# PHP and MySQL

## EXERCISE 3: PHOTO ALBUM SERVER VERSIONS B AND C

Take a look at the file you should have called **photoalbum-server-version-a.php**. This is a PHP script which prints out the photo album page based on an array of photograph data. You should be able to understand that we could do a database query to get the array of photograph data and print out a photo album page from that. To do so, we will need a table of photographs in the database. Rather than create the table using PHPMyAdmin and insert the required rows manually, download the resource archive **photographs-db.zip** from moodle and extract the files. Upload both to the server. There is a script called **create-photograph-db.php** and a file containing the data for the Beatles photographs. View **create-photograph-db.php** in a web browser. This will create the required table. Using PHPMyAdmin go and look at the table “photographs” to make sure it is there. Then add a line to **dbtest.php** to display it in an html table.

Now make a copy of your file **photoalbum-server-version-a.php** and call it **photoalbum-server-version-b.php**. Open this copy in NotePad++ and make the following changes:

1. Add a **require\_once** include statement at the top, just like the one in **dbtest.php**. This makes the **dbutils.php** functions available in it.
2. Delete all the lines initialising the hard coded arrays and replace them with the code to query the database – see below.

|  |
| --- |
| ~~$john['image'] = "b1.png";~~  ~~$john['name'] = "John";~~  ~~$john['description'] = "<p>This is John Lennon. He was one of the Beatles.</p><p>He was one of the two main song-writers.</p>";~~  ~~$paul['image'] = "b2.png";~~  ~~$paul['name'] = "Paul";~~  ~~$paul['description'] = "<p>This is Paul McCartney. He was also one of the Beatles.</p><p>He was the other main song-writer.</p>";~~  ~~$george['image'] = "b3.png";~~  ~~$george['name'] = "George";~~  ~~$george['description'] = "<p>This is George Harrison. He was also one of the Beatles.</p>";~~  ~~$ringo['image'] = "b4.png";~~  ~~$ringo['name'] = "Ringo";~~  ~~$ringo['description'] = "<p>This is Ringo Starr. He was also one of the Beatles.</p>";~~  ~~$rows[0] = $john;~~  ~~$rows[1] = $paul;~~  ~~$rows[2] = $george;~~  ~~$rows[3] = $ringo;~~  $pdo = connect();  $sql = "SELECT \* FROM `photographs` ORDER BY `photoid`";  $stmt = $pdo->query( $sql);  // get all rows in an array  $rows = $stmt->fetchAll(PDO::FETCH\_ASSOC); |

Now view **photoalbum-server-version-b.php** in a web browser. You should get a photo album page with the four Beatles pictures. Add a fifth picture of whatever you want by adding a row to the photographs table in PHPMyAdmin and uploading the corresponding image file to the **images** folder.

Although the result may look just the same, there is an important difference. To change the photographs we don’t have to edit the PHP script or any HTML. We just have to edit the database and the image files available in the **images** folder. That is important, because we know we can write PHP scripts which modify the database, and we know we can write PHP scripts which get data from the user.

The simplest modification is to delete photographs, rather than to add them, so we will do that first.

### Deleting a picture from the photo album

To delete a picture we actually need to do two things: delete the row from the database and delete the image file from the images folder. Each photograph has a unique ID number and we would need to use this number to identify which photo to delete.

Here is a function that would do it. Because we will want to use this in several versions of our photo album scripts, we will not add it to the main script but put it in another file of library functions. Because it is specific to the photo album application, we won’t put it in **dbutils.php**, instead we will create a new file called **photoalbum-common.php**. Copy and paste the following function into NotePad++ and then save it with the name **photoalbum-common.php**.

|  |
| --- |
| <?php  function deletePhotograph( $pdo, $deletionid) {  $errorMessage = "";  // retrieve name of image file so we can delete it  $stmt = $pdo->prepare("SELECT `image` FROM `photographs` WHERE `photoid`=?");  $stmt->execute( array( $deletionid));  $rows = $stmt->fetchAll(PDO::FETCH\_ASSOC);  if ( count( $rows) == 1) {  // delete file  unlink( "images/".$rows[0]['image']);  // delete database record  $stmt = $pdo->prepare("DELETE FROM `photographs` WHERE `photoid`=?");  $stmt->execute( array( $deletionid));  $affected\_rows = $stmt->rowCount();  } else if (count( $rows) > 1) {  $errorMessage .= "ID matches more than one record. ";  } else {  $errorMessage .= "ID not found: nothing to delete. ";  }  return $errorMessage;  }  ?> |

Upload **photoalbum-common.php** to the server into the main folder (*not* the lib folder). This function does three things:

1. It performs a SELECT query on the database to find the record with the given ID number. This is to find out the name of the image file to delete.
2. It deletes the file in the images folder which has the name from the database.
3. It then deletes the row from the database table which has the given ID number.

It returns either an empty string “” or an error message if the photograph record could not be found.

### Photo Album Server Version C

Save a copy of your **photoalbum-server-version-b.php** file with the name **photoalbum-server-version-c.php**. The first thing you need to do is to add a require\_once command to include the **photoalbum-common.php** file. Add this to the top of the file where dbutils.php is included, with the following line:

require\_once("photoalbum-common.php");

The ***deletePhotograph*** function is now available to use in **photoalbum-server-version-c.php**. To try it out do the following:

Use PHPMyAdmin to examine the photographs table and see which photograph has photoid ‘1’ and what its image file name is. If for some reason you don’t have a photograph with photoid ‘1’ just use a different number. Add the following line to **photoalbum-server-version-c.php** immediately after the **$pdo = connect();** line.

deletePhotograph( $pdo, 1);

View **photoalbum-server-version-c.php** in a web browser. The corresponding photograph should have disappeared. Go back to PHPMyAdmin and refresh the view of the table to see that the row has gone from the database. Then examine the contents of the images folder to see that the image has been deleted.

Deleting a hard-coded photoid from the album is not much use. We want the user to be able to specify a photoid to delete. We can do that in the URL by giving a name and value such as “deletionid=6”. The script can then look for the presence of a deletionid value in the $\_GET array and if it is there, delete the photograph. We can also make use of the fact that the ***deletePhotograph*** function returns an error message if the photograph could not be deleted.

Replace the call to ***deletePhotograph*** you just added to **photoalbum-server-version-c.php** with the following code:

if ( isset( $\_GET['deletionid'])) {

$errorMessage = deletePhotograph( $pdo, $\_GET['deletionid']);

if ( $errorMessage != "") {

print "<div class='errormessage'>$errorMessage</div>\n";

} else {

print "<div class='message'>Image deleted.</div>\n";

}

}

Now you can delete any photograph by adding **“?deletionid=n”** to the URL in the browser address bar, where ***n*** is the id number. Notice what happens if you try to delete the same id number more than once.

Of course what we want to be able to do is click on a link to delete a photograph. To do that, we can change the code in **photoalbum-server-version-c.php** that prints out the photograph information by adding the new line shown in blue below:

|  |
| --- |
| print " <div class='w3-container w3-row w3-col l4'>\n";  print " <div class='w3-container w3-card-8 w3-margin-8'>\n";  print " <div class='w3-container w3-half w3-image'>\n";  print " <a class='example-image-link' data-lightbox='example-1' href='images/".$row['image']."'><img src='images/".$row['image']."' class='example-image w3-circle'></a>\n";  print " <div class='w3-title w3-text-purple'>".$row['name']."</div>\n";  print " </div>\n";  print " <div class='w3-container w3-half'>\n";  print $row['description']."\n";  **print " <a href='?deletionid=".$row['photoid']."' class='link'>(Delete)</a>";**  print " </div>\n";  print " </div>\n";  print " </div>\n"; |

Now every image has a delete link allowing it to be removed. There are some issues with this system of course:

* It is too easy to accidentally delete a photograph: you should need to confirm you want to.
* Anyone who visits the page can delete photographs from your album!

The first of these is left as an extended task, the second is going to be dealt with later.

You may find your photo album is running short of pictures after you have deleted them, and it’s tedious to add them by hand. So the next exercise is to make the script accept uploaded photographs.

|  |
| --- |
| **EXERCISE 3: SERVER PHOTO ALBUM VERSIONS B AND C** |
| ***Required in your report (basic exercise).*** |
| * Clickable links to photo-album-server-version-b.php and photo-album-server-version-c.php * A complete listing of the source code of photo-album-server-version-c.php. * Reflection on the tasks and all activities involved. |
| ***Extended Tasks*** |
| * Fix the problem that there is no confirmation for deleting photographs by making the part of the script that reacts to the presence of the deletionid parameter also check for the presence of another parameter called deletionconfirm. If there is a deletionconfirm parameter, go ahead and delete the image, if not print out a message of the form **“Really delete photograph? Yes No”** where the *yes* is a link which includes both the deletionid value AND a value for deletionconfirm and the *no* is a link with no parameters, just taking you back to the normal page. |
| *Original additional work:*  *Investigate and experiment with any related subject matter that interests you.* |