Module 5 – Milestone Four

Kenneth Lockhart

Southern New Hampshire University

CS-499-13459-M01 Computer Science Capstone

Gene Bryant

April 06, 2025

I have chosen for category three, Databases, to focus on an artifact from CS 340. The artifact from CS 340 was a project focused on CRUD for a MongoDB database. This project was code with python and jupyter lab. The project focused on being able to create, read, update, and delete from a database. It then used jupyter in order to build a dash-based webpage for reading the data and displaying it to the user. Originally, this project was designed to work with an older version of jupyter and dash. This project was started at the tail end of 2024 and was submitted on 12/22/2024 as complete.

The primary reason for choosing this artifact for my ePortfolio was to convert the database from Linux to Windows. I figured it would be good practice converting software between the two as Jupyter supports both platforms and so does MongoDB. However, quickly I learned that this was far too simple, so I expanded the scope to not only converting the project to Windows but to also including the setup of MongoDB. This setup focuses on not only installing Windows but also on Linux and focus on security which was not a major focus of the original design. This allowed me to focus on skill such as Database Management, Database Design, web design, python, and jupyter notebook. These skills would be important to improve my ability to integrate such features together. In order to achieve these goals, I had to set up MongoDB from scratch, install the dependencies from scratch, and finally modify and adapt existing code to work with new environments. This enhancement gave me a better understanding of the project and helped being clarity to the process used throughout the project and felt made it feel as if it has come full circle.

The primary outcome I set out to achieve with this enhancement was to expand the functionality of this project to more diverse audiences via an expansion towards Windows from Linux. Windows is still a more dominant OS so opening up a CRUD based on Windows would

help expand the scope of the project. However, as the enhancement got underway, I was the opportunity to expand this to include a security mindset that would prevent exploits from accessing the database or design of the project. Such examples include protecting the admin account of the database, limiting permissions of exposed users accounts, and addressing reasons for these concerns. For example, while an account with access to the database would work for the project because it is exposed via the CRUD module it is important to ensure that account only does what is needed of it. Finally, looking into the remaining of the functionality of the project I found that it already met most of these outcomes aside from the one listed above.

In summary of this enhancement and modification of the artifact there are several things I had to face and learn. First, I had to learn how to setup both Jupyter Lab and MongoDB in order to test the functionality of the existing code on the Windows platform. Installing both was quite the process as I had never done such a task before. My biggest hurdle in the process was MongoDB as they have to focus a lot on cloud solutions such as Atlas which I felt was a little out of scope of this enhancement. I end up learning how to install MongoDB locally and how to upload the CSV data to it using MongoDB Compass. I had to prepare the MongoDB by creating a table called aac and a collection called animals. These where the names used in the code to access the data stored within MongoDB and to keep the modification simple, I reused them. Using MongoDB I could then import the CSV to this collection which finalized the setup of MongoDB in the data department. However, I still had to set up a user to access the data so I add an admin user via the shell so that I could access parts of the database using code. I then moved to Jupyter Lab where I modified the animal_shelter.py and notebook file to reflect the MongoDB changes. This included modifying the host, port, username, and password. With all the changes made it was time to run the project which led me to my second big hurdle dependencies. In the

interest of saving time the dependencies required are part of the new install.md in the files, however in particular it was a struggle to overcome and that is Dash. The version of Dash did not align with the one downloaded from pip. In the project an old version of Dash which had to have JupyterDash specified was not in the current version. This left me with a choice find the right version or big the bullet of updating Dash and fixing the issues. I opted for the second as most of the code was html, I figured the bug fixing would be limited, and I was correct. I modified a few specific lines to reflect the new methods for Dash and the application ran successfully. At this point I was surprised as the code just worked on Windows without issue. I was ready to pack up the project and submit it when a thought hit me. I am using an admin account that is hardcoded into the application. If anyone somehow reads the data, they will have full access to MongoDB. Red flags and alarms sounding I made a second account with read-only permission and swapped the user for that one. This did give me some worry though; the project had no way to warn the user of this potential security risk so I opted to create a new file called install.md which walked through the whole process outlined above and included statement that highly recommend to create a secondary account to prevent unauthorized access. Overall, the project was far more successful and easy then expected however it lacked some finalization steps for being a full product. I feel that I add those finalization steps and ensure that users are aware of security issues and the importance of taking secure steps to protect their database especially when they are open publicly.