

Problem Set 5

Setting up an online survey

Due: Thursday, May 8, 2003

***Important Note:* You are encouraged to work in teams of 2 for this assignment. If you do, the same grade will be given to both members. You may choose to work alone, but you will be expected to do the same amount of work. Do not leave this assignment for the last minute!**

In this assignment you will use Microsoft FrontPage to set up a data-gathering tool, in the form of a web survey, which takes consumer information from a web interface and saves it to either an Access or an Excel database.

You can purchase FrontPage from the campus bookstore.

There are appendices at the end of this problem set that will guide you through some common methods of designing forms and importing databases that you will need to perform in order to complete your web designs.

What to turn in:

The web address of your working web surveys (by email to the TA)

In this problem you will be creating a survey using FrontPage 2000 (or XP) to gather data on any topic of your interest. You can take any ideas from other classes or projects you may be working on which require you to gather some type of data about a product or service. Please feel free to choose any topic of interest to you. *Note: Regardless of what topic you choose, you should read the following survey sample. Its purpose is to evaluate customer satisfaction at the Sloan refresher court, and has important information and useful tips. You can turn this sample survey in, but **be aware you will receive extra credit for creating your own topic survey**. This sample assumes that you will raffle a prize among those who respond to your survey.*

SAMPLE SURVEY ON CUSTOMER SATISFACTION AT THE SLOAN REFRESHER COURSE:

Phase 1: Creating and importing the database to hold your results.

This step may not be necessary for those of you who have Access 2000 or newer. The reason is that FrontPage 2000 can automatically create the database to hold your results, but it will be in Access 2000 format (unreadable in Access 97). Because of that restriction, it is advised that those of you using an older version of Access should create this database prior, and then import it into your FrontPage project.

Step 1: Create a database that contains the survey results.

This is an important step. If your design is not well done, you will have problems later on and will have to come back to modify your file time and again. Use Microsoft Access to create one table (named *Results*, if you choose to) and remember where in you place this file. It should contain the following fields:

- Food Satisfaction Rating (numerical, expect values between 1 and 5)
- Hours Satisfaction Rating (numerical, expect values between 1 and 5)
- Service Satisfaction Rating (numerical, expect values between 1 and 5)
- Suggested Improvements (text, select **Allow Zero length: Yes** for all your text fields)
- Demographic information such as
 - Sex (text, Male or Female)
 - Program (text, MBA, LFM, MOT, etc.)
 - Graduation year (numerical, just the year)
 - Email (for those who want to be part of the raffle).
- And extra items you can obtain from respondents:
 - BrowserType (text, select **Allow Zero length: Yes** for all your text fields)
 - Address (text)
 - Timestamp (date)
 - Username (text)

Next, create a query that will tell you the number of responses (count) received by the database. You may want to name this query something like *TotalRespondents*.

Please note that you can save this database as an Excel file. The advantage of doing so is that Excel files can hold more information in a smaller file size and have more functionality in data analysis than Access. *The option is yours.* If you choose to use an Excel database, simply Choose “**Save As/ Export**” from the **File** menu in Access, then choose “**To an external File or Database**” and finally to “**Save as type: Microsoft Excel 97.**”

Step 2: Create a new web in Microsoft FrontPage 2000. This project will hold at least 3 web pages and the results database (Access or Excel) from Step 1.

Step 3: Import your results database into the web project.

[See Appendix A]. It would be useful if you named your database as *Results*, so we can refer to it further down in this exercise.

Phase 2: Design your survey page

In this phase you will create a FORM with questions, options, and text fields for users to input their responses. Furthermore, you will set this form to save the information into your database. This is the sticky part: if your database in phase 1 was not well designed, you will have problems saving to those fields, and will have to go back and modify your database so it accepts the inputs from FrontPage.

Step 1: Create your index.asp page.

This page will be saved in .asp format. That is, this page will be an *Active Server Pages script*, not just a plain .html file. You will be using the *Form Page Wizard* to create the questions in this script.

See [Appendix B](#) for instructions on using the Form Page Wizard. Make sure that you include the questions that deal with the quality of food, hours of operation, and service. Make the question scale 1 (lowest) to 5 (highest). It is recommended that you use radio buttons. Make another question for suggestions, with a text box for input. Finally, add questions for demographics: Sex (radio buttons for Male or Female), Program (a drop down menu would be nice), Year of Graduation (a numeric field will do), and Email (string field).

Make sure to include an introduction stating the purpose of the survey, expected time to fill out the survey, and any incentives for participating (e.g., raffle for movie passes, etc.).

Step 2: Set up the form to save to the database.

At this point, FrontPage does not know that you want to save the results of your questionnaire to your database. You need to indicate so.

Refer to [Appendix C](#) to learn how to save a form to a database. In this case, you will have to select the **Results** database (the only one we have) and the only table it should contain. Check to see that all fields are being saved, from food quality to demographics.

Moreover, ensure that the additional fields (browser type to user name) are correctly mapped.

Phase 3: Creating a response and an administrative page.

The response page should let users know their data was saved with no problem. The administrative page will be for you to both check on the number of people who have responded so far and to remotely pick up the **Results** file so you can analyze it.

Step 1: Create a response page.

You may want to name this page **Thankyou.htm**. It will serve as a thank-you note for survey participants, it can mention how soon they can expect to hear from you if they win the raffle (if you offer one), and links to redirect them to other sites (i.e. your home page) so they won't be lead to come back to the survey to fill it out again.

Step 2: Create an administrative page.

You may want to name it something like **admin.asp**, a script. Only you should know of the existence of this page that gives you access to the results file (email, IP addresses, and other sensitive information).

This page will show your response totals so far. To implement that, use Appendix D in conjunction with the query you created in Phase 1: **TotalRespondents**.

Next, create a hyperlink to the **Results.mdb** or **Results.xls** file itself, so you can remotely download and review this file (without having to physically touch the server for this). You can add this link simply by click-and-dragging the database file from the file view (on the left-hand side of the screen) onto the admin.asp page. Beware that your database file may be located inside the **fpdb** folder.

At this point, it is recommended that you save your files and run a test to make sure that the form is working correctly and information is being saved to your database. If this is the first time you are saving it, or if you are developing this web project in a folder outside your web server (i.e. PWS), you should use the **Publish** button and follow the instructions. This way, FrontPage saves all the extra files needed for the web site to run.

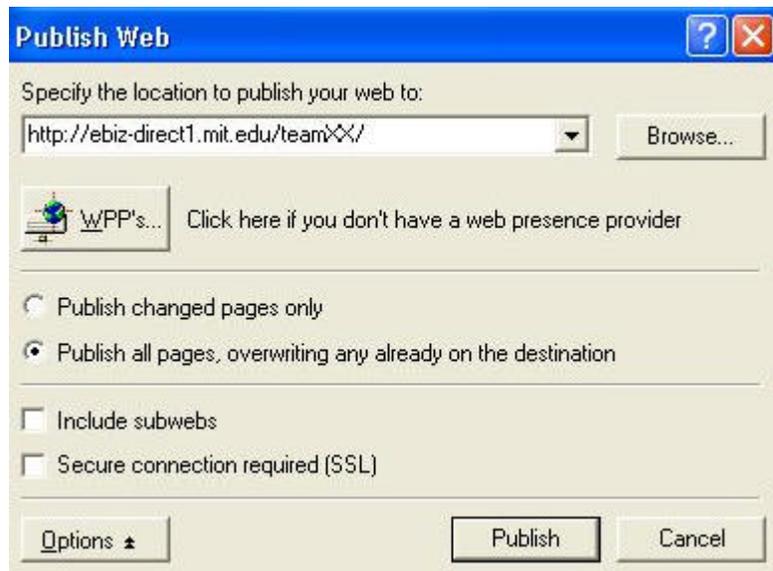
Phase 4: Design you own survey. (extra credit: 10%)

If you choose your own topic, please do not forget to include some introduction about the purpose of your survey in the first page (index.asp).

Phase 5: Publishing your survey (many thanks to the course TA for putting together these instructions)

To hand in the assignment, you will need to upload your work to a remote server. To do this, follow these directions:

- 1) Email the TA with the name of your team members. He will assign you a team number.
- 2) Complete the assignment as outlined. Make sure you use Frontpage (either 2000 or XP). Make sure you have imported your database (access file) into your web.
- 3) Go to the **File** menu and select **Publish Web...**
- 4) In the **Specify Location** box, enter: <http://ebiz-direct1.mit.edu/teamXX/> . But instead of “teamXX”, put the team number you are assigned (for example, if you are team 15, you would enter: <http://ebiz-direct1.mit.edu/team15/>). It should look like:



This screenshot is from the Microsoft FrontPage software program.

Hit the **Publish** button.

- 5) It will ask you for a username and password. The username is “**team1**” and the password is “**team1**” (it is team1 REGARDLESS of what team you are assigned to)
- 6) You are done. To check to see if your site works, check the URL: <http://ebiz-direct1.mit.edu/teamXX/> (if you also created an admin page as the assignment asks, the URL for that is <http://ebiz-direct1.mit.edu/teamXX/admin.asp>)

Done. Congratulations!!!

Appendix A: Importing and Defining a file as a Database

Use the following method when you have an Excel or Access database that you want to use as your working database in a FrontPage 2000 web project.

To Import a file: Click on **File** and **Import**. Select **Add File...**, and find your results database, click **OK**. If this is an .mdb file, it will ask you whether you want to use it as a database. Say yes and change the name from Database1 to something more meaningful.

If you are using an Access .mdb file, this step brings it into the web and defines it as your working database. You can check that this has happened by viewing your database web settings (**Tools** -> **Web Settings** -> **Database** tab) and ensuring your database is under the *connection* column.

If you are using other type of file (such as Excel), then you need to indicate to FrontPage that this file is a database. To do so, you need to change your web's settings as follows:

Defining a file as a database: Go to **Tools**, select **Web Settings...**, click on the **Database** tab, and choose **Add...** A new menu appears with a proposed name (such as **Database1**) for your file, but it is better to change it to a meaningful name. Under **Type of connection**, make sure that the radio button next to **File or Folder in current web** is selected. Click on **Browse** and choose your Access file. If you are using an Excel file, make sure to indicate a **File of type: Microsoft Excel Driver**” Click all OKs.

Appendix B: Using the Form Page Wizard

This method can save lots of time when designing a questionnaire to be responded by the use of one-of several answer forms (like radio buttons), any-of-many (like check boxes), or direct input (like text fields). Please follow the next steps:

- Click on **File, New, Page**, and select **Form Page Wizard**. Click **Next**.
- Click on **Add**. A new menu appears with several choices for type of input. Once you have selected a choice, you are returned to the first screen, where you can click **Add** again. Some of the items that we will be using from this menu are:
 - **Range**. This helps rate users' opinion on a scale. When you click **Next**, you can select it on a scale of 1 to 5, to use a radio buttons (by leaving the **use drop-down menu...** unchecked). You should select a meaningful name (such as *FoodQuality*) for this.
 - **Paragraph**: Used to input complete sentences, such as suggestions and opinions. Change the prompt of this question to fit your form (such as *Do you have any suggestions for improvement?*) and select a meaningful name for this input field.
 - **String**: Used to enter a short answer, such as email. Once again, choose a prompt (i.e. *Enter your email if you want to be part of our raffle*) and the name for this field.
 - **Number**: Used to enter information such as year of graduation.

- **One of Several:** Used to input data such as sex or program of study. To choose one of the options, you can use drop-down menus, radio buttons, or a list. Simply input your list of options in a different line and choose an appropriate name for the field.
- **Any of several:** Used to select more than one (or none) option. The only option is to create them as a list of check-boxes.
- **Boolean:** Used to provide a yes/no or true/false answer.
- Once you are done selecting choices for all your questions, make sure they are arranged in the order you want them to appear in your web page. Click **Next**.
- Choose the presentation layout options that best work for you (hint: you will seldom need to create a Table of Contents for a survey). And click **Next**.
- For output options, it is recommended that you select **use Custom CGI script**. The other two options will save to an html page or a text file, which we are not using in this problem set (we already have databases in place).
- Click **Finish**.

Feel free to move some of the fields to make your page more presentable. Moreover, it is a good idea to right-click and check on the **Form Field Properties** of your different fields. You can customize them to accept just a certain number or type of values (Validate...).

Appendix C: Saving a form to a database.

This method assumes you have created and defined a database properly defined to hold the results of an equally well formatted questionnaire form. The table from your database should have as many fields, and of the same data type, as you form, and vice versa. Otherwise, this method will result in an error and you will be forced to modify the design of either your database or your form.

- To start, right-click anywhere inside the form (identified as a dashed square where all the fields are included) and select **Form Properties**.
- Select the radio button next to **Send to database**, and choose a name for this form.
- Also choose if you want to post the results to this same page, or, if you are working with frames, to the appropriate frame. Click on the button next to **Target Frame**.
- Click on the **Options** button.
- Choose the **Database Connection to use** (i.e. *Results*) and the table to hold your form results. If you have Access 2000 installed in your computer, you can simply select **Create Database**, and FrontPage 2000 will automatically create it for you. If your form has more fields than your database, just click on **Update Database** to let FrontPage modify your Access file by adding the extra fields you need.
- If you have a confirmation page (such as *Thankyou.htm*) enter that page name in the **URL of confirmation page** for users to see when they click the *Submit* button.
- Next, select the **Saved Fields** tab and map each of your form fields to the proper database column. Do this by double-clicking on each field and selecting the right column in the menu that appears.

- Next, select the **Additional Fields** tab and map the browser-type, etc. fields to the proper database column. If you are not planning on keeping this information, then remove that additional field from the list (Click the **Remove** button).
- Click **OK** for both windows.

Appendix D: Posting results from a database into a page

This method will help you take information from a database table or query (but not a form or a report) and present it to your users in different forms. You can present the data for a static page (html) as a table or a list. For an interactive page (and ASP script) you can present the data as a drop-down menu or text fields, which users can choose and manipulate by the means of a *Submit* button.

- From the main menu, select **Insert, Database, and Results**.
- Select **Use an existing database connection**, and choose the database you are working with (such as Results, or Northwind). Click **Next**.
- Select the table or query you want to use (such as *TotalRespondents*, *Categories*, or *Products*), or create your own query. Click **Next**.
- By using the **Edit List** button, you can select what fields to display. When you want to ultimately display your results in a drop-down menu, you should select no more than two fields to display (such as the *CategoryName* and the *CategoryID*).
- You can use the **More Options** button to filter, limit, or sort the displayed results further. Following are the options you will find:
 - **Criteria:** Useful in setting up a search form that works like the “where” statement of a query. Click on **Add** and you will find a list of the different columns you are displaying. Select one of them to be equal (or not equal, greater than, etc.) to a given value. One example is for the Northwind database, you can select to display a product such that **CategoryID equals Category**, where you have preselected to display the **Products** table, and the **Category** variable comes from a previous page or frame.
 - **Ordering:** So that you can choose which field to “order by”
 - **Defaults:** Important if you want the page to display some initial results instead of an error message. One example, one displaying all the categories in the Northwind database, the **CategoryID** can be set to default 1, so that *Beverages* will be displayed when users first access the page.
 - **Message to display...:** You may want to change this field so that users know the reason why this page is blank (i.e., they need to select a product before they can see the pricing information on it).
- Click **Next**.
- Here, you can select four different types of output:
 - **Table (static):** gives you a list of the results of your table or query.
 - **List (static):** Also gives you a listing of results in different formats.
 - **List - Text Field or Scrolling text Option (interactive):** Outputs data inside a text box, which users can modify and submit back to your web application. One example is an order form, where users get a quantity to purchase (default 1) that

- they can change to buy 2 or more and submit back the new quantity for processing.
- **Drop-down list (interactive):** In this method users can select one of several choices and submit the option back for processing. One example is choosing a category from the Northwind database (beverages, condiments, etc.). The **Displayed value** would be the **CategoryName**, while the **Submitted value** would be the **CategoryID** (1, 2, etc.). This CategoryID would be sent to the next frame so that the products that belong to that category can be displayed.
 - Click **Next**.
 - This is the last part of the wizard. For the purpose of this exercise, you **should NOT select Add Search Form** in any instance. Click **Finish**.