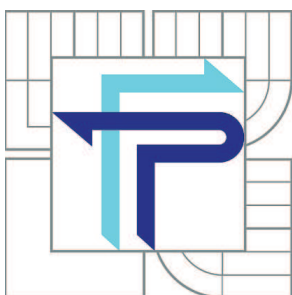


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH INTERNETOVÝCH STRÁNEK

WEBSITE DESIGN

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ZDENĚK VIČAR

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. MILOŠ KOCH, CSc.

BRNO 2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Vičar Zdeněk

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Návrh internetových stránek

v anglickém jazyce:

Website Design

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení, přínos návrhů řešení

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

CEDERHOLM, Dan. Flexibilní webdesign. 1.vyd. Brno: Computer Press, a.s., 2006.232 s. ISBN: 80-251-1018-4.

DOMES, Martin. Tvorba webových stránek. 1.vyd. Brno: Computer Press, a. s., 2006. 192 s. ISBN: 80-251-0920-8.

GUTMANS, Andi. Mistrovství v PHP 5 . 1.vyd. Brno: Computer Press, a.s.,2005. 520 s. ISBN 80-251-0799-X.

KRUG, Steve. Webdesign - Nenuťte uživatele přemýšlet. 2.vyd. Brno: Computer Press, a.s., 2006. 168 s. ISBN: 80-7226-892-9.

Vedoucí bakalářské práce: doc. Ing. Miloš Koch, CSc.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2010/2011.

L.S.

Ing. Jirí Kříž, Ph.D.
Ředitel ústavu

doc. RNDr. Anna Putnová, Ph.D., MBA
Děkan fakulty

V Brně, dne 16.05.2011

Abstrakt

Cílem této práce je návrh a zhotovení internetové stránky s administračním rozhraním pro občanské sdružení Bá-Bi. Zabývá se absencí internetové reklamy občanského sdružení a poskytuje návrh na nové internetové stránky, které by sloužily jako reklama pro sdružení, ale také jako informační portál pro návštěvníky stránek.

Klíčová slova

Internetová stránka, občanské sdružení Bá-Bi, HTML, XHTML, CSS, PHP, MySQL, SQL, Javascript

Abstract

The aim of this work is the design and construction a new web pages with administration interface for civic association Bá-Bi. It deals with the lack of internet advertising of this civic association and provides a proposal for a new web pages that should serve as an advertisement for association, but also as an information portal for web pages visitors.

Key Words

Web page, civic association Bá-Bi, HTML, XHTML, CSS, PHP, MySQL, SQL, Javascript

Bibliografická citace práce:

VIČAR, Z. *Návrh internetových stránek*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2011. 62 s. Vedoucí bakalářské práce doc. Ing. Miloš Koch, CSc..

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne

.....

Poděkování

Tímto bych velice rád poděkoval vedoucímu bakalářské práce, panu doc. Ing. Miloši Kochovi, CSc. za čas, který mi věnoval a za rady a připomínky, které mi poskytnul při zpracovávání této práce. Dále bych rád poděkoval své rodině a blízkým, za trpělivost, kterou se mnou v období psaní této práce měli.

Obsah

ÚVOD.....	10
VYMEZENÍ PROBLEMATIKY A CÍLE PRÁCE.....	11
VYMEZENÍ PROBLEMATIKY	11
CÍL PRÁCE.....	11
1 TEORETICKÁ VÝCHODISKA PRÁCE	12
1.1 STATICKE INTERNETOVÉ STRÁNKY	12
1.1.1 HTML.....	12
1.1.1.1 Historie a vývoj	12
1.1.1.2 Základy HTML a syntaxe.....	13
1.1.2 XHTML	16
1.1.2.1 Historie a vývoj	16
1.1.2.2 Syntaxe a rozdíly oproti HTML	17
1.1.3 CSS.....	20
1.1.3.1 Historie a vývoj	20
1.1.3.2 Způsoby použití CSS a syntaxe	22
1.2 DYNAMICKÉ INTERNETOVÉ STRÁNKY	26
1.2.1 PHP.....	26
1.2.1.1 Historie a vývoj	26
1.2.1.2 Nástroje pro práci s PHP	27
1.2.1.3 Základy PHP a syntaxe.....	28
1.2.2 MySQL	30
1.2.2.1 Charakteristika MySQL.....	30
1.2.2.2 Základy SQL.....	30
1.2.2.3 Propojení s PHP.....	33
1.2.3 Javascript.....	34
1.3 OPTIMALIZACE PRO VYHLEDÁVAČE – SEO.....	35
1.3.1 „On-page“ faktory.....	35
1.3.2 „Off-page“ faktory	36
1.3.3 SEO a etika	36
2 ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE	38

2.1	ABSENCE INTERNETOVÉ PREZENTACE.....	38
2.2	POŽADAVEK NA VYTVOŘENÍ PREZENTACE.....	38
3	VLASTNÍ NÁVRHY ŘEŠENÍ, PŘÍNOS NÁVRHŮ ŘEŠENÍ.....	39
3.1	NÁVRH DATABÁZE.....	39
3.2	VOLBA VZHLEDU	42
3.2.1	<i>Barevné schéma</i>	42
3.2.2	<i>Layout stránky</i>	43
3.3	STRUKTURA STRÁNKY	45
3.3.1	<i>Aktuality</i>	45
3.3.1.1	Správa aktualit.....	47
3.3.2	<i>Kniha návštěv</i>	48
3.3.3	<i>Video</i>	49
3.3.3.1	Správa videí.....	50
3.3.4	<i>Fotogalerie</i>	51
3.3.4.1	Správa fotogalerií	53
3.3.5	<i>Lektoři</i>	54
3.3.5.1	Správa lektorů.....	55
3.3.6	<i>Kontakt</i>	56
3.4	PROPOJENÍ SE SOCIÁLNÍ SÍTÍ FACEBOOK	57
3.5	EKONOMICKÉ ZHODNOCENÍ	58
4	ZÁVĚR	59
5	POUŽITÉ ZDROJE	60
5.1	KNIŽNÍ ZDROJE	60
5.2	ELEKTRONICKÉ ZDROJE	60
6	SEZNAMY	62
6.1	SEZNAM OBRÁZKŮ	62
6.2	SEZNAM TABULEK	62

Úvod

Žijeme v době, kdy se téměř každý již setkal se slovem internet. Většina z těchto lidí internet zná blíže, vědí, co to internet znamená, nebo jej denně používá k práci, studiu či zábavě. Vzhledem k tomuto faktu, je dnes pro začínající firmu velice důležité mít na internetu místo, které by sloužilo jako shromaždiště informací o firmě a zároveň tak fungovalo jako internetová reklama.

V této práci se budu zabývat návrhem a následnou tvorbou internetových stránek pro začínající občanské sdružení Bá-Bi. Sdružení by stránky potřebovalo nejen jako internetovou reklamu, ale zároveň by tyto stránky měly sloužit jako informační portál, kde budou pravidelně vystavovány články s aktuálním děním ve sdružení.

Vymezení problematiky a cíle práce

Vymezení problematiky

Občanské sdružení Bá-Bi se zabývá pořádáním dětských táborů a kulturních společenských akcí. Kontaktoval mě jeden ze zakladatelů sdružení s požadavkem na vytvoření internetových stránek. Přesněji po mě požadovali vytvoření internetových stránek s administračním rozhraním, kam by bylo možné se přihlásit a následně tyto stránky do jisté míry upravovat. Mezi požadované úpravy patří například psaní článků a jejich následná editace a mazání, přidávání a úprava táborových lektorů, přidávání a úprava videí z akcí, fotogalerie a návštěvní kniha.

Cíl práce

Cílem bakalářské práce je návrh a vytvoření internetových stránek tak, aby vyhovovaly požadavkům občanského sdružení Bá-Bi. Hlavní cíl práce je možné rozdělit na více dílčích cílů:

- Návrh MySQL databáze, která bude sloužit jako zdroj dat pro internetové stránky
- Návrh grafického vzhledu stránky s ohledem na cílovou skupinu návštěvníků a obor občanského sdružení
- Návrh internetových stránek s důrazem na intuitivnost a jednoduché ovládání

1 Teoretická východiska práce

Teoretická část práce je rozdělena do tří částí. Zatímco první část se bude zabývat nástroji, které jsou potřebné pro tvorbu statických internetových stránek, v části druhé představím nástroje používané k tvorbě dynamických internetových stránek. V poslední části si povíme, co to znamená SEO optimalizace webových stránek a jak ji na stránkách udělat.

1.1 Statické internetové stránky

Pod pojmem statické internetové stránky si lze představit takové stránky, které jsou neměnné a zůstanou napořád v takové podobě, ve které je autor zanechal. Jedinou možností úpravy takových stránek je potom zásah do zdrojového kódu stránky. (9)

Pro tvorbu statických internetových stránek se používá zejména jazyků HTML, XHTML a CSS, o kterých se podrobně rozepisují níže.

1.1.1 HTML

1.1.1.1 Historie a vývoj

Jazyk HTML (HyperText Markup Language – hypertextový značkovací jazyk) má na svědomí Tim Berners-Lee, který pro potřeby organizace CERN (Centre Européene de Rechere Nucléaire, Evropské centrum jaderného výzkumu) vytvořil v roce 1991 první definici tohoto jazyka. Tato verze je dnes známá jako HTML 0.9. Tato verze podporovala pokročilejší úpravy v textu, jako rozdělení textu do několika logických úrovní, několik druhů zvýraznění, vkládání obrázků a odkazů do textu.

Nicméně s rostoucími požadavky uživatelů www byla nutnost stávající verzi HTML obohacovat o nové prvky. V rámci zachování kompatibility mezi jednotlivými modifikacemi HTML vytvořil Berners-Lee návrh standardu HTML 2.0, který v sobě obsahoval všechny v té době používané prvky HTML. Tato verze měla dvě úrovně. Zatímco úroveň 1 (Level 1) pouze málo rozšiřovala předchozí verzi HTML, úroveň 2 (Level 2) již definuje práci s formuláři.

Další rozšíření jazyka je známé pod názvem HTML+, které už umělo vytvářet tabulky a matematické vzorce. Také se zde objevovali prvky umožňující preciznější kontrolu výsledného textu – vylepšené obtékání obrázků textem a styly dokumentů. Na jaře roku 1995 Dave Raggett z laboratoří Hewlett-Packard formalizoval HTML+ a

vytvořil jeho deklaraci, tzv. DTD (Document Type Declaration). Tím vznikl návrh pro novou verzi HTML 3.0. Bohužel kompletní podporu pro všechny prvky HTML 3.0 nabízel pouze experimentální prohlížeč Arena, který byl k dispozici pouze pod operačním systémem Unix.

Vzhledem k velmi špatné podpoře všech prvků verze 3.0, již na jaře roku 1996 bylo jasné, že přechod na tuto verzi byl příliš velký krok. Proto se členové W3C, které již v té době koordinovalo vývoj HTML standardů, na rozšíření původní verze HTML 2.0. Tímto vznikl jazyk HTML 3.2, který ani z daleka neobsahoval tolik prvků a možností jako verze 3.0. Z verze 3.0 si přenesl jen část tabulkových prvků, ostatní prvky jsou tvořeny směsí, kterou v té době podporovali nejnovější prohlížeče. (6)

Dalším krokem ve vývoji jazyka byla verze HTML 4.0, vydaná na konci roku 1997. (17) Tato verze se snažila dosáhnout původního účelu, kdy měly prvky označovat význam (sémantiku) jednotlivých částí dokumentu a výsledný vzhled má být ovlivněn připojeným stylovým předpisem. Pro tyto účely sloužily kaskádové styly dokumentů (CSS), které organizace W3C doporučuje již od konce roku 1996.

Následovala verze HTML 4.01, vydaná koncem roku 1999. (17) Tato verze pouze opravovala některé chyby v předchozí verzi a předpokládalo se, že se bude jednat o poslední verzi jazyka HTML. Nástupcem se měl stát jazyk XHTML.

Nicméně 7. března 2007 (19) vznikla nová pracovní skupina lidí, která měla za cíl vytvořit novou verzi HTML. Termín vypuštění nové HTML verze 5 do světa je nastaven na rozmezí let 2010-2012 a předpokládané ukončení vývoje této verze je plánováno na rok 2022. (12)

1.1.1.2 Základy HTML a syntaxe

Když už je historie úspěšně za námi, můžeme se blíže podívat na samotnou podobu jazyka HTML. Základním stavebním kamenem celého jazyka jsou takzvané „tagy“ neboli značky. Ty poznáme tak, že jsou uzavřené v hranatých závorkách. Některé tagy jsou párové, které musí být ukončené uzavíracím tagem, ostatní jsou nepárové a žádný uzavírací tag nepotřebují. Názorně si toto vysvětlíme na příkladu.

```
<b>tučně psaný text</b>  
<br>  
<i>text psaný kurzívou</i>
```

Tento jednoduchý příklad používá nejprve párový tag `` ukončený ``. Tento tag se používá pro tučné zvýraznění písma – vše, co je uvnitř tohoto tagu bude psáno tučně. Následuje tag `
`, který funguje v HTML jako odřádkování a celý příklad je ukončený dalším párovým tagem `<i>` a `</i>`, který text v něm uzavřený zvýrazňuje kurzívou. Výsledek by vypadal nějak takto:

tučně psaný text
text psaný kurzívou

Samotné tagy je možno také vnořovat do sebe a aplikovat tak efekt více tagů zároveň na jednu část textu. Ukážeme si opět jednoduchý příklad, ve kterém do nám již známého párového tagu `` uzavřeme text obalený v dalším párovém tagu `<i></i>`. V kódu HTML by tento příklad byl zapsán takto:

```
<b><i>tučně psaný text kurzívou</i></b>
```

A výsledek bude vypadat následovně:

tučně psaný text kurzívou

U vnořování tagů je ale nutné podotknout, že se tagy nesmí navzájem křížit. Na našem příkladu by to znamenalo, že nesmíme ukončit tag `` dříve, než je ukončený tag `<i>`.

Některé tagy mohou mít také upravitelné vlastnosti, hovoříme pak o „atributech tagu“. Například tag `` má jeden z atributů „face“, který určuje použitý font textu. V kódu HTML by se vše zapsalo takto:

```
<font face="Times New Roman">Times New Roman</font>
```

Atribut jsem v kódu obalil uvozovkami, což je v tomto případě, kdy hodnota atributu obsahuje mezery, povinné. V jiných případech je sice použití uvozovek nepovinné, ale v rámci zachování stejného formátu a předcházení zbytečným chybám se doporučuje psát uvozovky všude.

Další vlastnost jazyka HTML je, že není citlivý na velikost písmen při psaní tagů – tomuto se říká „non case sensitive“. V praxi to znamená, že tagy můžeme psát malými i velkými písmeny a obojí způsob bude správně.

```
<b><i>tučně psaný text kurzívou</i></b>  
<B><I>tučně psaný text kurzívou</I></B>
```

V tomto příkladu je použit nám již známý kód, ale jednou jsou tagy psány malými písmeny a podruhé velkými. Obě varianty HTML akceptuje a jejich výsledek bude naprosto totožný, záleží tedy jen na autorovi kódu, jakou variantu zvolí. Nicméně je doporučeno tagy psát malými písmeny, protože jiné programovací jazyky používané pro tvorbu webu (například Javascript) jsou tzv. „case sensitive“, tudíž pak HTML dokument psaný malými písmeny má sjednocený vzhled i s částmi psanými v Javascriptu.

Než se dostaneme k samotnému vzhledu celého HTML dokumentu, povíme si ještě poslední vlastnost tohoto jazyka, tou je chápání mezer v textu. HTML jazyk totiž chápe tabulátor, odřádkování (například klávesou ENTER) i vložení dvou mezer za sebou, jako jednu mezeru. V praxi to znamená, že jakkoliv velký prázdný prostor v textu HTML kód zobrazí jako jednu mezeru. V případech, kdy z nějakého důvodu potřebujeme vložit mezer více, se používá speciální značka ` `, což je pevná mezera (z anglického „non breaking space“). Tato značka je jednou z mnoha, kterým říkáme entity. Tyto začínají znakem `&` a jsou ukončené středníkem. Entity se používají v případech, kdy potřebujeme do textu vložit nějaký znak, který by HTML chápalo jako část kódu a tudíž by jej nezobrazilo jako text. Typickým příkladem může být vložení části zdrojového kódu do textu tak, aby byl vidět i samotný zdrojový kód, nikoliv jen jeho výsledek. Pro tyto účely se používá entita `<` a `>`, které znázorňují ostré závorky `<` a `>`. Názorně si ukážeme, jak by to mělo vypadat:

```
&lt;b&gt;tučně psaný text&lt;/b&gt;
```

Tato část kódu by měla následující výsledek:

```
<b>tučně psaný text</b>
```

Nyní, když jsme obeznámeni se základy syntaxe jazyka HTML, si můžeme ukázat strukturu HTML dokumentu.

```
<html>
  <head>
    <title></title>
  </head>
  <body>
  </body>
</html>
```

Jak můžeme vidět na ukázce struktury, celý dokument je obalen v párovém tagu `<html></html>`. Uvnitř tohoto tagu je document rozdělen na dvě části – head (neboli hlavička) a body (neboli tělo dokumentu). Hlavička obsahuje různé údaje, v příkladu je v hlavičce uvedený tag `<title></title>`, který značí titulek celé stránky. Do hlavičky se také zapisuje připojování stylového předpisu, javascriptu, aj. Tyto údaje sice nejsou přímo zobrazeny na stránce ve formě textu, ale jsou pro celý dokument velice důležité. Samotné tělo dokumentu pak obsahuje vše, co vidíme na výsledné stránce. (13)

1.1.2 XHTML

1.1.2.1 Historie a vývoj

Ještě než si řekneme o XHTML (Extensible Hypertext Markup Language – rozšiřitelný hypertextový značkovací jazyk) něco podrobnějšího, je nutné si něco málo říci o XML. XML (Extensible Markup Language – rozšiřitelný značkovací jazyk) je obecný značkovací jazyk, který je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Spojením v té době již zastaralého a nevyvíjeného HTML 4.01 a XML, vyvinuli vývojáři z W3C první verzi jazyka XHTML označenou 1.0. Na základě toho vydali 26. ledna 2000 (21) doporučení, nebo také specifikace, které určují správné znění jazyka. V této specifikaci byl jazyk rozdělený na tři verze: XHTML 1.0 Strict, XHTML 1.0 Transitional a XHTML 1.0 Frameset. Zatímco verze Strict, jak již název napovídá, zcela striktně dodržuje specifikace verze 1.0 a nepovoluje do XHTML kódu zapojit žádné značky, které by ovlivnily vzhled dokumentu (toto je řešeno pomocí CSS – o CSS si povíme více v další kapitole), tak verze Transitional a Frameset jsou již mírnější a dovolují používání některých zastaralých značek. Například verze Frameset nám dovoluje v kódu používat staré značky `<frame>` sloužící pro rozdělení stránky do rámců.

Poslední používaná verze XHTML je 2.0, jejíž poslední pracovní návrh byl vydán 26. července 2006. (20) Vývojové práce na této verzi byly ukončeny na konci roku 2009. (19) V současné době vývojový tým W3C pracuje na spojení HTML 5 a XML do nové verze XHTML 5. (16)

1.1.2.2 Syntaxe a rozdíly oproti HTML

Jak už samotný název „rozšiřitelný“ napovídá, bude XHTML obsahovat oproti klasickému HTML nějaké odlišnosti. Nebude se ale většinou jednat o rozšíření, ale spíše o zúžení a osekání původního HTML. XHTML jako takové totiž nepřináší vůbec žádné nové možnosti, ale spíše omezení pro autora kódu. Některá omezení již většina autorů zná z HTML, v XHTML se stali pouze povinnými:

- Používání uvozovek (apostrofů) u atributů
- Zakázáno křížení tagů

Ostatní omezení, či rozdíly, jsou už zcela nové a pro XHTML typické:

- Tagy a atributy psány malými písmeny
- Ukončení nepárových tagů lomítkem
- Každý atribut musí mít nějakou hodnotu
- Jiný zápis interního javascriptu a stylu
- Dokument by měl obsahovat XML prolog
- Dokument vyžaduje správný doctype

Používání uvozovek (apostrofů) u atributů

Jak jsme se dozvěděli výše, v sekci o HTML, používání uvozovek u atributů tagů bylo v HTML povinné jen tehdy, kdy byla hodnota atributu s mezerami. V XHTML se uvozovky musejí používat vždy. Názorně si ukážeme příklad správného i špatného zápisu v XHTML. Následující část kódu by v HTML vložila do stránky obrázek „obr.jpg“ o velikosti 100x100 pixelů.

```
<img src=obr.jpg width=100 height=100>
```

V XHTML by takovýto zápis byl špatný, nebo také nevalidní. Správně by tedy měl tento zápis v XHTML vypadat takto:

```

```

Další možnost, která je XHTML validní, je použití apostrofů místo uvozovek:

```
<img src='obr.jpg' width='100' height='100' />
```

Zakázáno křížení tagů

Zákaz křížení tagů platil i v HTML, takže není vůbec překvapením, že i XHTML toto zakazuje. Opět si ukážeme správné a špatné použití více vnořených tagů. Následující část kódu představuje správné použití více tagů vnořených do sebe.

```
<b><i>tučně psaný text kurzívou</i></b>
```

Nyní si ukážeme, jak by to v žádném případě nemělo vypadat:

```
<b><i>tučně psaný text kurzívou</b></i>
```

Takto zapsaný kód sice většina prohlížečů zobrazí, ale jedná se o nevalidní a nesprávný zápis.

Tagy a atributy psány malými písmeny

Zatímco HTML byl tzv. jazyk non-case sensitive, tedy bylo úplně jedno, jestli bude tag nebo atribut psán malými, či velkými písmeny, u XHTML už toto neplatí. XHTML je tedy case sensitive a navíc podle specifikace, musejí být všechny tagy a jejich atributy psány malými písmeny. Názorně si toto ukážeme na již známém kousku kódu:

```
<IMG SRC=obr.jpg WIDHT=100 heiGHT=100>
```

Takto zapsaný kód je podle HTML naprosto v pořádku. Ale v XHTML by tento zápis musel být upraven do následující podoby:

```

```

Ukončení nepárových tagů lomítkem

Další z odlišností je ukončování všech tagů, a to i nepárových. Z HTML již známe tzv. párové tagy, které obsahují vždy i ukončovací tag se zpětným lomítkem. Nicméně jak v HTML, tak i v XHTML existují také nepárové tagy, které se ovšem v HTML ukončovat nemuseli. V XHTML je tomu jinak. Zatímco takovýto zápis by byl v HTML správně, tak podle XHTML specifikací by byl nevalidní:

```

```

Abychom dosáhli validity v XHTML, musíme tedy tento tag ukončit zpětným lomítkem.

```

```

Každý atribut musí mít nějakou hodnotu

Některé atributy, které se v HTML objevují, nemají přiřazenou hodnotu. Jako příklad nám může posloužit tag select, který se používá pro rozevírací nabídku. Tento tag obsahuje atribut multiple, dovolující vybrání více položek z nabídky. V HTML kódu by byl zapsán následovně:

```
<select name="auta" size="3" multiple>
  <option value="polozka1">polozka1</option>
  <option value="polozka2">polozka2</option>
</select>
```

Pro nás je v tomto příkladu důležitý atribut multiple, který nemá přiřazenou žádnou hodnotu. Toto je v XHTML zakázané a proto bychom museli tento kód upravit do následující podoby:

```
<select name="auta" size="3" multiple="multiple">
  <option value="polozka1">polozka1</option>
  <option value="polozka2">polozka2</option>
</select>
```

Jiný zápis interního javascriptu a stylu

„Skript je v XHTML chápán jako neznámý puding, tedy beztvářý kus dokumentu. Musí se obalit zápisem CDATA. Zápis samotného řetězce CDATA je potřeba ještě javascriptově zakomentovat, aby se nevyhodnocoval.“ (14)

Aby byl tedy zápis skriptu, nebo i stylu XHTML správně, měl by vypadat následovně:

```
<script type="text/javascript">
  /* <![CDATA[ */
  samotný zápis javascriptu
  /* ]]> */
</script>
```

V případě připojení externího skriptu k dokumentu je zápis stejný jako v HTML:

```
<script type="text/javascript" src="externi-skript.js">
</script>
```

Dokument by měl obsahovat XML prolog

Celý XHTML dokument by měl začínat XML deklarací, kde je uvedena verze použitého XML a kódování celého dokumentu. XML deklarace, kdy je dokument kódován pomocí UTF-8 a je použita verze XML 1.0 bude vypadat následovně:

```
<?xml version="1.0" encoding="utf-8"?>
```

Bohužel tuto deklaraci některé starší prohlížeče neakceptují a vyžadují na prvním řádku dokumentu doctype (např. Explorer verze 6). Proto se v praxi dosti často tato deklarace vypouští, i za cenu neplatného XHTML dokumentu.

Dokument vyžaduje správný doctype

Každý dokument XHTML by měl mít správný doctype. Doctype určuje, podle jaké specifikace má být celý dokument zpracován a také, podle jakých pravidel bude kontrolována jeho validita. Proto musí být pro každou verzi XHTML jiný doctype. Uvedeme si zde dvě ukázkové doctype deklarace, používané pro XHTML 1.0 Strict a Transitional.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

1.1.3 CSS

1.1.3.1 Historie a vývoj

Zatímco HTML a XHTML se při tvorbě webových stránek používá zejména pro vytvoření struktury, či obsahu dokumentu, CSS (Cascade Style Sheet – kaskádové styly) se používají pro formátování tohoto dokumentu do výsledné podoby.

„Oddělení struktury dokumentu od jeho layoutu (formátování) bylo cílem HTML od samých počátků v roce 1990. Již Tim Berners-Lee, který je považován za otce WWW, navrhl svůj první prohlížeč NeXT tak, že formátování dokumentu bylo řízeno

jednoduchým zabudovaným stylovým předpisem. Berners-Lee však jazyk tohoto stylového předpisu nikdy nepublikoval a spoléhal na autory prohlížečů, že tuto koncepci sami dále rozvinou. K tomu však, bohužel, nedošlo. První komerční prohlížeč, NSCA Mozaic z roku 1993, naopak okleštil možnosti uživatelského či autorského stylování HTML dokumentů na minimum.“ (10)

Postupem času se web stával čím dál oblíbenější a autorům pomalu začínala chybět možnost svůj HTML dokument jakkoliv upravit. Upravit vzhled dokumentu nebylo možné ani prostou změnou velikosti či barvy písma, což i v té době zvládali všechny textové editory.

Jako první na tento problém reagoval koncem roku 1994, Hakon Wium Lie, který publikoval první koncept „Cascading HTML Style Sheet“, který tehdy podpořil i Dave Raggett, hlavní architekt HTML3.0. Tento návrh se stal základem, ze kterého vycházejí kaskádové styly dodnes.

Bohužel, celou snahu o oddělení formátování od struktury dokumentu, kterou vynaložil Hakon Wium Lie zhatila první verze prohlížeče Netscape. Místo podpoření této snahy totiž Netscape zabudoval základní formátovací prvky přímo do jazyka HTML. *„Jak se záhy ukázalo, pro vývoj webu to znamenalo téměř katastrofu, neboť střet těchto dvou koncepcí zásadně poznamenal kompatibilitu a použitelnost webových dokumentů na mnoho let dopředu a jeho negativní důsledky neseme dodnes.“ (10)*

Po dlouhých diskusích, které se týkaly zejména koncepce kaskádových stylů, která navrhovala tzv. uživatelské stylové předpisy (tedy poslední slovo o formátování dokumentu by měl mít uživatel, nikoliv autor), byla v prosinci 1996 zveřejněna Specifikace kaskádových stylů 1. Tentokrát šlo o oficiální specifikaci, která měla podporu a doporučení přímo od konsorcia W3C, které chápalo kaskádové styly jako součást rozvoje HTML.

Pracovní skupina W3C zaměřená přímo na vývoj kaskádových stylů, je pak nadále rozvíjela a v květnu 1998 vydala doporučení Specifikace kaskádových stylů 2, která se používá dodnes.

Aktuálně se dokončuje doporučení CSS 2.1, které bylo zatím publikováno ve formě návrhu, a probíhají práce na verzi 3.

1.1.3.2 Způsoby použití CSS a syntaxe

Abychom mohli dokument upravit pomocí kaskádových stylů, musíme stylový předpis nějakým způsobem k dokumentu připojit. Ukážeme si zde tři nejpoužívanější metody.

První metodou je připojení stylového předpisu, který máme uložený v externím (samostatném) souboru. Tento způsob je hodně doporučovaný a osobně jej mám nejraději. Umožňuje jednoduchou správu celého webu, protože veškeré styly jsou umístěny na jednom místě, a nejsou rozesety po celém HTML dokumentu. Samotné připojení stylového předpisu se provádí pomocí prvku `<link>`, který se vkládá do hlavičky dokumentu (`<head>`). Prvek obsahuje pět atributů:

- **href** – tento atribut určuje cestu, kde se externí CSS soubor nachází
- **type** – atribut určující typ obsahu cílového souboru, v tomto případě musí mít vždy hodnotu `text/css`
- **rel** – určení typu vazby dokumentu k souboru, v tomto případě může mít dvě hodnoty
 - **stylesheet** – pro základní stylové předpisy
 - **alternative stylesheet** – pro alternativní stylové předpisy
- **media** – určuje médium, nebo média, pro které je cílový soubor určen. Atribut je nepovinný a používá se například při potřebě nastavit odlišný vzhled pro tisk.
- **title** – titulek daného stylového předpisu, také nepovinný atribut používaný hlavně u alternativních předpisů

Nyní si pár ukázek, jak by měl celý zápis vypadat. První zápis je obyčejný a připojí nám předpis ze souboru `style.css` do dokumentu. Druhý zápis je již s atributem `media`, tudíž se použije jen pro tisk dokumentu – pro tento účel bude připojen soubor `tisk.css`.

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

```
<link rel="stylesheet" type="text/css" href="tisk.css"
media="print">
```

Druhý způsob připojení stylů k dokumentu je pomocí prvku `<style>`. Tento způsob, se podobně jako předešlý, vkládá do hlavičky dokumentu (`<head>`), a má tři atributy, které již známe z prvku `link` – `type`, `media`, `title`. Samotný stylový předpis je

pak obsažen uvnitř párového tagu `<style>` `</style>`. Následující příklad ukazuje použití tohoto způsobu.

```
<style type="text/css">
  body {
    color: black;
    font-family: Arial;
    background: black;
  }
</style>
```

Třetím způsobem je přiřazení stylů jednotlivým prvkům pomocí atributu `style`. Osobně tento způsob nedoporučuji a využívám jen velice zřídka, protože je pak velice obtížné dokument s takto přiřazenými styly upravovat. V případě používání XHTML 1.1 nebo novější, je pak tento způsob dokonce nevalidní. Opět si ukážeme pár příkladů, jak by takovéto použití mohlo vypadat.

```
<p style="color: black; font-family: Arial"> Text psaný
černou barvou a fontem Arial.</p>
```

Syntaxe kaskádových stylů, tak jak je uvedena ve specifikaci, je velice obsáhlá a její obsažení v této práci by bylo zbytečné. Proto se zaměřím spíše na základní prvky syntaxe.

Stylový předpis se skládá z jedné či více definic. Rozlišujeme dva druhy definic, at-pravidla a sady pravidel. Pro naše potřeby se zaměříme na sady pravidel, někdy také nazývány jen jako pravidla. Ty se skládají ze selektoru, za kterým následuje blok deklarácí. Tento blok deklarácí je uzavřen ve složených závorkách a obsahuje vlastnost a hodnotu vlastnosti, oddělené dvojtečkou. Pro lepší pochopení zde uvedu část kódu, kterou jsem použil výše v jednom z příkladů a následně jej podrobně vysvětlím.

```
body {
  color: black;
  font-family: Arial;
  background: black;
}
```

V této části kódu můžeme vidět jedno pravidlo. Podle použitého selektoru „body“ můžeme říci, že toto pravidlo se bude týkat těla stránky. Následuje blok deklarácí

uzavřený ve složených závorkách { }. První deklarace se týká vlastnosti „color“ (tedy barva textu) a hodnota vlastnosti je „black“. Tato deklarace nám tedy říká, že barva textu na stránce bude černá. Další deklarace má vlastnost „font-family“ a hodnotu „Arial“ – toto znamená, že použitý textový font v dokumentu bude Arial. Poslední deklarace má vlastnost „background“ a hodnotu „black“, což znamená, že pozadí dokumentu bude černou barvou.

Další, poměrně často využívanou věcí v CSS, jsou selektory tříd a ID, které slouží pro vyčlenění jednoho či více prvků v dokumentu, kterému pak následně přiřadíme nějaké specifické formátování.

Selektory tříd slouží pro případy, kdy máme v dokumentu více prvků, ale některým z nich potřebujeme nadefinovat jiný formát. Těmto prvkům pak přiřadíme třídu a pomocí selektoru třídy pak nadefinujeme formát pouze pro tyto prvky. Použití tříd si ukážeme na příkladu, kdy budeme mít v dokumentu dva nadpisy, každý následovaný dvěma odstavci. Budeme chtít, aby první nadpis s odstavci byl formátovaný jinak, než nadpis druhý s odstavci. Zavedeme si proto třídu „a“, kterou přiřadíme prvnímu nadpisu a jeho odstavcům, a dále si zavedeme třídu „b“, kterou přiřadíme zbylému nadpisu a jeho odstavcům. Zápis v HTML by vypadal takto:

```
<h1 class="a">Nadpis</h1>
<p class="a">Odstavec textu</p>
<p class="a">Odstavec textu</p>
<h1 class="b">Nadpis</h1>
<p class="b">Odstavec textu</p>
<p class="b">Odstavec textu</p>
```

Následně napíšeme stylový předpis, který první nadpis a jeho odstavce nabarví červenou barvou a druhý nadpis a jeho odstavce nabarví na barvu modrou.

```
.a{
color: red;
}
.b{
color: blue;
}
```

Na rozdíl od tříd, slouží selektory ID pro vyčlenění jednoho prvku z dokumentu, kterému se přidělí jednoznačný identifikátor – ID. Tento identifikátor musí být jedinečný, to znamená, že žádný jiný prvek nesmí mít stejné ID. Poté můžeme tomuto

prvku pomocí stylového předpisu určit formátování. Jako příklad si můžeme uvést situaci, kdy máme v dokumentu tři odstavce textu, ale první z nich chceme mít naformátovaný odlišným způsobem. Tomuto odstavci pak přiřadíme jedinečný identifikátor, pomocí kterého pak napíšeme stylový předpis platný pouze pro tento odstavec. Podívejme se nejdříve na zápis v HTML:

```
<p id="prvni">Odstavec textu</p>
<p>Odstavec textu</p>
<p>Odstavec textu</p>
```

Nyní máme první odstavec s identifikátorem „první“ a napíšeme na něj stylový předpis, který tento odstavec obarví na červenou barvu.

```
#prvni{
color: red;
}
```

Další, a zároveň poslední věc, kterou si tu ukážeme, jsou pseudotřídy a dynamické pseudotřídy. Tyto mechanismy CSS uvádím zejména proto, že jsou hojně používané. Typický příklad použití pseudotřídy je navštívený a nenavštívený odkaz. Na mnoha stránkách na internetu si lze všimnout, že pokud jsme nějaký odkaz již navštívili, bude jeho formátování odlišné od ještě nenavštíveného odkazu – nejčastěji se mění barva písma. S tím souvisí i použití dynamické pseudotřídy, která se zase používá například pro změnu formátování textu odkazu, po najetí kurzorem myši na tento odkaz. Použití obou mechanismů si ukážeme právě na zmiňovaném odkazu. Nejdříve uvedu HTML kód odkazu na internetovou stránku Seznam.cz, a poté tomuto odkazu přiřadím stylový předpis s různým formátováním navštíveného, nenavštíveného odkazu a formátem po najetí kurzorem.

```
<a href="http://www.seznam.cz">Seznam.cz</a>

a.link{
color: red;
}
a.visited{
color: blue;
}
a.hover{
color: yellow;
}
```

Nyní bude nenavštívený odkaz červenou barvou, navštívený odkaz modrou barvou a pokud nad odkaz najedeme kurzorem myši, přebarví se na žlutou.

1.2 Dynamické internetové stránky

„Stránkám, které umožňují měnit obsah stránky, se říká dynamické webové stránky. Aby vůbec šlo provozovat dynamické webové stránky, musí vstoupit do hry ještě tzv. skriptovací jazyk, což je v podstatě programovací jazyk, který se podílí na vytvoření webové stránky nebo mění její obsah. Dynamiku do webové stránky tedy přidává skriptovací jazyk. Existují dva druhy skriptovacích jazyků. Jsou to jednak skriptovací jazyky, o které se stará prohlížeč (tedy klient) – ty se nazývají klientské skriptovací jazyky. A pak jsou ty, o které se stará server – ty se logicky nazývají serverové skriptovací jazyky. Klientské skripty umějí spíše vizuální efekty, různé animace atd. Serverové skripty naopak umožňují programovat plnohodnotné aplikace spolupracující s databázemi, poštou atd.“ (9)

Níže si popíšeme jeden z nejrozšířenějších serverových skriptovacích jazyků – PHP. Ten se nejčastěji používá v kombinaci s databázemi MySQL, o kterých si také něco málo povíme. Celou část o dynamických stránkách ukončíme Javascriptem, který je hojně používán zejména k vizuálním efektům na stránkách.

1.2.1 PHP

1.2.1.1 Historie a vývoj

„U zrodu celého systému byl původně soukromý program Rasmuse Lerdorfa. Napsal si jednoduchý systém pro své vlastní webové stránky, především pro sledování návštěvnosti. Kvůli zvýšení výkonnosti jej později přeprogramoval v C a uvolnil k používání pro několik svých známých. Těm se velmi zalíbil a požadovali stále nové vlastnosti a chodili s novými připomínkami, čímž udrželi systém při živoě a při vývoji. Při jeho uvolnění pro používání mu byl dán název Personal Home Page, PHP. Velké obliby se dočkal především ve své druhé verzi, obohacené především o velmi snadnou manipulaci s daty z formulářů – PHP/FI 2.0. FI v této zkratce znamenalo právě Form Interpreter.“ (1)

Když se v roce 1997 dvojice Andi Gutmans a Zeev Suraski snažili použít PHP/FI při vývoji komerčních aplikací, zjistili, že je pro tyto účely nevhodný a rozhodli

se tento stav změnit. Kompletně jazyk přepsali a tento výsledný jazyk označili jako PHP verze 3.

Další pokrok nastal v zimě 1998, kdy došlo k přepsání jádra PHP za účelem zvýšení výkonu pro složité aplikace. Nově vzniklé jádro bylo podle počátečních písmen jeho autorů pojmenováno Zend. Další verze, PHP 4, byla představena v roce 2000 a fungovala již na tomto novém jádru.

Jádro Zend se dočkalo dalšího vylepšení v roce 2004 a bylo použito pro novou verzi PHP 5, ve které bylo hlavní změnou vylepšení objektově orientovaného programování. Dnes je PHP 5, konkrétně verze PHP 5.3.5, poslední používaná verze a předešlá verze 4 již není podporována (i když některé free servery stále zastaralou verzi 4 používají). Aktuálně se pracuje na vývoji nové verze PHP 6, ale datum oficiálního vydání není známo. (9)

1.2.1.2 Nástroje pro práci s PHP

Vzhledem k faktu, že PHP je serverový skriptovací jazyk, tak pro práci s PHP budeme potřebovat webový server s podporou PHP. Jedna varianta spočívá v použití volně dostupných serverů, které se dají nalézt na internetu. Velice vhodný je server <http://www.php5.cz/>, který nabízí hosting stránek zdarma. Tento server je velice vhodný pro programátory a vývojáře, jelikož nabízí podporu nejnovějších verzí PHP a MySQL. V případě, že z nějakého důvodu nemůžeme, nebo nechceme používat tyto veřejné webové servery, existuje druhá varianta – webový server Apache, který si můžeme nainstalovat na vlastní počítač a zprovoznit si tak soukromý webový server. Tento webový server si můžete buďto nastavit sami – tuto variantu bych doporučoval pouze pro zkušené uživatele, nebo je možné si stáhnout hotové řešení, které lze nainstalovat pomocí velice přehledně zpracovaného průvodce. Jedno z takových řešení je například Complex Web Server. Tento program obsahuje kromě samotného webového serveru i databázi MySQL včetně databázového správce phpMyAdmin. Pro účely testování a tvorby dynamických webových stránek s podporou databáze MySQL je tento program více než vhodný.

Dalším nástrojem pro práci s PHP je textový editor pro editaci skriptů PHP. Tento nástroj není nutný, protože PHP skripty lze psát v libovolném textovém editoru – například poznámkový blok (notepad), který je součástí každé instalace systému Windows, poslouží pro tyto účely dobře. I přes tento fakt je ale velice doporučené

pracovat s PHP skripty v nějakém textovém editoru, který je na toto přímo stavěný. Takovéto editory nabízí velmi potřebné vlastnosti, jako například: otevření více souborů najednou, nebo zvýraznění syntaxe. Zvláště vlastnost zvýraznění syntaxe je neocenitelným pomocníkem při práci s komplexnějšími skripty, ve kterých se bez této vlastnosti lze snadno ztratit. Editorů, které toto vše zvládají, je mnoho, ale já osobně používám editor PSPad.

1.2.1.3 Základy PHP a syntaxe

Vzhledem k veliké obsáhlosti jazyka PHP a obrovskému množství různých funkcí, se v této práci nebudeme zabývat kompletním výčtem všeho, co PHP nabízí. Nejen, že by celý tento výčet znamenal publikaci o několika stech stranách, ale většina z těchto informací by byla nad rámec této práce. Proto se zaměřím pouze na základy PHP, jako je syntaxe jazyka nebo způsoby vkládání kódu do HTML.

Jednotlivé PHP příkazy jsou v HTML kódu obaleny dvěma značkami – otevírací a uzavírací. Tyto značky je možné zapsat čtyřmi způsoby. První a zároveň nejjednodušší způsob je uzavření kódu do značek: `<? ?>`. Tento způsob nefunguje vždy, záleží na nastavení PHP, které používáme. Názorná ukázka takto obaleného kódu:

```
<?
echo "první způsob";
?>
```

Druhý způsob je v podstatě jen delší verze předchozího způsobu, nicméně je nejdoporučovanější z důvodu, že funguje naprosto všude a je v souladu s HTML standardy.

```
<?php
echo "druhý způsob";
?>
```

Třetí způsob je velice podobný zápisu Javascriptu, podobně jako druhý způsob, i tento funguje vždy, nicméně nebývá často vídán.

```
<script language="php">
echo "třetí způsob";
</script>
```

Konečně poslední a zároveň nejméně používaný způsob, který je převzatý z jazyka ASP.

```
<%  
echo "čtvrtý způsob";  
>
```

Nyní je ten pravý čas si říci něco i o samotné syntaxi PHP. Jak jsem již dříve zmínil, nebudu zde vypisovat syntaxi všech funkcí, které v PHP existují, ale zaměřím se pouze na naprosté základy. Pro naše účely nám naprosto postačí vysvětlení základů na příkazu `echo`, který je velice hojně využíván. Tento příkaz slouží pro obyčejný výpis čehokoliv na stránku. Přejdeme tedy k příkladu:

```
<?php  
echo "Příklad příkazu echo.";  
?>
```

Celý příkaz je uzavřen ve značkách, které jsou nám již známé, jedná se o druhý způsob, jakým je možné PHP kód oddělit od jiného. Následuje samotný příkaz `echo`, mezera a samotný text, který chceme vypsát, obalený uvozovkami. Celý příkaz je ukončený středníkem, který označuje konec příkazu a případný začátek nového. Tento příklad by na stránce vypsal následující:

Příklad příkazu `echo`.

Poslední, a dle mého názoru velice důležitá věc, kterou si zde ukážeme, jsou komentáře. Komentáře existují v každém jazyce a slouží pro pochopení PHP kódu. V případě, že nějaký autor napíše svůj skript, bude pro ostatní velice obtížné se v něm vyznat a zorientovat. Pro tyto případy jsou zde právě komentáře, do kterých může autor napsat vysvětlivky ke skriptu tak, aby byly pro ostatní snadno pochopitelné. Zároveň takové komentáře mohou posloužit i samotnému autorovi, který se pak ve složitějších skriptech snáze orientuje. Existuje více způsobů, jakými můžeme komentáře do kódu vložit. Pokud chceme zapsat komentář na jeden řádek, můžeme si zvolit mezi dvěma způsoby, první z nich je uvozen dvojicí lomítek „//“ a druhý z nich je uvozen znakem mřížky „#“. V případě, že chceme napsat větší komentář, který by se na jeden řádek nevešel, musíme celý komentář uzavřít do značek „/*“ a „*/“. Celé si toto ukážeme na příkladu:

```

<?
echo "Příklad příkazu echo";
// komentář vložený prvním způsobem
# komentář vložený druhým způsobem
/*
Komentář, který
je rozložen, do více
řádků.
*/
?>

```

1.2.2 MySQL

1.2.2.1 Charakteristika MySQL

„MySQL je databázový server vytvořený komerční firmou, který se etabloval jako všudypřítomný server pro webové aplikace. MySQL není ani nejlepší databáze, ani nejkvalitnější, a dokonce není ani za všech okolností zadarmo. Přesto za její používání pro PHP na webové stránky platit nemusíte, ale při používání MySQL třeba v komerčních programech ano.

Ve srovnání s jinými databázovými systémy patří MySQL spíše k těm jednoduchým, ale zase je nenáročný na zdroje počítače a u některých operací nabízí zvýšení rychlosti. Jednoduše shrnuto, MySQL je malý, rychlý, jednoduchý a nenáročný databázový systém.“ (9)

Pokud potřebujeme s MySQL pracovat, musíme vědět, jakým způsobem s tímto systémem komunikovat. Tímto komunikačním prostředkem, je jazyk SQL (Structured Query Language – Strukturovaný dotazovací jazyk). Jazyk SQL je používán v mnoha jiných databázových systémech, a tak je jeho znalost při práci s databázemi velice důležitá. Jeden z nástrojů, o kterých jsem se zmiňoval výše - phpMyAdmin, sice dovoluje práci s databázemi i bez nutnosti ovládat jazyk SQL, nicméně při propojení samotného PHP a MySQL je zapotřebí používat dotazy psané právě v jazyce SQL. Z tohoto důvodu si zde uvedeme alespoň základní dotazy.

1.2.2.2 Základy SQL

Základní dotazy jazyka SQL si vysvětlíme postupně na úkolech, které na sebe budou navazovat. Ukážeme si zde, jak vytvořit databázi, poté vytvoříme tabulku, kterou naplníme daty a na závěr si ukážeme i nějaké dotazy sloužící pro výběr dat z tabulky.

Začneme tedy vytvořením databáze. Pokud chceme, aby naše webové stránky spolupracovaly s nějakou databází, jedná se nám zpravidla o nějaký způsob úložiště dat – tabulky – ve kterých bude uloženo to, co potřebujeme. Abychom ale tyto tabulky mohli využívat, je zapotřebí vytvořit prostor, ve kterých budou uloženy samotné tabulky s daty, tomuto prostoru se říká databáze. Pro tyto účely slouží dotaz „CREATE DATABASE“. Vytvoříme si tedy databázi, kterou pojmenujeme „test_databaze“.

```
CREATE DATABASE test_databaze
```

Nyní máme úložiště, do kterého můžeme začít vytvářet tabulky. K tomu se používá dotaz „CREATE TABLE“, který má následující syntaxi.

```
CREATE TABLE jmeno_tabulky(  
    sloupec1 datovy_typ,  
    sloupec2 datovy_typ  
)
```

Tento dotaz je již trochu složitější, takže si jej vysvětlíme podrobněji. Dotaz začíná slovy CREATE TABLE, za kterými následuje meze a jméno tabulky. Dále jsou v závorkách vypsány všechny sloupce tabulky s jejich příslušnými datovými typy. Datových typů je více a liší se podle používaného databázového systému. Vzhledem k tomu, že my pracujeme s MySQL, popíšeme si zde nejpoužívanější datové typy právě v tomto systému:

Textové datové typy

- **CHAR(velikost)** – datový typ pro řetězec o přesně stanovené délce (maximální délka řetězce je 255 znaků)
- **VARCHAR(velikost)** – datový typ pro řetězce s proměnlivou délkou, uvedená velikost je maximální možná délka (maximální délka řetězce je 255 znaků)
- **TEXT** – datový typ pro řetězce, které mohou být dlouhé až 65 535 znaků
- **ENUM** – datový typ, ve kterém je možno nastavit seznam možných hodnot, kterých může položka nabývat, typické použití je při uchování informací o pohlaví uživatelů – položka může nabývat dvou předem stanovených hodnot (muž/žena)

Číselné datové typy

- **INT(velikost)** – datový typ pro ukládání celých čísel

- **FLOAT(velikost,n)** – datový typ pro ukládání čísel s desetinnou čárkou, počet číslic za čárkou je udán parametrem „n“

Časové datové typy

- **DATE** – datový typ pro uložení data ve formátu YYYY-MM-DD (rok – měsíc - den)
- **TIME** – datový typ pro uložení času ve formátu HH:MM:SS (hodina : minuta : vteřina)
- **DATETIME** – datový typ pro uložení data i času ve formátu YYYY-MM-DD HH:MM:SS

S datovými typy jsme se již seznámili, takže se můžeme pustit do vytvoření tabulky, která bude zaznamenávat články napsané na webu. Tabulka bude obsahovat položky: id_clanku, jméno_ autora, nadpis_clanku, text_clanku, datum. Položka id_clanku je zavedena z důvodu, kdy potřebujeme jednoznačně identifikovat jeden jediný článek. Položka, která toto umožňuje, se nazývá primární klíč. Primární klíč může být složen i z kombinace několika položek, které by tak tvořili jedinečnou kombinaci pro každý článek, ale z důvodu přehlednosti, se používá položka „id“. Podíváme se tedy, jak by takovýto dotaz vypadal v praxi.

```
CREATE TABLE clanek(
  id_clanku int not null auto_increment,
  jmeno_ autora varchar(50),
  nadpis_clanku varchar(100),
  text_clanku text,
  datum datetime,
  primary key (id_clanku)
)
```

Parametry „not null“ a „auto_increment“ u prvního sloupce tabulky znamenají, že položka musí být vždy zadána (nesmí nabývat hodnoty NULL) a položka se bude automaticky zvětšovat o 1. Automatická inkrementace je proto, abychom později při plnění tabulky daty, nemuseli vždy zadávat hodnotu id ručně. Délky jednotlivých datových typů je potřeba rozumně zvolit, podle předpokládaných hodnot, které mají být v dané položce obsaženy.

Když už máme vytvořenou tabulku, je na čase ji naplnit daty a následně si ukázat, jak tyto data získáme z databáze zpět. Nejdříve tedy k naplnění tabulky, k tomu slouží příkaz „INSERT INTO“. Tento příkaz lze napsat dvěma způsoby. V prvním

z nich v příkazu neuvádíme, do kterých sloupců chceme zapisovat data. Při tomto způsobu je nutné zadat všechny hodnoty tabulky v pořadí, v jakém byla tabulka vytvořena. Není možné nějaké položky vynechat. V našem případě by tento způsob vypadal následovně:

```
INSERT INTO clanek
VALUES('Zdeněk Vičar','Nadpis článku 1','Text článku
1','2010-05-10 11:06:30')
```

Druhý způsob je pro případ, kdy nechceme, nebo z nějakého důvodu nemůžeme vyplnit všechny položky. Pak přímo v dotazu uvedeme, které položky chceme zadat a v jakém pořadí. Tento způsob si uvedeme na příkladu, kdy autor článku neuvede nadpis článku.

```
INSERT INTO clanek(jmeno_ autora, text_clanku, datum)
VALUES('Zdeněk Vičar','Text článku 2','2010-05-10 11:11:30')
```

Poslední dotaz, který si zde ukážeme, slouží pro výpis dat z tabulky. Budeme tedy chtít vypsat všechny záznamy z tabulky „clanek“. K tomuto slouží dotaz „SELECT“.

```
SELECT * FROM clanek
```

Nejdříve uvedeme SELECT, následně zvolíme, které položky chceme vypsat, poté zvedeme FROM a zvolíme, ze které tabulky se mají data vypsat. V tomto příkladu jsem použil při volbě položek znak *, který značí, že chceme vypsat všechny položky. V praxi velice často potřebujeme výsledná data nějak filtrovat, a k tomu slouží například klauzule WHERE. Pokud bychom například potřebovali vypsat všechny články, které napsal autor „Zdeněk Vičar“, dotaz by vypadal následovně:

```
SELECT * FROM clanek WHERE jmeno_ autora='Zdeněk Vičar'
```

1.2.2.3 Propojení s PHP

Když jsme již obeznámeni se základy SQL jazyka, můžeme si ukázat, jak propojíme PHP a MySQL. K tomuto účelu jsou v PHP zabudovány funkce, pomocí kterých můžeme s MySQL komunikovat.

Nejdříve je nutné se k databázi připojit, to provedeme pomocí funkce „mysqli_connect“. Funkce vrací objekt databázového připojení, který je pro další práci s databází vhodné si uchovat. Proto celé připojení uložíme do proměnné \$db_spojení. Celá funkce i s parametry má tuto syntaxi:

```
$db_spojeni = mysqli_connect(adresa_serveru, login, heslo,
jmeno_databaze, port);
```

Následuje zaslání dotazu SQL. K tomuto slouží funkce „`mysqli_query`“. Funkce má dva parametry, první z nich je databázové připojení, které již máme uschované v proměnné z dřívějšíka, druhý z nich je samotný SQL dotaz.

```
mysqli_query($db_spojeni, sql_prikaz);
```

Pokud potřebujeme data z databáze vytáhnout, nepostačí nám samotné zaslání dotazu do MySQL. Proto musíme výsledek funkce `mysqli_query` uložit do proměnné a uchovat si jej. Tímto si uložíme celý výsledek, který nám databáze vrátí. Abychom jej mohli nějak rozumně vypsát na stránku, budeme jen řádek po řádku číst a vypisovat. Ke čtení jednoho řádku slouží funkce „`mysqli_fetch_array`“. Celý kód pro získání dat z databáze a vypsání prvního řádku by vypadal následovně:

```
$vysledek_dotazu = mysqli_query($db_spojeni, sql_prikaz);
$radek = mysqli_fetch_array($vysledek_dotazu);
echo $radek;
```

Nyní zbývá již jen uzavřít spojení s databází pomocí funkce „`mysqli_close`“.

```
mysqli_close ($db_spojeni);
```

1.2.3 Javascript

Javascript vznikl při snaze o spojení tehdejší firmy Netscape Communications Corporation (známá jako Netscape) a Sun Microsystems, Inc. První zprávy o vydání nového jazyka se objevily 4. prosince roku 1995, a před oficiálním vypuštěním Javascriptu, se jazyk jmenoval LiveScript. První verze Javascript 1.0 byla vydána zároveň s prohlížečem Netscape Navigator 2.0. (8)

„Javascript je programovací jazyk interpretovaný, tzn., že se nemusí kompilovat a funguje pouze v prohlížeči. Umí zacházet s časem, ovládat prohlížeč, umožňuje zjišťování přítomnosti tzv. pluginů, případně přesměrovat na jinou stránku kde se nepoužívají. Uživatel může Javascript v prohlížeči zakázat. Největší nevýhodou je, že neumí přistupovat k souborům (kromě cookies) ani k žádným systémovým objektům. To vše z něj dělá pouze druhořadý, účelově použitelný pouze v HTML stránkách. Určité chyby mohou navíc pramenit z odlišných verzí jazyka i prohlížečů.“ (5)

Je to klientský skriptovací jazyk, který se používá zejména k přidání dynamičnosti na webové stránky. Nicméně, vzhledem k jeho vlastnosti, že jej lze uživatelem zakázat, se stal pro můj projekt velice nevhodný. Proto jsem všechny situace, kdy bych za jiných okolností Javascript použil, musel nahradit PHP alternativou, u které mám jistotu, že bude vždy funkční.

1.3 Optimalizace pro vyhledávače – SEO

V dnešní době, kdy je internet doslova zahlcen nespočtem stránek, je velice důležité umět upozornit právě na tu naši stránku. Tímto se zabývá Search Engine Marketing (tzv. SEM), který se dělí na dvě části. První část je placená, která spočívá v uhrazení nějakého poplatku přímo vyhledávači, který pak odkaz na naši stránku umístí například na přední místa, většinou mezi placené odkazy. Druhá část, která se jmenuje Search Engine Optimization (tzv. SEO) je sice také ve většině případů placená, ale tentokrát se neplatí poplatek vyhledávači, ale pouze tomu, kdo SEO na stránkách provádí. Optimalizaci webu pro vyhledávače můžeme dále dělit na „on-page“ a „off-page“ faktory.

1.3.1 „On-page“ faktory

Jak již název napovídá, jedná se o cokoli, co je umístěné přímo na stránce. Jsou to faktory, které jsou přímo ovlivnitelné autorem stránek. Uvedeme si zde některé příklady on-page faktorů.

Klíčová slova

„Výběr klíčových slov je nejdůležitější část SEO. Pokud totiž na stránce není určité klíčové slovo, tak ji pod tímto slovem nemůže vyhledávač ani najít. Celá problematika klíčových slov se zabývá výběrem relevantních vyhledávaných klíčových slov a jejich následnou optimalizací.“ (11)

„Základní pravidlo u klíčových slov je, že se musí hledat relevantní slova k danému webu. Pokud je uživatel na daný web nalákán s tím, že si myslí, že koupí rybičky a ve skutečnosti se octne na webu prodávající stolky, nic si nekoupí a naštve se. Příště již na odkaz na daný web neklikne.“ (11)

Titulek stránky

„Titulek stránky musí být umístěn v hlavičce stránky v párovém tagu <title>. Tvoří jej zpravidla několik slov či frází. Měl by stručně a výstižně uživatele informovat o

obsahu stránky. Pro každou stránku by měl být jedinečný. Titulek je jedním z nejdůležitějších tagů celé stránky – vyhledávače na něj kladou velký důraz. Zobrazuje se též ve výsledcích vyhledávání.“ (15)

Meta tag description

„Meta tag description některé vyhledávače zobrazují u popisku stránky ve výsledcích vyhledávání, takže se vyplatí zde napsat něco smysluplného. Description používá většina vyhledávačů. Stejně jako u titulku je důležité, aby u každé stránky bylo description, které ji nejlépe vystihuje, tedy pro každou stránku odlišné. Doporučená délka je do 250 znaků.“ (11)

Nadpisy na stránce

Nadpisy na stránce by se měli používat pro strukturování textu. Používají se k tomu HTML tagy <H1> až <H6>. Číslo v tomto případě určuje váhu textu uvnitř tagu. Nadpis H1 má největší váhu a na stránce by se měl vyskytovat pouze jednou. Ostatní nadpisy se již mohou opakovat.

Zvýraznění textu na stránce

Pokud potřebujeme na stránce zvýraznit nějakou část textu tak, aby tento text byl zvýrazněn nejen vizuálně, ale také pro vyhledávače, musíme pro to použít speciální tagy. Nejčastější formou zvýraznění je text psaný tučně, nebo kurzívou. Obyčejné tagy a <i></i> jsou však nevhodné, protože jsou pro vyhledávač nerozeznatelné, ten pak není schopen poznat, jestli se jedná o text s větší vahou či nikoliv. Proto se používají náhradní tagy (zvýrazní text tučně) a (zvázrází text kurzívou).

1.3.2 „Off-page“ faktory

Na rozdíl od prvního případu, off-page faktory autor stránky přímo ovlivnit nemůže. Jsou to všechny faktory, které nejsou přímo na stránce, ale mají také vliv na výsledku ve vyhledávačích. Nejčastěji se pod těmito faktory vyskytují tzv. zpětné odkazy. To jsou takové odkazy, které z jiných míst na internetu odkazují na naši stránku. Existuje několik způsobů, jak si vybudovat velkou síť zpětných odkazů.

1.3.3 SEO a etika

Etika v SEO se týká hlavně způsobu, jakým budujeme zpětné odkazy na naši stránku.

V nejlepším možném případě z hlediska etiky, by zpětné odkazy měli vznikat na základě kvalitního obsahu stránky, který lidi natolik zaujme, že se rozhodnou na tuto stránku odkázat (například ze své stránky, blogu, fóra apod.). V dnešní době se množí využití sociálních sítí, jako je například Facebook, pomocí kterých je možné získat poměrně hodně zpětných odkazů, nebo zvýšit návštěvnost svých stránek. Jedná se o velice silný nástroj propagace stránek.

Pokud se zmíníme o neetickém získávání zpětných odkazů, měli bychom uvést dvě nejčastěji používané metody. První neetickou metodou je využití spamu. Tato metoda je založena na spamování různých fór, nebo návštěvních knih, kde se vkládá odkaz na naši stránku. Druhou metodou, kterou lze považovat za neetickou je využití skrytého textu na stránce. Tento text bývá na stránce umístěn za účelem zvýšení frekvence klíčových slov. Aby ovšem nepůsobil rušivě, bývá například pomocí CSS ukryt tak, aby jej viděl pouze vyhledávač.

2 Analýza problému a současné situace

2.1 Absence internetové prezentace

Vzhledem k tomu, že občanské sdružení Bá-Bi funguje poměrně čerstvě, ještě nemají vyhotovenou žádnou internetovou prezentaci. Oba zakladatelé sdružení proto získávají zákazníky (děti, které se chtějí účastnit jejich táborů a akcí) díky svým kontaktům z dřívějšího působení v jiné organizaci. Tento příliv kontaktů však není dostačující a je nutné v co možná nejmenším časovém intervalu zajistit reklamu v podobě internetových stránek. Proto mne jeden ze zakladatelů kontaktoval se žádostí o vytvoření internetové prezentace.

2.2 Požadavek na vytvoření prezentace

Požadavky ze strany sdružení se hodně vyvíjely a po dlouhém domlouvání jsme společně dospěli k představě, co všechno by stránky měli obsahovat.

- Aktuality z dění ve sdružení a pořádaných akcí
- Fotogalerie z akcí
- Možnost vložení videí z akcí na web
- Kniha návštěv

K těmto čtyřem základním požadavkům postupně přibývaly požadavky další:

- Seznam táborových lektorů i s jejich osobními kartami
- Kontaktní emailový formulář

Samozřejmostí bylo podle organizace i administrační rozhraní, které by zajistilo oběma zakladatelům možnost vlastní správy webu, bez nutnosti dalšího zásahu z mé strany.

Úkony, které po administračním rozhraní požadovali, byly následující:

- Správa aktualit (vytvoření / smazání)
- Správa videí (vytvoření / smazání)
- Správa fotogalerií (vytvoření / naplnění / smazání)
- Správa lektorů (vytvoření karty lektora / smazání karty lektora)

Se sdružením byla domlouvána spolupráce i v budoucnu, pro případ potřeby rozšíření webu o nějaké další funkce, ze které však posléze sešlo.

3 Vlastní návrhy řešení, přínos návrhů řešení

3.1 Návrh databáze

Prvním bodem, který jsem začal řešit při práci na tomto projektu, byla databáze. Vzhledem k potřebě administračního rozhraní a článků, které byly součástí požadavků, bylo potřeba navrhnout takovou databázi, která by pro tyto účely byla vyhovující. Nejdříve zde popíšu všechny tabulky, které budou v databázi použity, a u každé uvedu důvody vytvoření právě této tabulky.

Tabulka uživatelů

Název položky	Datový typ	délka	popis položky
id_uzivatel	int		primary key
login	varchar	20	login uživatele, musí být jednoznačný
heslo	varchar	20	heslo uživatele
prava	enum		admin,user
jmeno	varchar	30	jméno uživatele
prijmeni	varchar	30	příjmení uživatele
email	varchar	50	email uživatele

Tabulka 1: Tabulka uživatelů, Zdroj:vlastní

Tato tabulka byla zavedena hlavně kvůli administraci stránek. Obsahuje tedy údaje o každém uživateli. Položka „prava“ byla zavedena z důvodu, kdyby bylo v budoucnu potřeba poskytnout vlastní účet na webu i rodičům dětí, které se účastní táborů a akcí. Plány sdružení byly takové, že na každé akci či táboru, bude k dispozici odborný pracovník, který vypracuje posudek na každé dítě. Rodiče by pak měli po přihlášení na web, nalézt tyto posudky ve svých soukromých složkách.

Tabulka galerií

Název položky	Datový typ	délka	popis položky
id_galerie	int		primary key
nazev	varchar	50	název galerie
datum	datetime		datum akce
cesta	varchar	100	cesta k adresáři galerie
popis	text		popis galerie
thumb	varchar	100	cesta k náhledu galerie

Tabulka 2: Tabulka galerií, Zdroj:vlastní

Tabulka, obsahující důležité informace a popisy jednotlivých fotogalerií uložených na webu. Vzhledem k plánované rozsáhlosti galerie, je v tabulce uložena pouze cesta k adresáři, ve kterém jsou samotné fotografie uloženy. Tímto jsem dosáhl velikého odlehčení databáze, která by jinak musela pracovat s obrovskými kvanty dat a celé načítání galerií by se tak velice zpomalovalo. Obdobně je řešen i náhled fotogalerie.

Tabulka lektorů

Název položky	Datový typ	délka	popis položky
id_lektor	int		primary key
jmeno	varchar	30	jmeno lektora
prijmeni	varchar	30	prijmeni lektora
prezdivka	varchar	30	prezdivka lektora
vek	int		vek lektora
zaliby	text		konicky lektora
cesta	varchar	100	cesta k adresari lektora

Tabulka 3: Tabulka lektorů, Zdroj:vlastní

Tato tabulka obsahuje informace o každém lektorovi, který se účastní táborů a akcí. Základní informace, jako je jméno, příjmení či přezdívka, jsou doplněny možností vytvoření vlastní galerie každého lektora. Tato je opět z důvodu odlehčení databázi, řešena pomocí cesty k vlastnímu adresáři s fotografiemi.

Tabulka článků

Název položky	Datový typ	délka	popis položky
id_clanek	int		primary key
id_uzivatel	int		cizí klíč -> tabulka uzivatel
id_lektor	int		cizí klíč -> tabulka lektor
nadpis	varchar	100	nadpis aktuality
text	text		text aktuality
datum	datetime		datum vytvoření aktuality
thumb	varchar	100	cesta k náhledu článku
obrazek	varchar	100	cesta k obrázku článku

Tabulka 4: Tabulka článků, Zdroj:vlastní

Tabulka obsahující samotné články a aktuality vystavované na webu. Každý článek obsahuje kromě obvyklých záznamů také náhled a obrázek. Toto zajišťuje možnost mít každý článek s fotografií, která se jej přímo týká. Další zvláštností je položka

„id_lektor“, která byla zavedena pro případy, kdy bylo potřeba článek prezentovat pod jménem jednoho z lektorů.

Tabulka komentářů ke článkům

Název položky	Datový typ	délka	popis položky
id_komentar	int		primary key
id_clanek	int		cizí klíč -> tabulka News
komentar	text		komentář ke článku

Tabulka 5: Tabulka komentářů ke článkům, Zdroj:vlastní

Velice jednoduchá tabulka, sloužící pro uschování případných komentářů, které je možno psát pod každý článek.

Tabulka knihy návštěv

Název položky	Datový typ	délka	popis položky
id_zaznam	int		primary key
id_uzivatel	int		cizí klíč -> tabulka user
komentar	text		text příspěvku
datum	datetime		datum příspěvku
jmeno	varchar	30	
prijmeni	varchar	30	
prava	enum		admin,user,nereg

Tabulka 6: Tabulka knihy návštěv, Zdroj:vlastní

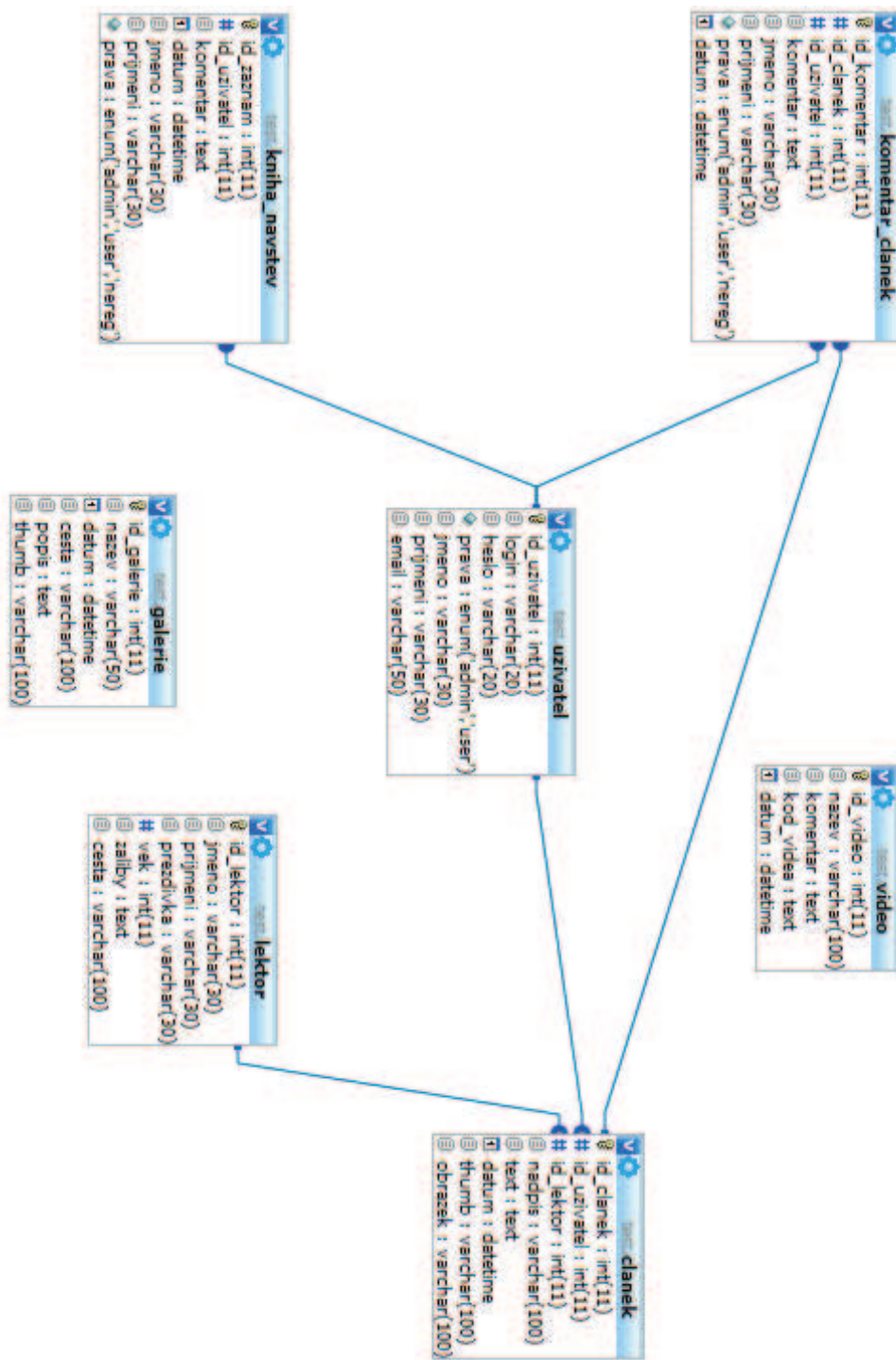
Tato tabulka, jak již název napovídá, slouží pro chod a funkčnost knihy návštěv. Udržují se v ní informace o autorovi příspěvku, datu vložení, apod.

Tabulka videí

Název položky	Datový typ	délka	popis položky
id_video	int		primary key
nazev	varchar	100	název videa
komentar	text		komentář/popis videa
kod_video	text		kód k vložení videa
datum	datetime		

Tabulka 7: Tabulka videí, Zdroj:vlastní

Poslední tabulka byla vytvořena na základě požadavku o vkládání videí na web. Samotný videosoubor však v databázi uložený není, je zde uložena pouze část adresy, kde je video uloženo.



Obrázek 1: Relační diagram databáze, Zdroj:vlastní

3.2 Volba vzhledu

3.2.1 Barevné schéma

Prvním krokem při volbě vzhledu stránky, je stanovení barevného schématu, kterého se při následné tvorbě designu budu držet. Zde jsem tuto práci měl ztíženou

faktem, že sdružení nemělo vytvořené logo – toto bylo pouze ve stádiu, kdy bylo zadáno grafikovi, aby jej vytvořil. Nicméně ke mně, jakožto tvůrci designu webu, se toto logo bohužel nedostalo. Se zakladateli sdružení jsem se proto snažil alespoň domluvit na požadovaných barvách, které by si na webu představovali. Bylo nám jasné, že vzhledem k působení sdružení v oblasti dětských táborů, bude muset být web laděn do veselých barev, které by nepůsobily na děti odpudivě. Cílem bylo vytvořit barevné schéma tak, aby na jednu stranu působilo příjemně a vesele, ale na stranu druhou nesmělo působit příliš přeplácane a chaoticky. Domluvili jsme se tedy na stránkách laděných do oranžové. Na pozadí byl použit přechod modré a bílé.

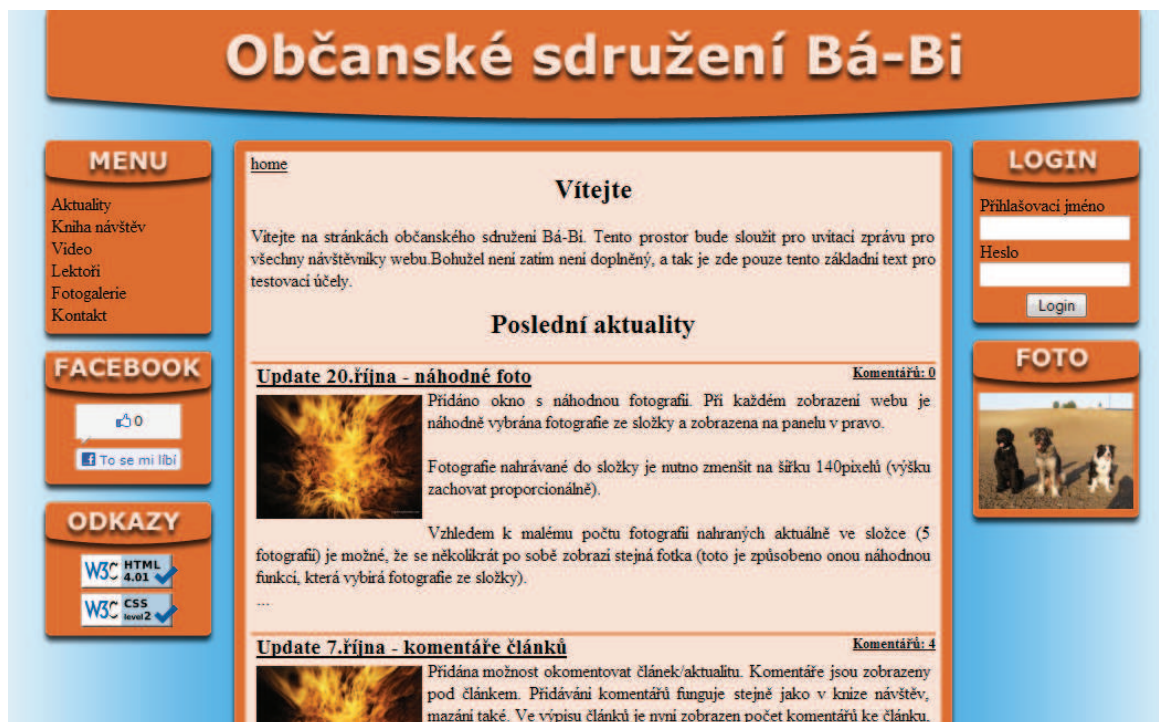
3.2.2 Layout stránky

Layout, neboli rozložení stránky, je dalším velice důležitým bodem při návrhu designu. Rozložení by mělo být uděláno tak, aby pohyb na stránkách a jejich ovládání bylo co nejvíce intuitivní. Jako základní rozvržení jsem proto zvolil velice známé a hojně používané schéma o třech sloupcích. Hlavičku webu tvoří logo sdružení, které bylo prozatím nahrazeno jednoduchým nápisem. Ten měl být později nahrazen již vyhotoveným logem sdružení. Samotné tělo webu je pak rozděleno na ony tři sloupce. Do levého sloupce jsem se rozhodl umístit hlavní menu, jelikož je toto místo pro menu velice obvyklé a pro mnohé uživatele již zažitě. Neměl by být tedy problém s jeho najitím a následnou orientací na webu. Pravý sloupec je použit pro přihlašovací formulář, kterým se administrátor webu může přihlásit do administrační sekce. Po úspěšném přihlášení je tento formulář nahrazen administračním menu, které obsahuje veškeré požadované úkony. Ve střední části těla webu je již umístěn samotný obsah. Celý design je ukončen patičkou, na kterou bylo v plánu napsat základní kontaktní informace na sdružení. Celá tvorba designu prošla několika návrhy, které jsem postupně předkládal k hodnocení zadavatelům projektu. Konečný design, který mi byl schválen, vypadal takto.



Obrázek 2: Design webu, Zdroj:vlastní

Tento design byl posléze ještě nepatrně upraven, když byl pod panel „Admin menu“ umístěn nový panel, na kterém se měli zobrazovat náhodné fotografie z akcí. Po této úpravě již byl celý vzhled rozřezán a postupně aplikován přímo na stránku. Pro představu, jak celý výsledek vypadá, zde uvedu další obrázek, který zachycuje podobu stránky těsně před plánovaným vypuštěním. Použité fotografie u článků a náhodného fota jsou pouze testovací, s činnostmi sdružení nemají nic společného.



Obrázek 3: Finální podoba webu, Zdroj: vlastní

3.3 Struktura stránky

V následující kapitole si projdeme jednotlivé části stránky a vysvětlíme si, jak fungují. U každé části si také popíšeme postup při její správě.

3.3.1 Aktuality

Aktuality, nebo také články, jsou hlavní informační součástí celého webu. Na úvodní stránce jsou vždy zobrazeny dva nejnovější články, které je možno, pomocí odkazu v nadpisu článku, zobrazit v celém znění. Toto můžeme vidět na obrázku výše, kde je vyfocena úvodní stránka. Kompletní archiv aktualit je k nalezení pod položkou „Aktuality“ v hlavním menu. Zde jsou vypsány všechny články od nejnovějšího po nejstarší a z důvodu přehlednosti je zobrazeno vždy jen pět článků na stránku. Listování v aktualitách je umožněno pomocí ovládacích prvků umístěných nad i pod výpisem článků. V rámci oživení celé sekce aktualit byla později přidána i možnost na každý článek reagovat komentářem. Toto se provádí pomocí formuláře umístěného pod každým článkem. Počty komentářů jsou pak zobrazovány i u náhledu článků.

Občanské sdružení Bá-Bi

MENU

- Aktuality
- Kniha návštěv
- Video
- Lektoři
- Fotogalerie
- Kontakt

FACEBOOK

0

To se mi líbí

ODKAZY

W3C HTML 4.01

W3C CSS level 2

home - aktuality

Aktuality

< << strana 1/3 >> >

Update 20.října - náhodné foto Komentářů: 0



Přidáno okno s náhodnou fotografií. Při každém zobrazení webu je náhodně vybrána fotografie ze složky a zobrazena na panelu v pravo.

Fotografie nahrávané do složky je nutno zmenšit na šířku 140pixelů (výšku zachovat proporcionálně).

Vzhledem k malému počtu fotografií nahraných aktuálně ve složce (5 fotografií) je možné, že se několikrát po sobě zobrazí stejná fotka (toto je způsobeno onou náhodnou funkcí, která vybírá fotografie ze složky).

...

Update 7.října - komentáře článků Komentářů: 4



Přidána možnost okomentovat článek/aktualitu. Komentáře jsou zobrazeny pod článkem. Přidávání komentářů funguje stejně jako v knize návštěv, mazání také. Ve výpisu článků je nyní zobrazen počet komentářů ke článku, který zároveň slouží jako odkaz směřující přímo na komentáře daného článku.

Prosím o vyzkoušení, jestli vše funguje správně. Já jsem na žádnou chyбку nenarazil, ale je možné, že jsem přehlédnul...

LOGIN

Přihlašovací jméno

Heslo

Login

FOTO



Obrázek 4: Výpis aktualit, Zdroj: vlastní

Komentáře ke článku

Jméno

Příjmení

Text příspěvku

Vypočti příklad: 1 + 0 =

Vlož příspěvek

< << strana 1/1 >> >

Martin Bárta - Administrátor	10.10.2010 v 21:18
Myslím, že vše funguje tak jak má.	
Martin Bárta	10.10.2010 v 21:17
Myslím, že vše funguje tak jak má.	
Zdeněk Vičar	7.10.2010 v 11:54
zkušební komentář ke článku - neregistrovaný uživatel	

Obrázek 5: Formulář pro vložení komentářů, Zdroj: vlastní

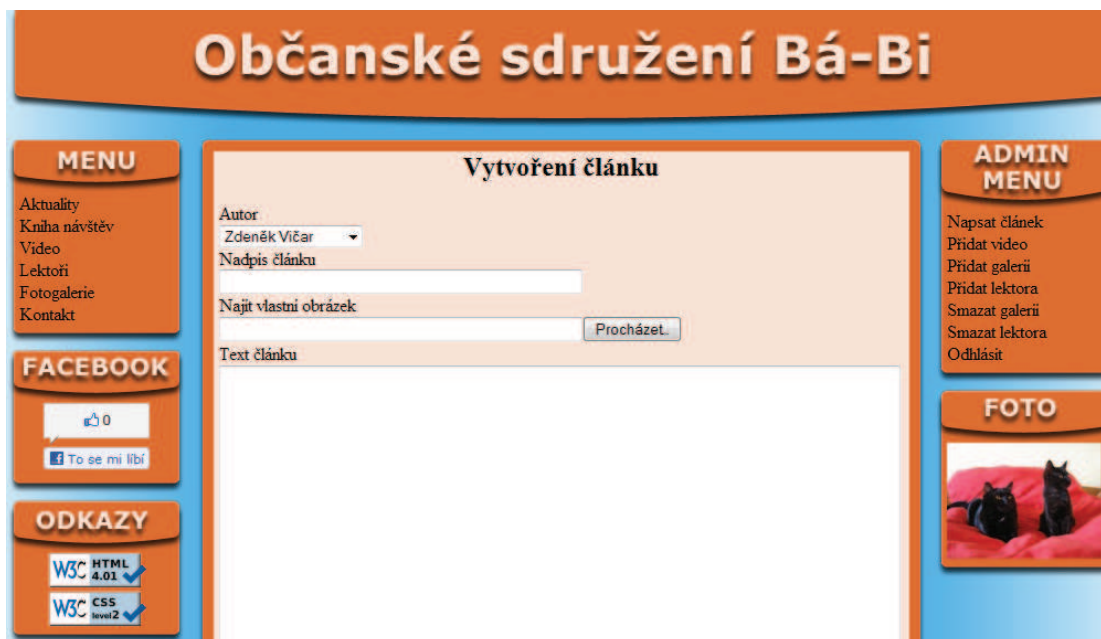
Nyní si řekneme, jak je zobrazování článků uděláno. Všechny články jsou uloženy, spolu se všemi potřebnými údaji, v databázi, ze které si je dle potřeby vytahují.

Zatímco na úvodní stránce jsem pomocí SQL dotazu vypsal pouze dva nejnovější články, v archivu aktualit je SQL dotaz mírně poupraven a vypisuje se pět článků na stránku. Ještě před samotným výpisem článků je proto nutné spočítat počet stran, a podle výsledku upravit navigační prvky. Aktuální strana je zjišťována pomocí parametru obsaženého v adrese stránky. Obdobně jsou řešeny i komentáře pod články. V rámci ošetření komentářů proti spam-botům, jsem použil ověřovací příklad, bez jehož správného vyřešení nelze příspěvek odeslat. Toto ověření ji není vyžadováno pro administrátory přihlášené na stránky, jelikož ti již svoji totožnost prokázali samotným přihlášením.

3.3.1.1 Správa aktualit

Správa aktualit je umožněna po nalogování pod administrátorským účtem z „Admin menu“ umístěného v pravé části webu. Zde najdeme položku „Napsat článek“. Zobrazí se nám formulář, kde můžeme zadat nadpis článku, text článku a připojit vlastní obrázek. Pokud žádný obrázek nepřipojíme, použije se obrázek přednastavený. V případě, že chceme použít obrázek vlastní, jej pomocí tlačítka „procházet“ najdeme ve vlastním počítači a po odeslání článku se tento obrázek automaticky uloží do složky. Zároveň dojde k vytvoření miniaturní obrázku pro účely zobrazení náhledu článku. Oba tyto soubory se uloží pod identifikátorem článku, aby bylo možné je posléze propojit se samotným článkem.

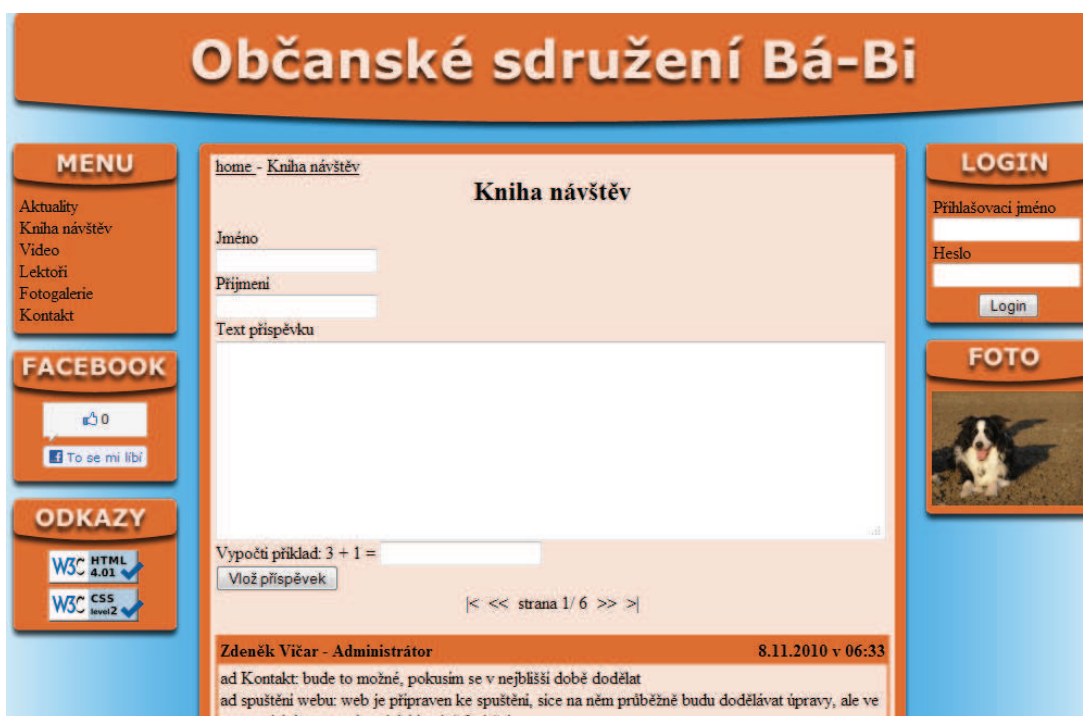
Položka autor, kterou můžeme najít ve formuláři, obsahuje automaticky jméno přihlášeného administrátora. Pouze pro případ, kdy by bylo nutné vložit na web článek, který napsal například nějaký lektor, zde bylo umožněno přiřadit ke článku jiného autora, než přihlášeného administrátora. Tato funkce byla zavedena přímo na žádost sdružení, které nechtělo lektorům vytvářet zvlášť účet, který by měl práva na přidávání článků na web. Případné smazání článku se provádí přímo ve výpisu aktualit, kdy se po úspěšném přihlášení jako administrátor, zobrazí odkaz „smazat“. Po kliknutí na tento odkaz dojde ještě k potvrzení, zobrazeném nad výpisem článků, a po potvrzení je již článek pomocí SQL dotazu vymazán z databáze. Smazán bude nejen záznam v databázi, ale také případné obrázky přiřazené ke článku.



Obrázek 6: Správa článků, Zdroj: vlastní

3.3.2 Kniha návštěv

Kniha návštěv by měla sloužit jako místo, kde případný návštěvník webu může zanechat svoje dojmy jak z webu samotného, tak třeba svoje zkušenosti se sdružením. Zaslát příspěvek je tedy povoleno i neregistrovanému uživateli, nicméně je tento příspěvek viditelně odlišený od příspěvku administrátora. Pokud bude reagovat v knize návštěv administrátor, řádně přihlášený přes formulář na pravé straně webu, bude jeho příspěvek odlišen barevným záhlavím, tučně psaným jménem a doplňujícím slovem „Administrátor“.



Obrázek 7: Kniha návštěv, Zdroj: vlastní

Samotná funkčnost knihy návštěv je opět řešena pomocí databáze. Kdy je každý článek uložen v databázi, ze které je pak pomocí SQL dotazu vypisováno vždy pět příspěvků na stránku. Samozřejmostí jsou ovládací prvky, které fungují naprosto totožně jako prvky u aktualit. Při odesílání příspěvku je opět zaveden kontrolní příklad.

3.3.3 Video

Podstránka, která zobrazuje videa uložená pomocí administrační sekce (ukládání videí bude vysvětleno dále). Videa zobrazována na této stránce jsou uložena na známém portálu Youtube.com, kde budou i do budoucna všechny záznamy z akcí ukládány. Přestože samotný server Youtube.com nabízí poměrně jednoduchou možnost, jak si na stránky můžeme vložit video uložené právě u nich, bohužel však ani jedna nabízená varianta nebyla validní. A tak jsem musel celý vkládací kód upravit. Dosáhl jsem výsledku, kdy je kód naprosto validní a zároveň umožňuje bezproblémové pouštění videí i s možností přepnutí do celoobrazovkového režimu (fullscreen).

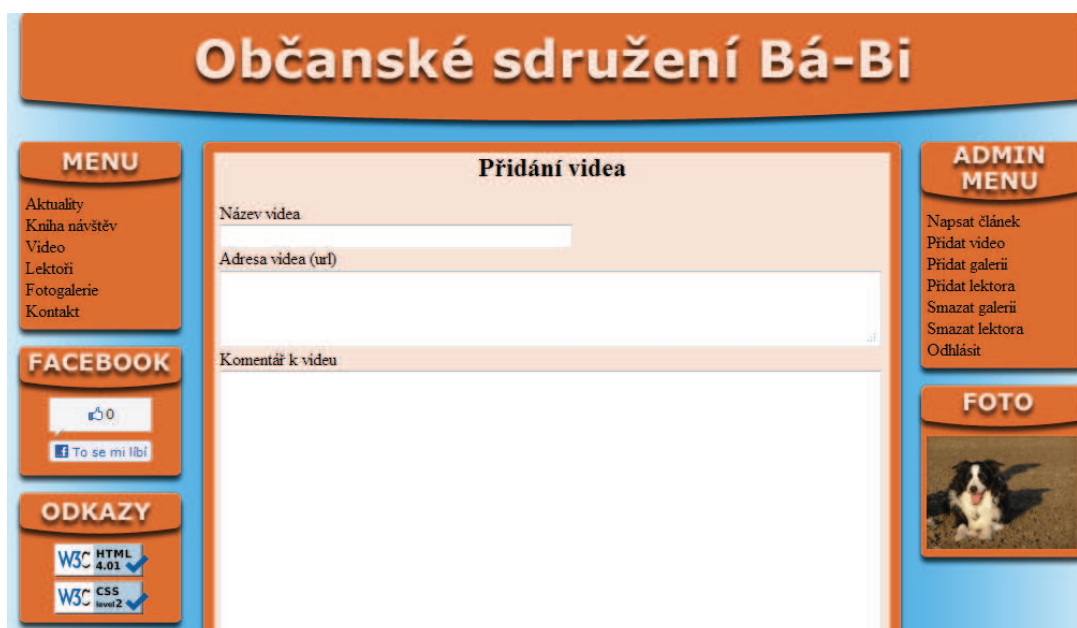


Obrázek 8: Videa, Zdroj: vlastní

Videa jsou na stránku opět vypisována z databáze, která ovšem neobsahuje celé video, ale pouze jeho identifikátor. Ten jsem získal po prozkoumání URL adresy, kdy jsem si všiml, že identifikátor pro každé video je předáván pomocí parametru právě v adrese odkazu.

3.3.3.1 Správa videí

Přidání videa na stránku je přístupné opět z admin menu, pod položkou „Přidat video“. Po kliknutí na tuto položku se objeví formulář, kde vyplníme název videa, URL adresu videa a nějaký komentář. Vzhledem k faktu, že spravovat stránku by měl i normální uživatel, zvolil jsem možnost, kdy se do adresy videa může vložit celá URL adresa zkopírovaná z prohlížeče. Samotný identifikátor videa se již pak automaticky z adresy vyřízne a uloží do databáze. Smazání videí je provedeno obdobně jako u článků. Po nalogování jako administrátor se tato možnost objeví přímo u výpisu videí. Kde bude u každého videa opět zobrazena možnost „smazat“.



Obrázek 9: Přidání videa, Zdroj: vlastní

3.3.4 Fotogalerie

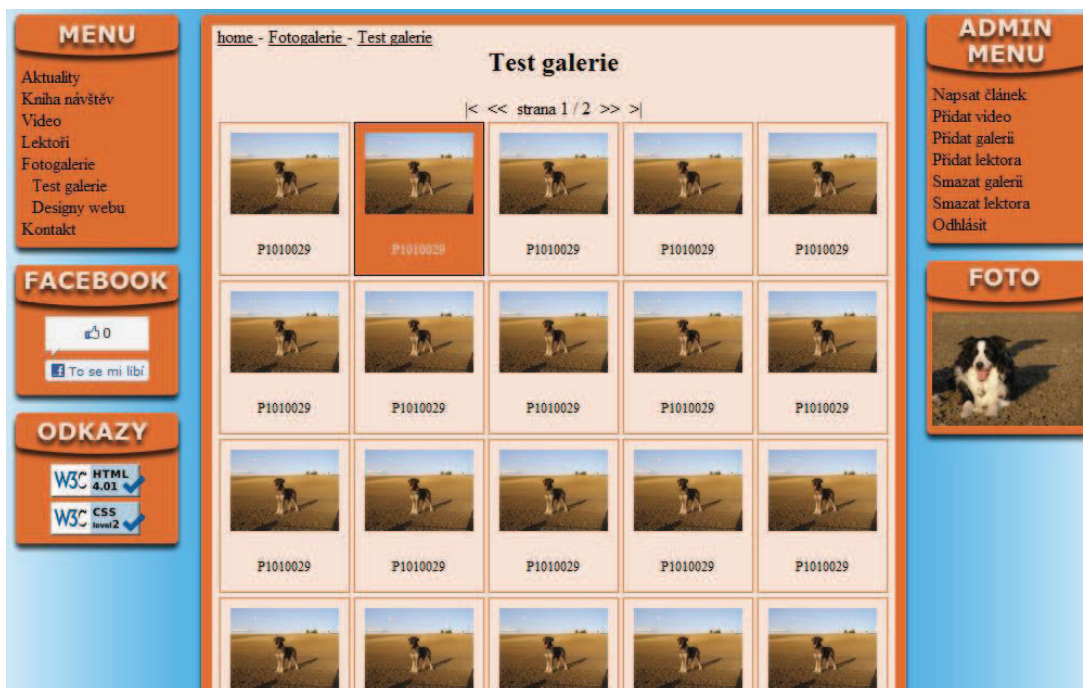
Z prvopočátku chtělo sdružení jako fotogalerii použít nějakou externí internetovou galerii, na kterou by byl na webu sdružení umístěn pouze odkaz. Z tohoto řešení nakonec sešlo a byla mi zadána i tvorba vlastní fotogalerie, pro inspiraci mi dodali pár adres na stránky, u kterých se jim fotogalerie zamlouvala. Bohužel všechny uvedené stránky používali fotogalerii vytvořenou pomocí Javascriptu, volně přístupnou na internetu pod názvem Lightbox 2. Toto řešení však bylo pro moje účely nepřijatelné, tak jsem byl nucen galerii vytvořit sám, pouze s přihlédnutím k designu již zmiňovaného Lightboxu. Moje fotogalerie je ovšem kompletně napsaná v PHP jazyce, tudíž nemůže dojít k situaci, kdy si uživatel ručně vypne Javascript a nebude mu pak fungovat fotogalerie tak, jak by měla. Podařilo se mi dosáhnout stejného vizuálního efektu i bez použití Javascriptu.



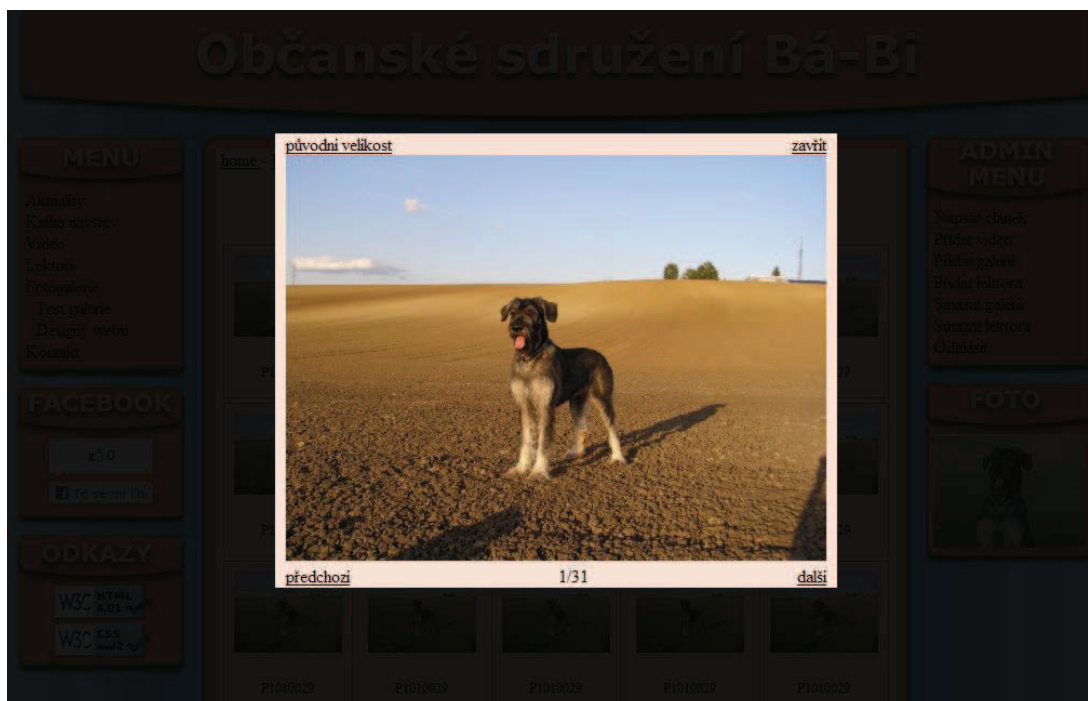
Obrázek 10: Výpis fotogalerií, Zdroj: vlastní

Nyní k samotnému řešení fotogalerie. Po kliknutí na odkaz „fotogalerie“ se dostaneme na stránku, kde jsou vypsané všechny dostupné fotogalerie i s náhledy a základními informacemi. Tyto informace obsahují počet fotografií v galerii, datum vytvoření a krátký popis galerie. Jednotlivé informační karty galerií jsou vypisovány opět z databáze. Zároveň je v této fázi v hlavním menu rozbaleno „podmenu“ fotogalerie, ve kterém jsou všechny galerie taktéž vypsané. Po vybrání příslušné galerie se nám již zobrazí výpis dvaceti miniatur fotografií. Samotné miniatury a velké verze fotografií ovšem již nejsou uloženy v databázi, ale v adresářích na serveru, kde je celý web uložen. Načítání miniatur tudíž není prováděno z databáze, ale přímo ze složky, kde jsou uloženy. Jediné, co se zde vytahuje z databáze je název galerie a cesta k adresáři s fotkami. Název galerie je poté předáván jako parametr v adrese stránky. Před samotným načtením a zobrazením všech miniatur dochází k otestování, jestli vůbec každá fotografie má vytvořenou miniaturu k náhledu. Pokud ano, miniatura se pouze zobrazí, ale pokud ještě vytvořená není, dojde automaticky k vytvoření miniatury a jejímu následnému uložení do patřičné složky. Díky tomuto faktu se může stát, že první zobrazení fotogalerie může trvat déle. Jakmile se ovšem miniatury jednou vytvoří, při každém dalším zobrazení je vše již velice rychlé. Po kliknutí na příslušnou miniaturu se nám ona fotografie zobrazí uprostřed celé stránky, kdy je zbytek stránky ztmaven,

aby nedocházelo k rušení při prohlížení fotografií. V tomto režimu pak můžeme v galerii listovat a postupně si ji celou prohlédnout. V případě, že bychom potřebovali zobrazit plnou velikost fotografie, je toto umožněno pomocí odkazu vlevo nahoře „původní velikost“.



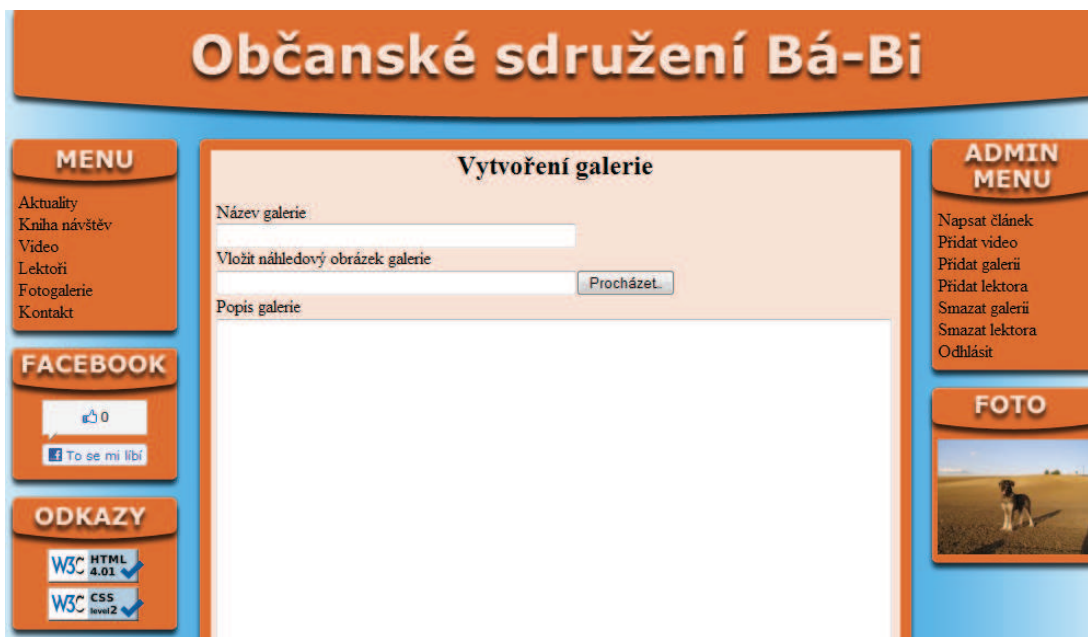
Obrázek 11: Výpis miniatur fotografií, Zdroj: vlastní



Obrázek 12: Zobrazení fotografie, Zdroj: vlastní

3.3.4.1 Správa fotogalerií

Přímo z administrátorského rozhraní stránky je možné pouze vytvořit nebo smazat galerii. Vytvoření galerie je prováděno opět přes formulář, kde uvedeme název a popis galerie a přiložíme náhledový obrázek. S odesláním tohoto formuláře se uloží příslušné záznamy do databáze a dojde k vytvoření složky pro novou galerii. Do této složky se také uloží vkládaný náhledový obrázek. Smazání galerie je řešeno obdobně jako mazání v předešlých případech, pouze s tou odlišností, že zde je pro tyto účely přímo položka v admin menu.



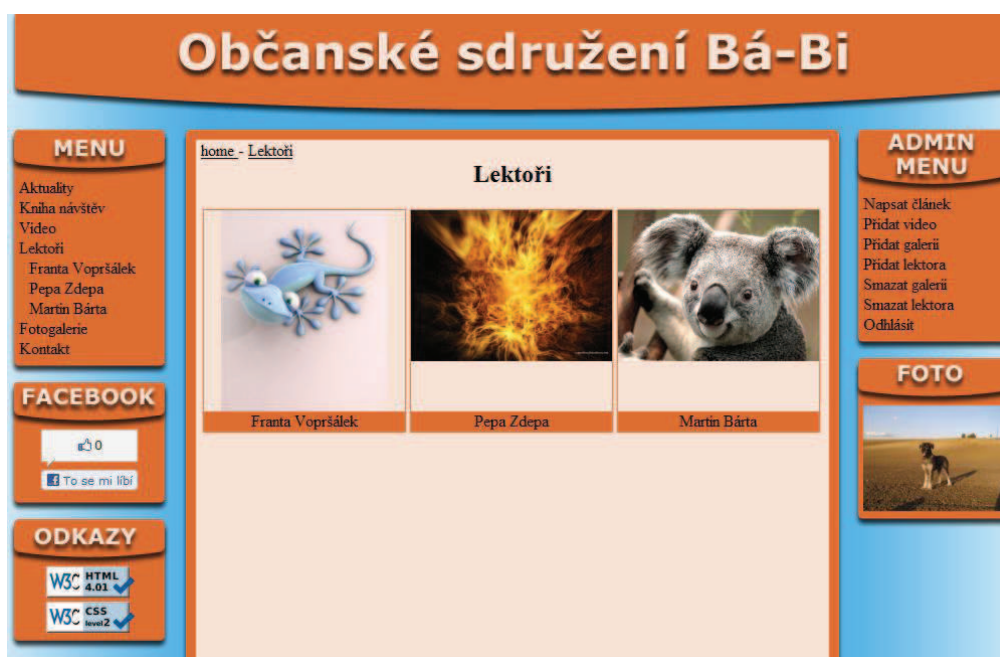
Obrázek 13: Přidání fotogalerie, Zdroj: vlastní

Správa samotných fotek, které v galerii budou zobrazeny, je řešena pomocí FTP. Kdy přes FTP nahrajeme do příslušné složky (která je pojmenována stejně jako název galerie) fotky. Miniatury fotek již nemusíme nijak řešit, k jejich vytvoření dojde automaticky při prvním zobrazení fotogalerie na webu.

3.3.5 Lektoři

Na žádost sdružení byla také na web přidána sekce Lektoři. V této sekci jsou uloženy osobní karty všech lektorů, kteří vedou tábory. Po kliknutí na odkaz v menu, se nám zobrazí náhledy všech lektorů, které obsahují fotografii a jméno lektora. Seznam lektorů se zobrazí také v hlavním menu. Po kliknutí na příslušného lektora se dostaneme již na osobní kartu, kde můžeme najít základní informace o lektorovi, jsou zde uvedeny také odkazy na všechny články, které lektor napsal. Později byla také přidána každému

lektorovi osobní fotogalerie. Ta je řešena totožně jako klasická fotogalerie na webu, vložené fotografie jsou tedy uloženy v adresářích pojmenovaných jako jméno lektorů.



Obrázek 14: Výpis lektorů, Zdroj: vlastní



Obrázek 15: Karta lektora, Zdroj: vlastní

3.3.5.1 Správa lektorů

Správa lektorů probíhá opět z části přes webové rozhraní, kde je možné založit a smazat kartu lektora. Pro založení je nutné zadat základní informace o lektorovi a vložit jeho profilovou fotografii. Záznamy jsou poté uloženy do databáze a fotografie do

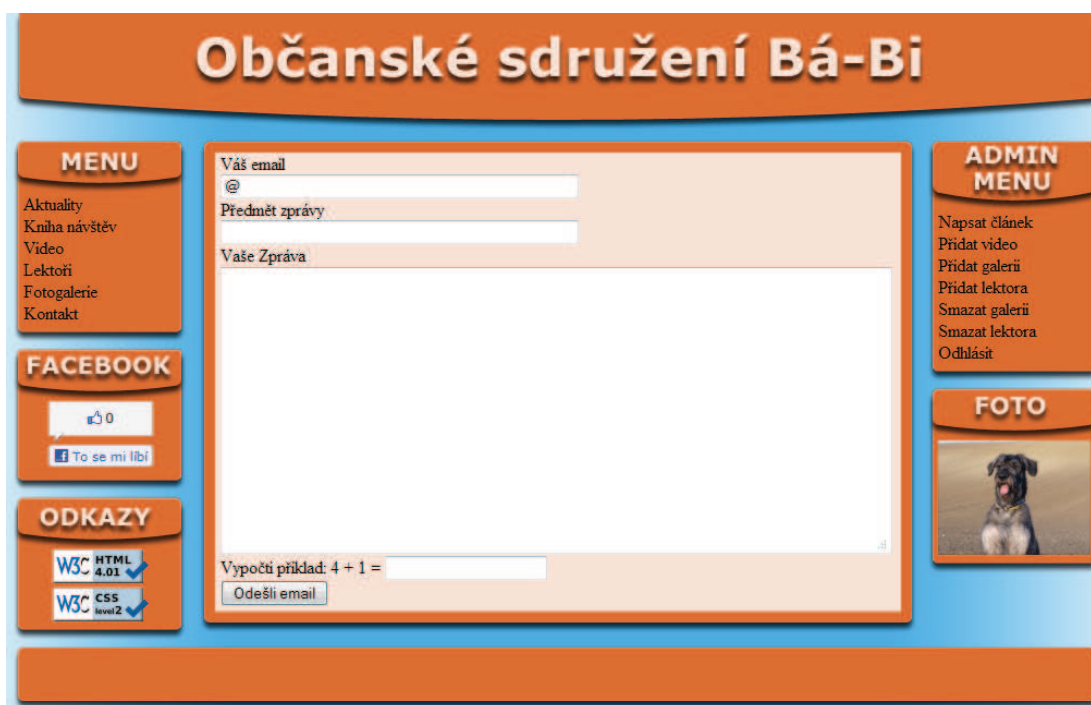
lektorovy složky, která je automaticky vytvořena. Po smazání lektora, které je přístupné z admin menu, dojde ke smazání záznamu v databázi a celé lektorovy složky i s fotografiemi.

The screenshot displays the 'Občanské sdružení Bá-Bi' website interface. The main content area is titled 'Vytvoření lektora' (Create Lecturer). It features a form with the following fields: 'Jméno' (Name), 'Příjmení' (Surname), 'Přezdívká' (Nickname), 'Věk' (Age), 'Vložit profilové foto' (Upload profile photo) with a 'Procházet...' (Browse...) button, and 'Záliby' (Hobbies). A 'Vytvoř lektora' (Create lecturer) button is located at the bottom of the form. The left sidebar contains a 'MENU' with links to 'Aktuality', 'Kniha návštěv', 'Video', 'Lektoři', 'Fotogalerie', and 'Kontakt'. Below the menu is a 'FACEBOOK' widget showing 0 likes and a 'To se mi líbí' button. Further down is an 'ODKAZY' section with icons for 'HTML 4.01' and 'CSS level2'. The right sidebar has an 'ADMIN MENU' with options: 'Napsat článek', 'Přidat video', 'Přidat galerii', 'Přidat lektora', 'Smazat galerii', 'Smazat lektora', and 'Odhlásit'. Below the admin menu is a 'FOTO' section displaying a small image of a dog.

Obrázek 16: Vytvoření lektora, Zdroj: vlastní

3.3.6 Kontakt

Sekce kontaktu byla přidána na web jako poslední, kdy sdružení požadovalo možnost odeslat email přímo z rozhraní stránek. Odesílání emailu je prováděno pomocí funkce `Mail()`, kterou je ovšem nutné nejdříve povolit na příslušném webovém hostingu. Problémy s nepovolením této funkce jsou ovšem v drtivé většině případů pouze na free hostingu, placené varianty tuto funkci povolenou mají a kontaktní formulář zde funguje naprosto bezproblémově. Do formuláře pak stačí zadat váš email, který se příjemci zobrazí v poli odesílatele pro případnou odpověď. Dále stačí již jen zadat předmět a obsah zprávy, kterou chcete zaslat.



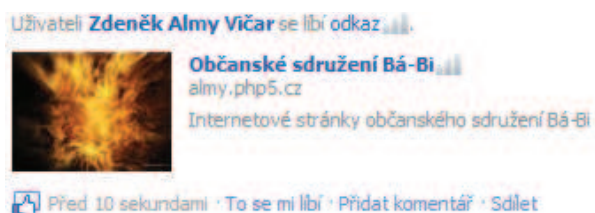
Obrázek 17: Kontaktní formulář, Zdroj: vlastní

3.4 Propojení se sociální sítí Facebook

Vzhledem k rostoucí oblibě sociálních sítí jsem se rozhodl, že celý web propojím s nejrozšířenější sociální sítí u nás, s Facebookem. K tomuto propojení jsem prozatím použil pouze jeden modul nabízený přímo autory Facebooku, který je umístěn pod hlavním menu. Pomocí tohoto modulu se lidé, kteří mají na Facebooku účet, mohou přidat tuto stránku mezi věci, které se jim líbí. Tímto dojde k výraznému rozšíření povědomí o tomto webu a případnému nárůstu zákazníků a zájemců o tábory.



Obrázek 18: Facebook panel, Zdroj: vlastní



Obrázek 19: Zobrazení na Facebooku, Zdroj: vlastní

Se sdružením jsem domlouval i založení profilu na Facebooku, aby bylo možné propojení s touto sítí dokončit. V takovém případě by pak každý článek psaný na webu byl automaticky zobrazen i na Facebooku a domnívám se, že by to velice prospělo nejen návštěvnosti stránek. Vzhledem k přerušení kontaktu se sdružením ovšem k dokončení této fáze nedošlo.

3.5 Ekonomické zhodnocení

Náklady na zavedení internetových stránek můžeme rozdělit do více oblastí. Nejdražší částí bývá zpravidla samotná tvorba stránek, které by splňovali podmínku administrátorské sekce, pomocí které bude docházet k aktualizaci a správě stránek. Dalším druhem nákladů je pořízení internetové domény a webhostingu, kde budou stránky nahrány. V poslední řadě pak přichází náklady na případnou pozdější aktualizaci stránek. Vzhledem k faktu, že celé internetové stránky byly pro sdružení vyhotoveny v rámci bakalářské práce, náklady na tvorbu stránek byly tedy nulové. Stránky obsahují administrační rozhraní, které umožňuje bezproblémovou správu stránek bez nutnosti zásahu tvůrce, tudíž i náklady na aktualizaci stránek budou nulové. Webhostingový program byl zvolen u firmy Webhosting C4, přičemž v nabídce je v ceně i doména. Náklady na tento webhosting budou ve výši 1200,- Kč za jeden rok, což jsou ve skutečnosti jediné náklady, které při zavedení mého návrhu bude sdružení nuceno uhradit.

Co se týče přínosů navrhovaného řešení, zde je jeden velice nezpochybnitelný přínos v podobě internetové reklamy, která doposud sdružení chyběla. Internetové stránky, které budou propojeny se sociální sítí Facebook, mohou velice zvýšit povědomí o tomto občanském sdružení a o činnosti, ve které se pohybuje. Toto pravděpodobně povede ke zvýšení poptávky po táborech pořádaných právě touto organizací. Díky již zmiňovanému propojení s Facebookem, přítomnosti komentářů a knihy návštěv, může internetová stránka fungovat také jako portál, kde se budou scházet účastníci již proběhlých akcí, což může vést k opakované účasti na jiných akcích. Samozřejmě bude do velké míry záležet na kvalitě táborů a akcí, které bude sdružení pořádat, nicméně internetové stránky mohou být velice nápomocny s hledáním nových zájemců a jejich následném udržení.

4 Závěr

Cílem celé mé práce byl návrh a následné vytvoření nových internetových stránek pro občanské sdružení Bá-Bi. Můj záměr byl vytvořit takové internetové stránky, které by fungovaly jako internetová reklama pro sdružení, ale i jako informační portál, který by všem návštěvníkům poskytl všechny informace o sdružení, dokonce i fotografie a videa z pořádaných akcí.

Prvním bodem byl návrh databáze, která by sloužila jako zdroj dat pro celé stránky. Databáze byla tedy navržena tak, aby tento požadavek splnila, a zároveň jsem při jejím navrhování bral ohled na budoucí požadavky sdružení, jako například zavedení uživatelských účtů na stránce.

Dalším bodem byl návrh designu webu, kde jsem se brát ohled na cílovou skupinu návštěvníků a obor působení sdružení. Proto byly zvoleny veselé barvy, které mají vytvářet příjemné místo, kam se budou návštěvníci rádi vracet.

Posledním bodem byl návrh samotných stránek s důrazem na intuitivnost a jednoduché ovládání. Proto bylo zvoleno rozvržení stránek takové, aby všechny ovládací prvky byly snadno dostupné a dobře viditelné.

Webové stránky byly následně otestovány v nejrozšířenějších prohlížečích (Firefox, Internet Explorer, Opera, Google Chrome), přičemž byl brán ohled i na staré verze těchto prohlížečů. V rámci zachování stejného vzhledu ve všech těchto prohlížečích sem byl nucen provést úpravy stylových předpisů, kdy sem pomocí podmíněných stylů přidal výjimky zejména pro Internet Explorer, který je velice problematický. Výsledný efekt je ale výborný a celý web je plně funkční i na nejproblémovější verzi 6, která je bohužel stále velice rozšířená.

Vzhledem k ukončení spolupráce s občanským sdružením Bá-Bi, které přestalo mít o tyto stránky zájem, nebyly stránky oficiálně spuštěny a k dispozici je pouze testovací verze umístěná na free hostingu.

5 Použité zdroje

5.1 Knižní zdroje

- [1] BRÁZA, Jiří. *PHP 5 : začínáme programovat*. Praha : Grada Publishing, 2005. 244 s. ISBN 80-247-1146-X.
- [2] CEDERHOLM, Dan. *Flexibilní webdesign*. 1.vyd. Brno: Computer Press, a.s., 2006. 232 s. ISBN: 80-251-1018-4.
- [3] DOMES, Martin. *Tvorba webových stránek*. 1.vyd. Brno: Computer Press, a. s., 2006. 192 s. ISBN: 80-251-0920-8.
- [4] GUTMANS, Andi. *Mistrovství v PHP 5* . 1.vyd. Brno: Computer Press, a.s.,2005. 520 s. ISBN 80-251-0799-X.
- [5] CHROMÝ, Jan. *Tvorba webových stránek*. Praha : Vysoká škola hotelová v Praze 8, katedra marketingu, 2005. 50 s. ISBN 80-86578-22-4.
- [6] KOSEK, Jiří. *HTML : tvorba dokonalých WWW stránek*. Praha : Grada Publishing, 1998. 296 s. ISBN 80-7169-608-0.
- [7] KRUG, Steve. *Webdesign - Nenuťte uživatele přemýšlet*. 2.vyd. Brno: Computer Press, a.s., 2006. 168 s. ISBN: 80-7226-892-9.
- [8] POLLOCK, John. *Javascript : a begginer*. [s.l.] : McGraw-Hill Osborne Media, 2009. 603 s. ISBN 978-0-07-163295-9.
- [9] PONKRÁČ, Miloslav. *PHP a MySQL : bez předchozích znalostí*. Brno : Computer Press, 2007. 221 s. ISBN 978-80-251-1758-3.
- [10] PROKOP, Marek. *CSS kaskádové styly pro webdesignery*. Brno : CP Books, 2005. 288 s. ISBN 80-251-0487-7.
- [11] SMIČKA, Radim. *Optimalizace pro vyhledávače - SEO : Jak zvýšit návštěvnost webu*. Dubany : Jaroslava Smičková, 2004. 120 s. ISBN 80-239-2961-5.

5.2 Elektronické zdroje

- [12] HASSMAN, Martin. *HTML 4 5 6 ...* [online]. 16. srpna 2007 [cit. 2010-12-12]. Jak je to s termínem dokončení HTML 5?. Dostupné z WWW: <<http://html456.blogspot.com/2007/08/jak-je-to-s-terminem-dokonceni-html5.html>>.

- [13] JANOVSKEÝ, Duřan. *Jak psát web* [online]. 2011 [cit. 2011-04-27]. Syntaxe HTML. Dostupné z WWW: <<http://www.jakpsatweb.cz/html/syntaxe.html>>.
- [14] JANOVSKEÝ, Duřan. *Jak psát web* [online]. 2011 [cit. 2011-05-03]. Syntaxe XHTML. Dostupné z WWW: <<http://www.jakpsatweb.cz/html/xhtmll.html>>.
- [15] *Seznam.cz* [online]. 2011 [cit. 2011-05-14]. On-page faktory. Dostupné z WWW: <http://napoveda.seznam.cz/cz/fulltext-hledani-v-internetu/optimalizace-webu/on-page-faktory/#titulek_stranky>.
- [16] *W3C* [online]. 19. říjen 2010 [cit. 2010-12-13]. HTML 5. Dostupné z WWW: <<http://www.w3.org/TR/html5/>>.
- [17] *W3C* [online]. 1997 [cit. 2011-04-25]. HTML 4.01 Specification. Dostupné z WWW: <<http://www.w3.org/TR/html401/>>.
- [18] *W3C* [online]. 2. července 2009 [cit. 2010-12-13]. W3C News Archive: 2009. Dostupné z WWW: <<http://www.w3.org/News/2009#entry-6601>>.
- [19] *W3C* [online]. 2007 [cit. 2010-12-12]. W3C HTML Working Group. Dostupné z WWW: <<http://www.w3.org/html/wg/>>.
- [20] *W3C* [online]. 26. července 2006 [cit. 2010-12-13]. XHTML 2.0. Dostupné z WWW: <<http://www.w3.org/TR/xhtmll2/>>.
- [21] *W3C* [online]. 26. ledna 2000 [cit. 2010-12-13]. XHTML 1.0 The Extensible HyperText Markup Language (Second Edition). Dostupné z WWW: <<http://www.w3.org/TR/xhtmll1/>>.

6 Seznamy

6.1 Seznam obrázků

Obrázek 1: Relační diagram databáze	42
Obrázek 2: Design webu.....	44
Obrázek 3: Finální podoba webu	45
Obrázek 4: Výpis aktualit	46
Obrázek 5: Formulář pro vložení komentářů.....	46
Obrázek 6: Správa článků	48
Obrázek 7: Kniha návštěv	49
Obrázek 8: Videá	50
Obrázek 9: Přidání videa.....	51
Obrázek 10: Výpis fotogalerí.....	52
Obrázek 11: Výpis miniatur fotografií.....	53
Obrázek 12: Zobrazení fotografie.....	53
Obrázek 13: Přidání fotogalerie	54
Obrázek 14: Výpis lektorů.....	55
Obrázek 15: Karta lektora.....	55
Obrázek 16: Vytvoření lektora	56
Obrázek 17: Kontaktní formulář.....	57
Obrázek 18: Facebook panel.....	57
Obrázek 19: Zobrazení na Facebooku	57

6.2 Seznam tabulek

Tabulka 1: Tabulka uživatelů	39
Tabulka 2: Tabulka galerií	39
Tabulka 3: Tabulka lektorů.....	40
Tabulka 4: Tabulka článků	40
Tabulka 5: Tabulka komentářů ke článkům	41
Tabulka 6: Tabulka knihy návštěv	41
Tabulka 7: Tabulka videí	41