# CS 340 README – William Cocomise

## About the Project

Grazioso Salvare is an international rescue-animal training company. The goal of this project is to make animal shelter data categorized and searchable. Which will aid in locating appropriate dogs for search-and-rescue training.

## Motivation

Grazioso Salvare collaborates with nonprofit animal shelters to identify candidates for their program. Global Rain is the custom software development firm contracted for this project. The deliverable full stack application will interact with shelter data. Helping identify and categorize dogs based on certain profiles.

## Getting Started

To get a local copy of this project up and running, follow these steps:

1. Clone the repository to your machine
2. Follow installation requirements below
3. Configure local database connection in AacCrud.py (<https://pymongo.readthedocs.io/en/stable/api/pymongo/mongo_client.html>)
4. Configure user authentication on local MongoDB instance (<https://www.mongodb.com/docs/manual/core/authentication/>)
5. Run the .ipynb test script in Jupyter Notebook

## Installation

Download and install these required tools:

1. Python (<https://www.python.org/downloads/>)
2. MongoDB (<https://www.mongodb.com/docs/manual/installation/>)
3. Jupyter Notebook (<https://jupyter.org/install>)

Included libraries (no installation required):

1. PyMongo

## Usage

This project has two major components. Data imports with MongoDB and CRUD operations via Python.

### Code Example

AacCrud.py module includes methods for CRUD operations:

# Create method to implement the C in CRUD.

def c(self, data):

if data is not None:

createData = self.database.animals.insert\_one(data)

return True if createData.inserted\_id else False

else:

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

return False

# Read method to implement the R in CRUD.

def r(self, data):

if data is not None:

readData = self.database.animals.find(data)

readList = list(readData)

return [] if not readList else readList

else:

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

return []

# Update method to implement the U in CRUD.

def u(self, data, newValue):

if data is not None:

updateData = self.database.animals.update\_many(data, {"$set":newValue})

return updateData.modified\_count

else:

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

return 0

# Delete method to implement the D in CRUD.

def d(self, data):

if data is not None:

deleteData = self.database.animals.delete\_many(data)

return deleteData.deleted\_count

else:

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

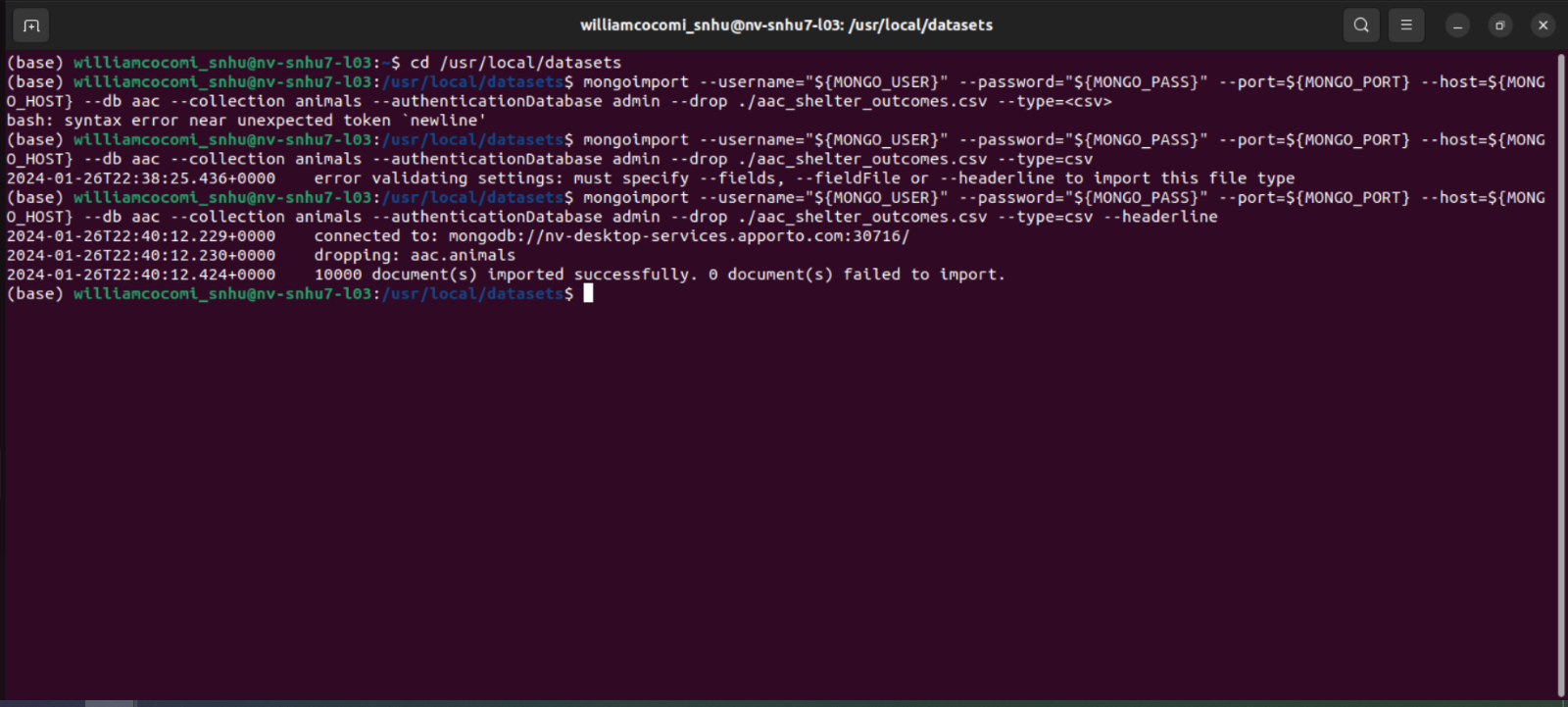
return 0

### Tests

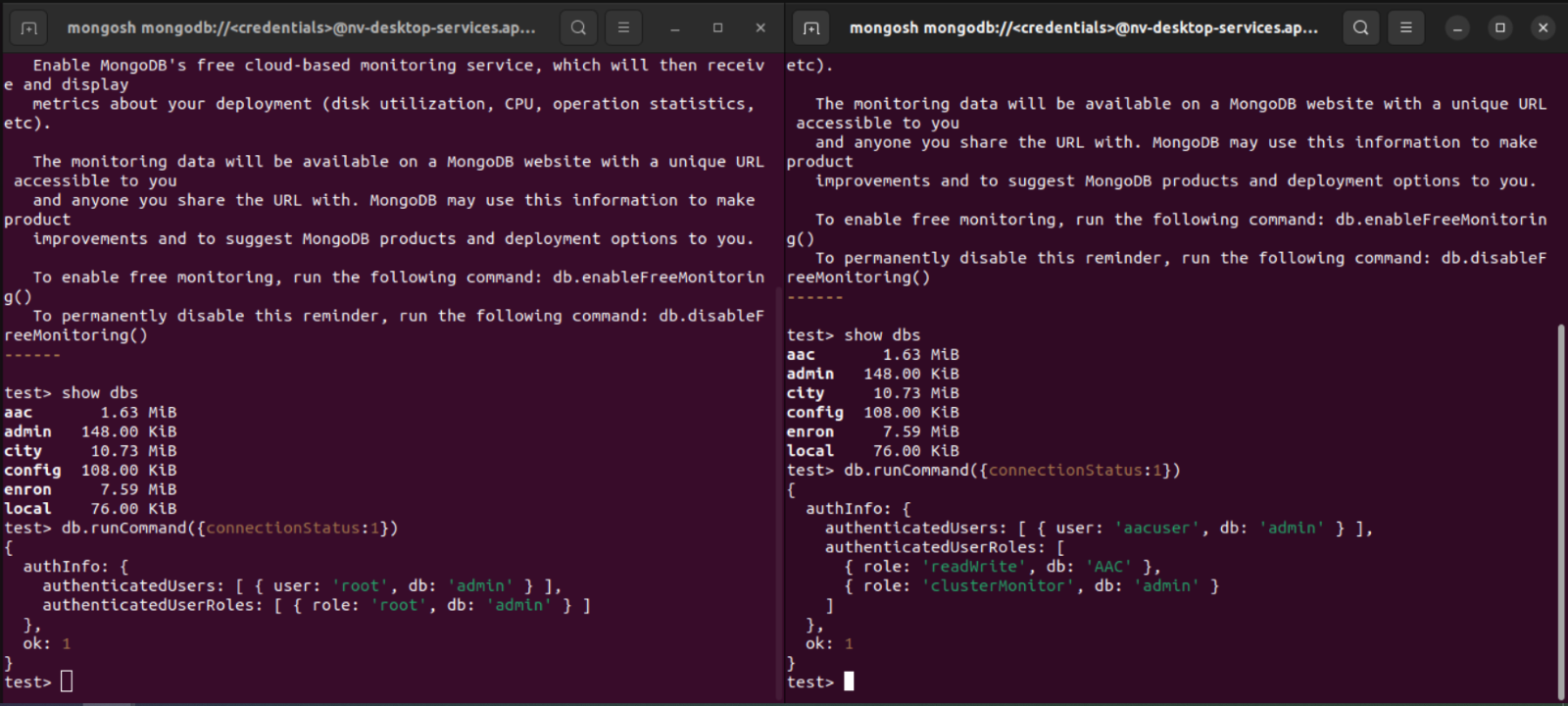
Testing script CS340-M4M-WC.ipynb can be run using Jupyter Notebook. This will ensure installation was successful and application is functional.

### Screenshots

Importing Shelter data via Mongo DB:

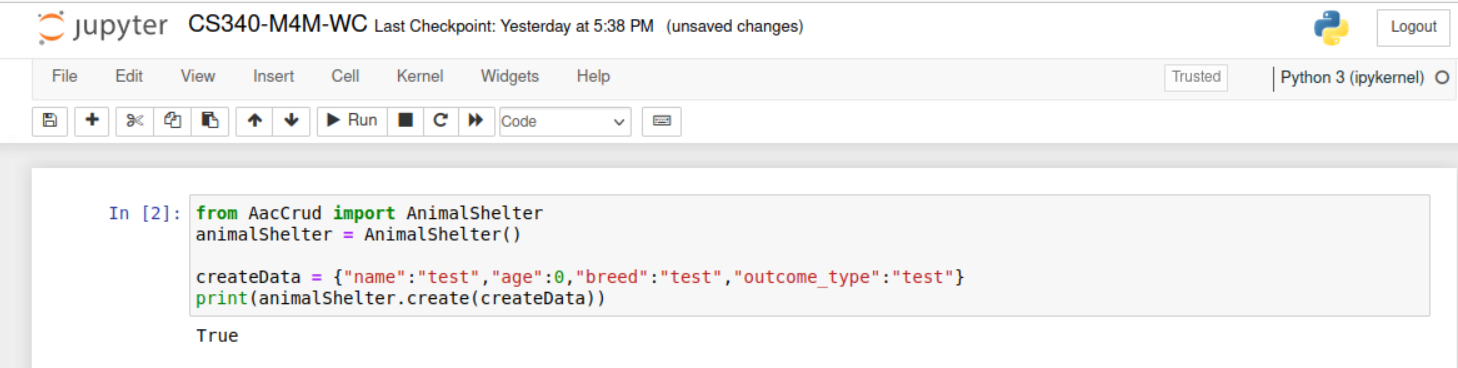


Admin and Non-Admin User Authentication:

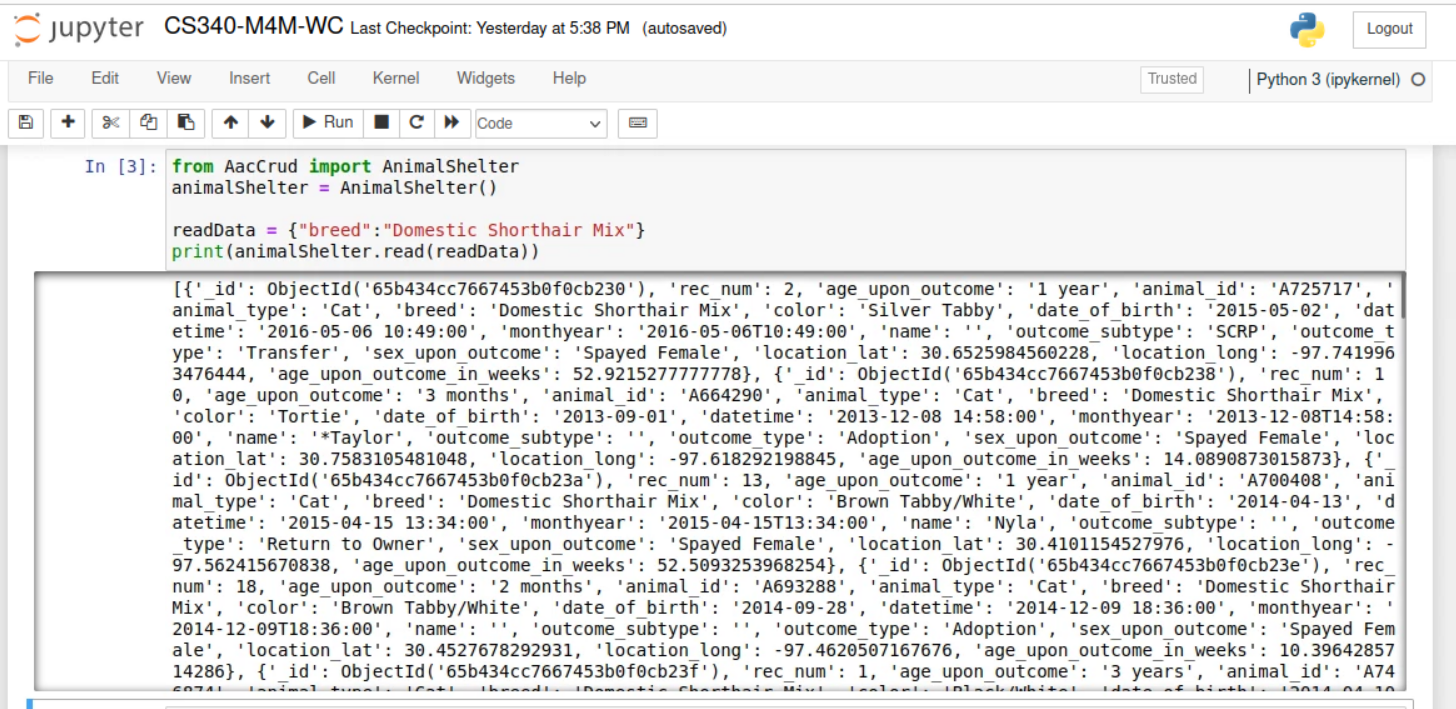


Test Scripts for CRUD Functionality:

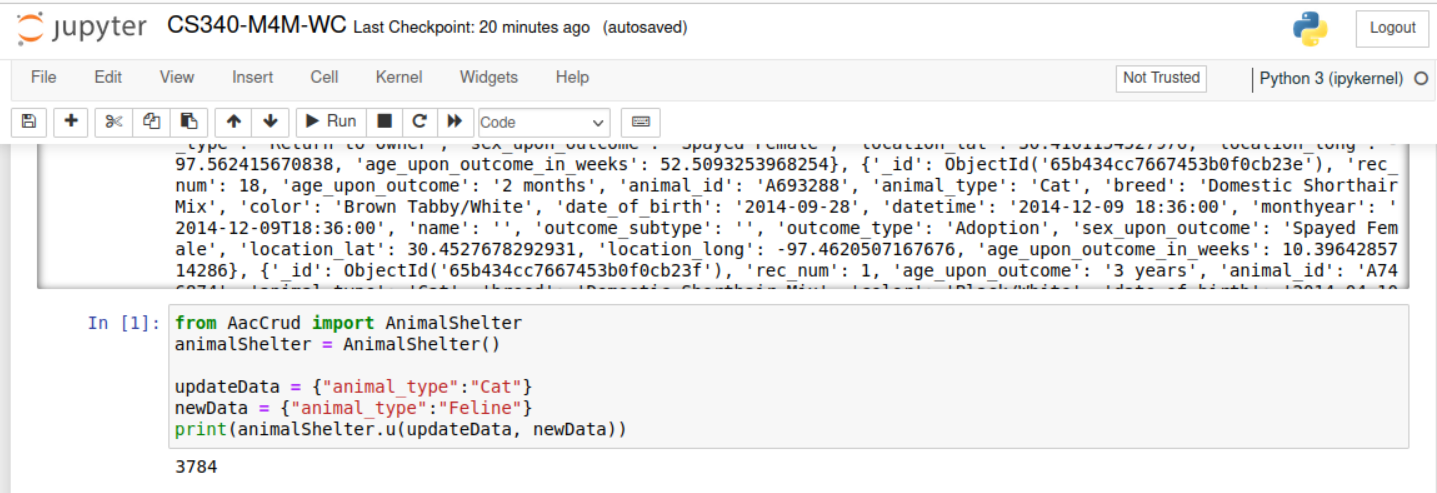
Create:



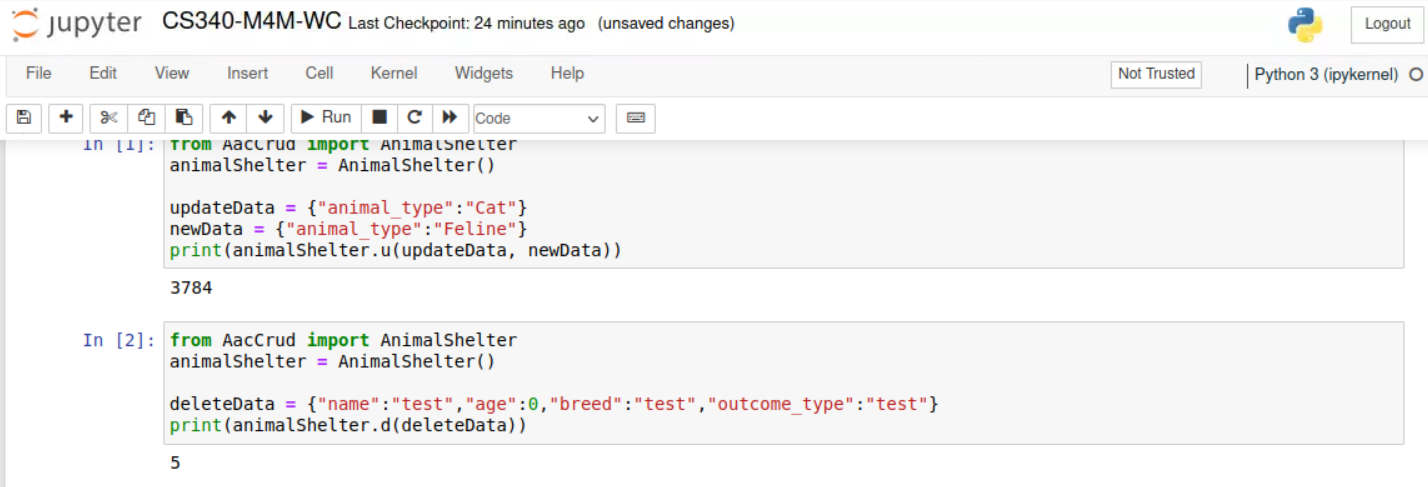
Read:



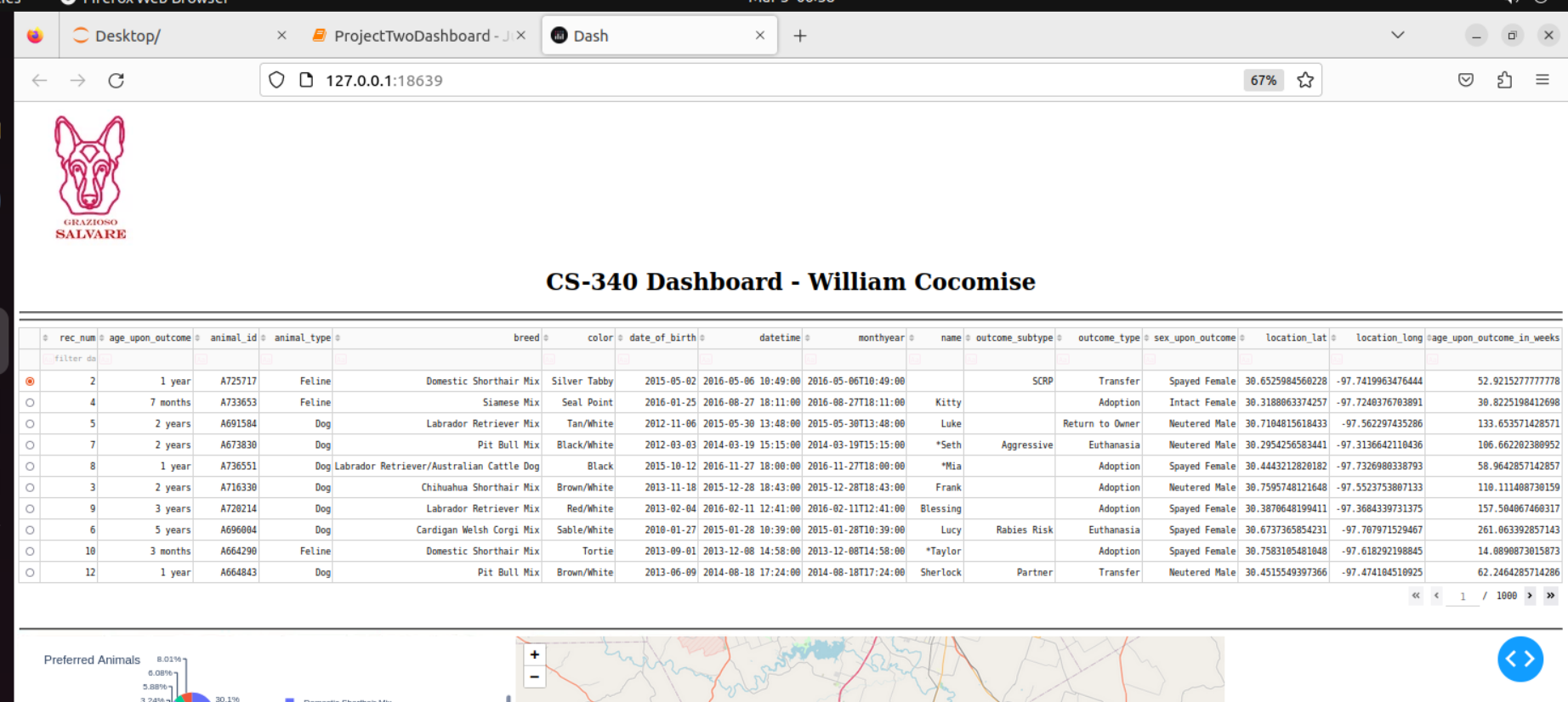
Update:

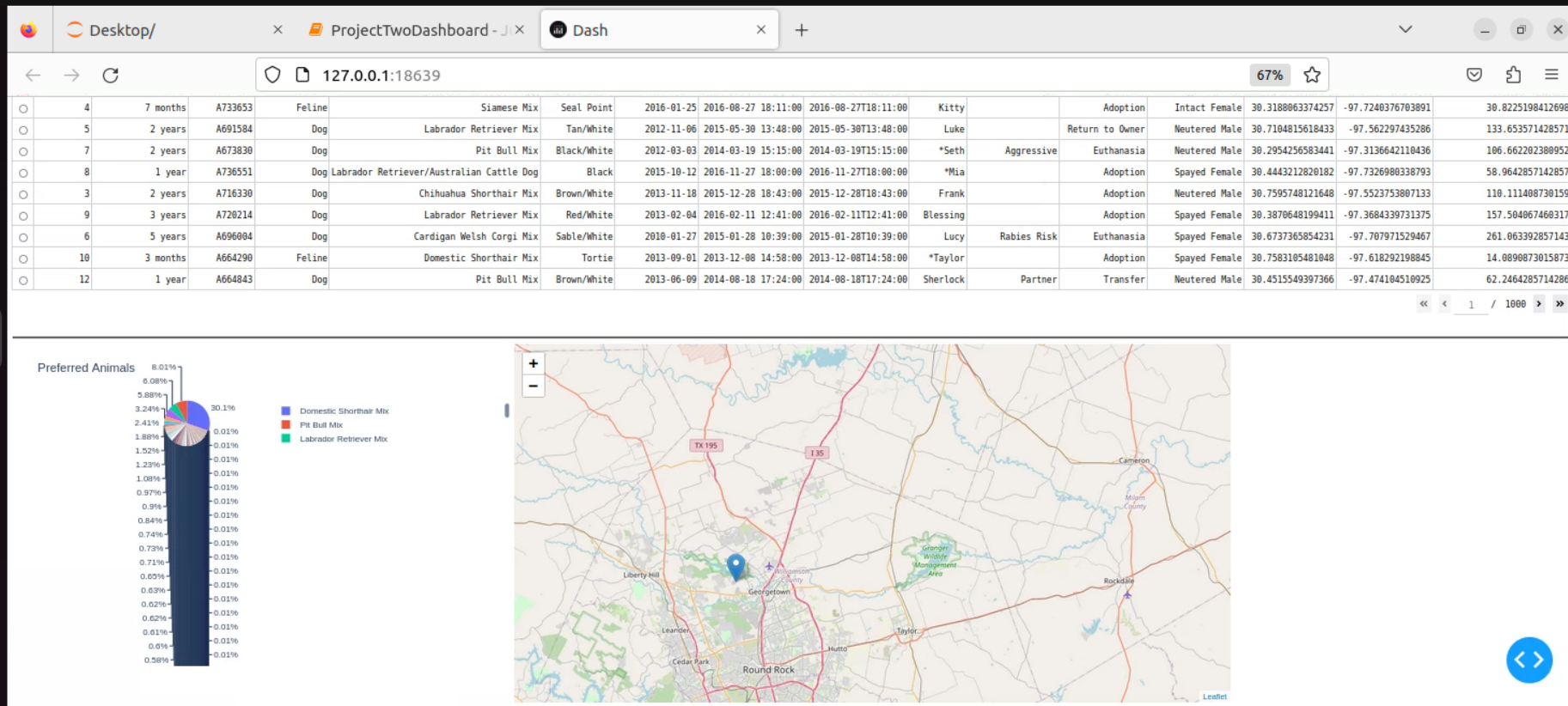


Delete:



Dashboard view:





## Contact

Lead Developer: William Cocomise, <https://github.com/b5yB>