# CS 340 README

**Client: Grazio Salvare (International Rescue-Animal Training Company)**

**Development Company: Global Rain | Lead Developer: Deonne Ludwig**

## About the Project/Project Title

Grazioso Salvare is seeking a software application that can work with existing data from animal shelters to identify and categorize available dogs. Global Rain has contracted for a full stack development of this application that will include a database and a client-facing web application dashboard. Users at Grazioso Salvare will access the database. Grazioso Salvare has also requested that the code for this project be open source and accessible on GitHub so similar organizations may use and adapt it. They have asked that you also create a README file to accompany your work.

## Motivation

Grazioso Salvare identifies dogs that are good candidates for search-and-rescue training. When trained, these dogs can find and help to rescue humans or other animals, often in life-threatening conditions. To help identify dogs for training, Grazioso Salvare has reached an agreement with a nonprofit agency that operates five animal shelters in the region around Austin, Texas. This nonprofit agency will provide Grazioso Salvare with data from their shelters.

Grazioso Salvare, looks for certain profiles in dogs to train. For instance, search-and-rescue training is generally more effective for dogs no more than two years old. Also, certain breeds of dogs are proficient at different types of rescue, such as water rescue, mountain or wilderness rescue, locating humans after a disaster, or finding a specific human by tracking their scent. This software will give them a more efficient way to identify and organize available dogs.

## Getting Started

When the interface is created the Grazioso Salvare user will be able to interact and visualize the database information through a fully functional MongoDB dashboard. The model is contained and accessed in MongoDB, the views are dashboard widgets, and the controller uses a CRUD Python module for queries as part of the interaction between components. The user will be able to filter the data that they need and create, read, update and delete records for the animals, etc. in the database.

## Installation

* **MongoDB and the mongo shell***– from the Linux shell prompt, execute the mongosh command to start the mongo shell. Mongoimport can be used to load needed databases and collections.*
* ***PyMongo*** *: Driver used to communicate between MongoDB and Python. It provides the ability to query,*

*retrieve results, write and delete data and run database commands.*

*“MongoDB stores data in flexible and schema-less JSON-like documents. Python has rich libraries that directly process JSON and BSON data formats. Python integrates well with MongoDB through drivers like PyMongo, MongoEngine etc. This makes MongoDB good for Python by eliminating rigidity in the database schema. “*

*How to Use Python with MongoDB-*[*https://www.mongodb.com/languages/python#:~:text=MongoDB%20stores%20data%20in%20flexible,rigidity%20in%20the%20database%20schema*](https://www.mongodb.com/languages/python#:~:text=MongoDB%20stores%20data%20in%20flexible,rigidity%20in%20the%20database%20schema)*.*

* **Dash Framework** *- Dash is an open-source Python framework for building web applications, created and maintained by the people at Plotly. Dash's web graphics are completely interactive because the framework is built on top of Ploty. js, a JavaScript library written and maintained by Ploty*

*Sproul, A. (2018, November 15). Dash: A beginner’s guide. Medium.* [*https://towardsdatascience.com/dash-a-beginners-guide-d118bd620b5d#:~:text=Dash%20is%20an%20open%20source,written%20and%20maintained%20by%20Ploty*](https://towardsdatascience.com/dash-a-beginners-guide-d118bd620b5d#:~:text=Dash%20is%20an%20open%20source,written%20and%20maintained%20by%20Ploty)*.*

**CRUD, insertOne(), Find(), FindOne() and MongoDB queries (.distinct, .updateOne, updateMany)** *– use operations of Create, Read, Update, and Delete (CRUD) to create databases of document collections, perform reading or querying operations to retrieve documents and update and remove documents and subdocuments.*

* **Single and compound indexes** *- to optimize the queries and also use the*explain*function to verify the use of indexes in the queries.*
* **User accounts** - *in Mongo to address security needs.*

## Usage

This program will provide indexing for optimizing queries, CRUD (Create, Read, Update, and Delete) functionality for the animal database, and authentication for database security.

* ***MongoDB Import Execution*** *– using the Linux terminal the user can use the mongoimport tool to load databases to the chosen directory and verify it is successful via a query in the mongo shell.*

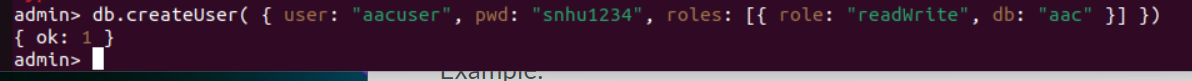
A screenshot of a computer program

Description automatically generated

A computer screen with text

Description automatically generated

* ***Authentication Execution*** *– to comply with the client’s security needs users are able to ensure user authentication to the database via the mongo shell. This can be verified by running mongosh as the user.*



A computer screen shot of a program

Description automatically generated

A screenshot of a computer program

Description automatically generated

* ***CRUD Functions Execution***
* ***CRUD*** *: (Create, Read, Update, and Delete) allows users to interact with databases to create data, access data via the UI by reading the data, update or edit the data, and delete the data.*
* ***PyMongo*** *: Driver used to communicate between MongoDB and Python. It provides the ability to query,*

*retrieve results, write and delete data and run database commands.*

A screenshot of a computer program

Description automatically generated

* ***CRUD Functionality Test Execution*** *– Jupyter Notebooks can be used by the user to create a Python testing script to ensure successful CRUB functionality.*

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

*Final testing script for application functionality- Jupyter Notebooks*

*A screenshot of a computer program

Description automatically generated*

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

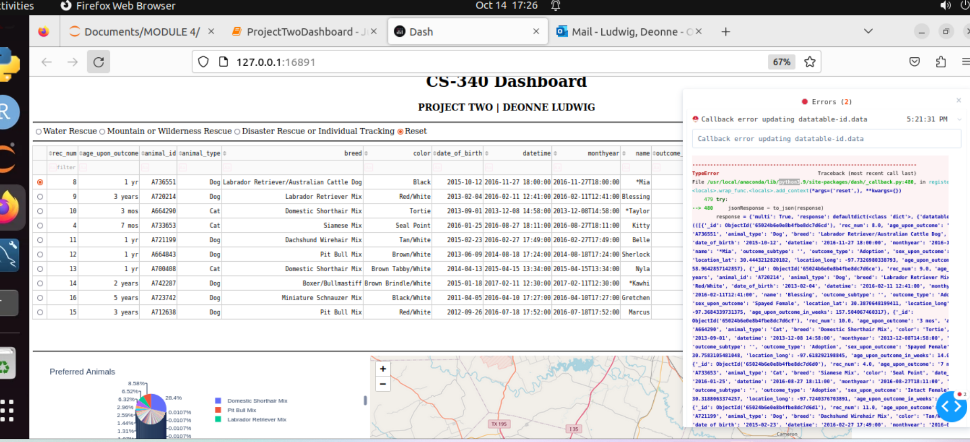
Description automatically generated

A screenshot of a computer program

Description automatically generated

Challenge faced:

Call back error for database-id/update\_dashboard



A screen shot of a computer code

Description automatically generatedA screen shot of a computer code

Description automatically generated

**FUNCTIONALITY DEMONSTRATION:**

**Starting state (Defaulted to show “Reset” option also) of dashboard -**

A screenshot of a computer

Description automatically generated

**Water Rescue filter selected on dashboard -**

A screenshot of a computer

Description automatically generated

**Mountain or Wilderness Rescue filter selected on dashboard -**

A screenshot of a computer

Description automatically generated

**Disaster Rescue or Individual Tracking filter selected on dashboard –**

A screenshot of a computer

Description automatically generated

## Contact

Deonne Ludwig – Global Rain