CS290\_Hamilton\_Proj6\_Reflection

Bennett Hamilton

5/15/23

Project 6: Reflect, Observe & Analyze

1. **In */manufacturers/index.php*, an error can occur if you try accessing a $\_COOKIE key for the favorite manufacturer when it does not exist. In what situation might there not be a favorite manufacturer? What do you use (in code) to manage the case where there is no favorite manufacturer in the cookie?** There might not be a favorite manufacturer if a user is accessing the site for the first time. You can handle it with an if statement using a isset() condition check if the cookie is set within the super global: $\_COOKIE.
2. **There are no star image files in the codebase, so how is it that visual blank and colored stars render in the web page? Explain how the browser accomplishes this, given the HTML it parses.** It appears that some Unicode is being read which outputs as specified stars. “&#11088;” renders a gold star and “&#9734;” renders a blank star. The php code uses a function called favoriteButton($cat\_id, $fav\_cat\_id) which decides which star to render. When favoriting something it links to the favor.php page which sets the favorite category id and returns back to the previous page with the updated variable. When the ids match a gold star is displayed.
3. **What would happen if the code in */categories/favor.php* and */manufacturers/favor.php* used the same cookie name? Explain why this is "good" or "bad."** This would be bad because then whenever you would favorite a category it would overwrite the favorite manufacturer id. It’s possible that they match and nothing changes but it is more likely that they would match and it would cause a change in both favorite ids. That would be a bug that would need fixing.
4. **Imagine you are viewing the manufacturers list page (*/manufacturers/index.php*). When you click on a blank star to "make it your favorite," think about what happens.**
   1. **How many requests and responses occur between the moment you click the blank star to the moment you see it colored in?** It would send a request to favor.php, which would then respond with the php code, within that code it requests the $id value from the query string and it responds, then it would request to set the cookie "favoriteCategoryId" value to $id, and then it requests a header location change and responds by going back to previous page, but with updated cookie this time.
   2. **What important data is sent in the query strings, request header, and response header?** The ids of the categories and the current favorite category ids.
5. Imagine you are viewing a motorcycle page (*/motorcycles/show.php*). When you click on the *Add to Cart* link, think about what happens.
   1. How many requests and responses occur between the moment you click the *Add to Cart* link and the moment you see the number of cart items change in the navbar's *Cart (N)* link?
   2. What important data is sent in the query strings, request header, and response header?
6. Imagine both Janet and Jonah are using your Motorcycle Mania site. Janet selects *Honda* as the favorite manufacturer, and then a minute later, Jonah selects *BMW* as the favorite manufacturer. Explain why *both* Janet and Jonah will or will not see a colored star next to *BMW*.
7. Sessions are an abstraction over cookies. Your current implementation of Motorcycle Mania uses plain cookies for the favorites but uses sessions (which are cookie-powered behind the scenes) for the wish list and cart. Use your implementation to select a favorite category and manufacturer and add a couple motorcycles to your wish list and shopping cart. Now, let's assess your understanding of the differences between plain cookies and sessions.
   1. **Open the Chrome Developer Tools *Application* tab and select *localhost* under *Cookies* in the sidebar.**
   2. **Click on your *Categories* or *Manufacturers* links in the navbar of your site. What is stored in the cookie? Name them. (You should see more than one thing.)** The favoriteCategoryId, favoriteManufacturerId, and the PHPSESSID
   3. **Click on the *Wish List* or *Cart* links in the navbar of your site. What is stored in the cookie?** Just the PHPSESSID
   4. **Why do you not see the motorcycle ids for the wish list or cart in the cookie?** Because they are in the session id, so the data is not stored in the cookies but rather the session id (which is in the cookie).
   5. **How is it possible for the server to know the motorcycle ids for the wish list and shopping cart, when those motorcycle ids are not present in the cookie?** They are pulled out of the session id and stored to a variable using $\_SESSION[“wishlist”] (along with the php implode(strarr) function)
   6. **Explain how PHPSESSID is used when your server-side code accesses $\_SESSION.** The session id is initialized when first running the page, this allows the server to identify which session data to retrieve for the particular user, when data is requested from the server the PHPSESSIONID is used to locate the file/database/etc to retrieve data from, the super global $\_SESSION php’s way of telling the server to retrieve data with a specific string lable (ex. “wishList”).
8. **Imagine that Jenna has visited your Motorcycle Mania site and added five motorcycles to her cart. "Dreamy!" she says, and then she goes about her life for a few days. If the server is configured such that sessions "expire" after 24 hours, what will Jenna notice when she visits Motorcycle Mania again after five days?** She will notice that her wishlist will be empty as the cookie expired after the first day and was no longer keeping track of the wishlist ids.