

# Možnosti využití CSS frameworků při návrhu moderního webu

Melánie Ležáková

---

Bakalářská práce  
2020



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

**ZADÁNÍ BAKALÁŘSKÉ PRÁCE**  
(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Melánie Ležáková**  
Osobní číslo: **A17131**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Softwarové inženýrství**  
Forma studia: **Prezenční**  
Téma práce: **Možnosti využití CSS frameworků při návrhu moderního webu**  
Téma práce anglicky: **The Possibilities of Using CSS Frameworks for Modern Web Design**

### Zásady pro vypracování

1. Prostudujte a popište vybrané CSS frameworky vhodné pro návrh moderního uživatelského prostředí webové aplikace. Věnujte se nástroji Bootstrap, ale uveďte i alternativní frameworky.
2. Vypracujte stručné srovnání vybraných nástrojů z hlediska nabízených funkcí (responzivní design, grid systém, komponentový design) a z hlediska rozšířenosti.
3. Vyberte některý z popsaných nástrojů pro praktickou implementaci uživatelského rozhraní pro moderní webovou prezentaci a popište způsob nasazení do projektu.
4. Věnujte se také další možné vlastní konfiguraci daného nástroje nebo konfiguraci sestavení produkčních souborů.
5. Pomocí vybraného nástroje proveďte implementaci šablony moderní responzivní webové prezentace a otestujte ji na různých zařízeních (mobilní telefon, tablet, desktop).

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. CROCKFORD, Douglas. JavaScript: the good parts. Sebastopol: O'Reilly, 2008. ISBN 05-965-1774-2.
2. OSMANI, Addy. Learning JavaScript design patterns: the good parts. Sebastopol, CA: O'Reilly Media, 2012. ISBN 14-493-3181-5.
3. Vue.js: up and running: building accessible and performant web apps. Sebastopol, California: O'Reilly Media, [2018]. ISBN 1491997249.
4. VRÁNA, Jakub. 1001 tipů a triků pro PHP. Brno: Computer Press, 2010. ISBN 978-80-251-2940-1.
5. ROBSON, Stuart. Jump Start Sass. Melbourne: SitePoint Pty, 2016. ISBN 9780994182678.
6. Introduction ? Bootstrap. Bootstrap ? The most popular HTML, CSS, and JS library in the world. [online]. Dostupné z: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
7. Sass: Documentation. Sass: Syntactically Awesome Style Sheets [online]. Copyright ? 2006 [cit. 29.09.2019]. Dostupné z: <https://sass-lang.com/documentation>

Vedoucí bakalářské práce:

**Ing. Radek Vala, Ph.D.**

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: 28. listopadu 2019  
Termín odevzdání bakalářské práce: 15. května 2020



---

**doc. Mgr. Milan Adámek, Ph.D.**  
děkan

**prof. Mgr. Roman Jašek, Ph.D.**  
ředitel ústavu

Ve Zlíně dne 9. prosince 2019

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Melánie Ležáková, v. r.

## **ABSTRAKT**

Tato práce se věnuje srovnání CSS frameworků a následnému nasazení do projektu v praxi. V teoretické části jsou rozebrány základní údaje ohledně jednotlivých frameworků, jejich historie, odlišnosti a technologie s nimi související. V praktické části proběhlo srovnání vybraných frameworků a zvolení jednoho z nich k implementaci do projektu. Dále je v této části popsána vlastní konfigurace daného nástroje. Výstupem praktické části je webová stránka, jejíž responzivita je otestována na vícero zařízeních.

Klíčová slova: CSS framework, kaskádové styly, Bootstrap

## **ABSTRACT**

This work deals with the comparison of CSS frameworks and subsequent deployment in the project in practice. The theoretical part discusses the basic data about individual frameworks, their history, differences and technologies related to them. In the practical part there was a comparison of selected frameworks and the selection of one of them for implementation in the project. This section also describes the actual configuration of the tool. The output of the practical part is a website whose responsiveness has been tested on multiple devices.

Keywords: CSS frameworks, cascade styles, Bootstrap

Tímto bych chtěla poděkovat mému vedoucímu práce Ing. Radkovi Valovi, Ph.D., za poskytnutí cenných rad a věcných připomínek, které mi pomohly tuto práci zkompletovat, a taktéž za vstřícnost při konzultacích.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>11</b>
<b>1 HTML</b> .....	<b>12</b>
1.1 STRUČNÁ HISTORIE .....	12
1.2 VÝVOJ HTML .....	13
1.3 POUŽITÍ .....	14
<b>2 CSS</b> .....	<b>15</b>
2.1 HISTORIE CSS .....	15
2.2 POUŽITÍ CSS .....	16
<b>3 CSS PREPROCESORY</b> .....	<b>18</b>
<b>4 CSS FRAMEWORKY</b> .....	<b>19</b>
4.1 CO JE TO FRAMEWORK.....	19
4.2 DRUHY A DĚLENÍ FRAMEWORKŮ .....	19
4.3 PŘÍKLADY CSS FRAMEWORKŮ .....	20
4.3.1 Foundation.....	20
4.3.2 Materialize.....	27
4.3.3 Bootstrap .....	31
<b>II PRAKTICKÁ ČÁST</b> .....	<b>34</b>
<b>5 ÚVOD DO PRAKTICKÉ ČÁSTI</b> .....	<b>35</b>
<b>6 SROVNÁNÍ FRAMEWORKŮ</b> .....	<b>36</b>
6.1 GRID.....	36
6.2 DOKUMENTACE .....	38
6.3 PODPORA.....	39
6.4 OBLÍBENOST.....	40
6.5 SHRNUÍ.....	41
<b>7 NÁVRH</b> .....	<b>43</b>
<b>8 NASAZENÍ FRAMEWORKU DO PROJEKTU A REALIZACE</b> .....	<b>44</b>
8.1 ZALOŽENÍ PROJEKTU .....	44
8.2 STARTER TEMPLATE .....	45
8.3 NAVIGAČNÍ MENU .....	45
8.4 SLIDER .....	49
8.5 GRID.....	50
8.6 TABULKY .....	52
8.7 VLASTNÍ KONFIGURACE FRAMEWORKU BOOTSTRAP.....	53

<b>9</b>	<b>OTESTOVÁNÍ RESPONZIVITY.....</b>	<b>59</b>
	<b>ZÁVĚR .....</b>	<b>62</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>63</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>67</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>68</b>
	<b>SEZNAM TABULEK.....</b>	<b>69</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>70</b>



## ÚVOD

Internet je v dnešní době dle zveřejněných dat využíván 59 % celosvětové populace [1]. Toto číslo každým rokem stoupá a s tím souvisí i zvyšující se nároky jak uživatelů, tak i kodérů na nové technologie, které je možno používat. Internetové stránky dnes dosahují vysoké návštěvnosti a díky tomu webové technologie patří mezi nejrychleji rozvíjející se počítačové oblasti. Na základě těchto skutečností se využívané technologie rok od roku rapidně mění a je žádoucí se tomu přizpůsobit z hlediska budoucího pokroku.

Tato bakalářská práce se zabývá tvorbou webových stránek a zaměřuje se na nasazení CSS frameworků v praxi a jejich vliv na danou problematiku z hlediska kodérů. V současné době počet webových stránek stále rapidně roste a již ke dnešnímu dni je uvedena existence více než 1,7 bilionu stránek [2]. Zájem o webovou prezentaci je obrovský ať už k soukromým, komerčním, anebo interním účelům. Na základě toho musí dnes kodéři přistupovat k jiným postupům – a to zejména k těm rychlejším, efektivnějším a modernějším vzhledem k tomu, že konkurence je veliká.

Uživatelé se dnes nespokojí se statickou a obyčejně vypadající stránkou, a proto je potřeba návštěvníka zaujmout a dát mu co nejlepší zkušenost z jeho pobytu na ní. Z toho důvodu se zadavatelé snaží vybírat si jen ty kodéry, jejichž stránky vypráví příběh a lze předpovědět nárůst návštěvnosti, což vede ke spokojenosti zadavatele. Na základě zvýšených nároků zadavatele jak v dynamické oblasti, tak v oblasti uživatelského rozhraní (dále jako UI<sup>1</sup>) se dnešní kodéři musí dennodenně zamýšlet, jak práci udělat co nejvíce zajímavě a efektivně, z čehož plyne, že čím méně času stráví na základních záležitostech při každém startu nového projektu, tím více času jim zbyde na samotnou funkcionalitu a zdokonalování UI<sup>1</sup> nebo UX<sup>2</sup>. Mimo jiné, i tuto problematiku řeší CSS frameworky a patří to mezi jeden z hlavních bodů, kterými se tato práce bude zabývat.

V následujících kapitolách teoretické části bude popsáno, jaké frameworky jsou vhodné pro návrh moderního uživatelského prostředí webové aplikace. Každý kodér má své vlastní požadavky na jednotlivé nástroje, a proto bude kladen důraz i na výhody, nevýhody, rozdíly a to, v čem se liší například od nejznámějšího a nejvíce mezi kodéry používaného frameworku Bootstrap, který je implementován i na platformě CMS Wordpress<sup>3</sup>. Použití

---

<sup>1</sup> User interface – uživatelské rozhraní

<sup>2</sup> User experience – uživatelská zkušenost

<sup>3</sup> Wordpress – je svobodný open source redakční publikační systém napsaný v PHP a MySQL, který je vyvíjený pod licencí GNU GPL

CSS frameworku by mělo usnadnit práci, výrazně ušetřit čas a případně i ulehčit kódování plně responzivního designu, který je dnes nadměru žádaný, ba dokonce i nutný vzhledem k tomu, že stránky, které nejsou responzivní, se při vyhledávání zobrazují ve výsledcích na mnohem nižších pozicích než stránky, které jsou responzivní. Z tohoto vyplývá, že dochází k výraznému a nechtěnému ovlivnění marketingu vlivem webové prezentace.

V praktické části bude realizováno nasazení jednoho z vybraných nástrojů do reálného projektu a popsáno, jakým byl přínosem při tvorbě webové stránky. Praktickým výsledným výstupem bakalářské práce je funkční webová stránka, jejíž správná responzivita je otestovaná na dostupných zařízeních a dále na simulátoru.

## **I. TEORETICKÁ ČÁST**

## 1 HTML

HTML je zkratkou pro slovní spojení hypertextový značkovací jazyk (Hypertext Markup Language). Jedná se o textový soubor, ve kterém je napsáno tělo webové stránky, jejíž veškerý obsah se zobrazuje ve webovém prohlížeči. Hypertext slouží pro označování odkazu, který propojuje odkazy v rámci jednoho a více webů. Napsaná struktura webu pomocí tagů určuje, jak se bude stránka zobrazovat, nikoliv jak bude designově nebo funkčně řešená. K ovlivňování vzhledu a funkcionality slouží dnes již nepostradatelné technologie CSS a JavaScript. Obsah HTML je psán pomocí tagů, díky nimž je možné zobrazit na stránce text, obrázky, nebo i video a jiné. Samotné tagy se ve webovém prohlížeči nezobrazují a uživatel je nevidí. K zobrazení tagů lze použít nástroje k tomu určené, jako třeba DevTools, které jsou implementované v prohlížeči. Bližší informace o DevTools na stránce <https://developers.google.com/web/tools/chrome-devtools>.

### 1.1 Stručná historie

Za dnešní webové stránky můžeme vděčit britskému vědci jménem Tim Berners-Lee [3], který navrhl zmiňovaný hypertextový značkovací jazyk (HTML) včetně protokolu HTTP [4]. Tento protokol je nezbytný ke komunikaci s WWW servery za účelem přenášení hypertextových dokumentů HTML a dnes již i XML. Jeho práce v základu navázala na teoretickou myšlenku publikovanou roku 1945 v článku, jehož autorem byl americký inženýr a vědec Vannevar Bush [5]. WWW neboli celosvětová síť byla poprvé spuštěna 6. srpna 1991 [6]. Na první existující stránce nebylo zobrazeno nic jiného než text s informacemi o této službě a hypertextové odkazy s tím spojené. K zobrazování webových stránek se využíval prohlížeč s názvem WorldWideWeb, jehož autorem nebyl nikdo jiný, než Tim Berners-Lee. Tento projekt byl vytvořen za účelem sdílení vědeckých prací a získaných informací mezi univerzitami a jinými vědeckými institucemi. [7] Web měl vědcům umožnit snadnější a rychlejší získávání informací potřebných k realizaci jejich projektů. Od tohoto momentu se nová technologie začala rapidně šířit i do dalších oblastí, a to vedlo k webu, který známe dnes.

## World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#).

[What's out there?](#)

Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#), [X11 Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#))

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help?](#)

If you would like to support the web..

[Getting code](#)

Getting the code by [anonymous FTP](#), etc.

Obrázek 1 První webová stránka (autor Tim Berners-Lee) [8]

## 1.2 Vývoj HTML

Prvotní myšlenkou Tima Berners-Leeho bylo, že autoři webových stránek nebudou muset znát značkovací jazyk, a proto ani specifikace tohoto jazyka nebyly zveřejněny [7]. Jeho plánem bylo implementovat do prohlížečů integrované editory pro webové stránky, avšak dosáhnout toho nebylo tak jednoduché, jak předpokládal, a proto došlo ke zveřejnění softwaru firmou CERN (evropská organizace jaderného výzkumu) i se specifikacemi jazyka. Na vývoji HTML se v té době nepodílelo velké množství vývojářů, což vedlo k zamrznutí vývoje HTML [3]. Čím více uživatelů začalo jevit zájem o webové stránky, tím více byl dosavadní stav značkovacího jazyka nedostačující. Tato skutečnost vedla k vydání HTML 2.0, na kterém pracovali převážně tehdejší vývojáři primitivních prohlížečů. Tato verze byla doplněna o nové prvky (např. formuláře) [3][7]. Verzi 3.0 představil Dave Raggett, nicméně nová vylepšení byla příliš náročná na tehdejší prohlížeče a z toho důvodu ji nebylo možné používat – proto vznikl standard 3.2, který jen minimálně rozšiřoval verzi 2.0 [7][9]. Zásadním pokrokem ve vývoji jazyka HTML byla verze 4.01, která přinesla oddělení kódování samotného vzhledu a webu. Byla zavedena podpora kaskádových stylů, na kterých kódování designu stojí. Rovněž byly implementovány složitější tabulky a nové tagy, které se využívají dodnes a jsou obsahem každého webu [10]. Vydáním této verze v roce 1997 se vývoj HTML na velmi dlouhou dobu zastavil a její vylepšení přišlo až v roce 2012 s HTML5, který je dnešním standardem podporovanými všemi prohlížeči. Tato verze přišla s mnoha novými vylepšeními a elementy, typy formulářů, a taky se zcela novou strukturou, díky čemuž je tělo webu přehlednější. Pro prohlížeče podporující standard HTML5 je jednodušší a více pochopitelné, co se v té části bude zobrazovat a co je obsahem daného

elementu, díky čemuž je načítání stránky výrazně rychlejší. Správné využití vlastností jednotlivých tagů se nazývá sémantické značkování.

### 1.3 Použití

Při tvorbě webu se kódér používání tagů nevyhne, a to ať už s využitím klasického HTML nebo různých frameworků, které mají definované vlastní elementy. V současnosti je při kódování možné ve standardu HTML5 použít přes 100 různých tagů (značek). Každý tag má svoji specifikaci a využití, ale ne všechny tagy se používají denně, a v praxi se reálně použije značně menší číslo, než je jejich celkový počet. Je možné si vystačit s těmi základními, bez kterých se správně funkční webová stránka neobejde.

V rámci vytvoření nového dokumentu mezi nejpoužívanější tagy patří ty, které tvoří strukturu HTML souboru, a bez nichž by webová stránka nebyla správně zobrazena nebo by nebyla schopná správné funkcionality.

```
<!DOCTYPE html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title></title>
  <meta name="description" content="">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="">
</head>
<body>
  <script src="" async defer></script>
</body>
</html>
```

Obrázek 2 Struktura HTML dokumentu vygenerovaná ve Visual Studio Code pomocí rozšíření html-5-boilerplate

Mezi další nejpoužívanější značky mimo ty tvořící strukturu HTML patří značky pro práci s textem, kam řadíme nadpisy, jejichž typy jsou rozděleny do šesti úrovní od **<h1>** až po **<h6>**, nebo samotné odstavce definované pomocí párového tagu **<p>Hello World</p>**.

## 2 CSS

CSS neboli Cascading Style Sheets (kaskádové styly) jsou určeny ke stylování HTML elementů. Pomocí kaskádových stylů mohou kodéři určovat, jak bude vypadat layout stránky nebo definovat barvy, tvar, vzhled a pozici jednotlivých HTML elementů. Cílem vytvoření CSS bylo oddělit grafickou část od struktury webu. Tvůrcem těchto stylů je mezinárodní konsorcium W3C (World Wide Consortium), tato organizace dnes určuje standardy HTML i CSS, které by měly být podporovány v prohlížečích [11].

### 2.1 Historie CSS

Vznik kaskádových stylů přišel až dlouho po vzniku HTML. Tento fakt může přijít spoustě lidí až neskutečný vzhledem k tomu, že dnes už bez dvojice HTML a CSS není možné si představit vytváření webové stránky. Tehdejší vývojáři se více zaměřovali na funkcionalitu HTML a potřebu stylovat samotné rozvržení webu nebo grafické úpravy jednotlivých elementů brali jako méně důležitý problém. S tímto ovšem nesouhlasili tehdejší kodéři webů a vývojáři webových prohlížečů.

S prvním vlastním souborem kaskádových stylů přišel Pei-Yuan Wei, vývojář a zakladatel webového prohlížeče ViolaWWW Browser [12]. S pomocí jeho stylovacího jazyku bylo možné ovlivnit barvu písma, pozadí, font a zarovnání textů. Jediným omezením byla nutnost používat Viola prohlížeče, protože to byl jediný prohlížeč, který tento jazyk podporoval. Pei-Yuan Wei měl ambice rozšiřovat podporu pro soubory kaskádových stylů i do dalších prohlížečů, avšak k této skutečnosti nikdy nedošlo [12]. Nicméně funkcionalita jeho jazyka byla základem pro další alternativy stylovacích jazyků, a také pro dnešní CSS.

Tabulka 1 Porovnání Viola-style stylopisu [65] s dnešním CSS stylopisem

Viola-style stylopis	CSS stylopis
<pre>(INPUT, P, H1     FGColor=black     BGColor=white     BDColor=black     align=left )</pre>	<pre>input, p, h1{     color:black;     background-color:white;     border-color:black;     text-align:left; }</pre>

Do vydání finální verze prvního CSS bylo pokusů o vytvoření správného stylovací jazyka mnoho, ovšem ani jeden z nich se dlouhodobě neuchytil. Mezi ně patřilo i rozšíření HTML jazyka vytvořením nových tagů, které ty původní stylovaly. S tímto přístupem přišel Marc Andreessen, vývojář prohlížečů Mosaic a Netscape [12].

Počáteční návrh CSS vymyslel Håkon Wium Lie z firmy CERN v roce 1994 [13]. Tato varianta se diametrálně lišila od kaskádových stylů používaných dnes, ale byla zajímavým návrhem, jakým směrem by se CSS mohlo vyvíjet. Toho se uchytil Bert Bos a je spoluvůrcem první veřejně vydané verze CSS1 v roce 1996, která uživatelům dovozovala základní operace se stylováním HTML elementů. První verze se převážně zaměřila na práci s textem, kdy největší změnou byly možnosti jako změna fontu, barvy a velikosti písma, ale i ostatní elementy získaly svá pravidla, mezi které patřily okraje, pozadí a pozicování jednotlivých elementů [14].

V roce 2000 došlo k vydání CSS2, nicméně s tím nastaly i problémy s kompatibilitou v prohlížečích, které tehdy nebyly ještě na takové úrovni, aby mohly nové změny přijmout a správně zobrazit [15]. Tímto problémem byli kodéři při vývoji webových stránek omezeni a museli být obezřetní při používání nového standardu CSS2, který dodal jak řadu rozšíření v práci s texty, tak novinky související s elementy jako pozice absolute, relative a fixed.

Dnes používané CSS3 obsahuje většinu vlastností z CSS2, přesto ale byly některé vlastnosti v revizi CSS2.1 odstraněny z důvodu špatné podpory a kompatibility v prohlížečích. CSS3 je obohaceno o animace pomocí 2D a 3D transformací s elementy, dále je možné používat pseudo-elementy a pseudo-tržidy, díky kterým lze například ovlivňovat interakce mezi myší a elementem, také je možné v této verzi přistupovat i k n-tému prvku a mnoho dalším vylepšením.

## 2.2 Použití CSS

Kaskádové styly se při kódování webu dají aplikovat na HTML strukturu třemi způsoby – interně, externě a v řádku jednotlivých html tagů pomocí „style“ atributu. Na následujícím řádkovém příkladu lze vidět, že odstavec s textem „Hello World“ bude mít červenou barvu, tučný text, velikost 25 px a zarovnání do středu rodičovského elementu.

```
<p style="color:red; font-weight:bold; font-size: 25px; text-align:center;">Hello World</p>
```

V případě využití externího .css souboru se soubor připojuje v html hlavičce. Za těchto okolností jsou všechny kaskádové styly v jednom nebo více kompletně oddělených



souborech, díky čemuž je zajištěna lepší přehlednost a také rychlejší načítání stránky, než kdyby všechny styly byly v hlavičce HTML souboru. Mezi velké výhody externího .css souboru patří znovu použitelnost a jednoduchost v globálních změnách.

```
<head>  
  <title>Bachelor thesis</title>  
  <link rel="stylesheet" href="assets/css/styles.css">  
</head>
```

### 3 CSS PREPROCESSORY

Jedná se o skriptovací jazyky, které eliminují mezery v kaskádových stylech. Pomocí těchto procesorů lze psát funkce a zakládat proměnné obsahující hodnoty s různými velikostmi textů, okrajů a dalších. Dále je možné do proměnných ukládat hodnoty barev a díky tomu si nadefinovat vlastní barevnou paletu. Při tomto přístupu není například nutné si pamatovat kód barvy využitý na stránce, ale je možné si ho vložit do proměnné a tuto proměnnou používat v celém souboru. Tyto jazyky nabízejí také možnost hníždění CSS pravidel, díky čemuž je kód o hodně přehlednější. Hníždění by vypadalo následovně [20].

```
.my-card{
  background-color:lightblue;
  p{
    text-align:center;
  }
  div{
    border:1px solid lightgrey;
  }
}
```

Značným rozdílem mezi preprocesory a klasickými kaskádovými styly je nutnost kompilace, aby byl prohlížeč schopen přečíst obsah souboru. Funkcionalita preprocesorů spočívá v kompilování skriptovacího jazyka do klasických kaskádových stylů [20][21]. Mezi nejvíce využívané preprocesory se řadí SASS a LESS. Stojí za zmínku, že kromě preprocesorů existují i postprocesory, které místo kompilování jiného jazyka pracují s klasickými kaskádovými styly a přidávají jim nějakou funkcionalitu navíc. Dá se říct, že se jedná o CSS na steroidech. Nicméně v porovnání s preprocesory jsou tyto procesory značně omezené.

Hlavním rozdílem mezi preprocesory SASS a LESS jsou programovací jazyky, na kterých jsou založeny. Zatímco základy preprocesoru SASS stojí na jazyku Ruby, LESS je postavený na JavaScriptu. Dalším odlišujícím faktorem, který může být rozhodující pro kodéra, je více možností, které SASS nabízí. Pomocí tohoto skriptovacího jazyka je možné psát cykly a podmínky, což je u druhé varianty LESS poněkud složitější a cyklu lze docílit například pomocí rekurze [59]. LESS je, co se týče syntaxe, bližší klasickým kaskádovým stylům, a proto může být pro začátečníka jednodušší se ho naučit. SASS je preprocessor, který nabízí dvě verze syntaxe – SASS a SCSS (Sassy CSS). Zatímco SASS je odlišná, trochu složitější a nepotřebuje složené závorky, tak SCSS je již jednodušší varianta, která je stejně jako LESS podobná CSS.

## 4 CSS FRAMEWORKY

Aplikační rámce s názvem CSS frameworky jsou prospěšným nástrojem nejen pro profesionály, ale i pro začátečníky. Kodérovi jsou k dispozici předdefinované styly, které může aplikovat na své elementy, aniž by musel zasahovat do stylopisů daného frameworku.

Tímto zároveň i odpadá nutnost při každém založení projektu začínat s prázdným souborem a je možné si vytvořit univerzální šablonu, jejíž znovu použitelnost vyústí v mnoho výhod, mezi něž patří úspora času a možnost zaměřit se více na jinou klíčovou problematiku daného projektu, než je stylování elementů. Některé CSS frameworky řeší i problematiku responzivnosti a správného zobrazení elementů na všech zařízeních.

Díky těmto mnoha výhodám, které frameworky přináší, je pro začátečníky příjemnější se se stylováním a kódováním webových stránek více sžít za předpokladu, že jim právě CSS dělá v procesu učení potíže nebo se chtějí při vývoji webových stránek zaměřit spíše na back-end místo front-endu.

### 4.1 Co je to framework

Framework je možné definovat jako sadu nástrojů, modulů a také knihoven, které mohou být uvnitř implementovány. Práce s využitím frameworku se liší tím, že tyto sady slouží k tomu, aby usnadnily celkový vývoj finální aplikace nebo jiného projektu. Vývoj těchto aplikačních rámců šetří čas vývojářům a kodérům jak při samotném zakládání nového projektu, tak i v průběhu vývoje jejich projektů, a to díky tomu, že opakovaně využívají sady nástrojů neboli kódů, které už někdo naprogramoval, a není nutné celým tím procesem procházet znovu. Každý framework má jiné vlastnosti a využití a pro správné nasazení a následující užití je nutné si nastudovat jejich dokumentace zvlášť.

### 4.2 Druhy a dělení frameworků

V oblasti vývoje webových stránek se rozlišují dva typy frameworků, které lze použít ve dvou různých vývojově oddělených částech, i přes jejich odlišnost spolu úzce souvisejících.

První ze dvou výše zmíněných částí je front-end zaměřující se na stranu klienta a zobrazení obsahu na jeho straně včetně celého vzhledu webové stránky [16]. Front-endové frameworky se zaměřují na manipulaci s nástroji HTML nebo CSS. Mezi front-end frameworky patří ale rovněž i ty, které nepracují s nástroji, nýbrž programovacími jazyky jako jsou například JavaScript a Type Script.

Back-end frameworky se soustředí na funkcionalitu webové stránky z pozadí neviditelného pro uživatele. Může se jednat o propojení se serverem nebo získávání potřebných dat z databáze. Back-endový vývoj je zároveň nutný při tvorbě klient-server webové aplikace. Back-end vývojáři mimo programovací část celého projektu připraví front-end kódérům data, které je třeba na stránce využít. Front-end kódéři musí tato data správně zobrazit na stránce podle návrhu nebo do míst jim určených.

### 4.3 Příklady CSS frameworků

Existence a použitelnost CSS frameworků se od dob prvních knihoven hodně rozrostla a kódéři již dnes mají na výběr z více frameworků, které mohou použít. Dříve se kódéři převážně spoléhali na Bootstrap, který patřil mezi první nejnámější a nejpoužívanější, dnes mu ale již konkurují i další známé a kvalitní frameworky jako například Foundation, Pure, Bulma, Semantic, Materialize a mnoho jiných. Nelze přímo uvést stupnici a posloupnost těch nejlepších, jelikož je každý z nich něčím výjimečný a není jednoznačný test, který by to dokázal ohodnotit. Existuje spousta článků, které se zabývají hodnocením, ale výsledek se neshoduje s dalšími zdroji, a proto se nedá na výstupy těchto článků nebo testů plně spolehnout. K popsání knihoven v této práci jsem se rozhodla vybrat tři, které pro letošní rok vyšly dle hodnocení na stránce LambdaTest nejlépe. Patří mezi ně nejpoužívanější Bootstrap, dále Foundation a Materialize [17].

#### 4.3.1 Foundation

Základy nového CSS frameworku s názvem Foundation vznikly v roce 2008, ale veřejnosti byly zpřístupněny jako open-source projekt, tedy volně k užití, až v roce 2011 [18]. Vývojáři tohoto dnes velmi úspěšného projektu byli zaměstnanci a vlastníci firmy ZURB. Firma na trhu působí již od roku 1998 [19]. Foundation je postaveno na základech ZURB Style Guide, což byl návod ke správnému stylování stránek a používání jednotlivých komponent [18]. Cílem projektu bylo vytvořit framework, který by z jejich dosavadních nejvíce využívaných částí kódů udělal celek, aby ho bylo možné nasadit již od samotného založení projektu, a tím dosáhnout rychlejší tvorby stránek, prototypování a snížit čas strávený eliminováním chybovosti v responzivité nebo v kolizích stylů.

Od dob vydání první verze uplynulo spousta času, framework Foundation si tak nese i spousty prvenství v tom, s čím přišel v použitelnosti jako první před ostatní konkurencí. Mezi tato prvenství patří mobile-first responsivity, tedy že CSS styly se při tvorbě stránky

jako první přizpůsobují na ty nejmenší zařízení (chytřejší telefony) a postupně se adaptují přes tablety a menší notebooky až k těm větším obrazovkám. Jedno z dalších prvenství, které tomuto frameworku patří, je využití preprocesoru SASS ve verzi Foundation 3.0 [18].

Preprocesor SASS je více popsán v dřívější kapitole, ale jen pro uvedení do věci, jedná se o rozšířenou syntaxi CSS. Pomocí SASS je možné ohýbat CSS k obrazu svému, vnořovat jednotlivá pravidla, což zaručuje mnohem lepší přehlednost kódu, a taky je možné psát funkce, mixiny a podmínky na provedení daných pravidel [20]. Rozdíl mezi CSS a SASS je, že CSS soubor ke spuštění a vykreslení nepotřebuje nic jiného než funkční prohlížeč, ale preprocesorový soubor k tomu potřebuje kompilátor, díky němuž je soubor zpracován a překompilován do klasických kaskádových stylů, které už mohou být zobrazeny prohlížečem [21]. Na základě tohoto přechodu z CSS na SASS Foundation získalo stylování webů větší možnosti ve funkcionalitě, dynamičnosti potažmo komplexitě designu.

Do roku 2015 se na tomto frameworku vývojáři z firmy ZURB podíleli aktivně, ale postupem času od toho ustoupili a dnes je již vyvíjen skupinou dobrovolníků, kteří se označují jako Yetinauti [22]. Jejich název je odvozen od maskota Yetiho ve skafandru, který uživatele vítá po příchodu na stránky.

Foundation se dělí na tři základní kategorie – Foundation for Sites, Foundation for Emails, Foundation for Apps. Každá z nich je jedinečná a zaměřená právě na svoji problematiku.

#### **4.3.1.1 *Foundation for Sites***

Foundation v oblasti vývoje webových stránek nabízí více směrů, kterými se může člověk ubírat při tvorbě stránky. Lze si vybrat z velkého množství responzivních a hotových šablon, které si je možné následně upravit dle vlastních nebo klientových požadavků. Mimo tyto šablony knihovna nabízí tzv. Building Blocks, což lze neoficiálně přeložit jako „stavební bloky“ [23]. Tyto bloky obsahují již hotové řešení sestavené z vícero UI komponent a lze je do projektu vložit pomocí kopírování, stažení z oficiálních stránek nebo instalováním za využití Foundation CLI<sup>1</sup>. Za bloky se mohou považovat hotové formuláře, menu, listy, hlavičky, patičky a cokoliv jiného, co je k nalezení v dokumentaci a je složeno z vícero elementů nebo komponent.

Jednou ze zajímavostí v nabídce bloků můžou být „kits“, které nejsou složeny jen z pár elementů, ale rovnou z celých bloků neboli sekcí, které mohou být v projektu využity.

---

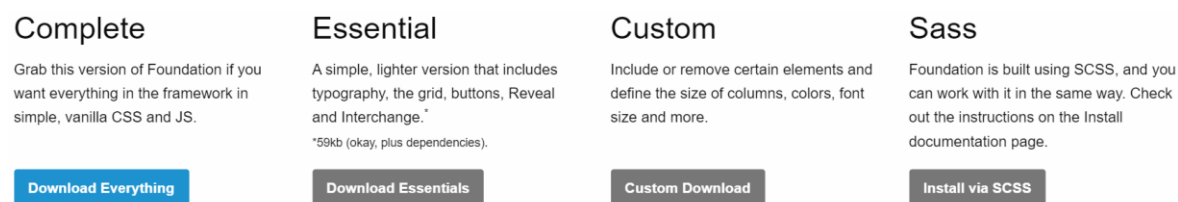
<sup>1</sup> CLI – Command Line Interface (příkazová řádka)

Například v kitu zaměřujícím se na portfolia je již obsáhlé menu, hlavička se sliderem, sekce s referencemi a mnoho dalších. Jednou z výhod Foundation Building Blocks a Building Kits může být, že není potřeba mít v projektu nainstalovaný celý framework, ale lze si vzít pouze ty části kódu, které jsou potřeba. To může vyústit v menší velikost a náročnost celého projektu, a zároveň je tím podpořeno jeho rychlejší načítání v prohlížeči.

#### 4.3.1.2 Instalace

Přidat frameworku do projektu lze více způsoby a je pouze na kodérovi, jaký si zvolí. Jednou z možností je stáhnout si požadovaný obsah ze zdrojových stránek frameworku (<https://download.get.foundation/sites/download/>) [24]. Velkou výhodou Foundation je volitelnost velikosti stahovaného balíčku, a pokud není potřeba veškerých funkcionalit, ale jen jejich části, tak lze získat plně funkční menší verzi podle požadavků. Výhoda menší velikosti se pozitivně odráží na rychlosti webových stránek a jejich načítání.

Sám Foundation framework nabízí v základu čtyři různé verze, jejichž obsah se liší v množství funkcionalit nebo komponent, které obsahují. Na obrázku číslo 4 je tento výběr vyobrazen.



Obrázek 3 Výběr balíčků frameworku Foundation ke stažení nebo instalaci [24]

Nejzajímavější možností z výše zmíněných balíčků je varianta Custom. V této variantě si sami volíme pomocí checkboxů, které komponenty mají být obsahem finálního staženého balíčku, a tudíž máme plnou kontrolu nad velikostí stahovaného souboru a zamítnutím nepotřebných komponent. Po vybrání veškerých chtěných komponent se přizpůsobuje jejich vzhled a výchozí hodnoty pomocí formuláře, díky kterému se při stahování tyto hodnoty nastaví do globálních proměnných a po stažení je možné pracovat s již upraveným frameworkem bez zásahu do globálních proměnných.

Choose Your Components:

All Foundation Components

Grid:

XY Grid

Float Grid

Flex Grid

No Grid

General:

Typography

Forms

Prototyping Helpers

Float Classes

Visibility Classes

Controls

Buttons

Close Button

Button Group

Slider

Switch

Set Your Defaults:

The Grid

# of Columns:

Total Gutter:

Max-Width:

Colors

Primary Color:

Secondary Color:

Alert Color:

Success Color:

Warning Color:

Body Font Color:

Header Font Color:

Obrázek 4 Ukázka formuláře před stažením balíčku Custom Foundation [24]

Na obrázku číslo 5 je možné vidět, že lze ovlivňovat vlastnosti jak samotného gridu včetně počtu sloupců a mezer mezi nimi, tak i barev jednotlivých komponent a textů.

Finální balíček obsahuje po stažení rozdělení jednotlivých souborů do složek, a zároveň i čistý index.html soubor, ve kterém lze začít nový projekt. V případě již existujícího projektu lze vložit stažené soubory do složky a v hlavním souboru index.html umístit do hlavičky všechny důležité odkazy na soubory, které se mají použít (kaskádové styly, JavaScript soubory aj.).

Jednou z dalších možností, jak získat do svého projektu Foundation framework, je instalace pomocí některého ze správce balíčků, mezi něž se řadí npm, yarn, gem, php composer, bower a další. V případě boweru a jeho kompatibility je lepší se zamyslet nad jinou variantou, jelikož tato technologie je již dnes považovaná za zastaralou. Samotné zdrojové stránky boweru nebo CLI upozorňují, že by bylo lepší svůj projekt spravovat pomocí yarn nebo npm správce balíčků. V případě již existujícího projektu, který byl spravován bowerem, je možné udělat migraci na jinou technologii pomocí návodu uvedeného na webových stránkách Boweru. Instalace pomocí správců se provádí kompletní a nainstalované balíčky již obsahují veškeré potřebné JS a SCSS soubory, ze kterých je možné čerpat.

Poslední možností k instalaci frameworku Foundation do projektu je využitím Foundation CLI<sup>1</sup>. CLI se do počítače nainstaluje globálně pomocí npm, aneb Node správce balíčků. V případě úspěšné instalace již stačí jen v příkazovém řádku zvolit cílovou složku projektu

<sup>1</sup> CLI – uživatelské rozhraní příkazového řádku

a založit nový projekt pomocí příkazu `foundation new`. Podobně jako u předchozí instalace, tak i v tomto případě se veškeré potřebné závislosti nainstalují již se založením projektu, a tudíž lze bez dalších komplikací začít s vývojem.

#### 4.3.1.3 Grid

Na rozdíl od jiných knihoven zaměřujících se na UI komponenty je Foundation jednou z mála, která nabízí více možností nastavení mřížky neboli více způsobů, jak rozvrhnout komponenty v rámci webové stránky s využitím vertikálních a horizontálních čar (grid). Volba gridu je čistě libovolná a záleží na kodérovi, který ze tří nabízených je pro jeho práci a zvyky vyhovující nejvíce. V nabídce jsou XY Grid, Float Grid a Flex Grid [25].

Prvním responzivním gridem, který tento framework nabízel, byl Float Grid [25][30]. Jeho responzivita spočívala v používání kombinace pravidel float, overflow, margin, padding a několika dalších. V porovnání s jeho nástupcem Flex Grid byl Float Grid o poznání těžší a zdouhavější k nakódování, ale až do nástupu Flexboxu neexistovalo jiné řešení, které by do té doby bylo dostupné. Vytvořením Float Gridu od Foundationu se kodérům usnadnila část práce, zároveň také se tím ušetřilo mnoho času tráveného zabýváním se responzivitou a správným nastavením float pravidel.

Stejně jako jiné společnosti, tak i Foundation vývojáři se vydali směrem dvanáctisloupcového gridu, jehož využití je snadné a účelové, a proto ho promítli do všech tří nabízených. Horizontálními a vertikálními třídami ovlivňujícími pozice a šířky elementů jsou „row“ a „columns“. Pro rozdílné rozvržení a roztažení přes sloupce na odlišných zařízeních se používají třídy s označením „small“, „medium“ a „large“.

Dalším gridem, na který se vývojáři zaměřili byl Flex Grid s využitím Flexboxu [26]. První návrh tohoto způsobu rozložení přišel v roce 2009, ale byl nestabilní a až do roku 2012 nebyl doporučovaný organizací W3C. Od roku 2012 byl už podporován většinou prohlížečů, jeho použití bylo rychlé, snadné a syntaxe byla jednoduchá [28]. V dnešní době je Flexbox nejpoužívanější variantou, jak dosáhnout responzivního designu. Toho si byli vědomi i v organizaci ZURB, a proto přišli s novým a spolehlivým Flex Gridem, který zahrnuli do jejich frameworku.

Nejnovějším a zatím posledním gridem, který Foundation nabízí, je již zmiňovaný XY Grid [27]. Tato mřížka se stala výchozí v rámci Foundation balíčku. Stejně jako u Flex Gridu, tak i zde se opírá základem o flexbox. Tyto gridy jsou od sebe odlišné v mnoha věcech. Pochopení a použití z dokumentace by mělo být mnohem rychlejší než u předešlých.



Zároveň je zde možnost využití pozicování elementů i v Y ose, což v předešlé variantě nešlo [29][30]. XY Grid cílí na nejjednodušší a nejrychlejší automatizaci velikosti ve vertikální rovině, čímž kodérovi odpadá nutnost psát vlastní stylovací pravidla pro tuto problematiku. Pomocí výchozí velikosti, která se nadefinuje rodičovskému elementu, se potomci roztáhnou do celé výšky a vyplní volný prostor. Stejně jako u horizontálních velikostí, tak i u vertikálních je možné obsah přizpůsobit pro zařízení různých velikostí pomocí tříd. Využití Y mřížky je vhodné v případě, že je známa vertikální velikost rodičovského elementu a jeho výška se poté rozděluje mezi jeho potomky. Za předpokladu, že máme <div> s velikostí 1000 px, jehož obsahem jsou dvě buňky (nový název třídy místo sloupce) a těmto dvěma buňkám přidělíme třídu auto, tak výsledná velikost obou elementů bude 500 px. Velikost se nemusí dělit pouze na půlku, ale je možné přidělit třídy jednotlivých zařízení s určujícím číslem finální velikosti tak, jak jsme zvyklí z X mřížky. Na následujícím příkladu kódu je toto dělení velikostí znázorněno.

```
<div class="grid-container">
  <div class="grid-y" style="height: 1000px;">
    <div class="cell small-4 medium-2 large-6">
    </div>
    <div class="cell small-8 medium-10 large-6">
    </div>
  </div>
</div>
```

Jednou z dalších příjemných výhod XY Gridu je volitelnost, zda buňky mají obsahovat vnější a vnitřní okraj, či nikoliv. Pokud jsou okraje chtěné, tak se pouze přidá ovlivňující třída, např. grid-padding-y.

#### 4.3.1.4 *Foundation for Emails*

S rapidně se měnící dobou a čím dál tím větší rozšířeností používání chytrých telefonů se firma zaměřila i na vývoj nového frameworku, který se výhradně soustředí na tvorbu moderních a responzivních e-mailů. Foundation for Emails, dříve zvaný Ink, byl vydán v roce 2015 [18]. Jeho účelem bylo urychlit tvorbu e-mailů a zajistit responzivitu kompatibilní na každém zařízení a nejlépe i v každé službě, a to ať už se jednalo o Outlook, Gmail nebo jakoukoliv jinou službu zaměřující se na elektronickou poštu.

Kodér, který se rozhodne tento e-mailový framework použít, si může sám zvolit, zda chce pracovat s CSS, nebo SASS, a podle toho zvolit typ balíčku, který si stáhne. Na základě jeho

zkušeností a času si dále může vybrat, zda bude elektronické zprávy tvořit pomocí již hotových responzivních komponent, nebo použije jednu z mnoha šablon, které jsou k dispozici.

HTML e-maily se posílají za účelem cíleně vizuálně zaujmout příjemce, a proto se používají pro reklamy, kampaně a elektronické letáky, které lákají čtenáře rozkliknout odkaz na webové stránky. Stejně jako u webů, tak i zde lze napsat svoje vlastní styly nebo přepsat ty předepsané základní a ovlivnit jejich design. Tímto lze docílit originality a je možné do šablon a komponent vdechnout identitu klienta nebo firmy, pro které se daný e-mail tvoří.

Při použití Foundation for Emails lze psát e-maily buď v klasickém HTML, nebo v šablonovacím jazyku s názvem Inky, který byl vytvořen právě pro tento framework. Zásadním rozdílem mezi e-mailem psaným v klasickém HTML a Inky je v jednoduchosti a rychlosti, které tento jazyk nabízí. Pro použití Inky je nutné pracovat s balíčkem obsahujícím preprocessor SASS [32].

V následujících ukázkách kódu přímo z dokumentace e-mailového frameworku lze vidět rozdíl mezi klasickým značkovacím a šablonovacím jazykem.

Dlouhý kód napsaný v HTML z dokumentace [33]:

```
<table align="center" class="container">
  <tbody>
    <tr>
      <td>
        <table class="row">
          <tbody>
            <tr>
              <th class="small-12 large-12 columns first last">
                <table>
                  <tr>
                    <th>Put content in me!</th>
                    <th class="expander"></th>
                  </tr>
                </table>
              </th>
            </tr>
          </tbody>
        </table>
      </td>
    </tr>
  </tbody>
</table>
```

```
</td>
</tr>
</tbody>
</table>
```

Krátký a rychlý kod psaný v Inky [33]:

```
<container>
  <row>
    <columns>Put content in me!</columns>
  </row>
</container>
```

Z ukázek výše lze vyvodit, že využití Foundation s preprocesorem je v tomto případě velmi výhodné pro každého kodéra. Nejen že je kód rychle napsaný, ale v případě větší složitosti designu je i o poznání přehlednější z hlediska zanořování.

### 4.3.2 Materialize

CSS Materialize patří v roce 2020 mezi nejvíce doporučované UI komponentové knihovny, které lze použít k webové tvorbě [17][34][35]. Vývojem této knihovny se zabývají vývojáři gigantické firmy Google ve spolupráci s týmem studentů z Carnegie Mellon University [36]. Materialize stejně jako Bootstrap, Foundation a mnoho jiných front-endových frameworků patří mezi open-source projekty, a tudíž je volně a zdarma k užití. Cílem vývoje bylo sjednotit uživatelskou zkušenost z procházení webových stránek na všech typech zařízení do jednotného konceptu, který by svým vzhledem zaujal každého návštěvníka stránek bez ohledu na zrovna používané zařízení. Tento framework může oslovit jak kodéra, tak i uživatele svou přehledností, jednoduchostí a efekty, mezi nimiž lze najít propracovaný a dnes tak používaný Parallax reagující na pozici a posun uživatele v rámci stránky. Knihovna zahrnuje mnoho předdefinovaných CSS, JS a HTML komponent, které lze jednoduše použít a modifikovat. Nabízí taktéž mnoho barevných schémat a hotových Starter templates neboli startovacích šablon k nasazení ihned. Výhodou těchto šablon je minimalizace času stráveného na počátečních kódech zaměřujících se na rozvržení a přizpůsobivost stránky, díky čemuž je možné se více orientovat na funkcionalitu.

Při prvotním použití Materialize je vhodné si tyto šablony stáhnout a z nich se pokusit pochopit, jakým způsobem se komponenty používají a jak z těchto šablon co nejvíce vytěžit.

Kodér se může rozhodnout pro dvě varianty – obyčejnou starter šablonu bez Parallaxu a s Parallaxem. Tato funkce obohacuje šablonu o efekt, při kterém překrytí vícero vrstev vytváří iluzi pohybu jednotlivých objektů. Každý z objektů se může pohybovat jinou rychlostí a jiným směrem, díky čemuž lze docílit příjemného interaktivního dojmu ze stránky.

K veškerému rozvržení komponent a šablon knihovna přistupuje v podobě systému flat design. Tento termín nemá odborné přeložení do českého jazyka. Flat design se zabývá responzivností a rychlostí načítání obsahu webové prezentace. Spoléhá se na minimalismus a jednoduchost návrhu, z čehož má uživatel čistý dojem a je pro něj snadné se orientovat v daném dokumentu. Využitím tohoto grafického návrhu dochází ke snižování náročnosti a plynulejšímu přizpůsobení na zařízení na základě velké, až úplné absence textur, stínů a jiných efektů, které tento proces zpomalují. Konkrétně tato část se ale nevztahuje na knihovnu od Googlu, který se rozhodl přizpůsobit flat design k obrazu svému a decentně využil i transformaci z-osy a rozsáhle do svých komponent zahrnul i stíny a jiné efekty, budící dojem, že je objekt v popředí nebo v dáli [37]. Název tohoto rozšířeného designu od Googlu je Material Design, který je využíván i v Android aplikacích [38]. Při volbě absence stínů a jiných efektů je dobré se více zaměřit na kontrastní barvy, barevné palety a jejich pozitivní vliv na zkušenost z návštěvy stránky.

Rozdíl mezi touto knihovnou a rozšířeným Bootstrapem je převážně ve svěžesti designu Materialize a nabídce komponent od Bootstrapu. Počet těchto nabízených modifikovatelných komponent vysoce převažuje knihovnu od Googlu, nicméně i v této oblasti má Materialize své vítězství, jelikož již zmiňované Parallax komponenty jsou plusem, které Bootstrap neobsahuje a nepracuje s nimi [37].

#### **4.3.2.1 Instalace**

Framework lze stáhnout ve dvou verzích, přičemž se jedna od té druhé liší přidanou hodnotou v podobě kompilovaných preprocesorových kaskádových stylů SASS. Při bližším zkoumání struktury souborů si lze všimnout, že v případě SASS verze jsou styly rozděleny do jednotlivých souborů po komponentách, zatímco v CSS verzi se všechny styly nachází v jednom datovém souboru, který lze použít jak ve zmenšené (minified) verzi, jejíž obsah je zredukovaný o redundantní pravidla, mezery a odřádkování, nebo také v plné verzi bez této redukce. To může být použito v procesu vývoje z důvodu lepší přehlednosti při hledání v kódu nebo ve snaze o jeho pochopení, ale všeobecně se nedoporučuje zasahovat do těchto pravidel a je lepší si založit vlastní soubor, který bude obsahovat pravidla, co přepíšou ta

původní, vzhledem k tomu, že se tyto řádky zpracovávají postupně a záleží na pořadí a specifitě pravidel. To stejné platí i pro .scss soubory.

Jak bylo zmíněno již výše, tak komponentovou knihovnu lze získat ve dvou verzích, a to vícero způsoby. Tím klasickým, po kterém nejspíš sáhne každý začátečník, je stáhnout si balíčky z jejich dokumentace, která je k nalezení na webových stránkách Materialize frameworku.

V této dokumentaci si lze přečíst i o dalších možnostech získání balíčků, mezi něž patří i CDN síť. Na tomto serveru je možné si vyhledat požadované odkazy obsahující souborovou strukturu nejen řešeného frameworku. Tyto odkazy se poté vloží do hlavičky index.html. Důvodem nahrání souborů na CDN síť, a ne na vlastní server je z hlediska bezpečnosti. CDN, neboli síť pro doručování dat, běží na několika serverech zároveň a v případě výpadku by nemělo dojít k pádu nalinkovaného obsahu do webové stránky. Další výhodou využití CDN je cachování do prohlížeče, což znamená, že když uživatel navštívil webovou stránku již dříve, tak prohlížeč nebude znovu stahovat soubory a načtení stránky je rychlejší.

Dalším a velmi používaným způsobem, jak získat framework do vlastního projektu, je použití správce balíčků – ten byl podrobně popsán již v kapitole o instalaci frameworku Foundation. V tomto případě jsou k dispozici dva správci, kteří jsou schopní nainstalovat knihovnu, a to je npm nebo bower.

#### 4.3.2.2 *Grid*

Při práci s různými frameworky může kodér narazit na odlišné způsoby rozvržení obsahu. Stejně jako u Bootstrapu, tak i zde je využitý dvanáctisloupcový grid neboli dvanáctisloupcové rozložení stránky. Pomocí těchto sloupců lze umisťovat elementy a komponenty. Kromě jejich umístění se zvoleným počtem sloupců určuje i jejich šířka nebo mezera mezi nimi. Tato mezera se nazývá offset [40].

Hlavním obalujícím elementem veškerého obsahu je <div> s třídou „container“. Tato třída má responzivně nastavenou šířku pro různé body zalomení za účelem lepšího zobrazování obsahu na odlišných zařízeních. Pro desktopové zobrazení je pravidlo šířky nastaveno na 70 % s vycentrováním divu na střed. To, jaká bude zvolená šířka, ovlivňuje velmi, jak je stránka vnímána a zda je pro uživatele přehledná. Obsah a veškeré texty by byly velmi nečitelné v případě, že by element byl roztažen přes celou šířku [40].

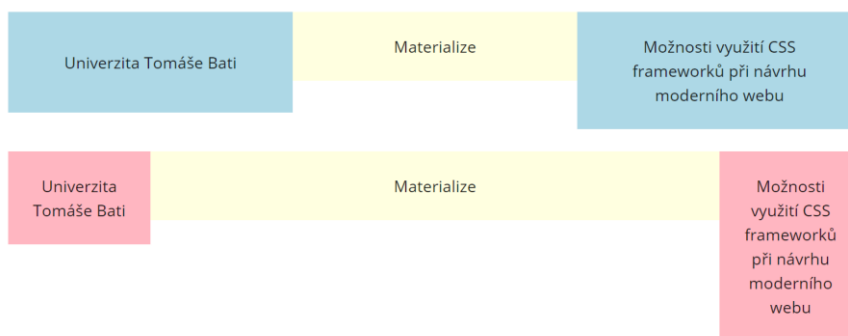
Následujícími důležitými třídami nastavujícími styly <div> tagů jsou „row“ a „column“, v překladu řada a sloupec. Row určuje pozici potomků horizontálně, kdežto sloupec vertikálně. Velikost jednotlivých sloupců se určuje pomocí přidáných tříd „s“ „m“ „l“. Každá ze zmíněných tříd se zaobírá jinou velikostí zařízení. Písmeno „s“ je pro mobile devices (mobilní zařízení/telefony) s velikostí <= 600 px, „m“ je pro tablety s šířkou <= 992 px a poslední „l“ patří desktopovým zařízením jako jsou počítače a notebooky s šířkou větší než 992 px [40].

Na následujícím obrázku lze vidět, jak je možné využít grid Materialize frameworku a jak jednoduché jeho použití je. Číslo za danou třídou vždy definuje počet sloupců, přes které se bude element zobrazovat na zařízení. V prvním sloupci jsou použity třídy col s12 m4 l4, z čehož vyplývá, že se jedná o sloupec, který bude na mobilu roztažen přes celou obrazovku a na tabletech a desktopech bude zabírat šířku čtyř sloupců.

```
<div class="container">
  <div class="row">
    <div class="col s12 m4 l4" style="background: lightblue;">
      | Univerzita Tomáše Bati
    </div>
    <div class="col s12 m4 l4" style="background: lightyellow;">
      | Materialize
    </div>
    <div class="col s12 m4 l4" style="background: lightblue;">
      | Možnosti využití CSS frameworků při návrhu moderního webu
    </div>
  </div>
  <div class="row">
    <div class="col s6 m4 l2" style="background: lightpink;">
      | Univerzita Tomáše Bati
    </div>
    <div class="col s6 m4 l8" style="background: lightyellow;">
      | Materialize
    </div>
    <div class="col s12 m4 l2" style="background: lightpink;">
      | Možnosti využití CSS frameworků při návrhu moderního webu
    </div>
  </div>
</div>
```

Obrázek 5 Ukázka kódu z Visual Studia – využití gridu Materialize

Grafické znázornění kódu v prohlížeči s využitím Materialize gridu je zobrazeno na následujícím obrázku, na kterém jsou sloupce barevně odlišeny.



Obrázek 6 Grafické zobrazení HTML kódu z obrázku č. 6

### 4.3.3 Bootstrap

V roce 2010 přišli dva vývojáři z organizace Twitter s novým nástrojem pojmenovaným Twitter Blueprint, který měl urychlit práci a zvýšit konzistenci kódu. Tento projekt byl původně zamýšlen k interním účelům, nicméně tito dva pánové, Mark Otto a Jacob Thornton, se později rozhodli svoje dílo vzít a udělat z něj plnohodnotný CSS framework za pomoci skupiny dalších vývojářů. Na velkém posunu ve vývoji se podílel i první Hack Week pořádaný organizací Twitter. Hackathonu se zúčastnilo mnoho talentovaných lidí, kteří se přispěním svých kódů do projektu podíleli na první oficiální verzi. První vydání open-source front-end frameworku proběhlo v roce 2011 a neslo název Bootstrap.

Do roku 2020 proběhlo již vydání vícero hlavních verzí a každá z nich znamenala velké změny v základu samotného frameworku.

Novým a velkým průlomem ve vydání druhé verze bylo kromě přidání nových komponent i přidání kaskádových stylů zaměřujících se pouze na responzivní rozlišení pomocí media queries. Jejich použití bylo volitelné a nacházely se v separátním souboru bootstrap-responsive.css, jejich kompilace probíhala díky oddělenému souboru zvlášť, a tudíž nebylo potřeba je mít zahrnuté v projektu. Dalším rozdílem, kterým byly kaskádové styly poznamenány, bylo odstranění podpory jiných gridů, jejichž počet sloupců přesahoval 12.

Ve třetím vydání se na rozdíl od druhého stala responzivita základem hlavního stylovacího souboru, a tím pádem již nebylo možné si nadále vybírat, zda chceme mít tuto část kódu v projektu obsaženou, či nikoliv. Obsah stránky stojí na principu mobile-first, z čehož plyne, že se stránka při kódování přizpůsobuje od nejmenších zařízení až k těm největším. Je snazší adaptovat obsah odspodu nežli svrchu při užití metody desktop-first, a to z toho důvodu, že desktopové rozložení nebývá vždy kompatibilní s uzpůsobením na menší zařízení. Využitím mobile-first přístupu, který byl adaptován i na všechny komponenty, bylo zajištěno, že

výsledná webová stránka byla vždy responzivní. V rámci módního trendu, kterému propadly i ostatní organizace zaměřující se na frameworky, se i Bootstrap vývojáři rozhodli přistoupit k flat designu, nicméně tato vlastnost ovlivňující komponenty nebyla povinná, a proto byla přidána do nových barevných témat.

V předposlední verzi, jejíž nástupce přišel až nedávno, došlo k rekonstrukci celého frameworkového balíčku za účelem využití nejnovějších, nejlepších a nejrychlejších technologií, které webový vývoj, jak ho dnes známe, může nabídnout. V rámci kaskádových stylů a preprocesorů došlo k migraci z původního LESS na SASS, což uvítalo nejspíš mnoho uživatelů Bootstrapu, vzhledem k tomu že SASS je používanější a oblíbenější než LESS na základě srovnání na Google Trends [66]. Další velkou a stěžejní změnou byl přechod na Flexbox z dosavadních responzivních metod využitých ve stylopisech. Kromě snadnějšího zarovnání a rozložení objektů se pomocí této změny rozšířily i možnosti v určování pořadí elementů a jejich přehazování na různých zařízeních, aniž by se muselo zasahovat do HTML kódu. Změna se dotkla i jednotek, kterými byly určovány velikosti textů a vnitřních/vnějších okrajů. V nové verzi se pracuje již s rem jednotkou, a nikoliv s pixely, což znamená, že cílené velikosti jsou relativní k velikosti písma na prvku `<html>`. Základní hodnota rem jednotky, pokud není nastaveno jinak, je 16 px.

S vydáním nejnovější verze s číslem 5 v letošním roce proběhly zásadní změny nikoliv v CSS, ale převážně ve využití programovacích jazyků JavaScript a jQuery. Po dlouhé době se Bootstrap vývojáři uchýlili ke kroku vypustit jQuery vzhledem k tomu, že dnes již jeho funkci dokáže nahradit čistý JavaScript se svými selektory. Zároveň tento krok mohl být způsoben i čím dál více rozšířenými javascriptovými frameworky, které se staly mezi uživateli oblíbené. Mezi tyto frameworky patří například React.js a Vue.js.

#### 4.3.3.1 Instalace

Podobně jako u předchozích frameworků, tak i v tomto případě si kodér může zvolit vícero variant, jakými dostat knihovnu do svého projektu. Jednou z těch, po které sáhne nejspíš každý začátečník, je stáhnout si balíček frameworku z oficiálních stránek, kde si přečte, jak s takovým balíčkem pracovat a jakým způsobem ho propojit do projektu. Další jednoduchou variantou je nalinkovat si závislosti frameworku do hlavičky projektu přes CDN. Obě tyto varianty obsahují minifikované CSS a JS soubory. Dalším oblíbeným způsobem získání různých balíčků, frameworků a nástrojů je ten s pomocí již zmiňovaného správce balíčků npm nebo yarn. Na rozdíl od předchozích frameworků je možné Bootstrap nainstalovat i do



projektů vyvíjených na platformě .NET. Toho se docílí pomocí správce balíčků s názvem NuGet. Tento správce je výhradně pro platformu .NET [49].

## **II. PRAKTICKÁ ČÁST**

## 5 ÚVOD DO PRAKTICKÉ ČÁSTI

V praktické části je popsáno srovnání jednotlivých frameworků. Na základě získaných dat tohoto srovnání jsou dle zvolených kritérií sestaveny tabulky, které poukazují na rozdíly mezi nimi. V druhé části je popsáno, jaký byl zvolen postup při tvorbě webové stránky od jejího samotného návrhu až po její realizaci a následné dokončení. K realizaci webové prezentace byl zvolen framework Bootstrap z důvodu jeho rozšířenosti, oblíbenosti a rozsáhlosti nabídky komponent. K problematice tohoto frameworku jsou dále popsány aktuální možné způsoby nakonfigurování vlastní šablony a jejího následného použití v praxi. Poslední částí této práce je tvorba kompletní webové prezentace, jejíž responzivita byla otestována pomocí patřičných nástrojů.

## 6 SROVNÁNÍ FRAMEWORKŮ

V této kapitole bude vypracováno srovnání tří výše zmiňovaných frameworků, které se zabývají CSS problematikou. Srovnání bude provedeno na základě uvedených dat, která byla vypsána k jednotlivým typům.

V první tabulce je srovnání obecných informací samotných frameworků. Tabulka obsahuje informace například o stáří, posledním stabilním vydání, a zároveň je v ní zaznamenaný menší přehled podporovaných technologií.

Tabulka 2 Obecné srovnání frameworků

	<b>Bootstrap</b>	<b>Foundation</b>	<b>Materialize</b>
<b>Stabilní verze</b>	4.5	6.6.3	1.0.0
<b>Licence</b>	MIT	MIT	MIT
<b>Datum 1. vydání</b>	8/2011	9/2011	11/2014
<b>Responzivní</b>	ANO	ANO	ANO
<b>JS</b>	ANO	ANO	ANO
<b>SASS</b>	ANO	ANO	ANO
<b>LESS</b>	ANO	NE	NE
<b>UI komponenty</b>	ANO	ANO	ANO

### 6.1 Grid

Grid by měl dnes být nedílnou součástí každého kvalitního CSS frameworku. Dle mého názoru je to jeden z důvodů, proč se kodér může rozhodnout pro nasazení frameworku do svého projektu, a proto je důležité, aby mu byl spíše přínosem, nikoliv na obtíž. Podrobné popsání mřížek a jejich vlastností proběhlo již v teoretické části. V následujících tabulkách budou zobrazeny vlastnosti a rozdíly mřížek jednotlivých frameworků.

V tabulce číslo 2 jsou vypsány typy mřížek, které frameworky v aktuálních verzích podporují.

Tabulka 3 Aktuální podpora různých typů gridu

	<b>Bootstrap</b>	<b>Foundation</b>	<b>Materialize</b>
<b>Flex Grid</b>	ANO	ANO	ANO
<b>XY Grid</b>	NE	ANO	NE
<b>Float Grid</b>	NE	NE	NE

V tabulce číslo 3 se nachází základní údaje, které mohou vývojáře zajímat. Mezi tyto údaje patří počet sloupců, s jakým typem mřížky pracuje, a taktéž počet tříd, které cílí na odlišné velikosti zařízení.

Tabulka 4 Vlastnosti mřížky

	<b>Bootstrap</b>	<b>Foundation</b>	<b>Materialize</b>
<b>Výchozí počet sloupců</b>	12	12	12
<b>Počet CSS tříd</b>	4	4	3

Následující tři tabulky jsou přiřazeny ke každému frameworku zvlášť. V tabulkách je zobrazena podpora různých zařízení s lišící se velikostí. Na tyto velikosti cílí CSS třídy obsahující pravidla ke stylování rozložení obsahu, a jak lze z tabulek vidět, tak každý framework pracuje s jinými body zalomení [45][46][40].

Tabulka 5 Podporované velikosti zařízení frameworkem Bootstrap

<b>CSS třída</b>	<b>Body zalomení</b>	<b>Zařízení</b>
.col-xs	< 768 px	Mobil
.col-sm	>= 768 px	Tablet
.col-md	>= 992 px	Malý notebook
.col-lg	>= 1200px	Počítač

Tabulka 6 Podporované velikosti zařízení frameworkem Materialize

<b>CSS třída</b>	<b>Body zalomení</b>	<b>Zařízení</b>
.s	< 600 px	Mobil
.m	>= 600 px	Tablet
.l	>= 992 px	Malý notebook
.xl	>= 1200px	Počítač

Z porovnání tabulek posuzujících frameworky Bootstrap a Materialize lze vyvodit, že jsou oba přizpůsobené pro čtyři různé velikosti zařízení a jejich body zalomení se z 50 % shodují.

Tabulka 7 Podporované velikosti zařízení frameworkem Foundation

<b>CSS třída</b>	<b>Body zalomení</b>	<b>Zařízení</b>
.small	< 640 px	Mobil
.medium	>= 640 px	Tablet
.large	>= 1024 px	Počítač

Z tabulky číslo 6 je možné vidět, že se vývojáři Foundation frameworku vydali jiným směrem než u předchozích dvou. Je uzpůsoben pouze na tři velikosti zařízení s úplně odlišnými body zalomení.

## 6.2 Dokumentace

Dokumentace je pro vývojáře velmi důležitou součástí každého nástroje nebo technologie, kterou použije nebo má v plánu se naučit a aplikovat ji do svého projektu. Již samotná kvalitně udělaná dokumentace může ovlivnit pohled člověka na daný nástroj a mít na něj pozitivní vliv. Pro každého vývojáře, který se chce naučit pracovat s něčím novým, je prvním zdrojem informací právě dokumentace. Je velmi pravděpodobné, že pokud je dokumentace nekvalitní nebo nese málo informací a není z ní úplně jasné, jak se má s daným frameworkem, nástrojem nebo technologií pracovat, tak vývojář půjde jiným směrem a dá šanci něčemu jinému. V této části srovnání popíšu, jaký dojem na mě udělaly jednotlivé dokumentace mnou zkoumaných frameworků.

Prvním frameworkem, jehož dokumentaci budu hodnotit, je Bootstrap. Tento zdroj informací na mě celkovým dojmem působil velmi propracovaně. Obsahuje jak velké množství obecných informací o samotném frameworku, tak i kvantum podrobných informací k samotným komponentám a jejich použití. Dokumentace sama o sobě vývojáře vede krok po kroku a snaží se ukázat co nejvíce z práce s frameworkem.

Další hodnocená dokumentace je od vývojářů frameworku Foundation. Zpočátku na mě působila podobným dojmem jako ta výše hodnoceného Bootstrapu, nicméně jak jsem se nořila dál do komponent, stavebních bloků, gridu a jiných částí, tak jsem z ní chvílemi nabývala dojmu, že je neúplná. U základních záležitostí je dokumentace podrobná, je jednoduché se v ní vyznat a přenést nově naučené do projektu, nicméně z dokumentace cítím, že jak šel framework dopředu, tak psaných informací ubylo. Vývojáři se vydali trochu jiným směrem a začali dělat část dokumentace formou vlogů. Tuto formu může spousta lidí uvítat, jelikož vidí na obrazovce v praxi, jak s tím vývojáři pracují, nicméně se můžou najít i lidé, kteří preferují konzervativnější cestu formou psaných informací.

Poslední dokumentace je od frameworku Materialize. Z prvotního zkoumání frameworku bych řekla, že je škoda, že v dokumentaci nebo na jejich webových stránkách nejsou k nalezení trochu podrobnější obecné informace a detaily z historie frameworku. Co se týče informací ke komponentám a vlastnostem frameworku, tak jsou dostačující, ale působí na mě občas trochu nepřehledně. V porovnání s prvními dvěma frameworky obsahuje mladší Materialize o dost méně nabízených komponent. Pravdou je, že je jednoduchý k použití, ale pokud mám hodnotit dokumentaci, tak ve srovnání s Bootstrapem je poměrně chudá.

### 6.3 Podpora

K porovnání aktuální podpory a posouzení aktivního vývoje jsem si vybrala data z GitHubu a správce balíčků npm. Na základě nich lze posoudit, že v případě Bootstrapu a Foundationu se jedná o frameworky, které jsou aktivně vyvíjené a aktuální. Zatímco u Materialize vyšla poslední stabilní před dvěma roky, z čehož lze odvodit, že vývoj je momentálně buď pozastaven, anebo značně zpomalen. Dále z čísel, které vyjadřují průměrný počet stažení vyplývá a potvrzuje se, že Bootstrap má opravdu velkou oblíbenost ze strany kodérů. Tato data jen dodatečně podporují statistiky ze Stack Overflow a Google Trends.

Tabulka 8 Aktuálnost frameworků dle dat ze služby GitHub [53][54][55]

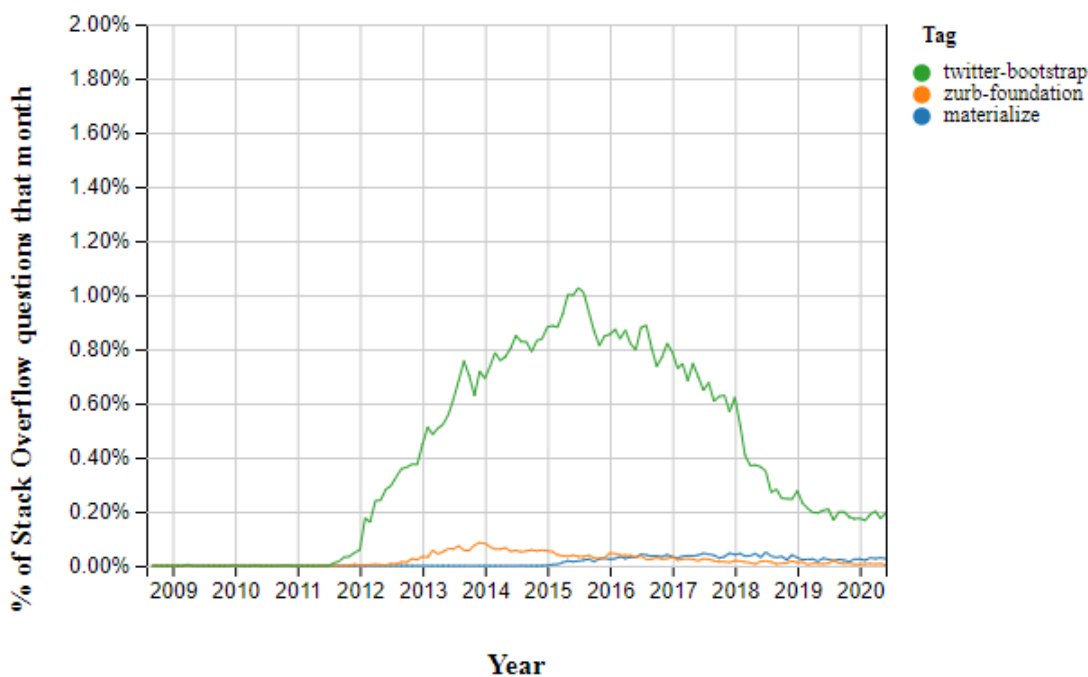
	<b>Bootstrap</b>	<b>Foundation</b>	<b>Materialize</b>
<b>Aktuální verze</b>	4.5	6.6.3	1.0.0
<b>Poslední aktualizace</b>	12. 5. 2020	9. 4. 2020	10. 9. 2018
<b>Počet vydaných verzí</b>	59	145	44

Tabulka 9 Průměrný počet stažení za týden z dat správce balíčků npm [50][51][52]

	<b>Bootstrap</b>	<b>Foundation</b>	<b>Materialize</b>
<b>Průměrný počet</b>	3, 566, 766×	150 969×	49 715×

## 6.4 Oblíbenost

V poslední fázi srovnávání jsem se zaměřila na oblíbenost dle naměřených čísel vyhledávacích dotazů. Vybrala jsem si dva zdroje, jimiž jsou Stack Overflow a Google Trends.[47][48]



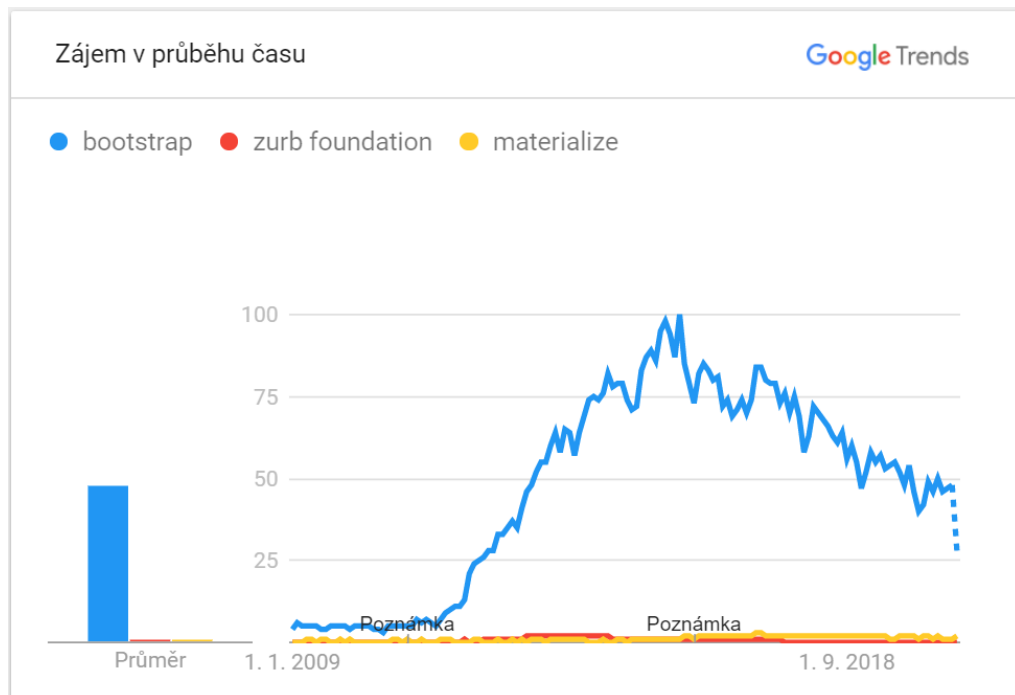
Obrázek 7 Srovnání počtu vyhledávaných dotazů na Stack Overflow[47]

Z obrázku, na kterém je graf s daty ze Stack Overflow, je více než patrné prvenství Bootstrapu. Za ním v tomto roce se umístil Materialize, který si vyměnil třetí místo



s Foundation v roce 2016. Materialize a Foundation si v porovnání s Bootstrapem, který je vysoko nad nimi, vedou velmi podobně.

Z obou obrázků ukazujících statistiky vyhledávaných dotazů lze určit, že dotazy na Bootstrap a problematiku s ním spojenou patří mezi nejvyhledávanější ze všech tří srovnávaných frameworků. Z toho lze usoudit, že patří taktéž mezi nepoužívanější.



Obrázek 8 Srovnání počtu vyhledávaných dotazů pomocí Google Trends [48]

## 6.5 Shrnutí

Mezi důležité faktory při výběrání vhodného frameworku patří podpora dnešních prohlížečů, kvalitně udělaná dokumentace a aktivní vývoj z hlediska budoucího posunu. Na tyto faktory by se měl zaměřit každý kodér, který se rozhoduje pro nasazení frameworku do svého projektu. Dále je třeba se soustředit na to, co samotný framework nabízí například z hlediska komponent. Zda jejich nabídka je dostačující a je možné je modifikovat.

Nasazení CSS frameworku nese své výhody, ale i nevýhody. Mezi výhody se dá zařadit, že jsou frameworky použitelné jak pro zkušené kodéry, tak pro amatéry. Dále je plusem ušetřený čas při vývoji webových stránek, díky čemuž se kodér nebo programátor může soustředit na složitější problematiku jak ve front-endu, tak back-endu, a může si pomocí frameworku v rychlosti nasadit základní responzivní layout stránky. V neposlední řadě odpadá starost ohledně implementace nových technologií, které se mohou ve verzích lišit.

Pokud se jedná o nainstalovaný balíček, nikoliv stažený, tak si tohle správce balíčků pohlídá sám a kodér bude mít u sebe vždy aktuální verzi s nejnovějšími technologiemi.

Do nevýhod využití CSS frameworků může být zařazeno velké množství kódu, který by projekt obsahoval navíc. Nicméně tento problém se dá vyřešit pomocí preprocesorů, kdy odpadá nutnost mít naimportovaný celý balíček a všechny jeho závislosti, ale stačí si naimportovat pouze potřebné komponenty. Dále můžou mít někteří kodéři problém se sžit s frameworkem, a to z toho důvodu, že si musí zvyknout na předem definované vlastnosti, které ne vždy lze obejít, a tudíž se v některých případech musí frameworku podřídít.

Dle mého názoru nese nasazení frameworku více výhod než nevýhod, a proto tato volba může být člověku hlavně přínosem, pokud si navykne na vlastnosti používání.

## 7 NÁVRH

Kódování webové stránky se nejlépe praktikuje podle grafického návrhu, na kterém lze vidět předem rozvržené elementy, komponenty a jednotlivé podstránky. Z předešlých zkušeností již vím, že kódovat a vymýšlet layout stránky za pochodu bývá často kontraproduktivní a při práci zdržující, protože se později často musí předělávat velké části za účelem konzistence celého webu. Na základě těchto zkušeností jsem se před samotným kódováním rozhodla si sama částečně stránku navrhnout s využitím online nástroje s názvem Figma, který se používá k navrhování wireframů a prototypování webových stránek a aplikací. Pro lepší výsledek jsem se rozhodla udělat návrh jak pro mobilní zařízení, tak i desktopové zobrazení. Na následujícím obrázku je možné vidět příklady jednotlivých návrhů z dvou podstránek pro desktopovou variantu, podle kterých budu finální projekt kódovat.



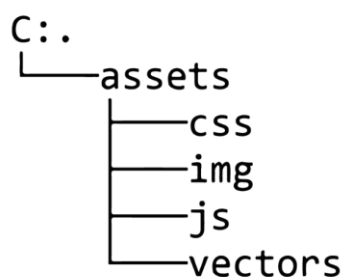
Obrázek 9 Návrh budoucí stránky

## 8 NASAZENÍ FRAMEWORKU DO PROJEKTU A REALIZACE

V této kapitole bude popsáno, jakým způsobem proběhlo založení projektu a následné nasazení zvoleného Bootstrap frameworku. Dále zde budou popsány příklady komponent, které byly využity při realizaci webové prezentace.

### 8.1 Založení projektu

Prvním krokem při započetí práce bylo založení projektu a vytvoření struktury potřebných složek a souborů. Strukturu byla vytvořena ve VS Code<sup>1</sup>, kde byl založen nový projekt a následně vytvořeny všechny podsložky, o kterých bylo možné do té doby říct, že jsou potřebné. Počáteční struktura vypadala následovně:



Obrázek 10 Počáteční struktura projektu

Dalším ne příliš nutným, ale velmi praktickým krokem bylo zvolit vhodnou organizaci k uložení projektu s verzovacím systémem. Výhoda využití verzovacího systému je úplná historie změn v projektu. Tímto se eliminuje riziko ztráty projektu, neuložených změn nebo špatného postupu díky možnému zpětnému návratu k poslednímu nebo jakémukoliv žádanému commitu (přidavku kódu a dalších změn).

V této fázi bylo již možné začít pracovat na realizaci webové stránky a nasadit zvolený framework. Na základní strukturu všech HTML souborů byl použit Bootstrap Starter Template. Do hlavičky souboru jsem si pomocí CDN naimportovala nutné soubory pro správnou funkčnost Bootstrap framework. Linky potřebné k vložení do hlavičky jsou převzaty z oficiální Bootstrap dokumentace. Pro tento způsob importu jsem se rozhodla z důvodu nižší velikosti finálního projektu, a taktéž pro nepotřebnost opětovného stahování obsahu balíčku v případě opakované návštěvy webové stránky. Dalším důležitým importem v hlavičce je stylovací soubor, na základě kterého se bude odvíjet většina vzhledu stránky.

---

<sup>1</sup> VS Code – Visual Studio Code

```
11 <!-- Bootstrap CSS -->
12 <link
13   rel="stylesheet"
14   href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
15   integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
16   crossorigin="anonymous"
17 />
18 <link
19   href="//netdna.bootstrapcdn.com/font-awesome/3.2.1/css/font-awesome.css"
20   rel="stylesheet"
21 />
22 <link rel="stylesheet" href="assets/css/style.css" />
```

Obrázek 11 Snímek z obrazovky – naimportované soubory v hlavičce

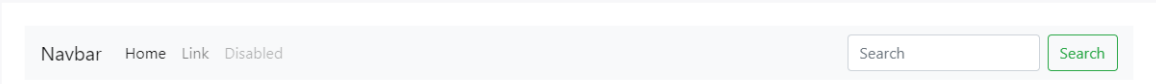
Po nastavení všech potřebných záležitostí jsem začala podle návrhu vytvořeného ve Figmě vyhotovovat stránku pomocí Bootstrap gridu a jeho komponent. Mezi prvními využitými komponentami, které jsem se následně snažila vzhledově přizpůsobit mému návrhu, byly navigační menu a slider s měnícími se fotkami.

## 8.2 Starter template

Prvním využitým blokem kódu z frameworku byla startovací šablona obsahující všechny závislosti k potřebnému fungování frameworku. Těmito soubory jsou výchozí kaskádové styly, skripty a knihovny Popper.js a jQuery.js. [64]

## 8.3 Navigační menu

Nabídka komponent s názvem navbar neboli navigační menu tohoto frameworku je poměrně rozsáhlá a podrobně zdokumentovaná. Je možné si vybrat jak mezi různými typy, tak i odlišnými barevnými tématy (tmavé/světlé/barevné). Z nabídky si kodér může vybrat, zda si přeje obyčejný jednoduchý navbar s odkazy nebo i s vyhledávacím polem a rozbalovacími menu, která mohou obsahovat další podmenu. Vše od Bootstrapu je plně responzivní, a to platí i pro navigační menu. Na menších zařízeních jsou odkazy v liště sbalené a skryté. Zobrazení je možné pomocí přítomného hamburger tlačítka (menu tlačítka).



```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarTogglerDemo03" aria-controls="navbarTogglerDemo03" aria-expa
  <span class="navbar-toggler-icon"></span>
</button>
<a class="navbar-brand" href="#">Navbar</a>

<div class="collapse navbar-collapse" id="navbarTogglerDemo03">
  <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
    <li class="nav-item active">
      <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link disabled" href="#">Disabled</a>
    </li>
  </ul>
  <form class="form-inline my-2 my-lg-0">
    <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-label="Search">
    <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
  </form>
</div>
</nav>
```

Obrázek 12 Ukázka původního kódu komponenty navbar [44]

Pro své potřeby jsem převzala menu z předchozího obrázku. Jeho obsahem jsou odkazy, místo pro logo (nebo název) a vyhledávací pole. Z kódu lze vyčíst, že komponenta využívá jako obalovací HTML element `<nav>`. Element prohlížeči říká, že v této části se bude nacházet navigace. Vnitřní část komponenty je tvořena pomocí seznamu `> li > a`, kde `<a>`, jsou odkazy do jednotlivých kategorií nebo podstránek. Cesta k těmto odkazům se vkládá pomocí atributu „href“.

Z návrhu je možné vyzorovat, že vložený navbar neodpovídá finálnímu vzhledu, který má být vyobrazen na webové stránce, a proto bylo potřeba ho upravit. Kód mého navbaru vypadá následovně.

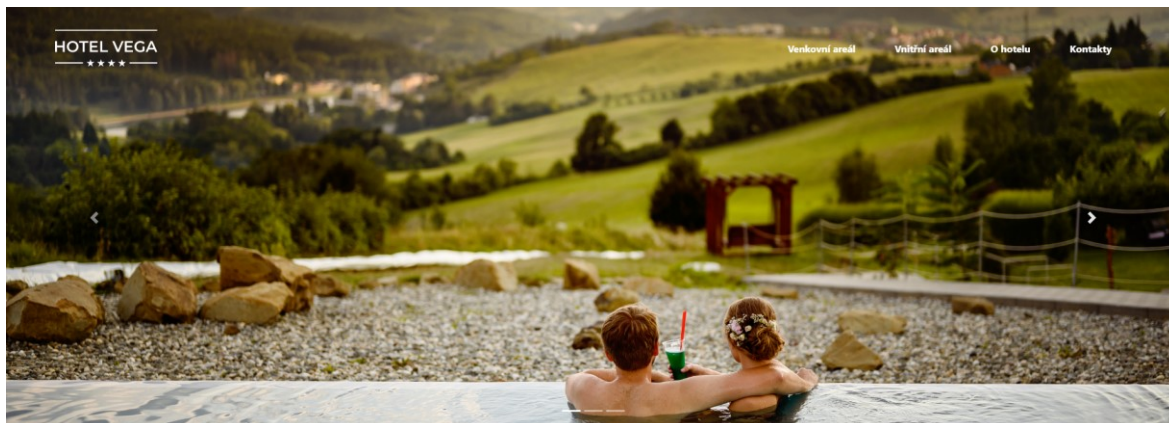
```
28 <nav class="navbar fixed-top navbar-expand-lg navbar-light" id="my-nav">
29   <a class="navbar-brand" href="#"
30     ></a>
32   <button
33     class="navbar-toggler"
34     type="button"
35     data-toggle="collapse"
36     data-target="#navbarText"
37     aria-controls="navbarText"
38     aria-expanded="false"
39     aria-label="Toggle navigation"
40   >
41     <span class="navbar-toggler-icon"></span>
42   </button>
43   <div class="collapse navbar-collapse" id="navbarText">
44     <ul class="navbar-nav ml-auto">
45       <li class="nav-item active">
46         <a class="nav-link" href="index.html"
47           >Venkovní areál <span class="sr-only">(current)</span></a>
48         </li>
49       <li class="nav-item">
50         <a class="nav-link" href="#">Vnitřní areál</a>
51       </li>
52       <li class="nav-item">
53         <a class="nav-link" href="about.html">O hotelu</a>
54       </li>
55       <li class="nav-item">
56         <a class="nav-link" href="#">Kontakty</a>
57       </li>
58     </ul>
59   </div>
60 </nav>
```

Obrázek 13 Modifikovaný kód Bootstrap navbaru

V levé části se nachází logo, které po interakci uživatele vrátí zpět do vrchní části webové prezentace. Z responzivních důvodů je logo vloženo ve formátu SVG. Logo mění velikost podle používané velikosti zařízení, a taky podle vertikální pozice uživatele na stránce.

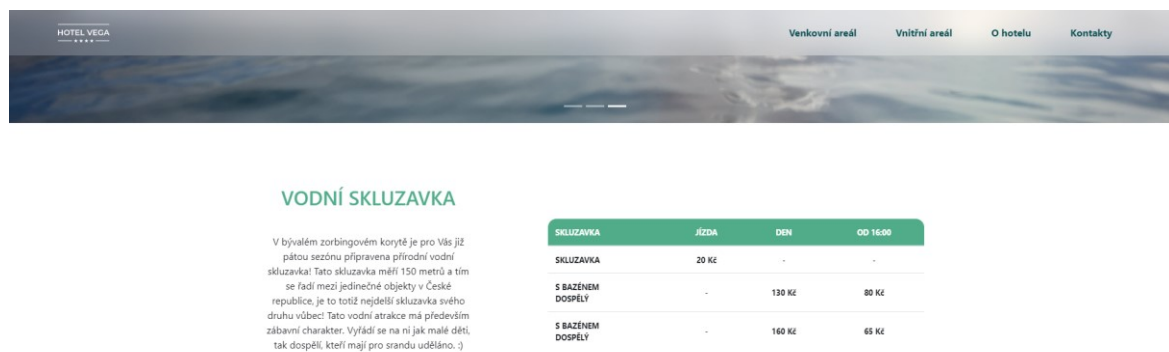
Část s vyhledávacím polem byla odstraněna, protože pro můj projekt byla nepotřebná. Dalším nepatrným rozdílem v kódu a výsledném vzhledu může být, že můj seznam odkazů je pozicován do pravé části navbaru. Tohoto výsledku bylo docíleno zaměněním přednastavené třídy `mr-auto` za `ml-auto` v `<ul>` seznamu. Třída `ml-auto` určuje, že vnější okraj z levé strany bude roztažený přes celou šířku, čímž se docílilo umístění odkazů na pravou stranu.





Obrázek 14 Vzhled navbaru po úpravách

Výchozí barva lišty je přepsána vlastními pravidly přiřazenými elementu nav. Důraz na tyto pravidla byl dosažen vlastností „!important“, která by měla zajistit jednoznačnou přednost před výchozími Bootstrap pravidly.



Obrázek 15 Vzhled navbaru po úpravách a scrollu

Jednou z posledních změn, které lišta obdržela, bylo ovlivnění velikosti a barvy založené na vertikálním scrollu (posunu) uživatele ve stránce. Pomocí JavaScriptu a vlastní CSS třídy jsem nastavila vlastnosti ovlivňující barvy lišty, pokud se uživatel posune vertikálně o 80px. Za tímto počínáním stála skutečnost, že původní transparentní barva občas způsobovala špatnou čitelnost loga a textu v průběhu scrollování v rámci stránky.

Při pohledu na obrázky s číslem 11, 13 a 14 lze bezpečně usoudit, že ačkoliv jsem použila responzivní Bootstrap komponentu s výchozím vzhledem, tak není pro kodéra vůbec problém si použité komponenty přizpůsobit vlastní představě, a to poměrně s rychlou aplikací. Právě díky této nekomplikované možnosti úpravy jde docílit odlišného vzhledu jakékoliv použité Bootstrap komponenty na stránce a tím získat zcela jedinečný design, který nemusí na první pohled vůbec působit jako od Bootstrapu.



## 8.4 Slider

Další použitou komponentou viditelnou i na obrázku č. 13 byl Carousel slider. Jedná se o bootstrapovou komponentu s 3D transformacemi, která ke svému správnému fungování potřebuje nejen Bootstrap kaskádové styly, ale taktéž JavaScript starající se o prolínání vícero fotek za sebou v rámci určitého času, jehož nastavitelnost trvání je ovlivnitelná pomocí poslání chtěné hodnoty do objektu. Výhodou Carousel slideru je opět jeho responzivita, ale taktéž spolehlivá a elegantní funkčnost dodávající uživateli lepší zkušenost z návštěvy stránek.

Slider se aktivuje zavoláním Carousel funkce na konci souboru [43] v ohraničující značce `<script>`.

```
<script>
    $(".carousel").carousel();
</script>
```

Z jednořádkového kódu vyplývá, že funkce bude zavolána na obalující element s třídou carousel. Výchozím intervalem prolínání je 5 sekund, nicméně tento čas může být změněn pomocí vložení jednoho řádku do scriptu navíc, nebo také posláním hodnoty do HTML elementu pomocí atributu. V následující ukázce je interval z výchozí hodnoty přenastavený na 3 sekundy.

```
<script>
    $(".carousel").carousel(
        interval:3000);
</script>
```

Mezi další možnosti modifikující funkcionalitu obrázkové prezentace patří keyboard, pause, ride a wrap. V základu má slider nastavené reakce na najetí myši. Tyto reakce mají za výsledek, že při najetí myši (hover) na obrázek se prezentace zastaví a zůstane stát na tom konkrétním objektu, na kterém se nachází. Prezentace se poté znovu spustí ve chvíli, kdy kurzor opustí oblast slideru. Toto chování může být upraveno možností pause, která přijímá dvě hodnoty - „hover“ a „false“. Keyboard a wrap možnosti jsou datového typu boolean s pouze dvěma hodnotami true nebo false. Nastavením keyboard na false se vyruší reakce slideru na klávesy (např. šipky).

V základu framework nabízí více variant sliderů. Ve výběru se nachází jednoduché slidery pouze s automatickou záměnou fotek, dále i popisky, indikátory a s šipkami po stranách, aby

se uživatel mohl dle libosti vrátit. Pro svůj projekt jsem zvolila variantu s indikátory a šipkami po stranách. Popisky tam nebyly potřeba.

## 8.5 Grid

Grid je část frameworku, která pro mě byla nejdůležitější a nejvíce používaná, jelikož jsem pomocí mřížky sestavila rozložení celé webové stránky podle návrhu bez sebemenších komplikací. Pracovat na rozložení pomocí Bootstrap mřížky bylo přínosné a po pochopení, jakým způsobem funguje, i svižné.

Rozložení funguje na principu responzivní dvanáctisloupcové mřížky, jejíž vnitřnosti se skládají z obalujícího elementu <div> s přiřazenou třídou container nebo container-fluid a dále hromadou elementů <div> s označením tříd row nebo column, které značí řádky a sloupce. Na příkladu snímku obrazovky na následujícím obrázku bude popsáno, jakým způsobem jsem se dopracovala k tomuto rozložení za využití mřížky.



Obrázek 16 Snímek z obrazovky k popsání funkčnosti gridu

Ze snímku lze vidět, že se v této sekci nachází tabulka s informacemi k cenám venkovní atrakce, a taky tabulka se stručným popisem lákající návštěvníka webu k návštěvě hotelu. Na první pohled se může zdát, že obalujícím elementem je div s třídou container, nicméně v pozadí se nachází relativní a absolutní elementy roztažené přes celou obrazovku, a proto byl použit container-fluid, do něhož byl následně vnořený container, jehož šířka na desktopu zabírá 70 % obrazovky.

Kód celé sekce je příliš dlouhý, jelikož obsahuje i jiné elementy, které souvisí s obsahem, nicméně pro ukázkou využití dvou containerů stačí následující kousek osekáný o nadbytečné části. Část s vnořeným containerem je tučně zvýrazněná.

```
<div class="container-fluid">
  <section class="pool-section" id="pool-section">
    <div class="pool-bg">
      <div class="wave wave-1 wave-mobile" id="wave-1-up">
        
      </div>
      <div class="wave wave-2 wave-mobile" id="wave-2-up">
        
      </div>
      <div class="wave-content">
        
      </div>
      <div class="wave-upper-content">
        <div class="container">
          <div class="row">
            <div class="col-12 col-md-12 col-lg-6">
            </div>
            <div class="col-12 col-md-12 offset-lg-1 col-lg" id="parent">
            </div>
          </div>
        </div>
      </div>
      <div class="wave wave-1 wave-mobile" id="wave-1-down">
        
      </div>
      <div class="wave wave-2 wave-mobile" id="wave-2-down">
        
      </div>
    </div>
  </section>
</div>
```

Z kódu lze vyčíst, že veškerý obsah v pozadí se nachází v container-fluid, zatímco informační část se nachází v obyčejném container, který je následně vyplněn řádkem se

sloupce, jejichž velikosti na různých zařízeních jsou dané pomocí zvolených tříd. Ze zvolených tříd lze vidět, že zatímco na nejmenších zařízeních jsou tyto sloupce roztažené přes celou obrazovku, tak v desktopové části zabírá levý objekt šířku o velikosti šesti sloupců z dvanáctisloupcového gridu, mezi nimi je mezera o velikosti jednoho sloupce a pravý objekt má plnou velikost ze zbylého prostoru.

Z této ukázky lze usoudit, že rozložení s využitím frameworku šetří spoustu řádků vlastních CSS pravidel, která by se mohla neustále opakovat, a zároveň poté i přizpůsobit na ostatní zařízení.

## 8.6 Tabulky

Jako poslední příklad použité komponenty v projektu jsem si vybrala tabulky. Stejně jako u ostatních komponent, tak i u této je nabízeno mnoho variant, z kterých je možné vybírat. V nabídce je více barevných témat, a taktéž je v dokumentaci podrobně vypsáno, jakou třídou lze ovlivnit chování tabulky. Tabulky můžou a nemusí být ohraničené, taktéž mohou mít ohraničení buněk, ale je možné si vybrat jen z horizontálního ohraničení (tedy ohraničení řádků). Tabulkám lze přiřadit responzivní vlastnost pomocí přiřazené třídy `table-responsive`, která u větší tabulky přesahující šířku obrazovku nastaví horizontální scrollbar, díky kterému nedochází k přetékání obsahu.

Pro mé účely byla nejvhodnější tabulka s označením `table-hover`, která měla světlé barevné téma, ohraničení řádek a interakci při najetí kurzorem do tabulky.

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

Obrázek 17 Výchozí tabulka s označením `table-hover`

Tabulku jsem si přizpůsobila pomocí vlastních CSS pravidel, která přepisovala ta výchozí, a tudíž bylo možné docílit téměř stejného vzhledu tabulky z původního návrhu.

SKLUZAVKA	JÍZDA	DEN	OD 16:00
SKLUZAVKA	20 Kč	-	-
S BAZÉNEM DOSPĚLÝ	-	130 Kč	80 Kč
S BAZÉNEM DOSPĚLÝ	-	160 Kč	65 Kč

Obrázek 18 Modifikovaná tabulka

Z následující ukázky kódu lze vidět, že stejně jako u gridu a layoutu dalších elementů, tak i u tabulek jsou využívány třídy `row` a `col` k určení šířky sloupců.

```
<thead>
  <tr>
    <th scope="col">SKLUZAVKA</th>
    <th scope="col">JÍZDA</th>
    <th scope="col">DEN</th>
    <th scope="col">OD 16:00</th>
  </tr>
</thead>
<tbody>
  <tr>
    <th scope="row">skluzavka</th>
    <td>20&nbsp;Kč</td>
    <td>-</td>
    <td>-</td>
  </tr>
</tbody>
```

## 8.7 Vlastní konfigurace frameworku Bootstrap

V kapitole zabývající se frameworkem Foundation byla zmíněná možnost vlastní konfigurace nástroje přímo na jejich webových stránkách. Bootstrap nabízel podobný nástroj v dřívější verzi s číslem 3, nicméně tato možnost byla od čtvrtého vydání odebrána. Pomocí tohoto nástroje si kodér mohl navolit, které komponenty bude stažený finální balíček obsahovat, díky čemuž se mohla výrazně snížit velikost souboru, pokud cílil jen na pár konkrétních částí frameworku. Dále bylo možné pracovat s LESS proměnnými a jejich hodnoty dle libosti měnit. Obsah nové proměnné nahrazoval výchozí barvy, okraje, odsazení, viditelnost a mnoho dalších. Při konfigurování vlastní šablony se kodér mohl

jednoduše zaměřit na to, aby barevné téma komponent odpovídalo jeho cílenému vzhledu. Stejným způsobem bylo možné modifikovat grid systém a jeho body zalomení. V tomto případě si kodér mohl navolit jiný počet sloupců, než bylo výchozích 12. V případě, že byl kodér z dřívějších zkušeností nebo jiných frameworků zvyklý na jiné hodnoty, tak si je mohl pro každé zařízení zvlášť změnit. [56]

## Media queries breakpoints

Define the breakpoints at which your layout will change, adapting to different screen sizes.

<p><b>@screen-xs</b></p> <input type="text" value="480px"/> <p>Deprecated <b>@screen-xs</b> as of v3.0.1</p>	<p><b>@screen-xs-min</b></p> <input type="text" value="@screen-xs"/> <p>Deprecated <b>@screen-xs-min</b> as of v3.2.0</p>	<p><b>@screen-phone</b></p> <input type="text" value="@screen-xs-min"/> <p>Deprecated <b>@screen-phone</b> as of v3.0.1</p>
<p><b>@screen-sm</b></p> <input type="text" value="768px"/> <p>Deprecated <b>@screen-sm</b> as of v3.0.1</p>	<p><b>@screen-sm-min</b></p> <input type="text" value="@screen-sm"/> <p>Deprecated <b>@screen-tablet</b> as of v3.0.1</p>	<p><b>@screen-tablet</b></p> <input type="text" value="@screen-sm-min"/> <p>Deprecated <b>@screen-tablet</b> as of v3.0.1</p>
<p><b>@screen-md</b></p> <input type="text" value="992px"/> <p>Deprecated <b>@screen-md</b> as of v3.0.1</p>	<p><b>@screen-md-min</b></p> <input type="text" value="@screen-md"/> <p>Deprecated <b>@screen-desktop</b> as of v3.0.1</p>	<p><b>@screen-desktop</b></p> <input type="text" value="@screen-md-min"/> <p>Deprecated <b>@screen-desktop</b> as of v3.0.1</p>
<p><b>@screen-lg</b></p> <input type="text" value="1200px"/> <p>Deprecated <b>@screen-lg</b> as of v3.0.1</p>	<p><b>@screen-lg-min</b></p> <input type="text" value="@screen-lg"/> <p>Deprecated <b>@screen-lg-desktop</b> as of v3.0.1</p>	<p><b>@screen-lg-desktop</b></p> <input type="text" value="@screen-lg-min"/> <p>Deprecated <b>@screen-lg-desktop</b> as of v3.0.1</p>
<p><b>@screen-xs-max</b></p> <input type="text" value="(@screen-sm-min - 1)"/>	<p><b>@screen-sm-max</b></p> <input type="text" value="(@screen-md-min - 1)"/>	<p><b>@screen-md-max</b></p> <input type="text" value="(@screen-lg-min - 1)"/>

## Grid system

Define your custom responsive grid.

<p><b>@grid-columns</b></p> <input type="text" value="12"/> <p>Number of columns in the grid.</p>	<p><b>@grid-gutter-width</b></p> <input type="text" value="30px"/> <p>Padding between columns. Gets divided in half for the left and right.</p>	<p><b>@grid-float-breakpoint</b></p> <input type="text" value="@screen-sm-min"/> <p>Point at which the navbar becomes uncollapsed.</p>
<p><b>@grid-float-breakpoint-max</b></p> <input type="text" value="(@grid-float-breakpoint - 1)"/> <p>Point at which the navbar begins collapsing.</p>		

Obrázek 19 Konfigurace mřížky a bodů zalomení ve frameworku Bootstrap 3 [56] Vlastní konfigurace se netýkala pouze možností ovlivňujících CSS pravidla, ale i obsahu JavaScript souboru s jQuery zásuvnými moduly. Tyto moduly se starají o správnou funkcionální efektů nebo funkcí aplikovaných na jednotlivé komponenty. Například Carousel modul se stará o prolínání fotografií v Carousel slideru. Bez tohoto modulu by se objekty uvnitř slideru nezaměňovaly a na stránce by byla vyobrazena jedna statická fotografie.

## jQuery plugins

[Toggle all](#)

Choose which jQuery plugins should be included in your custom JavaScript files. Unsure what to include? Read the [JavaScript](#) page in the docs.

### Linked to components

- Alert dismissal
- Advanced buttons
- Carousel functionality
- Dropdowns
- Modals
- Tooltips
- Popovers (requires Tooltips)
- Toggable tabs

### Magic

- Affix
- Collapse
- Scrollspy
- Transitions (required for any kind of animation)

### Obrázek 20 jQuery zásuvné moduly [56]

Po vybrání požadovaných komponent a nastavení jejich proměnných, které ovlivňovaly výsledný vzhled, bylo na konci konfiguratoru tlačítko, které zahájilo kompilování všech nastavení a následně proběhlo stažení všech upravených souborů. Výsledný balíček obsahoval potřebné CSS a JS soubory včetně konfiguračního souboru.

Jak bylo již zmiňováno výše, podpora tohoto nástroje od čtvrtého vydání byla odebrána. Na odebrání podpory měl značný vliv přechod z preprocesoru LESS na SASS. Aby nebyli uživatelé frameworku tímto krokem příliš omezeni, tak vývojáři přišli s jinou variantou, jak lze modifikovat výsledné soubory, které jsou použity v projektech. Sestavení vlastních témat je umožněno modifikováním SCSS souborů. Této modifikace lze docílit pomocí vlastního SCSS souboru, jež je oddělený od souborů frameworku Bootstrap, aby nedošlo ke kolizi pravidel nebo mylné úpravě výchozích souborů. Pomocí následného importu jednotlivých komponent nebo celého frameworku jsou požadované funkcionality obsaženy v novém SCSS souboru a je možné je modifikovat dle vlastních potřeb. Soubory kromě proměnných obsahují mapy, jež mohou být popsány jako CSS objekty, ve kterých jsou uloženy proměnné nebo barevné palety. Obsah těchto map lze modifikovat, přidávat další proměnné nebo také odebrat nepotřebné [61].

V případě, že by bylo nutné pracovat na úrovni zdrojových kódů frameworku Bootstrap a využít jeho sestavovací systém, je nutné nejprve nainstalovat Node.JS, Ruby, a Bundler, které se postarají o instalaci přidružených závislostí. Po instalaci je možné využít příkazy

jako „test“, nebo „run docs“. Bootstrap taktéž využívá nástroje Autoprefixer, který automaticky přidává vendorové prefixy, čímž šetří čas a množství psaného kódu [62].

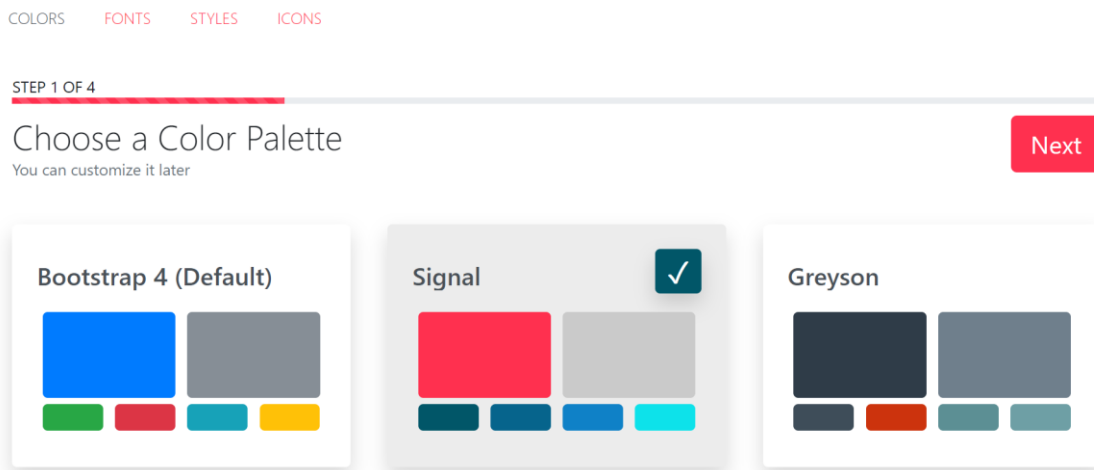
Knihovna webpack se stará o procházení veškerých zdrojových kódů a sestavení stromu závislostí jednotlivých souborů. Pomocí zaváděcích programů je možnost přidat různé závislosti do projektu, například možnost zpracování LESS nebo SASS. Oproti jiným knihovnám umožňuje webpack možnost automatické kompilace kódu při každé změně, kterou vývojář provede, a to za pomoci hlídačů (watchers). V případě využití webpacku v projektu je nutné Bootstrap nejprve nainstalovat pomocí správce balíčků npm. Spolu s ním je nutné nainstalovat i další knihovny, na jejichž funkcionalitě Bootstrap závisí, a to jQuery a Popper. Pro kompilaci vlastní konfigurace je nutné do webpacku přidat loadery sass-loader a post-css-loader s Autoprefixerem [61][62][63].

Tento způsob nejspíše nevyhovoval všem, a proto se postupem času začaly objevovat webové stránky, které se zabývají problematikou modifikace konfigurací Bootstrap závislostí. Na těchto stránkách je možné si nakonfigurovat vlastní komponenty a proměnné podobným způsobem, jako tomu bylo u konfiguračního nástroje obsaženého ve třetím vydání. Nové neoficiální webové stránky v porovnání s původním originálním nástrojem nabízejí mnoho nových funkcionalit včetně přednastavených stylů, témat a celých hotových šablon, které lze použít a následně měnit. Mezi tyto konfigurační stránky patří například bootstrap.build [57] a themestr.app [58], jejichž služby jsou zatím nezpлатněné.

V následujících odstavcích bude popsáno, jakým způsobem si lze nechat vygenerovat vlastní Bootstrap téma pomocí nástroje Themer ve službě themestr.app.

Po zvolení nástroje se uživatel dostane na průvodce s výběrem čtyř základních kroků k modifikaci každého téma. Těmito kroky jsou barevná paleta, font, styly komponent a styl ikon.

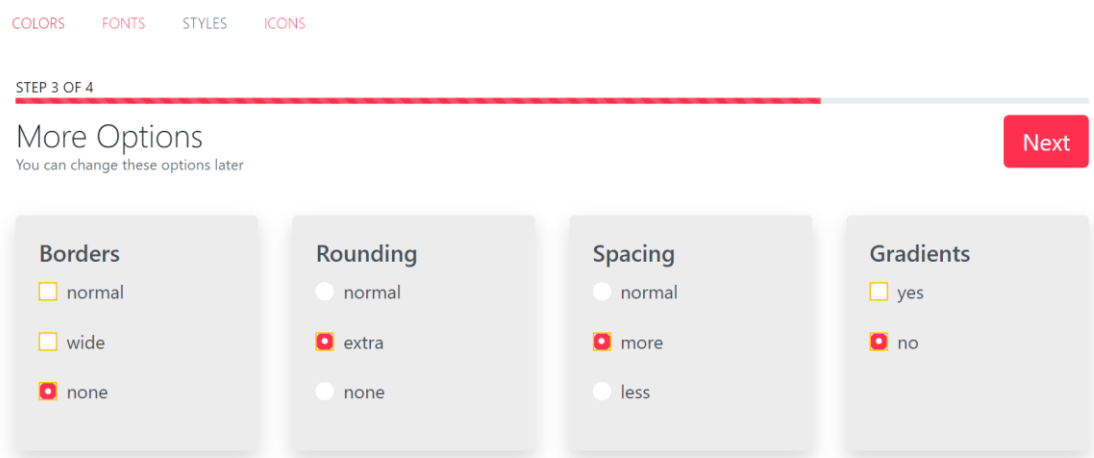




Obrázek 21 Themestr.app Themer – krok č. 1 [58]

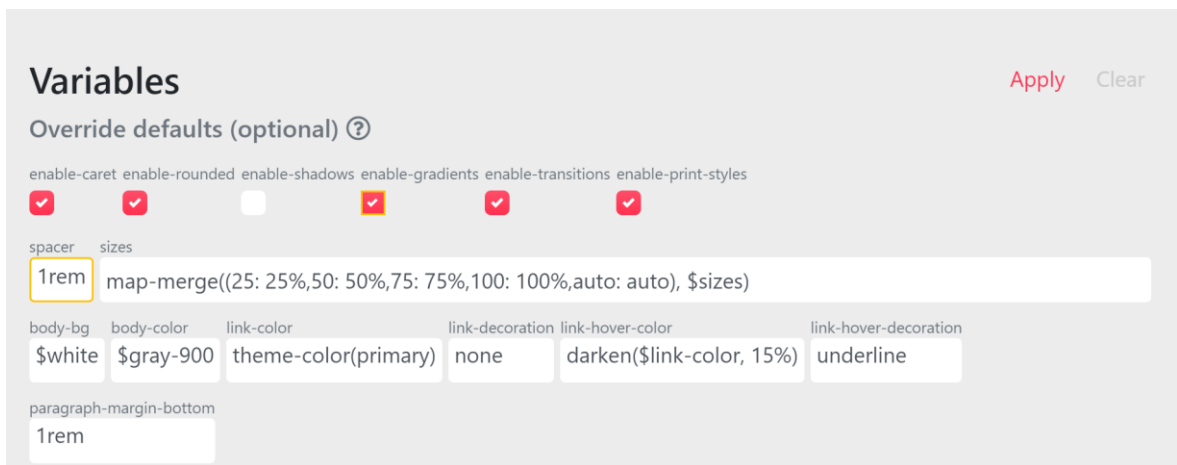
V prvním kroku se vybírá barevná paleta. Zde jsou dvě varianty, kterými je možné postupovat. Buď je možné si vybrat jednu paletu z již přednastavených, jak lze vidět na obrázku výše, anebo vybrat jakoukoliv, i když se zrovna nehodí k požadovanému vzhledu webu. V náhledu, který je k dispozici po čtvrtém kroku těchto výběrů je totiž možné si sestavit kompletně vlastní téma přepisováním hodnot všech proměnných k jednotlivým elementům a komponentám.

Po druhém kroku s výběrem fontu následuje základní definování, jak budou vypadat komponenty. Zda budou mít ohraničení, zaoblený okraj nebo jejich výplň bude z přechodu. Stejně jako tomu bylo u prvního kroku, tak i zde se dá dělat rozšířená modifikace, ale až v přehledu výsledného téma. Tento přehled a náhled se zobrazí po vybrání stylu ikonky v posledním kroku.



Obrázek 22 Themestr.app Themer – krok č. 3 [58]

V přehledu vytvořeného tématu budou zobrazeny výsledné změny včetně panelu, který nabízí rozšířené úpravy cílí již přímo na jednotlivé SCSS proměnné.



Obrázek 23 Obrázek 24 Themestr.app Themer – rozšířené úpravy [58]

V těchto nastaveních lze přepsat jakoukoliv nabízenou proměnnou. V SCSS inputu je zobrazeno, co vše bylo změněno a pokud s tím uživatel souhlasí, tak si nechá vygenerovat upravené CSS soubory, které lze poté stáhnout a vložit do projektu, díky čemuž získá vlastní téma.

## 9 OTESTOVÁNÍ RESPONZIVITY

V této části kapitoly se zaměřím na způsoby, jakými se dá ověřit, že výsledná stránka je správně responzivní a zobrazuje se na vícero jinak rozměrných zařízeních tak, jak má. Prvním, nejpohodlnějším a pravděpodobně nejvíce používaným způsobem jsou developerské (vývojářské) nástroje integrované v prohlížeči včetně emulátoru. Jedná se o sadu nástrojů, které mohou zpříjemnit a usnadnit práci již v průběhu vývoje webové prezentace. Pomocí těchto nástrojů lze procházet elementy vyskytující se na stránce včetně jejich vlastností, které lze v rámci prohlížeče následně upravit a sledovat změny v prohlížeči v reálném čase. Tímto způsobem se dají hledat i problematické elementy, které například přetékají přes rodičovský element nebo se z nějakého důvodu nezobrazují správně.

V prohlížeči Google Chrome lze nástroje zobrazit dvěma způsoby, a to kliknutím pravým tlačítkem myši a vybráním volby „Prozkoumat“, anebo pomocí kláves CTRL+SHIFT+I. Poté se na obrazovce zobrazí veškeré dostupné nástroje.

Ve vrchní liště nad zobrazovanou webovou stránkou se objevily možnosti k přepínání zobrazovaných zařízení, díky čemuž je možné otestovat responzivitu.



Obrázek 25 Google Chrome Developer Tools (DevTools) – responzivita

Z obrázku je patrné, že si uživatel může sám zvolit rozměry cílených obrazovek a zařízení, nebo případně přepnout na jiné zařízení pomocí přednastavených rozměrů. Například při volbě mobilu Moto G4 je možné vidět, že se rozměry změnila a je v prohlížeči znázorněno, jak by vypadal obsah právě na tomto zařízení. Výhodou tohoto způsobu kontroly responzivity je možnost kontroly přímo během vývoje, a tudíž i větší pravděpodobnost, že při vypuštění finální verze se bude jednat už o vyladěnou webovou stránku, co se responzivity týče.



Obrázek 26 DevTools – zobrazení obsahu na menším zařízení

Nevýhodou těchto nástrojů může být, že ač se vývojáři snaží přizpůsobit integrované emulátory co nejvíce skutečným zařízením, tak ne vždy může zobrazení obsahu reálně odpovídat fyzickému zařízení. To může být dáno tím, že se emulátory snaží přiblížit k shodnému rozlišení daného zařízení, nicméně neberou v potaz už další aspekty, jako například operační systém, používaný prohlížeč, jeho kompatibilitu aj. Na tento problém

jsem narazila v průběhu vývoje i já. Stránka se ve vývojářských nástrojích na testovaných virtuálních zařízeních jevila responzivní, nicméně při kontrole na fyzickém zařízení, konkrétně šlo o Apple iPad, se obsah nezobrazoval korektně. Problém byl jak v responzivitě, tak v použití moderních CSS vlastností, jelikož použitý prohlížeč nepodporoval některé vlastnosti z CSS3, které byly na webu použity.

Další možnosti k otestování responzivity webové prezentace jsou webové stránky zabývající se touto problematikou. Nevýhodou tohoto přístupu v porovnání s DevTools je, že testovaná webová prezentace musí být nahraná na hostingu a mít hypertextový odkaz, který se vloží do online nástroje k otestování. Po vložení odkazu je stránka v nástroji zobrazená na vícero zařízeních s různými rozměry a orientacemi (výška/šířka), díky čemuž může člověk vyhodnotit, zda je responzivita stránek kompatibilní s různými zařízeními či nikoliv. Těchto online nástrojů je mnoho. Příkladem mohou být tyto dvě stránky <http://responsivetesttool.com/>, <http://www.responsinator.com/>.

Řada organizací již v dnešní době nabízí služby s emulátory, které dokážou simulovat nejen rozlišení, ale i určitý typ počítače, operačního systému nebo prohlížeče. Díky těmto možnostem se dá otestovat i zpětná kompatibilita se staršími systémy nebo prohlížeči. Jedním z takových nástrojů je například BrowserStack, který nabízí více oddělených nástrojů řešících problematiku s testováním obsahu v odlišných prohlížečích, dále problematiku automatizovaného testování a taktéž vizuálního testování zaměřujícího se na UX a UI. Tyto pokročilé nástroje už jsou ale placené a je potřeba se zamyslet, zda jsou opravdu potřebné [60].

K otestování responzivity své výstupní stránky v průběhu vývoje jsem se rozhodla využít integrovaného emulátoru v DevTools, který mi umožňoval i rychlé náhledy změn pomocí ostatních nástrojů, které jsou v sadě obsaženy. Využívala jsem je k hledání chyb, náhledu změny barev, okrajů a poskládání objektů. Pro otestování responzivity finálních verzí na fyzických zařízeních jsem se rozhodla propojit svůj repozitář z verzovacího systému s platformou Netlify ([www.netlify.com](http://www.netlify.com)), která slouží k nasazení produkčních verzí. Pomocí této platformy jsem byla schopna při aktualizování větve se zásadními změnami nasadit zaktualizovanou verzi webových stránek a prohlédnout si ji online z kteréhokoliv zařízení v okolí.

## ZÁVĚR

Hlavním cílem této bakalářské práce bylo provést srovnání dostupných variant moderních CSS Frameworků, poukázat na jejich výhody, nevýhody a vyhodnotit vliv na proces vývoje responzivních webových stránek.

V prvním úseku teoretické části je čtenář seznámen se základními technologiemi, jejichž znalost je nezbytná k pochopení funkcionality pokročilých webových front-end frameworků, jež tuto znalost už v základu předpokládají. Dále proběhlo podrobné představení tří dostupných frameworků včetně jejich stručné historie. Tyto frameworky byly následně srovnány v praktické části.

Srovnání proběhlo na základě nastudování příslušných dokumentací a následného vyhodnocení podobností a rozdílů.

Dalším bodem práce bylo si vybrat jeden ze zkoumaných frameworků a implementovat ho do reálného projektu. V tomto případě se jednalo o framework Bootstrap, jehož nasazení do projektu se ukázalo dobrou volbou a značně urychlilo vývoj. V porovnání s kódováním webové prezentace bez využití frameworku se jednalo o poměrně plynulou práci. Z dřívějších zkušeností s kódováním internetových stránek na amatérské úrovni mohu vyhodnotit, že nasazení frameworku mi bylo obrovským přínosem, a to především v responzivitě, se kterou jsem dříve měla občas problémy.

Výstupem praktické části je funkční šablona webové prezentace, která může být nasazena do reálného projektu. Do budoucna se předpokládá, že tato šablona bude upravena k použití s redakčním systémem.

Posledním bodem v praktické části bylo otestování responzivity vytvořené šablony. Toho bylo docíleno pomocí dostupných technologií včetně vývojářských nástrojů implementovaných v prohlížeči Google Chrome. Testování na fyzických zařízeních proběhlo po nasazení finální verze na platformu Netlify.

## SEZNAM POUŽITÉ LITERATURY

- [1] CLEMENT, J. Digital users worldwide 2020. In: *Statista* [online] [cit. 24.07.2020]. Dostupné z: <https://www.statista.com/statistics/617136/digital-population-worldwide/>
- [2] G., Deyan. *How Many Websites Are There in 2020 - Active and Inactive!* [online]. 2019 [cit. 24.07.2020]. Dostupné z: <https://techjury.net/blog/how-many-websites-are-there/>
- [3] HTML History. In: *W3schools* [online] [cit. 28.02.2020]. Dostupné z: <https://www.w3schools.in/html-tutorial/history>
- [4] History of the Web. In: *World Wide Web Foundation* [online] [cit. 24.07.2020]. Dostupné z: <https://webfoundation.org/about/vision/history-of-the-web/>
- [5] Berners-Lee: Talk at Bush Symposium: Notes. In: [cit. 24.07.2020]. Dostupné z: [https://www.w3.org/Talks/9510\\_Bush/Talk.html](https://www.w3.org/Talks/9510_Bush/Talk.html)
- [6] NIX, Elizabeth. The World's First Web Site. In: *HISTORY* [online] [cit. 24.07.2020]. Dostupné z: <https://www.history.com/news/the-worlds-first-web-site>
- [7] Historie a vývoj HTML - - HTML5. In: [cit. 28.02.2020]. Dostupné z: <http://htmlguru.cz/uvod-historie.html>
- [8] The World Wide Web project. In: [cit. 24.07.2020]. Dostupné z: <http://info.cern.ch/hypertext/WWW/TheProject.html>
- [9] Different Versions Of HTML. In: [cit. 26.07.2020]. Dostupné z: <http://www.codefreetutorial.com/learn-html/76-different-versions-of-html>
- [10] The History of HTML | From the HTML 1.0 spec to XHTML 1.0... In: [cit. 26.07.2020]. Dostupné z: <https://www.yourhtmlsource.com/starthere/historyofhtml.html>
- [11] *Co je CSS - kaskádové styly - 1. díl* [online]. 2009 [cit. 29.02.2020]. Dostupné z: <https://blog.shoptet.cz/co-je-nutne-vedet-o-css-1-dil/>
- [12] A Look Back at the History of CSS. In: *CSS-Tricks* [online]. 18. 10. 2017 [cit. 01.03.2020]. Dostupné z: <https://css-tricks.com/look-back-history-css/>
- [13] Håkon Wium Lie: CSS was created to save HTML. In: *Root.cz* [online] [cit. 01.03.2020]. ISSN (SCHEME=ISSN) 1212-8309. Dostupné z: <https://www.root.cz/texty/hakon-wium-lie-css-was-created-to-save-html/>
- [14] Cascading Style Sheets, level 1. In: [cit. 27.07.2020]. Dostupné z: <https://www.w3.org/TR/REC-CSS1/#ref1>
- [15] Historie CSS. In: [cit. 27.07.2020]. Dostupné z: <https://www.jakpsatweb.cz/css/css-historie.html>
- [16] Difference Between Front-End & Back-End Developer | Pluralsight. In: [cit. 31.03.2020]. Dostupné z: <https://www.pluralsight.com/blog/film-games/whats-difference-front-end-back-end>

- [17] *11 Best CSS Frameworks To Look Forward In 2020* [online]. 2020 [cit. 27.07.2020]. Dostupné z: <https://www.lambdatest.com/blog/best-css-framework-2020/>
- [18] About Foundation | Foundation. In: [cit. 27.07.2020]. Dostupné z: <https://get.foundation/showcase/about.html>
- [19] ZURB - History. In: [cit. 27.07.2020]. Dostupné z: <https://zurb.com/history>
- [20] BITTNER, Honza. Lekce 2 - Úvod do CSS preprocesoru Sass. In: [cit. 27.07.2020]. Dostupné z: <https://www.itnetwork.cz/tutorial-moderni-webove-portfolio-sass>
- [21] HUGO GIRAUDEL. Sass Guidelines — Czech translation. In: [cit. 27.07.2020]. Dostupné z: <https://sass-guidelin.es/cz/>
- [22] Is Zurb Foundation in active development? · Issue #11767 · foundation/foundation-sites. In: *GitHub* [online] [cit. 27.07.2020]. Dostupné z: <https://github.com/foundation/foundation-sites/issues/11767>
- [23] Foundation Building Blocks. In: [cit. 27.07.2020]. Dostupné z: <https://zurb.com/blog/foundation-building-blocks-over-100-compo>
- [24] Download | Foundation for Sites. In: [cit. 27.07.2020]. Dostupné z: <https://download.get.foundation/sites/download/>
- [25] The Grid | Foundation for Sites 6 Docs. In: [cit. 27.07.2020]. Dostupné z: <https://get.foundation/sites/docs/grid.html>
- [26] Flex Grid | Foundation for Sites 6 Docs. In: [cit. 27.07.2020]. Dostupné z: <https://get.foundation/sites/docs/flex-grid.html>
- [27] XY Grid | Foundation for Sites 6 Docs. In: [cit. 27.07.2020]. Dostupné z: <https://get.foundation/sites/docs/xy-grid.html>
- [28] Flexbox History. In: [cit. 27.07.2020]. Dostupné z: <https://annairish.github.io/historicizing/history>
- [29] *The Foundation XY Grid: Taking Flexbox to Its Limits* [online]. 2017 [cit. 27.07.2020]. Dostupné z: <https://www.youtube.com/watch?v=1TIXCIXW0zc>
- [30] How to Choose the Right Foundation Grid. In: *Web Design Envato Tuts+* [online] [cit. 27.07.2020]. Dostupné z: <https://webdesign.tutsplus.com/tutorials/how-to-choose-the-right-foundation-grid--cms-29511>
- [31] ZURB Foundation Training | Learn the Flexbox based powerhouse XY Grid. In: [cit. 27.07.2020]. Dostupné z: <https://zurb.com/university/lessons/176>
- [32] Foundation for Emails 2 Docs | Getting Started with Sass. In: [cit. 27.07.2020]. Dostupné z: <https://get.foundation/emails/docs/sass-guide.html>
- [33] Foundation for Emails 2 Docs | Inky. In: [cit. 29.07.2020]. Dostupné z: <https://get.foundation/emails/docs/inky.html>



- [34] JEAN-BAPTISTE JUNG. Best CSS Frameworks for Building Better Websites in 2020. In: *CatsWhoCode* [online]. 15. 2. 2020 [cit. 29.07.2020]. Dostupné z: <https://catswhocode.com/css-frameworks/>
- [35] EMMANUEL. Top 6 popular CSS frameworks to use in 2020. In: *Medium* [online]. 16. 5. 2020 [cit. 29.07.2020]. Dostupné z: <https://medium.com/@liltuchi55/top-6-popular-css-frameworks-to-use-in-2020-d56fa7917d48>
- [36] About - Materialize. In: [cit. 29.07.2020]. Dostupné z: <https://materializecss.com/about.html>
- [37] Materialize - modern front-end framework for Material Design. In: *Quintagroup* [online] [cit. 13.07.2020]. Dostupné z: <https://quintagroup.com/cms/technology/materialize>
- [38] Introduction. In: *Material Design* [online] [cit. 29.07.2020]. Dostupné z: <https://material.io/design/introduction>
- [39] Flat design vs. Material Design. In: [cit. 29.07.2020]. Dostupné z: <https://www.justinmind.com/blog/flat-design-vs-material-design-differences/>
- [40] Grid - Materialize. In: [cit. 29.07.2020]. Dostupné z: <https://materializecss.com/grid.html>
- [41] Getting Started - Materialize. In: [cit. 29.07.2020]. Dostupné z: <https://materializecss.com/getting-started.html>
- [42] CONTRIBUTORS, Mark Otto, Jacob Thornton, and Bootstrap. History. In: [cit. 30.07.2020]. Dostupné z: <https://v4-alpha.getbootstrap.com/about/history/>
- [43] CONTRIBUTORS, Mark Otto, Jacob Thornton, and Bootstrap. Carousel. In: [cit. 02.08.2020]. Dostupné z: <https://getbootstrap.com/docs/4.0/components/carousel/>
- [44] CONTRIBUTORS, Mark Otto, Jacob Thornton, and Bootstrap. Navbar. In: [cit. 03.08.2020]. Dostupné z: <https://getbootstrap.com/docs/4.0/components/navbar/>
- [45] Media Queries | Foundation for Sites 6 Docs. In: [cit. 03.08.2020]. Dostupné z: <https://get.foundation/sites/docs/media-queries.html>
- [46] CONTRIBUTORS, Mark Otto, Jacob Thornton, and Bootstrap. Grid system. In: [cit. 03.08.2020]. Dostupné z: <https://getbootstrap.com/docs/4.0/layout/grid/>
- [47] Stack Overflow Trends. In: [cit. 03.08.2020]. Dostupné z: <https://insights.stackoverflow.com/trends?tags=materialize%2Czurb-foundation%2Ctwitter-bootstrap>
- [48] Google Trends. In: *Google Trends* [online] [cit. 03.08.2020]. Dostupné z: <https://trends.google.com/trends/explore?geo=US&q=bootstrap,zurb%20foundation,materialize>
- [49] CONTRIBUTORS, Mark Otto, Jacob Thornton, and Bootstrap. Download. In: [cit. 04.08.2020]. Dostupné z: <https://getbootstrap.com/docs/4.5/getting-started/download/>

- [50] materialize-css. In: *npm* [online] [cit. 04.08.2020]. Dostupné z: <https://www.npmjs.com/package/materialize-css>
- [51] bootstrap - npm. In: [cit. 04.08.2020]. Dostupné z: <https://www.npmjs.com/package/bootstrap>
- [52] foundation-sites. In: *npm* [online] [cit. 04.08.2020]. Dostupné z: <https://www.npmjs.com/package/foundation-sites>
- [53] WANG, Alvin. *Dogfalo/materialize* [online]. 2020 [cit. 04.08.2020]. Dostupné z: <https://github.com/Dogfalo/materialize>
- [54] *twbs/bootstrap* [online]. Bootstrap, 2020 [cit. 04.08.2020]. Dostupné z: <https://github.com/twbs/bootstrap>
- [55] *foundation/foundation-sites* [online]. Foundation, 2020 [cit. 04.08.2020]. Dostupné z: <https://github.com/foundation/foundation-sites>
- [56] Customize and download · Bootstrap. In: [cit. 04.08.2020]. Dostupné z: <https://getbootstrap.com/docs/3.4/customize/>
- [57] Free Bootstrap Builder - Build Themes & Templates. In: *Bootstrap.Build* [online] [cit. 04.08.2020]. Dostupné z: <https://bootstrap.build/>
- [58] Bootstrap Theme Builder and Customizer - Themestr.app. In: [cit. 04.08.2020]. Dostupné z: <https://themestr.app/>
- [59] Loops in CSS Preprocessors. In: *CSS-Tricks* [online]. 2. 12. 2016 [cit. 05.08.2020]. Dostupné z: <https://css-tricks.com/loops-css-preprocessors/>
- [60] Most Reliable App & Cross Browser Testing Platform | BrowserStack. In: [cit. 05.08.2020]. Dostupné z: <https://www.browserstack.com/>
- [61] CONTRIBUTORS, Mark Otto, Jacob Thornton, and Bootstrap. Webpack. In: [cit. 07.08.2020]. Dostupné z: <https://getbootstrap.com/docs/4.0/getting-started/webpack/>
- [62] CONTRIBUTORS, Mark Otto, Jacob Thornton, and Bootstrap. Theming Bootstrap. In: [cit. 07.08.2020]. Dostupné z: <https://getbootstrap.com/docs/4.0/getting-started/theming/>
- [63] CONTRIBUTORS, Mark Otto, Jacob Thornton, and Bootstrap. Build tools. In: [cit. 07.08.2020]. Dostupné z: <https://getbootstrap.com/docs/4.0/getting-started/build-tools/>
- [64] CONTRIBUTORS, Mark Otto, Jacob Thornton, and Bootstrap. Introduction. In: [cit. 08.08.2020]. Dostupné z: <https://getbootstrap.com/docs/4.0/getting-started/introduction/>
- [65] Stylesheet Demo. In: [cit. 09.08.2020]. Dostupné z: <http://www.viola.org/viola/style/sample.html>
- [66] Google Trends. In: *Google Trends* [online] [cit. 09.08.2020]. Dostupné z: [https://trends.google.com/trends/explore?q=%2Fm%2F054k6n\\_,%2Fm%2F0gjd0jv](https://trends.google.com/trends/explore?q=%2Fm%2F054k6n_,%2Fm%2F0gjd0jv)

## SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

- UI User interface – uživatelské rozhraní
- UX User experience – uživatelská zkušenost
- HTML HyperText Markup Language – hypertextový značkovací jazyk
- UTB Univerzita Tomáše Bati
- CSS Cascading Style Sheets – kaskádové styly
- SCSS Sassy CSS
- LESS Leaner Style Sheets
- SASS Syntatically Awesome Style Sheets
- CDN Content delivery network – síť pro doručování obsahu
- CMS Content Management System
- JS JavaScript

**SEZNAM OBRÁZKŮ**

Obrázek 1 První webová stránka (autor Tim Berners-Lee) [8] .....	13
Obrázek 2 Struktura HTML dokumentu vygenerovaná ve Visual Studio Code pomocí rozšíření html-5-boilerplate .....	14
Obrázek 3 Výběr balíčků frameworku Foundation ke stažení nebo instalaci [24].....	22
Obrázek 4 Ukázka formuláře před stažením balíčku Custom Foundation [24] .....	23
Obrázek 5 Ukázka kódu z Visual Studia – využití gridu Materialize .....	30
Obrázek 6 Grafické zobrazení HTML kódu z obrázku č. 6.....	31
Obrázek 7 Srovnání počtu vyhledávaných dotazů na Stack Overflow[47] .....	40
Obrázek 8 Srovnání počtu vyhledávaných dotazů pomocí Google Trends [48] .....	41
Obrázek 9 Návrh budoucí stránky .....	43
Obrázek 10 Počáteční struktura projektu.....	44
Obrázek 11 Snímek z obrazovky – naimportované soubory v hlavičce.....	45
Obrázek 12 Ukázka původního kódu komponenty navbar [44] .....	46
Obrázek 13 Modifikovaný kód Bootstrap navbaru.....	47
Obrázek 14 Vzhled navbaru po úpravách.....	48
Obrázek 15 Vzhled navbaru po úpravách a scrollu .....	48
Obrázek 16 Snímek z obrazovky k popsání funkčnosti gridu .....	50
Obrázek 17 Výchozí tabulka s označením table-hover .....	52
Obrázek 18 Modifikovaná tabulka .....	53
Obrázek 19 Konfigurace mřížky a bodů zalomení ve frameworku Bootstrap 3 [56].....	54
Obrázek 20 jQuery zásuvné moduly [56] .....	55
Obrázek 21 Themestr.app Themer – krok č. 1 [58].....	57
Obrázek 22 Themestr.app Themer – krok č. 3 [58].....	57
Obrázek 23 Obrázek 24 Themestr.app Themer – rozšířené úpravy [58].....	58
Obrázek 25 Google Chrome Developer Tools (DevTools) – responzivita .....	59
Obrázek 26 DevTools – zobrazení obsahu na menším zařízení .....	60

**SEZNAM TABULEK**

Tabulka 1 Porovnání Viola-style stylopisu [65] s dnešním CSS stylopisem .....	15
Tabulka 2 Obecné srovnání frameworků .....	36
Tabulka 3 Aktuální podpora různých typů gridu .....	37
Tabulka 4 Vlastnosti mřížky .....	37
Tabulka 5 Podporované velikosti zařízení frameworkem Bootstrap .....	37
Tabulka 6 Podporované velikosti zařízení frameworkem Materialize .....	38
Tabulka 7 Podporované velikosti zařízení frameworkem Foundation .....	38
Tabulka 8 Aktuálnost frameworků dle dat ze služby GitHub [53][54][55] .....	40
Tabulka 9 Průměrný počet stažení za týden z dat správce balíčků npm [50][51][52] .....	40

## SEZNAM PŘÍLOH

Příloha P I: CD s bakalářskou prací včetně zdrojových kódů

## **PŘÍLOHA P I: NÁZEV PŘÍLOHY**

Přiložené CD obsahuje:

- Bakalářskou práci ve formátu .pdf      BP\_MelanieLezakova\_A17131.pdf
- Zdrojové kódy      BP\_Prilohy\_MelanieLezakova\_A17131.zip