

**KRÓTKI KURS TWORZENIA
APLIKACJI BAZODANOWYCH
Z WYKORZYSTANIEM MYSQL I PHP**

OPRACOWANIE
mgr inż. Tomasz Szczepanik
www.szczepaniktomasz.pl

PROJEKT SEMESTRALNY POWINIEN ZAWIERAĆ:

Tabele (minimum 3, np. UZYTKOWNICY, DANE, MIEJSCOWOSCI)

Rejestracja

Logowanie: przynajmniej dwa typy użytkowników (administrator, zwykły użytkownik), sesja

Panel sterowania (linki do funkcji użytkownika zwykłego i admina)

Dokumentacja

Zmiana hasła

Przeglądanie danych użytkownika

Edycja danych użytkownika (oprócz loginu)

Przeglądanie danych

Dodawanie danych

Modyfikacja danych

Usuwanie danych

Wyszukiwanie danych

Rezerwacja produktu przez użytkownika

OCENA:

Praca na lekcji (aktywność)

Odpowiedź z teorii i wykorzystania jej w praktyce

Kolejny etap projektu

Omówienie projektu z elementami tworzenia projektu

LEKCJA

Tworzenie umowy:

- cel bazy

- funkcje użytkownika zwykłego, administratora

- tabele (jakie, ilość pól)

LEKCJA lub w domu

Zakładanie konta na serwerze zewnętrznym lub praca na z wykorzystaniem pakietu XAMPP

LEKCJA

Przypomnienie formularzy:

Formularz prosty: http://uczsieucz.cba.pl/strony_internetowe/formularz/index.html

Plik formularz.php

```
<?
```

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

```
echo"Twój login to: $login";
```

```
echo'Twój login to: $login';
```

```
?>
```

LEKCJA

Własne wykonanie formularz walutowego:

formularz walutowy: http://uczsieucz.cba.pl/strony_internetowe/waluty/index.html

LEKCJA

Omówienie tabel w phpmyAdmin

phpMyAdmin - system zarządzania bazami danych MySQL napisany w języku PHP, oparty na licencji GNU, umożliwia wykonywanie operacji z poziomu przeglądarki internetowej w graficznym środowisku.

Najpopularniejsze typy pól:

- **char(x)** - ciąg znaków o maksymalnej długości x, gdzie x nie może być większy od 255
- **blob** - binarny ciąg znaków o długości ograniczonej przez pamięć twojego serwera
- **text** - tekstowy ciąg znaków o długości ograniczonej przez pamięć twojego serwera
- **integer** - liczba całkowita z przedziału -2147483647 do 2147483647
- **data** - data w formacie określonym przez ustawienia serwera
- **year** - rok, jeśli zostanie podany zły, jego wartość zmieni się w 0000
- **decimal(x, y)** - liczba dziesiętna, gdzie x oznacza maksymalną liczbę cyfr, a y maksymalną liczbę cyfr po przecinku

MySQL - DBMS opierający się na języku zapytań **SQL** (ang. **Structured Query Language**), umożliwiającemu zarządzanie bazą danych. MySQL tworzony jest przez szwedzką firmę MySQL AB od 1995r., kupioną 16 stycznia 2008 roku przez Sun Microsystems.

Wygląd PHPMyAdmin:

STRONA GŁÓWNA: nazwa serwera, nazwa bazy danych, metoda porównywania napisów

BAZA DANYCH:

tworzenie tabeli: OPERACJE-Utwórz nową tabelę w bazie danych

Operacje wykonywane za pomocą MySQL:

Funkcje biblioteki mysqli	Zwracana wartość
<code>mysqli_connect(serwer, użytkownik, hasło, nazwa_bazy)</code>	id połączenia lub FALSE, gdy niepowodzenie
<code>mysqli_select_db(id_polaczenia, nazwa_bazy)</code>	TRUE/FALSE w zależności od stanu operacji
<code>mysqli_error(id_polaczenia)</code>	Tekst komunikatu błędu
<code>mysqli_close(id_polaczenia)</code>	TRUE/FALSE w zależności od stanu operacji
<code>mysqli_query(id_polaczenia, zapytanie)</code>	Wynik zapytania
<code>mysqli_fetch_row(wynik_zapytania)</code>	Tablica numeryczna odpowiadająca wierszowi zapytania
<code>mysqli_fetch_array(wynik_zapytania)</code>	Tablica zawierająca kolejny wiersz z podanych w wyniku zapytania lub FALSE, jeżeli nie ma więcej wierszy w wyniku zapytania
<code>mysqli_num_rows(wynik_zapytania)</code>	Liczba wierszy w podanym zapytaniu
<code>mysqli_num_fields(wynik_zapytania)</code>	Liczba kolumn w podanym zapytaniu

tworzenie tabeli na przykładzie tabeli UZYTKOWNICY:

NAZWY TABEL I PÓL ZAPISYWANE SĄ W ZNAKACH GRAWIS (NA KLAWIATURZE ZE ZNAKIEM TYLDY) INNE WARTOŚCI SĄ PISANE W APOSTROFACH

```
CREATE TABLE `UZYTKOWNICY` (`ID_UZYTKOWNICY` INT NOT NULL AUTO_INCREMENT PRIMARY KEY, `LOGIN` TEXT NOT NULL, `HASLO` TEXT NOT NULL, `NAZWISKO` VARCHAR( 30 ) NULL, `IMIE` VARCHAR( 30 ) NULL, `KOD` INT NULL, `MIEJSCOWOSC` VARCHAR( 30 ) DEFAULT 'Gorlice', `NR` VARCHAR( 10 ) NULL, `TELEFON` INT NOT NULL);
```

usuwanie tabel:

```
DROP TABLE `UZYTKOWNICY`;
```

dodawanie rekordu:

```
INSERT INTO `UZYTKOWNICY` (`ID_UZYTKOWNICY`, `LOGIN`, `HASLO`, `NAZWISKO`, `IMIE`, `KOD`, `MIEJSCOWOSC`, `NR`, `TELEFON`) VALUES (
```

```
NULL , 'belfer', 'hasełeczko', 'Mickiewicz', 'Adam', 38-300, 'Gorlice',  
'3F', 700600500);
```

modyfikacja rekordu:

```
UPDATE `UZYTKOWNICY` SET `HASLO` = 'innehaslo', `NAZWISKO` = 'Mickeymouse'  
WHERE `UZYTKOWNICY`.`ID_UZYTKOWNICY` =1;
```

usuwanie rekordu:

```
DELETE FROM `UZYTKOWNICY` WHERE `UZYTKOWNICY`.`ID_UZYTKOWNICY` =1;
```

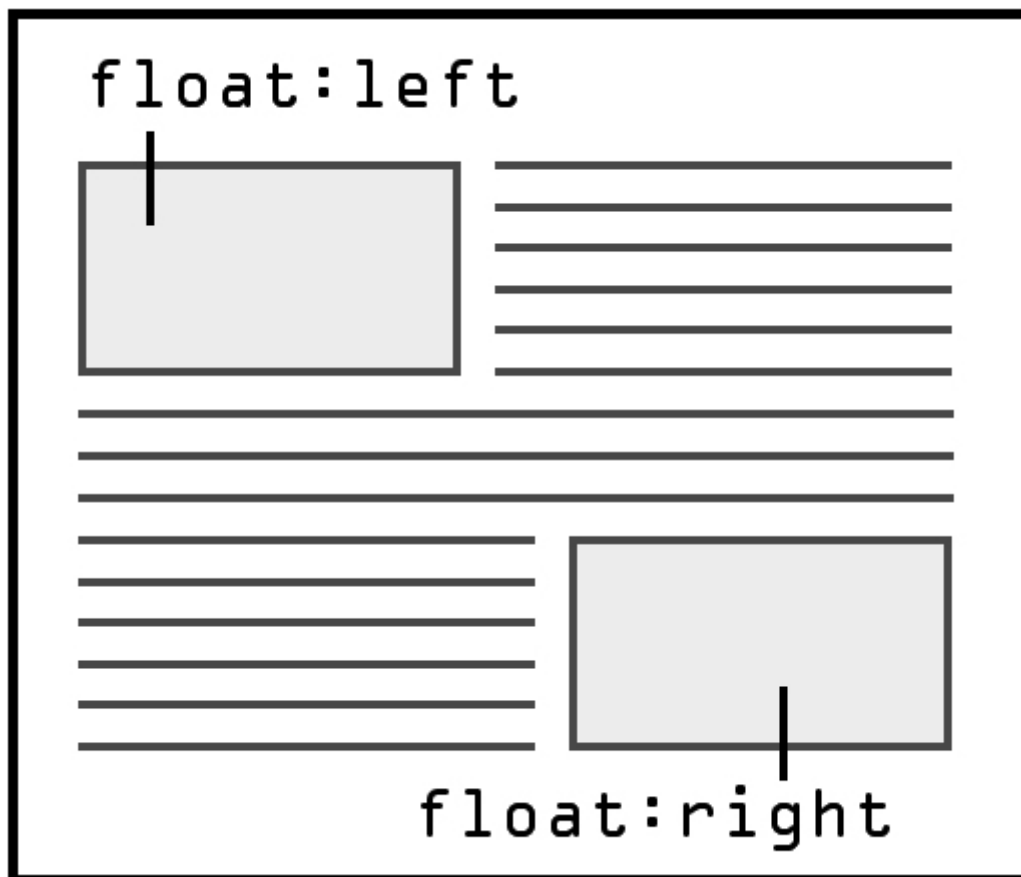
przeglądanie rekordu:

```
SELECT * FROM `UZYTKOWNICY`; //przeglądanie wszystkich danych z tabeli  
SELECT * FROM `UZYTKOWNICY` WHERE `UZYTKOWNICY`.`IMIE` ='Julita' ORDER BY  
`NAZWISKO` DESC; //przeglądanie wybranych rekordów zawierających imię  
Julita w kolejności alfabetycznej według nazwiska
```

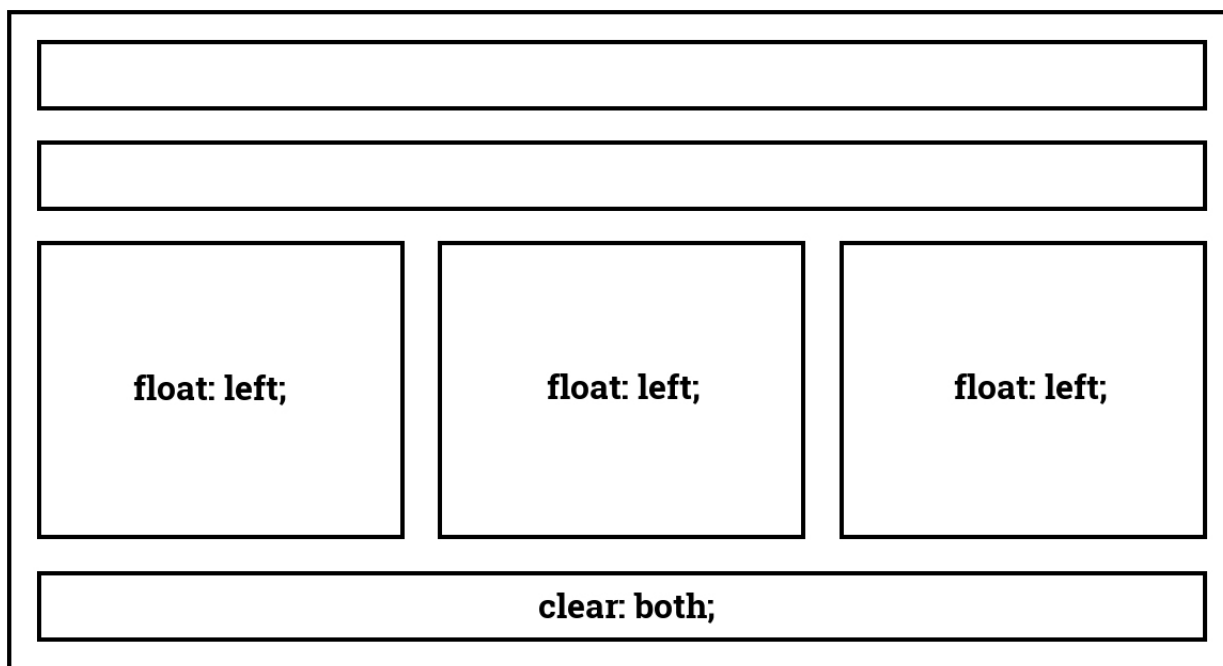
Wprowadzanie aktualnej daty: CURRENT_DATE lub \$dzis= date("Y-m-d"); pole data musi być typu DATE

LEKCJA

Układanie bloków: realizujemy za pomocą float (opływanie elementu) :



W praktyce metoda ta polega na tym, iż wszystkie bloki, które mają być ułożone obok siebie, będą posiadały w swoich klasach CSS dodatkowy atrybut **float:left**. A następny blok, który **już nie ma się układać poziomo**, ma posiadać zapis czyszczący opływanie: **clear:both**;



Atrybut **float:left** powoduje opływanie, układanie obok siebie każdego kolejnego bloku. Aby kolejny blok mógł pojawić się poniżej należy wyłączyć opływanie: `clear:both`.

LEKCJA

Tworzenie rejestracji

Katalog rejestracja a w nim:

plik: `index.html` (z formularzem do rejestracji danych z podwójnym wprowadzeniem hasła)

plik: `rejestracja.php` (z funkcjami rejestracji, sprawdzanie błędów rejestracji i wprowadzania danych do bazy ze sprawdzeniem czy login i hasło zostało wprowadzone i czy hasła są takie same)

Przycisk powrót w razie błędnych danych:

```
<a href=javascript:history.go(-1);>powrót</a>
```

LEKCJA

Łączenie z bazą:

Utworzenie pliku „`funkcje.php`” logowania do bazy, który będzie dołączany do innych plików w celu wykonywania operacji na bazie:

```
<?
/*funkcja łączenia się z baza danych powinna znaleźć się na początku
każdego pliku, w którym wykonuje się zapytania do bazy*/
function dbConnect(){
/* połączenie z baza danych */
global $połączenie;
/*argumenty pisz w apostrofach*/
if ($połączenie=mysqli_connect('localhost','root','','wedkowanie')){
}
else{
/* Pobierz informacje o błędzie połączenia.*/
echo "Nieudane połączenie z baza danych MySQL.<br />";
}
}
?>
```

LEKCJA

Dokończenie rejestracji

ŁĄCZENIE Z BAZĄ I DODANIE REKORDU:

```
include("../funkcje.php"); /* dołączanie zawartości pliku funkcje.php przez podanie ścieżki dostępu*/
```

```
dbConnect(); //wywołanie funkcji połączenia z bazą danych
```

SPRAWDZENIE CZY LOGIN JEST WOLNY

```
$zapytanie1="SELECT * FROM `UZYTKOWNICY` WHERE `login` ='$login'";  
$ile=mysql_num_rows(mysql_query($zapytanie1));  
echo"$ile";  
if($ile==0)  
{  
/*tutaj wpisz zapytanie dodające rekord do bazy*/  
}  
else  
{  
echo"login zajęty";  
}
```

wykonaj w phpMyAdmin dodanie rekordu do tabeli UZYTKOWNICY i skopiuj kod SQL pod funkcją dbConnect(), a następnie uzupełnij kod funkcjami wykonania poleceń SQL:

```
$zapytanie="tu wklej polecenie SQL z phpMyAdmina ";//przypisanie zmiennej $zapytanie polecenia SQL
```

```
mysql_query($zapytanie); //przetwarzanie zapytania mysql podanego jako argument
```

```
if(mysql_query($zapytanie))
```

```
echo"wykonano poprawnie";
```

```
else
```

```
echo"błąd wykonania zapytania";
```

```
mysql_close($c); //zamyka połączenie z bazą danych
```

- przesyłanie hasła do bazy w sposób zaszyfrowany funkcją md5(), lub password()
- sprawdzenie czy login nie jest już zajęty

```
$zapytanie="INSERT ..... Values (null, password($haslo)) "
```

LEKCJA: LOGOWANIE

plik logowanie.php

```
<?
```

```
include("../funkcje.php");
```

```
$login=$_POST['login'];
```

```
$haslo=$_POST['haslo'];
```

```
if($login==NULL || $haslo==NULL)
```

```
echo"nie podałeś loginu lub hasła<br>
```

```
<a href=javascript:history.go(-1)>powrót</a>";
```

```
else{
```

```
dbConnect();
```

```
$zapytanie="SELECT * FROM uzytkownicy WHERE `login`='$login' AND
```

```
`haslo`=password('$haslo');";
```

```
$ile=mysql_num_rows(mysql_query($zapytanie));
```

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

```
echo "zostałeś zalogowany";
```

```
header("location:../po_zalogowaniu.php");
```

```

    }
    else{
        echo"podałeś nieprawidłowe dane<br>
        <a href=javascript:history.go(-1)>powrót</a>";
    }
        mysql_close($c);
    }
?>

```

LEKCJA: SESJE

plik logowanie.php z dodaniem obsługi sesji

```

<?
session_start();
include("../funkcje.php");
$login=$_POST['login'];
$haslo=$_POST['haslo'];
if($login==NULL || $haslo==NULL)
    echo"nie podałeś loginu lub hasła<br>
    <a href=javascript:history.go(-1)>powrót</a>";
else{
    dbConnect();
    $zapytanie="SELECT * FROM uzytkownicy WHERE `login`=' $login' AND
'haslo'=password(' $haslo')";
    $ile=mysql_num_rows(mysql_query($zapytanie));
    if($ile==1){
        echo "zostałeś zalogowany";
        $_SESSION['dopuszczony']=$login; //zmienna sesyjna
header("location:../po_zalogowaniu.php");
    }
    else{
        echo"podałeś nieprawidłowe dane<br>
        <a href=javascript:history.go(-1)>powrót</a>";
    }
        mysql_close($c);
    }
?>

```

KAŻDY INNY PLIK DOSTĘPNY PO ZALOGOWANIU MUSI ZAWIERAĆ:

```

<?php
session_start();
if(isset($_SESSION['dopuszczony'])){
    /* tu wpisujemy linie kodu związane z przekazywaniem danych po
zalogowaniu*/
$zalogowany=$_SESSION['dopuszczony'];
echo"Witaj $zalogowany";
echo"<a href=\"wyloguj.php\">wyloguj się</a>";
}
else
    echo"Brak uprawnień do przeglądania zawartości, musisz się
zalogować";
?>

```

PLIK NISZCZĄCY SESJĘ wyloguj.php

Do tego pliku muszą być linki na każdej stronie dostępnej po zalogowaniu

```

<?php
session_start();

```

```
unset($_SESSION['dopuszczony']); /*usuwanie zmiennej sesyjnej
usuwanie większej ilości zmiennych sesyjnych naraz
$_SESSION = array();*/
session_destroy(); //niszczenie sesji
echo"wylogowano użytkownika";
?>
```

LEKCJA: GRUPY UŻYTKOWNIKÓW

Rozróżnimy dwie grupy użytkowników:

admin – użytkownik z pełnymi prawami: zmiana hasła i edycja swoich danych, dodawanie, edycja, usuwanie użytkowników, dodawanie, edycja, usuwanie innych danych

plik logowanie.php z dodaniem panelu sterowania:

```
<?
session_start();
include("../funkcje.php");
$login=$_POST['login'];
$haslo=$_POST['haslo'];
if($login==NULL || $haslo==NULL)
    echo"nie podałeś loginu lub hasła"
else{
    dbConnect();
    $zapytanie="SELECT * FROM uzytkownicy";
    $zalogowany=0;
    $wynik=mysql_query($zapytanie); //przetwarzanie zapytania mysql
//podanego jako argument zapisany w zmiennej $query
    $pobrzanych=mysql_num_rows($wynik); //funkcja zliczająca ilość
//rekordów
    while($rekord = mysql_fetch_array($wynik)) { //przejdźcie po rekordach
//tabeli
        $login_baza=$rekord['LOGIN']; //przypisanie pod zmienną
$login_baza wartości pola LOGIN danego rekordu
        $haslo_baza=$rekord['HASLO'];
        if($login=="admin" && $haslo_baza==$haslo){
            $zalogowany=1;
            $_SESSION['dopuszczony']=$login; //zmienna sesyjna
            echo "zostałeś zalogowany jako admin
            <a href=\"po_zalogowaniu_admin.php\">przejdź dalej</a>";
            header("location:../po_zalogowaniu_admin.php");
            break;
        }
        else if($login_baza==$login && $haslo_baza==$haslo){
            $zalogowany=2;
            echo "zostałeś zalogowany jako $login";
            <a href=\"po_zalogowaniu.php\">przejdź dalej</a>
            header("location:../po_zalogowaniu.php");
            $_SESSION['dopuszczony']=$login; //zmienna sesyjna
            break;
        }
        mysql_close($c);
    }

if($zalogowany==0){
    echo"podałeś nieprawidłowe dane";
}
?>
```

LEKCJA: TWORZENIE PANELU STEROWANIA

Panel sterowania ma być dostępny dopiero po zalogowaniu. Najlepiej jest utworzyć sobie szablon oparty na ramkach lub tabelkach:

zawartość pliku naglowek.html	
zawartość pliku menu.html	zawartość tego pliku, który jest dostępny po wybraniu opcji z menu

**PRZYKŁAD SZABLONU OPARTEGO NA TABELKACH:
KAŻDY PLIK DOSTĘPNY PO ZALOGOWANIU MUSI MIEĆ STRUKTURĘ PLIKU
po_zalogowaniu.php:**

```
<?
session_start();
if(isset($_SESSION['dopuszczony'])) {
$login=$_SESSION['dopuszczony'];
?>
<table height="100%" width="100%">
  <tr>
    <td width="100%" height="10%" colspan="2">
      <?
      include("naglowek.html");
      ?>
    </td>
  </tr>
  <tr>
    <td width="100%" height="10%">
      <?
      if($login=="admin")
      include("menu_admin.html");
      else
      include("menu.html");
      ?>
    </td>
    <td width="100%" height="10%">
      <?
      include("strona_glowna.html");
      ?>
    </td>
  </tr>
</table>
<?
}
else
echo"Brak uprawnień do przeglądania zawartości, musisz się zalogować";
?>
```

LEKCJA

Wyświetlanie, wyszukiwanie, dodawanie, edycja, usuwanie danych.

Najpierw należy utworzyć formularz, w którym będziemy mogli wprowadzać dane, a w nowym pliku wykorzystać:

FUNKCJE ŁĄCZENIA Z BAZĄ:

```
include("../funkcje.php");
dbConnect();
```

WSTAWIANIE:

fragment kodu wstawiającego dane do tabeli:

```
//tutaj wstawiamy wywołanie funkcji łączenia się z bazą danych
$zapytanie = "INSERT INTO `test` ( `ID_TEST` , `LICZBA` , `ZNAKI` ,
`LANCUCH` ) VALUES ( NULL , '7', 'ala nie ma', 'lata osa kolo nosa')";
$wynik = mysql_query($zapytanie);
```

PRZEGLĄDANIE:

fragment kodu wypisującego pola z tabeli:

```
//tutaj wstawiamy wywołanie funkcji łączenia się z bazą danych
$zapytanie = "SELECT * FROM uzytkownicy";
$wynik = mysql_query($zapytanie);
echo '<table>';
while ($rekord = mysql_fetch_row($wynik)) {
echo '<tr><td>'. $rekord[0] .'</td><td>'. $rekord[1] .'</td><td>'.
$rekord[2] .'</td><td>'. $rekord[3] .' </td></tr>';
}
echo '</table>';
```

UWAGA!

Zamiast funkcji `mysql_fetch_row($wynik)` można użyć funkcji `mysql_fetch_array($wynik)`. Wówczas można w odwołaniu się do pola `$rekord[0]` jako argument podać rzeczywistą nazwę pola, np. `$rekord[nazwa_pola]`.

fragment kodu wypisującego pola z przyciskiem usuwania rekordów i modyfikacji rekordów:

```
//tutaj wstawiamy wywołanie funkcji łączenia się z bazą danych
$zapytanie = "SELECT * FROM test";
$wynik = mysql_query($zapytanie);
echo '<table>';
while ($rekord = mysql_fetch_row($wynik)) {
echo '<tr><td>'. $rekord[0] .'</td><td>'. $rekord[1] .'</td><td>'. $rekord[2]
.'</td><td>'. $rekord[3] .' </td> <td>';
echo '<form method="GET" action="usun.php">
<input type="hidden" name="id" value='. $rekord[0] .'>
<input type="submit" value="usuń"></form></td>

</tr><td>';
echo '<form method="GET" action="edytuj.php"><input type="hidden" name="id"
value='. $rekord[0] .'><input type="submit" value="edytuj"></form></td></tr>';
}
echo '</table>';
```

USUWANIE:

fragment kodu usuwającego rekordy:

```
//tutaj wstawiamy funkcję łączenia się z bazą danych i ją wywołujemy
```

```

$id=$_GET[id];

$zapytanie = " DELETE FROM `test` WHERE `test`.`ID_test`=$id";
$result = mysql_query($zapytanie);
echo "usunięto dane";
echo "<a href=\"przeogladaaj.php\">powrót</a>";

```

EDYCJA:

Aby dane edytować należy przesłać z pliku przeogladaaj.php dane do edycji i wstawić je do pliku edytuj.php w miejsca pól tekstowych w formularzu

Fragment kodu wypisującego pola z przyciskiem usuwania rekordów i modyfikacji rekordów i danymi do modyfikacji:

//tutaj wstawiamy wywołanie funkcji łączenia się z bazą danych

```

$id=$_GET[id];

$zapytanie = "SELECT * FROM `test` WHERE `test`.`ID_test`=$id";
$wynik = mysql_query($zapytanie);
echo '<table>';
while ($rekord = mysql_fetch_row($wynik)) {
echo '<tr><td>';
echo '<form method="GET" action="zmodyfikuj.php">
<input type="hidden" name="id" value='.$rekord[0].'>
<input type="text" name="nazwisko" value='.$rekord[1].'>
<input type="text" name="imie" value='.$rekord[2].'>
<input type="text" name="login" value='.$rekord[3].'>
<input type="submit" value="edytuj"></form></td></tr>';
}
echo '</table>';

```

Plik zmodyfikuj.php

```

<?
//tutaj wstawiamy funkcję łączenia się z bazą danych i ją wywołujemy
$id=$_GET[id];
$imie=$_GET[imie];
$nazwisko=$_GET[nazwisko];
$login=$_GET[login];
$zapytanie = " UPDATE `test` SET `imie` = '$imie', `nazwisko` =
'$nazwisko', `login` = '$login' WHERE `id` = $id";
mysql_query($zapytanie);
echo "zmodyfikowano dane";
echo "<a href=\"przeogladaaj.php\">powrót</a>";

```

WYSZUKIWANIE DANYCH:

Aby wyszukać dane należy uzupełnić formularz w pliku szukaj.php i przesłać dane do pliku wyszukaj.php:

Plik szukaj.php

```

<form method="GET" action="wyszukaj.php">
imie<input type="text" name="imie">
nazwisko<input type="text" name="nazwisko">
login<input type="text" name="login">
<input type="submit" value="wyszukaj"></form>

```

Plik wyszukaj.php

```

<?
//tutaj wstawiamy funkcję łączenia się z bazą danych i ją wywołujemy
$imie=$_GET[imie];

```

```

$nazwisko=$_GET[nazwisko];
$login=$_GET[login];
$zapytanie = "SELECT * FROM `test` WHERE `test`.`nazwisko` ='$nazwisko' AND
`test`.imie='$imie' ORDER BY `nazwisko`, `imie` DESC";
$wynik = mysql_query($zapytanie);
echo '<table>';

while ($rekord = mysql_fetch_assoc($wynik)) {
echo '<tr><td>'.$rekord[0].'</td><td>'. $rekord[1] .'</td><td>'. $rekord[2]
.'</td><td>'. $rekord[3] .' </td><td>';
echo '<form method="GET" action="usun.php">
<input type="hidden" name="id" value=".'$rekord[0].'><input type=submit
value=usun></form></td><td>';
echo '<form method="GET" action="edytuj.php"><input type="hidden" name="id"
value=.'$rekord[0].'><input type="hidden" name="nazwisko"
value=.'$rekord[1].'><input type="hidden" name="imie"
value=.'$rekord[2].'><input type="hidden" name="login"
value=.'$rekord[3].'><input type=submit value=edytuj></form></td></tr>';
}
echo '</table>';
?>

```

LEKCJA: DODAWANIE PLIKÓW DO BAZY

Pliki przechowuje się w bazie w polu typu BLOB (BLOB, MEDIUMBLOB, TINYBLOB i LONGBLOB).

W formularzu należy utworzyć pole wyboru z przyciskiem „Przełączaj” pozwalające na dodanie z dysku pliku do przesłania do bazy:

Plik z formularzem: dodaj.php

```

<FORM ACTION="upload.php" METHOD="POST" ENCTYPE="multipart/form-data">
Zdjęcie: </td><td><INPUT type="file" name="zdjecie">
<input type="submit" name="ok" value="Wyślij zdjęcie do bazy"/>
</FORM>

```

Plik dodający do bazy: upload.php

```

<?
include("../funkcje.php");
$fhandle = fopen($_FILES['zdjecie']['tmp_name'], "r");/*otwarcie pliku do odczytu*/
$content = base64_encode(fread($fhandle, $_FILES['zdjecie']['size']));/*zapisanie zawartości
pliku w zmiennej $content*/
fclose($fhandle);
dbConnect();
$zapytanie = mysql_query("INSERT INTO zdjecia (zdjecie) VALUES ('".$content."");/*
dodanie danych do bazy*/
$adres = "ADRES_STRONY/showimage.php?id=".mysql_insert_id();/*wskazanie adresu
zdjęcia*/
echo "<br/><img src='".$adres.'" />";/* wyświetlenie zdjęcia z pliku*/
?>

```

Funkcja wyświetlająca zdjęcie znajduje się w pliku showimage.php

```

<?
include("../funkcje.php");
dbConnect();
header("Content-type: image/jpg;"); /*plik musi być traktowany przez przeglądarkę jak
zdjęcie*/

```

```

$result = mysql_query("SELECT zdjecie FROM rowery WHERE id=".$_GET['id']);
if(mysql_num_rows($result) != 0)
{
    $row = mysql_fetch_assoc($result);/* wyświetlenie rekordu ze zdjęciem*/
    echo base64_decode($row['zdjecie']);
}
?>

```

LEKCJA

ELEMENTY JAVA SCRIPT NA STRONIE

Utwórz katalog java a w nim plik index.html i wklej do niego poniższy kod:

```

<script language="JavaScript">
document.write("rzut kostką: ");//wyświetla tekst zawarty w
cudzysłowiu
a=Math.random();//losuje dowolną wartość
document.write(a+"<br>");//wyświetla wartość zmiennej
</script>

```

Zamień linię „document write()” na linię poniższą:

```

alert("wylosowano: "+a);//wyświetla w oknie dialogowym tekst

```

Zmodyfikuj linię pogrubioną:

```

<script language="JavaScript">
document.write("rzut kostką: ");//wyświetla tekst zawarty w
cudzysłowiu
a=Math.round(Math.random()*5+1);//losuje wartość z przedziału
1-6 i zaokrągla liczbę do liczby całkowitej
document.write(a+"<br>");//wyświetla wartość zmiennej
</script>

```

Utwórz w katalogu java sześć obrazków z rozszerzeniem *.jpg (każdy plik ma być nazwany: „kosc_NR.jpg”, gdzie NR oznacza ilość oczek na kostce):

```

<script language="JavaScript">
document.write("rzut kostką: ");//wyświetla tekst zawarty w
cudzysłowiu
a=Math.round(Math.random()*5+1);//losuje wartość z przedziału
1-6 i zaokrągla liczbę do liczby całkowitej
document.write("<img src=kosc_"+a+".jpg><br>");//wyświetla
obrazek o nazwie związanej z liczbą oczek
</script>

```

ZADANIE:

Utwórz programik uruchamiany za pomocą przycisku i wyświetlający rzut dwoma kostkami.

LEKCJA

Dokumentacja do programu rajdy

1. Wprowadzenie (cel bazy)
2. Obszar działalności obsługiwany przez system
3. Funkcje (przypadki użycia)
4. Tabele

1. Wprowadzenie (Cel bazy)

Program rajdy powstał, aby usprawnić obsługę rajdów turystycznych zarówno od strony organizatora takich rajdów jak i uczestników. Program realizuje definiowanie tras podczas danego rajdu oraz odcinków – składowych tras.

Po uruchomieniu programu pracę rozpoczynamy od wyboru poziomu: uczestnika lub organizatora. Następnie organizator może definiować trasę w ramach rajdu i poszczególne odcinki, na które składa się trasa. Odcinki zawierają następujące dane: numer odcinka (odcinki muszą być numerowane w kolejności począwszy od numeru 1), początek i koniec odcinka (przy czym początek następnego odcinka musi być tożsamy z końcem poprzedniego), stopień trudności, datę, jego długość (w kilometrach) i czas przejścia odcinka (podany w godzinach). Organizator rajdu może wnieść poprawki do trasy lub przypisać ją innemu rajdowi poprzez „edycję trasy”. Można wydrukować zestawienie rajdów służące do promocji rajdów i zapoznanie się z poszczególnymi rajdami przez przyszłych uczestników. Uczestnik zapisuje się na rajd poprzez wypełnienie formularza przy trasie konkretnego rajdu, na którą zdecyduje się wybrać, podając: nazwisko, imię, płeć, PESEL, adres zamieszkania i numer kontaktowy. Organizator może edytować dane uczestnika, jeśli zajdzie taka potrzeba poprzez znalezienie go wśród uczestników poszczególnych rajdów lub poprzez wpisanie (znalezienie jego danych na liście wyboru) w wyszukiwarce. Organizator może stworzyć sobie bazę noclegów, którą będzie mógł wykorzystać podczas rezerwacji noclegów dla uczestników na szlaku danego rajdu. Istnieje także możliwość sprawdzania przez organizatora, czy dany uczestnik uregulował należności za nocleg, bądź ile jeszcze mu zostało do zapłacenia. Można wydrukować listę noclegów, dzięki czemu popadaniu jej do publicznej wiadomości, uczestnik będzie mógł wybrać (o ile będzie taka możliwość), w jakim lokum noclegowym będzie mógł spędzić noc. Program sprawdza, ilu uczestników można zapisać na nocleg w jednym lokum noclegowym i po zapisaniu uczestnika na nocleg, program zmniejsza liczbę miejsc możliwych do zapisania na nocleg.

Istnieje możliwość rozbudowy programu, aby spełniał więcej funkcji niż dotychczasowa wersja. Można stworzyć między innymi funkcje sprawdzające, czy dany użytkownik nie zapisał się na więcej niż dwie trasy w ramach rajdu w tym samym czasie.

2. Obszar działalności obsługiwany przez system

Należy tu wpisać, co system obsługuje, a co trzeba byłoby robić „ręcznie”, gdyby system nie istniał.

3. Funkcje (przypadki użycia)

Nazwa funkcji	Opis funkcji	Dane wejściowe	Dane wyjściowe
Zmiana hasła	Po wprowadzeniu starego hasła i	Hasło stare z formularza i bazy	Zamienione nowe hasło w bazie danych

	zweryfikowaniu go z aktualnie ustalonym oraz sprawdzeniu czy dwukrotnie powtórzone nowe hasła się zgadzają, hasło jest zmieniane.	danych do porównania; hasła nowe z formularza	
--	---	---	--

4. Tabele:

nazwa bazy danych:

.....

nazwa tabeli: UZYTKOWNICY

Pole	Typ	Metoda porównywania napisów	Atrybuty	Null	Domyślnie	Dodatkowo
ID_uzytkownicy	int(11)			Nie	<i>None</i>	auto_increment
login	text	utf8_polish_ci		Nie	<i>None</i>	
haslo	text	utf8_polish_ci		Nie	<i>None</i>	

ZAPYTANIA SQL:

W DOKUMENTACJI ZAPYTAŃ <http://dev.mysql.com> WYSTĘPUJĄ NASTĘPUJĄCE OZNACZENIA:

{połączenie1 | połączenie2 } – wybór obowiązkowy jednego z poleceń oddzielonych |
[opcja] – wybór opcjonalny

TWORZENIE BAZY DANYCH:

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name  
    [create_specification] ...
```

create_specification:

```
[DEFAULT] CHARACTER SET [=] charset_name  
| [DEFAULT] COLLATE [=] collation_name
```

PRZYKŁAD:

```
CREATE DATABASE IF NOT EXISTS baza_testowa CHARACTER SET=utf8  
COLLATE=utf8_polish_ci
```

WYBRANIE BAZY DANYCH:

```
USE db_name
```

PRZYKŁAD:

```
USE baza_test
```

WYŚWIETLENIE WSZYSTKICH BAZ DANYCH

```
SHOW DATABASES
```

TWORZENIE TABEL:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name  
    (create_definition,...)  
    [table_options]  
    [partition_options]
```

PRZYKŁAD:

```
CREATE TABLE IF NOT EXISTS uzytkownicy  
(id_uzytkownicy INT(11) AUTO_INCREMENT PRIMARY KEY NOT NULL, login  
TEXT(30) NOT NULL, haslo TEXT(30) NOT NULL, nazwisko TEXT(30) NULL, imie  
TEXT(30) NULL);
```