

Webová prezentace skautského oddílu

Jan Šlahůnek

2006



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2005/2006

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jan ŠLAHŮNEK**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Webová prezentace skautské organizace**

Zásady pro vypracování:

1. Seznámení se s interaktivní webovou prezentací a její tvorbou
2. Realizace webového portálu skautské organizace
3. Vytvoření administrační sekce s možností úpravy jednotlivých kategorií portálu
4. Systém registrace uživatelů a následná možnost sledování změn na stránkách emailem

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Kosek J.: PHP – tvorba interaktivních internetových aplikací, Grada 1998

Kolektiv autorů: PHP Programujeme profesionálně, Computer Press,a.s. 2001

Meyer E.: Erik Meyer o CSS – Ovládněte kaskádové styly, Zoner Press, 2004

Vedoucí bakalářské práce:

RNDr. Ing. Miloš Krčmář

Ústav aplikované informatiky

Datum zadání bakalářské práce:

14. února 2006

Termín odevzdání bakalářské práce:

16. června 2006

Ve Zlíně dne 14. února 2006



prof. Ing. Vladimír Vašek, CSc.
pověřený děkan



doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Bakalářská práce se zabývá tvorbou webových aplikací, zvláště pak prezentací Skautského oddílu Domanín. Zkoumá metody a techniky potřebné pro tvorbu dynamických stránek, zejména se zaměřuje na základní programovací metody (X)HTML, CSS, PHP a MySQL a jejich vzájemné propojení. Součástí práce je i stručný popis jednotlivých částí samotných internetových stránek.

Klíčová slova: webový portál, dynamické stránky, databáze

ABSTRACT

The bachelor thesis deals with development of web applications, particularly the presentation of Scout group Domanín. The work examines methods and techniques necessary for development of dynamic web pages, where the accent is put on basic programming methods like (X)HTML, CSS, PHP, MySQL and their mutual connection. One part of the work also describes individual parts of the web portal itself.

Keywords: web page, dynamic pages, database

Tímto bych chtěl poděkovat vedoucímu mé bakalářské práce panu RNDr. Ing. Miloši Krčmářovi za odborné vedení, cenné rady a podnětné připomínky udílené během vypracovávání této práce. Poděkování patří taky celé mojí rodině za podporu a finanční zabezpečení.

Prohlašuji, že jsem zadanou bakalářskou práci vypracoval samostatně a veškerou použitou literaturu citoval.

Ve Zlíně, 11. 08. 2006

.....
Jan Šlahůnek

OBSAH

ÚVOD	9
1 TVORBA DYNAMICKÝCH STRÁNEK	10
1.1 Skripty běžící na straně serveru	10
1.1.1 CGI – skripty	10
1.1.2 Server Side JavaScript (SSJS)	11
1.1.3 Active Server Page (ASP).....	11
1.1.4 Hypertext Preprocessor (PHP).....	11
1.2 Skripty běžící na straně klienta	12
1.2.1 JavaScript.....	12
1.2.2 Java - Aplet	12
1.2.3 Dynamic HTML (DHTML).....	12
2 ÚVOD DO TVORBY WEBOVÝCH APLIKACÍ	13
2.1 XHTML	13
2.2 Kaskádové styly (CSS)	16
2.2.1 Syntaxe CSS	16
2.3 Hypertext Preprocessor (PHP).....	19
2.3.1 Vstup do PHP	19
2.3.2 Vkládání PHP skriptů do HTML kódu	19
2.3.3 Syntaxe PHP	20
2.4 Relační databáze	21
2.4.1 Návrh databáze pro web	23
2.4.2 Databáze MySQL	24
2.4.3 MySQL dotazy.....	24
2.4.4 Výpis základních funkcí pro práci s databází MySQL	26
3 TVORBA WEBOVÉ PREZENTACE	28
3.1 Použitý software a hardware při tvorbě prezentace	28
3.1.1 Použitý software	28
3.1.2 Použitý hardware	29
3.2 Konstrukce kódu	30
3.2.1 Základní popis.....	30
3.2.2 Popis jednotlivých odkazů menu	30
3.2.3 Systém registrace uživatelů	32
3.3 Správa stránek.....	35

3.3.1	Přihlášení uživatelů.....	35
3.3.2	Popis jednotlivých odkazů administrátorského menu	35
3.3.3	Další možnosti administrátora systému	39
3.4	Databáze v praxi	40
3.4.1	Struktura použitých tabulek	41
ZÁVĚR		43
SEZNAM POUŽITÉ LITERATURY.....		44
SEZNA POUŽITÝCH SYMBOLŮ A ZKRATEK.....		45
SEZNAM OBRÁZKŮ		46
SEZNAM TABULEK.....		47
SEZNAM PŘÍLOH.....		48

ÚVOD

Síť WWW (World Wide Web) se mění velmi rychle a mnoha způsoby. Zdá se to jako včera, když byl málo známý jazyk s neobvyklým názvem HTML (Hyper Text Markup Language) používán některými fyziky k propojení vědeckých dokumentů do skupiny serverů CERN. Novinkou bylo, že se text dal číst kdekoliv na světě pomocí jednoduchého programu, a že byl umožněn přesun na jiné související informace. To bylo taky hlavní využití tohoto jazyka v roce 1990, jenž se poměrně rychle rozšířil mezi ostatní vědní obory.

V době spuštění služby World-Wide-Web jsme si vystačili s pouhými třemi technologiemi. První z nich byl již zmíněný jazyk HTML, který sloužil k zápisu webových stránek. Druhou nezbytnou technologií je přenosový protokol HTTP (Hyper Text Transfer Protocol), který zajišťuje přenos HTML-stránek z WWW-serveru do prohlížeče. Třetí technologií nezbytnou pro implementování služby WWW jsou URL (Uniform Resource Locator). Každý objekt přístupný na Webu má svoji jedinečnou URL-adresu, jež slouží k vytváření odkazů na daný objekt.

Těchto technologií se rozhodl využít i skautský oddíl Domanín, jehož členem jsem již dlouhých 12 let. I to byl důvod, proč jsem se rád tohoto projektu ujal. Věřím, že tímto přispěji k lepší komunikaci mezi členy, vedením oddílu, rodiči a další širokou veřejností, jež bude mít možnost sledovat aktuální dění v oddíle a aktivně se účastnit oddílových akcí. K vytvoření tohoto projektu však nestačí tři výše popsané technologie. Je třeba, aby stránky dokázaly pružně reagovat na provedené změny a události. Využil jsem proto výhod, které nabízejí další pokročilejší funkce a jazyky, zejména pak jazyk PHP, a tím dodal projektu patřičnou dynamičnost.

První dvě kapitoly jsou v převážné míře převzaty z literatury, zbylá část práce popisuje vlastní tvorbu webových stránek.

1 TVORBA DYNAMICKÝCH STRÁNEK

Postupným vývojem internetu se zvyšují nároky na jeho využití a tím se zvyšují také nároky na programové vybavení. Vyvíjí se nové programovací jazyky, které se dělí na dvě základní skupiny.

- a) jazyky prováděné na straně serveru
- b) jazyky prováděné na straně klienta

Následující tabulka stručně popisuje základní rozdělní jazyků:

Tabulka I. Dělení jazyků

Na straně serveru	Na straně klienta
CGI-skripty	JavaScript
Server Side JavaScript (SSJS)	Java-applet
Active Server Page (ASP)	Dynamic HTML (DHTML)
HyperText Preprocesor (PHP)	

Výše uvedené skupiny technologií nalézají využití každá zvlášť, ale v mnoha případech dochází k jejich kombinování. Proto se dá říct, že ani jedna není hodnotnější než ta druhá.

1.1 Skripty běžící na straně serveru

Skripty běžící na straně serveru mají značnou výhodu v tom, že jsou nezávislé na klientském prohlížeči a mohou využívat objekty umístěné na serveru, nebo přistupovat k databázi a ke klientovi přenášet konkrétní údaje. Mezi nejrozšířenější skriptovací jazyky běžící na straně serveru patří ASP a PHP. Pro svoji práci jsem využil jazyk PHP, kterým se budeme dále podrobněji zabývat.

1.1.1 CGI – skripty

Jejich název je odvozen od rozhraní CGI (Common Gateway Interface). Toto rozhraní definuje způsob spuštění programu a předávání dat mezi WWW-serverem a programem. Psaní CGI-skriptů nebylo úplně snadné. Pro jejich psaní se používali nejčastěji různé interpretované programy, jako Pearl, nebo příkazové shelly Unixu. Nebyl však problém použít v podstatě libovolný programovací jazyk, a tak existuje mnoho CGI-skriptů napsaných v jazycích C, C++ a Java.

Pro tvorbu CGI-skriptu byla nutná znalost nějakého programovacího jazyka. Kromě toho musel člověk ovládat rozhraní CGI, které nepředávalo parametry zrovna v šikovním formátu. V CGI-skriptech navíc spoustu práce stálo dostatečné zabezpečení skriptu.

CGI-skripty generují svůj výstup v jazyce HTML. Jazyk HTML však nejde přímo kombinovat s jinými jazyky. Každý řádek HTML kódu se zadával jako parametr příkazu print či echo podle použitého jazyka. Správa větších aplikací je rovněž náročná, protože aplikace je roztroušena v mnoha samostatných souborech s HTML stránkami a CGI skripty. [1]

1.1.2 Server Side JavaScript (SSJS)

Vzhledem k obrovskému úspěchu JavaScriptu, firma Netscape vyvinula tuto technologii. Jde o využití JavaScriptu na straně serveru. Na serverech Netscape tak šlo do HTML stránek psát skripty, které se provedou přímo na serveru. Uvnitř stránky byly skripty uzavřeny mezi tagy <SERVER> a </SERVER>, server tak snadno rozpoznal, které části stránky se mají interpretovat. Výsledkem skriptů musel být HTML kód, který se doplnil do zbytku stránky a prohlížeči se již zaslala obyčejná HTML stránka. SSJS umožňuje snadnou práci s daty z formulářů a databázemi. Výsledný efekt těchto aplikací je stejný jako u CGI skriptů, hlavní výhoda spočívá v jednoduchosti SSJS.

1.1.3 Active Server Page (ASP)

Active Server Pages jsou obdobou SSJS. Za vznikem této technologie stojí firma Microsoft, která nechtěla zůstat pozadu a reagovala tak na SSJS. Jako programovací jazyk je možné použít VBScript - což je skriptovací jazyk velmi příbuzný Visual Basicu, JScript - což je implementace JavaScriptu od Microsoftu nebo PerScriptl a Python. Systémy samozřejmě nejsou kompatibilní – ASP používá jiné značky k oddělení skriptu od stránky a hierarchie objektů, které zpřístupňují všechny důležité údaje, je rovněž rozdílná. Vytvořené skripty se mohou odvolávat na komponenty, které se zpracovávají na web serveru a zajistí přístup k databázím a k dalším aplikacím a prostředkům. Server, který zpracovává skript, vytvoří HTML stránku, která je odeslána ke zpracování prohlížečem klienta. Zpracování na straně serveru nesnižuje výkonnost a funkčnost prohlížeče.

1.1.4 Hypertext Preprocessor (PHP)

PHP je skriptovací jazyk zabudovaný na straně serveru. To znamená, že pracuje uvnitř dokumentu HTML a propůjčuje mu tak schopnost generování požadovaného obsahu.

Hlavní výhodou PHP oproti ASP a SSJS je to, že je zcela zdarma, na rozdíl od těchto komerčních produktů. Dalšími výhodami jsou jednoduchost, nejpřirozenější práce s databázemi a nezávislost na platformě. Lze ho zprovoznit na serveru, který běží pomocí OS Linux nebo třeba OS Windows. Bylo vytvořeno v roce 1994 a je šířeno jako open source produkt. Současná hlavní verze tohoto produktu nese číslo 5. [1,2]

1.2 Skripty běžící na straně klienta

Skripty běžící na straně klienta nepotřebují ke svému běhu žádnou zpětnou vazbu na server, ale pracují pouze na uživatelské počítači. Tím je dosažena podstatně rychlejší reakce na vyvolanou akci, což je jeden z největších kladů těchto skriptů. Další využití je v kontrole zadaných údajů ve vyplněných formulářích před zpracováním serverem, nebo dynamickému vytváření menu a jiným úpravám statické stránky, jako je například zobrazení aktuálního data, času apod. Některé technologie nejsou nezávislé na platformě (ActiveX). Tyto skripty jsou nejčastěji psány v JavaScriptu. [3]

1.2.1 JavaScript

V roce 1996 vznikl nový jazyk s názvem JavaScript. Tento jednoduchý jazyk se syntaxí vycházející s jazykem Java, byl produktem firmy Netscape, která jej zabudovala do svého legendárního prohlížeče Netscape Navigator. JavaScript se zapisoval přímo do HTML-kódu stránky a uměl posloužit v mnoha situacích. Jeho nejčastější použití bylo ve spojení s formuláři, avšak využívá se i na měnění polohy okna prohlížeče, k rozpoznávání prohlížeče a na další vylepšení interaktivnosti stránek.

1.2.2 Java - Aplet

Jazyk Java je produktem firmy Sun Microsystems. Tento jazyk umožňuje psaní Java-pletů, což jsou krátké programy začleněny přímo do HTML kódu. Na webové stránce mají vyhrazen prostor, který je zcela pod jejich kontrolou. Využití java-pletů je v různých webových animacích, hrách a třeba navigacích webu. Jejich nevýhodou je pomalé načítání a špatná podpora některých prohlížečů. Naopak výhodou je nezávislost na platformě. Pro korektní zobrazení stránek, které používají Java Aplet, je třeba mít prohlížeč s podporou těchto apletů, nebo mít nainstalovaný JVM (Java Virtual Machine).

1.2.3 Dynamic HTML (DHTML)

V roce 1997 přišel Microsoft ve svém prohlížeči MSIE 4.0 s novou technologií dynamického HTML (DHTML). DHTML vychází ze stejné myšlenky jako JavaScript, ale

mnohem více ji dotahuje. V DHTML je přístupný každý objekt webovské stránky a s každým objektem můžeme kdykoliv manipulovat – měnit jeho obsah, způsob zobrazení či polohu. DHTML je ve skutečnosti kombinace CSS (Cascading Style Sheet), DOM (Document Object Model), skriptovacích jazyků (JavaScript, VBScript) a HTML. Nelze ho tedy chápat jako úplně nový skriptovací jazyk, ale jako směsici výše uvedených jazyků. [4]

2 ÚVOD DO TVORBY WEBOVÝCH APLIKACÍ

V dnešní době je důležité uvědomit si, že ač je efektivní web design, který hezky vypadá důležitý, není to všechno co by měl web mít. Pokud je webový server špatně vytvořen, nebo obsahuje spoustu chyb, uživatelé přestanou mít zájem takový web navštěvovat a brzy na něj zapomenou. Důležitou součástí webových komunikací je samotný prohlížeč (anglicky browser). Jedná se o program, který dekóduje poměrně úsporná data a jednoduché instrukce zaslané serverem, sestaví z nich stránku a tu zobrazí ve svém hlavním okně. Proto je velmi důležité znát podporu webového prohlížeče a jeho schopnosti. Pokud tyto znalosti nemáme, může se lehce stát, že naše prezentace se stanou nedostupné pro spoustu uživatelů využívající jiné prohlížeče. [5]

2.1 XHTML

Jazyk XHTML je přetvoření jazyka HTML pomocí XML (Extensible Markup Language). XML je stejně jako HTML odvozený z jazyka SGML, ale nabízí podstatně větší možnosti. Zjednodušeně řečeno, XHTML popisuje, jaká pravidla musí designér dodržovat při psaní HTML, aby bylo učiněno zadost i standardu XML. Takže pokud se bude při psaní webových stránek brát v úvahu XHTML, je zaručeno že dokument vyhovuje jak standardu HTML, tak i XML. Například v HTML lze psát veškeré značky velkými i malými písmeny dle libosti. Podle XHTML je naopak nutné psát veškeré značky malými písmeny a atributy musí mít vždy uvozovky.

Jazyk (X)HTML je charakterizován množinou značek a jejich atributů, které jsou pro danou verzi jazyka HTML definovány. Mezi značky se uzavírají části textu a tím se určuje význam (sémantika) obsaženého textu. Názvy jednotlivých značek se uzavírají mezi úhlové závorky („<“ a „>“). Část dokumentu mezi značkami tvoří tzv. element (prvek) dokumentu. Elementy mohou být párové i nepárové.

Obecná struktura stránky

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs">
<head>
<title>.:6. skautský oddíl Domanín.:</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2" />
<link rel="stylesheet" type="text/css" href="style.css"/>
<link rel="shortcut icon" href="/pictures/ymca.ico" />
<meta name="author" content="Jan Slahunek" />
<meta name="description" content="YMCA Skaut Domanín" />
<meta name="keywords" content="YMCA, ymca, skaut, Domanín, skautský oddíl,
domanín" />
<script src="java.js" type="text/javascript"></script>
</head>
<body>
.
.
.
</body>
</html>
```

Pokud má být webová stránka zároveň XML dokumentem, musí se nějak určit, kde se najde tzv. DTD (Document Type Definition) soubor. Je to návod pro prohlížeč zpracovávající dokument. Říká mu, jaké elementy dokument používá a jak s nimi zacházet. Existují tři definice DTD a každá z nich definuje jinou sadu XHTML tagů, událostí atd.

Stránky typu Strict:

To jsou webové stránky, které jdou s dobou a jejichž autoři se rozhodli očistit je od všech relikvií, které jsou v jazyce XHTML i HTML jako pozůstatek minulých dob a zároveň se ukázaly jako nevyspělé. Jednoduše řečeno, mnoho věcí jde v obou jazycích udělat více způsoby a autoři stránky s označením Strict používají ze všech možných způsobů vždy ten nejnovější. Starší způsoby jsou pro ně tabu.

Příklad deklarace DTD v dokumentu

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"DTD/xhtml11-strict.dtd">
```

Stránky typu Transitional:

To jsou normální webové stránky, které používají vše, co vyhovuje standardu, ale nezřikají se ani starších možností, pokud standardu vyhovují. Nepoužívají ale rámy (frames) a vše, co s tím souvisí.

Příklad deklarace DTD v dokumentu

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"DTD/xhtml11-transitional.dtd">
```

Stránky typu Frameset:

Tato kategorie zahrnuje úplně vše, co kategorie předcházející a navíc přidává rámy. Najde ale nejmenší uplatnění, protože rámců už se v dnešní době moc nevyužívá.

Příklad deklarace DTD v dokumentu

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"DTD/xhtml11-frameset.dtd">
```

2.2 Kaskádové styly (CSS)

CSS (Cascading Style Sheets) neboli kaskádové styly vznikly jako souhrn metod pro úpravu vzhledu stránek. První návrh normy byl zveřejněn v roce 1994, v roce 1996 byla pak vydána specifikace CSS 1, v roce 1998 CSS 2, nyní se pracuje na verzi CSS 3. CSS se využívá k formátování obsahu HTML, XHTML a XML dokumentů. Ve srovnání s formátováním pomocí atributů v HTML formátovací schopnosti rozšiřuje. Styly umožňují přesně určit, jak bude který element vypadat. Narozdíl od atributů lze stylem definovat jednotný vzhled elementu pro celý dokument (např. že všechny nadpisy úrovně 1 budou červené) a to jediným zápisem pro příslušný element (nikoli v každém tagu příslušného elementu). Stejně tak lze pomocí stylu určit odlišné formátování pro třeba jen jediný výskyt určitého elementu. Tím se dá jednak zbavit velkého množství kódu, jednak se tento kód stane mnohem přehlednější. Navíc pokud se jednou změní například barva písma všech odstavců, je to otázka několika málo vteřin, měnit každý atribut u každého elementu v HTML by byla katastrofa. Jeden styl lze snadno použít pro libovolné množství stránek. [6]

2.2.1 Syntaxe CSS

Styl se skládá z pravidel pro jednotlivé elementy, které mají být formátovány. Každé takové pravidlo má dvě části, **selektor** (název elementu, pro který má toto pravidlo platit) a **deklaraci** (co pro něj má platit). V deklaraci se určuje vlastnost a její hodnota. Je uzavřena do složených závorek a zápis vypadá takto:

```
selektor {vlastnost: hodnota_vlastnosti}
```

Konkrétně:

```
h1 {color: green}
```

Selektorem, tedy elementem, který formátuje je zde *h1* (nadpis 1. úrovně). Deklarací je *{color: green}*. Ta určuje, že vlastnost *color* bude mít hodnotu *green*. Celé dohromady to tedy znamená, že všechny nadpisy 1. úrovně v dokumentu budou mít zelenou barvu.

Elementu lze určit více než jednu vlastnost, jednotlivé vlastnosti se od sebe oddělují středníkem. Takto lze definovat libovolné množství vlastností.

Konkrétně:

```
selektor {vlastnost1: hodnota_vlastnosti1; vlastnost2: hodnota_vlastnosti2;}
```

Dvěma elementům lze určit jejich společná vlastnost tak, že se od sebe oddělí jednotlivé selektory čárkou.

Konkrétně:

```
selektor1, selektor2 {vlastnost: hodnota_vlastnosti;}
```

Dědičnost

Většina vlastností se dědí. To znamená, že element, který nemá vlastnost definovanou jí dědí po nadřazeném elementu. Týká se to především vlastností písma — barvy, velikosti, stylu atd. Pokud tedy chceme definovat nějakou vlastnost, kterou budou mít všechny elementy společnou (a později případně je vytvářet výjimky) definujeme ji pro element *body*.

Komentáře

Pokud si chceme ke stylu psát nějaké poznámky pro lepší orientaci, zapíšeme ji do komentářů. Ty se v CSS tvoří pomocí `/*` a `*/`. Mezi hvězdičky pak můžeme umístit i několikařádkový komentář, ten se samozřejmě ve výsledném zobrazení neobjeví.

Konkrétně:

```
body {color: green} /* tady si píší komentář, že mám všechny texty zelené*/
```

Připojení stylu k dokumentům

Styl můžeme k dokumentu připojit několika způsoby, můžeme definovat přímo v dokumentu nebo v externím souboru, způsoby můžeme i kombinovat.

Externí soubor

Pokud chceme mít styl uložený v externím souboru (což je velmi výhodné při používání jednoho stylu pro více dokumentů), v nějakém textovém editoru uložíme námi definovaný styl do souboru s příponou `css`. Ten pak připojíme k dokumentu zápisem v hlavičce (tj. mezi tagy `<head>` a `</head>`) buď v tagu *link* nebo *style*.

Konkrétně:

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

```
<style type="text/css">@import "styl.css";</style>
```


Pokud je styl umístěn na jiném serveru, tak můžeme použít zápis:

```
<style type="text/css">@import url("http://www.neco.cz/styl.css");</style>
```

Zápisem `@import "styl.css";` můžeme také vložit jeden styl do druhého stylu.

Definování stylu uvnitř dokumentu

To můžeme provést opět v tagu *style* kam tentokrát místo odkazu na externí styl umístíme přímo definici stylu.

Konkrétně:

```
<style type="text/css">body {color: green}</style>
```

Nebo můžeme definovat styl přímo nějakému elementu, což se hodí zvláště v případě, kdy máme definovaný jednotný styl, ale pro například jedno konkrétní slovo chceme použít jiné pravidlo. Potom použijeme v příslušném tagu atribut *style*.

Konkrétně:

```
<h1 "style=color: green">nadpis</h1>
```

Váha stylů

Pokud ve stylu definujeme pro stejný element stejnou vlastnost dvakrát, vyšší váhu má ta deklarace, která byla definovaná později (myšleno na pozdějším řádku) a ta se také provede. Pokud bychom chtěli některé deklaraci přiřadit větší důležitost, použijeme *!important*.

Konkrétně:

```
h1 {color: green !important}
```

2.3 Hypertext Preprocessor (PHP)

2.3.1 Vstup do PHP

PHP je skriptovací jazyk zabudovaný na straně serveru. To znamená že pracuje uvnitř dokumentu HTML a propůjčuje mu tak schopnost generování požadovaného obsahu.

Jeho vznik se datuje roku 1994, kdy Rasmus Lerdorf dal dohromady kombinaci skriptů v Perlu, aby zjistil kdo se díval na jeho výsledky. Pozvolna se lidé začali o tyto skripty zajímat, takže byly později vydány jako balíček „Personal Home Page“ tools (původní význam PHP). Později lidé začali používat tyto nástroje pro mnohem komplikovanější věci a vývoj přešel z jednoho člověka na skupinu hlavních programátorů. Jazyk prošel velkým vývojem a nyní je z něj vospělý a široce uplatnitelný produkt.

PHP kód je součástí klasického HTML kódu. Například můžeme mít HTML stránku s částmi PHP, které zpracovávají, nebo jen kontrolují různé informace. Přednost PHP je v tom, že může informace buď samostatně zpracovávat nebo přímo generovat HTML kód. Pokud se prohlížeč snaží do počítače natáhnout stránky, které obsahují PHP, je PHP kód zpracován modulem Apache (nebo CGI) a vykonán. Tento spuštěný kód nahradí původní PHP kód, přičemž původní HTML zůstane nedotčeno.

PHP je produkt Open source, což znamená že má každý přístup k jeho zdrojovému kódu, může ho používat, upravovat a dál distribuovat, vše bez jakýchkoliv poplatků. [2]

2.3.2 Vkládání PHP skriptů do HTML kódu

Zdrojové kódy PHP skriptu se vkládají přímo do HTML kódu stránky. PHP se vkládá také do známých tagů, které musí být správně ukončeny (aby server poznal kde jednotlivé skripty končí). Je několik možností, jak vložit PHP skript do HTML kódu.

Krátký styl

Tato možnost je nejvhodnější pro jednoduché a krátké kódy.

Konkrétně:

```
<? echo ‘<p>Způsob vkládání PHP skriptů</p>’; ?>
```

Použití toho stylu se nedoporučuje, protože ačkoliv je standardně povolen, správci jej příležitostně vypínají. Navíc nevyhovuje formátu XML.

XML styl

Tento způsob se používá pro lepší členění kódu a pro generování XML stránek.

Konkrétně:

```
<?php echo '<p>Způsob vkládání PHP skriptů</p>'; ?>
```

Tento styl značek je asi tou nejlepší variantou, protože administrátor serveru ho nemůže vypnout. Tím lze garantovat, že bude dostupný na všech serverech.

SCRIPT styl

Způsob vhodný pro vkládání složitých kódů.

Konkrétně:

```
<script language='php'> echo '<p> Způsob vkládání PHP skriptů </p>'; </script>
```

Tento styl značek je nejdelší, znají především uživatelé, kteří používají JavaScript nebo VBScript.

2.3.3 Syntaxe PHP

Syntaxe PHP se velice podobá syntaxi programovacího jazyka C. Každý příkaz musí být oddělen středníkem od příkazu dalšího.

Konkrétně:

```
<?
```

```
echo 'První řádek';
```

```
echo 'Druhý řádek';
```

```
?>
```

Komentáře v PHP

Komentář je něco, co program při zpracování ignoruje a slouží tedy pouze uživateli. Vysvětluje účel skriptu, kdo ho napsal, proč ho napsal a tak dále. Komentáře mohou být v PHP jednořádkové nebo víceřádkové a používají se na ně znaky "//" (dvě dopředná lomítka), "#" (mřížka) a dvojice "/*" a "*/" (lomítko+hvězdička --- hvězdička+lomítko).

Proměnné v PHP

Jako všechny programovací jazyky, i PHP nám umožňuje uložit části dat do proměnných a pak k nim přistupovat pomocí zápisu názvu proměnné. V PHP musí názvy všech proměnných začínat znakem dolar \$. V PHP nemusíme uvádět typy proměnných, to znamená, že nemusíme naše proměnné deklarovat a uvádět druh dat, který budou obsahovat. V PHP se proměnné inicializují typicky přiřazením hodnoty.

Konkrétně:

```
<?
```

```
$uzivatel = Honza;
```

```
echo '$uzivatel';
```

```
?>
```

2.4 Relační databáze

Relační databáze ukládají data místo do jednoho velkého uložště spíše do oddělených tabulek. Zajišťuje to rychlost a flexibilitu. Přístup k údajům uloženým v databázi obstarává program, kterému se říká DBMS (Database Management System).

Tabulka II. BDSM

DBMS
MS SQL Server
dBase
Access
Informix
Oracle
Sybase
MySQL

Relační databáze prorazily hlavně svou jednoduchostí - všechno je uloženo v tabulkách, tabulky mají sloupce, v každém sloupci jsou data určitého typu.

Sloupcům tabulky obvykle říkáme v databázové terminologii *položky* nebo *atributy*. Jednotlivé řádky se pak nazývají *záznamy*. Aby šlo s tabulkami a v nich uloženými údaji pracovat, musí být nějak jednoznačně identifikovány. Každý sloupec je proto pojmenován – má svůj název. Tento název pak používáme, když se odvoláváme na obsah určitého atributu, a ne na celý záznam.

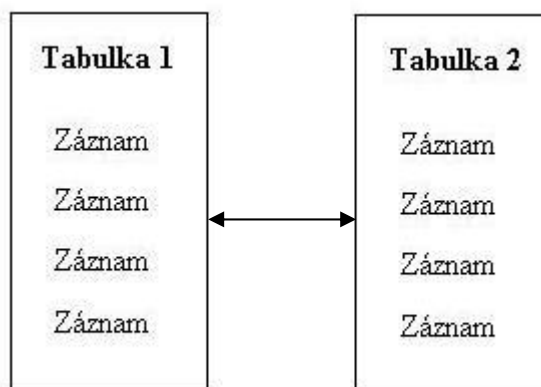
Každá tabulka musí obsahovat tzv. *primární klíč*. Primární klíč je atribut, jehož hodnota je pro každý záznam jedinečná. Databáze se většinou skládá z několika tabulek a používá klíče jako odkazy z jedné tabulky na jinou. [2]

Nastavení relací mezi aplikačními objekty

Relace je něco, co vytváří vztah mezi jednotlivými objekty, v tomto případě tabulkami. Je tedy nutné vyhledat jednotlivé vztahy mezi objekty (tabulkami) a následně určit, o jaký druh relace se jedná. Relace jsou trojího druhu. 1:1, 1:N nebo M:N, kde M a N jsou celá čísla větší než 1.

Relace 1:1

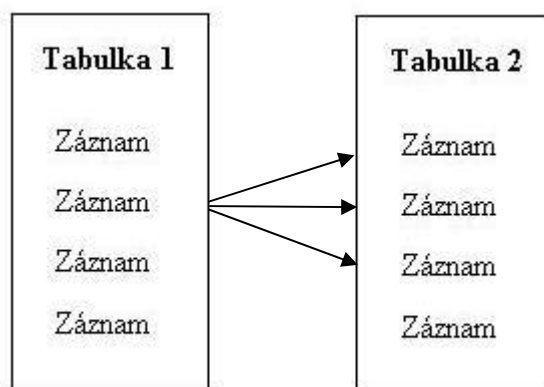
Relace 1:1 je nejsnazší typ relace. Tato relace vzniká mezi dvěma tabulkami, kde jedna hodnota primárního klíče v hlavní tabulce odpovídá hodnotě pole v druhé tabulce.



Obr. 1. Relace 1:1

Relace 1:N

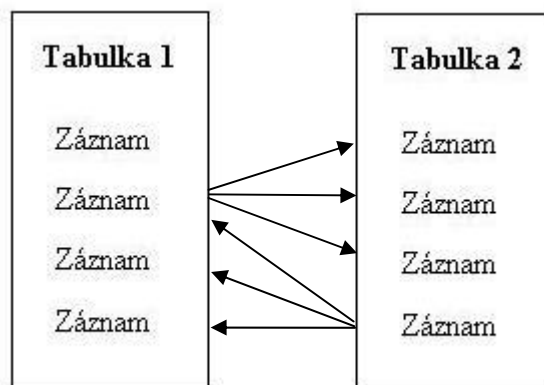
Tato relace opětovně vzniká mezi dvěma tabulkami, kde jedna hodnota primárního klíče v hlavní tabulce odpovídá několika hodnotám pole v druhé tabulce.



Obr. 2. Relace 1:N

Relace M:N

Tato relace vzniká mezi dvěma tabulkami, kde každý záznam (jedno které z těchto tabulek) může odpovídat několika záznamům z druhé tabulky.



Obr. 3. Relace M:N

2.4.1 Návrh databáze pro web

Při tvorbě databáze je nejdůležitější její návrh. Se špatným návrhem totiž padá celá databáze. Při návrhu je tedy nutno brát ohled zejména na předpokládané požadavky (hlavně dotazy). Důležité totiž je, aby v navržené databázi šly rychle realizovat vyhledávací a aktualizací dotazy.

2.4.2 Databáze MySQL

Jednoduše se dá říci, že MySQL je databázový systém. Takových systémů existuje celá řada, například Oracle, MS SQL, nebo třeba Sybase. MySQL je systém, který se etabloval především ve webových aplikacích, a který je dost preferovaný při spolupráci s PHP. Nejpoužívanějším jazykem pro přístup k relačním databázovým systémům (RDBMS) je jazyk SQL. Používá se k ukládání dat do databází a k získávání dat s databází.

SQL je anglická zkratka pro System Query Language, což je vlastně programovací jazyk, jímž se pomocí klienta ptáme SQL serveru a ten následně při správně položeném dotazu odpoví. Odpověď je formulována většinou do tabulky o jednom, více či dokonce žádném řádku (to pokud dotazu nevyhoví ani jedna položka). Z tabulky pak můžeme pomocí skriptů "vytáhnout" potřebná data a dále je používat. Mezi standardním SQL a SQL implementovaným v MySQL existují drobné rozdíly.

MySQL je však víc než jen pouhá databáze. Je to systém, který databázi spravuje. [2]

2.4.3 MySQL dotazy

SELECT – příkaz pro vybrání řádku z databáze

Parametry:

SELECT	základní příkaz, určuje že chceme něco vybrat
sloupec1,sloupec2	sloupce které chceme vybrat oddělené čárkou. Pokud chceme vybrat všechny sloupce, můžeme použít místo vyjmenování znak *
FROM	říká že teď přijde název tabulky ze které vybíráme data
tabulka	název tabulky
	při výběru jsou i další možnosti jako WHERE, ORDER, LIMIT

Konkrétně:

```
SELECT * FROM uzivatele WHERE jmeno='Jan';
```

Tento příkaz vybere z tabulky „uzivatele“ všechny řádky a jejich hodnoty, ve kterých je sloupec „jmeno“ roven hodnotě „Jan“.

INSERT - příkaz pro vkládání řádků do tabulky

Parametry:

INSERT INTO	základní příkaz určující že chceme vložit řádek do tabulky
tabulka	název tabulky
sloupec1,sloupec2	vyjmenování sloupců do kterých budeme vkládat hodnoty (musíme dávat pozor na pořadí)
VALUES	nyní přijdou hodnoty
sloupec1,sloupec2	vyjmenování hodnot musí být ve stejném pořadí jako sloupce

Konkrétně:

```
INSERT INTO clenove (jmeno,prijmeni) VALUES ('.$jmeno.', '$prijmeni.');
```

Tento příkaz vloží do tabulky „clenove“ údaje, které namísto \$jmeno a \$prijmeni nahradí údaji z vyplněného formuláře při vložení nového člena do databáze.

UPDATE – příkaz pro upravení řádku v databázi

Parametry:

UPDATE	základní příkaz určující že chceme něco upravit
tabulka	název tabulky
SET	chceme upravit:
sloupec1="hodnota", sloupec2="hodnota2"	jakou hodnotu chceme přiřadit kterému sloupci

Konkrétně:

```
UPDATE clenove SET jmeno = 'Pavel' WHERE jmeno = 'Jan';
```

Vykonáním tohoto příkazu docílíme změny všech jmen „Jan“ na jméno „Pavel“ .

DELETE – příkaz pro vymazání dat z databáze

Parametry:

DELETE FROM	chceme smazat řádku z tabulky
tabulka	název tabulky
	zde je podstatný příkaz WHERE , jelikož bez něj dojde k smazání všech záznamů v tabulce

Konkrétně:

```
DELETE FROM clenove WHERE jmeno = 'Jan';
```

Vykonání tohoto příkazu má za následek vymazání řádku se jménem Jan.

2.4.4 Výpis základních funkcí pro práci s databází MySQL**mysql_connect**

Funkce, která vytvoří spojení s MySQL serverem.

mysql_close

Funkce, která ukončí (zavře) spojení s MySQL serverem.

mysql_affected_rows

Funkce, která vrátí počet změněných záznamů v MySQL po posledním dotazu.

mysql_create_db

Funkce, která vytvoří MySQL databázi.

mysql_db_query

Funkce, která provede MySQL dotaz.

mysql_error

Funkce, která vrátí text chybové zprávy předchozího MySQL příkazu.

mysql_fetch_array

Funkce, která načte výsledný řádek do asociativního, čís.pole nebo obojího.

mysql_fetch_assoc

Funkce, která načte výsledný řádek do asociativního pole.

mysql_fetch_row

Funkce, která načte výsledný řádek do pole.

mysql_free_result

Funkce, která uvolní výsledek z paměti.

mysql_num_fields

Funkce, která vrátí počet položek ve výsledku.

mysql_num_rows

Funkce, která vrátí počet záznamů ve výsledku.

mysql_query

Funkce, která pošle MySQL dotaz.

mysql_result -

Funkce, která načte výslednou hodnotu jedné položky.

mysql_select_db

Funkce, která vybere MySQL databázi.

mysql_tablename

Funkce, která načte jméno tabulky.

3 TVORBA WEBOVÉ PREZENTACE

Ke tvorbě webové prezentace jsem se snažil využít prostředků a možností, které současná doba nabízí. Dynamičnosti stránek jsem docílil pomocí programovacího jazyka PHP a databáze MySQL. Rozvržení prezentace je vytvořeno pomocí CSS. Pro další formátování je z větší části využito tabulek z důvodu jednodušší optimalizace pro různé prohlížeče. V jednom případě jsem použil skriptovací jazyk JavaScript, další využití jsem pro něj ve své prezentaci nenašel. Stránky byly testovány a optimalizovány pro nejrozšířenější prohlížeče Internet Explorer 6.0 a Mozilla Firefox 1.5.

3.1 Použitý software a hardware při tvorbě prezentace

3.1.1 Použitý software

Tvorba webové prezentace se téměř neobejde bez specializovaných programů, které velmi zpříjemní a zjednoduší práci. I já jsem využil možností šikovných utilit, které se mi pro tuto práci nabízely.

PHP, HTML, CSS a JavaScript

Pro programování PHP, HTML a CSS jsem používal volně šiřitelný program PSPad od firmy Prog-Soft s.r.o.. Ačkoliv tento program neoplývá maximem různých funkcí, jeho zvýrazňování syntaxe kódu a především velká rychlost byly vlastnosti, které mi vyhovovaly natolik, že jsem se rozhodl právě pro něj. To samozřejmě nejsou všechny možnosti, které tento šikovný program nabízí, pro mě však byly těmi nejdůležitějšími.

Grafika

Při vytváření grafiky jsem se spoléhal pouze na jeden, ale za to velmi mocný grafický editor. Tím byl Adobe Photoshop CS. Tento nástroj nabízí velké spektrum funkcí a možností pro vytváření designu.

3.1.2 Použitý hardware

PC sestava

Vytvoření prezentace nevyžadovalo zvlášť velké nároky na použitý hardware, jediné větší nároky snad vznikly při tvorbě designu použitím grafického editoru. Zde by se uplatnilo rozšíření operační paměti a rychlejší procesor. Konfigurace počítače, na kterém vznikala prezentace je následující:

- Procesor AMD Athlon XP 1800+
- Operační paměť DDRAM 256MB
- Grafická karta NVIDIA GeForce4 Ti 4200 128MB
- Pevný disk Maxtor 80GB

Webový server

Při realizaci projektu se nabízelo více možností, kde testovat webové stránky. Ať už to byly programy, pomocí kterých lze server provozovat přímo na PC, nebo různé internetové servery. Rozhodl jsem se pro placený server společnosti Dobraspolecnost, který nabízí oproti freehostingovým serverům řadu výhod. Za mírný poplatek jsem tak své stránky mohl umístit na server s dobrou dostupností a s podporou technologií, které jsem při své práci využíval. Navíc tak lze celou aplikaci lépe otestovat. Webhosting má následující parametry:

- Velikost prostoru 400MB
- 10 e-mailových schránek
- Podpora PHP, MySQL 4.0.x, POP3, SMTP
- Možnost využití SSL (https)
- 2GB limit přenosu přes web

[9]

Webová prezentace je dostupná na adrese <http://www.ymcadomanin.info>

3.2 Konstrukce kódu

3.2.1 Základní popis

Nejdůležitějším prvkem při tvorbě webové prezentace je technologie CSS. Pomocí této technologie můžeme umisťovat jednotlivé prvky na stránce, formátovat a vytvářet konečný vzhled stránek.

Základní rozvržení stránky

Základní rozvržení stránky se provádělo několika způsoby. Buď pomocí rámců nebo pomocí tabulkového layoutu. Druhá jmenovaná technika je často používána i dnes, ale má řadu nevýhod, mezi něž patří nahuštění zdrojového kódu. To vede k vyčerpání přenosové kapacity a prodloužení doby načítání takovýchto stránek. Nejlepší možností je využití CSS. U tohoto projektu je základní rozvržení stránky tvořeno technologií plovoucích prvků, ale u dalšího formátování je využito i tabulek.

Menu prezentace

Pro svou prezentaci jsem zvolil svislé menu, které je realizováno pomocí elementů sloužících pro tvorbu seznamu - `` a ``. Jeho výsledné podoby bylo dosaženo pomocí CSS. Funkčnost tohoto menu musí být zajištěna v kterémkoliv prohlížeči, protože se jedná o základní nabídku stránek a bez něj by stránky neměly smysl. Ve webovém prohlížeči Mozilla Firefox byla funkčnost menu bezproblémová. V prohlížeči Internet Explorer se objevily drobné odlišnosti, které bylo třeba vyladit úpravou formátování CSS. Nyní je menu optimalizováno pro oba dva prohlížeče.

3.2.2 Popis jednotlivých odkazů menu

Aktuality

Stránka se automaticky načte při zadání adresy <http://ymcadomanin.info>. Obsahuje krátké zprávy informující o nadcházejících událostech, či různých zajímavostech souvisejících s oddílem. Jsou seřazeny od nejaktuálnější po nejstarší. U každé aktuality je zobrazeno datum, kdy byla přidána a také jméno autora, který ji vložil. Návštěvník stránek má tak okamžitý přehled o aktuálním dění a činnosti oddílu.

Naše historie

Zde se návštěvníci mohou dozvědět něco o historii a vývoji skautingu v naší obci, o vzniku našeho skautského oddílu a o událostech, které jeho založení předcházely.

Členové oddílu

Na této stránce je přehledný seznam všech registrovaných členů. U každého člena jsou k dispozici základní údaje, jako jméno, příjmení, datum narození, registrace a družinka, kterou navštěvuje.

Plánované akce

Je to důležitá část stránek, sloužící všem uživatelům, bez ohledu na to, jestli jsou registrovaní členové oddílu nebo ne. Najdou zde seznam všech plánovaných akcí, dočtou se bližší informace a v případě zájmu se na akci mohou předběžně přihlásit. Jedinou podmínkou je registrace na stránkách, kterou má možnost provést každý návštěvník. Přihlášení na akci není závazné, slouží organizátorům jako informace zájmu o akci. K dispozici je potom kompletní seznam přihlášených uživatelů na danou akci.

Fotogalerie

Jak už název napovídá, návštěvníci zde naleznou fotografie z různých akcí a činností oddílu. Fotogalerie je rozdělena do adresářové struktury s jednotlivými miniaturami obrázků, které se po kliknutí myši zobrazí v původní velikosti. Zároveň je k dispozici jednoduchá navigace sloužící k plynulému prohlížení fotografií.

Kontakt

Přesný název organizace, oddílu, adresy, to všechno naleznou návštěvníci na stránce kontakt. A nejen to. Najdou zde kontaktní údaje na jednotlivé funkcionáře naší organizace a eventuálně přihlášku do našeho oddílu.

Návštěvní kniha

Každý uživatel má možnost přispět svými postřehy a poznatky do návštěvní knihy. K tomu stačí vyplnit pouze jméno a text zprávy, zcela na uživateli je, zda zadá i osobní email. Pokud je návštěvník zaregistrován a přihlášen, jméno a email se vyplní automaticky údaji uloženými v databázi, které uživatel zadal při registraci. K oživení knihy a lepšímu vyjádření emocí jsou k dispozici obrázkové smajlíci, které návštěvník vloží pouhým kliknutím myši na něj.

3.2.3 Systém registrace uživatelů

Každý návštěvník stránek, ať už je členem oddílu, nebo se zajímá o činnost oddílu, má možnost zaregistrovat se do osobního portálu kliknutím na odkaz „Registrovat“. Pomocí tohoto portálu pak může požádat o zasílání aktuálních informací na e-mail, může se přihlašovat na jednotlivé akce, má přehled o akcích, na něž je již přihlášen, a je mu povoleno jednotlivé nastavení měnit. Registrační formulář je na Obr. 4. Při registraci musí uživatel vyplnit přezdívkou, pod kterou se bude do portálu přihlašovat. Tato přezdívka je jedinečná a nikdo další už nemůže mít stejnou. Při pokusu registrovat již existující přezdívku je uživatel upozorněn chybovou hláškou (Obr. 5). Dále je nutné, aby každý uživatel vyplnil jméno, příjmení, heslo pro přihlášení a e-mail. E-mailová adresa musí být zapsána ve správném tvaru tzn. `neco@neco.xx` a zároveň tuto adresu již nesmí někdo z uživatelů používat. Pokud nastane některá z těchto situací, je uživatel opět upozorněn chybovou hláškou (Obr. 6). Poslední možností registrace je zaškrtnutím potvrdit zasílání informací a novinek na tuto adresu.

REGISTRACE NOVÉHO UŽIVATELE

Přezdívka:	<input type="text"/>
Jméno:	<input type="text"/>
Příjmení:	<input type="text"/>
Email:	<input type="text"/>
Heslo:	<input type="text"/>
Potvrzení hesla:	<input type="text"/>

Chci dostávat novinky a informace emailem

Obr. 4. Registrační formulář

REGISTRACE NOVÉHO UŽIVATELE

Musíte vyplnit všechny údaje!
Tato přezdívka již existuje, vyberte prosím jinou!

Přezdívka:

Jméno:

Příjmení:

Email:

Heslo:

Potvrzení hesla:

Chci dostávat novinky a informace emailem

Obr. 5. Chybná registrace – přezdívka

REGISTRACE NOVÉHO UŽIVATELE

Musíte vyplnit všechny údaje!
Neplatná emailová adresa!

Přezdívka:

Jméno:

Příjmení:

Email:

Heslo:

Potvrzení hesla:

Chci dostávat novinky a informace emailem

Obr. 6. Chybná registrace – email

Registrovaní uživatelé mají možnost se nezávazně přihlašovat na plánované akce. Seznam plánových akcí je na Obr. 7, kde každý přihlášený uživatel okamžitě vidí, jestli je na akci přihlášen nebo ne. Po kliknutí na odkaz „Detaily“ se zobrazí veškeré dostupné informace o akci (Obr. 8). Datum konání, popis akce, organizátora, počet lidí přihlášených na akci s možností zobrazení seznamu přihlášených účastníků a status uživatele, který je zrovna v portálu přihlášen. Pokud je na akci přihlášen, zobrazí se tlačítko „Odhlásit“, pokud není tak tlačítko „Přihlásit“.

PLÁNOVANÉ AKCE		
Název akce	Datum konání	Detaily
Volby & Sunshine holiday action	20.8.06	Zobrazit
* Domaninske Merunkobrani 2006	15.8.06	Zobrazit
* Extrem tuning day with skoda 120L	10.8.06	Zobrazit
* Kolotoce a houpačky po dlouhé době opět v Domaníně	6.8.06	Zobrazit
* Výlet do přírody pěkně po svých :o)	29.7.2006	Zobrazit

*Hvězdička značí, že jste na příslušnou akci přihlášení

Obr. 7. Seznam plánovaných akcí

AKCE "Výlet do přírody pěkně po svých :o)"	
Datum konání:	29.7.2006
Popis akce:	Krátká výprava do přírody. Půjdeme se podívat do nedalekého lesa, kde proběhnou různé dovednostní soutěže.Hlavně si vezměte dobrou obuv a sportovní oblečení. Zájemci hlase se u Zbyňka.
Organizátor:	Zbyněk a Honza
Přihlášeno lidí:	2 Zobrazit seznam
Můj status:	Jsem přihlášen na akci Odhlásit

Obr. 8. Detailní zobrazení akce

3.3 Správa stránek

Webový portál obsahuje velmi důležitou funkci, bez které by tato aplikace ani neměla smysl, a tím je správa služeb. Její princip je založen na určitém strukturování uživatelů pomocí přidělování oprávnění. Každý uživatel je zařazen do určité kategorie práv, které umožňují rozdílné zásahy do správy. Aplikace rozlišuje 3 kategorie oprávnění.

Tabulka III. Přiřazení kategorií práv

Kategorie	Oprávnění
1	Administrátor systému
2	Administrátor systému s omezením
3	Běžný uživatel

3.3.1 Přihlášení uživatelů

Pro přihlašování a odhlašování lze použít více způsobů. Mezi nejčastější metody patří zajištění přístupu pomocí předdefinovaných systémových proměnných PHP (`$PHP_AUTH_USER` a `$PHP_AUTH_PW`). Tyto proměnné umožňují spolupracovat s autorizačním procesem HTTP. Vzhledem k lepšímu grafickému ztvárnění a menšímu zásahu do grafického konceptu stránky, jsem se rozhodl pro jiné řešení. Jde o metodu využití tzv. řízení sezení. Tato sezení jsou v PHP řízená jedinečným popisovačem sezení (SID), což jsou zašifrovaná náhodná čísla. PHP generuje SID automaticky a ukládá jej po dobu trvání sezení u klienta v tzv. cookie. Samotné přihlášení probíhá následujícím způsobem. Nejprve se porovnají informace předané formulářem s údaji v databázi a v případě shody se přiřadí danému uživateli SID.

3.3.2 Popis jednotlivých odkazů administrátorského menu

Správa oddílu

Zde má správce systému k dispozici podrobný seznam všech registrovaných členů a má možnost úpravy osobních údajů (Obr. 9) či jejich mazání. Další možností je vložení nového člena (Obr. 10), družinky, nebo úprava stávajících. Pokud dojde ke změně názvu družinky, projeví se tato změna i v přehledu členů, tudíž název družinky do které člen patří, je neustále aktuální.

SPRÁVA ODDÍLU

 Vložit nového člena
  Vložit nebo upravit družinku

Příjmení	Jméno	Datum narození	Editace
<input type="checkbox"/> Pohanková	Bára	12.6.1992	Upravit
<input type="checkbox"/> Machacek	Miroslav	5.12.1983	Upravit
<input type="checkbox"/> Cuták	Tomáš	5.8.1991	Upravit
<input type="checkbox"/> Vašinová	Marie	20.9.1986	Upravit
<input type="checkbox"/> Pohanka	Vojta	6.7.1993	Upravit
<input type="checkbox"/> Šlahůnková	Hana	6.6.1990	Upravit
<input type="checkbox"/> Vašina	Zbyněk	10.5.1982	Upravit
<input type="checkbox"/> Šlahůnek	Jan	25.12.1983	Upravit

Celkově registrováno členů: 8

vybrané členy

Obr. 9. Správa oddílu

VLOŽIT NEBO UPRAVIT ČLENA

Přezdívka:

Jméno:

Příjmení:

Datum narození:

Pohlaví: muž žena

Registrován od:

Družinka:

Vložena fotka: Ano Ne

Obr. 10. Vložení nového člena

Správa akcí

Správu akcí tvoří podobně jako u správy členů jejich seznam, kde je u každé akce možnost jejího odstranění nebo úpravy. Nachází se zde také volba vložení nové akce (Obr. 11). U jednotlivých akcí je nutno vyplnit datum jejich konání, název, stručný popis a jméno organizátora.

AKCE - VLOŽENÍ, EDITACE

Datum konání:	<input type="text" value="29.7.2006"/>
Název akce:	<input type="text" value="Výlet do přírody pěkně po svých :o)"/>
Popis akce:	<input type="text" value="Krátká výprava do přírody. Půjdeme se podívat do nedalekého lesa, kde proběhnou různé dovednostní soutěže. Hlavně si vezměte dobrou obuv a sportovní oblečení.
Zájemci hlase se u Zbyňka."/>
Organizátor:	<input type="text" value="Zbyněk a Honza"/>

Obr. 11. Vložení nebo editace akce

Správa fotek

Fotogalerie na stránkách je vytvořena tak, že v databázi je uložen její název, jemuž je přiděleno unikátní číslo. Toto je číslo je následně pomocí PHP předáváno v proměnné jako součást cesty k adresáři s fotkami, který je umístěn na serveru. Administrátor pomocí správy fotek vytváří jednotlivé fotogalerie, kterým přiřadí libovolný název. Důležité je, aby fotografie umístěné na serveru byly v adresáři, jehož název je tvořen číslem stejným, jako je unikátní číslo u konkrétní fotogalerie. V opačném případě by mohlo dojít k situaci, že by fotografie neodpovídaly fotogalerii, ke které jsou přiřazeny.

Zpráva e-mailem

Zde má administrátor možnost rozesílat informace a novinky uživatelům, kteří si tuto službu vyžádali při registraci. Stačí použít jednoduchý formulář (Obr. 12) a zpráva se rozešle všem těmto uživatelům. Po odeslání se zároveň uloží do archivu, kde se dá zpětně přečíst. Administrátor má k dispozici i seznam lidí, kterým bude e-mail doručen.

ZASLAT ZPRÁVU EMAILEM

Předmět:

Zpráva:

Počet uživatelů, kteří odebírají emaily: 6

Pro detailní výpis uživatelů, kteří odebírají emaily klikněte [zde](#)

Pro zobrazení archivu odeslaných emailů klikněte [zde](#)

Obr. 12. Zpráva e-mailem

Uživatelé

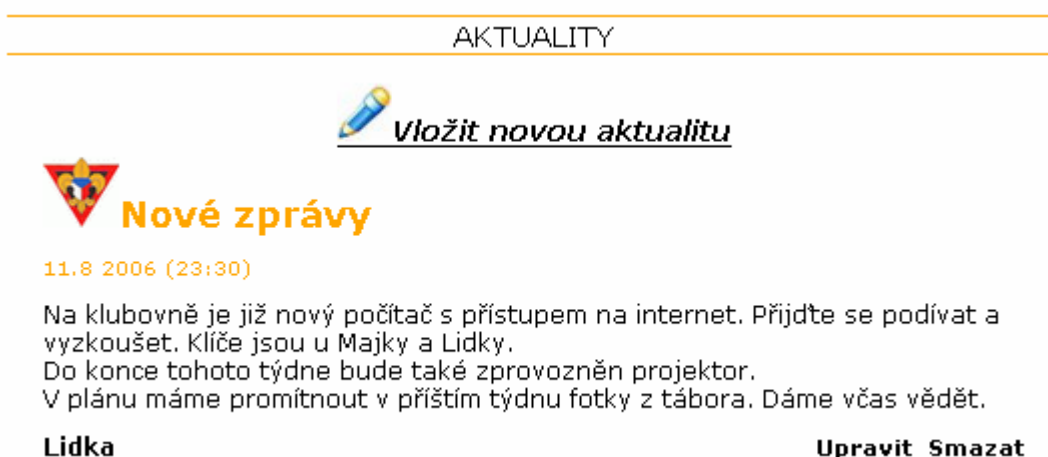
Tuto položku má v administrátorském menu pouze administrátor s oprávněním 1. Najde zde kompletní seznam zaregistrovaných uživatelů se všemi detaily, jako je např. datum registrace, nebo datum a čas posledního přihlášení. Zároveň slouží ke správě uživatelů. Jednotlivé uživatele lze mazat, nebo jim přidělovat oprávnění.

Poslední přístupy

Stejně jako v předchozím případě i tento odkaz v menu najde pouze administrátor s oprávněním 1. Nachází se zde výpis posledních 20 přihlášených uživatelů do portálu a význam je pouze jako informativní funkce.

3.3.3 Další možnosti administrátora systému

Možnosti administrátora se neomezují pouze na položky zahrnuté v administrátorském menu. Po kliknutí na odkaz **Aktuality** se automaticky zobrazí možnost „Vložit novou aktualitu“ přímo nad výpisem aktualit. Administrátorovi s oprávněním 1 se zároveň u každé aktuality zobrazí možnost jejího smazání či úpravy (Obr. 13). Systém aktualit je navržen tím způsobem, že administrátor s omezením (oprávnění 2), nemůže mazat ani upravovat aktuality vložené plnohodnotným administrátorem (oprávnění 1). Při vkládání aktuality se vyplní nadpis a obsah textu. Uživatel má možnost podepsat se pod aktualitou jiným jménem, než pod kterým se přihlašuje.



Obr. 13. Možnosti aktualit

Každý administrátor má též právo zasahovat do dalších kategorií hlavního menu. Při zobrazení stránky **Naše historie** mu navíc přibude položka „Upravit historii“, pomocí níž může přímo upravovat obsah této stránky. Při prohlížení stránky **Návštěvní kniha** se mu u každého příspěvku zobrazí symbol popelnice. Kliknutím na tento symbol bude příspěvek smazán. Je to výhodné z hlediska nevhodných příspěvků a nejrychlejší cesta jak takový příspěvek odstranit. Mazání takového příspěvku přímo v databázi by bylo zdlouhavé a pracné.

3.4 Databáze v praxi

Pro svůj projekt jsem zvolil relační databázi MySQL, která je nejčastěji využívána při tvorbě webových stránek. Pomocí PHP skriptu, který využívá funkci `mysql_connect` se přihlašuji k databázi. Tento skript mám umístěn v souboru *db.php* a má následující podobu:

```
<?
```

```
//promene pro pristup k databazi
```

```
$dbjmeno="jmeno";
```

```
$dbheslo="heslo";
```

```
$nazev_db="nazevdb";
```

```
$server="db1.clevernet.cz";
```

```
$link=mysql_connect($server,$dbjmeno,$dbheslo)
```

```
    or die("Nastala chyba(pripojeni k databazi)");
```

```
$select_db=mysql_select_db($nazev_db,$link)
```

```
    or die("Nastala chyba(vyber databaze)");
```

```
?>
```

Na tento soubor se pak odkazují následujícím příkazem:

```
<?
```

```
require "db.php";
```

```
?>
```

Databáze obsahuje celkem 11 tabulek, z nichž některé jsou propojeny pomocí relací.

3.4.1 Struktura použitých tabulek

Uzivatele

Tato tabulka obsahuje celkem 9 sloupců pro ukládání informací. Jedná se asi o nejdůležitější a základní tabulku celé databáze. Jsou v ní uložena id čísla uživatelů, přezdívky uživatelů, jména a příjmení, hesla, e-mailové adresy, oprávnění daných uživatelů, kod, datum a čas registrace a informaci o odebrání zpráv emailem.

Pristupy

V této tabulce jsou ukládány údaje o přihlášení uživatelů do portálu. Obsahuje id číslo přístupu, id číslo uživatele, přezdívku a datum a čas přihlášení.

Aktuality

Tabulka obsahuje 6 sloupců pro ukládání údajů týkajících se aktualit oddílu. Jedná se o id číslo aktuality, nadpis, obsah textu, jméno autora, id číslo autora a datum a čas vložení aktuality.

Planakce

Jedná se o tabulku do které jsou ukládány informace o plánovaných akcích. Obsahuje 6 sloupců: id číslo akce, datum konání akce, název akce, popis akce, organizátora a autora vložení akce do systému.

Prihlasakce

Tato tabulka slouží pro propojení tabulek Planakce a Uzivatele. Díky ní jsou uživatelé přiřazováni k jednotlivým akcím na které jsou přihlášení. Obsahuje 3 sloupce: id číslo záznamu, id číslo plánované akce a id číslo uživatele.

Clenove

Tabulka sloužící pro ukládání údajů registrovaných členů oddílu, jako například: id číslo člena, přezdívka, jméno, příjmení, pohlaví, datum narození, rok registrace a id číslo družinky. Celkově tedy obsahuje 8 sloupců.

Druzinky

V této tabulce jsou uloženy pouze dva údaje a to id číslo družinky a název družinky. Z této tabulky jsou potom údaje používány u registrovaných členů.

Archivzprav

Do této tabulky se ukládají zprávy odeslané uživatelům emailem. Obsahuje 5 sloupců: id číslo zprávy, předmět, obsah zprávy, autor a datum odeslání.

Historie

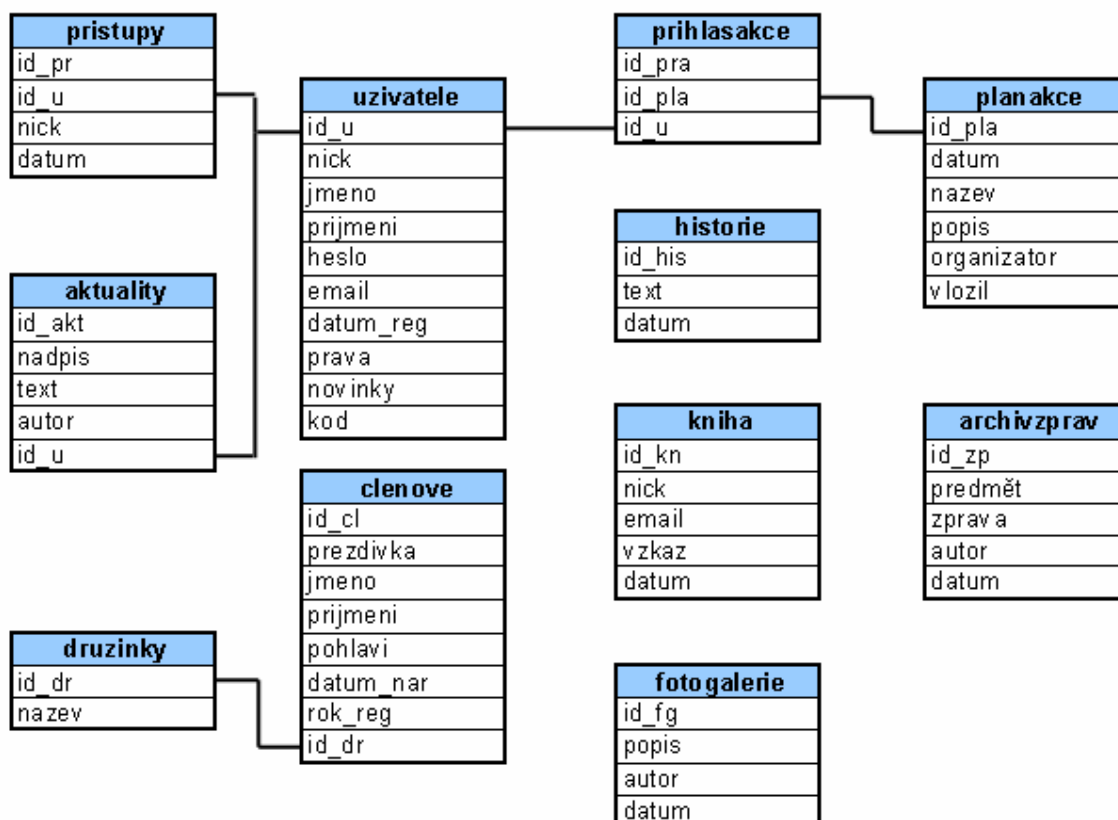
Tabulka ve které je uložena historie oddílu. Obsahuje pouze 3 sloupce: id číslo historie, text historie a datum poslední úpravy.

Fotogalerie

Obsahuje názvy všech fotogalerií a další hodnoty, které jsou využívány pro načítání fotek ze serveru. Obsahuje 4 sloupce: id číslo fotogalerie, popis, autora který fotogalerii vytvořila datum vložení fotogalerie.

Kniha

V tabulce jsou uloženy příspěvky, které návštěvník stránek vyplní do návštěvní knihy. Obsahuje 5 sloupců: id číslo příspěvku, autor, email, vzkaz a datum vložení příspěvku.



Obr. 14. Struktura tabulek databáze

ZÁVĚR

Hlavní náplní mojí práce bylo vytvoření webového portálu Skautského oddílu Domanín. Stěžejními body tohoto projektu jsou teoretická a praktická část.

V teoretické části jsem postupně představil nejčastěji používané technologie pro tvorbu dynamických stránek běžících na straně serveru i klienta. Důraz jsem kladl zejména na technologie XHTML, CSS, PHP a MySQL. Popsal jsem zde výhody a nevýhody jednotlivých technologií, a na základě těchto poznatků jsem zvolil pro tvorbu těchto stránek ty neoptimálnější.

V praktické části jsem se pak zabýval samotnou realizací projektu. Nejprve byla navrhována grafická podoba portálu, poté podoba a rozmístění jednotlivých prvků. V databázovém systému jsem posléze navrhl databázi s potřebnými tabulkami a sloupci. Dynamičnosti stránek jsem dosáhl pomocí jazyka PHP. To mi umožnilo vytvořit systém registrace a přihlašování uživatelů do systému. Uživatelům tato možnost přináší spoustu drobných výhod, mezi které patří zejména neustálý přísun aktuálních informací zasílaných e-mailem. Jednoduchý systém přidělování oprávnění usnadňuje spravování stránek, protože jakmile má uživatel administrátorská práva, nabízí se mu široká paleta nástrojů na údržbu většiny služeb. Mezi ně lze zařadit zejména zasílání zpráv e-mailem, úpravu jednotlivých kategorií portálu, správu aktualit apod.

Uživatelské prostředí jsem navrhnul tak, aby bylo co nejintuitivnější, a to hlavně z hlediska administrace. Pozdějšímu rozšiřování služeb jsem se snažil přizpůsobit i návrh databáze a samotných zdrojových kódů tak, aby nabízely co největší flexibilitu. Další etapou vývoje této aplikace by například mohlo být zasílání e-mailových zpráv mezi jednotlivými uživateli. Možností, jak rozšířit uživatelskou či administrátorskou sekci je víc, a v budoucnu přikročím k jejich realizaci.

V současné době jsou stránky plně funkční a mají za sebou vlnu prvních návštěvníků. Vzhledem k jejich pozitivním ohlasům snad můžu usuzovat, že se mi webový portál Skautského oddílu Domanín povedl. Věřím že bude dobře sloužit všem členům oddílu, vedení i náhodným návštěvníkům a tím plnit svou funkci.

SEZNAM POUŽITÉ LITERATURY

- [1] KOSEK Jiří: *PHP – Tvorba interaktivních internetových prezentací*, Grada Publishing, 1998, ISBN 80-7169-373-1
- [2] Kolektiv autorů: *PHP programujeme profesionálně*, Computer Press,a.s. 2001, ISBN 80-7226-310-2
- [3] *WWW Stránky – Jiří Kosek*. Dostupný z URL: <<http://badame.vse.cz>>
- [4] *Živě.cz*. Dostupný z URL: <<http://www.zive.cz>>
- [5] POWELL, THOMAS A.: *Web Design – Kompletní průvodce*, Computer Press, 2004, ISBN 80-7226-949-6
- [6] *Webtvorba*. Dostupný z URL: <<http://www.webtvorba.cz>>
- [7] *Interval.cz*. Dostupný z URL: <<http://www.interval.cz>>
- [8] *Builder.cz*. Dostupný z URL: <<http://www.builder.cz>>
- [9] *Dobraspolecnost.cz – poskytovatel webhostingových a serverhostingových služeb*. Dostupný z URL: <<http://www.dobraspolecnost.cz>>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

CERN Conseil Européen pour la recherche nucléaire

WWW Wolrd Wide Web

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

URL Uniform Ressource Locator

CGI Common Gateway Interference

SSJS Server Side Java Script

ASP Active Server Page

DOM Document Object Model

PHP Hypertext Preprocessor

DHTML Dynamic HTML

JVM Java Virtual Machine

XML Extensible Markup Language

DTD Document Type Definition

CSS Cascading Style Sheets

DBMS Database Managment System

RDBMS Relační Databázový Systém

SID Session ID

SEZNAM OBRÁZKŮ

Obr. 1. Relace 1:1	22
Obr. 2. Relace 1:N	23
Obr. 3. Relace M:N	23
Obr. 4. Registrační formulář	32
Obr. 5. Chybná registrace – příjmení	33
Obr. 6. Chybná registrace – email	33
Obr. 7. Seznam plánovaných akcí	34
Obr. 8. Detailní zobrazení akce	34
Obr. 9. Správa oddílů	36
Obr. 10. Vložení nového člena	36
Obr. 11. Vložení nebo editace akce	37
Obr. 12. Zpráva e-mailem	38
Obr. 13. Možnosti aktualit	39
Obr. 14. Struktura tabulek databáze	42

SEZNAM TABULEK

Tabulka I. Dělení jazyků.....	10
Tabulka II. BDSM	21
Tabulka III. Přiřazení kategorií práv.....	35

SEZNAM PŘÍLOH

- P1 MySQL příkazy pro tvorbu databáze
- P2 Datové typy sloupců databáze MySQL
- P3 Seznam superglobálních proměnných
- P4 Poděkování a zhodnocení prezentace

P1: MySQL příkazy pro tvorbu databáze

```
CREATE TABLE `aktuality` (  
  `id_akt` int(10) unsigned NOT NULL auto_increment,  
  `nadpis` varchar(140) NOT NULL default "",  
  `text` text NOT NULL,  
  `autor` varchar(40) NOT NULL default "",  
  `id_aut` int(10) unsigned NOT NULL default '0',  
  `datum` varchar(20) NOT NULL default "",  
  PRIMARY KEY (`id_akt`)  
) TYPE=MyISAM ;
```

```
CREATE TABLE `archivzprav` (  
  `id_zp` int(10) unsigned NOT NULL auto_increment,  
  `predmet` varchar(100) NOT NULL default "",  
  `zprava` text NOT NULL,  
  `autor` varchar(40) NOT NULL default "",  
  `datum` varchar(20) NOT NULL default "",  
  PRIMARY KEY (`id_zp`)  
) TYPE=MyISAM ;
```

```
CREATE TABLE `druzinky` (  
  `id_dr` int(10) unsigned NOT NULL auto_increment,  
  `nazev` varchar(60) NOT NULL default "",  
  PRIMARY KEY (`id_dr`)  
) TYPE=MyISAM ;
```



```
CREATE TABLE `clenove` (  
  `id_cl` int(10) unsigned NOT NULL auto_increment,  
  `prezdivka` varchar(40) NOT NULL default "",  
  `jmeno` varchar(40) NOT NULL default "",  
  `prijmeni` varchar(100) NOT NULL default "",  
  `pohlavi` int(1) unsigned NOT NULL default '1',  
  `datum_nar` varchar(20) NOT NULL default "",  
  `rok_reg` int(4) unsigned NOT NULL default '0',  
  `id_dr` int(10) unsigned NOT NULL default '0',  
  PRIMARY KEY (`id_cl`)  
) TYPE=MyISAM ;
```

```
CREATE TABLE `fotogalerie` (  
  `id_fg` int(10) unsigned NOT NULL auto_increment,  
  `popis` varchar(120) NOT NULL default "",  
  `autor` varchar(20) NOT NULL default "",  
  `datum` varchar(20) NOT NULL default "",  
  PRIMARY KEY (`id_fg`)  
) TYPE=MyISAM PACK_KEYS=0 ;
```

```
CREATE TABLE `historie` (  
  `id_his` int(10) unsigned NOT NULL auto_increment,  
  `text` text NOT NULL,  
  `datum` varchar(20) NOT NULL default "",  
  PRIMARY KEY (`id_his`)  
) TYPE=MyISAM ;
```

```
CREATE TABLE `kniha` (  
  `id_kn` int(10) unsigned NOT NULL auto_increment,  
  `nick` varchar(40) NOT NULL default "",  
  `email` varchar(60) NOT NULL default "",  
  `vzkaz` text NOT NULL,  
  `datum` varchar(20) NOT NULL default "",  
  PRIMARY KEY (`id_kn`)  
) TYPE=MyISAM ;
```

```
CREATE TABLE `planakce` (  
  `id_pla` int(10) unsigned NOT NULL auto_increment,  
  `datum` varchar(40) NOT NULL default "",  
  `nazev` varchar(120) NOT NULL default "",  
  `popis` text NOT NULL,  
  `organizator` varchar(40) NOT NULL default "",  
  `vlozil` varchar(40) NOT NULL default "",  
  PRIMARY KEY (`id_pla`)  
) TYPE=MyISAM ;
```

```
CREATE TABLE `prihlasakce` (  
  `id_pra` int(10) unsigned NOT NULL auto_increment,  
  `id_pla` int(10) unsigned NOT NULL default '0',  
  `id_u` int(10) unsigned NOT NULL default '0',  
  PRIMARY KEY (`id_pra`)  
) TYPE=MyISAM ;
```

```
CREATE TABLE `pristupy` (  
  `id_pr` int(10) unsigned NOT NULL auto_increment,  
  `id_u` int(10) unsigned NOT NULL default '0',  
  `nick` varchar(20) NOT NULL default "",  
  `datum` varchar(20) NOT NULL default "",  
  PRIMARY KEY (`id_pr`)  
) TYPE=MyISAM ;
```

```
CREATE TABLE `uzivatele` (  
  `id_u` int(10) unsigned NOT NULL auto_increment,  
  `nick` varchar(20) NOT NULL default "",  
  `jmeno` varchar(40) NOT NULL default "",  
  `prijmeni` varchar(40) NOT NULL default "",  
  `heslo` varchar(20) NOT NULL default "",  
  `email` varchar(60) NOT NULL default "",  
  `datum_reg` varchar(20) NOT NULL default "",  
  `prava` int(10) unsigned NOT NULL default '3',  
  `novinky` int(10) unsigned NOT NULL default '0',  
  PRIMARY KEY (`id_u`)  
) TYPE=MyISAM ;
```

P2: Datové typy sloupců databáze MySQL

Celočíselné datové typy

Typ	Rozsah	Obsadí bajtů	Popis
TINYINT	-128 až 127 nebo 0 až 255	1	Velmi malá celá čísla
BIT	Synonymum pro TINYINT		
BOOL	Synonymum pro TINYINT		
SMALLINT	-32768 až 32767 nebo 0 až 65535	2	Malá celá čísla
MEDIUMINT	-8388608 až 8388607 nebo 0 až 16777210	3	Středně velká celá čísla
INT	-2^{31} až $2^{31}-1$ nebo 0 až $2^{32}-1$	4	Obvyklá celá čísla
INTEGER	Synonymum pro INT		
BIGINT	-2^{63} až $2^{63}-1$ nebo 0 až $2^{64}-1$	8	Velká celá čísla

Datové typy pro čísla s pohyblivou řádovou čárkou

Typ	Rozsah	Obsadí	Popis
FLOAT	$\pm 1,175494351E-38$ $\pm 3,402823466E+38$	4	Číslo s pohyblivou čárkou v jednoduché přesnosti. Odpovídá FLOAT(4).
DOUBLE	$\pm 1,7976931348623157E+308$ $\pm 2,2250738585072014E-308$	8	Číslo pohyblivou čárkou ve dvojnásobné

			přesnosti. Odpovídá FLOAT(8)
REAL	Synonymum pro DOUBLE		
DECIMAL	Různý	M+2	Číslo v pohyblivé řádové čárce uložené jako char. Rozsah závisí na zobrazované velikosti
NUMERIC	Synonymum pro DECIMAL		
DEC	Synonymum pro DECIMAL		
FIXED	Synonymum pro DECIMAL		

Datové typy pro datum a čas

Typ	Rozsah	Popis
DATE	1000-01-01	Datum. Zobrazí se ve formátu RRRR-MM-DD.
TIME	-838:59:59 838:59:59	Čas. Zobrazí se ve formátu hh:mm:ss.
DATETIME	1000-01-01 00:00:00	Datum a čas. Zobrazí se ve formátu RRRR-MM-DD hh:mm:ss
TIMESTAMP(M)	1970-01-01 00:00:00 Někdy v roce 2037	Časová známka – vhodná pro „cejkování“ transakcí. Zobrazovací formát závisí na hodnotě M.

		Viz. Následující tabulka
YEAR	70-69 (1970-2069) 1901-2155	Rok. Můžete určit, zda se vypíše na dvě nebo čtyři místa. Každá verze má vlastní rozsah.

Různé typy zobrazení datového typu `TIMESTAMP`

Typ	Zobrazí
TIMESTAMP	RRRRMMDDhhmmss
TIMESTAMP(14)	RRRRMMDDhhmmss
TIMESTAMP(12)	RRMMDDhhmmss
TIMESTAMP(10)	RRMMDDhhmm
TIMESTAMP(8)	RRRRMMDD
TIMESTAMP(6)	RRMMDD
TIMESTAMP(4)	RRMM
TIMESTAMP(2)	RR

Datové typy reprezentující prostý textový řetězec

Typ	Rozsah	Popis
CHAR	0 až 255 znaků	Textový řetězec o pevné délce
VARCHAR	1 až 255 znaků	Textový řetězec s proměnnou délkou znaků

Datové typy TEXT a BLOB

Typ	Znaky	Popis
TINYBLOB	255	Malé pole typu BLOB
TINYTEXT	255	Malé pole typu TEXT
BLOB	65 535	Pole BLOB normální velikosti
TEXT	65 535	Pole TEXT normální velikosti
MEDIUMBLOB	16 777215	Středně velké pole typu BLOB
MEDIUMTEXT	16 777215	Středně velké pole typu TEXT
LOB	4 294 967 295	Velké pole typu BLOB
LONGTEXT	4 294 967 295	Velké pole typu TEXT

Datové typy ENUM a SET

Typ	Max. hodnot v množině	Popis
ENUM	65 535	Sloupce tohoto typu mohou obsahovat pouze jednu z uvedených hodnot nebo NULL
SET	64	Sloupce tohoto typu mohou obsahovat několik z uvedených hodnot nebo NULL.

P3: Seznam superglobálních proměnných

Název	Popis
\$_SERVER	Proměnná, kde jsou uloženy důležité informace související s webem a webovým serverem.
\$_ENV	Proměnná, kde jsou uloženy informace o prostředí na serveru.
\$_COOKIE	Proměnná, pomocí které můžeme přistupovat ke cookies.
\$_GET	Obsahuje informace předávané pomocí URL.
\$_POST	Obsahuje informace odeslané pomocí formuláře.
\$_FILES	Obsahuje informace o uploadovaných souborech.
\$_REQUEST	Obsahuje informace odeslané pomocí formuláře – (\$_POST i \$_GET).
\$_SESSION	Obsahuje jedinečné informace o klientech.

P4: Poděkování a zhodnocení prezentace

Zhodnocení tvorby stránek 6. oddílu YMCA skautů z Domanína

Jsem velmi spokojena se zpracováním webových stránek. Líbí se mi jak po stránce grafické, tak praktické.

Stránky jsou vytvořeny přehledně a nabízí časté aktualizace díky možnosti registrace členů, kteří se mohou nejen přihlašovat na oddílové akce ale také vkládat příspěvky. Vedoucí mohou vkládat nové členy svých družin. Také fotogalerie a otevírání fotografií v ní uložené je přehledné.

Stránky jsou neustále ve vývoji a věřím, že Jan Šlahůnek, který je vytvořil, bude neustále zvyšovat jejich úroveň, čímž přispěje také k lepší prezentaci našeho skautského oddílu.

.....

Marie Pokorná

vedoucí 6. oddílu YMCA skautů z Domanína

V Domaníně dne 30.8.2006