**Updating an Order**

Often a customer will place an item in the cart, only to decide to remove it later on. Or instead, the customer may decide to go back and request more (or less) of an individual item. Instead of making the customer replace the entire shopping cart contents, you should have a system that lets the customer modify individual products within the cart.

The Food Store application includes code to allow the customer to review and modify an existing shopping cart. The navigation bar you created earlier contains a link to review the contents of the shopping cart:

<a href="index.php?content=reviewcart"><strong>Review shopping cart</strong></a></td>

This link uses the reviewcart.inc.php file to display the shopping cart contents in the main section of the Web page.

The Review Cart Web page

Let's create this file now.

1. Create the file *reviewcart.inc.php* in the store folder.
2. Open the file with a text editor, and enter the following code:

<?php

echo "<h2>Review your shopping cart contents</h2><br>\n";

$total = 0;

echo "<table width=\"100%\" cellpadding=\"1\" border=\"1\">\n";

echo "<tr><td>Product</td><td>Price</td><td>Quantity</td><td>Total</td><td> </td>\n";

foreach($\_SESSION['cart'] as $prodid => $quantity)

{

$query = "SELECT description, price FROM products WHERE prodid = $prodid";

$result = mysql\_query($query);

$row = mysql\_fetch\_array($result, MYSQL\_ASSOC);

$description = $row['description'];

$price = $row['price'];

$subtotal = $price \* $quantity;

$total += $subtotal;

printf("<tr><td>%s</td><td>%s</td><td>%d</td><td>$%.2f</td>\n",

$description, $price, $quantity, $subtotal);

echo "<td><a href=\"index.php?content=updateitem&id;=$prodid\">Modify</a></td></tr>\n";

}

printf("<tr><td colspan=\"3\"><b>Total</b></td><td>$%.2f</td></tr>\n", $total);

echo "</table>\n";

?>

1. Save the file and exit the text editor.

This code should look familiar to you. It uses the same code as the real-time cart display section on the main Web page, which displays the contents of the shopping cart.

Since the items are indexed using their prodid values, the array values are numerical, but not necessarily in order. This would make iterating through the individual products in the cart impossible using a standard for loop. Instead, we can use our new friend, the *foreach*loop:

foreach($\_SESSION['cart'] as $prodid => $quantity)

The foreach loop retrieves the individual prodid keys and the associated quantities stored as values for each key. After retrieving the product ID and quantity values from the session cookie, the code sends an SQL query to the MySQL database to retrieve the description and price of the product. This allows you to display the product in a form that the customer can understand.

As the code displays each product in the shopping cart, it also produces a link for the customer:

echo "<td><a href=\"index.php?content=updateitem&id;=$prodid\">Modify</a></td></tr>\n";

This link takes the customer to the updateitem.inc.php include file. This allows the customer to update or remove the product in the shopping cart.

The Update item Web page

Let's make this file next:

1. Create a file called *updateitem.inc.php* in the store folder.
2. Open the file in a text editor and add the following code:

<?php

echo "<h2>Update item in cart</h2><br>\n";

$prodid = $\_GET['id'];

$query = "SELECT description, price FROM products where prodid = $prodid";

$result = mysql\_query($query);

$row = mysql\_fetch\_array($result, MYSQL\_ASSOC);

$description = $row['description'];

$price = $row['price'];

$quantity = $\_SESSION['cart'][$prodid];

echo "<form action=\"index.php\" method=\"post\">\n";

echo "<input type=\"hidden\" name=\"content\" value=\"changeitem\">\n";

echo "<input type=\"hidden\" name=\"prodid\" value=\"$prodid\">\n";

echo "<table width=\"100%\" cellpadding=\"1\" border=\"1\">\n";

echo "<tr><td><h3>Product ID</h3></td><td>$prodid</td></tr>\n";

echo "<tr><td><h3>Description</h3></td><td>$description</td></tr>\n";

printf("<tr><td><h3>Price</h3></td><td>$%.2lf</td></tr>\n", $price);

echo "<tr><td><h3>Quantity</h3></td><td><input type=\"text\" name=\"quantity\" value=\"$quantity\"></td></tr>\n";

$total = $price \* $quantity;

printf("<tr><td><h3>Total</h3></td><td>\$%.2lf</td></tr>\n", $total);

echo "</table>\n";

echo "<input type=\"submit\" name=\"button\" value=\"Update\">\n";

echo "<input type=\"submit\" name=\"button\" value=\"Remove Item from cart\">\n";

echo "</form>\n";

?>

1. Save the file and exit the editor.

This PHP code produces a simple HTML form, and populates the form with the information about the product and the current quantity in the shopping cart. You'll notice that there are two submit buttons in this form. One is to update the quantity value and one to remove the item from the cart.

The final piece of code is the changeitem.inc.php file:

1. Create the file *changeitem.inc.php* in the store folder.
2. Open the file in a text editor and add the following code:

<?php

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

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

if ($button == "Update")

{

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

$\_SESSION['cart'][$prodid] = $quantity;

echo "<h2>Item quantity updated in cart</h2>\n";

echo "<a href=\"index.php?content=reviewcart\">Review cart</a>\n";

} else

{

unset($\_SESSION['cart'][$prodid]);

echo "<h2>Item removed from cart</h2>\n";

echo "<a href=\"index.php?content=reviewcart\">Review cart</a>\n";

}

?>

1. Save the file and exit the text editor.

This simply determines which submit button on the updateitem page the customer clicked. Since both buttons use the same HTML variable, all you need to do is check the value of the HTML variable. If the customer clicks the Update button, the quantity value replaces the appropriate session cookie with the new value. If the customer clicks the Remove button, the code deletes the session cookie using the unset() function.

That's quite a lot of code to cover in one lesson! Let's move over to Chapter 5 and review what we did today.