# CS 340 README Template

## About the Project/Project Title

Global Rain was contacted by Grazioso Salvare to use the existing database and create an application for a dashboard to be used by them and other users. They want to be able to interact with the information from the databases by utilizing CRUD functionalities in mongoDB. The goal is to have an open source version for database visualization with Grazioso Salvare and giving a blueprint for other companies to utilize the same software to create their own versions.

## Motivation

The goal of this project is to have an application for finding ideal candidates for search and rescue training dogs. The database holds information from various shelters and the user will be able to find good matches for dogs to train through this dashboard.

## Getting Started

In order to get the project running you will need to log in to the mongoshell we have been using through the linuxshell for mongoDB. Then we can import specific data that we will utilize and create the index which will be handled in the mongoshell. We will then want to create a user account for the specific database so we are able to login when we access the dashboard. The generic source code will be open source on GitHub where you will be able to find the steps on how to add and view data in your own database. We have used Jupyter Noebook to create the necessary files for the pymongo requirements.

## Installation

Linuxshell – The linux shell is used for a lack of special commands that windows and mac can require. It is also more cost effective.

Mongoshell – Used to interact with the Mongo database we’re using

Jupyter Notebook – This is used to create mock code for what we decide to add to the database. We can create .py files and test out the functionality with python3 that will actually run our code.

Python – The simple language we’re using with more user friendly commands to easily make changes to the database.

PyMongo – We are using this to import python code into Mongo for our dashboard

RESTful – used in along with HTTP protocol to create the API

## Usage

Code examples, screenshots in the roadmap section.

mongoimport –port \*\*\*\*\* --db AAC –collection animals –type csv –headerline –file aac\_shelter\_outcomes.csv

db.animals.find({breed: "Maltese"}).explain()

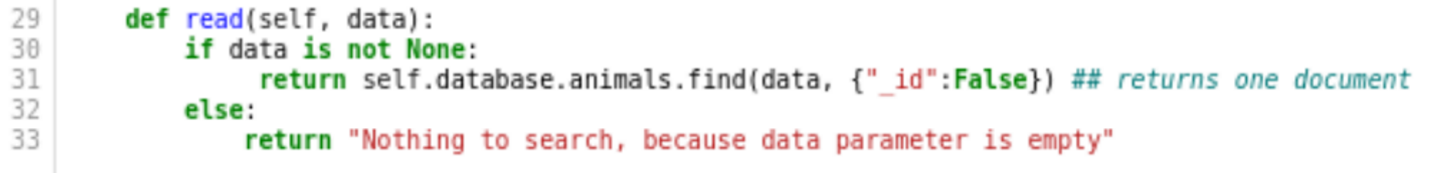
db.inspection.updateMany({“address.city”:”ROSEDALE”}, {$set:{“address.zip”:”76114”}})

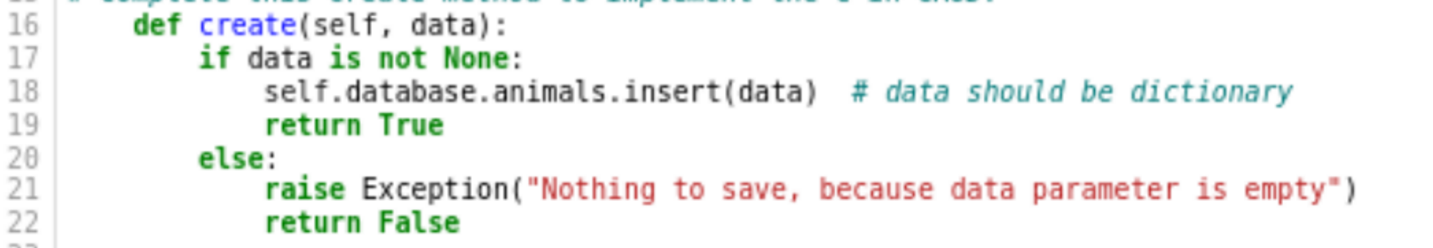
db.inspection.deleteOne({“result”:”Violation Issued”})

db.createUser({user: “aacuser”, pwd: “insertpassword”, roles: [{role: “readWrite”, db: “AAC”}]})

### Code Example

This code demonstrates how we can create and retrieve animals from the database

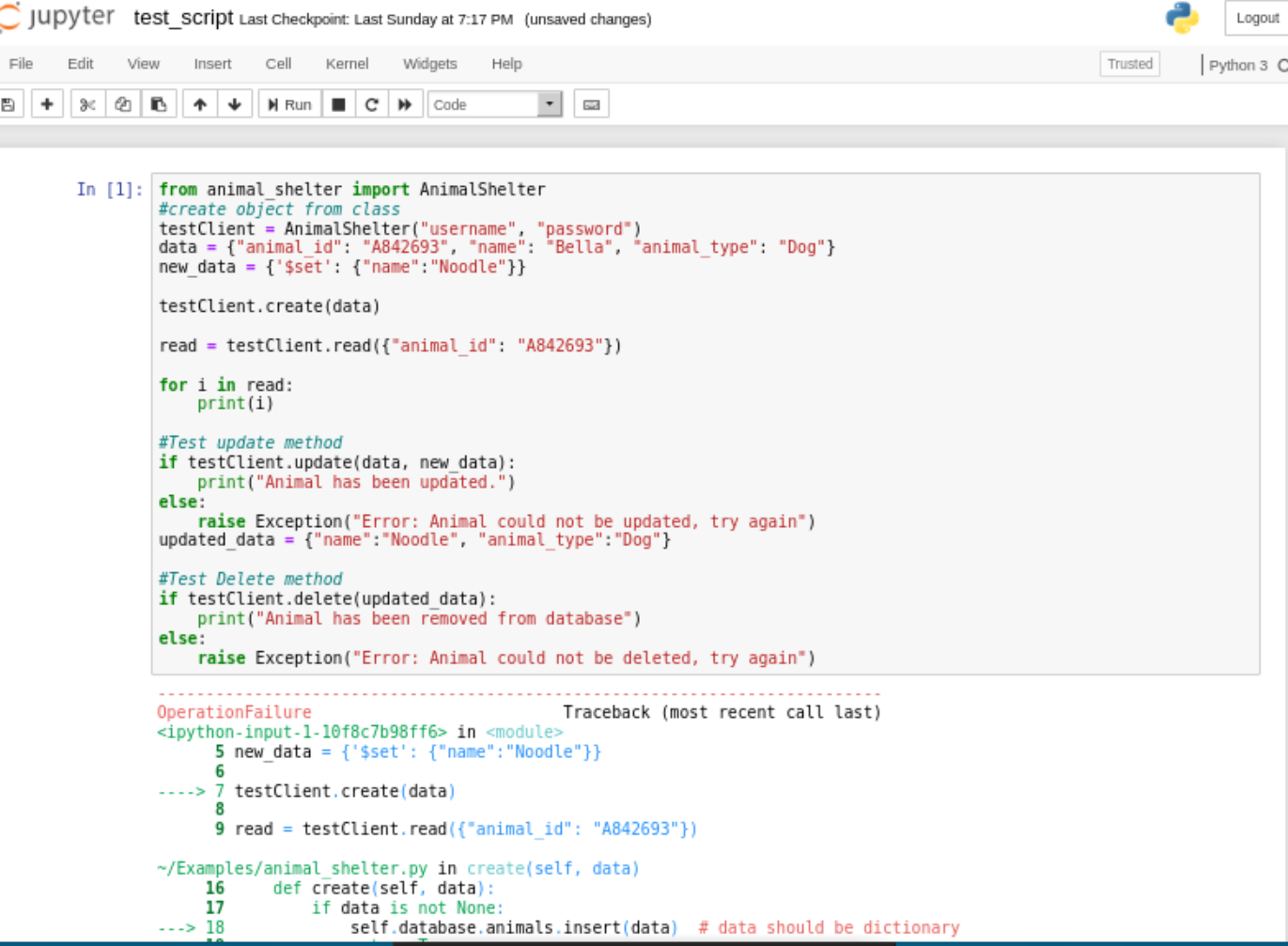
**

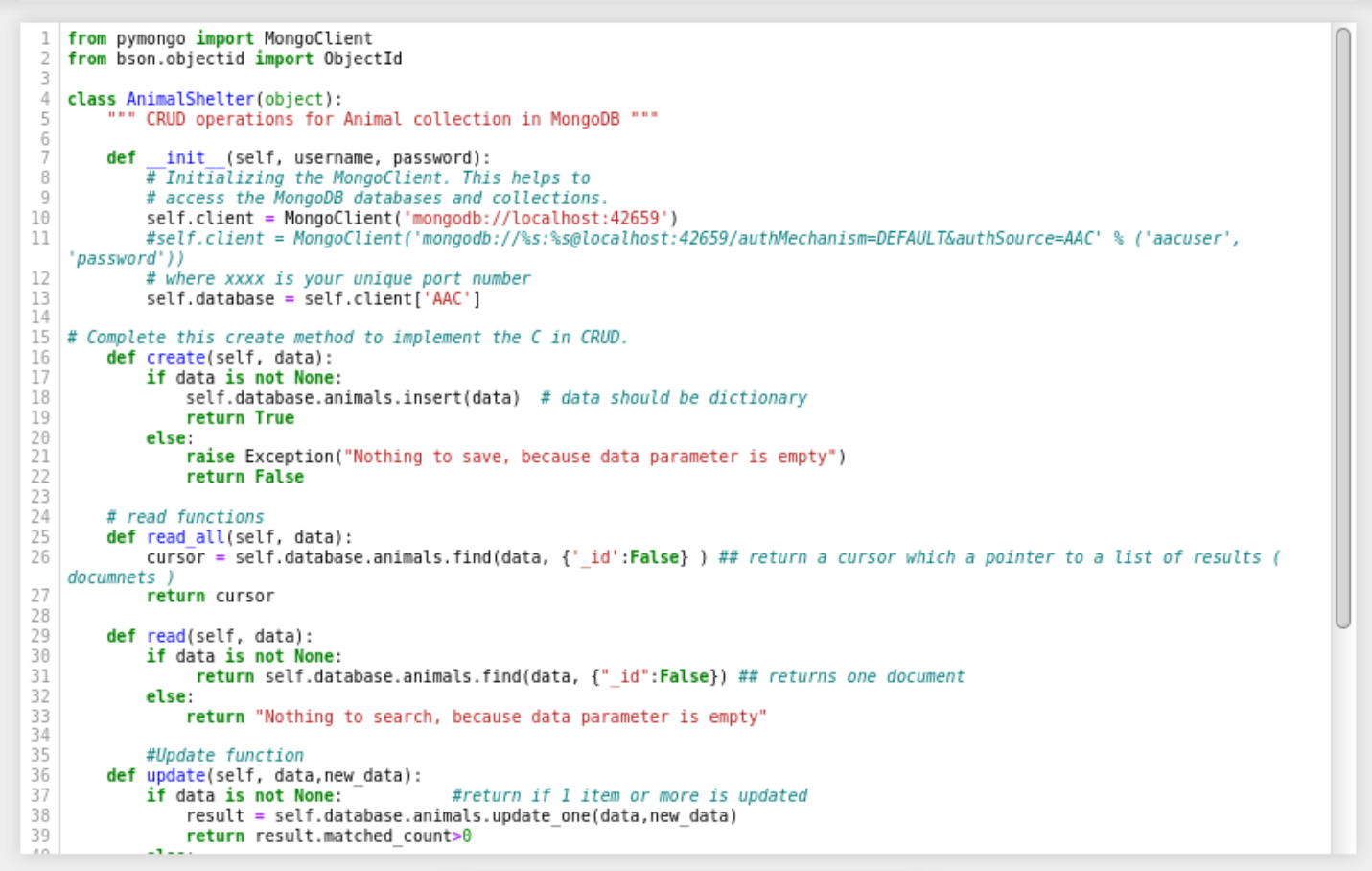


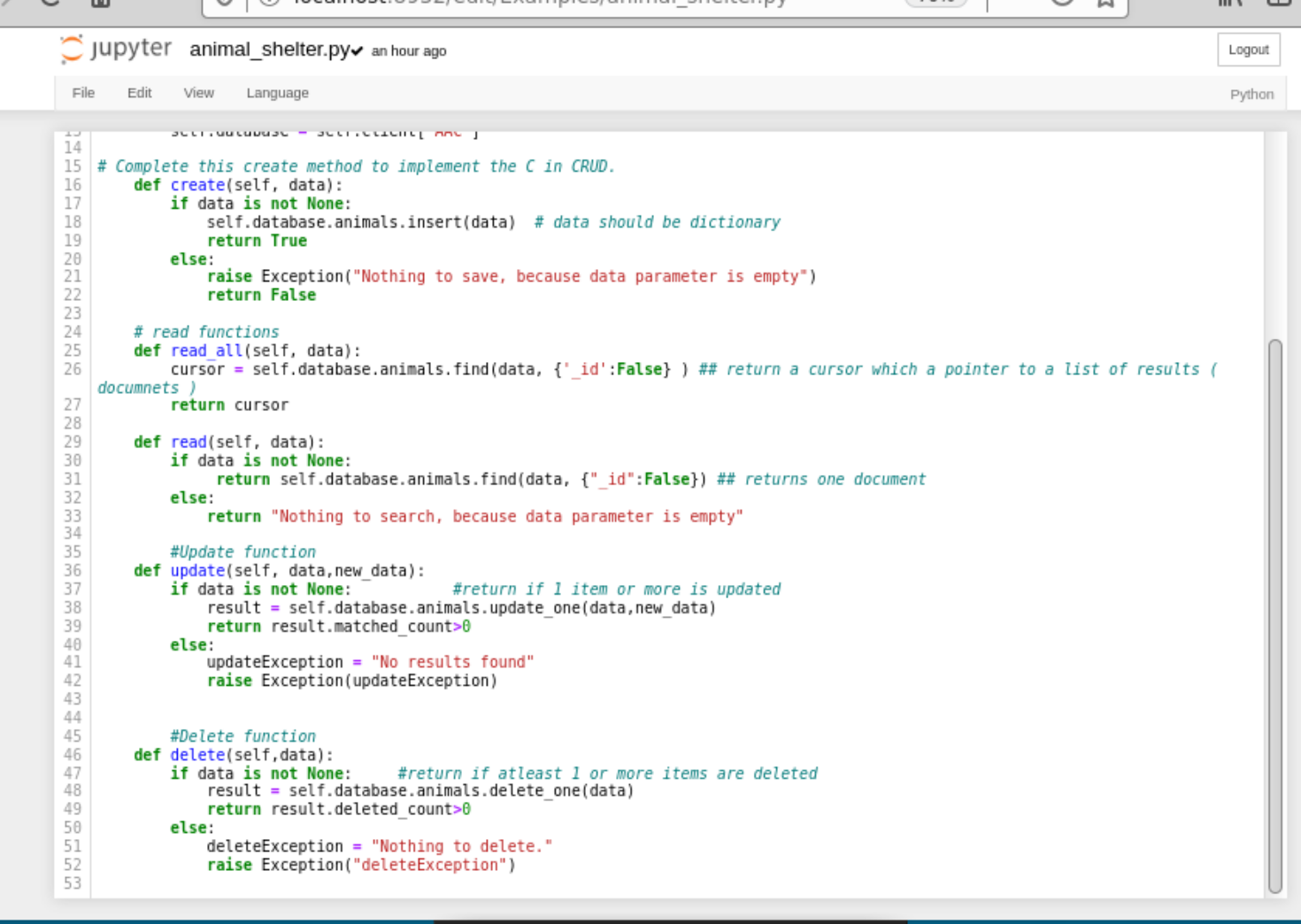
### Tests

An example of how to run the tests with this code in an ipynb file and loop after you store a read because it stores it in a pymongo object so you have to use the for loop to iterate through the object and pull the actual data

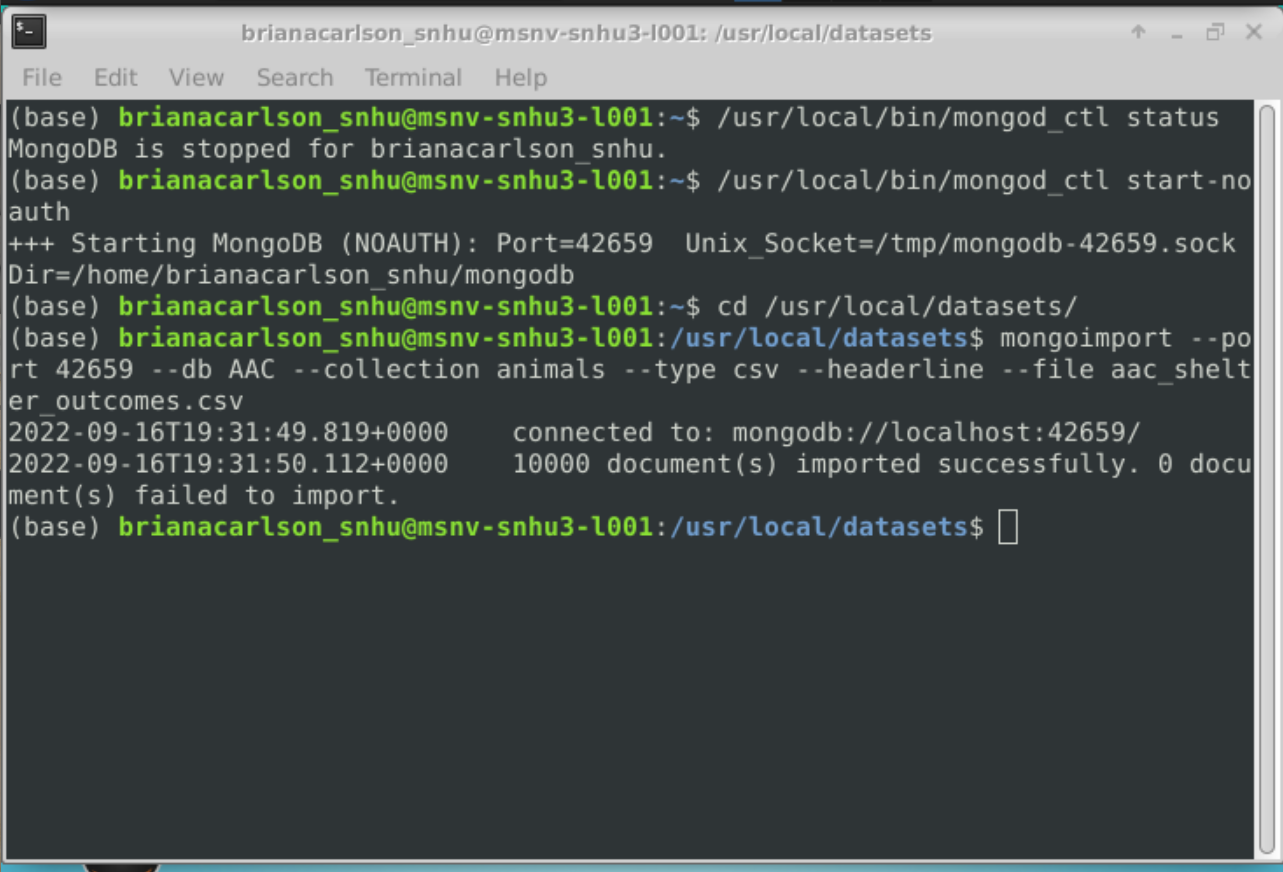
### Screenshots

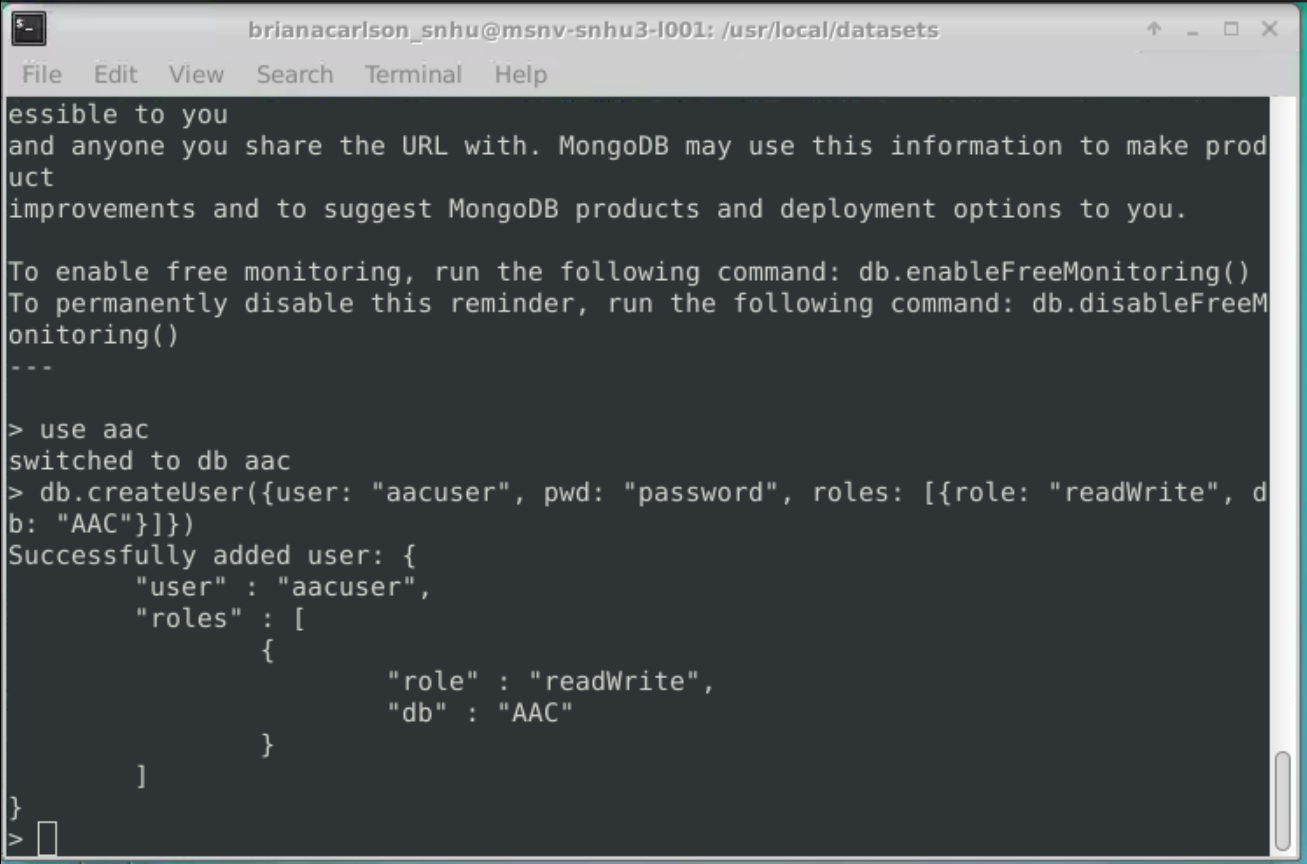
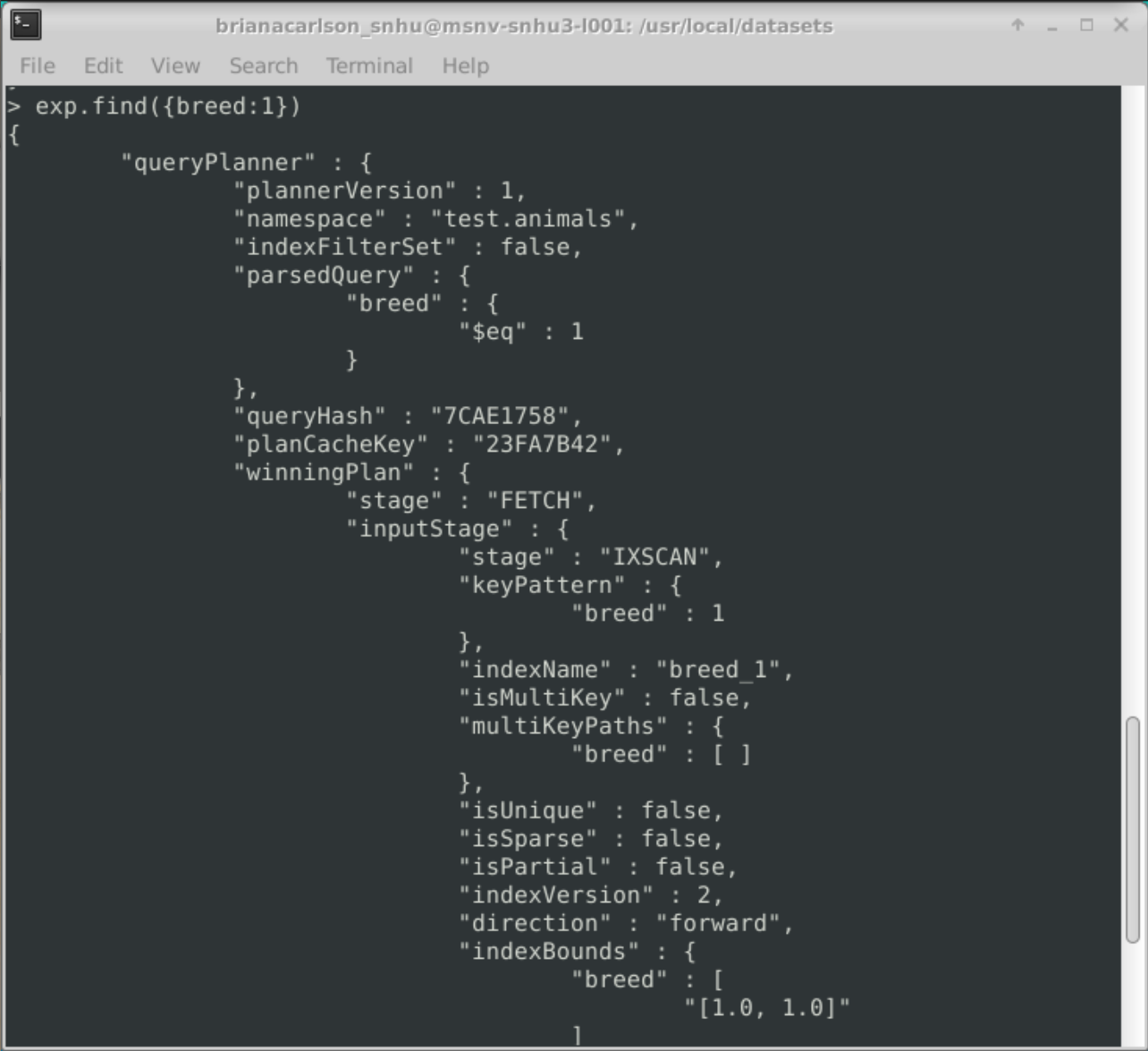
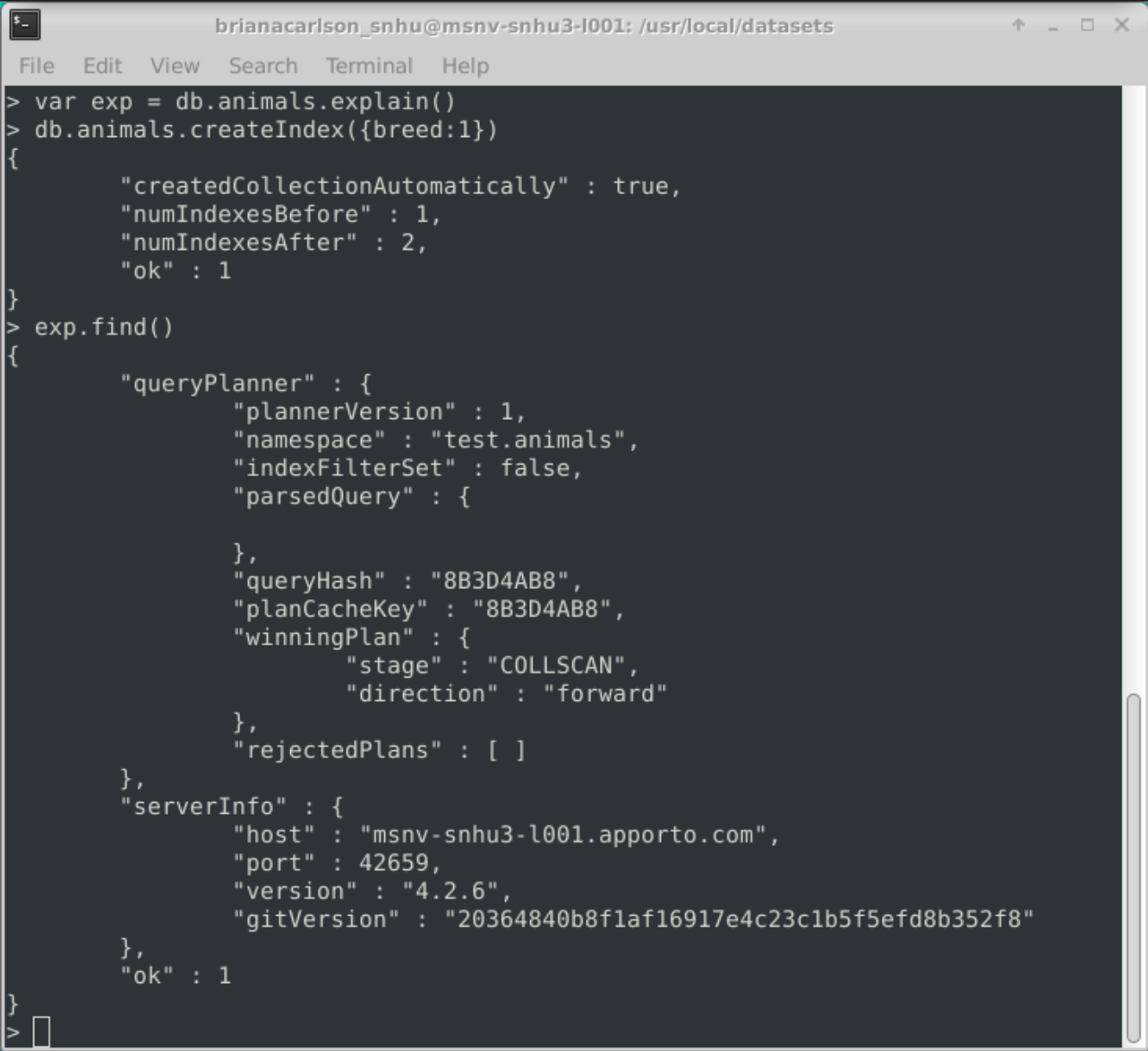






## Roadmap/Features (Optional)

**

**

**

## Contact

Your name:

Briana Carlson