

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Bakalářská práce

Webové standardy a responzivní webdesign

Kašpar Vít

© 2016 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Vít Kašpar

Informatika

Název práce

Webové standardy a responzivní webdesign

Název anglicky

Web standards and responsive webdesign

Cíle práce

Cílem bakalářské práce je analyzovat stávající možnosti responzivního designu a charakterizovat možné nové postupy. Praktickým výstupem bakalářské práce bude vytvořit komplexní responzivní webovou šablonu na základě výsledků zjištěných v rešeršní části práce.

Metodika

První část bakalářské práce bude věnována explanaci teoretických východisek práce. Informace v této části budou získány studiem odborné literatury zabývající se zvoleným tématem. Praktická část bakalářské práce bude zaměřena na vytvoření responzivní webové šablony. Na základě syntézy teoretických poznatků a výsledků praktické části práce budou formulovány závěry bakalářské práce.

Doporučený rozsah práce

30 – 40 stran

Klíčová slova

Responzivní web, HTML5, CSS3, Webová sémantika, jQuery

Doporučené zdroje informací

CSS Animation: An Interactive Guide [online]. 2014 [cit. 2015-06-25]. Dostupné z:

<http://cssanimationbook.com/>

Dive into HTML5 [online]. 2015 [cit. 2015-06-25]. Dostupné z: <http://kniha.html5.cz/>

Google Inc. [online]. 2015 [cit. 2015-06-25]. Dostupné z:

<https://developers.google.com/web/fundamentals/layouts/rwd-fundamentals/>

LAWSON, B. a SHARP, R. Introducing HTML5. Pearson Education, 2010. ISBN 978-0-321-71796-2

LUBBERS, P., ALBERS, B. a SALIM, F. HTML5: programujeme moderní webové aplikace. Vyd. 1. Brno:

Computer Press, 2011, 304 s. ISBN 978-80-251-3539-6

Mozilla Foundation [online]. 2015 [cit. 2015-06-25]. Dostupné z:

https://developer.mozilla.org/en-US/docs/Web_Development/Responsive_Web_design

Smashing Magazine [online]. 2015 [cit. 2015-06-25]. Dostupné z:

<http://www.smashingmagazine.com/2015/03/31/real-life-responsive-web-design-smashing-book-5/>

UGurus [online]. 2015 [cit. 2015-06-25]. Dostupné z:

<https://www.ugurus.com/responsive-design-best-practices>

W3C [online]. 2015 [cit. 2015-06-25]. Dostupné z: <http://www.w3.org/>

Předběžný termín obhajoby

2015/16 LS – PEF

Vedoucí práce

Ing. Petr Benda, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 28. 10. 2015

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 10. 11. 2015

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 20. 02. 2016

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Webové standardy a responzivní webdesign" jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 11. 3. 2016

Poděkování

Rád bych touto cestou poděkoval panu Ing. Petru Bendovi, Ph. D. jako vedoucímu práce za pomoc při práci na projektu a oběma mým rodičům Ing. Františku Kašparovi a Mgr. Evě Kašparové, kteří mi umožnili studovat vysokou školu a také za jejich podporu během mých studií.

Webové standardy a responzivní webdesign

Souhrn

Bakalářská práce „Webové standardy a responzivní webdesign“ se zabývá metodami a postupy vhodnými pro návrh a vývoj responzivního webu. Součástí práce je také historie vývoje webu a technologií, které využívá. Práce je rozdělena na teoretickou a praktickou část.

Teoretická část se zabývá historií webu, vývojem jazyka HTML až po jeho nejnovější verzi HTML5 a jeho sémantiku, kaskádovými styly CSS a jejich novinkami, historií webových prohlížečů a zejména technikami vhodnými pro vytvoření responzivního webu.

Praktická část je zaměřena na tvorbu responzivního webu s použitím metod a postupů pocházejících z teoretické části.

Klíčová slova: HTML5, responzivní webdesign, CSS3, jQuery, webové prohlížeče, historie webu, W3C

Web standards and responsive webdesign

Summary

The Bachelor's thesis „Web standards and responsive webdesign“ deals with methods and procedures that are suitable for developing responsive website. The thesis also includes familiarization with the origins and history of webdesign and related technologies that it draws upon. The thesis is divided into theoretical and practical subsection.

Theoretical part deals with the history of the web, the development of HTML to its latest version of HTML5 and its semantics, Cascading Style Sheets and its innovations, history of web browsers and particularly techniques suitable for creating responsive web.

The practical part is focused on creating a responsive website using methods and procedures derived from the theoretical subsection.

Keywords: HTML5, responsive webdesign, CSS3, jQuery, web browsers, web history, W3C

Obsah

1. Úvod	1
2. Cíl práce a metodika.....	2
3. Historie a vývoj jazyka HTML.....	3
3.1. HTTP protokol	3
3.2. Doménový systém pojmenování.....	4
3.3. První grafický webový prohlížeč.....	5
3.4. World Wide Web konsorcium a jazyk HTML	5
3.5. Kaskádové styly	6
4. HTML sémantika a kaskádové styly.....	8
4.1. HTML 5.....	8
4.2. CSS3.....	12
4.3. CSS preprocesory	14
5. Webové prohlížeče	16
5.1. Internet Explorer.....	16
5.2. Mozilla Firefox.....	17
5.3. Google Chrome	18
5.4. Opera	19
5.5. Safari.....	19
5.6. Podíl současných prohlížečů na trhu	20
6. Responzivní webdesign.....	22
6.1. Historie	22
6.2. Metody návrhu webových stránek.....	23
6.3. Grafický návrh	23
6.4. Obrázky.....	25
6.5. Responzivní grid systémy	27
6.6. Jednotky	28
6.7. Google speed insights	28
7. Praktická část bakalářské práce	30
7.1. Design	30
7.2. HTML část.....	31
7.3. Animace.....	35

7.4. Fonty	36
7.5. Obrázky	36
7.6. Kaskádové styly	37
8. Závěr	40
9. Přílohy	42
9.1. Seznam obrázků	42
9.2. Seznam tabulek	43

1. Úvod

První webové stránky byly značně omezeny a mimo textový obsah zobrazovaly jen několik málo dalších prvků (př.: nadpisy, seznamy, odstavce, odkazy, obrázky...), čímž byli také tvůrci webů omezeni ve své kreativitě. Převážnou většinu obsahu vytvářeli fanoušci internetu a obsah byl věnován hlavně jejich koníčkům a zálibám. Postupem času, vývojem jazyka HTML a příchodem kaskádových stylů, se dostalo webdesignérům do ruky více možností, jak upravovat nejen obsah, ale také vzhled webových stránek. V roce 2011 bylo po celém světě celkem 35 % uživatelů internetu. V roce 2014 už bylo v České republice 39,5 % uživatelů přistupujících k webovému obsahu prostřednictvím mobilního zařízení jako je mobilní telefon nebo tablet. Vývoj tzv. chytrých mobilních telefonů, které používá v České republice více než 59 % populace, donutil vývojáře ubírat se novým směrem.

Návštěvníky webů bylo třeba na internetových stránkách udržet co nejdéle, nabídnout jim pohodlně obsah, který hledali, a stejně tak jim webové stránky nabídnout rychle a přenést přitom co nejméně dat. Proto se webové stránky začaly upravovat tak, aby se zobrazovaly na všech zařízeních (stolní počítač, tablet a mobilní telefon) ideálně.

Za tímto účelem se řada vývojářů rozhodla usnadnit práci sobě i ostatním a došlo k postupnému vývoji nástrojů, které usnadňovaly zápis kaskádových stylů, HTML kódu i skriptovacích jazyků.

Responzivní webdesign se tak stal světovým fenoménem popisujícím způsoby a postupy, jak upravovat webové stránky v závislosti na velikosti okna webového prohlížeče.

2. Cíl práce a metodika

Cílem bakalářské práce je analýza stávajících možností responzivního designu a charakteristika možných postupů návrhu a vývoje responzivního webu. Praktickým výstupem práce bude vytvoření komplexní responzivní webové šablony na základě výsledků zjištěných v rešeršní části práce.

První část bakalářské práce bude věnována explanaci teoretických východisek práce. Informace v této části budou získány studiem odborné literatury zabývající se vývojem a vhodnými postupy pro vývoj responzivního webu. Praktická část bakalářské práce bude zaměřena na vytvoření responzivní webové šablony na základě syntézy teoretických poznatků z teoretické části práce.

3. Historie a vývoj jazyka HTML

Jazyk HTML je zkratkou z anglického HyperText Markup Language. Tento značkovací jazyk se používá pro tvorbu webových stránek, které jsou propojené hypertextovými odkazy.

World Wide Web započal svůj život v roce 1989 ve Švýcarském CERNu. Tom Berners-Lee pracoval ve výpočetní části střediska, kde přišel s konceptem sdílení dat a informací mezi jednotlivými organizacemi tak, aby si výzkumníci mohli jednoduše poskytovat a sdílet své výsledky mezi sebou. Hlavní myšlenkou nebylo samotné sdílení souborů, ale textových informací, které by byly navzájem propojené pomocí hypertextových odkazů. [17]

HTML založené Timem Berners-Leem bylo silně závislé na jazyce SGML, který byl používán pro strukturování obsahu pomocí paragrafů, nadpisů, číslovaných a nečíslovaných seznamů atd. Tento jazyk však neuměl vytvářet hypertextové odkazy, pomocí kterých je možné procházet z jednoho dokumentu na dokument druhý. [41]

3.1. HTTP protokol

První verze protokolu (HTTP 0.9) byla navržena Tomem Bernersem-Leem a jeho funkce spočívá ve výměně hypertextových dokumentů ve formátu HTML. Server přijme požadavek a následně v závislosti na tomto požadavku zašle zpět webovou stránku. Dotaz ani odpověď ale neobsahovaly hlavičku ani jiné meta informace.

```
$> telnet google.com 80
Connected to 74.125.xxx.xxx
GET /about/
(hypertext response)
(connection closed)
```

Obrázek: 1 - HTTP protokol 0.9 [28]

V květnu roku 1996 pracovní skupina HTTP publikovala dokument pro implementaci HTTP 1.0, která přidávala k požadavku i k odpovědi hlavičku.

```
$> telnet website.org 80

Connected to xxx.xxx.xxx.xxx

GET /rfc/rfc1945.txt HTTP/1.0 1
User-Agent: CERN-LineMode/2.15 libwww/2.17b3
Accept: */*

HTTP/1.0 200 OK 2
Content-Type: text/plain
Content-Length: 137582
Expires: Thu, 01 Dec 1997 16:00:00 GMT
Last-Modified: Wed, 1 May 1996 12:45:26 GMT
Server: Apache 0.84

(plain-text response)
(connection closed)
```

Obrázek: 2 - HTTP protokol 1.0 [28]

Necelých sedm měsíců po uvedení standardu HTTP 1.0 vznikla také verze HTTP 1.1, která opravovala chyby předešlé verze protokolu a zároveň se z ní stala verze, která je známá dnes.

HTTP 1.1 řešila následující nejasnosti:

- výkonnost
- propojení
- převod blokového kódování
- byte-range
- kešování (Cache)
- přenosové kódování

[11; 28]

3.2. Doménový systém pojmenování

Domain Name System (DNS) je hierarchickým systémem doménových jmen. Úkolem serveru a stejně zvaného protokolu je převod doménových jmen na IP adresy uzlů sítě.

Systém DNS byl vyvinut v roce 1983 Paulem Mockapetrisem a v roce 1987 byl protokol aktualizován do dnešní podoby. [7]

Tento systém pojmenování usnadňuje uživatelům používání internetu tím, že není potřeba znát jednotlivé IP adresy, ale pouze jejich slovní ekvivalenty.

Doménová jména byla rozdělena do jednotlivých úrovní, které jsou odděleny tečkou. Úrovně jsou číslovány pozpátku, kde doména první úrovně je zároveň doménou nejvyšší úrovně a bývá nejčastěji zkratkou daného státu (např. .cz, .de,...). [6]

3.3. První grafický webový prohlížeč

Během vývoje jazyka HTML došlo na univerzitě v Illinois k vývoji prvního grafického internetového prohlížeče, který byl nazván Mosaic (1994). Později byl portován mimo Unix a bylo jej možné používat také na platformě Apple Macintosh a počítačích s Microsoft Windows, a to zcela bezplatně.

Na základě jeho zdrojového kódu byly vyvinuty dnes známé webové prohlížeče Internet Explorer a Netscape Navigator.

Před vývojem grafických webových prohlížečů došlo k vývoji textových, kde mezi nejznámější patří např. Lynx. [12]

3.4. World Wide Web konsorcium a jazyk HTML

V roce 1994, pět let po vytvoření první verze jazyka HTML, zakladatel jazyka HTML Tom Berners-Lee založil konsorcium, které mělo za úkol sjednotit různé verze jazyka HTML od různých výrobců a dále se starat o další rozvoj jazyka a samotného webu. [39]

V červenci roku 1994 došlo k definici druhé verze jazyka HTML, která zachycovala stav jazyka HTML k tomuto datu, a přidala specifikaci pro interaktivní formuláře a podporu grafiky.

V lednu roku 1997 byla specifikována verze HTML 3.2 a zachycovala jazyk k tomuto datu. K definici jazyka byly přidány nově tabulky, zarovnání textu a prvky pro ovlivňování vzhledu.

Verze HTML 4 a 4.01 byly vydány v roce 1997 a 1999 a opravily chyby z předešlých verzí. Zároveň byly přidány nové prvky pro tvorbu tabulek, formulářů a podporu pro rámy (frames), což představovalo rozdělení okna prohlížeče na několik částí a zobrazování více HTML stránek naráz. Ve verzi HTML 4 bylo snahou dosáhnout původních účelů jazyka a to přidáním konkrétního sémantického významu každému prvku.

V roce 2000 došlo ke specifikaci jazyka XHTML 1.0, který měl plně nahradit HTML 4.01, jehož vývoj byl do té doby pozastaven. Jazyk XHTML se dělil na tři části a to na XHTML Strict, Transitional a Frameset. Rozdíl mezi jazykem HTML a XHTML byl následující:

- každý tag je párový a je potřeba jej ukončit,
- tagy a atributy je nutné psát malými písmeny,
- veškeré hodnoty atributů musí být ohraničeny uvozovkami,
- dokument musí začínat XML deklarací

Aktuální verzí jazyka je HTML 5. Její hlavní změnou je ukončení závislosti HTML na SGML. Opravuje spoustu předešlých chyb, vyřazuje zastaralé tagy a přidává řadu nových. [17; 41]

Přehled verzí jazyka HTML a roku, kdy byla verze vydána:

HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML 5	2014

Tabulka 1 - Přehled verzí HTML a roku jejich vydání (autor)

3.5. Kaskádové styly

V roce 1995 došlo k zasedání konsorcia W3C z důvodu jednání nad kaskádovými styly. Název kaskádové vychází s faktu, že jednotlivé styly jsou psány do kaskády pod sebe a mohou se překrývat, čímž zvyšují svou efektivnost.

Smyslem kaskádových stylů je oddělit vzhled dokumentu od jeho obsahu a struktury. Původně už samotné HTML obsahovalo několik možností jak upravovat vzhled webové stránky, ale samotné použití nemělo tolik možností, jak vzhled upravit. [37] [9]

4. HTML sémantika a kaskádové styly

HTML sémantika popisuje pravidla struktury HTML dokumentu a příkládá význam jednotlivým HTML tagům.

4.1. HTML 5

Příchodem HTML 5 v roce 2014 došlo k několika změnám v sémantickém významu dosavadních HTML značek. Dále došlo k vyřazení zastaralých značek a přidání několika nových.

Význam HTML značek v minulosti znamenal také různá vizuální zobrazení daného prvku. Příchod kaskádových stylů změnil způsob a možnost zobrazení těchto prvků.

Na sémantický význam jednotlivých značek nahlíží automatické nástroje, které posuzují význam a strukturují nebo jinak nakládají s obsahem webových stránek. Jako příklad takového nástroje lze uvést internetové vyhledávače (Google, Seznam a podobně), které procházejí HTML kód webových stránek. Svým uživatelům následně zobrazují ve vyhledávání výsledky, které obsahují titulek stránky, hlavní nadpis (H1) a další obsah webových stránek. [4; 9]

Vývoj webových stránek byl v minulosti pomalejší než je tomu dnes a webové prohlížeče obsahovaly řadu chyb, kterým se tehdejší vývojáři přizpůsobovali. S příchodem tehdy pokročilého Internet Exploreru 5, který podporoval standardy HTML o mnoho lépe než jeho konkurence, došlo k nesprávnému zobrazování starších webových stránek. Tyto stránky předpokládaly, že nová verze prohlížeče bude obsahovat stejné chyby, jako ty předešlé.

Řešení bylo v přiloženém typu dokumentu, který se začal přidávat do kódu webových stránek. Webový prohlížeč při načítání webové stránky zkontroloval typ dokumentu a na jeho základě zobrazil stránky podle nových standardů. Starší webové stránky zpravidla typ dokumentu neobsahovaly a v tomto případě je webový prohlížeč interpretoval podle standardů starých.

Myšlenka s typem dokumentu se rozšířila a webové prohlížeče začaly podporovat dva modely zobrazení pro webové stránky. Prvním modelem byl tzv. „Quirks mode“ a druhý „Standardní režim“. [30; 31]

V HTML5 vypadá zápis typu dokumentu následovně:

```
<!DOCTYPE html>
```

Obrázek: 3 - Zápis Doctype v HTML5 (autor)

a říká prohlížeči, aby zobrazil webovou stránku ve standardním režimu. [41]

Struktura webové stránky připomíná strom. Prvky, které jsou součástí tohoto stromu, se nazývají „rodiče“, „děti“ nebo „sourozenci“.

Hlavní prvek se nazývá „kořenový prvek“ a obaluje celý obsah webové stránky. Vypadá následovně:

```
<html>
  ...
  ...
</html>
```

Obrázek: 4 - Kořenový prvek HTML (autor)

Prvním potomkem kořenového prvku `<html>` je hlavička dokumentu. Hlavička dokumentu obsahuje meta informace o stránce, cestu ke kaskádovým stylům nebo přímo vložený (inline) kus JavaScriptového kódu. [30]

Ukázka hlavičky dokumentu vypadá následovně:

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>My Weblog</title>
  <link rel="stylesheet" type="text/css" href="style-original.css" />
  <link rel="shortcut icon" href="/favicon.ico" />
</head>
```

Obrázek: 5 - Hlavička HTML (autor)

Meta značky umístěné v hlavičce dokumentu obecně slouží k popisu daného dokumentu (stránky). Mezi takové meta značky patří:

- description
- keywords
- robots
- content-language
- charset
- author

Meta značky, jako jsou „description“ nebo „keywords“ popisují, co je obsahem webové stránky. Další značky určují kódování webové stránky („charset“), autora webových stránek („author“) nebo nám upravují viditelnost stránek pro vyhledávače („robots“). [30]

S příchodem nové verze jazyka HTML došlo k odstranění zastaralých značek a k příchodu několika nových, mezi které patří:

- **<section>** - Obecný úsek dokumentu nebo aplikace, který typicky obsahuje nadpis.
- **<nav>** - Část stránky obsahující odkazy na další stránky, typicky hlavní navigaci stránek.
- **<article>** - Samostatný celek stránky v dokumentu, který je určen k opakovanému použití.
- **<header>** - Představuje skupinu úvodních nebo navigačních prvků .
- **<footer>** - Představuje patičku pro nejbližšího předka nebo pro celou stránku.
- **<aside>** - Představuje část stránky, která se netýká hlavní části obsahu.

[9; 30]

Nadpisy webových stránek určují strukturu celého dokumentu. HTML 5 dává možnost použít nadpis první až šesté úrovně (<h1> - <h6>), kde nadpis první úrovně určuje název uzlu dokumentu. [30]

Ukázka struktury webové stránky:

```
Název webové stránky (H1)
|
+-- Podnadpis webové stránky (H2)
|
+-- Další podnadpis webové stránky (H2)
    |
    +-- Podnadpis podnadpisu webové stránky (H3)
```

Obrázek: 6 - Struktura nadpisů v HTML [29]

Do HTML 5 byla nově přidána možnost použít tag <hgroup> a vytvořit tak například slogan doplňující název webové stránky:

```

<header>
  <hgroup>
    <h1>Název dokumentu</h1>
    <h2>Slogan</h2>
  </hgroup>
  ...
</header>

```

Obrázek: 7 - Využití <hgroup> ve struktuře nadpisů [29]

Tento nový zápis dvou navazujících nadpisů vytváří jeden uzel dokumentu, na rozdíl od předešlého sémantického významu. [30]

```

Název webové stránky (H1 + slogan H2)
|
+-- Podnadpis webové stránky (H2)
|
+-- Další podnadpis webové stránky (H2)
    |
    +-- Podnadpis podnadpisu webové stránky (H3)

```

Obrázek: 8 - Struktura nadpisů s využitím <hgroup> [29]

HTML 5 má oproti předešlé verzi použití více nadpisů H1 v jediném dokumentu. Umožňuje to nová značka <article>, která definuje samostatný uzel dokumentu. Každý uzel dokumentu pak může obsahovat vlastní nadpis první úrovně. [4; 30; 38]

Párový tag <nav> slouží pro označení hlavní navigace. Označení navigace má hlavně význam pro automatické roboty a hendikepované uživatele webu, kteří tak mohou snáze procházet web, přeskakovat jeho části a dostat se rychleji například k navigaci. [30]

```

<nav>
  <ul>
    <li><a href="">...</a></li>
    <li><a href="">...</a></li>
    <li><a href="">...</a></li>
    ...
  </ul>
</nav>

```

Obrázek: 9 - Ukázka použití navigace v HTML5 [29]

Ani dnes není zaručeno, že každý prohlížeč a jeho verze budou podporovat celý standard HTML 5. Pro tento případ vznikla JavaScriptová knihovna nazvaná Modernizr.js a je vhodné ji vkládat do hlavičky dokumentu (<head>).

Knihovna je sadou logických podmínek pro každou detekovanou vlastnost, která je nalezena v DOM dokumentu. Pokud je vlastnost podporována, je vrácena pravda, v opačném případě nepravda. Na tomto základě je možné přiřazovat prvkům třídy a zobrazovat je rozdílně pro zařízení nepodporující nový standard. [8; 9; 30]

4.2. CSS3

Posledním standardem pro kaskádové styly je třetí verze tohoto standardu, která mimo jiné přišla s animacemi, zakulacenými rohy, stíny, rotacemi, průhledností nebo RGBA modelem barev.

CSS3 je také na rozdíl od druhé verze standardu dělena do modulů, kde každý modul může přidávat nové možnosti a rozšiřovat tak funkce definované v CSS 2. [40]

Pro jednoduché animace elementů, je v CSS3 možné použít tzv. transition efekt, který spočívá ve stanovení doby trvání animace a specifikaci vlastností elementu, které se mají změnit.

I když CSS transformace nejsou plnohodnotným animačním nástrojem, je možné pomocí nich dosáhnout spousty animací, jako jsou animované hover stavy tlačítek, zvýrazňování rámečků u vstupních polí nebo animované přechody. [24; 49]

Jednoduchá CSS transformace má následující tvar:

```
transition:
  (_hlidane_vlastnosti_)
  (_trvani_animace_)
  (_funkce_prubehu_)
  (_zpozdeni_)
  (, _dalsi_transition_);
```

Obrázek: 10 - Animace v CSS3 [49]

Před příchodem CSS3 bylo jediným možným způsobem, jak zakulatit rohy obsahu nebo vytvořit kulatou fotografii pouze pomocí obrázku. Jednalo se o metodu poměrně náročnou, neboť bylo potřeba vytvořit obrázky v požadovaném rozměru, s požadovaným zakulacením a barvou, ale také definovat HTML značky, které reprezentovaly zakulacenou část.

CSS3 umožňuje zakulacovat rohy všech prvků jednoduše pomocí vlastnosti **border-radius**. Tato vlastnost nese parametr velikosti zakulacení, případně je možné zakulatit jen ty hrany, které chceme (`border-top-left-radius`, `border-bottom-right-radius`, atd.).

Pomocí zakulacování rohů je možné také docílit kulatých fotografií, kde přidáním této vlastnosti k obrázku s dostatečnou velikostí zakulacení v parametru, se z hranaté fotografie stane kulatá, bez potřeby vytvářet a umístit na fotografii poloprůhledný obrázek. [24; 34]

Pomocí kaskádových stylů lze velmi jednoduše přidávat stín k blokovým prvkům a textu pomocí vlastností **text-shadow** a **box-shadow**. Tyto dvě vlastnosti obsahují parametry pro:

- horizontální pozici,
- vertikální pozici,
- rádius rozostření,
- barvu.

[46; 47]

CSS3 rotace nebo také 3D transformace umožňují pomocí kaskádových stylů otáčet HTML elementy o určitý počet stupňů a docílit tak zajímavých vizuálních efektů. Animace se provádí pomocí vlastnosti **transform**, ke které se přidává jedna ze tří hodnot (`rotateX`, `rotateY` nebo `rotateZ`) společně s hodnotou ve stupních, např.: [24; 48]

```
transform: rotateX(150deg);
```

Obrázek: 11 - Rotece v CSS3 [48]

Nově také došlo na rozšíření barevného modelu, kde se ke standardnímu RGB modelu přidal stupeň Alpha, který udává hodnotu průhlednosti v procentech (1 = 100 %, 0 = 0 %). Tato metoda šetří čas a datový přenos. U předešlých možností bylo možné dosáhnout poloprůhledného pozadí prvků pouze pomocí poloprůhledného PNG obrázku, který byl opakován v dané oblasti. [22; 24]

```
color: rgba([red], [green], [blue], [alpha]);
```

Obrázek: 12 - Použití RGBA modelu [22]

4.3. CSS preprocessory

Cesta k zjednodušení zápisu a znovu použitelnosti kódu se ukázala v kompilování zápisu a tak roku 2007 vznikl SASS. Jeho úkolem bylo zjednodušit a rozšířit samotné CSS. Tento zápis nevyhovoval všem a v roce 2009 vznikl dnes hodně rozšířený LESS, jehož zápis celý vychází ze zápisu CSS.

Překladače a preprocessory umožňují využívat proměnné, výpočetní výrazy, objektové „mixins“, vnořování pravidel a další funkce. Díky užívání preprocesoru je možné zkracovat zápis CSS, zrychlit práci a následnou kompilaci ještě snížit jeho datovou velikost. [20]

LESS byl vyvinut pro zjednodušení zápisu kaskádových stylů. Zavádí užitečné vlastnosti jako jsou:

- **Proměnné** – Proměnné se zapisují pomocí znaku zavináč a názvu proměnné např.: @black. Takovouto proměnnou je poté možné znovu použít, čímž odpadá nutnost pamatovat si hodnotu této proměnné.
- **Rozšíření výrazu** – Pomocí rozšiřování výrazu je možné vzít vlastnost předešlého prvku a doplnit ji o vlastnosti nové. Rozšíření se provádí pomocí **:extend()**, kde v kulatých závorkách je obsažen selektor rozšiřujícího elementu.
- **MIXINS** – Zavádí funkce do kaskádových stylů. Jejich zápis je téměř totožný se zápisem klasického CSS. Pomocí **.navez-funkce([parametr]){[vlastnosti]}**, lze vytvořit znovupoužitelnou sadu vlastností.

[20]

Díky mix-in lze v LESS dosáhnout vytváření funkcí, které můžeme používat na více místech v dokumentu a vyhnout se tak duplikování kódu.

Taková funkce v LESS lze zapsat s parametrem a bez parametru a v závislosti na tom ji také volat.

Funkce bez parametru má následující zápis (tučně):

```

.a, #b {
  color: red;
}
.mixin-class {
  .a();
}
.mixin-id {
  #b();
}

```

Obrázek: 13 - Použití funkcí v LESS [20]

Lze ji volat pomocí názvu a kulatých závorek, jak je ukázáno u třídy `.mixin-class` a `.mixin-id`. Tento zápis je vhodný pro jednoduché recyklování kusu CSS zápisu.

V případě, že chceme přidat funkci parametr, je zápis o něco jiný a vypadá následovně:

```

.border-radius(@radius) {
  -webkit-border-radius: @radius;
  -moz-border-radius: @radius;
  border-radius: @radius;
}

```

Obrázek: 14 - Funkce `border-radius()` [34]

Pro jeho ukázkou je zvolena vlastnost **border-radius**, která má svůj zápis rozdílný pro každou skupinu webových prohlížečů. Pomocí mix-inu s parametrem je možné docílit jednotného zápisu, kde se zbytek vyřeší samotným doplněním.

Funkce je doplněna parametrem nesoucím hodnotu zaoblení, která je doplněna do každé vlastnosti samostatně. [21]

Výsledný soubor **.less** je potřeba zkompileovat a získat z něho známý zápis CSS. Nad tímto souborem je třeba použít sadu funkcí potřebnou pro kompilaci.

Pro tyto potřeby lze použít příkazovou řádku a nad souborem spustit sadu JavaScriptových příkazů nebo použít software třetí strany. Mezi nejlepší aplikace patří CodeKIT, vyvíjená pro platformu OS X.

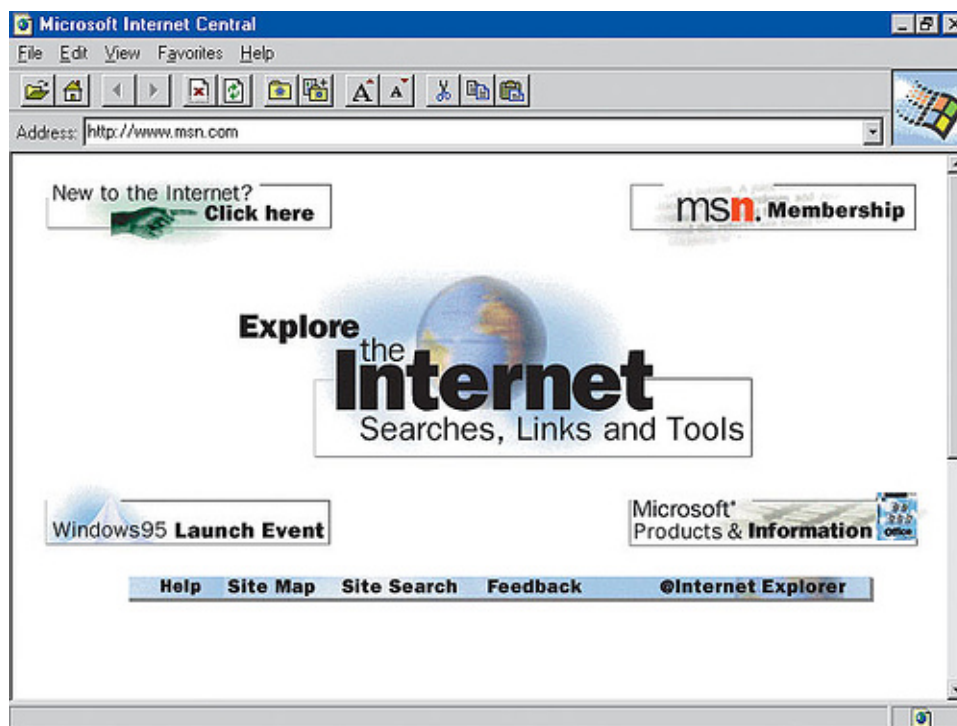
Aplikace hlídá nejenom správnost zápisu, ale také zobrazuje místo chyby, minifikuje výsledný CSS soubor, HTML, JavaScriptový kód a redukuje velikost vložených obrázků. [14]

5. Webové prohlížeče

Webové prohlížeče fungují jako brána do internetu a umožňují zobrazovat webové stránky na různých typech zařízení. Mezi nejznámější a nejpoužívanější internetové prohlížeče patří Internet Explorer, Mozilla Firefox, Google Chrome, Opera a Safari.

5.1. Internet Explorer

První verze prohlížeče Internet Explorer pochází z roku 1995 a byla odvozena z prohlížeče Spyglass Mosaic vyvíjeného společností NCSA, později přejmenovaného na Netscape Navigator.



Obrázek: 15 - Internet Explorer 1.0 [35]

Druhá verze prohlížeče byla dostupná v dubnu roku 1996 pro platformu Windows 3.1 a Macintosh.

V srpnu téhož roku byla dostupná již třetí verze prohlížeče, která se stala součástí Windows 95. Třetí verze prohlížeče obsahovala nově Mail, News 1.0 a Adresář.

Čtvrtá verze prohlížeče přišla v září 1997 a byla integrována hlouběji do systému Windows a podporovala DHTML, Outlook Express 4.0 a RSS.

Březen roku 1999 se zapsal příchodem páté verze prohlížeče, který podporoval CSS 1 a 2, automatické doplňování, kódování jazyků a přidání oblíbených položek. Tato verze byla součástí MS Windows 98.

Internet Explorer 6 byl vypuštěn roku 2001 a obsahoval tzv. „Quirks mode“. Mód simuloval verzi IE 5.5. a byl součástí Windows XP. Prohlížeč byl kritizován za jeho značné bezpečnostní chyby a nedostatečnou podporu nových standardů.

Po pěti letech přišla další verze prohlížeče s označením Internet Explorer 7 a byla součástí tehdy aktuálních Windows Vista. Sedmá verze prohlížeče vylepšila podporu standardů, a i když neprošla testem ACID 2 ani 3, dostala vylepšení bezpečnosti, phishing filtr, silnější šifrování a tlačítko pro smazání historie prohlížení.

Internet Explorer 8 byl uvolněn roku 2009 jako další vylepšení prohlížeče. IE 8 byl první prohlížeč ze série, který úspěšně prošel testem ACID 2. Zlepšil podporu pro RSS, CSS a došlo k přidání podpory pro technologii Ajax.

V roce 2011 vyšla 9 verze prohlížeče a od doby Internet Exploreru 2 nebyla součástí operačního systému Windows. Devátá verze prohlížeče podporovala několik vlastností z nových CSS 3 a vylepšovala výkon pro JavaScript. Implementovala podporu pro vektorovou grafiku (SVG), hardware akceleraci nebo podporu pro HTML5 video a audio. Devátá verze prohlížeče také úspěšně prošla ACID 3 testem, který slouží pro testování dodržování standardů DOM a JavaScriptu.

Internet Explorer 10 je dostupný od roku 2012 a byl základním prohlížečem ve Windows 8. Plně podporuje CSS3, HW akceleraci a HTML5.

Internet Explorer 11 je dnes nejnovější verzí prohlížeče, která byla uvedena roku 2013. Dočkala se redesignu nástrojů pro vývojáře a dostala podporu pro WebGL.

Další verzí v pořadí je Microsoft Edge, která uzavírá dosavadní číslování verzí prohlížeče. Stává se také základním prohlížečem současných chytrých telefonů a tabletů s operačním systémem Windows Mobile. [25; 27; 31]

5.2. Mozilla Firefox

Prohlížeč Mozilla Firefox započal svůj život v roce 1998 po zveřejnění zdrojového kódu prohlížeče Netscape Navigator. Okolo projektu se začala shromažďovat velká skupinka nadšenců, kteří měli jediný cíl, a to vytvářet zdarma software pro lepší prožitky z internetu.

Po několika letech Mozilla vydala verzi prohlížeče s označením Mozilla 1.0, jednalo se o první verzi prohlížeče a byla vydána v roce 2002. Bohužel ji v tomto roce nepoužívalo dost uživatelů, neboť 90 % z nich využívalo Internet Explorer, který byl součástí systému Windows. V roce 2003 byla založena společnost nesoucí název Mozilla Foundation. Nezisková organizace, která se starala o vývoj Firefoxu a Thunderbirdu.

Firefox 1.0 byl zveřejněn v roce 2004 a měl veliký úspěch. Během prvního roku si jej instalovalo více než 100 milionů uživatelů.

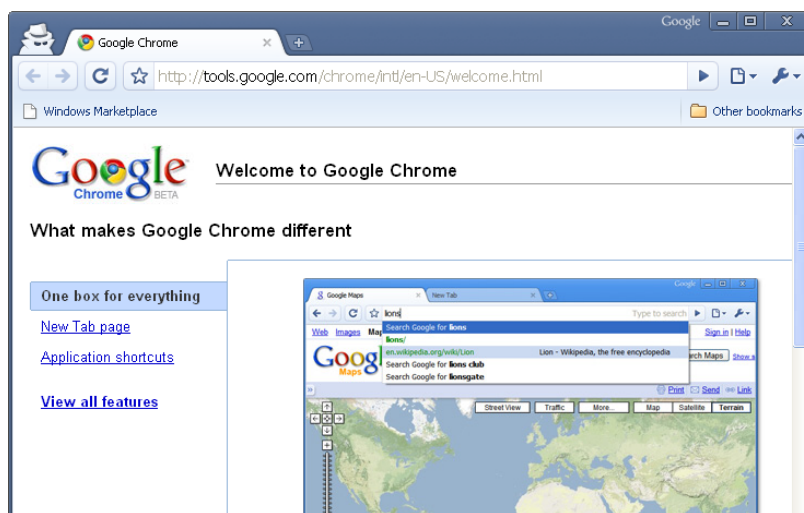
Na rozdíl od Internet Exploreru se od roku 2011 prováděl update prohlížeče každých 5-6 týdnů. Docházelo tak k rychlejšímu odstraňování chyb, příchodu nových funkcí a vylepšení.

Aktuální verze prohlížeče nese označení Mozilla Firefox 40. [25; 26; 31]

5.3. Google Chrome

Dnes velice populární prohlížeč Google Chrome byl založen na renderovacím jádru WebKit. Jeho nejnovější verze je založena na jádru Blink, který je jednodušší a výkonnější.

První stabilní verze prohlížeče byla vydána v roce 2008. Ještě téhož roku se prohlížeč Google Chrome dostal do první pětky prohlížečů a stál tak za Internet Explorerem, Firefoxem a prohlížečem Safari.



Obrázek: 16 - Google Chrome [3]

Od verze Google Chrome 3 jsou pro uživatele k dispozici vlastní motivy změny vzhledu prohlížeče.

Čtvrtá verze přišla s možností instalovat rozšíření.

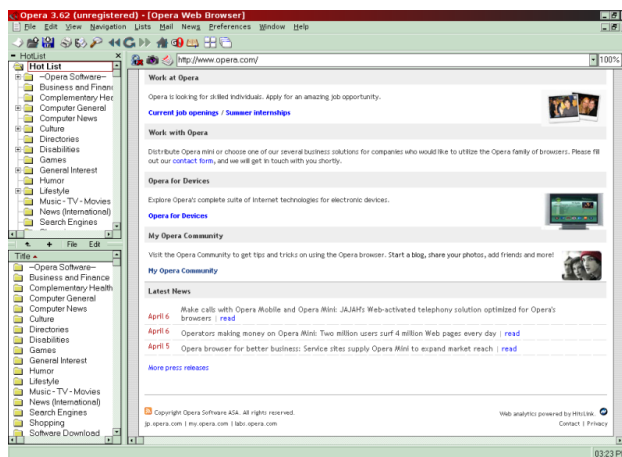
Webový prohlížeč Google Chrome je dostupný pro desktopové operační systémy, tablety a mobilní telefony.

Poslední aktuální verzi prohlížeče je verze Google Chrome 45. [25; 44]

5.4. Opera

Historie webového prohlížeče sahá do roku 1994, kde začala jako projekt nazvaný Telenor Norské telekomunikační společnosti. Jako samostatný se projekt stal roku 1995 a byl přejmenován na Opera Software.

Verze 2.0 byla demonstrována na třetí mezinárodní WWW konferenci v roce 1995. Třetí verze prohlížeče byla představena v roce 1997 a podporovala JavaScript a kaskádové styly.



Obrázek: 17 - Prohlížeč Opera 3.62 [43]

V roce 2013 Opera oznámila přechod na jádro WebKit a následně Blink. Jde o stejné jádro, které používá prohlížeč Google Chrome. Opera Software vyvíjí jádro společně se společností Google. [15; 45]

5.5. Safari

Safari je webovým prohlížečem distribuovaným společně s operačním systémem OS X a iOS. První stabilní verze prohlížeče byla vydána v roce 2003 a v roce 2007 i pro platformu Windows.

Webový prohlížeč Safari používá pro vykreslování jádro WebKit. Aktuální verzi prohlížeče Safari je jeho verze s označením Safari 8 vydaná současně s novou verzí OS X Yosemite. [25]

5.6. Podíl současných prohlížečů na trhu

V počátcích rozvoje internetu byl jednoznačným lídrem v používání Internet Explorer, který k lednu roku 2003 měl celých 84,6 % podílu na trhu.

2003			
Mozilla	Internet Explorer	Netscape	Opera
4,0 %	84,6 %	4,0 %	-

Tabulka 2 - Podíl vyhledávačů roce 2003 [42]

S příchodem prohlížeče Firefox v roce 2004 se začal poměr mezi prohlížeči měnit a své příznivce na svou stranu přitahovala Mozilla Firefox.

2005				
Mozilla	Internet Explorer	Firefox	Netscape	Opera
3,4 %	74,5 %	16,6 %	1,1 %	1,9 %

Tabulka 3 - Podíl vyhledávačů v roce 2005 [42]

V roce 2008 začal trh ovlivňovat prohlížeč Google Chrome společně s rostoucí popularitou Firefoxu, který po 10 letech ovládal téměř polovinu trhu.

2008				
Google Chrome	Internet Explorer	Firefox	Safari	Opera
3,6 %	46,0 %	44,4 %	2,7	2,4

Tabulka 4 - Podíl vyhledávačů v roce 2008 [42]

Jednalo se o první roky, kdy se Internet Explorer dostal pod 50% hranici. Za dva roky poté vzrostla popularita prohlížeče Google Chrome a dostala se v prosinci roku 2010 na 22,4 %.

2010				
Google Chrome	Internet Explorer	Firefox	Safari	Opera
22,4 %	27,5 %	43,5 %	3,8 %	2,2 %

Tabulka 5 - Podíl vyhledávačů v roce 2010 [42]

O další dva roky později se roku 2012 stal Google Chrome nejpoužívanějším webovým prohlížečem, zatímco Internet Explorer klesl pod hranici 15 %.

2012				
Google Chrome	Internet Explorer	Firefox	Safari	Opera
46,9 %	14,7 %	31,1 %	4,2 %	2,1 %

Tabulka 6 - Podíl vyhledávačů v roce 2012 [42]

K srpnu roku 2015 je podíl prohlížečů následující.

2015				
Google Chrome	Internet Explorer	Firefox	Safari	Opera
64 %	6,6 %	21,2 %	4,5 %	2,2 %

Tabulka 7 - Podíl vyhledávačů v roce 2015 [42]

Podíl prohlížeče Internet Explorer se za posledních 12 let dostal z 84 % na necelých 7 % na rozdíl od Chromu, který za posledních 7 let stoupá a používá jej nadpoloviční většina uživatelů internetu. [42]

6. Responzivní webdesign

S příchodem mobilních telefonů a ostatních mobilních zařízení se výrazně změnil přístup k vývoji webových stránek. Nově je potřeba webová stránka zobrazovat různým způsobem na různých mobilních zařízeních a poskytnout tak uživatelům maximální komfort. S tím souvisejí také techniky pro snížení počtu požadavků odeslaných na server a datová velikost stránek, která kvůli datovým limitům a rychlosti mobilního připojení k internetu hraje významnou roli.

Responzivní webové stránky se vytvářejí nejčastěji pomocí tzv. „Media Queries“, která jsou součástí specifikace CSS3, kde je možné rozeznat vlastnosti zařízení. Pomocí této techniky je možné docílit různého zobrazení stránek pro různá zařízení. Stále je ale potřeba myslet na různé zobrazení webových stránek v různých webových prohlížečích, neboť zobrazení jednotlivých prvků v prohlížečích není stále jednotné.

6.1. Historie

Vývojáři chtěli, aby webové stránky byly zobrazeny co největšímu počtu uživatelů. Pomocí fixní šířky webové stránky, která byla mezi 800px a 1024px, bylo docíleno jisté optimalizace stránek pro největší možný počet rozlišení.

Po příchodu prvních chytrých telefonů se začalo přemýšlet o různém zobrazení webů pro tyto zařízení. Pomocí JavaScriptových knihoven a CSS2 se začaly vytvářet první mobilní webové stránky, které byly uživatelům zobrazovány při přístupu z mobilního telefonu. Tyto mobilní webové stránky byly optimalizovány především na co nejnižší datovou velikost, neboť standardní velikost FUP mobilních operátorů byla v té době v řádu desítek megabajtů dat přenesených za měsíc.

Pojem responzivní webdesign specifikoval až Ethan Marcotte v novinách A List Apart. V těchto novinách zveřejnil, jak výrazně se mění prostředí internetu a přibývá mobilních uživatelů. Domníval se, že jedinou cestou jak se dostat dopředu, je využití fixního nebo fluidního layoutu stránek.

Responzivní webdesign, jak ho známe dnes, je kombinací fluidního layoutu, který používá relativní jednotky, flexibilních obrázků a mediálních knihoven („Media Queries“), které podle zařízení aplikují na stránku různé styly. [24; 32]

6.2. Metody návrhu webových stránek

Webové stránky lze navrhovat ze dvou různých pohledů. Ten první, a dnes ještě stále nejpoužívanější, je tzv. „desktop first“ a jeho návrh je soustředěn nejprve na desktopová zařízení a až následně je návrh transformován na mobilní zařízení.

Druhá metoda je nazývána „mobile first“ a zaměřuje se nejprve na návrh stránek pro mobilní zařízení a až následně pro zařízení pevná.

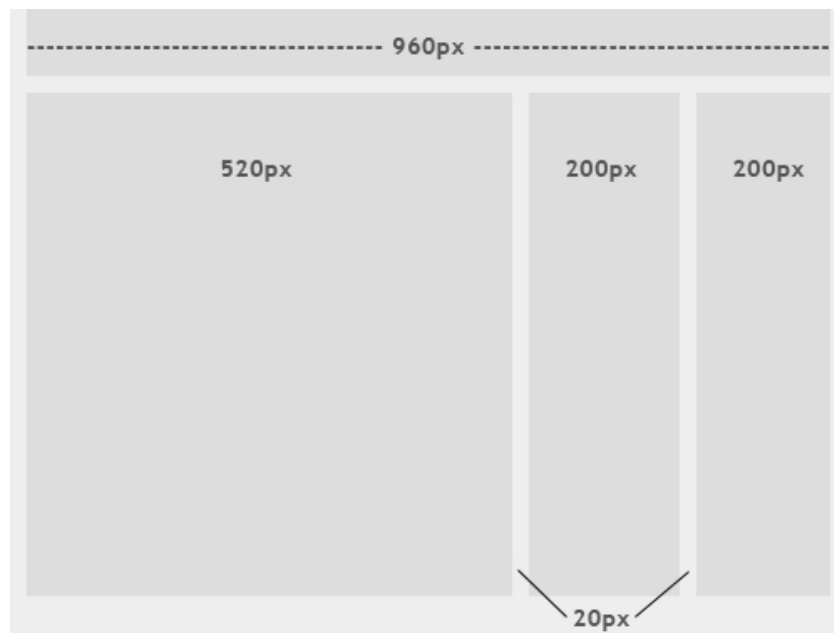
„Mobile first“ metoda je úspornější na samotný návrh a také na množství zdrojového kódu, který je nezbytný pro její pozdější implementaci. Z tohoto důvodu je tato metoda považována za efektivnější a tím pádem také lepší. Na druhou stranu je tato metoda náročnější na vytvoření prvotního návrhu, neboť je potřeba celý návrh detailněji promyslet.

O metodě „mobil first“ se poprvé zmínil Luke Wroblewski, a to necelé dva roky po uvedení prvního Apple iPhone na trh. Šest let poté už mobilní telefony zabíraly více než 40 % trhu. [2; 36]

6.3. Grafický návrh

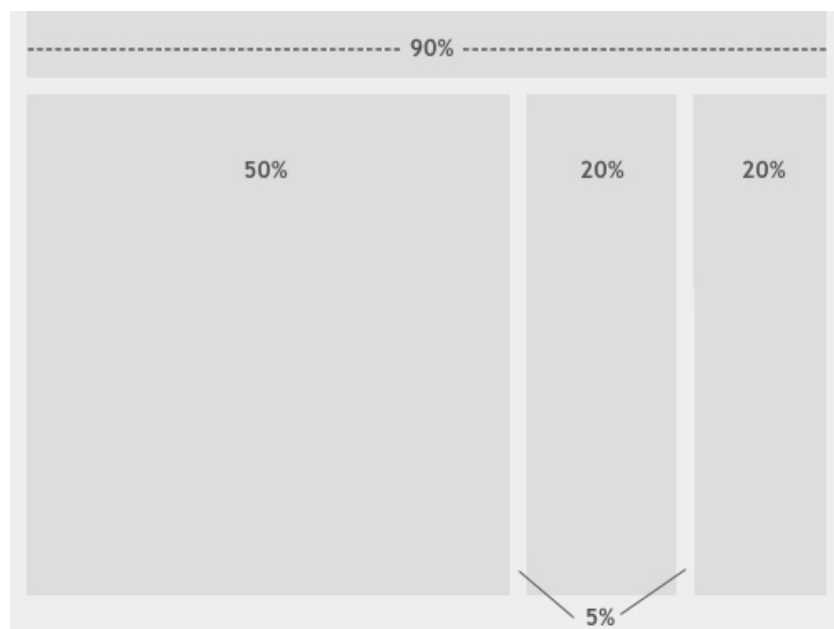
Webové stránky mohou být roztaženy na celou šířku webu nebo jako horizontální pruh uprostřed prohlížeče. Rozložení webu se dělí na fixní a fluidní.

Fixní layout je složen z obalovacího boxu, který má danou pevnou šířku a jeho obsah je uveden pevnou nebo procentuální hodnotou. U tohoto typu layoutu není důležité jaké rozlišení má zařízení, na kterém je webová stránka zobrazena (Obr. 1). [16]



Obrázek: 18 - Fixní layout [16]

Fluidní layout má veškeré komponenty s procentuální hodnotou. Díky tomu jsou prvky automaticky přizpůsobeny velikosti zařízení a jsou na všech zařízeních zobrazeny totožně. [16]



Obrázek: 19 - Fluidní layout [16]

ulehčující následné pozicování obrázků pomocí CSS. Jednotlivé obrázky jsou pozicovány na web určením X a Y souřadnice.

U responzivního webu je třeba často vytvářet hned dvě sady obrázků. Jedna sada je určena pro zařízení s nižší hustotou bodů a druhá sada pro zařízení s „retina“ displejem. Bez vytváření dvou sad obrázků by obrázky s vysokým rozlišením byly neostře a rozmazané.

Nevýhodou jsou různé datové nároky, neboť sada obrázků pro vysoká rozlišení je obvykle 2x větší než sada normální. [19; 24]

Scalable Vector Graphics je formátem pro vektorovou grafiku, jeho standard pochází z roku 2001.

Výhody obrázků ve formátu SVG jsou ve výrazné datové úspoře. Obsah je strojově čitelný a neztrácí vizuální kvalitu u zařízení s vysokokapacitními displeji.

Na druhou stranu se SVG obrázky používají pouze pro zobrazování jednoduchých obrázků, protože jsou výpočetně náročné. Ostatní obrázky je potřeba stále používat jako bitmapu. [9; 23]

Ikony pomocí font jsou známé už od první verze HTML a CSS. Jejich praktické využití se dostavilo až dnes, kdy jsou vhodné nástroje pro jejich snadnou implementaci do webové stránky.

Největší výhodou je „nekonečné“ zvětšování a neomezená změna barvy.

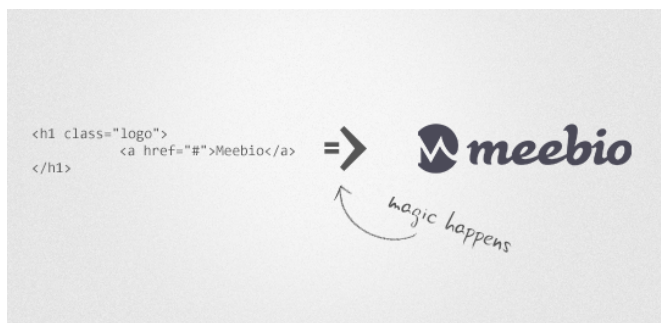
Do stránek lze tento ikonový font vkládat přes @Font-face, stejně jako fonty, které nejsou součástí operačních systémů. Implementace vypadá následovně:

```
@font-face {
  font-family: 'Entypo';
  src: url('/fonts/Entypo/entypo.eot');
  src: url('/fonts/Entypo/entypo.eot?iefix') format('eot'),
       url('/fonts/Entypo/entypo.woff') format('woff'),
       url('/fonts/Entypo/entypo.ttf') format('truetype'),
       url('/fonts/Entypo/entypo.svg#entypo') format('svg');
  font-weight: normal;
  font-style: normal;
}
```

Obrázek: 21 - Použití @Font-face metody [18]

Problém s fonty může nastat v případě, že prohlížeč nebude podporovat techniku @Font-face. V takovém případě bude webová stránka plná nesrozumitelných znaků.

Tento problém se řeší přes tzv. „ligatury“. Jedná se o spojení několika znaků, pod kterými je následně ikona dostupná. [18; 24]

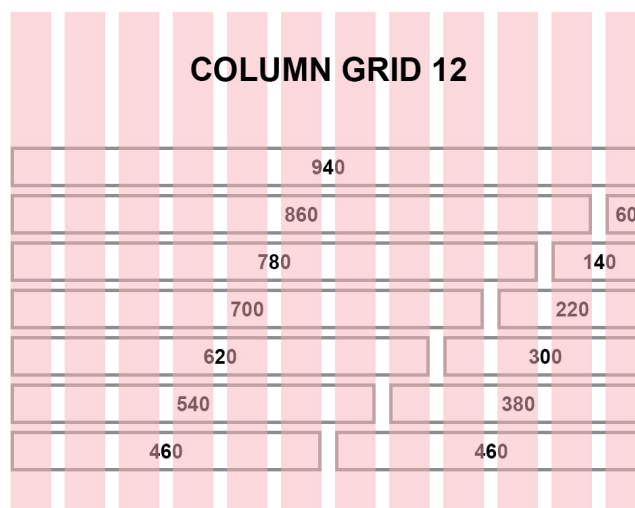


Obrázek: 22 – Použití ligatury [18]

6.5. Responzivní grid systémy

Pro urychlení vývoje webových stránek lze využít některý z grid systémů. Grid systém je ve své podstatě hrubou strukturou návrhu webu a slouží jako vodítko.

Grid systém je obvykle rozdělen do dvanácti sloupců s deseti mezerami. Velikost sloupce může být například 60px a velikost mezery 20px. Dohromady to dělá 960px na šířku. [1; 5]



Obrázek: 23 - Grid systémy (autor)

6.6. Jednotky

Webové stránky jsou sestaveny ze samostatných bloků, které mají určenou výšku a šířku v určených jednotkách. Jednotky, které lze použít jsou buď relativní (např. procenta) nebo absolutní (např. pixely).

U responzivního webu je vhodné použít relativní jednotky pro layout, neboť rozlišení zařízení je velmi variabilní, a absolutní jednotky pro fonty. Tím je docíleno shodného zobrazení jak struktury, tak jednotlivých prvků webu. [13]

Mezi relativní jednotky patří:

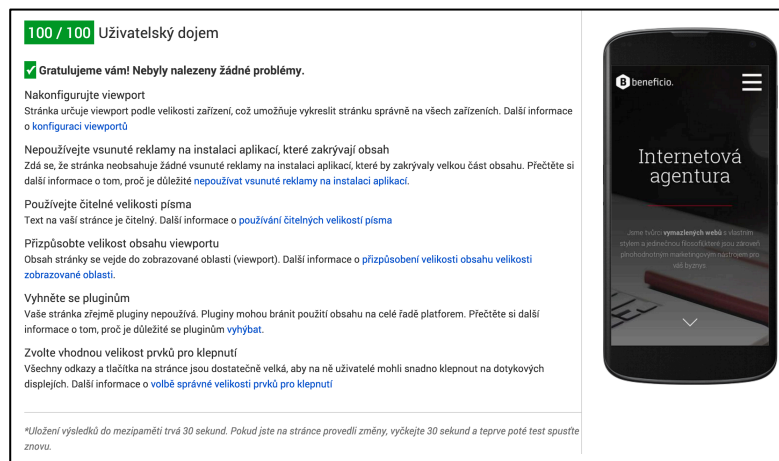
- **EM** – Jednotky relativní k velikosti fontu (odpovídá šířce písmene M).
- **REM** – Jednotky relativní k velikosti základního fontu (= Root EM).
- **EX** – Jednotky relativní k x-výšce aktuálního fontu (odpovídá výšce písmene x).
- **VW** – Jednotky relativní k 1 % šířky displaye.
- **VH** – Jednotky relativní k 1 % výšky displaye.

Absolutní jednotky jsou potom:

- **PX** – Velikost obrazového bodu.
- **MM** – 1 mm = 3,75 px.
- **CM** – 1 cm = 37,8 px.
- **PT** – 3 pt = 4 px.
- **PC** – 1 px = 16 px.

6.7. Google speed insights

Google, jako nynější světová jednička ve vyhledávání, uvolnil nástroj pro vývojáře nejen mobilních webů. Nástroj pro zvýšení jejich efektivnosti a prožitku uživatelů při jejich prohlížení.



Obrázek: 24 – Google pagespeed test (autor)

Zároveň se jedná o nástroj, který posuzuje samotnou optimalizaci stránek pro mobilní zařízení a radí vývojářům, jak stránky vylepšit, případně upravit.

Každá webová stránka, testovaná nástrojem Google Page Insights je považována za výkonnou, jestliže dosahuje alespoň 85 bodů hodnocení.

U stránek se hodnotí rychlost načtení a tedy: konfigurace serveru, HTML struktura stránek a využití externích zdrojů jako je JavaScript, CSS a obrázky. [10]

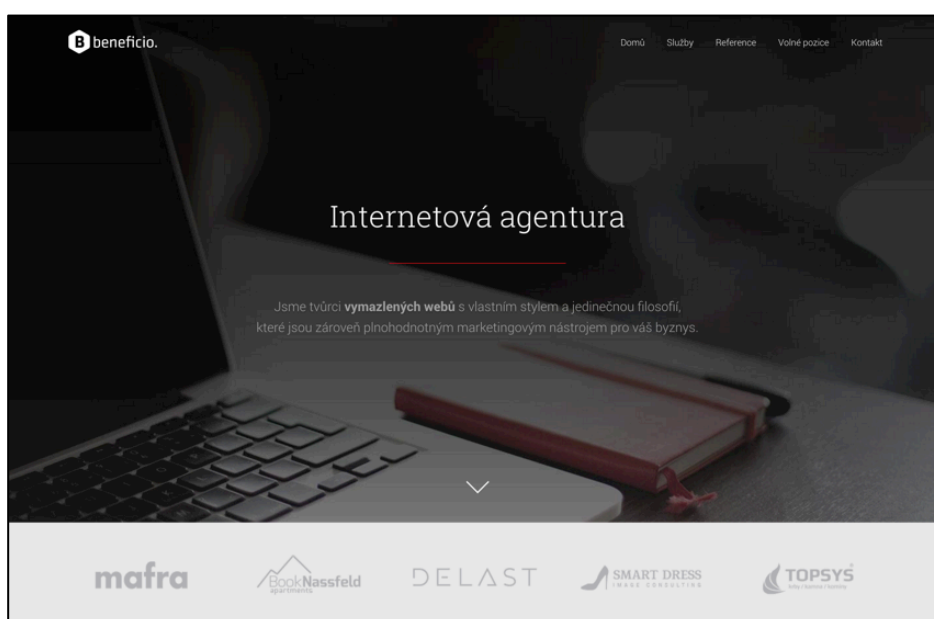
7. Praktická část bakalářské práce

Pro účely bakalářské práce byly vytvořeny nové webové stránky internetové agentury. Responzivní webové stránky byly vytvořeny pomocí metody využívající techniku media queries. Dále byly použity CSS sprite obrázky pro ikony a opakující se obrázky pro snížení počtu HTTP požadavků na server a odstranění prodlevy mezi načítáním obrázků pro HOVER stavy elementů. Kaskádové styly byly generovány pomocí LESS preprocesoru pomocí aplikace CodeKit a byly následně spojeny do jednoho souboru a minifikovány. Tím bylo docíleno dalšího snížení HTTP požadavků a tím zrychlení načítání stránek. Pro výkonnost stránek byl použit nástroj Google Page Insights, pomocí kterého byly stránky optimalizovány pro maximální uživatelský komfort a rychlost.

7.1. Design

Design webových stránek byl přizpůsoben snadné implementaci pro mobilní zařízení. Vše je tedy zobrazeno v jednosloupcovém fixním layoutu. Z důvodu zvýšení použitelnosti stránek na co největší počet zařízení a zároveň zajištění stejného vykreslení prvků na různých monitorech, byla vybrána pro fixní layout maximální šířka 1280px.

Šíře stránek je přizpůsobena šířce obrazovky až po rozlišení 1024px. Pro nižší rozlišení je stránka upravena na 100% šířky okna prohlížeče.



Obrázek: 25 - Ukázka desktopové verze stránek (autor)

U mobilní varianty stránek je navigace schována za tzv. „Burger“. Po kliknutí na burger dojde k vysunutí navigace stránek. Obsah stránek je soustředěn pod sebe tak, aby byla zachována jeho struktura.



Obrázek: 26 - Ukázka mobilní verze stránek (autor)

7.2. HTML část

Struktura HTML stránek vychází z fixního layoutu. Hlavička dokumentu obsahuje informace o typu dokumentu a použité verzi HTML.


```

<!doctype html>
<html class="no-js" lang="cs">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<title>Tvorba webových stránek a internetový marketing - Beneficio Media,
s.r.o.</title>
<meta name="description" content="Jsme tvůrci vymazlených webů s vlastním
stylem a jedinečnou filosofií, které jsou zároveň plnohodnotným marketingovým
nástrojem pro váš byznys.">
<meta name="keywords" content="internetová agentura, digitální agentura,
tvorba www, tvorba webových stránek, webdesign, online marketing, marketing">
<meta name="author" content="Beneficio Media, s.r.o.">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="robots" content="index, follow">
<link rel="shortcut icon" href="/favicon.ico">
<link rel="stylesheet" href="/css/style.css">
<!--[if lte IE 8]><script src="/js/respond.min.js"></script><![endif]-->
<script src="/js/modernizr.js"></script>
</head>

```

Obrázek: 27 - Doctype a hlavička projektu (autor)

Hlavička dále obsahuje název dokumentu, popis dané stránky a klíčová slova specifikující její obsah. Způsobu vykreslování stránky na mobilních zařízeních (viewport) je přidělena hodnota 1. Tato hodnota znamená, že stránky budou zobrazeny vždy v měřítku 1:1. Tím je docíleno správnému zobrazení a rozmístění prvků na mobilních zařízeních. Informace pro roboty (robots) informují vyhledávací služby o tom, že stránky mají zařadit do svého vyhledávání. Nakonec byly připojeny kaskádové styly a knihovna modernizr pro optimalizaci zobrazení stránek ve starších verzích prohlížečů.

```
<body>
  <div class="page">
    <header class="header">
      ...
    </header>
    <section class="content">
      ...
    </section>
    <footer class="footer">
      ...
    </footer>
  </div>
</body>
```

Obrázek: 28 - Struktura těla projektu (autor)

Stránky byly rozděleny na hlavičku, tělo dokumentu a patičku. Hlavička společně s patičkou byly společné pro všechny stránky a podstránky, zatímco tělo dokumentu bylo pro každou jednotlivou stránku jedinečné.

Hlavička dokumentu byla členěna na část obsahující logo a dvouúrovňovou navigaci stránek. Pro zjednodušenou práci s responzivností stránek byl vnitřek hlavičky ještě zabalen do vnitřního divu `<div class="in"></div>`, který určuje šířku obsahu stránky.

Metoda zabalení obsahu do nadřazeného prvku se velmi osvědčila. Tím, že je definována nadřazenému prvku pevná velikost a jeho potomkům procentuální velikost, lze jednoduše přizpůsobovat obsah různým rozlišením pouhou změnou velikosti jednoho nadřazeného prvku. Ve své podstatě se jedná o kombinaci fixního a fluidního layoutu.

```

<header class="header">
<div class="in">
<a href="/" class="logo" title="Beneficio Media, s.r.o. | Tvorba webových
stránek a internetový marketing"></a>
<nav>
<span class="mobileBurger"></span>
<ul>
<li><a href="/" title="Beneficio Media, s.r.o.">Domů</a></li>
<li class="submenu" id="sluzby"><a href="#" title="Tvorba webových stránek a
internetový marketing">Služby</a>
<ul>
<li><a href="/tvorba-webovych-stranek" title="Tvorba profesionálních webových
stránek a aplikací">Tvorba <strong>webových stránek</strong></a></li>
<li><a href="/internetovy-marketing" title="Přísun kvalitních návštěvníků a
budování silné značky díky online marketingu">Interentový
<strong>marketing</strong></a></li>
</ul>
</li>
<li><a href="/reference" title="Ukázka webových a marketingových
úspěchů">Reference</a></li>
<li><a href="/volne-pozice" title="Volné pracovní pozice">Volné
pozice</a></li>
<li><a href="/kontakt" title="Kontaktní informace">Kontakt</a></li>
</ul>
</nav>
<hgroup>
<h1>Internetová agentura</h1>
<h2>Jsme tvůrci <strong>vymazlených webů</strong> s vlastním stylem a
jedinečnou filosofií,<br>které jsou zároveň plnohodnotným marketingovým
nástrojem pro váš byznys.</h2>
</hgroup>
<div class="arrow"></div>
</div>
</header>

```

Obrázek: 29 - Hlavička těla projektu (autor)

Logo bylo vytvořeno pomocí běžného HTML tagu `` s popisem atributu ALT, namísto pouhého nahrazení textu obrázkem na pozadí. Nahrazení textu obrázkem je výhodné z pohledu SEO optimalizace. Má větší efekt na všech podstránkách opakovat obrázek, namísto překrývání stejného textu obrázkem loga.

Navigace stránek byla uzavřena HTML tagem `<nav>` a hlavní nadpis stránek byl uzavřen tagem `<hgroup>` kvůli správné struktuře uzlů nadpisů.

Patička stránek obsahuje informace o subjektu, copyright, mapu s adresou a také tlačítko pro rychlý přesun na vršek stránky.

```

<footer class="footer">
<div class="googleMap">
<section class="contactBox">
<header><h2><strong>Přijďte</strong> za námi</h2></header>
<p>
Beneficio Media, s.r.o.<br>
Bohdalecká 8/1420, Praha 10, 101 00<br>
Sídlíme v 1. patře.
</p>
<div class="redBox">
<ul>
<li><a href="mailto:info@beneficio.cz" onclick="ga('send', 'event', 'mailto', 'footer', 'info@beneficio.cz');">info@<strong>beneficio.cz</strong></a></li>
<li>+420 <strong>721 352 621</strong></li>
</ul>
</div>
</section>
<div id="map"></div>
</div>
<div class="copy">
© 2015 Beneficio Media, s.r.o.
<span id="btnUp">Nahoru</span>
</div>
</footer>

```

Obrázek: 30 - Patička projektu (autor)

Mapa s adresou je importována skrze Google Maps API a tlačítko pro přesun na vršek stránky je řešeno funkcí `animate()` z JavaScriptové knihovny jQuery. Kliknutím na šipku v pravém dolním rohu tedy dojde k animovanému přesunu na vršek stránky.

Připojení javascriptových knihoven je provedeno na úplném konci HTML dokumentu, aby nedocházelo ke zbytečné prodlevě během načítání webové stránky. Dochází tedy nejprve k načtení CSS souborů a HTML dokumentu a až následně k načtení a provedení skriptů. Uživateli je tak webová stránka zobrazena rychleji.

```

<script src="{basePath}/js/jquery-1.10.1.min.js"></script>
<script src="{basePath}/plugins/Swiper-master/dist/js/swiper.min.js"></script>
<script src="{basePath}/plugins/colorbox-master/jquery.colorbox-min.js"></script>
<script src="https://maps.googleapis.com/maps/api/js"></script>
<script src="{basePath}/plugins/WOW-master/dist/wow.min.js"></script>
<script src="{basePath}/js/min/global-min.js"></script>

```

7.3. Animace

Webové stránky byly doplněny animacemi po najetí obrazovky na jednotlivé prvky. K efektu animace byla použita knihovna WOW. Ta umožňuje přiřadit třídy s názvem efektu a určit tak směr a typ animace.

```
<article class="wow fadeInLeft">
```

Obrázek: 31 - Použití knihovny WOW (autor)

7.4. Fonty

Fonty jsou do stránek vloženy pomocí vlastnosti `@font-face` a jsou načítány do prohlížeče přímo ze stránek, namísto z vlastního počítače.

Nevýhodou načítání „vlastních“ fontů na stránce je jejich pomalejší vykreslení oproti fontům, které jsou běžně součástí operačního systému. Bohužel pro potřeby designu stránek nebyla jiná efektivnější možnost, jak vlastní fonty do stránky importovat.

Druhou možností je importovat fonty z online knihovny společnosti Google, nicméně prodleva mezi načtením byla mnohem větší než načítání ze serveru, na kterém jsou uloženy samotné stránky.

Pro účely optimalizace jsou fonty ořezány o znaky, které nepoužívá čeština, tím je zamezeno načítání kompletní znakové sady, které stejně není potřeba.

Fonty jsou do stránky přidány následovně:

```
@font-face {
  font-family: 'light';
  src: url('../fonts/roboto/roboto_light/Roboto-Light-webfont.eot');
  src: url('../fonts/roboto/roboto_light/Roboto-Light-
webfont.eot?#iefix') format('embedded-opentype'),
      url('../fonts/roboto/roboto_light/Roboto-Light-webfont.woff')
format('woff'),
      url('../fonts/roboto/roboto_light/Roboto-Light-webfont.ttf')
format('truetype'),
      url('../fonts/roboto/roboto_light/Roboto-Light-
webfont.svg#robotolight') format('svg');
  font-weight: normal;
  font-style: normal;
}
```

Obrázek: 32 - Použití fontů pomocí `@font-face` (autor)

7.5. Obrázky

Aby bylo dosaženo co nejnižšího počtu požadavků odeslaných na server, byly nejčastěji používané obrázky seskupeny do jednoho obrázku. Takto zabalené obrázky, které jsou dále na stránce rozmístěny pomocí kaskádových stylů, se nazývají CSS Sprites.



Obrázek: 33 – CSS sprite projektu (autor)

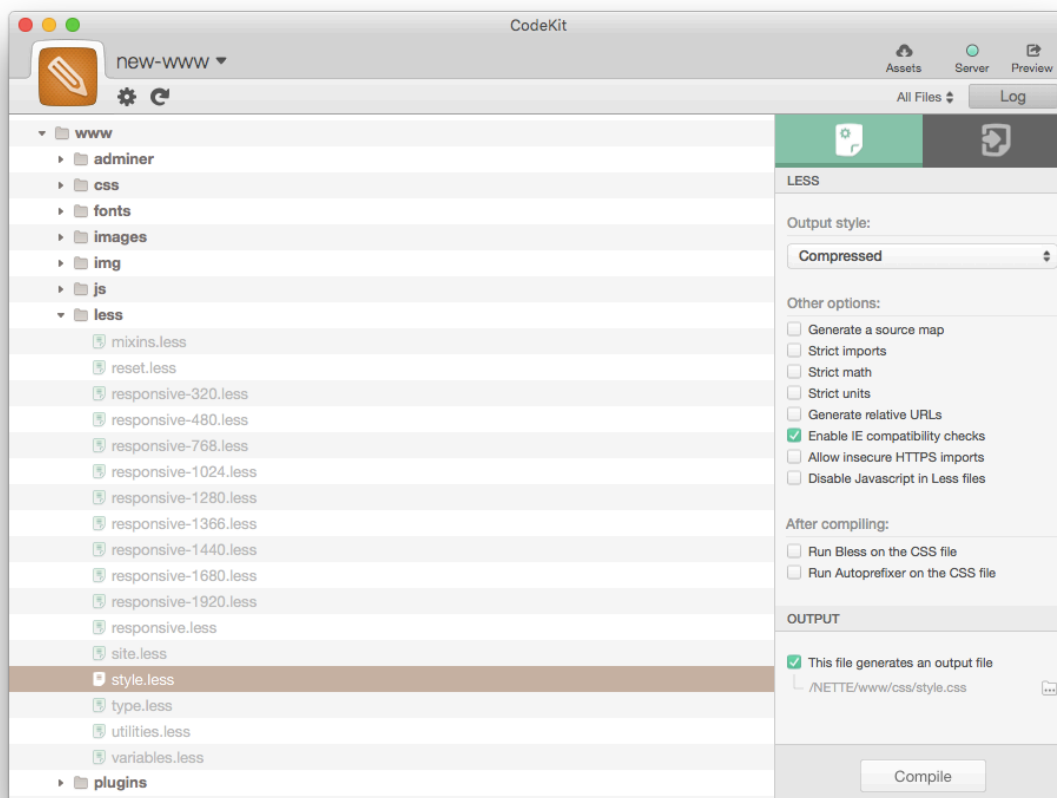
Obrázky byly vytvořeny v dvojnásobném rozlišení než je běžné pro desktopový počítač, aby nebylo potřeba vytvářet pro mobilní telefony obrázky ve větším rozlišení samostatně.

Dvojnásobným rozlišením bylo docíleno ostrých ikon jak na desktopovém zařízení, tak na mobilních zařízeních s vysokým rozlišením, jako je například Apple iPhone, iPad atd. Navíc jsou obrázky optimalizovány i pro nové verze počítačů Apple s retina displayem, tak pro ostatní zařízení s vysokým rozlišením.

Obsahové grafické prvky jsou v případě stránky s referencemi kompromisem mezi datovou velikostí a vzhledem. Během vývoje stránek byl nalezen problém s redukcí velikosti u větších poloprůhledných obrázků v PNG formátu. Obrázky v rozměru 1600x560px mají v průměru 500kB. Při vyšším počtu těchto obrázků na jedné stránce se jedná o více než 7 MB dat. Tím vzniká při pomalejším připojení k internetu nepohodlné prohlížení, neboť jsou obrázky načteny jen částečně.

7.6. Kaskádové styly

Kaskádové styly jsou generovány ze souborů LESS pomocí aplikace codeKit.



Obrázek: 34 – CodeKIT (auto)

Aplikace spravuje kompilaci souborů a správnou interpretaci CSS vlastností do konečného souboru s kaskádovými styly.

Po přidání projektu do aplikace byla automaticky sledována změna souborů, kde po jejich uložení došlo k opětovné kompilaci a vytvoření finálního style.css souboru.

Struktura kaskádových stylů byla rozdělena do několika samostatných souborů pro zlepšení orientace a pořádku. Soubory byly rozděleny na soubory věnované obecným částem, proměnným, utilitám a tzv. Mixin soubor, který obsahuje funkce a předpřipravené části jako je clearfix, borderradius, přidané pluginy atd..., aby byly veškeré kaskádové styly v jednom CSS souboru. Tím došlo ke snížení počtu HTTP požadavků a zrychlení načítání stránky.

Kaskádové styly byly rozděleny pro lepší orientaci do několika souborů mezi nimi. Prvním souborem je reset.less, který nastavuje defaultní hodnoty jednotlivým prvkům tak, aby byly ve všech prohlížečích zobrazeny stejně.

Další vložený styl je swiper.min.css, pro implementaci pohyblivého slideru, colorbox.css pro popup okno s kontaktním formulářem a animate.less doplňující knihovnu podporující animace.

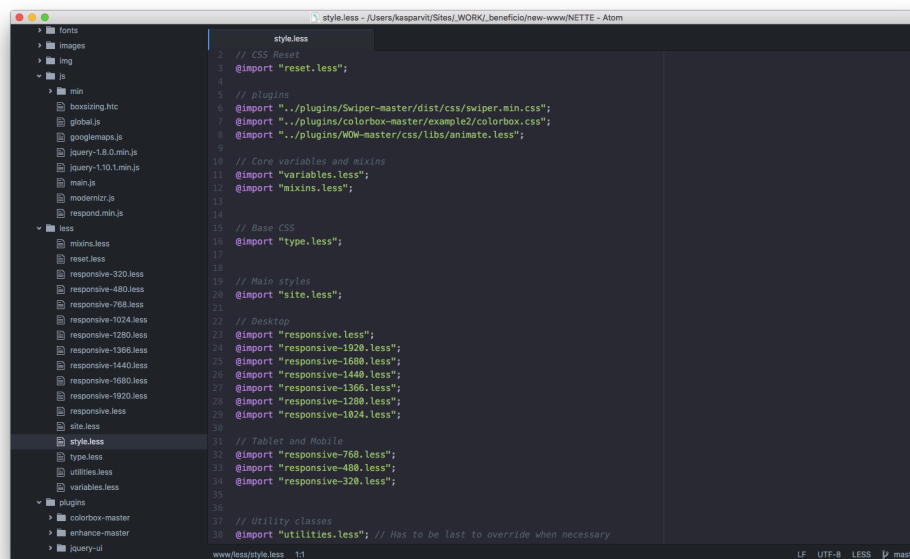
Důležitými styly jsou potom variable.less obsahující globální proměnné a již zmíněný mixin.less, obsahující funkce a předpřipravené části. Definice fontů jsou obsaženy v souboru type.less. Soubor obsahuje importované fonty pomocí @font-face metody a vynulování vlastností pro ostatní prvky, jako jsou číslované a nečíslované seznamy, odstavce, tabulky, atd..

Hlavním stylem je site.less. Tento soubor obsahuje kaskádové styly definující webové stránky jako celek. V souboru se definuje layout, pozice prvků, barvy boxů atd..

Zbytek připojených stylů obsahuje definice pro responzivní verzi stránek. Styly byly rozděleny podle běžných rozlišení šířky od 320px do 1920px. V každém souboru jsou definovány vlastnosti stránek rozšiřující soubor site.less pro konkrétní rozlišení zařízení.

Veškeré soubory byly importovány do jednoho souboru (style.less). Po uložení kteréhokoliv souboru dojde k nové komprimaci všech souborů do výsledného souboru (style.css).

Výsledný soubor byl navíc minifikován. Minifikací došlo k odstranění veškerých bílých znaků (mezery, tabulátory) a tím byla snížena celková velikost souboru.



```
1 // CSS Reset
2 @import "reset.less";
3
4 // plugins
5
6 @import "../plugins/Swiper-master/dist/css/swiper.min.css";
7 @import "../plugins/colorbox-master/example2/colorbox.css";
8 @import "../plugins/MOM-master/css/Libs/animate.less";
9
10 // Core variables and mixins
11 @import "variables.less";
12 @import "mixins.less";
13
14 // Base CSS
15 @import "type.less";
16
17 // Main styles
18 @import "site.less";
19
20 // Desktop
21 @import "responsive-1920.less";
22 @import "responsive-1680.less";
23 @import "responsive-1440.less";
24 @import "responsive-1280.less";
25 @import "responsive-1024.less";
26 @import "responsive-768.less";
27 @import "responsive-480.less";
28 @import "responsive-320.less";
29
30 // Tablet and Mobile
31 @import "responsive-768.less";
32 @import "responsive-480.less";
33 @import "responsive-320.less";
34
35 // Utility classes
36 @import "utilities.less"; // Has to be last to override when necessary
```

Obrázek: 35 - Atom.app (autor)

8. Závěr

Z historie vývoje internetu vyplývá, že dostupné technologie se mění opravdu rychle a je potřeba s nimi držet krok. Za posledních sedm let se dramaticky změnil podíl webových prohlížečů a na přední pozici se dostal Google Chrome, který vystřídal Internet Explorer a Firefox.

Jako jediný ve svém vývoji zaostal značkový jazyk HTML, který se od svého počátku zásadně nezměnil. Po bezmála patnácti letech a příchodu nové verze HTML5, došlo k výraznějším změnám v sémantickém významu jednotlivých HTML značek a k přidání řady nových, které pomocí atributů umožňují mimo jiné pracovat různě s formulářovými prvky na mobilním telefonu a stolním počítači.

Největší pokrok byl zaznamenán v návrhu uživatelského rozhraní (UI/UX) a v celkovém přístupu k návrhu webu. Při návrhu je potřeba nově počítat s širokou škálou nových mobilních zařízení s vysokými „retina“ rozlišeními a také s tabletovými zařízeními. Rostoucí počet uživatelů internetu, kteří přistupují k webu skrze mobilní telefon a tablet, přispěl k výraznému ztížení návrhu webu a k příchodu tzv. „Responzivního webdesignu“.

Ruku v ruce s příchodem mobilních zařízení došlo k optimalizaci webů tak, aby je bylo nejenom pohodlné ovládat, ale také, aby je bylo možné načíst na mobilním telefonu v reálném čase. Zároveň nesmí docházelo k nadměrnému stahování dat během prohlížení webu.

V praktické části bakalářské práce byly vytvořeny webové stránky internetové agentury, které byly přizpůsobeny novým trendům, mezi které patří především funkčnost v mobilních zařízeních, jako je mobilní telefon a tablet. Snížení datové velikosti jednotlivých souborů, pomocí komprese ušetřilo téměř 50% přenesených dat během načítání webu. Minifikace kaskádových stylů snížila datovou velikost stylů. Spojením veškerých stylů do jednoho, navíc došlo ke snížení počtu požadavků na server a zrychlení načítání stránek. Tím bylo docíleno celkového zrychlení webu a zvýšení uživatelského komfortu během prohlížení.

Pomocí techniky CSS sprite zjištěné v teoretické části práce jsou obrázky nejen ostré na zařízeních s vysokým rozlišením, ale také mají nižší datovou velikost a zároveň snižují počet HTTP požadavků na server. V on-line nástroji Google Pagespeed dosáhl projekt skóre 90 z 100 bodů ve výkonnosti a 100 z 100 bodů z uživatelského dojmu. Tento

test dokazuje, že informace zjištěné v teoretické části bakalářské práce fungují v praxi a vedou ke zvýšení rychlosti načítání stránek i k vyššímu uživatelskému prožitku z prohlížení webu.

9. Přílohy

Zdrojové soubory praktické části jsou dostupné na webové adrese:

<https://drive.google.com/folderview?id=0B2qQlgInG95sTU1UcUVCY1Z4cWM&usp=sharing>

Živá prezentace je dostupná na adrese www.beneficio.cz.

9.1. Seznam obrázků

Obrázek: 1 - HTTP protokol 0.9 [28]	3
Obrázek: 2 - HTTP protokol 1.0 [28]	4
Obrázek: 3 - Zápis Doctype v HTML5 (autor).....	9
Obrázek: 4 - Kořenový prvek HTML (autor)	9
Obrázek: 5 - Hlavička HTML (autor).....	9
Obrázek: 6 - Struktura nadpisů v HTML [29]	10
Obrázek: 7 - Využití <hgroup> ve struktuře nadpisů [29]	11
Obrázek: 8 - Struktura nadpisů s využitím <hgroup> [29].....	11
Obrázek: 9 - Ukázka použití navigace v HTML5 [29].....	11
Obrázek: 10 - Animace v CSS3 [49]	12
Obrázek: 11 - Rotece v CSS3 [48].....	13
Obrázek: 12 - Použití RGBA modelu [22]	13
Obrázek: 13 - Použití funkcí v LESS [20].....	15
Obrázek: 14 - Funkce border-radius() [34].....	15
Obrázek: 15 - Internet Explorer 1.0 [35]	16
Obrázek: 16 - Google Chrome [3]	18
Obrázek: 17 - Prohlížeč Opera 3.62 [43].....	19
Obrázek: 18 - Fixní layout [16]	24
Obrázek: 19 - Fluidní layout [16]	24
Obrázek: 20 – CSS sprite [19]	25
Obrázek: 21 - Použití @Font-face metody [18]	26
Obrázek: 22 – Použití ligatury [18]	27
Obrázek: 23 - Grid systémy (autor).....	27
Obrázek: 24 – Google pagespeed test (autor).....	29
Obrázek: 25 - Ukázka desktopové verze stránek (autor).....	30

Obrázek: 26 - Ukázka mobilní verze stránek (autor).....	31
Obrázek: 27 - Doctype a hlavička projektu (autor)	32
Obrázek: 28 - Struktura těla projektu (autor).....	33
Obrázek: 29 - Hlavička těla projektu (autor)	34
Obrázek: 30 - Patička projektu (autor)	35
Obrázek: 31 - Použití knihovny WOW (autor).....	36
Obrázek: 32 - Použití fontů pomocí @font-face (autor).....	36
Obrázek: 33 – CSS sprite projektu (autor).....	37
Obrázek: 34 – CodeKIT (auto)	38
Obrázek: 35 - Atom.app (autor).....	39

9.2. Seznam tabulek

Tabulka 1 - Přehled verzí HTML a roku jejich vydání (autor).....	6
Tabulka 2 - Podíl vyhledávačů roce 2003 [42].....	20
Tabulka 3 - Podíl vyhledávačů v roce 2005 [42].....	20
Tabulka 4 - Podíl vyhledávačů v roce 2008 [42].....	20
Tabulka 5 - Podíl vyhledávačů v roce 2010 [42].....	21
Tabulka 6 - Podíl vyhledávačů v roce 2012 [42].....	21
Tabulka 7 - Podíl vyhledávačů v roce 2015 [42].....	21

9.3. Seznam použité literatury

1. AMRAN, A. Úvod do grid systémů [online]. 2011 [cit. 2016-3-3]. Dostupné z: <http://blog.meebio.cz/clanek/158/uvod-do-grid-systemu/>
2. BITTNER, H. 1. díl - Mobile first responzivní design [online]. 2015 [cit. 2016-3-3]. Dostupné z: <http://www.itnetwork.cz/html-css/responzivni-webdesign/mobile-first-responzivni-design/>
3. BROTHERSOFT. Google Chrome 1.0.154.48 Screenshots [online]. [cit. 2016-7.3.]. Dostupné z: <http://google-chrome.brothersoft.com/google-chrome1.0.154.48-screenshots-2>
4. CRAIG GRANNELL, V. S., DIONYSIOS SYNODINOS. The Essential Guide to HTML5 and CSS3 Web Design. Vyd. 1: Friends of Ed, 2012. ISBN 978-1-4302-3786-0

5. CREATIVEBLOG. Design sites using the Responsive Grid System [online]. 2012 [cit. 2016-3-3]. Dostupné z: <http://www.creativebloq.com/css3/design-sites-using-responsive-grid-system-9122926>
6. CZ.NIC. O DOMÉNÁCH A DNS [online]. [cit. 2016-3-3]. Dostupné z: <http://www.dnssec.cz/page/312/about-domains-and-dns/>
7. FAME, I. H. O. Paul Mockapetris [online]. [cit. 2016-3-3]. Dostupné z: <http://www.internethalloffame.org/inductees/paul-mockapetris>
8. FARUK, P., ALEX, RYAN, PATRICK, STU, AND RICHARD. Respond to your user's browser features. [online]. [cit. 2016-3-3]. Dostupné z: <https://modernizr.com/>
9. GOLDSTEINOVÁ ALEXIS, L. L., WEYLOVÁ ESTELLE. HTML5 a CSS3 pro webové designéry. Vyd. 1: Zoner Press, 2011. ISBN 978-80-7413-166-0
10. GOOGLE. About PageSpeed Insights [online]. 2015 [cit. 2016-3-3]. Dostupné z: <https://developers.google.com/speed/docs/insights/about>
11. GRIGORIK, I. High Performance Browser Networking: What every web developer should know about networking and web performance. Vyd. 11. 9. 2013. ISBN 1449344763
12. HOLOUBEK, R. Historie a vývoj webových prohlížečů [online]. 2014 [cit. 2016-3-3]. Dostupné z: http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xholoub1_historie_browseru.htm
13. JANOVSÝ, D. Délkové jednotky v CSS [online]. [cit. 2016-3-3]. Dostupné z: <http://www.jakpsatweb.cz/css/css-jednotky.html>
14. JONES, B. Codekit [online]. [cit. 2016-3-3]. Dostupné z: <https://incident57.com/codekit/index.html>
15. KAŽMIERCZAK, B. Opera 17 goes Stable [online]. 2013 [cit. 2016-3-3]. Dostupné z: <https://http://www.opera.com/blogs/desktop/2013/10/opera-17-final/>
16. KNIGHT, K. Fixed vs. Fluid vs. Elastic Layout: What's The Right One For You? [online]. 2009 [cit. 2016-3-3]. Dostupné z: <http://www.smashingmagazine.com/2009/06/fixed-vs-fluid-vs-elastic-layout-whats-the-right-one-for-you/>
17. KOSEK, J. Historie a vývoj HTML [online]. [cit. 2016-3-3]. Dostupné z: <http://htmlguru.cz/uvod-historie.html>
18. KŘÍŽ, P. Jak na ikonové fonty [online]. 2013 [cit. 2016-3-3]. Dostupné z: <http://blog.meebio.cz/clanek/162/jak-na-ikonove-fonty/>
19. LENNARTZ, S. The Mystery Of CSS Sprites: Techniques, Tools And Tutorials [online]. 2009 [cit. 2016-3-3]. Dostupné z: <https://http://www.smashingmagazine.com/2009/04/the-mystery-of-css-sprites-techniques-tools-and-tutorials/>
20. LESSCSS. Getting Started [online]. [cit. 2016-3.3.]. Dostupné z: <http://lesscss.org/>
21. LESSCSS. Mixins [online]. [cit. 2016-3-3]. Dostupné z: <http://lesscss.org/features/-mixins-feature>

22. MCDONALD, D. Color [online]. 2016 [cit. 2016-3-3]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Web/CSS/color_value#rgba\(\)](https://developer.mozilla.org/en-US/docs/Web/CSS/color_value#rgba())
23. MICHÁLEK, M. SVG [online]. 2016 [cit. 2016-3-3]. Dostupné z: <http://www.vzhurudolu.cz/prirucka/svg>
24. MILLS, C. Practical CSS3: Develop and Design. Vyd. 1: Peachpit Press, 2012. ISBN 978-0-321-82372-4
25. ML., J. M. Webové prohlížeče: současní lídři a jejich historie (2/2) [online]. 2013 [cit. 2016-3-3]. Dostupné z: <http://www.root.cz/clanky/webove-prohlizece-soucasni-lidri-a-jejich-historie-2-2/>
26. MOZILLA. Looking back. Looking ahead. [online]. [cit. 2016-3-3]. Dostupné z: <https://http://www.mozilla.org/en-US/about/history/>
27. NEAGLE, C. A visual history of Microsoft's Internet Explorer [online]. 2015 [cit. 2016-3-3]. Dostupné z: <http://www.networkworld.com/article/2897729/software/a-visual-history-of-microsofts-internet-explorer.html>
28. OREILLY. Chapter 9. Brief History of HTTP [online]. [cit. 2016-3-3]. Dostupné z: <http://chimera.labs.oreilly.com/books/1230000000545/ch09.html>
29. PILGRIM, M. DIVE INTO HTML5 [online]. 2009 [cit. 2016-3-3]. Dostupné z: <http://kniha.html5.cz/>
30. PILIGRAM, M. CO TO VŠECHNO ZNAMENÁ? [online]. 2009 [cit. 2016-3-3]. Dostupné z: <http://kniha.html5.cz/semantics.html>
31. SIKOS, L. Web Standards. Vyd. 1: Apress, 2011. ISBN 978-1-4302-4041-9
32. SOOJIAN, C. RESPONSIVE WEB DESIGN: A BRIEF HISTORY [online]. 2015 [cit. 2016-3-3]. Dostupné z: <http://www.socialmediatoday.com/content/responsive-web-design-brief-history>
33. SPLASHOMNIMEDIA. Desktop-First vs. Mobile-First Web Design: Which is Best for YOUR Business? [online]. 2013 [cit. 2016-3-3]. Dostupné z: <http://www.slideshare.net/splashomnimedia/desktopfirst-vs-mobilefirst-web-design>
34. SWISHER, J. Border-radius [online]. 2016 [cit. 2016-3-3]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS/border-radius>
35. VANHEMERT, K. Internet Explorer Turns 15 Today [online]. 2010 [cit. 2016-3-3]. Dostupné z: <http://gizmodo.com/5614186/internet-explorer-turns-15-today>
36. VOGT, N. Responsive CSS "Mobile First" vs "Desktop First" Media Queries [online]. 2014 [cit. 2016-3-3]. Dostupné z: <http://www.h3xed.com/web-development/responsive-css-mobile-first-vs-desktop-first-media-queries>
37. W3C. The CSS saga [online]. [cit. 2016-3-3]. Dostupné z: <https://http://www.w3.org/Style/LieBos2e/history/>
38. W3C. Heading content [online]. [cit. 2016-3-3]. Dostupné z: <https://http://www.w3.org/TR/html5/dom.html - heading-content>
39. W3C. The history of the Web [online]. [cit. 2016-3-3]. Dostupné z: https://http://www.w3.org/wiki/The_history_of_the_Web
40. W3C. Introduction to CSS3 [online]. [cit. 2016-3-3]. Dostupné z: <https://http://www.w3.org/TR/2001/WD-css3-roadmap-20010523/>

41. W3C. World Wide Web Consortium [online]. [cit. 2016-3-3]. Dostupné z: <https://http://www.w3.org/>
42. W3SCHOOLS. Browser Statistics [online]. 2016 [cit. 2016-3-3]. Dostupné z: http://www.w3schools.com/browsers/browsers_stats.asp
43. WIKIPEDIA. History of the Opera web browser [online]. [cit. 2016-7-3]. Dostupné z: https://en.wikipedia.org/wiki/History_of_the_Opera_web_browser
44. WIKIPEDIA. Google Chrome [online]. 2016 [cit. 2016-3-3]. Dostupné z: https://en.wikipedia.org/wiki/Google_Chrome
45. WILSON, B. Opera [online]. [cit. 2016-3-3]. Dostupné z: <http://www.blooberry.com/indexdot/history/opera.htm>
46. ZARTNER, S. Box-shadow [online]. 2015 [cit. 2016-3-3]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS/box-shadow>
47. ZARTNER, S. Text-shadow [online]. 2016 [cit. 2016-3-3]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS/text-shadow>
48. ZARTNER, S. Transform-function [online]. 2016 [cit. 2016-3-3]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function#rotate\(\)](https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function#rotate())
49. ZARTNER, S. Transition [online]. 2016 [cit. 2016-3-3]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS/transition>