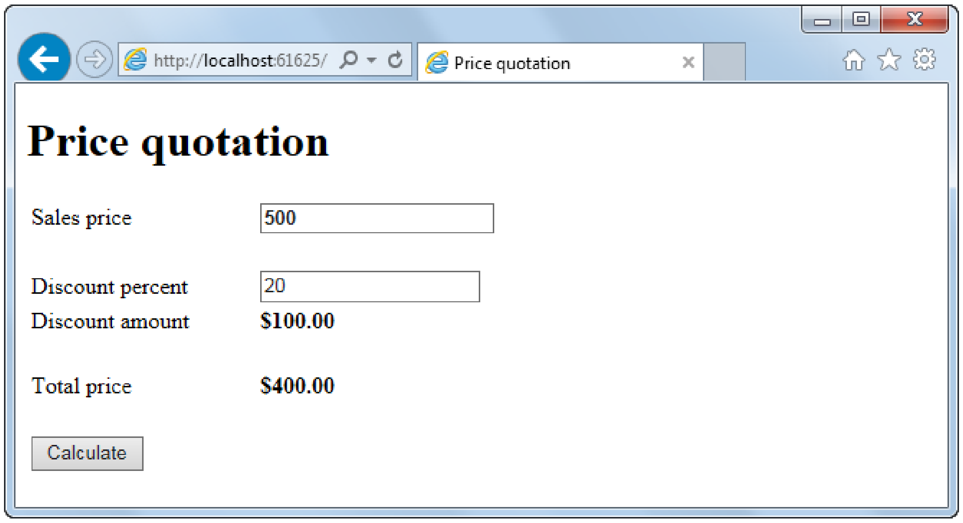
Part 1 Build the Quotation application

For this exercise, you will build a one-page application with the user interface that's shown below. This time, the user will enter the sales price and discount percent, and the application will calculate and display the discount amount and total price. If a user entry is invalid, an error message will be displayed to the right of the related text box.



Price quotation

Sales price	500
Discount percent	20
Discount amount	**$100.00**
Total price	**$400.00**

Calculate

Start a new web application and build the form

1. Start a new web application named XEx02Quotation in your exercises_extra folder.
2. Add a web form with the name Default.aspx. Type "Price quotation" inside the title tags in the head section of the document, and delete the default div element in the body section.
3. Add an h1 heading to the form element with the text shown above.
4. Add a table to the form below the heading with 8 rows and 3 columns. (The third column will be used for validation controls.)
5. Add the text and button shown above to the cells in the first, third, fourth, sixth, and eighth rows of the first column.
6. Adjust the width of the table by dragging its right handle, and adjust the widths of the columns by dragging their right borders so the widths are similar to what's shown above. (It may take some time to get this right, and the form may look different in Design view than it does when you run the application.)
7. Add text boxes to the first and third rows in the second column, and add label controls to the fourth and six rows in the second column.
8. Use the Properties window to set appropriate IDs for the controls and to format the first text box and the two labels with bold type. Then, use the Properties window to set the text for the button to "Calculate".
9. Test this form to see how it looks in a browser, and make whatever adjustments are necessary.

Add the C# code for the form

10. Create an event handler for the Click event of the Calculate button. This handler should calculate the discount amount and total price and display them in currency format as shown above.
11. Test this form to see whether it works correctly, and make whatever corrections are necessary.

Add validators for the text boxes

12. Add a required field validator in the column to the right of each text box that tests whether an entry has been made in the text box. If an entry hasn't been made, "Required" should be displayed.
13. Add a range validator in the column to the right of each text box. The one for the Sales Price should test to see whether the entry is between 10 and 1000, and an appropriate message should be displayed if it isn't. The one for the Discount Percent should test whether the entry is between 10 and 50, and an appropriate message should be displayed if it isn't.
14. If necessary, adjust the C# code for the page so it only does the calculations if the entries are valid.
15. Run the application, and see that it throws an error because it's using unobtrusive validation but can't find the jquery library.
16. Turn off unobtrusive validation for the page. Then, run and test the application, and make whatever corrections are necessary.

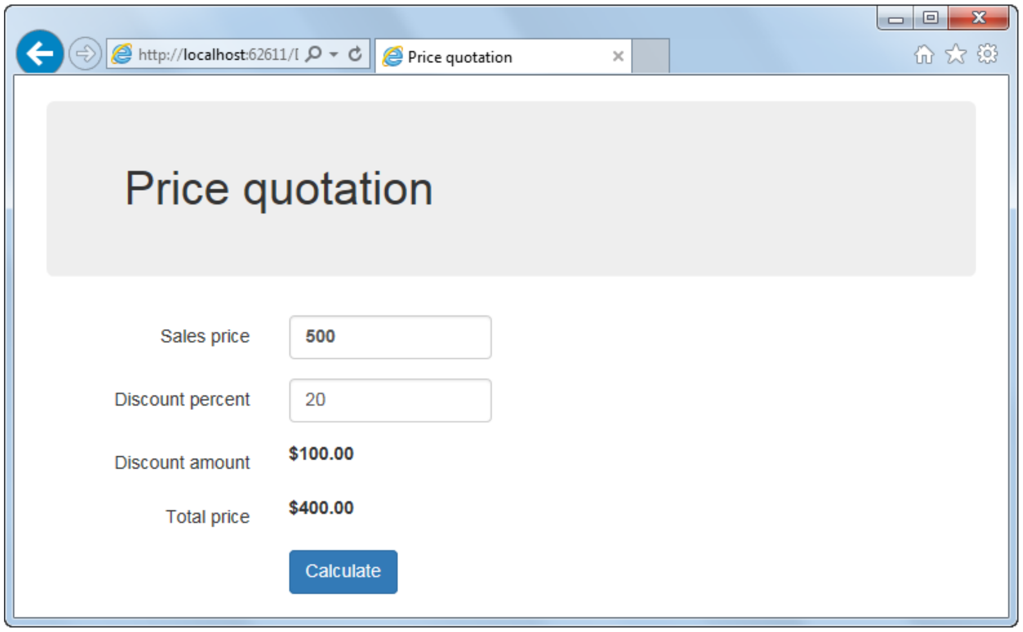
Make any final adjustments

17. Take a final look at the application, and make any adjustments for improving the look of the application, the operation of the application, or the clarity and logic of the code.

Part 2 Use CSS and Bootstrap to format the Quotation application

The application for this exercise is the same as the one for exercise 2-1, except that the formatting is done with CSS and Bootstrap instead of a table. When you're done, the application will look like the one below.

For this exercise, you'll start a new web application and add Bootstrap to it. Then, you'll add the Default.aspx file and its code-behind file from the application for exercise 2-1. Finally, you'll work in Source view to modify the Default.aspx file so the web application uses Bootstrap and custom CSS for formatting.



Price quotation

Sales price	500
Discount percent	20
Discount amount	$100.00
Total price	$400.00

Start a new web application and work with an existing file

1. Start a new web application named XEx03Quotation in your exercises_extra folder.
2. Add the Bootstrap NuGet package to your application.
3. Add the Default.aspx file from your solution for extra exercise 2-1 to this web application. Then, display the page in Source view and delete the entire Style element in the head section of the document.
4. Change the opening table tag from table to main. Then, remove the tags for the rows and columns of the table, but keep the aspx code for the text box controls, label controls, validators, and button.
5. Add HTML label elements that identify the server controls for the text boxes and the label controls that display the results. Then, delete the Font-Bold properties for the first text box and the two label controls.
6. Right-click the Default.aspx file and select the View in Browser command. Review how the application looks now, and then leave the browser running.

Set up the head section of the document to work with CSS and Bootstrap

7. Add the following viewport metatag to the head section:

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

8. Add a new style sheet named site.css to the Content folder.
9. Drag the Bootstrap style sheet and the new site.css file from the Content folder to the head section to create link tags. Remember that site.css needs to come after the Bootstrap style sheet.
10. Drag the Bootstrap and jquery JavaScript files from the Scripts folder to the head section to create script tags. Remember that the Bootstrap script tag needs to come after jquery script tag.

Use the Bootstrap grid CSS classes to lay out the page

For the remainder of this exercise, you should view your changes after each step to be sure they worked correctly. To do that, you can save the changes in Visual Studio, switch to the running browser, and then refresh the browser.

11. Set the class attribute of the form element to "form-horizontal", and change the class attribute of the main element to "container". Then, move the h1 element so it's inside the main element, and set its class attribute to "jumbotron".
12. Add a div element after the h1 element, and set its class attribute to "form-group". Then, use the Bootstrap column CSS classes to make the sales price label span 3 columns, the sales price text box span 3 columns, and the sales price validation controls span 6 columns. Use nested div tags as needed to make everything line up the way you want it to.
13. Add form-group div elements for the discount percent, discount amount, and total price fields, and put the fields in columns like the ones for the sales prive fields.
14. Add a form-group div for the calculate button, and offset it 3 columns so it lines up with the text boxes and label controls above it.

Use Bootstrap CSS classes for additional formatting

15. Add the "control-label" class to the HTML label elements that identify the server controls, add the "form-control" class to the text box server controls, and add the "btn" and "btn-primary" classes to the button server control.
16. Add the "text-danger" class to the validation controls. To view the changes, enter invalid data after you refresh the browser.

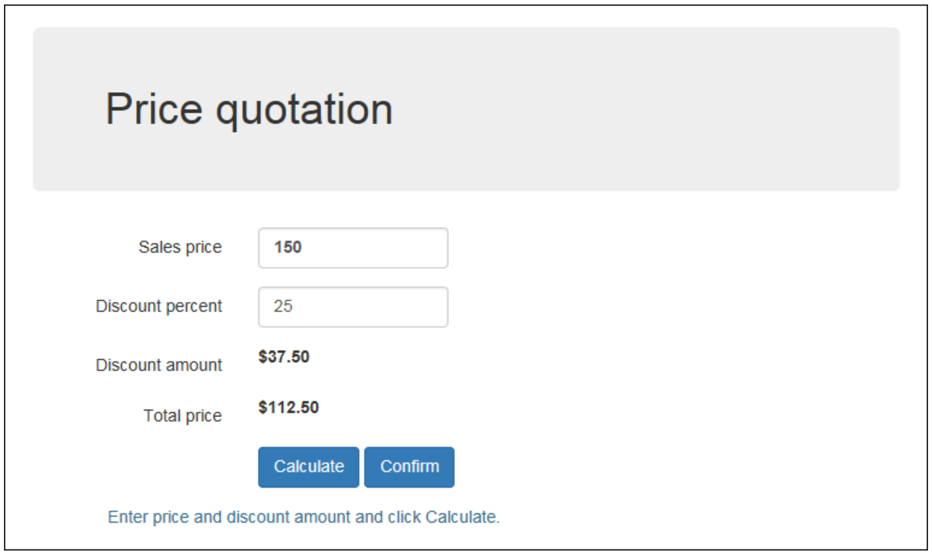
Use custom CSS classes

17. Open the site.css file and change its default type selector from body to label. Then, add a rule that sets the font-weight to normal. This will override the default Bootstrap style of making label elements bold.
18. Still in site.css, code a class selector named .bold. Then, add a rule that sets the font-weight to bold. Finally, add the "bold" class to the first text box and the two ASP.NET label server controls.

Part 3 Enhance the the Quotation application

The application for this exercise is an enhanced version of the one for exercise 3-1. First, the Quotation has a confirm button to the right of the Calculate button. Second, the Confirm button redirects to a Confirmation page.

The Quotation page (Default.aspx)

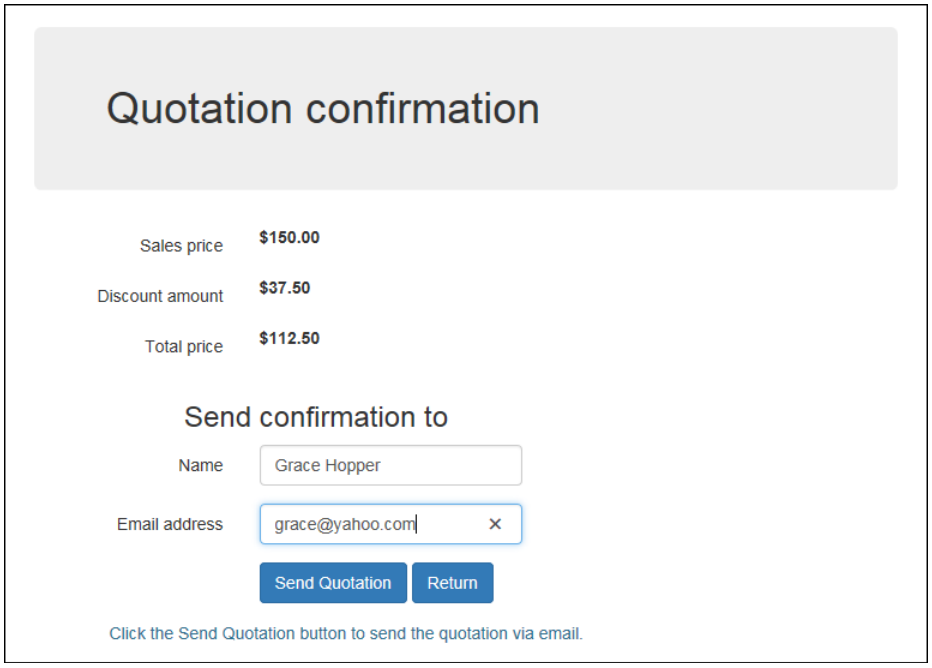


Price quotation

Sales price	150
Discount percent	25
Discount amount	$37.50
Total price	$112.50

Enter price and discount amount and click Calculate.

The Confirmation page (Confirm.aspx)



Quotation confirmation

Sales price	$150.00
Discount amount	$37.50
Total price	$112.50

Send confirmation to

Name	Grace Hopper
Email address	grace@yahoo.com

Click the Send Quotation button to send the quotation via email.

Open the web application for this exercise and start enhancing its pages

1. Open the web application named XEx04Quotation in your exercises_extra folder. It includes the aspx and code-behind files for the pages shown above, but the first page doesn't have the code for the Confirm button and the second page doesn't have the

code for either of the buttons that are shown. And neither page has the code for the label with the message that's displayed below the buttons.

2. Add the Confirm button to the Quotation page right after the Calculate button, and set its CssClass property to the btn and btn-primary classes.
3. Add the Send Quotation and Return buttons to the Confirmation page, and set the appropriate CssClass values. Also, set the properties for the Return button so it goes back to the Quotation page and doesn't cause validation.
4. On each page, add a label control below the buttons. For each label, set the ID to lblMessage, the CssClass to text-info, and the Text as shown above.
5. Test the application to see how it's going, and make any adjustments.

Add the C# code that makes the application work

6. Create a Click event handler for the Confirm button of the Quotation page. This button should redirect to the Confirmation page, which will display the quotation that is being confirmed by getting the data from session state.
7. To make this work, the Click event handler for the Calculate button of the Quotation page should save the sales price, discount amount, and total price in session state. Now, add that code to that event handler.
8. When the user clicks the Confirm button on the Quotation page to go to the Confirmation page, the Load event handler for the Confirmation page should get the data from session state and display it as shown above. Now, add that code to the Load event handler.
9. Add an event handler for the Click event of the Send Quotation button. If the entries for this form are valid, this handler should use the data entered by the user to display a message below the buttons that says: "Quotation sent to <name> at <email>." It should also set the values in session state to null.

Test and refine the operation of the application.

10. Code and test what you have so far. At this point, all four buttons should work, although you may be able to find errors by trying different sequences of button clicks. If, for example, the user clicks on the Confirm button on the Quotation page before clicking on the Calculate button, the application may blow up because there isn't any data in session state.
11. To fix errors like that, add code to the Click event handler for the Confirm button on the Quotation page that tests whether the value for the sales price in session state is null. If it is, this message should be displayed in the label below the buttons: "Click the Calculate button before you confirm." If it isn't, the handler should redirect to the Confirmation page.
12. Do the same test in the Load event handler for the Confirmation page. If the sales price in session state is null, don't display the values on the page. If it isn't, get the values from session state and display them.

Make any final adjustments

13. Take a final look at the application, and make any adjustments for improving the look of the application, the operation of the application, or the clarity and logic of the code.