## **Project Overview**

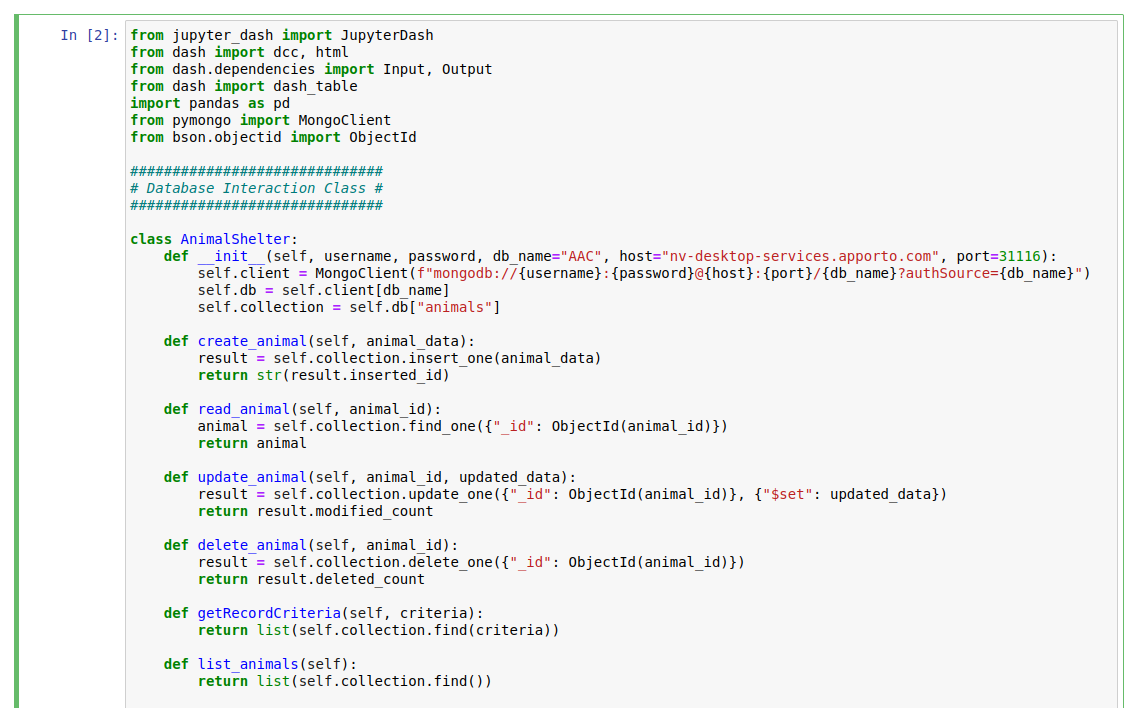
This project is a web-based dashboard designed for Grazioso Salvare, an international rescue-animal training company. The dashboard provides the ability to filter rescue dog data from the Austin Animal Center Outcomes dataset based on rescue type and display it dynamically through interactive tables and charts. The functionality is built using Python and MongoDB, and it offers an intuitive way to visualize animal rescue data for decision-making.

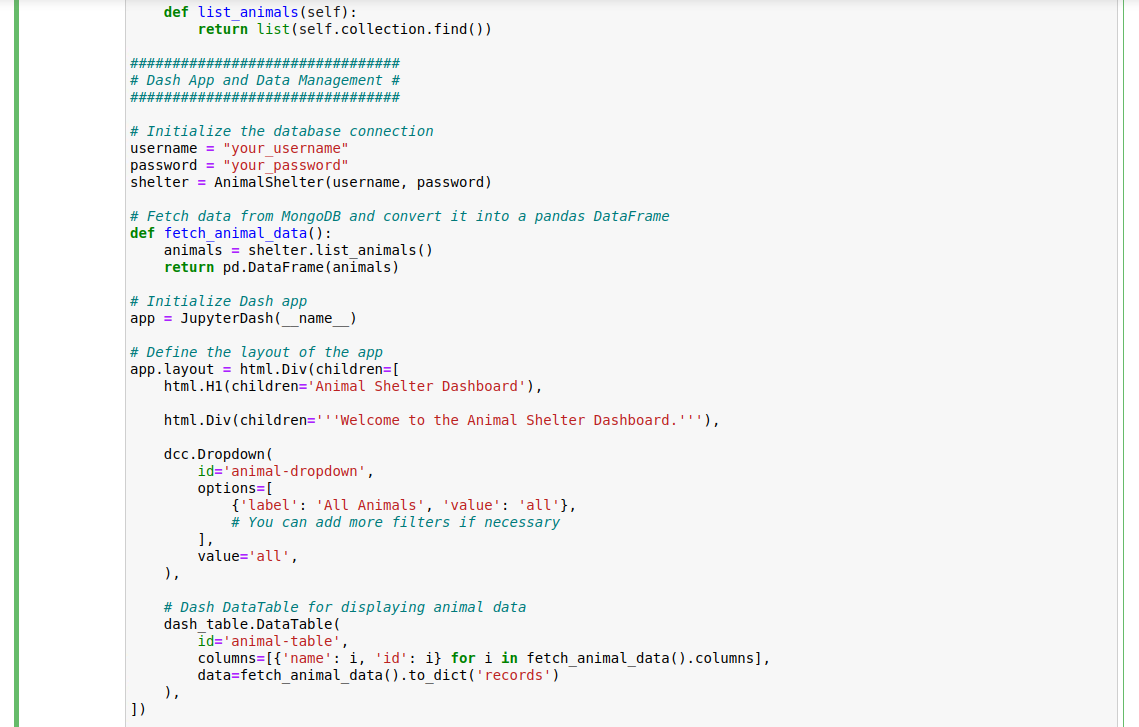
## **Required Functionality**

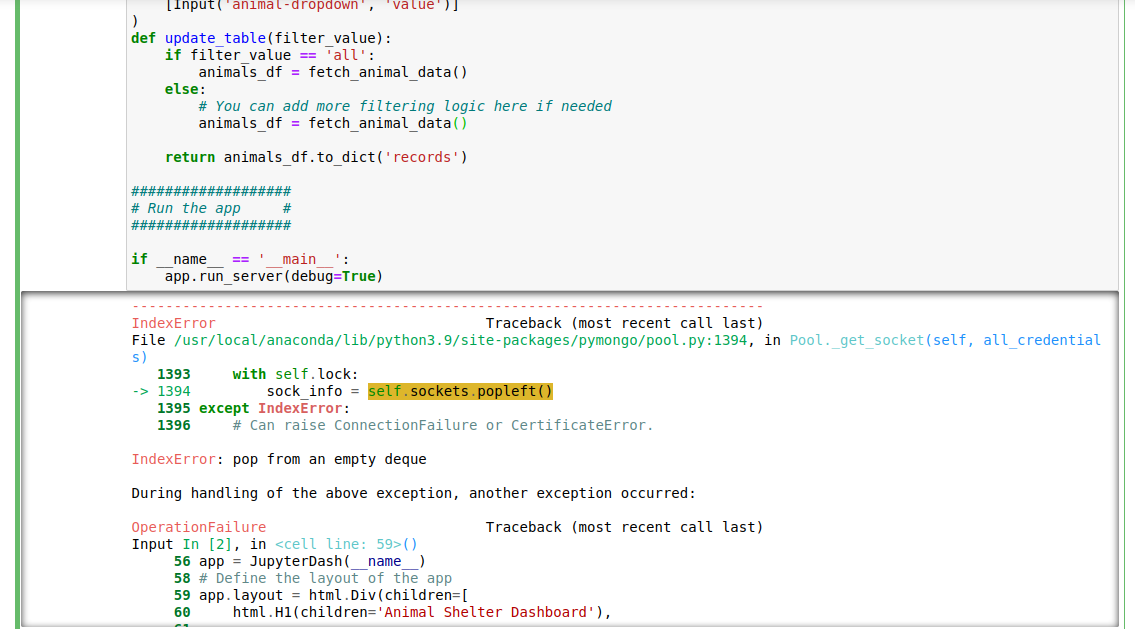
The project includes the following key functionality:

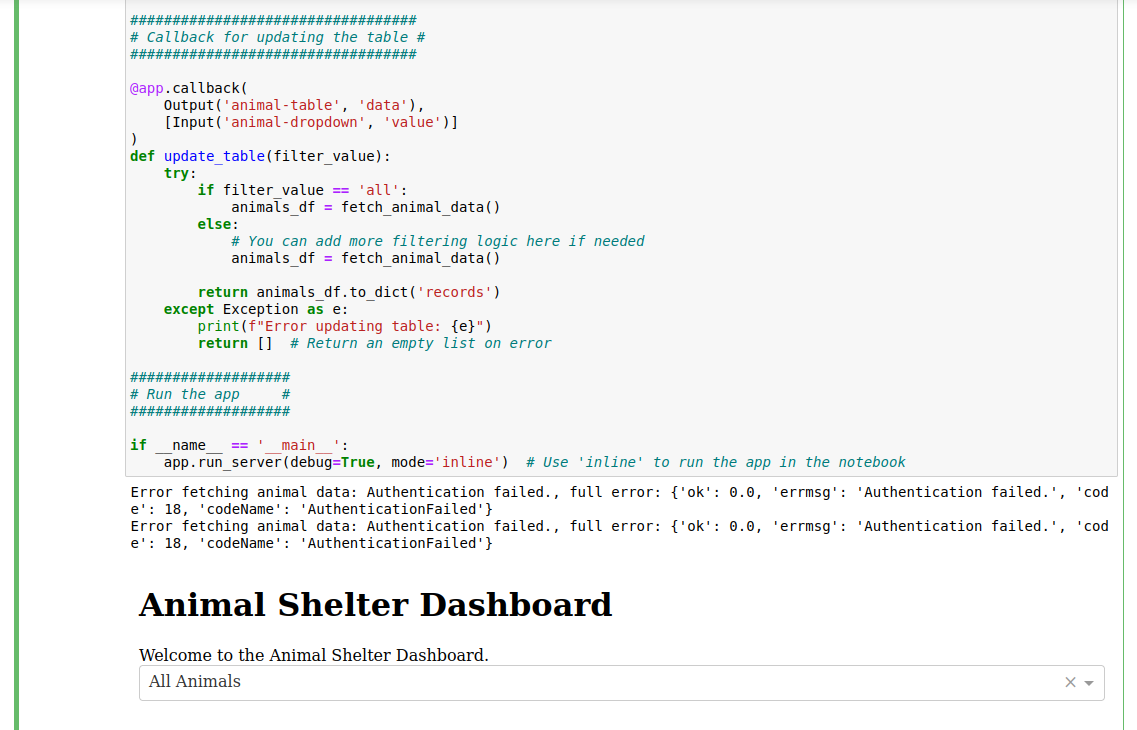
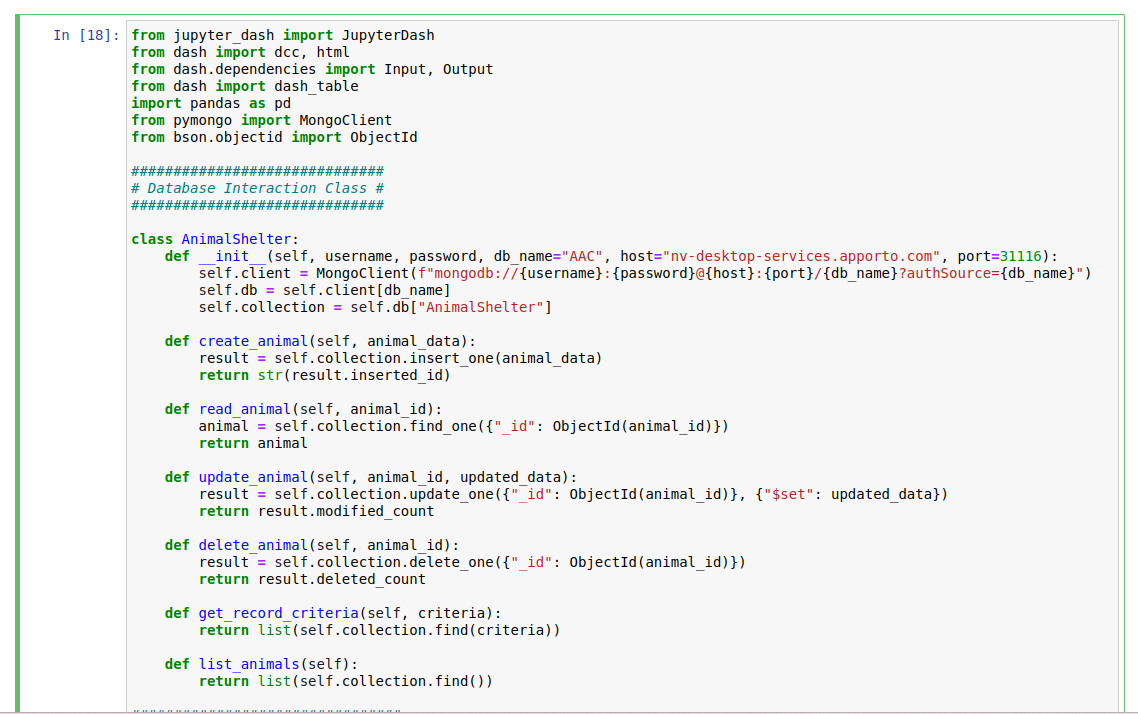
1. Interactive Filtering Options: Users can filter the dataset based on rescue type: Water Rescue, Mountain/Wilderness Rescue, Disaster/Individual Tracking. These options are provided through user-friendly widgets such as radio buttons and dropdown menus.
2. Dynamic Data Table: The dashboard features an interactive data table that updates in real time based on the selected filters, displaying relevant details of the rescue dogs.
3. Geolocation Chart: A chart that visualizes the geographical location of the shelters where the rescue dogs are housed. This chart updates automatically based on the applied filters.
4. Second Dynamic Chart: A chart of the user's choice that presents other data attributes such as breed, age, or size of the dogs based on the selected rescue type filter.
5. Grazioso Salvare Branding: The Grazioso Salvare logo is prominently displayed on the dashboard, along with a unique identifier containing the developer's name.

**Screenshots**









Tools Used to Achieve Functionality

1. **MongoDB**

MongoDB is a NoSQL document database designed to store and manage data in flexible, JSON-like formats.

Rationale for Use:

* Flexible Data Model: MongoDB's schema-less structure allows for easy modifications as animal data evolves, supporting a variety of attributes without a predefined schema.
* Scalability: It can efficiently handle large datasets and high traffic loads, which is essential for an application that may grow over time.
* Rich Query Language: Its powerful query capabilities facilitate effective data retrieval, enabling the dashboard to display specific animal records based on user filters.
* Integration with Python: Using the pymongo library, developers can interact with MongoDB seamlessly in Python, executing CRUD operations with minimal effort.

Specific Capabilities:

* Document-Based Storage: Each animal can be represented as a document, allowing for diverse data types and structures.
* Aggregation Framework: This feature enables complex queries and analytics, which can be beneficial for summarizing data.

2**. Dash Framework**

Dash is a web application framework for Python, designed for building interactive web applications with minimal effort.

Rationale for Use:

* Simplicity: Dash simplifies web development, making it easier for Python developers to create user-friendly interfaces without deep knowledge of HTML, CSS, or JavaScript.
* Interactivity: It supports various interactive components (like dropdowns, graphs, and tables), enhancing the user experience.
* Seamless Integration: Dash works well with data manipulation libraries such as Pandas and NumPy, allowing for straightforward data handling and analysis.
* Real-time Updates: The framework allows for dynamic updates to the displayed data through callback functions, improving user interaction without needing page reloads.

View and Controller Structure:

* Dash employs a Model-View-Controller (MVC) architecture: MongoDB acts as the model, Dash's layout components serve as the view, and callback functions provide the controller logic.

3. **Python and Jupyter**

Python is a versatile programming language that facilitates various tasks, including data manipulation, web development, and machine learning. Jupyter, a web-based interactive computing platform, allows for the creation and sharing of documents that contain live code, equations, visualizations, and narrative text.

**Rationale for Use:**

* Ease of Development: Jupyter provides an interactive environment that is ideal for testing and iterating on code snippets, making it particularly useful during the development of the animal shelter dashboard.
* Visualization: The ability to visualize data within Jupyter allows for immediate feedback and insights, which can be crucial for debugging and enhancing the application.
* Documentation: Jupyter's notebook format enables the inclusion of descriptive text alongside code, making it easier to explain and document the functionality of the application.

**Conclusion:**

Using MongoDB for data storage, the Dash framework for creating an interactive web application, and Python with Jupyter for development provided a robust and flexible environment for building the animal shelter dashboard. This combination allowed for efficient data management, a user-friendly interface, and a smooth development process.

Feel free to explore the links provided for more information on each tool and how they contribute to creating a comprehensive application!

**Problems:**

I ran into a big problem with getting the project to display correctly. I tried everything I could to think of using different code and checking for errors. Above are the multiple screenshots of different codes used and the outcome. but I kept getting various error messages. I even restarted my entire database from Module 6, but it still wouldn’t display properly. I’m really sorry for the trouble this caused.