Střední průmyslová škola Edvarda Beneše
a Obchodní akademie Břeclav,
příspěvková organizace

Elektrotechnika – Informační technologie

Maturitní práce

Vytvoření systému pro správu studentských projektů s využitím jazyka PHP a databázového systému MySQL

Vedoucí práce:
Ing. Vilém Závodný

Vypracoval:
Filip Krolop

Úvod

Maturitní práce není jen zpracování zadaného témat, ale každá část má svůj význam, ve kterém student projeví znalost učiva z různých předmětů za celé studium. Vše je zpracováno v původní technické zprávě.

Cíl práce

Vytvořit uživatelsky přívětivý systém pro správu studentských projektů. Systém je administrován přes internet.

Osnova řešení – sub části maturitní práce

1. Analýza problému – záměr návrhu
2. Teoretické zpracování systému
3. Vytvoření prostředí – programování systému
4. Ověření fungování a odladění systému
5. Zpracování dokumentace

Technická zpráva

Je zpracována jedna technická zpráva maturitní práce dle pokynů. Rozsah zprávy min. 20 stran textu (podrobné pokyny jsou zadány v elektronické podobě v systému LMS). Součástí zpracování je podklad do sborníku (cca 2 stránky), publicita projektu – Internetová stránka, plakát.

Termín odevzdání práce: 31. března 2016

Poděkování

Tímto bych chtěl poděkovat panu Ing. Vilému Závodnému za jeho čas, ochotu při konzultacích a za vědomosti získané v jeho hodinách.

Prohlášení

Prohlašuji, že tato práce je mým autorským dílem, kterou jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

V Břeclavi dne 31.3.2016 Filip Krolop

Abstract

- **Krolop Filip:** Vytvoření systému pro správu studentských projektů s využitím jazyka PHP a databázového systému MySQL
- **Zpracování dokumentace:** Břeclav 2016

Projekt názorně ukazuje využití PHP a databázový systém SQL.

Obsah

Úvod a cíl práce

Má maturitní práce se zabývá problematikou PHP a databázového systému (dále jen "DBS") SQL.

Úkolem je vytvoření interaktivního systému v PHP s podporou DBS SQL pro pro správu studentských projektů a také rozšíření znalostí v oblasti PHP a DBS SQL.

1. Základní struktura webových stránek

Kapitola osvětluje základní pojmy a jazyky, se kterými se setkáme při vytváření webových stránek s názorným příkladem jejich použití v mé práci.

1.1 Co je to HTML

HTML je jazyk, který se používá k vytváření základní obsahové kostry webových stránek. Dříve jazyk HTML sloužil i k formátování vzhledu (v současnosti se k tomu kvůli zachování přístupnosti webu používají kaskádové styly, které umožňují vytvářet vzhled jako druhou, na obsahu nezávislou vrstvu).

Název HTML je zkratkou od HyperText Markup Language – textový značkovací jazyk. Slovo HyperText zde vyjadřuje možnost vzájemně propojovat texty na základě odkazů, Markup označuje schopnost jazyka HTML dávat významy jednotlivým blokům textu s pomocí speciálních značek nazývaných tagy a elementy (např. vypsat část textu tučně nebo ji třeba určit jako nadpis).

Jazyk HTML vznikl v roce 1990 ve Švýcarsku a postupně se vyvíjel v závislosti na nejpoužívanějších prohlížečích až k současné verzi HTML 4.01, u které byl vývoj ukončen, neboť na ni navazuje modernější jazyk XHTML.

1.2 Co je to CSS

Kaskádové styly, známé také pod zkratkou CSS (z anglického Cascading Style Sheets) jsou moderním jazykem umožňujícím účinné formátování stránek psaných v jazycích HTML, XHTML či XML. Slovo kaskádové, jež mají CSS v názvu, značí jejich nejcharakterističtější vlastnost – jednotlivá pravidla kaskádových stylů se mohou vzájemně překrývat, což zvyšuje jejich efektivnost.

Jsou-li kaskádové styly správně používány, umožňují naprosté oddělení vzhledu dokumentu od jeho obsahu (tzv. beztabulkové layouty). Toto oddělení obou vrstev (prezentační a strukturální) zvyšuje přístupnost webu a právě v něm spočívá hlavní rozdíl proti formátování s pomocí atributů, jež se používalo dříve.

Další výhody kaskádových stylů proti používání samotného HTML:

- větší možnosti formátování
- snazší správa větších prezentací (CSS šablony)
- rychlejší načítání stránky (kaskádové styly se snadno kešují)
- menší zatížení serveru
- společně s JavaScriptem lze s CSS vytvářet DHTML

Kaskádové styly se však netýkají jen obrazovky klasických prohlížečů, CSS se používají i k formátování tiskové verze, lze jimi ovlivnit zobrazení stránky na mobilních zařízeních nebo třeba audio výstup slepeckých čteček.

Vznik kaskádových stylů se datuje k roku 1997, jejich vytvoření iniciovala organizace W3C.

1.3 Co je to PHP

PHP je programovací jazyk, který pracuje na straně serveru. Původní význam zkratky PHP byl Personal Home Page. Vzniklo v roce 1996, od té doby prošlo velkými změnami a nyní tato zkratka znamená PHP: Hypertext Preprocessor. PHP umí krom jiného ukládat, měnit a mazat data. Vše se odehrává na webovém serveru, kde jsou uloženy zdrojové kódy webových stránek. PHP skript se nejprve provede na serveru a poté se odešle prohlížeči pouze výsledek (například skript nejprve spočítá, kolik je 300/30 a prohlížeči odešle jen číslo 10). Ve zdrojovém kódu tedy najdeme jen číslo "10" (to je rozdíl oproti JavaScriptu, který počítá přímo v prohlížeči). Zdrojový kód PHP na rozdíl od JavaScriptu a HTML v prohlížeči nezobrazíme.

Pomocí PHP je možné vytvořit např. diskuzní fórum, knihu návštěv, počítadlo, anketu, graf a mnoho dalšího. Navíc lze stránky propojit s databázemi, např. MySQL. Na webových stránkách se obvykle opakují některé části, hlavička s odkazy, menu, patička. S PHP lze snadno vytvořit šablonu pro web, do které se budou vkládat soubory s menu, patičkou atd. Můžeme tedy mít menu jen jednou zapsané a do dalších stránek ho pouze kopírovat.

Webová stránka s prvky PHP má nejčastěji koncovku **.php**. Lze se však setkat i s dalšími koncovkami, např. .phtml, php3, php4, php5. Některé hostingy dle koncovky

určovaly, pod jakou verzí PHP skript spustit (aktuální je 7). To je však velice výjimečné a dnes v naprosté většině případů postačuje koncovka .php.

1.4 Co je to JavaScript

JavaScript je objektově orientovaný programovací jazyk, využívaný při tvorbě webových stránek. Na rozdíl od serverových programovacích jazyků (například PHP) sloužících ke generování kódu samotné stránky, JavaScript běží na straně klienta, tedy v prohlížeči až po stažení do vašeho počítače.

JavaScript se používá především pro vytváření interaktivních webových stránek. Příkladem použití mohou být nejrůznější kontroly správného vyplnění formulářů, obrázky měnící se po přejetí myší, rozbalovací menu atd. JavaScript se také často používá k měření statistik návštěvnosti.

Společně s jazykem HTML (informační kostrou stránky) a CSS (formátováním vzhledu stránky) je JavaScript součástí DHTML, souboru technik a postupů zaměřených na zlepšení uživatelského rozhraní a zvýšení prožitku z používání stránek. K tomu JavaScript využívá tzv. DOM, rozhraní umožňující přistupovat k jednotlivým prvkům stránky.

JavaScript vyvinula společnost Netscape v roce 1995 a v roce 1998 byl standardizován organizací ISO. JavaScript se poté stal základem pro další programovací jazyky, např. ActionScript, používaný v technologiích Flash a Flash Lite.

1.5 Ukázka kódu

S využitím všech výše zmíněných skriptovacích jazyků byl vytvořen i můj maturitní projekt. Pro demonstraci syntaxe bych si v této části kapitoly dovolil ukázat zdrojový kód hlavní stránky (souboru **index.php**) mého systému pro správu projektů.

```
<?php
session_start();

$uzivatel_prihlasen = $_SESSION['prihlasen'];

?>
<!doctype html>
<html>
<head>
     <meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
     <meta http-equiv="Content-language" content="cs" />
     <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
      <meta name="viewport" content="width=device-width, initial-
scale=1">
      <link rel="stylesheet" type="text/css" href="css/obecne.css">
      <link rel="stylesheet" type="text/css" href="css/menu.css">

<?php
      include "components/hlavicka-css-js.php";

      if ($_GET["page"]!="novyprojekt" &&
$_GET["page"]!="upravitprojekt") {
            echo '<script src="http://code.jquery.com/jquery-
latest.min.js" type="text/javascript"></script>';
      }
?>
      <title>Databáze maturitních projektů</title>
</head>
<body>
<?php
      include "components/menu.php";
      include "components/definice-sekci.php";
?>
</body>
</html>
```

Vysvětlivky k jednotlivým částem kódu:

- PHP instrukce „**Session“** slouží k uložení hodnot do proměnných, které se je nutno zachovat i při přechodech mezi jednotlivými stránkami, typicky přesměrování. Hodí se zejména při tvorbě přihlašovacího mechanismu, kdy je třeba, aby si při pohybu uživatele v prostředí administrace prohlížeč pamatoval, že je uživatel přihlášen.

  Instrukce „**Include**“ do kódu hlavní stránky zahrne celý obsah jiného dokumentu. Rozdělení kódu do více souborů výrazně zpřehledňuje orientaci ve vytvořené aplikaci. V mém případě se do hlavičky stránky přidají odkazy na externí styly a skripty, využité pro design webu, do těla hlavní stránky se připojí menu a obsah konkrétní stránky.

- **!doctype** - neboli prvek pro definici typu dokumentu (DTD) určuje verzi HTML či XHTML v níž je dokument psán

- **<html>** uvozuje a zakončuje celou stránku, head vymezuje hlavičku dokumentu, která obsahuje základní informace o stránce a připojené skripty a styly.

- **<title>** - titulek stránky, její název. Zobrazuje se úplně nahoře v horním pruhu prohlížeče.

- **<meta>** - obsahuje důležité informace o dokumentu (použitá znaková sada na stránce, klíčová slova sloužící internetovým vyhledávačům apod.)
- **<body>** - tělo stránky, do něj se zapisuje veškerý obsah HTML stránky

Vkládání obsahů do stránky (soubor **/components/definice-sekci.php**):

```
<div>
<?php
      if (!$_GET["page"]) {
       include "sekce/vypis-projektu.php";
      }else{
            Switch($_GET["page"]){
                  case "login";
                        include "components/login.php";
                        break;
                  case "projekty";
                        include "sekce/vypis-projektu.php";
                        break;
                  case "logout";
                        include "components/logout.php";
                        break;
                  case "uzivatele";
                        include "sekce/uzivatele.php";
                        break;
                  case "nastaveni";
                        include "sekce/nastaveni.php";
                        break;
                  case "profil";
                        include "sekce/profil.php";
                        break;
                  case "mujprojekt";
                        include "sekce/mujprojekt.php";
                        break;
                  case "sprava";
                        include "sekce/sprava-projektu.php";
                        break;
                  case "upravitprojekt";
                        include "components/upravit-projekt.php";
                        break;
                  case "novyprojekt";
                        include "components/novy-projekt.php";
                        break;
            }
      }
?>
</div>
```

V závislosti na tom, ve které sekci se bude uživatel nacházet, se do těla hlavní stránky vkládají obsahy konkrétních stránek. Tento problém jsem vyřešil využitím řídící struktury **Switch**, která v závislosti na splněné podmínce (uživatel je na jedné

z definovaných stránek definovaných v návěští **case**) vnoří do těla hlavní stránky požadovaný obsah.

1.6 Tabulka výpisu projektů

Ukázka zpracování tabulky pro výpis projektů (soubor **/sekce/vypis-projektu.php**):

```
<?php
...
echo "
<table class='tabulkacss' cellspacing='0'>
<thead><tr>
    <th><a href='?page=projekty&razeni=id-09'
class='razeni'>&#8595;</a> ID <a href='?page=projekty&razeni=id-90'
class='razeni'>&#8593;</a></th>
    <th><a href='?page=projekty&razeni=nazevprojektu-az'
class='razeni'>&#8595;</a> Název projektu <a
href='?page=projekty&razeni=nazevprojektu-za'
class='razeni'>&#8593;</a></th>
    <th><a href='?page=projekty&razeni=kategorie-az'
class='razeni'>&#8595;</a> Kategorie <a
href='?page=projekty&razeni=kategorie-za'
class='razeni'>&#8593;</a></th>
    <th><a href='?page=projekty&razeni=autor-az'
class='razeni'>&#8595;</a> Autor <a
href='?page=projekty&razeni=autor-za'
class='razeni'>&#8593;</a></th>
    <th><a href='?page=projekty&razeni=trida-az'
class='razeni'>&#8595;</a> Třída <a
href='?page=projekty&razeni=trida-za'
class='razeni'>&#8593;</a></th>
<th><a href='?page=projekty&razeni=rok-09'
class='razeni'>&#8595;</a> Rok <a href='?page=projekty&razeni=rok-
90' class='razeni'>&#8593;</a></th>
<th><a href='?page=projekty&razeni=vedouci-az'
class='razeni'>&#8595;</a> Vedoucí práce <a
href='?page=projekty&razeni=vedouci-za'
class='razeni'>&#8593;</a></th>
<th>PDF</th>
</tr></thead><tbody>";

if (mysql_num_rows($result) > 0) {
     while($row = mysql_fetch_assoc($result)) {
          echo "<tr><td>" . $row["id"]. "</td>
          <td>" . $row["nazev_projektu"]. "</td>
          <td>" . $row["kategorie"]. "</td>
          <td>" . $row["autor"]. "</td>
          <td>" . $row["trida"]. "</td>
          <td>" . $row["rok"]. "</td>
          <td>" . $row["vedouci_prace"]. "</td>
          <td><a href='?page=projekty&pdf=".$row["id"]."'><img
src='./img/pdf.png' alt='Stáhnout PDF'></a></td></tr>";
     }
```

```
}

else {
      echo "<tr><td>" . $row["id"]. "</td>
      <td>" . $row["nazev_projektu"]. "</td>
      <td>" . $row["kategorie"]. "</td>
      <td>" . $row["autor"]. "</td>
      <td>" . $row["trida"]. "</td>
      <td>" . $row["rok"]. "</td>
      <td>" . $row["vedouci_prace"]. "</td>
      <td> </td></tr>";
}

echo "</tbody></table>";
...
?>
```

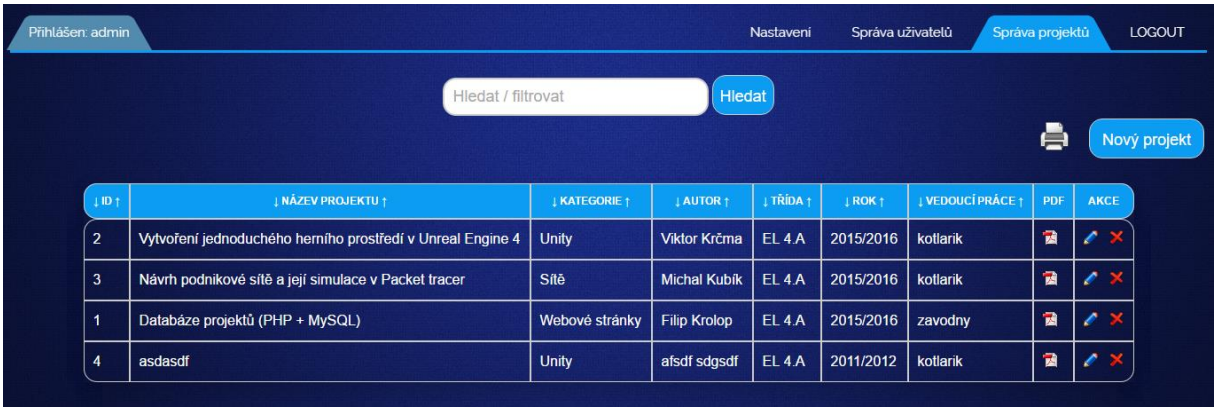
Jak si můžete z ukázky všimnout, jedná se o klasickou tabulku doplněnou o hlavičkové buňky **<th>**, která je stylizovaná pomocí CSS. Šipky ↑↓ sloužící pro řazení výpisu jsou v HTML zastoupeny Unicode kódy **(↑** pro šipku nahoru, **↓** pro šipku dolů). Způsob provedení řazení je proveden rovněž pomocí řídící direktivy Switch, kdy je důležitý předávaný GET parametr **razeni** v URL.

```
Switch($_GET["razeni"]){
      case "";
                  $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc";
                  $result = mysql_query($sql);
            break;
      case "id-09";
                  $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY id";
                  $result = mysql_query($sql);
            break;
      case "id-90";
                  $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY id DESC";
                  $result = mysql_query($sql);
            break;
      case "nazevprojektu-az";
                  $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY
nazev_projektu";
                  $result = mysql_query($sql);
            break;
      case "nazevprojektu-za";
                  $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY
nazev_projektu DESC";
                  $result = mysql_query($sql);
            break;
      case "kategorie-az";
                  $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY kategorie";
```

```
                $result = mysql_query($sql);
            break;
      case "kategorie-za";
                $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY kategorie
DESC";
                $result = mysql_query($sql);
            break;
      case "autor-az";
                $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY autor";
                $result = mysql_query($sql);
            break;
      case "autor-za";
                $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY autor DESC";
                $result = mysql_query($sql);
            break;
      case "trida-az";
                $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY trida";
                $result = mysql_query($sql);
            break;
      case "trida-za";
                $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY trida DESC";
                $result = mysql_query($sql);
            break;
      case "rok-09";
                $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY rok";
                $result = mysql_query($sql);
            break;
      case "rok-90";
                $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY rok DESC";
                $result = mysql_query($sql);
            break;
      case "vedouci-az";
                $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY
vedouci_prace";
                $result = mysql_query($sql);
            break;
      case "vedouci-za";
                $sql = "SELECT id, nazev_projektu, kategorie, autor,
trida, rok, vedouci_prace FROM projekty $hlinc ORDER BY
vedouci_prace DESC";
                $result = mysql_query($sql);
            break;
}
```

Ukázka kódu pro řazení projektů ze skriptu /components/vypis-razeni.php

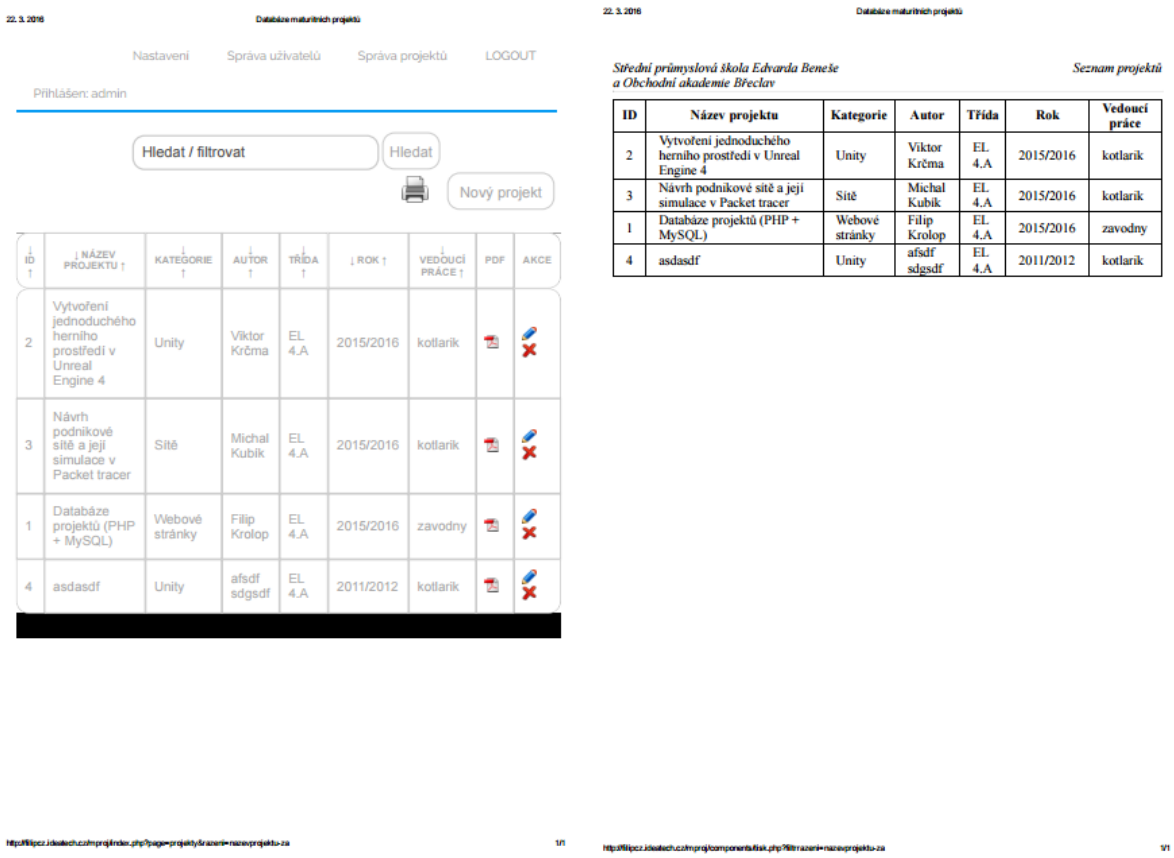
Proměnná **$hlinc** je nastavena pouze v případě, že bylo použito vyhledávací pole a výsledky vyhledávání jsou filtrovány. Pokud se tak stalo, doplní se na místo **$hlinc** do SQL dotazu podmínka *„WHERE nazev_projektu LIKE '%$hledat%' OR kategorie LIKE '%$hledat%' OR autor LIKE '%$hledat%' OR trida LIKE '%$hledat%' OR rok LIKE '%$hledat%' OR vedouci_prace LIKE '%$hledat%'“* a vyfiltrované výsledky mohou být dále řazeny. Najdeme-li např. v adresním řádku prohlížeče **razeni=nazevprojektu-za**, znamená to, že všechny vložené projekty budou seřazeny sestupně podle názvu. Situaci vystihuje následující obrázek:



Obr.1: Stránka výpisu projektů

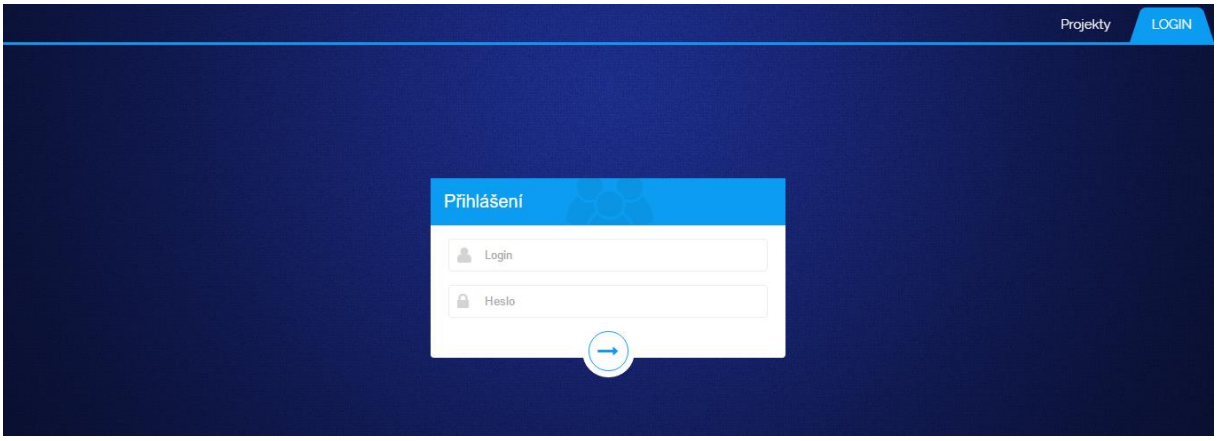
1.7 Okno pro tisk

Vypsané výsledky je možné vytisknout kliknutím na ikonu tiskárny. Protože by nebylo příliš vhodné tisknout takto stylizovanou tabulku včetně menu v záhlaví stránky, po kliknutí na ikonu se otevře nové okno prohlížeče s obyčejnou neformátovanou tabulkou vhodnou právě pro tisk.



Obr.2: Porovnání optimalizované tabulky pro tisk (vpravo) s původní tabulkou projektů (vlevo).

1.8 Přihlášení do administrace



Obr.3: Stránka pro přihlášení do administrace

V základu se lze do systému přihlásit dvěma způsoby: pevně vytvořeným účtem administrátora (admin) nebo loginem do školní sítě. Při odeslání vyplněného formuláře dojde ke kontrole pomocí protokolu IMAP, zda je účet ve školní síti (při prvotním přihlášení takového uživatele dojde k zápisu jeho účtu do databáze

s nastavenou hodnotou IMAP=1, při každém dalším přihlášení se ověří krom platnosti přihlašovacích údajů také hodnota IMAP). Pokud účet není součástí školní sítě, dojde k ověření platnosti zadaných přihlašovacích informací pro lokální databázový účet. Nesedí-li přihlašovací jméno či heslo nebo je účet blokován, uživatel je o této skutečnosti informován chybovou hláškou. Skript pro přihlašování vypadá následovně:

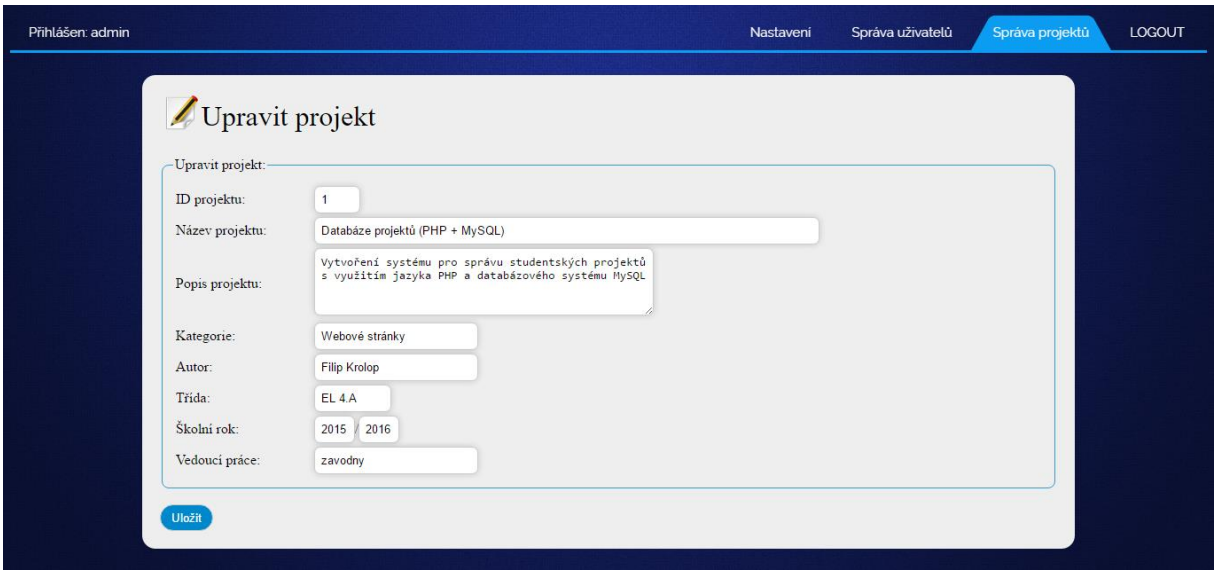
```
<?php

if($_POST['login']!="" && $_POST['heslo']!="") {
      $login = $_POST['login'];
      $heslo = $_POST['heslo'];

// Ověření, zda je login ve školní síti
      $server = "{intra.spsbv.cz:143/notls}";
      @$mbox = imap_open($server, $_POST["login"], $_POST["heslo"]);

// Je ve školní síti a platí přihl. údaje.
      if ($mbox) {
            include "login-imap.php";
      }

// Není ve školní síti, ověření lokálního loginu
      else{
            include "login-mysql.php";
      }

      if($login == $login_def && $heslo == $heslo_def &&
!isset($uzivatelblokovan)){
            session_start();
            $_SESSION['prihlasen'] = $login;

            echo
"<script>window.location='index.php?page=projekty'</script>";
      }

      else if(isset($uzivatelblokovan)){
            echo '
                  <div class="msg">
                        Přihlášení se nezdařilo. <br />
                        Uživatel blokován.
                  </div>
            ';
      }

      else {
            echo '
                  <div class="msg">
                        Špatné jméno nebo heslo. <br />
                        Přihlaste se prosím znovu.
                  </div>
            ';
      }
```

```
}

else {
      if($_POST) {
            echo '
                  <div class="msg">
                        Chyba! <br />
                        Uživatelské jméno a heslo nebylo vyplněno.
                  </div>
            ';
      }
}

?>
```

Ukázka kódu ze skriptu /components/login.php

1.9 Vkládání a úprava projektů

Samozřejmostí databázového systému je i možnost přidat nový projekt nebo upravit či odstranit stávající. Přidání projektu je možné provést po kliknutí na tlačítko **Nový projekt** umístěné v pravé části nad tabulkou projektů, k úpravě či odstranění projektu přejdeme po kliknutí na příslušnou ikonu ve sloupci **Akce** tabulky. Formulář pro úpravu projektu či vytvoření nového je velmi podobný, pro ukázku volím obrázek stránky pro úpravu projektu.



Obr.4: Stránka pro úpravu projektu

2. Databázový systém SQL

Při tvorbě svého projektu jsem pro snadnější práci s větším množstvím dat využil databázového systému. V této kapitole bych rád objasnil, jak jej ve svém projektu využívám a popsal základní práci s ním.

2.1 Co je to SQL

SQL je především interaktivní dotazovací jazyk - umožňuje získat odpovědi i na velmi komplikované dotazy téměř ihned. Je nástrojem neprocedurálním, s množinovým přístupem k datům a je jazykem standardizovaným, je srozumitelný, protože chápe data v podobě tabulek, což je snadno pochopitelné i uživatelům. Pracuje s relačními databázemi, ve kterých se uživatel dívá na data v podobě soustavy provázaných tabulek. Každá tabulka představuje množinu dat, která je uspořádaná v řádcích (záznamech) a sloupcích (položkách). Na hodnotu dat se uživatel odkazuje jako na prvek v matici.

2.2 Co je to MySQL

MySQL je relační databáze typu DBMS (database managment system) a vychází z deklarativního programovacího jazyka SQL (Structured Query Language). Je šířen jako Open Source.

Díky své licenci a rychlosti je v poslední době téměř nejoblíbenějším systémem. MySQL je malý, rychlý a jednoduchý databázový systém. Databáze MySQL má některá omezení, které obsahují jiné databázové systémy, např. robustní Oracle. Právě díky tomu dosahuje vynikající rychlosti.

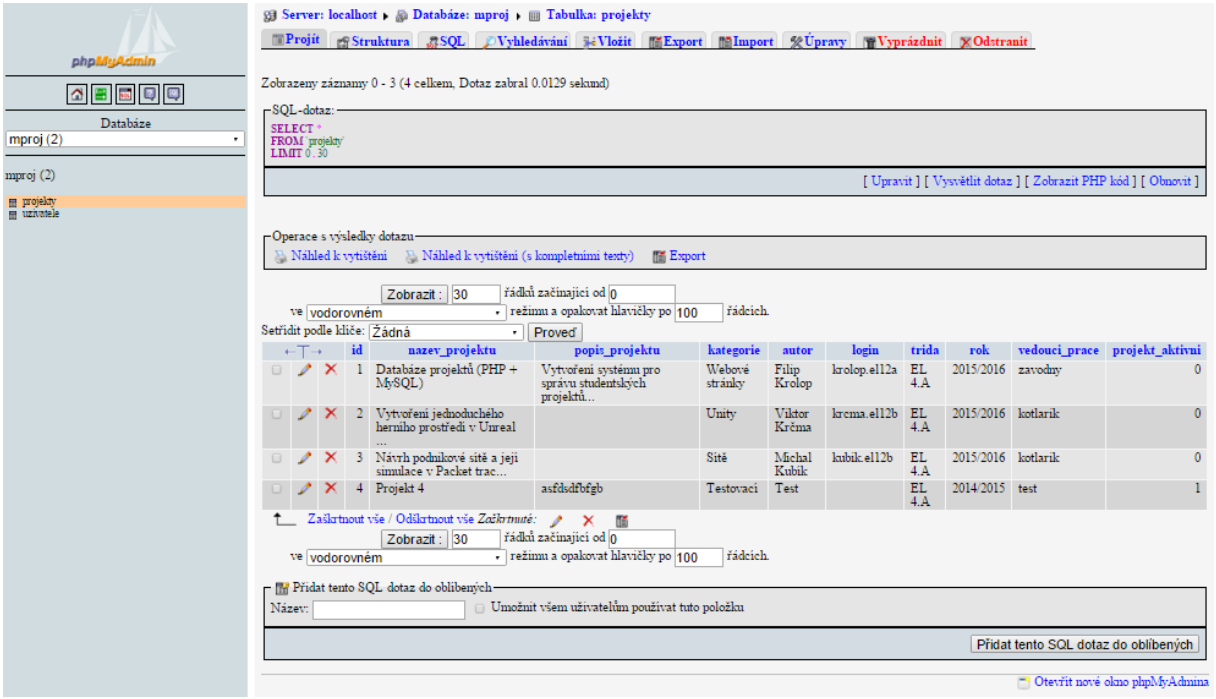
MySQL je zkratka z angl. My Structured Query Language = systém pro řízení databází. Do MySQL lze ukládat různá data (texty, obrázky atd.), s nimiž lze dále jednoduše pracovat (třídit, řadit, filtrovat apod.). Nejčastěji se MySQL používá ve spojení s jazykem PHP, které umožňuje přístup k uloženým datům.

Každá databáze v MySQL obsahuje tabulky, každá tabulka má sloupce a řádky – v každém řádku jsou záznamy předem určeného typu.

Databáze MySQL je jeden z prvních hojně rozšířených systémů. Práce s tímto systémem se dá využít v C, C++, Java, Perl, PHP, Python, Tcl, Visual Basic nebo .NET.

2.3 Co je to phpMyAdmin

Pro jednoduchou správu MySQL databází se používá nástroj PhpMyAdmin. PhpMyAdmin je Open Source program napsaný v PHP, který umožňuje zálohování, vytváření tabulek, vkládání, editaci a mazání záznamů v tabulkách, vytváření databází apod. PhpMyAdmin je pokročilý nástroj pro kompletní správu MySQL systému přes webové rozhraní.



Obr. 5: Ukázka prostředí phpMyAdmin

2.4 Co je to SQL injection

SQL injection je technika napadení databázové vrstvy programu vsunutím (odtud „injection“) kódu přes neošetřený vstup a vykonání vlastního, samozřejmě pozměněného, SQL dotazu. Toto nechtěné chování vzniká při propojení aplikační vrstvy s databázovou vrstvou (téměř vždy se totiž jedná o dva různé programy) a zabraňuje se mu pomocí jednoduchého escapování potenciálně nebezpečných znaků.

V klasickém případě je útok na internetové stránky prováděn přes neošetřený formulář, manipulací s URL nebo třeba i podstrčením zákeřně upravené cookie. Na internetu je však stále velké množství webů, spravovaných převážně nezkušenými programátory, kteří o této technice útoku prostě neví a tuto kritickou chybu opomíjejí.

Proti SQL injection je třeba zajistit webové formuláře všude tam, kde dochází ke komunikaci s MySQL databází a kde se na základě odeslaných formulářových dat provádí SQL dotaz.

Ukázka útoku na nezajištěný formulář:

Odesláním vyplněného přihlašovacího formuláře se na serveru provede MySQL podobný dotaz:

```
$vyberlogin = mysql_query("SELECT login, heslo, opravneni, blokovat
FROM uzivatele WHERE login='$login' AND heslo='$heslo'");
```

Pokud však uživatel zadá jako jméno **admin' or '1'='1';DROP TABLE uzivatele;--** ,dojde ke smazání celé tabulky uživatelů v databázi MySQL. Odeslaný SQL dotaz bude mít totiž následující podobu:

```
$vyberlogin = mysql_query("SELECT login, heslo, opravneni, blokovat
FROM uzivatele WHERE login='admin' or '1'='1';DROP TABLE uzivatele;-
-' AND heslo='$heslo'");
```

Podmínka **or '1'='1'** způsobí, že může být vybrán kterýkoli existující login, pokud uživatelský účet s názvem admin v databázi neexistuje (protože podmínka '1'='1' je vždy pravdivá), následný příkaz **DROP TABLE** smaže tabulku uživatelů a dvě pomlčky za tímto příkazem vytvoří ze zbylé části původního příkazu poznámku, takže k ověření správnosti hesla vůbec nedojde.

2.5 Zabezpečení proti SQL injection v mé práci

PHP nabízí vícero možností, jak ošetřit formulářová pole proti tomuto útoku, pro základní zabezpečení by měly ve většině případů stačit následující dva příkazy:

```
stripslashes($nasRetezec);
mysql_real_escape_string($nasRetezec);
```

stripslashes(); - odstraní z předávaného řetězce zpětná lomítka, „vyčistí“ řetězec

mysql_real_escape_string(); - escapování (zalomítkování) všech „nebezpečných“ znaků (\x00, \n, \r, \, ', ", \x1a) zpětným lomítkem. Za nebezpečné znaky se považují ty znaky, které dokáží zásadně pozměnit charakter SQL dotazu.

Použitím těchto dvou řádků skriptu jsem se snažil ošetřit všechny formulářové vstupy ve své práci.

Závěr

Výstupem je plně funkční webová stránka pro správu studentských projektů. Díky této maturitní práci jsem si prohloubil znalosti v oblasti tvoření webových stránek, jazyka PHP a následným propojením s databázovým systémem MySQL. Mimo dokumentaci byl k práci vytvořen propagační plakát, CD s přílohami a podklad do sborníku.

Seznam použité literatury a informačních zdrojů

Teorie a pojmy:

Co je HTML | Adaptic. Adaptic.cz. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://www.adaptic.cz/znalosti/slovnicek/html/

Co jsou Kaskádové styly | Adaptic. Adaptic.cz. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://www.adaptic.cz/znalosti/slovnicek/kaskadove-styly/

Co je JavaScript | Adaptic. Adaptic.cz. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://www.adaptic.cz/znalosti/slovnicek/javascript/

PHP /základy/. tvorba-webu.cz. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://www.tvorba-webu.cz/php/

Co je HTML | Adaptic. Adaptic.cz. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://www.adaptic.cz/znalosti/slovnicek/html/

Kurz: Programové vybavení 2. e-learning.spsbv.cz. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://e-learning.spsbv.cz/course/view.php?id=83

Co je to databáze MySQL?. artic-studio.net. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: https://www.artic-studio.net/slovnicek-pojmu/databaze-mysql/

SQL injection – Wikipedie. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-03-23]. Dostupné z: https://cs.wikipedia.org/wiki/SQL_injection

Skripty:

PHP Select Data From MySQL. w3schools.com. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://www.w3schools.com/php/php_mysql_select.asp

MySQL Commands. pantz.org. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: https://www.pantz.org/software/mysql/mysqlcommands.html

On/Off Flipswitch HTML5/CSS3 Generator - Proto.io. proto.io. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: https://proto.io/freebies/onoff/

Flat Tabbed Menu | CSS MenuMaker. cssmenumaker.com. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://cssmenumaker.com/menu/flat-tabbed-menu

CSS: Checkbox switch effect works only for the first checkbox - Stack Overflow. stackoverflow.com. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://stackoverflow.com/questions/35745201/css-checkbox-switch-effect-works-only-for-the-first-checkbox

Javascript: Delayed submit is not working correctly - Stack Overflow. stackoverflow.com. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://stackoverflow.com/questions/35748850/javascript-delayed-submit-is-not-working-correctly

PHP Login Form with Sessions | FormGet. FormGet.com. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://www.formget.com/login-form-in-php/

A Complete Guide to the Table Element | CSS-Tricks. CSS-TRICKS.com. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: https://css-tricks.com/complete-guide-table-element/

50+ Beautiful CSS HTML5 Login Form Templates - freshDesignweb. freshDesignweb.com. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: https://www.freshdesignweb.com/css-login-form-templates/

Autocomplete textbox using jQuery, PHP and MySQL - CodexWorld. CODEXWORLD.com. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://www.codexworld.com/autocomplete-textbox-using-jquery-php-mysql/

ZMĚNY V ROZVRHU. spsbv.cz. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://info.spsbv.cz/rozvrh/

Dotazník. spsbv.cz. [online]. 23.3.2016 [cit. 2016-03-23]. Dostupné z: http://info.spsbv.cz/anketa1/