**How to Use the Code Editor**

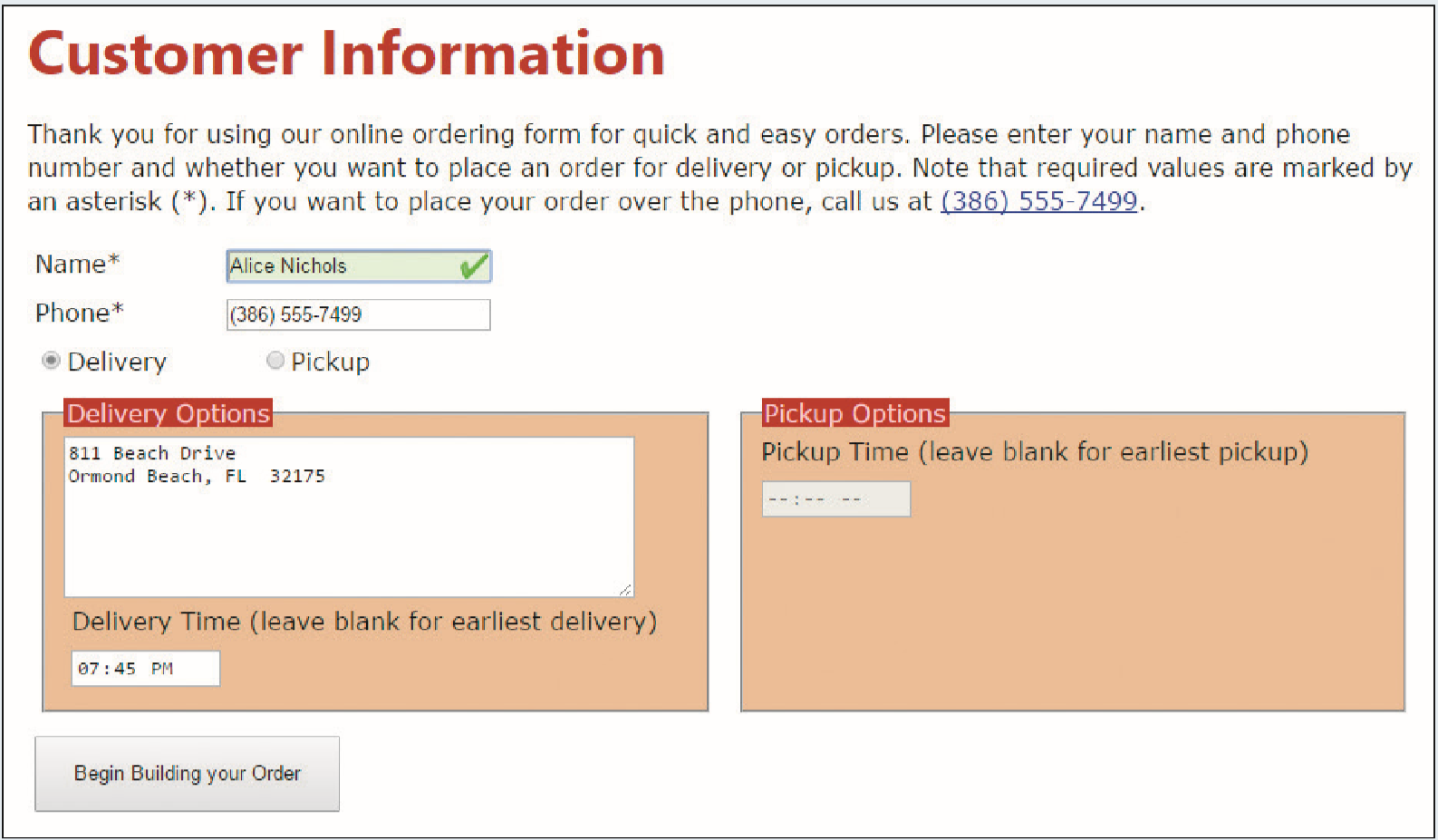
1. Select the "Run Code" button to execute the program.
2. Select the "Calculate Grade" button to generate a score based on the completed tasks.
3. Continue to modify, run, and calculate your code until you are happy with the grade.
4. Select the "Submit" button to turn in the assignment to your instructor. Complete the coding activity. Your instructor will grade this activity manually.

**How to Use the Code Editor**

1. Select on the "Run Code" button to execute the program.
2. Make any necessary modifications to the code.
3. Continue to modify and run your code until you are happy with the output.
4. Select the "Submit" button to turn in the assignment to your instructor.

**Summary**

Alice wants you to start work on an online order form for customers to place orders through the Red Ball Pizza website. The form will span several pages in which customers will specify whether the order is for pickup or delivery and will indicate the toppings they want on their pizza(s). *Figure 7–63* shows a preview of the form customers will use to indicate their delivery option (including an address or pickup and at what time they want their order).

 *Figure 7-63*

Alice has already written some of the HTML code for the web pages and designed many of the style sheets. Your job will be to write the code for the form elements and validation styles.

**Document Setup**

Open *rb\_customer.html*, *rb\_build.html*, and *rb\_validate.css* files and enter your **name** and the **date** in the comment section of each file. Return to the *rb\_customer.html* file and within the document head, insert links to the *rb\_forms2.css* and *rb\_validate.css* files.

Still within the document head, use the script element to link the file to the *rb\_formsubmit2.js* file.

**Create the Form**

Scroll down to the section element and, directly after the initial paragraph, insert a form element that employs the action at the fictional address [*http://www.example.com/redball/customer*](http://www.example.com/redball/customer) using the post method.

Within the form element, insert a div element that encloses a label with the text **Name\*** associated with the nameBox control. Also, within the div element, add an input text box with the ID nameBox, field name custName, and placeholder text **First and Last Name**. Make custName a required field.

**Create the Form Continued...**

Create a second div element in the web form that encloses a label with the text **Phone\*** associated with the phoneBox control. Within the div element, add an input box with the ID phoneBox, field name custPhone, and placeholder text **(nnn) nnn-nnnn**. Make custPhone a required field and have any text entry follow the regular expression pattern: ^\d{10}$|^(\(\d{3}\)\s\*)?\d{3}[\s-]?\d{4}$.

You can copy the regular expression code from the *rb\_regex2.txt*.

Add another div element to the web form containing the following code:

1. Insert an input element to create an option button for the orderType field with the ID delivery. Make the option button checked by default. After the option button, insert a label associated with the delivery control containing the text **Delivery**.
2. Add an input element to create a second option button for the orderType field with the ID pickup, followed by a label associated with the pickup control containing the text **Pickup**.

Next within the form, create a field set with the ID deliveryInfo. Within this field set, add the following:

1. A legend containing the text **Delivery Options**.
2. A text area box with the ID addressBox and field name of delAddress containing the placeholder text **Enter delivery address**.
3. A label containing the text **Delivery Time** (leave blank for earliest delivery) associated with the delBox control.
4. Add an input element with the ID delBox and field name delTime for storing delivery time values. Use a data type of time for the control.

**Create the Form Continued...**

Next within the web form, create a field set with the ID pickupInfo containing the following information for pickup orders:

1. A legend containing the text **Pickup Options**.
2. A label containing the text **Pickup Time** (leave blank for earliest pickup) associated with the pickupBox control.
3. Add an input element with the ID pickupBox and field name pickupTime for storing time values. Add the disabled attribute to the tag to disable this control when the form is initially opened. Use a data type of time for the control.

Finally, within the form, add a div element containing a submit button displaying the text **Begin Building your Order**.

**Validation Styles**

Open the *rb\_validate.css* file and add validation styles for the web form. Within the "Validation Styles" section, add the following style rules:

1. A rule that displays input, select, and textarea elements that have the focus with a background color of **rgb(255, 255, 180)**.
2. A rule that displays the nameBox and phoneBox controls that have the focus and contain valid data with a background color of **rgb(220, 255, 220)** and the background image file *rb\_okay.png* at the right with no tiling contained within the background.
3. A rule that displays the nameBox and phoneBox controls that have the focus and invalid data with a background color of **rgb(255, 230, 230)** and the background image file *rb\_warning.png* at the right with no tiling contained within the background.

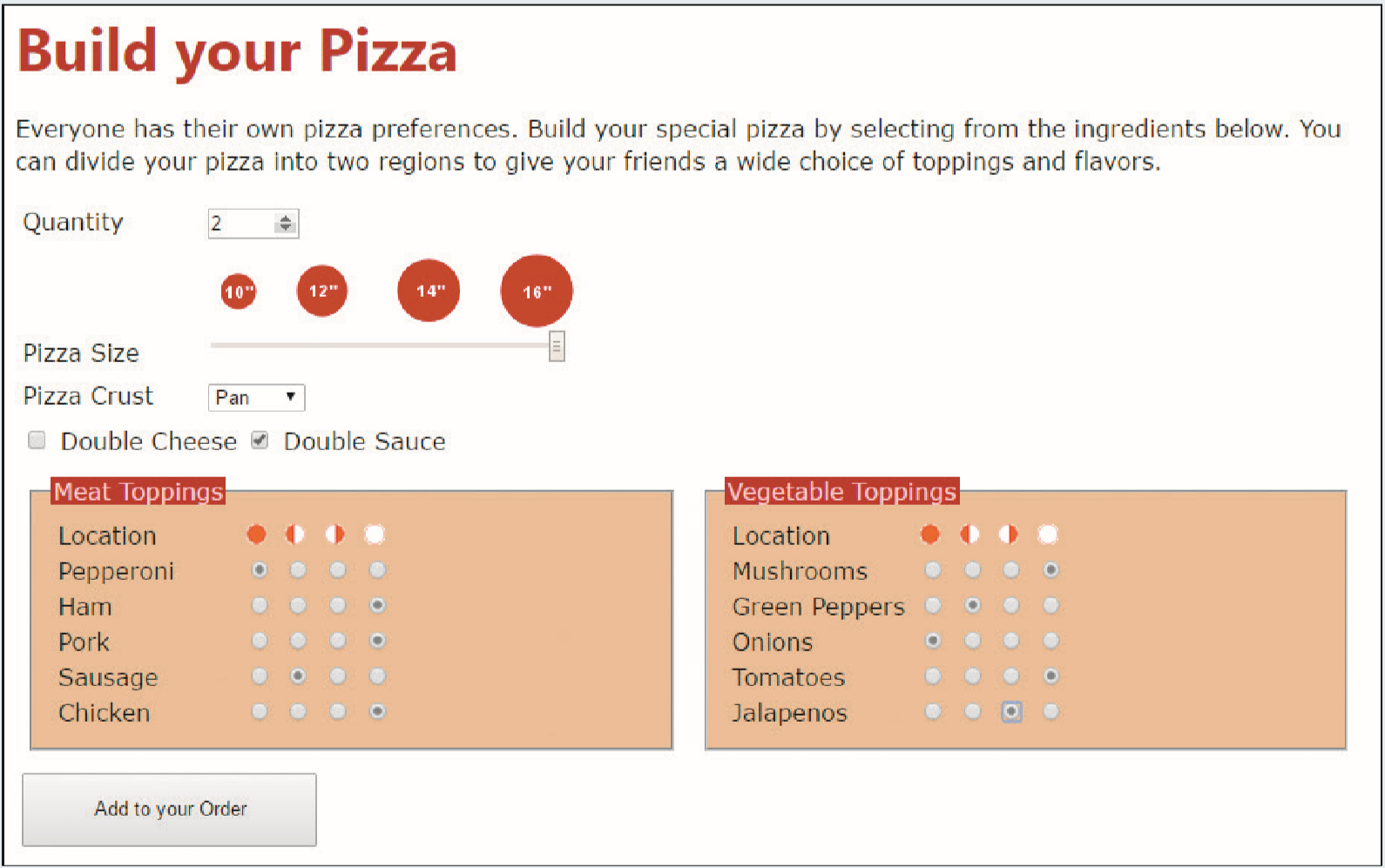
**Verify Form Behavior and Styles**

Open the *rb\_customer.html* file in the browser preview. Verify the following:

1. The content and the layout of the form resemble the form shown in *Figure 7–63*.
2. If you submit the form by clicking the **Begin Building your Own** button with no customer name or phone number, the browser warns you of the missing values.
3. As you enter text into the custName field, the input box background changes to show that the field value is valid.
4. When you enter a phone number into the custPhone field, the input box provides inline validation to indicate whether a valid phone number has been entered.
5. When you click the submit button for a successfully completed form, the browser displays the alert message that the form data passes the initial validation test.

The script file used with this web page is written to enable only either the delivery option or the pickup option but not both.

Next, you will create a form that customers will use to build their customized pizzas. A preview of the form is shown in *Figure 7–64*.

 *Figure 7-64*

**Document Setup**

Return to the *rb\_build.html* file and insert a link to the *rb\_forms2.css* file and add a script element to link the file to the *rb\_formsubmit2.js* file. Scroll down to the section element, insert a form element below the paragraph element that employs the action at the fictional address http://www.example.com/redball/build using the post method.

**Create the Pizza Customization Form**

Within the form element, add a div element containing the following elements:

1. A label with the text **Quantity** associated with the quantityBox control.
2. A spinner control with the ID quantityBox and the field name **pizzaQuantity**. Have the value of the field range from **1** to **10** with a default value of **1**.

Add a second div element that displays images of the pizza sizes, containing the following elements:

1. An inline image set to the image file *rb\_sizes.png*.
2. A label with the text **Pizza Size** associated with the sizeBox control.
3. A range slider with the ID sizeBox and the field name pizzaSize ranging from **10** to **16** in steps of **2** with a default value of **14**.

**Create the Pizza Customization Form Continued...**

Add a div element that provides the selection of pizza crusts containing the following:

1. The label **Pizza Crust** associated with the crustBox control.
2. A selection list for the pizzaCrust field with the ID crustBox and containing the following option values and text: **Thin**, **Thick**, **Stuffed**, and **Pan**.

Add a div element containing a check box with the ID cheeseBox for the doubleCheese field followed by the label **Double Cheese** associated with the cheeseBox control. Then, add a second check box with the ID sauceBox for the doubleSauce field followed by the label **Double Sauce** also associated with that check box.

**Meat Topping Options**

Customers can choose what to place on their pizzas. Create a field set containing the legend **Meat Toppings**. Add the following content to the field set:

1. A div element containing the label Location but not associated with any form control. Next to the label, place the inline images *rb\_full.png*, *rb\_left.png*, *rb\_right.png*, and *rb\_none.png* with the alternate text **full**, **left**, **right**, and **none** used to graphically indicate where the meat ingredients should be placed on the pizza (on the full pie, the left side, the right side, or nowhere).
2. A div element containing the label **Pepperoni** and followed by four option buttons belonging to the pepperoni field and with the values **full**, **left**, **right**, and **none**. Make **none** checked by default.
3. Repeat *Step 2* above to insert div elements with the values used in *Step b* but associated with the **ham**, **pork**, **sausage**, and **chicken** fields.

**Vegetable Toppings**

Using *Figure 7–60* as your guide, repeat the *Meat Topping Options* step above to create a field set with the legend **Vegetable Toppings**, followed by div elements with the values used in *Meat Topping Options* but associated with the **mushrooms**, **green peppers**, **onions**, **tomatoes**, and **jalapenos** fields.

At the bottom of the form, add a div element containing a submit button with the text **Add to your Order**.

Open *rb\_build.html* in your browser. Verify that the content and layout of the form resemble that shown in *Figure 7–64*. Verify that all of the form controls work as expected, that is, you can only select one location for each ingredient option at a time.

**Task 1**: When you have completed the assignment, check this task, open the task list and submit your work to your instructor.

**Task #01:** When you have completed the assignment, check this task and submit your work to your instructor.

RUN CODE

SUBMIT