Owen Lindsey

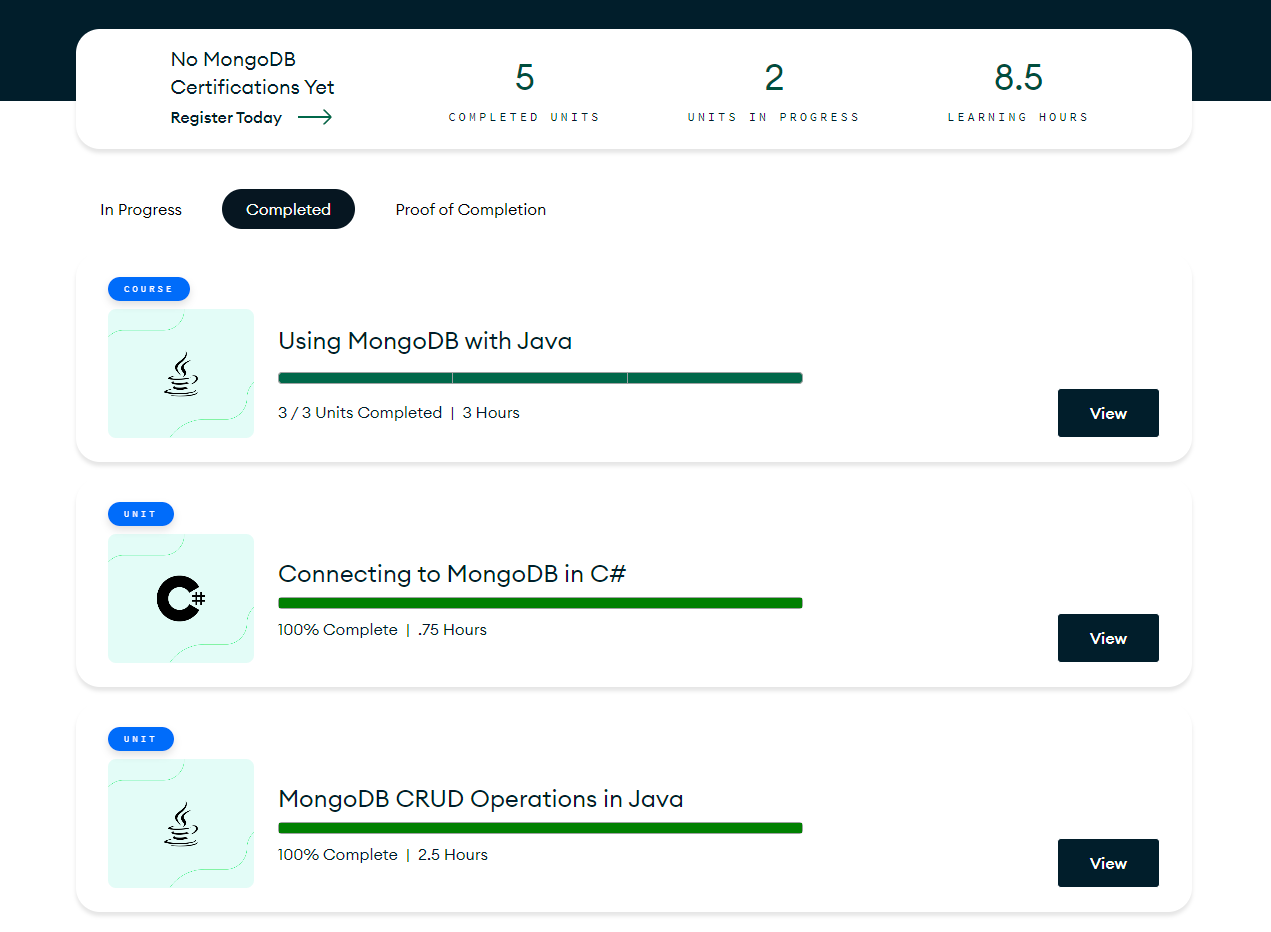
Professor Sluiter

CST-345

Activity 7

12/3/2023

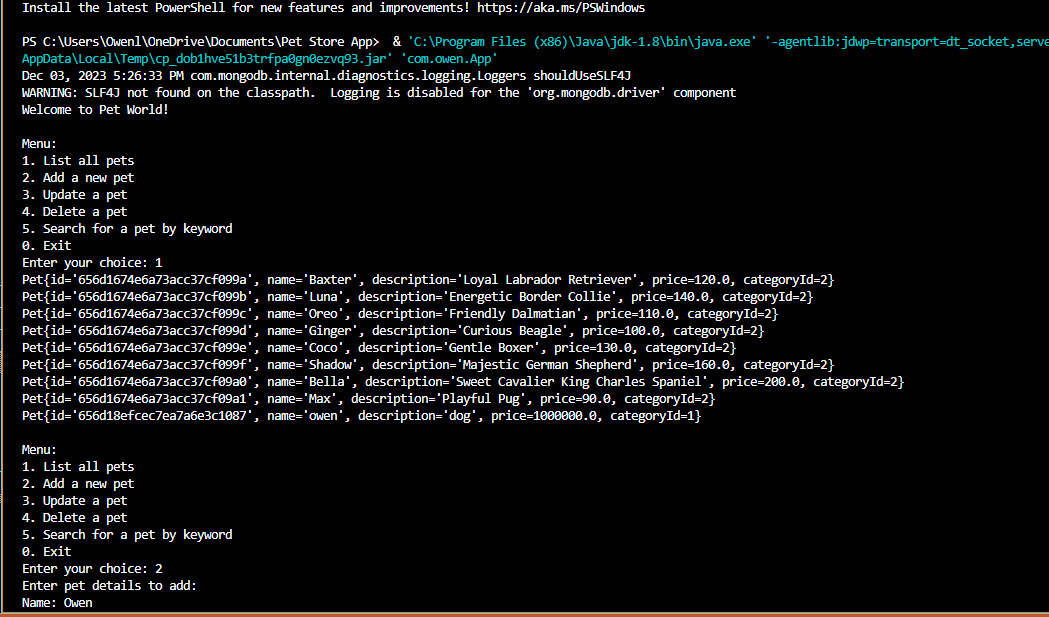
Mongo University:



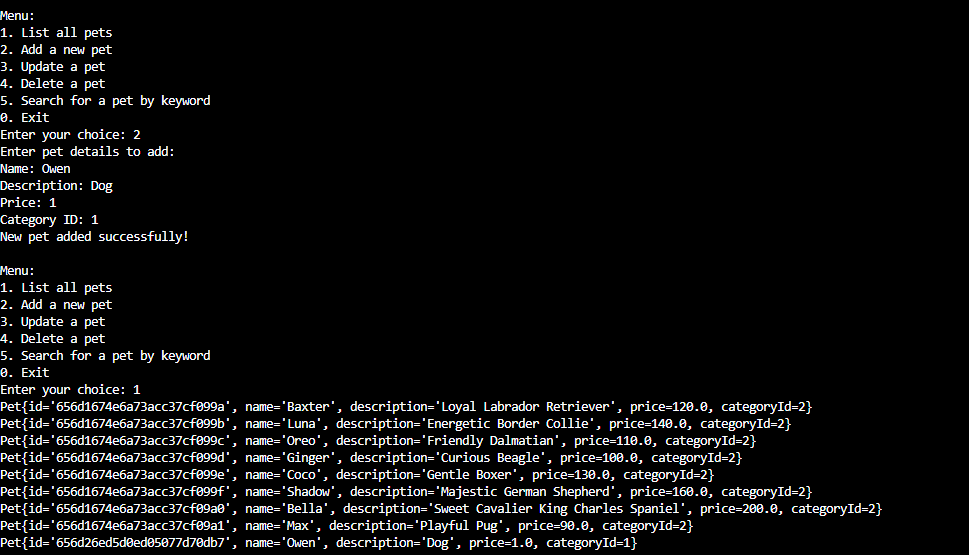
Part 2: Pets Mongo DB with a Java application

This application follows a menu that is labeled 0-5. I will take you through the operations of the menu sequentially from 1-5 as 0 is just an exit application command.

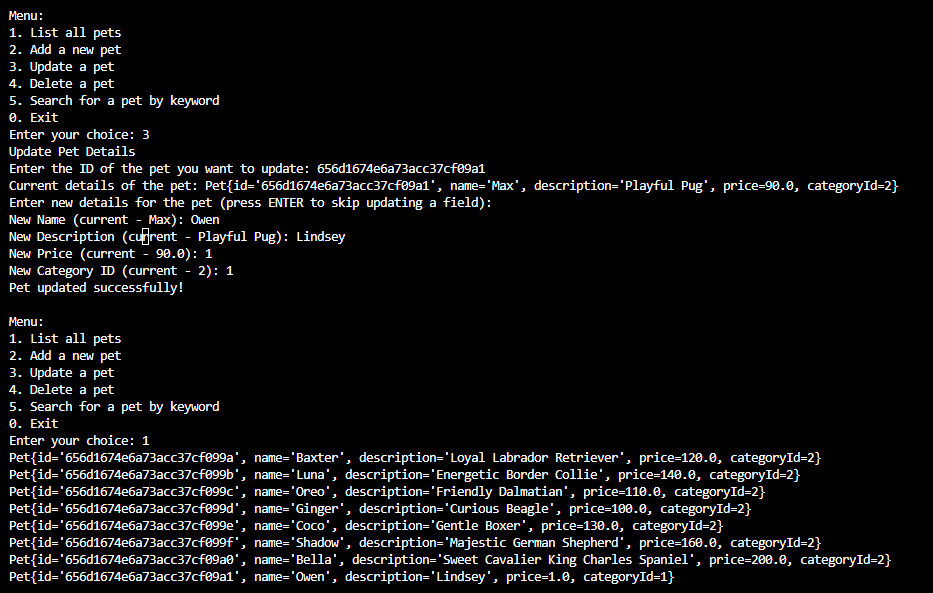
If the user supplies the console with the number “1” the user will see the full list of pets in our database. Here is a screenshot of the console showing all the animals.



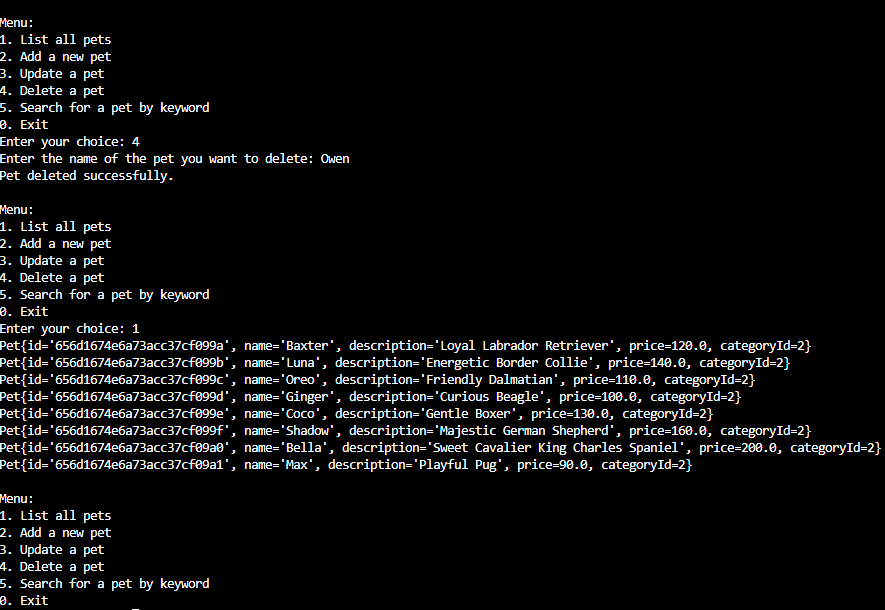
If the user writes the number “2” in the console the user will be able to add a whole new pet to the database. The program will ask them to fill out certain fields like name, description, etc. Then it will be assigned a random ID number and stored in the database. This is a screenshot of that operation.



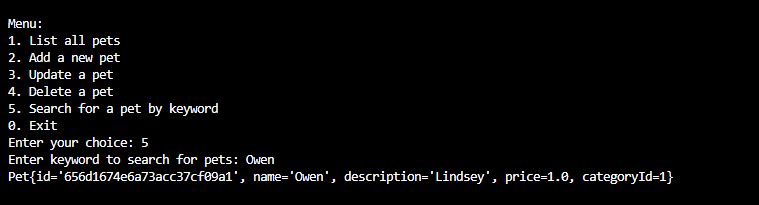
Option 3 provides the user with the ability to update any given detail of a pet in our collection. We can change its name, description and then view those changes by selecting option 1 and viewing the updated list of current pets in the collection.



Option 4 allows the user to remove a pet from the collection by its name. Type the name and a success message will show to the user and then use option 1 to view the updated collection.



Then finally option 5 will allow the user to search for any single pet in the collection. The user will select 5 and then be prompted to provide the name of the pet they are looking for. This screenshot shows successful searching through the collection.



Summary of lesson:

In this lesson, we explored the development of a Java console application for a pet store, emphasizing database interaction, object-oriented programming, and user interface design. We designed aPet class to model pet entities, encapsulating attributes like ID, name, description, price, and category ID, demonstrating the principles of encapsulation and data abstraction.

The PetStoreDataAccessObject (DAO) class highlighted the implementation of database operations, including CRUD (Create, Read, Update, Delete) functionalities, using MongoDB as the backend database. We utilized the MongoDB Java driver for database connectivity and executed queries to interact with the database. The application's architecture was highlighted by separating the data access layer (DAO) from the user interface in the App class, which handled user input and displayed information in the console.

This separation of concerns ensures maintainability and scalability. Additionally, we addressed common programming challenges such as handling user input, managing exceptions, and ensuring data type compatibility between the application and the database. The lesson culminated in integrating these components to build a functional console-based application for managing a pet store's inventory, demonstrating a practical application of database programming in a real-world scenario.