

John Odejar (jao57)
CS 2300
Project 2
Rationale

For the inputs for adding an entry to the data file, I only checked for three things. One of them was that none of the inputs were empty when submitted. All 5 inputs had to be filled out for the entry to be submitted and written into the file. The second thing I checked for was that the input for the Voice Actor category had less than 50 characters in it and contained a string consisting only of letters, numbers, hyphens, underscores, and periods. The third thing I checked for when validating the new entry input was that the input was not already in there. I had to check every category for each character to see if all 5 categories of the input matched with all 5 categories of an entry in the file. If there was a match, the input would not write a duplicate entry into the file. Also, when comparing the data in the file and the input for duplicate entries, I ignored whether a character was uppercase or lowercase when comparing the two strings. For the search inputs, I checked that the two fields for the first filter out the five filters I had on my site were not empty. The other four filters were optional and did not need to be filled out for submission. I also ignored the case of a character when comparing data on the file and the search input.

I limited the Voice Actor category to that size and those characters since most voice actors go by their normal names which are just letters while others have stage names with the extra characters I added. The Hair Color and Gender categories did not need to be validated since I gave a set list of the possible inputs that could be put in for the category. I decided to use the options list since there were very few values that could be inputted for the categories. There are only so many hair colors and genders out there so I thought using the list was a better way to do this when compared to using regexes and if-else statements to limit and validate user input. I also did not put any limitations on the fields for the anime title and character name since there are many strange titles out there that use a wide variety of characters and are unusually long. With characters that are aliens and computer programs or have weird nicknames, I didn't think it would be a good idea to limit the values that could be inputted for these fields. I also prevented duplicate entries since I only needed one version of an entry in my catalog. For the search, I only had 5 possible categories to filter by so I put them all into an options list. I also didn't put any limitations on the search input because there are a variety of values the user could input depending on the category he chooses to filter by.

Examples of search over multiple categories:

1. Anime Title → Fairy Tail and Gender → Female and Hair Color → Blue
 - a. Show = Wendy Marvell
 - b. Not Show = Natsu Dragneel and Erza Scarlet
2. Anime Title → Fairy Tail and Gender → Female
 - a. Show = Wendy Marvell and Erza Scarlet
 - b. Not Show = Natsu Dragneel
3. Character Name → Son Goku and Gender → Male
 - a. Show = Son Goku from Dragon Ball and Son Goku from Naruto
 - b. Not Show = Saitama and Accelerator

When doing a search, I simply made it an “and” search because I wanted each filter to narrow the entries. By adding more filters, the search would narrow down the number of entries that would show up in the search results.

Other things that I implemented when creating this web page were to use a second page to display the search results. This made it easier for the website’s visitor to know that their search worked. With the search results, I put a message saying how many entries matched the search. I also put links both on top of the page and bottom of the page that linked back to the homepage so that if there is a long results list, the user wouldn’t have to scroll all the way to the top or bottom of the page to see the link to get to the homepage. Users cannot reach the results page through a link and can only go to the page after submitting a search. Additionally, I kept my search results and the full catalog organized by having the entries be listed in alphabetical order by character name. I also organized the catalog and search results so that case is ignored when sorting the data.

Furthermore, I implemented an accordion for the 5 filter searches. The accordion allows the user to click on the filter header to show or hide the input fields for the searches. I used jQuery to implement this. I had created a div for each filter search and split that div into two divs – one for the content and one for the header. The jQuery would slide the content div up and down to show and hide it. The content div was told to be shown by adding a class called “active” to the header div, and that “active” class was removed when the content was hidden. I added the code to have the first filter show by default. I chose to use the accordion because I didn’t want the 5 filter search divs to take up a lot of space especially since all but the first one is optional to fill out. I used the accordion to save space and make the page look more organized. I had decided to create 5 filter divs so that the user may input a search that narrows down the characters by all the different categories.

Based off jQuery code from this site: <http://inspirationalpixels.com/tutorials/creating-an-accordion-with-html-css-jquery>