

PHPPamokos.lt

(/)

Kelias: PHP online-kursas pradedantiesiems (<http://phpamokos.lt/php-kursas/php-kursas-pradedantiesiems/4>) > 16. Mūsų pilnas projektas

16. Mūsų pilnas projektas

Atėjo laikas visa tai, ko jau išmokome, pritaikyti praktikoje. Bent jau didelę dalį. Priminsiu - šio kurso eigoje kuriame **krepšinio aikštelių duomenų bazės** projektą. Po aikštelių realiai gali slėptis bet kokie objektai - galėsite pakeisti projekto temą pagal save. Mūsų projektas turės tokias pagrindines funkcijas:

- Aikštelių paieška
- Konkrečios aikštelės atvaizdavimas
- Administratoriaus prisijungimas ir aikštelių valdymas
- Tekstiniai puslapiai - apie mus ir pan.

Pradėkime nuo plataus mūsų projekto vaizdo - tiesiog sukurkime pagrindinius failus, kol kas nesigilindami į jų turinį.

Projekto failų struktūra

Dažniausia praktika, skirstant viso projekto struktūrą, turėti vieną pagrindinį **index.php** failą, kuriame yra daug visokių **include()** funkcijų, priklausomai nuo to, koks puslapis yra užkraunamas. Mūsų atveju - tai bus kažkas tokio:

index.php:

```
session_start();
include('header.php');
include('menu.php');

$action = (isset($_GET['action'])) ? $_GET['action'] : '';
switch ($action) {
    case 'apie': include('about.php'); break;
    case 'aikstele': include('court.php'); break;
    case 'paieska': include('search.php'); break;
    case 'kontaktai': include('contact.php'); break;
    case 'admin': include('admin.php'); break;
    default: include('home.php'); break;
}

include('footer.php');
```

Kaip matote, turėsime URL struktūrą tokio pavidalo: **<http://localhost/php/index.php?action=xxxxx>** kur "localhost/php" priklauso nuo to, į kurį katalogą patalpinote savo projektą, o "xxxxx" - kokį puslapį norite užkrauti.

Atkreipkite dėmesį į du dalykus: kaip mes gauname kintamąjį **\$action** (tikriname, ar **\$_GET** parametras egzistuoja, kitaip gautume PHP klaidą) ir su **switch** operatoriaus funkcija **default** užkrauname **home.php**, jeigu **\$action** nėra lygus nei

vienam iš mūsų išvardintų puslapių arba yra tuščias.

Header ir Footer - HTML griaučiai

Toliau pereikime prie **header.php** ir **footer.php**. Ten bus tiesiog HTML struktūra - puslapio pagrindai, kad teisingai būtų atvaizduojama naršyklėje.

header.php:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Krepšinio aikštelės</title>
</head>
<body>
<div>
<a href="index.php">Krepšinio aikštelės</a>
</div>
<div>
```

footer.php:

```
</div>
<div>
<p>© PHPPamokos.lt 2014</p>
</div>
</body>
</html>
```

Tokia yra minimali HTML dokumento struktūra, kurios mums reikia. Kol kas nesigilinkime į dizainą ir apipavidalinimą - tą kiek vėliau apiforminsime su CSS.

Pagrindinis meniu

Menu tiesiog išvardiname visus puslapius - vėlgi HTML gabaliukas:

menu.php:

```
<div>
<a href="index.php">Pagrindinis</a> |
<a href="index.php?action=paieska">Paieška</a> |
<a href="index.php?action=apie">Apie</a> |
<a href="index.php?action=kontaktai">Kontaktai</a>
</div>
```

Vidiniai puslapiai - minimaliai

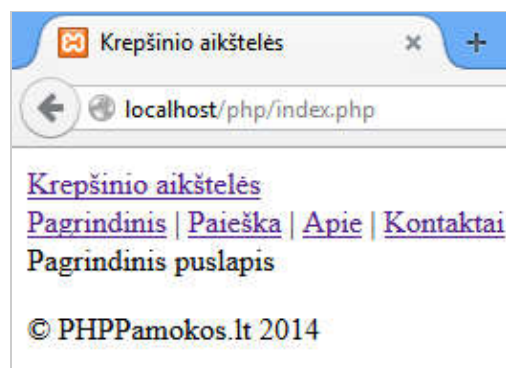
Dabar sukurkime vidinius puslapius - tiesiog juose užpildysime pavadinimus, kol kas.

Tai būtų **home.php**, **search.php**, **about.php**, **contact.php**.

Pvz, **home.php**:

```
echo 'Pagrindinis puslapis';
```

Šiam etapui vizualiai turėtume turėti kažką tokio:



Aikštelių valdymas - duomenų bazė

Ok, turime vizualius griaučius, dabar pagalvokime apie duomenis, kaip bus saugomos mūsų aikštelės. Mums realiai reikia dviejų duomenų bazės lentelių - **aikštelės** ir **miestai**. Tokias ir sukurkime:

cities

id

title

courts

id

city_id

address

description

create_time

Labai paprasta, ar ne? Taigi eikime į phpMyAdmin ir fiziškai sukurkime tas lenteles. Nepamirškime laukams **id** uždėti savybes **primary** ir **auto_increment**.

Table	Action	Rows
<input type="checkbox"/> cities	Browse Structure Search Insert Empty Drop	0
<input type="checkbox"/> courts	Browse Structure Search Insert Empty Drop	0
2 tables	Sum	0

Aikštelių valdymas - administratoriaus prisijungimas

Atėjo laikas susitvarkyti su sesijomis. Tiksliau, pirmiausia sukurkime puslapį administratoriaus prisijungimui - paslėpime

jį nuo išorės, kad jį žinotų tik tie, kas žino adresą. Tai bus **index.php?action=admin** adresas, kuriame bus prisijungimo forma, ir jei administratorius sėkmingai prisijungia - tada jį perkelia į tą patį **index.php?action=admin**, bet jau su sukurta sesija.

admin.php:

```
if (isset($_POST['login'])) {
    if (isset($_POST['name']) && $_POST['name'] == 'admin' && isset($_POST['password'])
    && $_POST['password'] == 'krepsinis') {
        $_SESSION['username'] = 'admin';
    }
}

if (isset($_SESSION['username'])) {
    ?>
    <div>
        <a href="index.php?action=admin&function=courts">Aikštelės</a> |
        <a href="index.php?action=admin&function=cities">Miestai</a> |
        <a href="logout.php">Atsijungti</a>
    </div>
    <?php
} else {
    ?>
    <br />
    <form action="" method="post">
        Vardas:
        <br />
        <input type="text" name="name" />
        <br /><br />
        Slaptažodis:
        <br />
        <input type="password" name="password" />
        <br /><br />
        <input type="submit" name="login" value=" Prisijungti " />
    </form>
    <?php
}
```

Trumpai tariant - viršuje patikriname, ar nėra vykdomas prisijungimas, ir jei taip - tada patikriname slaptažodį bei sukuriame sesiją. O tada - jei sesija yra, tada parodome meniu, o jei ne - prisijungimo formą.

Na, ir toliau einame prie konkretaus redagavimo. Sukurkime failus **cities.php** ir **courts.php**, kuriuos "inklūdinsime" į **admin.php**, daugmaž taip:

```
if (isset($_SESSION['username'])) {
    // ...
    $function = (isset($_GET['function'])) ? $_GET['function'] : '';
    if ($function == 'cities') include('cities.php');
    if ($function == 'courts') include('courts.php');
} else { ...
```

O mūsų **cities.php** failo pagrindas atrodoys taip:

```
<h2>Miestai</h2>
<table border="1" cellpadding="2" cellspacing="0">
  <tr>
    <th>Miestas</th>
    <th>Veiksmas</th>
  </tr>
  <? if (isset($cities) && count($cities) > 0): ?>
    <? foreach ($cities as $city): ?>
      <tr>
        <td><?=$city['title'];?></td>
        <td>
          <a href="index.php?action=admin&function=cities&
edit=<?=$city['id'];?>">Redaguoti</a>
          <a href="index.php?action=admin&function=cities&delete=<?=$city['id'];?>"
onclick="return confirm('Ar tikrai?')">Trinti</a>
        </td>
      </tr>
    <? endforeach; ?>
  <? else: ?>
    <tr>
      <td colspan="2">Miestų dar nėra įvesta.</td>
    </tr>
  <? endif; ?>
</table>

<hr />
<form action="" method="post">
  Naujas miestas:
  <br />
  <input type="text" name="title" />
  <input type="submit" name="insert_submit" value=" Saugoti " />
</form>
```

Vizualiai - turim štai ką:



Klasė darbui su duomenų baze

Dabar - pradėkime dirbti su duomenimis. Failo **index.php** viršuje įrašykime duomenų bazės prisijungimo eilutę (parametrus pakeiskite pagal savo vartotoją/slaptąžodį):

```
session_start();
$db = new PDO('mysql:host=localhost;dbname=php;charset=utf8', 'xxxxxx', 'xxxxxx');
```

Visą darbą su duomenų baze iškelsime į **klases** ir tam sukursime atskirą katalogą **classes** - laikysimės objektinio programavimo principų. Štai kaip atrodys klasė **cities** - joje sukursime visas funkcijas miestų apdorojimui: miestų sąrašas, konkretaus miesto vaizdavimas, miesto įterpimas, atnaujinimas ir trynimasis.

classes/cities.php:

```
class Cities {

    function getCities() {
        global $db;
        $stmt = $db->query('SELECT * FROM cities');
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }

    function addCity($title) {
        global $db;
        $stmt = $db->prepare("INSERT INTO cities (title) VALUES (:title)");
        $stmt->execute(array(':title' => $title));
    }

    function getCity($id) {
        global $db;
```

```
$stmt = $db->prepare("SELECT * FROM cities where id = :id");
$stmt->execute(array(':id' => (int)$id));
return $stmt->fetch(PDO::FETCH_ASSOC);
}

function updateCity($id, $title) {
    global $db;
    $stmt = $db->prepare("UPDATE cities set title = :title where id = :id");
    $stmt->execute(array(':title' => $title, ':id' => (int)$id));
}

function deleteCity($id) {
    global $db;
    $stmt = $db->prepare("DELETE FROM cities where id = :id");
    $stmt->execute(array(':id' => (int)$id));
}

}

$class_cities = new Cities();
```

Čia gali būti keli nesuprantami (ar pamiršti) dalykai:

- **global \$db**: index.php faile sukūrėme \$db objektą darbui su DB tai kad jį panaudotume funkcijos viduje, turime jį iškviešti, nes kitaip funkcijos viduje jis "nematomas"
- **\$class_cities = new Cities()**: kad būtų patogiau, sukuriame objektą iškart klasės faile, kad galėtume jį po to naudoti visur. Pavadinau \$class_cities, kad po to nesupainiotume kur nors su \$cities kintamuoju, kuris reikš miestų sąrašą.
- **(int)\$id**: prieš kintamąjį esantis tipas užtikrina, kad kintamasis yra skaičius, jei jis nėra skaičius tai konvertuojamas į 0.

Toliau - panaudokime tas funkcijas: failo **cities.php** viršuje įterpkime tokias eilutes:

```
if (isset($_POST['insert_submit'])) $class_cities->addCity($_POST['title']);
if (isset($_POST['update_submit'])) $class_cities->updateCity($_POST['id'],
$_POST['title']);
if (isset($_GET['delete'])) $class_cities->deleteCity($_GET['delete']);
```

Ir paskutinis dalykas - redagavimo forma: tam tiesiog patikrinkime, ar yra parametras 'edit':

```
if (isset($_GET['edit'])) {
    $city = $class_cities->getCity($_GET['edit']);
    // .... redagavimo forma
} else {
    // .... miestų sąrašas
}
```

Viskas, miestų valdymas baigtas! Pačių aikštelių valdymą palieku jūsų namų darbams, tai bus beveik copy-paste darbas, tik su vienu dalyku - pabandykite padaryti, kad prie aikštelės galima būtų pasirinkti miestą iš sąrašo su <select> komanda.

Pagrindinis puslapis - naujausios aikštelės

Pereikime prie išorės: pagrindiniame puslapyje pavaizduokime naujausias aikšteles. Tam panaudosime tas pačias mūsų sukurtas klases (ar jaučiate objektinio programavimo privalumus?) - tai tiesiog sukurkime naują funkciją klasėje Courts ir iškvieskime ją iš **home.php**:

classes/courts.php:

```
function getNewestCourts() {
    global $db;
    $stmt = $db->query('SELECT courts.*, cities.title as city FROM courts
        JOIN cities ON courts.city_id = cities.id
        ORDER BY courts.id desc
        LIMIT 10');
    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}
```

home.php:

```
<?php
$courts = $class_courts->getNewestCourts();
?>
<h2>Naujausios aikštelės</h2>
<ul>
<? foreach ($courts as $court): ?>
<li>
<b><?=$court['address'];?>, <?=$court['city'];?></b>
<div style="margin:5px 0 10px 0"><?=$court['description'];?></div>
</li>
<? endforeach; ?>
</ul>
```

Aikštelių paieška

Patalpinkime paieškos formą iškart po pagrindiniu meniu - tai **index.php** pridedame po meniu tokią eilutę:

```
include('search_form.php');
```

Na ir pats failas **search_form.php**:

```
<?php
$cities = $class_cities->getCities();
?>
<div>
<form action="index.php">
```



```
<input type="hidden" name="action" value="paieska" />
<select name="city_id">
<? foreach ($cities as $city): ?>
<option value="<?=$city['id'];?>"
<? if (isset($_GET['city_id'])
&& $_GET['city_id'] == $city['id']) echo ' selected="selected"'; ?>
><?=$city['title'];?></option>
<? endforeach; ?>
</select>
<input type="text" name="address" />
<input type="submit" name="search" value="Ieškoti" />
</form>
</div>
```

Ir tada faile **search.php** pateikiame rezultatus:

```
<?php
$courts = $class_courts->searchCourts($_GET['city_id'], $_GET['address']);
?>

<h2>Aikštelių paieškos rezultatai</h2>

<ul>
<? foreach ($courts as $court): ?>
<li>
<b><?=$court['address'];?>, <?=$court['city'];?></b>
<div style="margin:5px 0 10px 0"><?=$court['description'];?></div>
</li>
<? endforeach; ?>
</ul>
```

Na, ir kas liko - tekstiniai puslapiai: "Apie mus" ir "Kontaktai", tai juos užpildykite patys HTML kalba.

Šis mini-projektas jau veikia, ir jūs galite ant jo "užmauti" norimą dizainą ir išplėsti pačius objektus: pridėti laukų, pakeisti pačius pavadinimus, vietoje aikštelių daryti bet ką kitą - kirpyklas, odontologijos kabinetus ir t.t.

< 15. Darbas su MySQL duomenų baze (<http://phpmamokos.lt/php-kursas/php-kursas-pradedantiesiems/4/pamoka/31>)

(c) 2015-2018. Visais klausimais kreipkitės povilas@laraveldaily.com