

# MODULE 4

Advanced PHP: Form processing and Business Logic-Cookies- Sessions & MySQL Integration- Connecting to MySQL with PHP- Performing CREATE, DELETE, INSERT, SELECT and UPDATE operations on MySQL table - Working with MySQL data-Reading from Database- Dynamic Content

## PHP COOKIES

### What is a Cookie?

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

---

### Create Cookies With PHP

A cookie is created with the `setcookie()` function.

#### Syntax

```
setcookie(name, value, expire, path, domain, secure, httponly);
```

Only the *name* parameter is required. All other parameters are optional.

```
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 *
30), "/"); // 86400 = 1 day
?>

<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    • echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```

## PHP Create/Retrieve a Cookie

The following example creates a cookie named "user" with the value "John Doe". The cookie will expire after 30 days (86400 \* 30). The "/" means that the cookie is available in entire website (otherwise, select the directory you prefer).

We then retrieve the value of the cookie "user" (using the global variable `$_COOKIE`). We also use the `isset()` function to find out if the cookie is set:

The `setcookie()` function must appear BEFORE the `<html>` tag.

The value of the cookie is automatically URLencoded when sending the cookie, and automatically decoded when received (to prevent URLencoding, use `setrawcookie()` instead)

## Deleting a Cookie

In PHP, to delete a cookie, you just have to call `setcookie()` with the name argument only, which effectively resets all stored values:

```
setcookie("vegetable");
```

To be absolutely sure your cookie is no longer valid or storing any values at all, you could also set the cookie with a date that you are sure has already expired—a date in the past:

```
setcookie("vegetable", "", time()-60, "/", ".yourdomain.com", 0);
```

When deleting a cookie in this manner, make sure you pass `setcookie()` the same path, domain, and secure parameters as you did when originally setting the cookie.

Similarly, in JavaScript you delete a cookie with reset values, like so:

```
document.cookie = "vegetable=; expires=Thu, 01 Jan 1970 00:00:00 GMT";
```

In this case, the cookie expiration is set to a specific date in the past, meaning it is expired and thus will never be set.

## PHP Session

- session is a way to store information (in variables) to be used across multiple page information is not stored on the users computer
- By default, session variables last until the user closes the browser Session variables hold information about one single user, and are available to all pages in one application

- session is started with the session\_start() function.
- Session variables are set with the PHP global variable: \$\_SESSION
- session\_start() function must be the very first thing in your document before any HTML tags
- session variables are not passed individually to each new page

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

To remove all global session variables and destroy the session, use `session_unset()` and `session_destroy()` :

## Example

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// remove all session variables
session_unset();

// destroy the session
session_destroy();
?>

</body>
</html>
```

## Connecting to MySQL with PHP

### Making a Connection

- The basic syntax for a connection to MySQL is as follows:  
`$mysqli = mysqli_connect("hostname", "username", "password", "database");`
- The value of `$mysqli` is the result of the function and is used in later functions for communicating with MySQL.
- With sample values inserted, the connection code looks like this:

```
$mysqli = mysqli_connect("localhost", "testuser", "somepass", "testDB");
```

```
1: <?php
2: $mysqli = new mysqli("localhost", "testuser", "somepass", "testDB");
3:
4: if (mysqli_connect_errno()) {
5:     printf("Connect failed: %s\n", mysqli_connect_error());
6:     exit();
7: } else {
8:     printf("Host information: %s\n", mysqli_get_host_info($mysqli));
9: }
10: ?>
```

- Save this script as `mysqlconnect.php` and place it in the document area of your web server.
- Access the script with your web browser and you will see something like the following, if the connection was successful:  
Host information: localhost via TCP/IP
- You might also see something like this:  
Host information: localhost via UNIX socket
- If the connection fails, an error message is printed. Line 5 generates an error via the `mysqli_connect_error()` function. An example is shown here, which is the output that occurs when the password is changed to an incorrect password for this user:  
Connect failed: Access denied for user 'testuser'@'localhost' (using password: YES)

- Although the connection closes when the script finishes its execution, it is a good practice to
- close the connection explicitly. You can see how to do this in line 9 of Listing 18.2, using the `mysqli_close()` function.

### **LISTING 18.2** The Modified Simple Connection Script

---

```
1: <?php
2: $mysqli = new mysqli("localhost", "testuser", "somepass", "testDB");
3:
4: if (mysqli_connect_errno()) {
5:     printf("Connect failed: %s\n", mysqli_connect_error());
6:     exit();
7: } else {
8:     printf("Host information: %s\n", mysqli_get_host_info($mysqli));
9:     mysqli_close($mysqli);
10: }
11: ?>
```

## Executing Queries

- `mysqli_query()` function in PHP is used to send your SQL query to MySQL.

```
1: <?php
2: $mysqli = mysqli_connect("localhost", "testuser", "somepass", "testDB");
3:
4: if (mysqli_connect_errno()) {
5:     printf("Connect failed: %s\n", mysqli_connect_error());
6:     exit();
7: } else {
8:     $sql = "CREATE TABLE testTable
9:         (id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
10:          testField VARCHAR(75))";
11:     $res = mysqli_query($mysqli, $sql);
12:
13:     if ($res === TRUE) {
14:         echo "Table testTable successfully created.";
15:     } else {
16:         printf("Could not create table: %s\n", mysqli_error($mysqli));
17:     }
18:
19:     mysqli_close($mysqli);
20: }
21: ?>
```

The `mysqli_query` function returns a value of `true` or `false`, and this value is checked in the `if...else` statement beginning in line 13. If the value of `$res` is `true`, a success message is printed to the screen. If you access MySQL through the command-line interface to verify the creation of the `testTable` table, you will see the following output of `DESCRIBE testTable`:

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
testField	varchar(75)	YES		NULL	

## Inserting Data with PHP

```
1: <?php
2: $mysqli = mysqli_connect("localhost", "testuser", "somepass", "testDB");
3:
4: if (mysqli_connect_errno()) {
5:     printf("Connect failed: %s\n", mysqli_connect_error());
6:     exit();
7: } else {
8:     $sql = "INSERT INTO testTable (testField) VALUES ('some value')";
9:     $res = mysqli_query($mysqli, $sql);
10:
11:     if ($res === TRUE) {
12:         echo "A record has been inserted.";
13:     } else {
14:         printf("Could not insert record: %s\n", mysqli_error($mysqli));
15:     }
16:
17:     mysqli_close($mysqli);
18: }
19: ?>
```



## LISTING 18.5 An Insert Form

---

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Record Insertion Form</title>
</head>
<body>
  <form action="insert.php" method="post">
    <p><label for="testfield">Text to Add:</label><br>
    <input type="text" id="testfield" name="testfield" size="30"></p>
    <button type="submit" name="submit" value="insert">Insert Record</button>
  </form>
</body>
</html>
```

## LISTING 18.6 An Insert Script Used with the Form

---

```
1: <?php
2: $mysqli = mysqli_connect("localhost", "testuser", "somepass", "testDB");
3:
4: if (mysqli_connect_errno()) {
5:     printf("Connect failed: %s\n", mysqli_connect_error());
6:     exit();
7: } else {
8:     $clean_text = mysqli_real_escape_string($mysqli, $_POST['testfield']);
9:     $sql = "INSERT INTO testTable (testField)
10:          VALUES ('".$clean_text."')";
11:     $res = mysqli_query($mysqli, $sql);
12:
13:     if ($res === TRUE) {
```

```
14:         echo "A record has been inserted.";
15:     } else {
16:         printf("Could not insert record: %s\n", mysqli_error($mysqli));
17:     }
18:
19:     mysqli_close($mysqli);
20: }
21: ?>
```



The screenshot shows a web browser window with the title 'Record Insertion Form'. The address bar displays 'http://www.yourdomain.com/insert\_form.html'. The form contains a text input field labeled 'Text to Add:' and a button labeled 'Insert Record'.

**FIGURE 18.1**

The HTML form for adding a record.

Enter a string in the Text to Add field, as shown in Figure 18.2.



The screenshot shows the same web browser window as Figure 18.1, but the 'Text to Add:' input field now contains the text 'Little Bobby Tables'. The 'Insert Record' button remains visible below the input field.

**FIGURE 18.2**

Text typed in the form field.

**FIGURE 18.3**

The record has been successfully added.

```
SELECT * FROM testTable;
```

The output should be as follows:

```
+-----+-----+
| id | testField |
+-----+-----+
| 1 | some value |
| 2 | Little Bobby Tables |
+-----+-----+
2 rows in set (0.00 sec)
```

## Retrieving Data with PHP

### **LISTING 18.7** A Script to Retrieve Data

```
1: <?php
2: $mysqli = mysqli_connect("localhost", "testuser", "somepass", "testDB");
3:
4: if (mysqli_connect_errno()) {
5:     printf("Connect failed: %s\n", mysqli_connect_error());
6:     exit();
7: } else {
8:     $sql = "SELECT * FROM testTable";
9:     $res = mysqli_query($mysqli, $sql);
10:
11:     if ($res) {
12:         $number_of_rows = mysqli_num_rows($res);
13:         printf("Result set has %d rows.\n", $number_of_rows);
14:     } else {
15:         printf("Could not retrieve records: %s\n", mysqli_error($mysqli));
16:     }
17:
18:     mysqli_free_result($res);
19:     mysqli_close($mysqli);
20: }
21: ?>
```

Save this script as `count.php`, place it in your web server document directory, and access it through your web browser. You should see a message like the following. (The actual number will vary depending on how many records you inserted into the table.)

Result set has 4 rows.

Line 12 uses the `mysqli_num_rows()` function to retrieve the number of rows in the resultset (`$res`), and it places the value in a variable called `$number_of_rows`. Line 13 prints this number to your browser. The number should be equal to the number of records you inserted during testing.

There's a new function in this listing that was not in previous listings. Line 18 shows the use of the `mysqli_free_result()` function. Using `mysqli_free_result()` before closing the connection with `mysqli_close()` ensures that all memory associated with the query and the result is freed for use by other scripts.

## Delete data from Mysql

Let's look at the "MyGuests" table:

id	firstname	lastname	email	reg_date
1	John	Doe	john@example.com	2014-10-22 14:26:15
2	Mary	Moe	mary@example.com	2014-10-23 10:22:30
3	Julie	Dooley	julie@example.com	2014-10-26 10:48:23

- ```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";

if (mysqli_query($conn, $sql)) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

## UPDATE

Let's look at the "MyGuests" table:

| id | firstname | lastname | email            | reg_date            |
|----|-----------|----------|------------------|---------------------|
| 1  | John      | Doe      | john@example.com | 2014-10-22 14:26:15 |
| 2  | Mary      | Moe      | mary@example.com | 2014-10-23 10:22:30 |

- ```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

if (mysqli_query($conn, $sql)) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

## Dynamic Content

- PHP can dynamically change the HTML5 it outputs based on a user's input.

```

1  <!DOCTYPE html>
2
3  <!-- Fig. 19.20: dynamicForm.php -->
4  <!-- Dynamic form. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Registration Form</title>
9          <style type = "text/css">
10             p      { margin: 0px; }
11             .error  { color: red }
12             p.head  { font-weight: bold; margin-top: 10px; }
13             label   { width: 5em; float: left; }
14          </style>
15      </head>
16      <body>
17          <?php
18              // variables used in script
19              $fname = isset($_POST[ "fname" ]) ? $_POST[ "fname" ] : "";
20              $lname = isset($_POST[ "lname" ]) ? $_POST[ "lname" ] : "";

```

```

21              $email = isset($_POST[ "email" ]) ? $_POST[ "email" ] : "";
22              $phone = isset($_POST[ "phone" ]) ? $_POST[ "phone" ] : "";
23              $book = isset($_POST[ "book" ]) ? $_POST[ "book" ] : "";
24              $os = isset($_POST[ "os" ]) ? $_POST[ "os" ] : "";
25              $iserror = false;
26              $formerrors =
27                  array( "fnameerror" => false, "lnameerror" => false,
28                      "emailerror" => false, "phoneerror" => false );
29
30              // array of book titles
31              $booklist = array( "Internet and WWW How to Program",
32                              "C++ How to Program", "Java How to Program",
33                              "Visual Basic How to Program" );
34
35              // array of possible operating systems
36              $systemlist = array( "Windows", "Mac OS X", "Linux", "Other" );
37
38              // array of name values for the text input fields
39              $inputlist = array( "fname" => "First Name",
40                              "lname" => "Last Name", "email" => "Email",
41                              "phone" => "Phone" );
42
43              // ensure that all fields have been filled in correctly
44              if ( isset( $_POST[ "submit" ] ) )
45              {
46                  if ( $fname == "" )
47                  {
48                      $formerrors[ "fnameerror" ] = true;
49                      $iserror = true;
50                  } // end if

```



```
51
52     if ( $lname == "" )
53     {
54         $formerrors[ "lnameerror" ] = true;
55         $iserror = true;
56     } // end if
57
58     if ( $email == "" )
59     {
60         $formerrors[ "emailerror" ] = true;
61         $iserror = true;
62     } // end if
63
64     if ( !preg_match( "/^\([0-9]{3}\) [0-9]{3}-[0-9]{4}$/",
65         $phone ) )
66     {
67         $formerrors[ "phoneerror" ] = true;
68         $iserror = true;
69     } // end if
70
71     if ( !$iserror )
72     {
73
74         // build INSERT query
75         $query = "INSERT INTO contacts " .
76             "( LastName, FirstName, Email, Phone, Book, OS ) " .
77             "VALUES ( '$lname', '$fname', '$email', " .
78             "'". mysql_real_escape_string( $phone ) .
79             "', '$book', '$os' )";
80
81         // Connect to MySQL
82         if ( !( $database = mysql_connect( "localhost",
83             "iw3htp", "password" ) ) )
84             die( "<p>Could not connect to database</p>" );
85
86         // open MailingList database
87         if ( !mysql_select_db( "MailingList", $database ) )
88             die( "<p>Could not open MailingList database</p>" );
89
90         // execute query in MailingList database
91         if ( !( $result = mysql_query( $query, $database ) ) )
92         {
93             print( "<p>Could not execute query!</p>" );
94             die( mysql_error() );
95         } // end if
96
97         mysql_close( $database );
```



```

97
98     print( "<p>Hi $fname. Thank you for completing the survey.
99           You have been added to the $book mailing list.</p>
100           <p class = 'head'>The following information has been
101             saved in our database:</p>
102             <p>Name: $fname $lname</p>
103             <p>Email: $email</p>
104             <p>Phone: $phone</p>
105             <p>OS: $os</p>
106             <p><a href = 'formDatabase.php'>Click here to view
107               entire database.</a></p>
108             <p class = 'head'>This is only a sample form.
109               You have not been added to a mailing list.</p>
110           </body></html>" );
111     die(); // finish the page
112   } // end if
113 } // end if
114
115 print( "<h1>Sample Registration Form</h1>
116       <p>Please fill in all fields and click Register.</p>" );
117
118 if ( $iserror )
119 {
120     print( "<p class = 'error'>Fields with * need to be filled
121           in properly.</p>" );
122 } // end if
123
124
125 print( "<!-- post form data to dynamicForm.php -->
126       <form method = 'post' action = 'dynamicForm.php'>
127       <h2>User Information</h2>
128
129       <!-- create four text boxes for user input -->" );
130 foreach ( $inputlist as $inputname => $inputalt )
131 {
132     print( "<div><label>$inputalt:</label><input type = 'text'
133           name = '$inputname' value = '' . $$inputname . '>' );
134
135     if ( $formerrors[ ( $inputname )."error" ] == true )
136         print( "<span class = 'error'>*</span>" );
137
138     print( "</div>" );
139 } // end foreach
140
141 if ( $formerrors[ "phoneerror" ] )
142     print( "<p class = 'error'>Must be in the form
143           (555)555-5555" );
144
145 print( "<h2>Publications</h2>
146       <p>Which book would you like information about?</p>
147
148       <!-- create drop-down list containing book names -->
149       <select name = 'book'>" );

```

```

150     foreach ( $booklist as $currbook )
151     {
152         print( "<option" .
153             ($currbook == $book ? " selected" : ">") .
154             $currbook . "</option>" );
155     } // end foreach
156
157     print( "</select>
158         <h2>Operating System</h2>
159         <p>Which operating system do you use?</p>
160
161         <!-- create five radio buttons -->" );
162
163     $counter = 0;
164
165     foreach ( $systemlist as $currsystem )
166     {
167         print( "<input type = 'radio' name = 'os'
168             value = '$currsystem' " );
169
170         if ( ( !$os && $counter == 0 ) || ( $currsystem == $os ) )
171             print( "checked" );
172
173         print( ">$currsystem" );
174         ++$counter;
175     } // end foreach
176
177
178     print( "<!-- create a submit button -->
179         <p class = 'head'><input type = 'submit' name = 'submit'
180         value = 'Register'></p></form></body></html>" );
181 ?><!-- end PHP script -->

```

a) Registration form after it was submitted with a missing field and an incorrectly formatted phone number



Registration Form

localhost/ch19/fig19\_20-21/dynamicForm.php

## Sample Registration Form

Please fill in all fields and click Register.  
Fields with \* need to be filled in properly.

### User Information

First Name:

Last Name:

Email:

Phone:

Must be in the form (555)555-5555

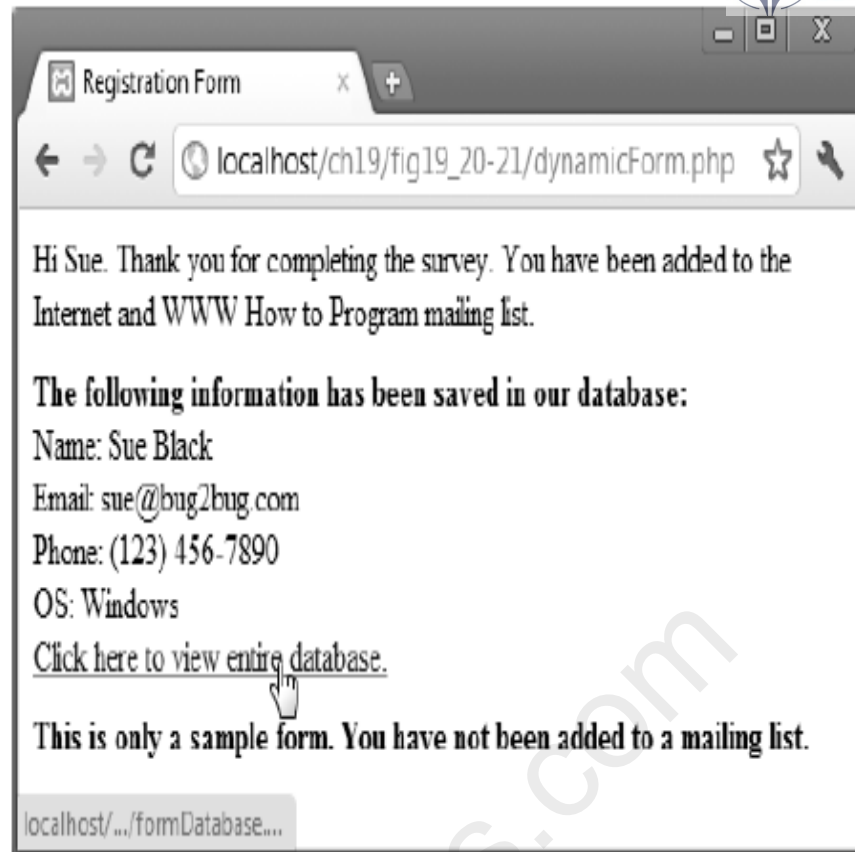
### Publications

Which book would you like information about?

### Operating System

Which operating system do you use?  
☒ Windows ☐ Mac OS X ☐ Linux ☐ Other

b) Confirmation page displayed after the user properly fills in the form and the information is stored in the database



```

1  <!DOCTYPE html>
2
3  <!-- Fig. 19.21: formDatabase.php -->
4  <!-- Displaying the MailingList database. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Search Results</title>
9          <style type = "text/css">
10             table { background-color: lightblue;
11                     border: 1px solid gray;
12                     border-collapse: collapse; }
13             th, td { padding: 5px; border: 1px solid gray; }
14             tr:nth-child(even) { background-color: white; }
15             tr:first-child { background-color: lightgreen; }
16          </style>
17      </head>
18      <body>
19          <?php
20              // build SELECT query
21              $query = "SELECT * FROM contacts";
22
23              // Connect to MySQL
24              if ( !( $database = mysql_connect( "localhost",
25              "iw3htp", "password" ) ) )
26                  die( "<p>Could not connect to database</p></body></html>" );

```

```

27
28 // open MailingList database
29 if ( !mysql_select_db( "MailingList", $database ) )
30     die( "<p>Could not open MailingList database</p>
31         </body></html>" );
32
33 // query MailingList database
34 if ( !( $result = mysql_query( $query, $database ) ) )
35 {
36     print( "<p>Could not execute query!</p>" );
37     die( mysql_error() . "</body></html>" );
38 } // end if
39 ?><!-- end PHP script -->
40
41 <h1>Mailing List Contacts</h1>
42 <table>
43     <caption>Contacts stored in the database</caption>
44     <tr>
45         <th>ID</th>
46         <th>Last Name</th>
47         <th>First Name</th>
48         <th>E-mail Address</th>
49         <th>Phone Number</th>
50         <th>Book</th>
51         <th>Operating System</th>
52     </tr>

```

```

53 <?php
54 // fetch each record in result set
55 for ( $counter = 0; $row = mysql_fetch_row( $result );
56     ++$counter )
57 {
58     // build table to display results
59     print( "<tr>" );
60
61     foreach ( $row as $key => $value )
62         print( "<td>$value</td>" );
63
64     print( "</tr>" );
65 } // end for
66
67 mysql_close( $database );
68 ?><!-- end PHP script -->
69 </table>
70 </body>
71 </html>

```



ID	Last Name	First Name	E-mail Address	Phone Number	Book	Operating System
1	Black	Sue	sue@bug2bug.com	(123) 456-7890	Internet and WWW How to Program	Windows