Briefly describe the artifact. What is it? When was it created? The Grazioso Salvare Animal Shelter Dashboard is a web-based application that was built through Dash and MongoDB. This was created in CS340 during the September to October term. This database is meant to assist in managing rescue animal data. This interactive dashboard enables users to filter rescue animals based on their rescue type, their geolocations, and be able to analyze through age distribution through their breed. This application was designed to enhance decision-making for an animal shelter through providing real-time insights into their database. The project was trying to follow a Model-View-Controller architecture, where the model was the MongoDB database that contained the data, the view would be the dash-based web interface that had the dropdown filters and visualizations, and the controller was the backend logic for the data retrieval and visualization.

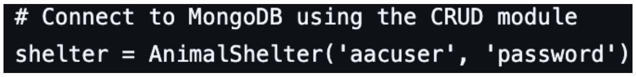
Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in algorithms and data structure? How was the artifact improved? This artifact has been included in the ePortfolio because it is able to showcase expertise in multiple key Computer Science competencies:

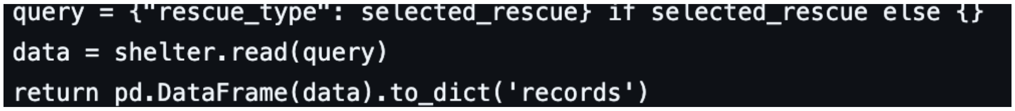
- Data Structures & Algorithms:
    - It is able to efficiently retrieve and filter through data from MongoDB based on the selections of the user.
    - It can utilize query optimization techniques to improve database performance.
- Software Engineering & Web Development:
    - There is an implementation of a Dash web framework for a more interactive UI component.

    - And it demonstrates the MVC principles when structuring the application.
- Database Management & Optimization:
    - There is a connection to MongoDB for persistent storage.
    - Implements query filtering and projections to optimize the performance of the program.
    - Uses CRUD (Create, Read, Update, Delete) operations to interact with the database.
- Data Analytics & Visualization:
    - It can display the geolocation data through scatter map which shows the animal rescue locations.
    - It provides a bar chart visualization to analyze age distribution by breed.

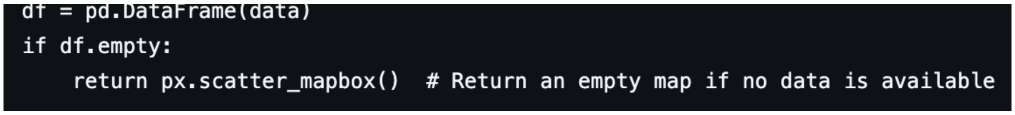
Including this artifact in the ePortfolio highlights, in my opinion, a strong representation for technical skills in data-driven application development, full-stack web engineering, and database management. This artifact is meant to represent a real-world demonstration of how data structures, algorithms, and visualization techniques are used to solve any practical problems that could realistically occur in the real world. There are many improvements that should be made to the code and the program altogether, but the above states what the code is meant to do, and what it intends to do.

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? There can be several enhancements that can be implemented to improve the original artifact, because there are many gaps within the code that needs to be improved upon, especially on factors such as, security, performance, usability and visualization clarity.

- Security Enhancement – for the use of environmental variables for database credentials
  - The issue with the original code is that the database credentials were hardcoded in the source code, the enhancement will make sure that the hardcoded credentials would be removed with environment variables for better security, which prevents exposure of any sensitive information in the source code repositories.
    - 

      ```
      # Connect to MongoDB using the CRUD module
      shelter = AnimalShelter('aacuser', 'password')
      ```
    - **Enhancement**:
      - import os
      - shelter = AnimalShelter(os.getenv(‘Mongo_USER’), os.getenv
- Query Optimization for a Better Performance
  - The original code retrieved all fields from MongoDB which led to an unnecessary data transfer. The enhancement would use MongoDB projections to fetch only the necessary fields, which will reduce data size, improve query performance and reduce processing time.
    - 

      ```
      query = {"rescue_type": selected_rescue} if selected_rescue else {}
      data = shelter.read(query)
      return pd.DataFrame(data).to_dict('records')
      ```
    - **Enhancement**:
      - query = {"rescue_type": selected_rescue} if selected_rescue else {}
      - data = shelter.read(query, projection={"_id": 0, "name": 1, "breed": 1, "age": 1, "rescue_type": 1, "location": 1})

- Handling Missing Data in Visualization:
    - The issue is that if the dataset was empty, then visualizations could return blank figures which will lead to a bad user experience. The enhancement would add fallback visualization with a clear message where there isn't any data available, which will improve user feedback and prevent any UI confusion.
        - 

          ```
          df = pd.DataFrame(data)
          if df.empty:
              return px.scatter_mapbox()  # Return an empty map if no data is available
          ```
        - **Enhancement:**
            - if df.empty:
                - return px.scatter_mapbox(
                -     title="No Data Available",
                -     lat=[0], lon=[0], zoom=1
            -     )
- UI/UX Improvements:
    - With UI/UX, there can be multiple enhancements that can be made to ensure that the experience with this program is sufficient.
        - Adding search and filtering options in the DataTable for there to be better usability.
        - Implementing different types of styling to improve the appearance of the dashboard.
    - Despite UI/UX enhancements not being seen as important as other factors that are also involved, I think that making improvements on UI/UX can improve the general interactions that users would have with the program because there can be a positive interaction in every aspect.

Within these enhancements, I was able to meet different course outcomes:

- Software Engineering and Design, I was able to design and develop and improve an interactive web application that was able to efficiently retrieve and visualize data. Through refining the UI and improving the database queries, I think that I was able to demonstrate my abilities to implement software solutions that can deliver industry value.
- Algorithms and Data Structures, I optimized the database queries using efficient filtering methods that can ensure quick data retrieval. In addition, I think handling missing geolocation data required quite a bit of designing an approach to prevent any application crashes while also maintaining visualization accuracy.
- Databases, through my understanding of NoSQL databases, I think that enhancing this artifact helped me think more in depth about data integrity and different optimization strategies.
- Security, I clearly didn’t prioritize this in the first round for this project because I didn’t really consider proper authentication. But now, I have considered security aspects by ensuring that there are proper methods in place when connecting to the database and safeguarding all of the data from any unauthorized access.

Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face? Enhancing and modifying different areas of the animal shelter database artifact was an important learning experience that helped me grow a better understanding of software design, algorithms and databases. Throughout the development, I was able to finetune my skills when it came to, implementing a functional and user-friendly dashboard through using the different platforms, while also making sure that the application was effectively filtered and could present the data appropriately. A vital part of my learning experience was when I had to improve the database queries to optimize performance. Initially, querying for rescue types was straightforward and understandable but as the dataset increased in size, there had to be more efficient filtering mechanisms to ensure that there would be a quick retrieval of relevant records. Another area of growth would be when I enhanced the UI and ensuring and ensured that users would be able to easily interact with the data. I tried to improve the usability of the dropdown selection and refine different parts that users would interact with. I think the biggest challenge that I faced was trying to ensure that there would be a seamless interaction between MongoDB and the Dash web application. I had to overcome many problems when the data wouldn't conform to the expected/planned structure which required more thinking and processing than necessary. Also trying to debug certain functions in Dash was quite difficult, because there were many components that were involved, and trying to ensure that there is constant smooth communication between the variables proved to be no easy task. With these enhancements that I have attempted to make, I think I have achieved a better amount of progress than my last milestone. I think I was confused in the beginning of this course of the expectations but now that I have seemingly understood the different areas of this course, I am excited to explore further about improvements that can be made to my artifact.