# CS 340 README Template

*Use this template to complete your README file. When completing the template, keep the headings as they are so that your document has a clear organization. Remove the italicized prompt text after you have completed each section for a polished final document.*

## About the Project/Project Title

*Grazioso Salvare Animal Rescue Dashboard*

*The project aims to provide a comprehensive dashboard for Grazioso Salvare, featuring a Python module for performing CRUD operations (Create, Read, Update, Delete) on a MongoDB database. The module facilitates easy interaction with MongoDB, allowing users to insert documents into a specified collection and query documents based on specific criteria. The dashboard includes interactive data tables, charts, and geolocation maps for visualizing rescue operations.*

## Motivation

*The motivation behind this project is to simplify database operations for developers working with MongoDB. By providing a reusable Python module, developers can streamline the process of interacting with MongoDB databases, reducing the need for manual database management tasks and enhancing productivity. Additionally, the dashboard provides Grazioso Salvare with an efficient way to monitor and analyze animal rescue operations.*

## Getting Started

*Database and User Authentication:* A computer screen shot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

*CRUD Python Module: Download or clone the provided CRUD Python module (crud.py) and place it in your project directory.*

*Installation: Ensure you have the required dependencies installed. The module relies on the pymongo library, which can be installed via pip:*

*pip install pymongo*

## *Usage:* Usage

## Run the Dash application: from the ProjectTwoDashboard.ipynb in Jupiter notebooks

## Access the dashboard through your web browser at the address provided in the console output (usually <http://127.0.0.1:8050/>).

## Use the interactive options to filter data based on different rescue operations(screenshot provided:

## Water Rescue

## A screenshot of a computer Description automatically generated

## Mountain or Wilderness Rescue A screenshot of a computer Description automatically generated

## Disaster or Individual TrackingA screenshot of a computer Description automatically generated

## Reset to show all data

## View the data table, pie chart, and geolocation map, which update based on the selected filter option.A screenshot of a computer Description automatically generated

## Installation

## *The project utilizes the following tools:*

## *Python: The primary programming language used for development.*

## *pymongo: A Python driver for MongoDB, used for interacting with the MongoDB database.*

## *Dash: A Python framework for building analytical web applications.*

## *Plotly: A graphing library used to create interactive charts and graphs.*

## *Dash Leaflet: A Dash component for creating interactive maps.*

## *pandas: A data analysis and manipulation library.* Describe the required functionality of the project.

## *The project required creating an interactive dashboard for Grazioso Salvare to monitor animal rescue operations. The dashboard includes functionality for CRUD operations on MongoDB, filtering data based on rescue types, and visualizing data in a table, pie chart, and geolocation map. The four filter types supported are Water Rescue, Mountain or Wilderness Rescue, Disaster or Individual Tracking, anda Reset to show all data. The dashboard updates dynamically based on the selected filter.*

## Describe the tools used to achieve this functionality and a rationale for why these tools were used.

## *The following tools were used to achieve the required functionality:*

## *Python: The primary programming language for its simplicity and readability.*

## *pymongo: To interact with the MongoDB database and perform CRUD operations.*

## *Dash: For building the interactive web application.*

## *Plotly: To create interactive charts and graphs.*

## *Dash Leaflet: To incorporate interactive maps into the dashboard.*

## *pandas: For data manipulation and analysis.*

## *These tools were chosen for their ease of integration with each other and their powerful capabilities in handling data and creating interactive visualizations.*

## Explain why MongoDB was used as the model component of the development, including what specific qualities or capabilities it provides for interfacing with Python.

## *MongoDB was chosen as the model component because of its flexibility in handling unstructured data and its scalability. Its document-oriented storage allows for complex data structures to be easily represented. The pymongo library provides a straightforward API for interfacing MongoDB with Python, enabling seamless integration of database operations within the application. MongoDB's powerful querying capabilities and support for indexing make it ideal for handling the varied and dynamic data involved in animal rescue operations.*

## Explain the Dash framework that provides the view and controller structure for the web application.

## *Dash is a Python framework for building analytical web applications. It combines Flask for the web framework, Plotly for interactive graphs, and React.js for the front end. Dash abstracts away the complexities of web development, allowing developers to create interactive, data-driven applications with pure Python. It supports a wide range of interactive features such as callbacks, which are used to update the visualizations and data in real-time based on user inputs. Dash's seamless integration with Plotly and its component-based architecture make it an excellent choice for building sophisticated data dashboards.*

## Explain the steps that were taken to complete the project.

## *Set up the environment: Installed necessary libraries and set up MongoDB.*

## *Developed CRUD module: Created a Python module to handle CRUD operations with MongoDB.*

## *Built the Dash application: Developed the web application using Dash, including the layout and callbacks for interactive features.*

## *Integrated visualizations: Added a data table, pie chart, and geolocation map to display the data.*

## *Implemented filtering options: Added radio buttons to filter data based on rescue types and dynamically update the dashboard.*

## *Tested the application: Tested the functionality of the dashboard and fixed any issues encountered.*

## *Documented the project: Created a comprehensive README file and included screenshots of the dashboard in use.*

## Identify any challenges that were encountered and explain how those challenges were overcome.

*I think there were a couple of challenges I faced, one was I was never able to get the accuser to work correctly even after various attempts and verify it was an actual user. The next would be integrating all of these components so that they a.) functioned and b.) flowed well with each other.*

## Contact

Your name: Lee Kitchen