CST8209 – Mini Project for Practical final

Objective

Using JavaScript you will be:

- Create and validate form with miscellaneous input types, submit and reset.
- Manipulate DOM based on user interaction
- Manifest programming skills using private/public functions, loops, classes and proper naming and outlining convention using JavaScript.

Instructions

- You must use at least one class, think what you can model as class.
- You must use the module pattern, or some equivalent that allows private/public functions.
- You must choose which functions should be public or private correctly
- All JavaScript and CSS must be in external files.
- Events cannot be added in the HTML, they must be unobtrusive using best practice.
- You must submit a ZIP file in the following format: CST8209_firstName_lastName_studentNumber.zip
- You must not directly copy any code or algorithms from other sources (get hints, use your own code)

Folder Structure

1. CST8209_firstName_lastName_studentNumber
    a. scripts
        i. [whatever script files you have]
    b. css
        i. [whatever css files you have]
    c. images
        i. where you have your item image files (jpg, png)
    d. Index.html
    e. data.json

Requirements

Create a webpage of your choice that delivers at least the followings:

1) Page Title and header, followed by a paragraph stating what the page is about
2) Form with different types of inputs and validation rule
   a) Text (min and max length)
   b) Number (min and max of your choice)
   c) Date (verify input to be after specific date
   d) A List that allows editing input and single selection (Required)
   e) Checkbox
   f) Radio(at least 3 options) (Required)
   g) Submit and reset buttons(Required)
   h) A read only input to show the total(please see #4)
3) Form validation:
   a) Proper styles and warning to user to indicate validation rule
   b) Required input to be indicated ahead
   c) Don't used tables to structure your webform, let's use fieldset and legend

   https://developer.mozilla.org/en-US/docs/Learn/Forms/How_to_structure_a_web_form
4) Shopping items :

   Create a presentation of at least 4 columns (see example below) - please try not to use a <table> like the example below.

   a) An item has a name, image and price (you can add more if you wish, like: category, quantity, etc.)
   b) Your shopping list will load from a **data.json** file, make sure the image file are already stored in **images** folder of your website folder
   c) When you click on the picture you update the amount and subtotal (some will use +/-buttons)
   d) The total should be reflected in the form (input 2.h up)
   e) Display for each item: image and name, you can keep your price hidden if you wish or as an data-attribute in the image element.

5) Summary:

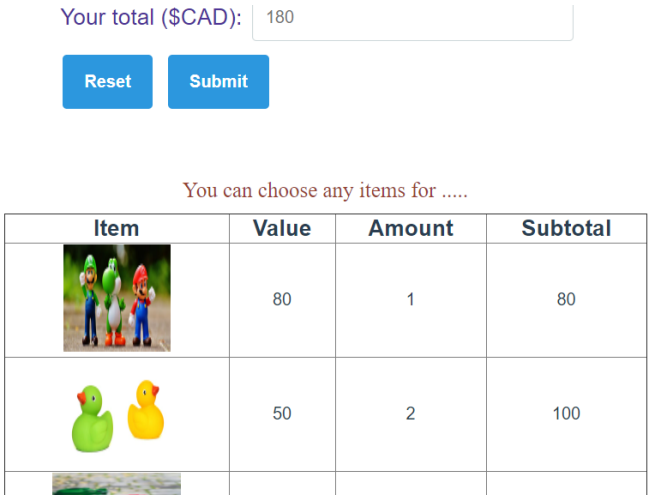
Below the item table dynamically display 1 bullet list (ul) and 1 ordered lit (ol)

User info:

- Name
- Address
- …

You picked the following items

1) 2 Duck
2) 3 Car
3) …



Marking Scheme Rubric

	Marks
Code is organized and properly formatted, indentation, JavaScript and CSS must be in external files, …	10
Comments (useful, help understand the program)	5

Instructions and conventions followed: • Names of var, functions, … are meaningful and following conventions • Class used and properly defined • Private/public functions • Events properly defined	15
DOM displayed with proper conventions	10
Form validation Form submission/reset works correctly, no alerts	15
Front-end HTML and CSS look good and makes sense	10
Json data hamdling	5
Total	70