

Obsah

Zoznam použitých skratiek	4
Anotácia	5
Annotation	5
1. Úvod	6
2. Čo je PHP	7
2.1 História	7
2.2 Prečo PHP	8
3. Základy jazyka PHP	10
3.1 Premenné	10
3.2 Konštanty	11
3.3 Príkazy	11
3.3.1 Vetvenie programu	12
3.3.2 Cykly	14
3.4 Polia	15
4. PHP a databáza MySQL	17
4.1 Spojenie s MySQL	17
4.2 Príkazy	18
5. Realizácia	21
5.1 Administrátorská časť	21
5.2 Užívateľská časť	23
6. Záver	27
7. Zoznam použitej literatúry	28
PodĎakovanie	29
8. Príloha - CD	30

Zoznam použitých skratiek

WWW – World Wide Web

HTML – Hyper Text Markup Language

PHP – Hypertext Preprocessor (pôvodne Personal Home Page)

SQL – Structured Query Language

API – Application Programming Interface

ODBC – Open Database Connectivity

HTTP – Hyper Text Transfer Protocol

Anotácia

Názov: Tvorba dynamických HTML stránok

Meno študenta: Zuzana Švančarová

Vedúci semestrálneho projektu: Ing. Dalibor Puna

Rok odovzdania: máj 2006

Projekt sa zaoberá vytvorením dynamickej HTML stránky, ktorá je založená na programovacom jazyku PHP a databáze MySQL.

Annotation

Title: Creating dynamic HTML web sites

Author: Zuzana Švančarová

Supervisor: Ing. Dalibor Puna

Year submission: May 2006

1. Úvod

V dnešnom, modernom svete už asi každý pozná a pravdepodobne aj používa internet. Stačí si sadnúť k počítaču, ktorý je samozrejme „online“, napísať www..... a môžete si napr. prečítať správy, zaplatiť faktúru za mobilný telefón, napísať priateľovi do Austrálie atď. a to všetko za pár sekúnd. Internet je najrýchlejší systém s množstvom informácií a každý si v ňom nájde, čo potrebuje.

V posledných rokoch internet prenikol naozaj všade, vrátane školstva. Každá väčšia škola, či fakulta má svoju internetovú stránku. Preto sa budeme zaoberať rozvojom a vylepšením aktuálnych aplikácií na webovej stránke našej fakulty. Budeme pracovať s databázou rozvrhu hodín. Pokúsime sa zjednodušiť vytváranie nového rozvrhu pomocou databázy starého a uľahčiť tak prácu užívateľom (profesorom).

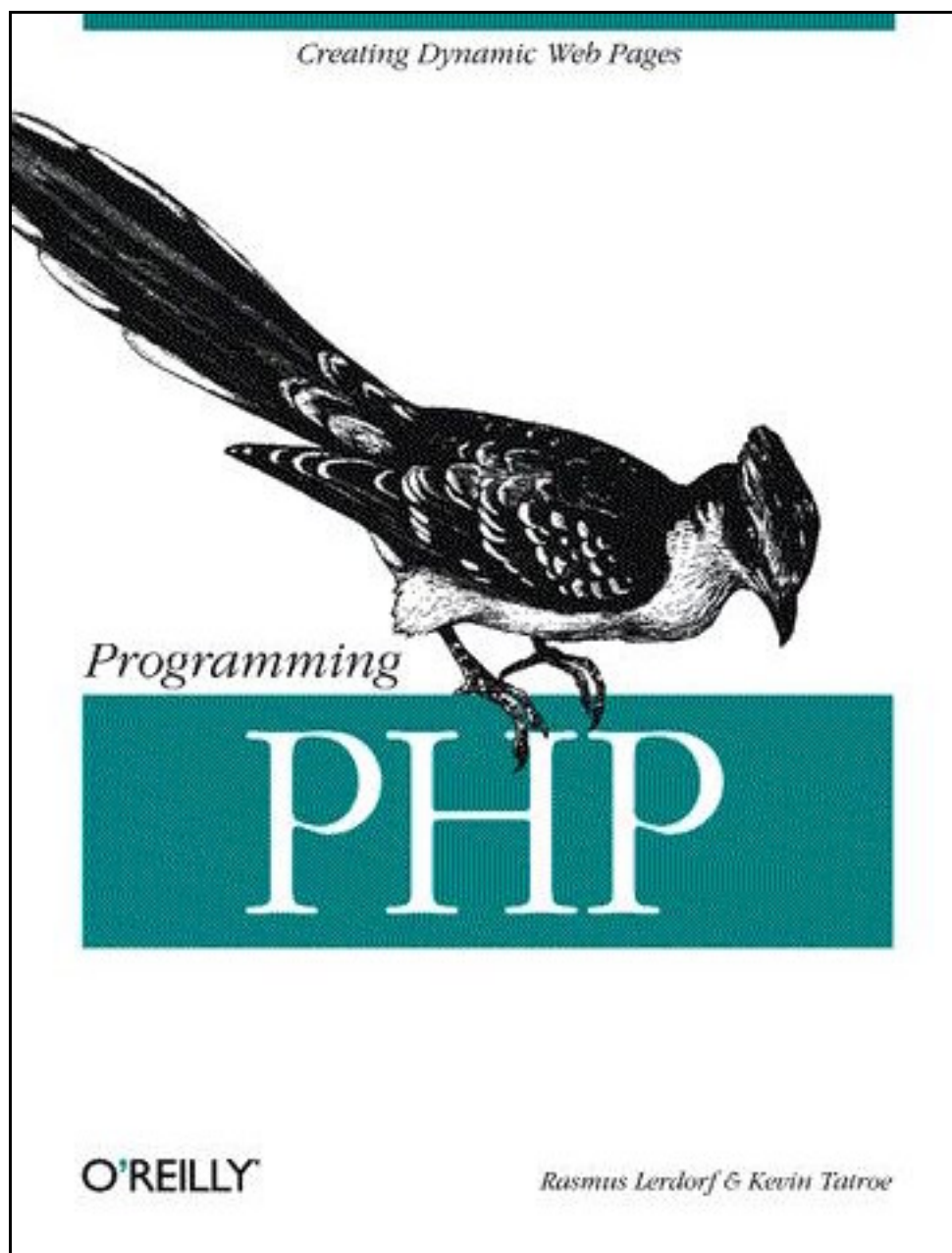
Najskôr sa oboznámime so základmi jazyka PHP a databázou MySQL, v druhej časti samotnou tvorbou aplikácie. Program bude mať dve časti a to administrátorskú a užívateľskú časť. V administrátorskej časti sa budeme zaoberať tvorbou aplikácie pre výber uloženia aktuálnej databázy rozvrhu do záložnej databázy buď pre letný alebo zimný semester. V užívateľskej časti budeme pracovať s touto záložnou databázou, pretože aktuálna databáza rozvrhu bude zmazaná a voľná pre nový rozvrh. Vytvoríme PHP kód, pomocou ktorého prepíšeme uložené záznamy starého rozvrhu na nový rozvrh. Tento kód bude včlenený do už existujúcej aplikácie na tvorbu rozvrhu vo forme formulára. Doteraz sa tento formulár musel vždy nanovo vypisovať pre každý predmet aj keď sa príliš nezmenil. Naš program použije staré záznamy, ktoré stačí vo formulári iba upraviť alebo ich nezmenené uloží do nového aktuálneho rozvrhu.

2. Čo je to PHP

PHP je skriptovací jazyk, ktorý umožňuje ľahko a rýchlo vytvoriť www stránky, ktoré sa generujú dynamicky, napríklad z databázy či súboru. PHP je mocným nástrojom na tvorbu moderného webu, umožňuje naprogramovať široké portfólio internetových aplikácií - a to ľahko a rýchlo, nakoľko sa veľmi podobá na jazyk C.

2.1 História

PHP sa zrodilo niekedy v roku 1994. Vznikol ako program v Perle, ktorý slúžil na evidenciu prístupov k www stránkam pána Rasmusa Lerdorfa, ktorý bol jeho prapôvodcom. Nakoľko však Perl ako interpretovaný jazyk priveľmi zaťažoval www server, autor svoj software prepísal do jazyka C. Postupne ho používalo čoraz viac ľudí. Systém bol neskôr rozšírený o požiadavky jeho používateľov a bol vydaný pod názvom „Personal Home Page“ Tools (pôvodný význam PHP). Po zabudovaní SQL modulu, ktorý umožňoval PHP spolupracovať s databázami a vytvárať formuláre, sa názov zmenil na Personal Home Page Form Interpreter (PHP/FI). Tento systém vznikol ako kompilát prvotného PHP a programu, ktorý sprístupňoval výsledky databázových dopytov na webe (Form Interpreter). To bolo v roku 1995. Neskôr sa začal tento systém používať na oveľa komplikovanejšie veci, a preto prešiel vývoj na skupinu programátorov. To bol začiatok PHP3. Skupinka programátorov (Rasmus Lerdorf, Andi Gutmans, Zeev Suraski, Stig Bakken, Shane Caraveo a Jim Winstead) zlepšila a rozšírila program a pridala jednoduché API, ktoré umožňuje voľnosť v pridávaní funkcií. Bola zdokonalená aj syntax jazyka. K dispozícii je dnes verzia PHP4 vyvíjaná pod kódovým názvom ZEND. Táto verzia je 8-10 krát rýchlejšia ako PHP3 a disponuje rôznymi chuťovkami v štýle jazyka C, resp. C++.



Obr. 2.1 Prvá publikácia PHP

2.2 Prečo PHP

Mnoho webových sídiel má statický obsah. Stránkami týchto sídiel sú rôzne dokumenty alebo články, ktoré sa skladajú z jednoduchého textu, obrázkov a odkazov na iné dokumenty. Tu sa stačí držať HTML ako overenej a spoľahlivej metódy tvorby webových stránok. Stále viac sa začínajú používať na internete

aplikácie, ktoré sú dynamické, pretože ich obsah sa mení podľa dát a akcií užívateľa. PHP pracuje vo vnútri dokumentu HTML a požičiava mu tak schopnosť generovania požadovaného obsahu.

HTML

Súbor, ktorý má príponu htm alebo html, je považovaný za internetovú stránku. Je napísaný v jazyku HTML, čo je dnes najrozšírenejší značkový jazyk. HTML kód je kombináciou normálneho textu a značiek. Zmysel HTML značiek je v oddeľovaní a vyznačovaní častí textu. Značkám sa tiež hovorí tagy. Podrobnosťami sa v tomto projekte nebudeme zaoberať.

PHP verzus HTML

Rozdiel medzi PHP a HTML je v tom, ako s nimi zaobchádza server.

HTML stránky: Keď príde požiadavka na stránku od prehliadača, webový server prevedie tieto kroky: prečíta požiadavku prehliadača, nájde stránku na serveri a odošle stránku cez internet späť prehliadaču.

PHP stránky: Namiesto prenosu statickej stránky HTML chceme, aby server urobil určité rozhodnutia a vytvoril stránku, ktorá odpovedá nášmu kódu PHP. Preto server vykoná nasledujúce kroky: prečíta požiadavku prehliadača, nájde stránku na serveri, interpretuje kód PHP, upraví stránku a odošle stránku cez internet späť prehliadaču.

Niekoľko príkladov, ktoré nie sú pomocou HTML dostupné:

- vytvorenie stránok prispôbených pre zobrazenie iba tých vecí, ktoré budú konkrétnemu užívateľovi zaujímať
- zobrazenie a aktualizácia databáz na webovej stránke
- získavanie spätnej väzby od užívateľa

3. Základy jazyka PHP

Kód PHP je uložený ako čistý text vo formáte ASCII, takže ho môžeme písať vo väčšine textových editorov napr. Notepad++ vo Windows.

Ako sme už spomenuli v predchádzajúcej kapitole, kód PHP je vlastne obyčajná www stránka doplnená o špeciálne príkazy. Aby server vedel, že sa nejedná len o obyčajnú www stránku, používajú sa odlišné prípony. Stránka v jazyku PHP by mala mať príponu .php alebo .phtml. So súbormi s takouto koncovkou sa zachádza odlišne. Je v nich vyhľadávaný kód PHP. Webový server začína v režime HTML. Do režimu PHP prejde až v okamihu, keď sa stretne so značkou PHP, ktorá sa používa na opustenie HTML. Existuje viacero spôsobov ako to urobiť, my spomenieme aspoň jeden:

```
<HTML>
<?php
    echo ("Text generovaný PHP");
?>
</HTML>
```

Server HTTP vie, že keď sa stretne s týmto mechanizmom, má začať spracovávať kód PHP. Odošle výstup, v tomto prípade generovaný príkazom echo, prehliadaču. Keď je dosiahnutý koniec kódu (?>), server sa vráti do režimu HTML a pokračuje v odosielaní dokumentu.

3.1 Premenné

Všetky programovacie jazyky aj PHP nám umožňujú uložiť časti dát do premenných a potom k nim pristupovať pomocou zápisu názvu premennej. Názvy všetkých premenných musia začínať znakom \$. V jazyku PHP nemusíme naše premenné definovať a uvádzať typ dát, ktoré budú obsahovať.

Napr.:

```
$meno = "Zuzana";
echo ($meno);
```

- vypíše „Zuzana“

PHP má tri základné dátové typy: celé čísla (integer), desatinné čísla (double) a reťazce (string). Okrem nich existujú ďalšie napr. polia (array). Tento dátový typ je určený automaticky hodnotou, ktorú premennej priradíme a môže sa meniť počas programu podľa toho, ako sa mení jej hodnota.

```
$x = 5;           // $x je celé číslo  
$x = 4.2;        // $x je desatinné číslo  
$x = "ABCD";     // $x je reťazec
```

3.2 Konštanty

Sú to hodnoty, ktoré sa nikdy nemenia. Na rozdiel od premenných sa názvy konštánt nezačínajú znakom \$. V podstate existujú dva druhy: literálne a symbolické. Literálne konštanty sú jednoducho nemenné hodnoty. Symbolické konštanty sú vyjadrované pomocou priradenia identifikátora k danej hodnote. Pre definovanie konštánt sa používa funkcia `define()`:

```
define ("SKOLA", "Fakulta chemickej a potravinárskej  
technológie");  
echo ("Moje pôsobisko je" . SKOLA);
```

Takto môžeme nadefinovať konštantu SKOLA s hodnotou „Fakulta chemickej a potravinárskej technológie“ a používať ju ako odkaz na názov fakulty v celom programe. V prípade zmeny, stačí prepísať len jedinú hodnotu a to v mieste, kde bola definovaná.

3.3 Príkazy

Príkazy, podmienky a cykly dávajú programom schopnosť rozhodovania, tvoria kostru celej aplikácie.

Príkazy na vetvenie programov

- príkaz `if`
- príkaz `switch`

Cykly

- cyklus while
- cyklus for

3.3.1 Vetvenie programu

PHP ponúka dva príkazy, ktoré umožňujú vykonávať určitú časť skriptu podmienene. Slúžia na to príkazy `if` a `switch` v rôznych variáciách.

Príkazy `if`

Najjednoduchšia verzia použitia vetviaceho príkazu `if` je daná nasledujúcou syntaxou: *if (výraz) príkaz ;*

Častejšie sa však stretávame so situáciou ak pri splnení podmienky potrebujeme vykonať činnosť A a pri nesplnení činnosť B. Nato slúži výborne príkaz `if-else`, ktorý má nasledujúcu syntax:

```
if ( výraz )
    príkaz_A ;
else
    príkaz_B ;
```

Namiesto príkazov `príkaz_A` a `príkaz_B` môžeme použiť zložený príkaz.

Ak ani to nestačí, môžeme ešte použiť príkaz `elseif`. V prípade nesplnenia hlavnej podmienky nám umožňuje testovať alternatívne podmienky. Príkaz uvedený za `elseif` sa vykoná, ak nie je splnená podmienka pre `if` a zároveň je splnená podmienka za `elseif`. Časť `elseif` môžeme použiť aj viackrát. Príkaz `else` sa vykoná len vtedy, ak nie je splnená žiadna podmienka `elseif`.

Príklady:

Vydelíme premennú `$delenec` premennou `$delitel`, ale len v prípade, že premenná `$delitel` sa nerovná nule, aby nedošlo k deleniu nulou.

```
if ( $delitel !=0 ) $podiel = $delenec / $delitel ;
```

Porovnáme premenné \$a a \$b a podľa toho, ktorá podmienka bude splnená, vypíše sa príslušná hláška.

```
if ( $a < $b )
    echo ("a je menšie ako b");
elseif ($a == $b)
    echo ("a a b sa rovnajú");
else
    echo ("a je väčšie ako b");
```

Príkaz switch

V prípade, že máme viac podmienkových vetiev, oplatí sa používať príkaz switch. Jeho syntax je nasledovná:

```
switch ( výraz1 )
{
    case HODNOTA A :
        príkazA1;
        príkazA2;
        .....
        break;
    case HODNOTA B :
        príkaz B1;
        príkaz B2;
        ...
        break;
    .....
    default:
        príkaz XXX1;
}
```

Najprv sa vyhodnotí výraz1 a potom sú postupne prechádzané hodnoty za case, až kým sa nenájde hodnota, ktorá sa rovná výrazu1. Následne sa vykonávajú príkazy, až kým sa nenarazí na príkaz break alebo koniec príkazu switch. Časť default sa vykoná vtedy, ak ani jedna hodnota sa nerovná výrazu1.

```
switch ($CisloDna)
{
    case "1" : echo ("Pondelok");
               break;
    case "2" : echo ("Utorok");
               break;
    case "3" : echo ("Streda");
               break;
    case "4" : echo ("Štvrtok");
               break;
    case "5" : echo ("Piatok");
               break;

    default: echo ("Víkend");
}
```

3.3.2 Cykly

Sú určené k opakovanému vykonávaniu bloku kódu v zadanom počte alebo do splnenia podmienky.

Príkaz while

Tento iteračný príkaz testuje podmienky cyklu ešte pred prechodom cyklu, čo znamená že cyklus nemusí prebehnúť ani raz. Príkaz while používame, ak závisí ukončovacia podmienka na nejakom príkaze v tele cyklu. Zadaný cyklus sa teda prevádza len vtedy, ak je splnená podmienka na začiatku, preto sa mu aj hovorí cyklus s podmienkou na začiatku.

while (výraz) príkaz ;

Príkaz do-while

Podmienka sa tu testuje až po prechode cyklom, teda cyklus prebehne najmenej raz a hovorí sa mu cyklus s podmienkou na konci.

```
do
{
    príkazy;
} while ( výraz ) ;
```

Príkaz for

Je príkaz cyklu, ktorý použijeme v prípade, že je dopredu známy počet prechodov cyklom. Jeho syntax je nasledovná:

```
for (výraz1 ; výraz2; výraz3) príkaz;
```

Výraz1 je vyhodnotený pred začiatkom prevádzania cyklu. Potom je vyhodnotený výraz2. Ak je vrátené TRUE, prevádza sa telo cyklu príkaz. Na konci každého prechodu cyklom sa vyhodnotí výraz3, čo sa deje tak dlho, pokiaľ má výraz2 hodnotu TRUE. Obvykle sa výraz1 používa na inicializáciu riadiacej premennej cyklu, výraz2 je podmienka prevádzania cyklu a výraz3 slúži na pravidelnú aktualizáciu riadiacej premennej cyklu.

Vypíše čísla 1 až 10.

```
for ($i = 1 ; $i < 10 ; $i++)
{
    echo $i ;
}
```

3.4 Polia

Pole, na rozdiel od premennej, obsahuje viac hodnôt. Skladá sa z viacerých prvkov (každý má svoju hodnotu) a kľúča alebo indexu, pomocou ktorého na prvok odkazujeme. Indexom býva prevažne celé číslo alebo reťazec.

Najjednoduchším typom je pole, ktoré obsahuje sériu prvkov s indexmi od nuly. Napríklad, keď budeme mať pole \$miestnost a každý jeho prvok bude obsahovať číslo miestnosti, potom bude vyzeráť takto:

\$miestnost[0]	\$miestnost[1]	\$miestnost[2]	\$miestnost[3]
"CH16"	"CH261"	"CH307"	"BC667"

Inicializácia poľa

Jedným spôsobom je jednoduché priradenie hodnôt premennej poľu:

```
$miestnost[ ] = "CH16";  
$miestnost[ ] = "CH261";  
$miestnost[ ] = "CH307";
```

Ak priamo neurčíme indexy v hranatých zátvorkách, budú mu priradené postupne 0, 1, 2 atď. Samozrejme, pokiaľ chceme, môžeme priraďovať indexy podľa potreby.

Iný spôsob inicializácie poľa je pomocou funkcie array():

```
$miestnost = array ( "CH16", "CH261", "CH307" );
```

4. PHP a databáza MySQL

Ak chceme, aby naša stránka bola naozaj dynamická, musíme vyvinúť webovú aplikáciu založenú na databáze. PHP podporuje rozhranie API pre prístup k veľkému počtu databáz, ako sú Oracle, Sybase, MySQL atď. ODBC je štandardné rozhranie API pre prístup k databáze, ktorú PHP podporuje.

MySQL je malý, kompaktný, jednoducho použiteľný databázový server, ideálny pre malé a stredné aplikácie. Je asi trikrát až štyrikrát rýchlejší ako mnoho iných komerčných databáz a je distribuovaný zdarma. Aj preto je MySQL taký obľúbený.

4.1 Spojenie s MySQL

- `mysql_connect` - inicializuje spojenie s databázou MySQL

```
mysql_connect("nazov_mysql_servera", "uzivatel", "heslo");
```

`nazov_mysql_servera` - je meno MySQL servera poprípade počítača, na ktorom beží MySQL

`uzivatel` - je užívateľ MySQL servera, pod ktorým sa prihlasujeme

`heslo` - je naše heslo k DB MySQL

Napr. pokiaľ je na serveri vytvorený užívateľ `root` s heslom `root`, tak potom by prihlásenie na počítač `localhost` vypadalo takto:

```
mysql_connect("localhost", "root", "root");
```

- `mysql_close` - ukončí aktuálne spojenie s MySQL

```
mysql_close("spojenie");
```

`spojenie` - je voliteľné, môže byť použité aktuálne spojenie

- `mysql_create_db` - vytvára databázu

```
mysql_create_db("meno_databazy", "spojenie");
```

`meno_databazy` - je meno databázy, ktorú chceme vytvoriť

`spojenie` - je voliteľné, môže byť použité aktuálne spojenie

- `mysql_drop_db` - odstraňuje databázu

```
mysql_drop_db("meno_databazy", "spojenie");
```

`meno_databazy` - je meno databázy, ktorú chceme vymazať

`spojenie` - je voliteľné, môže byť použité aktuálne spojenie

- `mysql_select_db` – vyberá databázu, ktorá má byť aktívna

```
mysql_select_db("meno_databazy", "spojenie");
```

`meno_databazy` – je meno databázy, s ktorou chceme pracovať

`spojenie` – je pripojenie k serveru MySQL

- `mysql_query` - odosiela náš SQL príkaz k spracovaniu na server MySQL

```
mysql_query("SQL_prikaz", "spojenie");
```

`SQL_prikaz` - je SQL príkaz, ktorý sa má odoslať

`spojenie` - je identifikátor aktuálneho spojenia

4.2 Príkazy

- `mysql_num_rows` - vracia nám počet riadkov (táto funkcia je určená k zisteniu počtu riadkov v dotaze typu SELECT)

```
mysql_num_rows("ukazatel_na_vysledok_dotazu");
```

`ukazatel_na_vysledok_dotazu` - najčastejšie to býva výsledok z `mysql_query`.

- `mysql_fetch_array` - túto funkciu najčastejšie použijeme pre získanie riadkov z dát vrátených funkciou `mysql_query` (zoberie výsledok SQL príkazu a vloží ho do pola)

```
mysql_fetch_array("SQL_prikaz", "spojenie");
```

V okamihu, keď je výsledok uložený do pola, môžeme normálne použiť cyklus `for` alebo `while` k tomu, aby sme prečítali jej obsah. Hodnota každého stĺpca je priradená jednotlivým premenným, s ktorými budeme v našich skriptoch pracovať.

- `mysql_result` - získava vrátené záznamy z určitého výsledku

```
mysql_result(vysledok);
```

`vysledok` - je vygenerovaný pomocou `mysql_query`, `mysql_list_tables` alebo `mysql_list_dbs`.

- Príkaz `CREATE TABLE` – vytvorí tabuľku

Syntax je:

```
CREATE TABLE nazov_tabulky ( definicia_vytvorenia )
    [moznosti_tabulky ] [ prikaz_volby ];

definicia_vytvorenia:
nazov_stlpca typ_stlpca [ NOT NULL / NULL ][ DEFAULT hodnota ]
[ AUTO_INCREMENT]
alebo PRIMARY KEY (nazov_indexu_stlpca) alebo iné

typ_stlpca:
TINYINT[(dĺžka)]
INT[(dĺžka)]
VARCHAR[(dĺžka)]

moznosti_tabulky:
TYPE=ISAM / MYISAM / HEAP
alebo AUTO_INCREMENT=1 a iné
```

Ďalšie možnosti nebudeme vypisovať, pretože sme ich v našom programe nevyužili.

`NULL` - je hodnota, ktorá značí, že pole nie je prázdne, obsahuje len neznámu hodnotu

- príkaz SELECT - jeden z najobvyklejších príkazov v MySQL na zobrazovanie dát

```
SELECT stlpec1, stlpec2 ...  
FROM meno_tabulky;
```

Takisto si môžeme zvoliť jeden až všetky stĺpce, pričom ich oddeľujeme čiarkou. Niekedy sa ale stáva, že budeme potrebovať vypísať iba dáta, ktoré v určitom stĺpci obsahujú nejaké slovo, podľa ktorého ho nájdeme.

Na to nám slúži príkaz WHERE. Syntax je:

```
SELECT stlpec FROM meno_tabulky WHERE stlpec="nejaka_hodnota";
```

- príkaz ORDER BY

S uloženými záznamami sa dá ešte ďalej pracovať a to tak, že môžeme záznamy rôzne zoskupovať.

```
ORDER BY nejaky_stlpec DESC;
```

DESC je nastaviteľný parameter a znamená to „vráť záznam v zostupnom poradí“. Ak chceme zoradiť vzostupne, použijeme (namiesto DESC) parameter ASC.

- príkaz INNER JOIN

Tento príkaz použijeme vtedy, ak chceme vybrať len tie údaje z tabuľky, ktoré sa nachádzajú aj v inej tabuľke.

```
SELECT stlpec1  
FROM meno_tabulky1  
INNER JOIN meno_tabulky2  
ON meno_tabulky1. stlpec1 = meno_tabulky2. stlpec1
```

5. Tvorba programu

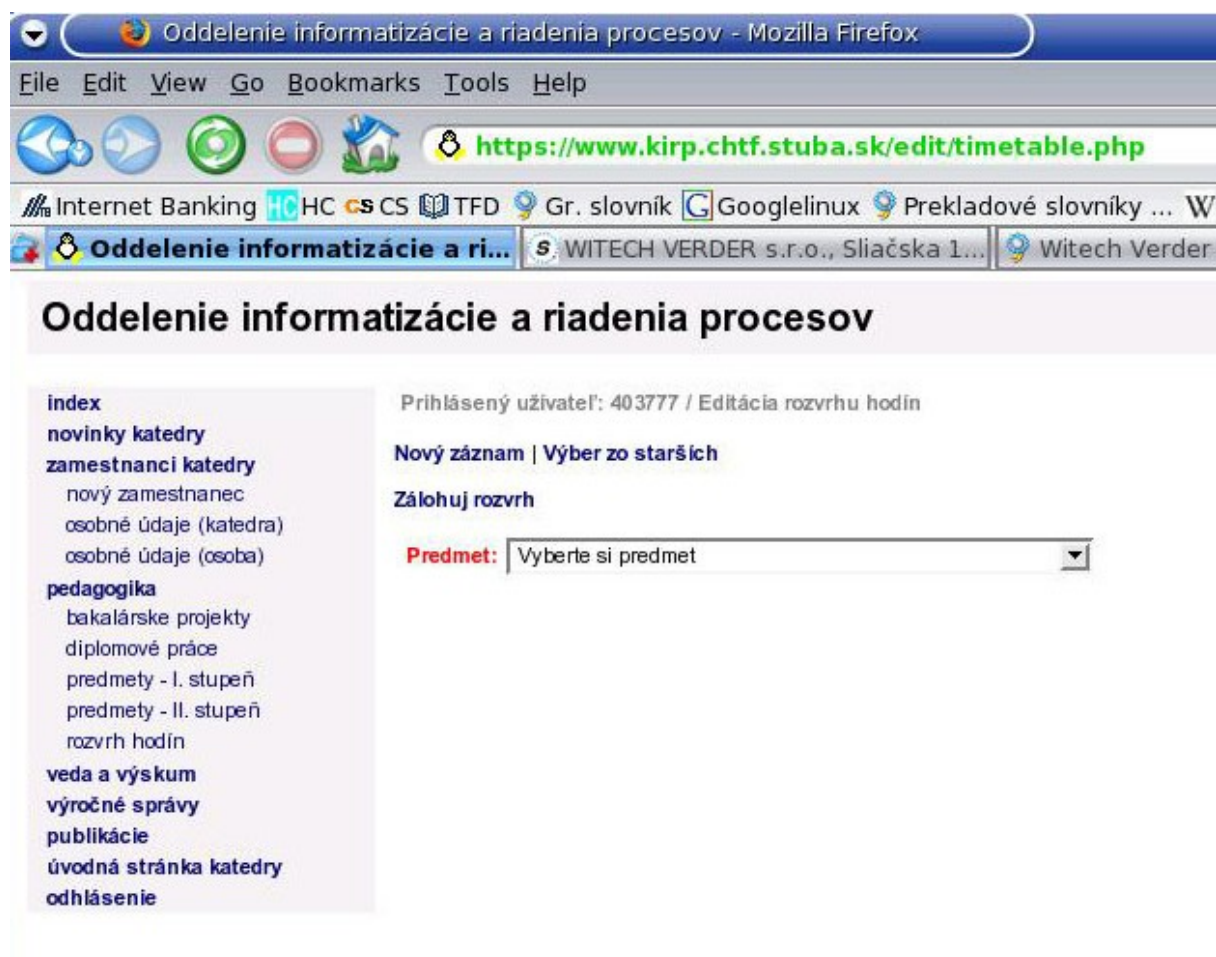
Najskôr si vytvoríme súbor config.php, v ktorom si zadefinujeme server, užívateľa, heslo, tabuľku a pripojíme sa na databázu. Ošetríme aj prípady, keď sa nám nepodarí pripojiť na databázu alebo ukončiť spojenie s databázou. Vtedy sa nám vypíše chybové hlásenie. V ďalších súboroch tieto údaje už nemusíme celé vypisovať, ale vložíme do nich celý súbor config.php pomocou príkazu INCLUDE. Nie je to nutné, ale bude to jednoduchšie a prehľadnejšie. Zdroje všetkých súborov sa nachádzajú v prílohe na CD.

5.1 Administrátorská časť

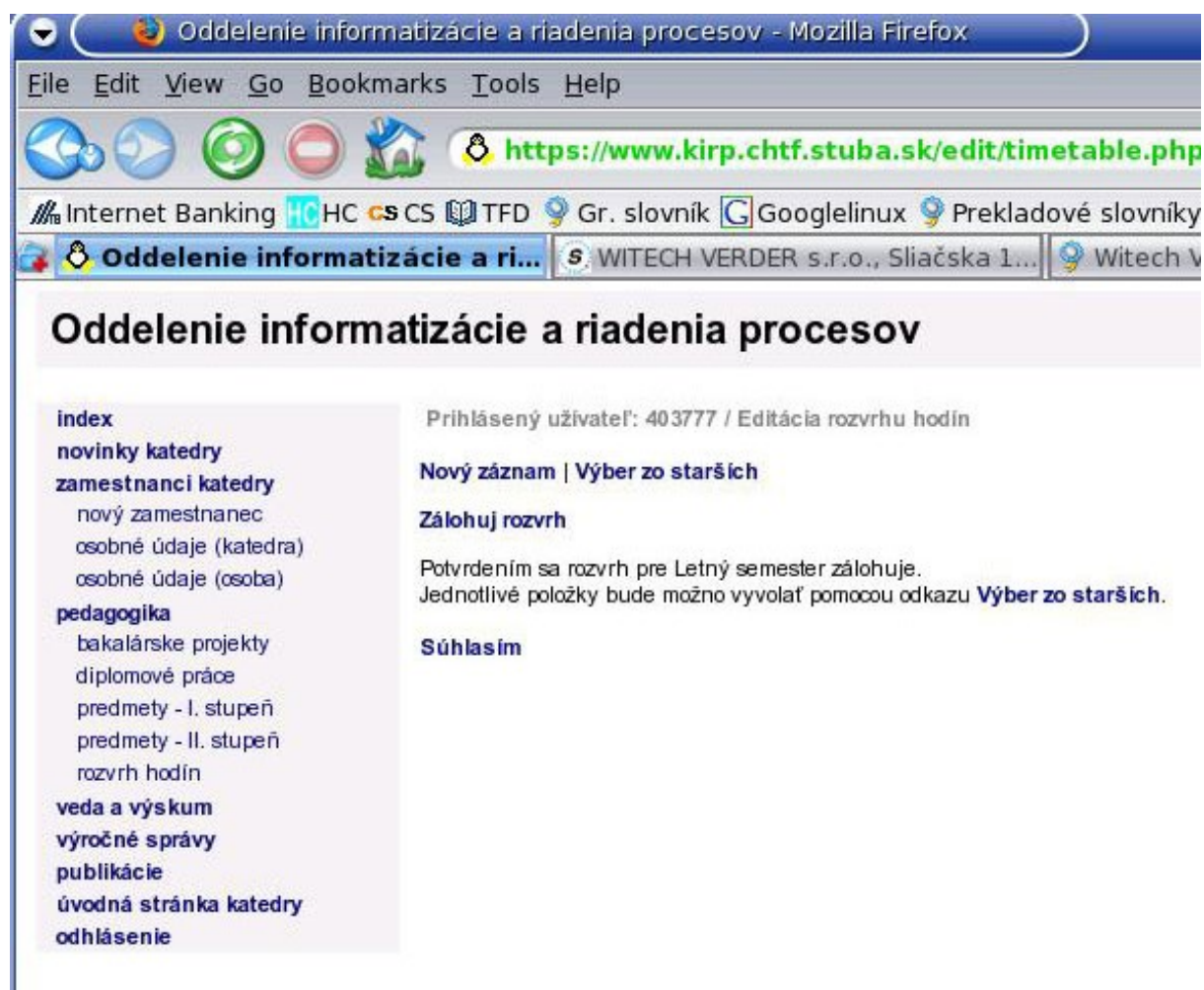
V tejto časti vytvoríme program, pomocou ktorého administrátor web stránky zálohuje rozvrh. Program nám na stránke dáva možnosť zálohovania rozvrhu. Po kliknutí na tento odkaz sa vypíše aký rozvrh sa zálohuje, pre letný alebo zimný semester. Po odsúhlasení sa tabuľka rozvrh skopíruje a jednotlivé položky bude možné vyvolať pomocou odkazu výber zo starších. Je rozdelený na tri súbory:

- kopirovat.php
- leto.php (zima.php)
- vytvorit.sql

Zdroje týchto súborov sa nachádzajú v prílohe na CD.



Obr. 5.1 Ukážka možnosti zálohovania rozvrhu



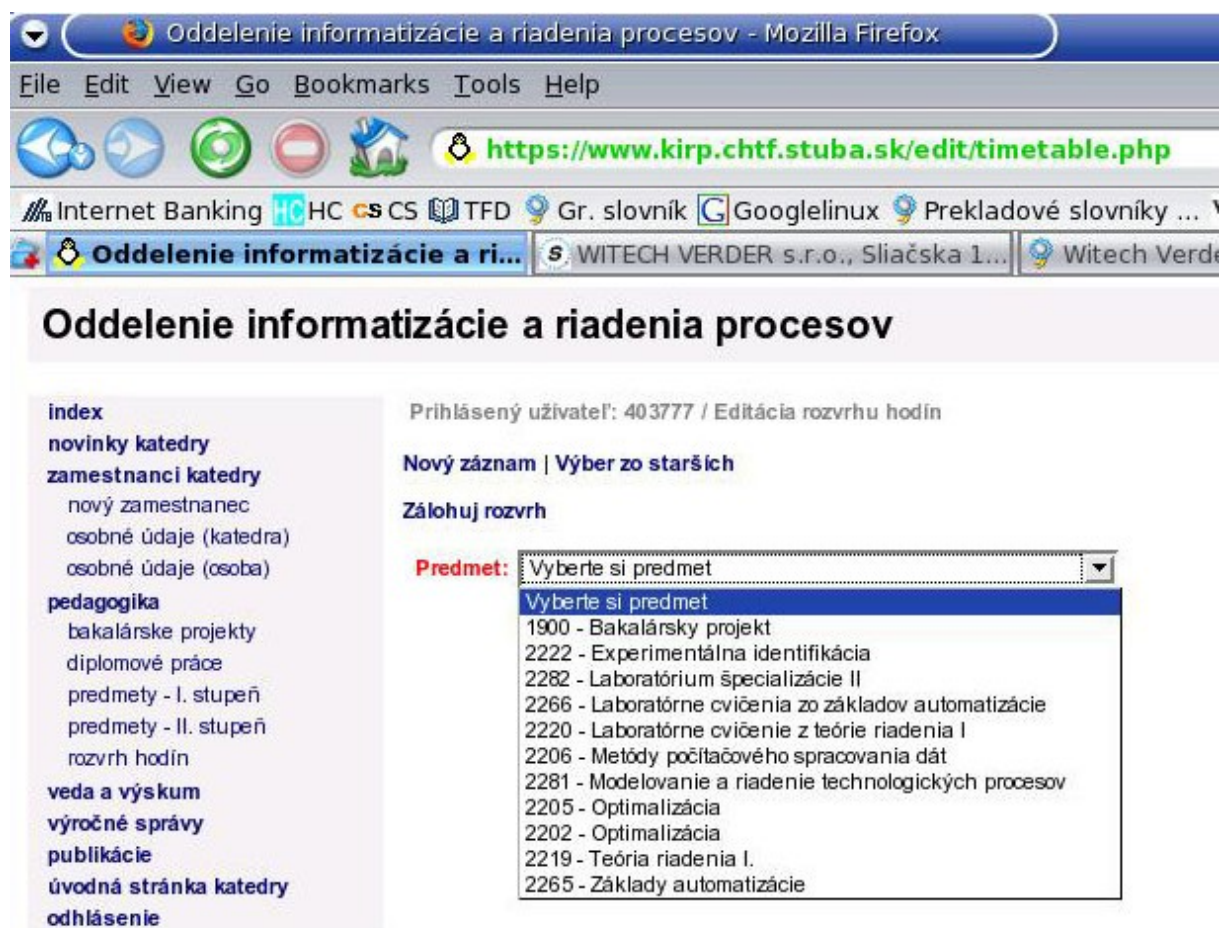
Obr. 5.2 Priebeh zálohovania rozvrhu

5.2 Užívateľská časť

Táto časť je rozdelená na dva súbory:

- predmety.php
- rozvrh.php

Súbor predmety.php vyberie z tabuľky predmety_bc len tie predmety, ktoré sa učia na katedre informatizácie a riadenia procesov a zároveň sa nachádzajú aj v databáze rozvrhu. Zobrazí ich vo výberovom poli aj s príslušnými kódmi predmetov.



Obr. 5.3 Výberové pole predmetov

Po výbere jedného z nich sa zobrazí rozvrh daného predmetu. Tento rozvrh vytvorí súbor rozvrh.php. Zobrazí sa tabuľka zoradená podľa dní a času, kedy sa tieto predmety vyučujú. Ku každému záznamu sa vypíše aj číslo krúžku, ročník, miestnosť a profesor. Tieto záznamy sa vyberajú z tabuľky rozvrh_leto (rozvrh_zima).

Oddelenie informatizácie a riadenia procesov - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://www.kirp.chtf.stuba.sk/edit/timetable.php

Internet Banking HC CS TFD Gr. slovník Googlelinux Prekladové slovníky ... Wikipedia

Oddelenie informatizácie a ri... WITECH VERDER s.r.o., Sliačska 1... Witech Verder

Oddelenie informatizácie a riadenia procesov

Prihlásený užívateľ: 403777 / Editácia rozvrhu hodín

Nový záznam | Výber zo starších

Zálohuj rozvrh

Predmet: Vyberte si predmet

Pre zadaný predmet existujú nasledujúce záznamy

Pondelok

Hodina	Predmet	Miest.	Krúžok	Ročník	Vyučujúci
11:00 - 13:00	Laboratorné cvičenia zo základov automatizácie	CH222	25	2	Závacká
11:00 - 13:00	Laboratorné cvičenia zo základov automatizácie	CH247	25	2	Vasičkaninová
15:00 - 17:00	Laboratorné cvičenia zo základov automatizácie	CH222	31	3	Závacká
15:00 - 17:00	Laboratorné cvičenia zo základov automatizácie	CH247	31	3	Karšaiová

Streda

Hodina	Predmet	Miest.	Krúžok	Ročník	Vyučujúci
09:00 - 11:00	Laboratorné cvičenia zo základov automatizácie	CH247	37	3	Karšaiová
11:00 - 13:00	Laboratorné cvičenia zo základov automatizácie	CH247	33	3	Karšaiová
13:00 - 15:00	Laboratorné cvičenia zo základov automatizácie	CH247	34	3	Fikar
15:00 - 17:00	Laboratorné cvičenia zo základov automatizácie	CH247	35	3	Vasičkaninová
16:00 - 18:00	Laboratorné cvičenia zo základov automatizácie	CH222	36	3	Puna

Štvrtok

Hodina	Predmet	Miest.	Krúžok	Ročník	Vyučujúci
07:00 - 09:00	Laboratorné cvičenia zo základov automatizácie	CH247	21	2	Karšaiová

Piatok

Hodina	Predmet	Miest.	Krúžok	Ročník	Vyučujúci
10:00 - 12:00	Laboratorné cvičenia zo základov automatizácie	CH222	30	3	Puna
10:00 - 12:00	Laboratorné cvičenia zo základov automatizácie	CH247	30	3	Bakošová

Poznámka: Prednáška Seminár Laboratorné cvičenie Cvičenie

Obr. 5.4 Zobrazenie rozvrhu vybraného predmetu

Názov predmetu sa zobrazí ako odkaz, ktorý po kliknutí naň, pošle ID číslo záznamu z tabuľky rozvrh_leto (rozvrh_zima) do už existujúcej aplikácie vo forme formulára na tvorbu nového rozvrhu. Zdroje týchto súborov sa nachádzajú v prílohe na CD.

Oddelenie informatizácie a riadenia procesov - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://www.kirp.chtf.stuba.sk/edit/timetable.php

Internet Banking HC HC CS CS TFD Gr. slovník Googlelinux Prekladové slovníky ... V

Oddelenie informatizácie a ri... WITECH VERDER s.r.o., Sliačska 1... Witech Verde

Oddelenie informatizácie a riadenia procesov

Prihlásený užívateľ: 403777 / Editácia rozvrhu hodín

Nový záznam | Výber zo starších

Zálohuj rozvrh

Predmet: 2266 - Laboratórne cvičenia zo základov automatizácie

Vyučujúci: Závacká Jana
ďalší vyučujúci

Miestnosť: CH222 pridat' novú miestnosť

Deň: Pondelok

Hodina: Od: 11:00 Do: 13:00

Forma: laboratórne cvičenie

Kružok: 25
Vyberte si kružok pridat' nový kružok

Ročník: 2. ročník

Počet hodín: 26 (za celý semester)
pridať

index
novinky katedry
zamestnanci katedry
nový zamestnanec
osobné údaje (katedra)
osobné údaje (osoba)
pedagogika
bakalárske projekty
diplomové práce
predmety - I. stupeň
predmety - II. stupeň
rozvrh hodín
veda a výskum
výročné správy
publikácie
úvodná stránka katedry
odhlásenie

Obr. 5.5 Formulár na tvorbu nového záznamu

6. Záver

Pri práci na tomto projekte sme sa naučili pracovať s databázami MySQL a oboznámili sme sa s programovacím jazykom PHP.

Naším cieľom bolo vytvoriť program, ktorý by uľahčoval prácu pri tvorbe nového rozvrhu hodín na našej fakulte. To sa nám aj podarilo. Zjednodušili a uľahčili sme profesorom tvorbu nových záznamov použitím starých.

Prispeli sme tak k modernejšiemu a efektívnejšiemu riešeniu tvorby rozvrhu hodín a tým sme aj študentom umožnili rýchlejší prístup k týmto informáciám.

7. Zoznam použitej literatúry

- [1] J. Castagnetto, H. Rawat, S. Schumann, Ch. Scollo, D. Veliath: PHP Programujeme profesionálne. Computer Press, Praha 2002
- [2] Ola Hansen: SQL Join, http://www.w3schools.com/sql/sql_join.asp
- [3] Martin Gulač: Kurz MySQL, <http://www.programujte.com>
- [4] Milan Gacík: Úvod do PHP, http://www.php.sk/php_tutor
- [5] Dušan Janovský: Syntaxe HTML, <http://www.jakpsatweb.cz/html>
- [6] Peter Halmo: Kurz PHP, <http://www.programujte.com>

Pod'akovanie

Týmto by som chcela poďakovať projektovému vedúcemu Ing. Daliborovi Punovi za jeho rady a pomoc pri práci na tomto projekte a za jeho čas, ktorý mi venoval. Ďakujem aj ostatným profesorom a asistentom na našej škole, ktorý podnietili môj záujem o štúdium. Samozrejme by som chcela poďakovať mojim priateľom a rodine, ktorý ma v práci podporovali a umožnili mi tento projekt ukončiť.

8. Príloha - CD