Project 2 - Fence Pricing Calculator

Goals:

1. Familiarity with web page development using HTML and JavaScript
2. Organizing ideas and presenting ideas in a logical fashion
3. Create a calculator-style GUI

For this project you will be creating a GUI (graphical user interface) for the fence-pricing program you developed in Lab5. If you were not able to complete Lab5, ask the instructor for a working copy of the program to use in your project. It is highly recommended that you review chapter 18 and the bean counter example discussed in class.

You are welcome to talk to other students about your project, however, ALL of the coding must be done by yourself. Your TA and your instructor will be able to help you if you have questions on HTML, JavaScript, or the code that you have written. It is highly recommended that you start the project early. Your TA and Instructor may not be available the evening that the project is due, plan ahead so you can ask questions if you get stuck.

The examples of the GUI presented in this document are meant to be used as a guideline, you are welcome to change the GUI as you see fit. However, all the basic functionality must be the same. Meaning, feel free to change anything about the design or add additional options, but in the end, any user should be able to calculate the price of a fence using all of the options discussed in this document and Lab5.

To complete this project you will need to turn in a link to your web page on Canvas. In your public_html folder save the web page as fence_pricing.html, the link to your page should be

sw.cs.wwu.edu/~yourusername/fence_pricing.html

Guideline:

1. Create a styled HTML page to hold your calculator
   - Download the template blank_project2.html
   - Add any HTML or CSS styling
2. **Add a table of buttons**

   - The table should have 4 columns and 4 rows
   - Each cell should have a button tag exactly like:

     ```html
     <td><button form = "fen" onclick=' '> b </button></td>
     ```

   - Add any HTML or CSS styling

3. **Remove any unnecessary buttons and add textboxes**

   - Delete any unnecessary buttons and leave a blank cell `<td></td>`
   - Add the price box (this will hold the final cost of the fencing):

     ```html
     <td><form name="fen">
            <input type="text" id="disp" value="0.00" size="5" onchange=' '/>
        </form></td>
     ```

   Note the form is called `fen` which we have referenced on each of the button values. Thus, whenever a button is pressed, its value will be stored in the form called `fen`. Also, notice that the id is called `disp`. We will use `disp.value` to change the value displayed in this box after the fencing price is calculated.

   - Add the length box (this will read the user's input and store it for use in the program):

     ```html
     <td><form name="fen2">
            <input type="text" size="5" name="len">
        </form></td>
     ```

   Note that the name of the input is `len`. We will use this name to reference the amount the user has entered into the length box for the program.
• Add the width box similar to the length box but using the form name fen3 and input name wid

4. Add button names and headings for the columns

• Remember &nbsp will add white space if necessary.

5. Now we're ready to add the program

• First change the onclick=' ' on each of the buttons to overwrite a variable value. IE, if the variable that is storing the number of gates is named gates, you'll want the 3 button’s onclick=' ' to change to:

```javascript
onclick='gates = 3'
```

Remember, all of the buttons holding some sort of user choice need to be changed to store the appropriate choice, whether that is number of gates, or the fencing type.
- After all of the buttons’ `onclick` values have been changed, initialize variables at the beginning of the body. Any variable used in the program should be initialized here. After the top body tag, use a script tag:

```html
<body>
  <script>
    var some_variable = "initial string information";
    var some_other_variable = 56;
  </script>
</body>
```

The names I’ve used are just examples you should replace them with names to match the variables in the program you created in Lab5, and the appropriate initial values.

- Update the CLEAR button to re-initialize any variables that may have been selected by the user. Do this by changing the button’s `onclick = ''` to be something like:

```javascript
onclick= 'some_variable = "initial string information";
    some_other_variable = 56; disp.value="0.00"
'
```

where each of the variables initialized in the step above are set back to their original values.

Note, this will not clear the text boxes for length and width, which is fine for our purposes.

- Update the TOTAL button to run the program you created in Lab5. Do this by changing the button’s `onclick = ''` to hold the function you wrote in Lab5 inside the quotes. As long as you’ve used the same variable names as your program in Lab5, you should be able to copy and paste the function from Lab5 into your `<script>` tag, after you initialize the variables.

Notice we’ve stored the user’s length and width entries in a different form than the rest of the input values. To use those values, we’ll need to store them in their appropriate variables. To do this add these lines to the function:

```javascript
length = document.fen2.len.value;
width = document.fen3.wid.value;
```

Now the variables `length` and `width` will hold the values for the length and width of the yard that the user entered and you can use them in the rest of the program.

The other difference is how we display the price to the user. We want it to be displayed in the price box, so you’ll need to add the line:

```javascript
disp.value = price;
```

where `price` is the variable which stores the price value. Remember this line should be at the end of the function after the price amount has been fully calculated.

If you want to only display two decimal places use the line:
disp.value=(Math.round(price*100)/100).toFixed(2);

Your final product should look something like this:

![Fence Price Calculator](image)

Selections made were CHAINLINK and 2 to get the displayed result.

As mentioned earlier, feel free to add any additional features or styling. Your grade will be determined by whether or not your calculator functions properly and the overall design. If you have any questions about how something will be graded, please contact the instructor.