Project: Online Store - v1.4

Course	INFO 1232 – Javascript I
Professor	Daniel Malnati
Submission	Project Evaluation Folder on FOL
Due Date	April 24th, 2020 @ 11:59pm

You will be creating a web page *simulating* an Online Store. The theme and look of the web page are up to you. (Flower store, Shoe Store, Book Store, etc)
The Store will feature the following functionality:

- Display Items available by category
- Display individual Item Details
- Add/Remove items from cart
- Calculate Order totals (subtotal, tax, shipping, total)
- Support multiple currencies
- Review system for individual items
- Display current Datetime info

***Note**: You do not have to use provided starting code, you are encouraged to structure and style the web page however you like. However, the web page must meet the same functionality detailed in this outline.*

Breakdown:

1. Create some Objects:

Store Item	
Id	String – the id for the product
Name	String – the name of the product
Price (Canadian)	Float – the price of the item
Qty on Hand	Int – the amount of the item available
Max per customer	Int – the max amount that can be added to cart
Category	String – the category of the Item
Cost of Shipping	Float – the cost of shipping
Reviews	Array – containing some reviews (strings)
Description	String – the description of the product
Image	String – link to the image file

Cart Item	
Id	String – the id for the product
Price (Canadian)	Float – the price of the item
Qty	Int – the amount of this product in the cart
Shipping	Float – cost of shipping for this item

2. Create some global variables:
    - An empty array for the store items
    - An empty array for the cart items

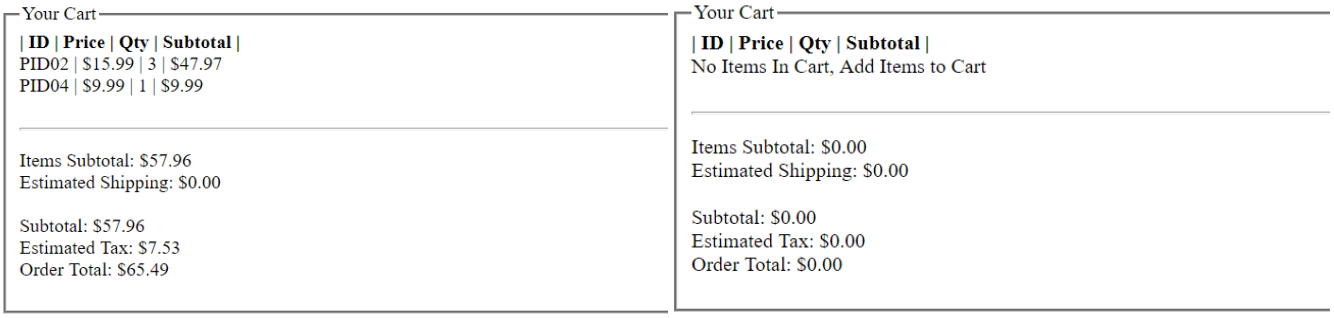
3. Create an initialize function that will be called by the **onload** event on the <body>. In this function:
    - Set the current Day/Time
    - Populate the store items array with at least **15** store item objects, with varying data
    - Call your function that will display the store items array (step 4)
    - Call your function that will display the cart items array (step 5)

4. **Display Store Items:**
   Create a function that will display the store items array in a Dynamically generated table. This function will be called when the page loads and when the user changes the item category.
    - Output for each item: id, name, price, qty on hand, max per customer category and image

   These details should be displayed using the converted currency. *(Step 7)*

5. **Display Cart Items:**
   Create a function that will display the cart items array in a Dynamically generated table. This function will be called when the user adds/removes items from the cart.
    - If there is nothing in the cart, output "No Items in Cart"
    - Output for each item: id, price, qty, (price * qty)

   These details should be displayed using the converted currency. *(Step 7)*



6. **<u>Calculate Cart Totals:</u>**
   Create a function that will calculate and display the totals for the order in the cart.
   - Output the Subtotal of the cart
   - Output the estimated shipping
   - Output the Subtotal (item subtotal + shipping)
   - Output the Tax
   - Output the Total (Subtotal + Tax)

   These details should be displayed using the converted currency. *(Step 7)*

Note: *You can use the Canadian tax (13%) even if the selected currency is not Canadian. Shipping cost can just be the cost of each item in the cart added together.*

7. **<u>Currency Selection:</u>**
   - There should be an element on the page for the user to use to select their preferred currency.
   - The selected currency should be displayed/used throughout the web page. *(Wherever a price is displayed, it is converted to the currency selected before it is displayed)*
   - The currencies are: Canadian (default) and an additional currency of your choice

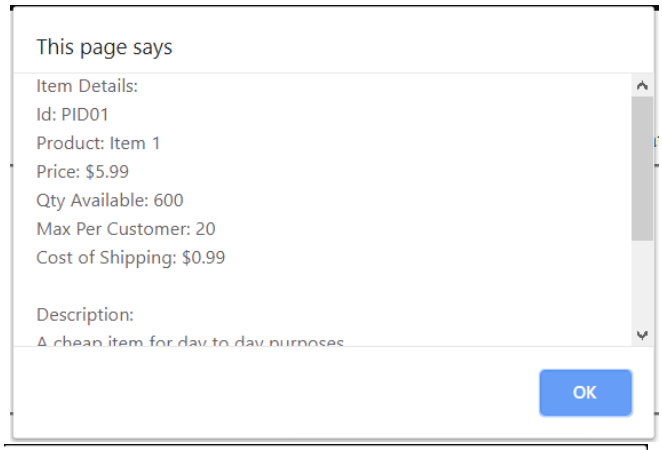
8. **<u>Add to Cart:</u>**
   - This can be done many ways:
     - An add button next to each item in the table that adds this item to the cart
     - A section of the page where the user enters the item id and qty they want to add to the cart
     - A dropdown for the user to choose the qty
     - Or any other way you wish

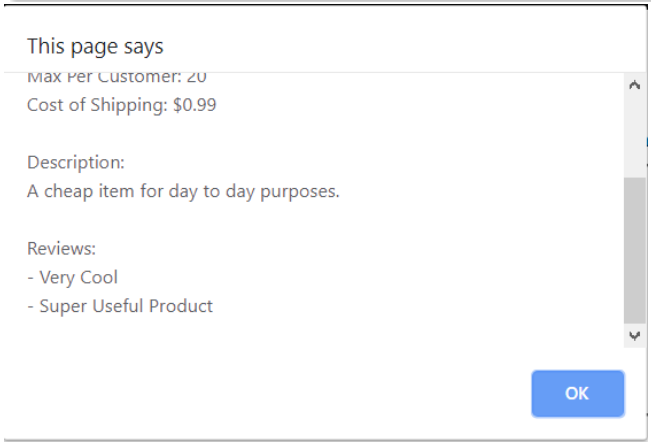
- When an item is being added to the cart, create a cart item object and populate it with the necessary details from the store item object. Then add this cart object to the cart object array. (This should then trigger your earlier function to display cart details.
    - If the Item (with the same ID) is already in the cart, instead of creating a new Cart Item Object, simply modify the qty of the existing cart item.
        - If the new qty exceeds the max per customer for that item, set the qty to the max amount per customer for that item. (If qty is 2 and you're adding 2 to the qty, but the max per customer is 3, set qty to 3, not 4)
    - The Qty on hand in the store item object should change when the item is added to the cart. (If I add 2 to the cart, I remove 2 from the qtyOnHand for the store item
- Call your function for displaying cart items
- Call your function for calculating cart pricing

9. **Remove from Cart:**
    - Similarly, to the Add step (above), There should be a way for the user to remove items from the cart
    - Remove the qty of that item with the corresponding ID from the cart array.
        - If the qty is below zero, remove the item entirely from the array
    - Call your function for displaying cart items
    - Call your function for calculating cart pricing

10. **Validation:**
    - When the user is entering any information (adding/removing/reviewing items) there should be some validation going on.
        - Item Id: *(If the user is entering an item ID on the webpage, the following should be validated)*
            - Validate that a value is entered
            - Validate there are no spaces in the value
            - Validate that the Item Id exists in the inventory
        - Item Qtys: *(adding/removing)*
            - Validate the number is not less than 1
            - Validate that there is enough quantityOnHand for an item when adding items to the cart.
            - Validate that the qty being added to the cart does not exceed the max number of items per customer value for each item.

11. **Display Item Details:**

- There should be a way for the user to view more item details for an item.
- In an alert, (or an alternate section/way of displaying)
    - Display all the Store Item properties (description, reviews etc) (with the exception of the images. Although you can if you would like)
    - 

    - 

12. **Review Item:**

- There should be a way for the user to review a specific item.
- Add the review to the reviews array for the store item object specified by the user
- The added review should be displayed when the user views item details.

This Project is worth 20% of your final grade

Marks Awarded For	Marks Available
Coding Style • **Header With Description** • **Function Comments** • **Internal Comments** • **Proper Indentation and use of whitespace** • **Naming conventions followed for variables, functions, objects, arrays. All representative of what they do**	**3**
Functional Code • **Global arrays and variables** • **Object Constuctors** • **initialize()** • **displayStoreItems()** • **displayCartItems()** • **validate()** • **addItemToCart()** • **removeItemFromCart()** • **itemDetails()** • **calculateCart()** • **review()** • **currencyChange()**	**1** **2** **6** **4** **4** **4** **5** **4** **4** **2** **2** **2**
User Experience • **Provide a controlled user experience by clearing text fields, changing colours of text fields, setting focus to text fields as required, and outputting input to the page as entered or selected, resetting any select boxes to their default after selection, etc.**	**5**
Overall appearance of the web page • **Arrangement of page elements, colour themes, font style and size, suitable images** • **Output is properly formatted with correct use of white space, without spelling and grammatical errors, and a currency format is used for money amounts**	**2**
TOTAL:	**50**

Program Submission:

Zip your web page file and submit the zipped file to the Project Submissions folder. Be sure to test your own submission to ensure that it can be unzipped and run.

Submit your project on time!

Project submissions must be made on time! Late projects will be subject to divisional policy on missed test and late projects. In accordance with this policy, no late projects will be accepted without prior notification being received by the instructor from the student.

Submit your own work!

It is considered cheating to submit work done by another student or from another source. Helping another student cheat by sharing your work with them is also not tolerated. Students are encouraged to share ideas and to work together on practice exercises, but any code or documentation prepared for a project must be done by the individual student. Penalties for cheating or helping another student cheat may include being assigned zero on the project with even more severe penalties if you are caught cheating more than once. Just submit your own work and benefit from having made the effort on your own.

Note: if you have taken this course before you must submit a new project

Sample Images: