Ty Wheeler  
CS 340  
2/20/25  
Project Two: ReadMe

## **About the Project**

The Grazioso Salvare Dashboard Project is a client/server application designed to help Grazioso Salvare analyze and visualize shelter animal data to identify and categorize dogs suitable for search and rescue training. The project involves using MongoDB to store and manage data provided by animal shelters and leveraging Dash to create a dynamic web-based dashboard for efficient data visualization and interaction.

The project addresses the need to efficiently filter and select dogs based on specific criteria such as age, breed, and rescue specializations. This streamlines Grazioso Salvare's mission to train dogs for rescue operations.

## **Motivation**

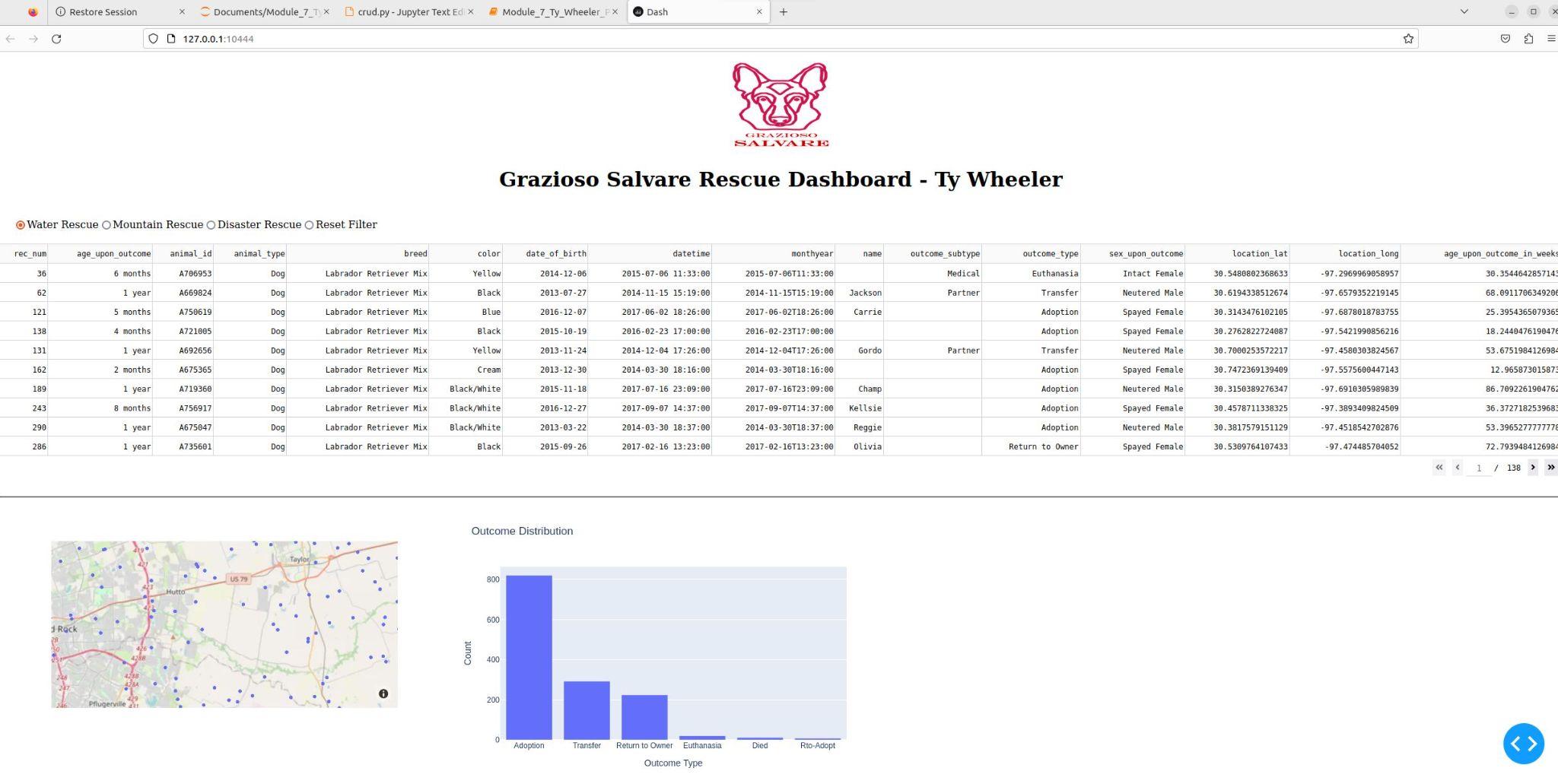
Grazioso Salvare's innovative mission to train search-and-rescue dogs requires an efficient system for analyzing shelter data. This project was developed to automate the process of identifying suitable candidates for training. By organizing and categorizing data in a database and providing a user-friendly dashboard for visualization, the project reduces manual effort and ensures the right dogs are selected for the right rescue operations.

## **Required Functionality**

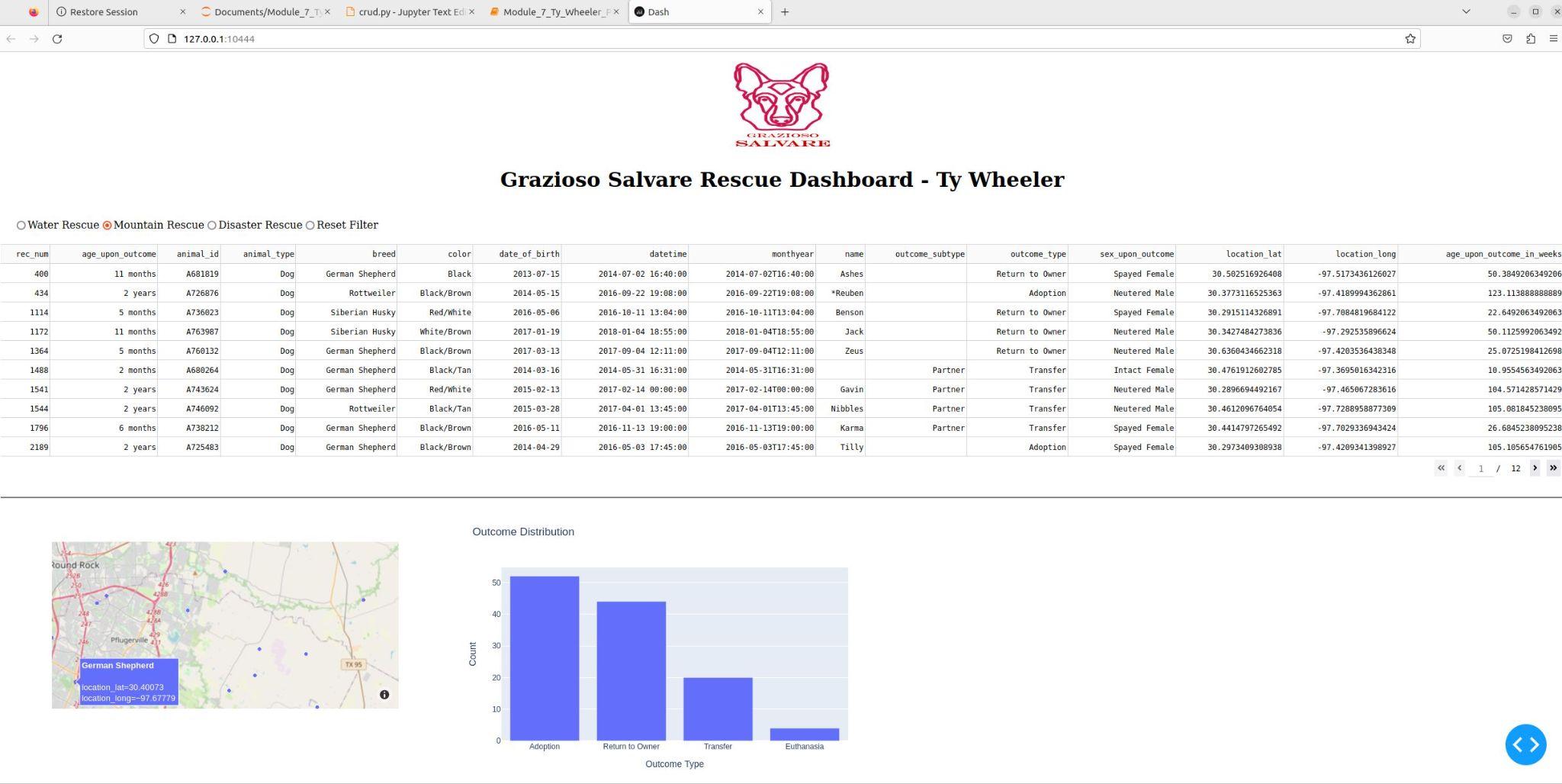
The dashboard enables users to:

* Filter and query animal data stored in MongoDB.
* Visualize data using interactive charts and tables.
* Identify suitable dogs for search and rescue training based on specific criteria.

Dashboard displays relevant selected information using filters above the table.



Upon selecting filters the table, graph and map update with relevant selections.



## **Tools and Technologies**

* MongoDB: Used as the backend database due to its flexibility in handling large datasets and seamless integration with Python. It allows efficient querying and filtering of the animal outcomes data.
* Dash: Used to create the interactive dashboard. Dash is ideal for building analytical web applications with minimal overhead and is well suited for Python based applications.
* Python: Used for scripting and backend logic, enabling seamless interaction with MongoDB and Dash.
* Jupyter Notebook: Used for testing database operations and validating functionality before deployment.

## **Why MongoDB?**

MongoDB was chosen for the project due to:

* Its flexible document structure which allows handling varied shelter data without predefined schemas.
* Its scalability and performance enable fast queries on larger datasets, like the animal outcomes data set.
* Its seamless integration with Python, via PyMongo, allows for straightforward CRUD operations.

## **Why Dash?**

Dash was chosen due to:

* Its ability to create interactive web applications in pure Python.
* Built-in support for data visualization, making it ideal for presenting MongoDB data dynamically.
* Its lightweight framework doesn’t require extensive front end development knowledge.

## **Installation and Setup**

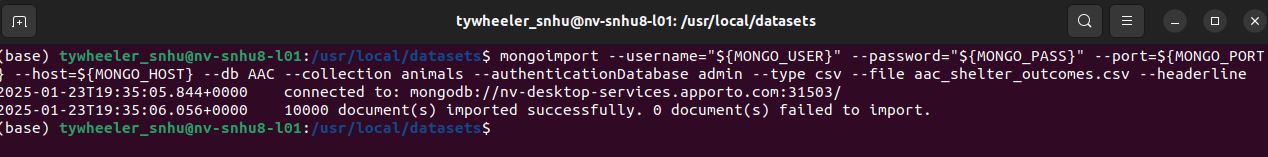
Ensure you have the following installed:

* Python 3
* MongoDB (running locally or on a remote server)
* Jupyter Notebook (for testing and validation)
* Required Python libraries: PyMongo, Dash, Pandas

### **Importing the Database Data**

To import the shelter data into MongoDB, use the following command:

mongoimport --db AAC --collection animals --type csv --file /usr/local/datasets/aac\_shelter\_outcomes.csv --headerline --host nv-desktop-services.apporto.com --port 31503 --username aacuser --password CS340 --authenticationDatabase admin



This command imports the dataset into the AAC database under the animals collection.

## **Running the Dashboard**

1. Clone the repository
2. Install dependencies
3. Ensure MongoDB is running and accessible
4. Run the .ipynb file and select the Dash link it generates
5. The dashboard will appear in a new browser window to use and interact with data.

## **Usage**

The project includes a Python module that performs CRUD operations on the MongoDB database and a Dash powered web application for data visualization.

### **Testing in Jupyter**

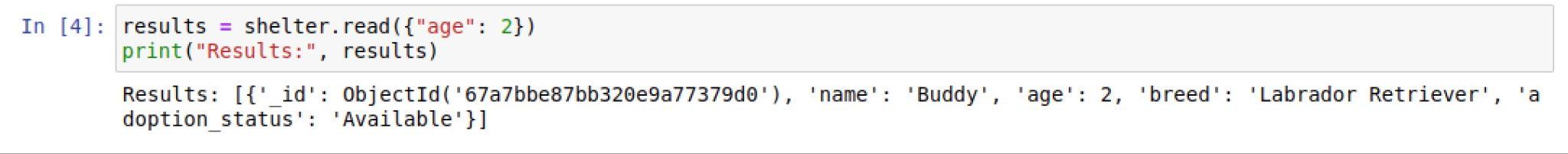
* Open Testing.ipynb in Jupyter Notebook.
* Run each cell to:
  + Connect to MongoDB.
  + Insert sample records.
  + Perform read, update, and delete operations.
  + Validate that the operations execute correctly.

### CRUD Operations

* Create: Insert new animal data into MongoDB.



* Read: Query and retrieve data based on search criteria.



* Update: Modify existing records.



* Delete: Remove records from the database.



## **Steps Taken to Complete the Project**

1. Set up MongoDB: Imported shelter data into the database.
2. Developed CRUD operations: Implemented the database queries in Python.
3. Built the Dash dashboard: Created interactive visualizations to display filtered animal data.
4. Tested in Jupyter Notebook: Verified database functionality before deployment.
5. Deployed the application: Ensured the dashboard runs smoothly on a local server.

## **Challenges and Solutions**

* Database Connection Errors: There were some intermittent connection failures. Configuring MongoDB authentication correctly solved this issue.

## **Roadmap / Future Enhancements**

* Machine Learning Integration: Predictive modeling for identifying optimal rescue dog candidates.
* Cloud Deployment: Hosting the dashboard on a cloud platform for wider accessibility.

## **Contact**

Ty Wheeler  
 Email: ty.wheeler@snhu.edu