

Galerie fotografií na Internetu

Gallery of photos on the Internet

Miroslav Kovaříček

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: **Miroslav KOVÁŘÍČEK**
Osobní číslo: **A09025**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **prezenční**

Téma práce: **Galerie fotografií na Internetu**

Zásady pro vypracování:

1. Vytvořte webovou galerii fotek dle požadavků.
2. Navrhněte strukturu a vzhled aplikace.
3. Využijte objektové PHP a JavaScript (jQuery).
4. Umožněte hromadný upload souborů a hodnocení fotografií.
5. Věnujte pozornost bezpečnosti aplikace.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

1. JANOVSKEÝ, Dušan. Jak psát web [online]. [2003] [cit. 2013-01-17]. Dostupné z: <http://www.jakpsatweb.cz/>
2. GILMORE, Jason W. Velká kniha PHP 5 MySQL: kompendium znalostí pro začátečníky i profesionály. Vyd. 1. Brno: Zoner Press, 2005, 711 s. ISBN 80-868-1520-X.
3. JQUERY. JQuery: The Write Less, Do More, JavaScript Library [online]. [2010] [cit. 2013-01-17]. Dostupné z: <http://jquery.com/>
4. KAY, Ben. Bunnyfire.co.uk [online]. [April 22nd, 2009] [cit. 2013-01-17]. Dostupné z: <http://bunnyfire.co.uk/>
5. A BEAUTIFUL SITE, LLC. Bunnyfire.co.uk [online]. [2007] [cit. 2013-01-17]. Dostupné z: <http://www.abeautifulsite.net/blog/>
6. SCHLOSSNAGLE, George. Pokročilé programování v PHP 5. Brno: Zoner Press, 2004, 640 s. ISBN 80-868-1514-5.

Vedoucí bakalářské práce:

doc. Ing. Martin Sysel, Ph.D.

Ústav počítačových a komunikačních systémů

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

Bakalářská práce se zabývá webovou galerií. Hlavním bodem bylo vytvořit přehledné stránky pro jednoduchou práci a orientaci. Pro práci bylo vytyčeno několik cílů. Jako je vytvořit stromovou strukturu složek, která bude umožňovat rychlou a efektivní práci se složkami a daty. Dalším cílem bylo využití hromadného nahrávání fotografií či umožnit uživatelům hodnotit obrázky. V práci bylo využito jQuery knihovny.

Klíčová slova: internetové stránky, WWW, PHP, MySQL, JAVASCRIPT, HTML, jQuery

ABSTRACT

This Bachelor thesis deals with web galleries. The main point was to create a user-friendly site for simple work and orientation. Several objectives had to be defined, such as creating a tree folder which will allow fast and effective work with folders and data, using bulk-uploading photos, or allowing users to rate pictures. jQuery library was used for this thesis.

Keywords: website, WWW, PHP, MySQL, JAVASCRIPT, HTML, jQuery

Rád bych poděkoval vedoucímu mé bakalářské práce doc. Ing. Martinu Syslovi, Ph.D. za vedení v průběhu práce, cenné rady pro tvorbu a vylepšení a za všechny starosti spojené s odevzdáním této 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

.....
podpis diplomanta

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST	10
1 POROVNÁNÍ DOSTUPNÝCH GALERIÍ.....	11
1.1 GALERIE.CZ.....	11
1.2 RAJCE.IDNES.CZ	12
1.3 ZONERAMA.COM.....	13
1.4 POROVNÁNÍ.....	14
2 POUŽITÉ PROGRAMOVACÍ JAZYKY	15
2.1 TECHNOLOGIE WWW	15
2.2 HTML.....	16
2.3 CSS.....	16
2.3.1 Přímý inline zápis stylu pomocí atributu style.....	17
2.3.2 Zápis stylů do elementu <code>style</code>	17
2.3.3 Připojení externího souboru pomocí http hlavičky <code>link</code>	17
2.4 PHP.....	18
2.5 JAVASCRIPT	19
2.6 MYSQL	20
2.7 JQUERY	21
2.7.1 Možnosti načtení JQuery knihovny	21
2.7.1.1 Jako jeden javascriptový soubor, obsahujícího všechny funkce pro DOM, Ajax, události a efekty.	21
2.7.1.2 Načtení pomocí Google AJAX Libraries API.	21
2.7.1.3 Načítání jQuery přímo ze serverů Google	21
2.7.1.4 Načtení JQuery přímo z domovské stránky projektu.....	22
3 POUŽITÉ PROGRAMY A PŘÍSTUP K PRÁCI	23
3.1 PHPMYADMIN	23
3.2 XAMPP	23
3.3 PSPAD.....	24
3.4 GIMP	24
3.5 UKLÁDÁNÍ DAT A TVORBA STROMOVÉ STRUKTURY	24
3.5.1 Úskalí relačních databází	24
3.5.2 Úvod.....	25
3.5.3 Klasický přístup	25
3.5.3.1 Úprava dat.....	25
3.5.4 Rozšíření ploché tabulky	25
3.6 DRUHY ŠIFROVÁNÍ HESEL PRO DATABÁZE	26
3.6.1 MD	26
3.6.1.1 MD4.....	26
3.6.1.2 MD5.....	27
3.6.2 SHA.....	27
3.6.2.1 SHA 0 a SHA 1.....	27
3.6.2.2 SHA 2	27

3.6.3	salted hash	28
3.7	PŘENOS HESEL NA SERVER	28
3.7.1	HTTPS.....	28
3.7.2	hash na straně klienta	28
3.8	CHYBY NA WEBU	29
3.8.1	SQL Injection	29
3.8.2	Cross-site scripting(XSS).....	29
II	PRAKTICKÁ ČÁST	31
1.	NÁVRH	32
2.	TVORBA	34
	ZÁVĚR	42
	ZÁVĚR V ANGLIČTINĚ	43
	SEZNAM POUŽITÉ LITERATURY	43
	HYPERTEXT PREPROCESSOR	48
	EXTENSIBLE HYPERTEXT MARKUP LANGUAGE	48
	SEZNAM OBRÁZKŮ:	49
	SEZNAM PŘÍLOH.....	50

ÚVOD

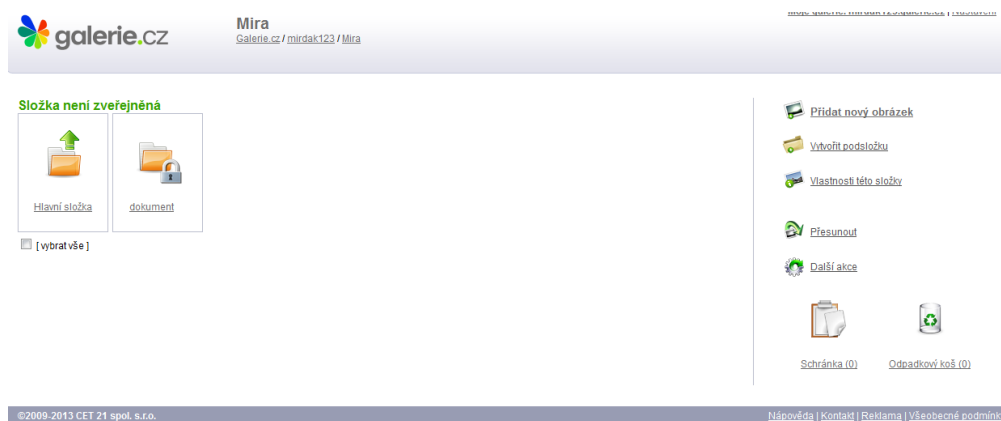
Webové galerie obecně slouží uživatelům k prohlížení fotografií na internetu. Avšak dost uživatelů používá webovou galerii jako prostředek k zviditelnění či ukázce jejich práce, koníčku nebo zážitku z mnoha různých akcí či dovolených. Pro možnost přidávat fotografie do galerií je vyžadována registrace. Po registraci uživatel však získá prostor a možnost vytvářet složky, alby v galerii což obyčejný návštěvník bez registrace nemá. U vytvořených složek/alb mají uživatelé možnost zabezpečení svých složek/alb či zakázat přístup veřejnosti k vybraným složkám. Do takto vytvořených/zabezpečených složek/alb může uživatel nahrávat své fotografie. Galerie taky nabízí základní úpravu fotografií a to nejméně možnost jejich otočení či smazání jednotlivých fotografií. Prostor (paměť) pro nahrávání data se u jednotlivých galerií výrazně liší. Většina galerií pro možnost uložení většího množství dat vyžaduje rozšíření jimi přidělené paměti za finanční obnos. Důležitým prvkem u galerií je samotné prohlížení fotografií. Které bývá přístupné i široké veřejnosti kde galerie nabízí možnost hodnocení či přidání své poznámky k jednotlivým fotografiím. Jednotné galerie nabízí výše zmíněné základní prvky v mnoha různých stylech a úpravách. Jednotlivé galerie taky věnují rozdílnou pozornost zabezpečení jak fotografií, tak složek. Některé galerie nabízí ochranu fotografií pomocí vodoznaku nebo zabezpečení složek pomocí svého hesla.

I. TEORETICKÁ ČÁST

1 POROVNÁNÍ DOSTUPNÝCH GALERIÍ

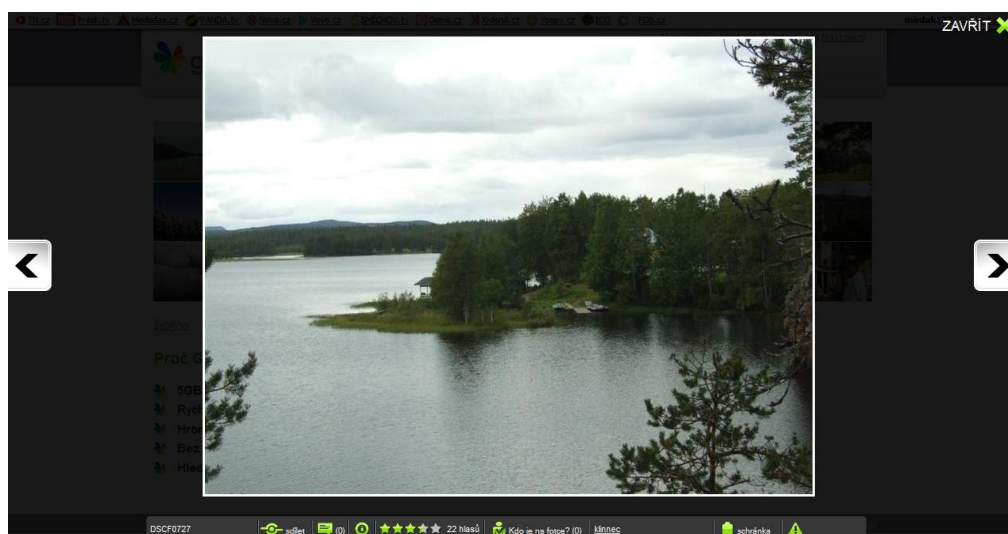
1.1 Galerie.cz

Po spuštění galerie.cz [32] se vám zobrazí, úvodní stránka která vás naláká na výhody této galerie a nabídne vám registraci. Bez samotné registrace máte oprávnění se jen podívat na nabídnuté fotografie. Po přihlášení se objeví příjemné prostředí pro práci se složkami. Kde je možné vytvořené složky zabezpečit heslem, nebo je nastavit veřejnosti nedostupné. Svým uživatelům poskytuje až 5GB místa pro fotografie. Také umožňuje hromadné nahrávání fotografií. Trošku nepřehledné je zobrazení jednotlivých složek a jejich vzájemná návaznost.



Obrázek 1 – galerie.cz menu

Galerie.cz obsahuje kvalitní prohlížení fotografií kde je pěkně přehledně zapracováno hodnocení, komentáře a další podrobné údaje o fotografiích. V galerii je taky možnost základních prací, z fotkou otočení, mazání.



Obrázek 2 – galerie.cz zobrazení fotek

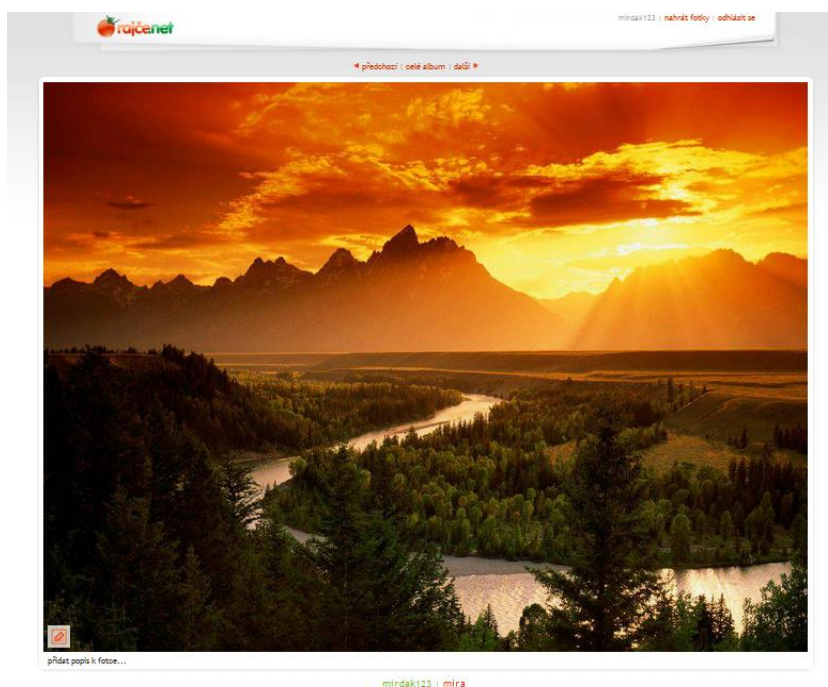
1.2 Rajce.idnes.cz

Rajce.idnes.cz [33] po zobrazení úvodní stránky vám nabízí prohlédnutí fotografií a, nebo přímo vás vybídne k nahrání fotografií. Rajce nabízí dvě možnosti správy vašeho účtu, který si pro práci v galerii musíte vytvořit. Po vytvoření máte možnost pracovat přímo na stránkách, nebo si stáhnout a nainstalovat aplikaci pro práci z galerií. Rajce nabízí neomezenou kapacitu pro nahrání dat. Na stránkách stejně jako na galerie.cz chybí přehlednost a možnost zobrazení uspořádání složek či alb. Máme možnost po vytvoření alby mu nastavit práva a omezení a to stejná jako na galerii.cz tedy heslování nebo zakázání viditelnosti pro veřejnost.



Obrázek 3 – rajce.idnes.cz menu

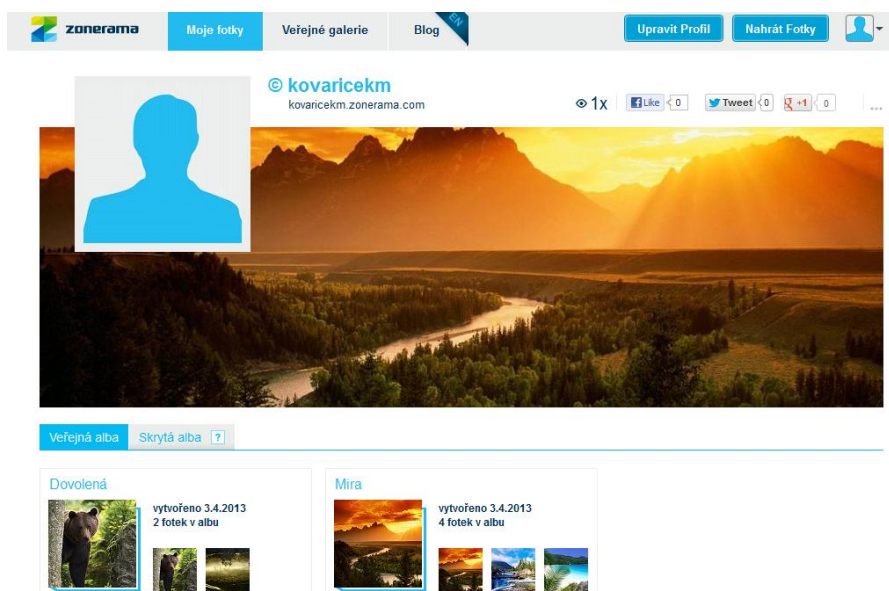
Naproti tomu má rajče oproti galerii.cz k dispozici zabezpečení fotografií ve složkách, albech přes možnost vodoznaku. U fotografií je možnost přidat název či poznámky. Nepovedené je samotné prohlížení, které neumožňuje žádné další funkce kromě přidání komentáře.



Obrázek 4 – rajce.idnes.cz zobrazování fotografií

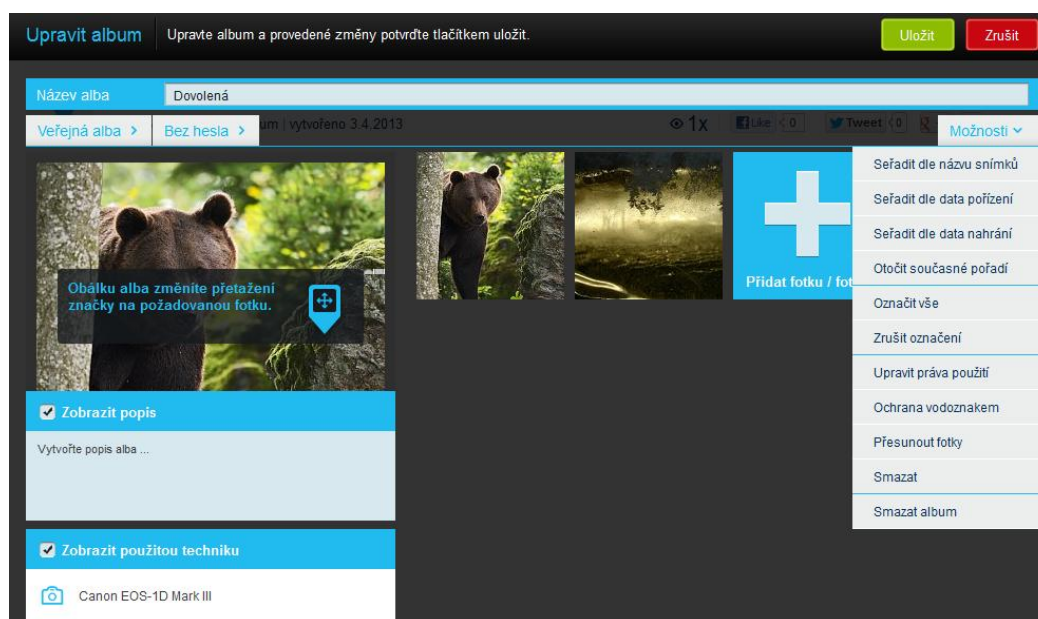
1.3 zonerama.com

Zonerama.com [34] vám hned na úvod nabídne registraci bez provedení registrace, nemáte žádné možnosti. Zato po přihlášení vám nabídne velice příjemné prostředí pro práci. Kde máte možnosti prohlížet veřejné galerie, nebo nahrát vlastní fotky či upravit se váš profil. Při kliknutí na nahrání fotografií vás vybídne k vytvoření nového alba nebo úpravě stávajícího. Je zde velký výběr zabezpečení jak alba, tak samotných fotografií.



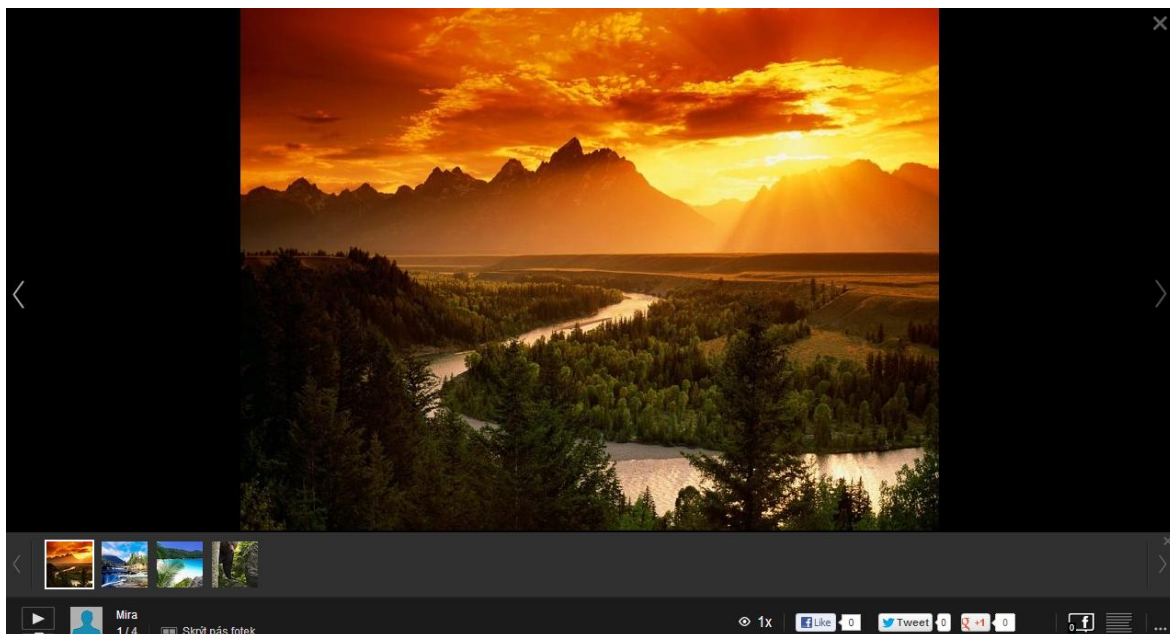
Obrázek 5 – zonerama.com profil uživatele

Po vytvoření alba je možnost album libovolně upravovat. Horší je to u samotných fotografií kde není možnost otáčení.



Obrázek 6 – zonerama.com menu

Samotné prohlížení fotografií je kvalitně graficky zpracováno akorát zde postrádám možnost hodnocení nebo získání bližších informací o fotografii.



Obrázek 7 – zonerva.com zobrazování fotografií

1.4 Porovnání

Popisované galerie určitě nejsou jediné dostupné na internetu se, jich nachází spousta ať už lepe či hůře zpracované. Představené galerie postrádají přehlednost a návaznost vytvořených alb či složek tohle ve své galerii řeším přes stromovou strukturu složek. Samotné vytváření a zabezpečení složek je ve všech galeriích stejné v mé galerii jsou stejné až na možnost zabezpečit složku, album heslem ale naproti tomu jsem rozšířil možnosti práv u jednotlivých uživatelů a jejich možnost přidání a úpravu pro jednotlivé složky. Zobrazení fotografií je téměř totožné jako u mé galerie až na možnost přidání komentáře u fotografií.

2 POUŽITÉ PROGRAMOVACÍ JAZYKY

2.1 Technologie WWW

Web (hovorová zkratka pro World Wide Web - WWW) je graficky orientovaná služba počítačové sítě Internet, což je celosvětová komunikační a informační síť. Služba WWW umožňuje spojení formátovaného textu, grafiky, zvuků případně animací do tzv. WWW stránek, které mohou být mezi sebou propojeny odkazy (linky). Vyvoláním odkazů (většinou kliknutím myši) je zobrazena nová stránka. Vzhledem k těmto vlastnostem je používání služby WWW jednoduché i pro méně fundovaného uživatele. [1]

WWW pracuje na principu klient-server (dříve v dvoustupňové architektuře, nyní stále častěji v modelu třívrstvé architektury klientská vrstva, aplikační vrstva, datová vrstva). Komunikační síť je postavena na přenosovém protokolu TCP/IP a každý server je jednoznačně identifikován svojí IP adresou. V nynější době se používá 32 bitová verze IPv4, kde se adresa skládá ze 4 částí (4 byte = 32 bitů). Každá část může nabývat hodnot 0 až 255. Na síti je umístěno mnoho serverů. Čísla se mnohdy špatně pamatují. Proto jsou servery dostupné přes tzv. doménová jména pro uživatele podstatně lépe zapamatovatelná. Klient zadá adresu serveru a požaduje po něm zaslání informace (WWW stránky). Aby si obě strany „rozuměly“, dotazy a odpovědi jsou prováděny protokolem http (Hyper Text Transfer Protocol). Server přijme požadavek klienta (pokud je správně napsána adresa serveru) a odešle zpět požadované informace (pokud jsou k dispozici). Při prvním přístupu server zasílá v drtivé většině stránku z výchozího bodu okruhu společných informací - homepage. Zvolí-li uživatel na své stránce vyvolání odkazu (linku), celá akce se opakuje. [1]

Získání informací je svázáno do různých pravidel. Výměna probíhá pomocí různých protokolů (TCP/IP, http, ftp atd.), dokument k zobrazení je psán dle určitého předpisu (např. HTML Hyper Text Markup Language). Prezenci výsledné informace u klienta (uživatele) provádějí tzv. prohlížeče (browsersy). Prohlížeč je program, který zobrazí požadovaný dokument (WWW stránku) na klientském počítači. V nynější době jsou prohlížeče plně graficky orientované, pouze textové prohlížeče byly používány na počátku rozmachu Internetu. Mezi nejznámější prohlížeče patří Microsoft Internet Explorer (včetně soudních sporů, zda může nebo nemůže být součástí operačního systému), Mozilla, FireFox, Opera, Netscape Navigator a další. [1]

Chod Internetu a WWW je rozprostřen po celém světě. Aby si všichni rozuměli a nedocházelo k různým pádům a haváriím, bylo nutno stanovit jasná pravidla – standardy. Návrhem pravidel pro Internet či web se zabývá několik různých organizací nebo komunit, z nichž nejznámější je konsorcium World Wide Web Consortium (ve zkratce W3C). [1]

Pro pohodlnou adresaci v Internetu je důležitý pojem doména. Doména je adresovatelná část počítačové sítě. Je organizována v tzv. doménových jménech definující podmnožinu počítačů spojených do určité jednotky (organizační, teritoriální...). O chod doménových jmen států se starají národní správci. V ČR je to CZ.NIC, což je zájmové sdružení právnických osob. Sdružení vede Centrální registr a provádí registraci a správu doménových jmen. [1]

2.2 HTML

HyperText Markup Language, označovaný zkratkou HTML, je značkovací jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu. Tento protokol vznikl v roce 1990. Během let se jazyk vyvinul z verze 0.9 (rok 1991) až do verze 5 (2007). Jazyk je aplikací dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language). Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.[2][3][4]

HTML jazyk je množinou značek (tzv. tagů) a jejich atributů definovaných pro danou verzi. Mezi značky se uzavírají části textu dokumentu a tím se určuje význam (sémantika) obsaženého textu. Názvy jednotlivých značek se uzavírají do hranatých závorek < >. HTML tagy nejčastěji přicházejí v párech jako <h1> a </ h1>, i když některé značky, známé jako prázdné prvky, jsou nepárová, například . První značka v páru je počáteční značka, druhá značka je značka konec (jsou tzv. tagy otevírání a zavírání). HTML prvky tvoří stavební kameny všech webových stránek. [2][3][4]

Cílem webového prohlížeče je číst HTML dokumenty a skládat je do optických nebo akustických webových stránek. Prohlížeč nepodporuje zobrazení HTML značky, ale používá tagy k výkladu obsahu stránky. [2][3][4]

Ukázka základního rozložení stránky:

```
<!-- ohraničení internetové stránky začátek html dokumentu -->
<html xmlns="http://www.w3.org/1999/xhtml">
<head> <!-- Začátek html dokumentu (hlavička) -->
<!-- Nastavení kódování -->
<meta http-equiv='Content-Type' content='text/html; charset=utf-8' />
  <!-- Nastavení jazyka -->
<meta http-equiv='Content-language' content='cs' />
  <title>Titulek stránky</title>
</head> <!-- Konec html dokumentu (hlavička) -->
  <body> <!-- Tělo dokumentu -->
<div><!-- Samotný obsah stránky který má být vidět --></div>
</body>
</html>
```

2.3 CSS

CSS vzniklo někdy kolem roku 1997. Je to kolekce metod pro grafickou úpravu webových stránek napsaných v jazycích HTML, XHTML nebo XML. Ta zkratka znamená Cascading Style Sheets, česky "kaskádové styly". Kaskádové, protože se na sebe mohou vrstvit definice stylu, ale platí jenom ta poslední.[5][6][7][8]

Jazyk byl navržen standardizační organizací W3C, autorem prvotního návrhu byl Håkon Wium Lie. Byly vydány zatím tři úrovně specifikace CSS1 a CSS2, CSS3. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují

celou řadu elementů, které nepopisují obsah a strukturu dokumentu, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takový vývoj žádoucí. [5][6][7][8]

Existuje několik možných způsobů, jak aplikovat kaskádové styly v HTML dokumentu.

2.3.1 Přímý inline zápis stylu pomocí atributu style.

Tato pravidla budou aplikována pouze na dotyčný element, jak je zobrazeno níže.

```
<p style="color: red; text-decoration: underline">
Tento odstavec bude červený a podtržený</p>
```

2.3.2 Zápis stylů do elementu style.

Takové styly se aplikují na celou stránku podle předepsaných selektorů.

Ukázka zápisu stylů do elementu style:

```
<style type="text/css">
#hlavicka{
    width: 200px;
    height: 450px;
}
</style>
```

2.3.3 Připojení externího souboru pomocí http hlavičky link.

Ukázka nahrání externího souboru se styly:

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

Definice kaskádových stylů se skládá z několika *pravidel*. Každý příkaz obsahuje *selektor* a *blok deklarací*. Každý blok deklarací pak obsahuje seznam *deklarací* oddělených středníky; a každá deklarace se skládá z identifikátoru vlastnosti, následuje dvojtečka: a hodnota vlastnosti. [5][6][7][8]

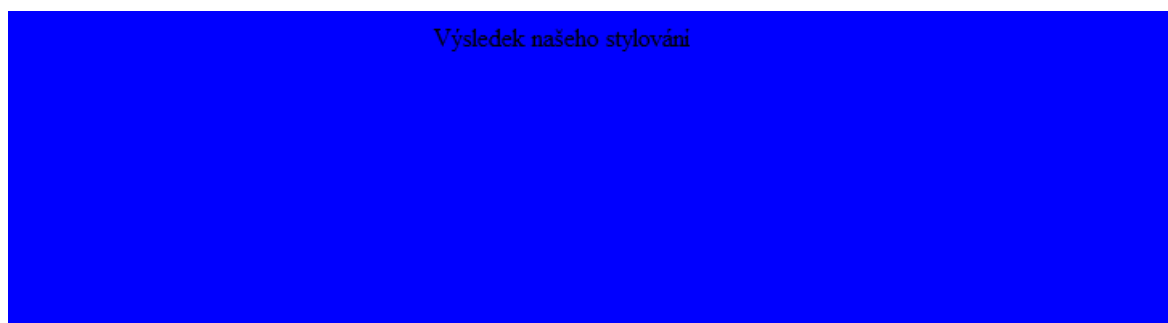
Ukázka kaskádových stylů:

```
body {
height: 100%;
text-align: center;
background: blue;
}
```

body – selektor kterým si vybíráme kterou část html dokumentu chceme graficky upravit.

{ } – ohraničení bloku deklarací.

text-align: center – identifikátor vlastností kterým říkáme, co chceme provést s vybraným blokem kódu.



Obrázek 8 – Výsledek CSS stylování

Pro celé tělo stránky (body) jsme nastavili 100% velikost v prohlížeči, text jsme zarovnali na střed a pozadí jsme nastavili na modré.

2.4 PHP

PHP (*Hypertext Preprocessor*, česky PHP: Hypertextový preprocesor“) je skriptovací programovací jazyk. Je určený především pro programování dynamických internetových stránek a webových aplikací například ve formátu HTML, XHTML či WML. PHP lze použít i k tvorbě konzolových a desktopových aplikací. Pro desktopové použití existuje kompilovaná forma jazyka. [9][10][11]

Při použití PHP pro dynamické stránky jsou skripty prováděny na straně serveru k uživateli, je přenášén až výsledek jejich činnosti. PHP skript lze vložit přímo do HTML mezi značky `<?php ... ?>`. Aby PHP fungovalo, vytvořená stránka musí mít koncovku .php. Výhodou je, že zobrazí jen tu část, která je požadována. [9][10][11]

Interpret PHP skriptu je možné volat pomocí příkazového řádku, dotazovacích metod HTTP nebo pomocí webových služeb. Syntaxe jazyka je inspirována několika programovacími jazyky (Perl, C, Pascal a Java). PHP je nezávislý na platformě, rozdíly v různých operačních systémech se omezují na několik systémově závislých funkcí a skripty lze většinou mezi operačními systémy přenášet bez jakýchkoli úprav. [9][10][11]

PHP je nejrozšířenějším skriptovacím jazykem pro web. Oblíbeným se stal především díky jednoduchosti použití, bohaté zásobě funkcí. V kombinaci s operačním systémem Linux, databázovým systémem (obvykle MySQL nebo PostgreSQL) a webovým serverem Apache je často využíván k tvorbě webových aplikací. Pro tuto kombinaci se vžila zkratka LAMP – tedy spojení Linux, Apache, MySQL a PHP, Perl nebo Python. [9][10][11]

Ukázka výpisu textu na stránky:

```
<?php
echo ("Vypíše se jen obsah v uvozovkách")
?>
```

2.5 JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape. Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, často vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků. Jeho syntaxe patří do rodiny jazyků C/C++/Java. Slovo Java je však součástí jeho názvu pouze z marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe. JavaScript byl v červenci 1997 standardizován asociací ECMA (European Computer Manufacturers Association) a v srpnu 1998 ISO (International Organization for Standardization). Standardizovaná verze JavaScriptu je pojmenována jako ECMAScript a z ní byly odvozeny i další implementace, jako je například ActionScript. JavaScript byl původně obchodní název implementace společnosti Netscape, kde byl vyvíjen nejprve pod názvem Mocha, později LiveScript, ohlášen byl společně se společností Sun Microsystems v prosinci 1995 jako doplněk k jazykům HTML a Java. Pro verzi firmy Microsoft je použit název JScript. Ten je podporován platformou .NET. [12][13][14]

Program v JavaScriptu se obvykle spouští až po stažení WWW stránky z Internetu (tzv. na straně klienta), na rozdíl od ostatních jiných interpretovaných programovacích jazyků (např. PHP a ASP), které se spouštějí na straně serveru ještě před stažením z Internetu. Z toho plynou jistá bezpečnostní omezení, JavaScript např. nemůže pracovat se soubory, aby tím neohrozil soukromí uživatele. JavaScript je možné použít i na straně serveru. První implementací JavaScriptu na straně serveru byl LiveWire firmy Netscape vypuštěný roku 1996, dnes existuje několik možností včetně opensource implementace Rhinola založené na Rhino, gcj, Node.js a Apache. [12][13][14]

Kromě DHTML se JavaScript používá k psaní rozšíření pro mnohé aplikace, například Adobe Acrobat. [12][13][14]

JavaScript je také možno spouštět v operačních systémech Windows pomocí programu Windows Script Host a nahradit tak dávkové soubory MS-DOS. [12][13][14]

JavaScript je možné vložit přímo do HTML kódu nebo načíst externí soubor.

Ukázka externího vložení JavaScriptu:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv='Content-Type' content='text/html; charset=utf-8' />
<meta http-equiv='Content-language' content='cs' />
  <title>Titulek stránky</title>
<!-- Načítání externího souboru javascript.js -->
<skript type="text/javascript" src="javascript.js"></skript>
</head>
<body>
<!-- Obsah stránky který se zobrazuje -->
</body>
</html>
```

Ukázka zápisu JavaScriptu přímo do HTML:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv='Content-Type' content='text/html; charset=utf-8' />
<meta http-equiv='Content-language' content='cs' />
  <title>Titulek stránky</title>
  <!-- Načítání externího souboru javascript.js -->
  <skript type="text/javascript" language="javascript">
  <!-- javascriptový kód -->
  </script>
</head>
<body>
  <!-- Obsah stránky který se zobrazuje -->
  <skript type="text/javascript" language="javascript">
  <!-- javascriptový kód -->
  </script>
</body>
</html>
```

2.6 MySQL

MySQL je databázový systém, vytvořený švédskou firmou MySQL AB, nyní vlastněný společností Sun Microsystems, dceřinou společností Oracle Corporation. Jeho hlavními autory jsou Michael Widenius a David Axmark. Je považován za úspěšného průkopníka dvojího licencování – je k dispozici jak pod bezplatnou licenci GPL, tak pod komerční placenou licenci. [15][16]

MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními. [15][16]

Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, je používán v mnoha používaných databázích. Velmi oblíbená a často nasazovaná je kombinace Linux, MySQL, PHP a Apache jako základní software webového serveru. [15][16]

MySQL byl od počátku optimalizován především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporoval pohledy, trigger, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, kdy začaly tyto funkce nejčastějším uživatelům produktu – programátorům webových stránek – poněkud scházet. [15][16]

Ukázka práce z databází a připojení pomocí MySql:

```
<body>
<?php
    $spojeni = mysql_connect("localhost", "root") //nastavení spojení
se serverem
    or die("Nelze se připojit"); //v případě že se nezdaří navázat
spojení
    print ("Spojení navázáno"); //spojení bylo úspěšně navázáno
```

```
//ukázka vkládání dat do databáze
$sql = "INSERT INTO `galerie`.`fotky` (`Nazev`, `ID_slozky`) VALUES (
`Ukládám do databáze`, `1`)";

MySQL_DB_Query("galerie" , $sql , $spojeni); //odeslání požadavku serveru
$chyba = MySQL_Error(); //v případě chyby
```

2.7 jQuery

jQuery je rychlá a bohatá javascriptová knihovna, která klade důraz na interakci mezi JavaScriptem a HTML. Byla vydána Johnem Resigem v lednu 2006 na newyorském BarCampu. jQuery je svobodný a otevřený software pod duální licencí MIT a GPL. (General Public License). jQuery je syntaxe navržena tak, aby usnadnila orientaci v dokumentu pomocí, zvoleného DOM prvku, můžete vytvářet animace, zpracovávat události, a vyvíjet aplikace pomocí AJAXU. jQuery také poskytuje možnosti pro vývojáře, jak vytvořit vlastní plug-iny v horní části knihovny JavaScript. To umožňuje vývojářům vytvářet abstrakce na nízké úrovni interakce a animace, pokročilé efekty a na vysoké úrovni. Modulární přístup k jQuery knihovnám umožňuje vytvoření silných dynamických webových stránek a webových aplikací. [17]

2.7.1 Možnosti načtení JQuery knihovny

2.7.1.1 *Jako jeden javascriptový soubor, obsahujícího všechny funkce pro DOM, Ajax, události a efekty.*

Ukázka načtení JQuery ze souboru:

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

2.7.1.2 *Načtení pomocí Google AJAX Libraries API.*

Tento způsob získávání knihovny má mnoho výhod včetně unifikovaného cejchování a snížení odezvy.

Ukázka načtení JQuery z Google AJAX Libraries API:

```
<script
type="application/javascript"src="http://www.google.com/jsapi"></script>
<script type="application/javascript">
Google.load("jquery","1.3.2");
</script>
```

2.7.1.3 *Načítání jQuery přímo ze serverů Google*

Ukázka načtení JQuery z Google serveru:

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

2.7.1.4 Načtení JQuery přímo z domovské stránky projektu

Ukázka načtení JQuery z domovské stránky projektu:

```
<script type="application/javascript" src="http://code.jquery.com/jquery-
latest.min.js">
</script>
<script type="text/javascript" src="jquery-1.6.2.min.js"></script>
```

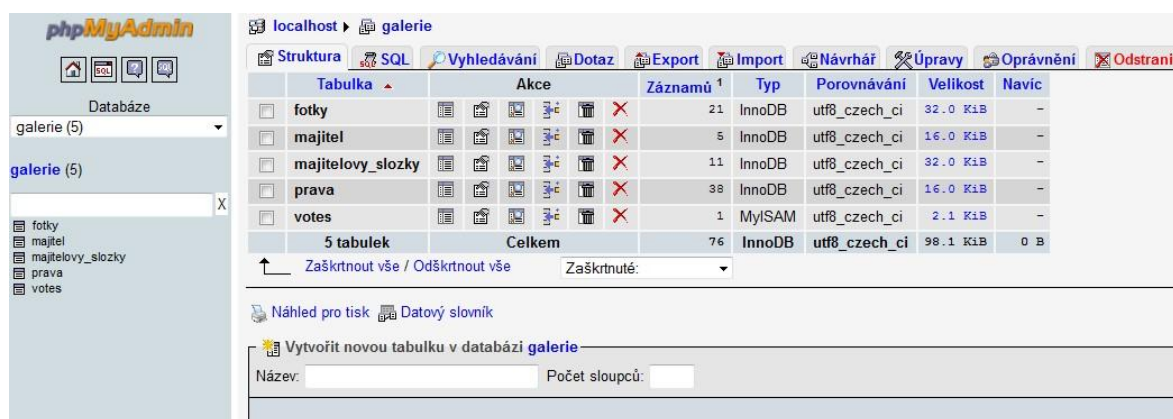
Ukázka práce z JQuery(spuštění funkce po kliknutí myší):

```
<script type="text/javascript" language="javascript">
$(document).ready(function() {
$("#registrace").click(function(){
$("#vypis").load('login/registrace.php');
});
});
</script>
```

3 POUŽITÉ PROGRAMY A PŘÍSTUP K PRÁCI

3.1 PHPMyAdmin

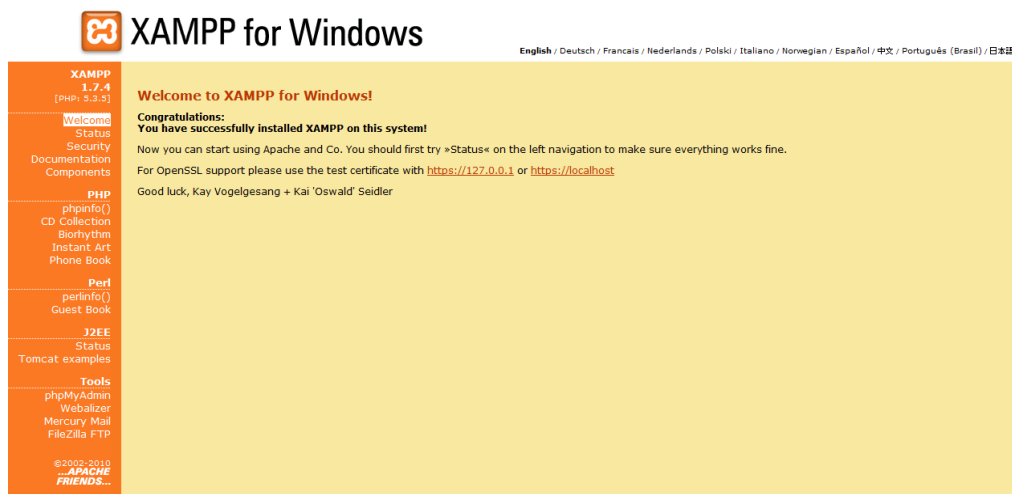
Programový systém phpMyAdmin je nástroj napsaný v jazyce PHP umožňující jednoduchou správu obsahu databáze MySQL prostřednictvím webového rozhraní. V současné době umožňuje vytvářet/rušit databáze, vytvářet/upravovat/rušit tabulky, upravovat/mazat /přidávat záznamy do tabulek, provádět SQL příkazy a spravovat klíče. PhpMyAdmin funguje přes internetové rozhraní. Jedná se o jeden z nejpopulárnějších nástrojů pro správu databáze. Je k dispozici v 57 jazycích.[18]



Obrázek 9 – Ukázka pracovního prostředí phpMyAdmin

3.2 XAMPP

XAMPP je zdarma a open source multiplatformní webový server balík, skládající se hlavně z Apache, HTTP Server, MySQL databáze, a zprostředkovatelem pro skripty napsané v PHP a Perl programovacím jazyce. Program je vydán pod GNU (General Public Licence) a působí jako volný webový server schopen obsluhovat dynamické stránky. XAMPP je k dispozici pro Microsoft Windows, Linux, Solaris a Mac OS X, a používá se zejména pro projekty, vývoj webových aplikací.[19]



Obrázek 10 – Ukázka pracovního prostředí XAMPP

3.3 PSPad

PSPad editor je volně šiřitelný (freeware) univerzální editor pro MS Windows. PSPad využijí všichni, kteří:

- pracují s prostým textem - velké možnosti formátování textu
- vytvářejí webové stránky - obsahuje řadu unikátních funkcí, které ušetří spoustu času

programují a potřebují IDE pro svůj kompilátor - odchytávání a parsování výstupu kompilace, integrace helpu.[20]

3.4 Gimp

GIMP neboli GNU Image Manipulation Program („GNU program pro úpravy grafiky“) je svobodná multiplatformní aplikace pro úpravu a vytváření rastrové grafiky. Používá se zejména pro úpravy fotografií, tvorbu webové grafiky a podobné účely. Kromě široké škály rastrových nástrojů obsahuje i některé vektorové funkce, které jsou užitečnou pomůckou při práci s rastrovou grafikou (cesty, písma atd.). GIMP je dnes oficiální součástí projektu GNU. GIMP je dostupný zdarma včetně zdrojových kódů pod licencí GPL.[21][22][23]

3.5 Ukládání dat a tvorba stromové struktury

Stromové datové struktury představují pro relační databáze poměrně těžko zpracovatelný problém. Problémy vyplývají ze samotné podstaty relačního databázového modelu, který není pro ukládání tohoto typu dat příliš vhodný. Dosáhnout toho, aby byla práce s těmito strukturami efektivní, představuje nelehký úkol. [24]

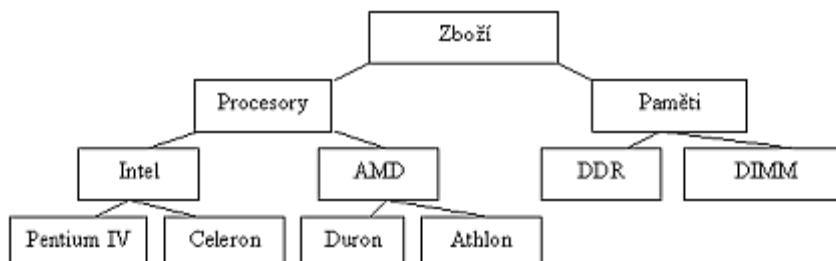
3.5.1 Úskalí relačních databází

Relační databázové systémy stále zůstávají nejpoužívanějším typem databázových systémů, i když máme již několik let k dispozici v mnoha ohledech pokročilejší objektové databáze. Jedna z největších výhod objektových databází je snadná práce se složitě strukturovanými daty. Naopak relační databáze využívající k ukládání dat plochých relačních tabulek mají s těmito daty značné potíže. [24]

Konkrétním případem obtížně zpracovatelných dat jsou data s hierarchickou, respektive stromovou strukturou. V objektové databázi mohou být stromová data uložena přímo v takové podobě, jakou využívá aplikace, která se k této databázi připojuje. Naopak při použití relační databáze musíme data transformovat tak, abychom je mohli uložit do ploché relační tabulky, a při čtení dat z databáze je musíme zpětně transformovat do podoby stromu. [24]

3.5.2 Úvod

Velká pozornost je věnována datové struktuře a algoritmům, kterými se mohou stromová data zpracovávat (získávat a editovat). Příklady jsme realizovali v jazyce PHP pro jeho jednoduchost, úspornost a rozšířenost. Pro ukázkou je vybrán strom z e-shopu. [24]



Obrázek 11 – Stromová struktura

3.5.3 Klasický přístup

Nejznámějším a také nejčastěji využívaným způsobem, který lze při ukládání stromových struktur do relační databáze použít, je model, kdy je součástí každého prvku stromu také reference na prvek rodičovský. Nejvýše postavený prvek stromu, zvaný kořen, má referenci nastavenou na 0 nebo NULL. [24]

ID	NAME	PARENT_ID
1	Zboží	0
2	Procesory	1
3	Intel	2
4	Pentium IV	3
5	Celeron	3
6	AMD	2

Obrázek 12 – Klasická datová struktura pro uložení stromu v relační tabulce

3.5.3.1 Úprava dat

Editace dat je v tomto případě naprosto triviální. Při ukládání nového uzlu stačí znát ID nadřazené kategorie. Přesun větve do jiné části stromu je také velice jednoduchý, stačí pouze změnit PARENT_ID daného uzlu. [24]

3.5.4 Rozšíření ploché tabulky

Pro zvýšení efektivity modelu z předchozí části, můžeme datovou strukturu rozšířit o další atributy, které nám umožní rychlejší přístup k datům. Bude to atribut ORD, který představuje pořadí uzlu v daném stromu, a atribut LEVEL, který představuje zanoření, respektive úroveň uzlu. [24]

ID	NAME	PARENT_ID	ORD	LEVEL
1	Zboží	0	1	0
2	Procesory	1	2	1
3	Intel	2	3	2
4	Pentium IV	3	4	3
5	Celeron	3	5	3
6	AMD	2	6	2

Obrázek 13 – Rozšíření klasického přístupu o pořadí a hloubku zanoření

3.6 Druhy šifrování hesel pro databáze

Údaje o uživateli bývají uloženy na serveru v nějaké databázi. Součástí údajů je uživatelské jméno, informace o hesle a další informace, podle toho, co aplikace vyžaduje. [25]

Mnohem lepší variantou než ukládat samotné heslo je ukládání hesla v podobě *hashe* – „otisku“, který je výsledkem speciální matematické funkce. Obecně tyto funkce pracují tak, že berou vstupní data a k nim vrací řetězec, který má některé specifické vlastnosti: [25]

1. pro stejná vstupní data je stejný otisk,
2. má konstantní délku,
3. drobná změna vstupních dat vyvolá velkou změnu ve výsledku,
4. je prakticky nemožné nalézt různá vstupní data se stejným hashem
5. z výsledného řetězce je v praxi nemožné rekonstruovat původní text.

Hashovacích funkcí je mnoho, ale ve webových aplikacích jsou nejpoužívanější funkce z rodiny MD (téměř výhradně MD5, MD4) a SHA (SHA1, SHA2). U MD5 byly nalezeny chyby v návrhu, které snižují jeho bezpečnost, takže mnozí tvůrci přechází na SHA algoritmy. Je na místě podotknout, že SHA1 byl rovněž označen za slabší a doporučeno je používat algoritmy SHA2. [25]

3.6.1 MD

Message-Digest algorithm je v kryptografii rodina hašovacích funkcí, která z libovolného vstupu dat vytváří výstup fixní délky, který je označován jako hash (česky někdy psán i jako haš), otisk, miniatura a podobně (anglicky fingerprint). Jeho hlavní vlastností je, že malá změna na vstupu vede k velké změně na výstupu, tj. k vytvoření zásadně odlišného otisku. [26][27][28][29]

3.6.1.1 MD4

MD4 je jednou ze série kryptografických hašovacích funkcí, které navrhl profesor Ronald L. Rivest pracující v institutu MIT (Rivest, 1994). Později byly v hašování nalezeny nedostatky panem Hansem Dobbertinem. [26][27][28][29]

3.6.1.2 MD5

Algoritmus MD5 se prosadil do mnoha aplikací (např. pro kontrolu integrity souborů nebo ukládání hesel). MD5 je popsán v internetovém standardu RFC 1321 a vytváří otisk o velikosti 128 bitů. Byl vytvořen v roce 1991 Ronaldem Rivestem, aby nahradil dřívější hašovací funkci MD4. Dobbertin v roce 1996 oznámil kolizi kompresní funkce MD5 (Dobbertin, 1996). Zatímco to nebyl útok na kompletní MD5 hash funkci, bylo to dost podstatné pro kryptografy, aby doporučili přechod na náhradu, například SHA-1 nebo RIPEMD-160. [26][27][28][29]

3.6.2 SHA

SHA (Secure Hash Algorithm) je rozšířená hašovací funkce, která vytváří ze vstupních dat výstup (otisk) fixní délky. Otisk je též označován jako miniatura, kontrolní součet (v zásadě nesprávné označení), fingerprint, hash (česky někdy psán i jako haš). Jeho hlavní vlastností je, že malá změna na vstupu vede k velké změně na výstupu, tj. k vytvoření zásadně odlišného otisku. [29][30]

SHA navrhla organizace NSA (Národní bezpečnostní agentura v USA) a vydal NIST (Národní institut pro standardy v USA) jako americký federální standard (FIPS). SHA je rodina pěti algoritmů: SHA-1, SHA-224, SHA-256, SHA-384 a SHA-512. Poslední čtyři varianty se souhrnně uvádějí jako SHA-2. SHA-1 vytvoří obraz zprávy dlouhý 160 bitů. Číslo u ostatních čtyř algoritmů značí délku výstupního otisku v bitech.

SHA se používá u několika různých protokolů a aplikací, včetně TLS a SSL, PGP, SSH, S/MIME a IPsec, ale i pro kontrolu integrity souborů nebo ukládání hesel. Je považována za nástupce hašovací funkce MD5. [29][30]

3.6.2.1 SHA 0 a SHA 1

Původní specifikace algoritmu byla publikována v roce 1993 jako Secure Hash Standard (FIPS PUB 180) vedením americké normalizační agentury NIST. Tato verze je dnes známá jako SHA-0. NSA jí stáhla krátce po vydání a nahradila ji upravenou verzí, vydanou v roce 1995 (FIPS PUB 180-1) obvykle uváděnou jako SHA-1. SHA-1 se liší od SHA-0 pouze jednou bitovou rotací provedenou pomocí jednocestné funkce. Změna byla udělána podle NSA. Oprava vady v původním algoritmu snižuje šifrovací bezpečnost. NSA neposkytla žádné další objasnění nebo vysvětlení, jaká vada byla opravena. Slabé stránky byly následovně ohlášeny v SHA-0 i v SHA-1. SHA-1 se zdá být více obranyschopná proti útokům, to podporuje tvrzení NSA, že změnami stoupá bezpečnost. [29][30]

SHA-1 (stejně jako SHA-0) vytváří 160 bitový obraz zprávy s maximální délkou $2^{64} - 1$ bitů. Je založený na principech, které používal Ronald L. Rivest z Massachusetts Institute of Technology (MIT) v návrhu MD4 a MD5 algoritmů. [29][30]

3.6.2.2 SHA 2

NIST zveřejnila čtyři další hashovací funkce SHA, které jsou pojmenovány podle své délky (v bitech): SHA-224, SHA-256, SHA-384 a SHA-512. Algoritmy jsou společně označovány jako SHA-2. [30] [31]

Algoritmy byly poprvé zveřejněny v roce 2001 v návrhu standardu FIPS PUB 180-2, který obsahoval i SHA-1 a byl vydán jako oficiální standard v roce 2002. V únoru 2004 byla zveřejněna změna, která definovala další variantu SHA-224, která odpovídá délkou klíče dvou-klíčového Triple DES. Tyto varianty jsou patentovány v patentu US 6829355 a Spojené státy je uvolnily k použití bez licenčních poplatků. [30] [31]

SHA-2 není široce používáno i přes lepší zabezpečení (ve srovnání se SHA-1). Důvodem může být nedostatek podpory pro SHA-2 na systémech Windows XP SP2 a starších, nedostatečné vnímání naléhavosti přejít na SHA-2, protože kolize SHA-1 ještě nebyly nalezeny, nebo čekání na standardizaci SHA-3. SHA-256 se používá k ověřování softwarových balíčků linuxové distribuce Debian. SHA-256 a SHA-512 jsou navrženy pro použití v DNSSEC. Rovněž datové schránky v České republice používají SHA-2. Podle směrnice NIST americké vládní agentury přestaly používat SHA-1 v roce 2010. Také dokončení standardizace SHA-3 mohou urychlit odchod od SHA-1. [30] [31]

3.6.3 salted hash

Tato možnost není zas až tolik odlišná od předchozí, je však trochu odolnější proti brute-force útokům. Spočívá v tom, že k heslu připojíte náhodnou posloupnost znaků, takzvaný **salt**. Vzniklý řetězec pak teprve hašujete. Útočník tak nemůže použít již předem hašovaný slovník, ale musí slova ze svého slovníku spojit se saltem a teprve potom hašovat. Tento postup ho alespoň o něco zpomalí, nikoli však zastaví. Spíše, řekl bych, odradí.

Autoři vesměs zastávají postoj, že salt jako takový není nic tajného a může se klidně uložit do databáze jako další pole. Tajné je však to, jak je salt použit (jestli je připojen na začátek, na konec, za druhý znak a podobně). [35] [36]

3.7 Přenos hesel na server

3.7.1 HTTPS

HTTPS je šifrovanou variantou internetového protokolu HTTP pro přenos webových stránek. Zkratka HTTPS pochází z anglických slov *HyperText Transfer Protocol Secure*.

Protokol HTTPS umožňuje **chráněný přístup** k webovému serveru tím, že veškerou přenášenou komunikaci šifruje algoritmem SSL nebo TLS. To je důležité při přenášení citlivých informací (např. čísla kreditní karty). HTTPS se také používá k **autorizaci přístupu** k webu. [37][38]

Pro zvýšení bezpečnosti vyžaduje prohlížeč komunikující přes HTTPS tzv. **certifikát**. Ten buď může být podepsán tzv. **certifikační autoritou** zaručující pravost certifikátu, nebo si jej vlastník serveru může vydat sám. V takovém případě pak prohlížeč zobrazí uživateli varování. [37][38]

3.7.2 hash na straně klienta

Technika **výzva-odpověď** funguje tak, že server pošle klientovi výzvu, klient k této výzvě připojí své heslo a serveru pošle otisk tohoto spojení. Server na své straně provede totéž, a pokud výsledky odpovídají, tak uživatele přihlásí, jinak ho odmítne. Bezpečnost tohoto řešení je založena na tom, že server každou výzvu posílá jen jednou a pokud se útočníkovi podaří odpověď klienta zachytit, k ničemu mu to neposlouží, protože stejnou výzvu už server nikdy nepošle. [39]

3.8 Chyby na webu

3.8.1 SQL Injection

SQL Injection je bezpečnostní chyba založená na možnosti manipulovat s daty v databázi bez nutnosti mít k nim legitimní přístup. Na první pohled by se mohlo zdát, že tato chyba je problémem webových technologií. Opak je pravdou. SQL Injection je problémem všech aplikací pracujících s databází. Zneužití může vést k získání citlivých údajů, jakými jsou přihlašovací údaje, osobní údaje (rodná čísla, čísla bankovních účtů.) a v některých případech může vést k vykonání systémového příkazu, případně k ovládnutí celého serveru/počítače. Principem je vkládání nových/rozšiřujících SQL dotazů do již existujících SQL dotazů. [40][41][42]

3.8.2 Cross-site scripting(XSS)

Cross-site scripting (XSS) je metoda narušení WWW stránek využitím bezpečnostních chyb ve skriptech (především neošetřené vstupy). Útočník díky těmto chybám v zabezpečení webové aplikace dokáže do stránek podstrčit svůj vlastní javascriptový kód, což může využít buď pouze k poškození vzhledu stránky, jejímu znefunkčnění nebo dokonce k získávání citlivých údajů návštěvníků stránek. [43][44]

1. Stored (persistent, second-order)

Patří k nejnebezpečnějším XSS útokům. Oběť na napadené stránky nemusí vstupovat přes upravený link. Skript je generován přímo z databáze. Útočník skript do databáze může vložit např. přes komentáře v diskusních fórech, ale i přímým vložením do databáze (např. přes SQL Injection). Pokud je skript načítán z databáze, pak je ohrožen každý, kdo navštíví napadenou stránku. [43][44]

2. Reflected (non-persistent, okamžitý)

Tento velmi běžný typ XSS útoku se projevuje okamžitým vykonáním útočného skriptu. Webová aplikace zobrazí výsledek okamžitě po odeslání požadavku, skript se nikam neukládá. V praxi se běžně využívá parametrů v URL, které se nahradí nebo doplní útočnou konstrukcí skriptu. Pokud na takové URL oběť klikne, vykoná se útočný skript a útočník může např. zcizit údaje uložené v uživatelské prohlídce. [43][44]

Ukázka XSS útoku který se projeví okamžitě:

```
<?php echo $_GET['nadpis']; ?>
```

stačí uživateli podstrčit url upravenou například takto:

```
http://URL/stranka.php?nadpis=cokoliv<script>alert('Toto je úspěšný XSS útok.');
```

3. DOM (Document Object Model) based XSS (lokální)

Je velmi podobný reflected XSS. Avšak k útoku je využit existujícího klientského skriptu – přenesením (např. z URL.) kódu do skriptu stránky (viz. Scénář 3. uvedený níže). Lokální

skripty umožňují poměrně nebezpečné útoky (volání `document.location`, `document.URL`, `document.referrer` a dalších DOM metod v těle skriptu), neboť některé prohlížeče považují lokální skripty za bezpečné. [43][44]

Je nutné poznamenat, že XSS slabiny nesouvisí pouze s JavaScriptem. XSS slabiny můžeme nalézt i v technologiích, jako jsou ActiveX, Flash, VBScript, HTML, Java – při kategorizaci XSS s tím bývá někdy docela zmatek. [43][44]

a na stránku vstoupíme přes standardní link:

```
http://URL/stranka.html?jmeno=Alice
```

stránka standardně vypíše pouze „Ahoj Alice“. Útočník však může link pozměnit na:

```
http://URL/stranka.html?jmeno=<script>alert('Toto je úspěšný XSS útok.');
```

a tím vykoná zákeřný kód.

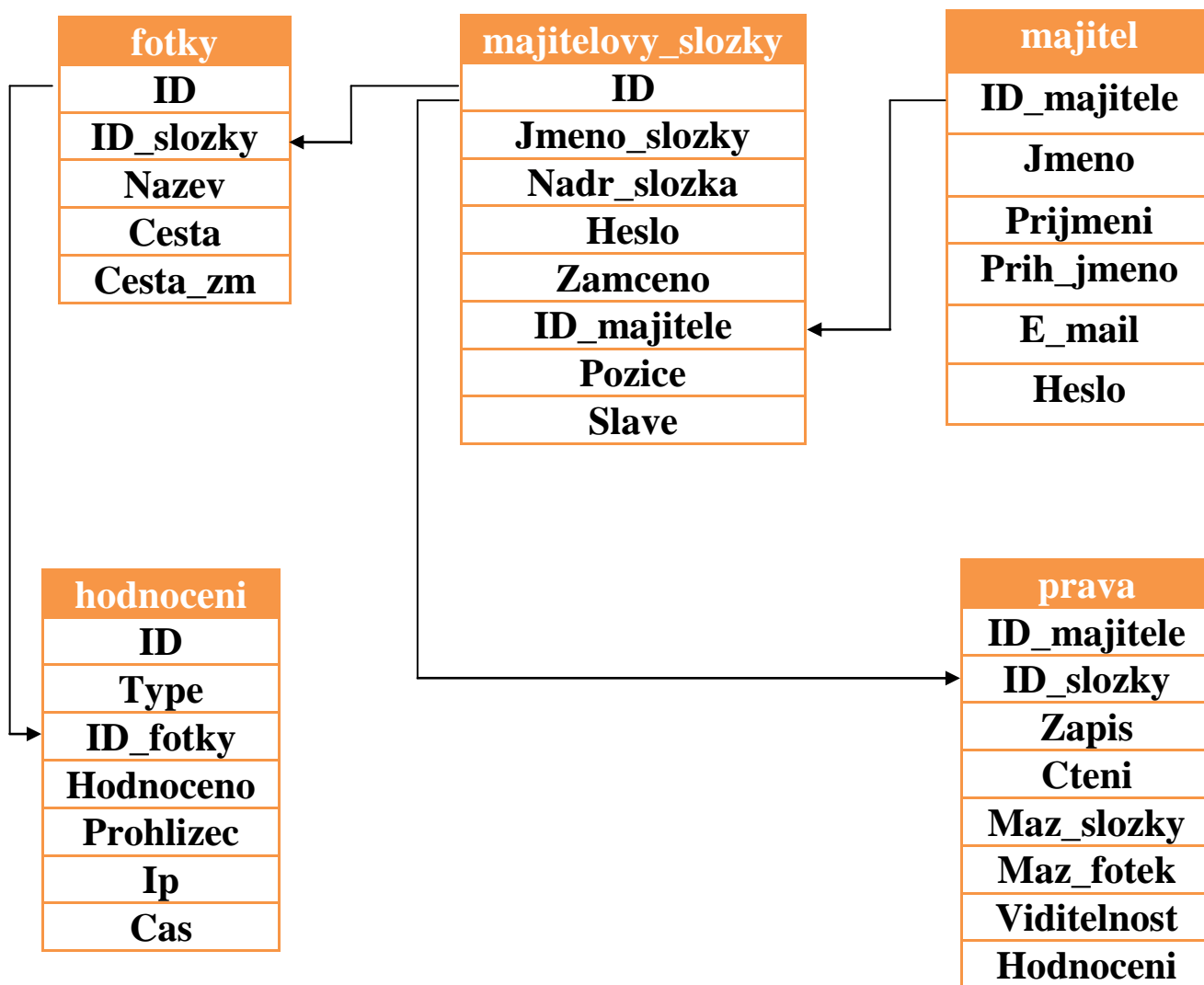
II. PRAKTICKÁ ČÁST

1. NÁVRH

Důležitou prací při vytváření stránek bylo samotné rozvrhnutí stránky a postupu práce.

1.1 Návrh databáze

V první fázi vývoje bylo důležité promyslet a vytvořit celou databázi, která bude s danou galerií spolupracovat. Důležité taky bylo promyslet, jakým způsobem bude potřeba jednotlivé tabulky propojit.



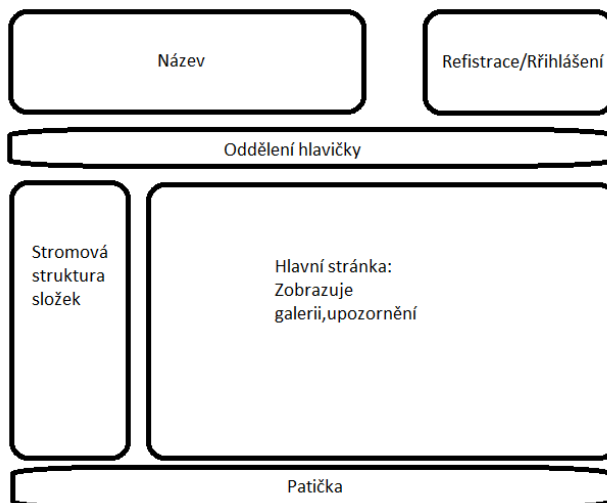
Obrázek 14 – Návrh DB

Tabulka *majitel* slouží k ukládání údajů registrovaných uživatelů, jejich přihlašovací jména, hesla, emaily a další potřebné údaje.

Tabulka *majitelovy_slozky* je asi nejzajímavější. V této tabulce jsou zde uložena všechna potřebná data pro stromovou strukturu. Bylo vytvořeno spojení 1:N s tabulkami *fotky* a *prava*. *ID* je ve tvaru *AUTO_INCREMENT* a to z toho důvodu, aby nedošlo později k situaci, že dvě nebo více složek budou mít stejné ID. Tabulce byl přidělen primární klíč z důvodu ukládání velkého množství dat, kde je možné díky oddělenému souboru rychlejší vyhledávání. *Jmeno_slozky* udává jména, která byla vyplněna uživateli. *Nadr_slozka* uchovává ID nadřazené složky ve stromové struktuře, aby se při výpisu zaručila správná návaznost složek na sebe. *Heslo* a *Zamceno* ještě nebyly využity. *ID_majitele* udává, který z uživatelů danou složku vytvořil. Později tohoto parametru bylo využíváno ke zjištění a přidělení práv samotnému majiteli. Sloupec *Pozice* udává umístění složky ve stromové struktuře, tedy říká, jestli je složka jako jediná nebo je ve stejné úrovni další složka. A díky tomuto parametru, lze poznat pořadí složek, jak byly vytvořeny a v jaké úrovni se nachází. Tabulka *fotky* ukrývá záznam o veškerých fotografiích, které se nahrávají do galerie. ID je opět specifické a neopakuje se, aby nedošlo k tomu, že fotky by měly stejné ID a tedy by se jedna z nich ztratila. *ID_slozky* je důležité ke správnému zobrazení fotografií. Uchovává údaj, do které složky byly fotografie nahrány. *Cesta_zm* a *Cesta* zaznamenávají cestu, kde je fotka fyzicky uložena pro pozdější načítání v galerii.

1.2 Návrh rozložení stránky

Byl vytvořen později a zobrazuje rozložení webové stránky. Rozložení stránky lze změnit zásahem do hlavního souboru s kaskádovými styly.



Obrázek 15 – Rozložení stránky

2. TVORBA

2.1 Registrace/Přihlášení

Přihlašování a registrace je jedním z důležitých prvků celé práce. Celý proces je složen z pěti samostatných souborů. Prvním je *index.php*, který na začátku spustí session a ověří, zda je nějaký uživatel již registrován.

Ukázka spuštění session a kontrola zda je uživatel přihlášen:

```
session_start(); //start session
header("Cache-control: private");
if (!isset($_SESSION['Uziv']) or $_SESSION['Uziv']== "")
```

Tahle část je důležitá především proto, že kdyby uživatel provedl přenačtení stránky, tak by došlo bez tohoto ověření k odhlášení, a bylo by potřeba znovu zadávat heslo a přihlašovací jméno. Dále bylo v souboru vytvořeno přihlašování pomocí formuláře a textových polí, do kterých je možno zadávat uživatele a heslo. Pod těmito okny se nachází tlačítka pro přihlášení, či registraci. Ve spodní části kódu byla vytvořena část, která kontroluje, zda je uživatel přihlášen. V případě, že se přihlášení provedlo, bude zobrazeno jeho přihlašovací jméno a možnost k odhlášení. Pokud přihlášení není provedeno správně, bude zobrazeno upozornění. Stiskem tlačítka *přihlásit se*, budou veškeré údaje odeslány pomocí ajaxu na další stránku (*login.php*), kde odeslaná data budou přijata a zkontrolována funkcí `mysql_real_escape_string()` pro případ, že by došlo k pokusu do proměnných vložit nekorektní znaky. Tato funkce napomáhá při ochraně před SQL Injection. Po převzetí hesla bude provedeno zašifrování pomocí SHA2.

Ukázka zahašování hesla:

```
$SHA2heslo = hash("sha256", "$heslo");
```

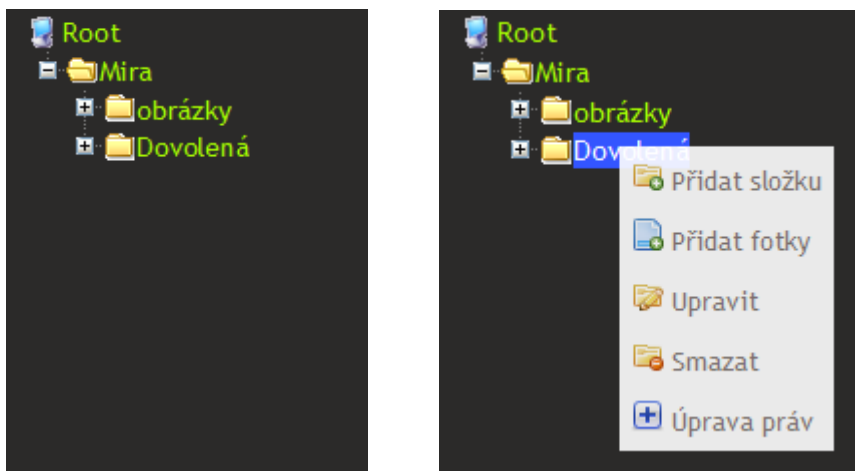
Hešování bude provedeno ještě na straně klienta a následně bude odesláno již v zahašované podobě, aby se předešlo případnému odcizení údajů. Při přihlašování bude provedena kontrola, a pokud bude vše souhlasit se záznamem v databázi, bude provedeno přihlášení uživatele.

V případě, že bude stisknuto tlačítko *registrace*, tak bude načtena stránka *registrace.php* do předem připraveného divu, kde bude zobrazen klasický formulář k vyplnění. Po případném vyplnění všech údajů, budou data odeslána opět přes ajax. Po odeslání bude provedena kontrola, zda byly všechny položky vyplněny. Pokud k tomu nedošlo, bude zobrazeno upozornění k nápravě. Vyplněná data se po odeslání neztratí, ale zůstanou vyplněna v textových oknech. V případě, že vše proběhne správně, tak bude heslo opět zabezpečeno a uloženo do databáze spolu s ostatními údaji. Další ošetření je provedeno u emailu, kde dochází ke kontrole, zda je zadán správně ve formě *uživatel@doména*. Následná kontrola bude provedena u hesla, kde je nastavena minimální délka a taky bude provedena kontrola dalším vyplněním pole a porovnáním hesla.

Posledním souborem je *logout.php*, kde dochází po stisku odkazu *odhlásit se*, k vyprázdnění session a k přesměrování stránky na hlavní soubor *index.php*

2.2 Stromová struktura

Dalším prvkem galerie je samotná stromová struktura, kde bylo využito toho, že struktura byla již navržena a zpracována.[45] Na začátku bylo nutno upravit kód stromové struktury tak, aby bylo dosaženo spolupráce s již navrženou databází.



Obrázek 16 – Stromová struktura a rozbalovací menu

Dalším krokem bylo upravení zobrazovacího menu tak, aby bylo zobrazováno jen to, co bylo potřeba. Zde bylo potřeba zcela upravit předešlý kód.

Ukázka zobrazovacího menu:

```
langManager.load("cz");
var treeOps = new TreeOperations();
$(document).ready(function() {
    // výběr funkcí v menu
    $('#myMenu1.addFolder').click(function() { treeOps.addElementReq(true); });
    $('#myMenu1.addDoc').click(function() { treeOps.addElementFt(); });
    $('#myMenu1.edit, #myMenu2.edit').click(function() { treeOps.updateElementNameReq(); });
    $('#myMenu1.delete, #myMenu2.delete').click(function() { treeOps.deleteElementReq(); });
    $('#myMenu1.addLaw').click(function() { treeOps.addLaw(); });
    $('#myMenu1.deleteLaw').click(function() { treeOps.deleteLaw(); });

    // texty přiřazené k operacím
    $('#myMenu1.addDoc').append(langManager.addDocMenu);
    $('#myMenu1.addFolder').append(langManager.addFolderMenu);
    $('#myMenu1.edit, #myMenu2.edit').append(langManager.editMenu);
    $('#myMenu1.delete, #myMenu2.delete').append(langManager.deleteMenu);
    $('#myMenu1.addLaw').append(langManager.addLaw);
    $('#myMenu1.deleteLaw').append(langManager.deleteLaw);
});
```

Bylo zde zapracováno několik dalších souborů. Jedním z nich je soubor na ověření uživatele, kde bylo potřeba oddělit registrovaného uživatele, od běžného uživatele, který se přišel podívat na fotografie. Později bylo zjištěno, že tohle je nedostačující a bylo potřeba do práce zahrnout určitá práva pro jednotlivé složky a uživatele podle toho, co by majitel přidělil. Proto byla vytvořena položka Úprava práv, která byla přiblížena v kapitole 2.3. Po přidání nových práv bylo nutné, aby došlo po kliku pravým tlačítkem myši na danou složku k zobrazení jen takových možností, která byla uživatelům přidělena od majitele.

Proto byl soubor *over_uzivatele* rozšířen o všechny tyto možnosti práv a výsledkem bylo, že se zobrazují jen funkce, které má daný uživatel v právech přiděleny. Dalším cílem bylo, aby si každý majitel mohl založit pouze jednu podsložku hlavního kořene root. K tomuto byl vytvořen soubor *poc_slozek*, který provádí kontrolu, zda uživatel, který je aktuálně přihlášen, má již jednu podsložku kořene root založenou. Pokud ano, bude zobrazeno upozornění a daná složka se nevytvoří. Další úprava byla provedena pro mazání složek, kde bylo potřeba kontrolovat, zda přihlášený majitel má právo mazat složky. Pokud mu tohle právo bylo přiděleno, dochází ještě ke kontrole, zda daná složka obsahuje nějaké podsložky či fotografie. V takovémto případě bude zobrazeno upozornění, zda i za těchto podmínek si přeje odstranit složku. Pokud bude stisknuto tlačítko *zrušit*, akce se neprovede a vše zůstane. V opačném případě bude odstraněn záznam z databáze a tím i složky a podsložky ve stromové struktuře a případně i fotografie. Další podstatnou funkcí je hromadné nahrávání fotografií, které bude zmíněno v kapitole 2.4. V poslední řadě bylo důležité, aby byly složky zobrazovány uživatelům tak, jak bylo nastaveno v právech u dané složky. Zabezpečení zobrazování bylo vyřešeno pomocí práva viditelnosti u složky, které lze nastavit v právech. Stromová struktura se provádí hned na začátku souboru *index.php*.

Ukázka volání stromové struktury:

```
$rootName = "Root";
$treeElements=$treeManager->
getElementList(null,"strom/manageStructure.php");
```

Byl proveden dotaz na funkci *getElementList*, která provedla připojení k databázi a odeslala samotný dotaz.

Ukázka sql dotazu na databázi:

```
$sql = sprintf("SELECT Id, Jmeno_slozky, slave, Viditelnost FROM
majitelovy_slozky LEFT JOIN prava ON majitelovy_slozky.Id =
prava.Id_slozky
WHERE majitelovy_slozky.Nadr_slozka = %d AND prava.Viditelnost = 1 AND
prava.ID_majitele= 0 ORDER BY Pozice ", $Nadr_slozka);
```

Zde bylo důležité použití funkce *sprintf()*, která neupravuje znaky jako je %, ale namísto toho k nim přiřadí dané číslo. Díky tomu se získají požadované informace a výsledek se odešle zpět na úvodní stránku, kde bude zobrazen výsledek.

Ukázka výpisu stromové struktury na stránky:

```
<ul class="simpleTree" >
    <li class="root" id='<?php echo $treeManager->getRootId();
?>'><span><?php echo $rootName; ?></span>
<ul><?php echo $treeElements; ?></ul>
    </li>
</ul>
```

2.3 Úprava práv

Kvůli nedostatečnému zabezpečení složek a fotografií bylo nutné přidat další tabulku do databáze s dalším rozšířením práv. Bylo přidáno právo čtení, což umožnilo uživatelům si fotografie prohlížet. Právo zápisu, které umožnilo uživatelům vytvářet a upravovat složky a taky přidávat fotografie. Dále bylo přidáno právo mazání složek, což umožnilo uživatelům s přiděleným právem mazat složky i s obsahem (fotografie). Další právo mazání fotografií, dává možnost uživatelům mazat jednotlivé fotografie přímo na stránkách. Právo viditelnosti, umožňuje samotné zobrazení složky ve stromové struktuře. Přidáním práva hodnocení, bylo umožněno uživateli hodnotit zobrazené fotografie. Pro zobrazení a práci s přidávanými právy, bylo nutno přidat tlačítko do zobrazovaného menu ve stromové struktuře. Po kliku na ono tlačítko, bude zobrazena tabulka, kde jsou vypsaní uživatelé, kterým bylo přiděleno u dané složky nějaké oprávnění. Jsou zde zobrazena tlačítka pro možnosti:

-Upravit práva:

Tlačítko umožňuje provést úpravu zobrazených práv. Ta jsou zobrazena pomocí zaškrťovacích políček a lze je jednoduše a efektivně upravovat u zobrazených uživatelů. Po dokončení potřebných úprav a po kliku na tlačítko *Upravit práva* se daná data odešlou a upraví v databázi a zpětně budou zobrazena již upravená.

-Odebrat práva:

Po stisku bude zobrazen select(rozbalovací menu) takových uživatelů, kteří u dané složky mají přiděleno nějaké z práv. Po vybrání uživatele, kterému by měla být práva odebrána a stisku tlačítka *odebrat* dojde ke kontrole. A to proto, aby si majitel složky nemohl odebrat oprávnění sám sobě a tak ztratit možnost kontroly svých dat. Pokud není majitelem, práva se odeberou.

-Přidat práva uživateli:

Tlačítko zobrazí opět nabídku, která obsahuje pouze takové uživatele, kteří doposud žádné právo u dané složky nemají. Tohoto se nedařilo docílit sql dotazem, tak byla využita funkce na porovnání dvou polí.

```
$pole_v1= array_diff($pole1,$pole3);
```

Výsledkem bylo pole uživatelů, kteří zatím žádné právo nemají. Po vybrání jednoho z daných uživatelů lze pomocí zaškrťovacího políčka (checkboxu) lehce a elegantně zaškrtnout právo, které má být přidáno a následným potvrzením práva přidělit.

2.4 Hromadné nahrávání

Dále bylo vytvořeno hromadné nahrávání fotografií. Tam bylo využito již zpracovaného kódu[46] a byly provedeny úpravy tak, aby výsledný kód zapadal do koncepce stránky. Ve stromové struktuře složek bylo opět přepracováno zobrazovací menu, které zobrazuje možnosti práce uživatele s danou složkou. Do menu byla přidána možnost *přidat fotky*, kde po kliku bude zavolána funkce, která provede zjištění, na které složce se aktuálně nachází. Následně bude zobrazen soubor pro nahrávání do již připraveného divu.

Ukázka načtení souboru:

```
var Nadr_slozka= $("span[class='active']").parent().attr('id');
$("#zobraz").load("upload/index.php",{Nadr_slozka: Nadr_slozka});
```

Na stránce se zobrazí tlačítko *Vybrat*, kde po kliku bude zobrazeno okno, kde lze vybrat data pro nahrávání. Po ukončení výběru dojde k zobrazení seznamu námi vybraných dat.



Obrázek 17 – Nahrávání fotografií

Zde bude zobrazena možnost pro zrušení vybraných dat, nebo potvrzení klikem na tlačítko *Uložit*. Při nahrávání fotografií bylo nutno ošetřit, aby dané nahrávané fotografie nebyly až příliš velké a tím nezabíraly zbytečně místo na serveru. Při ukládání byla zároveň vytvořena zmenšenina obrázku.

Ukázka volání funkcí:

```
$img->resize(700, 450, true, false);
$img->save_jpg("uploads/$name");
```

Zavoláním funkce *resize* bude provedeno zmenšení obrázku. A pokud dojde k případu, že obrázek je již dostatečně zmenšen už při nahrávání, tak bude ponechán v původní velikosti a uložen.

Ukázka funkce pro úpravu fotografií:

```
public function resize($width, $height, $aspect_ratio=true, $force=false)
{
    // $force=false zaruci, ze obrazky mensi nez zadane rozliseni
    // nebudou
    // zvetsovany
    if ($width>$this->get_width()&&$height>$this-
    >get_height()&& !$force)
        return $this;

    if ($width==0) $width=$this->get_width();
    if ($height==0) $height=$this->get_height();

    // spocitam nove rozmery
```

```

if ($aspect_ratio) {
    $widthRatio=$this->get_width()/ $width;
    $heightRatio=$this->get_height()/ $height;
    $aspectRatio = $this->get_width()/ $this->get_height();

    if ($widthRatio>$heightRatio)
        $height = $width / $aspectRatio;
    else
        $width = $height * $aspectRatio;
}

// zmenim velikost
$img=imagecreatetruecolor($width,$height);
if(!@imagecopyresampled($img,$this->
>img,0,0,0,0,$width,$height,$this->get_width(),$this->get_height()))
return false;
else {
    $this->img = $img;
    return true;
}
}

```

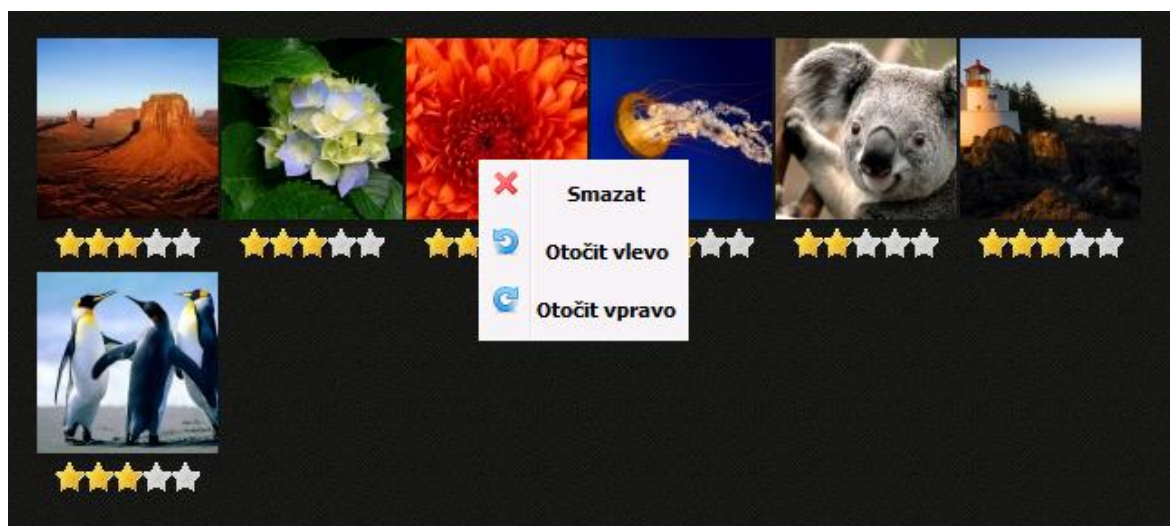
Poté bude uložena cesta jak ke zmenšenině obrázku, tak k samotné fotografii. Ještě před úpravami je potřeba fotografii přesunout na server do předem nastavené a nedefinované složky.

Ukázka přesunu fotografií:

```
move_uploaded_file($tempFile,$targetFile);
```

2.5 Operace s fotografiemi

Byly přidány další základní možnosti pro již zaregistrované uživatele.



Obrázek 18 – Práce s fotografiemi

A to v podobě otočení doleva či doprava a možnost smazání fotografie. Každá z jednotlivých operací je obsažena v samostatném souboru. Ve všech souborech byla na začátku provedena kontrola, zda přihlášený uživatel má oprávnění pro práci s fotografiemi. Pokud ano, bude zavolána funkce potřebná pro provedení dané operace:

```
$img->rotate_left();
$img->save_jpg("../../$obr_vel");
```

Funkce provádějící otočení:

```
public function rotate_left()
{
    $this->img = imagerotate($this->img, 90, 0);
    return true;
}
```

Při provádění otáčení bylo potřeba otočit jak zmenšený, tak i normální obrázek. Smazání bude provedeno o něco složitěji, kde na začátku souboru *smazat_ft.php* dojde k ověření uživatele, zda vlastní oprávnění provést akci. Pokud ano, proběhne kontrola, zda soubor existuje. V případě že ano, dojde ke změně přístupových práv a bude provedeno jeho odstranění.

Ukázka změny oprávnění a odebrání fotografií:

```
while ($zaznam1=MySQL_Fetch_Array($vysledek1)):
    if (file_exists("../../".$zaznam1['Cesta']))
    {
        chmod("../../".$zaznam1['Cesta_zm'], 0777);
        chmod("../../".$zaznam1['Cesta'], 0777);
        unlink("../../".$zaznam1['Cesta_zm']);
        unlink("../../".$zaznam1['Cesta']);
    }
endwhile;
```

Po fyzickém odstranění fotografie ze složky bude provedeno odstranění záznamu z databáze:

```
$dotaz="DELETE FROM fotky WHERE ID='$ID_fot'";
```

2.6 Načítání fotografií

Zobrazování fotografií bylo ošetřeno tak, aby při vstupu na stránku nebyla vidět samotná cesta k daným obrázkům. Proto bylo zobrazování provedeno přes soubory.

```
<a class="aid_1" href="load_ft/nacti_ft_or.php?id=51&name=P1000861.JPG" n
umid="1">

```

V souborech byla provedena kontrola přihlášení uživatele, a zda má patřičná oprávnění pro zobrazení fotografií. Pokud ne, tak bude zobrazeno upozornění. Pokud ano, pokračuje se dál a bude zjištěna cesta k obrázku. V poslední fázi bude obrázek načten jako data a poslán na úvodní stránku:

```
$img_data = file_get_contents("../$cesta");
header("Content-type: image/jpg");
echo $img_data;
```

Tahle operace bude provedena i pro zobrazení zmenšeného náhledu.

2.7 Datová struktura

Při tvorbě webu byla zvolena struktura složek tak, že jednotlivé soubory (.php) mají téměř každý vlastní složku, což se na začátku zdálo jako poměrně přehledné. Později s růstem počtu souborů bylo zjištěno, že se předpokládaná přehlednost ztratila a zvolený systém je špatný a nepřehledný.

ZÁVĚR

Cílem bakalářské práce bylo vytvořit webovou galerii z předem stanovených částí aplikace. Jako je stromová struktura složek, hromadné nahrávání fotografií, hodnocení fotografií a věnovat pozornost zabezpečení aplikace.

V bakalářské práci byly popsány a rozebrány všechny prvky, ze kterých se můžou webové aplikace skládat či je obsahovat. Byly popsány taktéž programy potřebné k vytvoření webové aplikace. Prostor byl věnován taky vysvětlení funkčnosti stromové struktury, nebo možnostem a druhům zabezpečení hesel ve webových aplikacích. Část práce byla taky věnována nejznámějším možnostem útoku na aplikace na internetu.

Výsledkem praktické části je funkční webová aplikace, kde bylo potřeba zapracovat předem stanovené části práce. Při tvorbě stromové struktury složek bylo zjištěno několik potíží, především zabezpečit jednotlivé složky tak, aby jejich majitel měl veškeré možnosti pro práci. Naopak neregistrovaní uživatelé či běžně registrovaný uživatel měl možnost provádět jen takové operace se složkami dotyčného majitele, které mu majitel složky povolí. K tomuto účelu byly vytvořeny práva pro jednotlivé funkce ve stromové struktuře. Zabezpečení zobrazení povolených funkcí v právech majitele bylo dosaženo kontrolou při každém kliku na složku, a podle záznamu v tabulce práv jsou zobrazeny jen funkce, které jsou povoleny majitelem u dané složky. Další problém bylo nutno odstranit při možnosti mazání složek, kde bylo potřeba kontroly, zda daná složka neobsahuje další podsložky či fotografie. Řešením byla úprava funkce tak, aby kontrolovala, zdali je prázdná či ne. V případě, že něco obsahuje je zobrazeno upozornění, kde je možnost rozhodnutí potvrdit nebo zrušit. Další částí práce bylo přidání hromadného nahrávání fotografií, kde byl objeven problém s velikostí dnešních obrázků. Tento problém byl odstraněn pomocí funkce na úpravu velikosti obrázků, která byla zapracována přímo do samotného nahrávání. Zároveň při zmenšování je vytvořen náhled na obrázek, aby se urychlilo nahrávání obrázku na stránky. Dalším prvkem je hodnocení fotografií, které bylo přidáno pod zmenšeniny. U zmenšenin byla přidána třída. Následně v galerii byl upraven kód, aby se zobrazovaly jen náhledy s danou třídou.

Ve výsledku se podařilo všechny aspekty práce do aplikace zapracovat. Při tvorbě bylo zjištěno několik nedostatků, které bych chtěl v budoucnu odstranit či dodělat. Například přidělat možnost úpravy uživatelových údajů po registraci, nebo přesunout hodnocení přímo do galerie, kde se fotografie prohlíží. Zde přidat možnost komentáře k fotografiím.

ZÁVĚR V ANGLIČTINĚ

The purpose of this bachelor's thesis was to create web gallery from in advance stated parts of application. As is the tree structure of folders, aggregate upload of photos, classification of photos and pay attention to the security of the application.

In the bachelor's thesis were described and analyzed all parts from which can be the web application consist of or contain. There also were described the programs needed for creating web applications. A certain space was devoted to explaining the functionality of the tree structure, or the possibilities and the variety of securing passwords in web applications. Part of the thesis was also devoted to the most known possibilities of attack on applications on the internet.

The result of the practical part is functional web application, where was demanded to include the parts of work which were stated in advance. During the process of creating the tree structure of folders was found a number of problems, the main was to secure the folders the way that the user has all possibilities for work. On the other hand unregistered users or commonly registered user has only the rights to do operations with the owner's folders which are allowed by the owner. To this purpose were created rights for separate functions in the tree structure. Securing the visual display of allowed functions in the rights of owner was achieved by control by every click on the folder and according to the note in table of rights are displayed only the functions, which are allowed by the owner in each folder. Another problem was to eliminate by the option of erasing folders, where was needed a control of the folder for containing another photos or subfolders. This was solved by fitting the function so it controls if it is empty or not. In case that it contains something a warning is displayed where is option the decision confirm or cancel. Next part of work was adding the aggregate upload of photos, where was discovered a problem with the size of today pictures. This was eliminated with the help of function for adjusting the size of pictures which was implemented straight to the uploading process. Simultaneously with downsizing is created a preview of the picture to speed up the uploading of the photo on the web. Next element is rating of the photos, which was added under the miniatures. There also was added class by the miniatures. Afterwards was adapted the code in gallery to show only previews with certain class.

As a result was managed to include all the aspects of the work to the application. During the creating was found a number of imperfections which I would like to eliminate or finish up in the future. For example attach the possibility of fitting the user data after registration or move the rating of the photos right into gallery where are the photos viewed. There can also be included a possibility to add comment to the photos.

SEZNAM POUŽITÉ LITERATURY

- [1] Web4company. © *web4company.cz* [online]. 2005 - 2013 [cit. 2013-03-28]. Dostupné z: <http://www.web4company.cz/technologie-zakladni-informace>
- [2] HTML. *Tvorba-webu* [online]. 2003 - 2008 [cit. 2013-03-28]. Dostupné z: <http://www.tvorba-webu.cz/html>
- [3] HTML. *HTML Tutorials* [online]. Ross Shannon, 2000-2013 [cit.2013-03-28].Dostupné z: <http://www.yourhtmlsource.com/starthere/whatishtml.html>
- [4] HTML. *W3C* [online]. 2013 [cit. 2013-03-28]. Dostupné z: <http://www.w3.org>
- [5] CSS. *W3C* [online]. 2013 [cit. 2013-03-28]. Dostupné z: <http://www.w3.org/Style/CSS/Overview.cs.html>
- [6] CSS. *W3C* [online]. 2013 [cit. 2013-03-28]. Dostupné z: <http://www.w3.org/Style/Examples/011/firstcssw.cs.html>
- [7] CSS. *Tvorba-webu* [online]. 2003 - 2008 [cit. 2013-03-28]. Dostupné z: <http://www.tvorba-webu.cz/css/>
- [8] CSS. *Jakpsatweb* [online]. Dušan Janovský, 2003 - 2012, 06. prosince 2012. [cit. 2013-03-28]. Dostupné z: <http://www.jakpsatweb.cz/css/css-uvod.html>
- [9] PHP. *Php* [online]. 2001-2013 [cit. 2013-03-28]. Dostupné z: <http://www.php.net/manual/en/preface.php>
- [10] PHP. *Tizag.com* [online]. 2003-2008 [cit. 2013-03-28]. Dostupné z: <http://www.tizag.com/phpT/.php>
- [11] PHP: php. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-30]. Dostupné z: <http://cs.wikipedia.org/wiki/PHP>
- [12] JavaScript. *Adaptic* [online]. 2005–2013 [cit. 2013-03-28]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/javascript/>
- [13] JavaScript. *Tvorba-webu* [online]. 2003 - 2008 [cit. 2013-03-28]. Dostupné z: <http://www.tvorba-webu.cz/javascript/>
- [14] JavaScript. *PESTUJEME WEB* [online]. 2008 - 2010 [cit. 2013-03-28]. Dostupné z: <http://www.pestujemeweb.cz/obsah/javascript/javascript-uvod.php>
- [15] MySQL. *MySQL* [online]. 2013 [cit. 2013-03-28]. Dostupné z: <http://www.mysql.com/>

- [16] MySQL. *Adaptic: MySQL* [online]. 2005–2013 [cit. 2013-03-28]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/mysql/>
- [17] JQuery. *JQuery* [online]. 2006 [cit. 2013-03-28]. Dostupné z: <http://jquery.com>
- [18] PhpMyAdmin. *PhpMyAdmin* [online]. 2003 - 2013 [cit. 2013-03-29]. Dostupné z: http://www.phpmyadmin.net/home_page/index.php
- [19] XAMPP: xampp. *XAMP Web Server* [online]. Free Software | Designs by Fab Themes & Web2feel Bloggerized by DheTemplate.com & WordPress Theme, 2011 [cit. 2012-05-01]. Dostupné z: <http://jomdownloadsoftware.blogspot.com/2009/10/xamp-web-server.html>
- [20] PSpad. *PSPad freeware editor* [online]. WebDesign PAY & SOFT, 2001 - 2012 Jan Fiala [cit. 2012-05-01]. Dostupné z: <http://www.pspad.com/cz/>
- [21] Gimp. *PC-guru* [online]. 2000-2008 [cit. 2013-03-29]. Dostupné z: <http://www.pc-guru.cz/gimp>
- [22] Gimp. *Gimp* [online]. 2001-2013 [cit. 2013-03-29]. Dostupné z: <http://www.gimp.org/>
- [23] Gimp. *Návod pro gimp* [online]. 2002 [cit. 2013-03-29]. Dostupné z: <http://gimp.4fan.cz/>
- [24] Ukládání dat a tvorba stromové struktury. *Interval.cz* [online]. ZONER software, a.s., 2009 [cit. 2012-05-01]. Dostupné z: <http://interval.cz/clanky/metody-ukladani-stromovych-dat-v-relacnich-databazich/>
- [25] Hesla. *Zdroják.cz* [online]. 2011 [cit. 2013-03-29]. Dostupné z: <http://www.zdrojak.cz/clanky/nekolik-poznamek-k-heslum>
- [26] MD. *MD* [online]. 1992 [cit. 2013-03-29]. Dostupné z: <http://tools.ietf.org/html/rfc1321>
- [27] MD. *Message-Digest Algorithm* [online]. 1998 - 2012 [cit. 2013-03-29]. Dostupné z: <http://www.networksorcery.com/enp/data/md5.htm>
- [28] MD. *PŘEHLED NEJDŮLEŽITĚJŠÍCH HASHOVACÍCH FUNKCÍ* [online]. 1999 - 2013 [cit. 2013-03-29]. Dostupné z: <http://www.kryptografie.wz.cz/data/hash.htm>
- [29] MD. *W3C* [online]. 1997 [cit. 2013-03-29]. Dostupné z: http://www.w3.org/TR/1998/REC-DSig-label/MD5-1_0

- [29] SHA. *Bruce Schneier* [online]. 2000-2012 [cit. 2013-03-29]. Dostupné z: http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html
- [30] SHA. *PŘEHLED NEJDŮLEŽITĚJŠÍCH HASHOVACÍCH FUNKCÍ* [online]. 1999 - 2013 [cit. 2013-03-29]. Dostupné z: <http://www.kryptografie.wz.cz/data/hash.htm>
- [31] SHA. *Free Internet Security* [online]. 2002-2013 [cit. 2013-03-29]. Dostupné z: <http://www.securitysupervisor.com/security-q-a/encryption/243-what-is-sha-2>
- [32] Galerie. *Galerie.cz* [online]. 2009-2013 [cit. 2013-04-03]. Dostupné z: <http://galerie.cz/>
- [33] Galerie. *Rajce.idnes.cz* [online]. 2005-2013 [cit. 2013-04-03]. Dostupné z: <http://www.rajce.idnes.cz/>
- [34] Galerie. *Zonerama* [online]. 2013 [cit. 2013-04-03]. Dostupné z: <http://www.zonerama.com>
- [35] Hash. *Salt* [online]. 2011 [cit. 2013-04-03]. Dostupné z: <http://interval.cz/clanky/salted-hash-dalsi-krok-ke-zvyseni-bezpecnosti>
- [36] Hash. *Salt* [online]. 2011 [cit. 2013-04-03]. Dostupné z: <http://www.zdrojak.cz/clanky/nekolik-poznamek-k-heslum/>
- [37] HTTPS. *Adaptic* [online]. 2005-2013 [cit. 2013-04-03]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/https/>
- [38] HTTPS. *Alpiro* [online]. 2006-2013 [cit. 2013-04-03]. Dostupné z: <https://www.alpiro.cz/https.html>
- [39] Hash na straně klienta. *Root.cz* [online]. 1998 – 2013 [cit. 2013-04-03]. Dostupné z: <http://www.root.cz/clanky/bezpecne-prihlasovani-uzivatelu/>
- [40] SQL injection. *Binary Flow* [online]. 2010-2013 [cit. 2013-04-05]. Dostupné z: <http://bflow.security-portal.cz/sql-injection-full-paper/>
- [41] SQL injection. *Programujte.com* [online]. 2003-2013 [cit. 2013-04-05]. Dostupné z: <http://programujte.com/clanek/2007041802-sql-injection-a-zabezpeceni/>
- [42] SQL injection. *Zdeněk Večeřa* [online]. 2008-2012 [cit. 2013-04-05]. Dostupné z: http://blog.zdenekvecera.cz/item/jak-na-to-sql-injection-magic_quotes_gpc-addslashes-a-stripslashes

- [43] XSS. *Root.cz* [online]. 1998 – 2013 [cit. 2013-04-05]. Dostupné z: <http://www.root.cz/clanky/xss-stale-na-scene/>
- [44] XSS. *Rdroják.cz* [online]. 2009 [cit. 2013-04-05]. Dostupné z: <http://www.zdrojak.cz/clanky/co-je-xss-jak-mu-predchazet/>
- [45] Stromová struktura složek. *{P}* [online]. 2009 [cit. 2013-04-09]. Dostupné z: <http://code.google.com/p/editable-jquery-tree-with-php-codes>
- [46] Hromadné nahrávání souborů. *Uploadify* [online]. 2013 [cit. 2013-04-10]. Dostupné z: <http://www.uploadify.com>

Seznam použitých symbolů a zkratek

WWW	- World Wide Web
HTTP	- Hypertext Transfer Protocol
URL	- Uniform Resource Locator
HTML	- HyperText Markup Language
CSS	- Cascading Style Sheets
PHP	- Hypertext Preprocessor
XHTML	- Extensible HyperText Markup Language
MySQL	- My Structured Query Language
DOM	- Document Object Model
DBMS	- DataBase Managment Systém
GPL	- General Public License
MD	- Message-Digest algorithm
SHA	- Secure Hash Algorithm
HTTPS	- Hypertext Transfer Protocol Secure
XSS	- Cress-site scripting

SEZNAM OBRÁZKŮ:

- Obrázek 1 – galerie.cz menu
- Obrázek 2 – galerie.cz zobrazení fotek
- Obrázek 3 – rajce.idnes.cz menu
- Obrázek 4 – rajce.idnes.cz zobrazování fotografií
- Obrázek 5 – zonerva.com profil uživatele
- Obrázek 6 – zonerva.com menu
- Obrázek 7 – zonerva.com zobrazování fotografií
- Obrázek 8 – Výsledek CSS stylování
- Obrázek 9 – Ukázka pracovního prostředí phpMyAdmin
- Obrázek 10 – Ukázka pracovního prostředí XAMPP
- Obrázek 11 – Stromová struktura
- Obrázek 12 – Klasická datová struktura pro uložení stromu v relační tabulce
- Obrázek 13 – Rozšíření klasického přístupu o pořadí a hloubku zanoření
- Obrázek 14 – Návrh DB
- Obrázek 15 – Rozložení stránky
- Obrázek 16 – Stromová struktura a rozbalovací menu
- Obrázek 17 – Nahrávání fotografií
- Obrázek 18 – Práce s fotografiemi
- Obrázek 19 – Úvod instalace
- Obrázek 20 – Tvorba databáze
- Obrázek 21 – Dokončení databáze
- Obrázek 22 – Úvodní obrazovka
- Obrázek 23 – Registrace
- Obrázek 24 – Přihlášení
- Obrázek 25 – Zobrazovací menu
- Obrázek 26 – Nahrávání fotografií
- Obrázek 27 – Seznam nahrávaných fotografií
- Obrázek 28 – Práce s fotografiemi

SEZNAM PŘÍLOH

P I Návod k obsluze.

PŘÍLOHA 1: NÁVOD K ODSLUŽE

Instalace:

Po stažení aplikace z internetu a následném spuštění vás přivítá úvodní stránka, která vás provede instalací aplikace:

Vítejte při Instalaci Webové galerie

- *Úvod*
- Instalace databáze
- Dokončení

Další

Obrázek 19 – Úvod instalace

Po kliknutí na tlačítko *další* budete vyzváni k zadání parametrů důležitých pro vytvoření databáze. Ještě před vyplněním požadovaných parametrů je nutné si na serveru založit vlastní databázi. (Např: DB host:localhost, DB login:root, DB password:“nemusí být zadáno“ DB Name:“jméno databáze které jsem si vytvořil na serveru“). Po připravení databáze můžeme vyplnit údaje:

- *Úvod*
- Instalace databáze
- Dokončení

DB host	<input type="text" value="localhost"/>
DB login	<input type="text" value="root"/>
DB password	<input type="text"/>
DB Name	<input type="text" value="galerie"/>
<input type="button" value="Vytvořit databázi"/>	

Obrázek 20 – Tvorba databáze

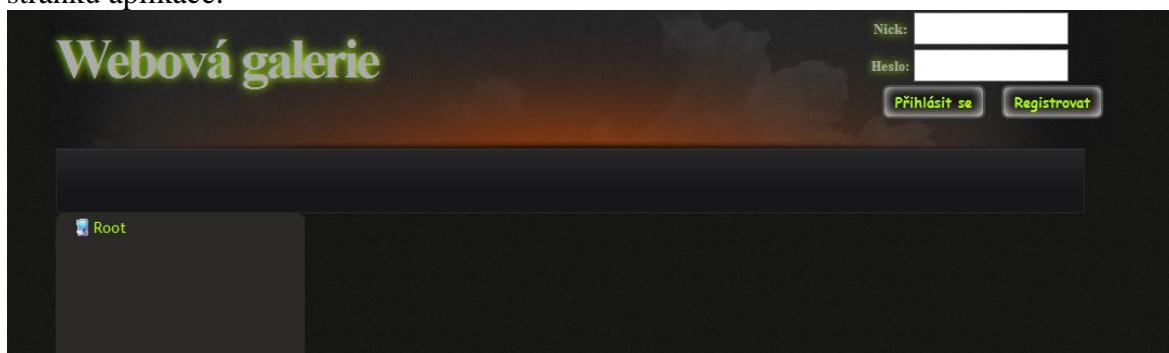
Po vyplnění můžete stisknout tlačítko: *Vytvořit databázi*, které provede vytvoření veškerých tabulek ve vaší databázi a taky nastaví potřebné soubory. Pokud vše proběhne správně, objeví se nápis:

Instalace databáze byla úspěšně dokončena.

Dokončit

Obrázek 21 – Dokončení databáze

V opačném případě budete upozorněni na případné chyby. Po kliknutí na tlačítko *Dokončit* je aplikace nainstalována a připravena k použití. A hned budete přesměrováni na hlavní stránku aplikace:



Obrázek 22 – Úvodní obrazovka

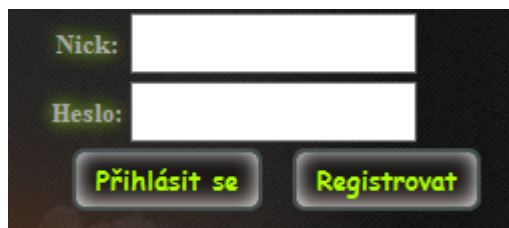
Ovládání:

Po zobrazení úvodní obrazovky je nutné provést registraci uživatele:

Nick:	<input type="text"/>
Jmeno:	<input type="text"/>
Prijmeni:	<input type="text"/>
Heslo:	<input type="text"/>
Ověření hesla:	<input type="text"/>
Email:	<input type="text"/>
<input type="button" value="Registrovat se"/>	

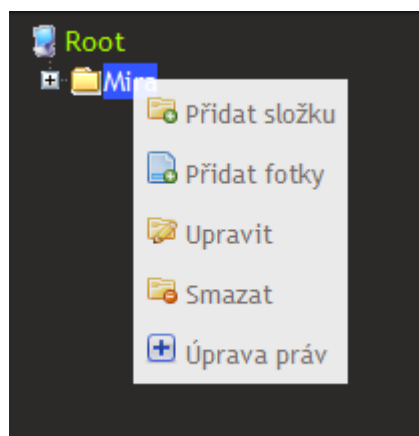
Obrázek 23 – Registrace

Zde je potřeba vyplnit veškeré požadované údaje. V případě nějaké chyby budete upozorněni a vyzváni k nápravě. Po úspěšně dokončené registraci budete vybídnuti k přihlášení:



Obrázek 24 – Přihlášení

Kde vyplníte Nick, Heslo a stisknete *Přihlásit se*, pokud je vše v pořádku, budete přihlášení a můžete začít pracovat v aplikaci. V opačném případě budete upozorněni. Pro vytvoření složky je potřeba kliknout pravým tlačítkem myši na root. Zobrazí se vám možnost vytvoření složky. Po kliknutí na tuhle možnost se zobrazí textové okénko pro vyplnění názvu složky. Pro dokončení stiskněte enter. Pod nápisem root se vám zobrazí vaše složka a opět po kliknutí pravým bude zobrazeno rozbalovací menu, kde zvolíte operaci, kterou budete chtít provést:

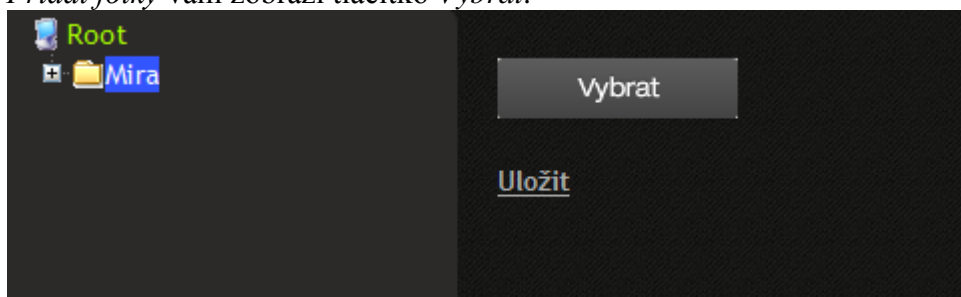


Obrázek 25 – Zobrazovací menu

Funkce *Upravit* slouží k přejmenování složky.

Funkce *Smazat* slouží ke smazání složky.

Funkce *Přidat fotky* vám zobrazí tlačítko *Vybrat*:



Obrázek 26 – Nahrávání fotografií

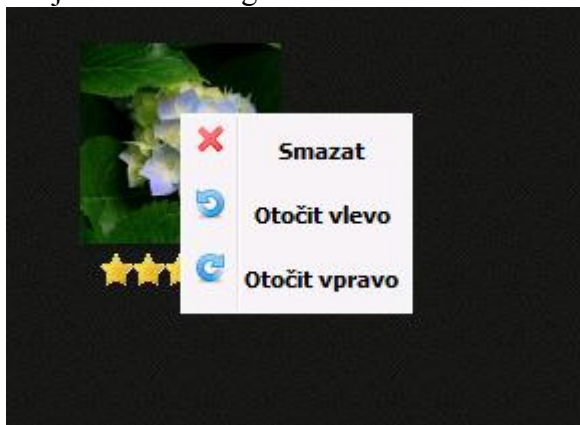
Po kliku na tlačítko budete mít možnost si dané fotografie vyhledat a nahrát jejich seznam na stránky:



Obrázek 27 – Seznam nahrávaných fotografií

Zde máte možnost seznam upravit a nežádoucí fotografie odstranit pomocí křížku. Pokud je vše v pořádku fotografie budou nahrány na server. A po kliku na danou složku si je můžete prohlížet.

Funkce *Upravit práva* vám umožní přidávat či odebírat práva u jednotlivých složek. Jsou zde k dispozici základní operace pro fotografie, kterých docílíme tím, že klikneme pravým tlačítkem myši na jednotlivé fotografie:



Obrázek 28 – Práce s fotografiemi

Tímto bych vám chtěl popřát hodně zdaru při práci s aplikací. A taky poděkovat, že používáte naši aplikaci