Over the course of the last few years I have learned quite a bit regarding computer science. Initially, my only experience was that of an avid gamer with a minor amount of other computer and coding knowledge. Now, I have gained the ability to code, work in an agile environment, and even understand more about how the internet works. When considering working in a team environment, I have learned about the agile scrum and how quick development cycles work. With this small team, each person has their own specialized abilities and can be brought together to work on certain features. The quick development cycle also keeps everyone on the team in daily communication through scrum meetings to discuss what has been done, and what is still to do. This agile learning also works in communication with stakeholders. While not at the daily meeting, the stakeholders will be brought in at the end of every cycle be that weekly, bi-weekly, or however long it is. This allows the developers to communicate what has been completed, and the stakeholders to communicate what needs to be done or changed. This communication helps provide constant feedback both ways and allows time to not be wasted on unnecessary features.

I also learned about object-oriented programming, and a multitude of data structures and algorithms. In general, data structures can be quite simple or complex depending on the needs of the programming. Stacks and queues are used to organize processes into either a last in first out, or first in first out order. Linked lists and data trees can store data linearly or in branches. All these methods can be used depending on the needs of the program. I think my greatest strength has come in software engineering and database management. A good software engineer can utilize the many available libraries to their advantage. For example, in this portfolio, there are many functionalities present in the node.js runtime that allow for many functions to be performed without having to create the specific functionality. This is shown with the crypto

dependency which was used to hash information, or the express framework which provides the backbone for using http requests. Finally, with security I have learned a few of the ways that bad actors do what they do. By knowing some of the tools that are commonly used, you can create a more secure application. Even basic security like demonstrated in the API portion of this portfolio requires a user to be logged in and of a certain role to be able to access specific areas of the program. This basic level of security helps prevent problems caused by the users accidentally making mistakes which is just one of many security levels.

The three chosen artifacts attempt to display the more technical aspects of my education. My first artifact displays the ability to use object-oriented programming to enhance the first program I ever created. This software structure allows for simplification of the program and expanded the functionality without greatly increasing the lines of code. The second artifact showcased use of algorithms such as quicksort to help speed up the searching process. This is one example of an algorithm that can be used to allow a program to access data more efficiently. The third artifact demonstrates my knowledge of database administration as well as security. Creating private methods that require a specific user role to be allowed to edit the data provides security. These methods then properly access the database to return the correct information to the user.