System logowania i rejestracji

Spis treści

Baza danych do przechowywania użytkowników

Informacje dotyczące **kont użytkowników** najlepiej przechowywać w bazie **MySQL.** Na prawdę – nie warto bawić się w pliki tekstowe pisząc tak rozbudowane skrypty. Tabela będzie przechowywać podstawowe informacje takie jak *id*, *login*, *hasło*, *email*, *data rejestracji*, *data logowania* oraz *adres IP*.

```
CREATE TABLE IF NOT EXISTS `uzytkownicy` (
    `id` int(10) NOT NULL AUTO_INCREMENT,
    `login` varchar(255) NOT NULL,
    `haslo` varchar(255) NOT NULL,
    `email` varchar(255) NOT NULL,
    `rejestracja` int(10) NOT NULL,
    `logowanie` int(10) NOT NULL,
    `ip` varchar(15) NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 AUTO_INCREMENT=1;
```

Posiadając bazę danych proponuję od razu **dodać do niej pierwszego użytkownika** – administratora. Dzięki temu od razu będziemy widzieć efekty pracy. Dane logowania to:

```
Login: admin
Hasło: haslo
```

Hasło będziemy przechowywać w bazie danych **w postaci zakodowanego ciągu MD5**. Dzięki temu w razie włamania do serwisu, włamywacz nie będzie posiadał listy haseł, a jedynie ich zakodowane odpowiedniki. Podczas logowania hasło wpisane przez użytkownika także jest od razu kodowane i porównywane z wartością z bazy.

Uwaga! Warto wspomnieć, że w obecnych czasach kodowanie MD5 jest bardzo łatwe do złamania (mimo że jest to kodowanie jednostronne). Z tego powodu, pisząc bezpieczny skrypt należy skorzystać z innej funkcji szyfrującej hasła niż **MD5**. Dobrym i często stosowanym rozwiązaniem jest także **użycie „soli”** (z ang. salt). Polega to na szyfrowaniu haseł poddanych jakiejś transformacji tekstu.

Dodajmy konto administratora do bazy danych:

```
INSERT INTO `uzytkownicy` (`id`, `login`, `haslo`, `email`, `rejestracja`,
`logowanie`, `ip`)
    VALUES (1, 'admin', '207023ccb44feb4d7dadca005ce29a64',
'admin@admin.pl', '1357063200', '1357063200', '127.0.0.1');
```

Formularz HTML rejestracji użytkownika

Formularz rejestracji ma za zadanie pobrać od użytkownika niezbędne dane wymagane do utworzenia konta. Dane zostaną wysłane do **skryptu PHP** metodą *POST* po kliknięciu w przycisk. Formularz znajduje się w pliku *rejestracja.php*.

```
<form method="POST" action="rejestracja.php">
    <b>Login:</b> <input type="text" name="login"><br><br>
    <b>Hasło:</b> <input type="password" name="haslo1"><br>
    <b>Powtórz hasło:</b> <input type="password" name="haslo2"><br><br>
    <b>Email:</b> <input type="text" name="email"><br>
    <input type="submit" value="Utwórz konto" name="rejestruj">
</form>
```

Skrypt rejestracji PHP

Skrypt dopisujmy do tego samego pliku co formularz. Przed utworzeniem konta, **w skrypcie należy wykonać poniższe kroki**:

1. Połączyć się z bazą MySQL
2. Pobrać i przefiltrować dane formularza z tablicy super globalnej $POST*.
3. Sprawdzić czy login nie jest zajęty.
4. Zapisać w bazie danych.
5. Zamknąć połączenie MySQL.

Zawsze przed zapisem jakichkolwiek danych do bazy danych, **należy odpowiednio przygotować zapisywaną zmienną**. Inaczej skrypt będzie źle zabezpieczony. Ataki **SQL Injection** to ciągle jedne z najczęstszych ataków. Ja utworzyłem specjalną funkcję, która zabezpiecza dane pochodzące od użytkownika:

```
<?php
    mysql_connect("localhost","admin","haslo");
    mysql_select_db("baza");

    function filtruj($zmienna)
    {
        if(get_magic_quotes_gpc())
            $zmienna = stripslashes($zmienna); // usuwamy slashe

        // usuwamy spacje, tagi html oraz niebezpieczne znaki
        return mysql_real_escape_string(htmlspecialchars(trim($zmienna)));
    }

    if (isset($_POST['rejestruj']))
    {
        $login = filtruj($_POST['login']);
        $haslo1 = filtruj($_POST['haslo1']);
        $haslo2 = filtruj($_POST['haslo2']);
        $email = filtruj($_POST['email']);
        $ip = filtruj($_SERVER['REMOTE_ADDR']);

        // sprawdzamy czy login nie jest już w bazie
        if (mysql_num_rows(mysql_query("SELECT login FROM uzytkownicy WHERE
login = '".$login."';")) == 0)
        {
```

```
            if ($haslo1 == $haslo2) // sprawdzamy czy hasła takie same
            {
                mysql_query("INSERT INTO `uzytkownicy` (`login`, `haslo`,
`email`, `rejestracja`, `logowanie`, `ip`)
                    VALUES ('".$login."', '".md5($haslo1)."', '".$email."',
'".time()."', '".time()."', '".$ip."');");

                echo "Konto zostało utworzone!";
            }
            else echo "Hasła nie są takie same";
        }
        else echo "Podany login jest już zajęty.";
    }
?>
```

Formularz HTML logowania użytkownika

Mając gotową rejestrację i strukturę bazy danych, **można wziąć się za skrypt logowania**. Będzie w nim wymagane użycie **sesji**. Sesje startuje się na samym początku dokumentu, jeszcze przed kodem HTML. Podobnie jak przy rejestracji, dane są przesyłane do bazy metodą *POST*. Warto dodać też **połączenie z bazą danych** na górze dokumentu, tam gdzie startujemy sesję. Nazwa pliku to *logowanie.php*.

```
<?php
    session_start();
    mysql_connect("localhost","admin","haslo");
    mysql_select_db("baza");
?>

<form method="POST" action="logowanie.php">
    <b>Login:</b> <input type="text" name="login"><br>
    <b>Hasło:</b> <input type="password" name="haslo"><br>
    <input type="submit" value="Zaloguj" name="loguj">
</form>
```

Skrypt PHP odpowiedzialny za logowanie umieścimy w tym samym pliku co formularz. **Logując się filtrujemy dane**. Następnie sprawdzamy czy określony login i hasło występują w bazie, jeżeli tak – logujemy użytkownika. W momencie logowania należy uaktualnić **informacje o użytkowniku** czyli datę logowania oraz aktualny adres IP.

```
<?php
    function filtruj($zmienna)
    {
        if(get_magic_quotes_gpc())
            $zmienna = stripslashes($zmienna); // usuwamy slashe

        // usuwamy spacje, tagi html oraz niebezpieczne znaki
        return mysql_real_escape_string(htmlspecialchars(trim($zmienna)));
    }

    if (isset($_POST['loguj']))
    {
        $login = filtruj($_POST['login']);
        $haslo = filtruj($_POST['haslo']);
        $ip = filtruj($_SERVER['REMOTE_ADDR']);

        // sprawdzamy czy login i hasło są dobre
```

```
        if (mysql_num_rows(mysql_query("SELECT login, haslo FROM
uzytkownicy WHERE login = '".$login."' AND haslo = '".md5($haslo)."';")) >
0)
        {
            // uaktualniamy date logowania oraz ip
            mysql_query("UPDATE `uzytkownicy` SET (`logowanie` =
'".time()."', `ip` = '".$ip."'') WHERE login = '".$login."';");

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

            // zalogowany
        }
        else echo "Wpisano złe dane.";
    }
?>
```

Jak wykonać profesjonalny system logowania i dostępu na hasło?

Chcesz stworzyć profesjonalny system do rejestrowania użytkowników, logowania na hasło, przypominania haseł i wylogowania.

Wielokrotnie zastrzegałem się, że takiego systemu nie opiszę. Z prostego powodu - każda witryna stosuje inny system logowania ponieważ jest on pisany pod konkretne cele i założenia. Stąd opcje i możliwości systemu zwykle są bardzo różne. Postanowiłem jednak napisać w miarę uniwersalne rozwiązanie, które można przemodelować do innego wariantu.

System logowania to jedna z bardziej złożonych części każdego serwisu. Musi być niezawodny, spełniać pewne standardy i dodatkowo musi być na tyle prosty, aby jakiekolwiek zmiany w przyszłości były możliwe, szczególnie gdy ma służyć całemu serwisowi w wielu różnych działach.

Kilka założeń naszego systemu:
• rejestracja wymaga potwierdzenia listu z odnośnikiem wysyłanego do użytkownika po wypełnieniu formularza - to bardzo ważne, bo zapobiega zapisaniu przypadkowych osób do systemu i spełnia rolę systemu antyspamowego
• dane wpisywane do formularza rejestracji są testowane pod względem poprawności, podanie adresu e-mail jest obowiązkowe
• hasła użytkowników są szyfrowane
• jest możliwość przypomnienia hasła, jeżeli użytkownik o nim zapomni
• użytkownik może zmienić swoje dane w ograniczonym zakresie
• do zalogowania wykorzystywany jest mechanizm sesji

Jeszcze raz kładę nacisk na fakt, że każdy system jest inny. Jeden pozwala zmienić adres e-mail użytkownika, inny nie, bo groziłoby to poważnymi konsekwencjami. Jeden zezwala na skasowanie konta użytkownika, inny nie, bo spójność danych zostałaby naruszona, jeżeli ten użytkownik brał udział w dyskusjach na forum lub dodawał treści do serwisu. A zauważ co by się stało gdyby nowy użytkownik założył konto o takim samym loginie, jak miał poprzedni użytkownik...

Różnych drobnych, ale istotnych niuansów jest wiele. Dlatego podstawą jest solidne zaplanowanie systemu logowania i przemyślenie, jakie dane będą potrzebne. Jeżeli wiesz co chcesz uzyskać możesz przystąpić do pracy.

Baza danych

Mój plan jest już w głowie. Pierwsza czynność to stworzenie tabeli o nazwie `logowanie` w bazie danych, która będzie przechowywała informacje o użytkowniku:

```
CREATE TABLE logowanie (
  login VARCHAR(50) NOT NULL PRIMARY KEY,
  haslo VARCHAR(32) NOT NULL,
  kod VARCHAR(32) NOT NULL,
  status TINYINT UNSIGNED NOT NULL,
  data DATETIME NOT NULL,
  email VARCHAR(120) NOT NULL,

  imie VARCHAR(200)
)
```

Pola oznaczają:
• `login` - unikatowy login użytkownika do 50 znaków
• `haslo` - hasło użytkownika, zawsze 32 znaki ponieważ będzie zakodowane algorytmem MD5
• `kod` - specjalny kod wysyłany do potwierdzenia rejestracji
• `status` - pole opisujące stan konta - będę używał dwóch wartości: 1 gdy użytkownik oczekuje na rejestrację i 5 gdy już jest zarejestrowany
• `data` - data operacji (np. rejestracji)
• `e-mail` - adres e-mail użytkownika
• `imie` - dodatkowe pole tekstowe na imie, takich pól może być więcej, np. na wiek, zawód, opis zainteresowań, itd.

Rejestracja

Początek za nami. Zacznijmy od systemu rejestracji. Umieść go na stronie o nazwie `rejestracja.php`:

```
<?
$mysql_host = "localhost";
$mysql_login = "user";
$mysql_haslo = "password";
$mysql_baza = "baza1";
$mysql_tabela = "logowanie";
$twoj_adres = "twoj@adres.email.pl";

$opcja = trim($_REQUEST["opcja"]);

if ($opcja=="test") {

  // ************************************************
  // ********** 1. sprawdzenie danych i dodanie użytkownika
  // ************************************************

  $login = htmlspecialchars(stripslashes(trim($_POST["login"])),
ENT_QUOTES);
```

```
  $haslo = htmlspecialchars(stripslashes(trim($_POST["haslo"])),
ENT_QUOTES);
  $email = htmlspecialchars(stripslashes(trim($_POST["email"])),
ENT_QUOTES);
  $imie = htmlspecialchars(stripslashes(trim($_POST["imie"])), ENT_QUOTES);

  if (strlen($login)<3 or strlen($login)>50
      or !eregi("^[a-zA-Z0-9_.]+$",$login)) { $blad++;
    echo "<span style=\"color:red;\">Login musi mieć od 3 do 50 znaków
    bez polskich liter i spacji!</span><br />";
  } else {
    if ($baza = mysql_connect($mysql_host, $mysql_login, $mysql_haslo)) {
      if (mysql_select_db($mysql_baza)) {
       $wynik=mysql_query("SELECT * FROM $mysql_tabela WHERE
login='$login'");
      } else echo "Nie można połączyć się z bazą";
      mysql_close($baza);
    } else echo "Nie można połączyć się z serwerem MySQL";
    if (mysql_num_rows($wynik)<>0) { $blad++;
      echo "<span style=\"color:red;\">Login już został przez kogoś użyty!.
      Zaproponuj inny!</span><br />";
    }
  }

  if (strlen($haslo)<6 or strlen($haslo)>50
     or !eregi("^[a-zA-Z0-9]+$",$haslo)) { $blad++;
    echo "<span style=\"color:red;\">Hasło musi mieć od 6 do 50 znaków
    bez polskich liter i spacji!</span><br />";
  }

  if (!eregi("^[0-9a-z_.-]+@([0-9a-z-]+\.)+[a-z]{2,4}$",$email)) { $blad++;
    echo "<span style=\"color:red;\">E-mail nie został
    podany prawidłowo!</span><br />";
  }

  if ($blad==0) {
    $kod = uniqid(rand());
    $haslo = md5($haslo); // zaszyfrowanie hasla
    if ($baza = mysql_connect($mysql_host, $mysql_login, $mysql_haslo)) {
      if (mysql_select_db($mysql_baza)) {
        $wynik = mysql_query("INSERT INTO $mysql_tabela
        VALUES('$login', '$haslo', '$kod', 1, NOW(), '$email', '$imie')");
      } else echo "Nie można połączyć się z bazą";
      mysql_close($baza);
    } else echo "Nie można połączyć się z serwerem MySQL";

    if ($wynik) {
$list="
Aby potwierdzic rejestracje kliknij w ciagu 48 godzin na adres:
http://adres.pl/rejestracja.php?opcja=potwierdz&kod=$kod
Jezeli nie chcesz sie rejestrowac, zignoruj ten list.
";
      mail($email, "Rejestracja", $list, "From: <$twoj_adres>");
      echo "<p>Aby dokończyć proces rejestracji odbierz e-mail</p>";
    }
  } else $opcja="";
}

if ($opcja=="") {
```

```
  // ************************************************
  // ********** 2. formularz zakładania konta
  // ************************************************

echo <<<KONIEC
<form action="rejestracja.php" method="post">
<input type="hidden" name="opcja" value="test" />
<table>
<tr>
  <td>login:*</td>
  <td><input type="text" name="login" value="$login" /></td>
</tr>
<tr>
  <td>hasło:*</td>
  <td><input type="password" name="haslo" value="$haslo" /></td>
</tr>
<tr>
  <td>e-mail:*</td>
  <td><input type="text" name="email" value="$email" /></td>
</tr>
<tr>
  <td>imie i nazwisko:</td>
  <td><input type="text" name="imie" value="$imie" /></td>
</tr>
<tr>
  <td> </td>
  <td><input type="submit" value=" OK, rejestruję się!" /></td>
</tr>
</table>
</form>
KONIEC;
}

if ($opcja=="potwierdz") {
  // ************************************************
  // ********** 3. potwierdzenie rejestracji, uaktywnienie użytkownika
  // ************************************************

  $kod = htmlspecialchars(stripslashes(trim($_GET["kod"])), ENT_QUOTES);
  if ($kod<>"") {
    if ($baza = mysql_connect($mysql_host, $mysql_login, $mysql_haslo)) {
      if (mysql_select_db($mysql_baza)) {
        $wynik = mysql_query("DELETE FROM $mysql_tabela
          WHERE data<=DATE_SUB(NOW(),INTERVAL 2 DAY) and status=1");
        $wynik = mysql_query("UPDATE $mysql_tabela
          SET status='5', data=NOW() WHERE kod='$kod' and status=1");
        $wynik = mysql_query("SELECT * FROM $mysql_tabela
          WHERE kod='$kod' and status=5");
      } else echo "Nie można połączyć się z bazą";
      mysql_close($baza);
    } else echo "Nie można połączyć się z serwerem MySQL";
    if (mysql_num_rows($wynik)==1) {
      $dane = mysql_fetch_array($wynik);
      echo "<p>Dziękujemy. Rejestracja została zakończona poprawnie.</p>";
    }
  }
  if ($kod=="" or mysql_num_rows($wynik)<>1) {
    echo "<p>Rejestracja nie może zostać dokończona -
    sprawdź czy link jest poprawny!</p>";
  }
}
```

```
if ($opcja=="przypomnij") {

  // ************************************************
  // ********** 4. formularz przypominania danych
  // ************************************************

echo <<<KONIEC
<p>Wpisz login użyty podczas rejestracji.
Po chwili otrzymasz mailem nowe hasło.</p>
<form action="rejestracja.php" method="post">
<input type="hidden" name="opcja" value="wyslijhaslo" />
<table>
<tr>
  <td>login:</td>
  <td><input type="text" name="login" value="$login" /></td>
</tr>
<tr>
  <td> </td>
  <td><input type="submit" value=" przypomnij " /></td>
</tr>
</table>
</form>
KONIEC;
}

function haslo() {
  $min = 6; $max = 12;
  for($i=0;$i<rand($min,$max);$i++) {
    $znak=chr(rand(48,122));
    if (eregi("[0-9a-zA-Z]",$znak)) $haslo .= $znak;
    else $i--;
  }
  return $haslo;
}

if ($opcja=="wyslijhaslo") {

  // ************************************************
  // ********** 5. zmiana hasla i wyslanie go do uzytkownika
  // ************************************************

  $login = htmlspecialchars(stripslashes(trim($_POST["login"])),
ENT_QUOTES);
  $hasloczytelne = haslo();
  $haslo = md5($hasloczytelne);
  if ($login<>"") {
    if ($baza = mysql_connect($mysql_host, $mysql_login, $mysql_haslo)) {
      if (mysql_select_db($mysql_baza)) {
        $wynik = mysql_query("UPDATE $mysql_tabela
        SET haslo='$haslo' WHERE login='$login' and status=5");
        $wynik = mysql_query("SELECT * FROM $mysql_tabela
           WHERE login='$login' and status=5");
      } else echo "Nie można połączyć się z bazą";
      mysql_close($baza);
    }
    if (mysql_num_rows($wynik)==1) {
      $dane = mysql_fetch_array($wynik);
      $email = $dane["email"];
      $list="Oto przypominane haslo: $hasloczytelne";
      mail($email, "Przypomnienie hasla", $list,"From: <$twoj_adres>");
```

```
      echo "<p>Hasło zostało wysłane mailem...</p>";
    } else {
      echo "<p>Użytkownik o podanym loginie nie istnieje!</p>";
    }
  }
}
?>
```

System rejestracji i potwierdzania jest największy. Zmienna `$opcja` przekazywana przez odnośnik lub formularz określa, który moduł (warunek `if`) zostanie uruchomiony. Jeżeli w odnośniku nie będzie żadnych parametrów, pojawi się formularz do zakładania konta (nr 2 w numerowanych modułach/warunkach powyżej).

Po wypełnieniu formularza realizowany jest moduł 1. Testuje on czy dane są poprawne. Np. login musi mieć od 3 do 50 znaków z zakresu a-z, A-Z, 0-9 oraz podkreślenie lub kropkę. Nie może też występować wcześniej w bazie. Testowane jest także hasło i e-mail.

Jeżeli są błędy pojawiają się na czerwono, a użytkownik musi poprawiać dane w formularzu do skutku. Jeżeli wszystko jest OK, tworzony jest losowy kod do potwierdzenia listu oraz szyfrowane jest hasło. Dane zapisywane są do tabeli, a do użytkownika wysyłany jest list, który musi potwierdzić.

W momencie wysyłki status użytkownika równy jest 1, gdy potwierdzi wpis zmieni się na 5. Będzie można wykorzystywać dane tylko tych użytkowników, którzy mają pole status=5, bo tylko wtedy potwierdzili swoje dane.

Użytkownik dostaje list. Po kliknięciu na link obsługuje go nasz moduł 4 potwierdzający rejestrację. Jeżeli dane są poprawne, a więc kod się zgadza, użytkownik ma zmieniany status na 5. Przy okazji kasowani są użytkownicy, którzy nie potwierdzili rejestracji przez 2 dni aby nie blokować dłużej wolnych loginów.

Moduł 4 i 5 realizują funkcję przypomnienia hasła. Ponieważ hasła nie można uzyskać z bazy danych (jest ono szyfrowane jednostronnie). Po wypełnieniu formularza przypominania z loginem, generowane jest nowe hasło użytkownika i wysyłane jest do niego mailem.

Ze względów bezpieczeństwa wysyłane jest tylko hasło, bez loginu (gdyby ktoś odczytał list nie zaloguje się bez znajomości loginu). Podczas przypominania podawany jest tylko login, którego użytkownik nie może zapomnieć.

Logowanie

Mamy już zarejestrowanych użytkowników, więc możemy ich logować do systemu. Do testu wystarczy jakaś prostacka strona z ramką. Niech nazywa się `index.php`:

```
<?
session_start();

$mysql_host = "localhost";
$mysql_login = "user";
$mysql_haslo = "password";
$mysql_baza = "baza1";
$mysql_tabela = "logowanie";
```

```
// ************************************************
// ********** wylogowanie i zalogowanie z ustaleniem sesji...
// ************************************************

$login = $_POST["login"];
$haslo = $_POST["haslo"];

if ($_GET["login"]=="koniec") { // wylogowanie
  session_unset(); session_destroy();
} else if ($login<>"" and $haslo<>"") {
  $haslo = md5($haslo);
  if ($baza = mysql_connect($mysql_host, $mysql_login, $mysql_haslo)) {
    if (mysql_select_db($mysql_baza)) {
      $wynik=mysql_query("SELECT * FROM $mysql_tabela WHERE
        login='$login' and haslo='$haslo' and status=5");
      if (mysql_num_rows($wynik)==1) {
        $dane=mysql_fetch_array($wynik);
        $_SESSION["zalogowany"]="tak";
        $_SESSION["login"]=$dane["login"];
        $_SESSION["imie"]=$dane["imie"];
        $_SESSION["email"]=$dane["email"];
      }
    } else echo "Nie można połączyć się z bazą";
    mysql_close($baza);
  }
}
?>

<table border="1" width="100%">
<tr><td valign="top" width="200">

<p>
lewy bok strony
</p>

<?

// ************************************************
// ********** panel formularza do zalogowania i wylogowania
// ************************************************

if ($_SESSION["zalogowany"]=="tak") {
  echo "<p>Witaj <b>".$_SESSION["login"]."</b></p>";
  echo "<br><a href=\"index.php?login=koniec\">wyloguj się</a>";
  echo "<br><a href=\"zmiana.php\">zmień hasło</a>";
} else {

echo <<<KONIEC
<form action="index.php" method="post">
<table>
<tr>
  <td align="right">login: </td>
  <td><input type="text" name="login" /></td>
</tr>
<tr>
  <td align="right">hasło: </td>
  <td><input type="password" name="haslo" /></td>
</tr>
<tr>
  <td colspan="2" align="right">
  <input type="submit" value="zaloguj" /></td>
```

```
</tr>
</table>
</form>
<a href="rejestracja.php?opcja=przypomnij">przypomnij hasło</a></br />
<a href="rejestracja.php">ZAREJESTRUJ SIĘ!</a>
KONIEC;

}
?>

</td><td valign="top">

<p>
prawy bok strony...
</p>
<p>
<a href="index.php">index.php</a> - <a href="index2.php">index2.php</a>
</p>

<?
if ($_SESSION["zalogowany"]=="tak") {
 echo "<p>TEN tekst widzi tylko zalogowany użytkownik! </p>";
}
?>

</td>
<tr>
</table>
```

Może nawet część kodu nie jest Ci obca? Opisywałem logowanie i wylogowanie z użyciem sesji w osobnej poradzie (link na końcu tekstu).

Pierwszą czynnością na stronie jest funkcja uaktywniająca sesję `session_start()`. Następnie pobieram z formularza logowania login i hasło. Jeżeli login ma nazwę "koniec", wtedy użytkownik jest wylogowany z serwisu.

Jeżeli dane są poprawne, użytkownik zostaje zalogowany (zwróć uwagę, że status musi być równy 5) i ustawiane są zmienne sesji, z których można korzystać na innych stronach. Zapisuję login, e-mail i imie oraz ustawiam zmienną `$_SESSION["zalogowany"]="tak"`, aby nie odwoływać się co chwilę do bazy i sprawdzać czy użytkownik faktycznie jest zalogowany.

Gdzieś dalej na stronie umieszczam formularz logowania. Może on być umieszczony na wszystkich stronach serwisu. Jeżeli użytkownik jest zalogowany zobaczy powitanie i opcję wylogowania oraz zmiany hasła/danych, jeżeli nie jest zalogowany zobaczy okienko logowania.

Do testów poruszania się między stronami możesz wykorzystać inną stronę, np. `index2.php`:

```
<?
session_start();
?>

<table border="1" width="100%">
<tr><td valign="top" width="200">
```

```
<p>
lewy bok strony
</p>

<?

// ************************************************
// ********** panel formularza do zalogowania i wylogowania
// ************************************************

if ($_SESSION["zalogowany"]=="tak") {
  echo "<p>Witaj <b>".$_SESSION["login"]."</b></p>";
  echo "<br><a href=\"index.php?login=koniec\">wyloguj się</a>";
  echo "<br><a href=\"zmiana.php\">zmień hasło</a>";
} else {

echo <<<KONIEC
<form action="index.php" method="post">
<table>
<tr>
  <td align="right">login: </td>
  <td><input type="text" name="login" /></td>
</tr>
<tr>
  <td align="right">hasło: </td>
  <td><input type="password" name="haslo" /></td>
</tr>
<tr>
  <td colspan="2" align="right">
  <input type="submit" value="zaloguj" /></td>
</tr>
</table>
</form>
<a href="rejestracja.php?opcja=przypomnij">przypomnij hasło</a></br />
<a href="rejestracja.php">ZAREJESTRUJ SIĘ!</a>
KONIEC;

}
?>

</td><td valign="top">

<p>
prawy bok strony...
</p>
<p>
<a href="index.php">index.php</a> - <a href="index2.php">index2.php</a>
</p>

<?
if ($_SESSION["zalogowany"]=="tak") {
 echo "<p>TEN tekst widzi tylko zalogowany użytkownik! </p>";
}
?>

</td>
<tr>
</table>
```

Na tej stronie umieszczam oczywiście funkcję `session_start();` i tylko okienko logowania. Zalogowanie odbywa się na stronie `index.php`.

Zmiana danych

Zmiana danych jest dosyć newralgicznym punktem. Na pewno nie można zmieniać loginu bo jest to klucz do innych danych. Można (a nawet trzeba) zmieniać hasło. Pytanie czy można zmieniać e-mail?

Użytkownikowi może się zmienić adres e-mail, to fakt. Ale co się stanie, gdy ktoś wykorzysta swoje konto do niezbyt uczciwych celów, po czym zmieni szybko e-mail? Tutaj jest wiele dylematów i zależy to od systemu. U mnie nie będzie takiej opcji.

Warto rozważyć czy ewentualnie stworzyć zwykłe pole zamiany, czy cały system potwierdzania nowego adresu e-mail, aby mieć pewność, że nowy adres został podany poprawnie i ktoś odebrał w nim nasz list potwierdzający.

Ok, wykonajmy zatem zmianę hasła i imienia użytkownika. Wszystko umieść w pliku `zmiana.php`. Zmianę będzie mógł przeprowadzić tylko użytkownik, który jest zalogowany w systemie.

```
<?
session_start();

$mysql_host = "localhost";
$mysql_login = "user";
$mysql_haslo = "password";
$mysql_baza = "baza1";
$mysql_tabela = "logowanie";

$opcja = trim($_REQUEST["opcja"]);

if ($opcja=="zmien" and $_SESSION["zalogowany"]=="tak") {
  $haslo = htmlspecialchars(stripslashes(trim($_POST["haslo"])),
ENT_QUOTES);
  $imie = htmlspecialchars(stripslashes(trim($_POST["imie"])), ENT_QUOTES);

  if (strlen($haslo)<6 or strlen($haslo)>50
     or !eregi("^[a-zA-Z0-9]+$",$haslo)) { $blad++;
    echo "<span style=\"color:red;\">Hasło musi mieć od 6 do 50 znaków
    bez polskich liter i spacji!</span><br />";
  }

  if ($blad==0) {
    $kod = uniqid(rand());
    $haslo = md5($haslo); // zaszyfrowanie hasla

    if ($baza = mysql_connect($mysql_host, $mysql_login, $mysql_haslo)) {
      if (mysql_select_db($mysql_baza)) {
        $wynik = mysql_query("UPDATE $mysql_tabela
        SET haslo='$haslo', imie='$imie' WHERE
login='{$_SESSION["login"]}'");
      } else echo "Nie można połączyć się z bazą";
      mysql_close($baza);
    } else echo "Nie można połączyć się z serwerem MySQL";
    if ($wynik) {
      echo "Dane zostały zmienione";
      $_SESSION["imie"]=$imie;
    }
```

```
  } else echo "Dane nie zostały zmienione!";
}

if ($_SESSION["zalogowany"]=="tak") {

    if ($baza = mysql_connect($mysql_host, $mysql_login, $mysql_haslo)) {
      if (mysql_select_db($mysql_baza)) {
        $wynik=mysql_query("SELECT * FROM $mysql_tabela WHERE
          login='{$_SESSION["login"]}'");
        if (mysql_num_rows($wynik)==1) {
          $dane=mysql_fetch_array($wynik);

echo <<<KONIEC
<form action="zmiana.php" method="post">
<input type="hidden" name="opcja" value="zmien" />

<table>
<tr>
  <td align="right">hasło: </td>
  <td><input type="password" name="haslo" /></td>
</tr>
<tr>
  <td align="right">imie i nazwisko: </td>
  <td><input type="text" name="imie" value="{$dane["imie"]}" /></td>
</tr>
<tr>
  <td colspan="2" align="right">
  <input type="submit" value="zmien" /></td>
</tr>
</table>
</form>
KONIEC;
        }
      } else echo "Nie można połączyć się z bazą";
      mysql_close($baza);
    } else echo "Nie można połączyć się z serwerem MySQL";

}

?>
```

Tu również pojawia się formularz i po wprowadzeniu zmian weryfikowane są dane (jak podczas rejestracji). Jeżeli wszystko jest ok, zmiany zostają zapisane w bazie i zmiennych sesji.

Koniec

Nasz system jest gotowy w postaci surowej, a więc musisz dodać cały HTML i szablony swojego serwisu. Jak widzisz, nawet w tak prostej realizacji jest wiele pytań, na które trzeba sobie odpowiedzieć. Dlatego system warto przemyśleć i wymodelować tak, aby spełniał konkretne założenia serwisu.