# CS 340 README

## About the Project/Project Title

The Grazioso Salvare Project is a program to help identify and categorize data from local animal shelters that are good candidates for search and rescue training.

## Motivation

The data is already up for grabs, so by taking this project to fruition we are getting these service animals identified more quickly, which is important as it has been stated that under two years old is the most effective age, and saving lives sooner.

## Getting Started

The program is as simple as accessing the database and then importing the data. Once the data is loaded, log in and then simple querying commands present the user with the exact information that’s needed.

## Installation

For this program you’ll need MongoDB and Python installed locally, both which are lightweight.

## Usage

### Code Example

# Complete this create method to implement the C in CRUD.

def create(self, data):

if data is not None:

self.database.animals.insert\_one(data) # data should be dictionary

if isinstance(data, dict): # check if variable is a dictionary

print('Created Succesfully')

return True

else: # else it is false

print('Invalid input, please try again')

return False

else:

raise Exception("Nothing to save, because data parameter is empty")

### Tests

from AnimalShelter import AnimalShelter

from bson.objectid import ObjectId

animal = AnimalShelter()

penguine = { "animal\_id": "A333302", "animal\_type" : "Penguine", "breed" : "Emperor", "color" : "black and white"}

animal.create(penguine)

animal.read(penguine)

### Screenshots

*A screenshot of a computer screen

Description automatically generated*

The above screenshot shows the creation of the .py file to C and R

**

The above screenshot shows the test and success of creating an object

A screenshot of a computer

AI-generated content may be incorrect.

Above is a screenshot showing the update happening to the document with a before and after.

A screenshot of a computer code

AI-generated content may be incorrect.

After deleting the document, it can no longer be read.

This is the starting state of the project 2 dashboard:

A screenshot of a computer

AI-generated content may be incorrect.

This is the Water Rescue filter on the project 2 dashboard:

A screenshot of a computer

AI-generated content may be incorrect.

This is the Mountain or Wilderness Rescue filter on the project 2 dashboard:

A screenshot of a computer

AI-generated content may be incorrect.

This is the Disaster or Individual Tracking on the project 2 dashboard:

A screenshot of a computer

AI-generated content may be incorrect.

This is the Reset filter on the project 2 dashboard:

A screenshot of a computer

AI-generated content may be incorrect.

About the Tools used for this project:

We started by using MongoDB as the database for this project. It’s compatible with json files and python drivers make it the optimal choice to import and export the data in a user friendly manner. The Dash framework we used for visualization was chosen as it has well established drivers that use Python as the go between to fetch and then display the desired data. Each of these are modern solutions that have active support and stand the test of time.

See:

[Dash](https://dash.plotly.com/)- The visualization software

[MongoDB](https://www.mongodb.com/docs/manual/)- The database software

[PyMongo](https://pymongo.readthedocs.io/en/stable/tutorial.html)- The python go between driver

To get this project up and running we had to create a database to house and access of the data from cooperating animal shelters. We set up a non system admin user account to be able to connect our python drivers to, to be able to read from the database. After importing the data, we set up a dashboard to display the data, and added filtering to the specifications requested by the client. The result is a dashboard that displays current data and can easily receive new data to update.

The main challenge was getting to know the visualization software. Each software has it’s own idiosyncrasies and I wasn’t too familiar with HTML style elements so it was learning a few things at once.

## Contact

Your name: Michael Abell