Business Inventory Management System

Sana Yousuf

June 12, 2003

1 Title:

Business Inventory Management System

Description:This program would be useful for retail stores by helping stores track and order supplies/items. A store would be sent a list of items in the latest shipment. The store would then be presented with an easy way of both checking for discrepancies and reporting them back to the distribution center. The store could also order items from the distribution center. The store would then be able to track its shipments, to better get an idea of when new items/supplies are going to arrive.

2 Abstract

Problem: There is no current way for a store to track incoming shipments, and the old method of paper and pen checking-in is time consuming and an electronic method would be

useful.

Method: Online tracking and ordering system would greatly decrease time needed to check in orders while also giving the stores an upperhand with customers as they would be able to know what items would be arriving in the next shipment.

Tools: This program will have to combine many languages and scripts in order to achieve the goal. Some things that will be used are: php, MySql, and html.

Results: No final results yet

Conclusion: No final conclusion yet

3 Introduction

Purpose: In order to better help stores track and order shipments, a program using databases, tables, and the internet will speed up transition times. The purpose of this program is help businesses with the tracking of inventory, orders, and shipments.

For many companies, these three operations are handled seperatly, while in practicality, they shold be connected. There is no way that one can keep a healthy system going without interconnecting the parts that rely on the same information (ie the inventory). Therefore the goal of this project is to not only connect these three essencial parts of a successfull business, but to also connect between the individual stores of the business and the company's distribution center.

This project is worth doing because not only does it include many types of programming (ie beneficial for me the programmer) but also has value in the business management area.

This program would also be beneficial for smaller companies who need an easy and effective way to communicate between their stores and distribution center. After the finalization of the project, it could be packaged and tested out by an actual company, and therefore this program would help companies who need to upgrade their management practices.

The results of this project will be helpful to these companies as they obviously would like a program that does the job in the most efficient manner. Also the results can be applied by adding more components to the project and/or tweaking different aspects of it so as that the company can be satisfied and so that all their requirements are met. This might have to be done on a case by case basis as each company has their own set of requirements they would like in a program such as this. Subsequently, this program might branch out more and into different fields in order to incorporate the many aspects that a company may need to adminster and track.

Scope: To do this, there are a few things to keep in mind. First there will be a log in page, with two different log ins: one for stores and one for the distribution center. From there each has different options -

For the store:

1. Check shipment
2. Send order
3. View inventory
4. Check messages

For the DC:

1. View orders
2. Send shipments
3. View/modify inventory
4. Check messages

Each one of these parts is run by table driven php and html pages that rely on a MySql database. In the database itself, there are quite a few tables, each with a seperate function. Most of these tables keep track of an item, which consists of a sku number, description, color, price, and sometimes, quantity. There is also one table that holds the store numbers and passwords.

The expected results of this project is to have an encompassing online program that will entail all of the parts mentioned above. The program will be able to track orders, shipments, and will be able to control inventory management. Hopefully, it will be easy and applicable to smaller companies who need a practical solultion to inventory and tracking problems.

4 Background

My idea for this project came when I used to work for Laura Ashley, a retail store. Our method of inventory control, shipments, and ordering was all done by paper and this was a very time consuming method. Therefore I decided to use an online method to speed up the transfer of information between the Company and the stores. This would help improve the

business because it would give an upperhand to both levels of the company - the Distribution Center would easily be able to see which items are needed (demand) and then provide them immediatly (supply) while the Stores would be able to order items and also check shipments easily, improving customer relations.

One company that actually exists, Lawson Software, markets systems that are very similar to my project. These systems are web-based and allow their customers, smaller businesses, to effecitively manage their data, including inventory, shipments, orders, personnel, payroll, etc. The programs efficiently manage the data in simple, effective tables in HTML format, to help the users as much as possible. They have an advanced inventory management system that helps their customers manage data between levels of their company (distribution center, stores, administration, etc.) Their customers are usually smaller companies that have outdated methods for managements, such as using manual invoice checkin. They also used an "instant alert" system as well as databases. Therefore this company has the same goals as I did for my project and they use many of the same procedures and have similar actions. Therefore this is a good basis for research as it is very interesting and helpful to learn from a company that actually exists and actually has customers and a market for their product.

5 Procedure and Development

My procedure was to develop the individual parts of the program seperatly and then bring them together at the end. Here are the seperate parts and if they are completed:

1. Construct Inventory – DONE

2. Inventory Viewable by Stores – DONE

3. Inventory Modifiable by Distribution Center (DC)– DONE

4. Orders from Stores to DC – DONE

5. Orders viewable by DC – DONE

6. Shipment Sent to Stores from DC – DONE

7. Shipment Checked by Stores – DONE

8. Message System – DONE

Before describing what each part above does, it is important to know the tables in mySql that were used for this project. These are tables that were used and there function is explained:

1. Discrepency – This table holds all the discrepencies reported by the stores. The variables are: store (int3), sku (int7), qsent (int3), qreceived (int3), qdiff (int3), and date (varchar8). One example is (56, 1687931, 10, 12, 2, 5/10/03).

2. Inventory – This table holds the items the company sells. The variables are: item (varchar20), sku (int7), price (double5,2), and color (varchar10). One example is (Beaded Belt, 6498219, 25.00, Multi).

3. Messages – This table holds the message sent after actions. The messages that can be sent are: "Order sent", "Shipment checked", "Discrepency Found", "Shipment

Sent", and "Inventory Modified." The variables are: Too (char3), Fromm (char3), Subject (varchar25), and Date (varchar8). One example is (All, DC, Inventory Modified, 6/03/03).

4. Order112 – This table holds the ordered items from Store 112. The variables are: item (varchar20), sku (int7), price (double5,2), color (varchar10), and quantity (int3).

5. Order364 – This table holds the ordered items from Store 364. The variables are: item (varchar20), sku (int7), price (double5,2), color (varchar10), and quantity (int3).

6. Order56 – This table holds the ordered items from Store 56. The variables are: item (varchar20), sku (int7), price (double5,2), color (varchar10), and quantity (int3).

7. Order650 – This table holds the ordered items from Store 650. The variables are: item (varchar20), sku (int7), price (double5,2), color (varchar10), and quantity (int3).

8. Order872 – This table holds the ordered items from Store 872. The variables are: item (varchar20), sku (int7), price (double5,2), color (varchar10), and quantity (int3).

9. Ship112 – This table holds the shipped items to Store 112. The variables are: item (varchar20), sku (int7), price (double5,2), color (varchar10), and quantity (int3).

10. Ship364 – This table holds the shipped items to Store 364. The variables are: item (varchar20), sku (int7), price (double5,2), color (varchar10), and quantity (int3).

11. Ship56 – This table holds the shipped items to Store 56. The variables are: item (varchar20), sku (int7), price (double5,2), color (varchar10), and quantity (int3).

12. Ship650 – This table holds the shipped items to Store 650. The variables are: item (varchar20), sku (int7), price (double5,2), color (varchar10), and quantity (int3).

13. Ship872 – This table holds the shipped items to Store 872. The variables are: item (varchar20), sku (int7), price (double5,2), color (varchar10), and quantity (int3).

14. Stores – This table holds the store id numbers and their passwords. The password for the DC does not need to be stored. The variables are: Number (int3) and Password (int5).

The following is a short summary of what each part does:

Construct Inventory – This is a database that stores all the items the company has. The variables associated with each item is its description, sku, color, and price. These are saved to the table "inventory." I accomplished this by first creating "items" on paper and then using a mySql table, I inputed all the items.

Inventory Viewable by Stores – This allows the stores to see the database in an organized manner so they can keep themselves updated on what the company has to offer to its customers. The data is displayed in a table format and can be sorted, but not modified. The coding for this page is in php(during which it accesses the mysql table) and html.

Inventory Modifiable by Distribution Center – This allows the DC to modify, delete, or add items to the inventory. This is essential because inventorys are always changing and this will keep the store updated. After an action has been taken with the inventory, a message is sent to all the stores to report the change. This page is coded using php(during which it accesses the mysql table) and html.

Orders from Stores to DC – Whenever a store needs a certain item, it can easily go to this page and order from the inventory list. This page offers options to delete items from an order, and to then finalize it into the database. This is done by having a seperate table for each of the stores in the database and the corresponding table is modified when the store requests an item. After this event a message is sent to the DC reporting the order. This page is also coded using php and html. The main page is Storeorder.php, which connects to storeshipOrder.php (this page adds the items to the table), which continues to finalize.php (outputs the items to be ordered) and finally it can connect to storeshipOrder2.php which deletes an item from the order.

Orders viewable by DC – This allows the DC to view the databases of each stores order. That way the DC can send the store exactly what it needs. The DC chooses from a primary menu that has the store numbers as options and then they can view the order. Also these items that have been ordered are starred (**) on the shipment page for easy supply/demand actions. This page is in html and php and are outputed into a table.

Shipment sent to Stores from DC – This allows the DC to send shipments to the stores. The DC is first asked which store to send a shipment to. Then it displays the inventory with an add button and it also stars the items that that specific store ordered. As items are added, they are added to a table for that stores shipment. At the end, the table is displayed, can be edited, and then finally is finalized. After this action, a message is sent to the corresponding store reporting the shipment. This too is in html and php. The main page is dcship.php which shows the inventory, which connects to dcshipOrder.php that adds the

items to the shipment table. It then continues on to finalize2.php which shows the items to be shipped, which can then go to dcshipOrder2.php that deletes an item from the shipment table.

Shipment Checked by Stores – This will allow the stores to check in the shipment sent by the DC and to report back any descrepencies found in the shipment. This is done by displaying the table that holds that stores's shipment. Then they are given the input field to input how many of the item they actually got. This is then sent to another page, where if they received the same amount sent, it is simply erased from their shipment table. However if there was a discrepency, it is saved into another table for discrepencies that also holds the store number and then it is erased from the shipment table. This page continues until all the items in the shipment have been checked in and then they can return to the menu. Also the Distribution Center can view these discrepencies as one of their menu options. This allows them to view the errors, and the data fields for this include the store number, the sku of the item, the quantity sent, the quantity received, and the difference betweeen the last two (for example -5 or +2). The page is coded in html and php. The main page is storecheck.php that lists the items in the shipment. When each item is checked, it is redirected to storecheckOrder.php and from there, if the quantities match, it redirects to storecheckGood.php, otherwise it goes to storecheckBad.php. The storecheckBad page adds the discrepency to the corresponding table, and then reconnects to storecheckGood which erases the just checked in item from the shipment table. Then this redirects back to the storecheck page for the other items.

Message System – This will allow the stores and DC to get updates for the following actions:

1. Shipment checked in by a store
2. Ordered made by a store
3. Shipment sent by DC
4. Inventory changed by DC

To do this, after each event the "return to menu" button actually links to the message creator page. This page is sent the "to" and "from" data as well as a number that corresponds to the even that just took place (ie "2" stands for "shipment sent" etc.) Then the message creator page uses this data and creates a variable for the date (using function getdate()) and then adds all this to the database of messages. Then the page redirects back to the menu page to which it corresponds. Also created were two message viewing centers, one for the stores and one for the DC. These are table driven pages that display the messages (for the stores, it displays messages to and from their store number and messages sent to "all", for dc it displays all messages) and also gives the option of deleting messages sent to them. There is only one message table and only the correct messages are displayed using if statements.

All of these pages have crucial data that is passed between them. This data is in the variables "ViewMethod", which holds either the store id number or DC, and "pass" which holds the password. This is important because the store action pages need to know which store is being worked with and the password is used to make sure no one is connecting in to

seperate action pages.

6 Discussion and Results

Discussion: I spent the first couple weeks working with java and java applets. But then I decided to just stick to php, html, and mysql which would make it a lot easier to interface together. I recommend that before anyone start their project, that they think about what languages they want to use and if those languages can work together, or at least pick the languages that interface together the best. This would have undoubtly saved me a lot of time.

After I decided on what languages to use, I sat down and picked out what I expected from my project and what I wanted to have done by the end of the year. I knew that my project had many little parts so the goal was to finish the main parts and then add more functions if I had time. I wanted to get the main parts finished and with time permitting, I was going to add the message system and reformat the pages for better usability.

The hardest thing part of the program was figuring out the php, html, and mysql interfacing. Redirecting data to other pages confused me alot and I spent a lot of time trying to figure this out. However once this was accomplished, the rest of the project was very easy because it was a matter of copy/pasting and modifiying.

Since I did finish my project with time to spare, I was able to make the message center. This was important because while it was one of my goals for the project, it was the last one and the one I hadnt planned on getting to. But luckily I was able to figure it out and have

it working in time for the end of my project.

Results: After completing the project, it was very important to test it. Many times I found little errors that caused massive problems in the mysql tables. So it was crucial that I tested each part of my project so that I could be certain that it actually worked as planned and as a project in its entirety.

In order to test the project, I "ran" each part and took notes on the results. So for every action I tested, I recorded the results. This was important because I was able to see how each action worked and what the results were and how different parts of the program connected together. Another important outcome of my testing was that I realized that there were a few more options that I wanted to add to the project, if I had had extra time. Ultimately, I found that all parts of my project did indeed work and that I had accomplished all that I had set out to do at the beginning of the year.

7 Conclusions

As stated earlier, the goals for this project were accomplished. All the parts work correctly and interface together to create a usable system that helps smaller retail businesses efficiently and easily manage their inventory. The project used a web-based program that further increased its usability and also was very user friendly.

The most important goal for the project was to make sure it actually stood up to what it was supposed to do. This was done and I believe this type of system is very important for companies because it is very advantageous in many ways. Now a company can easily

track and manage their inventory, orders, shipments, and messages with simple web-based table applications. The old methods of manual invoices and faxes, which were very unreliable, has been done away with and has been replaced by methods that are faster and more efficient. This obviously is important in the retail world because the access and process of data information can lead to growth and improved customer service. Therefore I believe a program of this type can be used by many companies which can profit off of these important applications.

It should also be noted that such programs actually do exist in the real world. There are companies out there that write and package such programs for their customers who are smaller companies. These companies are in need of updates as many of the oldfashioned ones still rely on older methods of information transfer. Therefore I think its a good real world application, which in itself is very important. Why create a project when there is no need or advantage to it? Once a project can actually be used and benefited from, then it is an actual accomplishment.

8 References

The following page includes all the references used in this project.

http://www.tjhsst.edu/ syousuf/techlab/references.html

9 Appendixes, Code Listing

1. Login form

```
{

<center>

< table bgcolor="green" cellpadding="12" border="1" >

<tr><td colspan="2"><center><h1><i><b><LogIn></b><i></i><i></h1><i></center>

<i></td><i></tr>

<tr><td><h1><i><b>Store Number or DC:</b></i></h1></td><td>

<form action=***blah.php***" name="login">

<SELECT NAME = "ViewMethod">

<OPTION> 56

<OPTION> 112

<OPTION> 364

<OPTION> 872

<OPTION> 650

<OPTION> DC

</SELECT></td></tr>

<tr><td><h1><i><b>Password:</b></i></h1></td>

<td><input name="pass" type="password"></td></tr>

<tr><td><center><input type="submit" value="Login"></center></td>

<td><center><br><input type="Reset"></form></td></tr></table></center>
```

```
}
// the form is then redirected (look for the stars above) to blah.php
// the following code would be enclosed by <? ... ?> for php but if I do, then
//you wouldnt be able to see it :)
$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxx");  // connects to mySql
mysql_select_db("syousuf", $link);
$query= "select * from stores";
$result = mysql_query($query, $link);
$num_rows = mysql_num_rows($result);
if($ViewMethod=="DC")   // checks distribution center password
  {
     if($pass=="66666")
     header("Location: action2.php"); // if correct, send to this page
     else
     header("Location: blah2.php"); // if incorrect, sent here
  }
else
  {
     for($i=0; $i<$num_rows; $i++) // finds the corresponding data in database
     {
  $row=mysql_fetch_array($result);
```

```
$num=$row["Number"];
$realpass=$row["Password"];
if($num==$ViewMethod)
           {
if($realpass==$pass) // checks the passwords
header("Location: action.php");   // if correct, go here
else
header("Location: blah2.php"); // if incorrect, go here
   }
     }
  }
// action.php and action2.php are the pages where the stores/dc work
// here is the code for blah2.php (the page where for incorrect password)
{
Print("Error - incorrect password -  Will Refresh Automatically");
echo< 'meta http-equiv="refresh" content="2; URL=hello.php"'> // refreshes page aut
<a href="hello.php">Click here if browser does not refresh </a>
}
```

2. Inventory Page Modifyable

```
if(!$sort)
 $sort="item";
```

```
if(!$add)

$add="No";

if(!$amount)

$amount=0;

$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "*********");

mysql_select_db("syousuf", $link);

$query= "select * from inventory order by $sort";

$result = mysql_query($query, $link);

$num_rows = mysql_num_rows($result);

<br>

<br>

<table border=2 align=center width=80% cellpadding=4>

 <tr>

  <td align=center colspan=6>

   <b><font size="+2">Sort by: </font><form action="inventory.php" method=post>

<select name=sort>

<option>Item

<option>Sku

<option>Price

<option>Color

</select><br clear=all>
```

```
<input type="submit" value="Refresh">
</form></b>
  </td>
 </tr>
 <tr>
   <td width=10%>
   <u>Item</u>
  </td>
  <td width=10%>
   <u>Sku</u>
  </td>
  <td width=10%>
   <u>Price</u>
  </td>
  <td width=10%>
   <u>Color</u>
  </td>
  </tr>
for($x=0; $x<$num_rows; $x++)
{
$row=mysql_fetch_array($result);
```

```
$item=$row["item"];

$sku=$row["sku"];

$price=$row["price"];

$color=$row["color"];

Print<<<ENDHTML

<tr>

<td>$item</td>

<td>$sku</td>

<td>$ $price</td>

<td>$color</td>

</tr>

ENDHTML;

}

for($zoo=0; $zoo<$amount; $zoo++)

{

Print<<<ENDHTML

<tr>

<td><input name="a" type="value"></td>

<td><input name="b" type="value"></td>

<td><input name="c" type="value"></td>

<td><input name="d" type="value"></td>
```

```
</tr>

ENDHTML;

}

$amount=0;

</table>

<br><br>

<form action="inventoryM.php" method=post>

Add items to inventory?

<select name=add>

<option>Yes

<option>No

</select><br clear=all>

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

</form>

if($add=="Yes")

{

Print<<<ENDHTML

<form action="inventoryM.php" method=post>

How Many?

<select name=amount>

<option>1
```

```
<option>2
<option>3
<option>4
<option>5
</select><br clear=all>
<input type="submit" value="Add">
</form>
ENDHTML;
}
$add="No";
```

3. Store Ordering Page

```
This page lists out the items in the inventory and allows the user to input the
quantity of an item he/she would like to order
?>
if(!$sort)
 $sort="item";
$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxxx");
mysql_select_db("syousuf", $link);
$query= "select * from inventory order by $sort";
$result = mysql_query($query, $link);
$num_rows = mysql_num_rows($result);
```

```
$array= array();

for($blah=0; $blah<$num_rows; $blah++)

$array["$blah"] =$blah;

Print<<<ENDHTML

<title> Store #$ViewMethod - Send Shipment to DC

</title>

<body bgcolor="#006600" text="white">

<table border=2 align=center width=80% cellpadding=4>

 <tr>

  <td align=center colspan=6>

   <b><font size="+2">Order Form for Store $ViewMethod: </font></b>

  </td>

 </tr>

<tr>

   <td width=30%>

   <u>Item</u>

  </td>

  <td width=18%>

   <u>Sku</u>

  </td>

  <td width=18%>
```

```
   <u>Price</u>
  </td>
  <td width=18%>
   <u>Color</u>
</td>
<td width=4%>
   <u>Quantity to Send</u>
  </td>
 <td width=4%>
 <u> Order Item <u>
 <td>
</tr>
ENDHTML;
for($x=0; $x<$num_rows; $x++)
{
$row=mysql_fetch_array($result);
$item=$row["item"];
$sku=$row["sku"];
$price=$row["price"];
$color=$row["color"];
Print<<<ENDHTML
```

```
<form
action="storeshipOrder.php?a=$item&b=$sku&c=$price&d=$color&ViewMethod=$ViewMethod&
<tr>
<td>$item</td>
<td>$sku</td>
<td>$ $price</td>
<td>$color</td>
<td><input type=text name="e"  value="0"></td>
<td>
<input type="submit" value="Order"></form> </td> </tr>
ENDHTML;
}
Print<<<ENDHTML
 <tr>
  <td align=center colspan=6>
   <b>  <form action="finalize.php?ViewMethod=$ViewMethod&pass=$pass" method=post><
</form>
</table>
ENDHTML;
?>
```

***StoreshipOrder.php

The order for an item is then sent here and added to a database. Then the user is sent back to the order page so they can order other items.

```
<?
$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxx");
mysql_select_db("syousuf", $link);
$query= "select * from order56";
$result = mysql_query($query, $link);
if($ViewMethod==56)
{
mysql_query("INSERT INTO order56 SET item='$a', sku=$b, price=$c,
color='$d', quantity='$e'");
}
if($ViewMethod==112)
{
mysql_query("INSERT INTO order112 SET item='$a', sku=$b, price=$c,
color='$d', quantity='$e'");
}
if($ViewMethod==364)
{
mysql_query("INSERT INTO order364 SET item='$a', sku=$b, price=$c,
```

```
color='$d', quantity='$e'");
}
if($ViewMethod==872)
{
mysql_query("INSERT INTO order872 SET item='$a', sku=$b, price=$c,
color='$d', quantity='$e'");
}
else
{
mysql_query("INSERT INTO order650 SET item='$a', sku=$b, price=$c,
color='$d', quantity='$e'");
}
Header("Location: storeship.php?ViewMethod=$ViewMethod&pass=$pass");
?>
```

**StoreshipOrder2.php

On this page the user can delete an item that they had chosen to order but no longer would like.

```
<?
$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxx");
mysql_select_db("syousuf", $link);
$query= "select * from order56";
```

```
$result = mysql_query($query, $link);

if($ViewMethod==56)

mysql_query("delete from order56 where sku='$a'");

if($ViewMethod==112)

mysql_query("delete from order112 where sku='$a'");

if($ViewMethod==364)

mysql_query("delete from order364 where sku='$a'");

if($ViewMethod==872)

mysql_query("delete from order872 where sku='$a'");

else

mysql_query("delete from order650 where sku='$a'");

Header("Location: finalize.php?ViewMethod=$ViewMethod&pass=$pass");

?>
```

**finalize.php

This is the finalizing page. They user can see what items they have ordered, delete any items they dont want, and can finalize the order and return to the menu page.

```
<table border=2 align=center width=80% cellpadding=4>

 <tr>

  <td align=center colspan=6>

   <b><font size="+2">Finalize Order: </font></b>
```

```
  </td>
 </tr>
<tr>
   <td width=30%>
   <u>Item</u>
  </td>
  <td width=18%>
   <u>Sku</u>
  </td>
  <td width=18%>
   <u>Price</u>
  </td>
  <td width=18%>
   <u>Color</u>
</td>
<td width=4%>
   <u>Quantity to Send</u>
  </td>
 <td width=4%>
 <u> Delete <u>
 <td>
```

```
</tr>

<?

$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxx");

mysql_select_db("syousuf", $link);

if($ViewMethod==56)

$query= "select * from order56";

if($ViewMethod==112)

$query= "select * from order112";

if($ViewMethod==364)

$query= "select * from order364";

if($ViewMethod==872)

$query= "select * from order872";

else

$query= "select * from order650";

$result = mysql_query($query, $link);

$num_rows = mysql_num_rows($result);

for($x=0; $x<$num_rows; $x++)

{

$row=mysql_fetch_array($result);

$item=$row["item"];

$sku=$row["sku"];
```

```
$price=$row["price"];

$color=$row["color"];

$quantity=$row["quantity"];

Print<<<ENDHTML

<form action="storeshipOrder2.php?a=$sku&ViewMethod=$ViewMethod&pass=$pass" method=

<tr>

<td>$item</td>

<td>$sku</td>

<td>$ $price</td>

<td>$color</td>

<td>$quantity</td>

<td><input type="submit" value="Delete"></form> </td> </tr>

ENDHTML;

 }

Print<<<ENDHTML

<tr>

  <td align=center colspan=6>

   <b>  <form action="action.php?ViewMethod=$ViewMethod&pass=$pass" method=post><fo

</form>


</table>
```

```
ENDHTML;

?>
```

4. DC shipment page

```
*******dcship.php

This page lets the DC choose which store to send a shipment to

Then it displays the inventory with a Quantity add button

It also stars the items that store requested (aka ordered)

To add to the shipment table, it goes to dcshipOrder.php

After all items have been ordered, it goes to the finalize page


<?

Print<<<ENDHTML


<title> Send Shipment to a Store

</title>

<body bgcolor="#006600" text="white">

<font size="+1">

Send Shipment to a Store

</font>

<br>

** items that were ordered by store
```

```
<br><br><br>

ENDHTML;


if($choice=="")

{

Print<<<ENDHTML

<table bgcolor="green" cellpadding="12" border="1">

<tr><td><h1><i><b>Choose store:</b></i></h1></td><td><form

action="dcship.php?ViewMethod=$ViewMethod&pass=$pass" method="post" name="Store"><S

"choice">

<OPTION> 56

<OPTION> 112

<OPTION> 364

<OPTION> 872

<OPTION> 650

</SELECT></td>

<td><input type="submit" value="Go">

</td></table>

<br><br><br>

ENDHTML;

}
```

```
else

{

Print<<<ENDHTML

<table border=2 align=center width=80% cellpadding=4>

 <tr>

  <td align=center colspan=6>

   <b><font size="+2">ORDER from #$choice: </font></b>

  </td>

 </tr>

<tr>

   <td width=30%>

   <u>Item</u>

  </td>

  <td width=18%>

   <u>Sku</u>

 </td>

  <td width=18%>

   <u>Price</u>

  </td>

  <td width=18%>
```

```
   <u>Color</u>
</td>
<td width=4%>
   <u>Quantity to Send</u>
  </td>
 <td width=4%>
 <u> Order Item <u>
 <td>
</tr>
ENDHTML;



$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxx");
mysql_select_db("syousuf", $link);

if($choice=="56")
$query2= "select * from order56";

if($choice=="112")
$query2= "select * from order112";
```

```
if($choice=="364")

$query2= "select * from order364";


if($choice=="872")

$query2= "select * from order872";


if($choice==650)

$query2= "select * from order650";


$sort="item";

$query= "select * from inventory order by $sort";


$result = mysql_query($query, $link);

$num_rows = mysql_num_rows($result);




for($x=0; $x<$num_rows; $x++)

{

$row=mysql_fetch_array($result);

$item=$row["item"];

$sku=$row["sku"];
```

```
$price=$row["price"];

$color=$row["color"];

$quantity=$row["quantity"];


$answer="no";


$result2 = mysql_query($query2, $link);

$num_rows2 = mysql_num_rows($result2);


for($y=0;$y<$num_rows2; $y++)

{

$row2=mysql_fetch_array($result2);

$sku2=$row2["sku"];

if($sku2==$sku)

$answer="yes";

}




Print<<<ENDHTML

<form

action="dcshipOrder.php?a=$item&b=$sku&c=$price&d=$color&ViewMethod=$ViewMethod&pas
```

```
method=post>

<tr>

ENDHTML;

if($answer=="yes")

echo('<td>**');

else

echo('<td>');

Print<<<ENDHTML

$item</td>

<td>$sku</td>

<td>$ $price</td>

<td>$color</td>

<td><input type=text name="e"  value="0"></td>

<td>

<input type="submit" value="Order"></form> </td> </tr>

ENDHTML;


}

Print<<<ENDHTML

 <tr>

  <td align=center colspan=6>
```

```
   <b>  <form action="finalize2.php?ViewMethod=$ViewMethod&pass=$pass&choice=$choic
size="+2">Submit:
 <input type="submit" value="Finished">
</form>

</table>
ENDHTML;

}
?>

********dcshipOrder.php
Here the page gets the info for the item to be ordered
it then adds it into the correct shipment table
The it redirects back to the first page

<?
if($e==0)
Header("Location: dcship.php?ViewMethod=$ViewMethod&pass=$pass&choice=$choice");

else
```

```
{
$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxxx");
mysql_select_db("syousuf", $link);

$query= "select * from ship56";
$result = mysql_query($query, $link);

if($choice==56)
{
mysql_query("INSERT INTO ship56 SET item='$a', sku=$b, price=$c,
color='$d', quantity='$e'");
}

if($choice==112)
{
mysql_query("INSERT INTO ship112 SET item='$a', sku=$b, price=$c,
color='$d', quantity='$e'");
}

if($choice==364)
{
```

```
mysql_query("INSERT INTO ship364 SET item='$a', sku=$b, price=$c,

color='$d', quantity='$e'");

}


if($choice==872)

{

mysql_query("INSERT INTO ship872 SET item='$a', sku=$b, price=$c,

color='$d', quantity='$e'");

}


else

{

mysql_query("INSERT INTO ship650 SET item='$a', sku=$b, price=$c,

color='$d', quantity='$e'");

}


Header("Location: dcship.php?ViewMethod=$ViewMethod&pass=$pass&choice=$choice");

}


?>
```

*******dcshipOrder2.php

```
Here the DC can delete items from the shipment it has just chosen

This page redirects back to the finalize page

<?

$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxx");

mysql_select_db("syousuf", $link);


$query= "select * from ship56";

$result = mysql_query($query, $link);


if($choice==56)

mysql_query("delete from ship56 where sku='$a'");


if($choice==112)

mysql_query("delete from ship112 where sku='$a'");


if($choice==364)

mysql_query("delete from ship364 where sku='$a'");


if($choice==872)

mysql_query("delete from ship872 where sku='$a'");

else
```

```
mysql_query("delete from ship650 where sku='$a'");


Header("Location: finalize2.php?ViewMethod=$ViewMethod&pass=$pass&choice=$choice");
?>


*******finalize2.php
This page outputs the shipment table
it also gives the option of deleting items
Then it returns to the DC menu page
<?
Print<<<ENDHTML


<table border=2 align=center width=80% cellpadding=4>
 <tr>
  <td align=center colspan=6>
   <b><font size="+2">Finalize Shipment to #$choice: </font></b>
  </td>
 </tr>
<tr>
   <td width=30%>
   <u>Item</u>
```

```
  </td>
  <td width=18%>
   <u>Sku</u>
  </td>
  <td width=18%>
   <u>Price</u>
  </td>
  <td width=18%>
   <u>Color</u>
</td>
<td width=4%>
   <u>Quantity to Send</u>
  </td>
 <td width=4%>
 <u> Delete <u>
 <td>
</tr>
ENDHTML;
$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxx");
mysql_select_db("syousuf", $link);
 if($choice==56)
```

```
$query= "select * from ship56";

 if($choice==112)

$query= "select * from ship112";

if($choice==364)

$query= "select * from ship364";

if($choice==872)

$query= "select * from ship872";

if($choice==650)

$query= "select * from ship650";

$result = mysql_query($query, $link);

$num_rows = mysql_num_rows($result);

for($x=0; $x<$num_rows; $x++)

{

$row=mysql_fetch_array($result);

$item=$row["item"];

$sku=$row["sku"];

$price=$row["price"];

$color=$row["color"];

$quantity=$row["quantity"];

Print<<<ENDHTML

<form action="dcshipOrder2.php?a=$sku&ViewMethod=$ViewMethod&pass=$pass&choice=$cho
```

```
<tr>

<td>$item</td>

<td>$sku</td>

<td>$ $price</td>

<td>$color</td>

<td>$quantity</td>

<td><input type="submit" value="Delete"></form> </td> </tr>

ENDHTML;

 }

Print<<<ENDHTML

<tr>

  <td align=center colspan=6>

   <b>  <form action="dcship.php?ViewMethod=$ViewMethod&pass=$pass&choice=$choice"

size="+2">Return to Order Form:

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

</form>

</tr>

<tr>

  <td align=center colspan=6>

   <b>  <form action="action2.php?ViewMethod=$ViewMethod&pass=$pass" method=post><f

<input type="submit" value="Finished">
```

```
</form>

</table>

ENDHTML;

?>
```

5. Store Check-in page

```
*********storecheck.php

This page lists the item sent to this store

Then they can input the quantity they actually received

Then this info is sent to storecheckOrder.php

Finally, when all items have been checked in, the store may go back to the menu pag

<?

Print<<<ENDHTML

<title> Check In Shipment for Store #$ViewMethod

</title>

<body bgcolor="#006600" text="white">

<table border=2 align=center width=80% cellpadding=4>

<tr>

  <td align=center colspan=7>

   <b><font size="+2">Shipment from DC: </font></b>

  </td>

 </tr>
```

```
 <tr>
   <td width=10%>
   <u>Item</u>
  </td>
  <td width=10%>
   <u>Sku</u>
  </td>
  <td width=10%>
   <u>Price</u>
  </td>
  <td width=10%>
   <u>Color</u>
  </td>
 <td width=10%>
   <u>Quantity Sent</u>
  </td>
 <td width=5%>
   <u>Quantity Received</u>
</td>
<td width=10%>
   <u>Check in</u>
```

```
</td>

  </tr>

ENDHTML;

$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxxx");

mysql_select_db("syousuf", $link);

if($ViewMethod==56)

$query= "select * from ship56";

if($ViewMethod==112)

$query= "select * from ship112";

if($ViewMethod==650)

$query= "select * from ship650";

if($ViewMethod==872)

$query= "select * from ship872";

if($ViewMethod==364)

$query= "select * from ship364";

$result = mysql_query($query, $link);

$num_rows = mysql_num_rows($result);

for($x=0; $x<$num_rows; $x++)

{

$row=mysql_fetch_array($result);

$item=$row["item"];
```

```
$sku=$row["sku"];

$price=$row["price"];

$color=$row["color"];

$quantity=$row["quantity"];

Print<<<ENDHTML

<form

action="storecheckOrder.php?a=$sku&b=$quantity&ViewMethod=$ViewMethod&pass=$pass"

method=post>

<tr>

<td>$item</td>

<td>$sku</td>

<td>$ $price</td>

<td>$color</td>

<td>$quantity</td>

<td><input type=text name="c"  value="$quantity"></td>

<td>

<input type="submit" value="Check"></form> </td> </tr>

ENDHTML;

}


Print<<<ENDHTML
```

```
<tr>
  <td align=center colspan=7>
   <b><font size="+2">Return to Menu: </font><form action="action.php?ViewMethod=$V
<input type="submit" value="Return">
</form></b>
  </td>
 </tr>
ENDHTML;
//?>
</table>
*********storecheckOrder.php
This page simply redirects the information
If the quantity sent and received matches, it goes to storecheckGood.php
Otherwise, it goes to storecheckBad.php
<?
if($b==$c)
header("Location: storeshipGood.php?ViewMethod=$ViewMethod&pass=$pass&sku=$a");
else
header("Location: storeshipBad.php?ViewMethod=$ViewMethod&pass=$pass&sku=$a&b=$b&c=
?>
```

**********storecheckGood.php

Here the item is simply deleted from the shipment page for that store

then it is redirected back to the store check in page

```
<?
$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxxxx");
mysql_select_db("syousuf", $link);
if($ViewMethod=="56")
mysql_query("delete from ship56 where sku='$sku'");
if($ViewMethod=="112")
mysql_query("delete from ship112 where sku='$sku'");
if($ViewMethod=="364")
mysql_query("delete from ship364 where sku='$sku'");
 if($ViewMethod=="872")
mysql_query("delete from ship872 where sku='$sku'");
if($ViewMethod==650)
mysql_query("delete from ship650 where sku='$sku'");
header ("Location: storecheck.php?ViewMethod=$ViewMethod&pass=$pass");
?>
```

**********storecheckBad.php

Here the discrepency is placed into a table called discrepency

Then it is redirected to storecheckGood.php so it can be deleted from the shipment

```
<?
if(!$ViewMethod)
header ("Location: login.php");
else
{
$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxxxxx");
mysql_select_db("syousuf", $link);
$diff=$c-$b;
mysql_query("INSERT INTO discrepency SET sku='$sku', qsent='$b', qreceived='$c', qd
store='$ViewMethod'");
header ("Location: storecheckGood.php?ViewMethod=$ViewMethod&pass=$pass&sku=$sku");
}
?>
```

6. DC Discrepency Page

```
*******checkdis.php
This pages allows the DC to view all discrepencies
The data fields are: Store number, sku of item, quantity sent, quantity
receieved, and quantity difference
<html>
 <title> Check In Discrepencies </title>
<body bgcolor="#006600" text="white">
```

```
<table border=2 align=center width=80% cellpadding=5>
<tr>
  <td align=center colspan=7>
   <b><font size="+2">Discrepencies: </font></b>
  </td>
 </tr>
 <tr>
   <td width=10%>
   <u>Store</u>
  </td>
  <td width=10%>
   <u>Sku</u>
  </td>
  <td width=10%>
   <u>Quantity Sent</u>
  </td>
  <td width=10%>
   <u>Quantity Received</u>
  </td>
 <td width=10%>
   <u>Quantity Difference</u>
```

```
  </td>
  </tr>
 </html>
<?
   $link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxxxx");
mysql_select_db("syousuf", $link);
 $query= "select * from discrepency";
$result = mysql_query($query, $link);
$num_rows = mysql_num_rows($result);
for($x=0; $x<$num_rows; $x++)
{
$row=mysql_fetch_array($result);
$store=$row["store"];
$sku=$row["sku"];
$qsent=$row["qsent"];
$qreceived=$row["qreceived"];
$qdiff=$row["qdiff"];
Print<<<ENDHTML
<tr>
<td>$store</td>
<td>$sku</td>
```

```
<td>$qsent</td>

<td>$qreceived</td>

<td>

ENDHTML;

if($qdiff>0)

Print('+');

Print<<<ENDHTML

$qdiff</td>

ENDHTML;

}

Print<<<ENDHTML

<tr>

  <td align=center colspan=7>

   <b><font size="+2">Return to Menu: </font><form action="action2.php?ViewMethod=$

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

</form></b>

 </td>

 </tr>

ENDHTML;

?>

</table>
```

```
*****message.php
This is the coding for the message center for the stores.  It lets the store
view the messages that only pertain to them and also lets the delete the
messages to them. (the deleting is done by connection to page messageErase.php
<html>
<body bgcolor="#006600" text="white">
<title> Message Center </title>
</html>
<?
$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxx");
mysql_select_db("syousuf", $link);
$query= "select * from messages order by Date";
$result = mysql_query($query, $link);
$num_rows = mysql_num_rows($result);
?>
<table border=2 align=center width=80% cellpadding=4>
 <tr><td align=center colspan=6>
 <b><font size="+2">Messages: </font> </b> </td>
 </tr>
 <tr>
```

```
   <td width=10%>
   <u>To</u>
  </td>
  <td width=10%>
   <u>From</u>
  </td>
  <td width=10%>
   <u>Date</u>
  </td>
  <td width=10%>
   <u>Message</u>
  </td>
  <td width=10%>
   <u>Delete</u>
  </td>
  </tr>
  <?
for($x=0; $x<$num_rows; $x++)
{
$row=mysql_fetch_array($result);
$too=$row["Too"];
```

```
$fromm=$row["Fromm"];

if($too==$ViewMethod || $fromm==$ViewMethod)

{

$date=$row["Date"];

$message=$row["Subject"];


if($too==$ViewMethod)

{

Print<<<ENDHTML

<form action="messageErase.php?a=$too&b=$fromm&c=$message&ViewMethod=$ViewMethod&pa

<tr>

<td>$too</td>

<td>$fromm</td>

<td>$date</td>

<td>$message</td>

<td><input type="submit" value="Delete"></form></td>

</tr>

ENDHTML;

}

else

{
```

```
Print<<<ENDHTML

<tr>

<td>$too</td>

<td>$fromm</td>

<td>$date</td>

<td>$message</td>

<td>---</td>

</tr>

ENDHTML;

}

}

}

Print<<<ENDHTML

<tr>

  <td align=center colspan=6>

   <b><font size="+2">Return to Menu: </font><form action="action.php?ViewMethod=$V

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

</form></b>

  </td>

 </tr>

ENDHTML;
```

```
?>

</table>

****messageErase.php

This is the page that erase the message sent from the store message center

<?

$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxx");

mysql_select_db("syousuf", $link);

$query= "select * from messages";

$result = mysql_query($query, $link);

mysql_query("delete from messages where Too='$a' and Fromm='$b' and

Subject='$c'");


Header("Location: message.php?ViewMethod=$ViewMethod&pass=$pass");

?>

***message2.php

This page is identical to message.php except it is for the DC so it shows all

messages.  It also gives the DC the option to delete messages sent to them.

(messageErase2.php)

<html>

<body bgcolor="#006600" text="white">

<title> Message Center </title>
```

```
</html>

<?

$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxx");

mysql_select_db("syousuf", $link);


$query= "select * from messages order by Date";

$result = mysql_query($query, $link);

$num_rows = mysql_num_rows($result);

?>

<table border=2 align=center width=80% cellpadding=4>

 <tr><td align=center colspan=6>

 <b><font size="+2">Messages: </font> </b> </td>

 </tr>

 <tr>

   <td width=10%>

   <u>To</u>

  </td>

  <td width=10%>

   <u>From</u>

  </td>

  <td width=10%>
```

```
   <u>Date</u>
  </td>
  <td width=10%>
   <u>Message</u>
  </td>
  <td width=10%>
   <u>Delete</u>
  </td>
  </tr>
 <?
for($x=0; $x<$num_rows; $x++)
{
$row=mysql_fetch_array($result);
$too=$row["Too"];
$fromm=$row["Fromm"];
if($too==$ViewMethod || $fromm==$ViewMethod)
{
$date=$row["Date"];
$message=$row["Subject"];
if($too==$ViewMethod)
{
```

```
Print<<<ENDHTML

<form action="messageErase2.php?a=$too&b=$fromm&c=$message&ViewMethod=$ViewMethod&p

<tr>

<td>$too</td>

<td>$fromm</td>

<td>$date</td>

<td>$message</td>

<td><input type="submit" value="Delete"></form></td>

</tr>

ENDHTML;

}

else

{

Print<<<ENDHTML

<tr>

<td>$too</td>

<td>$fromm</td>

<td>$date</td>

<td>$message</td>

<td>---</td>

</tr>
```

```
ENDHTML;
}
}
}
Print<<<ENDHTML
<tr>
  <td align=center colspan=6>
   <b><font size="+2">Return to Menu: </font><form action="action2.php?ViewMethod=$
<input type="submit" value="Return">
</form></b>
  </td>
 </tr>
ENDHTML;
?>
</table>


******messageErase2.php
This is the delete page for the DC message center
<?
$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxx");
mysql_select_db("syousuf", $link);
```

```
$query= "select * from messages";
$result = mysql_query($query, $link);
mysql_query("delete from messages where Too='$a' and Fromm='$b' and
Subject='$c'");
Header("Location: message2.php?ViewMethod=$ViewMethod&pass=$pass");
?>
```

8. Automatic Message System

```
***** This page creates the messages by taking in data after every event
<?
if($ViewMethod=="")
Header("Location: blah2.php");
else
{
$link=mysql_connect("mysql.tjhsst.edu", "syousuf", "xxxxxx");
mysql_select_db("syousuf", $link);
$query= "select * from messages";
$result = mysql_query($query, $link);
//The following bit gets the date information
$today= getdate();
$month= $today['mon'];
$day= $today['mday'];
```

```
//the following is needed to have May as 07 instead of just 7
if($month<10)
$month= "0$month";
if($day<10)
$day = "0$day";
// This line combines the fields together to make the final date
$tdate= "$month/$day/03";
//this "todo" is sent from the page where the event happened
if($todo==1)
mysql_query("insert into messages set Too='$to', Fromm='$ViewMethod',
Subject='Order Sent', Date='$tdate'");
if($todo==2)
mysql_query("insert into messages set Too='$to', Fromm='$ViewMethod',
Subject='Shipment Checked', Date='$tdate'");
if($todo==3)
mysql_query("insert into messages set Too='$to', Fromm='$ViewMethod',
Subject='Discrepency Found', Date='$tdate'");
if($todo==4)
mysql_query("insert into messages set Too='$to', Fromm='$ViewMethod',
Subject='Shipment Sent', Date='$tdate'");
if($todo==5)
```

```
mysql_query("insert into messages set Too='$to', Fromm='$ViewMethod',
Subject='Inventory Modified', Date='$tdate'");
}
//the information is then redirected back to the menu pages.
if($ViewMethod=="DC")
Header("Location: action2.php?ViewMethod=$ViewMethod&pass=66666");
else
Header("Location: action.php?ViewMethod=$ViewMethod&pass=$pass");
?>
```