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

# *Lab 2 – Fun with Controls*

Date assigned: Thursday, January 26, 2017

Date due: **Thursday, January 26, 2017, 4:00 p.m.**

**Learning Objectives**

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

* Understand how to create a simple ASP.NET website
* Understand how to create ASP.NET Web Pages
* Create and use CSS in Visual Studio
* Work with various ASP .NET Controls

Lab Set-Up

1. Remember, you must use CSS for all formatting and styling of your web pages. All CSS must be kept in a separate file with an appropriate name. CSS files should be stored in a folder called “styles” which is a subfolder of the root folder of the website.
2. Make sure you name the controls properly. See the Moodle page for standards on how the pages are to be named.
3. Start Visual Studio and create a new Web site (File=>New=>Web Site). Make it an ASP .NET Empty Web Site. Name the site *yourusername\_C40L02*.

To do:

**Part A – Single Page Application**

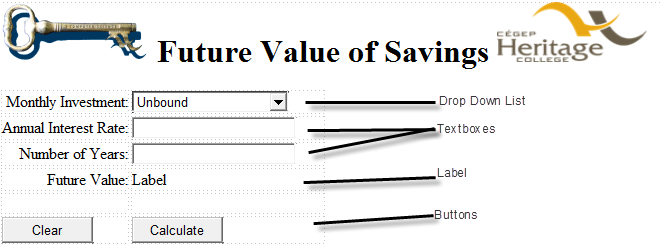
1. Add a folder named images to the site (right-click on site name and choose Add -> New Folder) and add the Heritage logo and the CS logo to it (Select the folder, right-click, choose Add Existing Item…, browse to the file and select it). You can find the file for this logo on the S drive for the course.
2. Create a new Web Form by right-clicking on the site name in the Solution Explorer and choosing Add New Item. Choose to Add a Web Form. Call the new item PresentValue.aspx. Make sure you add an appropriate title to the page.
3. Switch to Design mode and drag the image of the Computer Science logo to the div inside the form. Look at the source code and you will see that a new class called auto-style1 has been added to the top of the file. Change the name of the class to imageSize and change the width to 150px and height to 65px (you will have to change the class name in the image tag as well). Next to the logo add the words Future Value of Savings and format it as a header level 1 (use the drop down in the list)
4. On the end of the line add the Heritage College logo. Once again, a new auto-style will be created. Delete the new style and add the class imageSize to the second image as well.
5. Go to the next line (press return at end of the line) and add a table by selecting Table -> Insert Table. Set the number of rows to 6. You can leave the rest as defaults and select OK.

Note that a new style called auto-style1 is created.

1. Go into source code and remove the style auto-style1 and remove the class attribute on the table element.
2. In the first 4 rows of the table in the first column add the following text:
   1. Monthly Investment:
   2. Annual Interest Rate:
   3. Number of Years:
   4. Future Value:

Select all four of the cells in the table and use the properties window to change the **style** of all four cells to have the text be right aligned.

1. In the second column of the first row add a drop down list by selecting the Toolbox DropDownList and dragging it to the location.
2. Do the same to add a text box to the next two rows and a label to the column after Future value. Add two buttons in the final row (skip the fifth row). Use the properties panel to update the fields to look like the following image:

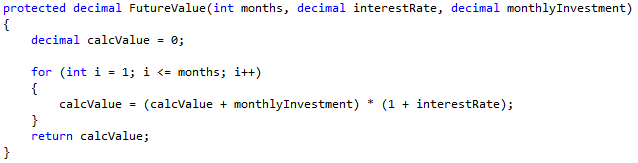


The fields (drop down list, text boxes and label are each 12 em and the buttons are each 7 em wide.

The ID of the Drop Down List is ddlInvest. The ID of the interest rate text box is txtInterest. The ID of the Number of Years text box is txtYears. The ID of the Future Value label is lblFuture. The IDs of the buttons are btnCalc and btnClear respectively.

Note: Visual Studio may add more auto-styles to the <td> elements. Remove all of them.

1. Change to source view and change the title of the page to “My Future Value”.
2. Press F5 to run the application. If the dialog box asks whether you want to modify the web.config file to enable debugging, click the OK button. Now, test to see what works by clicking on the controls. Also, check that the web form looks the way it’s supposed to. Stop the application by exiting the browser. You will also need to stop the execution in Visual Studio by pressing the stop button in the menu bar.
3. Now to add some code (test at each step!!!):
   1. In design view, double-click on an open area outside of all the controls. This opens the code behind file (PresentValue.aspx.cs) and creates a protected method for the Page\_Load event. Add code to the Page\_Load event handler to add values to the drop down list. Add values from 50 to 500 (inclusive) in increments of 50.
   2. Switch to the asp page in Design view and Double click on the Clear button and add code to set the drop down list to 50 and clear the other fields.
   3. Add the following method manually to the code behind the page:



* 1. Switch to the asp page in Design view and Double click on the Calculate button and add code to the click event. Get the values from the fields and convert them to the correct type. Then call the FutureValue method to calculate the future value and display that as a dollar amount in the future text label.

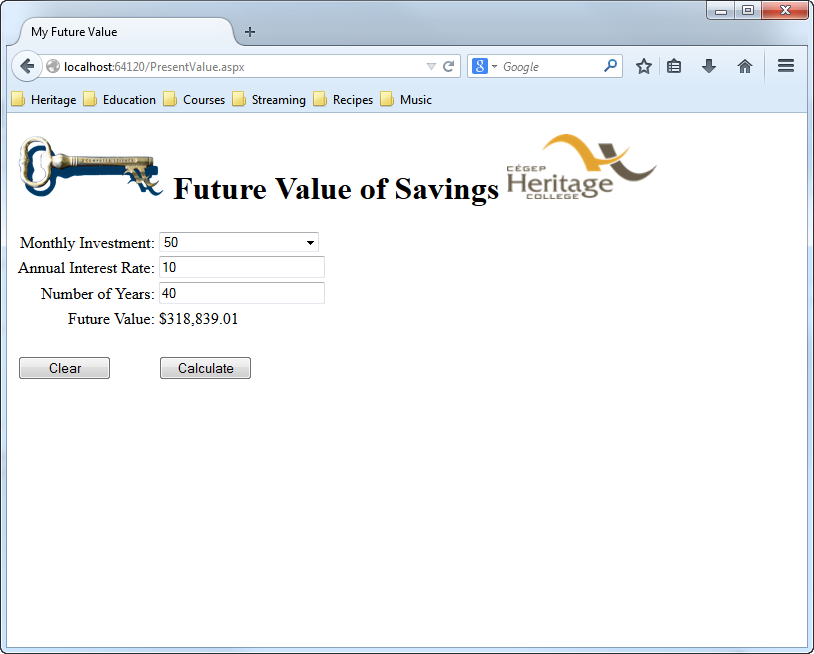
1. Run and test the application under a number of conditions. Change some properties and test it some more.

Some values: 100, 12, 1 -> Future Value = $1280.93

500,5,10 -> Future Value = $77,964.64

50, 10, 40 -> Future Value = $318,839.01

1. I purposely left an error in the code. Add a comment at the location of the error as to what the error is and WHY IT OCCURS. To correct the error you will need to add the following code somewhere: 



1. Move the existing to an external CSS file and add more to make it look nice.

**Part B – Events R Us**

1. Create an ASP .NET file, called Events.aspx which calculates the cost of an event such as a wedding.
   1. The form contains controls for:
      1. Name for the Event,
      2. Number of Guests,
      3. Selects from 1 of 5 costs per guest,
      4. Live music, DJ or mixed,
      5. Open Bar
   2. Possible costs for guests are (use a drop down list)
      1. $20 (Cold Buffet)
      2. $30 (Hot Buffet)
      3. $40 (Table Service, one course meal)
      4. $60(Table Service, three course meal)
      5. $100 (Table Service, six course meal)
   3. The cost of the guests is the number of guests \* selected cost per guest.
   4. Music is selected by Radio button of three values and costs $500 regardless of which is selected. If no music is selected, there is no cost.
   5. Open Bar is selected via checkbox and costs $30 per guest if selected.
2. When the user pressed the Calculate button, perform the calculations to determine the cost of the wedding. The results can be displayed below the form. Make sure that you display the results nicely formatted with a descriptive header and company information (Events ‘R’ Us) and the values in a table or div such as the following example. Use label controls to display the entered and calculated values.
3. Use External CSS that you create in Visual Studio and attach to the ASP page.

Events ‘R’ Us for All your Events

Cost Estimate for Lady GaGa’s Wedding

|  |  |  |
| --- | --- | --- |
| Item | Cost | Totals |
| Number of Guests | 400 |  |
| Cost per Guest (1 Course Meal, With Service) | $40 |  |
| **Total Cost for Guests** | | **$16,000** |
| Music | Live | **$500** |
| Open Bar (cost per guest = $30) | Yes | **$12,000** |
| **Total Cost** | | **$28,500** |

1. Your file does NOT have to look exactly like this, but it should be nicely formatted (including numbers and dollars) and easy to read the information.

**Part C – Other Controls**

1. One of the specialty controls that ASP.NET includes is the Calendar control, which displays a graphic calendar with selectable dates. The calendar control has numerous properties that you can use to format the display of the calendar. However, you can use the predefined autoformat scheme to format the calendar as well.
2. Create a page (web form) called Others.aspx. Add two calendar controls to the page. Format them using two different formats (predefined ones are okay if you like). Add a label below each calendar which will hold the date selected for that calendar. Add two more labels on the page which will hold the date difference in Days and in Seconds
3. Double click on one of the calendar controls (in design mode) to create a selection changed event handler. In the event handler, assign the corresponding label to the string value of the selected date. To do this, assign the Label1.Text to the Calendar SelectedDate ToString using the “D” as the parameter.
4. Check if the other calendar control has a selected date. If the other calendar has a date selected, calculate the difference between the selected dates. If the other calendar does not have a date selected, do nothing.
   1. To calculate the difference you can subtract one date from the other like this:  
      TimeSpan ts = Convert.ToDateTime(date1) - Convert.ToDateTime(date2);  
      (ts.TotalDays); //difference in days
5. Display the Date difference in days in one of the remaining labels and the date difference in seconds in the other.
6. Repeat steps 2 and 3 for the other calendar control.
7. Once you have it working this way, add a protected method which is passed the two dates and returns the number of seconds difference and use this to update the label with the value. Add another protected method that is passed two dates and returns the number of days difference and use his to update the label with the value.

**To submit**

When you have completed the lab exercise, create a single zip file called YourUserName\_C40L02.zip and copy the file to the Moodle page for the course.