Lukas Garcia

June 17, 2017

COP 4854

**Project 2 – AngularJS Website Manager**

AngularJS, talk about a step up from JavaScript. When I first heard about using the library, for a moment, I thought that I was going to be in trouble because I barely know as much as I do about JavaScript. I know enough to get back, but when I started using Angular, it felt like the use of having to write out so much code was lifted. The relief of being able to write less to do the same amount of work, was an awesome feeling. Yet, I feel like the curve might be a just a tad bit steep to get through it all. The basics are interesting, but when I got my feet into the thicker mess of it all, I felt like having to slam my head into a wall.

When I started working on the project, the basics were common to me after watching the tutorial. The thing that got me the most was the different uses people were applying Angular and writing it out. For my project, I stuck with what I was taught, setting up everything like the tutorial on Angular’s website had shown. But, when I ran through examples, I saw many people were using the ‘$scope’ syntax to get through their work. Maybe, if I did use it, it would’ve taking me much shorter of a time trying to figure out most of my bugs. The main complaint with writing it out the way we are shown in the video library’s website is that whatever I wrote as the main scope name/variable of my function, I would have to type that out into the html file.

I went through a thicket of code just to come out on top, halfway through, I felt like going back and rewriting it all, but by the time I got there, I said “screw it,” and kept writing the code I had in the beginning. So, from here on, I just stuck with what I had and managed to go through different examples of to-do lists. I found some nice explanatory ones, and I saw some simple examples. But, for my project, because I am just learning how to play around with AngularJS, I just stuck to a simple layout.

Wanted to use something simple as well because I didn’t want to bog down the user with so many features. If the idea of the app is to do a couple of things, it should do those features at this moment in time. If the time comes, then maybe later I would implement more features. So, this time around, I stuck with what was needed for the job to get done. The application must add, edit, check, and delete an item, this being a bookmark, and it had to keep the data persistent. I ran into the issue of keeping the data persistent, and I’m still having issues with that at the current time that I write this report. But, that wasn’t the one thing that kept me up all night. Adding, checking, and deleting items I found reasonable after scouring the internet and looking for some good examples.

I could bump into several examples that seemed to run the same through websites like [www.jsfiddle.net](http://www.jsfiddle.net) and [www.codepen.io](http://www.codepen.io). I got the most help from using what others could recreate on jsfiddle. I saw one example, and I wanted to use it since it gave me everything I wanted to learn for the project at hand. I was able to recreate the main function for the bookmark items, but I was struggling to figure out some functions just because the user was using the ‘$scope’ syntax throughout his project. There were some items that looked too much, and I made sure to steer clear from them. The biggest issue that seemed to have taken the most of my time was trying to figure out how to edit an item in a list.

Editing, the one thing that should’ve been easy to do, but no, it took me roughly 6 hours to figure out what was going on. I went online and looked for an answer to my problem. I kept running into code that was beyond my outer limits, but I kept looking. At first, I thought it was something that had to do with the ‘ng-dblclick’ call function within the HTML file. I kept swapping out functions and trying different variations of editing an item. I think I went through 4 examples of unusual ways of editing, but nothing was solving my problem. At that point, I stopped and took a break.

Even after taking a break, the problem kept running up behind my mind. What was causing the issue? I went back to my computer and looked online some more. I went over and over again through the functions I had created for editing. I modified a bucket ton of code within the HTML file because I know for certain I was doing something wrong. Knowing full well that I must’ve been doing something wrong. Maybe I had place the wrong ‘ng’ syntax snippets in the wrong spots. Some code snippets were not firing within <span> or <div>, and it was driving me nuts. At one point, I was so worried that I was going to destroy my code from rewriting, that I created two separate .js and .html files. These files allowed me to dig in, but I still had an issue five hours in.

I took one last break and then got back online to see if I could find any useful information. Then, I hit the jackpot of useful information on [www.stackoverflow.com](http://www.stackoverflow.com) that I wish was planted into an Angular doc file. Someone was having the same issue, but I learned at that point that AngularJS has an issue with reading ‘text’ and ‘specific-text’. So, for example, in my **bookmarkList.bookmark**s function that I used to set up my original website links to help guide the first user, I had set up the initial URL’s using text; like so:

**bookmarkList.bookmarks = [ {text:'www.google.com', done: false}, {text:'www.humblebundle.com', done: false}];**

The main issue with this is that because the values are set to text, it’s a value that Angular cannot identify amongst the majority of the text surrounding it. I don’t know how true this is, but from reading what I read on stack overflow. I had to modify the ‘text’ fields and change it to ‘name’ fields instead. I felt like pulling my hair because that was staring in my face for six hours and didn’t even think of looking at it. After rummaging through my code and changing that function to this:

**bookmarkList.bookmarks = [ {name:'www.google.com', done: false}, {name:'www.humblebundle.com', done: false}];**

as well as modifying the HTML code, my edit functions finally started to work. I was happy to have finally gone over that hill. A good place that helped me with this too was [here](https://www.undefinednull.com/2014/08/11/a-brief-walk-through-of-the-ng-options-in-angularjs/).

The one thing that I am still having issues with is trying to get the data to stay persistent within someone’s local storage. I added the storage js file into my HTML file, and have it connected, but I cannot find a good place that teaches how to implement the functions into the project. I feel like I probably missed a sizable chunk of code because everywhere I go, the local storage files are mingled in with the rest of the code. That means I would have to go back and write in code that could lead to more bugs. The problem is that if this does lead to bugs, I wouldn’t know if the bugs would lead me to something good, or in the end completely backwards.

So, in the end, I ended up learning a good chunk about writing in proper naming conventions for the Angular. Also, regarding the color scheme for the project, I tried to cover the same color aspects that I used in project 1, the resume. I wanted to keep everything minimal and used the same color palate. I also include hover effects to both the “Add to Favorites” button and the “Delete ‘x‘ bookmarks” button. Created the hover effects so that a user will have better sight as to what they are doing and what they are about to click. I formatted the way the edit input field shows up so that it can appear below the text field that is being edited. I would’ve like to include this on the same line, but I feel like this a better explanation so it can show the user what is was they clicked on before they went into edit mode. I used a dark slate background and wrapped around both the header and the body of the application.

At the end of the day, I was satisfied with what I had made. Still, it gripes me that I couldn’t figure how to stare data locally to a person’s computer. If there is a better explanation out there, I’m bound to find it. Maybe it is something simple, but I doubt it. Either way, this turned out to be a good learning experience, but I think the Angular curve is still a little bit steep. Then again, I find this method of coding, as well as web development, much easier to understand than that of coding actual software. I can manipulate that status of an object, and modify how that reacts within an environment. If this was something like C++ or C#, it is a possibility that I would’ve had to seek extra help from outside sources. It would’ve taken me longer to write the code, and I would be in a deeper hole. But, like everything, it is a learning experience worth going after. I hope that the future is programmable as the AngularJS project.