

Webová prezentace mykologického kroužku v Nivnici

**Web presentations mycological club
in Nivnice**

Marek Ulman

Bakalářská práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2012/2013

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Marek ULMAN**
Osobní číslo: **A09006**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **prezenční**

Téma práce: **Webová prezentace mykologického kroužku v Nivnici**

Zásady pro vypracování:

1. Vypracujte přehledovou studii o webových prezentacích mykologických klubů a kroužků v ČR jak z hlediska obsahu, tak i použitých technologií.
2. Seznamte se s požadavky členů Mykologického klubu v Nivnici na formu a obsah jeho webové prezentace.
3. Popište prostředí a technologie, které při tvorbě stránek použijete.
4. Na základě požadavků vytvořte webovou prezentaci a podrobně popište práci s ní, především možnosti ditace. Věnujte pozornost zabezpečení aplikace.
5. Zhodnoťte možnosti budoucího rozšíření a vylepšení vytvořené prezentace.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. ZEMEK, Lukáš. *Bezpečnost webových aplikací*. Praha, 2012. Bakalářská práce (Bc.). Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky. Vedoucí práce doc. Ing. Martin Sysel, Ph.D.
2. PROKOP, Marek. *CSS kaskádové styly pro webdesignéry*. 2. vyd. Praha: COMPUTER PRESS, 2005. ISBN 80-86593-35-5.
3. DLOUHÝ, Radek. *PHP v příkladech*. Vyd. 1. Kralice na Hané: Computer Media, 2007. ISBN 80-86686-83-3.
4. GUTMANS, Andi, Derick RETHANS a Stig SAETHER BAKKEN. *Mistrovství v PHP 5*. 2. vyd. Praha: COMPUTER PRESS, 2007. ISBN 8025115194.
5. NARAMORE, Elizabeth, Jason GERNER, Yann Le SCOUARNEC, Timothy BORONCZYK. *PHP 6, MySQL, Apache: Vytváříme webové aplikace*. 2. vyd. Praha: COMPUTER PRESS, 2011. ISBN 978-80-251-27.
6. LACKO, Luboslav. *PHP 5 a MySQL 5: Hotová řešení*. 1. vyd. Brno: COMPUTER PRESS, 2007. ISBN 978-80-251-1695-1.

Vedoucí bakalářské práce:

Ing. Libor Pekař

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

24. února 2013

Termín odevzdání bakalářské práce:

14. června 2013

Ve Zlíně dne 24. února 2013

prof. Ing. Vladimír Vašek, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

ABSTRAKT

Hlavním cílem práce je navrhnout a následně vytvořit webovou prezentaci mykologického kroužku dle průzkumu již existujících webových prezentací na internetu a požadavků členů kroužku. V teoretické části se práce zaměřuje na podrobný popis použitých technologií pro návrh prezentace.

Praktická část se zabývá jednotlivými kroky návrhu webové prezentace od grafického vzhledu stránek přes její zabezpečení proti útoku až po umístění na web. V závěru jsou uvedeny možnosti budoucího rozšíření prezentace.

Klíčová slova: Webová prezentace, PHP, MySQL, JavaScript, HTML, CSS

ABSTRACT

The main goal of this thesis is to design and then create a web presentation of a chosen mycological hobby group according to the survey of existing websites and according to the group members' requirements. In the theoretical part the thesis focuses on a detailed description of the technologies used to design the presentation.

The practical part deals with the various steps in the graphic design of a web presentation, security against external attack, and placing the presentation on a web site. Conclusion is focused on a future possible extension of the presentation.

Keywords: Web presentation, PHP, MySQL, JavaScript, HTML, CSS

Děkuji vedoucímu mé bakalářské práce, panu Ing. Liborovi Pekařovi, za cenné rady a čas, který mně během psaní mé práce ochotně poskytl. Dále bych chtěl poděkovat mé rodině, která mně poskytla podmínky pro psaní bakalářské práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné

Ve Zlíně, 10.6.2013

podpis diplomanta

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 PROGRAMOVACÍ JAZYKY	12
1.1 HTML.....	12
1.1.1 Klady a zápory HTML	12
1.2 PHP.....	13
1.2.1 Úvod do PHP.....	13
1.2.2 Programování a začlenění PHP do kódu.....	13
1.2.3 Nejpoužívanější operátory	14
1.2.4 Řídící struktury.....	16
1.2.5 Zpracování formulářů.....	18
1.2.5.1 Metoda GET	18
1.2.5.2 Metoda POST	18
1.2.6 SESSION.....	18
1.2.7 Výhody a nevýhody PHP	19
1.3 CSS.....	19
1.3.1 Zápis CSS.....	19
1.3.1.1 prvek <link>.....	20
1.3.1.2 prvek <style>	20
1.3.1.3 Atribut Style.....	20
1.3.2 Optimalizace pro různé prohlížeče.....	21
1.3.3 Klady a zápory	21
1.4 SQL	21
1.4.1 Databáze	21
1.4.2 MySQL.....	22
1.4.3 Spojování více tabulek	22
1.4.4 Získávání údajů z MySQL databáze	22
1.4.5 Hlavní příkazy pro tvorbu databáze v MySQL	23
1.4.6 Některé příkazy pro tvorbu dotazu v SQL	23
1.4.7 Výhody a nevýhody MySQL	23
1.4.8 Možná náhrady MySQL.....	24
1.5 JAVASCRIPT	24
1.6 JQUERY.....	25
1.6.1 Výhody JQuery	25
2 WEBHOSTING	26
2.1 DOMÉNY A JEJICH TYPY.....	26
2.2 POŽADAVKY NA WEBHOSTING	26
2.2.1 prostor pro ukládání dat	26
2.2.2 Podpora webových služeb.....	27
2.2.3 Podpora vlastních E-mailů	27
2.2.4 Nahrávání obsahu přes FTP	27
2.2.5 Technická podpora	27
2.2.6 Podpora serverového nastavení .htaccess	27
2.2.7 Ostatní požadavky webhostingu	27

3	PROSTŘEDÍ PRO VÝVOJ	28
3.1	PSPAD EDITOR	28
3.2	WAMP SERVER.....	29
3.3	ADOBE PHOTOSHOP CS5.....	29
3.3.1	Výhody a nevýhody	29
3.4	FCKEDITOR	30
II	PRAKTICKÁ ČÁST	32
4	ANALÝZA	33
4.1	PŘEHLED NĚKTERÝCH EXISTUJÍCÍCH PREZENTACÍ MYKOLOGICKÝCH KROUŽKŮ	33
4.1.1	Mykologický kroužek Plzeň	33
4.1.1.1	Nevýhody použité technologie	33
4.1.2	Mykologický klub Brno	34
4.1.2.1	Výhody.....	34
4.1.2.2	Nevýhody.....	34
4.1.3	Další Mykologické kroužky	35
4.2	POŽADAVKY ČLENŮ KROUŽKU NA FUNKCI WEBOVÉ PREZENTACE.....	35
4.2.1	administrace stránek	35
4.2.2	Uživatelská část.....	36
4.3	ZPŮSOB REALIZACE A ZVOLENÁ TECHNOLOGIE	36
5	NÁVRH WEBOVÉ PREZENTACE	37
5.1	REALIZACE DATABÁZE	37
5.1.1	Tabulky	37
	Tabulka eventcalendar	37
	Tabulka photos.....	37
	Tabulka kategorie	38
	Tabulka tbl_admins.....	38
	Tabulka tbl_nav	39
	Tabulka tbl_pages	39
5.2	STRUKTURA ADRESÁŘŮ.....	40
5.2.1	Propojení s databází	40
5.3	GRAFICKÝ DESIGN STRÁNEK	40
5.3.1	Úvodní obrazovka	41
5.3.2	Uživatelská sekce	41
5.3.3	Administrační sekce	42
5.4	SESTAVENÍ SLOŽITĚJŠÍCH DOTAZŮ DO DATABÁZE V ADMINISTRAČNÍ SEKCI.....	43
5.4.1	Seznam stránek.....	43
5.4.2	Vyhledávání v seznamu stránek	43
5.5	ZAČLENĚNÍ FCKEDITORU	43
5.6	KALENDÁŘ UDÁLOSTÍ	44
5.7	OVLÁDÁNÍ A FUNKCE STRÁNEK.....	45
5.7.1	Možnosti návštěvníka stránek	45
5.7.2	Možnosti a Ovládání administrační části	45
5.7.2.1	Seznam stránek	46
5.7.2.2	Nová stránka	46

5.7.2.3	Navigace	47
5.7.2.4	E - mail a heslo	48
5.7.2.5	Galerie.....	48
5.7.2.6	Události.....	49
5.7.2.7	Odhlásit.....	50
6	ZPROVOZNĚNÍ NA SERVERU	51
6.1	NA LOKÁLNÍ STANICI.....	51
6.2	NA PLACENÉM WEBHOSTINGU	51
7	ZABEZPEČENÍ PREZENTACE.....	52
7.1	ZABEZPEČENÍ HESLA ADMINISTRÁTORA V DATABÁZI.....	52
7.2	OŠETŘENÍ TYPŮ SOUBORŮ	53
7.3	ZABEZPEČENÍ FCKEDITORU	53
7.4	DALŠÍ TYPY ÚTOKŮ	54
7.4.1	SQL Injection	54
7.4.2	Ochrana proti SQL Injection	54
7.4.3	Zabezpečení URL adresy	57
7.4.4	XSS - Cross Site Scripting	58
7.4.4.1	Ošetření.....	58
7.5	ZABEZPEČENÍ KONTAKTNÍHO FORMULÁŘE.....	58
7.6	OMEZENÍ PŘÍSTUPU K SOUBORŮM	59
7.7	NEPOVOLENÝ PŘÍSTUP DO ČÁSTÍ ADMINISTRAČNÍ SEKCE	59
8	MOŽNOSTI BUDOUCÍHO VYLEPŠENÍ PREZENTACE.....	60
	ZÁVĚR	61
	ZÁVĚR V ANGLIČTINĚ.....	62
	SEZNAM POUŽITÉ LITERATURY.....	63
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	65
	SEZNAM OBRÁZKŮ	66
	SEZNAM TABULEK.....	67
	SEZNAM PŘÍLOH.....	68

ÚVOD

Dnes a denně se na internetu prostřednictvím internetových stránek setkáváme s nejrůznějšími informacemi, které jsou snadno a všem dostupné prostřednictvím počítačů, chytrých telefonů a podobně. Proto se různé firmy a veřejné instituce snaží prezentovat prostřednictvím webu. Většina informací na stránkách se uchovává pomocí některých z dostupných databázových systémů.

Důležitým aspektem je i grafická podoba a struktura stránek, kde případný návštěvník okamžitě ví, jak se rychle dostat k požadovaným informacím. K tomuto účelu se na stránkách nachází různá menu tvořena jednotlivými odkazy na jednotlivé stránky například u internetových obchodů je to kategorie nabízených produktů u různých kroužků to můžou být informace o konaných akcích a jeho pravidlech společně s kontaktními informacemi.

Hlavním cílem této práce je vytvořit webovou prezentaci Mykologického kroužku Nivnice, kde u takových prezentací se případní návštěvníci mohou dočíst všechny potřebné informace, sledovat průběh konaných akcí s možností prohlížet fotogalerie, případně pomocí kontaktního formuláře nebo diskusního fóra zaslat svůj dotaz, proto budou stránky následně umístěny na jednom z placených webhostingů, který zaručí jejich bezproblémový chod a stálou dostupnost.

Inspirací pro návrh pro mě byly již existující mykologické kroužky.

I. TEORETICKÁ ČÁST

1 PROGRAMOVACÍ JAZYKY

1.1 HTML

HTML vznikl z anglického názvu Hypertext Markup Language. Je to typ značkovacího jazyka, který se používá pro vytváření webových dokumentů (www stránek), které jsou následně zpřístupněny prostřednictvím WWW (World Wide Web) služby, která přenáší informace a dokumenty v rámci Internetu. Od svého vzniku v roce 1991 se jazyk HTML postupně vyvíjel a zlepšoval a to především díky organizaci W3C (World Wide Web Consortium) kvůli které se postupně stal standardem až do jeho poslední oficiální specifikace HTML 5 u které vývoj skončil. [7] a [8]

Samotný dokument HTML se skládá z příkazů, kterým se taky jinak říká tagy. Samotný zápis tagů probíhá ve špičatých závorkách < > a dělí se na tagy párové a nepárové. U párových tagů se musí koncový tag uzavřít lomítkem. Párový tag je tedy například <TITLE> název stránky </TITLE>, kde tento tag slouží pro vložení názvu dokumentu HTML. Pro horizontální čáru použijeme nepárový tag <HR>.[7]

Hlavní kostra HTML stránky:

```
<html>
    <head>
        <title>první stránka</title>
    </head>
    <body>
        vlastní obsah.
    </body>
</html>
```

1.1.1 Klady a zápory HTML

Klady:

- pro prostý text jednoduchá editace
- jednoduchý na naučení
- podpora na různých platformách [7]

Zápory:

- umožňuje vytvořit jen statické stránky.
- limitované možnosti stylizace (dnes stylizace pomocí CSS) [7]

1.2 PHP

Jazyk PHP vznikl za účelem, aby bylo možné rozšířit statické HTML stránky o dynamické skripty, proto jazyk PHP nazýváme dynamickým programovacím jazykem. Umožňuje tedy dynamické změny obsahu bez nutnosti upravovat ručně zdrojové kódy jak tomu je u statických HTML stránek. Samotná struktura příkazů není pro uživatele viditelná. Uživatel vidí jen strukturu HTML tagů, respektive výsledek vykonaného PHP příkazu. [5] a [11]

1.2.1 Úvod do PHP

Zkratka PHP byla původně nazývána jako Personal Home Page (osobní domovská stránka) a sloužila především při práci s formuláři na www stránkách. Byla zpočátku označována jako PHP/FI, kterou v roce 1995 vyvinul Rasmus Lerdorfa a vycházel přitom z jazyka perl. Soubory vytvořené v PHP mají příponu*.php. Veškeré operace se dějí prostřednictvím webového serveru na který je odeslán požadavek a následně se vyhodnotí a zpracuje samotný příkaz a server už pak jen odešle výsledek příkazu prohlížeči. Proto pro dynamické stránky potřebujeme server na kterém je budeme provozovat. Dnes nejpoužívanější verzí je PHP 5, kdy ve vývoji je už verze PHP 6. [5] a [4]

1.2.2 Programování a začlenění PHP do kódu

Jazyk PHP byl od začátku vyvíjen pro použití pro dynamické webové aplikace a především díky množstvím různých funkcí a jednoduchým začleněním do již rozšířeného jazyka HTML se stal oblíbeným mezi programátory webových aplikací. Aby webový server poznal, že se na stránce nachází PHP skript je nutné soubor uložit s koncovkou.php se kterou je zaručena podpora na všech moderních serverech s podporou PHP. Skripty se vkládají mezi značky <?php "samotný kód" ?>. [6] a [5]

Příklad vložení PHP kódu do HTML tagů.

aktualni_datum.php

```
<html>
<head><title> aktualni_cas </title></head>
<body>
//mezi tagy <h1> </h1> se zapíše funkce , která zjistí
aktuální datum
<h1>aktuální datum je: <?php echoDate("d.m.Y ")?? </h1>
</body>
</html>
```

výstupem bude aktuální datum a čas ve tvaru: 20.05.2013.

z kódu lze vidět i vkládání komentářů a to dvěma způsoby:

- */* pro víceřádkové komentáře */*
- *// pro jednořádkové komentáře*

1.2.3 Nejpoužívanější operátory

Typů operátorů se v PHP nachází celá řada pro je můžeme rozdělit na:

- číselné operátory společně s operátorem pro spojení dvou řetězců,
- operátory přiřazení,
- operátory pro porovnání,
- logické operátory. [4]

Číselné operátory

Pro všechny binární operátory platí, že umí pracovat jen s číselnými operandy. Jiné typu jsou automaticky převedeny na příslušné číselné hodnoty. [4]

Tabulka 1 – Číselné operátory

Operátor	Název	Hodnota
+	Sčítání	$\$x + \y
-	Odcítání	$\$x - \y
*	Násobení	$\$x * \y
/	Dělení	$\$x / \y
%	Modulo	$\$x \% \y

Uložení jednotlivých výsledků můžeme následně ukládat do zvolených proměnných a následně s ním dále pracovat. Pro příklad: \$výsledek=\$a + \$b. [4]

Operátory přiřazení

Slouží pro uložení hodnoty do proměnné například: \$pocet = 20 uloží do proměnné pocet hodnotu 20. Máme několik typů a to výše zmíněný jednoduchý operátor (=) a složené operátory, kde jejich funkce je zřejmá z tabulky. Jedná se v podstatě o kombinaci operátorů, kde první operátor slouží pro stanovení operace, která se vykoná společně s operátorem na pravé straně. [4]

Tabulka 2 – Operátory přiřazení

Složené operátor	Použití	Ekvivalentní zápis
+=	\$a += \$b	\$a = \$a + \$b
-=	\$a -= \$b	\$a = \$a - \$b
*=	\$a *= \$b	\$a = \$a * \$b
/=	\$a /= \$b	\$a = \$a / \$b
%=	\$a %= \$b	\$a = \$a % \$b
.=	\$a .= \$b	\$a = \$a .\$b

kde operátor .= slouží pro spojení řetězců například:

```
$b = "Ahoj ";
```

```
$b .= "tady!"; // nastaví $b na "Ahoj tady!", stejně
```

```
jako $b = $b . "tady!";
```

Porovnávací operátory

Operátory porovnání mají za úkol porovnat dvě hodnoty a vrátit výsledek ve tvaru pravda nebo nepravda. Rozhodnutí jsou založena na daném typu operátoru, zda se dvě hodnoty rovnají, zda jedna hodnota je větší a podobně. [4]

Tabulka 3 – Porovnávací operátory

Porovnávací operátor	Název	Použití
==	Rovnost	\$a == \$b
!=	Nerovnost	\$a != \$b
>	Je větší než	\$a > \$b
<	Je menší než	\$a < \$b
>=	Větší nebo roven	\$a >= \$b
<=	Menší nebo roven	\$a <= \$b

Logické operátory

Nejprve jsou hodnoty převedeny na logickou 1 nebo logickou 0 a následně sou porovnány jedním z operátorů, kde jako výsledek je vrácena hodnota pravda nebo nepravda. [4]

Tabulka 4 – Logické operátory

Operátor	Název	Hodnota
&&, and	Logický součin (konjunkce)	Obě hodnoty musí být pravda
, or	Logický součet – (disjunkce)	Alespoň jeden ze dvou je pravda
xor	Výlučný logický součet	Pravdivý jeden nebo druhý

1.2.4 Řídící struktury

Jako i ostatní programovací jazyky tak i PHP obsahuje běžné řídicí struktury, které můžeme rozdělit na podmínkové a cyklické. Podmínkové řídicí struktury testují podmínku uvedenou v závorce a podle toho jestli je podmínka splněna pokračují dál v bloku programu nebo se zastaví. Mezi nepoužívanější patří určitě bezesporu podmínka If.

Cyklické struktury fungují tak, že se podmínka testuje tak dlouho, dokud není splněna. V PHP můžeme použít 4 cykly a to cyklus while, do while, for a foreach. [4] a [3]

Podmínka if – else

Podmínka if - else funguje tak, že pokud je podmínka uvedená v kulatých závorkách pravdivá, tak se vykoná první blok příkazů ve složených závorkách v opačném příkaze se vykoná druhý blok else. Můžeme použít i příkaz elseif pro další dotazování v průběhu kódu. [4]

Podmínkový příkaz Switch

Konstrukce příkazu switch zjednodušuje několik dotazů pomocí elseif. Vyhodnocuje se podmínka za klíčovým slovem switch a porovnává se s hodnotou u klíčového slova case. Při nalezení shody se vykoná část příkazu v bloku case a příkazem break se skončí. [4]

Cyklus while

Cyklus while funguje tak, že dokud je splněna podmínka, která se nachází za klíčovým slovem while, tak se provádí jeden nebo více příkazů stále dokola nebo pokud podmínka neplatí neproběhne ani jednou. [3]

Cyklus do – while

Tento cyklus pracuje podobně jako cyklus while jediný rozdíl je v tom, že cyklus proběhne vždy minimálně jednou i když podmínka neplatí.[4]

Cyklus for

Cyklus typu for je svou funkcí nejsložitější a jeho zápis je následující:

```
for (výraz1;výraz2;výraz3) příkaz;
```

příklad cyklu for:

```
for ($i=0;$i<100;$i ++)  
{  
    echo $i;  
}
```

Následující cyklus vypíše čísla od 0 do 99, kdy výraz1 se vyhodnocuje vždy na začátku cyklu, výraz 2 při každém průchodu cyklem a výraz3 na konci cyklu. [6]

Cyklus foreach

Pro procházení polem je tento cyklus nejlepším řešením, díky své jednoduchosti použití. Cyklus postupně prochází napříč celým polem a hodnoty ukládá třeba do nějaké proměnné. [4]

Ukázka cyklu foreach:

```
$prvky_pole = array (a,b,c,d,e,f);  
foreach ($prvky_pole as $retezec)  
{  
    echo retezec je:$retezec \n";  
}
```

Cyklus tedy prochází postupně pole prvků a postupně je vypisuje v proměnné \$retezec. [6]

1.2.5 Zpracování formulářů

Zpracovávat a odesílat data prostřednictvím formulářů pro další zpracování můžeme provádět za pomoci metod GET a POST. [11]

1.2.5.1 Metoda GET

Hodnoty zadané pomocí formuláře s metodou GET dojde k předání do adresy URL, proto je vhodná především pro vyhledávání díky tomu, že metoda GET vytváří stránku s odkazem se kterým můžeme dále pracovat. [11] a [6]

1.2.5.2 Metoda POST

Pomocí dat z formulářů metodou POST jsou data posílána na nějaké vstupy, kterými může být proměnná nebo nějaký příkaz třeba pro odesílání e - mailu, uložení do databáze a podobně. Touto metodou se odeslané data vůbec nezobrazí v URL adrese jak je tomu u metody GET. [11] a [6]

1.2.6 SESSION

Pro přihlášení např. do administrační části aplikace je potřeba nějakým způsobem uchovat po určitou dobu přihlašovací údaje jinak by nebylo možné vstupovat do všech částí administrace nebo procházet například e-shop. Je tudíž potřeba přenášet údaje z daty s formulářů v nějaké proměnné napříč celým webem.

K tomu nám v PHP slouží relace neboli session, která v sobě přenáší například údaje o přihlášeném uživateli zadané přes přihlašovací formulář metodou POST. Pokud chceme přistupovat například do administrační sekce je potřeba zavolat `session_start()`, od této doby je zahájena relace a možnost ukládat nebo měnit data. Pokud chceme ukončit relaci zavoláme funkci `unset($_SESSION["promena"])`, tím bude obsah proměnné odebrán. [6] a [3]

1.2.7 Výhody a nevýhody PHP

Výhody:

- zaměření pro tvorbu stránek,
 - napříč všemi platformami,
 - rychlost,
 - k dispozici zdarma,
 - množství možností,
 - přívětivé prostředí kódu,
 - množství již vytvořených funkcí,
 - stále je vylepšován,
 - implementace pro komunikaci z různými databázovými systémy.
- [4] a [5]

Nevýhody:

- chybí ladící nástroj,
jediná chyba v zápisu kódu způsobí výpis chybové hlášky. [4] a [5]

1.3 CSS

O Oddělení vzhledu od samotné struktury kódu se uvažovalo už v samotném začátku vývoje HTML, kdy po dlouhých diskuzích a vývoji v roce 1996 byla vydána specifikace CSS (Cascading Style sheets). Hlavním rysem CSS je oddělení struktury kódu od rozvržení jednotlivých prvků na stránce. Další důležitou vlastností jsou rozsáhlejší formátovací možnosti oproti formátování pomocí tagů HTML a tím je pak zaručena vyšší přístupnost webových dokumentů z různých zdrojů jako chytré telefony, tablety atd.

[2] a [7]

1.3.1 Zápis CSS

Pro zápis CSS máme 3 nejpoužívanější způsoby, jak můžeme připojit styly.

- pomocí prvku <link>
- pomocí prvku <style>,
- pomocí atributu style u HTML prvků [2]

1.3.1.1 prvek <link>

používá se pro připojení externího souboru se styly. Je to nejpoužívanější způsob jak připojit soubor se stylem. Výhody jsou následující. [2]

- pro formátování celého webu stačí jediný soubor,
- urychlení načítání ostatních dokumentů na stránce,
- oddělení struktury kódu a formátování,
- zjednodušení případných změn ve správě webu. [2] a [7]

příklad zápisu pomocí <link>:

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

1.3.1.2 prvek <style>

Pomocí tohoto prvku můžeme stylový předpis vložit do HTML dokumentu pomocí prvku <style>. Tato možnost zápisu se používá méně často. [2]

příklad zápisu pomocí <style>:

```
<style type="text/css">
h1 {
margin-top: 0;
font_size: 180%;
}
</style>
```

1.3.1.3 Atribut Style

Můžeme přiřadit styly i jednotlivým prvkům HTML.

příklad zápisu pomocí atributu Style:

```
<body>
<h1 style="font-size: 150%; color: #f00">nadpis</h1>
```

1.3.2 Optimalizace pro různé prohlížeče

Z počátku používání kaskádových stylů se objevovaly problémy s ne zcela správným zobrazením v různých prohlížečích a to především u Internet Exploreru do verze 7, kde docházelo ke špatnému zobrazení. U dnešních moderních prohlížečů jakými jsou Google Chrome, Firefox, Opera atd. se už jen občas setkáme se špatným zobrazením z některého z prvků CSS, proto je dobré počítat s možnými problémy a dle toho upravit stylové předpisy. [2]

1.3.3 Klady a zápory

Klady:

- obohacení o mnoho nových možností formátování obsahu
- případná změna v jediném souboru,
- rychlejší načítání ostatních elementů stránek,
- jasné oddělení struktury kódu a formátování,
- jednodušší výsledný kód a přehlednost.[2] a [7]

Zápory:

- podpora v méně rozšířených prohlížečích, kde může nastat problém se zobrazením [2] a [7]

1.4 SQL

SQL (Structured Query Language) patří mezi dotazovací jazyk, který se vyznačuje tím, že pomocí příkazu v jazyku SQL se odesílají dotazy na databázový server např. MySQL, který daný dotaz vyhodnotí a pošle odpovídající výsledek nejčastěji ve formě údajů z nějaké tabulky uložené v databázi. Jazyk vznikl díky spolupráci firem IBM, Oracle a dalších a postupně byl vydán jako standart SQL-2 a dále se vylepšuje. [13] a [11]

1.4.1 Databáze

Pod pojmem databáze si můžeme představit nějaký soubor dat v určité struktuře, která obsahuje jednu nebo několik tabulek, které sou mezi sebou propojeny. Pro získání určitých dat z různých tabulek vytváříme SQL dotazy. [11] a [13]

1.4.2 MySQL

Jde o volně dostupný databázový server, kde jeho nejčastější použití je právě s jazykem SQL a PHP, kde slouží především pro spravování a komunikaci s databází. Uplatnění má především ve webových publikačních portálech, e - shopech apod., kde je kladen důraz na ukládání a zpracování velkého množství dat. Databázový server se chová jako služba, která poskytuje pro oprávněné uživatele a aplikace přístup k datům a jejich spravování. Pro správu databází slouží klientská konzole mysql.exe nebo pohodlnější rozhraní phpMyadmin, které je součástí databázového serveru MySQL. [6] a [11]

1.4.3 Spojování více tabulek

Abychom mohli přistupovat k údajům v rámci několika tabulek, musíme vytvořit SQL dotaz pro jejich propojení jako např. vybrat název fotky z jedné tabulky podle kategorie, která se nachází v jiné tabulce. Pro tyto účely nám slouží klauzule WHERE nebo JOIN. [13]

například:

```
SELECT tbl_pages.pg_id, tbl_pages.pg_link, tbl_nav.nav_name
FROM tbl_admins, tbl_pages, tbl_nav WHERE tbl_pages.pg_nav =
tbl_nav.nav_id;
```

SQL dotaz propojí 3 tabulky.

1.4.4 Získávání údajů z MySQL databáze

Pro přístup k údajům z tabulek, které jsou uloženy v databázi někde na serveru se využívá jazyka PHP v kombinaci s SQL dotazy.[6]

Ukázka skutečného dotazu pomocí kombinace PHP a SQL:

```
$sqlEvent = "select * FROM eventcalendar ";
// z tabulky eventcalendar se vyberou všechny záznamy
$resultEvents = mysql_query($sqlEvent);
// pomocí mysql_query se zpracuje SQL dotaz
echo "<br>";
while ($events = mysql_fetch_array($resultEvents)) {
echo "Aktuální událost pro tento den: <br> <br> ";
echo "Title: ".$events['Title']."<br>";
echo "Detail: ".$events['Detail']."<br>";
```

}

Pomocí cyklu while se vypíší všechny nalezené údaje, které byly vráceny funkcí `mysql_fetch_array` jako pole.

1.4.5 Hlavní příkazy pro tvorbu databáze v MySQL

Pomocí následujících příkazů můžeme vytvářet databáze a tabulky nebo provádět jejich úpravy. [10]

Tabulka 5 – Příkazy pro práci s databází MySQL

Dotaz	Operace
CREATE	Vytvoří novou databázi
DROP	Odstraní databázi
ALTER TABLE	Provádění změn v struktuře tabulky
CREATE TABLE	Vytvoření nové tabulky v databázi
DROP TABLE	Odstranění tabulky z databáze
CREATE VIEW	Vytvoření nového pohledu
DROP VIEW	Odstranění pohledu

1.4.6 Některé příkazy pro tvorbu dotazu v SQL

Tabulka 6 – Příkazy pro tvorbu dotazů MySQL

Dotaz	Operace
DELETE	Vymazání záznamů z dané tabulky
INSERT	Přidání nového záznamu do dané tabulky
SELECT	Výběr záznamů z dané tabulky
ORDER BY	Spojení daných tabulek
WHERE	Spojení nebo omezení údajů z více tabulek

1.4.7 Výhody a nevýhody MySQL

Výhody:

- vysoká stabilita a rychlost
- nejpoužívanější na trhu,
- vysoká uživatelská podpora,
- k dispozici je zdarma,

- rychlost oproti konkurenčním produktům. [10] a [11]

Nevýhody:

- neumožňuje vytvořit složitější programové konstrukce,
- v opravdu náročných a neustále zatížených databázích nemá takový, výkon jako konkurenční produkty. [10] a [6]

1.4.8 Možná náhrady MySQL

Mezi konkurenty patří například PostgreSQL nebo Oracle, ale nejsou rozšířené v tak vysoké míře jako MySQL. [11]

1.5 JavaScript

Patří mezi programovací jazyk, který je oproti PHP vyhodnocen na straně klienta nikoliv na serveru. Skript je tedy zpracován v prohlížeči uživatele. Jeho hlavní vlastností je přidat do stránek interaktivní prvky mezi které patří například různé tlačítka, které po kliknutí zobrazí varovné okno. Tohle se může hodit, když chceme zabránit nechtěnému smazání fotky z databáze. Přidáme proto javascriptem potvrzovací okno s potvrzením smazání.

Struktura jazyka má podobný zápis jako například programovací jazyk java. Aby jsme mohli spustit nějaký javascript musíme ho umístit mezi párový tag `<script>` `</script>` do struktury HTML stránky, tím docílíme toho, že prohlížeč script pozná a vyhodnotí. [8]

příklad: vyskakovací okno v javascriptu:

```
<body>

<script>

    x = prompt("Zadej svoje jméno", "");

    document.write("Tvoje jméno tedy je ");

    document.write(x);

</script>

</body>
```

Tímto skriptem se zobrazí dialogové okno pro zadání řetězce, kdy následně po jeho zadání se vložený řetězec zobrazí.

1.6 JQuery

Je to v podstatě rozšíření javascriptu, kde se pomocí nových funkcí přidávají nové funkce, které za pomoci samotného javascriptu jsou dosti náročné. Hlavní využití je jednoduše vytvářet, manipulovat nebo mazat prvky v dokumentu pomocí interaktivních funkcí jako je změna velikostí formulářů tabulek a jiných prvků za pomoci kurzoru myši. Umožňuje rovněž přidat pro libovolný prvek jakým je třeba tabulka CSS styly, efekty jako zmizení, blikání a podobně. JQuery si získalo velkou oblibu u programátorů webových aplikací díky nepřehlednému množství již existujících řešení, funkcí a pluginů pro práci s obrázky, formuláři a podobně, které jednoduchou úpravou a následným začleněním do kódu stránek ožíví jejich obsah. Použití si najde v náročných webových aplikacích jakými jsou různé informační portály, e - shopy, webové galerie a podobně. [12]

1.6.1 Výhody JQuery

- obrovské množství pluginů mezi kterými si každý najde co potřebuje,
- podrobná dokumentace a množství publikací,
- snad nejlepší implementace práce s CSS styly,
- dostupné zdarma jako open source,
- stále se vyvíjí díky velké podpoře ze strany vývojářů,
- společně s PHP je to mocný nástroj pro tvorbu webových aplikací.

[12]

2 WEBHOSTING

Pro umístění internetových stránek na internet potřebujeme nahrát samotný obsah webu na nějaký server, což je počítač, který díky aplikačnímu rozhraní a tím, že je připojen do sítě internetu umožňuje zpracovávat požadavky od ostatních klientských počítačů. Obsah stránek můžeme umístit případně na nějaký z velkého množství freehostingů, který bývá svými možnostmi značně omezen a nezaručuje nám vždy správnou funkčnost stránek a nemůžeme si volit vlastní název domény. Proto je lepší využít služeb placeného hostingu, který zaručí bezproblémový chod našich stránek se spoustou možností správy a ochrany umístěných stránek. [9] a [6]

2.1 Domény a jejich typy

Pro vytvoření názvu adresy WWW stránek, podle které se k nim bude přistupovat v rámci celého internetu nám slouží doména. Máme 3 různé typy domén. Doména prvního řádu je specifická podle dané země a je to vlastně koncovka např. domena.cz, .sk, .ru a podobně. Vlastní doménu 1.řádu si proto zaregistrovat nemůžeme. Pro vytvoření názvu stránek si můžeme zaregistrovat doménu 2. řádu např. ve tvaru nazevstranky.cz, kde cena se pohybuje kolem 300 Kč a v rámci internetu je nejpoužívanější. Dále existuje ještě doména 3.řádu, která již běží na nějaké již existující doméně 2.řádu a její tvar například je: test.marekulman.cz, která se používá především v rámci freehostingů. Její použití je ale v rámci prezentace např. stránky firmy nevhodné, protože to nevypadá dobře. [9]

2.2 požadavky na webhosting

2.2.1 prostor pro ukládání dat

Pro ukládání velkého množství dat je zapotřebí, aby hosting nabízel dostatečnou kapacitu pro naše data mezi které patří například: databáze, fotky, videa a podobně. Dnešní placené hostingy nabízejí velikost prostoru v rámci desítek až 100 GB, to ovšem neplatí pro freehostingy, kde se kapacita pohybuje kolem desítek MB málo kdy více. [9]

2.2.2 Podpora webových služeb

Pro stránky pracující s databázemi, které pro komunikaci využívají technologie MySQL a PHP je důležitá jejich podpora a to včetně prostředí pro správu databází jakým je například phpMyAdmin na straně serveru na kterém stránky běží.

U placených hostingů je podpora téměř vždy zaručena, kde u freehostingů to není pokaždé samozřejmostí. Proto si při výběru hostingu musíme dát na tuto skutečnost pozor. [9]

2.2.3 Podpora vlastních E-mailů

Pro vlastní stránky je dobré mít i vlastní e-mailové schránky pro přijímání a odesílání e-mailů, kde počet schránek a jejich kapacita se liší od typu hostingu. Některé hostingy nabízí i zálohování e-mailů s ochranou pomocí antivirových programů. [9]

2.2.4 Nahrávání obsahu přes FTP

Pro lepší správu jednotlivých souborů webových stránek je dobré, když hosting disponuje protokolem FTP, kdy nahrávání nebo změna stránek je daleko jednodušší. [9]

2.2.5 Technická podpora

Při výpadku nebo náhlé nefunkčnosti stránek je díky technické podpoře umožněno rychle opravit vzniklý problém a ostatní problémy nebo dotazy ohledně správy webového obsahu. [9]

2.2.6 Podpora serverového nastavení .htaccess

Nastavením tohoto souboru je možné zamezit přístupu do určitých adresářů, vypisovat chybové hlášky, chránit heslem některé části webu a spoustu dalších možností, které závisí na konfiguraci serverů daného hostingu. [9]

2.2.7 Ostatní požadavky webhostingu

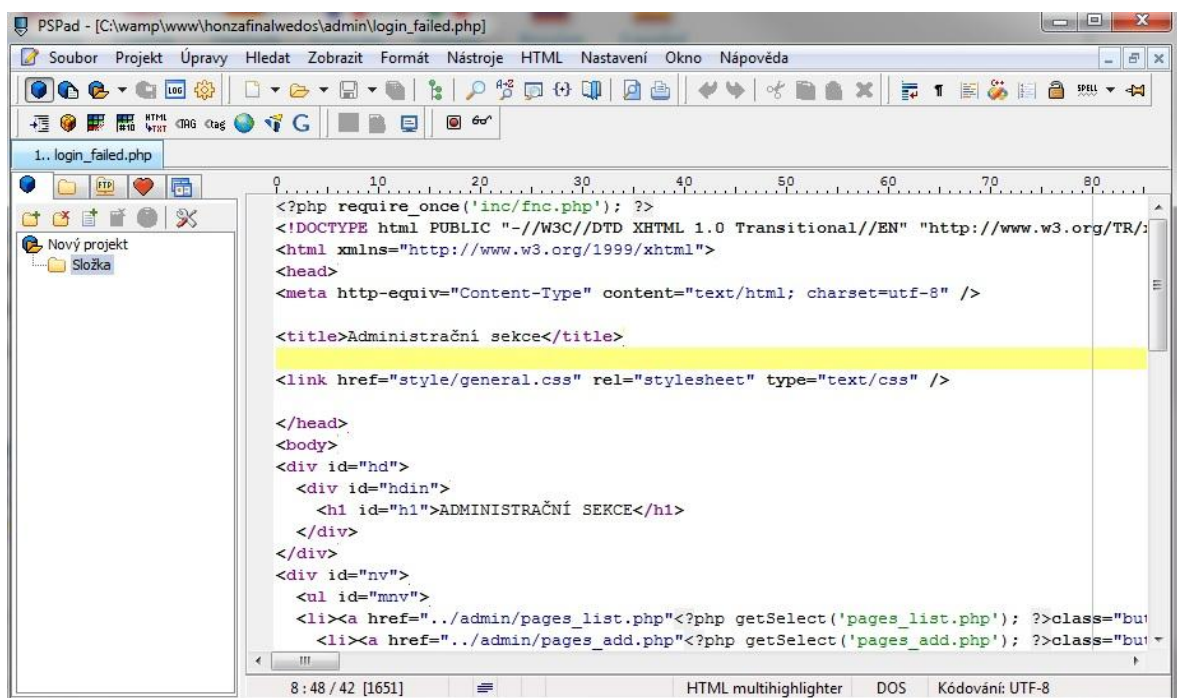
Asi nejdůležitějším parametrem je stálá dostupnost stránek, která se u nejlepších hostingů pohybuje kolem 99,9 % a zajišťuje tedy možnost zobrazit stránky 24 hodin denně bez výpadků. Dalším důležitým parametrem je rychlost a množství přenášených dat, které hosting zvládne. Posledním kritériem je poměr ceny k množství nabízených služeb, kdy vždy neplatí čím dražší hosting, tím je lepší, proto záleží na mnoha aspektech.[9]

3 PROSTŘEDÍ PRO VÝVOJ

Pro vývoj webové prezentace je potřeba zvolit správné vývojové prostředí pro všechny kroky návrhu. Pro psaní vlastního programového kódu bylo zvoleno prostředí programu PSPad, který díky svým možnostem je pro tento účel postačující. Pro tvorbu některých grafických prvků pro web bylo využito programu Adobe Photoshop CS5 a následně pro vytvoření webového serveru na počítači s Windows byl vybrán program WampServer.

3.1 PSPad editor

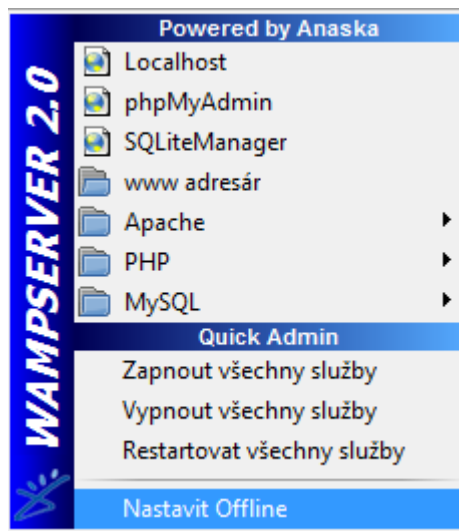
Slouží jako univerzální textový editor, který pomáhá při vývoji v různých programovacích jazycích jako je např. PHP, HTML, MySQL, apod. Jeho velkou výhodou je zvýraznění a rozlišení různých typů programovacích jazyků pomocí různých barev a zpřehlednění výsledného kódu. Navíc obsahuje mnoho vyspělých funkcí, které dokážou ušetřit drahocenný čas. Umožňuje práci na několika projektech současně dále pak konverzi mezi různými typy kódování. Pro jazyk HTML umožňuje i náhled stránky. Samozřejmostí je číslování řádků, které značně ušetří orientaci při ladění případných chyb. Program je navíc volně šiřitelný. [15]



Obrázek 1 – Prostředí programu PSPad

3.2 WampServer

WampServer je program, který obsahuje všechny součásti pro správný běh webového serveru v jednom. Mezi jeho přednosti patří jednoduchost použití a správa jednotlivých služeb jako je Apache, MySQL, PHP a jejich nastavení. Je k dispozici pro systém windows. Ovládací panel umožňuje rychlý přístup ke všem jeho možnostem. [16]



Obrázek 2 - Ovládací panel WampServeru

3.3 Adobe Photoshop CS5

Photoshop patří mezi oblíbené profesionální editory pro práci s bitmapovou grafikou, své uplatnění najde především při úpravách a retuši fotografií, tvorbě webového designu přes různé webové prvky až po tvorbu log. Obsahuje širokou paletu užitečných nástrojů pro úpravu obrázků a tvorbu nových grafických prvků. Umožňuje pracovat s vrstvami, kanály, maskami a nabízí mnoho dalších užitečných funkcí. Hlavní výhodou je rychlost práce především díky přebornému množství klávesových zkratk a možnosti vytvoření si vlastních pro daný typ nástroje. [17]

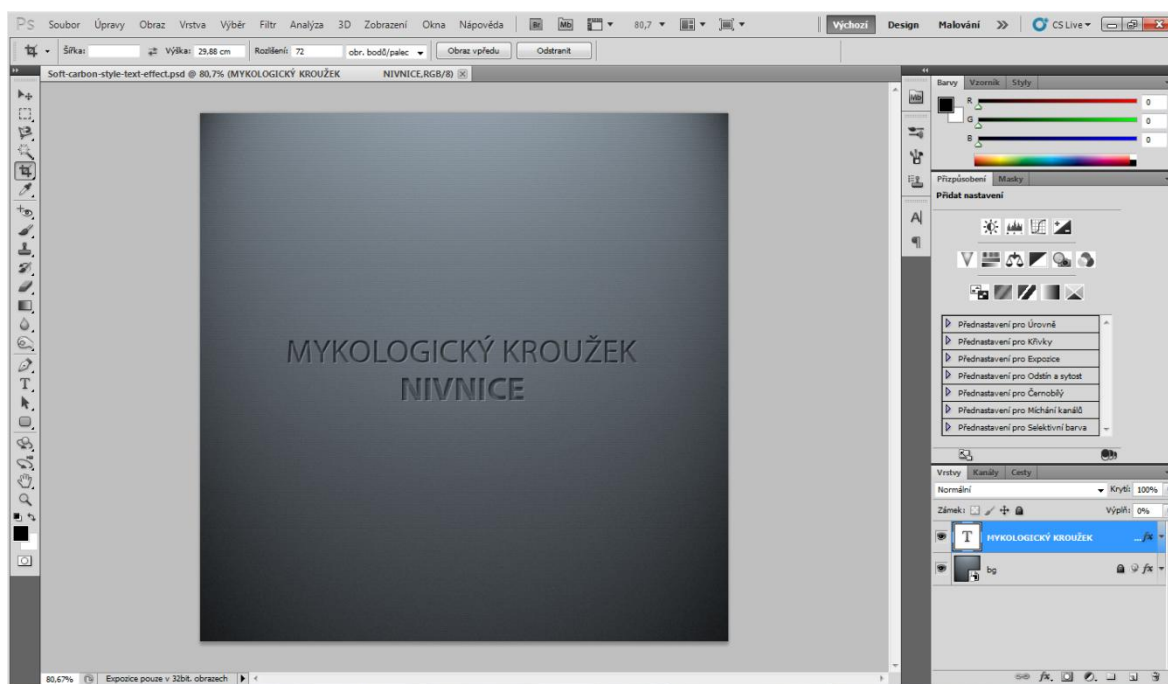
3.3.1 Výhody a nevýhody

Výhody:

- po zvládnutí hlavních funkcí programu je pak radost v něm tvořit,
- podpora napříč operačními systémy,
- podpora široké řady formátů,
- široké množství nástrojů,
- možnost práce ve vrstvách,
- schopnost práce jak s rastrovými, tak vektorovými formáty,
- přívětivé uživatelské prostředí. [17]

Nevýhody:

- není zadarmo, ale je možnost využít 30-denní zkušební verzi
- vysoká cena plné verze. [17]

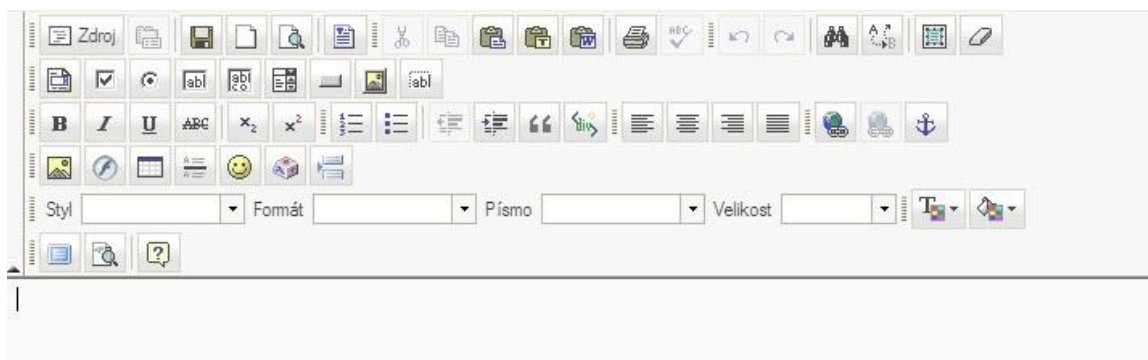


Obrázek 3 – prostředí photoshopu CS5 při tvorbě grafiky

3.4 FCKeditor

Tento volně šiřitelný textový editor, který díky podobnému ovládní jako většina kancelářských aplikací umožňuje jednoduše psát články, vytvářet tabulky, vkládat obrázky a to včetně snadného formátování obsahu. Využívá se proto v mnoha redakčních

systemech. Výhodou je, že se nemusí složitě instalovat a je téměř ihned připraven k začlenění do stránek. Pomocí konfiguračních souborů je možné si editor přizpůsobit podle vlastních požadavků. FCKeditor obsahuje i souborový manažer pro nahrávání různých typů souborů. [14]



Obrázek 4 – Prostředí FCKeditoru

II. PRAKTICKÁ ČÁST

4 ANALÝZA

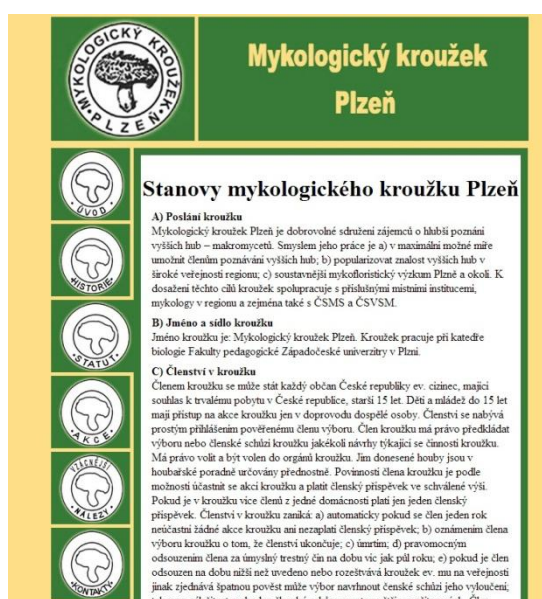
4.1 Přehled některých existujících prezentací Mykologických kroužků

4.1.1 Mykologický kroužek Plzeň

Prezentace kroužku je vytvořena pomocí statických HTML stránek bez administračního prostředí. Pro změnu obsahu je tedy nutné ručně přepsat obsah každé stránky ve zdrojových souborech. Stránka obsahuje menu které odkazuje na jednotlivé stránky, které obsahují základní informace o kroužku jakými jsou jeho historie, uskutečněné akce a stanovy kroužku. Jediným dynamickým prvkem na stránce je v menu kontakty diskusní fórum realizované pomocí volně dostupné knihy návštěv od společnosti BlueBoard.cz, která je formou odkazu ve zdrojovém HTML kódu umístěna na stránky. Proto veškeré skripty a ukládaná data jsou zpracovávány na jejich serverech nikoliv na stránkách kroužku. Rozmístění prvků na stránce jako je menu, obsahová část je realizována pomocí stylů CSS.

4.1.1.1 Nevýhody použité technologie

- minimální změna obsahu
- pro změnu obsahu je nutná editace zdrojových souborů
- zastaralý grafický design
- nemožnost jednoduše přidávat nebo odstraňovat fotky



Obrázek 5 – stránky Mykologického kroužku Plzeň

4.1.2 Mykologický klub Brno

Prezentace je vytvořena prostřednictvím free služby webzdarma.cz, která je již založena na moderních technologiích jakými jsou PHP, MySQL, Javascript. Samotný obsah je ukládán prostřednictvím databáze. Stránky klubu jsou kompletně nastýlovány pomocí CSS stylů, kde výsledkem je pěkná grafická podoba stránek.

Po stránce obsahu se zde nachází základní informace o kroužku, kontakty na vedení klubu a uskutečněné akce včetně pěkně zpracované fotogalerie u již uskutečněných akcí, která je vytvořena za pomoci javascriptu. Dále pak možnost přidat komentář k jednotlivým článkům na stránkách.

4.1.2.1 Výhody

- již vyvinutý a stabilní redakční systém pro změnu obsahu
- moderní technologie
- příjemný design
- možnost vložit různé moduly na stránky jako část pro komentáře, galerie apod.
- zdarma

4.1.2.2 Nevýhody

- pevně dané možnosti úpravy stránky
- není zaručena dostupnost díky umístění na freehostingu
- nemožnost změny webu dle vlastních specifických požadavků



Obrázek 6 – stránky Mykologického klubu Brno

4.1.3 Další Mykologické kroužky

Mezi další kroužky, které sem navštívil je Mykologický klub Hradec Králové, Mykologický kroužek Mělník a Mykologický klub Nezvěstice, které byly opět vytvořeny pomocí některých volně dostupných služeb jako jsou webgarden, estranky a webzdarma, proto fungují na stejných výše popsaných technologiích a mají podobnou strukturu a obsah.

4.2 Požadavky členů kroužku na funkci webové prezentace

Po schůzce s jedním z členů kroužku, který bude rovněž webovou prezentaci spravovat byly stanoveny následující cíle na vzhled a funkci webové prezentace.

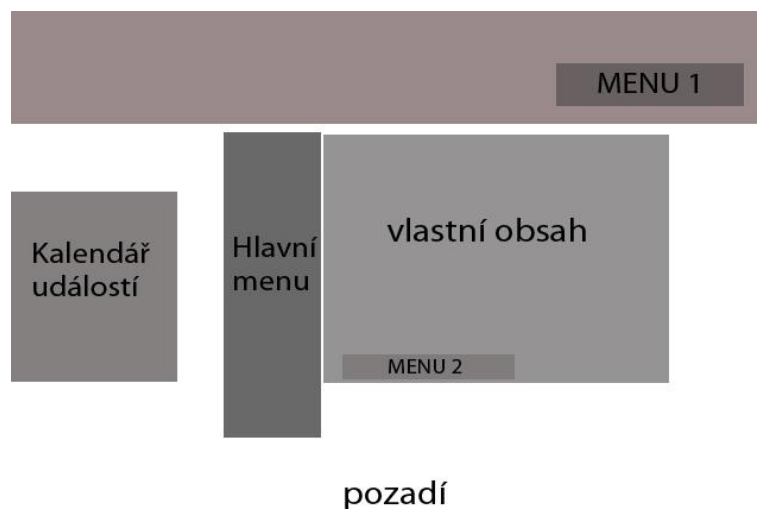
4.2.1 administrace stránek

Administrační sekce by měla mít jednoduché uživatelské rozhraní pro editaci uživatelské sekce stránek a měla by obsahovat následující možnosti:

- **Seznam dostupných stránek** – kde budou zobrazeny všechny existující stránky u kterých bude možná případná editace obsahu a případně úplné odstranění. Dále pak možnost vyhledávání daných stránek.
- **Vytvoření nové stránky** – v další sekci půjde vytvořit úplně nová stránka, kde pro vkládání obsahu bude použito přívětivé prostředí podobné z editorů Microsoft Office.
- **Pořadí stránek** – bude možné měnit pořadí stránek.
- **Sekce pro změnu kontaktního emailu a hesla administrátora** – V této části bude mít administrátor možnost měnit nebo přidávat kontaktní emaily a přístupové heslo do administrační sekce
- **Fotogalerie** – v této sekci půjde vytvářet, mazat jednotlivé alba a nahrávat jednotlivé fotky s možností přidání popisku do příslušných alb ze kterých bude možnost smazat libovolnou fotku a rovněž celé album se všemi fotkami.
- **Kalendář událostí** – jednoduchý kalendář s možností přidávat, editovat nebo mazat naplánované události.
- **zabezpečení aplikace** – zabezpečení proti zneužití citlivých dat.

4.2.2 Uživatelská část

- **Rozložení prvků stránky** – přehledné rozvržení



Obrázek 7 – předpokládané rozložení prvků na stránce

- **Grafický design** – čistý a jednoduchý v odstínech zelené barvy.

4.3 Způsob realizace a zvolená technologie

Pro realizaci byla zvolena databázi řízená aplikace na technologii MySQL a PHP, kde všechny data budou ukládány v databázi a pomocí dotazů jazyka SQL v kombinaci s PHP bude probíhat vzájemná komunikace mezi databázovým serverem a prezentací. Toto řešení je pro vytváření dynamických stránek nejrozšířenější. Pro fotogalerii budou do databáze zaznamenávány jen názvy fotek a názvy jednotlivých alb, kde samotné fotky a jejich miniatury budou ukládány do určených složek na disku. Pro grafický návrh a umístění prvků na stránky bude využito stylů CSS.

5 NÁVRH WEBOVÉ PREZENTACE

Návrh webových stránek se skládá z několika kroků - návrhu databáze, struktury stránek a grafického vzhledu.

5.1 Realizace databáze

5.1.1 Tabulky

Databáze obsahuje 5 tabulek, jejichž funkce jsou následující:

Tabulka eventcalendar

V tabulce se nachází 5 sloupců, kde sloupec ID slouží pro ukládání jedinečného čísla pro vytvořené události. Do sloupce Title se ukládá název události, sloupec Detail slouží pro podrobnější popis, u které se do eventDate ukládá datum konání a do dateAdded datum jejího vytvoření.

Tabulka 7 – Tabulka eventcalendar

Sloupec	Typ
ID	int(11) NOT NULL auto_increment PRIMARY KEY (`ID`)
Title	varchar(65) NOT NULL
Detail	varchar(255) NOT NULL
eventDate	varchar(10) NOT NULL
dateAdded	date NOT NULL

Tabulka photos

V tabulce opět najdeme sloupec ID, který je typu primary_key a má hodnotu auto_increment což zaručuje pokaždé jedinečné číslo dané fotky. Do photo_name se ukládá název fotky, podle kterého se současně uloží fotka i do složek. Soubory jsou tedy ukládány mimo databázi. Do date_added se zaznamená datum nahrání fotky. Uloží se také číslo kategorie pro přesné zařazení a případně stručný popis fotky.

Tabulka 8 – photos

Sloupec	Typ
id	int(30) NOT NULL auto_increment PRIMARY KEY (`id`)
photo_name	varchar(200) NOT NULL
date_added	datetime NOT NULL
popisek	text NOT NULL
kategorie	int(5) NOT NULL

Tabulka kategorie

Tabulka kategorie slouží pro ukládání názvu vytvořených alb pokaždé s jedinečným číslem ID, které je typu PRIMARY KEY a má hodnotu auto_increment.

Tabulka 9– Tabulka kategorie

Sloupec	Typ
id	int(5) NOT NULL auto_increment, PRIMARY KEY (`id`)
nazev	varchar(60) NOT NULL

Tabulka tbl_admins

V tabulce se zaznamenává přihlašovací jméno, heslo a případně e-mail pro přístup do administrační sekce, kde adm_pass ve kterém se nachází heslo je v databázi zašifrováno s využitím funkce SHA-1. Položka adm_email může být prázdná.

Tabulka 10 – Tabulka tbl_admins

Sloupec	Typ
adm_id	int(11) NOT NULL auto_increment PRIMARY KEY (`adm_id`)
adm_username	varchar(15) NOT NULL
adm_pass	text NOT NULL
adm_email	varchar(255) default NULL

Tabulka tbl_nav

V tabulce jsou uloženy názvy možných umístění menu, které je možné vytvořit v uživatelské části.

Tabulka 11 – Tabulka tbl_nav

Sloupec	Typ
nav_id	int(2) NOT NULL auto_increment PRIMARY KEY (`nav_id`)
nav_name	varchar(15) NOT NULL

Tabulka tbl_pages

Tabulka slouží pro ukládání údajů o stránkách vytvořených nebo editovaných v administrační sekci, které jsou následně vytvořeny v uživatelské části stránek. Do pg_link se ukládá odkaz nově vytvořené stránky, který se zobrazí v menu, jehož pozici určuje typ menu, které se ukládá do pg_nav a pro změnu pořadí stránek slouží pg_order. Do pg_title se ukládá název stránky. Samotný obsah stránky je ukládán v pg_cont.

Tabulka 12 – Tabulka tbl_pages

Sloupec	Typ
pg_id	int(11) NOT NULL auto_increment PRIMARY KEY (`pg_id`)
pg_link	varchar(150) default "
pg_title	varchar(255) default NULL
pg_cont	text
pg_nav	int(2) default NULL
pg_order	tinyint(3) default NULL

Všechny tabulky byly vytvořeny v prostředí phpMyadmin databázové aplikace MySQL a následně exportovány do jednoho souboru `database.sql`, který se nachází v příloze.

5.2 Struktura adresářů

Struktura jednotlivých adresářů je důležitá pro snadnou orientaci v množství souborů, které obsahuje. Proto byla zvolena následující struktura.

```
/                # kořen webu
/admin           # soubory administrační části a FCKeditoru
/Connections    # propojení s databází
/inc            # include soubory,pluginy,soubory knihoven
/style          # soubory CSS stylů
/images        # ostatní grafické soubory
/image         # ukládání obrázku z galerie
```

5.2.1 Propojení s databází

Pro propojení s databází jsme vytvořil ve složce /Connections/ nový soubor `coondb.php`, kde jsem nastavil trvalé propojení s databází.

```
<?php
    $hostname_coondb = "localhost";
    $database_coondb = "cms";
    $username_coondb = "root";
    $password_coondb = "password";
    $coondb = mysql_pconnect($hostname_coondb, $username_coondb,
    $password_coondb) or
    trigger_error(mysql_error(),E_USER_ERROR);
?>
```

Byly nastaveny přístupové údaje jakými jsou jméno, heslo a název databáze pro propojení s MySQL databází.

Soubor je potřeba připojit na všech stránkách, které budou přistupovat do databáze příkazem `<?php require_once('Connections/coondb.php'); ?>`

5.3 Grafický design stránek

Pro návrh grafiky webu byl zvolen čistý a jednoduchý design v odstínech zelené barvy s využitím převážně stylů CSS, kde pro snadnou editaci jsou soubory stylů uživatelské a administrační sekce odděleny.

5.3.1 Úvodní obrazovka

Úvodní obrazovka `index.php` je vytvořena pomocí CSS, název kroužku a tlačítko `vstoupit` bylo vytvořeno pomocí programu photoshop. Pro efekt "blikání" názvu kroužku byla vytvořena funkce v jQuery.



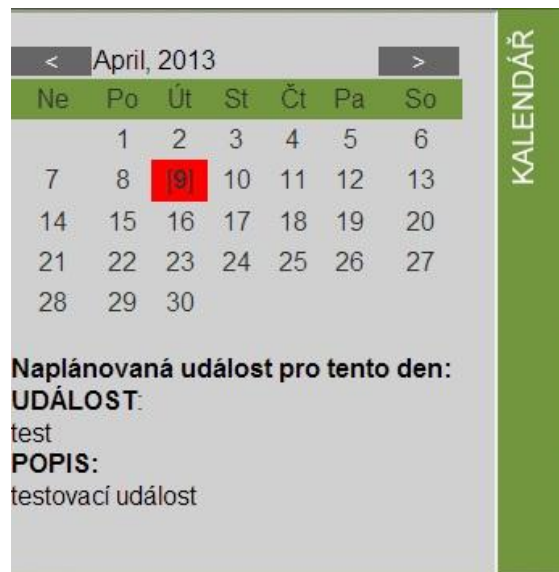
Obrázek 8 – úvodní obrazovka

5.3.2 Uživatelská sekce

Grafická podoba stránek, kterou vidí návštěvník byla napozicována a nastýlována pomocí HTML tagů `<div>` v kombinaci s CSS. Soubor se styly pro uživatelskou část se nachází ve složce `style/general.css`. Soubory stylů uživatelské a administrační části jsou vytvořeny zvlášť. Tlačítko administrace a název kroužku bylo vytvořeno pomocí programu photoshop. Kalendář je také nastýlován pomocí CSS a je vytvořen pomocí plovoucího vnořeného rámu `Iframe`, kde automatické zasouvání a vysouvání zajišťuje vytvořená funkce v JQuery a CSS. Volba měsíců pomocí tlačítek je zajištěna pomocí javascriptové události `onclick`.



Obrázek 9 – uživatelská část prostřednictvím CSS - bez obsahu



Obrázek 10 – kalendář událostí v CSS

5.3.3 Administrační sekce

Rozložení prvků na stránce bylo rovněž napozicováno a nastylováno pomocí HTML tagů `<div>` v kombinaci s CSS. Soubor se styly pro administrační část se nachází v `admin/style/general.css`, pro případnou změnu vzhledu.



Obrázek 11 – grafická podoba hotového administračního prostředí

5.4 Sestavení složitějších dotazů do databáze v administrační sekci

5.4.1 Seznam stránek

```
"SELECT tbl_pages.pg_id, tbl_pages.pg_link, tbl_nav.nav_name
FROM tbl_admins, tbl_pages, tbl_nav WHERE tbl_pages.pg_nav =
tbl_nav.nav_id ORDER BY tbl_nav.nav_id, tbl_pages.pg_link
ASC";
```

Vyberou se všechny dostupné stránky pro právě přihlášeného uživatele.

5.4.2 Vyhledávání v seznamu stránek

```
"SELECT tbl_pages.pg_id, tbl_pages.pg_link, tbl_nav.nav_name
FROM tbl_admins, tbl_pages, tbl_nav WHERE tbl_pages.pg_nav =
tbl_nav.nav_id AND (tbl_pages.pg_link LIKE '%$srch%' OR
tbl_pages.pg_cont LIKE '%$srch%' OR tbl_pages.pg_title LIKE
'%$srch%' ) ORDER BY tbl_nav.nav_id, tbl_pages.pg_link ASC";
```

V dotazu jsou vybrány všechny stránky z databáze a pomocí operátoru LIKE '%\$srch%' se porovná a následně zobrazí požadovaná stránka, jejíž název se zadá do vyhledávacího formuláře.

5.5 Začlenění FCKeditoru

Začlenění FCKeditoru do administrační sekce pro vytvoření nebo editaci stránky provedeme zavoláním instance třídy na určené pozici v HTML kódu.

```
<?php
    $oFCKeditor = new FCKeditor('FCKeditor1');
    $oFCKeditor -> BasePath='fckeditor/';
    $oFCKeditor -> Value='získání obsahu z databáze pomocí datazů';
    $oFCKeditor -> Create();
?>
```

Ještě před tím je nutné v kódu includovat soubory. K tomu slouží funkce **include_once**.

```
<?php
    include_once("fckeditor/fckeditor.php");
?>
```

Od této chvíle je FCKeditor připraven k použití.

5.6 Kalendář událostí

Kalendář je vytvořen pomocí HTML, CSS, Javascriptu a PHP, kde pro ukládání a čtení záznamů se využívá databáze. Kalendář obsahuje 2 verze. Jedna verze se nachází v administračním rozhraní a umožňuje veškeré funkce. Druhá verze je umístěna v uživatelské sekci stránek, kde má jen omezené možnosti.

Pro změnu měsíců pomocí tlačítek je využita javascriptová metoda onclick, která po stisku tlačítka pro změnu měsíce zavolá javascript funkci, která řeší změnu měsíce.

ukázka části funkce pro změnu měsíce směrem dozadu, která je napojena na tlačítko zpět <

```
function goLastMonth(month, year) { // funkce má dva parametry,
    kterými jsou měsíc a rok, které jsou předávány

    if(month == 1) --year;
    month = 13;
    }
    --month
    ...
```

funkce pracuje tak, že se první otestuje zda není první měsíc pokud ano, tak se nastaví měsíc na hodnotu 13, protože rok má 12 měsíců v opačném případě se od aktuálního měsíce odečte jeden měsíc směrem dozadu.

Příklad ukládání do databáze pomocí PHP a SQL:

```
$eventdate = $month."/".$day."/".$year;
// uložení aktuálně vybraného data do $eventdate
$sqlinsert = "INSERT into
eventcalendar(Title,Detail,eventDate,dateAdded) values
('".$title."','".$detail."','".$eventdate."',now())";
// samotný zápis do databáze
$resultinginsert = mysql_query($sqlinsert);
// samotné provedení dotazu
if($resultinginsert ) {
echo "Událost úspěšně vytvořena...";
} else {
echo "Chyba zkuste znovu...";
// neočekávaná chyba
}
```

5.7 Ovládání a funkce stránek

5.7.1 Možnosti návštěvníka stránek

Návštěvník může prohlížet obsah jednotlivých stránek, prohlížet si fotky v existujících albách, kde po kliknutí na miniaturu obrázku se zobrazí jeho zvětšená verze za pomoci jQuery. Dále pak může odesílat případné dotazy pomocí kontaktního formuláře a ve vysouvacím kalendáři událostí sledovat nadcházející pořádané akce.



Obrázek 12 – prostředí návštěvníka stránky

Kontakt

Pokud máte dotazy kontaktujte Nás

Doplněte chybějící pole.

Jméno:

Email:

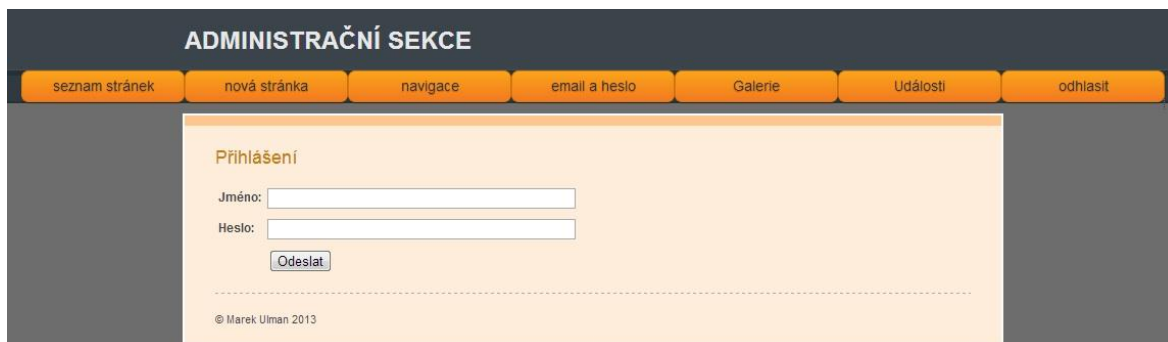
Telefon:

Zpráva:

Obrázek 13 – kontaktní formulář

5.7.2 Možnosti a Ovládání administrační části

Pro vstup do administrační sekce slouží tlačítko administrace v uživatelské sekci stránek nebo zadání adresy `/admin/login.php` v názvu stránky. Po vstoupení do administrační sekce se administrátorovi stránek zobrazí následující stránka.

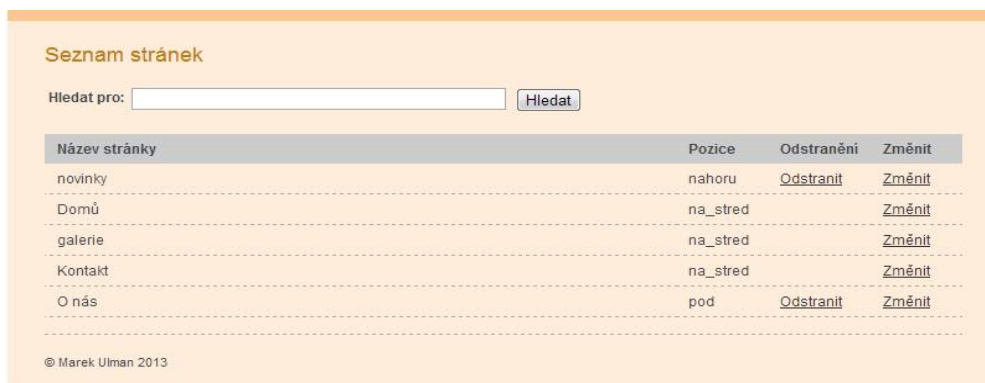


Obrázek 14 – administrační sekce

Zde administrátor zadá přihlašovací údaje a po úspěšném přihlášení bude mít možnost přístupu do všech sekcí administrace. V opačném případě bude upozorněn na zadání nesprávných údajů a odkázán na úvodní stránku pro přihlášení.

5.7.2.1 Seznam stránek

Po úspěšném přihlášení se zobrazí sekce seznam stránek, která obsahuje všechny existující stránky a umožňuje případnou změnu obsahu, pozici menu nebo trvalé odstranění z databáze a uživatelské sekce. Při větším počtu stránek nabízí i vyhledávání podle jejich názvu a tím i rychlejší nalezení požadované stránky. Stránky galerie, Domů a kontakt jsou ve zdrojovém kódu ošetřeny proti možnosti smazání.

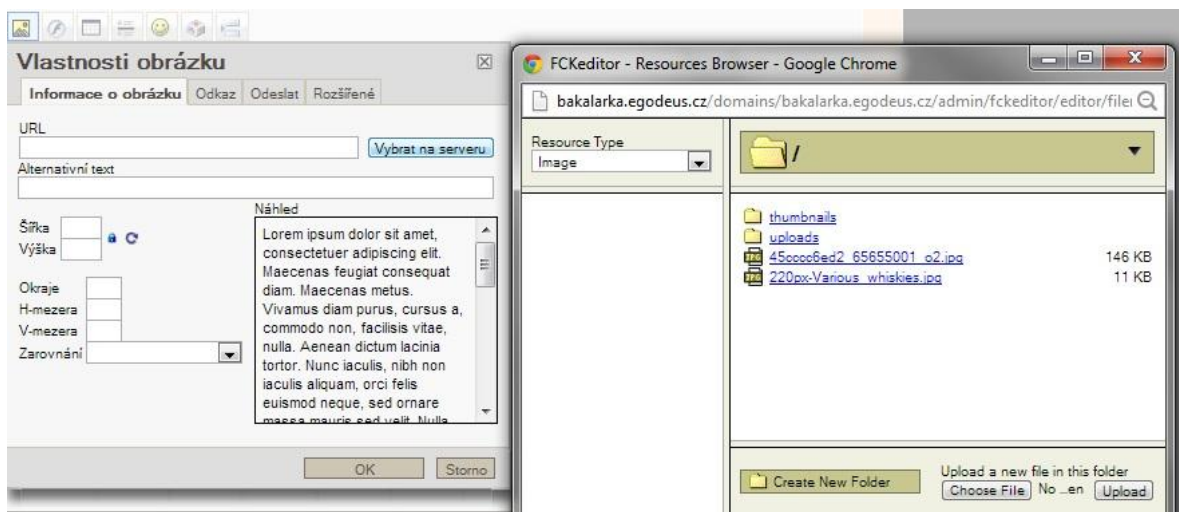


Obrázek 15 – seznam stránek

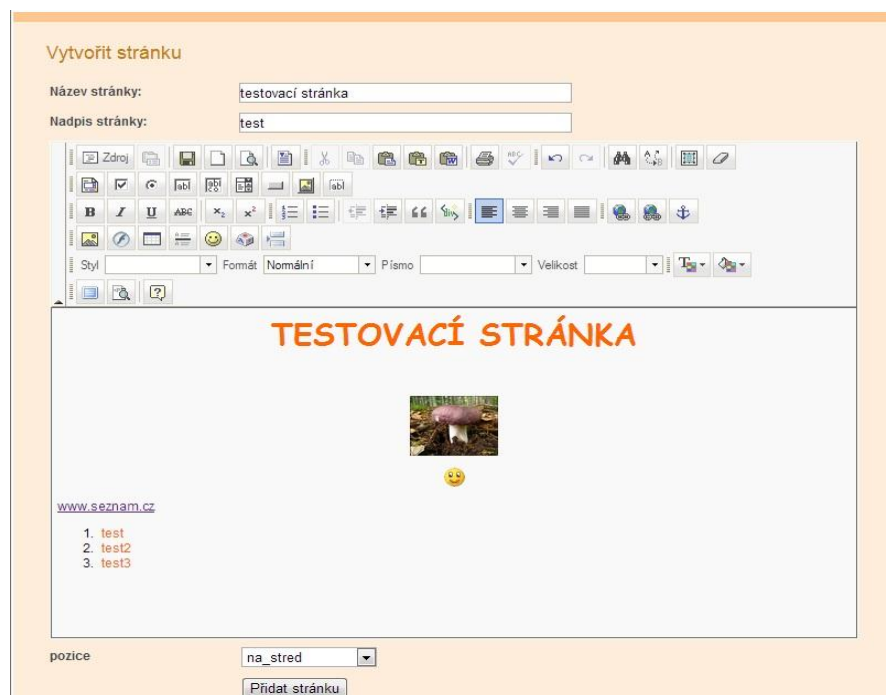
5.7.2.2 Nová stránka

V této části je možné přidávat nové stránky pro prezentaci. Vkládání nového obsahu vč. nahrávání nových obrázků prostřednictvím souborového manažeru (viz. Obrázek 13) je snadné díky FCKeditoru popsanému výše, který byl přeložen z velké části do češtiny. Po naplnění nového obsahu a vyplnění všech povinných polí jakými jsou název stránky,

nadpis a pozice umístění menu a kliknutím na vytvořit stránku dojde k jejímu uložení do databáze.



Obrázek 16 – souborový manažer FCKeditoru pro vkládání obrázků



Obrázek 17 – vytvoření nové stránky v prostředí FCKeditoru

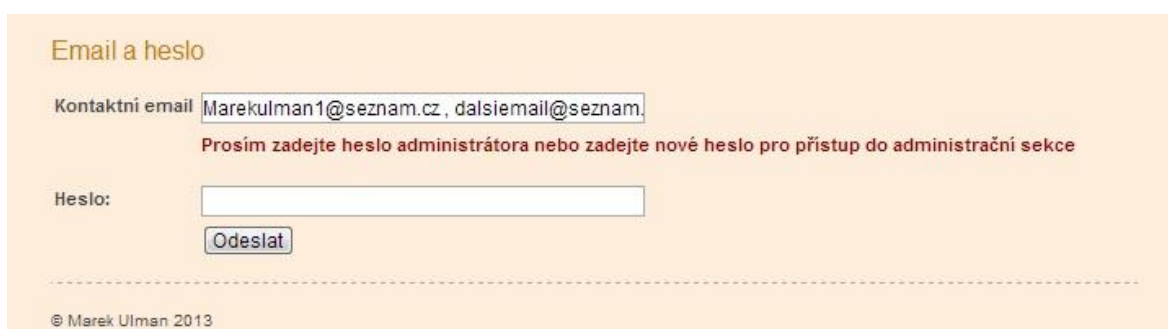
5.7.2.3 Navigace

V sekci navigace je možné měnit pořadí stránek, které se projeví v seznamu stránek a v uživatelské části. Nejnižší číslo představuje první stránku naopak nejvyšší poslední. Po stisknutí tlačítka aktualizovat dojde ke změně pořadí.

5.7.2.4 E - mail a heslo

Tato část slouží pro změnu nebo zadání e - mailové adresy pro zasílání zpráv z kontaktního formuláře a taky pro změnu hesla administrátora. E-maily se oddělují čárkou. Na vložené e -maily budou zasílány zprávy z kontaktního formuláře, který se nachází v uživatelské části. Pro následné uložení do databáze je nutné zadat heslo administrátora.

Pokud chceme změnit heslo do administrační části, tak místo původního hesla zadáme nové, které se pomocí hash funkce sha-1 uloží v zašifrované podobě do databáze. Při odhlášení a následném dalším přihlášení musíme zadat již nové heslo.



Obrázek 18 – změna e-mailu a hesla

5.7.2.5 Galerie

Tato část slouží pro administraci fotek, kde je možná provádět následující operace:

- vytvářet nebo mazat jednotlivé kategorie,
- smazat celou kategorii vč. všech fotek,
- smazat zvolenou fotku ze zvolené kategorie,
- nahrávat nové fotky ve formátu jpg, jpeg, png do zvolené kategorie,
- zadávat stručný popis fotky,
- zobrazení seznamu všech dostupných kategorií

Chyba! - Musíte zvolit kategorii

Kategorie

vytvořit nové album:

pro vytvoření nové kategorie zadejte název:

houby1 ✘

test ✘

Název_soubor: No file chosen vyber album:

žádné album ▼

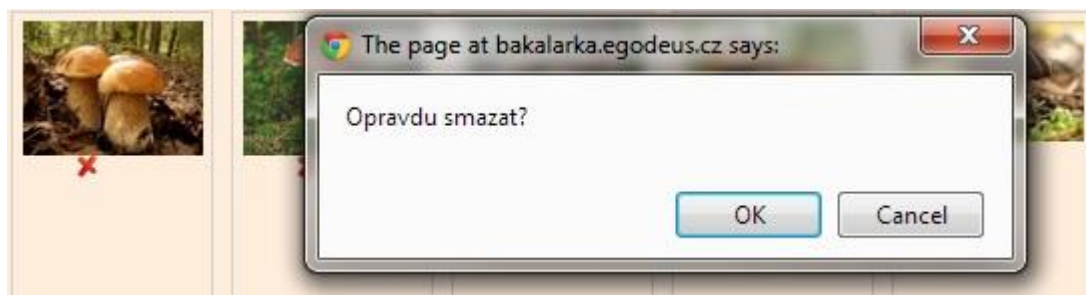
Popisek:

všechny dostupné alba - [zpět na seznam alb](#)

[houby1](#)

[test](#)

Obrázek 19 – nahrávání nových fotek



Obrázek 20 – smazání vybrané fotky s ověřením

5.7.2.6 Události

V této sekci se provádí přidání, editace nebo mazání událostí v kalendáři. Kalendář zvýrazňuje zelenou barvou aktuální den a červeně den pro který je naplánována nějaká událost. Kliknutím na červeně označený den se zobrazí aktuální událost, kterou je možné editovat nebo s potvrzením úplně smazat. Pro vytvoření nové události stačí kliknout na nějaký den ve zvoleném měsíci. Přidávání, editace a mazání událostí je možné pouze v administrační sekci.



Obrázek 21 – kalendář událostí administrace

5.7.2.7 Odhlásit

Kliknutím na tlačítko odhlásit dojde k odhlášení uživatele a zamezení přístupu do jednotlivých sekcí administrace. Bude zobrazena na stránku pro přihlášení..

6 ZPROVOZNĚNÍ NA SERVERU

6.1 Na lokální stanici

Ke zprovoznění webové prezentace na lokální stanici především pro účely testování musíme mít nainstalovaný webový server, nejčastěji používaným je server Apache. Pro tento účel byl zvolen program WampServer, který v sobě obsahuje kromě Apache i PHP a MySQL a pro snazší práci s databází i prostředí phpMyadmin. Po instalaci WampServeru stačí už jen zkopírovat obsah stránek do složky `wamp/www/` a nahrát databázi v prostředí phpMyAdmin. Soubor `databaze.sql` se nachází v příloze. Poté už jen zbývá v konfiguračním souboru `Connection/coondb.php` nastavit propojení s databází. A Nakonec v internetovém prohlížeči zadat `localhost/prezentace` a zobrazí se nám příslušné stránky.

6.2 Na placeném Webhostingu

Samotná práce bude v nejbližší době umístěna na některém placeném hostingu s vlastní doménou 2.řádu. Pro zkušební provoz byla prezentace umístěna na placený hosting společnosti WEDOS Internet, a.s., která disponuje neomezeným datovým prostorem, neomezeným počtem e-mailových schránek, PHP 5.3 a PHP 5.4, MySQL 5.5 s prostorem 1 GB pro databáze a FTP. Výhodou je i široká podpora nastavení `.htaccess`, dostupností 99,99%, neomezeným množstvím přenášených dat (traffic) a nabízí spoustu dalších možností. Prezentace byla zaregistrována s doménou 2.řádu jako `bakalarka.geodeus.cz`.

Po schválení radou Mykologického kroužku Nivnice bude zaregistrována doména 2.řádu a prezentace se umístí na placený hosting, kde následně administrátor stránek vloží obsah.

7 ZABEZPEČENÍ PREZENTACE

V dnešní době, která je plná internetových hrozeb je zapotřebí dobře zabezpečit stránky proti odcizení citlivých údajů, mezi které patří útoky do administračních rozhraní stránek a údajů z databází prostřednictvím chyb v zabezpečení webových stránek, kde největší nebezpečí představují vstupy na stránce jakými jsou například vstupní formuláře pro odesílání údajů do databáze.

7.1 Zabezpečení hesla administrátora v databázi

Heslo do administračního rozhraní stránek bylo zabezpečeno pomocí Hash funkce sha1, která vstupní řetězec zakóduje jako otisk, který se uloží do databáze ve tvaru "kódu", kde hlavní výhodou funkcí typu hash je, že zjištění původního hesla z otisku nebylo zatím prolomeno.

příklad zakódování hesla pomocí funkce sha1:

```
vstupní řetězec: password
```

```
pomocí sha1:5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8
```

```
vstupní řetězec: password85
```

```
pomocí sha1:741a03fecc831b80fb70ccf1f9735d99ce9f9bad
```

zakódování řetězce lišícího se jen přidáním dvou znaků a to konkrétně čísla 85 vznikne úplně jiný zakódovaný výstup, proto odvození vstupu by bylo možné jen vyzkoušením všech možností. V PHP můžeme využít i jiné hash funkce jako je MD5 nebo pomocí parametrů funkce Hash třeba sha 256.

zápis pomocí kódu v PHP:

```
$password=sha1($_POST['frmpass']);
```

7.2 Ošetření typů souborů

Do galerie fotek lze nahrávat jen soubory obrázků typu png, jpg a jpeg, proto jiné formáty nejsou povoleny. Při pokusu nahrát jiný typ souboru než je povolený bude vypsáno upozornění "špatný formát souboru!". Při nahrávání souborů v sekci nová stránka v prostředí FCKeditoru je umožněno vkládat obrazové formáty typu: jpg, JPG, jpeg, JPEG, png, kde ostatní typy souborů nebudou akceptovány.

Ošetření proti jiným než povoleným typům souborů je v obou případech zajištěno podmínkou, zdaly daný typ souboru je obsažen v poli.

```
$type = $_FILES["file_upload"]["type"];  
switch($type)  
{  
  
    case "image/jpeg":  
        $ext = ".jpeg";  
        break;  
    ...  
}
```

kde se následně otestuje zda je formát obsažen v poli.

```
IF ($_FILES["file_upload"]["type"]== "image/jpg") || ...  
...
```

pokud daný formát nevyhovuje zobrazí se chybová hláška v opačném případě se pokračuje dále v kódu. Díky kombinaci s funkcí **exif_imagetype()**, která ověřuje typ grafického formátu čtením informací a struktury vkládaného souboru.

7.3 Zabezpečení FCKeditoru

Pro přístup do souborového systému editoru pro nahrávání a výběr fotek má přístup jen přihlášený administrátor stránky. Proto v konfiguračním souboru editoru config.php testujeme zda je přihlášený uživatel. Pokud ano povolíme přístup do souborového systému v opačném případě je souborový systém zakázán.

```
if (isset($_SESSION['MM_Username']) && $_SESSION['MM_Username'] != ' ' )
{
$Config['Enabled'] = true ;
} else {
$Config['Enabled'] = false ;
}
```

kde funkce `isset` slouží pro ověření platnosti určité proměnné.

proměnná `$Config['Enabled']` je klíčová pro funkci souborového systému pokud je rovna `false`, tak není funkční žádná jeho možnost a je zobrazena chybová hláška.

7.4 Další typy útoků

Mezi nejčastější typy útoků patří SQL Injection a XSS (Cross-site scripting) přes neošetřené formuláře, které využívají metody POST a přes adresu URL, kde se předávají parametry metodou GET.

7.4.1 SQL Injection

Využívá možnost získání citlivých dat u webových aplikací, které pracují s daty uloženými v databázi. Pomocí logicky upořádaných dotazů je v nezabezpečené aplikaci možnost získat například přihlašovací údaje pro přístup do administrace stránek a tudíž možnost měnit jejich obsah. U internetových obchodů je to možnost odcizení důvěryhodných údajů o zákaznících a podobně. Proto je nutné zabezpečit webové stránky před touto hrozbou. [1]

7.4.2 Ochrana proti SQL Injection

Pro ošetření možnosti útoku sem využil funkce `GetSQLValue`, která otestuje správnost dotazu pomocí kombinace níže uvedených funkcí, které sou v PHP vytvořeny pro eliminaci SQL útoku. Funkce ověří korektnost SQL dotazu a případné nedovolené znaky nahradí jinými znaky, které znemožní provedení dotazu.

mezi tyto funkce patří:

mysql_real_escape_string, která nahradí speciální znaky, kterými jsou:

\x00, \n, \r, \, ', " zpětným lomítkem \

příklad funkce:

```
<?php
    $retezec= " test'funkce";
    $osetreni_funkce=mysql_real_escape_string($retezec);
    printf ("ošetřený řetězec: %\n", $osetreni_funkce);
?>
```

Výstupem bude ošetřený řetězec ve tvaru: test\' funkce

mysql_escape_string její funkce je obdobná předchozí funkci, proto záleží na verzi PHP, která funkce bude fungovat

get_magic_quotes_gpc

tato funkce má v PHP hlavní funkci především u \$_POST, \$_GET, \$_COOKIE, kde automaticky přidá / například před ', ", \.

příklad:

pokud odešlu prostřednictvím formuláře metodou \$_POST['navez']

třeba " test'n ", tak pokud je funkce zapnuta se do proměnné \$_POST['navez'] uloží " test\n

proto pro následné uložení do databáze se musí využít funkce **stripslashes()**, která pro změnu odebere nadbytečná zpětná lomítka. Účel těchto funkcí je tedy vzájemnou kombinací bránit se SQL Injection ještě před provedením vlastního SQL dotazu. Funkce GetSQLValueString() využívá podmíněného (ternárního) operátoru (? :),

který má následující tvar:

podmínka ? hodnota je True : hodnota je false;

příklad:

```
$vek=17;
```

```
$typ= $vek >16 ? 'dospívající ' : ' dítě' ;
```

Je to víceméně jiný zápis podmínky if, který je ale více přehledný.

Funkce GetSQLValueString pracuje zkráceně následovně:

```
$theValue = get_magic_quotes_gpc()? stripslashes($theValue) :
    $theValue;
```

//pokud existuje funkce **get_magic_quotes_gpc()**, tak se použije **stripslashes()** v opačném případě se na hodnotu v proměně \$theValue aplikují funkce níže.

```
$theValue = function_exists("mysql_real_escape_string") ?
mysql_real_escape_string($theValue):mysql_escape_string($theValue);

switch ($theType) {

    case "text":

        $theValue = ($theValue != "")? "'" . $theValue . "'" : "NULL";

        break; ... return $theValue;
```

// pomocí case se zjistí datový typ, kterým může být text, int, date a podobně. Po ošetření vstupní hodnoty kombinací výše uvedených funkcí je ve vrácené proměnné \$theValue korektní řetězec pro SQL dotaz.

Funkce se volá u všech přístupů do databáze například:

```
$Login=sprintf("SELECT adm_username, adm_pass FROM tbl_admins
WHERE adm_username=%s AND adm_pass=%s",
GetSQLValueString($loginUsername, "text"),
GetSQLValueString($password, "text"));
```

Kde \$loginUsername, \$password jsou řetězce odesílané metodou \$_POST prostřednictvím formuláře.

například: při zadání škodlivého dotazu do databáze ve tvaru:

```
SELECT nick, email FROM uziv WHERE nick = ' OR id = 2 ORDER
BY id DESC --'
```

// po ošetření funkcí je znemožněn případný útok, kde nepovolené znaky sou nahrazeny lomítky

```
'SELECT nick, email FROM uziv WHERE nick = '\'' OR id = 2
ORDER BY id DESC --\''
```

7.4.3 Zabezpečení URL adresy

Pro zamezení útoku přes URL adresu prohlížeče, která přenáší data metodou \$_GET se pomocí regulárních výrazů zamezí případnému útoku.

ošetření URL adresy:

```
if (isset($_GET['p']) && $_GET['p'] != '' ) {
if(!eregi("[0-9]{1,2}$", $_GET['p'])) {
$_GET['p'] = 1;
$p = $_GET['p'];
}
$_GET['p'] = 1;
} else {
$_GET['p'] = 1;
}
}
```

Kde se pomocí funkce **eregi()** zajistí, že je možné zadávat jen čísla stránek v rozsahu 0-99. Při zadání jiné hodnoty nebo řetězce se zobrazí stránka Domů, která má číslo předávané v URL adrese = 1. Pro ještě lepší ochranu jsou metody \$_GET chráněny funkcí popsanou výše v SQL Injection.

Vstupní formuláře byly otestovány proti útoku pomocí doplňku pro prohlížeč Firefox **SQL inject me**, který zkouší škodlivé SQL dotazy. Všechny proběhlé testy byly v pořádku, tudíž aplikace v testu obstála.

7.4.4 XSS - Cross Site Scripting

Hlavní vlastnost útoku na nezabezpečenou stránku spočívá ve vložení vlastního HTML kódu, který může útočnickovi umožnit vložit do stránky svůj skript nebo podstrčit svůj formulář a získat tak informace, aniž by to návštěvník stránky nějak poznal. Může například vložit do URL adresy prohlížeče nějaký skript, kterým zamezí zamrznutí stránky. Proto je nutné tuto okolnost ošetřit ve všech \$_POST a \$_GET metodách, které nejčastěji pracují s formuláři. Podstatou je nahradit HTML značky jinými a tím zabránit jejich funkci.[1]

7.4.4.1 Ošetření

Pro nahrazení HTML znaků použijeme v PHP funkci **htmlspecialchars()**, která nahradí HTML značky a zamezí tím jejich funkčnosti.

příklad:

```
<?php
$odstraneni = htmlspecialchars("<a
href='test'>Test</a>");
echo $odstraneni;
?>
```

Na výstupu se tedy nahradí znaky na `Test`;

Vstupní formuláře byly otestovány proti útoku XSS pomocí doplňku pro prohlížeč Firefox **XSS me**, který zkouší vkládání scriptů. Nebyl nalezen žádný problém.

7.5 Zabezpečení kontaktního formuláře

Kontaktní formulář obsahuje kontrolu vyplnění všech povinných polí, kdy při nevyplnění některého z nich je uživatel na skutečnost upozorněn. Všechny zadávané údaje jsou ošetřeny proti XSS funkcí **htmlspecialchars()** proti vložení spustitelného skriptu. Dále pomocí regulárních výrazů je ošetřen správný formát e-mailu, kdy při zadání neplatného e-mailu bude uživatele upozorněn. Při správnosti všech údajů bude e-mail odeslán.

Ověření správného formátu e-mailu:

```
$checkEmail = '/^[^@]+@[^\s\r\n\';,;%]+$/';
```

zamezení vložení škodlivého kódu

```
$value="'.htmlspecialchars($_POST['frmname']).'";
```

7.6 Omezení přístupu k souborům

Pro zabránění přístupu ke konfiguračním souborům bylo využito konfiguračního souboru `.htaccess`, který díky tomu, že běží na straně webového serveru znemožní přístup a případné zneužití důležitých dat z konfiguračních souborů. Umístěním souboru `.htaccess` do složky `Connections/` ve které se nachází konfigurační soubor s přístupovými údaji do databáze nebude soubor `coondb.php` možné zobrazit a získat tak citlivé údaje.

parametry v souboru `htaccess`:

```
<Files "coondb.php">
Order Allow,Deny
Deny from All
</Files>
```

7.7 Nepovolený přístup do částí administrační sekce

Byla vytvořena funkce, která bez přihlášení uživatele znemožní přístup do všech částí administrační sekce. Funkce je volána v každé části kódu administrace.

vytvořená funkce:

```
function verity_deus() {
    if($_SESSION['MM_Username']!=""):
// pokud existuje relace s údaji o přihlášeném uživateli je
umožněn přístup do všech částí administrační sekce
    else:
// session neexistuje uživatel není přihlašen, pošleme ho na
stránku pro přihlášení
    header("Location: login.php" );
endif;
}
```

8 MOŽNOSTI BUDOUCÍHO VYLEPŠENÍ PREZENTACE

Aplikace je už od začátku navržena a koncipována pro případné přidání nových funkcí do administrační nebo uživatelské sekce prezentace. Do budoucna je v plánu naprogramování diskusního fóra, které bude jako již vytvořené části komunikovat s databází. Pro přidání je nutná jednoduchá úprava databáze přidáním nových specifických tabulek a pro začlenění do administrační části stačí vytvořit v menu administrace novou položku pro správu fóra. Funkce diskusního fóra budou probrány s členy kroužku.

Změna grafického vzhledu administrační nebo uživatelské části je rovněž jednoduchá a to díky dvěma různým souborům stylů CSS pro každou část, které určují umístění a grafický vzhled všech částí stránek. Prezentaci je tedy možné v budoucnu nadále vylepšovat jak po stránkách funkcí tak grafického vzhledu.

ZÁVĚR

Webové prezentace různých kroužků, sdružení a podobně jsou v dnešním světě internetu značně rozšířené a to díky dnes již dostupným technologiím pro jejich vytvoření. Poskytují jejím návštěvníkům možnost jednoduše sledovat informace o jejich aktivitách.

Cílem této práce bylo na základě požadavků členů sdružení Mykologického kroužku Nivnice a seznámení se s některými již existujícími Mykologickými kroužky a kluby vyskytující se na internetu následně navrhnout a vytvořit webovou prezentaci jejich kroužku se zaměřením na jednoduchost ovládání a její zabezpečení.

Pro návrh jsem vybral technologii MySQL a PHP, která je pro návrh nejvhodnější a pro podobné webové aplikace hojně rozšířená. V teoretické části popisuji použité technologie, které sem použil pro realizaci prezentace. V praktické části jsem se snažil popsat jednotlivé kroky návrhu až po popis samotného ovládání a umístění stránek na web. Zmínil jsme i možnosti dalšího rozšíření prezentace.

Podářilo se mi vytvořit funkční zabezpečenou webovou prezentaci s mnoha funkcemi, která bude prezentovat kroužek na internetu a případným návštěvníkům stránek poskytnout cenné informace o jeho dění.

ZÁVĚR V ANGLIČTINĚ

Web presentations of various clubs, associations and so on, they are greatly enhanced thanks to the technologies available today. They provide its visitors the ability to easily track information about activities they are interested in.

The aim of this work was to create a design and create a web presentation with focus on its security and esines to use. It was is based on the cooperation with the members of mycological group Nivnice and familiarization with some existing mycological groups and clubs appearing on the internet.

For the design I chose MySQL and PHP technology that is best suited and widely used for such kind of design. In the theoretical part I describe the technology that I used for the realization of practical prezentace. In the practical part I tried to describe the steps from the the design to the control and the placement of the pages on the web. I mentioned also the possibility of further extension of the presentation.

I managed to create a functional secure web presence with many features that will present the group on the internet and will help any visitors of the site to find valuable information about its events.

SEZNAM POUŽITÉ LITERATURY

- [1] ZEMEK, Lukáš. Bezpečnost webových aplikací. Praha, 2012. Bakalářská práce (Bc.). Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky. Vedoucí práce doc. Ing. Martin Sysel, Ph.D.
- [2] PROKOP, Marek. CSS kaskádové styly pro webdesignéry. 2. vyd. Praha: COMPUTER PRESS, 2005. ISBN 80-86593-35-5.
- [3] DLOUHÝ, Radek. PHP v příkladech. Vyd. 1. Kralice na Hané: Computer Media, 2007. ISBN 80-86686-83-3.
- [4] GUTMANS, Andi, Derick RETHANS a Stig SAETHER BAKKEN. Mistrovství v PHP 5. 2. vyd. Praha: COMPUTER PRESS, 2007. ISBN 8025115194.
- [5] NARAMORE, Elizabeth, Jason GERNER, Yann Le SCOUARNEC, Timothy BORONCZYK. PHP 6, MySQL, Apache: Vytváříme webové aplikace. 2.vyd. Praha: COMPUTER PRESS, 2011. ISBN 978-80-251-27.
- [6] LACKO, Luboslav. PHP 5 a MySQL 5: Hotová řešení. 1. vyd. Brno: COMPUTER PRESS, 2007. ISBN 978-80-251-1695-1.
- [7] SCHAFER, Steven M. *HTML, XHTML a CSS: bible [pro tvorbu WWW stránek] : 4. vydání*. 1. vyd. Praha: Grada, 2009, 647 s. ISBN 978-80-251-2940-1.
- [8] JAN, Stejskal. Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu. Vyd. 1. Brno: Computer Press, 2004, 250 s. ISBN 80-251-0167-3.
- [9] Jak vybrat hosting. [online]. [cit. 2012-07-31]. Dostupné z: www.jakvybrathosting.cz/
- [10] LACKO, Luboslav. SQL: kapesní přehled. Vyd. 1. Brno: CP Books, 2005, 96 s. ISBN 80-251-0788-4.
- [11] KOFLER, Michael a Bernd ÖGGL. PHP 5 a MySQL 5: průvodce webového programátora. Vyd. 1. Brno: Computer Press, 2007, 607 s. ISBN 978-80-251-1813-9.
- [12] VERENS, Kae. JQuery 1.3 with PHP. Birmingham: Packt Publishing, 2009. ISBN 978-1-847196-98-9.
- [13] LACKO, Luboslav. *1001 tipů a triků pro SQL*. Vyd. 1. Brno: Computer Press, 2011, 416 s. ISBN 978-80-251-3010-0.

- [14] Overview. What is FCKeditor? [online]. [30 April 2013] [cit. 2013-06-03]. Dostupné z:
http://docs.cksource.com/FCKeditor_2.x/Users_Guide/Overview
- [15] PSPad. Textový editor PSPad [online]. 25.1.2013 [cit. 2013-05-18]. Dostupné z:
<http://www.pspad.com/cz/>
- [16] WampServer, the web development platform on Windows - Apache, MySQL, PHP. WAMPSEVER [online]. 30.4.2013 [cit. 2013-05-18]. Dostupné z:
<http://www.wampserver.com/en/>
- [17] Adobe Photoshop CS5. Adobe Photoshop CS5 [online]. [cit. 2013-06-05]. Dostupné z:
<http://www.amsoft.cz/Produkty/adobe/photoshop/main.html>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

HTML	Hyper Text Markup Language
PHP	Hypertext Preprocessor
SQL	Structured Query Language - dotazovací jazyk
CSS	Cascading Style Sheets – kaskádové styly
URL	Uniform Resource Locator – slouží k přesnému umístění zdroje informací
HTTP	Hypertext Transfer Protocol – slouží pro výměnu hypertextových dokumentů
PDF	Portable Document Format – dokumentový formát
WWW	World Wide Web
FTP	File Transfer Protocol – slouží pro přenos souborů pomocí počítačové sítě
ID	Identifikační číslo
XSS	Cross-site scripting – metoda narušení webů
JPEG	Joint Photographic Experts Group (někdy jen JPG)
PNG	Portable Network Graphics
CD	Compact disk
W3C	World Wide Web Consortium

SEZNAM OBRÁZKŮ

Obrázek 1 – Prostředí programu PSPad	28
Obrázek 2 - Ovládací panel WampServeru	29
Obrázek 3 – prostředí photoshopu CS5 při tvorbě grafiky	30
Obrázek 4 – Prostředí FCKeditoru	31
Obrázek 5 – stránky Mykologického kroužku Plzeň.....	33
Obrázek 6 – stránky Mykologického klubu Brno.....	34
Obrázek 7 – předpokládané rozložení prvků na stránce	36
Obrázek 8 – úvodní obrazovka	41
Obrázek 9 – uživatelská část prostřednictvím CSS - bez obsahu	41
Obrázek 10 – kalendář událostí v CSS	42
Obrázek 11 – grafická podoba hotového administračního prostředí	42
Obrázek 12 – prostředí návštěvníka stránky.....	45
Obrázek 13 – kontaktní formulář	45
Obrázek 14 – administrační sekce	46
Obrázek 15 – seznam stránek	46
Obrázek 16 – souborový manažer FCKeditoru pro vkládání obrázků	47
Obrázek 17 – vytvoření nové stránky v prostředí FCKeditoru.....	47
Obrázek 18 – změna e-mailu a hesla	48
Obrázek 19 – nahrávání nových fotek	49
Obrázek 20 – smazání vybrané fotky s ověřením.....	49
Obrázek 21 – kalendář událostí administrace	50

SEZNAM TABULEK

Tabulka 1 – Číselné operátory	14
Tabulka 2 – Operátory přiřazení	15
Tabulka 3 – Porovnávací operátory	15
Tabulka 4 – Logické operátory	16
Tabulka 5 – Příkazy pro práci s databází MySQL	23
Tabulka 6 – Příkazy pro tvorbu dotazů MySQL	23
Tabulka 7 – Tabulka eventcalendar	37
Tabulka 8 – photos	38
Tabulka 9– Tabulka kategorie	38
Tabulka 10 – Tabulka tbl_admins	38
Tabulka 11 –Tabulka tbl_nav	39
Tabulka 12– Tabulka tbl_pages	39

SEZNAM PŘÍLOH

- PI Příložené CD obsahující elektronickou verzi této bakalářské práce a všechny zdrojové soubory opatřené komentáři.