Reflective Essay

Name: Gearoid Sheehan

Student ID: R00151523

Introduction

At the beginning of this term, I was highly enthusiastic about studying the front-end frameworks module. While I knew it was going to be testing and a lot of work, I had used the Angular framework previously when on my internship last year and was looking forward to further developing skills I had learned there. Something I was particularly interested in improving was my ability to use observables in typescript in order to make my web applications reactive. I also was looking forward to furthering my knowledge of styling languages. I have always felt CSS is not properly taught to developers when in university despite having to use it multiple times across our degree.

It was clear from the project briefs that the projects allowed for a great deal of creativity as they were not tightly coupled to a certain design. However, the module was not all plain sailing and there were times throughout the term where I met challenges when attempting to complete the projects. There were moments where I was unsure whether I would get a sufficient amount of the projects done to obtain a decent result, both due to the length of the projects themselves and the ongoing workload of other modules. Due to the fact that I was using a different framework to what was being taught in the lectures, I had to take it upon myself to find decent tutorials and articles in order to get the required information to code the projects. Overall, I am happy with how the researching went. I found Angular's official documentation to be clear and concise, and I was able to solidify that information by watching YouTube channels such as 'Programming with Mosh' which gave excellent Angular tutorials. If I had one discrepancy with the decision to use Angular, it would be that there were several times I watched tutorials and used information from online which ended up being for AngularJS, causing moments of frustration and confusion.

Reflection

I first attempted the 'rock, paper, scissors' early on in the semester. I liked the way in which both of the two assignments were given to us at the very start, as this allowed me to plan ahead. When creating the new Angular application, I was unsure which kind of styling I was going to utilize. I decided to go with SCSS, because even though I used very little SCSS in the end, I didn't want a situation where I installed CSS at the beginning and had to change to SCSS midway through the project. I found this project particularly easy for the most part. The logic used for the game was simple and the architecture was very condensed, with very little data being passed between components and services etc. I spent one weekend getting the bulk of the project done, however I was confused with regards how to I would store sample data, as well as create a secure login when there was no backend involved in the project. One of the first elements I added to the project was the AlertifyJS library. A task which I completed during my internship the creation of an Angular

service with this library, so I was already very well versed in how to use it. I heavily relied on the Bootstrap framework to structure and style the page, using its grid layout with rows and columns. As I had used very little of Bootstrap before, I found structuring the layout difficult at the start. I used the container free layout. I chose this layout as there were very few elements to the websites design and there was no need for unnecessary grouping. I decided to go with a fun, interactive theme where the hands would change during the countdown like a real game of rock, paper, scissors. I felt this added to the whole 'game' aspect of the website quite well. Due to the fact I was struggling to figure out how to create the login, I was also not sure how I would incorporate security into the website. As I had the bulk of the work done, I decided to leave it for the mean time and begin work on the second project.

The second project was the CIT Smart Multifunctional Sports Arena. Immediately, I could tell this project was going to be a much bigger challenge than the previous project. To begin with, I got a notebook and sketched out all my ideas on how the website layout would work. For a while, I could not figure out how I would make full use of Angular's component based architecture, as opposed to having a separate component for each page like a traditional static website. After some time delegating and researching, I decided to go with a card based layout. This layout would display all the sports arenas clubs in a grid fashion on the home page with minimal information, and then they would expand when clicked to provide full info and booking option using a modal. This proved very effective and I was ecstatic with the end result. I also created a nav-bar for the website, but this presented issues of its own. While the nav-bar worked as intended, I was struggling to add the functionality where it would only appear when a user was logged in, and not all the time on pages such as the login. In order to solve this problem, I decided to backtrack and figure out how to add a login and register system once and for all.

After much research, I eventually decided to create dummy data in typescript arrays, with the dummy user login data being compared with the incoming login data for validation. I created a login page using template driven forms with Bootstraps error checking and the Bootstrap Jumbotron class. While this worked to a point, the issue persisted after login as when I moved to a different route the user would not remain logged in. I fixed this by adding the logged in user data to local storage when the user logged in. On each page load, the page checks the local storage if there is a user object present. This worked perfectly, and I was then able to move on to creating a route guard to stop users which were not logged in from accessing pages via entering the routes manually.

Once I had the login issue fixed, I thought I had gotten over most of the problems I was going to face completing the projects. But another issue nearly immediately began clear. I was struggling to install font awesome so that I could use their icons in the project. Interestingly, I could get it installed to the point where certain icons would work, but the majority didn't seem to want to render on my webpages. Admittedly, I eventually could not figure out what was going wrong and settled for whatever similar icons I could find which actually rendered. Due to this, you may notice some of the icons used are out of context.

Now that the core structure of the website had been created, I was able to plough through the rest of the required features. I found that much of the feature creation after this was using template driven forms for booking clubs, making appointments with trainers and editing user profiles. Due to this, I

was able to copy and paste much of the code from what I had already created for the login. I was also able to add a register and forgot password modal on the login page, which allowed for new users to create accounts and existing users to reset their password. As I used the same card based layout for the rest of the components, I was also able to copy and past much of the SCSS also. I did my best to keep a sports based theme, with the CIT colours and logo featuring throughout the project of course. Using Angular pipes, I added search bars into the home, booking and appointment components so that the user could filter through the values by entering the clubs name/trainer name into the search box.

Next was to implement the features I had been looking forward to learning about the most, which was using observables and RXJS operators to make the website reactive. Using knowledge, I gained from a YouTube tutorial, I implemented a feature where when appointments and bookings were cancelled or added, the component refreshed in real-time and either removed or added the data to the component without refreshing the whole page.

As I was coming towards the deadline fast, I decided that I had done as much as possible for the Smart Multifunctional Sports Arena project. I also knew I had to return to the Rock, Paper, Scissors project to implement the login system I had now figured out. While this was a copy and paste job again, I was under a lot of time pressure. Due to this, I regrettably had to leave out the 'Training-plan' feature in the second project. I went back to finish off the first project and then decided to submit my work.

Conclusion

Overall, this was a module which I highly enjoyed completing. I feel that I have learned a great deal about web development during its course, which is an area of computer science I have always had a big interest in. What I liked most of all about the module was the freedom of creativity. While I used Angular framework on my placement, I was always very limited in the scope of the tasks I could complete, as the designs for them were already made and I had no say in them, I just had to build them. In these projects I as left to my own free will, which I believe resulted in my knowledge being greatly increased as I was able to try out new things. If I was to complete the module again, there are a few changes I would make in my approach. The main change would be organization of my time. While I started the projects very early on in the semester, I tended to work on them infrequently in blocks of a day or two at a time. I feel that I could have saved a lot of time if I had spread out the work a bit better, as often I became consumed on a problem for hours where it might have been better to leave it alone and come back to it the next day with a fresh mind. Having said that, there were also times where I got large amounts done by focusing on just that module for the day, so maybe a bit more of a hybrid approach would have had the best outcome.

I have also concluded that the Angular framework is my favourite technology I have learned in the realm of computer science so far. I highly enjoyed using its features and was very impressed by its extensive range of libraries. Visual Studio Code, the editor I used, was also a joy. Angular and Visual Studio Code is a match made in heaven, with both complimenting each other extremely well. Features such as intellisense and the integrated source control for git (which I used for both projects) made the process of completing the projects so much easier.

As a whole, I am very pleased with how the module went and the way in which the projects turned out. While I am a bit disappointed that I did not get to finish the very last part of the Smart Multifunctional Sports Arena project, I am content with the features that I did manage to complete

for it. I will most definitely return to completing it in full when I have some more free time, which is a fair representation of how much I enjoyed coding the projects in the first place.