SY306 Lab Seven

Cookies: JavaScript and Python

Introduction

This week we learned about cookies in JavaScript and Python. In this lab you will practice those skills by improving your shopping cart implementation so a user can browse the website and keep the shopping cart state. We will also add the "buy" functionality.

A. SETUP - DO THIS FIRST!

You must create a folder in your `public_html` called "Lab07" (without the quotes) and store your work in that directory. Copy your files from Lab06 (or Lab04) to Lab07.

VERY IMPORTANT: To allow the webserver to later create files when running a Python script, the webserver user needs to have extra permissions on your Lab07 directory. To enable this:

Type the following in the terminal window:

```
cd public_html

setfacl -R -m u:www-data:rwx Lab07
setfacl -R -dm u:www-data:rwx Lab07
setfacl -R -m u:m21xxxx:rwx Lab07
setfacl -R -dm u:m21xxxx:rwx Lab07
```

You will have to repeat this step every time you create a new directory where you want to allow the webserver to create files!

You might still need to do `chmod a+rx` for the webserver to read and execute your files.

IMPORTANT: Read the entire lab so you understand the requirements, know what to expect, and can properly plan your work.

B. Shopping Cart [40 points]

In lab 4 you added a rudimentary shopping cart to your website. Now you will improve on that.

1. You should have the file products.html and any JavaScript code for the shopping cart from your lab 04 (or later), copied into Lab07. The shopping cart functionality you already implemented should allow you to add items to cart and display the cart content.
   Your shoppingCart should be an array (unless you already implemented cookies for your shopping cart). If the basic shopping cart functionality from Lab 4 is not working, please contact your instructor.

2. Create a shoppingCart.js file that will contain useful functions for your shopping cart. Copy-paste the cookie-related functions from the cookie_js.html code to your shoppingCart.js file. Since this is an external JavaScript file, no HTML tags, including the <script> tag should be included in the .js file.

You will now write two helper functions that will help you implement the shopping cart functionality using cookies, so the users can browse your website and maintain a shopping cart.

3. In your shoppingCart.js file, write a function **saveCart()** that stores the shopping cart array you have from the previous labs into a cookie with identifier "cart". You can have an input parameter array for the function to be more generic, but it is not required.

Hint: Since cookie values are strings, you will need to convert the shopping cart array to a string, with the elements of the array separated by some special character. JavaScript Array objects have a `toString()` method that converts an array to a comma-separated string. For example, if you have `myArray = [1,2,3], myArray.toString()` will return the string "1,2,3". You can read more about this on w3schools. You can use JavaScript's default way of converting the arrays to strings, or come up with your own way. In any case, it is important that you know how your array is translated to a string, in particular what delimiter you are using, because you will have to do the reverse transformation next (from string to array).

4. In your shoppingCart.js file, write a function **restoreCart()** that reads the cookie with identifier "cart" and initializes the shopping cart array with the elements found in the cookie "cart". If the "cart" cookie does not exist or is empty (null), your function should initialize the shopping cart array to the empty array (or whatever your code uses when there are no items in the shopping cart). The function can return the shopping cart array rather than just initialize your shopping cart array, but that is not required.

Hint: String has a useful split() method that you can use. Just make sure you do not try to split() the empty string, as the result is not what you expect. You can read more about split on w3schools.

5. Modify your products.html file which contains the shopping cart functionality to use the external script shoppingCart.js, so the functions you created are available in products.html.

Hint: You can link to an external JavaScript file like this:

```
<script type = "text/javascript" src = "shoppingCart.js"> </script>
```

6. **Add to cart:** Move the **addToCart()** function from products.html to the shoppingCart.js file, so you have all the functions there. Modify your addToCart() function so the products added by a user to the shopping cart are saved in the "cart" cookie. The easiest way to achieve this is to add a call to restoreCart() at the start of your function to populate your shopping cart, work with the shopping cart array as before, and add a call to **saveCart()** at the end of your function.

7. **View cart:** Move the **viewCart()** function from products.html to the shoppingCart.js file. Modify your viewCart() function to display the products from your "cart" cookie. Again the easiest way to do this is to call the **restoreCart()** function and then work with the shopping cart array as before (lab 4).

Remember: Be sure to check that addToCart() and viewCart() still work properly in products.html!

C. Checkout [45 points]

1. In your products.html, add a "checkout" link which directs to the file shoppingCart.html that you will modify next.

2. Modify shoppingCart.html to display the current contents of the shopping cart, just as you did in Lab04 with products.html - use the external script shoppingCart.js and the viewCart() function that you already created. Copy the HTML element you used in products.html to display the cart in shoppingCart.html. To ensure the viewCart() function will still work, make sure you place the call to viewCart() after the HTML element you used to display the cart.

3. Add HTML code to shoppingCart.html to display a form (you should also leave the list of products in the cart there) with two input fields: one for delivery address, one for credit card number, and three radio buttons for the user to select the specific credit card type [Visa, MasterCard, Amex]. Add a "Confirm Purchase" submit button to the form. The action should be "shoppingCart.py"

4. Create the shoppingCart.py script that will process the purchase by recording the sale information in SALES.txt. The script will need to display back to the user a confirmation message. See details below on what the script should do:

5. Instruct the browser to delete the "cart" cookie, so there are no products in the cart anymore.

6. Save the credit card type & number, delivery address, and items purchased as a new line in the SALES.txt file. You should use \t as separator between the different fields in the file, and between the different purchased items. **Note:** the credit card information and delivery address come from the form, while the items purchased come from the "cart" cookie (automatically sent by the browser).

7. Display a confirmation message to the user. The confirmation message should contain the delivery address, the list of products purchased, and a "shop more" link which directs the user back to products.html

8. **Security:** Escape the customer delivery address to HTML safe character sequences before displaying them back in the above confirmation message.

D. Final Steps [15 points]

1. **Documentation**: ensure you have appropriate comments in your JavaScript code and Python script.
2. Create the following four links in your top-level default.html page under the heading "Lab07"
    - a. Under the name "Products", make a link to your Lab07/products.html page
    - b. Under the name "shoppingCart.html" make a link to your Lab07/shoppingCart.html JavaScript file
    - c. Under the name "shoppingCart.js" make a link to your Lab07/shoppingCart.js JavaScript file
    - d. Under the name "SALES.txt" make a link to your SALES.txt file

E. Extra Credit

For a nominal amount of extra credit do the following: (NOTE: saving a backup copy of your working lab is recommended before starting on this).

1.

2. **(5 points)** Implement **B.3** and **B.4** using JSON.

F. Deliverables

1. A new shoppingCart.js JavaScript script
2. A modified products.html that now uses cookies for the shopping cart
3. A new/modified shoppingCart.html page used for checkout
4. A new Python script shoppingCart.py to allow the user to buy products
5. You should have all the pieces working as described in sections B, C, & D above.
6. All of your files should be in a folder called "Lab07" (without the quotes) in your `public_html`. **Your instructor will assume that your web pages are viewable at http://midn.cyber.usna.edu/~m21xxxx/Lab07/products.html** . You may want to check that this URL is viewable and that everything works correctly *from a computer where somebody else is logged in*.
7. **Turn in:**

i. **Paper submission:** turn in the following hardcopy at the beginning of Lab08, stapled together in the following order (coversheet on top):

1. A completed assignment coversheet. Your comments will help us improve the course.
2. A printout of the source of your shoppingCart.py file.
3. A screen-shot of your browser window showing the confirmation message you displayed after a user bought some products

ii. **Electronics submission:** Submit all Lab07 files via the online system: `submit.cs.usna.edu` by the following deadlines:

- Section **4341**: 23:59, Wednesday, February 27th
- Sections **1151, 3351, 5551**: 23:59, Thursday, February 28th

8. **NOTE:** Lab07 Solution will be revealed via the course calendar at 00:00 on March 6, 2019. Per course policy, *"..expect **minimal** credit for late assignments that are received after answers are revealed."*