Yony Rober

Project 1 - Pet Supply Management System

Description:

I built a comprehensive Pet Supply Management System using Python which would allow employees to manage pets, customers, food, and toys with efficient use of time.

I started by creating an item class that represents the items, the price, and the quantity. I added methods that display the item and the quantity.

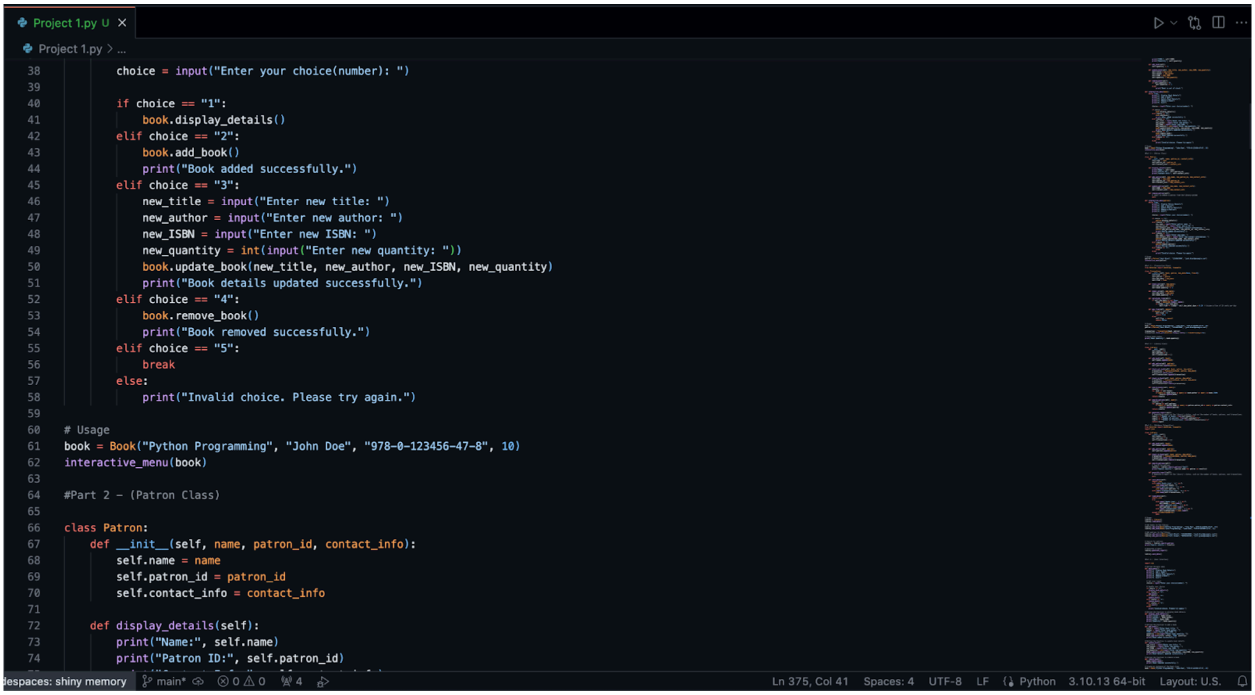
Next, I created a class to manage inventory. The inventory is a list that is edited by the methods in the class. The structure allows the user to manage the inventory effectively, adding and removing items as needed while keeping track of their details.

The third step was creating a transaction class that handles checkouts, returns, fines, and monetized aspects. I implemented methods for checking books in, checking books out, and keeping track of the due dates as well as fines.

The fourth step was building a library class to manage the entire library system. This was the most difficult part leading up to it because it required me to include all previous data into one “managing system” class which allowed me to search for books, manage patrons, handle transactions, and generate a report back as the output.

The fifth step was to set up data storage using a file-based system. I used JSON as the imported file format. This step was very difficult; I had trouble incorporating the functions to act accordingly. The search functions proved to be the most difficult. It took me hours just to figure out that my attributes had to be different for that part of the code.

The sixth step was to develop a graphic user interface to interact with the Library Managing System. I used the previous functions: the addition of a book, removal of a book, update of a book, and the details of a book to incorporate them into a “main menu”.



```python
        choice = input("Enter your choice(number): ")

        if choice == "1":
            book.display_details()
        elif choice == "2":
            book.add_book()
            print("Book added successfully.")
        elif choice == "3":
            new_title = input("Enter new title: ")
            new_author = input("Enter new author: ")
            new_ISBN = input("Enter new ISBN: ")
            new_quantity = int(input("Enter new quantity: "))
            book.update_book(new_title, new_author, new_ISBN, new_quantity)
            print("Book details updated successfully.")
        elif choice == "4":
            book.remove_book()
            print("Book removed successfully.")
        elif choice == "5":
            break
        else:
            print("Invalid choice. Please try again.")

# Usage
book = Book("Python Programming", "John Doe", "978-0-123456-47-8", 10)
interactive_menu(book)

#Part 2 - (Patron Class)

class Patron:
    def __init__(self, name, patron_id, contact_info):
        self.name = name
        self.patron_id = patron_id
        self.contact_info = contact_info

    def display_details(self):
        print("Name:", self.name)
        print("Patron ID:", self.patron_id)
```

The final step of the coding portion was to implement a role-based access control for librarians and administrators. These are the advanced features. I created a user class that requires an authentication system to log into the interface. I created a librarian class with certain power to access. For example, the librarian can update the books and patrons. I also created an administrative class that generates reports.

Using The Code:

I have clear instructions to use the code. All inputs will be easy to provide. Press the number of the action you choose to complete and proceed accordingly depending on what the output says.

Example of Code Running:

Sources:

Code Academy

W3 Schools

Google Colab(Strictly for Assistance)

Link to Github:

Github