**Project 02\_08\_01**

In this Project we will build a Web page that allows visitors to your site to sign a guest book that is saved to a database. The document will also display the entries in the guest book.

 

1. Create a new file called ***SignGuestBook.php***. Scaffold a basic HTML code layout into it. Set the <title> to ***SignGuestBook.php***. Make sure to have the ***modernizr*** <script> linked in.
2. Inside the ***<body>*** element, place the following code for the form:  
   <body>:  
    ***<h1>Sign Guest Book</h1>  
    <?php  
    $hostname = "localhost";  
    $username = "root";  
    $password = "";  
    $DBName = "guestbook";  
    $tablename = "visitors";  
    $firstName = "";  
    $lastName = "";  
    $formErrorCount = 0;  
    ?>  
    <form action="SignGuestBook.php" method="post">  
    <p><strong>First Name: </strong><br>  
    <input type="text" name="firstName" value="<?php   
    echo $firstName; ?>"></p>  
    <p><strong>Last Name: </strong><br>  
    <input type="text" name="lastName" value="<?php echo   
    $lastName; ?>"></p>  
    <p><input type="submit" name="submit"   
    value="Submit"></p>  
    </form>***</body>  
   Perform a server/browser test.
3. Now let’s check to see if the form was submitted, as follows:  
    $formErrorCount = 0;  
    ***if (isset($\_POST['submit'])) {  
     
    }***Perform a server/browser test. The form should show. When Submit is clicked, the form should disappear.
4. Now let’s do a validation test on the input fields as follows:  
    if (isset($\_POST['submit'])) {  
    ***$firstName = stripslashes($\_POST['firstName']);  
    $firstName = trim($firstName);  
    $lastName = stripslashes($\_POST['lastName']);  
    $lastName = trim($lastName);  
    if (empty($firstName) || empty($lastName)) {  
    echo "<p>You must enter your first and last   
    <strong>name</strong>.</p>\n";  
    ++$formErrorCount;  
    }*** }  
   Do a server/browser test. Make sure you test both fields empty, either field empty, and both fields filled in.
5. We will be connecting to the database multiple times during this application, so let’s build the connect into a ***function*** we can reuse. At the top of the script, add the following function definition:  
    <?php  
    ***function connectToDB($hostname, $username, $password) {  
    $DBConnect = mysqli\_connect($hostname, $username,   
    $password);  
    if (!$DBConnect) {  
    echo "<p>Connection error: " . mysqli\_connect\_error()   
    . "</p>\n";  
    }  
    return $DBConnect;  
    }***
6. Now let’s call the database connection function if our form has no errors. Place the following code just before the closing PHP delimiter:  
    if (empty($firstName) === 0 || empty($lastName)) {  
    echo "<p>You must enter your first and last   
    <strong>name</strong>.</p>\n";  
    ++$formErrorCount;  
    }  
    ***if ($formErrorCount === 0) {  
    $DBConnect = connectToDB($hostname, $username,   
    $password);  
    if ($DBConnect) {  
    echo "<p>Connection successful!</p>\n";  
    mysqli\_close($DBConnect);  
    }  
    }***Do a server/browser test, once again checking all input possibilities.
7. We will be selecting the database multiple times during this application, so let’s build the select into a ***function*** we can reuse. Bellow the ***connectToDB()*** function, add the following function definition:  
    ***function selectDB($DBConnect, $DBName) {  
    $success = mysqli\_select\_db($DBConnect, $DBName);  
    if ($success) {  
    echo "<p>Successfully selected the \"$DBName\"   
    database.</p>\n";  
    }  
    else {  
    echo "<p>Could not select the \"$DBName\" database:   
    " .   
    mysqli\_error($DBConnect) . ", creating it.</p>\n";  
    $sql = "CREATE DATABASE $DBName";  
    if (mysqli\_query($DBConnect, $sql)) {  
    echo "<p>Successfully created the \"$DBName\"   
    database.</p>\n";  
    $success = mysqli\_select\_db($DBConnect,   
    $DBName);  
    if ($success) {  
    echo "<p>Successfully selected the \"$DBName\"   
    database.</p>\n";  
    }  
    }  
    else {  
    echo "<p>Could not create the \"$DBName\"   
    database: " .   
    mysqli\_error($DBConnect) . "</p>\n";  
    }  
    }  
    return $success;  
    }***Do a server/browser test, once again checking all input possibilities. Check in MySQL Monitor that the database was created.
8. Now let’s call the database select function, surrounding the successful connection ***echo*** statement:  
    if ($DBConnect) {  
    ***if (selectDB($DBConnect, $DBName)) {*** echo "<p>Connection successful!</p>\n";  
    ***}*** mysqli\_close($DBConnect);  
    }  
   Do a server/browser test, once again checking all input possibilities.
9. We will be checking to see if the table exists multiple times during this application, and building it if it doesn’t, so let’s build this into a ***function*** we can reuse. Bellow the ***selectDB()*** function, add the following function definition:  
   ***function createTable($DBConnect, $tablename) {  
    $success = false;  
    $sql = "SHOW TABLES LIKE '$tablename'";  
    $result = mysqli\_query($DBConnect, $sql);  
    if (mysqli\_num\_rows($result) === 0) {  
    echo "The <strong>$tablename</strong> table does   
    not exist, creating table.<br>\n";  
    $sql = "CREATE TABLE $tablename (countID SMALLINT   
    NOT NULL AUTO\_INCREMENT PRIMARY KEY,   
    lastName VARCHAR(40), firstName   
    VARCHAR(40))";  
    $result = mysqli\_query($DBConnect, $sql);  
    if ($result === false) {  
    $success = false;  
    echo "<p>Unable to create the $tablename   
    table.</p>";  
    echo "<p>Error code " . mysqli\_errno($DBConnect) .   
    ": " . mysqli\_error($DBConnect) . "</p>";  
    }  
    else {  
    $success = true;  
    echo "<p>Successfully created the $tablename   
    table.</p>";  
    }  
    }  
    else {  
    $success = true;|  
    echo "The $tablename table already exists.<br>\n";  
    }  
    return $success;  
    }***Do a server/browser test, once again checking all input possibilities.
10. Now let’s call the create table function, surrounding the successful connection ***echo*** statement:  
     if (selectDB($DBConnect, $DBName)) {  
     ***if (createTable($DBConnect, $tablename)) {***  
     echo "<p>Connection successful!</p>\n";  
     ***}***  
     }  
      
    Do a server/browser test, once again checking all input possibilities. Check in MySQL Monitor that the table exists and number of rows.
11. Now it is time to add data to the table. Enter the following code after the successful connection ***echo*** statement:  
     echo "<p>Connection successful!</p>\n";  
     ***$sql = "INSERT INTO $tablename   
     VALUES(NULL, '$lastName',   
     '$firstName')";  
     $result = mysqli\_query($DBConnect, $sql);  
     if ($result === false) {  
     echo "<p>Unable to execute the   
     query.</p>";  
     echo "<p>Error code " .   
     mysqli\_errno($DBConnect) . ": " .   
     mysqli\_error($DBConnect) . "</p>";  
     }  
     else {  
     echo "<h3>Thank you for signing our guest   
     book!</h3>";  
     $firstName = "";  
     $lastName = "";  
     }***  
    Do a server/browser test, once again checking all input possibilities. Check in MySQL Monitor that the data exists and number of rows.
12. We would like to display the Visitors Log at the bottom of the page, after the form. We will also use the functions we built to get our database connection, and our table selected or created. We will build another PHP script below the final ***</form>*** tag as follows:  
     </form> ***<?php  
     $DBConnect = connectToDB($hostname, $username,   
     $password);  
     if ($DBConnect) {  
     if (selectDB($DBConnect, $DBName)) {  
     if (createTable($DBConnect, $tablename)) {  
     echo "<p>Connection successful!</p>\n";  
     echo "<h2>Visitors Log</h2>";  
     }  
     }  
     mysqli\_close($DBConnect);  
     }  
     ?>***Do a server/browser test.
13. We need to populate the Visitor’s Log with data. We will use some of the functionality we have learned to retrieve data into indexed arrays and display that in the log in a <table>:  
     echo "<h2>Visitors Log</h2>";  
     ***$sql = "SELECT \* FROM $tablename";  
     $result = mysqli\_query($DBConnect, $sql);  
     if (mysqli\_num\_rows($result) == 0) {  
     echo "<p>There are no entries in the quest   
     book!</p>";  
     }  
     else {  
     echo "<table width='60%' border='1'>";  
     echo "<tr>";  
     echo "<th>Visitor</th>";  
     echo "<th>First Name</th>";  
     echo "<th>Last Name</th>";  
     echo "</tr>";  
     while ($row = mysqli\_fetch\_row($result)) {  
     echo "<tr>";  
     echo "<td width='10%' style='text-align:   
     center'>$row[0]</td>";  
     echo "<td>$row[1]</td>";  
     echo "<td>$row[2]</td>";  
     echo "</tr>";  
     }  
     echo "</table>";  
     mysqli\_free\_result($result);  
     }***Do a server/browser test.
14. Now we can get rid of all of our debug messages. Comment out all of the debug ***echo*** statements from the script. Be sure to leave the following ones:  
     ***echo "<p>Connection error: " . mysqli\_error()   
     . "</p>\n";  
     echo "<p>You must enter your first and last   
     <strong>name</strong>.</p>\n";  
     echo "<h3>Thank you for signing our guest   
     book!</h3>";***Do a server/browser test.