Online Business Rating Web Community

By

Robert P. Brown III

Submitted to
the Faculty of the Information Engineering Technology Program
in Partial Fulfillment of the requirements for
the Degree of Bachelor of Science
in Information Engineering Technology

University of Cincinnati
College of Applied Science

March 2001

Acknowledgements

I would like to thank all those who have stood by me and encouraged me when I felt like I could not continue with this project. Doug Saunders for always encouraging and supporting me throughout the entire project. Ashika Wijesekera, Candise Wise, Caryle Clear, Jeff Black, C.E. Reutter, Jen Fritz, Robin Miller, Pam Naber and Priya Shah for their honest feedback and assistance on the final documentation. My advisor, Russ McMahon for his advice, guidance and honest feedback throughout the development of the project. Sam Geonetta, for his assistance on the user interface and layout of the website. Finally, Soleda Leung, for her encouragement and enthusiasm as my instructor, without her I may not have been here today.

Table of Contents

List of Figures

Abstract

The Internet allows people to communicate to massive amounts of people more easily than ever. With the number of complaints to the Better Business Bureau on the rise, people need to become more aware of good and below average businesses.

This project is an online web community that allows people to read and post their opinions/reactions about businesses. Anyone with Internet access can quickly pull up information on a wide variety of businesses and see how others rated that business. In addition, anyone can rate a business for others to see.

The application consists of a three-tier architecture, a data server, web server and client browser. With the addition of a scaleable design, the existing application framework outlined in this report can be extended with minor changes.

Online Business Rating Web Community

1. Statement of the problem

In 1997, the Better Business Bureau assisted people with 2,044,048 complaints, excluding complaints about the auto industry. In 1998, they assisted the public with 2,761,929 complaints; a thirty-five percent increase over 1997's total, as shown in Figure 1 (6, p. 1). Holly Cherico, public relations for the Better Business Bureau states, "The trend shows that the number of complaints has risen over the past few years and shows that it will continue to rise" (15).

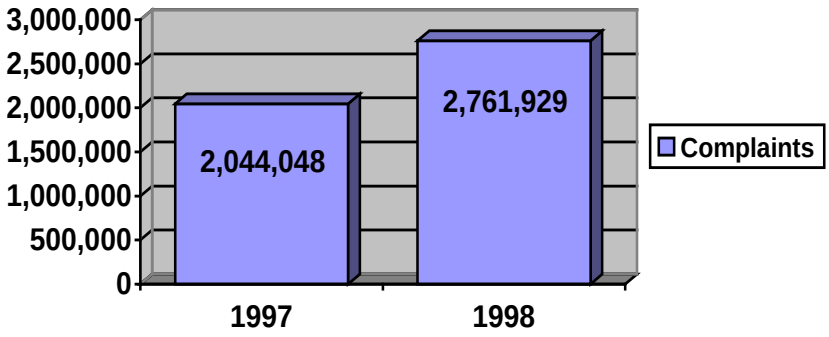


Figure 1. Graph of Complaints To BBB.

In 1998 the Better Business Bureau (BBB) processed 380,000 consumer complaints and out of these 380,000 complaints 61.6% were settled (15). This means that 234,080 cases were settled to the satisfaction of the consumer or the business has done a fair and honest job. What about the other 145,920 other cases that are unsettled? With new businesses and e-commerce taking off the number of complaints are going to rise and so are the number of unresolved problems (15).

1.1 Definition of the Need

Many consumers, myself included, do not find out about bad businesses until it is too late; and finding out about good businesses is usually through word of mouth. As a consumer, there are two possible options for consumers to solve business-related problems. Option one; resolve the issue with the company. This method, however, is not always effective. What if the company is unable or unwilling to assist you, or perhaps the company no longer exists. The second is that you can file a complaint with the Better Business Bureau. This method requires you to provide a large amount of information in order to decide if the complaint can be handled and to process it. After the BBB has opened a case on a business the arbitration period alone can take up to four weeks and even then the consumer may not be satisfied with the result. Finally, consumers relying on information from the BBB alone do not get any specifics on the complaints against that company (7).

2. Review of the Literature

This project is an online Web community that allows users to give feedback about businesses in the Greater Cincinnati/Tri-State Area. In addition, users can view feedback about local businesses. This project involves creating both the design and development of the web site, as well as, the back-end functionality and creation of the database to warehouse the data.

This idea can be a valuable asset to the community. The Internet has leveled the playing field for the average person by providing a medium that anyone can publicly publish to. Even though the Internet has done this people still need to have a centralized

place to go where all their voices can be heard and spoken. People need to have instant access to information about businesses before paying for their services. This is the void that my web community will help to fill.

This web community allows people to search for the specific business they wish to inquire about. They are able to locate a business by selecting the appropriate service area from the home page such as 'Automobiles', 'Food Service', and 'Health Care'. In addition, users are able to instantly search for a business from any page by entering the name of the business in the search box. Once a user has located the business that they are looking for, they are given information such as the name, area of service, address, phone number, a description and image of the business, if available, and the overall rating of this business (if users have rated it). The overall rating is calculated by combining all ratings for that particular business.

Not only does the web community allow users to view a particular business, but they can also rate the business. Because security is an issue, to be able to rate a business, the user will have to register using a valid e-mail address. The user has to fill out a form submitting a user ID, password and a valid e-mail address. The user will then receive an e-mail with a link to activate their account. Once their account has been activated they can then log onto the site and rate any business they choose. In order to be fair each user can rate each business only once; they may, however, go back and change their feedback and or rating at a later time.

With an online community allowing users to rate business, consumers can find out where to go to receive better service, as well as, which businesses to avoid altogether. In addition, such a community will allow businesses to see what they are doing wrong.

By having access to the public perception of the business, a less than favorable company can possibly improve its services. The good businesses, on the other hand, will receive positive reinforcement.

The web community is hosted on Microsoft's Internet Information Server, using Active Server Pages to provide dynamic content and functionality. The back-end data is warehoused by Microsoft SQL Server v7.0. I have also expanded my knowledge in Active Server Pages to create the functionality that is needed for this dynamic site. I have also gained the knowledge and experience in creating COM objects in Visual Basic that are used in pages where processing speed is a must (rating entry) (59). Finally, I have gained experience in graphic design giving the online web community an attractive look and feel to it (61).

3.Description of the solution

3.1 User profile

The targeted users of this web site are the general public. The only requirement that they must fulfill is that they have some basic computer understanding and know how to navigate the World Wide Web. My intentions were to create a site that is as intuitive as possible.

3.2 Design Protocols

Every page in the site has a standard layout. At the top of every page there is a banner with the site's logo and a search feature on the far right. The search feature consists of a textbox and a 'Go' button. Below the top banner is the site navigation bar. These buttons span the length of the page and appear as square buttons. Each button has

a rollover effect applied to it; this lets the user know that these are active navigation buttons. Additionally, inactive buttons appear on the top; but will be a darker shade of color the active buttons use. This shows the user that the button is currently not active. The page content appears below the site navigation bar; the page that the user is on determines the content. Finally, there is a page footer that appears below the page content. The footer is displayed in a small font size and consists of links to major sections of the site for quick navigation. An illustrated diagram of the site can be found in Appendix A. The user has access to the following pages while on the site.

Page 1: *index.asp* – Figure 4 on page 31

This is the entrance to the site. It greets the user and provides them with several options: browsing by category, adding a listing, searching for a particular business, registration and help. The page has a banner at the top with navigation buttons below it. The page also has a list of different top-level business categories, which when clicked on will take them to the subcategories in that particular area. The index.asp page will also display the last ten updated businesses.

Page 2: *allcats.asp* – Figure 5 on page 34

This is a page that may or may not be displayed on the site. If the top level of categories becomes too long to be displayed on the index.asp page then a link is displayed allowing the user to access this page. It then displays, in alphabetical order, all of the categories with their associated subcategories. This allows the user on the site to more easily navigate into different categories and subcategories.

Page 3: *subcatlist.asp* – Figure 6 on page 36

This page is displayed when a user clicks on a top-level business category. This lists all the subcategories for that top-level category. Each of the subcategories is a link and when clicked it will take the user to an Active Server Page, *blist.asp*, which lists all the businesses in the selected subcategory.

Page 4: *blist.asp* – Figure 7 on page 38

This page is displayed when the user selects a subcategory. It lists every business that falls under the selected subcategory. Each business listing is a hyperlink that will take the user to the information page about that business.

Page 5: *business.asp* – Figure 8 on page 40

Business.asp is executed when the user selects a business from the *blist.asp* page or from the *search.asp* page; it presents information on a selected business. For each business the information displayed includes: the name, address, city, state, zip code, phone number, web address, e-mail address and a photo of the business. Additional information, such as number of ratings the business has received and user ratings are shown. The rating information displayed is determined by the subcategory that the business falls into.

This page presents the user with two options. The first option is to rate the business and will take them to the information entry page for the business, *ratebiz.asp*. The second will allow the user to view other community members' entries for a particular business.

Page 6: *search.asp* – Figure 9 on page 42

This page is linked to every page on the site. It is executed when a user enters a business name in the search textbox located on the page header. It displays a list of

businesses that match the business name. The results are generated using a Soundex query in SQL Server.

Page 7: *asearch.asp* – Figure 10 on page 43

This is an advanced search page; it allows the user to search for listings by entering more than one criteria. The criteria that can be searched by is the business name, zip code and subcategory. When the user clicks "submit" it will execute a search and return the results broken down by category and subcategory. Each listing found is a hyperlink to the business information page, *business.asp*.

Page 8: *reg.asp* – Figure 11 on page 44

This allows the user to become a registered user of the system. When the page is initially displayed it gives the user a brief list of instructions on what is necessary to become a member of the system and instructions on what to fill out on the form. There is some JavaScript on the form that checks to make sure the necessary fields are filled in. The user will then submit the information that will take them to the *vreg.asp* page.

Page 9: *vreg.asp* – Figure 12 on page 46

This page takes the information submitted by *reg.asp* and checks the validity of the data. This includes a valid e-mail address and user name. It then queries the database for the username to check for duplicates. If a duplicate is detected, it will redirect the user to the *reg.asp* page and notify them that someone else has already claimed the username that they have chosen. It will also check to see if another user has used the e-mail address; this is to prevent a user from multiple accounts.

If all the information is valid, it then displays a confirmation message containing all of their information asking them if it is correct. At this point the user has two options,

the first is to click ‘Back’ and be taken back to *reg.asp* to make any modification that he or she wants; or to click ‘Next’ and create the account.

Page 10: *regok.asp* – Figure 13 on page 47

This page takes all the information that the user submitted and verified on *vreg.asp* and makes an entry in the database. It then creates an e-mail with a link for the user to click to activate their account. The link will be to *vnewacct.asp* and have a query string attached to identify what user is activating his or her account. This will ensure that the user has a valid e-mail account and to secure that one user cannot register more than once with the same e-mail account.

Page 11: *vnewacct.asp* – Figure 14 on page 49

The purpose of this page is to verify a user and activate his or her account. The page is displayed when a user receives an e-mail with the activation link. When activated it reads the query string and extracts the information within it. It then queries the database to validate the information in the query string. If it is invalid, the user will receive a message that the link they have used is invalid. If the information validates, then the page queries the users record from the database and activates the account, if not already activated. The user then receives a notification that the account has been activated and allows them to continue into the site. If the account is already active, then they receive a message stating that their account is already active and allows them to continue into the site.

Page 12: *ratebiz.asp* – Figure 15 on page 51

This page links off of the business listings page and allows the user to rate the business that they are currently viewing. If the user is not logged on already, they will be

presented with a login page (*login.asp*) where they may do so; once they are authenticated they are redirected to the business-rating page where they can begin to enter their feedback. If the user has already submitted a rating for the selected business, they are redirected to *reratebiz.asp* where they are able to modify the existing rating they have on file.

This page allows each user to rate a business in three areas, the quality of the service they received, the value of the service they received and if they would go back. Then the user must answer questions that pertain to that particular type of business. These are pulled from a library that is tied to the category that the business falls into. Once the user has completed the form they will be sent to *vratebiz.asp* to verify that all the information they have entered is correct.

Page 13: *reratebiz.asp* – Figure 16 on page 53

This page is presented to the user if they choose to rate a business that they have previously rated. This page displays the same information as *ratebiz.asp* but it already has all the questions filled out with the user's previous answers. It allows the user to modify their answers to the questions and re-submit the rating. When the user submits the page they are sent to *vratebiz.asp* to verify that all the information that they have entered is correct.

Page 14: *vratebiz.asp* – Figure 17 on page 55

This page displays the feedback that the user entered on *ratebiz.asp* and allows them to go 'back' to make changes or 'next' to go to *enterrate.asp* and update their feedback into the database for that particular business.

Page 15: *enterrate.asp* – Figure 18 on page 56

This page makes the update or addition to the database for a particular business and a particular user. It does all of the processing of updating/adding the record. When completed it displays a congratulations message and gives them a link back to the business information page so they can see how their opinion has made a difference.

Page 16: *login.asp* – Figure 19 on page 57

This page allows the user to login so that they can rate businesses. It prompts the user for their username and password and logs them onto the system. If the credentials the user enters are incorrect they are told why they could not log in. If the credentials are valid they are logged into the system.

Page 17: *logoff.asp* – Figure 21 on page 60

This page allows the user to logout of the system. When they initially enter the page they are asked if they would like to log out with a confirmation button. This allows the user to change their mind. If the user confirms logout by clicking the Confirm button, they are logged out of the system.

Page 18: *about.asp* – Figure 21 on page 61

This page displays an about page for JustWent.Com.

4. Objectives of the project ("Deliverables")

The following were the completion objectives:

- A fully functional, interactive, intuitive web site.
- A scaleable web application where additional functionality can be added with minor changes.

- An efficient database design that can handle a large number of simultaneous transactions.
- Efficient coding of Active Server Pages to provide efficient performance when the pages are executed and when site traffic is high.
- Fully functional search capabilities.
- Effective user registration and verification using e-mail as a carrier for verification/cancellation links.

5. Design and Development

5.1 Budget

Hardware Requirements:

NT Web Server running IIS 4.0 or IIS 5.0	$4,000
Database Sever running SQL 7.0 or SQL 2000	$4,000
Dedicated high speed Internet connection (Leased Line)	$1,500.00/month

Software Requirements:

Microsoft Visual InterDev	$550.00
Microsoft SQL Server v7.0 or SQL 2000	$1,000.00
Internet Information Server 4.0 or 5.0	Free with Win NT or 2000, Respectively.
Microsoft Visual Basic	$900.00

Other Requirements:

Domain Name Registration for JustWent.Com	$19.00/year
Dedicated Hosting with SQL Server	$50.00/month

5.2 Timeline

Senior Design I – Winter 1999

Research and information

Design and grow initial idea

Preliminary site research

Preliminary database schema

Senior Design II – Summer 2000

Week 1
Review site layout and overall functionality.
Review table schema and make any necessary changes.
Week 2
Set up database and tables on SQL Server.
Begin initial site creation.
Week 3
Research ASP information to develop necessary functionality.
Continue with site creation.
Week 4
Continue research on ASP and any other necessary web technologies.
Continue with site creation.
Week 5
Set up basic site and page layout.
Populate database with data.
Week 6
Continue with site creation.
Week 7
Finalize and debug site. Prepare site for working prototype stage.
Week 8
Work on any additional issues that arise with project.
Week 9
Prepare for presentation in week 10.
Week 10
Demonstrate working prototype.

Autumn Quarter – Autumn 2001

Throughout the quarter

Work on final product.

Senior Design III – Winter 2001

Week 1
Begin work on COM objects.
Enhance site functionality.
Week 2
Implement and test COM objects.
Test stored procedures.
Week 3
Work on sites graphical/visual aspect.
Continue to update the site.
Week 4
Finalize and complete any graphical/visual aspects.
Test site functionality.
Week 5
Begin documentation.
Week 7
Complete any coding/graphic tasks.
Week 8
Finish documentation.
Prepare for final presentation.
Week 9
Submit document and final project.
Prepare for final presentation.
Week 10
Demonstrate final product.

5.3 Software

The software requirements for this project include:

- Microsoft Visual Studio (Visual InterDev and Visual Basic 6.0)
- Microsoft SQL Server 7.0
- Windows NT or Windows 2000

5.4 Hardware

The hardware requirements for this project are:

- Web Server running Windows NT v4.0 or Windows 2000 Server
- Database Server running SQL Server v7.0 or SQL Server 2000
- Client PC for testing

6. Proof of Design

The architecture of this project is comprised of three components, the client, web server and database server. The diagram for this architecture can be found in Figure 2 below.

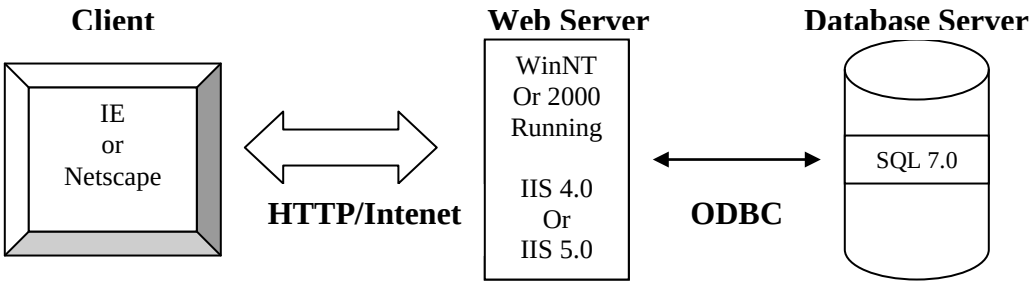


Figure 2. Web Application Architecture.

The application can be used on any client running a web browser that supports HTML and JavaScript (v1.0) such as Internet Explorer and Netscape. The project's application files reside on the Web Server which must be running either Internet Information Server (IIS) version 4.0 or version 5.0; IIS is shipped with Windows NT4.0 and Windows 2000, respectively. Finally, a database server running SQL 7.0 is used as the data warehouse for the application.

The application requires a strong database backend that is needed to house all the data in addition to managing all the connections and processing the commands sent from the application. SQL Server was chosen for three reasons, the first, SQL Server is an ODBC compliant database. The second, SQL Server's integration into Windows; because of this it can use either an ODBC or OLE DB connection. OLE DB is a database connectivity implementation by Microsoft that allows a faster more efficient connection between the database and web server; because of this, the connection between the web

server has been set up as OLE DB, although ODBC can be used resulting in some performance penalties (22, p.258). Third, SQL Server can handle multiple connections and provides industrial strength support for transactions.

I have created a SQL Server database "rbrownsd" consisting of 14 tables illustrated in Figure 3.

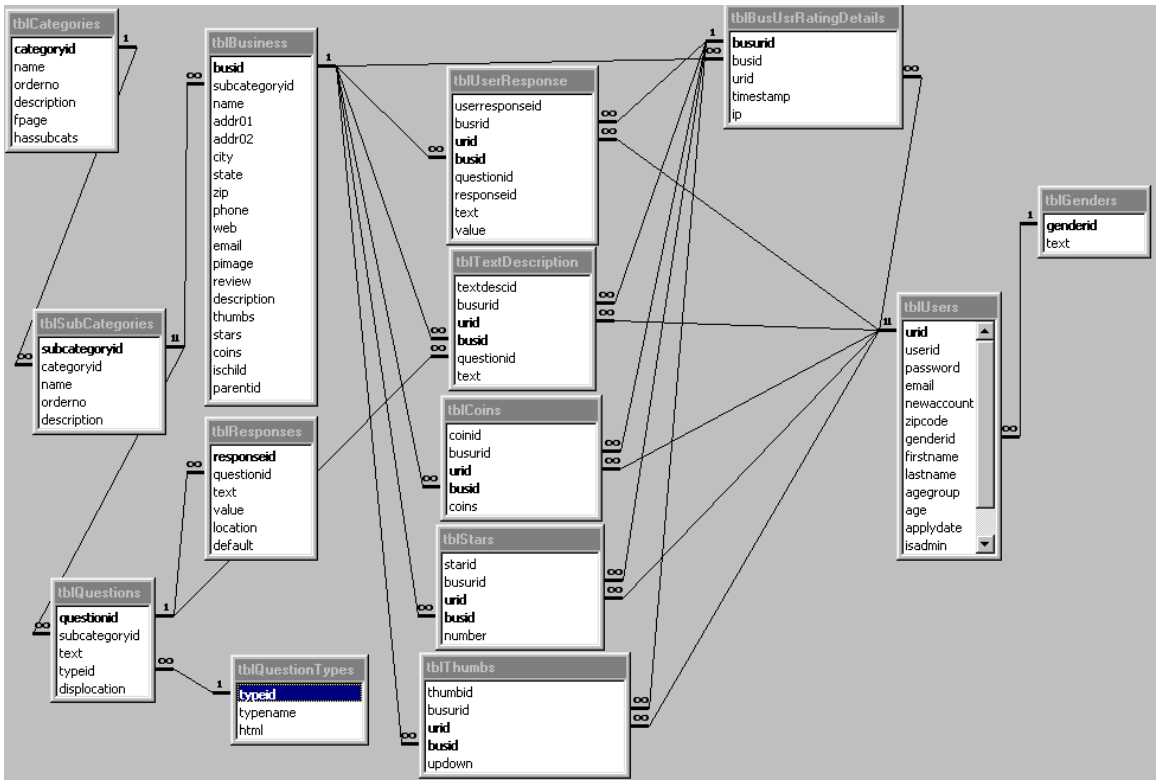


Figure 3. Web Application Database Schema.

The SQL Code to create the tables can be found in Appendix B.

The tables in the database are:

- tblCategories – This table stores all the master category names for the web application; the primary key for the table is *categoryid*.

- tblSubCategories – This table stores all the subcategories for a particular category; the primary key for the table is *subcategoryid* and references the table tblCategories by the *categoryid* field.
- tblQuestions – This table holds all the questions in the database; the primary key for this table is *questioned*. Every entry in the table is related to a specific subcategory, the field name *subcategoryid* points to the appropriate subcategory in tblSubCategories.
- tblQuestionTypes – This table holds information on what type of question this is and how it is displayed on the client's machine; its primary key is *typeid*.
- tblResponses – This table holds all the possible responses for a particular question; it's primary key is *responseid*. Every entry in this table is related to an entry in the table *tblQuestions* the field *questionid* points to the entry in *tblQuestions*.
- tblBusiness – This holds all the information about the businesses in the application; it's primary key is *busid*. Each business is related to one subcategory the field *subcateogoryid* points to an entry in the table tblSubCategories *subcategoryid* field.
- tblUsers – This table holds information about the users on the system; it's primary key is *urid*. This table holds the user's userid, password, e-mail address, first name, last name, application date, gender id, zipcode and age and age group.
- tblGenders – This table holds information about the gender types; it's primary key is gendered. This table is used to display the different genders, male and female. It is related to the table tblUsers by the field *genderid*.
- tblBusUsrRatingDetails – This table holds a primary key for each unique user to business rating; it's primary key is busurid. The table automatically generates a

busurid for each unique entry of businessid and userid. The primary key is used in the tables' tblUserResponse, tblTextDescription, tblCoins, tblStars and tblThumbs. This was done because it is more efficient to query by one field instead of by two fields in the related tables; therefore boosting query response time.

- tblUserResponse – This table holds all the responses for questions that have a numeric or short text answer (less than 255 characters); the primary key on this table is *userresponseid*. This table will hold a numeric response id from the related table *tblResponses*. This table is related to four tables, the first is *tblBusUsrRatingDetails* by the field *busurid*. The second relationship is to *tblUsers* by the field *urid*, the third, to the table *tblBusiness* by the field *busid*; and finally to the table *tblQuestions* by the field *questionid*.[1]
- tblTextDescription – This table holds all the long text answers, 255 characters or less; the primary key on this table is *textdescid*.
- tblCoins – This table holds the user's rating of how expensive the business is; it's primary key is *coinid*. Each entry in this table is related to a distinct user and business in *tblBusUsrRatingDetails* by the field *busurid*.
- tblStars – This table holds the user's rating of how many stars each business receives; it's primary key is *starid*. Each entry in this table related to an entry in *tblBusUsrRatingDetails* by the field *busurid*.
- tblThumbs – This table holds the user's opinion if they will go back to the business or not; it's primary key is *thumbid*. Each entry in this table is related to a distinct user and business in *tblBusUsrRatingDetails* by the field *busurid*.

Each of these tables has been normalized to the *3rd normal form* with one exception; there are two redundant data fields in five of the tables (*tblUserResponse, tblTextDescription, tblCoins, tblStars, and tblThumbs*). The first redundant data field is *urid, which* relates to the *urid* field in the *tblUsers* table. The second is *busid*, which relates to the busid field in the table tblBusiness. The redundancy is used to speed up execution of certain queries, resulting in faster data retrieval. For example, if the code needed to query all the users who said that they would go back to a particular business, the SQL statement would be able to query on the table tblThumbs using the business' primary key, *busid*. If the data redundancy did not exist, the code would have to execute a SQL statement to fetch all the business-to-user IDs, *busurid,* for a particular business. Then the resulting list of id's would be used to query against the *tblThumbs* table to get a count of all the entries where the users entered they would go back, a very inefficient method.

Finally, I have included extra fields in one of the tables, *tblBusiness*; these extra fields are hooks for later enhancements to the web application. In the table *tblBusiness* I have included the fields *ischild* and *parentid*, which allows the application to recognize when an entry is part of a franchised business. In the same table, I have also included the fields, *thumbs, stars* and *coins* so that I could turn on and off the ratings about how many stars and coins a business gets and listing how many people would go back and would not go back.

In addition to the fourteen tables the project contains 28 stored procedures, which are described below. The SQL code to create the following stored procedures can be found in Appendix C.

[1] This relationship is used in the Active Server Pages where Data Shaping queries questions and responses.

- sp_AddUser – This procedure adds a new user to the table tblUsers. It takes nine input parameters with information to be added to the table and returns a recordset with two fields, the newly created user id *urid*, and the date the record was created *applydate*.
- sp_BusinessQuestionsBySubCategory – This procedure returns a recordset of questions. It takes a category id as the input parameter and returns a recordset containing five fields: the question id, question text, question type it, question type name and question type html name.
- sp_CountRatingsForBusiness – This procedure returns the count of user ratings for a particular business. It takes a business id as the input parameter and returns a recordset of one record with one field in it with the count of ratings.[2]
- sp_DeleteStarCoinThumb – This procedure deletes all records in the table *tblCoins*, *tblStars* and *tblThumbs* for a particular business-to-user-rating-id. It takes a business-to-user rating id as the input parameter and returns no recordset or output parameters.
- sp_DeleteTextDescription – This procedure deletes all records in the table *tblTextDescription* for a particular business-to-user rating id. It takes a business-to-user rating id as the input parameter and returns no recordset or output parameters.
- sp_DeleteUserResponses – This procedure deletes all the records in the table *tblUserResponse* for a particular business-to-user rating id. It takes a business-to-user rating id as the input parameter and returns no recordset or output parameters.
- sp_GetAllCategories – This procedure returns a recordset of all categories and a count of all the related subcategories for each of the entries in *tblCategories*. The procedure does not take any input parameters. It returns a recordset containing five fields:

[2] Due to the overhead of creating a recordset from a stored procedure, this procedure will be re-coded in future enhancements to return the count of ratings as an output parameter instead of a recordset.

category id, category name, an order number field,[3] if it has subcategories and a count of subcategories related to this record.

- sp_GetAllPossibleResponses – This procedure returns a recordset of all the entries in the table *tblResponses*. The procedure does not take any input parameters. It returns a recordset containing six fields the response id, related question id, response text, response numerical value, display position, default answer; in order by question number and display position.
- sp_GetAllPossibleResponsesByQuestionID[4] – This procedure returns a recordset of all entries in the table *tblResponses* for a particular question id. The procedure takes a question id as the input parameter and returns a recordset containing the same six fields as *sp_GetAllPossibleResponses*.
- sp_GetAllSubCategories – This procedure returns a recordset of all the entries in the table *tblSubCategories* and a count of all related businesses. The procedure does not take any input parameters. It returns a recordset of five fields the category id that the entry is related to, the subcategory id, the name of the subcategory, the display position and a count of all the businesses that fall under this subcategory.
- sp_GetBusiness - This procedure returns a recordset of businesses that are related to a specific category id. It takes one input parameter, the subcategory id and returns a recordset containing two fields, the business id and the name of the business.
- sp_GetGeneralBusinessInfo – This procedure returns information on a particular business. It takes one input parameter, the business id and returns a recordset containing twenty-four fields. These fields are the business name, the address (city,

[3] Presently not used, the recordset is in order by the name of the category.
[4] Modified version of *sp_GetAllPossibleAnsers*, used as a more efficient for returning related responses.

state and zip code), its phone number, web address, e-mail address, a pointer to an image of the business, a review of the business.[5] The procedure also returns statistical and extended information about the business; the category and subcategory name that the business falls into, the sum of the people who would go back, the sum of the people who would not go back, a count of the number of entries in the tables *tblStars* and *tblCoins* for this business, and finally the sum of the total number of stars as well as the sum of the total number of coins for this business.

- sp_GetQuestionResponsesAndTotals – This procedure returns all the responses from the table *tblResponses* with a total of how many times this response appears in the table *tblUserResponse* for a particular business. This procedure takes one input parameter a business id, used for calculating the number of responses for that business. It returns a recordset containing four fields, the related question id for this response, the response id, the response text and a count of how many times this response has been chosen for a particular business (based on business id).
- sp_GetQuestionsByBusinessID – This procedure returns a recordset with all the questions for a particular business. It has one input parameter, the business id. It takes the business id, queries the subcategory id and then returns a recordset with all the questions for that particular subcategory id. If no such business exists or no subcategory id exists for the business it will execute the same query but with criteria that will return an empty recordset. This procedure returns one recordset containing two fields, the question id and the type id.
- sp_GetSubCategories – This procedure returns a recordset of all the subcategories for a particular category. The procedure has one input parameter, the category id and

[5] The *review of the business* is presently an unused hook for future enhancements.

returns one recordset containing four fields. The fields in the returned recordset are, the subcategory id, the name of the subcategory, the display order number, and a count of the number of businesses that fall into this subcategory.

- sp_GetUserResponseCoinStarThumb – This procedure returns the user's previous entries of stars, coins and if they would go back for a particular business. The procedure takes one input parameter, the business-to-user rating id. It returns three output parameters, a value in *tblCoins* with the number of coins they had previously entered, a value in *tblStars* with the number of stars for this particular business and a value from *tblThumbs* associated if they would or would not go back to the business.
- sp_GetUserResponsesBasedOnBUSURID – This procedure returns a recordset containing the responses for a particular business-to-user id from the table *tblUserResponse*. The procedure takes one input parameter, the business-to-user id and returns one recordset. This recordset contains four fields, the corresponding question id, the response id, the answer text and the answer numerical value. Each of the fields returned are prefixed with "tUR" (i.e. questionid is returned as tURquestionid).
- sp_GetUserTextResponsesBasedOnBUSURID – This procedure returns a recordset containing the responses from the table *tblTextDescription*. The procedure takes one input parameter, the business-to-user id and returns one recordset. This recordset contains two fields, the question id of the corresponding question and the actual text answer that the user had previously entered. Each of the fields returned are prefixed with "tTD" (i.e. questionid is returned as tTDquestionid).

- sp_GetUserTimeStamp – This procedure returns a recordset with the users application date that is stored in *tblUsers*. The procedure takes one input parameter, the user id and returns a recordset containing one record and one field, the application date.[6]
- sp_HasUserRatedBusiness – This procedure returns a count of how many times a user has rated a particular business. It has two input parameters, the user id and business id. It returns a count of the number of business-to-user rating ids' for that particular user id and business id in the form of an output parameter.
- sp_HasUserRatedBusinessWithID – This procedure determines if a user has rated a particular business and if so return the business-to-user id. It takes two input parameters, the user id and the business id. It then queries a count of business-to-user rating IDs if the count is zero then it sets the business-to-user rating id return value to zero. Otherwise it will query the business-to-user id and return the id back as an output parameter.
- sp_HasUserRatedBusinessWithID_CreateNoExist – This procedure determines if a user has rated a particular business and if not create an entry. It takes three input parameters, the user id, the business id and the IP address of the user. It queries a count of business-to-user rating ids', if the count is zero then it creates a new entry and stores the newly created business-to-user id. If the count is greater than zero it queries the business-to-user id and stores it. It has two output parameters the business-to-user id that was created or retrieved and the count of business-to-user ID's for that particular user (either 0 or 1).

[6] Due to the overhead of creating a recordset from a stored procedure, this procedure will be re-coded in future enhancements to return the timestamp as an output parameter instead of a recordset.

- sp_InsertNewRatingGetID – This procedure takes three input parameters, creates a new business-to-user rating id and returns it to the calling code as an output parameter. The input parameters are the user's id, the business id and the user's IP address. The return value is the newly created business-to-user rating id.
- sp_InsertStarCoinThumb – This procedure enters the users rating for the number of stars and coins for a business into *tblStars* and *tblCoins*, respectively. In addition, it also enters if the user would go back to the business or not into table *tblThumbs*. The procedure takes five input parameters, the number of stars the user selected, the number of coins that the user selected, if the user would go back or not, a valid business-to-user rating id, the user's id and the business' id. It does not return a recordset or output parameters.
- sp_InsertTextDescription – This procedure enters a response into the table *tblTextDescription*. It has five input parameters, the business-to-user rating id, the user's id, the business' id, the related question id, and the actual text entered by the user. It does not return a recordset or output parameters.
- sp_InsertUserResponses – This procedure enters a response into the table *tblUserResponse*. It has seven input values, the business-to-user rating id, the user's id, the business id, the related question id, the related response id (if any), the text of the question and the text response, if any, entered by the user. The procedure does not return a recordset or any output parameters.
- sp_ValidateBusinessWithExtraInfo – This procedure checks to see if a business id is valid. The procedure takes one input parameter, the business id and returns three output parameters. The parameters returned are a flag *@isvalid* which is '0' if the

business id is not valid or '1' if the business id is valid. In addition, it returns the subcategory id and the name of the business if the business id is valid; if the business id is invalid then these parameters are empty strings.

- sp_ValidateBusinessWithInfo – This procedure is exactly the same as *sp_ValidateBusinessWithExtraInfo*.

These stored procedures are precompiled SQL statements that are used within the application. Because these are precompiled and reside within the database, they execute much faster because the database server does not need to compile them (65, p.23). Additionally, the stored procedures provide a level of abstraction in the overall application design. Because stored procedures reside on the database they are executed in the application by calling the stored procedure function name from ADO code, thus separating the actual SQL statement from the Active Server Page code. If changes need to be made the SQL statement they are made at the database level, this means by changing one stored procedure many pages get the new functionality of the procedure.

The portion of the application that exists on the web server consists of eighteen Active Server Pages (ASP Pages), one *global.asa* file, one cascading style sheet (globalstyle.css) and one ActiveX component. The code for the ActiveX component can be found in Appendix D.

The ActiveX component is a precompiled dynamic linking library (dll) consisting of ten properties, six subroutines and five public functions. The component is used in some of the application's pages. It provides faster execution and more robust processing

of the data, in some cases it provides more functionality than using just ASP code alone (24). The twelve properties of the component are:

- bDebugInfo – This is a Boolean value, when set to true it shows debugging messages when the component's methods and subroutines are executed. This property is read and writeable.
- bJavaScriptPopUp – This is a Boolean value, when set to true the component will display a pop-up progress window on the clients browser when some of the components methods and subroutines are executed. This property is read and writeable.
- lBusinessID – This is a Long value; it contains the *BusinessID* of the current business being processed by the component. This property is read and writeable.
- lBusinessUserID – This is a Long value; it contains the *Business-To-User Rating ID* of the current rating being processed. This property is read and writeable.
- lUserID – This is a Long value; it contains the user's unique ID. This property is read and write able.
- lSubCategoryID – This is a Long value; it contains the *SubCategoryID* of the current business being processed. This property is read and writeable.
- sBusinessName – This is a String value; it contains the name of the Business that is currently being processed. This property is read only.
- sDBConnectString – This is a String value; it contains a connection string to connect to a valid ODBC or OLE DB data source. This property is read and writeable.

- sValidationFailPage – This is a String value; it contains the filename of a page to go to when data validation fails. This property is read and writeable.[7]
- sValidationSuccessPage – This is a String value; it contains the filename of a page to go to when data validation passes. This property is read and writeable.

The components six public subroutines and five public functions handle the processing of the user's rating for the application. These routines are as follows:

- Public Sub OnStartPage() – This is called when the component is instantiated in an Active Server Page. It is used to get the Active Server Page's intrinsic objects such as Response and Request.
- Public Sub OnEndPage() – This is called when the component is destroyed in an Active Server Page. It is used to clean up any objects that the component had created and are still resident in memory.
- Public Sub ShowRatingForm() – This subroutine generates a rating form for a particular business. The properties that must be set are *sDBConnectString*, *lBusinessID*, *lSubCategoryID*; *bJavaScriptPopUp* is optional.
- Public Sub ShowReRateForm() – This subroutine generates a rating form for a particular business and user; the form generated will have the users previous answers all set as the default. The properties that must be set are *sDBConnectString*, *lBusinessUserID, lSubCategoryID, lUserID.*
- Public Sub displayValidationForm() - This subroutine generates a form with the user's answers that they have just submitted. It checks to make sure that each question has an answer; if not it marks the question in red and continues checking the

[7] This is used in the displayValidationForm() method.

rest of the responses. If all of the questions have a response then it allows the user to continue to the next page specified by the property *sValidationSuccessPage*. If there are responses missing it will allow you to continue to the next page specified by the property *sValidationFailPage*. The properties that must be set are *sDBConnectString*, *lBusinessUserID*, *lSubCategoryID*, *lUserID*; *bJavaScriptPopUp* is optional.

- Public Function insertUserRatingIntoDatabase() – This function takes the users rating for a particular business and enters it into the database.[8] It begins by checking to see if the user has rated the business before, if so it obtains the *business-to-user rating id;* otherwise it generates a new one after it begins the database transaction. Next, it begins a database transaction and enters all the user's responses into the database. If it encounters a question without a response the function rollback the transaction and return *false*. Otherwise the function commits the transaction and returns *true*. The properties that must be set are *sDBConnectString*, *lSubCategoryID*, *lUserID* and *lBusinessID*; *bJavaScriptPopUp* is optional.
- Public Function isValidBusiness() – This function checks to see if the business ID set in the property *lBusinessID*, is a valid business id. If the business id is valid it returns true, and sets the properties *lSubCategoryID* and *sBusinessName*. Otherwise, it returns *false* and the properties *lSubCategoryID* and *sBusinessName* are empty. The properties that must be set are *sDBConnectString* and *lBusinessID*.
- Public Function HasUserRatedBusinessBefore() – This function checks to see if a user has rated a particular business already and returns the integer '1' if they have or '0' if they have not. If the user has already rated the business the property lBusinessUserID

[8] The data isaccessed via the Request.Form object.

is set with the business-to-user rating ID. The properties that must be set are *sDBConnectString*, *lUserID* and *lBusinessID*.

- Private Function OpenProgressWindow() – This function outputs JavaScript that opens a window on the client's browser. It takes one input parameter, a string variable *strHeading*, which will be the text that appears on the top line of the created window. It will always return the integer value '1'.
- Private Sub UpdateStats() – This function outputs JavaScript to the client's browser that will change the text in the progress window opened by *OpenProgressWindow()*. It takes six input parameters, *sHeading*, *sLine1*, *sLine2*, *sLine3*, *sLine4* and *sLine5*; all string variables that hold the text to be displayed in the window.
- Private Function CloseProgressWindow() – This function outputs JavaScript to the client's browser that closes the progress window opened by *OpenProgressWindow()*. It will always return the integer value '1'.

The eighteen Active Server Pages that make up the bulk of the web application are as follows…

Figure 4. Page 1: Index.asp

This is the default page for the web application and is used as the starting point for the application. This page displays a welcome screen to the user with some instructions on how to start. At this point the user has several options to choose from:

- They can enter a business name in the search box to find a particular listing.
- They can select one of the yellow navigation buttons:
    - Home – This will return the user to this page.[9]

[9] Page Index.asp.

- Categories – This will provide a hierarchical listing of all the categories with their associated subcategories.[10]
- Last Business – This will take the user back to the last business that they viewed.[11]
- Register – This will allow the user to sign up for an account.[12]
- Login/Logout – This will allow the user to login if they are not already and logout if they are not already logged out.[13]
- About - This will take them to an informational page about the site.[14]

- They can select a subcategory from the abbreviated listing in the center of the page.
- They can select a business that was one of the last ten rated businesses.

The "Last 10 Rated" table displays a list of the last ten businesses that have been rated and provides you a link to list the business's information. The code used to generate this is the following:

```
    <!--Last 10 business rated-->
    <table align="right" width="150px" id="lastten" border="1"
bordercolor="#81e2e3" cellpadding="0" cellspacing="0">
    <tr><td align="center" colspan="2"><font color="#000000">Last 10
Rated</font></td></tr>
   <%
   Set oConn = Server.CreateObject("ADODB.Connection")
   Set oCmd = Server.CreateObject("ADODB.Command")
   Set oRS = Server.CreateObject("ADODB.Recordset")

   oConn.Open Session("s_dsn")
   oCmd.ActiveConnection = oConn
   oCmd.CommandText = "sp_GetLastTenRatings"
   Set oRS = oCmd.Execute

   iCounter = 0
   While Not oRS.EOF
    iCounter = iCounter + 1
    lRSBusID = oRS("busid") + 0
    sRSBusName = oRS("name") & ""
    lRSBusUserRatingID = oRS("busurid")
    lRSUserID = oRS("urid") + 0
    sRSUserName = oRS("userid") & ""
    Response.Write "<tr><td width=100 id=""lastten"">" & iCounter & ".)<a
href=""business.asp?bid=" & lRSBusID & Chr(34) & ">" & sRSBusName & "</a></td>"
    Response.Write "<td width=50 id=""lastten"">" & sRSUserName & "</td></tr>"
& vbCrLf

    oRS.MoveNext
   Wend
   Response.Write "</table>"
   If oConn.State = 1 Then
```

[10] Page Allcats.asp.
[11] Page Business.asp.
[12] Page Reg.asp.
[13] Page Login.asp or Logout.asp, respectively.
[14] Page About.asp.

```
 oConn.Close
End If
If oRS.State = 1 Then
 oRS.Close
End If
Set oCmd = Nothing
Set oRS = Nothing
Set oConn = Nothing
%>
</table>
```

This code opens a connection to the database, executes the stored procedure *sp_GetLastTenRatings* and displays each one using a while loop.[15] It will then close the recordset and connection to the database and destroy the command, recordset and connection objects by setting the equal to nothing.

[15] This feature was added during final development to show ease of scaleability and extensibility of the framework.

Figure 5. Page 2: Allcats.asp

This page displays a hierarchical display of all the categories and their related subcategories that the user can select from. Each subcategory has a number following it, which is the total number of businesses that fall into that particular subcategory. Each subcategory is a link that the user can click on to get a listing of the businesses that fall under that subcategory.

The hierarchical display of the categories and subcategories is created using the ADO Shape command (41). A complex series of related SQL statements are executed all

at once by executing them in a shape command. This is demonstrated in the following code:

```
   sShapeCommand = "SHAPE {{CALL dbo.sp_GetAllCategories}} APPEND({{CALL
dbo.sp_GetAllSubCategories}} as rsSubCats RELATE categoryid TO categoryid)"

      'Set the provider to Microsoft Data Shaping
      oConn.Provider = "MSDataShape"
      oConn.Open Session("s_dsn") 'Open the connection
      oCmd.ActiveConnection = oConn 'Set the active connection
      oCmd.CommandText = sShapeCommand  'Use the shape command text

      Set oCategoryRS = oCmd.Execute 'Get the recordset back
```

When this command is executed a recordset is returned with all the categories, within this recordset is a field, *rsSubCats*, which is a pointer to a sub-recordset of all the subcategories for that particular category. The categories and subcategories are then displayed on the browser using the following code:

```
      Response.Write "<ul>" & vbCrLf
      While Not oCategoryRS.EOF
        Response.Write "<li>"
        Response.Write oCategoryRS("name") & "  (" &
oCategoryRS("total") & ")<br>"
        Response.Write "</li>" & vbCrLf

        Set oSubCatsRS = oCategoryRS("rsSubCats").Value

        Response.Write "<ul>"
        While Not oSubCatsRS.Eof
          Response.Write "<li>"
          Response.Write "   " & "<a href=" & Chr(34) &
"blist.asp?scid=" & oSubCatsRS("subcategoryid") & Chr(34) & ">" &
oSubCatsRS("name") & "</a> (" & oSubCatsRS("total") & ")<br>"
          Response.Write "</li>" & vbCrLf
          oSubCatsRS.MoveNext
          Response.Flush
        Wend
        Response.Write "</ul>" & vbCrLf

        oCategoryRS.MoveNext
      Wend
      Response.Write "</ul>" & vbCrLf
      Response.Write "<br>"
```

Finally, all recordsets and connections to the database are closed; the recordsets, connection and command objects are set equal to nothing, destroying them.

Figure 6. Page 3: Subcatlist.asp

This page displays a list of all the subcategories that are related to a particular category. Each subcategory that is displayed is a link to the list of businesses that fall into that particular subcategory.

When the page is executed, the code checks to see if the query string contains a value for *catid*, which is a category id. If this value does not exist or is not numeric then the user is automatically redirected back to the index.asp page. If the extracted category id is valid then the following code is executed:

```
   oConn.Open Session("s_dsn")
      oCmd.ActiveConnection = oConn
      oCmd.CommandText = "sp_GetSubCategories"
      oCmd.CommandType = 4  'adCmdStoredProc
      oCmd.Parameters.Append oCmd.CreateParameter("categoryid", 3, 1, 4,
iCatID)


      Set oSubCategoryRS = oCmd.Execute

      If Not oSubCategoryRS.EOF Then
         'Display all the records retrieved from the Stored Procedure.
```

```
        While Not oSubCategoryRS.EOF
          Response.Write "<a href=" & Chr(34) & "blist.asp?scid=" &
oSubCategoryRS("subcategoryid") & Chr(34) & ">" & oSubCategoryRS("name") &
"</a>  (" & oSubCategoryRS("total") & ")<br>"
          oSubCategoryRS.MoveNext
        Wend
      Else
        Response.Write "<center>There are no subcategories for this particular
category.</center>"
      End If


     oSubCategoryRS.Close 'Close the recordset
     oConn.Close 'Close the connection
     Set oSubCategoryRS = Nothing 'Clean up after yourself!
     Set oCmd = Nothing
     Set oConn = Nothing
```

A connection to the database is opened and the stored procedure *sp_GetSubCategories* is executed returning a recordset. If the recordset is empty then the user is given a message that the selected category has no subcategories. If the recordset is not empty then the user is give a list of subcategories with the total number of businesses that are listed in the respective subcategory. Each subcategory is displayed as separate link to a page that lists all the businesses for that subcategory (*blist.asp*). The connection and the recordset are then closed and all the objects are destroyed.

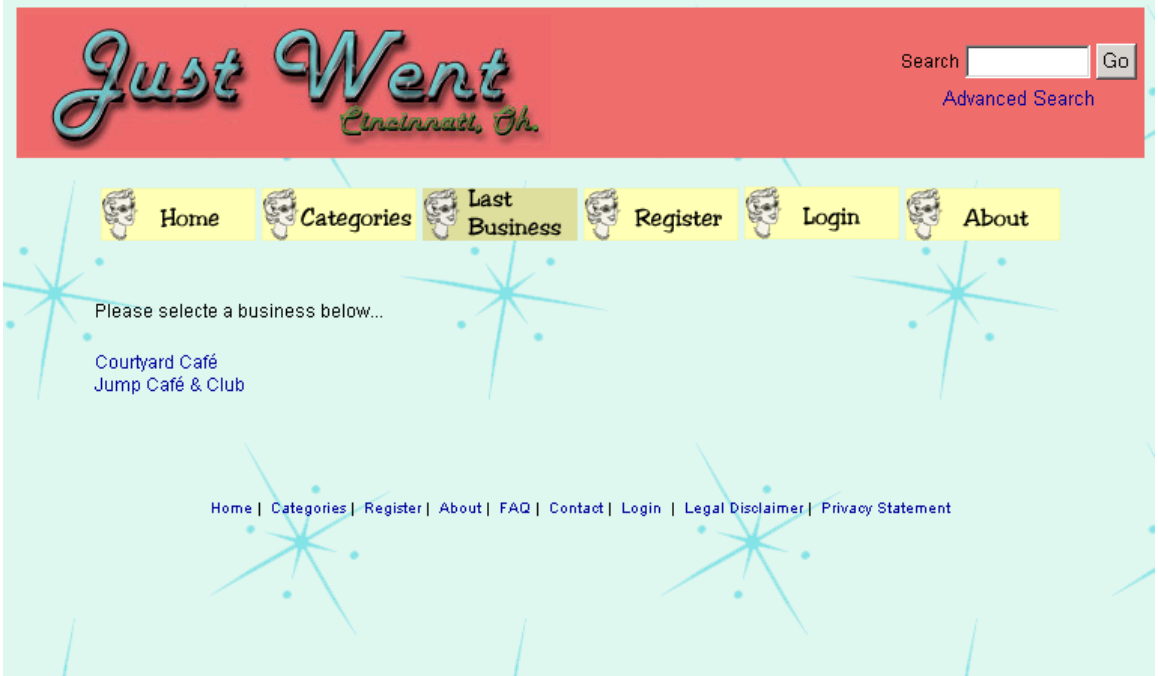


Figure 7. Page 4: Blist.asp

This page displays all the businesses that fall under a particular subcategory. This page is called by passing a query string in the URL with the only parameter being the subcategory id, *scid*. The code extracts the value for the subcategory id and the following code will execute to display a list of the associated businesses.

```
    oConn.Open Session("s_dsn")
    oCmd.ActiveConnection = oConn
    oCmd.CommandType = 4  'adCmdStoredProc
    oCmd.CommandText = "sp_GetBusiness"
    oCmd.Parameters("@subcategoryid") = iSubCatID
    Set oBusRS = oCmd.Execute

    While Not oBusRS.EOF
      Response.Write "<a href=" & Chr(34) & "business.asp?bid=" &
oBusRS("busid") & Chr(34) & ">" & oBusRS("name") & "</a>"
      Response.Write "<br>"
      oBusRS.MoveNext
    Wend
    oBusRS.Close
    oConn.Close
    Set oBusRS = Nothing
    Set oCmd = Nothing
    Set oConn = Nothing
```

A connection is then opened to the database and the stored procedure *sp_GetBusiness* is executed using the ADO Command object's *execute* method. The command returns a recordset and the results are displayed to the browser using a While loop. The recordset and the connection are then closed and all three objects will be destroyed by setting them equal to Nothing.

Figure 8. Page 5: Business.asp

This page (Figure 8) displays information about a selected business to the user. The information includes how many users have rated the business, what the average number of stars that the business has received, the average number of coins the business has received and how many people would and would not go back. In addition, the business name, mailing address, web address and e-mail address are displayed along with a photo; if no photo is available, a "No Photo Available" image will be displayed instead. This screen displays the selected business and all the info about that business. The query that is executed to get this section of information is the following:

```
  '2.) Now get the Business Information from the database
  '-------------------------------------------------------
    'Tell it what stored procedure to execute
    oBusInfoCmd.CommandText = "sp_GetGeneralBusinessInfo"
    oBusInfoCmd.CommandType = 4 'adCmdStoredProc Tell it that this is a stored
procedure
    oBusInfoCmd.Parameters("@busid") = iBusID  'Set the parameters needed.
    Set oBusInfoRS = oBusInfoCmd.Execute  'Execute it!
```

After the information about the business is displayed, a list of questions and responses are displayed. These questions are generated depending on what subcategory that the business falls under. Under each question are all the possible responses with the number of people who chose that response in the form of a percentage. This information is queried from the database using ADO Data Shaping. The following code is used to extract the hierarchical question and response data for the business:

```
   'Now lets process those questions...
      oConnShape.Provider = "MSDataShape"
      oConnShape.Open Session("s_dsn")
      oCmdQues.ActiveConnection = oConnShape


       sSQL="SHAPE {{ CALL dbo.sp_BusinessQuestionsBySubCategory(" & iSubCatID &
")}} "
      sSQL = sSQL & "APPEND ({{ CALL dbo.sp_GetQuestionResponsesAndTotals(" &
iBusID & ")}} "
      sSQL = sSQL & "RELATE questionid TO questionid) AS responseinfo"
```

```
oCmdQues.CommandText = sSQL
Set oQuestionRS = oCmdQues.Execute 'Execute the statement.
```

A doubly nested While loop will then display the data to the browser. After all the questions and responses have been displayed to the browser, the associated recordsets and connections are closed. The objects are then destroyed to avoid any memory leaks.

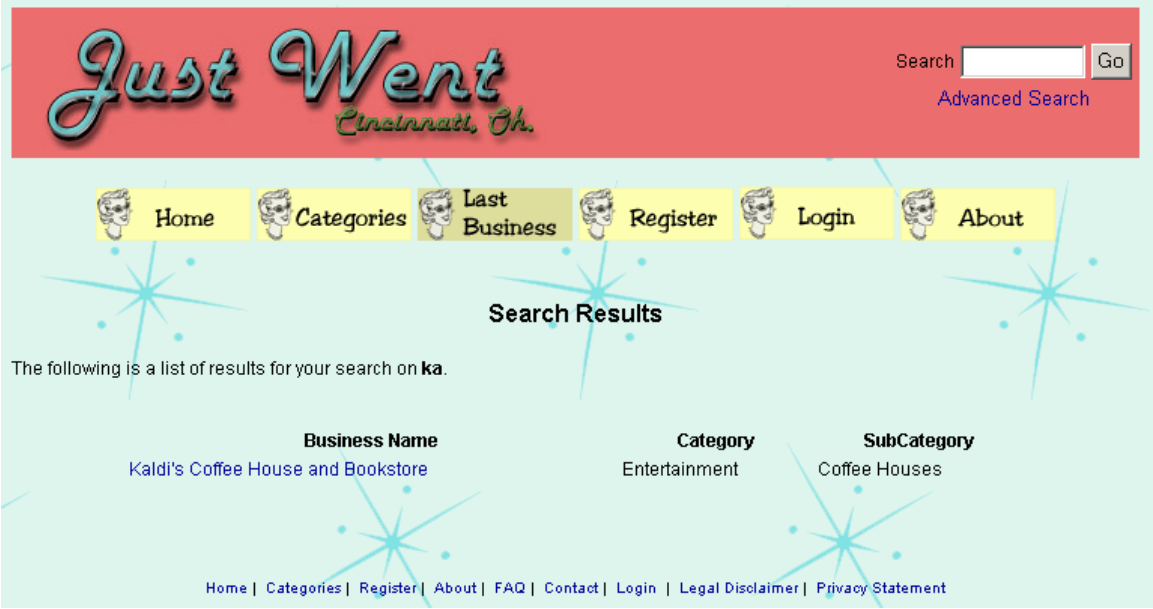


Figure 9. Page 6: Search.asp

This page displays a listing of all the businesses that matched the search criteria entered in the search textbox displayed at the top of each page. Upon execution, this page takes the business name entered and queries the database for businesses that match using the SQL Soundex function and the Like keyword. The results are displayed to the user ordered by category name, subcategory name and business name; each result returned is a hyperlink to the business information page for that particular business.

.

Figure 10. Page 7: Asearch.asp

This page allows the user to search for a particular business or group of businesses based on three different types of criteria, a business name, a zip code and by subcategory. The user may enter data for as many of the search areas as needed; but at least one must be filled out to conduct the search. When the form is filled out and submitted the page reloads itself and extracts the information the user had submitted in

the form. It then dynamically builds a SQL query string depending on what data has been entered and queries it against the database. The results are listed in order of category name, subcategory name and business name; each result is a hyperlink to that particular businesses information page.

Figure 11. Page 8: Reg.asp

This page is the first page of the user registration process. There are two different methods that this page can be executed. The first is if the user signs up for an account. When the page is executed this way it will check to see if the user is already logged in. If

so, they are given a message stating that they are already logged in and they must logout if they would like to create a new account. The code does this is:

```
If Session("b_loggedin") = True Then
  'Tell the user that they are already logged in.
  Response.Write "<H3 align=center>Already Registered?</h3>" & vbCrLf
  Response.Write "<P align=left>It looks as if you are already logged in. "
  Response.Write "If you would like to make changes to your user profile then
click <u>here</u>"
  Response.Write "otherwise, you may logout <u>here</u>.</p>"
```

If the user is not logged in they are asked to fill in a registration form and click the submit button at the bottom of the page to continue.

The second way, that this page can be run, is from the registration verification page, *vreg.asp*. When the registration verification page is executed it checks for required information, if any is missing it redirects back to this registration page and appends a query string so that the user's information is repopulated and an appropriate error message displayed.

Figure 12. Page 9: Vreg.asp

When this page is executed, it checks to see if all the information that was required on the initial registration page, *reg.asp*, has been entered. If not, it redirects the user back to the initial registration page, with an appended query string so that the registration page is displayed with the original information and an appropriate error message displayed.

If all the information is correct the user will be asked to verify the information and press the “Next” button to create the account.

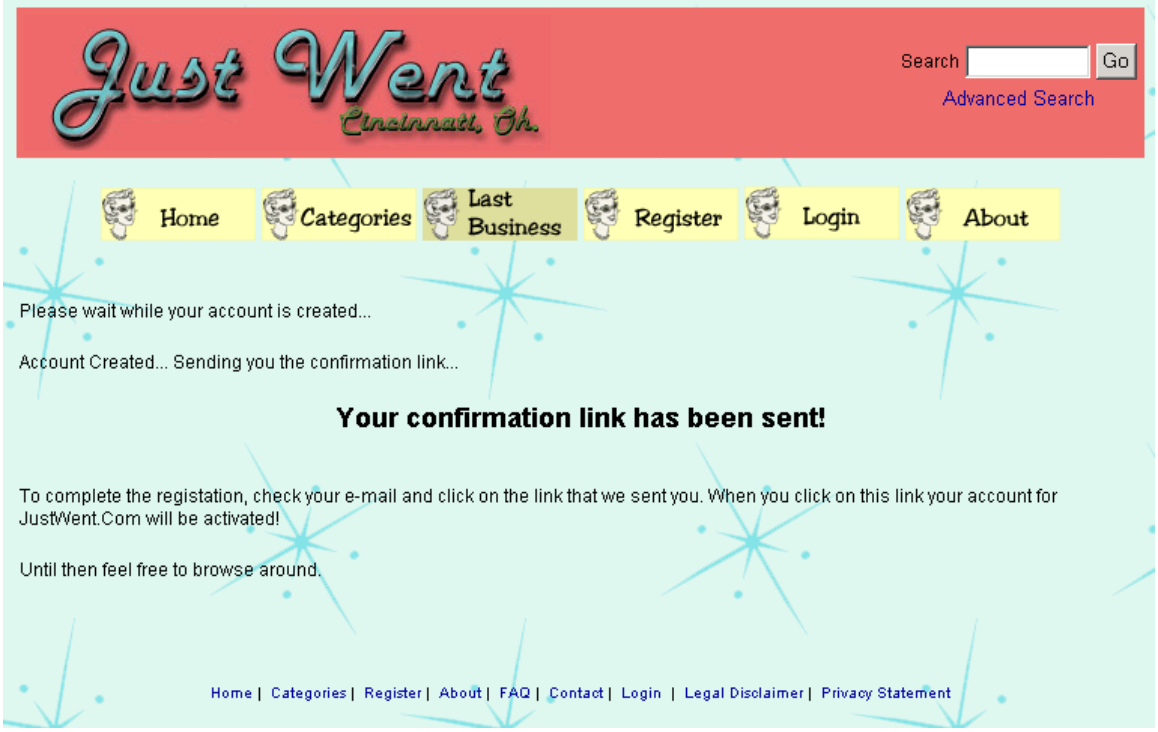


Figure 13. Page 10: Regok.asp

This page is executed when the users clicks the "Next" button on the registration verification page, *vreg.asp*. When it executes it checks to see if all the required information is available; if not, it will redirect to the initial registration page, *reg.asp*, with an appended query string so that the user receives an appropriate error message and can make changes. If all the information is valid it will then insert the user's registration data into the table *tblUsers*. At this point the user's account is still inactive. The page then sends out an e-mail to the user, at the e-mail address entered in the registration form, so the user can activate the account. The code to send out the e-mail uses the Collaborative Data Objects for NT, CDONTS, and is listed as follows:

```
Set oNTMail = Server.CreateObject("CDONTS.NewMail")
'oNTMail.From = "browrp@email.uc.edu"  'Set up who it is from.
oNTMail.From = "register@JustWent.com"
oNTMail.To = sEmail 'Set up who it goes to
oNTMail.Subject = "Just Went Registration Confirmation" 'Set up the subject
line.
'Set up the body of the text.
sMsgBody = sFirstname & " " & sLastname & "," & vbCrLf
```

```
sMsgBody = sMsgBody & "This e-mail is for you to confirm your registration at
JustWent.Com... "
sMsgBody = sMsgBody & " To confirm your registration and become a user at
JustWent.Com please "
sMsgBody = sMsgBody & " click the link that says Confirm below.  " & vbCrLf &
vbCrLf
sMsgBody = sMsgBody & "If you did not register with JustWent.Com please click
the link that says "
sMsgBody = sMsgBody & "Not Me below..." & vbcrLf
sMsgBody = sMsgBody & "Confirm -> " & sGoodLink & vbCrLf & vbCrLf
sMsgBody = sMsgBody & "Not Me -> " & sBadLink & vbCrLf & vbCrLf
sMsgBody = sMsgBody & "Thank you for taking the time to visit
www.JustWent.Com!" & vbCrLf
sMsgBody = sMsgBody & "-Robert Brown"
'Assign the body of the text to the message.
oNTMail.Body = sMsgBody

oNTMail.Send() 'Send the message
Set oNTMail = Nothing 'Clean up the object after your done...
```

The e-mail contains two links, the first is to activate the account and the second will be to delete the account. The second link has been added so that the application does not accumulate people's information that did not originally sign up for the service.

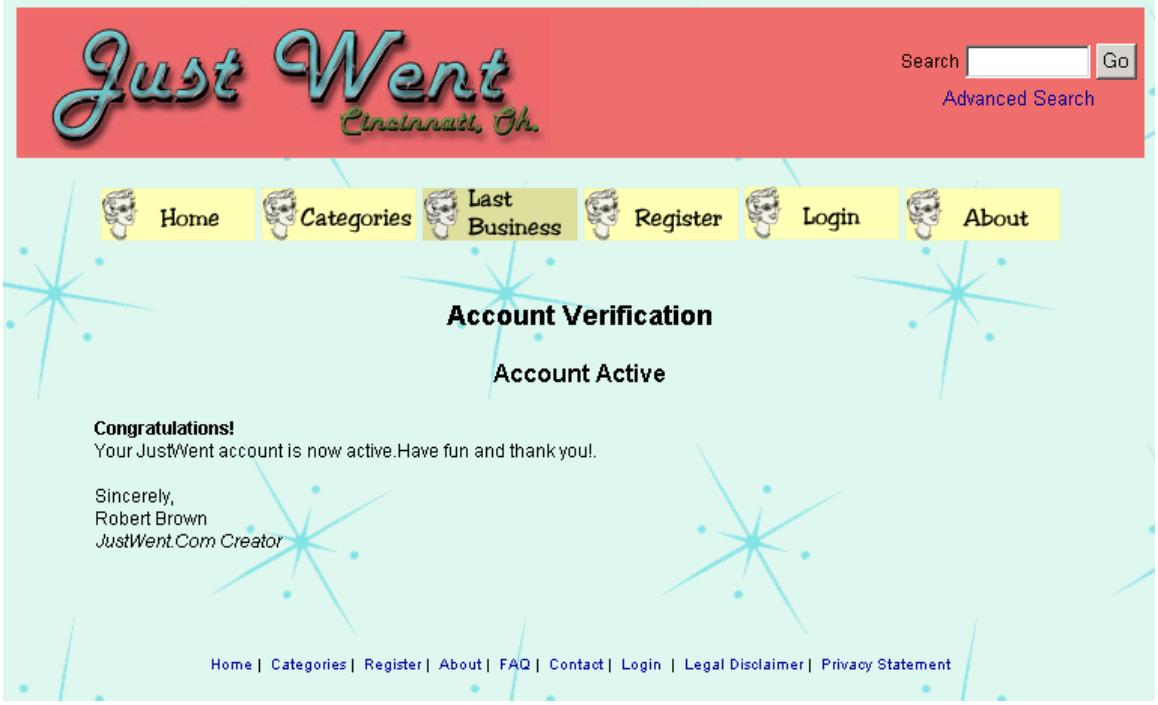


Figure 14. Page 11: Vnewacct.asp

This page is executed when the user selects one of the links from the e-mail that the application sent in the final registration page, *regok.asp*. Upon execution the page extracts the query string data from the link that the user selected. It first validates that all the required information is present using the following code:

```
'Grab all the values.
lUserID = Request.QueryString("u")
sCode = LCase(CStr(Request.QueryString("s") & ""))
sValidChecksum = CStr(Request.QueryString("d") & "")

If lUserID = 0 or sCode="" or sValidChecksum="" Then
      Response.Redirect "index.asp" 'Missing data in the URL redirect them...
  Response.End
End If
```

If all the required information is present execution continues, otherwise the user is redirected to the home page of the web application, *index.asp*. Next, the code verifies that the data is authentic by using the checksum sent in the query string. If the checksum is invalid the user is given a message that the URL that they clicked on is invalid; otherwise execution continues. Then the page checks to make sure that the user is not

already active, if so they are given a message stating that their account is already active and processing is terminated. Finally, if the user is not active and the checksum is valid the users account is either activated or deleted depending on which link the user clicked on.

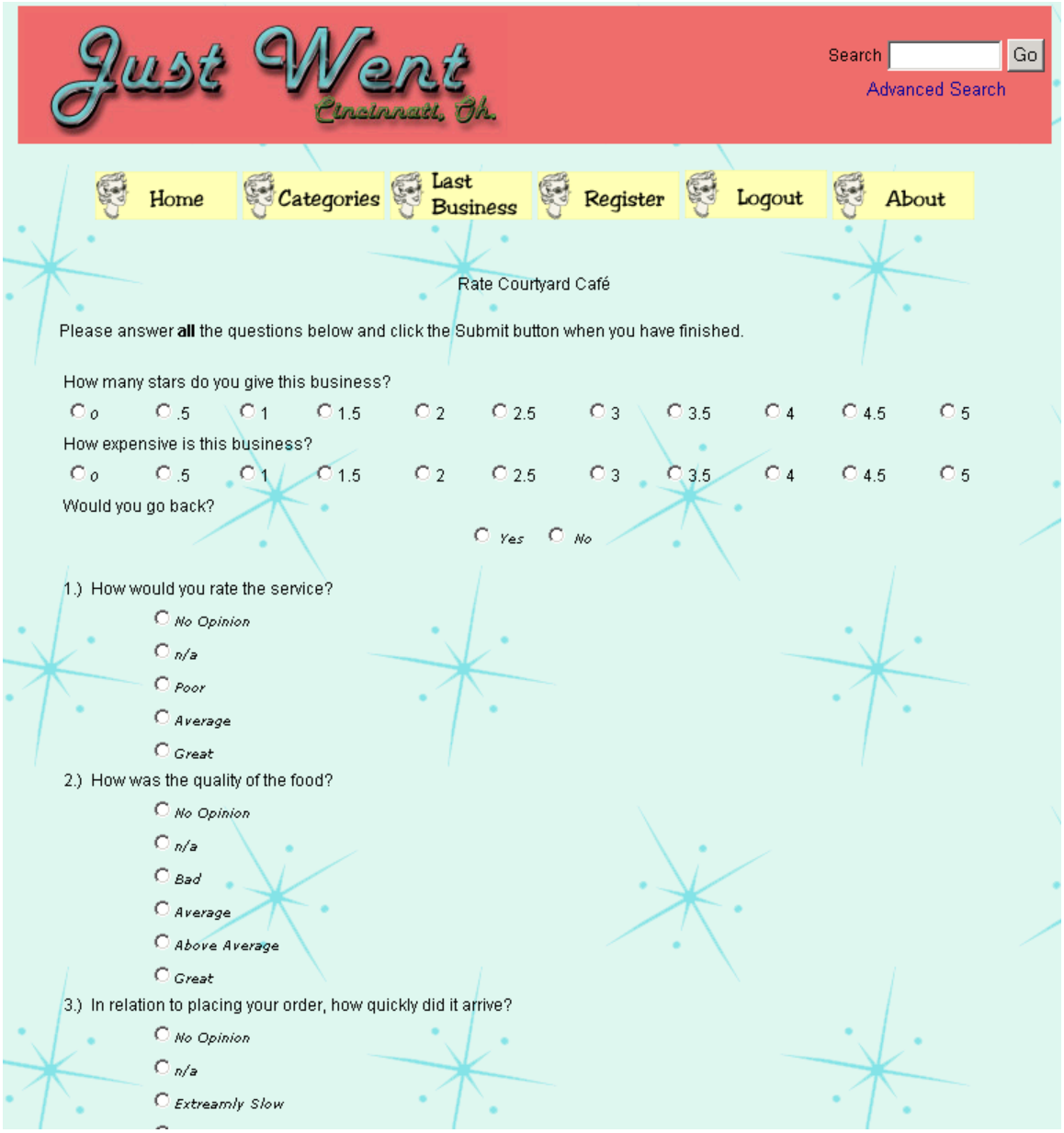


Figure 15. Page 12: Ratebiz.asp

This page is loaded when the user selects the "Rate This Business" link from the business information page, *business.asp*. Upon entering it checks to see if the user has logged in yet; if not the user is redirected to a login page, *login.asp*. Otherwise, the page then instantiates the ActiveX component, JustWent.DataForm, for use in the rest of the processing of the page. Next, the components method

hasUserRatedBusinessBefore is called. If the value returned is *true* then the user is redirected to the re-rating page, *rerate.asp*. Next, the components method *isValidBusiness* is called to determine if the business being rated is valid. If the business is valid, processing continues and the rating form is displayed by calling the components *ShowRatingForm()* method. Upon completion of the page, the object is destroyed.

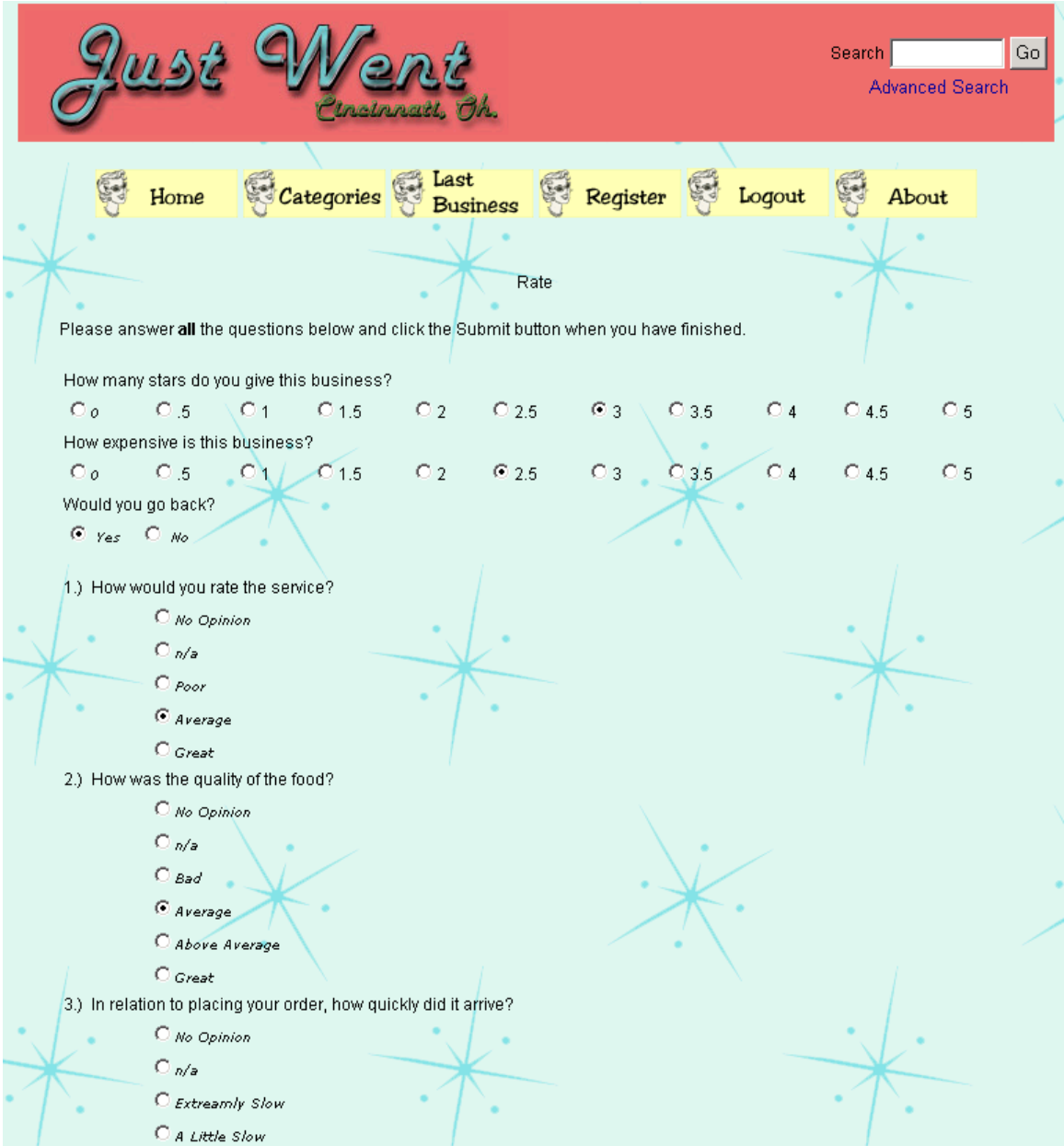


Figure 16. Page 13: Reratebiz.asp

This page is executed when the user has chosen to rate a business and the rating entry page, *ratebiz.asp*, has determine that the user has already rated this business. Upon entering the page the ActiveX component, JustWent.DataForm, is instantiated. The page then calls the components *isValidBusiness()* method to determine if the business to be re-rated is valid. Next the components *HasUserRatedBusinessBefore()* method is called to

determine if the user should be accessing this page. If the method returns *true* then processing on this page continues; if *false*, the user is redirected back to the business-rating page, *ratebiz.asp*. Finally, the component's method *showReRateForm()* is called to display the rating form with all the user's previous answers; after this method call the object is destroyed.

Figure 17. Page 14: Vratebiz.asp

This page is executed when the user submits their rating from the rating or re-rating forms. It accepts the data from these pages in a *post* request method. Upon entering the ActiveX component, JustWent.DataForm, is instantiated and it's user id, business id and database connect string properties are all set. The page then calls the component's *isValidBusiness()* method and the return value is stored. If the method returns *false* then an error is displayed to the user, otherwise processing continues. Next,

the components properties *sValidationFailPage* and *sValidationSuccessPage* are set and the *displayValidationForm()* method is called to display the validation page to the user. After the call the component is destroyed.

Figure 18. Page 15: Enterrate.asp

This page is executed when the user has successfully passed the rating verification page, *vratebiz.asp*. Upon entering the page, the code checks to see if the user is logged in, if not the user is directed to the login page.[16] If the user is logged on, then the ActiveX component JustWent.DataForm is created and the required properties are set. These properties are the databases connect string, the user id, business id and the JavaScript pop-up property is set to *true*.

The page then calls the component's *insertUserRatingIntoDatabase()* method to store the information into the database. The method returns a Boolean value of *true* if the entry was successful or *false* if the entry failed[17]. If the method failed, then the page will set the components *sValidationFailPage* and *sValidationSuccessPage* properties and call

[16] If the user makes it to this page, they should already be logged in.
[17] This usually occurs when all the questions do not have a valid user response.

the *displayValidationForm()* method; redisplaying the validation form with all of their answers already entered and questions that are missing a response noted. Otherwise, the user will be given a congratulations message and a link back to that business so they can immediately see how their opinion has affected the business's rating. Finally, the page will destroy the JustWent.DataForm object.

Figure 19. Page 16: Login.asp

This page allows the user to login to the web application. When the page is executed it checks to make sure that the user is not already logged in. If they are, it displays an error message stating that they have already logged in and gives them a link to logout.

If the user is not logged in, they are presented with a form asking for their username and password. Once the user enters their credentials and submits the form, the data will be sent to this page where the following code will execute:

```
      Set oConn = Server.CreateObject("ADODB.Connection")
      Set oUserRS = Server.CreateObject("ADODB.Recordset")
      oConn.Open Session("s_dsn")
      sSQL = "SELECT urid, firstname, lastname, email, password FROM tblUsers
where userid='" & sUserID & "'"
      oUserRS.Open sSQL, oConn, 3, 1

      If Not oUserRS.EOF Then
        sRSPass = oUserRS("password") 'Grab the real password from the
database.
        'Test to see if the passwords match.
        If sRSPass = sPass Then
          bIsValid = True 'Set the bIsValid to TRUE
          'Now lets set some session variables...
          Session("b_loggedin") = True
          Session("i_usernumber") = oUserRS("urid")
          Session("s_userid") = sUserID
          Session("s_firstname") = oUserRS("firstname")
          Session("s_lastname") = oUserRS("lastname")
          Session("s_email") = oUserRS("email")
          Session("s_currentip") = Request.ServerVariables("IP")
          Session("d_loginDate") = Now()
          'Now give them a message that they are logged in...
          If Session("s_lastScript")<>"" Then
            Response.Redirect Session("s_lastScript")
            Session("s_lastScript") = "" 'Reset this variable to nothing
         End If
          Response.Write "Welcome " & Session("s_userid") & "(" &
Session("s_firstname") & ")!!!!"
          Response.Write "You are now logged into JustWent.Com...  Take a look
around and let us know "
          Response.Write "what you have to say."

        Else
          bIsValid = False
          Response.Write "The password you have entered is incorrect...<br>"
          Response.Write "Please go <a href=""javascript:back()"">back</a> and
try again.<br><br>"
          Response.Write "Forgot your password click <a
href=""userpass.asp"">here</a> and we will send it "
          Response.Write "to you.<br>"
          Response.Flush
        End If
```

The code will open a connection to the database and query it for the user's credentials based on the username. If the recordset returned is empty, the user is given a message that the username could not be found and provided with a link to go back and try again. If the recordset has records, but the password entered does not match the password in the database, the user is given a message that the password is incorrect. Finally, if the recordset contains records and the password entered matches the password in the database then the user is logged onto the system. (Figure 20.)

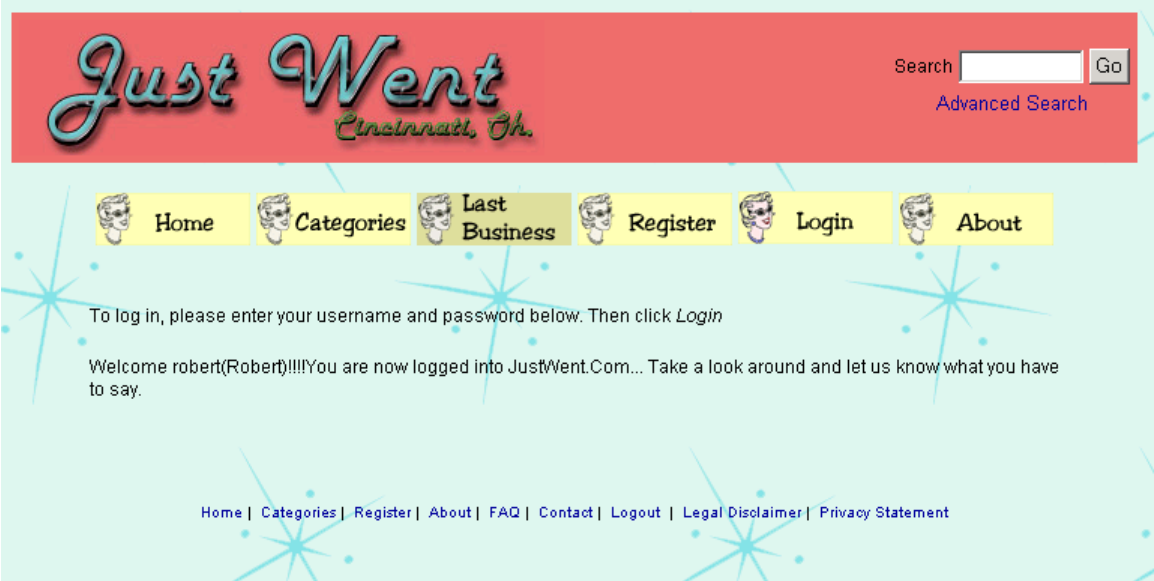


Figure 20. Successful Logon Screen.

Session variables will then be populated to hold information about the user until the session times out or is abandoned when the user logs out.

Figure 21. Page 17: Logoff.asp

This page allows the user to logoff of the system. When the page first executes, it presents the user with a logout confirmation page; the user must click the confirm button to logout of the site. When the user clicks the confirm button the page reloads itself and executes the following code:

```
If Request.Form.Count > 0 Then
  If LCase(Request.Form("confirm") & "") = "confirm" Then
    bConfirmed = True
    'Wipe all session info here
     Session.Abandon
  End If
End If
```

The code checks to see if form data has been posted to this page using *Request.Form.Count*. If the count returned is greater than zero, the code checks to see if the value *confirm* is included in the form data. If the value confirm is found in the form data, the session is abandoned using *Session.Abandon*, logging out the user.

Figure 22. Page 18: About.asp

This page gives the user information about the web community such as the mission statement, what it's purpose it, who can use it, who it is for and how it works.

7. Conclusion and Recommendations

I have chosen this project because of a concept I had several years ago while having Saturday brunch in Kaldis' Coffee House; this Senior Design project has enabled me to fulfill this idea. When I began this project, I chose Microsoft as the platform for development using ASP as the scripting language and SQL Server as the backend database. I would have liked to develop the application using ASP.Net and SQL Server 2000. This would have allowed me to gain experience using the next generation scripting language by Microsoft. In addition using SQL Server 2000 would have prepared me for the changes that come from moving to SQL Server 7 to SQL Server 2000

If I would have to choose another platform I would have liked to write the application in PHP, *Personal Home Page,* as the scripting language. For the backend database I would have chosen MySQL; and ran the application using the Apache Web Server on a Solaris box. This would have allowed me to separate my code from some of the proprietary coding techniques such as the ADO Data Shaping, allowing my code to be portable to more platforms. At the same time this would have alleviated me from some of the costs of the licenses required for Windows NT/2000 and SQL Server.

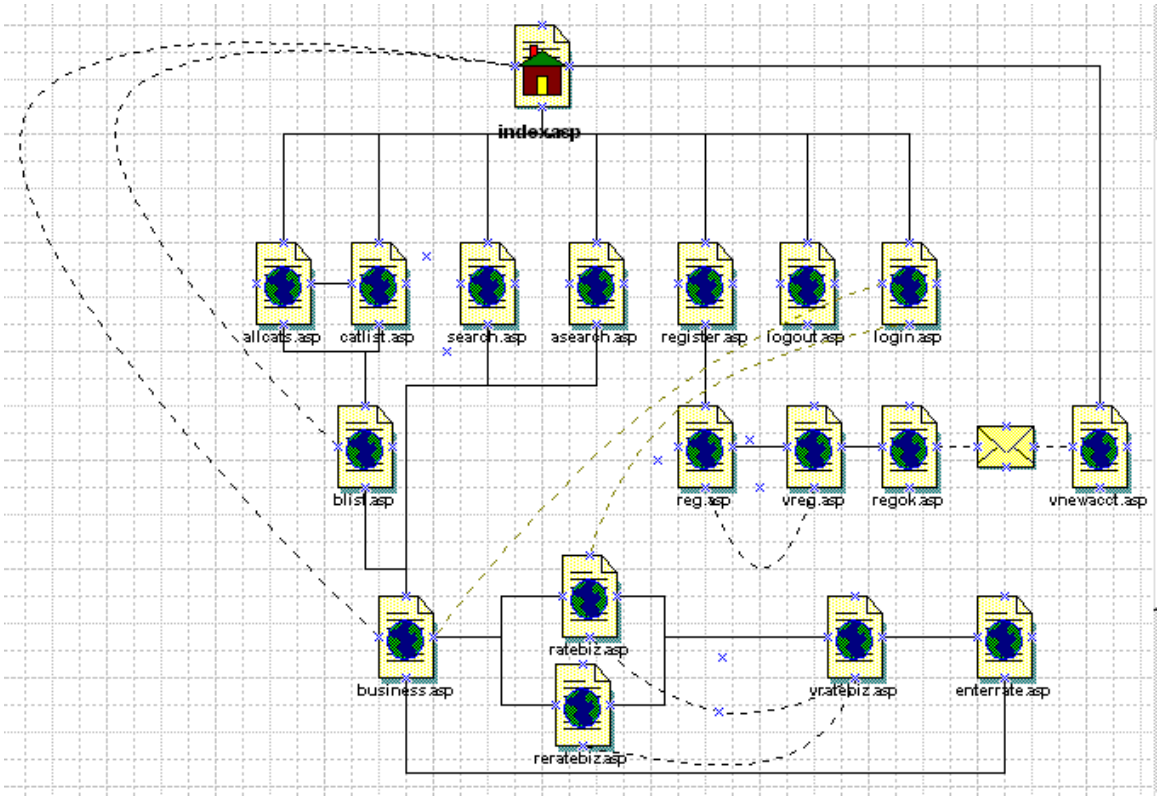
After reviewing the timeline, a considerable amount of time should have been given to the debugging and testing of the applications ActiveX component. Early testing brought up several problems such as the lack of a testing environment; Internet Information Services was not installed locally for testing purposes. In addition, I had run into a few minor errors that had taken more time than I had scheduled for; all have since been fixed.

The design specifications for this project have been completed in full. The application is a fully functional business rating web community. Its framework has been developed to allow for scalability and extensibility; this has already been demonstrated in some of this documentation.[18]

This web application and its backend database are proof of my knowledge gained by the Information Engineering Technology program and demonstrate my competence as Information Engineering professional.

[18] The *Last 10 Rated* listing on index.asp is an extended feature.

Appendix A.
Site Diagram of JustWent.Com



Appendix B.
SQL Code To Create Tables

```
CREATE TABLE [dbo].[tblBusiness] (
	[busid] [int] IDENTITY (1, 1) NOT NULL ,
	[subcategoryid] [int] NULL ,
	[name] [varchar] (50) NULL ,
	[addr01] [varchar] (50) NULL ,
	[addr02] [varchar] (50) NULL ,
	[city] [varchar] (50) NULL ,
	[state] [varchar] (50) NULL ,
	[zip] [varchar] (50) NULL ,
	[phone] [varchar] (50) NULL ,
	[web] [varchar] (50) NULL ,
	[email] [varchar] (50) NULL ,
	[pimage] [varchar] (50) NULL ,
	[review] [text] NULL ,
	[description] [text] NULL ,
	[thumbs] [bit] NOT NULL ,
	[stars] [bit] NOT NULL ,
	[coins] [bit] NOT NULL ,
	[ischild] [bit] NOT NULL ,
	[pareintid] [int] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblBusUsrRatingDetails] (
	[busurid] [int] IDENTITY (1, 1) NOT NULL ,
	[busid] [int] NOT NULL ,
	[urid] [int] NOT NULL ,
	[timestamp] [datetime] NOT NULL ,
	[ip] [varchar] (50) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblCategories] (
	[categoryid] [int] IDENTITY (1, 1) NOT NULL ,
	[name] [varchar] (50) NOT NULL ,
	[orderno] [int] NULL ,
	[description] [varchar] (50) NULL ,
	[fpage] [bit] NOT NULL ,
	[hassubcats] [bit] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblCoins] (
	[coinid] [int] IDENTITY (1, 1) NOT NULL ,
	[busurid] [int] NULL ,
	[urid] [int] NOT NULL ,
	[busid] [int] NOT NULL ,
	[coins] [real] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblGenders] (
	[genderid] [int] IDENTITY (1, 1) NOT NULL ,
	[text] [varchar] (50) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblQuestions] (
```

```
	[questionid] [int] IDENTITY (1, 1) NOT NULL ,
	[subcategoryid] [int] NOT NULL ,
	[text] [varchar] (255) NULL ,
	[typeid] [int] NOT NULL ,
	[displocation] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblQuestionTypes] (
	[typeid] [int] IDENTITY (1, 1) NOT NULL ,
	[typename] [varchar] (50) NULL ,
	[html] [varchar] (50) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblResponses] (
	[responseid] [int] IDENTITY (1, 1) NOT NULL ,
	[questionid] [int] NOT NULL ,
	[text] [varchar] (50) NULL ,
	[value] [varchar] (50) NULL ,
	[location] [int] NULL ,
	[default] [bit] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblStars] (
	[starid] [int] IDENTITY (1, 1) NOT NULL ,
	[busurid] [int] NULL ,
	[urid] [int] NOT NULL ,
	[busid] [int] NOT NULL ,
	[number] [real] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblSubCategories] (
	[subcategoryid] [int] IDENTITY (1, 1) NOT NULL ,
	[categoryid] [int] NULL ,
	[name] [varchar] (50) NOT NULL ,
	[orderno] [int] NULL ,
	[description] [varchar] (50) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblTextDescription] (
	[textdescid] [int] IDENTITY (1, 1) NOT NULL ,
	[busurid] [int] NULL ,
	[urid] [int] NOT NULL ,
	[busid] [int] NOT NULL ,
	[questionid] [int] NOT NULL ,
	[text] [text] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblThumbs] (
	[thumbid] [int] IDENTITY (1, 1) NOT NULL ,
	[busurid] [int] NULL ,
	[urid] [int] NOT NULL ,
	[busid] [int] NOT NULL ,
	[updown] [int] NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblUserResponse] (
```

```
	[userresponseid] [int] IDENTITY (1, 1) NOT NULL ,
	[busrid] [int] NOT NULL ,
	[urid] [int] NOT NULL ,
	[busid] [int] NOT NULL ,
	[questionid] [int] NULL ,
	[responseid] [int] NULL ,
	[text] [varchar] (50) NULL ,
	[value] [varchar] (50) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[tblUsers] (
	[urid] [int] IDENTITY (1, 1) NOT NULL ,
	[userid] [varchar] (50) NOT NULL ,
	[password] [varchar] (50) NOT NULL ,
	[email] [varchar] (50) NOT NULL ,
	[newaccount] [bit] NOT NULL ,
	[zipcode] [varchar] (50) NULL ,
	[genderid] [int] NULL ,
	[firstname] [varchar] (50) NULL ,
	[lastname] [varchar] (50) NULL ,
	[agegroup] [int] NULL ,
	[age] [int] NULL ,
	[applydate] [datetime] NULL ,
	[isadmin] [bit] NOT NULL
) ON [PRIMARY]
GO
```

Appendix C.
SQL Code To Create Stored Procedures

```
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_AddUser(@userid varchar(50), @password varchar(50), @email
varchar(150), @firstname varchar(50),
                                @lastname varchar(50), @age int, @agegroup int,
@genderid int, @zipcode varchar(50))
AS

DECLARE @newid int

INSERT INTO tblUsers(userid, password, email, zipcode, genderid, firstname,
lastname, agegroup, age)
                        VALUES(@userid, @password, @email, @zipcode, @genderid,
@firstname, @lastname, @agegroup, @age)

SELECT @newid = @@IDENTITY

SELECT urid, applydate FROM tblUsers WHERE urid = @newid


GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_BusinessQuestionsBySubCategory(@subcatid int)
AS
SELECT tblQuestions.questionid, tblQuestions.text, tblQuestions.typeid,
tblQuestionTypes.typename, tblQuestionTypes.html
 FROM tblQuestions
  INNER JOIN tblQuestionTypes ON tblQuestionTypes.typeid=tblQuestions.typeid
WHERE subcategoryid=@subcatid


GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_CountRatingsForBusiness (@busid int)
AS
SELECT COUNT(busurid) AS ratingtotal FROM tblBusUsrRatingDetails WHERE
busid=@busid

GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_DeleteStarCoinThumb(@bususerid int)
AS
```

```
/*
This is used in enterrate.asp
It will delete all the coin, stars and thumb ratings for a particular business-
to-user-id.
*/
DELETE tblCoins WHERE busurid=@bususerid

DELETE tblStars WHERE busurid=@bususerid

DELETE tblThumbs WHERE busurid=@bususerid

GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_DeleteTextDescription(@bususerid int)
 AS
/*
This is used in enterrate.asp
This will delete all the records in tblTextDescription for a specific business-
to-user-response id.
*/

DELETE tblTextDescription where busurid=@bususerid

GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_DeleteUserResponses(@bususerid int)
AS
/*
This is used in enterrate.asp.
It will delete all the users' ratings for a particular business-to-user-rating
id.
It will be used inside of a transaction statement so that that the deletes will
not occur until
the transaction has fully completed.
A fully completed transaction will be where they have answered all the
questions and all the sp_Insert
stored procedures have executed.
*/
DELETE tblUserResponse WHERE busrid=@bususerid

GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_GetAllCategories
AS
/*This was a previous version it only listed just the categories...*/
/*
SELECT categoryid, name, orderno, hassubcats FROM tblCategories ORDER BY name
*/
```

```
/*This will list each category and give a total of how many subcategories are
related to it*/
SELECT categoryid, name, orderno, hassubcats,
(SELECT count(subcategoryid) FROM tblSubCategories
  WHERE tblSubCategories.categoryid=tblCategories.categoryid) AS total
FROM tblCategories ORDER BY name



GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_GetAllPossibleResponses
AS
/*This is used in conjunction with sp_BusinessQuestionsBySubCategory in
ratebiz.asp*/
SELECT responseid, questionid, [text], value, location, [default] FROM
tblResponses ORDER BY questionid, location



GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_GetAllPossibleResponsesByQuestionID (@qid int)
AS
/*This is used in conjunction with sp_BusinessQuestionsBySubCategory in
enterrate.asp
It may also be used in ratebiz.asp and reratebiz.asp because this is more
efficient
when used in the shape statement.*/
SELECT responseid, questionid, [text], value, location, [default] FROM
tblResponses WHERE questionid=@qid ORDER BY questionid, location




GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_GetAllSubCategories
AS
SELECT categoryid, subcategoryid, name, orderno,
(SELECT count(busid) FROM tblBusiness
  WHERE tblBusiness.subcategoryid=tblSubCategories.subcategoryid) AS total
FROM tblSubCategories ORDER BY name

GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO
```

```
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

/*Used In:
-blist.asp
*/
CREATE PROCEDURE sp_GetBusiness(@subcategoryid int)
AS
SELECT busid, name FROM tblBusiness WHERE subcategoryid=@subcategoryid ORDER BY
name


GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

/*
Description:
 This will get all the business info for a particular business and all the
counts for the thumbs, stars and coins
Used In:
 business.asp
*/
CREATE PROCEDURE sp_GetGeneralBusinessInfo (@busid int)
AS
/*
This was an older version, it did not pull the categories or the sub categories
that I needed.
SELECT name, addr01, addr02, city, state, zip, phone, web, email, pimage,
review, description, thumbs, stars, coins,
 (SELECT count(thumbid) FROM tblThumbs WHERE busid=@busid and updown=0) AS
thumbdowncount,
 (SELECT count(thumbid) FROM tblThumbs WHERE busid=@busid and updown=1) AS
thumbupcount,
 (SELECT count(starid) FROM tblStars WHERE busid=@busid) AS starscount,
 (SELECT count(coinid) FROM tblCoins WHERE busid=@busid) AS coincount FROM
tblBusiness WHERE busid=@busid
*/
SELECT tblBusiness.name, tblBusiness.addr01, tblBusiness.addr02,
tblBusiness.city,
       tblBusiness.state, tblBusiness.zip, tblBusiness.phone, tblBusiness.web,
tblBusiness.email,
       tblBusiness.pimage, tblBusiness.review, tblBusiness.description,
       tblBusiness.thumbs, tblBusiness.stars, tblBusiness.coins,
       tblCategories.name AS catname,  tblSubCategories.subcategoryid AS
subcatid, tblSubCategories.name AS subcatname,
 (SELECT count(thumbid) FROM tblThumbs WHERE busid=@busid and updown=0) AS
thumbdowncount,
 (SELECT count(thumbid) FROM tblThumbs WHERE busid=@busid and updown=1) AS
thumbupcount,
 (SELECT count(starid) FROM tblStars WHERE busid=@busid) AS starscount,
 (SELECT sum(number) FROM tblStars WHERE busid=@busid) AS starsum,
 (SELECT count(coinid) FROM tblCoins WHERE busid=@busid) AS coincount,
 (SELECT sum(coins) FROM tblCoins WHERE busid=@busid) AS coinsum
FROM (tblCategories INNER JOIN tblSubCategories ON tblCategories.categoryid =
tblSubCategories.categoryid)
 INNER JOIN tblBusiness ON tblSubCategories.subcategoryid =
tblBusiness.subcategoryid
WHERE busid=@busid
```

```
GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_GetQuestionResponsesAndTotals (@busid int)
/*This is used in conjunction with sp_BusinessQuestionsBySubCategory in
business.asp*/
AS
/*
SELECT questionid, responseid, text,
   (SELECT count(userresponseid) FROM tblUserResponse WHERE
tblUserResponse.questionid=tblResponses.questionid
          AND busid=@busid) AS responsetotal
FROM tblResponses
*/


--Modification on 02/04/2001 @ 1:24am
SELECT questionid, responseid, text,
   (SELECT count(userresponseid) FROM tblUserResponse WHERE
tblUserResponse.responseid=tblResponses.responseid
          AND busid=@busid) AS responsetotal
FROM tblResponses
GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_GetQuestionsByBusinessID(@busid int)
 AS
/*
This will be used in enterrate.asp
It will take a business id and get the subcategoryid and then return a
recordset
with all the questions for that particular subcategory id.
*/
DECLARE @subcatid int

SELECT @subcatid = subcategoryid FROM tblBusiness WHERE busid = @busid

IF EXISTS(SELECT @subcatid)
BEGIN
  SELECT questionid, typeid FROM tblQuestions WHERE subcategoryid=@subcatid
ORDER BY displocation
END

ELSE
BEGIN
 SELECT questionid, typeid FROM tblQuestions WHERE subcategoryid=-1
END


GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO
```

```
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_GetSubCategories(@categoryid int)
AS
SELECT subcategoryid, name, orderno, (SELECT count(busid) FROM tblBusiness
  WHERE tblBusiness.subcategoryid=tblSubCategories.subcategoryid) AS total
FROM tblSubCategories where categoryid=@categoryid ORDER BY name




GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_GetUserResponseCoinStarThumb(@busurid int, @coins real
OUTPUT, @stars real OUTPUT, @thumbs int OUTPUT)
AS

/*
This procedure will get the entry for:
-Number of coins
-Number of stars
-If they would go back
for a given Business-To-User-Rating ID...

NOTE:  The Business-To-User-Rating ID is unique for a specific individual
rating a specific business...
This will be used in the ReRate.asp page

*/

/*SELECT tblBusUsrRatingDetails.busurid, tblCoins.coins, tblStars.number,
tblThumbs.updown
  FROM
    (
        (tblBusUsrRatingDetails LEFT JOIN tblCoins ON
tblBusUsrRatingDetails.busurid = tblCoins.busurid)
        LEFT JOIN tblStars ON tblBusUsrRatingDetails.busurid =
tblStars.busurid)
        LEFT JOIN tblThumbs ON tblBusUsrRatingDetails.busurid =
tblThumbs.busurid

        WHERE (((tblBusUsrRatingDetails.busurid)=@busurid))
*/
SET ROWCOUNT 1


SELECT @coins=tblCoins.coins, @stars=tblStars.number, @thumbs=tblThumbs.updown
  FROM
    (
        (tblBusUsrRatingDetails LEFT JOIN tblCoins ON
tblBusUsrRatingDetails.busurid = tblCoins.busurid)
        LEFT JOIN tblStars ON tblBusUsrRatingDetails.busurid =
tblStars.busurid)
        LEFT JOIN tblThumbs ON tblBusUsrRatingDetails.busurid =
tblThumbs.busurid

        WHERE (((tblBusUsrRatingDetails.busurid)=@busurid))
```

```
SET ROWCOUNT 0



GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_GetUserResponsesBasedOnBUSURID(@busurid int)
AS

/*
This will return all the User Responses
from the table >>tblUserResponse<< for a particular
Business-User-Rating ID

NOTE: The fields are ALIASED because this Stored Procedure
      is being used in a Shape Command.  The parser MAY get
      confused if more than one field has the same name.
*/
       SELECT questionid AS tURquestionid, responseid AS tURresponseid,
                text AS tURtext, value AS tURvalue
              FROM tblUserResponse WHERE busrid=@busurid


GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_GetUserTextResponsesBasedOnBUSURID(@busurid int)
AS
/*
This will return all the User Responses
from the table >>tblTextDescription<< for a particular
Business-User-Rating ID

NOTE: The fields are ALIASED because this Stored Procedure
      is being used in a Shape Command.  The parser MAY get
      confused if more than one field has the same name.

*/
       SELECT questionid AS tTDquestionid, text AS tTDtext
              FROM tblTextDescription WHERE busurid=@busurid


GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO
```

```
CREATE PROCEDURE sp_GetUserTimeStamp(@urid int) AS
/*
This will return the date/time stamp for the user for use in RegOK.asp
*/
SELECT applydate FROM tblUsers WHERE urid=@urid


GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_HasUserRatedBusiness(@userid int, @busid int, @answer int
OUTPUT)  AS
/* This procedure will return the number of times that a user has rated a
particular business
    the result should be only 0 or 1.
 */

SELECT @answer = count(busurid) FROM tblBusUsrRatingDetails
       WHERE busid = @busid AND urid=@userid

RETURN @answer


GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_HasUserRatedBusinessWithID(@userid int, @busid int, @answer
int OUTPUT, @bususrid int OUTPUT)  AS
/* This procedure will return the number of times that a user has rated a
particular business
    the result should be only 0 or 1.
 */



SELECT @answer = count(busurid) FROM tblBusUsrRatingDetails
       WHERE busid = @busid AND urid=@userid

IF @answer = 0
  BEGIN
      SET @bususrid = 0
  END

ELSE
  BEGIN
      SET ROWCOUNT 1 /*Get only one row from this query*/
      SELECT @bususrid = busurid FROM tblBusUsrRatingDetails
         WHERE busid = @busid AND urid=@userid
      SET ROWCOUNT 0 /*Now lets let SQL return all rows*/

  END

RETURN @answer
```

```
GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_HasUserRatedBusinessWithID_CreateNoExist(@userid int,
@busid int, @ip varchar(50), @answer int OUTPUT, @bususrid int OUTPUT)
AS
/* sp_HasUserRatedBusinessWithID_CreateNoExist

This procedure will do the following:
       Check to see how many ratings the user has in the system for a particular
business
         it should be either 0 or 1.  This is stored in @answer

       If the answer is =0 it will enter the userid and business id into
tblBusUsrRatingDetails table
         and return the new bususerid to the calling code.

USAGE: enterrate.asp
NOTE:  This is used in a transaction statement so if the transaction is rolled
back then this entry will not occur.
 */



SELECT @answer = count(busurid) FROM tblBusUsrRatingDetails
       WHERE busid = @busid AND urid=@userid

IF @answer = 0
--Insert the userID and businessID and get the BusUserID back
  BEGIN
      INSERT INTO tblBusUsrRatingDetails(busid, urid, ip)
              VALUES(@busid, @userid, @ip)

      SELECT @bususrid = @@IDENTITY
  END

ELSE
  BEGIN
      SET ROWCOUNT 1 /*Get only one row from this query*/
      SELECT @bususrid = busurid FROM tblBusUsrRatingDetails
         WHERE busid = @busid AND urid=@userid
      SET ROWCOUNT 0 /*Now lets let SQL return all rows*/

  END

RETURN @answer




GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO
```

```
CREATE PROCEDURE sp_InsertNewRatingGetID(@busid int, @userid int, @ip
varchar(50), @bususerid int OUTPUT)
 AS
/*
This is used in enterrate.asp to enter the user id, business id, and IP address
and it
returns a business-to-user-rating id.

02/04/2001
No longer used on enterrate.asp...
REASON: This has been replaced by sp_HasUserRatedBusinessWithID_CreateNoExist.

*/

INSERT INTO tblBusUsrRatingDetails(busid, urid, ip)
  VALUES(@busid, @userid, @ip)

SELECT @bususerid = @@IDENTITY

RETURN @bususerid


GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_InsertStarCoinThumb(@stars real, @coins real, @thumbs int,
@bususerid int, @userid int, @busid int)

 AS
/*
This procedure is used in enterrate.asp to enter the rating for the business'
coins stars and thumbs.
*/
IF EXISTS(SELECT @stars)
BEGIN
  INSERT INTO tblStars( busurid, urid, busid, number)
    VALUES(@bususerid, @userid, @busid, @stars)
END

IF EXISTS(SELECT @coins)
BEGIN
  INSERT INTO tblCoins(busurid, urid, busid, coins)
    VALUES(@bususerid, @userid, @busid, @coins)
END

IF EXISTS(SELECT @thumbs)
BEGIN
  INSERT INTO tblThumbs(busurid, urid, busid, updown)
    VALUES(@bususerid, @userid, @busid, @thumbs)
END


GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO
```

```
CREATE PROCEDURE sp_InsertTextDescription(@bususerid int, @userid int, @busid
int, @questionid int, @text text)
AS
/*
This procedure is used in enterrate.asp to put Long Text answers in
tblTextDescription table.
*/
INSERT INTO tblTextDescription(busurid, urid, busid, questionid, text)
  VALUES(@bususerid, @userid, @busid, @questionid, @text)

GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_InsertUserResponses(@bususerid int, @userid int, @busid
int, @questionid int,
                                        @responseid int, @text varchar(255),
@value varchar(255))
 AS
/*
This procedure is used in enterrate.asp to insert the
users answers to questions into tblUserResponse
*/
INSERT INTO tblUserResponse(busrid, urid, busid, questionid, responseid, text,
value)
  VALUES(@bususerid, @userid, @busid, @questionid, @responseid, @text, @value)

GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE  sp_ValidateBusinessWithExtraInfo(@busid int, @isvalid int
OUTPUT, @subcatid int OUTPUT, @busname varchar(50) OUTPUT)
AS
/*
This stored procedure will query the database for the business id and check to
see if there is a valid entry
If there is a valid entry it will return:
       isvalid=1
       subcatid = the subcategory ID for that business entry
If there is not a valid entry it will return
       isvalid=0
       subcatid = 0

This stored procedure is being used in ratebiz.asp and reratebiz.asps
*/
IF EXISTS(SELECT subcategoryid FROM tblBusiness WHERE busid=@busid)

  BEGIN
      SET @isvalid = 1
      SELECT @subcatid = subcategoryid, @busname=name FROM tblBusiness WHERE
busid=@busid
  END

ELSE

  BEGIN
```

```
      SET @isvalid=0
      SET @subcatid=0
      SET @busname=''
  END

RETURN @isvalid



GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

CREATE PROCEDURE sp_ValidateBusinessWithInfo(@busid int, @isvalid int OUTPUT,
@subcatid int OUTPUT, @busname varchar(50) OUTPUT)
AS
/*
This stored procedure will query the database for the business id and check to
see if there is a valid entry
If there is a valid entry it will return:
       isvalid=1
       subcatid = the subcategory ID for that business entry
If there is not a valid entry it will return
       isvalid=0
       subcatid = 0

This stored procedure is being used in ratebiz.asp and reratebiz.asps
*/
IF EXISTS(SELECT subcategoryid FROM tblBusiness WHERE busid=@busid)

  BEGIN
      SET @isvalid = 1
      SELECT @subcatid = subcategoryid, @busname=name  FROM tblBusiness WHERE
busid=@busid
  END

ELSE

  BEGIN
      SET @isvalid=0
      SET @subcatid=0
      SET @busname=''
  END

RETURN @isvalid


GO
SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO
```

Appendix D.
ActiveX Component Code

```
'/*************************************************************************
'/JustWent.DataForm
'/
'/Author: Robert Brown
'/Create Date:2/1/2001
'/Last Modify:2/5/2001, 02/07/2001
'/Description: This class is to be used in the JustWent ActiveX dll.
'/             It will encapsulate most of the business logic and displaying
'/             of pages to the user.
'/
'/Notes:
'/             It currently uses the OnStartPage and OnEndPage
'/             but it must be modified to use the GetObjectContext() object.
'/             so that it can be invloved in page level transactions.
'/
'/             Use strings to hold the output Response.Write may have
additional
'/             overhead...
'/*************************************************************************
Option Explicit 'DECLARE EVERYTHING


Private m_sErrorDescription As String   'sErrDesc -> Error Description
Private m_lErrorCode As Long 'lErrCode -> Error code

Private m_iDatabaseConnectTimeout As Integer    'iDBConnectTimeout ->
Connection Timeout to Database
Private m_sDatabaseConnectString    'sDBConnectString -> Database connect
string
Private m_iDatabaseQueryTimeout As Integer  'iDBQueryTimeout -> Timeout length

Private m_bDebug As Boolean 'bDebugInfo -> Show Debug Info on browser?
Private m_bJavaScriptDisplay As Boolean 'bJavaScriptPopUp -> Display the
JavaScriptPopUpWindow?

Private m_lBusID As Long    'lBusinessID -> Business ID
Private m_lUserID As Long   'lUserID -> Users ID
Private m_lSubCatID As Long 'lSubCategoryID -> SubCategory ID
Private m_lBusToUserID As Long  'lBusinessUserID -> BusinessToUserID

'Properities used for specific subroutines...

'Used in displayValidationForm()
Private m_sValidationSuccessPage As String  'sValidationSuccessPage
'Used in displayValidationForm() Page to goto if the form is filled out.
Private m_sValidationFailPage As String 'sValidationFailPage
'Used in displayValidationForm Page to goto if the form is not filled out.

'Used in isValidBusiness() As Boolean
Private m_iIsValidBusiness As Integer   'iIsValidBusiness
'Used to return if the business is a valid business or not.
Private m_sBusinessName As String 'sBusinessName
'Used to hold the business name.


'These are internal variables used to access the ASP intrinsics.
Private m_oScriptContext As ScriptingContext
Private m_oResponse As Response
```

```
Private m_oRequest As Request
Private m_oServer As Server
Private m_oSession As Session
Private m_oApplication As Application

'/*************************************************************************
'/P R O P E R T Y   G E T   &   S E T
'/
'/B E G I N
'/*************************************************************************




'/*************************************************************************
'/P U B L I C   S U B R O U T I N E   S E C T I O N S
'/
'/B E G I N
'/*************************************************************************


Public Sub OnStartPage(PassedScriptingContext As ScriptingContext)
'Called when the component is created in ASP

    'Get a handle on these objects so I can read/write from client and server.
    Set m_oScriptContext = PassedScriptingContext
    Set m_oResponse = PassedScriptingContext.Response
    Set m_oRequest = PassedScriptingContext.Request
    Set m_oServer = PassedScriptingContext.Server
    Set m_oSession = PassedScriptingContext.Session
    Set m_oApplication = PassedScriptingContext.Application

    'Set up some default values.
    m_sErrorDescription = ""
    m_lErrorCode = 0
    m_iDatabaseConnectTimeout = 180
    m_iDatabaseQueryTimeout = 180

    m_sDatabaseConnectString = ""


    m_bDebug = False


End Sub

Public Sub OnEndPage()
'Called when the component is destroyed in ASP

    'Clean up memory...
    Set m_oApplication = Nothing
    Set m_oSession = Nothing
    Set m_oServer = Nothing
    Set m_oRequest = Nothing
    Set m_oResponse = Nothing
    Set m_oScriptContext = Nothing

End Sub

Public Sub ShowRatingForm()

Dim oShapeConn As ADODB.Connection
Dim oQuestionRS As ADODB.Recordset
```

```
Dim oQuesRespRS As ADODB.Recordset

Dim sSQL As String
Dim intCounter As Integer, iTypeID As Integer, lQuestionID As Long
Dim bSelectFirstRun As Boolean
Dim iReturn As Integer

'This subroutine will display all the questions for a particular
'business.

  If m_bJavaScriptDisplay = True Then
    iReturn = OpenProgressWindow("<center>Loading Questions<br>Please
Wait</center>")
  End If

    'This subroutine will display the rating form to the user.

    Set oShapeConn = CreateObject("ADODB.Connection")
    Set oQuestionRS = CreateObject("ADODB.Recordset")
    Set oQuesRespRS = CreateObject("ADODB.Recordset")

    'Generate the SQL statement to get all the questions and responses for this
category.
    sSQL = "SHAPE {{ CALL dbo.sp_BusinessQuestionsBySubCategory(" & m_lSubCatID
& ")}} "
    sSQL = sSQL & "APPEND ({{ CALL dbo.sp_GetAllPossibleResponses}} "
    sSQL = sSQL & "RELATE questionid TO questionid) AS questionresponses"

    oShapeConn.Provider = "MSDataShape"
    oShapeConn.Open m_sDatabaseConnectString
    'Open the recordset with adOpenStatic=3 and adLockReadOnly=1
    oQuestionRS.Open sSQL, oShapeConn, adOpenStatic, adLockReadOnly


    intCounter = 0
    If Not oQuestionRS.EOF Then
            m_oResponse.Write "<form action=""vratebiz.asp"" method=""post"">"
& vbCrLf

            m_oResponse.Write "<table align=""center"" width=""100%""
class=""quest"">"
            m_oResponse.Write "<tr><td class=""questiontext""
colspan=""11"">How many stars do you give this business?</td></tr>" & vbCrLf
            m_oResponse.Write "<tr><td class=""questionres"">"
            m_oResponse.Write "<input type=""radio"" name=""stars""
value=""0"">0  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""stars""
value="".5"">.5  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""stars""
value=""1"">1  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""stars""
value=""1.5"">1.5  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""stars""
value=""2"">2  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""stars""
value=""2.5"">2.5  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""stars""
value=""3"">3  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""stars""
value=""3.5"">3.5  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""stars""
value=""4"">4  </td><td>"
```

```
            m_oResponse.Write "<input type=""radio"" name=""stars""
value=""4.5"">4.5  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""stars""
value=""5"">5  "
            m_oResponse.Write "</td></tr>" & vbCrLf
            m_oResponse.Write "</table>" & vbCrLf

            m_oResponse.Write "<table align=""center"" width=""100%""
class=""quest"">"
            m_oResponse.Write "<tr><td class=""questiontext""
colspan=""11"">How expensive is this business?</td></tr>" & vbCrLf
            m_oResponse.Write "<tr><td class=""questionres"">"
            m_oResponse.Write "<input type=""radio"" name=""coins""
value=""0"">0  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""coins""
value="".5"">.5  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""coins""
value=""1"">1  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""coins""
value=""1.5"">1.5  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""coins""
value=""2"">2  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""coins""
value=""2.5"">2.5  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""coins""
value=""3"">3  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""coins""
value=""3.5"">3.5  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""coins""
value=""4"">4  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""coins""
value=""4.5"">4.5  </td><td>"
            m_oResponse.Write "<input type=""radio"" name=""coins""
value=""5"">5  "
            m_oResponse.Write "</td></tr>" & vbCrLf
            m_oResponse.Write "</table>" & vbCrLf

            m_oResponse.Write "<table align=""center"" width=""100%""
class=""quest"">" & vbCrLf
            m_oResponse.Write "<tr><td class=""questiontext""
colspan=""2"">Would you go back?</td></tr>" & vbCrLf
            m_oResponse.Write "<tr><td class=""questionres"" colspan=""2""
align=""center"">"
            m_oResponse.Write "<input type=""radio"" name=""goback""
value=""1""> Yes   "
            m_oResponse.Write "<input type=""radio"" name=""goback""
value=""0""> No "
            m_oResponse.Write "</td></tr>" & vbCrLf
            m_oResponse.Write "</table>"

            m_oResponse.Write "<table align=""center"" width=""100%""
class=""quest"">" & vbCrLf
        While Not oQuestionRS.EOF
        intCounter = intCounter + 1 'Increment the question counter.
        m_oResponse.Write "<tr><td colspan=""2"" class=""questiontext"">"
        m_oResponse.Write intCounter & ".)  " & oQuestionRS("text") &
"<br>"
        m_oResponse.Write "</td></tr>" & vbCrLf
        iTypeID = oQuestionRS("typeid")
        lQuestionID = oQuestionRS("questionid")

        If iTypeID <> 5 And iTypeID <> 6 Then
        'This means that the question is not a text or textarea question.
```

```
            bSelectFirstRun = True  'This is the begining of a new question set
the flag back to false.
            Set oQuesRespRS = oQuestionRS("questionresponses").Value


            While Not oQuesRespRS.EOF
                If m_bJavaScriptDisplay = True Then
                  UpdateStats "<center>Writing Out Questions</center>",
"Question No: " & intCounter, "", "", "", ""
                End If
                If bSelectFirstRun = True Then
                  m_oResponse.Write "<tr>"
                  m_oResponse.Write "<td>  </td>"
                End If
                Select Case iTypeID
                    Case 1  'unassigned
                        'Don't display any answers at this time 1 is a default.
                    Case 2  'radio button
                        m_oResponse.Write "<td class=""questionres"">"
                        If oQuesRespRS("default") = 1 Then
                            m_oResponse.Write "<INPUT TYPE=""radio"" NAME=" &
Chr(34) & "QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) &
oQuesRespRS("responseid") & Chr(34) & " checked>" & oQuesRespRS("text") &
vbCrLf
                        Else
                            m_oResponse.Write "<INPUT TYPE=""radio"" NAME=" &
Chr(34) & "QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) &
oQuesRespRS("responseid") & Chr(34) & " >" & oQuesRespRS("text") & vbCrLf
                        End If
                        m_oResponse.Write "</td></tr>" & vbCrLf
                    Case 3  'select box/dropdown

                        If bSelectFirstRun = True Then
                            m_oResponse.Write "<td class=""questionres"">"
                            m_oResponse.Write "<SELECT NAME=" & Chr(34) & "QID"
& lQuestionID & Chr(34) & ">"
                            bSelectFirstRun = False
                        End If


                        If oQuesRespRS("default") = 1 Then
                            m_oResponse.Write "<OPTION NAME=" & Chr(34) & "QID"
& lQuestionID & Chr(34) & " VALUE=" & Chr(34) & oQuesRespRS("responseid") &
Chr(34) & " selected>" & oQuesRespRS("text") & "</option>" & vbCrLf
                        Else
                            m_oResponse.Write "<OPTION NAME=" & Chr(34) & "QID"
& lQuestionID & Chr(34) & " VALUE=" & Chr(34) & oQuesRespRS("responseid") &
Chr(34) & " >" & oQuesRespRS("text") & "</option>" & vbCrLf
                        End If


                    Case 4  'checkbox
                        m_oResponse.Write "<td class=""questionres"">"
                        If oQuesRespRS("default") = 1 Then
                            m_oResponse.Write "<INPUT TYPE=""checkbox"" NAME="
& Chr(34) & "QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) &
oQuesRespRS("responseid") & Chr(34) & " checked>" & oQuesRespRS("text")
                        Else
                            m_oResponse.Write "<INPUT TYPE=""checkbox"" NAME="
& Chr(34) & "QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) &
oQuesRespRS("responseid") & Chr(34) & ">" & oQuesRespRS("text")
                        End If
                        m_oResponse.Write "</td></tr>" & vbCrLf
```

```
                    Case Else
                        'Default here
                End Select
                oQuesRespRS.MoveNext 'Goto the next possible response for this
question


            Wend
            If bSelectFirstRun = False Then
                'The select has already been displayed
                bSelectFirstRun = True
                m_oResponse.Write "</SELECT>"
                m_oResponse.Write "</td></tr>" & vbCrLf
            End If
        ElseIf iTypeID = 5 Then
            Set oQuesRespRS = oQuestionRS("questionresponses").Value
            If oQuesRespRS.EOF Then
              m_oResponse.Write "<tr><td>  </td><td
class=""questionres"">"
              m_oResponse.Write "<INPUT TYPE=""TEXT"" NAME=" & Chr(34) & "QID"
& lQuestionID & Chr(34) & " VALUE="" "">"
              m_oResponse.Write "</td></tr>" & vbCrLf
            Else
              m_oResponse.Write "<tr><td>  </td><td
class=""questionres"">"
              m_oResponse.Write "<INPUT TYPE=""TEXT"" NAME=" & Chr(34) & "QID"
& lQuestionID & Chr(34) & " VALUE=" & Chr(34) & (oQuesRespRS("value") & "") &
Chr(34) & ">"
              m_oResponse.Write "</td></tr>" & vbCrLf
            End If

        ElseIf iTypeID = 6 Then
            Set oQuesRespRS = oQuestionRS("questionresponses").Value
            If oQuesRespRS.EOF Then
              m_oResponse.Write "<tr><td>  </td><td
class=""questionres"">"
               'm_oResponse.Write "<TEXTAREA NAME=" & Chr(34) & "QID" &
lQuestionID & Chr(34) & ">" & oQuesRespRS("value") & "</textarea><br>"
              m_oResponse.Write "<TEXTAREA NAME=" & Chr(34) & "QID" &
lQuestionID & Chr(34) & " rows=4 cols=75> </textarea>"
              m_oResponse.Write "</td></tr>" & vbCrLf
            Else
              m_oResponse.Write "<tr><td>  </td><td
class=""questionres"">"
               'm_oResponse.Write "<TEXTAREA NAME=" & Chr(34) & "QID" &
lQuestionID & Chr(34) & ">" & oQuesRespRS("value") & "</textarea><br>"
              m_oResponse.Write "<TEXTAREA NAME=" & Chr(34) & "QID" &
lQuestionID & Chr(34) & " rows=4 cols=75>" & (oQuesRespRS("value") & "") &
"</textarea>"
              m_oResponse.Write "</td></tr>" & vbCrLf
            End If
        Else
            'Do nothing IS is not set up or the TypeID is invalid.
            m_oResponse.Write "ID Invalid"
        End If 'If iTypeID<>5 && iTypeID<>6

        oQuestionRS.MoveNext

        'Move to the next question.
        Wend    'While NOT oQuestionRS.Eof
        m_oResponse.Write "<br>"
        m_oResponse.Write "<input type=""hidden"" name=""sid"" value=" &
Chr(34) & m_oSession.SessionID & Chr(34) & ">" & vbCrLf
```

```
        m_oResponse.Write "<input type=""hidden"" name=""bid"" value=" &
Chr(34) & m_lBusID & Chr(34) & ">" & vbCrLf

        If m_bJavaScriptDisplay = True Then
          UpdateStats "<center>All Done</center>", "", "", "", "", ""
        End If


        m_oResponse.Write "<tr><td><input type=""submit"" value=""Rate It""
id=1 name=1></td>"
        m_oResponse.Write "<td><input type=""reset"" value=""Clear
Form""></td></tr>"
        m_oResponse.Write "</table>" 'End the table for this question
        m_oResponse.Write "</form>" & vbCrLf

        If m_bJavaScriptDisplay = True Then
          iReturn = CloseProgressWindow()
        End If
    Else
        m_oResponse.Write "<center><font color=""#990000"">No Questions For
This SubCategory Yet</font></center>"
    End If

If oShapeConn.State = adStateOpen Then
  oShapeConn.Close
End If

If oQuestionRS.State = adStateOpen Then
  oQuestionRS.Close
End If

If oQuesRespRS.State = adStateOpen Then
  oQuesRespRS.Close
End If


Set oShapeConn = Nothing
Set oQuestionRS = Nothing
Set oQuesRespRS = Nothing


End Sub



Public Sub displayValidationForm()
'PURPOSE: This subroutine will display a rating form for a particular
'         business.  It should be used when the user has POSTED data
'         to a particular page.  It will automatically repopulate the
'         form based on their answers.
'         Additionally, it will only allow them to continue to enterrate.asp
'         page once all the data has successfully been filled.
'
'IN: Properities set by user/code.
'OUT: No return variable

Dim oShapeConn As ADODB.Connection
Dim oQuestionRS As ADODB.Recordset, oQuesRespRS As ADODB.Recordset
Dim sSQL As String

'This holds the form tag and is randomly generated at the end of the
'of the while loop.  It will point to a pass page and a fail page depending
'if the form has been filled out.
```

```
Dim sFormOutputTag As String

'String that will be sent out to the browser once the form has been validated.
'Also it is more efficient than calling Response.Write method over and over.
Dim sOutput As String
'Keeps count of the questions for display purposes only.
Dim intCounter As Integer

'These hold form values for the Stars, Coins and the Thumbs.
Dim iStar As Single, iCoin As Single
Dim iThumb As Integer

Dim lAnswerValue As Long 'Holds the answer value from the form
Dim sAnswerText As String   'Holds the answer text from the form

Dim sRSQuestionText As String   'Holds the question text from the Question's
Recordset
Dim iTypeID As Integer  'Holds the type id from the Question Recordset
Dim lQuestionID As Long  'Holds the current questionid from the Question
Recordset
Dim sQuestionID As String   'String created from the Recordset's question id
(QID + current questionid)
Dim bSelectFirstRun As Boolean 'Flag for drop downs determins if it is the
first question for a select box.
Dim bFormComplete As Boolean
Dim lRSResponseID  As Long 'Response ID from the database
Dim sRSResponseText As String 'Response Text from the database
Dim sAnswerValue As String 'Text response user entered from the Form
Dim iReturn As Integer

Dim oSCConn As ADODB.Connection
Dim oSCCmd As ADODB.Command

Dim vList As Variant
Dim vItem As Variant

Dim bFoundItem As Boolean



bSelectFirstRun = True
'Assume that the form is complete, this will change to false if a question is
missing
bFormComplete = True

'Hey we need the stars coins and thumbs in here....
iStar = CSng(m_oRequest.Form("stars"))
iCoin = CSng(m_oRequest.Form("coins"))
iThumb = CSng(m_oRequest.Form("goback"))

'Now lets begin to check to make sure that we have all the questions needed.

  If m_bJavaScriptDisplay = True Then
    iReturn = OpenProgressWindow("<center>Generating Validation Form</center>")
  End If

  'If the subcatID is less than one then get it
  'this may be less than one if this is called from the enterrate.asp
  'page and all the data was not there.
  If m_lSubCatID < 1 Then
    Set oSCConn = CreateObject("ADODB.Connection")
    Set oSCCmd = CreateObject("ADODB.Command")
```

```
    oSCConn.Open m_sDatabaseConnectString
    oSCCmd.ActiveConnection = oSCConn
    oSCCmd.CommandType = adCmdStoredProc
    oSCCmd.CommandText = "sp_ValidateBusinessWithExtraInfo"
    oSCCmd.Parameters.Append oSCCmd.CreateParameter("@busid", adInteger,
adParamInput, 4, m_lBusID)
    oSCCmd.Parameters.Append oSCCmd.CreateParameter("@isvalid", adInteger,
adParamOutput, 0, 0)
    oSCCmd.Parameters.Append oSCCmd.CreateParameter("@subcatid", adInteger,
adParamOutput, 0, 0)
    oSCCmd.Parameters.Append oSCCmd.CreateParameter("@busname", adVarChar,
adParamOutput, 50, "")
    oSCCmd.Execute

    m_lSubCatID = CLng(oSCCmd("@subcatid"))

    Set oSCCmd = Nothing
    oSCConn.Close
    Set oSCConn = Nothing

  End If

    'This subroutine will display the rating form to the user.

    Set oShapeConn = CreateObject("ADODB.Connection")
    Set oQuestionRS = CreateObject("ADODB.Recordset")
    Set oQuesRespRS = CreateObject("ADODB.Recordset")

    'Generate the SQL statement to get all the questions and responses for this
category.
    sSQL = "SHAPE {{ CALL dbo.sp_BusinessQuestionsBySubCategory(" & m_lSubCatID
& ")}} "
    sSQL = sSQL & "APPEND ({{ CALL dbo.sp_GetAllPossibleResponses}} "
    sSQL = sSQL & "RELATE questionid TO questionid) AS questionresponses"

    oShapeConn.Provider = "MSDataShape"
    oShapeConn.Open m_sDatabaseConnectString
    'Open the recordset with adOpenStatic=3 and adLockReadOnly=1
    oQuestionRS.Open sSQL, oShapeConn, adOpenStatic, adLockReadOnly

    intCounter = 0
    If Not oQuestionRS.EOF Then


            'sOutput = sOutput &  "<form action=""vratebiz.asp""
method=""post"" id=form1 name=form1>" & vbCrLf

            'sOutput = sOutput & "<form action=""enterrate.asp""
method=""post"" id=form1 name=form1>" & vbCrLf

      sOutput = sOutput & "<table align=""center"" width=""100%""
class=""quest"">"
            sOutput = sOutput & "<tr><td class=""questiontext""
colspan=""11"">How many stars do you give this business?</td></tr>" & vbCrLf
            sOutput = sOutput & "<tr><td class=""questionres"">"

            sOutput = sOutput & "<input type=""radio"" name=""stars""
value=""0"" "
            If iStar = 0 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">0  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""stars""
value="".5"" "
```

```
            If iStar = 0.5 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">.5  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""stars""
value=""1"" "
            If iStar = 1 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">1  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""stars""
value=""1.5"" "
            If iStar = 1.5 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">1.5  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""stars""
value=""2"" "
            If iStar = 2 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">2  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""stars""
value=""2.5"" "
            If iStar = 2.5 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">2.5  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""stars""
value=""3"" "
            If iStar = 3 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">3  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""stars""
value=""3.5"" "
            If iStar = 3.5 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">3.5  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""stars""
value=""4"" "
            If iStar = 4 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">4  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""stars""
value=""4.5"" "
            If iStar = 4.5 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">4.5  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""stars""
value=""5"" "
            If iStar = 5 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">5  "
            sOutput = sOutput & "</td></tr>" & vbCrLf
            sOutput = sOutput & "</table>" & vbCrLf

            sOutput = sOutput & "<table align=""center"" width=""100%""
class=""quest"">"
            sOutput = sOutput & "<tr><td class=""questiontext""
colspan=""11"">How expensive is this business?</td></tr>" & vbCrLf
            sOutput = sOutput & "<tr><td class=""questionres"">"
            sOutput = sOutput & "<input type=""radio"" name=""coins""
value=""0"" "
            If iCoin = 0 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">0  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""coins""
value="".5"" "
            If iCoin = 0.5 Then sOutput = sOutput & "checked"
```

```
            sOutput = sOutput & ">.5  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""coins""
value=""1"" "
            If iCoin = 1 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">1  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""coins""
value=""1.5"" "
            If iCoin = 1.5 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">1.5  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""coins""
value=""2"" "
            If iCoin = 2 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">2  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""coins""
value=""2.5"" "
            If iCoin = 2.5 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">2.5  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""coins""
value=""3"" "
            If iCoin = 3 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">3  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""coins""
value=""3.5"" "
            If iCoin = 3.5 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">3.5  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""coins""
value=""4"" "
            If iCoin = 4 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">4  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""coins""
value=""4.5"" "
            If iCoin = 4.5 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">4.5  </td><td>"

            sOutput = sOutput & "<input type=""radio"" name=""coins""
value=""5"" "
            If iCoin = 5 Then sOutput = sOutput & "checked"
            sOutput = sOutput & ">5  "
            sOutput = sOutput & "</td></tr>" & vbCrLf
            sOutput = sOutput & "</table>" & vbCrLf


            sOutput = sOutput & "<table align=""center"" width=""100%""
class=""quest"">" & vbCrLf
            sOutput = sOutput & "<tr><td class=""questiontext""
colspan=""2"">Would you go back?</td></tr>" & vbCrLf
            sOutput = sOutput & "<tr><td class=""questionres"" colspan=""2"">"
            sOutput = sOutput & "<input type=""radio"" name=""goback""
value=""1"" "
            If iThumb = 1 Then sOutput = sOutput & "checked"
            sOutput = sOutput & "> Yes   "

            sOutput = sOutput & "<input type=""radio"" name=""goback""
value=""0"" "
            If iThumb = 0 Then sOutput = sOutput & "checked"
```

```
            sOutput = sOutput & "> No "
            sOutput = sOutput & "</td></tr>" & vbCrLf
            sOutput = sOutput & "</table>"

            sOutput = sOutput & "<table align=""center"" width=""100%""
class=""quest"">" & vbCrLf


        While Not oQuestionRS.EOF
        intCounter = intCounter + 1 'Increment the question counter.
        lAnswerValue = 0
        sAnswerText = ""


        sRSQuestionText = (oQuestionRS("text") & "") 'Grab the question Text
        iTypeID = oQuestionRS("typeid") 'Grab the question type id
        lQuestionID = oQuestionRS("questionid") 'Grab the unique QuestionID
        sQuestionID = "QID" & lQuestionID 'Put toegether the name to get the
answer back.

        'sOutput = sOutput &  "<tr><td colspan=""2"" class=""questiontext"">"
        'sOutput = sOutput &  intCounter & ".)  " &
oQuestionRS("text") & "<br>"
        'sOutput = sOutput &  "</td></tr>" & vbCrLf
        If m_bJavaScriptDisplay = True Then
          UpdateStats "<center>Displaying Questions</center>", "Question No: "
& intCounter, "", "", "", ""
        End If

        If iTypeID <> 5 And iTypeID <> 6 Then
        'This means that the question is not a text or textarea question.
            bSelectFirstRun = True  'This is the begining of a new question set
the flag back to false.

            If iTypeID <> 4 Then
              lAnswerValue = CInt(m_oRequest.Form(sQuestionID) + 0) 'Get the
answer back for this question.
              If lAnswerValue = 0 Then
                'They must have not answered this question
                bFormComplete = False
                sOutput = sOutput & "<tr><td colspan=""2""
class=""badquestiontext"">"
                sOutput = sOutput & intCounter & ".)  " &
sRSQuestionText & "<br>"
                sOutput = sOutput & "</td></tr>" & vbCrLf
              Else
                sOutput = sOutput & "<tr><td colspan=""2""
class=""questiontext"">"
                sOutput = sOutput & intCounter & ".)  " &
sRSQuestionText & "<br>"
                sOutput = sOutput & "</td></tr>" & vbCrLf
              End If

            Else
              Set vList = m_oRequest.Form(sQuestionID)
              If CStr(vList) = "" Then
                'They must have not answered this question
                bFormComplete = False
                sOutput = sOutput & "<tr><td colspan=""2""
class=""badquestiontext"">"
                sOutput = sOutput & intCounter & ".)  " &
sRSQuestionText & "<br>"
                sOutput = sOutput & "</td></tr>" & vbCrLf
```

```
              Else
                sOutput = sOutput & "<tr><td colspan=""2""
class=""questiontext"">"
                sOutput = sOutput & intCounter & ".)  " &
sRSQuestionText & "<br>"
                sOutput = sOutput & "</td></tr>" & vbCrLf
              End If

            End If

            'sOutput = sOutput & "Answer is:" & lAnswerValue & ":<br>"



            Set oQuesRespRS = oQuestionRS("questionresponses").Value


            While Not oQuesRespRS.EOF
                lRSResponseID = 0
                sRSResponseText = ""

                lRSResponseID = CLng(oQuesRespRS("responseid"))
                sRSResponseText = CStr(oQuesRespRS("text") & "")

                If bSelectFirstRun = True Then
                  sOutput = sOutput & "<tr>"
                  sOutput = sOutput & "<td>  </td>"
                End If

                Select Case iTypeID
                    Case 1  'unassigned
                        'Don't display any answers at this time 1 is a default.
                    Case 2  'radio button
                        sOutput = sOutput & "<td class=""questionres"">"
                        If lAnswerValue = lRSResponseID Then
                            sOutput = sOutput & "<INPUT TYPE=""radio"" NAME=" &
Chr(34) & "QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) & lRSResponseID &
Chr(34) & " checked>" & sRSResponseText & vbCrLf
                        Else
                            sOutput = sOutput & "<INPUT TYPE=""radio"" NAME=" &
Chr(34) & "QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) & lRSResponseID &
Chr(34) & " >" & sRSResponseText & vbCrLf
                        End If
                        sOutput = sOutput & "</td></tr>" & vbCrLf
                    Case 3  'select box/dropdown

                        If bSelectFirstRun = True Then
                            sOutput = sOutput & "<td class=""questionres"">"
                            sOutput = sOutput & "<SELECT NAME=" & Chr(34) &
"QID" & lQuestionID & Chr(34) & ">"
                            bSelectFirstRun = False
                        End If


                        If lAnswerValue = oQuesRespRS("responseid") Then
                            sOutput = sOutput & "<OPTION NAME=" & Chr(34) &
"QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) & lRSResponseID & Chr(34) &
" selected>" & sRSResponseText & "</option>" & vbCrLf
                        Else
                            sOutput = sOutput & "<OPTION NAME=" & Chr(34) &
"QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) & lRSResponseID & Chr(34) &
" >" & sRSResponseText & "</option>" & vbCrLf
                        End If
```

```
                    Case 4  'checkbox
                        sOutput = sOutput & "<td class=""questionres"">"
                        'If lAnswerValue = oQuesRespRS("responseid") Then
                        bFoundItem = False

                        For Each vItem In vList
                          If CLng(vItem) = lRSResponseID Then
                            bFoundItem = True
                          End If
                        Next
                        If bFoundItem = True Then
                            sOutput = sOutput & "<INPUT TYPE=""checkbox""
NAME=" & Chr(34) & "QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) &
lRSResponseID & Chr(34) & " checked>" & sRSResponseText
                        Else
                            sOutput = sOutput & "<INPUT TYPE=""checkbox""
NAME=" & Chr(34) & "QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) &
lRSResponseID & Chr(34) & ">" & sRSResponseText
                        End If
                        sOutput = sOutput & "</td></tr>" & vbCrLf
                    Case Else
                        'Default here
                End Select
                oQuesRespRS.MoveNext 'Goto the next possible response for this
question


            Wend
            If bSelectFirstRun = False Then
                'The select has already been displayed
                bSelectFirstRun = True
                sOutput = sOutput & "</SELECT>"
                sOutput = sOutput & "</td></tr>" & vbCrLf
            End If

        ElseIf iTypeID = 5 Then
            sAnswerValue = Trim(m_oRequest.Form(sQuestionID) & "") 'Get the
answer back for this question.
            If Len(sAnswerValue) <= 0 Then
                            bFormComplete = False
                            sOutput = sOutput & "<tr><td colspan=""2""
class=""badquestiontext"">"
                            sOutput = sOutput & intCounter & ".)  " &
sRSQuestionText & "<br>"
                            sOutput = sOutput & "</td></tr>" & vbCrLf
                        Else
                            sOutput = sOutput & "<tr><td colspan=""2""
class=""questiontext"">"
                            sOutput = sOutput & intCounter & ".)  " &
sRSQuestionText & "<br>"
                            sOutput = sOutput & "</td></tr>" & vbCrLf
                        End If
            sOutput = sOutput & "<tr><td>  </td><td
class=""questionres"">"
            sOutput = sOutput & "<INPUT TYPE=""TEXT"" NAME=" & Chr(34) & "QID"
& lQuestionID & Chr(34) & " VALUE=" & Chr(34) & sAnswerValue & Chr(34) & ">"
            sOutput = sOutput & "</td></tr>" & vbCrLf


        ElseIf iTypeID = 6 Then
                    sAnswerValue = Trim(m_oRequest.Form(sQuestionID) & "")
```

```
            If Len(sAnswerValue) <= 0 Then
                            bFormComplete = False
                            sOutput = sOutput & "<tr><td colspan=""2""
class=""badquestiontext"">"
                            sOutput = sOutput & intCounter & ".)  " &
sRSQuestionText & "<br>"
                            sOutput = sOutput & "</td></tr>" & vbCrLf
                        Else
                            sOutput = sOutput & "<tr><td colspan=""2""
class=""questiontext"">"
                            sOutput = sOutput & intCounter & ".)  " &
sRSQuestionText & "<br>"
                            sOutput = sOutput & "</td></tr>" & vbCrLf
                        End If

            sOutput = sOutput & "<tr><td>  </td><td
class=""questionres"">"
            'sOutput = sOutput &  "<TEXTAREA NAME=" & Chr(34) & "QID" &
lQuestionID & Chr(34) & ">" & oQuesRespRS("value") & "</textarea><br>"
            sOutput = sOutput & "<TEXTAREA NAME=" & Chr(34) & "QID" &
lQuestionID & Chr(34) & " rows=4 cols=75>" & sAnswerValue & "</textarea>"
            sOutput = sOutput & "</td></tr>" & vbCrLf
        Else
            'Do nothing IS is not set up or the TypeID is invalid.
            sOutput = sOutput & "ID Invalid"
        End If 'If iTypeID<>5 && iTypeID<>6

        oQuestionRS.MoveNext



        'Move to the next question.
        Wend    'While NOT oQuestionRS.Eof

        sOutput = sOutput & "<tr><td align=""center"">"
        sOutput = sOutput & "<input type=""hidden"" name=""sid"" value=" &
Chr(34) & m_oSession.SessionID & Chr(34) & ">" & vbCrLf
        sOutput = sOutput & "<input type=""hidden"" name=""bid"" value=" &
Chr(34) & m_lBusID & Chr(34) & ">" & vbCrLf

        If bFormComplete = False Then
            sOutput = sOutput & "<input type=""submit"" value=""Validate It"">"
        Else
            sOutput = sOutput & "<input type=""submit"" value=""Post It!"">"
        End If


        sOutput = sOutput & "<input type=""reset"" value=""Clear Form"">"

        sOutput = sOutput & "</td></tr>"


        sOutput = sOutput & "</table>"  'End the table for this question
        sOutput = sOutput & "</form>" & vbCrLf

        If bFormComplete = True Then
          sFormOutputTag = "<form action=" & Chr(34) & m_sValidationSuccessPage
& Chr(34) & " method=""post"" id=form1 name=form1>" & vbCrLf
        Else
          sFormOutputTag = "<form action=" & Chr(34) & m_sValidationFailPage &
Chr(34) & " method=""post"" id=form1 name=form1>" & vbCrLf
        End If
```

```
        If m_bJavaScriptDisplay = True Then
          UpdateStats "<center>All Done</center>", "", "", "", "", ""
        End If
    Else
        sOutput = sOutput & "<center><font color=""#990000"">Invalid Business
ID</font></center>"
    End If



    'oShapeConn.Close
    'oQuestionRS.Close
    'oQuesRespRS.Close

    Set oShapeConn = Nothing
    Set oQuestionRS = Nothing
    Set oQuesRespRS = Nothing

    If m_bJavaScriptDisplay = True Then
      iReturn = CloseProgressWindow()
    End If


    'Put the final string together here...
    sOutput = sFormOutputTag & sOutput

    m_oResponse.Write sOutput


End Sub

Public Sub ShowReRateForm()
'PURPOSE: This subroutine will display the question form to the user
'         with their previous rating information already filled out
'
'
'
'
'
'IN: Properities set by user/code.
'OUT: No return variable



  Dim oShapeConn As ADODB.Connection, oConn As ADODB.Connection
  Dim oQuestionRS As ADODB.Recordset, oQuesRespRS As ADODB.Recordset
  Dim oUserRespRS As ADODB.Recordset, oUserTextRestRS As ADODB.Recordset
  Dim oUserTextRespRS As ADODB.Recordset

  Dim oAtAGlanceCmd As ADODB.Command

  Dim iCoin As Single, iStar As Single
  Dim iThumb As Integer

  Dim intCounter As Integer
  Dim lAnswerValue As Long

  Dim sAnswerText As String
  Dim sRSQuestionText As String
  Dim iTypeID As Integer
  Dim lQuestionID As Long
  Dim bSelectFirstRun As Boolean
  Dim lRSResponseID As Long
```

```
  Dim sResponseText As String

  Dim sRSResponseText As String
  Dim sSQL As String
  Dim sAnswerValue As String

  Dim vList As Variant
  Dim vItem As Variant
  Dim bRecordFound As Boolean

  Set oShapeConn = CreateObject("ADODB.Connection")
  Set oQuestionRS = CreateObject("ADODB.Recordset")  'Holds the question text.
  Set oQuesRespRS = CreateObject("ADODB.Recordset")  'Holds the possible
responses.
  Set oUserRespRS = CreateObject("ADODB.Recordset")  'Holds the actual user
response.
  Set oUserTextRespRS = CreateObject("ADODB.Recordset")  'Holds the acutal user
text response.

  Set oConn = CreateObject("ADODB.Connection")
  Set oAtAGlanceCmd = CreateObject("ADODB.Command")
  'Open a database connection
  oConn.Open m_sDatabaseConnectString

  'Get the at a glance statistics that the user had entered before.
  oAtAGlanceCmd.ActiveConnection = oConn
  oAtAGlanceCmd.CommandType = adCmdStoredProc
  oAtAGlanceCmd.CommandText = "sp_GetUserResponseCoinStarThumb"
  'name, datatype 3=int::4=decimal, in/out, size, value
  oAtAGlanceCmd.Parameters.Append oAtAGlanceCmd.CreateParameter("@busurid",
adInteger, adParamInput, 4, m_lBusToUserID)
  oAtAGlanceCmd.Parameters.Append oAtAGlanceCmd.CreateParameter("@coins",
adSingle, adParamOutput, 0, 0)
  oAtAGlanceCmd.Parameters.Append oAtAGlanceCmd.CreateParameter("@stars",
adSingle, adParamOutput, 0, 0)
  oAtAGlanceCmd.Parameters.Append oAtAGlanceCmd.CreateParameter("@thumbs",
adInteger, adParamOutput, 0, 0)
  oAtAGlanceCmd.Execute

  'Get the values back from the stored procedure.
  iCoin = oAtAGlanceCmd("@coins")
  iStar = oAtAGlanceCmd("@stars")
  iThumb = oAtAGlanceCmd("@thumbs")

  Set oAtAGlanceCmd = Nothing
  oConn.Close
  Set oConn = Nothing


    'Generate the SQL statement to get all the questions and responses for this
category.
    sSQL = "SHAPE {{ CALL dbo.sp_BusinessQuestionsBySubCategory(" & m_lSubCatID
& ")}} "
    sSQL = sSQL & "APPEND "
    sSQL = sSQL & "({{ CALL dbo.sp_GetAllPossibleResponses}} "
    sSQL = sSQL & "RELATE questionid TO questionid) AS questionresponses, "
    sSQL = sSQL & "({{ CALL dbo.sp_GetUserResponsesBasedOnBUSURID(" &
m_lBusToUserID & ")}} "
    sSQL = sSQL & "RELATE questionid TO tURquestionid) AS userResponses, "
    sSQL = sSQL & "({{ CALL dbo.sp_GetUserTextResponsesBasedOnBUSURID(" &
m_lBusToUserID & ")}}"
    sSQL = sSQL & "RELATE questionid TO tTDquestionid) AS userTextResponses"
```

```
    oShapeConn.Provider = "MSDataShape"
    oShapeConn.Open m_sDatabaseConnectString
    'Open the recordset with adOpenStatic=3 and adLockReadOnly=1
    oQuestionRS.Open sSQL, oShapeConn, adOpenStatic, adLockReadOnly


    intCounter = 0
    If Not oQuestionRS.EOF Then


      m_oResponse.Write "<form action=""vratebiz.asp"" method=""post"" id=form1
name=form1>" & vbCrLf
      m_oResponse.Write "<table align=""center"" width=""100%""
class=""quest"">"
      m_oResponse.Write "<tr><td class=""questiontext"" colspan=""11"">How many
stars do you give this business?</td></tr>" & vbCrLf
      m_oResponse.Write "<tr><td class=""questionres"">"

      m_oResponse.Write "<input type=""radio"" name=""stars"" value=""0"" "
      If iStar = 0 Then m_oResponse.Write "checked"
      m_oResponse.Write ">0  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""stars"" value="".5"" "
      If iStar = 0.5 Then m_oResponse.Write "checked"
      m_oResponse.Write ">.5  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""stars"" value=""1"" "
      If iStar = 1 Then m_oResponse.Write "checked"
      m_oResponse.Write ">1  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""stars"" value=""1.5"" "
      If iStar = 1.5 Then m_oResponse.Write "checked"
      m_oResponse.Write ">1.5  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""stars"" value=""2"" "
      If iStar = 2 Then m_oResponse.Write "checked"
      m_oResponse.Write ">2  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""stars"" value=""2.5"" "
      If iStar = 2.5 Then m_oResponse.Write "checked"
      m_oResponse.Write ">2.5  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""stars"" value=""3"" "
      If iStar = 3 Then m_oResponse.Write "checked"
      m_oResponse.Write ">3  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""stars"" value=""3.5"" "
      If iStar = 3.5 Then m_oResponse.Write "checked"
      m_oResponse.Write ">3.5  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""stars"" value=""4"" "
      If iStar = 4 Then m_oResponse.Write "checked"
      m_oResponse.Write ">4  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""stars"" value=""4.5"" "
      If iStar = 4.5 Then m_oResponse.Write "checked"
      m_oResponse.Write ">4.5  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""stars"" value=""5"" "
      If iStar = 5 Then m_oResponse.Write "checked"
      m_oResponse.Write ">5  "
      m_oResponse.Write "</td></tr>" & vbCrLf
```

```
      m_oResponse.Write "</table>" & vbCrLf

      m_oResponse.Write "<table align=""center"" width=""100%""
class=""quest"">"
      m_oResponse.Write "<tr><td class=""questiontext"" colspan=""11"">How
expensive is this business?</td></tr>" & vbCrLf
      m_oResponse.Write "<tr><td class=""questionres"">"
      m_oResponse.Write "<input type=""radio"" name=""coins"" value=""0"" "
      If iCoin = 0 Then m_oResponse.Write "checked"
      m_oResponse.Write ">0  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""coins"" value="".5"" "
      If iCoin = 0.5 Then m_oResponse.Write "checked"
      m_oResponse.Write ">.5  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""coins"" value=""1"" "
      If iCoin = 1 Then m_oResponse.Write "checked"
      m_oResponse.Write ">1  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""coins"" value=""1.5"" "
      If iCoin = 1.5 Then m_oResponse.Write "checked"
      m_oResponse.Write ">1.5  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""coins"" value=""2"" "
      If iCoin = 2 Then m_oResponse.Write "checked"
      m_oResponse.Write ">2  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""coins"" value=""2.5"" "
      If iCoin = 2.5 Then m_oResponse.Write "checked"
      m_oResponse.Write ">2.5  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""coins"" value=""3"" "
      If iCoin = 3 Then m_oResponse.Write "checked"
      m_oResponse.Write ">3  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""coins"" value=""3.5"" "
      If iCoin = 3.5 Then m_oResponse.Write "checked"
      m_oResponse.Write ">3.5  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""coins"" value=""4"" "
      If iCoin = 4 Then m_oResponse.Write "checked"
      m_oResponse.Write ">4  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""coins"" value=""4.5"" "
      If iCoin = 4.5 Then m_oResponse.Write "checked"
      m_oResponse.Write ">4.5  </td><td>"

      m_oResponse.Write "<input type=""radio"" name=""coins"" value=""5"" "
      If iCoin = 5 Then m_oResponse.Write "checked"
      m_oResponse.Write ">5  "
      m_oResponse.Write "</td></tr>" & vbCrLf
      m_oResponse.Write "</table>" & vbCrLf


      m_oResponse.Write "<table align=""center"" width=""100%""
class=""quest"">" & vbCrLf
      m_oResponse.Write "<tr><td class=""questiontext"" colspan=""2"">Would you
go back?</td></tr>" & vbCrLf
      m_oResponse.Write "<tr><td class=""questionres"" colspan=""2"">"
      m_oResponse.Write "<input type=""radio"" name=""goback"" value=""1"" "
      If iThumb = 1 Then m_oResponse.Write "checked"
      m_oResponse.Write "> Yes   "
```

```
      m_oResponse.Write "<input type=""radio"" name=""goback"" value=""0"" "
      If iThumb = 0 Then m_oResponse.Write "checked"
      m_oResponse.Write "> No "
      m_oResponse.Write "</td></tr>" & vbCrLf
      m_oResponse.Write "</table>"

      m_oResponse.Write "<table align=""center"" width=""100%""
class=""quest"">" & vbCrLf


        While Not oQuestionRS.EOF
        intCounter = intCounter + 1 'Increment the question counter.
        lAnswerValue = 0
        sAnswerText = ""

        sRSQuestionText = (oQuestionRS("text") & "") 'Grab the question Text
        iTypeID = oQuestionRS("typeid") 'Grab the question type id
        lQuestionID = oQuestionRS("questionid") 'Grab the unique QuestionID


        m_oResponse.Write "<tr><td colspan=""2"" class=""questiontext"">"
        m_oResponse.Write intCounter & ".)  " & oQuestionRS("text") &
"<br>"
        m_oResponse.Write "</td></tr>" & vbCrLf

        If iTypeID <> 5 And iTypeID <> 6 And iTypeID <> 4 Then
        'This means that the question is not a text or textarea question.
            bSelectFirstRun = True  'This is the begining of a new question set
the flag back to false.

            'Grab a recordset of the possible responses.
            Set oQuesRespRS = oQuestionRS("questionresponses").Value
            'Grab a recordset of the actual user responses.
            Set oUserRespRS = oQuestionRS("userResponses").Value

            'Initialize the variable to zero
            lAnswerValue = 0
            If Not oUserRespRS.EOF Then
              lAnswerValue = CInt(oUserRespRS("tURresponseid") + 0) 'Get the
users response id.
            End If


            While Not oQuesRespRS.EOF
                lRSResponseID = 0
                sRSResponseText = ""

                lRSResponseID = CLng(oQuesRespRS("responseid"))
                sRSResponseText = CStr(oQuesRespRS("text") & "")

                If bSelectFirstRun = True Then
                  m_oResponse.Write "<tr>"
                  m_oResponse.Write "<td>  </td>"
                End If

                Select Case iTypeID
                    Case 1  'unassigned
                        'Don't display any answers at this time 1 is a default.
                    Case 2  'radio button
                        m_oResponse.Write "<td class=""questionres"">"
                        If lAnswerValue = lRSResponseID Then
```

```
                            m_oResponse.Write "<INPUT TYPE=""radio"" NAME=" &
Chr(34) & "QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) & lRSResponseID &
Chr(34) & " checked>" & sRSResponseText & vbCrLf
                        Else
                            m_oResponse.Write "<INPUT TYPE=""radio"" NAME=" &
Chr(34) & "QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) & lRSResponseID &
Chr(34) & " >" & sRSResponseText & vbCrLf
                        End If
                        m_oResponse.Write "</td></tr>" & vbCrLf
                    Case 3  'select box/dropdown

                        If bSelectFirstRun = True Then
                            m_oResponse.Write "<td class=""questionres"">"
                            m_oResponse.Write "<SELECT NAME=" & Chr(34) & "QID"
& lQuestionID & Chr(34) & ">"
                            bSelectFirstRun = False
                        End If


                        If lAnswerValue = oQuesRespRS("responseid") Then
                            m_oResponse.Write "<OPTION NAME=" & Chr(34) & "QID"
& lQuestionID & Chr(34) & " VALUE=" & Chr(34) & lRSResponseID & Chr(34) & "
selected>" & sRSResponseText & "</option>" & vbCrLf
                        Else
                            m_oResponse.Write "<OPTION NAME=" & Chr(34) & "QID"
& lQuestionID & Chr(34) & " VALUE=" & Chr(34) & lRSResponseID & Chr(34) & " >"
& sRSResponseText & "</option>" & vbCrLf
                        End If


                    Case 4  'checkbox
                        m_oResponse.Write "<td class=""questionres"">"
                        If lAnswerValue = oQuesRespRS("responseid") Then
                            m_oResponse.Write "<INPUT TYPE=""checkbox"" NAME="
& Chr(34) & "QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) & lRSResponseID
& Chr(34) & " checked>" & sRSResponseText
                        Else
                            m_oResponse.Write "<INPUT TYPE=""checkbox"" NAME="
& Chr(34) & "QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) & lRSResponseID
& Chr(34) & ">" & sRSResponseText
                        End If
                        m_oResponse.Write "</td></tr>" & vbCrLf
                    Case Else
                        'Default here
                End Select
                oQuesRespRS.MoveNext 'Goto the next possible Response for this
question


            Wend  'End Of While responses to question.
            If bSelectFirstRun = False Then
                'The select has already been displayed
                bSelectFirstRun = True
                m_oResponse.Write "</SELECT>"
                m_oResponse.Write "</td></tr>" & vbCrLf
            End If

        ElseIf iTypeID = 4 Then 'Checkbox
            'Grab a recordset of the possible responses.
            Set oQuesRespRS = oQuestionRS("questionresponses").Value
            'Grab a recordset of the actual user responses.
            Set oUserRespRS = oQuestionRS("userResponses").Value
```

```
            While Not oQuesRespRS.EOF
              lRSResponseID = CLng(oQuesRespRS("responseid"))
              sRSResponseText = CStr(oQuesRespRS("text") & "")
              bRecordFound = False

              While Not oUserRespRS.EOF And bRecordFound = False
                If CLng(oUserRespRS("tURresponseid") + 0) = lRSResponseID Then
'Get the users response id.
                  bRecordFound = True
                End If
                oUserRespRS.MoveNext
              Wend  'WHILE ACTUAL RESPONSES
              oUserRespRS.MoveFirst
              m_oResponse.Write "<td class=""questionres"">"
              If bRecordFound = True Then
                m_oResponse.Write "<INPUT TYPE=""checkbox"" NAME=" & Chr(34) &
"QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) & lRSResponseID & Chr(34) &
" checked>" & sRSResponseText
              Else
                m_oResponse.Write "<INPUT TYPE=""checkbox"" NAME=" & Chr(34) &
"QID" & lQuestionID & Chr(34) & " VALUE=" & Chr(34) & lRSResponseID & Chr(34) &
">" & sRSResponseText
              End If
              m_oResponse.Write "</td></tr>" & vbCrLf

              oQuesRespRS.MoveNext

            Wend 'While POSSIBLE RESPONSES




        ElseIf iTypeID = 5 Then 'Short Text
          'Get the user responses recordset.
          Set oUserRespRS = oQuestionRS("userResponses").Value

            sAnswerValue = ""
            If Not oUserRespRS.EOF Then
              sAnswerValue = (oUserRespRS("tURvalue") & "") 'or tURValue
            End If
            m_oResponse.Write "<tr><td>  </td><td
class=""questionres"">"
            m_oResponse.Write "<INPUT TYPE=""TEXT"" NAME=" & Chr(34) & "QID" &
lQuestionID & Chr(34) & " VALUE=" & Chr(34) & sAnswerValue & Chr(34) & ">"
            m_oResponse.Write "</td></tr>" & vbCrLf


        ElseIf iTypeID = 6 Then
          Set oUserTextRespRS = oQuestionRS("userTextResponses").Value

          sAnswerValue = ""
          If Not oUserTextRespRS.EOF Then
            sAnswerValue = (oUserTextRespRS("tTDtext") & "")
          End If


            m_oResponse.Write "<tr><td>  </td><td
class=""questionres"">"
            'Response.Write "<TEXTAREA NAME=" & Chr(34) & "QID" & lQuestionID &
Chr(34) & ">" & oQuesRespRS("value") & "</textarea><br>"
            m_oResponse.Write "<TEXTAREA NAME=" & Chr(34) & "QID" & lQuestionID
& Chr(34) & " rows=4 cols=75>" & sAnswerValue & "</textarea>"
```

```
            m_oResponse.Write "</td></tr>" & vbCrLf
        Else
            'Do nothing IS is not set up or the TypeID is invalid.
            m_oResponse.Write "ID Invalid"
        End If 'If iTypeID<>5 && iTypeID<>6

        oQuestionRS.MoveNext
        'Move to the next question.
        Wend    'While NOT oQuestionRS.Eof


        m_oResponse.Write "<br>"
        m_oResponse.Write "<input type=""hidden"" name=""sid"" value=" &
Chr(34) & m_oSession.SessionID & Chr(34) & ">" & vbCrLf
        m_oResponse.Write "<input type=""hidden"" name=""bid"" value=" &
Chr(34) & m_lBusID & Chr(34) & ">" & vbCrLf

        m_oResponse.Write "<tr><td><input type=""submit"" value=""Rate It""
id=1 name=1></td>"
        m_oResponse.Write "<td><input type=""reset"" value=""Clear Form"" id=1
name=1></td></tr>"

        m_oResponse.Write "</table>" 'End the table for this question
        m_oResponse.Write "</form>" & vbCrLf
    Else
        m_oResponse.Write "<center><font color=""#990000"">Invalid Business
ID</font></center>"
    End If



If oQuestionRS.State = adStateOpen Then
  oQuestionRS.Close
End If

If oQuesRespRS.State = adStateOpen Then
  oQuesRespRS.Close
End If

If oUserRespRS.State = adStateOpen Then
  oUserRespRS.Close
End If

If oShapeConn.State = adStateOpen Then
  oShapeConn.Close
End If


    'oShapeConn.Close
    'oQuestionRS.Close
    'oQuesRespRS.Close

    Set oShapeConn = Nothing
    Set oQuestionRS = Nothing
    Set oQuesRespRS = Nothing
    Set oUserRespRS = Nothing



End Sub

'/*****************************************************************************
'/P U B L I C   F U N C T I O N   S E C T I O N
```

```
'/
'/B E G I N
'/*************************************************************************

Public Function insertUserRatingIntoDatabase() As Variant
'PURPOSE: This subroutine will take posted data and enter it into
'         the database for a particular user and business.
'
'IN: Properities set by user/code
'OUT: Properity bFormComplete
'     TRUE=All form data present and posted.
'     FALSE=Form data not all present, db transaction rolled back.

Dim oQuesConn As ADODB.Connection, oInsertConn As ADODB.Connection
Dim oQuesCommand As ADODB.Command, oInsertCmd As ADODB.Command
Dim oQuestionList As ADODB.Recordset
Dim iRec As Long 'Used to hold the return of Rows Affected

'Holds the count of ratings for this distinct userid and distinct business id
'should be either 0 or 1
Dim lRateCount As Long, lBusUserID As Long

'Holds a true/false value if a user has rated the business before.
Dim bRatedBefore As Boolean
'Flag to say that the form is incomplete, this starts off as false
'changed when an answer is not found for a question
Dim bFormIncomplete As Boolean

'These hold the values for the stars coins and thumbs from the form
Dim iStar As Single, iCoin As Single
Dim iThumb As Integer

'This holds the names in the multiple selection of check boxes.
Dim vChkResponseValue As Variant

Dim bQuestionHasAnswers As Boolean

Dim iReturn As Integer

Dim iResponseType As Integer  'Is the response type from the question recordset
Dim lResponseID As Long
Dim lQuestionID As Long
Dim sResponseText As String
Dim iQuestionResponseType As Integer
Dim sQuestionFormName As String 'Holds the Form Name for the current question
to get the users answer.
Dim intCounter As Integer

Dim vList As Variant


'Some initialization here...
bRatedBefore = False
bFormIncomplete = False


If m_bJavaScriptDisplay = True Then
  iReturn = OpenProgressWindow("<center>Posting Your Results</center>")
End If


'Now begin to insert all the users responses into the database.
'If an answer is not present then ROLLBACK the transaction and call DispForm.
```

```
Set oQuesConn = CreateObject("ADODB.Connection")
Set oQuesCommand = CreateObject("ADODB.Command")
Set oQuestionList = CreateObject("ADODB.Recordset")

Set oInsertConn = CreateObject("ADODB.Connection")
Set oInsertCmd = CreateObject("ADODB.Command")

'CREATE PROCEDURE sp_GetQuestionsByBusinessID(@busid int)

'Generate the SQL statement to get all the questions and responses for this
category.
oQuesConn.Open m_sDatabaseConnectString

If m_bJavaScriptDisplay = True Then
  UpdateStats "<center>Posting Your Results</center>", "<center>Getting
Question ID's</center>", "<center>For Validation</center>", "", "", ""
End If
''oQuesCommand.ActiveConnection = oQuesConn
''oQuesCommand.CommandType = 4 'Tell it is stored procedure(4)
''oQuesCommand.CommandText = "sp_GetQuestionsByBusinessID"
''oQuesCommand.Parameters.Append( oQuesCommand.CreateParameter("@busid", 3,
&H0001, 4, iBusID))
''oQuesCommand.Parameters("@busid")
''Set oQuestionList = oQuesCommand.Execute
'Open the recordset with adOpenStatic=3 and adLockReadOnly=1
oQuestionList.Open "sp_GetQuestionsByBusinessID(" & m_lBusID & ")", oQuesConn,
3, 1
If m_bDebug = True Then
    m_oResponse.Write "Business ID:" & m_lBusID & "<br>"
    m_oResponse.Write "User ID:" & m_lUserID & "<br>"
    m_oResponse.Write "BusUserID:" & m_lBusToUserID & "<br>"
End If
'Response.Write "Begining question info"
'While not oQuestionList.EOF
'  Response.Write "Question ID:"
'  Response.Write oQuestionList("questionid")
'  Response.Write "<br>"
'  Response.Write "TypeID:"
'  Response.Write oQuestionList("typeid")
'  Response.Write "<br><br>"
'  oQuestionList.MoveNext
'Wend
'Response.Write "End of question info"
'Response.Flush


If m_bJavaScriptDisplay = True Then
  UpdateStats "<center>Posting Your Results</center>", "Checking to make sure
everything is there...", "", "", "", ""
End If
'Now open a connection to the table that allows me to do the inserts
oInsertConn.Open m_sDatabaseConnectString
oInsertConn.BeginTrans
oInsertCmd.ActiveConnection = oInsertConn

'Here is what happens first
'1.) Check to see if the user has already rated the business or not
'    and get a user-to-business rating id either existing one
'    or create a new one.

'Here is what happens if they already rated
'1.) CREATE PROCEDURE sp_DeleteUserResponses(@bususerid int)
```

```
'2.) CREATE PROCEDURE sp_DeleteStarCoinThumb(@bususerid int)
'3.) CREATE PROCEDURE sp_DeleteTextDescription(@bususerid int)

'Here are the order of the store procedure calls.
'1.) CREATE PROCEDURE sp_InsertNewRatingGetID(@busid int,
'           @userid int, @ip varchar(50), @bususerid int OUTPUT)
'2.) CREATE PROCEDURE sp_InsertStarCoinThumb(@stars real, @coins real,
'           @thumbs int, @bususerid int, @userid int, @busid int)
'3.) CREATE PROCEDURE sp_InsertUserResponses(@bususerid int, @userid int,
'           @busid int, @questionid int,    @responseid int, @text
varchar(255),
'           @value varchar(255))
'4.) CREATE PROCEDURE sp_InsertTextDescription(@bususerid int, @userid int,
'           @busid int, @questionid int, @text text)

'/-----------------------------------------------
'CHECK TO SEE IF USER HAS RATED BUSINESS BEFORE
'This will take the userID, businessID, and the UsersIP address
'it will check to see if the user has rated the business before
'if so it will return the user-to-business rating id and @answer =>1
'if not it will create a new user-to-business rating id and return it to the
'code with @answer=0
'CREATE PROCEDURE sp_HasUserRatedBusinessWithID_CreateNoExist(@userid int,
'                  @busid int, @ip varchar(50), @answer int OUTPUT,
'                  @bususrid int OUTPUT)
oInsertCmd.CommandText = "sp_HasUserRatedBusinessWithID_CreateNoExist"
oInsertCmd.CommandType = 4 'Stored procedure
oInsertCmd.Parameters("@userid") = m_lUserID
oInsertCmd.Parameters("@busid") = m_lBusID
oInsertCmd.Parameters("@ip") = CStr(m_oRequest.ServerVariables("REMOTE_ADDR") &
"")
oInsertCmd.Execute


lRateCount = oInsertCmd("@answer")  'Get the count of how many ratings were
found.
lBusUserID = oInsertCmd("@bususrid") 'Get the business-to-user rating id.
'add code here to set the property...
m_lBusToUserID = lBusUserID

If lRateCount >= 1 Then
  bRatedBefore = True
Else
  bRatedBefore = False
End If

If m_bDebug = True Then
    m_oResponse.Write "Checked for user rating existance: lRateCount:" &
lRateCount & ":: iBusUserID:" & m_lBusToUserID & "<br>" & vbCrLf
    m_oResponse.Flush
End If

'/-----------------------------------------------
'IF USER HAS RATED BEFORE DELETE ALL ANSWERS.
If bRatedBefore = True Then
  'Issue all deletes here under oInsertConn

  If m_bJavaScriptDisplay = True Then
    UpdateStats "<center>Posting Your Results</center>", "<center>Your Re-
Rating This Business</center>", "Clearing out your old answers", "", "", ""
  End If

    oInsertCmd.CommandText = "sp_DeleteStarCoinThumb"
```

```
    oInsertCmd.CommandType = 4  'Tell it that it is a stored procedure.
    oInsertCmd.Parameters("@bususerid") = m_lBusToUserID
    oInsertCmd.Execute iRec

    If m_bDebug Then
        m_oResponse.Write "Deleting all the users previous data:" & iRec &
"<br>"
    End If

    oInsertCmd.CommandText = "sp_DeleteUserResponses"
    oInsertCmd.CommandType = 4
    oInsertCmd.Parameters("@bususerid") = m_lBusToUserID
    oInsertCmd.Execute iRec

    If m_bDebug Then
        m_oResponse.Write "Deleting all the users previous data:" & iRec &
"<br>"
    End If

    oInsertCmd.CommandText = "sp_DeleteTextDescription"
    oInsertCmd.CommandType = 4
    oInsertCmd.Parameters("@bususerid") = m_lBusToUserID
    oInsertCmd.Execute iRec

    If m_bDebug Then
        m_oResponse.Write "Deleting all the users previous data:" & iRec &
"<br>"
        m_oResponse.Flush
    End If

End If

'/-----------------------------------------------
'Insert the Coin, Star and Thumb info.
'-Call stored procedure to insert these values here...
'CREATE PROCEDURE sp_InsertStarCoinThumb(@stars real, @coins real,
'                   @thumbs int, @bususerid int, @userid int, @busid int)


iStar = CSng(m_oRequest.Form("stars"))
iCoin = CSng(m_oRequest.Form("coins"))
iThumb = CInt(m_oRequest.Form("goback") + 0)

  oInsertCmd.CommandText = "sp_InsertStarCoinThumb"
  oInsertCmd.CommandType = 4
  oInsertCmd.Parameters("@stars") = iStar
  oInsertCmd.Parameters("@coins") = iCoin
  oInsertCmd.Parameters("@thumbs") = iThumb
  oInsertCmd.Parameters("@bususerid") = m_lBusToUserID
  oInsertCmd.Parameters("@userid") = m_lUserID
  oInsertCmd.Parameters("@busid") = m_lBusID
  oInsertCmd.Execute iRec

If m_bDebug Then
    m_oResponse.Write "Inserted stars coins and thumbs: " & iRec
    m_oResponse.Flush
End If

'/-----------------------------------------------
'Now begin to insert all the responses here...
'-Loop through all questions and put each one in the database
'-if response is not there abort the transaction and redisplay
'-a data entry page.
```

```
intCounter = 0

While Not oQuestionList.EOF And bFormIncomplete = False
  intCounter = intCounter + 1

  iResponseType = 0
  lResponseID = 0
  sResponseText = ""

  sQuestionFormName = "QID" & oQuestionList("questionid")
  lQuestionID = oQuestionList("questionid")
  iQuestionResponseType = oQuestionList("typeid")

  If m_bDebug Then
    m_oResponse.Write "sQuestionFormName:" & sQuestionFormName & "<br>"
    m_oResponse.Write "lQuestionID:" & lQuestionID & "<br>"
    m_oResponse.Write "iQuestionResponseType:" & iQuestionResponseType & "<br>"
  End If

  If m_bJavaScriptDisplay = True Then
    UpdateStats "<center>Posting Your Results</center>",
"<center>Processing</center>", "Question No: " & intCounter, "", "", ""
  End If

  If iQuestionResponseType = 1 Then 'Unassigned
        'Don't do anything

  ElseIf iQuestionResponseType = 2 Then 'Radio Button
        lResponseID = CInt(m_oRequest.Form(sQuestionFormName) + 0)

        If lResponseID = 0 Then
            bFormIncomplete = True
        Else
            'sp_InsertUserResponses(@bususerid int, @userid int,
            '     @busid int, @questionid int,  @responseid int, @text
varchar(255),
        '     @value varchar(255))
            oInsertCmd.CommandText = "sp_InsertUserResponses"
            oInsertCmd.CommandType = 4  'Tell it that it is a stored procedure.
            oInsertCmd.Parameters("@bususerid") = m_lBusToUserID
            oInsertCmd.Parameters("@userid") = m_lUserID
            oInsertCmd.Parameters("@busid") = m_lBusID
            oInsertCmd.Parameters("@questionid") = lQuestionID
            oInsertCmd.Parameters("@responseid") = lResponseID
            oInsertCmd.Parameters("@text") = ""
            oInsertCmd.Parameters("@value") = ""

            oInsertCmd.Execute
        End If  'End Of VALID Radio Selections

  ElseIf iQuestionResponseType = 3 Then 'Select Box - Single
        lResponseID = CInt(m_oRequest.Form(sQuestionFormName) + 0)

        If lResponseID = 0 Then
            bFormIncomplete = True
        Else
            'sp_InsertUserResponses(@bususerid int, @userid int,
            '     @busid int, @questionid int,  @responseid int, @text
varchar(255),
        '     @value varchar(255))
            oInsertCmd.CommandText = "sp_InsertUserResponses"
            oInsertCmd.CommandType = 4  'Tell it that it is a stored procedure.
            oInsertCmd.Parameters("@bususerid") = m_lBusToUserID
```

```
            oInsertCmd.Parameters("@userid") = m_lUserID
            oInsertCmd.Parameters("@busid") = m_lBusID
            oInsertCmd.Parameters("@questionid") = lQuestionID
            oInsertCmd.Parameters("@responseid") = lResponseID
            oInsertCmd.Parameters("@text") = ""
            oInsertCmd.Parameters("@value") = ""

            oInsertCmd.Execute
        End If  'End Of VALID Selections

  ElseIf iQuestionResponseType = 4 Then 'Checkbox - Multiple
        Set vList = m_oRequest.Form(sQuestionFormName)
        'lResponseID = CLng(m_oRequest.Form(sQuestionFormName) + 0)

        bQuestionHasAnswers = False


        For Each vChkResponseValue In m_oRequest.Form(sQuestionFormName)
            bQuestionHasAnswers = True

            'sp_InsertUserResponses(@bususerid int, @userid int,
            '     @busid int, @questionid int,  @responseid int, @text
varchar(255),
        '     @value varchar(255))
            oInsertCmd.CommandText = "sp_InsertUserResponses"
            oInsertCmd.CommandType = 4  'Tell it that it is a stored procedure.
            oInsertCmd.Parameters("@bususerid") = m_lBusToUserID
            oInsertCmd.Parameters("@userid") = m_lUserID
            oInsertCmd.Parameters("@busid") = m_lBusID
            oInsertCmd.Parameters("@questionid") = lQuestionID
            oInsertCmd.Parameters("@responseid") = CLng(vChkResponseValue)
            oInsertCmd.Parameters("@text") = ""
            oInsertCmd.Parameters("@value") = ""

            oInsertCmd.Execute
        Next

        If bQuestionHasAnswers = False Then
            bFormIncomplete = True
        Else
            bFormIncomplete = False
        End If 'END OF if Question Had Answers


  ElseIf iQuestionResponseType = 5 Then 'Short Text
        sResponseText = ""
        sResponseText = CStr(m_oRequest.Form(sQuestionFormName) & "")
        'Take out vbCrLf, Carriage Returns, Line Feeds, single and double
quotes.
        'sResponseText = StripSpecialChars(sResponseText)

        If Len(sResponseText) < 1 Then
            bFormIncomplete = True
        Else
                  'sp_InsertUserResponses(@bususerid int, @userid int,
            '     @busid int, @questionid int,  @responseid int, @text
varchar(255),
        '     @value varchar(255))
            oInsertCmd.CommandText = "sp_InsertUserResponses"
            oInsertCmd.CommandType = 4  'Tell it that it is a stored procedure.
            oInsertCmd.Parameters("@bususerid") = m_lBusToUserID
            oInsertCmd.Parameters("@userid") = m_lUserID
            oInsertCmd.Parameters("@busid") = m_lBusID
```

```
            oInsertCmd.Parameters("@questionid") = lQuestionID
            oInsertCmd.Parameters("@responseid") = 0
            oInsertCmd.Parameters("@text") = ""
            oInsertCmd.Parameters("@value") = sResponseText

            oInsertCmd.Execute
        End If  'End Of VALID Short Text Response

  ElseIf iQuestionResponseType = 6 Then 'Long Text
        sResponseText = ""
        sResponseText = CStr(m_oRequest.Form(sQuestionFormName) & "")

        'Take out vbCrLf, Carriage Returns, Line Feeds, single and double
quotes.
        'sResponseText = StripSpecialChars(sResponseText)
        If sResponseText = "" Then
            bFormIncomplete = True
        Else
'sp_InsertTextDescription(@bususerid int, @userid int,
'           @busid int, @questionid int, @text text)
            oInsertCmd.CommandText = "sp_InsertTextDescription"
            oInsertCmd.CommandType = 4  'Tell it that it is a stored procedure.
            oInsertCmd.Parameters("@bususerid") = m_lBusToUserID
            oInsertCmd.Parameters("@userid") = m_lUserID
            oInsertCmd.Parameters("@busid") = m_lBusID
            oInsertCmd.Parameters("@questionid") = lQuestionID
            oInsertCmd.Parameters("@text") = sResponseText
            oInsertCmd.Execute
        End If 'End Of VALID Text Resposne

  End If 'End of If QuestionTYPE = 1,2,3,4,5,6

  oQuestionList.MoveNext 'Jump to the next question.

Wend    'While Not Questions.EOF and bFormIncomplete=False



If oQuestionList.State = adStateOpen Then
  oQuestionList.Close
End If

If oQuesConn.State = adStateOpen Then
  oQuesConn.Close
End If


If bFormIncomplete = True Then
    'Then Roll the transaction back and close all the recordsets.

    oInsertConn.RollbackTrans
    oInsertConn.Close
    If m_bJavaScriptDisplay = True Then
      UpdateStats "<center>Posting Your Results</center>", "<center>Form
Incomplete</center>", " ", "<center>You must answer all the
questions</center>", "", ""
    End If

Else
    'Looks like the form had all we needed...
    'Lets post it and tell them that they have been updated.
    oInsertConn.CommitTrans
    oInsertConn.Close
```

```
    If m_bJavaScriptDisplay = True Then
      UpdateStats "<center>Posting Your Results</center>", "<center>All
Done</center>", "", "", "", ""
    End If


End If

If oQuestionList.State = adStateOpen Then
  oQuestionList.Close
End If


'Clean up Time
Set oQuesConn = Nothing
Set oQuestionList = Nothing
Set oInsertCmd = Nothing
Set oInsertConn = Nothing

If m_bJavaScriptDisplay = True Then
  iReturn = CloseProgressWindow()
End If


If bFormIncomplete = True Then
  insertUserRatingIntoDatabase = False
Else
  insertUserRatingIntoDatabase = True
End If


End Function


Public Function HasUserRatedBusinessBefore() As Variant
'PURPOSE: Will check to see if a user has rated a particular business before.
'         Return Values: TRUE - User has rated business before.
'                        FALSE - User has not rated business before.
'
'IN: Properities set by user/code.
'OUT: Return value from function

Dim oConn As ADODB.Connection
Dim oUserCmd As ADODB.Command

Dim iRateCount As Integer
Dim lBusUsrID As Long


Set oConn = CreateObject("ADODB.Connection")
Set oUserCmd = CreateObject("ADODB.Command")

oConn.Open m_sDatabaseConnectString
oUserCmd.ActiveConnection = oConn
oUserCmd.CommandType = adCmdStoredProc   'Tell it that this is a stored
procedure
oUserCmd.CommandText = "sp_HasUserRatedBusinessWithID"
'oUserCmd.Parameters("@userid") = iUser
'oUserCmd.Parameters("@busid") = 2

oUserCmd.Parameters.Append oUserCmd.CreateParameter("@userid", adInteger,
adParamInput, 4, m_lUserID)
```

```
oUserCmd.Parameters.Append oUserCmd.CreateParameter("@busid", adInteger,
adParamInput, 4, m_lBusID)
oUserCmd.Parameters.Append oUserCmd.CreateParameter("@answer", adInteger,
adParamOutput, 0, 0)
oUserCmd.Parameters.Append oUserCmd.CreateParameter("@bususrid", adInteger,
adParamOutput, 0, 0)

oUserCmd.Execute

iRateCount = oUserCmd("@answer")    'Grab the value so we can use it later
lBusUsrID = oUserCmd("@bususrid")

m_lBusToUserID = lBusUsrID

oConn.Close
Set oUserCmd = Nothing
Set oConn = Nothing

HasUserRatedBusinessBefore = iRateCount


End Function


Public Function isValidBusiness() As Variant
'PURPOSE: Function will determine if a business entry exists for a
'         specified business id.  Additionally, if the business does
'         exist then it will then populate properities for the
'         business name, subcategory id, and business name.
'IN: Properities set by user/code.
'OUT: Return value: TRUE=The business exists,
'                        properities set with additional info.
'                   FALSE=The business does not exist,
'                        properities are empty strings.

Dim oConn As ADODB.Connection
Dim oBusinessInfoCmd As ADODB.Command

Dim iIsValid As Integer
Dim lSubCatID As Long
Dim sBusinessName As String


Set oConn = CreateObject("ADODB.Connection")
Set oBusinessInfoCmd = CreateObject("ADODB.Command")

oConn.Open m_sDatabaseConnectString

oBusinessInfoCmd.ActiveConnection = oConn
oBusinessInfoCmd.CommandType = 4 'Stored procedure
oBusinessInfoCmd.CommandText = "sp_ValidateBusinessWithInfo"
oBusinessInfoCmd.Parameters.Append oBusinessInfoCmd.CreateParameter("@busid",
adInteger, adParamInput, 4, m_lBusID)
oBusinessInfoCmd.Parameters.Append oBusinessInfoCmd.CreateParameter("@isvalid",
adInteger, adParamReturnValue, 0, 0)
oBusinessInfoCmd.Parameters.Append
oBusinessInfoCmd.CreateParameter("@subcatid", adInteger, adParamOutput, 0, 0)
oBusinessInfoCmd.Parameters.Append oBusinessInfoCmd.CreateParameter("@busname",
adVarChar, adParamOutput, 50, "")
'oBusinessInfoCmd.Parameters.Append oBusinessInfoCmd.CreateParameter("@busid",
adInteger, adParamInput, 4, m_lBusID)
'oBusinessInfoCmd.Parameters.Append
oBusinessInfoCmd.CreateParameter("@isvalid", adInteger,adparamou &H4, 0, 0)
```

```
'oBusinessInfoCmd.Parameters.Append
oBusinessInfoCmd.CreateParameter("@subcatid", 3, &H2, 0, 0)
'oBusinessInfoCmd.Parameters.Append
oBusinessInfoCmd.CreateParameter("@busname", 200, &H2, 50, "")
oBusinessInfoCmd.Execute

iIsValid = oBusinessInfoCmd("@isvalid")
lSubCatID = oBusinessInfoCmd("@subcatid")
sBusinessName = oBusinessInfoCmd("@busname")

m_iIsValidBusiness = iIsValid   'Set the isValidBusiness Property.
m_lSubCatID = lSubCatID 'Set the Sub Category ID Property.
m_sBusinessName = sBusinessName 'Set the BusinessName Property.


If iIsValid = 1 Then
    isValidBusiness = True
Else
    isValidBusiness = False
End If


End Function


Private Function OpenProgressWindow(ByVal strHeading As String)
'PURPOSE: This function will write JavaScript out to the browser
'         so that a status/information window opens up.
'         It will give the code-write a heading line and five
'         text lines to work with, each with their own assigned
'         stylesheet classes so that each line is more customizable.
'IN: strHeading=A heading to be displayed right away.
'OUT: 1 for complete sucessfully.



    m_oResponse.Write "<script language=""javascript"">" & vbCrLf
    m_oResponse.Write "var whnd;" & vbCrLf
    m_oResponse.Write "var pdoc;" & vbCrLf
    m_oResponse.Write "whnd =
window.open('','progress','directories=no,height=200,width=400,resizable=no,sta
tus=no,toolbar=no;',false);" & vbCrLf
    m_oResponse.Write "pdoc = whnd.document;" & vbCrLf
    m_oResponse.Write "pdoc.writeln('<html>');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('<head>');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('<link title=""normal"" rel=""stylesheet""
href=""globalstyle.css"" type=""text/css"">');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</head>');" & vbCrLf

    m_oResponse.Write "pdoc.writeln('<body bgcolor=""#FFFFFF"">');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('<table align=center border=0 cellspacing=1
width=300px>');" & vbCrLf

    'Status box heading...
    'm_oResponse.Write "pdoc.writeln('<tr><td colspan=2 align=center
bgcolor=#c0c0c0>" & strHeading & "</td></tr>');"

    m_oResponse.Write "pdoc.writeln('<tr class=""jspuheadingtr"">');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('<td class=""jspuheadingtd"">');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('<div id=""jspuheading"">');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('" & strHeading & "');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</td></tr>');" & vbCrLf
```

```
    m_oResponse.Write "pdoc.writeln('<tr class=""jsputrline1"">');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('<td class=""jsputdline1"" align=left>');"
& vbCrLf
    m_oResponse.Write "pdoc.writeln('<div id=""line1"">');" & vbCrLf
    m_oResponse.Write "pdoc.writeln(' ');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</div>');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</td></tr>');" & vbCrLf

    m_oResponse.Write "pdoc.writeln('<tr class=""jsputrline2"">');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('<td class=""jsputdline2"" align=left>');"
& vbCrLf
    m_oResponse.Write "pdoc.writeln('<div id=""line2""');" & vbCrLf
    m_oResponse.Write "pdoc.writeln(' ');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</div>');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</td></tr>');" & vbCrLf

    m_oResponse.Write "pdoc.writeln('<tr class=""jsputrline3"">');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('<td class=""jsputdline3"" align=left>');"
& vbCrLf
    m_oResponse.Write "pdoc.writeln('<div id=""line3""');" & vbCrLf
    m_oResponse.Write "pdoc.writeln(' ');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</div>');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</td></tr>');" & vbCrLf

    m_oResponse.Write "pdoc.writeln('<tr class=""jsputrline4"">');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('<td class=""jsputdline4"" align=left>');"
& vbCrLf
    m_oResponse.Write "pdoc.writeln('<div id=""line4""');" & vbCrLf
    m_oResponse.Write "pdoc.writeln(' ');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</div>');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</td></tr>');" & vbCrLf

    m_oResponse.Write "pdoc.writeln('<tr class=""jsputrline5"">');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('<td class=""jsputdline5"" align=left>');"
& vbCrLf
    m_oResponse.Write "pdoc.writeln('<div id=""line5""');" & vbCrLf
    m_oResponse.Write "pdoc.writeln(' ');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</div>');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</td></tr>');" & vbCrLf

    m_oResponse.Write "pdoc.writeln('</table>');" & vbCrLf


    m_oResponse.Write "pdoc.writeln('<br><br>');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</body>');" & vbCrLf
    m_oResponse.Write "pdoc.writeln('</html>');" & vbCrLf
    m_oResponse.Write "</script>" & vbCrLf

m_oResponse.Flush

    OpenProgressWindow = 1

End Function

Private Function CloseProgressWindow()
'PURPOSE: This function will write JavaScript code
'         out to the browser to close the PopUp Window
'IN: None
'OUT: 1=Successfully completed
'
'Closes the progress window.
    m_oResponse.Write "<script>"
    m_oResponse.Write "whnd.close()" & vbCrLf
```

```
    m_oResponse.Write "</script>"

    m_oResponse.Flush

    CloseProgressWindow = 1

End Function

Private Sub UpdateStats(ByRef sHeading As String, _
                        ByRef sLine1 As String, _
                        ByRef sLine2 As String, _
                        ByRef sLine3 As String, _
                        ByRef sLine4 As String, _
                        ByRef sLine5 As String)
'PURPOSE: This subroutine will write out JavaScript code
'         to the browser to update the information in the
'         JavaScript popup window.
'IN: None
'OUT: None

'This will update the status window to let the user know how much
'of the file has been created.
Dim sOutput As String
sOutput = "<script>" & vbCrLf

If sHeading <> "" Then
    sOutput = sOutput & "pdoc.all.jspuheading.innerHTML=" & Chr(34) & sHeading
& Chr(34) & ";" & vbCrLf
End If

If sLine1 <> "" Then
    sOutput = sOutput & "pdoc.all.line1.innerHTML=" & Chr(34) & sLine1 &
Chr(34) & ";" & vbCrLf
End If

If sLine2 <> "" Then
    sOutput = sOutput & "pdoc.all.line2.innerHTML=" & Chr(34) & sLine2 &
Chr(34) & ";" & vbCrLf
End If

If sLine3 <> "" Then
    sOutput = sOutput & "pdoc.all.line3.innerHTML=" & Chr(34) & sLine3 &
Chr(34) & ";" & vbCrLf
End If

If sLine4 <> "" Then
    sOutput = sOutput & "pdoc.all.line4.innerHTML=" & Chr(34) & sLine4 &
Chr(34) & ";" & vbCrLf
End If

If sLine5 <> "" Then
    sOutput = sOutput & "pdoc.all.line5.innerHTML=" & Chr(34) & sLine5 &
Chr(34) & ";" & vbCrLf
End If

sOutput = sOutput & "</script>" & vbCrLf

m_oResponse.Write sOutput

m_oResponse.Flush

End Sub
```

```
Private Function StripSpecialChars(sIn As String) As String


    If Not VarType(sIn) = vbString Then StripSpecialChars = sIn: Exit Function

    If InStr(1, sIn, vbCrLf, vbTextCompare) Then
      sIn = Replace(sIn, vbCrLf, " ")
    End If

    If InStr(1, sIn, Chr(13), vbTextCompare) Then
        sIn = Replace(sIn, Chr(13), "")
    End If

    If InStr(1, sIn, Chr(10), vbTextCompare) Then
        sIn = Replace(sIn, Chr(10), "")
    End If

    If InStr(1, sIn, Chr(34), vbTextCompare) Then
        sIn = Replace(sIn, Chr(34), (Chr(34) & Chr(34)))
    End If

    If InStr(1, sIn, "'", vbTextCompare) Then
      sIn = Replace(sIn, "'", "''")
    End If


    StripSpecialChars = sIn
End Function


Public Property Get sErrDesc() As Variant
    sErrDesc = m_sErrorDescription
End Property


Public Property Get lErrCode() As Variant
    lErrCode = m_lErrorCode
End Property

Public Property Get sDBConnectString() As Variant
    sDBConnectString = m_sDatabaseConnectString
End Property

Public Property Let sDBConnectString(ByVal vNewValue As Variant)
    If CStr(vNewValue & "") = "" Then
        m_sDatabaseConnectString = ""
    Else
        m_sDatabaseConnectString = CStr(vNewValue & "")
    End If
End Property

Public Property Get bJavaScriptPopUp() As Variant
    bJavaScriptPopUp = m_bJavaScriptDisplay
End Property

Public Property Let bJavaScriptPopUp(ByVal vNewValue As Variant)
    If vNewValue = Null Then
        m_bJavaScriptDisplay = False
    Else
        m_bJavaScriptDisplay = CBool(vNewValue)
    End If
End Property
```

```
Public Property Get bDebugInfo() As Variant
    bDebugInfo = m_bDebug
End Property

Public Property Let bDebugInfo(ByVal vNewValue As Variant)
    If vNewValue = Null Then
        m_bDebug = False
    Else
        m_bDebug = CBool(vNewValue)
    End If
End Property

Public Property Get lBusinessID() As Variant
    lBusinessID = m_lBusID
End Property

Public Property Let lBusinessID(ByVal vNewValue As Variant)
    m_lBusID = CLng(vNewValue + 0)
End Property

Public Property Get lUserID() As Variant
    lUserID = m_lUserID
End Property

Public Property Let lUserID(ByVal vNewValue As Variant)
    m_lUserID = CLng(vNewValue)
End Property

Public Property Get lSubCategoryID() As Variant
    lSubCategoryID = m_lSubCatID
End Property

Public Property Let lSubCategoryID(ByVal vNewValue As Variant)
    m_lSubCatID = CLng(vNewValue + 0)
End Property

Public Property Get lBusinessUserID() As Variant
  lBusinessUserID = m_lBusToUserID
End Property

Public Property Let lBusinessUserID(ByVal vNewValue As Variant)
    m_lBusToUserID = CLng(vNewValue + 0)
End Property

Public Property Get sValidationSuccessPage() As Variant
    sValidationSuccessPage = m_sValidationSuccessPage
End Property

Public Property Let sValidationSuccessPage(ByVal vNewValue As Variant)
    m_sValidationSuccessPage = CStr(vNewValue & "")
End Property

Public Property Get sValidationFailPage() As Variant
    sValidationFailPage = m_sValidationFailPage
End Property

Public Property Let sValidationFailPage(ByVal vNewValue As Variant)
    m_sValidationFailPage = CStr(vNewValue & "")
End Property

Public Property Get sBusinessName() As Variant
  sBusinessName = m_sBusinessName
```

```
End Property
```

References

1. Ambethkar, Venkatraman. "Using SQL Server 7 Web Assistant to Improve performance of Asp Pages, Part 2". http://www.4guysfromrolla.com/webtech/062200-1.2.shtml. June 22, 2000.

2. Ambethkar, Venkatraman. "Using SQL Server 7 Web Assistant to Improve performance of Asp Pages". http://www.4guysfromrolla.com/webtech/062200-1.shtml. June 22, 1000.

3. Arjona, Ramon. "Creating Hierarchical Recordsets With the Data Shape Provider – Part I". http://www.asptoday.com/content/articles/20000904.asp. September 4, 2000.

4. Beamer, Greg. "Moving From ASP to VB COM". http://www.asptoday.com/content/articles/20000121.asp. January 21, 2000.

5. Benedetti, Gaddo F. "Dynamically Generated and Validated Forms". http://www.15seconds.com/issue/000225.htm. February 25, 2000.

6. Better Business Bureau. "1998 Annual Inquiry and Complaint Summary for U.S. Better Business Bureaus". http://www.bbb.org/alerts/20000201.asp. February 1, 2000.

7. Better Business Bureau. "Business and Charity Reports". http://www.bbb.org/reports/index.asp. March 2000.

8. Better Business Bureau. "Complaint Assistance". http://www.bbb.org/helpdesk/complaint.asp. March 2000.

9. Better Business Bureau. "Consumer Complaint Form". http://www.bbb.org/complaints/consumerform.asp. March 2000.

10. Better Business Bureau. "Dispute Resolution (DRD) Division". http://www.bbb.org/complaints/aboutResolution.asp. March 2000.

11. Blackman, Barry. "Simplify Your Life with SQL 7 Subqueries". http://www.asptoday.com/articles/20000731.htm. July 31, 2000.

12. Blexrud, Chris. "The Idispatch Interface". http://www.asptoday.com/articles/19990917.htm. September 17, 2999.

13. Bollaert, Jodi. "Talk to me, Baby". WebTechniques. December 2000. Volume 5, Issue 12. pp 27-29.

14. Cashel, Jim. "Online Communities and The Law – An Interview with John Delaney". http://www.onlinecommunityreport.com/features/law/. February 2, 2000.

15. Cherico, Holly. Public Relations, Better Business Bureau. Telephone Interview. March 6, 2000.

16. CNET Builder.Com. "Contracts for every occasion". http://home.cnet.com/webbuilding/0-3885-8-4500031-1.html?tag=st.bl.3880.pro_h.3885-8-4500031. January 23, 2001.

17. Cunningham, Holly. "Iconic you: custom bookmark icons". http://home.cnet.com/webbuilding/0-7690-8-4580011-1.html?tag=st.bl.3880.pro_h.7690-8-4580011-1. February 2, 2001.

18. Dean, Doug. "How To Utilize Database Transactions Within a VB Component From an ASP Page". http://www.4guysfromrolla.com/webtech/081800-1.shtml. August 18, 2000.

19. Durlacher Research. "Creating Community Online". http://www.durlacher.co.uk/research/resrepdetail21.asp. January 1 2000.

20. Easton, Marc. "Migrating COM components from VB to C++ Using ATL". http://www.asptoday.com/articles/20000705.htm. July 5, 2000.

21. Forum One. "Three Steps To Building Your Own Web Forum" http://www.forumone.com/build.htm. November 1999.

22. Francis, Fedorov, Harrison, Homer, Murphy, Sussman, Smith and Wood. *Active Server Pages 2.0 Professional*. Wrox Press. Copyright 1999.

23. Fraser, Laura. "The Experience of Being Suddenly Rich". http://www.thestandard.com/article/display/0,1151,7812,00.html. November 29, 2000.

24. Gill, Darren. "DIY VB Components for ASP". http://www.asptoday.com/articles/19990819.htm. August 19, 1999.

25. Grossnickle, Joshua and Raskin, Oliver. "Market Research On The Web". http://hotwired.lycos.com/webmonkey/e-business/tutorials/tutorial4.html. January 2000.

26. Haneng, Alexander. "Live Stats from your Website". http://www.asptoday.com/articles/19991201.htm. December 1, 1999.

27. Henderson, Kenneth W. "Extract from Chapter 14 The Guru's Guide to Transact-SQL". http://www.vb2themax.com/HtmlDoc.asp?Table=Books&ID=2700. January 2001.

28. Holzschlag, Molly E. "Elements of Style". WebTechniques. March 2001. Volume 6, Issue03. pp34-37.

29. Holzschlag, Molly E. "Type Fundamentals For Nondesigners". WebTechniques. January 2001. Volume 6, Issue 01. pp30-33.

30. Howard, Robert. "Scaling VB COM with MTS". http://www.asptoday.com/articles/19990422.htm. April 22, 1999.

31. Johnson, Jeff. "Using Global.asa Correctly". http://www.asptoday.com/articles/19990413.htm. April 13, 1999.

32. Kamath, Manohar. "@@IDENTITY crisis". http://www.kamath.com/tutorials/tut001_identity.asp. August 13, 1999.

33. Lewallen, Jim. "What's New in ADO 2.0". http://msdn.microsoft.com/library/techart/msdn_newado20.htm. July 1998.

34. Lewis, John R. "How to Expose an Array Property". http://www.aspzone.com/articles/john/HowToExposeArray/HowToExposeArray.asp. December 10, 1998.

35. Llibre, Juan T. "ASP & IIS Optimization". http://www.asptoday.com/content/articles/20000901.asp. September 1, 2000.

36. Meier, J.D. "ASP Component Guidelines". http://msdn.microsoft.com/workshop/server/asp/server01242000.asp. January 24, 2000.

37. Microsoft Product Support Services. "HOWTO: Lifetime of a COM Component User IIS, ASP, and RDS". http://support.microsoft.com/support/kb/articles/Q166/2/79.asp. August 4, 2000.

38. Microsoft Product Support Services. "HOWTO: Obtain ObjectContext with ObjectControl Inside VB COM DLL From ASP and MTS". http://support.microsoft.com/support/kb/articles/Q238/2/74.ASP. January 9, 2001.

39. Microsoft Product Support Services. "INFO: Design Guidelines for VB Components Under ASP". http://support.microsoft.com/support/kb/articles/Q243/5/48.ASP. January 9, 2001.

40. Microsoft Product Support Services. "INFO: Do Not Store STA Objects in Session or Application". http://support.microsoft.com/support/kb/articles/Q243/5/43.asp. December 22, 1999.

41. Microsoft Product Support Services. "HOWTO: Call a Parameterized Child Command Through the MSDataShape Provider". http://support.microsoft.com/support/kb/articles/Q249/0/27.ASP. October 21, 2000.

42. Moran, Randie. "powers of observation". WebTechniques. December 2000. Volume 5, Issue 12. pp39-41.

43. MSDN Online Library. "Implementing with Visual Basic". http://msdn.microsoft.com/library/default.asp?URL=/library/psdk/iisref/crtc32n9.htm. November 15, 2000.

44. Negrino, Tom & Smith, Dori. *JavaScript For The World Wide Web 2nd Edition*. Peachpit Press. Copyright 1998.

45. Nickell, Joe Ashbrook. "Do One Thing. Market Like Hell". http://www.thestandard.com/article/display/0,1151,9212,00.html. January 28, 2000.

46. Pantone Access Color. "Color Emotion". http://www.pantone.com/ezine/fullstory.asp?story=2. February 9, 2001.

47. Pattison, Tec. "Creating Internet Applications with Visual Basic". http://msdn.microsoft.com/library/periodic/period98/vb.htm. December 1998. Microsoft Interactive Developer.

48. Power ASP. "Using the CDONTS component to send email from ASP pages". http://www.powerasp.com/content/hintstips/asp-email.asp. February 9, 2001.

49. Punch, Linda. "A Threat in the Cards For Online Retailers". Electronic Commerce World. March, 2000. Vol. 10, No. 3.

50. Putcha, Anila. "Marketing Meets Psyc 101". WebTechniques. March 2001. Volume 6, Issue 03. pp29-31.

51. Rifkin, Jeremy. "The Age of Access". http://www.thestandard.com/article/display/0,1151,12726,00.html. March 13, 2000.

52. Rosenheim, Mimi. "Now That's Service!". WebTechniques. March 2001. Volume 6, Issue 03. pp24-27.

53. Ryan, S. "Will community work for you?" http://www.builder.com/Business/Community/ss01.html. CNet, Builder.Com. December 2, 1997.

54. Sivakumar, Srinivasa. "Data Shaping with ADO – Part 1". http://www.asptoday.com/content/articles/20000807.asp. August 7, 2000.

55. Sivakumar, Srinivasa. "Data Shaping with ADO – Part 2". http://www.asptoday.com/content/articles/20000814.asp. August 14, 2000.

56. Sivakumar, Srinivasa. "Tips to Improve ASP Application Performance". http://www.15seconds.com/issue/000106.htm. January 6, 2000.

57. Smith, Steve. "Tips on Using Microsoft Transaction Server – Part 2". http://www.asptoday.com/articles/20000615.htm. June 15, 2000.

58. Smith, Steve. "Tips on Using Microsoft Transaction Server – Part 1". http://www.asptoday.com/articles/20000614.htm. June 14, 2000.

59. Stanislaw, Keith. "The Use of ActiveX DLLs with ASP". http://www.asptoday.com/articles/19990812.htm. August 12, 1999.

60. Vieira, João . "How Many People Are on your Site Right Now?". http://www.4guysfromrolla.com/webtech/061399-2.shtml. June 6, 1999.

61. Walker, Cathie. "Building an Online Community: What's Worked For Me." http://webdeveloper.internet.com/opinion/opinion_building_a_community.html. September 29, 1999.

62. Wells, Garth. "Implementing a Dynamic WHERE Clause". http://www.sqlteam.com/item.asp?ItemID=2077. January 14, 2001.

63. Williams, David M. "Using an ActiveX Server-Side Component with ASP". http://www.asptoday.com/articles/20000321.htm. March 21, 2000.

64. Wilson, Todd. "Building an ASP Component Using Visual J++". http://www.asptoday.com/articles/20000327.htm. March 27, 2000.

65. Wynkoop, Stephen. *Special Edition Using SQL Server 7.0*. Que. Copyright 1999.