Zachary Hess

CS2830 Final Project

Professor Wergeles

Link to Page: http://ec2-34-203-230-101.compute-1.amazonaws.com/

\*The numbers are the requirements stated on the Final Project Criteria Sheet

1. 5 Required Tags and HTML5 - <!DOCTYPE html> is used to ensure html 5 and the other required tags are present on all pages.
2. Consistent Design/ Interface - The design of all of my pages is very similar using the same color scheme as can be seen in the .css files. The selection bar that I have also put on the top of each page is the same so there is a lot of similarity
3. Implementation – Each page of mine has as much overlying code as possible as seen with the selection bar I created. I do not have to edit an element multiple places in my code.
4. Login/ Logout – My home page, parts page, and feedback page are all unprotected content. By clicking on the builds page you are sent to a login page. Using the correct username and password you will get to the builds page where it says you are logged in and has a logout button that upon click sends you back to the home page. After you log out if you click on the builds page again you will be sent back to the login. This code can be found in login.html, connect.php, and builds.html
5. ID/ Password – The ID of “test” and Password of “pass” were set in connect.php and will work to give you access to the builds page using the login system.
6. PHP – PHP can be found used heavily in my login system (connect.php). It also can be found connected to my feedback page (history.html) using the GET method where the feedback provided by the user is sent to comments.php.
7. GET/ POST – GET is used on the feedback page (history.html line 49) to send the comments made by a user to the php page (comments.php). GET was used on the feedback page because that does not need to be protected content. POST is used on line 17 on login.html to send the login information to (connect.php) to check if the username and password entered are correct.
8. Form Elements – I used another form element besides what is on the login page on the feedback page (history.html line 49). Here the user enters their comments and feedback about the site into the form and they are sent to comments.php.
9. Input Response - The user provides input on my site within login.html where they enter their login information and also on the feedback page (history.html). If the user were to leave a field blank on the login page (login.html or error.html if they have already got their information wrong) they would be presented with a small message saying the field must be completed. For the login page if the user enters their username and password wrong they are given a window alert saying that the information entered is incorrect and after acknowledging the alert will be presented with an empty form.
10. Multiple Photos – There are multiple photos on my webpage on index.html and on parts.html. Within index.html (Home Page) all of the photos are present immediately, within parts.html (Parts Page) after clicking on a part you are presented with a short description and two photos of the part. This holds for each part listed so there are at least 18 photos on the parts page including the two in the header.
11. Embedded Video – After logging in through the login page (login.html) you are directed to the Builds page (builds.html) where you will find an embedded Youtube Video present in the middle of the page. The code for this can be found on line 54 of builds.html.
12. JavaScript – Each page utilizes JavaScript as each are linked to theWorks.js. Within theWorks.js you will find a javascript function to load new pages on line 1 and another JavaScript function for an AJAX request on line 283.
13. jQuery – jQuery is utilized within theWorks.js and executed within parts.html to show and hide the photos and descriptions of each part as you click on a new one. You can find the jQuery code from lines 31 to 279 in theWorks.js. Each page on my site also has jQuery linked to it right before the end of the body tag.
14. jQueryUI – I utilize two jQuery UI functions, .show() and .hide(). These can be seen used from lines 31 to 279 in theWorks.js. With these functions I was able to show and hide the information on the parts page (parts.html) each time you clicked on a new part to learn about i.
15. AJAX – I used AJAX within the builds page (builds.html) to reveal the name of a website where a user could get help building their own computer. The code is on line 61 in builds.html and the function is it calling is a JavaScript function utilizing AJAX on line 283 of theWorks.js.