Alex Ketavongsa - ISTE-240 Project 2 Grading Rubric

Validation

All pages validate:

I ran my pages through w3.org validation and it all passes validation.

All pages display properly in all browsers:

I tested my pages on Chrome, Safari and Firefox. The page is optimized to work correctly and media queries work as expected.

Design

Design principles (CRAP, color, layout etc) – Good Design COUNTS!

I kept the same exact design principles as I had on the midterm project. I think that my webpage meets CRAP because the alignment of pictures and text took some time, they are placed in logical order and have good amount of spacing. The contrast of the pages are good, it is easy on the eye and the headings stands out. There is repetition on each page, following the same layout, font and color. The alignment of the pages makes the text easy to read. I have a content box in the center of each page so there is margin space between the body text. Proximity is displayed by the captions below each of the images giving a brief description for that image.

Organization/Navigation – mobile friendly:

I used float on almost all of the images. I wanted to make the pages "flow" so I alternated the float between left and right. That makes the pages more interesting by not having the images all on one side. The organization of my content makes sense. The pages only discuss what is relevant to that topic. My navigation bar is listed in logical order, starting from the history page and ending with the feedback page. The media queries hide the navigation bar on smaller screens to a “hamburger-style” to save some space for text.

Corrections from the midterm project:

I lost points for having an incorrect process.php landing page from my survey page. Since then, I corrected my process.php page from the first midterm project to have the same layout and design as the rest of my website.

Must Be Mobile Friendly – media queries – Responsive design:

I have media queries that will collapse the navigation bar to the “hamburger” style when the screen size is between 374px to 768px.

Technology

Original JavaScript:

I used a photo slideshow as my original JavaScript. I added the slideshow in a separate div to “contain” that image so that regardless of the photo size it does not move any body text.

DHTML component:

I added a rotation animation effect on my site’s logo. When you hover over the logo in the navigation bar, it will spin the image 360 degrees. When you hover out, the image will rotate back to the original position.

Contact form; Comments saved to and pulled from database (insert and select):

I created a separate comments.php page for the comments section. I added the link to the navigation bar as well. The JavaScript validates the name field to make sure it isn’t empty, then adds the comment number, name, actual comment, and date into a database. Afterwards, the database record is retrieved and appended to a comments table.

Form validation with JavaScript:

I used JavaScript to validate my forms and kept most of the same code from the midterm project. I added some new functions, for example, to validate the name field for the comment box. Another cool thing I did was add a “focus” function on the survey page. If the user leaves a required field empty, the page will jump to the top and highlight the required fields and display an error message. I think this was a good add-on because after adding more checkboxes it requires you to scroll to the bottom to submit the form. It can be hard to notice the error message unless the user scrolled back up the page. The “focus” function makes it easier for the user to see the error and correct it.

Form data validation and sanitization with PHP:

In my process3.php file, I performed data validation and sanitization for the values retrieved from the survey form. I checked each $_POST variable with the test_input() function from the survey form to see if it was not empty and store the value to a variable. If it was empty, I just echoed an error message to the email/database. Also, I sanitized the PHP to strip any harmful characters that someone might use to inject into my form. I used the trim, stripslashes, htmlspecialchars functions and regular expressions to filter out the harmful characters.

PHP includes on every page (header and footer):

Every PHP page that includes a header.php and footer.php file. All it does is call that file which contains my header and footer code. It makes the code cleaner and easier to read. I had to use some PHP to modify the navigation bar to get the current page that it’s on. This was used to get the “active” page to display the nav button of the current page in green.