Daniel Leon

Southern New Hampshire University

CS340

December 14, 2025

**Project 2 README File**

**Dashboard Functionality**

The required functionality of this project is to display data from a csv file via MongoDB in a table created with Jupyter Notebook. The table will have options to filter the data by rescue type and preferred dog breeds. The filters include water rescue, mountain or wilderness rescue, disaster or individual tracking, and reset which returns all widgets to their original unfiltered state. Here are some examples:

Original state

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Water Rescue

A white sheet with black text and numbers

AI-generated content may be incorrect.

A screenshot of a map

AI-generated content may be incorrect.

Mountain or Wilderness Rescue

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Disaster or Individual Tracking

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a map

AI-generated content may be incorrect.

Reset

A close-up of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

I also added filters for the number of rows displayed at a time for a better user experience. Here are some examples:

6 rows

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

4 rows

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Tools for Functionality**

MongoDB is the database used to store the data for this project. It stores data as documents to provide more flexibility for data structures. Here we take a csv file of our data and display it in an organized table.

<http://www.mongodb.com>

Python is the language of choice for this project due to its dynamic data handling capabilities which is great for data analysis. This attribute compliments MongoDB’s ability to handle unstructured data.

<http://www.python.org>

Pymongo is the official Python driver for connecting and interacting with MongoDB for things such as our CRUD application. It can be installed via your terminal by typing *pip install pymongo*.

<http://pymongo.readthedocs.io>

A Jupyter Notebook is a web-based application used to create documents of code, visualizations, equations, etc. We use it here to create a visualization of our data which is displayed as a table on a web-based dashboard. Jupyter Notebook can be installed via your terminal by typing *pip install notebook*.

<http://jupyter.org>

Dash is a Python framework used for building web applications and interactive dashboards. It was created by Plotly so it has access to Plotly’s graphing library to create great data visualizations with ease. It can be used with the Python library Pandas which is used for data analysis and manipulation. Dash, Plotly, and Pandas can be installed via your terminal by typing *pip install dash plotly pandas*.

<http://dash.plotly.com>

**Steps**

First, we import our csv file containing the animal shelter data into our MongoDB database. Next, we create our Python CRUD module that we use to connect to the database to access and manipulate the data. We also set up admin and user accounts for authentication in MongoDB and implement access via our CRUD module. We then import the CRUD module into our Jupyter Notebook and write the code for our Dash Plotly dashboard and widgets. Finally, we launch the application and test the functionality.

**Challenges**

One major challenge I faced with this project was with authentication inside my Jupyter Notebook. To overcome this, I just read up on some examples from Google and watched some Youtube videos to get a better understanding of how everything is supposed to work to achieve this. Another challenge was with having to change the port number when launching the application. Changing the port itself was easy, figuring out that that was the issue was the challenge. I Googled the error code that I received and read multiple articles on the subject to gain a better understanding of what was happening, which led me to the solution.