

Programuj aplikácie
pre tvorbu webu

PHP

Programovanie
OOP PHP aplikácií

Peter Šoltýs

Knihu venujem rodine a priateľom

Peter Šoltýs

Obsah

Úvod	3
Vysvetlenie pojmov	4
Zdrojový kód	5
1. Výpis	6
2. Vloženie dát	9
3. Registrácia užívateľa	14
4. Vymazanie dát	18
5. Editovanie dát	21
6. Prihlásenie a odhlásenie	23
6.1 Prihlásenie	23
6.2 Odhlásenie	26
7. Sesion a Cookies	27
7.1 Cookies	27
7.2 Session	29
8. Export dát do xm	31
9. Zobrazenie dát z xml dokumentu	33
10. Export dát do php dokumentu	35
11. Stránkovanie	41
12. Fulltext	43
Záver	46
Príloha	47

Úvod

Kniha Programovanie php aplikácii Vás naučí programovať php aplikácie s použitím objektovo orientovaného programovania. Kniha je určená pre mierne pokročilých programátorov, ktorí majú záujem prejsť na moderný spôsob programovania php aplikácii. Kniha Vás prevedie tvorbou internetového katalógu. Presným postupom dokážete naprogramovať katalóg s exportom do xml a php dokumentu.

Peter Šoltýs

Vysvetlenie pojmov

Objektovo-orientované programovanie je moderný typ programovania, ktorý šetrí kód pre tvorbu php aplikácii. Objektovo-orientované programovanie je komplikovanejšie riešenie pre naprogramovanie aplikácie, ale obsahuje množstvo výhod, ktoré určite oceníte. S objektovo-orientovaným programovaním jednoducho vytvoríte väčšiu aplikáciu a kód môžete viackrát použiť.

Vysvetlenie pojmov pre OOP programátora

Triedy: Pre objekt existuje kód s metódami a hodnotami.

Objekty: Zahŕňa dáta pre fungovanie aplikácie.

Dedičnosť: Dedenie vlastnosti hlavnej triedy pre podtriedy.

Polymorfizmus: Je možné definovať triedy do viacerých kategórii.

Rozhranie: Špecifikuje funkcionality objektu.

Zapúzdrenie: Objekt chráni dáta pred prístupom .

Zdrojový kód

Zdrojový kód si môžete stiahnuť na [www stránke](http://www.chytreknihy.solydesign.info):

<http://www.chytreknihy.solydesign.info>

PHP aplikácie sú v knihe podrobne vysvetlené. Pre jednoduchú prácu som vytvoril malý framework. Aplikácie využívajú kód frameworku pre všetky aplikácie knihy **Programovanie aplikácií v OOP PHP**. Celý súbor frameworku je podrobne vysvetlený.

1.Výpis

Základom každej modernej aplikácie je výpis dát z databázy. Výpis je jednoduchý. Najprv je potrebné si všimnúť v súbore frameworku class.user.php vytvorenú triedu user. Pod triedou je vytvorený objekt pre pripojenie databázy.

public function __construct() znamená, že príkaz platí pre všetky funkcie objektu. Prihlasovacie údaje pre pripojenie databázy nájdeme v súbore **db_config.php**, kde po otvorení nájdeme prihlasovacie údaje ako **DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_DATABASE**.

Pre prácu s databázami slúži nástroj **phpMyAdmin**, kde si môžete vytvoriť databázu a jednotlivé tabuľky s atribútmi. phpMyAdmin je free riešenie, preto si ho môžete stiahnuť na adrese: **http://www.phpmyadmin.net**

Pred tým si však nainštalujte php server Xamp, Appserv alebo iný. Cez príkaz localhost/phpMyAdmin spustíte cez prehliadač phpMyAdmin.

Prvým krokom pre výpis dát je vložiť v mieste výpisu kód(**viewlink.php**):

```
<?php
include_once 'include/class.user.php';
$user = new User();

?>

<?php $user->get_listlink($subcategory);?>
```

Ďalší kód nájdeme (**class.user.php**)

```
include_once 'include/class.user.php';
```

Zahrňa súbor frameworku kde sú funkcie, ktoré používame pre naše aplikácie.

```
$user = new User();
```

Vytvára nový objekt User

```
<?php $user->get_listlink($subcategory);?>
```

Volá metódu `get_listlink` s premenou `$subcategory`

V súbore frameworku je volaná funkcia `get_listlink`.

```

public function get_listlink($subcategory){
    $query_select = "SELECT * FROM link WHERE subcategory='$subcategory'
ORDER BY id DESC";
$result_select = mysql_query($query_select) or die(mysql_error());
$rows = array();
while($row = mysql_fetch_array($result_select))
    $rows[] = $row;
foreach($rows as $rowss)
{
$link = $rowss['links'];
$namesite = $rowss['namesite'];

$email = $rowss['email'];
echo"<div style='color:white; width:600px; height:25px;'>";
echo"<div style='color:white; width:500px; height:25px; float:left;'>";
echo"<div style='color:white; width:400px; height:25px; float:left;'>";
echo"<div style='background:#FBFBFB; color:#4F4F4F; width:200px; height:25px; float:left;
font-size:14px;'>".$link."</div>";
echo"<div style='background:#FBFBFB; color:#4F4F4F; width:200px; height:25px;
float:right;font-size:14px;'>".$namesite."</div>";
echo"</div>";
echo"<div style='background:#FBFBFB; color:#4F4F4F; width:100px; height:25px;
float:right;font-size:14px;'><a href='viewxml.php?namesite=".$namesite.'">link</a></div>";
echo"</div>";
echo"<div style='background:#FBFBFB; color:#4F4F4F; width:100px; height:25px;
float:right;font-size:14px;'><a href='admin/page/".$namesite.".php'>stránka</a></div>";
Echo"</div>";
}
}
}
public function get_listlink($subcategory)

```

verejná funkcia get_listlink sa premenou \$subcategory

```

$query_select = "SELECT * FROM link WHERE subcategory='$subcategory' ORDER
BY id DESC";

```

Je to príkaz sql, ktorý hovorí vyber všetky atribúty tabuľky link, kde atribút subcategory sa rovná premennej \$subcategory a usporiadaj vzostupne dáta podľa id.

```

$result_select = mysql_query($query_select) or die(mysql_error());

```


Funkcia **mysql_query** vykoná príkaz SQL alebo zamietne prístup do databázy.

```
while($row = mysql_fetch_array($result_select))
```

Pre výpis dát z databázy slúži funkcia `mysql_fetch_array`, ktorá dáta preniesie do premennej `$row`.

```
$rows[] = $row;
```

Vytvoríme pole `$rows[]`, kde dáta sú dané z premennej `$row`.

```
foreach($rows as $rowss)
```

Pre moderný výpis v php slúži príkaz `foreach`, ktorý dáta poľa preniesie do premennej `$rowss`.

```
$link = $rowss['links'];
```

Načítanie premennej z databázy sa realizuje príkazom cez prenesenú premennú `$rowss`, ktorá vypíše z databázy všetky dáta s atribútom `links` do premennej `$link`.

Kód pod php kódom sú definované css vlastnosti. Pre reálny výpis slúži príkaz `echo`, ktorý vypíše premenné z databázy.

```
echo"<div style='background:#FBFBFB; color:#4F4F4F; width:200px; height:25px; float:left; font-size:14px;'>".$link."</div>";
```

Kliknite na kategóriu alebo subkategóriu pre zobrazenie dát.

2.Vloženie dát

Vloženie dát do databázy využijeme pri registrácii nového užívateľa alebo pridanie nového linku. To bude náš prípad, ktorý si popíšeme. Najprv si ukážeme formulár pre vloženie linku.(**pridatlink.php**)

```
<form action="insertlinktodb.php" method="post">
<div class="sumarray_form">
<div class="text_form">link stránky:(s http://)</div><div class="array_form"><input
type="text" size="33" name="links"></div>
</div>
<div class="sumarray_form">
<div class="text_form">názov stránky:</div><div class="array_form"><input type="text"
size="33" name="namesite"></div>
</div>
<div class="sumarray_form">
<div class="text_form">e-mail:</div><div class="array_form"><input type="text" size="33"
name="email"></div>
</div>
<div class="sumarray_form_long">
<div class="text_form_long">popis:</div><div class="array_form_long"><textarea rows="4"
cols="30" name="description"></textarea></div>
</div>
<div class="sumarray_form">
<div class="text_form">kategória:</div><div class="array_form"><select
name="category"><option value="">
<?php include_once 'include/class.user.php';
$category = new User();
?>
<?php $category->select_category(); ?>
</select></div>
</div>
<div class="sumarray_form">
<div class="text_form">subkategória:</div><div class="array_form"><select
name="subcategory" required><option value="">
<?php include_once 'include/class.user.php';
$subcategory = new User();
?>
```

```

<?php $subcategory->select_subcategory(); ?>
</select></div>
</div>
<div class="sumarray_form_long">
<div class="text_form_long">keywords:</div><div class="array_form_long"><textarea
rows="4" cols="30" name="keywords"></textarea></div>
</div>
<div class="sumarray_form">
<div class="text_form"></div><div class="array_form"><input type="submit" name="submit"
value="Pridaj www stránku"></div>
</div>
</Form>

```

```

<form action="insertlinktodb.php" method="post">

```

Formulár je nastavený na metódu post a spracovanie požiadavky je vedené na php skript **insertlinktodb.php**.

```

<div class="text_form">link stránky:(s http://)</div><div class="array_form"><input
type="text" size="33" name="links"></div>

```

Pomocou css je formulár vykreslený. Základy css sú nevyhnutné pre zvládnutie tejto knihy.

Textové pole formulára je definované príkazom `input type="text"`, veľkosť `size="33"` a názov pre spracovanie požiadavky `name="links"`.

```

<div class="sumarray_form">
<div class="text_form">kategória:</div><div class="array_form"><select
name="category"><option value="">
<?php include_once 'include/class.user.php';
$category = new User();
?>
<?php $category->select_category(); ?>
</select></div>
</div>

```

Pre výpis hodnôt poľa je vytvorená aplikácia. Vypíše všetky dáta z databázy. Postup je taký istý ako a pri aplikácii výpisu dát z databázy. V aplikácii je potrebné vytvoriť objekt.

```
<div class="sumarray_form">
<div class="text_form"></div><div class="array_form"><input type="submit" name="submit"
value="Pridaj www stránku"></div>
</div>
```

Pre spustenie skriptu je vytvorené tlačítko `input type="submit"` s názvom pre spracovanie skriptu `name="submit"` a názov pre zobrazenie `value="Pridaj www stránku"`.

Ďalej budeme pokračovať v analýze skriptu. ([insertlinktodb.php](#))

```
<?php

include_once 'include/class.user.php';
$link = new User();

if (isset($_REQUEST['submit'])){
    $test = $link->insert_link($links, $namesite, $email, $description, $category,
$subcategory, $keywords);
    if ($test) {
        // Registration Success
        echo "<div style='text-align:center'>Registrácia bola úspešná<a
href='linkgenerate.php?namesite=" . $namesite . "'>Kliknete tu</a> pre prihlásenie</div>";
    }else {

        echo "<div style='text-align:center'>Registrácia nebola úspešná a e-mail alebo
užívateľské

mono už existuje</div>";
    }
}
?>
```

Príkaz `include` Vám vloží framework kód, ktorý budeme používať pre vloženie dát.

```
include_once 'include/class.user.php';
```

Vytvoríme objekt

```
$link = new User();
```

```
if (isset($_REQUEST['submit'])){
```

funkcia `isset` kontroluje, či bolo potvrdenie tlačítka s menom `submit`

```
$test = $link->insert_link($links, $namesite, $email, $description, $category, $subcategory,
$keywords);
```

Premenná \$test prechádza s objektom do metódy insert_link a má úlohu spracovať premenné \$links, \$namesite, \$email, \$description, \$category, \$subcategory, \$keywords pomocou vložené frameworku. (**class.user.php**)

```
public function insert_link
($links,$namesite,$email,$description,$category,$subcategory,$keywords){

    $check = mysql_query("SELECT * FROM link WHERE links='$links' OR
namesite='$namesite'") or die(mysql_error());

    $count_row = mysql_num_rows($check);

    if ($count_row == 0){//echo "k3";
        $result = mysql_query("INSERT INTO link SET links='$links',
namesite='$namesite',email='$email', description='$description', category='$category',
subcategory='$subcategory', keywords='$keywords'") or die(mysql_error());

        return $result;
    }
    else {return false;}
}
```

Funkcia insert_link s inicializovanými premennými.

```
$Links,$namesite,$email,$description,$category,$subcategory,$keywords
public function
insert_link($links,$namesite,$email,$description,$category,$subcategory,$keywords)

$check = mysql_query("SELECT * FROM link WHERE links='$links' OR
namesite='$namesite'") or die(mysql_error());
```

mysql_query je príkaz pre vykonanie sql príkazu. Ten nám hovorí: Vyber všetky atribúty tabuľky link, kde atribút links sa rovná premennej \$links alebo atribút namesite sa rovná premennej \$namesite.

```
$count_row = mysql_num_rows($check);
```

Funkcia mysql_num_rows nám spočíta počet riadkov v tabuľke.

```
if ($count_row == 0)
```

Ak sa počet riadkov rovná 0 potom vloží do databázy hodnoty premenných \$links,\$namesite,\$email,\$description,\$category,\$subcategory,\$keywords.

```
$result = mysql_query("INSERT INTO link SET links='$links',  
namesite='$namesite',email='$email', description='$description', category='$category',  
subcategory='$subcategory', keywords='$keywords'") or die(mysql_error());
```

```
mysql_query je príkaz, ktorý nám vykoná sql príkaz v databáze tabuľky link.  
else {return false;}
```

Ak sa počet riadkov nerovná 0 potom príkaz sa nevykoná.

3. Registrácia užívateľa

Registrácia užívateľa je bežnou aplikáciou www stránok alebo portálov, či katalógov. Ľudský povedané registrácia užívateľa je vloženie dát do databázy. Registrácia užívateľa je často používanou aplikáciou pri získaní informácii o klientovi. Pre e-shop alebo pre rôzne portály je registrácia užívateľa častou operáciou vloženia dát do databázy. (**registeruser.php**)

```
<form action="reg_script.php" method="post">
  <div class="sum_form">
    <div class="l_form">Vaše meno:</div>
    <div class="r_form"><input type="text" name="fullname"></div>
  </div>
  <div class="sum_form">
    <div class="l_form">Prihlasovacie meno:</div>
    <div class="r_form"><input type="text" name="uname"></div>
  </div>
  <div class="sum_form">
    <div class="l_form">Email:</div>
    <div class="r_form"><input type="email" name="uemail"></div>
  </div>
  <div class="sum_form">
    <div class="l_form">Heslo:</div>
    <div class="r_form"><input type="password" name="upass"></div>
  </div>
  <div class="l_form"><input class="btn" type="submit" name="submit"
value="Registrovat"></div>

</Form>
```

Najprv si vysvetlíme formulár pre registráciu užívateľa. Pomocou html vytvoríme formulár s názvami textových polí Vaše meno – fullname, Prihlasovacie meno – uname, Email - uemail, Heslo - upass. Potvrdením tlačítka odošleme dáta cez príslušnú aplikáciu do databázy.

Aplikácia prenosu dát do frameworku. (reg_script.php)

```
<?php

include_once 'include/class.user.php';
$user = new User();

if (isset($_REQUEST['submit'])){
    $register = $user->reg_user($fullname, $uname, $upass, $uemail);
    if ($register) {

        echo "<div style='text-align:center'>Registrácia bola úspešná <a
href='index.php'>Kliknite tu</a> </div>";
    }else {

        echo "<div style='text-align:center'>Registrácia nebola úspešná login alebo heslo je
už používané </div>";
    }
}
?>
```

Príkazom include je vložený framework. Vytvoril sa nový objekt User.

```
include_once 'include/class.user.php';
$user = new User();
```

Podmienka kontroluje či bolo stlačené tlačítko pre odoslanie formuláru.

```
if (isset($_REQUEST['submit']))
```

Dochádza k prenosu dát cez funkciu frameworku do databázy cez jednotlivé premenné.

```
$register = $user->reg_user($fullname, $uname, $upass, $uemail);
```

Podmienka if vykoná cez premennú \$register, či bola splnená podmienka alebo nie.

```
if ($register)
```


Funkcia reg_user vo využitom frameworku. (**class.user.php**)

```
public function reg_user($name,$username,$password,$email){
```

```
    $password = md5($password);
```

```
    $check = mysql_query("SELECT * FROM users WHERE  
uname='$username' OR uemail='$email'") or die(mysql_error());
```

```
    $count_row = mysql_num_rows($check);
```

```
if ($count_row == 0){
```

```
    $result = mysql_query("INSERT INTO users SET  
uname='$username', upass='$password', fullname='$name', uemail='$email'") or  
die(mysql_error());
```

```
    return $result;
```

```
    }
```

```
    else {return false;}
```

```
}
```

Vo funkcii function reg_user sú zahrnuté premenné \$name,\$username,\$password,\$email, ktorých hodnoty boli vložené po splnení podmienky do databázy.

```
public function reg_user($name,$username,$password,$email)
```

```
$password = md5($password);
```

Heslo je zakryptované cez metódu md5.

Cez príkaz mysql_query sa vykoná sql príkaz (výber atribúty z tabuľky users, kde atribút uname sa rovná hodnote premennej \$username alebo atribút uemail sa rovná hodnote premennej \$email).

```
$Check = mysql_query("SELECT * FROM users WHERE uname='$username' OR uemail='$email'") or die(mysql_error());
```

Funkcia `mysql_num_rows` spočíta počet riadkov tabuľky z premennej `$check`.
`$count_row = mysql_num_rows($check);`

Podmienka ak počet riadkov tabuľky sa rovná 0 potom dáta budú vložené do databázy.

```
if ($count_row == 0){  
    $result = mysql_query("INSERT INTO users SET  
uname='$username', upass='$password', fullname='$name', uemail='$email'") or  
die(mysql_error());
```

Ak nebude počet riadkov 0, potom podmienka nie splnená a dáta sa do databázy nevložia.

4.Vymazanie dát

Na e-shopoch, ale na všetkých portáloch musíme pre zrušenie objednávky alebo niektorých dát vymazať riadok databázy. Jedným klikom vykonáme vymazanie jedného atribútu alebo viacerých atribútov databázy.

(**viewlinkdelete.php**). PHP skript nájdete v priečinku admin. Otvorte ho a môžete vymazať záznam linku.

```
<div class="body">
<div style="color:white; width:600px; height:25px;">
<div style="color:white; width:500px; height:25px; float:left;">
<div style="color:white; width:400px; height:25px; float:left;">
<div style="background:#4F4F4F; color:#FBFBFB; width:200px; height:25px; float:left; font-size:14px;">link</div>
<div style="background:#4F4F4F; color:#FBFBFB; width:200px; height:25px; float:right;font-size:14px;">name site</div>
</div>
<div style="background:#4F4F4F; color:#FBFBFB; width:100px; height:25px; float:right;font-size:14px;">delete</div>
</div>

<div style="background:#4F4F4F; color:#FBFBFB; width:100px; height:25px; float:right;font-size:14px;">export</div>
</div>
<?php
    include_once 'include/class.user.php';
    $user = new User();

?>
<?php $user->get_listlink();?>
</div>
```

Pomocou html/css môžeme vytvoriť grafickú štruktúru výpisu dát z databázy.

```
<?php
    include_once 'include/class.user.php';
    $user = new User();

?>
<?php $user->get_listlink();?>
```

Príkazom include zahrnieme framework do databázy.

```
include_once 'include/class.user.php';
```

Vytvoríme objekt, v ktorom je volaná funkcia `get_listlink()`

```
$user = new User();  
<?php $user->get_listlink($subcategory);?>
```

V ďalšom súbore zahrnieme framework a vytvoríme objekt User.
(linkdelete.php)

```
<?php  
include_once 'include/class.user.php';  
$user = new User();  
$id = $_GET['id'];  
?>  
<?php $user->get_linkdelete($id);?>
```

Na prenesenie dát id sa používa metóda GET. Metóda GET sa zobrazí v url adrese.

```
$id = $_GET['id'];  
Cez objekt user je volaná funkcia v frameworku get_linkdelete($id), ktorá obsahuje premennú id.
```

```
<?php $user->get_linkdelete($id);?>
```

Vo frameworku sa nachádza aplikácia `get_linkdelete($id)`, ktorá vymaže na základe premennej id riadok v databáze.**(class.user.php)**

```
public function get_linkdelete($id){  
  
    $query_select = "DELETE FROM link WHERE id='$id'";  
    $result_select = mysql_query($query_select);  
    if($result_select){  
        echo "článok bol vymazaný";  
    }else{  
        echo "článok nebol vymazaný";  
    }  
}
```

Cez príkaz `DELETE FROM link` vymaže riadok z tabuľky `link`, kde `WHERE id='$id'` atribút tabuľky `id` sa rovná premennej `$id`.

```
$query_select = "DELETE FROM link WHERE id='$id'";
```

Cez funkciu `mysql_query` vykoná sa sql príkaz premennej `$query_select`.

```
$result_select = mysql_query($query_select);
```

V aplikácii existuje podmienka, ktorá vyjadruje ak existuje premenná `$result_select` – ak bol vykonaný SQL príkaz vymazania riadku tabuľky.
`if($result_select)`

Potom vypíše cez príkaz `echo` článok bol vymazaný.

```
echo "článok bol vymazaný";  
ak nebola splnená podmienka vypíše cez príkaz echo.  
echo "článok nebol vymazaný";
```

Jednoduchá aplikácia vymazania riadku tabuľky sa využíva pri vývoji aplikácii bežne. Každú aplikáciu dôkladne preštudujte, aby ste vedeli vytvoriť katalóg pre pridávania spätných odkazov.

5.Editovanie dát

Chceli by sme zmeniť prihlasovacie údaje. Pre úpravu databázy slúži príkaz **UPDATE**. Pre editácia dát potrebujem výpis dát, ktoré chceme editovať. Pre editáciu vytvoríme formulár užívateľského mena a hesla. (**changelogin.php**)

```
<form action="getlogin.php" method="post">
<div class="sumarray_form">
<div class="text_form">login:</div><div class="array_form"><input type="text" size="33"
name="login"></div>
</div>
<div class="sumarray_form">
<div class="text_form">password:</div><div class="array_form"><input type="text"
size="33" name="password"></div>
</div>
<div class="sumarray_form">
<div class="text_form"></div><div class="array_form"><input type="submit"
name="submit"></div>
</Div>
</Form>
```

Súbor nájdeme v priečinku admin/changelogin.php.

<input type="text" zobrazí formulárový textový element s veľkosťou 33 a s pridaným menom login. Pre potvrdenie dát slúži tlačítko vytvorené<input type="submit" s názvom submit.

Nasledovný súbor nájdete na **admin/ getlogin.php**

```
include_once 'include/class.user.php';
```

Príkazom include zahrnieme framework súbor class.user.php.

```
$change = new User();
```

Vytvoríme nový objekt.

```
if (isset($_REQUEST['submit'])){
extract($_REQUEST);
```

Vytvorená podmienka je vykonaná, keď bolo stlačené tlačítko s názvom submit.

```
$test = $change->get_getlogin($login, $password);
```

Premenné \$login, \$password prechádzajú do skriptu frameworku.

Ďalšia podmienka na kontroluje splnenie úlohy skriptu s premenou \$test . Ak je podmienka splnená prihlasovacie údaje boli editované. Ak nie dáta zostanú nezmenené.

```

public function get_getlogin($login, $password){
$passwords = md5($password);

        $query_select = "UPDATE admin SET login = '$login', password = '$passwords'
WHERE uid = '1'";
$result_select = mysql_query($query_select);
if($result_select){
    echo "login a heslo zmenené<a href='index.php'>back</a>";
}else{
    echo "login a heslo nezmenené<a href='index.php'>back</a>";
}
}
}

```

Funkcia `get_getlogin` zahŕňa premenné `$login`, `$password`. Heslo v premennej `$password` je zašifrované cez `md5`. SQL príkazom **UPDATE** budeme editovať dáta v stĺpcoch tabuľky `login` a `password` s premennými, kde hodnota stĺpca `uid` tabuľky `admin` sa rovná `1`.

```

$query_select = "UPDATE admin SET login = '$login', password = '$passwords' WHERE
uid = '1'";

```

Cez PHP príkaz `mysql_query` sa vykoná sql príkaz pre editovanie dát.

```

$result_select = mysql_query($query_select);

```

Podmienka cez premennú `$result_select` je uskutočnená, ak existuje premenná `$result_select`, ak neexistuje premenná editovanie prihlasovacích údajom sa nezrealizuje.

6. Prihlásenie a odhlásenie

6.1 Prihlásenie

Pre prihlásenie slúži webový formulár zverejnený na www stránke. Jednoducho užívateľ vyplní polia login a password a potvrdí tlačídlom prihlásenie.

(**admin.php**)

```
<div class="admin">
<div class="name">Admin</div>
<form action="administrator.php" method="post">
<div class="login"><input type="text" name="login"></div>
<div class="password"><input type="password" name="password"></div>
<div class="button"><input type="submit" name="submit" value="prihlás sa"></div>
</Form>
```

Formulár je navrhnutý podľa HTML/CSS.

```
<form action="administrator.php" method="post">
Akcia formulára po potvrdení tlačítka bude prebiehať podľa php súboru.
```

Administrátor podľa metódy post.

```
<input type="text" name="login">
```

Vstupné textové pole login vyplníme premenou.

```
<input type="password" name="password">
```

Textové pole pre heslo vyplníme pre prihlásenie heslom.

```
<input type="submit" name="submit" value="prihlás sa">
```

Pre potvrdenie slúži tlačítka submit s hodnotou prihlás sa.

Pre prenos dát zo vstupných poli slúži súbor (**administrator.php**).

```
<?php
    session_start();
    include_once 'include/class.user.php';
    $user = new User();

    if (isset($_REQUEST['submit'])) {

        $test = $user->check_login($login, $password);
        if ($test) {

            header("location:index.php");
        }else {

            echo 'Zlý login alebo heslo';
        }
    }
?>
```


Štartuje session pre prihlásenie.

```
session_start();
```

Framework je vložený do súboru administrator.php.

```
include_once 'include/class.user.php';
```

Vytvorený je objekt user.

```
$user = new User();
```

Realizuje sa podmienka po potvrdení tlačítka submit.

```
if (isset($_REQUEST['submit']))
```

Ak podmienka je splnená funkcia frameworku kontroluje prihlasovacie údaje s databázou s premennými \$login a \$password.

```
$test = $user->check_login($login, $password);
```

Existuje ďalšia podmienka v súbore Ak existuje hodnota premennej \$login (premenná pre prihlasovacie údaje).

Potom

```
if ($test) {
```

```
    header("location:index.php");
```

```
}else {
```

```
    echo 'Zlé prihlasovacie meno a heslo';
```

```
}
```

sa vykoná presmerovanie na stránku index.php.

```
header("location:index.php");
```

Ak neexistuje hodnota premennej \$test (premenná pre prihlasovacie údaje) potom vypíše cez príkaz echo Zlé prihlasovacie meno a heslo.

Funkcia **check_login** s premennými \$login, \$password(class.user.php)

```
    public function check_login($login, $password){
```

```
        $passwords = md5($password);
```

```
        $result = mysql_query("SELECT uid from admin WHERE login='$login' and password='$password'");
```

```
        $user_data = mysql_fetch_array($result);
```

```
        $count_row = mysql_num_rows($result);
```

```
        if ($count_row == 1) {
```

```
            $_SESSION['login'] = true;
```

```
            $_SESSION['uid'] = $user_data['uid'];
```

```
            return true;
```

```
        }
```

```
        else{
```

```
            return false;
```

```
        }
```

```
    }
```

Popíšeme si kód pre kontrolu prihlasovacích údajov.

Existuje premenná `$password` zašifrovaná podľa šifry md5. Šifra je daná pre bezpečnosť prihlásenia užívateľa `$passwords = md5($password)`;

Príkaz **`mysql_query`** vykoná sql príkaz: Vyberie stĺpec tabuľky `uid` z tabuľky `admin`, kde stĺpec `login` sa rovná premennej `$login` a súčasne stĺpec `password` sa rovná premennej `$password`.

```
$result = mysql_query("SELECT uid from admin WHERE login='$login' and password='$password');
```

Funkcia **`mysql_fetch_array`** urobí výpis z databázy všetkých vybraných stĺpcov tabuľky.

```
$user_data = mysql_fetch_array($result);
```

Funkcia **`mysql_num_rows`** spočíta počet riadkov v tabuľke s danou podmienkou.

```
$count_row = mysql_num_rows($result);
```

Ak premenná `$count_row` sa rovná 1 potom sa prihlásenie uskutoční.

```
if ($count_row == 1) {
```

tak, že vytvorí sa premenná `$_SESSION` s premenou `login` a `uid`, ktorá bude identifikovať užívateľa pre prihlásenie.

```
$_SESSION['login'] = true;
    $_SESSION['uid'] = $user_data['uid'];
    return true;
```

Ak podmienka nebude splnená(nájde viac záznamov alebo 0 záznamov s tým istým loginom a heslom tak prihlásenie sa neuskutoční.

Else

```
{
    return false;
}
```

6.2 Odhlásenie

Odhlásenie je nevyhnutnou aplikáciou pri každom portály, ktorý pracuje s dátami užívateľa. Kód si podrobne vysvetlíme: (**logout.php**). Skript Vám na servery nemusí fungovať pre nezapnutie funkcie session.

```
session_start();
include_once 'include/class.user.php';
$user = new User();
if (isset($_GET['q'])){
    $user->user_logout();
    header("location:index.php");
}
```

Štartuje prihlásenie session. Každý sa určite pýta, čo je session. Je to vygenerované číslo, ktoré chráni dáta pred zneužitým.

```
session_start();
Do súboru vložíme framework
include_once 'include/class.user.php';
Vytvoríme objekt User.
```

```
$user = new User();
```

Existuje podmienka, ktorá overuje kliknutie na odkaz odhlásiť. Identifikátor pre odhlásenie je q. Ak užívateľ na odkaz neklikol tal samozrejme odhlásenie sa nezrealizuje.

```
if (isset($_GET['q']))
```

Objekt načíta funkciu z frameworku user_logout();

```
$User->user_logout();
```

Ak podmienka odhlásenie je splnená presmeruje sa užívateľ na stránku index.php.

```
header("location:index.php");
```

Pri odhlásení sa zruší session.(**class.user.php**)

Vo frameworku pre odhlásenie existuje funkcia user_logout.

```
$_SESSION['login'] = FALSE;
```

premenná session login je nepravdivá.

```
Session_destroy();
```

Zruší kód session.

7.Sesion a cookies

7.1 Cookies

Cookies je aplikácia ktorá slúži k vytvoreniu ochrannej premennej. Slúži na zabezpečenie prihlásenia užívateľa a zabránenie prístupu iným užívateľom. Váš prehliadač si zapamätá premennú cookies.

Kód cookies je nasledovný:

```
setcookie('id', $row[id]);  
setcookie('user_name', $row[user_name]);
```

Prehliadač si zapamätá premenu id a user_name a príkazom echo premennú vypíše:

```
<?php echo $_COOKIE['user_name']; ?>
```

Pre odhlásenie sa používa presmerovanie na stránku index.php.

```
header("location:index.php");
```

Celý kód prihlásenie je tu:(user_login_cokkie/index.php)

```
?php  
$message="";  
if(count($_POST)>0) {  
$conn = mysql_connect("localhost","","");  
mysql_select_db("", $conn);  
$result = mysql_query("SELECT * FROM session WHERE user_name=" .  
$_POST["user_name"] . " and password = " . $_POST["password"]. "");  
$row = mysql_fetch_array($result);  
if(is_array($row)) {  
setcookie('id', $row[id]);  
setcookie('user_name', $row[user_name]);  
}else {  
$message = "Nesprávny login alebo heslo!";  
}  
}  
if(isset($row[id])) {  
header("Location:user_dashboard.php");  
}  
?>
```

```

<html>
<head>
<title>User Login</title>
<link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
<form name="frmUser" method="post" action="">
<div class="message"><?php if($message!="") {echo $message; }?></div>
<table border="0" cellpadding="10" cellspacing="1" width="500" align="center">
<tr class="tableheader">
<td align="center" colspan="2">Zadaj prihlasovacie údaje </td>
</tr>
<tr class="tablerow">
<td align="right">Prihlasovacie heslo</td>
<td><input type="text" name="user_name"></td>
</tr>
<tr class="tablerow">
<td align="right">Heslo</td>
<td><input type="password" name="password"></td>
</tr>
<tr class="tableheader">
<td align="center" colspan="2"><input type="submit" name="submit" value="prihlasit"></td>
</tr>
</table>
</form>
</Body>
</html>

```

Cookies je starším spôsobom ochrany modernejším je session.

7.2 Session

Najprv musíme štartovať session `session_start()`;

```
$_SESSION["user_id"] = $row[id];  
$_SESSION["user_name"] = $row[user_name];
```

Premenná session id a user_name sa načíta z databázy. Premenná session bude zapamätateľná určitú dobu alebo pri odhlásení sa zruší premenná.

Pre odhlásenie existuje funkcia unset, ktorá zruší premennú session. Na začiatku musíme štartovať premennú session.

```
<?php  
session_start();  
unset($_SESSION["user_id"]);  
unset($_SESSION["user_name"]);  
header("Location:index.php");  
?>
```

Ak existuje premenná session cez príkaz echo vypíše hodnotu premennej user_name.

```
<?php echo $_SESSION["user_name"]; ?>
```

Celý kód prihlásenia je tu:([user_login_session/index.php](#))

```
<?php  
$message="";  
if(count($_POST)>0) {  
$conn = mysql_connect("localhost", "", "");  
mysql_select_db("", $conn);  
$result = mysql_query("SELECT * FROM session WHERE user_name=" .  
$_POST["user_name"] . " and password = " . $_POST["password"]. "");  
$row = mysql_fetch_array($result);  
if(is_array($row)) {  
$_SESSION["user_id"] = $row[id];  
$_SESSION["user_name"] = $row[user_name];  
}else {  
$message = "Nesprávny login alebo heslo!";  
}  
}  
if(isset($_SESSION["user_id"])) {  
header("Location:user_dashboard.php");  
}  
?>
```

```
<html>
<head>
<title>User Login</title>
<link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
<form name="frmUser" method="post" action="">
<div class="message"><?php if($message!="") {echo $message; }?></div>
<table border="0" cellpadding="10" cellspacing="1" width="500" align="center">
<tr class="tableheader">
<td align="center" colspan="2">Zadaj prihlasovacie údaje </td>
</tr>
<tr class="tablerow">
<td align="right">Prihlasovacie heslo</td>
<td><input type="text" name="user_name"></td>
</tr>
<tr class="tablerow">
<td align="right">Heslo</td>
<td><input type="password" name="password"></td>
</tr>
<tr class="tableheader">
<td align="center" colspan="2"><input type="submit" name="submit" value="prihlasit"></td>
</tr>
</table>
</form>
</Body>
</html>
```

8. Export dát do xml

Export dát do xml súboru je dôležitý v mnohých prípadoch. XML súbor je využívaný pre tlačenie, konvertovanie do iných typoch súborov. Pri vložení linku sa automaticky vyexportuje xml dokument s daným názvom v priečinku page. Popíšeme si skript pre prenos premennej. ([linkgenerate.php](#))

```
<?Php
    include_once 'include/class.user.php';
    $user = new User();
    $namesite = $_GET['namesite'];
?>
<?php $user->get_generate($namesite);?>
```

Zahrnieme framework do súboru:

```
include_once 'include/class.user.php';
```

Vytvoríme objekt User

```
$user = new User();
```

Cez metódu GET je prenesená premenná odkazom

```
<a href='viewxml.php?namesite=".$namesite.'">link</a>
```

```
$namesite = $_GET['namesite'];
```

Vytvorená funkcia `get_generate($namesite)` realizuje prenos premennej pre generovanie dokumentu xml.

```
<?php $user->get_generate($namesite);?>
```

Vo frameworku nájdeme funkciu `get_generate`, ktorá generuje dokument XML (`include/class.user.php`)

```
public function get_generate($namesite){
    $result = mysql_query("select * from link WHERE namesite='$namesite'");
    $xml = new SimpleXMLElement('<xml/>');
    while($row = mysql_fetch_assoc($result)) {
        $mydata = $xml->addChild('mydata');
        $mydata->addChild('id',$row['id']);
        $mydata->addChild('link',$row['links']);
        $mydata->addChild('namesite',$row['namesite']);
        $mydata->addChild('email',$row['email']);
        $mydata->addChild('description',$row['description']);
        $mydata->addChild('category',$row['category']);
        $mydata->addChild('subcategory',$row['subcategory']);
    }
    $fp = fopen("./page/$namesite.xml","wb");
    fwrite($fp,$xml->asXML());
    fclose($fp);
}
```


Funkcia `mysql_query` vykoná SQL príkaz, kde vyberie všetky stĺpce z tabuľky `link` kde stĺpec `namesite` sa rovná premennej `namesite`.

```
$result = mysql_query("select * from link WHERE namesite='$namesite'");
```

Vytvorí sa jednoduchý prvok objektu `<xml/>`

```
$xml = new SimpleXMLElement('<xml/>');
```

Pridajú sa hodnoty stĺpcov pre xml objekt. Funkcia **`mysql_fetch_assoc`** vykoná zobrazenie dát xml dokumentu. Jednotlivé stĺpce tabuľky sa zobrazia z db cez príkaz `$row['links']` a je priradený názov stĺpca tabuľky.

```
while($row = mysql_fetch_assoc($result)) {  
    $mydata = $xml->addChild('mydata');  
    $mydata->addChild('id',$row['id']);  
    $mydata->addChild('link',$row['links']);  
    $mydata->addChild('namesite',$row['namesite']);  
    $mydata->addChild('email',$row['email']);  
    $mydata->addChild('description',$row['description']);  
    $mydata->addChild('category',$row['category']);  
    $mydata->addChild('subcategory',$row['subcategory']);  
}
```

Príkaz `fopen` vytvorí dokument s cestou `/page` pre písanie s atribútom `wb`.

```
$fp = fopen("./page/$namesite.xml","wb");
```

Príkazom `fwrite` vypíše xml dokument s hodnotami z databázy.

```
fwrite($fp,$xml->asXML());
```

Funkcia `fclose` ukončí výpis dokumentu

```
fclose($fp);
```

V priečinku `page` sa vytvorí xml dokument, ktorý môžete otvoriť a preštudovať kód.

9. Zobrazenie dát z xml dokumentu

Zobrazenie dát z xml dokumentu nám poslúži v prípade, že chceme urobiť zálohu dát do xml a následne ich zobraziť. Po vložení nového odkazu klikneme na danú kategóriu, ktorú sme vybrali pri vložení linku. Kliknete na odkaz link a zobrazia sa Vám vložené údaje z xml dokumentu, Ak klikneme na odkaz stránka nezobrazí sa nám nič, pretože export dát do php stránky nebol vykonaný.

O exporte php dokumentu sa dočítate v ďalšej kapitole.

Zobrazia sa Vám údaje z xml dokumentu.

```
<?php
    include_once 'include/class.user.php';
    $user = new User();
    $namesite = $_GET['namesite'];
    ?>
<?php $user->get_viewxml($namesite);?>
```

Do súboru zahrnieme framework:

```
include_once 'include/class.user.php';
```

Vytvorený je objekt User

```
$user = new User();
```

Pre zobrazenie je dôležitý prenos premennej \$namesite.

```
$namesite = $_GET['namesite'];
```

Funkcia `get_viewxml($namesite)` je volaná z framework pre výpis dát z xml súboru.

```
<?php $user->get_viewxml($namesite);?>
```

Kompletný kód pre výpis xml dokumentu.(include/class.user.php)

Funkcia `get_viewxml` nám vypíše dáta z html dokumentu.

```
public function get_viewxml($namesite){
    $source = "./page/$namesite.xml";
    $xmlstr = file_get_contents($source);
    $xmlcont = new SimpleXMLElement($xmlstr);
    foreach($xmlcont as $url)
    {
```

```

echo"<div style='color:white; width:500px; height:25px;'>";
echo"<div style='background-color: white; color:#2F2F2F; width:100px; height:25px;
float:left; font-size:14px;'>Link</div>";
echo"<div style='background-color: #3F3F3F; color:white; width:400px; height:25px;
float:right; font-size:14px;'>".$url->links."</div>";
echo"</div>";
echo"<div style='color:white; width:500px; height:25px;'>";
echo"<div style='background-color: white; color:#2F2F2F; width:100px; height:25px;
float:left; font-size:14px;'>Názov stránky</div>";
echo"<div style='background-color: #3F3F3F; color:white; width:400px; height:25px;
float:right; font-size:14px;'>".$url->namesite."</div>";
echo"</div>";
echo"<div style='color:white; width:500px; height:25px;'>";
echo"<div style='background-color: white; color:#2F2F2F; width:100px; height:25px;
float:left; font-size:14px;'>Email</div>";
echo"<div style='background-color: #3F3F3F; color:white; width:400px; height:25px;
float:right; font-size:14px;'>".$url->email."</div>";
echo"</div>";
echo"<div style='color:white; width:500px; height:25px;'>";
echo"<div style='background-color: white; color:#2F2F2F; width:100px; height:25px;
float:left;'>Podkategória</div>";
echo"<div style='background-color: #3F3F3F; color:white; width:400px; height:25px;
float:right; font-size:14px;'>".$url->category."</div>";
echo"</div>";
echo"<div style='color:white; width:500px; height:25px;'>";
echo"<div style='background-color: white; color:#2F2F2F; width:100px; height:25px;
float:left; font-size:14px;'>Podkategória</div>";
echo"<div style='background-color: #3F3F3F; color:white; width:400px; height:25px;
float:right; font-size:14px;'>".$url->subcategory."</div>";
echo"</div>";
echo"<div style='color:white; width:500px;'>";
echo"<div style='background-color: white; color:#2F2F2F; width:100px; float:left; font-
size:14px;'>Obsah</div>";
echo"<div style='background-color: #3F3F3F; color:white; width:400px; float:right; font-
size:14px;'>".$url->description."</div>";echo"</div>";
}
}

```

Cez premennú \$source vytvorí cestu, z ktorej xml dokument bude načítaný. \$source = "./page/\$namesite.xml"; Funkcia get_viewxml nám prenesie premennú \$namesite, ktorá identifikuje xml súbor pre jeho zobrazenie.

```

$xmlstr = file_get_contents($source);
$xmlcont = new SimpleXMLElement($xmlstr);
foreach($xmlcont as $url)

```

Príkazy načítajú reťazec s premennej \$source pre výpis sa využije funkcia foreach, ktorá prenesie hodnoty \$xmlcont as \$url. Pre zobrazenie hodnôt premennej \$url s usmernením na párový xml tag links Kód pre zobrazenie dát links. \$url->links.

10.Export dát do php dokumentu

Export dát do php dokumentu sa vytvára vtedy, keď je potrebné riešiť seo marketing. Napríklad vytvoril som katalóg a každá nová pridaná stránka - firma katalógu musí byť vyhľadávačmi zaindexovaná. Export dát sa vykonáva z databázy, kde boli pridané dáta užívateľom. Pôjdete do adminu (priečink admin) otvoríte index.php kliknete na odkaz viewlink, zobrazia sa Vám všetky vložené záznamy. Kliknete na export a vytvoríte php dokument. Zobrazíme si hlavnú stránku a kliknete na odkaz stránky linka otvoríme php stránku. V admine do priečinku page a article sa nám vyexportujú php dokumenty link a article. Otvoríme v admine dokument. (**Articleexport.php**)

```
<?php
    include_once 'include/class.user.php';
    $user = new User();
$namearticle = $_GET['namearticle'];
?>
<?php $user->get_createhtmlart($namearticle);?>
</div>
```

Do súboru sme vložili framework cez príkaz include.

```
Include_once 'include/class.user.php';
```

Vytvorili sme objekt User

```
$user = new User();
```

Prenos premennej \$namearticle sa realizuje cez metódu GET

```
$namearticle = $_GET['namearticle'];
```

Volanie funkcie z frameworku sa vykona cez kód (include/class.user.php)

```
<?php $user->get_createhtmlart($namearticle);?>
```

Teraz si popíšeme kód funkcie vo frameworku `get_createhtmlart()`

```
public function get_createhtmlart($namearticle)
{
$result=mysql_query("SELECT * FROM article WHERE namearticle = '$namearticle'
ORDER BY id DESC LIMIT 1");
if ($result) {
    while ($record = mysql_fetch_array($result)) {
        $namearticle=$record[namearticle];
        $description=$record[description];
        $keywords=$record[keywords];
        $email=$record[email];
    }
}
```

```

$text = "
<html>
  <head>
    <meta http-equiv='content-type' content='text/html; charset=windows-1250'>
    <meta name='keywords' content='$keywords' />
    <meta name='description' content='$description' />
    <link rel='stylesheet' type='text/css' href='style.css'>
    <title>$namesite</title>
  </head>
  <body>
    <div class='wrapper'>
<div class='header'>
<div class='logo'></div>
<div class='fulltext'>
</div>
</div>
<div class='menu'>
<ul>
<li><a href='back.php'>back</a></li>
</ul>
</div>
<div class='content'>
<div class='wrap'>
<div class='left'>
</div>
<div class='body'>
<div style='color:white; width:500px; height:25px;'>
<div style='background-color: white; color:#2F2F2F; width:100px; height:25px; float:left;
font-size:14px;'>Názov článku</div>
<div style='background-color: #3F3F3F; color:white; width:400px; height:25px; float:right;
font-size:14px;'>".$namearticle."</div>
</Div>
div style='color:white; width:500px; height:25px;'>
<div style='background-color: white; color:#2F2F2F; width:100px; height:25px; float:left;
font-size:14px;'>Email</div>
<div style='background-color: #3F3F3F; color:white; width:400px; height:25px; float:right;
font-size:14px;'>".$email."</div>
</div>
<div style='color:white; width:500px;'>
<div style='background-color: white; color:#2F2F2F; width:100px; float:left; font-
size:14px;'>Obsah</div>
<div style='background-color: #3F3F3F; color:white; width:400px; float:right; font-
size:14px;'>".$description."</div>

```

```

</div>
</div>
</div>
<div class='right'></div>
</div>
<div class='footer'><div class='link'><a href='#'>Peter Šoltýs - Solydesign</a></div</div>
</div>
  </body>
</html>

```

```

";
}
}
if(!$namearticle)
echo "html súbor nebol vytvorený.";

exit;
}
else
{
$file_directory = "article/";
$filename = $file_directory.$namearticle . ".php";

```

```

$fp = fopen($filename, "w");
$fw = fwrite( $fp, $text );
fclose($fp);
}
}

```

Funkcia `get_createhtmlart` obsahuje premennú `$namearticle`.

```
public function get_createhtmlart($namearticle)
```

Vykoná sa cez funkciu **mysql_query** SQL príkaz, kde vyberie všetky stĺpce tabuľky `article`, kde stĺpec tabuľky sa rovná premennej `$namearticle` a výsledky vypíše podľa poradia `id` pričom sa zobrazí iba 1 výsledok.

```
$Result=mysql_query("SELECT * FROM article WHERE namearticle = '$namearticle'
ORDER BY id DESC LIMIT 1");
```

Ak premenná `$result` existuje a bol vykonaný sql príkaz potom sa zrealizuje výpis pomocou funkcie **mysql_fetch_array** jednotlivých stĺpcov tabuľky ako napríklad `$namearticle=$record[namearticle]`;

```

if ($result) {
    while ($record = mysql_fetch_array($result)) {
        $namearticle=$record[namearticle];
        $description=$record[description];
        $keywords=$record[keywords];
        $email=$record[email];
    }
}

```

Cez premennú text sa zrealizuje výpis jednotlivých premenných v php dokumente. Pomocou HTML/CSS sa navrhla štruktúra php dokumentu.

```

$text = "
<html>
  <head>
    <meta http-equiv='content-type' content='text/html; charset=windows-1250'>
    <meta name='keywords' content='$keywords' />
    <meta name='description' content='$description' />
    <link rel='stylesheet' type='text/css' href='style.css'>
    <title>$namesite</title>
  </head>
  <body>
    <div class='wrapper'>
<div class='header'>
<div class='logo'></div>
<div class='fulltext'>
</div>
</div>
<div class='menu'>
<ul>
<li><a href='back.php'>back</a></li>
</ul>
</div>
<div class='content'>
<div class='wrap'>
<div class='left'>
</Div>
<div class='body'>
<div style='color:white; width:500px; height:25px;'>
<div style='background-color: white; color:#2F2F2F; width:100px; height:25px; float:left;
font-size:14px;'>Názov článku</div>
<div style='background-color: #3F3F3F; color:white; width:400px; height:25px; float:right;
font-size:14px;'>".$namearticle."</div>
</div>

```

```

<div style='color:white; width:500px; height:25px;'>
<div style='background-color: white; color:#2F2F2F; width:100px; height:25px; float:left;
font-size:14px;'>Email</div>
<div style='background-color: #3F3F3F; color:white; width:400px; height:25px; float:right;
font-size:14px;'>".$email."</div>
</div>
<div style='color:white; width:500px;'>
<div style='background-color: white; color:#2F2F2F; width:100px; float:left; font-
size:14px;'>Obsah</div>
<div style='background-color: #3F3F3F; color:white; width:400px; float:right; font-
size:14px;'>".$description."</div>
</div>
</div>
</div>
<div class='right'></div>
</div>
<div class='footer'><div class='link'><a href='#'>Peter Šoltýs - Solydesign</a></div</div>
</div>
  </body>
</html>

```

Ak neexistuje hodnota v premennej \$namearticle, potom sa vypíše výsledok:
 html súbor nebol vytvorený a inak sa nastaví cestu pre vytvorenie php súbor v
 priečinku article, kde názov bude taký aká je hodnota premennej \$namearticle
 vypísana z databázy.

```

if(!$namearticle)
{
echo "html súbor nebol vytvorený.";
exit;
}
else
{
$file_directory = "article/";
$filename = $file_directory.$namearticle . ".php";

$fp = fopen($filename, "w");
$fw = fwrite( $fp, $text );
fclose($fp);
}

```


Cez funkciu fopen sa vytvorí php súbor s daným názvom a do súboru cez funkciu fwrite sa vypíše obsah premennej \$text. Súbor sa zatvorí cez funkciu **fclose**, ktorá zatvorí súbor.

```
$fp = fopen($filename, "w");  
fwrite( $fp, $text );  
fclose($fp);
```

11. Stránkovanie

Stránkovanie sa využíva pri načítaní ďalšej strany. Pričom ďalšie výpisy z databázy sa realizujú na ďalšej strane. V súbore je zahrnutý framework cez príkaz `include_once 'include/class.user.php'`; Stránkovanie nájdete iba v priečinku `admin`. Otvorte si framework a vo funkcii `get_listusers()` nájdete stránkovanie, ktoré si bližšie popíšeme.

Je vytvorený objekt `$user = new User();`

A nakoniec je volaná funkcia `get_listusers()` z frameworku

```
<?php
    include_once 'include/class.user.page.php';
    $user = new User();

?>
<?php $user->get_listusers();?>
```

Stránkovanie sa realizuje cez funkciu `get_listarticle`, kde začiatková hodnota je 0.

`$start=0`; a maximálna strana je číslo 3.

```
public function get_listarticle(){
    $start=0;
    $limit=3;
```

Ak existuje premenná `id`, potom premenná sa prenesie cez metódu `GET`.

A urobí sa výpočet na základe vzorca `$start=($id-1)*$limit`;

```
if(isset($_GET['id']))
{
    $id=$_GET['id'];
    $start=($id-1)*$limit;
}
```

Výpis výsledkov z databázy vyberie stĺpce tabuľky z databázy limitovaný premennými `LIMIT $start, $limit`

```
$query_select = "SELECT * FROM article LIMIT $start, $limit";
```

Pre výpis stránkovania sme vložili do frameworku tento kód.

```
$rows=mysql_num_rows(mysql_query("select * from users"));
$total=ceil($rows/$limit);
echo "<ul class='page'>";
    for($i=1;$i<=$total;$i++)
    {
        if($i==$id) {echo "<li class='current'>".$i."</li>"; }

        else {echo "<li><a href='?id=".$i.">".$i."</a></li>"; }
    }
echo "</ul>";
```

Funkcia **mysql_query** vykoná SQL príkaz a funkcia **mysql_num_rows** nám spočíta počet riadkov v tabuľke. Funkcia **ceil** vráti najbližšiu hodnotu. Pre zobrazenie stránkovania existuje interval `for($i=1;$i<=$total;$i++)`
Cez príkaz `echo` sa vypíše stránkovanie. Ak existujú veľa výsledkov tak sa zobrazia v stránkovaní aj ďalšie čísla stránok s výpismi z databázy.

12.Fulltext

Na moderných www stránkach nájdete vyhľadávanie. Fulltext slúži užívateľom stránky pre rýchle vyhľadávanie informácií. Dáta sa vyhľadávajú z databázy, do ktorej administrátor alebo užívateľ stránky vložil dáta.

Pre vyhľadávanie sme vytvorili formulár, kde akciu vykoná php skript fulltext.php. Formulár pre fulltext nájdete v php súbore.

(header.php)

```
<form action="fulltext.php" method="get">
<div class="wrap_fulltext">
<div class="array_fulltext"><input type="text" size="30" name="search"></div>
<div class="button_fulltext"><input type="submit" name="send" value="vyhľadaj"></div>
</div>
</form>
```

Vyhľadávanie prevedieme cez php súbor s kódom.**(fulltext.php)**

```
<?php
include_once 'include/class.user.php';
$user = new User();
$search = $_GET['search'];
?>
<?php $user->get_listfulltext($search);
?>
(include/ class.user.php)
```

Zahrnieme do súboru framework príkazom include, include_once 'include/class.user.php';

Vytvoríme objekt User \$user = new User();
Prenesieme premennú z formulára metódou GET \$search = \$_GET['search'];
Na koniec voláme funkciu get_listfulltext z frameworku
\$User->get_listfulltext(\$search);

Popíšeme si funkciu get_listfulltext s frameworku class.user.

(Include/class.user.php)

```
public function get_listfulltext($search){
$query_select = "SELECT * FROM link WHERE namesite LIKE \"%".$_GET['search']."%\"
OR keywords LIKE \"%".$_GET['search']."%\"";
$result_select = mysql_query($query_select) or die(mysql_error());
$rows = array();
$count_row = mysql_num_rows($result_select);
```

```

f ($count_row > 0)
{
while($row = mysql_fetch_array($result_select))
    $rows[] = $row;
foreach($rows as $rowss)
{
$link = stripslashes($rowss['links']);
$name-site = stripslashes($rowss['name-site']);
$email = stripslashes($rowss['email']);
echo "<div style='color:white; width:600px; height:25px;'>";
echo "<div style='color:white; width:400px; height:25px; float:left;'>";
echo "<div style='background:#FBFBFB; color:#4F4F4F; width:200px; height:25px; float:left; font-size:14px;'>".$link."</div>";
echo "<div style='background:#FBFBFB; color:#4F4F4F; width:200px; height:25px; float:right;font-size:14px;'>".$name-site."</div>";
echo "</div>";
echo "<div style='background:#FBFBFB; color:#4F4F4F; width:200px; height:25px; float:right;font-size:14px;'>".$email."</div>";
echo "</div>";
}
}
else
echo "nič sa nenašlo";
}

```

Funkcia `get_listfulltext` obsahuje premennú `$search`, čo je hodnota vložená do formulára pri vyhľadávaní.

```

$query_select = "SELECT * FROM link WHERE name-site LIKE \"%".$_GET['search']."%\" OR keywords LIKE \"%".$_GET['search']."%\"";
$result_select = mysql_query($query_select) or die(mysql_error());

```

Cez funkciu `mysql_query` sa vykoná SQL príkaz pre vyhľadanie výsledku.

`name-site LIKE \"%".$_GET['search']."%` V SQL príkaze je porovnávaný stĺpec `name-site` ku premennej `$_GET['search']`.

```

Dáta su definované v poli array();
$rows = array();
$count_row = mysql_num_rows($result_select);

```

Počet výsledkov nám vykoná funkcia `mysql_num_rows`.

```
if ($count_row > 0)
```

Ak počet je väčší ako 0 potom sa zrealizuje výpis výsledkov vyhľadávania.

```
while($row = mysql_fetch_array($result_select))
    $rows[] = $row;
foreach($rows as $rowss)
{
$link = stripslashes($rowss['links']);
$name = stripslashes($rowss['name']);
$email = stripslashes($rowss['email']);
```

Výpis dát sa realizuje funkciou **mysql_fetch_array**. Pole \$rows[] prenesie dáta výsledkov cez príkaz foreach. Funkcia stripslashes vyčistí dáta z databázy od lomítok.

Pomocou vytvorenia premenných \$link = stripslashes(\$rowss['links']); vypíšeme cez príkaz echo echo"<div style='background:#FBFBFB; color:#4F4F4F; width:200px; height:25px; float:left; font-size:14px;'>".\$link."</div>"; hodnotu premennej z databázy.

Ak množstvo výsledkov sa rovná 0 potom skript vypíše, že nič nenašlo.

```
else
    echo "nič sa nenašlo";
}
```

Záver

Kniha Vás určite naučila rýchlo a jednoducho tvoriť pomocou malého frameworku www stránky. Za kladné aplikácie obsahuje framework a preto tvorba aplikácií je rýchla a jednoduchá.



vyhladať

[Home](#)

[Vložiť link](#)

[Vložiť článok](#)

[Web](#)

[Články](#)

[Kontakt](#)

[Organizácie](#)

[Centrá voľného času](#), [Cirkevné organizácie](#), [Domovy dôchodcov](#), [Domov detí a mládeže](#), [knihy](#), [úrady](#), [Združenia](#)

[Cestovanie](#)

[Cestovné agentúry](#), [Info o počasi](#), [Zábavné atrakcie](#)

[Auto-moto](#)

[Autobazár](#), [Prislišenstvo](#), [Inzercia](#), [Predaj áut](#), [Poistenie áut](#), [Autoservisy](#)

[Firmy](#)

[IT firmy](#), [Remeselné firmy](#), [Ekonomické firmy](#), [Advokácke firmy](#), [Reality](#), [Finančné služby](#)

[Ekonomika](#)

[účtovníctvo](#), [Administratívne služby](#), [Banky](#), [Poistovne](#), [Teambuilding](#), [Personalistika](#)

[Priemysel a výroba](#)

[Chemický priemysel](#), [Automatizácia](#), [Strojársky priemysel](#), [Potravínársky priemysel](#), [Textilný priemysel](#), [Kozmetika](#)

[Počítače](#)

[Software](#), [Hardware](#), [Grafika](#), [Tvorba www stránky](#), [Programovanie](#), [Sieťe](#)

[Veda a technika](#)

[Chémia](#), [Biológia](#), [Informatika](#), [Matematika](#), [Medicina](#), [Astronómia](#), [Archeológia](#)

[Internetové obchody](#)

[Software](#), [Hardware](#), [Kozmetika](#), [Nábytok](#), [Knihy](#), [Hračky](#), [Hudba](#), [Hudobné nástroje](#)

[Vzdelávanie](#)

[Gymnázium](#), [Stredné odborné školy](#), [Vysoké školy](#), [Konzervatória](#)

[Reštaurácie a ubytovanie](#)

[Pizza](#), [Pub](#), [Reštaurácie](#), [Donášková služba](#), [Hoteli](#), [Penzióny](#), [Ambulancie](#), [Drogová závislosť](#), [Kliniky](#), [Nemocnice](#), [Chaty](#), [Ubytovne](#)

[Zdravotníctvo](#)

O autorovi

Peter Šoltýs (1986)

Autor knihy Programovanie OOP PHP aplikácii mal cieľ vytvoriť publikáciu, ktorá podrobne vysvetľuje PHP kód aplikácie. Je to piata kniha autora. Peter Šoltýs sa už 10 rokov venuje tvorbe webov s použitím CMS redakčných systémov alebo PHP programovaním. Programovanie je koníčkom autora aj profesiou. Už veľa rokov je informatika hlavnou náplňou života a kreativity osobnosti autora.

O knihe

Úlohou knihy je podrobne vysvetliť PHP aplikácie pre tvorbu webu. Každý riadok kódu je vysvetlený. Kniha je určené pre pokročilejších programátorov, ktorí majú zvládnuté základy PHP. Publikácie je rozdelená do dvanástich kapitol, kde v každej kapitole je podrobne opísaná aplikácia.

Prečo by ste knihu preštudovať? PHP je silný nástroj pre tvorbu www stránok alebo portálov, či nových CMS systémov.

Objektovo-orientované programovanie je moderným typom programovania, ktorý šetrí čas a skrakuje kód. Verím, že kniha Vám pomôže pochopiť OOP PHP programovanie a stane sa pre Vás hobby. Informatika je naozaj krásna veda, ktorá ponúka priestor pre kreativitu. Dobrým informatikom sa stanete vtedy, keď získate vzťah s dámou pekným menom Informatika. Budete môcť zakladať niečo nové, čo robí život krajším.