**Project Two: Austin Animal Center Outcomes Dashboard**

This project creates a dashboard that visualizes data from the Austin Animal Center Outcomes dataset. The dashboard includes interactive widgets to filter the data, a dynamic data table, and charts that respond to updates from the data table. The project was completed using Python, MongoDB, and the Dash framework.

**Functionality**

The dashboard includes the following features:

* Interactive options to filter the Austin Animal Center Outcomes data set
* A data table which dynamically responds to the filtering options
* A geolocation chart and a second chart of your choice that dynamically respond to the filtering options

The dashboard also includes the Grazioso Salvare logo and a unique identifier containing your name.

The following screenshots demonstrate the functionality of the dashboard:

Graphical user interface

Description automatically generated with medium confidenceGraphical user interface, chart

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, chart, application

Description automatically generatedTable

Description automatically generated with medium confidenceGraphical user interface, chart, application, pie chart

Description automatically generatedTable

Description automatically generatedGraphical user interface, chart

Description automatically generated

**Tools Used**

The following tools were used to achieve the functionality of the dashboard:

* Python: programming language used to develop the dashboard
* MongoDB: database used to store the Austin Animal Center Outcomes data
* Dash: framework used to create the dashboard

Python was chosen as the programming language for this project because of its versatility and popularity in the data science community. MongoDB was chosen as the database because it provides a flexible and scalable solution for storing and querying data. Dash was chosen as the framework because it provides a simple and efficient way to create interactive web applications with Python.

**MongoDB**

MongoDB was used as the model component of the development because it provides several specific qualities and capabilities that are well-suited to interfacing with Python:

* BSON format: MongoDB stores data in a binary JSON format called BSON, which can be easily parsed and manipulated with Python.
* Document-oriented: MongoDB stores data as documents, which makes it easy to work with data that has a flexible schema, such as the Austin Animal Center Outcomes data set.
* Query language: MongoDB provides a rich query language that allows for complex queries and aggregations, which can be constructed using Python.

**Dash Framework**

Dash is a framework for building web applications with Python. It provides a simple and efficient way to create interactive web applications by using reactive programming techniques. Dash allows developers to create web applications using familiar Python libraries and syntax, which makes it easy to get started with web development in Python.

**Resources**

The following resources and software applications were used in the development of the project:

* Austin Animal Center Outcomes dataset
* Dash documentation
* MongoDB documentation

**Steps Taken**

To complete the dashboard project, the following steps were taken:

1. Reviewed the Dashboard Specifications Document provided by the UI/UX developer at Global Rain to understand the required functionality for the dashboard.
2. Created a data table on the dashboard to display an unfiltered view of the Austin Animal Center Outcomes data set, utilizing the previous CRUD Python module from Project One to retrieve all data from MongoDB.
3. Developed database queries that matched the required filter functionality based on the Rescue Type and Preferred Dog Breeds Table provided in the Dashboard Specifications Document.
4. Developed controller pieces to create interactive options that allowed for the selection of data based on filtering functions, such as radio items or dropdowns, using the Dash Core Components.
5. Modified the data table to be an interactive data table that responded to input from the interactive options, as well as created charts that displayed data in response to updates from the data table, including a geolocation chart and a second chart of our choice.
6. Tested and deployed the dashboard to ensure that all components worked properly, including taking screenshots or creating a screencast of the dashboard and widget functionality.

**Challenges Encountered and Overcame**

One of the main challenges encountered during the project was understanding how to properly use the Dash framework to create interactive web applications. To overcome this challenge, we spent time reviewing the Dash documentation and examples, as well as working through various tutorials and exercises to gain a better understanding of the framework. Additionally, we encountered some issues with properly querying the MongoDB database using our CRUD Python module but were able to resolve these issues by carefully reviewing the module code and documentation, as well as testing various queries and ensuring that they returned the expected results.