DOUGLAS COLLEGE

COMMERCE AND BUSINESS ADMINISTRATION
COURSE OUTLINE AND SCHEDULE

CSIS 3380-001: ADVANCED WEB PROGRAMMING WITH JAVASCRIPT & AJAX

Assignment 3: Storing and Retrieving State Information Using Cookies

Instructions

Please begin by renaming assignment 3 folder to include your name as follows: *YourNameCSIS3380-AS3 (e.g., PAdamsCSIS3380-AS3)*. After you complete the assignment, zip/compress the assignment 3 folder and upload it to the blackboard.

Note: The assignment is to be completed individually. Any form of cheating or sharing of work may have serious consequences.

Due Date: Friday, November 30, at 11:59 PM

Background

HTTP was originally designed to be stateless, which means that web browsers stored no persistent data about a visit to a website. Today, there are many reasons for maintaining state information, which among other things, allows a server to:

- Customize individual web pages based on user preferences
- Temporarily store information as a user navigates within multi-page forms
- Provide shopping carts that store items selected until checkout

In this assignment, you are going to use cookies to store and retrieve state data (i.e., user inputs/selections) as a user navigates within multi-page forms.

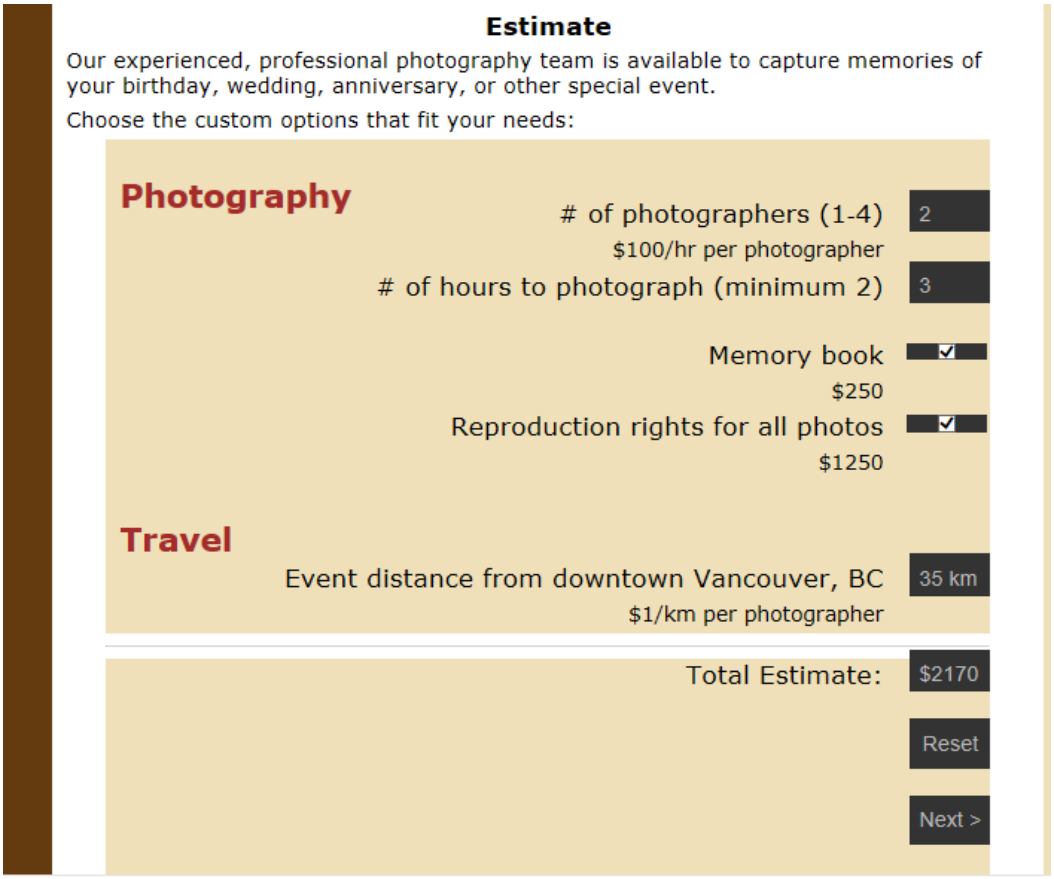
Details

In this assignment, you will capture customer order/booking for Fan Trick Fine Art Photography located in Vancouver, BC. Fan Trick Fine Art Photography sells photographic prints and offers special event photography services. The owners want to expand their website to offer information about digital photography, and to provide a rate estimator for their services for prospective customers.

You are provided three HTML document templates named "*orderEstimate.html*", "*customerInfo.html*", and "*orderSummary.html*". Prospective customers place their order/booking on the *orderEstimate.html* page, enter their details on the *customerInfo.html*, and then order summary is displayed on the *orderSummary.html* page. **Note**: the *orderEstimate.html* page should have its ***form action*** attribute value set to *customerInfo.html* and **method** attribute value set to *POST*. Likewise, the customerInfo.html page should have its ***form action*** attribute value set to *orderSummary.html* and **method** attribute value set to *POST*. The three pages are linked as follows:

What you should do

1. Please use external JS scripts for all your work.
2. Create a script called "*estimateOrder.js*" that will be linked to the *orderEstimate.html* page. The *estimateOrder.js* script should have a function called *getOrderEstimate* that will compute total cost estimate based on user selections on the *orderEstimate.html* page. Note that if a user decides to change selections/entries, the cost estimate should be adjusted automatically (for example, checking Memory book option adds $250 to total estimate, whereas unchecking that options decreases total estimate by $250).
3. In the estimateOrders.js script, define a function called createEventListeners that will process *onchange* events as users make selections on the *orderEstimate.html* page. Note that this function calls the *getOrderEstimate* function for each user selection/ input on the *orderEstimate.html* page. The screenshot on the next page shows the selections I have made and the order estimate thereof.
4. Create a script called "*orderEstimateCookies.js*" that will be used to create cookies to store user selections/inputs on the *orderEstimate.html* page.
5. In the "*orderEstimateCookies.js*" script, define a function called *setCookie* that has name and value as parameters (remember form inputs are encoded in "name=value" pairs before being submitted to the server), which should be used to set a cookie for each *name=value* pair on the *orderEstimate.html* page. Set the expiry date of the cookie to 30 days and path to the path of the *orderEstimate.html* page.
6. In the "*orderEstimateCookies.js*" script, define a function called *storeValues* that will call the *setCookie* function to store each *name=value* pair on the *orderEstimate.html* page.
7. To ensure that cookies are written over everytime you load/refresh the *orderEstimate.html* page, call the *storeValues* function using form "onsubmit" event.



8. Create a script called "*customerInfoCookies.js*" that will be linked to the "*customerInfo.html*" page. The "*customerInfoCookies.js*" script will be used to create cookies to store customer info on the *customerInfo.html* page.
9. In the "*customerInfoCookies.js*" script, define a function called *setCookie* that has name and value as parameters (remember form inputs are encoded in "name=value" pairs before being submitted to the server), which should be used to set a cookie for each *name=value* pair on *customerInfo.html* page. Set the expiry date of the cookie to 30 days and path to the path of the *customerInfo.html* page.
10. In the "*customerInfoCookies.js*" script, define a function called *storeValues* that will call the *setCookie* function to store each *name=value* pair on the *customerInfo.html* page.
11. To ensure that cookies are written over everytime you load/refresh the *customerInfo.html* page, call the *storeValues* function using form "onsubmit" event.
12. Create a script called "*orderSummary.js*" that will be linked to the "*orderSummary.html*" page.
13. Inside the "*orderSummary.js*" script, define a function called *displayOrderSummary* that will retrieve cookie data, parse it, and display it. To display the data in an appropriate format, You will need to use *decodeURIComponent()* function to remove any URI encoding. You will also need to use several methods of the String class

(including substring(), split(), replace(), and indexOf()) as well as array manipulation. Note that in order use split() and replace() methods, you will need to define Regular Expressions. For privacy reasons, do not display customer's credit number. Instead, credit card digits should be replaced with "*", except the last four digits (e.g., 4321567812349876 should be displayed as **********9876). For the purpose of the assignment, assume that Visa and MasterCard cards consist of 16 digits, while American Express and Discover cards consist of 15 digits.

The order summary output should be displayed as shown in the following demonstration (note: page header omitted below – doesn't mean that you should delete it):

- Number of photographers: 2
- Photography hours: 2
- Memory book: on
- Reproduction rights: on
- Distance: 35 km
- Total estimate: $2170
- Event type: birthday
- Event date: 11/30/2018
- Event venue: Vancouver, BC
- Customer name: Bill Adams
- Customer email: badams@yahoo.com
- Customer phone: 604-123-7654
- Card type: Visa
- Card number: **********9876
- Card expiry date: 01/2019

Note: Payment for your booking with be processed at the function

Assignment Submission

Zip your assignment 3 folder and upload it to the Blackboard.