Project one

SNHU | CS-340-T4528

Client/Server Development

Ethan Daugherty  
4/3/2022

2022

**Project One**

1. Upload the Austin Animal Center Outcomes data set into MongoDB by **inserting a CSV file using the appropriate MongoDB import tool**. The data set is located in the Supporting Materials section. Complete the import using the mongoimport tool and **take screenshots** of both the import command and its execution. These screenshots will later be included in your README file.  
     
   Note: If you completed the Module Three Milestone, you have already completed this step. Be sure to include your screenshots from the Module Three Milestone in your README file.  
   Graphical user interface, text, application

   Description automatically generated
2. Create an administrator account and a user account in the mongo shell to **ensure user authentication to the database and collection** that was created. Be sure to take a screenshot of the mongo shell execution command screen that shows your login process with both accounts. This screenshot will later be included in your README file.  
     
   Note: If you completed the Module Three Milestone, you have already completed this step. Be sure to include your screenshots from the Module Three Milestone in your README file.  
   Text

   Description automatically generated  
   Graphical user interface, text

   Description automatically generated  
   Text

   Description automatically generated  
   Text

   Description automatically generated
3. Next, you must develop a Python module in a PY file, using object-oriented programming methodology, to enable CRUD functionality for the database. To support code reusability, your Python code needs to be importable as a module by other Python scripts.  
     
   **Develop** a CRUD class that, when instantiated, provides the following functionality:

* **A Create method that inserts a document into a specified MongoDB database and collection**
  + Input -> argument to function will be a set of key/value pairs in the data type acceptable to the MongoDB driver insert API call.
  + Return -> “True” if successful insert, else “False”.

Graphical user interface, text, application, email

Description automatically generated

* **A Read method that queries for document(s) from a specified MongoDB database and specified collection**
  + Input -> arguments to function should be the key/value lookup pair to use with the MongoDB driver find API call.
  + Return -> result in cursor if successful, else MongoDB returned error message.

Graphical user interface, text, application, email

Description automatically generated

* **An Update method that queries for and changes document(s) from a specified MongoDB database and specified collection**
  + Input -> arguments to function should be the key/value lookup pair to use with the MongoDB driver find API call. Last argument to function will be a set of key/value pairs in the data type acceptable to the MongoDB driver insert API call.
  + Return -> result in JSON format if successful, else MongoDB returned error message.

Graphical user interface, text

Description automatically generated

* **A Delete method that queries for and removes document(s) from a specified MongoDB database and specified collection**
  + Input -> arguments to function should be the key/value lookup pair to use with the MongoDB driver find API call.
  + Return -> result in JSON format if successful, else MongoDB returned error message.

Graphical user interface, text

Description automatically generatedAs you develop your code, be sure to **use industry standard best practices** such as proper naming conventions, exception handling, and in-line comments. This will ensure that your code is easy to read and reusable for future projects. Refer to the Python Style Guide, located in the Supporting Materials section, to help with these industry standard best practices.  
  
Note: If you completed the Module Four Milestone, you have already developed the Create and Read functionality.

1. Finally, you must test your Python module to make sure that it works. To do this, **create a Python script that imports your CRUD Python module to call and test all instances of CRUD functionality**. This script should be created in a separate Jupyter Notebook (IPYNB) file, and should import and instantiate an object from your CRUD library to effect changes in MongoDB. Be sure to use the username and password for the “aacuser” account for authentication when instantiating the class. After creating your script, execute it in Jupyter Notebook and take screenshots of the commands and their execution. These screenshots will later be included in your README file.  
     
   Note: If you completed the Module Four Milestone, you have already begun this work. Expand your script to call and test the Update and Delete functionality.

Authentication:

Text

Description automatically generated

New entry creation with boolean outcome:  
Text, letter

Description automatically generated

Invalid creation script with outcome:

Text

Description automatically generated

Query to locate my created animal:  
Text, letter

Description automatically generated

In valid query to search for animal:  
Graphical user interface, text, application, email

Description automatically generated  
  
Animal Documentation update:  
Graphical user interface

Description automatically generated  
Text, letter

Description automatically generated  
Invalid documentation update:  
Graphical user interface, text, application, email

Description automatically generated  
Valid documentation deletion:  
Graphical user interface, text, application, chat or text message

Description automatically generated

Invalid deletion:  
Text

Description automatically generated