7-1 Final Project

Portfolio

[GitHub Pages ePortfolio](https://tvsmith150snhu.github.io/SNHU-CS-Capstone/)

(https://tvsmith150snhu.github.io/SNHU-CS-Capstone/)

Computer Science Capstone (CS-499-H6771)

Troy Smith

Southern New Hampshire University

Professor Brooke Goggin M.S., M.S., M.S., Ed.D (ABD)

August 14, 2022

Table of Contents

Strengths (Self-Assessment)

Software Engineering and Security

I entered the Computer Science Program at Southern New Hampshire University in September of 2017 and continued that trek for five years while working a full-time career and supporting a family. I have learned how computer hardware, software and applications work in concert to provide problem solving tools for small and large business and personal users as well. At the base of these problem solving tools are well written and informatively commented code that afford computer scientists not only the ability to solve the problem, but allows others to enhance the code and make it better or more versatile to solve other problems or modernize it for future endeavors. This explores strategies for building collaborative environments that enable diverse audiences to support organizational decision making in the field of computer science and is shown in the Software Design and Engineering artifact presented in this portfolio in which I enhance the original authentication program written in Java by writing it in Python and improving the MD5 hash function's speed, reducing resource utilization on the computer's hardware. This artifact also demonstrates how I have developed a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources by creating the MD5 authentication and reducing the amount of resources necessary to run it. This artifact presentation also includes a flowchart and demonstrated my ability to design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts. These artifacts demonstrate my strength and ability to develop and maintain code in multiple coding languages and produce flowcharts or in some cases Pseudocode.

Data Structures and Algorithms

As a working professional, I hope to solidify my future as a computer scientist in my existing career and possibly advance into a more lucrative career, should the opportunity present. As data increases and technology advances, computer scientists will prevail as pillars within most if not all industries and be necessary to contend and maintain complex business systems. Being able to design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution, while maintaining the trade-offs involved in design choices is paramount. I demonstrate my ability to construct and work with algorithms and data structures in my Binary Search Tree algorithm artifact written in the C++ language. The original algorithm searches files in the CSV format, while the enhanced algorithm is written in the Python language and searches an XLSX formatted file using traversal method in a more efficient manner. These artifacts demonstrate my strength to work with algorithms and data structures as a competent computer scientist.

Databases

The ability to create, read, update and delete data within a secure and accessible database in another trait of a well-educated and versatile computer scientist. The ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals is demonstrated in my database artifact, where I display a dashboard created for a company to utilize a Mongo database, secure that database with credentials and am able to use the CRUD (Create, Read, Update and Delete) method within it. I then show my abilities to create an enhanced ReSTful (Representation, State, and Transfer) API using a locally hosted Mongo database that includes JSON formatted collections. These artifacts display my strength to create and use database structures.

Team Environments

During my courses at SNHU, I have learned that teamwork is imperative when designing complex programs and software. To keep the team focused and project milestones met, methodologies such as Agile were introduced and discussed during courses to introduce us to productive team concepts. Other sharing methods, such as GitHub and Bit Bucket were introduced to provide a team sharing platform to improve and advance code. These sites allow blame and version recovery capabilities to correct mistakes and educate team members in a constructive manner. Exposure to these team environments and tools will benefit me as I become a productive member of a team of computer scientists.

Communication to Stakeholders

The ability to communicate to stakeholders in written, oral and a visual manner is vital for a successful computer scientist. Written communication may be in the form of comments within lines of code, readme files or a simple email. Visual communication may be in flowcharts, diagrams or Power Point presentations. Oral communication may be in person, in virtual meetings or in videos, such as the Code Review milestone of this portfolio. Stakeholders may include potential employers, current superiors, customers, or other collaborators on the same team. Communication delivered in a concise and prompt manner will benefit and promote computer scientists that utilize it in an educated manner.

Milestone One

This Milestone includes a recorded narrative of my selection of artifacts for the portfolio, the review of the original code and my plan to enhance the code.

Please enjoy my video of Code Review.

Milestone Two

The Artifact chosen to display my work and skills learned in Software Engineering and Design is the Authentication program written in Java code and created in my IT 145 course on April 22, 2018. This program uses the MD5 hash method to authenticate users into the Zoo Authentication System allowing them three chances before stopping any further attempts. Once successful authentication is achieved, the user is greeted with a message according to their role in the Zoo system, which are admin, veterinarian, zookeeper or coder.

Inclusion of this artifact in my portfolio demonstrates my ability write code in the Java language to solve problems, in this case a means to authenticate users into a specific system. It also incorporates the Computer Science program outcome of demonstrating the ability to use well-founded and innovative techniques, skills and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry specific goals by proving my worth as a computer scientist that has the skills to code in Java and then transform that language into Python for the enhancement portion of the project or the second artifact in this Milestone. I achieved this in the enhanced code by reducing the lines of code and containing all the necessary methods within the main function, instead of multiple files, demonstrating innovative technique, skills and tools. Other evidence of the Computer Science program outcome is in the development of a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources. I achieve this in both artifacts as they both use the MD5 hash method to store and utilize the user password. I have the skills to develop the code in a readable and transformable format that may be built upon for further enhancement alone, by someone else or within a team and possess the traits of a promising new computer scientist.

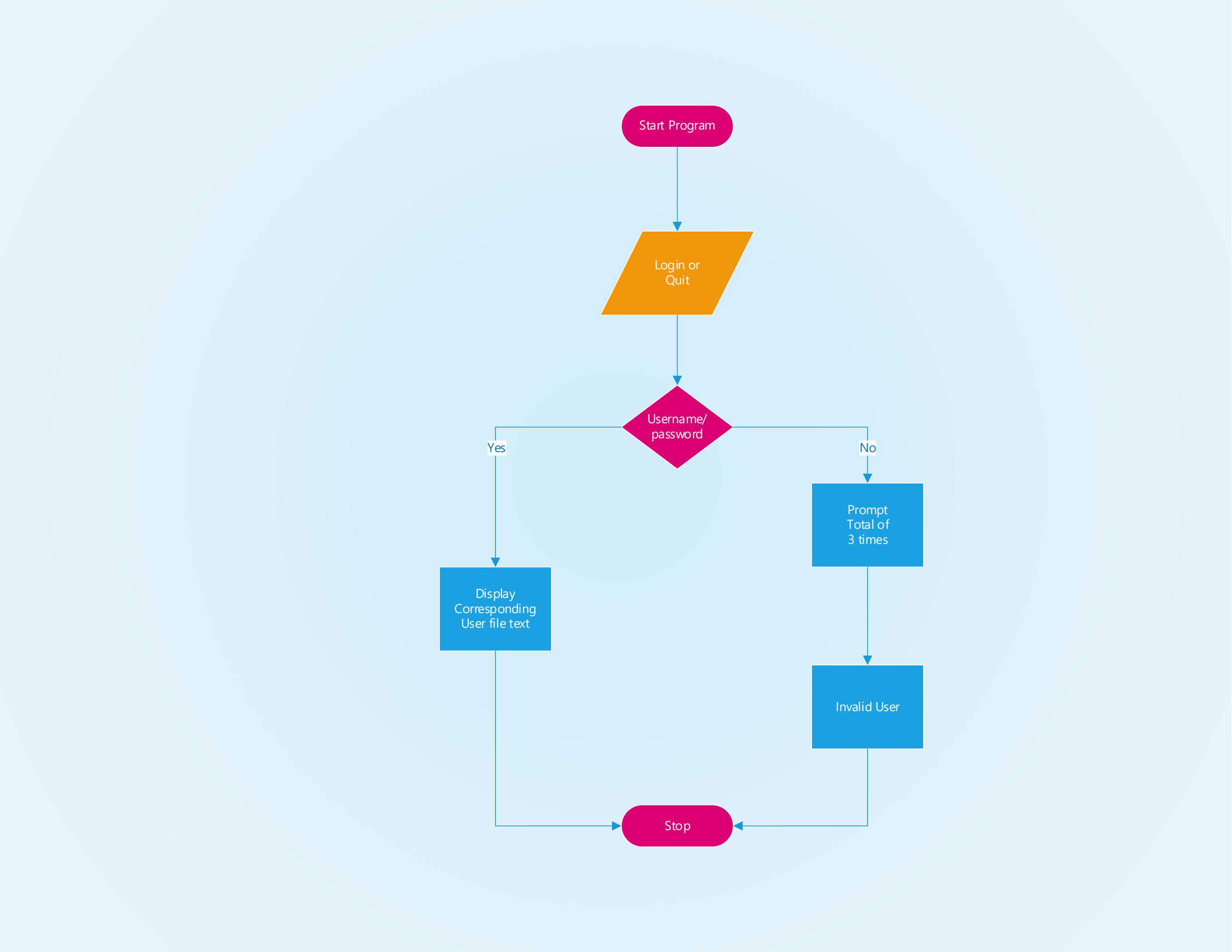
I have enhanced the program by writing it in the Python language and condensing it into less lines of code in one file instead of multiple files with more lines of code. The original code also required the program to completely exit or stop before another user could authenticate. The enhanced Python code allows the program to continue to run and allow another user to authenticate into the system.

Python was the language used in my first introduction to coding in my CS 200 course in 2017 and has continued to intrigue me as an easy to read language that is versatile enough to be used in many different problem solving situations. Although still new to coding and learning more each time I code, I find Python to be a very legible and straight forward language.

The original code had to be updated when files were moved or ran on other systems with a different file directory. Often errors occurred when the file could not be found. I learned this was not the case with the enhanced version and the Python code – the files opened flawlessly and less lines of code meant less work. The enhanced version does include the usernames within the code presenting somewhat of a security risk, but the passwords are kept hidden with the MD5 hash method. The original code kept the usernames and passwords in a separate credentials file that could be hidden or protected for security purposes. The argument could be made to keep the entire code hidden for both the Java and Python code in that regard.

Most of the challenge in enhancing the code was recalling the nuances of the Python language that I hadn't used in quite a while. Having a full time career while obtaining my Bachelor's degree has resulted in taking one course at a time over the span of five years. That leaves gaps in the last time I coded or used certain skills. Thankfully reviewing my notes and visiting site on the web and watching videos brought back enough remembrance to get me through the challenge.

<u>Flowchart</u>:


Start Program
Login or Quit
Username/ password
Yes
No
Prompt Total of 3 times
Display Corresponding User file text
Invalid User
Stop

****

Java Code:

```
/*Troy Smith
 *IT-145-X4663
 *Final Project
 *Option 1: Authentication System
 *April 22, 2018
 */

/*This program authenticates a user with the MD5 hash method
 *The user is greeted with message from a text file
 * according their role within the system.
 * The user is granted 3 tries before the system exits
 * and notifies them they have failed to authenticate.
 */

package zooauthentication;
//Import libraries and classes
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Scanner;
import static zooauthentication.UserRole.printAdminRole;
import static zooauthentication.UserRole.printCoderRole;
import static zooauthentication.UserRole.printVetRole;
import static zooauthentication.UserRole.printZooKeepRole;

public class UserLogin {
    //Method used to prompt user for login information
    public static void loginPrompt() {
```

```
//Declare string variable for user input
String userInput;
//Import scanner for user input
Scanner scanr4input = new Scanner(System.in);
//Post welcome message, then prompt user to Login or Exit
System.out.println("********************************************");
System.out.println("* Welcome to the Zoo Authentication System *");
System.out.println("********************************************");
System.out.println("Enter \"LOGIN\" to use the system, or");
System.out.println("Enter \"EXIT\" to exit the system: ");
userInput = scanr4input.nextLine();
//Allow user to exit system
if (userInput.equalsIgnoreCase(
        "EXIT")) {
    System.exit(0);
}
//Prevent user from entering random text.  Must enter "EXIT" or "LOGIN"
while (!userInput.equalsIgnoreCase("LOGIN")) {
    System.out.println("INVALID INPUT!  Enter \"EXIT\" or \"LOGIN\": ");
    userInput = scanr4input.nextLine();
    if (userInput.equalsIgnoreCase(
            "EXIT")) {
        System.exit(0);
    }
}
//Declare String Variable for user password
String usrPswrd;
//Declare flag to control flow in loops
boolean validUser = false;
//Declare variable to limit number of login attempts
int numTries = 3;
//Create scanner object to read credentials.txt file
BufferedReader credScanr = new BufferedReader(new InputStreamReader(System.in));
```

```
//Prompt user for username and password
//Loop for login attempts
try {

    do {
        //Decrement number of attempts to login
        numTries--;
        //Username prompt
        System.out.println("\nEnter Username or \"EXIT\" to quit: ");
        //Get username from scanner input
        String userName = credScanr.readLine();
        //Loop to provide exit before attempts to
        if (userName.equalsIgnoreCase(
                "EXIT")) {
            System.exit(0);
        }
        //Password prompt
        System.out.println("Enter Password: ");
        //Get username from scanner input
        String password = credScanr.readLine();
        //Convert password using MD5 hash
        MessageDigest md123 = MessageDigest.getInstance("md5");

        md123.update(password.getBytes());

        byte[] bytes12 = md123.digest();

        StringBuilder sb1 = new StringBuilder();
        for (byte b : bytes12) {
            sb1.append(String.format("%02x", b & 0xff));
        }
```

```
usrPswrd = sb1.toString();
//Declares variable from lines of text in credentials.txt file
String credFileLine;
//Open credentials.txt file
BufferedReader credFile = new BufferedReader(new FileReader("G:\\My Drive\\SNHU\\IT145\\Final Project\\ZooAuthentication\\ZooAuthentication\\src\\zooauthentication\\credentials.txt"));
//Loop to find MD5 hash text
while ((credFileLine = credFile.readLine()) != null) {
    //Create array with line containing MD5 hash
    String[] lineINcredfile = credFileLine.split("\t");
    //Parse username in credentials.txt file
    if (lineINcredfile[0].equals(userName)) {
        //Parse password in credentials.txt file
        if (lineINcredfile[1].equals(usrPswrd)) {
            //Flag determines true if MD5 hash was found
            validUser = true;
            //Exit loop
            break;
        }
    }
}

//Loop to search for user roles in credentials.txt file
if (validUser == true) {
    //Call method in UserRole class for admin.txt file
    if (credFileLine.contains("admin")) {
        //Call printAdminRole in UserRole class
        printAdminRole();
    }
    //Call method in UserRole class for veterinarian.txt file
    if (credFileLine.contains("veterinarian")) {
        //Call printVetRole in UserRole class
        printVetRole();
```

```
            }
            //Call method in UserRole class for zookeeper.txt file
            if (credFileLine.contains("zookeeper")) {
                //Call printZooKeepRole in UserRole class
                printZooKeepRole();
            }
            //Call method in UserRole class for coder.txt file
            if (credFileLine.contains("coder")) {
                //Call printZooKeepRole in UserRole class
                printCoderRole();
            }
            break;

        } else {
            //Notify user if login attempt fails
            System.out.println("\nInvalid Username or Password.");
            //Encourage user to keep going
            System.out.println(" Try again.");
            //Notify user of number of attempts remaining (3,2,1)
            System.out.println(numTries + " more attemptes remain.");
            //Loop condition to terminate login attempts
        if (numTries == 0) {
            //Print user notification of excessive attempts
            System.out.println("\nMAX attempts exceeded!");

            System.out.println(" Exiting ...");
            //Exit command
            System.exit(0);
        }
        }
      //Stopping point for login attempts
    } while (numTries > 0);
//Exception handler if credentials file fails to open
```

```
        } catch (NoSuchAlgorithmException | IOException e1) {
            System.out.println("\nFILE \"credentials.txt\" NOT FOUND!");
            System.out.println("Please update line 93 of UserLogin class to correct path\n");
        }
    }
}
```

Python Code:

```
# Troy Smith
# CS-499-H6771
# Milestone Two
# Zoo Authentication System (Converted from Java)

#This program allows a user to authenticate into the system and
#when successful authentication is acheived, the user is greeted
#with a welcome message according to their role in the system.

#Use the credentials.txt file for testing with usernames and passwords

# Imports
import sys
import hashlib


# Convert password into MD5 hash
# This function uses the hashlib library to convert the user password (string) into hash

def MD5ConversionFunc(userPassword, Type):
    password = userPassword
    userRole = Type
    hashed = hashlib.md5(password.encode())  # uses hashlib to encode hashed password

    hashed2 = hashed.hexdigest()  # returns the digest for the hashed pass; returns as string containing only -
    # hexadecimals

    passCheck(hashed2, userRole)

    return hashed2  # Returns hashed password


# Function that compares the entered password to the stored password
# The hashed passwords are stored in the lists

def passCheck(password, zooType):
    userPassword = password
    userRole = zooType

    # List to store passwords

    keeperPswrd = ["108de81c31bf9c622f76876b74e9285f", "17b1b7d8a706696ed220bc414f729ad3"]
    adminPswrd = ["3e34baa4ee2ff767af8c120a496742b5", "0d107d09f5bbe40cade3de5c71e9e9b7"]
    vetPswrd = ["a584efafa8f9ea7fe5cf18442f32b07b", "3adea92111e6307f8f2aae4721e77900"]
    coderPswrd = ["68e47349d02711aebd5dcc7478b3972d"]

    # Validate password and open the coresponding text file based on user role
    # Once authentication is successful, FileOpen is called to open the correct file

    if userPassword in keeperPswrd and userRole == "zookeeper":
        FileOpen(userRole)

    elif userPassword in adminPswrd and userRole == "admin":
        FileOpen(userRole)
    elif userPassword in vetPswrd and userRole == "veterinarian":
        FileOpen(userRole)

    elif userPassword in coderPswrd and userRole == "coder":
        FileOpen(userRole)

    else:
        print("\nInvalid Password!\n")  # If password not valid, return to login loop


# Function for opening and reading .txt files. Each file is related to one of the four user roles

def FileOpen(userFileSelect):
    if userFileSelect == "zookeeper":  # Read the zookeeper file and print text
        file1 = open("zooKeeper.txt", "r")
        print(file1.read())
        print("\n")
        LogOut()

    if userFileSelect == "admin":  # Read the admin file and print text
        file2 = open("admin.txt", "r")
        print(file2.read())
        print("\n")
        LogOut()

    if userFileSelect == "veterinarian":  # Read the veterinarian file and print text
        file3 = open("veterinarian.txt", "r")
        print(file3.read())
        print("\n")
        LogOut()

    if userFileSelect == "coder":  # Read the coder file and print text
        file4 = open("coder.txt", "r")
        print(file4.read())
        print("\n")
        LogOut()


# Logout function - Prompt user to logout or exit

def LogOut():

    logoutLoop = True

    print("Logout/return to menu or exit?")
    print("1 - Logout and return to menu")
    print("2 - Exit")

    while logoutLoop:  # Loop to ensure choice of 1 or 2

        logoutSelection = input()

        if logoutSelection == '1':
            logoutLoop = False
            main()  # Return to main function

        elif logoutSelection == '2':
            print("Goodbye!")
            sys.exit()  # Exit / terminates the program

        else:
            print("Error, please type 1 or 2\n")
            continue


# Main function - used to prompt user to login or exit
```

```
def main():
    menu = True
    attempts = 3

    print("Welcome to the authentication system.")

    while menu:  # Login loop

        print("Please enter a number to continue: \n1-Login \n2-Exit")
        menuSelection = input()

        if menuSelection == '1':  # Menu selection 1
            while attempts != 0:  # Counter to allow only 3 attempts
                print("Enter Username: ")
                userName = input()  # Store username

                print("Enter Password: ")
                userPassword = input()  # Store user password

                if userName in ["griffin.keys", "donald.monkey"]:
                    userRole = "zookeeper"  # Assigns user role zookeeper
                    MD5ConversionFunc(userPassword, userRole)  # Call md5 password conversion
                    attempts -= 1  # Attempt decrement

                elif userName in ["rosario.dawson", "bruce.grizzlybear"]:
                    userRole = "admin"
                    MD5ConversionFunc(userPassword, userRole)
                    attempts -= 1

                elif userName in ["bernie.gorilla", "jerome.grizzlybear"]:
                    userRole = "veterinarian"
                    MD5ConversionFunc(userPassword, userRole)
                    attempts -= 1

                elif userName in ["troy.smith"]:
                    userRole = "coder"
                    MD5ConversionFunc(userPassword, userRole)
                    attempts -= 1

                else:
                    print("Invalid user name!\n")
                    attempts -= 1  # Attempt counter decrement
                    print(attempts, "Attempt(s) left\n")

                    if attempts == 0:
                        print("\nOut of login attempts. Goodbye!")
                        sys.exit()  # Exits if max attempts is reached

                    continue  # Loop

            break  # Loop

        if menuSelection == '2':  # Menu selection 2

            yesNoLoop = True  # Variable for loop
            while yesNoLoop:  # Loop for yes or no choice

                print("Are you sure you want to exit? (y/n)")
                yesNo = input()

                if yesNo in ['y', 'Y']:
                    print("Goodbye!")
                    sys.exit()  # Exits if user input is upper or lowercase Y

                elif yesNo in ['n', 'N']:
```

```
                    yesNoLoop = False  # Ends loop and returns to main if user input is uppercase or lowercase N
                    main()

                else:
                    print("Error, please type y or n!")  # Error message for improper user input
                    continue # Loop
        else:
            print("\nError, please enter 1 or 2\n\n")  # Must input correct data
            continue  # Loop


if __name__ == "__main__":
    main()
```

Milestone Three

The Artifact I have chosen to display my work and skills learned in Algorithms and Data Structure is Binary Search Tree Algorithm from my CS 260 course on February 16, 2020. The program reads from a CSV file and uses the algorithm to search the data listed within the file. It then uses a linked list to store the data to sort and search it.

Inclusion of this artifact in my portfolio demonstrates my understanding of algorithms and data structure within the C++ coding language and provides evidence of the expected outcome of the Computer Science program by proving my ability to design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practice standards appropriate to its solution, while managing the trade-offs involved in design choices. I achieved this in my enhanced artifact by coding the new algorithm to decrease time and increase speed by using Python's traversal method and reducing the number of lines of code. The new code interacts with live data in a more modern XLSX format opposed to the CSV format by the original code.

I have enhanced the artifact by using the algorithm and data structure in a similar manner within the Python language and using a traversal method also. The original code accessed a CSV file, while the enhanced use of the algorithm accesses an Excel sheet in the XLSX format allowing the user to select from a menu of choices to display the items already in the file, add a linked list of data, add items to the spreadsheet, traverse or access the Binary tree in pre order, inversely or in order and then exit the program. This code is useful in creating user lists of any kind to store items within an Excel document and allows the user to add to the provided excel list within two columns – one with the name of the item and the other with the quantity of that item in an Items List.

The data structures used in the code are a singly linked list and a binary tree. The singly linked list uses one append method to append the beginning or front and uses nodes to traverse through the elements of data by setting the pointer to the consecutive or next node. The binary tree algorithm allows the user to add ten items to the root and contains the left and right pointers as well, which connect each child branch on either side of the tree. The nodes are then created by comparing the value of the higher branch on the tree or the parent nodes. The Python coding language uses three traversals of the binary tree – pre-order, in-order and post-order (reversal) within the program. The C++ language is more complex in memory usage in the utilization of pointers in comparison to the Python language, as Python uses a more “pre-determined” and easier method of implementing these data structures.

Python was the language used in my first introduction to coding in my CS 200 course in 2017 and has continued to intrigue me as an easy to read language that is versatile enough to be used in many different problem solving situations. Although still new to coding and learning more each time I code, I find Python to be a very legible and straight forward language. This project introduced me to using the Pandas library and some of its functions reading and writing to MS Excel type files that are useful to store items.

Challenges faced in this project were recalling how to configure the Eclipse IDE in the Windows environment, then discovering the Visual Studio Code IDE and configuring openpyxl, xlsxwriter and xlrd packages within it. I first contacted Professor John Watson from the CS 260 course who graciously sent the configurations page that I failed to save from the original course that explained how to set up Eclipse in Windows for the original code. However, during my wait for his email reply, I discovered my MacBook Air could run the code in Eclipse without any special configuration of the IDE! This surprised me because I replaced the MacBook since the course. Thankfully I allowed the MacBook to update from the previous one and it actually had the program stored in history! Even so, I needed another IDE that was versatile and provided a better environment for the Python enhanced code. I quickly discovered the Visual Studio Code IDE (not to be confused with Visual Basic Studio) and its intuitive qualities, but struggled to get the openpyxl, xlsxwriter and xlrd packages configured within that environment on both the Mac and Windows platforms. After a few YouTube video and Stack Overflow tutorials I was in business! I was able to compile the enhanced code and see the algorithm and data structure work as planned and then open the MS Excel worksheet to see the items I added to the list from the user choice menu.

This artifact and its enhancement meets the course objectives in innovative skills and techniques used for accomplishing my goals by displaying how I am able to implement algorithms and data structures to write and read from the Excel Workbook using intuitive user input from the code that is kept clean and legible in a manner that other computer scientists can review and improve. It solves user problems by allowing them to add data to the worksheet from within the program using libraries such as Pandas that uses openpyxl to write to the xlsx format. Overall this project gave me a better understanding of data structures and algorithms and how to use them to provide solutions and solve problems.

Pseudocode:

This code will input data from a CSV file into a container that can be sorted by two algorithms – a selection and quick sort method. It will produce an output menu of:

```
  Menu:
    1. Load Bids
    2. Display All Bids
    3. Find Bid
    4. Remove Bid
    9. Exit
  Enter choice:
```

PSUEDOCODE:

```
//============================================================
// Global definitions visible to all methods and classes
//============================================================
CREATE forward declarations;
DEFINE a structure to hold bid information;
        string bid ID;
        string title;
        string fund;
        double amount;
                SET amount = 0.0;
        DEFINE structure for tree node;
                left node (set to nullptr);
                right node (set to nullptr);
//============================================================
// Binary Search Tree class definition
//============================================================
CREATE private class;
    Node root;
        Void add node;
        Void inOrder node;
        Remove node function;
CREATE public class;
    Binary Search Tree;
    Void InOrder function;
    Void insert;
    Void remove;
    Bid search using bid Id string;
CREATE default constructor;
    Node “root” SET to nullptr;
CREATE destructor;
    Recurse from root deleting every node;
TRAVERSE the tree in order;
    CALL BinarySearchTree function;
INSERT a bid;
    CALL BinarySearchTree function :: Insert bid;
        IF list is empty (root = nullptr);
            INSERT bid as root node;
        ELSE
            ADD node after root node;
REMOVE a bid;
    CALL BinarySearchTree function :: Remove bid;
SEARCH for a bid;
    IMPLEMENT searching the tree for a bid using bid ID;
        Start searching from the root node;
        Keep looping downwards until bottom reached or bid is found;
            WHILE current node isn’t null;
                IF current node matches, return it;
                IF bid is smaller than current then traverse left;
                ELSE traverse right side of tree;
```

```
//============================================================
// Static methods used for testing
//============================================================
DISPLAY the bid information;
    bid ID: bid title, |, bid amount, |, bid fund;
PROMPT user for bid information;
    “Enter ID: “
        INPUT bid ID;
    “Enter title”
        INPUT bid title;
    “Enter fund”
        INPUT bid fund;
    “Enter amount”
        INPUT bid amount;
    RETURN bid;
LOAD a CSV file containing bids into a container;
    INITIALIZE the CSV Parser;
        TRY
            loop to read rows of a CSV file;
            FOR
                INITIALIZE a bid from data in current row;
                    bid ID;
                    bid title;
                    bid fund;
                    bid amount;
            PUSH this bid to the end;
                push_back (bid);
        CATCH
            list not found (if CSV file isn’t found);

//============================================================
// Quick Sort method logic over bid title
//============================================================
PARTITION the vector of bids into two parts, high and low;
    low bid = beginning;
    high bid = end;
    PICK the middle element as the PIVOT point;
        PIVOT = begin + (end – beginning) / 2;
        WHILE not done (done = false (Boolean));
            keep incrementing low as long as it is less than the pivot;
            keep decrementing high as long as it is less than the pivot;
                IF
                    low is greater than or equal to high (>=);
                        done is TRUE;
                ELSE
                    SWAP the low and high bids using the built-in
                vector method;
            MOVE endpoints closer;
RETURN high;
PERFORM Quick Sort method on bid title;
    IF
        there are zero or one bid;
          done – begin is greater than or equal to end (there is nothing to sort);
    PARTITION bids into high and low bids;
        SORT low bids;
            Recursively call quicksort using midpoint value (begin to mid);
        SORT high bids;
            Recursively call quicksort using midpoint value (begin +1 to end);
//============================================================
// Selection Sort method logic over bid title
//============================================================
INDEX the middle bid;
DEFINE POS
    POS is the position within the bids that marks sorted/unsorted
        FOR
            pos is less than (<) bid size;
```

```
                        min = pos;
                increment pos (pos ++);
                FOR
                        j = pos + 1;
                        IF
                                j < 0;
                                        min = j;
                IF
                        min does not equal (! = ) pos;
                                SWAP pos and min;

//==============================================================
// Main method
//==============================================================
PROCESS command line arguments;
        SWITCH
                Case 2: CREAT CSV path;
                        default is "eBid_Monthly_Sales.csv";
DEFINE a vector to hold all the bids;
DEFINE method for clock ticks;
CREATE menu for user input;
        WHILE
                choice does NOT = 9;


                        SWITCH
                                case 1:
                                        Load bids;
                                                include clock tick method;
                                case 2:
                                        loop and display the bids read;
                                case 3:
                                        INITIALIZE timer;
```

```
                    CALL selection sort method;
                case 4:
                    INITIALIZE timer;
                    CALL quick sort method;
    IF
        User input = 9;
            PRINT “Goodbye.”;
RETURN 0;
```

Data Structure Examples:

Classes:

```
# Class for linked list - Creates node containing head and next pointer

class node:
    def __init__(self, data=None):  # 'Constructor'
        self.data = data
        self.next = None


# Linked list class - Contains linked list print and add node
# Nodes are added to the front of the list instead of the back

class linkedList:
    def __init__(self):  # 'Constructor'
        self.head = node()  # head node
        self.tail = None  # Tail set to none
        self.length = 0  # Length

    # Add node to the front

    def addNodeF(self, newData):
        newNode = node(newData)

        # New node added to the head of the list

        newNode.next = self.head
        self.head = newNode

    # For printing linked list

    def printList(self):
        pn = self.head  # set head
        while pn:
            print(pn.data)  # print data
            pn = pn.next  # set to next


# Class for binary tree and search

class treeNode:
    def __init__(self, data=None):  # Constructor
        self.data = data
        self.left = None  # left branch
        self.right = None  # right branch

    # Create the tree - left and right nodes

    def addTree(self, data=None):
        if data == self.data:
```

```
        return

    # left branches (child)

    if data < self.data:
        if self.left is None:
            self.left = treeNode(data)
        else:
            self.left.addTree(data)

    # right branches (child)

    else:
        if self.right is None:
            self.right = treeNode(data)
        else:
            self.right.addTree(data)

# Tree traversal
# Pre order traversal

def traversal(self, root=None):
    elements = []  # list of elements
    if root:
        elements.append(root.data)  # current node
        elements += self.traversal(root.left)  # traverse left branch from root
        elements += self.traversal(root.right)  # traverse right branch from root

    return elements  # Return list of elements
```

Traversal in Binary Tree:

```
# Inverse traversal tree function

def userTreeSearchIn():
    menuLoop = True
    root = treeNode("Root")  # Assign root node

    while menuLoop is True:  # Menu loop

        print("Tree root is word 'Root' - Branches are created alphabetically")
        print("Would you like to add a new leaf? (y/n)")
        yesNo = input()

        if yesNo in ["y", "Y"]:

            # Loop for user to enter 10 'items' which will be put into the binary tree

            print("Enter 10 items: ")
            for i in range(10):
                tree = raw_input()
                root.addTree(tree)

            print("\n")
            print(root.inTraversal(root), "\n")  # Traversal for viewing tree
            menuLoop = False  # Stop menu loop

        elif yesNo in ["n", "N"]:
            menuLoop = False  # Stop menu loop

        else:
            print("Error, please enter y or n\n")
            continue  # Ensure proper input
```

Singly Linked List Search:

```
def listSearch(itemData, amountData):
    menuLoop = True
    items = itemData
    amounts = amountData
    lst1 = []  # List for items and amounts

    list1 = linkedList()
    list1.head = node(items)  # head node is set to the items in the spreadsheet
    n2 = node(" ")  # and white space
    n3 = node(amounts)  # node 3 for the amounts

    # Set node and list order

    list1.head.next = n2
    n2.next = n3

    while menuLoop is True:  # loop

        print("Would you like to add a new node? (y/n)")
        yesNo = input()

        if yesNo in ["y", "Y"]:

            # Iterates ten times for user objects
            # These objects are saved to a list

            print("Enter new node: ")
            print("Enter 10 objects: ")
            for i in range(10):
                n4 = raw_input()  # Takes input for list
                lst1.append(n4)  # Append the list
            list1.addNodeF(lst1)  # Places the list in the first node
            list1.printList()  # Display the linked list
            print("\n")
            menuLoop = False  # Stop loop

        elif yesNo in ["n", "N"]:
            list1.printList()  # Display linked list without user data
            print("\n")
            menuLoop = False  # Stop loop

        else:
            print("Error, please enter y or n\n")
            continue  # Ensure proper input
```

Accessing the .xlsx Format:

```
def addData():
    addLoop = True

    while addLoop:  # Loop for y/n selection

        print("Please type the name of the item: ")
        userItem = input()  # Input for the item

        print("How many? ")
        userAmount = input()  # Input for the quantity

        # Data frame for adding data to the excel spreadsheet
        df2 = pd.DataFrame({"items": [userItem], "Amounts": [userAmount]})
```

```
# Use openpyxl to write to file
writer = pd.ExcelWriter("ItemList.xlsx", engine="openpyxl", mode='a', if_sheet_exists='overlay')

# Load spreadsheet as workbook
writer.book = load_workbook("ItemList.xlsx")

# Copy the current spreadsheet
writer.sheets = dict((ws.title, ws) for ws in writer.book.worksheets)

# Read the Excel file
reader = pd.read_excel(r'ItemList.xlsx')

# Write to the spreadsheet
df2.to_excel(writer, index=False, header=False, startrow=len(reader) + 1)

# Cease inputs
writer.close()

print("Data added\nAdd more data? (Y/N)")

# Choice to add more data or return to menu
yesNo = input()

if yesNo in ['y', 'Y']:
    continue  # loop again

elif yesNo in ['n', 'N']:
    addLoop = False  # Stop loop
    main()  # the one and only main method
else:
    print("Error, Returning to menu\n")
    break  # Exit loop
```

Use of Pandas:

```
def main():
    # Import the data sheet from Excel

    # Pandas reads the Excel spreadsheet and assigns it to a variable

    df2 = pd.read_excel("ItemList.xlsx", header=None, names=["items", "Amounts"])

    # The two main columns are converted to lists (Items and Amounts)
    # The variables are also declared here to be passed to other functions

    items = df2["items"].values.tolist()
    amounts = df2["Amounts"].values.tolist()
    print("Data Loaded Successfully!\n")
```

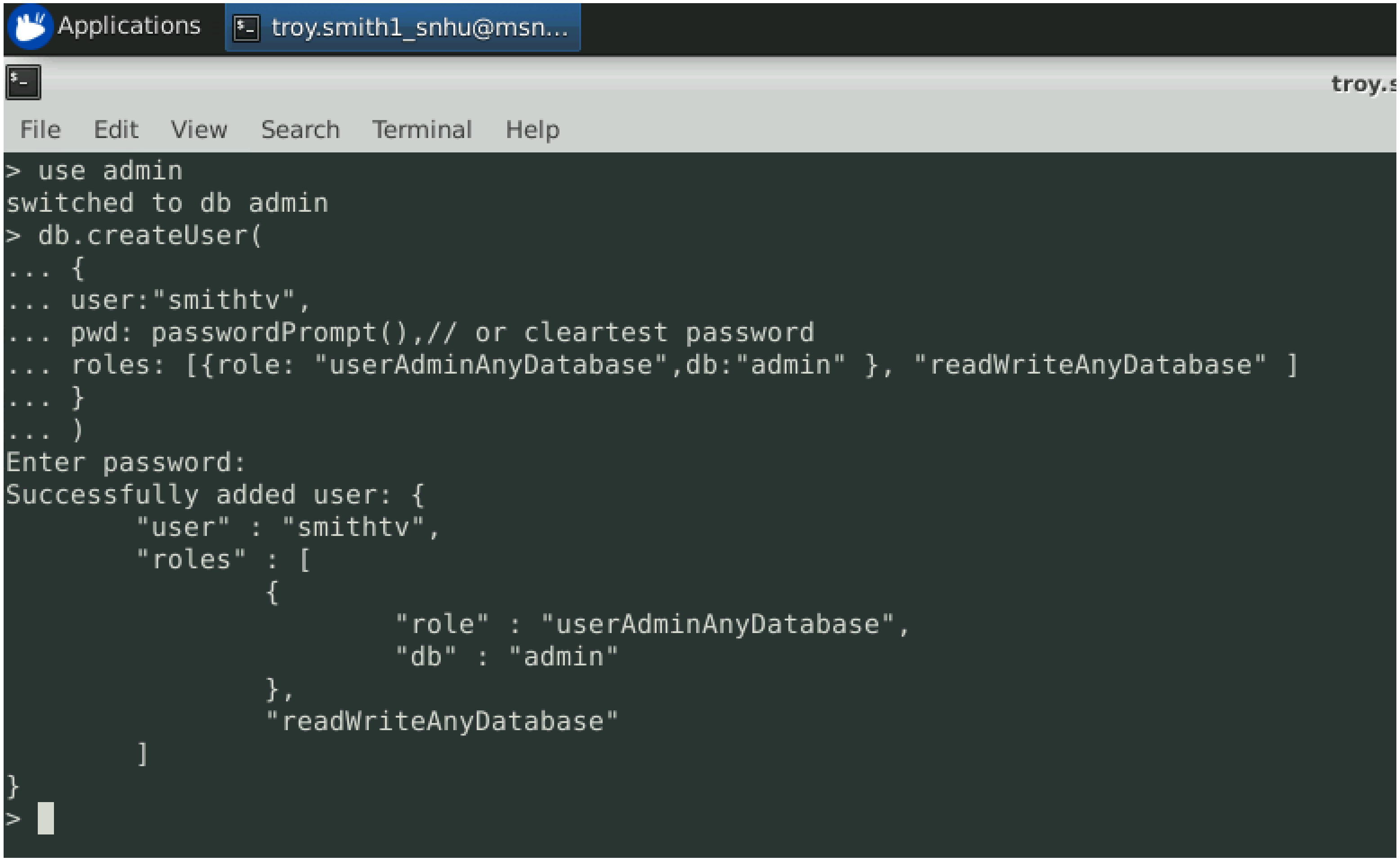
Milestone Four

The Artifact I have chosen to display my ability to store, manipulate and access data within a database is the one from my CS 340 course on October 18, 2021. The code worked within a provided Mongo database hosted on a Virtual Machine within the course. I also constructed code to provide a dashboard that accessed the database and referenced maps in correlation to the data.

Databases are included in most applications, websites and systems used throughout many industries and are required to be protected and maintained by Computer Scientists that are well equipped to handle the tasks. Inclusion of this database artifact in my portfolio demonstrates my ability to do just that and prove my worth and skills within database work and my ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals. I accomplished in both the original artifact and the enhanced artifact by developing a CRUD mongo DB allows users to create, read, update and delete items within collections contained within the database models. The original artifact also uses authentication methods to develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources.

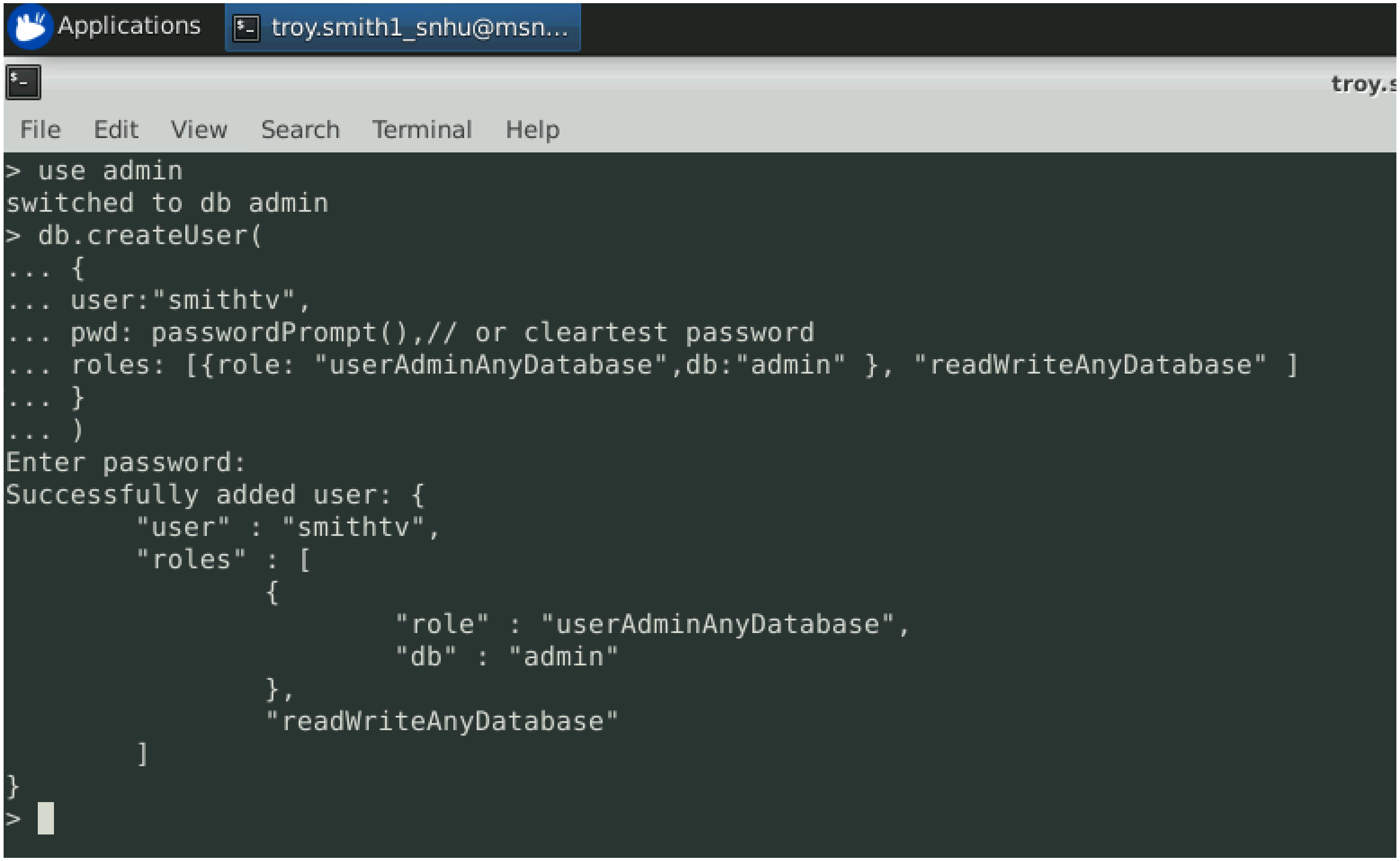
This is an example of me create an account within a Mongo DB for an administrator and user within a terminal.

Administrative:



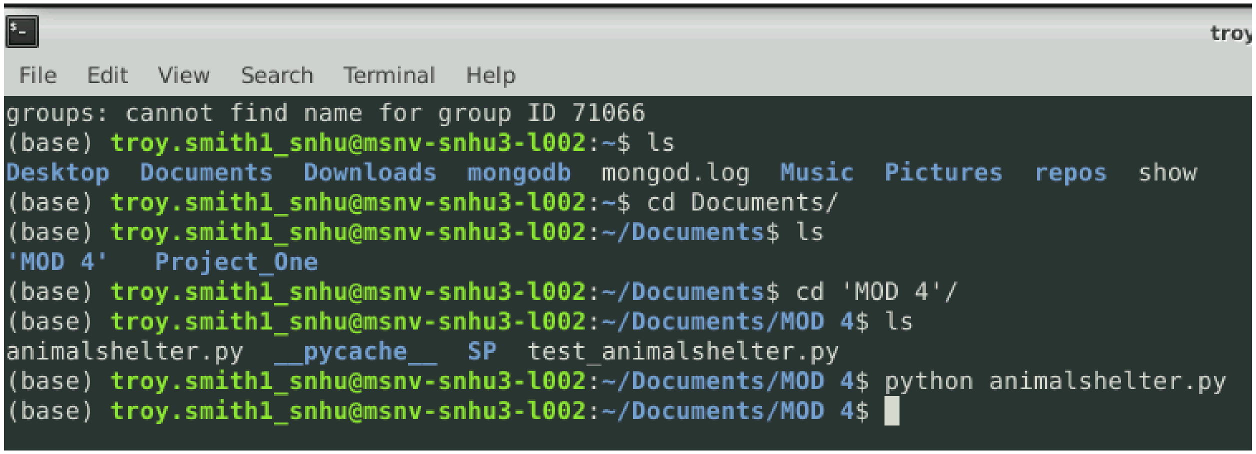
```
> use admin
switched to db admin
> db.createUser(
... {
... user:"smithtv",
... pwd: passwordPrompt(),// or cleartest password
... roles: [{role: "userAdminAnyDatabase",db:"admin" }, "readWriteAnyDatabase" ]
... }
... )
Enter password:
Successfully added user: {
        "user" : "smithtv",
        "roles" : [
                {
                        "role" : "userAdminAnyDatabase",
                        "db" : "admin"
                },
                "readWriteAnyDatabase"
        ]
}
> 
```

User:

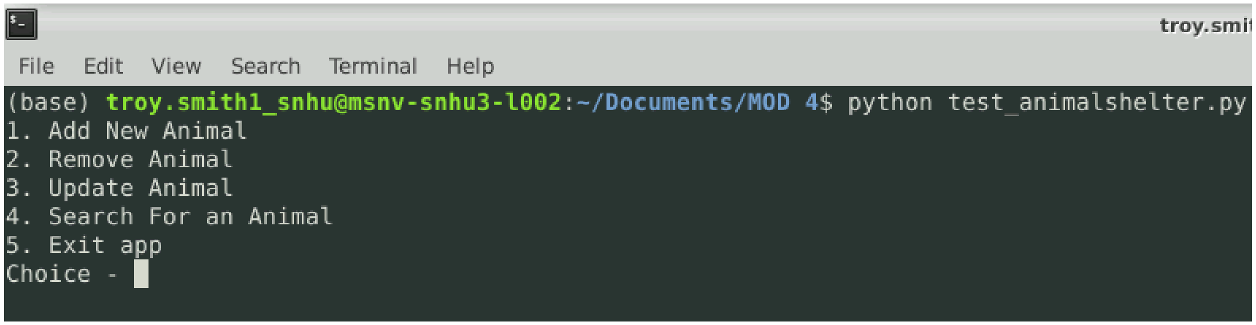


```
> use admin
switched to db admin
> db.createUser(
... {
... user:"smithtv",
... pwd: passwordPrompt(),// or cleartest password
... roles: [{role: "userAdminAnyDatabase",db:"admin" }, "readWriteAnyDatabase" ]
... }
... )
Enter password:
Successfully added user: {
        "user" : "smithtv",
        "roles" : [
                {
                        "role" : "userAdminAnyDatabase",
                        "db" : "admin"
                },
                "readWriteAnyDatabase"
        ]
}
> 
```

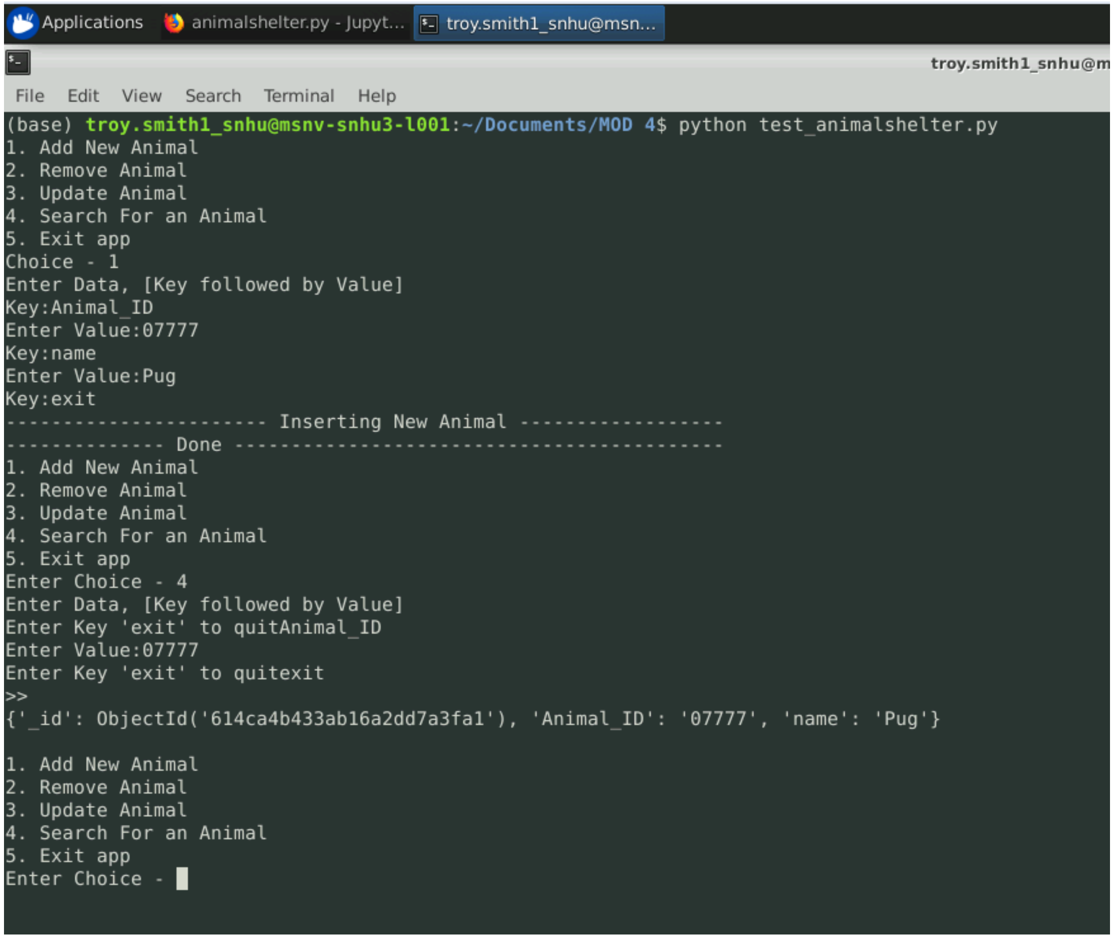
Here I create a Python test script that imports a CRUD Python module to call and test the ‘create’ and ‘read’ functionality.



```
groups: cannot find name for group ID 71066
(base) troy.smith1_snhu@msnv-snhu3-l002:~$ ls
Desktop  Documents  Downloads  mongodb  mongod.log  Music  Pictures  repos  show
(base) troy.smith1_snhu@msnv-snhu3-l002:~$ cd Documents/
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents$ ls
'MOD 4'   Project_One
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents$ cd 'MOD 4'/
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents/MOD 4$ ls
animalshelter.py  __pycache__  SP  test_animalshelter.py
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents/MOD 4$ python animalshelter.py
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents/MOD 4$ 
```

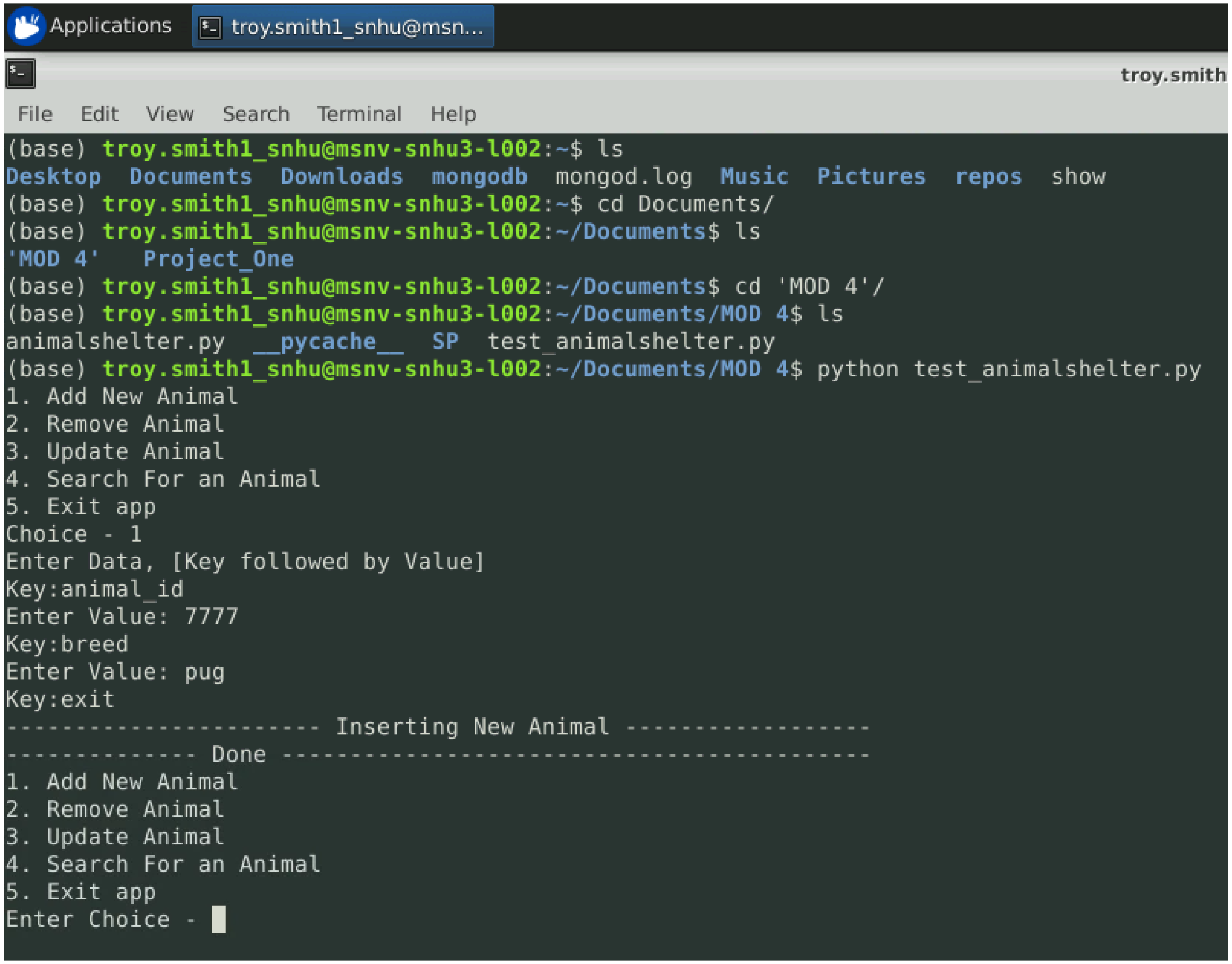


```
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents/MOD 4$ python test_animalshelter.py
1. Add New Animal
2. Remove Animal
3. Update Animal
4. Search For an Animal
5. Exit app
Choice - 
```


Applications
animalshelter.py - Jupyt...
troy.smith1_snhu@msn...
troy.smith1_snhu@m
File Edit View Search Terminal Help

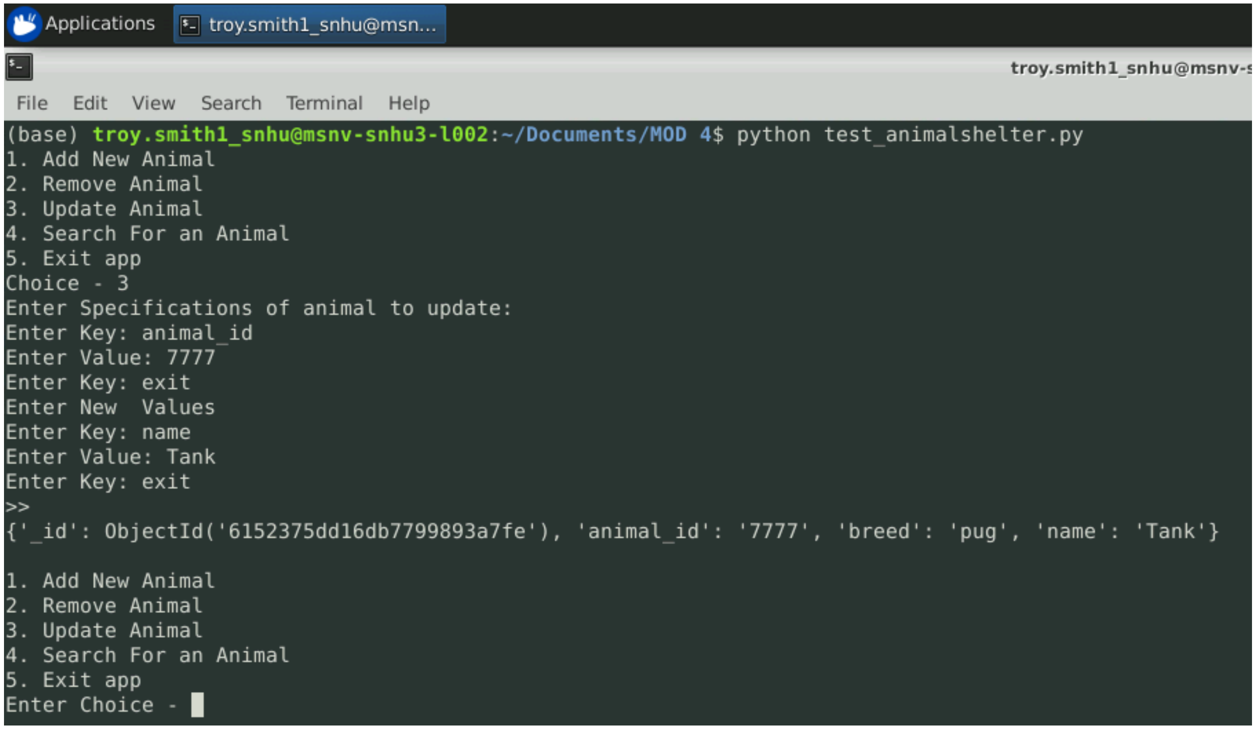
```
(base) troy.smith1_snhu@msnv-snhu3-l001:~/Documents/MOD 4$ python test_animalshelter.py
1. Add New Animal
2. Remove Animal
3. Update Animal
4. Search For an Animal
5. Exit app
Choice - 1
Enter Data, [Key followed by Value]
Key:Animal_ID
Enter Value:07777
Key:name
Enter Value:Pug
Key:exit
----------------------- Inserting New Animal -----------------
-------------- Done -------------------------------------------
1. Add New Animal
2. Remove Animal
3. Update Animal
4. Search For an Animal
5. Exit app
Enter Choice - 4
Enter Data, [Key followed by Value]
Enter Key 'exit' to quitAnimal_ID
Enter Value:07777
Enter Key 'exit' to quitexit
>>
{'_id': ObjectId('614ca4b433ab16a2dd7a3fa1'), 'Animal_ID': '07777', 'name': 'Pug'}

1. Add New Animal
2. Remove Animal
3. Update Animal
4. Search For an Animal
5. Exit app
Enter Choice - 
```

And add new Item (Create)



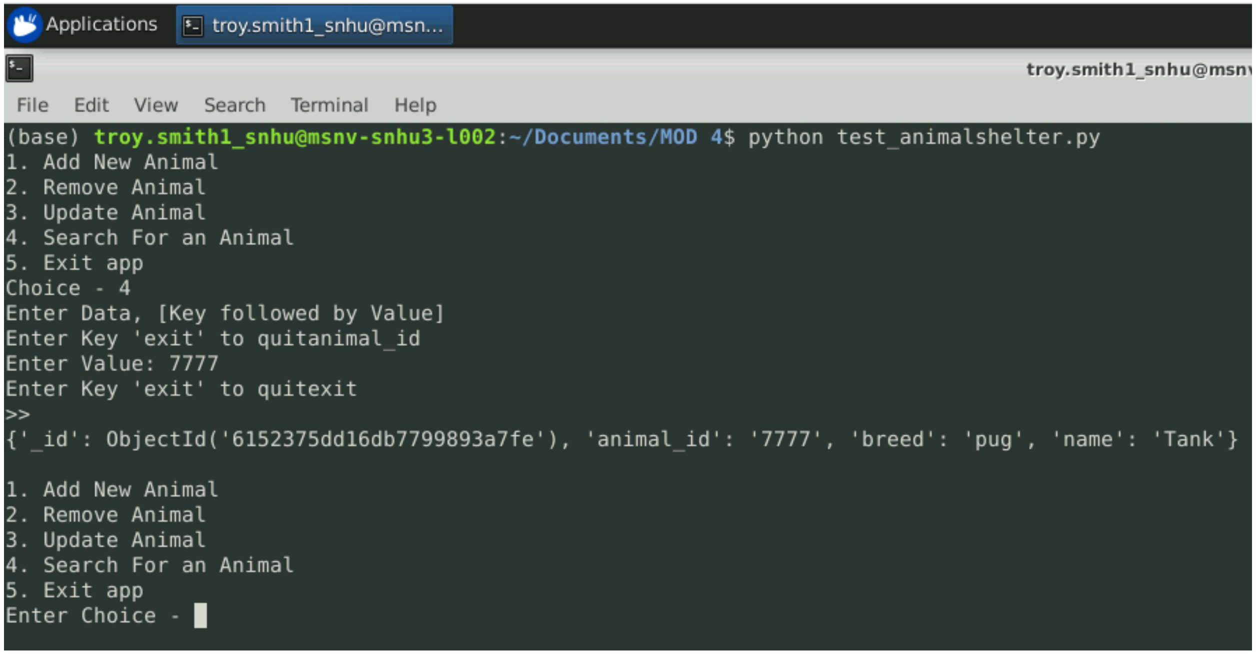
```
(base) troy.smith1_snhu@msnv-snhu3-l002:~$ ls
Desktop  Documents  Downloads  mongodb  mongod.log  Music  Pictures  repos  show
(base) troy.smith1_snhu@msnv-snhu3-l002:~$ cd Documents/
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents$ ls
'MOD 4'   Project_One
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents$ cd 'MOD 4'/
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents/MOD 4$ ls
animalshelter.py  __pycache__  SP  test_animalshelter.py
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents/MOD 4$ python test_animalshelter.py
1. Add New Animal
2. Remove Animal
3. Update Animal
4. Search For an Animal
5. Exit app
Choice - 1
Enter Data, [Key followed by Value]
Key:animal_id
Enter Value: 7777
Key:breed
Enter Value: pug
Key:exit
----------------------- Inserting New Animal ------------------
-------------- Done -------------------------------------------
1. Add New Animal
2. Remove Animal
3. Update Animal
4. Search For an Animal
5. Exit app
Enter Choice - 
```

Update an item (Update)



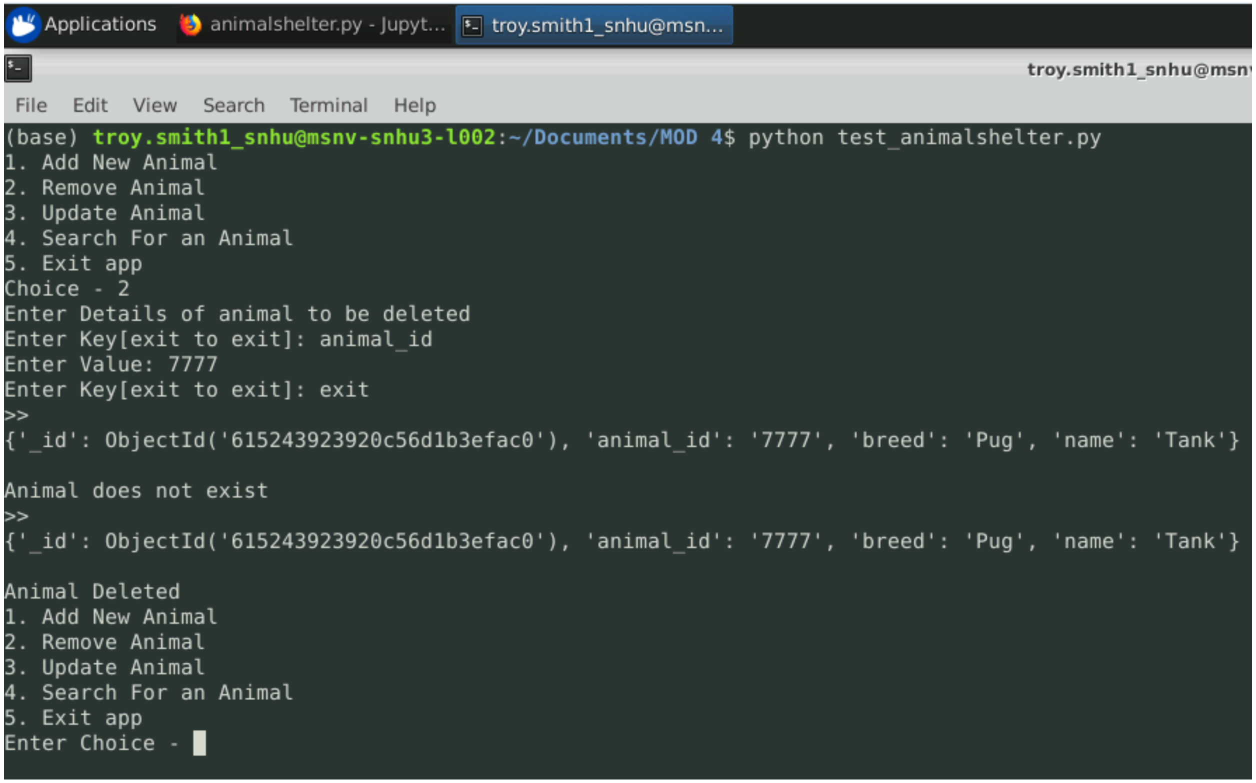
```
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents/MOD 4$ python test_animalshelter.py
1. Add New Animal
2. Remove Animal
3. Update Animal
4. Search For an Animal
5. Exit app
Choice - 3
Enter Specifications of animal to update:
Enter Key: animal_id
Enter Value: 7777
Enter Key: exit
Enter New  Values
Enter Key: name
Enter Value: Tank
Enter Key: exit
>>
{'_id': ObjectId('6152375dd16db7799893a7fe'), 'animal_id': '7777', 'breed': 'pug', 'name': 'Tank'}

1. Add New Animal
2. Remove Animal
3. Update Animal
4. Search For an Animal
5. Exit app
Enter Choice - 
```

Search for an Item (Read)



```
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents/MOD 4$ python test_animalshelter.py
1. Add New Animal
2. Remove Animal
3. Update Animal
4. Search For an Animal
5. Exit app
Choice - 4
Enter Data, [Key followed by Value]
Enter Key 'exit' to quitanimal_id
Enter Value: 7777
Enter Key 'exit' to quitexit
>>
{'_id': ObjectId('6152375dd16db7799893a7fe'), 'animal_id': '7777', 'breed': 'pug', 'name': 'Tank'}

1. Add New Animal
2. Remove Animal
3. Update Animal
4. Search For an Animal
5. Exit app
Enter Choice -
```

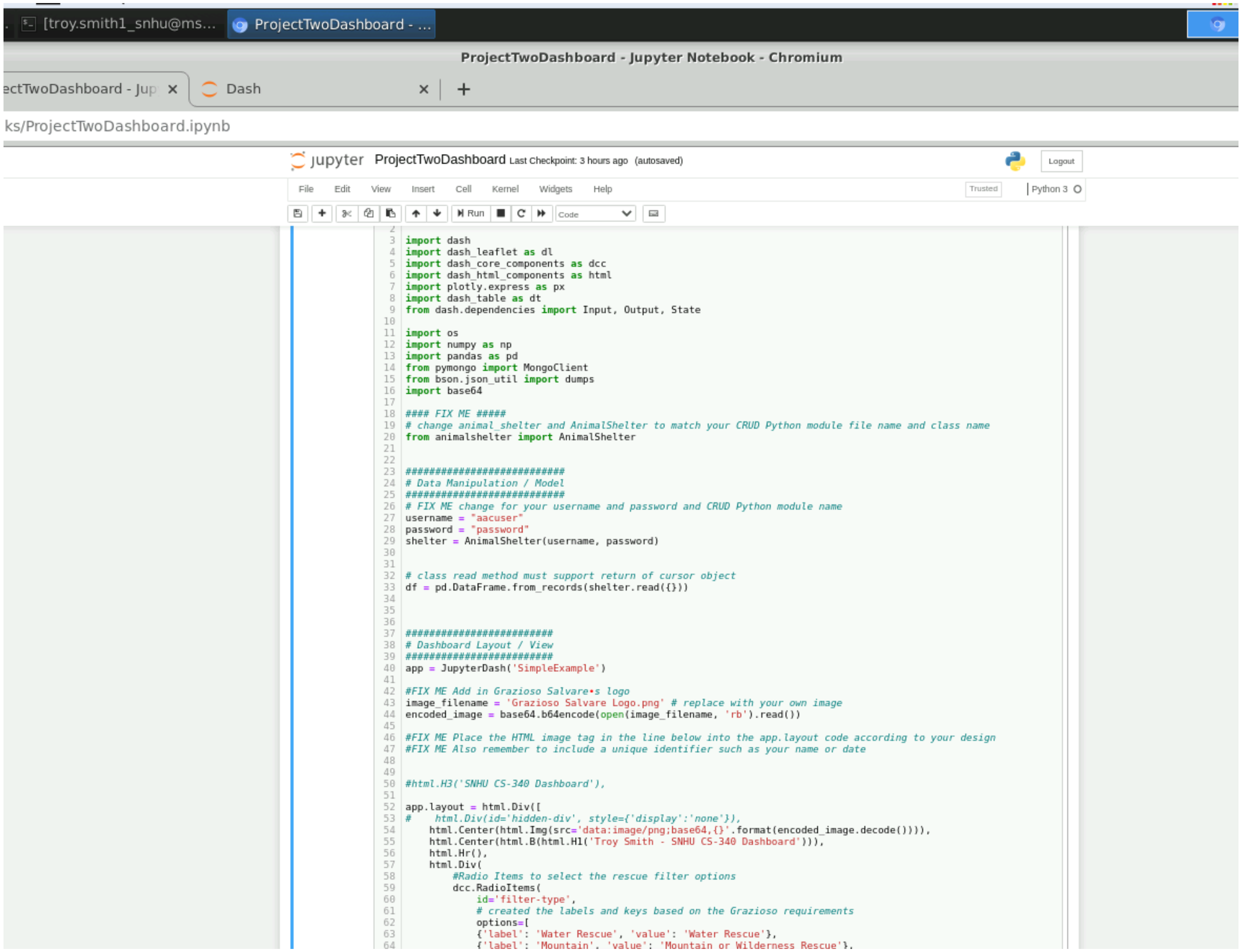
Remove an item (Delete)



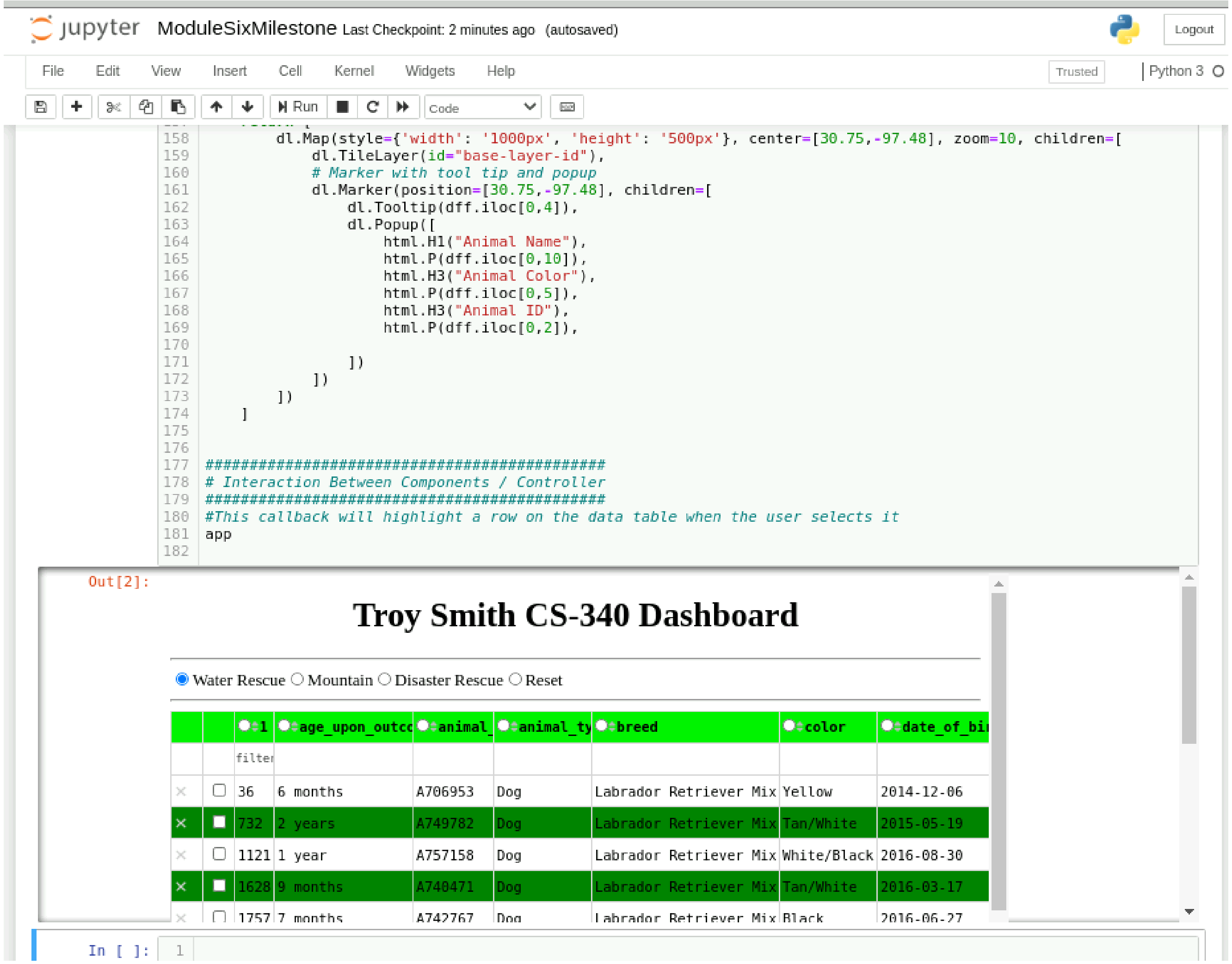
```
(base) troy.smith1_snhu@msnv-snhu3-l002:~/Documents/MOD 4$ python test_animalshelter.py
1. Add New Animal
2. Remove Animal
3. Update Animal
4. Search For an Animal
5. Exit app
Choice - 2
Enter Details of animal to be deleted
Enter Key[exit to exit]: animal_id
Enter Value: 7777
Enter Key[exit to exit]: exit
>>
{'_id': ObjectId('615243923920c56d1b3efac0'), 'animal_id': '7777', 'breed': 'Pug', 'name': 'Tank'}

Animal does not exist
>>
{'_id': ObjectId('615243923920c56d1b3efac0'), 'animal_id': '7777', 'breed': 'Pug', 'name': 'Tank'}

Animal Deleted
1. Add New Animal
2. Remove Animal
3. Update Animal
4. Search For an Animal
5. Exit app
Enter Choice -
```

7-1 Final Project– TROY SMITH

Here are examples of data visualization using the dashboard code found within the Project Two Dashboard file.


[troy.smith1_snhu@ms...
ProjectTwoDashboard - ...
ProjectTwoDashboard - Jupyter Notebook - Chromium
ectTwoDashboard - Jup
Dash
ks/ProjectTwoDashboard.ipynb
jupyter
ProjectTwoDashboard Last Checkpoint: 3 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
import dash
import dash_leaflet as dl
import dash_core_components as dcc
import dash_html_components as html
import plotly.express as px
import dash_table as dt
from dash.dependencies import Input, Output, State

import os
import numpy as np
import pandas as pd
from pymongo import MongoClient
from bson.json_util import dumps
import base64

#### FIX ME #####
# change animal_shelter and AnimalShelter to match your CRUD Python module file name and class name
from animalshelter import AnimalShelter


###########################
# Data Manipulation / Model
###########################
# FIX ME change for your username and password and CRUD Python module name
username = "aacuser"
password = "password"
shelter = AnimalShelter(username, password)


# class read method must support return of cursor object
df = pd.DataFrame.from_records(shelter.read({}))



#########################
# Dashboard Layout / View
#########################
app = JupyterDash('SimpleExample')

#FIX ME Add in Grazioso Salvare•s logo
image_filename = 'Grazioso Salvare Logo.png' # replace with your own image
encoded_image = base64.b64encode(open(image_filename, 'rb').read())

#FIX ME Place the HTML image tag in the line below into the app.layout code according to your design
#FIX ME Also remember to include a unique identifier such as your name or date


#html.H3('SNHU CS-340 Dashboard'),

app.layout = html.Div([
#    html.Div(id='hidden-div', style={'display':'none'}),
    html.Center(html.Img(src='data:image/png;base64,{}'.format(encoded_image.decode()))),
    html.Center(html.B(html.H1('Troy Smith - SNHU CS-340 Dashboard'))),
    html.Hr(),
    html.Div(
        #Radio Items to select the rescue filter options
        dcc.RadioItems(
            id='filter-type',
            # created the labels and keys based on the Grazioso requirements
            options=[
            {'label': 'Water Rescue', 'value': 'Water Rescue'},
            {'label': 'Mountain', 'value': 'Mountain or Wilderness Rescue'},
```



```
        dl.Map(style={'width': '1000px', 'height': '500px'}, center=[30.75,-97.48], zoom=10, children=[
            dl.TileLayer(id="base-layer-id"),
            # Marker with tool tip and popup
            dl.Marker(position=[30.75,-97.48], children=[
                dl.Tooltip(dff.iloc[0,4]),
                dl.Popup([
                    html.H1("Animal Name"),
                    html.P(dff.iloc[0,10]),
                    html.H3("Animal Color"),
                    html.P(dff.iloc[0,5]),
                    html.H3("Animal ID"),
                    html.P(dff.iloc[0,2]),

                ])
            ])
        ])
    ]


#############################################
# Interaction Between Components / Controller
#############################################
#This callback will highlight a row on the data table when the user selects it
app
```

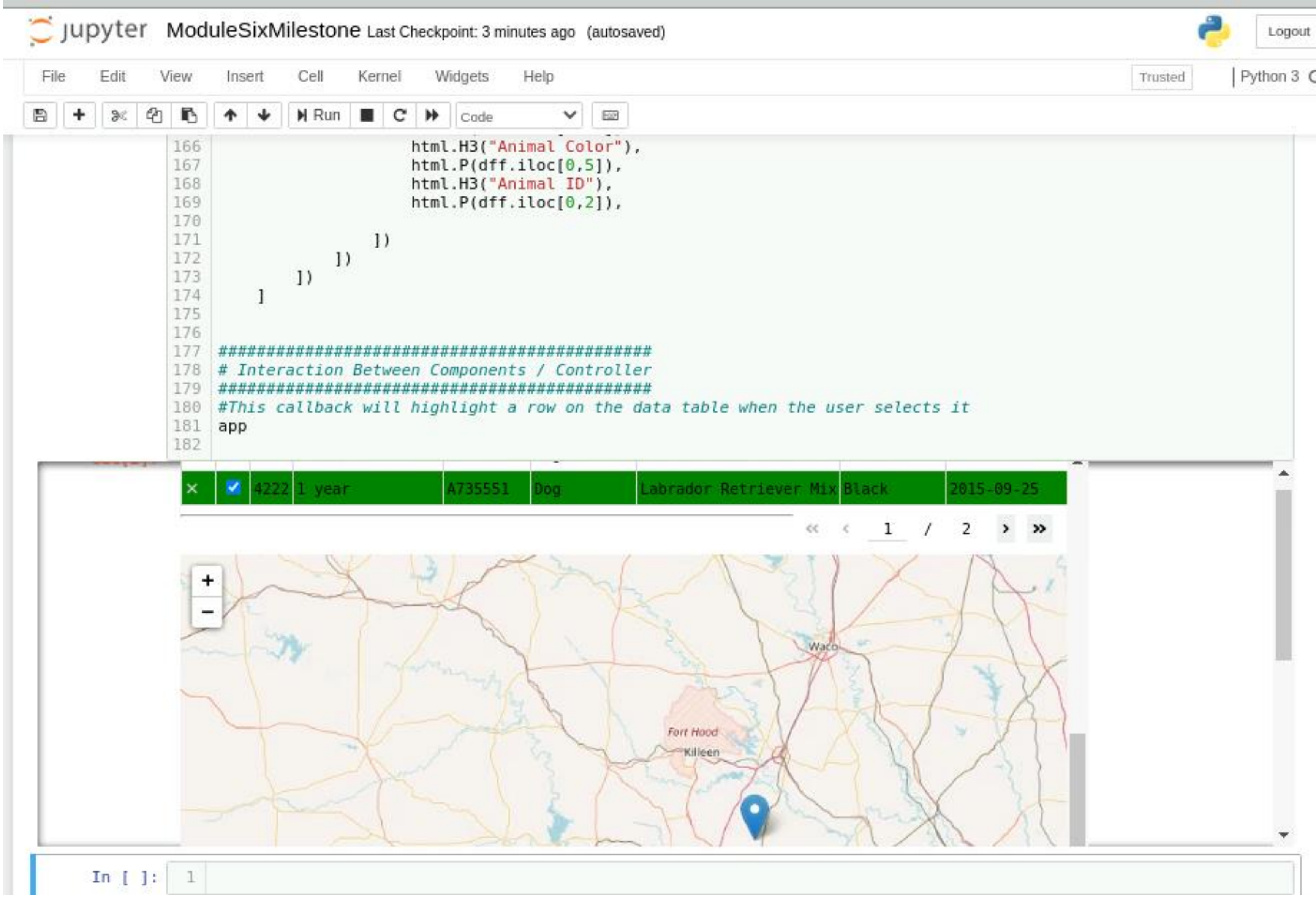
Out[2]:

Troy Smith CS-340 Dashboard

● Water Rescue ○ Mountain ○ Disaster Rescue ○ Reset

		○1	○age_upon_outco	○animal_	○animal_ty	○breed	○color	○date_of_bi
		filter						
×	□	36	6 months	A706953	Dog	Labrador Retriever Mix	Yellow	2014-12-06
×	□	732	2 years	A749782	Dog	Labrador Retriever Mix	Tan/White	2015-05-19
×	□	1121	1 year	A757158	Dog	Labrador Retriever Mix	White/Black	2016-08-30
×	□	1628	9 months	A740471	Dog	Labrador Retriever Mix	Tan/White	2016-03-17
×	□	1757	7 months	A742767	Dog	Labrador Retriever Mix	Black	2016-06-27

In [ ]:

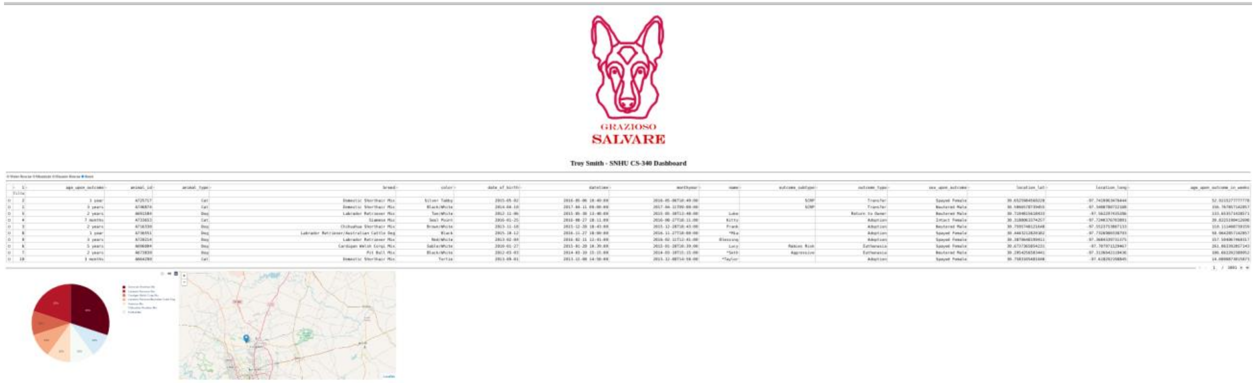


```
                    html.H3("Animal Color"),
                    html.P(dff.iloc[0,5]),
                    html.H3("Animal ID"),
                    html.P(dff.iloc[0,2]),

                ])
            ])
        ])
    ]


##################################################
# Interaction Between Components / Controller
##################################################
#This callback will highlight a row on the data table when the user selects it
app
```

✕	☑	4222	1 year	A735551	Dog	Labrador Retriever Mix	Black	2015-09-25

« ‹ 1 / 2 › »

In [ ]:

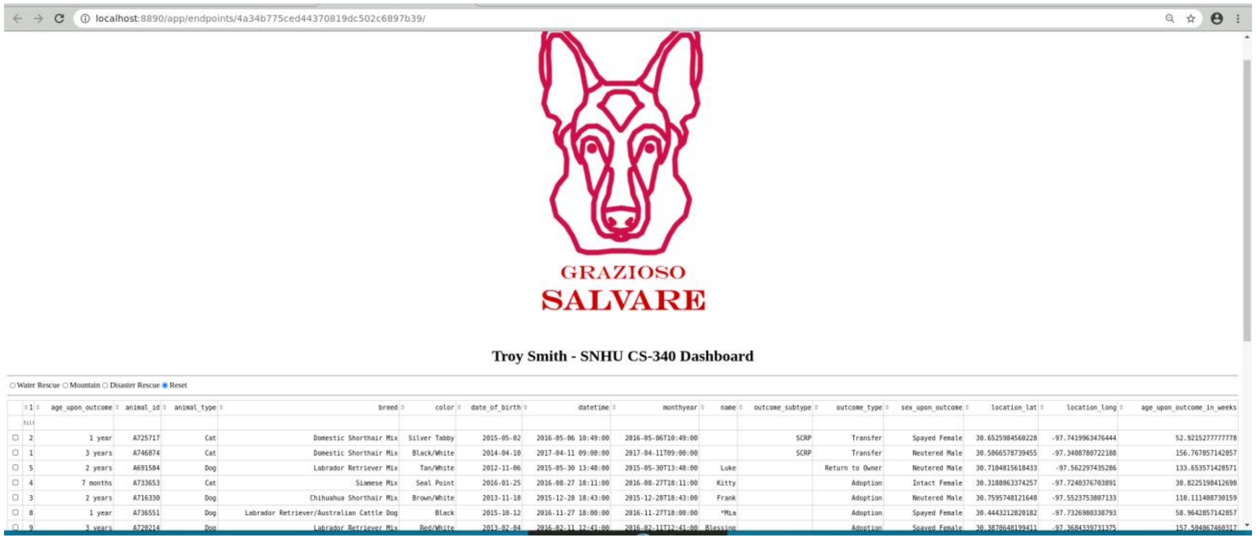


GRAZIOSO SALVARE

Troy Smith - SNHU CS-340 Dashboard

○ Water Rescue ○ Mountain ○ Disaster Rescue ● Reset

	1	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_weeks
	filte															
□	2	1 year	A725717	Cat	Domestic Shorthair Mix	Silver Tabby	2015-05-02	2016-05-06 10:49:00	2016-05-06T10:49:00		SCRP	Transfer	Spayed Female	30.6525984560228	-97.7419963476444	52.9215277777778
□	1	3 years	A746874	Cat	Domestic Shorthair Mix	Black/White	2014-04-10	2017-04-11 09:00:00	2017-04-11T09:00:00		SCRP	Transfer	Neutered Male	30.5066578739455	-97.3408780722188	156.767857142857
□	5	2 years	A691584	Dog	Labrador Retriever Mix	Tan/White	2012-11-06	2015-05-30 13:48:00	2015-05-30T13:48:00	Luke		Return to Owner	Neutered Male	30.7104815618433	-97.562297435286	133.653571428571
□	4	7 months	A733653	Cat	Siamese Mix	Seal Point	2016-01-25	2016-08-27 18:11:00	2016-08-27T18:11:00	Kitty		Adoption	Intact Female	30.3188063374257	-97.7240376703891	30.8225198412698
□	3	2 years	A716330	Dog	Chihuahua Shorthair Mix	Brown/White	2013-11-18	2015-12-28 18:43:00	2015-12-28T18:43:00	Frank		Adoption	Neutered Male	30.7595748121648	-97.5523753807133	110.111408730159
□	8	1 year	A736551	Dog	Labrador Retriever/Australian Cattle Dog	Black	2015-10-12	2016-11-27 18:00:00	2016-11-27T18:00:00	*Mia		Adoption	Spayed Female	30.4443212820182	-97.7326980338793	58.9642857142857
□	9	3 years	A720214	Dog	Labrador Retriever Mix	Red/White	2013-02-04	2016-02-11 12:41:00	2016-02-11T12:41:00	Blessing		Adoption	Spayed Female	30.3878648199411	-97.3684339731375	157.504067460317
□	6	5 years	[illegible]	Dog	Cardigan Welsh Corgi Mix	Sable/White	[illegible]	[illegible]	[illegible]	Lucy	Rabies Risk	Euthanasia	Spayed Female	[illegible]	[illegible]	[illegible]
□	7	2 years	[illegible]	Dog	Pit Bull Mix	Black/White	[illegible]	[illegible]	[illegible]	*Seth	Aggressive	Euthanasia	Neutered Male	[illegible]	[illegible]	[illegible]
□	10	3 months	[illegible]	Cat	Domestic Shorthair Mix	Tortie	[illegible]	[illegible]	[illegible]	*Taylor		Adoption	Spayed Female	[illegible]	[illegible]	[illegible]



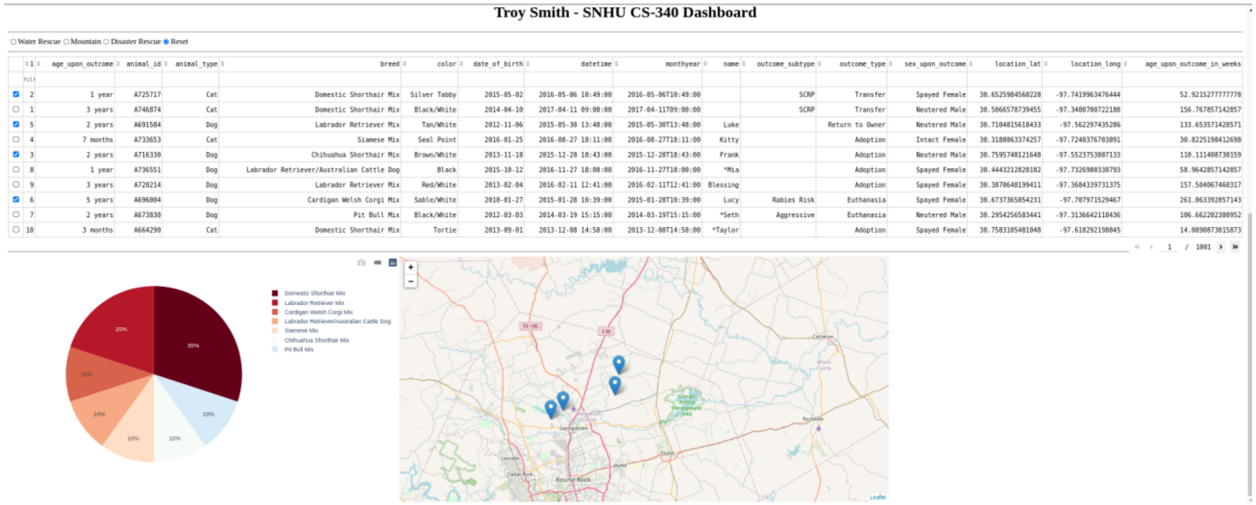
localhost:8890/app/endpoints/4a34b775ced44370819dc502c6897b39/

GRAZIOSO SALVARE

Troy Smith - SNHU CS-340 Dashboard

○ Water Rescue ○ Mountain ○ Disaster Rescue ● Reset

	1	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_weeks
	fil															
□	2	1 year	A725717	Cat	Domestic Shorthair Mix	Silver Tabby	2015-05-02	2016-05-06 10:49:00	2016-05-06T10:49:00		SCRP	Transfer	Spayed Female	30.6525984560228	-97.7419963476444	52.9215277777778
□	1	3 years	A746874	Cat	Domestic Shorthair Mix	Black/White	2014-04-10	2017-04-11 09:00:00	2017-04-11T09:00:00		SCRP	Transfer	Neutered Male	30.5066578739455	-97.3408780722188	156.767857142857
□	5	2 years	A691584	Dog	Labrador Retriever Mix	Tan/White	2012-11-06	2015-05-30 13:48:00	2015-05-30T13:48:00	Luke		Return to Owner	Neutered Male	30.7104815618433	-97.562297435286	133.653571428571
□	4	7 months	A733653	Cat	Siamese Mix	Seal Point	2016-01-25	2016-08-27 18:11:00	2016-08-27T18:11:00	Kitty		Adoption	Intact Female	30.3188063374257	-97.7240376703891	30.8225198412698
□	3	2 years	A716330	Dog	Chihuahua Shorthair Mix	Brown/White	2013-11-18	2015-12-28 18:43:00	2015-12-28T18:43:00	Frank		Adoption	Neutered Male	30.7595748121648	-97.5523753807133	110.111408730159
□	8	1 year	A736551	Dog	Labrador Retriever/Australian Cattle Dog	Black	2015-10-12	2016-11-27 18:00:00	2016-11-27T18:00:00	*Mia		Adoption	Spayed Female	30.4443212820182	-97.7326980338793	58.9642857142857
□	9	3 years	A720214	Dog	Labrador Retriever Mix	Red/White	2013-02-04	2016-02-11 12:41:00	2016-02-11T12:41:00	Blessing		Adoption	Spayed Female	30.3878648199411	-97.3684339731375	157.504067460317

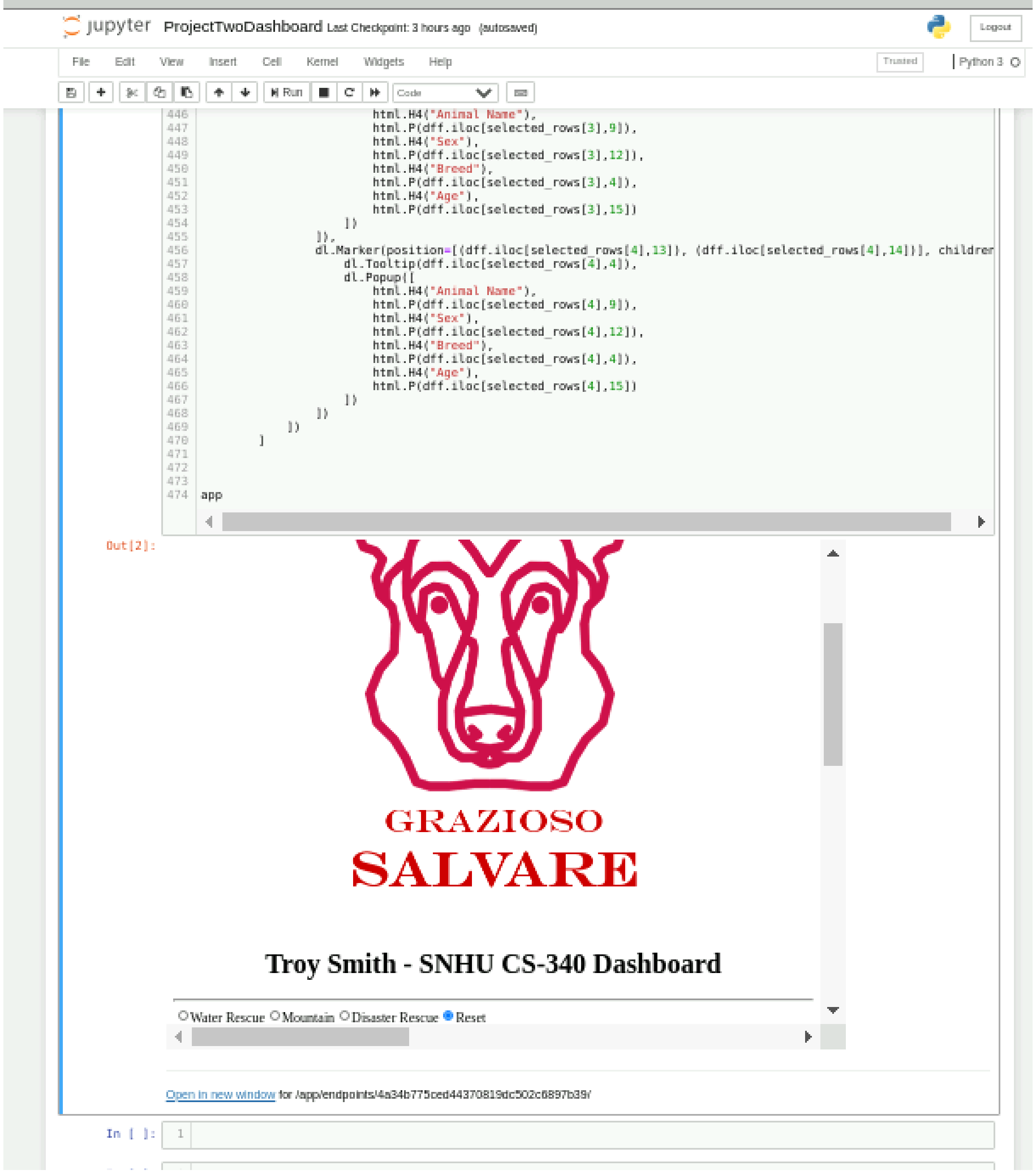


Troy Smith - SNHU CS-340 Dashboard

○ Water Rescue ○ Mountain ○ Disaster Rescue ● Reset

	1	age_upon_outcome	animal_id	animal_type	breed	color	date_of_birth	datetime	monthyear	name	outcome_subtype	outcome_type	sex_upon_outcome	location_lat	location_long	age_upon_outcome_in_weeks
☑	2	1 year	A725717	Cat	Domestic Shorthair Mix	Silver Tabby	2015-05-02	2016-05-06 10:49:00	2016-05-06T10:49:00		SCRP	Transfer	Spayed Female	30.6525984560228	-97.7419963476444	52.9215277777778
□	1	3 years	A746874	Cat	Domestic Shorthair Mix	Black/White	2014-04-10	2017-04-11 09:00:00	2017-04-11T09:00:00		SCRP	Transfer	Neutered Male	30.5066578739455	-97.3408780722188	156.767857142857
☑	5	2 years	A691584	Dog	Labrador Retriever Mix	Tan/White	2012-11-06	2015-05-30 13:48:00	2015-05-30T13:48:00	Luke		Return to Owner	Neutered Male	30.7104815618433	-97.562297435286	133.653571428571
□	4	7 months	A733653	Cat	Siamese Mix	Seal Point	2016-01-25	2016-08-27 18:11:00	2016-08-27T18:11:00	Kitty		Adoption	Intact Female	30.3188063374257	-97.7240376703891	30.8225198412698
☑	3	2 years	A716330	Dog	Chihuahua Shorthair Mix	Brown/White	2013-11-18	2015-12-28 18:43:00	2015-12-28T18:43:00	Frank		Adoption	Neutered Male	30.7595748121648	-97.5523753807133	110.111408730159
□	8	1 year	A736551	Dog	Labrador Retriever/Australian Cattle Dog	Black	2015-10-12	2016-11-27 18:00:00	2016-11-27T18:00:00	*Mia		Adoption	Spayed Female	30.4443212820182	-97.7326988338793	58.9642857142857
□	9	3 years	A720214	Dog	Labrador Retriever Mix	Red/White	2013-02-04	2016-02-11 12:41:00	2016-02-11T12:41:00	Blessing		Adoption	Spayed Female	30.3870648199411	-97.3684339731375	157.504067460317
☑	6	5 years	A696004	Dog	Cardigan Welsh Corgi Mix	Sable/White	2010-01-27	2015-01-28 10:39:00	2015-01-28T10:39:00	Lucy	Rabies Risk	Euthanasia	Spayed Female	30.6737365854231	-97.707971529467	261.063392857143
□	7	2 years	A673830	Dog	Pit Bull Mix	Black/White	2012-03-03	2014-03-19 15:15:00	2014-03-19T15:15:00	*Seth	Aggressive	Euthanasia	Neutered Male	30.2954256583441	-97.3136642110436	106.662202380952
□	10	3 months	A664290	Cat	Domestic Shorthair Mix	Tortie	2013-09-01	2013-12-08 14:58:00	2013-12-08T14:58:00	*Taylor		Adoption	Spayed Female	30.7583105481048	-97.618292198845	14.0890873015873

« ‹ 1 / 1001 › »



```
                        html.H4("Animal Name"),
                        html.P(dff.iloc[selected_rows[3],9]),
                        html.H4("Sex"),
                        html.P(dff.iloc[selected_rows[3],12]),
                        html.H4("Breed"),
                        html.P(dff.iloc[selected_rows[3],4]),
                        html.H4("Age"),
                        html.P(dff.iloc[selected_rows[3],15])
                    ])
                ]),
                dl.Marker(position=[(dff.iloc[selected_rows[4],13]), (dff.iloc[selected_rows[4],14])], childrer
                    dl.Tooltip(dff.iloc[selected_rows[4],4]),
                    dl.Popup([
                        html.H4("Animal Name"),
                        html.P(dff.iloc[selected_rows[4],9]),
                        html.H4("Sex"),
                        html.P(dff.iloc[selected_rows[4],12]),
                        html.H4("Breed"),
                        html.P(dff.iloc[selected_rows[4],4]),
                        html.H4("Age"),
                        html.P(dff.iloc[selected_rows[4],15])
                    ])
                ])
            ])
        ]



app
```

Out[2]:

Troy Smith - SNHU CS-340 Dashboard

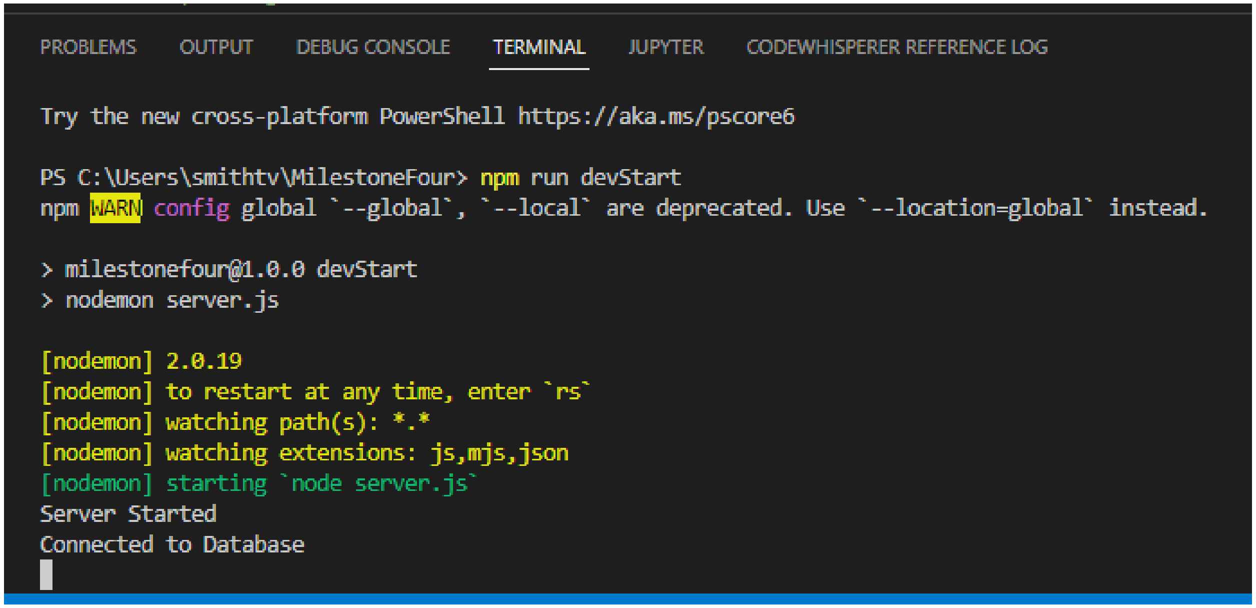
○ Water Rescue ○ Mountain ○ Disaster Rescue ● Reset

Open in new window for /app/endpoints/4a34b775ced44370819dc502c6897b39/

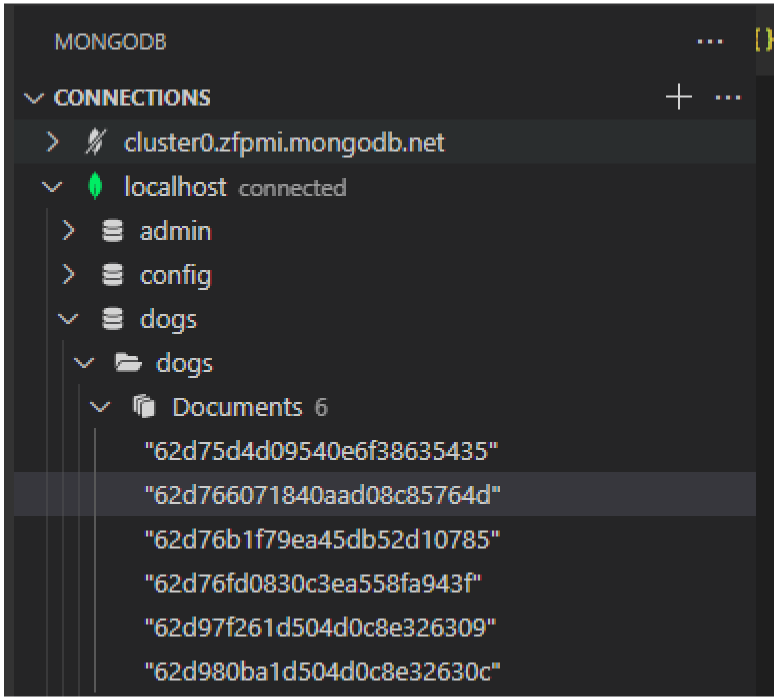
In [ ]:

For the enhancement portion of this project, I chose to create a RESTful API that creates and accesses a Mongo DB. REST stands for Representation, State, Transfer or basically the same as CRUD. I utilized the features of the Visual Studio Code IDE to construct and manipulate the database hosted on the local device using port 3000.

Here is an example of the successful connection to the database.



```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\smithtv\MilestoneFour> npm run devStart
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

> milestonefour@1.0.0 devStart
> nodemon server.js

[nodemon] 2.0.19
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
Server Started
Connected to Database
```



MONGODB

CONNECTIONS

cluster0.zfpmi.mongodb.net

localhost connected

admin

config

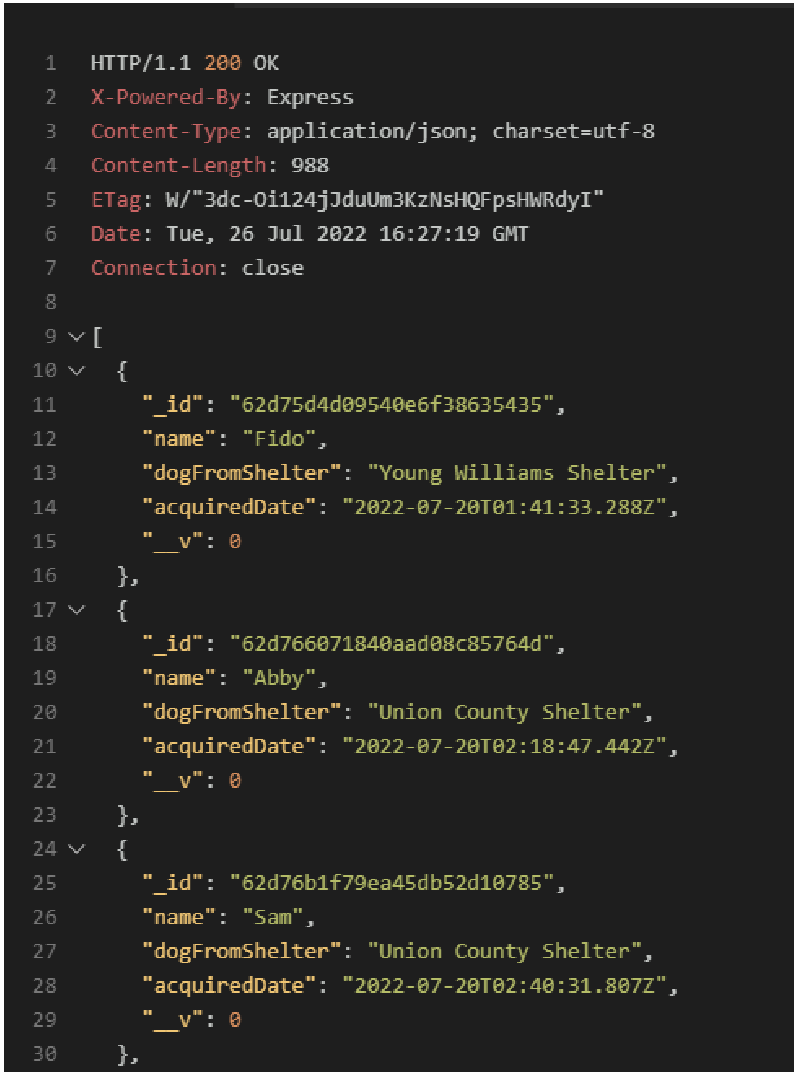
dogs

dogs

Documents 6

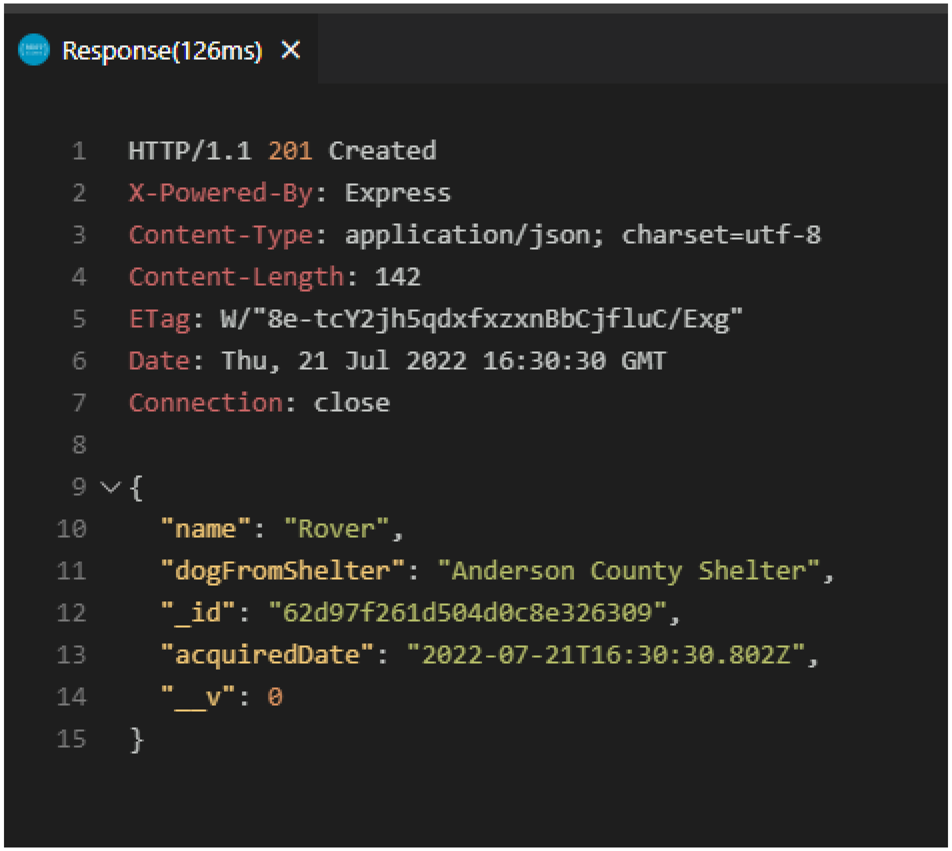
```
"62d75d4d09540e6f38635435"
"62d766071840aad08c85764d"
"62d76b1f79ea45db52d10785"
"62d76fd0830c3ea558fa943f"
"62d97f261d504d0c8e326309"
"62d980ba1d504d0c8e32630c"
```

Accessing items within the database.



```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 988
ETag: W/"3dc-Oi124jJduUm3KzNsHQFpsHWRdyI"
Date: Tue, 26 Jul 2022 16:27:19 GMT
Connection: close

[
  {
    "_id": "62d75d4d09540e6f38635435",
    "name": "Fido",
    "dogFromShelter": "Young Williams Shelter",
    "acquiredDate": "2022-07-20T01:41:33.288Z",
    "__v": 0
  },
  {
    "_id": "62d766071840aad08c85764d",
    "name": "Abby",
    "dogFromShelter": "Union County Shelter",
    "acquiredDate": "2022-07-20T02:18:47.442Z",
    "__v": 0
  },
  {
    "_id": "62d76b1f79ea45db52d10785",
    "name": "Sam",
    "dogFromShelter": "Union County Shelter",
    "acquiredDate": "2022-07-20T02:40:31.807Z",
    "__v": 0
  },
```

Creating one within the database.



```
HTTP/1.1 201 Created
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 142
ETag: W/"8e-tcY2jh5qdxfxzxnBbCjfluC/Exg"
Date: Thu, 21 Jul 2022 16:30:30 GMT
Connection: close

{
  "name": "Rover",
  "dogFromShelter": "Anderson County Shelter",
  "_id": "62d97f261d504d0c8e326309",
  "acquiredDate": "2022-07-21T16:30:30.802Z",
  "__v": 0
}
```

This project met me course objective from Module One by displaying my skills and abilities to work with and within databases. I consider it a valuable inclusion in my portfolio and learned more than I anticipated during its construction. . Inclusion of this database artifact in my portfolio demonstrates my ability to do just that and prove my worth and skills within database work and my ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals. I accomplished in both the original artifact and the enhanced artifact by developing a CRUD mongo DB allows users to create, read, update and delete items within collections contained within the database models. The original artifact also uses authentication methods to develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources.

This was my first time working with the REST API and was impressed with the Visual Studio Code's ability to intuitively install necessary and helpful extensions at will. It included the REST client that allowed me to run my .rest commands in a real-time mode to see the results immediately and included all the necessary components to construct my code.

There weren't as many challenges as I first anticipated, but one in particular was coding the server file correctly and getting the expected "Server Started" and "Connected to Database" prompt. Thankfully after reviewing several YouTube videos and searching Stack Overflow posts, I successfully started the server and proceeded with the other code.

In the end, I learned much more about REST and how it operates within the VS Code IDE. I consider the VS Code to be my new favorite IDE!

Thank you for taking the time to view my Portfolio!