CS 499 Capstone: Professional Self-Assessment

Cody Pritchett

Southern New Hampshire University

The enhancements made within my artifact for my CS 499 Capstone project were exciting to plan and execute, learning experiences each their own, and humbling in the aspect that I have felt constant waves of triumph and inadequacy during development. Computer Science as a whole program has shown me that, while I have learned an incredible amount in a short time, I have barely a drop in the bucket of understanding and knowledge in the industry, but that is okay. The enhancements that I created for my weather station project showcases several of my strengths as a prospective developer and computer science student. I was able to take a very simple and rudimentary project, plan several iterative enhancements, and execute their implementation. With those enhancements I was able to improve the software design, implement data structures, produce databases, and present meaningful data to stake holders. With each enhancement, I was able to thoughtfully implement each requirement, abide by conventional coding standards, show clear comments, and accurately reflect on each iteration. While this project was worked on by just myself, I was able to demonstrate modern developing practices highlighting my potential as a future developer. By working on my project in sprints, I was able to simulate an agile development cycle in a single man project. Each new update is accurately documented with update notes reflecting fixes and changes, mirroring my own ticketing system. Each development action or approach I took was to represent collaboration in a team environment.

For my artifact choice, I opted for a single artifact with three enhancements. This decision was made because I felt strongly about the original project, and I felt that I could implement each of the enhancements into the single artifact. My artifact is a prototype weather station written in Python that operates on a Raspberry Pi with expansion board. This is a project selected from CS 350: Emerging Systems and Technologies course. The artifact uses environmental sensors connected to the Grovepi expansion board and, using Python, displays temperature and humidity data to the console and LCD screen for the user to observe. I took CS 350 in the summer of 2020 and found this course extremely interesting and highly beneficial to my understanding of hardware/software relationships.

***Software Design:***To expand the project past being just a script, several other Python scripts were written to support the weather station. Rather than the weather station starting in an indefinite loop, the project now initializes with a terminal menu, giving the user the option to run each module (light sensor, DHT sensor and LCD screen) through a test ensuring its full functionality. The user can run the weather station from this menu as well as exit the program. To further increase the ease of use, I created an executable script, allowing the user to simply double click a desktop icon to run the program.

***Data Structures:*** I first created two Python dictionaries, one for day and one for night values, to store temperature and humidity values. These data structures are initialized outside the While True loop in the main weather station python script. To give these values meaning, a variable for a timestamp is initialized, formatted, and dumped to json compatible formatting. This timestamp is initialized and assigned to every loop iteration, giving the temperature and humidity values a date and time of their logging. Further down in the main weather station script is an if else loop with conditions for the light sensor. Previously this would either display data to the console or display a message that it is too dark. The if else loop now will store a timestamp, temperature, and humidity data to the daylight data structure, writing it and appending it to a day light json file if light conditions are bright. If it is too dark, t the if else loop now will store a timestamp, temperature, and humidity data to the nighttime data structure, writing and appending it to the nighttime json file. This gives us two separate json files to work with and display, as well as exact times.

***Databases:*** Utilizing the JSON files generated from my last enhancements, I created a visual display of the data utilizing HTML and JavaScript. This allows the end user to view the data in a web browser/graphical display. There are two graphs, both are multi-series graphs. There is a day display with temperature and humidity values with the timestamp on the x axis, and another graph for the night data. This allows the end user to view all the data collected for day and night on separate graphs. Each data point collected can be rolled over with the cursor and the temperature, humidity and timestamp will be displayed.