

Dynamické webstránky – ZO ČSCH Zdounky

Dynamic webpages – ZO ČSCH Zdounky

Alexander Déduch

ABSTRAKT

Cílem této bakalářské práce je vytvoření dynamických webových stránek pro ZO ČSCH Zdounky. Stránky budou následně použity v běžném internetovém provozu. První část práce je věnována historii internetu a technologiím použitým při vytváření stránek. V další části je uvedena struktura webových stránek s náhledy obrazovek a struktura databáze s popisem jednotlivých tabulek. Posledním část práce obsahuje popis webových stránek z pohledu návštěvníka i administrátora.

Klíčová slova: webové technologie, PHP, MySQL, dynamické webstránky

ABSTRACT

The main aim of the thesis is to create the web presentation for the club ZO ČSCH Zdounky. The webpages will be used in regular internet traffic. The first part of this thesis deals with the history of internet and the technologies used to create the project. Furthermore, I present the structure of the dynamic webpages with several views and the structure of the database accompanied by the description of the tables. The last part contain the description of the webpages from the point of view of a visitor and an administrator.

Keywords: web technologies, PHP, MySQL, dynamic webpages

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

Ve Zlíně, 15.května 2008

.....

Alexander Déduch

OBSAH

ÚVOD	6
I TEORETICKÁ ČÁST	7
1 HISTORIE WWW	8
1.1 VZNIK WWW	8
1.2 VÝVOJ WWW.....	10
1.3 HISTORIE JEDNOTLIVÝCH VERZÍ HTML	10
2 TECHNOLOGIE WWW	13
2.1 PHP.....	13
2.1.1 Historie PHP.....	13
2.1.2 Popis jazyka PHP	15
2.1.3 Základní syntaxe a příkazy jazyka PHP	16
2.2 MYSQL.....	19
2.2.1 Popis MySQL.....	19
2.2.2 Základy práce s databází a její vytvoření	19
2.2.3 Struktura MySQL.....	20
2.2.4 Návrh a vytváření tabulek	20
2.2.5 Základní SQL dotazy.....	21
2.2.6 Propojení PHP a MySQL	22
2.3 AJAX.....	22
2.3.1 Popis a využití jazyka.....	23
2.3.2 Výhody a nevýhody použití.....	24
2.4 CSS.....	25
2.4.1 Popis CSS.....	26
2.4.2 Vývoj CSS.....	26
2.4.3 Syntaxe CSS.....	27
2.4.4 Výhody a nevýhody	27
II PRAKTICKÁ ČÁST	29
3 TVORBA WEBOVÝCH STRÁNEK	30
3.1 PŮVODNÍ STRÁNKY.....	30
3.1.1 Architektura původních stránek	30
3.1.2 Návrh nových stránek.....	30
3.2 NÁVRH A STRUKTURA POUŽITÉ DATABÁZE.....	31
3.2.1 Použité tabulky a jejich struktura	31
3.3 POPIS A STRUKTURA STRÁNEK.....	33
3.3.1 Popis jednotlivých odkazů menu.....	33
4.1 POPIS ADMINISTRAČNÍ SEKCE	36
4.1.1 Editace článků pomocí FCK editoru	36
4.1.2 Přihlášení do administrační sekce	37
4.1.3 Správa uživatelských účtů.....	39
4.1.4 Články	39

4.1.5	Změna uživatelského profilu	40
4.2	KATALOG PRODEJNÝCH ZVÍŘAT	40
4.2.1	Vložení nové skupiny zvířat.....	40
4.2.2	Vložení nové zvířete do skupiny	40
4.2.3	Úprava a mazání skupin a zvířat	41
4.2.4	Vyhledávání v člancích	41
ZÁVĚR		42
CONCLUSION		43
SEZNAM POUŽITÉ LITERATURY		44
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK		45
SEZNAM OBRÁZKŮ		47
SEZNAM TABULEK.....		48
SEZNAM PŘÍLOH.....		49

ÚVOD

V šedesátých letech se snažila americká armáda najít řešení jak zajistit, aby armádní počítače rozmístěné po celém území USA mohly spolu bez problému komunikovat, a to i v případě, že část této sítě bude vyřazena z provozu. Tehdy asi ještě nikdo netušil, co za revoluci způsobí internet po celém světě. Když poskytl potřebné finanční prostředky z resort americké obrany, konkrétně grantová agentura ministerstva obrany USA, s názvem ARPA (Advanced Research Projects Agency) mohl s začít s plnohodnotným vývojem sítě. Podle této grantové agentury byla experimentální síť, která vznikla v roce 1969 také pojmenována jako ARPANET. Po té, v roce 1980, přišel ve švýcarském institutu pro jaderný výzkum CERN Tim Berners-Lee s myšlenkou hypertextu, což mělo usnadnit sdílení a aktualizaci informací mezi výzkumníky. V roce 1989 již měl CERN největší internetový server v Evropě a Tim Berners-Lee oživil tehdy zapomenutou myšlenku hypertextu. Již v listopadu roku 1990 předvedl první prototyp WWW serveru, jenž pojmenoval jednoduše http. Ačkoliv bylo v roce 1984 k Internetu připojeno pouhých 1000 počítačů, už v roce 1992 bylo k Internetu připojeno více než jeden milion. Postupem času se stával internet stále přístupnějším i veřejnosti. Dnes si lidé život bez internetu nedokáží představit. Přes internet lidé navzájem komunikují, čerpají cenné informace, nakupují v internetových obchodech. Na internetu se stále více prezentují kromě firem i různé zájmové spolky.

Náplní této bakalářské práce bylo vytvořit právě jeden takový projekt. Jedná se o webové stránky ZO ČSCH Zdouňky. Hlavním cílem bylo vytvořit dynamické web stránky s novou grafickou podobou. Zároveň by měly poskytnout informace o činnosti chovatelů Zdouňka. Dynamičnosti stránek bylo dosaženo použitím webových technologií CSS, PHP, MySQL, JavaScript.

První dvě kapitoly v této práci popisují historii vzniku internetu a vybrané používané technologie při tvorbě webových prezentací. Další kapitoly již popisují samotný projekt jak z pohledu návštěvníka, tak i z pohledu administrátora.

I. TEORETICKÁ ČÁST

1 HISTORIE WWW

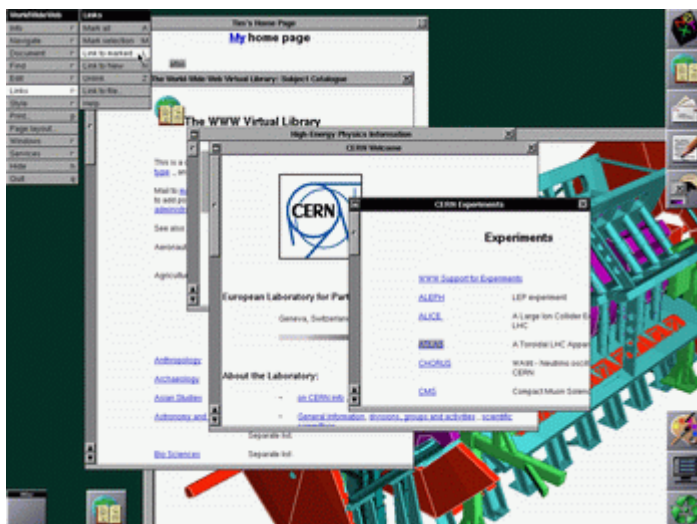
World Wide Web (WWW, nebo také pouze zkráceně web), ve volném překladu „Celosvětová pavučina“, je označení pro aplikace internetového protokolu HTTP, který byl původně určený pouze pro výměnu hypertextových dokumentů ve formátu HTML. V češtině se slovo web často používá nejen pro označení celosvětové sítě dokumentů, ale také pro označení soustavy dokumentů, které jsou dostupné na témže webovém serveru nebo na téže internetové doméně nejnižšího stupně (internetové stránce).

Dokumenty umístěné na počítačových serverech jsou adresovány pomocí URL. Součástí URL je doména a jméno počítače. Název naprosté většiny těchto serverů začíná zkratkou www, avšak je možné používat libovolné jméno vyhovující pravidlům URL.

Protokol HTTP je dnes již používán i pro přenos jiných dokumentů, než jen souborů ve tvaru HTML a výraz World Wide Web se postupně stává pro laickou veřejnost synonymem pro internetové aplikace.

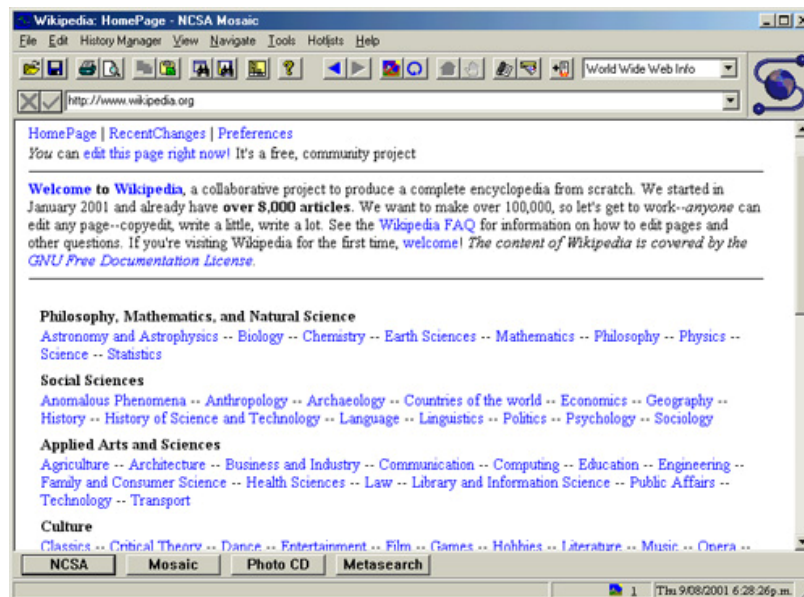
1.1 Vznik WWW

Za vznikem WWW stojí Tim Berners-Lee, který pracoval ve švýcarském institutu pro jaderný výzkum CERN. V roce 1980 přišel s myšlenkou hypertextu ve snaze usnadnit sdílení a aktualizaci informací mezi jednotlivými výzkumníky. Za čtyři roky, po opětovném návratu Tima Berners-Leeho do výzkumného ústavu, tzn. v roce 1989 již měl CERN největší internetový server v Evropě a proto mohla být myšlenka realizována. V té době se pro tvorbu dokumentů obvykle používal TeX (sázecí systém), Postscript a také SGML. Tim Berners-Lee si uvědomoval, že potřebují něco jednoduššího. Proto naprogramoval první server, který nazval jednoduše *httpd*. Poté vytvořil první webové stránky, které byly spuštěny 6. srpna 1991 a webový prohlížeč. Ten měl název WorldWideWeb, běžel na systémech NeXTSTEP a byl zároveň i prvním WYSIWYG HTML editorem pro tvorbu www stránek. Posléze byl tento webový prohlížeč přejmenován na prohlížeč Nexus.



Obr. 1 - První webový prohlížeč Nexus

Prohlížeč Nexus ale nebyl určen obyčejnému uživateli. Prohlížet web jim umožnil program Mosaic, i když ten byl původně určen spíše pro jinou platformu. S jeho vývojem začali v roce 1992 v NCSA (National Center for Supercomputing Applications) pánové Marc Andreessen a Eric Bina. Původně byl naprogramován pro X Window pro systémy Unix. První verze byla uvolněna v roce 1993. Koncem roku byla uvedena i verze 1.0 pro systémy Apple Macintosh a také Microsoft Windows. Již od prvních interních testovacích verzí podporoval plně HTML, obrázky formátu GIF a prostřednictvím externích prohlížečů i JPEG, MPEG, QuickTime, PostScript a další. Uživatelské prostředí obsahovalo Toolbar a stavový řádek, kde se zobrazovala adresa odkazů i aktuální stav nahrávání stránky. Uživatel si mimo jiné mohl kompletně zvolit styl písma na stránkách. Program si také pamatoval historii navštívených stránek. [9],[10]



Obr. 2 - Webový prohlížeč Mosaic

1.2 Vývoj WWW

Zveřejněním webového prohlížeče Mosaic, který je někdy označován jako "killer application" devadesátých let začíná růst počet uživatelů počítačů, a s tím i počet instalací prohlížečů a serverů WWW. V roce 1994 byla uspořádána první mezinárodní konference věnovaná pouze systému WWW ve středisku CERN. Poté se vývoj www přesouvá z CERNU do francouzského institutu INRIA. Ještě téhož roku vznikla nekomerční organizace W3C, která sjednotila verze jazyka HTML od různých výrobců a dohodla se s nimi na základních principech a komponentách nových standardů. Největší boom zažívá systém WWW na počátku roku 1995. Dnes je World Wide Web Consortium (W3C) je mezinárodní konsorcium, jehož členové společně s veřejností vyvíjejí webové standardy pro World Wide Web.

1.3 Historie jednotlivých verzí HTML

Kvůli velkému rozvoji webu a bylo nutné pro HTML definovat určité standardy. První verze (verze 0.9) HTML jazyka byla vydána zhruba v roce 1991 a nepodporovala grafický režim. Tato verze ještě nebyla považována za oficiální. Tou se stala verze HTML 2.0. Původně se myslelo, že vývoj jazyka HTML skončí v prosinci roku 1999 a bude se používat pouze XHTML, která je založena na XML jazyku. Ale 7. března 2007 byla založena nová pracovní skupina HTML, jejíž cílem je vývoj nové verze HTML.

HTML verze 2.0

Historie značkovacích jazyků pro vytváření webových stránek začala specifikace, kterou vydala Internet Engineering Task Force (IETF), internetová standardizační organizace, v roce 1995 jako RFC. Tento jazyk byl jakousi podmnožinou SGML, univerzálního značkovacího jazyka, který je pod patronací standardizační organizace ISO. SGML si můžeme představit jako nástroj k definici jakýchkoli jazyků pro popis informací.

Specifikace HTML 2.0 ale neobsahovala příliš mnoho prvků k vytváření pohledných grafických stránek, a tak byla hlavně kvůli tlaku ze strany Netscape a Microsoftu uvolněna specifikace HTML 3.2

HTML verze 3.2

Byla vydána v roce 1997. Připravovaná verze HTML 3.0 nebyla nikdy přijata jako standard, protože byla příliš složitá a žádná firma nebyla schopna naprogramovat její podporu ve svém prohlížeči. Standard už vydalo konsorcium W3C, stejně jako následující verze. Přidává k jazyku tabulky, zarovnávání textu a stylové elementy pro ovlivňování vzhledu.

HTML verze 4.0

Byla vydána 18. prosince 1997. V této verzi se již definuje vcelku mocný jazyk pro definici struktury dokumentu. Vzhled dokumentu určuje skoro výhradně CSS. Organizace W3C ale chtěla zároveň nechat výrobcům prohlížečů a autorům stránek čas k přechodu na tuto normu. Proto došlo k rozdělení na verzi HTML 4.0 a na HTML 4.0 Strict. Dále na verzi HTML 4.0 Transitional což byla přechodná norma, která zachovává vzhledové atributy HTML 3.2 a zároveň obsahuje všechny výhody a vylepšení HTML 4.0 Strict, především v oblasti přístupnosti a CSS a na verzi HTML 4.0 Frameset. Tato norma se téměř shoduje s HTML 4.0 Transitional, je pouze rozšířena o elementy pro definici rámců.

HTML verze 4.01

Tato verze byla vydána v roce 1999. Tato verze opravuje některé chyby předchozí verze a přidává některé nové tagy. Tato verze obsahuje kompletní popis jazyka a všechny následující specifikace jazyků pro tvorbu webu, které dosud vyšly, se na ni odkazují.

Budoucnost vývoje HTML

V květnu 2007 bylo odhlasováno, že základem nové specifikace se stanou Web Applications 1.0 a Web Forms 2.0 ze specifikace WHATWG. Jako název nové specifikace bylo odhlasováno HTML 5.0 a specifikace by měla být hotova v roce 2010 (odtud ji začnou vývojáři webových aplikací používat). S finální verzí, která bude zbavená všech chyb se však odhaduje až na rok 2022.

2 TECHNOLOGIE WWW

S postupným vývojem internetu se začaly zvyšovat nároky na jeho využití, s čímž také plně souvisí požadavek na zdokonalování a vytváření nových programovacích jazyků.

Dnešní programovací jazyky pro tvorbu webových stránek se dělí na dvě skupiny stránek. První skupina obsahuje programovací jazyky, jejichž skripty se provádí přímo na straně klienta. Řadí se zde programovací jazyky JavaScript, Java Applet, DHTML. Druhou skupinou jsou pak programovací jazyky, jejichž skripty se provádí na straně serveru. Mezi tyto jazyky patří PHP, ASP.NET a AJAX.

Dané programovací jazyky se mohou využívat samostatně. V dnešní době ovšem stále častěji dochází ke kombinaci těchto jazyků. Je proto zřejmé, že každý jazyk má své uplatnění při programování a nedá se říct, že jeden je hodnotnější než ten druhý.

2.1 PHP

PHP původně Personal Home Page je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, XHTML či WML, což je velmi výhodné pro tvorbu webových aplikací. PHP lze ovšem také použít i k tvorbě konzolových a desktopových aplikací.

2.1.1 Historie PHP

PHP je nástupcem staršího produktu nazvaného PHP/FI, které vytvořil Rasmus Lerdorf v roce 1995 jako prostředek pro záznamy o čtenářích svého online životopisu. Na počátku se jednalo o jednoduchou sadu skriptů v jazyce Perl. Tuto sadu nazval Personal Home Page Tools. Protože byla potřeba větší funkčnost, napsal Rasmus mnohem rozsáhlejší implementaci v jazyku C, která byla schopna komunikovat s databázemi a umožňovala uživatelům vyvíjet jednoduché dynamické aplikace pro Web. Rasmus se rozhodl uvolnit tento zdrojový kód PHP/FI pro všechny, aby ho mohl kdokoliv používat, opravovat případné chyby či ho nejrůzněji vylepšovat.

V roce 1997 se PHP/FI 2.0, druhá implementace psaná v jazyku C, stala kultovní záležitostí pro tisíce uživatelů po celém světě. I když do projektu začalo svými kusy kódu přispívat více lidí, stále to byl velký projekt jednoho muže.

PHP/FI 2.0 bylo oficiálně uvolněno až v listopadu 1997, poté co strávilo většinu svého života v betaverzích. Krátce nato bylo následováno první alfa verzí PHP 3.0. [11]

PHP 3

PHP 3.0 bylo oficiálně uvolněno v červnu 1998, poté co bylo skoro 9 měsíců veřejně testováno. Byla to první verze tohoto jazyka, která se velmi blížila takovému PHP, jak ho známe dnes. Vytvořili ho Andi Gutmans a Zeev Suraski v roce 1997 jako kompletně přepsaný celek, poté co shledali PHP/FI 2.0 výrazně "poddimezované" pro vývoj svých aplikací pro e-komerci. Ve snaze spolupracovat a zahájit budování nad existující uživatelskou základnou PHP/FI, rozhodli se Andi, Rasmus a Zeev pracovat společně a prohlásit PHP 3.0 za oficiálního nástupce PHP/FI 2.0, a vývoj PHP/FI 2.0 byl v podstatě zastaven.

Jednou z nejsilnějších zbraní PHP 3.0 byly jeho obrovské možnosti rozšíření. K poskytnutí pevné infrastruktury pro mnoho různých databází, protokolů a API koncovým uživatelům, přilákaly možnosti rozšíření PHP 3.0 také tucty vývojářů, kteří se připojili a vytvořili nové rozšiřující moduly. Toto byl nesporně klíč k obrovskému úspěchu PHP 3.0. Jiným klíčovým prvkem v PHP 3.0 byla podpora objektově orientované syntaxe a mnohem silnější a konzistentnější syntaxe jazyka.

Nový jazyk byl uvolněn pod novým názvem, který odstranil implikaci omezeného osobního použití, kterou neslo označení PHP/FI 2.0. Byl nazván pouze 'PHP', což je rekurzivní akronym - PHP: Hypertext Preprocessor.

Na konci roku 1998 vyrostlo PHP do rozsahu instalací v řádu (odhadem) desítek tisíc uživatelů a stovek tisíc Webů. V době svého vrcholu bylo PHP 3.0 instalováno na 10 % všech WWW serverů na Internetu.

PHP 4

V zimě roku 1998, krátce po oficiálním uvolnění PHP 3.0, začali Andi Gutmans a Zeev Suraski pracovat na přepsání jádra PHP. Cílem návrhu bylo zvýšit výkon pro složité aplikace a zlepšit modularitu kódové báze PHP.

Nový engine, nazvaný 'Zend Engine', který byl sestaven z křestních jmen jeho autorů, úspěšně splnil cíle návrhu a byl uveden v polovině roku 1999. PHP 4.0, založené na tomto enginu a doplněné širokou škálou nových prvků, bylo oficiálně uvolněno v květnu 2000, necelé dva roky po svém předchůdci, PHP 3.0. K podstatně zvýšenému výkonu této verze,

přidává PHP 4.0 další klíčové prvky, jako je podpora pro mnoho WWW serverů, HTTP sessions, buffering výstupu, bezpečnější způsoby zpracování vstupů uživatele a mnoho nových jazykových konstruktů.

PHP 5

Na vývoji PHP 5 se začalo pracovat již v roce 2002. Základem této verze je zcela přepracovaný Zend Engine 2, který jednak přinesl vyšší výkon kritických PHP aplikací, jednak umožnil zakomponovat do PHP řadu pokročilejších programovacích struktur, čímž především otevřel cestu kvalitnější podpoře objektově orientovaného programování v PHP.

První betaverze PHP 5 byly veřejnosti k dispozici na jaře roku 2003 a první oficiální verze PHP 5 byla uvolněna 13. července 2004.

PHP 5 je do značné míry zpětně kompatibilní s PHP 4. V nové verzi jazyka byly především posíleny bezpečnostní mechanismy (což může vést k nefunkčnosti některých špatně postavených aplikací pro PHP 4) a uveden nový, podstatně kvalitnější objektový model, umožňující používat PHP jako skutečný objektově orientovaný jazyk.

Novinky v PHP 5

- objektově orientované programování
- kompletní přepis MySQL rozšíření
- souprava spolupracujících XML nástrojů

2.1.2 Popis jazyka PHP

PHP je v současnosti velmi rozšířená technologie umožňující snadné programování na straně serveru (server-side programming). Toho lze využít k tvorbě různých interaktivních webových stránek. Stručně lze říci, že skript napsaný v PHP je proveden na serveru podle zadaných kritérií a výsledek je odeslán volajícímu počítači stejným způsobem, jakým se odesílají běžné statické (XHTML) stránky. Jakmile je však stránka načtena klientem, pomocí PHP ji již není možné dále měnit.

Změnu stránky bez nutnosti jejího opětovného načtení provádíme skripty na straně klienta - například pomocí JavaScriptu. Výhoda těchto programovacích jazyků spočívá tedy v tom,

že není třeba pro změnu stránky neustále obnovovat obsah stránky. Jejich veliké omezení však spočívá nejen v možnostech, ale také v jejich (ne)bezpečnosti. Protože skripty se vykonávají přímo na počítači uživatele, mohl by potenciální útočník bez problémů vykonat nebezpečný kód. Z tohoto důvodu jsou například v JavaScriptu vypnuty funkce, které by přímo pracovaly se soubory na harddisku. Díky tomu se JavaScript využívá spíše pro okamžitou úpravu CSS kódu či jako "doplněk" při vytváření webové grafiky. Pro "opravdové" funkce internetových stránek se ale používají skripty na straně serveru.

PHP je programovací jazyk umožňující procedurální i objektově orientované programování. Znalost objektově orientovaného programování (OOP) tedy může být při práci v PHP výhodou, není však nutnou podmínkou. PHP také patří mezi jazyky, kde například není nutné předem definovat typ proměnných, navíc jakákoli proměnná může kdykoli změnit svůj typ. Jednoduše řečeno, co se týče psaní kódu, z PHP při psaní skriptů "sálá" určitá volnost a neomezenost. Na druhé straně záleží plně na programátorovi, jaký si bude v kódu udržovat pořádek. [3],[4]

Některé z předností PHP

- vysoká výkonnost
- rozhraní pro mnoho druhů databázových systémů
- zabudované knihovny pro implementaci mnoha běžných webových úloh
- nízké náklady
- snadná výuka a použití
- přenositelnost

2.1.3 Základní syntaxe a příkazy jazyka PHP

PHP je programovací jazyk umožňující procedurální i objektově orientované programování navíc patří mezi jazyky, kde například není nutné předem definovat typ proměnných. PHP skripty jsou prováděny na straně serveru, k uživateli je přenášěn až výsledek jejich činnosti. Syntaxe jazyka kombinuje hned několik programovacích jazyků (Perl, C, Pascal a Java). PHP je nezávislý na platformě, skripty fungují bez úprav na mnoha různých operačních systémech.

Jazyk PHP je dynamicky typový. Datový typ proměnné se určí v okamžiku přiřazení hodnoty. Díky tomu má PHP dva typy porovnání, '==' stejný jako v C a '===', který platí jen když jsou oba dva výrazy stejného typu. Pole jsou heterogenní, mohou tedy obsahovat jakékoli údaje, stejně tak jako jejich indexy. Řetězce lze uzavírat jak do uvozovek - obsah je parsován, tak do apostrofů - obsah není parsován.

Začínající a ukončovací PHP tagy	Popis
<?php ... ?>	nejčastěji používané tagy pro označení PHP skriptu
<? ... ?>	
<script language="php"> ... </script>	další z možných označení PHP skriptu, avšak se moc nedoporučuje používat

Tab. 1 - Přehled začínajících a ukončovacích PHP tagů

Kód jazyka PHP je především tvořen příkazy. Příkazy vykonávají určitou funkci a krok za krokem se provádí námi naprogramovaný algoritmus. Příkazy mohou mít různý tvar. Může se jednat o volání funkce, libovolnou početní operaci, přiřazení nejrůznější hodnoty. Jednotlivé příkazy je nutné od sebe oddělovat středníkem.

Proměnné

Každý programovací jazyk pracuje s nějakými hodnotami. A právě proměnné slouží k uchování těchto hodnot. Každá proměnná má název (někdy se mu říká identifikátor) a hodnotu (někdy se označuje termínem literál). V každém okamžiku je hodnota proměnné v programu jednoznačně daná, Každá proměnná musí mít také jednoznačný název. Ten v PHP začíná znakem dolaru (\$) a následuje (bez mezery) nějakým pojmenováním. První znak toho pojmenování musí být buď písmeno a-z nebo podtržítka. Nesmí to být číslo ani nic jiného. Názvy proměnných v PHP rozlišují mezi malými a velkými písmeny, proto si musíme dát pozor na správný zápis. V názvech proměnných můžete používat české znaky včetně diakritiky. Desetinná čísla se zadávají s tečkou, ne s čárkou. Řetězce se uzavírají do uvozovek nebo do apostrofů.

Datové typy

PHP rozlišuje celkem 8 datových typů. Z toho 4 jsou skalární datové typy, 2 tzv. uzavřené datové typy a poslední 2 speciální datové typy. V PHP je práce s datovými typy zjednodušena do té míry, že se o ně prakticky nemusíme starat. Každá proměnná může nabývat hodnot libovolného typu a je dokonce možné během programu do proměnné přiřazovat hodnoty různých datových typů.[1]

Skupina	Datový typ	Hodnota (Popis)
skalární	boolean	true, false
	integer	od -2^{32} do $+2^{32}$
	float	od $-1,7 \cdot 10^{308}$ do $+1,7 \cdot 10^{308}$
	string	textový řetězec
uzavřené	array	pole, speciální datová struktura
	object	proměnná, jejíž typ je určitá třída
speciální	resource	tento typ lze nalézt až od PHP4
	NULL	

Tab. 2 - Datové typy užívané v jazyce PHP

Komentáře

Je dobré když je každý kód dostatečně okomentován, aby kdokoliv kdo ho bude následně upravovat se v něm ihned zorientoval. Komentáře můžeme zapsat do jednoho řádku, pro ty použijeme znaky //, # a nebo použijeme víceřádkové uzavřené v této dvojici znaků /* .. */

```
<?php
    // komentář na jeden řádek
    příkaz1;
    # komentář druhého příkazu
    příkaz2;
    /* komentář víceřádkový
    Příkaz3;
    Příkaz4;
    */
?>
```

2.2 MySQL

SQL je v podstatě strukturovaný dotazovací jazyk, který je nejčastěji používán pro přístup k relačním databázím. Prostřednictvím klientské části systému kontaktujeme serverovou část a zadanými dotazy zjišťujeme potřebné informace, které bývají při výstupu uspořádány do tabulek. Právě tabulka je základním stavebním prvkem SQL databáze. Je tvořena sloupci a řádky a shrnuje přesně specifikované informace, které si od serveru našim dotazem vyžádáme.[5]

2.2.1 Popis MySQL

MySQL je databázový systém, vytvořený švédskou firmou MySQL AB. Jeho hlavními autory jsou Michael "Monty" Widenius a David Axmark. Je považován za úspěšného průkopníka dvojího licencování – je k dispozici jak pod bezplatnou licenci GPL, tak pod komerční placenou licenci. MySQL je multiplatformní databáze. Komunikace s databází probíhá, pomocí jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními. 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, má vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace MySQL, PHP a Apache jako základní software webového serveru.

MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení: má jen jednoduché způsoby zálohování, a až donedávna nepodporovalo pohledy, trigger, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, kdy začaly nejčastějším uživatelům produktu – programátorům webových stránek – již poněkud scházet.

2.2.2 Základy práce s databází a její vytvoření

Na začátku nutné nejdříve vytvořit databázi, ve které můžeme později uchovávat data v tabulkách. K vytvoření a práci s databází se využívají následující příkazy.

```
CREATE DATABASE vyrobky;
```

Příkaz pro vytvoření nové databáze pojmenované vyrobky.

DROP DATABASE friends;

Smazání databáze friends.

SHOW DATABASES;

Vypíše všechny databáze na serveru.

2.2.3 Struktura MySQL

MySQL je systémem správy relačních databází. V MySQL se každá tabulka skládá ze samostatných polí neboli sloupců, které zastupují jednotlivé části uložené informace. existují 3 datové typy:

- číslo
- text
- datum a čas.

Datový typ	Specifikace	Popis	Velikost
číslo	SMALLINT	Číslo v rozmezí -32768 a 32768	2 bajty
	INT	Číslo v rozmezí -214783648 a 214783648	4 bajty
	REAL	Reálné číslo	4 bajty
	FLOAT	Číslo s pohyblivou desetinou čárkou	4 bajty
text	CHAR	Pevná délka řetězce od 0 do 255 znaků	délka
	VARCHAR	Proměnná délka řetězce od 0 do 255 znaků	délka+1 bajt
	TEXT	Proměnná délka řetězce až 65535 znaků	délka+2 bajty
datum a čas	DATE	Formát YYYY-MM-DD	3 bajty
	TIME	Formát HH:MM:SS	3 bajty
	DATETIME	Formát YYYY-MM-DD HH:MM:SS	8 bajtů
	YEAR	Formát YY	2 bajty

Tab. 3 - Vybrané datové typy používané v MySQL

2.2.4 Návrh a vytváření tabulek

K vytvoření tabulky databáze se používá následující příkaz:

```
CREATE TABLE movie (
    movie_name VARCHAR (20),
```

```
movie_type tinyint(2) NOT NULL default 0,  
note TEXT,  
id INT NOT NULL AUTO_INCREMENT,  
PRIMARY KEY (id));
```

MySQL vyžaduje, aby tabulka měla alespoň jeden index, podle něhož, lze vyhledávání výrazně urychlit. Primární klíč je jedinečný identifikátor, který umožňuje jednoznačně určit každý řádek. Sloupec, jenž obsahuje údaje považované za primární klíče, nesmí v žádném řádku obsahovat nejen prázdné, ale i duplicitní hodnoty.

MySQL nabízí funkci `auto_increment`, která umožňuje automaticky zvyšovat hodnotu v daném sloupci s každým novým řádkem. Tento parametr je užitečný zejména v případech, kdy chcete mít jistotu, že bude primární klíč zaplněn pouze jedinečnými čísly.

2.2.5 Základní SQL dotazy

Vložení záznamů do tabulky

Pro vložení nového záznamu do tabulky se používá příkaz `INSERT`. Za tímto příkazem následuje název tabulky, do kterých chceme vkládat hodnoty, poté následuje direktiva `VALUES` a na závěr vyjmenujeme hodnoty přiřazované polím tabulky ve stejném pořadí, v jakém jsme je specifikovali na začátku příkazu. [6]

```
INSERT INTO movie (movie_name, movie_type, note)  
VALUES ('madagaskar', '2', 'komedie');
```

Změna polí na řádku tabulky

Pro editaci hodnot jednotlivých položek tabulky používáme příkaz `UPDATE`. Je nutné uvést, ve které tabulce se mají data upravit. Následuje vyjmenování veškerých změn, které chceme provést. Na konec příkazu se ještě uvádí, kterých řádků tabulky se změny týkají.

```
UPDATE movie SET movie_name = 'jedna', note = 'drama'  
WHERE movie_name = 'madagaskar';
```

Zrušení záznamu v tabulce

Pro zrušení řádku v tabulce používáme příkaz `DELETE`. I zde je nutné uvést, ze které tabulky chceme řádek smazat, a řádky, kterých se změny týkají.

```
DELETE FROM movie WHERE movie_name = 'jedna';
```

Výběr záznamů z tabulky

Nejpoužívanější příkaz v jazyce SQL je příkaz `SELECT`, který slouží pro výběr dat z tabulky.

```
SELECT movie_name, movie_type, note FROM movie
WHERE name = 'jedna';
```

2.2.6 Propojení PHP a MySQL

Pro připojení k databázi se v PHP používá funkce `mysql_connect`. Ta obsahuje parametry: název hostitele, uživatelské jméno a heslo. Dále je ještě nutné zadat jméno databáze, ke které se chceme připojit. K výběru databáze slouží funkce `mysql_select_db`. Propojení PHP a MySQL pak vypadá následovně:

```
<?php
mysql_connect("localhost","uzivatel","heslo");
mysql_select_db("jmeno_db");
?>
```

Pokud připojení dopadlo v pořádku, můžeme začít pracovat s naší databází, kterou jsme vybrali v propojovacím příkazu.

Výpis základních funkcí pro práci s databází MySQL

`mysql_connect()` – připojí se k server MySQL

`mysql_close()` – Ukončí (zavře) MySQL spojení

`mysql_fetch_rows()` - Načte výsledný řádek z databázového dotazu

`mysql_fetch_array()` – Načte několik řádků z výsledného dat. dotazu

`mysql_num_rows()` - Vrací počet všech záznamů ve výsledku.

`mysql_query()` – používá se odeslání příkazu MySQL na server

2.3 AJAX

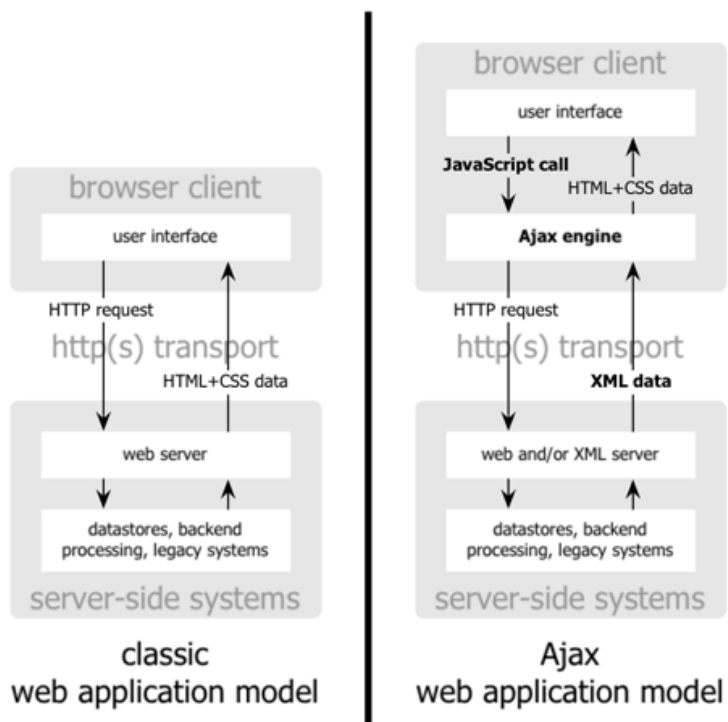
AJAX neboli Asynchronous JavaScript and XML je obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich

znovu načítání. Na rozdíl od klasických webových aplikací poskytují uživatelsky příjemnější prostředí, ale vyžadují použití moderních webových prohlížečů. [9]

- HTML (nebo XHTML) a CSS pro prezentaci informací
- DOM a JavaScript pro zobrazování a dynamické změny prezentovaných informací
- XMLHttpRequest pro asynchronní výměnu dat s webovým serverem

2.3.1 Popis a využití jazyka

Ajax je způsob využití technologie JavaScript, objektu XMLHttpRequest a případně dalších technologií (běžně XML, DOM, CSS) pro vytváření rychlých a uživatelsky přívětivých internetových stránek. Hodí se všude tam, kde probíhá častá komunikace uživatele se serverem a kde by se vždy kvůli nepatrnému množství nových dat nebo obyčejné validaci vstupu musela zbytečně stáhnout celá HTML stránka. Použití AJAXu umožňuje provést základní validaci uživatelského vstupu již v prohlížeči (pomocí JavaScriptu), odeslat data na server, přijmout a popřípadě i zpracovat odpověď a nakonec patřičně změnit obsah stránky. Přitom všechny tyto akce mohou probíhat na pozadí (odtud slovo asynchronous - asynchronní), takže uživatel může mezitím se stránkou dále pracovat nebo být alespoň průběžně informován o průběhu zpracování jeho požadavku. AJAX tak šetří uživatelův čas, snižuje přenos dat a v mnohých případech může i výrazně odlehčit serveru, pokud se podaří část zátěže přenést na webový prohlížeč. Rozdíl mezi tradiční webovou aplikací a webovou aplikací vytvořenou pomocí technologie AJAX je znázorněn na Obr.3



Obr. 3 - Srovnání AJAXu a klasické technologie

Levá strana diagramu popisuje použití klasických webových aplikací při zobrazování stránek. Prohlížeč posílá HTTP požadavky na server, který vrací odpověď v podobě HTML a CSS, které je následně zobrazeno v prohlížeči.

Pravá strana diagramu ukazuje model s využitím interaktivní technologie AJAX. Na první pohled přibyla do části prohlížeče další vrstva nazvaná „Ajax engine“. Požadavky z uživatelského rozhraní jsou převedeny na volání metod JavaScriptu v Ajax engine, který komunikuje se serverem. Tento server potom vrací odpověď ve formátu XML dokumentu. Ajax engine tento dokument následně transformuje prohlížečem zpět na klasické HTML a CSS.

2.3.2 Výhody a nevýhody použití

Mezi výhody patří odstranění nutnosti znovunačtení a překreslení celé stránky při každé operaci, které jsou nutné u klasického modelu WWW stránek. Pokud například uživatel klikne na tlačítko pro udělení hlasu v nějaké anketě, celá stránka se musí znovu načíst ze serveru, třebaže se na ní jen například aktualizují výsledky hlasování a veškerý zbytek obsahu zůstává stejný. Prostřednictvím AJAXu proběhne odeslání hlasu uživatele na pozadí, server zašle jen ty části stránky, které se změnily, a jen tyto části se uživateli na stránce

aktualizují a překreslí. Uživatel tak má pocit mnohem větší plynulosti práce, která se blíží běžným desktopovým aplikacím.

Z toho vyplývá také potenciál snížit zátěž na webové servery a síť obecně. Jelikož není potřeba při každém požadavku sestavit celý HTML dokument, ale pouze provedené změny, je množství vyměňovaných dat výrazně nižší a teoreticky to může mít příznivý vliv i na zátěž databázových serverů či dalších backendových systémů. AJAX však naopak může zvýšit *počet* vyměňovaných HTTP požadavků, třebaže přenášejí nižší množství dat tak při nevhodné implementaci zátěž neklesne.

Mezi nevýhody patří hlavně změny v paradigmatu používání webu: webové stránky se chovají jako plnohodnotná aplikace se složitou vnitřní logikou, nikoli jako posloupnost stránek, mezi kterými se lze navigovat i pomocí tlačítek *Zpět* a *Další*. Moderní AJAXové aplikace jsou schopny funkce těchto tlačítek (přinejmenším částečně) obnovit za použití různých technik (např. využití části adresy za znakem # či pomocí neviditelných IFRAMEs).

Problémem AJAXových aplikací také může být síťová latence: potřeba komunikace přes Internet má negativní dopady na rychlost odezvy a interaktivitu uživatelského rozhraní. Pokud uživateli není jasně signalizováno, že aplikace zpracovává jeho požadavek (a na pozadí komunikuje se serverem), jediné, co zaregistruje, je zpožděná reakce (mezitím se dokonce může snažit operaci spustit znovu, neboť se domnívá, že systém jeho příkaz ignoroval).

Další nevýhodou AJAXu je nutnost používat moderní grafické prohlížeče, které podporují potřebné technologie. Všechny dnešní běžné prohlížeče však tyto technologie alespoň v základu podporují.

2.4 CSS

CSS vznikly někdy v roce 1996 jako reakce na chaoticky se vyvíjející HTML jazyk. Snahou bylo oddělit informaci o obsahu od formátovacích pokynů.

Existují dvě verze CSS. CSS1 (neboli CSS) a CSS2. Někde mezi tím stojí CSS-P (pozicování - přídavek k CSS1). Všechny verze byly napřed v návrhu (draft) a pak se přijala jejich konečná verze (final).

Historie CSS je vlastně historií jejich podpory (a nepodpory) ve webových prohlížečích (protože ta se na rozdíl od standardů mění, jak přicházejí nové prohlížeče). Každý prohlížeč totiž styly interpretuje trochu jinak, některé prohlížeče neznají některé vlastnosti atd.

2.4.1 Popis CSS

CSS je zkratka pro anglický název *Cascading Style Sheets*. Je to jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML.

Jazyk byl navržen standardizační organizací W3C, autorem prvotního návrhu byl Håkon Wium Lie.

Hlavním smyslem je 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. [11]

2.4.2 Vývoj CSS

CSS 1

První velmi rozšířený prohlížeč, který trochu podporoval CSS, byl Internet Explorer verze 3 z roku asi 1996. Podpora CSS se omezovala na písma a barvy.

Podpory se kaskádové styly dočkaly až ve čtvrtých verzích prohlížečů Internet Explorer a Netscape Navigator v roce 1998. Prohlížeče se snažily podporovat celé CSS1 včetně pozicování.

Zejména u Netscapu však nebyla podpora různých CSS vlastností zahrnuta.. To byl hlavní důvod, proč se kaskádové styly dlouho nedostaly do oblíbenosti širokých vrstev webmasterů..

Ani Internet Explorer 4 CSS1 zcela nepodporoval. Obsahoval ale i některé vlastnosti, které v definici CSS1 nejsou (např. filtry, systémová písma a barvy). Dále byla zavedena užitečná pseudotřída `a:hover`, která umožňuje změnu textových odkazů při přejíždění myší. Bohužel žádný z výše jmenovaných prohlížečů neumožňuje speciálně formátovat tzv. pseudoelementy: první písmeno a první řádek elementu, ačkoliv to tehdy CSS1 požadovalo

Až Internet Explorer 5 přišel s lepší podporou CSS. Ve verzi 5.5 se objevily možnosti barvení rolovacích lišt, zvětšování objektů a převrácení textu, což jsou věci, které také do CSS1 nepatřily.

CSS 2

CSS2 byla druhá verze kaskádových stylů. Obsahuje další vlastnosti pro formátování písma, ale i spousty dalších praktických věcí. Konečná verze v roce 2000. CSS2 se začala běžně používat pro formátování XML dokumentů.

Jako předchozí verze i CSS 2 měl problém s podporou u webových prohlížečů. Proto nelze tento styl spolehlivě použít.

Hlavní předností CSS2 jsou různá média, na které se zaměřuje. Kromě obrazovkového výstupu se totiž styly dají použít na odlišný tisk, na zobrazení na mobilech, projekcích, či dokonce na hlasový výstup nebo zařízení pro slepecké písmo.

2.4.3 Syntaxe CSS

Stylový předpis se skládá z posloupnosti pravidel. Každé pravidlo určuje vzhled některého elementu dokumentu, nebo skupiny elementů. Pravidlo začíná tzv. selektorem, který specifikuje („adresuje“) skupinu elementů. Selektor je následován seznamem deklarácí, které určují vzhled vybrané skupiny elementů. Celý seznam je uzavřen ve složených závorkách a jednotlivé deklarace jsou odděleny středníkem. Za poslední deklarací středník už být nemusí.

Příklad stylu:

```
h2 {                               /* vzhled nadpisu první úrovně */
  margin: 5px;                      /* okraj šířky 5 pixelů      */
  font-size: 12pt                   /* velikost fontu 12 bodů   */
}

p {                                  /* styl odstavce          */
  text-align: center;               /* text centrovat         */
  line-height: 10pt;               /* výška řádku 10 bodů   */
}
```

2.4.4 Výhody a nevýhody

Používání kaskádových stylů ve srovnání se samotným HTML v praxi přináší výhody nabízí rozsáhlejší formátovací možnosti než samotné HTML. Např. pro formátování bloku textu – tj. určení vzdálenosti od jejich elementu či okraje stránky nenabízí HTML nic. CSS

má vlastnosti padding a margin. V HTML by bylo potřeba vytvořit složitou konstrukci vnořených tabulek.

Shrnutí výhod:

- rozsáhlejší možnosti formátování
- konzistentní styl
- oddělení struktury a stylu
- dynamická práce se styly
- formátování XML dokumentů
- větší kompatibilita alternativních webových prohlížečů
- kratší doba načítání stránky

Už od počátku používání CSS stylů byla hlavní nevýhodou nedostatečná podpora v majoritních prohlížečích. Hlavní příčinou byly chyby v implementaci CSS. Proto bylo a je nsnadné napsat kód tak, aby se v běžných prohlížečích výsledek zobrazil stejně. Situace se poslední dobou značně zlepšuje, mj. opravami chyb v prohlížečích a nástupem nových verzí Internet Exploreru.

Nedostatečná podpora se dá řešit pomocí různých typů tzv. hacků

Příklad použití:

```
<!--[if IE]>  
<style type="text/css">  
#left {  
left: 150px;  
}  
</style>  
<![endif]>
```

II. PRAKTICKÁ ČÁST

3 TVORBA WEBOVÝCH STRÁNEK

3.1 Původní stránky

3.1.1 Architektura původních stránek

Původní stránky ZO ČSCH Zdounky byly jen tzv. statické stránky. Bylo zde využito technologii HTML a JavaScript. Neexistovala žádná administrační sekce, pomocí které by bylo možné měnit obsah stránek. Veškeré úpravy stránek se prováděly přímou editací zdrojového kódu, což bylo časově náročné a zdlouhavé.



Obr. 4 - Původní vzhled stránek ZO ČSCH Zdounky

3.1.2 Návrh nových stránek

Při návrhu stránek byl hlavní důraz kladen na přehlednost a snadnou orientaci. Na původních stránkách byly některé nedostatky, které bylo nutné odstranit, aby byly výše jmenované podmínky splněny. Zvláště hlavní menu s odkazy na jednotlivé stránky bylo nevýrazné a text splýval s pozadím. Proto se z původního konceptu vycházelo jen zčásti. Třísloupcový layout byl nahrazen dvousloupcovým. Celý layout je tak rozdělen do čtyř hlavních částí. Na

logo stránek s menu, hlavní část pro zobrazení článků a sidebar s přihlašovacím formulářem a hledáním. Dolní část stránek je tvořena footerem s kontaktními údaji. Na design stránky je použita technologie CSS. Samotný design je optimalizován pro korektní zobrazení ve všech běžných prohlížečích, které se dnes používají:

- Mozilla Firefox
- Microsoft Internet Explorer
- Opera
- Safari

Dále byla navržena administrační sekce s různými právy přístupu, která slouží hlavně k aktualizaci stránek a správě uživatelů. Kód stránek je založen na webových technologiích PHP a MySQL, AJAX, HTML a Javascript.

3.2 Návrh a struktura použité databáze

Pro ukládání článků a informací o registrovaných uživateli byla použita relační databáze MySQL. Tento databázový systém je k dispozici jak pod bezplatnou licenci GPL, tak pod komerční placenou licenci a je to jeden nejpoužívanějších databázových systémů. Databáze obsahuje celkem 6 tabulek, z nichž některé jsou propojeny pomocí relací. U každé z tabulek se používá znaková sada utf8. Pro spojení s databází se v projektu využívá soubor conn.php.

3.2.1 Použité tabulky a jejich struktura

Access_levels

Daná tabulka obsahuje 2 sloupce. Slouží pro ukládání dat pro nastavení úrovně uživatelských oprávnění. Obsahuje název úrovně oprávnění a dané číslo úrovně.

Articles

Tabulka obsahuje celkem 7 sloupců. Do této tabulky se ukládají všechny údaje, které se týkají článků. Obsahuje id článku a id autora, který článek napsal. Dále obsahuje datum napsání a vydání článku, nadpis a hlavní text článku. Poslední sloupec obsahuje zda-li byl článek vydán. Engine tabulky je nastaven na typ MYISAM. Tento typ umožňuje fulltextové vyhledávání v člancích. Sloupce title a body jsou označeny jako FULLTEXT KEY. To

znamená, že při fulltextovém vyhledávání na stránkách se vyhledává právě v těchto sloupcích. Dále jsou sloupce `author_id` a `date_submitted` označeny klíčovou frází KEY. To umožňuje mnohem rychlejší a efektivnější vyhledání dat v databázi. Podle těchto tabulek se na stránkách vyhledávají články.

Comments

Tabulka obsahuje celkem 5 sloupců. Slouží pro ukládání komentářů ke článkům. Obsahuje id komentáře, id článku, datum, jméno autora a vlastní text komentáře.

Katalog

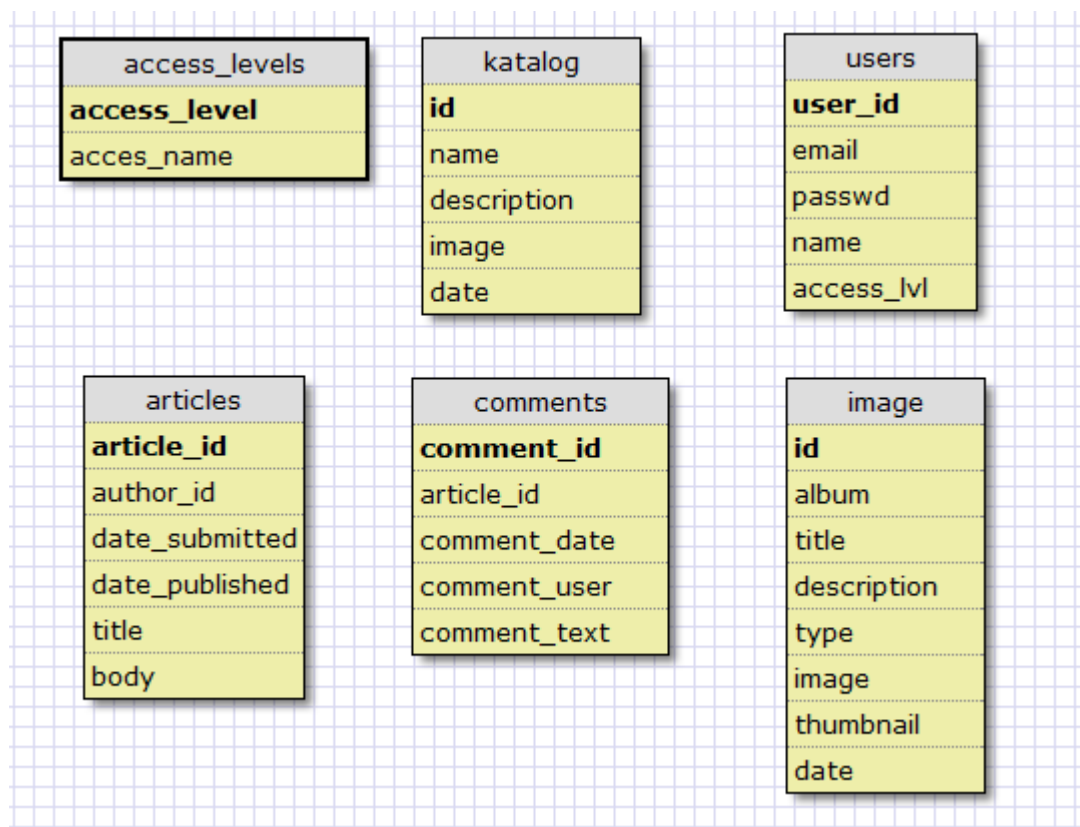
Tabulka slouží k ukládání informací o jednotlivých skupinách prodejních zvířat. Obsahuje celkem 5 sloupců. Do sloupců se ukládá id skupiny, název skupiny, popis, datum vytvoření a název souboru obrázku, který charakterizuje skupinu zvířat.

Image

Tato tabulka obsahuje informace o obrázcích, které spadají pod určitou skupinu prodejních zvířat. Obsahuje 8 sloupců. Do těch se ukládá id skupiny, id obrázku, nadpis obrázku. Dále popis obrázku zvířete, typ souboru obrázku, datum vložení, název souboru obrázku a název souboru náhledu obrázku.

Users

Tato tabulka slouží pro ukládání registrovaných uživatelů a jejich informací. Obsahuje celkem 5 sloupců, do kterých se ukládá id a jméno uživatele, email, heslo a úroveň oprávnění uživatele. Při vytváření tabulky byl vytvořen rovněž jedinečný klíč – UNIQUE KEY pro zajištění neopakovatelnosti adresy elektronické pošty.



Obr. 5 – Struktura tabulek databáze

3.3 Popis a struktura stránek

Každá stránka obsahuje odkaz na externí soubor s definicí CSS stylů. Pomocí těchto stylů lze umisťovat jednotlivé prvky na stránku, formátovat a vytvářet konečný vzhled stránek. Většina stránek se také skládá z několika externích souborů, které se vždy připojují k dané stránce pomocí příkazu *require*. Tohoto příkladu se využívá, pokud se chceme například spojit s databází nebo chceme zpracovat data získané z formuláře.

3.3.1 Popis jednotlivých odkazů menu

Aktuality

Tato stránka je nastavena jako výchozí domovská stránka. V hlavní části této stránky se nachází články, které se zde vypisují z databáze podle datumu vložení od nejnovějšího po starší s možností stránkování. Dále zde najdeme navigační menu pro prohlížení stránek, formulář pro přihlášení registrovaných uživatelů do administrační sekce, políčko pro vyhledávání článků, logo chovatelského spolku a kontakty na členy.

Výstavy

V této části uživatel získá informace o chystaných výstavách pořádaných místním spolkem chovatelů popř. jiným spolkem. Seznam výstav je generován z databáze. Pro jejich zobrazení slouží soubor `vystavy.php`.

Prodejná zvířata

Tato část patří katalogu prodejních zvířat od chovatelů Zdounecka. Návštěvníci mají k dispozici stránku se všemi skupinami prodejních zvířat jejich popisem. Po kliknutí na jednotlivou skupinu se uživateli zobrazí seznam všech zvířat s náhledy fotografií. Pokud se chce uživatel dozvědět podrobnosti o jednotlivém zvířeti, stačí mu kliknout na odkaz pod jednotlivými zvířaty. Tím se mu zobrazí stránka s detailní fotografií a popisem zvířete. Nad galerií se navíc zobrazuje odkaz, v jaké skupině se uživatel nachází, což ulehčuje orientaci uživatele.

Fotogalerie

Na této stránce si mohou uživatelé prohlédnout fotografie pořízené chovateli Zdounecka. Nachází se zde náhledy jednotlivých fotografií. Při kliknutí na náhled fotografie se otevře originál obrázku. Po té je možné zobrazovat jednotlivé fotografie pomocí šipek v okně.

Chovatelé

Na stránkách si mohou uživatelé prohlédnout tabulku se všemi současnými chovateli spolku ale i tabulku s čestnými členy spolku. Tabulka navíc podporuje třídění. To znamená, že pokud uživatel klikne na políčko jméno v tabulce, seřadí se záznamy v tabulce vzestupně nebo sestupně podle abecedy.

Odkazy

Na této stránce se jsou vypsány odkazy jiných spolku chovatelů(viz. Obr. 6)

Odkazy

- ♦ [Chovatelské stránky](#)
- ♦ [český svaz chovatelů drobného zvířectva](#)
- ♦ [Slovenský zväz chovateľov](#)

Obr. 6 - Stránka s odkazy

4 ADMINISTRAČNÍ SEKCE

4.1 Popis administrační sekce

Jednotlivé uživatelské účty pro správu webové prezentace vytváří hlavní administrátor. Při registraci nového uživatele zadá administrátor do formuláře jeho jméno, email a vybere mu příslušná přístupová práva. Následně jsou na zadanou emailovou adresu odeslány automaticky vygenerované přihlašovací jméno a heslo pro vstup do systému, které se skládají z písmen a čísel. V tomto systému existují dvě různá přístupová práva.

Práva	Popis
2	Administrátor
1	Autor

Tab. 4 - Přehled přístupových práv

Úroveň přístupových práv je nastavena následovně. Uživatelé s přidělenými právy 1 mají pouze možnost psát články upravovat svůj profil, přidávat nové skupiny do katalogu prodejních zvířat, případně přidávat výstavy.

Uživatel s úrovní práva 2 má stejné pravomoce jako uživatel s úrovní 1. Navíc mohou přidávat nové uživatele do systému, editovat nebo mazat tyto uživatele.

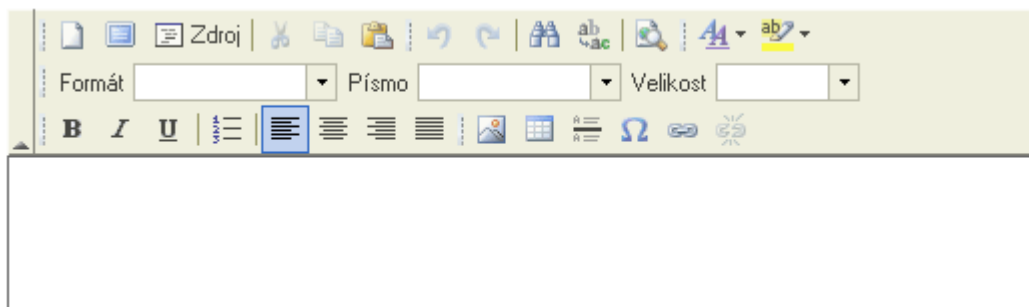
Na každé stránce v administrační sekci se kontroluje pomocí proměnné `$_SESSION["access_lvl"]`, která byla registrována při úspěšném přihlášení uživatele do administrace, zda má uživatel právo vstupu na danou stránku nebo nikoliv. Pokud uživatel nemá povolen přístup, je automaticky přesměrován na úvodní stránku administrace.

Dle těchto přístupových práv se také generuje ovládací menu. Každá skupina má své menu a nevidí položky druhé skupiny, které nemají spolu společné.

4.1.1 Editace článků pomocí FCK editoru

Pro lepší je pro editaci článků a aktualit použit FCK editor. Jedná se o tzv. WYSIWYG editor v překladu „co vidíš to dostaneš“. Je to principem podobný textový editor jako například známý Microsoft Word. Napsaný a nastýlovaný text překládá po zpracování do HTML kódu. Uživatel tedy uvidí úplně stejný text, tak jak ho napsal sám autor. Editor na-

víc podporuje přímý upload souboru na server. Odpadá tak nahrávání souborů pomocí formuláře na samostatné webové stránce.



Obr. 7 - Náhled FCK editoru pro editaci článků

4.1.2 Přihlášení do administrační sekce

Přihlášení do administrace probíhá pomocí formuláře, do kterého uživatel zadá svoje přihlašovací jméno a heslo, které mu při registraci přišly na email, nebo které si již změnil podle svých potřeb při prvním přihlášení do systému. Údaje se odešlou a následuje zpracování, které provádí samostatný přihlašovací skript. Ten se připojí k MySQL databázi, ve které se ověřují zadané údaje. Před tím se ještě zašifruje zadané heslo pomocí hashovací funkce MD5, protože hesla v databázi jsou uložena v tomto formátu. Následuje hledání údajů v databázi. Pokud není nalezen žádný shodný údaj, vypíše se na obrazovku chybová hláška. V opačném případě se z databáze vyberou veškeré údaje o právě přihlašovaném uživateli (id uživatele, jméno, email, přístupová práva, ...), které se uloží do objektu `$_SESSION[“”][..]`. S tímto se potom dále pracuje na ostatních stránkách.



Přihlášení uživatele

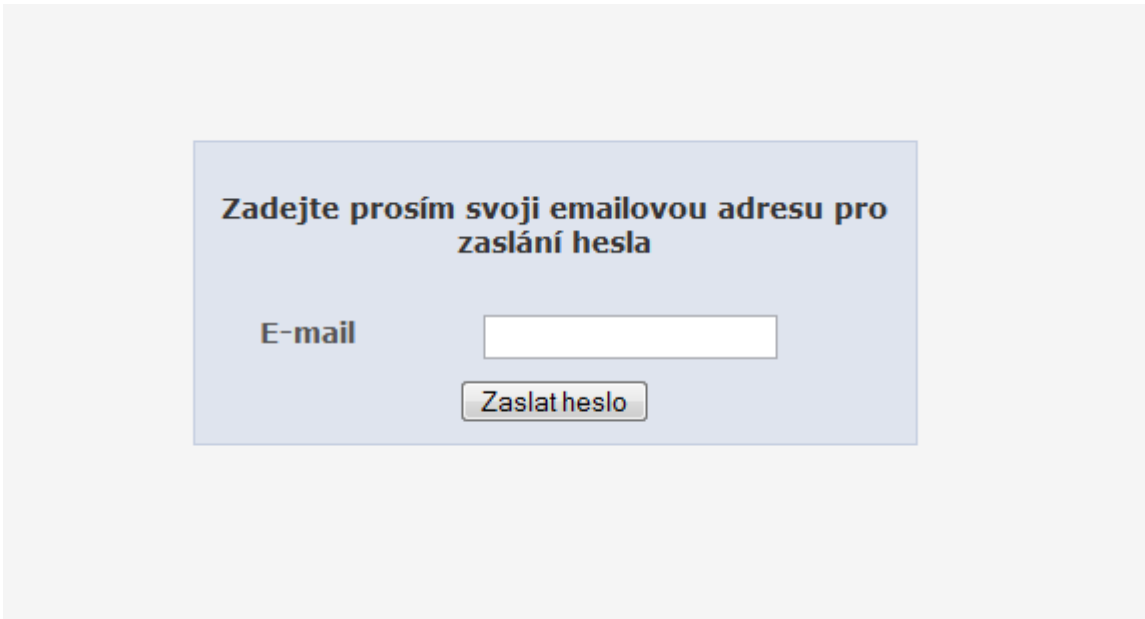
E-mail

Heslo

[Zapomněli jste heslo?](#)

Obr. 8 - Přihlášení do administrace

Pokud uživatel zapomene heslo pro přihlášení do administrace, má možnost si požádat o vygenerování nového hesla. Jedinou podmínkou je vyplnění formuláře, do kterého zadá svůj kontaktní email, který má uveden ve svém účtu. Po odeslání se v systému ověří, zda jsou zadané údaje obsažené v databázi. Následně, pokud je vše v pořádku, vygeneruje systém nové heslo. To se následně odešle na uživatelův email a opět za hashované se uloží do databáze.



Obr. 5 - Formulář pro zaslání hesla

4.1.3 Správa uživatelských účtů

Správu uživatelů může provádět jen administrátor. Administrátor může měnit přístupová práva uživatele nebo jeho účet natrvalo smazat. Přístupová práva se mění vybráním požadované úrovně oprávnění ve formuláři, kde může administrátor následně vybrat nová přístupová práva. Dále je možné měnit uživatelské jméno ale i heslo. Nové údaje se po klepnutí na tlačítko změnit údaje uloží do databáze. Důležitou věcí je, že administrátoři si navzájem nemohou své účty blokovat ani smazat.

4.1.4 Články

Uživatelé s právy 1 a 2, mají možnost vkládat a editovat nové články. Při vkládání článků je použit již zmiňovaný FCK editor, pomocí něhož mohou uživatelé své aktuality různě zvýrazňovat, vkládat do nich obrázky apod. Pokud je článek vytvořen, uloží se jako návrh do databáze a zobrazí se v nabídce jako článek ke zveřejnění. V té době je možné článek ještě pořád editovat nebo smazat. Po té, co je článek zveřejněn, jsou aktualizována data v databázi o článku. Takový článek již nelze smazat, ale jde stáhnout, čímž se dostane opět do kategorie nezveřejněných článků. Pokud zůstane článek zveřejněn, zobrazuje se v nabídce zveřejněné články.



Obr. 60 - Přehled všech článků

4.1.5 Změna uživatelského profilu

Každý uživatel, který má vytvořený účet, má možnost upravovat svůj profil. Jedná se především o úpravy jména, přihlašovacího jména, emailu a hesla. Pokud si uživatel vybere přihlašovací jméno, které již existuje, systém mu nedovolí své úpravy uložit. Při změně hesla je uživatel vyzván pro opakování nově vkládaného hesla. Pokud si daná hesla neodpovídají, systém opět zahlásí chybu.

4.2 Katalog prodejních zvířat

4.2.1 Vložení nové skupiny zvířat

Vkládání do tohoto katalogu mají jak autoři, tak i administrátor. (Viz. Tab.4). Pro vložení nové skupiny musí uživatel nejprve zadat název nové kategorie, kterou chce uložit. Dále vyplní název a krátký popis skupiny zvířat. Uživatel také může vložit ke každé skupině fotografii. Ta se při zpracování, zmenší na požadovanou velikost a bude zobrazena jako fotografie reprezentující danou skupinu.

4.2.2 Vložení nové zvířete do skupiny

Každý registrovaný uživatel může také přidávat jednotlivá zvířata v rámci skupiny. Pro vložení zvířete je nutné zadat skupinu, název zvířete, popis a případně i fotografii zvířete. Po té je zvíře uloženo do databáze v rámci určité skupiny.

4.2.3 Úprava a mazání skupin a zvířat

Operaci může provádět autor i administrátor. Pro úpravu stačí kliknout na odkaz upravit skupinu. Následně se zobrazí formulář vyplněný údaji z databáze a náhled obrázku. Uživatel tak vidí přímo prováděné změny. Po potvrzení jsou změny uloženy do databáze. Pro smazání skupiny stačí kliknout na odkaz smazat skupinu. Před úplným vymazáním údajů z databáze je nutné potvrdit ještě jednou akci smazání. Po potvrzení jsou údaje skupiny z databáze vymazány. Ten samý postup platí i pro jednotlivá zvířata ve skupině.

4.2.4 Vyhledávání v článcích

Na stránkách má každý návštěvník možnost hledat v článcích. K tomuto účelu slouží jednoduchý formulář s políčkem pro zadání hledaného výrazu, který se nachází na hlavní stránce. K vyhledávání dojde, pokud není vyhledávací pole prázdné. Pokud se tak stane, bude uživatel upozorněn, aby zadal hledaný výraz a hledání opakoval. Jestliže byl nalezen nějaký záznam, zobrazí se výsledek hledání na stránce. V opačném případě se zobrazí oznámení, že nebyl nalezen žádný článek.



Hledat v článcích

Obr. 71 - Vyhledávání v článcích

ZÁVĚR

Hlavní náplní této bakalářské práce bylo vytvořit moderní a dynamické stránky pro ZO ČSCH Zdounky. Stránky, které by přilákaly nové návštěvníky a ulehčily práci správcům webu.

Teoretická část této práce se zabývá historií vzniku internetu a používanými technologiemi pro tvorbu webových stránek. Dále je popisována technologie AJAX a technologie CSS, která se používá na grafický návrh webových stránek.

V praktické části byla řešena samotná realizace projektu. Nejprve byla navrhnutá grafická podoba stránek, poté rozmístění jednotlivých prvků na stránce. Následoval návrh databáze a tabulek se sloupci. Dynamičnosti stránek bylo dosaženo pomocí jazyka PHP.

Webové stránky jsou připraveny na další vývoj do budoucna, kterým se hodlám i nadále zabývat.

Vytvořené webové stránky jsou zcela funkční a splňují všechny zadané požadavky. V systému je dále možné provádět změny, vkládat a psát nové články atd.

CONCLUSION

The main aim of the thesis was to create modern and dynamic web pages for ZO ČSCH Zdounek. The web pages which would bring up new visitors and make work easier to administrator of pages.

The theoretical part describe the history of internet and the using of technologies for creation of the web pages. The last part theoretical part describe web technology AJAX and technology CSS for creation of layout in web pages.

The practical part deals with the project realization. The first step was to design the graphic visage of the web pages followed by position of individual elements. Next step was to design the database and needed tables with columns. The dynamics of the pages was reached by using PHP scripting language.

The created web pages are completely functioning and complies with all requirements. It is also possible to make the modification, insert a write new articles and so on.

SEZNAM POUŽITÉ LITERATURY

- [1] TANSLEY, David. *PHP a MySQL Vytváříme Dynamické Webové Stránky*, SoftPress, 2003, 480 s, ISBN 80-86497-40-2
- [2] WELLING, Luke, THOMSON, Laura. *PHP a Mysql - Rozvoj webových aplikací*, SoftPress, 2005, 912 s, ISBN 80-86497-60-7
- [3] KOSEK, Jiří. *PHP - tvorba interaktivních internetových aplikací*, Grada Publishing, 1998, 492 s, ISBN 80-7169-373-1
- [4] MACH, Jakub. *PHP pro úplné začátečníky*, CP Books, 2005, 167 s, ISBN 80-7226-834-1.
- [5] WELLING, Luke, THOMSON, Laura. *MySQL - Průvodce základy databázového systému*, CP Books, 2005, 255 s, ISBN 80-251-0671-3.
- [6] Kolektiv autorů. *PHP5, MySQL, Apache - Vytváříme webové aplikace*, CP Books, 2006, 816 s, ISBN 80-251-1073-7
- [7] MEYER, Eric. *Eric Meyer o CSS - Ovládněte kaskádové styly*, Zoner Press, 2004, 336 s, ISBN 80-86815-03.
- [8] *Wikipedie, otevřená encyklopedie* [online]. [cit. 2008-04-19]. Dostupný z WWW: <cs.wikipedia.org>
- [9] *World Wide Web Consortium* [online]. 15.ledna 2001 [cit. 2008-04-19]. Dostupný z WWW: <www.w3.org>
- [10] *Historie PHP a souvisejících projektů* [online]. [cit. 2008-04-19]. Dostupný z WWW: <http://php.mirror.camelnetwork.com/manual/cs/history.php>.
- [11] *Cascading Style Sheets* [online]. [cit.2008-04-19]. Dostupný z WWW: < http://cs.wikipedia.org/wiki/Cascading_Style_Sheets>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AJAX	Asynchronous JavaScript and XML Technologie pro vývoj interaktivních webových aplikací
ASCII	American Standard Code for Information Interchange Tabulka definující anglické znaky a jiné další znaky používané v informatice
ASP.NET	Active Server Pages Technologie pro vývoj webových aplikací
CERN	European Organization for Nuclear Research Evropská organizace pro jaderný výzkum
CSS	Cascading Style Sheets Jazyk pro formátování internetových stránek
DOM	Document Object Model Objektově orientovaná reprezentace XML nebo HTML dokumentu
HTML	Hyper Text Markup Language Jazyk pro vytváření a publikování webových stránek
HTTP	Hyper Text Transfer Protocol Internetový protokol původně určený pro výměnu hypertextových dokumentů
IETF	Internet Engineering Task Force Otevřená organizace starající se o rozvoj internetu a definice standardů
INRIA	Institut national de recherche en informatique et en automatique Institut pro vývoj www
NCSA	National Center for Supercomputer Applications Mezinárodní centrum pro superpočítačové aplikace

PHP	Hypertext Preprocessor, původně Personal Home Page Technologie pro vývoj webových aplikací
SGML	Standard Generalized Markup Language Systém pro definici strukturovaných dokumentů
URL	Uniform Resource Locator Řetězec znaků s definovanou strukturou sloužící přesné specifikaci umístění zdrojů informací.
W3C	World Wide Web Consortium Mezinárodní konsorcium vyvíjející webové standardy
WYSIWYG	What you see is what you get Způsob editace dokumentů, při kterém je verze zobrazená na obrazovce vzhledově totožná s výslednou verzí dokumentu.
WWW	World Wide Web Celosvětová soustava propojených hypertextových dokumentů

SEZNAM OBRÁZKŮ

<i>Obr. 1 - První webový prohlížeč Nexus</i>	<i>9</i>
<i>Obr. 2 - Webový prohlížeč Mosaic.....</i>	<i>10</i>
<i>Obr. 3 - Srovnání AJAXu a klasické technologie.....</i>	<i>24</i>
<i>Obr. 4 - Původní vzhled stránek ZO ČSCH Zdounky</i>	<i>30</i>
<i>Obr. 5 – Struktura tabulek databáze.....</i>	<i>33</i>
<i>Obr. 6 – Stránka s odkazy.....</i>	<i>35</i>
<i>Obr. 7 – Náhled FCK editoru pro editaci článků.....</i>	<i>37</i>
<i>Obr. 8 – Přihlášení do administrace</i>	<i>38</i>
<i>Obr. 9 - Formulář pro zaslání hesla.....</i>	<i>39</i>
<i>Obr. 10 - Přehled všech článků.....</i>	<i>40</i>
<i>Obr. 11 - Vyhledávání v člancích</i>	<i>41</i>

SEZNAM TABULEK

<i>Tab. 1 - Přehled začínajících a ukončovacích PHP tagů</i>	17
<i>Tab. 2 - Datové typy používané v PHP</i>	18
<i>Tab. 3 - Vybrané datové typy používané v MySQL</i>	20
<i>Tab. 4 - Přehled přístupových práv</i>	36

SEZNAM PŘÍLOH

- P I MySQL příkazy pro vytvoření databáze
- P II Zdrojový kód souboru pro připojení k databázi (*conn.php*)
- P III Kód pro připojení a vytvoření FCK editoru

PŘÍLOHA P I: MYSQL PŘÍKAZY PRO VYTVOŘENÍ DATABÁZE

```
CREATE TABLE IF NOT EXISTS articles (  
  article_id int(11) NOT NULL auto_increment,  
  author_id int(11) NOT NULL default '0',  
  is_published tinyint(1) NOT NULL default '0',  
  date_submitted datetime NOT NULL default '0000-00-00 00:00:00',  
  date_published datetime NOT NULL default '0000-00-00 00:00:00',  
  title varchar(255) NOT NULL default "",  
  body mediumtext NOT NULL,  
  PRIMARY KEY (article_id),  
  KEY IdxArticle (author_id,date_submitted),  
  FULLTEXT KEY IdxText (title,body)  
) ENGINE=MYISAM
```

```
CREATE TABLE IF NOT EXISTS access_levels (  
  access_lvl tinyint(4) NOT NULL auto_increment,  
  access_name varchar(50) NOT NULL default "",  
  PRIMARY KEY (access_lvl)  
)
```

```
CREATE TABLE IF NOT EXISTS comments (  
  comment_id int(11) NOT NULL auto_increment,  
  article_id int(11) NOT NULL default '0',  
  comment_date datetime NOT NULL default '0000-00-00 00:00:00',  
  comment_user int(11) NOT NULL default '0',  
  comment text NOT NULL,  
  PRIMARY KEY (comment_id),  
  KEY IdxComment (article_id)  
)
```

```
CREATE TABLE IF NOT EXISTS users (  
  user_id int(11) NOT NULL auto_increment,  
  email varchar(255) NOT NULL default "",  
  passwd varchar(50) NOT NULL default "",  
  name varchar(100) NOT NULL default "",  
  access_lvl tinyint(4) NOT NULL default '1',  
  PRIMARY KEY (user_id),  
  UNIQUE KEY uniq_email (email)  
)
```

```
CREATE TABLE katalog (  
  id INT NOT NULL AUTO_INCREMENT,  
  name VARCHAR(64) NOT NULL,  
  description TEXT NOT NULL,  
  image VARCHAR(64) NOT NULL,  
  date DATETIME NOT NULL,  
  PRIMARY KEY(al_id)  
);
```

```
CREATE TABLE image (  
  im_id INT NOT NULL AUTO_INCREMENT,  
  im_album_id INT NOT NULL,  
  im_title VARCHAR(64) NOT NULL,  
  im_description TEXT NOT NULL,  
  im_type VARCHAR(30) NOT NULL,  
  im_image VARCHAR(60) NOT NULL,  
  im_thumbnail VARCHAR(60) NOT NULL,  
  im_date DATETIME NOT NULL,  
  PRIMARY KEY(im_id)  
);
```

PŘÍLOHA P II: ZDROJOVÝ KÓD SOUBORU PRO PŘIPOJENÍ K DATABÁZI (CONN.PHP)

```
<?php

// údaje pro připojení k databázi

define('SQL_HOST','localhost'); // připojení k databázi

define('SQL_USER','root');

define('SQL_PASS','');

define('SQL_DB','cms');

$conn = mysql_connect(SQL_HOST, SQL_USER, SQL_PASS)

or die('Vyskytla se následující chyba: Nelze se pripojit k databázi; ' . mysql_error());

// nastaveni znakové sady databáze

mysql_query("SET CHARACTER SET utf8");

mysql_query("SET NAMES utf8");

// výběr databáze

mysql_select_db(SQL_DB, $conn)

or die('Vyskytla se následující chyba: Nejde vybrat databáze; ' . mysql_error());

// výběr vhodné databáze

?>
```

PŘÍLOHA P III: KÓD PRO PŘIPOJENÍ A VYTVOŘENÍ FCK EDITORU

```
<?
    // pripojeni externiho souboru
    include("fckeditor/fckeditor.php") ;

    // vytvoreni nove instance artikle_cs
    $oFCKeditor = new FCKeditor('article_cs') ;

    // nastaveni cest
    $oFCKeditor->BasePath= $trackfckeditor;

    //nastaveni maximalni vysky editoru
    $oFCKeditor->Height= '400px';

    // nazev panelu nastroju, který se ma nacist
    $oFCKeditor->ToolbarSet = 'panel';

    // vytvoreni FCK editoru
    $oFCKeditor->Create() ;
?>
```

