**Intent:**

The purpose of this project is to design a Library Management System (LMS) using Python that efficiently manages library resources such as student records and book records. The system allows users to perform operations like creating, displaying, modifying, and deleting records, as well as issuing and depositing books. Data is stored permanently in files using the pickle module, ensuring easy access and management of information while promoting independent learning and effective resource handling.

**Project Overview:**

A Library Management System (LMS) manages resources in a library, improving efficiency and supporting independent learning.

Implementation Options:

Option 1:

Use a database to implement the system.

Option 2:

Use two files: book.dat and student.dat to permanently store book and student information. Suggested tools:

pickle module: For writing and reading data.

os module: For managing filenames (create/rename/remove).

**Imports and Setup**

import pickle

import os

**Student Management Functions**

Create Student Record

def create\_student ():

student = {}

student['id'] = input ("Enter student ID: ")

student['name'] = input ("Enter student name: ")

with open ("student.dat", "ab") as f:

pickle. dump (student, f)

print ("Student record created.\n")

**Display All Student Records**

def display\_all\_students ():

try:

with open ("student.dat", "rb") as f:

while True:

student = pickle. load(f)

print(student)

except EOFError:

pass

except FileNotFoundError:

print ("No student records found.\n")

**Display Specific Student Record**

def display\_specific\_student(student\_id):

found = False

try:

with open ("student.dat", "rb") as f:

while True:

student = pickle. load(f)

if student['id'] == student\_id:

print(student)

found = True

break

except EOFError:

if not found:

print ("Student not found.\n")

except FileNotFoundError:

print ("No student records found.\n")

**Book Management Functions**

Create Book

def create\_book ():

book = {}

book['id'] = input ("Enter book ID: ")

book['title'] = input ("Enter book title: ")

with open("book.dat", "ab") as f:

pickle.dump(book, f)

print("Book record created.\n")

**Display All Books**

def display\_all\_books ():

try:

with open("book.dat", "rb") as f:

while True:

book = pickle.load(f)

print(book)

except EOFError:

pass

except FileNotFoundError:

print("No book records found.\n")

**Display Specific Book**

def display\_specific\_book(book\_id):

found = False

try:

with open("book.dat", "rb") as f:

while True:

book = pickle.load(f)

if book['id'] == book\_id:

print(book)

found = True

break

except EOFError:

if not found:

print("Book not found.\n")

except FileNotFoundError:

print("No book records found.\n")

**Main Menu**

def main\_menu():

while True:

print("\nLibrary Management System")

print("1. Book Issue (To be implemented)")

print("2. Book Deposit (To be implemented)")

print("3. Administration Menu")

print("4. Exit")

choice = input("Enter your choice: ")

if choice == '1':

print("Book Issue feature under development.\n")

elif choice == '2':

print("Book Deposit feature under development.\n")

elif choice == '3':

admin\_menu()

elif choice == '4':

break

else:

print("Invalid choice. Try again.\n")

**Administration Menu**

def admin\_menu():

while True:

print("\nAdministration Menu")

print("1. Create Student Record")

print("2. Display All Students")

print("3. Display Specific Student")

print("4. Create Book")

print("5. Display All Books")

print("6. Display Specific Book")

print("7. Back to Main Menu")

choice = input("Enter your choice: ")

if choice == '1':

create\_student()

elif choice == '2':

display\_all\_students()

elif choice == '3':

student\_id = input("Enter Student ID: ")

display\_specific\_student(student\_id)

elif choice == '4':

create\_book()

elif choice == '5':

display\_all\_books()

elif choice == '6':

book\_id = input("Enter Book ID: ")

display\_specific\_book(book\_id)

elif choice == '7':

break

else:

print("Invalid choice. Try again.\n")

**Run Program**

if \_name\_ == "\_main\_":

main\_menu ()

**Summary**

* This Python program uses pickle for file-based permanent storage.
* Students and books are stored in separate .dat files.
* Book issue and deposit features are placeholders you can add later.
* Menus are fully navigable through the console.