LAB #2 – Using Web server controls in ASP.NET applications

Student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Purpose: The purpose of this Lab assignment is to:

* Become familiar with Web server controls
* Use Web server controls in simple ASP.NET applications

References:

* Textbook.
* Lecture notes

This assignment may be completed individually by all the students. Submit the solution in a zip file using **assignment link on blackboard**. The file must be named according to the following rule:

yourlastname\_CEIL865Labnumber. Your **project should also use the same name**.

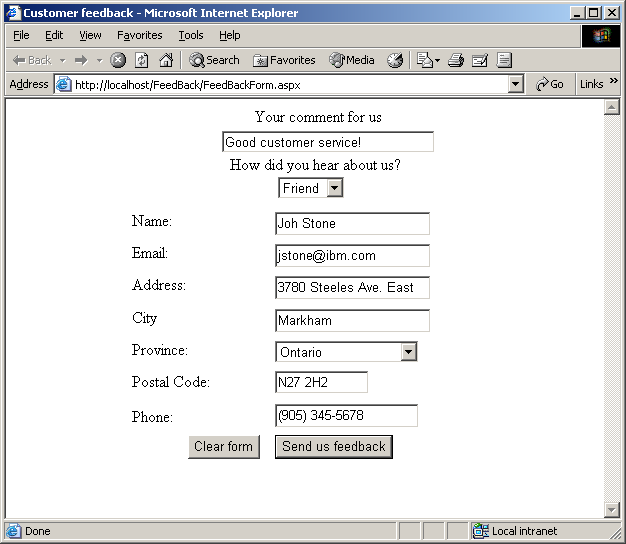
Example: smith\_CEIL865Lab1

If there are two exercises, add to the file name: \_Ex1 or \_Ex2.

Example: smith\_CEIL865Lab2\_Ex1

**Exercise #1**

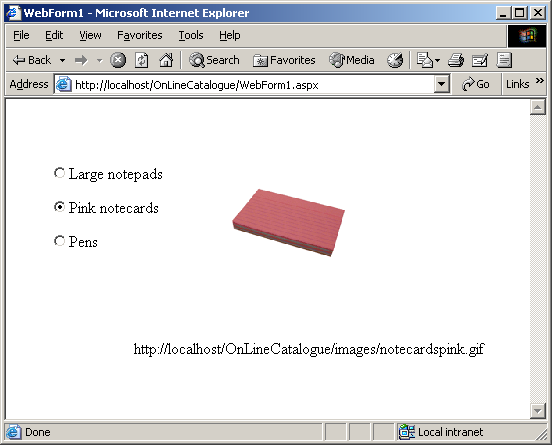
In this exercise you will create a customer feedback similar to the form shown below:



Use Labels, Textboxes, DropDownLists and Buttons to design the form. Provide the interaction with the user. When the user clicks on the *Send* button, display the collected information in a *ListBox* web server control, which can be placed on the same page. Allow the user to clear the form by providing the code for the *Clear* button.

**Exercise #2**

In this exercise you will design the product list page for an Online catalogue. Create a Web page that displays the product list as a RadioButtonList control. Display a default graphics when the page loads. When the visitor clicks a radio button the related graphic image should be displayed in place of the default graphic. The RadioButtonList control *values* should be the name of the graphic image, and the *text* should be the name of the product. You must obtain the value of the selected item as is shown in my code.



Here is a possible alternative of the code for the event-handler method:

public void Check\_Clicked(object sender, System.EventArgs e)

{

//

for (int i=0; i<rdProducts.Items.Count; i++)

{

if (rdProducts.Items[i].**Selected**)

{

Image1.ImageUrl= Request.Url.GetLeftPart(UriPartial.Authority) +

Request.ApplicationPath + "/images/" +rdProducts.Items[i].Value;

Label1.Text=Request.Url.GetLeftPart(UriPartial.Authority) + Request.ApplicationPath + "/images/" + rdProducts.Items[i].Value;

}

}

}

You should set the **AutoPostBack** property of the **RadioButtonList** to **true**. Store the images in the *images* directory which is a subdirectory of application’s directory.