

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství (PEF)



Bakalářská práce

Webdesign kadeřnického salónu

Filip Boye-Kofi

© 2017 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Filip Boye-Kofi

Informatika

Název práce

Webdesign kadeřnického salónu

Název anglicky

Webdesign of a hairdressing saloon

Cíle práce

Cílem práce je navrhnout a vytvořit funkční prototyp webové stránky. Součástí dokumentace projektu bude popis postupu webové stránky za použití běžných technologií, které se vyskytují u většiny českých poskytovatelů internetových služeb v různých možných druzích aktivních webových stránek s databází.

Metodika

V práci budou použity metody a standardy softwarového inženýrství. Použijte prvky responzivního designu. Navržené řešení musí být funkční u standardního poskytovatele základních internetových služeb, které zahrnují pouze PHP, MySQL a přístup k souborům přes FTP.

Doporučený rozsah práce

30-40 stran

Klíčová slova

Webová stránka

Doporučené zdroje informací

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

RESIG, John. JavaScript a Ajax: moderní programování webových aplikací. Brno: Computer Press, 2007. ISBN 978-80-251-1824-5.

VRÁNA, Jakub. 1001 tipů a triků pro PHP. Brno: Computer Press, 2010. ISBN 978-80-251-2940-1.



Předběžný termín obhajoby

2017/18 LS – PEF

Vedoucí práce

doc. Ing. Vojtěch Merunka, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 7. 3. 2018

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 7. 3. 2018

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 15. 03. 2018

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci „Webdesign kadeřnického salónu“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15. března 2018

Poděkování

Na tomto místě bych rád poděkoval svému vedoucímu bakalářské práce doc. Ing. Vojtěchu Merunkovi, Ph.D. za odborné vedení, který byl vždy trpělivý a od kterého jsem rád přijímal rady a připomínky, které mi poskytoval při psaní této práce.

Webdesign kadeřnického salónu

Abstrakt

Tato bakalářská práce je zaměřena na vývoj webové stránky pro kadeřnický salón, kde jsou implementovány technologie pro moderní a svěží web. Zároveň implementují a popisují nástroje, které jsou nezbytnou součástí programátora.

Klíčová slova: Webová stránka, PHP, Framework, Database, Javascript, Knihovna

Webdesign of a hairdressing saloon

Abstract

This bachelor thesis focuses on the development of web applications for the hairdressing salon, where technologies for modern and fresh ones are implemented. At the same time, they implement and describe frameworks that are the daily routine of a programmer.

Keywords: Web page, PHP, Framework, Database, Javascript, Libraries

Obsah

1	Úvod	10
2	Cíl práce a metodika	11
	Cíl práce	11
	Metodika	11
3	Teoretická východiska	12
3.1	Existující nástroje	12
3.1.1	WordPress	12
3.1.2	Formoid	12
3.2	Používané Frameworky a databáze při tvorbě webu	12
3.2.1	Laravel Framework	13
3.2.2	Nette	14
3.2.3	MySQL	14
3.3	Používané technologie při tvorbě webu	14
3.3.1	jQuery	15
3.3.2	Ajax	15
3.3.3	JavaScript	16
3.3.4	Responzivní design	17
3.3.5	CSS	18
3.3.6	HTML	20
	Níže si uvedeme, jaké tagy se mohou vyskytovat u body:	21
3.3.7	Bootstrap	22
3.3.8	PHP	24
4	Vlastní práce	25
4.1	Implementace technologií	25
4.1.1	HTML	25
4.1.1.1	Horní menu	26
4.1.1.2	Úvodní strana	27
4.1.1.3	Naši umělci	27
4.1.1.4	O nás	29
4.1.1.5	Ceník	30
4.1.1.6	Fotogalerie	31
4.1.1.7	Kontakt	32
4.1.1.8	Kontaktní formulář	33
4.1.1.9	Děkovná stránka v HTML	34
4.1.2	Technologie v praxi	34
4.1.2.1	Horní menu pomocí tříd	35
4.1.2.2	Úvodní strana pomocí technologií	36
4.1.2.3	Naši umělci pomocí technologií	39

4.1.2.4	Oddělovací čára pomocí technologií	40
4.1.2.5	O nás v Cascading Style Sheet	40
4.1.2.6	Ceník pomocí technologií	41
4.1.2.7	Fotogalerie pomocí technologií	42
4.1.2.8	Kontakt	43
4.1.2.9	Scrolling pomocí JavaScriptu	43
4.1.2.10	Dotazník pomocí PHP a MySQL	44
5	Závěr	46
6	Seznam použitých zdrojů	47
6.1	Monografické zdroje	47
6.2	Internetové zdroje	47
7	Přílohy	50

Seznam obrázků

Obrázek 1	- Logo Nejznámějšího frameworku	13
Obrázek 2	- Logo MySql	14
Obrázek 3	- Příklady responzivního designu	18
Obrázek 4	- Logo CSS	20
Obrázek 5	- Logo HTML	22
Obrázek 6	- Logo pohyblivé animce pomocí bootstrapu	23
Obrázek 7	- Kód v PhpStorm	31
Obrázek 8	- Výsledek CSS	38

Seznam tabulek

Tabulka 1	- Kód pro responzivní deisgn	17
Tabulka 2	- První způsob pro implementaci style	18
Tabulka 3	- Druhý způsob pro implementaci style	19
Tabulka 4	- Třetí způsob pro implementaci style	19
Tabulka 5	- Tabulka způsobů zápisů stylů	20
Tabulka 6	- Ukázka párového tagu	21
Tabulka 7	- Tagy u body	21
Tabulka 8	- Tagy a atributy u formulářů	22
Tabulka 9	- Horní menu pouze v HTML pro mobily	26
Tabulka 10	- Kód „Horní menu“ pouze v HTML pro desktopy	26
Tabulka 11	- Kód „Úvodní stránka“ pouze v HTML	27
Tabulka 12	- Kód "Naši umělci" pouze v HTML	27
Tabulka 13	- Kód pro tlačítka bootstrapu pouze v HTML	28
Tabulka 14	- Zbytek Kódu pro tlačítka bootstrapu pouze v HTML	28
Tabulka 15	- Kód "O nás" pouze v HTML	29
Tabulka 16	- Kód "Ceník" pouze v HTML	30
Tabulka 17	- Kus kódu horní tabulky	30
Tabulka 18	- Kód "Fotogalerie" pouze v HTML	31
Tabulka 19	- Kód "Kontakt" pouze v HTML	32
Tabulka 20	- Kód "Otevírací doba" pouze v HTML	33
Tabulka 21	- Kontaktní formulář pouze v HTML	33
Tabulka 22	- Děkovná stránka pouze v HTML	34
Tabulka 23	- Body v CSS	35
Tabulka 24	- Třída pro horní menu v CSS	35
Tabulka 25	- Třída pro změnu barev v horním menu v CSS	36

Tabulka 26 - Třída pro horní menu pomocí stylu	36
Tabulka 27 - Třída pro úvod	36
Tabulka 28 - Třída pro tlačítko	37
Tabulka 29 - Responzivní design	38
Tabulka 30 - Třída bootstrapu	39
Tabulka 31 - Kód pro animaci	39
Tabulka 32 - Animace při načítání stránky	39
Tabulka 33 - Přesměrování	39
Tabulka 34 - Manuální proklikávání	39
Tabulka 35 - Implementace třídy pro čáru v HTML	40
Tabulka 36 - Třída pro čáru	40
Tabulka 37 - Třidy v sekci "O nás"	41
Tabulka 38 - Přímý styl	41
Tabulka 39 - Třída v "Ceník"	42
Tabulka 40 - Třidy pro sekci "Fotogalerie"	42
Tabulka 41 - Třída pro okraje textu	43
Tabulka 42 - Třída pro vizuální prvky v otevírací době	43
Tabulka 43 - Scrollování	44
Tabulka 44 - PHP kód pro zpracování dat	44

1 Úvod

Internet dnes představuje celosvětovou síť vzájemně propojených počítačů pomocí www stránek. Stránky jsou vytvořeny různými technologiemi. Na internetu se můžeme setkat i s jednoduchými a amatérsky vyhlížejícími stránkami. Všem tvůrcům jde o to, aby svým uživatelům a zákazníkům přiblížili svoji činnost a umožnili jim si vytvořit představu o jejich službách. S rostoucím počtem osob pohybujících se na internetu tato prezentace nabývá na důležitosti a stává se základním marketingovým nástrojem a standartním prostředkem.

Mé téma pojednává o procesu vytvoření fungující webové stránky, kde uživatelé mohou plně využít potenciál daného webu, v tomto případě kadeřnictví. Chci poukázat na celý proces tvorby webové stránky. Návrh by měl po implementaci odpovídat potřebám uživatele a byznysu. Webové stránky by měli být pro uživatele snadno pochopitelné, použitelné a užitečné.

V teoretické části práce jsou uvedeny pojmy, které souvisí s tvorbou webových stránek, dělení technologií a používané technologie. Praktická část bude popisovat faktický proces jejich tvorby.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem práce je navrhnout a vytvořit funkční prototyp webové stránky. Součástí dokumentace projektu bude popis postupu webové stránky za použití běžných technologií, které se vyskytují u většiny českých poskytovatelů internetových služeb v různých možných druzích aktivních webových stránek s databází.

2.2 Metodika

V práci budou použity metody a standardy softwarového inženýrství. Budou použity prvky responzivního designu. Navržené řešení musí být funkční u standardního poskytovatele základních internetových služeb, které zahrnují pouze PHP, MySQL a přístup k souborům přes FTP.

3 Teoretická východiska

3.1 Existující nástroje

3.1.1 WordPress

Domnívám se, že mnoho lidí používá pro tvorbu webové stránky WordPress. To je v pořádku, pokud se budeme věnovat tvorbě webové stránky „povrchově“ nebo pro tvorbu blogů. Pokud požadujeme vytvořit plnohodnotnou webovou stránku se složitým systémem, tak WordPress nebude dostačující. Tento nástroj je psaný v PHP a databáze v MySQL. Byl vytvořen pro masu lidí, neboť je „user-friendly“. WordPress obsahuje a dodržuje standardy XML, CSS a XHTML. Tento nástroj je velmi rozsáhlý po stránce plug-in pro rozšíření funkcí. [4]

3.1.2 Formoid

Vybral jsem si k popisu Formoid, protože chci popsat nástroj i z jiného úhlu pohledu. Jde o nástroj určený jak pro začátečníky, tak i pro pokročilé programátory. Pomocí tohoto programu můžeme vytvářet jednoduché, ale i komplexní formuláře bez znalostí jazyka, který se nám vygeneruje do kusu kódu. Výhodou Formoidu je přehledné prostředí, kde jsou zobrazeny rady, a proto člověk nemusí být nejlepší programátor. Výhodou je pohodlná vizuální tvorba, která se vygeneruje do HTML/CSS a vytvoří i PHP kód pro práci s daty z formulářů. Zajímavá je možnost Captcha, která slouží proti spamu, jenž může zatížit webovou stránku. Nutno je zmínit také responzivní design. Tento nástroj se používá jak pro mobilní systémy, tak i pro desktopové, popřípadě i tabletové systémy. Je založen na bázi „drag n drop“, kdy přetáhneme prvek do určité oblasti a v pozadí se vygeneruje kód. [5]

3.2 Používané Frameworky a databáze při tvorbě webu

Framework slouží jako pomocník programátorovi. Ve frameworku se může kolaborovat s API, ale také může obsahovat pomocné programy. Nesmíme zapomenout, že framework disponuje tím, že doporučuje postupy při vývoji aplikací. Framework převezme typickou problematiku, přičemž se usnadní vývoj tak, aby vývoj se upíral přesnějším směrem. Framework se skládá ze dvou prvků „Hot spots“ a „Frozen spots“.

Hot spots se mění v unikátní funkcionalitu, o kterou se stará programátor. Frozen spots se naproti tomu vůbec nemění. [3] Setkal jsem se s různými názory, že frameworky „zpomalují“ a „ztrácejí čas“. Z mého pohledu je největším problémem to, že pokud smažeme Framework, nebudeme moci spustit některé aplikace. Ukážeme si dva základní frameworky, se kterými jsem se setkal a které je dobré znát při tvorbě webové stránky. Jedná se o základy. [6] [7] [8]

3.2.1 Laravel Framework

Laravel Framework dále jen LF je open source v PHP, který nám nabízí několik funkcionalit. Umožňuje vývojářům pracovat s kešováním, posíláním e-mailů a komunikací v databázi. Dále si zde můžeme nastavit, co se právě děje v naší aplikaci. Laravel poskytuje masivní protokolovací služby, které umožňují ukládat zprávy do souborů. Systémové chyby můžou upozornit vývojářský tým na Slack. Laravel disponuje několika přístupy k ověření příchozích dat. Laravel poskytuje velmi jednoduchou a elegantní API pomocí populární knihovny SwiftMailer s ovladači pro Mailgun, SMTP, Mailgun, SparkPost, PHP email funkce, což nám umožní rychle odesílat poštu prostřednictvím cloudových nebo lokálních služeb. [9] [10]

Obrázek 1 - Logo Nejznámějšího frameworku



zdroj: <https://www.codeforest.net/getting-started-laravel>

3.2.2 Nette

První z českých frameworků je Nette v php s open sourcem, s tímhle frameworkem jsem pracoval při tvorbě on-line shopu. Je psaný v PHP. Tento framework používají známé webové stránky jako je například CSFD.CZ nebo ULOZ.TO. Nette disponuje eliminací bezpečnostních děr uvnitř systému jako „session hijacking“. Výhoda stojící za zmínku je i to, že vás tlačí k čistému kódu pro budoucí rozšíření. [11]

3.2.3 MySQL

MySQL je databáze, která je založená na tabulkách neboli relační databáze. Každá tabulka se odkazuje na reálný svět. Položky jsou řádky a atributy - chceme-li vlastnosti, které jednotlivé položky mají. Každý sloupec má svůj specifický datový typ Int, Date či Varchar. Každý řádek by se měl opatřit nejlépe umělým identifikátorem (může se použít i identifikátor z reálného světa - třeba číslo občanského průkazu), který má za úkol danou věc očíslovat. Tabulky pak mohou mezi sebou komunikovat. Každá tabulka by měla mít svůj primární klíč neboli „Primary key“, díky němuž můžeme propojovat tabulky a vybírat hodnoty ze dvou tabulek do jedné. [12] [13]

Obrázek 2 - Logo MySql



zdroj: <https://howtolearn.me/how-learn-mysql-free/mysql-logo/>

3.3 Používané technologie při tvorbě webu

V této kapitole se budu věnovat popisu technologií webových stránek. Jsou to především věci, které usnadňují práci programátorům. O technologiích budu mluvit

obecně, protože je mnoho technologií, které člověk může a nemusí používat. My se zaměříme na věci ohledně dat a jejich použitelnosti na webu. Má práce je zaměřena jiným směrem. Budu stručně uvádět příklady využití dané technologie. Můj hlavní programovací jazyk bude PHP - objektově orientovaný, který se v dnešním světě považuje za jeden z těch „nejhlavnějších“ jazyků vůbec. Důvod, proč jsem si zvolil tento jazyk je, že jsem s ním pracoval a mám s ním zkušenosti. Výhodou je široký výběr knihoven, které ulehčují práci vývojáři. Vývojové prostředí jsem použil PhpStorm. Nastíníme si pár informací ohledně JavaScriptu, jQuery, responzivní design, bootstrap.

3.3.1 jQuery

jQuery je populární JavaScript knihovna používaná pro interaktivní webové stránky. Je to komplexní nástroj, který pomůže programátorovi vytvořit projekt. Důvodem této knihovny je používat snadněji JavaScript. jQuery bere mnoho běžných úloh, které mají mnoho řádků v JavaScriptu a zabalí je do metody, kterou lze volat pomocí jednoho řádku. Je mnoho plug-in, které jsou založené na této technologii. Pomocí této technologie můžeme pro uživatele udělat web více přívětivější a komfortnější než webové stránky v zastaralých nástrojích. jQuery obsahuje a disponuje několika funkcionalitami např. Ajax, který je důležitý pro tvorbu webové stránky. Umožňuje manipulovat s CSS. [14] [15]

Tuto knihovnu získáme pomocí tohoto příkazu

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
```

Tímto se načítá knihovna a lze s ní poté libovolně manipulovat.

3.3.2 Ajax

Ajax, neboli “Asynchronous Javascript And XML” značí, že se jedná o Javascript, který pracuje s XML. Při používání Ajaxu nemusí uživatel použít reloading stránky. Javascript se vytváří na straně klienta a má schopnost provádět asynchronní operace. Javascript zavolá script, který je uložen na serveru a nezdržuje se odpovědí, ale místo toho vykonává kód. V podstatě Ajax není žádný framework, ani nástroj z třetí strany, na kterém bychom byli závislí, ale pracuje pomocí Javascriptu. Máme možnost

vyměňovat data se skriptem v PHP na webové stránce pomocí XML,HTML,JSON. Ajax je ta správná technologie při formuláři nebo při vyplnění prvku. Zkontroluje data před odesláním. Ajax umožňuje asynchronní volání směrem na webový server. Tato metoda umožňuje, aby web browser klienta zabránil čekání na příchod všech dat, než dovolí uživateli fungovat. Díky tomu, že webová stránka disponuje Ajaxem, je rychlejší, a proto uživatelsky komfortnější. Tato technologie byla tvořena z několika důvodů - jeden z nich byla rychlost a výkonnost, která zlepšovala funkčnost webové aplikace. Příklad hodnocení filmů na čsfd.cz kdy uživatelé ohodnotí film a jsou zadány hodnoty rovnou do databáze, aniž by se odesílala stránka. [15] [16]

3.3.3 JavaScript

Javascript je skriptovací jazyk, který ovlivnil webové stránky, které používáme dnes. Javascript dále jen JS je opak PHP co se týká entit, se kterými pracují tyto jazyky. Na střední škole jsme se učili to, co využívá mnoho webů, a to editor textu, který je většinou implementován zásluhou JS. Má ale i jiné funkce, jako je po najetí myši - roletka. Webové stránky často používají aktuální čas, to je také zásluhou JS. JS, ale nesmíme přeceňovat. Javascript je na straně klienta. Klient si ho může sám přepsat nebo vypnout, z tohoto důvodu je validace čehokoliv nebezpečná. Jazyk je objektově orientovaný. Některé html kódy mohou obsahovat škodlivý JS v podobě IFRAME. Většina rozšíření webového prohlížeče Google Chrome jsou použity pomocí javascriptu. Open office dovoluje použít JavaScript jako skriptovací jazyk. Unity 3D umožňuje pomocí upraveného JavaScriptu skriptovat pomocí Mono. Důvod vývoje JS byl, že stránky, u kterých bylo použito pouze HTML, byly velmi statické. Proto se vytvořil nový skriptovací jazyk, abychom mohli dodat dynamiku. Cílem JS bylo uživatelům dát možnost interakci s danou webovou stránkou. Postupem času se JavaScript “zmodernizoval”. Díky tomuto jazyku můžeme vytvářet i webové aplikace. Implementací JS můžeme pomocí tagu v HTML dokumentu a nebo podle mého názoru lepší řešení pomocí externího souboru. Pomocí tagu stačí zadat `<script>...</script>` ve kterém bude určitý kód. Rozdíl externího oproti tagu je pouze ten, že dáváme source tudíž “src” a vkládáme do `<headu>`. [17] [2]

3.3.4 Responzivní design

Moje zkušenost mi říká, že valná část webových stránek je přizpůsobena na mobilní zařízení, PC, tablety pomocí technologie „responsive design“. Podle statistiky je 52% návštěvníků tvořeno lidmi, kteří zobrazují webové stránky pomocí mobilních telefonů, 44% je tvořeno skupinou, která stránky navštívuje z desktopů a pouze 4% je zastoupení uživatelů tabletů. Důvod proč toto vysvětluji je, že webová stránka se chová a hlavně vypadá jinak na PC, než na mobilním zařízení. Pojem responzivní design je tvorba webových stránek, která se dokáže adaptovat na dané zařízení s využitím media breakpoints. Přístupy jsou pouze dva, desktop first a mobile first. [18]

Desktop first už podle názvu znamená, že se stránka nejdříve udělá pro desktop a následně pro mobily. Bohužel nevýhoda je, že změna pro zařízení je složitá a vyžaduje přepsání kódu. Zatímco mobile first používá opačný přístup, kdy se snaží vytvořit stránku na mobily a následně na PC. Pro shrnutí responzivita je jedna z technologií, se kterými vypadá náš web “moderněji”. [19]

Vybral jsem ukázkový kód pomocí, kterého se dělá responzivní design. Tento kód je uložen v style.css

Tabulka 1 - Kód pro responzivní design

```
@media(min-width:500px){  
    h2{  
        color:red;  
    }  
}
```

Zdroj: autor

Zde je obrázek, jak vypadá responzivní design pro desktopy, tablety a mobilní zařízení.

Obrázek 3 - Příklady responzivního designu



zdroj: https://cs.wikipedia.org/wiki/Responzivn%C3%AD_web_design

3.3.5 CSS

Obecně CSS je “upgrade” HTML jazyka o grafické prvky, které zkrášlí web tam, kde už ho nedokáže HTML “zkrášlit”. Zkratka CSS je “Cascading Style Sheets”. Tento jazyk je značkovací jako HTML. CCS dělíme na CSS2 a CSS3. Dokumentace je rozdělena na “moduly”. Tento jazyk byl vytvořen organizací W3C - “World Wide Web Consortium”. První z nich je přímo ve zdroji daného textu pomocí atributu “style=...” náš učitel na střední škole to označoval jako “přímý styl” a já tohle označení používám dodnes. Druhý ze tří stylů je v hlavičce nadefinovaný jako seznam úkonů, které se objeví v kódu a musí se přeformátovat, jak je uvedeno v stylesheetu. Třetí a poslední způsob je využitím externího stylesheetu, který je z mého pohledu nejlepší. Odkazuje se pomocí souboru a pojmenovává style.css nebo spíše doporučuji toto výstižné pojmenování. [20] [1]

Nyní ukáži, jak vypadá CSS zápis style.

První zápis vypadá takto:

Tabulka 2 – První způsob pro implementaci style

```
<p style="color: blue">Bakalářská práce.</p>
```

zdroj: autor

Druhý zápis vypadá takto:

Tabulka 3 - Druhý způsob pro implementaci style

```
<style>
  p{color:red}
</style>
```

zdroj: autor

Třetí zápis vypadá takto:

Tabulka 4 – Třetí způsob pro implementaci style

```
html,body{
  margin:0;
  padding:0;
  height:100%;
  width:100%;
  background-color:rgb(144,186,205);
}
```

zdroj: autor

Do kódu musíme zapsat `<link rel="stylesheet" type="text/css" href="style(název souboru může být jakýkoliv).css">`

Třetí zápis je v kódu a říká nám, že šířka vnějšího okraje je nulová (margin), šířka vnitřního okraje je též nulová. Výška je 100 procent a šířka je také 100 procent a to vše se obarví podle RGB. Tento kus kódu se aplikuje na HTML a BODY. Hodnoty, které do některých atributů můžeme definovat podle:

procent (height: 100%)

pixelů (img {width: 50px;})

pomocí slov (float: right; float: auto, float: left,)

U barev je to:

RGB: rgb(144, 186, 205)

Hexadecimální barvou: #ff0000

slovy: "red"

Obrázek 4 - Logo CSS



zdroj: <https://appendto.com/2016/11/what-is-css-grouping-and-what-is-it-used-for/>

Tabulka způsobů zápisů

Tabulka 5 - Tabulka způsobů zápisů stylů

TAG	<code>body {width: 100%;}</code>
Třída	<code>.animace {text-align: center;}</code>
ID	<code>#icon2 { border-radius: 50px 50px; }</code>
Deklarace více tagů	<code>H1, H2 {color: blue}</code>
Selektor potomka	<code>ol li {font-style: italic}</code>
Pseudotřída	<code>a:hover {color: green}</code>

zdroj: jakpsatweb.cz

3.3.6 HTML

HTML - HyperText Markup Language je v informatice hlavním jazykem, který vytváří základ webové stránky. Díky němuž můžeme nadefinovat „layout“ tabulky. Tento jazyk je značkovací a neobsahuje žádné proměnné. Zapotřebí je pouze znalost všech tagů, atributů. HTML má svojí specifickou strukturu. Vložení tohoto tagu říkáme, že textový soubor je HTML dokument. Tento dokument se pak dělí na dvě části hlavička (head) a tělo (body). Hlavička informuje prohlížeče a vyhledávače, v těle je

samotný kód. Body obsahuje informaci, co se má zobrazit na stránce. [21] Některé tagy jsou párové a některé nejsou. Pokud začíná počáteční a koncovou značkou je párový. Přesný příklad je tento:

Tabulka 6 - Ukázka párového tagu

```
<form>
  <input type="submit" value="odeslat">
</form>
```

zdroj: autor

zatímco nepárový vypadají takto:

```
<br>
```

Níže si uvedeme, jaké tagy se mohou vyskytovat u body:

Tabulka 7 - Tagy u body

Atribut tagu	Co to je	Jaké hodnoty mohou být
Background	Obrázek na pozadí	Uložený obrázek, URL
Text	Barva textu	Barva
Bgcolor	Barva pozadí	Barva

zdroj: autor

Formuláře jsou také jednou z hlavních věcí HTML, kdy pomocí nich nadefinujeme textboxy, inputy a mnoho dalších věcí. Je to párové, takže to musí být ukončeno zpětným lomítkem.

Tabulka 8 - Tagy a atributy u formulářů

Tag:	Atribut:	Co to je:	Jak to vypadá:	Párový/Nepárový
Form		Formular		Ano
Input	Radio	Prepinac	<input type="radio"/>	Ne
Input	Text	Text	<input type="text" value="BP"/>	Ne
Input	Checkbox	Zaskrtavac	<input type="checkbox"/>	Ne
Input	Password	Heslo	<input type="password" value=".."/>	Ne
Input	Submit	Tlacidlo odesilani	<input type="submit" value="BP"/>	Ne

zdroj: jakpsatweb.cz

HTML je “základním kamenem” k vytvoření webové stránky. Je ovšem nezbytné umět pracovat i s různými jinými nástroji a technologiemi.

HTML5 je nejnovější verzí HTML kterou dnes můžeme používat. Tato verze vyšla na konci října 2014. Rozdíl mezi HTML4 a HTML5 je velký, odpovídá tomu i časový horizont 17 let. Jedna z největších “inovací” je přímá podpora přehrávání videí v prohlížeči. [1]

Obrázek 5 - Logo HTML



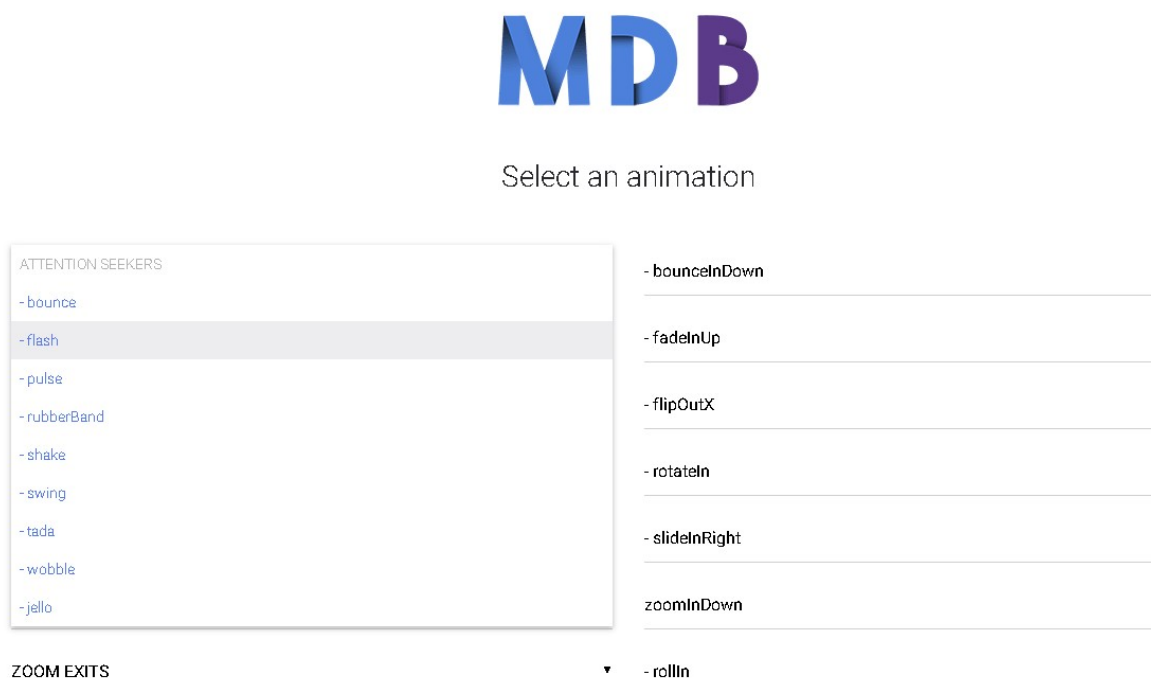
zdroj: https://cs.wikipedia.org/wiki/HyperText_Markup_Language

3.3.7 Bootstrap

Bootstrap obsahuje, ty nejmodernější prvky webových aplikací a stránek. Obsahuje flat design, responzivní design, přístup mobile-first. Bootstrap nabízí přímo své

základní šablony. Existují hotové a designově prvotřídně vypracované. Nejsou to pouze holá těla, ale hotové webové stránky s moderním designem a prvky, které si pak můžeme podle sebe upravit. Jsou pod apache licencí. Lze najít i profesionálnější šablony za zlomek ceny toho, co by stála na míru vytvořená od vývojáře. Google vydal design, který používá ve svých produktech. Vzhled používá stíny, animace, kulaté rohy. Animací je nespočet stačí si vybrat a ten pak implementovat. K použití Bootstrapu je potřeba znalost značkovacích jazyků HTML a CSS. [22] [23]

Obrázek 6 - Logo pohyblivé animce pomocí bootstrapu



zdroj: autor

Zde je několik animací na výběr, kdy si stačí najít své a podle názvu zadat do kódu. Určitá animace, kterou vyberete, se vám přednese na logu MDB, tím zjistíte, co daná animace dělá. Pak dostane webová stránka profesionalitu a žádanost, kterou vyžaduje každý klient, který chce vše na míru.

3.3.8 PHP

Většina aplikací se píše v dnešní době pro web. Webové aplikace mají větší výhodu než desktopové aplikace. Vývoj aplikace pro web je velmi výhodný a najde si více uživatelů než desktopových. Webová aplikace funguje na principu, že se klient poptá na určitý dokument. CGI skript, který pracuje na straně serveru, vygeneruje na stránku to, co klient požaduje. Stránku nalezne na serveru. Za nejpoužívanější CGI skriptovací jazyk se považuje PHP. Většinou to funguje, tak že uživatel zadá csfd.cz/simpsonovi-s12 a odešle požadavek, PHP následně zpracuje požadavek, poté se v našem případě napojí do databáze, ze které vytáhne data, co požadoval klient. Vygeneruje se HTML s určitými daty. Uživatel uvidí stránku, která byla dynamicky zpracována do statické.
[24] [25] [3]

4 Vlastní práce

4.1 Implementace technologií

V této kapitole se budu věnovat postupu a popisu technologií při implementaci do webové stránky. Zde hodně programátorů používá své ověřené zdroje, pro nás to platí totéž. Nejdříve musíme zajistit kam ukládat veškeré data (MySQL). Mnou ověřený softwarový balíček je XAMPP, který podporuje služby, jak MySQL, tak i Apache.

- (X)Webový server – PC, který zodpovídá za vyřizování požadavků HTTP od klientů
- (A)Apache - Podpora několika funkcí, autentizace schémat.
- (M)Databáze – umožňuje manipulovat se souborem a přístup k nim. Propojena mezi sebou pomocí klíči.
- (P)PHP
- (P)Perl

Jsou to především věci, které usnadňují práci programátorům. O technologiích budu mluvit obecně, protože je hodně technologií, které můžeme a nemusíme používat. My si probereme věci ohledně dat, jejich použitelnosti na webu, i když má práce je zaměřena jiným směrem. Budu stručně uvádět příklady využití dané technologie. Můj hlavní programovací jazyk bude PHP objektově orientovaný, který se v dnešním světě bere za jeden z těch „nejhlavnějších“ jazyků. Důvod proč jsem si zvolil tento jazyk, je, že jsem v něm pracoval, takže mám zkušenosti, ale hlavně široký výběr knihoven, které ulehčují práci vývojáři. Vývojové prostředí jsem použil PhpStorm. Nastíníme si něco ohledně javascriptu.

4.1.1 HTML

Vytvoříme si HTML dokument s názvem index.html kam budeme psát jednoduchý kód. Nejdříve si nadefinujeme HTML dokument pomocí `<!DOCTYPE html>`, poté uvedeme párový tag hlavičku neboli head v `<html></html>` s tím, že atributem lang bude s hodnotou „cs“, neboť chceme stránku pro české znaky. Tagem Title, nadefinujeme titulek stránky „Kadeřnictví“. Zavedeme též tag „meta“, který poskytuje informaci o dokumentu, např.: charset. To znamená kódování souboru, máme tam i

name=viewport, což znamená, že v responzivním designu zakáže mobilním zařízením předstírat větší šířku, než ve skutečnosti mají. Máme tam několik odkazujících se na CSS, jako animate.css, waypoints.css, style.css a jeden odkazující se na webovou stránku jQuery. Poté jsou tam implementovány JS prvky pomocí `<script></script>` a ještě máme v hlavičce přímo i style, kde máme některé ID napsané v CSS.

4.1.1.1 Horní menu

Celé horní menu musíme dosadit do příkazu `<nav> </nav>`, ve kterém budou tři vnořené divy, které jsou obohaceny o určité třídy, které vysvětlují v sekci určené pro třídy. V divích máme zaveden příkaz `button` pro vizuální vzhled na mobilních zařízeních, které obohatíme o `spany`, do kterých vložíme `classu icon-bar` pod ní vložíme jednoduchý tag pro linkování, kam vložíme další `class` kvůli firemnímu logu. Kód vypadá následovně:

Tabulka 9 - Horní menu pouze v HTML pro mobily

```
<button>
  <span></span>
  <span></span>
  <span></span>
</button>
<a class="navbar-brand" href="#"></a>
```

zdroj: <https://www.w3newbie.com/tutorials/>

V dalším divu musíme umístit už samotné panely různých stránek, které se budou odkazovat na výstižná místa, která uděláme pomocí odrazek `...`, které jsou implementovány v ``. Tyto odrážky budou obsahovat Domů, Naši umělci, Ceník, O nás, Fotogalerie, Kontakt. Pomocí `href` se budeme moci přepínat mezi stránkami, v našem případě se budeme odkazovat pomocí „BoxXY“, které bude hledat tu stejnou hodnotu ale v atributu ID.

Tabulka 10 - Kód „Horní menu“ pouze v HTML pro desktopy

```
<ol>
  <li><a href="#" >Domů</a></li>
  <li><a href="#">Cenik</a></li>
  <li><a href="#">O nás</a></li>
```

```
<li><a href="#">Fotogalerie</a></li>
<li><a href="#">Kontakt</a></li>
</ol>
```

Zdroj: autor

4.1.1.2 Úvodní strana

V příkazu `<section>...</section>` máme vnořené dva `div` s různými `classy`, ale co chci zmínit jsou dva `section`, které se nám budou starat o animace dvou různých textů. Začneme animací začátečního textu, s tím nám pomůžou podpůrné webové stránky přímo od vývojářů. Stáhnutý dokument dáme do složky CSS. Pomocí těchto příkazů nám CSS umožní pracovat s animací. CSS udělá animaci, která vypadá na první pohled kvalitně. Pomocí příkazu `data-os-animation` můžeme nastavit, jak nám bude text „poskakovat“. Příkaz `data-os-animation-delay` určuje, za jaký čas se provede animace. To zavedeme i u druhého textu v druhém `section` „Chci vědět více“ ve výsledku vypadá kód takto:

Tabulka 11 - Kód „Úvodní stránka“ pouze v HTML

```
<section>
  <div>
    <div>
      <section>
        <h1>Kadeřnictví umělců</h1>
      </section>
      <section >
        <a href="#">Chci vědět více</a>
      </section>
    </div>
  </div>
</section>
```

Zdroj: autor

Pomocí `<h1> </h1>` jsme dosadili „Naší Umělci“.

4.1.1.3 Naši umělci

Použijeme několik vnořených `<div>` kde budou uloženy tagy `` a `<h4>` s jmény

Tabulka 12 - Kód "Naši umělci" pouze v HTML

```
<div>
  <div>
    <div>
      
```

```

        <h4>Adam Martinovský</h4>
    </div>
    <div>
        
        <h4>Veronika Skoupá</h4>
    </div>
    <div>
        
        <h4>Filip Boye-Kofir</h4>
    </div>
</div>
</div>

```

zdroj: autor

Bloky máme tři, `img src="xx.jpg"` nám říká, kde jsou uloženy firemní fotky, vypadá to jednoduše, ale když se to doladí pomocí ID a Class, tak vizuálně stránka bude velmi dobrá. `<h4>` je nadpis, který jsem použil pro pojmenování zaměstnanců.

Tabulka 13 - Kód pro tlačítka bootstrapu pouze v HTML

```

<ol>
  <li></li>
  <li></li>
  <li></li>
</ol>

```

Zdroj: <https://www.w3newbie.com/tutorials/>

Pomocí tohoto kódu bude bootstrap schopen udělat vizuální efekt.

Tabulka 14 - Zbytek Kódu pro tlačítka bootstrapu pouze v HTML

```

<a>
  <span></span>
  <span>Next</span>
</a>
<a>
  <span></span>
  <span>Previous</span>
</a>

```

Zdroj: <https://www.w3newbie.com/tutorials/>

Tento kód pomocí třídy přidává levé a pravé tlačítko, které uživateli umožňuje manuálně používat slide. Dole na konci kódu už jsou zaráženy do divu jednotlivé aktuality nebo informace o daném webu.

4.1.1.4 O nás

Sekce „O nás“ je složena z několika vnořených `<div></div>`, které mají v sobě několik tříd, které má na starosti bootstrap. Zde je kód:

Tabulka 15 - Kód "O nás" pouze v HTML

```
<div id="Box2">
  <div>
    <div ><h4>O nás</h4>
      <hr>
      <p> XY</p>
    </div>
    <div></div>
  </div>
</div>
<div>
  <div>
    <div>
      <h4>Střih</h4>
      <p>XY</p>
    </div>
    <div>
      
    </div>
  </div>
  <div>
    <h4>Výbava</h4>
    <p>XZ</p>
  </div>
  <div>
    
  </div>
</div>
</div>
```

zdroj: autor

Takto vypadá výsledek kódu na konci. Bez CSS tento kód vypadá málo obsáhle, ale je potřeba si určit a udělat „Layout“.

4.1.1.5 Ceník

Každé kadeřnictví bude chtít, aby klientela mohla vidět na jejich webových stránkách cenu jednotlivého stříhu. Ukážu názorně kód a probereme krok po kroku a jaký důvod vedl k daným tagům.

Tabulka 16 - Kód "Ceník" pouze v HTML

```
<table>
  <thead>
    <tr>
      <th>Strihy</th>
      <th></th>
      <th></th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>dámský stříh</td>
      <td>od 500 Kč</td>
    </tr>
  </tbody>
</table>
```

Zdroj: autor

Abychom dosáhli tohoto výsledku, tak nejprve musíme udělat nadpis pomocí `<h4>Ceník</h4>`, který je umístěn nad čarou `<hr>`, která dodává vizuálně kvalitní vzhled, musíme použít atribut `align` a dosadit hodnotu "left", aby to bylo hezky uspořádáno na levou stranu. Dalším krokem je tabulka neboli `<table></table>`. Začneme hlavičkou tabulky pomocí `<thead></thead>` kam vložíme tento kód.

Tabulka 17 - Kus kódu horní tabulky

```
<tr>
  <th>Strihy</th>
  <th></th>
  <th></th>
  <th></th>
</tr>
```

Zdroj: autor

<tr> nám pomůže udělat table row, což znamená řádek tabulky, kam vložíme <th></th> tagy, ale jelikož máme pouze jeden, tak vyplníme pouze Střihy, neboť nebudou už žádné hlavičky. Pod tag </thead> vložíme <tbody></tbody>, což znamená table body, kam vložíme už samotné hodnoty tabulky. Podle obrázku bude šest <tr></tr> tagů a dvanáct <td></td> tagů, kdy vyplníme podle obrázku a zakončíme čarou, která je doplněná o třídu.

Obrázek 7 - Kód v PhpStorm

```
<tr>
  <td>dámský střih</td>
  <td>od 689 Kč</td>
</tr>
<tr>
  <td>pánský střih</td>
  <td>od 389 Kč</td>
</tr>
```

zdroj: autor

4.1.1.6 Fotogalerie

Jedna z posledních stránek je fotogalerie, kde budeme muset udělat opět několik divu, na které pak implikujeme třídy. Pomocí tagu <a>, do kterého vložíme příkaz tam kde jsou XY vložíme cestu uložených obrázků, kód vypadá po napsání takto:

Tabulka 18 - Kód "Fotogalerie" pouze v HTML

```
<div id="Box4">
  <div>
    <div>
      <div>
        <a href="XY/XY.jpg">
          
        </a>
      </div>
      <div>
        <a href="XZ/XZ.jpg">
          
        </a>
      </div>
    </div>
  </div>
</div>
```

zdroj: autor

Tento kód je volně dostupný na webových serverech.

4.1.1.7 Kontakt

Jedna z posledních částí mé webové stránky je Kontakt, ve kterém jsou opět divy a některé třídy, které řeší bootstrap. Nadpis “Kontakt” jsem vybral s `<h4></h4>`, kde mi tato úroveň přišla optimální. Vyplníme text, který přísluší ke kontaktu, což je místo kadeřnictví, telefonní číslo a e-mail. Nadpis “Kadeřnictví umělců” jsem schválně umístil na střed, aby to bylo vizuálně odděleno od e-mailu a telefonního čísla.

Tabulka 19 - Kód "Kontakt" pouze v HTML

```
<div>
  <div><h4>Kontakt</h4>
  <hr align="left">
  <p>Číslo pro rezervaci volejte: +420 739 622 210<br>
  Email: boyekofi@cezemedelskauniverzita.czu.cz</p>
  <h2>Kadeřnictví Umělců</h2>
  <p style="text-align: center">
  Taškent<br>
  Plukovníka Mráze 1182/24<br>
  102 00 Praha 10</p>
</div>
</div>
```

zdroj: autor

A pak je jednoduchý table s názvem “Otevírací doba”, který jsem vyplnil stejným postupem jako u table s ceníkem. Kód vypadá takto:

Tabulka 20 - Kód "Otevírací doba" pouze v HTML

```
<table id="tabletime" cellspacing="5">
  <tbody>
    <tr>
      <td>pondělí</td>
      <td class="sloupecOD">XY - XY</td>
    </tr>
    <tr>
      <td>úterý</td>
      <td class="sloupecOD">XY - XY</td>
    </tr>
    <tr>
      <td>sobota</td>
      <td class="sloupecOD">Objednani pres telefon</td>
    </tr>
    <tr>
      <td>neděle</td>
      <td class="sloupecOD">Objednani pres telefon</td>
    </tr>
  </tbody>
</table>
```

Zdroj: autor

4.1.1.8 Kontaktní formulář

Pomocí kontaktního kalendáře nám mohou zákazníci posílat dotazy, které nebudeme omezovat podmínkami. Zde je holý kód:

Tabulka 21 - Kontaktní formulář pouze v HTML

```
<div >
  <form>
    <p>Jméno: <input placeholder="Jméno" required name="username" class="form-control"></p>
    <p>&nbsp;&nbsp;&nbsp;Email: <input placeholder="E-mailová adresa" type="email" name="email"
required class="form-control" ></p>
    <textarea placeholder="Napište nám dotaz" rows="10" required name="m" class="form-control"
></textarea>
    <br>
    <p><input " type="submit" value="Odeslat dotaz" name="mybtn" class="btn btn-default" >
</span></p>
  </form>
</div>
```

zdroj: <http://www.developphp.com/video/JavaScript>

Placeholder znamená, co se objeví za textaci v tlačítku. V našem případě to bude tedy „Jméno“, „ E-mailová adresa“, „ Napište nám dotaz“ a u type submit to bude určovat atribut value s hodnotou „Odeslat dotaz“. Máme nadefinovaný přímý styl width: 200px; a třídu pomocí Bootstrap vložíme k určitým prvkům. class="form-control" použijí pro první tři inputy a pro tlačítko submit použijí třídu „ class="btn btn-default“.

4.1.1.9 Děkovná stránka v HTML

Je velice jednoduchá stránka, která se zobrazuje poté, co uživatel zadá úspěšně dotaz.

Kód vypadá takto:

Tabulka 22 - Děkovná stránka pouze v HTML

```
<body>
  <div class="jumbotron text-xs-center">
    <h1 class="display-3">Děkujeme za váš dotaz</h1>
    <p class="lead"><strong>Brzy se vám ozveme</strong> </p>
    <hr>
    <p class="lead">
      <a class="btn btn-primary btn-sm" href="index.php" role="button">Jít zpět</a>
    </p>
  </div>
</body>
```

zdroj: <https://codepen.io/JacobLett/pen/vyegPV>

4.1.2 Technologie v praxi

V této kapitole budu popisovat veškeré náležitosti ohledně stylu v mém projektu, který je nezbytný pro tvorbu webových stránek. Představím a popíši různé typy stylů a implementaci, kterou v kódu můžete vidět. Díky bootstrapu od společnosti twitter a mnoho dalších technologií pomáhá programátorům soustředit se na specifické detaily. Jak napovídá název kapitoly, budeme zde probírat vše o CSS, JavaScript, jQuery a bootstrapu. Některé classy jsou nadefinovány rovnou v technologii, takže classy nebudu nijak popisovat. Nejdříve ukáži obecné, to bude úvodní strana, tělo a html

stránka. Začneme nejdříve třídou, která bude působit na HTML a Body. Jedná se o margin, padding, height, width a background-color

Tabulka 23 - Body v CSS

```
html, body {  
    margin: 0;  
    padding: 0;  
    height: 100%;  
    width: 100%;  
    background-color: rgb(144, 186, 205);  
}
```

Zdroj: autor

Margin a padding jsem zvolil proto, aby okraje nebyly žádné a k tomu, aby stránka byla přes celý monitor. Používám width a height na sto procent a samozřejmě background-color je pro barvu mého pozadí, zde jsem si vybral oceánově modrou barvu.

4.1.2.1 Horní menu pomocí tříd

Zde jsou jednoduché styly, které řeší zarovnání, okraje a barvy.

Tabulka 24 - Třída pro horní menu v CSS

```
.navbar-brand {  
    float: left;  
    min-height: 55%;  
    padding: 0 15px 5px;  
}
```

zdroj: <https://www.w3newbie.com/tutorials/>

Jsou tam i mimo jiné další styly, které jsou uloženy v bootstrapu.

Tabulka 25 - Třída pro změnu barev v horním menu v CSS

```
.navbar-inverse .navbar-nav .active a, .navbar-inverse .navbar-nav .active a:focus, .navbar-inverse .navbar-nav .active a:hover {  
    color: #fff;  
    background-color: rgb(144, 186, 205);  
}
```

zdroj: <https://www.w3newbie.com/tutorials/>

Tabulka 26 - Třída pro horní menu pomocí stylu

```
.navbar-inverse .navbar-nav li a {  
    color: #d5d5d5;  
}
```

zdroj: <https://www.w3newbie.com/tutorials/>

4.1.2.2 Úvodní strana pomocí technologií

Budu popisovat třídu, která působí na úvodní obrázek. Třída, která vyplňuje přední stránku se jmenuje úvod. Width, height jsou opět na sto procent s tím, že background color už v tuto chvíli nechceme, ale požadujeme background-img a aby se nám tento obrázek neopakoval, tak napíšeme další příkaz. Změníme to jen na background-repeat s hodnotou no-repeat. Kód pak vypadá takto:

Tabulka 27 - Třída pro úvod

```
uvod{  
    margin: auto;  
    height: 100%;  
    width: 100%;  
    background-image: url("pozadi.jpg");  
    background-repeat: no-repeat;  
    display: table;  
    top: 0;  
    background-size: cover;  
}
```

zdroj: autor

Po tomto kódu je nutné vyřešit animaci skákajícího textu „Kadernictví umělců“. Pro formátování animace jsem použil třídu „animace“, kde definuji šířku, zarovnání s tím, že udělám další třídu, kde určím typ písma, velikost písma a barvu písma. Pod

„Kadeřnictvím umělců“ je ještě jedna animace a to „Chci vědět více“, která se už chová jako link a na ni též máme třídu

```
.tlacitko-start
```

kde opět definuji typ písma, velikost písma, barvu písma,

```
text-transform: uppercase;
```

nám říká, že jakákoliv hodnota bude automaticky přepsána na velká písmena,

```
text-decoration: none;
```

tento příkaz znamená, že text „Chci vědět více“ který je současně link nebude podtržen, neboť jsou takto podtrženy všechny linky.

```
border: solid #73BAD1;
```

zde jsem u rámečku zvolil tuto barvu s hodnotou solid a v neposlední řadě tu máme příkaz

```
transition: all 0.5s;
```

kteřá rozhoduje, jak dlouho bude trvat zbarvení určité barvy na jinou barvu. Poslední třída, která působí na úvodní text je pseudotřída se jménem tlacitko-start: hover. Hover znamená přejetí myši na link. V této pseudotřídě je uložena barva, dekora a rámeček.

Tabulka 28 - Třída pro tlačítko

```
tlacitko-start: hover {  
  color: #CBDFD6;  
  border: solid #CBDFD6 3px;  
  text-decoration: none;  
}
```

zdroj: autor

Obrázek 8 - Výsledek CSS



zdroj: autor

Responzivitu tohoto textu popřípadě animace uděláme pomoci @media screen, zde si ukážeme kus kódu pro zobrazení stránky se šířkou 900px

Tabulka 29 - Responzivní design

```
@media screen and (max-width: 900px) {  
  .animace {  
    padding-bottom: 30%;  
  }  
  .animace h1 {  
    font-size: 450%;  
  }  
  .tlacitko-start {  
    font-size: 130%;  
    padding: 9px 17px;  
  }  
}
```

zdroj: <https://www.w3newbie.com/tutorials/>

Nyní se dostáváme k samotné animaci. Pokud budeme chtít použít animaci, budeme muset zadat třídu jménem.

Tabulka 30 - Třída bootstrapu

```
class="os-animation"
```

zdroj: <https://www.w3newbie.com/tutorials/>

A jak už jsem uváděl, tak na stránce (viz. Kapitola 1.1.5) si vybereme animaci. Já jsem zvolil ze zdola, která skočí nahoru.

Tabulka 31 - Kód pro animaci

```
data-os-animation="bounceInUp"
```

zdroj: <https://www.w3newbie.com/tutorials/>

4.1.2.3 Naší umělci pomocí technologií

Pomocí `<div>` a `` jsme udělali tři kolečka pomocí kterých budeme překlikávat na jiné umělce, ale aby to fungovalo musíme je na sebe odkazovat. „Carousels“ vyžaduje ID. Carousel nám udělá vizuálně krásná tlačítka pomocí, kterých můžeme proklikávat mezi kadeřníky příkazem „data-ride="carousel"“. Tento efekt musíme vizuálně sladit s ostatními prvky, proto musíme použít příkaz „style="padding-bottom: XYpx“ a v neposlední radě to musíme dosadit do ` .. ` a `..`. `data-slide-to="XY"`. To nám umožní, aby tlačítka přesměrovala obsah podle specifického kliku. V našem případě jsme ID pojmenovali „Slide“. Zde jsou náležitě kusy kódu:

Tabulka 32 - Animace při načítání stránky

```
data-ride="carousel"
```

zdroj: <https://www.w3newbie.com/tutorials/>

Říkáme bootstrapu, aby udělal („připravil“) kolečka, když se stránka načítá

Tabulka 33 - Přesměrování

```
data-slide-to="2"
```

zdroj: <https://www.w3newbie.com/tutorials/>

Specifikujeme, na který slide jít, když kliknem na konkrétní tlačítko.

Tabulka 34 - Manuální proklikávání

```
class="left carousel-control" data-slide="prev"  
class="right carousel-control" data-slide="next"
```

zdroj: <https://www.w3newbie.com/tutorials/>

Tento kód umožňuje uživateli měnit mezi stránkami manuálně.

4.1.2.4 Oddělovací čára pomocí technologií

Aby stránka nebyla vizuálně „splácaná“, tak použijeme čáru, která bude sloužit k rozdělení jednotlivých částí webové stránky. Pomocí tříd lze udělat z jednoduché čáry v celku kvalitní design. Na začátek v HTML uděláme tento kód.

Tabulka 35 - Implementace třídy pro čáru v HTML

```
<p class="cara"></p>
```

zdroj: autor

Kdy v třídě „cara“ mám nadefinovaných několik hodnot.

Tabulka 36 - Třída pro čáru

```
.cara {  
    display: block;  
    width: 99%;  
    height: 20px;  
    background-color: black;  
    transform: rotate(-.9deg);  
    margin: 100px auto;  
}
```

Zdroj: autor

Zde jsem zvolil display s hodnotou block, neboť jsem chtěl čáru jako prvek, aby se zobrazoval s řádkovým zalomením. Šířku jsem nadefinoval devadesát devět s výškou pouhých 20px, aby čára nebyla příliš velká a široká. Barvu jsem zvolil černou, která ladí s textem. Transform: rotate definuje míru zakřivení.

4.1.2.5 O nás v Cascading Style Sheet

Prvnímu odstavci jsem navolil zarovnání doleva. Pomocí přímého stylu jsem nadefinoval šířku čáry a umístění čáry. též třídu „img-responsive“. Odstavec

jsem vygeneroval pomocí lorem ipsum generator náhodný text. Výsledek tohoto kódu je zde.

Tabulka 37 - Třídy v sekci "O nás"

```
<div class="container" id="Box2">
  <div class="row">
    <div class="col-md-6"><h4 style="text-align: left">O nás</h4>
      <hr style="width:300px" align="left">
      <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quis autem vel eum iure
reprehenderit
      qui in
      </p>
    </div>
    <div class="col-md-6">
    </div>
  </div>
</div>
```

zdroj: <https://www.w3newbie.com/tutorials/>

Všechny tyto třídy obsahují bootstrap, takže jsem jen implementoval podle své potřeby. Takto jednotlivě jsem to udělal pro tři sekce.

4.1.2.6 Ceník pomocí technologií

Ceník jsem udělal pomocí nadpisu H4, kde jsem udělal i přímý styl.

Tabulka 38 - Přímý styl

```
style="text-align: left"
```

zdroj: autor

Pomocí bootstrapu jsem zvolil třídu, která naformátuje celý table, zde je příklad

Tabulka 39 - Třída v "Ceník"

```
<table class="table table-dark table-hover" style="font-size:14px">
  <thead>
    <tr>
      <th>Strihy</th>
      <th></th>
      <th></th>
      <th></th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>dámský střih</td>
      <td>od 689 Kč</td>
    </tr>
  </tbody>
</table>
```

zdroj: https://www.w3schools.com/bootstrap4/bootstrap_tables.asp

Výsledek práce je tento a opět jsem použil třídu pro čáru, aby zvýraznila mezeru mezi sekcemi.

4.1.2.7 Fotogalerie pomocí technologií

Díky tomu, že jsou zdarma volně šířitelné zdrojové kódy tak jsem použil jeden, který používám v práci často neboť je designově vyřešen ve všech směrech.

Tabulka 40 - Třídy pro sekci "Fotogalerie"

```
<div class="container gallery-container">
  <div class="tz-gallery">
    <div class="row">
      <div class="col-sm-12 col-md-4">
        <a class="lightbox" href="XZ/XZ.jpg">
          
        </a>
      </div>
    </div>
  </div>
</div>
```

zdroj: <https://tutorialzine.com/2017/02/freebie-4-bootstrap-galleries>

Ve třídách jsou uvedeny hodnoty, jak se má zobrazovat obrázek po najetí a dalších stylistických funkcích, které jsem již vysvětloval v minulých kapitolách.

4.1.2.8 Kontakt

Zde je jen několik tříd a jen jednu jsem poupravil a to tak, aby dny měly větší mezeru od časů.

Tabulka 41 - Třída pro okraje textu

```
.sloupecOD {  
    padding-left: 120px;  
}
```

zdroj: autor

Grafické úpravy jsem udělal pomocí ID „openingtime“ a vypadají nějak takto:

Tabulka 42 - Třída pro vizuální prvky v otevírací době

```
#openingtime {  
    position: relative;  
    padding: 50px;  
    float: right;  
    color: white;  
    /* asi smazu  
    margin-top: -50px;  
    */  
    background-color: lightgrey;  
    background-size: 100%;  
    border-radius: 15%;  
}
```

zdroj: autor

Všechny příkazy jsem už sepisoval v minulých kapitolách.

4.1.2.9 Scrolling pomocí JavaScriptu

Zde si popíšeme JavaScript, který nám udělá to, že pokud klikneme v horním menu na dané tlačítko, tak JS plynule přejede na danou část. Tento efekt dodá webové stránce dojem moderního webu. Nejdříve si udělám `<script></script>` a napíši, kdy náš dokument bude připraven. Budeme chtít, aby udělal naší funkci. Uvnitř funkce

uděláme proměnnou scrollLink. Teď budeme po kliknutí na scrollLink požadovat, aby se něco stalo. Script se bude aplikovat na body a html, kde použijeme animace, což nám dovolí plynule sjíždět. Dále je tam příkaz ohledně toho, kolik sekund bude mít na to plynule přejet. Poslední část JS kódu je jedna z jednoduchých věcí, kdy horní menu nemění aktivní položku, takže pokud jsme v části o ceníku, tak stále v menu bude zvýrazněna složka „domů“ a pomocí if v JS zajišťuje aktivní část a tu předchozí smaže. Zde je tento kód:

Tabulka 43 - Scrollování

```
$(document).ready(function () {
    var scrollLink = $('scroll');
    scrollLink.click(function (e) {
        e.preventDefault();
        $('body,html').animate({
            scrollTop: $(this.hash).offset().top
        }, 1000);
    });
    $(window).scroll(function () {
        var scrollbarLocation = $(this).scrollTop();
        scrollLink.each(function () {
            var sectionOffset = $(this.hash).offset().top - 20;
            if (sectionOffset <= scrollbarLocation) {
                $(this).parent().addClass('active');
                $(this).parent().siblings().removeClass('active');
            }
        })
    })
})
```

zdroj: <https://codepen.io/kevinpowell/pen/dWzGox>

4.1.2.10 Dotazník pomocí PHP a MySQL

Pomocí PHP vytvoříme formulář, který bude ukládat dotazy pomocí MySQL. Zde je kód, který jsem použil a používám:

Tabulka 44 - PHP kód pro zpracování dat

```
<?php
$con= mysqli_connect('localhost','root','password');
if(!$con)
{
    echo 'No connect';
}
```

```
if(!mysqli_select_db($con,'tutorial')){
    echo'No data';
}
$email= $_POST['email'];
$name= $_POST['username'];
$comment= $_POST['comment'];
$sql = "INSERT INTO person (Name,Email,Comment)VALUES ('$name','$email','$comment)";
if (!mysqli_query($con,$sql)){
    echo'error';
}
else {
    header("refresh:0; url=thanks.html");
}
```

zdroj: autor

Kde propojím databázi a následně si určím proměnné, do kterých se ukládají hodnoty napsané ve formuláři. HTML vložíme do form “insert.php” a pomocí “name” se budou moci ukládat hodnoty do databáze. Poté co se úspěšně zapíše do database, PHP kód nasměruje na děkovnou stránku “thanks.html”.

5 Závěr

V mé práci jsou stručně popsány nejpoužívanější webové technologie, které souvisí s tvorbou webových stránek. Důležitým přínosem této práce je vysvětlení a seznámení s tímto druhem tvorby internetových stránek.

Na závěr bych rád uvedl, že informací k danému tématu je mnoho. Některé informace se dají najít v knihovně, ale asi většina a velké množství je na internetu. Vždy jsem našel, co jsem potřeboval ke své práci. Díky této práci se mi podařilo prohloubit znalosti. Cílem práce bylo vytvořit moderní dynamické webové stránky, dle mého názoru se mi tento cíl podařil uskutečnit, i když nadále může dojít k jejich zdokonalování, například rozšíření o aplikaci objednávání.

6 Seznam použitých zdrojů

6.1 Monografické zdroje

- [1] HOGAN, Brian P. HTML5 a CSS3: výukový kurz webového vývojáře. Brno: Computer Press, 2011. ISBN 978-80-251-3576-1.
- [2] RESIG, John. JavaScript a Ajax: moderní programování webových aplikací. Brno: Computer Press, 2007. ISBN 978-80-251-1824-5.
- [3] VRÁNA, Jakub. 1001 tipů a triků pro PHP. Brno: Computer Press, 2010. ISBN 978-80-251-2940-1.

6.2 Internetové zdroje

- [4] WordPress [online]. 2018 [cit. 19. 2. 2018]. Dostupné z: <https://cs.wikipedia.org/wiki/WordPress>
- [5] STUDENIČ Adam. *Webová aplikace pro správu dotazů uživatelů webové stránky*. Brno, 2014. Bakalářská práce. Masarykova univerzita. Fakulta informatiky.
- [6] Marc Clifton, Framework [online]. 2003 [cit. 19. 2. 2018]. Dostupné z: <https://www.codeproject.com/Articles/5381/What-Is-A-Framework>
- [7] Framework Web [online]. 2018 [cit. 19. 2. 2018]. Dostupné z: https://en.wikipedia.org/wiki/Web_framework
- [8] Petr Daněk, Velký test PHP frameworků [online]. 2014 [cit. 19. 2. 2018]. Dostupné z: <https://www.itnetwork.cz/php/knihovny/php-tutorial-uvod-do-knihoven-a-frameworku>
- [9] Laravel documentation. [Online]. 2018 [Citace: 7. 3. 2018.]. Dostupné z: <https://laravel.com/docs/5.6>
- [10] Laravel. [Online]. 2018 [Citace: 7. 3. 2018.]. Dostupné z: <https://cs.wikipedia.org/wiki/Laravel>
- [11] Documentation [online]. 2018 [cit. 19. 2. 2018]. Dostupné z: <https://doc.nette.org/cs/2.4/>
- [11] MySQL [online]. 2018 [cit. 19. 2. 2018]. Dostupné z: <http://www.oracle.com/technetwork/database/mysql/index.html>

- [12] David Čápka, Úvod do MySQL a příprava prostředí [online]. 2013 [cit. 19. 2. 2018]. Dostupné z: <https://www.itnetwork.cz/mysql/mysql-tutorial-uvod-a-priprava-prostredi>
- [13] jQuery API. [Online]. 2018 [Citace: 7. 3. 2018.]. Dostupné z: <http://api.jquery.com/>
- [14] David Čápka, Úvod do jQuery. [Online]. 2014 [Citace: 7. 3. 2018.]. Dostupné z: <https://www.itnetwork.cz/javascript/jquery-zaklady/javascript-tutorial-funkcionalni-programovani-a-jquery-webova-kalkulacka>
- [15] Ajax technology. w3schools. [Online]. 2018 [Citace: 7. 3. 2018.]. Dostupné z: https://www.w3schools.com/js/js_ajax_intro.asp
- [16] Ajax. [Online]. 2017 [Citace: 7. 3. 2018.]. Dostupné z: <https://cs.wikipedia.org/wiki/AJAX>
- [17] David Čápka, Úvod do JavaScriptu [online]. 2013 [cit. 19. 2. 2018]. Dostupné z: <https://www.itnetwork.cz/javascript/zaklady/javascript-tutorial-uvod-do-javascriptu-nepochopeny-jazyk>
- [18] Honza Bittner, Mobile first responzivní design [online]. 2015 [cit. 19. 2. 2018]. Dostupné z: <https://www.itnetwork.cz/html-css/responzivni-webdesign/mobile-first-responzivni-design>
- [19] Responsive Web Design. [Online]. 2018 [Citace: 7. 3. 2018.] Dostupné z: <https://developers.google.com/web/fundamentals/design-and-ux/responsive/>
- [20] Dušan Janovský, CSS [online]. 2017 [cit. 19. 2. 2018]. Dostupné z: <https://www.jakpsatweb.cz/css/>
- [21] Dušan Janovský, HTML příručka [online]. 2018 [cit. 19. 2. 2018]. Dostupné z: <https://www.jakpsatweb.cz/html/>
- [22] David Čápka, CSS frameworku Bootstrap [online]. 2017 [cit. 19. 2. 2018]. Dostupné z: <https://www.itnetwork.cz/html-css/bootstrap/uvod-do-css-frameworku-bootstrap>
- [23] Introduction [online]. 2018 [cit. 19. 2. 2018]. Dostupné z: <https://getbootstrap.com/docs/4.0/getting-started/introduction/>
- [24] David Čápka, Úvod do PHP a webových aplikací [online]. 2015 [cit. 19. 2. 2018]. Dostupné z: <https://www.itnetwork.cz/php/zaklady/php-tutorial-uvod-do-webovych-aplikaci>

[25] Dušan Janovský, PHP -- Jak začít [online]. 2018 [cit. 19. 2. 2018]. Dostupné z:
<https://www.jakpsatweb.cz/php/jak-zacit.html>

7 Přílohy

Příloha č. 1: CD s přiloženými soubory webové stránky a databáze