

Responzivní webdesign a jeho praktické využití

Bakalářská práce

Vedoucí práce:

Ing. Jiří Lýsek, Ph.D.

Ondřej Slíva

Brno 2015

Na tomto místě bych rád poděkoval Ing. Jiřímu Lýskovi, Ph.D., který trpělivě vedl moji bakalářskou práci a poskytoval mi cenné rady a náměty během jejího vypracovávání. Velký dík patří rovněž mojí rodině, přítelkyni a nejbližším přátelům za trpělivost, pochopení a podporu. Děkuji také firmě Návrh loga, který mi umožnila vyzkoušet si, jak vypadá pracovní pozice webového kodéra v praxi, ve které jsem se ocitl při tvorbě praktické části této bakalářské práce.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Responzivní webdesign a jeho praktické využití** vypracoval/a samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom/a, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 5. ledna 2015

Abstract

Slíva, O. Responsive webdesign and its practical use. Bachelor thesis. Brno: Mendel University, 2015.

This bachelor thesis deals with responsive webdesign. In the theoretical part is mentioned what actually responsive webdesign is, what does it deal with, why it makes sense to deal with it, what is the alternative to it and what are its advantages and disadvantages. Furthermore are mentioned meta tag viewport and three basic elements of responsive webdesign, which are fluid layout, media queries and flexible images. The practical part deals with coding of responsive websites (according to Mobile first approach) for the company Návrh loga according to obtained graphical design.

Keywords

Responsive, webdesign, mobile first, coding, HTML, CSS.

Abstrakt

Slíva, O. Responzivní webdesign a jeho praktické využití. Bakalářská práce. Brno: Mendelova univerzita v Brně, 2015.

Tato bakalářská práce se zabývá responzivním webdesignem. V teoretické části je zmíněno co to vlastně responzivní webdesign je, čím se zabývá, proč má smysl se mu věnovat, jaká k němu existuje alternativa a jaké jsou jeho výhody a nevýhody. Dále je zmíněn meta tag viewport a tři základní prvky responzivního webdesignu, kterými jsou plovoucí layout, media queries a flexibilní obrázky. Praktická část se zabývá nakódováním responzivních webových stránek (dle přístupu Mobile first) pro firmu Návrh loga dle obdrženého grafického návrhu.

Klíčová slova

Responzivní, webdesign, mobile first, kódování, HTML, CSS.

Obsah

1	Úvod a cíl práce	15
1.1	Úvod	15
1.2	Cíl práce	16
2	Co je responzivní webdesign	17
2.1	Dostupné informace a zdroje	17
2.2	Proč se mu věnovat.....	18
2.3	Jiný způsob řešení – samostatné weby	21
2.4	Výhody	22
2.5	Nevýhody.....	23
3	Základní prvky	25
3.1	Plovoucí layout.....	25
3.1.1	Layout s pevnou šířkou	25
3.1.2	Princip plovoucího layoutu	25
3.1.3	Změna velikosti fontů.....	26
3.1.3.1	Pixely.....	26
3.1.3.2	Jednotka em	27
3.1.3.3	Jednotka rem	27
3.1.3.4	Procenta.....	28
3.1.4	Přepočet pixelů na relativní jednotky	28
3.1.5	Mřížkový systém a frameworky.....	30
3.2	Media queries	31
3.2.1	Media types	32
3.2.2	Media features	32
3.2.3	Logická klíčová slova	33
3.2.4	Pravidla	33
3.2.5	Sestavení mediálního dotazu.....	33
3.2.6	Podpora v prohlížečích.....	34
3.3	Flexibilní obrázky	35

3.4	Meta tag viewport	35
4	Použité technologie	38
4.1	HTML	38
4.2	CSS.....	38
4.3	JavaScript.....	39
5	Použité nástroje	40
5.1	Sublime Text	40
5.2	Adobe Photoshop.....	40
5.3	Podio.....	41
6	Metodika	42
6.1	Obdržený grafický návrh od firmy Návrh loga.....	42
6.1.1	Řezání grafiky ve Photoshopu	43
6.2	Kódování responzivních webových stránek pro firmu Návrh loga.....	44
6.2.1	Přístupy desktop first a mobile first	44
6.2.2	Určování breakpointů	45
6.2.3	Použití relativních jednotek.....	47
6.2.4	Použitý styl písma.....	47
7	Responzivní web pro firmu Návrh loga	48
7.1	Menu	48
7.2	Hlavička a patička.....	49
7.3	Úvodní stránka.....	49
7.4	Sekce Proč logo	50
7.5	Sekce Tvorba loga.....	50
7.6	Sekce Cena loga	50
7.7	Sekce Ochranné známky	51
7.8	Sekce Firemní styl	51
7.9	Sekce Naše reference	52
7.10	Sekce Kontakty a o nás.....	52
8	Diskuse	53
8.1	Google Mobile-friedly test na použitelnost v mobilech.....	53

Obsah	11
8.2 Optimalizace rychlosti načítání	54
8.3 CSS pro tisk.....	54
9 Závěr	56
10 Literatura	58
A Snímky obrazovek	63
B Přiložené DVD	71

Seznam obrázků

Obr. 1	Trh s danými typy produktů podle počtu zásilek v letech 2010-2017 (zásilky v milionech)	19
Obr. 2	Procentuální podíl počtu návštěv internetových stránek z jednotlivých druhů zařízení v období říjen 2012 – říjen 2014 (celosvětově)	20
Obr. 3	Procentuální podíl trafficu z mobilních zařízení (mobilní telefony, tablety) podle oblasti, porovnání září 2014 a září 2013	21
Obr. 4	Ukázka návrhu vzhledu webu, který pro své rozměry používá pixely	29
Obr. 5	Ukázka hlavičky a patičky stránky (vlevo) a zobrazení menu (vpravo) při načtení webu na mobilním telefonu	64
Obr. 6	Menu, hlavička a patička stránky při zobrazení na tabletu	65
Obr. 7	Menu, hlavička a patička stránky při zobrazení na desktopu	65
Obr. 8	Úvodní stránka při zobrazení na mobilu, vlevo hořejší část, na kterou navazuje část zobrazená vpravo	66
Obr. 9	Úvodní stránka při zobrazení na tabletu	67
Obr. 10	Úvodní stránka při zobrazení na desktopu	68
Obr. 11	Zobrazení tabulky v sekci Ochranné známky, nahoře na mobilu, dole na desktopu	69
Obr. 12	Výsledek Google Mobile-friedly testu na použitelnost v mobilech	70
Obr. 13	Výsledek testu Google PageSpeed Insights – jeho části věnující se uživatelskému dojmu	70

Seznam tabulek

Tab. 1 Trh s danými typy produktů podle počtu zásilek a podílu na trhu v letech 2011-2013 (zásilky v milionech)	18
--	-----------

1 Úvod a cíl práce

1.1 Úvod

Prohlížení webových stránek bylo v minulosti výsadou zejména stolních počítačů, později se přidaly notebooky. Tomu byla přizpůsobena tvorba webových stránek, při které se zpravidla využíval tzv. fixní layout s pevně definovanou šířkou, jelikož rozlišení a velikost obrazovek jednotlivých druhů těchto zařízení si byly vcelku podobné a jejich spektrum nebylo příliš široké.

V posledních několika letech, ale dochází k trvalému a postupnému poklesu podílu návštěv webových stránek z těchto druhů zařízení. Tento jev je způsobený nástupem mobilních zařízení, ze kterých jejich uživatelé mohou přistupovat k Internetu. Jedná se hlavně o mobilní telefony (zejména chytré mobilní telefony, tzv. smartphony), tablety, ale i elektronické čtečky knih či herní konzole.

Prodej smartphonů a tabletů celosvětově rok od roku roste, stejně tak jako jejich podíl na trhu oproti zařízením jako jsou stolní počítače a notebooky, u kterých je trend přesně opačný. S tím souvisí i nezanedbatelný podíl návštěv webových stránek z těchto mobilních zařízení, který rovněž rok od roku stále roste a předpokládá se jeho další růst i v nejbližších letech. Například v České republice dosahuje podíl návštěv webových stránek z mobilních zařízení hodnoty 11–16%. Toto číslo ale výrazně zaostává za celosvětovým průměrem, jehož hodnota se pohybuje kolem 37%. Výjimkou nejsou ani státy, kde se tento podíl blíží 50%, z těch evropských se k této hodnotě blíží např. Nizozemsko, Dánsko, Švédsko či Polsko.

Zmiňovaná mobilní zařízení mají jednu důležitou odlišnost oproti stolním počítačům a notebookům. Je jí velikost zobrazovací plochy, která je výrazně menší a omezená. Prohlížení webových stránek vytvořených původně pro desktopy za pomoci fixního nepřizpůsobivého layoutu je na mobilních zařízeních velmi nepohodlné, obsah je zmenšený a tím pádem nečitelný, proto musíme stránky různě přibližovat a využívat posuvníků pro pohybování se po jejich jednotlivých částech, kdy velmi nepříjemným a nepohodlným je zejména horizontální posuvník. Orientace na takovém webu je poměrně složitá a jeho prohlížení a ovládání rozhodně není pro uživatele mobilních zařízení příjemným zážitkem.

Je tedy patrné, že vyvíjet stránky i nadále za pomoci fixního layoutu pouze pro desktopová zařízení není tou zcela vhodnou cestou. Je potřeba optimalizovat weby i pro mobilní zařízení, které mají typicky menší obrazovky. Jedním ze způsobů je responzivní webdesign, což je přístup k návrhu webových stránek, které přizpůsobují obsah stránky v závislosti na zobrazovacím zařízení a jeho vlastnostech. Jeho cílem je komfortní zobrazení pro všechny uživatele, aby byly stránky přehledné, snadno ovladatelné a práce s nimi příjemná a bezproblémová, nezávisle na tom, jaké zobrazovací zařízení je použito.

Responzivní webdesign je poměrně novou problematikou, první zmínka o něm pochází z 25. května 2010, kdy samotný termín poprvé použil webový vývojář Ethan Marcotte, který je tak považován za jeho zakladatele. V článku definoval tři základní prvky, na kterých stojí jeho princip.

Responzivní webdesign je velmi aktuálním tématem na poli tvorby webových stránek a stále se rozvíjí. Jeho znalost je velkou a ceněnou výhodou v praxi v případě ucházení se o zaměstnání na pozici webového kodéra či vývojáře. Jeho problematika mě velmi zaujala a v budoucnu bych se mu chtěl věnovat i při svojí práci. Z těchto důvodů jsem si právě responzivní webdesign vybral jako téma své bakalářské práce.

1.2 Cíl práce

Hlavním cílem této bakalářské práce je nakódování responzivních webových stránek pro firmu Návrh loga dle grafického návrhu, který jsem od ní obdržel ve formátu PSD. Budu se tedy nacházet v pozici, která se v praxi označuje jako kodér webových stránek, někdy také jako HTML kodér, HTML/CSS kodér či moderněji v angličtině Frontend Web Developer. Pro tvorbu webu budou použity technologie HTML, CSS a JavaScript. V části práce věnující se metodice popíšu, jak jsem při tvorbě webových stránek postupoval. Uvedu i popis jednotlivých sekcí webu, kde budu dbát na zmínění toho, v čem spočívá jejich responzivita.

V teoretické části bakalářské práce nejdříve zmíním co to vlastně responzivní webdesign je, čím se zabývá, proč má smysl se mu věnovat, jaká k němu existuje alternativa a jaké jsou jeho výhody a nevýhody. Následovat bude kapitola věnující se třem základním prvkům responzivního webdesignu, kterými jsou plovoucí layout, media queries a flexibilní obrázky. Zmíním i meta tag viewport, jehož použití je nezbytné pro správnou funkčnost responzivních webových stránek.

2 Co je responzivní webdesign

Responzivní webdesign je přístup k návrhu webových stránek, které přizpůsobují obsah stránky v závislosti na zobrazovacím zařízení a jeho vlastnostech. Jeho cílem je komfortní zobrazení pro všechny uživatele, aby byly stránky přehledné, snadno ovladatelné a práce s nimi příjemná a bezproblémová, ať už jejich návštěvník pracuje s jakýmkoliv zařízením typu stolní počítač, notebook, netbook, tablet, smartphone apod. Reaguje na rapidně rostoucí rozmanitost zařízení, která se připojují k Internetu. Neřeší ovšem jen zobrazení webových stránek, jedná se o komplexní přístup k návrhu webu, který se mimo jiné zabývá i věcmi jako je například prioritizace obsahu, rychlost načítání stránky, přizpůsobení ergonomickým požadavkům mobilních zařízení apod.

První zmínka o responzivním webdesignu se objevila 25. května 2010 na webu A List Apart v článku Responsive Web Design, který napsal Ethan Marcotte (Marcotte, 2010). Přístup, který zde popisoval, byl sice jednoduchý, ale zároveň revoluční. Použil k tomu tři existující nástroje – mediální dotazy, plovoucí struktury a škálovatelné obrázky – a vytvořil web, který se báječně zobrazoval na mnoha různých rozlišeních. V článku přesvědčoval designéry, aby nevytvářeli design na míru pro každé ze stále rostoucího počtu webových zařízení, ale chápali je jako různé aspekty stejného prožitku (Kadlec, 2014).

2.1 Dostupné informace a zdroje

Jelikož je problematika responzivního webdesignu vcelku novou záležitostí, tak je množství dostupných kvalitních zdrojů v českém jazyce poměrně malé. Například v době psaní bakalářské práce existovala pouze jedna kniha, která se zabývala přímo a jen responzivním webdesignem a byla v češtině – Responzivní design profesionálně od Tima Kadlece, kterou vydal Zoner Press v roce 2014 (Kadlec, 2014). Nejedná se ovšem o knihu, kterou by napsal český autor, ale pouze o překlad anglického originálu s názvem Implementing Responsive Design. Dále na trhu s českou odbornou počítačovou literaturou můžeme najít pouze pár knih, které se zabývají technologiemi HTML5 a CSS3, ve kterých je alespoň malá zmínka o responzivním webdesignu. Ta má bohužel vždy podobu jen několika jednotek stránek, maximálně jedné kapitoly. Jako příklad lze uvést HTML5 a CSS3 (Castro, Hyslop, 2012), Vytváříme mobilní web a aplikace pro chytré telefony a tablety (Castledine, Eftos, Wheeler, 2013), HTML5 a CSS3 (Hogan, 2011).

Z českých zdrojů tedy převažují ty internetové, které mají většinou podobu blogů, na kterých autoři různých kvalit a odbornosti publikují své názory, postoje a zkušenosti s responzivním webdesignem. K těm nejkvalitnějším lze zařadit blog Martina Michálka (www.vzhurudolu.cz), který má dlouholeté zkušenosti s webdesignem a specializuje se na problematiku responzivního webdesignu a s ním souvisejících technologií HTML5 a CSS3. Mimo jiné pořádá i odborné kurzy a přednášky z těchto oblastí.

Situace v cizojazyčných zdrojích je již daleko lepší než v těch českých. Například v anglickém jazyce je dostupné již vcelku velké množství knih, které se věnují přímo responzivnímu webdesignu. Rovněž komunita okolo této problematiky je daleko větší.

2.2 Proč se mu věnovat

Následující tabulka ukazuje vývoj množství zásilek jednotlivých druhů produktů (smartphone, tablet, přenosné PC, stolní PC) a jejich podíl na trhu v letech 2011, 2012 a 2013. Zde je nutno vysvětlit význam slova zásilka, který zdroj dat definuje jako zásilky do distribučních kanálů a koncovým zákazníkům.

Tab. 1 Trh s danými typy produktů podle počtu zásilek a podílu na trhu v letech 2011-2013 (zásilky v milionech)

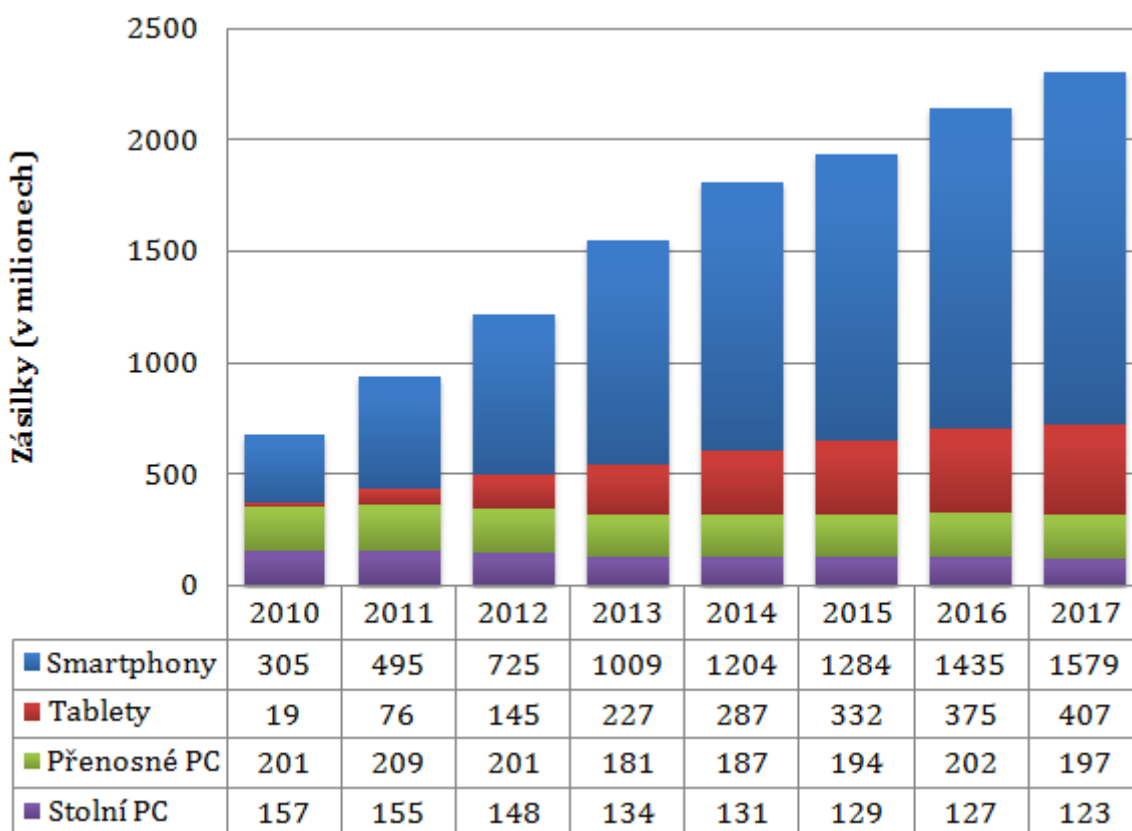
Typ produktu	2011 zásilky	2011 podíl na trhu	2012 zásilky	2012 podíl na trhu	2013 zásilky	2013 podíl na trhu
Smartphone	494,5	53,1%	722,4	60,1%	1013,2	65,1%
Tablet	72,0	7,7%	128,3	10,7%	227,3	14,6%
Přenosné PC	209,1	22,5%	202,0	16,8%	180,9	11,6%
Stolní PC	154,8	16,6%	148,4	12,4%	134,4	8,6%
Celkem	930,4	100,0%	1201,1	100,0%	1556,0	100,0%

Zdroj: <http://www.idc.com/getdoc.jsp?containerId=prUS23958513>,
<http://www.idc.com/getdoc.jsp?containerId=prUS24314413>

Z tabulky lze vyčíst, že rok od roku stoupá počet zásilek smartphonů a tabletů, a to vcelku dosti rychlým tempem. Například v roce 2012 byl nárůst počtu zásilek smartphonů 46,1% a tabletů dokonce 78,2% oproti roku 2011. U přenosných a stolních PC je trend opačný, jejich počet zásilek a podíl na trhu klesá. U přenosných PC klesl počet zásilek v roce 2012 o 3,4% a u stolních PC o 4,1% oproti roku 2011. Tento trend bude pokračovat i nadále do budoucnosti, jak můžeme vidět na následujícím obrázku, který zachycuje situaci na trhu v letech 2010 až 2017 s tím, že pro rok 2013 a následující se jedná o odhadovaná čísla. Můžeme konstatovat, že počet zásilek a podíl na trhu bude u smartphonů a tabletů i nadále růst. U přenosných PC bude klesat jejich podíl na trhu, zatímco počet zásilek bude oscilovat kolem hodnoty 200 milionů. U stolních PC bude počet zásilek, stejně tak jako jejich podíl na trhu, klesat.

Sledovat vývoj počtu těchto zařízení a jejich podíl na trhu je důležité vzhledem k tomu, že jejich uživatelé z nich mohou přistupovat k Internetu. Kromě těchto hlavních představitelů existují v dnešní době i další druhy zařízení, za pomoci kterých lidé konzumují webový obsah. Jako příklad mohou posloužit chytré televize, herní konzole (PlayStation, Xbox) či čtečky elektronických knih se zabudovanými prohlížeči. Je patrné, že rozmanitost zařízení s přístupem na Internet je opravdu

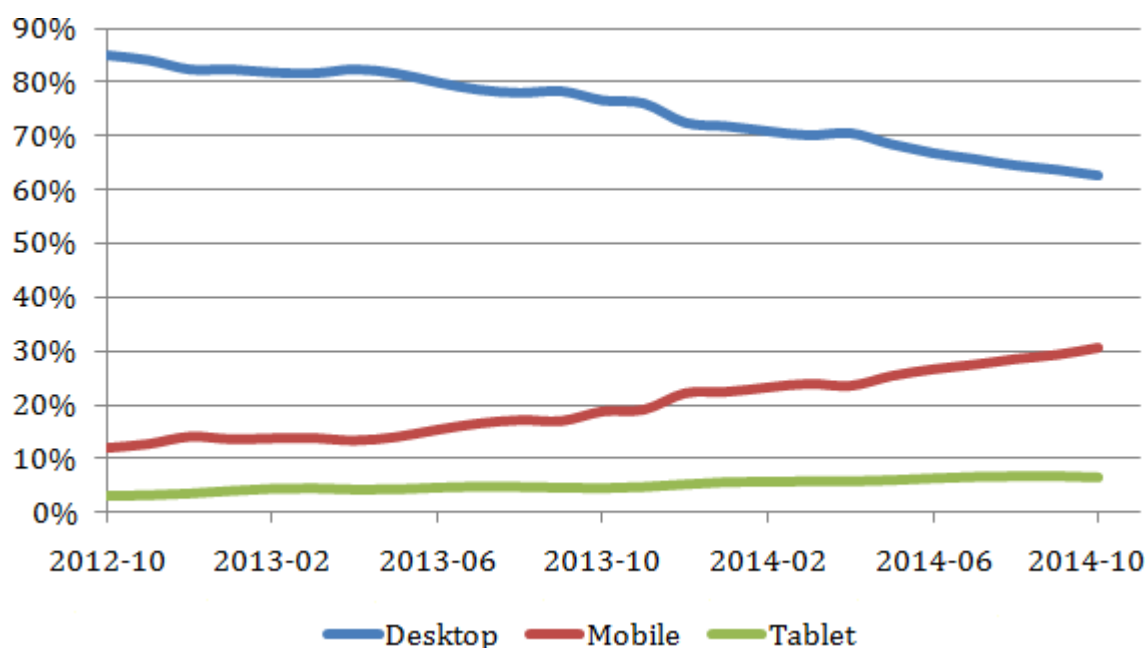
velká a navíc neklesá, ale ve skutečnosti narůstá – díky rychlosti změn v technologii, ale má na ni vliv i snižování nákladů na vytváření těchto zařízení (např. komoditizace hardwaru chytrých telefonů, zdarma dostupné operační systémy a snižování cen integrovaných zařízení typu systém na čipu). Ukazuje se, že trh podporuje tuto teorii – chytré telefony si může dovolit čím dál tím víc lidí. Snižuje se bariéra pro vstup a otevírají se brány pro další a další hráče, kteří mohou přijít na trh s dalšími zařízeními a systémy. S tím souvisí různorodost velikosti rozlišení těchto zařízení, ty oblíbené mají šířku obrazovky od 320 do 1920 pixelů, původně podporovaný úzus přestal platit – už neexistuje žádné standardní rozlišení (Kadlec, 2014).



Obr. 1 Trh s danými typy produktů podle počtu zásilek v letech 2010-2017 (zásilky v milionech)

Zdroj: <http://www.statista.com/statistics/272595/global-shipments-forecast-for-tablets-laptops-and-desktop-pcs/>, <http://www.statista.com/statistics/263441/global-smartphone-shipments-forecast/>

Rostoucí charakter má také podíl návštěvnosti webových stránek (tzv. traffic) z mobilních zařízení. V celosvětovém měřítku má aktuálně hodnotu 37,23% (z toho mobilní telefony 30,67%, tablety 6,56%) z celkového trafficu, jak můžeme vidět na následujícím obrázku, který zachycuje procentuální podíl počtu návštěv internetových stránek z jednotlivých zařízení (desktop, mobilní telefon, tablet) v období říjen 2012 až říjen 2014.

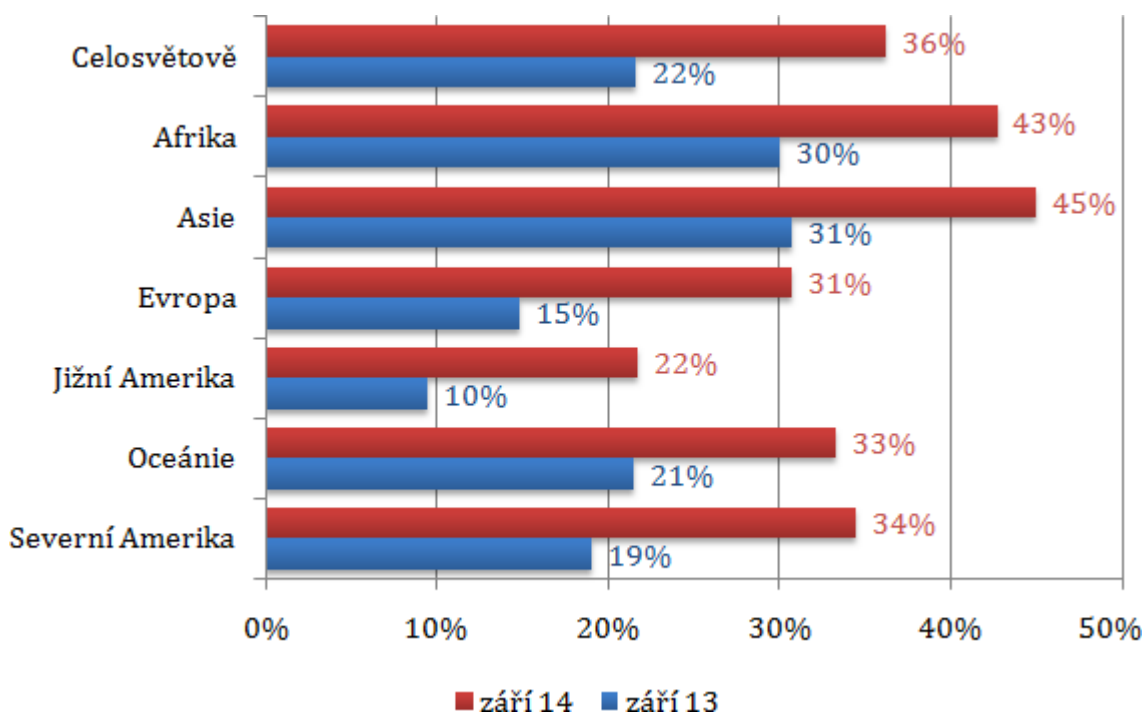


Obr. 2 Procentuální podíl počtu návštěv internetových stránek z jednotlivých druhů zařízení v období říjen 2012 – říjen 2014 (celosvětově)

Zdroj: <http://gs.statcounter.com/#desktop+mobile+tablet-comparison-ww-monthly-201210-201410>

Zajímavé je porovnání podílu trafficu z mobilních zařízení podle jednotlivých oblastí a jeho nárůst za pouhý jeden rok, což můžeme sledovat na dalším obrázku, kde je znázorněn podíl návštěvnosti webových stránek z mobilních zařízení v měsíci září v roce 2013 (modrá barva) a 2014 (červená barva). Například ve většině států v Africe je přístup k internetu větší z mobilních zařízení než z desktopu, stejně tomu tak je i v některých zemích Jižní Asie (StatCounter, říjen 2014a). V Evropě je největší traffic z mobilních zařízení v Nizozemsku (51,22%), Dánsku (46,81%), Švédsku (45,73%) a Polsku (45,45%)(StatCounter, říjen 2014b).

Pokud bychom se zaměřili pouze na Českou republiku, tak podíl návštěv webových stránek z mobilních zařízení zaostává za evropským průměrem a je jedním z nejmenších v Evropě, ovšem ani tak se nejedná o zanedbatelnou hodnotu, konkrétně je to 11–16% (Gemius, 2014; SPIR – NetMonitor, 2014; StatCounter, listopad 2014). Dle údajů z Mobile Internet Fóra 2013 je v České republice 3,7 milionu uživatelů internetu v mobilu, což představuje 27% z počtu aktivních SIM karet a podíl uživatelů, kteří používají pro připojení k Internetu chytrý telefon je 41% (Asociace provozovatelů mobilních sítí, 2013). Sdružení pro internetový rozvoj a jeho měsíční zpráva výzkumného projektu NetMonitor za říjen 2014 uvádí menší počet uživatelů mobilního internetu, a to téměř 3,2 milionu, což může být způsobeno tím, že toto číslo nezahrnuje uživatele, kteří k přístupu na internet využívají pouze mobilní aplikace a nepřístupují přes mobilní prohlížeč. V porovnání s celkovou evidovanou návštěvností tohoto projektu 46,14% internetových uživatelů využívá internet z mobilních zařízení (SPIR – NetMonitor, 2014).



Obr. 3 Procentuální podíl trafficu z mobilních zařízení (mobilní telefony, tablety) podle oblasti, porovnání září 2014 a září 2013

Zdroj: <http://gs.statcounter.com/#desktop+mobile+tablet-comparison-ww-monthly-201309-201409>

2.3 Jiný způsob řešení – samostatné weby

Tento způsob řešení spočívá ve vytvoření samostatných webů, které jsou určeny pro specifické druhy zařízení nebo dokonce pouze pro jedno jediné specifické zařízení. V praxi to obvykle znamená, že existuje jeden web pro mobilní a druhý pro desktopová zařízení. Tyto separátní weby pro mobilní uživatele jsou nejčastěji součástí subdomény m., mobile., t. nebo touch.

Začíná být čím dál víc běžné, že firma provozuje jeden web pro desktopová zařízení, další web pro tablety, ještě jeden pro mobilní zařízení, která je možné ovládat pomocí dotyků, a ještě jeden jednodušší web pro mobilní zařízení, která dotyky nepodporují – to znamená čtyři různé weby pro jedinou firmu.

Tento přístup má nepochybně své přednosti. Když vytváříme pro každý druh zařízení samostatný web, významně snadněji můžeme vytvářet prožitky šité na míru uživatelům těchto zařízení – ať už jde o kontext či chování. Zdali je to ale rozumné, už závisí na projektu, obchodních cílech firmy, na uživatelích, na schopnostech týmu, na rozpočtu apod.

I tento způsob řešení má ovšem svá negativa. Není dobře škálovatelný – máme čtyři weby, které musíme aktualizovat, testovat a udržovat. Navíc se předpokládá, že máme k dispozici nějaký detekční mechanismus, na jehož základě můžeme zjistit, kterým zařízením doručíme kterou z verzí našeho webu. Tento typ detekce (detekce prohlížeče na základě informací, které o sobě poskytuje při zasílání požadav-

ků serveru) je poměrně nespolehlivý. Řetězec uživatelského agenta (řetězec User-Agent v hlavičce HTTP požadavku) lze v mnoha prohlížečích snadno změnit (a velmi často se to i dělá, aby bylo možné popisovanou techniku obejít). I kdyby se nám podařilo na naší mobilní verzi webu implementovat spolehlivou funkcionalitu detekce aktuální generace mobilních prohlížečů, mohli bychom tímto přístupem způsobit nepředvídatelné problémy uživatelům zařízení, která neznáme. Je nutné průběžně detekční mechanismus aktualizovat, protože časem přibývají nová zařízení.

Je to otázka volby: Chceme uživatele nutit, aby používali web, který jsme pro ně vybrali? Nejlepší je nabídnout jim odkaz na standardní verzi webu (často se také používá slovní spojení plná verze) a obráceně a poté jejich rozhodnutí respektovat. Na navádění uživatelů na verzi webu určenou pro mobilní zařízení není nic špatného, ale neměli bychom je nijak dále omezovat a bránit jim v přístupu na plnou verzi stránek.

Dobrým příkladem správné implementace tohoto chování jsou weby služby Facebook. Ta nabízí dva weby určené pro mobilní zařízení: `web.touch.facebook.com`, na který jsou směrováni uživatelé mobilních zařízení s dotykovou obrazovkou, a `web.m.facebook.com`, určený uživatelům, kteří dotykovou obrazovku nemají. Oba weby umožňují provádění základních úkonů a interakcí, které jsou uživatelé ve službě Facebook zvyklí provádět. Verze webu určené pro mobilní zařízení však neumožňují provádět všechny úkony, které nabízí plná verze. Pokud uživatelé úkon, který na verzi webu určené pro mobilní zařízení provést nelze (nebo se provádí na standardní verzi snáze), provést potřebují, mají na obou verzích webu služby určených pro mobilní zařízení k dispozici šikovné odkazy (v zápatí stránek), pomocí nichž mohou snadno přejít na standardní verzi webu. Klíčové je vždy umožnit uživatelům snadný přístup na plnohodnotnou verzi webu a neochudit je o žádnou jeho funkcionalitu. Jednotlivé verze webu od sebe oddělujeme, ale neizolujeme (Kadlec, 2014; Castledine, Eftos, Wheeler, 2013).

Řešení za pomoci separátních mobilních webů se používalo hlavně dříve, ale i v dnešní době je stále v některých ojedinělých případech využitelné. Používá se např. pro starší, technologicky zastaralé weby nebo pro rozsáhlé weby s hlubokou strukturou. Rovněž je dobrou volbou v případě, že chceme uživatelům mobilních zařízení nabízet jiný obsah a funkcionalitu než v plné verzi webu. Stejně tak když už máme nějaký existující desktopový web a potřebujeme ho rychle optimalizovat i pro mobilní zařízení, protože vytvoření samostatné mobilní verze je snazší, rychlejší a levnější než responzivní redesign. Toto řešení je také možné vidět u starých webů, které teprve čeká redesign (Držka, Michálek, 2013; Michálek, 2014).

2.4 Výhody

Nikdo a nic není úplně dokonalé a i responzivní webdesign má jak své výhody, tak i nevýhody. Hlavní pro a proti se pokusím popsat v následujících dvou podkapitolách. Čerpal jsem z více zdrojů najednou, které jsem zkombinoval dohromady, proto je nebudu uvádět za každým z následujících tvrzení a uvedu je zde najednou

(Forgáč, 2013; Jashinsky, 2013; Prokop, 2013; Rampton, 2014; Ting, 2013; Vápeník, 2013). Dodal bych, že zmiňované výhody a nevýhody se v dostupných zdrojích shodovaly a překrývaly. Začnu kladnými stránkami.

Při použití responzivního webdesignu existuje pouze jediná webová stránka, jeden HTML kód, který je snadnější, efektivnější a méně finančně náročné spravovat a udržovat při porovnání s dřívějším řešením oddělené mobilní verze.

Příjemnější ovládání pro uživatele díky automatickému přizpůsobení rozvržení a vzhledu stránek v závislosti na zobrazovacím zařízení a jeho vlastnostech. Stránky se přizpůsobují uživatelům, nepřizpůsobují se uživatelé. Tím odpadají nepříjemné problémy typu zoomování, využívání vodorovného posuvníku apod., které zvyšují nepřehlednost a diskomfort při ovládání internetových stránek.

Jednotná URL adresa, která umožňuje snadné sdílení odkazů mezi uživateli. Odstraňuje problém, který se vyskytuje při existenci separátní mobilní verze webu, kdy uživatel mobilního zařízení prohlíží speciální verzi stránek optimalizovanou pro toto zařízení a pošle odkaz svému známému, který si ovšem bude web prohlížet na stolním PC, ale dostal odkaz pro mobilní verzi.

Jedna URL adresa je výhodnější i z pohledu optimalizace stránek pro internetové vyhledávače (SEO). Při oddělené mobilní a desktopové variantě by bylo SEO oslabeno. Navíc sám Google doporučuje používat responzivní webdesign jako nejlepší přístup pro tvorbu webu optimalizovaného pro mobilní zařízení (Far, 2012; Google Developers, 2014a).

Další výhodou jedné URL adresy je jednodušší centralizované sledování výkonu stránek v jednom grafu či statistice oproti více samostatným verzím webu.

Konkurenční výhoda, která může být způsobená tím, že naši konkurenti nemají stránky optimalizované pro mobilní zařízení a tak je jejich prohlížení na těchto zařízeních méně pohodlné a může uživatele odrazovat od jejich další návštěvy. Pro představu z velmi zajímavé studie Český košík roku vyplynulo, že pouze čtvrtina internetových obchodů má svoje weby optimalizované pro mobilní zařízení, ať už formou separátní mobilní verze nebo responzivního webdesignu. Tato studie zkoumala top 110 e-shopů v České republice, které dle odhadu reprezentují přes 70% celého trhu. Top 100 e-shopů bylo vybráno na základě počtu hodnocení na serveru Heureka.cz. Zbýlých 10 podle počtu lokálních fanoušků na sociální síti Facebook (Český košík roku, 2014; Lichý, 2014).

Automaticky se adaptuje i na nová zařízení, která při vývoji webu ještě ani neexistovala.

2.5 Nevýhody

Dále následují nevýhody hovořící proti responzivnímu webdesignu. Nutno podotknout, že jejich množství je menší oproti výhodám.

Příprava, rozvržení stránek (wireframes), grafický design, kódování a testování je o něco složitější, což způsobuje prodloužení celkové doby vývoje webu. S tím samozřejmě souvisí i větší finanční náklady než kdybychom responzivitě stránek vůbec neřešili nebo vytvářeli samostatnou mobilní verzi webu.

Delší doba načítání stránek oproti separátní mobilní verzi, která může způsobit rychlejší vyčerpání datového limitu uživateli mobilního zařízení (tzv. FUP). Tato nevýhoda se dá zmenšovat různými technikami jako např. důrazem na minimalizaci requestů a objemu dat, podmíněným načítáním, důkladným testováním apod.

Někteří uživatelé, hlavně při první návštěvě responzivních stránek z mobilního zařízení, mohou být zmatení z jiného uspořádání webu oproti vzhledu na desktopu, který dobře znali z dřívější doby. S tím souvisí obvyklá absence přepnutí responzivních stránek do klasického desktopového zobrazení (plné verze). Pokud tedy bude uživatelské rozhraní pro menší obrazovky mobilních zařízení navrženo nevhodně a uživatelé se v něm budou špatně orientovat, nejspíše náš web opustí.

3 Základní prvky

V této kapitole budou popsány tři základní principy, které responzivní webdesign využívá. Tyto tři pilíře byly určeny v již zmiňovaném Marcottově prvním článku o responzivním webdesignu vůbec, tvoří je: plovoucí layout, media queries (v češtině uváděno jako mediální dotazy) a flexibilní obrázky (Marcotte, 2010).

3.1 Plovoucí layout

Prvním krokem směrem k přijetí flexibility webu je používat pro internetové stránky plovoucí layouty tak, aby byly responzivní vzhledem k rozměrům zařízení.

3.1.1 Layout s pevnou šířkou

Ještě než popíšu princip plovoucího layoutu, tak krátce a ve stručnosti zmíním layout s pevnou šířkou (někdy také označován jako fixní), který je dalo by se říci opačným a protichůdným řešením. Jedná se o v současnosti nejběžnější implementaci webových stránek, které neřeší optimalizaci pro mobilní zařízení.

Při jeho použití je šířka webu omezena konkrétní pixelovou hodnotou (absolutní jednotkou), která se za žádných okolností nemění a zůstává zachována. Díky tomu, že víme přesně, s jakou šířkou se budou stránky zobrazovat, můžeme vytvářet graficky propracované a precizní návrhy, které budou na různých obrazovkách vypadat konzistentně. Layout s pevnou šířkou poskytuje největší míru stability a kontroly ze všech druhů layoutů, stránka se vždy zobrazí tak, jak ji designér navrhnul, bez ohledu na zařízení, z kterého ji uživatel právě prohlíží, z čehož vyplývá, že tento druh layoutu je nepřizpůsobivý, což je jeho největší nevýhodou.

Pokud návštěvník internetové stránky, která využívá fixní layout, má šířku okna prohlížeče menší než je šířka webu, uvidí pouze část této stránky a zobrazí se mu nepopulární vodorovný posuvník, což může vést k horší orientaci na stránce. Naopak v případě, že má větší šířku okna prohlížeče než je šířka webu, vzniká po stranách spousta prázdného místa a stránka pak působí jako úzký pruh ve středu obrazovky. Ani jeden z těchto případů není ideální a vedou k menšímu komfortu při ovládání tímto způsobem vytvořených webů.

Nepřaktičnost tohoto typu layoutu se rovněž ukáže při načtení stránky na mobilním zařízení, většina z nich totiž zobrazuje webové stránky tak, aby přesně vyplnily celou obrazovku, tím pádem je uživateli zobrazen zmenšený náhled webu, který si musí přiblížit a zobrazit si pouze jeho konkrétní část (Tranfici, 2013).

3.1.2 Princip plovoucího layoutu

Nyní již následuje popis prvního pilíře responzivního webdesignu, kterým je plovoucí layout, někdy také označován jako fluidní (z anglického fluid) či tekutý (z anglického liquid). Jak už samotný název napovídá, dokáže se přizpůsobovat velikosti zobrazovacího zařízení jako tekutina tvaru nádoby. Jeho základní myšlenkou je, že jednotlivé šířky elementů se udávají v procentech (relativních jednot-

kách), které umožňují obecnou definici rozměrů objektů na stránce. Díky tomu rozvržení webu zachovává poměr jednotlivých prvků, pružně reaguje a přizpůsobuje se různým šířkám zobrazovací plochy, čímž ji optimálněji a efektivněji využívá. Jeho použitím odstraníme problém se zobrazením vodorovného posuvníku na stránce (Nimesh, 2012).

I tento typ layoutu má své nevýhody. Například při zobrazení na zařízení s malou šířkou okna prohlížeče se obsah stránky příliš stlačí a může přetékat z úzkých prvků. Naopak v případě zobrazení na zařízení s velmi velkou šířkou okna prohlížeče se jednotlivé komponenty mohou roztáhnout tak, že stránka bude působit prázdně. Navíc musíme brát v potaz textový obsah, u kterého příliš dlouhé řádky s velkým počtem znaků způsobují horší orientaci v textu. Tento problém se dá zmírnit nastavením maximální šířky (CSS vlastnosti `max-width`) hlavního elementu obsahujícího celou stránku na konkrétní hodnotu zadanou většinou v absolutních jednotkách (nejčastěji pixel), což způsobí, že layout půjde zmenšovat, ale už nepůjde zvětšit právě přes nastavenou hodnotu. V obou zmiňovaných případech může stránka působit nejen špatným vizuálním dojmem, ale může být pro jejího návštěvníka velmi obtížně čitelná (Kadlec, 2014).

Jak je z předchozího odstavce patrné, plovoucí layout sám o sobě není postačující, aby zajistil, že design bude vypadat dobře na všech zařízeních, proto se kombinuje spolu s dalšími základními prvky responzivního webdesignu, které jsou popsány v následujících podkapitolách.

3.1.3 Změna velikosti fontů

Pokud ve svém návrhu internetových stránek akceptujeme flexibilitu webu, nezaobíráme se pouze jeho layoutem, ale znamená to také začít flexibilně měnit velikost fontů (písma).

3.1.3.1 Pixely

Pixely byly dříve velmi preferovanou a oblíbenou jednotkou pro nastavení velikosti písma na stránce. Důvodem je absolutní kontrola nad tím, jak prohlížeč zobrazí velikost textu, pokud ji nastavíme např. na 20px, tak všechny prohlížeče zobrazí tuto velikost precizním způsobem a shodně – jako 20px.

Za tuto úroveň kontroly se ovšem platí a přináší s sebou i několik nevýhod. Údržba takovýchto stránek je poměrně složitá, pokud se rozhodneme celkově zvýšit velikost fontu, budeme muset explicitně změnit každou z hodnot.

Další nepříjemností mohou být potíže týkající se přístupnosti webu. Prohlížeče umožňují uživateli stránku přibližovat (případně i oddalovat) dvěma způsoby. U prvního z nich se funkce přiblížení aplikuje na všechno, co je na stránce, zvětší se tedy všechny její elementy, nikoliv pouze text. Tento způsob dovolí uživateli zvětšovat a zmenšovat stránku, bez ohledu na to, v jakých jednotkách je definována velikost fontu. Při druhé metodě se mění pouze velikost textu, velikost dalších elementů zůstává stále stejná a nemění se. Zde nastává problém u prohlížeče Internet Explorer ve verzích starších jak deset, protože fonty definované v pixelech nejde

tímto způsobem zvětšovat, tím pádem si uživatel nemůže přizpůsobit velikost fon-
tu na stránce. Možností změnit velikost textu dáváme uživateli určitou úroveň kon-
troly a zlepšujeme přístupnost našeho webu (Pešout, 2012).

Definovat velikost fontů v pixelech navíc není příliš přívětivý způsob vzhle-
dem k budoucnosti. Různá zařízení mají odlišné velikosti obrazovek a hustotu pi-
xelů, takže font se zadanou konkrétní velikostí v pixelech, který na jednom zařízení
vypadá dobře, může na jiné obrazovce vypadat příliš malý, nebo naopak příliš vel-
ký. Udávání velikosti fontů v pixelech je jedním z příkladů boje proti flexibilitě
webu. Z tohoto důvodu responzivní webdesign tuto absolutní jednotku nevyužívá
a používá, stejně jako v případě layoutu, relativní jednotky, které jsou popsány
v následujících podkapitolách (Marcotte, 2011).

3.1.3.2 Jednotka em

Flexibilní a stále populárnější způsob jak měnit velikost fontů je použití jednotky
em. Jedno em je rovno aktuální velikosti fonu. Použitím této jednotky vyřešíme
výše zmiňovaný problém s nemožností zvětšit písmo prostřednictvím některých
prohlížečů.

Pokud chceme použít tuto techniku, je nutné nejdříve upravit základní veli-
kost fonu pro celou stránku, rozměr nadefinujeme v procentech. V praxi to nejčas-
těji znamená, že nastavíme velikost písma (CSS vlastnost `font-size`) pro ele-
ment body na 100%, což ve většině moderních prohlížečů odpovídá 16 pixelům.
Výsledkem této úpravy je flexibilní definice rozměru fonu. A právě na tento roz-
měr se odkazujeme pomocí relativní jednotky em, díky které zapíšeme, o kolik bu-
de velikost fonu větší, případně menší než je velikost nadřazeného HTML elemen-
tu (Pešout, 2012).

Jak předchozí věta naznačuje, jednotka em kaskáduje, což může být zrovna tak
žádoucí jako nežádoucí. Například v nepřehledném, nekomentovaném a špatně
strukturovaném kódu se ve velikosti fontů špatně orientuje a jsou nepředvídatel-
né. Vždy musíme brát v úvahu kaskádový efekt, dávat pozor na úroveň zanoření
a mít dobré povědomí o kontextu elementu, abychom věděli, který HTML element
je nadřazený (rodičovský) ve vztahu k tomu našemu, a z kterého se tudíž bude po-
čítat velikost fonu. Tuto komplikaci odstraňuje jednotka zmiňovaná v následující
podkapitole, kterou je rem (Marcotte, 2011).

Výhodou je, že pokud chceme změnit velikost všech fontů na stránce, stačí
upravit výše zmiňovanou základní velikost fonu (CSS vlastnost `font-size`
u elementu body), přičemž zbytek obsahu už se upraví automaticky a zachovají se
vzájemné proporce.

3.1.3.3 Jednotka rem

Další flexibilní možností pro změnu velikosti fontů je jednotka, která byla předsta-
vena v CSS3, je jí rem (akronym z root em). Chová se stejně jako jednotka em,
ovšem s jedním velmi důležitým rozdílem. Její velikost se vztahuje ke kořenovému
elementu (k elementu HTML) a nikoliv k nadřazenému (rodičovskému) elementu,

jak je tomu v případě dříve představené jednotky em. Díky tomu odpadají potíže s kaskádováním, které se vyskytují u vnořených prvků.

Stačí pouze definovat jednu základní velikost fontu pro celou webovou stránku (nastavením CSS vlastnosti `font-size` elementu HTML) a všechny jednotky rem budou odvozeny přímo z této velikosti. Tím pádem nemusíme dbát na to, jak jsou jednotlivé elementy navzájem zanořené.

Toto řešení je velmi elegantní, má značný potenciál a přináší větší přehlednost a předvídatelnost. Má ovšem jednu nevýhodu, kterou je nedostatečná podpora v prohlížečích, hlavně v těch mobilních. Jednotka rem je podporována v těchto prohlížečích: Internet Explorer 9+, Firefox 3.6+, Chrome 6+, Safari 5+ a Opera 11.6+. Podporu poskytují platformy Android 2.1+ a iOS 4+. Nelze ji tedy ještě považovat za univerzální jednotku. Pokud ji však chceme použít a ošetřit i případy prohlížečů, ve kterých není podporována, budeme muset nabídnout nouzovou volbu (tzv. fallback) v pixelech, který má následující podobu:

```
.css_trida {  
    font-size: 16px;  
    font-size: 1rem;  
}
```

Ve výše uvedeném kódu použijí prohlížeče, které podporují jednotku rem, její deklaraci, protože je uvedena jako poslední. Prohlížeče, které jednotku rem nepodporují, použijí první deklaraci nastavenou v pixelech a deklaraci rem, kterou neznají a nerozumí jí, budou ignorovat (Marcotte, 2011; Pešout, 2012).

3.1.3.4 Procenta

Další relativní jednotkou pro změnu velikosti fontů jsou procenta, která zmíním pouze krátce. Chovají se totiž podobným způsobem jako jednotka em, mají měnitelnou velikost prostřednictvím prohlížečů a také kaskádují.

Z teoretického hlediska není valný rozdíl mezi jednotkami em a procenty, ukazuje se však, že vývojáři pro fonty na webu preferují z těchto dvou jednotek spíše em. V podstatě pro to není žádný technický důvod, nicméně je vcelku logické používat jednotku em pro změnu velikosti textu, protože tato jednotka se přímo vztahuje k velikosti textu (Kadlec, 2014).

V případě, že bychom z nějakého důvodu chtěli použít procenta pro změnu velikosti fontů, je dobré dodržovat toto naše rozhodnutí v celém kódu a nemíchat jednotky em a procenta dohromady.

3.1.4 Přepočítání pixelů na relativní jednotky

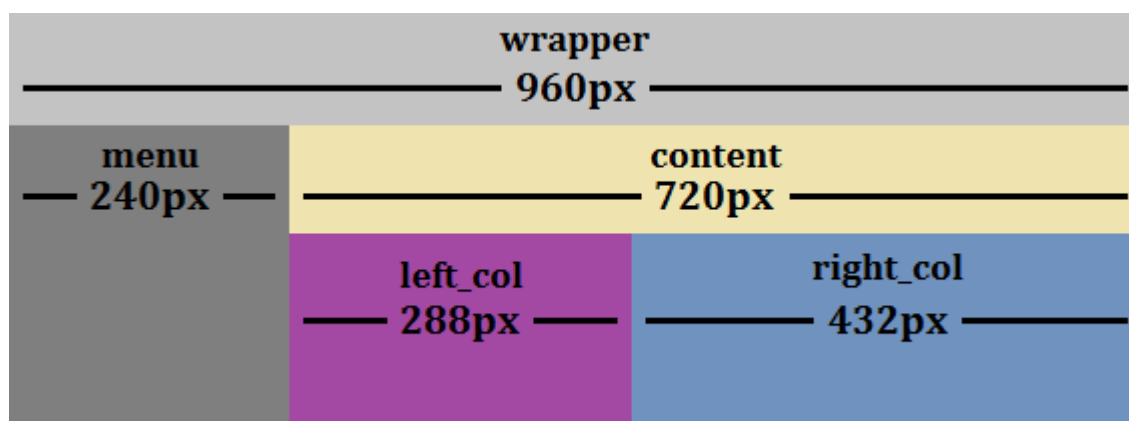
Při návrhu responzivní webové stránky se často setkáme s tím, že budeme pracovat s nějakým již existujícím webem, který používá absolutní jednotky – pixely. Pokud tomu tak nebude a my budeme pracovat úplně zčerstva a od začátku, mnohdy nastane situace, kdy dostaneme od grafika návrh vzhledu webu, který bu-

de vytvořený v grafickém nástroji, který rovněž pracuje s těmito pevnými velikostmi.

Pro vytvoření plovoucího layoutu, který je jedním ze základních prvků responzivního webdesignu, je nutné pixely přepočítat na relativní jednotky, které na rozdíl od těch absolutních, jsou flexibilní. Přepočet se řídí podle jednoduchého vzorce:

$$cíl \div kontext = výsledek \quad (1)$$

Cíl je velikost vnitřního prvku, kterou chceme převést do relativních jednotek. Kontext je velikost vnějšího prvku – nadřazeného (rodičovského) HTML elementu. Výsledek je relativní velikost vnitřního prvku vzhledem k vnějšímu, pokud ho vynásobíme číslem 100, dostane jeho hodnotu v procentech. Pro větší názornost poslouží následující zjednodušená ukázka. Představme si, že jsme od grafika dostali návrh vzhledu webu v pixelech, který je vidět na následujícím obrázku, a chceme jej převést do plovoucího layoutu. Hlavní element wrapper, který obaluje celou stránku, je široký 960px a obsahuje dva elementy – menu (s šířkou 240px) a content (s šířkou 720px), který ještě obsahuje elementy left_col (s šířkou 288px) a right_col (s šířkou 432px).



Obr. 4 Ukázka návrhu vzhledu webu, který pro své rozměry používá pixely

Aplikujeme výše zmiňovaný vzoreček a vypočteme šířku jednotlivých elementů:

- **menu:** $(240/960)*100=25\%$
- **content:** $(720/960)*100=75\%$
- **left_col:** $(288/720)*100=40\%$
- **right_col:** $(432/720)*100=60\%$

Jak je vidět z hodnot dosazených do výpočtů, kontextem pro elementy menu a content je element wrapper, který je jejich nadřazeným (rodičovským) elementem. Elementy left_col a right_col jsou obsaženy v elementu content, který je proto jejich kontextem.

Stejným způsobem bychom převedli z pixelů do relativních jednotek i rozměry ostatních prvků na stránce, jako například velikosti fontů na jednotky em (případně procenta), u kterých bychom museli dávat pozor na úroveň zanoření a co je zrovna u daného prvku jeho kontextem. V případě použití jednotky rem by nám tyto starosti odpadly, kontext by byl vždy stejný a neměnil by se, byla by jím velikost fontu kořenového elementu, kterou bychom si nadefinovali dle potřeby.

Dobrym zvykem z praxe je umísťovat výpočet do CSS komentáře za deklaraci velikosti prvku. Při případném navrácení ke kódu někdy v budoucnosti, budeme rychle vědět, odkud se daná hodnota velikosti vzala a jak jsme k ní došli. Dalším nepsaným pravidlem je uvádět námi vypočítané relativní velikosti prvků v jejich plném tvaru, to znamená ponechat našemu výsledku všechna desetinná místa a nechat zaokrouhlování na prohlížečích, abychom předešli potenciálnímu problému s rozpadnutím layoutu (Marcotte, 2009a).

3.1.5 Mřížkový systém a frameworky

Umísťovat jednotlivé prvky designu v rámci nějaké mřížky je velmi oblíbená praktika, která je mnohem starší než web samotný. Můžeme se s ní setkat například v sazbě tiskovin, dobrým příkladem jsou noviny se svojí typickou strukturou sloupečků. Mřížky pomáhají docílit nejenom vyváženosti správných rozestupů, ale také přehledného uspořádání prvků na stránce, které je příjemnější pro oko. Dobře implementovaný mřížkový systém (někdy také označován jako grid systém) způsobuje, že web vypadá méně chaoticky, je lépe čitelný a lépe se prohledává (Kadlec, 2014).

Mezi hlavní výhody grid systému patří (Vinh, 2010):

- Mřížky dodávají do prezentace informací řád a soulad.
- Mřížky umožňují návštěvníkům stránek lépe se orientovat a odhadnout, kde najít požadovanou informaci.
- S mřížkami se snadněji přidává nový obsah tak, aby byl konzistentní s všeobecnou vizí původní prezentace.
- Mřížky usnadňují spolupráci více lidí na jednom designu. Například pokud se grafik s kóděrem domluví, že se layout bude dělat na nějakou konkrétní mřížku, proces nakódování šablony se tím může výrazně urychlit a zjednodušit. Kódér už nemusí přemýšlet a měřit, jak je nějaký element široký, ale ví, že je navržen na 8 sloupečků stanovené mřížky a tak mu pouze přiřadí odpovídající CSS třídu.

Na internetu jsou dostupné různé frameworky, které jsou založeny na mřížkách. Některé z nich jsou responzivní a přizpůsobují se rozměrům prohlížeče. V určitých hraničních bodech (tzv. breakpointech) dojde k přeskládání prvků na stránce. Mezi nejznámější patří Bootstrap a jeho největší konkurent Foundation. V případě těchto dvou se jedná o komplexní frontend frameworky, které usnadňují práci s typografií, tvorbu layoutu, vytváření elementů uživatelského rozhraní a zároveň ošetřují zobrazování napříč platformami. Najdeme v nich styly pro tlačítka, tabulky,

formulářové prvky, menu apod. Nechybí ani JavaScript, jako například karusely, tooltips, záložky, alerty apod. Foundation dokonce nedávno uvolnil i verzi specializovanou na tvorbu responzivních webových aplikací. Je patrné, že tyto dva frameworky jsou opravdu robustní a pokrývají celou řadu prvků na webu, daní za to je jejich poněkud větší velikost. U obou dvou ale platí, že si je můžeme stáhnout v upravené podobě, dají se z nich tedy vytáhnout pouze prvky, které chceme na našem webu používat (např. pouze zmiňovaný grid systém) a tím ušetřit datovou velikost (Polzer, 2013).

Použití frameworků je snadné. Na všechny předdefinované a předem připravené prvky existuje konkrétní CSS třída, kterou stačí přiřadit v HTML kódu nějakému určitému elementu, což zajistí, že se bude chovat a vypadat tak, jak je v daném frameworku definováno.

Pokud by nám nevyhovoval žádný grid systém z dostupných frameworků, můžeme si definovat svůj vlastní. S tím nám mohou pomoci generátory, které jsou k dispozici na internetu. Podle námi zvolených parametrů vygenerují CSS kód, který tak nemusíme psát, součástí je i obrázek s vzhledem mřížky. Jako příklad může posloužit www.gridssystemgenerator.com nebo www.gridpak.com.

V případě, že se rozhodneme využít mřížkového systému, neměli bychom se vždy předem upínat na nějaký konkrétní, například již hotový grid, který by nás mohl svazovat. Měli bychom uvažovat spíše dle myšlenky, pro kterou se vžil termín content-out – zaměříme se na obsah webové prezentace (texty, obrázky, videa, reklamy, apod.) a nechme jej, ať sám sobě definuje mřížku, která se pro něj bude nejlépe hodit. Místo toho, abychom se snažili obsah vměstnat do nějaké předem připravené mřížky, vybudujme si svoji, která z obsahu samotného bude vycházet (Ford, 2014).

Grid systém rozhodně není žádné dogma, kterého by se bylo nutné držet při vytváření všech webových stránek. Je ovšem dobré vědět, že existuje, včetně dostupných frameworků a generátorů, a zvážit jeho použití v konkrétních případech. Může nám totiž být velkým pomocníkem a urychlit a zefektivnit práci na našem webu (Cejnek, 2014).

3.2 Media queries

Media queries (v češtině nejčastěji uváděno jako mediální dotazy) byly přidány ve specifikaci CSS3 a prakticky až jejich nástup umožnil vznik a rozšiřování responzivního webdesignu. Dávají nám možnost dotazovat se prohlížeče, zda je nějaký výraz pravdivý a pokud ano, můžeme načíst specifický blok kaskádových stylů. S jejich pomocí lze přizpůsobovat vzhled stránek v závislosti na typu a vlastnostech zobrazovacího zařízení.

Media queries mají čtyři základní komponenty (Kadlec, 2014):

- **Media types** – určujeme jimi typ zařízení, na který se chceme zaměřit.
- **Media features** – s jejich pomocí testujeme nějaké vlastnosti zařízení.
- **Logická klíčová slova** – umožňují vytvářet složitější dotazy.

- **Pravidla** – kaskádové styly, jimiž upravujeme zobrazení stránky při splnění daného výrazu.

3.2.1 Media types

Media types byly definovány už ve specifikaci CSS2 a existují tak již daleko déle než media queries. S pomocí media types můžeme rozlišovat typ zařízení, na kterém je stránka právě zobrazována. Díky tomu můžeme definovat, jakému typu zařízení budou přiřazeny vybrané kaskádové styly.

Standard definuje celkem deset různých typů zařízení, jsou to: all, braille, embossed, handheld, print, projection, screen, speech, tty a tv (W3C, 2011b). V praxi se ale téměř výhradně používají pouze all (pro všechny typy zařízení), screen (pro barevnou počítačovou obrazovku) a print (pro tisk nebo náhled před tiskem), které se dočkaly reálného rozšíření. Je tomu tak kvůli dlouhotrvající tendenci vývojářů používat výhradně screen, místo aby uvedli například handheld (pro mobilní zařízení, typicky s malou obrazovkou a limitovaným internetovým připojením) nebo tv (pro televizní zařízení). Proto se většina zařízení a prohlížečů rozhodla, že bude podporovat pouze screen místo svého specifického media type, aby webové stránky vypadaly podobně jako na osobních počítačích. Z tohoto důvodu je nevhodné pro odlišení vzhledu stránky na mobilních zařízeních používat pouze media type (Škrásek, 2013). Lepším řešením je detekce s využitím řetězce User-Agent v hlavice HTTP požadavku, která byla již dříve zmiňována v podkapitole zabývající se separátní mobilní verzí webu.

Přímo v CSS kódu může dotaz na media type vypadat takto:

```
@media print {  
    pravidla  
}
```

Pokud máme CSS kód pro konkrétní typ zařízení v externím souboru, můžeme dotaz na media type použít v tagu `link` a jeho atributu `media`:

```
<link rel="stylesheet" href="print.css" media="print" />
```

V obou výše uvedených ukázkách kódu se příslušný stylopis pro tisk bude aplikovat jen tehdy, pokud budeme chtít vytisknout danou webovou stránku nebo se na ni budeme dívat v režimu náhledu před tiskem.

V případě, že chceme zacílit na více typů zařízení, oddělíme je od sebe v dotazu čárkou, která má význam logické spojky `or`.

3.2.2 Media features

Media features jsou součástí CSS3. Rozšiřují výše zmiňované a starší media types, které jsou příliš obecné a nedostatečné pro konkrétní přizpůsobení stránky, jelikož nevypovídají nic o specifických vlastnostech zařízení. Proto vznikly media features, které umožňují testovat zařízení, které aktuálně prohlíží naši webovou stránku, zda má určitou schopnost či vlastnost.

Standard CSS3 definuje celkem třináct různých charakteristik, na které můžeme testovat zobrazovací zařízení (W3C, 2012). V praxi jsou nejpoužívanějšími vlastnostmi `width` (šířka zobrazovací oblasti, tedy okna prohlížeče), `height` (výška zobrazovací oblasti), `orientation` (orientace zařízení – na výšku nebo na šířku), `resolution` (rozlišení zařízení ve smyslu hustoty pixelů) a `aspect-ratio` (poměr šířky a výšky zobrazovací oblasti). Hodnoty jednotlivých vlastností bývají přesně zadané nebo případně je u většiny z nich možnost použití prefixů `min` a `max`.

Pro zápis dotazu na media features platí stejné možnosti jako v případě media types, tedy můžeme ho zapsat přímo do CSS kódu nebo do tagu `link` a jeho atributu `media`.

Media feature se uvádí v kulatých závorkách a v případě, že chceme otestovat zařízení na více vlastností, zapíšeme jednotlivé media features do závorek a pro jejich spojení použijeme klíčové slovo `and`, reprezentující význam logické spojky `and`.

3.2.3 Logická klíčová slova

Při sestavování media queries můžeme použít logická klíčová slova, díky kterým lze vytvářet složitější dotazy. K dispozici jsou následující (W3C, 2012):

- `and` – umožňuje testovat na současnou platnost několika výrazů media features. Zároveň slouží pro spojení dvou částí mediálního dotazu – media types a media features.
- `not` – neguje výsledek celého media query. Nemůžeme ho použít k negaci jednoho jediného testu, musí být uvedeno na začátku dotazu a předcházet jeho zbytku, čímž neguje celý za ním následující výraz.
- `or` – logickou spojku `or` zastupuje symbol čárky. Umožňuje kombinovat několik media queries oddělených právě čárkou. V případě, že je alespoň jeden z uvedených mediálních dotazů pravdivý, vyhodnotí se jako pravdivý i celý dotaz.
- `only` – skryje mediální dotazy před prohlížeči, které je nepodporují. Prohlížeče, které media queries podporují, budou toto klíčové slovo ignorovat a zpracují dotaz normálním způsobem.

3.2.4 Pravidla

Poslední ze čtyř komponent media queries jsou samotná stylová pravidla, která chceme aplikovat v případě pravdivého vyhodnocení mediálního dotazu. Jedná se o klasický CSS kód, který umístíme dovnitř samotného dotazu.

3.2.5 Sestavení mediálního dotazu

Po popsání všech čtyřech komponent, z kterých se skládají media queries, už můžeme sestavit výsledný mediální dotaz. Ten má následující obecnou podobu:

```
@media [not|only] typ [and] (výraz) {  
    pravidla  
}
```

Jak je vidět z výše uvedeného kódu, media queries kombinují media types (uváděné v části `typ`) a media features (uváděné v části `výraz`), které v dotazu spojíme za pomoci logického klíčového slova `and`. Každý mediální dotaz musí obsahovat media type, pokud žádný neuvedeme, vloží se automaticky `all`, které je výchozí hodnotou.

Opět máme dvě možnosti kam zapsat media queries. Stejně jako v případě výše zmiňovaných media types a media features je můžeme uvádět přímo v CSS kódu nebo v tagu `link` a jeho atributu `media` při připojování CSS souboru k HTML kódu.

Příklad mediálního dotazu zapsaného přímo v CSS kódu:

```
@media all and (min-width: 768px) {  
    pravidla  
}
```

Výše uvedený mediální dotaz testuje šířku zobrazovací oblasti, prefix `min` udává, že má být minimálně 768 pixelů. Pokud je tato podmínka splněna (kladně vyhodnocena), aplikují se `pravidla` (CSS kód) uvedený uvnitř mediálního dotazu, a to nezávislé na typu zobrazovacího zařízení, díky uvedení media type `all`.

3.2.6 Podpora v prohlížečích

Media queries podporují téměř všechny mobilní a desktopové prohlížeče. Komplikací je pouze Internet Explorer ve verzích starších než devět, které mediální dotazy bohužel nepodporují (Can I use, 2014). To může být problémem v případě mobile first přístupu, kdy staré verze Internet Exploreru nedokážou načíst styly určené pro desktop, ukryté právě v mediálních dotazech, a tím pádem se stránka zobrazí v nejjednodušší verzi rozložení určené pro zařízení s malou obrazovkou, tedy pro mobilní telefony.

Tento problém se dá vyřešit použitím podmíněných komentářů, díky kterým starým verzím Internet Exploreru můžeme přiřadit jiný stylpis. Dalším možným způsobem řešení je využití JavaScriptové knihovny, například Respond.js. Jedná se o tzv. polyfill, který umožní použití media queries i v těch prohlížečích, které je nepodporují. Konkrétně zmiňovaný Respond.js zprovozňuje pouze vlastnosti `min-width` a `max-width`, ty jsou ovšem nejpoužívanější a v drtivé většině případů postačují (Jehl, 2011).

3.3 Flexibilní obrázky

Třetím a posledním základním stavebním kamenem responzivního webdesignu jsou obrázky, které rovněž musíme udělat flexibilními, aby se přizpůsobovaly velikosti zobrazovacího zařízení. Jejich problémem je, že mají pevně stanovenou velikost, kterou si zachovávají při zobrazení na všech zařízeních, včetně těch nejmenších mobilních telefonů, pro které jsou ale moc velké.

Obrázky se dají poměrně snadno učinit flexibilními tím, že jim v CSS kódu nastavíme vlastnost `max-width` na hodnotu `100%`:

```
img {  
    max-width: 100%;  
}
```

Obrázky se potom přizpůsobují velikosti obalového elementu (kontejneru), v němž jsou umístěny s tím, že se zobrazí maximálně ve své původní plné velikosti. Během zmenšování si obrázek zachovává své původní proporce (poměr stran), tudíž nedochází k jeho deformaci. V souvislosti s tím se někdy do výše uvedeného kódu přidává nastavení CSS vlastnosti `height` na hodnotu `auto`. To ale není nezbytně nutné, jelikož k zachování proporcí dochází automaticky i bez uvedení této deklarace. Nutno dodat, že v HTML kódu nezadáváme šířku a výšku obrázku uvnitř tagu `img` v jeho attributech `width` a `height` (Marcotte, 2009b).

3.4 Meta tag viewport

Při tvorbě responzivní webové stránky je pro její správnou funkčnost důležitou součástí použití meta tagu `viewport`. Zde je potřeba vysvětlit význam slova `viewport` (do češtiny nejčastěji překládáno jako zobrazovaná oblast). Z hlediska desktopových prohlížečů představuje šířku prohlížeče (viditelnou oblast prohlížeče). V případě mobilních zařízení je tomu ale jinak. I přesto, že mají obrazovky rozměrově mnohem menší, snaží se zobrazit webové stránky v kompletní podobě za účelem poskytnutí úplného webového prožitku, což sebou přináší jisté komplikace.

Je nutno rozlišovat dva různé viewporty, konkrétně `viewport layout` a vizuální `viewport`. `Viewport layout` má pořád stejné míry, které se nemění, odkazuje na celou stránku. Oproti tomu vizuální `viewport` je ta část stránky, kterou právě vidíme na obrazovce a její velikost se může měnit například při přiblížení či oddálení stránky. Na desktopu oba pojmy odkazují na stejnou oblast, ale u mobilních zařízení a jejich prohlížečů tomu tak ve většině případů není. Ty totiž pro `viewport layout` nevrací svoji skutečnou velikost šířky rozlišení v pixelech, ale záměrně vrací vysoké rozměry a tváří se tak, že mají obrazovku širší, než tomu je ve skutečnosti. Například prohlížeč `mobile Safari` vrací `viewport layout` o šířce 980 pixelů, `Opera Mobile` 850 pixelů, `Android WebKit` 800 pixelů a `Internet Explorer` v mobilních zařízeních s operačním systémem `Windows Phone` vrací 974 pixelů (Kadlec, 2014).

Z výše zmiňovaného důvodu, že obrazovka mobilních zařízení je obvykle menší než okno prohlížeče, dochází k typickému stavu zobrazení webových stránek v těchto zařízeních. To většinou vypadá tak, že daný web se zobrazí na celou šířku obrazovky (je vidět kompletně celý), přičemž je ale celý oddálený, a tím pádem všechen jeho obsah zmenšený, což způsobuje špatnou čitelnost. Uživatel si musí přiblížit konkrétní část stránky, která ho zajímá.

Toto chování je ovšem v případě tvorby responzivních stránek nežádoucí. Naštěstí existuje možnost, jak se s něčím takovým vypořádat, a to právě díky meta tagu `viewport`, který umožňuje řídit změnu měřítka a `viewport` layoutu. Jeho formát je jednoduchý – v atributu `name` uvedeme, že se jedná o meta tag `viewport` a v atributu `content` vypíšeme seznam direktiv, které jsou odděleny čárkou:

```
<meta name="viewport" content="direktiva,direktiva" />
```

Umísťuje se do hlavičky (elementu `head`) HTML kódu. V jeho části `content` můžeme použít následující direktivy (W3C, 2011a):

- `width` – umožňuje nastavit `viewport` na konkrétní šířku (velikost v pixelech bez jednotky) nebo šířku zařízení. Kdybychom použili konkrétní šířku, která by neodpovídala šířce daného zařízení, změnilo by se odpovídajícím způsobem měřítko (stránky by byly oddálené, případně přiblížené). Můžeme použít proměnnou `device-width`, která zajistí, že velikost `viewportu` layoutu se vždy bude rovnat velikosti šířky obrazovky zařízení (v pixelech), ze kterého je daný web prohlížen.
- `height` – protějšek `width`, umožňuje specifikovat konkrétní výšku `viewportu`, v praxi se moc nepoužívá. Obdobně jako u `width` lze její hodnotu přizpůsobit výšce obrazovky zařízení pomocí proměnné `device-height`.
- `user-scalable` – říká prohlížeči, zda uživatel bude moci stránku přibližovat (zvětšovat) a oddalovat (zmenšovat). Pokud tuto direktivu vůbec neuvedeme, má výchozí hodnotu `yes` (web bude možné přibližovat a oddalovat). Některé stránky ji mají nastavenou na hodnotu `no` – typicky se tím zajišťuje tzv. na pixel perfektní (pixel perfect) zobrazení designu. Není to ale nejlepší řešení, omezuje přístupnost stránek a bere uživateli určitou možnost kontroly, což může být problémem pro návštěvníky se zrakovým postižením.
- `initial-scale` – nastavuje počáteční úroveň přiblížení stránky v rozsahu od 0.1 (10%) do 10 (1000%)
- `maximum-scale` – říká prohlížeči, jak moc může uživatel na stránce přibližovat (zvětšovat), rozsah je stejný jako v případě `initial-scale`. Při nastavení na hodnotu 1 (100%), uživatel nebude schopen přiblížit stránku.
- `minimum-scale` – říká prohlížeči, jak moc může uživatel stránku oddalovat (zmenšovat), opět stejný rozsah jako v případě `initial-scale`. Při nastavení na hodnotu 1 (100%), uživatel nebude schopen stránku oddalovat.

V praxi je nejběžnějším a doporučovaným způsobem použití meta tagu viewport pro responzivní weby v následující podobě (Google Developers, 2014b):

```
<meta name="viewport" content="width=device-width,initial-scale=1" />
```

Zbavíme se jím klasického oddáleného (zmenšeného) vzhledu stránky na mobilních zařízeních a jako omezení využijeme samotnou šířku obrazovky aktuálního zařízení. Pokud načteme stránku s výše zmíněnou podobou meta tagu viewport na nějakém mobilním zařízení, bude se zobrazovat úplně stejně, jako kdybychom ji prohlíželi na desktopu s malou šířkou okna prohlížeče.

Do budoucna se počítá s nahrazením meta tagu viewport syntaxí @viewport přímo v CSS kódu (W3C, 2011a).

4 Použité technologie

Pro vytvoření responzivních webových stránek pro firmu Návrh loga byly použity technologie HTML, CSS a JavaScript, které jsou stručně popsány v následujících podkapitolách.

4.1 HTML

Jedná se o značkovací jazyk, který se používá k tvorbě webových stránek, konkrétně slouží k vytvoření jejich struktury. Odděluje obsah webu od jeho prezentace (způsobu jakým má být zobrazen). Soustředí se pouze na popis obsahu samotného, na jeho sémantiku (význam).

HTML je zkratkou z anglického HyperText Markup Language. Slovo HyperText znamená možnost navzájem propojovat texty za pomoci odkazů. Markup vyjadřuje schopnost dávat význam jednotlivým částem dokumentu na základě použití pevně dané sady značek, které se nazývají tagy. Jednotlivé tagy jsou uzavřeny do špičatých závorek a dělí se na nepárové a párové, kterých je většina a musí být ukončeny koncovou značkou. Tagy mohou navíc obsahovat atributy, kterými se dá dále určovat, jak by se daný element měl chovat či zobrazit. Atributy jsou tvořeny svým názvem a většina z nich ještě navíc hodnotou, která je uzavřena v uvozovkách a od názvu oddělena rovnítkem (Mozilla Developer Network, 2014b).

4.2 CSS

CSS je zkratkou z anglického Cascading Style Sheets, česky kaskádové styly. Cílem vzniku kaskádových stylů bylo oddělení vzhledu webové stránky od jejího významu. Proto je CSS jazykem, který popisuje právě vzhled prvků stránky, na rozdíl od výše zmiňovaného HTML, které popisuje význam obsahu.

Základem CSS je tzv. pravidlo stylu, které říká, jak bude prvek, jemuž je styl určen, vypadat. Skládá se ze dvou částí, a to selektoru (případně více selektorů) a deklarace jedné nebo více vlastností. Selektor určuje, který HTML element bude vybrán k přiřazení daného stylu. Deklarace vlastnosti spočívá v přiřazení jedné (někdy i více) hodnot vlastnosti.

Kaskádové styly musíme do HTML dokumentu nějakým způsobem vložit. V praxi je nejpoužívanější cestou vytvoření samostatného externího CSS souboru, který pak s HTML dokumentem propojíme za pomoci tagu `link`, který se uvádí v hlavičce HTML dokumentu. Dalšími způsobem je řádkový styl, což znamená zápis deklarace vlastností přímo v atributu `style` u daného HTML elementu. Poslední a třetí možností je vložená šablona stylů. V jejím případě se CSS pravidla zapíší do elementu `style` v hlavičce HTML dokumentu (Domes, 2011).

4.3 JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk. Používá se pro tvorbu dynamických webových stránek a velmi často pro zvýšení interaktivity a atraktivity webu.

Skript napsaný v JavaScriptu se dá do HTML dokumentu vložit třemi způsoby velmi podobnými jako u výše zmiňovaného CSS. To znamená vytvoření samostatného externího souboru, který propojíme s naším HTML dokumentem za pomoci párového tagu `script`, v jehož atributu `src` uvedeme cestu k našemu skriptu. Další způsob spočívá v zápisu samotného kódu skriptu do párového tagu `script` přímo v HTML dokumentu. Třetí možností a v praxi nejméně používanou je tzv. inline zápis, který na rozdíl od předchozích dvou způsobů nevyužívá tag `script`, ale kód skriptu se zapisuje přímo do HTML tagu a jeho atributu, který reprezentuje nějakou událost (Mozilla Developer Network, 2014a).

5 Použité nástroje

Při tvorbě responzivních webových stránek pro firmu Návrh loga jsem použil nástroje, které jsou uvedeny a krátce představeny v následujících podkapitolách.

5.1 Sublime Text

Pro psaní veškerého kódu jsem použil multiplatformní textový editor zdrojových kódů Sublime Text 3. Hlavním důvodem bylo to, že jsem na něj zvyklý již z dřívější doby a plně mi vyhovuje.

Sublime Text je velmi výkonný a rychlý. Podporuje spoustu programovacích jazyků a je schopen zvýraznit jejich syntaxi. Disponuje napovídáním, které velmi výrazně zjednodušuje a urychluje psaní kódu. Má přehledné, jednoduché a přívětivé uživatelské rozhraní. K dispozici je navíc velké množství alternativních barevných schémat, případně je možné vytvořit si své vlastní. Jeho funkcionality je rozšiřitelná dostupnými pluginy, které se snadno a rychle instalují přes tzv. Package Control a díky početné komunitě je jejich množství velké. Má různé zobrazovací módy a možnosti rozdělení pracovní plochy. Samozřejmostí je hledání, včetně regulárních výrazů s typickou funkcí najít a nahradit v celém kódu, navíc text odpovídající regulárnímu výrazu se okamžitě zvýrazňuje. Další unikátní funkci je náhled souboru (minimapa kódu), což je zmenšený celý zdrojový kód na pravé straně editoru, který zlepšuje orientaci v dlouhých kódech. Sublime Text si pamatuje otevřené soubory a dokonce i pozici kurzoru či označený text v každém otevřeném souboru, tím pádem zavření editoru není žádný problém a při jeho příštím spuštění se otevře přesně ve stejném stavu, jako byl zavřen. Pro mě jednou z nejlepších a nejčastěji používaných funkcí je tzv. vícenásobný kurzor, kdy editor umožňuje vytvořit si v kódu libovolné množství kurzorů, díky tomu je možné psát na více míst zároveň.

5.2 Adobe Photoshop

Pro všechnu práci související s grafickým návrhem stránek jsem použil program Adobe Photoshop, konkrétně verzi 10.0 (CS3). Kompletní návrh celého webu byl připraven grafikem firmy Návrh loga ve formátu PSD, takže zde žádná jiná možnost volby v podstatě ani nebyla.

Adobe Photoshop je kvalitní a profesionální bitmapový grafický editor, který se postupem času stal světovým standardem pro tvorbu a editaci rastrové grafiky. Je ideálním nástrojem nejen pro úpravu fotografií, ale v praxi se velmi často používá právě i pro tvorbu a práci s webovou grafikou.

5.3 Podio

Firma Návrh loga používá platformu Podio, která slouží pro týmovou spolupráci a projektový management. Jedná se o jeden ze systémů pro řízení projektů a fungování firmy online.

S pomocí této webové aplikace lze snadno řídit projekty, komunikovat s kolegy či zákazníky, sdílet informace, zprovoznit firemní intranet apod. Mezi největší výhody Podia patří jeho flexibilita. Princip této služby totiž spočívá ve vytvoření si tzv. workspace, což je prostor pro náš projekt, do kterého si následně vkládáme a kombinujeme aplikace, které nejlépe vyhovují našim potřebám. K tomu slouží Podio App Market, ve kterém je dostupné mnoho aplikací, které jsou rozděleny do různých kategorií jako např. Class Work & Education, Event Management, Human Resources, Intranet, Legal & Finance, Management, Marketing, Meetings & Conferencing, PR & Communication, Project Management apod. V případě potřeby lze dokonce vytvořit si i vlastní aplikaci naklonováním již nějaké existující aplikace a její úpravy přesně na míru našim požadavkům.

Mezi hlavní funkce této služby patří úkoly (jejich přidělování a sledování plnění), kalendář s mnoha funkcemi, adresář všech kontaktů na jednom místě, sledování nového obsahu na jednom místě v reálném čase, snadná komunikace v týmu apod.

Podio jde propojit s některými dalšími populárními službami, jako jsou např. Google Apps (Google Drive, Google Calendar), DropBox, Microsoft OneDrive apod. Navíc je dostupné i jako mobilní aplikace, momentálně jsou podporovány operační systémy Android a iOS.

6 Metodika

Cílem praktické části je vytvoření responzivních webových stránek pro firmu Návrh loga, konkrétně se jedná o její webovou prezentaci. Z pohledu praxe jsem byl v pracovní pozici, která se označuje jako kodér webových stránek, někdy také jako HTML kodér, HTML/CSS kodér či moderněji Frontend Web Developer. Hlavním úkolem takového kodéra je převést obdržený grafický návrh do podoby webové stránky za pomoci technologií HTML a CSS, eventuálně i JavaScriptu. V mém případě ještě navíc hraje velmi důležitou roli, aby stránky byly responzivní, což v praxi zatím není zcela běžným požadavkem, i když se stává postupně stále častějším a u zaměstnavatelů v oboru čím dál více žádanějším.

6.1 Obdržený grafický návrh od firmy Návrh loga

Od grafika z firmy Návrh loga jsem dostal grafický návrh zpracovaný v bitmapovém grafickém editoru Adobe Photoshop. Šlo o jediný soubor ve formátu PSD, který obsahoval kompletní návrh vzhledu a obsahu celého webu a jeho jednotlivých částí. Každá ze sekcí webu byla od grafika zpracována ve své vlastní tzv. vrstvě, což mi umožnilo příjemnou a bezproblémovou práci s navrženou grafikou.

Ke grafice mi firma dodala i PDF dokument s názvem „Metodika pro kodéry/grafiky – sladění barevného profilu ve Photoshopu“, který obsahoval pět základních bodů. Byly jimi nastavení zobrazení barevného profilu, nastavení barevného profilu dokumentu, nastavení zobrazení na monitoru, otevření nového dokumentu a ukládání obrázků. Smyslem tohoto dokumentu je správné nastavení a synchronizace toho, jak se zobrazí barvy, Photoshop totiž pracuje s různými barevnými profily. Postup popsany v tomto dokumentu zaručí kompatibilitu stejné barevnosti výstupů grafika a kodéra během procesu tvorby webových stránek.

Co se týče samotného grafického návrhu, tak bych zde rád poznamenal, že se jedná o grafický návrh, který je téměř rok starý a připravený k použití. Vzhledem k tomu, že se nejedná o klientskou zakázku, která v praxi většinou nejvíce spěchá a má největší prioritu, ale naopak o vlastní prezentační web firmy, tak se tvorba webu dle tohoto návrhu stále odkládala na pozdější dobu. Nakonec byl návrh přiřazen mně, jakožto vhodný z toho hlediska, že nespěchal a tudíž byl příhodný i pro mě, jakožto začínajícího kodéra, jako moje první setkání s reálným procesem tvorby webových stránek přímo v praxi.

Rovněž je důležité zmínit, že tento návrh nepočítal s responzivním chováním stránky. Bylo tomu přesně naopak a původním záměrem byl fixní layout a tzv. pixel perfect zobrazení designu. Počítalo se s tím, že spousta věcí bude řešena pomocí obrázků a jejich pevně daných rozměrů, což není vhodné v případě použití responzivního webdesignu. Z tohoto důvodu bylo potřeba zamyslet se nad jednotlivými prvky stránky a oproti původnímu grafickému návrhu u některých z nich pozměnit jejich vzhled nebo je z webu úplně vypustit. Ostatně nad finální podobou některých sekcí webu ještě probíhá s firmou diskuze a jejich vzhled se bude upřesňovat.

6.1.1 Řezání grafiky ve Photoshopu

Nařezání grafiky z dodaného grafického návrhu je jedním z klíčových úkolů kodéra a nejinak tomu bylo i v mém případě. Šlo o to, vytáhnout si z PSD dokumentu všechny potřebné prvky (obrázky a textový obsah) pro převod samotného návrhu na webovou stránku.

V první části práce s PSD dokumentem jsem se zaměřil na export obrázků. Ty je nutné z grafického návrhu nejprve vyříznout. K tomu slouží v programu Adobe Photoshop nástroj Řez. S jeho pomocí je možné rozřezat návrh na obdélníkové oblasti, které se následně dají uložit jako samostatné soubory různých grafických formátů. Nástroj Řez se skládá ze dvou ikoněk reprezentujících dva nástroje, na které se Řez ještě dělí, jsou jimi Rozřezání a Výběr řezu. Rozřezání slouží pro tvorbu samotného řezu a Výběr řezu je určen pro označení a modifikaci řezu. Postupně jsem si tedy nařezal všechny potřebné obrázky, které jsem pak použil při kódování webu. Jednotlivé řezy jsem si pro větší přehlednost pojmenoval. Photoshop kromě mnou (jako uživatelem) vytvořených řezů, které zvýrazňuje modrou barvou, vytvořil ještě navíc zbytkové řezy, které zvýrazňuje šedou barvou, ty byly ale nepotřebné pro použití na webu, proto jsem si jich nevšímal a ignoroval je. Při řezání jednotlivých prvků jsem hojně využíval Správce vrstev, ve kterém jsem si skrýval vrstvy s nepotřebným obsahem, což mi umožnilo vyříznout si pouze a jen ten prvek, který jsem potřeboval. Při práci s vrstvami jsem často využíval možnost označení více vrstev (těch nepotřebných), které jsem sloučil do jedné jediné vrstvy, kterou jsem skryl. Velmi užitečnou funkcí byla také Historie, která umožňovala vrátit se o několik úkonů zpět, což se hodilo např. právě při sloučení více nepotřebných vrstev do jedné a jejich skrytí, kdy jsem si následně vyřízнул, co jsem potřeboval a poté přes Historii všechno vrátil zpět do původního stavu. Po nařezání všech potřebných obrázků už zbývalo pouze jejich vyexportování z PSD dokumentu. K tomu jsem použil funkci Uložit pro web. Ta umožňuje vyexportovat všechny řezy najednou nebo jen námi vybrané. Po výběru řezů, které chceme uložit, nastavíme už jen typ námi požadovaného výstupního grafického formátu, složku do které chceme obrázky uložit a jako poslední, které řezy chceme vyexportovat, kde můžeme nastavit vybrané řezy (které jsme si před tím zvolili), všechny uživatelské řezy (všechny řezy, které jsme vytvořili – ty zvýrazněné modrou barvou) nebo všechny řezy (včetně zbytkových automaticky vytvořených Photoshopem – těch zvýrazněných šedou barvou).

Druhá část práce s PSD dokumentem souvisela s veškerým textovým obsahem, který se měl na webu zobrazit. Spočívala ve zkopírování všech textů jednotlivých sekcí webu a jejich vložení do vytvořených HTML dokumentů. Samozřejmě jsem musel dodržovat strukturu a formátování textu, včetně použitého fontu, řezu písma a jeho barvy.

6.2 Kódování responzivních webových stránek pro firmu Návrh loga

Po tom, co jsem si nařezal grafiku z obdrženého grafického návrhu, vyexportoval si všechny potřebné obrázky a vytáhnul si všechny textový obsah, následovalo již samotné kódování webu v HTML a CSS.

Před začátkem psaní kódu jsem si ještě pečlivě prošel všechny sekce webu v grafickém návrhu a zamyslel se nad responzivním chováním každé jednotlivé části webu. Vzhledem k tomu, že grafický návrh, jak již bylo zmiňováno, původně nepočítal s responzivním provedením stránky, nebylo zcela přesně a do detailu zadáno responzivní chování a to, jak mají všechny jednotlivé části webu na menších zařízeních vypadat. Proto jsem si vždy před kódováním všech sekcí webové stránky nejdřív rozmyslel, jakým způsobem se daná sekce bude zobrazovat na malých obrazovkách a v čem přesně bude spočívat její responzivita.

Firma Návrh loga mi k pracím souvisejícím s kódováním webu dodala PDF dokument s názvem „Kodérské desatero“, jehož smyslem je upozornit kodéry na některé základní a opakované chyby, které se často vyskytují u výstupních kontrol projektů. Dokument hovoří o tom, že i výsledek nakódování webu by měl odpovídat nějakým elementárním grafickým pravidlům, především by měl být za všech okolností přehledný. Jeho obsahem jsou mimo jiné body týkající se typografie, zarovnání a rozmístění prvků na stránce, vzhledu a úpravy webových formulářů či popisu různých grafických formátů obrázků a jejich použití.

6.2.1 Přístupy desktop first a mobile first

Než začneme kódovat responzivní web, musíme se rozmyslet, zdali ho budeme budovat směrem z desktopu dolů nebo z mobilu směrem nahoru. Jedná se o dva rozdílné přístupy k tomu, jak navrhovat a kódovat responzivní webové stránky. Tyto přístupy se odlišují tím, která verze webu se bude navrhovat a kódovat jako první. Postupu z desktopu dolů se říká desktop first a postupu z mobilu směrem nahoru mobile first.

Princip metody desktop first spočívá v tom, že web se primárně navrhuje pro desktop (velké obrazovky) a teprve následně se propracovává k zařízením s menší obrazovkou (od tabletů až po mobilní telefony). Výchozím layoutem je tedy to, co typicky vidíme na obrazovce v prohlížeči na stolním počítači či notebooku. Při kódování kaskádových stylů se začíná těmi pro desktop, které se uvádí bez použití media queries. Teprve potom se prostřednictvím media queries, typicky s použitím dotazu na vlastnost `max-width`, uvádí kód, který původní desktopový layout zjednodušuje a přizpůsobuje ho menším obrazovkám. Při optimalizaci webových stránek pro mobilní zařízení není tento přístup příliš vhodným, protože CSS kód určen pro menší displeje je zpravidla jednodušší a vlastně tak popírá ten původní složitější pro desktopy v tom smyslu, že ruší různé věci a v zásadě se tak kód komplikuje a stává se méně přehledným. Filozofie desktop first neklade příliš velký důraz na mobilní zařízení. Spíše se snaží obsah primárně určen desktopům poskládat nějakým způsobem tak, aby byl čitelným a použitelným i na menších obrazov-

kách. Tato metoda byla hojně používána hlavně v době, kdy ještě mobilní zařízení nebyly tak dramaticky rozšířené a přístup z nich na webové stránky nebyl tak velký, jako je tomu nyní (Kadlec, 2014).

Mobile first je modernějším a pro responzivní webdesign vhodnějším přístupem, jehož autorem je Luke Wroblewski. Na rozdíl od desktop first je zde největší důraz kladen na mobilní zařízení. Právě pro ně nejdříve budujeme layout a píšeme CSS kód, který potom pomocí media queries, typicky s použitím dotazu na vlastnost `min-width`, postupně zesložitujeme a původní jednoduchý layout pro malé obrazovky vylepšujeme a uzpůsobujeme pro větší displeje počínaje tablety až po desktop. Hojným jevem mnoha webů je přeplnění nerelevantním a neužitečným obsahem, který je nejčastěji důsledkem snahy vyplnit prázdný prostor při návrhu, kdy se začíná nejdříve desktopovou verzí. Tento nedostatek se snaží metoda mobile first co nejvíce eliminovat. Díky tomu, že návrh webu začínáme od mobilních zařízení, nutí nás to k zaměření se pouze na důležitý obsah stránky. Hned od začátku jsme totiž omezeni velmi malým množstvím prostoru, které nám tyto zařízení poskytují, což nás nabádá ke snaze o co nejjednodušší návrh, kterým zachytíme hlavní podstatu webu. Je tedy třeba důsledně zvážit účel stránek, prioritu jejich jednotlivých prvků a to, které informace jsou pro uživatele nejdůležitější. Tímto přehledným způsobem navržené hlavní jádro webu se přenesou i do desktopové verze. Další výhodou toho přístupu je, že se zredukuje celková složitost CSS kódu, který bude přehlednějším, lépe upravovatelným a rozšiřitelným. Je totiž jednodušší napsat kód, který bude postupně rozšiřován a zesložitován než kód, u kterého dochází k jeho zestručnění a částečnému popírání, jako tomu je u metody desktop first (Wroblewski, 2009). Právě pro tyto zmíněné výhody jsem pro kódování responzivních webových stránek pro firmu Návrh loga zvolil přístup mobile first, podle kterého jsem postupoval. Nejdříve jsem tedy napsal kód layoutu pro mobilní zařízení, který jsem následně zesložitoval a upravoval pro větší obrazovky desktopu tvorbou breakpointů za pomoci media queries, které se dotazují na vlastnost `min-width`.

6.2.2 Určování breakpointů

Velmi důležitou součástí tvorby responzivních webových stránek je definování tzv. breakpointů (v češtině uváděno jako zlomové či hraniční body). Jedná se o určení hranic, po jejichž překročení se rozvržení stránky změní a její obsah se přeskládá tak, aby co nejlépe využil dostupnou plochu, byl co nejvíce čitelný a orientace v něm snadná.

Breakpointy se tvoří pomocí media queries, ve kterých se v praxi dotazuje zejména na vlastnost `width` s prefixem `min` při mobile first přístupu či s prefixem `max` při desktop first přístupu. Nejčastěji se tedy rozvržení webu mění na základě šířky zobrazovacího zařízení.

V minulosti tradičním, ale i v současnosti v praxi velmi často viditelným jevem je určování breakpointů s použitím řekněme standardních šířek typických pro jednotlivé druhy zařízení. Takovými typickými a hojně využívanými hodnotami jsou 320 pixelů pro nejmenší druhy mobilních telefonů, 480 pixelů pro ty větší z nich,

768 pixelů pro tablety, 1024 pixelů pro desktopy a 1200 pixelů a více pro zařízení s velkou šířkou obrazovky. Tento způsob definování breakpointů byl funkční hlavně v dřívějších dobách, kdy byl menší počet mobilních zařízení a rozmanitost šířek jejich obrazovek nebyla tak velká, jako je tomu nyní. V dnešní době ale tento postup není zcela vhodným a soustředováním se pouze na nějaké konkrétní šířky zařízení riskujeme, že budeme vyvíjet specificky jen pro tyto šířky a tím budeme ignorovat cokoli mezi nimi. Rovněž se nejedná o přístup přívětivý k budoucnosti. Co je populární dnes, nemusí být populární i zítra. Jakmile se objeví nějaké nové zařízení s odlišnou šířkou obrazovky, budeme muset vytvořit pro něj další breakpoint, abychom udrželi krok s vývojem, což není zrovna ideální řešení (Kadlec, 2014).

Mnohem lepším přístupem je nechat samotný obsah stránky, ať on sám určí, při jaké šířce má být další breakpoint a kolik jich vlastně celkem bude. Každý web by tak měl mít svoje vlastní breakpointy v místech, kde je zrovna potřebuje. Dobrým postupem je zkoušet zvětšovat šířku okna prohlížeče a pozorovat, kdy se objeví nějaký prostor pro vylepšování v momentě, kdy se stránka přestane ideálně zobrazovat (Frost, 2013). Tento přístup je rozhodně moudřejším a k budoucnosti přívětivějším řešením než tomu je v případě použití výše zmiňovaných typických šířek jednotlivých druhů zařízení a proto jsem se dle tohoto přístupu řídit i já při kódování responzivního webu pro firmu Návrh loga. Jednotlivé breakpointy jsem tedy určoval podle toho, kdy se objevilo nějaké dostupné volné místo pro vylepšení layoutu, případně se layout přestal ideálně a přehledně zobrazovat.

Responzivní webové stránky je možné učinit ještě flexibilnějšími a přístupnějšími, pokud budeme určovat jednotlivé breakpointy v media queries za pomoci jednotky em namísto pixelů. Důvod je ten, že při odlišné velikosti písma než je ta námi původně pro web nastavená, nesedí absolutní hodnoty breakpointů v pixelech. Pokud si tedy uživatel zvětší velikost písma, případně má nastavený zoom stránky, námi definované breakpointy začnou dělat problémy a jejich chování nebude odpovídat tomu, jak jsme původně zamýšleli, což může významně rozhodit náš layout, způsobit nečitelnost a špatnou orientaci na stránce. Souvisí to i s tím, že různá zařízení používají různou výchozí velikost písma. Tento nedostatek se dá odstranit tím, že pixelové breakpointy převedeme do jednotek em. Pro jejich přepočítání platí vzorec již zmiňovaný v teoretické části, tedy cíl (naš breakpoint v px) vydělíme kontextem (aktuální velikost fontu). Díky tomuto opatření bude náš layout optimalizovaný i v případě, když si uživatel zvětší web o několik úrovní (Gardner, 2012). Tento flexibilnější způsob definice breakpointů v jednotce em jsem během kódování použil i já, jako kontext jsem dosazoval číslo 16, vzhledem k tomu, že výchozí velikost písma ve většině moderních prohlížečů odpovídá právě 16 pixelům. Výjimku tvoří několik breakpointů, které mám v CSS kódu uvedeny na jeho konci a jsou definovány v pixelech. Jedná se ale o breakpointy, které se aplikují jen na části webu, ve kterých se nenachází žádný textový obsah, ale pouze obrázky, které mají pevně zadanou maximální velikost v pixelech a proto nemá smysl hodnotu těchto breakpointů přepočítávat do jednotky em.

6.2.3 Použití relativních jednotek

Jak již bylo zmíněno v teoretické části, jedním z klíčových prvků bez kterého se responzivní webdesign neobejde je plovoucí layout, který jsem během kódování samozřejmě využil i já. To znamená, že pro nastavení šířky jednotlivých elementů na webu jsem použil relativní jednotky, konkrétně procenta. Jejich využití se nevztahuje pouze k samotné šířce, tedy CSS vlastnosti `width`, ale i k ostatním vlastnostem, které celkovou šířku prvku ovlivňují. Takovými vlastnostmi jsou `padding` a `margin`, jejichž horizontální rozměry se rovněž musí uvádět v procentech.

Relativní jednotky jsem použil také při nastavování velikosti písma na webu. Zvolil jsem jednotku `rem`, jejíž hlavní výhodou oproti ostatním relativním jednotkám, jako jsou např. procenta či jednotka `em`, je to, že nekaskáduje a její velikost se vždy vztahuje ke kořenovému elementu, jak už ostatně bylo popsáno v teoretické části. CSS vlastnost `font-size` tedy v kódu vždy uvádím v jednotce `rem` s tím, že používám tzv. fallback pro prohlížeče, které tuto poměrně novou jednotku nepodporují. To znamená, že deklaraci v jednotce `rem` předchází vždy ještě deklarace v pixelech tak, jak to bylo popsáno v teoretické části v podkapitole věnující se jednotce `rem`. Velikost písma kořenového elementu (elementu HTML), od které se následně všechny velikosti fontu na webu odvíjí, jsem nastavil pomocí deklarace `font-size: 100%`, která říká prohlížeči, aby použil svoji výchozí velikost fontu, jejíž hodnota ve většině z nich odpovídá 16 pixelům.

6.2.4 Použitý styl písma

Pro veškerý textový obsah na webu je dle grafického návrhu zvolen font s názvem Exo 2. Pro jeho použití na stránkách využívám služby Google Fonts, což je online knihovna volně dostupných písem. Vložení fontu do stránky je jednoduché a spočívá pouze v umístění vygenerované podoby tagu `link` do hlavičky HTML kódu. Následně se již písmo může v CSS kódu běžně používat přiřazením jeho názvu do vlastnosti `font-family`.

7 Responzivní web pro firmu Návrh loga

V této kapitole budou popsány jednotlivé sekce (případně jejich jednotlivé části) nakódovaných responzivních webových stránek pro firmu Návrh loga. Při popisu stručně představím obsah každé ze sekcí, hlavní důraz budu klást na uvedení toho, v čem spočívá responzivita dané sekce a jak se liší její zobrazení na jednotlivých zařízeních. Pro lepší představu přiložím snímky obrazovek (tzv. screenshoty), které umístím do příloh této bakalářské práce.

7.1 Menu

Menu je typicky zobrazeno v každé ze sekcí webu. Liší se svým zobrazením, které má tři podoby.

Pro nejmenší obrazovky je využitý tzv. off-canvas druh navigace. Po kliknutí na tlačítko Menu v horní části stránky vyjede navigace z levé strany webu, čímž ho celý odsune bokem. Její jednotlivé položky jsou zobrazeny pod sebou a jejich velikost byla zvolena tak, aby se na ně dalo snadno kliknout i palcem. Nápis Menu je navíc doplněn i typickou ikonou, tzv. hamburgerem (v češtině se jí nejčastěji říká trojrovnítko či trojzubec), která má usnadnit uživatelské hledání navigace (viz snímek obrazovky z mobilního telefonu na obr. 5). Pro implementaci tohoto druhu navigace jsem použil dostupné hotové řešení (tzv. plugin) s názvem Pushy, jehož zdroj je uvedený v CSS kódu. Pushy menu se skládá ze dvou soborů, prvním jsou kaskádové styly pro definici jeho vzhledu, druhým je JavaScriptový kód, který zajišťuje jeho funkčnost, ke které je ještě navíc potřeba nalinkovat známou JavaScriptovou knihovnu jQuery. Pluginů, které řeší tento typ navigace je na internetu dostupné velké množství, jako další zástupce mohu zmínit např. Slidebars, Mmenu, bigSlide, jPanelMenu, Snap a Navobile. Pushy jsem zvolil hlavně z toho důvodu, že při mém testování různých druhů těchto pluginů na mobilních zařízeních byl jeho chod nejrychlejší a nejplynulejší. Vedle tlačítka menu, po jeho pravé straně, je pro lepší přehlednost a orientaci na webu uvedeno v jaké sekci se návštěvník aktuálně nachází.

Druhá varianta zobrazení menu se objeví, pokud obrazovka zařízení, které web prohlíží, dosahuje šířky odpovídající hodnotě alespoň 40em. V tomto případě je navigace opět v horní části stránky, tentokrát je ale rovnou zobrazena celá a každá její položka je tvořena ikonou a dvouřádkovým nápisem s názvem sekce webu. Menu je tvořeno osmi sekcemi, do kterých se web dělí s tím, že při tomto jeho rozložení je menu zobrazeno do dvou řádků po čtyřech sekcích (viz obr. 6).

Třetí způsob rozložení navigace je velmi podobný tomu druhému. Jediný rozdíl je ten, že všech osm položek menu je umístěných vedle sebe v jednom řádku díky většímu prostoru (viz obr. 7). Tímto stylem se menu objeví, pokud zobrazovací zařízení disponuje šířkou odpovídající hodnotě alespoň 60em.

7.2 Hlavička a patička

Stejně jako menu je i hlavička a patička stránky zobrazená ve všech sekcích webu. Pro oba tyto prvky existují dva různé druhy rozložení.

Hlavička stránky je pro menší displeje zobrazena do dvou řádků (viz obr. 5). V tom prvním je zobrazeno logo firmy (kliknutí na něj typicky vede na úvodní stránku) zarovnané na střed stránky. Druhý řádek je rozdělen na dvě poloviny, v levé je uveden základní kontakt na firmu (telefonní číslo a email) a v pravé je umístěn odkaz na profil firmy na sociální síti Facebook a její typické tlačítko like (ani jeden z těchto prvků není aktivním, jelikož firma ještě nemá vytvořený facebookový profil).

V patičce stránky je u menších obrazovek typicky umístěna informace o copyrightu a navíc jsou zde ještě velmi stručně shrnuty služby poskytované firmou Návrh loga (viz obr. 5).

Ke změně rozložení hlavičky i patičky dochází při stejném breakpointu, který říká, že šířka obrazovky musí být minimálně 55em. V tomto případě je hlavička stránky zobrazena pouze v jednom řádku (viz obr. 7), ve kterém se nachází stejné prvky jako při rozložení u menších obrazovek, tedy logo (tentokrát zarovnané doleva stránky), základní kontaktní údaje na firmu a odkaz na facebookový profil. Změna patičky webu spočívá v tom, že v jejím levém okraji se ještě navíc objeví malé logo firmy (viz obr. 7), kliknutí na něj samozřejmě opět vede na úvodní stránku.

7.3 Úvodní stránka

Úvodní stránka je vstupní branou webu. Obsahuje stručné informace o poskytovaných službách firmou Návrh loga s odkazy do jednotlivých sekcí, kde se návštěvník o těchto službách dozví více informací. Dále zde nalezneme reference, tedy ukázky několika log, které firma pro své zákazníky vytvořila v minulosti. Kliknutím na obrázek loga se dostaneme na stránku s jeho větším detailem a popisem. Pod ukázkou několika referencí je umístěno tlačítko, které odkazuje na sekci Naše reference, ve které jsou zobrazeny všechny reference. Poslední částí úvodní stránky jsou stručné informace o různých cenových variantách realizace loga. U každé z nich je tlačítko, které návštěvníka nasměruje do sekce Cena loga, kde se dozví veškeré potřebné informace o ceně realizace loga.

Layout úvodní stránky má tři varianty rozložení. První je určena pro malé displeje (viz obr. 8). V ní je úvodní text zkrácen pouze na jeden odstavec, pod kterým je zobrazeno tlačítko, odkazující do sekce Tvorba loga. Pod touto částí následuje ukázka pouze dvou referencí, pod kterou se opět nachází tlačítko pro přístup do sekce Naše reference. V poslední části jsou jednotlivé cenové varianty realizace loga, které jsou umístěny pod sebou.

Druhý typ rozložení je určený pro obrazovky, jejichž šířka je alespoň 40em (viz obr. 9). Úvodní text je zde zobrazený celý v levém sloupci zabírajícím něco málo přes polovinu layoutu, vedle něj se nachází lehce užší pravý sloupec se třemi referencemi. Pod touto částí následují dvě cenové varianty realizace loga, umístěny

v jednom řádku vedle sebe, třetí cenová varianta se nachází níže pod nimi a je zarovnaná na střed stránky.

Třetí způsob rozmístění prvků na úvodní stránce se zobrazí na zařízeních, které mají šířku obrazovky alespoň 60em (viz obr. 10). Rozdíl oproti předchozí variantě rozložení je v tom, že ke sloupci s úvodním textem a sloupci se třemi referencemi přibude ještě třetí sloupec, který se zobrazí vedle nich a obsahuje další tři reference. Změna je i u zobrazení jednotlivých cenových variant realizace loga, které jsou nyní v jednom řádku, tedy vedle sebe.

7.4 Sekce Proč logo

Tato sekce informuje návštěvníka a potenciálního klienta o tom, proč by měl investovat do kvalitního loga a jaké vlastnosti by takové logo mělo mít. Následuje zmínění deseti důvodů proč svěřit jeho návrh právě firmě Návrh loga.

Tato část webu obsahuje pouze textový obsah, jehož rozložení má dvě varianty. Tou první, pro menší obrazovky, je umístění všech textů pod sebe. Druhá varianta je určena zařízením s šířkou obrazovky alespoň 60em, při níž je číslovaný seznam, obsahující deset důvodů proč svěřit návrh loga právě firmě Návrh loga, rozdělen do dvou stejně širokých sloupců vedle sebe po pěti prvcích.

7.5 Sekce Tvorba loga

Účelem této sekce je přiblížit jejímu návštěvníkovi jakým způsobem logo vzniká. Je zde popsán celý proces návrhu loga v celkem sedmi krocích tak, jak postupuje firma Návrh loga.

Jelikož je v této sekci pouze krátký textový obsah, je zobrazena na všech zařízeních stejným způsobem. Všechn text je umístěn v jednom sloupci širokém přes celý layout.

7.6 Sekce Cena loga

V této sekci jsou uvedené tři různé cenové varianty realizace loga, u kterých je do detailu rozepsáno, jaké služby přesně obsahují a pro jaký typ firmy jsou vhodné. Nalezneme zde další informace související např. s platbou, termínem dodání, garancí vrácení peněz apod.

Tato sekce má tři různé možnosti rozložení prvků. Ty se liší pouze ve způsobu zobrazení jednotlivých cenových variant realizace loga, který je shodný jako na úvodní stránce. To znamená, že na malých displejích jsou umístěny pod sebou. Na obrazovkách se šířkou alespoň 47em jsou dvě cenové varianty v jednom řádku vedle sebe a třetí je pod nimi zarovnaná na střed stránky. U displejů s šířkou minimálně 67.5em se zobrazí všechny tři cenové varianty v jednom řádku do třech sloupců vedle sebe.

7.7 Sekce Ochranné známky

Tato sekce informuje o možnosti legislativní ochrany loga, názvu firmy či doménového jména související s registrací ochranné známky. Firma Návrh loga zde nabízí vyřízení celého byrokratického procesu za zákazníka.

Responzivita této sekce spočívá ve dvou věcech. Tou první je tabulka s cenami za jednotlivé úkony. Zobrazení tabulek na malých obrazovkách je všeobecně velmi zajímavou problematikou. Na internetu jsou volně k dispozici různé způsoby řešení, jako příklad mohu uvést některé jQuery pluginy jako jsou FooTable, Stacktable, Tablesaw či ngResponsiveTables. Princip těchto řešení spočívá v tom, že jednotlivé řádky tabulky se v definovaném breakpointu přemění na jeden dlouhý sloupec obsahující název hodnoty a samotnou hodnotu. Já jsem použil hotové řešení, jehož autorem je Geoff Yuen (odkaz na jeho kód je uveden v mém CSS kódu webu). Na jeho řešení mě zaujalo, že ke své funkci nepotřebuje jQuery a není realizováno JavaScriptem, ale že je implementováno čistě jen za použití CSS a data atributů, které jsou novinkou v HTML5. Porovnání způsobu zobrazení tabulky na mobilu a na desktopu je k vidění na obr. 11. Breakpoint, který přepíná mezi těmito dvěma způsoby zobrazení tabulky je nastaven na hodnotu 40em.

Druhá věc, týkající se responzivity této sekce spočívá v zobrazení obrázku typického symbolu pro registrovanou ochrannou známku při šířce obrazovky alespoň 55em, který se objeví napravo od textu a zabere třetinu šířky stránky.

7.8 Sekce Firemní styl

V této sekci je nabízeno kompletní zpracování firemního stylu neboli corporate identity. Jsou zde popsány jeho jednotlivé součásti (např. vizitky, razítka, letáky, webové stránky apod.) spolu s jejich cenami.

Responzivní chování je zde zastoupeno rozložením obrázků. První čtyři obrázky v této sekci, které reprezentují ukázkou některých prvků firemního stylu, jsou na nejmenších displejích zobrazeny postupně po jednom pod sebou. U zařízení s šířkou obrazovky alespoň 518px jsou obrázky umístěny do dvou řádků, přičemž na každém z nich jsou dva obrázky. Na displejích s šířkou minimálně 1017px se objeví všechny čtyři obrázky v jednom řádku vedle sebe.

Responzivně se chová i číslovaný seznam, který obsahuje typické prvky firemního stylu. Na menších obrazovkách je všech šest prvků zobrazeno pod sebou. Na displejích s šířkou alespoň 60em se změní rozložení číslovaného seznamu do dvou sloupců po třech prvcích.

Poslední částí této sekce webu, která ještě mění své rozložení, je ukázkou třech obrázků realizovaných webů. Na malých displejích jsou zobrazeny jednotlivé obrázky pod sebou. Od šířky 690px se zobrazí první obrázek na středu stránky v prvním řádku a pod ním v druhém řádku zbylé dva vedle sebe. Na obrazovkách s šířkou 1024px a větší jsou už umístěny všechny tři obrázky vedle sebe na jednom řádku.

7.9 Sekce Naše reference

Tato sekce obsahuje ukázky některých z log, které firma Návrh loga vytvořila v minulosti pro své klienty. Po kliknutí na zmenšený náhled loga se návštěvník dostane na stránku s jeho detailem, kde je logo zobrazeno ve větší velikosti a s krátkým popisem.

Responzivita spočívá v rozložení šesti obrázků s ukázkami jednotlivých log. Na malých obrazovkách je všech šest umístěno pod sebou a zarovnáno na střed stránky. Na displejích s šířkou alespoň 630px dojde k přeskládání obrázků, jsou zobrazeny ve třech řádcích, kdy na každém z nich jsou dvě loga. Pokud má obrazovka šířku minimálně 930px, obrázky budou umístěny ve dvou řádcích po třech ukázkách vytvořených log.

Různé způsoby zobrazení mají i stránky s detaily jednotlivých log. Na menších displejích se zobrazí obrázek s logem zarovnaný na střed stránky a teprve pod ním text s popisem loga. Na obrazovkách s šířkou alespoň 60em se rozložení stránky změní do dvou stejně širokých sloupců vedle sebe, kde v tom levém je obrázek s logem a v pravém jeho popis.

7.10 Sekce Kontakty a o nás

Poslední sekcí webu je Kontakty a o nás. Návštěvník zde nalezne kontaktní formulář pro odeslání zprávy na email firmy, stručné informace o firmě samotné, kontaktní a fakturační údaje a seznam odkazů na kontaktní místa ve městech, ve kterých firma Návrh loga působí.

Rozvržení této sekce má dvě varianty. Na zařízeních s větší obrazovkou, které mají šířku alespoň 55em, je stránka rozdělena do dvou sloupců. Levý z nich zabírá dvě třetiny její šířky a obsahuje kontaktní formulář a odstavec s popisem firmy. Pravý sloupec zabírá jednu třetinu stránky a jsou v něm uvedeny kontaktní a fakturační údaje a seznam odkazů kontaktních míst. Na displejích menších jak 55em se všichni obsah stránky zobrazí v jednom sloupci pod sebou. Sloupec, který je na větších obrazovkách původně umístěn vpravo se na těch menších zobrazí pod kontaktním formulářem a nad částí s popisem firmy.

8 Diskuse

Nakódované responzivní webové stránky pro firmu Návrh loga jsou umístěny na adrese <https://akela.mendelu.cz/~xsliva/BP/web/htm/>. Nejedná se ještě o úplně finální verzi webu, s firmou proběhne diskuse nad konečnou podobou jednotlivých sekcí a jejich částí, takže je možné, že požadavkem bude úprava některých z nich. Původní grafický návrh totiž nepočítal s tím, že celý web bude responzivní, ale naopak s tím, že se použije fixní layout. Tím pádem zadání pro způsob zobrazení jednotlivých sekcí webu na zařízeních s menší obrazovkou nebylo úplně podrobné a do detailu vypracované. Vzhledem k tomu, že jsem stránky kódoval během konce minulého roku, začínal jsem těsně před vánočními svátky, tak závěrečná diskuse s firmou proběhne začátkem tohoto roku, kde se upřesní finální podoba celého webu a jeho jednotlivých sekcí a částí. Web bude následně umístěn na adrese <http://www.navrh-loga.cz>, kde se nachází stará verze webové prezentace firmy.

8.1 Google Mobile-friendly test na použitelnost v mobilech

Web úspěšně prošel Google testem (nástrojem Mobile-Friendly Test), který analyzuje použitelnost internetových stránek v mobilních zařízeních (viz obr. 12). Rovněž obdržel plný počet bodů v dalším testu od Googlu, tentokrát v nástroji PageSpeed Insights a jeho části věnující se uživatelskému dojmu (viz obr. 13). Mezi kritéria těchto testů patří:

- Čitelnost textu bez nutnosti zoomování.
- Celý web se musí přizpůsobovat velikosti zobrazovacího zařízení, tak aby jeho návštěvník nemusel používat zoomování a vodorovný posuvník.
- Zobrazení odkazů na webu tak, aby se na ně dalo snadno kliknout, to znamená jejich dostatečnou velikost a volný prostor kolem nich.
- Vyhnutí se softwaru, jenž není běžný na mobilních zařízeních, jako je například Flash.

Úspěšné projití webu Mobile-friendly testem znamená pozitivní ovlivnění výsledku vyhledávání Google. Jak je uvedeno v oficiální zprávě Googlu „Helping users find mobile-friendly pages“ z 18. listopadu 2014 (Imaizumi, Phan; 2014), stránky optimalizované pro mobilní zařízení, které tímto testem úspěšně projdou po splnění výše zmíněných kritérií, budou ve výsledcích mobilního vyhledávání zvýhodněny. Konkrétně zobrazením informace o tom, že jsou tzv. „Mobile-friendly“, tedy vhodné pro mobilní zařízení, což bude indikováno zobrazením nápisu (tzv. labelu) pod URL adresou stránky ve výsledcích vyhledávání. Google uvádí, že tato novinka je postupně globálně zaváděna a bude v nejbližší době spuštěna ve všech jeho jazykových mutacích. Ve zprávě se dále píše, že pro Google je velmi důležitý webový prožitek uživatelů a to právě i na mobilních zařízeních, smyslem je učinit vyhledání

informací pro uživatele co nejjednodušším a celý tento proces usnadnit. Proto je toto opatření pouze prvním krokem pro zpříjemnění procházení webu mobilních uživatelů. S tím souvisí informace, že Google experimentuje i s tou možností, že by stránky optimalizované pro mobilní zařízení zvýhodňoval v indexaci a hodnocení, což je velmi slušným příslibem do budoucnosti pro weby, které takto optimalizovány jsou a nakonec i pro samotné uživatele.

8.2 Optimalizace rychlosti načítání

Pokud bych měl zmínit nějakou věc, která by se dala na webu vylepšit, uvedl bych zcela jistě optimalizaci rychlosti načítání. Ta je u responzivních stránek důležitá, vzhledem k tomu, že je předpoklad, že budou zobrazovány na mobilních zařízeních, které kromě rychlých Wi-Fi sítí mohou používat i mobilní datové připojení s malou rychlostí. Navíc je většina těchto datových tarifů omezena tzv. FUP limitem dat, po jehož překročení dojde k razantnímu snížení přenosové rychlosti. To může vést k tomu, že se stránka bude dlouho načítat, což může způsobit její opuštění návštěvníkem.

Zrychlení načítání webu dosáhneme zmenšením jeho datového objemu (kvůli pomalým přenosovým rychlostem mobilního připojení) a snížením počtu požadavků (tzv. requestů) na server (kvůli vysoké latenci mobilního připojení). V praxi probíhá realizace např. spojením všech CSS souborů do jednoho jediného (to samé platí i pro JavaScriptové soubory), který následně tzv. minifikujeme.

Největší zátěží na objem dat jsou na webu zcela jistě obrázky. Použitím postupu z podkapitoly 3.3 Flexibilní obrázky, zajistíme pouze flexibilitu obrázků, které se budou zmenšovat pouze, co se týče jejich rozměrů, ale nikoliv datové velikosti. Obrázek se vždy stáhne v jeho plné datové velikosti a teprve pak se zmenší rozměrově, dle svého obalového elementu. Pro mobilní zařízení jsou tedy stahovány stejné obrázky jako pro desktopy, které jsou pro malé obrazovky neúměrně a zbytečně velké. Tímto problémem se zabývá Responsive Images Community Group (dále jen RICG), která spadá pod konsorcium W3C. RICG vyvíjí řešení spočívající v použití nového HTML elementu `picture`, který umožňuje deklaraci více zdrojů (různých rozměrových a tím pádem i datových velikostí) pro jeden obrázek. `Picture` element ale zatím bohužel není součástí žádného standardu definovaného W3C, ale některé prohlížeče se ho rozhodly podporovat už nyní ve svých nejnovějších verzích. Nicméně kvůli slabé podpoře tohoto řešení se v praxi používají jiné metody. Zřejmě nejpoužívanější je použití `Picturefillu`, což je JavaScriptový soubor (tzv. `polyfill`), který emuluje chování právě zmiňovaného elementu `picture` v prohlížečích, které jej zatím nepodporují.

8.3 CSS pro tisk

Jako další možnost vylepšení webu se nabízí vytvoření kaskádových stylů určených pro tisk, které se aplikují v případě, že si návštěvník webu bude chtít některé informace vytisknout. Tento styl by měl odstranit přebytečnou grafiku, která nemá

žádnou informační hodnotu a navíc dbát na minimální spotřebu barvy. Tiskový styl má být strohý, pouze informační a obsahující relevantní informace, nikoliv však reprezentativní. Většinou je tedy jediným žádoucím grafickým prvkem logotyp, umístěný nahoře stránky sloužící k identifikaci dokumentu. Na škodu není také zobrazení kontaktních údajů.

9 Závěr

Hlavním cílem této bakalářské práce bylo nakódování responzivní webové stránky pro firmu Návrh loga dle obdrženého grafického návrhu vytvořeného v programu Adobe Photoshop. Nejdříve jsem se tedy musel naučit pracovat s tímto profesionálním bitmapovým grafickým editorem, abych si dokázal tzv. nařezat grafiku, což znamená vyexportování všech obrázků potřebných pro vytvoření webu z grafického návrhu.

Následovala samotná tvorba responzivních webových stránek, při které jsem využil technologie HTML, CSS a JavaScript. Při kódování jsem postupoval dle moderního přístupu Mobile first, začínal jsem tedy s kódováním layoutu pro nejmenší obrazovky mobilních zařízení a postupně jsem ho vylepšoval, za pomoci media queries v definovaných breakpointech, až k jeho verzi rozložení na desktopu. Jednotlivé breakpointy, při kterých web mění způsob svého uspořádání, jsem určoval podle obsahu samotného v místech, kde to zrovna bylo potřeba. Nepostupoval jsem tedy podle staršího způsobu definice breakpointů, kdy se jako hranice používaly hodnoty typických šířek rozlišení obrazovek jednotlivých druhů mobilních zařízení.

Konečným výsledkem je responzivní webová stránka umístěná na adrese <https://akela.mendelu.cz/~xsliva/BP/web/htm/>. Jak již bylo podrobně rozepsáno v předchozí kapitole Diskuse kladnou stránkou je, že web bez problémů prošel dvěma testy společnosti Google se ziskem plného počtu bodů. Jedná se o testy, které analyzují použitelnost internetových stránek v mobilních zařízeních a hodnotí jejich uživatelský dojem dle daných kritérií.

Finálnímu nasazení na oficiální doménu firmy nacházející se na adrese <http://www.navrh-loga.cz> bude předcházet ještě jedno sezení, na kterém se s firmou proberou eventuální konečné úpravy jednotlivých sekcí či částí webu. Firma pak získá kompletně zpracovaný responzivní web, který nahradí starou verzi firemní webové prezentace, která není responzivní a navíc ani obsahově úplná. Díky responzivnímu chování se bude nový web příjemně a přehledně zobrazovat všem jeho návštěvníkům, nehledě na jejich zobrazovací zařízení, což může být výhodou oproti konkurenci, která takto zpracované stránky nemá.

Případný prostor k vylepšení webu vidím v optimalizaci jeho rychlosti načítání. To znamená minimalizace jeho datového objemu a počtu požadavků na server, tak jak je podrobněji popsáno v předchozí kapitole Diskuse.

Na úplný závěr bych celkově zhodnotil responzivní webdesign jako takový. Myslím si, že v budoucnosti bude ještě více nabírat na důležitosti a bude častějším požadavkem při tvorbě webových stránek, až se časem stane téměř samozřejmostí. To souvisí s počtem mobilních zařízení, která budou schopny přistupovat na Internet a prohlížet tak webové stránky. Domnívám se, že jejich spektrum bude čím dál tím širší a i nadále bude pokračovat růst prodeje těchto zařízení a jejich obliba u uživatelů, což bude mít za následek další zvětšování podílu návštěv webových stránek právě z mobilních zařízení, což podpoří nutnost vytvářet responzivní stránky. V současnosti vidím vytváření responzivní webových stránek nejdůležitěj-

ší hlavně u prezentačních webů firem. Často se potenciální zákazník dostane poprvé do styku s firmou právě prostřednictvím jejich webových stránek, a pokud se mu stránky zobrazí v čitelné a přehledné formě i na mobilním zařízení, z kterých čím dál častěji jejich uživatelé vyhledávají informace, zcela jistě to na takového návštěvníka zapůsobí pozitivně například oproti konkurenční firmě, která responzivně zpracovaný web nemá a tudíž je jeho prohlížení nepohodlné. Budoucnost má určitě i online nakupování prostřednictvím mobilních zařízení a responzivně zpracovaných stránek internetových obchodů. Už nyní je například v Americe velmi zajímavý podíl nákupů v e-shopech uskutečněných právě prostřednictvím mobilních zařízení. Dle mého názoru si responzivní webdesign postupem času ještě více upevní svoje vnímání jako potřebného a velmi užitečného přístupu jak vytvářet webové stránky a stane se tak respektovaným tématem mezi webdesignéry.

10 Literatura

- ASOCIACE PROVOZOVATELŮ MOBILNÍCH SÍTÍ. *Internet v mobilu v reálných číslech* [online]. 10. 10. 2013 [cit. 2014-11-29]. Dostupné z: <http://www.apms.cz/stazeni-souboru/251/mif2013-uzivatele-internetu-v-mobilu.pdf?fd=fd>
- CAN I USE. *CSS3 Media Queries* [online]. 2014 [cit. 2014-12-16]. Dostupné z: <http://caniuse.com/#feat=css-mediaqueries>
- CASTLEDINE, E., EFTOS, M., WHEELER, M. *Vytváříme mobilní web a aplikace pro chytré telefony a tablety*. Brno: Computer Press, 2013. ISBN 978-80-251-3763-5.
- CASTRO, E., HYSLOP, B. *HTML5 a CSS3: Národní průvodce tvorbou WWW stránek*. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.
- CEJNEK, V. *Webdesign v Adobe Illustrator (3. díl – Grid)* [online]. 1. 10. 2014 [cit. 2014-12-10]. Dostupné z: <http://www.atlantic.cz/webdesign-adobe-illustrator-3-dil-grid/>
- ČESKÝ KOŠÍK ROKU. *Celá studie* [online]. 2014 [cit. 2014-11-18]. Dostupné z: <http://www.ceskykosikroku.cz/cela-studie.html>
- DOMES, M. *333 tipů a triků pro CSS*. Brno: Computer Press, 2011. ISBN 978-80-251-3366-8.
- DRŽKA, M., MICHÁLEK, K. *Obama nebo Romney? Rozhovor o mobilních verzích webů s Martinem Držkou* [online]. 27. 3. 2013 [cit. 2014-11-17]. Dostupné z: <http://ebrana.cz/magazin/obama-nebo-romney-rozhovor-o-mobilnich-verzich-webu-s-martinem-drzkou>
- FAR, P. *Recommendations for building smartphone-optimized websites* [online]. 6. 6. 2012 [cit. 2014-11-12]. Dostupné z: <http://googlewebmastercentral.blogspot.cz/2012/06/recommendations-for-building-smartphone.html>
- FORD, N. *Content-out Layout* [online]. 25. 3. 2014 [cit. 2014-12-10]. Dostupné z: <http://alistapart.com/article/content-out-layout>
- FORGÁČ, J. *Mobilní nebo responzivní web?* [online]. 9. 3. 2013 [cit. 2014-11-17]. Dostupné z: <http://www.artweby.cz/blog/mobilni-stranky-nebo-responzivni-web>
- FROST, B. *7 Habits of Highly Effective Media Queries* [online]. 18. 9. 2013 [cit. 2015-01-02]. Dostupné z: <http://bradfrost.com/blog/post/7-habits-of-highly-effective-media-queries/>
- GARDNER, L. *The EMs have it: Proportional Media Queries FTW!* [online]. 28. 3. 2012 [cit. 2015-01-02]. Dostupné z: <http://blog.cloudfour.com/the-ems-have-it-proportional-media-queries-ftw/>
- GEMIUS. *PC vs. Non-PC generated traffic* [online]. 29. 11. 2014 [cit. 2014-11-29]. Dostupné z: <http://rankings.cz/en/rankings/pc-vs-nonpc.html>

- GOOGLE DEVELOPERS. *Building Smartphone-Optimized Websites* [online]. 11. 2. 2014 [cit. 2014-11-12]. Dostupné z: <https://developers.google.com/webmasters/smartphone-sites/details>
- GOOGLE DEVELOPERS. *Configure the Viewport* [online]. 28. 8. 2014 [cit. 2014-12-14]. Dostupné z: <https://developers.google.com/speed/docs/insights/ConfigureViewport>
- HOGAN, B. *HTML5 a CSS3: Výukový kurz webového vývoje*. Brno: Computer Press, 2011. ISBN 978-80-251-3576-1.
- IMAIZUMI, R., PHAN, D. *Helping users find mobile-friendly pages* [online]. 18. 11. 2014 [cit. 2015-01-04]. Dostupné z: <http://googlewebmastercentral.blogspot.cz/2014/11/helping-users-find-mobile-friendly-pages.html>
- JASHINSKY, L. *The Pros and Cons of Responsive Web Design* [online]. 28. 2. 2013 [cit. 2014-11-12]. Dostupné z: <http://keystoneclick.com/blog/web-design/pros-and-cons-responsive-web-design>
- JEHL, S. *Respond.js* [online]. 2011 [cit. 2014-12-17]. Dostupné z: <https://github.com/scottjehl/Respond>
- KADLEC, T. *Responzivní design profesionálně*. Brno: Zoner Press, 2014. Encyklopedie Zoner Press. ISBN 978-80-7413-280-3.
- LICHÝ, A. *Projekt Český košík roku otestoval uživatelskou přívětivost českých e-shopů* [online]. 10. 9. 2014 [cit. 2014-11-18]. Dostupné z: <http://businessworld.cz/net/projekt-cesky-kosik-roku-otestoval-uzivatelskou-privetivost-ceskych-e-shopu-11880>
- MARCOTTE, E. *Fluid Grids* [online]. 3. 3. 2009 [cit. 2014-12-08]. Dostupné z: <http://alistapart.com/article/fluidgrids>
- MARCOTTE, E. *Fluid Images* [online]. 17. 4. 2009 [cit. 2015-01-04]. Dostupné z: <http://unstoppablerobotninja.com/entry/fluid-images>
- MARCOTTE, E. *Responsive Web Design* [online]. 25. 5. 2010 [cit. 2014-01-19]. Dostupné z: <http://alistapart.com/article/responsive-web-design>
- MARCOTTE, E. *Type study: Sizing the legible letter* [online]. 9. 11. 2011 [cit. 2014-12-06]. Dostupné z: <http://blog.typekit.com/2011/11/09/type-study-sizing-the-legible-letter/>
- MICHÁLEK, M. *Mobilní zařízení a responzivní webdesign* [online]. 13. 3. 2014 [cit. 2014-11-17]. Dostupné z: <http://www.wug.cz/praha/akce/628-Mobilni-zarizeni-a-responzivni-webdesign>
- MOZILLA DEVELOPER NETWORK. *JavaScript Overview* [online]. 7. 3. 2014 [cit. 2014-12-30]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/JavaScript_Overview
- MOZILLA DEVELOPER NETWORK. *Introduction to HTML* [online]. 26. 12. 2014 [cit. 2014-12-30]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Introduction>

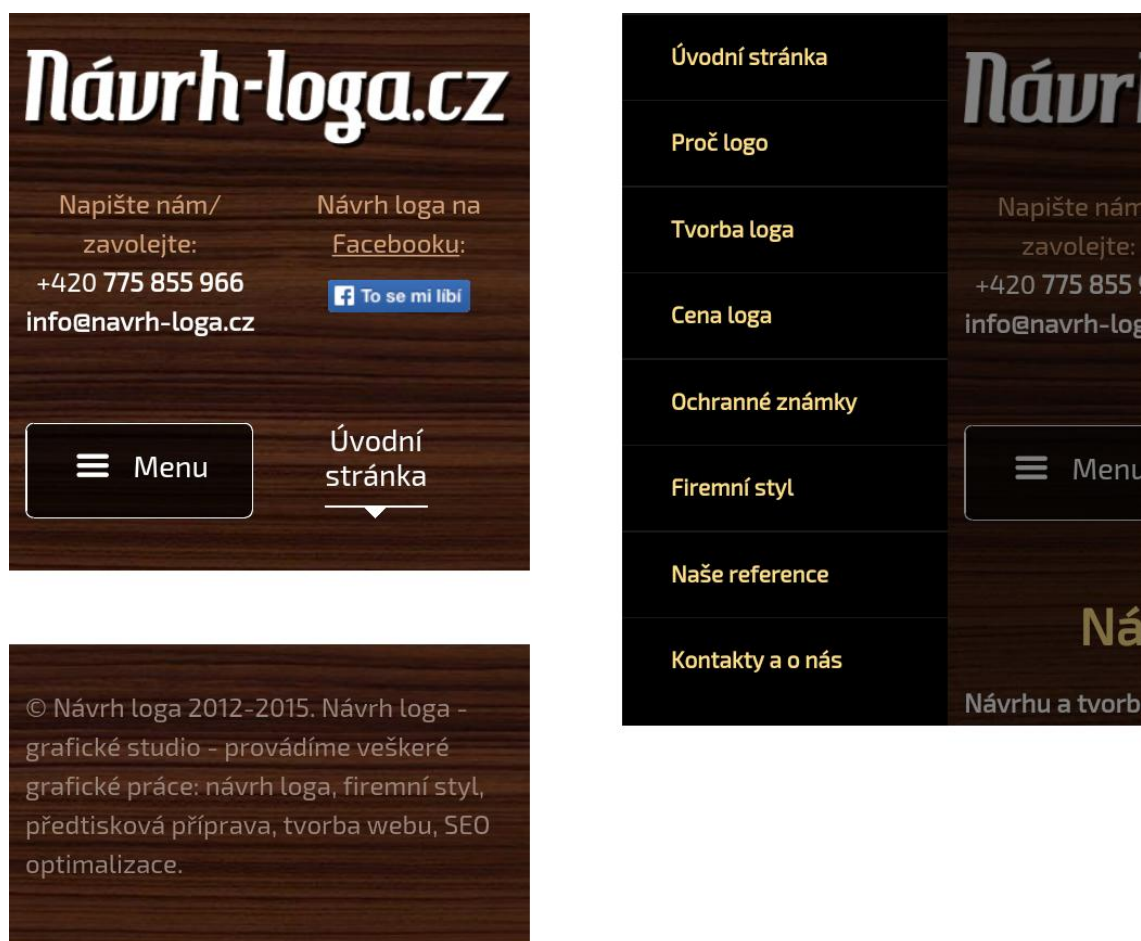
- NIMESH, R. *How Fluid Grids Work in Responsive Web Design* [online]. 29. 11. 2012 [cit. 2014-12-06]. Dostupné z: <http://www.1stwebdesigner.com/tutorials/fluid-grids-in-responsive-design/>
- PEŠOUT, M. *Určujte relativní velikost fontů pomocí CSS3* [online]. 9. 2. 2012 [cit. 2014-12-07]. Dostupné z: <https://igloonet.cz/blog/urcuje-relativni-velikost-fontu-pomoci-css3/>
- POLZER, J. *Výběr frameworků, které vám usnadní nejen tvorbu responsivního webu* [online]. 30. 9. 2013 [cit. 2014-12-10]. Dostupné z: <https://www.maxiorel.cz/vyber-frameworku-ktere-vam-usnadni-nejen-tvorbu-responsivniho-webu>
- PROKOP, O. *Responzivní webdesign – k čemu je to dobré?* [online]. 13. 2. 2013 [cit. 2014-11-12]. Dostupné z: <http://www.goodygoody.cz/2013/02/responzivni-webdesign-k-cemu-to-je-dobre/>
- RAMPTON, J. *How Important Is Responsive Design?* [online]. 5. 6. 2014 [cit. 2014-11-12]. Dostupné z: http://www.huffingtonpost.com/john-rampton/business-mobile-responsive-design_b_5267077.html
- SPIR - NETMONITOR. *Měsíční zpráva - Říjen 2014* [online]. Říjen 2014 [cit. 2014-11-29]. Dostupné z: http://www.netmonitor.cz/sites/default/files/vvnetmon/2014_10_netmonitor_offline_report.pdf
- STATCOUNTER. *Comparison in Czech Republic from Oct 2013 to Nov 2014* [online]. Listopad 2014 [cit. 2014-11-29]. Dostupné z: <http://gs.statcounter.com/#all-comparison-CZ-monthly-201310-201411>
- STATCOUNTER. *Comparison Per Country, Oct 2014* [online]. Říjen 2014a [cit. 2014-11-29]. Dostupné z: <http://gs.statcounter.com/#all-comparison-ww-monthly-201410-201410-map>
- STATCOUNTER. *Comparison Per Country in Europe, Oct 2014* [online]. Říjen 2014b [cit. 2014-11-29]. Dostupné z: <http://gs.statcounter.com/#all-comparison-eu-monthly-201410-201410-map>
- ŠKRÁŠEK, J. *Nástroje responsivního webdesignu* [online]. 12. 8. 2013 [cit. 2014-12-15]. Dostupné z: <http://programujte.com/clanek/2013062900-nastroje-responzivniho-webdesignu/>
- TING, P. *Pros and cons to responsive web design* [online]. 7. 6. 2013 [cit. 2014-11-12]. Dostupné z: <http://www.tigerspike.com/blog/2013/06/07/pros-and-cons-to-responsive-web-design/>
- TRANFICI, A. *Responsive Web Design: Fluid Layouts* [online]. 28. 1. 2013 [cit. 2014-12-05]. Dostupné z: <http://www.sitepoint.com/responsive-web-design-fluid-layouts/>
- VÁPENÍK, P. *Odvrácená strana responsivního webdesignu* [online]. 4. 3. 2013 [cit. 2014-11-12]. Dostupné z: <http://www.sunitka.cz/c/558-odvracena-strana-responzivniho-webdesignu>

- VINH, K. *Ordering disorder: grid principles for web design*. Berkeley, Calif.: New Riders, 2010. ISBN 03-217-0353-7.
- W3C. *CSS Device Adaptation* [online]. 15. 9. 2011 [cit. 2014-12-14]. Dostupné z: <http://www.w3.org/TR/css-device-adapt/#viewport-meta-element>
- W3C. *Media Queries* [online]. 19. 6. 2012 [cit. 2014-12-16]. Dostupné z: <http://www.w3.org/TR/css3-mediaqueries/>
- W3C. *Media types* [online]. 7. 6. 2011 [cit. 2014-12-15]. Dostupné z: <http://www.w3.org/TR/CSS2/media.html>
- WROBLEWSKI, L. *Mobile First* [online]. 2. 11. 2009 [cit. 2015-01-01]. Dostupné z: <http://www.lukew.com/ff/entry.asp?933>

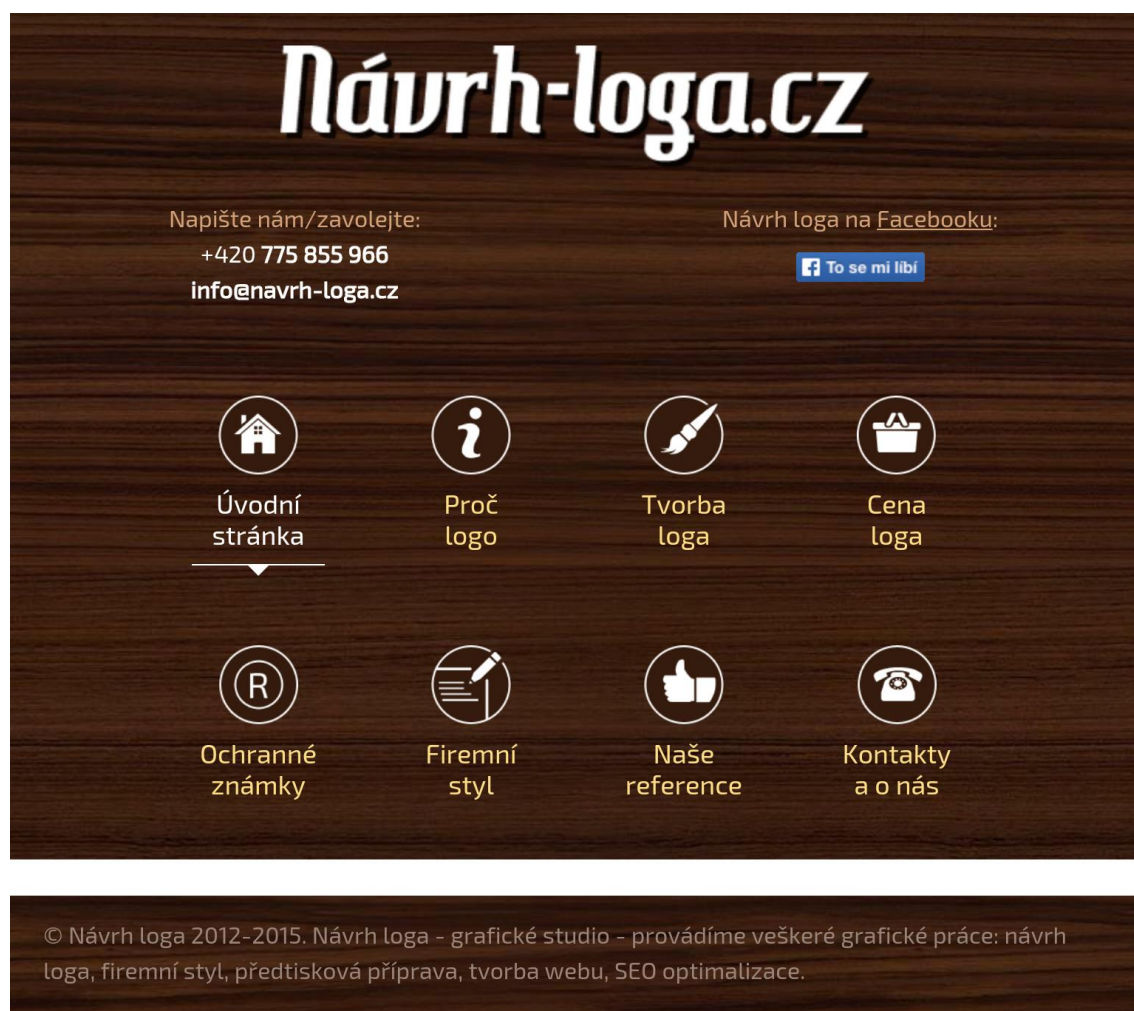
Přílohy

A Snímky obrazovek

Následující obrázky jsou snímky obrazovek (tzv. screenshoty) z různých druhů zařízení (mobilní telefon, tablet, desktop), na kterých byl načten nakódovaný responzivní web pro firmu Návrh loga.



Obr. 5 Ukázka hlavičky a patičky stránky (vlevo) a zobrazení menu (vpravo) při načtení webu na mobilním telefonu



Obr. 6 Menu, hlavička a patička stránky při zobrazení na tabletu





Obr. 7 Menu, hlavička a patička stránky při zobrazení na desktopu

Návrh loga

Návrhu a tvorbě loga je třeba věnovat patřičnou pozornost. Logo je důležitou součástí firemní identity. Logo o Vaší firmě vypovídá na první pohled. Právě na logu velmi záleží, jaký první dojem na potenciálního zákazníka uděláte. Více v sekci [Proč logo](#).

Přečíst celý článek

Další reference

Originální a kvalitní logo nemusí stát desetitisíce

Chci levné logo	2 490 Kč	
Dobré logo s malým rozpočtem		
<ul style="list-style-type: none"> ✓ Minimálně 2 kvalitní návrhy k výběru 	<div style="border: 1px solid black; padding: 5px; background-color: #333; color: white;">Více</div>	
Na logu mi záleží		4 490 Kč
Zlatá střední cesta, plnohodnotné logo		
<ul style="list-style-type: none"> ✓ Minimálně 4 kvalitní návrhy k výběru 	<div style="border: 1px solid black; padding: 5px; background-color: #333; color: white;">Více</div>	
Chci prvotřídní servis	9 490 Kč	
Profi logo s manuálem pro náročné		
<ul style="list-style-type: none"> ✓ Minimálně 8 kvalitních návrhů od více grafiků k výběru 	<div style="border: 1px solid black; padding: 5px; background-color: #333; color: white;">Více</div>	

Obr. 8 Úvodní stránka při zobrazení na mobilu, vlevo hořejší část, na kterou navazuje část zobrazená vpravo

Návrh loga

Návrhu a tvorbě loga je třeba věnovat patřičnou pozornost. Logo je důležitou součástí firemní identity. Logo o Vaší firmě vypovídá na první pohled. Právě na logu velmi záleží, jaký první dojem na potenciálního zákazníka uděláte. Více v sekci [Proč logo](#).

Nepodceňujte sílu loga a nechte si zpracovat [firemní styl](#) u nás.

Logo i jiné prvky firemního stylu by měly reprezentovat Vaší firmu přesně tak, jak je Vaším cílem. Proto každému projektu věnujeme maximální pozornost. Přečtěte si, [jak logo navrhujeme](#).

Na prezentaci záleží, mnohdy právě profesionální logo a vůbec celá jednotná [firemní identita](#) rozhoduje o výběru firmy více, než skutečná kvalita jejich služeb.

Kromě návrhu loga Vám rádi zajistíme také [realizaci moderního webu](#) nebo [registraci ochranné známky](#).





[Další reference](#)

Originální a kvalitní logo nemusí stát desetitisíce

<p>Chci levné logo 2 490 Kč</p> <p>Dobré logo s malým rozpočtem</p> <p>✓ Minimálně 2 kvalitní návrhy k výběru</p> <p>Více informací</p>	<p>Na logu mi záleží 4 490 Kč</p> <p>Zlatá střední cesta, plnohodnotné logo</p> <p>✓ Minimálně 4 kvalitní návrhy k výběru</p> <p>Více informací</p>
<p>Chci prvotřídní servis 9 490 Kč</p> <p>Profi logo s manuálem pro náročné</p> <p>✓ Minimálně 8 kvalitních návrhů od více grafiků k výběru</p> <p>Více informací</p>	

Obr. 9 Úvodní stránka při zobrazení na tabletu

Návrh loga

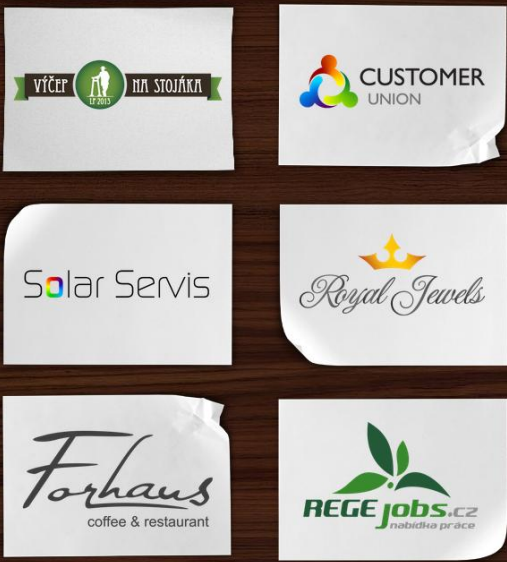
Návrhu a tvorbě loga je třeba věnovat patřičnou pozornost. Logo je důležitou součástí firemní identity. Logo o Vaší firmě vypovídá na první pohled. Právě na logu velmi záleží, jaký první dojem na potenciálního zákazníka uděláte. Více v sekci [Proč logo](#).

Nepodceňujte sílu loga a nechte si zpracovat **firemní styl** u nás.

Logo i jiné prvky firemního stylu by měly reprezentovat Vaší firmu přesně tak, jak je Vaším cílem. Proto každému projektu věnujeme maximální pozornost. Přečtěte si, **jak logo navrhujeme**.

Na prezentaci záleží, mnohdy právě profesionální logo a vůbec celá jednotná **firemní identita** rozhoduje o výběru firmy více, než skutečná kvalita jejich služeb.

Kromě návrhu loga Vám rádi zajistíme také **realizaci moderního webu** nebo **registraci ochranné známky**.



[Další reference](#)

Originální a kvalitní logo nemusí stát desetitisíce

<p>Chci levné logo 2 490 Kč</p> <p>Dobré logo s malým rozpočtem</p> <p>✓ Minimálně 2 kvalitní návrhy k výběru</p> <p>Více informací</p>	<p>Na logu mi záleží 4 490 Kč</p> <p>Zlatá střední cesta, plnohodnotné logo</p> <p>✓ Minimálně 4 kvalitní návrhy k výběru</p> <p>Více informací</p>	<p>Chci prvotřídní servis 9 490 Kč</p> <p>Profi logo s manuálem pro náročné</p> <p>✓ Minimálně 8 kvalitních návrhů od více grafiků k výběru</p> <p>Více informací</p>
--	--	--

Obr. 10 Úvodní stránka při zobrazení na desktopu

Ochranná známka ČR/SR: Rešerše na zápisnou způsobilost
Poplatek úřadu: -
Tarifní odměna: 1 000 Kč
Ochranná známka ČR/SR: Podání přihlášky ochranné známky do max 3 tříd
Poplatek úřadu: 5 000 Kč
Tarifní odměna: 4 000 Kč
Ochranná známka ČR/SR: Každá další třída v přihlášce
Poplatek úřadu: 500 Kč
Tarifní odměna: 500 Kč

Ochranná známka ČR/SR	Poplatek úřadu	Tarifní odměna
Rešerše na zápisnou způsobilost	-	1 000 Kč
Podání přihlášky ochranné známky do max 3 tříd	5 000 Kč	4 000 Kč
Každá další třída v přihlášce	500 Kč	500 Kč

Obr. 11 Zobrazení tabulky v sekci Ochranné známky, nahoře na mobilu, dole na desktopu

Google Developers

Průvodce pro mobily Začínáme Dokumentace Test použitelnosti v mobilech

Test použitelnosti v mobilech [Tweet](#) [Sdílet](#) [G+](#)

<https://akela.mendelu.cz/~xsliva/BP/web/html/> [ANALYZOVAT](#)

Skvěle! Tato stránka je použitelná v mobilních zařízeních.

Jak tuto stránku vidí Googlebot

Daří informace o stránkách vhodných pro mobily

Pokud vás v souvislosti s mobilními weby zajímá něco dalšího, přečtěte si naši [příručku k vytváření stránek pro mobily](#) nebo [principy návrhu webů](#) na stránkách Základy webu.

Používáte Nástroje pro webmastery Google?

Přihlaste se ke svému [účtu Nástrojů pro webmastery](#), abyste zjistili, kolik vašich stránek je použitelných v mobilních zařízeních.

Poskytnout zpětnou vazbu

Narazili jste na problém s testem? Máte komentáře nebo dotazy ohledně výsledků? Přidejte příspěvek do naší [diskusní skupiny](#).

Obr. 12 Výsledek Google Mobile-friendly testu na použitelnost v mobilech

100 / 100 Uživatelský dojem

✓ Gratulujeme vám! Nebyly nalezeny žádné problémy.

Nakonfigurujte viewport

Stránka určuje viewport podle velikosti zařízení, což umožňuje vykreslit stránku správně na všech zařízeních. Další informace o [konfiguraci viewportů](#)

Používejte čitelné velikosti písma

Text na vaší stránce je čitelný. Další informace o [používání čitelných velikostí písma](#)

Přizpůsobte velikost obsahu zobrazované oblasti

Obsah stránky se vchází do zobrazované oblasti. Další informace o [přizpůsobení velikosti obsahu velikosti zobrazované oblasti](#).

Vyhňte se pluginům

Vaše stránka zřejmě pluginy nepoužívá. Pluginy mohou bránit použití obsahu na celé řadě platform. Přečtěte si další informace o tom, proč je důležité se pluginům [vyhýbat](#).

Zvolte vhodnou velikost prvků pro klepnutí

Všechny odkazy a tlačítka na stránce jsou dostatečně velká, aby na ně uživatelé mohli snadno klepnout na dotykových displejích. Další informace o [volbě správné velikosti prvků pro klepnutí](#)

Obr. 13 Výsledek testu Google PageSpeed Insights – jeho části věnující se uživatelskému dojm

B Přiložené DVD

Přiložené DVD obsahuje veškeré zdrojové kódy (HTML, CSS, JavaScript) vytvořených responzivních webových stránek pro firmu Návrh loga. Je zde k nalezení i původní kompletní grafický návrh ve formátu PSD, vyexportovány jsou i podoby jednotlivých sekcí webu ve formátu PNG pro případ, že by zájemce neměl k dispozici program Adobe Photoshop či jiný pro otevření návrhu ve formátu PSD.