

My Personal Library Manager: A Console App to Organize and Track Your Books Using Python

```
#os module in Python:  
#The os module is a built-in Python module that provides functions to interact with the Operating System (like working with files, folders, and environment variables).  
import os  
  
# List to hold book dictionaries  
library = []  
  
# Function to Add Books  
def add_book(title, author, genre, year, status="unread"):  
    book = {  
        "title": title.strip(),  
        "author": author.strip(),  
        "genre": genre.strip(),  
        "year": int(year),  
        "status": status.lower()  
    }  
    library.append(book)  
    print(f"Book '{title}' added.")  
  
# Function to List Books  
def list_books(sorted_by_year=False):  
    if not library:  
        print("Library is empty.")  
        return  
    books = sorted(library, key=lambda x: x['year']) if sorted_by_year  
else library  
    for idx, book in enumerate(books, start=1):  
        print(f"{idx}. {book['title']} by {book['author']}  
[{book['genre']}, {book['year']}] - {book['status'].capitalize()}")  
  
# Function Search by Author  
def search_by_author(author_name):  
    return [book for book in library if author_name.lower() in  
book['author'].lower()]  
  
# Function to Delete Book  
def delete_book(title):  
    global library  
    library = list(filter(lambda book: book['title'].lower() !=  
title.lower(), library))  
    print(f"Book titled '{title}' has been removed if it existed.")
```

```

# Show Reading Summary
def show_reading_summary():
    if not library:
        print("No books in library.")
        return
    read = len(list(filter(lambda x: x['status'] == 'read', library)))
    unread = len(library) - read
    print(f"Read: {read} books ({(read/len(library))*100:.2f}%)")
    print(f"Unread: {unread} books ({(unread/len(library))*100:.2f}%)")

# Convert Titles of the Books to Capital Letter
def uppercase_titles():
    titles = list(map(lambda b: b['title'].upper(), library))
    print("Book Titles (UPPERCASE):")
    for title in titles:
        print("-", title)

# Function to Save library of books to a file
def save_to_file(filename):
    try:
        with open(filename, 'w') as f:

            for book in library:
                line = f"{book['title']},{book['author']}, {book['genre']},{book['year']},{book['status']}"
                f.write(line)
        print(f"Saved {len(library)} books to {filename}") # After saving all the books, this line prints a confirmation message. It says how many books were saved and the name of the file.
    except Exception as e:
        print("Error saving file:", e)

# Loading Book Data from a File
def load_from_file(filename):
    if not os.path.exists(filename):
        print("File not found. Starting with an empty library.")
        return
    try:
        with open(filename, 'r') as f:
            for line in f:
                title, author, genre, year, status =
line.strip().split(',')
                add_book(title, author, genre, int(year), status)
    except Exception as e:
        print("Error reading file:", e)

# Building the Menu Function for Your Library App
def menu():
    while True:

```

```

print("\n--- Personal Library Menu ---")
print("1. Add Book")
print("2. List All Books")
print("3. List Books Sorted by Year")
print("4. Search by Author")
print("5. Delete Book by Title")
print("6. Show Reading Summary")
print("7. Show UPPERCASE Book Titles (map)")
print("8. Save Library")
print("9. Load Library")
print("10. Exit")
choice = input("Choose an option: ")

if choice == "1":
    title = input("Title: ")
    author = input("Author: ")
    genre = input("Genre: ")
    year = input("Year: ")
    status = input("Status (read/unread): ")
    add_book(title, author, genre, year, status)
elif choice == "2":
    list_books()
elif choice == "3":
    list_books(sorted_by_year=True)
elif choice == "4":
    author = input("Enter author name: ")
    results = search_by_author(author)
    print(f"\n Found {len(results)} books by {author}:")
    for book in results:
        print(f"- {book['title']} ({book['year']})")
elif choice == "5":
    title = input("Enter title to delete: ")
    delete_book(title)
elif choice == "6":
    show_reading_summary()
elif choice == "7":
    uppercase_titles()
elif choice == "8":
    save_to_file("books_data.txt")
elif choice == "9":
    load_from_file("books_data.txt")
elif choice == "10":
    print("Goodbye!")
    break
else:
    print("Invalid option. Try again.")

## Running Your Library App with the Final Command
if __name__ == "__main__": # This line checks how the Python file is

```

being run.

```
menu()
```

```
--- Personal Library Menu ---
1. Add Book
2. List All Books
3. List Books Sorted by Year
4. Search by Author
5. Delete Book by Title
6. Show Reading Summary
7. Show UPPERCASE Book Titles (map)
8. Save Library
9. Load Library
10. Exit
```