**CSC 5750 – Principles of Web Technology**

**Homework 4 – [your name]**

**25 points – Due November 7, 6pm**

**Late deadline is November 9, 11:59pm, but 20% off**

**a)** Save this document with your name and the homework number somewhere in the file name.

**b)** Type/paste your answers into the document.

**c)** Gather the following files into a ZIP file:

● This document

● Cashier page (.HTML)

● Any other optional files specific to your implementation

**d)** Submit the ZIP file to the Blackboard item where you downloaded this document.

You've been hired again by the local restaurant that you created a web site for in Homeworks 1 and 3. Enhance that web site per the following:

**Cashier page**

● Add a two-dimensional array called **sales** that stores information about each restaurant sale including:

✓ Sale ID – a unique integer assigned to each sale.

✓ Entrée selected index – the index of the selected combo box value.

✓ Drink selected index – the index of the selected combo box value.

✓ Tip percentage – the integer tip percentage.

Therefore, an Nx4 array is used to hold sale information where N is the number of sales. The array should be declared script-level, outside any function, and initialized to *zero rows*. Three other script-level variables are needed:

✓ saleID – this provides the next sale ID as sales are added; initialize to 1000.

✓ saleCount – this tracks the number of sales in the array.

✓ salePtr – this points to the current sale in the array as the user iterates through them.

● Add function **loadData** that runs on Cashier page load and retrieves the state of restaurant sales. It reads the following four items from local storage:

✓ saleID

✓ saleCount

✓ salePtr

✓ sales – this item needs to be parsed into the array on retrieval.

● Add function **saveData** that runs on Cashier page unload and stores the state of restaurant sales. It stores the following four items in local storage:

✓ saleID

✓ saleCount

✓ salePtr

✓ sales – this item needs to be stringified from the array on storage.

● Add the following buttons:

✓ A **New** button that adds a sale with a unique ID to the array. First, expand the array by one row to make room for the new sale. The ID is placed in the new row. The first ID is 1001.

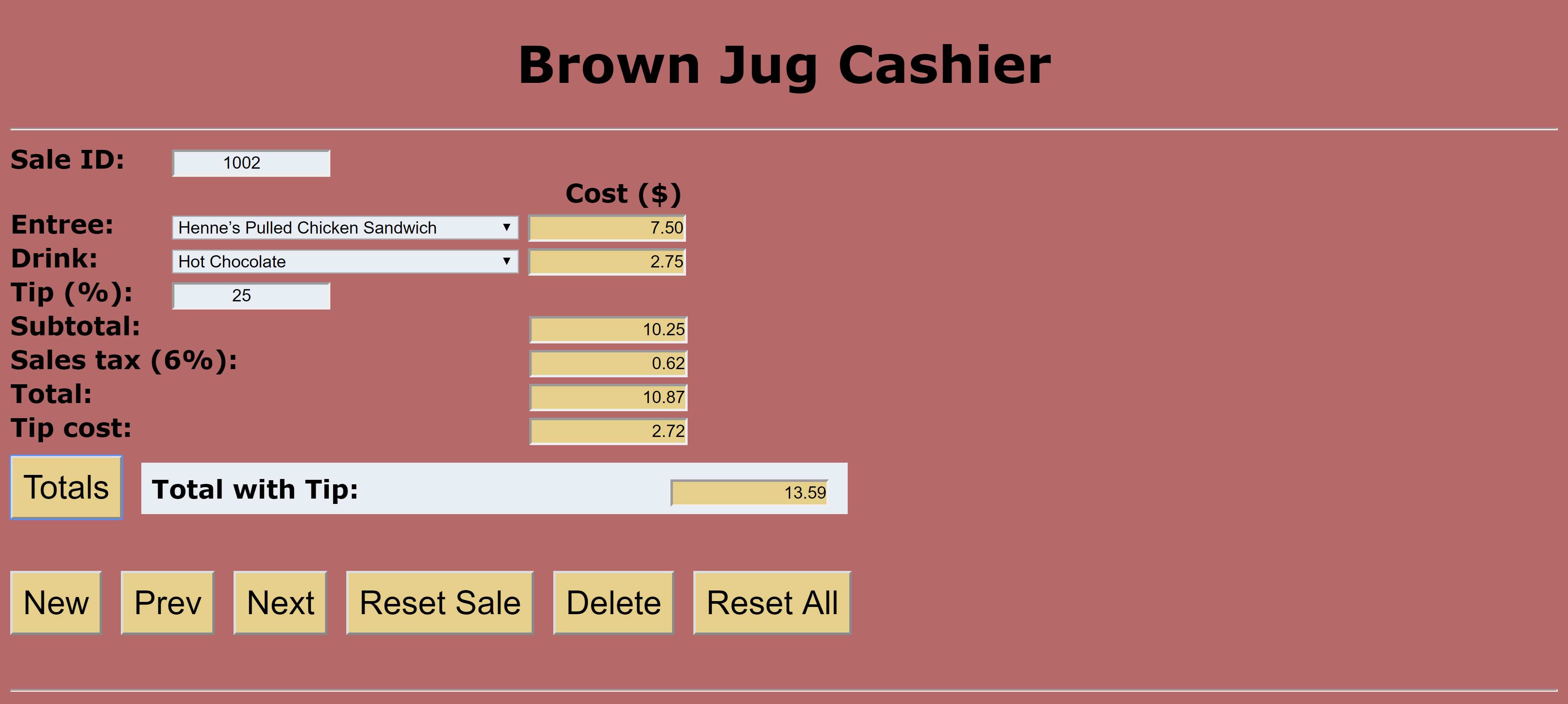
● A **Prev** button that saves the current sale data in the array and shows the previous sale. If clicked from the oldest sale, show the newest sale (wrap around). Use script-level variables **salePtr** and **saleCount** to accomplish this.

● A **Next** button that saves the current sale data in the array and shows the next sale. If clicked from the newest sale, show the oldest sale (wrap around) . Use script-level variables **salePtr** and **saleCount** to accomplish this.

● A **Reset sale** button that clears all information about the current sale from the screen, except the ID.

● A **Delete** button that removes the current sale from the array, and clears all information about the current sale from the screen. It then shows the next sale in the sales array. Contract the array by one row to adjust for the deleted sale.

● A **Reset all** button to clear all information from local storage, and all information about the current sale from the screen.



**Useful functions**

The following functions may be useful:

● resetGlobals – resets the four script-level variables.

● resetTotals – clears totals information about the current sale from the screen.

● saveSale – saves the current sale in the array before navigating away from it.

● showSale – retrieves the current sale from the array and shows it on the screen.

**Screenshots**

Include the following screenshots in this document:

● Cashier page – any sale with previous and next sales.

● Cashier page – the sale before the above sale.

● Cashier page – the sale after the above sale.

**Cashier page**

*[your Cashier page HTML code here]\**

**If possible, format your code like this:**

**Font “Courier New”**

**Font size “9”**

**Bold**

*[your Cashier page screenshots here]\*\**

\* **Copying-and-pasting web page code to a Word document**

1) From the HTML editor window, press **CTRL-A** and press **CTRL-C**.

2) From within the Word document, press **CTRL-V**.

\*\* **Copying-and-pasting application output to a Word document**

1) From the web page, maximize the browser window.

2) From the browser window, press **ALT-PrintScreen**.

3) From within the Word document, press **CTRL-V**.