**Assignment 2-1 Instructions: Improve the validation of the Countdown application**

In this assignment, you’ll improve the validation for the date entered by the user in the Countdown application. When you’re done, this application should display specific error messages for dates with invalid months or days.



1. Open the starter application(zipped) attached to this Assignment folder:
   1. Evaluations\Assignments\Assignment 2\Part1-count\_down.zip
2. Start the application, enter an invalid date like the one shown above, and note that the error message doesn’t accurately describe the error.
3. In the JavaScript file, find the if statement that checks whether the date string has a four-digit year. Following this code, add code that gets the month and day from the dateParts array that was created earlier in the code.
4. Add code that checks that a valid day was entered depending on the month that was entered. For example, April can have the days 1 through 30, and May can have the days 1 through 31. Be sure to account for leap years, which are years that are evenly divisible by 4. If the days are invalid, display an appropriate error message and return.
5. Add code that displays an error message and returns if a month other than 1 through 12 are entered.
6. In index.html, change the text in the <small> tag to your name and student number.

**Assignment 2-2 Instructions: Add dates to the Invoice application**

In this assignment, you’ll modify an Invoice application so it gets the invoice date for each invoice entered by the user and calculates the due date.

Table

Description automatically generated with medium confidence

1. Open the starter application(zipped) attached to this Assignment folder:
   1. Evaluations\Assignments\Assignment 2\Part2-invoice.zip
2. Start the application and click the Calculate button without entering a subtotal or invoice date. An error message will be displayed indicating that the subtotal must be a number greater than zero.
3. Enter a valid subtotal and click the Calculate button again. This time, the discount and invoice total will be calculated, but no invoice date or due date will be displayed.
4. Code a function that formats the Date object that’s passed to it in MM/DD/YYYY format and then returns the date string.
5. Add code to the click() event handler for the Calculate button that gets the invoice date and creates a Date object from it.
6. Add code that checks whether the invoice date is not equal to an empty string and whether the Date object is not a valid date. If so, display an error message, clear the controls, move the focus to the Invoice Date text box, and return.
7. Add an if statement that checks whether the invoice date is equal to an empty string. If so, use the current date as the default date. To do that, you’ll need to get the current date and format it using the function you coded in step 4.
8. Add code that calculates the due date as 30 days after the invoice date. Then, format that date.
9. Add code that sets the values of the Invoice Date and Due Date fields.
10. In index.html, change the text in the <small> tag to your name and student number.