

Dynamic Web-Based Business Processing Systems Using Active Server Pages

John J. (Jack) Maher (E-mail: jmaher@vt.edu), Virginia Tech

Abstract

Virtually every business organization is considering how to re-engineer appropriate business processes to utilize the Web to increase efficiency. The importance of event-driven application development in a relational database environment has become the primary model applied in business and web commerce related activities. By providing students with a fundamental understanding of how these information systems can be re-engineered to work efficiently in a web-based environment, we can help prepare the next generation of business professionals with a solid working knowledge to enable them to achieve rapid productivity in real world business environments.

This paper presents a Case study that describes a practical and manageable methodology which can be utilized to educate students concerning the concepts and skills necessary to design, develop, and implement dynamic web-based business processing systems. This case will provide the reader with the conceptual, practical, and technical knowledge necessary to understand the fundamentals of these web-based processing systems, and to explicitly describe how to develop these effective systems with a minimal amount of hardware and software resources.

1. Introduction

This paper presents a Case study that describes a practical and manageable methodology which can be utilized to educate students concerning the concepts and skills necessary to design, develop, and implement dynamic web-based business processing systems. The primary audience is intended to be university faculty teaching accounting and information systems courses. However, the concepts and methods are covered in such a manner to make them highly versatile and generalizable to many diverse settings. The methodology can effectively be utilized by anyone interested in developing and maintaining dynamic web browser-based information and/or electronic commerce systems. Specifically, the goal is to provide a straight-forward and practical introduction to a critically important advancement in accounting and business processing systems. Accounting and AIS students will enter the professional world to utilize, evaluate, provide assurance, and in some cases, help develop advanced business processing systems. A significant percentage of systems currently being developed are internet and web-based. This paper will provide the reader with the conceptual, practical, and technical knowledge necessary to understand the fundamentals of these web-based processing systems, and to explicitly describe how to develop these effective systems with a minimal amount of hardware and software resources. University professors, in turn, can explore specific processing systems of interest in more detail, and educate their students regarding procedures to develop and/or provide assurance regarding the next-generation system that has rapidly become the predominant, first-choice method for business transaction processing.

The particular application area utilized and described here is a web-based order entry business processing system for a sporting goods business, Gold Medal Sporting Goods. This setting is similar to one that students and the general population often interact with on a daily basis. This paper is structured in such a way that the emphasis is first placed upon mastery of the necessary conceptual view to develop and implement web-based business processing systems, followed by the specific technical details necessary to accomplish this using readily available and relatively inexpensive software products.

The Gold Medal case example is designed to illustrate three of the most fundamental and essential procedures utilized in business processing systems: First, the ability to display current records (or a subset of records) from a database table to the remote web user; Second, the ability to add a new transaction record to an existing table (e.g. a Customer's new order in the tblOrders table); and Third, to update a specific field of a specific existing record in a table (e.g. increase the Customer's outstanding balance in the tblCustomer's table).¹ It should be noted here that the first two of these steps can be relatively easily accomplished using popular software products such as Microsoft Frontpage 2002 or the Data Access Pages feature of Access 2002. However, the third and crucial, business processing task of updating existing fields of specific existing records can not be handled without some type of detailed code procedures such as the Active Server Pages (ASP) illustrated here. When a student possesses the conceptual and technical capability to comprehend and implement these three procedures, they will have obtained a critically important and extremely powerful and extensible fundamental foundation in web-based electronic commerce business processing systems.

An implemented version of the basic Order Entry processing system described here can be found on the web at: is.cob.vt.edu. Select "ClassASPWork" at the bottom of the screen. When prompted, you will need to enter a UserID (acis3534) and Password (hokies).

2. Background

Graduates in both accounting and accounting information systems (AIS) fields are expected to possess a certain minimum level of competency regarding technology and computer-related knowledge and skills involving the internet, Web, electronic commerce, networks, and databases, and spreadsheets (AAA, 2000; Davis, 1997). Accounting and AIS departments have increasingly integrated computer processes and applications into the mainstream curriculum in an attempt to ensure their graduates possess the necessary skills for successful use and integration of technology into their future professional career.

Educators and practitioners have long recognized the importance of increasingly sophisticated technical knowledge and skills for graduates of accounting and AIS programs. The American Accounting Association has consistently recommended the use of computer expertise in accounting education for over four decades (AAA, 1959; 1964; 1970; 1986; 2000). The largest professional services firms jointly issued a white paper describing the capabilities needed in accounting graduates (Arthur Andersen & Co. et al., 1989). Included among these is the ability to understand the use of information technology in the accounting practice, as well as the knowledge to locate, access, and organize information from electronic sources. The Accounting Education Change Commission identified capabilities needed by accounting graduates and called for increased coverage of computers and information systems in the accounting curriculum (AECC, 1990).

More recently, a major joint project of the AAA, AICPA, IMA, and the largest professional services firms clearly identified the critical need, as described by both practitioners and academics, of extensive information system coverage for current graduates (AAA, 2000). Furthermore, the project specifically recommended the inclusion of additional "broadening-type" information system courses into the accounting and AIS curriculum at both the undergraduate and graduate level. These, and many other authoritative bodies in various fields, have continually identified the importance of increasingly sophisticated technical knowledge and skills as an important graduation outcome. This overwhelming demand from academics and practitioners alike indicates that the accounting and AIS curriculum should include substantial technological literacy and information systems concepts and knowledge. The Gold Medal Sporting Goods Case described here provides an opportunity for educators to incorporate into their courses one of the most important technological advancements that has occurred in the field of accounting and information systems in several decades, i.e. web-based business processing systems.

3. Pedagogical Approach

The examples described here are purposely kept reasonably basic and straightforward to readily facilitate student understanding. However, the examples include the critical underlying concepts required to understand the basic functioning of these web-based systems, and also provide an important foundation which is necessary for any

extension and expansion of the system to a higher degree of complexity and realism. Incorporating realism into the classroom and case setting is a continual challenge for each professor and must, by its very nature, involve tradeoffs. If we speak only of general abstractions and concepts, students may find it difficult to relate to the material and apply it to a practical setting. Moreover, they may encounter an important problem in adapting quickly and efficiently to the practical environment they will face when they enter the professional world. On the other hand, should we, as professors, decide to implement full-scale real-world complexities and realism into each system project discussed in class, it could easily take the majority of the semester for the students to assimilate all the necessary specialized details that relate to a specific company. This could leave little time for introducing and teaching new concepts and extensions to the existing situation. Using this approach, our students may be trained well for the one particular type of company and situation we elected to focus our detail upon, but lose the important generalizable educational qualities necessary to allow them to be successful in a myriad of business environments.

With these tradeoffs in mind, the Gold Medal Sporting Goods setting utilized here is designed to be realistic enough for students to easily see the direct practicality of the exercise, yet, at the same time, avoid excessive complications that could make the development and implementation process overwhelming and unmanageable for one part of one course. If more realism is desired, more complexities can easily be incorporated into the case study. The goal here is to provide an understanding of important fundamental concepts as well as to include the critical practical and technical details for web-based business processing systems. Many of our students will graduate and become professional consultants and auditors who will be well paid to handle the vast number of individual peculiarities and problematic details that face their clients.

4. Gold Medal Case Setting

The practical and logistical settings of the case study are as follows. Students must design, develop, and implement one portion of the revenue business processing system for a hypothetical retail store that sells sporting goods from its web site, i.e. Gold Medal Sporting Goods. An important requirement for Gold Medal is to be able to receive orders from customers 24/7 over the web. The system must efficiently and effectively gather the necessary data and store each customer's order into the corporate database. The web customer should be provided with relevant, timely, and informative feedback concerning the order being placed. The customer must be able to select the appropriate product they wish to purchase and be provided with immediate feedback that the order has been successfully received.

The following assumptions are provided about Gold Medal's business setting. Some of the assumptions are necessary to provide the general setting, while others are enacted to simplify the business situation and make the development project more manageable for students in a reasonably short period of time as one project in one class.

- 1) Gold Medal requires a real-time, web-based processing system that permits simultaneous updating of multiple tables (files). Specifically, the system should update the tblCustomer table (i.e. the Customer's monetary balance outstanding), the tblProducts table (which includes the quantity-on-hand of all products sold by Gold Medal over its website), and the tblOrders table (which includes the specific details regarding each order received). A more complete view of the relevant tables and actual data fields is provided via the relationships view of the actual database shown in Figure 1.
- 2) When a customer submits an order over the web, the following processing needs to take place. First, a new transaction record must be added to the tblOrders table with the appropriate details. Second, the Customer's balance must be increased in the Customer's table. Finally, the quantity-on-hand of the specific inventory item is immediately reduced in the Products file.²

5. Web-Based System Development

The Gold Medal Sporting Goods system described here uses the Windows 2000 Professional Server (or Windows NT Server) operating system. The actual physical database, which includes the three tables involved, is developed in Microsoft Access. The customer web input form is developed using Microsoft Frontpage. The interaction between the webpage and the database is performed using Active Server Pages (ASP). Finally, Visual

Basic is the scripting language utilized to provide specific instructions to the ASP. There are a variety of operating systems and application software packages available to equivalently implement this system, e.g. the UNIX operating system, Dreamweaver for web page development, Java Script for scripting language. However, it should be recognized that this particular type of Microsoft system using Active Server Pages is highly popular and utilized in many current commercial web applications, e.g. Dell Computers Order Entry.

6. Preparation Of Web Server

The web server must be established with the necessary software and directory permissions to run Active Server Pages and allow controlled access to the desired directories (folders). This can be set up by the College or Departmental server administrator or technical resource person. Alternatively, you may require the students to develop and administer the server system on their own. Students can easily obtain and install Microsoft Personal Webserver (PWS) to establish a functional website on their own personal computer in the dorm/apartment. PWS is included on the installation CD's of many Microsoft standard operating systems such as Windows 98 and Windows 2000 Professional. Establishing their own web server is particularly effective if the students are connected to some type of 24/7 high speed digital communication line, e.g. Ethernet or cable-modem, that are found in many dorms and student apartments these days. The system can also be implemented with PWS using a basic analog phone modem based system.

For our case, Gold Medal Sporting Goods needs a Web Service running that can utilize Active Server Pages, e.g. PWS or Microsoft's Internet Information System (IIS) version 4 or later. The server should also have Microsoft Data Access Components (MDAC) installed which includes the ability to run Active X Data Objects (ADO). Fortunately, installing a new webserver will typically include ADO. The server administrator (e.g. the student) needs to set up and establish the desired folder(s) that will contain the actual database and programs with the necessary Read, Write, and Execute permissions, i.e. ability to execute ASP programs. Although it is not needed for our case study, other applications may require delete permission as well.

Once the server and web services are established, the next step is to develop and store the desired database and tables in the appropriate directory on the server. Gold Medal develops the database and tables using Microsoft Access. The database is called ASPOrderProcessing.mdb. The three tables and their respective fields were described previously in Figure 1.

7. Overview Of Processing That Occurs On Web Server Using ASP

To facilitate the pedagogical delivery to the students, the overall processing has been broken up into four manageable steps. The first step obtains the details of the customer's order via the web using the procedure InputValues.asp. Figure 2 provides the web view of InputValues.asp while Figure 3 displays the code view. The second step uses the procedure (ProcessAddRecord.asp) to take the information from InputValues.asp and add a new record to the Orders transaction table (see Figure 4). The third step is shown in Figure 5 (ProcessAddRecordAndEditRecords.asp) extends the previous program by adding the code necessary to update the Balance field of an existing customer in the tblCustomer table, and also to update the QtyOnHand field for an existing inventory item in the tblProducts table. The fourth and final ASP program (ListTableRecords.asp) (see Figure 6) provides a simple mechanism for students to view records from a particular table over the web using any standard browser. This is especially useful if the students are working on a remote server on which they cannot easily open the Access table (e.g. tblCustomer table) to verify their transactions and ensure that their ASP update programs are working correctly. However, if the students are working on their own local PC using Microsoft Personal Web Server (as discussed earlier) it would not be essential to utilize ListTableRecords.asp.

8. Preparation Of Web-Based Input Form

Gold Medal must develop the first input form to provide the web interface for their customers. This can be achieved using a variety of web development software packages and/or text editors. In our case, Gold Medal uses Frontpage 2002 to develop the input form and provide the necessary html code required to interact with the Web.

Webpage development software has advanced to the stage where users can develop and maintain reasonably effective web sites without having to write any html code directly. The developer simply types what s/he wishes to see and formats it in a manner similar to popular word processors such as Microsoft Word. The software automatically supplies the underlying html code and tags necessary to display the page(s) properly on the Web. Gold Medal's initial web order form should allow the user to enter the details required for the transaction and then pass the data to a second program to complete the necessary processing. The entry form should be saved with an .asp extension (rather than .html) so that the server will recognize the call to activate the ASP interpreter in the asp.dll file. This file interprets the code and allows the requested processing to be performed.

Gold Medal uses the order entry form shown in Figure 2 (InputValues.asp). You can see there is one dropdown combo box (cboCustNum) for the user to select his/her correct customer number and a second combo box (cboProdNum) to select the product number they wish to order. There are also three individual textboxes to collect the transaction date (txtTransdate), quantity desired (txtQuantity), and charge (txtcharge). There are also several label objects on the form to descriptively provide information to the user. All of the objects (controls) described here are similar to those found in any GUI event-driven application form such as those created on the PC level in Access or Visual Basic. The two command buttons at the bottom of the form are utilized to either "Submit" the customer's order for processing (cmdSubmit) or to "Reset" the form back to its original defaults (cmdReset). The html code created by Frontpage in making the Gold Medal customer input form (Input Values.asp) is shown in Figure 3. Some html "purists" have criticized the Frontpage software because it often produces some "extra" code that is not essential to the process as illustrated by the extraneous spacing and formatting html code. However, it is extremely easy to use and produces effective webpages in a very short period of time. Thus, it can be a great productivity enhancer. Figure 3 has line numbers assigned to lines of interest. Importantly, you can see that the "Form Method" code on line 1 is "POST", and the "Action" is set to "ProcessAddRecord.asp". This essentially means that the form will pass the variables collected in the InputValues.asp form (i.e. Customer Number, Product Number ...) to the ProcessAddRecord.asp program to perform the required processing.

9. ASP Program Details

At this point it's useful to discuss the important details regarding each of the three remaining programs utilized by Gold Medal on its web site without covering each and every line of code. To begin with, it should be recognized that ASP pages are typically a mixture of html formatting commands and ASP processing commands. You can recognize the ASP code because it will be enclosed in `<% ...%>` beginning and ending tags. You can develop the ASP page using Frontpage to develop the basic "look" of the page, and then add the desired .asp Visual Basic (VB) script using Frontpage in HTML view or using any text editor (e.g. Wordpad).

9.1. ProcessAddRecord.asp

The second program, shown in Figure 4, (ProcessAddRecord.asp) performs the important step of adding a new order to the Gold Medal database. Basically, it adds a new record to the tblOrders table, and then provides the web user with some basic feedback information that the transaction has been recorded. The first two executed ASP code lines establish a "Connection" object with the current database, informs ASP that this is an Access database (i.e. mdb), and defines the actual location of the database using its relative storage location. The third line of code opens the actual database in the Connection. Lines 4 uses an SQL statement to define the tblOrders table to open with all (*) its fields available. Line 5 defines the recordset (table) object, and Line 6 opens the desired records as rsTrans and specifies the characteristics of the actual recordset (table).

Lines 7 through 13 uses VB script to add a new record to the tblOrders table using the assigned rsTrans name, and replaces the blank fields in the new table with the data obtained from the input form comboboxes and textboxes. Line 14 provides the user with some basic feedback to let them know that the transaction was recorded. Lines 15 through 18 properly close out the environment, while line 19 provides a web link to return to the original input form.

This is the end of the ProcessAddRecord.asp program that writes to the database table. When students get the system working this far, they have achieved the very important steps of successfully identifying and interacting with the actual company database across the web. This is a major milestone in the overall completion of the project. The next program in the Gold Medal case extends the basic processing of ProcessAddRecord.asp to include the updating of existing balances in existing records.

9.2. ProcessAddRecordAndEditRecords.asp

The ProcessAddRecordAndEditRecords.asp program is an actual extension of ProcessAddRecord.asp so the description will continue with line 20 (see Figure 5). Line 20 uses an SQL statement to produce a record set which contains only the one matching customer record that was selected by the customer from the combobox (cb0Custno) on the input form. The customer balance field will be updated for this particular record. Lines 21 and 22 open the tblCustomer in a manner similar to that used previously to open the tblOrders table. Line 23 and 24 actually perform the update of the customer's Balance Outstanding by adding the cost of the new charges, while lines 25 and 26 provide feedback to the user informing them of their new balance.

A similar type of processing occurs for the Products table in lines 27-33. The tblProducts is opened with the single record that matches the product number selected in cboProdo on the form. The QtyOnHand field is reduced by the quantity ordered by the customer, and some basic feedback is provided listing the new quantity on hand for that inventory item. The Connection and tables are closed out, and the environment reset in lines 34-41. Finally, line 42 provides a link back to the input form.

9.3. ListTableRecords.asp

The final ASP program for the Gold Medal Case, ListTableRecords.asp, shown in Figure 6, provides the ability to view all the records of a particular table across the web. This is particularly useful when students are working in a remote environment away from the actual server on which the database and tables exist, and need to view the results of their work. Lines 1-6 are very similar to those found in the ProcessAddRecord.asp program described earlier. Basically, lines 1-3 establish the Connection, identify the database as an Access database, and open the Connection. Lines 4-6 opens the tblCustomer table with all the records and all the fields available. Lines 7-10 represent a loop that determines and identifies each field in the table. Lines 11-18 identify the column headings that describe the fields to the user on the web page. Lines 19-27 represent a second loop that lists all the records in the table. Finally, lines 27-31 properly close out the table and Connection.

10. Discussion And Extensions

The programs described here are designed to help the student learn the essential concepts and technical details of web business processing in a situation that provides a practical setting without becoming excessively complex and potentially overwhelming. In reality, it is highly unlikely that Gold Medal would allow its customers to be able to list all the records that exist in various Gold Medal tables. In addition, Gold Medal would not likely provide certain information to the customer that is provided here after their transaction is complete. However, the main purpose of the Gold Medal case is to illustrate how a student can create a web-based system that successfully allows a user to add records, display records, and update specific records on a remote database. These are powerful and fundamental concepts that are involved in any web-based processing systems. Additional realism and complexities can be added to the system by using these same concepts. For example, a customer or employee could be required to enter their userid and password upon entering the web-based system in order to authenticate themselves. This authentication can also be used to restrict each user to the particular subset of tables and records that they are permitted to access. These individual permissions can be added by the database server administrator or built into the ASP program.

Additionally, the simplified process of placing orders and updating customer balances and inventory quantities on hand can be broken into multiple individual steps. For a business to business transaction this might include filling the order, shipping the order, invoicing the customer, and receiving payment from the customer. Each

of these steps could be set up to work within a web-based corporate intranet that has ASP programs controlling each step of the way. Incorporation of each of these aspects would require a longer period of time that that necessary for the completion of the Gold Medal Sporting Goods case described here, but could be handled using the exact same basic concepts.

The concepts and processes described here are highly generalizable and can be applied to many different business applications. Some examples include purchasing, payroll, internal job costing, and inventory management. Essentially, the basic techniques can be applied to any business processing activity that utilizes digital input.

11. Sources Of Additional Information

There are a variety of books and online sources that can provide a great deal of information web processing and Active Server Pages. Books that have been very helpful in developing the ASP programs described in this article include: "Step by Step Web Database Development" (Buyens 2000), "Beginning Active Server Pages 2.0" (Francis et al., 1998), and "Database-Driven Web Sites" (Morrison and Morrison, 2000). Information can also be found directly from the Microsoft web site at <http://msdn.microsoft.com/library/default.asp>. This includes ASP code examples, reference guides, and other information on the subject area.

12. Conclusion

The business world is currently experiencing an important substantive revolution that is affecting all aspects of its operations. As illustrated by the words of David Calhoun, Vice-President of General Electric Corporation, "the task that employees at GE must undertake in the next few years is to obtain the knowledge of how to incorporate the Web into each of their daily business activities" (Virginia Tech, 10/19/2000).


Virtually every business organization is considering how to re-engineer appropriate business processes to utilize the Web to increase efficiency. Potential business processes and applications that can be re-engineered in this fashion include Order Entry/Revenue generation processes, Inventory management, and Purchasing/Expenditure processes. The importance of event-driven application development in a relational database environment has become the primary model applied in business and web commerce related activities. By providing students with a fundamental understanding of how these information systems can be re-engineered to work efficiently in a web-based environment, we can help prepare the next generation of business professionals with a solid working knowledge to enable them to achieve rapid productivity in real world business environments. Moreover, it will help us produce students who can enter the workforce with the basic knowledge and practical experience required to develop, utilize, audit, and monitor these state-of-the-art business process systems. 

Figure 1
Relationships View of Database

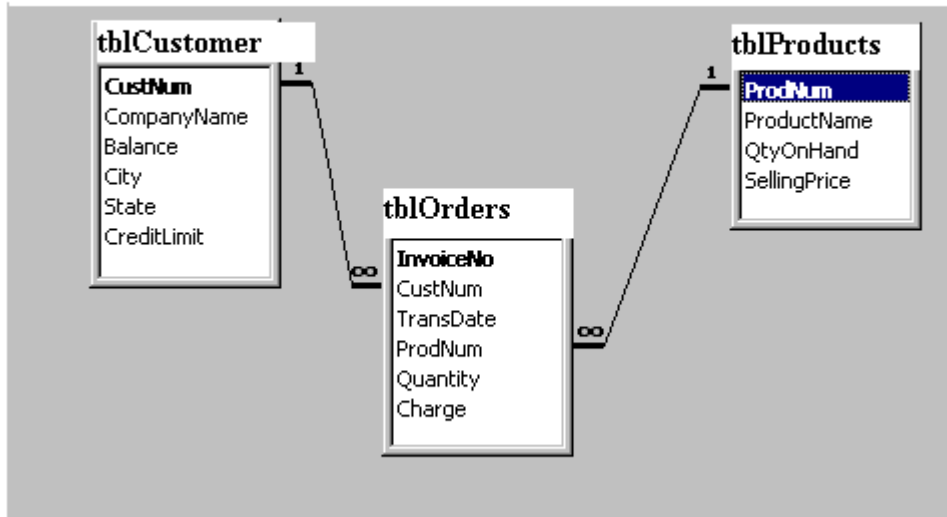


Figure 2 Input Values.asp
Data Entry Form For Transactions

Customer Number:

Transaction Date:

Product Number:

Quantity:

Charge:

Figure 4
ProcessAddRecord.asp

```
<html> <head> <title>transactions</title>
<!-- This ASP file name is called "ProcessAddRecord.asp" -->

<% 'This line indicates the beginning of ASP code as indicated by "<% " 'Establish a connection "myConnection" to database
1 Set myConnection = Server.CreateObject("ADODB.Connection")

'identify actual database as an Access database and map path
"relative" location; store this full path in "myDbase" variable to use in the open connection statement
2 myDbase = "driver={Microsoft Access Driver (*.mdb)};" & _
    "dbq=" & Server.MapPath("ASPOrderProcessing.mdb")
'One alternative to the "relative" location is to open the database using an "absolute" location with the line below
'openStr = "driver={Microsoft Access Driver (*.mdb)};" & _
'    "dbq=E:\StuPages\ClassWeb\FoldASP2\ASPOrderProcessing.mdb"

'Open your database and connection
3 myConnection.Open myDbase, "", ""

'Create a temporary variable called "mySQL" to hold the SQL statment to retrieve all fields and all records from
tblOrders
4 mySQL = "SELECT * FROM [tblOrders];"
'The next two lines opens a Recordset "rsOrders" using the "mySQL" statement in the "myConnection" connection
5 Set rsOrders = Server.CreateObject("ADODB.Recordset")
6 rsOrders.Open mySQL, myConnection, adOpenDynamic, adLockOptimistic, adCmdText

' Add a new record to tblOrders table and write contents of form comboboxes and textboxes into tblOrders fields
7 rsOrders.AddNew
8 rsOrders("CustNum") = Request("cboCustNum")
9 rsOrders("Transdate") = Request("txtTransDate")
10 rsOrders("ProdNum") = Request("cboProdNum")
11 rsOrders("Quantity") = Request("txtQuantity")
12 rsOrders("Charge") = Request("txtCharge")
13 rsOrders.Update

'Provide some brief feedback to the web user
14 Response.Write "Successfully Added New Record For Customer# " & Request("cboCustNum") & "<BR><P>"
'Add the entire code from above to the "ProcessAddRecordAndEditRecords.asp program"
%>
</head> <body>
<% 'Close the tables and connection; release the storage variables utilized (i.e. Nothing)
15 rsOrders.Close
16 Set rsOrders = Nothing
17 myConnection.Close
18 Set myConnection = Nothing %>
<!--Provide link to return to input form
19 <p>Return to Input Form <a href="InputValues.asp">InputValues.asp</a></p>
</body> </html>
```

Figure 5
ProcessAddRecordAndEditRecords.asp

```

<!--*****-->
'Insert the initial VBA script code from ProcessAddNewrecord.asp here
<% 'Use SQL statement to open a Recordset with one customer record and all fields from tblCustomer table
'The Where clause will open only the one specific Customer's record selected on the form
20 mySQL2 = "SELECT * FROM [tblCustomer] where CustNum = " & Request("cboCustNum") &";"

'The next two lines opens a Recordset "rsCustomer" using the "mySQL2" statement in the "myConnection"
connection
21 Set rsCustomer = Server.CreateObject("ADODB.Recordset")
22 rsCustomer.Open mySQL2, myConnection, adOpenDynamic, adLockOptimistic, adCmdText

'Update existing customer balance with new charge 'Note: do NOT need VBA command rsCustomer.Edit
23 rsCustomer("Balance") = rsCustomer("Balance") + rsOrders("Charge")
24 rsCustomer.Update
'Provide some feedback to webuser
25 Response.Write "      Successfully Updated Balance For Customer # " & Request("cboCustNum") &
"<BR><P>"
26 Response.Write "      New Balance is " & rsCustomer("Balance") & "<BR><P>"

'Open a Recordset with one product record and all fields from tblProducts table
'The Where clause will open only the one specific Product's record selected on the form
27 mySQL3 = "SELECT * FROM [tblProducts] where ProdNum = " & Request("cboProdNum") &";"
'open a Recordset "rsProducts" using the "mySQL3" statement in the "myConnection" connection
28 Set rsProducts = Server.CreateObject("ADODB.Recordset")
29 rsProducts.Open mySQL3, myConnection, adOpenDynamic, adLockOptimistic, adCmdText

'Reduce existing Quantity on Hand with quantity removed
30 rsProducts("QtyonHand") = rsProducts("QtyonHand") - rsOrders("Quantity")
31 rsProducts.Update

'Provide feedback to web user
32 Response.Write "      Successfully Updated QtyOnHand For Product # " & Request("cboProdNum") &
"<BR><P>"
33 Response.Write "      New QtyonHand is " & rsProducts("QtyonHand") & "<BR><P>"

%> </head> <body>
<!--*****-->
<% 'Close all tables and connection; release the storage variables utilized (i.e. Nothing)
34 rsOrders.Close
35 Set rsOrders = Nothing
36 rsCustomer.Close
37 Set rsCustomer = Nothing
38 rsProducts.Close
39 Set rsProducts = Nothing
40 myConnection.Close
41 Set myConnection = Nothing %>

<!--Provide link to return to input form
42 <p>Return to Input Form <a href="InputValues2.asp">InputValues2.asp</a></p>
</body> </html>

```

Figure 6
ListTableRecords.asp

```

<html><head> <title>members</title>
<! This program is called "ListTableRecords.asp" <!-- #include file="adovbs.inc" -->
<%
1 Set myConnection = Server.CreateObject("ADODB.Connection")
2 myDbase = "driver={Microsoft Access Driver (*.mdb)};" & _
  "dbq=" & Server.MapPath("ASPOrderProcessing.mdb")
3 myConnection.Open myDbase, "", ""

4 mySQL = "SELECT * FROM [tblCustomer];"
5 Set rsMyTable = Server.CreateObject("ADODB.Recordset")
6 rsMyTable.Open mySQL, myConnection, adOpenForwardOnly, adLockReadOnly, adCmdText
%> <
</head> <body>

<%
7 Set holdrec = Server.CreateObject("Scripting.Dictionary")
8 For Each fld in rsMyTable.fields
9 holdrec(fld.Name) = fld.Value
10 Next

%>
<! Describe column headings (labels) to user on web page-->
11 <table border="1" cellspacing="0" cellpadding="3">
12 <caption><b>Customers</b></caption>
  <tr>
13 <th>Customer Number</th> <th>Company Name</th> <th>Balance</th>
14 <th>City</th> <th>State</th> <th>CreditLimit</th>
  </tr>
  <%
'Loop through all records in tblCustomer table and display on web
15 Do While Not rsMyTable.EOF
  %>
  <tr valign="TOP">
16 <td><%=Server.HTMLEncode(rsMyTable.Fields("CustNum").Value)%> &nbsp;
17 <td><%=Server.HTMLEncode(rsMyTable.Fields("CompanyName").Value)%>&nbsp;</td>
18 <td><%=Server.HTMLEncode(rsMyTable.Fields("Balance").Value)%>&nbsp;</td>
19 <td><%=Server.HTMLEncode(rsMyTable.Fields("City").Value)%>&nbsp;</td>
20 <td><%=Server.HTMLEncode(rsMyTable.Fields("State").Value)%>&nbsp;</td>
21 <td><%=Server.HTMLEncode(rsMyTable.Fields("CreditLimit").Value)%>&nbsp;</td>
  </tr> <% 'move record pointer to next record in tblCustomer
22 rsMyTable.MoveNext
23 loop 'loop back to Do While statement

24 rsMyTable.Close
25 Set rsMyTable = Nothing
26 myConnection.Close
27 Set myConnection = Nothing
%> </table> </body> </html>

```

References

1. Accounting Education Change Commission. 1990. "Objectives of education for accountants: Position statement number 1", *Issues in Accounting Education* (Fall): 307-312.
2. American Accounting Association (AAA). 1959. "Report of committee on accounting instruction in electronic data processing", *Accounting Review* (April): 215-220.
3. ----- 1964. "Report of committee on accounting systems instruction", *Accounting Review* (July): 715-720.
4. ----- 1970. "Report of the 1968-69 committee on the role of the computer in accounting education", *Accounting Review* (supplement): 29-43.
5. ----- 1986. "Future accounting education: Preparing for the expanding profession", (Committee on the Future Structure, Content, and Scope of Accounting Education). *Issues in Accounting Education* (Spring): 168-195.
6. ----- 2000. "Accounting Education: Charting the Course through a Perilous Future", (W. Steve Albrecht and Robert J. Sack) *Accounting Education Series*, Volume No. 16, American Accounting Association (Sarasota, Florida).
7. Arthur Anderson & Co, Arthur Young, Coopers & Lybrand, Ernst & Whinney, Peat Marwick & Co, Price Waterhouse. 1989. *Perspectives on education: Capabilities for success in the accounting profession*. (New York).
8. Buyens, J. 2000. *Step by Step Web Database Development*. Microsoft Press. Redmond, Washington.
9. Calhoun, D. 2000. *Speech delivered to Pamplin College of Business audience*. October, 19, 2000. Virginia Tech, Blacksburg, Virginia.
10. Davis, P. 1997. "What computer skills do employers expect from recent college graduates?", *Technological Horizons in Education Journal*: 74-78.
11. Elliott, R. K. and D. M. Pallais. 1997. "Are you ready for new assurance services?", *Journal of Accountancy* (June): 47-51.
12. Francis, B., J. Kauffman, J.T. Llibre, D. Sussman, and C. Ullman. 1998. *Beginning Active Server Pages 2.0* Wrox Press Ltd. Chicago, Illinois.
13. Gelinas, U.J. and S.G. Sutton. 2002 *Accounting Information Systems* Fifth edition. South-Western, Thomson Learning. Cincinnati, Ohio.
14. Hall, J.A. 2001. *Accounting Information Systems* Third Edition. South-Western College Publishing-Thomson Learning. Cincinnati, Ohio
15. Hollander A.S., E.L. Denna, and J.O. Cherrington. 2000. *Accounting, Information Technology, and Business Solutions* Second Edition. Irwin McGraw-Hill U.S.A.
16. Morrison, M. and J. Morrison. 2000. *Database-Driven Web Sites*. Course Technology - Thomson Learning. Cambridge, MA.
17. Moscové S.A., M.G. Simkin, and N.A. Bagranoff. 1999. *Core concepts of Accounting Information Systems* Sixth Edition. Wiley. U.S.A.
18. Romney, M.B. and P.J. Steinbart. 2000. *Accounting Information Systems* Eighth Edition. Prentice-Hall Upper Saddle River, New Jersey.

Endnotes

1. It should be observed that deleting an existing record in a table is often considered a fourth common processing procedure. However, it can easily be subsumed in the third step, and does not add anything conceptually to the system or processing development. Hence, it is not included in this case study. Furthermore, traditional batch processing procedures that involve the systematic processing of new records in a transaction file and the subsequent updating of one or more "master" tables (files) is an important business processing procedure, but the description and processing involved is considered outside the scope of this case study.
2. To simplify the required processing, it is assumed each order contains only one product and the balance is increased when the order is taken. In reality, the customer's balance is likely to be increased when the order is shipped and invoiced. This two-stage extension could easily be incorporated into the case. This same

concept holds true for the inventory reduction in the tblProducts table. It is also assumed that the customer has a credit account in good standing with the firm. It should be pointed out that there are many excellent resources available that describe more complex and realistic revenue business processing systems in great detail. Some examples are: Accounting Information Systems, Gelinas and Sutton 2002; Accounting Information Systems, Hall 2001; Accounting, Information Technology, and Business Solutions, Hollander et al. 2000; Accounting Information Systems, Moscove et al. 1999; Accounting Information Systems, Romney and Steinbart 2000. These complexities can be readily added to the processing described here by incorporating the same basic concepts described in this case.

Notes

Endnotes

- ¹ It should be observed that deleting an existing record in a table is often considered a fourth common processing procedure. However, it can easily be subsumed in the third step, and does not add anything conceptually to the system or processing development. Hence, it is not included in this case study. Furthermore, traditional batch processing procedures that involve the systematic processing of new records in a transaction file and the subsequent updating of one or more “master” tables (files) is an important business processing procedure, but the description and processing involved is considered outside the scope of this case study.
- ² To simplify the required processing, it is assumed each order contains only one product and the balance is increased when the order is taken. In reality, the customer’s balance is likely to be increased when the order is shipped and invoiced. This two-stage extension could easily be incorporated into the case. This same concept holds true for the inventory reduction in the tblProducts table. It is also assumed that the customer has a credit account in good standing with the firm. It should be pointed out that there are many excellent resources available that describe more complex and realistic revenue business processing systems in great detail. Some examples are: Accounting Information Systems, Gelinas and Sutton 2002; Accounting Information Systems, Hall 2001; Accounting, Information Technology, and Business Solutions, Hollander et al. 2000; Accounting Information Systems, Moscove et al. 1999; Accounting Information Systems, Romney and Steinbart 2000. These complexities can be readily added to the processing described here by incorporating the same basic concepts described in this case.

Notes