# CS 340 PROJECT TWO README

## Functionality

Grazioso Salvare’s dashboard provides users with access to the unfiltered data from the database, as well as filter options based on rescue types of dogs. The categories of rescue available as filters are:

Water Rescue

Mountain or Wilderness Rescue

Disaster or Individual Tracking

Moreover, graphs make it easier to visualize the data being displayed. A pie chart shows the breeds, while a map shows the animals’ geolocation, and a histogram shows their ages. Examples of usages of these functionalities can be seen in the attached PDF files titled *1, 2, 3,* and *4*.

**Tools**

The following tools were used to create this project:

[Python](https://www.python.org/downloads/): An interpreted programming language that is dynamically typed and uses significant indentation.

[Dash](https://dash.plotly.com/installation): A Python framework that allows developers to easily create interactive web-based applications based on a layout and callback structure.

[Jupyter Notebook](https://jupyter.org/install): An interactive development environment that makes development in Python and Dash fast and efficient.

[MongoDB](https://docs.mongodb.com/manual/installation/): A NoSQL cross-platform database that stores data in documents and collections. The document data is stored in JSON-like format, which is analogous to dictionaries in Python. This allows for flexibility and makes working with the database more intuitive.

[PyMongo](https://docs.python.org/3/tutorial/errors.html): A middleware that provides a way for Python applications to communicate with MongoDB databases.

**Steps**

The first step to complete this project was creating the back end of the application. The shelter’s data was imported to a new MongoDB collection and credentials were created for users to access it. Then, a Python class was created to authenticate the user credentials and connect to the database and perform CRUD operations using PyMongo. The next step was to create the front end of the application using Dash. First, the layout was created and then functionality was implemented. Radio buttons were created for different rescue types of animals and a callback was implemented to filter the database documents based on the selected option. Charts and a map were also used to show the information captured in an interactive way.

**Challenges**

I faced two challenges when working on this project. First, when I would run the query to get all documents, I would get an IOPub data rate exceeded error. To solve this issue, I executed Jupyter Notebook from the terminal window using an option to increase the data rate limit. The second issue was related to the JSON serialization of the ObjectId field values imported from the database. Since the ObjectId value was not relevant to the project, I solved this issue by not importing that specific column to the application’s dataframe.

## Contact

Diego Bez Zambiazzi