Augustus Mendy

Southern New Hampshire University

CS-499-10450-M01

Professor Ramsey

October 05, 2025

5-2 Milestone Four Enhancement Three Databases

This paper is a narrative that goes along with the database artifact upgrades. It provides insight into the process of creating the chosen artifact and explains why it was included in this section of our ePortfolio. According to Southern New Hampshire University (2025), the narrative centers on the learning that took place during the fabrication of the item.

Prompt

For the databases category, the Salvare Search for Rescue Web App was chosen as the artifact. In order to locate and classify suitable canines for training for various rescues, the web application seeks to create an online interface that integrates with an existing database from animal shelters. For the computer science course CS340 Client/Server Development, the web application was planned, created, and designed. The Dash framework and the non-relational database MongoDB are used in the Python development of the application. PyMongo is a Python driver that integrates Python with MongoDB. The program can be used as a test tool in a Jupyter Notebook and as a computer terminal that has an internet browser. A CSV file of all the dogs that are currently in shelters is imported into MongoDB as part of the application's functionality. Dependencies including the Python PyMongo driver, Python libraries, and Dash framework are also loaded, and a Python source code and CRUD module are used to work with the data that has been imported into MongoDB.

This artifact was chosen because it had a multi-tier application that used an application programming interface (API) and a Model View Controller (MVC) in conjunction with a RESTful protocol design to extend the HTTP protocol. The MVC pattern's separation of concerns is its most alluring idea. The role of the model is to handle Python data structures, MongoDB, and data. The view is responsible for determining what the user sees and how on the Dash framework. The controller is in charge of pulling, modifying, and supplying the user with PyMongo driver data.

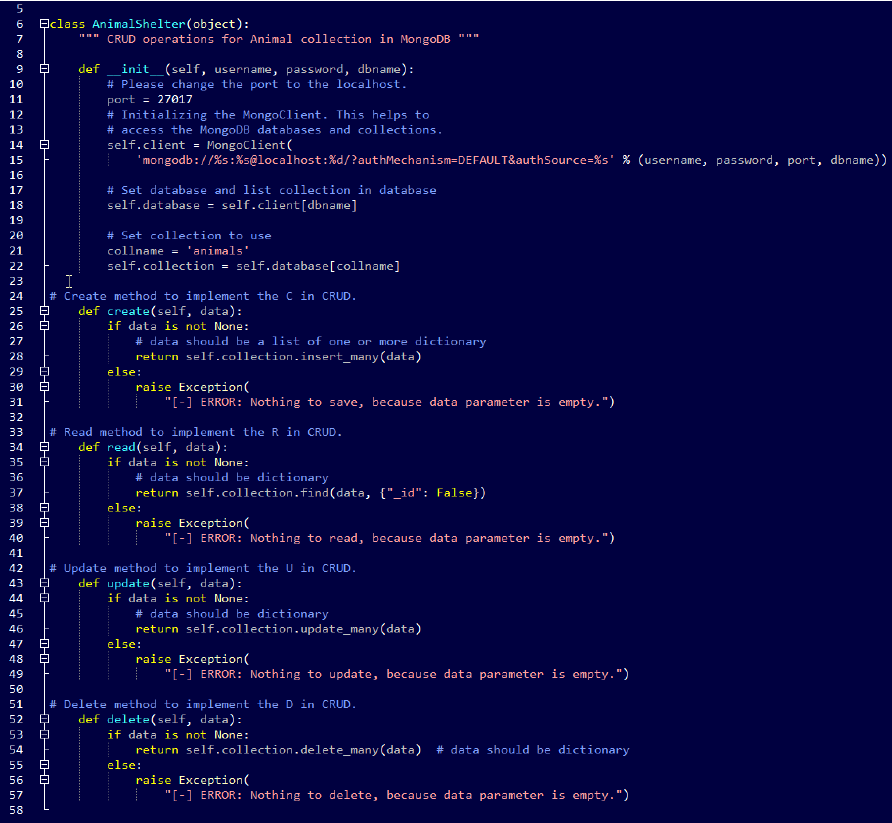
Every record in the MongoDB database is a document that is described in BSON, which is a binary version of the JSON-formatted data that the web application retrieves. When the user launches the main application file in the computer terminal, a new tab in their browser opens that points to the app address (for example, http://127.0.0.1:8050/). When the browser loads, the client-facing web application dashboard appears, as shown in the screen below:

A screenshot of a map

AI-generated content may be incorrect.

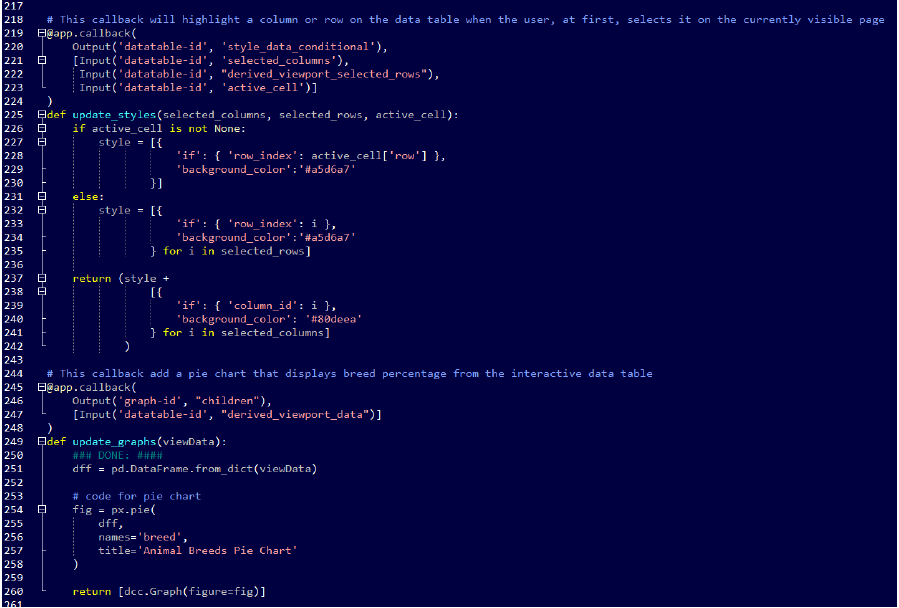
*Figure 1 Web App Client/Database Dashboard*

The artifact entails the engineering of procedures for input data validation as well as the architect and design of default denials for database record access. By anticipating adversarial exploits in software architecture and designs, this talent helps me develop a security mentality that protects privacy, mitigates design faults, and ensures improved data security and resource utilization. The source code supports clean code with descriptive function and variable names, is well-documented, and has an easy-to-maintain commenting style that complies with Python standards. The database records are handled by the CRUD module, which allows code reuse by allowing other Python programs to import it as a module.



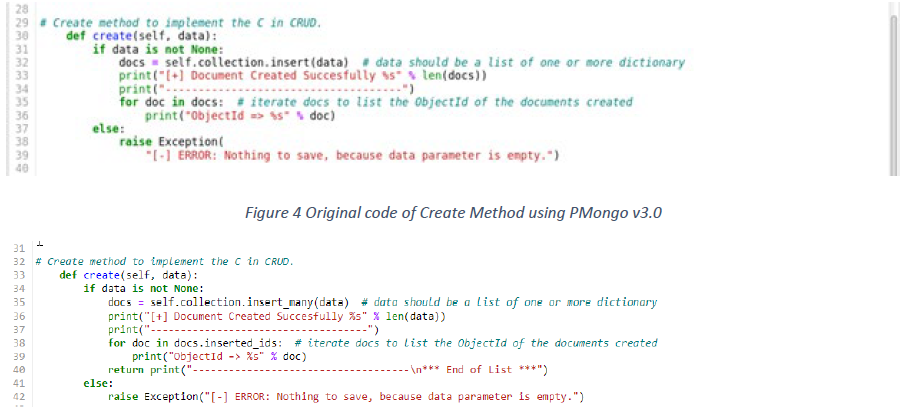
*Figure 2 AnimalShelter Class CRUD Module*

The programmatic implementation of the data structure allows for the effective usage of the stored variable values in various functions and callbacks via the web application. By utilizing algorithmic concepts, suitable computer science techniques, and standards, this method enhances the design and assessment of computing solutions that address a particular problem while managing the trade-offs associated with design decisions.



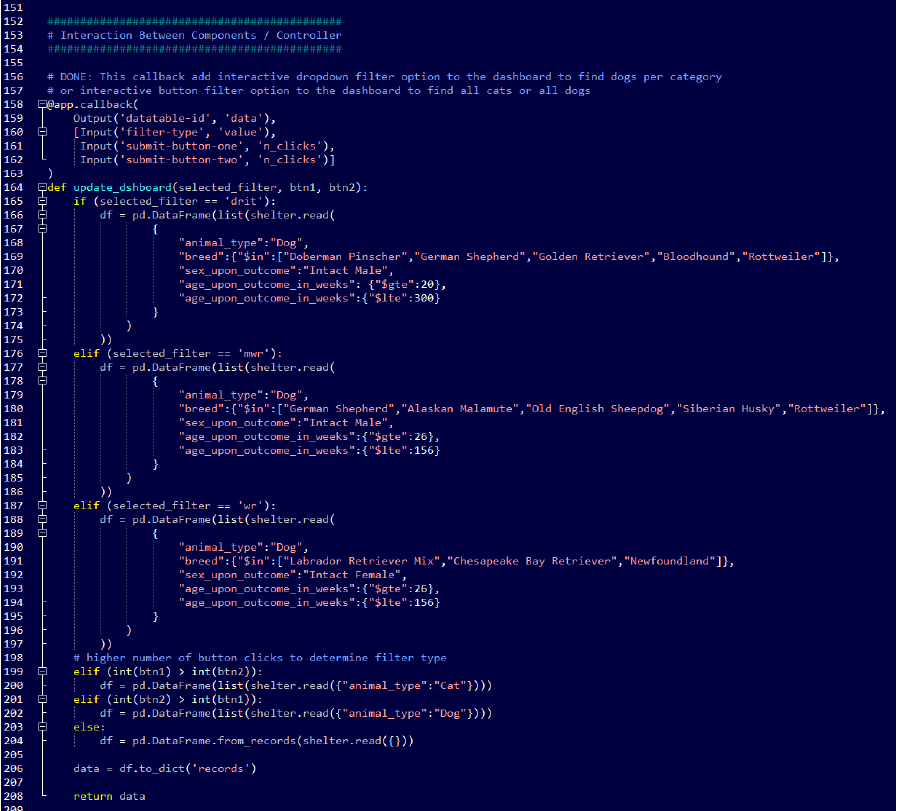
*Figure 3 App Source Code Example Screenshot*

The web application was created on the Apporto Virtual Lab, a university remote desktop Linux platform, hence the artifact enhancements concentrated on re-creating the web application in a Windows OS environment. I use the original documentation for the online application to follow the instructions on how to replicate the web application environment and update/revise the documentation based on the actions taken to run the web application in a Windows terminal. Because others put up the original Linux environment in which the web application was developed, the process was difficult. In our situation, before beginning to follow the instructions in the web application documentation, we must first complete the Python and MongoDB setup process. The setup procedure and the source code functionality update due to the PyMongo driver's latest version and the utilization of an upgraded MongoDB platform enhanced my capacity to apply sound and creative methods, abilities, and resources in computing practices to put computer solutions into practice that add value and achieve industry-specific objectives.



*Figure 5 Modified Code of Create Method Using PyMongo v4.0*

I use industry-standard Python code best practices and approaches in this artifact, like in-line comments, proper naming conventions, and formatting that complies with coding standards, to make the code easier to read and improve the organization of the application code. The computer code adheres to industry-defined formatting best practices, including indentation in accordance with relevant coding standards, and is simple to understand. The source code is uniformly formatted, including line breaks, and is well-structured and consistent in style. We employ proper syntax and rules based on best practices and programming applications. Given that the developed data structures are programmable, the values of the stored variables can be effectively utilized in different ways. Because they describe actions being taken on something, method names are verbs. Every scenario, including ELSE or DEFAULT clauses, is handled by an IF-ELIF block.



*Figure 6 App Source Code IF-ELIF Example*

Being a document database, MongoDB makes it simple to store both structured and unstructured data and offers the ability to manage large volumes of data. Documents are stored in a format similar to JSON. I don't have to worry about normalizing data because this format maps straight to native objects in contemporary programming languages, so it was an obvious choice. The web application has an easy-to-understand CRUD. It took longer to create the dashboard using the Dash framework, though. To create a simple and effective coding structure, comprehend and investigate how the dash core, HTML elements, and callbacks function. In light of the framework's extensive growth and the ongoing development of its libraries, periodic installation and upgrades are required. After the online application is replicated in a different environment, I update and materialize the documentation to help others replicate the web application in a Windows or Linux environment. The concise and thoroughly documented guide enhances my ability to create, develop, and present expert written and visual communication material that is technically sound, logical, and suitably tailored to certain audiences and situations.

As demonstrated in the brief video, I successfully recreated the online application within a Windows environment. Beyond the initial Linux environment, I was able to create a functional application, which necessitated investigating my work in other languages in order to create an artifact that could run on several operating systems, such as Windows and macOS.

References

“Southern New Hampshire University. CS499 Final Project Guidelines and Rubric. Retrieved from Module 1-3 Final Project Review: [https://learn.snhu.edu/content/enforced/1014915-CS-499-T4547-OL-TRAD UG.22EW4/Course%20Documents/CS%20499%20Final%20Project%20Guidelines%20and%20Rubric.pdf?\_&d2lSessionVal=sZSvR1M8MBrfiTVb6OZOu3IQB&ou=1014915](https://learn.snhu.edu/content/enforced/1014915-CS-499-T4547-OL-TRAD%20UG.22EW4/Course%20Documents/CS%20499%20Final%20Project%20Guidelines%20and%20Rubric.pdf?_&d2lSessionVal=sZSvR1M8MBrfiTVb6OZOu3IQB&ou=1014915)”.

“Southern New Hampshire University. Milestone Four Guidelines and Rubric Enhancement Three: Databases. Retrieved from Module 5-2 Milestone Four: Enhancement Three: Databases: <https://learn.snhu.edu/d2l/common/dialogs/quickLink/quickLink.d2l?ou=1014915&type=coursefile&fileId=Course+Documents%2fCS+499+Milestone+Four+Guidelines+and+Rubric.pdf>.”