

**CS499: Computer Science Capstone**

**Milestone Four**

Buddy Marcey

[Buddy.marcey@snhu.edu](mailto:Buddy.marcey@snhu.edu)

Southern New Hampshire University

December 1, 2024

## Enhancement Three: Databases

### **Briefly describe the artifact. What is it? When was it created?**

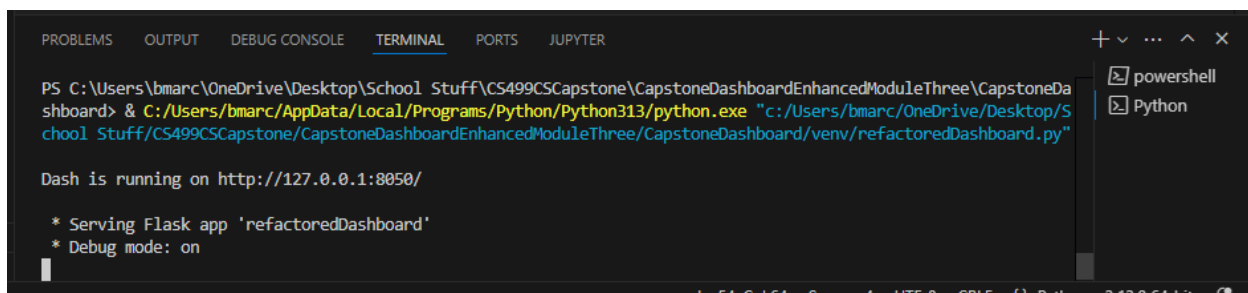
This artifact is the same as my first enhancement. It is a Python dashboard app that links to a database to present data to the end user. I created it during CS340 in August 2024. It uses a MongoDB database, a Python CRUD module as middleware, and a Plotly Dash dashboard for presentation.

### **Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in software development? How was the artifact improved?**

This artifact allows me to demonstrate proficiency in both database and design. I modernized the structure to bring it from a static dashboard working in a local system to a dynamic program that links to a database in the cloud, where it can be accessed from anywhere. The original used abstracted MongoDB implementation that was provided for us without us having to set it up; the current version is running on an EC2 instance on AWS. The middleware was refactored to account for this move. Finally, the dashboard was updated to provide a more dynamic user experience; the original has radio buttons that initiate prebuilt database queries, while the enhanced version allows the user to search for specific entries. The first goal was to update the implementation; the first line when running the original program says:

```
[1]
... c:\Users\bmarc\AppData\Local\Programs\Python\Python313\Lib\site-packages\dash\dash.py:579: UserWarning:
    JupyterDash is deprecated, use Dash instead.
    See https://dash.plotly.com/dash-in-jupyter for more details.
...
```

This meant that I would need to rebuild the program to use the updated version of Plotly Dash. Since there was no real reason for it to still be implemented in Jupyter Notebooks after enhancement, I built the new dashboard program as its own .py file, leaning on the Plotly Dash documentation to get it set up and running. The statically assigned radio buttons were removed, to be replaced with a custom search. Running the enhanced program now gives no deprecation warnings:



```
PS C:\Users\bmarc\OneDrive\Desktop\School Stuff\CS499CS\Capstone\CapstoneDashboardEnhancedModuleThree\CapstoneDashboard> & C:/Users/bmarc/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/bmarc/OneDrive/Desktop/School Stuff/CS499CS\Capstone\CapstoneDashboardEnhancedModuleThree\CapstoneDashboard/venv/refactoredDashboard.py"

Dash is running on http://127.0.0.1:8050/

* Serving Flask app 'refactoredDashboard'
* Debug mode: on
```

**Did you meet the course outcomes you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?**

The main course outcomes for this module are 1) Employ strategies for building collaborative environments that enable diverse audiences to support organizational decision making in the field of computer science, and 2) Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals (software engineering/design/database). These are demonstrated primarily through the development of this application as a whole; the idea is to provide a non-technical audience with a way to interact with data in a database without having to know how Python or MongoDB work. I believe my work on this application demonstrates an ability to meet these objectives.

**Reflect on the process of enhancing and modifying this artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

The documentation for Plotly Dash, along with the video supplement, helped me better understand how it works. The code for the main dashboard was largely provided for us during CS340 and the database implementation was completely abstracted. I have a much better understanding of how it works now, and the documentation goes over everything one might want to do in a dashboard in exhaustive detail. I ran into a couple of issues along the way; at first, my dashboard file was named dash.py, but that caused a circular import issue when trying to import the dash components. Another issue I have is that when I first initialize the dashboard, I get two errors on children callbacks from the program; the map and pie chart both load properly, but I get an error on initialization because it is attempting to callback data from an empty set, and is therefore attempting to access an index out of scope. It does not prevent the components from working, but I need to find out why they attempt to callback the data before the implementation; I likely have something working out of order. I may also attempt to write a custom search function that doesn't rely on the Plotly components to query database items, although that search capability is working properly on its own. One final enhancement might be to obscure some of the data from the table; right now, the output table shows all columns from the database, but the end user doesn't likely need geographic components to be shown, since there is already a map. The currently working model looks like this, after searching for Labrador:



©2000 Hasbro Entertainment Company. All Rights Reserved. Hasbro and the Hasbro logo are registered trademarks of Hasbro Entertainment Company. The characters and likenesses of the characters are trademarks of Hasbro Entertainment Company. The characters and likenesses of the characters are trademarks of Hasbro Entertainment Company.

[illegible]

*Staphylococcus aureus*



- estimate (interval 95%)
- estimate (interval 95%)
- estimate (interval 95%)

