**7-2 Project Two**

**CS-340-10388-M01 Client/Server Development**

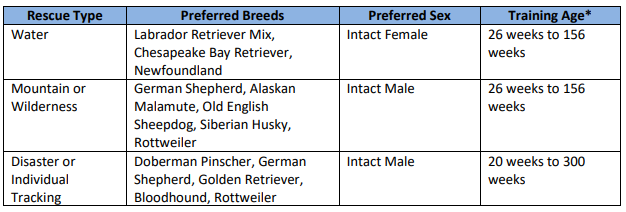
**Nick Tilton**

**October 24th, 2025**

**Required Functionality:**

This project involves utilizing the given animal shelter dataset to develop a comprehensive database system. The process encompasses implementing fundamental Create, Read, Update, and Delete (CRUD) operations to manage the data effectively. Subsequently, a user friendly dashboard is constructed to visualize and interact with the dataset. The dashboard is designed to present each animal's information in a structured table format, allowing users to apply predefined filters to locate specific animals, such as dogs that meet certain criteria. Furthermore, the dashboard features graphical representations of the filtered results, providing visual insights, and includes a map component that displays the geographic location of a selected animal. To improve usability and performance, the dataset view is limited to 10 entries per page, with pagination controls enabling smooth navigation through the remaining data. This project not only demonstrates data management skills but also emphasizes effective data visualization and user interface design principles. The following screenshots illustrate the various functionalities and features of the program, providing a visual overview of its capabilities and performance.

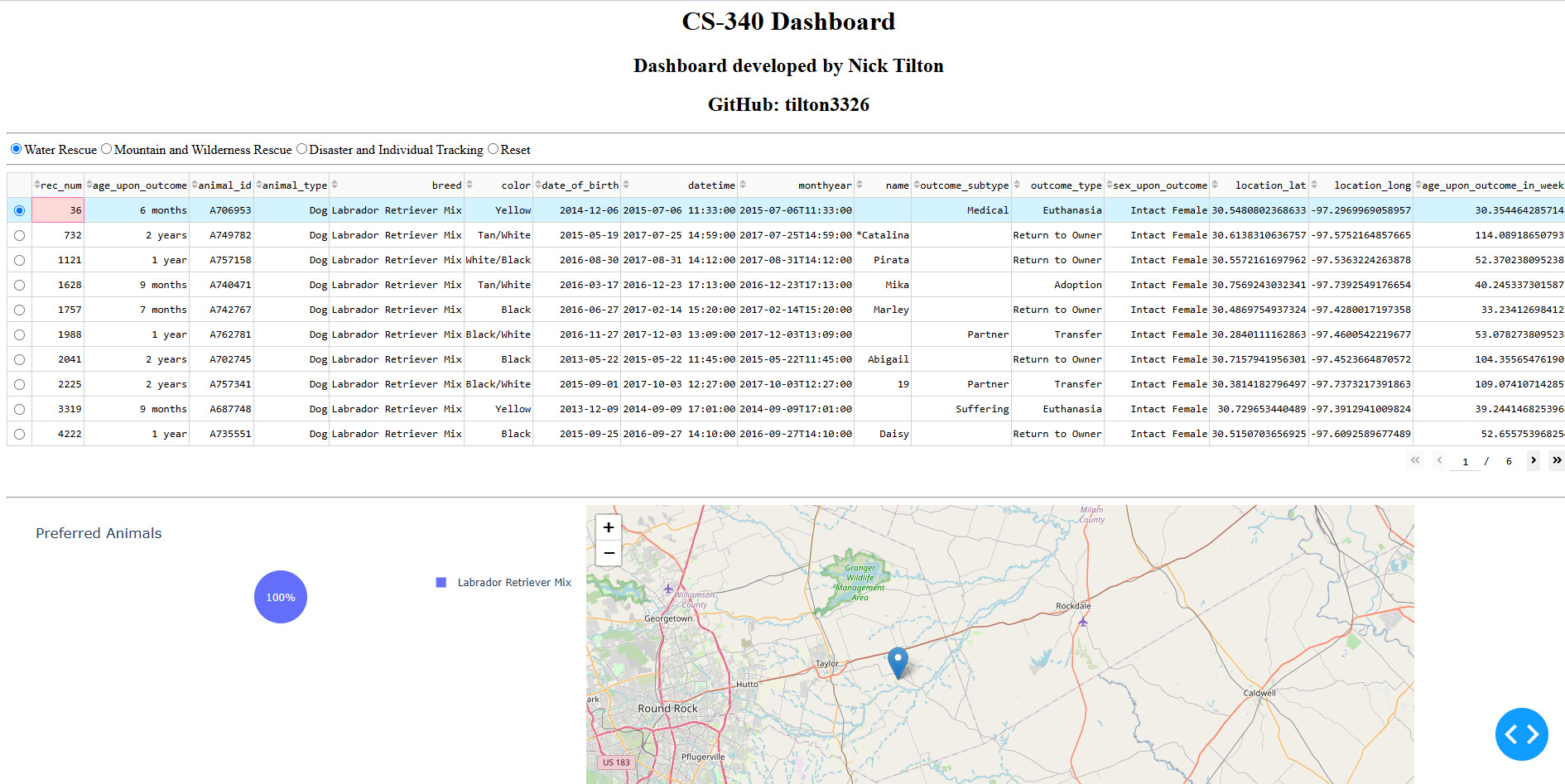
**Rubric Criteria for Filters:**



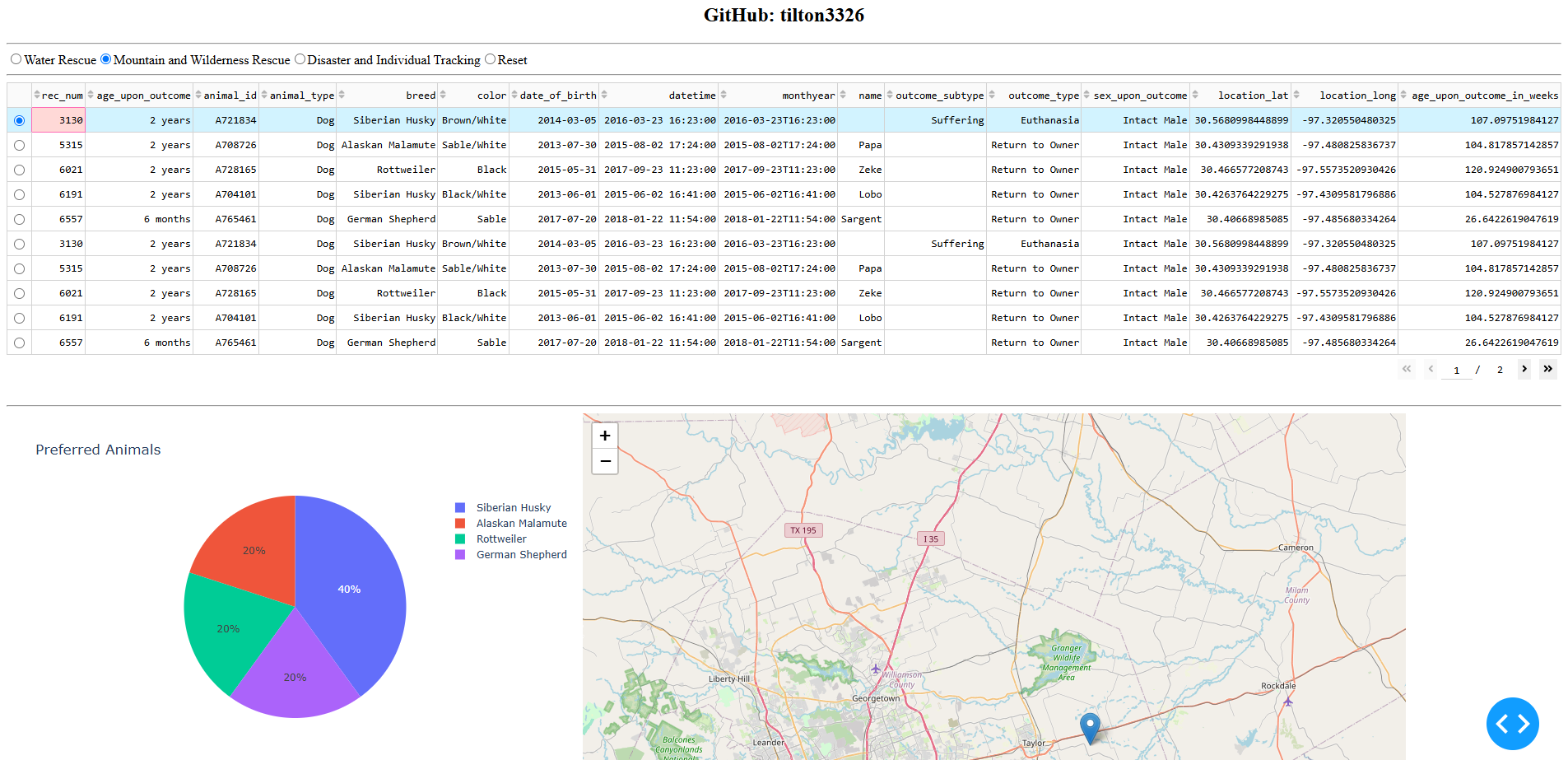
**Default landing page displaying query of all dogs (Reset Page):**



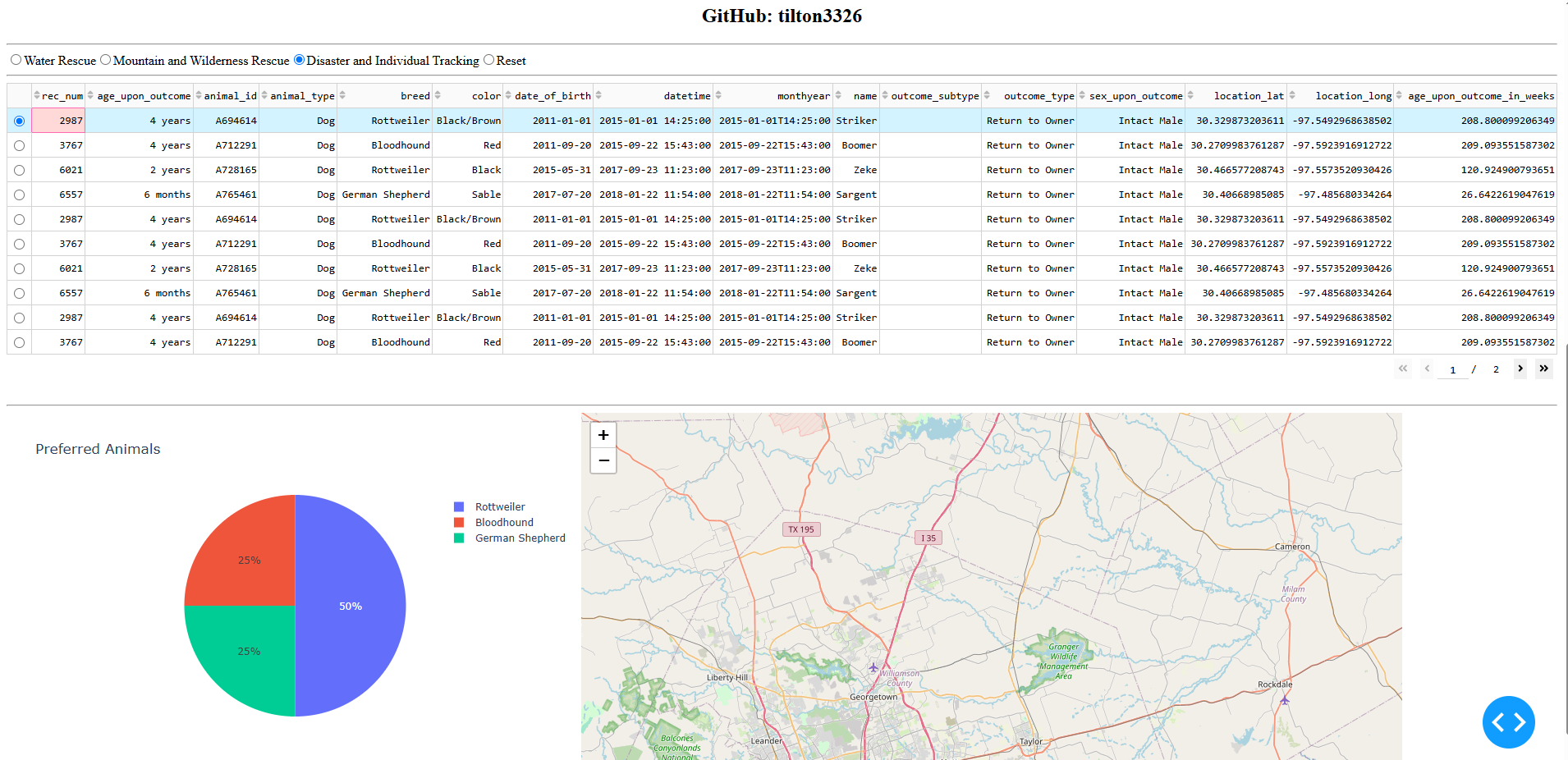
**Query of dogs via Water Rescue:**



**Query of dogs via Mountain and or Wilderness Rescue:**



**Query of dogs via Individual Tracking and or Disaster Rescue:**



**Tools Used:**

1. **MongoDB** – Chosen as the primary database due to its user friendly interface with Python and its flexible, document oriented architecture. This storage system was well suited to the structure of the dataset, and its scalability and efficient querying capabilities made it an excellent choice for managing the animal's data effectively.
2. **Python** – Served as the main programming language for developing the application's core functionalities. Python was used to implement CRUD (Create, Read, Update, Delete) operations, enabling robust data management. Additionally, it facilitated the development of the dashboard, integration of data visualizations, and overall application logic.
3. **Pandas** – Employed for data manipulation and processing after retrieving information from the database. Pandas provided powerful tools for filtering, analyzing, and preparing data, which supported dynamic display and visualization within the dashboard.
4. **Dash** – Dash allows for rapid development of visual components and user controls without requiring extensive front end programming knowledge. Its seamless integration with Python enables dynamic data display and interaction.
5. **PyPlot** – Used to generate interactive, visually appealing graphs that effectively illustrate data trends and insights. PyPlot’s capabilities enhanced data visualization, making complex information easier to interpret and analyze within the dashboard.

**Creates instructions for reproducing the project by explaining steps that were taken to complete the project:**

**Getting Started**

1. To begin using this program, start by opening MongoDB and importing the CSV file named "aac\_shelter\_outcome.csv" into the database. This step ensures that the data is properly loaded and ready for manipulation within the system.
2. Next, create both a simple index and a more complex index within the database. These indexes will help efficiently parse and organize the data stored within the documents, significantly improving data retrieval speed and overall query performance.
3. After setting up the indexes, create user accounts to secure access to the database. You should establish an Admin account with full privileges and a standard user account, named "aacuser," to enable controlled access and management of the data.
4. Once user accounts are created, ensure that Python is installed on your system. If it is not already installed, download and install the latest version of Python from an official source.
5. Finally, run the program from a Python notebook or an integrated development environment (IDE). This allows you to interact with the database, perform CRUD operations, and utilize the application's functionalities effectively.

**Installation**

1. To run and use both the Python (.py) and Jupyter Notebook (.ipynb) files for this project, you first need to install a current version of Python on your computer. Using the latest version ensures compatibility and smooth operation of all features and libraries.
2. Next, download the CSV data file and upload it into your project folder, as the program relies on this data for processing and analysis.
3. Additionally, you must install MongoDB by following the most recent official documentation. Proper installation of MongoDB is essential for accessing and managing the database effectively.
4. Completing these steps updating Python, uploading the CSV file, and installing MongoDB will prepare you to run the project files successfully and work efficiently within the dataset.

**Challenges Faced & How they were Overcome:**

The primary obstacles encountered during this project were related to minor mistakes within the Python scripts and some difficulties when working with MongoDB. Since I was unfamiliar with several practices and techniques involved, I experienced a learning curve at the beginning of the process. Navigating these new concepts requires additional effort and understanding. However, by actively engaging with course materials and conducting independent research online, I was able to identify and apply best practices. This approach allowed me to overcome initial challenges and successfully complete the assignment, gaining valuable experience in both Python programming and database management.