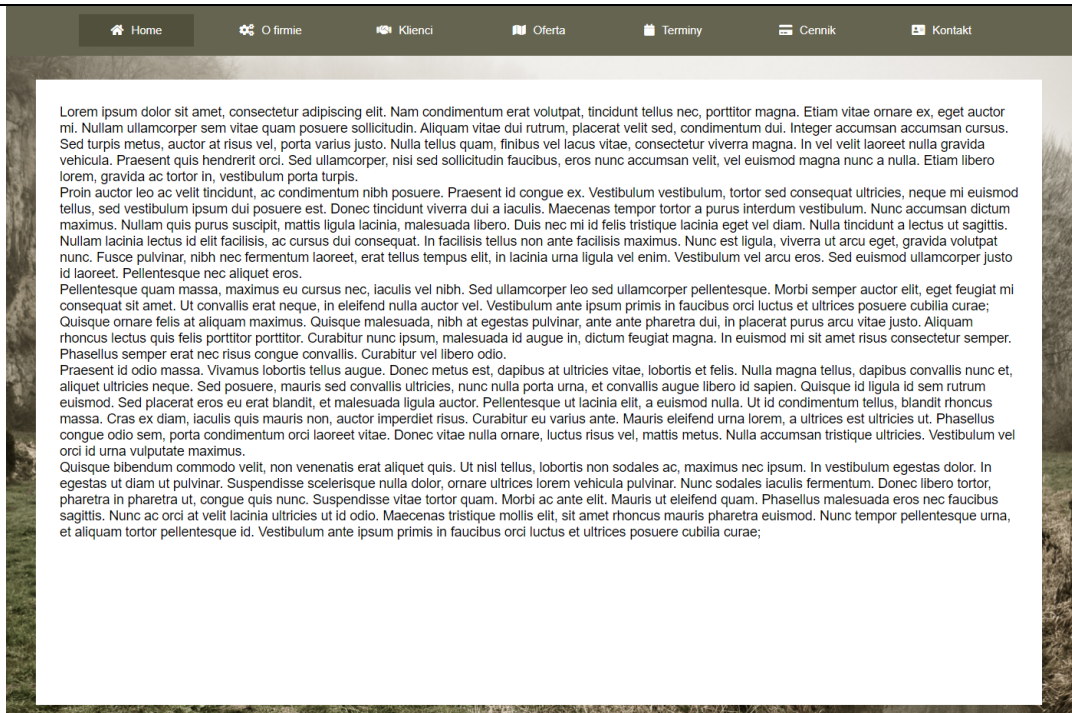


82. Odsuńmy tekst bloku content od krawędzi bloku i zwiększmy czcionkę.

```
#content{
    width: 1000px;
    margin-left: auto;
    margin-right:auto;
    background-color: white;
    margin-top: 25px;
    min-height: 600px;
    padding: 25px;
    font-size: 14px;
}
```



83. Odśwież stronę w przeglądarce www (Ryc. 35).



Ryc. 35. Widok strony po modyfikacji wyglądu bloku content

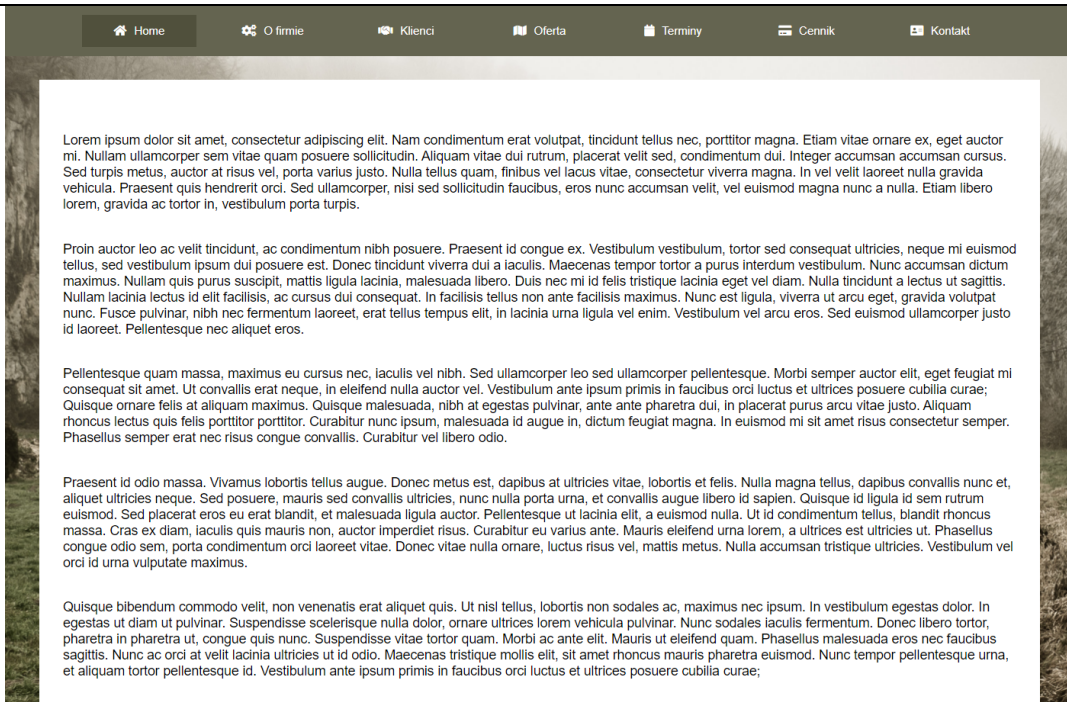


84. Kolejne paragrafy są do siebie brzydko przyrośnięte. Rozdzielmy je dodając do stylu paragrafu margines wewnętrzny 30 pikseli.

```
p{
    padding-top: 30px;
}
```



85. Odśwież stronę w przeglądarce www ([Ryc. 36](#)).



Ryc. 36. Wygląd strony po dodaniu marginesów do paragrafów bloku content



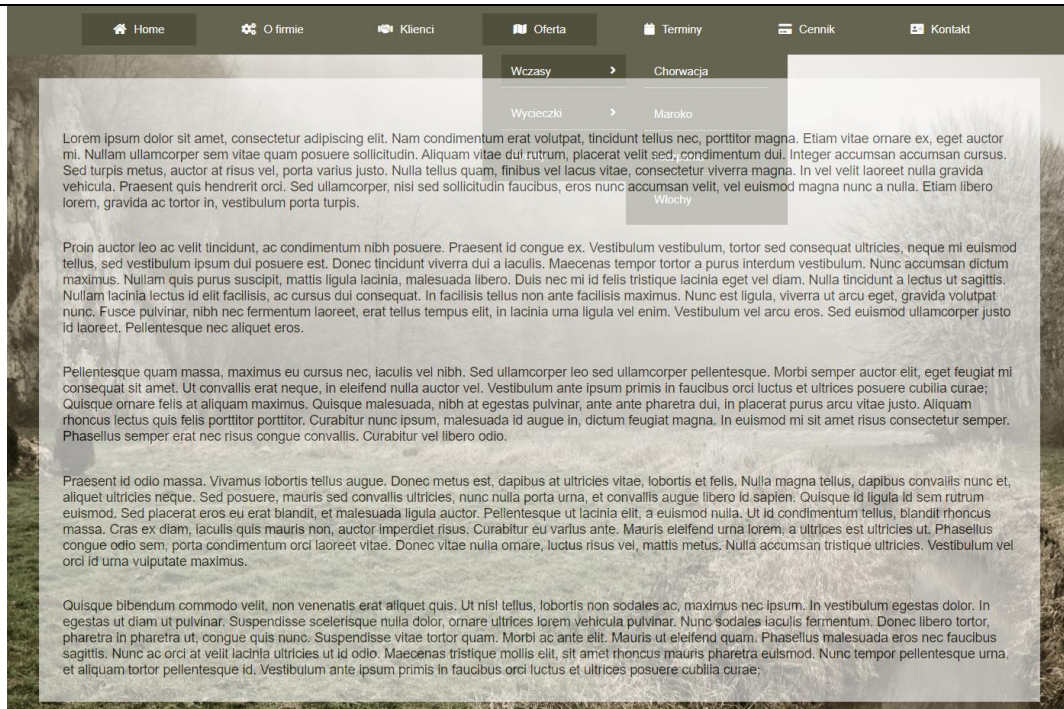
86. Blok content przykrywa zdjęcie przez co jego zamieszczenie staje się dyskusyjne. A gdyby tak nadać całemu blokowi content przezroczystość na poziomie 40%?

```

#content{
  width: 1000px;
  margin-left: auto;
  margin-right:auto;
  background-color: white;
  margin-top: 25px;
  min-height: 600px;
  padding: 25px;
  font-size: 14px;
  opacity: 0.6;
}
  
```



87. Odśwież stronę w przeglądarce www (Ryc. 37).



Ryc. 37, Wygląd strony po nadaniu blokowi `content` przezroczystości 40%



88. Blok `content` wygląda teraz świetnie. Powstał jednak inny problem

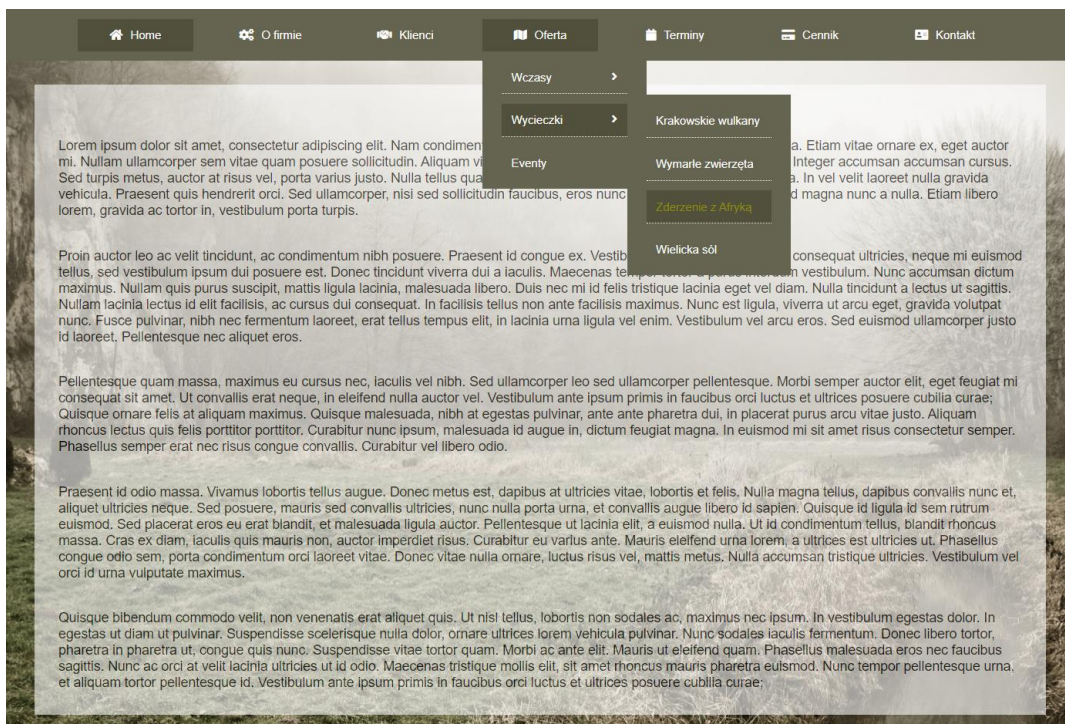
Jak widać na [Ryc. 37](#), w pionowej strukturze strony jest on widoczny wyżej niż menu bloku `menu_bar`. Rozwijane menu jest wyraźnie pod blokiem `content`. Musimy zmienić hierarchię wyświetlania obu w.w. blokom. Służy do tego właściwość `z-index: n`; gdzie n – jest poziomem hierarchii wyświetlania wyrażonym liczbą całkowitą. Im wyższe n tym warstwa wyżej jest wyświetla na scenie. Dodajmy odpowiednie definicje dla bloku `content` i elementów `` wewnątrz `` wewnątrz bloku `menu_bar`.

```
#menu_bar ul li{
  width: 120px;
  margin: 10px;
  padding: 10px;
  box-sizing: border-box;
  z-index: 1;
}
```

```
#content{
  width: 1000px;
  margin-left: auto;
  margin-right:auto;
  background-color: white;
  margin-top: 25px;
  min-height: 600px;
  padding: 25px;
  font-size: 14px;
  opacity: 0.6;
  z-index: 0;
}
```



89. Odśwież stronę w przeglądarce www (Ryc. 38).



Ryc. 38. Wygląd strony www po najechaniu kursorem na przycisk „Zderzenie z Afryką”



90. Poniżej bloku content ale wewnątrz bloku container dodajmy do pliku index.html blok footer tworzący stopkę strony www.

```
<div id="footer">
  &copy; 2021 Wszelkie prawa zastrzeżone.
</div>
```

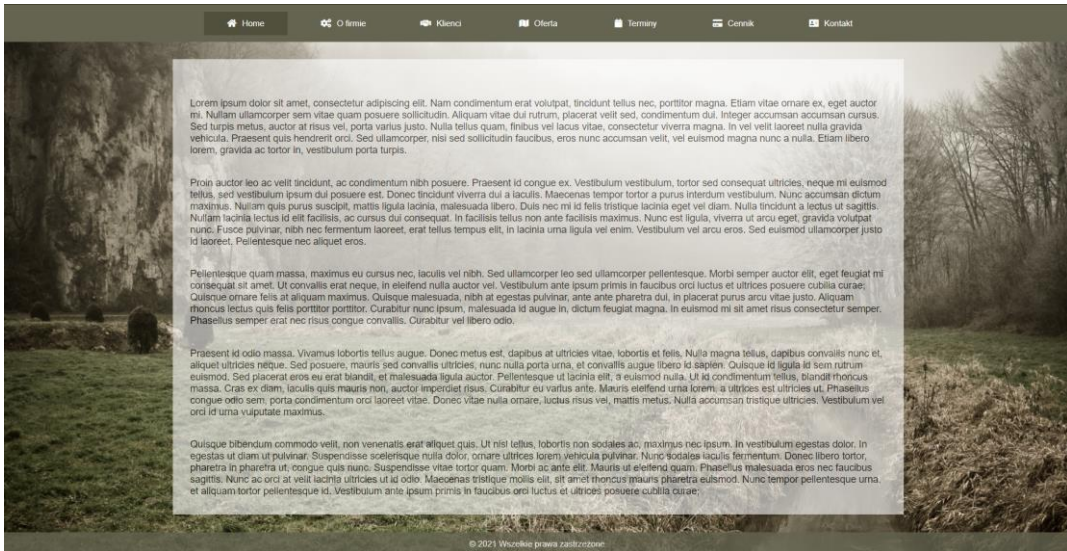


91. Wystylizujemy blok stopki:

```
#footer{
  margin-top: 25px;
  background-color: rgb(100,100,80);
  opacity: 0.8;
  color: #fff;
  padding: 10px;
  text-align: center;
}
```



92. Odśwież stronę w przeglądarce www (Ryc. 39).



Ryc. 39. Ukończony projekt strony www