**Showing Images With Product Information**

One of the options in the adminnav.inc.php file produces a select form area using the product categories as entries. This allows the manager to select a category for editing product information. It helps limit the products you display on the Web page. Now it's time to use that information to display your products.

If you remember, the adminnav.inc.php select form *action* attribute points to the admin.php main Web page with the *content* variable set to *editproducts*. This defines the page you need to create to display your product information.

The *editproducts.inc.php* file is responsible for displaying all of the products for a certain category. Let's create that file, then look at how it works:

1. Create a file called *editproducts.inc.php* in the *admin* folder under the store application area.
2. Use a text editor to add the following code to the file:

<?php

if (!isset($\_SESSION['store\_admin']))

{

echo "<h2>Sorry, you have not logged into the system</h2>\n";

echo "<a href=\"admin.php\">Please login</a>\n";

} else

{

echo "<h2>Click on a product to edit it</h2>\n";

$catid = $\_GET['cat'];

if (!isset($\_GET['page']))

$page = 1;

else

$page = $\_GET['page'];

showproducts($catid, $page, "admin.php?content=editproducts", "admin.php?content=updateproduct");

}

?>

That's not very much code, considering what the program's supposed to be doing! It first checks to make sure the Web page visitor is logged in, and then it retrieves two HTML variables: *cat* and *page*. It passes both of these variables to a new function, called showproduts(), along with a couple of URLs. The showproducts() function uses four parameters:

* The category ID from which to display products.
* The page of products to display.
* The URL of the current Web page.
* A URL for the Web page to go to if the visitor clicks on a product.

The Food Store has two situations where you'll need to display products. The store manager will need to list products to edit them, and the store customers will need to browse through products to select them for purchasing. By making the showproducts function as generic as possible, you can use it for both the store back-end and front-end applications (that's the whole point of creating a function).

The *page* value allows you to display only a subset of the total product list. This makes life a little easier for the viewer trying to browse the catalog. Instead of showing all of the products, the visitor will see a group of five products and links to go to the next or previous pages.

|  |
| --- |
| **Note:** If you didn't take the introduction course or if you need to refresh your memory on using paging, there's a link in the Supplementary Material section that explains how this works. |

Now you need to create the showproducts.php function file to handle displaying product data:

1. Create a file called *showproducts.php* in the *mylibrary* folder under the store folder.
2. Open the file with a text editor and enter the following code:

<?php

function showproducts($catid, $page, $currentpage, $newpage)

{

$query = "Select count(prodid) from products where catid = $catid";

$result = mysql\_query($query);

$row = mysql\_fetch\_array($result);

if ($row[0] == 0)

{

echo "<h2><br>Sorry, there are no products in this category</h2>\n";

}

else

{

$thispage = $page;

$totrecords = $row[0];

$recordsperpage = 5;

$offset = ($thispage - 1) \* $recordsperpage;

$totpages = ceil($totrecords / $recordsperpage);

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

echo "<tr><td><h2>Image</h2></td>\n";

echo "<td><h2>Product</h2></td>\n";

echo "<td><h2>Price</h2></td>\n";

echo "<td><h2>Quantity in Stock</h2></td>\n";

echo "<td><h2>Special</h2></td></tr>\n";

$query = "SELECT \* from products WHERE catid=$catid LIMIT $offset,$recordsperpage";

$result = mysql\_query($query);

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

{

$prodid = $row['prodid'];

$description = $row['description'];

$price = $row['price'];

$quantity = $row['quantity'];

$onsale = $row['onsale'];

echo "<tr><td>\n";

echo "<img src=\"showimage.php?id=$prodid\" width=\"80\" height=\"60\">";

echo "</td><td>\n";

echo "<a href=\"$newpage&id;=$prodid\">$description\n";

echo "</td><td>\n";

echo "$" . $price . "\n";

echo "</td><td>\n";

echo $quantity . "\n";

echo "</td><td>\n";

if ($onsale)

echo "On sale!\n";

else

echo "&nbsp;\n";

echo "</td></tr>\n";

}

echo "</table>\n";

// Code to implement paging

if ($thispage > 1)

{

$page = $thispage - 1;

$prevpage = "<a href=\"$currentpage&cat;=$catid&page;=$page\">Previous page</a>";

} else

{

$prevpage = " ";

}

if ($thispage < $totpages)

{

$page = $thispage + 1;

$nextpage = " <a href=\"$currentpage&cat;=$catid&page;=$page\">Next page</a>";

} else

{

$nextpage = " ";

}

if ($totpages > 1)

echo $prevpage . " " . $nextpage;

}

}

?>

1. Save the file and exit the editor.
2. Open the admin.php file in an editor, and add the showproducts.php file as an include file:

<?php

include("/mylibrary/login.php");

include("/mylibrary/getThumb.php");

include("/mylibrary/showproducts.php");

login();

?>

1. Save the file and exit the editor

You should recognize most of this code. It uses the HTML table technique you tested in the imgetest.php program to display the product image (using the <img> tag), the description, price, quantity in stock, and if the product is on sale. There are two things happening here:

* The product description is a link that passes the product ID value to the new page specified in the calling program.
* The onsale value is stored in the database as a Boolean value (true or false), so you need to use an if-then statement to display more descriptive text if the value is set.

The onsale code looks like this:

if ($onsale)

echo "On sale!\n";

else

echo "&nbsp;\n";

If the Boolean value is set, it displays the *On sale!* text. If not, you still need to put something in the table cell, or the browser won't paint the cell correctly. You use the HTML space character, &nbsp;, to enter a blank space.

That should do it. Test this out by selecting a category in the navigation area drop-down box, then click the **Browse** button. You should see the products in your database appear on the Web page!

Displaying products in a category

Now all that's left to do is create a form for the store manager to update products. We'll do that in Chapter 4.