# CS 340 README Animal Shelter

**Project Two by Daniel Williams**

**Animal Shelter System**

The objective of this application is to provide users with seamless access to the comprehensive database of animals within the CS 340 Austin Animal Center (AAC). Through this platform, users can efficiently search and employ filters to navigate the database. The primary intention is to cater to the needs of Grazioso Salvare, who commissioned this project to establish a versatile dashboard. A pivotal feature of this project is the integration of geolocation mapping, fostering user interaction, and facilitating the representation of data through interactive charts. These functionalities collectively enable the location and utilization of animals, particularly for search and rescue operations.

**Underlying Motivation**

The genesis of this program stemmed from the aspiration to rigorously test my adeptness in database management and the art of data manipulation. In this endeavor, Python emerged as the language of choice, seamlessly aligning with the underlying MongoDB architecture. Python's inherent versatility, succinctness, and compatibility with MongoDB rendered it a potent tool for this undertaking. Moreover, the expeditious compilation process, fueled by the use of Jupyter Notebook, added a layer of efficiency to the development workflow.

**Initial Steps**

**Kicking off this program entails the following steps:**

1. Initialization within the MongoDB environment followed by the importation of the CSV file named "aac\_shelter\_outcome.csv."

2. Subsequent creation of both a basic and an intricate index to effectively parse the intricate data encapsulated within the documents.

3. Rigorous establishment of user authentication mechanisms, featuring the creation of an Admin account as well as an "aacuser" account, thereby ensuring secure access to the database.

4. Provision of the prerequisite environment, either by installing or accessing Python, followed by the execution of the program within a notebook.

5. Culmination with the input of the designated dashboard address, thereby seamlessly connecting to the dynamic visualization.

**Prerequisites**

**Successful execution and engagement with this project necessitate the following:**

- A contemporary version of Python and Jupyter Notebook, imperative for running both the ".py" and ".ipynb" files.

- Unhindered access to libraries such as PyMongo, Dash, Plotly, Pandas, and Leaflet, which collectively empower the creation of robust functions within the Dash framework.