

WWW stránky pro výuku předmětu Metody umělé inteligence

WWW Pages for Education in Subject Methods of Artificial
Intelligence

Antonín Výborný

Bakalářská práce
2008



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2007/2008

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Antonín VÝBORNÝ**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **WWW stránky pro výuku předmětu Metody umělé inteligence**

Zásady pro vypracování:

Cílem bakalářské práce je naprogramování www stránek pro účely předmětu tak, aby měly jednotný design. Administrátor, by měl mít možnost vkládat informace, tvořit jednotlivá menu, podmenu sám ve vytvořeném uživatelském prostředí bez zásahu do html kódu.

1. Studium prostředí HTML, CSS, PHP a SQL.
2. Vytvořit design stránek pro předmět Metody umělé inteligence.
3. Vytvořit administrátorské rozhraní pro vytváření stránek, jejich úpravu, či mazání.
4. Závěr.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. Castagnetto J. a kol., PHP Programujeme profesionálně – 2. opravené vydání, Computer Press, Praha, 2001, ISBN 8072263102.
2. Welling L., PHP a MySQL, SoftPress, Praha, 2004, ISBN 8086497607.
3. Šimůnek M., SQL – kompletní kapesní průvodce, Grada, Praha, 1999, ISBN 8071696927.
4. Schurman E. M., Pardi W. J., Dynamické HTML v akci, Computer Press, 2000, ISBN 807226401X.

Vedoucí bakalářské práce:

Ing. Zuzana Oplatková

Ústav aplikované informatiky

Datum zadání bakalářské práce:

20. února 2008

Termín odevzdání bakalářské práce:

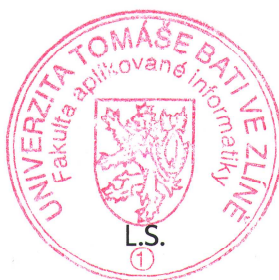
5. května 2008

Ve Zlíně dne 20. února 2008



prof. Ing. Vladimír Vašek, CSc.

děkan



doc. Ing. Ivan Zelinka, Ph.D.

ředitel ústavu

ABSTRAKT

Cílem této bakalářské práce je naprogramování webové aplikace pro výuku předmětu Metody umělé inteligence. Vytvořit jednotný design a administrátorské rozhraní, ve kterém má administrátor možnost vkládat informace, jednotlivá menu a podmenu sám bez zásahu do HTML kódu.

Klíčová slova: CMS, redakční systém, PHP, XHTML, CSS, MySQL, www, JavaScript

ABSTRACT

The aim of this bachelor project is to program the web application for education in Methods of artificial intelligence subject and to create an integrated design and administration interface where the administrator has an opportunity to insert information, menus and submenus on his own without the alteration of HTML code.

Keywords: CMS, content management, PHP, XHTML, CSS, MySQL, www, JavaScript

Na tomto místě bych rád poděkoval své vedoucí bakalářské práce Ing. Zuzaně Oplatkové, za odborné vedení, cenné rady, připomínky, a také za čas věnovaný mé práci.

Motto:

Dokumentace je jako sex: je-li dobrá, je velmi, velmi dobrá. Je-li špatná, je stále lepší než nic.

Dick Brandon

Prohlašuji, že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. v případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně

.....
Podpis diplomanta

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 REDAKČNÍ SYSTÉMY	10
1.1 HISTORIE	10
1.2 ZÁKLADNÍ FUNKCE REDAKČNÍCH SYSTÉMŮ	11
1.3 ROZDĚLENÍ CMS	11
2 MODERNÍ WEBOVÉ TECHNOLOGIE	12
2.1 ZNAČKOVACÍ JAZYKY	12
2.1.1 HTML.....	12
2.1.2 XML.....	13
2.1.3 XHTML.....	13
2.1.3.1 XHTML DTD	14
2.1.3.2 Verze XHTML.....	14
2.1.3.3 Nevýhody	14
2.1.3.4 Výhody.....	15
2.1.3.5 Základy syntaxe	15
2.1.3.6 Rozdíly mezi HTML a XHTML.....	16
2.2 PROGRAMOVACÍ JAZYKY	17
2.2.1 JavaScript	18
2.2.1.1 Vlastnosti JavaScriptu.....	18
2.2.1.2 Možnosti JavaScriptu.....	19
2.2.1.3 Základy syntaxe	19
2.2.1.4 DOM	20
2.2.1.5 TinyMCE	20
2.2.2 PHP	20
2.2.2.1 Výhody.....	21
2.2.2.2 Nevýhody	21
2.2.2.3 Základy syntaxe	21
2.2.3 JSP.....	22
2.2.3.1 Výhody a nevýhody.....	22
2.3 OSTATNÍ TECHNOLOGIE	22
2.3.1 MySQL.....	23
2.3.2 CSS.....	23
2.3.2.1 Výhody.....	23
2.3.2.2 Nevýhody	24
2.3.2.3 Použití CSS	24
II PRAKTICKÁ ČÁST	25
3 TVORBA WEBOVÉ APLIKACE	26
3.1 POUŽÍVANÝ SOFTWARE	26
3.1.1 Adobe Photoshop	26
3.1.2 IrfanView.....	26

3.1.3	PSPad	26
3.1.4	VertrigoServ	26
3.2	DESIGN	27
3.2.1	Inspirace	27
3.2.2	Tvorba vzhledu	27
3.2.3	Připojení CSS	28
3.3	MENU	29
3.3.1	Jak to funguje	29
3.4	DATABÁZE	30
3.4.1	Databáze administrátorů	30
3.4.2	Databáze článků	30
3.4.3	Databáze menu	30
4	ADMINISTRÁTORSKÉ ROZHRANÍ – CMS	32
4.1	STRUKTURA CELÉHO SYSTÉMU	32
4.2	PŘIHLAŠOVÁNÍ A AUTORIZACE	34
4.2.1	Bezpečnost	35
4.2.2	Odhlášení	35
4.3	PŘÍSTUP K DATABÁZI	35
4.4	GENEROVÁNÍ MENU	35
4.5	VÝBĚR ČLÁNKŮ	36
4.6	OBSLUŽNÉ SKRIPTY	36
4.6.1	Vytvořit složku	36
4.6.2	Vytvořit stránku	37
4.6.3	Úvodní stránka	37
4.6.4	Nahrát soubor	37
4.6.5	Mazat	38
4.6.6	Přesunout	38
4.6.7	Přejmenovat	38
4.6.8	Přiřadit odkaz	39
4.6.9	Upravit stránku	39
4.6.10	Vložit administrátora	39
4.6.11	Smazat administrátora	39
4.7	WWW STRÁNKY ONLINE	40
	ZÁVĚR	41
	ZÁVĚR V ANGLIČTINĚ	42
	SEZNAM POUŽITÉ LITERATURY	43
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	44
	SEZNAM OBRÁZKŮ	45
	SEZNAM PŘÍLOH	46

ÚVOD

V dnešní době, kdy se internet stal každodenní součástí života téměř všech lidí, neustále narůstá potřeba sdílet různorodé informace a data co možná nejpohodlněji a nejefektivněji.

Dříve byly webové stránky vytvářeny pouze statické, jejich správa a udržování byla výhradou programátorů, kteří znali jazyk *HTML*. Tvorba takových stránek nebyla příliš efektivní a každá i sebemenší změna musela být prováděna těmito programátory. Postupem času se doba stávala více uspěchanou a začala vznikat potřeba nějakého rafinovanějšího způsobu. Začaly vznikat programovací jazyky jako *PHP*, *JavaScript* či třeba *Java*, které dokázaly statické stránky „rozhybat“ tak, že se jejich obsah mohl měnit dle požadavků uživatele. Dnes je tento způsob již samozřejmostí, nicméně i tak vyžaduje programátora, který se o vytváření a údržbu musí starat. Netrvalo však dlouho a začaly se objevovat webové stránky, které umožnily návštěvníkům vytvářet a ukládat vlastní články, začaly vznikat velká diskusní fóra a obsáhlé portály, ve kterých kolektiv autorů publikuje stovky i tisíce nejrůznějších článků řazených do rozličných témat. Takovéto systémy jsou dnes známy pod pojmem *redakční systémy* nebo také *publikační systémy*. Vhodnější by bylo používat termín *systémy pro správu informací* (*CMS – z anglického content management system*), neboť řada dnešních takových systémů již neslouží pouze pro práci redakce. Tento způsob vytváření webu je fenoménem poslední doby a jeho největší předností je, že pro vytvoření stránek není nutno zasahovat do kódu a tak je zvládne i běžný uživatel internetu.

Tématem této bakalářské práce je naprogramovat takový CMS pro účely předmětu Metody umělé inteligence tak, aby byl administrátor schopen vytvářet kompletní web, vkládat menu, informace i soubory, třídit je, mazat, a také kdykoliv později upravovat.

I. TEORETICKÁ ČÁST

1 REDAKČNÍ SYSTÉMY

Redakční systém, nebo také systém pro správu informací (*CMS* z anglického *content management system*) je software určený k organizaci informací a především textů v redakcích. Může se zdát, že se tím pole působnosti omezuje pouze na redakce, faktem však zůstává, že se takovýto systém používá téměř všude, kde je potřeba publikovat velké množství informací a ty posléze třídit, vyhledávat nebo i upravovat. Nejčastější uplatnění těchto systémů dnes nalézáme u webových aplikací, někdy bývají dokonce provázeny podpůrnými programy instalovanými u klienta. [1]

1.1 Historie

V prvopočátcích byl nejčastější způsob publikování na internetu takový, že stránka, vytvořená v nějakém HTML editoru, byla pomocí FTP poslána na nějaký www server. To bylo sice snadné a rychlé, ale pro správu rozsáhlejších projektů naprosto nevyhovující, neboť nutil k mnoha rutinním činnostem. Například přidání jediného článku vyžadovalo nejen stránku vytvořit a poslat na server, ale také třeba úpravu titulní stránky s odkazem na nově vytvořenou stránku, úpravu archivu článků a všech stránek, které se na novou měly odkazovat. Vyžadovalo to tedy spoustu práce navíc, která neměla se samotným napsáním článku nic společného. Pro malé projekty, jako jsou osobní stránky, se však tohoto způsobu využívá často ještě dnes, zejména kvůli ceně.

Složitost údržby větších projektů však rapidně stoupá a s ní stoupá také počet možných chyb. To byl důvod vzniku prvních speciálních aplikací, kterým se díky svému nejčastějšímu využití začalo říkat redakční systémy.

Zpočátku se zdálo, že bude jednodušší a efektivnější, když bude redakční systém fungovat na straně klienta, což znamenalo instalaci softwaru v počítači redaktora, pomocí kterého jsou vytvářeny články a struktura webu, které poté pomocí FTP nahraje na server. Výhodou měla být zejména rychlá interakce programu a redaktora. Záhy se však ukázalo, že tímto způsobem nemohou být vytvořeny komplexní rozsáhlé systémy a už vůbec ne systémy, na kterých spolupracuje kolektiv autorů. Začaly tedy vznikat nové aplikace, založené na typu klient-server, kdy jsou veškerá data rovnou posílána na centrální server, k němuž mohou všichni uživatelé přistupovat v rámci svých autorských práv.

1.2 Základní funkce redakčních systémů

Mezi základní funkce CMS patří:

- Tvorba a editace článků, většinou pomocí webového formuláře (někdy také i jednoduchého WYSIWYG editoru).
- Správa uživatelů: administrátoři, redaktoři, běžní uživatelé.
- Správa souborů, článků či obrázků.
- Volba designu (většinou pomocí přednastavených šablon).
- Další funkce jako kalendářní funkce, statistiky přístupů, fulltextové vyhledávání.

1.3 Rozdělení CMS

Na trhu je spousta CMS, které můžeme dělit na komerční i nekomerční aplikace. Dále je můžeme dělit podle ceny či výkonu, který se může mnohdy značně lišit. Důležitým kritériem při výběru redakčního systému je ale také zaměření na určitou cílovou skupinu, neboť jejich využití bývá opravdu různé, a tím se liší také jejich funkční zařazení. v neposlední řadě je třeba dělit CMS i podle programovacího jazyka, který byl při tvorbě využíván. Těmi bývají nejčastěji *JavaScript*, *PHP* a *Java*. [1]

2 MODERNÍ WEBOVÉ TECHNOLOGIE

Vývoj jazyků užívaných při tvorbě webových stránek jde stále kupředu a v dnešní době již existuje nespočet standardů, které můžeme při tvorbě používat. Jazyky můžeme dělit na programovací a značkovací a jejich použití se značně liší.

2.1 Značkovací jazyky

Zdrojovým kódem značkovacích jazyků je obyčejný textový soubor umožňující jeho snadnou editaci, neboť obsahuje kombinaci informací a formátovacích značek (tagů). Nepochybně nejvýznamnějšími představiteli značkovacích jazyků jsou *XML* a *HTML*, které pomocí značek dokáží například vymežit nadpis nebo odkud kam sahají odstavce v textu. Mezi značkovací jazyky jsou někdy řazeny i procedurální jazyky jako *TeX* či *PostScript*, ale toto zařazení není příliš přesné, neboť v sobě obsahují také výkonné instrukce na úrovni programovacího jazyka - typicky určitou formu pamětí a proměnných. [1]

2.1.1 HTML

HTML je zkratkou *Hypertext Markup Language*, což překládáme jako „značkovací jazyk pro hypertext“. První návrh HTML pochází z roku 1989. Autory byli Tim Berners-Lee a Robert Caillau, kteří pracovali pro *CERN (Central Européene de Recherche Nucléaire)*. Kompletní jazyk HTML verze 1.0 byl dokončen v červenci 1993. Původně HTML obsahovalo pouze malou množinu značek, které umožňovaly vytvářet hypertextové dokumenty. Takto vytvořené dokumenty měly být primárně používány pro publikování na tehdy se tvořícím webu. HTML se rychle vyvíjelo, bohužel ne příliš koordinovaně. Už v listopadu 1993 Dave Raggett navrhnul další HTML specifikaci, která obsahovala tabulky, formuláře a rovnice. v červnu 1994 byla vytvořena pracovní skupina, která měla tyto změny zavést do HTML 2.0. Protože různých verzí bylo mnoho, ujal se vývoje HTML konsorcium *W3C* - společnost lidí, která tvoří doporučení ohledně různých webových formátů, z nichž některá se posléze stanou standardem.

Na konci roku 1994 se vynořili první prohlížeče HTML. Ale postupem času si začali výrobci prohlížečů přidávat do jazyka svoje vlastní značky, což mělo za následek to, že jeden dokument se zobrazoval v různých prohlížečích jinak. Také mnoho autorů dokumentů (většinou webových stránek) začalo používat původně logické značky (H1 je

nadpis) jako vizuální (H1 bude zobrazeno velkým písmem...). Proto vznikly kaskádové styly *CSS* (*Cascading Style Sheet*), posléze *CSS2*. Ty měly objevit už objevené - oddělení logické a vizuální struktury dokumentu. Kaskádové styly umožňují přizpůsobovat vzhled elementů svým potřebám, přitom jsou tyto stylové informace odděleně od původního dokumentu, většinou ve zvláštním souboru.

Každá další verze HTML přinesla několik novinek (například rámce), poslední verzí HTML je 4.01. Podle původního předpokladu se mělo jednat o poslední verzi, po které by se přešlo na *XHTML*, ale pár let po ukončení vývoje byla založena nová pracovní skupina HTML, jejímž cílem je vznik nové verze *HTML 5.0*. Tato verze je však ve vývoji, a zda přinese nějaké výrazné změny, se dozvíme teprve za několik let. [2]

2.1.2 XML

XML z anglického *eXtensible Markup Language*, česky rozšiřitelný značkovací jazyk, je obecný standardizovaný značkovací jazyk. Je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Jazyk umožňuje popsat strukturu dokumentu z hlediska věcného obsahu jednotlivých částí, nezabývá se sám o sobě vzhledem dokumentu nebo jeho částí. Prezentace dokumentu (vzhled) se potom definuje pomocí připojeného stylu. Jazyk XML nemá žádné předdefinované značky (tagy, názvy jednotlivých elementů) a také jeho syntaxe je mnohem přísnější než u HTML. [1]

2.1.3 XHTML

S příchodem každé nové technologie se řeší problém co nejméně bolestivého přechodu na tuto technologii. Zavádění všech doporučení a formátů návazných k XML je velmi zdoluhavé a pomalé, navíc většina z těchto formátů je v neustálém vývoji, tedy nějaká jejich stabilní a dobrá implementace je jen těžko myslitelná, natož proveditelná. Proto konsorcium W3C sáhlo po elegantním řešení: redefinovat HTML 4.01 jako instanci XML. Vznikl tak jazyk *XHTML* (*eXtended HyperText Markup Language* - rozšiřitelný hypertextový značkovací jazyk) verze 1.0, který má totožnou sadu značek jako HTML, ale je formálně definovanou instancí XML, čili může plně využívat všech (i teprve potencionálních) vymožeností XML. Přitom zůstává zpětně kompatibilní, a tedy běžný uživatel brouzdající po webu by v okně svého prohlížeče změnu vůbec neměl poznat. V dnešní době se spolu s HTML 5.0 paralelně pracuje na vývoji verze *XHTML 2.0*. [2]

2.1.3.1 XHTML DTD

DTD (*Document Type Definition* – definice typu dokumentu) slouží k popisu struktury XML dokumentu. v XHTML dokumentu je povinnost uvádět jej hned na první řádce.

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

2.1.3.2 Verze XHTML

XHTML 1.0 existuje ve třech verzích: *Strict*, *Transitional* a *Frameset*:

- *XHTML 1.0 Strict*: Používá se, pokud chceme mít strukturovaný dokument osvobozený od formátovacích značek souvisejících s rozvržením stránky. Předpokládá se jeho užití společně s CSS, které umožní dosáhnout potřebných grafických efektů.
- *XHTML 1.0 Transitional*: Je přechodným DTD pro webové stránky, který umožňuje používat překonané tagy. Je vhodný pro formátování stránek vytvářených pro staré prohlížeče, které nerozumí kaskádovým stylům CSS.
- *XHTML 1.0 Frameset*: Umožňuje používat zastaralé značky jako XHTML 1.0 Transitional a přidává podporu pro rámce.
- *XHTML 1.1*: Reformulace XHTML 1.0 Strict s několika změnami. Nejdůležitější jsou rozsáhlejší možnosti modularizace a odstranění prezenčních elementů, rámců a atributů *lang* a *name*, které byly v XHTML 1.0 Strict povoleny. Prezentace je kontrolována čistě pomocí CSS.

2.1.3.3 Nevýhody

Jako bezpochyby největší nevýhodu jazyka XHTML vidím prohlížeč Microsoft Internet Explorer 6, který si díky zásadní chybě neumí poradit s XML deklarací, čímž se jaksi ztrácí celá filozofie jazyka XHTML.

Někteří tvůrci by mohli tvrdit, že nevýhodou XHTML je také jeho přísnost a omezenější sada značek, což je podle mého názoru naopak jeho výhodou.

2.1.3.4 Výhody

Jak již bylo uvedeno, největší předností tohoto jazyka je právě jeho přísnost. Ta nutí webkodéra psát čistší kód, což je přinejmenším programátorskou ctností. Kód HTML obsahující chyby je sice špatný, nicméně prohlížeče se tyto chyby naučily tolerovat a výslednou stránku zobrazí správně. Kdežto, uděláme-li chybu v kódu XHTML, je to také špatně, ale prohlížeč stránku zobrazí i s chybou. Je to tedy výhoda nebo nevýhoda? Kdykoliv je na internetu publikován článek na tohle téma, ihned se rozpoutává vášnivá diskuse mezi zastánci obou jazyků. Pravda je ovšem někde mezi a říkat, že jeden jazyk je dobrý a druhý špatný, je mylné.

Filosofie jazyka XHTML nám dává mocný prostředek k oddělení obsahu od vzhledu. Říká, že celý dokument má obsahovat pouze značky takové, které nemají na vzhled webu žádný nebo minimální vliv. Vzhled stránky by poté měl být doplněn pomocí kaskádových stylů CSS.

XHTML má také další výhody jako je podpora XML fragmentů *MathML*, *SVG* a další, ale těchto využije málokdo.

2.1.3.5 Základy syntaxe

Každý XHTML dokument by měl mít minimálně následující strukturu:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"č
"DTD/xhtml11-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
<head>
<title>XHTML stránka</title>
</head>
<body>
<p>Tohle je jednoduchá XHTML stránka.</p>
</body>
</html>
```

První řádek říká, že jde o XML dokument verze 1.0 v kódování UTF-8 a následuje již výše uvedená deklarace typu dokumentu DTD. Mezi tagy `<html>` a `</html>` jsou vždy dva elementy, *head* (hlavička) a *body* (tělo dokumentu). Atribut *xmlns* v kořenovém elementu `html` určuje jmenný prostor a jazyk, který je v dokumentu použit.

Značky (tagy) se zapisují pomocí znaků *větší než* a *menší než* a mezi nimi je název tagu. Můžou být párové `<NAZEV>... </NAZEV>`, nebo nepárové `<NAZEV />`. Ukončovací tag má v názvu navíc lomítko.

Zde je ukázka několika známých tagů:

- `<h1></h1>` - `<h6></h6>`: Párový tag pro nadpisy. Existuje 6 úrovní
- `<p></p>`: Odstavce
- ``: Seznam s odrážkami
- ``: Číslovaný seznam
- `<div></div>`: Kontejnery pro určitý blok kódu
- `
`: Nepárový tag pro přechod na nový řádek

Elementy mohou mít své atributy, díky kterým mohou komunikovat například s kaskádovými styly a upřesnit tím formátování celého dokumentu. Jedním z nejdůležitějších atributů jsou jistě `id` a `class`, užívané spolu s kaskádovými styly:

```
<p class="uprostřed">Odstavec zarovnaný doprostřed</p>
```

K zarovnání tohoto elementu je však nutno nadefinovat atribut v CSS stylech. [3]

2.1.3.6 Rozdíly mezi HTML a XHTML

Jazyk XHTML vznikl vylepšením a ořezáním staršího HTML, přesto jsou téměř identické. Chtěl bych však zdůraznit slovo téměř a popsat zde několik hlavních rozdílů.

- XHTML je *case sensitive*, což znamená, že například zápisy tagů pro nadpis `<H1>` a `<h1>` nejsou totožné. v zásadě se všechny tagy píšou malými písmeny.
- Všechny elementy musí být vždy uzavřené. To, co v HTML prošlo v XHTML neprojde:

```
<p>Toto je odstavec  
<p>Tohle je další odstavec
```

Správný zápis je:

```
<p>Toto je odstavec </p>  
<p>Tohle je další odstavec </p>
```


- Tagy se nesmějí křížit:

```
<p><strong>Nějaký důležitý text</p></strong>
```

Správný zápis je:

```
<p><strong>Nějaký důležitý text</strong></p>
```

- V XHTML musí být všechny tagy ukončené a to včetně těch nepárových jako jsou `
` nebo ``. Zapisujeme je `
`, ``. [3]

2.2 Programovací jazyky

Na rozdíl od značkovacích jazyků jsou tyto prostředkem pro zápis algoritmů, což je posloupnost instrukcí, vedoucí k určitému řešení problému. Programovací jazyky lze dělit podle mnoha kritérií, ale nejčastěji je dělíme na jazyky interpretované a jazyky kompilované.

Interpretované jazyky překládá interpret až v době jejich spuštění, zatímco kompilované jsou ihned přeloženy kompilátorem do strojového kódu. Každá skupina má své výhody i nevýhody. Přední výhodou kompilovaných jazyků, jako je *C* nebo *Pascal*, je jejich rychlost, naproti tomu výhodou interpretovaných je jednodušší vývoj programů. Sem patří například *Java*, *PHP* či *Perl*.

Část interpretovaných jazyků se dále řadí do skupiny skriptovacích jazyků, které usnadňují programování ještě více. Skriptovací jazyky jistě neprodukují superrychlé programy, ale zde nejde o efektivitu kódu, jako spíše o efektivitu programování. Programování ve skriptovacích jazycích je opravdu velice snadné. Programátor se nemusí starat o deklarace proměnných, většinou zde totiž neexistuje typová kontrola, a tak proměnné, která obsahuje řetězec, může bez problémů přiřadit číselnou hodnotu. Může velice snadno převádět řetězce na čísla a naopak. Může bez obav použít neinicializovanou proměnnou a spolehnout se na to, že bude mít přednastavenou nulovou hodnotu. Všechny tyto věci sice nemají nic společného s dobrými programátorskými zvyky, ale neuvěřitelným způsobem usnadňují psaní programů. Další specialitou jsou vyšší datové typy. Patří sem třeba různé výtčové typy, anebo tzv. asociativní pole. Ty umožňují vytvořit pole, kde se místo číselného indexu využívá řetězec. Velice silnou zbraní těchto jazyků bývá zpracování textu, a to je pro změnu velice častý úkol v mnoha programech. Zpravidla se k tomu využívají tzv. regulární výrazy. Umožňují snadno spouštět externí programy a zpracovávat

jejich výstup a vyznačují se také velkou přenositelností. Mezi nejznámější skriptovací jazyky patří například *PHP* a *JavaScript*, které se staly jakýmsi fenoménem programování webových aplikací. Pro rozsáhlejší projekty se však častěji používá *Java (JSP)*, někdy také starší *ASP*. [1]

2.2.1 JavaScript

JavaScript je klientský skriptovací jazyk. To znamená, že je program vykonáván u klienta, tedy přímo v prohlížeči. Protikladem klientských skriptů jsou skripty serverové, které jsou vykonávány na serveru a na klienta jdou už jen výsledky. Mezi takové patří třeba skripty *PHP*. Výhoda klientských skriptovacích jazyků spočívá tedy v tom, že není třeba pro změnu stránky neustále obnovovat obsah stránky.

Tyto jazyky jsou také často velmi omezovány. Nejen v možnostech, ale i v jejich nebezpečnosti, protože skripty, které se vykonávají přímo na počítači uživatele, mohou bez problémů obsahovat a vykonat nebezpečný kód. Z tohoto důvodu jsou například v JavaScriptu vypnuty funkce, které by přímo pracovaly se soubory na harddisku.

JavaScript se díky tomu využívá spíše pro okamžitou úpravu *CSS* kódu či jako doplněk při vytváření webové grafiky. Pro opravdové funkce internetových stránek se ale používají skripty na straně serveru.

JavaScript se zapisuje přímo do *HTML* kódu, což je velká výhoda, protože je to jednoduché. Existují i jiné jazyky klientských skriptů, například *VBScript*, ale ty jsou však velice málo používané. Často je JavaScript zaměňován s *Javou*, ale *Java* je samostatný programovací jazyk. [3]

2.2.1.1 Vlastnosti JavaScriptu

- Interpretovaný jazyk, nemusí se kompilovat.
- *Case sensitive*, citlivý na velikosti písmen.
- Objektově orientovaný, využívá *DOM* (objektový model dokumentu).
- Závislý na prohlížeči.

2.2.1.2 Možnosti JavaScriptu

- Umožňuje interakci uživatele a dokumentu, řídit vzhled stránky a může přímo zapisovat do zdrojového kódu. Snadno se tak do dokumentu vloží informace o prohlížeči, aktuální čas, datum...
- Pomocí JavaScriptu můžeme manipulovat s obrázky a vytvářet jednoduché animace.
- Často se také používá pro kontrolu a ovládání formulářů. Kontrola je efektivnější a rychlejší než při použití serverových jazyků a minimalizuje přenos dat mezi klientem a serverem.
- Podpora cookies, malých souborů ukládaných na straně klienta. Použití je však vhodné jen k drobným účelům, které by neměly mít vliv na nezbytně nutné potřeby uživatele.
- Díky událostem je možno vytvářet interaktivní, živější stránky a obohatit je například o vysunovací či rozbalovací menu a spoustu dalších prvků.
- JavaScript neumí pracovat se soubory, kreslit grafiku ani pracovat se sítí.

2.2.1.3 Základy syntaxe

JavaScript můžeme zapisovat přímo do HTML kódu mezi tagy `<script>` a `</script>`, nebo také začleněním externího souboru `*.js`. Externích souborů se většinou užívají při větších skriptech nebo programech, zápis do kódu naopak pro menší skripty jako je kontrola formuláře. Můžeme také užít tzv. in-line zápisu, což je zápis přímo v některém z elementů stránky. Takový zápis nevyžaduje použití tagu `<script>` a užívá se pro reakci na nějaký uživatelův podnět. Tyto způsoby zapisování lze libovolně kombinovat, což je také efektivní. Příklad vypsání hlášky v JavaScriptu:

```
<script language="JavaScript" type="text/javascript">  
    alert("Vyskakující okno v JavaScriptu");  
</script>
```

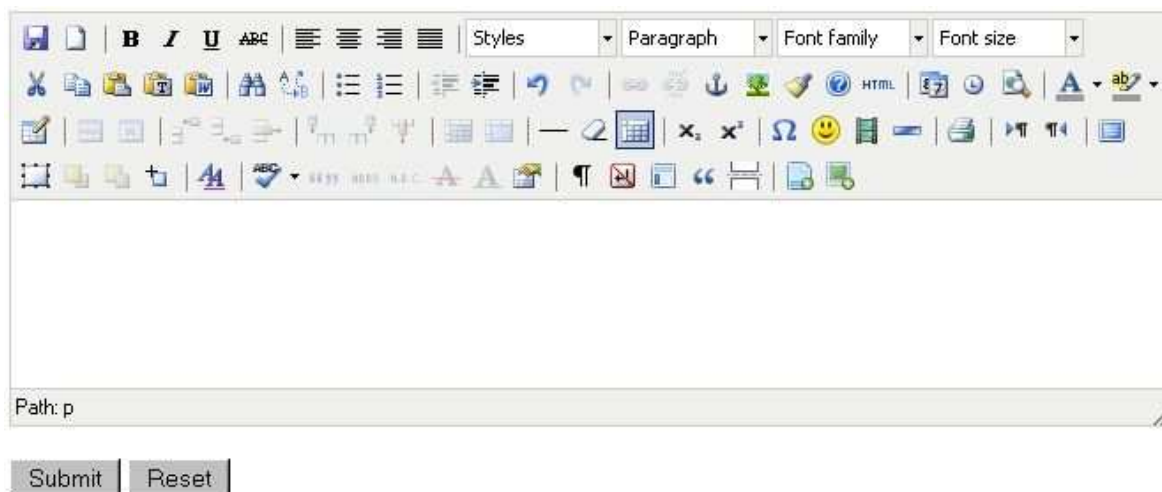
2.2.1.4 DOM

Dokument Object Model, česky objektový model dokumentu je aplikační programové rozhraní, pomocí kterého můžeme přistupovat ke každému objektu moderních značkovacích jazyků. Těmito objekty mohou být elementy, atributy, text a další.

DOM má stromovou strukturu, díky které dokáže odlišovat nadřazené, podřazené nebo rovnocenné elementy. Umožňuje na přesně určené místo přidat jakýkoli obsah a dovede to právě prostřednictvím JavaScriptu, což ho značně zpřístupňuje webovým tvůrcům. [4]

2.2.1.5 TinyMCE

O tom, že se dá v JavaScriptu napsat i skutečně rozsáhlá aplikace svědčí i webové založený textový editor *TinyMCE*. Jedná se o *open source* editor, který lze začlenit do stránek a pomocí formulářového prvku *textarea* lze posílat formátovaný XHTML1.0 výstup. [5]



Obr. 1 – TinyMCE editor

2.2.2 PHP

PHP - původně *Personal Home Page* vzniklo v roce 1995. Od té doby prošlo velkými změnami a nyní tato zkratka znamená *Hypertext Preprocessor*. v současnosti se jedná o velmi rozšířenou technologii, umožňující snadné programování na straně serveru. Právě toho lze využít k tvorbě různých interaktivních webových stránek. Stručně lze říci, že skript napsaný v PHP je proveden na serveru podle zadaných kritérií a výsledek je odeslán volajícímu počítači stejným způsobem, jakým se odesílají běžné statické HTML stránky.

Jakmile je však stránka načtena klientem, pomocí PHP ji již není možné dále měnit jako například u JavaScriptu. [6]

PHP je programovací jazyk umožňující procedurální i objektově orientované programování. PHP také patří mezi jazyky, kde například není nutné předem definovat typ proměnných, navíc jakákoli proměnná může kdykoli změnit svůj typ. Jednoduše řečeno, co se týče psaní kódu, z PHP při psaní skriptů "sálá" určitá volnost a neomezenost. Na druhé straně záleží plně na programátorovi, jaký si bude v kódu udržovat pořádek. [7]

2.2.2.1 Výhody

- PHP je velmi jednoduchý jazyk na pochopení.
- Má zabudovanou řadu funkcí, poskytujících při programování vysoký komfort.
- Snadno komunikuje s databázemi, jako je *MySQL*, *PostgreSQL*.
- PHP je multiplatformní a je podporován naprostou většinou webových serverů.
- Rychlý vývoj jazyka.

2.2.2.2 Nevýhody

- Objektově orientované programování je u PHP4 na velmi špatné úrovni a u PHP5 se stále ještě nevyrovná skutečným objektově orientovaným jazykům jako je například jazyk Java.
- Jednotlivé verze nejsou kompatibilní.

2.2.2.3 Základy syntaxe

Soubory obsahující kód PHP musí mít odlišnou koncovku od prostých HTML souborů. To kvůli tomu, že pro jejich chod je nutné, aby byly na serveru interpretovány a spuštěny. Většinou se používají soubory s koncovkou *.php. Skripty je možno vkládat mezi kód HTML nebo psát kompletně celý soubor v PHP, který může například (X)HTML stránku generovat. [8]

Kód PHP se zapisuje mezi značky `<?php` a `?>`. Proměnné není třeba předem deklarovat a určovat jejich typ, je však nutné aby jejich název začínal znakem „dolar“ \$. Jednotlivé příkazy je nutno oddělovat středníkem jako u většiny programovacích jazyků.

Ukázka PHP kódu:

```
<?php
    $roky = 24;
    echo „Právě je mi $roky let.“;
?>
```

Tento kousek kódu pošle prohlížeči zprávu *Právě je mi 24 let*. Proměnná *\$roky* se může samozřejmě kdykoliv měnit a příkaz `echo` slouží k výpisu zprávy. [8]

2.2.3 JSP

V roce 1995, kdy firma *Sun Microsystems* zveřejnila první informace o novém programovacím jazyce Java, málokdo se nad tím pozastavil. Během krátké doby si ale tento jazyk získal dominantní postavení a to především v programování pro internet. Objektově orientovaný přístup, vysoká bezpečnost a automatická správa paměti, to vše jsou hlavní přednosti Javy, která podstatně zvyšuje efektivitu práce programátora. Dalším pomocníkem je také množství knihoven, kterými tento jazyk disponuje.

Servlety jsou programy v Javě, které běží na serveru a vytvářejí HTML stránky.

Java Server Pages (JSP) je technologie, která umožňuje spojovat běžné statické stránky s dynamicky generovaným obsahem ze *servletů*. Dynamické části dokumentů se zapisují pomocí direktiv nebo speciálních značek, které většinou začínají `<%` a končí `%>`. [9]

2.2.3.1 Výhody a nevýhody

JSP má spoustu výhod oproti jiným alternativám jako ASP, PHP či JavaScriptu. Mezi nejdůležitější patří přenositelnost mezi operačními systémy a webovými servery a také to, že se JSP hodí více pro komplexní rozsáhlé aplikace, které vyžadují opětovné použití komponent. Jako nevýhodu vidím zřejmě jen nepříliš dobrou podporu ze strany webhostingových služeb, ale myslím si, že budoucnost této technologie je velice slibná. [9]

2.3 Ostatní technologie

Na vývoji moderního webu se dnes vedle již zmíněných technologií podílí spousta dalších jako například platforma *.NET Framework*, částečně vychází z Javy, ale nepředepisuje použití konkrétního programovacího jazyka. Mezi nejpoužívanější pak patří zejména *C#*, *Visual Basic* a *ASP*. Další technologií často používanou na webu je vektorový program

Flash, využívající vlastní interpretovaný jazyk *ActionScript*, který umožňuje jak tvorbu kompletních stránek, tak i vložení různých interaktivních bannerů, animací či her. Mezi další programovací jazyky užívané při tvorbě webu jsou třeba *Perl* či *Ajax*.

2.3.1 MySQL

MySQL je databázový systém, který je k dispozici jak pod bezplatnou licenci *GPL*, tak pod komerční placenou licenci. Jedná se o multiplatformní databázi a komunikace s ní probíhá pomocí jazyka *SQL*. Pro svou snadnou implementovatelnost, výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace *MySQL*, *PHP* a *Apache* jako základní software webového serveru. [10]

2.3.2 CSS

Kaskádové styly (*Cascading Style Sheets*) jsou jakousi nadstavbou značkovacích jazyků, určenou k definování jejich vzhledu. v současnosti se pracuje na verzi *CSS3*, ale skutečnost je taková, že většina internetových prohlížečů neumí ani nové funkce verze *CSS2*, proto budu popisovat starší první verzi.

CSS umožňují přiřadit každému prvku v *HTML* určité vlastnosti (vzhled, způsob zobrazení, umístění atd.). Dále umožňují definovat vlastnosti pro třídy a identifikátory, které odpovídají *class* a *id* z atributů značkovacích jazyků, ty poté tyto vlastnosti přebírají. [11]

2.3.2.1 Výhody

- *CSS* styly zjednodušují práci, a to velmi značně.
- Jsou k dispozici nové možnosti formátování, než má k dispozici *HTML*.
- Na všech stránkách je díky kaskádovým stylům konzistentní vzhled.
- Oddělují strukturu od stylu.
- Dynamická práce se styly, kompletní změna vzhledu celého webu.
- Urychlují načítání stránek.

2.3.2.2 Nevýhody

Patrně vůbec jedinou nevýhodou používání kaskádových stylů je ta, že různé prohlížeče interpretují stejný CSS kód jinak a je někdy obtížné jej napsat tak, aby se výsledek zobrazoval stejně.

2.3.2.3 Použití CSS

Styl se může nadeklarovat trojím způsobem:

- Přímo u jednotlivých elementů pomocí atributu *style*=“...”
- V hlavičce dokumentu pomocí seznamu stylů. Do stránky se zapisuje mezi tagy `<style>` a `</style>`
- Použitím externího souboru *.css, na který se stránka odkazuje tagem `<link>`

Osobně nejvíce používám externí soubory, zejména díky tomu, že pak všechny stránky vypadají podobně, mají stejný styl.

II. PRAKTICKÁ ČÁST

3 TVORBA WEBOVÉ APLIKACE

V teoretické části bylo popsáno několik hlavních technologií a jazyků, které jsou dnes běžně užívány při tvorbě webových stránek či aplikací. Při mé práci bylo využito značkovacího jazyka *XHTML* ve spojení s kaskádovými styly *CSS* a pro dynamiku byla zvolena kombinaci *JavaScriptu* a *PHP*, neboť tyto dva skriptovací jazyky se velmi dobře doplňují a každý z nich se hodí na něco jiného. Pro ukládání dat je ideální volbou databáze *MySQL*, navíc je ve spojení s *PHP* velmi rychlá a efektivní.

3.1 Používaný software

3.1.1 Adobe Photoshop

Pro návrh designu byl použit grafický editor *Adobe Photoshop*, který disponuje řadou profesionálních funkcí užitečných při tvorbě webové grafiky, umožňuje špičkovou práci s vrstvami a výsledný design se dá snadno rozřezat a optimalizovat na části pomocí sesterského programu *ImageReady*.

3.1.2 IrfanView

Tento grafický prohlížeč a editor sloužil jako pomocník při práci s obrázky, občas i k jejich optimalizaci a také k úpravě pro tuto dokumentaci.

3.1.3 PSPad

Ke psaní kódu, a to *XHTML*, *CSS*, *PHP* nebo *JavaScriptu* sice stačí obyčejný poznámkový blok, ale ten má velkou nevýhodu, nedisponuje zvýrazňováním syntaxe, což je pro delší skripty opravdu problém. Existuje mnoho kvalitních editorů, které jsou zaměřeny přímo pro tvorbu *www* stránek, z nichž jedním z nejoblíbenějších je *PSPad*. Podporuje desítky programovacích jazyků a má stovky nadstandardních funkcí jako správa projektů nebo *FTP* klient, je zdarma a hlavně je od českého autora.

3.1.4 VertrigoServ

Pro testování *PHP* skriptů je vhodné mít v počítači nainstalován virtuální server. Skripty lze poté ladit i na počítači ve kterém je aplikace vytvářena a není nutné být připojen k internetu. Výborným pomocníkem může být instalační balík *VertrigoServ*, jenž obsahuje

server Apache, PHP a MySQL systém. Instalace je velmi jednoduchá a všechny součásti mohou být ihned používány. Součástí je také *PhpMyAdmin*, nástroj napsaný v jazyce PHP pro administraci MySQL, který velmi usnadňuje tvorbu a správu databází.

3.2 Design

Existují programátoři, kteří nejprve píšou aplikaci a poté se zabývají designem, jiní nejprve tvoří design a teprve poté aplikační logiku. Druhý postup byl zvolen z toho důvodu, že pokud má tvůrce webu před sebou již vytvořený design, snadněji se rozhoduje, jaký zvolí postup při sestavování webu. Výhodou může být i fakt, že je občas nutné později upravit jedno nebo druhé a design se upravuje mnohem snadněji než skripty.

3.2.1 Inspirace

Nepochybně nejdůležitějším vodítkem při volbě vzhledu je vždy přání zákazníka, v mém případě se jednalo o přání vedoucí mé práce Ing. Zuzany Oplatkové, která bude zároveň i administrátorem webu. Design byl proto zvolen v kombinaci pastelových a pestrých barev tak, aby spolu vzájemně ladily a doplňovaly se. Bylo vyhledáno několik tutoriálů s webovými šablonami a ve Photoshopu vytvořeno několik návrhů, z nichž byl nakonec vybrán ten nejlepší.

3.2.2 Tvorba vzhledu

Samotná tvorba probíhala postupně od největších elementů jako pozadí po nejjemnější detaily jako stínování a barevné přechody. Dominantou je pak erb uprostřed hlavičky s vloženým obrázkem neuronu, který je jakýmsi symbolem a vzorem umělé inteligence, konkrétně modelu neuronových sítí. Jednotlivé části jsou od sebe ve Photoshopu odděleny ve vrstvách, což umožňuje jejich jednoduchou budoucí úpravu. Po dokončení designu byl výsledný soubor otevřen v programu ImageReady, kde byl rozřezán na nejdůležitější neopakující se části a uložen optimalizován ve formátu *PNG*.



Obr. 2 – Hlavička webové stránky

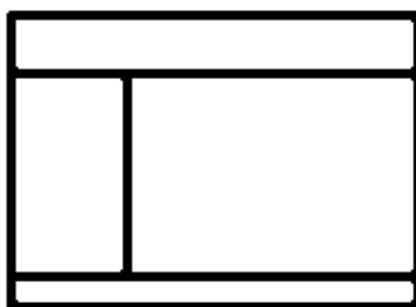
Formát *PNG* je grafický formát velmi vhodný pro webovou grafiku. Jeho největší předností je určitě podpora 24bitové barevné hloubky, bezztrátová komprese a možnost prokládání. Posledně jmenované umožňuje načtení a zobrazení obrázku po částech, což značně urychluje zobrazení webové stránky.

Později byly k hotovému designu přidány ikonky, které slouží k rozlišování souborů jako *.doc, *.pdf a další.

3.2.3 Připojení CSS

V dalším kroku bylo třeba sestavit jednotlivé rozřezané části, přičemž bylo snahou, aby zůstalo zachováno stejné zobrazení ve všech prohlížečích. Sestavený design byl testován na několika verzích *MSIE* od verze 5.01, dále na prohlížečích *Mozilla Firefox*, *Opera*, *Netscape Navigátor* a *Safari*.

Byla tedy vytvořena prázdná XHTML stránka a k ní byl pomocí externího souboru přiřazen styl. CSS soubor, pojmenován jako default.css, obsahuje barvy, písma, pozadí a styly některých tagů. Poté byl testovací soubor index.html rozdělen pomocí kontejnerů `<div>`, kterým byly přiřazeny identifikátory, pomocí nichž byl nadefinován styl a hranice jednotlivých částí webu. Byla zvolena klasická struktura webu, hlavička nahoře, patička dole, vlevo menu a zbytek zobrazovací plocha. Parametry byly zapsány v procentech, kvůli plovoucímu layoutu.



Obr. 3 – Klasické rozložení stránky

Ukázka kódu ze souboru default.css:

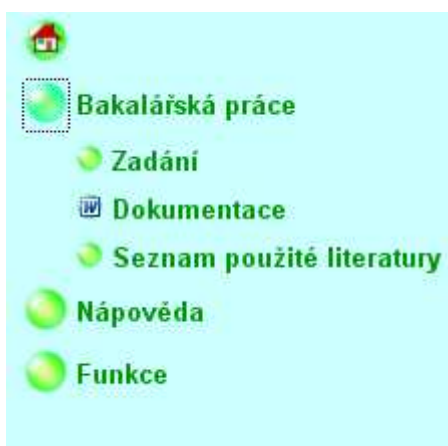
```
body {  
    font-family: Helvetica, Arial, Verdana, Tahoma, sans-serif;  
    font-size: 80%;  
    background: #99ffcc;  
    color: black;
```

```
}  
#head {  
  background: url("../images/mid.png") no-repeat 50% 0%;  
  height: 148px;  
}
```

V definici elementu *body* je uveden styl, jenž bude aplikován na celou stránku, font a barvy. Blok *head*, který začíná mřížkou, určuje styl elementu podle identifikátorů id. Konkrétně jde o nastavení erbu hlavičky, který se bude nacházet uprostřed, proto těch 50%.

3.3 Menu

Pro vytvoření dynamicky se rozbalujícího menu ve stylu průzkumníka je potřeba JavaScriptu neboť je nutná okamžitá odezva, proto musí být užito jazyka, který nepotřebuje odpověď serveru. Položky v menu mohou být dvojího druhu: složky a soubory. Soubory jsou definovány jako klasický odkaz a mohou odkazovat buď na fyzický soubor, nebo na libovolnou stránku. Se složkami je to ovšem složitější. Při kliknutí na složku chceme, aby se stromovitě rozbalil obsah této složky. Jeden ze způsobů jak toho dosáhnout, je přiřadit každému kontejneru představujícímu obsah složky jednoznačný identifikátor, a ten poté předávat JavaScriptové funkci jako parametr. Funkce v Javascriptu tento parametr zpracuje a umožní zviditelnění obsahu složky.



Obr. 4 - Menu

3.3.1 Jak to funguje

Funkce *Toggle* napsaná v JavaScriptu získá *id* určité složky a zjistí, zda je rozbalená či nikoliv. Zjednodušená funkce vypadá následovně:

```
function Toggle(item) {
```

```
obj=document.getElementById(item);
if (obj != null){
    visible=(obj.style.display!="none");
    key=document.getElementById("x" + item);
    if (visible) obj.style.display="none";
    else obj.style.display="block";
}
}
```

Kontejner, který je označen tímto *id* změní stav svého zobrazení. Buď se zviditelní, nebo schová.

3.4 Databáze

Databáze je velmi důležitou součástí celého systému. Je rozdělena na tři tabulky, určené ke třem různým účelům.

3.4.1 Databáze administrátorů

Tabulka *user* má sloupce *ID*, *jmeno* a *heslo*. *ID* je primárním klíčem a administrátor se zvoleným jménem má uloženo také heslo zašifrované pomocí šifry *MD5*.

3.4.2 Databáze článků

Tabulka *clanky* obsahuje sloupce 4: Primární klíč *ID*, *nazev*, *text* a *odkaz*. Každý článek tedy obsahuje automaticky generovaný nebo zadaný název, text vytvořený administrátorem v textovém editoru *TinyMCE* a také odkaz, který slouží k vyhledávání článku. K této tabulce mají přístup všichni návštěvníci webu, ale upravovat ji může jedině pověřený administrátor.

3.4.3 Databáze menu

Nejsložitější tabulka *menu*, která má 7 sloupců slouží k uchování položek v menu. Tato tabulka není přístupná jiným uživatelům stránek než administrátorům. Slouží k vygenerování fixního souboru, který je pomocí příkazu *require* vložen do každé stránky. Je to zejména kvůli rychlosti a efektivitě. Sloupec *ID* je jako obvykle primární klíč, *nazev* obsahuje řetězec s názvem, který bude zobrazen v menu. Sloupec *xnazev* je upravený název tak, aby neobsahoval českou diakritiku, a kvůli ošetření duplicity položek se stejným názvem může být přidáno pořadové číslo. Sloupec *odkaz* se odkazuje na stránku, jež podle

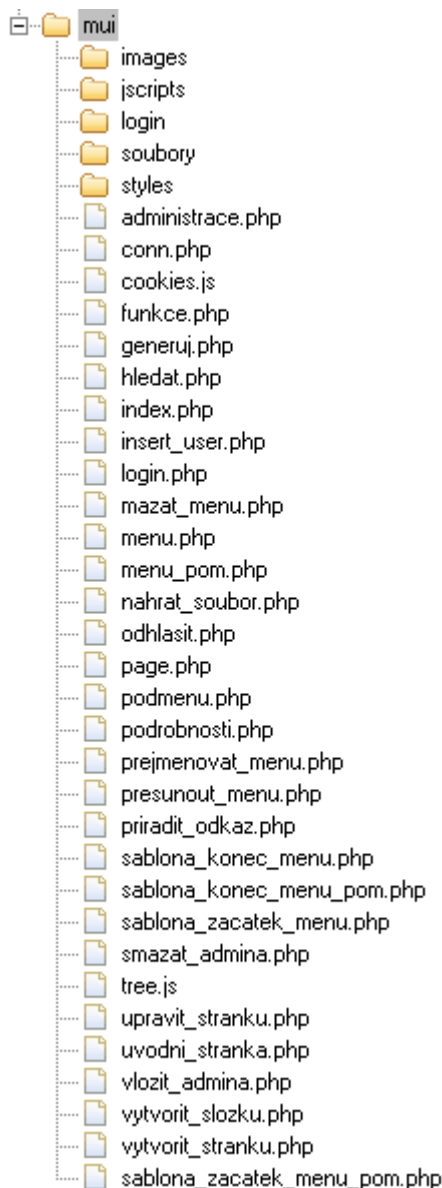
ID vyhledá článek v tabulce *clanky*. Sloupec *typ* je celočíselného formátu a určuje, o jaký druh souboru se jedná nebo zda jde o složku. Poslední dva sloupce *poradi* a *uroven* jsou opět celočíselné a pomocí nich určíme na jakém pořadí a v jaké hloubce zanoření bude daný prvek obsažen v menu.

4 ADMINISTRÁTORSKÉ ROZHRANÍ – CMS

Nyní jsou po ruce důležité stavební kameny: hotový design, fungující menu a databáze. Zbývá naprogramování skriptů, které budou obsluhovat automatickou správu obsahu webu. Úkol je to ze všeho nejobtížnější a je proto velmi vhodné rozdělit celou úlohu do vícera skriptů.

4.1 Struktura celého systému

Pro lepší pochopení je zde nejprve uvedena stromová struktura celého webu, nebo alespoň nejdůležitější součásti:



Obr. 5 – Stromová struktura webu

- *styles* – Adresář obsahující kaskádové styly.
- *soubory* – Adresář pro ukládání nahraných souborů.
- *jscripts* – Obsahuje soubory nutné k funkčnosti editoru TinyMCE.
- *images* – Adresář s obrázky určenými pro design.
- *administrace.php* – Ošetření, zda je přihlášen administrátor.
- *conn.php* – Připojení k databázi MySQL.
- *cookies.js* – Několik funkcí, které uchovávají stav rozbalení menu při aktualizaci nebo přesunu na jinou stránku.
- *funkce.php* – Soubor vytvořených funkcí PHP používaných ostatními skripty.
- *generuj.php* – Velmi důležitý skript, který je volán pokaždé, dojde-li ke změně v databázi. Dokáže vygenerovat nové menu obsahující aktuální soubory a složky. Ke generování používá informace z fixních, již vytvořených šablon a jednotlivé položky získává z databáze.
- *hledat.php* – Zatím neimplementovaná funkce pro fulltextové vyhledávání v článcích, připravena pro budoucí rozšíření.
- *index.php* – Základní interaktivní skript, který třídí informace, které má zobrazit dle požadavků uživatele. Užívá k tomu tzv. *superglobálních* proměnných v PHP. Dále tyto informace vkládá na správné místo a pošle je prohlížeči, který je následně zobrazí.
- *insert_user.php* – Pomocný skript k uložení administrátora do databáze.
- *login.php* – Přihlašovací skript.
- *mazat_menu.php* – Skript slouží k mazání souborů či celých adresářů v menu. Může, ale nemusí také smazat přiřazené články a soubory.
- *menu.php* – Menu vygenerované skriptem *generuj.php*, které je vloženo do stránky *index.php*.
- *menu_pom.php* – Pomocné menu generované při administrátorských úpravách.

- *nahrat_soubor.php* – Pomocí tohoto skriptu může administrátor vkládat soubory na web na určené místo.
- *odhlasit.php* – Odhlašovací skript.
- *page.php* – Základní stránka pro vyhledání článků. Vytahuje požadované texty z databáze a includeuje je do skriptu *index.php*.
- *podmenu.php* – Má využití pouze pro administrátora a nabízí mu vše, co může se stránkami dělat. Vkládání, mazání, editace článků, manipulace se složkami a soubory a mnoho dalších.
- *podrobnosti.php* – Zatím nevyužitý skript pro budoucí možná rozšíření.
- *prejmenovat_menu.php* – Skript slouží k přejmenování položek v menu.
- *presunout_menu.php* – k přesouvání položek i celých adresářů v menu.
- *priradit_odkaz.php* – Přiřadí odkaz již vytvořené stránce.
- *šablony* – Slouží ke generování menu.
- *smazat_admina.php* – Smaže vytvořeného administrátora.
- *tree.js* – JavaScriptové funkce pro rozbalování stromové adresářové struktury menu. Obsahuje navíc funkce pro možná rozšíření.
- *upravit_stranku.php* – Edituje již vytvořený článek. Není-li vytvořen, vytvoří jej.
- *uvodni_stranka.php* – Editace úvodní stránky.
- *vlozit_admina.php* – Umožňuje vložit nového administrátora pomocí jména a hesla.
- *vytvorit_slozku.php* – Vytvoří novou složku v rozbalovacím menu.
- *vytvorit_stranku.php* – Vytvoří novou stránku v menu a umožňuje k ní rovnou vytvořit článek, nebo jej editovat později skriptem *upravit_stranku.php*

4.2 Přihlašování a autorizace

Jelikož se jedná o aplikaci, která dělí uživatele pouze na administrátory a ty ostatní, není potřeba, aby se musel přihlašovat každý návštěvník. Přihlášení je tedy určeno pouze pro administrátora (může jich být i více), který se na přihlašovací stránku dostane přidáním

postfixu */login.php* (stačí i pouze */login*) k URL stránky. K přihlášení a kontrole administrátorů je použit formulář s výzvou jména a hesla. Přihlášený administrátor je pak uchován pomocí *sessions* v PHP a jsou mu zpřístupněny veškeré možnosti úpravy stránek.

4.2.1 Bezpečnost

Jednotlivé stránky a skripty na webu jsou dvojího druhu: privátní a veřejné. Privátní obsahují navíc kontroly pomocí *sessions*, při kterých zjistíme, zda má uživatel oprávnění vstupu. Pokud nemá, je ihned přesměrován na úvodní stránku. Na začátku každého skriptu, který je nutno vložit do stránky *index.php* je také kontrola, neboť skript nemůže být spuštěn bez načtení této stránky. Opět je návštěvník přesměrován na úvodní stránku. Třetím stupněm ochrany je zabránění otevření některých adresářů na webu.

4.2.2 Odhlášení

Princip odhlašování je při použití *sessions* jednoduchý, stačí jen smazat obsah proměnné, která uchovává přihlášeného administrátora a následně přesměrovat na úvodní stránku.

4.3 Přístup k databázi

K databázi je nutno být při každém dotazu přihlášen. To umožňuje skript *conn.php*, který obsahuje nadefinované konstanty, pomocí se kterých pak přihlašovací příkaz k vytvořené databázi připojí. Nepovede-li se přihlásit, je pouze vypsáno varovné hlášení. Konstanty jsou použity z důvodu umožnění připojit se k různým databázím na různých serverech, což zabrání neustálému přepisování tohoto skriptu.

4.4 Generování menu

Menu je připojeno ke každé stránce, proto je vhodné mít na serveru vytvořen soubor, který toto menu obsahuje. Druhou možností by bylo načítat menu z databáze, ale to by kvůli četnosti jeho použití nebylo příliš efektivní a také by bylo méně spolehlivé. Kvůli synchronizaci položek ve vytvořeném menu s položkami databáze je však nutností mít skript, který při každé změně menu tabulky v databázi aktualizuje soubor s vytvořeným menu. Tento skript je *generuj.php*. Využívá předdefinovaných šablon, které vpisuje do souboru *menu.php* a načítá z databáze postupně jednotlivé údaje, ze kterých podle typu

souboru a umístění vytvoří a zapíše novou položku do souboru *menu.php*. Na konci skriptu je tento soubor uložen a uzavřen.

4.5 Výběr článků

Články, které jsou uloženy v databázi, jsou dostupné pod odkazem, jež obsahuje identifikátor článku. Tento identifikátor slouží k rychlému vyhledání v databázi. v praxi to funguje tak, že při kliknutí na odkaz na libovolný článek je zavolán skript *page.php*, který se připojí k databázi a pomocí jedinečného ID a odkazu vyhledá název a text článku a následně jej dynamicky vloží do stránky *index.php*. Odkaz na stránku tedy může vypadat jako */index.php?page=363*, což říká, že má být z databáze vybrán článek s ID 363. Zde je na rozdíl od menu výběr z databáze namísto, neboť články jsou vyhledávány a vybírány pouze po jednom a není nutno generovat žádné soubory. Výpis nadpisu a textu z databáze je zobrazen v nezměněné podobě tak, jak byl uložen.

4.6 Obslužné skripty

K vytváření webových stránek bez zásahu do HTML kódu je potřeba několika skriptů, kde každý obsluhuje něco jiného a dohromady jsou schopny správy celého webu. v příloze je uveden manuál k práci s těmito skripty a zde je uveden popis, jak fungují.

Na začátku všech jsou bezpečnostní kontroly s případným přesměrováním a na konci je vždy zavolán skript *generuj.php*, který aktualizuje pozměněná data.

4.6.1 Vytvořit složku

Skript, který má za úkol vytvořit v menu novou složku pracuje tak, jako ostatní skripty na principu formulářů. Zadáním názvu a umístění je ihned vytvořena složka a zařazena do databáze i do menu. Formulář se skládá z jednoduchého pole pro název a z *radiobuttonů* (přepínací tlačítka, umožňující vybrat jen jednu z nabízených možností), které slouží k výběru umístění složky. Radiobuttony jsou dynamicky připojeny ke každému prvku rozbalovacího menu, které je vytvořeno z tabulky menu v databázi. Před samotným odesláním formuláře je JavaScriptem zkontrolováno, zda byl zadán název, neboť složka bez názvu jaksí postrádá smysl.

Po odeslání dat je proveden test, zda nebyl formulář poslán dvakrát, což se může stát například při aktualizaci stránky. Dále jsou poslané hodnoty z formuláře upraveny na požadovaný tvar tak, aby je bylo vhodné uložit do databáze. Je možné, i když nepravděpodobné, že bude někdy potřeba mít složky či soubory se stejným názvem. Proto bylo nezbytné provést ošetření, při kterém se pomocnému názvu *xnazev* do databáze bude ukládat hodnota s přidaným pořadovým číslem. Poté co byly vytvořeny všechny důležité parametry, je do tabulky vložen nový řádek, obsahující veškeré potřebné údaje.

4.6.2 Vytvořit stránku

Vytváření stránky probíhá ve dvou krocích. První krok je podobný na vytváření složky, je zadán název a umístění odkazu na stránku a výsledek je uložen do databáze. v druhém kroku je otevřen textový editor *TinyMCE*, ve kterém lze vytvářet stránku a zároveň ji formátovat. Pokud je vytvořený text správně naformátován, dojde k uložení článku. Pokud dojde k chybě, není článek uložen, ale stránka je i přesto vytvořena jako prázdná a lze ji tedy kdykoli později upravit.

4.6.3 Úvodní stránka

Úvodní stránka je vytvářena v tabulce *clanky* zvlášť, má fixní odkaz a je vytvářena a editována pomocí editoru *TinyMCE*.

4.6.4 Nahrát soubor

Nahrávání souborů na web funguje tak, že je zadán název odkazu (ten bude v menu zobrazen), vybrán soubor z počítače a opět je pomocí radiobuttonů vybráno umístění tohoto souboru v hierarchii menu. Pokud vyplněný formulář projde JavaScriptovou kontrolou, je odeslán ke zpracování na serveru. Tam jsou hodnoty z formuláře vyhodnoceny a soubor je nakopírován do složky soubory na serveru. Musí zde probíhat kontrola duplicity názvů souboru, jinak by docházelo k jejich přepsání. Proto je k souborům se stejným názvem přidávána číselná koncovka „*soubor(2).pdf*“. Skript automaticky detekuje typ souboru, a ten poté uvede do databáze, díky tomu je vybrána ikonka, jež bude u souboru zobrazena.

Aplikace je již od začátku vyvíjena tak, aby byla co nejnázev rozšiřitelná, a v současné době umí rozeznat 15 typů souborů a podle toho je třídí do 9 různých skupin. Neznámým

souborům je však taky přiřazena speciální ikonka, zejména kvůli tomu, aby byla odlišena od ikony stránky. [4]



Obr. 6 – Ikony typů souborů

4.6.5 Mazat

Skript *mazat_menu.php* slouží, jak již název napovídá, k mazání složek, stránek, souborů i článků. Pomocí dynamicky vytvořeného formuláře, který obsahuje položky menu s *checkboxy* (zaškrťovací políčka sloužící k výběru položek), jsou vybrány ty, které chceme smazat. Dále obsahuje nabídku, zda se má s odkazem smazat také článek či soubor. Implicitně je zaškrtnuto, že ano. Před odesláním je JavaScriptem zobrazen potvrzovací dotaz, kvůli omylnému smazání a po potvrzení je formulář poslán na server ke zpracování. Nejprve je vytvořen seznam, do kterého jsou v cyklech přidány položky ke smazání a jsou také přidány ty, které jsou uvnitř zaškrtnutých složek. Všechny prvky ve vytvořeném seznamu jsou postupně pomocí cyklů smazány z databáze, případně jsou smazány i články a soubory, které jsou jim přiřazeny.

4.6.6 Přesunout

Vygenerovaný formulář obsahuje jak *checkboxy* tak *radiobuttony* a s jejich pomocí vybíráme, co a kam budeme přesouvat. Kontroly na straně klienta, zda je formulář řádně vyplněn, zde nejsou třeba, ale na straně serveru je důležité ošetřit, zda se cílová a zdrojová složka neshodují. Pokud ano, nemůže být přesun proveden a skript vrátí stránku s JavaScriptovou chybovou hláškou. Pokud proběhne kontrola dobře, jsou upraveny veškeré položky v databázi, kterých přesun týká a na konci je vygenerováno aktualizované menu.

4.6.7 Přejmenovat

Přejmenování složky, souboru či odkazu je jednoduché, je pouze zadán nový název a vybrány položka z menu, kterou chceme přejmenovat. Opět probíhají kontroly na straně klienta i na straně serveru, které jsou schopny ošetřit nevyplnění formuláře i duplicitu názvů.

4.6.8 Přiřadit odkaz

Přiřazení odkazu slouží zejména k tomu, aby bylo možno odkazovat se z menu přímo na externí stránky, ale jako odkaz může sloužit i nějaká již vytvořená stránka. Odkaz lze přiřazovat pouze stránkám, ne složkám ani souborům. Tomu také odpovídá vygenerovaný formulář, který umožní vybrat pouze odkazy se stránkou či článkem. Při vyplnění povinné položky *odkaz* je ošetřeno, zda je zadán externí odkaz pomocí prefixu *www*. Pokud ano, může být automaticky doplněno *http://*, které způsobí uložení absolutního odkazu namísto relativního. Těsně před odesláním je ještě zobrazen potvrzovací dotaz, zda chceme pokračovat i přesto, že na stránce může být uložen článek. Pokud je, tak bude při potvrzení smazán podobným způsobem jako u *mazat_menu.php*. Odkaz bude v databázi také aktualizován.

4.6.9 Upravit stránku

Tento skript funguje v principu stejně jako skript pro vkládání stránky, jen nevytváří novou, ale edituje již vytvořenou stránku.

4.6.10 Vložit administrátora

Administrátorů, kteří spravují web, může být i více, proto je vhodné mít možnost je nějak vkládat. Samotné vkládání je jednoduché, po zadání jména a hesla (pro kontrolu musí být zadáno dvakrát) je do databáze uložen nový administrátor a jeho heslo je v databázi uloženo šifrovaně. Před vložením probíhá dvojitá kontrola formuláře, jedna na straně klienta a druhá na straně serveru.

4.6.11 Smazat administrátora

Pro smazání administrátora je nutno znát jeho jméno i heslo, a to z důvodu bezpečnosti. v budoucnu je možno skript rozšířit tak, že by měl jeden z administrátorů vyšší práva a mohl by ostatní smazat i bez nutnosti zadání hesla. Po ukončení skriptu je pomocí Javascriptu vypsána kontrolní hláška, zda byl daný administrátor nalezen v databázi a smazán či nikoliv.

4.7 WWW stránky online

Pro testování a vyzkoušení byly stránky umístěny na bezplatný server *Internet Centrum* (<http://ic.cz>) a jsou dostupné na adrese <http://mui.ic.cz>. Návod k užívání administrátorského rozhraní je uveden v příloze P I.

ZÁVĚR

Ve své bakalářské práci jsem se zaměřil na popis redakčních systémů (CMS). Shrnul jsem historii tvorby webových stránek a aplikací, jejichž vyvrcholením jsou dnes právě takové systémy. Dále jsem uvedl výčet několika hlavních technologií, které jsou využívány při tvorbě webu, a zaměřil jsem se zejména na ty, které jsem ve své práci použil. Jsou jimi XHTML, JavaScript, PHP, MySQL a CSS.

V praktické části jsem nejprve uvedl software, který jsem používal a vysvětlil tvorbu designu stránek. Dále jsem popsal naprogramování celé aplikace, přičemž jsem se snažil objasnit funkcionality všech částí a skriptů jednotlivě, a také při jejich vzájemné komunikaci, jenž je u větších projektů velmi důležitá.

Vytvořená aplikace umožňuje rychle a efektivně editovat jakékoliv informace, což bude jistě velkou oporou při častějších aktualizacích. Vytvořené stránky by měly být přínosem pro výuku předmětu Metody umělé inteligence. Měly by pomoci usnadnit práci studentům při vyhledávání informací, a také práci administrátorům, neboť díky způsobu vytváření menu a článků je nyní tvorba celého webu jednoduchou a rychlou záležitostí.

ZÁVĚR V ANGLIČTINĚ

In my bachelor project I focused on describing content management systems (CMS). I summarized the history of web sites production including the systems, which are nowadays on the top. Then I presented a list of some main technologies used for website production and I especially focused on those, which I had used in my project. Those are XHTML, JavaScript, PHP, MySQL and CSS.

First I named software I had used in the practical part and I explained the creation of web design. Thereafter I described the programming of whole application, whereas I tried to make clear the functionality of all parts of the scripts individually as well as in intercommunication, which is very important for larger projects.

Completed application makes quick and effective editing of articles possible, which should be great support for common updating. Created web sites should be benefit for education in Methods of artificial intelligence subject. It should facilitate students collecting information. It should also help administrators, because thanks to the way of generating the menus and articles creating of whole web is very easy and very quick business.

SEZNAM POUŽITÉ LITERATURY

- [1] Wikipedia – Internetová encyklopedie [online]. [cit. 2008-04-24]. Dostupný z WWW: <http://wikipedia.org>
- [2] Interval – Webdesign a e-komerce denně [online]. [cit. 2008-04-24]. Dostupný z WWW: <http://interval.cz/>
- [3] Programujte – Zaměřeno na informační technologie [online]. [cit. 2008-04-24]. Dostupný z WWW: <http://programujte.com/>
- [4] Schruman E. M., Pardi W. J., Dynamické HTML v akci, Computer Press, Praha, 2001, ISBN: 807226401X
- [5] TinyMCE – JavaScript kontent Editor [online]. [cit. 2008-04-24]. Dostupný z WWW: <http://tinymce.moxiecode.com/>
- [6] Castagnetto J. a kol., PHP Programujeme profesionálně - 2. opravené vydání, Computer Press, Praha, 2001, ISBN: 8072263102
- [7] Welling L., PHP a MySQL, SoftPress, Praha, 2004, ISBN 8086497607
- [8] Mach J., PHP pro úplné začátečníky, Computer Press, Brno, 2003, ISBN 8072268341
- [9] Hall M., Java servlety a stránky JSP, Neocortex, Praha, 2001, ISBN 8086330060
- [10] Šimůnek M., SQL – kompletní kapesní průvodce, Grada, Praha, 1999, ISBN 8071696927
- [11] Snížek M., CSS pro zelenáče, Neocortex, Praha, 2004, ISBN 8086330141

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ASP	Active Server Pages
CERN	Conseil Européen pour la Recherche Nucléaire
CMS	Content Management System
DOM	Document Object Model
DTD	Document Type Definition
FTP	File Transfer Protocol
GPL	GNU General Public License
HTML	HyperText Markup Language
JSP	Java Server Pages
MathML	Mathematical Markup Language
MD5	Message-Digest algorithm
MSIE	MicroSoft Internet Explorer
MUI	Metody Umělé Inteligence
PHP	Hypertext Preprocessor (Personal Home Page)
PNG	Portable Network Graphics
SQL	Structured Query Language
SVG	Scalable Vector Graphics
URL	Uniform Resource Locator
UTF-8	UCS Transformation Format
W3C	World Wide Web Consortium
WWW	World Wide Web
WYSIWYG	What You See Is What You Get
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language

SEZNAM OBRÁZKŮ

Obr. 1 – TinyMCE editor.....	20
Obr. 2 – Hlavička webové stránky.....	27
Obr. 3 – Klasické rozložení stránky.....	28
Obr. 4 – Menu.....	29
Obr. 5 – Stromová struktura webu.....	32
Obr. 6 – Ikony typu souborů	38

SEZNAM PŘÍLOH

- P I Manuál k administrátorskému rozhraní
- P II CD obsahující soubory se zdrojovými kódy www stránek, použité obrázky a dokumentaci v elektronické podobě.

PŘÍLOHA P I: MANUÁL K ADMINISTRÁTORSKÉMU ROZHRANÍ

1. Přihlašování

K přihlášení administrátora je nutno přidat k URL stránky postfix */login.php* nebo */login*.

Př. URL stránek: *http://mui.ic.cz*

URL přihlašovacího skriptu: *http://mui.ic.cz/login.php* nebo *http://mui.ic.cz/login*

Ke správnému přihlášení je důležité znát jméno a heslo administrátora.



Přihlašovací formulář pro administrátora

Po úspěšném přihlášení je zpřístupněno administrátorské rozhraní obsahující nabídku akcí, které umožňují kompletní správu celého webu.



Nabídka akcí pro správu webu

2. Vytváření složek v menu

Vkládání nových složek se provádí pomocí formuláře, který obsahuje:

- **Název** – Zde se zadává název nové složky
- **Výběr umístění** – Pomocí rozbalovacího menu a přepínacích tlačítek je vybráno místo, kam bude nově vytvořená složka umístěna.
- **Směr** – Směr umístění je přednastaven na „Do složky“, což znamená, je-li pomocí přepínacích tlačítek vybrána složka, je nová složka vložena na první pozici do vybrané složky. Je-li však vybrán směr „Pod složku“, bude nová složka vložena přímo za vybranou složku. Je-li vybrán místo složky odkaz či soubor, pak nezáleží na výběru směru a nová složka bude umístěna vždy přímo za vybraný odkaz nebo soubor.

The image displays two side-by-side screenshots of a web form for creating a folder. Both screenshots have a light blue background. The left screenshot shows the 'Název:' text box containing 'Nová složka'. Below it is a list of menu items, each with a radio button: 'Bakalářská práce', 'Nápověda' (selected), 'Administrace', and 'Funkce'. At the bottom, the 'Směr:' section has two radio buttons: 'Pod složku' and 'Do složky' (selected). A 'Vytvořit' button is at the bottom. The right screenshot shows the 'Název:' text box empty. The list of menu items is the same, but 'Nová složka' is selected. The 'Směr:' section has 'Pod složku' selected. A 'Vytvořit' button is at the bottom.

Vkládání složky

3. Vytvoření stránek

Vytváření stránek probíhá ve dvou krocích. v prvním kroku je zadán název a je vybráno umístění stránky, a to naprosto stejným způsobem jako při vytváření složky. v kroku druhém je pomocí editoru TinyMCE umožněno napsat celý článek a formátovat ho. Je zde také formulářové okénko, které slouží pro zadání nadpisu článku. Ten je přednastaven na

název odkazu v menu, ale je ho možné přejmenovat, aniž by to název odkazu ovlivnilo. Po uložení je článek ihned přístupný a je možno si ho prohlédnout kliknutím na vytvořený odkaz.

The screenshot shows a web editor interface with a light blue background. On the left, there is a sidebar with a 'Název:' field containing 'Pokusná stránka'. Below it are four radio buttons: the first is unselected, the second is selected and labeled 'Bakalářská práce', the third is unselected and labeled 'Nápověda', and the fourth is unselected and labeled 'Funkce'. Underneath is a 'Směr:' section with 'Pod složku:' selected and 'Do složky:' unselected. At the bottom of the sidebar is a 'Pokračovat' button. The main area on the right has a 'Název článku:' field with 'Pokusná stránka'. Below it is a rich text editor with a toolbar containing bold, italic, underline, list, and link icons, along with font and size dropdowns. The editor contains the text 'Tady je nějaký POKUSNÝ TEXT.' and 'Třeba i červený' in red. At the bottom of the main area is a 'Cesta: p » span' breadcrumb and 'Reset' and 'Uložit článek' buttons.

Vkládání stránky ve dvou krocích

The screenshot shows the same web editor interface as above, but with the 'Bakalářská práce' radio button selected. The 'Pokusná stránka' option is now visible in the list. The 'Směr:' section remains the same. The 'Pokračovat' button is at the bottom. To the right of the sidebar, a preview window shows the resulting page with the title 'Pokusná stránka' in green, the text 'Tady je nějaký POKUSNÝ TEXT.', and 'Třeba i červený' in red.

Vytvořený odkaz a článek

4. Úprava úvodní stránky

Úvodní stránka se vytváří stejným způsobem jako druhý krok při vytváření stránky. Formulář tedy obsahuje jen název úvodní stránky a textové pole, které lze opět upravovat pomocí editoru TinyMCE.

5. Nahrávání souborů na web

K nahrávání souborů slouží formulář, ve kterém vyplníme název odkazu, vybereme soubor z počítače, vybereme umístění odkazu, a ten je pak jednoduše nahrán na server. Je však umožněno nahrávat pouze soubory menší než 3 MB.

The image displays two side-by-side screenshots of a web file upload interface. Both screenshots show a form with the following elements: a 'Název odkazu:' text input field containing 'Nějaký soubor'; a 'Soubor:' text input field containing 'C:\Documents and Settings' and a 'Procházet...' button; a list of radio buttons for 'Soubory', 'Bakalářská práce', 'Nápověda', and 'Funkce', with 'Soubory' selected; a 'Směr:' section with radio buttons for 'Pod složku:' and 'Do složky:', with 'Do složky:' selected; and a 'Nahrát' button at the bottom. The right screenshot shows the 'Soubor:' field updated with 'Nějaký soubor' and the 'Soubory' radio button selected.

Nahrávání souboru do určené složky

6. Mazání stránek, souborů a adresářů

Mazání probíhá tak, že je pomocí zaškrťovacích políček vybrán jeden nebo více souborů, které chceme odstranit. Pokud vybereme složku, bude smazán také celý její obsah bez ohledu na to, zda byly vybrány i položky uvnitř. Formulář obsahuje také políčko „Smazat včetně článků a souborů“. Toto políčko je implicitně zaškrtnuto a slouží k tomu, aby byl smazán také článek nebo soubor, na který ukazuje odkaz, který chceme odstranit. Potvrzením kontrolní hlášky dojde ke smazání všech označených položek.



Smazání složky Soubory i s obsahem

7. Přesunování položek v menu

K přesouvání položek je určen formulář obsahující dva druhy zaškrtačacích políček. K vybírání položek, které se mají přesunout, slouží čtvercové políčka, kterých může být vybráno i více a k výběru umístění jsou pak určeny kulaté políčka, ze kterých je dovoleno vybrat pouze jedno. v případě složky je opět možno vybrat směr, aby bylo odlišeno, zda chceme přesouvat do složky nebo pod složku. Přesouvat složku sama do sebe je však z principiálních důvodů zakázáno.



Ukázka přesunutí složky na jiné místo

8. Přejmenování položek v menu

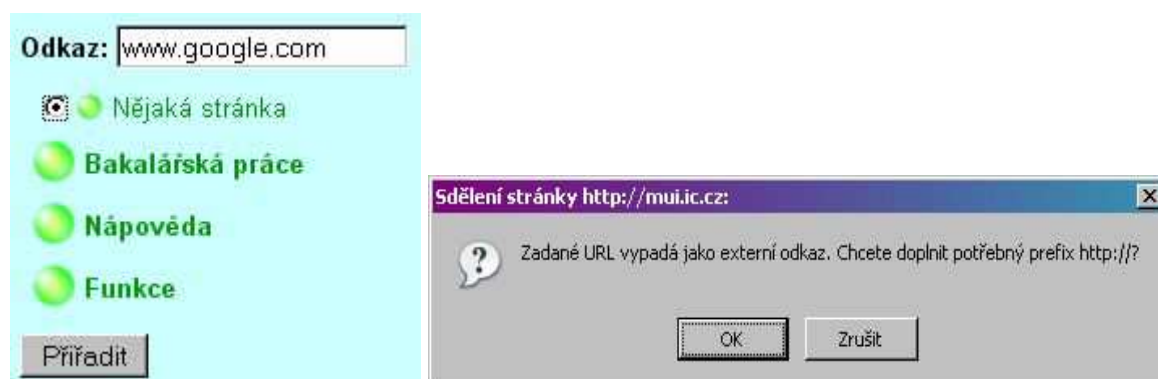
Přejmenování položek může být důležité, pokud chceme aktualizovat název nějaké položky v menu nebo zjistíme-li, že jsme někde napsali chybu a nechceme tuto položku smazat a znovu vytvářet. Stačí jen zadat nový název a vybrat položku, která má být změněna.



Přejmenování položky

9. Přiřazení odkazu položkám

Chceme-li mít v menu odkaz na nějaký již vytvořený článek nebo na nějakou externí stránku, slouží k tomu právě přiřazení odkazu. Potřebujeme k tomu znát jen URL adresu stránky. Tu zadáme do formulářového okénka *Odkaz* a vybereme položku, které bude odkaz přidělen. Pokud bude URL začínat `www`, skript detekuje, že se jedná o externí odkaz a bude tedy k odkazu přidán prefix `http://`. Pokud se na vybrané stránce bude nacházet nějaký článek, skript podá varovné hlášení a po jeho potvrzení bude tento článek smazán.

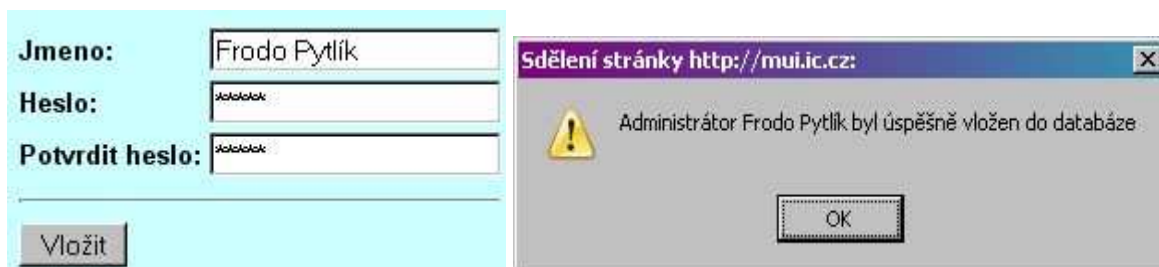


10. Editace již vytvořených článků

Editace článků umožňuje přístup k již vytvořeným článkům a jejich rychlou úpravu. Funguje na podobném principu jako vytváření článků. v prvním kroku nejprve vybereme článek, který chceme upravit a v druhém kroku, kdy je tento článek otevřen v editoru TinyMCE, jej můžeme libovolně upravit. Tlačítkem *Reset* jsou smazány veškeré změny a článek je obnoven tak, jak byl naposled uložen. Pokud v prvním kroku vybereme odkaz, který ukazuje například na externí stránky, vrátí skript hlášení o tom, že tuto stránku nelze editovat.

11. Vložení nového administrátora

Pro vložení administrátora musíme zadat jméno a heslo, které musí být pro kontrolu ještě potvrzeno. Jméno administrátora, jak je vidět na obrázcích, může být s diakritikou i s mezerami, ale musí být jedinečné. Není tedy možné vkládat dva administrátory se stejným jménem.



The image shows two parts of a web interface. On the left is a form for creating a new administrator. It has three input fields: 'Jmeno:' containing 'Frodo Pytlík', 'Heslo:' containing 'skokskok', and 'Potvrdit heslo:' containing 'skokskok'. Below the fields is a button labeled 'Vložit'. On the right is a dialog box titled 'Sdělení stránky http://mui.ic.cz:'. It contains a yellow warning icon and the text 'Administrátor Frodo Pytlík byl úspěšně vložen do databáze'. There is an 'OK' button at the bottom of the dialog.

Vložení nového administrátora

12. Smazání administrátora

Pro smazání je také třeba znát jméno i heslo, a to především z bezpečnostních důvodů. Administrátor tak nemůže smazat jiného administrátora bez jeho svolení. Pokud tedy nezná jeho heslo. Před definitivním smazáním je ještě potřeba potvrdit varovnou hlášku, která nás může upozornit na neopatrné smazání.



The image shows a form for deleting an administrator. It has two input fields: 'Jmeno:' containing 'Frodo Pytlík' and 'Heslo:' containing 'skokskok'. Below the fields is a button labeled 'Smazat'.

Smazání administrátora