Jason Chen

Programming Usable Interfaces

Section A

9 October 2016

Assignment 3: Implementing Prototype with HTML/CSS/JS

**Considerations**

For this assignment, I chose not to use jQuery as I wanted to see what I could with solely javascript. Furthermore, I felt that jQuery was out of the scope of this assignment, as this assignment was mainly an introduction to HTML/CSS/JS, and I felt that jQuery would make it too easy to complete the assignment. I was thinking of adding in animations without using jQuery, but I felt that it would take too much time. It should also be noted that I am in Section A so I did not make this mobile responsive, but if I were to do so, I would use media queries to adjust some styles of my current application.

As mentioned before, this is a light exercise on javascript, I mainly focused on the bare functionality of the to-do app: creating tasks, deleting them, and marking them as completed. I do validate if a task is empty or not before adding it to the "list," however, the date is not verified for simplicity. Furthermore, most were left out because JQUERY was not used (which would have made it quite easy). Some basic transitions for input focus or button hovers were incorporated via css. I added basic transitions for completing, adding, and deleting tasks using a CSS transition and adjusting opacity.

I tried to go the minimalist route, so I made the list simply take a task name and a due date. When the task is complete, you click the box next to the task, and the item moves to the bottom of the list and has a strike-through over the text, marking it as complete. In addition, both complete and incomplete tasks could be deleted.

If a task was created without a due date, the date section lists it as "none." For the sake of simplicity, dates were entered in as simply text inputs, date pickers are typically done through jQuery, so that was left out (and I did not want to make my own).

**Challenges**

A challenge that I faced for this assignment was actually the implementation of the todo list. I first thought it would be a cool way to have two arrays or dictionaries: one for incomplete and the other for completed tasks. With those two arrays, I would populate the elements, but I ran into some issues, and I found that it was just easier to populate upon adding a task (mainly an oversight on my part).

Another challenge that I faced was that I had a hard time trying to figure out how to delete a certain element’s parent element, so I had to do some research and console.log a bunch of things. I also had trouble with some css selectors like the nth child or how to style placeholder text, but they were not major.

The last challenge that I faced was trying to add transitions for completing, adding, and deleting tasks without using jQuery. I mainly just changed the opacity, but I found that I had to use a setTimeout() for it to work properly.

One thing to note is that my github repository is NOT private or empty. There is no way to change the repo as private, and I pushed my code to the repository out of common engineering practice. I talked to Anna about this, and there is nothing really that I can do. If someone copies my code, you can look at my commit history to see if I plagiarized any student’s code. Sorry for the inconvenience.

**Extra Features**

* Github repo: <https://github.com/jasonc1/taskr>
  + It’s also hosted as a github page: <http://jason-chen.net/taskr/>
* Animations (add, delete and complete task)
* Assigning tasks to people (add as text input – no groups implementation)

**External** **Tools**

Having prior knowledge and experience with front-end development, I mainly used two external tools for this assignment: Skeletoncss and StackOverflow. I used Skeletoncss for a very minimal and simplistic css framework. It is one of my favorites because it offers the bare necessities for a css framework: buttons, basic forms, grid system, and a few more. In my opinion, Skeletoncss provides just enough for me to do what I need, and it provides a huge amount of freedom so I can create my own styles. The rest of this application was made from scratch – no javascript framework or jQuery.

I used StackOverflow mainly to search up javascript DOM elements like selecting the nth child of a certain element, removing a parent node of a certain child, or how to add a style by setAttrbute(). Mentioned earlier, I also searched up how to push an array and things like that (for my implementation), but I changed it afterward. I had to search up a few more advanced selectors for css like :nth-child() and how to style a placeholder for an input. Last, I had to search up how to use setTimeout() for the use of transitions. The rest of the work I did was done on my own and with my prior knowledge.