

# **Databázový systém půjčovny nářadí**

Database system of rental tools

Matúš Varačka

---

Bakalářská práce  
2011



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

akademický rok: 2010/2011

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Matuš VARAČKA**  
Osobní číslo: **A08099**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační a řídicí technologie**

Téma práce: **Databázový systém půjčovny náradí**

Zásady pro vypracování:

1. Analyzujte problematiku a vypracujte literární rešerši na dané téma.
2. Navrhněte strukturu databáze systému a webových stránek.
3. Vytvořte webovou aplikaci a popište postup řešení.
4. Vytvořte a prezentujte 3D model objektu.
5. Implementujte systém, zajistěte správu a zabezpečení databáze.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

1. PROKOPOVÁ, Z.: Databázové systémy MySQL+PHP. FAI UTB Zlín, s. 126, 2006, Vysokoškolská skripta. ISBN 80-7318-486-9.
2. LACKO, L. PHP a MySQL ? hotová řešení. Computer Press, 2006, ISBN: 80-251-1249-7.
3. RIORDAN, Rebecca M. Vytváříme relační databázové aplikace. Praha : Computer Press, 2000. 280 s. ISBN 80-7226-360-9.
4. SCHNEIDER, R.,D. MySQL - Oficiální průvodce tvorbou, správou a laděním databází. Grada, ISBN: 80-247-1516-3.
5. ULLMAN, L. PHP a MySQL, Computer Press, Brno, 2004,534 s. ISBN 80-251-0063-4.
6. WELLING, L., THOMSON, L. MySQL Průvodce základy databázového systému. Computer Press, Brno, 2005, ISBN: 80-251-0671-3.
7. MASLAKOWSKI, Mark. Naučte se MySQL za 21 dní. Vyd. 1. Praha : Computer Press, 2001. 478 s. ISBN 8072264486.
8. WILLIAMS, Hugh E; KRÁSENSKÝ, David; LANE, David. PHP a MySQL : vytváříme webové databázové aplikace : podrobný průvodce tvůrce WWW stránek. Vyd. 1. Praha : Computer Press, 2002. 530 s. ISBN 8072267604

Vedoucí bakalářské práce:

doc. Ing. Zdenka Prokopová, CSc.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

25. února 2011

Termín odevzdání bakalářské práce:

7. června 2011

Ve Zlíně dne 25. února 2011

prof. Ing. Vladimír Vašek, CSc.  
děkan



prof. Ing. Vladimír Vašek, CSc.  
ředitel ústavu

## ABSTRAKT

Hlavným cieľom bakalárskej práce bolo vytvoriť funkčný databázový systém pre požičovňu náradia. Bakalárska práca vo svojej teoretickej časti približuje a oboznamuje čitateľa s jednotlivými pojmami týkajúcimi sa tvorby a návrhu webovej aplikácie. V praktickej časti sú popísané jednotlivé kroky postupu tvorby webovej aplikácie, ktorá má užívateľskú a administračnú časť, do ktorej sa užívateľ prihlasuje. Webová aplikácia komunikuje s databázou skladu, čo správcovi umožňuje kontrolovať svoj sklad. Súčasťou bakalárskej práce je aj 3D model objektu, v ktorom sa predajňa nachádza.

Klíčová slova: internetové stránky, PHP, MySQL, WWW, databáza

## ABSTRACT

The main objective of my thesis was create functional database system for tools rental. Bachelor thesis in it is theoretical part approaches and introduces a reader with several concepts related to creating and designing web applications. The practical part describes single steps of building web application that contains user and administration section where is a user logged in. The web application communicates with a database of store, which allows administrators to control their warehouse. Part of the thesis is a 3D object model, in which the store is located.

Keywords: web sites, PHP, MySQL, WWW, database

Ďakujem vedúcej mojej bakalárskej práce doc. Ing. Zdenke Prokopovej, CSc. z Ústavu počítačových a komunikačných systémov za odbornú pomoc. Ďalej ďakujem svojej rodine, svojej priateľke a kamarátom za poskytnutú podporu.

**Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

|  |           |
|--|-----------|
| <b>ÚVOD.....</b>                               | <b>9</b>  |
| <b>I TEORETICKÁ ČÁST .....</b>                 | <b>10</b> |
| <b>1 ANALÝZA SÚČASNÉHO STAVU.....</b>          | <b>11</b> |
| <b>2 PROGRAMOVACIE JAZYKY.....</b>             | <b>12</b> |
| 2.1 PHP.....                                   | 12        |
| 2.1.1 Zoznámenie s jazykom PHP .....           | 12        |
| 2.1.2 Prednosti.....                           | 13        |
| 2.1.3 Programovanie a funkčnosť PHP .....      | 13        |
| 2.1.4 Operátory.....                           | 14        |
| 2.1.5 Podmienky a cykly .....                  | 18        |
| 2.1.6 Metódy GET, POST a SESSION .....         | 22        |
| 2.2 MYSQL.....                                 | 23        |
| 2.2.1 Priblíženie pojmu MySQL .....            | 24        |
| 2.2.2 Prečo používať MySQL .....               | 24        |
| 2.2.3 Fungovanie webovej databáze.....         | 25        |
| 2.2.4 Príkazy v MySQL.....                     | 26        |
| <b>3 POUŽITÉ PROGRAMY .....</b>                | <b>28</b> |
| 3.1 PSPAD EDITOR .....                         | 28        |
| 3.2 EASY PHP.....                              | 29        |
| 3.2.1 Spustenie EasyPHP .....                  | 29        |
| 3.2.2 Server Apache .....                      | 30        |
| 3.3 CORELDRAW X4 .....                         | 31        |
| <b>II PRAKTICKÁ ČÁST .....</b>                 | <b>32</b> |
| <b>4 POSTUP PRI NÁVRHU APLIKÁCIE .....</b>     | <b>33</b> |
| 4.1 NÁVRH DATABÁZE .....                       | 33        |
| 4.2 ROZLOŽENIE STRÁNKY .....                   | 36        |
| 4.2.1 Užívateľská časť.....                    | 37        |
| 4.2.2 Administračná časť .....                 | 38        |
| 4.3 GRAFIKA A FUNKČNOSŤ STRÁNKY .....          | 39        |
| 4.3.1 Hlavička stránky.....                    | 40        |
| 4.3.2 Menu a stredný priestor .....            | 40        |
| <b>5 UŽÍVATEĽSKÁ ČASŤ .....</b>                | <b>42</b> |
| 5.1 JEDNOTLIVÉ POLOŽKY VERTIKÁLNEHO MENU ..... | 42        |
| 5.1.1 Podstránka Nabízime .....                | 43        |
| 5.1.2 Podstránka Vyhľadávani.....              | 44        |
| <b>6 ADMINISTRAČNÁ ČASŤ .....</b>              | <b>46</b> |

|          |   |           |
|----------|---|-----------|
| 6.1      | PRIHLÁSENIE SA DO ADMINISTRAČNEJ ČASTI .....    | 46        |
| 6.2      | STREDOVÝ PANEL, OVLÁDACIE TLAČIDLÁ .....        | 47        |
| 6.3      | HLAVNÉ MENU .....                               | 49        |
| 6.3.1    | Výpis tabuliek .....                            | 50        |
| 6.3.2    | Zmeniť záznam, vymazané položky.....            | 51        |
| 6.3.3    | Užívateľ.....                                   | 51        |
| <b>7</b> | <b>3D MODEL PREDAJNE.....</b>                   | <b>53</b> |
|          | <b>ZÁVĚR .....</b>                              | <b>55</b> |
|          | <b>ZÁVĚR V ANGLIČTINĚ.....</b>                  | <b>56</b> |
|          | <b>SEZNAM POUŽITÉ LITERATURY.....</b>           | <b>57</b> |
|          | <b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b> | <b>58</b> |
|          | <b>SEZNAM OBRÁZKŮ .....</b>                     | <b>59</b> |
|          | <b>SEZNAM TABULEK.....</b>                      | <b>60</b> |



## ÚVOD

V dnešnej dobe, kedy sa bez pripojenia na internet už pomaly ani nezaobídeme, množstvo firiem využíva možnosť svojej propagácie a reklamy práve prostredníctvom internetu.

Je to pomerne lacná forma reklamy, ktorá ak je vhodne aplikovaná, dokáže pôsobiť na širokú cieľovú skupinu ľudí. Navyše má zákazník takmer vždy možnosť túto stránku navštíviť a dozvedieť sa, čo potrebuje a kedy potrebuje.

Je bežné, ak niekto z vášho okolia povie, že si kúpil niečo cez internet, poprípade idem si pozrieť autobusový spoj domov práve cez internet. Túto službu umožňujú webové stránky, ktoré sú vybavené e-shopom či databázou. Je to výhodné ako pre firmy, tak pre zákazníkov a konečných spotrebiteľov. Stránky s databázou nám umožnia vybrať si konkrétny produkt na stránke firmy, konkrétny spoj v prípade vyhľadávateľov cestovného poriadku alebo konkrétne vlastnosti osoby v prípade internetových „zoznamiek“.

Návštevník navštívi stránku a pomocou preddefinovaných tlačidiel môže „ovládať“ databázu a vybrať údaje, ktoré sú v nej uložené. Zadaním alebo odkliknutím údajov, ktoré hľadá, sa mu z databázy zobrazia položky, o ktoré má záujem. Môže to byť spoj, ktorý ide zo zadaného mesta v zadanom čase, skupinka ľudí s hnedými očami alebo výrobok, ktorý si chce zakúpiť.

Práve túto možnosť, aby si mohol zákazník overiť dostupnosť ním hľadaného tovaru na sklade, dozvedieť sa jeho cenu a pre majiteľa firmy možnosť mať svoj sklad „pod kontrolou“, sa rozhodla aj firma PŮJČOVNA FILIPOVSKÝ.

## **I. TEORETICKÁ ČÁST**

## 1 ANALÝZA SÚČASNÉHO STAVU

Počas letných prázdnin som pracoval vo firme Půjčovna Filipovský. Firma sa zaoberá hlavne požičiavaním a predajom pracovného náradia. Pre rozšírenie pôsobnosti sa firma rozhodla predávať aj stavebný materiál, veci do domácnosti a do záhrady tj. celkovo rozšírila svoj sortiment.

Všimol som si, že každý deň prišlo niekoľko desiatok ľudí, ktorí sa pýtali či máme daný výrobok, poprípade koľko stojí, ak bol požičaný, tak kedy ho vrátia a podobne. Často to končilo tým, že si predajca a zákazník vymenili telefónne čísla a telefonicky si oznamovali, ako na tom daný výrobok je, či už je vrátený, kedy príde, koľko bude stáť a ďalšie podobné otázky boli na dennom poriadku. Na moju otázku, či by bolo pre nich jednoduchšie pozrieť na internet, vyhľadať si zadaný výrobok a kontrolovať jeho stav priamo na stránke, odpovedali jednoznačne áno. To ma priviedlo k tomu, že som sa rozhodol pre firmu Půjčovna Filipovský takúto webovú stránku navrhnuť a zrealizovať.

## 2 PROGRAMOVACIE JAZYKY

### 2.1 PHP

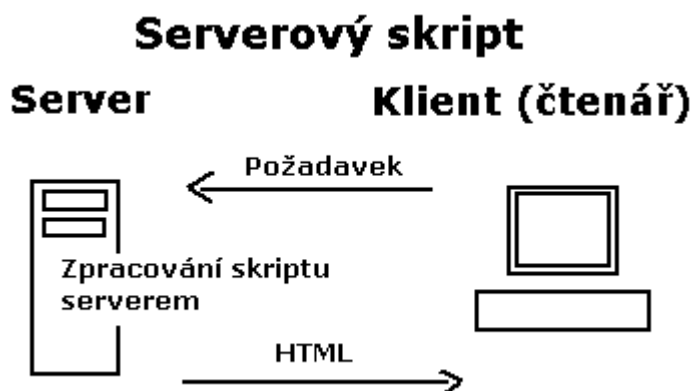
PHP je programovací jazyk, ktorý je označovaný ako dynamický. Od jazykov ako napríklad HTML sa odlišuje tým, že užívateľ nevidí jeho štruktúru označovanú ako vsuvka, ktorá je ukrytá medzi *tagmi* typickými pre jazyk PHP. Tieto súbory majú príponu \*.php a pri požiadavke na php stránku ich server rozpozná a postupne prechádza. Po vyhodnotení a spracovaní dotazov sa klientovi opäť zobrazí iba HTML s vykonaným dotazom z PHP.

#### 2.1.1 Zoznámenie s jazykom PHP

Vznik PHP sa datuje od roku 1994 ako výsledok práce pôvodne jedného autora, ktorý sa volal Rasmus Lerdorf. V neskoršej dobe ho rôznymi úpravami prerábali rôzni talentovaní ľudia. Celkovo si jazyk PHP prešiel tromi veľkými úpravami a prerábaním, vďaka ktorým má dnešnú podobu.

PHP patrí medzi *Open Source* produkty čo znamená, že zdrojový kód je plne dostupný. Zdrojový kód môžete voľne používať, sťahovať, distribuovať, a to všetko bez poplatku.

Ak sa rozhodneme vytvoriť PHP stránku, potrebujeme ju umiestniť na server, ktorý PHP podporuje.



Obr. 1. Komunikácia serveru

### 2.1.2 Prednosti

Najväčšou konkurenciou PHP sú Microsoft Active Server Pages označovaný ako *ASP*, *Java Server Pages* – *JSP* a *Perl*.

Pri porovnaní so spomínanou konkurenciou má PHP niekoľko hlavných výhod, medzi ktoré bez pochyb patria:

- Rozhranie pre veľké množstvo databázových systémov
- Vysoký výkon
- Nízke náklady
- Zabudované knižnice pre bežné webové úlohy
- Ľahké použitie a učenie
- Kompatibilita

PHP je schopné pripojiť sa k veľkému množstvu databázových systémov, ako sú napríklad *MySQL*, *mSQL*, *Oracle*, *PostgreSQL* či *Informix* bez akéhokoľvek medzičlánku.

Vďaka svojej efektívnosti a výkonnosti dokáže aj veľmi jednoduchý a nenáročný server obslúžiť niekoľko miliónov požiadaviek denne. Vďaka výkonnostným testom, ktoré zverejnila firma Zend Technologies sa preukázalo, že dokáže poraziť svojich konkurentov.

Keďže je PHP úplne zadarmo a je voľne stiahnuteľné, dá sa povedať, že patrí do kategórie nízko nákladových jazykov.

### 2.1.3 Programovanie a funkčnosť PHP

Od začiatku bol navrhovaný na využitie vo webových aplikáciách a práve preto obsahuje množstvo funkcií, ktoré sú určené k plneniu webových úloh. Vďaka jednoduchému kódu, na ktorý nám stačí pár riadkov nám umožňuje prácu s *cookies*, odosielať e-maily alebo generovať dokumenty vo formáte PDF. Naučiť sa programovať v jazyku PHP je pomerne jednoduché, ak ovládáte programovanie v jazykoch ako napríklad jazyk *C*, *C++*, *Java* či *Perl*. Kód je možné písať pod rôznymi operačnými systémami a kód bude vo väčšine prípadov fungovať bez akýchkoľvek problémov a práve táto vlastnosť robí PHP kompatibilným. [1]

Ukážka časti kódu s vnoreným PHP kódom:

```
<body>
<h1>Matematické operácie</h1>
<?php
echo "vedľa tohto textu sa vypíšu zadane čísla ";
$a = 3;
$b = 5;
$c = $a + $b;
echo $a ;
echo " a ";
echo $b;
echo ", súčet čísel je ";
echo $c;
?>
</body>
```

Tento kód bude na výstupe na obrazovku vyzeráť nasledovne:



*Obr. 2. Výstup na obrazovku*

Ako je z obrázku (Obr. 2) vidieť, na obrazovku sa vypísal text z *HTML* a samotný výstup z *PHP*.

#### **2.1.4 Operátory**

Operátory využívame pri vytváraní rôznych operáciách s premennými alebo s hodnotami. Ako operátory môžeme používať jednotlivé symboly alebo kombináciu týchto symbolov podľa danej formy. Operátory môžeme rozdeliť do niekoľkých skupín, a to:

- Aritmetické operátory
- Operátory reťazcov
- Priradovací operátor (jednoduchý a kombinovaný)
- Porovnávací operátor
- Logické operátory

## Aritmetické operátory

Aritmetické operátory sú bežné matematické operátory, ktoré sa využívajú pri základných matematických operáciách. Ako je možné vidieť z nasledujúcej tabuľky, tieto operátory sú ľahko pochopiteľné.

Tab. 1. Príklady a názvy aritmetických operátorov

| Operátor | Názov operácie             | Príklad      |
|----------|----------------------------|--------------|
| +        | Sčítanie                   | $\$x + \$y$  |
| -        | Odčítanie                  | $\$x - \$y$  |
| *        | Násobenie                  | $\$x * \$y$  |
| /        | Delenie                    | $\$x / \$y$  |
| %        | Modulus (zvyšok po delení) | $\$x \% \$y$ |

Výsledok jednotlivých operácií môžeme uložiť do zvolenej premennej. V tomto prípade premenná s názvom `vysledok` bude mať zápis `$vysledok = $x * $y;`.

Pri programovaní v jazyku PHP sa operátory pre násobenie a delenie nezapisujú spôsobom, na ktorý sme možno zvyknutý z matematických úkonov, a to použitím znakov bodka pre násobenie a dvojbodka pre delenie, ale operátormi `*` respektíve `/`.

Pomocou operátoru *modulus* dostaneme hodnotu zvyšku po delení dvoch hodnôt. Funkciu modulus si ukážeme v nasledujúcom príklade:

```
$x = 16;
$y = 3;
$vysledok = $x % $y;
```

Do premennej `$vysledok` sa uloží hodnota 1. Ak vydělíme tieto čísla, dostaneme hodnotu 5, a zvyšok čísla 1.

## Prirad'ovacie operátory

Prirad'ovacie operátory sa dajú rozdeliť na jednoduché a kombinované. Ako samotný názov napovedá, pri kombinovaných prirad'ovacích operátoroch pôjde o kombináciu samotných operátorov. Medzi jednoduché patrí operátor „`=`“. Pomocou tohto operátora dokážeme niektorej premennej priradiť konkrétnu hodnotu alebo hodnotu z výrazu. Príklad:

```
$c = 5 + ($x = 3) * 2;
```

V tomto prípade podobne ako pri matematických operáciách, zátvorka zvyšuje prioritu, čo znamená, že výraz v zátvorke sa prevedie ako prvý.

Operátory kombinovaného priradenia sú znázornené v tabuľke (Tab. 2.). Ako si môžete všimnúť, zápis pozostáva z kombinácie dvoch operátorov, kde prvý operátor znamená operáciu, ktorá sa prevedie medzi opakovanou a novou premennou.

Tab. 2. Príklady a názvy jednotlivých kombinovaných operátorov

| Kombinácia operátorov | Príklad zápisu          | Zápis je ekvivalentný s      |
|-----------------------|-------------------------|------------------------------|
| <code>+=</code>       | <code>\$a += \$b</code> | <code>\$a = \$a + \$b</code> |
| <code>-=</code>       | <code>\$a -= \$b</code> | <code>\$a = \$a - \$b</code> |
| <code>*=</code>       | <code>\$a *= \$b</code> | <code>\$a = \$a * \$b</code> |
| <code>/=</code>       | <code>\$a /= \$b</code> | <code>\$a = \$a / \$b</code> |
| <code>%=</code>       | <code>\$a %= \$b</code> | <code>\$a = \$a % \$b</code> |
| <code>.=</code>       | <code>\$a .= \$b</code> | <code>\$a = \$a . \$b</code> |

### Porovnávacie operátory

Výsledkom predchádzajúcich operácií boli konkrétne hodnoty. Pri použití porovnávacích operátorov sa do výsledku priraduje hodnota *true* alebo *false*. Inými slovami je to hodnota, ktorá môže nadobúdať iba dva stavy, a to podľa toho, či danú podmienku výrazu spĺňa – *true*, alebo nespĺňa – *false*. Tieto operátory sa často vyskytujú v podmienkach či cykloch. Častou chybou je, že často používaný porovnávací operátor `==` sa často zamieňa s operátorom priradenia `=`. Výsledkom tejto chyby dostávame taktiež dva stavy *true* a *false*. V prípade, že do premennej priradíme nulovú hodnotu, výsledkom je *false*, v opačnom prípade *true*. [1]



Tab. 3. Příklady a názvy jednotlivých porovnávacích operátorů

| Porovnávací operátor  | Názov                    | Příklad použití               |
|-----------------------|--------------------------|-------------------------------|
| <code>==</code>       | Rovnost                  | <code>\$a == \$b</code>       |
| <code>!=</code>       | Nerovnost                | <code>\$a != \$b</code>       |
| <code>&lt;&gt;</code> | Ekvivalent pre nerovnosť | <code>\$a &lt;&gt; \$b</code> |
| <code>&lt;</code>     | Menší ako                | <code>\$a &lt; \$b</code>     |
| <code>&gt;</code>     | Väčší ako                | <code>\$a &gt; \$b</code>     |
| <code>&lt;=</code>    | Menší alebo rovný ako    | <code>\$a &lt;= \$b</code>    |
| <code>&gt;=</code>    | Väčší alebo rovný ako    | <code>\$a &gt;= \$b</code>    |

### Logické operátory

Logické operátory používame v prípade, keď potrebujeme zistiť kombináciu jednotlivých logických výsledkov. Ak potrebujeme zistiť, či sa dané číslo nachádza na nejakom intervale, môžeme namiesto niekoľkých porovnávacích operátorov využiť zápis pomocou jedného logického operátora.

Příklad: Máme zadané číslo v premennej `$c` a potrebujeme zistiť, či sa toto číslo nachádza v rozmedzí medzi číslami 10 a 20. Bez použitia logického operátora by sme sa snažili zistiť výsledok pomocou niekoľkých podmienok, a to nasledovne:

```
$c > 10; a následne či $c < 20;
```

V takomto prípade sa k rovnakému výsledku dostaneme aj pri použití jediného operátora AND. Výsledkom týchto operátorov je taktiež bitová hodnota *true* a *false*. Rovnaký príklad sa dá pomocou logického operátora zapísať nasledovne:

```
$c > 10 && $c < 20;
```

Tieto operátory sa zapisujú medzi jednotlivé premenné, napríklad v tvare `$c AND $d`.

Tab. 4. Příklady, názvy a hodnoty výsledku jednotlivých logických operátorů

| Operátor                | Názov | Hodnota výsledku  |
|-------------------------|-------|---|
| <code>!</code>          | NOT   | Vracia opačnú hodnotu zadaného.                                 |
| <code>&amp;&amp;</code> | AND   | Ak sú oba operátory true, vracia true, v opačnom prípade false. |
| <code>  </code>         | OR    | Vracia true, ak aspoň jeden z dvoch operátorov sa rovná true.   |
| <code>and</code>        | AND   | To isté ako <code>&amp;&amp;</code> , ale s nižšou prioritou.   |
| <code>or</code>         | OR    | To isté ako <code>  </code> , ale s nižšou prioritou.           |

### 2.1.5 Podmienky a cykly

Podmienkové príkazy umožňujú vetvenie do rôznych príkazov, ktoré sa uskutočnia pri splnení respektíve nesplnení niektorej z podmienok. Podmienka sa zapisuje pomocou niektorých z operátorov, najčastejšie však pomocou porovnávacích operátorov. Do podmienky sa zapisuje, čo sa ma vykonať ak úvodná podmienka bola splnená a naopak čo v prípade, ak splnená nebola. V oboch prípadoch môže za prvotnou podmienkou nasledovať ďalšia a takýmto spôsobom sa dá vytvoriť rozsiahle riadenie a rozvetvenie programu.

Pre vetvenie sa v jazyku PHP vyskytujú dva spôsoby, a to buď pomocou `if`, ktorý má nepovinný dodatok `else` využívaný v prípade, ak sa podmienka nesplní a príkaz `switch`, ktorý sa do vetvenia zapisuje pomocou `case`.

#### Podmienkový príkaz `If – else`

Použitie príkazu `If – else` je rovnaké ako pri všetkých programovacích jazykoch. Na začiatku podmieneného príkazu sa testuje prvotná podmienka. Po overení podmienky nasleduje jeden alebo skupina príkazov, ktoré sa vykonajú v prípade, že prvotná podmienka bola splnená. Táto časť sa nazýva časťou *if-true*. Za touto časťou sa môže doplniť časť *if-false*, ktorá sa zapisuje pomocou príkazu `else`. Za týmto príkazom sa takisto môže nachádzať jeden alebo viac príkazov, ktoré sa vykonajú v prípade, že prvotná podmienka splnená nebola. Syntax podmienky `If – else` si ukážeme na nasledujúcom príklade:

```
If ($premenna > 5)
    echo "premenná je väčšia ako číslo 5";
else
{
    echo "premenná je menšia alebo rovná ako 5";
    echo „ak používame viac príkazov, sú v {}“;
}
```

Zložitejšie rozvetvenie s využitím viacerých testovacích podmienok si ukážeme na rovnakom príklade:

```
If ($premenna > 5)
    echo "premenná je väčšia ako číslo 5";
else {
    if($premenna == 5)
        echo "premenná je rovná číslu 5";
    else
        echo „premenná je menšia a nerovná číslu 5“;
}
```

V prvotnej podmienke testujeme, či je premenná väčšia ako číslo 5. V prípade, že áno, pomocou `echo` sa vypíše text, že premenná je väčšia. V prípade, že úvodná podmienka nie je splnená, prechádzame k príkazu `else`. Za týmto príkazom nasleduje viac príkazov, preto ich uzavrieme do zložených zátvoriek. V `else` testujeme ďalšou podmienkou, či je premenná rovná číslu 5. Ak áno, vypíše sa text, že nastala rovnosť. V opačnom prípade sa vypíše, že premenná nie je rovná a na základe prvej podmienky nie je ani väčšia ako číslo 5. Týmto spôsobom sa dá naprogramovať jednoduchá hra s názvom *Hádaj na aké číslo myslím*. Tento zložitý zápis sa dá zapísať aj jednoduchšie, pomocou logických operátorov.

Pomocou logických operátorov môžeme do prvotnej podmienky uviesť niekoľko testovaných podmienok. Môžeme tak zložitý a zdĺhavý kód previesť na stručný kód čím sa vyhneme novej chybe pri zápise. Takýto príkaz môže vyzeráť nasledovne:

```
if ($a == 3) || ($a == 9)
    echo "premenná a, je rovná hodnote 3 alebo 9";

if ($a == 3) || ($b == 9)
    echo "premenná a, je rovná hodnote 3 alebo
    premenná b je rovná hodnote 9";
```

### Podmienkový príkaz switch

Príkaz `switch` sa používa pri výbere zo zoznamu úkonov ako aj pri vetvení do viac alternatív. Je zadáný príkaz `switch`, za ktorým môže nasledovať napríklad premenná s určitou hodnotu. Pomocou jednotlivých `case` sa testuje, ktorú podmienku premenná spĺňa. Každý `case` obsahuje príkaz alebo skupinu príkazov, ktoré sa po splnení vykonajú. Na záver nasleduje príkaz `break`; pre ukončenie konkrétneho príkazu v `case`.

Príklad na testovanie hodnoty v premennej:

```
switch ($premenna){
    case 1:
        echo "Zvolené číslo 1";
        break;
    case 2:
        echo "Zvolené číslo 2";
        break;
    default:
        echo "Zvolené číslo bolo iné ako uvedené";
}
```

## Cykly

Cykklus nám umožňuje opakovane prevádzkať niektorý z príkazov až do splnenia podmienkového výrazu. V jazyku PHP sa dajú používať celkovo štyri typy cyklov. Medzi tieto typy patrí *while*, *do – while*, *for* a nový cyklus *foreach* ktorý sa používa hlavne pri poliach.

### Cykklus while

Cykklus *while* využíva najjednoduchšiu štruktúru zápisu. Pri tomto cykle sa zadá vstupná podmienka, za ktorou nasleduje zápis jedného alebo viacerých príkazov. Tieto príkazy sa vykonávajú v cykloch stále dokola, dokiaľ nie je úvodná podmienka splnená. V prípade, že použijeme viacero príkazov, musíme ich zapísať a oddeliť v zložených zátvorkách. Použitie cyklu *while* môže vyzeráť nasledovne:

```
$pocitadlo = 1;
while ($pocitadlo < 11)
{
    echo $pocitadlo;
    echo " ";
    $pocitadlo++; //zvýšenie hodnoty počítadla
}
```

V príklade je nastavená hodnota premennej *\$pocitadlo* na hodnotu 1. Vo vstupnej podmienke sa testuje, či hodnota tejto premennej je menšia ako 11. V prípade že áno, prechádza sa k príkazom. V tomto prípade sa na obrazovku vypíše aktuálna hodnota premennej, oddelí sa medzerou a na záver sa táto hodnota zvýši a opäť sa porovná vstupná podmienka. V prípade, že je podmienka aj naďalej splnená a teda, že premenná je menšia

ako číslo 11, pokračuje sa v prevádzaní príkazov až do doby, kedy podmienka splnená nie je. Na výstupe z tohto cyklu by sa na obrazovku vypísali všetky hodnoty od čísla 1 až po číslo 10.

### Cyklus do – while

Rozdiel medzi týmto príkazom a príkazom *while* spočíva iba v mieste kontroly podmienky. V prípade *while* sa otestovala podmienka a pri splnení podmienky sa zavola príkaz alebo skupina príkazov, ktoré sa vykonávali do doby, kedy podmienka splnená nebola. V prípade použitia cyklu *do – while* sa podmienka testuje až po ukončení jednotlivých príkazov. Telo s príkazmi sa opäť vykonáva do doby, kedy podmienka prestane platiť. Na príklade je znázornený rovnaký príklad ako v pri cykle *while*.

```
$pocitadlo = 1;
do
{
    echo $pocitadlo;
    echo " ";
    $pocitadlo++; //zvýšenie hodnoty počítadla
}
while ($pocitadlo < 11);
```

V tomto prípade sa najskôr na obrazovku vypísala hodnota premennej, zvýšila sa jej hodnota a až na záver sa testovala podmienka, či sa má tento cyklus vykonať znova, alebo či sa má ukončiť.

### Cyklus for

Tento príkaz zo spomínanej štvorice cyklov má najzložitejší zápis. Hlavička cyklu sa skladá z troch častí, ktoré sú od seba oddelené bodkočiarkou. Žiadna časť však nie je povinná. Za hlavičkou nasledujú inicializačné príkazy, ktoré sa prevádzajú len raz, a to pred prvým vstupom do tela cyklu. Takýto cyklus môže mať napríklad nasledovný zápis:

```
for($pocitadlo=1; $pocitadlo < 11; $pocitadlo++ )
{
    echo $pocitadlo;
    echo " ";
}
```

V tomto zápise cyklu sa opäť budú vypisovať na obrazovku hodnoty premennej od čísla 1 až po číslo 10 oddelené medzerou. V hlavičke cyklu sa nachádzajú tri časti. V prvej

časti sa zadáva počiatočná hodnota, do druhej časti sa zapisuje konečná podmienka cyklu a v záverečnej tretej fázy sa vyskytuje operácia, ktorá sa má s premennou previesť.

### Cyklus foreach

Patrí medzi „mladíkov“ medzi jednotlivými cyklami. Bol novo zavedený do verzie PHP4 a predstavuje komfortný spôsob pri prehľadávaní poľa. Najlepšie vysvetlenie tohto cyklu je na príklade.

```
$hodnoty_dlzok = array (2, 9, 13, 15);  
foreach($hodnoty_dlzok as $cm)  
{  
    $inch = $cm / 2.54;  
    echo „$cm centimetrov = $inch palcov\n“;  
}
```

V poli *\$hodnoty\_dlzok* sú uložené ľubovoľné hodnoty, ktoré chceme premeniť z dĺžky v centimetroch na dĺžku v palcoch. V cykle sme si tieto jednotlivé hodnoty pomenovali pomocou *as* ako premennú *\$cm*. Telo cyklu tvoria príkazy, ktoré postupne prechádzajú pole a každú hodnotu vydedia číslom 2,54. Následne sa vypíše pôvodná hodnota a prepočítaná hodnota. Tento cyklus sa opakuje až po vykonanie operácií na všetkých hodnotách z poľa. [2]

### 2.1.6 Metódy GET, POST a SESSION

Metóda určuje, akým spôsobom sa budú dáta odosielať k ďalším skriptom. Rozlišujeme dve metódy a to metódu *GET* a metódu *POST*.

#### **Metóda GET**

Dáta, ktoré sú odoslané pomocou metódy *GET*, sú prepísané a zobrazia sa v URL adrese. Tieto dáta sú oddelené otáznikom a sú viditeľné. Táto metóda nie je vhodná pre posielanie citlivých informácií. Využíva sa pri posielaní dát z krátkych formulárov.

#### **Metóda POST**

Táto metóda je vhodná pre posielanie rozsiahlejších dát. Dáta sa nepridávajú do samotnej URL, čo znamená, že nie sú vidieť. Dáta sú posielané ako samostatný HTTP objekt.

Jednoduchý formulár, ktorý ma jedno vstupné pole a ktorý dáta odosiela pomocou metódy *POST*, sa môže zapísať nasledovne:

```
<form action="skript_kde_data_chcem.php" method="post">  
<input type="text" name="trieda"></input>  
<input type="submit" value="ULOŽIT" name="submit">  
</form>
```

The image shows a screenshot of a web form. It consists of a single text input field with a light gray border and a light gray background. To the right of the input field is a submit button with a light gray border and a light gray background, containing the text 'ULOŽIT' in a bold, sans-serif font. The entire form is set against a solid gray background.

Obr. 3. Vstupný formulár

Preposielanie dát môžeme teda uskutočniť pomocou *Cookies*, alebo pomocou URL adresy. Citlivé údaje, ako môžu byť napríklad prístupové heslá, sa dajú posielat' pomocou bezpečnejšej metódy a to pomocou *SESSION*. [3]

## SESSION

Posielané dáta pomocou *SESSION* sú priamo ukladané na server, kde k nim programátor pristupuje pomocou špeciálne zadaných premenných.

Pred vytvorením *SESSION* sa musí zavolať funkcia *session\_start()*. Po zavolaní tejto funkcie, môžeme jednotlivé premenné vytvárať, ako aj pristupovať k už vytvoreným premenným. V nasledujúcich ukážkach je zobrazené ukladanie do novej premennej, ako aj načítanie hodnoty z vytvorenej premennej pomocou *SESSION*.

```
//uloženie hodnoty do novovytvorenej premennej  
session_start();  
$_SESSION['nova_session_premenna'] = $premenna;
```

```
//priradenie existujúcej hodnoty do novo vytvorenej  
premennej  
session_start();  
$nova_premenna = $_SESSION['ulozena_premenna'];
```

## 2.2 MySQL

Skratka SQL značí Structured Query Language, čo v preklade znamená štrukturovaný dotazovací jazyk. Jedná sa o najpoužívanejší jazyk, ktorý slúži k prístupu k relačným databázovým systémom. SQL sa používa na ukladanie dát do vytvorených databáz a k získavaniu dát z databáze. Používa sa v databázových systémoch ako sú Oracle, PostgreSQL alebo MySQL.

### 2.2.1 Priblíženie pojmu MySQL

*MySQL* je otvorený systém používaný pre správu relačných databáz spúšťaný vo viacerých vláknach, teda procesoch. Systém bol vyvinutý švédskou firmou *TcX*, keď potrebovali databázový systém, ktorý by bol veľmi rýchly a pružný. V dnešnej dobe je veľmi využívaný po celom svete. Používajú ho hlavne univerzity, poskytovatelia pripojenia k sieti Internet a rôzne iné neziskové organizácie. Veľkú obľubu získal pravdepodobne vďaka jeho cene, keďže väčšinou sa poskytuje zdarma. Databázový systém *MySQL* sa často zamieňa s pojmom *SQL*, čo je štrukturovaný jazyk pre dotazy, ktorý vyvinula spoločnosť *IBM*.

Relačná databáza je databáza, ktorá je zostavená z niekoľkých tabuliek, ktorých stĺpce sú viazané na stĺpce v iných tabuľkách. Týmto spôsobom dochádza k prepojeniu viacerých tabuliek, ktoré sú na sebe závislé. Vzťahy prepojenia – nazývané relácie, sú založené na kľúčových hodnotách uložených v jednotlivých stĺpcoch. Jednotlivé údaje sa ukladajú v riadkoch – záznamoch. Relačná databáza je veľmi intuitívna, čo znamená, že často napodobuje spôsob myslenia ľudí. Podobne ako ľudia zoskupuje podobné objekty do systematických celkov a zložité objekty rozkladá na jednoduchšie. Systém *MySQL* je podriadený využívaniu relačných modelov, čo celý systém ešte viac uľahčuje. [4]

### 2.2.2 Prečo používať MySQL

Najväčšou prednosťou *MySQL* v porovnaní s konkurenciou je rýchlosť a stabilita. Porovnávacia štúdia z roku 2002 vyhodnotila *MySQL* ako celkovo s najlepším výkonom. Hlavné výhody:

- Výkon
- Cena
- Stabilita
- Jednoduchosť použitia
- Funkčnosť

*MySQL* je bez pochyb veľmi rýchly. Na webových stránkach môžete nájsť rôzne porovnávacie testy, v ktorých sa porovná výkon *MySQL* s rôznymi databázovými systémami a *MySQL* vychádza z týchto testov víťazne.



Pre širokú škálu účelov je aplikácia úplne zdarma. Software je možné upravovať ako aj distribuovať. Ak však chcete aplikáciu redistribuovať ako časť komerčného produktu, musíte si zakúpiť licenciu.

Vývojári vždy kladli veľký dôraz na stabilitu systému. Všetky verzie musia prejsť radou testov. V poslednej rade je odrazom výkonnosti aj spokojnosť zákazníkov. V dnešnej dobe so systémom pracuje viac ako štyri milióny ľudí po celom svete.

Ďalšou kľúčovou vlastnosťou je jednoduchosť práce. Systém nepotrebuje žiadne zložité nastavenia a pre prácu postačia základné znalosti. [5]

### 2.2.3 Fungovanie webovej databáze

Databáza, ktorá je určená pre web, sa ovláda pomocou ovládacích skriptov. Komunikácia s databázou sa dá priblížiť v niekoľkých krokoch:

1. Prehliadač vyšle požiadavku HTTP na konkrétnu stránku. Ako príklad si predstavme databázu knižnice a my chceme vypísať všetky diela konkrétneho autora. Táto žiadosť sa vykoná pomocou nejakého formulára HTML. Výslednú stránku si nazvime ako *vysledok.php*.
2. Webový server prijme žiadosť o súbor *vysledok.php* a načíta súbor, ktorý potom predá do PHP k jeho spracovaniu.
3. PHP začne spracovávať skript, ktorý obsahuje príkaz pre pripojenie k databáze a príkaz k následnému vykonaniu niektorého dotazu, v tomto prípade k výpisu kníh daného autora.
4. MySQL server prijme dotaz ktorý spracuje a odošle výsledok späť do PHP
5. PHP dokončí spracovanie skriptu, ktorý väčšinou výsledok sformátuje do úhladnej stránky HTML.
6. Webový server odovzdá HTML späť prehliadaču a tým sa výsledok dotazu dostáva k samotnému užívateľovi. [5]

V kóde by tento sled príkazov mohol byť zapísaný nasledovne:

```
//pripojenie k databazy  
@ $db = mysql_pconnect("127.0.0.1","root","");
```

```
//výber vhodnej databazy
mysql_select_db("kniznica");

//výberový dotaz
$prikaz = `SELECT autor, dielo FROM knihy WHERE autor
LIKE $priezvisko`;
$vysledok = mysql_query($prikaz);

//zistenie počtu riadkov pre ďalší krok
$pocetriadkov = mysql_num_rows($vysledok);

//cyklus while na prechádzanie a následný výpis záznamov
while ($pole= mysql_fetch_assoc($vysledok)) {
    echo $pole["dielo"];}

//ukončenie pripojenia k databazy
mysql_close($db);
```

#### 2.2.4 Príkazy v MySQL

Ako je vo vyššie uvedenom príklade vidieť, jeden z výberových dotazov pomocou ktorého sa dajú z databázy vybrať nami zvolené údaje je príkaz *SELECT*. Okrem tohto dotazu existujú mnohé ďalšie, z ktorých do tabuľky uvediem tie najpoužívanejšie.

Tab. 5. Ukážka najpoužívanejších príkazov v MySQL

| dotaz       | Viaže sa s | Vysvetlenie            |
|-------------|------------|------------------------|
| ALTER TABLE | ADD, DROP  | Úprava tabuľky         |
| INSERT INTO | VALUES     | Vkladanie údajov       |
| SELECT      | FROM       | Výber z tabuľky        |
| UPDATE      | SET        | Zmena údajov v tabuľke |

Jednotlivé príkazy môžu v praxi vyzeráť nasledovne:

```
//vytorenie databázy
CREATE DATABASE knihnica;
USE knihnica;

//vytvorenie tabuľky so zadefinovaním stĺpcov
CREATE TABLE knihy(ID INT NOT NULL AUTO_INCREMENT ,
Dielo VARCHAR( 20 ) NOT NULL ,
Priezvisko VARCHAR( 22 ) NOT NULL ,
PRIMARY KEY ( ID)
);
```

```
//uloženie údajov do tabuľky
INSERT INTO kniznica.knihy (Dielo,Priezvisko) VALUES
('Titanic','Felinau');

//pridanie stĺpcu cena do existujúcej tabuľky
ALTER TABLE knihy ADD COLUMN Cena INT;

//výber diela a priezviska z tabuľky
SELECT Dielo, Priezvisko FROM knihy;

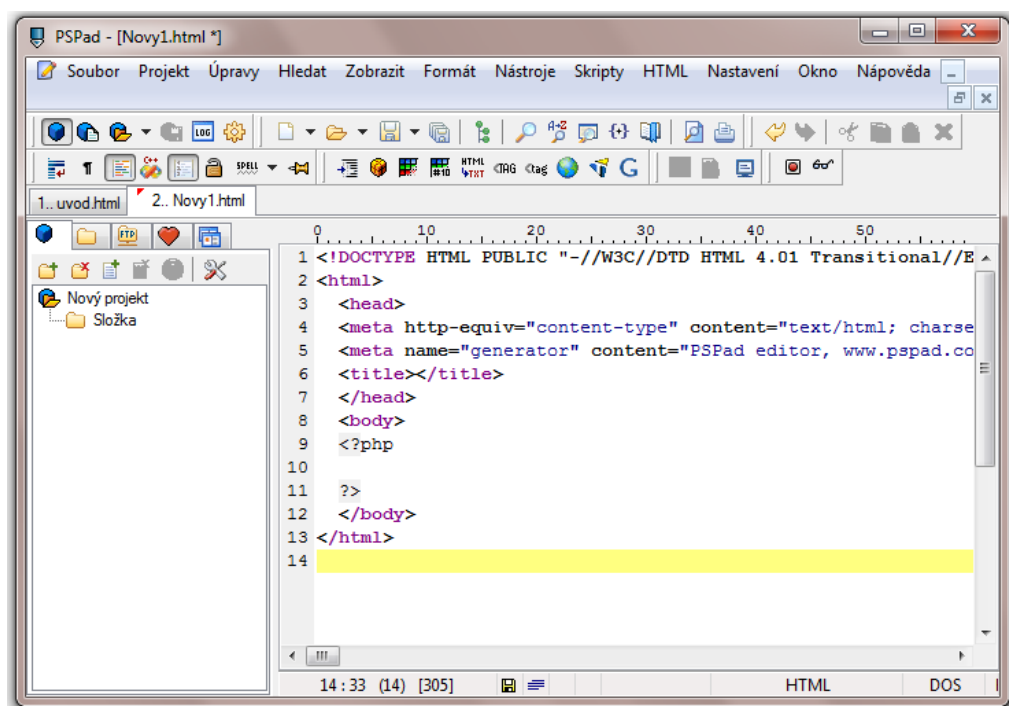
//upravenie ceny všetkých kníh o 20 korún
UPDATE knihy SET knihy.cena = knihy.cena - 20;
```

### 3 POUŽITÉ PROGRAMY

Na vytvorenie webovej aplikácie som použil niekoľko programov, ktoré mi pri práci uľahčili prácu.

#### 3.1 PSPad editor

PSPad patrí medzi voľne šíriteľné editory pre OS Windows. Týmto vzniká hneď prvá výhoda tohto editoru, že jeho zadováženie je úplne bezplatné. PSPad editor slúži pre písanie kódu pre nami zvolený druh súboru. Z ponuky si môžeme zvoliť aj mnou používaný formát \*.php ale aj \*.html. PSPad je určený pre všetkých, ktorí programujú webové stránky. Výhodou tohto editoru je to, že obsahuje veľké množstvo funkcií, ktoré nám ušetria veľa času. Vďaka farebnému zvýrazneniu pomôže užívateľovi aj pri hľadaní chyby respektíve ho na ňu upozorní. Pre formát \*.html poskytuje PSPad aj priamy náhľad webovej stránky vo zvolenom prehliadači. [6]



Obr. 4. Ukážka prostredia editora PSPad

## 3.2 Easy PHP

EasyPHP je samoinštalačný balíček ktorý obsahuje:

- skriptovací jazyk PHP
- http server Apache
- databázu MySQL
- webovú konfiguráciu PHPMyAdmin.

Konkurentmi tohto balíčka môžu byť napríklad PHPTriad alebo PHPHome.

Tento balíček umožňuje spustiť PHP skripty na vlastnom počítači, prostredníctvom *Localhost*. Pracuje na princípe spustenia rodičovského procesu, ktorý je nadriadený procesom, ktoré nazývame potomkovia a teda procesy, ktoré rodičovský proces riadi. Procesy potomkov spracujú jednotlivé požiadavky prehliadača, za ktoré sú zodpovedné.

### 3.2.1 Spustenie EasyPHP

Pri spustení *Apache* sa používa port 80. Môže sa stať, že tento port je obsadený niektorým z iných používaných programov. Budeme na to upozornení pri spustení *EasyPHP*. Tento program, ktorý port 80 obsadzuje, je potrebné ukončiť. Medzi tieto programy môže patriť napríklad *Skype* či *Matlab*. Ak neviete, ktorý program port 80 blokuje, alebo nemôžete z určitých dôvodov tento program vypnúť, existuje zmeniť hodnotu portu v súbore *httpd.conf*. Napriek tomu sa odporúča uvoľniť tento port pre jeho bezproblémové fungovanie.

Po úspešnom nainštalovaní balíka *EasyPHP* by sa v hlavnom paneli mala objaviť ikona písmeno čierne e, ktoré charakterizuje ikonu tohto programu a môžete overiť funkčnosť a správnosť inštalácie. V prvom rade je potrebné príslušnou ikonou spustiť chod všetkých služieb. Po spustení a po zadaní adresy *http://localhost* do prehliadača, by sa mala objaviť úvodná stránka. V prípade, že sa nezobrazí, existuje alternatíva v podobe zadania adresy *http://127.0.0.1* respektíve *http://127.0.0.1:8888*. Ak sa zobrazí úvodná stránka, pomocou pridania */home/* za adresu, dostanete sa do hlavnej stránky a obsahu *EasyPHP*. Do hlavnej stránky sa dostanete aj po kliknutí na ikonu pravým tlačidlom a zvolení možnosti *Administrace*. Po úspešnom nainštalovaní a spustení programu, môžete overovať a testovať svoje stránky priamo na servere vlastného počítača. [7]



*Obr. 5. Ikona EasyPHP*

### 3.2.2 Server Apache

Apache je veľmi výkonný a zároveň pre jeho jednoduchosť obľúbený webový server. Tento server je dostupný pre platformu Unix/Linux, ale taktiež aj pre systémy Windows.

NCSA server bol prvým WWW serverom, ktorý začali zdokonaľovať rôzni nadšenci. Títo do neho začali pridávať rôzne záplaty, aby zvýšili jeho výkon, funkčnosť a obmedzili bezpečnostné diery. Týmto spôsobom sa z pôvodného NCSA serveru stal nový, lepší, ktorému sa začalo hovoriť *Apache*.

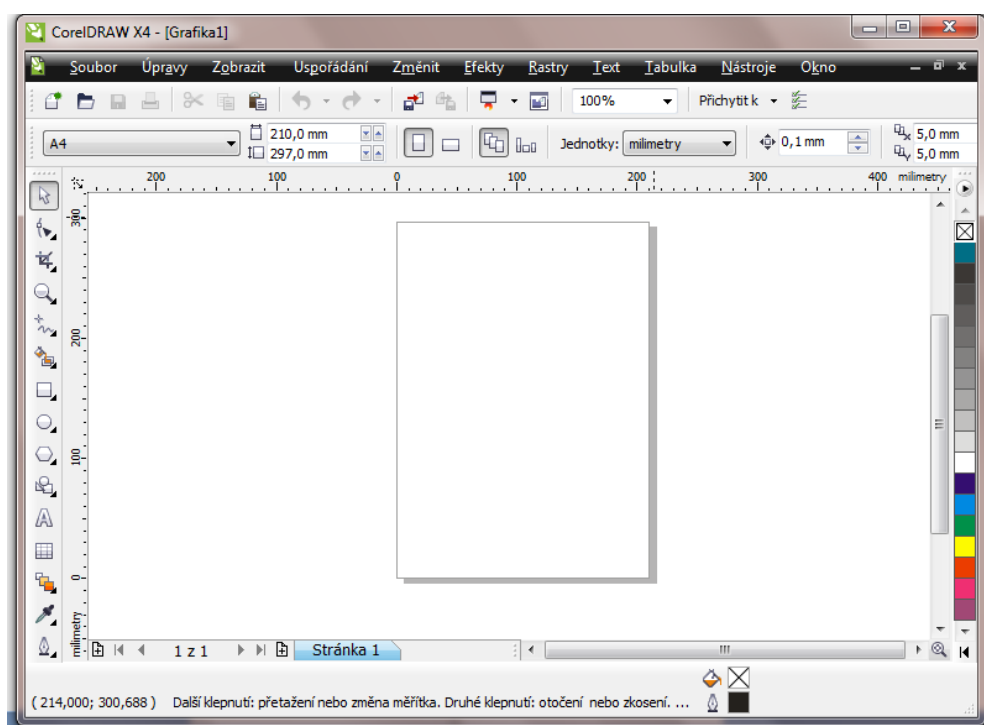
Apache je veľmi obľúbený server, o čom svedčí aj fakt, že jeho existenciu neobmedzil ani Microsoft so svojím serverom IIS. Apache je poskytovaný zdarma ako *open source*, a je prístupný zdrojový kód, ktorý umožňuje nadšencom podieľať sa na vývoji. V základnej konfigurácii ponúka niekoľko potrebných vlastností pre bezproblémové fungovanie www serveru. Poskytuje možnosť pridať niektorú z vlastností pomocou prídavných modulov, ktoré sú dostupné na internete.

Funkčnosť vytvorenej stránky je možné overiť aj priamo na lokálnom počítači, práve pomocou serveru *Apache*. [8]

### 3.3 CorelDRAW X4

*CorelDRAW* patří mezi velmi oblíbené a používané grafické programy. Program využívá vektorovou grafiku. Prispívá k efektívnější a bezproblémové integraci funkcí. Vďaka aplikáciám pre vektorové ilustrácie, digitálnemu spracovaniu obrazu, rozvrhnutiu stránky a pohyblivej grafiky je táto sada od *Corel* veľmi praktická. Inteligentné nástroje pomáhajú vytvoriť kreatívnejšie objekty a tvary, čím vznikajú pútavé návrhy. Pomocou grafického programu sa na webových aplikáciách dajú vytvoriť rôzne logá, hlavičky stránok alebo rôzne grafické doplnky pre web.

Prostredie programu *Corel* pôsobí veľmi prívetivo. V ľavej časti sú zobrazené hlavné nástroje na kreslenie a v pravej časti paleta farieb. Pomocou automatických tvarov, ako sú elipsa, krivky či rôzne maliarske techniky sa dajú navrhnuť logá, ktoré sa pomocou zložitejších úprav alebo pridávaním efektov môžu premeniť na profesionálne vyzerajúce návrhy.



Obr. 6. Ukážka prostredia CorelDRAW X4

## **II. PRAKTICKÁ ČÁST**



## 4 POSTUP PRI NÁVRHU APLIKÁCIE

Pred tým, ako začneme vytvárať webovú aplikáciu si musíme stanoviť základnú vec a to, čo všetko má obsahovať a „vedieť“. V mojom prípade sa mal užívateľ pri príchode na webovú stránku, dozvedieť základné informácie o firme, pozrieť si sídlo predajne priamo na mape, a to najhlavnejšie, mať možnosť overiť si dostupnosť a základné informácie o hľadanom výrobku. Keďže sa dostupnosť výrobkov, ako aj počet kusov mení s každým predajom a požičaním, musel som vytvoriť správčovskú časť, pomocou ktorej sa budú zaznamenávať všetky pohyby s výrobkami a tým sa automaticky meniť ich dostupnosť.

Začal som návrhom databáze, kde som si navrhol tabuľky, ich prepojenie a údaje ktoré sa v každej tabuľke budú nachádzať. Po navrhnutí databáze nasledovala samotná stránka, kde som si musel navrhnúť rozloženie užívateľskej časti, zvoliť farby stránky, grafiku, vyriešiť prihlasovanie pre správcu a nakoniec vytvoriť podobným postupom aj samotnú správčovskú časť.

Rozloženie stránky som prispôboval a stále obmeňoval podľa toho, čo všetko mala stránka obsahovať. Na záver som tieto návrhy konzultoval s majiteľom firmy, ktorý mi môj návrh schválil, a mohol som pokračovať v realizácii aplikácie.

### 4.1 Návrh databáze

Návrh databáze je nesmierne dôležitý pre čo najefektívnejšie využitie. Je potrebné si návrh tabuliek a ich naplnenie dôkladne premyslieť. Aby bola štruktúra databázy vhodná, nemali by jednotlivé tabuľky obsahovať veľké množstvo zbytočných alebo opakujúcich sa údajov. Taktiež by mala byť prehľadná s možnosťou ľahkého vyhľadávania požadujúcich záznamov. Každá tabuľka musí mať jednoznačný údaj, pomocou ktorého je daný záznam identifikovateľný.

Pri návrhu som sa odrážal od hlavnej tabuľky, ktorá má názov *Výrobok*. Do tejto tabuľky sa ukladajú hlavné informácie o výrobku, ako sú napríklad *názov*, *značka*, *cena* či *spôsob predaja*. Identifikátorom tejto tabuľky je identifikačné číslo výrobku, označené ako *ID\_V*, ktoré je unikátne v rámci celej tabuľky. Je nastavené na „auto increment“, čo znamená, že v každom novom zázname sa jeho hodnota automaticky zvýši o jednotku.

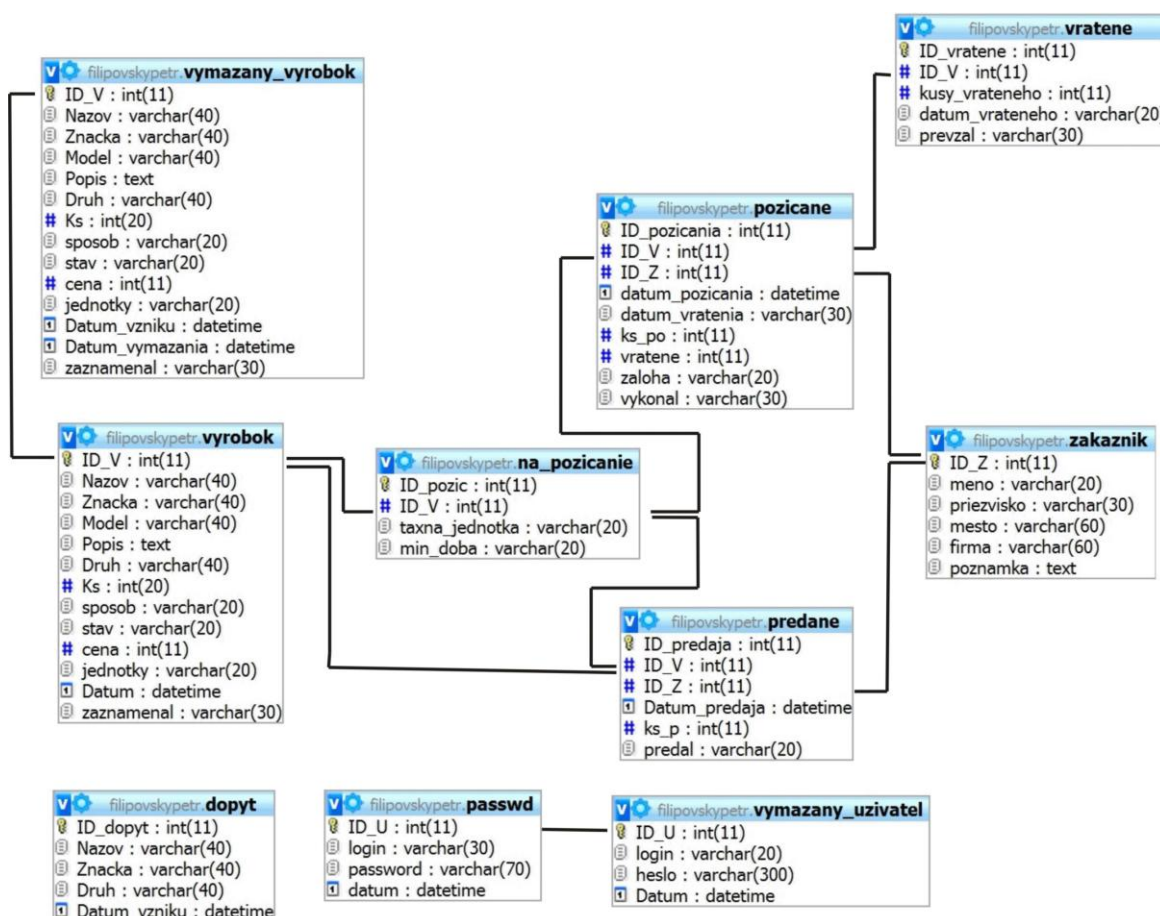
Ukážka vytvorenia tabuľky *Výrobok* spoločne so zadefinovaním celej štruktúry tabuľky:

```
CREATE TABLE IF NOT EXISTS `vyrobok` (  
  `ID_V` int(11) NOT NULL AUTO_INCREMENT,  
  `Nazov` varchar(40) COLLATE utf8_czech_ci DEFAULT NULL,  
  `Znacka` varchar(40) COLLATE utf8_czech_ci DEFAULT NULL,  
  `Model` varchar(40) COLLATE utf8_czech_ci DEFAULT NULL,  
  `Popis` text COLLATE utf8_czech_ci,  
  `Druh` varchar(40) COLLATE utf8_czech_ci DEFAULT NULL,  
  `Ks` int(20) DEFAULT NULL,  
  `sposob` varchar(20) COLLATE utf8_czech_ci DEFAULT NULL,  
  `stav` varchar(20) COLLATE utf8_czech_ci DEFAULT NULL,  
  `cena` int(11) DEFAULT NULL,  
  `jednotky` varchar(20) COLLATE utf8_czech_ci DEFAULT NULL,  
  `Datum` datetime DEFAULT NULL,  
  `zaznamenal` varchar(30) COLLATE utf8_czech_ci DEFAULT  
  NULL,  
  PRIMARY KEY (`ID_V`)  
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci  
AUTO_INCREMENT=142 ;
```

Tabuľka *Výrobok* je pomocou *ID\_V* prepojená s tabuľkou *VYMAZANÝ\_VÝROBOK*, do ktorej sa pri vymazaní niektorého z výrobkov prekopírujú všetky informácie o tomto výrobku, a z tabuľky *Výrobok* sa kvôli nedostupnosti vymažú. Ukážka príkazu:

```
'INSERT INTO vymazany_vyrobok (ID_V, Nazov, Znacka, Model,  
Popis, Druh, Ks, sposob, stav, cena, jednotky, Datum_vzniku,  
Datum_vymazania, zaznamenal) SELECT ID_V, Nazov, Znacka,  
Model, Popis, Druh, Ks, sposob, stav, cena, jednotky, Datum,  
NOW(), zaznamenal FROM vyrobok WHERE  
ID_V='.$_GET["mazanie_poloziiek"];
```

Celkový návrh štruktúry databáze vrátane prepojenia tabuliek je zrejmé so schémy (Obr.7.):



Obr. 7. Štruktúra databáze a prepojenie tabuliek

Tabuľka *Dopyt* je samostatná tabuľka, do ktorej sa ukladajú údaje, ktoré užívateľ zadá do vyhľadávania. V administračnej časti si potom prevádzkovateľ obchodu môže pozrieť, ktorý výrobok bol najviac vyhľadávaný alebo poprípade, ktorý výrobok v prípade veľkého záujmu by mal zadovážiť, ak ho ešte nemá. Každá tabuľka obsahuje unikátne ID pre danú tabuľku.

Tabuľka *Na\_pozicanie* obsahuje doplňujúce informácie k výrobkom, ktoré sú určené na zapožičanie. Táto tabuľka je prepojená s tabuľkou *Pozicane*, do ktorej sa ukladajú bližšie informácie o spôsobe a čase požičania výrobku na to určeného. S tým súvisí tabuľka *Vratene*, do ktorej sa ukladajú informácie o vrátení výrobku a zároveň, v prípade vrátenia výrobku, sa odoberajú alebo upravujú údaje v tabuľke *Pozicane*.

V tabuľke *Zakaznik* sú uložené informácie o zákazníkoch, ako je meno, priezvisko či mesto, z ktorého pochádza. Pomocou identifikačného čísla zákazníka, *ID\_Z*, sú tieto informácie o konkrétnom zákazníkovi pridelované jednotlivým predaným či požičaným výrobkom.

## 4.2 Rozloženie stránky

Jednou z vecí, ktoré by sme určite nemali pri návrhu podceňovať je rozloženie stránky. Aj keď sa to niekomu môže zdať zbytočné alebo nie až tak dôležité, netreba túto časť práce zanedbať. Od prvotného rozloženia sa odvíjajú ďalšie umiestnenia ovládania stránky či zobrazenie informácií. Určite nikto z návštevníkov nechce po stránke hľadať jednotlivé tlačidlá či text, a preto by malo mať rozloženie stránky svoju logiku, aby bola pre návštevníkov prehľadná, ľahko ovládateľná a neodradila ich po pár minútach.

Ukážka časti kódu z CSS štýlov pre nastavenie hlavného informačného panelu. V ukážke je vidieť nastavenie jeho veľkosti, font písma ako aj farba pozadia.

```
//názov triedy pre ktorú nastavenie platí
.stred {

//výška bloku
height: 400px;

//šírka bloku
width:610px;

//font písma pre celú triedu s následnými alternatívami
font-family: "Times New Roman", Times, serif;

//veľkosť písma
font-size: 19px;

//farba pozadia zadaná kódom farby
background-color: #DCDCDC;

//zarovnanie textu
text-align: center;

//posun celého bloku triedy
float: left;
}
```

#### 4.2.1 Uživatelská část

Pre budúcu stránku som sa rozhodol použiť jednoduché a prehľadné umiestnenie jednotlivých sekcií. Pomocou kaskádových CSS štýlov som jednotlivé rámce porozmiestňoval tak, aby vo finálnej podobe tvorili celok. Celá stránka je zarovnaná na stred s tým, že jednotlivé rámce majú nastavené „plávanie“ float: left. Stránku tvoria:

- Vrchná časť tzv. hlavička stránky:
  - Logo firmy
  - Názov firmy a stručná charakteristika
- Ľavá časť
  - Vertikálne menu, ktoré slúži k navigácii po stránke
- Stredná časť
  - Horizontálne menu, kde sa nachádza rozdelenie výrobkov a stručný popis ku každému z nich
  - Hlavný informačný panel
- Spodná časť tzv. pätička
  - Informácie o dátume poslednej aktualizácie stránky
  - Meno autora a správcu
  - Možnosť pre prihlásenie
  - Reklamný banner



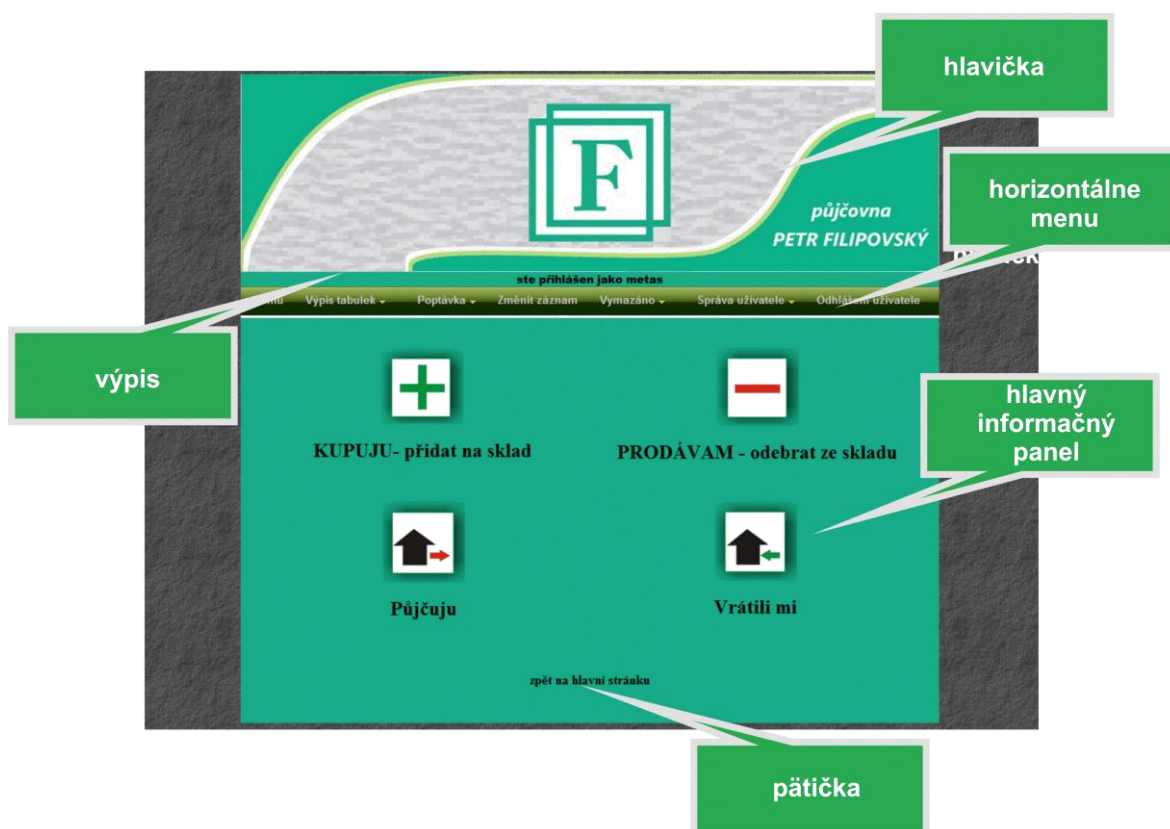
Obr. 8. Rozloženie užívateľskej časti

#### 4.2.2 Administračná časť

Administračná časť je na rozdiel od užívateľskej pomerne menej členitá. Skladá sa v podstate len z piatich častí:

- Vrchná časť tzv. hlavička stránky:
  - Logo firmy
  - Názov firmy a stručná charakteristika
- Výpis
  - Prihlasovacie meno práve prihláseného užívateľa
- Horizontálne menu
  - Rozdelené do 3 častí (zľava) – práca a informácie o výrobkoch, zmeny záznamov, práca s užívateľmi
- Stredná časť
  - Hlavný informačný panel a priestor pre zadávanie údajov

- Spodná časť tzv. pätička
  - Odkaz na návrat do užívateľskej časti stránky



Obr. 9. Rozloženie administračnej časti

### 4.3 Grafika a funkčnosť stránky

Vzhľad stránky môže už po prvom kliknutí odradiť alebo naopak prilákať návštevníkov. Pri návrhu grafiky stránky som sa opäť snažil zvoliť jednoduchosť. Nešiel som do príliš prehnaných a zložitých efektov, ale na druhej strane som nechcel ani statickú stránku s pár nadpismi. Pomocou JavaScript-ových jQuery som „oživil“ ľavé, vertikálne menu, ktoré sa po prechode myšou aktivuje, rozroluje a užívateľ má možnosť vybrať si časť, do ktorej chce v rámci stránky ísť. Celú stránku som viedol v štyroch základných farbách a to zelená, svetlo zelená, svetlo šedá a na doplnenie biela.

Pri voľbe farieb som vychádzal z farieb loga, ktorým som prispôbil aj hlavičku stránky.

### 4.3.1 Hlavička stránky

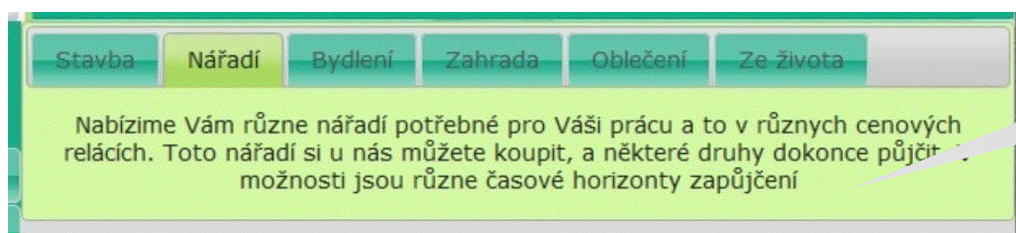
Hlavička slúži ako titulok celej stránky. Informuje o názve a prináša základné prvky či charakteristiky obsahu stránky. V mojom prípade to je logo, názov a stručná charakteristika firmy. Hlavičku som navrhol v programe na vektorovú grafiku CorelDraw.

Z dvoch kriviek som vytvoril oblasť, ktorá prechádza celým priestorom hlavičky. Začína v pravom hornom rohu a končí v ľavom dolnom rohu, kde nadväzuje na sekciu menu.

Uprostred tejto oblasti som pomocou textúr vytvoril dojem tehlovej steny s miernym rozostrením, čo by malo charakterizovať firmu zameranú na stavebné náradie. Farbu som volil v odtieňoch šedej, keďže som túto farbu plánoval použiť aj v hlavnom informačnom paneli. Do stredu som umiestnil logo firmy, ktoré pozostáva z dvoch prekrývajúcich sa štvorcov a v strede veľké písmeno F.

### 4.3.2 Menu a stredný priestor

Zvyšným sekciám stránky som pomocou kaskádových CSS štýlov nastavil farbu pozadia do odtieňov farieb vyplývajúcich z hlavičky. Horizontálne aj vertikálne menu majú pridelené rovnaké farby.



Obr. 10. Ukážka farieb menu

Ukážka nastavenia kódov farieb pre jQuery v oboch menu.

Pozadie v lište – šedá farba, kód #cccccc

```
ui-widget-header { border: 1px solid #aaaaaa; background:
#cccccc url(images/ui-bg_highlight-soft_75_cccccc_1x100.png)
```



Pozadie obsahu aktívnej záložky – svetlo zelená

```
.ui-widget-content { border: 1px solid #aaaaaa; background:
#d5f8a3 url(images/ui-bg_flat_75_d5f8a3_40x100.png)
```

Pozadie aktívnej záložky – tmavo zelená

```
ui-state-active { border: 1px solid #aaaaaa; background:
#abdb5e url(images/ui-bg_highlight-hard_65_abdb5e_1x100.png)
```

Neaktívna záložka – tmavo zelená

```
ui-state-active { border: 1px solid #aaaaaa; background:
#18ac88 url(images/ui-bg_glass_75_18ac88_1x400.png)
```

## 5 UŽÍVATEĽSKÁ ČASŤ

Celá aplikácia je rozdelená na dve časti, užívateľskú a administračnú časť. K užívateľskej časti má prístup bežný užívateľ, ktorý stránku navštívi. V tejto časti sa môže užívateľ dozvedieť základné informácie o firme, lokalitu firmy, prejsť priamo na zobrazenie predajne na mape, pozrieť si, čo všetko firma ponúka a to hlavné, vyhľadať a overiť dostupnosť hľadaného výrobku.

Pre celú užívateľskú časť sú charakteristické tri základné priestory. Vertikálne a horizontálne menu, ktoré slúžia na pohybovanie sa po stránke a stručným popisom ponuky a stredný panel, kde sa zobrazujú zvolené informácie. Obe spomínané menu sú odlišené od stredového panelu rovnakou, svetlo zelenou farbou.

### 5.1 Jednotlivé položky vertikálneho menu

Vertikálne menu pozostáva z piatich položiek, ktoré odkazujú na jednotlivé podstránky a to konkrétne:

- O nás
- Nabízíme
- Kontakt
- Kde nás najdete
- Vyhledávání

Prvá položka *O nás*, je zároveň aj počiatočnou podstránkou, na ktorej sa nachádza krátka prezentácia zostrihu fotiek a stručná charakteristika celej firmy. Prezentácia je tvorená pomocou jQuery.

Položka *Nabízíme*, ku ktorej sa neskôr dostaneme aj podrobnejšie, ponúka približný popis ponúkaných výrobkov v podobe foto-prezentácie s grafickým zobrazením najpredávanejších výrobkov. Obrázky v tejto prezentácii som vytvoril v grafickom programe CorelDraw X4.

Pre prípad získania akýchkoľvek kontaktných údajov je vytvorená položka *Kontakt*, na ktorej sa zobrazia základné kontaktné údaje ako sú napríklad, IČ, adresa, telefónne číslo na firmu a iné.

*Kde nás najdete* prináša možnosť kliknúť na obrázok mapy, ktorý vás priamo prepojí na mapu s GPS súradnicami. Je tu taktiež možnosť pustiť si krátke video s presným zobrazením sídla firmy. Toto video bolo nahraté pomocou nahrávania plochy programom CamStudio. Mapa, ktorá bola nahrávaná, bola zobrazená v programe Google Earth, ktorý ponúka možnosť vyznačenia konkrétneho bodu pomocou ulice alebo GPS súradníc. Toto video som spracoval v MovieMaker II a video vo finálnej podobe som vložil do tejto stránky.

K položke *Vyhľadávání* bude ešte venovaný zvlášť priestor, pretože je to položka asi najdôležitejšia a slúži k tomu, aby sa na podstránke *Vyhľadávání* dala overovať dostupnosť výrobkov.



Obr. 11. Vertikálne menu

### 5.1.1 Podstránka Nabízíme

Zákazník by mal mať možnosť získať prehľad o výrobkoch, ktoré firma ponúka. Na podstránke *Nabízíme* má túto možnosť ponúknutú hneď niekoľko krát. V horizontálnom menu sa zobrazí stručný výpis výrobkov rozdelených do jednotlivých sekcií, ktoré firma ponúka.

V tejto časti sa zároveň nachádza odkaz „pro podrobnější výpis“, pri ktorom sa vypíšu všetky výrobky pre danú sekciu, ktoré sa v databáze nachádzajú. Výrobky sú triedené podľa názvu. Znamená to, že výrobok s názvom Rukavice model Dowe a výrobok

s názvom Rukavice model Finch sa v podrobnom výpise vypíše podľa názvu, teda Rukavice.

Tento podrobný výpis slúži pre úplné zistenie zákazníka o ponuke výrobkov, ktoré má možnosť v tejto predajni zakúpiť. Jednotlivé položky výpisu slúžia zároveň aj ako odkazy pre samotné vyhľadávanie a tým zobrazenie detailnejších informácií o výrobku, ako sú napríklad cena či dostupnosť.

### 5.1.2 Podstránka Vyhledávání

Pre overenie stavu, ceny, počtu kusov, či iných informácií o výrobkoch, ktorými predajňa disponuje, slúži podstránka *Vyhledávání*. Užívateľovi sa zobrazí vyhľadávací formulár, ktorý tvoria tri vstupné políčka. Políčka názov, spôsob a druh hľadaného výrobku. Políčko spôsob a druh sú tvorené pomocou funkcie *select*, kde sú preddefinované hodnoty, ktoré je možné zadať. Uľahčuje to spôsob rozhodovania a vyhnutie sa zadávaniu nezmyselných výrazov. Tieto políčka majú prednastavenú hodnotu, ktorá je zobrazená pred samotným zadaním, pomocou vlastností políčka, vlastnosť *VALUE*.

Možnosť zadávania kľúčových pojmov je skutočne široká. Užívateľ môže zadať názov výrobku alebo len jeho časť, a zvyšné políčka môže nechať nevyplnené. Toto zabezpečuje funkcia `nazov LIKE '$nazov%'`, kde % zastupuje akýkoľvek znak.

To znamená, že v prípade, že do políčka názov zadáte „ruk“, budú vypísané všetky výrobky, ktorých názov začína na „ruk“, ako napríklad rukavice, rukovet' a podobne.

Ďalšou z možností je výpis všetkých výrobkov z danej sekcie, čo nastáva pri nevyplnenom názve a zvolenom druhu výrobku. Možná je samozrejme aj kombinácia týchto možností, a teda vyplnenie názvu, ako aj druhu výrobku.

Samozrejme, čím viac údajov je zadaných, tým menej záznamov sa vypíše a tým menej času potrvá, kým bude nájdený finálny výrobok. V prípade nezadania ani názvu ani druhu, bude zákazník upozornený, že nezadal žiadnu voľbu a tým pádom nebude žiaden záznam z databázy vypísaný.

Zadejte informace o výrobku:

| NÁZEV                | ZPŮSOB      | DRUH    |
|----------------------|-------------|---------|
| <input type="text"/> | na prodej ▼ | Vyber ▼ |

*Obr. 12. Vyhledávací formulár*

## 6 ADMINISTRAČNÁ ČASŤ

Po úspešnom prihlásení sa tento administrátor (správca) dostáva do administratívnej časti stránky. Táto časť slúži k „riadeniu“ skladu ako aj získavaniu informácií a prehľade o jednotlivých výrobkoch a ich transakciách. Aby sa bežný užívateľ v užívateľskej časti dozvedel, že daný výrobok bol požičaný, alebo že už bol vrátený a je opäť na sklade a k dispozícii, musí sa evidovať každý proces s týmto výrobkom. K tomuto evidovaniu slúži práve administratívna časť.

### 6.1 Prihlásenie sa do administratívnej časti

V užívateľskej časti sa v pravom dolnom rohu nachádza priestor pre prihlásenie sa. Toto prihlásenie je z bezpečnostných dôvodov umožnené iba pracovníkom firmy a povereným osobám, preto v užívateľskej časti nenájdete žiadnu registráciu, iba samotné prihlásenie.

Po prechode na podstránku *Login*, budete automaticky pomocou JavaScript-ovej hlášky *alert* upozornení, že prihlásenie je povolené iba zaregistrovaným užívateľom. Po zadaní mena a hesla sa skontroluje v databáze tabuľka, v ktorej sú tieto prihlasovacie údaje uchované z hľadiska bezpečnosti v zakódovanej forme. Ak sa vyhodnotí, že sa meno a heslo zhodujú s výsledkami z tabuľky, budete prihlásení do administratívnej časti, pričom na každej stránke bude v hornej časti vypísané vaše prihlasovacie meno, aby ste vedeli, aké užívateľské meno bude systém evidovať pri jednotlivých virtuálnych úpravách na sklade.

V prípade, že sa prihlásený užívateľ rozhodne prejsť do užívateľskej časti, a potom sa opätovne vrátiť do administratívnej, nemusí sa znova prihlasovať. Bude len na začiatku upozornený, pod akým menom je prihlásený, aby sa predišlo nedorozumeniam.

Spôsob prihlasovania je realizovaný pomocou *session*, ktorá sa vytvorí pri úspešnom prihlásení. Tieto *session* sa pri prechode na akúkoľvek podstránku kontrolujú dvoma spôsobmi. Kontroluje sa, či existuje *session*, v ktorej je pridelené prihlasovacie meno a druhou kontrolou je zhoda medzi obsahom druhej *session* a preddefinovanou hodnotou reťazca obsahujúceho kombináciu písmen a čísiel. V prípade, že oba kontrolné testy sú úspešné, je užívateľ úspešne prihlásený a má možnosť pohybovať sa po podstránkach.

Prihlásený užívateľ sa tu stretáva s jediným horizontálnym menu, pomocou ktorého môže vypisovať či upravovať zvolené údaje. V strednom paneli má k dispozícii štvoricu tlačidiel, ktoré sú určené na pridávanie nových výrobkov pri kúpe, odoberanie pri predaji, požičiavanie alebo naopak vrátenie výrobku od zákazníka. V spodnej časti sa nachádza odkaz, ktorý slúži na presmerovanie do užívateľskej časti stránky.

## 6.2 Stredový panel, ovládacie tlačidlá

Do stredového, hlavného panela je umiestnená štvorica preddefinovaných, najčastejšie používaných tlačidiel. Tieto tlačidlá boli navrhnuté pomocou grafického programu CorelDraw X4. Zobrazujú jednoduchý obrázok charakterizujúci funkciu tlačidla. Celému tlačidlu je pomocou Gaussovského rozostrenia pridaný efekt odtieňu na zelenej ploche stránky. Sú určené k evidencii pohybu výrobku. Konkrétne to sú tlačidlá:

- Kupuju
- Prodávám
- Půjčuju
- Vrátili mi

Po kliknutí na tlačidlo *Kupuju*, sa zobrazí formulár s položkami určenými pre zadávanie údajov o novo kúpenom výrobku. Pri nevyplnení políčka kusy alebo jeho vyplnení na hodnotu 0 nebude vykonaná žiadna operácia a budete upozornení o nezadaní počtu kusov.

Po vyplnení a uložení pomocou tlačidla *Ulož*, sa súčasne prechádza celá tabuľka Výrobok, a porovnávajú sa zadané hodnoty s hodnotami uloženými. Ak nastane zhoda, tento novo pridaný výrobok nie je uložený ako nový výrobok, iba sa upraví počet kusov predchádzajúcemu výrobku. Ak sa však v tabuľke takýto výrobok nenachádza, je pridaný ako nový výrobok.

V prípade, že je pri výrobku nastavený spôsob na hodnotu *Na půjčení*, po kliknutí na tlačidlo *Ulož* bude užívateľ automaticky prepojený na dodatočný formulár, v ktorom bude zadávať dodatočné údaje, ktoré by výrobok určený na požičanie mal obsahovať. Tieto údaje sa ukladajú do tabuľky *Na požičanie*. Sú to hodnoty ako taxná jednotka, minimálna doba požičania a podobne. Po ukončení celého procesu sa na obrazovke vypíšu údaje, ktoré boli zadané.

Ukážka kódu pre automatické presmerovanie v prípade zvolenia spôsobu na požičanie:

```
if ($sposob LIKE „na půjčení“) {  
    header('location: zadan_taxnu.php');  
    die;  
}
```

Proces vykonávaný pri kliknutí na *Prodávam* a *Půjčuju* funguje na podobnom princípe. Užívateľ opäť, ako pri *Kupuju*, vyplní vstupný formulár, ktorý slúži na vyhľadávanie. Môže zadávať kombináciu jednotlivých údajov, ale tak isto môže vyplniť len jedno políčko napríklad názov, alebo len časť názvu. Rovnako ako v prípade *Kupuju*, budete upozornený pri nevyplnení políčka určeného pre zadávanie počtu kusov a nevykoná sa žiadna operácia. Pri vyhľadávaní sa užívateľovi vypíšu všetky výrobky, ktoré zadanému dotazu vyhovujú. Tabuľka pripomína tú, ktorá sa vypisuje pri vyhľadávaní v užívateľskej časti s tým rozdielom, že v poslednom stĺpci je pri každom výrobku hodnota *Půjčit*, respektíve *Prodat*. Po kliknutí na toto pole, sa automaticky všetky zadávané hodnoty vzťahujú na výrobok, ku ktorému patrila táto hodnota. Inými slovami, táto zmena sa prevedie na výrobok, ktorý sa nachádza v tom istom riadku, do ktorého sme klikli na jednu z možností *Půjčit*, respektíve *Prodat*.

Funkcia tlačidla *Vrátili mi* funguje na rovnakom princípe, len je obohatená o kontrolné operácie. Zadané informácie o vrátenom výrobku sú pridané do tabuľky *Vrátené* s aktuálnym dátumom a do tabuľky výrobok sa pripočíta počet kusov vrátených výrobkov, z toho dôvodu, že už sú k dispozícii. Kontrolná operácia spočíva v kontrole počtu kusov zvoleného výrobku z vyhľadaných možností. V prípade, že sa hodnota vrátených kusov rovná počtu kusov požičaných, budú tieto údaje o požičanom výrobku v tabuľke *Požičané* vymazané. V opačnom prípade, sa počet kusov v tabuľke *Požičané* len upraví o hodnotu vrátených kusov.





Obr. 13. Středový panel, administrační část

### 6.3 Hlavné menu

Hlavné menu je funkčne rozdelené do 3 častí (zľava) – informácie o výrobkoch, zmeny záznamov, užívateľ. Po prechode myšou po jednotlivých položkách sa automaticky zobrazí ukrytá časť menu, ktorá k danej položke patrí. V ukrytej časti menu sa nachádza bližšie určenie samotnej položky, ktorá reprezentuje celú časť v hlavnom paneli menu.

Prvá časť, *Informácie o výrobkoch* pracuje s výpisom rôznych tabuliek z databázy, k získavaniu informácií o výrobkoch a o procesoch s nimi.

V časti *Zmeny záznamov* môže prihlásený užívateľ upravovať jednotlivé údaje o konkrétnom výrobku a obnovovať vymazaných užívateľov.

Posledná, tretia časť je určená pre možnosti užívateľa. Sú to funkcie ako registrácia nového užívateľa, zmena vlastného hesla, výpis všetkých užívateľov, ktorí majú možnosť sa do administračnej časti prihlásiť a samotné odhlásenie prihláseného užívateľa.

### 6.3.1 Výpis tabuliek

Prvá časť menu je zameraná na výpis tabuliek, informácie o výrobkoch, ako aj o fázach, v ktorých sa výrobky nachádzajú. Po prechode sa zobrazia ďalšie možnosti výpisu.

- *Všetchny položky* – výpis všetkých výrobkov, ktoré sa v databáze nachádzajú. K výpisu všetkých položiek patri dodatočná voľba kritérií, ktorými si vyberieme, v akom poradí sa majú položky vypísať. Výrobky môžu byť zoradené podľa názvu, dátumu alebo podľa jednotlivých druhov.
- *Prodané* – pod touto záložkou sa ukrýva výpis predaných výrobkov s časovým určením predaja, s údajmi o zákazníkovi, ktorý daný výrobok kúpil a prihlasovacie meno predávajúceho.
- *Půjčené* – výpis výrobkov, ktoré boli požičané, s časovým údajom, doba na akú bol výrobok požičaný a či bola a v akej výške uhradená zálohová platba.
- *Vrácené* – výpis výrobkov, ktoré boli vrátené a počet kusov, ktorý bol vrátený.
- *Zákazníci* – výpis údajov všetkých zákazníkov, ktorí si niekedy požičali alebo zakúpili niektorý z výrobkov.
- *Poptávka* – výpis výrazov, ktoré bežný užívateľ zadal do vyhľadávania v užívateľskej časti.

Ukážka časti kódu pre kontrolu zadávania počtu kusov na podstránke pre ukladanie:

```
$ks = $_POST['ks'];  
if($ks == 0){  
    $_SESSION["nulakupene"] = "1";  
    header('location: kupil_som.php');  
} else {unset($_SESSION['nulakupene']);}
```

Ukážka časti kódu pre kontrolu zadávania počtu kusov na podstránke pre zadávanie:

```
<h3 style="color: #CD3333">  
<?php  
if (ISSET ($_SESSION["nulakupene"]))  
{echo "nezadali ste počet kusů!";}  
>  
</h3>
```

### 6.3.2 Zmeniť záznam, vymazané položky

Zmena záznamu slúži na zmenu akéhokoľvek údaju ku ktorémukoľvek výrobku. Prihlásený užívateľ pomocou vyhľadávacieho formuláru vyberie výrobok, ktorý chce upraviť. Po vybratí výrobku sa zobrazí nový formulár, ktorý má vstupné polia vyplnené pôvodnými hodnotami konkrétneho výrobku. Pri kliknutí na tlačidlo *uložiť* sa skontroluje, ktoré polia boli zmenené a iba zmenené hodnoty sa v pôvodnom výrobku upravujú pomocou príkazu *update*.

V položke *Vymazáno* má prihlásený užívateľ možnosť výpisu vymazaných výrobkov ako aj vymazaných užívateľov s možnosťou obnovy užívateľského mena a hesla. Vymazanie užívateľa je možné využiť pre krátkodobé zablokovanie prístupu, a to tým spôsobom, že pri výpise užívateľov stačí vymazať užívateľské meno. Keď sa prihlásený užívateľ rozhodne prístup vymazanému užívateľovi znova umožniť, prejde do možnosti *vymazaný užívateľ*, kde ho jediným kliknutím dá obnoviť a vymazaný užívateľ sa môže opäť prihlasovať. Je to založené na princípe vymazania údajov z tabuľky, v ktorej sa kontroluje pri prihlasovaní a vloženie do záložnej tabuľky. Pri obnove užívateľa sú tieto údaje zo záložnej tabuľky späť vložené do hlavnej prihlasovacej tabuľky.

### 6.3.3 Užívateľ

Tretia časť pozičného rozdelenia slúži na správu užívateľov. Je umiestnená v pravej časti hlavného menu. V tejto časti má prihlásený užívateľ možnosť (právo) zaregistrovať nového užívateľa, zmeniť svoje heslo, vypísať si zoznam všetkých užívateľov, ktorým je prihlásenie umožnené a samotné odhlásenie práve prihláseného užívateľa.

Pod položkou *Správa užívateľa* v hlavnom menu sú ukryté tri položky, a to registrácia, zmena hesla a výpis užívateľov.

Pri registrácii musí prihlásený užívateľ vyplniť tri polia vstupného formulára, ako je prihlasovacie meno, heslo a zopakovanie zadaného hesla. Pri pokuse o registráciu sa kladie dôraz na dve bezpečnostné podmienky. Prvá podmienka je založená na prehľadávaní tabuľky s uloženými prihlasovacími údajmi a porovnávaním, či už takéto prihlasovacie meno existuje. V prípade že sa prihlasovacie meno zhoduje, znamená to, že už je použité a užívateľ bude upovedomený obsadenosťou tohto mena a výzvou o zmenu. V opačnom prípade dochádza k druhej kontrolnej podmienke, ktorou je zhoda oboch zadaných hesiel.

Ak zadané hodnoty splňajú obe podmienky, údaje o novom užívateľovi budú zapísané do databázy a prístup mu od tejto chvíle bude umožnený.

Ukážka časti kódu kde sa kontroluje originalita užívateľského mena:

```
$novemeno = $_POST['novemeno'];
$prikaz = 'SELECT login FROM passwd';
$vsledok = mysql_query($prikaz);
$pocetriadkov = mysql_num_rows($vsledok);

for ($i = 0; $i < $pocetriadkov; $i++)
{
    $riadok = mysql_fetch_array($vsledok);

    if ($riadok["login"] == $novemeno ) {
        echo "taky již existuje, ";
        echo ' <a href="registracia.php">ZKÚSTE NOVÉ JMÉNO</a>';
        exit;}
    }
```

Zmena hesla umožňuje prihlásenému užívateľovi zmeniť jeho aktuálne heslo. Užívateľ pri zmene hesla musí zadať svoje prihlasovacie meno, staré heslo, nové heslo a zopakovať nové heslo. Po odoslaní požiadavky na zmenu sa prehľadá tabuľka užívateľov a porovná sa staré heslo s heslami uloženými v tabuľke, pri zhode sa porovnáva prihlasovacie meno s menom v tabuľke a pri opätovnej zhode sa sleduje zhodnosť novo zadných hesiel. V prípade akejkoľvek nezhody sa požiadavka na zmenu hesla nevykoná.

Odhlásenie užívateľa slúži na odhlásenie práve prihláseného užívateľa. Keďže je celé prihlásenie založené na kontrole *session*, pri odhlásení užívateľa dochádza k zrušeniu práve tejto *session* pomocou dodatočného príkazu `unset($_SESSION['...'])`;

Po tomto kroku dochádza k odhláseniu užívateľa, ktorý sa od tejto chvíle do opätovného prihlásenia vracia do užívateľskej časti a opúšťa administračnú časť.



Obr. 14. Menu, administračná časť

## 7 3D MODEL PREDAJNE

Súčasťou bakalárskej práce je aj vymodelovanie 3D modelu predajne, keďže sa tým vo voľnom čase zaoberám. Neskôr som sa však rozhodol pre vymodelovanie celého areálu, v ktorom sa predajňa nachádza. Model pomáha k čo najživšej predstave reálneho vzhľadu predajne. V programe používam pôvodnú knižnicu programu, ktorú som doplnil o vlastné objekty. Do programu je v prostredí 2D umiestnená reálna fotka, na ktorej je zobrazený celý areál pri pohľade zhora. Vďaka tejto fotke som mal reálnu predstavu o rozmeroch jednotlivých budov, ako aj o samotnom vzhľade celého areálu.

Jednotlivé budovy, stromy, koľajnice, časti nábytku ako aj cesty vznikali postupným modelovaním a zoskupovaním jednotlivých objektov. K dispozícii sú nástroje, ako je napríklad stena, strecha, okná a dvere či podlaha. Jednotlivým nástrojom sa dajú priradovať vlastnosti a nastavovať materiál či rozmery objektu. Okrem objektov, ktoré som sám vymodeloval, obsahuje program aj knižnicu, v ktorej sa nachádzajú niektoré z používaných objektov. Sú to napríklad stôl, stoličky či kancelárske zariadenia. Jednotlivým objektom sa dá priradiť farba, materiál či rozmer jednotlivých častí objektu. Niektoré súčasti nábytku som vytvoril pomocou zoskupenia viacerých z preddefinovaných objektov. Takýmto spôsobom vznikol napríklad kancelársky stôl so zásuvkami, ktorý bol vytvorený pomocou zoskupenia a vhodného rozmiestnenia klasického stola a zásuvkovej kuchynskej skrinky. Všetky budovy v modeli areálu spĺňajú reálne rozmery a sú vymodelované iba pre vizuálnu podobu areálu. Jediná budova, ktorá zachováva nielen reálne rozmery, ale taktiež použitý materiál stavby, farbu ako aj zariadenie budovy, je samotná predajňa *Filipovský*. Po celom areály je možná virtuálna prechádzka, ktorej rýchlosť, pohľad ako aj samotný pohyb ovládame pomocou klávesnice podľa našich potrieb.



*Obr. 15. Ukážka fotografií 3D modelu*

## ZÁVĚR

Hlavným cieľom bakalárskej práce bolo vytvoriť webovú aplikáciu, ktorá by uľahčila zisťovanie informácií a dostupnosti o výrobkoch na sklade firmy *Filipovský*. Pri týchto stránkach bol kladený dôraz na jednoduchosť, funkčnosť a prehľadnosť. Základnou požiadavkou bolo, aby sa zákazník na stránkach dozvedel základné informácie o firme, ako je adresa, telefónne číslo, ale hlavne aby mal možnosť si pomocou vyhľadávania overiť dostupnosť hľadaného výrobku na sklade. Každý hľadaný pojem sa ukladá pre správcu do tabuľky. Keďže k zmene stavu dochádza pri každom pohybe výrobku ako je predaj, kúpenie na sklad či samotné požičanie a vrátenie, bolo potrebné vytvoriť správcovskú časť, v ktorej správca zaznamenáva každý jeden pohyb s daným výrobkom. Následne sa každý tento pohyb zaznamenáva do databázy, odkiaľ si tieto údaje môže správca nechať vypísať. Pomocou jednoduchého a prehľadného menu si správca môže sledovať a kontrolovať, ktorý výrobok komu predal, požičal, kedy a kto mu ho vrátil, tak isto ako aj zisťovať, po ktorých výrobkoch je najvyšší dopyt. Správca má taktiež možnosť kontrolovať všetky výrobky na sklade ako im aj meniť jednotlivé hodnoty.

V praktickej časti popisujem jednotlivé kroky, ktoré charakterizujú postup tvorby celej webovej aplikácie od grafického návrhu, tvorby loga až po samotné naprogramovanie a prepojenie s databázou.

Stránka obsahuje validný HTML kód ako aj CSS štýly, čo je možné si overiť na stránkach rôznych validátorov napríklad [www.validator.w3.org](http://www.validator.w3.org). Stránky sú kompatibilné vo všetkých najpoužívanejších prehliadačoch (Mozilla Firefox, Google Chrome, Internet Explorer 8, Opera).

Výsledkom bakalárskej práce je moderná, funkčná stránka, ktorá uľahčí prácu a prístup k výrobkom pre zákazníka, poskytne dokonalú správu skladu ako aj informácie o každom pohybe s výrobkom pre správcu a majiteľa firmy.

Vytvorená stránka bola umiestnená na server, ktorý podporuje PHP ako aj prácu s databázou MySQL. Stránky sú dostupné na [www.filipovskypetr.php5.cz](http://www.filipovskypetr.php5.cz).

## ZÁVĚR V ANGLIČTINĚ

The main objective of my work was create a Web application, that would facilitate inquiries and availability of products in Filipovský stock companies. At these sites, emphasis was placed on simplicity, functionality and clarity. The basic requirement was to get the customer known about basic information on the pages relating to the company such as address, phone number, but mainly to have the opportunity to check the availability of searched product in stock. Each search term is imposed for the manager to the table. Since the state change occurs in each movement of the product as a sale, buy-to-stock or the actual rent and return, it was necessary to develop an administration part in which the administrator records each one motion with a given product. Consequently, any such movement is recorded into the database where the administrator can list the file after the record was made. With a simple and transparent menu administrator can monitor and control the product which he sold, loaned, when and who returned it to him, as well as to identify the products whose demand is the highest. The administrator also has the ability to check all products in stock and then change each of their values.

The practical part describes the steps that characterize the process of creation throughout the web application such as graphic design, logo creation, programming and connection to the database.

This page contains valid HTML code and valid CSS styles, which you can check on various sites such as validators [www.validator.w3.org](http://www.validator.w3.org). Pages are compatible in the most used browsers (Mozilla Firefox, Google Chrome, Internet Explorer 8, Opera).

Result of thesis is a modern and functional website that facilitates work and access to product for the customer, providing the perfect stock management as well as information about each movement with the product destined for manager and for company owner.

Created page was placed on a server that supports PHP as well as work with MySQL database. Pages are available on [www.filipovskypetr.php5.cz](http://www.filipovskypetr.php5.cz).



## SEZNAM POUŽITÉ LITERATURY

- [1] WELLING, L., THOMSON, L. PHP a MySQL rozvoj webových aplikací. Praha : SoftPress s.r.o., 2003. 910 s. ISBN: 80-86497-60-7
- [2] WILLIAMS, H., LANE, D. PHP a MySQL Vytváříme webové databázové aplikace. Praha : Computer Press, 2005. 530 s. ISBN: 80-7226-760-4
- [3] JANOVSKEÝ, Dušan. *Jakpsatweb.cz* [online]. 2011 [cit. 2011-05-2]. Formuláře. Dostupné z WWW: <<http://www.jakpsatweb.cz/formulare-html.html>>
- [4] MASLAKOWSKI, Mark. *Naučte se MySQL za 21 dní*. Praha : Computer Press, 2001. 478 s. ISBN 80-7226-448-6
- [5] WELLING, Luke; THOMSON, Laura. *MySQL Průvodce základy databázového systému*. Vyd. 1. Brno : CP Books, a.s., 2005. 255 s. ISBN 80-251-0671-3
- [6] FIALA, Ján. *Www.pspad.com* [online]. 2009 [cit. 2009-07-12]. Textový editor PSPad. Dostupné z WWW: <http://www.pspad.com/cz/>
- [7] *Http://www.easyphp.org* [online]. 2010 [cit. 2010-05-21]. Easy PHP. Dostupné z WWW: <http://www.easyphp.org/>
- [8] JANOVSKEÝ, Dušan. *Jakpsatweb.cz* [online]. 2011 [cit. 2011-05-2]. PHP -- Jak začít. Dostupné z WWW: <<http://www.jakpsatweb.cz/php/jak-zacit.html#apache>>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

|      |   |
|------|---|
| PHP  | Hypertext Preprocessor, pôvodne Personal Home Page                        |
| HTML | HyperText Markup Language   |
| PDF  | Portable Document Format – dokumentový formát                             |
| URL  | Uniform Resource Locator – slúži k presnému umiestneniu zdroja informácií |
| SQL  | Structured Query Language – dotazovací jazyk                              |
| OS   | Operačný systém   |
| WWW  | World Wide Web  |
| ID   | Identifikačné číslo   |
| CSS  | Cascading Style Sheets – kaskádové štýly                                  |
| GPS  | Global Positioning System – globálny družicový polohový systém            |
| 3D   | Trojdimenziálny, Trojrozmerný   |
| 2D   | Dvojdimenziálny, Dvojrozmerný   |

**SEZNAM OBRÁZKŮ**

|  |    |
|--|----|
| Obr. 1 - Komunikácia serveru.....                      | 12 |
| Obr. 2 - Výstup na obrazovku.....                      | 14 |
| Obr. 3 - Vstupný formulár.....                         | 23 |
| Obr. 4 - Ukážka prostredia editora PSPad .....         | 28 |
| Obr. 5 - Ikona EasyPHP .....                           | 30 |
| Obr. 6 - Ukážka prostredia CorelDRAW X4.....           | 31 |
| Obr. 7 - Štruktúra databáze a prepojenie tabuliek..... | 35 |
| Obr. 8 - Rozloženie užívateľskej časti .....           | 38 |
| Obr. 9 - Rozloženie administračnej časti .....         | 39 |
| Obr. 10 - Ukážka farieb menu.....                      | 40 |
| Obr. 11 - Vertikálne menu .....                        | 43 |
| Obr. 12 - Vyhľadávací formulár .....                   | 45 |
| Obr. 13 - Stredový panel, administračná časť .....     | 49 |
| Obr. 14 - Menu, administračná časť .....               | 52 |
| Obr. 15 - Ukážka fotografií 3D modelu.....             | 54 |

**SEZNAM TABULEK**

|  |    |
|--|----|
| Tab. 1. Příklady a názvy aritmetických operátorů .....                           | 15 |
| Tab. 2. Příklady a názvy jednotlivých kombinovaných operátorů .....              | 16 |
| Tab. 3. Příklady a názvy jednotlivých porovnávacích operátorů .....              | 17 |
| Tab. 4. Příklady, názvy a hodnoty výsledku jednotlivých logických operátorů..... | 17 |
| Tab. 5. Ukážka nejčastějších příkazů v MySQL .....                               | 26 |