

Follow on Tutorial - Products

This tutorial is a follow on from the product application we started in lecture 7.

We completed the

- Product List View
- Product View

Make sure you have completed what we did in the lecture before moving on to this challenge.

In this tutorial we will complete

- Product Delete
- Product Update
- Product Create

Product Delete

We're going to add a page to delete a product .

To do this we're going to populate a form with data from our database table using the **id** of the product selected.

When we have deleted we will return to the products.php file

So we'll start by creating our form and adding data to it **from the table**

We'll make the input fields readonly so they can't be amended. Using the input attribute **readonly**

Start by creating the form

```
<h2>Delete Product</h2><br></br>
<form class='form-style' action="deleteProduct.php" method="post">
Product: <input type="text" name="product" value="<?php echo $product;
?>" readonly/>
Description: <input type="text" name="description" value="<?php echo
$product_desc; ?>" readonly/>
Cost: <input type="text" name="cost" value="<?php echo $cost; ?>"
readonly/>
<input type="submit" name="submit" value="Delete" class='button'/>
</form>
```

Save as deleteProduct.php

Add the following php code above the form in the same file

```
<?php
$pid = $_GET['id']; // from link in products.php
include('db.php');
$stmt = $DBH->prepare("SELECT * FROM Products WHERE id= :pid");
$stmt->bindValue(':pid', $pid);
$stmt->execute();
include('errordb.php');
$row = $stmt->fetch(PDO::FETCH_ASSOC);
$product = $row['product_name'];
$product_desc = $row['product_description'];
$cost = $row['cost'];
?>
```

This code will display the product data on page.

Add a few lines to **products.php** file to link to the delete page , directly under the viewProduct link

```
echo "<td>";  
echo "<a href=deleteProduct.php?id=".$row['id']. ">Delete</a>";  
echo "</td>";
```

Test the link in the browser

We need to distinguish between a GET and a POST in the php file. When we click the link in products.php we are doing a **GET**.

When we submit the form we are doing a **POST**

When we see a post we want to delete the row from the table using the DELETE sql method using the data from the form.

We need to add a line in the form to store the id of the row we are going to delete. We do not want to display this id on the form. So we will use the input type 'hidden'

```
<input type="hidden" name="pid" value="<?php echo $pid; ?>" />
```

Our HTML now looks like

```
<form class='form-style' action="deleteProduct.php" method="post">  
Product: <input type="text" name="product" value="<?php echo $product;  
?>" readonly/>  
Description: <input type="text" name="description" value="<?php echo  
$product_desc; ?>" readonly/>  
Cost: <input type="text" name="cost" value="<?php echo $cost; ?>"  
readonly/>  
<input type="hidden" name="pid" value="<?php echo $pid; ?>" />  
<input type="submit" name="submit" value="Delete" class='button' />  
</form>
```

The GET part of our php file now looks like

```
<?php
include('db.php'); //note we moved this line
if ($_GET){
    $pid = $_GET['id'];
    $stmt = $DBH->prepare("SELECT * FROM Products WHERE id=
:pid");
    $stmt->bindValue(':pid', $pid);
    $stmt->execute();
    include('errordb.php');
    $row = $stmt->fetch(PDO::FETCH_ASSOC);
    $product = $row['product_name'];
    $product_desc = $row['product_description'];
    $cost = $row['cost'];
}
?>
```

The POST part of our php file looks like

```
if ($_POST) {
    $pid = $_POST['pid']; // from hidden input field

    $stmt = $DBH->prepare("DELETE FROM Products WHERE id=
:pid");
    $stmt->bindValue(':pid', $pid);
    $stmt->execute();
    include('errordb.php');
    header("Location: products.php");
}
```

Add directly after the `if ($_GET){ }` inside the php tags.

Complete program looks like.

```

1
2 <?php
3 include('header.php');
4 include('db.php'); //note we moved this line
5
6 if ($_GET){
7     $pid = $_GET['id']; // from link in products.php
8     include('db.php');
9     $stmt = $DBH->prepare("SELECT * FROM Products WHERE id= :pid");
10    $stmt->bindValue(':pid', $pid);
11    $stmt->execute();
12    include('errordb.php');
13    $row = $stmt->fetch(PDO::FETCH_ASSOC);
14    $product = $row['product_name'];
15    $product_desc = $row['product_description'];
16    $cost = $row['cost'];
17 }
18 if ($_POST) {
19     $pid = $_POST['pid']; // from hidden input field
20     $product = $_POST['product'];
21     $product_desc = $_POST['description'];
22     $cost = $_POST['cost'];
23
24     $stmt = $DBH->prepare("DELETE FROM Products WHERE id= :pid");
25     $stmt->bindValue(':pid', $pid);
26     $stmt->execute();
27     include('errordb.php');
28     header("Location: products.php");
29 }
30 ?>
31 <h2>Delete Product</h2><br><br>
32 <form class='form-style' action="deleteProduct.php" method="post">
33 Product: <input type="text" name="product" value="<?php echo $product; ?>" readonly/>
34 Description: <input type="text" name="description" value="<?php echo $product_desc; ?>" readonly/>
35 Cost: <input type="text" name="cost" value="<?php echo $cost; ?>" readonly/>
36 <input type="hidden" name="pid" value="<?php echo $pid; ?>" />
37 <input type="submit" name="submit" value="Delete" class='button' />
38 </form>
39 <?php include('footer.php'); ?>

```

Product Update

We will start with making a copy of the deleteProduct.php file and call it updateProduct.php

We will make the necessary changes to make it an update file.

First change the title.

Second we will change the input form to no longer be **readonly**. We will remove these attributes. Change the action in the form. Our html will now look like (changes in red)

```
<form class='form-style' action="updateProduct.php" method="post">
Product: <input type="text" name="product" value="<?php echo $product;
?>" />
Description: <input type="text" name="description" value="<?php echo
$product_desc; ?>" />
Cost: <input type="text" name="cost" value="<?php echo $cost; ?>" />
<input type="hidden" name="pid" value="<?php echo $pid; ?>" />
<input type="submit" name="submit" value="Update" class='button'/>
</form>
</body>
</html>
```

The if(\$_GET) portion of the php will remain the same. The \$_POST portion requires some changes. Change the code to

```
if ($_POST) {
    $pid = $_POST['pid'];
    $product = $_POST['product'];
```

```
$product_desc = $_POST['description'];
$cost = $_POST['cost'];
$stmt = $DBH->prepare(
"UPDATE Products set product_name = :product,
                    product_description = :pd,
                    cost = :cost
                    WHERE id = :pid");
$stmt->bindValue(':pid', $pid);
$stmt->bindValue(':pd', $product_desc);
$stmt->bindValue(':product', $product);
$stmt->bindValue(':cost', $cost);
$stmt->execute();
include('errordb.php');
header("Location: products.php");
}
?>
```

Complete Program looks like


```

1  <?php
2  include('header.php');
3  include('db.php');
4  if ($_GET){
5      $pid = $_GET['id'];
6      $stmt = $DBH->prepare("SELECT * FROM Products WHERE id= :pid");
7      $stmt->bindValue(':pid', $pid);
8      $stmt->execute();
9      include('errordb.php');
10     $row = $stmt->fetch(PDO::FETCH_ASSOC);
11     $product = $row['product_name'];
12     $product_desc = $row['product_description'];
13     $cost = $row['cost'];
14 }
15 if ($_POST) {
16     $pid = $_POST['pid'];
17     $product = $_POST['product'];
18     $product_desc = $_POST['description'];
19     $cost = $_POST['cost'];
20     $stmt = $DBH->prepare("UPDATE Products set product_name = :product,
21                             product_description = :pd,
22                             cost = :cost
23                             WHERE id = :pid");
24     $stmt->bindValue(':pid', $pid);
25     $stmt->bindValue(':pd', $product_desc);
26     $stmt->bindValue(':product', $product);
27     $stmt->bindValue(':cost', $cost);
28     $stmt->execute();
29     include('errordb.php');
30     header("Location: products.php");
31 }
32 ?>

```

```

33 <h2>Update Product</h2>
34 <form class='form-style' action="updateProduct.php" method="post">
35     Product: <input type="text" name="product" value="<?php echo $product; ?>" />
36     Description: <input type="text" name="description" value="<?php echo $product_desc; ?>" />
37     Cost: <input type="text" name="cost" value="<?php echo $cost; ?>" />
38     <input type="hidden" name="pid" value="<?php echo $pid; ?>" />
39     <input type="submit" name="submit" value="Update" class='button' />
40 </form>
41 <?php include 'footer.php'; ?>

```


Now we need to make the update option available from the products list.
Add the link in the **products.php** file

```
echo "<td>";  
echo "<a href=updateProduct.php?id=".$row['id']. ">Edit</a>";  
echo "</td>";
```

Create New Product

We now want the ability to add a new product. Make a copy of the updateProduct.php file call it createProduct.php.

Change the title.

Completely remove the if (\$_GET) { } portion of the php code.

We need to change the sql in the \$_POST section from an UPDATE to an INSERT.

Change

```
$stmt = $DBH->prepare(
    "UPDATE Products set product_name = :product,
    product_description = :pd,
    cost = :cost
    WHERE id = :pid");
```

To

```
$stmt = $DBH->prepare("INSERT into Products
(product_name,product_description,cost)
Values(:product,:pd,:cost)");
```

Remove references to the \$pid in the php post and the input form and bindValue().

Also we need to initialise our database variables at the top of the php file

```
include('db.php');
```

```
$product = "";  
$product_desc = "";  
$cost = 0.00;  
if ($_POST) {
```

Change the value on button on form from Update to Save

Change the action on the form from updateProduct.php to createProduct.php

```
<h2>Create Product</h2><br></br>
```

```
<form class='form-style' action="createProduct.php" method="post">
```

So our php code is now

```

1  <?php
2  include('header.php');
3  include('db.php'); //note we moved this line
4  $product = "";
5  $product_desc = "";
6  $cost = 0.00;
7  if ($_POST) {
8      $product = $_POST['product'];
9      $product_desc = $_POST['description'];
10     $cost = $_POST['cost'];
11     $stmt = $DBH->prepare("INSERT into Products
12                             (product_name,product_description,cost)
13                             Values(:product,:pd,:cost)");
14     $stmt->bindValue(':pd', $product_desc);
15     $stmt->bindValue(':product', $product);
16     $stmt->bindValue(':cost', $cost);
17     $stmt->execute();
18     include('errorodb.php');
19     header("Location: products.php");
20 }
21 ?>
22 <h2>Create Product</h2><br></br>
23 <form class='form-style' action="createProduct.php" method="post">
24     Product: <input type="text" name="product" value="<?php echo $product; ?>" />
25     Description: <input type="text" name="description" value="<?php echo $product_desc; ?>" />
26     Cost: <input type="text" name="cost" value="<?php echo $cost; ?>" />
27     <input type="submit" name="submit" value="Save" class='button' />
28 </form>
29 <?php include('footer.php'); ?>

```

Run it by clicking the “New” option on the menu from index.php

Extra Challenge

Add validation to the updateProduct.php and createProduct.php for the form fields. Do you see an opportunity for an include file ?