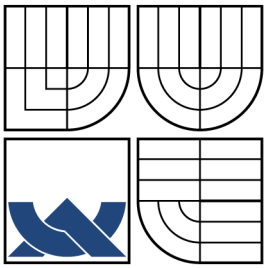
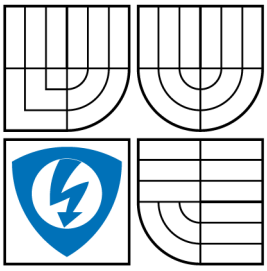


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH A REALIZACE SYSTÉMU PRO SPRÁVU OBSAHU WWW STRÁNEK

DESIGN AND IMPLEMENTATION OF SYSTEM FOR WEB CONTENT ADMINISTRATING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — SZABOLCS GARAI
AUTHOR

VEDOUCÍ PRÁCE — ING. PETRA LAMBERTOVÁ
SUPERVISOR

BRNO 2007

ZDE VLOŽIT LIST ZADÁNÍ

Z důvodu správného číslování stránek

ZDE VLOŽIT PRVNÍ LIST LICENČNÍ SMOUVY

Z důvodu správného číslování stránek

ZDE VLOŽIT DRUHÝ LIST LICENČNÍ SMOUVY

Z důvodu správného číslování stránek

ABSTRAKT

Práca sa zaoberá návrhom a realizáciou systému pre správu obsahu webových stránok a nástrojmi Apache, MySQL a PHP použitých pre chod takéhoto systému. Obsahovo je rozdelená na viacej častí. Postupne popisuje požiadavky systému pre správu, použité technológie pre zrelizovanie a nakoniec programovaciu časť a zabezpečenie. Počítač, s ktorým sme pracovali bežal na operačnom systéme Microsoft Windows VISTA.

KĽÚČOVÉ SLOVÁ

systém pre správu obsahu, dynamické web sránky, server, PHP, MySQL, Apache, https, SSL

ABSTRACT

Theme of my work is design and implementation of system for administrating web content and tools Apache, MySQL and PHP used for running of this system. The work consist of more parts. It describes demands of content management system, used technologies for realisation and for end progamming part and security. I worked with computer running on Microsoft Windows VISTA.

KEYWORDS

content management system, dynamic web pages, server, PHP, MySQL, Apache, https, SSL

GARAI SZ. *Návrh a realizace systému pro správu obsahu www stránek*. Místo: Vysoké Učení Technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací, 2007. 49 s., 7 s. příloh. Bakalárska práca. Vedoucí práce byl Ing. Petra Lambertová.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Návrh a realizace systému pro správu obsahu www stránek“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

(podpis autora)

POĎAKOVANIE

Touto cestou by som sa chcel poďakovať každému, kto mi nejakou formou pomohol pri riešení môjho zadania. Najväčšia vďaka patrí mojej konzultantke Ing. Petre Lambertovej, ktorá mi pomáhala pri zostavovaní a realizácii tejto práce.

V Brne dňa .....

.....

(podpis autora)

OBSAH

ÚVOD

V dnešnej dobe si uživateľ môže obsah internetovej prezentácie meniť vlastnými silami, bez toho aby vedel takéto prezentácie vytvárať. Slúžia k tomu systémy, ktorým sa súhrnne hovorí systémy pre správu obsahu (CMS), či nepresnejšie, v zúženom význame redakčné systémy. Pre návštevníka stránok je dôležité, aby obsah prezentácie bol aktuálny, zaujímavý a presný. Web so zastaralým alebo dokonca chybným obsahom je potom nielen zbytočný, ale vyvoláva negatívny dojem. Práve preto sa vyvýjajú systémy, v ktorých je údržba a aktualizácia obsahu velmi jednoduchá. Dokážu previesť dokumenty priamo do HTML formátu a tým sa výrazne skracuje proces publikovania informácií. Vzhľad stránok sa zachováva a autor sa teda môže plno sústrediť na samotný obsah. S tymito výhodami sa ušetria náklady spojené so spravovaním web stránky.

Dobre a kvalitne spracovaný systém je základným kameňom webových portálov, či elektronických obchodov. Pri programovaní redakčného systému je dobré sa najkôr zoznámiť so základnými požiadavkami takéhoto systému a potom ho navrhnúť podľa vlastných potrieb.

Na nasledujúcich stránkach sa bude čerpať z vedomostí, ktoré boli pozbierané pri realizácii CMS. Postupne sa prejde všetkým tým, čo je potrebné na vytvorenie takéhoto systému. Bude sa pracovať vyložene s open source technológiami. Práca sa nebude zaoberať základmi HTML kódu ani s kaskádovými štýlmi.[1] Pri tvorbe webových stránok za použitia bežného HTML, sa vyskytli mnohé obmedzenia, hlavne v statickom výstupe. Dynamický obsah sa vytvorí pomocou jazyka PHP a MySQL databázy. Syntax je podobná ako v iných moderných programovacích jazykoch, takže nieje problém sa rýchlo zorientovať.

Na začiatku budú podrobne popísané vlastnosti CMS. Ďalej sa charakterizuje jazyk PHP v spojení s MySQL databázou a základné zabezpečenie. Ukáže sa, ako vytvoriť na vlastnom počítači webový server za účelom testovania vytvorených dynamických stránok. Popritom sa preberú aj ostatné podrobnosti súvisiace s touto problematikou. V poslednej časti sa zhotoví samotný redakčný systém, popíšu jeho časti, funkcie a ochrana. Táto práca by mala priniesť nové poznatky a uľahčiť tak návrh a vytvorenie vlastného systému pre správu obsahu.

[1] Kaskádové štýly alebo CSS (skratka z angl. Cascading Style Sheets) je všeobecné rozšírenie HTML. Konzorcium W3C označuje CSS ako jednoduchý mechanizmus na vizuálne formátovanie internetových dokumentov. Štýly umožňujú oddeliť štruktúru HTML alebo XHTML od vzhľadu. Získa sa tým prehľadný a jednoduchý kód. CSS zaručuje rovnaké vykresľovanie vo všetkých prehliadačoch.

1 SYSTÉM PRE SPRÁVU OBSAHU

CMS je skratka anglického výrazu Content Management System, alebo systém pre správu obsahu. Je to softvérové riešenie slúžiace na organizáciu a jednoduché vytváranie dokumentov a rôzneho iného obsahu. Obsah môže pozostávať z textov, obrázkov, zvukov, videí a iných mediálnych elektronických súborov. Úlohou CMS systému je prehladne spracovať obsah rôzneho druhu a umožniť viacerým stranám prístup k istým materiálom. Takéto riešenie značne uľahčuje komunikáciu medzi používatelmi. Jediná správna definícia CMS neexistuje a rovnako nároky uživatelov sú značne odlišné. Neexistuje ani žiadny štandardný systém, ktorý by splňoval všetky požiadavky.

CMS nájde uplatnenie najmä vo firmách, kedže jednou z výhod je automatizácia firemných procesov, alebo workflow.[1] Pomocou tohto systému sa cez rôznych ľudí vo firme môže pohybovať materiál, ktorý možno pripomienkovať, meniť, alebo inak spoločne modifikovať. Často má CMS využitie aj ako nástroj na archiváciu dokumentov. Túto možnosť bežne využívajú mediálne firmy (televízie, vydavateľstvá atď.), ktorých predmet podnikania je priamo spojený s poskytovaním obsahu. CMS systémy však používajú aj všetky veľké medzinárodné spoločnosti na správu svojej firemnej komunikácie. Táto funkcia je však čím ďalej tým dôležitejšia aj pre malé a stredné firmy.

CMS systémy sa delia na štyry základné typy:

1. **WCM** (Webový CMS) - pomáha automatizovať vývoj v rôznych oblastiach webovej publikácie
2. **TCM** (Transakčný CMS) - využíva sa v oblasti e-komercie
3. **ICM** (Integrovaný CMS) - slúžiaci na spracovanie dokumentov a obsahu v oblasti podnikania
4. **PCM** (Publikačný CMS) - pomáha spracovať vývoj obsahu publikácií (manuály, knihy, ...)

1.1 Webový CMS

Dnes sa pre tento systém používa skratka WCM z anglického Web Content Management, alebo tiež publikačný či redakčný systém (ďalej RS). Táto časť CMS systému

[1] Workflow je schéma prevádzania nejakej komplexnejšej činnosti, rozpísanej na jednoduchšie činnosti a ich väzby. Obvykle se týmto pojmom popisuje technológia riadenia podnikov, projektov, či spracovanie dokumentov.

priamo súvisí s tvorbou, správou, organizáciou a publikovaním prezentácií vo forme internetových stránok. Najdôležitejšia vec, ktorú priniesli aplikácie CMS pre publikovanie na webe, je oddelenie dizajnu (grafickej úpravy) a funkcionality od obsahu. Obsah stránok je uložený v databázi, pričom grafická úprava a štruktúra stránok sú uložené zvlášť v špeciálnych šablónach. Šablóny sú vzorové HTML (značkovací jazyk pre hypertext – HyperText Markup Language) dokumenty, do ktorých sa vkladá na predom označené miesta obsah z databázy. Obsahová zmena si teda nevynúti zmenu dizajnu alebo opravu ostatných informácií. Okrem zverejňovania informácií na internete môže slúžiť aj ku správe dát. Uživatelia sa potom nestrácajú v dokumentoch a nájdu ich jednoducho a rýchlo na serveri, namiesto klasického archívu. Webové rozhranie sa z hľadiska spravovania delí na administrátorské a uživatelské. Vstup býva zabezpečený menom a heslom.

Vetšina týchto systémov sa skladá z častí, tzv. modulov. Moduly sú časti CMS systému, o ktoré sa dá rozšíriť základná verzia. Po prihlásení do systému sa vyberie modul, prostredníctvom ktorého sa vloží článok alebo informácia (napr. modul volné miesta, modul novinky atd.). Samotný článok sa pritom môže tvoriť a upravovať buď doplnkovým programovým vybavením, alebo priamo v editore s webovým rozhraním, ktorý býva súčasťou CMS. Vďaka tomu má uživatel možnosť rýchlo a jednoducho meniť obsah stránok, udržiavať ich aktuálne a to bez odborných technických znalostí. Obsah článku sa po uložení (publikácii) naplní do šablón, pomocou ktorých sa automaticky vytvorí štruktúra stránok a zobrazí návštevníkom, takže sa používatel môže zaoberať iba samotnou náplňou obsahu. Výhodou šablón nieje len to, že zaručujú rovnaký vzhľad všetkých stránok, ale významne zjednodušujú údržbu. Keď je potreba uskutočniť úpravu na všetkých stránkach, napr. výmena loga, stačí ju previesť v jedinej šablóne miesto úpravy všetkých stránok. Niektoré CMS umožňujú zobrazovať rovnaký obsah cez rôzne šablóny a tak ho publikovať v rôznych formátoch. Môžeme tak prevádzkovať okrem štandardnej HTML verzie aj verzie pre prenosné zariadenia, alebo exportovať dáta do XML pre ďalšie spracovanie.[2]

1.2 Základné požiadavky

Pri programovaní vlastného systému je výhoda, že bude fungovať presne podľa naších predstáv. Vyvynúť vlastný, a pritom kvalitný WCM systém je zložitá a dlhodobá

[2]XML bol vyvinutý a štandardizovaný konzorciom W3C (World Wide Web Consortium) ako pokračovanie jazyka SGML a HTML. Umožňuje jednoduché vytváranie konkrétnych značkovacích jazykov na rôzne účely a široké spektrum rôznych typov údajov. Tento jazyk je určený predovšetkým na výmenu údajov medzi aplikáciami a na publikovanie dokumentov.

úloha. Preto je treba počítať s vývojom nových verzií a neľahkými opravami chýb. Nasledujúce vlastnosti tvoria základ, ktorý by mal WCM systém ponúkať.

- **Decentralizovaná správa obsahu** musí umožniť viacerým uživatelom aktualizovať web. Plnenie webu potom nebude záležitosťou jedného člověka. Aktualizáciou sa rozumie vytváranie web stránok a položiek v menu.
- **Publikovanie bez znalosti HTML** – aj uživatelia bez zvláštnych technických znalostí by mali byť schopný publikovať. Znalosť HTML by pri práci s WCM systémom nemala byť nutná. Současťou WCM systému je väčšinou aj jednoduchý HTML editor tzv. WYSIWYG (čo vidíte, to dostanete – What You See Is What You Get)[3], ktorý umožňuje jednoduché editovanie textov podobne ako v bežnom textovom editore či jednoduchom systéme formátovania textu. Vytvárať odkazy na ľubovolnom mieste v texte, zvýrazňovať dôležité slová či možnosť jednoducho vkladať obrázky a popisovať ich sú ďalšie nenahraditeľné súčasti každého lepšieho textového editora.
- **Prístupové práva a autentifikácia** rozdelí správu webu medzi viacerých uživatelov. Určitá časť uživatelov môže mať práva len na pridávanie článkov, niektorý môžu meniť celý web. Súčasťou je registrácia a administrácia používatelov, ďalej pridelenie prístupu a práv používatelom prostredníctvom hesla.
- **Prístup cez webový prehliadač** zaručí, že pôjde stránky spravovať odkialkoľvek, kde je pripojenie k internetu. Nemusí sa inštalovať žiadna špeciálna aplikácia. Funkcionalita môže byť občas závislá na type a verzii prehliadača.
- **Proces publikovania** je pri spravodajských serveroch velmi dôležitý. Články by mali prejsť redakčným cyklom, tj. autor napíše článok, redaktor ho skontroluje a upraví a šéfredaktor rozhodne o jeho uverejnení.
- **Možnosť náhľadu** je pomôckou pre uživatelov, aby si mohli overiť, ako bude článok vypadať na webe.
- **Indexovanie a vyhľadávanie** je nevyhnutelná súčasť systému pre správu velkých webov. Kvalitné vyhľadávanie je dôležité predovšetkým pre návštevníkov stránok.
- **Personalizácia** umožňuje individuálne zobrazenie webu pre rôznych návštěvníkov, aby sa podľa požiadaviek a potrieb mohol používatel dostať čo najrýchlejšie k informáciám. Taktiež sa využíva pri komunikácii s návštevníkmi, napr. pre dosiahnutie oslovenia v hromadných e-mailových správach v prípade zmien. Tiež sa počíta s možnosťou viacerých jazykových variánt.

[3]princíp verného prenosu vizuálnej informácie, resp. informácie modelovanej na počítači do reality tak, že zodpovedá presne modelovanému obrazu s čo najmenším, resp. nebadateľným skreslením.

- **Export do iných formátov** umožňuje dostať obsah zo systému a spracovávať ho v iných editoroch. U väčších či moderných webov by mal systém podporovať možnosť publikácie pre systémy WAP a PDA. Samozrejmosťou je export dát do formátu XML.
- **Propojenie s e-commerce systémami** býva obvyklým pri profesionálnych CMS. Systém by mal byť otvorený a umožniť spoluprácu s inými aplikáciami.
- **Uchovávanie verzií obsahu** pre prípad, že by bolo potrebné vrátiť sa k staršej verzii obsahu.
- **Zabezpečenie voči útokom** aby nebolo možné pristupovať do systému iným osobám, ktoré by mohli zmeniť alebo vymazať obsah.

2 APACHE, MYSQL, PHP

Kombinácia sady nástrojov Apache+MySQL+PHP sa spolu často používajú na prevádzkovanie dynamických webových stránok.

2.1 Apache

Apache HTTP Server je softvérový webový server s otvoreným kódom pre väčšinu najpoužívanejších operačných systémov. V súčastnej dobe dodáva prehliadačom na celom svete väčšinu internetových stránok. Ponúka mnohé funkcie, ktoré prevyšujú funkcie komerčných produktov rovnakého typu. Najdôležitejšou vlastnosťou Apache je, že dokáže prevádzkovať súčasne viacero webových lokalít. Apache je vlastne program, ktorý je zodpovedný za vybavovanie požiadavkov HTTP od klientov, tj. najčastejšie webových prehliadačov. Vybavením požiadavku sa rozumie odoslanie požadovanej webovej stránky (obvykle dokumentu HTML) klientovi. Súčasťou odpovede zo strany servera je aj tzv. stavový kód, ktorý charakterizuje, či bol požiadavok vybavený v poriadku, alebo nie. Bežným stavovým kódom je stav OK s číslom 200. Ďalej sú to:

- 3xx - problémy spojené s presmerovaním
- 4xx - chyby súvisiace s vybavením požiadavku (stránka nieje dostupná, apod.)
- 5xx - interné chyby serveru

Obvykle server protokoluje prijímané požiadavky. To pomáhá správcovi webového serveru vytvárať štatistiky a podla typu a množstva požiadavkov optimalizovať obsah, zpôsob uloženia aj zpôsob prezentácie požadovaných dát. Webový server má v zásade dve možnosti, ako získavať informácie, ktoré vracia klientom. Sú to buď predom pripravené datové súbory (HTML stránky), ktoré webový server poskytne bez zmien klientovi, alebo sú najskôr na základe požiadavkov klienta zhromaždené dáta, spracované na servery a formátované pre prezentáciu v HTML kóde. Táto druhá možnosť sa tiež nazýva dynamickým obsahom. K dynamickému vytváraniu obsahu sa používacelá rada rôznych technológií, medzi ktoré patrí aj PHP. Aj keď je server schopný poskytnúť statický obsah výrazne rýchlejšie ako dynamický, na druhej strane je pomocou dynamického obsahu možné poskytovať mnohokrát väčší obsah informácií a aj reagovať na rôzne iné požiadavky klientov. V praxi sa často obidva spôsoby poskytovani obsahu kombinujú, napríklad pomocou cachovania.

Viac informácií je možné získať na domovkej stránke projektu. [1]

[1] http://www.apache.org/

2.2 PHP

PHP je sriptovací jazyk vytvorený práve pre generovanie web stránok. Je umiestenený ako bežiaci proces na strane servera. Do HTML stránky môžeme umiestniť PHP kód, ktorý sa vykoná zakaždým, keď príde požiadavka na zobrazenie tejto stránky. Tento PHP kód sa spracuje serverom, ktorý vygeneruje HTML alebo iný výstup a pošle ho späť žiadatelovi. HTML príkazy sú na strane klienta spracované webovým prehliadačom a uživateľ vidí výslednú formu stránky. PHP bol vytvorený v roku 1994 a to zásluhou človeka s menom Rasmuse Lerdorf. Po uverejnení sa ho chytili ďalší talentovaný ľudia a spoločne vytvorili produkt, dnes už vo verzii PHP 5, ktorý naďalej vyvýjajú. Začiatkom 21. storočia bol tento jazyk používaný na vyše piatich miliónov domén a dodnes tento počet neustále stúpa. Obrovskou výhodou tohto jazyka je fakt, že je Open Source produkt. To znamená, že prístup k jeho zdrojovým kódom je otvorený, môže sa zadarmo používať, upravovať a ďalej distribuovať.

PHP sa už od začiatku mohol používať ako modul webového serveru Apache. Od verzie 4 je dokonca možné ho nainštalovať aj ako ISAPI modul pre Microsoft Internet Information Server. Za ďalšiu výhodu je dobré poznamenať, že je zabudovaná podpora sessions, ktorá má obrovské využitie hlavne pri autentifikácii uživatelov. Zrýchlenie oproti starším verziám spočíva v použití Zend Engine[2]. Ešte väčší výkon sa dosiahne zriadením Zend Optimizer, Zend Cache alebo Zend Compiler.

Viac informácií je možné získať na domovkej stránke projektu. [3]

2.3 MySQL

MySQL je veľmi rýchly a robustný relačný databázový systém. Databáza umožňuje efektívne ukladať, hladať, triediť a získavať dáta. MySQL server má na starosti to, aby sa k databázi mohlo pripojiť viac uživatelov zároveň a zaisťuje, aby to boli iba oprávenený uživatelia. Inými slovami, je to multiuživatelský a multithreadový server. Používa SQL (Structured Query Language)[4], čo je celosvetovo používaný štandardný dotazovací jazyk pre databázy. MySQL databáza je verejnosti prístupná od roku 1996, ale jej korene siahajú až do roku 1979. Šíri sa pod licenciou Open Source, ale v prípade potreby je možné siahnuť aj po komerčnej licencii.

Viac informácií je možné získať na domovkej stránke projektu. [5]

[2] http://www.zend.com/

[3] http://www.php.net/

[4] Structured Query Language (SQL) je počítačový jazyk na manipuláciu (výber, vkladanie, úpravu a mazanie) a definíciu dát. V súčasnosti je to najpoužívanejší jazyk tohto druhu v relačných systémoch riadenia báz dát.

[5] http://www.mysql.com/

2.4 Výhody

Jednou zo skvelých vlastností PHP je, že perfektne funguje s rôznymi operačnými systémami na akomkoľvek plne funkčnom webovom serveri. MySQL databáza je podobne všestranná, a dokonca sú v PHP priamo implementované príkazy a funkcie pre prístup a prácu s databázami. Najčastejšie sa vyskytujúce konfigurácie sú dve nasledovné:

1. operačný systém Linux so serverom Apache + nástroje MySQL a PHP
2. Microsoft Windows (server edícia) s Microsoft Internet Information Server (IIS) + nástroje MySQL a PHP

3 BEZPEČNOSŤ

3.1 HTTPS

HyperText Transfer Protocol Secure/HTTP over SSL (HTTPS) umožňuje prístup k webovým stránkam na serveri s použitím šifrovania. Šifrovanie zabezpečuje vrstva Secure Sockets Layer (SSL). Vyvynula ho firma Netscape v roku 1996 ako nekomerčný otvorený protokol. SSL zaisťuje šifrovanie prenášaných dát a autentizáciu servera pomocou digitálnych certifikátov. Dáta sa zabezpečujú na prechode medzi aplikačnou a transportnou vrstvou (protokolom TCP/IP). SSL môžeme použiť pre bezpečné pripojenie prostredníctvom HTTP ale aj FTP, SMTP, POP3, IMAP4 a ďalších protokolov. Ku skratkám týchto zabezpečených protokolov sa pridáva písmeno „S“ (HTTPS, FTPS, ...). To, že je uskutočnené pripojenie na webové sránky zabezpečené pomocou SSL, sa spozná podľa adresy stránky, ktorá obsahuje taktiež písmeno „s“, napríklad https://localhost alebo podľa upozornenia prehliadača. Výhodou SSL protokolu je aj to, že programátor na využite tohoto zabezpečenia musí zaistiť len presmerovanie na adresu s HTTPS protokolom. Pre použitie SSL je potrebné mať jak na strane servera, tak aj na strane klienta (prehliadač) podporu tohoto protokolu.

3.1.1 Autentizácia pomocou digitálnych certifikátov

SSL certifikát by mal používať každý majiteľ webovej prezentácie, ktorá akýmkoľvek spôsobom zhromažďuje od svojich užívateľov dôverné údaje vo formulároch alebo ponúka napríklad prihlasovanie na stránky pomocou hesiel. Pri intranetových portáloch a hlavne elektronických obchodoch by malo byť používanie SSL zabezpečenia samozrejmosťou. Autentizácia znamená overenie pravosti klienta alebo servera s ktorým komunikujeme. Tento proces používa asymetrické šifrovanie, napríklad algoritmus RSA, tzv. digest a certifikáty.

- RSA (autori Rivest, Shamir a Adleman) je šifra s verejným kľúčom. Algoritmus je vhodný pre podpisovanie aj šifrovanie. Sila zabezpečenia záleží na dĺžke kľúča.
- Digestom sa rozumie výber znakov zo správy nejakou funkciou. Funkcia musí vyberať pre rôzne správy rôzne digesty (aby nevznikol rovnaký digest pre rôzne správy) a nemalo by byť možné zostrojiť k nej inverznú funkciu.
- Certifikát vydáva nezávislá certifikačná autorita a podpisuje ho svojim súkromným klúčom. Pomocou verejného klúča danej autority je potom možné overiť pravosť certifikátu. Certifikát obsahuje meno certifikačnej autority, meno

subjektu, pre ktorý bol vystavený, verejný kľúč subjektu a údaje o časovej platnosti. Podmienkou pre použitie certifikácie je mať samustatnú IP adresu pre doménu. Treba dať pozor pri vystavovaní, pretože medzi certifikátom pre „www.test.org“ a „test.org“ je rozdiel. Ak je správne definovaná cesta, sú šifrované aj podadresáre. Pri šifrovanom spojení sa v prehliadači sa objaví symbol uzavretej zámky, ktorý po kliknutí zobrazí podrobné informácie (názov, dátum vystavenia, platnosť . . .). Certifikát sa vydáva na verejný kľúč, ktorý je generovaný na servere, kde sa bude SSL používať.

Pokial nezáleží na tom, kto pracuje s aplikáciou, len sa potrebujú zabezpečiť informácie s ktorými sa pracuje, nie je potrebné použiť certifikát. Pri naviazaní spojenia so serverom cez https ponúkne prehliadač k použitiu certifikát serveru, ktorý zaisťuje správca serveru. Naopak ak je potrebné zaistiť, aby s aplikáciou komunikoval konkrétny objekt ako napríklad pri operácii s bankovým účtom, poskytne sa klientovy certifikát.

3.1.2 Certifikát pre Apache

Pre použitie certifikátov treba mať v Apachi nainštalovaný modul pre bezpečnostný protokol HTTPS (mod_ssl), nástroj pre podporu kryptografie s verejným klúčom (OpenSSL) a samozrejme vytvorené certifikáty. Použitý programový balík je už nakonfigurovaný pre tieto potreby a nieje potrebné použiť certifikát, preto sa tu nebude uvádzať postup kofigurácie[1], len sa v krátkosti popíše proces generovania kľúča (s popisom žiadosti o certifikát) a následná aplikáciu pre server.

1. Príkaz:

   ```
   openssl.exe genrsa -des3 -out server.key 1024
   ```

   vygeneruje 1024 bitový súkromný klúč servera, ktorý sa uloží do súboru **server.key**. Program požiada o zadanie hesla, ktoré poslúži na ochranu súkromného klúča. Bez hesla by bol certifikát nepoužitelný, preto si ho treba zapametať, alebo použiť klúč bez hesla a to vynechaním parametru -des3.
2. V prípade potreby (nie tento prípad) sa vygeneruje žiadosť o certifikát príkazom:

   ```
   openssl.exe req -new -key server.key -out server.csr
   ```

[1]Všetky potrebné informácie ohľadom inštalácie je možné nájsť na nasledujúcich stránkach: http://www.apache.org, http://www.modssl.org, http://www.openssl.org

Tento príkaz sa spýta na údaje budúceho certifikátu:
Common Name (CN): „názov domény v DNS zázname“
Organization Name: „meno firmy - majiteľa domény“
Organizational unit: „jednotka, účel - napr. e-komerce, internet, . . . “
Country Code: „SK“
State or Province: „Slovak republic“
Locality: „napr. Nové Zámky“
S takouto žiadosťou je potrebné dostaviť sa na niektorú registračnú autoritu, kde túto žiadosť spracujú.

3. V konfiguračnom súbore pre SSL[2] sa nastavia cesty pre súkromný kľúč a certifikát, napríklad:

```
SSLCertificateFile /apache/ssl.crt/server.crt
SSLCertificateKeyFile /apache/ssl.key/server.key
```

4. Reštartovanie Apache web serveru už s podporou SSL. Ak sa pri súkromnom kľúči zadalo heslo, Apache vyzve na jeho zadanie.

3.1.3 Priebeh komunikácie

SSL komunikácia je postavená na synchrónnom šifrovaní pomocou kľúča, ktorý je predaný asynchrónnym šifrovaním (princíp privátneho a verejného kľúča). Obe strany (A a B) teda majú svoj súkromný a verejný kľúč.

A overuje, že komunikuje s B a pošle mu inicializačnú správu. B odpovie na túto správu spolu so svojim certifikátom. A overí platnosť certifikátu. B odošle na A nezašifrovanú správu spolu s digestom, ktorý zašifruje svojím súkromným kľúčom. A rozšifruje pomocou verejného kľúča strany B správu, čím získa digest. Potom funkciou, ktorú použila strana B pre výpočet digestu aplikuje na nezašifrovanú časť správy, čím získa ďalší digest. Tieto dva digesty porovná, a ak sa rovnajú, komunikuje so správnou stranou. Ak by medzi túto komunikáciu vstúpila strana C, mohla by podvrhnúť svôj verejný kľúč a vydávať sa za objekt B. Použitím certifikátu je zaručené, že posielaný verejný kľúč patrí fakticky objektu B. Podobne je v poslednom kroku potrebné overenie. Iba strana. ktorá má správny súkromný kľúč (nielen verejný) úspešne prejde autorizáciou.

[2]záleží od distribúcie
XAMPP: ./apache/conf/extra/httpd-ssl.conf
zvyčajne ale: ./apache/conf/mod_ssl.conf

3.1.4 Naviazanie spojenia, šifrovanie prenášaných dát

Pri inicializácii spojenia a pre zaslanie klúča celej relácie sa v SSL používa opäť algoritmus RSA.

Klient odošle serveru požiadavku Client.Hello a svôj verejný klúč (ten je vygenerovaný pri inštalácii prehliadača). Server po prijatí požiadavku odpovie Server.Hello. Odpoveď zašifruje pomocou verejného klúča prehliadača. V tejto odpovedi sa nachádza aj verejný klúč servera. Po úspešnom prijatí správy Server.Hello odošle prehliadač serveru žiadosť o klúč ktorým bude šifrovaná celá relácia. Táto správa je zašifrovaná verejným klúčom servera. Ako odpoveď zašle server klúč zašifrovaný verejným klúčom prehliadača. Klient prijme klúč relácie a šifruje ostatnú komunikáciu týmto klúčom. V prípade HTTP prenosu sa teda šifrujú všetky HTTP požiadavky, čo má za následok určité zaťaženie. Programátor by mal zvážiť, ktoré požiadavky sa šifrujú a ktoré nie. Dohodnuté šifrovanie ostáva v platnosti pre viac spojení nasledujúcich po sebe. Nové klúče sa generujú pre každý prenos.

3.1.5 Použitie v aplikácii

Ak sa odosiela formulár, stačí ak sa odosielajú zašifrovane len zadané údaje, čiže formulár samotný nemusí byť na zabezpečenej stránke. Treba ale dbať na to, aby v action formulára bolo skutočne presmerovanie na https, preto je niekedy lepšie ponúknuť formulár už na zabezpečenej stránke, ako to pracne ošetrovať. Pomocou PHP je môžné ošetriť aj to, aby sa klienti pripájali len na zabezpečené stránky.

3.2 SQL injection

SQL injection je označenie bezpečnostnej chyby, ktorá umožňuje vsúvať do SQL kódu internetovej aplikácie vlastné kódy. Tieto informácie sa napríklad prostredníctvom formulára dostanú do zdrojového kódu, ktorý spracúva server. Útočník tak môže získať plnú kontrolu nad stránkou obídením hesla, alebo jednoducho môže zmazať všetky informácie, či vykonávať príkazy priamo na servery. Taktiež môže zmeniť vzhlad stránky alebo zbierať citlivé údaje. Chyba spočíva v zlej, alebo dokonca žiadnej kontroly vstypných údajov. Jedná sa o miesta, v ktorých prichádzajú údaje na spracovanie:

- formuláre (POST/GET)
- parametry URI
- HTTP komunikácia (vrátane Cookies a upload súborov)

Najčastejšou chybou sú skryté elementy vo formulároch. Tie možno vyčítať zo zdrojového kódu danej stránky a v prípade potreby zmeniť a odoslať tak rôzne dáta. Účet pod ktorým pristupuje aplikácia do SQL serveru by nemal mať práva systémového administrátora, taktiež aj práva mazať tabulky, či vytvárať nové tabulky (obrovských rozmerov zaťažujúce systémové prostriedky). Chybové hlášky by nemali zobrazovať príliš detailné informácie, alebo dokonca časti kódu.

3.2.1 Princíp

Vytvorenú aplikáciu je možné vyskúšať tým spôsobom, že do vstupných údajov sa vložia rôzne reťazce, ktorými sa overí bezpečnosť. Základný test spočíva v zadaní teťazca:

```
' or 1=1 --.
```

Ak bude výstupom chyba od SQL serveru je velmi pravdepodobné, že niesú ošetrené vstupy. Ak tento reťazec vložíme do políčka pre heslo, potom sa namiesto nasledujúceho príkazu:

```
cSQL = "SELECT uniqueid FROM Users Where UserName='"
& request("userid") & "' and Pwd='" & request("pwd") & "'"
```

vykoná príkaz:

```
SELECT uniqueid FROM Users Where UserName='' or 1=1 --'
and Pwd='heslo'
```

ktorý umožní vstup bez hesla. Odstráni sa nutná podmienka na porovnanie hesla, ktorú nahradí 1=1. Dve mínusové znamienka zabezpečia ignoráciu zvyšnej časti zdrojového kódu. Existujú rôzne kombinácie škodlivých vstupov z ktorých ešte uvediem ďalšie príklady:

```
'' or 1=1 --
or 1=1 --
' or 'a'='a
INFORMATION_SCHEMA.TABLES
INFORMATION_SCHEMA.COLUMNS
'; exec master..xp_cmdshell '...' --
sp_makewebtask, xp_startmail, xp_sendmail
```

Zabezpečenie voči týmto útokom sa mierne líši pre rôzne systémy ale princíp je vždy rovnaký:

- Vo vstupných reťazcoch nahradiť jednoduché apostrofy za zdvojené, čím znemožníme použitie apostrofu pre umelé ukončenie vykonávaného SQL príkazu.
- Prefiltrovať vstupy funkciou, ktorá ponechá len povolené znaky. Týmto spôsobom je vhodné ošetrovať aj vstupy, ktoré sa stanú výstupmi a tým pádom zaistiť aj ochranu pred ďalším typom útoku menom XSS[3].
- Overiť vstupy nie textových reťazcov, či zodpovedajú danému typu premennej (integer).
- Vstupy s pevne nadefinovanými možnými hodnotami overovať, či zodpovedajú presne tomu čo sa očakáva.

Inou možnosťou je využiť parametrizovatelné SQL, ktoré využíva to, že objekty používané pre vyvolanie SQL príkazov umožňujú predávať parametry spôsobmi, ktoré sami o sebe vykonajú potrebné zabezpečenie. Používanie uložených procedúr (stored procedures) umožňuje oddeliť SQL príkazy od skriptov.

[3]Cross-site scripting (XSS) - Útočník vďaka týmto chybám dokáže do stránok podstrčiť svoj vlastný javascriptový kód.

4 VYPRACOVANIE

4.1 Inštalácia servera

Pre zrýchlenie a zjednodušenie inštalácie servera na vlastný počítač sa použije program XAMPP. XAMPP je programový balík neziskovej organizácie Apache Friends. Šíri sa pod licenciou GNU General Public Licence. Je to balík (v čase písania tejto práce) obsahujúci Apache, MySQL, PHP s PEAR, Perl, mod_php, mod_perl, mod_ssl, OpenSSL, phpMyAdmin, Webalizer, Mercury Mail Transport System for Win32 and NetWare Systems v3.32, Ming, JpGraph, FileZilla FTP Server, mcrypt, eAccelerator, SQLite, a WEB-DAV s mod_auth_mysql. Naviac podporuje pridávanie AddOn[1] prídavných častí. Ako vidno, obsahuje rozsiahly Apache server, postačia však server Apache s mod_ssl, MySQL a PHP. Inštalácia je jednoduchá, stačí balík stiahnuť z oficiálnych stránok[2], rozbaliť a spustiť. Podporuje mnohé operačné systémy (Windows, GNU/Linux, Solaris SPARC, Mac OS X), práca sa zameria na Windows. XAMPP podporuje distribúcie Windows 98, NT, 2000, 2003, XP a Vista. Po spustení je môžné jeho funkčnosť overiť na základe priložených ukážkových programov. Funkčnosť sa skontroluje zadaním stránky http://localhost alebo http://127.0.0.1 do webového prehliadača, v ktorom by sa mal objaviť koreňový adresár serveru.

4.1.1 Základná konfigurácia

Pre správnu činnosť je treba odblokovať vo firewalle nasledujúce porty: http 80 (HTTP), https 443 (SSL), mysql 3306. Adresár ktorý predstavuje koreň dokumentov sa nachádza ako podzložka **./htdocs/** v hlavnej zložke nainštalovaného balíku. Štandardne je tam uložený súbor **index.html** (alebo **index.php**) ktorý sa zobrazí po zadaní http://localhost. Hierarchia indexových stránok je v XAMPP volaná nasledovne:

```
DirectoryIndex index.php index.php4 index.php3 index.cgi
index.pl index.html index.htm index.html.var index.phtml
```

Ďalej je dôležité spomenúť konfiguračný súbor pre PHP: **./apache/bin/php.ini** (pre PHP v CGI/CLI móde: ./php/) a pre Apache: **./apache/conf/httpd.conf**. Ak je potrebná zmena v konfigurácii, stačí príslušný súbor otvoriť textovým editorom, previesť zmeny, uložiť a reštartovať server. Konfigurácia MySQL je velmi príjemná cez prostredie phpMyAdmin. Po spustení sa môže stať, že je obsadený

[1] http://addons.xampp.org/
[2] http://www.apachefriends.org/

port číslo 80. Jendou z možností je upraviť súbor httpd.conf tak, aby sa Apache spúšťal pod iným portom. Apache aj MySQL sa dajú zastaviť alebo reštartovať. Do koreňového adresára vložíme nami vytvorené skripty a môžeme overovať ich funkčnosť, tak ako by sa správali na niektorom z internetových serverov, ktoré tieto nástroje obsahujú.

4.1.2 Základné zabezpečenie

Štandardná konfigurácia nieje dobrá z hľadiska zabezpečenia, pretože všetky možnosti sú povolené. Pri konfigurácii PHP sa odporúča zakázať globálne premenné a to tak, že do riadku kde sa nachádza **register_globals** sa za znamienko rovnosti dá hodnota Off. Prehľad zabezpečenia a nastavenie hesiel sa nachádza na adrese http://localhost/security/index.php. Je tam možné nastaviť heslo pre prístup do databázy a pre ochranu adresára (.htaccess). Ochrana adresára je výhodná pri napojení na sieť viacerých počítačov, kde je potrebné zamedziť prístup k webovému serveru. Po zadaní IP adresy do prehliadača sa totiž zobrazia stránky. Webový server Apache (démon httpd) síce beží pod právami administrátora, ale na obsluhu požiadaviek spúšťa niekoľko „potomkov“, ktorý už bežia s právami bežného používateľa.

4.2 Návrh a naprogramovanie

Cielom práce je vytvorenie redakčného systému, v ktorom bude možno vkladať, editovať a mazať články aj krátke správy. Ďalej vytvárať nových uživatelov, nové rubriky a u každého článku diskusné fórum. Keďže sa jedná a relativne rozsiahly projekt, samozrejme sa tým nemyslí to, že bude dokonalý, ale nebudú sa tu uvádzať výpisy zdrojových kódov. Kompletné zdrojové kódy v archivovanom formáte sa nachádzajúna priloženom médiu. Je rovnako možné, že archív bude obsahovať aj ďalšie pridané skipty, ktoré niesú súčasťou tejto dokumentácie pretože by prekračovali jej rozsah.

4.2.1 Štruktúra databázy

Predtým, ako sa začne písať samotný PHP kód, mala by sa premyslieť štruktúra databáze. Pre základné požiadavky postačí celkovo päť tabuliek, ktorých štruktúra bude nasledovná (súbor je pod názvom **install.sql**):

clanky

- id - identifikátor článku
- id_autor - id autora článku

- id_rubrika - id rubriky do ktorej patrí
- datum - unixový čas publikácie článku
- counter - počítadlo prístupov (prečítania) článkov
- priorita - priorita článku - čím vyššia, tým lepšie poradie v zozname - v rámci jedného dňa
- nadpis - názov/nadpis článku
- anotace - stručný obsah článku zobrazujúcí sa v zozname rubrík
- clanek - text celého článku
- poznamka - interné poznámky k článku (odkazy pre redaktorov/korektorov)
- stav - a znamená schválený článok, inak je článok neschválený

novinky

- id - identifikátor novinky
- novinka - text novinky/krátkej správy
- datum - unixový čas vloženia novinky
- stav - a znamená schválená novinka, inak je novinka neschválená

autori

- id - identifikátor autora
- login - login pre prihlásenie do systému
- pass - heslo pre prihlásenie kryptované pomocou funkcie MD5
- jmeno - plné meno autora
- email - email autora
- oautorovi - krátky popis autora
- prava - 1 = autor, 2 = redaktor/korektor, 3 = šéfredaktor
- stav - a = aktivní, ak je v položke iná hodnota, autor se nemôže prihlásiť

rubriky

- id - identifikátor rubriky
- rubrika - názov rubriky

fora

- id - identifikátor diskusného príspevku
- id_clanku - id článku, ku ktorému príspevok patrí
- datum - unixový čas vloženia príspevku

- jmeno - meno prispievajúceho
- email - email prispievajúceho
- predmet - predmet príspevku
- text - text príspevku
- notify - ak je a, budú na email prispievajúceho prichádza emaily s upozornením na nové príspevky

Pre začiatok práce so systémom bude vhodné do tabulky autori vložiť jedného uživateľa s právami šéfredaktora pomocou sql dotazu, ktorý je samozrejme tiež súčasťou archívu pod názvom **add_admin.sql**. Môžu sa samozrejme použiť vlastné údaje, pozor však na heslo, ktoré musí byť v databáze uložené kódovane v MD5.

Teraz neostáva nič iné ako samotné písanie zdrojových kódov. Najskôr sa vytvorí skript **conn.php**, ktorý bude obsahovať funkciu pre pripojenie k databázi a niekoľko premenných. Skript bude includovaný da ďalších skriptov redakčného systému.

4.2.2 Administračné prostredie

Vstup do administračného prostredia bude nasledovný: Prihlásenie bude prebiehať cez formulár priamo na stránke. Druhá možnosť by bola pomocou HTTP autorizácie, ale tá sa vynechá z estetických dôvodov. Stránky administrácie systému budú rozdelené do dvoch hlavných častí, horná bude navigačná a bude obsahovať odkazy na jednotlivé časti administrácie. Druhá časť bude slúžiť pre samotné administračné skripty. Prihlasovací skript sa nachádza priamo v súbore **index.php**, ktorý je mimochodom štandardným skriptom, vykonajúcim sa ako prvý pri zadaní požiadavky na daný koreňový adresár. Najskôr sa vyžiada autorizácia pomocou formulára. Po jeho vyplnení sa skript pokúsi v tabulke **autori** nájsť záznam s odpovedajúcim prihlasovacím menom a heslom. Kedže systém rozlišuje autorov na aktívnych a neaktívnych, bude sa môcť prihlásiť iba ten uživateľ, ktorý bude mať v stĺpečku stav hodnotu **a**. Ak sa takýto záznam nenájde v databázi, zakáže sa prístup a vypíše chybové hlásenie. Ak záznam nájde, zistia sa práva uživatela a do hornej časti okna prehliadača sa načítajú odkazy pre prácu s obsahom.

Skript **function.php** bude obsahovať definované funkcie, aby sa neskôr nemuseli stále pripisovať. Jednotlivé funkcie sa budú postupne popisovať tak, ako budú pribúdať. Nebudú sa vypisovať časti kódu, ktoré zatial nebude treba. Tento skript sa umiestni do koreňového adresára.

4.2.3 Adresárová štruktúra

Dôležitým krokom je tiež správne rozdelenie adresárovej štruktúry. Skriptov a iných súborov (napr. obrázky) bude viac a preto bude vhodné ich troška roztriediť. Ako adresár redakčného systému sa použije priamo koreňový adresár serveru určený pre webové stránky. Umiestnia sa doňho doteraz vytvorené súbory a vytvoria zložky **admin** a **img**. Nastavia sa práva týchto zložiek (chmod 706). Vo všetkých skriptoch v adresári **admin** sa bude konrolovať, či je správne prihlásený uživatel a aké má práva. Šéfredaktor má v systéme najvyššiu váhu a má preto všetky práva. Jeho podriadenými sú redaktori, ktorý majú rovnaké práva ako šéfredaktor, s tým rozdielom, že nemôžu vytvárať nových uživatelov a editovať už schválené články. Poslednými v poradí sú autori, ktorý do systému môžu vkladať články, textové správy a získať štatistiky o čítanosti vlastných článkov.

4.2.4 Menu

Volby v menu sú podľa práv rozdelené takto:

Šéfredaktor

- Editácia uživatelov
- Schválenie článkov
- Prehľad článkov
- plus volby s nižšími právami

Redaktor

- Korektúra článkov
- Krátke správy
- Rubriky
- Štatistiky článkov
- plus volby s nižšími právami

Autor

- Pridať článok
- Pridať novinku
- Moje štatistiky
- plus volby s nižšími právami (tj. aj prezeranie informácií pre návštevníkov)

Týmto by bola navigácia v administračnej časti hotová. Ďalej sa práca bude zaoberať jednotlivými položkami menu.

4.2.5 Uživatelia

Nasledujúce skripty sa budú starať o prehľadné zobrazenie stávajúcich uživatelov, editáciu ich osobných údajov, ich vytváranie a mazanie. Celkom sa o tieto procedúry budú starať tri skripty: **users.php**, **add_user.php** a **edit_user.php**. Umiestnia sa do adresára **admin**.

users.php : Kedže sa skript nachádza v administračnej zložke, najkôr sa overia prihlasovacie údaje uživatela a keď bude všetko v poriadku, zobrazí sa stránka so zoznamom všetkých autorov, v ktorom bude meno autora, stav, email, pozícia a počet článkov vložených v systéme. Ďalej sú pri každom uživatelovi tlačítka EDIT a DELETE. Mazať bude moné iba tých uživatelov, ktorý ešte do systému nezadali žiadny článok. Ak bude treba zmazať takéhoto uživatela, budú mu najskôr zmazané všetky články. Táto stránka obsahuje aj odkaz na vytvorenie nového uživatela.

add_user.php: Stará sa o pridanie nového uživatela. Pri prvom zavolaní sa zobrazí prázdny formulár, do ktorého bude možné vyplniť údaje o uživatelovi. Login, heslo, email a meno sú povinnými položkami. Keď nebudú vyplnené, skript opätovne zobrazí formulár, v ktorom bude treba doplniť chýbajúce údaje. Ak budú všetky poloky vyplnené, skript najskôr overí, či sa v databázi neopakuje zadaný login, a ak áno, zobrazí chybovú hlášku. Ak je všetko v poriadku, skript uloží informácie o uživatelovi do databázy a zobrazí zoznam uživatelov, kde by sa už nový uživatel mal objaviť.

edit_user.php: Obsluhuje formulár pre zmenu údajov o autoroch. Jedná sa vlastne o úpravu predošlého skriptu. Heslo ale nebude povinná položka, a keď sa nevyplní, nebude sa meniť.

Týmto by bola administrácia hotová. Teraz je možné vytvárať uživatelov s rôznymi právami, alebo zakazovať prístup už existujúcich.

4.2.6 Rubriky

Kedže sú už vytvorení uživatelia, ktorý budú môcť vkladať články a krátke správy, bude treba vytvoriť rubriky, do ktorých sa budú články autorov vkladať. Pre administráciu rubrík bude treba vytvoriť ďalšie skripty, ktoré sa budú volať **rubriky.php**, **add_rubrika.php** a **edit_rubrika.php**. Prvý bude zobrazovať zoznam všetkých rubrík a informácie ako počet článkov v rubrike. Taktiež sa bude starať o vymazanie rubriky. Druhý bude slúžiť k pridaniu novej rubriky a tretí k editácii názvov existujúcich rubrík.

rubriky.php: Skript vytvorí stránku so zoznamom rubrík. Pri každej bude zároveň uvedený počet článkov, ktoré rubrika obsahuje. Ďalej budú pri každej rubrike zobrazené tlačítka EDIT a DELETE. Rovnako ako u autorov, nebude možno zmazať rubriku, v ktorej sa už nejaký článok nachádza. Ak ju teda bude treba zmazať, budú

sa musieť najskôr presunúť všetky články do iných rubrík.
add_rubrika.php: zobrazí formulár pre zadanie mena novej rubriky. Po jeho zadaní a odoslaní formulára skript overí, či meno rubriky nieje zhodné s inou rubrikou. Ak bude všetko v poraidku, uloží sa nový záznam do tabulky **rubriky**. Ak nebude zadané meno rubriky, alebo bude zadané meno už existujúcej rubriky, zobrazí sa znova formulár spolu s chybovou hláškou.
edit_rubrika.php: Stará sa o editáciu mena existujúcich rubrík. Rovnako, ako predchádzajúci, sa zobrazí formulár s pôvodným názvom rubriky, kde je možné ho editova. Po odoslaní skript overí, či sa nové meno nezhoduje už s existujúcim a keď nie, prevedie sa zmena. Ak existuje, zobrazí sa formulár znova s chybovou hláškou.

4.2.7 Krátke správy

Krátke správy má možnosť vkladať ktorýkolvek uživatel redakčného systému, ich schvalovanie však závisí na redaktoroch a šéfredaktorovi. O tieto činnosti sa budú starať dva skripty. Jeden bude určený k pridávaniu noviniek a druhý k ich administrácii. Pre jednoduchosť nebude možné krátke správy upravovať. O pridávanie sa bude starať skript **addnovinka.phh** a administráciu **kz.php**. po kliknutí na odkaz Pridať novinku v navigačnom menu sa zobrazí formulár, do ktorého bude mono zadať text správy. Dĺžku správy obmedzíme na päťsto znakov. Po odoslaní sa novinka vloží do databázy, kde bude čakať na schválenie. Akonáhle bude niektorá z pridaných správ schválená, na stránkach sa bude objavovať spolu s niekoľkými najnovšími správami, v našom prípade piatimi.

addnovinka.phh : Tento skript je dostupný všetkým členom systému. Skript zobrazí formulár pre vloženie krátkej správy. Po tom, čo bude správa vložená a odošle sa formulár, skript vloží do databázy text správy. Tá bude čakať na schválenie.

kz.php : Tento skript zobrazí zoznam všetkých vložených, ale neschválených krátkych správ. Pri každej položke bude plné znenie správy spolu s tlačítkami EDIT a DELETE. Pre jednoduchosť je odobraná možnosť editácie správy. V prípade potreby sa však dá ľahko dopísať.

Ako nasledujúce sa ukáže, ako zariadiť správu článkov: pridávanie, korektúry, mazanie a schvalovanie.

4.2.8 Články

Pre funkčnosť bude potrebných šesť skriptov. Prvý sa volá **addclanek.php** a slúži k vloženiu článku do redakčného systému. Ďalší sa volá **prehled.php**. tento skript zobrazí prehľad všetkých článkov s odkazmi EDIT a DELETE. Zobrazované informácie budú nasledovné: autor článku, rubrika, v ktorej sa nachádza, dátum

vloženia (ak bol článok schválený, jedná sa o dátum publikácuie), priorita článku, čitatelnosť a stav článku (vložený, zdokumentovaný, schválený). Tretí skript bude **korektura.php**. Zobrazí rovnaký prehlad článkov, ako predchádzajúci, ale obsahuje naviac aj možnosť úpravy samotného článku. Štvrtým a piatym skriptom sú **stats.php** a **userstats.php**. Prvý zobrazí štatistiky čítanosti článku. Druhý je mierna modifikácia predchádzajúceho, s tým rozdielom, že sa jedná len o štatistiku článku príslušného autora. Posledným skriptom je **nahled.php**, čo je jednoduchý skript ktorý zobrazí článok v podobe HTML stránky.

addclanek.php : Tento skript bude vkladať do databázy nové články. Článok sa vždy vloží pod autorom, ktorý je práve prihlásený. Ak napríklad šéfredaktor vkladá článok za iného autora, ktorý nemá prístup k redakčnému systému, bude musieť autora zmeniť pri korektúre článku. Skript zobrazí formulár pre vloženie všetkých potrebných položiek (rubrika, nadpis, anotácia, text článku). Ďalej formulár obsahuje checkbox, ktorý keď sa zatrhne, spôsobí v článku zmenu všetkých prechodov na nový riadok na tag **br**. Po odoslaní formulára skript uloží získané dáta do databázy.

korektura.php : Aj keď by mal nasledovať skript **prehled.php**, preskočí sa, keďže je celý obsiahnutý v skripte **korektura.php**. Tento skript zobrazí obsah článkov v databáze a pokiaľ bude pri jeho zavolaní známa premenná **id**, zobrazí sa miesto zoznamu článkov formulár určený k editácii článku. Tento formulár je podobný formuláru pre vkladanie článkov, ale neobsahuje položku BEZ HTML a má naviac pole priorita, dátum a stav. Priorita určí dôležitosť článku v danom dni - v zozname článkov sa bude zobrazovať v rámci článkov z jedného dňa vyššie ako články bez alebo s nižšou prioritou. Dátumom sa určí dátum publikácie článku - môže sa tak jeho vydanie naplánovať kľudne aj mesiac dopredu. Stav určuje, v akom štádiu sa konkrétny článok nachádza. Či bol práve vložený, či už prešiel korekturou alebo či je už článok schválený. Návštevníkom stránok sa samozrejme budú zobrazovať len schválené články. Vo formulári je teda možnosť editovať všetky potrebné údaje. Po jeho odoslaní bude článok v databáze upravený.

stats.php a **userstats.php** : Zobrazujú štatistiky článkov podľa ich čítanosti. Články budú chronologicky usporiadané od najčítanejších po najmenej čítané. Oba skripty sú skoro totožné, avšak skript **stats.php** zobrazuje štatistiky článkov všetkých autorov a je určený pre redaktorov a šéfredaktora. Druhý skript je obmedzený len na štatistiky článkov príslušného autora.

nahled.php : Po zavolaní s parametrom **id** zobrazí požadovaný článok. Ten bude zobrazený ako náhľad na HTML stránku a otvorí sa v novom okne.

Týmto sa dokončilo programovanie administrátorskej časti redakčného systému. Ďalej ostáva prebrať klientskú časť.

4.2.9 Klientská časť

Zaoberá sa zobrazením zoznamov článkov s rozdelením do jednotlivých rubrík, zobrazením článkov podla autorov a nakoniec zobrazením samotného článku. V tejto časti sa doplní skript **index.php**, ktorý sa bude starať o zobrazenie zoznamu článkov a to buď všetkých, alebo len článkov v konkrétnej rubrike či konkrétneho autora. Skript sa ďalej bude starať o zobrazenie samotného článku. Najskôr sa popíše zásah do skriptu **function.php**, ktorý sa rozšíri o novú funkciu. Ďalej nový skript **levy.php**, ktorý bude vkladaný do hlavného skriptu a do skriptu **forum.php**, ktorý sa rozoberie v poslednej časti tejto práce.

function.php : Pridá sa funkciu **odkazy.php**, ktorá bude slúžiť k rozdeleniu zoznamu článkov do viacerých stránok s predom určeným počtom článkov na stránke. Tieto stránky budú medzi sebou vzájomne poprepájané odkazmi. Funkcia vráti časť HTML kódu s odkazmi na ďalšie stránky zoznamu.

levy.php : Bude obsahovať pravý stĺpec stránky, ktorý bude využívať ako hlavná stránka, tak aj stránky s diskusiami k článkom. Jeho štruktúru som navrhol tak, aby tu boli všetky dôležité informácie - rubriky, zoznam aktuálnych a najčastejších článkov, krátke správy a zoznam autorov k rýchlej volbe článkov jednotlivých autorov. Samozrejme sa usporiadanie týchto informácií môže zmeniť podľa vlastných potrieb.

index.php : Nasleduje najdôležitejší skript, ktorý sa stará o zobrazenie zoznamu článkov, usporiadaných podla autorov, či robrík, ďalej sa potom stará o zobrazenie samotného článku. Začne zoznamom rubrík, ktorých je celkom tri. Jeden je zoznam všetkých článkov a bude sa zobrazovať na titulnej stránke. Ďalej sa jedná o zoznam článkov rozdelených do jednotlivých rubrík a článkov rozdelených podľa autorov. O vytvorenie takéhoto zoznamu sa bude starať vždy jeden SQL požiadavok. Tie sa nachádzajú priamo v tomto skripte. Každá SQL požiadavka predchádza kratší SQL požiadavok, ktorý spočíta počet všetkých článkov, počet článkov autora, alebo počet článkov v rubrike. Do súčtu sú zaraďované len články, ktoré sú schválené a ich čas publikácie je rovnaký, alebo niší ako aktuálny. Výsledok tejto požiadavky bude slúžiť k rozpočítaniu hlavného výsledku do viacero stránok tak, aby sa nezobrazoval zoznam všetkých článkov naraz na jednej stránke. Ďalším výberom je výber samotného článku, ktorý sa už popísal. Pribudne k nemu však podmienka, ktorá bude zobrazovať len aktívne a publikované články.

Stále vyberanie z databázy nieje najlepšie riešenie, avšak je najjednoduchšie k pochopeniu systému. Je jasné, že vyberať najčítanejšie, najnovšie články, rovnako ako zoznamy rubrík je celkom neefektívne a v prípade velkej návštevnosti serveru by mohlo dochádzať k preťaženiu. Ak teda bude záťaž veľká, treba priprogramovať skripty, ktoré budú generovať súbory obsahujúce práve tieto informácie.

4.2.10 Diskusné fóra

Posledná časť sa zaoberá diskusnými fórami, pridávaním a zobrazovaním jednotlivých diskusných príspevkov. Všetky príspevky sa budú ukladať do databázy, odkial budú následovne čítané a zobrazované návštevníkom. Túto robotu nám bude robiť skript **forum.php**.

forum.php : Tento skript obdrží ako argument premennú **clanek**, v ktorej bude **id** článku. Ak táto premenná nebude existovať, skript zobrazí hlášku o tom, že článok neexistuje. Ked bude argument uvedený, vyhľadá sa v tabulke **clanky** názov článku (nadpis), ktorý sa použije vo výstupe skriptu. Keď názov článku nenájdeme, opätovne sa zobrazí hlášku o tom, že článok neexistuje. V opačnom prípade sa skript bude snažiť v tabulke **fora** nájsť všetky záznamy, kde bude v stĺpci **id_clanku** rovnaké číslo s **id** článku. Takéto záznamy sa zoradia podla dát zostupne a zobrazia. Nakoniec sa vloží formulár pre pridanie komentáru. Keď sa v tabulke nenájde žiadny príspevok patriaci k článku, ku ktorému sa fórum vzťahuje, zobrazí sa o tom jednoduchá hláška. Akonáhle návštevník vyplní formulár a odošle ho, skript sa bude snažiť vyhodnotiť dáta vyplnené vo formulári. Ak návštevník nezadá meno a text príspevku, bude mu vrátený formulár s chybovým hlásením a žiadosťou o vyplnení povinných položiek. Ak budú všetky povinné polia vyplnené, uložia sa dáta do databázy. Potom sa budú snažiť z databázy načítať všetky emailové adresy prispievatelov, ktorý zatrhli položku na upozorňovanie nových príspevkov.

4.3 Zabezpečenie

4.3.1 Prihlasovanie

Spôsob prihlasovania sa popíše na zjednodušenej aplikácii. Tá obsahuje iba jednoduché prostredie a nieje chránená proti SQL injection. Stránka obsahuje prihlasovací formulár, menu a obsah stránky. Formulár obsahuje dve základné premenné, ktorými sú prihlasovacie meno a heslo. Po ich zadaní a potvrdení sa údaje pošlú na zabezpečenú stránku (v našom prípade na tú istú) kde sa spracujú. Najskôr sa inicializuje posedenie session. Použitie session má výhodu v tom, že toto posedenie je viazané na okno prehliadača, takže ak by niekto zadal rovnakú stránku s rovnakými dátami, neuspel by. Skontroluje sa, či sa nejedná o klienta, ktorý už má posedenie. Nasleduje pripojenie k databázi a potom v prípade žiadosti odhlásenia prebehne odstránenie posedenia a presmerovanie na domovskú stránku. Ak prišla žiadosť o prilásenie, čo je tento prípad, aplikácia porovná zadané údaje s údajmi v databázi. Ak boli zle zadané údaje prihlásenie je neúspešné a presmeruje sa na domovskú stránku. V prípade úspechu je uživatel prihlásený, nastaví sa mu posedenie a premenné ktoré

obsahujú informácie a hodnoty potrebné pre ďalší beh aplikácie. Skript pokračuje v načítaní hlavičiek pre zabránenie cachovania a načíta súbor s funkciami. Nasleduje HTML kód v ktorom sa nastavia ďalšie hlavičky. Kedže už prebehlo prihlásenie, formulár sa nezobrazí, vidia sa však ďalšie položky v menu v závislosti od toho, aké sú pridelené práva. Ako posledné sa načíta zvolený obsah stránky zo zadaného súboru a uzavrie sa spojenie s databázou. Súbor s obsahom však pre korektné zobrazenie musí obsahovať nasledujúci kód:
súbor s obsahom, napríklad **main.php**

```
<?if(isset($check_page) && $check_page=="gogo"):?>
osah stranky main
<?
else:
echo '<meta http-equiv="refresh" content="0; url=./">';
echo ' <strong>LOADING...</strong>';
die;
endif;
?>
```

Zdrojový kód hlavného súboru **index.php** sa nachádza v druhej prílohe.

4.3.2 Použitie SSL

Sú tri spôsoby použitia a to buď použitím direktívy SSLRequireSSL, alebo priamo v aplikácii presmerovaním na zabezpečenú stránku, alebo vložením do zdrojového kódu súboru, na ktorý je potrebné pristupovať len v zabezpečenom móde nasledujúci kód:

```
<?php
$HTTPS_PORT=443;
// port SSL komunikacie
// administrator serveru moze nastavit iny

// overenie,ci sa pouziva HTTPS
if ($_SERVER["SERVER_PORT"] != HTTPS_PORT)
{
  // ak nie, pokusime sa nanho presmerovat
  if (empty $_GET["https"])
  {
    // ak je prazdna testovacia premenna
```

```
    // presmerujeme na zabezpecenu stranku
    // plus nastavime tuto premennu
    $retazec = "";
    $retazec .= $_SERVER["HTTP_HOST"];
    $retazec .= $_SERVER["SCRIPT_NAME"];
    $retazec .= "?";
    $retazec .= $_SERVER["QUERY_STRING"];
    $retazec .= "&https=1";
    header("Location: https://".$retazec);
  }
  else
  {
    echo "Nemožno uskutočniť zabezpečený prenos";
    exit;
  }
}
?>
<p>Zabezpečené</p>
```

Pristupovať na zabezpečený obsah treba obmedziť na miesta, kde je to fakt potrebné, pretože výmena overovacích informácií môže značne spomaliť našu aplikáciu. Určite by ale nemal chýbať pri prihlasovaní, registrácii a manipulácii s citlivými údajmi. Otázne je použitie formulárov. Keďže stačí iba odoslať údaje pomocou SSL a samotný formulár nemusí byť na zabezpečenej stránke, buď sa pracne ošetrí odoslanie každého formulára na zabezpečený obsah alebo vložením tohoto kódu sa načíta formulár na zabezpečenej stránke a presmerovanie nebude treba ošetrovať.

4.3.3 Ochrana pred SQL injection

Pre jednoduchú ochranu pred SQL injection je môžné použiť jednu z nasledujúcich funkcií (použitie záleží od verzie PHP).

Verzia PHP 4.3 a viac:

```
<?
/*
Funkcia: sql_fixacia( $xCode )
Popis: "Fixacia" stringu SQL kodu k prevencii pred SQL injection.
Parametre: $xCode : SQL kod ktory potrebujete fixovat.
Ukazka: mysql_query('UPDATE table SET value="' .
        sql_fixacia('" SET id='4'") . '" WHERE id="1"');
```

```
Poziadavky: PHP verzia 4.3 a viac
*/

function sql_fixacia( $xCode )
{
  if ( function_exists( "mysql_real_escape_string" ) )
  { // Ak je PHP verzia > 4.3.0
    $xCode = mysql_real_escape_string( $xCode );
    // Escapuje MySQL string.
  }
  else
  { // Ak je PHP verzia < 4.3.0
    $xCode = addslashes( $xCode );
    // Pred citlive charaktere vlozi spatne lomitka \
  }
  return $xCode; // Vrati fixovany kod
}
?>
```

Verzia PHP 5 a viac:

```
<?
/*
PHP 5+ ONLY - Prevencia pred SQL injection a XSS utokom
1 *REQUIRED* value, 1 <OPTIONAL> value to call this function:
$input = Vstupny input
$is_sql = Kontrola ci nahodou $input nie je sql query
Ukazka pouzitia:
$username = sterilize($_POST['username']);
$query = "SELECT * FROM users WHERE username = '$username'";
*/

function sterilize ($input, $is_sql = false)
{
  $input = htmlentities($input, ENT_QUOTES);
  if(get_magic_quotes_gpc ())
  {
    $input = stripslashes ($input);
  }
  if ($is_sql)
```

```
  {
    $input = mysql_real_escape_string ($input);
  }
  $input = strip_tags($input);
  $input = str_replace("
", "\n", $input);

  return $input;
}
?>
```

Druhá funkcia overuje, či nieje vložený do vstupného údaja SQL dotaz a zisťuje nastavenie magic quotes. Magic quotes je funkcia v PHP, ktorá pomáha zabrániť neskúseným vývojárom pred písaním zranitelného kódu a ochrániť tak pred SQL injection útokom. Funkcia sa teda vloží do súboru **funkcie.php** a upravia súbory v ktorých sa spracovávajú formuláre tak, aby sa dáta spracovali pred vložením do databáze.

Inou možnosťou je napríklad nastaviť magic quotes a pridať ku každému formulárovému polu funkciu addslashes (addshlashes($_POST['heslo'])). Funkcia addslashes všetok text odoslaný na server ohraničí apostrofmi, čím sa prestane interpretovať ako kód a bude sa vnímať ako čistý text. Je však nutná kontrola textu medzi apostrofmi (alebo inými znakmi), pretože útočník môže do textu vložiť apostrof a tým obísť túto ochranu. Pri následnom použití magic quotes by mal tento problém odstrániť.

ZÁVER

Prešli sa základné princípy tvorby CMS pomocou silných nástrojov PHP a MySQL. Veľký počet vývojárov webu tieto nástroje používajú nielen k tvorbe CMS, ale všeobecne dynamických webových stránok a rôznych iných projektov.

Oblasť systémov pre správu obsahu sa neustále vyvýja a preto sa nedá presne predpovedať, ako budú systémy vyzerať a aké budú ich ďalšie požiadavky. V súčastnej dobe sa tieto systémy sústredia hlavne na prácu s obsahom web stránky. Publikácia je potom záležitosťou samotnej aplikácie. Za pozornosť treba brať používanie formátu XML (rozšíriteľný značkovací jazyk – eXtensible Markup Language)[3], z ktorého je možné ďalej generovať ďalšie formáty (HTML, PDF, PostScript, atd.).

Nesmie sa zabúdať ani na ochranu takýchto systémov a používať len komponenty ktoré sú naozaj potrebné.

[3]XML bol vyvinutý a štandardizovaný konzorciom W3C (World Wide Web Consortium) ako pokračovanie jazyka SGML a HTML. Umožňuje jednoduché vytváranie konkrétnych značkovacích jazykov na rôzne účely a široké spektrum rôznych typov údajov.

LITERATÚRA

[1] KOCMAN, Jiří. *Redakční systém v PHP* [online]. 2001, poslední aktualizace 3. 4. 2003 [cit. 27. 11. 2007]. Dostupné z URL: <http://www.builder.cz/art/php/php_rs_1.html>.

[2] RŮŽIČKA, Pavel. *Bezpečnost především - použití SSL* [online]. 2002, poslední aktualizace 6. 6. 2002 [cit. 28. 05. 2008]. Dostupné z URL: <http://interval.cz/clanky/bezpecnost-predevsim-pouziti-ssl/>.

[3] ULLMAN, Larry. *PHP a MySQL - Názorný průvodce tvorbou dynamických WWW stránek.* 1. vydání, Brno : Computer Press, 2004. 534 s. ISBN 80-251-0063-4.

[4] WELLING, Luke, THOMSON, Laura. *PHP a MySQL - rozvoj webových aplikací.* Praha : SoftPress, 2002. 718 s. ISBN 80-86497-20-8.

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

CMS Systém pre správu obsahu – Content Management System

PHP skriptovací programovací jazyk – Hypertext PreProcessor

MySQL viacvláknový, viacužívateľský SQL relačný databázový server

WYSIWYG čo vidíte, to dostanete – What You See Is What You Get

SEO optimalizácia pre vyhľadávače – Search Engine Optimization

HTML značkovací jazyk pre hypertext – HyperText Markup Language

XML rozšíriteľný značkovací jazyk – eXtensible Markup Language

SQL Structured Query Language

IIS Microsoft Internet Information Server

ODBC Open Database Connectivity Standard

FAQ často kladené otázky – Frequently Asked Questions

XML rozšíriteľný značkovací jazyk – eXtensible Markup Language

HTTPS HyperText Transfer Protocol Secure/HTTP over SSL

SSL Secure Sockets Layer

XSS Cross-site scripting

ZOZNAM PRÍLOH

A PRVÁ PRÍLOHA - WEBOVÉ ODKAZY

V tejto prílohe sú uvedené niektoré z mnoho zdrojov dostupných online, v ktorých sa nachádzajú tutoriály, články, novinky a ukážkové kódy.

PHP.Net
http://www.php.net/
oficiálne stránky PHP, manuály, zdroje

ZEND.Com
http://www.zend.com/
engine ZEND, fóra, ukážky, kódy

PHPWizard.Net
http://www.phpwizard.net/
atraktívne aplikácie (napríklad phpMyAdmin), tutoriály

PHPBuilder.com
http://www.phpbuilder.net/
tutoriály, fórum, message board otázok a odpovedí

DevShed.com
http://www.devshed.com/
tutoriály viacerých vývojárskych jazykov (PHP, MySQL, Perl, atd.)

PX-PHP Cpde Exchange
http://www.px.sklar.com/
pre začínajúcich, ukážkové skripty, užitočné funkcie

WeberDev.com
http://www.weberdev.com/
tutoriály, ukázkové kódy, bezpečnosť, nutná registrácia

HotScripts.com
http://www.hotscripts.com/
výborná stránka, skripty v kategóriách, skripty v rôznych jazykoch

PHP Base Library
http://phplib.netuse.de/
velké projekty, pomôcky pre databázy a šablony, tutoriály

WebMonkey.com
http://www.webmonkey.com/
tutoriály pre pravé aplikácie, kódy, design, uživatelské rozhranie, multimédiá atd.

The PHP Club
http://www.phpclub.net/
pre začiatočníkov, novinky, recenzie kníh, kódy, fóra, FAQ, návody

The PHP Resource Index
http://www.phpresourceindex.com/

skripty, triedy, dokumentácie, prehladne radené do kategórií

PHP Developer

http://www.phpdeveloper.org/

skripty

Evil Walrus

http://www.evilvalrus.com/

super vyzerajúci portál skriptov

Oodie.com

http://www.oodie.com/

zoznam PHP hostingu zdarma, skripty

Source Forge

http://sourceforge.net/

obsiahly zdroj Open Source materiálov, vyhľadávanie kódu, CVS, mailing listy

The MySQL

http://www.mysql.com/

oficiálne stránky MySQL, vinikajúca dokumentácia, podpora, zdroje

SQL Tutorial

http://w3.one.net/~jhoffman/sqltut.html

kompletný tutoriál s cvičeniami a príkladmi

The SQL Course, http://www.sqlcourse2.com/

základný tutoriál, inštrukcie, online skúšanie

DatabaseCentral.Com

http://databasecentral.com/

informácie, tutoriály, tipy, FAQ, recenzie

The SQL Pro

http://www.inquiry.com/techtips/thesqlpro

možnosť kladenia otázok profesionálom

Apache Software

http://www.apache.org

zdroje a binárne súbory, online dokumentácia

Apache Week

http://www.apacheweek.com

online týždenník dôležitých informácií

Apache Today

http://www.apachetoday.com

denný zdroj noviniek a informácií, pre otázky nutná registrácia

B DRUHÁ PRÍLOHA - INDEX.PHP

hlavný súbor **index.php**

```
<?
// ************** IDENTIFICATION

 session_start();
 if (!Isset($_SESSION["genom"]))
 {
   $_SESSION["genom"] = "";
 }
 // ak nieje nastavene, je to zatial host
 $login_name = $_SESSION["genom"];
 // uzivatel je teraz host

// ************** DB CONNECTION

@$conn = mysql_connect("localhost","root","secret")
   or die ("Nelze navazat spojeni s databazi");
@$sdb = mysql_select_db("rsdb")
   or die ("Nenalezena databaze");

$ADMIN_MAIL = "szabko@gmail.com";
$pozice = array(0=>"Host","Autor","Redaktor","zaloha","root");

// ************** FCE

if(Isset($_GET['fce']) and $_GET['fce']=="logout")
{
  session_unset("genom");
  $login_name = "";
  echo '<meta http-equiv="refresh" content="0; url=http://'.
  $_SERVER['SERVER_NAME'].$_SERVER['SCRIPT_NAME'].
  '?go=main&message=1"> <strong>LOADING...</strong>';
  die;
}

// ************** LOGIN
```

```
if(Isset($_POST['login']))
{
  $skuska_meno = $_POST['meno'];
  $skuska_heslo = md5($_POST['heslo']);

  $sql = mysql_query("SELECT * FROM autori
  WHERE login='$skuska_meno'
  AND pass='$skuska_heslo'
  AND stav='a' ");

  $res = mysql_fetch_array($sql);
  // uvolnime pamet
  mysql_free_result($sql);

  if ($res['login']=="")
  {
    session_unset("genom");
    $login_name = "";

    $cas = StrFTime("%d/%m/%Y %H:%M:%S", Time());
    $ip = $_SERVER['REMOTE_ADDR'];
    $host = gethostbyaddr($_SERVER['REMOTE_ADDR']);

    echo '<meta http-equiv="refresh" content="0; url=http://'.
    $_SERVER['SERVER_NAME'].$_SERVER['SCRIPT_NAME'].
    '?go=main&message=2"> <strong>LOADING...</strong>';
    die;
  }
  else
  {
    $_SESSION["genom"] = $res['login'];
    echo '<meta http-equiv="refresh" content="0; url=http://'.
    $_SERVER['SERVER_NAME'].$_SERVER['SCRIPT_NAME'].
    '?go=main&message=3"> <strong>LOADING...</strong>';
    die;
  }
}
```

```
// ************** USER RIGHTS & INFO

if ($login_name=="")
{
  $stack_pravo = 0;
}
else
{
  $sqlidus = mysql_query("SELECT * FROM autori
  WHERE login='$login_name'");

  $idus = mysql_fetch_array($sqlidus);
  mysql_free_result($sqlidus);
  $stack_pravo = $idus['prava'];
  $stack_login = $idus['login'];
  $stack_jmeno = $idus['jmeno'];
  $stack_email = $idus['email'];
  $stack_popis = $idus['oautorovi'];
  $stack_id = $idus['id'];
}

// ************** HEADERS

if ($stack_pravo>0)
{
  Header("Pragma: No-cache");
  Header("Cache-Control: no-cache");
  Header("Expires: ".GMDate("D, d M Y H:i:s")."GMT");
}

// ************** BODY


if(File_Exists('./funkcie.php'))
{include './funkcie.php';}
else
{echo "chyba subor s funkciami";die;}
if(!isSet($_GET['go']))$_GET['go']="main";
?>
```

```
<html>
<head>
  <title>Projekt - RS</title>
    <meta http-equiv="content-type" content="text/html;
    charset=utf-8" />
    <meta http-equiv="content-language" content="sk" />
    <meta http-equiv="Pragma" content="no-cache" />
    <meta http-equiv="Cache-control" content="no-cache" />
    <meta name="robots" content="index, follow" />
    <meta name="author" content="Szabolcs Garai" />
    <meta name="copyright" content="" />
    <meta name="category" content="document" />
    <meta name="description" content="Redakcni system" />
    <meta name="keywords" content="max, 20, klucovych, slov" />
    <meta name="generator" content="PSPad" />
      <link rel="Shortcut Icon" href="favicon.ico" />
      <link rel="stylesheet" type="text/css"
      href="style.css" title="default">
</head>
<body>

PRIHLASOVACI FORMULAR
<?if(!($stack_pravo>0)):?>
<form method="post" action="
<?
echo "https://".
$_SERVER['SERVER_NAME'].
$_SERVER['SCRIPT_NAME'];
?>
">
Login <input class="a100" type="text" name="meno" />
Heslo <input class="a100" type="password" name="heslo" />
<input class="ok" type="submit" value="" name="login" />
</form>
<?endif;?>

PRIKLAD MENU
toto vidi kazdy navstevnik
```

```
<?if($stack_pravo>0):?>
<ul>
<?if($stack_pravo>=1):?>
<li>toto vidi autor</li>
<?endif;?>
<?if($stack_pravo>=2):?>
<li>toto vidi redaktor</li>
<?endif;?>
<?if($stack_pravo>=4):?>
<li>toto vidi sefredaktor</li>
</ul>
<?endif;?>

OBSAH STRANKY

<?php
$check_page = "gogo";
  if($_GET['go']<>"admin"
   and File_Exists('./'.$_GET['go'].'.php'))
    {
    include './'.$_GET['go'].'.php';
    }
  elseif($_GET['go']=="admin" and isset($_GET['to'])
  and File_Exists('./admin/'.$_GET['to'].'.php'))
    {
    include './admin/'.$_GET['to'].'.php';
    }
  else
    {
    echo "zadana stranka neexistuje";
    }
?>

</body>
</html>

<?Mysql_close($conn);?>
```