

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

Front-End frameworky pro tvorbu webové aplikace

Bc. Vladimír Pastorek

© 2020 ČZU v Praze

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Vladimír Pastorek

Systémové inženýrství a informatika
Informatika

Název práce

Front-End frameworky pro tvorbu webové aplikace

Název anglicky

Front-End frameworks for creating web application

Cíle práce

Cílem předkládané diplomové práce bude popis, analýza a porovnání nejrozšířenějších front-end frameworků pro vytváření webových aplikací.

V teoretické části práce budou popsány postupy a metodiky, v rámci současných webových technologií, které je možné využít při tvorbě webu. V oblasti vývoje front-endového prostředí bude kladen důraz na popsání sad nástrojů pro uplatnění kaskádových stylů a javascriptu.

Dílním výsledkem praktické části diplomové práce bude na základě syntézy zjištěných poznatků vytvoření webové aplikace pomocí různých metod a frameworků, které je možné při vývoji použít. Následně budou použité metody a frameworky porovnány na základě vhodně stanovených kritérií.

Metodika

Metodika diplomové práce je založena na studiu a analýze odborných informačních zdrojů vztahujících se ke zvolenému tématu. Praktická část práce bude zaměřena na návrh a následné vytvoření funkční webové aplikace pomocí odlišných metod a front-end frameworků. Výsledné verze webové aplikace budou otestovány a porovnány na základě vhodně stanovených kritérií. Syntézou poznatků teoretické a praktické části práce budou formulovány závěry diplomové práce.

Doporučený rozsah práce

60-80 stran

Klíčová slova

html, css, javascript, webová aplikace, frameworky, front-end, web

Doporučené zdroje informací

HOGAN, B P. *HTML5 a CSS3 : výukový kurz webového vývoje*. Brno: Computer Press, 2011. ISBN 978-80-251-3576-1.

HYSLOP, B. – CASTRO, E. *HTML5 a CSS3 : názorný průvodce tvorbou WWW stránek*. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.

JOERG, Kraus. *Introducing Bootstrap 4*. New York, NY: Springer Science+Business Media, 2016. ISBN 9781484223819.

MOFFITT, Jack a Frederic DAOUD. *Seven web frameworks in seven weeks: adventures in better web apps*. Raleigh, NC: Pragmatic Bookshelf, [2014]. ISBN 978-1937785635.

Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

Ing. Petr Benda, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 3. 9. 2019

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 14. 10. 2019

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 30. 03. 2020

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Front-End frameworky pro tvorbu webové aplikace" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2020

Poděkování

Rád bych touto cestou poděkoval panu Ing. Petru Bendovi Ph.D. za odborné vedení závěrečné práce a za pomoc při práci na projektu.

Front-End frameworky pro tvorbu webových aplikací

Souhrn

Diplomová práce je zaměřena na popsání a analýzu front-endových sad nástrojů pro tvorbu webu a webových aplikací. První část práce se zabývá teoretickými principy problematiky vývoje právě webu a webových aplikací, přičemž je kladen velký důraz na responzivní design, který je v současné době stěžejní a tyto nástroje se jím ve větší míře zabývají. Dále jsou popsány front-endové frameworky a použité metody vícekriteriální analýzy variant. Ve druhé části jsou vytvořeny verze webových aplikací dle popisovaných nejpoužívanějších nástrojů. Pro celkem pět frameworků je tedy vytvořeno pět webových aplikací, které slouží k určení a hodnocení vhodných kritérií. Za pomoci vícekriteriální analýzy variant, konkrétně metod váženého součtu a AHP (z anglického názvu Analytic Hierarchy Process) a získaných praktických poznatků z vývoje aplikace pomocí testovaných frameworků bylo zjištěno, že nejlépe hodnocený nástroj je Bootstrap. Výsledky jsou následně formulovány a komparovány se závěry z jiných odborných zdrojů.

Klíčová slova: html, css, javascript, webová aplikace, frameworky, front-end, web

Front-End frameworks for creating web application

Summary

The diploma thesis presents on describe and analysis front-end package of tools for creating websites and web applications. The first part of the thesis deals with the theoretical principles of development web and web application with strong emphasis on responsive design which is currently important, and these frameworks are more concerned with that. Further, front-end frameworks and methods of multi-criteria analysis of variants are described. In the second part are created versions of web application according to described most used tools. For a total of five frameworks, five web applications are created to determine and evaluate the appropriate criteria. Using the multi-criteria analysis of variants, specifically the methods of weighted sum and AHP (Analytic Hierarchy Process) and gained practical knowledge of application development using tested frameworks, was found that Bootstrap is the best rated tool. Then the results are formulated and compared with conclusions from other professional sources.

Keywords: html, css, javascript, web application, frameworks, front-end, web

Obsah

1 Úvod.....	11
2 Cíl práce a metodika	12
2.1 Cíl práce	12
2.2 Metodika	12
3 Teoretická východiska	13
3.1 Základní technologie pro vývoj webu a webových aplikací.....	13
3.1.1 Značkovací jazyk HTML.....	13
3.1.2 Kaskádové styly CSS.....	14
3.2 Responzivní web design.....	15
3.2.1 Historie.....	16
3.2.2 Vývoj front-endu.....	19
3.2.3 Layout a grafický návrh.....	20
3.2.4 Obrázky.....	22
3.2.5 Navigace a jednotky.....	25
3.2.6 Tabulky	26
3.2.7 Javascriptové možnosti	28
3.2.8 Rychlost webu.....	29
3.3 Front-end frameworky	31
3.3.1 Bootstrap.....	32
3.3.2 Foundation	35
3.3.3 PureCSS	37
3.3.4 Bulma.....	38
3.3.5 Materialize	40
3.4 Vícekriteriální analýza variant (VAV).....	41
3.4.1 Saatyho metoda.....	42
3.4.2 Metoda váženého součtu.....	43
3.4.3 Metoda AHP	44
4 Vlastní práce	45
4.1 Vývoj webové aplikace	45
4.2 Charakteristika hodnotících kritérií.....	50
4.2.1 Hodnotící kritéria - Bootstrap	54
4.2.2 Hodnotící kritéria - Foundation	56
4.2.3 Hodnotící kritéria - PureCSS	58
4.2.4 Hodnotící kritéria - Bulma.....	60
4.2.5 Hodnotící kritéria - Materialize	62
4.3 Analýza kritérií.....	64
4.3.1 Ohodnocení kritérií variant.....	64

4.3.2	Určení vah kritérií (Saatyho metoda).....	65
4.3.3	Analýza metodou váženého součtu.....	66
4.3.4	Analýza metodou AHP	67
5	Výsledky a diskuse	70
6	Závěr.....	72

Seznam obrázků

Obrázek 1 -	Tabulka vygenerovaná pomocí HTML kódu.....	13
Obrázek 2 -	Tabulka vygenerovaná upravená pomocí CSS kódu	14
Obrázek 3 -	Ilustrace responzivního designu.....	15
Obrázek 4 -	Čas strávený na internetu dle zařízení ve Spojených Státech	17
Obrázek 5 -	Porovnání webů mobile friendly.....	18
Obrázek 6 -	Nárůst návštěvnosti a tržeb providera Plusnet	19
Obrázek 7 -	Ilustrační wireframe sociální sítě Facebook	21
Obrázek 8 -	Grafický návrh responzivního webu	22
Obrázek 9 -	Porovnání velikostí klasických obrázkových formátu s formátem WebP	24
Obrázek 10 -	Ilustrace responzivních obrázků	25
Obrázek 11 -	Záhlaví webu na mobilním zařízení zobrazující hamburger menu.....	26
Obrázek 12 -	První způsob tvorby responzivní tabulky	27
Obrázek 13 -	Druhý způsob tvorby responzivní tabulky	27
Obrázek 14 -	Třetí způsob tvorby responzivní tabulky	27
Obrázek 15 -	Čtvrtý způsob tvorby responzivní tabulky	28
Obrázek 16 -	Kriteriální matice	42
Obrázek 17 -	Saatyho matice	43
Obrázek 18 -	Hierarchická struktura typické úlohy VAV	44
Obrázek 19 -	Wireframe vytvářené webové aplikace č.1	46
Obrázek 20 -	Wireframe vytvářené webové aplikace č.2	46
Obrázek 21 -	Zlomové body v grafické podobě u vybraných frameworků.....	48
Obrázek 22 -	Navigace řešená hamburger menu X scrollovací horizontální menu	48
Obrázek 23 -	Vytvořená webová aplikace - desktop verze	49
Obrázek 24 -	Vytvořená webová aplikace - mobilní verze	49
Obrázek 25 -	Měření rychlosti načtení webu pomocí Chrome DevTools	51
Obrázek 26 -	Grafické zobrazení výsledků prováděných analýz	70

Seznam tabulek

Tabulka 1 -	Používané jednotky při tvorbě webu	26
Tabulka 2 -	Souhrn hodnotících kritérií.....	54
Tabulka 3 -	Ohodnocení kritérií dle jednotlivých variant.....	65
Tabulka 4 -	Určené váhy kritérií dle Saatyho metody	65
Tabulka 5 -	Standardizovaná kriteriální matice	67
Tabulka 6 -	Celkové pořadí pomocí analýzy metodou váženého součtu	67
Tabulka 7 -	Váhy variant dle kritéria 1	68
Tabulka 8 -	Váhy variant dle kritéria 4.....	68
Tabulka 9 -	Souhrn normalizovaných dílčích vah jednotlivých variant podle všech kritérií ...	68

Tabulka 10 - Celkové pořadí pomocí analýzy metodou AHP	69
Tabulka 11 - Celkové výsledky prováděných analýz	70

1 Úvod

Internet se za posledních několik let změnil, stále se mění a vyvíjí. Stejně tak tomu je i s weby a webovými aplikacemi, které tvoří velkou část internetových služeb a internetového prostoru.

Při vývoji webu a webových aplikací je potřeba zajistit mnoho věcí. Dříve například, když webové služby byly v počítačích a ve své podstatě neexistovalo ani tolik technologií pro vytváření webu, neřešil se ani responzivní design. Je to v celku logické, chytré telefony v té době neexistovaly a spíše se kladl důraz na vývoj celkových technologií a jazyků. Ty již aktuálně responzivní design řešit umí společně s webovými prohlížeči a velkou zásluhu na tom má právě rozvoj chytrých zařízení. Do této kategorie spadají zejména telefony a tablety, ale mohou to být všechna zařízení, která mají vlastní operační systém a mohou si například web otevřít a zpracovat, například v internetovém prohlížeči.

Nicméně i kdyby existovala jen podkategorie chytrých telefonů, tak není pouze jedno toto zařízení. Výrobci smartphonů je mnoho a stejně tak modelů, které vydávají a která jsou každým rokem nahrazována novějšími. Variabilita těchto zařízení je tedy velmi vysoká a liší se mnoha parametry, kdy pro vývojáře webu je nejdůležitější zejména velikost zobrazovacího prostoru a rozlišení. Optimalizace veškerého obsahu je proto dalším důležitým krokem pro správné zobrazení a zajištění tak pohodlí všem návštěvníkům dané webové stránky, nebo aplikace. S tím následně mohou souviset obchodní příležitosti a další důležité aspekty pro zajištění zisku.

Celá disciplína tvorby webu a webových aplikací se dělí na front-end a back-end. Zároveň však je možné mít web postavený bez back-endové části a nepoužívat ji, protože web bez ní bude fungovat, pouze tak bude statický. Pokud by tomu bylo naopak a nebyla by zajištěna front-endová část, web by byl prakticky nepoužitelný. Proto při tomto smyslu uvažování je možné tvrdit, že front-end je důležitější a zejména také pro návštěvníky a uživatele, kteří vidí jen tuto část.

Proto se po výrazném rozvoji webových technologií rozšířily také webové sady nástrojů pro vývojáře – frameworky, které často sjednocují styl pro vývoj a řeší již mnoho problémů, prvků, komponentů, které mohou značně urychlit vytváření webu i webových aplikací a celkově zlepšit funkčnost a výkon.

Právě front-endovým frameworkům se věnuje tato práce. Ty jsou popsány společně s důležitými informacemi a následně analyzovány.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem předkládané diplomové práce bude popis, analýza a porovnání nejrozšířenějších front-end frameworků pro vytváření webových aplikací.

V teoretické části práce budou popsány postupy a metodiky, v rámci současných webových technologií, které je možné využít při tvorbě webu. V oblasti vývoje front-endového prostředí bude kladen důraz na popsání sad nástrojů pro uplatnění kaskádových stylů a Javascriptu.

Dílčím výsledkem praktické části diplomové práce bude na základě syntézy zjištěných poznatků vytvoření webové aplikace pomocí různých metod a frameworků, které je možné při vývoji použít. Následně budou použité metody a frameworky porovnány na základě vhodně stanovených kritérií.

2.2 Metodika

Metodika diplomové práce je založena na studiu a analýze odborných informačních zdrojů vztahujících se ke zvolenému tématu. Praktická část práce bude zaměřena na návrh a následné vytvoření funkční webové aplikace pomocí odlišných metod a front-end frameworků. Výsledné verze webové aplikace budou otestovány a porovnány na základě vhodně stanovených kritérií. Syntézou poznatků teoretické a praktické části práce budou formulovány závěry diplomové práce.

3 Teoretická východiska

3.1 Základní technologie pro vývoj webu a webových aplikací

3.1.1 Značkovací jazyk HTML

Za celkovým úspěchem webu stojí víceméně jednoduchý značkovací jazyk, který podporují téměř všechna zařízení – jazyk HTML. Každý web, nebo webová aplikace se zakládá alespoň na minimálním množství kódu jazyka HTML, bez toho by to nebyla webová stránka. HTML je vhodný pro definování významu obsahu na webu, následně jazyk CSS (který je taktéž důležitý při provozu webu) určuje, jak tento obsah vypadá (Castro, 2012).

Jazyk HTML vznikl na počátku 90. let v podobě stručného dokumentu popisující několik elementů používaných pro tvorbu webových stránek. Tyto elementy, prvky, popisují různé části webu, jako je například nadpis, odstavec, tabulka a podobně. Definují se specifickými značkami, které tak mají svou určitou roli a funkci, doplněné pomocí atributů. S přirozeným vývojem se různé verze HTML od sebe lišily, číslo definované jako verze jazyka se postupně zvyšovalo a aktuálně nejnovější verze tohoto jazyka je HTML5. Tato verze se snaží reflektovat potřeby současných, ale i budoucích webů a webových aplikací, přestože zdědil mnoho vlastností svých předchozích verzí. (Hogan, 2011; Castro, 2012).

```
<table class="tabulka1">
  <thead>
    <tr><th>Jméno</th><th>Město</th><th>Věk</th></tr>
  </thead>
  <tbody>
    <tr><td>Vladimír</td><td>Mělník</td><td>24 let</td></tr>
    <tr><td>Petra</td><td>Praha</td><td>20 let</td></tr>
    <tr><td>Petr</td><td>Brno</td><td>32 let</td></tr>
  </tbody>
</table>
```

Jméno	Město	Věk
Vladimír	Mělník	24 let
Petra	Praha	20 let
Petr	Brno	32 let

Obrázek 1 - Tabulka vygenerovaná pomocí HTML kódu

Zdroj: Vlastní zpracování

3.1.2 Kaskádové styly CSS

Jak již bylo zmíněno, jazyk HTML definuje význam obsahu na webu, a pro grafickou podobu obsahu, se využívá kaskádových stylů. Tyto styly se objevily až několik let po vzniku jazyka HTML, avšak stejně jako se vyvíjel jazyk HTML, tak se i analogicky vyvíjel jazyk CSS. Momentálně je v nejnovější verzi CSS3, který je přirozeným rozšířením svých starších verzí a je mnohem rozsáhlejší a mocnější než staré verze. Zavádí totiž několik vizuálních efektů, jako jsou například barevné přechody, zaoblené rohy a další funkce, což se v dřívějších verzích muselo řešit mimo CSS (Castro, 2012).

Pomocí selektorů se identifikuje HTML element, kterému se následně mohou měnit vlastnosti a styly. Nejpoužívanějšími jsou typové selektory, selektory ID, třídy, následníka a pseudotřídy. Možná je také multiselekce pro seskupování více prvků. Celá syntaxe je poměrně jednoduchá a skládá se z ucelených bloků jednotlivých deklarací oddělených středníkem pro dané vlastnosti. Následně je možnost použít celkové styly v externím importovaném souboru, nebo styly mít v HTML dokumentu, konkrétně v hlavičce `<head>` a ve značce `<style>`. V tomto případě se styly píší ve složených závorkách pro selektovaný prvek. Druhou možností je styl aplikovat přímo do vybraného HTML elementu, a to přes atribut `style=""`. (Hogan, 2011).

```
.tabulka1 { background: yellow; border-collapse: collapse; border: 1px solid #222; }  
.tabulka1 th { background: #222; color: white; text-align:left; }  
.tabulka1 th, .tabulka1 td { border: 1px solid #222; padding: 7px 9px; }  
.tabulka1 tr:nth-child(odd) { background: khaki; }
```

Jméno	Město	Věk
Vladimír	Mělník	24 let
Petra	Praha	20 let
Petr	Brno	32 let

Obrázek 2 - Tabulka vygenerovaná upravená pomocí CSS kódu

Zdroj: Vlastní zpracování

S vývojem technologií a následným příchodem chytrých zařízení, které dokážou zobrazovat web a webové aplikace, přišla potřeba celý obsah přizpůsobit velikosti zařízení. Tento proces přizpůsobení právě dokážou zpracovávat kaskádové styly.

3.2 Responzivní web design

Responzivnímu designu je věnována značná teoretická rešeršní část a celkově z jeho principu vychází i front-endové frameworky. Mnoho poznatků z této kapitoly je reflektováno i ve vlastní práci, kdy je na praktikách responzivního designu aplikován vývoj webových aplikací.

Pojem responzivní web označuje takovou webovou stránku, nebo webovou aplikaci, která dokáže obsah a svou strukturu přizpůsobit dle velikosti a rozlišení zařízení, na kterém je web zobrazen. Děje se tak zejména díky značkovacímu jazyku HTML (Hypertext Markup Language) a kaskádovým stylům CSS (Cascading Style Sheets) (W3C, 2015).

Responzivní web design, nebo také používající se zkratka RWD, je v současné době často neocenitelná disciplína v kontextu webového a webdesignového průmyslu. Zobrazovaný obsah nejen že musí vypadat dobře a jak již bylo zmíněno, musí se umět přizpůsobit, také musí být snadno použitelný. Také by se nemělo stát, že by nějaká obsahová část, která nemá problém se vejít na větší zařízení, následně na menším zobrazení zmizela. Informace na webu by se neměly ztrácet napříč zařízeními, musí se zkrátka adaptovat, přizpůsobit (Kadlec, 2014).



Obrázek 3 - Ilustrace responzivního designu

Zdroj: Cactusgroup.com

Responzivní web zobrazuje obsah založený na dostupném prostoru prohlížeče. Pokud návštěvník otevře responzivní web na ploše a poté změní velikost okna prohlížeče, bude se obsah dynamicky pohybovat, aby se uspořádal (alespoň teoreticky) optimálně pro okno

prohlížeče. V mobilních telefonech je tento proces automatický, prohlížeč zkontroluje dostupné místo a poté se obsah zobrazí v ideálním uspořádání (Cao, 2015).

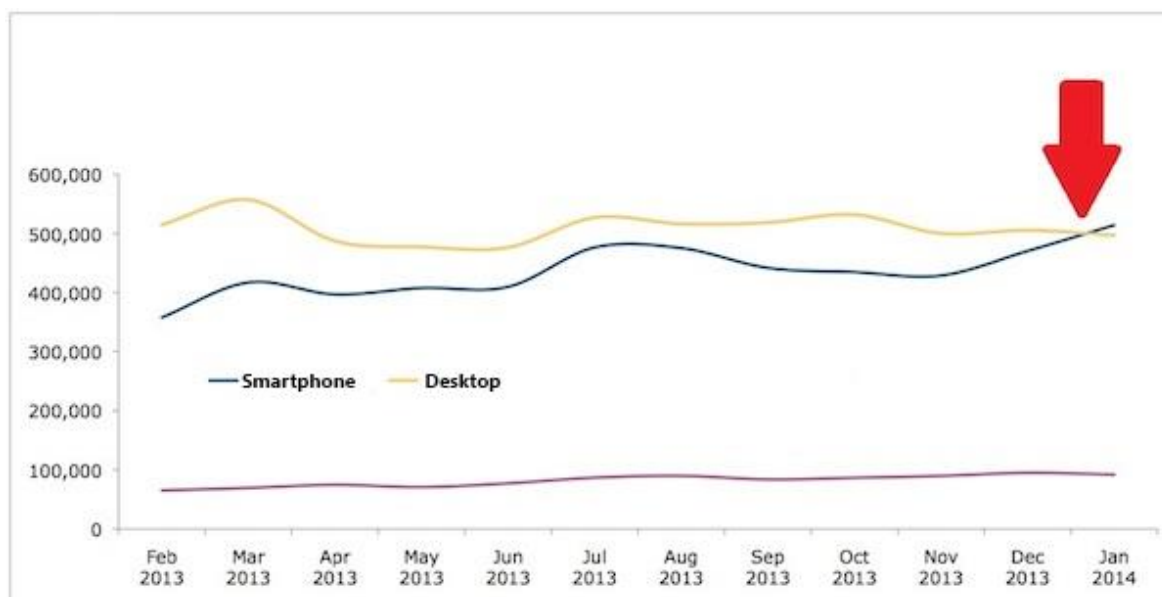
Takový proces je jednoduchý, plynulý a ve své podstatě to znamená, že uživatelé mohou mít přístup k obsahu na webu a prohlížet si jej na svém kapesním zařízení. Aby toto však bylo reálné, je vyžadována u responzivního designu velmi dobrá konceptualizace webu a znalost potřeb a přání koncových uživatelů. Jak bylo zmíněno, responzivní design je jednoduchý, a to i z hlediska implementace, kdy jeho nasazení na web vyžaduje méně práce. Také poskytuje méně práce pro následnou kontrolu designu pro každou velikost obrazovky (Itzkovitch, 2012).

Responzivní návrháři často vytvářejí jediný design, který se použije na všech obrazovkách, a obvykle má design větší rozlišení a následně se pomocí zlomových bodů určí, jaké úpravy budou provedeny pro dolní a horní hranice rozlišení u těchto zlomových bodů. Rovnoměrnost a plynulost jsou rozhodujícími faktory při poskytování dobré uživatelské zkušenosti a použitelnosti. Je důležité sledovat vizuální hierarchii responzivních návrhových projektů a následně je udržet, protože použité prvky se při dosažení zlomového bodu často přesouvají – zpravidla pod následující prvek. To ve výsledku znamená mnoho testů s mnoha různými zařízeními při různých rozlišeních, aby si vývojář byl jistý, že se vše a všude zobrazuje stejně a v pořádku. Pokud je layout a konstrukce prostoru relativně jednoduchá, bude se obsah dobře a snadno zobrazovat (Holst, 2012).

SEO je dalším velkým plusem pro použití responzivního designu. Weby, které využívají responzivního designu jsou pro vyhledávače přátelštější.

3.2.1 Historie

Důvod, proč se začal responzivní design tolik řešit, je zejména díky rozmanitosti a růstu využívání chytrých zařízeních – konkrétně smartphonů a tabletů. To mělo za následek nutnost vývoje standardů pro vytváření webových stránek. Jak uvádí Murtagh (2014), zlom přišel začátkem roku 2014 kdy, jak je patrné i z obrázku č.4, čas strávený na internetu v mobilním prohlížeči předběhl čas strávený v prohlížeči na počítači, alespoň dle průzkumů ve Spojených Státech.



Obrázek 4 - Čas strávený na internetu dle zařízení ve Spojených Státech
Zdroj: Searchenginewatch.com

Svůj chytrý telefon uživatelé nejčastěji využívali právě pro přístup na internet, kontrolu emailů. V té době se na trhu pohybovaly například smartphony Apple, konkrétně iPhone 6 a 6 plus a trh se začal zaplňovat zejména telefony větších rozměrů, hlavně tedy větších displejů, nejvíce se systémem Android (v té době ve vyvíjené verzi 5). Webové prohlížeče, stejně jako mobilní operační systémy a aplikace, se zdokonalovaly a přejímaly a implementovaly standardy, stejně tak nutily webové vývojáře optimalizovat weby a díky tomu také vývojáře frameworků a různých sad pro vývoj webů (Murtagh, 2014).

Následující rok, tj. 2015, udělal i největší vyhledávač Google aktualizaci týkající se webů a jejich řazení. Google začal upřednostňovat responzivní weby, nebo mobilní weby a zároveň jim dával značku „mobile-friendly“. To již byl i dost důležitý krok z marketingového hlediska pro weby, které se ve výsledcích předháněly. Vyhledávací engine hledal na webech hlavně kvalitní a relevantní výsledky, kdy text je snadno čitelný bez jakéhokoliv zvětšování, obsah se přizpůsobí a nejsou žádné problémy se zobrazením. Jak Google ve své oficiální zprávě uvádí, změny se týkají pouze při vyhledávání právě na mobilním zařízení, ve všech jazycích a také na individuálních podstránkách, a na celém webu (Google, 2015).



Obrázek 5 - Porovnání webů mobile friendly
Zdroj: Google.com

Svou kapitolu odehrál také mobilní web, ještě v dobách, kdy se responzivní web tolik neřešil. Mobilní weby se snadno poznají podle subdomény, v příkladu *seznam.cz*, kdy mobilní verze mají formát *m.seznam.cz*. V praxi tedy pokud prohlížeč, nebo server zjistí, že návštěvník používá mobilní zařízení, přesměruje ho tak na tuto mobilní verzi. Provozovatel webu má tedy v zásadě dvě verze webu – mobilní a desktopovou. Dle statistik je však patrné, že od této metody se opouští (Everts, 2013).

Pro toto období je tedy zásadních několik věcí:

- V roce 2014, který jak již bylo zmíněno, byl vcelku přelomový, také popularita a využívanost mobilních webů klesla o 20% a naopak responzivní weby vzrostly o 37% (Bustos, 2015).
- Více než třetina uživatelů stejně raději využila desktopovou verzi webu, pokud jsou na mobilním zařízení a dostanou na výběr ze dvou možností. Stejně tak uživatelé tráví více času na plné verzi pro desktopové zařízení a stejně tak konverze na webu přicházejí z plné verze (Everts, 2013).
- Internetový vyhledávač Google umisťuje lépe responzivní weby, než mobilní verze (Google, 2015).

- Mobilní weby a aplikace byly navrhovány pro specifické rozlišení smartphonů. Pokud by uživatel navštívil stránku na tabletu, dostane se mu taktéž mobilní verze, což na tabletech s větším displejem a rozlišením, nesplní očekávanou funkci (Everts, 2013).

Od mobilních webů se tedy ve větší míře opustilo a nyní jsou již spíše dodatečným řešením, nebo řešením pro velmi složité weby. Zejména tedy vývoj bude dražší a často nesplní svůj účel. Naopak redesign, a investice pro přechod na responzivní web může být tím nejlepším možným krokem, jak poukazuje studio od Googlu týkající se Britského Internetového Providera – Plusnet. Ten v té době podchytil rozšiřující se vlnu smartphonů a tabletů a při přechodu na responzivní web zaznamenal dvojnásobný nárůst návštěvnosti a desetinásobný nárůst konverzí na mobilech a tabletech (Google, 2013).

Think with Google English - United States

Plusnet Uses Responsive Web Design, Sees Traffic Grow 2x and Sales 10x on Smartphones and Tablets

*Obrázek 6 - Nárůst návštěvnosti a tržeb providera Plusnet
Zdroj: Google.com*

3.2.2 Vývoj front-endu

Při vyvíjení front-endu si je potřeba uvědomit, že se veškeré zpracování kódu děje ve webovém prohlížeči, tedy na straně klienta. Prohlížeč si kód textového formátu stáhne ze serveru a u sebe zpracuje, konkrétně HTML, CSS a Javascript. Výzvou spojenou s vývojem front-endu je to, že nástroje a techniky používané k vytvoření front-endu webové stránky se neustále mění a vyvíjí a vývojář musí mít velký přehled o aktuálních metodách. To, co platilo například před rokem dnes již nemusí být aktuální, ačkoliv se zpětná kompatibilita ve většině případů zachovává (Lindley, 2018).

Cílem navrhování webu je zajistit, aby uživatelé při otevření webu viděli informace ve formátu, který je snadno čitelný a relevantní. To je však komplikováno skutečností, že uživatelé používají velké množství zařízení s různou velikostí obrazovky a rozlišením. To nutí designera,

aby při navrhování webu tyto aspekty zohlednil. Musí se ujistit, že jejich stránky se zobrazují správně v různých prohlížečích (cross-browser), různých operačních systémech (cross-platform) a různých zařízeních (cross-device), což vyžaduje pečlivé plánování na straně vývojáře (Marcotte, 2014; Lindley, 2018).

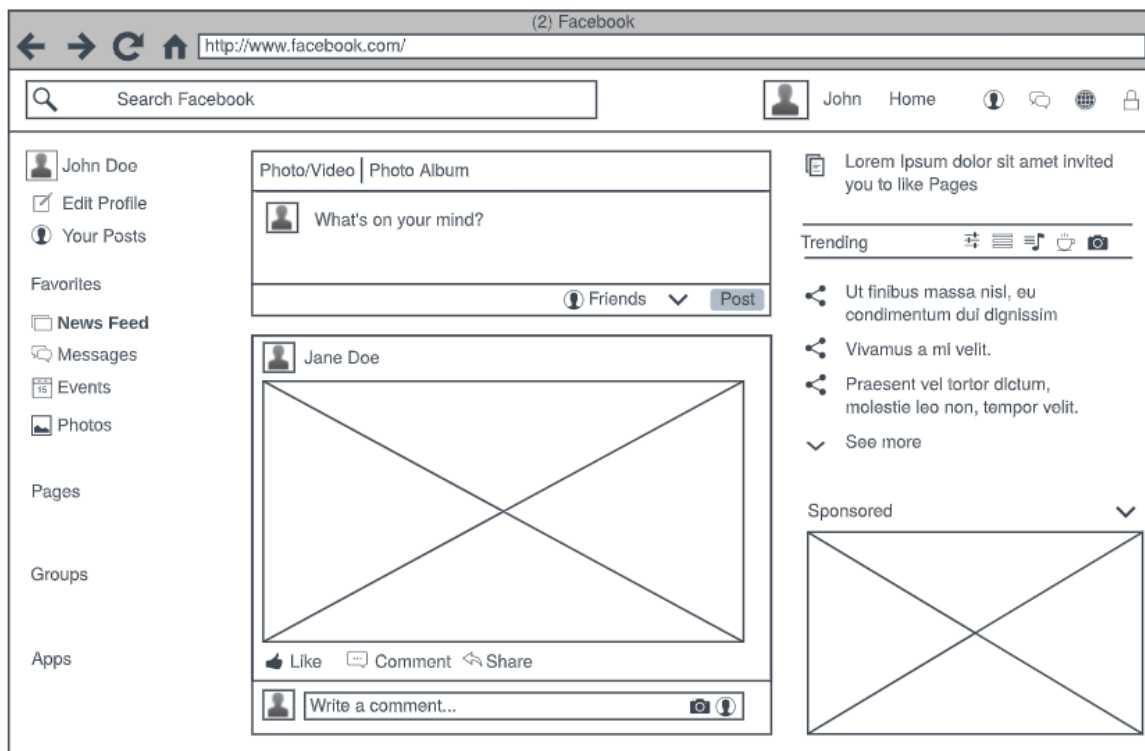
3.2.3 Layout a grafický návrh

Ještě před jakýmkoliv řešením barev, typografie, a dalších grafických detailů je dobré si rozvrhnout celkovou strukturu neboli layout. K tomu je vhodné používat drátové modely neboli wireframe. Drátový model webové stránky, či aplikace, známý také jako schéma stránky nebo schéma obrazovky, je vizuální průvodce, který představuje kostrovou strukturu webové stránky. Drátové snímky jsou vytvořeny za účelem uspořádání prvků tak, aby co nejlépe dosáhly konkrétního účelu. Účel je obvykle definován obchodním cílem a kreativním nápadem. Drátový model zobrazuje rozvržení stránky nebo uspořádání obsahu webových stránek, včetně prvků rozhraní a navigačních systémů, a jejich vzájemnou spolupráci (Konigi, 2009).

Jak již bylo zmíněno drátový model obvykle postrádá typografický styl, barvu nebo grafiku, protože hlavní zaměření spočívá ve funkčnosti, chování a prioritě obsahu. Drátové snímky mohou být kresby tužkou, náčrtky na tabuli, nebo mohou být produkovány prostřednictvím široké škály bezplatných nebo komerčních softwarových aplikací. Drátové rámce obvykle tvoří obchodní analytici, návrháři user experience, vývojáři, vizuální návrháři a ti, kteří mají zkušenosti s návrhem interakcí, informační architekturou a výzkumem uživatelů (Brown, 2011; Garrett 2010).

Drátový model webu spojuje základní koncepční strukturu nebo informační architekturu s povrchem nebo vizuálním designem webové stránky. Tyto modely pomáhají také stanovit funkčnost a vztahy mezi různými obrazovými šablonami webových stránek. Iterační proces, vytváření drátových modelů, je účinným způsobem, jak rychle vytvořit prototypy stránek, přičemž se pak měří praktičnost konceptu návrhu. Wireframing obvykle začíná mezi strukturálními pracemi – jako jsou například vývojové diagramy nebo mapy stránek. V rámci procesu vytváření webových stránek může wireframing být místem, kde se nápady a jednotlivé prvky struktury stávají hmatatelnými (Wodtke, 2009).

Kromě webových stránek se drátové modely používají k prototypování mobilních webů, počítačových aplikací nebo jiných produktů založených na obrazovkách, takže nabízí velké spektrum využití.

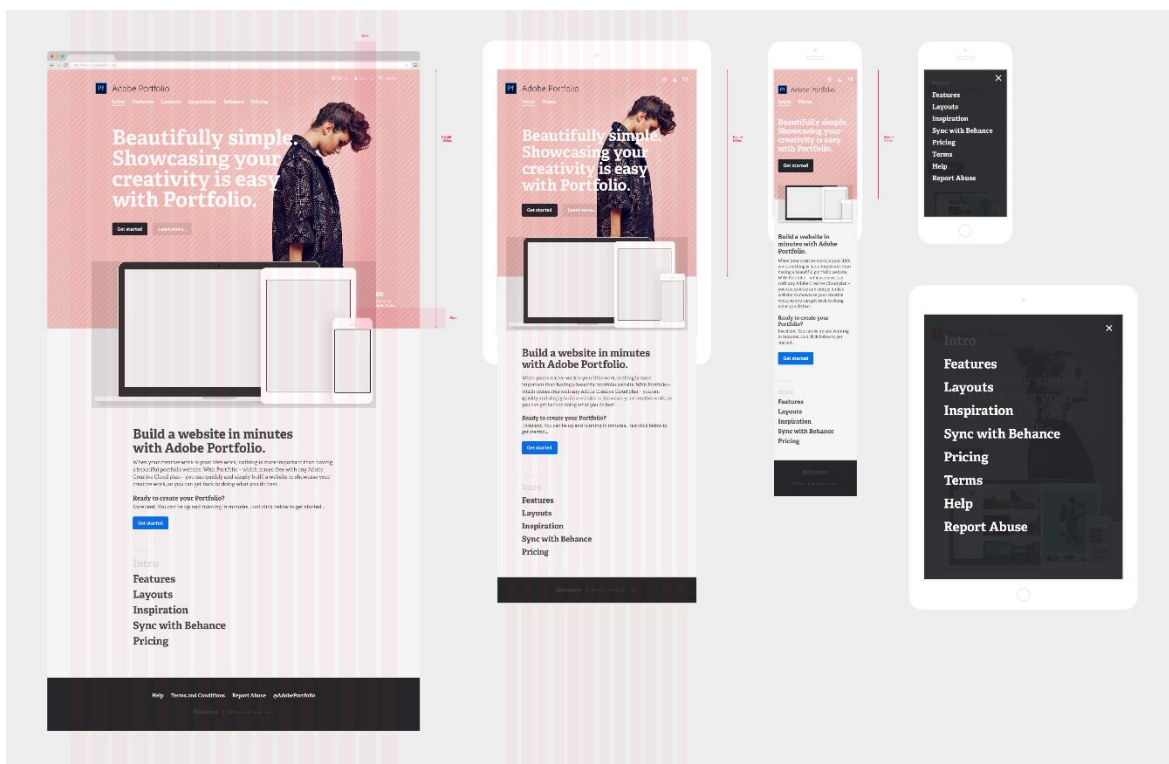


Obrázek 7 - Ilustrační wireframe sociální sítě Facebook

Zdroj: Lucidchart.com

Grafický návrh webového designu lze následně provádět v nejrůznějších grafických programech, které ať již se přímo specializují jen na web, nebo v komplexních grafických editorech, jako může být například Adobe Photoshop. Důležitým faktem je, že se již řeší reálné grafické komponenty a prvky, které budou na webu použity. Správně by měl být grafický návrh vytvořen na všechny typy zařízení, respektive tedy ke každému zlomovému bodu. Ve výsledku se může skládat návrh, například jen hlavní strany, z pěti různých sestav návrhu. Stejně tak se v nějakém zlomovém bodu některé prvky mohou skrýt a uživatel je následně zobrazí vytvořením nějaké akce, například klepnutím na další prvek. Příkladem může být navigace, tudíž i pro takovéto prvky by měl být návrh vytvořen (Couldwell, 2018; Mozilla, 2018).

Čím komplexnější webová stránka, nebo aplikace, tím více detailů je třeba do návrhu přenést a pak jen záleží na specifikacích, zda i další podstránky mají být designovány.



Obrázek 8 - Grafický návrh responzivního webu

Zdroj: Adobe Portfolio marketing website

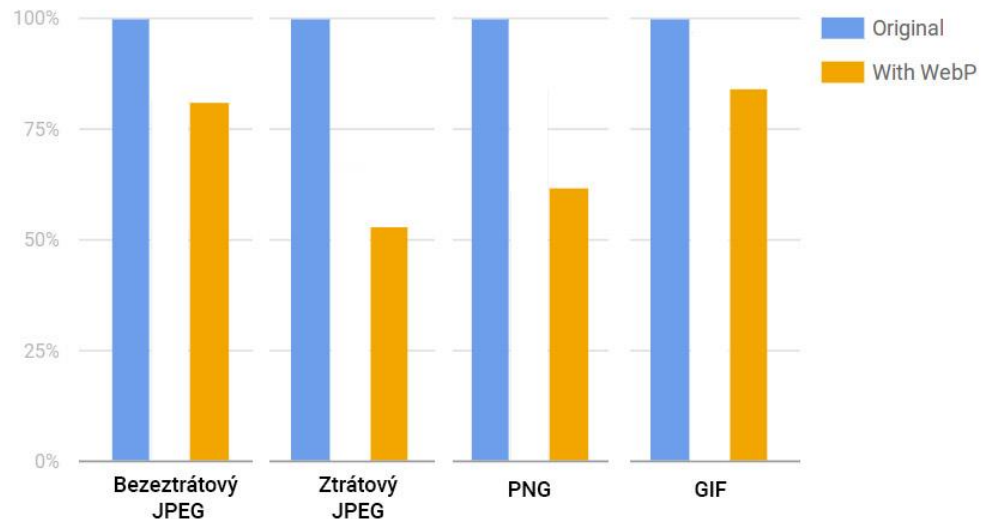
3.2.4 Obrázky

Při hotovém grafickém návrhu začne proces převádění obsahu do webové podoby, tzv. kódování. To s sebou nese i určitá opatření týkající se obrázků, kromě toho, aby byly samozřejmě ideálně responzivní.

Textový obsah není problém, protože textová pole nejsou rastrová, ale obrázek může být a může na něm být patrný rastr, pokud začne měnit své přirozené rozměry. Kromě toho je v dnešní době k dispozici mnohem více technologií a typů obrázků, než jsou základní BMP, JPG, GIF a PNG. Při výběru správného typu také záleží na tom, jaká cílová zařízení je třeba podporovat, jak složitá musí být grafika a kolik interaktivity grafika vyžaduje. Aby toho nebylo málo, je třeba si uvědomit, že každý druh obrázku může mít odlišnou kvalitu. Zejména pak, pokud by měl být web optimální, nesmí se dlouho načítat a tím jde ruku v ruce velikost načítaných souborů. V neposlední řadě mohou uživatelé na web přistupovat přes mobilní data, která jsou závislá na kvalitě signálu, a často jsou omezeny spotřebou množství dat (Mozilla, 2018).

Prvním úkolem tedy bude vybrat správný formát pro daný obrázek. Níže je seznam základních:

- JPG je základní formát obrázků, který byl publikován v roce 1992 a umí s ním pracovat každý prohlížeč. Jedná se o prostý obrázkový formát, nejsou v něm možné animace, či průhlednost.
- PNG je také základní formát obrázků z roku 1996, který však již umí pracovat s průhledností. Také si však uchovává k danému pixelu tabulku pro jednotlivé barvy a je často velikostně náročný.
- GIF je starší ještě než oba výše uvedené, publikován byl roku 1987 a největší jeho předností je malá velikost a možné animace. Nevýhodou pak může být, že si umí uchovat informace pouze o 256 barvách v daném obrázku.
- SVG je vektorový formát, oproti uvedeným rastrovým využívá tedy čar, křivek namísto pixelů. Formát je řešen formou XML, kde má uložené informace o daných křivkách a prohlížeč jej na základě toho následně vykreslí. Tyto obrázky mohou generovat přechody a jejich největší výhodou je, že je možné je libovolně zmenšovat, či zvětšovat bez ztráty kvality. Také mohou být animovány pomocí Javascriptu. Bohužel jsou však použitelné pouze na jednoduché obrazce a fotografie v tomto formátu tak nelze snadno vykreslit. Také vytvoření SVG obrázku je dost často náročnější, než u rastrového (Gremillion, 2015).
- Existují také formáty nové generace, příkladem může být formát WebP vyvíjený společností Google, který umí jak průhlednost, tak animace, a ještě k tomu nabízí možnost výrazného snížení datového objemu v porovnání s JPEG i PNG což může být v průměru 30% ušetřené velikosti. Velkou nevýhodou je slabá podpora prohlížečů, nicméně v současné době je toto technologický možné a pravděpodobně v budoucnu najdou takovéto formáty široké uplatnění (Google, 2018).



Obrázek 9 - Porovnání velikostí klasických obrázkových formátů s formátem WebP

Zdroj: Cloudflare.com

- Také je možnost si nechat vykreslit grafiku samotným prohlížečem. Ten umí na základě vlastností v kaskádových stylech, nebo díky HTML prvkům a atributům vytvořit nejrůznější grafické podoby, které mnohdy dokážou samotný obrázek nahradit. Příkladem mohou být barevné přechody, animace, geometrické útvary, stíny a tak dále (Gremillion, 2015).

Pokud je vybrán správný formát obrázku, je potřeba jej správně zobrazit na cílovém zařízení a zajistit tak, aby byl responzivní.

- Nejjednodušší a zároveň velmi praktické řešení, je pomocí CSS nastavit vlastnost maximální šířky 100% pro obrázky.

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

To má za následek, že pokud je šířka okna menší než daný obrázek, tak se šířka obrázku přizpůsobí velikosti okna a zároveň se zachovávají proporce – tím se tedy logicky i snižuje výška obrázku v daném poměru. Další výhodou je, že tuto CSS vlastnost podporují všechny prohlížeče.

- Další možností, kterou ovšem podporují jen moderní prohlížeče, je použití HTML prvku <picture>, který může obsahovat více zdrojů pro obrázek, zároveň s informací, kdy se má tento zdroj použít.


```
<picture alt="Popis obrázku">
  
  <source srcset="obrazek-vetsi.png" media="(min-width: 640px)">
  <source srcset="obrazek-mensi.png" media="(max-width: 639px)">
</picture>
```

V základu je načten obrazek.png, avšak při změně šířky okna se tento obrázek nahrazuje jím přiřazeným zdrojem.

- WebGL/Canvas umožňuje vytvořit plátno, ve kterém je možné vykreslit interaktivní grafiku pomocí prvku HTML `<canvas>`, a poté pomocí rozhraní Canvas API vytvořit různé tvary, čáry, importovat obrazové soubory, vytvářet text a mnoho dalšího. Nicméně tato varianta, vzhledem k rozsáhlosti a náročnosti na vytvoření obsahu, je dobrá spíše pro vizuální zobrazení her a komplexnějších vizualizací dat, tudíž u standardních prvků uživatelského rozhraní to není příliš efektivní. Kromě toho jsou výsledné obrazy pouze rastrové obrázky.



Obrázek 10 - Ilustrace responzivních obrázků

Zdroj: lynda.com

3.2.5 Navigace a jednotky

Web vlastníci více podstránek dost často obsahuje také navigaci s odkazy, které na ně přesměrují. Standardně bývají dva typy: horizontální a vertikální navigace, stránka však může mít také kombinaci obou. Ta se v případě dosažení určitého zlomového bodu většinou schová a uživatel ji zobrazí klepnutím na ikonu menu, dost často to bývají tři vodorovné čáry pod sebou, nazývané také jako „hamburger menu“.



Obrázek 11 - Záhleví webu na mobilním zařízení zobrazující hamburger menu

Zdroj: Getbootstrap.com

Při kódování responzivního designu lze použít jakékoliv jednotky. Nejjednodušší je jednotka pixelu, která je jednoznačně a absolutně určená hodnotou. Samozřejmě, může být přepočtená dle velikosti zařízení – typicky tomu tak bývá u mobilních zařízení, které mohou mít velká rozlišení. Při kódování pomocí pixelů však rozvržení bude omezující a nebude se dobře škálovat na velkém množství platform. Použitím relativních jednotek *ems* nebo *rems* však může mít vývojář velikost textu, která se vykreslí v konstantním poměru k velikosti obrazovky, nebo přesněji k velikosti bloku obsahu, v němž je obsažen (Michálek, 2018).

Jednotka	Jak počítá rozměr?
rem	relativně k velikosti písma na prvku <html>
em	relativně k velikosti písma na elementu
px	přepočtený pixel, CSS pixel
%	procenta relativně k rodičovskému elementu
vw	procento ze šířky okna prohlížeče
vh	procento z výšky okna prohlížeče

Tabulka 1 - Používané jednotky při tvorbě webu

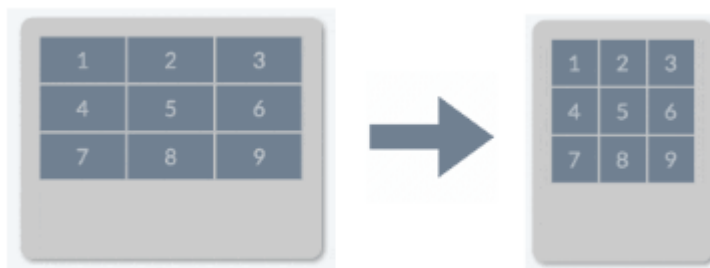
Zdroj: Vzhurudolu.cz

3.2.6 Tabulky

Tabulky, zejména pak pokud jsou velmi rozsáhle, s sebou mohou přinášet další problémy při tvorbě responzivního designu. Existuje mnoho způsobů, jak technicky vytvořit responzivní tabulku a níže jsou 4 příkladné způsoby, jak s tabulkami pracovat.

- Tabulce se nastaví maximální šířka 100% `max-width: 100%`, podobně jako je tomu u obrázků. Toto řešení je vhodné, pokud tabulka nemá moc sloupců,

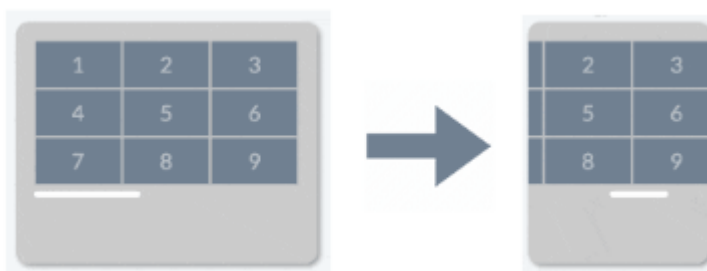
v opačném případě je možné nedůležité sloupce skrýt, nebo nastavit zobrazení po určité akci.



Obrázek 12 - První způsob tvorby responzivní tabulky

Zdroj: Vlastní zpracování

- Tabulka se obalí do prvku, ideálně `<div>` a tomu se nastaví CSS vlastnost `overflow`, přičemž si tabulka zachová původní obsah a uživatel je nucen posouvat tabulku po horizontální ose. Jedná se o jednoduché a efektní řešení, kdy však na první pohled nelze vidět všechna data pro daný řádek.



Obrázek 13 - Druhý způsob tvorby responzivní tabulky

Zdroj: Vlastní zpracování

- Tabulka zobrazí jednotlivé řádky v jednoduchých jednosloupcových tabulkách. Jsou tak vidět pohromadě veškerá data které jsou v daném řádku. Řešení je jednoduché a stačí zaměnit vlastnost u tabulky `display:table` na `display:block`.



Obrázek 14 - Třetí způsob tvorby responzivní tabulky

- Čtvrtý způsob je podobné řešení jako předchozí příklad, avšak se sloučí všechna sloupcová data. Po technické stránce je toto náročnější a nelze to provést jednoduchou záměnou vlastností, je třeba změnit celé HTML prvky, nebo pracovat s Javascriptem (Rizzo, 2017; Jarosz, 2018; W3C, 2015).



Obrázek 15 - Čtvrtý způsob tvorby responzivní tabulky

Zdroj: Vlastní zpracování

3.2.7 Javascriptové možnosti

JavaScript je jedním z nejpopulárnějších programovacích jazyků a spolu s HTML a CSS představují základní technologie, které stojí za webem. Vzhledem k tomu, že většina internetových prohlížečů má vestavěné jádro pro spouštění JavaScriptu, tak může fungovat na téměř každém zařízení. Responzivní design se dá také řešit pomocí JavaScriptu, metoda `window.matchMedia()` vrací objekt `MediaQueryList` představující výsledky zadaného řetězce dotazů na média CSS. Hodnota metody `matchMedia()` může být jakákoliv z funkcí vlastností CSS `@media`, jako je min-výška, min-šířka, orientace atd. Avšak media queries byly zavedeny v CSS3 a jsou jednou z klíčových složek responzivního webového designu. Dotazy na média se používají k určení šířky a výšky výřezu, aby webové stránky vypadaly dobře na všech zařízeních a přes kaskádové styly se k nim přistupuje snáze (W3C, 2018; Codeburst, 2018).

Front-endové frameworky často vlastní také JavaScriptové knihovny, které spolupracují s určitými prvky, zejména kvůli jejich atributům, kde se mohou nacházet informace právě pro tyto skripty.

3.2.7.1 JQuery

JQuery je knihovna JavaScriptu určená ke zjednodušení procházení stromu HTML DOM (Document Object Model) a manipulace s ním, jakož i zpracování událostí, animace CSS a Ajax. DOM neboli Document Object Model (objektový model dokumentu) je objektově

orientovaná reprezentace XML nebo HTML dokumentu. DOM je API umožňující přístup či modifikaci obsahu, struktury, nebo stylu dokumentu, či jeho částí. JQuery je software s otevřeným zdrojovým kódem využívající povolenou licenci MIT. Dle průzkumů serveru W3Techs (2019) jQuery používá 73 % z 10 milionů nejpopulárnějších webů. Analýza webu také ukazuje, že se jedná o nejrozšířenější knihovnu Javascriptu s velkým rozpětím, která má 3 až 4krát větší využití než jakákoliv jiná knihovna Javascriptu (Duckett, 2014).

Syntaxe jQuery je navržena tak, aby usnadnila navigaci v dokumentu, výběr prvků DOM, vytváření animací, zpracování událostí a práci s Ajaxem. Existují také modulární pluginy do knihovny jQuery a ty umožňují vytváření výkonných dynamických prvků pro weby a webové aplikace. Knihovna zjednodušuje syntaxi pro vyhledávání, výběr a manipulaci s prvky DOM. Například lze jednoduše nalézt prvek v dokumentu s určitou vlastností, nebo atributem (např. všechny obrázky s vybraným popiskem), a ten následně snadno změnit (např. pozičně, změnit viditelnost). Také se snadno vytváří události po nějaké reakci s určitým prvkem (např. kliknutí na element, nebo přejetí po něm) (Duckett, 2014).

Základním konceptem jQuery je použitím dolaru „\$“. Jedná se o jmenný prostor pro a tato funkce je potřeba například právě při vybírání elementů, nebo aplikování funkcí.

```
$(".vybrana_trida").hide(400);
```

Jak již bylo zmíněno, front-endové frameworky často využívají Javascript a také ve většině případů jsou tyto scripty napsány pomocí knihovny jQuery, tudíž bez importování knihovny jQuery budou využívané frameworkové Javascriptové knihovny nefunkční.

3.2.8 Rychlost webu

Rychlost webu, zejména jeho načtení závisí nejen na straně klienta, kterou vývojář stejně neovlivní – jako je například rychlost internetu, či výkon zařízení, ale také na straně serveru, respektive dat, které jsou zasílány návštěvníkovi. Lze tak ušetřit nejen velikosti souborů, ale hlavně čas (načtení webu), který může být v určitých marketingových krocích považován za jeden z nejdůležitějších.

Při používání knihoven a frameworků, které jsou často velmi komplexní, se vývojář může dostat do fáze, kdy nevyužije zdaleka veškeré komponenty, které se mu nabízejí v knihovnách a tím se tyto části stávají zbytečnými a zvětšují tak objem celkového webu. Frameworky mohou nabízet vlastní přizpůsobení vybráním obsahu komponent, které bude vývojář potřebovat, avšak i tak často v těchto balíčcích bývají funkce, které nejsou použity. V neposlední řadě

existují také softwary – pro kaskádové styly, které porovnají vytvořený web s importovanými knihovnamy a následně zredukují a vyřadí nepoužité vlastnosti pro prvky (Google, 2018).

Pro další zefektivnění načítání webu existuje souhrn rad a doporučení:

- Zobrazovat obrázky ve formátech nové generace a efektivním kódováním a se správnou velikostí společně s odložením načítání obrázků mimo obrazovku
- Snížit vliv kódu třetích stran
- Statické podklady zobrazovat s efektivními zásadami pro mezipaměť
- Předejít přenášení enormního množství dat
- Nepoužívat příliš velký model DOM
- Minifikovat kód CSS a Javascript
- Odstranit nepoužívané styly CSS
- K potřebným zdrojům se připojovat předem
- Zkrátit doby odezvy serveru (TTFB)
- Nepoužívat několik přesměrování stránky (Google, 2018)

3.2.8.1 Minifikace

Minifikace, nebo také občas používané pod pojmem minimalizace, je proces odstranění všech nepotřebných znaků ze zdrojových kódů interpretovaných programovacích jazyků nebo značkovacích jazyků beze změny jejich funkčnosti. Tyto zbytečné znaky obvykle zahrnují znaky mezer, nové řádkové znaky, komentáře a někdy blokují oddělovače, které se používají k přidání čitelnosti kódu, ale nemusí se provádět. Minifikace zmenšuje velikost zdrojového kódu a zefektivňuje a zrychluje jeho přenos. Na webu to následně může v praxi znamenat rychlejší načtení stránek (Crockford, 2011).

Nevýhodou minifikace je následná nepřehlednost v kódu, proto je dobré jakékoliv úpravy a změny provádět v originálním souboru a ten následně minifikovat a odkazovat přímo na něj. Minifikaci nelze zaměňovat za kompresi, lze ji odlišit od obecnější koncepce komprese dat v tom, že minifikovaný soubor může být interpretován okamžitě bez nutnosti kroku dekomprese, to znamená, že stejný interpret může pracovat jak s původním, tak se zmenšeným zdrojem (Crockford, 2011).

```
var array = [];  
for (var x = 0; x < 10; x++) {  
    array[x] = x;  
}
```

Výše uvedený kód je ekvivalentem kódů uvedených níže:

```
var array=[];for(var x=0;x<10;x++){array[x]=x;}
```

```
for(var a=[x=0];x<10;a[x]=x++);
```

Příčemž v druhém uvedeném kódu jsou pouze vynechány nepotřebné mezery a u třetího kódu je následně zkrácen zápis povolenou syntaxí. Minifikaci tak často používají rozsáhlé komplexní knihovny a frameworky.

3.3 Front-end frameworky

Front-endovým frameworkům se věnuje podstatná část vlastní práce, kdy je s jejich pomocí vyvíjena webová aplikace a jednotlivé používané frameworky jsou porovnávány a analyzovány.

Používání frameworků může velmi urychlit celkový vývoj webu, či webových aplikací. Celkově komplexní a kvalitní front-endové frameworky pomohu zlepšit celkovou funkčnost a také sjednotit a optimalizovat všechny prvky a komponenty webu. S rychlým vývojem a zdokonalováním webových technologií lze říct, že tyto technologie jsou aktuálně ve velmi pokročilém stavu a lze díky nim zastat takové (ať již designové, nebo funkční) věci, které by před několika lety bylo možné vytvořit jen stěží (Čápka, 2018).

Pro sjednocení veškerých funkčních a používaných prvků, které v základu nejsou responzivní, se využívají knihovny, které mají toto za úkol. Často také řeší plno problémů, které vývojáře mohou potkat a usnadní jim tak práci. Frameworky poskytují responzivní mřížky, které jsou většinou to nejdůležitější pro dané nástroje. Dále komponenty a šablony uživatelského rozhraní pro HTML a CSS, včetně typografie, formulářů, tlačítek, tabulek, navigace a dalších prvků rozhraní, stejně tak i volitelné funkce poskytované rozšířeními Javascriptu, nebo konfigurací potřebných komponent, které budou reálně využity (Moffitt, 2014).

Vzhledem k tomu, že neexistuje žádná komplexní studie a statistika využívání frameworků na internetu, budou dále popsány takové frameworky, které jsou dostupné zdarma a nejvíce rozšířené na webové službě GitHub. GitHub nabízí podporu vývoje jakéhokoliv softwaru za pomoci verzovacího nástroje Git. Zároveň však již nyní obsahuje a nabízí funkce sociálních sítí a lze vypořádat, které projekty uživatele zajímají a na základě toho budou popsány nejpoužívanější front-end frameworky (Github, 2018).

3.3.1 Bootstrap

Bootstrap vytvořili zaměstnanci firmy Twitter, kdy malá skupinka vývojářů, v čele s Markem Otto a Jacobem Thorntonem vyvíjeli framework pro interní účely. Zejména z důvodu konzistence zdrojových kódů – na již dřívější interní systémy byla používána celá řada knihoven a bylo tak velmi těžké celý tento systém udržovat. Při vývoji však zjistili, že vytváří něco mnohem podstatnějšího než jen interní nástroj. O několik měsíců později skončili s dřívější verzí Bootstrapu a používali jej jako metodu k dokumentaci a ke sdílení společných návrhových vzorů a aktiv v rámci společnosti. Po letech práce na Bootstrapu, kdy na ní pracovali během nocí a víkendů, vydali tento framework jako open source na konci srpna roku 2010. Od té doby se stal nejoblíbenějším projektem na GitHubu (Krause, 2016; Otto, 2012).

Níže je sepsán HTML kód, který importuje všechny důležité knihovny frameworku a vytváří tak prostředí pro následný vývoj za použití komponent Bootstrapu (verze 4.3.1).

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <title>Hello world!</title>
  </head>
  <body>
    <h1>Hello world!</h1>
    <script src=js/jquery.min.js><script>
    <script src="js/popper.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

3.3.1.1 Mřížkový systém

Mřížkový systém neboli „Grid system“ je pravděpodobně to největší devízou frameworku Bootstrap a defacto všech front-end frameworků. Ten funguje v součinnosti se zlomovými body pro danou šířku okna v zařízení. V tomto systému lze přiřazením třídy elementu říct, jaký

zlomek šířky má zabírat a hlavně, kdy dochází ke zlomu, tj. kdy bude mít daný element plnou šířku.

V základu má mřížka 12 sloupců na šířku okna, pokud však vývojář nechce používat všech 12 sloupců jednotlivě, můžete je seskupit dohromady a vytvořit širší sloupce.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Tím, že Bootstrap (ve verzi 4) již používá flexbox, není nutné vždy skládat obsah do 12 sloupců. A samozřejmě každý sloupcovitý element musí obsahovat informaci, kdy má bod zlomu, popřípadě, jaký zlomek šířky rodičovského elementu zabírá. Proto je v Bootstrapu připraveno 5 základních tříd, které tyto informace drží:

- Třída `.col-[číslo]` (extra malé zařízení - šířka obrazovky menší než 576px – neprovádí se zlom)
- Třída `.col-sm-[číslo]` (malé zařízení - šířka obrazovky větší než 577px)
- Třída `.col-md-[číslo]` (střední zařízení - šířka obrazovky větší než 768px)
- Třída `.col-lg-[číslo]` (velké zařízení - šířka obrazovky větší než 992px)
- Třída `.col-xl-[číslo]` (extra velké zařízení - šířka obrazovky větší než 1200px)

Na konci za pomlčkou, kde je zkratka velikosti zařízení se může doplnit tedy zlomek šířky 12 sloupcové mřížky, avšak není to podmínkou, vzhledem k tomu (jak již bylo zmiňováno) je použit flexbox, a ten by si jej sám dopočítal. Takže taková třída může vypadat i následovně: `<div class="col"></div>`. Třídy mezi sebou mohou být i kombinovány a tím se vytvoří mnohem dynamičtější a flexibilnější rozložení obsahu na webu. Správně by ještě tyto elementy měly mít rodičovský prvek, který definuje řádek. Ve výsledku tento „grid“ dělá opravdovou mřížku a pro příklad, pokud by se měla vytvořit jednořádková tří sloupcová mřížka, bude kód následující:

```
<div class="container">
  <div class="row">
    <div class="col-sm-3">a</div>
    <div class="col-sm-3">b</div>
    <div class="col-sm-3">c</div>
  </div>
</div>
```

Ve výše uvedeném příkladu je použita třída `.row`, která definuje daný řádek pro použité sloupce, nicméně i tato třída by měla mít rodičovský prvek, který definuje šířku okna v zařízení. Zejména u větších zařízeních s vysokým rozlišením by nemuselo být chtěné začínat obsah ihned z kraje okna. Proto existují dvě třídy `.container` a `.container-fluid`, které identifikují šířku obsahu. První zmíněná třída má různou šířku pro dané zlomové body a obsah se následně centruje na střed. Druhá zmíněná třída využívá 100% šířky v okně prohlížeče (Bootstrap.com, 2019).

3.3.1.2 Bootstrap 3 vs Bootstrap 4

Ze všech verzí Bootstrapu je verze 3, která byla oficiálně vydána v první stabilní verzi na konci září roku 2013, zatím ta nejdéle používaná a také nejvíce rozšířená. To se může samozřejmě změnit, přičemž verze 4 jakožto nástupce byla vydána v první stabilní verzi začátkem roku 2018. Jak může být patrné, novější verze využívá novějších technologií, například vlastnosti kaskádových stylů verze 3 (CSS3), či HTML5 elementy nebo atributy. Tím samozřejmě následně odpadá podpora starších prohlížečů, jako například Internet Explorer verze 9 a nižší (Otto, 2013).

Největší změnou, jakožto i nejdůležitější komponenta frameworku je v mřížkovém systému, který již byl popisován. Ve verzi 4 je používána vlastnost `flexbox` a tato užitečná vlastnost umožňuje změnit velikost sloupců mřížky, spravovat zarovnání rozvržení, procházet součásti a snadno měnit i komplexní sestavení rozvržení. Verze 3 používá vlastnosti `float`, která není tak flexibilní, a musí se u něj také více ošetřovat řádky, protože tato funkce pracuje na bázi obtékání. Novější verze také používá o jeden zlomový bod navíc a také mají lehce odlišné vlastnosti (Otto, 2018).

U verzí byla provedena aktualizace zdrojového souboru CSS z LESS na SASS. SASS je nepochybně mnohem silnější ve srovnání s LESS a má dokonce další rozšiřující užitečné funkce, jako jsou vnořené mediální dotazy, jednotky `rem` a `em`, mixování, smyčky, logické operátory a další. Je pravda, že LESS je mnohem jednodušší se učit, což znamená, že uživatelé,

kteří jsou ve světě webového designu a open-source frameworků nováčky, budou potřebovat delší čas, aby se s SASS naučili pracovat. Existuje však také spousta možností pro kompilaci SAAS do CSS (Bootstrap.com, 2019).

Dalšími, spíše drobnými změnami, jsou poupravení názvů u tříd k jednotlivým komponentám, změna základní velikosti písma na 16px a využívání relativních jednotek *rem*, namísto *px* (pixelů). Verze 4 přinesla spíše nové funkce, avšak jedna komponenta odpadla, a tím jsou glyphicons, což byl v základu importovaný soubor písem, který však obsahoval sadu ikon, které se snadno dají použít. Vývojáři namísto toho doporučují importování podobné sady FontAwesome (Bootstrap.com, 2019).

3.3.2 Foundation

Foundation je framework vyvíjený firmou Zurb, což je společnost pro design a návrhy produktů. Firma pomáhá společností navrhovat digitální produkty jako jsou například software, mobilní aplikace, weby a integrované služby (Zmijewski, 2019).

První verzi svého frameworku firma vydala na konci roku 2011 a nyní je nejnovější verze 6, která ještě nevyužívá plně nejnovějších HTML a CSS standardů a podporují jej i starší prohlížeče, například Internet Explorer 9, respektive si vývojář může vybrat, jak danou komponentu vyřešit napříč kompatibilitou. Vývojářský tým však začal pracovat na další verzi Foundation 7, která s největší pravděpodobností zruší podporu starších prohlížečů a plně implementuje novější technologie, jako je flexbox nebo možná automaticky počítaný mřížkový systém (Foundation.zurb.com, 2019).

```
<!doctype html>
<html class="no-js">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello world!</title>
    <link rel="stylesheet" href="css/foundation.css">
  </head>
  <body>
    <h1>Hello world!</h1>
    <script src="js/vendor/jquery.js"></script>
    <script src="js/vendor/what-input.js"></script>
```

```
<script src="js/vendor/foundation.min.js"></script>
<script> $(document).foundation(); </script>
</body>
</html>
```

3.3.2.1 Mřížkový systém

Jak již bylo zmiňováno, Foundation nabízí alternativy u některých komponent a je tomu například i u mřížkového systému. V základu má 3 typy mřížkových systémů, prvním je systém Float Grid, podporující starší prohlížeče a využívá vlastnost `float`. Druhým je Flex Grid, využívající vlastnost `flexbox`. Zmíněné dva systémy jsou v základu vypnuté a aktivován je třetí, názvem XY Grid. Ten funguje velmi podobně jako standardní plovoucí mřížka (Float Grid), ale obsahuje řadu užitečných funkcí, které jsou možné pouze u flexboxu, jako je například vodorovné a svislé zarovnání, automatické přizpůsobení velikosti a celá svislá mřížka – proto pojmenování dle os XY (Grid) (Foundation.zurb.com, 2019).

Aby byla možnost opět plně využít mřížku, je třeba mít zlomové body. U Foundationu jsou v základu následující tři zlomové body, které se značí následujícími třídami:

- Třída `.small-[číslo]` (extra malé zařízení - šířka obrazovky menší než 640px)
- Třída `.medium-[číslo]` (malé zařízení - šířka obrazovky větší než 641px)
- Třída `.large-[číslo]` (střední zařízení - šířka obrazovky větší než 1024px)

I zde je možné využít maximálně 12 sloupcové mřížky, avšak nestačí jen použít název zlomového bodu jako třídu, ale je nutné přidat další třídu `.cell`. Pro případné použití buňky bez zlomu se jednoduše nepoužije žádná třída zlomu. Takže výsledná buňka může být například ve tvaru `<div class="cell small"></div>` (se zlomovým bodem) nebo `<div class="cell"></div>` (bez zlomového bodu).

Pokud však při vývoji nebudou tyto tři zlomové body stačit, je možné si další přidat, popřípadě stávající upravit v případě využití SASS verze.

```
$breakpoints: (
  small: 0px,
  medium: 640px,
  large: 1024px,
  xlarge: 1200px,
```

```
xxlarge: 1440px,  
);
```

Jako u zmiňovaného Bootstrapu i zde se musí definovat řádky. Ty se značí třídou `.grid-x`, která musí být rodičovským elementem jednotlivých sloupcových buněk. A zároveň i tento prvek by měl být obalen elementem určující šířku. O to se stará třída `.grid-container`, která má v základu 1200px, avšak přidáním další třídy `.fluid` se šířka roztáhne na 100%.

```
<div class="grid-container">  
  <div class="grid-x">  
    <div class="cell small-3">a</div>  
    <div class="cell small-3">b</div>  
    <div class="cell small-3">c</div>  
  </div>  
</div>
```

3.3.3 PureCSS

Pure CSS je odlehčený a responzivní framework vytvořen firmou Yahoo v roce 2014. Prezentuje se jako velmi jednoduchý nástroj, který není velikostně rozsáhlý, avšak nabízí plno funkcí a je možnost díky němu vytvořit flexibilní layouty (Purecss.io, 2019).

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <meta name="viewport" content="width=device-width, initial-  
scale=1.0">  
    <title>Hello world!</title>  
    <link rel="stylesheet" href="css/pure-min.css">  
  </head>  
  <body>  
  </body>  
</html>
```

3.3.3.1 Mřížkový systém

Mřížkový systém je u PureCSS v porovnání s ostatními frameworky netradiční. Nabízí možnost si vytvořit až 24 sloupcovou mřížku, která umožní vytvořit velmi rozsáhlé a flexibilní

možnosti. V základu nástroj nemá žádnou kontejnerovou třídu a vývojář si tak případnou šířku obsahu na vyšším rozlišení musí navolit sám – nejčastěji vlastností `max-width`.

Mřížka se aktivuje třídou `.pure-g`, do které se následně vnoří elementy s třídami `.pure-u` a následně se volí velikost sloupce a jaká jeho část je zabráná. Syntaxe je `.pure-u-[číslo - zabraná část sloupce]-[číslo - velikost sloupce]`. Například `.pure-u-1-3` zabere jeden ze tří sloupců. Třída `.pure-5-24` zabere pět částí z 24 sloupcové mřížky. Možnost je následně tyto obaly kombinovat se zlomovými body.

Níže jsou popsány zlomové body za použití pěti tříd:

- Třída `.pure-u-[číslo]` (mobilní zařízení - šířka obrazovky menší než 567px)
- Třída `.pure-u-sm-[číslo]` (mobilní zařízení - šířka obrazovky větší než 568px)
- Třída `.pure-u-md-[číslo]` (tabletová zařízení - šířka obrazovky větší než 768px)
- Třída `.pure-u-lg-[číslo]` (desktopová zařízení - šířka obrazovky větší než 1024px)
- Třída `.pure-u-xl-[číslo]` (větší desktopová zařízení - šířka obrazovky větší než 1280px)

```
<div class="pure-g">
  <div class="pure-u-1 pure-u-sm-1-3">a</div>
  <div class="pure-u-1 pure-u-sm-1-3">b</div>
  <div class="pure-u-1 pure-u-sm-1-3">c</div>
</div>
```

3.3.4 Bulma

Framework Bulma vyvinul fron-endový designer, vývojář, učitel a autor knih Jeremy Thomas. Tento nástroj se vyznačuje svou funkcionalitou bez použití knihoven Javascriptu a je napsán čistě pomocí kaskádových stylů. Je lehký konfigurovatelný a je založen na flexboxu. Uzpůsoben je však tak, aby vývojář používající Bulmu nemusel znát veškeré zásady flexboxu a je postaven tak, aby byl co nejjednodušší (Thomas, 2018).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
```

```

    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Hello world!</title>
    <link type="text/css" rel="stylesheet" href="css/bulma.min.css">
</head>
<body>
</body>
</html>

```

3.3.4.1 Mřížkový systém

Mřížkový systém u Bulmy v základu má 12 sloupců a má značně rozšířené možnost práce s ní a se zlomovými body. Ještě před ní je vhodné vložit obsah do kontejneru a ten mění svou šířku podle toho, v jakém zlomovém bodu se nachází, značí se třídou `.container` a pro nastavení 100% šířky k ní stačí přidat další třídu `.is-fluid`. Pro obalení buněk mřížky se používá třída `.columns` a už zde je možné definovat zlomové body, které se pak aktivují pro všechny potomky, nebo je lze použít u jednotlivých buněk.

Zlomové body se definují pomocí pěti následujících tříd:

- Třída `.is-mobile` `.is-[číslo]` (mobilní zařízení - šířka obrazovky menší než 768px)
- Třída `.is-tablet` `.is-[číslo]` (tabletová zařízení - šířka obrazovky větší než 769px)
- Třída `.is-desktop` `.is-[číslo]` (desktopová zařízení - šířka obrazovky větší než 1024px)
- Třída `.is-widescreen` `.is-[číslo]` (větší desktopová zařízení - šířka obrazovky větší než 1216px)
- Třída `.is-fullhd` `.is-[číslo]` (větší desktopová zařízení - šířka obrazovky větší než 1408px)

Zbývá označit jednotlivé buňky pro sloupce a řádky mřížek a to třídou `.column`.

```

<div class="container">
  <div class="columns is-mobile">
    <div class="column is-4">a</div>
    <div class="column is-4">b</div>

```

```
<div class="column is-4">c</div>
</div>
</div>
```

3.3.5 Materialize

Materialize, nebo také MaterializeCSS je vyvíjen týmem studentů z Carnegie Mellon University. Framework je založen na bázi Material Designu, který je vytvořený a navržený společností Google. Jedná se o design, který kombinuje klasické principy úspěšného designu spolu s inovacemi a technologiemi. Cílem je vyvinout designovou strukturu a návrh, který umožňuje jednotné uživatelské prostředí napříč všemi jejich produkty na jakékoli platformě (Wang 2019).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Hello world!</title>
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
    <link type="text/css" rel="stylesheet"
href="css/materialize.min.css">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>

    <script type="text/javascript" src="js/materialize.min.js"></script>
  </body>
</html>
```

3.3.5.1 Mřížkový systém

Mřížka má v základu 12 sloupců a je možné je různě kombinovat. Rozdílem oproti jiným nástrojům je, že Materialize využívá procentuální šířku hlavního kontejneru, který se značí třídou `.container`. Ten má dynamický rozměr dle toho, v jakém zlomovém bodu se nachází. Dále se jednotlivé sloupcové buňky řadí do řádků, které se označí třídou `.row` a je to rodičovský

prvek buněk a zároveň potomek elementu s kontejnerem, který však není povinný v případě, že jej vývojář nechce použít a mít obsah roztáhlý na celé šířce okna.

Níže jsou popsány zlomové body za použití čtyř tříd:

- Třída `.s[číslo]` (mobilní zařízení - šířka obrazovky menší než 600px a šířka kontejneru je 70%)
- Třída `.m[číslo]` (tabletová zařízení - šířka obrazovky větší než 601px a šířka kontejneru je 85%)
- Třída `.l[číslo]` (desktopová zařízení - šířka obrazovky větší než 992px a šířka kontejneru je 90%)
- Třída `.xl[číslo]` (větší desktopová zařízení - šířka obrazovky větší než 1200px a šířka kontejneru je 90%)

Zbývá označit jednotlivé buňky pro sloupce a řádky mřížek a to třídou `.col`.

```
<div class="container">
  <div class="row">
    <div class="col s3">a</div>
    <div class="col s3">b</div>
    <div class="col s3">c</div>
  </div>
</div>
```

3.4 Vícekriteriální analýza variant (VAV)

Popisované front-endové frameworky budou v praktické části porovnávány a analyzovány. Tato rešeršní část popisuje aplikované metody použité analýzy.

Pro posuzování a porovnávání variant podle více kritérií se používají modely vícekriteriální analýzy variant, které zobrazují tyto rozhodovací problémy a charakterizují téměř každou rozhodovací situaci. Právě vstup více kritérií do rozhodovacího procesu přináší konflikty vyplývající z obecné kontroverznosti kritérií. Varianty musí být pečlivě vybrány, stejně jako volba jednotlivých kritérií. Tato kritéria musí být zejména nezávislá a měla by pokrývat všechna hlediska výběru, ideálně v přehledném množství (Šubrt, 2011).

Účelem využití modelů vícekriteriální analýzy variant v diplomové práci je nalezení nejvhodnější a nejlepší varianty při zohlednění veškerých důležitých vlastností front-endových frameworků.

$$\begin{array}{c}
 \mathbf{a}_1 \\
 \mathbf{a}_2 \\
 \vdots \\
 \mathbf{a}_m
 \end{array}
 \begin{pmatrix}
 \mathbf{f}_1 & \mathbf{f}_2 & \dots & \mathbf{f}_n \\
 \mathbf{y}_{11} & \mathbf{y}_{12} & \dots & \mathbf{y}_{1n} \\
 \mathbf{y}_{21} & \mathbf{y}_{22} & \dots & \mathbf{y}_{2n} \\
 \dots & \dots & \dots & \dots \\
 \mathbf{y}_{m1} & \mathbf{y}_{m2} & \dots & \mathbf{y}_{mn}
 \end{pmatrix}$$

Obrázek 16 - Kriteriaální matice

Zdroj: Šubrt, 2011

Jak je patrné z obrázku č.16, v modelech vícekritériaální analýzy, hodnocení variant podle kritérií je uspořádáno v kriteriaální matici. Prvek y_{ij} vyjadřuje hodnocení i -té varianty podle j -tého kritéria. Celkově je v matici dána konečná množina variant m , které jsou ohodnoceny podle n kritérií.

Kritéria ještě mohou být rozdělena podle povahy a kvantifikovatelnosti. V případě, že se při rozhodování vychází z toho, že nejlepší varianty podle vybraného kritéria mají nejvyšší hodnoty, jsou to kritéria maximalizační. V opačném případě, kdy se hledá nejnižší hodnota, jsou tato kritéria minimalizační. Dle druhého rozdělení je možné mít kritéria kvantitativní, která tvoří objektivně měřitelné údaje. Naopak hodnoty, které se nedají změřit jsou následně subjektivně odhadnuté uživatelem a v těchto případech se například používají bodové stupnice. Tato kritéria se nazývají kvalitativní (Šubrt, 2011).

3.4.1 Saatyho metoda

Ohodnocené varianty dle kritérií však často jsou různého charakteru a často také nemají stejnou váhu. Saatyho metoda slouží právě k určení těchto vah za postupu párového porovnávání kritérií (Šubrt, 2011).

Pro toto ohodnocení se používá devítibodová stupnice popsaná níže:

- 1 – rovnocenná kritéria i a j
- 3 – slabě preferované kritérium i a j
- 5 – silně preferované kritérium i a j
- 7 – velmi silně preferované kritérium i a j
- 9 – absolutně preferované kritérium i a j

Je však možné používat i mezistupně (hodnoty 2, 4, 6, 8).

Párově porovnaná kritéria je nutné následně zapsat do reciproké matice čtvercového řádu $n \times n$. Platí tedy, že $s_{ij} = 1/s_{ji}$ a odhaduje podíl vah i -tého a j -tého kritéria. Diagonála obsahuje vždy hodnoty jedna, protože každé kritérium je samo sobě rovnocenné (Šubrt, 2011).

$$S = \begin{pmatrix} 1 & s_{12} & \dots & s_{1n} \\ 1/s_{12} & 1 & \dots & s_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ 1/s_{1k} & 1/s_{12} & \dots & 1 \end{pmatrix}$$

Obrázek 17 - Saatyho matice

Zdroj: Šubrt, 2011

Následně lze pro další operace odhadnuté váhy kvantifikovat. Nejčastěji se využívá postup výpočtu vah normalizovaného geometrického průměru řádků Saatyho matice. Hodnoty se vypočtou jako geometrický průměr b_i a následně se normalizují pomocí vzorce v_i (Houška, 2013).

$$b_i = \sqrt[n]{\prod_{j=1}^n s_{ij}} \quad v_i = \frac{b_i}{\sum_{i=1}^n b_i}$$

3.4.2 Metoda váženého součtu

Metoda váženého součtu je speciálním případem metody funkce užitku a konstruuje celkové hodnocení pro každou variantu, lze ji proto používat pro nalezení té nejvýhodnější varianty, stejně tak určit pořadí ostatních variant. Pro zjištění celkového užitku dané varianty je potřeba ještě znát ideální variantu H , která je ve všech kritériích naplněna těmi nejlepšími hodnotami s ohodnocením h_j a bazální variantu D , která naopak obsahuje ty nejhorší možné hodnoty s ohodnocením d_j . Následně je možné sestavit kritériální matici, variant a kritérií, kde se dílčí hodnoty vypočtou pomocí vzorce r_{ij} (Šubrt, 2011).

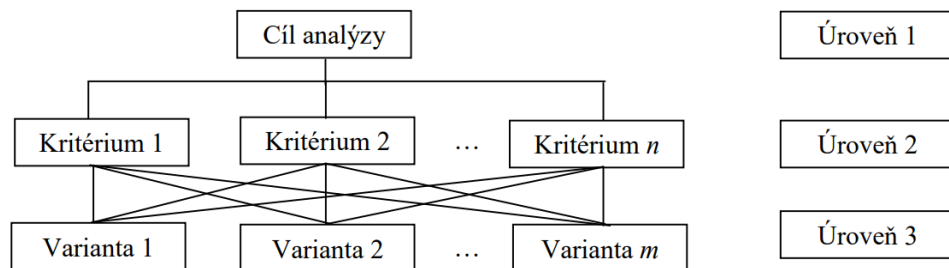
$$r_{ij} = \frac{y_{ij} - d_j}{h_j - d_j}$$

Po kompletaci matice stačí seřadit vzestupně hodnoty vypočtené pomocí vzorce $u(a_i)$.

$$u(a_i) = \sum_{j=1}^k v_j r_{ij}$$

3.4.3 Metoda AHP

Metoda AHP (Analytic Hierarchy Process) poskytuje rámec pro přípravu rozhodování ve složitých situacích a tento proces rozhodování se snaží urychlit a zjednodušit. Jedná se o metodu, která rozkládá nestrukturované situace na jednodušší komponenty a je díky tomu vytvořen hierarchický systém. Tuto metodu je možné použít pro jakýkoliv typ informace o preferenčních vztazích mezi komponentami modelu (Šubrt, 2011).



Obrázek 18 - Hierarchická struktura typické úlohy VAV

Zdroj: Šubrt, 2011

Na každé úrovni hierarchické struktury se použije Saatyho metoda kvantitativního párového porovnávání hodnot. Díky subjektivnímu hodnocení se pak přiřazuje jejich důležitost (váhy variant dle kritérií) a syntézou těchto hodnocení se pak stanoví ta komponenta, která má nejvyšší prioritu (Šubrt, 2011; Houška, 2013).

4 Vlastní práce

Vlastní práce je rozdělena do tří částí, které jsou rozčleněny do podkapitol. Nejdříve je vytvořena obsahově stejná webová aplikace pomocí všech vybraných front-endových frameworků. Výběr frameworků, kterých je dohromady 5, je popsán níže, a tudíž je vytvořeno 5 verzí webové aplikace. V druhé části jsou charakterizována vybraná kritéria a následně i frameworky hodnoceny dle těchto kritérií. Ve třetí části je provedena analýza dle popsaných metod v teoretické části.

Výběr jednotlivých frameworků je učiněn pomocí dostupných funkcí webové služby GitHub, která podporuje vývoj softwaru, jak je popsáno v teoretické části a stejně tak byly popsány i tyto nástroje. Lze tak identifikovat nástroje, které jsou populární a nejvyužívanější. Dále byl výběr konzultován s odborníky z praxe.

Níže je seznam vybraných nejpoužívanějších nástrojů, společně s jejich verzemi:

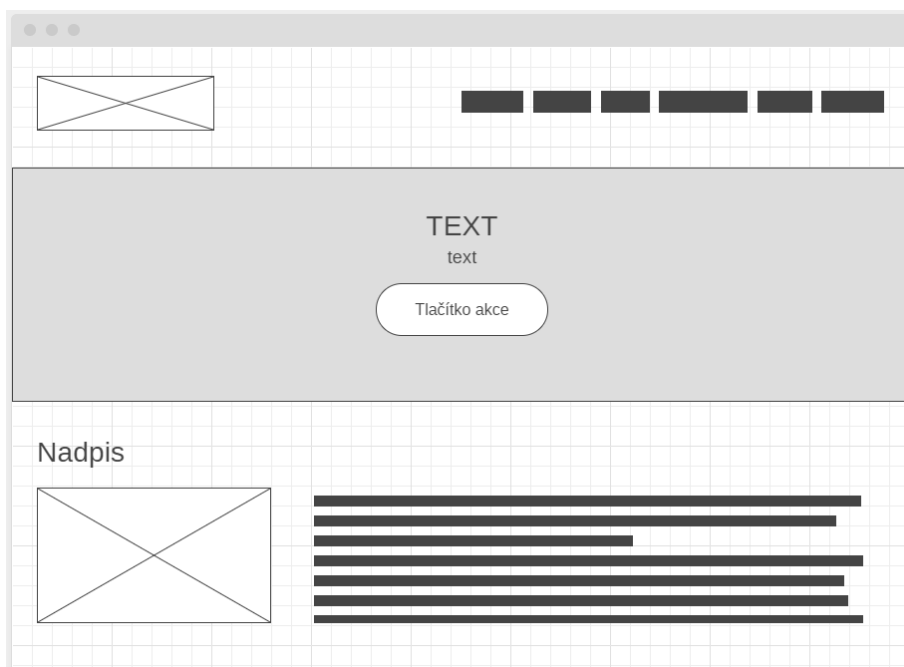
- Bootstrap (*verze 4.3.1*)
- Foundation (*verze 6.5.3*)
- PureCSS (*verze 1.0.1*)
- Bulma (*verze 0.8.0*)
- Materialize (*verze 1.0.0*)

4.1 Vývoj webové aplikace

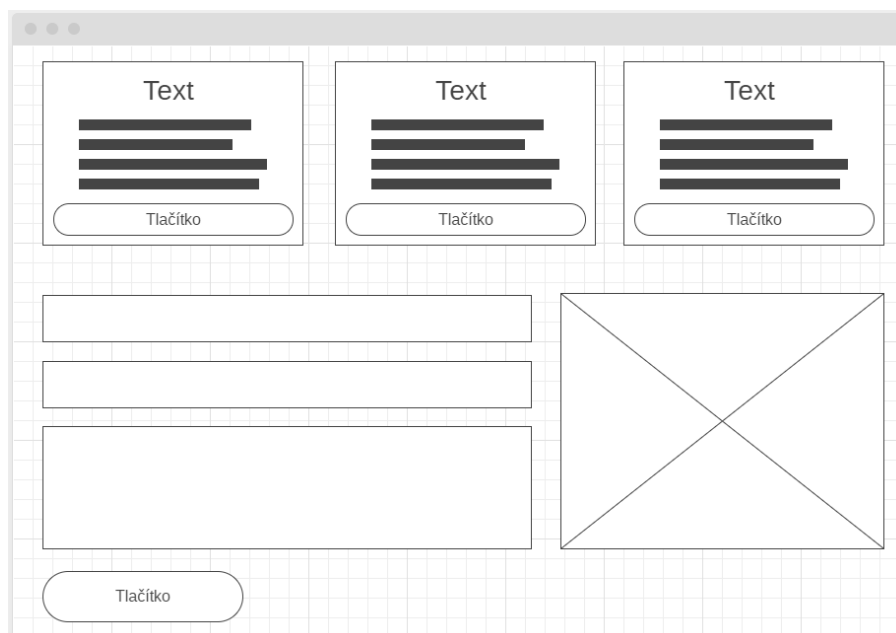
První část vlastní práce se skládá z popisu vytvářené webové aplikace, kde jsou aplikovány popisované nástroje (viz. literární řešerše, kapitola Front-end frameworky). Na základě tohoto vývoje si řešitel má možnost vyzkoušet nástroje a získat praktické poznatky implementovaných nástrojů, které jsou následně také aplikovány při volbě kritérií a jejich hodnocení v následné analýze. Verzí je celkově 5, každá reprezentuje framework, ve kterém je web vyvíjen.

Vytvářená webová aplikace obsahuje nejrůznější prvky, které nabízejí nástroje pro vývoj, aby co nejvíce obsadila spektrum komponent, které mohou být při vývoji různých webových aplikací a webu použity. Zároveň však také bude stále připomínat aktuálně provozované weby a webové aplikace. Obsahově je web na všech verzích stejný, stejně tak bude snaha o dodržení stejné struktury, avšak lze předpokládat, že některé prvky či komponenty nebudou ve všech nástrojích dostupné, proto budou alternativně upraveny, nebo vynechány. V praxi by bylo ideálním řešením implementací individuálního řešení, staženou dodatečnou knihovnou, funkcí, ale zde pro zachování a použití pouze porovnávaných nástrojů bude tato možnost nevyužita.

Struktura webu bude jednoduchá, rozdělená do horizontálních sekcí orientované pod sebou. Pro návrh této struktury je použito drátových modelů.



Obrázek 19 - Wireframe vytvářené webové aplikace č.1
Zdroj: Vlastní zpracování



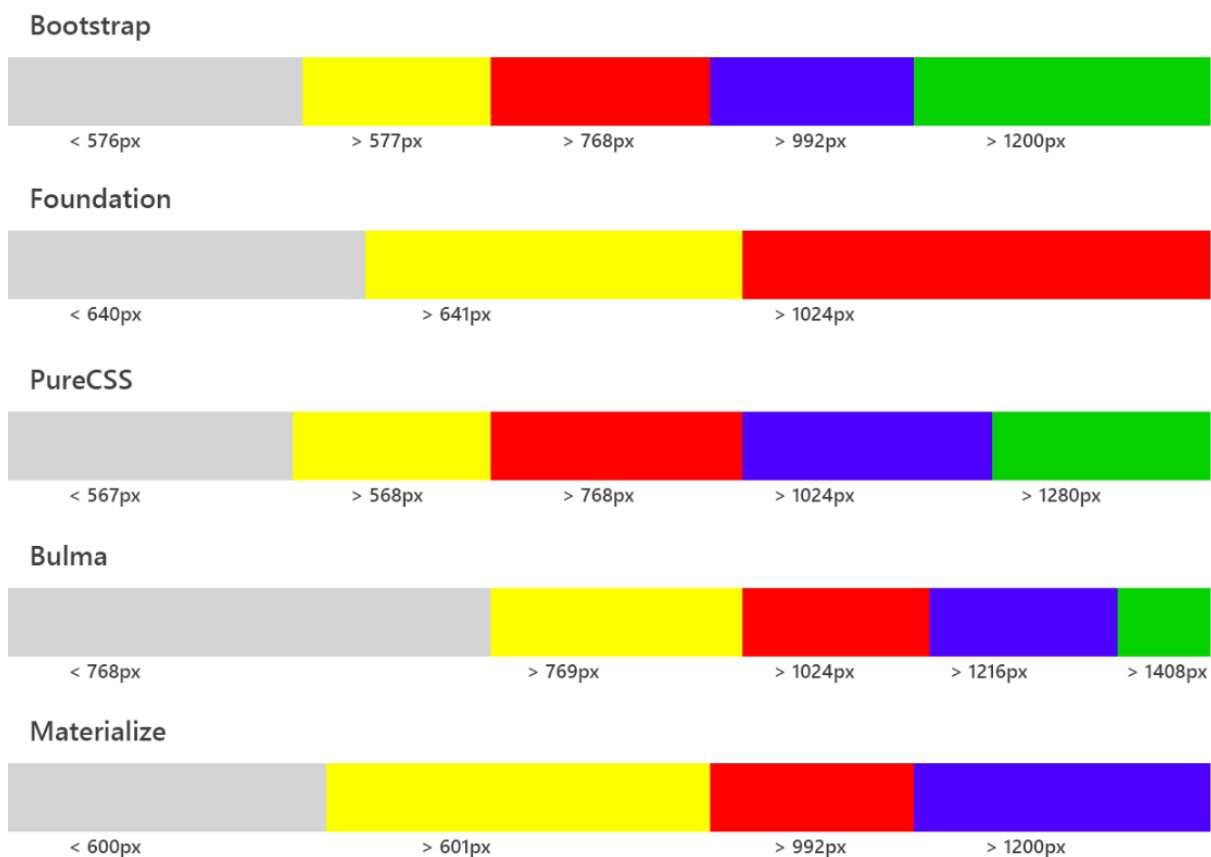
Obrázek 20 - Wireframe vytvářené webové aplikace č.2
Zdroj: Vlastní zpracování

Implementovány jsou všechny doporučované knihovny dle dokumentace vývojářů, to znamená i Javascriptové soubory, přičemž volání Javascriptových funkcí bude v základu provedeno opět pouze při doporučení, jako tomu je například u frameworku Foundation, viz kód níže. Externí kaskádové styly použity nejsou, pro globální stylizaci však u všech verzí jsou použity základní úpravy některých vlastností určitých elementů

```
<script src="js/vendor/jquery.js"></script>
<script src="js/vendor/what-input.js"></script>
<script src="js/vendor/foundation.min.js"></script>
<script> $(document).foundation(); </script>
```

Vyvíjené webové aplikace jsou často více komplexní, sofistikované a využívají dat jiných systémů, na které jsou napojeny, například pomocí API. Pro testovací účely není potřeba toto napojení provádět, pokud se testují front-entové frameworky. Ve výsledku by funkcionality rozhraní API nebyly řazeny mezi kritéria. Zároveň by bylo vhodné například výstupní data formátovat a pracovat s nimi pomocí nástrojů využívajících na webu či aplikaci. Nicméně pro alespoň částečnou kompenzaci a možnost dotazování se cizích serverů a simulaci určitého časového prodlení v komunikaci, je importována `iframe` mapa společnosti Google.

Při vývoji a zejména při řešení webu pro responzivní design jsou využívány základní nabízené zlomové body jednotlivých nástrojů. Nástroje umožňují i modifikaci a přidávání zlomových bodů, dle potřeby. Často je možné také plno vlastností editovat a pracovat s nimi pomocí preprocesorů, například SASS. Avšak při stažení a ve snadně implementované, již zkompileované knihovny, jsou základní zlomové body předpřipraveny. Proto budou brány pouze tyto body v potaz a následně mohou být také objektivně ohodnoceny.

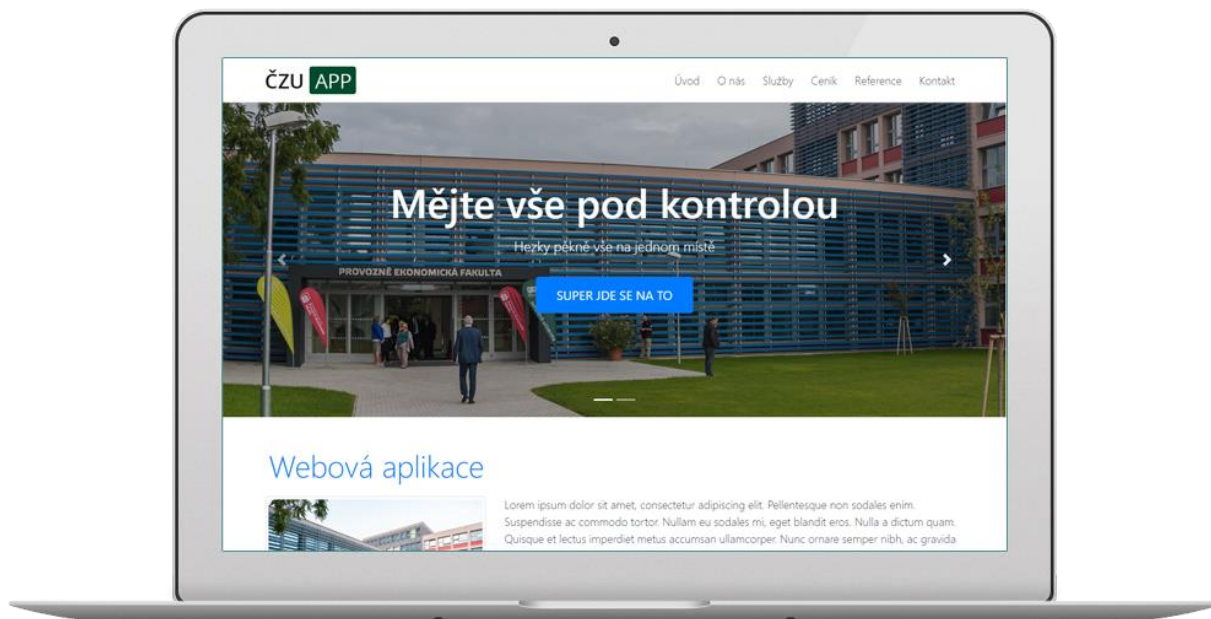


Obrázek 21 - Zlomové body v grafické podobě u vybraných frameworků
Zdroj: Vlastní zpracování

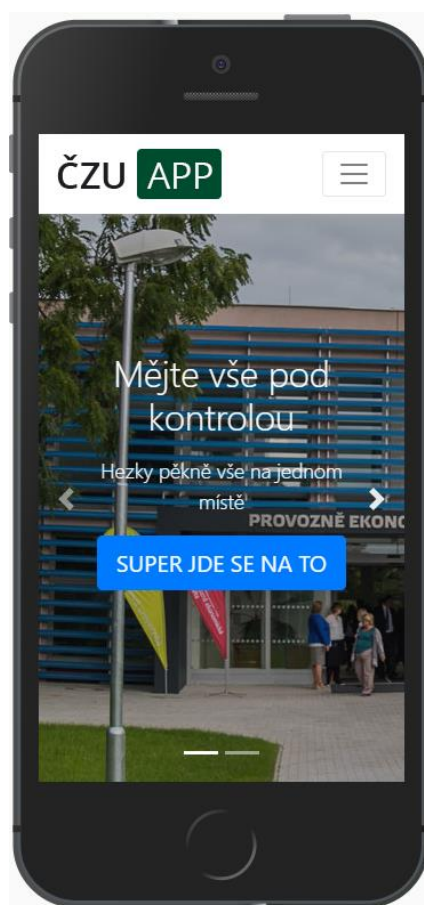
Absence některých komponent zkomplikovala vývoj. Respektive se musela hledat alternativní řešení pro zachování funkčnosti a příkladem je responzivní navigace, která v základu u nástroje PureCSS není řešena. Alternativa je použití jednoduché responzivní „scrollovací“ (posunovací) lišty. V případě jednostupňového menu je využití i reálné a zastává tak navigační funkci.



Obrázek 22 - Navigace řešená hamburger menu X scrollovací horizontální menu
Zdroj: vlastní zpracování



Obrázek 23 - Vytvořená webová aplikace - desktop verze
Zdroj: Vlastní zpracování



Obrázek 24 - Vytvořená webová aplikace - mobilní verze
Zdroj: Vlastní zpracování

4.2 Charakteristika hodnotících kritérií

V následném procesu rozhodování a porovnávání je klíčové určit si daná kritéria. Ty jsou nejdříve obecně popsána a následně aplikována pro každý framework v rámci podkapitol.

Výběr hodnotících kritérií je založen na základě získaných informací studiem odborných zdrojů týkající se responzivního designu, včetně její tvorby. V potaz jsou také brány důležité aspekty při používání webu což i bylo konzultováno ve volné diskusi s webovými profesionály. Vývojáři v komunitním fóru společnosti Google (Developers.google.com, 2020) upozorňují na důležitost velikosti a rychlosti načítání webu. Dále také, jak uvádí server Stackshare.io (2020) a také jeho komunita, nejdůležitější komponentou, zejména u front-endových frameworků, je samotný mřížkový systém a zlomové body. Následně to pak jsou komponenty, které jsou často technicky složitějšího rozsahu a prvky, které potřebují mnoho úprav, například navigace, nebo formuláře. Tyto informace také potvrzuje webový server určený pro vývojáře Geekflare.com (2020) a jsou i následně použity při hodnocení konkrétních nástrojů, zejména u kvalitativních kritérií, která jsou ohodnocena určitými body.

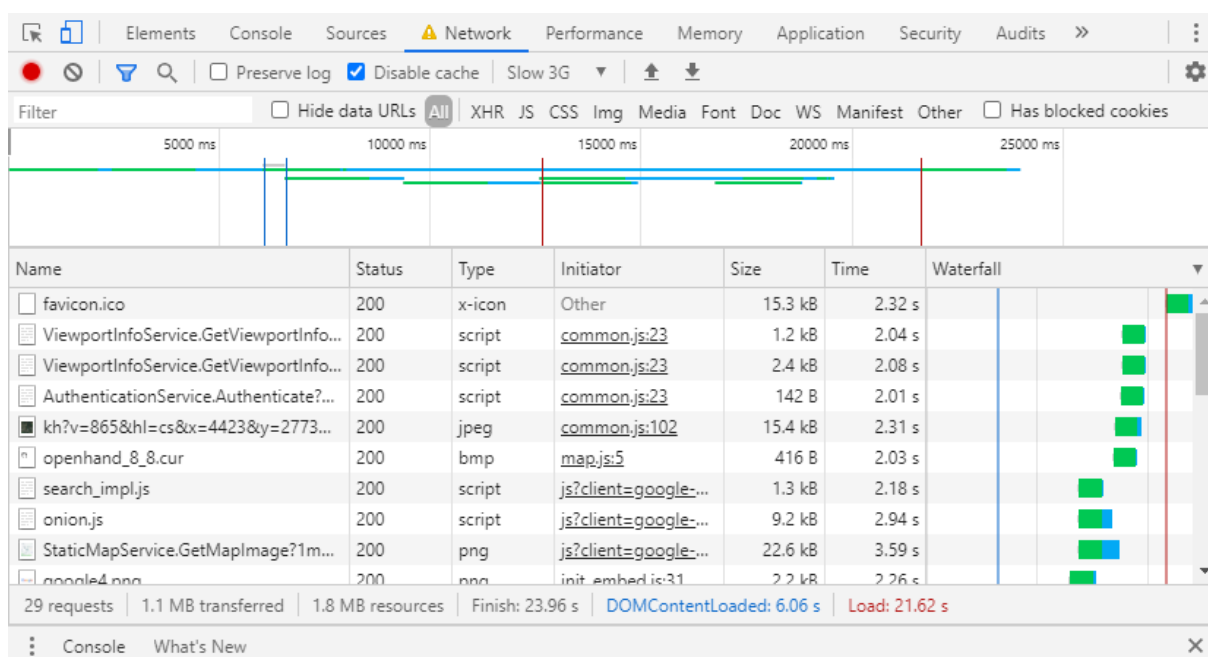
Kritéria jsou zohledněna pro vývojáře, kteří tak potřebují ideální komplexní nástroj pro tvorbu responzivního webu a webových aplikací. Jednotlivá kritéria zastupují ať již důležité komponenty, tak kvalitu a stav dokumentace, či podpory pro daný nástroj. Kritéria jsou sepsána níže:

Velikost

Vzhledem k tomu, že se jedná o knihovny, které se musí implementovat, tak tyto importované soubory obsahují velké množství zdrojového kódu. Zásadní věcí, která i nepřímou může naznačit množství možností, úprav, funkcí je právě velikost používaného frameworku. Pokud budou mít frameworkové části velký objem, naroste tak i celková velikost webu, či aplikace a tím se může prodloužit doba načtení webu, což je nežádoucí efekt. Také, vzhledem k tomu, že jsou knihovny zaměřeny na responzivní design, předpokládá se využívání webu na mobilním zařízení. Ta se v dnešní době umožňují připojit k internetu pomocí mobilních dat, které poskytuje mobilní operátor a často jsou uživatelé limitováni a omezeni v celkovém množství přenesených dat (FUP). Tím je také dobré návštěvníkům umožnit tuto úsporu a nezatěžovat je příliš vysokou velikostí. Velikost je součtem všech využívaných a doporučených knihoven, které celý framework poskytuje a zároveň se používají jen minified verze všech importovaných souborů.

Rychlost načtení

Měření rychlosti načtení webové aplikace bude probíhat pomocí vývojářských nástrojů Google Chrome DevTools. Tyto vývojářské nástroje umožňují měřit rychlost načtení webů společně se všemi důležitými ukazateli a zároveň umožňuje simulovat načtení webu pomocí různě kvalitního připojení. Měření bude prováděno za pomoci simulace horšího a méně kvalitního 3G připojení, které může vykazovat větší odchylky měření. Měřený údaj bude zaokrouhlen na desetiny a interpretován časovými jednotkami až po kompletně stažené a načtené struktuře webu, tudíž nebudou přítomny žádné styly, které by blokovali provádění Javascriptu.



Obrázek 25 - Měření rychlosti načtení webu pomocí Chrome DevTools

Zdroj: Vlastní zpracování

Dokumentace a implementace

Často vzhledem k rozsahu a možnostem využití frameworků je dobré zkontrolovat dokumentaci, kterou výrobce nástroje poskytuje. Ideální je, pokud se v ní nachází všechny důležité informace a návody, jak prvky používat. V bodovém hodnocení bude rozhodovat zejména kvalita, struktura a obsah dokumentace.

Implementace je často rozlišena prostředím a nástroji, s kterým vývojář pracuje. Standardně bývají nástroje rozděleny mezi zkompilovanými a ty, které si vývojář upraví a zkompiluje sám pomocí určitého preprocesoru.

Mřížkový systém a zlomové body

Mřížkový systém a následná možnost tvorby layoutu je často velmi důležitá komponenta. Díky dobře zpracovanému mřížkovému systému je následně možnost snadno mřížky kombinovat, umisťovat je a vytvářet tak složité struktury. Těchto systémů, jak s mřížkami pracovat je velké množství, respektive způsobů, jak ji vytvořit. Zohlednit se tak musí často kompatibilita a další technické možnosti. Často pak v kombinaci se zlomovými body jsou celé mřížkové systémy responzivní. Ty také mají za úkol zjednodušit práci se strukturou, přičemž není dobré mít mnoho či velmi málo těchto bodů.

Navigace

Navigační prvky a uzpůsobení celého panelu, často nazývaného jako menu, je důležitým aspektem webu a aplikací, pro přechody mezi podstránkami nebo sekcemi. Často se také řeší, aby celá navigace byla responzivní, protože navigačních odkazů může být velký počet, což na vyšším zobrazovacím zařízení nemusí být problém, ale na mobilních perifériích se tyto prvky nemusejí vejít. Proto se často tyto odkazy následně schovají pod nějaký navigační prvek, často to bývá „hamburger menu“. Navigace ale nemusí být jen jeden panel s těmito odkazujícími elementy, někdy je potřeba mít i sekundární menu, nebo navigace dílčích obsahových prvků, či panelů.

Typografické prvky a formuláře

Základní typografií může být doladění standardních textových prvků, které již v základu mají určité vlastnosti (například nadpisy, seznamy, fonty). Speciálnějšími prvky mohou být různé druhy tlačítek, značek, odznaků, nebo upozorňovacích oken.

Formuláře mohou být také vylepšeny než nativní vykreslené prohlížečem. Ty mají často svůj styl a na různých prohlížečích i systémech mohou být vzhledově odlišná. Mnoho textových polí však mohou mít pomocí HTML5 atributů již v základu vstupní ochrany, nebo jsou připraveny pro vstup určité očekávané hodnoty (například datum). Frameworky tedy mohou odlišné styly prohlížečů sloučit a nabídnout tak ucelené formulářové prvky a samozřejmě i něco navíc přidat. Standardně tak také vyřešit responzivitě.

Tabulky

Tabulky mohou také v základu působit neatraktivně, navíc není nijak řešen responzivní design tabulek. To také, jak bylo popsáno v teoretické části, může být často problémem, vyřešit

responzivní design u tabulek a dá se tak řešit více metodami. Počet a kvalita bude v hodnocení určující.

GUI a přizpůsobení

Použitím HTML elementů bez jakýchkoliv stylů je dopřáno prohlížeči vykreslit prvky dle jeho výchozího nastavení. Jedná se následně o obyčejné prvky, které nejsou nikterak graficky atraktivní. Frameworky mohou tyto odlišné styly prohlížečů sloučit a nabídnout tak ucelené grafické prostředí, které bez dalších úprav může být uživatelsky příjemné. Následně, pokud je definován seznam tříd, atributů, tak tato grafika může nabývat například různorodých barev. Připravena tak může být sada pro určité druhy akcí.

Javascriptové knihovny

Javascript může v mnoha ohledech prvkům pomoci, například s jejich dynamikou, či efekty. Celkově však pro základní funkčnost není zcela nezbytný, framework si dokáže v mnoha ohledech pomoci kaskádovými styly – především CSS3. Implementací knihovny se zvyšuje velikost načítané stránky a může se ještě navýšit při použití dalších nezbytných nástrojů, jako je jQuery. Bude se brát proto v úvahu, zda velikost knihovny je dostatečná v porovnání funkčních záležitostí, které nabízí.

Všechna kritéria jsou v souhrnné tabulce č.2. Kritérium 1 a Kritérium 2 jsou kvantitativního charakteru s minimalizační povahou. První tato dvě kritéria je tedy možné změřit a budou pro ně zvoleny vhodné jednotky pro jasnou charakteristiku a následnou analýzu a porovnávání. Zbylá kritéria jsou kvalitativního charakteru s maximalizační povahou. Nelze je nikterak změřit, proto budou bodově ohodnocena dle objektivního názoru řešitele, který získal při vývoji webové aplikace za využití testovaných frameworků praktické poznatky a může všechna vybraná kritéria objektivně hodnotit. Zejména bude brána v potaz komplexní stránka daných komponent, následně možnosti využití a kvalita.

Popisovaná kritéria jsou následně aplikována v rámci následujících podkapitol na každý front-endový framework, který byl použit.

	Kritérium	Jednotka
1	Velikost	kB
2	Rychlost načtení	sekundy
3	Dokumentace a implementace	bodové ohodnocení (0 – 10)
4	Mřížkový systém a zlomové body	
5	Navigace	
6	Typografické prvky a formuláře	
7	Tabulky	
8	GUI a přizpůsobení	
9	Javascriptové knihovny	
10	Podpora prohlížečů	

Tabulka 2 - Souhrn hodnotících kritérií

Zdroj: Vlastní zpracování

4.2.1 Hodnotící kritéria - Bootstrap

Bootstrap byl analyzován ve verzi 4.3.1, který byl v době analýzy nejaktuálnější stabilní verzí.

Velikost

14 kB (*index.html*) + 153 kB (*bootstrap.min.css*) + 87 kB (*jquery.min.js*) + 21 kB (*popper.min.js*) + 57 kB (*bootstrap.min.js*) = 332 kB

Rychlost načtení

Rychlost načtení: 6,1 s

Dokumentace a implementace

Dokumentace obsahuje velké množství informací ať již ohledně nastavení samotného frameworku, tak i obsažených komponent, kde jsou podrobně popsány jejich funkce, využití, společně s praktickými ukázkami kódu a ilustračním zobrazením. Vzhledem k rozsahu nástroje, je dokumentace rozsáhlá, rozdělena do mnoha rozřazených kapitol.

Implementovat je možné Bootstrap ručně, pomocí stažení již zkompileovaných souborů, které se následně dají nalinkovat z vlastního serveru. Je také možnost nástroj nestahovat a nalinkovat si potřebné kompilované soubory přes CDN server. Ještě je také možnost získat Bootstrap s pomocí CSS preprocesoru SASS a pomocí pěti balíčkových systémů.

Bodové hodnocení: 8

Mřížkový systém a zlomové body

Využití mřížkového systému je snadné a v základu nastaveno na 12 sloupců a 4 zlomové body. Je možné mřížky velmi kombinovat, vkládat do sebe, či samostatně uskupovat. Je také nabízeno plno tříd, které mřížkám mohou měnit proporce či umístění a zároveň pak kooperují se zlomovými body.

Bodové hodnocení: 9

Navigace

Je možné použít připravená navigační okna a panely, která slouží jako menu stránky. Nasazení víceúrovňové vrstvy v navigaci také není problém, stejně tak jako jsou připravené prostory například pro logo v záhlaví a následné odkazy. V nižších rozlišeních pak odkazové položky v navigaci v základu nahrazuje hamburger menu.

Dále jsou obsaženy další navigační prvky, zejména jako funkčně jednoduché, které neslouží jako hlavní navigační menu. Lze však s nimi snadno vytvářet sekundární odkazová menu, či jako záložky obsahových karet.

Bodové hodnocení: 9

Typografické prvky a formuláře

Bootstrap obsahuje velké množství připravených typografických prvků. Základem jsou jednoduché a nastavené textové elementy, které plní základní funkce. Avšak přidáním tříd se mohou prvky značně měnit a zastávat tak funkce například tlačítek, výstražných oken atd. Definovány jsou také různé barevné varianty. Je možné využít opravdu velké spektrum připravených prvků, které plní důležité funkce.

Připravené jsou také formuláře, které se následně stanou také snadno přizpůsobitelnými a nejedná se následně o standardně prohlížečem vykreslená pole (každý prohlížeč může vykreslovat v jiné grafické podobě), ale o elegantní prvky, které jsou také snadno přizpůsobitelné.

Bodové hodnocení: 10

Tabulky

Tabulky je možné jednoduše stylizovat do připravených stylů, kterých je v množství jednotek. Tabulky nejsou v základu responzivní, stačí k nim však přidat třídu tabulky jsou

responzivní, a to metodou zachování si původní obsahu, kdy uživatel je nucen posouvat tabulku po horizontální ose. To je také jediná předpřipravená možnost.

Bodové hodnocení: 6

GUI a přizpůsobení

Grafické rozhraní a stylizační nastavení připravených prvků je na vysoké úrovni. Používá se jednoduchý moderní design a při nutnosti nastavení, či přenastavení například barev a dalších grafických detailů nabízí nástroj mnoho kaskádových tříd.

Bodové hodnocení: 9

Javascriptové knihovny

Bootstrap nabízí možnost využití vlastní Javascriptové knihovny, ta je založena na jQuery, takže musí být nejdříve importována. Na některé skripty týkající se vyskakovacích a upozorňovacích oken je třeba ještě importovat knihovnu Popper. Bootstrapové Javascriptové funkce většinou dodávají prvkům dynamiku a dobře spolupracují.

Bodové hodnocení: 9

Podpora prohlížečů

Bootstrap je podporován na nejnovějších a posledních stabilních verzích internetových prohlížečů. Co se týče Internet Exploreru, ten je podporován od verze 10+. Pouze u verze IE 10 není plná podpora HTML5 elementů a CSS3 vlastností. V případě nutnosti podpory ještě starších prohlížečů, například Internet Explorer 8 nebo 9 je možnost použít starší verzi Bootstrap 3, kde podpora zajištěna je.

Bodové hodnocení: 6

4.2.2 Hodnotící kritéria - Foundation

Foundation byl analyzován ve verzi 6.5.3, který byl v době analýzy nejaktuálnější stabilní verzí.

Velikost

14 kB (*index.html*) + 142 kB (*foundation.min.css*) + 87 kB (*jquery.min.js*) + 14 kB (*what-input.js*) + 479 kB (*foundation.min.js*) = 736 kB

Rychlost načtení

Rychlost načtení: 12,2 s

Dokumentace a implementace

Dokumentace nejdříve nabízí instruktážní videa, kde jsou často vysvětleny principy a informace. Ty jsou doplněny textovým obsahem s dostatečným vysvětlením, praktickými ukázkami, ať již použitého kódu, či vlastní zobrazení komponent, či funkce. Dokumentace je rozsáhlá a strukturovaná do jednotlivých kapitol.

Foundation je možné implementovat ručně, stažením zkompileovaných souborů. V případě zájmu je možné, taktéž kompilované soubory nalinkovat pomocí CDN serveru. V neposlední řadě je možné framework používat za pomoci preprocesoru SASS a instalovat pomocí celkem osmi balíčkových systémů.

Bodové hodnocení: 9

Mřížkový systém a zlomové body

K dispozici jsou 3 druhy mřížkových systému, avšak dle autora je nejpohodlnější využívat tu ve výchozím módu aktivovanou. Avšak v případě potřeb, kompatibility, je možné aktivovat zbylé dvě. V základu je nastaveno 12 sloupců a 3 zlomové body. Jsou také dostupné atributy a třídy, díky kterým lze mřížky snadno polohovat a měnit jim proporce.

Bodové hodnocení: 9

Navigace

Navigace je zpracována jednoduše, nabízí však pokročilejší metody funkcí panelů a menu. Ty se týkají horizontálních i vertikálních pozičně nastavených navigací. K dispozici jsou také jednoduché menu prvky, které mohou být použity například k sekundárním navigacím, či pro obsahové karty.

Bodové hodnocení: 8

Typografické prvky a formuláře

Základní typografické a textové prvky jsou ve Foundationu dostupné. Není však mnoho dalších speciálně zastoupených funkčních prvků, proto si tak pak vývojář musí vystačit s minimem, případně si vytvořit vlastní.

Formuláře existují také v minimalistické podobě, nenajde se zde velké množství nabízených formulářových prvků a formuláře jsou další z jednoduše zastávajících komponent.

Bodové hodnocení: 6

Tabulky

Tabulky nenabízí tolik stylizačních a připravených stylů, zato je možné responzivní tabulky řešit pomocí tří metod. Tabulky se na mobilních zařízeních tak mohou roztáhnout do celé šířky, nebo horizontálně posouvat tabulku v prohlížeči, nebo seskupit řádky do jedné buňky pod sebou. Funkční část tedy převyšuje tu grafickou.

Bodové hodnocení: 8

GUI a přizpůsobení

Grafické rozhraní je velmi jednoduché a spíše je zaměřeno na tu funkční část. Stejně tak není nabízeno moc tříd, které tolik upravují grafické změny u jednotlivých prvků a elementů.

Bodové hodnocení: 7

Javascriptové knihovny

Foundation má vlastní Javascriptovou knihovnu, která je stěžejní pro komplexní fungování nástroje. Vzhledem k tomu, že tento framework nabízí množství funkčních prvků, často kooperují s Javascriptem a často ani bez nich nemají smysl.

Bodové hodnocení: 9

Podpora prohlížečů

Podpora je zajištěna na nejnovějších a posledních stabilních verzích internetových prohlížečů. Internet Explorer je podporován od verze 9+.

Bodové hodnocení: 7

4.2.3 Hodnotící kritéria - PureCSS

PureCSS byl analyzován ve verzi 1.0.1, který byl v době analýzy nejaktuálnější stabilní verzí.

Velikost

14 kB (*index.html*) + 17 kB (*pure.min.css*) = 31 kB

Rychlost načtení

Rychlost načtení: 2,1 s

Dokumentace a implementace

Dokumentace není rozsáhlá, je velmi jednoduchá, i vzhledem k jednoduchosti frameworku a nabízí poměrně málo příkladů.

PureCSS se může implementovat pomocí jednoho CSS souboru pomocí jeho stažení a nahraní na vlastní server, nebo odkázáním na CDN server. PureCSS však nepodporuje SASS, a instalovat jej je možné pomocí dvou základních balíčkovacích systémů.

Bodové hodnocení: 5

Mřížkový systém a zlomové body

Flexibilní a velmi rozsáhlé a využitelné mřížky je možné pomocí až 24 sloupcové mřížky, která může být doladěna 4 zlomovými body. Layout může být nastaven velmi rozmanitě, ale nenabízí moc pozičních a speciálních tříd, které mohou mřížkové obaly modifikovat.

Bodové hodnocení: 7

Navigace

PureCSS nenabízí v základu žádný předpřipravený panel pro hlavní navigační prvky, které by zároveň obsahovalo prostory například pro logo a linkové navigační prvky. Dostupné je pouze velmi jednoduché a základní navigační menu, připomínající spíše sekundární odkazy a v základu není implementována responzivní navigace s například populárním hamburger menu.

Bodové hodnocení: 2

Typografické prvky a formuláře

Základní typografické elementy jsou upraveny minimálně, a speciálních prvků plnicí dodatečné grafické a funkční prvky nejsou dostupné ve velké míře, kromě například tlačítek.

Základní pole formulářů jsou značně upravena a zjednodušena a mohou s responzivními prvky a mřížkami posloužit pro základní využití.

Bodové hodnocení: 4

Tabulky

Tabulek pro úpravu do stylů zde nejsou téměř žádné. Stejně tak v základu nenabízí řešení pro responzivní tabulky.

Bodové hodnocení: 1

GUI a přizpůsobení

Grafika testovaného nástroje je velmi jednoduchá, často nerozeznatelná od běžných prvků, pouze lépe optimalizovaná. Funkčnost převyšuje tak grafickou stránku a pro přizpůsobení komponent zde není moc předdefinovaných tříd.

Bodové hodnocení: 3

Javascriptové knihovny

Žádnou Javascriptovou knihovnu PureCSS nenabízí, není možné tak dynamizovat prvky, či přidat jejich funkčnost díky přiložené knihovně.

Bodové hodnocení: 0

Podpora prohlížečů

PureCSS je podporována na nejnovějších stabilních verzích internetových prohlížečů, avšak často podporuje i starší verze. Podpora Internetu Exploreru je zajištěna pro verze 8+.

Bodové hodnocení: 10

4.2.4 Hodnotící kritéria - Bulma

Bulma byl analyzován ve verzi 0.8.0, který byl v době analýzy nejaktuálnější stabilní verzí.

Velikost

14 kB (*index.html*) + 190 kB (*bulma.min.css*) = 204 kB

Rychlost načtení

Rychlost načtení: 2,4 s

Dokumentace a implementace

Dokumentace obsahuje velké množství informací ať již ohledně nastavení samotného frameworku, tak i obsažených komponent, kde jsou podrobně popsány jejich funkce, využití, společně s praktickými ukázkami kódu a ilustračním zobrazením. Vzhledem k rozsahu nástroje, je dokumentace rozsáhlá, rozdělena do mnoha rozřazených kapitol.

Bulma se může implementovat pomocí jednoho CSS souboru pomocí jeho stažení a nahrání na vlastní server, nebo odkázáním na CDN server. Bulma podporuje také SASS, a instalovat jej je možné pomocí třech balíčkovacích systémů.

Bodové hodnocení: 7

Mřížkový systém a zlomové body

Za pomoci mřížkového systému je možné s Bulmou vytvořit i velmi složité layouts, v základu je nastaveno 12 sloupců a 4 zlomové body. Nástroj však obsahuje velké množství speciálních tříd, které ze základních mřížek může vytvořit skupenství, které mřížky nepřipomínají.

Bodové hodnocení: 8

Navigace

Velké množství a stylizace navigačních prvků pro hlavní menu je obsaženo v nástroji. Prostorově připravené na klasické prvky, chybí pokročilejší funkce hlavního panelu a prostředků pro jednoduché navigační prvky, například pro sekundární navigace zde také moc není.

Bodové hodnocení: 4

Typografické prvky a formuláře

Základní typografické elementy jsou upraveny minimálně, avšak speciálních prvků je zde více. Ty plní dodatečné funkční prostředky a nabízí možnosti dalších úprav editací, či přidáním kaskádových tříd.

Pro formuláře je zde vybavena samostatná sekce, kde jednotlivých tříd a možností nastavení je plno. Pouze pro některé typy polí zde chybí lepší úpravy.

Bodové hodnocení: 6

Tabulky

Tabulek pro úpravu do stylů je zde velmi malé množství. Stejně tak v základu nenabízí řešení pro responzivní tabulky.

Bodové hodnocení: 3

GUI a přizpůsobení

Grafické prostředí a grafické nadstavby pro komponenty a elementy je jednoduché minimalistické a moderní. Pro některé komponenty však nabízí velmi malou možnost dalšího přizpůsobení.

Bodové hodnocení: 5

Javascriptové knihovny

Žádnou knihovnu Bulma nenabízí, není možné tak v základu dynamizovat prvky, či přidat jejich funkčnost. V dokumentaci jsou však u některých komponent dodatečně popsány nativně Javascriptové funkce, které si musí v případě zájmu vývojář implementovat sám.

Bodové hodnocení: 4

Podpora prohlížečů

Bulma je podporována na nejnovějších stabilních verzích internetových prohlížečů. Podpora Internetu Exploreru je zajištěno pro verze 10+.

Bodové hodnocení: 6

4.2.5 Hodnotící kritéria - Materialize

Materialize byl analyzován ve verzi 1.0.0, který byl v době analýzy nejaktuálnější stabilní verzí.

Velikost

14 kB (*index.html*) + 139 kB (*materialize.min.css*) + 163 kB (*materialize.min.js*) = 316 kB

Rychlost načtení

Rychlost načtení: 5,4 s

Dokumentace a implementace

Dokumentace je dobře zpracovaná, rozdělená do kapitol a jednotlivé komponenty jsou doprovázeny ukázkami kódů společně s praktickými vizuálními příklady, někdy i videi.

Implementace je možná stažením zkompileovaných souborů a následnou vlastní implementací. Stejně tak se zkompileované soubory mohou implementovat odkázáním na CDN servery. Je možné využívat CSS preprocesor SASS a instalovat oficiálně pomocí dvou balíčkovacích systému, neoficiálně pak další tři.

Bodové hodnocení: 8

Mřížkový systém a zlomové body

Mřížkový systém je jednoduchý, nabízející v základu 12 sloupců a 3 zlomové body. Mřížky lze také kombinovat, vkládat a pozičně upravovat. Dodatečných možností však není mnoho.

Bodové hodnocení: 7

Navigace

Připraveno je plno nastavení a přizpůsobení pro hlavní navigační panel sloužící jako menu. Prostor je na přidání loga, odkazů a další užitečných prvků. Responzivní zobrazení na mobilních zařízeních se provádí transformací navigačních odkazů do hamburger menu. Chybí pouze jednoduché další nastavení pro například sekundární menu. Pro mobilní zařízení je ještě možnost jednoduše nastavit „Drag Out Menu“, které reaguje u smartphonů na gesto přejetí z vnější strany displeje na vnitřní a následně se zobrazí boční menu.

Bodové hodnocení: 7

Typografické prvky a formuláře

Základních typografických úprav u Materialize moc není. Lepším případem je u speciálních prvků označujících tlačítka, odznaky atd. Těch je o poznání více. Další výhodou je možnost integrace 932 ikon Material Designu přímo od Google.

Samostatnou kapitolou, jak tomu je i v dokumentaci, jsou formuláře. Ty mají svůj ocenitelný a nenapodobitelný styl. Je nabízeno mnoho variant i jednotlivých typů polí. Je zde velké množství nastavení pro formuláře, a to je i jedna ze silných stránek tohoto nástroje.

Bodové hodnocení: 9

Tabulky

Tabulky jsou řešeny velmi jednoduše. Moc stylů zde na výběr není. A responzivní design u tabulek se dá v základu vyřešit jednou základní metodou, a to zachováním proporcí tabulky a horizontálním posouváním v prohlížeči.

Bodové hodnocení: 5

GUI a přizpůsobení

Grafické přizpůsobení je na stylu Material Designu od firmy Google. Tato firma si drží tento styl a využívá jej pro většinu svých služeb. Materialize, kdy i název frameworku je

podobný používanému stylu, tento styl dodržuje a je možné ho tak řadit na vyšší úroveň. Chybí pouze následně větší množství grafických stylů použitím tříd, a tak je nucen si vývojář některé grafické prvky zpracovat sám

Bodové hodnocení: 7

Javascriptové knihovny

Materialize využívá vlastní Javascriptové knihovny a nabízí tak možnost své prvky dynamizovat, dodávat jim tak efekty a další funkcionality, není jich mnoho, ale dobře kooperují s vlastními elementy, výhodou je nenutnost implementovat knihovnu jQuery.

Bodové hodnocení: 8

Podpora prohlížečů

Podpora prohlížečů je pouze pro aktuální a nejnovější stabilní verze. Internet Explorer je podporován až ve verzi 11+.

Bodové hodnocení: 4

4.3 Analýza kritérií

Tato část vlastní práce analyzuje front-endové frameworky dle ohodnocených kritérií. Pro určení jednotlivých vah pro kritéria je použita Saatyho metoda, která je doporučena, pokud hodnocení provádí pouze jeden člověk, a to samotný řešitel. Dále je pro vyhodnocení analýzy použito metod váženého součtu a AHP. Dle studované literatury se tyto metody velmi hodí, je možné je použít pro jakýkoliv typ informace o preferenčních vztazích mezi komponentami modelu (Šubrt, 2011).

4.3.1 Ohodnocení kritérií variant

V tabulce č.3 je uspořádán komplexní souhrn ohodnocených kritérií jednotlivých variant front-endových frameworků. První dvě kritéria jsou naměřená a za ideální hodnotu se považuje ta nejmenší, zbylá kritéria jsou bodově ohodnocena a zde se za ideální hodnotu považuje ta nejvyšší.

Framework		Kritéria									
		1	2	3	4	5	6	7	8	9	10
1	Bootstrap	332	6,1	8	9	9	10	6	9	9	6
2	Foundation	736	12,2	9	9	8	6	8	7	9	7
3	PureCSS	31	2,1	5	7	2	4	1	3	0	10
4	Bulma	204	2,4	7	8	4	6	3	5	4	6
5	Materialize	316	5,4	8	7	7	9	5	7	8	4
Povaha		min	min	max	max	max	max	max	max	max	max

Tabulka 3 - Ohodnocení kritérií dle jednotlivých variant

Zdroj: Vlastní zpracování

4.3.2 Určení vah kritérií (Saatyho metoda)

Dle Saatyho metody jsou určeny a vypočteny jednotlivé váhy daných kritérií. Všechna kritéria mají proti sobě určené preference a díky tomu se dá zjistit jejich vliv, důležitost. Nutností pro následné analýzy je výpočet normalizovaných hodnot vah kritérií. Pro kontrolu by se měla suma všech normalizovaných vah rovnat jedné. Celkový tento souhrn je v tabulce č.4.

											Váha kritéria	Normalizovaná hodnota váhy kritéria
Kritéria	1	2	3	4	5	6	7	8	9	10		
1	1	1/3	1/3	1/5	1/3	1/5	3	1/5	1	5	3,41	0,08
2	3	1	1/3	1/5	1/3	1/3	3	1/3	3	5	4,07	0,09
3	3	3	1	1/7	1/3	1/3	3	1/5	5	3	4,36	0,10
4	5	5	7	1	7	5	7	3	7	9	7,48	0,17
5	3	3	3	1/7	1	1/5	1	1/5	3	3	4,19	0,10
6	5	3	3	1/5	5	1	5	1	5	5	5,76	0,13
7	1/3	1/3	1/3	1/7	1	1/5	1	1/7	5	5	3,67	0,08
8	5	3	5	1/3	5	1	7	1	7	9	6,58	0,15
9	1	1/3	1/5	1/7	1/3	1/5	1/5	1/7	1	3	2,56	0,06
10	1/5	1/5	1/3	1/9	1/3	1/5	1/5	1/9	1/3	1	1,74	0,04
Suma											43,82	1

Tabulka 4 - Určené váhy kritérií dle Saatyho metody

Zdroj: Vlastní zpracování

Váhy jednotlivých kritérií jsou založeny na datech dle ohodnocených kritérií a také na základě vlastních zkušenostech autora. Pomocí následných výpočtů je možné ve výsledné tabulce vidět největší preference, zejména se týkají kritéria 4 (mřížkový systém a zlomové body), dále pak ještě kritérium 8 (GUI a přizpůsobení) a kritérium 6 (typografické prvky a formuláře). Naopak méně důležité kritérium je kritérium 10 (podpora prohlížečů).

4.3.3 Analýza metodou váženého součtu

První analýza front-endových frameworků je prováděna pomocí metody váženého součtu. Aby bylo možné tuto analýzu provést, je nutné nejdříve nalézt ideální a bazální variantu. Bazální variantou se rozumí hodnota toho nejhůře ohodnoceného kritéria, ideální varianta je naopak ta nejlépe ohodnocená. Toto se provede u všech kritérií, avšak je nutná dbát na povahy kritérií, nestačí tak vybrat pouze nejvyšší a nejnižší hodnoty. Například určení těchto variant je možné provést v excelu za pomoci funkcí $min()$ a $max()$.

$$H_{ideální}$$

$$= (k1_{min}; k2_{min}; k3_{max}; k4_{max}; k5_{max}; k6_{max}; k7_{max}; k8_{max}; k9_{max}; k10_{max})$$

$$H = (31; 2,1; 9; 9; 9; 10; 8; 9; 9; 10)$$

$$D = (736; 12,2; 5; 7; 2; 4; 1; 3; 0; 4)$$

Jak již bylo zmíněno, ideální a bazální varianty jsou nezbytné pro další kroky v analýze. Díky jejich hodnotám je možné určit hodnoty standardizované kritériální matice. Tyto hodnoty reprezentují dílčí užitky jednotlivých kritérií u všech frameworkových variant. Pro výpočet dílčích užitků je tedy nutné znát ideální a bazální hodnoty a také normalizovanou hodnotu váhy kritéria, kterou jsme již získali pomocí Saatyho metody. Níže je vyobrazena názorná ukázka výpočtu hodnoty v kritériální matici pro variantu 1 a kritérium 1.

$$r_{11} = \frac{332 - 736}{31 - 736} = 0,57$$

V/K	1	2	3	4	5	6	7	8	9	10
1	0,57	0,60	0,75	1,00	1,00	1,00	0,71	1,00	1,00	0,33
2	0,00	0,00	1,00	1,00	0,86	0,33	1,00	0,67	1,00	0,50
3	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1,00

4	0,75	0,97	0,50	0,50	0,29	0,33	0,29	0,33	0,44	0,33
5	0,60	0,67	0,75	0,00	0,71	0,83	0,57	0,67	0,89	0,00

Tabulka 5 - Standardizovaná kritériální matice

Zdroj: Vlastní zpracování

Po kompletaci matice je nutné pro zjištění celkového užitku sečíst daný řádek (pokud řádky reprezentují varianty), avšak hodnota musí být vynásobena normalizovanou hodnotou váhy kritéria. Níže je opět vyobrazen příklad výpočtu, nyní však celkového užitku pro vybranou variantu.

Celkový užitek varianty 1:

$$0,57 * 0,078 + 0,6 * 0,093 + 0,75 * 0,1 + 1 * 0,171 + 1 * 0,096 + 1 * 0,131 + 0,71 * 0,084 + 1 * 0,15 + 1 * 0,058 + 0,33 * 0,04 = \mathbf{0,85}$$

		Celkový užitek variant	Pořadí
1	Bootstrap	0,85	1.
2	Foundation	0,66	2.
3	PureCSS	0,21	5.
4	Bulma	0,47	4.
5	Materialize	0,56	3.

Tabulka 6 - Celkové pořadí pomocí analýzy metodou váženého součtu

Zdroj: Vlastní zpracování

Po výpočtu celkových užiteků všech variant je vhodné si výsledky rozřadit a následně porovnat hodnoty. Vyšší hodnota znamená vyšší užitek a v tomto případě, díky analýze metodou váženého součtu, je nejvhodnější nástroj Bootstrap, naopak nejméně vhodný je PureCSS.

4.3.4 Analýza metodou AHP

Pro další způsob, jak analyzovat frameworky, je použita metoda AHP. Nejdříve je nutné zjistit váhy jednotlivých porovnávacích variant dle všech kritérií. Pro zjištění těchto údajů bude použita Saatyho metoda párového porovnávání a díky tomu se podaří zjistit váhy, avšak pro další kroky analýzy je třeba normalizované váhy určit dle hodnotících kritérií. Pro znázornění jsou uvedeny příklady v tabulkách č.7 a č.8.

						Váha varianty	Normalizovaná váha varianty	Normalizovaná váha varianty podle kritéria 1
Varianty	1	2	3	4	5			
1	1	4	1	1/3	1/6	2,55	0,168	0,013
2	1/4	1	1/4	1/7	1/9	1,324	0,087	0,007
3	6	9	6	3	1	5,000	0,329	0,026
4	3	7	3	1	1/3	3,789	0,249	0,020
5	1	4	1	1/3	1/6	2,550	0,168	0,013
Suma						15,209	1	0,08

Tabulka 7 - Váhy variant dle kritéria 1

Zdroj: Vlastní zpracování

						Váha varianty	Normalizovaná váha varianty	Normalizovaná váha varianty podle kritéria 4
Varianty	1	2	3	4	5			
1	1	1	3	2	3	3,162	0,256	0,044
2	1	1	3	2	3	3,162	0,256	0,044
3	1/3	1/3	1	1/2	1	1,780	0,144	0,025
4	1/2	1/2	2	1	2	2,449	0,199	0,034
5	1/3	1/3	1	1/2	1	1,780	0,144	0,025
Suma						12,333	1	0,170

Tabulka 8 - Váhy variant dle kritéria 4

Zdroj: Vlastní zpracování

Díky postupnému získávání dat podle všech kritérií je následně možné sestavit souhrn všech dílčích vah, které po sečtení (řádku, pokud řádky reprezentují varianty) vrátí výsledné hodnoty, které je možné porovnat.

V/K	1	2	3	4	5	6	7	8	9	10
1	0,013	0,015	0,023	0,044	0,029	0,039	0,018	0,046	0,016	0,007
2	0,007	0,007	0,028	0,044	0,026	0,022	0,027	0,034	0,016	0,009
3	0,026	0,027	0,012	0,025	0,009	0,013	0,008	0,014	0,005	0,013
4	0,020	0,024	0,016	0,034	0,014	0,022	0,011	0,023	0,009	0,007
5	0,013	0,017	0,021	0,025	0,022	0,033	0,016	0,034	0,014	0,004

Tabulka 9 - Souhrn normalizovaných dílčích vah jednotlivých variant podle všech kritérií

Zdroj: Vlastní zpracování

		Výsledná hodnota váhy variant	Pořadí
1	Bootstrap	0,249	1.
2	Foundation	0,219	2.
3	PureCSS	0,151	5.
4	Bulma	0,180	4.
5	Materialize	0,200	3.

Tabulka 10 - Celkové pořadí pomocí analýzy metodou AHP

Zdroj: Vlastní zpracování

Po součtu dílčích vah všech variant, rozřazení a následnému porovnání hodnot, kdy vyšší celková hodnota váhy znamená lepší výsledek, je patrné, že je nejvhodnější nástroj Bootstrap, a naopak nejméně vhodný je PureCSS. Stejné pořadí je zaznamenáno i u analýzy váženého součtu.

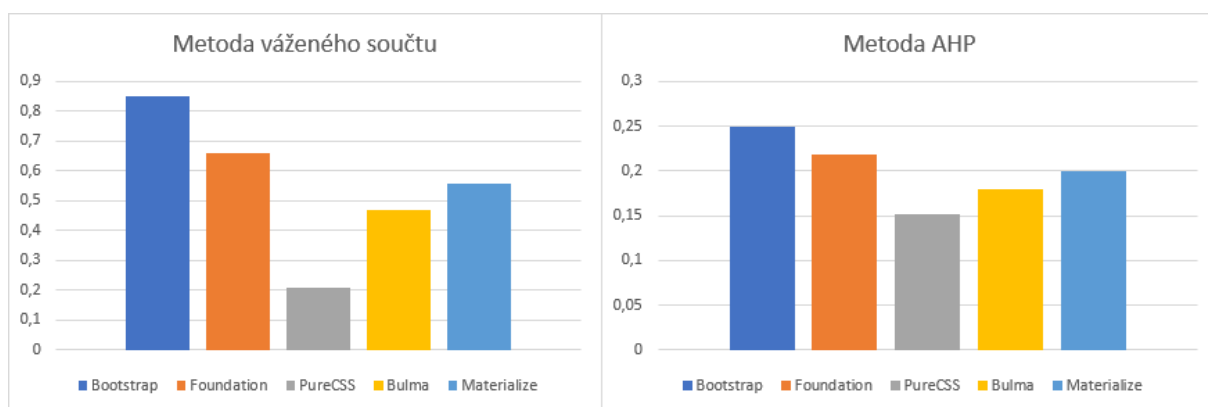
5 Výsledky a diskuse

Výsledné hodnoty vícekriteriální analýzy variant byly sloučeny do tabulky č.11 a je možné tak výsledky snadno komparovat mezi sebou.

		Výsledky analýzy metodou váženého součtu	Výsledky analýzy metodou AHP	Pořadí
1	Bootstrap	0,85	0,249	1.
2	Foundation	0,66	0,219	2.
3	PureCSS	0,21	0,151	5.
4	Bulma	0,47	0,180	4.
5	Materialize	0,56	0,200	3.

Tabulka 11 - Celkové výsledky prováděných analýz
Zdroj: Vlastní zpracování

Dále, obrázek č.26 graficky znázorňuje hodnoty výsledků analýz za pomoci sloupcového grafu. Díky tomu je možné snadno vidět rozdíly hodnot, společně s rozdílem jednotlivých analýz.



Obrázek 26 - Grafické zobrazení výsledků prováděných analýz
Zdroj: Vlastní zpracování

Dvě různé metody vícekriteriální analýzy variant přinesly stejné výsledky pořadí jednotlivých porovnávaných variant dle kritérií. První v pořadí se jako nejlépe hodnocenou variantu stal komplexní nástroj Bootstrap. Ten dominoval zejména ve všech kritériích s vysokou váhou.

Framework Foundation by mohl konkurovat výslednými hodnotami Bootstrapu, zejména pokud by dosahoval lepších výsledků v prvních dvou kritériích. Velikost a následná rychlost

načtení je nevýhodou tohoto nástroje a v těchto ohledech značně ztrácí. Tím jej analýzy, posunuly na druhou příčku v celkovém hodnocení. V ostatních kritériích by značně nezaostával, naopak v kritériích, které mají vysokou váhu, je hodnocen nadprůměrně a v některých si naopak vedl lépe než prvně umístěný framework.

Nástroj Bootstrap nelze určit jako absolutně nejlepší ve všech ostatních kritériích, přestože ve většině faktorů měl lepší hodnocení než třetí umístění nástroj Materialize a čtvrtý Bulma. Tyto dva nástroje, které se umístily na třetím a čtvrtém místě mají k sobě velmi blízko. Konkrétně svou funkční stránkou a také ve finále bodovým ohodnocením. Dle autorova názoru se jedná o jednodušší frameworky, které však mohou vyhovovat v určitých oblastech. Při vývoji specifických projektů a webových aplikací pro jednotlivé oblasti můžou být mnohem důležitější některá kritéria s nižší váhou, přestože autor je hodnotil objektivně.

Absence některých komponent má pochopitelně za následek nižší bodové hodnocení, pokud je chybějící komponenta zastoupena v některém z kritérií. Naopak dochází ve větší míře ke snížení celkového objemu nástroje, což se může projevit právě ve velikosti a rychlosti načtení. Nicméně se framework nestává tolik komplexním a dle názoru autora je vhodnější zastoupit důležitou komponentu, byť třeba jednoduchým alternativním řešením.

Právě jednoduchá sada nástrojů PureCSS nabízí rozsáhlou podporu napříč všemi prohlížeči a jejich aktuálními i neaktuálními verzemi, přestože toto kritérium nemá tak vysokou váhu, zejména díky statistickým zdrojům, které poukazují na malé procento uživatelů používající zastaralé prohlížeče. Dalším silným faktorem tohoto nástroje je velmi nízká velikost knihoven, které nabízí, stejně tak rychlost načtení webu, či webové aplikace. Tento aspekt již může hrát větší roli při výběru požadovaného nástroje, proto také má vyšší váhu, avšak celkově PureCSS naráží na nedostatek funkčních prvků a komponent, a proto mu patří poslední, páté místo v celkovém pořadí z provedených analýz.

Přímá komparace analyzovaných nástrojů a jejich výsledků s jinými nalezenými zdroji a publikacemi nebyla ve větší míře možná. V žádné z dohledatelných analýz a testů nebylo přítomno přímo těchto pět variant frameworků. Pokud by se vybraly jen některé, tak bývají porovnávány zejména ty nejpoužívanější, které jsou Bootstrap a Foundation. Rozšířený server pro vývojáře Stackshare.io (2020) společně s vývojářskou komunitou tyto dva nástroje porovnává. Často se však liší vybraná kritéria, preference řešitele, avšak ve většině případů vítězí Bootstrap, ačkoliv Foundation tolik nezaostává, stejně jako tomu bylo v prováděné analýze v této práci.

6 Závěr

Diplomová práce byla zaměřena na sady nástrojů pro tvorbu webu a webových aplikací, konkrétně front-endové frameworky. V teoretické části byly popsány webové technologie a zejména responzivní design společně s jeho důležitými aspekty. Dále byly popsány nejpoužívanější front-endové frameworky, které responzivní design a celkový vývoj webu a webových aplikací usnadňují. Na základě syntézy zjištěných poznatků získaných studiem odborných informačních zdrojů a zpracováním literárních rešerší, bylo možné získat teoretický základ pro následnou vlastní práci, která se skládala ze tří částí.

V první části vlastní práce bylo pět vybraných front-endových frameworků aplikováno při tvorbě webové aplikace. Ta byla vyvíjena pro každý nástroj zvlášť, pouze se stejným obsahem. Na základě této tvorby bylo získáno nezbytných poznatků pro výběr vhodných kritérií a následné hodnocení dle hodnotících kritérií.

Druhá praktická část se zabývala charakteristikou hodnotících kritérií, přičemž některá kritéria a jejich hodnoty byly přímo součástí webové aplikace, která byla vytvořena. Další kritéria byla vybrána na základě získaných informací, a ty také pomohly co nejpřesněji a nejobjektivněji ohodnotit hodnotící varianty. Jako kritéria byly vybrány: velikost webové aplikace společně s importovaným frameworkem, rychlost načtení v podobě simulovaného mobilního připojení pro znatelnější rozdíly, dokumentace a možnosti implementace nástroje, mřížkový systém a zlomové body, navigace, typografické prvky a formuláře, tabulky, grafické prostředí nástroje a možnost přizpůsobení, Javascriptové knihovny nabízející nástroj a podpora prohlížečů.

Ve třetí praktické části byl navržen metodický postup, který zahrnoval určení vah pro kritéria včetně následné analýzy vybraných variant front-endových frameworků, konkrétně hodnocení vícekritériální analýzy variant. Pro určení vah jednotlivých kritérií byla použita Saatyho metoda, následně pro určení nejvhodnější varianty byly použity metody váženého součtu a metody AHP. Výsledky analýz dodaly stejné pořadí variant a nejlépe ohodnocenou variantou se stal nástroj Bootstrap, který dominoval ve všech důležitých kritériích. Nejednalo se tedy však o absolutně nejlepší variantu ve všech dalších kritériích. Některé nástroje jsou zaměřeny na specifické oblasti a tyto oblasti také byly zahrnuty v souboru kritérií. Tím tyto nástroje mohly v tomto kritériu dosahovat lepších výsledků, avšak i přes tyto aspekty to v celkovém hodnocení nestačilo a komplexní nástroj Bootstrap tak lze považovat za nejlépe hodnocenou kompromisní variantu v této práci.

Spolu s rychlým rozvojem webových technologií se vyvíjí i nástroje, které usnadňují vývojářům práci a čas. Proto je pro funkční a moderní web potřeba současné verze webových technologií a nástrojů průběžně kontrolovat a analyzovat. Zastaralé technologie a nástroje mohou mít následně negativní dopad ve všech aspektech na provozované weby a webové aplikace.

Tato práce společně s jejími výsledky bude sloužit pro následnou analýzu a porovnávání front-endových frameworků prováděné v budoucnu. Výzkum prováděný v rámci diplomové práce je možné dále rozvíjet – pro oblast testovaných frameworků, kdy je možné další nástroje přidávat, či aktualizovat stávající. Konečná analýza pak přináší doporučení při výběru komplexních nástrojů pro tvorbu webu a webové aplikace, například pro firmy a koncové vývojáře.

Seznam použitých zdrojů

BROWN, Dan, 2011. Communicating Design: Developing Web Site Documentation for Design and Planning. 2nd Edition. Berkeley, CA: New Riders. ISBN 978-0321712462.

BUSTOS, Linda, 2015. Dropping Like it's Hot: M-Dot Sites to Lose 50% Share in 2015 [online]. [cit. 2020-02-01]. Dostupné z: <http://www.getelastic.com/dropping-like-its-hot-m-dot-sites-to-lose-50-share-in-2015/>

Bootstrap.com [online], 2019. [cit. 2020-02-10]. Dostupné z: <https://getbootstrap.com/>

CAO, Jerry, 2015. Responsive vs. Adaptive Design: What's the Best Choice for Designers? [online]. [cit. 2020-02-01]. Dostupné z: <http://studio.uxpin.com/blog/responsive-vs-adaptive-design-whats-best-choice-designers/>

CASTRO, Elizabeth a Bruce HYSLOP, 2012. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. Brno: Computer Press. ISBN 978-80-251-3733-8.

COULDWELL, Andrew, 2018. Responsive design: And the role of development in design [online]. [cit. 2020-02-01]. Dostupné z: <https://medium.com/owl-studios/responsive-design-af7a1f14b991>

Codeburst, 2018. JavaScript: A Step Further in Responsive Design [online]. [cit. 2020-02-01]. Dostupné z: <https://codeburst.io/javascript-a-step-further-in-responsive-design-5110732914f2>

ČÁPKA, David, 2018. ITnetwork.cz. Lekce 1 - Úvod do CSS frameworku Bootstrap [online]. [cit. 2020-03-07]. Dostupné z: <https://www.itnetwork.cz/html-css/bootstrap/kurz/uvod-do-css-frameworku-bootstrap>

Developers.google.com, 2020. Why Performance Matters [online]. [cit. 2020-03-18]. Dostupné z: <https://developers.google.com/web/fundamentals/performance/why-performance-matters>

DUCKETT, Jon, 2014. JavaScript and JQuery: interactive front-end web development. Indianapolis, IN: Wiley. ISBN 978-111-8531-648.

EVERTS, Tammy, 2013. Six reasons to ditch your m.site in 2014 [online]. [cit. 2020-02-01]. Dostupné z: <http://www.webperformancetoday.com/2013/12/04/six-reasons-to-ditch-your-m-site-in-2014/>

FRAIN, Ben, c2012. Responsive web design with HTML5 and CSS3: learn responsive design using HTML5 a CSS3 to adapt websites to any browser or screen size. Birmingham: Pack Publishing. ISBN 978-184-9693-189.

Foundation.zurb.com [online], 2019. [cit. 2020-02-10]. Dostupné z: <https://foundation.zurb.com/>

GARRETT, Jesse James, 2011. The elements of user experience: user-centered design for the web and beyond. 2nd Edition. Berkeley, CA: New Riders. ISBN 978-0321683687.

Geekflare.com, 2020. 10 Best CSS Frameworks for Front-End Developers [online]. [cit. 2020-03-09]. Dostupné z: <https://geekflare.com/best-css-frameworks/>

GREMILLION, Ben, Jerry CAO, Kamil ZIEBA a Zack RUTHERFORD, 2015. Responsive Web Design Best Practices [online]. [cit. 2020-02-01]. Dostupné z: <https://www.uxpin.com/studio/ebooks/responsive-web-design-best-practices/>

Github, 2018. Saving repositories with stars [online]. [cit. 2020-02-01]. Dostupné z: <https://help.github.com/en/github/getting-started-with-github/saving-repositories-with-stars>

Google, 2012. Recommendations for building smartphone-optimized websites [online]. [cit. 2020-02-01]. Dostupné z: <http://googlewebmastercentral.blogspot.ca/2012/06/recommendations-for-building-smartphone.html>

Google, 2013. Plusnet Uses Responsive Web Design, Sees Traffic Grow 2x and Sales 10x on Smartphones and Tablets [online]. [cit. 2020-02-01]. Dostupné z: <https://www.thinkwithgoogle.com/marketing-resources/experience-design/plusnet-responsive-web-design/>

Google, 2015. Rolling out the mobile-friendly update [online]. [cit. 2020-02-01]. Dostupné z: <https://webmasters.googleblog.com/2015/04/rolling-out-mobile-friendly-update.html>

Google, 2018. A new image format for the Web [online]. [cit. 2020-02-01]. Dostupné z: <https://developers.google.com/speed/webp>

Google, 2018. Lighthouse Scoring Guide [online]. [cit. 2020-02-01]. Dostupné z: <https://developers.google.com/web/tools/lighthouse/v3/scoring>

HOGAN, Brian P., 2011. HTML5 a CSS3: výukový kurz webového vývojáře. Brno: Computer Press. ISBN 978-80-251-3576-1.

HOLST, Christian, 2012. Adaptive vs. Responsive Layouts and Optimal Form Field Labels [online]. [cit. 2020-02-01]. Dostupné z: <http://www.smashingmagazine.com/2012/11/08/ux-design-qa-with-christian-holst/>

HOUŠKA, Milan, 2013. Kurz: Ekonomicko matematické metody I - KS - ZS 13/14 [online]. [cit. 2020-02-22]. Dostupné z: <https://moodle.czu.cz/course/view.php?id=805>

ITZKOVITCH, Avi, 2012. Creating an Adaptive System To Enhance UX [online]. [cit. 2020-02-01]. Dostupné z: <http://www.smashingmagazine.com/2012/12/creating-an-adaptive-system-to-enhance-ux/>

JAROSZ, Michal, 2018. 5 Practical Solutions to Make Responsive Data Tables [online]. [cit. 2020-02-01]. Dostupné z: <https://medium.com/appnroll-publication/5-practical-solutions-to-make-responsive-data-tables-ff031c48b122>

KADLEC, Tim, 2014. Responzivní design profesionálně. Brno: Zoner Press. Encyklopedie Zoner Press. ISBN 978-80-7413-280-3.

KRAUSE, Jörg, [2016]. Introducing Bootstrap 4. New York, NY: Apress. ISBN 978-148-4223-819.

Konigi, 2009. Wireframes [online]. [cit. 2020-02-01]. Dostupné z: konigi.com/uiref/wireframes/

LINDLEY, Cody, 2018. Front-End Developer Handbook 2018 [online]. [cit. 2020-02-01]. Dostupné z: <https://frontendmasters.com/books/front-end-handbook/2018/>

MARCOTTE, Ethan, 2014. Responsive Web Design. 2nd edition. New York: A Book Apart. ISBN 978-1-9375571-8-8.

MICHÁLEK, Martin, 2017. Vzhůru do (responzivního) webdesignu. Verze 1.1. Praha: vlastním nákladem autora. ISBN 978-808-8253-006.

MICHÁLEK, Martin, 2018. Jednotky pro tvorbu webu (em, rem, %, px, vh, vw): Kde použít jakou? [online]. [cit. 2020-02-01]. Dostupné z: <https://www.vzhurudolu.cz/prirucka/jednotky>

MOFFITT, Jack a Frederic DAOUD, [2014]. Seven web frameworks in seven weeks: adventures in better web apps. Raleigh, NC: Pragmatic Bookshelf. ISBN 978-1937785635.

MURTAGH, Rebecca, 2014. Mobile Now Exceeds PC: The Biggest Shift Since the Internet Began. Search Engine Watch [online]. [cit. 2020-02-01]. Dostupné z: <https://www.searchenginewatch.com/2014/07/08/mobile-now-exceeds-pc-the-biggest-shift-since-the-internet-began/>

Mozilla, 2018. Graphic design for responsive sites [online]. [cit. 2020-02-01]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Responsive/Graphics_for_responsive_sites

OTTO, Mark, 2012. Bootstrap in A List Apart No. 342 [online]. [cit. 2020-02-10]. Dostupné z: <https://markdotto.com/2012/01/17/bootstrap-in-a-list-apart-342/>

OTTO, Mark, 2013. Bootstrap 3 released [online]. [cit. 2020-02-10]. Dostupné z: <https://blog.getbootstrap.com/2013/08/19/bootstrap-3-released/>

- OTTO, Mark, 2018. Bootstrap 4 [online]. [cit. 2020-02-10]. Dostupné z: <https://blog.getbootstrap.com/2018/01/18/bootstrap-4/>
- Purecss.io: A set of small, responsive CSS modules that you can use in every web project. [online], 2019. [cit. 2020-02-10]. Dostupné z: <https://purecss.io/>
- RIZZO, Davide, 2017. Accessible, Simple, Responsive Tables [online]. [cit. 2020-02-01]. Dostupné z: <https://css-tricks.com/accessible-simple-responsive-tables/>
- Stackshare.io [online], 2020. [cit. 2020-03-18]. Dostupné z: <https://stackshare.io/stackups/bootstrap-vs-foundation>
- ŠUBRT, Tomáš, 2011. Ekonomicko-matematické metody. Plzeň: Vydavatelství a nakladatelství Aleš Čeněk. ISBN ISBN978-80-7380-345-2.
- THOMAS, Jeremy, 2018. Creating Interfaces with Bulma. Santa Rosa: Bleeding Edge Press. ISBN 9781939902498.
- UZAYR, Sufyan, 2013. Responsive vs. Adaptive Design [online]. [cit. 2020-02-01]. Dostupné z: <http://torquemag.io/responsive-design-vs-adaptive-design/>
- W3C, 2015. How TO - Responsive Tables [online]. [cit. 2020-02-01]. Dostupné z: https://www.w3schools.com/howto/howto_css_table_responsive.asp
- W3C, 2015. Responsive Web Design Introduction [online]. [cit. 2019-10-5]. Dostupné z: https://www.w3schools.com/css/css_rwd_intro.asp
- W3C, 2018. How TO - Media Queries with JavaScript [online]. [cit. 2020-02-01]. Dostupné z: https://www.w3schools.com/howto/howto_js_media_queries.asp
- W3Techs, 2018. Usage statistics of JavaScript libraries for websites [online]. [cit. 2020-02-01]. Dostupné z: https://w3techs.com/technologies/overview/javascript_library/all
- WANG, Alvin, 2019. About Materialize [online]. [cit. 2020-02-10]. Dostupné z: <https://materializecss.com/about.html>
- WODTKE, Christina a Austin GOVELLA, 2009. Information Architecture: Blueprints for the Web. 2nd Edition. Berkeley, CA: New Riders. ISBN 978-0321600806.
- ZMIJEWSKI, Bryan, 2019. About Zurb [online]. [cit. 2020-02-10]. Dostupné z: <https://zurb.com/about>