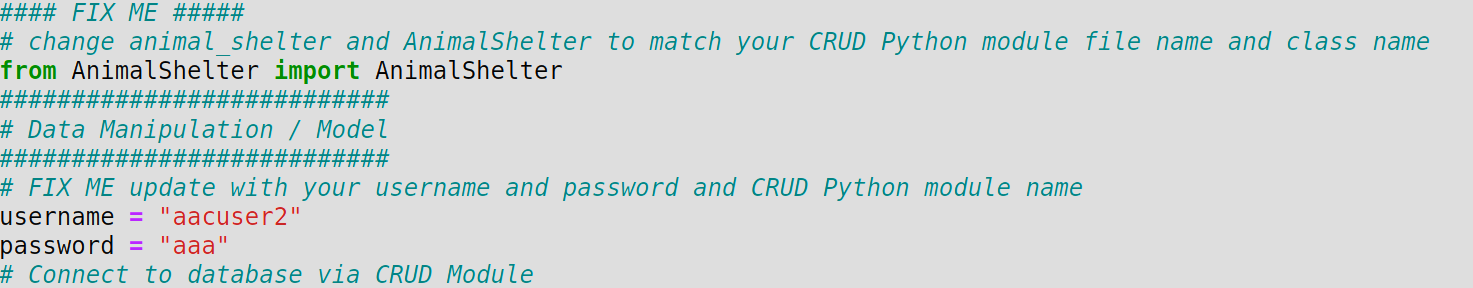
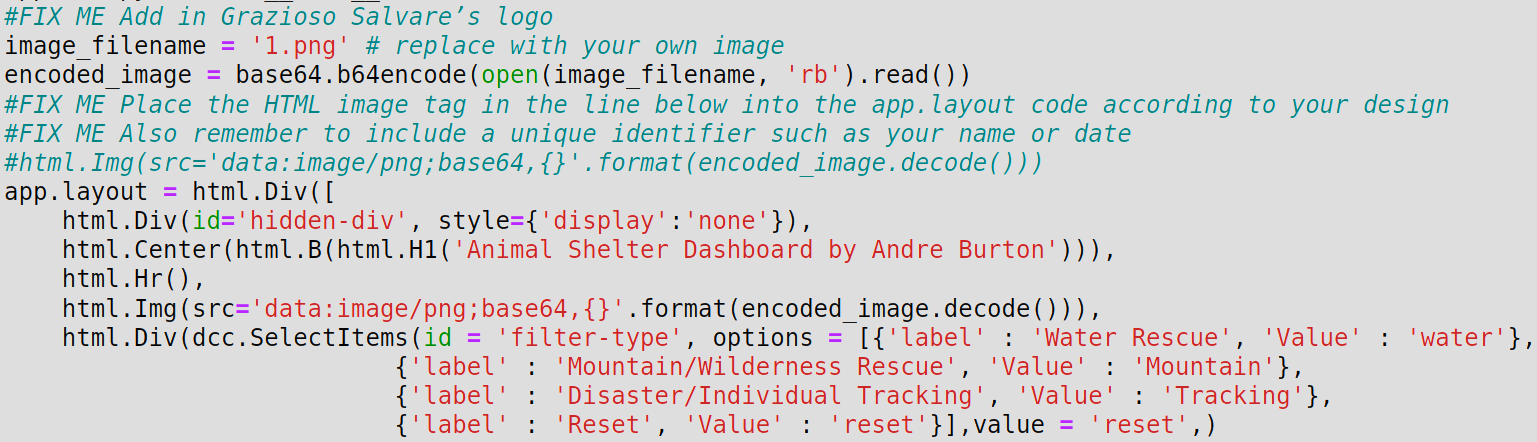
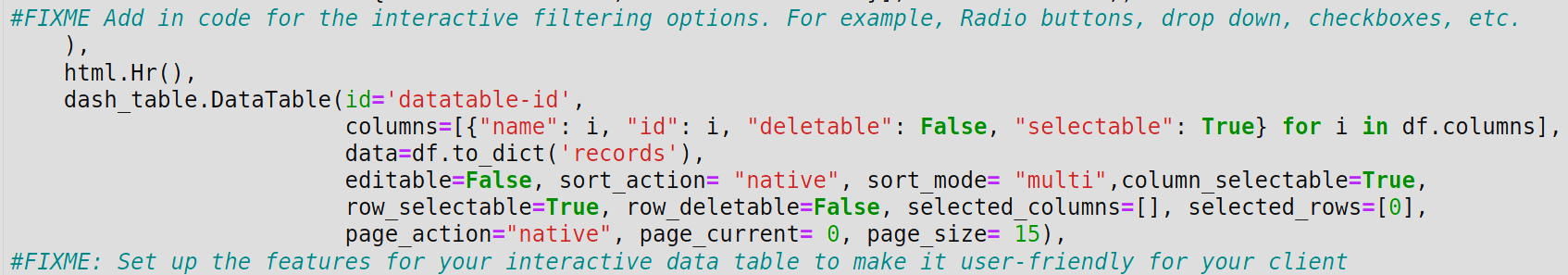
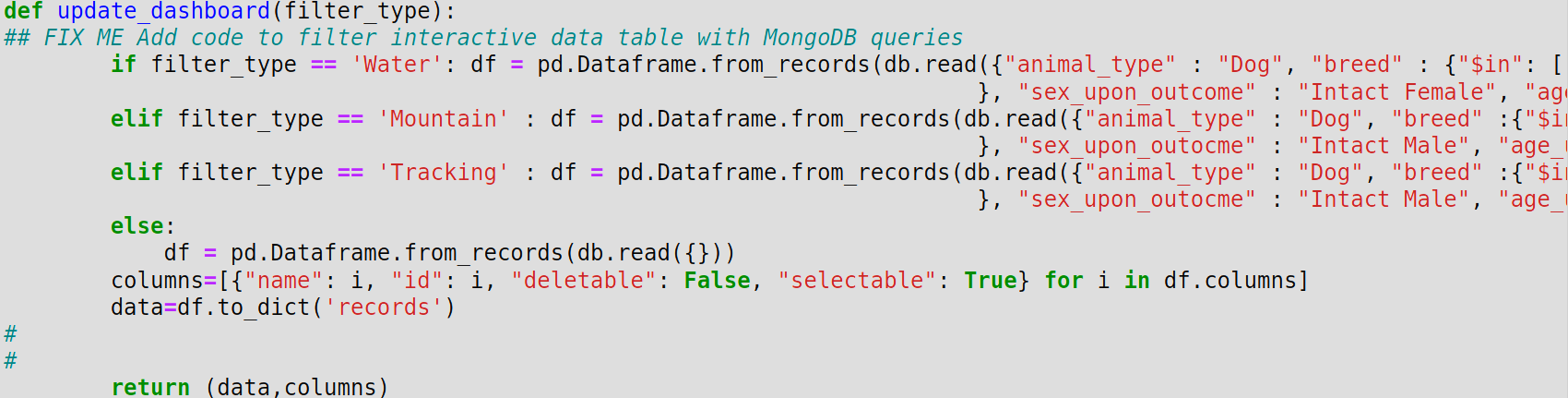
**Required Functionality**

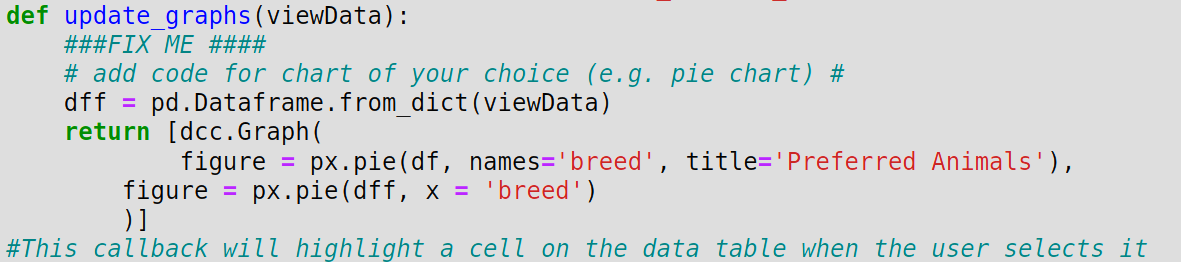
I was unfortunately unable to get the program to run correctly due to server timeout errors, that I was not able to fix and not sure how to fix for future wise. Below I will post screenshots of the code written for each part(FIX me parts) of the functionality to work if the server had proper connection.

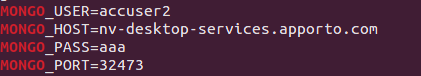










****

**Tools Used**

The tools needed in order to run this properly are MongoDB which holds the database information and allows for connections when being called by python which is important because python is able to connect to web interfaces and mongoDB provides the database information when called.. The next needed tool is something that runs python which in this example jupyter notebook was used. You will then need to create a .py and a .ipynb files the .py file is where the CRUD is initiated and the .ipynb is where the application page will be called from. Next you will need some libraries in order to have everything communicate properly one of them being Dash which is an open-source python framework which assists in building the web application in our project. The reason Dash is so important is because it is used as a model view controller where the model is the data source, the view is the layout of the dashboard, and the controller connects the model and the view which is exactly what we need in this web application (Dash, 2023). To set up Dash you can go to there website at “[Installation | Dash for Python Documentation | Plotly](https://dash.plotly.com/installation)”. Some hardware tools needed are a computer that can connect to the internet and some sort of internet provider.

**Steps Taken**

The steps below will be simple but you can go back to the images above to get a better understanding of the indepth steps.

1. Open MongoDB and import the proper file in this example aac\_shelter\_outcomes.csv was used.
2. Then you need to create an admin account setting up important information which will be used later such as username, password, database, etc. this will allow you to log into your own account when running the web application.
3. Next you will need to set up your .py(CRUD) and .ipynb(web application) file, input the needed information.
4. An extra step is also to create a testing application to verify that the CRUD model is running correctly.
5. Lastly you can run your .ipynb file and it should output a URL to follow which will show you your completed website.

**Challenges Encountered**

I had numerous challenges when setting up this project, most of them being small and easily fixable. The small challenges consisted of coding errors that the program was able to let me know of. The single big challenge I had which was mentioned earlier was a server timeout issue which prevented me from connecting to the server which in turn prevented the web application from loading properly. I did some research and made some changes to see if it would fix the issue but unfortunately the issue was ongoing and I'm sure it is something small I'm forgetting to do, I will continue to work on it and hopefully get it to run. I will update this if a successful run is able to be processed.