
MATURITNÍ PRÁCE

Internetové stránky

Jméno: Václav Štrobl

Školní rok: 2015 / 2016

Třída: ME4

Zadání maturitní práce

Téma maturitní práce: Vytvořte internetové stránky s vlastním námětem

Maturant: Václav Štrobl

Vedoucí práce: Jiří Stibor

Konzultant: Pavel Tatar

Pokyny pro vypracování:

- vytvořte internetové stránky na vlastní téma
- vysvětlete použitou technologii
- jednoznačně definujte, které části jsou originálním dílem
- využívejte standardní knihovny programovacích jazyků

Výsledky práce zpracujte do textu o délce 10 až 15 stran. Zprávu odevzdejte ve dvou tištěných exemplářích a v elektronické podobě.

Datum odevzdání maturitní práce: do 14.4.2016 do 12:00 hod.

V Praze dne 15.10.2015

Ing. Marcela Davídková Antošová, CSc.

Ředitelka VOŠ a SŠSE

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil jsem literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze, dne

.....
Podpis



ANOTACE

Účelem této maturitní práce bylo vytvořit internetové stránky na vlastní téma. Jako téma jsem zvolil prezentaci firmy, která se zabývá vývojem webových stránek. K vytvoření těchto internetových stránek byly použity jazyky HTML, CSS a jQuery knihovna pro jazyk JavaScript. Svým rozsahem nejsou tyto webové stránky příliš velké a ani se v budoucnosti nebude dynamicky měnit jejich obsah. Proto jsou vytvořené jako statické webové stránky. Hlavní kód webových stránek je napsaný v programu PSPad. Některé kusy nebo doplňky HTML kódu jsou pouze vloženy. Jedná se o části kódu které slouží pro komunikaci se službami třetích stran jako je například Google Analytics, Facebook nebo Google+. Tyto služby již dodaly vygenerovaný kód, který se pro správnou funkci služeb, do HTML kódu pouze vkládá.

ANNOTATION

The purpose of this graduation work was to create a website at your own topic. As a topic I chose a company presentation, which develops websites. For create these websites were used languages HTML, CSS and jQuery library for JavaScript. These websites are not too big by size and they will not dynamically change the content in future. Therefore, they are formed as a static web pages. Main web site code is written in PSPad. Some pieces or supplements of HTML code are only inserted. This is pieces of code that are used to communicate with third party services like Google Analytics, Facebook or Google+. These services already delivered the generated code, which for proper function of the service, into HTML only inserts.

Obsah

1	Úvod	6
2	Typy webových stránek	6
2.1	Statické stránky	6
2.2	Dynamické stránky	6
2.3	Redakční systémy	7
3.2.1	Základní funkce CMS	7
3.2.2	Přiblížení redakčních systémů	7
2.4	Internetové obchody	8
3	Technologie webových stránek	8
3.1	HTML	9
1.3.1	Vývoj jazyka	9
1.3.2	Popis jazyka	9
1.3.3	Struktura dokumentu	10
1.3.4	Příklad zdrojového kódu	11
1.3.5	Programy pro tvorbu HTML	11
3.2	CSS - kaskádové styly	11
2.3.1	Zápis stylů	12
2.3.2	Propojení CSS a HTML	13
3.3	jQuery	14
3.3.1	Funkce	14
3.3.2	Použití	14
3.4	Další jazyky	16
3.5	Google analytics	17
4	Definice částí webových stránek	17
4.1	HTML kód	17
4.2	CSS kód	18
4.3	jQuery kód	18
4.4	Grafika	19
4.5	Obsah	19
5	Závěr	19
6	Seznam příloh	20
7	Použitá literatura a odkazy	20

1 Úvod

Webové stránky jsou dokumenty, které je možné zobrazit ve webovém prohlížeči. Jsou obvykle umístěny na webovém hostingu (virtuální úložiště na nějakém serveru) a pod konkrétní doménou (url adresa - například: www.domena.cz) jsou přístupné skrz internet. Dají se ale také zobrazit tzv. "offline", přímo ze souborů webových stránek které jsou uloženy v zařízení, v kterém webové stránky chceme zobrazit. Webové stránky jsou tvořeny textem, obrázky, zvuky a dalším obsahem dle typu webových stránek. Účelem webových stránek je především přenášení konkrétních informací.

2 Typy webových stránek

2.1 *Statické stránky*

Statické stránky jsou stránky, které jsou dodány v souborech, v kterých je také uložen jejich obsah. Většinou nejsou tvořeny žádným systémem, ale jsou naprogramovány do finální podoby, v které už zůstávají. Každá podstránka webových stránek má svůj soubor a využívá se technologií HTML a CSS v kombinaci se skripty (například JavaScript). Pro jakoukoli další editaci těchto webových stránek je potřeba programátor, který obsah upravuje ve zdrojovém kódu souborů těchto stránek. Příkladem statických stránek je tato maturitní práce. Jejich výroba většinou bývá jednodušší a méně nákladná, toto však platí pouze do určitého rozsahu stránek. Proto se hodí pouze pro menší webové stránky nebo prezentace kde nebude potřeba často nebo vůbec měnit obsah. Je tedy vhodná především pro prezentaci menších firem či produktů a nebo pro osobní weby.

2.2 *Dynamické stránky*

Dynamické webové stránky jsou stránky, jejichž obsah se generuje z informací uložených např. v databázi. To znamená, že jejich obsah se může libovolně měnit v závislosti na nastavení těchto stránek (můžou se změnit například po zásahu administrátorem, uživatelem, dosáhnutím časové doby nebo kombinací dalších faktorů). V dnešní době se tyto stránky využívají nejvíce. Dynamická stránka může být například blog, sociální síť, webová aplikace nebo hra, větší webové stránky, fóra, informační systémy a další. K tvorbě těchto

webových stránek se dnes využívají především tzv. redakční systémy neboli systémy pro správu obsahu. Výhodou těchto stránek oproti statickým je možnost pojmout obrovské množství dat a ukládání/čtení dat z databáze, jejich obsah si může uživatel sám upravovat a není omezen pouze na malý rozsah webu a programátory. Nevýhodou je však jejich vyšší pořizovací cena a složitost naprogramování takového webu. Hodí se pro jakékoliv větší weby.

2.3 Redakční systémy

Jak bylo zmíněno v předchozí podkapitole, k tvorbě dynamických webových stránek se většinou využívají redakční systémy (CMS, systémy pro správu obsahu).

3.2.1 Základní funkce CMS

Smyslem těchto systémů je umožnit lidem tvořit webové stránky nebo upravovat jejich obsah tak jak potřebují i **bez znalosti programování**.

Základními funkcemi jsou tedy:

- Správa souborů, obrázků nebo galerií
- Modifikace, tvorba nebo publikace textů, dokumentů nebo článků
- Možnost nastavení vzhledu webových stránek

Dále lze do systémů instalovat různé doplňky, které umožňují uživateli upravovat více prvků na webu a celkově si co nejlépe uzpůsobit web ke své představě.

3.2.2 Přiblížení redakčních systémů

Redakční systém můžeme rozdělit na dvě části:

1. Front end

Front-endem se označuje část webových stránek kterou vidí uživatel. Obvykle zde může číst a sledovat webové stránky, které vytvořil administrátor nebo správce webu v Back end části. Vidí tedy výsledný web.

2. Back end

Je to část webu, kde může administrátor web spravovat a upravovat tak jak potřebuje. U většiny redakčních systémů je tato část přístupná pod doménou *www.webovastranka.cz/administrator*, kde se po zadání přihlašovacího jména a hesla dostane správce do administrace webu. Ta je graficky zpracovaná tak, aby správce nemusel umět programovat. Webové stránky tedy upravuje pouze pomocí jednoduchých editorů textu nebo pomocí dalších doplňků.

Dnes existuje již mnoho redakčních systémů, které jsou svým rozsahem, jednoduchostí nebo možnostmi vhodné pro všechny typy webových stránek. Existují buď systémy zdarma anebo placené. Jako nejčastěji používané redakční systémy zde zmíním Wordpress, Joomla a Drupal.

2.4 Internetové obchody

Internetový obchod (e-shop, e-store, elektronický obchod...) je typ webových stránek, které primárně slouží k prodeji produktů nebo služeb. Obvykle jsou založeny na systémech pro správu obsahu, které jsou přímo dělané pro potřeby obchodu. Obchodník si tedy může nejen editovat vzhled webových stránek, ale může si sám vkládat nebo upravovat produkty a jejich detaily. Dnešní systémy pro internetové obchody také umožňují mnohem více možností než jen spravovat produkty obchodu. Umožňují také například sledovat stavy objednávek, sledovat jestli za zboží již zákazník zaplatil, automaticky vytvářejí faktury a zpracovávají platby zákazníků, sledují stav skladů nebo produktů a celý proces prodeje je z velké části automatizovaný. Obchodníkovi tedy odpadá řada starostí, protože ho systém sám na důležité informace upozorní. Jeho prací už je pouze, po objednávce a zaplacení zákazníkem, odeslat zboží, což i v některých případech systém dělá sám (pošle pokyny pro doručení třetí straně, která produkty dodává). Systémy se stejně jako u redakčních systémů dají rozdělit na placené a nep placené. Mezi ty nejčastěji používané patří Magento, WooCommerce nebo Opencart.

3 Technologie webových stránek

Webové stránky se dají tvořit několika způsoby, způsob volíme podle rozsahu a náročnosti webových stránek. V mé této maturitní práci jsem použil technologii psaní webových stránek "v ruce". To znamená, že jsem ke tvorbě webových stránek nepoužil žádný

předem připravený tzv. redakční systém, ale veškerý kód jsem napsal sám v editoru.

K tvorbě webových stránek jsem využíval jazyků HTML, CSS3 a knihovnu jQuery pro zjednodušený zápis jazyku JavaScript. Podívejme se tedy blíže na jednotlivé jazyky a jejich význam při tvorbě webových stránek.

3.1 HTML

HTML neboli HyperText Markup Language je značkovací jazyk který je používán pro tvorbu webových stránek. Je jedním z hlavních jazyků pro vytváření stránek v systému WWW (World Wide Web) díky kterému můžeme publikovat dokumenty na internet.

1.3.1 Vývoj jazyka

V roce 1990, byl Bernersem-Leem navržen jazyk HTML a protokol HTTP (HyperText Transfer Protocol - protokol pro přenos hypertextu) jako zjednodušení v té době již používaných jazyků (TeX, PostScript, SGML). Zároveň také Tim Berners-Lee napsal první webový prohlížeč, který se jmenoval WorldWideWeb. V roce 1993 vznikl prohlížeč Mosaic, který byl k dispozici pro IBM PC a Macintosh. Byl to první prohlížeč s grafickým uživatelským rozhraním. Následoval rychlý rozvoj webu.

Postupně se jazyk po verzích (0.9, 1.2, 2.0, 3.2, 4.0, 4.01, 5) rozvíjel a do jeho možností se přidávala podpora grafických prvků, tabulky, zarovnávání textu a ovlivňování vzhledu, formuláře a další detaily. Verze 4.01 byla vydána 24. prosince 1999 komunitou W3C, poté následovala patnáctiletá odmlka, po které byla vydána nová verze 5.0 (aktuální). Verze 5.0 ukončila závislost jazyka HTML na SGML a přinesla mnoho nových vylepšení a podporu moderních a nových technologií.

1.3.2 Popis jazyka

Jazyk HTML je řešen množinou značek (tagů) a jejich vlastností definovaných pro danou verzi. Tento zápis se postupně vyvíjel a upravoval podle aktuálních trendů internetu. Mezi značky se uzavírají části textu a tím se určuje význam textu. Názvy značek a jejich vlastností se píšou do úhlových závorek (např.: `<a>` nebo ``).

Podle typu značky je nazýváme buď párové, to znamená že značka musí mít začátek a konec, nebo nepárové - značka nemusí být ukončena. Například značka ` Někjaký text ` udělá tlusté písmo u textu který je uvnitř značky. Tato značka je párová takže musí mít počáteční značku a koncovou značku s lomítkem. Značka `` udělá obrázek, jehož umístění je definováno atributem `src`, není párová, takže nemusí být ukončena další stejnou značkou s lomítkem.

Dále mohou značky obsahovat také další důležité atributy.

Například:

```
<a href="http://www.ukazkovyweb.cz" target="_blank" class="nejakatrída"> Text odkazu </a>
```

`<a>` je značka odkazu takže po kliknutí na Text odkazu nás značka odkáže na nějakou adresu která je obsažená v atributu "`href`" - v tomto případě na `http://www.ukazkovyweb.cz`, `target="_blank"` je atribut který prohlížeči řekne, aby odkaz otevřel v novém okně prohlížeče a `class` je název třídy pro tuto značku, která se dá dále použít v jazyku CSS pro grafické nastýlování tohoto odkazu. *Text odkazu* je to co vidí uživatel v prohlížeči a na co také může kliknout, `<a>` je značka párová takže je ukončena značkou ``.

Dále může také dokument obsahovat komentáře (pomocný text pro programátora, který prohlížeč "nevidí"), kódy skriptovacích jazyků a kaskádové styly.

1.3.3 Struktura dokumentu

V jazyku HTML má dokument předepsanou strukturu.

1. Deklarace typu dokumentu - značka `<!DOCTYPE html>`, řekne prohlížeči že otevřel HTML dokument, často se do této deklarace také píše verze HTML dokumentu.
2. Hlavní element - značka `html` (`<html>` a `</html>`), uzavírá celý dokument.
3. Hlavička dokumentu – značka `head` (`<head>` a `</head>`), obsahuje metadata, která platí pro celý dokument. V hlavičce je uvedeno například kódování, autor, popis webu, klíčová slova, titulek, název dokumentu a další.
4. Tělo dokumentu – značka `body` (`<body>` a `</body>`), v této značce jsou uzavřeny všechny značky obsahu dokumentu, které může uživatel vidět v prohlížeči.

1.3.4 Příklad zdrojového kódu

Příklad struktury velice jednoduchého HTML kódu:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<!-- v předchozím řádku je definované kódování dokumentu a toto je komentář -->
<title>Titulek </title>
</head>
<body>
<h1> Nadpis </h1>
<div> <p> Text v blokovém elementu div </p> </div>
</body>
</html>
```

1.3.5 Programy pro tvorbu HTML

Programy pomocí kterých tvoříme HTML kód se nazývají editory.

Editory můžeme rozlišit na 2 druhy:

Textové editory - HTML kód můžeme psát například i v poznámkovém bloku (notepad), dnes ale už existují programy speciálně uzpůsobené pro psaní kódu. Například PSPad (byl použit k tvorbě této maturitní práce). Takové editory odlišují například značky a atributy jinou barvou než je klasický text, takže je snazší se v kódu orientovat.

WYSIWYG editory - WYSIWYG (What you see is what you get - To co vidíš je to, co dostaneš). Jsou to editory, které pracují s grafickou předlohou webu. K jejich užívání většinou uživatel nemusí znát HTML kód, ale poskládá si grafickou podobu webové stránky z obrázků či grafických prvků a editor mu poté vygeneruje dokument ve formátu HTML. Takovým editorem je například Adobe Dreamweaver.

3.2 CSS - kaskádové styly

Kaskádové styly (v originále Cascading Style Sheets, proto zkratka CSS), jsou jazyk kterým se definuje vzhled jednotlivých elementů napsaných v jazyce HTML. Byly vydány verze CSS1, CSS2 a CSS3.

Některé příklady co můžeme pomocí CSS dělat:

Uvádím pouze pár příkladu, jelikož kompletní seznam by byl delší, než je nezbytně nutné:

- Nastavit barvu nebo velikost písma
- Nastavit šířku nebo výšku obrázků
- Nastavit formátování textu
- Odsazení odstavců
- Nastavit pozadí čehokoliv
- V kombinaci se skripty může i animovat části stránek
- Nastavit velikosti bloků, tabulek
- A mnoho dalšího....

2.3.1 Zápis stylů

Každý styl má svá pravidla, podle kterých musí být zapsaný. Pravidla obsahují selektory a bloky deklarácí. Každý blok deklarácí obsahuje deklarace samotné, které jsou oddělené středníky " ; " a každá deklarace má vlastnost, poté dvojtečku " : " a hodnotu vlastnosti. Takže samotná deklarace vypadá takto: *vlastnost:hodnota*;

Ukážeme si to na příkladu:

```
.styl1 {  
color: white; // určuje, že html značka s třídou "styl1" bude mít bílou barvu  
position: absolute; // určuje, že html značka s třídou "styl1" bude mít absolutní typ  
pozicování  
left: 0px; // určuje, že html značka s třídou "styl1" bude umístěna 0px od levého kraje  
prohlížeče  
}
```

".styl1" je selektor, neboli název třídy pro kterou budou platit deklarace které jsou uvnitř závorek, **"color: white;"** je už samotná deklarace, **"color"** je vlastnost a **"white"** je její hodnota.

Selektory

CSS má více selektorů neboli možností zápisu či pravidel pro zápis, která se dají kombinovat. Zde jsou ty základní:

div – Tato deklarace bude platit pro všechny elementy `<div>` obsažené v dokumentu HTML.

div a – Tato deklarace bude platit pro všechny elementy `<a>`, které se nacházejí v elementech `<div>`. Je jedno v které hloubce dokumentu.

body>div – Tato deklarace bude platit pro všechny elementy `<div>`, které jsou o úroveň níže

než element `<body>` (neboli jsou dětmi elementu `body`).

.nejakatrída – Tato deklarace bude platit pro všechny elementy, které mají v HTML dokumentu nastavenou třídu "**nejakatrída**". To se provádí v HTML dokumentu pomocí atributu "**class**".

Deklarace atributu *class*: ` Někáký odkaz `

#id – Tato deklarace bude platit pro všechny elementy, které mají v HTML dokumentu nastavený atribut `id` s hodnotou **nejakeid**.

Deklarace `id` atributu: `<p id="nejakeid"> Někáký text </p>`

:link, **:visited**, **:hover** a **:active** – Tyto deklarace budou po řadě platit pro dosud nenavštívené odkazy; navštívené odkazy; část textu, na kterém stojí ukazatel myši a aktivní odkaz.

2.3.2 Propojení CSS a HTML

Existuje více možností jak použít kaskádové styly v HTML dokumentu. Nejčastěji se používá samostatný CSS soubor. Tento soubor má v sobě napsaná pravidla pro html elementy a má koncovku `.css`. Jeho vložení do HTML dokumentu probíhá skrz vložení `link` elementu.

Například:

```
<head>
  <link rel="stylesheet" type="text/css" href="css/index.css" >
</head>
```

Dále můžeme zapsat kaskádový styl do elementu `<style>` v HTML dokumentu:

```
<style type="text/css">

.trída{
position :relative;
top: 0px;
}

</style>
```

Další z možností je zapsat styl do atributu značky v HTML dokumentu, příklad:

```
<a style="color: black;"> Někáký odkaz </a>
```

3.3 jQuery

jQuery je rychlá, malá a na funkce bohatá knihovna pro JavaScript. Dělá úkony, jako jsou například manipulace a změny v elementech HTML dokumentu, zpracovává události, vytváří animace a funguje napříč drtivou většinou prohlížečů. Díky kombinaci všestrannosti a rozšiřitelnosti, jQuery změnila způsob jakým lidé zapisovaly JavaScript. Jejím motem je "write less, do more" (napiš méně, udělej více). Umožňuje především zapisovat kód javascriptu mnohem snazším a kratším způsobem. Vznikla v lednu 2006 a je to otevřený software pod licencí MIT.

3.3.1 Funkce

jQuery nabízí například následující funkce:

- Funkce pro procházení a změnu DOM (*objektový model dokumentu, API umožňující přístup či modifikaci obsahu, struktury, nebo stylu dokumentu, či jeho částí.*)
- Události
- Manipulace s CSS
- Efekty a animace
- AJAX - (*Asynchronous JavaScript and XML, obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich kompletního znovunačítání za pomoci asynchronního zpracování webových stránek pomocí knihovny napsané v JavaScriptu.*)
- Rozšiřitelnost

3.3.2 Použití

jQuery je většinou šířena jako jeden javascriptový soubor který je načten do HTML dokumentu.

Můžeme ho vložit přímo ze složky s webem:

```
<script type="application/javascript" src="js/jquery.js"></script>
```

Další možností, která je velice oblíbená, je načtení jQuery knihovny přímo ze serverů Google:

```
<script  
type="application/javascript"  
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js">  
</script>
```

jQuery lze také načíst v aktuální verzi přímo z domovské stránky projektu:

```
<script  
type="application/javascript" src="http://code.jquery.com/jquery-latest.min.js">  
</script>
```

Vložení jQuery do HTML dokumentu nám dovolí knihovnu používat, můžeme se tedy pustit k samotné práci s knihovnou.

Pro příklad uvedu pouze některé kódy, které popíšu, další syntaxe a práce s jQuery je popsána na jejích webových stránkách <https://jquery.com/>.

Začněme tedy jednoduchou syntaxí jQuery:

```
$("# prvek, s kterým budeme něco dělat ").funkce(" parametry funkce ");
```

Nejjednodušší syntaxe, prvek s kterým něco chceme udělat můžeme podobně jako v kaskádových stylech zapsat pomocí třídy a nebo id.

Zde je jQuery v porovnání s javascriptem:

```
$("# prvek ").funkce(" parametry "); - jQuery
```

```
document.getElementById(" prvek ").funkce(" parametry "); - JavaScript
```

V porovnání je tedy vidět že jQuery umožňuje zápis stejné syntaxe, v porovnání v JavaScriptem, způsobem který je dvakrát až třikrát kratší.

Pro použití jQuery musíme před syntaxi zapsat buď znak \$ nebo slovo jQuery, obojí má ten samý význam, je tedy jedno jestli si zvolíme první nebo druhou možnost. Zápis by vypadal takto:

```
$("# prvek ").funkce(" parametry "); - možnost s $
```

```
jQuery(" prvek ").funkce(" parametry "); - možnost s jQuery
```

Oba řádky dělají to samé.

Dále můžeme pomocí jQuery řetězit funkce takže je nemusíme vypisovat všechny jednotlivě:

```
$("# prvek ").funkce1(" parametry ").funkce2(" parametry ").funkce3(" parametry ");
```

Dále uvedu příklad jednoho z kódů webových stránek této maturitní práce:

```
<script>
```

```
$(".zavritformuspech1").click(function() {
```

```
/* Po kliknutí myši na element s třídou "zavritformsupech1" udělá funkci. To co je uvnitř složených závorek je funkce */
```

```
    $(".tmavepozadi").hide();
```

```
/* V kaskádovém stylu nastaví elementu s třídou "tmavepozadi" vlastnost "display" na hodnotu "none", to má za následek schování elementu tak že ho uživatel nevidí */
```

```
});
```

```
</script>
```

3.4 Další jazyky

Pro tvorbu webových stránek se také dají použít další jazyky. Například jazyk PHP. Využívá se ve složitějších webových stránkách. Například pro komunikaci s MySQL databází. Dále existuje ještě spousta dalších knihoven podobných například výše zmíněné jQuery, většinou se jedná o zjednodušení některých hlavních skriptovacích jazyků.

3.5 *Google analytics*

Google analytics, je nástroj pro měření dat na webových stránkách od společnosti Google. Umožňuje získávat různá data od návštěvníků webu jako je jejich chování, používaná zařízení, používané prohlížeče a další. Tyto informace následně pomáhají tvůrcům webových stránek vylepšovat webové stránky tak, aby byly pro návštěvníka co nejlepší na prohlížení. Pro použití této služby se musí do webových stránek vložit speciální měřicí kód, který v sobě obsahuje měřicí číslo, to aby služba věděla k jakému Google účtu je tento kód přiřazen. Google Analytics je nejpoužívanější nástroj v této oblasti služeb.

4 Definice částí webových stránek

V této kapitole, dle zadání mé maturitní práce, definuji, které části jsou originálním dílem. V následujících řádcích se budu zabývat spíše tím, co není originálním dílem, jelikož drtivá většina originálním dílem je.

Tyto webové stránky nejsou svým rozsahem výrazně veliké a ani se nebudou v blízké době měnit nebo nepoužívají napojení na databázi. Proto jsem se rozhodl z hlediska náročnosti tvořit tyto webové stránky jako statické. Není v nich tedy použit žádný redakční systém. Stránky jsou psané pomocí technologií HTML + CSS + jQuery v programu PSPad. Některé části kódu však nejsou originální ale pouze vložené, například z jiných služeb.

4.1 *HTML kód*

Většina HTML kódu je originálním dílem, ale některé jeho části jsou pouze nakopírovaným kódem třetích stran pro podporu služeb třetích stran.

Konkrétně jde o styly fontů společnosti Google které jsou vloženy ve značkách `<link>` v hlavičce souborů HTML. Ty slouží k použití nadstandardních fontů na webových stránkách. Bez načtení těchto stylů by nebylo možné mít toto písmo na webu.

Dále jsou v HTML souborech vloženy odkazy pro načtení jQuery knihovny ze serverů společnosti Google. Tato knihovna mohla být také vložena přímo v souboru do adresáře z webovými stránkami a načtena odtud. Její načtení jsem však zvolil tímto způsobem, protože

na serverech společnosti Google je knihovna stále aktuální a je to pro mě pohodlnější. Bez načtení této knihovny by na webových stránkách nemohly fungovat jQuery skripty.

Jako další neoriginální část HTML kódu je kód ke službě Google Analytics, jedná se o script kterým se při každém otevření webové stránky v jakémkoli prohlížeči ze sítě internet, uloží data do služby Google Analytics. To mi umožňuje, podle mého sledovacího čísla a nastavením tohoto scriptu, sledovat detaily o zařízeních, které tento web navštívily. Naprogramovat podobnou službu by bylo příliš složité, a proto jsem použil službu Google Analytics a vložil její měřicí kód do těchto stránek. K tomu jsem ještě vložil kód služby Google Tag Manager která přímo souvisí se službou Google Analytics a umožňuje mi ještě lepší nastavení sledování zařízení.

Další částí v HTML kódu která není originální je script, který načte rozhraní ke službě společnosti Facebook. Díky její existenci v kódu, mohu používat funkce sociální sítě Facebook a lidé mohou dát této stránce tzv. "like" ve službě Facebook přímo z těchto stránek v zápatí.

V zápatí jsou dále vloženy části kódu od společností Facebook a Google pro sociální sítě které na stránkách vygenerují tlačítka pro tzv. "like" na sociální síti Facebook a "follow" na sociální síti Google+.

V HTML dokumentu "kontakt.html" je také vložen kód ke službě Google Maps, od společnosti Google. Jeho přítomnost zde, umožňuje načtení miniatury mapy a po kliknutí na miniaturu následné otevření předem nadefinované adresy na stránkách Google Maps.

4.2 CSS kód

CSS kód je kompletně originální.

4.3 jQuery kód

Samotná jQuery knihovna je načtená pomocí skriptů vložených v HTML dokumentů ze serveru společnosti Google. Jednotlivé skripty jsou již však originálním dílem.

4.4 Grafika

O grafiku jednotlivých částí webu se starají především kódy CSS kde jsou nadefinovány například barvy, velikosti nebo pozice jednotlivých prvků. Obrázky načtené na webu přes HTML značky jsou z většiny originálním dílem.

Ikony jsou vytvořené v programu Adobe Flash nebo v programu Adobe Photoshop. Pouze některá pozadí nebo fotografie jsou obrázky stažené z volných zdrojů na internetu.

4.5 Obsah

Veškerý text na webu a další obsah je originálním dílem.

5 Závěr

Při programování těchto webových stránek jsem ze začátku nenarazil na žádné výrazné obtíže. Vzhledem k tomu že jsem dělal webové stránky, které mají především prodat službu potenciálním zákazníkům, bylo mnohem těžší vymyslet web jako takový než jeho samotné programování. Musel jsem tedy nejprve vymyslet, jak bude web vypadat, co bude obsahovat a také jak nejlépe oslovit zákazníka. Samotné programování a tvorba webu zabrala méně času, než plánování a přemýšlení nad obchodní strategií webu. Při programování jsem se však potkal s některými problémy. Když tvoříte dynamické webové stránky, většinou vám redakční systémy nabízejí tzv. responzivní šablony. Jsou to šablony, které se přizpůsobují svým vzhledem všem typům rozlišení a zařízení. U statických stránek jsem zpočátku na toto zapomněl a poté se web na jednotlivých zařízeních zobrazoval špatně. Musel jsem tedy upravit celkovou koncepci a technologické rozvržení webu tak, aby se jednotlivé prvky na stránce přizpůsobovali velikostně samy, podle rozlišení. To se mi povedlo, ale pouze z části. Webové stránky se teď zobrazují správně pouze na rozlišeních od šířky 1024px a výše. To znamená, že na většině počítačů se webové stránky zobrazí správně, na mobilních telefonech už však ne. Proto budu v budoucnu dělat ještě samostatnou verzi pro mobilní telefony která bude mít zjednodušenou grafickou podobu, uzpůsobenou na výšku pro malé displeje. Jakmile tedy uživatel vstoupí na webové stránky, jQuery skript inicializuje jaké zařízení uživatel používá a pokud bude zařízení mobilní telefon (rozlišení nižší než 1024px), webová stránka se přepne na mobilní verzi webu.

6 Seznam příloh

Příloha č. 1 CD disk

7 Použitá literatura a odkazy

- [Lit. 1] https://cs.wikipedia.org/wiki/Webov%C3%A1_str%C3%A1nka
[Lit. 2] https://cs.wikipedia.org/wiki/HyperText_Markup_Language
[Lit. 3] https://cs.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_styl