

**Univerzita Hradec Králové**  
**Fakulta informatiky a managementu**  
**Katedra informatiky a kvantitativních metod**

# **Responzivní design webového portálu**

Diplomová práce

Autor: Marie Janáčková

Studijní obor: Informační management (5)

Vedoucí práce: Ing. Pavel Kříž, Ph.D.

Hradec Králové

duben 2015

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracovala samostatně a s použitím uvedené literatury.

V Hradci Králové dne 29. 4. 2015

Marie Janáčková

#### Poděkování:

Ráda bych poděkovala vedoucímu práce, Ing. Pavlu Křížovi, Ph.D., za odborné vedení a cenné rady při zpracovávání diplomové práce. Dále bych chtěla poděkovat občanskému sdružení Naši politici o. s. za poskytnutí možnosti implementovat responzivní design na jejich webovém portálu. Velký dík patří manželovi a rodině za podporu při psaní této práce.

## **Anotace**

Tématem práce je responzivní design jako způsob řešení problémů při zobrazení webových stránek na různých zařízeních. Cílem práce je shrnout poznatky v oblasti responzivního designu a vytvořit obecný návod k problematice tvorby responzivních webů. Práce analyzuje responzivní přístup při tvorbě webu a doporučované postupy pro návrh a implementaci. Poznatky jsou aplikovány při tvorbě responzivního designu pro webový portál [www.nasipolitici.cz](http://www.nasipolitici.cz). Práce nabízí přehled identifikovaných prvků, které mohou být součástí webové stránky a kterým je potřeba při tvorbě responzivního webu věnovat pozornost.

## **Annotation**

### **Title: Responsive Design of Web Portal**

This diploma thesis deals with responsive design as a way of problem solving for web pages when displayed on various devices. The aim is to summarise the findings in the area of responsive design and create a general manual for the creation of responsive websites. The thesis analyses the responsive approach for creating a website and the recommended methods for the design and the implementation. The findings are then applied to the creation of a responsive design of the web portal [www.nasipolitici.cz](http://www.nasipolitici.cz). The thesis offers an overview of the recognized elements that could be a part of the website, and which require one's attention when creating a responsive website.

## Obsah

<b>1</b>	<b>Úvod</b> .....	<b>1</b>
<b>2</b>	<b>Zobrazovací zařízení a design webu</b> .....	<b>2</b>
2.1	Současné trendy.....	2
2.2	Mobilní zařízení .....	4
2.3	Rozvržení designu webu .....	5
2.3.1	Statické rozvržení.....	6
2.3.2	Flexibilní rozvržení.....	7
2.3.3	Elastické rozvržení.....	8
2.3.4	Kombinované rozvržení.....	9
2.4	Responzivní design webu.....	9
2.4.1	Media queries.....	10
2.4.2	Flexibilní obrázky.....	12
2.4.3	Mřížka .....	15
2.4.4	Typografie .....	15
2.5	CSS preprocesory .....	16
2.5.1	Proměnné, mixins.....	17
2.5.2	Import souborů a validace .....	17
2.5.3	Zanořování prvků.....	18
2.6	Responzivní frameworky .....	18
2.7	Doporučení pro tvorbu responzivního webu.....	19
2.7.1	Určení breakpointů.....	19
2.7.2	Velikost písma pomocí relativních jednotek.....	20
2.7.3	Optimalizace pro velká rozlišení .....	20
2.7.4	Optimalizace pro dotykové displeje.....	21
2.7.5	Jednotný přístup k webu napříč zařízeními.....	21

2.7.6	Správná šířka zobrazení .....	22
2.8	Testování responzivního designu .....	22
<b>3</b>	<b>Analýza výchozího stavu webu Nasipolitici.cz .....</b>	<b>24</b>
3.1	Analýza návštěvnosti .....	24
3.2	Výchozí technické řešení webu.....	26
<b>4</b>	<b>Implementace responzivního designu .....</b>	<b>27</b>
4.1	Identifikování typů stránek.....	27
4.2	Stanovení priorit.....	27
4.3	Vytvoření prototypu.....	29
4.4	Postup implementace .....	30
4.4.1	Navigace .....	31
4.4.2	Mřížkový layout .....	36
4.4.3	Skrývání prvků .....	37
4.4.4	Změna pořadí prvků.....	38
4.4.5	Formuláře .....	40
4.4.6	Obrázky .....	42
4.4.7	Graf.....	44
4.4.8	Stromová struktura .....	46
4.4.9	Carousel.....	50
4.4.10	Záložky.....	50
4.4.11	Podpora starších prohlížečů .....	53
<b>5</b>	<b>Shrnutí a testování výsledného řešení .....</b>	<b>55</b>
5.1	Testování .....	55
5.2	Shrnutí výsledného řešení .....	56
<b>6</b>	<b>Závěr.....</b>	<b>58</b>

## 1 Úvod

Responzivní design webu je velmi aktuální téma, které reaguje na potřebu přizpůsobení webových stránek různé velikosti displejů na stále rozmanitější škále mobilních zařízení, která mají možnost přistupovat na internet. Kromě toho, že je to téma aktuální, je také velmi mladé. Responzivní design webu se teprve formuje a velkou rychlostí vyvíjí. Rozhodnutí společnosti implementovat své webové stránky v responzivním designu neznamena jen držet krok s dobou, ale především nepřijít o uživatele, kteří preferují prohlížení webu na mobilních zařízeních, a zajistit jim potřebné pohodlí.

Občanské sdružení Naši politici o. s. mi nabídlo možnost vypracování diplomové práce na téma responzivního designu, jejíž součástí bude realizace responzivního designu na jejich informačním portálu [www.nasipolitici.cz](http://www.nasipolitici.cz). Toto téma práce jsem si zvolila proto, že implementace designu webu je oblast, která mě zajímá a kterou jsem se již dříve zabývala, a zároveň se v případě responzivního designu jedná o velmi aktuální téma.

Hlavním cílem této práce je popsat tvorbu responzivního designu webového portálu. Jde především o vytvoření obecného postupu implementace, který bude obsahovat návod, jak vytvořit responzivní design od fáze návrhu až po testování a jak implementovat jednotlivé prvky webové stránky, jako je například navigace.

Druhým cílem práce je vytvoření a nasazení responzivního designu pro informační portál [www.nasipolitici.cz](http://www.nasipolitici.cz), který bude vycházet z původního neresponzivního designu. Tato implementace bude sloužit pro demonstraci popisovaných případů ve vytvořeném obecném návodu.

Nejprve se práce zabývá teoretickými východiskami pro implementaci responzivního designu. Obsahuje popis responzivního přístupu k tvorbě webu, dále představení některých nástrojů a *frameworků*, které mohou pomoci při implementaci, a nakonec vybraná doporučení, která je dobré při tvorbě responzivního webu využít.

Následuje samotný postup implementace. Jeho obsahem je také popis identifikovaných částí webu, kterým je potřeba při tvorbě responzivního webu věnovat pozornost. V závěru se práce zabývá testováním provedené implementace a vyhodnocením výsledků.

Diplomová práce obsahuje cizí výrazy a anglicismy, které se obvykle nepřekládají a nemají zažitý český překlad. Tyto pojmy jsou pro přehlednost označeny kurzívou, a pokud nejsou příliš známé, jsou vysvětleny přímo v textu práce.

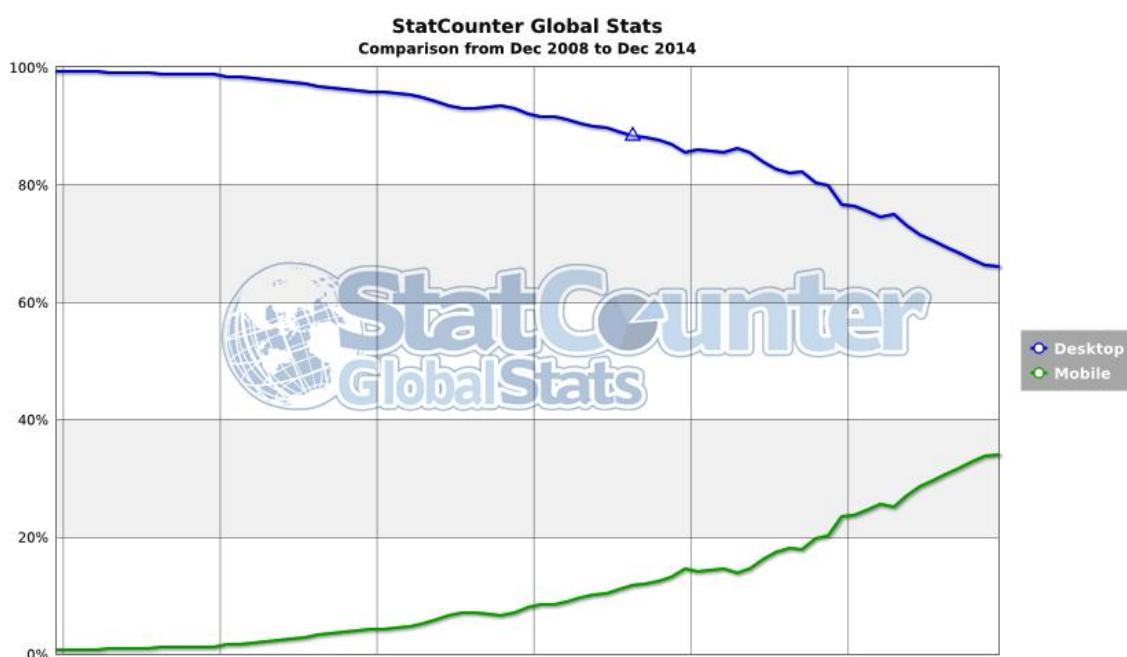
## 2 Zobrazovací zařízení a design webu

Vývoj mobilních zařízení v poslední době a jejich častější využívání k přístupu na internet, vedl k nutnosti definování nového přístupu k navrhování designu webu. Řešením má být responzivní design webu, který reaguje na rozmanitost dostupných zařízení. Následující podkapitoly představí trendy v oblasti mobilních zařízení. Pozornost je věnována jednotlivým částem responzivního designu, dále tomu, jaké technologie lze při jeho implementaci využít, a nakonec jsou uvedena různá doporučení, kterým je třeba při responzivním přístupu věnovat pozornost.

### 2.1 Současné trendy

Ještě donedávna byla většina rozlišení obrazovek buď 800x600 pixelů nebo 1024x768 pixelů. Díky tomu byl návrh stránek jednodušší v porovnání s velkou škálou rozlišení obrazovek a displejů na zařízeních, které přistupují na internet dnes [1].

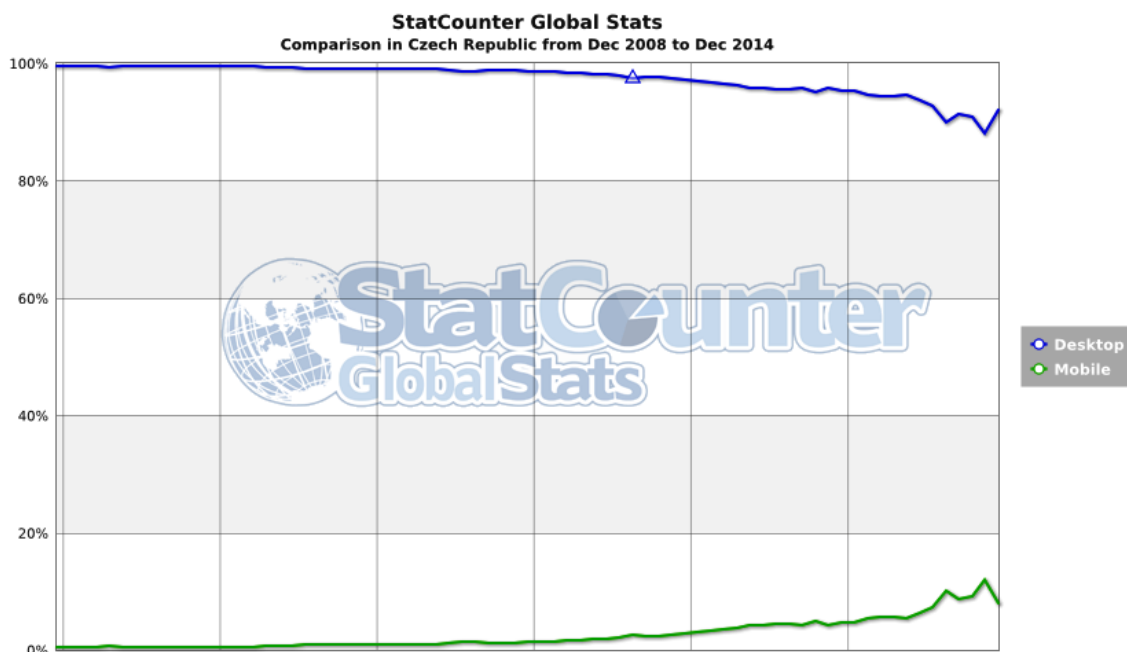
V Grafu 1 je vidět, že od roku 2008 pokračuje znatelný nárůst přístupu z mobilních zařízení oproti klasickým počítačům a notebookům. V celosvětové statistice bylo v lednu 2010 98,4 % přístupů z desktopů a 1,56 % přístupů z mobilních zařízení. Oproti tomu v prosinci roku 2014 bylo z desktopů zaznamenáno 65,97 % přístupů a z mobilních zařízení 33,3 % přístupů [2].



Graf 1 – trend přístupů z mobilů a tabletů 2008–2014, převzato z [2]

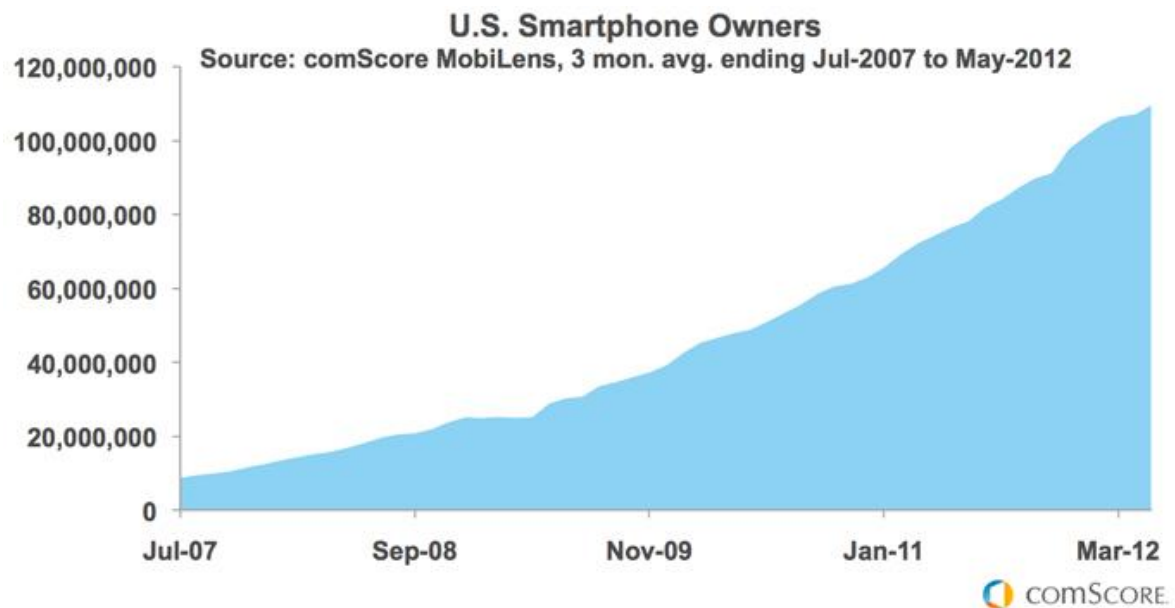


V případě statistiky přístupů pro Českou republiku je poměr počtu přístupů z mobilních zařízení k počtu přístupů z desktopů mnohem nižší. Dle společnosti StatCounter [3] bylo v prosinci 2014 zaznamenáno pouze 7,97 % přístupů na web z mobilních zařízení. Může to být způsobeno tím, že se v České republice trh s mobily a tablety rozvíjí s určitým zpožděním oproti například americkému trhu. Mobilní data v České republice byla dlouho dobu pro většinu uživatelů cenově nedostupná a pokrytí (kromě nejpomalejšího GPRS) také nebylo velké. Vzestupná tendence je však v Grafu 2 také patrná.



**Graf 2 – trend přístupů z mobilů a tabletů v České republice 2008–2014, převzato z [3]**

Nárůst počtu používaných mobilních zařízení pro přístup na internet na úkor stolních počítačů a notebooků odstartovalo v roce 2007 představení Apple iPhone, který ukázal nové pojetí chytrého telefonu zahrnující například velký kapacitní displej ovládaný gesty prstů [4]. Tím byl zcela změněn trh s chytrými telefony, který se začal rychle rozvíjet (Graf 3). Mobilní telefony série iPhone a jim podobná zařízení ostatních výrobců začaly nabízet mnohem pohodlnější přístup na internet a dotykové ovládání pomocí prstů.



Graf 3 – nárůst počtu majitelů chytrých telefonů v USA 2007–2012, převzato z [5]

## 2.2 Mobilní zařízení

V návaznosti na rostoucí počet mobilních zařízení a jejich možnosti přístupu na internet se objevil problém se zobrazováním webových stránek na malém rozlišení displejů těchto zařízení. Webové stránky nebyly připraveny a optimalizovány pro tak velký počet různých rozlišení. Především nebyly navrženy pro tak malé zobrazovací plochy.

Dříve byla běžná praxe optimalizovat webovou stránku pouze pro několik rozlišení. Přičemž byla pokryta většina návštěvníků webových stránek. Například dle statistiky webu W3Counter [6] a StatCounter [7], mělo v březnu roku 2009 více jak 70 % návštěvníků webových stránek rozlišení široké 1024 nebo 1280 pixelů.

Jak ale zajistit optimalizaci webové stránky pro více různých rozlišení? Co když bude za rok nebo dva na trhu představeno zařízení s novým rozlišením? Teoreticky by tak bylo potřeba počítat s jakýmkoliv možným rozlišením.

Postupně se objevily dva přístupy, jak se s optimalizací webů pro mobilní zařízení vypořádat. První volbou je vytvořit samostatnou verzi webové stránky pro mobilní zařízení, druhou volbou je responzivní design.

Výhodou i nevýhodou samostatné verze pro mobily je oddělená správa těchto webů. Ben Frain [8] poukazuje na to, že v určitých situacích, pokud to rozpočet dovolí, může být mobilní verze webové stránky správnou volbou. Tato aplikace totiž může zobrazovat odlišný obsah a design než webová stránka. Má také mnohem větší potenciál v podobě možnosti interakce s uživatelem založené na konkrétním zařízení, aktuální

poloze a dalších technických možnostech daného zařízení. Jako příklad uvádí řetězec pizza restaurací, který může mít „standardní“ webovou stránku a k tomu „mobilní“ verzi, která obsahuje navíc funkci vyhledávání nejbližší restaurace za pomoci rozšířené reality založené na aktuální GPS poloze. Toto řešení přesahuje hranice responzivního designu.

Z existence více oddělených webů také vyplývá nutnost správy více projektů. V případě nějakého nového prvku je nutná úprava ve všech verzích webu. Také vyvstává otázka, jakou verzi zobrazovat uživatelům tabletů. Standardní verze se na menším rozlišení tabletu většinou neobejde bez horizontálního posuvníku, mobilní verze je zase zbytečně jednoduchá, protože je určena pro mnohem menší displeje mobilních telefonů.

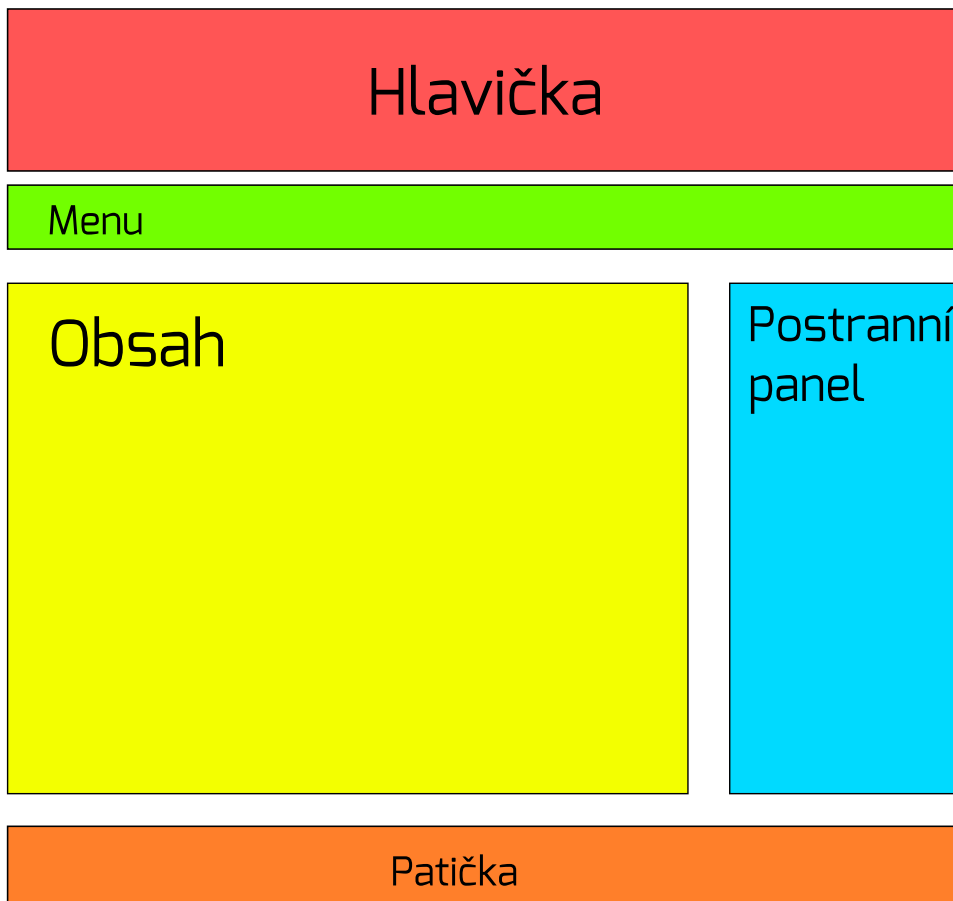
Implementace responzivního designu neznamena jen vytvoření webu pro mobilní zařízení, ale obecně jde o to, vytvořit takový web, který se přizpůsobí jakémukoliv zařízení [9]. Na rozdíl od předchozího řešení (více oddělených webů) odpadá správa několika projektů, kde se funkcionality duplikuje. Samotný vývoj responzivního designu není tak rychlý, protože se jedná o komplexní řešení, které musí nebo by mělo pokrývat všechna dostupná zařízení.

Responzivní přístup k designu webu bude podrobněji vysvětlen a popsán v následujících podkapitolách.

### **2.3 Rozvržení designu webu**

Design libovolného webu lze většinou rozdělit na několik samostatných funkčních celků (Obrázek 1). Tyto prvky mají na stránce určené místo většinou podle své priority a také podle konceptu designu celého webu.

Pro libovolný návrh rozvržení jednotlivých prvků webu (layout) se vychází z velikosti zobrazovací plochy prohlížeče, kde bude web vykreslen. Při návrhu takového rozvržení je stěžejní vždy šířka zobrazovací plochy, protože uživatel čte obsah shora dolů. Jelikož může být nabízený obsah různě dlouhý, tak v případě, že je delší než plocha, která je pro vykreslení webu k dispozici, zobrazí webový prohlížeč vertikální posuvník. Analogicky pokud se layout nevejde na šířku zobrazovací plochy, prohlížeč zobrazí horizontální posuvník. Ten je však ve většině případů nežádoucí, protože posouvat web vertikálně i horizontálně není pro uživatele komfortní. Aby bylo možné zobrazit web bez horizontálního posuvníku, je třeba znát šířku zobrazovací plochy. Šířka rozvržení webu by měla být vždy menší než je zobrazovací plocha prohlížeče uživatele.



Obrázek 1 – příklad layoutu webu rozděleného na několik funkčních celků

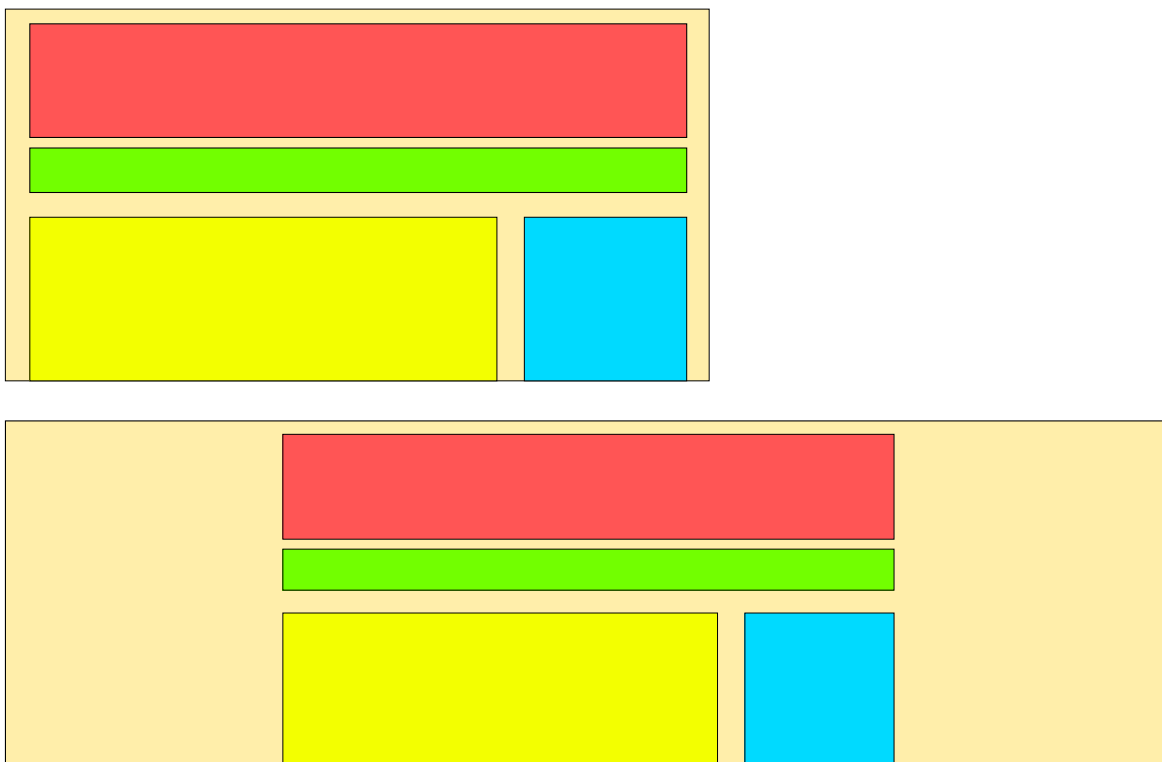
K designu webu lze přistupovat různým způsobem. Podle Tima Kadlece [10] je možné definovat čtyři typy rozvržení webu: statické, flexibilní, elastické a kombinované.

### 2.3.1 Statické rozvržení

Statické rozvržení webu má definovanou pevnou šířku pro všechny prvky layoutu. Šířka je definována absolutně pomocí jednotek pixelů. Znamená to, že šířka jednotlivých prvků je stejná pro všechna rozlišení (Obrázek 2).

Tim Kadlec ve své knize *Implementing Responsive Design* [10] zmiňuje hlavní problém statického rozvržení. Při návrhu takového layoutu je potřeba správně určit šířku zobrazovaného webu tak, aby vyhovovala většině návštěvníků. V tom je ale kámen úrazu. Dokonce i před nástupem mobilních zařízení a tabletů nebyl tento úkol jednoduchý. Různé velikosti monitorů znamenaly různé rozlišení a tedy i jinou šířku okna prohlížeče, ve kterém se stránka vykresluje. Navíc uživatel nemusí mít prohlížeč maximalizovaný na celou plochu monitoru nebo může používat nějakou boční lištu v prohlížeči, která zobrazovací plochu také o něco zúží. To znamená, že i v rámci jednoho rozlišení může

mít uživatel k dispozici jiný počet pixelů pro zobrazení webové stránky. V současné době se tento problém ještě více prohlubuje díky velké škále mobilních zařízení. V těch je zobrazení většinou řešeno tak, že mobilní prohlížeč zobrazí stránku zmenšenou a uživatel si pomocí gesta přibližuje jednotlivé části stránky. Toto řešení má však jistě daleko od kvalitního prožitku z prohlížení webu.

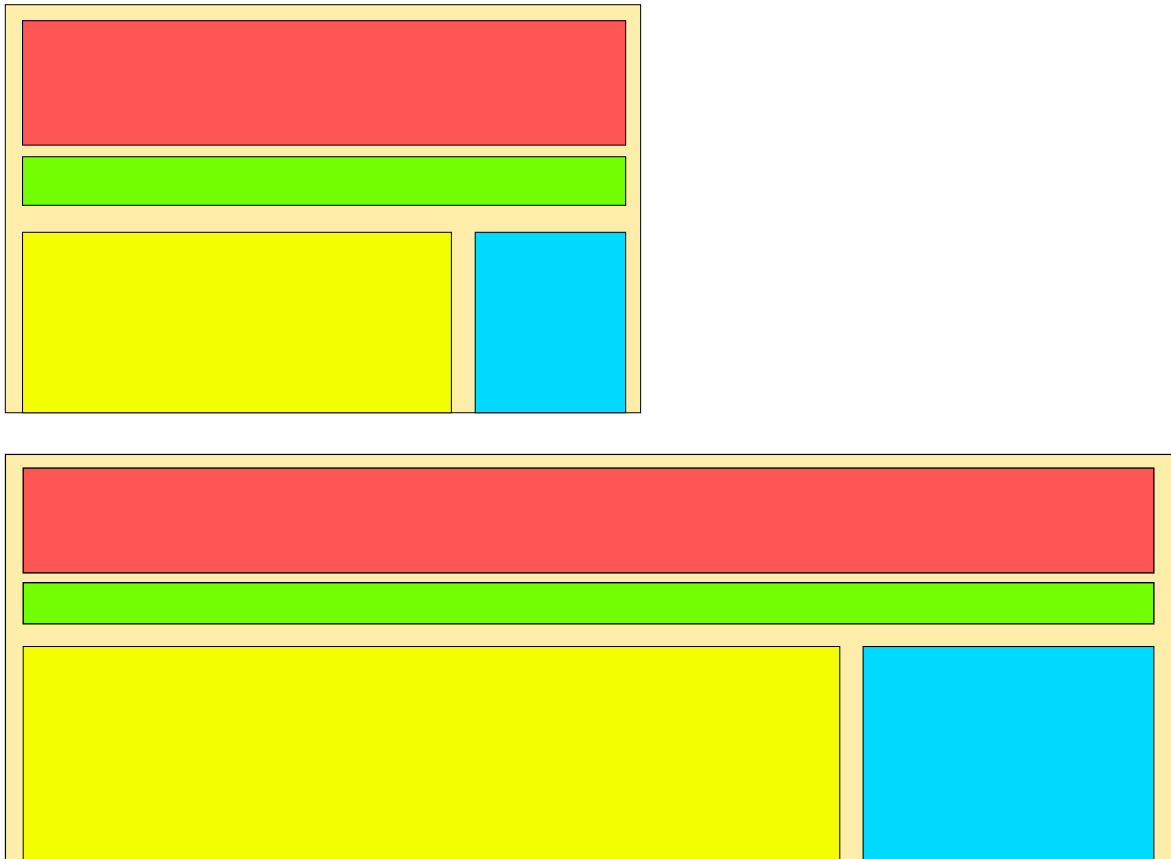


**Obrázek 2 – ukázka statického rozložení webu**

Pokud by byla stanovena optimální šířka pro všechny návštěvníky daného webu, byl by tento způsob rozvržení správnou volbou, protože zajišťuje kontrolu nad všemi prvky, které jsou zobrazeny, a také je nejjednodušší na implementaci. Není zde žádný dynamický prvek a vše je zobrazeno přesně tak, jak bylo navrženo a definováno.

### **2.3.2 Flexibilní rozvržení**

Flexibilní rozvržení má definovanou šířku jednotlivých prvků relativně k aktuální šířce zobrazovací plochy. Šířka není definovaná pomocí pixelů, ale pomocí relativních procent. 100 % představuje šířku zobrazovací plochy, vše ostatní je relativní právě k této šířce. Proto je šířka jednotlivých prvků pro každé rozlišení různá (Obrázek 3).



**Obrázek 3 - ukázka flexibilního rozložení webu**

Flexibilní layout už není závislý na šířce zobrazovací plochy, protože jsou všechny prvky definovány relativně. Pouze tento přístup však není dostačující pro všechna rozlišení. Uživatel narazí na problém příliš širokého textu, který se špatně čte v případě, že bude mít hodně široký monitor. Naopak na úzké zobrazovací ploše prohlížeče mobilního telefonu bude několika sloupcový design nepoužitelný, protože bude každý sloupec příliš úzký [10].

### **2.3.3 Elastické rozvržení**

Elastické rozvržení je podobné flexibilnímu, jen není definováno relativně k aktuální šířce zobrazovací plochy, ale relativně k velikosti písma na stránce. Šířka jednotlivých prvků je definovaná pomocí relativních *jednotek em*<sup>1</sup>. Jedna *jednotka em* odpovídá

---

<sup>1</sup> Název této jednotky vychází z písmena „M“, které bylo dříve používáno pro určení velikosti písma, protože je největší ze všech písmen. Dnes je *jednotka em* používána pro stanovení velikosti čehokoliv, co souvisí s aktuální velikostí písma, z které *jednotka em* vychází [8].

aktuální velikosti písma. Pokud má tělo stránky nastavenou velikost písma 16 pixelů, pak dvě *jednotky em* odpovídají 32 pixelům.

Výhodou tohoto rozvržení je, že lze přesně kontrolovat počet slov na řádku [10]. Více o tom, jaká by měla být správná délka řádku, je popsáno v podkapitole 2.4.4. Pokud si uživatel v prohlížeči zvětší nebo zmenší písmo na stránce, jednotlivé prvky se tomu přizpůsobí, protože jejich velikost se odvíjí od velikosti písma.

Bohužel pouze pomocí toto přístupu se nelze vyhnout nechtěným horizontálním posuvníkům, protože šířka jednotlivých prvků nijak nesouvisí s aktuální velikostí zobrazovací plochy [10].

### **2.3.4 Kombinované rozvržení**

Kombinovaný layout představuje spojení několika předešlých přístupů. Typicky jde o jeden fixní sloupec, který je vždy stejně široký. Tento požadavek vzniká například, pokud je v něm umístěná reklama, která není responzivní a zobrazuje se vždy ve stejné velikosti. Zbytek obsahu webu je pak definován pomocí procent [10].

I v případě kombinovaného rozvržení se může objevit nechtěný horizontální posuvník. Pokud je fixní sloupec definován s šířkou 300 pixelů a druhý sloupec s šířkou 70 %, pak se obsah nevejde na zobrazovací plochy s celkovou šířkou menší než 1000 pixelů. Řešením je využití vlastností HTML tabulky v CSS. Tato technika umožňuje vytvořit jeden sloupec (nebo více sloupců) s fixní velikostí. Zbytek obsahu se přizpůsobí zbývajícím šířce zobrazovací plochy [10].

## **2.4 Responzivní design webu**

Zajištěním přívětivého prostředí a snadného zobrazení informací pro uživatele používajícího libovolné zařízení se zabývá responzivní design webu. Termín responzivní webdesign (anglicky *responsive webdesign*) poprvé použil v roce 2010 Ethan Marcotte ve svém článku „Responsive Web Design“ [11], kde popisuje stále rostoucí počet různých zařízení přistupujících na internet. Řešením podle něj není vytvářet web na míru pro každý druh zařízení zvlášť, ale design, který se bude přizpůsobovat – responzivní design webu. Ten definoval jako použití *media queries*, responzivních obrázků a plovoucí mřížky (anglicky *fluid grid*).

V této podkapitole jsou popsány tři základní principy responzivního designu tak, jak je definoval Ethan Marcotte [11]. Kromě těchto tří hlavních prvků lze nalézt i další, jako je například typografie [12], které je v této podkapitole také popsána.

### 2.4.1 Media queries

*Media queries* je vlastnost CSS, která byla definována ve specifikaci CSS3 [13]. Pomocí *media queries* je možné definovat jednotlivé CSS styly pouze pro konkrétní šířku zobrazovací plochy nebo i pro další vlastnosti jako je rozlišení, velikost displeje nebo jeho orientace. Díky tomu je možné na různých zařízeních zobrazit odlišný design webu s jiným uspořádáním prvků.

Obecná definice vypadá takto: `@media typ_zařízení and (vlastnost: hodnota)`.

Typ zařízení se používá především pro rozlišení mezi standardním zobrazením `screen` a zobrazením pro tisk `print`. Vlastnost a její hodnota se vztahuje k zařízení, na kterém je webová stránka zobrazována. Seznam vybraných vlastností, které je možné použít, je uveden v Tabulce 1. U některých vlastností je možné použít prefix `min-` nebo `max-`, který určuje minimální nebo maximální hodnotu [14].

Název vlastnosti	Popis vlastnosti	Lze použít s prefixem min/max
<b>width</b>	Šířka zobrazovací plochy displeje zařízení (typicky šířka okna prohlížeče)	ANO
<b>height</b>	Výška zobrazovací plochy displeje zařízení (typicky výška okna prohlížeče)	ANO
<b>device-width</b>	Šířka displeje zařízení (typicky šířka monitoru)	ANO
<b>device-height</b>	Výška displeje zařízení (typicky výška monitoru)	ANO
<b>orientation</b>	Otočení displeje na výšku (portrait) nebo na šířku (landscape)	NE
<b>aspect-ratio</b>	Poměr mezi výškou (vlastnost width) a šířkou (vlastnost height) zobrazovací plochy	ANO
<b>device-aspect-ratio</b>	Poměr mezi výškou (vlastnost device-width) a šířkou (vlastnost device-height) displeje	ANO
<b>resolution</b>	Rozlišení displeje zařízení	ANO

Tabulka 1 – seznam vybraných vlastností media queries dle specifikace W3C [13]



V uvedeném příkladu (Ukázka kódu 1) je aplikován definovaný styl `font-size` v případě, že je zobrazovací plocha, na které je web vykreslen, minimálně 350 pixelů.

```
@media screen and (min-width: 350px) {  
    font-size: 1em;  
}
```

#### Ukázka kódu 1 – příklad zápisu *media queries* v CSS

*Media queries* lze použít přímo v CSS (Ukázka kódu 1). Potom se vztahuje k bloku CSS kódu, který následuje. Nebo je lze definovat v hlavičce HTML při importu CSS souboru (Ukázka kódu 2). Pak se vztahuje na všechny CSS styly v souboru obsažené [14].

```
<link rel="stylesheet" type="text/css" media="screen and (min-width:  
350px)" href="/css/style.css" />
```

#### Ukázka kódu 2 – použití *media queries* v hlavičce HTML souboru

Každý bod, ve kterém dojde ke změně vzhledu stránky a je aplikován styl uvnitř *media query*, je nazýván *breakpointem* [10].

Při tvorbě responzivního designu se vždy vychází z nějakého základního rozložení, buď je to desktopová verze, nebo nejjednodušší mobilní verze. Tyto dva přístupy platí i při tvorbě CSS stylů s použitím *media queries*. Vždy je nutné při výběru vhodného přístupu vzít v potaz konkrétní situaci. Obecně je doporučován přístup **mobile first** [8] [10] [14]. Ten spočívá v tom, že je vytvořen základní styl, který je použitelný pro jakékoliv zařízení, a to i takové, které nepodporuje *media queries* a JavaScript. Jedná se o základní layout pro mobilní zařízení. Tento styl je pak postupně obohacován a rozšiřován pomocí *media queries*. Díky nim může být zaručena minimální nutná šířka zobrazovací plochy. Tento přístup zajistí to, že na zařízeních, která nepodporují *media queries* ani JavaScript, bude zobrazena základní jednoduchá verze rozložení webu, která je určena pro jakékoliv rozlišení.

Kromě podpory všech zařízení má tento přístup ještě výhodu v tom, že většinou zkracuje a zjednodušuje složitost CSS [10].

Téměř všechny desktopové i mobilní verze internetových prohlížečů podporují *media queries*, kromě Internet Exploreru, který přidal podporu až od verze devět [15].

Podle statistik nástroje Google Analytics pro web [www.nasipolitici.cz](http://www.nasipolitici.cz) je návštěvníků, kteří používají starší verzi Internet Exploreru než je verze devět, kolem 15 %, což

není zanedbatelné procento uživatelů. Je proto potřeba v těchto prohlížečích zajistit přijatelné zobrazení webu [12].

Pokud je použit přístup *mobile first*, tak se uživatelům starších verzí Internet Exploreru zobrazí základní verze webu určená pro mobilní zařízení, i když si web prohlíží na desktopu. Tento problém lze vyřešit tak, že je vytvořen speciální soubor s CSS styly pro desktopové rozložení designu webu, které jsou určeny pouze pro starší verze Internet Exploreru. Tento soubor lze připojit do hlavičky HTML stránky a obalit ho speciálním podmíněným komentářem tak (Ukázka kódu 3), že si ho ostatní prohlížeče nebudou všimnout [10]. Způsob, jak lze soubor vytvořit je popsán v podkapitole 4.4.11.

```
<!--[if (lt IE 9)&(!IEMobile)]>
<link rel="stylesheet" href="style-old_ie.css" media="all"
type="text/css"> <![endif]-->
<!--[if (gt IE 8)|(IEMobile)]><!-->
<link rel="stylesheet" href="/css/style2.css" media="all"
type="text/css">
<!--<![endif]-->
```

#### **Ukázka kódu 3 – použití podmíněných komentářů k odlišení CSS stylů pro starší verze Internet Exploreru**

Další možností je využít JavaScriptovou knihovnu Respond.js (dostupné z <https://github.com/scottjehl/Respond>) nebo knihovnu css3-mediaqueries.js (dostupné z <https://code.google.com/p/css3-mediaqueries-js>). Pomocí nich lze přidat chybějící podporu *media queries* v prohlížečích, které je nepodporují [16].

#### **2.4.2 Flexibilní obrázky**

V responzivním designu webu je u většiny obrázků žádoucí, aby se jejich velikost přizpůsobovala velikosti zobrazené stránky. Pokud má obrázek (element `img`) v HTML nastavenou vlastnost `width` a `height`, potom se vykreslí vždy ve stejné velikosti podle zadaných hodnot těchto vlastností. Obrázek by pak mohl být v mobilním zobrazení příliš velký nebo v desktopovém zobrazení zase zbytečně malý. Řešením je vlastnosti `width` a `height` v HTML nenastavovat a velikost řídit pomocí CSS [10]. Příklad takového nastavení je uveden níže (Ukázka kódu 4).

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

#### Ukázka kódu 4 - dynamická šířka obrázku pomocí CSS

Nastavení vlastnosti `max-width` a `width` na hodnotu „100%“ říká obrázku, aby se roztáhnul na celou šířku HTML elementu, do kterého je vložen a zároveň nepřesáhl jeho šířku [10]. Obrázek ale musí být vždy dostatečně velký, protože se vždy roztáhne na celou šířku nadřazeného HTML elementu. Na druhou stranu na malém displeji je zobrazen zmenšený obrázek, ale prohlížeč stejně načítá obrázek v původní velikosti. Zbytečně to prodlužuje dobu načítání stránky a stahuje data navíc. Responzivním řešením je mít obrázek v několika velikostech a nechat prohlížeč stáhnout jen jeden obrázek v potřebné velikosti. K tomu lze využít některé techniky, které jsou podrobněji popsány dále v této podkapitole. Druhou možností je zobrazit obrázek, jen pokud je k dispozici dostatečně velká zobrazovací plocha, a v ostatních případech ho nahradit textovou reprezentací nebo třeba tabulkou. Toto řešení je dobré použít v případě, že nemá význam zobrazovat obrázek zmenšený. Možný způsob implementace je popsán v podkapitole 4.4.6.

Řešení **Adaptive Images** (dostupné z <http://adaptive-images.com>) zajišťuje pomocí JavaScriptu doručení správné velikosti obrázku podle aktuální velikosti zobrazovací plochy. Na základě zjištěné velikosti upraví velikost zdrojového obrázku, uloží ji do předem určené složky, aby nebylo nutné generování znovu opakovat. Prohlížeč odešle požadavek na obrázek a dostane jako odpověď upravený obrázek ve správné velikosti.

Výhodou řešení Adaptive Images je fakt, že není potřeba nijak upravovat stávající strukturu HTML nebo kód. Proto je toto řešení vhodné pro již existující rozsáhlé weby, kde by byla úprava kódu napříč všemi stránkami časově náročná. Pomocí Adaptive Images už ale nelze docílit toho, aby se pro určitá rozlišení zobrazoval například jen výřez z daného obrázku [10].

Technologie Adaptive Images je pouze řešením, které nahrazuje absenci standardu pro doručení obrázku do prohlížeče v různých velikostech v závislosti na používaném

zařízení. Postupně se však ustálilo řešení, které se stalo součástí HTML specifikace [17]. Jedná se o rozšíření elementu `img` a přidání elementu `picture`.

**Rozšířením elementu `img`** byly přidány dva nové atributy. První atribut je `srcset`, ve kterém lze definovat cestu k různým velikostem obrázků. Druhým atributem je `sizes`, kde lze definovat, jak má být obrázek široký. Šířku lze definovat buď absolutně, tedy například 200 pixelů, nebo relativně vzhledem k aktuální šířce zobrazovací plochy. Šířku lze také definovat odlišně pro různé *breakpointy*.

Více možností než element `img`, nabízí **nový element `picture`**. Na rozdíl od elementu `img`, který v atributu `srcset` obsahuje sadu různých velikostí jednoho obrázku, umožňuje element `picture` definovat více takových setů pro odlišné obrázky. Odlišnými obrázky se myslí například jiný poměr stran nebo výřez z původního obrázku. Nejedná se o jiný obrázek, ale pouze modifikovaný jiným způsobem, než jen změnou velikosti [18].

V obou popsáných případech si na základě poskytnutých informací prohlížeč sám vyhodnotí, jaký zdroj obrázku vybrat a jak velký ho vykreslit. Výhodou je, že rozhodování o tom, který obrázek zobrazit, je přenecháno prohlížeči, který by měl mít dostatek informací potřebných k takovému rozhodnutí. Je to například velikost zobrazovací plochy, rozlišení nebo rychlost připojení. Díky tomu, že není potřeba se starat o rozhodovací logiku, je implementace poměrně snadná [12].

Podpora v atributu `srcset` a elementu `picture` zatím není velká, ale chystané nové verze prohlížečů kromě Internet Exploreru a Opery Mini by již měli tuto novou specifikaci podporovat [19].

Absenci podpory v prohlížečích lze naštěstí řešit pomocí JavaScriptové knihovny `Picturefill` od Scotta Jehla (dostupné z <http://scottjehl.github.io/picturefill>). Do budoucna se počítá s postupným zavedením podpory a knihovna by tak již neměla být potřeba.

**Obrázky na pozadí stránky** jsou definovány pomocí CSS, díky tomu lze použít *media queries*. Prohlížeč stáhne velký obrázek jen v případě, že je opravdu potřeba. Je možné také poskytnout lepší obrázek pro obrazovky s vysokým rozlišením. Doporučuje se použít větší rozlišení obrázku, který je ale vykreslen zmenšený tak, aby v jednom bodě bylo zobrazeno více pixelů. Na obrazovkách s vysokým rozlišením je potom obrázek vnímán jako ostřejší [12].

### 2.4.3 Mřížka

Použití mřížky v grafickém designu a typografii je zcela běžná praktika. Na poli designu webu to také není žádná novinka. V obou těchto případech se pracuje s fixními rozměry stránky, nebo alespoň s pevnou šířkou, pokud se jedná o web. Mřížka má definovaný počet sloupců, případně i řádků, a pevně nastavenou jejich šířku. Do mřížky se pak obsah vkládá tak, aby byl vždy zarovnán k nějakému sloupci nebo řádku [14].

V případě responzivního designu je nutné počítat s libovolnými rozměry stránky, proto je použití mřížky trochu specifické. Mřížka má pevně definovaný počet sloupců, jejich šířka je proměnlivá, protože se mění i šířka stránky. Obsah je vkládán do mřížky a společně s ní se přizpůsobuje velikosti zobrazovací plochy.

Design s pomocí mřížky má několik nesporných výhod. Zajišťuje, že přidání nového prvku nebo obsahu je snazší. Dále nastavuje určitou základní koncepci designu, které je možné se držet. Také uživatel se v mřížce lépe orientuje, je pro něj snazší odhadnout, kde najít hledaný obsah [20].

Na použití mřížky je založena většina responzivních *frameworků*. Obvykle se jedná o mřížku s 12 sloupci. Responzivní *frameworky* jsou popsány dále v podkapitole 2.6.

Tim Kadlec [10] poukazuje, že na použití mřížky o 12 sloupcích není nic špatného, ale protože je hojně využívána, nedává prostor pro vznik zajímavého, neokoukaného layoutu. Doporučuje počet sloupců vždy pro každý nový layout pečlivě promyslet a nebát se použít klidně tři nebo pěti sloupcovou mřížku.

### 2.4.4 Typografie

Typografie neodmyslitelně patří k webovému designu, protože textový obsah je nedílnou a většinou i hlavní součástí webových stránek. V responzivním designu je třeba správné typografii věnovat zvláštní pozornost. Velikost plochy, na které je text vykreslen, se mění, protože je webová stránka zobrazována na různých zařízeních. Proto je nezbytné měnit velikost a způsob zobrazení textu tak, aby se uživateli četl co nejpohodlněji.

Pro text zobrazený na webové stránce jsou nejdůležitější tři parametry: počet slov na jednom řádku, velikost písma a velikost řádkování [21]. Nelze obecně říct, jaké jsou správné hodnoty pro jednotlivá rozlišení. Existuje však několik obecných pravidel

a doporučení, kterými se lze řídit. Vždy je to však otázka zkušenosti, osobního vkusu a určitého kompromisu.

Aby se text četl dobře, mělo by na jednom řádku být 45 až 75 znaků včetně mezer a interpunkčních znamének [22]. Laura Franz [21] doporučuje na základě vlastní zkušenosti pro textový obsah na webu až 85 znaků. Pokud je znaků více, je možné písmo zvětšit, tak aby bylo na jednom řádku méně znaků. Jak bude zmíněno dále, ne vždy je dobré toto pravidlo přesně dodržovat. Platí však, že znaků může být i méně než 45, ale více než 85 by jich být nemělo.

Obecně text vypadá menší na mobilních zařízeních než na monitorech notebooků a stolních počítačů. Na druhou stranu platí, že čím je displej menší, tím na něj uživatel dívá z menší vzdálenosti. Přestože na mobilních zařízeních sledují uživatelé displej z menší vzdálenosti, je pohodlnější číst o něco větší písmo i za cenu toho, že není dodržena ideální délka řádku, která tak může být menší než 45 znaků [21].

Na velkém rozlišení je lepší zúžit sloupec s textem a zachovat rozumnou velikost písma a délku řádků. Při snaze zaplnit celou šířku zobrazovací plochy by totiž muselo být písmo přehnaně velké, aby byl dodržen správný počet znaků na řádek. Proto není žádoucí zaplňovat textem celou šířku obrazovky [21].

Kromě počtu slov na řádku a velikosti písma lze ještě měnit velikost řádkování. Jeho velikost se doporučuje zvolit jako 150 % velikosti písma. Na malých displejích mobilních zařízení se doporučuje řádkování ještě o něco zvětšit [21].

## **2.5 CSS preprocesory**

CSS, tedy kaskádové styly, neodmyslitelně patří k tvorbě popisu vzhledu webové stránky. Jednoduchost kaskádových stylů s sebou nese nevýhody v podobě dlouhého kódu, který je nepřehledný a špatně se udržuje. Proto vznikly různé nadstavby a preprocesory, které do kaskádových stylů zavádějí prvky a vlastnosti z programovacích jazyků. Jde například o proměnné, funkce nebo výrazy [23].

Často používané jsou preprocesory LESS a SASS. Soubor napsaný v LESS nebo SASS je pak nějakým překladačem zkompileován do CSS souboru. Výsledný soubor s kaskádovými styly je použit stejně, jako kdyby k jeho napsání nebyl využit žádný preprocesor. Jde tak vlastně o usnadnění práce vývojářům. Uživatel nijak nepozná, že byla použita nějaká nadstavba CSS.

LESS i SASS mají hodně vlastností podobných, jen mají různý styl zápisu. Přesto se v některých věcech liší a některé vlastnosti jsou v jednom preprocesoru jednodušší k použití než v druhém. Při výběru vhodného preprocesoru záleží na konkrétním použití a osobní preferenci.

Dostupných vlastností a možností použití preprocesorů je mnoho, proto zde bude uveden jen krátký popis vybraných vlastností, které samotné CSS nenabízí.

### **2.5.1 Proměnné, mixins**

Možnost definice proměnné znamená, že je hodnota nějaké vlastnosti uložena jen na jednom místě. Typicky se jedná o hodnotu barvy nebo *breakpointu*. Na všech ostatních místech se používá pouze reference na danou proměnnou. Následná změna hodnoty znamená změnu jen na jednom místě.

Stejně tak, jako je možné definovat proměnné, je možné definovat na jednom místě několik vlastností a jejich hodnot. Definice takové skupiny hodnot se nazývá *mixin*. Tuto skupinu je možné si pojmenovat a pomocí reference ji použít na více místech v kódu [24] [25]. V případě změny pak opět stačí upravit kód na jednom místě. Použití *mixin* je ukázáno v podkapitolách 4.4.2 a 4.4.11.

### **2.5.2 Import souborů a validace**

Pokud rozdělíme CSS do více souborů, je potřeba tyto soubory samostatně definovat v HTML stránce. Každý soubor s kaskádovými styly navíc znamená další request prohlížeče. LESS i SASS preprocesory umožňují definovat jeden i více souborů, které budou do aktuálního souboru při kompilaci importovány [24] [25]. Import je také možné řetězit. Díky tomu je pak možné udržovat kód ve více souborech, ale výsledný kaskádový styl je jen jeden soubor.

Při kompilování LESS nebo SASS souboru do CSS je provedena kontrola syntaxe. Vývojář je díky tomu upozorněn na časté chyby, kterými jsou například chybějící středníky nebo závorky.

### 2.5.3 Zanořování prvků

V kaskádových stylech do sebe nelze jednotlivé prvky zanořovat a vytvářet tak logickou stromovou strukturu. Možnost zanořování jednotlivých prvků přináší větší přehlednost, znamená méně kódu a lépe odpovídá stromové struktuře HTML (Ukázka kódu 5).

#### CSS

```
#page #topMenu {
  width: 100%;
}
#page #topMenu.nojs #collapse-menu {
  display: none;
}
#page #topMenu.nojs ul {
  display: block !important;
}
#page #topMenu.nojs #socialNetworks {
  display: block;
}
#page #topMenu #collapse-menu {
  float: right;
}
#page #topMenu #collapse-menu:after {
  content: "";
}
@media (min-width: 990px) {
  #page #topMenu #collapse-menu {
    display: none;
  }
}
```

#### LESS

```
#page #topMenu {
  width: 100%;
}
&.nojs {
  #collapse-menu {
    display: none;
  }
  ul {
    display: block !important;
  }
  #socialNetworks {
    display: block;
  }
}
#collapse-menu {
  float: right;
}
&:after {
  content: "";
}
@media (min-width: 990px) {
  #collapse-menu {
    display: none;
  }
}
```

Ukázka kódu 5 – příklad stejného kódu napsaného v CSS a LESS

## 2.6 Responzivní frameworky

Responzivní *frameworky* usnadňují tvorbu responzivního designu. Obsahují připravené styly pro vytvoření designu pomocí mřížky a také různé připravené moduly, jako je navigace, obrázková galerie, tabulky a další. Tyto moduly lze jednoduše vzít a použít. Přizpůsobení jejich vzhledu je také snadné. Výhodou také je, že *frameworky* používají zavedené a odzkoušené postupy.



Důvodem, proč využít *framework* bude nejčastěji snížení času potřebného pro vytvoření webu. Responzivní *framework* lze využít pouze pro vytvoření rychlého prototypu nebo jako základ budoucího webu, ale také pro kompletní tvorbu celého responzivního designu webu [26].

Protože responzivních *frameworků* existuje celá řada, jsou zde popsány pouze dva patrně nejznámější zástupci [26].

**Bootstrap** (dostupné z <http://getbootstrap.com>) byl vytvořen týmem Twitteru. Jedná se o robustní *framework*, který obsahuje velmi mnoho různých komponent, jako jsou různé druhy navigace, tlačítek, tabulek a mnoho dalších. Jeho výhodou je to, že má velkou uživatelskou základnu [1]. Je založený na 12 sloupcové mřížce a zdrojové soubory jsou k dispozici v LESS, který lze pomocí proměnných snadno upravit.

**Framework Foundation** (dostupné z <http://foundation.zurb.com>) stejně jako Bootstrap používá mřížku o 12 sloupcích. Je založený na preprocesoru SASS, zdrojové soubory lze díky tomu také snadno přizpůsobovat. Jako jeho výhoda je uváděno [1] použití více sémantických názvů tříd. Například pro vytvoření elementu `div` se dvěma sloupci používá Bootstrap třídu `col-md-2`, Foundation používá třídy `medium-2` a `columns`.

## 2.7 Doporučení pro tvorbu responzivního webu

V této podkapitole jsou uvedena různá doporučení a tipy zkušených tvůrců responzivních webů.

### 2.7.1 Určení breakpointů

Pro určení umístění jednotlivých *breakpointů* by se nabízelo vycházet z nejběžnějších šířek zobrazovacích zařízení používaných v současné době. Je však doporučováno, aby *breakpointy* byly stanoveny podle designu webu [1] [16] [12]. A to tak, že se postupuje od rozložení pro nejmenší (nebo největší) rozlišení, postupně se rozlišení zvětšuje (zmenšuje). Ve chvíli, kdy už rozložení webu nevypadá dobře, tak se tato šířka stane novým *breakpointem* [16]. Nemusí jít jen o rozložení jednotlivých prvků na stránce, ale i o jejich šířku, okraje, odsazení a také velikost textu.

*Breakpointů*, kde se mění uspořádání jednotlivých prvků na stránce, nebo kde se mění počet sloupců, obvykle nebývá mnoho. Tyto *breakpointy* by se daly nazvat jako hlavní. Mnohem více pak bývá menších *breakpointů*, kde se mění různá odsazení, velikost

textu, drobně upravuje šířka nějakého prvku [12]. Vždy je dobré se držet pravidla, že čím méně *breakpointů*, tím je snazší následná údržba.

### 2.7.2 Velikost písma pomocí relativních jednotek

Pro definování velikosti písma na webu lze použít *jednotky em*, pixely nebo procenta. Zatímco pixely vyjadřují velikost absolutně, *jednotky em* a procenta popisují velikost písma relativně k nadřazenému elementu. Vzniká zde kaskáda, která vychází ze základní velikosti písma definované na nejvyšším elementu v HTML stránce – elementu `body`. Toho lze v responzivním designu využít v situaci, kdy je potřeba celkově zvětšit nebo zmenšit písmo na celém webu pro určitý *breakpoint*. V takovém případě stačí upravit výchozí velikost písma v elementu `body`. Pokud by nebyla velikost textu definovaná v relativních jednotkách, ale v absolutních pixelech. Bylo by potřeba definovat novou velikost ve všech elementech, které mají nastavenou velikost textu [10].

Mezi procenty a *jednotkami em* není příliš velký rozdíl. Není žádný objektivní důvod proč preferovat jednu nebo druhou jednotku. Pro velikost textu je používanější *jednotka em*, pravděpodobně z toho důvodu, že vychází právě z velikosti textu [10].

Ať už budou pro velikost textu použity procenta nebo *jednotky em*, pro nastavení základní velikosti písma v elementu `body` je dobré použít procenta. Internetový prohlížeč Internet Explorer totiž špatně počítá velikost písma v *jednotkách em*, pokud je základní velikost v elementu `body` definovaná také v *jednotkách em*. Tomu se lze vyhnout použitím procent. Ve většině případů bude mít velikost písma v elementu `body` hodnotu „100%“, což ve většině prohlížečů odpovídá 16 pixelům [10].

### 2.7.3 Optimalizace pro velká rozlišení

Optimalizace pro velká rozlišení není tak důležitá jako pro malá rozlišení, protože design pro malá rozlišení je dobře použitelný i na velkých rozlišeních, obráceně to neplatí. Přesto by se na optimalizaci pro velká rozlišení nemělo zapomínat.

V první řadě je potřeba zachovat dobrou čitelnost textu zvolením vhodného počtu slov na řádku i za cenu toho, že bude na velkém rozlišení zobrazen v relativně úzkém sloupci. Na vzniklé volné místo po stranách lze přemístit méně důležitý obsah, který by jinak musel být až na konci stránky. Pokud je obsah založen na kolekci nějakých prvků, typicky obrázků, lze je zobrazit v dlaždicích, které mohou zaplnit celý prostor velké obrazovky [27].

#### **2.7.4 Optimalizace pro dotykové displeje**

Responzivní webová stránka by měla být optimalizovaná pro dotykové displeje. Bohužel nelze určit, jestli bude webová stránka zobrazována na zařízení s dotykovým displejem nebo ne. Malé rozlišení nemusí znamenat dotykový displej, i když tomu tak dnes ve většině případů díky chytrým telefonům je. Existují však i ultrabooky s dotykovým displejem, které kombinují funkci notebooku a tabletu. S takovým zařízením lze pracovat stejně jako s notebookem a displej ovládat pomocí myši, ale také lze odpojit klávesnici a používat dotykový displej jako tablet.

I kdyby bylo možné detekovat, jestli má displej, na kterém je webová stránka zobrazena, dotykovou funkci, uživatel nemusí obrazovku používat v dotykovém režimu. Ovládání webové stránky, která je optimalizovaná pro dotykové displeje na nedotykové obrazovce, je pro uživatele komfortní. Kdežto v obrácené situaci, kdy uživatel s dotykovým displejem ovládá stránku, která není optimalizovaná pro dotykové obrazovky, tomu tak není. Proto by měl být web vždy navržen pro dotykové displeje, i když je někteří uživatelé mít nebudou [12].

Dostačující velikost prvků pro pohodlné ovládání ukazováčkem je 45–75 pixelů, pro palec je to šířka 72 pixelů [28]. Neméně důležitým faktorem, jako je velikost dotykových prvků, je volný prostor kolem nich. Menší tlačítko, kolem kterého bude dostatek prázdného prostoru, se bude ovládat lépe než větší tlačítko, ke kterému je těsně umístěn další ovládací prvek [12].

#### **2.7.5 Jednotný přístup k webu napříč zařízeními**

Tim Kadlec [10] upozorňuje na to, že je potřeba k responzivnímu webu přistupovat jako k jednotnému webu. Častou chybou je rozdělovat design na web pro mobily, web pro tablety a web pro desktopy. Je potřeba si uvědomit, že uživatelé svá zařízení během dne střídají a pokud si otevřou stránku na jednom zařízení a později se k ní vrátí na jiném zařízení, měla by jim být povědomá, měli by se na ni dobře orientovat. Nikdy by neměla nastat situace, kdy bude uživateli na webu něco chybět, protože na to byl zvyklý z verze webu pro jiné zařízení. Přístup k responzivnímu webu by měl být napříč všemi zařízeními jednotný.

### 2.7.6 Správná šířka zobrazení

Na některých mobilních zařízeních neodpovídá výchozí šířka zobrazení jejich skutečné šířce [10] [14]. Například prohlížeč na mobilním zařízení s displejem širokým 320px může na stránku vykreslit zobrazení pro šířku 980px. Patrně je to proto, aby bylo možné zobrazit větší část webu, který nemá responzivní design, a ne jen malý výřez jeho části. V responzivním designu už toto chování není žádoucí. Pro správné použití *media queries* je potřeba znát skutečnou šířku zařízení. Popsanému chování některých prohlížečů lze zamezit přidáním `viewport` meta tagu do hlavičky HTML dokumentu (Ukázka kódu 6).

```
<meta name="viewport" content="width=device-width" />
```

#### Ukázka kódu 6 – meta tag viewport v hlavičce HTML dokumentu

Toto řešení funguje dobře, jde však pouze o dočasné řešení, které by mělo umožnit přechod na použití nového pravidla `@viewport` přímo v CSS – Ukázka kódu 7 [10].

```
@viewport {  
    width: device-width;  
}
```

#### Ukázka kódu 7 – použití pravidla viewport v CSS

Řešení v CSS však ještě není plně podporováno všemi prohlížeči, ale očekává se, že časem plně nahradí meta tag `viewport`.

## 2.8 Testování responzivního designu

Responzivní design by měl být dobře použitelný na jakémkoliv zařízení, proto musí být velká pozornost věnována nejen vývoji, ale i testování. Protože zařízení s přístupem na internet existuje opravdu hodně, není to zrovna jednoduchý úkol.

Ideální je testovat na skutečných zařízeních, ale zároveň je to velmi nákladné. Určitě nelze pokrýt všechna zařízení [12]. Proto pokud je to možné, je dobré vybrat alespoň nějaký reprezentativní vzorek skutečných zařízení, které nejčastěji používá skupina uživatelů, na které je web zacílený.

Dobrou volbou jsou také emulátory skutečných zařízení. Ta mají určitá omezení, protože rychlost zařízení a odezva nemusí odpovídat skutečnému zařízení. Chybí zde také fyzický kontakt se zařízením, ovládání tak nemusí být úplně stejné. Přesto je to

výborný zdroj informací o tom, jestli se na daném zařízení stránka zobrazí tak, jak bylo předpokládáno [12].

Poslední volbou, která však většinou přichází na řadu nejdříve, a to již v průběhu vývoje, jsou nástroje pro vývojáře integrované přímo do prohlížečů jako je například Chrome nebo Firefox. Takový nástroj umožňuje rychle přepínat mezi zobrazeními na různých zařízeních, která jsou definována svým rozlišením. Lze také simulovat událost vyvolanou dotykem prstu [12].

Pro usnadnění testování na různých zařízeních existují různé služby jako například BrowserStack (dostupné z <http://www.browserstack.com>), které umožňují vzdálené testování na reálných zařízeních, případně emulátorech [12].

To, jaké možnosti testování zvolit a využít, hodně záleží na dostupném rozpočtu a konkrétním projektu. Vždy je však důležité tuto část nepodceňovat.

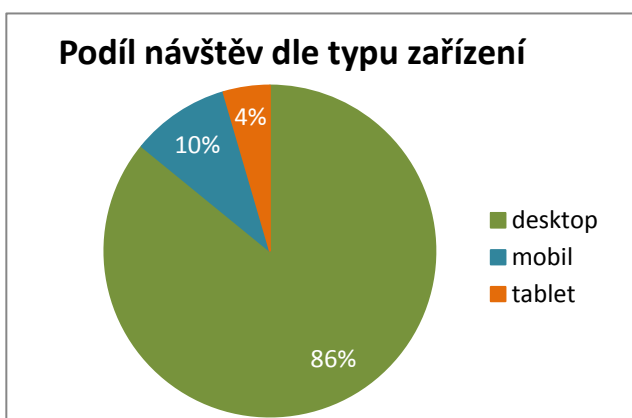
### 3 Analýza výchozího stavu webu Nasipolitici.cz

Na webovém portálu Nasipolitici.cz byl v roce 2011 nasazen nový design webu. Jeho rozložení bylo fixní, tedy mělo pevnou šířku a nebylo responzivní. Cílem nasazení responzivního designu webu bylo reagovat na aktuální trend dnešní doby, kterým je snaha přinést příjemnější zážitek z prohlížení webu na mobilních zařízeních. Oprávněnost této změny dokládají i statistiky návštěvnosti, které ukazují, že návštěvníci z mobilních zařízení na web chodí (viz následující podkapitola 3.1).

#### 3.1 Analýza návštěvnosti

Nástroj Google Analytics toho může hodně povědět o chování návštěvníků webu a o jejich zařízeních. Pomůže určit význam zavedení responzivního designu. Před jeho implementací poskytne informaci o tom, kolik návštěvníků na web přistupuje z mobilních zařízení a pomůže v rozhodování, na jakých zařízeních by bylo dobré vytvořený responzivní design testovat. Po nasazení responzivního designu je možné zkontrolovat, jestli to mělo vliv na chování uživatelů a jestli se zvýšila návštěvnost z mobilních zařízení.

Graf 4<sup>2</sup>, ukazuje, že většina uživatelů webu Nasipolitici.cz přichází z desktopových zařízení. Těch, kteří používají mobilní zařízení, je kolem 14 %. Dá se předpokládat, že po zavedení responzivního designu by se mohlo toto číslo zvýšit [10]. Přesto ani v současné době nejde o zanedbatelné procento uživatelů.



Graf 4 – podíl návštěv webového portálu Nasipolitici.cz z jednotlivých zařízení k celkovému počtu návštěv

---

<sup>2</sup> Data byla získána z nástroje Google Analytics webu Nasipolitici.cz za období 1. 8. 2014 – 3. 4. 2015

V následující tabulce, která obsahuje informace o návštěvnosti webového portálu Nasipolitici.cz<sup>3</sup>, jsou data rozdělena podle typu zařízení na desktop, mobil a tablet.

Kategorie zařízení	Míra okamžitého opuštění	Počet stránek na 1 návštěvu	Průměrná doba trvání návštěvy (s)
desktop	70,26 %	2,11	71,47
mobil	80,39 %	1,52	47,30
tablet	74,50 %	1,79	55,52
<b>Celkový průměr</b>	<b>71,42 %</b>	<b>2,04</b>	<b>68,44</b>

Tabulka 2 – přehled vybraných metrik rozdělených podle typu zařízení

**Míra okamžitého opuštění**, která vyjadřuje podíl návštěv, při kterých uživatel navštívil jen vstupní stránku a nepokračoval na žádnou další, je u mobilů i tabletů vyšší než u desktopů. Je možné předpokládat, že je to způsobeno absencí responzivního designu, kdy je především pro uživatele mobilů prohlížení webu nepohodlné. Pokud je tato domněnka pravdivá, tak se to po implementaci responzivního designu projeví snížením míry okamžitého opuštění u tabletů a mobilů.

Kromě absence responzivního designu může míra okamžitého opuštění souviset i s rozdílným důvodem používání jednotlivých zařízení, kdy například na mobilním zařízení uživatel častěji potřebuje vyhledat konkrétní informaci, a pokud ji dostane hned na vstupní stránce, nebude již na webové stránce déle setrávat. Přičemž vstupní stránka nemusí nutně znamenat domovskou stránku webu, protože se může uživatel přes vyhledávač dostat již na nějakou konkrétní stránku, která obsahuje jím hledanou informaci.

**Počet stránek na 1 návštěvu** je průměrný počet stránek, které si uživatel během jedné návštěvy zobrazí. Číslo je u desktopů vyšší než u tabletů a mobilů. Souvisí s předchozím ukazatelem, protože míra okamžitého opuštění znamená zobrazení jedné stránky na jednu návštěvu. Po implementaci responzivního designu by se i hodnota počtu stránek na jednu návštěvu mohla u tabletů a mobilů zvýšit, opět to ale může souviset s různým způsobem používání těchto zařízení.

Nejvyšší míru okamžitého opuštění a nejnižší počet stránek na jednu návštěvu mají mobily, naopak nejnižší míru okamžitého opuštění a nejvyšší počet stránek na jednu

---

<sup>3</sup> Data byla získána z nástroje Google Analytics webu Nasipolitici.cz za období 1. 8. 2014 – 3. 4. 2015

návštěvu mají desktopy. Tablety mají svou hodnotu vždy někde mezi. Mohlo by to být tím, že tablet má větší displej a rozlišení než mobil a je na něm pohodlnější prohlížet stránku, která nemá implementovaný responzivní design, protože není potřeba stránku tolik přibližovat a posouvat. Většina uživatelů používá tablet primárně z pohodlí domova [29], dá se tedy předpokládat, že budou trávit více času prohlížením stránek, než jen hledáním faktů.

**Průměrná doba trvání návštěvy** je průměrný počet sekund, který uživatel stráví prohlížením stránek při jedné návštěvě. U desktopů je čas výrazně delší než u mobilů a tabletů. Na jednu stranu je žádoucí udržet uživatele na webu co nejdéle. Hodnota by tak měla být co nejvyšší. Na druhou stranu, pokud uživatele zajímá pouze konkrétní informace, může být kratší čas známkou toho, že požadovanou informaci našel rychleji.

Analýzou dat pro web Nasipolitici.cz z Google Analytics bylo zjištěno, že implementace responzivního designu by mohla uživatelům webu pomoci. Tento předpoklad by však bylo potřeba ověřit dalšími průzkumy a testováním uživatelského chování, to je však již nad rámec této práce.

### **3.2 Výchozí technické řešení webu**

Implementace responzivního designu na webovém portálu Nasipolitici.cz znamenala úpravu původního řešení. To nebylo nahrazeno zcela novou implementací, ale byly provedeny pouze nutné změny pro zavedení responzivního designu.

Ve výchozím technickém řešení *front-endu* webu Nasipolitici.cz jsou použity JavaScriptové knihovny jQuery 1.5.1, jQuery UI 1.8.11, jCarouselLite, jsTree 1.0 a několik dalších knihoven, které nejsou pro tuto práci podstatné. Kaskádové styly jsou psány přímo v CSS, není využit žádný preprocesor. Statické stránky jsou napsány v HTML, dynamické jsou generovány pomocí šablonovacího systému Smarty. V šablonách je obsažen zdrojový kód pro HTML stránku nebo její část, ale místo konkrétního obsahu jsou napsány pouze zástupné znaky. Šablony se pak různé vkládají do sebe, tak aby se minimalizovala duplicita kódu.

Zdrojový kód je dobře čitelný a strukturovaný. Samotný web je však poměrně obsáhlý a zahrnuje velké množství různých stránek. Zdrojové kódy pro HTML stránky jsou tak rozděleny do více než 20 HTML souborů a přibližně 80 šablon.



## 4 Implementace responzivního designu

Provedená implementace responzivního designu na webovém portálu Nasipolitici.cz ukazuje jeden z možných způsobů jak postupovat při úpravě stávajícího designu webu na responzivní. Následující podkapitoly obsahují obecný popis jednotlivých kroků od návrhu, vytvoření prototypu až po samotnou implementaci, ve které je podrobně popsán postup implementace vybraných částí webu.

### 4.1 Identifikování typů stránek

Před samotnou implementací je vhodné vytvořit několik prototypů a na jejich základě sjednotit představy vývojáře a zadavatele. Některé věci je možné si ujasnit, až když si majitel webu může návrh vyzkoušet a „osahat“.

Je potřeba identifikovat jednotlivé typy stránek z pohledu rozložení jednotlivých prvků. Na základě identifikovaných stránek jsou vytvořeny prototypy. V případě webového portálu Nasipolitici.cz to byla úvodní stránka a poté zbytek webu, kde jako modelová stránka byl vybrán detail politika, protože obsahuje hodně různých prvků.

### 4.2 Stanovení priorit

V případě nového návrhu designu webu se většinou doporučuje [8] [30] [14] začít od návrhu mobilního webu a poté ho postupně upravovat pro větší rozlišení. Luke Wroblewski tento přístup nazval *Mobile First* a představil ho poprvé v roce 2009 ve stejnojmenném článku [31]. Výhodou tohoto přístupu je to, že donutí soustředit se pouze na důležitý obsah, který má být uživateli nabídnut. Současným trendem je co největší zjednodušování webů tak, aby uživateli byl nabídnut pouze relevantní obsah bez dalších výplní<sup>4</sup> navíc. Pokud je nejprve navrhován web pro malé rozlišení mobilního telefonu, není možné si dovolit zobrazovat nějaké nedůležité prvky, protože je pracováno s velmi omezeným prostorem [32].

Pokud je však upravován stávající design webu, je možné zvolit přístup opačný a web pro menší rozlišení postupně zjednodušovat. Při tomto řešení je však třeba se zamyslet

---

<sup>4</sup> Výplní je zde myšlen jakýkoliv obsah, který návštěvníkovi webu nepřináší žádnou informační hodnotu. Typicky je to například reklama nebo uživatelská anketa.

nad důležitostí jednotlivých prvků a zvážit jejich informační hodnotu, a zda je zobrazovat pouze pro větší rozlišení nebo jestli je nevpustit úplně.

Protože v případě webového portálu Nasipolitici.cz bylo třeba vycházet z původního vzhledu webu, byla po konzultaci se zadavatelem zvolena druhá varianta, tedy úprava webu pro menší rozlišení, která vychází ze stávajícího designu. Bylo třeba stanovit priority jednotlivých prvků, ze kterých lze vycházet při tvorbě jednotlivých prototypů.

Priority byly zadavatelem stanoveny ve čtyřech úrovních podle důležitosti, které jsou na obrázcích (Obrázek 4 a Obrázek 5) reprezentovány odlišnými barvami. Červenou barvou jsou označeny nejdůležitější prvky, které jsou pro návštěvníka webu stěžejní a obsahují hlavní informace, případně navigační položky. Žluté jsou prvky obsahující doplňující informace. Zelenou barvou jsou označeny části, které nejsou pro uživatele důležité, ale přesto mají na webu své místo. Modré jsou prvky, které nepřinášejí žádnou informační hodnotu, jsou již nějakým způsobem zastaralé a nemají na webu své místo, do budoucna se s nimi již nepočítá.



Obrázek 4 – stanovené priority jednotlivých prvků na webu Nasipolitici.cz (homepage)



Obrázek 5 – stanovené priority jednotlivých prvků na webu Nasipolitici.cz (detail politika)

### 4.3 Vytvoření prototypu

Na základě identifikovaných stránek a stanovených priorit je možné implementovat jednotlivé prototypy. Ty lze vytvořit jako statické webové stránky v několika verzích podle toho, jaké jsou v navrhovaném designu zvoleny *breakpointy*. Vzniknou například tři prototypy, jeden pro rozlišení se šířkou menší než 400 pixelů, další pro rozlišení se šířkou mezi 400 pixelů a 800 pixelů a třetí pro největší rozlišení. Tento přístup umožňuje navrhnout rozložení jednotlivých prvků v závislosti na různém rozlišení bez nutnosti implementace responzivního designu. V případě prototypu to může být dostačující, protože jeho účel je pouze demonstrace proměny vzhledu webu na různých zařízeních.

Druhou možností je vytvořit responzivní prototyp. Je vhodné využít nějaký responzivní *framework*, aby implementace prototypu nezabrala tolik času, protože hlavním cílem je demonstrace budoucího vzhledu a ujasnění si požadavků na něj na základě tohoto prototypu. Vytvoření responzivního prototypu sice pravděpodobně zabere o něco více času než implementace statických prototypů, nespornou výhodou však je možnost lepší demonstrace responzivity vytvořeného prototypu. Zákazník si dokáže lépe představit smysl responzivního designu, když uvidí, jak webová stránka přímo na změnu rozlišení reaguje [10]. Je to výhodné i pro samotné vývojáře a grafiky, protože se tak doladí různé detaily, které by na několika statických návrzích nebyly vidět [14].

Při návrhu responzivního designu webového portálu Nasipolitici.cz byla zvolena druhá možnost – vytvoření responzivního prototypu. Aby byla implementace co nejjednodušší, byl použit responzivní *framework* Bootstrap. Výhody tohoto *frameworku* jsou popsány v podkapitole 2.6.

Pomocí mřížky a komponent z Bootstrapu byly implementovány dva prototypy dle identifikovaných typů stránek, které na velkém rozlišení vycházejí ze stávajícího designu. Pro jednotlivé části stránek byly použity odpovídající komponenty z Bootstrapu, jako je rozbalovací menu, obrázkový prvek s rotujícími záložkami nebo panel. Tyto prvky byly již připraveny pro responzivní design. Výchozí rozložení webu bylo rozděleno na dvanáct sloupců, což odpovídá základní mřížce, kterou používá Bootstrap. Dvanáct sloupců poskytuje dostatečnou variabilitu pro umístování prvků v rámci stránky a odpovídá rozdělení ve výchozím designu webu.

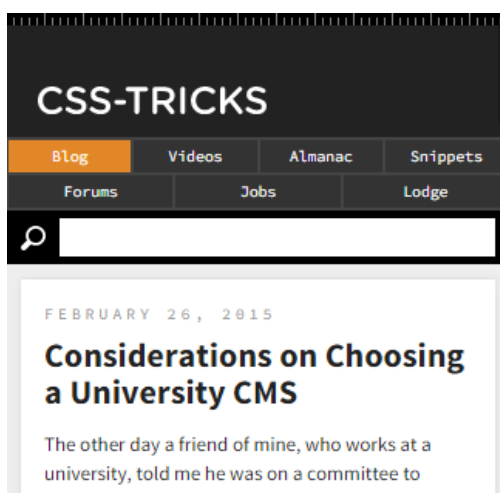
#### **4.4 Postup implementace**

V této podkapitole je popsán obecný postup implementace responzivního designu jednotlivých částí webu. Zahrnuje vytvoření mřížkového layoutu, zajištění skrývání prvků nebo změny jejich pořadí, zajištění podpory v požadovaných prohlížečích a dále implementaci jednotlivých prvků jako je například navigace nebo formuláře. Vybrané řešení je vždy demonstrováno příkladem implementace responzivního designu na webu Nasipolitici.cz.

#### 4.4.1 Navigace

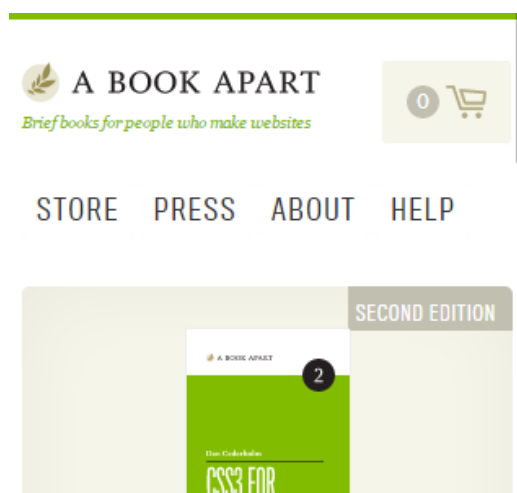
Navigace na webové stránce většinou obsahuje několik odkazů, které vedou do různých sekcí webu. Tyto odkazy lze zobrazit vedle sebe, pod sebou nebo nějakým jiným způsobem. Velmi rozšířeným způsobem zobrazování navigace na velké zobrazovací ploše je horizontální menu, kdy jsou jednotlivé navigační odkazy uspořádány vedle sebe. Pokud je při návrhu designu webu vycházeno z rozložení na malé zobrazovací ploše, je většinou použito jiné než horizontální rozložení menu, protože je k dispozici pouze malá šířka displeje. Se zobrazením menu na malé zobrazovací ploše se lze vypořádat dvěma základními způsoby.

První možností je jednotlivé odkazy pouze přeuspořádat, pokud se vedle sebe nevejdou a zobrazit je například po dvojicích nebo pod sebou. Odkazy ale budou viditelné vždy. Nevýhodou tohoto přístupu je, že obsah je vždy schovaný až pod zobrazeným menu a pokud je navigace příliš vysoká, musí uživatel vždy posouvat stránku dolů, aby se dostal k obsahu, kvůli kterému přišel. Proto je tento způsob řešení vhodný v případě, že má webová stránka jednoduché menu, které neobsahuje mnoho odkazů. Výhodou řešení je to, že návštěvník nemusí odkazy nikde hledat. V případě rozbalovacího menu by si nemusel všimnout, že je možné menu rozbalit, a k navigaci by se nemusel vůbec dostat. Příklad takového řešení je vidět na webu [css-tricks.com](http://css-tricks.com) (Obrázek 6), kde jsou odkazy zobrazeny ve dvou řádcích a dále na webu [abookapart.com](http://abookapart.com) (Obrázek 7), kde jsou pouze čtyři odkazy s krátkým textem a lze je i na úzké zobrazovací ploše vykreslit v jednom řádku.



Obrázek 6 – zobrazení všech navigačních odkazů i na úzké zobrazovací ploše

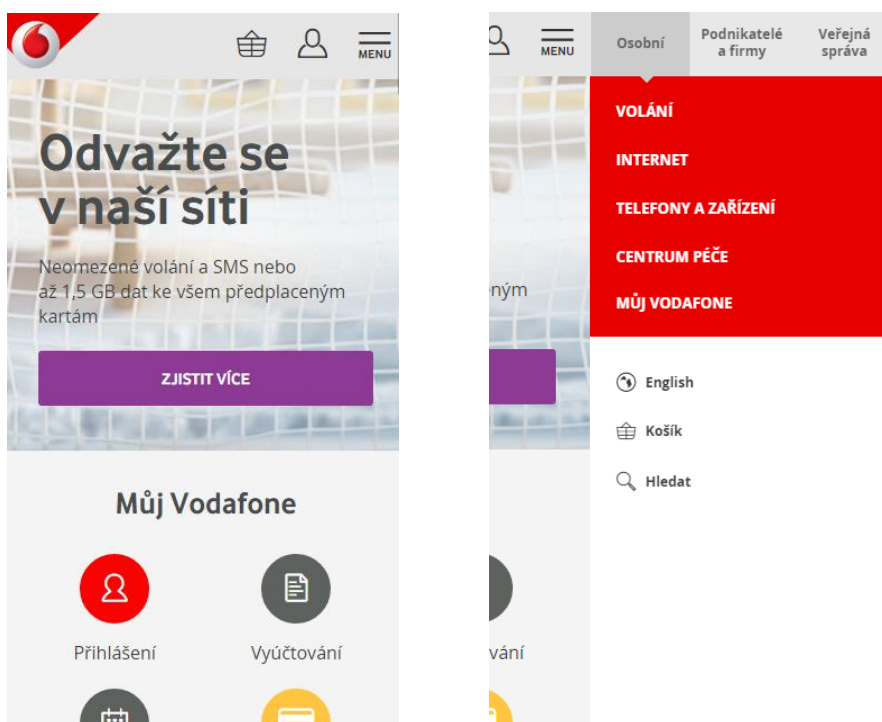
(Zdroj: <https://css-tricks.com>, 1. 3. 2015)



Obrázek 7 – zobrazení všech navigačních odkazů i na úzké zobrazovací ploše

(Zdroj: <http://abookapart.com>, 1. 3. 2015)

Druhou možností zobrazení navigace na malé zobrazovací ploše je jednotlivé odkazy schovat a zobrazit je až na vyžádání uživatele. To může být realizováno buď vytáhnutím menu ze strany displeje, jako je tomu například na webu vodafone.cz (Obrázek 8) nebo kliknutím na nějaké tlačítko, které menu rozbalí a ukáže odkazy, které nebyly viditelné, jak bude popsáno dále v této kapitole na příkladu webu Nasipolitici.cz. Při takovém rozbalení se obsah stránky posune a vytvoří prostor pro zobrazené odkazy. Takové řešení má tu výhodu, že po načtení stránky na malé zobrazovací ploše vidí uživatel hned obsah webové stránky, místo toho, aby viděl jen jednotlivé odkazy navigace. Tento způsob je vhodný tehdy, pokud navigace obsahuje více odkazů a je pravděpodobné, že její zobrazení na malé zobrazovací ploše nedovolí zobrazení části hlavního obsahu, aniž by musel uživatel stránku posouvat.



**Obrázek 8 – příklad vysouvací navigace na webu (Zdroj: <http://www.vodafone.cz>, 1. 3. 2015)**

Jak už bylo jednou zmíněno, nevýhodou schování navigace je možnost, že si uživatel nevšimne toho, že jde menu vysunout nebo zobrazit po kliknutí na tlačítko. Proto je potřeba dodržovat určité zavedené konvence nebo alespoň zajistit, aby bylo ovládání navigace uživateli dostatečně zřejmé.

V současné době se používá řešení, kdy se tlačítko pro rozbalení menu označuje třemi vodorovnými pruhy (☰). Přesto nemusí být tento symbol všem uživatelům známý, proto se doporučuje přidat popisek „menu“, který jim funkci tlačítka napoví [33].

Další nevýhodou schované navigace je, že pokud nebude prohlížeč, na kterém je webová stránka zobrazena, podporovat JavaScript, tak nebude fungovat ani vysouvání nebo zobrazování navigace řešené pomocí JavaScriptu. V takovém případě se uživatel k navigaci nemá možnost dostat. Proto je třeba zajistit, aby v případě, kdy není prohlížečem podporován JavaScript, nebyla navigace schována, ale zobrazena. Dostupnost navigace má v tomto případě přednost před tím, že bude muset uživatel posouvat stránku, aby se dostal k samotnému obsahu.

Základní navigace webu Nasipolitici.cz obsahuje šest textových odkazů. Text posledního odkazu se může měnit v závislosti na aktuálních politických volbách. Kromě šesti zmíněných odkazů je ještě součástí navigace pět ikon odkazujících na různé sociální sítě. Pokud by byly zobrazeny všechny prvky vedle sebe na jednom řádku, byla by jejich šířka asi 1100 pixelů (je potřeba počítat s delším názvem posledního odkazu) a výška 35 pixelů. Toto zobrazení je vidět na Obrázku 9.



**Obrázek 9 – jednořádková navigace na webu Nasipolitici.cz**

Pokud by byly všechny navigační prvky poskládány pod sebe, tak by jejich výška byla 344 pixelů a minimální potřebná šířka 220 pixelů. Toto zobrazení je vidět na Obrázku 10. Jednotlivé odkazy jsou vyšší proto, aby bylo snadné je stisknout i pomocí prstu na dotykovém displeji.



**Obrázek 10 – navigace na webu Nasipolitici.cz poskládaná pod sebe**

Pokud by bylo pro web Nasipolitici.cz zvoleno řešení, kdy bude navigace stále viditelná, byl by samotný obsah posunutý o výšku navigačních prvků pod sebou, tedy o 344

pixelů, navíc by bylo nutné k tomuto číslu přičíst výšku hlavičky, která by byla 100–50 pixelů. Celkem by samotný obsah začínal o více než 400 pixelů níže od začátku stránky. To je příliš mnoho, protože spousta zařízení může mít na výšku menší zobrazovací plochu než 400 pixelů, a to zejména v případě, že bude stránka zobrazována v poloze zařízení na šířku. Proto byl v tomto případě zvolen způsob skrývání navigace, a to pomocí tlačítka.

Bylo přidáno tlačítko, které po klnutí ukáže navigační položky, po dalším kliknutí je zase schová (Obrázek 11). Toto řešení bylo implementováno pomocí knihovny jQuery a její funkce `toggleClass()`. Ta funguje tak, že po zavolání na daném HTML elementu přiřadí CSS třídu `active`, pokud na daném elementu tato třída není, v opačném případě tuto třídu odebere (Ukázka kódu 8). Pomocí třídy `active` je zajištěno zobrazení a schování navigačních položek (Ukázka kódu 9).



Obrázek 11 – otevřené a zavřené menu na webu Nasipolitici.cz

```
$('#collapse-menu').click(function() {
    $('#topMenu ul').toggleClass('active');
    $('#topMenu socialNetworks').toggleClass('active');
});
```

Ukázka kódu 8 – zobrazování a schovávání menu pomocí JavaScriptu

```
#topMenu ul, #socialNetworks {
    display: none;
    &.active {
        display: block;
    }
}
```

Ukázka kódu 9 – nastavení stylů pro zobrazování a schovávání menu v LESS



Na začátku podkapitoly bylo uvedeno, že menu zobrazené v jednom řádku je široké 1100 pixelů. Z toho důvodu by bylo nutné zobrazit schovávací menu na všech zobrazovacích plochách, které jsou užší než 1100 pixelů. Protože je však možné navigaci uspořádat do dvou řádků, které ještě nezaberou moc místa na výšku, bylo toto řešení použito pro zobrazovací plochy, které jsou široké alespoň 990 pixelů, což je minimální šířka, která je potřeba pro zobrazení dvouřádkové navigace. Aby se jednotlivé odkazy zobrazovaly v mřížce pod sebou, bylo nutné každému textovému odkazu nastavit šířku na 33 % (Ukázka kódu 10), protože potom každý odkaz zabere jednu třetinu prostoru určeného pro zobrazení navigačních odkazů. Inspirací pro toto řešení byla dvouřádková mřížková navigace Ethana Marcotta [14]. Výsledek je vidět na Obrázku 12.

```
ul#topMenu1 li {  
    float: left;  
    width: 33%;  
}
```

#### Ukázka kódu 10 – nastavení stylů pro dvouřádkovou navigaci v LESS



Obrázek 12 – dvouřádková navigace na webu Nasipolitici.cz

Jak již bylo zmíněno dříve v této podkapitole, pokud je použito řešení skrývání a zobrazování menu pomocí JavaScriptu, je také potřeba zajistit, aby byla navigace viditelná v případě zobrazení webové stránky v prohlížeči nepodporujícím JavaScript. Řešením je spustit po načtení stránky nějaký JavaScript. Pokud se provede, je JavaScript v daném prohlížeči podporován. V případě webu Nasipolitici.cz bylo zvoleno řešení s použitím JavaScriptové funkce `removeClass()`, která odebere konkrétní CSS třídu `nojs` na HTML elementu s id `topMenu` (Ukázka kódu 11).

```
$('#topMenu').removeClass('nojs');
```

#### Ukázka kódu 11 – detekce podpory JavaScriptu pomocí odebrání třídy

Pokud prohlížeč podporuje JavaScript, tak je třída `nojs` odebrána a navigace schována do doby, než uživatel klikne na tlačítko „MENU“. Pokud, ale není JavaScript podporován, je díky CSS třídě `nojs` navigace zobrazena (Ukázka kódu 12).

```

// Není podporován JavaScript
// Schování tlačítka MENU pro a zobrazení navigace
#topMenu.nojs {
  #collapse-menu {
    display: none;
  }
  ul {
    display: block !important;
  }
  #socialNetworks {
    display: block;
  }
}
// Je podporován JavaScript
// Zobrazení tlačítka MENU a schování navigace
#topMenu
ul {
  display: none;
  ...
}
#socialNetworks {
  display: none;
  ...
}
}

```

**Ukázka kódu 12 - nastavení stylů pro zobrazení navigace, pokud není podporován JavaScript v LESS**

#### 4.4.2 Mřížkový layout

V podkapitole 4.3 bylo popsáno vytvoření prototypu pomocí Bootstrapu, kde byl design rozdělen do mřížky o dvanácti sloupcích. Tato mřížka byla zachována i při samotné implementaci. Byl ale zvolen trochu jiný přístup. Velikost prvku v jednotlivých velikostech zobrazení nebyla definována v HTML pomocí CSS tříd Bootstrapu jako jsou například `col-md-6` nebo `hidden-sm`, ale byla implementována přímo v CSS tak, aby nebylo třeba upravovat strukturu HTML šablon, ale pouze CSS. K tomu byly využity LESS *mixins*, které Bootstrap také nabízí.

V uvedeném příkladu (Ukázka kódu 13) jsou použity dvě Bootstrap *mixins*: `make-row()` a `make-md-column(@columns)`. První zajišťuje vytvoření vizuálního řádku,

do něj jsou pak vkládány elementy, které se zarovnávají vedle sebe do sloupců, pokud je k dispozici dostatek místa, nebo pod sebe. Například `make-lg-column(10)` na jednom elementu zajistí roztažení přes deset pomyslných sloupců z celkových dvanácti. Na druhém elementu obdobně `make-lg-column(2)` způsobí roztažení přes zbývající dva pomyslné sloupce z dvanácti. Znaky `lg`, `md` a `sm` znamenají použití pro různé šířky zobrazení, dle definovaných *breakpointů*. Element s id `defaultRightBox` bude na odpovídající šířce zobrazení schován díky *mixin* `make-sm-hidden()`. Použití *mixin* `make-md-column(9)` je tedy stejné jako kdyby byla přímo na elementu definována CSS třída `col-md-9`.

```
#main {
  .make-row();

  #mainInLeft {
    .make-sm-column(12);
    .make-md-column(9);
    .make-lg-column(10);
  }

  #defaultRightBox {
    .make-sm-hidden();
    .make-md-column(3);
    .make-lg-column(2);
  }
}
```

**Ukázka kódu 13 – vytvoření mřížky za pomoci Bootstrap *mixins* v LESS**

### 4.4.3 Skrývání prvků

V předchozí podkapitole již bylo ukázáno schování části obsažené v kontejneru s id `defaultRightBox`. Tato sekce obsahuje seznam odkazů na nejnovější články, anketu a seznam organizací, které podporují webový portál. Anketa a seznam organizací jsou pouze doplňující informace, stejně tak i seznam odkazů, protože uživatel se může k daným článkům dostat i jinak. Je tedy možné dané prvky schovat.

Další možností je schovat některé obrázky, které uživateli nepřinášejí žádnou informační hodnotu. Pokud budou schovány obrázky, je lepší, aby prohlížeč vůbec negeneroval

request pro jejich stažení, když je uživatel stejně neuvidí. Implementace tohoto řešení je popsána v podkapitole 4.4.6.

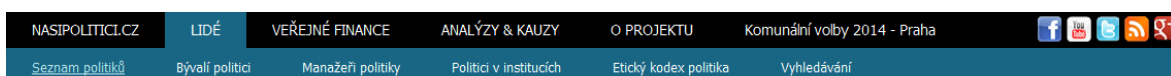
V případě schovávání jakýchkoli částí webu je třeba dát velký pozor na to, aby nebyly schovány navigační odkazy, bez kterých se uživatel nedostane na cílovou stránku jiným způsobem. Dále je potřeba schovávat jen takové prvky, které mají pouze výplňový charakter a u kterých lze předpokládat, že je nebude uživatel na webu postrádat. Výhodou je přístup k návrhu webu zvaný *mobile first*, kdy je základní web tvořen pro malá rozlišení obrazovky mobilních zařízení a obsahuje vše důležité. Výplně jsou případně postupně přidány až pro větší rozlišení.

#### 4.4.4 Změna pořadí prvků

Může nastat situace, kdy by se hodilo změnit pořadí HTML elementů pro nějaký *breakpoint*. HTML je navrženo tak, že jednotlivé prvky se skládají za sebou ve stejném pořadí, jako jsou definovány v jeho struktuře. Pomocí CSS je možné docílit prohození vedle sebe, ale už ne prvků pod sebou. Například Bootstrap [34] umožňuje ve své implementaci mřížky změnit pořadí sloupců v jednom řádku pomocí CSS tříd `.col-md-push-*` a `.col-md-pull-*`. K tomu využívá CSS vlastnosti `left` a `right`. Řešením pro změnu pořadí prvků, které nejsou zobrazeny v jednom řádku, ale pod sebou, je použití JavaScriptu [10].

**AppendAround** (dostupné z <https://github.com/filamentgroup/AppendAround>) je JavaScriptová knihovna, která umožní přesun části HTML kódu z jednoho umístění do jiného v rámci jednoho HTML souboru. Je potřeba mít dva `div` kontejnery se stejnou hodnotou atributu `data-set`. A pomocí CSS *media queries* zajistit, aby byl vždy pouze jeden z nich viditelný a druhý měl nastavené `display: none`. Knihovna AppendAround pak zajistí, aby byl obsah kontejneru `div` vložen pouze do toho, který je na stránce viditelný [12].

Na webovém portálu Nasipolitici.cz je na všech stránkách kromě homepage zobrazeno podmenu umístěné pod hlavní navigací, které se váže k aktuálně zobrazené sekci webu. Při širokém zobrazení je navigace zobrazena ve dvou řádcích (Obrázek 13).



Obrázek 13 - navigace včetně podmenu při širokém zobrazení na webu Nasipolitici.cz

Při úzkém zobrazení se jednotlivé navigační odkazy skládají pod sebe a to v tom pořadí, v jakém jsou zapsány v HTML, tedy nejprve hlavní navigační odkazy, potom odkazy na sociální sítě a nakonec odkazy obsažené v podmenu (Ukázka kódu 14). Během provedeného uživatelského testování (viz podkapitola 5.1) bylo zjištěno, že si uživatel na malém displeji mobilního telefonu nevšimne a ani nepředpokládá, že by pod odkazy na sociální sítě ještě mohla navigace pokračovat. Proto bylo implementováno řešení, kdy jsou v případě, že se odkazy skládají pod sebe, přesunuty odkazy z podmenu do hlavního menu pod aktivní odkaz aktuálně zobrazované sekce. Stav před přesunem a zobrazovaný stav po přesunu, který vidí uživatel, je vidět na Obrázku 14.

```
<ul id="topMenu1">...</ul>
<div id="socialNetworks">...</div>
<ul id="topMenu2">...</ul>
```

**Ukázka kódu 14 – ukázka struktury HTML navigace na webu Nasipolitici.cz**



**Obrázek 14 – přesunutí podmenu v rámci úzkého zobrazení navigace na webu Nasipolitici.cz**

Pro implementaci popisovaného přesunu podmenu byla využita jQuery metoda `insertAfter()`. Ta v případě úzkého zobrazení zajistí potřebný přesun obsahu elementu `#topMenu2` za aktivní položku odkazu hlavního menu (`#topMenu1 li.active`). V případě širokého zobrazení pak přesun podmenu (`#topMenu2`) až za odkazy na sociální sítě (`#socialNetworks`). Přesun je proveden jak při načtení stránky, tak při případné změně velikosti okna prohlížeče (Ukázka kódu 15).

```
function swapMenu2() {
    if (window.matchMedia("(max-width: 989px)").matches) {
        $("#topMenu2").insertAfter($("#topMenu1 li.active"));
    } else {
        $("#topMenu2").insertAfter($("#socialNetworks"));
    };
}

$(document).ready(function() {
    swapMenu2();
});

$(window).resize(function() {
    swapMenu2();
});
```

**Ukázka kódu 15 - implementace změny pořadí prvků v navigaci na základě šířky zobrazení na webu Nasipolitici.cz**

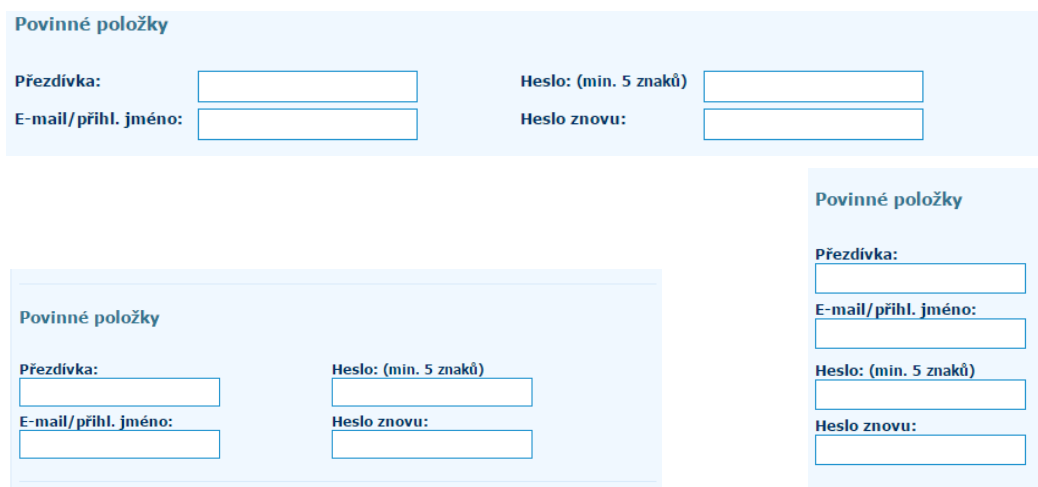
#### 4.4.5 Formuláře

Formuláře jsou běžnou součástí webových stránek. Při implementaci responzivního formuláře je třeba vytvořit jeho rozložení tak, aby se přizpůsoboval velikosti stránky. Formulář se skládá z několika textových polí. Každé pole má svůj popis, ten se standardně nachází buď vlevo, nebo vpravo od textového pole. Displej na mobilu je ale většinou malý na to, aby se tam mohl zobrazit popis a textové pole vedle sebe. Proto je lepší přesunout v takovém případě popis nad textové pole. Jednak se tím ušetří místo do šířky a navíc je popis viditelný i v případě, kdy se na displeji telefonu zobrazí virtuální klávesnice. Druhou možností je zobrazit popis přímo v textovém poli. Zde je ale třeba dát pozor na to, aby si uživatel nespletl popis s předvyplněnou hodnotou [32].

Při implementaci formulářů je dobré využít nových hodnot vlastnosti `input type`, které byly definovány v HTML5 [17]. Jsou to například hodnoty `email`, `number` nebo `url`. Většina mobilních prohlížečů na základě toho zobrazí upravenou klávesnici,

kteřá bude obsahovat znak „@“ v pŕípadě e-mailu, alfanumerickou klávesnici v pŕípadě čísla nebo znaky „/“ a „.com“ pro pole typu URL. Použití těchto nových hodnot vlastnosti `input type` je bezpečné, protože prohlížeče, které jim nerozumějí, budou hodnotu pŕekládat jako normální textové pole [32].

Na webu Nasipolitici.cz se nachází formulář pro registraci nového diskutujícího uživatele. Na Obrázku 15 je vidět, že pokud je dostatek místa pro zobrazení, bylo použito rozložení do dvou sloupců. Při užším zobrazení se pŕesunou popisky jednotlivých polí nad ně tak, aby nebyly sloupce široké. Při nejužším zobrazení se pŕesnou jednotlivé pole pod sebe.



Obrázek 15 - registrační formulář webu Nasipolitici.cz při různých šířkách zobrazení

V popisovaném formuláři, který obsahuje editační políčko pro zadání pŕezdívky, e-mailu a hesla, byly použity pŕíslušné hodnoty vlastnosti `input type`, kterými jsou `text`, `email` a `password` (Ukázka kódu 16).

```
<fieldset class="loginData loginDataLeft">
  <label for="reg_nick">Pŕezdívka:</label>
  <input id="reg_nick" name="reg_nick" type="text">
  <label for="reg_email">E-mail/pŕihl. jméno:</label>
  <input id="reg_email" name="reg_email" type="email">
</fieldset>
<fieldset class="loginData">
  <label for="reg_passwd">Heslo</label>
  <input id="reg_passwd" name="reg_passwd" type="password">
  <label for="reg_passwd2">Heslo znovu:</label>
  <input id="reg_passwd2" name="reg_passwd2" type="password">
</fieldset>
```

Ukázka kódu 16 - část HTML kódu formuláře pro registraci nového uživatele na webu Nasipolitici.cz

Registrační formulář má dvě poloviny, každá je tvořena elementem `fieldset`. Ten má nastavenou relativní šířku na 49 % a vlastnost `float`. Díky tomu je zajištěno zobrazení ve dvou sloupcích. Elementy `label` a `input` pak mají nastaveny absolutní šířku v pixelech. Díky tomu je zajištěno, že pokud je součet jejich šířek větší, než je polovina stránky, zobrazí se popisek nad editačním políčkem (Ukázka kódu 17).

```
.loginData {
  width: 49%;
  &.loginDataLeft {
    float: left;
  }
  label {
    width: 150px;
    float: left;
  }
  input {
    width: 178px;
    margin-bottom: 5px;
  }
}
```

**Ukázka kódu 17 - ukázka LESS pro různé uspořádání popisků a editačních políček ve formuláři při různých šířkách zobrazení na webu Nasipolitici.cz**

#### 4.4.6 Obrázky

Kromě načítání obrázků v různých velikostech dle typu zařízení byla v podkapitole 2.4.2 naznačena i možnost načítání obrázků pro vybraná zařízení. V některých případech může být na webové stránce zobrazen velký obrázek, který při určité velikosti postrádá svůj význam (například pokud obsahuje text). Takový obrázek je dobré nahradit textem, pokud to je možné.

V případě, že se jedná o ilustrační obrázek, kde by nahrazení textem nedávalo příliš smysl, je někdy jednodušší obrázek bez ztráty významu vypustit. Obrázek ale nestačí jen schovat pomocí CSS stylů, protože nastavení stylu na `display: none` nezabrání tomu, aby si prohlížeč obrázek stáhnul. Jedná se o další request, který zpomaluje načítání stránky, zvyšuje objem přenesených dat a uživateli nepřináší žádnou přidanou hodnotu, protože pro něj není obrázek viditelný. Řešením takového případu je použití



JavaScriptu, který přidá obrázek do HTML stránky pouze v případě, že je zobrazovací plocha prohlížeče širší, než je stanovená hodnota.

Na webu Nasipolitici.cz byl takový příklad identifikován na stránce „Státní rozklikávací rozpočet – budovanistatu.cz“, kde je zobrazen velký obrázek s náhledem na webovou stránku rozpočtu (Obrázek 16). Když je obrázek zmenšen tak, že je jeho šířka menší než 700–800 pixelů, přestává být již rozeznatelné písmo na obrázku a tím i jeho skutečný obsah. Obrázek má kromě funkce náhledu na webovou stránku rozpočtu také funkci odkazu na tento web. Proto bylo vhodné obrázek nahradit jednoduchým textovým odkazem (Obrázek 17).

Státní rozklikávací rozpočet - budovanistatu.cz

**Rozklikávací rozpočet**  
Rozpočet

Graf Tabulka Lidé Metodika

Jméno	2004	2005	2006	2007	2008	2009	2010	
Státní rozpočet	860 013 280 330	922 685 777 760	1 020 596 584 860	1 092 149 680 380	1 083 944 083 960	1 167 070 425 710	1 156 857 957 990	
313 - Ministerstvo práce a sociálních věcí	313 963 147 810	337 783 122 490	372 125 448 070	435 092 117 670	441 762 792 980	472 463 549 350	476 364 573 500	
398 - Všeobecná pokladniční správa	179 727 826 000	164 732 255 000	180 149 518 000	137 702 864 830	138 286 052 890	127 225 618 310	131 646 535 170	
333 - Ministerstvo školství, mládeže a tělovýchovy	95 597 969 510	102 577 296 700	110 881 868 910	123 199 072 530	119 382 689 960	127 631 764 780	124 267 243 590	
327 - Ministerstvo dopravy	18 915 923 930	26 838 567 530	27 122 543 710	37 771 875 630	60 719 220 800	68 219 577 010	63 727 522 250	
314 - Ministerstvo vnitra	47 669 148 270	51 516 702 440	55 610 488 710	58 233 813 220	59 828 355 360	59 764 172 030	59 480 266 080	
329 - Ministerstvo zemědělství	29 299 278 840	50 307 580 460	50 313 400 710	65 818 419 640	45 644 702 240	55 879 584 270	52 862 773 640	
307 - Ministerstvo obrany	52 481 164 780	58 445 016 370	63 104 665 420	61 338 072 780	52 896 322 540	59 725 830 160	50 855 459 980	
396 - Státní dluh	26 833 725 860	25 431 178 800	31 608 441 920	34 126 074 800	44 751 343 740	53 072 861 180	48 740 767 890	
317 - Ministerstvo pro místní rozvoj	5 793 680 450	5 962 121 090	16 263 821 530	18 076 580 260	9 268 157 150	19 832 628 530	34 804 707 810	
336 - Ministerstvo spravedlnosti	18 652 415 010	20 336 205 770	19 436 036 580	20 841 895 950	21 859 009 190	22 262 991 270	20 889 989 050	
322 - Ministerstvo průmyslu a obchodu	11 721 524 030	14 213 186 040	14 573 878 950	19 358 210 830	16 842 737 660	18 174 205 510	17 033 490 690	
312 - Ministerstvo financí	14 310 377 900	16 084 505 570	16 443 647 840	17 314 430 030	17 315 359 840	18 867 289 620	16 734 207 260	
315 - Ministerstvo životního prostředí	4 348 824 190	4 602 900 110	13 777 937 030	15 560 112 150	8 886 516 970	12 818 570 690	14 840 220 110	
335 - Ministerstvo zdravotnictví	9 649 161 440	9 676 457 990	10 405 411 270	10 704 584 610	8 823 261 660	11 315 812 570	9 425 894 930	
334 - Ministerstvo kultury	6 549 237 460	6 497 302 170	7 095 090 780	8 300 420 170	7 940 642 940	7 844 562 500	7 706 828 160	
306 - Ministerstvo zahraničních věcí	5 945 482 420	6 441 228 400	6 707 798 850	6 910 581 250	7 098 748 880	7 321 897 810	6 363 628 830	
361 - Akademie věd České republiky	4 085 191 700	4 565 619 930	4 858 751 810	5 716 166 370	5 555 282 160	5 913 722 440	5 026 632 430	
346 - Český úřad zeměměřičký a katastrální	2 367 428 580	2 450 487 100	2 549 589 120	2 817 254 940	2 806 479 940	3 173 452 680	3 342 391 320	
374 - Správa státních hmotných rezerv		1 955 777 880	2 005 211 490	2 197 564 930	2 380 574 700	2 611 284 320	2 148 358 310	
321 - Grantová agentura České republiky	1 246 791 740	1 339 732 440	1 410 296 880	1 543 945 800	1 537 645 190	1 780 099 290	1 988 635 830	

„Projekt Budovanistatu.cz poprvé zpřístupňuje odborné i laické veřejnosti rozsáhlý soubor dat sledujících hospodaření složek státu v transparentní a dobře srozumitelné podobě tzv. "rozklikávacího rozpočtu".

Server Budovanistatu.cz nabízí veřejnosti strukturované „rozklikávací“ přehledy výdajové struktury státu (rozpočtové skladby) z období

Obrázek 16 – zobrazení velkého obrázku na webu Nasipolitici.cz, který po zmenšení stránky přestává být přehledný

## Státní rozklikávací rozpočet - budovanistatu.cz

[Klikněte a podívejte se Rozklikávací rozpočet - Budovanistatu.cz](http://budovanistatu.cz)

„Projekt Budovanistatu.cz poprvé zpřístupňuje odborné i laické veřejnosti rozsáhlý soubor dat sledujících hospodaření složek státu v transparentní a dobře srozumitelné podobě tzv. "rozklikávacího rozpočtu".

Server Budovanistatu.cz nabízí veřejnosti strukturované „rozklikávací“ přehledy výdajové struktury státu (rozpočtové skladby) z období let 2004-2010. Účetní data jsou doplněna o vlastní

**Obrázek 17 – zobrazení odkazu nahrazující obrázek při menších zobrazeních na webu Nasipolitici.cz**

Použitý HTML kód obrázku a potřebný JavaScript lze vidět v Ukázce kódu 18 a 19. V případě, že není prohlížečem podporován JavaScript, zobrazí se pouze textový odkaz. Za pomoci JavaScriptu a jeho funkce `matchMedia()` je testována šířka zobrazovací plochy [10]. Podmínka je splněna, když je šířka alespoň 700 pixelů. Potom je smazán text uvnitř odkazu pomocí funkce `empty()` z knihovny jQuery a následně do něj vložen obrázek pomocí funkce `appendTo()`, také z knihovny jQuery.

```
<a href="http://budovanistatu.cz" id="budovaniStatuLink">Klikněte a  
podívejte se na Rozklikávací rozpočet – Budovanistatu.cz</a>
```

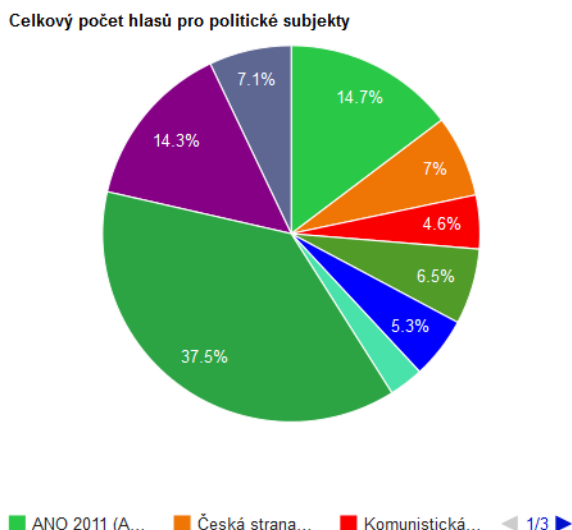
**Ukázka kódu 18 – odkaz, do kterého je vkládán obrázek pomocí JavaScriptu v HTML**

```
if (!window.matchMedia("(max-width: 699px)").matches) {  
    $("#budovaniStatuLink").empty();  
    $("<img src='/images/ostatni/budovanistatu-tabulka.png'  
alt='Rozklikávací rozpočet – Budovanistatu.cz' title='Klikněte a  
podívejte se Rozklikávací rozpočet – Budovanistatu.cz' style='width:  
100%;' />").appendTo("#budovaniStatuLink")  
};
```

**Ukázka kódu 19 – nahrazení textu odkazu obrázkem v JavaScriptu**

### 4.4.7 Graf

Google Chart Api (dostupné z <https://developers.google.com/chart>) je nástroj, který umožňuje vykreslit různé typy grafů přímo ve webové stránce na základě poskytnutých dat. Webový portál Nasipolitici.cz obsahuje na stránkách jednotlivých politických voleb koláčové grafy (Obrázek 18), které jsou vykresleny pomocí Google Chart Api. Tyto grafy měly absolutně definovanou velikost v pixelech. Pro potřeby responzivního designu bylo potřeba zajistit změnu velikosti grafu podle šířky zobrazovací plochy.



**Obrázek 18 - graf vykreslený pomocí Google Chart Api na webu Nasipolitici.cz**

V Ukázce kódu 20 je uveden příklad zobrazení grafu s relativně definovanou velikostí. Pro vykreslení grafu je třeba definovat data pro graf (proměnná `data`) a nastavení grafu (proměnný `options`). Nakonec je zavolána funkce `draw(data, options)`. Kromě titulku, umístění legendy a dalších nastavení je možné definovat velikost grafu (`chartArea`). Ta se uvádí buď jako číslo, které se vyhodnocuje jako pixely nebo ji lze definovat pomocí textového řetězce obsahující velikost v procentech. V ukázce je nastavena šířka na 100 % a výška na 60 %. Díky tomu se graf roztáhne na celou šířku kontejneru s id `chart`, do kterého je vložen. Ten má nastavenou šířku na 100 %, ale zároveň má nastavenou i maximální šířku pro případ širokého zobrazení (Ukázka kódu 21).

```

var data = ...;
var options = {
  'title': 'Celkový počet hlasů pro politické subjekty',
  'legend': { position: 'bottom', alignment: 'start'},
  'chartArea': { left:20, top:35, width: "100%", height:"60%" }
};
var chart = new
google.visualization.PieChart(document.getElementById('chart'));
chart.draw(data, options);

```

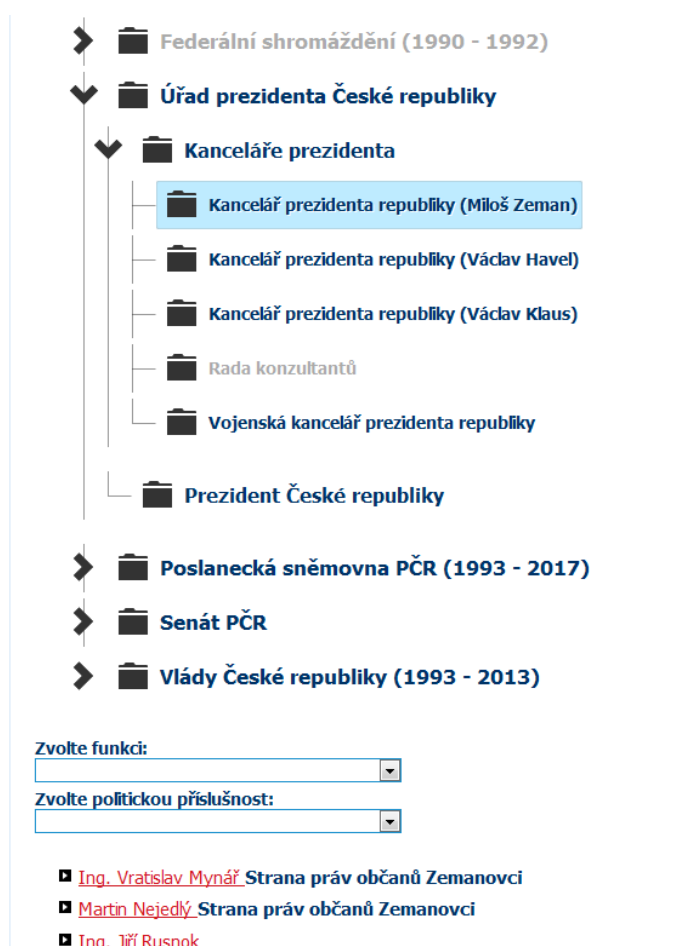
**Ukázka kódu 20 - příklad vykreslení grafu s relativně definovanou velikostí pomocí Google Chart Api na webu Nasipolitici.cz**

```
#chart {
  max-width: 440px;
  width: 100%;
}
```

Ukázka kódu 21 – definice stylů pro kontejner obsahující responzivní graf

#### 4.4.8 Stromová struktura

Řešení jsTree (dostupné z <http://www.jstree.com>) je jQuery *plugin*, který poskytuje zobrazení interaktivní stromové struktury. Na webovém portálu Nasipolitici.cz je použit na stránce s parametrickým vyhledáváním. *Plugin* má implementovanou podporu responzivního designu od verze 3. Do šířky zobrazení 768 pixelů se strom zobrazí s většími rozestupy a většími obrázky, aby na jednotlivé položky šlo pohodlně kliknout i za pomoci prstů na dotykovém displeji (Obrázek 19). Na širší zobrazovací ploše zobrazí menší ikonky (Obrázek 20).



Obrázek 19 – zobrazení stromu (jsTree) pro užší zobrazovací plochu na stránce parametrického vyhledávání na webu Nasipolitici.cz



**Obrázek 20 – zobrazení stromu (jsTree) na stránce parametrického vyhledávání na webu Nasipolitici.cz**

Na stránce parametrického vyhledávání, kde je strom zobrazen, se po kliknutí na položku stromu zobrazí odkazy na politiky, které patří pod vybranou kategorii. V případě, že je pro zobrazení k dispozici dostatečně široký prostor, je možné tyto odkazy zobrazit vpravo od stromu. Pro užší layouty bylo nutné tyto odkazy přesunout až pod rozbalovací strom. Objevil se však problém, kdy uživatel na malém displeji vidí jen část stromu a po kliknutí na nějaký prvek netuší, že se mu dole zobrazily příslušné odkazy. Proto byla implementována animace, která po vybrání uzlu stromu posune viditelnou část stránky tak, aby byly vidět odkazy (Ukázka kódu 22).

```

if (window.matchMedia("(max-width: 699px)").matches) {
    $("html, body").animate({ scrollTop:
    $('#searchSelect').offset().top - 5}, 1000);
}

```

#### Ukázka kódu 22 – animace posunu viditelné části stránky v JavaScriptu

V grafickém stylu, který je dodáván spolu s jsTree, je nastaven *breakpoint* pro zobrazení responzivního designu 768 pixelů. V CSS souboru, který obsahuje styly, byl tento *breakpoint* upraven na 700 pixelů tak, aby odpovídal jednomu z hlavních *breakpointů* webového portálu Nasipolitici.cz.

Protože byla původně použita verze 1, bylo potřeba aktualizovat jsTree na verzi 3 a provést další úpravy. Nejdříve bylo potřeba stáhnout novou verzi a vyměnit zdrojové soubory. Data v interaktivním stromu jsou načítána z generovaných dat typu JSON. V nové verzi se změnila jejich struktura. Bylo potřeba přejmenovat data

na `text`, `attr` na `li_attr` a `state` na objekt obsahující atribut `opened` s hodnotou `true` nebo `false`. Původní použitá struktura dat je uvedena v Ukázce kódu 23, nová struktura dat v Ukázce kódu 24. Místa, která byla potřeba změnit, jsou označena tučně.

```
{
  data      : "string",
  state     : "string",
  children  : [],
  attr      : {
    id       : "string",
    class    : "string",
  }
}
```

#### Ukázka kódu 23 – původní struktura dat typu JSON pro jsTree verze 1

```
{
  text      : "string",
  state     : {
    opened   : boolean
  },
  children  : [],
  li_attr   : {
    id       : "string",
    class    : "string",
  }
}
```

#### Ukázka kódu 24 – nová struktura dat typu JSON pro jsTree verze 3

Ve verzi 3 se změnil způsob inicializace jsTree na stránce, kde je použitý (Ukázka kódu 25 a 26). Hlavní změnou je zahrnutí předtím samostatných *pluginů* „themes“ a „json\_data“ přímo do jsTree, které ve verzi 1 umožňovaly použití jiného grafického tématu a načítání dat pomocí JSONu. Dále byl ve verzi 3 zrušen *plugin* „ui“, který omezil počet najednou vybraných prvků stromu. Standardně si uživatel může pomocí klávesy CTRL vybrat více položek. To je v případě vyhledávacího stromu na webovém portálu Nasipolitici.cz nežádoucí. Verze 3 implementuje funkci omezení počtu vybraných položek přímo v samotném jsTree.

Při inicializaci jsTree lze namapovat některou z událostí, například vybraní položky stromu, a na základě této události vykonat nějakou funkci. Ve verzi 1 bylo mapování implementováno pomocí funkce `bind()`, ve verzi 3 je to funkce `on()`.

```

$('#jsTree')
.jstree({
  'ui' : {
    'select_limit' : 1,
    'selected_parent_close': false
  },
  'themes': {
    'theme': 'název použitého tématu'
  },
  'json_data' : {
    'ajax' : {
      'url' : 'URL, na které se získá JSON data'
    }
  },
  'plugins': ['themes', 'json_data', 'ui']
})
.bind('select_node.jstree', function (e, data) {
  // kód, který bude spuštěn po výběru konkrétní položky stromu
  // vybranou položku lze získat pomocí data.rslt.obj.attr('id')
});

```

#### Ukázka kódu 25 – původní implementace inicializace jsTree verze 1

```

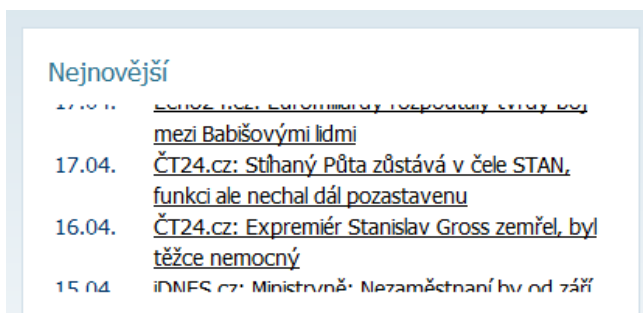
$('#jsTree')
.jstree({
  'core' : {
    'multiple' : false,
    'themes': {
      'name': 'název použitého tématu',
      'url': 'url ke stylům použitého tématu',
      'dir': 'složka ve které jsou uložena použitá témata',
      'icons': true,
      'responsive': true
    },
    'data' : {
      'url' : 'URL, na které se získá JSON data'
    }
  }
})
.on('select_node.jstree', function (e, data) {
  // kód, který bude spuštěn po výběru konkrétní položky stromu
  // vybranou položku lze získat pomocí data.selected
});

```

#### Ukázka kódu 26 – nová implementace inicializace jsTree verze 3

#### 4.4.9 Carousel

Na webovém portálu Nasipolitici.cz se nachází prvek s rotujícími odkazy na články (Obrázek 23). K jeho implementaci byl ve výchozí verzi použit `jCarousel Lite` jQuery *plugin* (dostupné z <http://www.gmarwaha.com/jquery/jcarouselite/index.php>). Toto řešení je plně použitelné v responzivním designu, protože velikost karuselu se přizpůsobuje velikosti elementu `div`, do kterého je vložen. Proto nebyly potřeba žádné dodatečné úpravy.



Obrázek 21 – prvek s rotujícími odkazy na články (*carousel*) na webu Nasipolitici.cz

#### 4.4.10 Záložky

Na domovské stránce webu Nasipolitici.cz je použit prvek, kde jsou čtyři odkazy pro přepínání viditelné záložky a jedna záložka na které je text s obrázkem. Taby se každých 15 sekund automaticky přepínají. Záložky lze také přepnout ručně pomocí odkazů.

Tento prvek je vytvořen pomocí *pluginu* jQuery UI Tabs (dostupné z <http://jqueryui.com/tabs>). Pro layout se šířkou minimálně 700 pixelů je zachován původní vzhled (Obrázek 22). Pro užší rozložení stránky jsou zobrazeny všechny čtyři obrázky s popisem najednou a záložky nejsou viditelné (Obrázek 23). Odkazy společně s obrázky by se na malý displej špatně vešly a nebylo by to pro uživatele tak přehledné, proto byla zvolena možnost schování odkazů a zobrazení všech záložek.



Obrázek 22 – původní vzhled prvku se záložkami na domovské stránce webu Nasipolitici.cz



## Obchod s bídou




Obchod s bídou chce poukázat na problém začarovaného kruhu, který v řadě regionů spojuje sociálně vyloučené lokality, zneužívání tíživé situace ekonomicky slabých občanů, volební korupci a vyvádění veřejných prostředků. Cílem je také nabídnout a aktivně prosazovat účinná řešení.

## NOVÝ POLITIK


**PhDr. Mgr. Ivan Langr**

Náměstek primátora města Liberce (SZ), sociolog, novinář. V roce 1995 absolvoval Pedagogickou fakultu UJEP v Ústí nad

Labem. Tři roky pracoval jako učitel českého jazyka a dějepisu na gymnáziu. V letech 1998–2007 působil jako novinář, nejprve rok v deníku MF DNES, následně v deníku Právo. Do roku 2014 pracoval jako tiskový mluvčí a vedoucí marketingu ZOO Liberec, p.o. Vzdělání si rozšířil o studium politologie na FF Univerzity Hradec Králové a o studium veřejné a sociální politiky na FSV v Praze. Od roku 2014 je členem Strany zelených.... [další](#) 


## KAUZA



Analýza: Sobotkův kabinet provedl v porovnání se dvěma předchozími vládami nejrazantnější změny dozorčích rad a představenstev státních podniků. Více než třetina nominovaných je stále spojena s politickými stranami. Nevládní... [celý článek](#) 

## KOMENTÁŘ TÝDNE



Úmrtí Stanislava Grosse nám dalo vzpomenout na éru, kdy bylo v politice prý možné všechno. Kdo se prokousal do nejvyšších mocenských pater, nezávisle na věku, vzdělání, zkušenostech z praxe, mohl absolvovat „bezstarostnou jízdu“: naplno... [celý článek](#) 

### Obrázek 23 – vzhled prvku se záložkami na domovské stránce webu Nasipolitici.cz při úzkém zobrazení

Automatické přepínání tabů bylo implementováno pomocí JavaScriptové funkce `setInterval()`, protože novější verze jQuery UI nepodporují automatické přepínání prvků. Bylo také potřeba vypnout přepínání v případě, že jsou všechny taby viditelné. K tomu byla použita JavaScriptová metoda `matchMedia()`, pomocí které lze zjistit, jestli je splněna podmínka CSS *media query* [10]. Dále byla použita JavaScriptová funkce `resize()`, která provede vložený blok kódu pokaždé, když je změněna velikost okna prohlížeče. V popisovaném příkladu záložek to znamená, že když uživatel otevře prohlížeč na velkém monitoru a zobrazí se mu rozložení s výchozím vzhledem prvku se záložkami, tak se zapne automatické přepínání záložek. Když potom okno zúží tak, že se mu zobrazí rozložení bez záložkových odkazů se všemi obrázky viditelnými najednou (v tomto případě je to méně než 699 pixelů), je animace vypnuta. Příklad popisované implementace je vidět v Ukázce kódu 27.

```

var interval;
var n=4; //number of tabs
var current_tab=0;
function startRotate() {
    if (!interval) {
        $("#index_tabs").tabs();
        interval = setInterval(function() {
            current_tab = (++current_tab < n ? current_tab : 0);
            $("#index_tabs").tabs( "option", "active", current_tab);
        }, 15 * 1000 /* 15sec */);
        $('#index_tabs').tabs("refresh");
    }
}
function stopRotate() {
    if (interval) {
        clearTimeout(interval);
        interval = false;
        $('.tab_item').attr("aria-hidden","false");
        $('.tab_item').css("display", "block");
    }
}
$(document).ready(function() {
    $('#index_tabs').tabs({
        hide: { effect: "fadeOut"},
        show: { effect: "fadeIn"}
    });
    if (!window.matchMedia("(max-width: 699px)").matches) {
        startRotate();
    } else {
        $('.tab_item').attr("aria-hidden","false");
        $('.tab_item').css("display", "block");
    }
}
$(window).resize(function() {
    if (window.matchMedia("(max-width: 699px)").matches) {
        stopRotate();
    } else {
        startRotate();
    }
});

```

**Ukázka kódu 27 – vypnutí animace jQuery UI tabs pro určitý breakpoint v JavaScriptu**

#### 4.4.11 Podpora starších prohlížečů

V kapitole 2.4.1 bylo zmíněno použití zvláštního souboru s kaskádovými styly za pomoci podmíněných komentářů v hlavičce HTML. Tento soubor byl určen pro starší verze Internet Exploreru, aby zajistil doručení stylů pro desktopové zobrazení, pokud byl použit přístup *mobile first*. Kdyby měl být tento soubor vytvářen ručně, nedávalo by jeho použití příliš smysl, protože by znamenalo udržovat dva soubory s kaskádovými styly. Při každé změně by bylo nutné provést úpravu na dvou místech. Při použití preprocesoru může být soubor se styly pro Internet Explorer generován automaticky [12].

Jake Archibald [35] popsal způsob, jak pomocí SASS preprocesoru automaticky vytvořit speciální CSS soubor pro Internet Explorer, ve kterém jsou obsaženy styly pro desktopové zobrazení bez *media queries*. Colin Bacon [36] poté popsal tento způsob implementace za pomoci LESS preprocesoru.

Protože byl při implantaci webového portálu Nasipolitici.cz využit preprocesor LESS, byl pro vytvoření speciálního CSS souboru použit postup podle Colina Bacona [36].

Do souboru `mixins.less`, který obsahuje LESS proměnné a *mixins* byla vložena definice dvou proměnných (viz Ukázka kódu 28).

```
@fix-mqs: false;
@old-ie: false;
```

#### Ukázka kódu 28 – definice LESS proměnných pro vytvoření stylů pro starší verze Internet Exploreru

Ve stejném souboru pak byly definovány *mixins* (Ukázka kódu 29), které byly následně použity všude místo přímého zápisu *media queries* (Ukázka kódu 30).

```
// Mixin that will create the fixed IE styles
.respond-min(@width; @rules) when (@old-ie = true) and (@fix-mqs >=
@width) {
    @rules();
}
// Mixin that will wrap our content in a media query
.respond-min(@width; @rules) when (@old-ie = false) {
    @media screen and (min-width: @width) {
        @rules();
    }
}
```

#### Ukázka kódu 29 – definice LESS *mixins* pro vytvoření stylů pro starší verze Internet Exploreru

```

body {
    font-size: 1.2em;
    .respond-min(@screen-md-min; {
        font-size: 1em;
    });
}

```

#### Ukázka kódu 30 – použití LESS *mixins* namísto *media queries*

Nakonec byl vytvořen LESS soubor `style-old_ie.less` (Ukázka kódu 31), ze kterého je generováno CSS pro starší verze Internet Exploreru, které obsahuje pouze import stylů hlavního LESS souboru a nastavení proměnných, které přepíší nastavení ze souboru `mixins.less`.

```

@import "style.less";
@old-ie: true;
@fix-mqs: @screen-lg-min;

```

#### Ukázka kódu 31 – obsah LESS souboru, ze kterého bude vytvořen soubor se styly pro starší verze Internet Exploreru

Ve výsledném CSS souboru `style-old_ie.css` (Ukázka kódu 32) jsou zahrnuty všechny CSS styly pro definovanou hodnotu *breakpointu* ve `style-old_ie.less` (proměnná `@fix-mqs`). Není zde použito žádné *media query*.

```

body {
    font-size: 1.2em;
    font-size: 1em;
}

```

#### Ukázka kódu 32 – výsledný CSS soubor pro starší verze Internet Exploreru při použití LESS *mixins* namísto *media queries* pro vytvoření stylů

Ve výsledném druhém CSS souboru `style.css` (Ukázka kódu 33) jsou obsaženy styly včetně definovaných *media queries*.

```

body {
    font-size: 1.2em;
}
@media screen and (min-width: @width) {
    body {
        font-size: 1em;
    }
}

```

#### Ukázka kódu 33 – výsledný CSS soubor pro všechny prohlížeče kromě starší verze Internet Exploreru při použití LESS *mixins* namísto *media queries* pro vytvoření stylů

## 5 Shrnutí a testování výsledného řešení

Testování je důležitou součástí implementace každého produktu. Následující podkapitoly představí popis provedeného testování během implementace a před odevzdáním hotového řešení, popis výsledků testování a shrnutí provedeného řešení responzivního designu webového portálu Nasipolitici.cz.

### 5.1 Testování

Již v průběhu implementace bylo řešení responzivního designu webového portálu Nasipolitici.cz průběžně testováno pomocí vývojářských nástrojů prohlížeče Chrome a Firefox a také na vybraných mobilních zařízeních (mobily a tablety). Dále byla průběžně získávána zpětná vazba od zadavatelů řešení.

Před předáním hotového řešení bylo provedeno důkladnější testování na vybraných fyzických zařízeních a také pomocí služby BrowserStack (dostupné z <http://www.browserstack.com>), kde byly získány screenshoty z různých zařízení pokrývající různé rozlišení a prohlížeče. Díky tomu bylo odhaleno pár drobných nedostatků, jako je například malá velikost některých prvků nedostačující pro pohodlné ovládání na dotykovém displeji nebo špatné zobrazení některých prvků ve starším Internet Exploreru. Tyto chyby, které byly díky testování odhaleny, mohly být ještě před předáním opraveny.

Bylo provedeno malé uživatelské testování. Pět uživatelů mělo postupně vyhledat určené věci na responzivní verzi webu Nasipolitici.cz na mobilním telefonu Samsung Galaxy S4, následně ještě dostali jeden úkol na neresponzivní verzi. Během testování byli sledováni a bylo zaznamenáváno jejich chování (Příloha č. 1). Mohli také sami říct, co se jim nelíbí, nebo s čím mají problém. Nakonec byli dotázáni, jestli by nějaké verzi dali přednost. A pokud ano, tak jaké jsou její výhody a nevýhody.

Je nutné počítat s tím, že uživatelé byli jistě ovlivněni pořadím prohlížených webů. Smyslem testování však nebylo porovnávat responzivní verzi proti neresponzivní, ale spíše to, jak uživatelé vnímají responzivní web a jestli jim vyhovuje. Cílem bylo také získat určitý odstup a nadhled, který samotný vývojář nemá a podívat se na web očima běžného návštěvníka. Tento účel splnila i velmi malá skupina uživatelů.

Všichni zúčastnění se shodli na tom, že responzivní design webu má na mobilním zařízení smysl, dali by mu přednost. Mezi uváděnými výhodami responzivního designu byla shodně všemi testovanými uživateli uváděna možnost prohlížení stránky bez nutnosti neustálého přibližování jednotlivých částí webu. Dále bylo mezi výhodami

uvedeno, že responzivní webová stránka vypadá jako by byla mobilní aplikací. Pozitivně hodnocena byla možnost ovládní prohlížení webu jednou rukou a také to, že není nutné se na mobil dívat tak zblízka jako u neresponzivních stránek, protože je responzivní web lépe čitelný a přehlednější. Jako nevýhoda responzivních webových stránek byla uvedena obava, že v této verzi webu nejsou dostupné všechny prvky, které jsou k dispozici v desktopové verzi.

Na základě pozorování chování uživatelů při práci s webem bylo zjištěno, že uživatelé při hledání konkrétní informace raději využijí vyhledávací políčko v hlavičce stránky a následně vybírají vhodnou stránku z nalezených výsledků, než aby procházeli web pomocí navigačního menu. Je možné předpokládat, že tento způsob pohybu uživatele na stránkách je spojen s uživatelským cílem najít konkrétní informaci. Ukazuje to na důležitost daného prvku a potvrzuje to rozhodnutí zachovat vyhledávací políčko dobře viditelné a přístupné pro všechna rozlišení.

Pokud uživatelé využili navigačního menu, ve všech případech bylo zaznamenáno, že si nevšimli podmenu, které bylo schované až pod odkazy na sociální sítě a nebylo ve viditelné části displeje. Na základě tohoto zjištění byla provedena úprava a podmenu bylo přesunuto výše tak, jak bylo popsáno v podkapitole 4.4.4.

Byla identifikována špatná přístupnost stránek týkajících se starších voleb (například Volby do Evropského parlamentu 2014 nebo Prezidentské volby 2013), protože v hlavní navigaci je pouze odkaz vedoucí na stránku posledních nebo aktuálních voleb. Pro zobrazení rozcestníku na jednotlivé volby je třeba v podmenu na stránce posledních voleb zvolit možnost „Volby“, která je však označena jako aktivní, takže uživatele nenapadne na ni kliknout. Na toto zjištění nebylo v rámci diplomové práce reagováno, protože se nejedná o problém responzivního designu. Je to však případný podnět pro další zlepšení webu Nasipolitici.cz.

## **5.2 Shrnutí výsledného řešení**

Povedlo se implementovat a nasadit responzivní design informačního portálu Nasipolitici.cz. Ten vychází z původního vzhledu webu. Je to z toho důvodu, že práce se zabývá především implementací responzivního designu a pro změnu designu by byl potřeba kvalitní grafický návrh nového vzhledu. Při změně stávajícího designu na responzivní je však vždy potřeba mít na paměti přístup *mobile first*, i když podle něj nebylo při návrhu designu webu postupováno. Web by neměl obsahovat zbytečné prvky navíc, které nepřinášejí žádnou informační hodnotu a zbytečně odvádí uživatelovu pozornost. V případě, že je při návrhu webu postupováno podle tohoto

přístupu, tedy od vzhladu na nejmenším displeji mobilního zařízení, není se třeba bát toho, že by web obsahoval zbytečné prvky navíc.

V průběhu implementace responzivního designu byla paralelně prováděna na produkční verzi webu běžná údržba webu a bylo přidáno několik nových sekcí, jakou byla například stránka věnovaná aktuálním volbám. Protože vývoj responzivního designu probíhal poměrně dlouho v samostatné vývojové větvi, bylo potřeba zasahovat do co nejméně částí webu, aby bylo po jeho dokončení možné sloučit responzivní verzi s produkční verzí. Kvůli tomu byla většina implementace řešena pouze pomocí CSS a JavaScriptu a do jednotlivých šablon HTML stránek bylo zasahováno co nejméně. Díky tomu je vidět, že lze responzivní design implementovat především na úrovni kaskádových stylů s využitím JavaScriptu.

Při testování na různých zařízeních byly odhaleny různé drobné chyby. V Internet Exploreru 8 byla nalezena chyba v implementaci použití jQuery Tabs, která mohla být opravena ještě před nasazením. To jen dokládá důležitost testování před nasazením responzivního designu, které je potřeba nepodceňovat.

Před samotnou implementací se také jako přínosný ukázal nástroj Google Analytics, který může pomoci při rozhodování o investici do responzivního designu webu. Ukazuje, že návštěvníci z mobilních zařízení na webovou stránku chodí a to i přesto, že není responzivní a nenabízí jim tak pohodlné prohlížení. Po nasazení responzivního designu počet návštěvníků z mobilních zařízení tedy ještě může vzrůst.

Z provedeného uživatelského testování nelze usuzovat žádné obecné závěry o chování uživatelů na webu, k tomu by bylo potřeba s pomocí vhodné metodiky pracovat s mnohem větším souborem uživatelů. Nelze také na základě toho vyvozovat jednoznačný závěr, že má responzivní verze webu smysl. Relevantnější údaje podporující využití responzivního webu by mohla ukázat data z Google Analytics po nějaké době od nasazení responzivní verze.

Uživatelské testování pomohlo odhalit některé nedostatky, které snižovaly pohodlí při procházení webu. Samotný vývojář se dívá na web jinak, protože ho zná z mnoha úhlů a také většinou o webu přemýšlí jinak, než kdyby byl návštěvníkem daného webu. Testování uživatelé nebyli zatíženi tímto pohledem a viděli stránku poprvé, takže nebyli nijak ovlivněni. Uživatelské testování pomůže podívat se na webovou stránku z jiného pohledu, jinýma očima.

## 6 Závěr

Práce představila responzivní přístup, jeho části a obecná doporučení, která je dobré znát před samotnou tvorbou responzivního designu. Byl popsán postup implementace od návrhu webu po testování, který může sloužit jako návod při tvorbě responzivního designu jiného webového portálu.

Dále se diplomová práce zabývala vytvořením responzivního designu webového portálu [www.nasipolitici.cz](http://www.nasipolitici.cz). Ten se podařilo navrhnout, implementovat a také byl produkčně nasazen provozovateli webu.

Prvotní analýza potvrdila předpoklad, že návštěvnost webového portálu z mobilních zařízení není malá (14 %). Byly také identifikovány některé metriky v nástroji Google Analytics (například Míra okamžitého opuštění), které lze využít pro analýzu návštěvnosti před a po implementaci responzivního designu webu.

Ve fázi návrhu, která předcházela samotné implementaci, bylo zajímavé to, že se vycházelo z původního designu webu. Při návrhu responzivního designu bývá popisován postup, kdy se vytváří celý koncept nového webu včetně nového designu. V případě, který popisovala tato práce, se vycházelo z původního designu webu. Ukázalo se, že při dodržení doporučovaných postupů, jako je přístup *mobile first*, je i tento způsob velmi dobře použitelný.

Při diskuzích o responzivní verzi webu s občanským sdružením Naši politici o. s. se velmi dobře osvědčily vytvořené responzivní prototypy. Potvrdilo se, že jsou velmi užitečné právě tím, že nejsou pouze statické.

Během implementace byly identifikovány různé prvky, které se nacházejí na webových stránkách. V této diplomové práci je popsána možnost jejich začlenění do responzivní stránky a nastíněna jejich implementace. Za zmínku stojí určitě problematika navigace, která se i díky podmenu stává velmi komplexním prvkem. Řešena byla také podpora starších prohlížečů, kde bylo zajímavým způsobem s využitím CSS preprocesoru vyřešeno zobrazení desktopové varianty webu ve starších verzích Internet Exploreru.

Tato práce však rozhodně není vyčerpávajícím popisem využití responzivního přístupu k tvorbě webu. Zajímavé by bylo věnovat pozornost responzivním obrázkům. Na webovém portálu [www.nasipolitici.cz](http://www.nasipolitici.cz) bylo řešeno pouze nenačítání velkých obrázků



na mobilních zařízeních na vybraných stránkách, kde by znamenaly zbytečnou zátěž při využití mobilního internetu. Zde by byl prostor pro nasazení komplexního řešení, které by optimalizovalo načítání velikostí obrázků napříč celým webovým portálem. Kromě omezení velikosti stránky by také mohlo řešit podporu retina displejů, tedy displejů s vysokým rozlišením, pro které by mohl být načítán obrázek ve vyšším rozlišení než pro běžné displeje.

Prostor pro další výzkum se také nachází v oblasti odlišné práce s webovými stránkami v závislosti na používaném zařízení. Pokud platí předpoklad, že uživatelé využívají prohlížení webových stránek na chytrém telefonu především k hledání konkrétních informací, bylo by zajímavé tuto myšlenku promítnout do responzivního designu webu. A zjistit, jakým způsobem by se z tohoto pohledu měla lišit struktura a zobrazení webu na různých zařízeních.

Práci lze využít jako základ pro implementaci responzivního designu webového portálu, ale také jako zdroj informací při rozhodování o smyslu a následném návrhu responzivního designu.

## Seznam použité literatury

- [1] SNELL, Jeremy. Flexible Everything: Getting Responsive With Web Design. *Computers in Libraries*. 2013, vol. 33, no. 3, s. 12-16, Apr. 2013. ISSN 10417915.
- [2] Comparison from Dec 2008 to Dec 2014. *StatCounter: GlobalStats*. [online]. © 1999-2015 [cit. 2015-01-25]. Dostupné z: <http://gs.statcounter.com/#desktop+mobile-comparison-ww-monthly-200812-201412>.
- [3] Comparison from Dec 2008 to Dec 2014. *StatCounter: GlobalStats*. [online]. © 1999-2015 [cit. 2015-01-24]. Dostupné z: <http://gs.statcounter.com/#desktop+mobile-comparison-CZ-monthly-200812-201412>.
- [4] FROMMER Dan. HISTORY LESSON: How The iPhone Changed Smartphones Forever. *Business Insider*. [online]. Jun. 6, 2011 [cit. 2015-04-06]. Dostupné z: <http://www.businessinsider.com/iphone-android-smartphones-2011-6?op=1>.
- [5] RADWANICK Sarah. 5 Years Later: A Look Back at the Rise of the iPhone. *comScore, Inc.* [online]. June 29, 2012 [cit. 2015-04-06]. Dostupné z: <http://www.comscore.com/Insights/Blog/5-Years-Later-A-Look-Back-at-the-Rise-of-the-iPhone>.
- [6] Global Web Stats – March 2009. *W3Counter*. [online]. 03/31/2009 [cit. 2015-04-01]. Dostupné z: <http://www.w3counter.com/globalstats.php?year=2009&month=3>.
- [7] Top 10 Desktop Screen Resolutions on Mar 2009. *StatCounter GlobalStats*. [online]. © 1999–2015 [cit. 2015-04-01]. Dostupné z: <http://gs.statcounter.com/#desktop-resolution-ww-monthly-200903-200903-bar>.
- [8] FRAIN Ben. *Responsive Web Design with HTML5 and CSS3*. Birmingham: Packt Publishing, 2012. ISBN 978-1-84969-318-9.
- [9] MARCOTTE Ethan. Toffee-nosed. *Unstoppable Robot Ninja*. [online]. © 2015 [cit. 2015-01-25]. Dostupné z: <http://gs.statcounter.com/#desktop-resolution-ww-monthly-200903-200903-bar>.
- [10] KADLEC Tim. *Implementing Responsive Design: Building sites for an anywhere, everywhere web*. Barkeley: New Riders, 2013. ISBN 978-0-321-82168-3.
- [11] MARCOTTE Ethan. Responsive Web Design. *A List Apart*. [online]. 2010, vol. 306 [cit. 2015-03-28]. ISSN 1534-0295. Dostupné z : <http://alistapart.com/article/responsive-web-design>.

- [12] JEHL Scott. *Responsible Responsive Design*. New York: A Book Apart, 2014. ISBN 978-1-9375571-6-4.
- [13] Media Queries. W3C. [online]. 19 June 2012 [cit. 2015-01-31]. Dostupné z: <http://www.w3.org/TR/2012/REC-css3-mediaqueries-20120619>.
- [14] MARCOTTE Ethan. *Responsive Web Design*. 2nd edition. New York: A Book Apart, 2014. ISBN 978-1-9375571-8-8.
- [15] CSS3 Media Queries. *Can I use...: Support tables for HTML5, CSS3, etc.* [online]. [2015] [cit. 2015-03-28]. Dostupné z: <http://caniuse.com/#feat=css-mediaqueries>.
- [16] FREDERICK, Kathryn. Responsive Web Design 101: How to Make Your Site Responsive. *Computers in Libraries*. 2013, vol. 33, no. 6, s. 11-14, July-August 2013. ISSN 10417915.
- [17] HTML 5.1. W3C. [online]. 23 March 2015 [cit. 2015-04-03]. Dostupné z: <http://www.w3.org/TR/2015/WD-html51-20150323>.
- [18] PORTIS Eric. Responsive Images in Practice. *A List Apart*. [online]. 2014, vol. 407 [cit. 2015-04-03]. ISSN 1534-0295. Dostupné z: <http://alistapart.com/article/responsive-images-in-practice>.
- [19] Srcset attribute. *Can I use...: Support tables for HTML5, CSS3, etc.* [online]. [2015] [cit. 2015-04-03]. Dostupné z: <http://caniuse.com/#feat=srcset>.
- [20] VINH Khoi. *Ordering Disorder: Grid Principles for Web Design*. Berkeley: New Riders, 2011. ISBN 978-0321703538.
- [21] FRANZ Laura. Size Matters: Balancing Line Length And Font Size In Responsive Web Design. *Smashing Magazine*. [online]. September 29th, 2014 [cit. 2015-04-02]. Dostupné z: <http://www.smashingmagazine.com/2014/09/29/balancing-line-length-font-size-responsive-web-design>.
- [22] BRINGHURST Robert. *The Elements of Typographic Style*. Vancouver: Hartley and Marks Publishers, 2004. ISBN 978-0881792065.
- [23] MALÝ Martin. LESS: stejné CSS za méně peněz. *Zdroják*. [online]. 21. 10. 2010 [cit. 2015-04-04]. Dostupné z: <http://www.zdrojak.cz/clanky/less-stejne-css-za-mene-penez>.
- [24] JOBSEN Bass a Amin MEYGHANI. *Less Web Development Cookbook*. Birmingham: Packt Publishing, 2015. ISBN 978-1783981489.

- [25] FRAIN Ben. *Sass and Compass for Designers*. Birmingham: Packt Publishing, 2013. ISBN 978-1849694544.
- [26] KRAMER Jen. Responsive Design Frameworks: Just Because You Can, Should You?. *Smashing Magazine*. [online]. February 19th, 2014 [cit. 2015-04-03]. Dostupné z: <http://www.smashingmagazine.com/2014/02/19/responsive-design-frameworks-just-because-you-can-should-you>.
- [27] PICK Mike. Surveying the Big Screen. *A List Apart*. [online]. 2013, vol. 386 [cit. 2015-04-03]. ISSN 1534-0295. Dostupné z: <http://alistapart.com/article/surveying-the-big-screen>.
- [28] T Anthony. Finger-Friendly Design: Ideal Mobile Touchscreen Target Sizes. *Smashing Magazine*. [online]. February 21st, 2012 [cit. 2015-04-03]. Dostupné z: <http://www.smashingmagazine.com/2012/02/21/finger-friendly-design-ideal-mobile-touchscreen-target-sizes>.
- [29] AdMob Tablet Survey. *Google*. [online]. March 2011 [cit. 2015-04-06]. Dostupné z: <http://services.google.com/fh/files/blogs/AdMob%20-%20Tablet%20Survey.pdf>.
- [30] HARB Eva, Paul KAPPELLARI, Steven LUONG a Norbert SPOT. Responsive Web Design. [online]. 06 Dec 2011 [cit. 2015-01-11]. Dostupné z: <http://courses.iicm.tugraz.at/iaweb/surveys/ws2011/g3-survey-resp-web-design.pdf>.
- [31] WROBLEWSKI Luke. Mobile First. *LukeW Ideation + Design*. [online]. 3.11.2009 [cit. 2015-03-16]. Dostupné z: <http://www.lukew.com/ff/entry.asp?933>.
- [32] WROBLEWSKI Luke. *Mobile First*. New York: A Book Apart, 2011. ISBN 978-1-937557-02-7.
- [33] KARDYS Dennis. *Mobile Web Triage: What to Do When Your Website Is Not Mobile Friendly*. Berkeley: Peachpit Press, 2014. ISBN 978-0-13-381752-2.
- [34] OTTO Mark, Jacob THORNTON a Bootstrap contributors. CSS. *Bootstrap*. [online]. [2015] [cit. 2015-01-15]. Dostupné z: <http://getbootstrap.com/css>.
- [35] ARCHIBALD Jake. IE-friendly mobile-first CSS with Sass 3.2. [online]. [cca. 2012] [cit. 2015-04-15]. Dostupné z: <http://jakearchibald.github.io/sass-ie>.
- [36] BACON Colin. Life is now easier, Create media query mixins with rulesets – LESS CSS. *I am Bacon*. [online]. 2014-07-04 [cit. 2015-04-15]. Dostupné z: <http://www.iambacon.co.uk/blog/life-is-now-easier-create-media-query-mixins-with-rulesets-less-css>.

## Seznam ukázek kódu

<b>Ukázka kódu 1</b> – příklad zápisu media queries v CSS .....	11
<b>Ukázka kódu 2</b> – použití media queries v hlavičce HTML souboru.....	11
<b>Ukázka kódu 3</b> – použití podmíněných komentářů k odlišení CSS stylů pro starší verze Internet Exploreru.....	12
<b>Ukázka kódu 4</b> – dynamická šířka obrázku pomocí CSS.....	13
<b>Ukázka kódu 5</b> – příklad stejného kódu napsaného v CSS a LESS.....	18
<b>Ukázka kódu 6</b> – meta tag viewport v hlavičce HTML dokumentu .....	22
<b>Ukázka kódu 7</b> – použití pravidla viewport v CSS .....	22
<b>Ukázka kódu 8</b> – zobrazování a schovávání menu pomocí JavaScriptu .....	34
<b>Ukázka kódu 9</b> – nastavení stylů pro zobrazování a schovávání menu v LESS.....	34
<b>Ukázka kódu 10</b> – nastavení stylů pro dvouřádkovou navigaci v LESS.....	35
<b>Ukázka kódu 11</b> – detekce podpory JavaScriptu pomocí odebrání třídy .....	35
<b>Ukázka kódu 12</b> – nastavení stylů pro zobrazení navigace, pokud není podporován JavaScript v LESS .....	36
<b>Ukázka kódu 13</b> – vytvoření mřížky za pomoci Bootstrap mixins v LESS.....	37
<b>Ukázka kódu 14</b> – ukázka struktury HTML navigace na webu Nasipolitici.cz.....	39
<b>Ukázka kódu 15</b> – implementace změny pořadí prvků v navigaci na základě šířky zobrazení na webu Nasipolitici.cz .....	40
<b>Ukázka kódu 16</b> – část HTML kódu formuláře pro registraci nového uživatele na webu Nasipolitici.cz.....	41
<b>Ukázka kódu 17</b> – ukázka LESS pro různé uspořádání popisků a editačních políček ve formuláři při různých šířkách zobrazení na webu Nasipolitici.cz .....	42
<b>Ukázka kódu 18</b> – odkaz, do kterého je vkládán obrázek pomocí JavaScriptu v HTML.....	44
<b>Ukázka kódu 19</b> – nahrazení textu odkazu obrázkem v JavaScriptu .....	44
<b>Ukázka kódu 20</b> – příklad vykreslení grafu s relativně definovanou velikostí pomocí Google Chart Api na webu Nasipolitici.cz.....	45
<b>Ukázka kódu 21</b> – definice stylů pro kontejner obsahující responzivní graf .....	46

<b>Ukázka kódu 22</b> – animace posunu viditelné části stránky v JavaScriptu .....	47
<b>Ukázka kódu 23</b> – původní struktura dat typu JSON pro jsTree verze 1 .....	48
<b>Ukázka kódu 24</b> – nová struktura dat typu JSON pro jsTree verze 3 .....	48
<b>Ukázka kódu 25</b> – původní implementace inicializace jsTree verze 1 .....	49
<b>Ukázka kódu 26</b> – nová implementace inicializace jsTree verze 3 .....	49
<b>Ukázka kódu 27</b> – vypnutí animace jQuery UI tabs pro určitý breakpoint v JavaScriptu.....	52
<b>Ukázka kódu 28</b> – definice LESS proměnných pro vytvoření stylů pro starší verze Internet Exploreru.....	53
<b>Ukázka kódu 29</b> – definice LESS mixins pro vytvoření stylů pro starší verze Internet Exploreru.....	53
<b>Ukázka kódu 30</b> – použití LESS mixins namísto media queries.....	54
<b>Ukázka kódu 31</b> – obsah LESS souboru, ze kterého bude vytvořen soubor se styly pro starší verze Internet Exploreru.....	54
<b>Ukázka kódu 32</b> – výsledný CSS soubor pro starší verze Internet Exploreru při použití LESS mixins namísto media queries pro vytvoření stylů.....	54
<b>Ukázka kódu 33</b> – výsledný CSS soubor pro všechny prohlížeče kromě starší verze Internet Exploreru při použití LESS mixins namísto media queries pro vytvoření stylů.....	54

## Seznam obrázků

<b>Obrázek 1</b> – příklad layoutu webu rozděleného na několik funkčních celků.....	6
<b>Obrázek 2</b> – ukázka statického rozložení webu .....	7
<b>Obrázek 3</b> – ukázka flexibilního rozložení webu .....	8
<b>Obrázek 4</b> – stanovené priority jednotlivých prvků na webu Nasipolitici.cz (homepage).....	28
<b>Obrázek 5</b> – stanovené priority jednotlivých prvků na webu Nasipolitici.cz (detail politika).....	29
<b>Obrázek 6</b> – zobrazení všech navigačních odkazů i na úzké zobrazovací ploše.....	31
<b>Obrázek 7</b> – zobrazení všech navigačních odkazů i na úzké zobrazovací ploše.....	31
<b>Obrázek 8</b> – příklad vysouvací navigace na webu.....	32
<b>Obrázek 9</b> – jednořádková navigace na webu Nasipolitici.cz .....	33
<b>Obrázek 10</b> – navigace na webu Nasipolitici.cz poskládaná pod sebe .....	33
<b>Obrázek 11</b> – otevřené a zavřené menu na webu Nasipolitici.cz .....	34
<b>Obrázek 12</b> – dvouřádková navigace na webu Nasipolitici.cz.....	35
<b>Obrázek 13</b> – navigace včetně podmenu při širokém zobrazení na webu Nasipolitici.cz .....	38
<b>Obrázek 14</b> – přesunutí podmenu v rámci úzkého zobrazení navigace na webu Nasipolitici.cz .....	39
<b>Obrázek 15</b> – registrační formulář webu Nasipolitici.cz při různých šířkách zobrazení.....	41
<b>Obrázek 16</b> – zobrazení velkého obrázku na webu Nasipolitici.cz, který po zmenšení stránky přestává být přehledný .....	43
<b>Obrázek 17</b> – zobrazení odkazu nahrazující obrázek při menších zobrazeních na webu Nasipolitici.cz.....	44
<b>Obrázek 18</b> – graf vykreslený pomocí Google Chart Api na webu Nasipolitici.cz.....	45
<b>Obrázek 19</b> – zobrazení stromu (jsTree) pro užší zobrazovací plochu na stránce parametrického vyhledávání na webu Nasipolitici.cz.....	46

<b>Obrázek 20</b> – zobrazení stromu (jsTree) na stránce parametrického vyhledávání na webu Nasipolitici.cz .....	47
<b>Obrázek 21</b> – prvek s rotujícími odkazy na články ( <i>carousel</i> ) na webu Nasipolitici.cz...50	
<b>Obrázek 22</b> – původní vzhled prvku se záložkami na domovské stránce webu Nasipolitici.cz .....	50
<b>Obrázek 23</b> – vzhled prvku se záložkami na domovské stránce webu Nasipolitici.cz při úzkém zobrazení.....	51

## **Seznam tabulek**

<b>Tabulka 1</b> – seznam vybraných vlastností media queries dle specifikace W3C [13] .....	10
<b>Tabulka 2</b> – přehled vybraných metrik rozdělených podle typu zařízení .....	25

## **Seznam grafů**

<b>Graf 1</b> – trend přístupů z mobilů a tabletů 2008–2014, převzato z [2].....	2
<b>Graf 2</b> – trend přístupů z mobilů a tabletů v České republice 2008–2014, převzato z [3] .....	3
<b>Graf 3</b> – nárůst počtu majitelů chytrých telefonů v USA 2007–2012, převzato z [5] .....	4
<b>Graf 4</b> – podíl návštěv webového portálu Nasipolitici.cz z jednotlivých zařízení k celkovému počtu návštěv.....	24

## **Seznam příloh**

Příloha č. 1 .....	67
Příloha č. 2 .....	69



## Výsledky provedeného uživatelského testování

Seznam zadaných úkolů na responzivní verzi:

- Zjistit, kdo stojí za projektem Nasipolitici.cz.
- Zobrazit poslední kauzu zveřejněnou na webu.
- Zobrazit seznam kandidátů v prezidentských volbách.
- Najít životopis vybraného politika.

Seznam zadaných úkolů na neresponzivní verzi:

- Najít seznam veřejných zakázek a zjistit, která měla nejvyšší vysoutěženou cenu.

Celkem bylo testováno pět uživatelů.

V následujících tabulkách je uveden vždy počet uživatelů, u kterých bylo zaznamenáno popsané chování, nebo počet těch, kteří vyslovili popsaný názor.

Popis chování a poznámky uživatelů	Počet uživatelů
Používá integrované Google vyhledávání v hlavičce pro pohyb na webu.	4
Nečeká, že by mohlo být menu delší, než je viditelná část, nebo že by mohlo pokračovat pod odkazy na sociální sítě (podmenu bylo umístěno až pod odkazy na sociální sítě) – pouze uživatelé, kteří alespoň jednou využili navigačního menu.	3
V detailu politika klikne na odkaz Životopis v domnění, že se mu zobrazí strukturovaný životopis.	3
Alespoň jednou využil navigačního menu.	3
Myslí si, že nápis Vyhledávání politiků v hlavičce je pouze popis vyhledávacího políčka. Nepozná, že jde o odkaz na vyhledávání politiků.	2
Nepoužívá vyhledávání a má problém nalézt prezidentské volby.	2
Na homepage dole chce kliknout na "O nás", což je pouze nadpis a ne odkaz.	1
V parametrickém vyhledávání politiků je našeptávání moc nízké, špatně se na to klikne.	1

Popis výhod a nevýhod responzivního webu	Počet uživatelů
Říká, že responzivní verze má smysl.	5
Pozitivně hodnotí, že nemusí stránku neustále přibližovat a oddalovat.	5
Stránka mu přijde přehlednější než tomu je v případě neresponzivního webu.	4
Mobilní telefon si nemusí dávat tak blízko, jako na neresponzivním webu.	3
Pozitivně hodnotí podobnost responzivního webu s mobilní aplikací.	1
Říká, že nevěří mobilním verzím webu, protože si myslí, že tam není vše, na co jsou zvyklí z desktopové verze.	1

**Obsah přiloženého CD**

<b>Janackova_Marie_diplomova_prace.pdf</b>	Elektronická verze diplomové práce
<b>src/puvodni</b>	Původní zdrojové kódy klíčových Smarty šablon a CSS
<b>src/nove</b>	Upravené zdrojové kódy klíčových Smarty šablon a CSS/LESS



**UNIVERZITA HRADEC KRÁLOVÉ**  
**Fakulta informatiky a managementu**  
Rokitanského 62, 500 03 Hradec Králové, tel: 493 331 111, fax: 493 332 235

## Zadání k závěrečné práci

Jméno a příjmení studenta:

**Marie Vlachová**

Obor studia:

Informační management (5)

Jméno a příjmení vedoucího práce:

**Pavel Kříž**

Název práce:

**Responzivní design webového portálu**

Název práce v AJ:

Responsive Design of Web Portal

Podtitul práce:

Podtitul práce v AJ:

Cíl práce: Navrhnout a implementovat responzivní design pro existující webový portál. Práce se zaměří na obecnou problematiku tvorby responzivních webů a ukázková realizace bude provedena na portálu [www.nasipolitici.cz](http://www.nasipolitici.cz).

Osnova práce:

1. Úvod
2. Cíl
3. Teoretická východiska pro výslednou práci
4. Analýza současného stavu webu Nasipolitici.cz
5. Implementace responzivního designu
6. Shrnutí a testování výsledného řešení
7. Závěr

Projednáno dne: 14.10.14

Podpis studenta

Podpis vedoucího práce