

# Chapter 3

## ASP Web Page with Controls

---

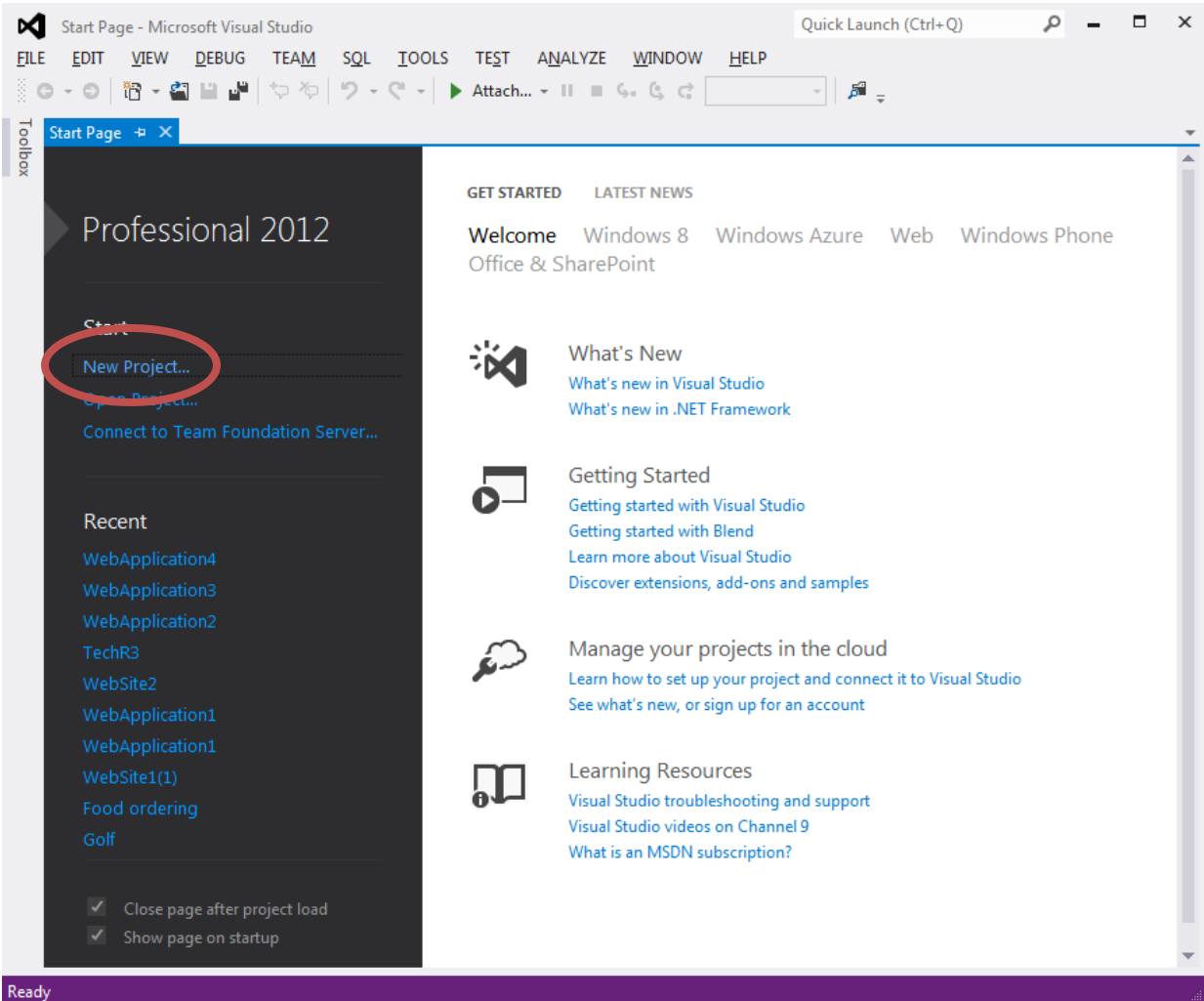
**In this chapter, you will learn how to use the following ASP.Net Application functions to World Class standards:**

- **Opening ASP.Net Editor**
- **Using the Split Screen with a Web Application**
- **Renaming the Web Form**
- **Insert a HTML Title**
- **Insert a Header into the Web Page**
- **Inserting a Paragraph into the Web Page**
- **Using Floating Division Tags**
- **Inserting Labels in Another Table inside a Division**
- **Add a Table**
- **Adding Code for the Labels**
- **Save the Web Page**
- **View the Web Page**

## Open the ASP.Net Editor

To open the ASP.Net Web Application Editor in Microsoft Visual Studio is essential to creating dynamic web pages with controls. In this version of the World Class CAD – ASP.Net, we are using Visual Studio 2012.

To open a new project, we select New Project on the left side of the Start Page.



**Figure 3.1 – The Start Page**

We start a new Windows Application Project by picking the Windows under Visual C # in the left pan of the New Project window. Then we pick ASP.Net Empty Web Application in the center pane.

At the bottom of the Window, we name the project, Address Book. We make a folder for our work called ASP Dot Net on the desktop, on our flash drive or in the Documents folder. We make another folder called Project. On the New Project window, we browse to the Project

folder location. The solution name is the same as the project name.

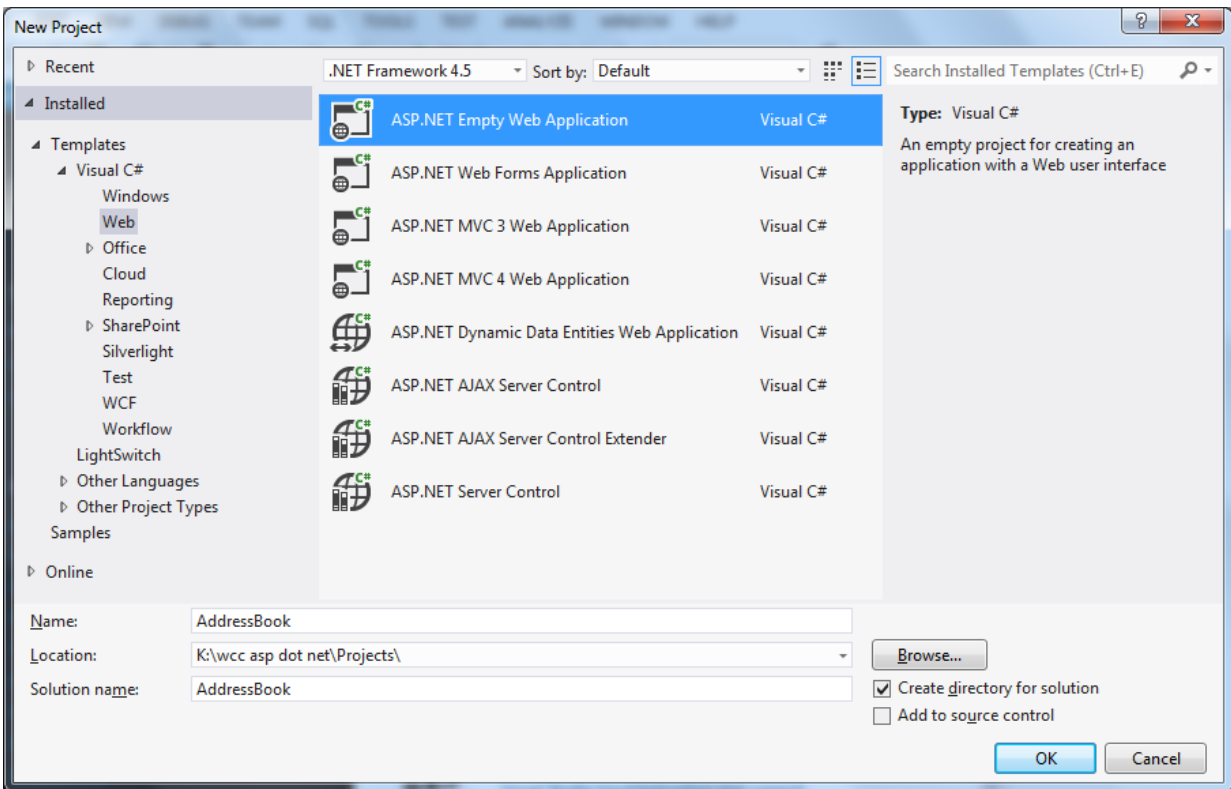


Figure 3.2 – New Project

The Address Book web project opens.

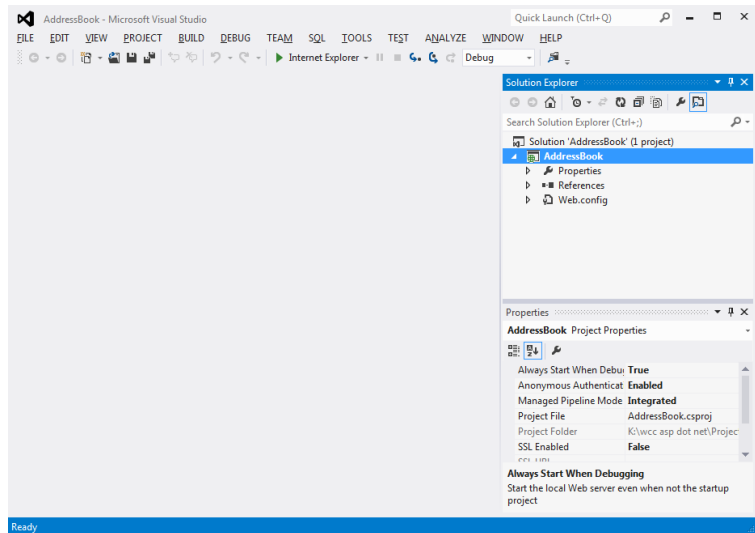


Figure 3.3 –The Address Book Program

Next, we will add a new item, which we will call webform1. On the main menu, select Project and choose Add New Item from the drop down menu.

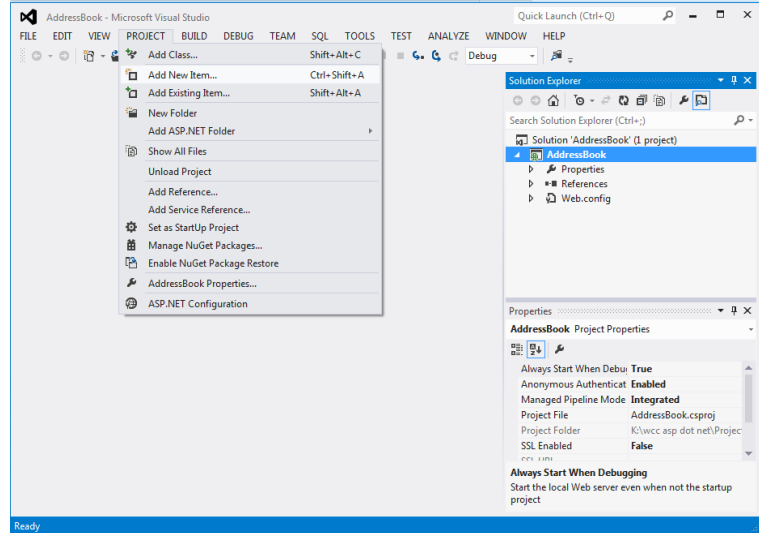


Figure 3.4 –Add a New Item: A Webform

The Add New Item window will appear and we will select New Form from the center list. The default name for the web form is WebForm1.aspx. We will type in address\_book.aspx for the file name. Then we click on the Add button.

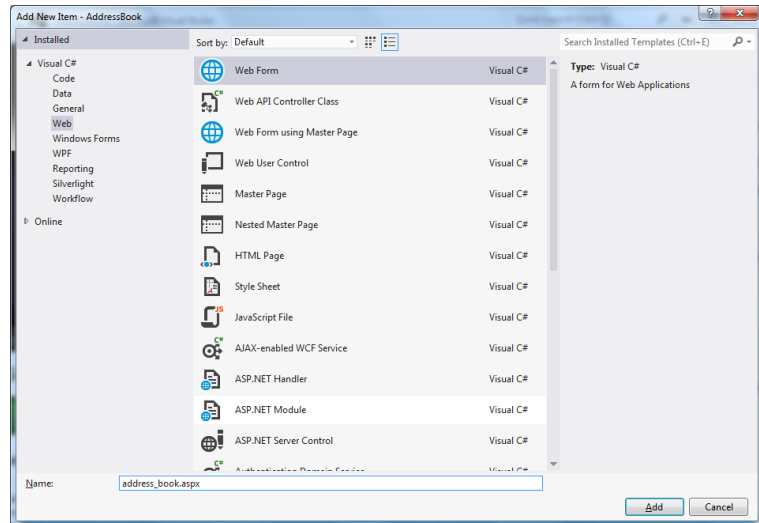


Figure 3.5 –Add a New Item Window

In the Solution Explorer on the right side of Visual Studio, we can see the new ASP.Net web page called address\_book.aspx. We should highlight the file and we can see the HTML source code in the left pane.

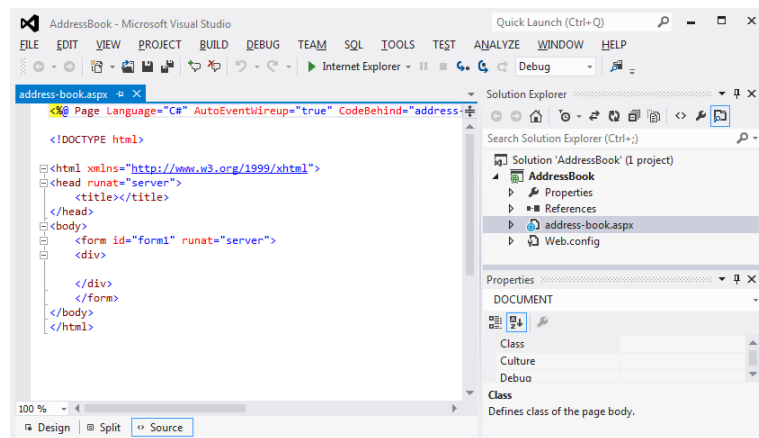


Figure 3.6 –WebForm1.aspx

## Using the Split Screen with a Web Application

For graphical presentations such as web pages, we will want to look at the design the same time that we see the HTML code. At the bottom of the page, we can see the Design – Split – Source options. We will choose the Split button to allow us to see both the source code and the design at the same time.

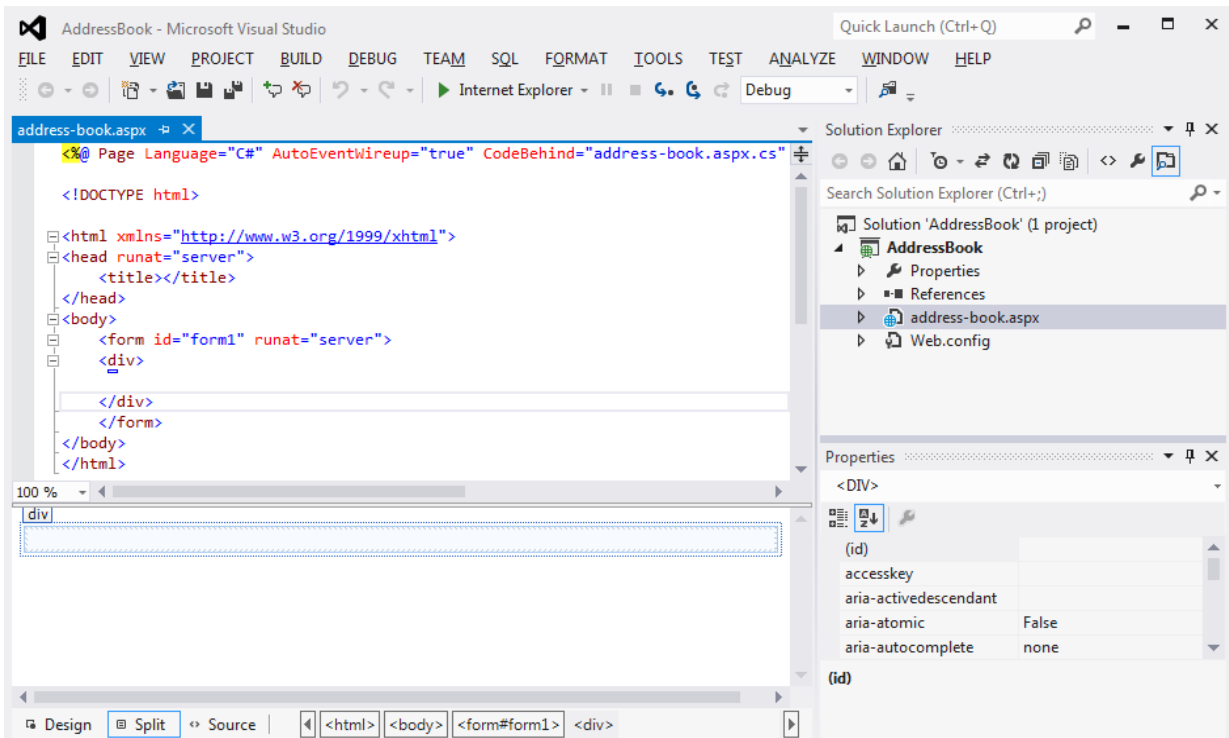


Figure 3.7 – The Split Screen to See Both Design and the Source Code

## Renaming the Web Form

We will type Address Book between the title tag <title> and close title tag </title>.

**<title>Address Book</title>**

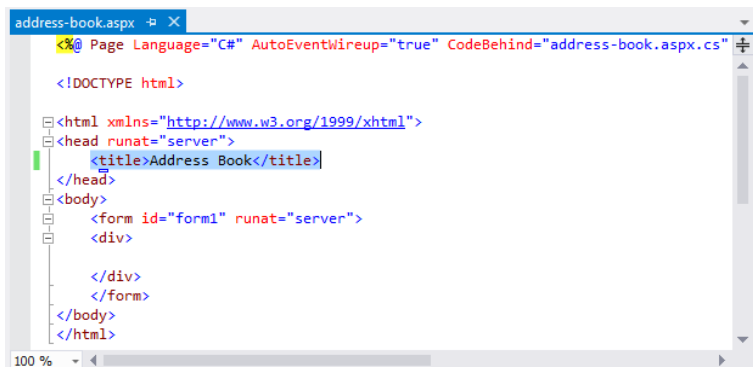


Figure 3.9 – Web Page Title Source Code

## Inserting a Header into the Web Page

We will add a division tag `<div>` right after the body tag `<body>`. And in between the division tags, we will insert a header1 tag `<h1>` with the text Address Book and the close h1 tag `</h1>`.

```
<body>
  <div>
    <h1>Address Book</h1>
  </div>
```

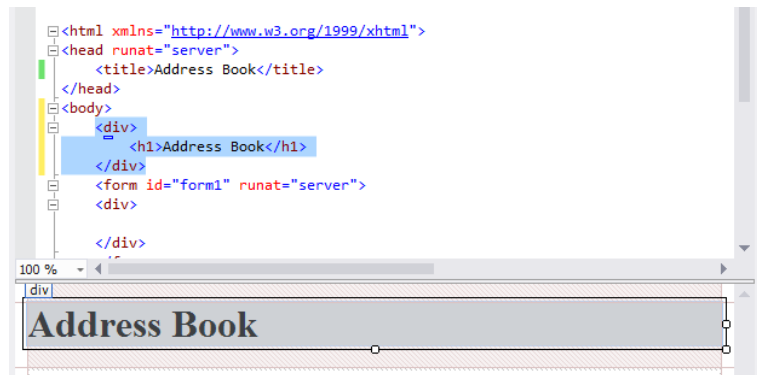


Figure 3.10 – Placing a Label on the Form

We will add a style tag `<style>` right after the `!DOCTYPE html` tag. And in between the style tags, we will insert a definition for the font-family and font-size for the body of the web page and then close the style tag `</style>`.

```
<style>
  BODY
  {
    font-family: Arial;
    font-size: 10pt;
  }
</style>
```

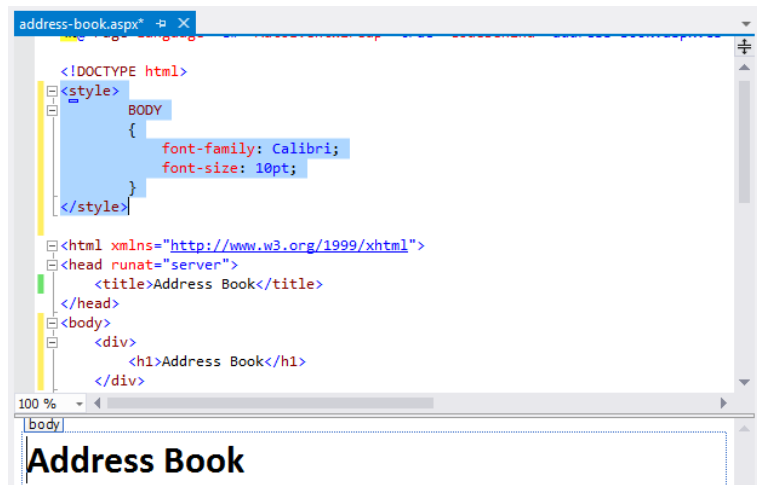


Figure 3.11 – Cascading Style Sheet

## Inserting a Paragraph into the Web Page

In between a new set of division tags and after the h1 close tag `</h1>`; we will add a paragraph tag `<p>` and the close p tag `</p>`. And, we will type the text as shown below.

```
<div>
  <p>Add your Name, Address, City, State, Zip code, Phone, Email and press the send button</p>
</div>
```



Figure 3.12 – Paragraph Tag

## Using Floating Division Tags

We then add a division tag `<div>` followed by the style definition that will set the width at 300 pixels and float to the left.

```
<div style="width:300px;float:left;">
```

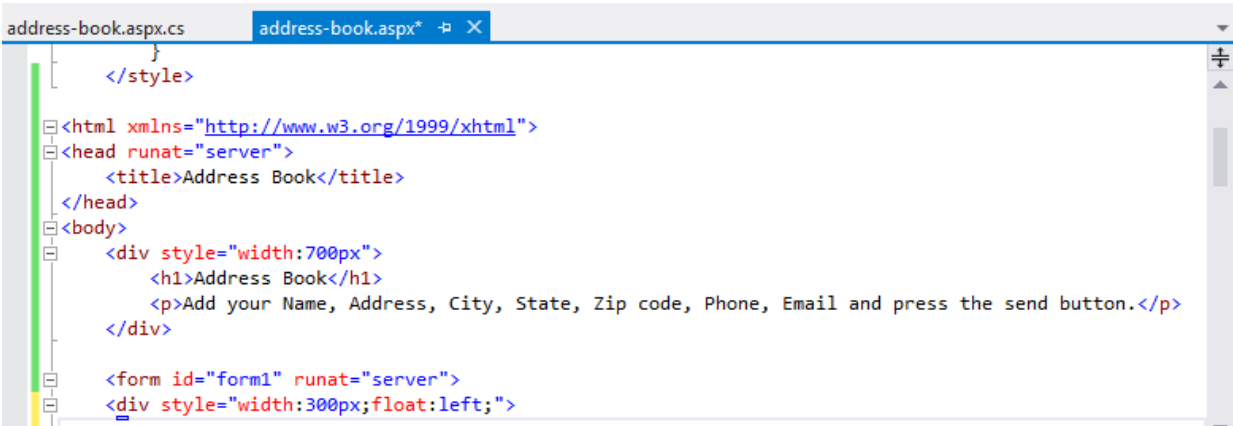


Figure 3.13 – Insert a Division that is 300 Pixels Wide

## Add a Table

The next section of the web page is the textboxes that will contain the Name, Address, City, State, Zip code, Phone and Email to input. This comes after we inserted a division tag that is 300 pixel wide and floats to the left. Inside of the div tag is a HTML5 table. In the first row, we will have the tr tag `<tr>`. In the first cell, the td tag `<td>`, we will type Name and close the td tag `</td>`.

In the next cell `<td>`, we will click on the Toolbox on the left side of the Visual Studio and then double click on Textbox tool. The `asp:textbox` code will be inserted. Close the cell with `</td>`. Make the textbox 150 pixel wide name it `txtName`.

We will insert a total of seven textboxes and close the table.

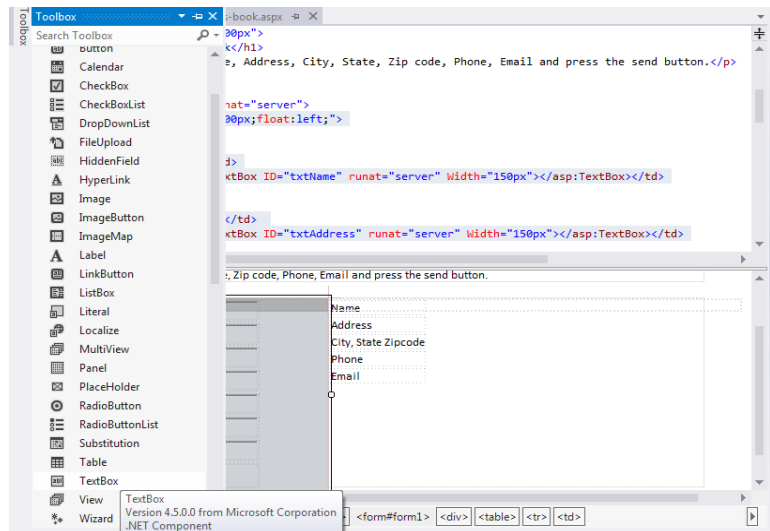


Figure 3.14 – Insert a Textbox

The code for the seven textboxes in the table follows:

```
<div style="width:300px;float:left;">
  <table>
    <tr>
      <td>Name</td>
      <td><asp:TextBox ID="txtName" runat="server" Width="150px"></asp:TextBox></td>
    </tr>
    <tr>
      <td>Address</td>
      <td><asp:TextBox ID="txtAddress" runat="server" Width="150px"></asp:TextBox></td>
    </tr>
    <tr>
      <td>City</td>
      <td><asp:TextBox ID="txtCity" runat="server" Width="150px"></asp:TextBox></td>
    </tr>
    <tr>
      <td>State</td>
      <td><asp:TextBox ID="txtState" runat="server" Width="150px"></asp:TextBox></td>
    </tr>
    <tr>
      <td>Zipcode</td>
      <td><asp:TextBox ID="txtZipcode" runat="server" Width="150px"></asp:TextBox></td>
    </tr>
    <tr>
      <td>Phone</td>
      <td><asp:TextBox ID="txtPhone" runat="server" Width="150px"></asp:TextBox></td>
    </tr>
    <tr>
      <td>Email</td>
      <td><asp:TextBox ID="txtEmail" runat="server" Width="150px"></asp:TextBox></td>
    </tr>
    <tr>
      <td><asp:Button ID="Button2" runat="server" Text="Submit" OnClick="Button2_Click" /></td>
      <td></td>
    </tr>
  </table>
</div>
```



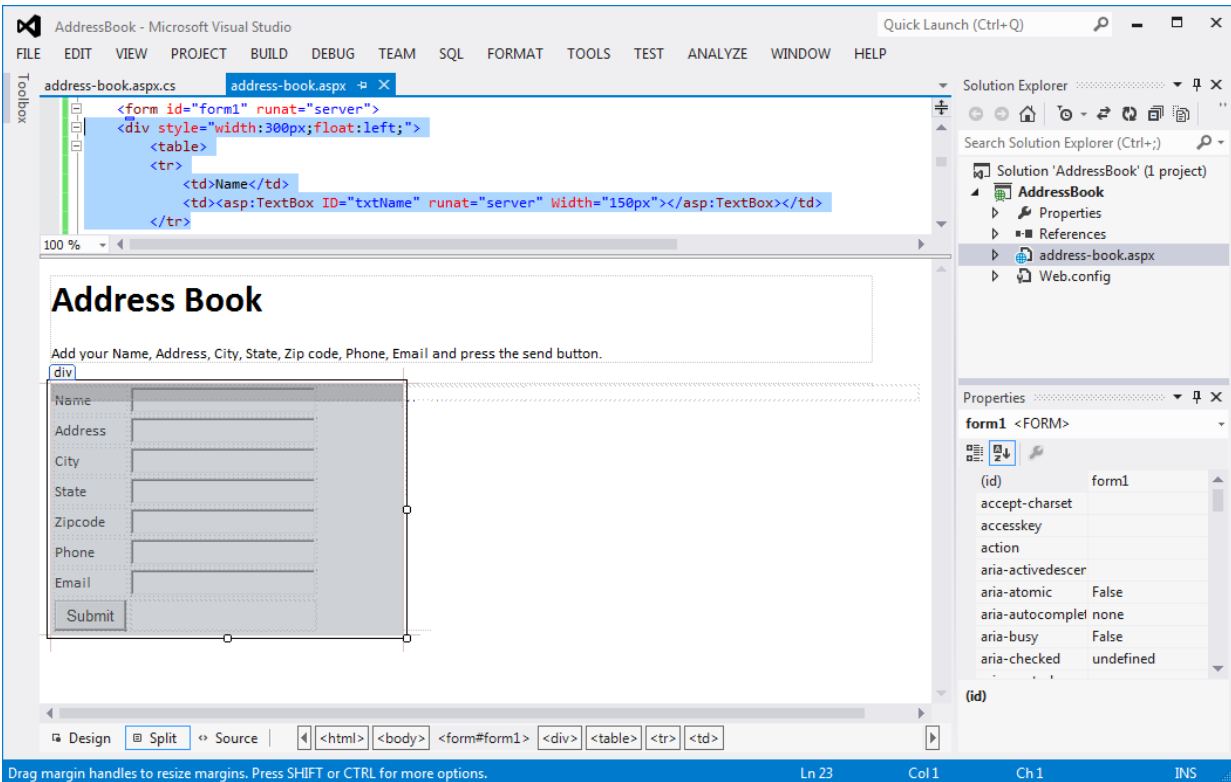


Figure 3.15 – Add a Table and Textboxes

## Inserting Labels in Another Table inside a Division

The next section of the web page is the labels that will contain the Name, Address, City, State, Zip code, Phone and Email that was inputted. This comes after we inserted a division tag that is 400 pixel wide and floats to the left. Inside of the div tag is a HTML5 table. In the first row, we will have the tr tag <tr>. Inside the row, we will insert the td tag <td>.

In the cell <td>, we will click on the Toolbox on the left side of the Visual Studio and then double click on Label tool. The asp:textbox code will be inserted. Close the cell with </td>.

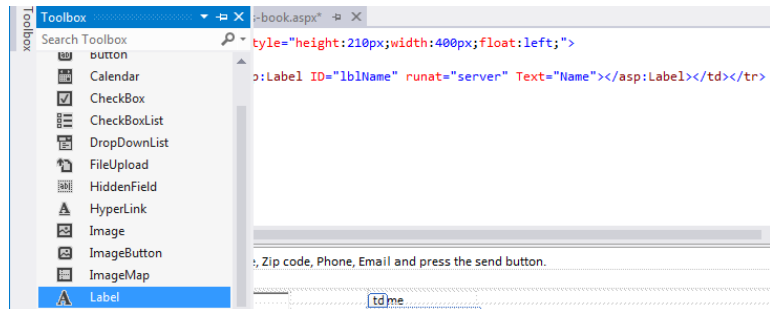
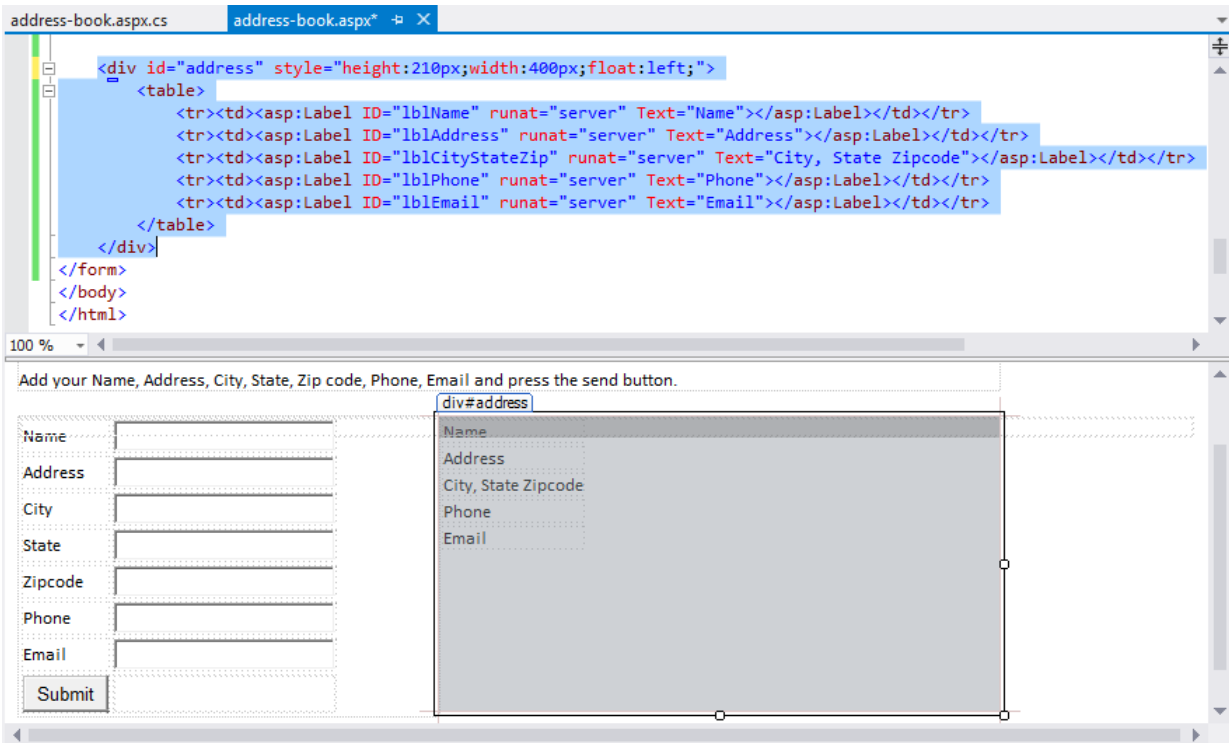


Figure 3.16 – Add a Label



**Figure 3.17 – Add a Table and Textboxes**

The code for the seven labels in the table follows:

```

<div id="address" style="height:210px;width:400px;float:left;">
  <table>
    <tr><td><asp:Label ID="lblName" runat="server" Text="Name"></asp:Label></td></tr>
    <tr><td><asp:Label ID="lblAddress" runat="server" Text="Address"></asp:Label></td></tr>
    <tr><td><asp:Label ID="lblCityStateZip" runat="server" Text="City, State Zipcode"></asp:Label></td></tr>
    <tr><td><asp:Label ID="lblPhone" runat="server" Text="Phone"></asp:Label></td></tr>
    <tr><td><asp:Label ID="lblEmail" runat="server" Text="Email"></asp:Label></td></tr>
  </table>
</div>

```

## Adding Code for the Labels

We then double click on the Submit button and the address-book.aspx.cs tab will appear. We see the Button2\_Click event code. We will type the following:

```

protected void Button2_Click(object sender, EventArgs e)
{
  lblName.Text = txtName.Text;
  lblAddress.Text = txtAddress.Text;
  lblCityStateZip.Text = txtCity.Text + ", " + txtState.Text + " " + txtZipcode.Text;
  lblPhone.Text = txtPhone.Text;
  lblEmail.Text = txtEmail.Text;
}

```

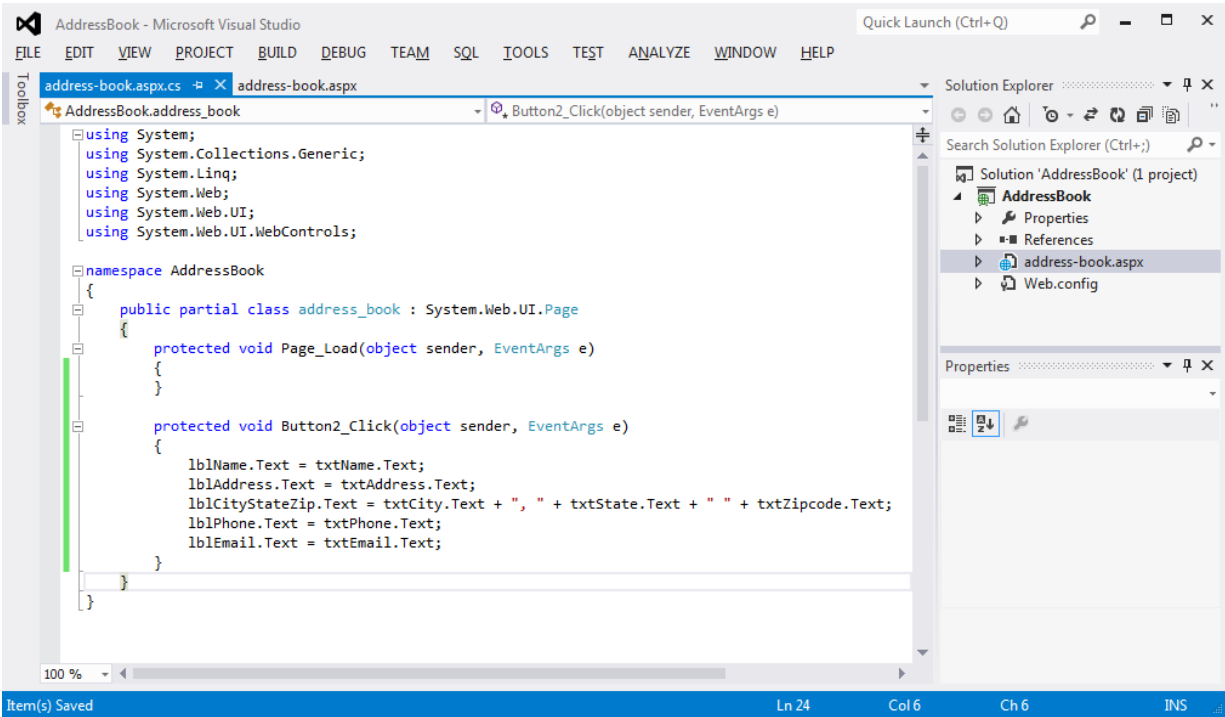


Figure 3.18 – Submit Event Code

## Saving the Web Page

To save the web page, we choose File from the main menu and then click on Save All.

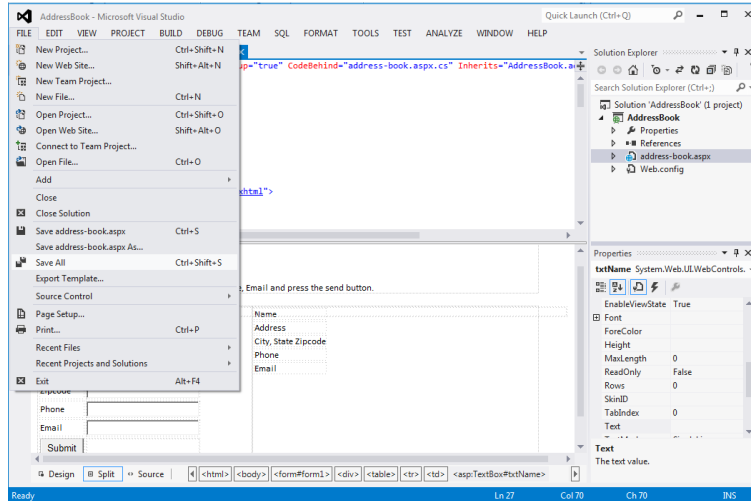


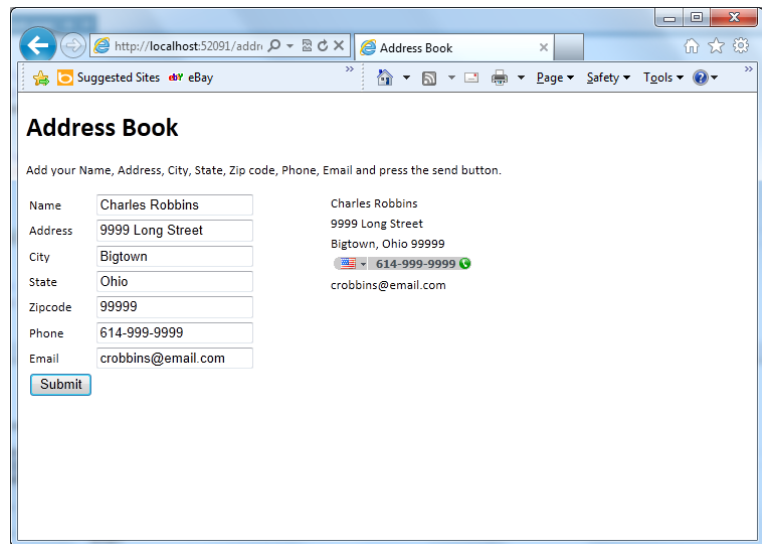
Figure 3.19 – Saving the Web Page

## Viewing the Web Page

---

After noting that the web page is saved, press the F5 to launch the Address Book web page in the web browser. The Address Book web page will appear in the browser as shown in Figure 3.20.

Type in the information and press the Submit button. We can check the information in the address to the right.



**Figure 3.20 – Launching the Web Page**

If our program does not function correctly, go back to the code and check the syntax against the program shown in this chapter. Repeat any processes to check or Beta test the program. When the program is working perfectly, save and close the project.

There are many variations of this ASP.Net Application we can practice and obtain information from a personal computer. While we are practicing with forms, we can learn how to use variables, strings and comments. These are skills that we want to commit to memory.

**\* World Class CAD Challenge 191-2 \* - Write an ASP.Net Web Application that displays a single header, allow the user to type in their name, address, city, state, zip code, phone and email. Then when the button is pressed, the web page will show the information.**

**Continue this drill four times using some other form designs, each time completing the ASP.Net Project in less than 1 hour to maintain your World Class ranking.**