# Animal Shelter

## About

Animal Shelter is an application which is used to create, read, update, and delete (CRUD) records from a MongoDB database.

## Motivation

Animal Shelter was created to facilitate record keeping for the Austin Animal Control Center (AAC) database.

## Getting Started

To run the Animal Shelter application, you must have the following prerequisite software installed:

\* Python 3.6+

\* MongoDB

\* PyMongo

\* Jupyter Notebook

## Usage  
Animal Shelter can be executed from within Jupyter Notebook by performing the following actions:  
\* Open 'ProjectTwoDashboard.ipynb' notebook

\* Modify the port number used to instantiate the AnimalShelter object to match your environment  
\* Press the 'Run' button

\* Click 'Open in new window' link

\* Please wait patiently…

## Roadmap/Features

\* Enhance user interface to allow for INSERT, UPDATE, DELETE features

## Contact

ryan.debraal@snhu.edu

## License

[MIT](<https://choosealicense.com/licenses/mit/>)

## Functionality

The application in its current state can be used to navigate the AAC database, filter by various conditions, visualize the GPS location where the animal was found, and generate a breakdown of outcomes per filtered section.  
  
In the future this application can be expanded to create a fully realized CRUD application capable of maintaining the data retained and administrated by the Austin Animal Control Center.

## Tools and rationale

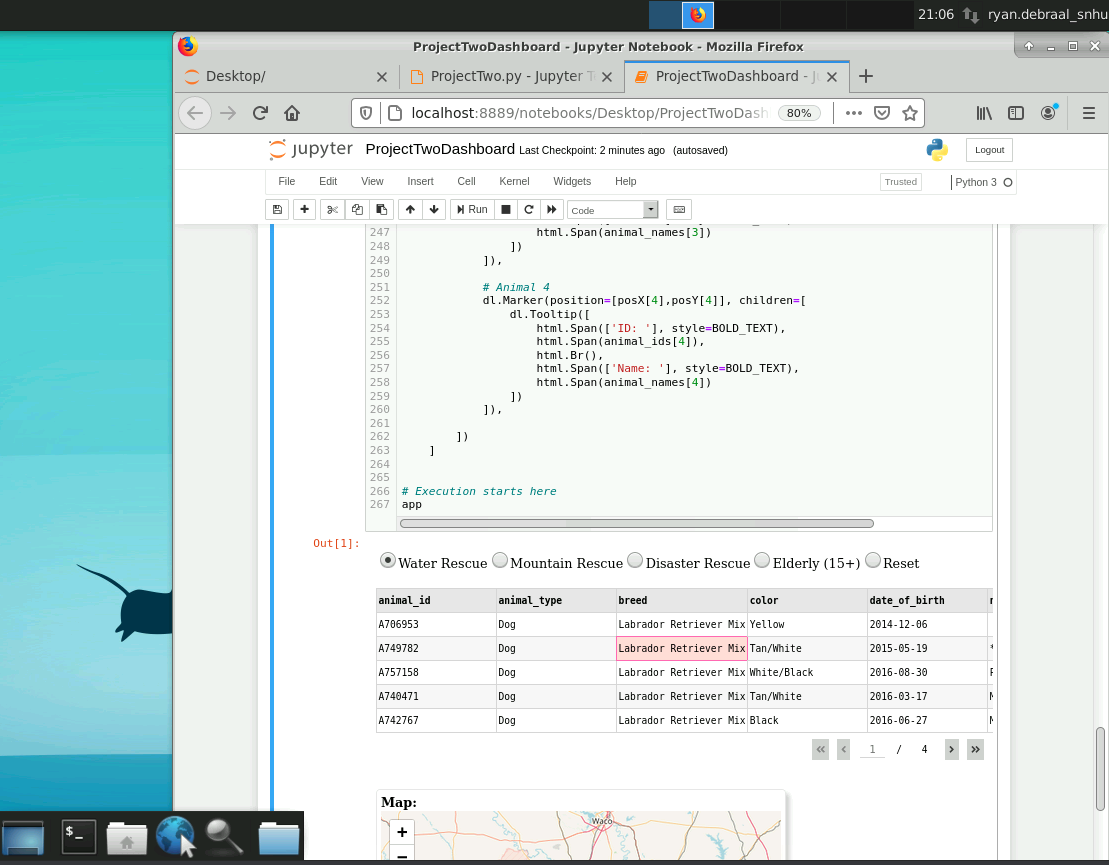
This application was developed to leverage the pre-existing instance of MongoDB, which is a document-based, “No-SQL” database.

The data access layer (DAL), or Controller in MVC parlance, utilizes PyMongo (<https://pymongo.readthedocs.io/en/stable/>) to create, read, update, and delete (CRUD) data.

The user interface, or View in MVC parlance, is powered by Dash (<https://dash.plotly.com/introduction>), a framework for building web analytic applications.  
  
These technologies were chosen for their relative ease of use and rapid stand-up time. Each component of this n-tier application was developed modularly so that it could be enhanced or swapped out easily in the future as need dictates. For example, if the client decided that they wanted the application to run on the Django web framework, they could simply reutilize the existing CRUD.py component and write a new user interface.

## Screenshots

The following reference screenshots show the application at runtime. To view these images in more detail please refer to the included reference screenshots or execute the application in your own instance of Jupyter Notebook.

### Development Environment  


### Water Rescue  
Graphical user interface, chart, application

Description automatically generated

### Mountain Rescue  
Chart

Description automatically generated

### Disaster Rescue Graphical user interface

Description automatically generated

### Elderly (15+) Chart, table

Description automatically generated

### ResetTable

Description automatically generated