

## Obsah

Grafický dizajn .....	2
Obrázky.....	2
Texty .....	3
Menu .....	3
Nadpisy.....	3
Responzivita .....	4
Meta data .....	4
Page Title .....	4
Meta Description.....	5
Java Script.....	5
CSS .....	6
Minifying.....	6
Favorite Icon.....	7
Sitemap.....	8
URL adresy .....	10
Redirecting .....	12
Google Analytics.....	14
Google Search Console.....	14
Ladenie a hľadanie chýb.....	14
Ako ďalej.....	15

## Grafický dizajn

Prvé, čo si každý na webovej stránke všimne je jej vzhľad. Vytvorenie naozaj peknej webovej stránky nie je až také ťažké ako by sa mohlo zdať. Čím viac stránok vytvoríš tým lepšie ti to pôjde.

Základom je správna kombinácia vhodných farieb, výber typu písma, pekné obrázky a prehľadné a štruktúrované usporiadanie prvkov. Inšpiráciu vzhľadu pre nový web môžeš hľadať napríklad na <https://www.templatemonster.com>.

Obrázky môžeš nájsť pekné aj v bezplatných fotobankách. Napríklad na <https://pixabay.com> alebo na <https://www.pexels.com>. Tie najkrajšie sú v platených fotobankách ako napríklad na <https://www.shutterstock.com>. Výber je naozaj široký a určite sa oplatí používať obrázky aj z platených fotobánk, pretože to ušetrí veľa času a nestojí to veľa, ak to porovnáme s celkovou cenou webu.

Úplne ideálny obrázok nájdeš málokedy, takže hľadaj niečo blízke potrebám webu a obrázok dolad' v nejakom grafickom editore.

Ak chceš ešte viac upútať pozornosť návštevníka webu, tak skús použiť aj video, ktoré vložíš na pozadie stránky ako je to napríklad na <http://www.dufeksoft.com>.

Vylepšiť vzhľad webu sa dá výberom správneho fontu. Pekné a bezplatné fonty nájdeš na <https://fonts.google.com>. Keď sa rozhodneš niektorý z nich použiť, tak si over, či podporujú aj slovenské znaky s diakritikou.

Pekným doplnkom sú aj rôzne malé grafické ikony, ktorými môžeš dopĺňať nadpisy alebo ilustrovať niektoré vlastnosti, či odkazy napríklad na sociálne siete. Dobrý pomocníkom je Font Awesome, ktorý nájdeš na adrese <http://fontawesome.io>. Určite sa ho nauč používať. Zakomponuj ho do svojho webu, aby som si to mohol pozrieť. Ale urob to tak, aby súbory fontov boli súčasťou tvojho webu a neodkazoval si na nejaké cudzie weby, pretože ak by cudzí web nefungoval, tak ani Font Awesome by na tvojom webe nebežal.

## Obrázky

Obrázky sme už spomínali. Okrem ich vzhľadu je však dôležitý aj ďalší faktor, ktorým sú rozmery obrázka a jeho veľkosť. Aby sa web načítal, čo najrýchlejšie, tak je potrebné, aby veľkosť obrázka bola, čo najmenšia.

Vždy pamätaj na to koľko a akých obrázkov na webe používaš. Obrázky, ktoré majú prekryvať celú obrazovku sa snaž vtesnať do veľkosti 100 KB.

Viac o dôvodoch a spôsobe úprav obrázkov si prečítaj na našom blogu v článkoch na adrese

<http://blog.dufeksoft.sk/post/2017/05/17/ako-spravne-pouzivat-obrazky-na-webovych-strankach>

---

a

<http://blog.dufeksoft.sk/post/2017/05/25/ako-upravit-obrazky-pre-webove-stranky>.

## Texty

Texty, ktoré sú na webe použité majú návštevníka webu zaujať a presvedčiť ho, aby nakúpil alebo využil služby, ktoré daný web ponúka.

Práca s textom je samostatná kapitola, ktorá mnohokrát rozhoduje o úspechu webu viac ako samotný vzhľad webu.

Super informácie a rýchlo kurz za malý peniaz nájdeš na <http://365copytriku.cz>. Určite si to pozri a za pár dní sa z teba stane takmer profi copywriter.

## Menu

Úlohou menu je ponúknuť návštevníkovi webu na výber jednotlivé možnosti, ktoré obsah webu ponúka.

Pri tvorbe menu sa najčastejšie robia dve základné chyby.

Prvá, že sa do menu sústreďí komplet celý obsah webu a menu sa tak stáva neprehľadným a návštevník nevie rýchlo nájsť, čo potrebuje. Menu by malo byť stručné a nemalo byť návštevníka webu zahltiť informáciami.

Druhá častá chyba je, že menu obsahuje všeobecné nadpisy ako Úvod, O nás, Cenník, či Kontakt. Pri tvorbe webu treba myslieť na to, že ho nenavštevujú iba ľudia, ale aj rôzne automaty, ktoré web indexujú a potom ponúkajú jeho výsledky vo vyhľadávaní. Keďže menu je na každej podstránke webu totožné, tak je pre indexáciu obsahu webu veľmi dôležité. Nikto na webe nevyhľadáva Úvod, O nás, či Kontakt. Ľudia hľadajú konkrétne produkty a služby a tak ich názov zakomponujte aj do menu, aby ich robot od Google indexoval a pomohli ste danému webu sa dostať vyššie vo výsledkoch vyhľadávania.

## Nadpisy

Podobne ako menu aj nadpisy tvoria dôležitú súčasť textov a samotného obsahu webu. To, čo platí pre texty a menu, platí aj pre nadpisy.

Majú byť stručné, výstižné a pútavé. Vyhýbajte sa v nadpisoch rôznym všeobecným frázam ako Úvod, O nás, či Naša firma. Texty nadpisov sú opäť indexované a ovplyvnia postavenie webu vo výsledkoch vyhľadávania, preto by mali obsahovať výrazy, pre ktoré chceš daný web optimalizovať pre vyhľadávanie cez Google.

A veľmi dôležité je dodržať hierarchiu nadpisov. Na každej stránke by mal byť jeden nadpis prvej úrovne, teda H1. A potom postupne môžeš použiť nadpisy ďalších úrovní H2, H3, H4 atď. Dôležité je dodržať ich hierarchiu a nepoužiť napríklad H3 skôr ako H2.

## Responzivita

Dnes ľudia navštevujú webové stránky z počítačov, tabletov, mobilov a ďalších zariadení. Ak nechceš návštevníka webu odstrašiť a odradiť od ďalších návštev konkrétneho webu, musíš vzhľad webu a jeho funkčnosť prispôbiť pre jednotlivé zariadenia, aby aj na malom displeji na mobile sa dalo s webom dobre pracovať.

Existujú rôzne prístupy ako sa to dá realizovať, ale najpoužívanejšia cesta je Bootstrap. Na adrese <http://getbootstrap.com/getting-started/> si pozri, čo to Bootstrap je a aké má možnosti. V zásade je Bootstrap iba niekoľko CSS a JS súborov, ktoré definujú určité štýly a správanie prvkov na webovej stránke. Na adrese <https://www.w3schools.com/bootstrap/> nájdeš tutoriály s konkrétnymi príkladmi, kde sa krok za krokom naučíš Bootstrap používať.

Aj tvoja stránka musí byť responzívna, takže je spracuj použitím Bootstrap knižnice. Neskôr si to skontrolujem.

## Meta data

Výsledný HTML kód webu musí obsahovať aj tzv. META TAGY, ktoré určujú ako sa má web zobrazovať.

Nižšie nájdeš ukážku z hlavičky konkrétneho webu. Minimálne nižšie uvedené META TAGY by mal každý web obsahovať. Vyhľadaj a naštuduj si ich význam a spôsob použitia.

```
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="copyright" content="(c) SPEMB, spol.s r.o." />
<meta name="Keywords" content="sanačné omietky, plesneň v dome, plesneň v byte" />
<meta name="description" content="Sanačné omietky. Zdravý domov bez plesne." />
<meta name="language" content="sk" />
<meta name="content-language" content="sk" />
<meta name="author" content="SPEMB, spol.s r.o." />
<meta name="robots" content="index, follow" />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

## Page Title

Každá podstránka na webe musí obsahovať zápis `<title>Nadpis stránky</title>`. Ten určuje nadpis stránky, ktorý býva zobrazený v hlavičke prehliadača.

Okrem toho je však Title súčasťou indexácie webovej stránky pre vyhľadávanie cez Google a preto je veľmi dôležité dodržať dve základné pravidlá. Každá stránka musí mať iný Title a Title nesmie byť príliš dlhý. Nie je vhodné, aby bol dlhší ako 70 znakov.

Prečo 70 znakov, si prečítaj na <http://www.theseempost.com/new-title-description-lengths-for-google-seo>.

## Meta Description

Podobne ako Title tak aj Meta tag Description `<meta name="description" content="Sanačné omietky. Zdravý domov bez plesne." />`. Je súčasťou výsledkov vyhľadávania. Aj pre Description platí, že pre každú stránku má byť tento popis jedinečný. Description býva dlhšie ako Title a o jeho ideálnej dĺžke si si mohol prečítať na adrese <http://www.theseempost.com/new-title-description-lengths-for-google-seo>.

## Java Script

Java Script je dôležitou súčasťou webových stránok. Vďaka nemu môžu byť webové stránky rýchlejšie, flexibilnejšie a môžu poskytnúť návštevníkovi webu lepší zážitok.

Ak chceš vytvárať perfektné webové stránky, tak bez použitia Java Scriptu sa nezaobídeš. Ak si to ešte neurobil, tak sa pusť do jeho štúdia na adrese <https://www.w3schools.com/js>.

Užívatelia pozerajú webové stránky na počítačoch, mobiloch, tabletoch a využívajú rôzne prehliadače ako Chrome, Safari, Internet Explorer, Firefox, Operu a iné. Zapracovanie Java Scriptu a spracovanie jeho príkazov sa môže v jednotlivých prehliadačoch líšiť a môže spôsobiť nefunkčnosť webových stránok, preto by si mal vždy web vyskúšať vo viacerých prehliadačoch.

Našťastie existujú knižnice, ktoré už vytvorili iní vývojári a poskytujú ich bezplatne. V nich riešia aj rozdiely medzi prehliadačmi a poskytujú rôzne výborné funkcie, ktoré ti pri tvorbe webu pomôžu. Nutnosťou je poznať a používať knižnicu JQuery. Zoznámiť sa s ňou môžeš na adrese <https://jquery.com> a naštuduj si jej používanie na adrese <https://www.w3schools.com/jquery>.

Vždy, keď je to možné snaž sa všetok JavaScript kód dať mimo HTML kódu webovej stránky. Ak prehliadač načíta JavaScript súbor, tak si ho uloží a ďalšie načítanie podstránky rovnakého webu, ktorý používa rovnaký JavaScript súbor, bude rýchlejšie, pretože daný JavaScript súbor už prehliadač má a nebude ho znovu sťahovať zo servera.

Má to však aj jednu nevýhodu. Ak sa JavaScript súbor medzičasom zmenil a zákazník príde na web znovu, nemusí web fungovať správne, keďže prehliadač nestiahne nový JavaScript súbor, ale použije starý, ktorý má uložený. To sa dá vyriešiť doplnkovým parametrom v odkaze na JavaScript súbor v HTML kóde stránky a tým vynútiť jeho nové načítanie.

Chcem, aby si JQuery knižnicu použil aj na svojom webe. Ale podobne ako pre Font Awesome chcem, aby súbory JQuery boli súčasťou tvojho webu a neodkazoval si na iné weby.

Skús napríklad zapracovať možnosť zobrazenia videa z YouTube podobne ako je to cez tlačidlo **Pozrite video** na webe <http://www.dufeksoft.com/sk>.

## CSS

CSS je jazyk, ktorý sa používa pre určenie vzhľadu webu. Farby, veľkosť písma, tvary prvkov, rôzne transformácie a mnoho ďalších funkcií poskytuje CSS.

Nepochybujem, že si sa s ním už zoznámil, ale pre istotu prikladám odkaz na tutoriál na adrese <https://www.w3schools.com/css>.

Z rovnakých dôvodov ako JavaScript by aj všetok CSS kód mal byť ukladaný do samostatných súborov. Rovnako pre CSS súbory platí aj spôsob sťahovania a ich uchovávanie v prehliadačoch.

## Minifying

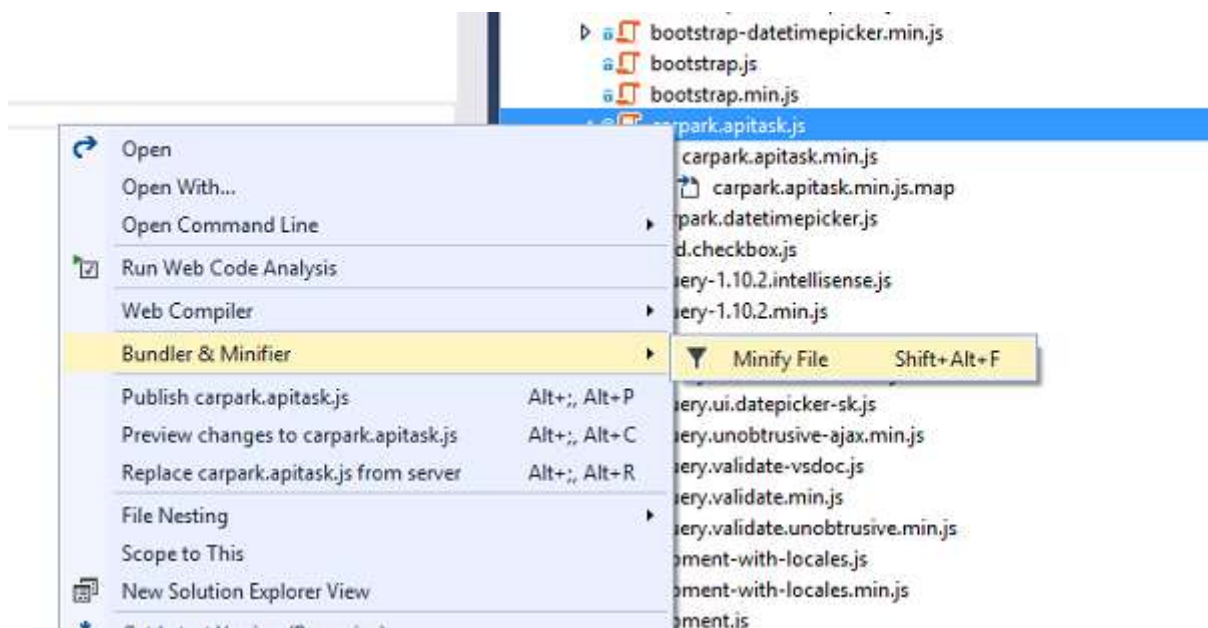
Okrem vzhľadu je dôležitá aj rýchlosť načítania webových stránok. Čím menej dát sa posieľa zo servera ku klientovi, tým rýchlejšie sa web zobrazí.

O obrázkoch sme už písali, teraz sú na rade ďalšie typy súborov.

JavaScript a CSS súbory obsahujú množstvo medzier a prázdnych riadkov, aby sa dali lepšie čítať a aby sa v nich programátor lepšie orientoval. Ale pre samotné zobrazenie webu sú tieto medzery a prázdne riadky úplne zbytočné a zbytočne nafukujú veľkosť súboru.

Preto je dobré sa týchto prázdny vecí zbaviť a tento proces sa nazýva Minifying. Robiť to ručne, by samozrejme zabralo veľa času a tak ja na to priamo vo Visual Studiu funkcia.

Funkciu vyvoláš cez kontextové menu kliknutím pravým tlačidlo myši na príslušný JS alebo CSS súbor ako to znázorňuje nasledovný obrázok.



Po tejto akcii vznikne nový súbor s prídavkom min. Ten sa automaticky vždy vygeneruje, keď uložíš vo Visual Studiu pôvodný súbor. Pri editácii, tak stále pracuješ s pôvodným súborom, ale na nasadenie a testovanie webu už používaš zmenšený súbor.

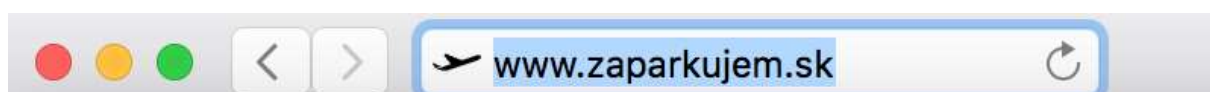
Zápis použitia súboru v MVC projekte potom vyzerá nasledovne.

```
<script src="@Url.Content("~/Scripts/carpark.apitask.min.js")" type="text/javascript"></script>
```

## Favorite Icon

Ilustračná ikonka webu je drobnosť. Ikona sa zobrazuje v hlavičke alebo adresnom riadku prehliadača a väčšina ľudí si ju možno ani nevšimne.

Napríklad pre web [www.zaparkujem.sk](http://www.zaparkujem.sk) v prehliadači Safari to vyzerá nasledovne. Je tam ikonka lietadla, keďže daná stránka poskytuje služby rezervácie parkovania pri letisku v Bratislave.



Keď si niekto pripne odkaz na web na plochu počítača, tak už si ikonu človek určite všimne a uvidí, že daný web developer si dal naozaj na vzhľade webu záležať.

Najlepšie je vytvoriť PNG súbor v rozlíšení 64 x 64 px. PNG preto, lebo môžeš použiť aj transparentnosť, takže ikona bude mať zachované pekné obrysy.

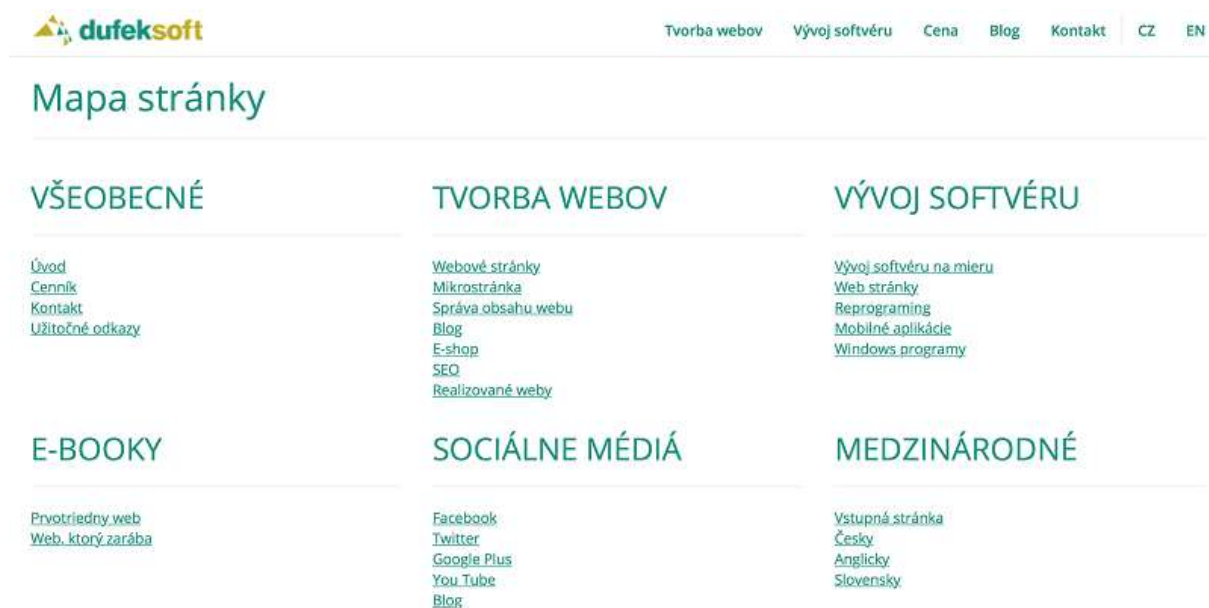
Obrázok skonvertuješ do ICO formátu napríklad na adrese <https://iconverticons.com/online>. Vznikne ICO súbor so zoznamom obrázkov v rôznom rozlíšení pre rôzne typy zobrazenia a rozlíšenia

obrazoviek. Súbor zaraď niekde do projektu vo Visual Studiu a zapíš odkaz naň do časti HEAD v HTML kóde webovej stránky.

```
<link rel="shortcut icon" href="@Url.Content("~/Styles/images/parkovanie-letisko.ico")" />
```

## Sitemap

Mapa stránok má dva účely. Ak návštevník webu niečo hľadá, môže to nájsť oveľa rýchlejšie, ak mu dáš k dispozícii nahliadnúť do štruktúry webu. Napríklad nasledovne.



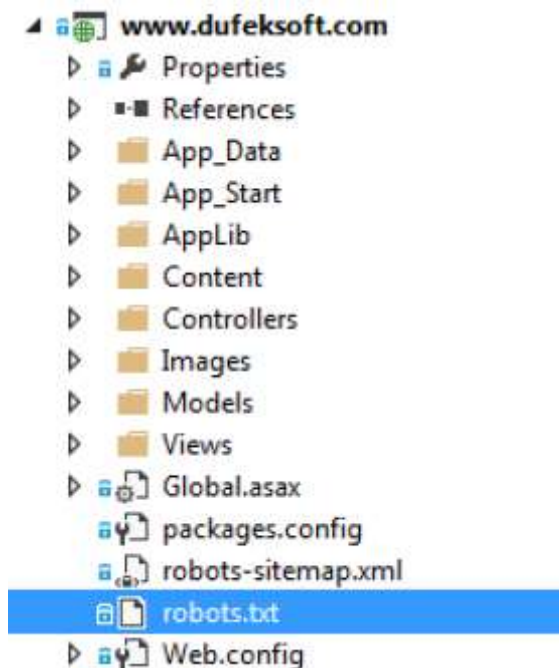
The screenshot shows the Dufeksoft website header with the logo and navigation menu (Tvorba webov, Vývoj softvéru, Cena, Blog, Kontakt, CZ, EN). Below the header is a section titled "Mapa stránky" (Sitemap) with a grid of categories and their sub-items:

- VŠEOBECNÉ**
  - Úvod
  - Cenník
  - Kontakt
  - Užitočné odkazy
- TVORBA WEBOV**
  - Webové stránky
  - Mikrostránka
  - Správa obsahu webu
  - Blog
  - E-shop
  - SEO
  - Realizované weby
- VÝVOJ SOFTVÉRU**
  - Vývoj softvéru na mieru
  - Web stránky
  - Reprogramming
  - Mobilné aplikácie
  - Windows programy
- E-BOOKY**
  - Prvotriedny web
  - Web, ktorý zarába
- SOCIÁLNE MÉDIÁ**
  - Facebook
  - Twitter
  - Google Plus
  - You Tube
  - Blog
- MEDZINÁRODNÉ**
  - Vstupná stránka
  - Česky
  - Anglicky
  - Slovensky

Oveľa dôležitejšie je však vytvoriť mapu stránok pre Google robota a iné automatizované nástroje, ktoré sa snažia prechádzať celý web a indexujú obsah internetu pre vyhľadávanie a iné služby.

Základom je vytvorenie súboru Robots.txt, ktorý umiestni priamo do koreňového adresára webu.





Obsah súboru môže byť nasledovný.

```
Sitemap: http://www.dufeksoft.com/robots-sitemap.xml
User-agent: *
Disallow:
```

Najdôležitejšou časťou v tomto súbore je odkaz na súbor robots-sitemap.xml. Na názve nezáleží. Môže to byť statický súbor alebo aj akákoľvek dynamická URL adresa, prípadne aj viac odkazov, ale musí to vracať dáta v jasne definovanom formáte, ktorý je pre Sitemap súbory určený.

Môže to byť niečo nasledovné.

```
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url><loc>http://www.dufeksoft.com/</loc></url>
  <url><loc>http://www.dufeksoft.com/en</loc></url>
  <url><loc>http://www.dufeksoft.com/en/software-development</loc></url>
  <url><loc>http://www.dufeksoft.com/en/reprogramming</loc></url>
  <url><loc>http://www.dufeksoft.com/en/website-design</loc></url>
  <url><loc>http://www.dufeksoft.com/en/website-design/websites-delivered</loc></url>
  <url><loc>http://www.dufeksoft.com/en/windows-apps</loc></url>
  <url><loc>http://www.dufeksoft.com/en/mobile-apps</loc></url>
  <url><loc>http://www.dufeksoft.com/en/price</loc></url>
  <url><loc>http://www.dufeksoft.com/en/contact</loc></url>
  <url><loc>http://www.dufeksoft.com/en/sitemap</loc></url>
  <url><loc>http://www.dufeksoft.com/en/useful-links</loc></url>
  <url><loc>http://www.dufeksoft.com/sk</loc></url>
  <url><loc>http://www.dufeksoft.com/sk/tvorba-webov</loc></url>
```

Viac o Sitemap a Robots súboroch si vyhľadaj a prečítaj na internete. Sú to dôležité prvky SEO optimalizácie.

## URL adresy

Každá podstránka webu má svoju jednoznačnú URL adresu. V prípade tvorby webu formou MVC projektu vo Visual Studiu je logika tvorby URL adres daná názvom Controllera a názvom príslušnej Action metódy.

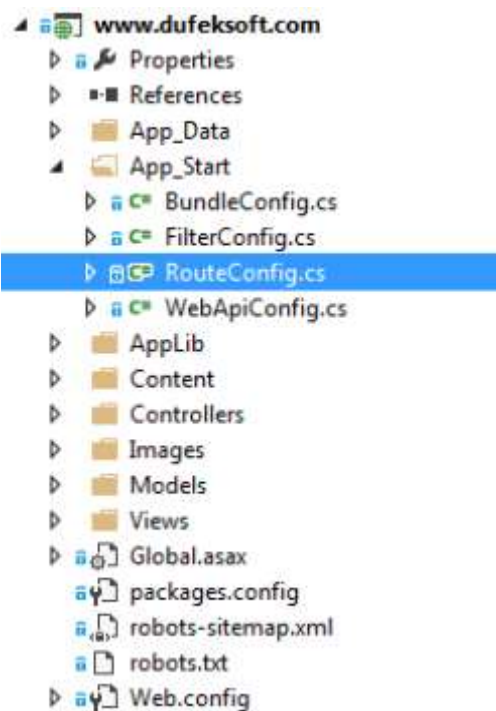
Napríklad pre web [www.dufeksoft.com](http://www.dufeksoft.com) ak použijeme Controller triedu s názvom `HomeSkController` a nasledovný kódom.

```
public class HomeSkController : _DufeksoftController
{
    public HomeController()
        : base("HomeSk")
    {
    }

    public ActionResult Portfolio()
    {
        SetLanguageAction("Portfolio");
        SetMenuCssClass("Portfolio", "Portfolio");
        return View();
    }
}
```

Tak zápis `@Url.Action("Portfolio", "Home")` vygeneruje URL adresu <http://www.dufeksoft.com/HomeSk/Portfolio>. Webová stránka bude plne funkčná, avšak z pohľadu SEO optimalizácie je to chybné riešenie. Uvedená adresa obsahuje stránku s informáciami o softvérových službách. Ak Google robot indexuje URL adresu <http://www.dufeksoft.com/HomeSk/Portfolio>, tak podľa URL adresy to nespojí so softvérovými službami. To zistí, až keď sa ponorí do obsahu danej webovej stránky a stránka tak stráca na hodnotení. Ak by však URL adresa mala tvar <http://www.dufeksoft.com/sk/vyvoj-softveru>, tak je to oveľa lepšie, pretože už názov URL adresy hovorí o tom, že stránka obsahuje informácie o vývoji softvéru.

Našťastie MVC framework na toto pamätá, takže docieľiť optimálny tvar URL adres nie je vôbec ťažké. Nájdi vo Visual Studiu v príslušnom projekte súbor `RouteConfig.cs` a v ňom môžeš určiť, aké budú tvary URL adres.



Kód môže vyzerat' nasledovne.

```
public class RouteConfig
{
    public static void RegisterRoutes(RouteCollection routes)
    {
        routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

        //
        // SEO optimized URLs
        //

        // HomeSk controller
        routes.MapRoute("sk", "sk", new { controller = "HomeSk", action = "Index" });
        routes.MapRoute("sk/tvorba-webov", "sk/tvorba-webov", new { controller = "HomeSk", action =
"Web_" });
        routes.MapRoute("sk/tvorba-webov/seo", "sk/tvorba-webov/seo", new { controller = "HomeSk",
action = "Web_Seo" });
        routes.MapRoute("sk/tvorba-webov/blog", "sk/tvorba-webov/blog", new { controller =
"HomeSk", action = "Web_Blog" });
        routes.MapRoute("sk/tvorba-webov/eshop", "sk/tvorba-webov/eshop", new { controller =
"HomeSk", action = "Web_Eshop" });
        routes.MapRoute("sk/tvorba-webov/mikrostranka", "sk/tvorba-webov/mikrostranka", new {
controller = "HomeSk", action = "Web_Microsite" });
        routes.MapRoute("sk/tvorba-webov/cms", "sk/tvorba-webov/cms", new { controller = "HomeSk",
action = "Web_Cms" });
        routes.MapRoute("sk/tvorba-webov/realizovane-weby", "sk/tvorba-webov/realizovane-weby", new
{ controller = "HomeSk", action = "Web_Projects" });
    }
}
```

Viac o SEO optimized URL si nájdí na internete.

## Redirecting

Ďalším faktorom SEO optimalizácie je adresovanie už neexistujúcich webových adries a adresy s identickým obsahom.

Mnoho webových projektov má za cieľ vytvoriť novú verziu webových stránok, takže nie vždy vytváraš úplne nový web. Obsah pôvodného webu už mal Google zindexovaný a môže mať veľmi dobré postavenie vo výsledkoch vyhľadávania.

Ak nový web vytvoríš s úplne inými URL adresami podstránok ako mal pôvodný web, môžeš tým vlastníkovi webu dosť uškodiť a pripraviť ho o zákazníkov. Tým, že staré URL adresy už nebudú funkčné, Google to bude považovať za chybu a bude znižovať hodnotenie webu a nebude vedieť ani zobrazíť vo vyhľadávaní výsledky pôvodného webu a zatiaľ ani výsledky nového webu. Chvíľu mu to potrvá, kým to celé zaktualizuje. A to môže trvať aj niekoľko týždňov. A dostať sa na pôvodné pozície vo vyhľadávaní bude trvať mesiace a možno aj roky.

Preto si daj záležať, aby nové URL adresy nahradili tie pôvodné. Ak to nie je možné, tak zabezpeč, aby stará URL vykonala presmerovanie na novú URL adresu technikou Permanent Redirect.

Môžeš na to vytvoriť napríklad samostatný Controller a v spojení s mapovaním URL adries podľa predchádzajúcej kapitoly v súbore RouteConfig.cs dosiahneš požadovaný výsledok.

Kód pre RedirectController môže vyzeráť nasledovne.

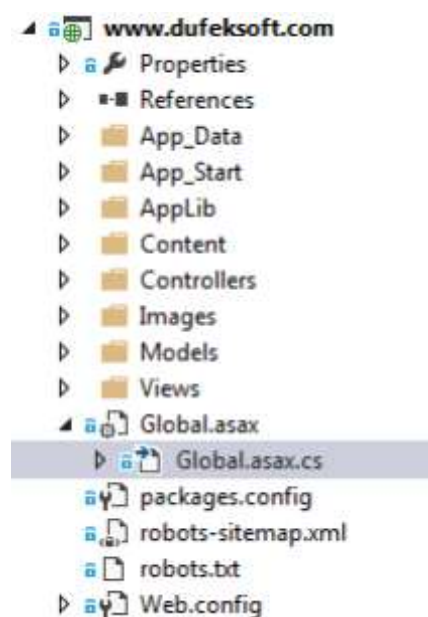
```
public class RedirectController : Controller
{
    //
    // GET: /Redirect/

    public ActionResult Cena()
    {
        return RedirectPermanent(Url.Action("Price", "HomeSk"));
    }
    public ActionResult Kontakt()
    {
        return RedirectPermanent(Url.Action("Contact", "HomeSk"));
    }
    public ActionResult OnlineUctovnictvo()
    {
        return RedirectPermanent(Url.Action("Reprogramming", "HomeSk"));
    }
    public ActionResult Bookkeeping()
    {
        return RedirectPermanent(Url.Action("Reprogramming", "HomeSk"));
    }
    public ActionResult ProduktyASluzby()
    {
        return RedirectPermanent(Url.Action("Portfolio", "HomeSk"));
    }
}
```

Ďalším aspektom je, že Google penalizuje stránky s rovnakým obsahom. Berie to ako podvodnú techniku, kde sa majiteľ webu snaží budovať obrovský obsah, ale v zásade je obsah malý, len je to stále to isté iba na inej adrese.

Takže v prvom rade sa vyhni tomu, že budeš úplne tie isté texty používať na rôznych stránkach. Ak to nerobíš, tak aj tak ešte nemáš úlohu splnenú. Väčšina webov bude ten istý obsah zobrazovať aj na adrese s www aj na adrese bez www. Napríklad <http://www.dufeksoft.com> zobrazí to isté ako <http://dufeksoft.com>. Ak to však vyskúšaš zistíš, že pri prístupe na <http://dufeksoft.com> sa zobrazí adresa <http://www.dufeksoft.com>. A tak to má byť. Musíš zaistiť, že sa vykoná presmerovanie vždy na jednu webovú adresu a nebude sa zobrazovať ten istý web na rôznych adresách. Čiže nastaviš Permanent Redirect requestov bez www na requesty s www.

Dosiahnuť sa to dá viacerými spôsobmi. Môžeš využiť napríklad súbor Global.asax.cs a v ňom metódu Application\_BeginRequest.



```
protected void Application_BeginRequest(object sender, EventArgs e)
{
    if (HttpContext.Current.Request.Url.ToString().ToLower().Contains("http://dufeksoft.com"))
    {
        HttpContext.Current.Response.Status = "301 Moved Permanently";
        HttpContext.Current.Response.AddHeader("Location",
            Request.Url.ToString().ToLower().Replace("http://dufeksoft.com", "http://www.dufeksoft.com"));
    }
}
```

## Google Analytics

Google Analytics je nástroj, ktorým môžeš merať návštevnosť webových stránok. Jeho základná verzia je bezplatná. Ak niekto seriózne podniká a považuje svoj web za relevantný marketingový a predajný nástroj, nemala by mu chýbať integrácia na Google Analytics.

Prepojenie na Google Analytics sa dá nastaviť veľmi jednoducho. Je to iba začlenenie konkrétneho JavaScript kódu do daného webu.

Viac o možnostiach Google Analytics si nájdí na internete. Vytvoríte si tam svoje vlastné konto a začnete skúšať.

<https://www.google.com/analytics>

## Google Search Console

Google Search Console je ďalším výborným bezplatným nástrojom. Aj na tento by si mali web napojiť.

Tento nástroj umožňuje identifikovať chyby, na ktoré Google narazil pri prehľadávaní daného webu, umožňuje manuálne zasielať Sitemap súbory a tým urýchliť indexáciu webu a mnoho ďalších vecí.

Ak by sa napríklad nebudaj stalo, že web, ktorý spravuješ bol napadnutý škodlivým kódom a Google to zistí, tak zablokuje prístup na daný web aj Google Chrome aj Firefox. Vtedy sa ti bude hodiť Google Search Console, pretože po oprave webu, môžeš práve cez Google Search Console poslať žiadosť na sprístupnenie webu.

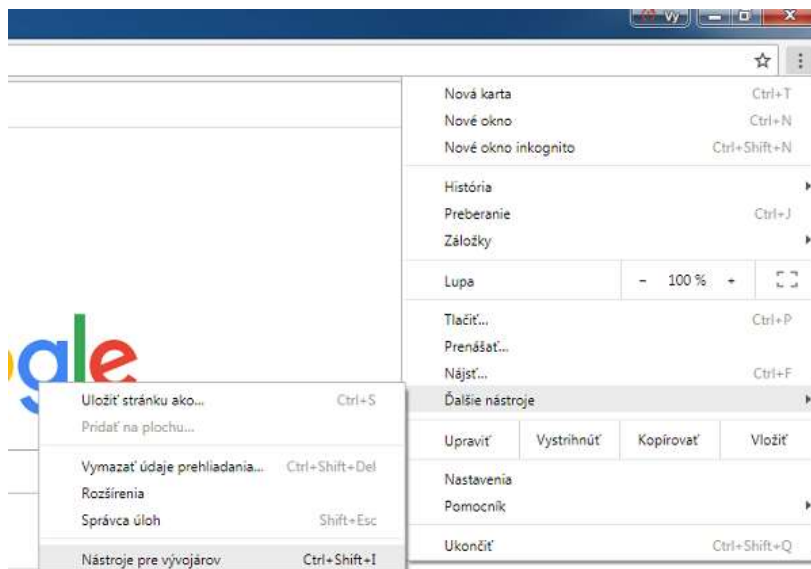
<https://www.google.com/webmasters>

## Ladenie a hľadanie chýb

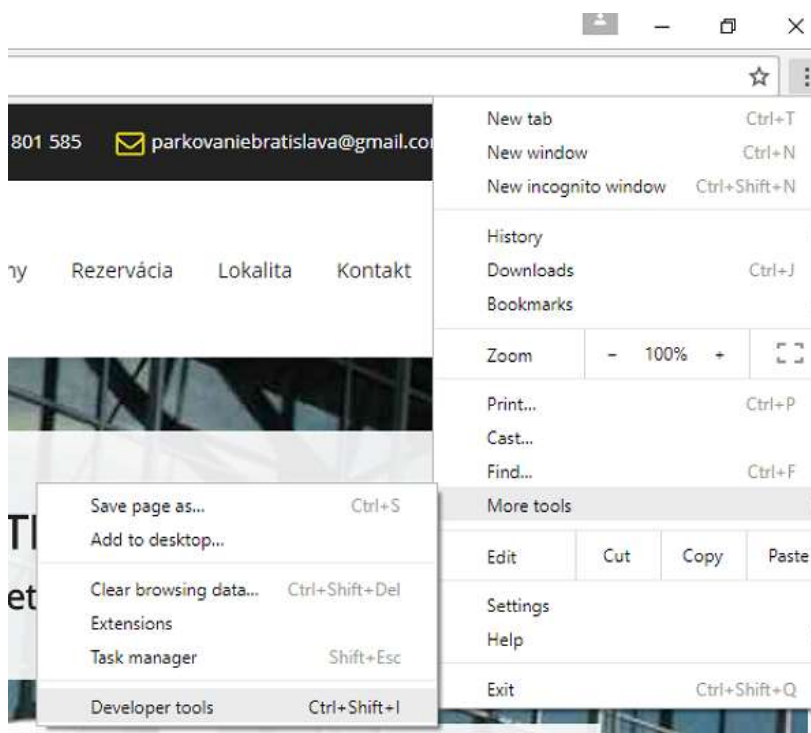
Človek nie je neomylný a tak aj pri vývoji webu urobíš mnoho chýb, než sa ti ho podarí reálne nasadiť. Visual Studio má preto k dispozícii tzv. Debug mód, kedy môžeš kód kedykoľvek zastaviť a pozrieť sa na hodnoty jednotlivých premenných.

Ak si Debug mód ešte neobjavil, tak si o ňom vygoogli a prečítaj čo najviac.

Nie všetky časti kódu však dokážeš odhaliť a nájsť cez Debugger vo Visual Studiu. Najmä chyby v CSS a JavaScripte sa hľadajú oveľa ťažšie. K tomu ti ale zasa pomôžu nástroje pre vývojárov, ktoré obsahujú jednotlivé prehliadače. Napríklad Google Chrome má Nástroje pre vývojárov, ku ktorým sa dostaneš nasledovne.



alebo



## Ako ďalej

Ak si zvládol všetky predchádzajúce kapitoly a vytvoril web, ktorý spĺňa jednotlivé popísané kritériá, tak môžeš začať vytvárať základné webové stránky v prvotriednej kvalite.

Tvorba takej stránky by ti nemala zabrať viac ako dva dni a zákazník ti za ňu nezaplatí viac ako 200 Euro.

Aby si sa dopracoval k zložitejším webom, tak si teraz tak niekde v polovici cesty.

Ostáva nám prejsť ešte napojenie na databázu, aby web mohol pracovať s dynamickými údajmi, aby sa ľudia mohli na web registrovať a vkladať tam rôzne dáta.

A posledným krokom bude naučiť sa pracovať s redakčným systémom. Naučím ťa vytvoriť web, ktorý poskytne jeho majiteľovi možnosť meniť obsah webu bez toho, aby si ty ako programátor musel čokoľvek robiť.

Keď to zvládneš, tak sa môžeš začať živiť tvorbou webových stránok.