# CS 340 README for Grazioso Salvare’s Custom Dashboard

**Project Overview**

The Grazioso Salvare dashboard project aims to provide a comprehensive interface for managing animal shelter data. It includes features for viewing, filtering, and editing animal records stored in a MongoDB database. Below are the functionalities implemented:

* Dashboard Interface: Interactive dashboard displaying animal shelter data, including a data table, pie chart visualization, and geolocation mapping.
* Data Filtering: Radio buttons to filter data based on rescue types: Water Rescue, Mountain or Wilderness Rescue, and Disaster Rescue.
* CRUD Operations: Implemented Create, Read, Update, and Delete operations for managing animal records.

**Tools Used**

The following tools were utilized to achieve the functionality of the Grazioso Salvare dashboard:

* **Dash**: A Python framework for building web applications, providing the view and controller structure for the web application.
* **Jupyter** **Dash**: An extension that enables running Dash applications within Jupyter notebooks.
* **Dash** **Leaflet**: A Dash component library for creating interactive maps.
* **Plotly** **Express**: Used for generating pie chart visualizations.
* **Pandas**: Utilized for data manipulation and handling.
* **Matplotlib**: Used for additional data visualization, though not directly involved in the dashboard interface.

**MongoDB Usage**

MongoDB was selected as the model component of the development due to its specific qualities and capabilities for interfacing with Python:

* **Document-Oriented**: MongoDB's document-based storage model aligns well with Python's dictionary data structure, making it intuitive for data manipulation.
* **Flexible** **Schema**: Allows for dynamic schema changes, facilitating easy updates and modifications to the data model.
* **Rich** **Query** **Language**: MongoDB's query language provides powerful features for filtering and querying data, essential for the dashboard's functionality.

**Dash Framework**

Dash framework was chosen for building the Grazioso Salvare dashboard due to its simplicity and flexibility:

* **Pythonic** **Syntax**: Dash enables building web applications using pure Python, making development straightforward and accessible.
* **Component-Based Structure**: Dash follows a component-based architecture, allowing for modular development and easy customization of individual components.
* **Integration with Plotly**: Dash seamlessly integrates with Plotly to create interactive visualizations and enhance the dashboard's data presentation capabilities.

**Project Completion Steps**

The project was completed following these steps:

1. **Data** **Retrieval**: Animal data was retrieved from MongoDB using the provided CRUD Python module.
2. **Dashboard** **Setup**: Jupyter Dash was utilized to set up the Dash application within a Jupyter notebook.
3. **Data** **Visualization**: Plotly Express and Dash Leaflet were employed for creating interactive charts and maps.
4. **Interaction** **Handling**: Callback functions were implemented to handle user interactions, such as filtering data and updating visualizations.
5. **Testing** **and** **Deployment**: The dashboard was tested for functionality and deployed within the Jupyter environment.

**Challenges Faced**

For individuals endeavoring to replicate the project, a significant challenge arose in filtering the data from the Austin Animal Shelter. Grazioso Salvare had specific criteria regarding dog breeds, yet the data in the database proved to be less than pristine. For instance, while Grazioso Salvare sought information on the Chesapeake Bay Retriever breed, it was recorded as "Chesa Bay Retr" in the raw dataset. Consequently, implementing pattern matching became necessary to generate an accurate list of animals. To achieve this, Regular Expressions were utilized for pattern matching. However, it's noteworthy that Pymongo doesn't inherently support regular expressions and thus necessitated a different formatting approach to be compatible with the Pymongo API.

Top of Form

## Contact

Your name: Brad Follett