Janice Ainembabazi

Southern New Hampshire University

CS-340-18683-M01 Client/Server Development

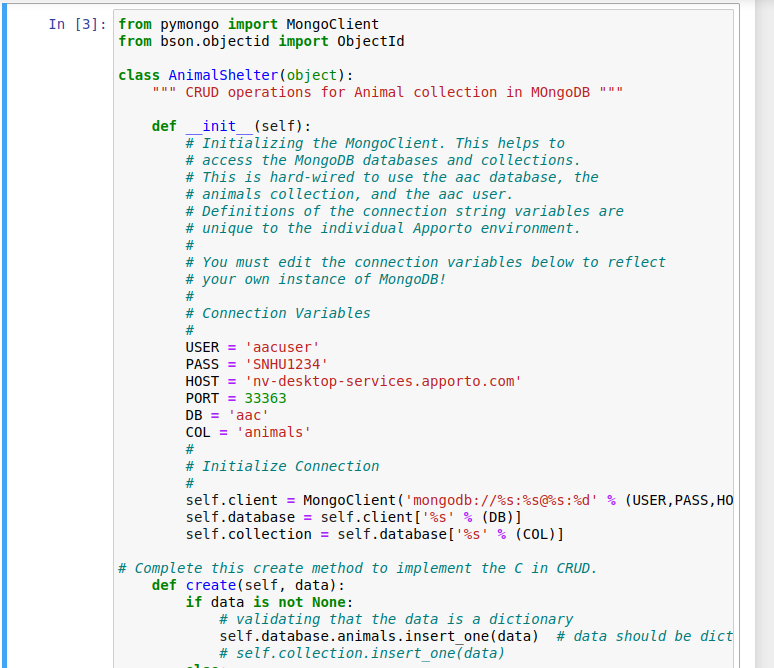
Prof. Jeff Sanford

November 24th 2024

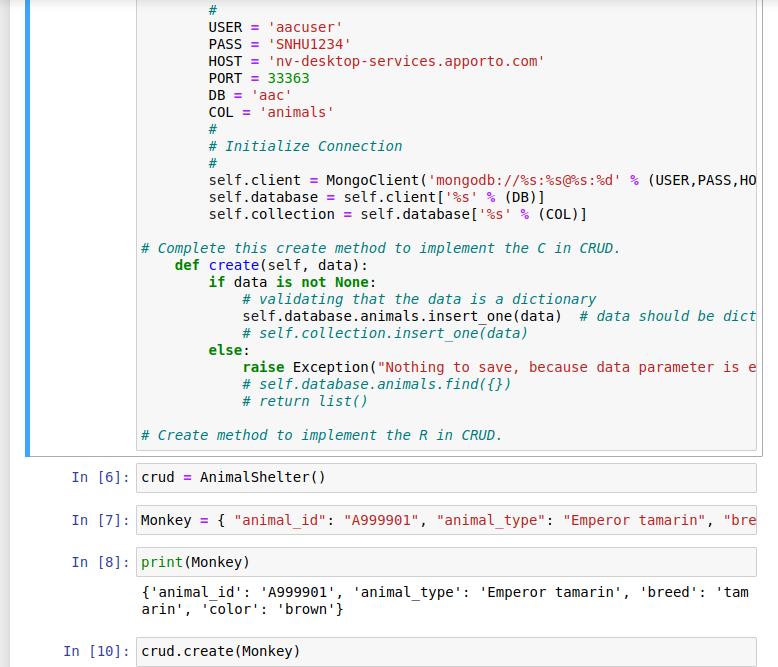
To implement a Python class that handles CRUD (Create, Read, Update, and Delete) operations for managing animal data in a MongoDB databas, I started by creating the AnimalShelter class, which includes methods for both creating and reading animal records in MongoDB. To do this, I used the pymongo library to connect to the database and perform the operations.

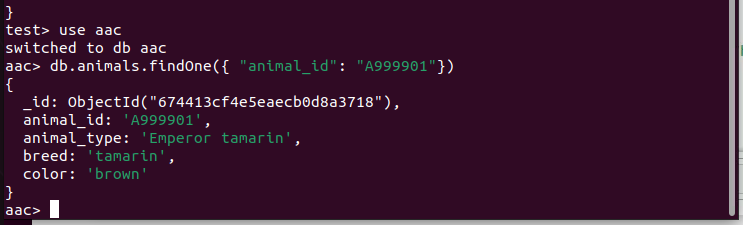
The first step I took was to write the \_\_init\_\_ method, which initializes the connection to the MongoDB database. I used the credentials for the aacuser account to authenticate and access the database. The connection string included the username, password, host, port, and database name to connect to the aac database and the animals collection.

Next, I implemented the create method. This method allows me to insert a new animal record into the database. I created a sample animal record as a dictionary, which includes information like the animal’s ID, name, species, age, and adoption details. When I called this method, it inserted the data into the MongoDB database.



For the read functionality, I implemented a read method that queries the database for documents matching a given query. I used this method to find all animals of a specific species. For testing, I searched for animals of the species "Monkey." When I ran this query, it returned all the matching records from the database, which I then printed to the screen.



:

By running this script in a Jupyter Notebook, I was able to test the functionality of both the create and read methods. The create method inserted a new animal record successfully, and the read method retrieved the animal records that matched my query. This allowed me to confirm that the database operations were working correctly.



