

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ TECHNÓGIE
ÚSTAV INFORMATIZÁCIE, AUTOMATIZÁCIE A MATEMATIKY
ODDELENIE MATEMATIKY



Tvorba dynamicky generovaných HTML formulárov
pomocou PHP a MySQL

Vypracoval: Róbert Lanák

Školiteľ: Ing. Tomáš Hirmajer

Bratislava 2007

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ
TECHNOLÓGIE STU V BRATISLAVE**

Oddelenie matematiky Číslo: 1/2006 Vec: Zadanie semestrálneho projektu na ukončenie
bakalárskeho štúdia.

1. Meno študenta /ky/: Róbert Lanák

(u vydatých aj rodné)

1. **2. Meno vedúceho projektu: Ing. Tomáš Hirmajer**
2. **3. Názov projektu:**

/So všetkými podrobnosťami: formulácia úlohy, rozsahu, podmienok riešenia,
harmonogramu riešenia a pod./

**Tvorba dynamicky generovaných HTML formulárov pomocou PHP a
MySQL**

1. **4. Termín odovzdania záverečnej práce projektu: 20. 05. 2006**
2. **5. Záverečná práca projektu sa odovzdáva v 2 zviazaných exemplároch vedúcemu projektu.**



Dátum: 20. 02. 2006 Doc. Dr. Ing. Miroslav Fikar vedúci oddelenia

Ďakujem vedúcemu bakalárskej práce Ing. Tomášovi Hirmajerovi za pomoc pri získavaní vedomostí z oblasti programovania a celkovo za vedenie, rady a pripomienky, ktoré mi poskytol pri vypracovávaní bakalárskej práce.

Abstrakt

Cieľom tejto práce je vytvoriť dynamickú webovú stránku pre použitie a administráciu dovolení na Oddelení informatizácie a riadenia procesov. Táto stránka je klasická dynamická webová stránka s databázou MySQL a PHP kódom. Proces vytvárania webovej stránky pozostáva z niekoľkých krokov: návrhu dizajnu, vytvorenia zdrojových kódov, importovania údajov do databázy a testovania funkčnosti. MySQL poskytuje široké možnosti vyhľadávania, pridávania a vymazávania záznamov v databáze.

Ako už bolo spomenuté, táto stránka bude používaná zamestnancami oddelenia k administrácii dovolenkových formulárov pomocou internetu. Výhodou tejto stránky je vyplniť dovolenkový formulár rýchlo s preddefinovanými informáciami o zamestnancovi a tiež zobrazenie aktuálneho zostatku dní dovolenky. Ďalšou výhodou je, že dovolenkové lístky sú automaticky prevedené do formy určenej k tlači.

Abstract

The aim of this work is to create dynamic web page for use and administration of vacations on Department of Information Engineering and Process Control. This page is a classic dynamic web page with MySQL database and PHP code. The process of creating the web page consists of several steps: proposition of the design, formatting of source codes, importing data into the database and finally testing of the functionality. MySQL provides wide possibilities of searching, deleting and adding reports into the database.

As were mentioned before, this page will be used by employees of our Department to administrate the vacation forms with help of Internet. The advantage of this page is to fill vacation forms quickly with predefined information about employer and it also allows displaying actual residual situation of vacations. Next advantage is that the vacation forms are automatically converted into the printed version.

Obsah

ZOZNAM SKRATIEK	vii
ZOZNAM OBRÁZKOV	viii
ÚVOD	9
1. TEORETICKÁ ČASŤ	10
1.1 APACHE.....	10
1.2 JAZYK PHP	10
1.2.1 Základné vlastnosti PHP	11
1.2.2 Ako pracuje PHP	11
1.2.3 HTML Formuláre	12
1.3 DATABÁZOVÉ SYSTÉMY	13
1.3.1 Základné vlastnosti databázových systémov.....	13
1.3.2 Vývoj databázových systémov	14
1.4 DATABÁZA MYSQL.....	16
1.4.1 Základné vlastnosti MySQL.....	16
1.5 PHPMYADMIN	16
2. APLIKÁCIE JAZYKA PHP A SQL	18
2.1 ZÁKLADNÁ SYNTAX JAZYKA PHP	18
2.2 ZÁKLADNÉ PRÍKAZY MYSQL POUŽÍVANÉ V PHP	19
3. PRAKTICKÁ ČASŤ	22
3.1 VYTVORENIE TABULIEK V DATABÁZE MYSQL	22
3.2 VYTVORENIE WEBOVÉHO ROZHRANIA.....	23
3.2.1 Vytvorenie hlavnej stránky	25
3.2.2 Vytvorenie ponuky „Online formulár“	26
3.2.3 Vytvorenie stránky „odoslane.php“	27
3.2.4 Vytvorenie stránky „ulozene.php“	28
3.2.5 Vytvorenie ponuky „Vyhľadať záznam“	29
3.2.6 Vytvorenie ponuky „Osoba“	29
3.2.7 Vytvorenie ponuky „Dovolenky“	30
3.2.8 Vytvorenie ponuky „Vyhľadať admin“	31
3.2.9 Vytvorenie ponuky „Nápoveda“	33

4. ZÁVER.....	34
LITERATÚRA	35

Zoznam skratiek

HTML	Hypertext Markup Language	Hypertextový značkovací jazyk
PHP	Hypertext Preprocessor	Hypertextový preprocesor
SQL	Structured Query Language	Štruktúrovaný dotazovací jazyk
WWW	World Wide Web	Celosvetová sieť
URL	Uniform Resource Locator	Adresa lokácie na sieti
DBMS	Database Management Systems	Riadiaci databázový systém
HTTP	Hypertext Transfer Protocol	Hypertextový prenosový protokol
POP3	Post Office Protocol	Poštový protokol
FTP	File Transfer Protocol	Protokol prenosu súborov

Zoznam obrázkov

Obr. 1	Spôsob generovania HTML dokumentu
Obr. 2	HTML formulár
Obr. 3	PHPMyAdmin
Obr. 4	Prehľad tabuliek v databáze MySQL
Obr. 5	Úvodná stránka „index.php“
Obr. 6	Zdrojový kód súboru „index.php“
Obr. 7	Stránka „dovolenka.php“ po vyplnení údajov
Obr. 8	Stránka „odoslane.php“
Obr. 9	Výsledok hľadania osobného čísla 11111 za rok 2007
Obr. 10	Ponuka „Osoba“
Obr. 11	Ponuka „Dovolenky“
Obr. 12	Ponuka „Vyhľadať admin“
Obr. 13	Ponuka „Nápoveda“

ÚVOD

Jedným z jazykov, ktorý sa používa pri tvorbe webových stránok je jazyk HTML (Hypertext Markup Language). Je to značkovací jazyk, pomocou ktorého sa definuje štruktúra webovej stránky. HTML je tu kvôli tvorbe jasných a presne štrukturovaných dokumentov. Z dnešného pohľadu je to však pre rôzne aplikácie nepostačujúce. Vývoj smeruje k väčšej interaktivite webu, ktorá pohotovo reaguje na požiadavky užívateľa. Prostredie prehliadačov sa stáva novou platformou, na ktorej môžu byť spustené najrôznejšie aplikácie. Do popredia sa dostávajú dynamické webové stránky generované na strane servera.

K vývoju dynamických stránok slúži niekoľko programovacích jazykov. Jedným z najúspešnejších a najrozšírenejších je jazyk PHP (PHP Hypertext Preprocessor). Jeho začiatky sa datujú do roku 1994. PHP je skriptovací jazyk, ktorý je vykonávaný na strane servera. Každú stránku obsahujúcu PHP skripty server najskôr prevezme a následne vykoná všetky príkazy v PHP. Klientovi pošle už len čistý HTML kód, ktorý je výsledkom behu skriptu.

Ďalším jazykom, ktorý sa často používa pri tvorbe dynamicky generovaných stránok je jazyk SQL (Structured Query Language). Bol vyvinutý s úmyslom poskytnúť účinný nástroj pre manipuláciu s dátami uloženými v relačných databázach. Medzi najznámejšie databázy patrí MySQL, ktorá je aj súčasťou programového balíka PHP Home Edition, ktorý bol použitý pri tvorbe tejto bakalárskej. Tento balíček obsahuje WWW (World Wide Web) server Apache, programovací jazyk PHP a databázu MySQL. Existujú aj rôzne iné podobné balíky, napr. Vertrigo.

Cieľom bakalárskej práce bolo vytvoriť dynamicky generované HTML formuláre pomocou PHP a MySQL, ktoré budú slúžiť na jednoduchšie evidovanie a spracovanie čerpaných dovolení. Táto práca zahŕňa vytvorenie tabuliek v databáze MySQL a vytvorenie webového rozhrania.

1. TEORETICKÁ ČASŤ

1.1 Apache

WWW - Server Apache je v súčasnosti najrozšírenejší WWW server, ktorý je možné prevádzkovať na rôznych platformách operačných systémov [1]. Má jednoduchú konfiguráciu a nízke nároky na operačný systém. Používa sa nielen na serveroch s operačnými systémami Unix a Linux ale aj na počítačoch s Microsoft Windows, BeOS, Mac a pod.

Apache bol vyvinutý na základe úspešného WWW servera NCSA. V roku 1995 vznikla skupina vývojárov APACHE Group, ktorá sa neskôr zmenila na APACHE Software Foundation. V tom istom roku prišla na svet prvá verzia Apache.

1.2 Jazyk PHP

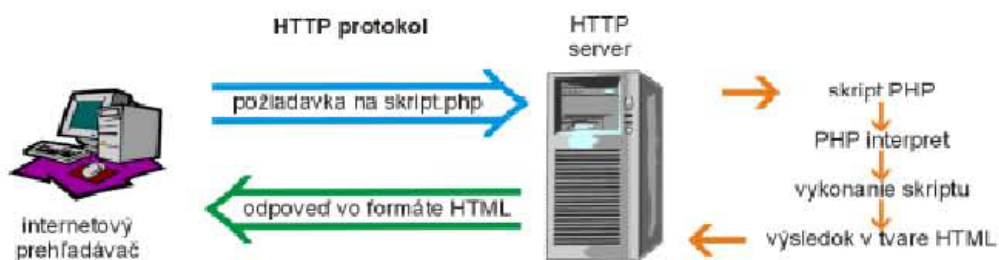
PHP je Open Source skriptovací jazyk [1], ktorý patrí medzi najpoužívanější jazyky pri programovaní webových stránok a jeho počiatky sa datujú do roku 1994. Vtedy sa Rasmus Lerdorf rozhodol vytvoriť jednoduchý systém pre počítanie prístupov ku svojim stránkam. Tento kód bol napísaný v PERLe. Za nejakú dobu bol systém prepísaný do jazyka C, pretože perlowski kód dosť zaťažoval server. Sada týchto skriptov bola ešte toho istého roku vydaná pod názvom Personal Home Page Tools, skrátene PHP. Neskôr sa táto sada premenovala na Personal Home Page Construction Kit. U toho však nezostalo. Po zabudovaní SQL modulu, ktorý umožňoval PHP spolupracovať s databázou, sa názov produktu zmenil na „Personal Home Page Form Interpreter“ (PHP/FI). Potom nastúpila verzia PHP 3.0. Oproti predošlej verzii bol systém urýchlenný a rozšírený o mnoho ďalších funkcií. V roku 2000 vyšla verzia PHP 4.0 (vyvíjaná pod názvom ZEND). Pôvodný názov PHP sa zmenil na hypertextový preprocesor (Hypertext Preprocessor). V súčasnosti je k dispozícii verzia 5.1.4. Údaje z apríla 2004 hovoria, že PHP beží na viac ako 15 000 000 doménach. PHP je využiteľné na automatické generovanie častí stránok (profesionálne grafy, rozsiahle tabuľky a pod.) a na predspracovanie informácií z formulárov na strane servera.

1.2.1 Základné vlastnosti PHP

- PHP je skriptovací jazyk vykonávaný na strane servera a vkladáný do bežného HTML kódu
- Súbor, ktorý obsahuje PHP sa ukladá najčastejšie s príponou .php alebo .php3
- PHP beží na strane servera a je optimalizované pre prostredie WWW
- PHP ľahko komunikuje s databázami ako sú napr. MySQL, PostgreSQL a rada ďalších
- PHP je multiplatformové a možno ho prevádzkovať na väčšine webových serverov a na väčšine dnes existujúcich operačných systémov
- PHP má syntax veľmi podobnú jazyku C a preto je blízky väčšine vývojárov
- PHP môže slúžiť ako brána k ďalším službám, pretože obsahuje podporu pre niektoré protokoly, ako sú napr. HTTP, FTP, POP3 a podobne
- PHP podporuje prácu s obrázkami a umožňuje pracovať s PDF dokumentmi

1.2.2 Ako pracuje PHP

Internetový prehliadač (napr. Internet Explorer, Firefox a pod.) vyšle požiadavku na zobrazenie dokumentu – skriptu PHP. Požiadavka dorazí k HTTP serveru, ten zistí (podľa toho ako je konfigurovaný), že dokument je skript a preto ho pošle PHP interpretovi. PHP interpret skript vykoná a výsledok behu skriptu (väčšinou to bude HTML formát) vráti späť HTTP serveru. Ten odošle výsledok internetovému prehliadaču, ktorý ho zobrazí používateľovi (Obr. 1).



Obr. 1. Spôsob generovania HTML dokumentu

1.2.3 HTML Formuláre

Jazyk HTML obsahuje podporu formulárov [2], ktoré slúžia pre načítanie údajov od používateľa.

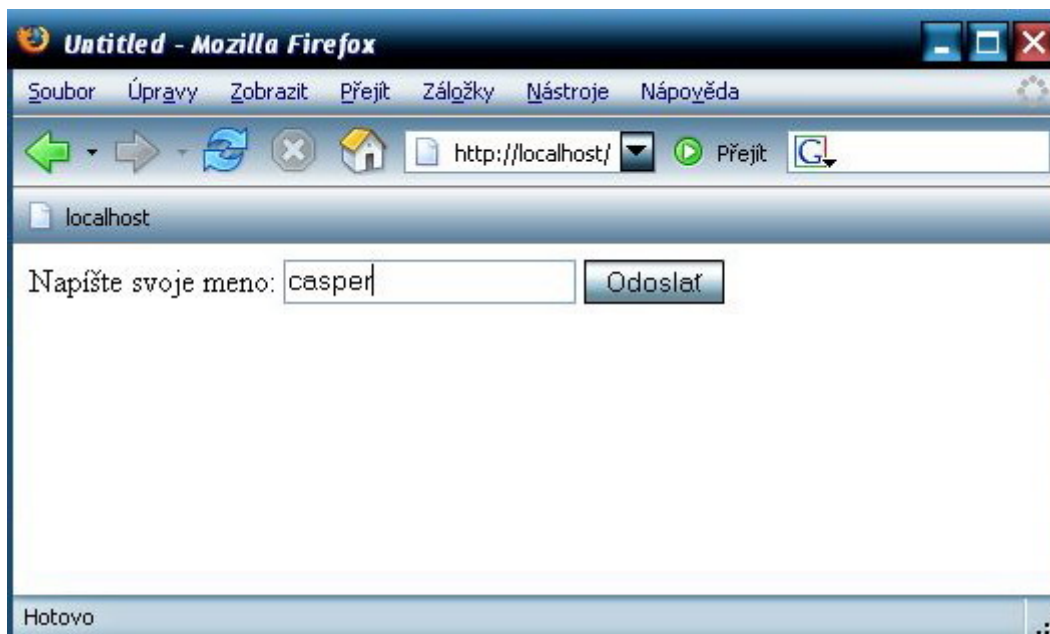
Webový formulár je časť stránky zložená z rôznych tlačidiel, vstupných polí a prepínačov, ktoré slúžia pre získanie dát od užívateľa [3]. Na vloženie formulára do stránky slúži párový tag `<form> </form>`. Ten má dva povinné atribúty a to ACTION a METHOD. Môže ale mať aj viac atribútov. Atribút ACTION určuje URL (Uniform Resource Locator) adresu skriptu, ktorý daný formulár spracuje. Atribút METHOD určuje akým spôsobom budú odoslané údaje z formulára predané serveru. Môžu byť odoslané metódou GET alebo POST. Metóda GET sa používa skôr pre odosielanie menších formulárov alebo takých formulároch, pri ktorých nezáleží na bezpečnosti, pretože táto metóda pripojí odosielané údaje za URL. Metóda POST kóduje zadané dáta rovnako ako metóda GET, ale nepripája ich za URL. Ďalšími atribútmi v tagu FORM môžu byť napr. „name“, „enctype“, „target“, „onsubmit“ a pod. Základným prvkom formulára je textové políčko, do ktorého sa môžu písať požadované informácie. Tagom, ktorý toto políčko definuje je `<input>`. Tento tag môže obsahovať niekoľko parametrov, napr. „name“, „type“, „size“, „value“ a pod. Ďalšími elementami vo formulároch sú napr. rolety, prepínače a zaškrŕavacie polia.

Po vyplnení formulára musí dať užívateľ na vedomie prehliadaču, že môže s obsahom formulára ďalej pracovať. Je tu možnosť, že sa obsah odošle na server alebo predá inému riadiacemu skriptu. To sa väčšinou vykoná kliknutím na tlačidlo, ktoré býva umiestnené na konci formulára. Tlačítko sa definuje opäť klasickým spôsobom a to značkou input, ktorá obsahuje atribút `type="submit"` alebo `type="button"`.

Príklad jednoduchého HTML formulára:

```
<html>
<head><title>Untitled</title></head>
<body>
<form action="spracuj.php" method="post">
Napište svoje meno: <input type="text" name="meno">
<input type="submit" name="odosli" value="Odoslať">
</form>
</body>
</html>
```

Výsledok príkladu je zobrazený na obr. 2.



Obr. 2 HTML formulár

1.3 Databázové systémy

1.3.1 Základné vlastnosti databázových systémov

Databázové systémy vo všeobecnosti charakterizujú predovšetkým nasledujúce vlastnosti:

1. **Perzistencia** (stálosť) - životnosť dát nie je ohraničená časom vykonávania programu - t.j. dáta existujú aj po ukončení programu, ktorý s nimi manipuluje. Tieto dáta sa uchovávajú v externých (sekundárnych) pamätiach: disky, magnetické pásky, a iné dátové nosiče.
2. **Správa externej pamäte** - väčšina dát sa uchováva v externých pamätiach. Ak chce program aktualizovať niektoré údaje, musí ich najprv načítať z externej do operačnej pamäte, tam ich spracovať a potom opätovne zapísať do externej pamäte. Preto musí databázový systém poskytovať prostriedky na zefektívnenie práce s externou pamäťou.
3. **Konkurentnosť** - s uchovávanými dátami môže súčasne pracovať viac používateľov. Systém musí zabezpečiť riešenie potenciálnych konfliktov, ak sa pokúsi s rovnakými údajmi naraz manipulovať niekoľko používateľov.
4. **Bezpečnosť** - databázové systémy sa často používajú v „citlivých“ oblastiach, ako napríklad bankové systémy a pod. Systém musí byť preto chránený pred nepovoleným

prístupom. Najlepšou ochranou sú vo všeobecnosti organizačné opatrenia v organizácii, kde sa systém prevádzkuje. Databázový systém musí umožniť identifikáciu používateľa a priradiť mu dáta a programy, ku ktorým má prístup (tzv. autorizácia).

5. **Spoločnosť** - keďže uchovávané dáta bývajú často veľmi dôležité, nesmie dôjsť k ich strate alebo poškodeniu pri zlyhaní hardwaru alebo softwaru. Ochrana pred zlyhaním hardwaru sa zabezpečuje predovšetkým hardwarovými prostriedkami. Databázový systém ju podporuje iba čiastočne - prostriedkami na zálohovanie dát do záložnej externej pamäte. Výraznejšiu ochranu poskytuje pred zlyhaním softwaru. Zahŕňa predovšetkým transakčný mechanizmus, umožňujúci zachovanie konzistentnosti databázy. Transakciu tvorí skupina príkazov definovaná používateľom, ktorá sa buď vykoná celá, alebo sa nevykoná vôbec.
6. **Dotazovanie** - na bežnú prácu s uchovávanými dátami využívajú koneční užívatelia aplikačné programy, vytvorené programátormi. Používatelia však často krát potrebujú vyhľadávať údaje v báze dát podľa určitých kritérií. Tieto kritériá nebývajú známe vopred a sú zvyčajne natoľko jednoduché a dočasné, že sa pre ne neoplatí vytvárať špeciálne programy. Databázový systém má preto poskytovať jazyk veľmi vysokej úrovne, ktorý dovolí používateľovi so základnou znalosťou štruktúry bázy dát vyhľadávanie dát podľa jednoduchých kritérií.

1.3.2 Vývoj databázových systémov

Databázové systémy prešli historicky tromi a vstupujú do štvrtej generácie:

1. Systémy súborov
2. Hierarchické a sieťové databázy
3. Relačné databázy
4. Objektovo - orientované databázy

Systémy súborov

V počiatočnej ére počítačov obsluhoval programátor externé pamäte na nízkej úrovni - v jazyku riadiacej jednotky externej pamäte. S výnimkou perzistencie si všetky ostatné požiadavky kladené na databázové systémy musel zabezpečovať sám. Vylepšenie priniesli

operačné systémy (OS) so systémami súborov (file system - príklad FAT16, NTFS). Systém umožnil „fyzickú“ nezávislosť od konkrétneho typu externej pamäte. Tvorca aplikácie programoval na úrovni vyššieho programovacieho jazyka. Prvé operačné systémy boli dávkovo orientované a jednouchádzačské. Narábali s pomalými externými pamäťami so sekvenčným prístupom (magnetické bubny a pásy). Modernjšie OS priniesli interaktívnu prácu viacerých používateľov a vďaka pokroku v hardvare využívali rýchlejšie externé pamäte s „priamym“ prístupom (disky). Pre pohodlnejšiu a efektívnejšiu prácu s perzistentnými údajmi sa dodávali špecializované knižničné funkcie, umožňujúce definovanie a prístup na úrovni záznamov (record), ich triedenie, tvorbu indexových súborov a pod.

Hierarchické a sieťové databázy

Koncom 60. a začiatkom 70. rokov vznikli prvé systémy, ponúkajúce úplnejšie riešenie databázových požiadaviek. Dostali názov DBMS (Database Management Systems). DBMS na rozdiel od systémov súborov umožnili konkurenčnú prácu na úrovni záznamov, transakčný mechanizmus, prostriedky na zálohovanie a autorizáciu, nastavovanie parametrov pre správu externej pamäte [4].

Relačné databázy

Tretiu generáciu databázových systémov, ktorá oproti predchádzajúcej priniesla kvalitatívny posun, tvoria relačné databázy. Ich teoretický základ položil už v roku 1970 Codd. Relačné databázové systémy zjednodušili tvorbu a využívanie aplikácií. Narábanie s databázou nadobudlo charakter práce s obyčajnými tabuľkami. Riadky tabuľky predstavovali záznamy jedného typu a stĺpce ich atribúty. Tabuľky medzi sebou odkazovali pomocou stĺpcov s rovnakými menami. K relačným databázam bol pridaný štruktúrovaný dotazovací jazyk SQL.

Objektovo – orientované databázy

Relačné DBMS boli navrhnuté a ukázali sa ako veľmi vhodné pre „agendové“ databázové aplikácie, napríklad bankové a poisťovacie systémy, účtovníctvo a pod. Po úspechu databázových systémov v týchto oblastiach sa objavili požiadavky na ich uplatnenie aj v „nových“ oblastiach aplikácie, ktorými boli napr. riadenie výroby (v strojárskom, chemickom a inom závode), spracovanie dokumentov (hypertexty, grafická dokumentácia, automatizácia kancelárskych prác), softwarové inžinierstvo (CASE systémy) a pod.

Objektovo orientované databázy priniesli oproti relačným databázam bohatšie dátové aj procedurálne modelovanie.

1.4 Databáza MySQL

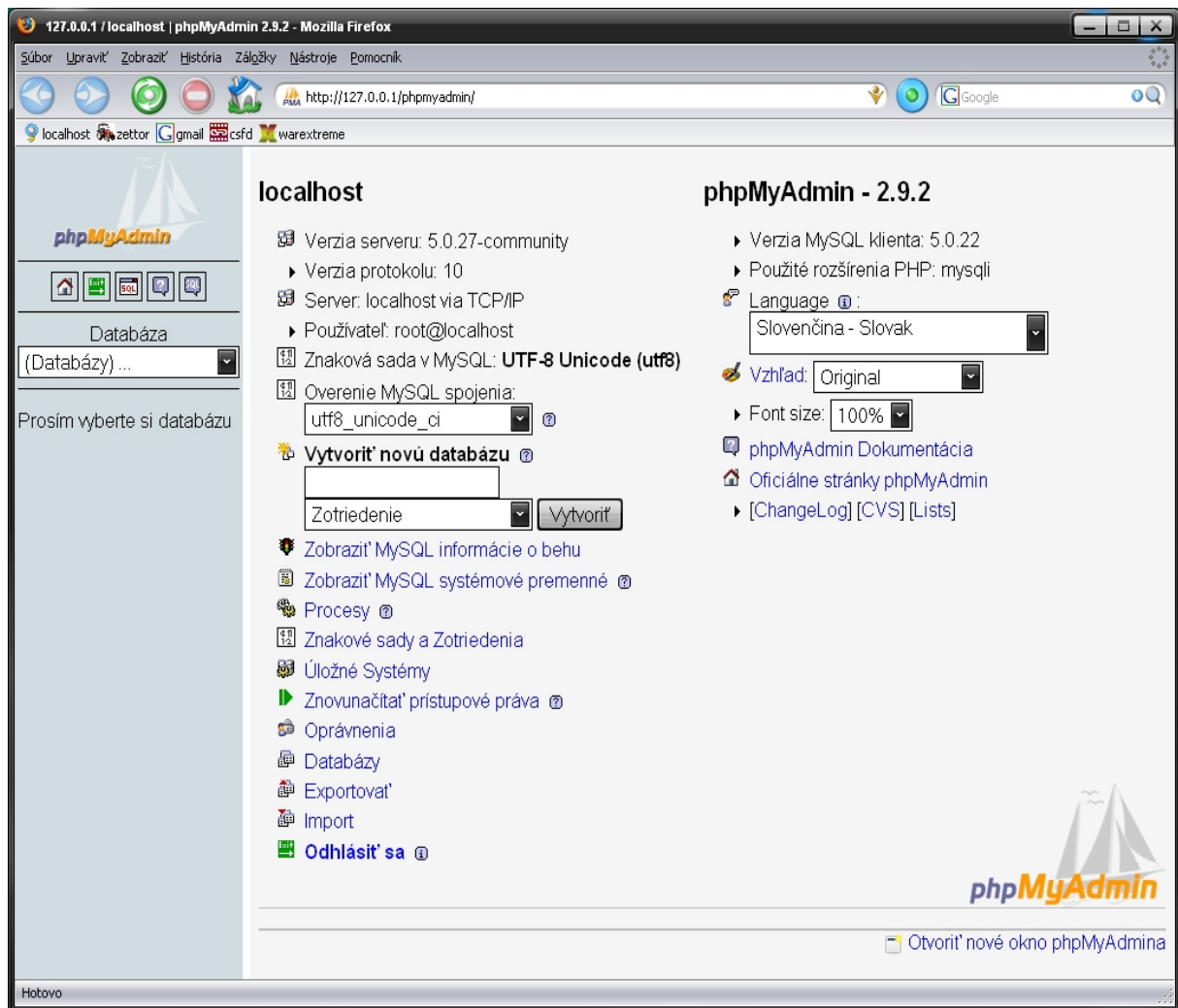
Databáza MySQL sa stala svetovo najpopulárnejšou Open Source databázou uznávanou vývojármi pre jej rýchlosť s akou vykonáva funkcie webových aplikácií spojené s overovaním (authentication), splnomocňovaním (authorization) a personalizáciou (personalization) oprávnených užívateľov [5]. Medzi jej hlavné prednosti patrí jednoduchá inštalácia a obsluha. MySQL je relačná databáza a preto v nej môžeme vytvárať tabuľky, pridávať záznamy, či vyberať dáta v súlade s normou SQL.

1.4.1 Základné vlastnosti MySQL

- MySQL je rýchly a výkonný databázový systém umožňujúci spracovanie veľkého množstva údajov
- MySQL je open - source projekt
- Jednoduchá administrácia
- Podporu pre MySQL ponúka napr. programovací jazyk PHP
- rozhranie pre programovacie jazyky Perl, C++, Ruby

1.5 PHPMyAdmin

Je jeden z najlepších nástrojov na správu databázy MySQL cez webové rozhranie (Obr.3). PHPMyAdmin je aplikácia napísaná v jazyku PHP. S jeho pomocou môžeme pripraviť veľmi rýchlo novú tabuľku, pridávať alebo odstraňovať záznamy, exportovať záznamy alebo odhaliť chyby v SQL príkazoch. V ľavej časti sa nachádza zoznam všetkých databáz prípadne tabuliek prístupných danému užívateľovi a v pravej časti je množstvo funkcií na správu databázy, editovanie vybranej databázy a tabuliek v nej.



Obr. 3 PHPMYAdmin

2. APLIKÁCIE JAZYKA PHP A SQL

2.1 Základná syntax jazyka PHP

Syntax PHP je veľmi podobná, takmer identická so syntaxou jazyka C. PHP príkazy a funkcie sa od textu HTML oddeľujú značkami `<?php ?>` alebo skrátené `<? ?>`. Príkaz `echo` slúži na vypísanie požadovaného textu alebo HTML kódu, ktorý sa odošle do prehliadača.

Príklad:

```
<?php echo "Toto je skript v jazyku PHP."; ?>
```

V jednom riadku môže byť vložených aj viac príkazov ako jeden, ale medzi každým príkazom musí byť bodkočiarka. Príklad pre zadávanie príkazov:

```
<?php  
echo "Prvý riadok";  
echo "Druhý riadok";  
?>
```

Komentár je niečo, čo program pri spracovávaní ignoruje a slúži teda len pre užívateľa. Komentáre sa používajú hlavne pri dlhších skriptoch, kde pomáhajú pri orientácii, prípadne vysvetľujú kód.

Jednoriadkový komentár:

```
<?php  
// toto je jednoriadkový komentár  
alebo  
# toto je jednoriadkový komentár  
?>
```

Viacriadkový komentár

```
<?php  
/*  
Neviem čo by som  
sem ešte napísal  
*/  
?>
```

Každý programovací jazyk pracuje s nejakými hodnotami. To, do čoho sa tieto hodnoty ukladajú sa nazýva premenné. Každá premenná má svoj názov a hodnotu. Premenné v PHP musia vždy začínať znakom \$ (dolára), prvý znak musí byť písmeno alebo podtržník. Nesmie to byť číslo ani nič iné. Premenné v PHP rozlišujú medzi malými a veľkými písmenami: \$DATUM sa nerovná \$datum.

PHP je štruktúrovaný jazyk. To znamená, že spolu súvisiace príkazy je možné združiť do blokov, ktoré tvoria celkovú štruktúru programu. Typickým príkladom sú podmienky:

```
<?php
if (podmienka1) {
    príkaz 1;
}
else {
    if (podmienka2) {
        príkaz 2;
    }
    else {
        príkaz 3;
    }
}
?>
```

2.2 Základné príkazy MySQL používané v PHP

V prvom rade musíme mať skript na pripojenie k databáze. Môže mať napríklad tvar:

```
<?php
Function pripoj($server, $user, $password, $databaza) {
    $server = "server";
    $user = "user";
    $password = "password";
    $databaza = "databaza";
    if (!@$spojenie = MySQL_Connect("server", "user", "password"))
    { echo "Nepodarilo sa pripojiť k databáze!"; }
    if (!@$select = MySQL_Select_Db("databaza"))
    { echo "Chyba pri výbere databázy!"; }
}
?>
```

Vytvoríme si hore uvedenú funkciu na pripojenie k databáze. Potom túto funkciu zavoláme tam, kde potrebujeme ale väčšinou sa to robí iba raz a to na začiatku stránky. Skript overí, či

sa spojenie podarilo. Ak áno, pokračuje bez problémov ďalej a vyberie databázu, v ktorej máme vytvorené tabuľky. V prípade, že sa spojenie nepodarí, vygeneruje sa chybové hlásenie. Do premennej „server“ sa ukladá adresa, kde beží MySQL. Väčšinou je to localhost. Do premennej „user“ sa vkladá meno užívateľa pre pripojenie k MySQL. Do premennej „password“ sa vkladá heslo pre pripojenie. Do premennej „databaza“ sa napíše meno databázy, kde sú uložené údaje. Databáza môže obsahovať x tabuliek a každá tabuľka y dát. Tabuľka vytvára určité pravidlá pre ukladanie záznamov.

Medzi najzákladnejšie príkazy patria:

Vytvorenie databázy

```
CREATE DATABASE meno
```

Vytvorenie tabuľky

```
CREATE TABLE ludia (  
  id int(8) auto_increment,  
  name varchar(250) not null,  
  password varchar(250) not null,  
  PRIMARY KEY id(id)  
) TYPE = MyISAM;
```

Hore uvedeným príkazom sme v MySQL vytvorili tabuľku „ludia“. Obsahuje 3 stĺpce. Stĺpec „id“, ktorého hodnota sa pri doplnení dát do tabuľky zvýši vždy o 1. Stĺpec „name“ typu reťazec o dĺžke znakov 250, do ktorého musia byť vždy zapísané dáta (to zabezpečuje parameter not null) a stĺpec „password“ s rovnakými vlastnosťami ako stĺpec „name“. Príkaz PRIMARY KEY id(id) zabezpečí rýchlejšie vyhľadávanie v záznamoch.

Vkladanie záznamov do tabuľky

Nasledujúci príkaz vloží záznam do tabuľky ludia, kde „Id je 1, „name“ je Robo a „password“ je bomba.

```
mysql_query("INSERT INTO ludia VALUES ('1', 'Robo', 'bomba')");
```

Výber záznamov z tabuľky

Nasledovným príkazom sa vyberajú záznamy z tabuľky. Podmienkou môže byť napr. name=“Robo“

```
mysql_query("SELECT * FROM ludia WHERE podmienka");
```

Odstránenie záznamu z tabuľky

Na odstránenie záznamu z tabuľky „ludia“ sa používa nasledovný príkaz. Podmienkou môže byť napr. id=“4“

```
mysql_query("DELETE FROM ludia WHERE podmienka");
```

3. PRAKTICKÁ ČASŤ

Praktická časť pozostávala z dvoch hlavných častí:

- vytvorenia tabuliek v databáze MySQL
- vytvorenia webového rozhrania pre správu dovoleniek

3.1 Vytvorenie tabuliek v databáze MySQL

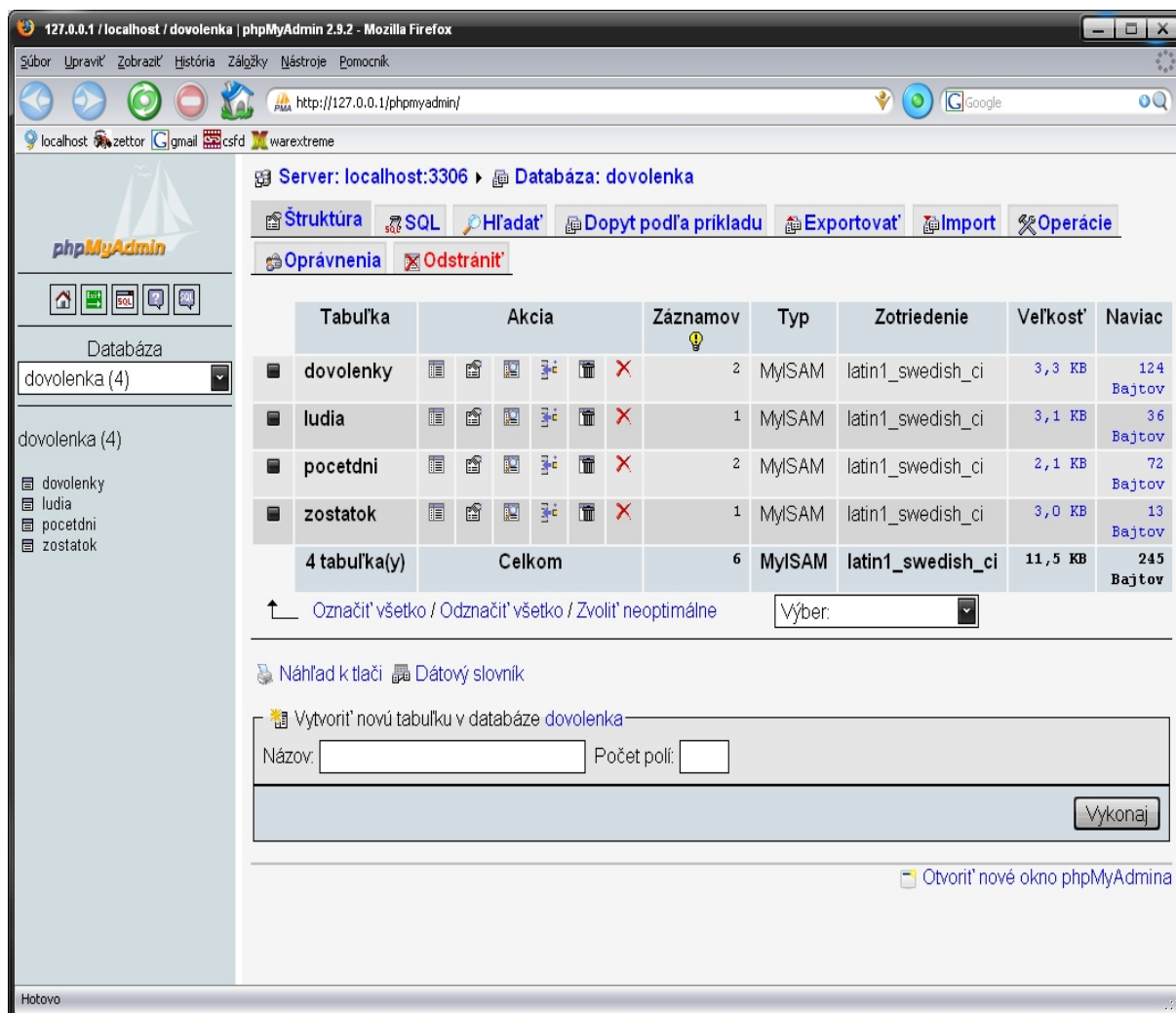
Na vytvorenie tabuliek v MySQL bol použitý PHPMyAdmin. Bolo potrebné vytvoriť nasledovné tabuľky: „ľudia“, „dovolenky“, „počet dní“ a „zostatok“ (Obr. 4). Každá z tabuliek obsahuje polia, do ktorých sa vkladajú záznamy. Do každého poľa sa zadefinovalo o aký typ ide. Napríklad integer, varchar a pod.

Tabuľka „ľudia“ obsahuje informácie a zamestnancoch a jej štruktúra je nasledovná. V prvom stĺpci sa nachádza id záznamu, v druhom osobné číslo zamestnanca, v treťom jeho meno a priezvisko. Štvrtý stĺpec obsahuje názov útvaru a posledný číslo útvaru. Ako primárny kľúč je nastavené osobné číslo, čo zabezpečí rýchlejšie vyhľadávanie v tabuľke. Vkladanie osôb do tejto tabuľky, prípadne editovanie sa robí v administrátorskej časti v menu „Osoba“.

Ďalej bola vytvorená tabuľka „dovolenky“, kde sa budú postupne ukladať záznamy dovoleniek. Obsahuje 7 stĺpcov: „id“, „os_c“, „od“, „do“, „tj“, „miesto“ a „dátum“. Záznamy sú do nej vložené vyplnením formulára v užívateľskom rozhraní.

Treťou tabuľkou je „počet dní“, kde sa ukladá počet dní dovolenky, ktorú môže zamestnanec čerpať za rok. Obsahuje 4 stĺpce: „id“, „osc“, „rok“ a „dni“. Pridávanie záznamov do tejto tabuľky sa robí v administrátorskej časti rozhrania v menu „Dovolenky“.

Ako posledná bola vytvorená tabuľka „zostatok“, ktorá obsahuje aktuálny počet dní dovolenky. Skladá sa z 3 stĺpcov: „id“, „osc“ a „zostatok“. Záznam sa do tabuľky vkladá automaticky pridaním osoby a s čerpaním dovolenky sa zostatok aktualizuje.



Obr.4 Prehľad tabuliek v databáze MySQL

3.2 Vytvorenie webového rozhrania

Po zadání URL adresy do prehliadača sa načíta „index.php“ a zobrazí sa webová stránka, ktorá je rozdelená na dve časti. V ľavej časti obrazovky je výberové menu a v pravej sa zobrazuje daná stránka podľa požiadaviek užívateľa. Keďže ide iba o modul, ktorý sa má implementovať do zabezpečenej stránky, prihlasovanie užívateľov nebolo vytvorené a zobrazujú sa všetky položky menu. V hlavnom menu sa nachádzajú tlačidlá „Online formulár“, „Vyhľadať záznam“, „Osoba“, „Dovolenky“, „Vyhľadať admin“ a „Nápoveda“. Tlačidlá „Online formulár“ a „Vyhľadať záznam“ patria medzi užívateľské a zvyšné sú administrátorské. Tlačidlo „Nápoveda“ je spoločné pre obe skupiny. Po načítaní stránky sa v pravej časti zobrazí ako prednastavená stránka „Online formulár“.

Dovolenky - Mozilla Firefox

Súbor Upraviť Zobraziť História Záložky Nástroje Pomocník

http://localhost/projekt/projekt_28_03_2007/dovolenka/index.php?menu=

localhost zettor gmail csfd warextreme

Online formulár
Vyhľadať záznam
Osoba
Dovolenky
Vyhľadať admin
Nápoveda

Online formulár

Napište osobné číslo: *

Priezvisko, meno, titul:

Osobné číslo:

Číslo útvaru:

Útvar:

Od: *

Do: * vrátane

t.j. : pracovných dní

« Marec 2007 »							« Marec 2007 »						
Po	Ut	St	Št	Pi	So	Ne	Po	Ut	St	Št	Pi	So	Ne
				1	2	3 4					1	2	3 4
5	6	7	8	9	10 11		5	6	7	8	9	10 11	
12	13	14	15	16	17 18		12	13	14	15	16	17 18	
19	20	21	22	23	24 25		19	20	21	22	23	24 25	
26	27	28	29	30	31		26	27	28	29	30	31	

Miesto pobytu na dovolenke:

Dátum: (* povinné údaje)

W3C HTML 4.01 ✓ W3C CSS ✓

Design @ 2007 Casper

Hotovo

Obr. 5 Úvodná stránka „index.php“

3.2.1 Vytvorenie hlavnej stránky

Na začiatku bola vytvorená hlavná stránka „index.php“, ktorej obsah je zobrazený na obrázku 5. Jej zdrojový kód je zobrazený na obrázku 6.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>

<meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
<title>Dovolenky</title>
<link rel="stylesheet" href="style.css" type="text/css">

</head>
<body>

<?php require "funkcie.php";?>

<table border="0" class="tabulka" align="center">
<tr>
<td width="130" valign="top" bgcolor="#EBEBEB" align="center"><br>
<?php include "menu.php" ?><br>
</td>
<td valign="top" width="660" class="tdtabulka"><?php ukazmenu() ?></td>
</tr>
</table>

<table border="0" width="790" align="center">
<tr>
<td><p>
<a href="http://validator.w3.org/check?uri=referer">
</a>
<a href="http://jigsaw.w3.org/css-validator/">
</a>
</p>
</td>
<td align="right" class="td2"> Design © 2007 Casper</td>
</tr>
</table>

</body>
</html>
```

Obr. 6 Zdrojový kód súboru „index.php“

V prvom riadku tohto kódu je zadefinované podľa akých pravidiel sa majú spracovávať jednotlivé príkazy v dokumente. To zabezpečuje DOCTYPE (DOCument TYPE definition). Vo vnútri tagu <title> a </title> sa nachádza názov stránky, ktorý sa zobrazí v prehliadači. Medzi <head> a </head> je daný riadok, pomocou ktorého sa aktivuje slovenská diakritika a odkaz na súbor so štýlmi.

3.2.2 Vytvorenie ponuky „Online formulár“

Stránka „dovolenka.php“ obsahuje hlavný formulár na vyplňovanie dovolenky (Obr. 7). Do prvého políčka zamestnanec vpiše svoje osobné číslo a stlačí tlačidlo odoslať, ktoré sa nachádza vedľa. Jeho stlačením sa spustí skript, ktorý v databáze v tabuľke „ľudia“ vyhľadá dané osobné číslo a údaje k nemu priradené. Tieto údaje, meno, priezvisko, číslo útvaru a útvar sa automaticky vložia do daných políčok. Ako druhé ešte vyhľadá záznamy dovoleniek v danom mesiaci, ktorý je práve vykreslený v kalendári a zvýrazní dni, v ktorých už dovolenku čerpá. Nasleduje výber dní dovolenky z kalendárov. Sviatky, soboty a nedele sú zobrazené inou farbou kvôli lepšej prehľadnosti. Z prvého kalendára si zamestnanec vyberie dátum odkedy chce dovolenku a z druhého dokedy. Ak sú vybrané oba dátumy, javascript „rozdiel.js“ automaticky vyráta ich rozdiel a odráta soboty, nedele a sviatky. Rozdiel potom vloží do políčka „tj“. Do poľa „miesto pobytu na dovolenke“ môže zamestnanec napísať, kde bude tráviť dovolenku. V políčku „dátum“ je aktuálny dátum vyplňovania formulára. Pod ním sa nachádzajú tlačidlá „Odoslať“ a „Vymazať“. Stlačením „Odoslať“ dôjde k spracovaniu údajov. Pred spracovaním sa však skontroluje, či sú vyplnené všetky údaje (výnimkou je miesto pobytu na dovolenke). Ak nie, zobrazí sa varovné okno, ktoré upozorní na nevyplnené políčka. Ak sú všetky údaje v poriadku, formulár spracuje stránka „odoslane.php“.

The screenshot shows a Mozilla Firefox browser window displaying a web form titled "Online formulár". The form is for submitting vacation requests. The data entered is as follows:

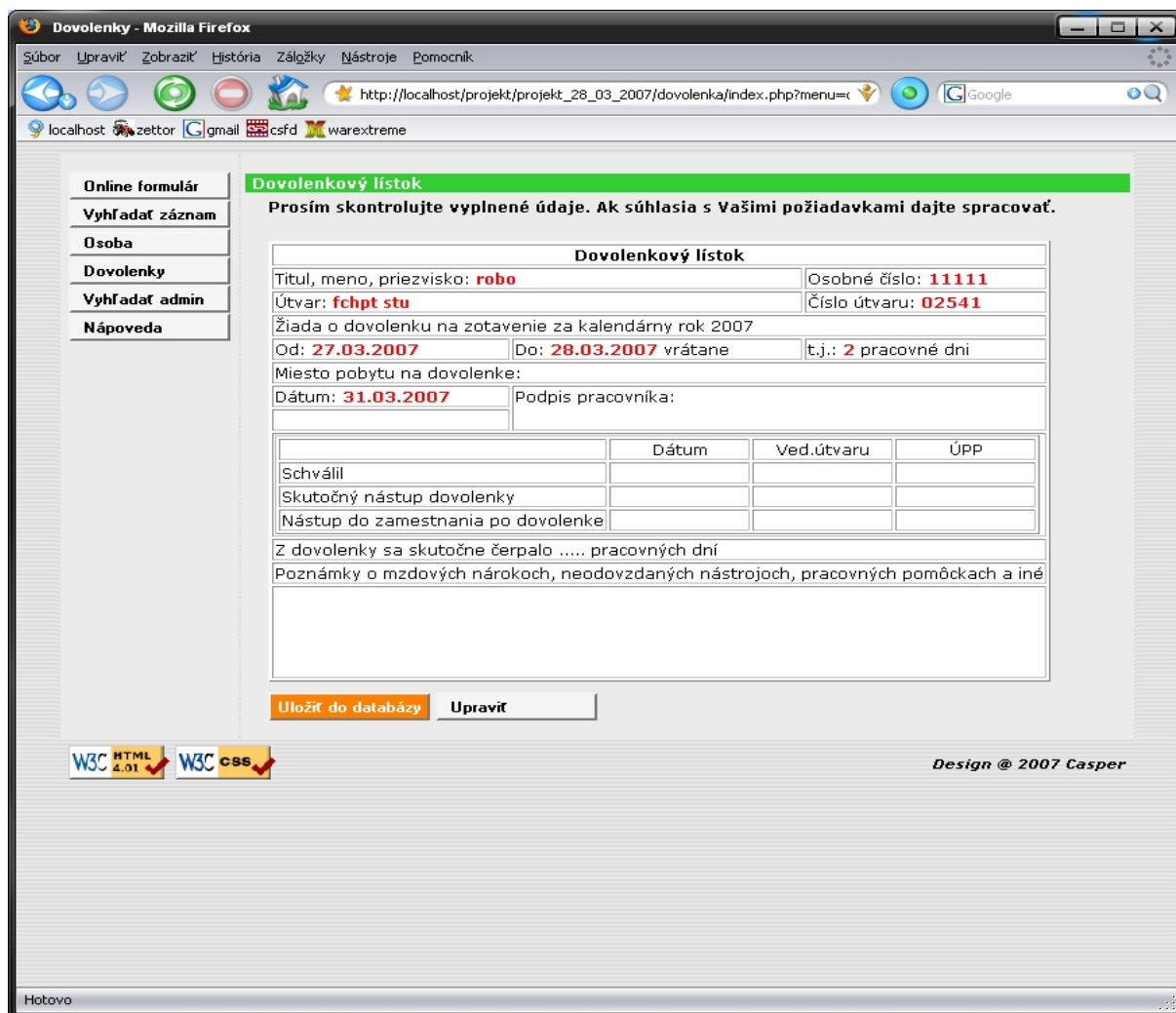
- Napište osobné číslo:** 11111
- Priezvisko, meno, titul:** robo
- Osobné číslo:** 11111
- Číslo útvaru:** 02541
- Útvar:** fchpt stu
- Od:** 27.03.2007
- Do:** 28.03.2007 vrátane
- t.j.:** 2 pracovných dní
- Miesto pobytu na dovolenke:** (empty text box)
- Dátum:** 31.03.2007 (* povinné údaje)

At the bottom of the form, there are two buttons: "Odoslať" (orange) and "Vymazať" (white). The browser's address bar shows the URL: `http://localhost/projekt/projekt_28_03_2007/dovolenka/index.php?menu=`. The browser's status bar at the bottom indicates "Hotovo".

Obr. 7 Stránka „dovolenka.php“ po vyplnení údajov

3.2.3 Vytvorenie stránky „odoslane.php“

Ako už bolo spomenuté, táto stránka spracováva formulár z „dovolenka.php“. Po jeho spracovaní sa automaticky generuje dovolenkový lístok. Ak dovolenka spadá do jedného mesiaca, vygeneruje sa jeden lístok. Keď je cez dva mesiace, vygenerujú sa dva lístky. Po ich skontrolovaní možno tieto údaje uložiť do databázy. Toto uloženie zabezpečuje stránka „ulozene.php“. Ak údaje nesúhlasia, tlačidlom „Upraviť“ sa dostaneme späť na stránku „dovolenka.php“ a môžeme ich zmeniť.



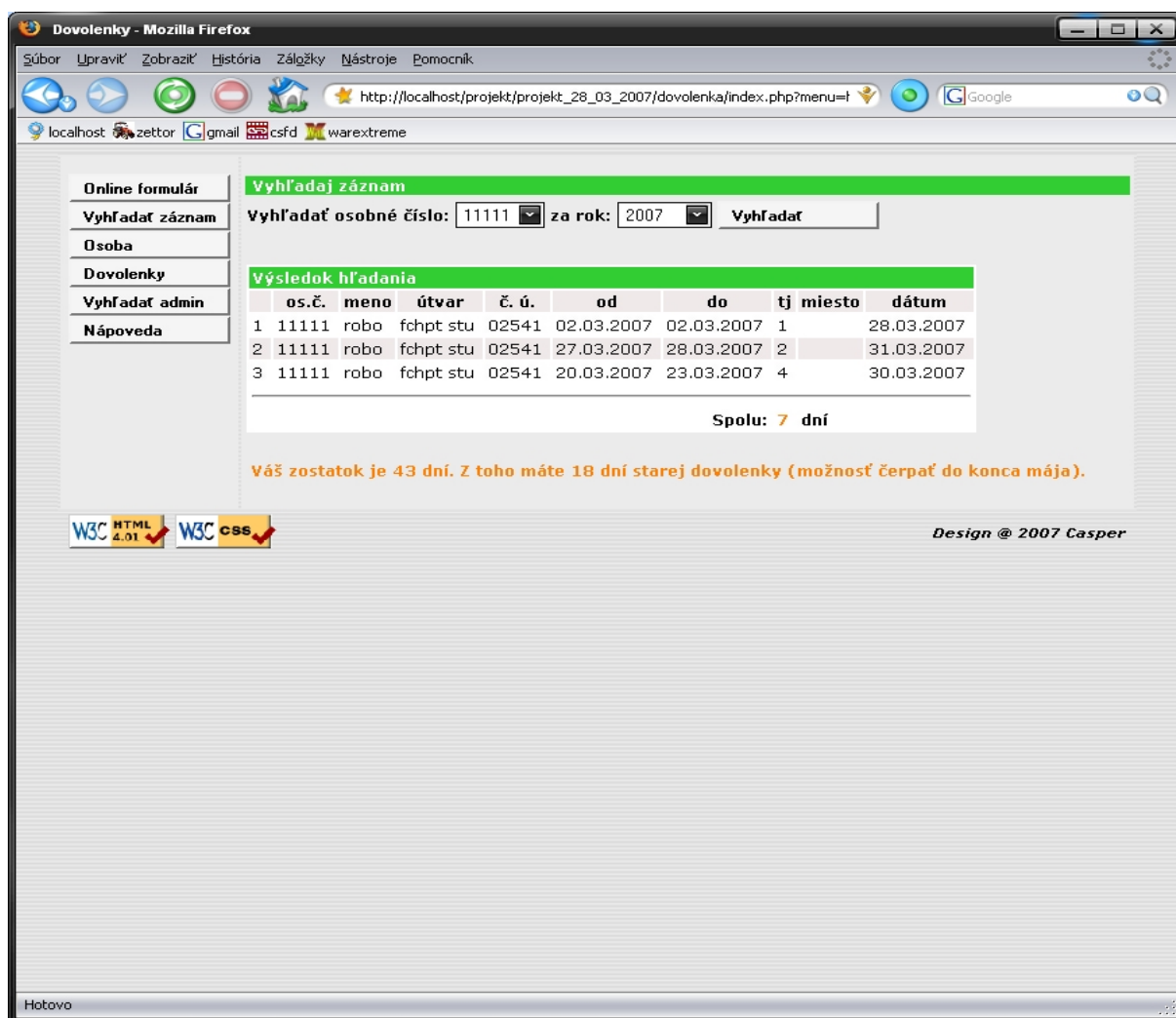
Obr. 8 Stránka „odoslane.php“

3.2.4 Vytvorenie stránky „ulozene.php“

Na tejto stránke sa nachádzajú vygenerované dovolenkové lístky z predchádzajúcej stránky a PHP skript, ktorý zabezpečuje ukladanie záznamov dovolenky do databázy. Ten najskôr skontroluje, či sa daný záznam v nej už nenachádza, či v danom čase zamestnanec už nemá vyplnenú dovolenku a nakoniec zistí, koľko dní dovolenky mu ešte zostáva. Ak má dostatočný počet zostávajúcich dní, záznam sa uloží. Ak nemá, vypíše sa koľko dní má k dispozícii a je nutné znovu vyplniť formulár. V spodnej časti je tlačidlo „Vytlačiť“. Jeho stlačením sa otvorí nové okno, kde sú už lístky pripravené na vytlačenie. Po otvorení okna je potrebné v hornom menu prehliadača kliknúť na „súbor“ a tam vybrať možnosť „vytlačiť“.

3.2.5 Vytvorenie ponuky „Vyhľadať záznam“

Táto ponuka slúži užívateľom na prehliadanie záznamov dovoleniek a je realizovaná súborom „hladaj.php“ (Obr. 9). Nachádza sa tu formulár na výber záznamov z databázy. Obsahuje dve rolety, v prvej sa vyberá osobné číslo, v druhej rok, za ktorý sa majú záznamy zobrazovať.

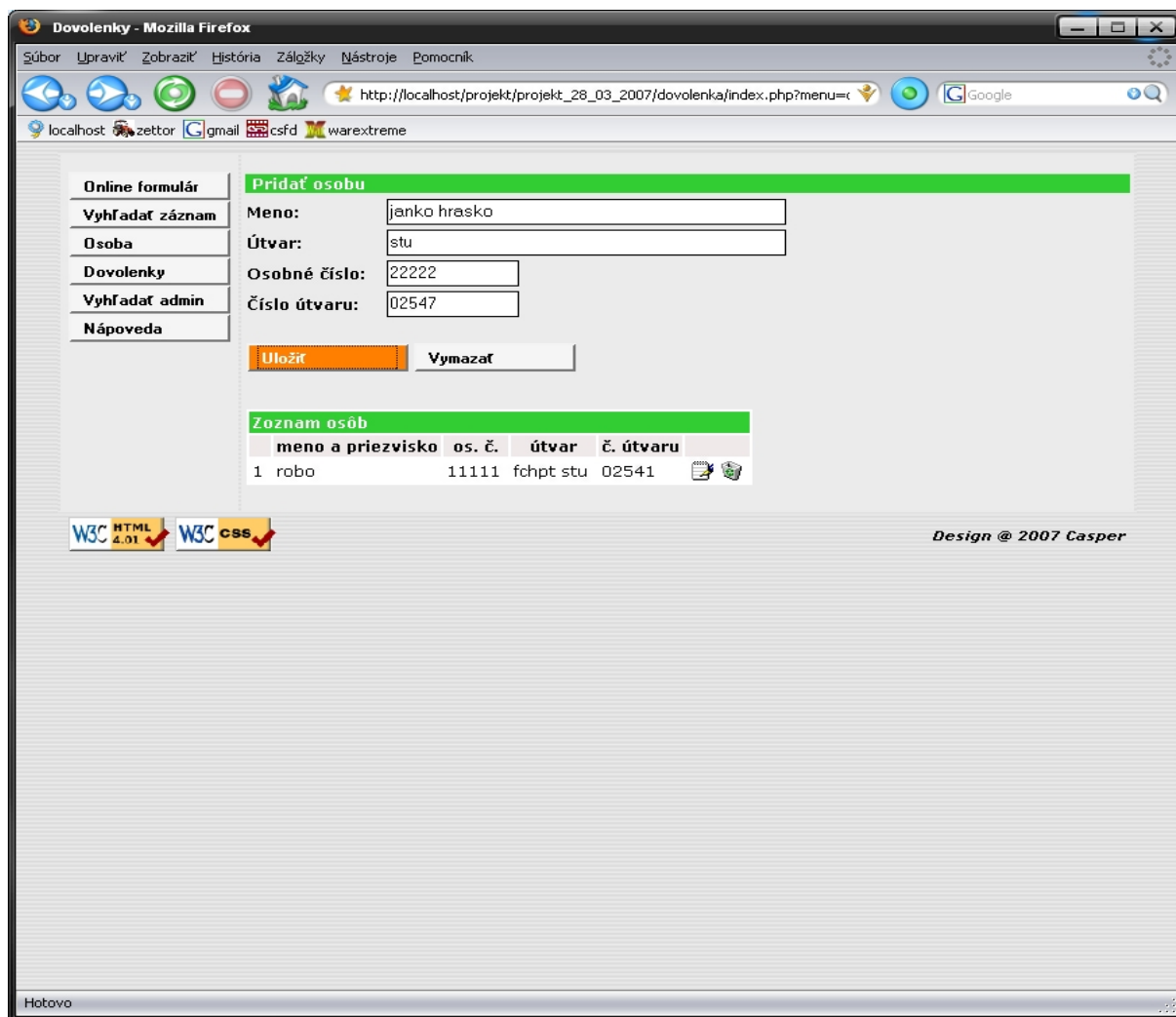


Obr. 9 Výsledok hľadania osobného čísla 11111 za rok 2007

3.2.6 Vytvorenie ponuky „Osoba“

Táto ponuka (Obr. 10) slúži administrátorovi na vkladanie osôb do tabuľky „ľudia“ v databáze. V hornej časti sa nachádza formulár na vloženie osoby. Tu sa musia vyplniť všetky polia formulára a skript zabezpečí vloženie. Po jej vložení, ďalšia časť kódu vloží dané osobné číslo aj do tabuľky „zostatok“. Pod týmto formulárom sa zobrazuje tabuľka s uloženými osobami. V nej je možné záznamy editovať (ikona ) alebo vymazať (ikona ).

Pri zmene osobného čísla osoby sa toto číslo automaticky zmení vo všetkých štyroch tabuľkách v databáze (ľudia, dovolenky, počet dní, zostatok), prípadne vymazaním sa vymažú

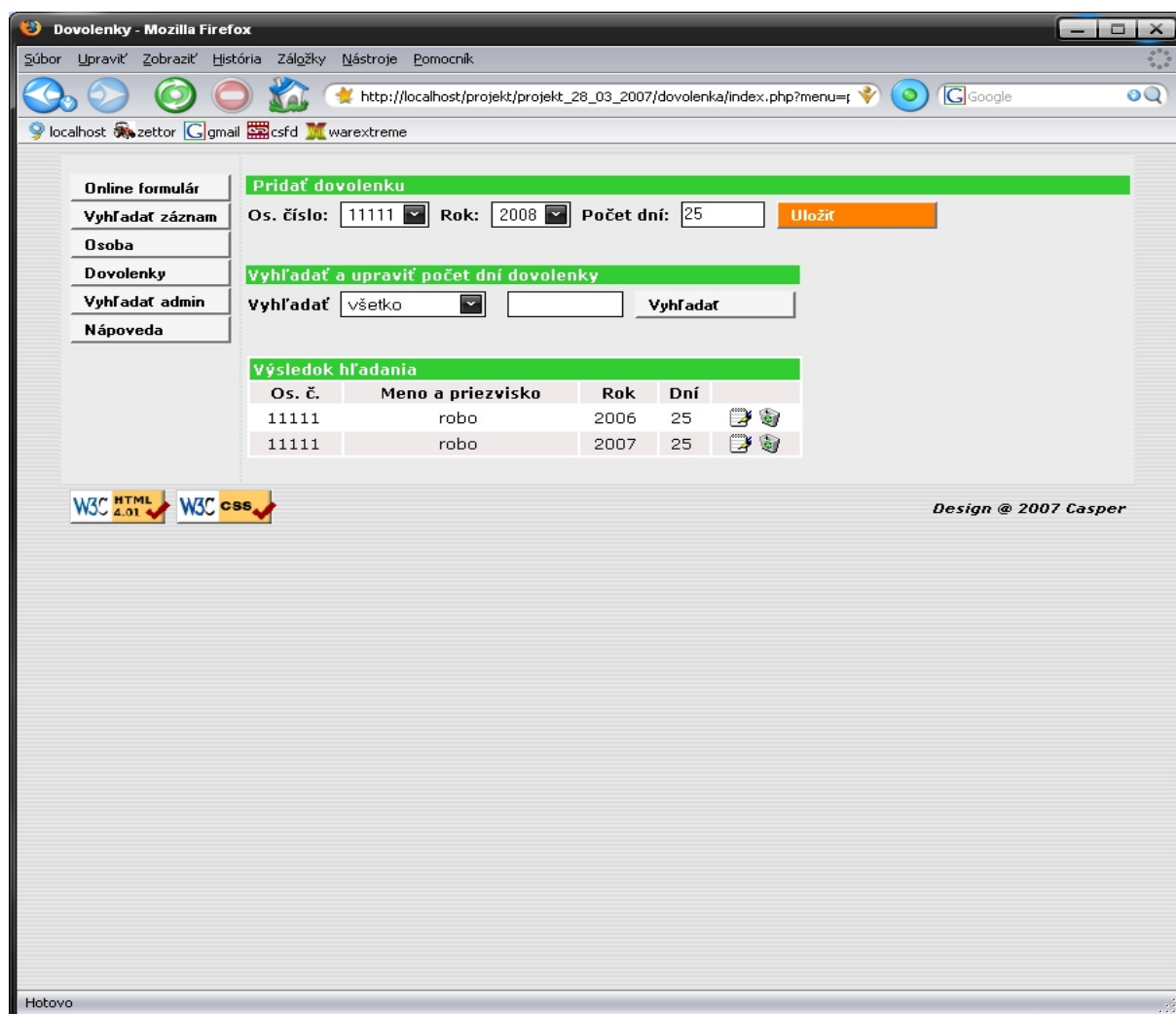


Obr. 10 Ponuka „Osoba“

3.2.7 Vytvorenie ponuky „Dovolenky“

Táto ponuka (Obr. 11) slúži na vkladanie počtu dní dovolenky, prípadne ich zmenu v tabuľke „počet dní“. Nachádzajú sa tu dva formuláre. Prvý slúži na vkladanie dní dovolenky zvlášť pre každý rok. V prvej rolete sa vyberie osobné číslo, v druhej rok, a do poľa sa napíše počet dní. Tlačidlom „Uložiť“ sa záznam uloží, ale až po overení či sa už daný záznam nenachádza v databáze. Druhý formulár slúži na vyhľadanie počtu dní dovolenky. Po zadaní vyhľadávacích kritérií sa zobrazí tabuľka so záznamami. Možnosti vyhľadávania sú:

vyhľadať všetky záznamy, vyhľadať rok alebo osobné číslo. Tu je znovu možné záznam buď upraviť (ikona ) alebo vymazať (ikona )

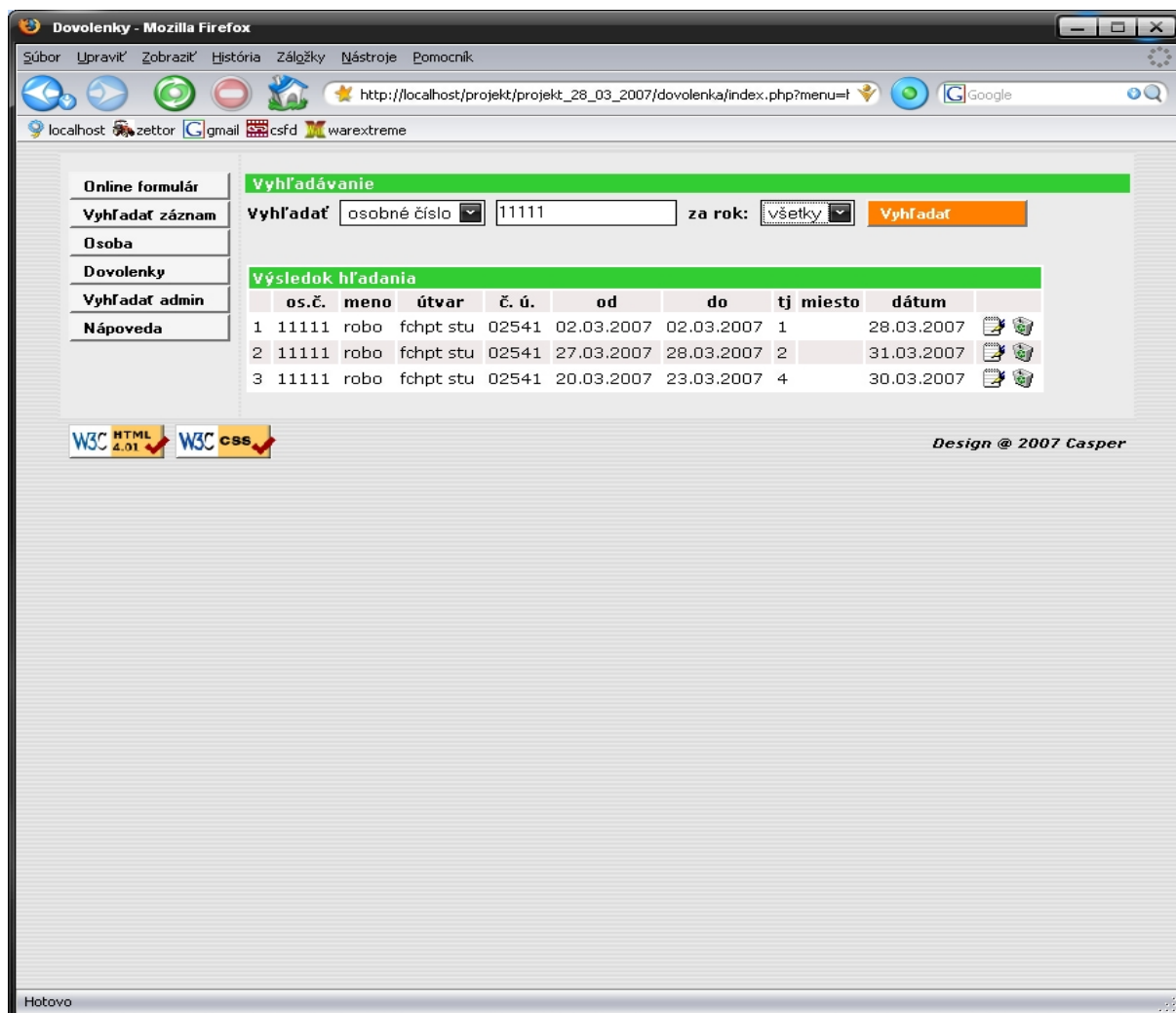


Obr. 11 Ponuka „Dovolenky“

3.2.8 Vytvorenie ponuky „Vyhľadať admin“

Táto ponuka slúži administrátorovi na prehliadanie dovolenkových lístkov a ich možné editovanie, prípadne vymazávanie. Po zadaní parametrov vyhľadávania sa zobrazí tabuľka s nájdenými záznamami (Obr. 12). Parametrami sú vyhľadanie všetkých záznamov alebo daného osobného čísla za určitý rok. Za každým záznamom je ikona  (upraviť) a ikona  (vymazať). Kliknutím na ikonku zmeniť záznam sa zobrazí stránka „uprav.php“, ktorá má rovnaký vzhľad ako „Online formulár“, kde je možné zmeniť dni odkedy a dokedy sa čerpá

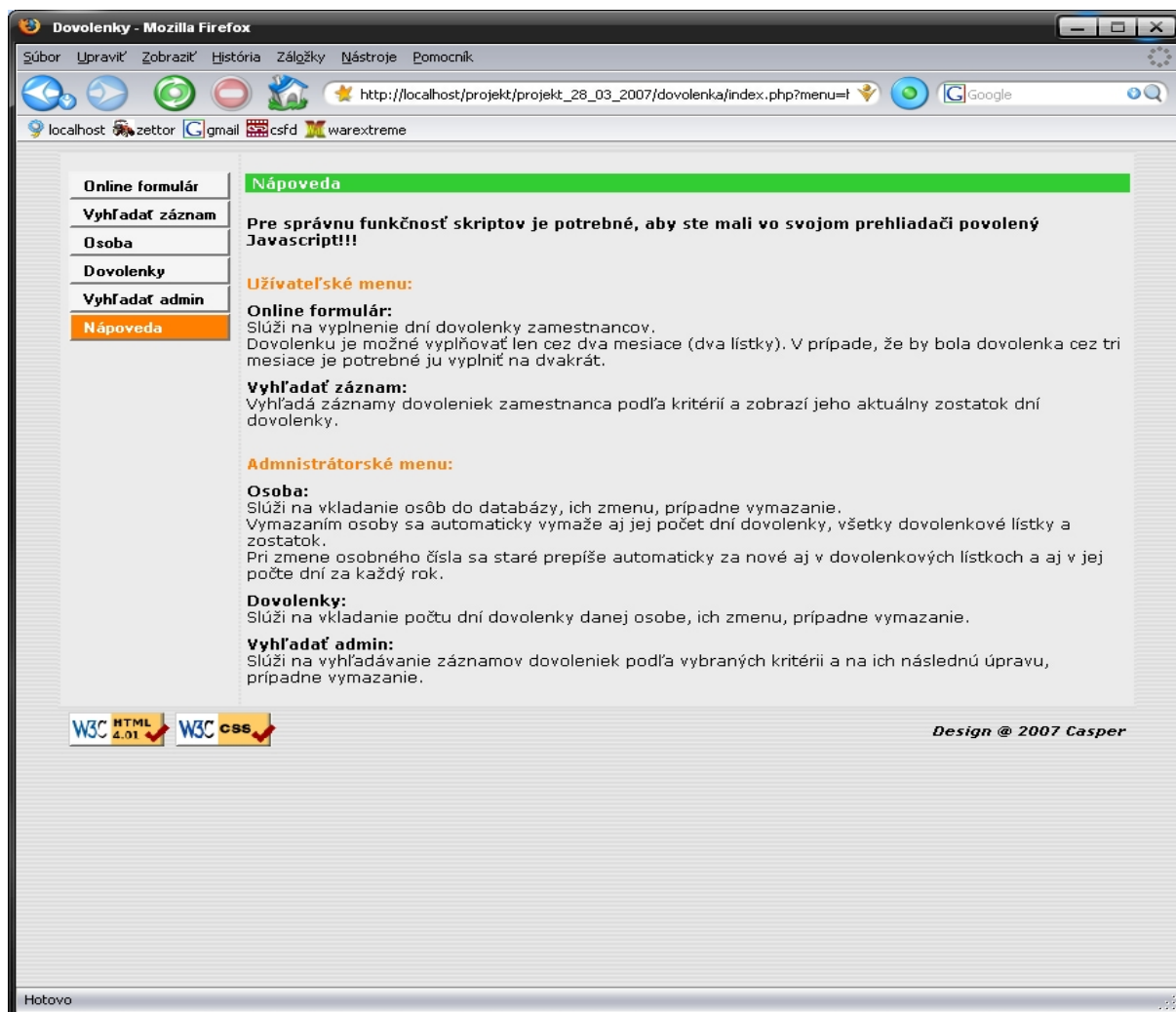
dovolenka. Zmenením údajov nastane aktualizácia záznamu v tabuľke „dovolenky“ a aj zostatku v tabuľke „zostatok“.



Obr. 12 Ponuka „Vyhľadať admin“

3.2.9 Vytvorenie ponuky „Nápoveda“

Na záver bola vytvorená ponuka „Nápoveda“, ktorá obsahuje návod na prácu s modulom dovolenka (Obr. 13).



Obr. 13 Ponuka „Nápoveda“

4. ZÁVER

Cieľom bakalárskej práce bolo vytvoriť dynamickú webovú stránku pre správu a tlač dovoleniek. Postup vytvárania stránky bol rovnaký ako je tomu v bežnej praxi. Začal sa návrhom dizajnu, vypracovania zdrojových kódov, importom údajov do databázy až po záverečné testovanie. Ako prvé boli vytvorené formuláre na vyplňovanie dovolenky spolu so vzhľadom stránky. Neskôr sa prešlo k vytvoreniu PHP kódu, ktorý spracováva informácie z formulárov a ukladá ich do databázy. Ako posledné bolo treba zabezpečiť možnosti vyhľadávania záznamov v databáze ako aj ich prípadnú úpravu a vymazávanie.

Táto stránka bude využívaná zamestnancami Oddelenia informatizácie a riadenia procesov na vyplňovanie dovolenkových lístkov prostredníctvom internetu. Jej výhoda oproti doterajšiemu ručnému vypisovaniu spočíva v možnosti vyplniť si dovolenkový lístok kedykoľvek a kdekoľvek. Ďalšou výhodou je zobrazenie aktuálneho zostatku dní starej a novej dovolenky ako aj prehľadné správy už čerpaných dovoleniek. Pozitívom je tiež fakt, že po vybratí dátumu od a do javascript automaticky odráta od rozdielu dní soboty, nedele, sviatky a v závislosti na tom či sa dovolenka čerpá v jednom mesiaci alebo cez dva sa vygeneruje príslušný počet dovolenkových lístkov. Sviatky sa každý rok automaticky generujú. Po vybratí osobného čísla sa v danom mesiaci v kalendári zobrazia už čerpané dni dovolenky.

Zabezpečenie stránok nebolo robené z dôvodu implementácie do katedrových stránok, ktoré už zabezpečenie obsahujú (login, heslo).

V budúcnosti by sa mohlo zabezpečiť schvaľovanie dovolenky prostredníctvom automatického zasielania e-mailu vedúcemu oddelenia, po ktorého schválení by sa záznam uložil do databázy. Časom by sa táto stránka mohla implementovať aj na všetky oddelenia FCHPT.

LITERATÚRA

- [1] PHP a MySQL vytváříme webové databázové aplikace, Hugh E. Williams & David Lane, Computer Press 2002, 143 00, Praha 4, <http://www.cpress.cz>, str. 9 – 16.
- [2] <http://www.php.sk/download.php>, Milan Gacik, start.pdf, kapitola 10, HTML formuláře, PHP prvé kroky.
- [3] <http://www.linuxsoft.cz>, LINUXSOFT, Informace nejen ze sveta linuxu, Copyright (c) 2003-2006, Design: www.megadesign.cz .
- [4] <http://www.ipower.sk/dbs/> , Databázové informačné systémy, RNDr. Peter Szabó.
- [5] <http://www.mysql.com>, The World's Most Popular Open Source Database, MySQL AB, © 1995-2006 MySQL AB. All rights reserved.