**Inserting form data in database**

Last lesson, you connected to a database and used PHP and SQL to query that database. This lesson you will link up everything.

* Create a simple form using the following code:

<!-- include the encoding type of the form using the enctype attribute when attaching files-->

<form method="POST" action="upload.php" enctype="multipart/form-data">

<label for="name">Name</label>

<input name="name" id="name"><br>

<label for="address">Address</label>

<input name="address" id ="address"><br>

<label for="Suburb">Suburb</label>

<input name="suburb" id ="suburb"><br>

<label for="image">Picture</label>

<input type="file" name="choosefile" id="image"><br>

<input type="submit" name="submit" value="submit">

</form>

You can style the form up to make it more user friendly.

* Set up a database and table

Create or use an existing database to store the form data. In the example script below ‘test’ is the name of the database and ‘image’ the name of the table. Fields for the table should be id (primary key), filename, name, address and suburb.

Apply appropriate data types to each of these fields.

* Process the data and insert it into a database using the following code

<?php

$msg = "";

if (isset($\_POST['submit'])) {

$filename = $\_FILES["choosefile"]["name"];

$tempname = $\_FILES["choosefile"]["tmp\_name"];

$folder = "image/".$filename;

$name = $\_POST['name'];

$address = $\_POST['address'];

$suburb = $\_POST['suburb'];

// connect with the database

$db = mysqli\_connect("localhost", "root", "", "test");

// query to insert the submitted data

$sql = "INSERT INTO image (filename,name,address,suburb) VALUES ('$filename','$name','$address','$suburb')";

// function to execute above query

mysqli\_query($db, $sql);

// Add the image to the folder

if (move\_uploaded\_file($tempname, $folder)) {

$msg = "Image uploaded successfully";

}else{

$msg = "Failed to upload image";

}

}

echo $msg;

?>

* Query the database to display the required data

<?php

if (!isset($\_POST["submit"])) {

?>

<body>

<div class="wrapper">

<form action="<?php echo $\_SERVER['PHP\_SELF']; ?>" method="post">

<p>

<label for="burb">Suburb: </label>

<input type="text" name="suburb" id="burb"></p>

<p><input type="submit" name="submit"></p>

</form>

<?php

} else {

?>

<table>

<tr>

<th>Photo</th>

<th>Name</th>

<th>Address</th>

<th>Suburb</th>

</tr>

<?php

$db = mysqli\_connect("localhost", "root", "", "test");

$suburb = mysqli\_real\_escape\_string($db,$\_POST["suburb"]);

$result = mysqli\_query($db, "SELECT \* FROM image where suburb = '$suburb'");

$intCount = mysqli\_num\_rows($result);

if($intCount == 0){

echo "There are no records for this search";

}

else{

echo "<p>Detail:</p>";

while($pic = mysqli\_fetch\_array($result)){

echo "<tr><td><img src=image/" . $pic['filename']." width=75 height=50></td><td>".$pic['name']."</td><td>".$pic['address']."</td><td>".$pic['suburb']."</td></tr>";

}

}

?>

</table>

<?php

}

?>

</div>

Update the styles for usability. Add a back button to make testing easier.

**Login form**

You will now build on your knowledge of sessions to set up a script that check username and password. If the user gets it wrong three times then they are locked out – for this example it’s only 30 seconds. In the real world, it would be a life sentence.

Once logged in the user will have access to a separate admin section and stay logged in until they choose to log out.

First create a database, then add the following 2 tables:

CREATE TABLE `loginlogs` (

`id` int(11) NOT NULL,

`IpAddress` varchar(100) NOT NULL,

`TryTime` bigint(20) NOT NULL

)

CREATE TABLE `user` (

`id` int(11) NOT NULL,

`username` varchar(50) NOT NULL,

`password` varchar(100) NOT NULL

)

Next, set up your connection to your database in a separate file conn.php. You will link this to every page needed for this application.

Now set up the login and processing page:

<?php

session\_start();

include\_once('conn.php');

$msg='';

if(isset($\_POST['submit'])){

$time=time()-30;

$ip\_address=getIpAddr();

// Getting total count of hits on the basis of IP

$query=mysqli\_query($con,"select count(\*) as total\_count from loginlogs where TryTime > $time and IpAddress='$ip\_address'");

$check\_login\_row=mysqli\_fetch\_assoc($query);

$total\_count=$check\_login\_row['total\_count'];

//Checking if the attempt 3, or you can set the number of attempts here. For now we taking only 3 fail attempts

if($total\_count==3){

$msg="To many failed login attempts. Please login after 30 sec";

}else{

//Getting Post Values

$username=$\_POST['username'];

$password=$\_POST['password'];

// Coding for login

$res=mysqli\_query($con,"select \* from user where username='$username'");

while($row = mysqli\_fetch\_array($res)){

$pass = $row['password'];

}

//need to use password\_hash() to populate the database with encrypted password

if(password\_verify($password,$pass)){

$\_SESSION['IS\_LOGIN']='yes';

mysqli\_query($con,"delete from loginlogs where IpAddress='$ip\_address'");

header("Location:admin.php");}

else{

$total\_count++;

$rem\_attm=3-$total\_count;

if($rem\_attm==0){

$msg="To many failed login attempts. Please login after 30 sec";

}else{

$msg="Please enter valid login details.<br/>$rem\_attm attempts remaining";

}

$try\_time=time();

mysqli\_query($con,"insert into loginlogs(IpAddress,TryTime) values('$ip\_address','$try\_time')");

}

}

}

// Getting IP Address

function getIpAddr(){

if (!empty($\_SERVER['HTTP\_CLIENT\_IP'])){

$ipAddr=$\_SERVER['HTTP\_CLIENT\_IP'];

}elseif (!empty($\_SERVER['HTTP\_X\_FORWARDED\_FOR'])){

$ipAddr=$\_SERVER['HTTP\_X\_FORWARDED\_FOR'];

}else{

$ipAddr=$\_SERVER['REMOTE\_ADDR'];

}

return $ipAddr;

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="robots" content="noindex, nofollow">

<title>Login Form</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link href="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">

<style type="text/css">

body {margin: 0;padding: 0;background-color: #b81e17;height: 100vh;}

#login .container #login-row #login-column #login-box {margin-top: 60px;max-width: 600px;height: 320px;border: 1px solid #9C9C9C;background-color: #EAEAEA;}

#login .container #login-row #login-column #login-box #login-form {padding: 40px;}

#result{color:red;}

</style>

<script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<script src="//maxcdn.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"></script>

</head>

<body>

<div id="login">

<h3 class="text-center text-white pt-5">Login form</h3>

<div class="container">

<div id="login-row" class="row justify-content-center align-items-center">

<div id="login-column" class="col-md-6">

<div id="login-box" class="col-md-12">

<form id="login-form" class="form" method="post">

<div class="form-group">

<label for="username" class="text-info">Username:</label><br>

<input type="text" name="username" id="username" class="form-control" required>

</div>

<div class="form-group">

<label for="password" class="text-info">Password:</label><br>

<input type="password" name="password" id="password" class="form-control" required>

</div>

<div class="form-group">

<input type="submit" name="submit" class="btn btn-info btn-md" value="Submit">

</div>

<div id="result"><?php echo $msg?></div>

</form>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

Once a user has logged in they will now have access to the admin section. Create the following and save as admin.php

<?php

session\_start();

if(!isset($\_SESSION['IS\_LOGIN'])){

?>

<script>

window.location.href='index.php';

</script>

<?php

}else{

echo "Welcome";

}

?> ||

<a href="logout.php">Logout</a>

Test this by trying to access the admin section directly – you should not be able to. Use the following for the logout.php

<?php

session\_start();

unset($\_SESSION['IS\_LOGIN']);

session\_destroy();

?>

<script>

window.location.href='login.php';

</script>