

Lab 9: Authenticating User Using PHP Session

Login

- Logging in / logging on / signing in/ signing on – process by which an individual gains access to a computer system by identifying and authenticating themselves
- User credentials normally some form of 'username' and a matching 'password'
- These credentials referred to as a login/logon/sign-in/sign-on
- When access is no longer needed, the user can log out/log off/sign out/sign off

Building Login Page

- Create new web folder and name it as **project1** in folder **www**.
- Create folder **css** and **image** under folder **project1**
- In this lab we will use lab practice to create login page to authenticate the valid user prior to enter **project1** application. The valid user will be redirected to **project1** home page. If the user doesn't have an account, he/she can make new registration to login into the system.
- The users are categorised in two level: 1=admin, 2=user1. The default level for new registrations is 2.
- The menu options will be displayed based on user level.
- We will use all pages which created in **assignment 1**. You need to copy all pages (together with images & css) →(**index.html, photoGallery.html & currentAcademic.html**) to **project1** folder.
- Change (rename) all HTML file extension to .php (**index.html** → **index.php**, **photoGallery.html** → **photoGallery.php**, **aboutUs.html** → **aboutUs.php**)

Step 1 –Creating the Database Table

- Create database name it as **demo**.
- Create a relevant table inside your MySQL database
- For example: create a **users** table (to store username, userlevel and password)

```
CREATE TABLE users (  
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    username VARCHAR(50) NOT NULL UNIQUE,  
    userpassword VARCHAR(255) NOT NULL,  
    userlevel INT NOT NULL,  
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

- An auto-updated column is automatically updated to the current timestamp when the value of any other column in the row is changed from its current value

Step 2 – Creating the Config File

- After creating the table, create PHP script () to connect to the MySQL database server
- Lets create a file named 'dbConnect.php' with the following code:

```
<?php  
  
/* Database credentials. Assuming you are running MySQL  
server with default setting (user 'root' with no password) */  
define('DB_SERVER', 'localhost');  
define('DB_USERNAME', 'root');  
define('DB_PASSWORD', '');  
define('DB_NAME', 'demo');  
  
/* Attempt to connect to MySQL database */  
$dbCon = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);  
  
// Check connection  
if ($dbCon === false){  
    die("ERROR: Could not connect. " . mysqli_connect_error());  
}  
  
?>
```

Step 3 – Creating the Login Form

- Create another PHP file 'login.php'
- After user successfully login, it will create a session with the server and redirect to home page.
- The following is the example code

```
1  <?php
2  // Include config file
3  require_once("dbConnect.php");
4
5  // Define variables and initialize with empty values
6  $username = $password = ""; $message = "";
7
8  // Processing form data when form is submitted
9  if ($_SERVER["REQUEST_METHOD"] == "POST") {
10     session_start();//start a session
11     // Validate username
12     $username = trim($_POST["username"]);
13     $password = md5(trim($_POST["password"]));
14     // Prepare a select statement
15     $sql = "SELECT userlevel FROM users WHERE username = '$username'
16           and userpassword = '$password' ";
17
18     $result = mysqli_query($dbCon,$sql); //execute SQL statement
19     if (!$result)
20         die("Database access failed: " . mysqli_error($dbCon));
21
22     // get number of rows returned
23     $rows = mysqli_num_rows($result);
24     if ($rows) { //correct username & password
25         $row = mysqli_fetch_array($result);
26         $_SESSION['loggedin'] = true;
27         $_SESSION['username'] = $username;
28         $_SESSION['userlevel'] = $row['userlevel'];
29         header("location: index.php");
30     } else {
31         $message = 'Wrong Username & Password';
32     }
33     // Close statement
34     mysqli_close($dbCon);
35 }
36 ?>
```

Lab 9: Authenticating User Using PHP Session

```
37 <!DOCTYPE html>
38 <html lang="en">
39   <head>
40     <meta charset="UTF-8">
41     <title>Sign In</title>
42     <link rel="stylesheet" href="css/bootstrap.css">
43     <style type="text/css">
44       body { font: 14px sans-serif; }
45       .message { color: red;}
46     </style>
47   </head>
48   <body>
49     <div class="wrapper">
50       <h2>Login Page</h2>
51       <p>Please fill this form to access the system.</p>
52       <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
53         <div class="message"><?php if($message!=""){echo $message;}?></div>
54         <div class="form-group">
55           <label>Username</label>
56           <input type="text" name="username" class="form-control" required>
57         </div>
58         <div class="form-group">
59           <label>Password</label>
60           <input type="password" name="password" class="form-control" required>
61         </div>
62         <div class="form-group">
63           <input type="submit" class="btn btn-primary" value="Submit">
64           <input type="reset" class="btn btn-secondary" value="Reset">
65         </div>
66       </form><br>
67       <p>Don't have an account? <a href="register.php">Register Here</a>.</p>
68     </div>
69   </body>
70 </html>
```

Step 4 – Creating the Registration Form

- Create another PHP file 'register.php' with following example code.

Lab 9: Authenticating User Using PHP Session

```
1  <?php
2  // Include dbConnect.php file
3  require_once("dbConnect.php");
4  // Define variables and initialize with empty values
5  $username = $password = $confirm_password = $userlevel = "";
6  $username_err = $password_err = $confirm_password_err = "";
7
8  // Processing form data when form is submitted
9  if ($_SERVER["REQUEST_METHOD"] == "POST"){
10     // Validate username
11     if (empty(trim($_POST["username"]))) {
12         $username_err = "Please enter a username.";
13     } else {
14         // Prepare a select statement
15         $sql = "SELECT id FROM users WHERE username = ?";
16         if ($stmt = mysqli_prepare($dbCon, $sql)){
17             // Bind variables to the prepared statement as parameters
18             mysqli_stmt_bind_param($stmt, "s", $param_username);
19
20             // set parameters
21             $param_username = trim($_POST["username"]);
22             // Attempt to execute the prepared statement
23             if (mysqli_stmt_execute($stmt)){
24                 /* store result */
25                 mysqli_stmt_store_result($stmt);
26                 if (mysqli_stmt_num_rows($stmt) == 1){
27                     $username_err = "This username is already taken.";
28                     $username = trim($_POST["username"]);
29                 } else {
30                     $username = trim($_POST["username"]);
31                 }
32             }
33             else {
34                 echo "Oops! Something went wrong. Please try again later.";
35             }
36         }
37         // Close statement
38         mysqli_stmt_close($stmt);
39     }
40
41     // Validate password
42     if (empty(trim($_POST["password"]))) {
43         $password_err = "Please enter a password.";
44     } elseif (strlen(trim($_POST["password"])) < 6){
45         $password_err = "Password must have at least 6 characters.";
46     } else {
47         $password = trim($_POST["password"]);
48     }
49     // validate confirm password
50     if (empty(trim($_POST["confirm_password"]))) {
51         $confirm_password_err = "Please enter confirm password.";
52     } else {
53         $confirm_password = trim($_POST["confirm_password"]);
54         if (empty($password_err) && ($password != $confirm_password)){
55             $confirm_password_err = "Password & Confirm Password did not match.";
56         }
57     }
58 }
```

Lab 9: Authenticating User Using PHP Session

```
57 // Check input errors before inserting in database
58 if (empty($username_err) && empty($password_err) && empty($confirm_password_err)){
59     // Prepare an insert statement
60     $sql = "INSERT INTO users (username, userpassword, userlevel) VALUES (?, ?, ?) ";
61
62     if ($stmt = mysqli_prepare($dbCon, $sql)){
63         // Bind variables to the prepared statement as parameters
64         mysqli_stmt_bind_param($stmt, "sss", $param_username, $param_password, $param_userlevel);
65         // set parameters
66         $param_username = $username;
67         $param_password = md5($password); // encrypt a password
68         $param_userlevel = trim($_POST["userlevel"]);
69
70         // Attempt to execute the prepared statement
71         if (mysqli_stmt_execute($stmt)){
72             echo "<script> alert('Successfully created an account'); </script>";
73             // Redirect to login page
74             echo "<script> location.href='login.php'; </script>";
75         } else {
76             echo "Something went wrong. Please try again later.";
77         }
78     }
79     // Close statement
80     mysqli_stmt_close($stmt);
81 }
82 // Close connection
83 mysqli_close($dbCon);
84 }
85 ?>
86
```

Lab 9: Authenticating User Using PHP Session

```
87 <!DOCTYPE html>
88 <html>
89   <head>
90     <title>Sign Up</title>
91     <link rel="stylesheet" href="css/bootstrap.css">
92     <style type="text/css">
93       body { font: 14px sans-serif; }
94       .message { color: red; }
95     </style>
96   </head>
97
98   <body>
99     <div class="wrapper">
100       <h2>Sign Up</h2>
101       <p>Please fill this form to create an account.</p>
102       <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
103         <div class="form-group">
104           <label>Username</label>
105           <input type="text" name="username" class="form-control" value="<?php echo $username; ?>">
106           <span class="message"><?php echo $username_err; ?></span>
107         </div>
108
109         <div class="form-group">
110           <label>Password</label>
111           <input type="password" name="password" class="form-control" value="<?php echo $password; ?>">
112           <span class="message"><?php echo $password_err; ?></span>
113         </div>
114
115         <div class="form-group">
116           <label>Confirm Password</label>
117           <input type="password" name="confirm_password" class="form-control"
118             value="<?php echo $confirm_password; ?>">
119           <span class="message"><?php echo $confirm_password_err; ?></span>
120         </div>
121
122         <div class="form-group">
123           <input type="submit" class="btn btn-primary" value="Submit">
124           <input type="reset" class="btn btn-secondary" value="Reset">
125         </div>
126         <input type="hidden" name="userlevel" value="2">
127       </form>
128     </div>
129   </body>
```

Step 5 – Creating the Logout page

- Create another PHP file 'logout.php'
- After user click logout, it will delete a session with the server and redirect to login page.
- The following is the example code

```
1  <?php
2  // Initialize the session
3  session_start();
4
5  // Unset all of the session variables
6  $_SESSION = array();
7
8  // Destroy the session.
9  session_destroy();
10
11 // Redirect to login page
12 header("location: login.php");
13 exit;
14 ?>
```

Step 6 – Create menu options based on user category (level)

- Create 'menuAdmin.php' with the following code.

```
1 <ul class="navbar navbar-light bg-light navbar-left">
2   <li><a href="index.php">Home</a></li>
3   <li><a href="photoGallery.php">Photo Gallery</a></li>
4   <li><a href="aboutUs.php">About US</a></li>
5   <li><a href="logout.php">Logout</a></li>
6 </ul>
```

- Create 'menuUser1.php' with the following code.

```
1 <ul class="navbar navbar-light bg-light navbar-left">
2   <li><a href="index.php">Home</a></li>
3   <li><a href="photoGallery.php">Photo Gallery</a></li>
4   <li><a href="logout.php">Logout</a></li>
5 </ul>
```

Note: you can more menu based on user level.

- Modify 'index.php', 'photoGallery.php', and 'aboutUs.php' to check whether the user is logged in, if not then redirect him to login page.
- Open index.php and type the following code

```
1 <?php
2   session_start();
3   // Check if the user is logged in, if not then redirect him to login page
4   if (!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true){
5       header("location: login.php");
6       exit;
7   }
8   $usrlevel = $_SESSION["userlevel"];
9   $usrname = $_SESSION["username"];
10  ?>
11  <!DOCTYPE html>
12  <html lang="en">
13  <head>
14      <meta charset="UTF-8">
15      <meta http-equiv="X-UA-Compatible" content="IE=edge">
16      <meta name="viewport" content="width=device-width, initial-scale=1.0">
17      <title>Home Page</title>
18      <link rel="stylesheet" href="css/bootstrap.css">
19  </head>
20  <body>
21      <?php
22          switch($usrlevel) {
23              case 1:
24                  include('menuAdmin.php');
25                  break;
26              case 2:
27                  include('menuUser1.php');
28                  break;
29          }
30      ?>
31      <h1>Welcome <?php echo $usrname; ?> to My Home Page</h1>
32      <p>This is my website</p>
33      <p></p>
35  </body>
36  </html>
```

- Do as above steps for file 'photoGallery.php', and 'aboutUs.php'.
- For session authentication and menu option, write code in each page as shown in the rectangle
- Change userlevel=1 in table **Users** if you want to make that user as admin.