**CS 499 Code Review Script — Stacey Griggs**

Hi, my name is Stacey Griggs, and this is my code review for CS 499.

For this milestone, I will be walking you through one of my earlier projects, reviewing the code, and explaining how I plan to enhance it in three areas: software engineering and design, algorithms and data structures, and databases.

**Structure**

Does the code completely and correctly implement the design?  
Yes, the dashboard meets the original goals: displaying shelter data in a table, pie chart, and map, with filtering options.

Does the code conform to any pertinent coding standards?  
Mostly, but some parts of the code need better formatting and consistent naming to align with Python and Dash best practices.

Is the code well-structured, consistent in style, and consistently formatted?  
There are areas that could be cleaned up, especially in layout and callback structure. I plan to refactor and organize those.

Are there any uncalled-for or unneeded procedures or any unreachable code?  
No major unused code. Some of the functions are outdated and could be refactored. I also included test print statements that could be removed.

Are there any leftover stubs or test routines in the code?  
A few pieces of placeholder code (like old div IDs or empty layout blocks) still exist. These will be cleaned out in the enhancement.

Can any code be replaced by calls to external reusable components or library functions?  
Dash components are used well, but layout code could be streamlined using helper functions.

Are there any blocks of repeated code that could be condensed into a single procedure?  
Callback functions repeat some logic, I will look at simplifying them.

Is storage use efficient?  
Yes, but queries could be more efficient by retrieving only needed fields, this will be part of my enhancement.

Are symbolics used rather than “magic number” constants or string constants?  
There are some hard-coded strings (like column names) that could be moved to constants to make it easier.

Are any modules excessively complex and should be restructured or split into multiple routines?  
The main layout and some callbacks are long and need to be broken up into smaller, more manageable parts.

**Documentation**   
The code is well commented but will need reviewed after enhancements.

**Variables**  
Variables have clear and descriptive names. There are no major type issues since Dash handles most data types

**Arithmetic Operations**  
The app does not perform complex math or floating-points. It mostly displays and filters data.

**Loops and Branches**  
No traditional loops or case statements are used; logic is handled mostly through Dash callbacks. Some callbacks could be simplified. Logic is easy to follow and works for layout updates based on user input.

**Defensive Programming**

Error Handling and Input Checks:  
Currently, user inputs for dropdowns and radio buttons are not validated for unexpected values.   
The database queries do lack error handling if no data is returned.  
My enhancement plan will prevent empty queries and will add logging errors when things go wrong.

**Software Engineering and Design**

What the code does now:

This project is a Python dashboard built using Dash. It connects to a MongoDB database and allows users to explore animal shelter data. Users can filter animals by rescue type and view the results in a table, a pie chart, and a map. There is also a logo, some layout styling, and interactive options for users.

The dashboard works, but there are a few things I would like to improve. First, the layout could be cleaner. Some of the styling is inconsistent. The pie chart and map are responsive, but I want to organize how everything fits together better.

Also, the code does not include any user login or security when connecting to the database. Anyone running the code could access the data without credentials. That is not great from a software design perspective.

Some of the callback functions are long and could be improved. There is no real error handling either, if the connection fails or the data is empty, the app just loads blank.

Enhancement plan:

I want to add a login screen so that users need to enter their MongoDB username and password before they can access the data. I will use environment variables to store the credentials, instead of hardcoding them.

I will also clean up the code structure for the layout and callbacks to make them easier to read. These changes will show my skills in secure programming and modular design.

**Algorithms and Data Structures**

What the code does now:

Right now, the dashboard updates the charts and map automatically when a user selects options in the radio buttons or the dropdown menu. Every time you change something, the page tries to update instantly. This is not the best approach, and it does not work the best if users want to select multiple options before submitting.

Enhancement plan:

For this enhancement I am going to add a Submit button. Instead of updating right away, the app will wait until the user finishes choosing filters and clicks the button. I will use Dash’s State inputs with the submit button’s click as the trigger.

This makes the logic more controlled. It also improves the user experience by avoiding partial updates. It shows that I understand event-driven programming and how to handle state and inputs in a more scalable way.

**Databases**

What the code does now:

The app connects to MongoDB using a custom AnimalShelter class. It reads data from the database using the read() method with some filters for breed, animal type, and more.

In the original project, I did not finish adding user authentication for connecting to the database. It also pulls all the data even if only a little is needed, which slows things down. There is no check on the user’s input, and there is no error message or logging if something goes wrong.

Enhancement plan:

My enhancement will be adding user credentials for the database connection. I will update the class to accept a username and password to add security.

I will add basic input sanitization and error handling so that the app gives clear messages if the query fails. These changes will improve security, performance, and reliability.

Conclusion

This project already connects the front-end dashboard with a database and shows filtered animal data using visual tools. But I plan to improve the structure, security, and user control with simple enhancements.

I will improve the UI, add login authentication, use a submit button for better control, and clean up how I query and handle data.

Thanks for watching my code review and I am looking forward to continuing this work in CS 499!