CS-340 Client/Server Development – Project Two README

Eric Slutz

Southern New Hampshire University

Table of Contents

[README 3](#_Toc132472582)

[Dashboard Functionality 3](#_Toc132472583)

[Filter: All 3](#_Toc132472584)

[Filter: Water Rescue 4](#_Toc132472585)

[Filter: Mountain or Wilderness Rescue 5](#_Toc132472586)

[Filter: Disaster or Individual Tracking 6](#_Toc132472587)

[Tools Used 7](#_Toc132472588)

[MongoDB 7](#_Toc132472589)

[Python 7](#_Toc132472590)

[PyMongo 7](#_Toc132472591)

[Dash Framework 7](#_Toc132472592)

[Steps Taken 7](#_Toc132472593)

[Challenges Encountered 7](#_Toc132472594)

# README

## Dashboard Functionality

### Filter: All

This resets the filter and displays all animals in the table, updating the widgets accordingly.

Table

Description automatically generated

Chart

Description automatically generated with low confidence

### Filter: Water Rescue

This filters to display dogs that meet the requirements for water rescue in the table, updating the widgets accordingly.

Graphical user interface, application, table

Description automatically generated

Chart

Description automatically generated with medium confidence

### Filter: Mountain or Wilderness Rescue

This filters to display dogs that meet the requirements for mountain or wilderness rescue in the table, updating the widgets accordingly.

Table

Description automatically generated

Chart, application

Description automatically generated

### Filter: Disaster or Individual Tracking

This filters to display dogs that meet the requirements for disaster or individual tracking in the table, updating the widgets accordingly.

Graphical user interface, application, table

Description automatically generated

Chart

Description automatically generated

## Tools Used

### MongoDB

<https://www.mongodb.com>

MongoDB is a good choice for use with frontend programs. Instead of having to define columns and rows, each record is stored as a document containing all of its own properties and values. This lends itself well to creating interactive dashboards for displaying data.

### Python

<https://www.python.org>

Python was chosen for this project due to its ease of use and the wide availability of support and add on features and libraries.

### PyMongo

<https://pymongo.readthedocs.io/en/stable/>

PyMongo is the recommend driver for using Python to interact with MongoDb.

### Dash Framework

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

Dash is a Python framework that works well for creating web-based dashboards to display data.

## Steps Taken

1. Import animal shelter data into a MongoDB database.
2. Create/use a CRUD module to connect to the database using Python.
3. Build a frontend dashboard using Dash to display documents from the database.

## Challenges Encountered

I encountered many challenges during this process, far more than I should have experienced building this simple dashboard. Some of that is due to a lack a familiarity building a frontend using Python or Dash. Another part is using out-of-date versions of all of these tools and therefore being forced to use Apporto to make sure all the correct versions were used. Apporto can be annoying and slow down the development process, which is why I try to avoid it if possible. By far the biggest challenge was using Jupyter Notebook. It was a terrible experience that made the project unnecessarily more difficult and it’s not clear why it was needed or chosen for this class. These challenges were overcome by pushing through and trying variations until I got it to work the way I wanted.