File I/O Reflection

I thought the File I/O activity was a very useful way to practice source control with group dynamics early during the Full Stack Summer Camp. It took us a couple cycles of branching off from dev and merging back into dev to become comfortable with the process; yet, we ran into less issues with source control than I foresaw. I believe our team worked well dividing up work and completing that work as well in a timely manner. This exercise was great for practicing teamwork, patience, and other soft skills required when working in big groups. Although our group created a Trello board, we did not take advantage of the tool as much as we could of. The project was small and made using the Trello board feel less useful than it would be in a larger project with more moving parts.

Unit Converter – Client-Side Reflection

The breaking up of the Client-Side and Server-Side portions of the Unit Converter project proved to be helpful when learning HTML, CSS, and especially JavaScript. The team dynamics yet again proved to be fruitful, even though we were down by one to two people each of the days. We used w3schools.com as well as moqups.com to our advantage so we could focus on making our project look good and have a great user experience. Our plan for workflow included making an index page first to use as a template for the other pages. This proved to be useful for styling all the other pages but created a bottle neck as we decided to wait to add the JavaScript until the index view’s HTML and CSS was copied over to our other blank views for individual alterations.

Unit Converter – Server-Side Reflection

I believe this portion was perhaps the largest conceptual jump. Creating only one controller for this project made it more difficult for all members of the team to learn and practice writing code for the controller. It also bottlenecks the workflow in a team since generally one to two people must work on the controller before implementing the views due to their reliance on the controller. However, splitting up the controller from the rest of the lower layers of the stack allowed for space to learn about controllers and how razor code operates in a .cshtml file. Although the best uses for razor code were highlighted, I got the impression that razor code may still be an enigma for some and could be better studied with individual activities that reinforce the learning of razor code.

Final Project Reflection

CAN DO AFTER THURSDAY

Full Stack Summer Camp Reflection

Overall this experience has been miles better than last year’s and I’ve learned so much more than I remember from the previous. The “random” grouping of teammates was very useful for developing the soft skills needed when working in a team. Taking notes this year proved to help myself learn and pay better attention to the material being presented. The increase in note taking, collaboration and opportunity to learn from those who have previous knowledge of the material allowed for much better quality in our projects and quicker learning arc over the three weeks we have had.

I think the process could be improved by using the summer camp as a “project” where our team could have a sprint every week or two. This would allow us to practice agile development and more specifically scrum when dealing with a multitude of projects such as our portfolio, final project and individual learning activities. This would also allow for teams to practice with Trello and foster more use of slack for communication in contrast to resource sharing. Another improvement could come from…