

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Таныгин Максим Олегович
Должность: и.о. декана факультета фундаментальной и прикладной информатики
Дата подписания: 21.09.2023 13:06:21
Уникальный программный ключ:
65ab2aa0d384efe8480e6a4c688eddbc475e411a

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ

Проректор по учебной работе

О.Г.Локтионова

2016 г.



**РЕГИСТРАЦИЯ И АВТОРИЗАЦИЯ ПОЛЬЗОВАТЕЛЕЙ
WEB-САЙТА**

Методические указания к лабораторным работам
по курсу «Web-программирование»

Курск 2016

УДК 681.3

Составитель Е.И.Аникина

Рецензент

Кандидат технических наук, доцент кафедры информатики и
прикладной математики *Н.Н. Бочанова*

Регистрация и авторизация пользователей web-сайта:
методические указания к лабораторным работам по курсу «Web-
программирование» /Юго-Зап. гос. ун-т; сост. Е.И.Аникина. Курск,
2016. 12 с.

Содержит теоретические сведения и задания для выполнения лабораторной работы по изучению технологии программирования на языке PHP процедур регистрации и авторизации пользователей web-сайта .

Предназначено для студентов всех направлений подготовки, изучающих основы Web-программирования.

Текст печатается в авторской редакции.

Подписано в печать . Формат 60x84 1/16.
Усл. печ. л. . Уч.-изд. л. . Тираж 100 экз. Заказ . Бесплатно.

Юго-Западный государственный университет
305040, Курск, ул.50 лет Октября, 94.

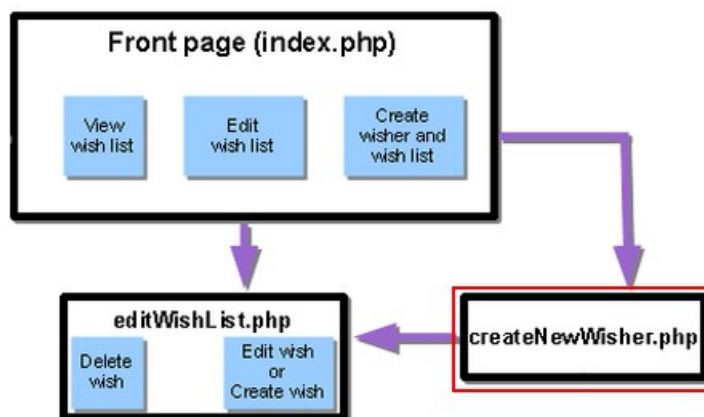
Создание нового пользователя приложения

В этом уроке рассматривается расширение приложения добавлением функции "Create a New Wisher".

Реализация затрагивает файл `index.php`, и при этом будут созданы два новых файла с именами `createNewWisher.php` и `editWishList.php`.

Данный пример использования Create a New Wisher ("Создание нового пользователя") состоит из трех действий:

1. Пользователь открывает файл `index.php` титульной страницы и щелкает ссылку для регистрации.
2. Пользователь переходит на страницу `createNewWisher.php` для создания нового автора пожелания.
3. После создания нового автора пожелания пользователь переключается на `editWishList.php`, где для него можно создать список желаний.



Текущий документ является частью краткого учебного курса "Создание приложения базы данных в IDE NetBeans для PHP".

Добавление ссылки для начала создания нового автора пожелания

Откройте `index.php`. Добавьте пустую строку под закрывающим тегом `</form>`. Введите в эту пустую строку следующий блок кода:

```
<br>Still don't have a wish list?! <a href="createNewWisher.php">Create now</a>
```

Где

- `Still don't have a wish list?!` – это текст, который отобразится на странице рядом со ссылкой.
- `` – это код ссылки, открывающей страницу `createNewWisher.php`.
- `Create now` – это текст, который отобразится в качестве ссылки.

Создание новых веб-страниц PHP

Создайте две новые веб-страницы PHP в исходных файлах проекта, как описано в уроке 2.

- `createNewWisher.php`
- `editWishList.php`

В editWishList.php добавьте текст "Hello!" к тексту в формате HTML, а в остальном оставьте всё как было. Этот файл будет изменен в последующих уроках, однако на данный момент он необходим в качестве объекта ссылки для createNewWisher.php. В оставшейся части этого урока рассматривается изменение файла createNewWisher.php.

Добавление формы HTML для ввода данных нового автора пожелания

Введите или вставьте следующий блок HTML в строку createNewWisher.php под блоком PHP:

```
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <title></title>
  </head>
  <body>
    Welcome!<br>
    <form action="createNewWisher.php" method="POST">
      Your name: <input type="text" name="user"/><br/>
      Password: <input type="password" name="password"/><br/>
      Please confirm your password: <input type="password"
name="password2"/><br/>
      <input type="submit" value="Register"/>
    </form>
  </body>
</html>
```

Примечание. Тип password это специальный тип текстового поля, в котором символы заменяются звездочками. Код представляет собой форму HTML, позволяющую ввести имя и пароль нового автора пожелания в текстовые поля. При нажатии кнопки "Register" введенные данные передаются для проверки допустимости на ту же страницу – createNewWisher.php.

Примечание. Предупреждения от средства проверки HTML можно проигнорировать.

Проверка допустимости данных и добавление их в базу данных

В этом разделе мы добавим код PHP к createNewWisher.php. Добавьте этот код к блоку PHP на верху файла. Блок PHP должен находиться над кодом HTML **all**, пустыми строками или пробелами. Расположение блока кода PHP является важным для правильного функционирования оператора переадресации. Внутри блока PHP введите или вставьте в указанном порядке блоки кода, описанные ниже в данном разделе.

Добавьте следующий код для проверки допустимости данных:

1. Необходимо инициализировать переменные. Первая группа переменных осуществляет передачу параметров доступа к базе данных, а другая группа переменных используется в работе кода PHP.
2.

```
/** database connection credentials */
$dbHost="localhost"; //on MySql
```
3.

```
$dbXeHost="localhost/XE";
$dbUsername="phpuser";
$dbPassword="phpuserpw";
```

```

4.
5. /** other variables */
6. $userNamesUnique = true;
7. $passwordsValid = true;
8. $userIsEmpty = false;
9. $passwordsEmpty = false;
10. $password2IsEmpty = false;
11.

```

12. Под переменными следует добавить блок if . Параметр блока if выполняет проверку того, что страница была запрошена из нее самой посредством метода POST. Если это не так, дальнейшие проверки допустимости не выполняются, и на экран выводится страница с пустыми полями, как описано выше.

```

13. /** Check that the page was requested from itself via the POST method. */
14. if ($_SERVER["REQUEST_METHOD"] == "POST") {
15.

```

16. Внутри фигурных скобок блока if добавьте другой блок if , позволяющий проверить, ввел ли пользователь имя автора пожелания в поле. Если текстовое поле "user" является пустым, значение \$userIsEmpty меняется на "true".

```

17. /** Check that the page was requested from itself via the POST method. */
18. if ($_SERVER["REQUEST_METHOD"] == "POST") {
19.
20. /** Check whether the user has filled in the wisher's name in the text field "user" */
21.     if ($_POST["user"]=="") {
22.         $userIsEmpty = true;
23.     }

```

24. Добавьте код, устанавливающий подключение к базе данных. Если установить подключение невозможно, то выводится ошибка MySQL или Oracle OCI8.

Для базы данных MySQL

```

/** Check that the page was requested from itself via the POST method. */
if ($_SERVER["REQUEST_METHOD"] == "POST") {

/** Check whether the user has filled in the wisher's name in the text field "user" */
if ($_POST["user"]=="") {
    $userIsEmpty = true;
}

/** Create database connection */

$con = mysqli_connect($dbHost, $dbUsername, $dbPassword);
if (!$con) {
    exit('Connect Error (' . mysqli_connect_errno() . ') '
        . mysqli_connect_error());
}
//set the default client character set
mysqli_set_charset($con, 'utf-8');

```

}

25. Добавьте код, позволяющий проверить, существует ли пользователь, имя которого соответствует указанному в поле "user". Эта задача выполняется путем поиска идентификационного номера автора пожелания в соответствии с именем, указанным в поле "user". Если такой номер существует, значение \$userNamesUnique меняется на "false".

Для базы данных MySQL

```

/** Check that the page was requested from itself via the POST method. */
if ($_SERVER["REQUEST_METHOD"] == "POST") {

/** Check whether the user has filled in the wisher's name in the text field "user" */

    if ($_POST["user"]=="") {
        $userIsEmpty = true;
    }

/** Create database connection */
    $con = mysqli_connect($dbHost, $dbUsername, $dbPassword);
    if (!$con) {
        exit('Connect Error (' . mysqli_connect_errno() . ') '
            . mysqli_connect_error());
    }
/**set the default client character set */
    mysqli_set_charset($con, 'utf-8');
/** Check whether a user whose name matches the "user" field already
exists */

    mysqli_select_db($con, "wishlist");
    $user = mysqli_real_escape_string($con, $_POST["user"]);
    $wisher = mysqli_query($con, "SELECT id FROM wishers WHERE
name='".$user."'");
    $wisherIDnum=mysqli_num_rows($wisher);
    if ($wisherIDnum) {
        $userNamesUnique = false;
    }
}
}

```

26. После кода, проверяющего уникальность пользователя, добавьте серию блоков if , проверяющих, правильно ли пользователь ввел и подтвердил пароль. Код выполняет проверку того, что поля "Password" ("password") и "Confirm Password" ("password2") заполнены и идентичны друг другу. В противном случае значения соответствующих логических переменных также изменяются.

```

27. if ($_POST["password"]=="") {
    $passwordIsEmpty = true;
28. }
    if ($_POST["password2"]=="") {
        $password2IsEmpty = true;

```

- ```

29. }
 if ($_POST["password"]!= $_POST["password2"]) {
 $passwordIsValid = false;
 }
}

```
30. Завершите блок if (`$_SERVER['REQUEST_METHOD'] == "POST"`), добавив код, вставляющий новую запись в базу данных "Wishers". В соответствии с кодом выполняется проверка того, что имя пользователя указано однозначно и что пароль введен и подтвержден правильно. Если эти условия выполнены, код извлекает значения "user" и "password" из формы HTML и вставляет их соответственно в столбцы "Name" и "Password", относящиеся к новой строке в базе данных "Wishers". После создания строки код закрывает подключение к базе данных и переадресует приложение на страницу editWishList.php.

### Для базы данных MySQL

```

/** Check that the page was requested from itself via the POST method. */
if ($_SERVER['REQUEST_METHOD'] == "POST") {
 /** Check whether the user has filled in the wisher's name in the text field
"user" */
 if ($_POST['user'] == "") {
 $userIsEmpty = true;
 }

 /** Create database connection */
 $con = mysqli_connect($dbHost, $dbUsername, $dbPassword);
 if (!$con) {
 exit('Connect Error (' . mysqli_connect_errno() . ') '
 . mysqli_connect_error());
 }
 //set the default client character set
 mysqli_set_charset($con, 'utf-8');

 /** Check whether a user whose name matches the "user" field already exists
*/
 mysqli_select_db($con, "wishlist");
 $user = mysqli_real_escape_string($con, $_POST['user']);
 $wisher = mysqli_query($con, "SELECT id FROM wishers WHERE name=" .
$user."");
 $wisherIDnum=mysqli_num_rows($wisher);
 if ($wisherIDnum) {
 $userNamesUnique = false;
 }

 /** Check whether a password was entered and confirmed correctly */
 if ($_POST['password'] == "") {
 $passwordIsEmpty = true;
 }
 if ($_POST['password2'] == "") {
 $password2IsEmpty = true;
 }
 if ($_POST['password'] != $_POST['password2']) {

```

```

 $passwordIsValid = false;
}

/** Check whether the boolean values show that the input data was validated
successfully.
* If the data was validated successfully, add it as a new entry in the "wishers"
database.
* After adding the new entry, close the connection and redirect the application
to editWishList.php.
*/
if (!$userIsEmpty && $userNamesUnique && !$passwordIsEmpty && !
$password2IsEmpty && $passwordIsValid) {
 $password = mysqli_real_escape_string($con, $_POST['password']);
 mysqli_select_db($con, "wishlist");
 mysqli_query($con, "INSERT wishers (name, password) VALUES ('" .
$user . "', '" . $password . "')");
 mysqli_free_result($wisher);
 mysqli_close($con);
 header('Location: editWishList.php');
 exit;
}
}
whether a user whose name matches the "user" field already exists */
$query = "select ID from wishers where name = :user_bv";
$stmt = oci_parse($con, $query);
$user = $_POST['user'];
$wisherID = null;
oci_bind_by_name($stmt, ':user_bv', $user);
oci_execute($stmt);

/**Each user name should be unique. Check if the submitted user already exists.
*/
$row = oci_fetch_array($stmt, OCI_ASSOC);
if ($row) {
 $wisherID = $row['ID'];
}
if ($wisherID != null) {
 $userNamesUnique = false;
}
//Check for the existence and validity of the password
if ($_POST['password'] == "") {
 $passwordIsEmpty = true;
}
if ($_POST['password2'] == "") {
 $password2IsEmpty = true;
}
if ($_POST['password'] != $_POST['password2']) {
 $passwordIsValid = false;
}
/** Check whether the boolean values show that the input data was validated
successfully.

```



```

* If the data was validated successfully, add it as a new entry in the "wishers"
database.
* After adding the new entry, close the connection and redirect the application
to editWishList.php.
*/
if (!$userIsEmpty && $userNamesUnique && !$passwordIsEmpty && !
$password2IsEmpty && $passwordsValid) {

 $query = "INSERT INTO wishers (name, password) VALUES
(:user_bv, :pwd_bv)";
 $stid = oci_parse($con, $query);
 $pwd = $_POST['password'];
 oci_bind_by_name($stid, ':user_bv', $user);
 oci_bind_by_name($stid, ':pwd_bv', $pwd);
 oci_execute($stid);
 oci_free_statement($stid);
 oci_close($con);
 header('Location: editWishList.php');
 exit;
}
}

```

## Отображение сообщений об ошибках в форме ввода

Перейдем к реализации вывода сообщений об ошибках при неверно введенных данных. Реализация основывается на проверках допустимости и изменении значений логических переменных, описанных в разделе [Проверка допустимости данных и добавление их в базу данных](#).

1. Введите следующий блок кода PHP в форме ввода HTML непосредственно под именем пользователя:

```

2. Welcome!

<form action="createNewWisher.php" method="POST">
 Your name: <input type="text" name="user"/>

3.
4.
5. <?php
6. if ($userIsEmpty) {
7. echo ("Enter your name, please!");
8. echo ("
");
9. }
10. if (!$userNamesUnique) {
11. echo ("The person already exists. Please check the spelling and try
again");
12. echo ("
");
13. }
 ?>

```

14. Введите следующий блок кода PHP в форме ввода HTML под кодом для ввода пароля:

```

15. Password: <input type="password" name="password"/>


```

16.

```

17. <?php
18. if ($passwordIsEmpty) {
19. echo ("Enter the password, please!");
20. echo ("
");
21. }
 ?>

```

22. Введите следующий блок кода PHP в форме ввода HTML под кодом для подтверждения пароля:

```

23. Please confirm your password: <input type="password"
 name="password2"/>


```

24.

25.

26.

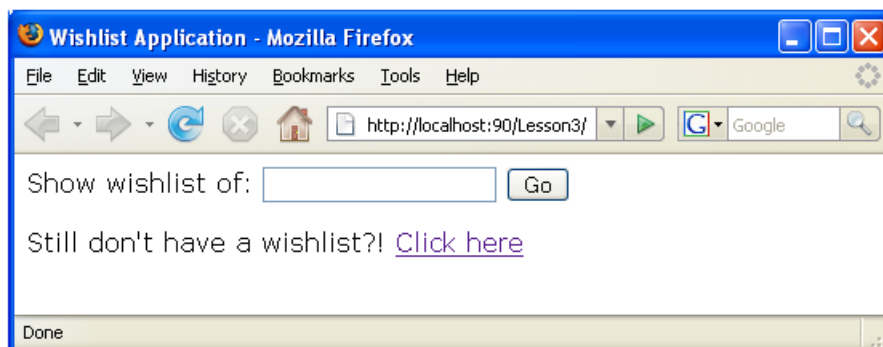
```

27. <?php
28. if ($password2IsEmpty) {
29. echo ("Confirm your password, please");
30. echo ("
");
31. }
32. if (!$password2IsEmpty && !$passwordIsValid) {
33. echo ("The passwords do not match!");
34. echo ("
");
35. }
 ?>

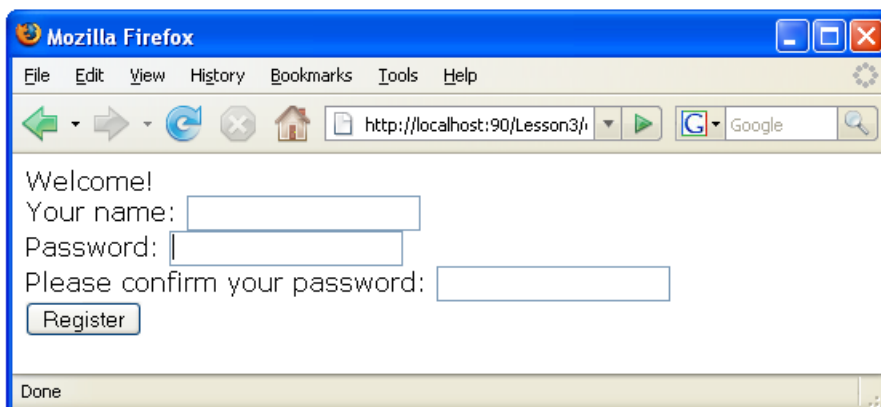
```

Тестирование функциональных возможностей по созданию нового пользователя "Create New Wisher"

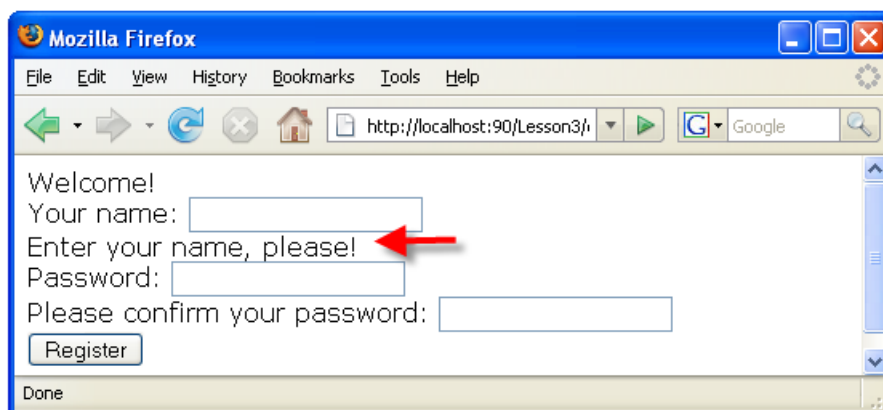
1. Запустите приложение. Откроется страница-указатель.
- 2.



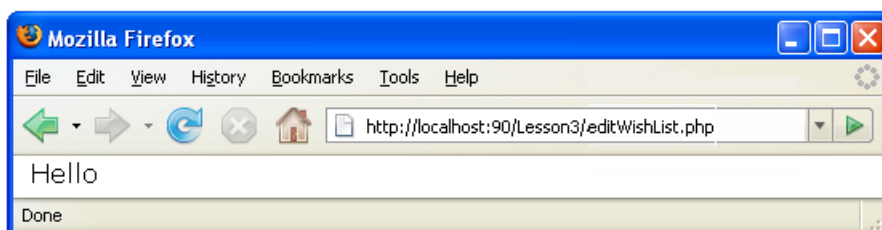
3. На странице-указателе щелкните ссылку рядом с текстом "Still don't have a wish list?" Откроется следующая форма:



4. Оставьте поля пустыми и нажмите кнопку Register ("Зарегистрировать"). На экране появится сообщение об ошибке.



5. Введите имя зарегистрированного пользователя, например, "Tom" в поле "Your name", внимательно заполните другие поля и нажмите кнопку "Register". На экране появится сообщение об ошибке.
6. Заполните поля "Password" и "Please confirm your password" различными значениями и нажмите кнопку "Register". На экране появится сообщение об ошибке.
7. Введите "Bob" в поле "Your name", укажите в полях пароля один и тот же пароль и нажмите кнопку "Register". Откроется пустая страница, однако переадресация осуществляется правильно, поскольку URL-адрес заканчивается текстом "editWishList.php":
- 8.



9. Проверьте, что данные сохранены в базе данных, путем перехода к разделу "Wishers" в окне "Services", расположенном под узлом "wishlist1", и выбора "View Data" в контекстном меню

