# *Web Programming IV (420-C40-HR)*

# *Lab 6 – Master Pages*

Date assigned: Thursday, February 9, 2017

Date due: **Thursday, February 9, 2017, 4:00 p.m.**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

* Work with Master Pages

Lab Set-Up

1. Create a folder **YourUserName\_C40\_L06** in your H:\420-C40\Labs folder.

To do:

**Part A – Indo-US College**

1. For this project, you create pages for the Indo-US College Database that you have been using in your database courses. We are not going to connect to the actual database at this point, but we are going to create the screens and ‘simulate’ data.
2. We are going to start using Projects rather than web site. Web sites are good, but are limited. Projects provide more flexibility in a number of areas. Start Visual Studio and create a Visual Basic Project Web project (File->New->Project->Templates->Visual C#->Web->ASP.NET Empty Web Application) called StudentInfo. The site must be in the C40\_L06 folder you created above.
3. You are going to create a page to search for students and a page to ask the students questions about a course they took.
4. Master Page
   1. Create a Master page called mySite.master which contains a header and tag line. The header is “Indo-US College”. The tag line is “Caring About Your Education”. Format the header nicely using CSS. Make sure this is a HTML5 document.
   2. Add a footer to the master page which says:

To contact us, call us at 819-555-0400 or email us at survey@indous.com

* 1. Make sure that the footer is formatted nicely to match the header information. Also, as usual, make it a little bit smaller than the surrounding text, centre it, etc.

1. Create a Student Survey page called studentsurvey.aspx which uses the master page for formatting. The page has the following controls:
   1. A textbox to enter a student ID
   2. A button called Get Courses
   3. A list box which will be populated with the list of courses the student is taking
   4. A list of survey questions as follows:

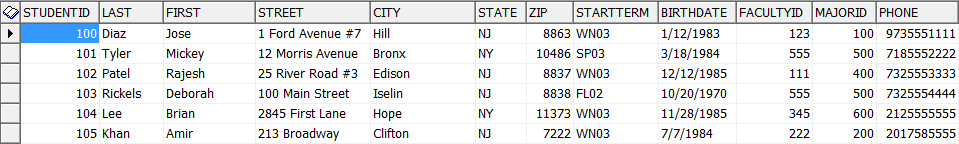
Met expectations

Professor’s Knowledge

Fair assessments

* 1. Each of the survey questions has 4 possible answers as radio buttons for Not satisfied, Somewhat satisfied, Satisfied, Completely satisfied.
  2. An Additional comments text box
  3. A check box for “Please contact me to discuss this survey”
  4. Two radio buttons for “Contact by email” and “Contact by phone”
  5. A Submit button

1. The operation of the page is as follows:
   1. When the Student Survey page is first displayed, all of the controls except for the Student ID text box and the Get Courses button are disabled.
   2. When the user enters a student ID and clicks the Get Courses button, any courses that the student is currently taking are displayed in the list box and all of the controls on the page are enabled. You will need to make up at least 3 courses to add to the list.
   3. To complete the survey, the user selects an incident, makes the appropriate selections and entries, and then clicks the Submit button. Then, the Survey Complete page is displayed (see below).
   4. The “Contact” radio buttons are disabled unless the checkbox “Please contact me to discuss this survey” is checked.
2. Survey Complete page. Create a survey complete page with the following specifications:
   1. Is called surveycomplete.aspx and uses the same master page.
   2. The page displays a message that depends on whether or not the customer has asked to be contacted. If the customer does not want to be contact display “Thank you for completing the survey. We value your comments”. If the customer does want to be contacted, display the **additional** message, We will contact you by (phone/email) as soon as possible (where the value of phone/email is dependent on the radio button selected on the previous page.
   3. The customer can close this page or complete another survey by clicking a Return to Survey button.
3. Specifications
   1. Create a class named Student that consists of attributes for each of the following fields:



Check the Oracle database to determine the type/length of each field

Use the Public Property methods in C# to define the gets and sets (the short form way is fine)

* 1. Create a class named Course with the following fields:
     1. CSID (Course Section ID) as Integer
     2. CourseID as Integer
     3. CourseTitle as String
     4. TermID as String
     5. FacultyName as String

Use the Public Property methods in C# to define the gets and sets (the short method is fine)

Create a method called CourseDisplay to format the courses and display them in the list box. The courses should be formatted nicely using some delimiter (I recommend a /) between the values. For example:

EN100/Basic English/Fall 2012/Jones (you do not need to display the CSID).

When the survey page is loaded and a course ID is entered, create at least 3 courses to display in the list box (you can look at the data in Oracle if you want data values). That is, create 3 course objects and call the CourseDisplay method to format them for display in the list box. You can use the same courses regardless of the student entered.

* 1. Create a class named Survey that consists of the following fields that represent the data for a survey. Use the Public Property methods in C# to define the gets and sets (The short method is fine)

StudentID Integer   
CSID Integer  
MetExpect Integer  
ProfKnowledge Integer  
FairAssess Integer  
Comments String  
Contact Boolean  
ContactBy String

* 1. Use labels instead of HTML within the survey page so that you can enable and disable the controls along with the other controls.
  2. Use radio button lists to implement the ratings for the three questions in the survey. The Text properties for the list items in these lists should be set as above, and the Value properties should be set to 1, 2, 3, and 4.
  3. Use a radio button group for the last two radio buttons on the page.
  4. Use required field validators for the Student ID text box and the Course list box, and use a compare validator for the Student ID text box that checks for an integer value. In addition, use validation groups so that the validators for the Student ID text box are executed only when the customer clicks the Get Courses button, and the validator for the Survey list box is executed only when the customer clicks the Submit button.
  5. When the Course Survey page is first displayed, set the focus to the student ID text box. Then, when the customer clicks the Get Incidents button move the focus to the Course list box.
  6. When populating the Incidents list box, create list items whose Text property is set to the value of the CourseDisplay method of the incident and whose Value property is set to the CS ID. In addition, the Text property of the first item in the list box should be set as -- Select a Course -- and the Value property of this item should be set to “None.”
  7. When the survey is submitted, the application should create a Survey object and set its properties appropriately. The Survey Complete page is displayed. (Although a complete application would also perform some processing on the Survey object, this project doesn’t include that processing).
  8. The Survey Complete page displays a nicely formatted message thanking the user for participating and a link button allowing the user to return to the previous page.

**To submit**

When you have completed the lab exercise, create a single zip file called YourUserName\_C40L06.zip. The zip file must contain all of the parts of the lab in the folder you created at the beginning of the lab. Copy the file to the Moodle page for the course.