# CS 340 README Template

## About the Project/Project Title

*The project I am working on is a python project for a company called Grazioso Salvare. The project is meant for me to learn how to use python and MongoDB. I needed to create a dashboard that displayed information about different rescue dogs. The dashboard uses real-world information from the Austin Animal Center.*

## Motivation

*The goal of this project is to learn how to use Python and MongoDB. I will also learn how to use databases with MongoDB and how to edit them. It also taught me how to use CRUD with databases and how to add interactive features to them.*

## Getting Started

*First, you need to access the virtual environment.*

*Once logged in, open the MongoDB terminal.*

*Next import the AAC csv file using this command.* *mongoimport --username aacuser --password ‘pass123’ --host localhost --port 27017 --authenticationDatabase admin --db AAC --collection animals --type csv --file /usr/local/datasets/aac\_shelter\_outcomes.csv –headerline*

*Next open the Jupyter Notebook application and run the ProjectTwoDashboard.ipynb.*

## Installation

*You will need python 3.8*

*Pymongo*

*MongoDB*

*And Jupyter Notebook*

## Usage

*Use this space to show useful examples of how your project works and how it can be used. Be sure to include examples of your code, tests, and screenshots.*

*The dashboard allows users to:*

*Filter dogs by rescue*

*View different results in a datatable*

*See the different breeds in a pie chart*

*Select a dog and see where the location of the dog is.*

### Code Example

*# Callbacks*

*@app.callback(Output('datatable-id', 'data'), [Input('filter-type', 'value')])*

*def update\_dashboard(filter\_type):*

*query = {}*

*if filter\_type == 'water':*

*query = {*

*"breed": {"$in": ["Labrador Retriever Mix", "Chesapeake Bay Retriever", "Newfoundland"]},*

*"sex\_upon\_outcome": "Intact Female",*

*"age\_upon\_outcome\_in\_weeks": {"$gte": 26, "$lte": 156}*

*}*

*elif filter\_type == 'mountain':*

*query = {*

*"breed": {"$in": ["German Shepherd", "Alaskan Malamute", "Old English Sheepdog", "Siberian Husky", "Rottweiler"]},*

*"sex\_upon\_outcome": "Intact Male",*

*"age\_upon\_outcome\_in\_weeks": {"$gte": 26, "$lte": 156}*

*}*

*elif filter\_type == 'disaster':*

*query = {*

*"breed": {"$in": ["Doberman Pinscher", "German Shepherd", "Golden Retriever", "Bloodhound", "Rottweiler"]},*

*"sex\_upon\_outcome": "Intact Male",*

*"age\_upon\_outcome\_in\_weeks": {"$gte": 20, "$lte": 300}*

*}*

### Tests

*Test for this project were conducted in Jupyter Notebook. I verified that a document was created using the criteria above.*

### Screenshots

A screenshot of a computer

AI-generated content may be incorrect.

*Testing for water rescue dogs*

A screenshot of a computer

AI-generated content may be incorrect.

*Testing for mountain rescue dogs*

A screenshot of a computer

AI-generated content may be incorrect.

*Testing for disaster rescue dogs*

## Contact

Your name: Daniel Evers