The artifact I chose was a project that incorporated python, mongodb, and jupyter. I was tasked with creating a website dashboard that could retrieve data from a mongodb. The artifact incorporated data that the company gave me on their rescue animals. I needed to create the dashboard with a username and login and its ui, make the ui easy and appealing. I also had to create the data entries and have them be retrievable online. Ultimately, while the process progressed I wasn’t able to get my dashboard up and running. Early tests with the dashboard saw the creation of a dashboard UI with a working login and password. When I utilized jupyter to connect the Mongodb the dashboard stopped loading and the data did not populate. I am reviewing the dashboards UI and the code that creates the interaction between the dashboard and the database. I created this application about 16 weeks ago. I felt it deserved to be in here because it utilized a lot of different programs, python, mongodb and jupyter notebook. I selected this piece because I put a lot of time into it and while the code looks great the dashboard never loaded after the inclusion of the database interaction. Even though the data was created in mongodb and I am able to retrieve it locally and even though without the database interaction the dashboard loaded and utilized a username and password, together I have an error that I am working to correct. I am going to rewrite the data structure and the dashboard code to showcase that I can develop this application. So far I have improved the dashboards ui code. I have condensed and cleaned up the interactive features to make it more user friendly and get it to load. My next update will be to the improve the interaction between the components and the controller by rewriting the code to filter interactive data table with MongoDB queries. As of now my biggest challenge has been determining why the addition of my interactive data filters caused my dashboard not to load. I intend to get the dashboard to load and appropriately filter the database.

The next update involved reviewing the code for the database and controller. I made a few changes to the query method. The method uses the and or callback for the data retrieval. I specified a few searchable traits, breed, gender, and age. The and or function allows for combinations of specific details to narrow down the dog search. I also realized that my app callback was missing a line for my viewport graph. I had somehow omitted the viewport selected rows so, on the def update\_graphs nothing was happening. The new changes allow fof querying of ym database and the abnility ot generate a user px.pie chart that will break down the query into percentages.