# CS 340 README

## About the Project/Project Two: Grazioso Salvare Animal Rescue Dashboard

This project is a fully interactive web dashboard for Grazioso Salvare, built using Python, Dash, and MongoDB. It allows users to explore, filter, and visualize animal rescue data from the Austin Animal Center. The dashboard was created to help the Grazioso Salvare team make data-driven decisions and quickly access information about animals that meet criteria for different types of rescue missions.

## Motivation

This project was developed for SNHU CS-340 to demonstrate real-world applications of the MVC design pattern, CRUD operations, and interactive dashboard building with Python and Dash. The aim was to provide Grazioso Salvare with a practical, maintainable tool for their operations, while applying classroom concepts hands-on.

## Getting Started

**To run this dashboard in your Apporto or local environment:**

1. Make sure MongoDB is running, with the provided animal dataset loaded.
2. Ensure you have the following files in your project directory:
   * animal\_shelter.py (your CRUD module)
   * The provided dashboard .ipynb or .py file
   * The Grazioso Salvare logo image
3. Install required libraries (if not already installed):
   * Install with: pip install pandas dash dash-leaflet plotly jupyter-dash pymongo
4. Start your Jupyter Notebook.
5. Open the dashboard as directed in your Jupyter server output.

Database and Authentication:

* Database Name: AAC
* Collection Name: animals
* Authentication credentials:
  + Username: aacuser
  + Password: SNHU1234
  + Host: nv-desktop-services.apporto.com
  + Port: 32575

## Installation

* Install with: pip install pandas dash dash-leaflet plotly jupyter-dash pymongo

Tools and libraries

* Python 3.10+ – Primary programming language for the dashboard and backend logic.
* Dash – Web application framework for building interactive dashboards in Python.
* JupyterDash – Extension for running Dash apps directly inside Jupyter Notebooks.
* Dash Leaflet - Dash component library for interactive geolocation maps.
* Plotly Express – High-level interface for interactive charts and visualizations.
* Pandas – Library for data manipulation and analysis.
* PyMongo – Official MongoDB driver for Python; enables communication with the database.
* MongoDB – Document-based NoSQL database; pre-installed and running on the Apporto virtual environment.
* Jupyter Notebook – Interactive development environment for Python, used for development, testing, and demonstration.

**MongoDB**

MongoDB was chosen as the model component for several key reasons:

* Document-Oriented Storage:  
  Each animal record can have a flexible structure, accommodating changes to data fields without major schema redesign.
* Powerful Query Capabilities:  
  Allows filtering and aggregation directly in the database, enabling the dashboard to display only relevant records based on user-selected criteria.
* Seamless Python Integration:  
  The pymongo library makes it straightforward to read and write data between MongoDB and Python data structures.
* Scalability:  
  Handles large datasets and can be scaled for future needs of Grazioso Salvare.
* Real-Time Updates:  
  Changes in the database can be reflected instantly in the dashboard.

**Dash**

Dash was selected to provide both the view (user interface) and the controller (input/output logic) for the dashboard:

* Component-Based UI:  
  Allows assembling complex dashboards from reusable widgets like tables, charts, and maps.
* Reactive Programming Model:  
  Dash callbacks connect UI elements to backend logic, so changes in filters or selections update the data instantly.
* Pure Python:  
  No need to write separate HTML, JavaScript, or CSS unless customization is required.
* Rich Visualization Support:  
  Built on Plotly, making it ideal for interactive data science dashboards.

**External Resources**

* [Dash Documentation](https://dash.plotly.com/)
* [Plotly Express Documentation](https://plotly.com/python/plotly-express/)
* [MongoDB Documentation](https://www.mongodb.com/docs/)
* [PyMongo Documentation](https://pymongo.readthedocs.io/en/stable/)
* [Dash Leaflet Documentation](https://www.dash-leaflet.com/)

**Steps Taken to Complete the Project**

1. Developed the CRUD module:  
   Created animal\_shelter.py using Python and PyMongo for database access and manipulation.
2. Imported and tested the CRUD module:  
   Loaded the animal dataset into MongoDB and verified insert, read, update, and delete operations.
3. Designed the dashboard layout:  
   Created a clear, accessible user interface using Dash components for the header, filter, data table, and charts.
4. Implemented filtering logic:  
   Developed callback functions to filter animal records by rescue type and update all dashboard components.
5. Added visualization widgets:  
   Built the breed distribution chart and interactive map, wiring them to the data table and filter controls.
6. Integrated branding:  
   Embedded the Grazioso Salvare logo and included a unique identifier.
7. Tested the dashboard:  
   Ran the app in JupyterDash, tested all filters, and took screenshots for documentation.
8. Documented the project:  
   Completed this README and included instructions, rationale, and screenshots.

**Challenges Encountered and Solutions**

* Filter Logic Complexity:  
  Rescue type filters required combining multiple fields (breed, age, sex) for accurate results.  
  Solution: Mapped requirements from the UI/UX specification into Pandas filter logic.
* Missing or Incomplete Data:  
  Some records were missing geolocation or other fields.  
  Solution: Added error handling and fallback values for charts and maps.
* Visual Layout:  
  Balancing clarity and information density for clients who are not technical users.  
  Solution: Used clear headers, spacing, and concise instructions in the dashboard layout.

## Usage

1. Launch the dashboard.
2. Select a rescue type using the dropdown to filter animals by mission suitability.
3. Explore the data table, click columns to sort, use the filter to search, and select rows.
4. View the breed chart and map, which update in real time based on your selections.

**Required Functionality**

The dashboard provides the following key features, all verified with hands-on testing and the screenshots included below:

* Interactive filtering:  
  Users can filter animals by rescue type (Water, Mountain/Wilderness, Disaster/Individual Tracking, or All) using a dropdown menu.
* Dynamic data table:  
  Displays all animal records with pagination, sorting, column selection highlighting, and search/filter options. The table updates instantly when a filter is selected.
* Breed distribution chart:  
  A live pie chart shows the top 10 breeds present in the filtered dataset.
* Geolocation map:  
  Each animal’s geographic location is shown interactively on a map, updated when a row is selected in the data table.
* Branding:  
  The Grazioso Salvare logo with URL anchor tag and unique identifier are displayed in the dashboard header.

**Screenshots**

* Default dashboard (all animals)A screenshot of a computer

  AI-generated content may be incorrect.
* Water rescue filter appliedA screenshot of a computer

  AI-generated content may be incorrect.
* Mountain/Wilderness rescue filter appliedA screenshot of a computer

  AI-generated content may be incorrect.
* Disaster/Individual Tracking filter appliedA screenshot of a computer

  AI-generated content may be incorrect.
* Reset/all animalsA screenshot of a computer

  AI-generated content may be incorrect.

## Contact

Your name: Fnu Samim