# Grazioso Salvare Animal Shelter Dashboard README

## About the Animal Shelter Dashboard

This project is a web-based dashboard for an animal shelter named Grazioso Salvare. The dashboard provides functionality for animal rescue teams to view the animals available for rescue, and to filter the animals based on their rescue needs. The dashboard also displays the location of each animal on a map and provides other important information, such as breed, age, sex, and training information.

**Functionality**

The required functionality of the project includes the ability to filter animals based on rescue type, view a map of animal locations, and view additional information about each animal.

Tools used to achieve this functionality include:

* Python: used as the primary programming language for the project
* Dash: used as the web framework for creating the dashboard
* Plotly: used for data visualization
* Pandas: used for data manipulation
* MongoDB: used as the database management system

## Getting Started

To get started with the Animal Shelter CRUD Python module, clone the project repository to your local machine and follow the installation instructions to set up the required tools and libraries.

## Installation

1. Install Python 3.x if not already installed: [Download Python](https://www.python.org/downloads/)
2. Install MongoDB: [Download MongoDB](https://www.mongodb.com/try/download/community)
3. Install pymongo library: **pip install pymongo**

**Database and User Authentication**

MongoDB database was set up for the animal shelter, and user authentication was configured. Make sure to set up a user with appropriate permissions for your MongoDB instance.

**MongoDB**

MongoDB was chosen as the model component of the development because it provides a flexible and scalable document-based data model that allows for easy storage and retrieval of data. Additionally, MongoDB has a robust query language and provides support for complex queries and aggregations.

**Dash**

The Dash framework was used to provide the view and controller structure for the web application. Dash is a powerful tool for building data-driven web applications in Python, and it provides a simple syntax for creating interactive dashboards with responsive, modern user interfaces.

**Resources and software applications**

Resources and software applications that were used in this project include:

* Jupyter Notebook: used for developing and testing the code
* PyCharm: used as the IDE for developing the code
* MongoDB Atlas: used as the cloud-based database management system
* Heroku: used for deployment of the web application

**Steps taken to complete the project**:

1. Create a CRUD module for interacting with the MongoDB database.
2. Import data into the MongoDB database using the CRUD module.
3. Develop data filtering functions for rescue teams to use to filter animals based on their rescue needs.
4. Create interactive options using Dash Core Components to allow for selection of data based on filtering functions.
5. Create interactive data tables and charts that respond to input from the interactive options.
6. Test and deploy the dashboard to Heroku.

**Challenges**

Challenges encountered during the project include authentication issues with the MongoDB database. Credentials were verified and a direct connection was made through the terminal, but the application refused to establish connection. This challenge was not resolved and prevented the ability to view the output of the code for the dashboard to verify its functionality.

Graphical user interface, text, application

Description automatically generated

## Contact

Dave Barnes