The included ScreenCast shows the functionality of the Grazioso Salvare Filtering web dashboard. But in words the functionality of the project is to enable Grazioso Salvare to use pre-set filtering options to check the Austin Animal Center for potential dog training recruits and to display it through a graphical web-based user interface.

MongoDB was used as the model component of the development for multiple reasons, one being Document oriented storage. There is also its ability to “play nicely“ with Python through many different avenues. Which include an official MongoDB driver (“PyMongo”), the support of multiple frameworks and APIs which in turn support web development like this project’s use of the Dash framework and extensive library support. There are other reasons such as MongoDB’s data schema flexibility that enables developers to continue working on a project as the database is still being built and organized.

The Dash framework provides this project with view and controller structure as a Python framework that simplifies the act of creating a user interface for web applications without requiring the developers in question to use JavaScript, HTML, or CSS. Dash does this by acting as a library of functions that when used acts as an abstraction of the three previously mentioned languages and enables UI creation and UI interactivity through the enabling of controller logic through callback functions.

The steps taken to complete this project included:

1. Importing the AAC CSV file into a MongoDB database
2. Creating a new user for said database
3. Creating a CRUD module to interface with MongoDB in Python
4. Reviewing the Specs of the Project and building towards those specs
5. Testing and Debugging

Challenges:

Some of the more notable challenges included:

1. The first implementation of the CRUD Module could read data from the database fine. However, querying properly was another matter and led to a few headaches it was finally solved by tweaking the function to take projections or filter qualities beyond a simple query. Without this change it seemed to fail when querying even if no projection was needed.
2. The Pie Chart…. The visualization element was a headache for awhile and one of still some concern but works as intended just displays poorly when fed the default unfiltered data frame. Was solved by changing the data frame and the create pie chart function to hold a Boolean for an “aggregation check”.

Links:

<https://dash.plotly.com/>

https://www.geeksforgeeks.org/