# BCIT

**Comp 4952 Technical Programming 2**

**Technical Programming Option**

# Option Head Mirela Gutica

**Fall 2015**

Mark: \_\_\_\_\_\_\_\_ /100

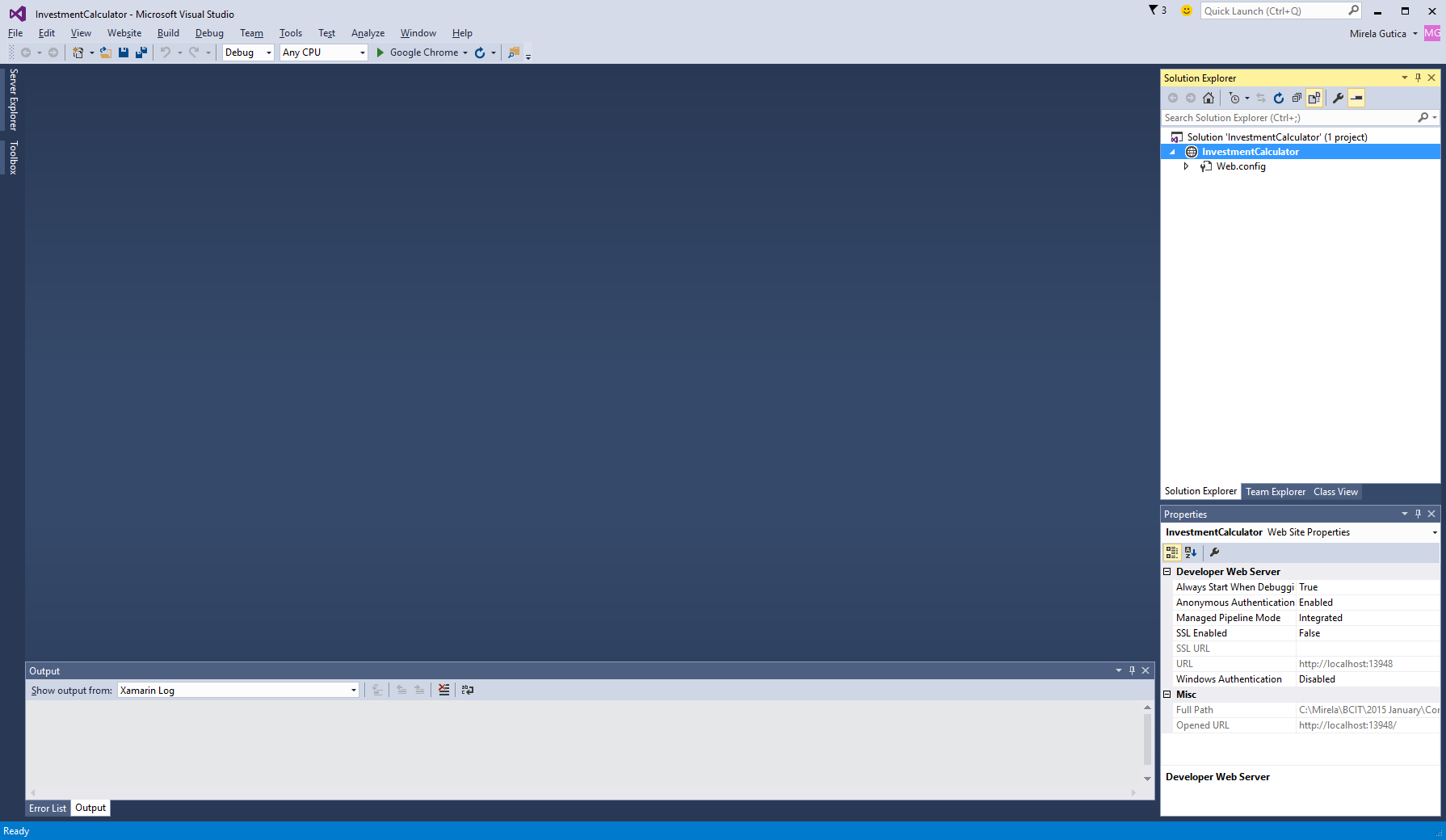
Lab 2

**How to Create a Web Application with Web Forms in ASP.NET**

This is an in-class assignment. The in-class assignment should be finalized no later than Monday, October 12, 2015**. No late assignments will be accepted**.

**Requirements**:

1. The assignment has 2 parts: coding and questions. Submit your work in D2L and provide at the beginning of the word/text document containing the answers the link what I have to use to test your solution.
2. You will design and implement a simple Web Form website that will expose you to the most important aspects of ASP.NET development, data flow and web server controls.
3. The application calculates the future value of investments based on user input. You will add validation and style.
4. Use Visual Studio 2015.
5. Start a new “Web Site”
6. Select ASP.NET Empty Web Site. Choose a folder for your website.

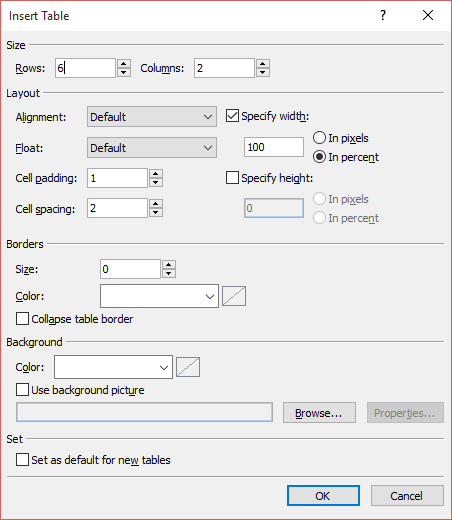
Observe that the new created website has only Web.config folder.

By default, the website is created under My Documents\Visual Studio 2015\Projects. You can choose a different directory.

You can change the default directory: Tools/Options/Projects/Solutions/General Category.

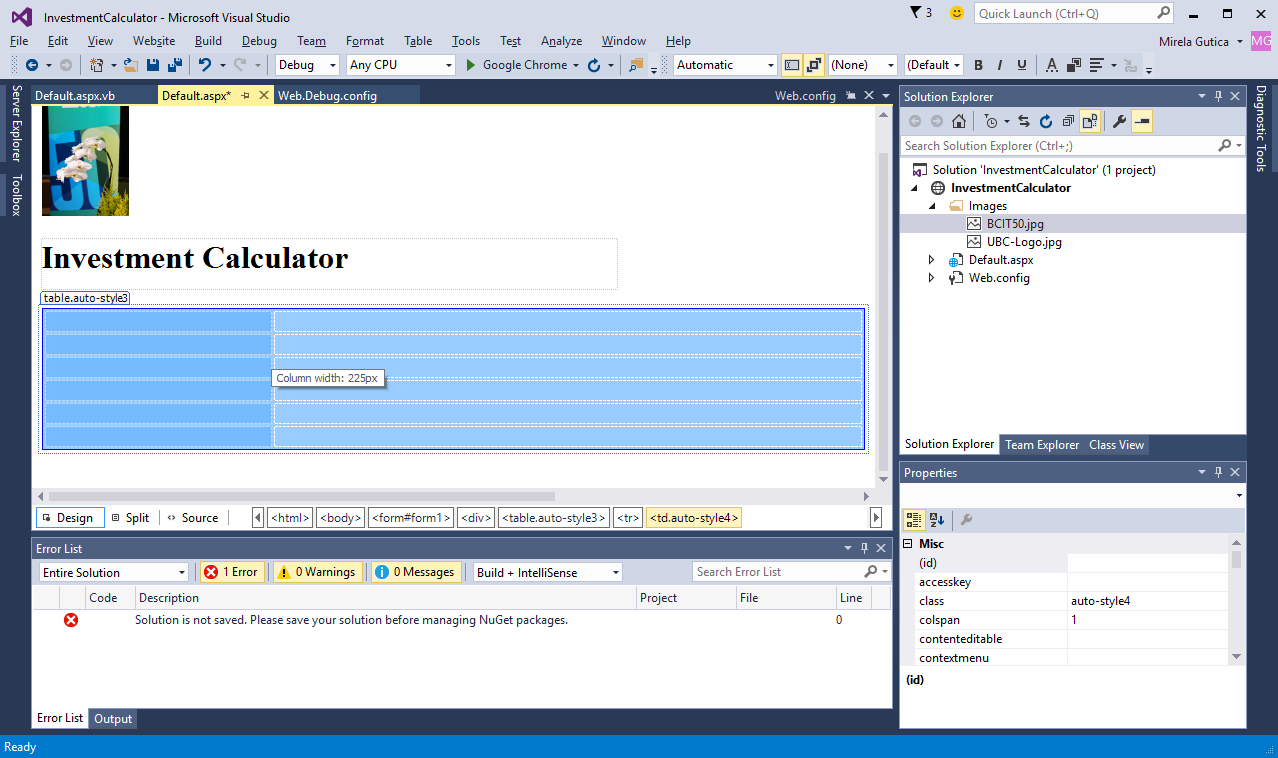
When you create a project/website you can select a target framework: By default, 4.5.2 is used by Visual Studio 2015.

1. Observe the designer, the source code and the code behind.
2. Add a Web Form to your website. Use the Add New Item dialog. Select the language C#. The new web form is called: default.
3. Add a folder to the website. Name it “Images”. Add an image/logo to the folder (use Add Existing Item…).
4. Note that you can also add a special folder: “ASP.NET folder”.
5. Add HTLM elements to the form (two ways):
   1. In the designer: e.g., drag the image to the designer.
   2. In the source type “<” and the intelliSense will help you enter snippets, tags, attributes.
6. Add a logo.
7. Add a title to your website including your name (e.g., Peter’s Investment Calculator).
8. Add a table to your form. By default, use “flow layout” (text and controls are positioned from left to right and from top to bottom). A table gives you the flow control. Another way to achieve flow control is to use CSSs.
   1. In designer, use the Table menu (an alternative choice is to use the Table control from the Toolbox).
   2. Use the Insert Table dialog to define the table’s properties.

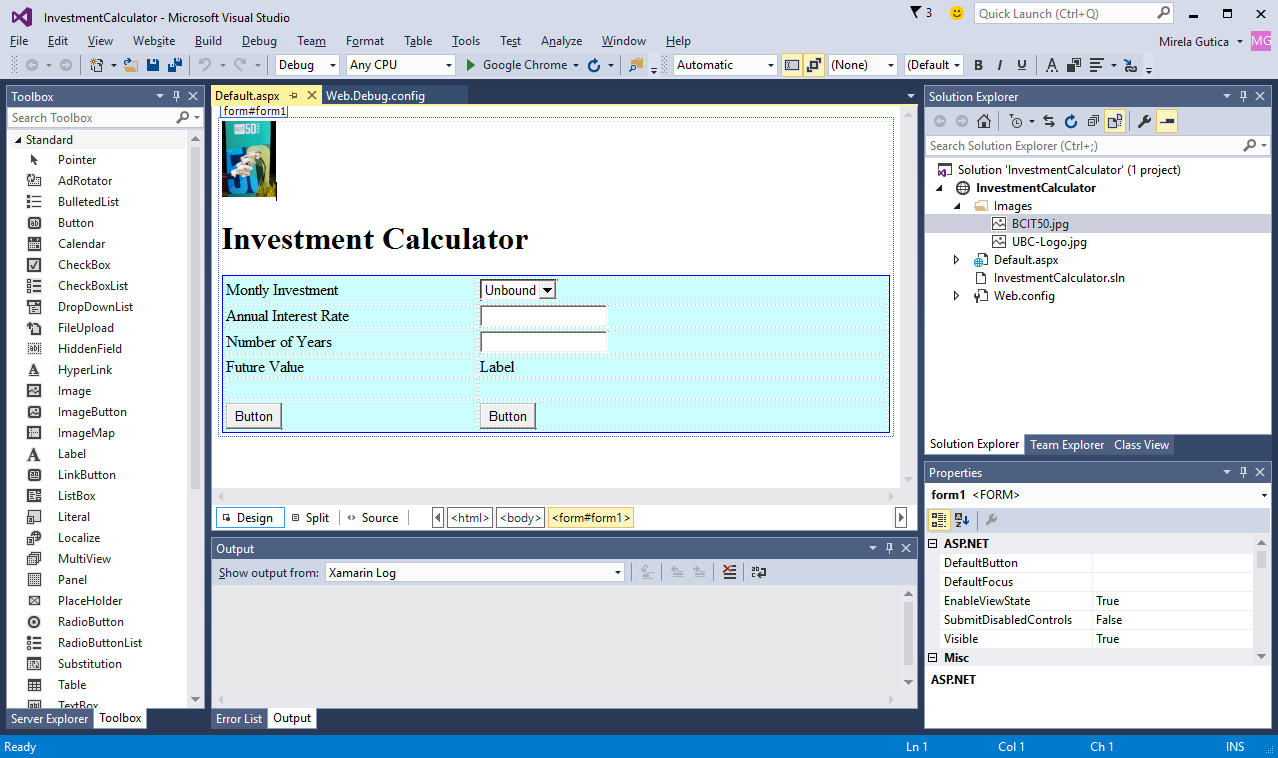


* 1. You can format the table after you create it. You can resize rows/columns by dragging theirs borders (bottom and right, respectively).

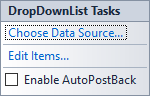
1. Add web server controls to the web form.
   1. Observe in the Toolbox the categories of controls available in ASP.NET Web Forms.
   2. Web controls can be (1) dragged to the form/table in designer or (2) inserting a snippet of code in html.
   3. As in C#, use the Properties window to set properties and events to web controls.



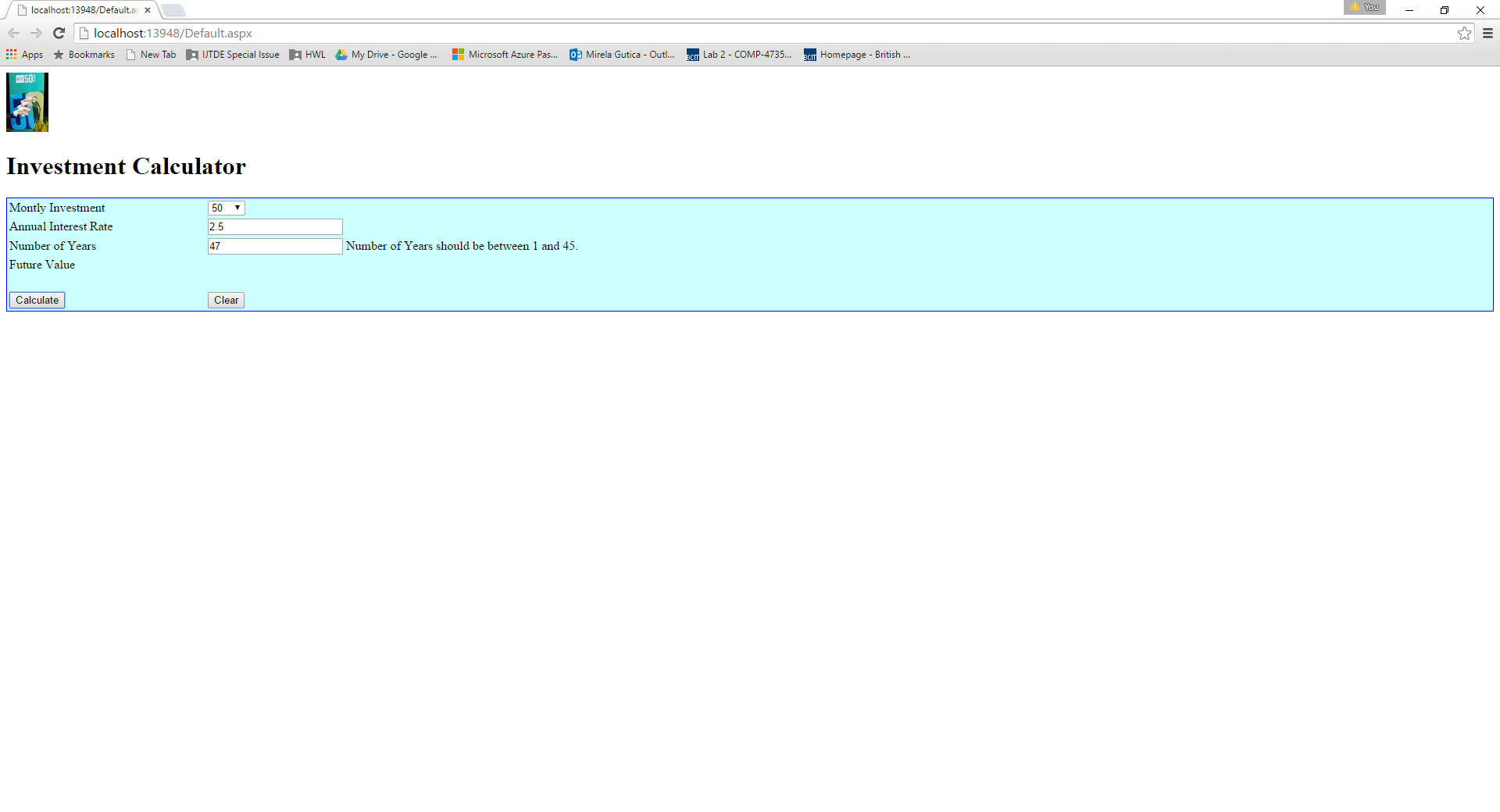
* 1. Many web server controls have smart tag menus that provide options for performing common tasks and setting common properties.
  2. Drag on your table the following controls: one dropdownlist, two textboxes, one label and two buttons.
  3. Name the controls properly.
  4. Give initial values of your choice to the inserted textbox controls.



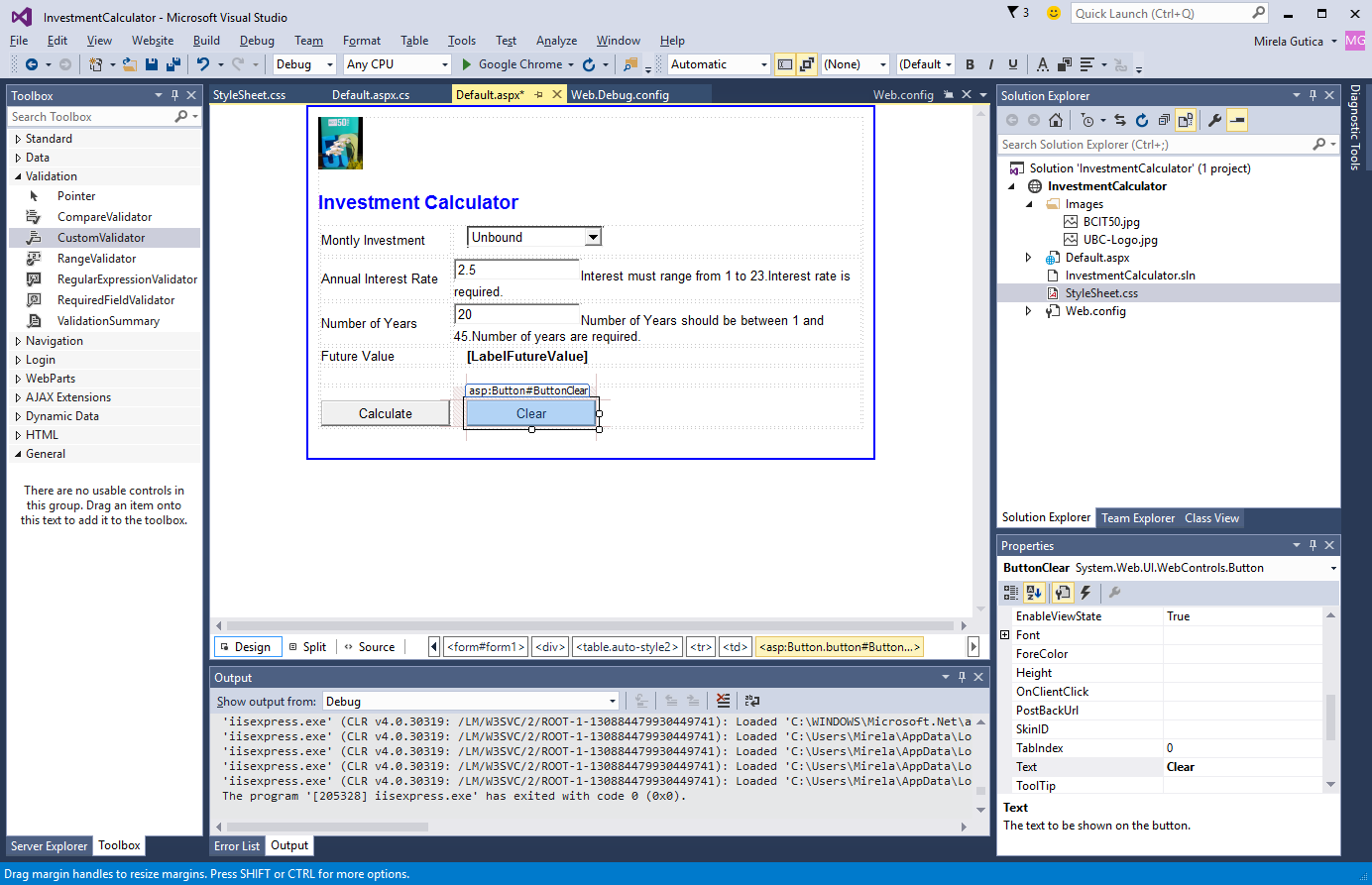
* 1. Observe the smart tag menu for the dropdownlist web control and how it can be connected to a data source. Leave the control unbound for now.

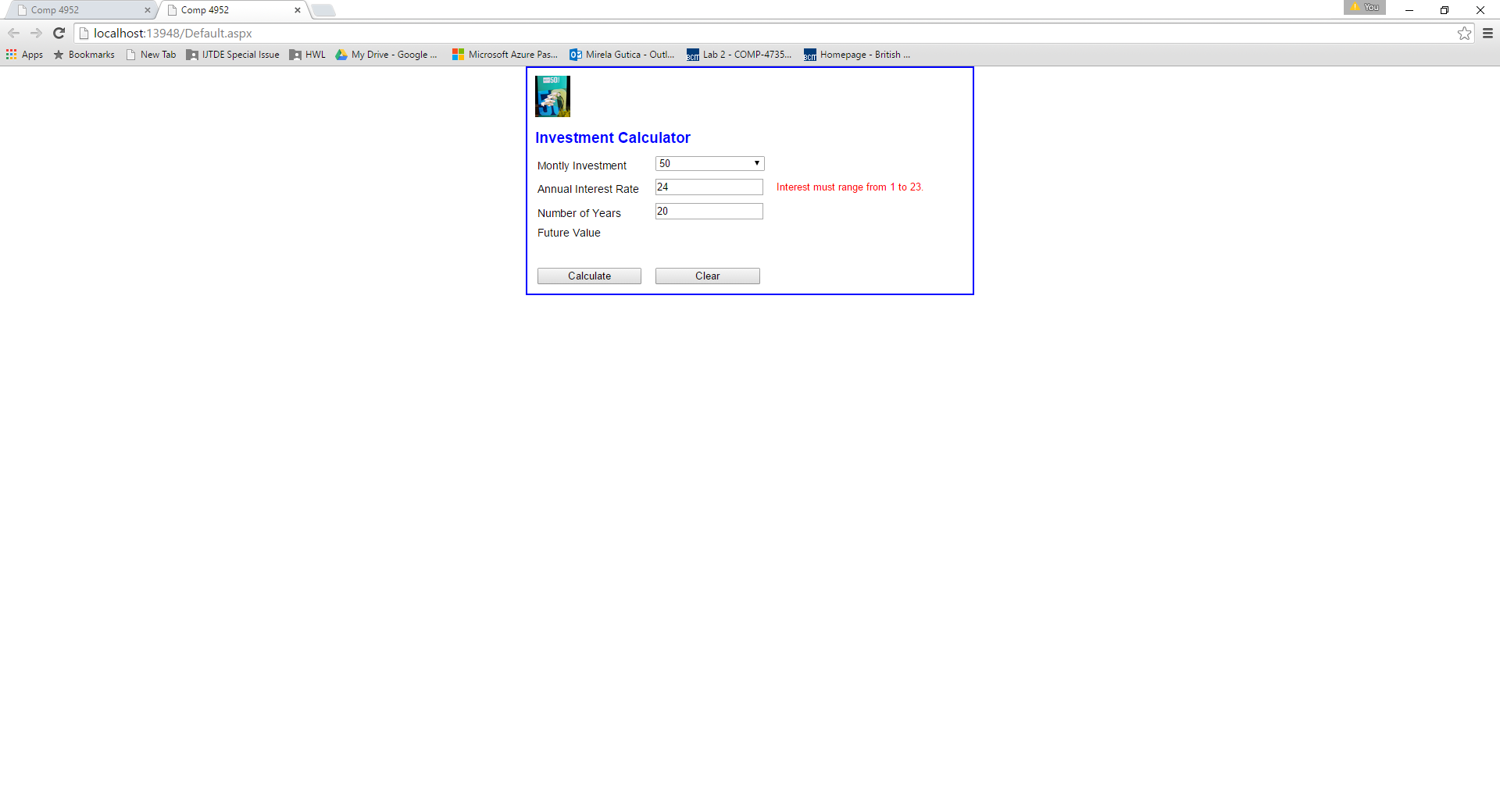


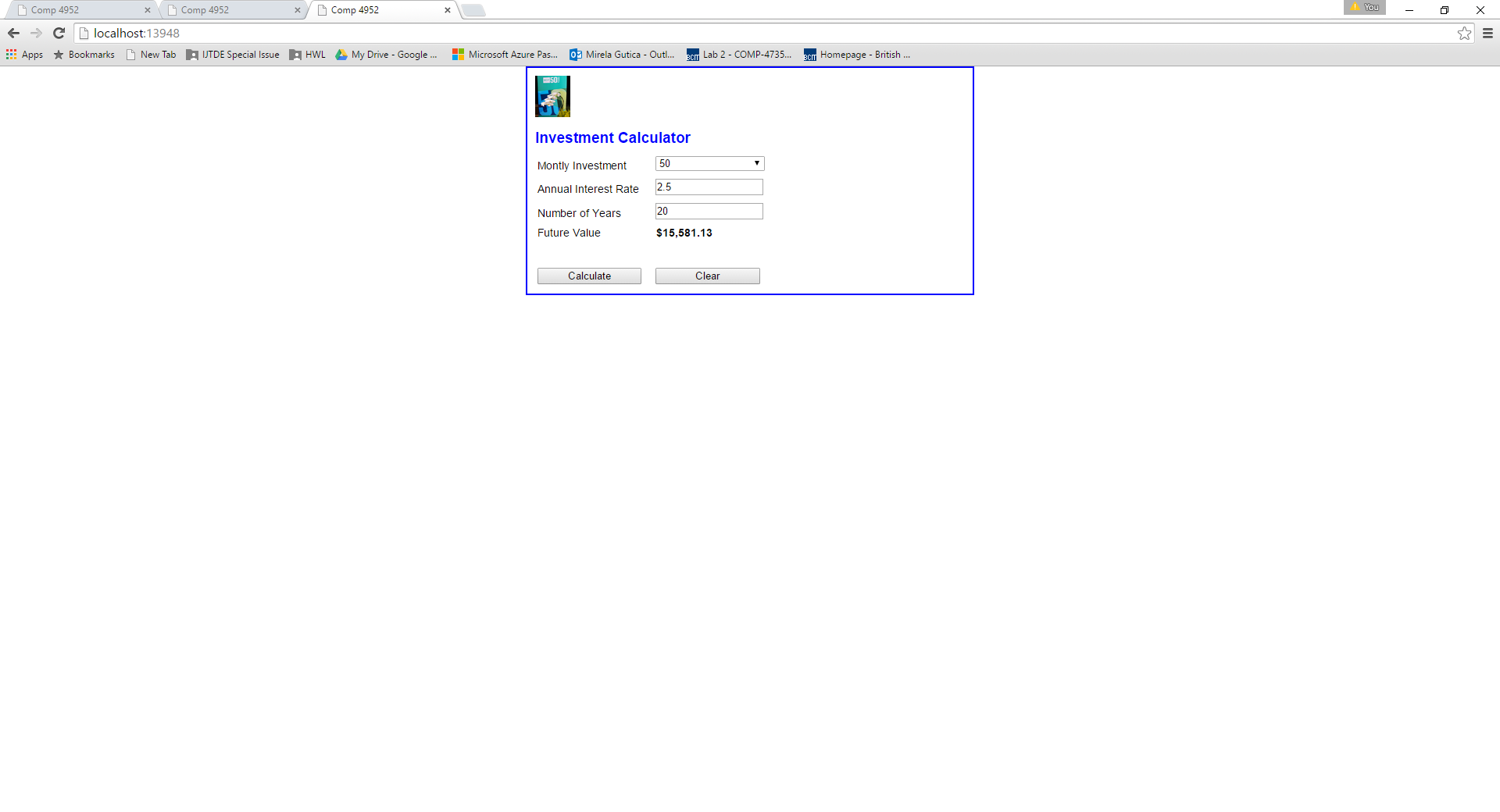
1. Add functionality to your web site in code-behind:
   * 1. In the Page\_Load event load the monthly investment choices.
     2. In the ButtonCalculate event add your investment calculations.
     3. In the ButtonClear clear the content.
2. Add validation.
   1. Add field validators to your textbox controls.
      1. Add range validators
         1. Set up the relevant properties in the properties window: display (dynamic), error message, control to validate, range, type, id.
      2. Add field required validators
         1. Set up the relevant properties in the properties window: display (dynamic), error message, control to validate, id.



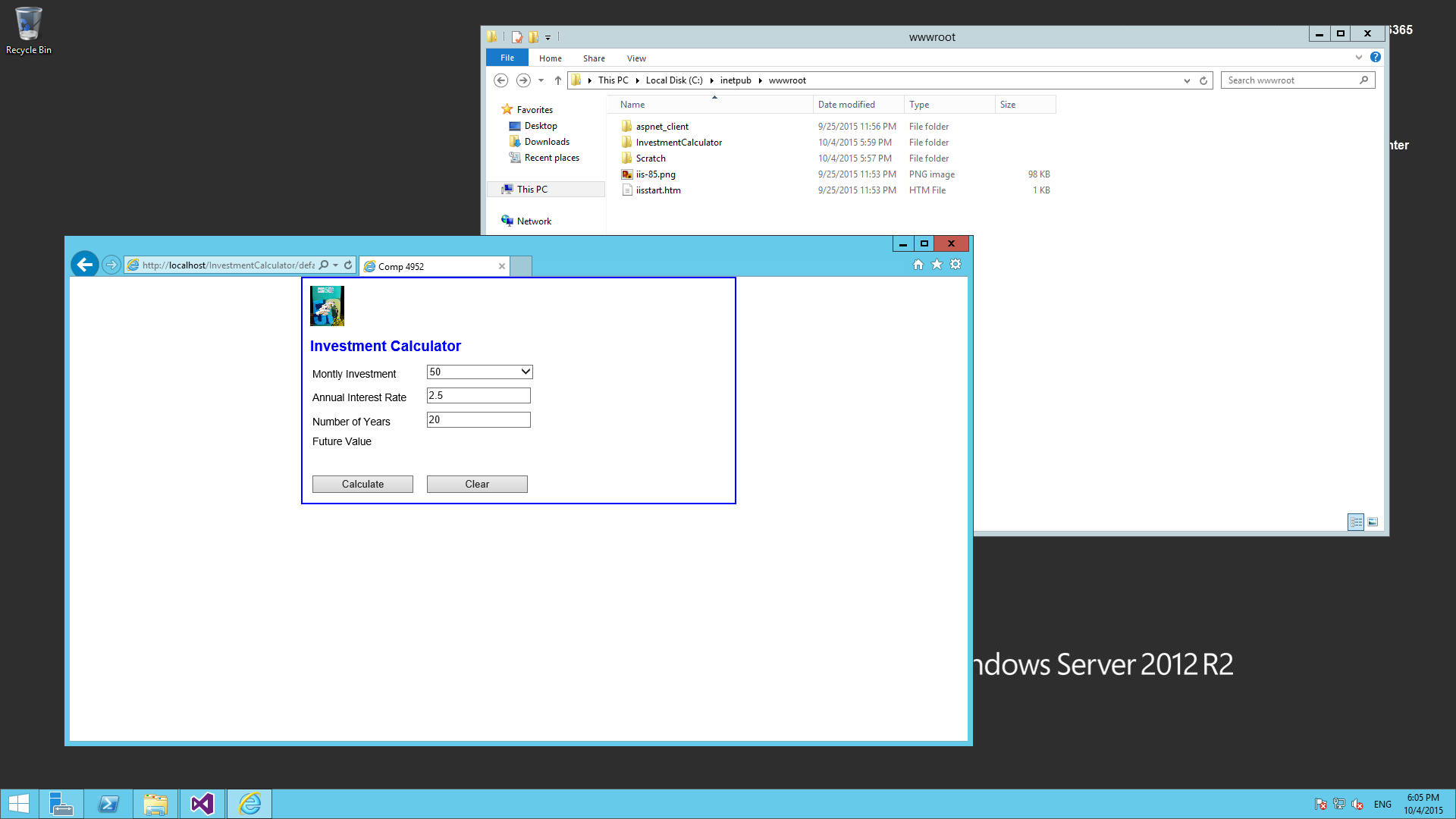
1. Add CSS style.
   1. Add the CSS style of your choice such that the interface design is improved. Add the corresponding CSS class to each element of the interface.







1. Test your application.
2. Upload(copy in your azure virtual server) the folder with the your website under your directory c:\inetpub\wwwroot
3. To test your website functionality use: <http://localhost/Name_of_folder/default.aspx>



Answer the questions:

1. **What is happening when a user requests a page from a remote server?**
2. **Is it any client-side code executing in your assignment application? If so, how is the client-side code executed in ASP.NET?**
3. **What is the role of the line:**

**<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="\_Default" %>**

1. **What is the role of a default web form?**
2. **What special folders are in ASP.NET and what is their role?**
3. **What is the difference between web server controls and HTML controls?**
4. **What is the role of property IsPostBack and how should be used?**
5. **What control validators are available in ASP.NET, how should be used?**
6. **What is the role of isValid and how should be used?**
7. **What is happening when the user hits the refresh button?**