**Enhancement 2 Narrative**

The artifact I worked on for this enhancement is the service layer of my Python Dash dashboard, which interfaces with MongoDB through my CRUD module. This component was originally created in an earlier course to support queries for the Austin Animal Center dataset and display results in the dashboard’s table and metrics. The original version was functional but limited: it used basic find queries without optimizations and recalculated aggregations each time. In the Capstone, I revisited this artifact to strengthen its efficiency, scalability, and algorithmic rigor.

I selected this artifact for my ePortfolio because it highlights my ability to apply algorithms and data structures in a practical way to improve application performance. The enhancements demonstrate key skills: implementing keyset pagination for efficient query traversal, designing compound indexes aligned with frequent filters, and building caching mechanisms for repeated aggregations like top breeds and age histograms. These improvements show how I can go beyond simply making an application work to making it robust, scalable, and performant. The refactoring also showcases my ability to think critically about trade-offs in algorithmic design.

Yes, I met the outcomes I planned for this category in Module One. I specifically wanted to demonstrate algorithmic thinking and efficiency improvements, and I achieved that through caching strategies, index creation, and normalized signatures for cache keys. These directly map to the outcome of designing and evaluating computing solutions with algorithmic principles and trade-offs. At this point, I do not need to revise my outcome-coverage plans, because this enhancement clearly addresses the goals I had set for the algorithms and data structures component of the Capstone.

Through the process of enhancing and modifying this artifact, I learned how much impact indexing and caching can have on performance, particularly in data-heavy applications. I also deepened my understanding of keyset pagination and why it scales better than offset-based approaches. One of the main challenges I faced was dealing with conflicting Python packages and ensuring that the correct versions of PyMongo and its bson library were installed, which required methodical debugging. Overall, the experience reinforced the importance of not just solving problems but solving them efficiently and with attention to maintainability.