# README

## About

This project should be able to perform basic CRUD operations on a MongoDB database using Python functions then interface the results into an live, interactive data table and charts that the client side will be able to filter through and read. The title of this project is “PyMongo Dashboard”

## Motivation

This project exists to meet the client needs of Grazioso Salvare and allow them to have a dashboard which lets them easily filter and locate rescue animals based on need.. It also serves as a basic framework for a dashboard utilizing Pymongo, Dash and MongoDB that could be easily repurposed for different data.

## Installation

Things you will need to install will be MongoDB, PyMongo library, Dash framework, Jupyter Notebooks, a linux distribution and the github repository containing the files needed for the program. Once this is all installed you have to make accounts for authentication and make sure that the account you plan on using has at least read access to the database. Then you will want to import the database you want to work with. Next you will need to upload the files you got from the Github repository namely the .py and .ipynb files into the Jupyter notebook. Once all of this is done you should be able to select run from the Jupyter notebook and get the results.

**Tools Used**

The main tools used here were MongoDB, the Dash framework, PyMongo and Jupyter Notebooks. I chose MongoDB because it is a nonrelational database and therefore it lends well to tasks like this where the individual items in a database are not related to or dependent on each other. The animals in the database don’t need each other to make sense and they aren’t directly related in any way. I Jupyter notebooks is a way to run and test python code. I used this because it is the best way to interface PyMongo code with the MongoDB database while still being able to see the results of my work all on the same page. I chose PyMongo because it is a widely used method of writing code that is based on Python so it includes things like functions and variables but it also is able to accept MongoDB queries so this was the perfect choice when trying to retrieve data from MongoDB using Python script. Lastly the dash framework allows for us to use things like radio buttons, sliders and other interactive features that make a user interface more enjoyable and easy to use. I chose to go with this because for the purposes of this assignment I simply needed interactive radio buttons and the Dash framework has a very simple way of implementing that.

**Challenges**

The primary challenge that I had to overcome was to make sure that everything synced up and worked properly. I had difficulty getting the PyMongo code to connect to the MongoDB database. I was able to overcome this by consulting with my teacher and realizing that I had to be more careful with the database URL and password. I also had some issues with getting the .ipynb file to run properly and careful study of the Jupyter documentation helped to fix that issue. The last issue I would like to touch on was getting the queries to return the right types of data but this was resolved by reviewing the specification documentation and making sure that I accounted for their requirements and structured my queries properly.

### Code Example (Water Rescue Query)

def queryWaterRescue(data):

client = pymongo.MongoClient("mongodb://localhost:52382/AAC")

database = client

collection = database["animals"]

query = {"breed": 'Labrador Retriever Mix', "breed": 'Chesapeake Bay Retriever', "breed": 'Newfoundland',

"sex\_upon\_outcome": 'Intact Female', "age\_upon\_outcome\_in\_weeks": {$gte: 26}, "age\_upon\_outcome\_in\_weeks": {$lte: 156} }

mydoc = mycol.find(myquery)

for x in mydoc:

print(x)

## Roadmap/Features

There is currently a connection issue that is preventing the database from establishing a connection to the client. It is being labeled as a timeout error which to me implies there was some sort of connection rather than authentication issue as it doesn’t seem to have even gotten the chance to authenticate.

## Contact

Renaldo Musto