CIS 3342 Project 1 – Pizza Builder Form

The goal of this assignment is to give you some experience creating a web application that includes a simple HTML page with a form that is processed by an ASPX server-side page.

Requirements:

1. Create a simple HTML page (pizza\_builder.html) and design the user interface, which will contain the HTML input form.
   1. Add simple HTML form controls (HFCs) for the user to enter their name, phone number, and address information.
   2. Add an HTML form control drop-down box (select/option list) or radio button group to allow the customer to select delivery or pickup.
   3. Add an HTML form control drop-down box (select/option list) or radio button group to allow the customer to select the size of the pizza (small, medium, large, extra-large).
   4. Add HTML form controls that allows the customer to make selections and customize their pizza.
      1. Selection for regular crust, stuffed crust, deep dish, thin crust, or cheesy garlic butter crust.
      2. Selection for sauce (tomato, white, etc)
      3. Checkboxes for toppings like pepperoni, sausage, bacon, ham, onions, peppers, etc… The checkboxes will allow the customer to choose as many as they like.
      4. Checkboxes premium additions like extra cheese, extra meat, etc… The checkboxes will allow the customer to choose as many as they like.
      5. Family Meal option that includes a choice of 2-liter soda (coke, pepsi, Dr. Pepper, Ginger Ale, etc…) and a choice for a side like (salad, fries, wings, etc…). This meal option increases the cost of the order.
      6. Option to tip the staff. The tip will get added to the total cost of the order during server-side processing. Use a textbox to allow the customer to enter a numerical value for the tip. A blank textbox would signify the customer is not including a tip.
      7. Provide at least 5 choices for each of the above categories.
      8. You must use a mixture of drop-down boxes, radio button groups, and checkboxes to allow the user to build their customized pizza order. This means you cannot use a drop-down box for everything.
2. Server-side processing
   1. The ASPX page only needs to contain a simple ASPX label or labels (WFC) to display the necessary output to the customer.
   2. Retrieve the values for all form controls using the Request object.
   3. Display the user’s order with all their selections.
   4. Calculate and display the cost for the order including tax and tip.
      1. You should assign prices for each option.
      2. Display the breakdown for the order (cost of each item / choice, tax amount, tip amount, and grand total.)

1. Good Design:
   1. The pages of your web application should be attractive, professional looking and presentable. This means they should make use of images, colors, and proper alignments to present the content. You’re welcome to use tools like Bootstrap to enhance the presentation of your pages or write the CSS yourself.
   2. Make your presentation clear to the user, providing on-screen instructions wherever needed both for data entry and output.
   3. You need to use a proper naming convention for all controls and in your code. I expect you to properly name your classes, variables, methods, etc…
   4. **You must use component-based software design. This means writing as much code in classes and functions of classes instead of in the GUI.**
2. Hint: you will need the HTML Form tag to be setup properly for this to work. The method and action attributes are the most important. The **method attribute** indicates the type of HTTP Request and how the form’s data will be sent. The POST method sends the form’s data in the body of the HTTP Request and the GET method sends the form’s data in the URL of the HTTP Request. The **action attribute** indicates the page that will process the form.  
     
   <FORM id=”frmPizzaBuilder” method=”post” action=”SomePage.aspx” >

**Submission:**~~You need to publish your web application to the Project1 folder on our cis-iis2 web server, upload your code to Canvas by creating a zip file of your entire Visual Studio Solution folder, update your Home Page to include a link to the HTML page that contains the sandwich building form and publish the Home Page to your website’s root directory, and provide the URL to your web application’s start page (Home Page containing links to your projects). Make sure a current version of your solution is located in your G:\cis3342 folder. Projects that are not submitted properly will not be graded.   
  
The grade for the required elements is based on the published version of your web application. This means the elements must work from the published web application, not just the solution submitted through Canvas. It’s important to make sure your published web application is the current version and that everything works. You shouldn’t publish to the Project folder after the project has been submitted. Otherwise, it will be considered late or may not be accepted if it’s after the deadline. If you make changes to your project, you will need to republish and resubmit the project. Generally, you shouldn’t publish or make changes until after you receive a grade.~~  
You need to zip the root folder for your solution into a single zip file and submit the assignment in Canvas. To submit the assignment, you need to click the Assignment’s Title “**Project 1**” to view the submission form and upload the file.

**Make sure you properly submit your assignment and that it works. Programs that don’t run or don’t contain all the necessary files will not be graded and marked late.**

Please be sure to save your work periodically as you proceed and also back it up. You may want to store it on your flash drive. If you are going to zip an application in order to store it, BE SURE TO FIRST CLOSE Visual Studio. If you do store information on your flash drive be sure to copy it to a hard drive on your computer before working with the project.