

### Instructions

1. You do not have to include header or body comments – but if you are not sure how to do something, or have an incomplete / non-working solution, you may want to comment your ideas about how the solution may work and what you intended to do ... it might be worth part marks.
2. You will be marked on your ability to demonstrate:
  - a. How a multi-page web application works
  - b. State management
  - c. Input validation
  - d. Server-side programming
  - e. Styling within an application
3. You may choose your own application filenames. Please ensure that your *starting page* is obvious ...
4. You must choose one server technology (either ASP, PHP or ASP.NET) to demonstrate the server-side programming.
5. Please read the entire set of requirements before you begin to code.
6. When complete, please ZIP up all source code (web pages, style sheets, *cleaned* VS projects, etc.) and submit to the **Final Exam - Submission** drop-box in eConestoga.
  - a. Please give your ZIP submission the filename *lastName-firstInitial.zip* (e.g. if you are Sally Jones – then your ZIP should be named jones-s.zip)
  - b. If you are working with a partner then include both your names in the ZIP filename (e.g. if Sally Jones is working with John Smith – then your ZIP should be named jones-s-smith-j.zip)
  - c. Only one (1) partner needs to submit to the drop-box in this case
  - d. **You may partner with anyone in the course if you wish** – just remember to follow the ZIP naming scheme above in order to convey both partner's names
7. As always please ensure that your solution runs in Microsoft Edge as well as Chrome.

**Your solution is due by 11:00pm on Saturday December 9, 2023. This 11:00pm deadline is firm and there is no room for any late submissions. The drop-box will disappear at 11:01pm and no late submissions will be accepted. So make sure to give yourself time to develop and test the solution and submit before 11:00pm!**

# WDD – Final Exam – Practical Component

## Requirements

PROG 2001

The program you are to create is a simple “**SET Pizza Shop**” online ordering application. It consists of the following:

1. There must be four pages to this application. For the purposes of this description, they are called *Page1*, *Page2*, *Page3* and *Page4* – you may call them whatever you like. Please ensure that the starting page is appropriately named
  - a. NOTE: Depending on the server-side technology you choose to use you may want or have a fifth page in order to service the AJAX calls
2. *Page1* asks for the user’s name only (first and last names). This prompt will gather this information all at once.
  - a. This means that there only needs to be a single prompt and text-box on this page and it will allow the user to enter both their first and last name into a single text-box (e.g. the customer could enter “Joe Smith”)
  - b. For simplicity sake – you can assume that there are no spaces found in either the person’s first name or the person’s last name
  - c. The only validation required is to make sure the entry is not blank and is comprised of alphabetic letters only
3. As long as the name is valid, *Page2* is presented with
  - a. A greeting saying “**Ciao *name***” where *name* is the first name entered on *Page1*
  - b. Inform the user that at the SET Pizza Shop, they can only order one (1) large pizza. And that pizza comes automatically with sauce and cheese
  - c. Show the list of all of the items that they can order on the pizza. There are five allowable items that can be added to the pizza – pepperoni, mushrooms, green olives, green peppers and double cheese
4. The user can choose as many (or few) of the toppings as they wish.
  - a. As the toppings are selected, a total price is updated dynamically on the page
  - b. This is to be done through an AJAX or jQuery/AJAX interaction with the server
  - c. The server performs the price calculation as described in the next requirement.
  - d. Please use check-boxes as the means to allow the user to select (or unselect) the various toppings
  - e. A submit button is to exist on *Page2* in order to complete the order – the button is to be labeled

### **Make It!**

5. The information selected in *Page2* is processed by the server (in the AJAX call) as follows:
  - a. The base price for the pizza (no toppings) is \$10.00 (as mentioned it includes cheese and sauce)
  - b. The mushrooms, green olives and green peppers each add an additional \$1.00
  - c. The pepperoni adds an additional \$1.50
  - d. The double cheese adds an additional \$2.25
  - e. NOTE: It is not possible to allow “double toppings” – meaning that the user cannot ask for or have multiple of the same kind of topping
  - f. NOTE: Prices include taxes – no need for adding H.S.T, sub-totaling, etc.
6. Once *Page2* is submitted a summary of the order is presented in *Page3* (a confirmation page)
  - a. *Page3* has the same greeting as *Page2*. It also has buttons to either **CONFIRM** or **CANCEL** the order
  - b. Make sure that this page displays the order that the user has placed (i.e. the pizza toppings chosen and the total price)
  - c. Ask the user to review their and then confirm or cancel it
7. *Page4* thanks the user (by their full name (first and last name)) and mentions their choice of cancellation or confirmation.

8. The “look and feel” (i.e. styling) of each of the four pages in this solution must be the same:
  - a. Each page must have the same centered heading with the text “**SET Pizza Shop**” (ensure that this text is displayed in bold white font on an orange background)
  - b. The styling on the rest of the page is up to you ... make sure that each of the 4 pages looks the same
    - Each page must have the same background colour (NOTE: The background colour cannot be white or orange)
    - Feel free to use images on your pages – just remember that each page needs to look the same
9. One thing to keep in mind while designing and developing this “SET Pizza Shop” website is - think like the user – streamline their purpose for visiting your website to order pizza (e.g. make sure to always tell the user what you are expecting from them and have good error messages that inform the user what is wrong and what you need from them, etc.)
10. Your application will also be marked based on the style, neatness and professionalism
  - a. You must use (and include) an **external** CSS style sheet
  - b. Use this style sheet to define the appearance of the heading and the background for the entire site. Perhaps even the font used ...