**Project 2 Report – Jared Selling**

For this project, I created a simple web app that lets users create a list of bookmarks for their favorite websites. While conceptually straightforward, the introduction of AngularJS tremendously steepened the learning curve.

I started out by going through the recommended resources for the module, playing with TodoMVC and following along with the youtube tutorials provided. However, I quickly found some of these tutorials to be out of date, as my code didn’t work when following along. A quick check in the comments section confirmed that some of the code being demonstrated in the Todo list tutorial had been deprecated. Fortunately, I discovered a very thorough, but somewhat rough and clearly unedited tutorial series which showed how to create a todo list that saved to local storage with create, update, and delete functionality.

Rather than start from one of the prefabricated todo list apps provided in the module’s resources, I opted to follow along with this tutorial and create my own app, while simultaneously making the appropriate modifications for a bookmarking app. When I completed this, I had a very rough outline of functionality implemented, but there were a lot of things I still needed to change. For example, the tutorial didn’t include buttons, relying instead on ng-keyup directives to detect the “enter” key. I kept this functionality of course, but I felt the app would be more intuitive if I included buttons so I added those and used the ng-click directive to bind them to appropriate functions. I did find that I was unable to recycle the add and edit functionality from just pressing enter because those functions only executed if event.which == 13, which of course didn’t apply to a button click. Thus I ended up with the same functionality in different functions that executed their content based on different conditions. This bothered me a bit, and I’m positive there was a cleaner way to implement these features, but due to time constraints, I just accepted it and moved on.

My next challenge arose when creating a dropdown form to edit existing list items. I tried setting the appropriate indices of my itemList array to $scope.editURL and $scope.editDescription which were the variables I had bound to the edit form’s input boxes through ng-model. However, this failed to update my list array. After a lot of fruitless research and variations, I finally settled on a rather awkward workaround that selected a specific child node of event.target.parentNode.childNodes[i]. I was painfully aware that this was not “the angular way” as when I added elements like <br> to my edit form, I had to go back and adjust the index of the childNode I was selecting in my script.

Finally, the primary issue I ran into was that my ng-model=”editIsVisible”, which controlled the visible/hidden status of my dropdown edit form, would not update when I toggled it in angular. The Boolean value would appear to change when I logged it to the console, but it wasn’t causing the form’s visibility to toggle. Some research revealed that the digest loop was not updating, so I bound the document to a click and ran a function that used $scope.apply to force my Boolean variable editIsVisible to false. Then, when the user clicked out of the form, it would disappear as desired. I found myself using $(document).click() to simulate this click event and force the digest loop to update whenever the user submitted a form to either edit an item or add a new item. After doing this, the edit menu’s visibility would always toggle as intended. Again, there was probably a cleaner way to implement this, and I wish I had more time to try different things, but I settled on this workaround due to time constraints.

Overall, I found this to be an immensely educational project. I learned a lot about Angular, but recognize there is still much more for me to learn. I am satisfied with the functionality of the project, and mostly, but not entirely satisfied with my implementation. I will likely continue to perfect it whenever I find some spare time. This project was successfully tested in Google Chrome and Internet Explorer.