Self-Assessment

Working though my past course work and developing my e-portfolio for this course emphasizes my ability to grow both as a developer and a designer. SNHU has provided me with projects that simulate a real working environment. Whether it comes to working on a collaborative project using git-hub with a large team or developing a unique program based off user needs. Using an agile based approach to tackle problems and communicate issues with shareholders as well as other team members. Each class has a unique and relative approach to the computer science field in different areas. Areas of study ranged from server development and protection to visual 3d representation using code. Another benefit is the different languages that were used throughout my program. Initially the schoolwork was done in python, after a few courses it turned to java, then C++ and even C for embedded systems. This was greatly beneficial to me as a student as I have familiarity with all of the above languages and environments.

The artifacts/projects that I chose emphasize the growth I have developed throughout my schooling here at SNHU. The vector sorting program uses data from a CV file of a history of bids. By using a parser all the data is loaded into a program that can be sorted and searched through for easy access of data. The program itself essentially serves as an auction house. The quicksort logic displays bids on properties and how much time is remaining on given lots. When A bid is placed the list automatically resorts accordingly using a vector sorting method and timer.

The animal shelter database pulls data from an existing database and allows for the sorting of dog breeds based off stakeholders needs. In this case, the owners of a rescue corporation know that certain dog breeds are better at performing specific actions than others. So, they needed a way to sort through and search the database for pre-existing animals that fit their needs. This program was run in MongoDB as well as a app interface associated with it.

The last GPIO artifact was the most challenging and drew on all my coding knowledge thus far. The program is a thermostat that is run off a TI external board. By pressing buttons on the board, a user can adjust the temperature on the thermostat to either turn on cooling or heat, which ever is called for. The temperature is then displayed on the lcd screen on the board. Every cycle period the current temperature reading is sent to a remote server to record the data. This allows for the user to check the temperature remotely using an app or cloud server.