|  |  |  |
| --- | --- | --- |
| **Laboratorium Projektowania Aplikacji Internetowych 1**  Wydział Elektrotechniki Automatyki i Informatyki  Politechnika Świętokrzyska | | |
| Studia: **Stacjonarne I stopnia** | Kierunek: **Informatyka** | |
| Data wykonania:  **05.12.2017** | Zespół: **06** | Grupa: **3ID13B** |
| Godzina wykonania:  **8:00 – 9:30** | Skład zespołu:   1. **Bartłomiej Osak** 2. **Tomasz Pasternak** | |
| Numer laboratorium: | Temat ćwiczenia: | |
| **4** | **PHP** | |

**Zadanie 2.**

**Obsłużyć połączenie z serwerami baz danych MySQL.**

1. **Wyświetlając dane z tabel.**
2. **Wstawiając dane do tabel.**
3. **Poprawiając dane w tabeli.**
4. **Usuwając dane z tabeli.**
5. **Sortując dane w tabeli.**

Przed rozpoczęciem połączenia z serwerami baz danych MySQL należało napisać skrypt tworzący bazę danych wraz z tabelą o temacie podanym na zajęciach. Nasz temat to tabela z państwami.

**Kod źródłowy skryptu tworzącego bazę danych oraz tabelę – createDB.php:**

<?php

$server = 'localhost';

$user = 'root';

$pass = 'admin';

$connect = mysql\_connect($server, $user, $pass);

mysql\_query("DROP DATABASE IF EXISTS 313b",$connect);

if(!$connect)

{

die('Error: ' . mysql\_error());

}

if (mysql\_query('CREATE DATABASE 313b', $connect))

{

echo '<strong>Stworzono baze:</strong> 313b';

}

else

{

echo 'Error: ' . mysql\_error();

}

mysql\_select\_db("313b",$connect);

$createTable = " CREATE TABLE IF NOT EXISTS `country` (

`id` int(10) NOT NULL AUTO\_INCREMENT,

`name` varchar(200) NOT NULL,

`capital` varchar(200) NOT NULL,

PRIMARY KEY (`id`)

)";

if(mysql\_query($createTable,$connect))

{

echo '<br><strong>Stworzono tabele</strong>: country';

}

else

{

echo 'Error: ' . mysql\_error();

}

?>

<br><a href="index.php"> POWRÓT </a>

**Wynik zadziałania:**



**Opis:**

Na początku należało połączyć się z serwerem MySQL podając nazwę hosta, login oraz wymagane hasło. Następnie użyto procedury mysql\_connect, która nawiązała połączenie z serwerem. Następnym etapem było wykonanie zapytania tworzącego bazę danych o nazwie 313b. W przypadku sukcesu wywołania zostanie wypisany komunikat o treści „Stworzono bazę: 313b”. Następnie nawiązaliśmy połączenie z nowo utworzoną bazą danych za pomocą funkcji mysql\_select\_db. Dalej napisaliśmy zapytanie tworzące tabelę o nazwie country z trzema kolumnami: id, name, capital. Pierwsze pole jest auto-inkrementowane rozpoczynając od 1 oraz pełni ono funkcję klucza głównego naszej tabeli. Dwie pozostałe kolumny określone są typem VARCHAR(200) oraz występuje przy nich klauzula NOT NULL. Przy kluczu głównym klauzula NOT NULL jest tworzona automatycznie. W przypadku sukcesu wywołania zapytania zostanie wypisany komunikat „Stworzono tabelę: country”. W przypadku niepowodzenia zostanie wypisany komunikat błędu.

Następnie napisaliśmy skrypt, który umożliwia nam łączenie się z utworzoną bazą w zadaniu 2.

**Kod źródłowy skryptu łączenia z bazą danych – config.php:**

<?php

$host = 'localhost';

$user = 'root';

$pass = 'admin';

$db = '313b';

$conn = mysql\_connect($host, $user, $pass);

if (!$conn)

die('Error: ' . mysql\_error());

$db = mysql\_select\_db($db, $conn);

if (!$db)

die('Error: ' . mysql\_error());

session\_start();

?>

**Opis:**

Celem połączenia się z serwerem MySQL użyliśmy funkcji mysql\_connect podając w parametrach wywołania nazwę hosta, użytkownika oraz hasło. Następnie nawiązujemy połączenie z bazą danych 313b za pomocą funkcji mysql\_select\_db. W przypadku błędu zostanie zwrócony błąd.

Następnym etapem było stworzenie pliku index.php, który umożliwia prostsze uruchomienie zadań wykonanych na zajęciach.

**Kod źródłowy – index.php:**

<?php

include('config.php');

$z = $\_GET['zad'];

switch ($z) {

case 1:

include('createDB.php');

break;

case 2:

include('z2.php');

break;

default:

echo '<h2><center>PAI - laboratorium 4 B.Osak T.Pasternak 3ID13B<center></h2><hr>';

echo '<h3><center>Należy włączyć obsługę ciasteczek w przeglądarce.<center></h3><hr>';

echo '<a href="?zad=1"> TWORZENIE BAZY ORAZ TABELI </a><br><hr>';

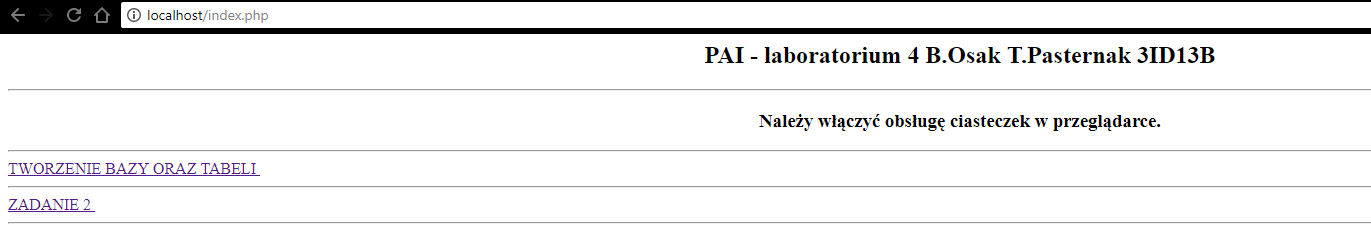
echo '<a href="z2.php"> ZADANIE 2 </a><br><hr>';

break;

}

?>

**Wynik zadziałania:**

  
  
Po przygotowaniu wszystkich plików przystąpiliśmy do wykonywania poszczególnych podpunktów zadania.

**Ad a). Wyświetlanie danych z tabeli.**

**Kod źródłowy – fragment kodu odpowiadający za wyświetlanie danych z tabeli – z2.php:**

<?php

**…**

if(!$\_GET)

{

$query = "SELECT \* FROM country";

$cookieValue = "asc";

setcookie($cookieName,$cookieValue);

}

else

{

$orderSort = $\_GET["order"];

$sort = $\_GET['sort'];

$cookieValue = $\_COOKIE[$cookieName];

if($orderSort == $cookieValue)

{

$query = "SELECT \* FROM country ORDER BY " .$sort. " " .$orderDESC;

$cookieValue = $orderDESC;

setcookie($cookieName,$cookieValue);

}

else

{

$query = "SELECT \* FROM country ORDER BY " .$sort. " " .$orderASC;

$cookieValue = $orderASC;

setcookie($cookieName,$cookieValue);

}

}

$resultQuery = mysql\_query($query);

echo

'<form action="?" method="post">

<input type="hidden" name="delete" value="delete">

<center>

<table border="1" style="border-collapse: collapse;font-family: Arial; font-size:14px;">

<tr>

<th> <a href="?sort=id&order=asc">ID</a> </th>

<th> <a href="?sort=name&order=asc">PAŃSTWO</a> </th>

<th> <a href="?sort=capital&order=asc">STOLICA</a> </th>

<th> <input type="submit" name="edit" value="EDYTUJ"> </th>

<th> <input type="submit" name="delete" value="USUŃ"> </th></tr>';

while($row=mysql\_fetch\_row($resultQuery))

{

echo '<tr><td>' .$row[0]. '</td>

<td>' .$row[1]. '</td>

<td>' .$row[2]. '</td>

<td> <input type="radio" name="idEditRow" value=' .$row[0]. '></td>

<td> <input type="checkbox" name="delete[]" value=' .$row[0]. '> </td>

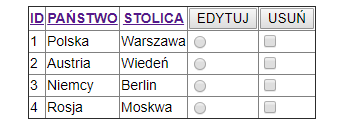
</tr>';

}

echo "</table></form><hr>";

?>

**Wynik zadziałania:**



**Opis:**

Przedstawiony powyżej kod zawiera również elementy, które odnoszą się do sortowania tabeli. Elementami, które odnoszą się do podstawowego wyświetlania tabeli jest wykorzystanie prostego zapytania SELECT \* FROM country. Jest ono wykonywane za pomocą funkcji mysql\_query(). Wynik zapytania przechowywany jest w zmiennej resultQuery. Celem zbudowania tabeli korzystamy z pętli while, która kręci się po zawartości resultQuery pobierając kolejne wiersze tabeli za pomocą funkcji mysql\_fetch\_row. Działanie tego mechanizmu można porównać do kursorów w bazach danych Oracle. W pętli w każdej iteracji wykonywany jest kod HTML tworzący kolejne kolumny z wierszami wraz z radio oraz checkboxami w celu wykorzystania ich w następnych punktach zadania. Pętla iteruje do momentu, aż przejdzie po wszystkich wartościach zapisanych w kursorze.

**Ad b). Wstawianie danych do tabel.**

**Kod źródłowy – fragment kodu odpowiadający za wstawianie danych do tabeli – z2.php:**

<?php

**…**

if(**isset(**$\_POST["add"])**) //dopisano w domu isset (celem pozbycia się note)**

{

if(!empty($name) && !empty($capital))

{

mysql\_query("INSERT INTO country SET name = '" .$name. "', capital='" .$capital. "'");

}

header("Location: ?");

}

**…**

?>

…

<?php

…

$result = mysql\_result(mysql\_query("SELECT MAX(id) as id FROM country"),0) + 1;

echo

'<p style="font-family: Arial; font-size: 14px;"><strong>Dodawanie:</strong></p>

<form action="?" method="post">

<input style="max-width: 30px;" type="text" name="id" placeholder = "' .$result. '" readonly>

<input type="text" name="name" placeholder="Państwo">

<input type="text" name="capital" placeholder="Stolica">

<input type="submit" name="add" value="Dodaj">

</form></center>';

…

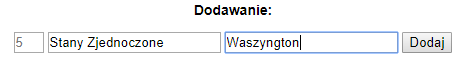
?>

**Wynik zadziałania:**

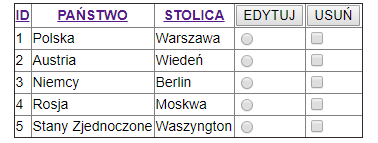
* Formularz do wpisywania danych (pusty):



* Formularz do wpisywania danych (wypełniony):



* Zaktualizowana tabela:



**Opis:**

W celu wpisania danych tworzony jest formularz. Pierwsze pole formularza generowane jest automatycznie poprzez zapytanie zwracające maksymalne id znajdujące się w tabeli. Do tej wartości dodawana jest wartość 1. Pole to nie podlega edycji przez użytkownika, informuje go tylko o tym na jakiej pozycji zostanie dodany nowy wiersz danych w tabeli bazy danych. Dwa następne pola to zwykłe pola tekstowe służące do wpisywania nazwy państwa oraz jego stolicy. Po kliknięciu w przycisk Dodaj zostaje wysyłana wartość „add”. Wartość „add” zostaje przechwycona przez $\_POST[„add”] i zostaje uruchomiony blok dodawania wiersza. W bloku znajduje się funkcja mysql\_query z prostym zapytaniem typu INSERT INTO. Jednakże przed jego wykonaniem sprawdzana jest zawartość pola name (nazwa państwa) oraz capital (nazwa stolicy). Jeśli nie jest puste to zapytanie zostanie wykonane i tabela zarówno na serwerze jak i tabela HTML zostaje zaktualizowana o nowy wiersz.

**Ad c). Poprawianie danych w tabeli.**

**Kod źródłowy – fragment kodu odpowiadający za modyfikację danych do tabeli – z2.php:**

<?php

if(**isset(**$\_POST["modify"])) **//dopisano w domu isset (celem pozbycia się note)**

{

$idEditRow = htmlspecialchars(trim($\_POST["idEditRow"]));

if(!empty($name) && !empty($capital){

mysql\_query("UPDATE country SET name = '" .$name. "', capital='" .$capital. "' WHERE id = '" .$idEditRow. "'");

}

header("Location: ?");

}

?>

…

<?php

if(**isset(**$\_POST["edit"])) { **//dopisano w domu isset (celem pozbycia się note)**

$idEditRow = $\_POST["idEditRow"];

if($idEditRow >= 1) {

echo

'<center><hr>

<p style="font-family: Arial; font-size: 14px;"><strong>Modyfikacja:</strong></p>

<form action="z2.php" method="post">

<input style="max-width: 30px;" type="text" name="id" placeholder = "' .$idEditRow. '" readonly>

<input type="text" name="name" placeholder="Państwo">

<input type="text" name="capital" placeholder="Stolica">

<input type="hidden" name="idEditRow" value="' .$idEditRow. '">

<input type="submit" name="modify" value="Edytuj"><hr style="margin-top: 16px;"></center>';

}

}

…

?>

**Wynik zadziałania:**

* Zaznaczenie wiersza do edycji:



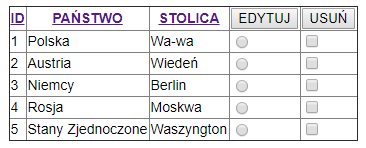
* Formularz do edycji:



* Formularz uzupełniony:



* Zaktualizowana tabela:



**Opis:**

Celem wykonania operacji modyfikacji danego wiersza wyświetlanej tabeli należy zaznaczyć przycisk radio i kliknąć przycisk EDYTUJ. Po zaznaczeniu pojawi się na dole strony formularz do uzupełnienia. Przycisk radio wysyła wartość „idEditRow”, czyli id aktualnie zaznaczonego wiersza. Przycisk EDYTUJ wysyła tekst „edit”, która jest przejmowana przez blok z warunkiem $\_POST[„edit”]. W bloku tym do zmiennej idEditRow przypisana jest wartość odebrana przez $\_POST od przycisku radio, czyli zapisywane jest tam id modyfikowanego wiersza. Następnie tworzy jest analogiczny formularz do formularza służącego do dodawania, który został opisany w poprzednim punkcie. Po naciśnięciu przycisku Edytuj, który towarzyszy bezpośrednio formularzowi modyfikacji zostaje wysłana wartość „modify”. W bloku z warunkiem $\_POST[„modify”] dane ulegają walidacji i wykonywane jest zapytanie umieszczone w funkcji mysql\_query() typu UPDATE.

Po wykonaniu się operacji dane w tabeli są zaktualizowane.

**Ad d). Usuwanie danych z tabeli.**

**Kod źródłowy – fragment kodu odpowiadający za usuwanie danych z tabeli – z2.php:**

<?php

…

if(**isset(**$\_POST["delete"])) **//dopisano w domu isset (celem pozbycia się note)**

{

$deleteRow = $\_POST["delete"];

**if(is\_array($deleteRow)) //dopisano w domu is\_array (celem pozbycia się warninga)**

{

foreach($deleteRow as $idDeleteRow)

{

mysql\_query("DELETE FROM country WHERE id = '" .$idDeleteRow. "'");

header('Location: ?');

}

}

$result = mysql\_result(mysql\_query("SELECT COUNT(\*) FROM country"));

if($result == 0)

{

mysql\_query("ALTER TABLE country AUTO\_INCREMENT = 1");

}

}

?>

…

<?php

…

echo

'<form action="?" method="post">

<input type="hidden" name="delete" value="delete">

<center>

<table border="1" style="border-collapse: collapse;font-family: Arial; font-size:14px;">

<tr>

<th> <a href="?sort=id&order=asc">ID</a> </th>

<th> <a href="?sort=name&order=asc">PAŃSTWO</a> </th>

<th> <a href="?sort=capital&order=asc">STOLICA</a> </th>

<th> <input type="submit" name="edit" value="EDYTUJ"> </th>

<th> <input type="submit" name="delete" value="USUŃ"> </th>

</tr>';

…

while($row=mysql\_fetch\_row($resultQuery))

{

echo '<tr><td>' .$row[0]. '</td>

<td>' .$row[1]. '</td>

<td>' .$row[2]. '</td>

<td> <input type="radio" name="idEditRow" value=' .$row[0]. '></td>

<td> <input type="checkbox" name="delete[]" value=' .$row[0]. '> </td>

</tr>';

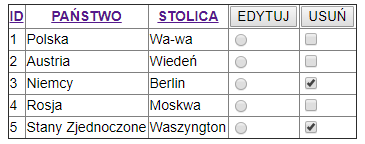
}

echo "</table></form><hr>";

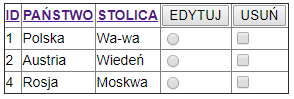
?>

**Wynik zadziałania:**

* Zaznaczenie wiersza do usunięcia:



* Zaktualizowana tabela:



**Opis:**

Celem wykonania operacji usuwania należy skorzystać z pola checkbox i zaznaczyć wiersze, które chcemy usunąć. Następnie należy kliknąć przycisk USUŃ i tabela zostanie zaktualizowana, ponieważ wiersze zostaną usunięte. Zaznaczając wiersze do usunięcia ich id zapisywane są tablicy delete[]. Przyciśnięcie przycisku USUŃ wysyła wartość „delete”. Jest ona wysyłana i odbierana przez blok z warunkiem $\_POST[„delete”], w którym to następuje odczyt tablicy delete. W pętli foreach iterujemy po całej zawartości tej tablicy i wykonywane są operacje DELETE z id pobieranym z tablicy. W przypadku, gdy usuniemy wszystkie rekordy auto inkrementacja jest resetowana do wartości 1. Jest to zrealizowane za pomocą zapytania SELECT COUNT, które zwraca nam ilość wierszy obecnie przechowywanych w tabeli. Jeśli zapytanie zwróci wartość 0 to auto inkrementacja jest resetowana. Po wykonaniu operacji widok tabeli jest zaktualizowany.

**Ad e). Sortowanie danych w tabeli.**

**Kod źródłowy – fragment kodu odpowiadający za sortowanie danych w tabeli – z2.php:**

<?php

$orderASC = "asc";

$orderDESC = "desc";

$cookieName = "sortTableColumn";

if(!$\_GET) {

$query = "SELECT \* FROM country";

$cookieValue = "asc";

setcookie($cookieName,$cookieValue);

}

else

{

$orderSort = $\_GET["order"];

$sort = $\_GET['sort'];

$cookieValue = $\_COOKIE[$cookieName];

if($orderSort == $cookieValue) {

$query = "SELECT \* FROM country ORDER BY " .$sort. " " .$orderDESC;

$cookieValue = $orderDESC;

setcookie($cookieName,$cookieValue);

}

else {

$query = "SELECT \* FROM country ORDER BY " .$sort. " " .$orderASC;

$cookieValue = $orderASC;

setcookie($cookieName,$cookieValue);

}

}

$resultQuery = mysql\_query($query);

echo

'<form action="?" method="post">

<input type="hidden" name="delete" value="delete">

<center>

<table border="1" style="border-collapse: collapse;font-family: Arial; font-size:14px;">

<tr>

<th> <a href="?sort=id&order=asc">ID</a> </th>

<th> <a href="?sort=name&order=asc">PAŃSTWO</a> </th>

<th> <a href="?sort=capital&order=asc">STOLICA</a> </th>

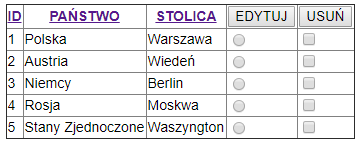
<th> <input type="submit" name="edit" value="EDYTUJ"> </th>

<th> <input type="submit" name="delete" value="USUŃ"> </th>

</tr>';

**Wyniki zadziałania:**

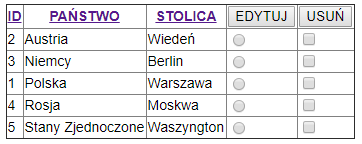
* Sortowanie domyślne:



* Sortowanie po ID malejąco:



* Sortowanie po nazwie państwa rosnąco:



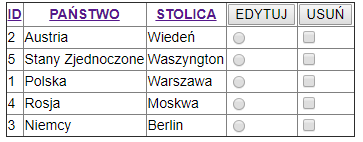
* Sortowanie po nazwie państwa malejąco:



* Sortowanie po nazwie stolicy rosnąco:



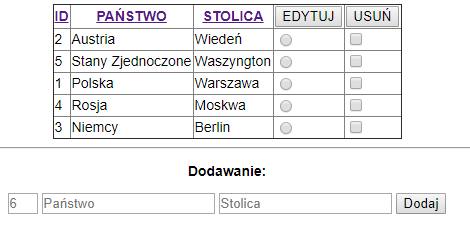
* Sortowanie po nazwie stolicy malejąco:



**Opis:**

Sortowanie odbywa się poprzez klikanie na nazwę kolumny. Działa dwustronnie – malejąco oraz rosnąco. Domyślnym sortowaniem jest sortowanie po ID malejąco. Przy podstawowym wypisaniu tabeli do zmiennej cookieValue przypisywany jest kierunek sortowania – „asc” oraz ustanowione jest ciasteczko przechowujące tą wartość. Następnie zgodnie z wybraną kolumną jest odczytywana wartość z ciasteczka, która określa kierunek sortowania. Jeśli raz naciśniemy np. państwo to będzie posortowane rosnąco i do ciasteczka zostanie zapisana wartość „desc”, drugi raz malejąco, więc do ciasteczka zostanie przyspisana wartość „asc”. Zasada tyczy się wszystkich kolumn. Przy każdym kliknięciu tabela jest na bieżąco aktualizowana.

**Ogólny wygląd strony:**



**Ogólny wygląd strony po wybraniu wiersza do edycji:**

