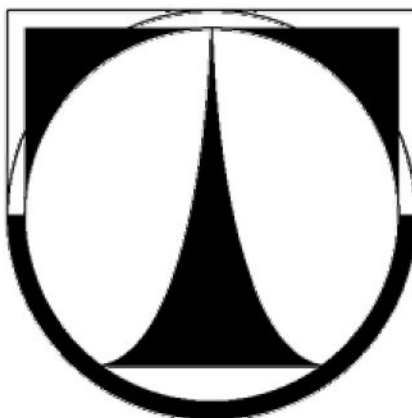


TECHNICKÁ UNIVERZITA V LIBERCI

FAKULTA STROJNÍ

Katedra aplikované kybernetiky



Studijní program
B2341 - Strojní inženýrství

Zaměření
Inženýrská informatika

**INFORMAČNÍ SYSTÉM PRO POŘÁDÁNÍ KONFERENCÍ NA
TECHNICKÉ UNIVERZITĚ V LIBERCI**

Autor:	Adam Kryštof
Vedoucí bakalářské práce:	Ing. Michal Moučka, Ph.D.
Konzultant bakalářské práce:	Ing. Jan Blažek

Liberec 2010

ANOTACE

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta strojní

Katedra aplikované kybernetiky

Studijní program:	B2341 - Strojní inženýrství
Zaměření:	Inženýrská informatika
Autor:	Adam Kryštof
Téma práce:	Informační systém pro pořádání konferencí na Technické univerzitě v Liberci
Theme of work:	Information system for organizing seminars and conferences on Technical university of Liberec
Rok obhajoby BP:	2010
Vedoucí BP:	Ing. Michal Moučka, Ph.D.

Anotace:

Úkolem mé bakalářské práce bylo vytvořit dynamické webové stránky pro pořádání konferencí a seminářů na Technické univerzitě v Liberci organizované Fakultou strojní.

Tyto stránky mají sloužit jak pro zobrazování informací o akcích pro návštěvníky, tak také pro registraci na samotné akce, zobrazení účtu účastníka a jeho úpravy.

Dále mají být určeny administrátorům pro zakládání samotných akcí a jejich správu.

Abych mohl tyto stránky vytvořit, bylo potřeba zvolit vhodný programovací jazyk, a protože samotný HTML by kýženého výsledku nedosáhl, jsou stránky programované v PHP a jako u ostatních moderních webových aplikací jsou použity prvky CSS.

Co tyto zkratky znamenají a základní informace o těchto jazycích je možné se dozvědět v první části této bakalářské práce, samotný popis a funkce webových stránek v části druhé.

Annotation

The task of my thesis was to create dynamic web pages for conferences and seminars on Technical university of Liberec. These conferences and seminars are organized by the Faculty of Mechanical Engineering. These pages served for display the information about events for visitors, and for registration on the event, and view the participant's account. These sites are also designed for administrators for the creation and editing events. These pages are programmed by HTML, PHP and CSS languages. In the first part of this thesis you can learn what do these abbreviations mean, and basic information about these languages, features of website for conferences and seminars are explained in part two.

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 - školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

Poděkování

Na tomto místě bych rád poděkoval Ing. Janu Blažkovi za odbornou pomoc a Ing. Michalu Moučkovi, Ph.D. za cenné připomínky a vedení, čímž přispěli k vypracování této bakalářské práce.

Dále děkuji svým známým a rodině za všestrannou podporu poskytovanou při psaní této bakalářské práce.

Obsah

1 Předmluva.....	7
1.1 Cíl práce.....	8
1.2 Motivace.....	8
2 Webové stránky.....	9
a) Statické webové stránky.....	9
b) Dynamické webové stránky.....	9
2.1 HTML.....	9
2.1.1 Koncepce	10
2.1.2 Struktura dokumentu.....	11
2.2 Zobrazování v prohlížečích.....	12
2.3 Editory HTML.....	14
a) Textové editory.....	14
b) WYSIWYG editory.....	14
3 CSS Kaskádové styly.....	16
3.1 Selektory.....	17
4 PHP.....	19
4.1 Data vydání hlavních verzí.....	20
4.2 Výhody a nevýhody PHP.....	21
4.3 Příklad.....	22
4.4 Požadavky PHP.....	23
a) internetový server.....	23
b) Nejjednodušší domácí instalace.....	23
c) Alternativní instalace.....	24
d) Práce se serverem Apache.....	24
e) Kořenová složka.....	25
f) Konfigurace Apache.....	25
g) Vyhodnocování PHP.....	25
h) konfigurace PHP.....	25
i) nastavení SMTP.....	25
j) Práce s databází MySQL.....	26
k) PHP MyAdmin.....	26
5 Tvorba webu.....	27
5.1 Databáze.....	27
5.2 Soubory s CSS.....	30
5.3 Frontend – index.php.....	31
5.3.1 HTML část.....	32
5.3.2 PHP část.....	33
a) Proměnné typu \$_GET.....	33
b) Proměnné typu \$_POST.....	33
c) Proměnné typu \$_SESSION.....	33
d) funkce echo a printf.....	34
5.3.3 Výpis z databáze.....	35

5.4 Backend – administrace.php.....	36
5.4.1 HTML část.....	36
5.4.2 PHP část.....	37
a) Vkládání příspěvků.....	38
b) Úprava příspěvků.....	39
c) Smazání příspěvků.....	40
d) Výpis článků.....	40
6 Stránky konference.....	42
6.1 Frontend.....	42
6.2 Backend.....	46
7 Současný stav webu.....	51
7.1 Závěr.....	52
8 Seznam použité literatury.....	53
9 Přílohy.....	54
A) Ukázka stránek.....	55
B) Požadavky na spuštění a instalace.....	56
C) Seznam příloh v elektronické podobě.....	56

1 PŘEDMLUVA

Každému je asi jasné, jak taková stránka vypadá a že určitě bude obsahovat odkazy na jiná místa v internetu . Odkazy tak vytvářejí složitou síť spojující stránky po celém světě a od toho také název World Wide Web neboli celosvětová pavoučí síť.

Seskupení odkazů s textem, které odkazují na další související místo, je dnes velmi populární a používá se pro něj termín hypertext. Tímto způsobem jsou třeba organizovány i nápovědy ve většině programech stejně jako i v operačním systému windows.

Doposud jsem nezminil věc, která je na Webu a internetu asi ta nejdůležitější a to je jeho nezávislost na platformě. To znamená, že stejnou stránku je možné si prohlížet na počítači s operačním systémem Windows, Linux stejně jako na počítačích Apple s OS Leopard.

Jak je této kompatibility dosaženo? Celý internet je totiž založen na daných standardech (normách), které určují možnosti komunikace v této síti, formáty používaných datových přenosů apod. Tyto standardy přitom nejsou vázány nějakými konkrétními vlastnostmi daného operačního systému, takže je lze bez jakýchkoliv obtíží použít na všech dostupných platformách.

Jednotlivé stránky mohou být následně programovány několika různými programovacími jazyky, kde jedním z nich je jazyk HTML, který slouží k popisu toho, co www-stránka obsahuje. Obecný popis zmíněného jazyka a další informace o něm jsou uvedeny v jedné z prvních kapitol.

Dalším z používaných jazyků bude bezesporu CSS, který se stará o design stránek a rozšiřuje možnosti HTML, umí lépe formátovat písma, pracovat s obrázky, rozděluje stránky do bloků. Jako poslední uvedu PHP, což je jazyk, který z obyčejných statických stránek udělá dynamické a nejen to. Díky jeho všestrannosti a možnosti použití databáze se hodí pro internetové obchody, diskusní fóra, aukce apod.

1.1 Cíl práce

Svou práci jsem rozdělil do několika bloků, kde první půlka obsahuje obecné informace o všech použitých programovacích jazycích a v druhé půlce je uveden podrobný návod na tvorbu jednoduchého dynamického webu. Celá práce je zakončena popisem samotných stránek týkající se pořádání seminářů a konferencí na TUL.

Jeden z prvních bloků je zaměřen na HTML. Uvedena je historie, vývoj a samotný popis tohoto jazyka. Vyjmenovány jsou výhody, přednosti i nevýhody použití této metody při vytváření webu spolu s ukázkami příkladů v HTML. Celý blok je zakončen popisem programů určených pro tvorbu webu a je v něm popsáno i chování prohlížečů při zobrazování stránek.

Další kapitola má za cíl vysvětlit, co znamená zkratka CSS. Jako první se řekne pár věcí z minulosti tohoto jazyka, kdo ho vytvořil a za jakým účelem a posléze je již probrána podstata CSS spolu s uvedením několika příkladů (jak si změnit pozadí stránky, formátování textu a jeho pozicování, práce s obrázky, jak je správně umístit do stránky apod.)

Jedno z posledních témat teoretické části se týká použití dynamických prvků na webových stránkách. Tedy prvků, které budou reagovat na návštěvníka a které si bude moci měnit. Jako příklad je uvedeno přihlašování do redakčního systému, vkládání příspěvků nebo odeslání dat z formuláře do MySQL databáze za pomoci PHP. Nejdříve je vysvětleno co PHP je, jak se používá a posléze jak ho správně nainstalovat a používat na počítači.

V teoretické části je tedy ukázáno, co všechno je potřeba k vytvoření dynamického webu, a v praktické je již tvorba takového webu vysvětlena a to díky podrobnému návodu, který obsahuje výpis všech souborů, které jsou zapotřebí ke zprovoznění dynamických stránek.

Poslední blok je zaměřen na samotné stránky konferencí a seminářů. Jsou v něm popsány všechny soubory, které tento web obsahuje, a je určena jejich úloha v systému.

1.2 Motivace

Webové stránky vytvářím již několik let a hodně mě to baví, proto jsem si vybral toto téma. V PHP a CSS jsem začal programovat teprve před 3 roky, kde jsem do té doby dělal pouze menší projekty v HTML. Od té doby jsem již vytvořil několik webů pro známé, mám i osobní stránky v PHP, ale nikdy nic rozsáhlejšího. Uvítal jsem tedy možnost, že mohu vytvořit projekt takových rozměrů jako jsou stránky pro Konference a semináře.

Díky tomu jsem se také sám naučil mnoho nových postupů a „zkratek“ při práci na webových stránkách a správě jejich databází.

2 WEBOVÉ STRÁNKY

Webové nebo také internetové stránky jsou formou prezentace firem či jednotlivců, na které se mohou dívat uživatelé internetu prostřednictvím internetového prohlížeče

Nezpochybnitelnou výhodou webových stránek je možnost dle potřeby je měnit a tím udržovat stále aktuální obsah (nabídku). Webové stránky pak na rozdíl od jiných typů informačních médií (tištěným, audio, video) nejen že přináší úsporu finančních zdrojů při jakékoliv změně obsahu, ale umí popřípadě také generovat zisk a to díky přílivu nových zákazníků. Dále také urychlují veškeré operace, kde bylo dříve potřeba posílat vše v tištěné podobě poštou, dnes se dá zvládnout pomocí jednoduchého formuláře na webu.

Stránky se podle provedení dělí na statické a dynamické.

a) Statické webové stránky

Statické webové stránky zastupují stránky, které mezi sebou vzájemně propojují odkazy a jsou psané pouze v HTML. Každá z těchto stránek se po naprogramování již nemění. Nevýhodou tedy je jejich obtížná úprava při dalším rozšiřování webu (nutnost předělat každou samostatně) a zejména také fakt, že nedokáží zobrazovat data uložená v databázi. Další nevýhodou je nemožnost použití jakýchkoliv prvků, které by reagovaly na uživatele. V tomto typu stránek je editace článků, fulltextové vyhledávání a další podobné prvky nemožné vytvořit.

Naopak výhodou je, že dnes již existuje spousta WYSIWYG editorů, které umožňují vytvořit web, aniž by měl autor jakékoliv znalosti HTML jazyka.

Statické www stránky se proto hodí zejména pro malé weby a jsou dávno překonány tzv. dynamickými stránkami.

b) Dynamické www stránky

Podstatou dynamických stránek je programovací technologie, program (software) a data uložená v databázi. Program (online obchod, redakční systém, elektronická aukce apod.), který je umístěný na internetovém serveru, pak na základě požadavku na zobrazení konkrétní stránky ji dynamicky sestavuje a odesílá do prohlížeče uživatele. Zjednodušeně se tak dá říci, že stránka je sestavena na základě definovaných parametrů, které určují vzhled, a dat získaných z databáze (data mohou být seznam účastníků, seznam článků nebo konkrétní článek). Sestavení zajišťuje program běžící například na technologii PHP, ASP nebo ASP.NET. Dynamické stránky tedy fyzicky neexistují v uloženém stavu na rozdíl od stránek statických. Uživatel si v nich může vyhledávat slova na stránce, přepínat jazyky pomocí databáze, vkládat příspěvky apod.

2.1 HTML

HTML je zkratkou anglických slov HyperText Markup Language, což se dá přeložit jako značkovací jazyk pro hypertext. Jedná se o jeden z jazyků, kterým se vytvářejí stránky v systému World Wide Web a který umožňuje publikaci dokumentů na Internetu. [4]

Tento jazyk je vlastně forma dříve vyvinutého univerzálního značkovacího jazyka SGML (*Standard Generalized Markup Language*), který má i dnes za úkol poskytnout standardní architekturu pro vytváření, předávání, uchovávání a zpracování různorodých dokumentů v elektronické podobě. Samotný vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.

Vývoj tohoto jazyka začal v roce 1989, kdy Tim Berners-Lee a Robert Caillau spolupracovali na propojeném informačním systému pro CERN, výzkumné centrum fyziky poblíž Ženevy ve Švýcarsku. V této době se pro tvorbu dokumentů obvykle používaly jazyky TEX, PostScript a také SGML. Berners-Lee věděl, že potřebují něco jednoduššího a v roce 1990 byl navržen jazyk HTML a protokol pro jeho přenos v počítačové síti – HTTP (*HyperText Transfer Protocol – přenosový protokol hypertextu*). Zároveň také Tim Berners-Lee napsal první webový prohlížeč, který nazval World Wide Web.

Postupně pak vycházely následující verze HTML:

Verze 0.9

Byla vydána zhruba v roce 1991. Nepodporovala grafický režim (verze, kterou vytvořil Tim Berners-Lee).

Verze 2.0

Ukazuje, v jakém stavu se nacházel jazyk v polovině roku 1994. Standard vydala roku 1995 komunita IETF (*Internet Engineering Task Force*). Je to první verze, která odpovídala syntaxi SGML. Přidává k původní specifikaci interaktivní formuláře a podporu grafiky.

Verze 3.2

Byla vydána 14. ledna 1997. Připravovaná verze HTML 3.0 nebyla nikdy přijata jako standard, protože byla příliš složitá a žádná firma nebyla schopna naprogramovat její podporu. Standard v té době vydalo už W3C, stejně jako následující verze. Přidává k jazyku tabulky, zarovnávání textu a stylové elementy pro ovlivňování vzhledu.

Verze 4.0

Byla vydána 18. prosince 1997. Do specifikace jazyka přibýly nové prvky pro tvorbu tabulek, formulářů a nově byly standardizovány rámy (*frames*). Tato verze se snaží dosáhnout původního účelu – prvky by měly vyznačovat význam (obsah) jednotlivých částí dokumentu, vzhled má být ovlivňován připojovanými styly.

Verze 4.01

Byla vydána 24. prosince 1999. Tato verze opravuje některé chyby verze předchozí. Podle původního předpokladu se mělo jednat o poslední verzi, po které by se přešlo na XHTML.

Verze 5

7. března 2007 byla založena nová pracovní skupina HTML, jejíž cílem je vývoj nové verze 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 verze bylo zvoleno HTML 5. Tato verze by měla být hotova v letech 2010-2012 a ukončení jejího vývoje se po vyřešení problémů a opravení všech chyb odhaduje až na rok 2022. [2]

2.1.1 Koncepce

Jazyk HTML je od verze 2.0 aplikací SGML (vyvíjená verze HTML5 již závislost na SGML obsahovat nebude). Je charakterizován množinou značek a jejich nastavením, která se ale mohou verzemi lišit. Mezi značky se píše částí textu dokumentu a tím se určuje význam (obsah) napsaného textu. Názvy jednotlivých značek se píše mezi úhlové závorky < a >. Část dokumentu tvořená otevírací značkou, nějakým obsahem a odpovídající ukončovací značkou tvoří tzv. *element* (prvek) dokumentu.

Tak například `<strong>` je otevírací značka pro zvýraznění textu a `<strong> Nějaký text </strong>` je prvek obsahující zvýrazněný text. Součástí obsahu prvku mohou být další vnořené prvky. Atributy jsou doplňující informace, které upřesňují vlastnosti jednotlivých prvků. [2]

Značky jsou většinou párové, kde značka na konci je shodná se značkou na začátku, jen má před svým názvem znak lomítka / .

Příklad pro označení odstavce:

```
<p>Text odstavce</p>
```

Některé značky jsou nepárové – nemají žádný obsah a nepoužívají koncovou značku.

Příklad nepárové značky pro nový řádek (tlačítko enter):

```
<br>
```

Značky mohou obsahovat prvky, které popisují jejich vlastnosti nebo nesou jinou informaci. Příkladem může být odkaz (značka `a`), jehož prvek `href` říká, kam se uživatel po kliknutí na něj dostane (v tomto příkladu na stránku `http://google.com`):

```
<a href="http://google.com">text odkazu</a>
```

Nebo další možnosti pro zápis odkazu – odkaz, který se otevře v novém okně/panelu:

```
<a href="http://google.com" target="_blank">text odkazu</a>
```

A jiná možnosti zápisu – odkaz, který se otevře v novém okně / panelu, obsahuje titulek (zobrazí se po najetí kurzorem na daný odkaz) a má přiřazenou třídu, kterou pak můžeme využít například v CSS:

```
<a href="http://google.com" target="_blank" title="titulek"
class="navez_tridy">text odkazu</a>
```

Pro každou verzi existuje definice pravidel DTD (*Document Type Definition*). Od verze 4.01 musí být odkaz na deklaraci DTD v dokumentu uveden pomocí klíčového slova `DOCTYPE`.

DTD definuje pro určitou verzi elementy a atributy, které lze používat.

Dokument může mimo značkování obsahovat další prvky:

- Direktivy – začínají znaky `<!`, jsou určeny pouze pro prohlížeč.
- Komentáře – pomocné texty pro programátora, nejsou součástí obsahu dokumentu a nezobrazují se (prohlížeč je ignoruje). Příklad komentáře je uveden níže.
- Kód skriptovacích jazyků. [2]

2.1.2 Struktura dokumentu

Stránky napsané v HTML mají předepsaný tvar

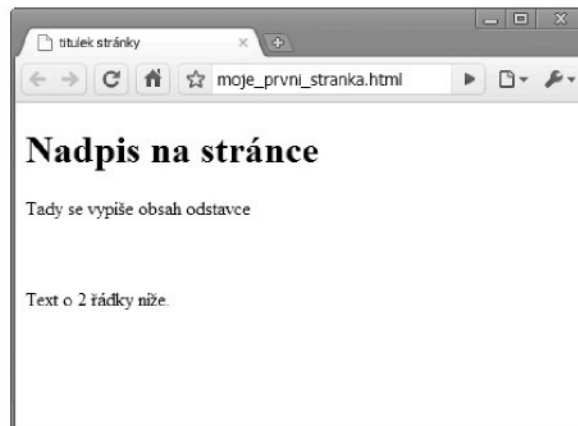
- Deklarace DTD – je povinná až ve verzi 4.01, je uvedena direktivou `<!DOCTYPE`.
- Kořenový element – element `html` (značky `<html>` a `</html>`) označuje celý dokument. Kořenový element je povinný, ale otevírací a ukončovací značka samotná povinná není (pokud tyto značky nebudou v těle dokumentu uvedeny, prohlížeč si je sám doplní podle obsahu).

- Hlavička elementu – obsahuje metadata, která se vztahují k celé stránce. Definují např. název dokumentu, jazyk, kódování, klíčová slova, popis, použitý styl zobrazení apod. Hlavička se zapisuje mezi značky <head> a </head>. Element head je opět povinný, ale jeho otevírací a koncová značka povinná není, prohlížeč ji sám doplní podle obsahu.
- Tělo dokumentu – obsahuje vlastní text stránky. Vymezuje se značkami <body> a </body>. Element body je povinný, ale jeho otevírací a koncová značka povinná není, prohlížeč ji sám doplní podle obsahu. [2]

Příklad jednoduché stránky napsané ve verzi HTML 4.01 vypadá pak takto:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <!-- toto je poznámka která se nezobrazí -->
  <head>
    <title>titulek stránky</title>
  </head>
  <!-- samotné tělo dokumentu -->
  <body>
    <h1>Nadpis na stránce</h1>
    <p>Tady se vypíše obsah odstavce</p>
    <br><br>
    Text o 2 řádky níže.
  </body>
</html>
```

Zobrazení této stránky je na obr.1.



Obr.1: Webová stránka v HTML

2.2 Zobrazování v prohlížečích

Webové prohlížeče jsou programy, jejichž úkolem je prezentovat stránky na nějakém zobrazovacím zařízení, nejčastěji monitoru počítače. Stránka je prohlížečem načítána a rozkládána na jednotlivé části kódu. Prohlížeč pak obsahuje tabulku značek, které podporuje. Moderní prohlížeče jako je Opera a Mozilla dokonce umožňují „přidávat“ vlastní značky a třídy a umožňují tak jejich nastavování pomocí kaskádových stylů (CSS).

Protože ale nejrozšířenější prohlížeč Windows Internet explorers tuto možnost nepodporuje a obsah neznámých prvků zobrazí zcela normálně, bez stylu, tak se vlastní prvky prakticky nepoužívají.

Každému prvku je pak přiřazen *styl* (způsob zobrazení). Styly mohou být uvedeny ve stylovém předpisu. Vlastnosti stylů, které nejsou předepsány, doplní prohlížeč podle základního stylu, který má definován. Některé prohlížeče umožňují uživateli základní styly definovat.

Moderní prohlížeče pracují obecně ve dvou základních režimech:

- *Standardní režim* – režim snažící se dodržovat definované standardy;
- *Quirk mód* – režim zaměřený na zpětnou kompatibilitu, i pokud takové chování není v souladu se standardy.

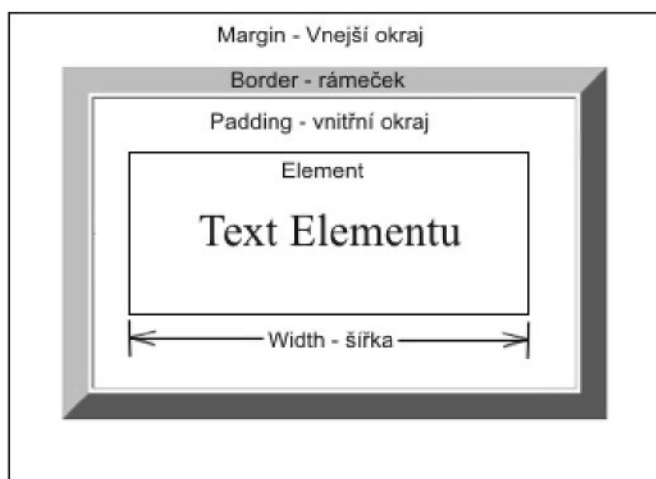
Tyto režimy chování zavedl Internet Explorer ve své páté verzi z důvodu zpětné kompatibility. Microsoft při vytváření nové verze prohlížeče zamýšlel, aby se v něm zobrazovaly správně stránky, které již existují, ale také nové stránky, které jsou psány podle nových standardů. To, jaký režim (a chování) prohlížeč použije, závisí hlavně na uvedení direktivy `<!DOCTYPE`, protože většina starších stránek ji vůbec neobsahuje. Později tyto typy chování částečně přebrala Mozilla i Opera, kde však rozdíly mezi režimy nejsou zas tolik markantní. Existence nestandardního režimu je výsledkem vývoje, kdy si výrobci jednotlivých prohlížečů upravovali definici HTML podle svých potřeb a prohlížeče podporovaly nestandardní elementy a syntaxe. Řada těchto „vylepšení“ byla následně přejímána do standardů, některé však byly zase v dalších verzích odstraněny. [1]

Rozdílů v jednotlivých režimech je pak několik, ale za podstatné se dají považovat tři:

- jinak se počítá šířka a výška (boxmodel)
- jinak zobrazují velikost písma (quirk mód má písmo o stupeň větší)
- quirk mód dovoluje nastavovat rozměry řádkovým prvkům

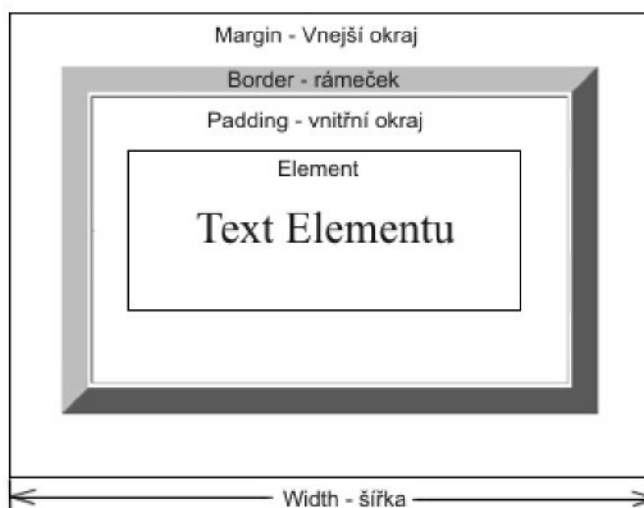
Tyto rozdíly pak mohou zapříčinit, že stejná webová stránka vypadá jinak na různých typech prohlížečů. Obrázky jsou pokaždé na trochu jiném místě, text má větší font a je jinak odsazený od okraje, stránka má nestandardní rozměry. [1]

Prohlížeč si jednotlivé elementy na stránce rozdělí do obdélníků a v nich určuje rozměry, čemuž se také říká boxmodel. Tato specifikace udává (a standardní mód to respektuje), že do rozměrů prvků se nepočítá rámeček (border) a vnitřní okraj (padding), což je ukázáno na obr.2.



Obr.2: Správná interpretace okrajů

Oproti tomu quirk mód (a zejména Internet Explorer) do šířky počítá border i padding, což je zobrazeno na obr.3



Obr.3: Chybná interpretace vlastnosti width u IE

Proto některé stránky mohou vypadat jinak v Internet Exploreru a jinak v Mozille a Opeře. [1]

2.3 Editory HTML

Editory HTML jsou programy určené pro tvorbu webových stránek. Zvládají tak vytvořit soubory typu HTM nebo HTML ty propracovanější i PHP a jiné formáty. Editory HTML se dají obecně rozdělit na textové editory, do kterých se vkládá samotný HTML kód a na WYSIWYG editory, které pracují již s hotovou stránkou.

a) Textové editory

Textovým editorem HTML může být ve své podstatě jakýkoliv program pracující s textem, třeba i Poznámkový blok ve Windows. V praxi se ale používají mnohem sofistikovanější programy, které velmi urychlují a usnadňují práci. Běžný editor HTML třeba zvládá barevně zvýraznit jednotlivé části kódu, jako například HTML značky či atributy od normálního textu, a určit tak, kde značka začíná a kde končí, dokáže „napovídat“ značky, zná chytré tabulátory nebo zvládá opravit dokument podle předepsané DTD. Mezi takové nejrozšířenější editory HTML u nás určitě patří PSPpad. [2]

b) WYSIWYG editory

WYSIWYG je zkratka od anglického „What you see is what you get“, což se dá přeložit jako „Co vidíš, to dostaneš“. Tyto WYSIWYG editory pracují na opačném principu než textové editory – ve WYSIWYG editoru se pracuje s již vytvořenou stránkou od programu a obecně neplatí, že by uživatel tohoto editoru musel znát jazyk HTML. Ve WYSIWYG editoru si může uživatel poskládat stránku jak se mu zlíbí a program následně vygeneruje požadovaný kód HTML sám. Mezi nejznámější takovéto editory patří *Adobe Dreamweaver* nebo *Expression Web* (novější verze *Microsoft Frontpage*). [2]

PsPad

PsPad je celosvětově rozšířený freewareový textový editor a editor zdrojových kódů pro platformu Microsoft Windows vyvíjený v prostředí Delphi. Program vyvíjí moravský programátor Jan Fiala, první verze vyšla v roce 2001.

PsPad je navržen jako univerzální editor pro editaci prostých textů a zdrojových kódů mnoha programovacích, skriptovacích a značkovacích jazyků. Umožňuje správu projektů, manipulaci s více soubory najednou pomocí moderního rozhraní MDI (Multiple Document Interface), průzkumník kódů a zvýrazňování syntaxe. Obsahuje také značné množství interních nástrojů např. FTP klienta, HEX Editor, průzkumník kódu a další.

Možnosti zpracování textu jsou v PsPadu poměrně široké – editor podporuje makra, sloupcové bloky, vyhledávání a nahrazování s podporou regulárních výrazů, různé operace s označeným blokem textu (např. převod na velká / malá písmena, slučování nových řádků nebo pevné zarovnávání na nastavitelnou šířku řádky) a další. V nastavení nechybí pokročilé možnosti jako chytré, skutečné nebo mezerami nahrazované tabulátory. [6]

3 CCS – KASKÁDOVÉ STYLY

CCS (*Cascading Style Sheets*) jsou v překladu kaskádové styly, což je jazyk pro určení způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML.

CCS byl sestaven standardizační organizací W3C, kde autorem prvotní verze byl Håkon Wium Lie. Doposud byly vydány zatím dvě úrovně specifikace CSS1 a CSS2, dokončuje se revize CSS 2.1 a pracuje se na verzi CSS3. Hlavním úkolem kaskádových stylů je umožnit vývojářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML ve svých starších verzích, ale kvůli nedostatečným standardům a konkurenčnímu boji výrobců prohlížečů se vyvinul jinak. [2]

CSS se do dokumentu vkládá díky tzv. tabulkám kaskádových stylů, které se skládají z několika pravidel. Každé pravidlo obsahuje název (selektor) a část s deklarací. Každá část s deklarací pak obsahuje seznam samotných nastavení, následuje dvojtečka : a hodnota vlastnosti. Deklarace je ukončena středníkem ;. Před středníkem ještě může nepovinně být označení !important, které zvýší sílu deklarace.

Příklad pravidla:

```
body {  
    background-color: black;  
    color: white;  
    padding: 10px !important;  
}
```

Celý blok má označení „body“ (selektor) a pod tímto názvem se bude volat v programu, v závorkách následuje část s deklarací, řádek „background-color: black;“ je samotná deklarace, která nastavuje pozadí na černo, „background-color“ je tedy vlastnost a „black“ je hodnota. Celý kód pak nastavuje barvu pozadí na černo, barvu textu na bílo a odsazení textu od okraje na 10 pixelů.

Používání kaskádových stylů má oproti samotnému HTML mnoho výhod. Některé z nich jsou:

- Větší možnosti formátování. Například 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 se to dalo provést složitým vytvořením vnořených tabulek.
- Jednodušší údržba webové prezentace. Pokud je nutné změnit nějaký detail, jako třeba barvu písma nebo pozadí, nemusí se složitě procházet HTML kód nebo různé HTML šablony a přepisovat je, ale může se změnit pouze jedna vlastnost v CSS souboru, který je načten ve všech stránkách.
- Oddělení struktury a stylu. V jednom (HTML) dokumentu je napsaný obsah s texty a v druhém (CSS) dokumentu je definice vzhledu stránek. Tím se dosáhne snadnějšího strojového zpracování samotného obsahu stránek, do kterého se nepletou prvky definující vzhled.
- Načítání (ukládání) stylů – webový prohlížeč si může soubor se styly uložit do své vnitřní paměti, čímž může být dosaženo rychlejšího načtení stránky. Na druhou stranu při použití externího CSS souboru dochází k dalšímu HTTP požadavku navíc, na rozdíl od toho, když by bylo použito buď přímé vložení CSS do dokumentu nebo přímo formátování HTML.

- Mohou také existovat různé styly pro různá výstupní zařízení. Jednoduše se může nadefinovat různý styl pro tisk, projekci, mobil, PDA, běžný prohlížeč či dokonce pro každý prohlížeč jiný styl. Jazyk CSS nezapomíná ani na zrakově postižené – je možno napsat styly pro hlasový syntetizátor nebo hmatovou čtečku Braillova písma.
- Samotný uživatel si může napsat svůj vlastní CSS styl pro libovolnou stránku. Většina prohlížečů nějakým způsobem podporuje uživatelské styly, takže uživatel si může například nastavit, aby měl všechny odkazy na všech stránkách vždy podtržené nebo aby na tomto konkrétním webu mělo písmo vždy černou barvu.

Mezi nevýhody CSS patří to, že není vždy dostatečně podporován v nejpoužívanějších prohlížečích. Prohlížeče obsahují v implementaci CSS chyby a je někdy opravdu nesnadné napsat kód tak, aby se ve všech (resp. v běžných) prohlížečích výsledek zobrazil stejně. Situace se poslední dobou velmi zlepšuje, mj. opravami chyb v prohlížečích a příchodem nových verzí Internet Exploreru, který byl ve své šesté verzi zdrojem častých chyb v CSS. Aktuální verze IE9 by již měla podporovat i CSS3.

Přes nespočet výhod a minimum nevýhod zde je i pár limitujících faktorů:

- CSS selektory neposkytují přístup k nadřazeným (rodičovským) prvkům, jako to umožňují složitější jazyky, což může být například XPath. Nelze například nastylovat jen ty odstavce, které obsahují odkaz.
- Horizontální vkládání prvků do stránky je intuitivní a jednoduché, naopak vertikální stylování už působí problém a prvky mohou různě „odletovat“. Například na obvyčejné vertikální centrování obsahu se musí používat poměrně složité kombinace nastavení, které ještě nemusí být úplně kompatibilní.
- CSS neposkytuje možnost pro symbolický zápis proměnné nebo konstanty, všechny hodnoty musí být vepsány přímo v kódu. Například pokud se na vícero místech používá stejná barva, nemůže se použít symbolický zápis `barva=red`; a poté už jen psát proměnnou `barva`, všude se musí uvádět přímo hodnota `red`.
- CSS neumí počítat výrazy, prohlížeče třeba vůbec neporozumí výrazu `margin-left: 10% - 3em + 4px`; Webdesigner si to vše musí spočítat sám a napsat přesné hodnoty.
- CSS2 nenabízí žádnou možnost pro tvorbu kulatých rámečků nebo jiných kulatých objektů. Pracuje pouze s obdélníky.
- CSS2 nenabízí žádnou možnost, jak jednomu elementu přiřadit více obrázků na pozadí.

3.1 Selektory

CSS obsahuje mnoho různých selektorů, které lze kombinovat. Mezi základní patří:

- `body` – Toto nastavení bude platit pro všechny výskyty elementu `body`.
- `body p` – Toto nastavení bude platit pro všechny elementy `p`, které se nachází v elementu `body`, v jakémkoliv místě souboru.
- `body>div` – Toto nastavení bude platit pro všechny elementy `div`, které jsou přímo pod elementem `body`. To znamená, že pokud bychom měli element `div`, který se nachází v `<body><blockquote><div>...`, toto nastavení by pro něj neplatilo, protože tento `div` není přímým potomkem elementu `body`.
- `.trida` – Toto nastavení bude platit pro všechny elementy, které mají v HTML nastavenou třídu `trida`. To se provádí pomocí HTML atributu `class`
`<div class="trida">.....</div>`
- `#id` – Toto nastavení bude platit pro všechny elementy, které mají v HTML nastavený identifikátor `id`. To se provádí pomocí HTML atributu `id`
`<div id="trida">...</div>`

- sel1, sel2, sel3 – Pokud chceme pro více selektorů stejná nastavení, uděláme to pomocí čárek. Následující nastavení pak budou platit pro všechny selektory.

Pro definování jednotlivých selektorů v HTML dokumentu se využívá jednoho z následujících postupů:

1. Přímé vložení stylu do dokumentu pomocí atributu `style`. Tato pravidla budou aplikována pouze na dotýčný element.

Příklad:

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

2. Zápis stylů do elementu `style`. Takové styly se aplikují na celou stránku podle předepsaných selektorů.

Příklad:

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

```
</style>
```

Což vytvoří element „hlavicka“ o velikosti 200px na 450px.

3. Načtení stylů z externího souboru pomocí elementu `link`, který se obvykle zapisuje do hlavičky stránky.

Příklad:

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

4. Připojení externího souboru pomocí http hlavičky `link`.

Příklad:

```
Link <fancy.css>; rel=stylesheet, <fancy.css>;rel=stylesheet
```

Samozřejmě stačí ovládat jenom jeden způsob. Já nejčastěji používám externí css soubor.

Občas se používají podmíněné komentáře, které umožňují definovat na úrovni HTML kód, který bude viditelný pouze Internet Exploreru, či naopak. Příklad:

```
<!--[if IE]>  
<style type="text/css">  
#upozorneni {  
    color: red;  
}  
</style>  
<![endif]>
```

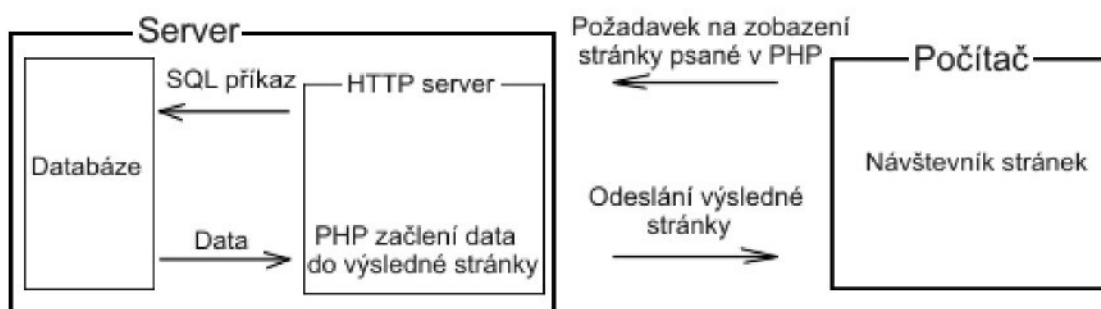
Tento skript bude přijímat pouze Internet Explorerem, ostatní prohlížeče uvidí obyčejný HTML komentář a na vnitřní kód tak nebudou reagovat.

4 PHP

PHP (anglicky: *Hypertext Preprocessor*, což se přeloží jako Hypertextový preprocesor, původně *Personal Home Page*) je skriptovací jazyk, který se používá především pro programování dynamických webových stránek. Nejčastěji se zařazuje přímo do struktury jazyka HTML, XHTML či WML, kde se tohoto využívá právě při tvorbě webových aplikací. PHP se také může použít i k tvorbě konzolových a desktopových aplikací.

PHP skripty jsou prováděny na straně serveru, k uživateli je přenášén až výsledek jejich činnosti. Základy jazyka jsou inspirovány několika programovacími jazyky (Perl, C, Pascal a Java). Umí pracovat s různými druhy knihoven - např. zpracování textu, grafiky, práci se soubory, přístup k většině databázových systémů (mj. MySQL, ODBC, Oracle, PostgreSQL, MSSQL), podporuje celou řadu internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP...) [3]

Schéma zpracování PHP skriptů



Obr.4: Princip sestavování PHP stránek

PHP se hojně rozšířilo především díky jednoduchosti použití a tomu, že spojuje vlastnosti více programovacích jazyků a nechává tak vývojáři částečnou svobodu v psaní kódu. Obvykle se používá v kombinaci s operačním systémem Linux (na windows funguje samozřejmě také), databázovým systémem (většinou MySQL nebo PostgreSQL) a webovým serverem Apache. Pro tuto kombinaci se vžila zkratka LAMP – tedy spojení Linux, Apache, MySQL a PHP nebo Perl.

Mezi základní vlastnosti PHP patří:

- Jazyk PHP je dynamicky typový, tzn. že datový typ proměnné se určí v okamžiku přiřazení hodnoty.
- Kvůli tomu se používají v PHP dva operátory porovnání: `==` a `===` – při použití prvního dochází před porovnáním ke konverzi, při použití druhého je výraz pravdivý, jen když jsou obě dvě proměnné stejného datového typu a jejich obsah má stejnou hodnotu.
- Pole se dají indexovat číselnými hodnotami (jako v jazyce C) nebo mohou fungovat jako šifrovací tabulky. Jedno pole může obsahovat oba typy indexů.
- Pole jsou heterogenní (stejně pole může mít prvky různých typů).

4.1 Data vydání hlavních verzí

Řada	Verze	Datum vydání	Poznámka
1.x	1.0.0	8. června 1995	Oficiální název „Personal Home Page Tools (PHP Tools)“. Poprvé byl použit název „PHP“.
2.x	2.0.0	16. dubna 1996	Autor chtěl "nejrychlejší a nejjednodušší nástroj" pro tvorbu dynamických webových stránek.
3.x	3.0.0	6. června 1998	Tým vývojářů se z původní jedné osoby rozšířil na více osob. Zeev Suraski a And Gutmans přepsali celý základ pro tuto verzi.
	4.0.0	22. května 2000	Přidán pokročilý dvoustupňový systém parse/execute (analyzovat/vykonat) syntaktické analýzy tagu - Zend engine.
	4.1.0	10. prosince 2001	Byly představeny 'superglobalní proměnné' (\$_GET, \$_POST, \$_SESSION, etc.)
	4.2.0	22. dubna 2002	zakázání register_globals ve výchozím nastavení. Data přijatá prostřednictvím sítě nejsou přímo vkládány do názvů globálních proměnných, uzavírá se tím možnost využití bezpečnostních děr.
4.x	4.3.0	27. prosince 2002	Představeno CLI, jako doplněk CGI.
	4.4.0	11. srpna 2005	Dodány stránky nápovědy pro skripty phpize a php-config.
	4.4.8	3. ledna 2008	Oprava některých bezpečnostních prvků a oprava chyb. Měla to být poslední update verze PHP 4. Bezpečnostní updaty se vydávají do 8. srpna 2008
	4.4.9	7. srpna 2008	Oprava některých bezpečnostních prvků a oprava chyb. Poslední vydání verze ze série PHP 4.4.
	5.0.0	13. července 2004	Zend Engine II s novým objektovým modelováním.
	5.1.0	24. listopadu 2005	Zlepšení výkonu zavedením kompilátoru proměnných v přepracovaném Engine PHP.
	5.2.0	2. listopadu 2006	Povolen filtr přípon ve výchozím nastavení.
	5.2.6	1. května 2008	oprava bezpečnostních děr a chyb
5.x	5.2.8	8. prosince 2008	oprava chyb
	5.3	30. června 2009	jmenné prostory, oprava chyb, změny ve funkcích a doplňcích
	5.3.1	19. listopadu 2009	oprava chyb
	5.3.2	4. března 2010	oprava chyb, přidání podpory SHA-256 a SHA-512

[3]

Význam

červeně	staré verze; již nepodporované
žlutá	stará verze; stále podporovaná
zelená	Současné verze

4.2 Výhody a nevýhody PHP

PHP je poměrně zajímavý nástroj pro tvorbu webových aplikací, který kombinuje několik různých programovacích jazyků, a pokud někdo ovládá třeba Cčko, nebude mu pochopení základů PHP činit žádné potíže. Bohužel jako každý programovací jazyk má i tento své klady a zápory.

a) Výhody

- PHP je specializované na webové stránky.
- Základní knihovna PHP obsahuje rozsáhlý soubor funkcí, další funkce jsou v PECL.
- Standardně podporuje mnoho databázových systémů.
- Multiplatformnost (zejména Linux a Microsoft Windows).
- Možnost využití základních funkcí operačního systému (možná nekompatibilita s jiným OS).
- Rozsáhlá podpora na internetu – PHP je prakticky standardem, o které se dá najít spousty článků a internetových fór.
- Velké množství projektů a kódů, které lze zdarma využít (WordPress, phpBB a další).
- Poměrně podrobná dokumentace.
- Velmi svobodná licence – svobodnější než GPL.

b) Nevýhody PHP

- Poměrně složitý vývoj v minulosti, který si s sebou PHP nese dosud (hodně se měnící příkazy či jejich parametry atd...).
- Horší podpora Unicode (tabulky znaků), pouze přes PHP knihovnu (v PHP 6 má být již Unicode podporován standardně).
- V základní distribuci chybí ladící nástroj pro odstranění chyb.
- Po zpracování požadavku neudrhuje obsah aplikace, vytváří jej vždy znovu od začátku, což oslabuje výkon.
- Některá bezpečnostní opatření jako `safe_mode`, `open_basedir` jsou implementovány ne zrovna elegantně (pro PHP6 se uvažuje o jejich zrušení).

4.3 Příklad

V této kapitole předvedu jednoduchý příklad stránky programované v PHP.

```
<?php //každý php skript začíná těmito znaky
//mimořádně poznámky se píšou za dvě lomítka a nebudou vůbec zobrazeny
// V prvním příkladě je proměnná typu string (tečka spojuje více řetězců)
$retez = "Ahoj, jak se vede" . ', mě dobře' . " a nevadí, že je použito
několik oddělovačů";

// V tomto příkladě je proměnná číslo (int)
$promena = 100;

//Vypíše proměnou promena
echo"$promena";

//Opět vypíše proměnou promena
printf('%s',$promena);

// Do proměnné je možné vložit pole, které obsahuje jak čísla, tak znaky či
další pole
$pole = array('a', 'b', 1, 2, array('první' => 'podpole', 'vytištěno'));

// Nenahlásí chybu (jenom varování) a vytiskne 'Array'
print($pole);

// Vypíše obsah proměnné pole
print_r($pole);

// Test porovnání
$cislo = 100;
$retez = '100';

// Toto porovnání ('==') platí díky automatické typové konverzi
if ($retez == $cislo)
{
    echo 'Shodují si';
}

// Ale porovnání pomocí '===' neplatí, protože nejsou stejného typu
if ($retez === $cislo)
{
    echo 'Shodují se';
}
else
{
    echo 'Neshodují se';
}

//celý php skript se ukončí znaky
?>
```

4.4 Požadavky PHP

1. Přístup na internetový server, který podporuje PHP (v dnešní době existuje řada free serverů nebo serverů, které poskytují své místo za rozumnou cenu).
2. Domácí stanice (počítač), na které je nainstalovaný server pro ladění PHP skriptů (převážně o tom bude tento blok).

Oba servery -- internetový i domácí (testovací) musejí mít nainstalovány některé programy:

1. HTTP server, například Apache nebo IIS
2. podporu pro PHP na tomto serveru
3. program pro práci s databází např. MySQL

Tyto programy musejí být nakonfigurovány pro vzájemnou spolupráci. U ostrého internetového serveru všechno zpravidla nainstaluje a nakonfiguruje jeho správce, ale testovací server si musí každý nainstalovat sám. Na něm si lze napsat a otestovat skripty, které se potom odesílají na ostrý internetový server. [7]

a) Internetový server

Některé hostingové servery nabízejí umístění stránek s podporou PHP zdarma. Jejich úroveň ovšem není tak kvalitní jako u placených poskytovatelů. Mezi freehostingy s podporou PHP patří např. *webzdarma.cz*, *ic.cz*, *php5.cz* nebo *php5.sk*.

V praxi je výhodnější si zaplatit server u nějakého placené poskytovatele pro jeho lepší a rychlejší připojení a tím plynulejší chod stránek.

Pokud je na počítači trvalé připojení k Internetu, nemusí se ani vytvářet testovací server, a může se vyvíjet rovnou na ostrém serveru. Není to ale úplně dobrý nápad, už třeba z důvodů zálohování, testování nebo rychlosti práce se testovací instalace doporučuje založit. Po nastavení domácího (testovacího) serveru podle toho internetového se mohou PHP soubory po otestování posílat přes FTP klienta na internet. Pokud je testovací server správně nastaven a skripty na něm fungují, tak by měly fungovat i na ostrém serveru. [7]

b) Nejjednodušší domácí instalace

Existují relativně snadné instalace i pro uživatele Windows: balíčky programů PHPTriad, PHP Home, Intranetový server nebo EasyPHP. Tyto balíčky obsahují všechny tři potřebné součásti pro správné fungování na OS Windows.

1. server Apache
2. modul PHP pro Apache
3. databázi MySQL

Seznam PHP balíčků s adresami, kde se dají stáhnout:

- EasyPHP má domovskou adresu <http://www.easyphp.org>
- PHPTriad má domovskou adresu <http://www.slunecnice.cz>
- PHP Home má domovskou stránku <http://sourceforge.net/projects/phphome/>
- Complex web server (česky) od Miloslava Pankráce se dá stáhnout na stránkách <http://ponkrac.net/complex-web-server/cs>
- nebo třeba XAMPP, který obsahuje i na jiné jazyky (např. Perl) má domovskou adresu <http://www.apachefriends.org/>

Všechny uvedené programy mají freeware licenci, takže jde o plné verze programů bez jakéhokoliv omezení. Zdarma přitom neznamená špatnou kvalitu -- Apache a MySQL jsou programy na profesionální úrovni.

Další výhodou zmíněných balíčků je, že po nainstalování jsou všechny 3 programy nastavené a nemusí se už nic předělávat.

V případě EasyPHP se spustí instalační program a ten provede vše potřebné sám. Úspěšnost instalace se zkontroluje po jeho spuštění, kdy se ukáže Apache started a MySQL started. Podobně je to i u ostatních balíčků. [7]

c) Alternativní instalace

Pokud nevyhovují zmíněné balíčky, Intranetový server nebo služba PHP Home, je možné stáhnout si jednotlivé programy a nakonfigurovat je ručně. Domovské stránky programů lze najít podle jejich názvu.

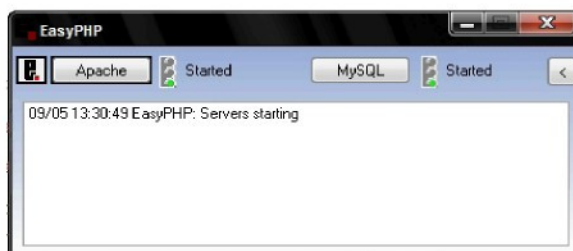
Pokud internetový server, na který se bude publikovat, používá nějaký jiný software než Apache a MySQL tak by bylo dobré, aby testovací konfigurace co nejlépe odpovídala podmínkám, které fungují na "ostrém" serveru, aby nevznikaly problémy po odeslání testovaných stránek.

Alternativou k Apachi jsou servery od Microsoftu. IIS (Internet Information Server) je špičkou v oboru, ale není zdarma. Oproti tomu PWS (Personal Web Server) zdarma je, ale není moc kvalitní.

U databází je výběr větší. Na linuxových serverech bývá často databáze PostgreSQL. Na windowsových IIS lze zase přes rozhraní ODBC použít jako databázi obyčejnou tabulku Excelu nebo velkou databázi Accessu (což jde i s Apachem a balíčky). [7]

d) Práce se serverem Apache

Po instalaci balíčku EasyPHP se Apache ve Windows spouští automaticky s programem. Je možné ho také vypnout nebo zapnout manuálně v programu.



Obr.5: Rozhraní EasyPHP

Jestli je EasyPHP nebo jiný balíček správně nainstalovaný se pozná tím, že po spuštění začne fungovat testovací server. Na testovací server se přistupuje zadáním adresy `http://localhost` do prohlížeče. Na této adrese se nachází kořenová složka serveru.

Pokud se pracuje na lokální síti s konfigurovaným DNS serverem, může se stát, že `http://localhost` nebude fungovat. Pak je třeba zkusit `http://` + jméno počítače, např. `http://muj_pocitac/`. Pokud ani to nefunguje, zadává se ještě vlastní IP adresa nebo číslo `http://127.0.0.1/`, což je IP adresa každého lokálního počítače.

Může se také teoreticky stát, že Apache bude chybně fungovat kvůli jiné aplikaci, která náhodou také pracuje na portu 80. Řešením je buď tuto aplikaci vypnout, nebo do souboru `httpd.conf` nastavit nějaký jiný port (např. Port 8080). Poté se na server přistupuje přes adresu `http://localhost:8080`. [7]

e) Kořenová složka

Kořenová složka by měla obsahovat všechny publikované soubory a skripty. Aby PHP skripty správně fungovaly, musejí být uloženy v této složce nebo jejich podadresářích - jinak řečeno musí k nim být přístup přes adresu `http://localhost`.

Apache má při instalaci v EasyPHP nastavený jako kořenovou složku `c:\EasyPHP\www\`. Tuto složku lze změnit nastavením konfigurace v souboru `httpd.conf`.

f) Konfigurace Apache

Všechna nastavení serveru Apache se provádí v souboru `httpd.conf` (upravit se dá v kterémkoli textovém editoru). Tento soubor se nachází ve složce `c:\EasyPHP\apache\conf\`. Co určuje každý řádek se dá pochopit z komentářů a nápověd. (Míříž # na začátku řádku je komentář). Správným nastavením je možné vytvořit virtuální servery, přesměrování, přihlašování a podobné věci. V první chvíli ale není nutné nastavovat nic.

Příklad pro změnu kořenové složky se provádí přepsáním řádku:

```
DocumentRoot "C:/ EasyPHP/www"
```

třeba na

```
DocumentRoot "c:/moje_slozka"
```

Změny v konfiguračním souboru se projeví až po restartu Apache. Ten lze provést restartováním celého EasyPHP nebo položkou `restart Apache` v programu. [7]

g) Vyhodnocování PHP

Na vytvořené dokumenty v PHP se vždy přistupuje přes lokální server. Příklad takové adresy může být: `http://localhost/muj_projekt.php`. Pokud se PHP soubor otevře přímo z pevného disku (např. `c:\EasyPHP\www\muj_projekt.php`), tak skripty v něm obsažené nebudou fungovat, protože nejsou zpracovány serverem. Jakýkoliv soubor s PHP skripty se tedy musí odeslat na server, ten ho zpracuje a uživateli odešle výsledek. Například se na server odešle řádek `<echo>Nějaký text</echo>` a v prohlížeči se zobrazí pouze „Nějaký text“. V tom je velká výhoda a bezpečnost PHP, že uživateli se nezobrazuje celý PHP skript ale pouze jeho výsledná forma již zpracovaná serverem.

h) Konfigurace PHP

Veškeré nastavení PHP skriptů se provádí v souboru `php.ini`, který má EasyPHP umístěný v adresáři `C:\EasyPHP\Apache`. Dříve se tento soubor jmenoval `php3.ini` a byl v systémové složce Windows, ale to už dnes neplatí. V souboru `php.ini` obvykle není třeba nic měnit, alespoň zpočátku. Význam jednotlivých direktiv bývá patrný z komentářů. Podpora MySQL je v posledních distribucích různých instalačních balíčků už zahrnuta také automaticky. [7]

i) Nastavení SMTP

Pokud je potřeba z testovacího serveru odesílat pomocí PHP maily, musí se nastavit adresa SMTP. SMTP je server, který zvládne (a je ochoten) vzít odesílaný mail a poslat ho dále do Internetu. Normálně je v `php.ini` nastaveno `smtp = localhost`, což ve Windows samozřejmě nefunguje. Je třeba nastavit adresu smtp serveru, na který se bude možné připojit. Pokud je připojení přes telefonní linku, je SMTP server určený poskytovatelem, např. `smtp = smtp.volny.cz` nebo podobně. V ostatních případech je dobré se poradit se správcem, kdo je poskytovatelem připojení a zda nemá vlastní SMTP server. Na ostrých serverech, které používají Linux, bývá nastaveno `smtp = localhost`, (SMTP server je ten samý stroj). To je pro běžného uživatele k ničemu, protože na ostrém serveru většinou nemá možnost upravovat `php.ini`, to provádí jeho správce.) [7]

5 TVORBA PROJEKTU

Po teoretické části je již zřejmé, co je potřeba znát při tvorbě dynamického webu. V dalších kapitolách jsou uvedeny všechny soubory, které takový web obsahuje i s jejich popisem. Každá ze stránek je vytvořena v HTML s použitím prvků CCS i PHP a data se ukládají do MySQL databáze. Na tomto projektu je ukázáno, jak vypadá taková struktura a kód dynamického webu. Samotné stránky konference jsou pak vysvětleny jen z funkčního hlediska.

Nejdříve ale ještě stojí zmínit, že skoro všechny poloprofesionální až profesionální weby s dynamickým obsahem mají koncepci "frontend-backend". To jednoduše znamená, že existují stránky pro uživatele stránek (tedy frontend), na nichž je zobrazeno to, co potřebuje návštěvník webu vidět, a stránky pro administrátora, kde je zase naopak to, co potřebuje ke vkládání dat (backend). Někde "mezi" tím vším jsou data k zobrazení, převážně uložená v databázi. Celé rozložení je na obrázku 7. [4]



Obr.7: Struktura webu

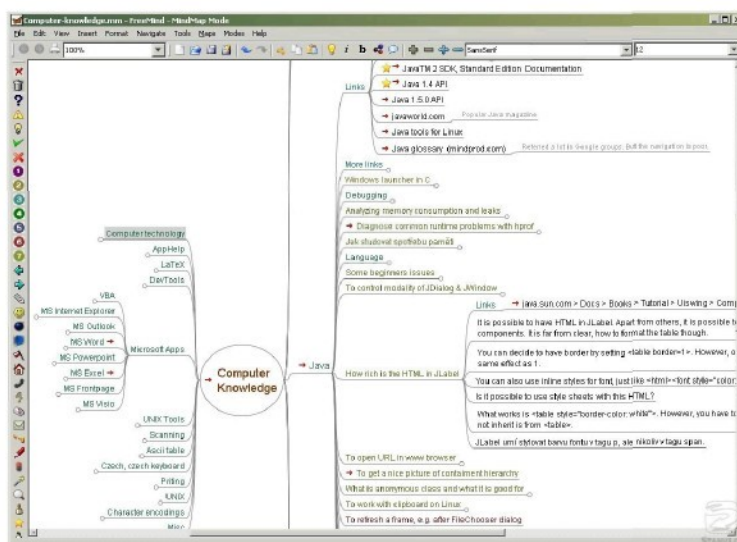
Uvádím to proto, že při práci na následujícím projektu je nejdříve popsána část pro návštěvníky stránek (výpis položek z databáze do webu) a poté až část pro administrátory, kteří příspěvky zakládají a upravují.

Jako první věc, než se začne se samotným navrhováním stránek, je určitě dobré si uvědomit, jak budou stránky vypadat a jakou budou mít funkci (co budou zobrazovat – osobní web, firemní stránky, eshop) a tomu přizpůsobit jejich ergonomii, tedy rozmístění odkazů, textu, obrázků apod. Viděl jsem několik webů, které měly vynikající design, ale nedalo se v nich vůbec vyznat přes množství „efektních“ prvků.

Pokud není při vývoji webu přítomný grafik (webdesigner) nebo není vůbec žádná představa o tom, jak mají stránky vypadat, je dobré se podívat na internet po nějakých webových šablonách (freetemplates), což jsou již vytvořené stránky s designem a programováním, které se dají volně stáhnout (free – *zdarma*, template - *šablona*). Jako příklad se mohou uvést stránky www.freewebtemplates.com, kde je tyto šablony možné zdarma stáhnout.

5.1 Databáze

Před tím, než se začne databáze vytvářet, by bylo dobré si někde nakreslit, jak bude vypadat. Samozřejmě pro jednoduché stránky stačí databáze třeba jen s 1 tabulkou o několika položkách, ale pokud se plánuje větší projekt, jako je třeba e-shop, je určitě lepší si databázi někde předem naskicovat. K tomu účelu se hodí program freemind (<http://freemind.sourceforge.net>), který umožňuje jednoduše a rychle navrhovat databáze i se zobrazením různých závislostí.



Obr.8: Program freemind

Pro tvorbu databáze je pak nezbytné mít nainstalovaný jeden z balíčků, který obsahuje potřebné aplikace, o kterých se psalo v teoretické části. Po spuštění phpMyAdmina na lokálním serveru (přes příkaz `http://localhost/adresa_phpMyAdmina`) se zobrazí úvodní obrazovka aplikace ukázaná na obr.6. Protože je phpAdmin spuštěn na lokálním serveru, je možné si databázi vytvořit a pojmenovat sám, na ostrém serveru většinou název databáze určí administrátor nebo poskytovatel služby. Já jsem databázi pojmenoval *projekt*. Když je databáze vytvořena, tak je možné do ní vkládat tabulky a to lze 2 způsoby.

- 1) Vyplněním názvu a počtu sloupečků v okně „Vytvořit novou tabulku v databázi *projekt*“, kde se po odeslání musí ještě určit typ jednotlivých sloupečků. Typ sloupečku určuje, jaké proměnné bude obsahovat. Proměnné jsou většinou stejného typu jako je u ostatních programovacích jazyků, tedy: řetězec (VARCHAR), text (TEXT), celé číslo (INT), desetinné číslo (FLOAT) a další.
- 2) Druhým způsobem je vytvoření tabulky pomocí SQL skriptu. Ten se vkládá do pole označeného „Spustit SQL dotaz(y) na databázi *projekt*“ a odešle. Tímto způsobem lze nejen vytvořit tabulku, ale společně s tím do ní vložit i záznamy. Takto se tabulky nejen vytvářejí, ale i zálohují. Po exportu do SQL vznikne kód, který po opětovném vložení do databáze vytvoří zálohovanou tabulku i s jejím obsahem.

Příklad vytvoření tabulky s definovaným obsahem pro tento projekt:

```
CREATE TABLE IF NOT EXISTS `clanky` (
  `ID` int(2) NOT NULL,
  `nadpis` varchar(200) NOT NULL,
  `text` varchar(2000) NOT NULL,
  `autor` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
INSERT INTO `clanky` (`ID`, `nadpis`, `text`, `autor`) VALUES
(1, 'První článek', 'Tento článek se vytvoří při založení databáze.', 'Jan Dlouhý');
```

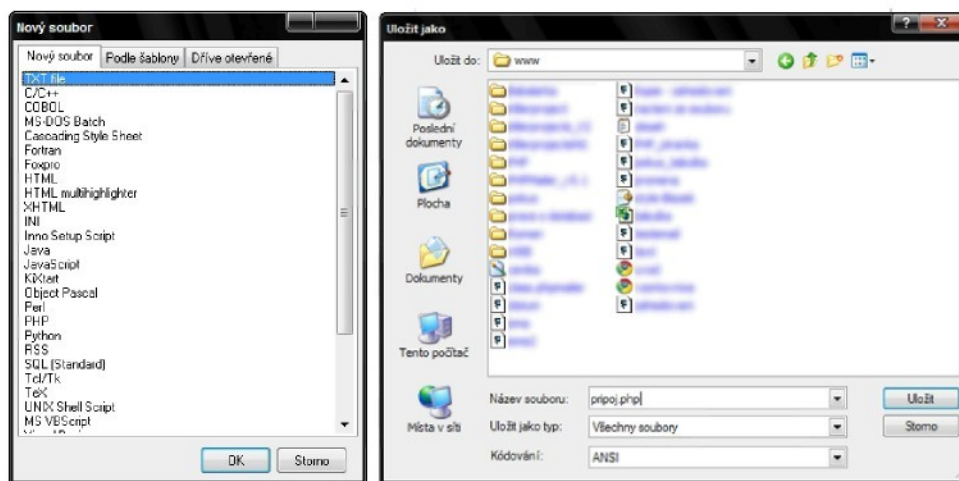
Uvedený skript vytvoří tabulku s názvem „clanky“, která má 4 sloupce (ID, nadpis, text, autor) a je v ní vložen jeden řádek. Struktura tabulky i s jejím obsahem je na obr.9.

Server: localhost		Databáze: projekt		Tabulka: clanky			
Projít	Struktura	SQL	Vyhledávání	Vložit	Export	Import	
	Sloupec	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Extra
☐	ID	int(2)			Ne	Žádná	
☐	nadpis	varchar(200)	utf8_general_ci		Ne	Žádná	
☐	text	varchar(2000)	utf8_general_ci		Ne	Žádná	
☐	autor	varchar(100)	utf8_general_ci		Ano	NULL	
	ID	nadpis	text		autor		
☐	1	První článek	Tento článek se vytvoří při založení databáze.		Jan Dlouhý		

Obr.9: Výpis databáze.

Pro připojení stránek do databáze byl pak vytvořen samostatný soubor, který se jmenuje „pripoj.php“. Díky tomuto souboru mohou stránky komunikovat s databází a vytvořenou tabulkou.

Po otevření kteréhokoliv textového editoru, např.poznámkového bloku windows nebo lépe PsPad, se zvolí typ souboru jako php. U programu PsPad se určuje typ jako první při zakládání nového projektu, u poznámkového bloku až při ukládání, když se soubor pojmenuje jako *pripoj.php*).



Obr.10: Určení typu souboru v PsPad a poznámkovém bloku

Soubor pripoj.php obsahuje následující kód:

```
<?php
$db = mysql_connect("server", "admin", "heslo") or die ("nepovedlo se
pripojit k serveru");
mysql_select_db("projekt",$db) or die ("nepovedlo se pripojit k databazi");
?>
```

Řádek začínající „\$db = mysql_connect.“ říká, k jakému serveru se má stránka připojit. Místo slov *server*, *admin* a *root* se doplní hodnoty, které jsou nastaveny v phpMyAdminovi pro přístup na server. Pokud se nelze spojit se serverem (chyba samotného serveru nebo jsou špatně vyplněny přihlašovací údaje), napíše se varovná zpráva a skript se ukončí.

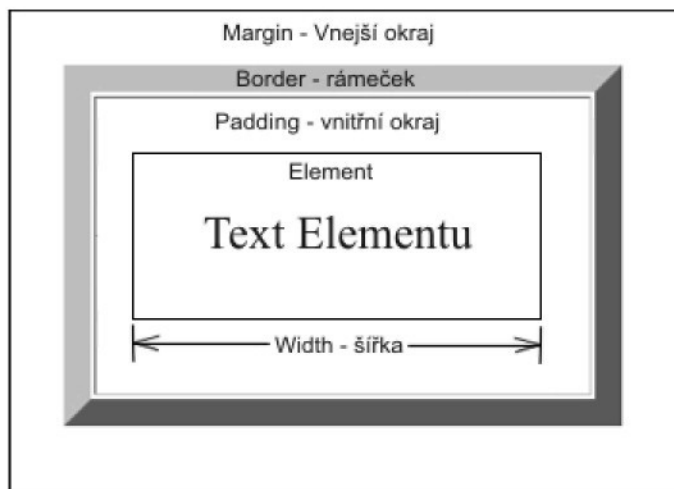
Druhý řádek říká, do které databáze se má připojit, v tomto případě databáze „projekt“. Pokud se opět nepovede spojit s databází, vrátí se negativní odpověď.

5.2 Soubor s CSS styly

Pro použití CSS stylů v tomto příkladě je využita metoda s externím souborem, kde všechny styly jsou uloženy v jednom souboru. Protože výpis celého souboru by byl hodně dlouhý, napíšu jen pár základních tříd. Celý soubor i s projektem je přiložen na CD u této práce.

```
body { margin:0; padding:0; width:100%; background:#778899 ;color: #000000;
font: normal 12px/14px "trebuchet ms", arial, tahoma,verdana, helvetica,
sans-serif;}
```

- „body“ bez tečky na začátku, je selektor s nastavením stylu pro celý dokument.
- `margin` a `padding` určují odsazení elementů od okraje viz.obrázek 11.
- `width` určuje šířku stránky, která se má využít (zde 100%).
- `Background` nastavuje pozadí, nastavena jen barva - zadává se anglický název (red,green,yellow) nebo její šestnáctkový kód ve tvaru #XXXXXX (např. #ff0000, #008000, #ffff00). Samozřejmě se dá za pozadí použít i obrázek, se zápisem `background:url(obrazek.jpg) right no-repeat;` v závorce je pak adresa obrázku (může být absolutní `http://moje_stranky/obrazek.jpg` nebo relativní `../obrazky.jpg`). Slovo `right` znamená zarovnání obrázku doprava nebo doleva a `no-repeat` je neopakovat obrázek (může být také `repeat-x` nebo `repeat-y` pro opakování ve zvolené ose).



Obr.11 Ukázka určení rozměrů na stránce

```
.body h1 { font: bold 26px Arial, Helvetica, sans-serif; color:#000000; }
```

Tento řádek určuje styl pro nadpis `h1`, přesněji druh písma tlusté (`bold`), typ písma a jeho barvu. Protože je ale pod třídou `.body` (s tečkou), na stránce se tento styl zobrazí až po jeho otevření.

Příklad:

```
<h1>Tady ještě styl nebude fungovat</h1>
<div class="body">
  <h1>Tady už styl fungovat bude je totiž už pod rodičovskou třídou</h1>
</div>
```

Stejným způsobem je možné nastavit styl pro odstavce (.body p), odkazy (.body a) a všechny nadpisy (.body h2, h3, h4). Jestliže uvedené rozdělení nestačí, lze vytvořit ještě podtřídy, které budou na hlavní závislé.

Příklad:

V CSS souboru se nastaví jednotlivé styly:

```
.body { font: normal 12px Arial, Helvetica, sans-serif;}
.body .cervene {color:red;}
.tlusty { font: bold 12px Helvetica, Arial, sans-serif; color: #008000;}
```

Na stránce se vyvolají:

```
<div class="body">
  <div class="cervene">
    Tady bude červený text který si vezme druh písma z prvního třídy a
    barvu z druhé
  </div>
</div>
<div class="tlusty">Zde bude zelený tučný text</div>
```

Třída body obsahuje podtřídu „cervene“ která mění barvu písma, ne však jeho velikost nebo styl. Samostatná třída „tlusty“ má určen jak vlastní styl písma, tak jeho velikost a barvu.

```
.patka { height:20px; text-align:center; text-decoration:none;}
.hlavicka {height:225px; background:url(vrsek.jpg) right no-repeat;}
```

Třída patka nastavuje výšku obdélníku na 20px, zarovnává v něm text na střed (je ještě možné doprava, doleva nebo nahoru) a vypíná „dekorativní“ prvky jako jsou podtržení, přeškrtnutí, nadtržení nebo blikání u textu.

Poslední třída „hlavicka“ vytvoří obdélník o výšce 225px, který má na pozadí obrázek zarovnaný pravým okrajem k pravé straně.

5.3 Frontend-index.php

Část pro uživatele je jen jedna stránka, která zobrazuje články z databáze. Celé rozhraní funguje tak, že jsou nejdříve zobrazeny nadpisy jednotlivých článků a po kliknutí na ně se zobrazí celý článek i s jeho autorem. Blokové schéma stránky je na obrázku 12. Jednotlivé části kódu použité v souboru jsou zvýrazněny šedým podkladem.



Obr.12: Blokové schéma souboru index.php

Soubor `index.php` je velmi důležitý pro funkčnost celého webu. Když je do prohlížeče zadána obecná adresa stránky, tak ten jako první začne na serveru vyhledávat právě `index.php`. Obecná adresa znamená, že není zadáný žádný upřesňující parametr. Příklad obecné adresy je `http://muj_projekt.cz`. Jestliže na serveru nebude soubor `index.html` nebo `index.php` nalezen, prohlížeč nahlásí chybu 403 o chybějící úvodní stránce.

Pokud je zadána adresa upřesňující soubor, prohlížeč `index.php` bude ignorovat a zobrazí požadovanou stránku. Příklad adresy upřesňující soubor je `http://muj_projekt.cz/prvni.php`, kde hledaný soubor je `prvni.php`.

5.3.1 HTML část

Každý HTML soubor, který se dnes umísťuje na server, by měl mít uvedenou direktivu, která vypadá takto:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

 (1)

Jedná se o nastavení prohlížeč a říká mu, jak má stránku zobrazit. U ostatních webů to nemusí být přesně tento zápis, ale měl by být hodně podobný (v tomto případě nastavuje prohlížeč na "režim zpětné kompatibility", o kterém se psalo v teoretické části).

```
<head>
```

 (2)

```
<title>Nase stránky</title>
```

 (3)

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
```

 (4)

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

 (5)

```
</head>
```

 (6)

Kód zapsaný mezi párovými značkami `<head></head>` (2) (6) určuje veškerá nastavení hlavičky na stránce jako například: název, který se zobrazí v hlavičce prohlížeče (3) a použitý jazyk (kódování) (4), v tomto případě nastavuje češtinu (windows-1250). Dále lze ještě povolit prohledávání stránek vyhledávacími roboty, napsat klíčová slova, pod kterými vyhledávač najde stránku, autora, copyright a samozřejmě nastavení souboru s kaskádovými styly `<link href="style.css" ...>` (5).

Příklad nastavení pro hlavičku:

```
<link rel="icon" href="nejaka_ikona.ico" type="image/ico" />
<link rel="SHORTCUT ICON" href="nejaka_ikona.ico" />
<meta name="author" content="autor" />
<meta name="keywords" content="klíčová slova,pro vyhledáč" />
<meta name="description" content="popis stránky" />
```

Některé uvedené možnosti nejsou použity, protože pro tento projekt jsou nepotřebné.

Ted' už bude následovat samotný obsah webu:

```
<div class="main">
```

 (7)

```
<div class="hlavicka"></div>
```

 (8)

```
<div class="body">
```

 (9)

Nejdříve třída `main` (7) nastaví velikost stránky, pozadí a odsazení od okraje, následně `hlavicka` (8) vloží do vytvořeného elementu obrázek a třída `body` (9) má pak informace o typech písma, nadpisů a odsazení (`hlavicka` a `body` jsou již „dětmi“ třídy `main`).

```
<h1>Nějaký nadpis</h1>
```

 (10)

Párová značka `<h1></h1>` (10) vloží nadpis s textem „Nějaký nadpis“.

5.3.2 PHP část

```
<?php (11)
include("pripoj.php"); (12)
```

(11) Velmi důležitá část kódu, která říká, že následující skripty budou v PHP, doposud jsou pouze v HTML. Řádek (12) načítá soubor *pripoj.php*, který obsahuje data o připojení do databáze.

```
if(isset($_GET['ukaz'])){$ukaz=$_GET['ukaz'];}else{$ukaz=NULL;} (13)
```

Tento skript určuje proměnnou `$ukaz` (proměnné se zapisují se znakem “\$”) (13). Pokud je přijata proměnná `$_GET['ukaz']`, pak se `$ukaz` nastaví podle ní, jinak je `$ukaz` rovna nule.

a) Proměnné typu `$_GET`

Tento typ je velmi důležitý, protože si pomocí něj posílají stránky mezi sebou hodnoty proměnných. Pro tuto komunikaci se využívá odkazu doplněný znakem „?” nebo pokud je třeba poslat více proměnných tak „&”.

Příklad: `<a href="nejaka.php?pr1=1&pr2=ahoj">`

Tento odkaz odešle na stránku *nejaka.php* proměnné `pr1` s hodnotou `1` a `pr2` s řetězcem „ahoj”. Na cílové stránce „nejaka.php” se pro lepší práci s touto proměnnou upraví tvarem `if(isset($_GET['pr1'])){$pr1=$_GET['pr1']}` do podoby `$pr1=1`.

Tento způsob posílání proměnných ale není moc bezpečný, protože kdokoli může v adresovém řádku přepsat zadanou adresu a změnit tak hodnotu proměnné. Proto se nedoporučuje odesílat tímto způsobem nějaká důležitá data, jako například jméno, heslo apod. Pro odeslání proměnných z jedné stránky na druhou, které nepotřebují větší zabezpečí, se ale hodí skvěle.

Pokud je nutné odeslat data mezi stránkami bezpečnou cestou, je lepší použít proměnné typu `$_POST['nazev']` a `$_SESSION['nazev']`.

b) Proměnné typu `$_POST`

Tento druh proměnných používají pro komunikaci mezi stránkami formuláře. Výhodou této komunikace je, že nikdo nevidí, jak se proměnná jmenuje a jakou má hodnotu, takže do procesu nemůže nijak zasáhnout.

Formulář je určen řádkem `<form method="post" action="cílová_stránka">` a pod něj se zapisují všechny proměnné, které se mají odeslat. Ty se pak určují buď vyplněním textového pole, zatrhnutím nebo přepnutím položky apod. Formulář se nakonec odesílá pomocí tlačítka `<input type="submit" name="nazev" value="text tlačítka">`

c) Proměnné typu `$_SESSION`

Další speciální proměnnou jsou proměnné typu `$_SESSION`. Nejdříve je nutné používání těchto proměnných na serveru povolit pomocí zápisu `session_start()`. Posléze už je možné začít tyto proměnné používat. Bez spuštění nebudou fungovat! Specialitou tohoto typu je, že proměnné `SESSION` si server zapamatuje (dokud je prohlížeč otevřený nebo podle nastavené doby), čehož se využívá např. při přihlašování adminů, uživatelů, nastavení položek na stránce apod. Tento typ je zcela bezpečný, protože nikdo nevidí, jak se proměnná jmenuje a jakou má hodnotu.

d) Funkce echo a printf

```
echo "Toto je text napsaný pomocí PHP a funkce echo"; (14)
printf('<h2>Výpis článků</h2>'); (15)
```

Text se v PHP vypisuje na stránku jiným způsobem než v HTML. V něm stačí napsat pouze samotný text bez jakéhokoliv parametru a ten se vypíše. U PHP funguje vypisování textu trochu jinak a to za pomoci funkcí echo nebo printf. V podstatě mají tyto značky stejnou funkci, ale rozdíly mezi nimi jsou. Například echo se používá pokud chceme v PHP vypsat jednoduchý text nebo základní HTML značky:

```
echo "<h1>Nadpis</h1>";
echo "<p><b>text v odstavci a tučně</b></p>";
což bude provedeno jako by to byl obyčejný HTML kód.
```

```
echo "$promenna";
```

Tímto způsobem se vypisuje obsah proměnné.

Pozor, pokud se napíše echo '\$promenna' bude vypsán přesný text (název proměnné) mezi uvozovkami. Takže **ne** obsah proměnné ale text „\$promenna“

Z tohoto důvodu je zřejmé, že echo nelze použít pro odkazy a kusy kódu, kde se používají dvojité uvozovky "".

```
echo "<a tref='nejaka.html'>Odkaz</a>"; -NELZE
echo "<div class='odsad'>Nějaký text</div>"; -NELZE
echo '<a tref="nejaka.html">Odkaz</a> '; -NELZE
```

Pro tyto případy je tu funkce printf. Pomocí ní je již možné provést všechny HTML kódy a mnohem víc. Má to ovšem opět svoje pravidla, která se musí dodržovat. Stejně jako u echo se některé jednodušší znaky mohou zapsat pomocí dvojitých uvozek "".

```
printf("<h1>Toto bude nadpis</h1>");
printf("<p><b>Toto bude tučný text</b></p>");
printf("$nejaka_promena");
```

Opět by ale vyvstal problém, kdyby byl potřeba napsat odkaz. Pro tento případ se funkce printf píše s jednoduchými uvozovkami '. Pozor ale při zápisu proměnných, ty by takto nefungovaly a byly by interpretovány opět jako text. Proměnné se zapisují až za uvozovky, jsou odděleny čárkou a v kódu je zastupuje řetězec %s. Proměnné pak jsou zapisovány ve stejném pořadí jako jsou seřazeni jejich zástupci.

Příklad:

```
printf('%s %s ', $promena1 $promena2);
printf('<a href="nejaka_stranka.php?prl=%s">odkaz</a>', $prl);
```

5.3.3 Výpis z databáze

```
$result = mysql_query("SELECT * FROM clanky"); (16)
```

```
$myrow = mysql_fetch_array($result); (17)
```

Proměnná `$result` (16) určuje tabulku, ze které se budou vybírat záznamy. Tento výběr se dá zúžit pomocí upřesňujících parametrů.

Příklady:

```
$result = mysql_query("SELECT * FROM clanky WHERE jmeno='$jmeno'");
```

vyhledá v tabulce `clanky` řádek, který ve sloupečku `jmeno` má uložený záznam s `$jmeno`. V tomto případě může existovat více záznamů.

```
$vyhledej = mysql_query("SELECT * FROM clanky WHERE ID='1'");
```

Tento zápis již přesně specifikuje hledaný řádek a to s `ID=1`. Proto je vždy nezbytné určit v tabulce sloupeček, který bude mít každý řádek jedinečný a podle kterého půjde vyhledat. Řádek (17) ukládá výsledek hledání do pole `$myrow`.

```
do{ (18)
```

(18) Začátek cyklu

```
printf('<a href="?ukaz=%s"><h3>%s</h3></a>', $myrow['ID'], $myrow['nadpis']); (19)
```

(19) Vypíše obsah sloupečku `'nadpis'` a to formou odkazu směřující na stejnou stránku, na které se nachází. Po kliknutí na odkaz je nastavena proměnná `ukaz` s hodnotou `ID` řádku, který určuje spuštěný cyklus. Pokud tedy není u odkazu uvedena žádná adresa, odkazuje vždy na stejnou stránku, kde je umístěn. Protože se nachází na stránce `index.php`, jde o zkrácený zápis adresy `"index.php?ukaz=%s"`.

```
if($myrow[4]==NULL){$autor="neznámí";}else{$autor=$myrow[4];} (20)
```

Když se sloupeček 3 (sloupečky lze určovat jak jménem tak jejich pořadím. První má číslo 0, druhý 1, třetí 2...) rovná `NULL` (je prázdný), tak `$autor` obsahuje řetězec „neznámý“ v ostatních případech `$autor` je rovna sloupečku 3.

```
if($ukaz==$myrow['ID']){printf('<p>%s</p><p>Autor: %s</p>', $myrow[2], $autor);} (21)
```

Jestliže se `$ukaz` rovná sloupečku `ID`, tak se vypíše sloupeček 2 a proměnná `autor`. Tento řádek zajišťuje, že po kliknutí na nadpis se zobrazí celý článek i s autorem.

```
}while ($myrow = mysql_fetch_array($result)); (22)
```

Ukončení cyklu z řádku (18) – cyklus bude prováděn, dokud budou v databázi zápisy.

```
?> (23)
```

```
</div> (24)
```

```
</div> (25)
```

```
<div class="patka"><a href="administrace.php">Administrace</a></div> (26)
```

```
</body> (27)
```

```
</html> (28)
```

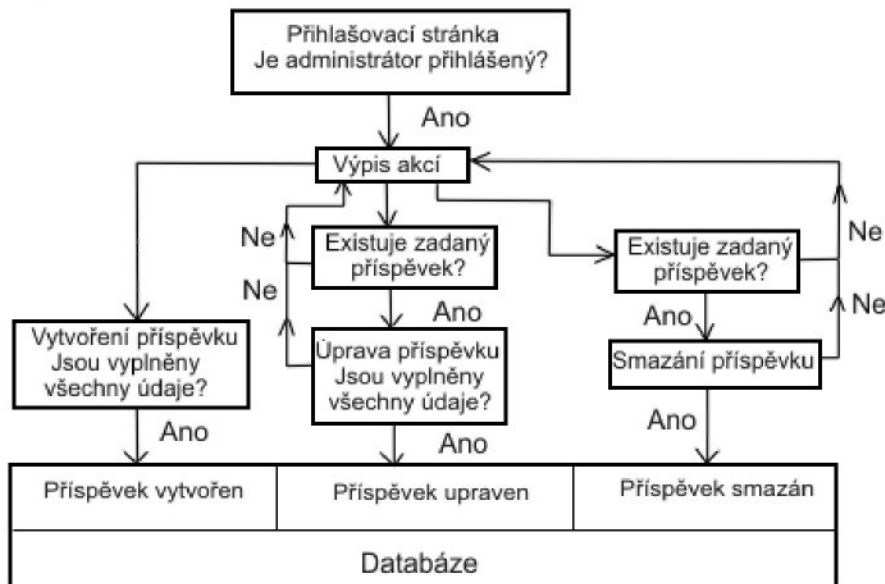
Nyní zbývá uzavřít PHP skripty (23), všechny CSS třídy (24) (25) a HTML (27). Na řádku (26) je odkaz do administrátorského režimu.

5.4 Backend – administrace.php

Administrační část obsahuje jediný soubor, který se jmenuje *administrace.php*. Na úvodní stránce je třeba vyplnit jméno a heslo. Pro zjednodušení je jméno a heslo pevně dané v souboru. V praxi se používají databáze a zadané údaje se kontrolují s databází. Princip je ale u obou postupů stejný. Pokud je jméno a heslo odesláno, začne se porovnávat s řetězcem v databázi. Jestliže se shodují, je nastavena proměnná `$_SESSION['admin']`, která určuje administrátorská práva a umožní zobrazit jinak skrytý obsah stránky.

Pokud je administrátor přihlášen, načte se stránka se seznamem článků, možností přidat, editovat a mazat články.

Kliknutí na odkaz přidat nebo upravit příspěvek se zobrazí nová stránka s formulářem pro vyplnění nadpisu, textu a autora článku. Po jeho odeslání bude v databázi vytvořen záznam. Při mazání článků se web nejdříve zeptá, jestli se má článek smazat, a pokud je potvrzeno, že ano, bude z databáze odstraněn. Poslední věc na stránce je odkaz pro odhlášení, který zruší proměnnou `$_SESSION['admin']`.



Obr.13: Blokové schéma administrace.php

5.4.1 HTML část

Základní html kód je stejný jako v části pro návštěvníky.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> (29)
<head> (30)
  <title>Nase stránky</title> (31)
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1250"> (32)
  <link href="style.css" rel="stylesheet" type="text/css"> (33)
</head> (34)
<body> (35)
<div class="main"> (36)
<div class="hlavicka"></div> (37)
<div class="body"> (38)
```

5.4.2 PHP část

Znovu se začíná otvíracím znakem pro PHP (39) .

```
<?php (39)
session_start(); (40)
include("pripoj.php"); (41)
```

Program začne používat proměnné SESSION (40) a otevře souboru s informacemi pro připojení do databáze (41) .

```
if(isset($_GET['ukaz'])){$ukaz=$_GET['ukaz'];}else{$ukaz=NULL;} (42)
```

```
if(isset($_GET['logout']) && $_GET['logout']=="true")
{unset($_SESSION['admin']);} (43)
```

Řádek (43) odhlásí admina, smaže proměnnou \$_SESSION['admin'], pouze pokud byla nastavena \$_GET['logout'], tedy kliklo se na odkaz *odhlásit se*.

```
if(isset($_GET['uprav']) && $_GET['uprav']!=NULL){$uprav=$_GET['uprav'];}
if(isset($_GET['smaz']) && $_GET['smaz']!=NULL){$smaz=$_GET['smaz'];}
if(isset($_POST['submit'])){ $submit=$_POST['submit'];}else{$submit=NULL;}
if(isset($_POST['submit2'])){ $submit2=$_POST['submit2'];}else{$submit2=NULL;}
if(isset($_POST['jmeno'])){ $jmeno=$_POST['jmeno'];}else{$jmeno=NULL;}
if(isset($_POST['heslo'])){ $heslo=$_POST['heslo'];}else{$heslo=NULL;}
if(isset($_POST['nadpis'])){ $nadpis=$_POST['nadpis'];}else{$nadpis=NULL;}
if(isset($_POST['text'])){ $text=$_POST['text'];}else{$text=NULL;}
if(isset($_POST['autor'])){ $autor=$_POST['autor'];}else{$autor=NULL;} (44)
```

Nastavení všech použitých proměnných (44).

```
if($submit){ (45)
    if($_POST['jmeno']=="jmeno"){ (46)
        if($_POST['heslo']=="heslo"){ (47)
            $_SESSION['admin']=1; (48)
        }else{echo"Bylo zadáno špatné heslo";} (49)
        }else{echo"Bylo zadáno špatné jméno";} (50)
    } (51)
```

Jestliže je odeslán přihlašovací formulář pomocí tlačítka \$submit(45), provede se zadáný skript (46) - (50). Porovnáním zadaného jména a hesla s databází se nastaví (nebo nikoliv) \$_SESSION['admin'] .

```
if(isset($_SESSION['admin']) && $_SESSION['admin']=="1"){ ?> (52)
```

Pokud není nastavena proměnná \$_SESSION['admin'] nebo se nerovná 1, je programem následující část kódu vynechána. Řádek (52) zajišťuje, že bez správně vyplněného jména a hesla nebudou ostatní stránky kromě přihlašovací viditelné.

```
<H1>Administrace</H1> (53)
<p><a href="administrace.php">Domů</a> (54)
    <a class="admin" href="?pridej">Vytvořit nový článek</a> (55)
    <a class="admin" href="administrace.php?logout=true">Odhlásit se</a> (56)
</p> (57)
<?php (58)
```

Krátký html kód, který vytvoří nabídku s odkazy na hlavní stránku, pro vytvoření článku a pro odhlášení. I když bylo ukončeno PHP kódování (52), je tato i následující část kódu nepřihlášeným uživatelům skryta. Podmínka s řádku (52) stále platí.

a) Vkládání příspěvků

Následující kód se zobrazí pouze při zakládání nového příspěvku, což určuje řádek s podmínkou (59). Řádek (60) vymezuje část, která se provede až po odeslání formuláře.

```
if(isset($_GET['pridej'])) { (59)
    if($submit2) { (60)
        for($i=1;$i<=399;$i++): (61)
            $nove_id = mysql_query("SELECT * FROM clanky WHERE ID=$i"/*,$db*/); (62)
            $radek = mysql_fetch_array($nove_id); (63)
            if ($radek["ID"]==NULL){break;} (64)
        endfor; (65)
```

Pro určení nového ID byl napsán tento skript (61)–(65). Prohledává od začátku databázi a pokud najde chybějící řádek, tak hledání ukončí a tím zvolí nové ID.

```
if($nadpis==NULL or $text==NULL){printf('<div class="cervene">Článek nebyl
vytvořen chybí některé údaje</div>');} (66)
else{ (67)
```

Jestli jsou odeslané proměnné \$nadpis nebo \$text nulové, napíše se varovná zpráva (66), jinak skript pokračuje dál (67).

```
$vloz = "INSERT INTO clanky (ID,nadpis,text,autor) VALUES
        ('$i','$nadpis','$text','$autor')"; (68)
$result = mysql_query($vloz); (69)
printf('<div class="zelene">Článek byl vytvořen</div>');}} ?> (70)
```

Když jsou splněny všechny předchozí podmínky, data se vloží do databáze. Proměnná \$vloz (68) určuje, která data se mají do jakého sloupce vložit, proměnná \$result (69) vykoná samotné vložení. Pro ověření se vypíše zpráva (70) a aby byla 100% jistota o vytvoření záznamu, je možné si nechat vložený řádek z databáze vypsát. Tento skript končí uzavřením podmínek submit2 (60), else (67) a PHP skriptů. Nyní zbývá vytvořit formulář.

```
<div class="odsad"> (71)
<table> (72)
```

Funkce pro začátek tabulky

```
<form method="post" action="<?php $PHP_SELF ?>"> (73)
```

Nastavení formuláře, metody odeslání dat POST a místo, kam se data odešlou.

```
<tr><td colspan="2"><H4>Přidání příspěvku</h4></td></tr> (74)
```

Každý řádek tabulky se zapisuje mezi párovou značku <tr>, sloupcečky mezi <td>, pokud se spojují 2 sloupcečky vodorovně, tak pomocí zápisu <td colspan="2">, vertikálně je to <td rowspan="2">.

<td colspan="2">		
<td rowspan="2">		

Obr.14: Rozvržení tabulky

```
<tr><td>Nadpis:</td><td><input type="text" size="75" name="nadpis"
value="<?php echo"$nadpis"; ?>"></tr> (75)
```

Vytvoří textové pole o 1 řádku a určené velikosti. Po odeslání je jeho obsah uložen do proměnné \$_POST['nadpis'].

```
<tr><td>Text:</td><td><textarea name="text" cols="54" rows="3"><?php
echo"$text"; ?></textarea></tr>
```

 (76)

Tato část vkládá do stránky textové pole o více řádcích a jeho obsah uloží do proměnné \$_POST['nadpis'].

```
<tr><td>Autor:</td><td><input type="text" size="50" name="autor"
value="<?php echo"$autor"; ?>"></tr>
```

 (77)

```
<tr><td colspan="2">
```

 (78)

```
<input type="submit" name="submit2" value="vytvořit"></td></tr>
```

 (79)

Tlačítko pojmenované submit2 (79) všechny vyplněná data odešle pro zkontrolování na řádky (60) – (67).

```
</table></div> <?php }
```

 (80)

Uzavření tabulky i části pro zakládání příspěvků.

b) Úprava příspěvků

Další kód platí pouze při splnění podmínky pro úpravu.

```
if(isset($uprav)){
```

 (81)

```
if($submit2){
```

 (82)

```
if($nadpis==NULL or $text==NULL){printf('<div class="cervene">Článek
nebyl změněn.Chyběly některé údaje</div>');}
```

 (83)

```
else{
```

 (84)

Vysvětleno v části zakládání příspěvků.

```
$uprav = "UPDATE clanky SET nadpis='$nadpis',text='$text',autor='$autor'
WHERE ID='$uprav'";
```

 (85)

```
$result = mysql_query($uprav);
```

 (86)

```
printf('<div class="zelene">Článek byl upraven</div>');}}
```

 (87)

Proměnná \$uprav (85) nastavuje, jaká tabulka se má v databázi upravit, jaké sloupcečky a jaký řádek. Pokud by zadaným parametrům odpovídalo více řádků, upraví se všechny shodné. Následně je napsána ověřující zpráva a uzavřena podmínka else a submit2 (87).

```
$result1 = mysql_query("SELECT * FROM clanky WHERE ID='$uprav'");
```

 (88)

```
$myrow1 = mysql_fetch_array($result1);
```

 (89)

```
$nadpis=$myrow1['nadpis'];
```

 (90)

```
$text=$myrow1['text'];
```

 (91)

```
$autor=$myrow1['autor'];
```

 (92)

```
?>
```

 (93)

Protože se upravují data z databáze, je nezbytné si je nejdříve zobrazit. O to se postará část kódu (88) – (93), která otevře tabulku z databáze na určeném řádku (88) (pokud se jich shoduje více, vybere první) a za jednotlivé proměnné se uloží obsahy daných sloupceček (90) – (92).

```
<div class="odsad"><table>
<form method="post" action="<?php $PHP_SELF ?>">
<tr><td colspan="2"><H4>Úprava příspěvku</h4></td></tr>
<tr><td>Nadpis:</td><td><input type="text" size="75" name="nadpis" value="<?php
echo"$nadpis"; ?>"></tr>
<tr><td>Text:</td><td><textarea name="text" cols="54" rows="3"><?php echo"$text";
?></textarea></tr>
<tr><td>Autor:</td><td><input type="text" size="50" name="autor" value="<?php
echo"$autor"; ?>"></tr>
<tr><td colspan="2"><input type="submit" name="submit2" value="upravit"></td></tr>
</table></div> <?php
}
```

(94)

Poslední část je shodná jako při zakládání příspěvek, jen s tím rozdílem, že tady se zobrazí v textových polích položky z databáze.

c) Smazání příspěvků

Další část je v základu stejná jako předchozí dvě. Nejdříve se určí podmínka, za které má být obsah stránky zobrazený (\$smaz) (95) a co se stane, když je odsouhlaseno smazání příspěvku \$_GET['ano'] (96).

```
if(isset($smaz)){ (95)
if(isset($_GET['ano'])) { (96)

    $sql = "DELETE FROM clanky WHERE ID='$smaz'"; (97)
    $result = mysql_query($sql); (98)
    printf('<div class="cervene">Článek byl smazán</div>'); (99)
} else{
```

Tato část smaže z databáze určený řádek a napíše potvrzující zprávu.

```
$result2 = mysql_query("SELECT * FROM clanky WHERE ID='$smaz'"); (100)
$myrow2 = mysql_fetch_array($result2); (101)
$nadpis=$myrow2['nadpis']; (102)
?> (103)
<h3>Smazání článku</h3> (104)
<p>Chcete opravdu smazat Článek <?php echo"$nadpis?</p>"; (105)
printf('<p><a class="admin"
href="administrace.php?smaz=%s&ano">ANO</a>', $smaz);?> (106)
<a class="admin" href="administrace.php">NE</a></p> (107)
<?php (108)
}} (109)
```

Tento výřez kódu je v podstatě jen obslužný a se samotnou databází nic nedělá. Pokud se klikne na odkaz pro smazání příspěvku, zobrazí se tato stránka, která má za úkol zeptat se na smazání článku. Řádky (100)–(102) slouží k nalezení nadpisu v databázi. Druhá část nadpis vypíše a zadá možnosti smazání ANO či NE. ANO určí proměnnou \$_GET['ano'] a dojde ke smazání článku, NE odkazuje zpět na hlavní stránku.

d) Výpis článků

```
printf('<h2>Výpis článků</h2>'); (110)
$result = mysql_query("SELECT * FROM clanky"); (111)
$myrow = mysql_fetch_array($result); (112)
do{ (113)
    if($myrow[4]==NULL){ $autor="neznámí"; } else{ $autor=$myrow[4]; } (114)
    printf('<h3>%s</h3>
        <p>%s<br><i>Autor: %s</i></p>', $myrow['nadpis'], $myrow[2], $autor); (116)

    printf('<a class="admin" href="?uprav=%s">Upravit článek</a>
        <a class="admin" href="?smaz=%s">Smazat
článek</a>', $myrow['ID'], $myrow[0]); (117)

}while ($myrow = mysql_fetch_array($result)); (118)
echo"<br><br>"; (119)
} (120)
```

Výpis článků je stejný jako části pro návštěvníky. Jediný rozdíl je, že pod každým článkem přibyl odkaz pro úpravu a smazání příspěvku. Celý kód uzavře podmínka z řádku (52). Pokud tato podmínka neplatí, provede se zbylý kus programu.

```

else{
<center>
<table>
<form method="post" action="<?php $PHP_SELF ?>">
<tr><td colspan="2"><H4>Přihlášení</h4></td></tr>
<tr><td>Jméno:</td><td><input type="text" size="25" name="jmeno"
value="<?php echo"$jmeno"; ?>"></tr>
<tr><td>Heslo:</td><td><input type="password" size="25" name="heslo"></tr>
<tr><td colspan="2"><input type="submit" name="submit" value="Přihlásit
se"></td></tr>
</table>
</center>
<?php
}
?>

```

(121)

Předposlední část je přihlašovací formulář, který se zobrazí pouze pokud není administrátor přihlášen. (121).

```

</div>
</div><div class="patka"><a href="index.php">Moje stránky</a></div>
</body>
</html>

```

(122)

Celý program je ukončen stejně jako část pro návštěvníky. Uzavře se PHP, HTML a vloží se odkaz na stránky pro návštěvníky.

Tímto je uvedený projekt hotov. Když se zvýrazněný kód vloží do určených souborů a uloží, tak by měly stránky fungovat. Výsledná podoba obou částí (jak pro návštěvníky, tak administrátory) je na obr.15 a 16.



Obr.15: Stránky pro návštěvníky



Obr.16: Administrátorský režim

6 STRÁNKY KONFERENCE

Shodně jako projekt, který byl popsán v minulé kapitole, jsou i stránky konference rozděleny na část pro návštěvníky (účastníky akcí) a část pro administrátory. Vypisovat kód samotného webu zde nebudu, protože by to bylo moc zdlouhavé a v základu funguje na stejných principech. Popíšu jen, co má jednotlivá stránka za úkol. Soubory samotných stránek jsou přiloženy na CD.

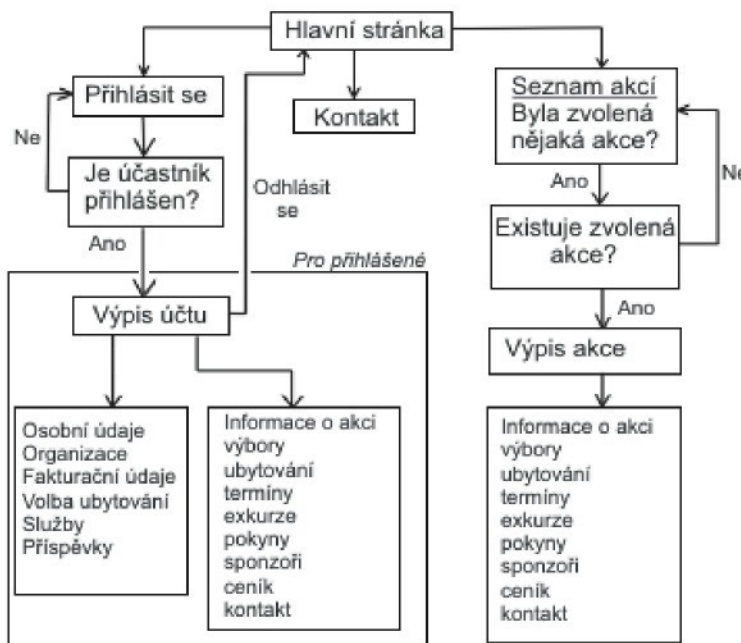
Ještě bych chtěl zmínit, že design stránek vychází ze stažené šablony stránky (templeat), který byl ale hodně upraven pro potřeby těchto stránek.

6.1 Frontend

Návštěvníci stránek si v této části mohou prohlédnout obecné informace o pořádaných akcích a díky přiložené přihlášce se na ni i přihlásit. Pomocí emailu, který po registraci účastník akce obdrží, si aktivuje účet a posléze již může upravovat svůj účet a odesílat příspěvky na zvolenou akci.

Struktura webu – Část pro uživatele

Struktura webu (část pro uživatele) vypadá asi takto:



Obr.17: Blokové schéma části pro účastníky

a) Hlavní stránka – index.php

Slouží jako rozcestník mezi všemi funkcemi na webu. Nachází se na ní seznam 3 nejdříve pořádaných akcí, seřazených podle data konání. Pokud je některá akce vybrána, zobrazí se její výpis spolu s nabídkou odkazů pro další informace o akci. Na hlavní stránce je mimo jiné okno pro přihlášení do systému, kam se vyplňuje email a heslo. Pokud jsou tyto údaje odeslány na server, porovnají se s databází a jestli se najdou shodné řetězce, vytvoří se proměnná `$_SESSION['ID']`, která určuje pořadové číslo účastníka v databázi a přihlásí ho tak do systému. Díky tomu si účastník může prohlížet svůj účet stejně jako ho upravovat a odesílat příspěvky na akce.

b) Stránka pro přihlášení – prihlas.php

Na této stránce je možné se přihlásit do systému, stejně jako na úvodní stránce. Opět po odeslání dat a jejich ověření je vytvořena proměnná s pořadovým číslem účastníka, která umožňuje přístup k jeho účtu. Protože jsou hesla v databázi zašifrovaná, musí se kontrolované (odeslané) heslo nejdříve zašifrovat a pak teprve porovnat s databází.

c) Výpis akcí– akce.php

Jestliže není vybrána žádná akce, tak se vypisují informace o třech nejdříve pořádaných akcích a v levém menu je seznam všech pořádaných akcí seřazených podle kateder.

Pokud je některá akce vybrána (kliknutím na název akce v seznamu), vytvoří se proměnná `$_GET['detail']`. Po vytvoření této proměnné je zobrazen výpis zvolené akce. V menu vlevo jsou odkazy pro další informace o akci.

d) Výbory - vybory.php

Tato stránka slouží k výpisu výborů u zvolené akce. Jde pouze o výpis textu z databáze.

e) Termíny - termíny.php

Tato stránka slouží k výpisu termínů u zvolené akce. Jde pouze o výpis textu z databáze.

f) Pokyny pro Autory - pokyny.php

Tato stránka slouží k výpisu pokynů u zvolené akce. Pokyny jsou buď formou textu nebo jako odkaz na soubor, který je možné si stáhnout. Tento soubor pak může obsahovat doplňkové informace jako šablony, vzory apod.

g) Kontakt – kontakt.php

Pokud je vybraná některá akce nebo když je účastník přihlášen, zobrazí se kontakt na pořadatele zvolené akce. V ostatních případech se vypíše kontakt na správce stránek.

h) Registrace – registrace.php

Pro registraci do systému slouží tato stránka. Je na ní umístěno 5 textových polí, do kterých se doplňuje jméno, příjmení, email, heslo a ověření hesla. Aby byla registrace vložena do databáze, musí platit několik podmínek:

- jméno nesmí být prázdné
- příjmení nesmí být prázdné
- email nesmí být prázdný a nesmí se shodovat s některým z emailů v databázi. Toto je jeden z unikátních prvků, podle kterých se při přihlašování určuje uživatel. Jméno a příjmení mohou mít 2 lidé stejné ale email nikoliv.
- Hesla nesmí být prázdná a musí se shodovat. Hesla jsou pak v databázi ještě šifrovaná, což znemožňuje jejich zneužití (zloději by byla k ničemu). Když uživatel heslo zapomene, musí se mu heslo změnit a zaslat nové.

Po zaregistrování je na uvedený email zaslána zpráva s odkazem na ověřující stránku. Pokud účastník na tento odkaz neklikne, účet mu nebude fungovat. Vždy když se přihlásí, tak se mu zobrazí zpráva o tom, že účet nebyl aktivován. Pokud náhodou email nedostal nebo ho smazal, je tu možnost zaslat ho znovu.

Pokud je již návštěvník v systému zaregistrován a chtěl by se zúčastnit více akcí, tak pro tento případ je tu odkaz na spodku stránky, který směřuje na formulář, do kterého se vyplní email a odešle. Jestliže je zadán email v databázi, bude uživatel zaregistrován i na tuto akci.

i) Sponzoři – sponzori.php

Na této stránce jsou vypsáni všichni sponzoři, kteří patří k vybrané akci. Jsou zobrazeny jejich loga s odkazem na domovské stránky.

j) Informace o účtu účastníka – account.php

Na této stránce jsou zobrazena všechna data, která účastník akce vyplnil, a upozornění u položek, které ještě chybí doplnit. Je to tedy výpis osobních údajů, fakturačních údajů, zvoleného ubytování, výpis všech služeb s cenami a výslednou cenou, kterou má zaplatit. Seznam příspěvků slouží i jako odkaz na jednotlivé příspěvky. Ve výpisu služeb je ještě odkaz pro vytisknutí celého účtu do PDF.

k) Osobní údaje – update.php

Zde se doplňují osobní údaje účastníka. Je to formulář o 8 položkách, do kterých se doplňuje titul před a za jménem, jméno, příjmení, ulice, město, PSČ a stát. Pro správné odeslání do databáze je zde podmínka, že PSČ musí být číslo o 5-ti nebo 6-ti cifrách. Pokud je PSČ ve správném tvaru, tak se všechny vyplněné údaje vloží do databáze.

Pro změnu hesla jsou tu 3 pole, do kterých se zadává staré heslo, nové heslo a potvrzení nového hesla. Když se staré heslo shoduje se záznamem v databázi (opět se musí nejdříve zašifrovat a pak porovnat) a nová hesla se shodují, tak se záznam odešle a heslo se upraví.

l) Informace o organizaci – update.php

Do 6 textových polí je třeba doplnit název organizace, ulici, město, PSČ, stát a telefon. Podmínky pro odeslání jsou takové, že PSČ musí být číslo o 5-ti nebo 6-ti cifrách a telefon musí být ve formátu ('+YYYYXXXXXXXX' nebo 'XXXXXXXX'). Zbylé údaje mohou být jakýkoliv řetězec.

m) Fakturační údaje – update.php

Ve fakturačních údajích jsou stejná pole jako u předchozího případu (platí stejné podmínky), doplněné o položky IČO a DIČO. Na stránce je také tlačítko pro vložení dat s „informacemi o organizaci“, které zkopíruje data z databáze a doplní je do příslušných řádků.

n) Volba ubytování – update.php

Ubytování se vybírá z položkového menu, ve kterém je vypsán název, cena za noc, jestli se jedná o samotný pokoj či nikoliv a zda-li bude cena za ubytování započtena do celkového součtu. Všechny tyto možnosti nastavuje administrátor a uživatel si pak jen volí ze seznamu nabízených ubytování. Uživatel si určuje od kdy do kdy se hodlá ubytovat (tím se vypočítá výsledná cena za ubytování) a s kým se chce ubytovat, což se volí z výpisu účastníků přihlášených na akce.

Podmínek pro odeslání je tu trochu víc a to pokud má zvolené ubytování určeno, že jde o samostatný pokoj, pak se spolubydlící automaticky nuluje (není povolen). Pokud je určeno, že se cena za pokoj nezapočítává do celkové ceny, je o tom proveden záznam v příslušném sloupečku databáze a ve výsledném součtu nebude k ceně přičtena. Poslední podmínka určuje, že datum konce ubytování nesmí být dříve než jeho začátek.

o) Druh účasti a služby – sluzby.php

Na této stránce se vybírá cena vložného a to ze seznamu možností účasti na akci. Volí se zde také doplňkové služby, které si účastník akce přeje. Jedná se tak o exkurze, exponáty, různé druhy publikací apod.

p) Výpis/nahrání příspěvků – prispavek.php

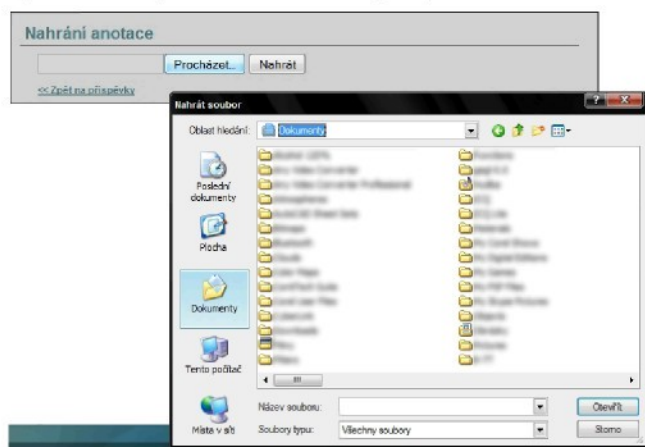
Pomocí této stránky posílá autor příspěvky na akci. Po kliknutí na „vytvořit příspěvek“ se zobrazí formulář s kolonkami, do kterých se doplňují informace o akci:

- Určení typu příspěvku (přednáška, vývěska, publikace bez účasti).
- Název akce.
- Název akce anglicky.
- Text anotace.
- Text anotace anglicky.
- Vybírají se autoři příspěvku ze seznamu přihlášených.
- Určuje se, zda bude příspěvek přednesen v angličtině.

Od každého typu je pak možné nahrát pouze 3 příspěvky. Po vytvoření příspěvku se objeví v seznamu jeho výpis. Pod tímto výpisem jsou možnosti pro upravení/smazání příspěvku, nahrání/smazání souboru s anotací, nahrání/smazání souboru s textem příspěvku. Nahrání souboru na server se provádí pomocí funkce k tomu určené, která využívá FTP protokol. Do databáze se pak ještě ukládá cesta k souboru.

r) Nahrání souboru s anotací, textem – upd_prispevek.php

Pro odeslání souboru na server se používá stránka upd_prispevek.php. Nejdříve je však potřeba soubor najít (provádí se pomocí textového pole).



Obr.18: Výběr souboru který se má nahrát na server

Když je soubor vybrán, uloží se do první proměnné cesty k jeho umístění (tedy zdroje), poté se určí také místo, kam se má nahrát. Tato složka se nachází na adrese *prispvky/ID katedry/ID akce/emailu autora/nové jméno souboru.kon*. Aby nedošlo k tomu, že by se nahrály 2 stejně pojmenované soubory a ty by se navzájem přepsaly, je ještě před odesláním soubor přejmenován a to na *typ.email.ID autora.koncovka*. Výsledná cesta pak může vypadat např. *prispvky/1/5/text.email@email.cz.6.doc*. Tato varianta byla zvolena z toho důvodu, aby nemohlo dojít ke zmiňované duplicitě a také že soubory na internetu nesmí obsahovat háčky ani čárky.

Když je určen jak zdroj, tak cíl, je zapotřebí vytvořit samotnou složku. Nejdříve se zjistí, jestli složka existuje, pokud ne, tak se pomocí funkce `mkdir($cil, 0777 );` tato složka vytvoří a nastaví se jí práva, aby do ní bylo možné soubory odesílat (0777 – oprávnění pro všechny návštěvníky, že mohou číst, zapisovat a spouštět soubory).

Odeslání souboru se provede díky následujícímu kódu:

```
$nazev_souboru = $_FILES['fupload']['tmp_name']; // Zjistí cestu k souboru
```

```
$copy = move_uploaded_file($nazev_souboru, $cil) //odešle soubor
```

s) Smazání souboru ze serveru – del_pripevek.php

Smazání souboru se provádí tím způsobem, že se z databáze vybere řádek s adresou souboru a funkcí `unlink("$cesta");` se soubor smaže. Nesmí se ještě zapomenout smazat samotný záznam z databáze a tím byl soubor vymazán.

t) Seznam účastníků – seznam.php

Někteří účastníci chtějí vidět, kdo je přihlášen na stejnou akci jako oni. Z tohoto důvodu je tu seznam přihlášených lidí. V něm je uvedeno jméno, příjmení a organizace, ze které přijeli.

u) Odhlášení – logout.php

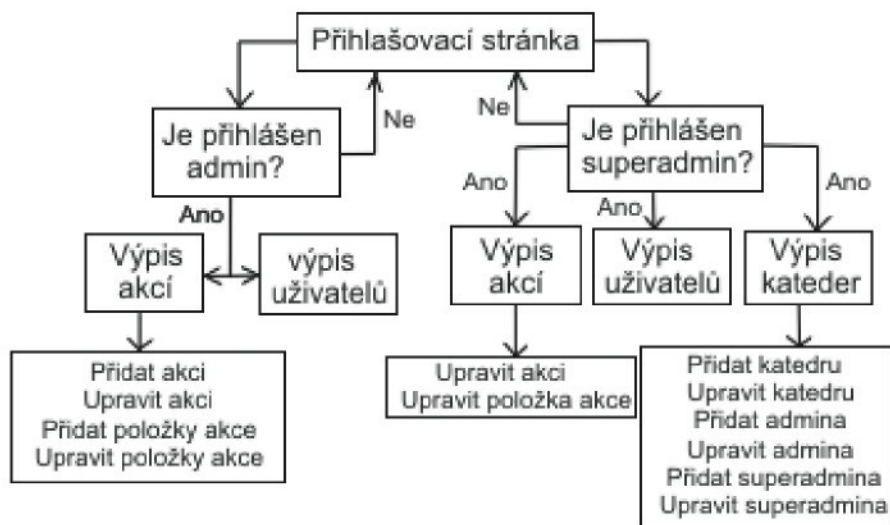
Jediným účelem této stránky je smazat dočasnou proměnnou `$_SESSION['ID']`, která určuje číslo přihlášeného účastníka. Po jeho smazání je odhlášen a přesměrován na úvodní stránku.

Tím je část pro návštěvníky stránek a účastníky akcí vyčerpána a přejde se na část pro administrátory.

6.2 BACKEND

Struktura webu – Část pro administrátory

Zjednodušená struktura webu - část pro administrátory vypadá takto:



Obr.19: Blokové schéma části pro administrátory

Do systému se mohou přihlašovat 2 druhy administrátorů. První z nich je obyčejný administrátor, který spravuje akce katedry, ke které byl přihlášen. Může je tedy zakládat, mazat, upravovat a vkládat do nich všechny potřebné položky

Druhým typem je superadministrátor, který zakládá katedry a k nim pak přiřazuje administrátory. Zakládá také další superadministrátory, ale co se týče akcí, tak je nemůže zakládat. Je schopen je ale upravovat a mazat stejně jako jejich položky.

a) Přihlašovací stránka – index.php

Přihlašování administrátorů funguje stejně jako přihlašování účastníků. Po vyplnění jména a hesla a ověření v databázi jsou vytvořeny proměnné:

- `$_SESSION['ID_adm']` (určující číslo administrátora)
- `$_SESSION['right']` (určí zda jde o admina nebo superadmina)
- `$_SESSION['ID_kat']` (určení katedry ke které administrátor patří. Superadministrátor má ID katedry 0).

Když jsou všechny uvedené proměnné vytvořeny, je administrátor přihlášen.

Po přihlášení se zobrazí tabulka s údaji o katedře – počet administrátorů katedry, počet vytvořených akcí a počet přihlášených účastníků na akce katedry. U superadministrátora je uveden počet všech přihlášených účastníků na akce.

b) Změna přihlašovacích údajů - upd_login.php

Každý administrátor i superadministrátor si zde může změnit svoje přihlašovací údaje. Je třeba vyplnit nové jméno, staré heslo, nové heslo a ověření nového hesla. Jestliže se staré heslo shoduje se záznamem v databázi a nová hesla jsou shodná, je záznam upraven.

c) Úprava kontaktu – kontakt_main.php

Tato stránka se zobrazí pouze superadministrátorovi, který má ID_adm rovno 0 (neplést s ID_kat), což je jediný administrátor, který nejde smazat, jedná se totiž o hlavního administrátora stránek. Do formuláře pak doplní kontaktní údaje na sebe, které se zobrazí v části pro návštěvníky.

d) Úprava textů – texty.php

Na této stránce, která se zobrazí pouze hlavnímu superadministrátorovi, je možné editovat všechny texty, které jsou zobrazeny na stránkách pro návštěvníky, tedy kromě samotných údajů o akci a uživateli. Například když je v účtu účastníka napsáno:

Jméno: Jan Pecina

Tak první část tohoto zápisu zde upravit jde, druhá nikoliv.

e) Administrátoři, katedry – admins.php

Jako první při zakládání administrátora je nutné vytvořit katedru, ke které bude přiřazen (neplatí pro superadministrátory).

Katedra se vytvoří vyplněním názvu (česky a anglicky) a odesláním do databáze. U administrátora se doplňuje jméno, heslo, ověření hesla a ze seznamu se vybere katedra. Podmínky pro jejich vytvoření jsou u administrátora i superadministrátora stejné. Zadaná hesla se musí shodovat a jméno nesmí být prázdné. Pokud se ještě zjistí, že jméno administrátora již v databázi je, tak se také nevytvoří.

f) Výpis účastníků

Na této stránce je výpis přihlášených účastníků akce a to buď všech (pokud je přihlášen superadministrátor) nebo jen určité katedry (pokud je přihlášen administrátor). V tomto výpisu je zobrazeno celé jméno, organizace, email, počet vytvořených příspěvků uživatele a stav jeho účtu. Stav účtu může být:

- neověřený – účastník neodpověděl na email, který dostal při registraci
- nekompletní – chybí doplnit některé údaje
- kompletní – všechny údaje jsou doplněny.

Administrátor pak může účet účastníkovi aktivovat ručně pomocí odkazu nebo pomocí dalšího odkazu účastníka smazat spolu s jeho příspěvky.

g) Smazání účastníka – del_user.php

Při smazání účastníka jsou z databáze odstraněny všechny vyplněné údaje o uživateli spolu s jeho vybranými službami a ze serveru jsou odstraněny všechny jeho příspěvky.

h) Výpis účastníka – user.php

V podstatě je to samé jako stránka „Informace o účtu účastníka“, kde si zde administrátor může zjistit informace, co účastník doplnil a co ještě chybí doplnit, prohlédnout si jeho příspěvky a vytisknout fakturu.

i) Výpis akcí – akce.php

Když je přihlášen superadministrátor, tak jsou vypsány všechny akce, pokud je přihlášen administrátor, tak jen akce katedry, ke které byl určen. Vybráním akce se zobrazí její výpis s nabídkou dalších možností, které jdou u akce upravit nebo do ní přidat (výbory, ceník, ubytování apod.).

Administrátor má ve výpisu ještě možnost si vytisknout fakturační údaje o účastnících jednotlivých akcí a v levém menu položku pro vytvoření nové akce.

j) Založení akce – add_akce.php

Když je vybrána tato možnost, tak je nejdříve administrátor vyzván k určení typu akce, kde je na výběr mezi konferencí a seminářem.

U konference se do připraveného formuláře doplňuje název akce, určení, cíl, termín konání akce (podle něj je nabídnut účastníkům termín ubytování), místo konání, obsah konference a poznámky. Podmínky pro odeslání jsou takové, že musí být vyplněny všechny údaje, kromě poznámek – ty povinné nejsou.

U semináře je položek víc a to: název, garant, určení, cíl, počet hodin semináře, termín a místo konání, obsah, lektori a poznámky. Podmínky jsou zde stejné jako pro založení konference.

Na konci stránky je ještě tlačítko pro „vyčištění“ formuláře. Po odeslání totiž záznamy v jednotlivých polích zůstávají pro ověření nebo kdyby byly potřeba při založení další akce. Když odeslaná data nejsou potřeba, tak pomocí tohoto odkazu se odstraní.

k) Úprava akce – upd_akce.php

Úprava akce probíhá stejně jako vytvoření akce. Záznamy se doplňují do stejného formuláře, ve kterém jsou již vyplněny dříve zadané údaje o akci. Ty se tedy mohou upravit a odeslat zpět do databáze.

l) Založení/úprava akce anglicky - add_akce_aj.php

Založení a úprava anglických textů pro akci probíhá stejně jako při práci na české verzi akce. Pouze jsou zde vyplněny některé informace, které jsou shodné pro oba jazyky a to počet hodin semináře a termín konání akce.

m) Ceník – cenik.php

V ceníku jsou uvedeny všechny doplňkové služby, které si může účastník vybrat vyjma ubytování a vložného, které mají ceník vlastní. Jedná se tak třeba o položky jako publikaci ve sborníku, publikaci na DVD, volba exponátu apod. Položka se do ceníku přidává vyplněním 3 polí a to názvem položky, názvem anglicky a cenou za položku. Povinné je vyplnit první a třetí pole, bez nich nebude záznam vytvořen.

n) Ubytování – ubytovani.php

Do ubytování se stejně jako u ceníku doplňují 3 údaje (název, název anglicky a cena za noc). Povinné jsou opět první a třetí údaj. Administrátor ještě může určit, pomocí zatrhávajících políček, jestli jde o samostatný pokoj nebo jestli se bude cena započítávat do celkové ceny.

o) Výbory – vybory.php

Zde se doplňují, upravují a mažou výbory pro dané akce.

p) Možnosti účasti – účast.php

Na této stránce je výpis všech možností, kterými se dá přihlásit na akci. Formulář pro založení takové možnosti má 5 polí, do kterých se zadává:

- Možnost účasti (česky) – toto je položka, ze které si uživatel vybírá, když se chce na akci přihlásit. Taková možnost vypadá třeba takto: „jako student, s přednáškou“ apod.
Ve výsledku se účastníkovi ukáže **Akce se zúčastním: jako student.....**
- Možnost anglicky
- Položka v ceníku – tato položka se zobrazí ve výpisu služeb účastníka k zaplacení poté, co si vybral typ účasti. Doplňuje se zde např.: „Normální vložné, studentské vložné“ apod.
- Položka v ceníku anglicky
- Cena – cena vložného, která se má zaplatit

Podmínkou pro vytvoření je vyplnění všech položek česky a určení ceny.

Ještě je zde možné zvolit, jestli bude povoleno na akci přihlásit exponát a pokud ano, tak se určuje cena za jeho vystavení.

r) Termíny – termíny.php

Zde se doplňují, upravují a mažou termíny pro dané akce.

s) Sponzoři – sponzori.php

Pro vytvoření sponzora je nejdříve nutné nahrát jeho logo. V textovém poli se vybere soubor a ten se posléze odešle na server. Logo se pak na server nahraje ve formě, která určuje složku a jméno souboru: *sponzori/ID_katedry/ID_akce/ID_sponzora.kon*. Znovu tak nemůže dojít k duplicitě souborů. Když je logo na serveru, lze k němu přidat odkaz na stránky sponzora (požadovaný tvar odkazu je: *http://adresa.cz* nebo *http://www.neco.cz*). Při smazání loga je z databáze odstraněn odkaz a ze serveru soubor s obrázkem.

t) Pokyny pro autory – pokyny.php

Do pokynů pro autory se nahrávají pomocí formuláře texty, které obsahují informace s organizačními pokyny. Když je potřeba, tak se mohou poslat i soubory se šablonami a vzory příspěvků. Na stránce se pak zobrazí odkaz na soubor a účastník si ho bude moci stáhnout. Protože název souboru zůstává stejný (jak ho má admin pojmenovaný, tak se nahraje na server), je nezbytné, aby neobsahoval žádné čárky, háčky a mezery v názvu, jinak nebude fungovat. Na server se nahraje, ale odkaz bude nefunkční – v internetu se háčky ani čárky prostě nepoužívají!

u) Kontakt – kontakt.php

Zde se doplňuje kontakt na katedru, který se pak zobrazí u vybrané akce.

v) Exkurze – exkurze.php

Na této stránce se zakládají exkurze pořádané spolu s akcí. Exkurze se zakládá vyplněním 5 polí (názvu česky, názvu anglicky, popisu exkurze česky, popisu exkurze anglicky a ceny za exkurzi). Pro vytvoření exkurze je nutné vyplnit všechna pole česky a pole s cenou.

**Fakulta Strojní
Technické univerzity v Liberci**
pořádá KONFERENCE a SEMINÁŘE

Domů Akce Přihlásit se Kontakt

Akce

Zde jsou informace o akci na kterou jste se přihlásil(a). Pokud máte registraci na více akcí tak je lze přepínat v "souhrnném přehledu".

ČESKY ENGLISH

Katedra aplikované kybernetiky Principia Cybernetica 2010

- Informace o akci
- Výbory
- Termíny
- Pokyny pro autory
- Kontakt
- Časák
- Registrace
- Sponzoři
- << Zpět na akce

Informace o akci

Určení: Katedra aplikované kybernetiky, Fakulty strojní, Technické univerzity v Liberci pořádá v návaznosti na dlouholetou tradici 33. setkání odborníků a podepopů z kateder a ústavů technické kybernetiky, automatizace, řízení a řídicích systémů z vysokých škol a univerzít České a Slovenské republiky. V rámci setkání je pořádána mezinárodní konference Principia Cybernetica 2010.

Cíl: Naše liberecká katedra byla spolupořadatelem historicky prvního Sjezdu kateder automatizace v roce 1978 a věříme, že bude důstojným hostitelem i v roce 2010.

Cílem mezinárodního setkání je konference výzev, zhodnocení úspěchů i problémů spojených s oborem našich pracovišť. Stejně tak je cílem setkání i diskuse o problémech spojených se zapojováním výuky automatizačních oborů na různých stupních vysokoškolského studia.

Nevětším očekávaným přínosem konference Principia Cybernetica 2010 by však mělo být osobní setkání pracovníků odborných pracovišť dvou zemi se společnou historií.

Přestože byly naše státy rozděleny již před 176 lety, stále jsou řešeny podobné problémy vyplývající ze stejných zájmů a dále jsou nacházena společná témata, společné projekty jako je například vydávání společných publikací bez jazykové bariéry.

Odprezentované články na konferenci budou publikovány ve sborníku vydaném v elektronické podobě na CD. Vybrané články po recenzi budou publikovány ve speciálním vydání mezinárodního časopisu Journal of the Science in Thermodynamics and Fluid Mechanics.

Termín konání: 8. 9. 2010 - 10. 9. 2010

Místo konání: Místem setkání kateder automatizace a kybernetiky Principia Cybernetica 2010 je severočeská metropole Liberec a její okolí. Přednášky a jednání budou probíhat v areálu Technické univerzity v Liberci.

Program:

středa 8. 9.

- 8:00 - 12:00 Registrace účastníků na recepci v IC
- 12:00 - 13:00 Oběd v menze Husova
- 13:00 - 13:40 Otevření konference v IC, přivítání účastníků, úvodní slovo děkanů.
- 13:40 - 15:00 Představení jednotlivých pracovišť (cca 5min na pracoviště)
- 15:00 - 15:30 Přestávka na občerstvení
- 15:30 - 18:00 Prohlídka laboratorní Katedry aplikované kybernetiky a pracovišť Fakulty mechatroniky, informatiky a mechatronických studií
- 18:00 - 19:00 Ubytování koleje Harcov (budova F)
- 19:00 Společenský večer v 5 klubu

čtvrtek 9. 9.

- 8:00 - 9:30 Snídaně
- 9:30 - 12:00 Vyzvané přednášky, prezentace příspěvků formou posterů v IC
- 12:00 - 13:00 Oběd v menze Husova
- 13:00 - 18:30 Exkurze (výroba korálů, Boskovské jeskyně)
- 18:30 - 21:00 Liberec Ještěd - diskuse účastníků, večeře
- 21:00 Návrat na koleje Harcov

pátek 10. 9.

- 8:00 - 9:30 Snídaně
- 9:30 - 11:00 Zasedání v IC, diskuse u kulatého stolu na aktuální téma
- 11:00 - 13:00 Oběd v menze Husova
- 13:00 Zakoření konference

Poznámky: Seznam pořadatelů v historii:
1978 Liberec, 1979 Ostrava, 1980 Košice, 1981 Ostrava, 1982 Praha, 1983 Liberec, 1984 Bratislava, 1985 Brno, 1986 Píseň, 1987 Košice, 1988 Opatava, 1989 Zlín, 1991 Píseň (Chod), 1992 Košice, 1993 Brno, 1994 Praha, 1995 Liberec, 1996 Bratislava, 1997 Zlín, 1998 Opatava, 1999 Brno, 2000 Píseň, 2001 Praha, 2002 Košice, 2003 Liberec, 2004 Bratislava, 2005 Ostrava, 2006 Zlín, 2007 Praha, 2008 Brno, 2009 Košice, 2010 Liberec.

© Copyright 2010, Katedra aplikované kybernetiky, TUL
Domů | Kontakt | WEBMASTER

Obr.20: Ukázka stránky s otevřenou akcí

Další náhledy stránek jsou vloženy v příloze.

7 SOUČASNÝ STAV WEBU

Konečná verze webu je již umístěna na serveru katedry kybernetiky (adresa <http://www.kky.tul.cz/konference>), takže je už možné se do systému přihlašovat. Před samotným spuštěním byla několik týdnů prováděna fáze testování, která měla za úkol odstranit případné chyby a nedostatky v uživatelském rozhraní (jak pro účastníky, tak pro adminy). Testování jsem prováděl sám za pomoci několika lidí, které jsem požádal, a konečné úpravy určoval vedoucí této práce pan Moučka. Tato verze je tedy již odladěná a žádnými neduhy by neměla trpět.

Databáze k webu nyní obsahuje 17 tabulek, ve kterých jsou uložena všechna data o účastnících, akcích, a texty pro jazykové verze stránek. Bohužel v současné době nebyly přeloženy všechny texty do angličtiny, takže anglická verze zatím není přístupná. Pokud by byly přeloženy tyto texty i do jiných jazyků, byla by tu možnost i dalších jazykových mutací těchto stránek.

Pokud bude chtít některá katedra do systému vložit vlastní akci, musí se obrátit na katedru kybernetiky, která jim administrátora vytvoří. Bez vytvořeného administrátora není možné se do systému přihlásit.

Prostor k dalším úpravám stránek by tu určitě ještě byl a to zejména v oblasti exportu dat do PDF dokumentů, který se provádí přes externí modul, se kterým jsem bohužel doposud nepracoval, a tak není tato část zcela dokonalá. Ve výsledku ale všechny části fungují, takže bych mohl říci, že hlavní cíl této práce byl splněn.

7.1 závěr

Cílem této práce bylo seznámit se s vývojem webových aplikací a zejména pak dynamických webových stránek vytvořených pomocí PHP kombinovaných s MySQL databázemi. V úvodu jsem uvedl všechny potřebné programovací jazyky, kterými jsou HTML, CSS a PHP. Poukázal jsem na všechny jejich přednosti a nevýhody, které s sebou může přinést jejich praktické použití. Velká část byla věnována nastavení samotného PHP na počítači, které obnáší vytvoření lokálního serveru a jeho konfiguraci.

Co se týká dalšího bodu zadání vytvoření stránek pro konference a semináře, je výsledná aplikace umístěna na serveru a je možné se v ní přihlašovat na akce a samotné akce i zakládat. Navržené rozhraní je pro uživatele a administrátory velmi podobné a to s ohledem na co nejlepší přehlednost nabídek.

Kvůli velkému rozsahu kódu nejsou vysvětleny jednotlivé soubory stránek konferencí seminářů, ale jsou popsány pouze z funkčního hlediska. Rozbor kódu je proveden v praktické části na jednodušším příkladě, který i tak zabírá několik stránek. Samotný rozbor obsahuje vysvětlení, co má jednotlivý řádek v aplikaci za úkol. Protože je celá aplikace programovaná ve 3 různých jazycích, bylo by dobré se věnovat programování webových stránek několik let, aby se pochopila kompletní struktura jednotlivých souborů. Základy se ale díky uvedeným textům pochopit dají.

Pro mne samotného byla tato práce určitě přínosem, protože jsem si rozšířil vědomosti o programování v PHP a zjistil jsem, jak probíhají jednotlivé fáze ve vývoji rozsáhlejšího projektu. Toto poznání nyní budu moci použít při vlastní tvorbě jiných webových aplikací, o které jsem byl při psaní této bakalářské práce požádán.

Pokud bych měl v bodech napsat, co mě tato práce přinesla:

- Naučil jsem se, jak si rozvrhnout databázi, aby byla co nejefektivnější.
- Jak co nejlépe pracovat na stránkách s databází.
- Díky stažitelným stylům stránek se zjednoduší práce při vývoji.
- Více jsem začal používat kaskádové styly.
- Zjistil jsem, že jedna z nejtěžších částí vývoje je samotné ladění a odstraňování chyb na stránkách.

8 Seznam použité literatury

- [1] Janovský, D. [mody-prohlizecu.html](http://www.jakpsatweb.cz/css/mody-prohlizecu.html). jakpsatweb.cz.
<http://www.jakpsatweb.cz/css/mody-prohlizecu.html> .
- [2] anon. [wikipedie](http://cs.wikipedia.org/wiki/HTML). <http://cs.wikipedia.org/wiki/HTML>.
- [3] anon. [wikipedie](http://cs.wikipedia.org/wiki/PHP). <http://cs.wikipedia.org/wiki/PHP>.
- [4] Zajíc, P. [linuxsoft](http://www.linuxsoft.cz/article.php?id_article=551). http://www.linuxsoft.cz/article.php?id_article=551.
- [5] anon. [mediacentrik](http://www.mediacentrik.cz/slovník). <http://www.mediacentrik.cz/slovník> .
- [6] anon. [wikipedie](http://cs.wikipedia.org/wiki/PSPad). <http://cs.wikipedia.org/wiki/PSPad>.
- [7] Janovský, D. [jak-zacit.html](http://www.jakpsatweb.cz/php/jak-zacit.html). jakpsatweb.cz.
<http://www.jakpsatweb.cz/php/jak-zacit.html>.

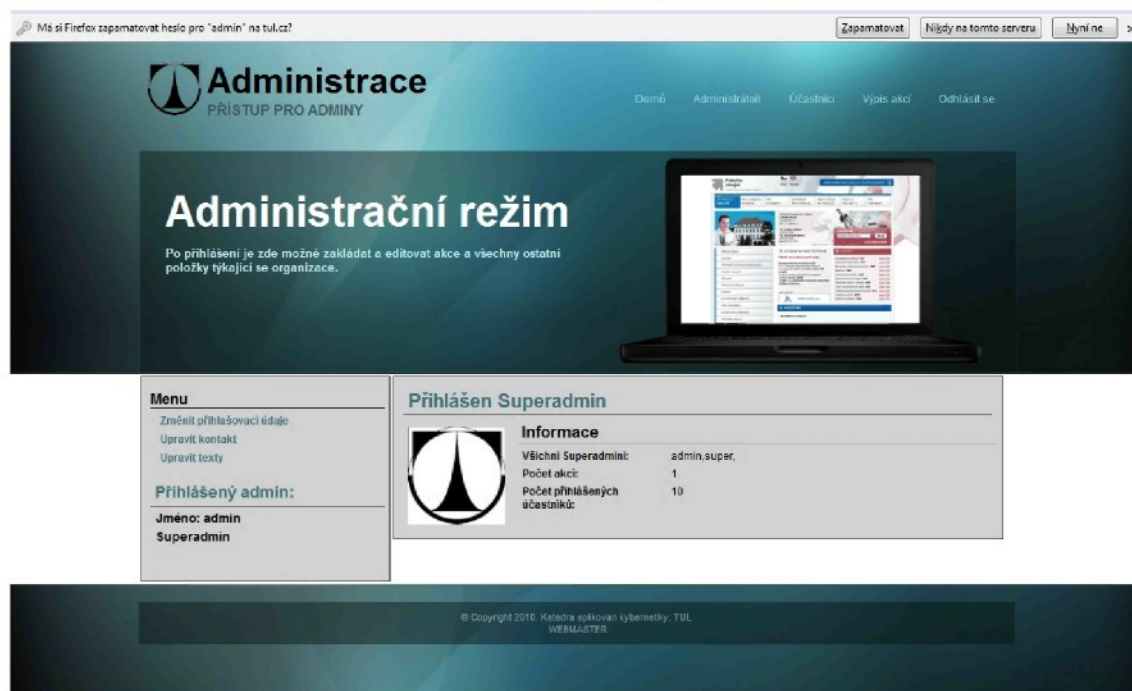
Přílohy

A) Ukázka stránek

Na následujících obrázcích jsou zobrazeny stránky konferencí a seminářů. Je na nich ukázána jak část pro uživatele tak administrátorská část (přihlášený je superadministrátor).



Obr.A1: Zobrazení stránek pro uživatele



Obr.A2: Zobrazení stránek pro adminy

B) Požadavky na spuštění a instalace

Pro správné fungování těchto stránek je třeba mít nainstalované tyto 3 aplikace:

- 1) server Apache
- 2) modul PHP pro Apache
- 3) databázi MySQL.

Všechny tyto aplikace obsahuje balíček EasyPHP.

Instalace

Před odesláním stránek na server je třeba nastavit v souboru *pripoj.php* následující řádek:
`$db=mysql_connect("server", "admin", "heslo")` a místo slov *server, admin, heslo* doplnit údaje, které jsou nastaveny pro připojení do databáze (PHPMyAdmin).

Jako další je potřeba vytvořit databázi a to pomocí souboru „databáze.php“, který se zkopíruje do kořenové složky serveru a spustí.

Nakonec se i zbylé soubory zkopírují do kořenové složky na server. Stránky se spouští přes soubor *index.php* (např. „<http://localhost/slozka/index.php>“).

Administrátorský režim je na stránce *index.php*, která se nachází ve složce *admin* (např. „<http://localhost/slozka/admin/index.php>“).

C) Seznam příloh v elektronické podobě

Příloha č.1: Bakalářská práce - text

Příloha č.2: Stránky pro konference a semináře

Příloha č.3: Balíček aplikací EasyPHP

Návod ke zprovoznění stránek

- 1) Ve složce EasyPHP na CD se spustí aplikace „EasyPHP-5.3.0-setup“. Pomocí jednoduchého průvodce instalací se aplikace nainstaluje na disk.
- 2) Ve složce na disku kam se aplikace nainstalovala se spustí „EasyPHP-5.3.0.exe“. Spuštěné okno aplikace by mělo vypadat následovně:

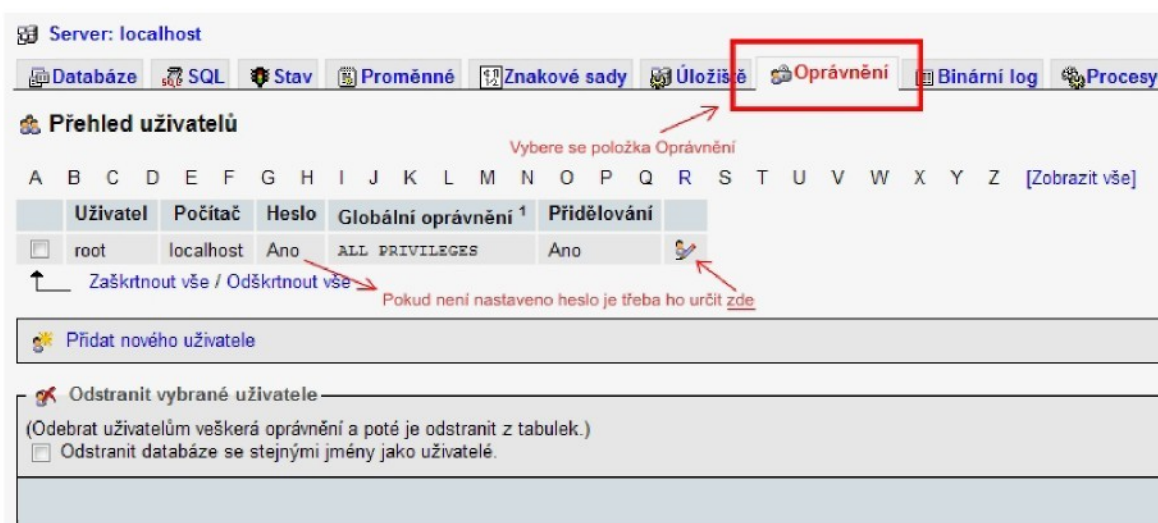


- 3) Ve stejné složce na disku kam se aplikace nainstalovala se otevře složka „phpmyadmin“ a v ní soubor „config.inc“. Následně je třeba upravit některé řádky podle následujícího příkladu:

```
/* Authentication type */
$cfg['Servers'][$i]['auth_type'] = 'config';
/* Server parameters */
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'mysql';
$cfg['Servers'][$i]['AllowNoPassword'] = false;
$cfg['Servers'][$i]['hide_db'] =
'(mysql|information_schema|phpmyadmin)';
$cfg['Servers'][$i]['compress'] = false;
/* Select mysqli if your server has it */
$cfg['Servers'][$i]['extension'] = 'mysqli';
```

Toto nastavení určí název serveru jako „localhost“, jméno admina jako „root“ a heslo „mysql“.

- 4) Do internetového prohlížeče se zadá adresa: „http://localhost/home/mysql/“. Tím se otevře PHPMYadmin. V něm se zvolí položka oprávnění a pokud je v přehledu uživatelů u uživatele „root“ určeno Heslo „Ne“ bude ho třeba určit.



Heslo se určuje v následujícím formuláři:

Změnit heslo

☐ Žádné heslo

☒ Heslo:

Vyplni se obě heslo jako "mysql"

Hašovací funkce pro heslo: ☒ MySQL 4.1+ ☐ Kompatibilní s MySQL 4.0

Vytvořit heslo

Vytvořit

Tímto se vytvoří

Proved

- 5) Zkopíruje se obsah složky „*Stranky_pro_konference_a_seminare*“ z CD do složky, kam bylo EasyPHP nainstalováno a to do podsložky „*www*“ např.: „*D:\programy\EasyPHP5.3.0\www*“.
- 6) V prohlížeči se zadá adresa „*http://localhost/databaze.php*“. Tato stránka má za úkol vytvořit databázi a tabulky. Pokud se ukáže „Databáze a tabulky byly vytvořeny“ je vše v pořádku a tabulky byly vytvořeny. Jestliže se ale ukáže „nepovedlo se připojit k serveru“ tak byly špatně vyplněny přihlašovací údaje v PHPMyadminovi. Je nezbytné aby server byl *localhost*, správce *root* a heslo *mysql*.
- 7) Stránky se spouštějí přes adresu „*http://localhost/index.php*“. Pokud se ukáže „nepovedlo se připojit k serveru“ byly špatně vyplněny údaje pro přístup do PHPMyadmina. Jestliže se ukáže „nepovedlo se připojit k databázi“ znamená to, že databáze ještě nebyla vytvořena.
- 8) Do administrátorského režimu se vstupuje přes stránku „*http://localhost/admin/index.php*“.
Jméno je: **admin**
Heslo je: **heslo**

Stránky by pak měly vypadat takto:

Fakulta Strojní
Technické univerzity v Liberci
pořádá KONFERENCE a SEMINÁŘE

Domů Akce Přihlásit se Kontakt

Vítejte

Tyto stránky slouží pro registraci účastníků na konferencích pořádaných katedrami Fakulty strojní TUL v Liberci.
Pro registraci si nejdříve vyberte akci, na kterou se chcete přihlásit, a následně vyplňte formulář u příslušné akce.

ČESKY ENGLISH

Připravované akce

Registrace na Konference

Pomocí těchto stránek se můžete zaregistrovat na konferenci a semináře pořádané Fakultou strojní Technické univerzity v Liberci.

Jak se mám zaregistrovat?

Výpis akcí naleznete pod položkou Akce nahore v menu. Akce jsou rozděleny podle kateder a po kliknutí na příslušnou akci se Vám zobrazí všechny informace o požadované akci spolu s elektronickou přihláškou.

Po vyplnění základních údajů a odeslání přihlášky Vám bude zpět zaslán email, který musíte potvrdit, aby byl Váš účet plně aktivován.

Po prvním přihlášení nezapomeňte doplnit zbývající údaje, aby byla Vaše přihláška přijata. Dále je možné nahrát do systému Vaše příspěvky a to v české i anglické verzi.

Přihlásit se

Email:

Heslo:

Zapomněl heslo

Přihlásit se

© Copyright 2010. Katedra aplikované fyziky, TUL
Domů | Kontakt | WEBMASTER