LIBRARY MANAGEMENT SYSTEM

A Library Management System (LMS) gives access to and manages the resources in your library. A well-chosen system will increase your library’s efficiency, save valuable administration time, lead to a better educational experience for pupils and help develop independent learning.

OUR PROJECT WORK

Create following menu options: 1. BOOK ISSUE  
2. BOOK DEPOSIT  
3. ADMINISTRATION MENU

* • CREATE STUDENT RECORD
* • DISPLAY ALL STUDENTS RECORD
* • DISPLAY SPECIFIC STUDENT RECORD
* • MODIFY STUDENT RECORD
* • DELETE STUDENT RECORD
* • CREATE BOOK
* • DISPLAY ALL BOOKS
* • DISPLAY SPECIFIC BOOK
* • MODIFY BOOK
* • DELETE BOOK RECORD

4. EXIT

Hand Notes for:

Projectwork :

from pickle import load, dump  
from os import remove, rename  
import os  
import random  
import time  
  
  
print()  
print("\t \t \t Library Management Project \t \t \t")  
print()  
a = input("Enter to Continue:")  
a = str(a)  
if a.isalpha():  
 pass  
  
class book:  
 def \_\_init\_\_(self, bnum=" ", bnam=" ", wrnam=" "):  
 self.bnum = bnum  
 self.bnam = bnam  
 self.wrnam = wrnam  
  
 def create\_book(self):  
 print()  
 print("\t \t \t Creating record of the Book \t \t \t")  
 print()  
 self.bnum = input("\t Enroll book number: ")  
 print()  
 self.bnam = input("\t Provide name of the book: ")  
 print()  
 self.wrnam = input("\t Provide name of writer: ")  
 print()  
 print()  
 print("\t \t \t Book Created \t \t \t")  
  
 def display\_book(self):  
 print()  
 print()  
 print("\t \t Book number: ", self.bnum)  
 print()  
 print("\t \t Book name: ", self.bnam)  
 print()  
 print("\t \t Writer name: ", self.wrnam)  
 print()  
 print()  
  
 def modify\_book(self):  
 print()  
 print()  
 print("\t \t Book number: ", self.bnum)  
 print()  
 self.bnam = input("\t \t Enroll New Book name: ")  
 print()  
 self.wrnam = input("\t \t Writer name: ")  
 print()  
 print()  
 print("\t \t Book is modifed")  
 print()  
  
 def ret\_bnum(self):  
 return (self.bnum)  
  
 def report\_book(self):  
 print(self.bnum, self.bnam, self.wrnam)  
  
  
class student:  
 def \_\_init\_\_(self, adnum=" ", name=" ", stdnum=" ", token=0):  
 self.adnum = adnum  
 self.name = name  
 self.stdnum = stdnum  
 self.token = token  
  
 def createstud(self):  
 print()  
 print("\t \t \t Create Student Record \t \t \t")  
 print()  
 self.adnum = input("\t \t Enter Admission number: ")  
 print()  
 self.name = input("\t \t Enter name of student: ")  
 self.stdnum = " "  
 self.token = 0  
 print()  
 print()  
 print("\t \t \t Student record created \t \t \t")  
 print()  
 print('#' \* 70)  
 print()  
  
 def showstud(self):  
 print()  
 print()  
 print("\t Admission num :", self.adnum)  
 print()  
 print("\t Name:", self.name)  
 print()  
 print("\t Stdnum:", self.stdnum)  
 print()  
 print()  
  
 def displaystud(self):  
 print()  
 print("\t Admission number of Student is:", self.adnum)  
 print()  
 print("\t Name of student is:", self.name)  
 if (self.token == 1):  
 print("\t Book number is:", self.stdnum)  
  
 def modifystud(self):  
 print()  
 print("\t Admission num:", self.adnum)  
 print()  
 self.name = input("\t New Student name:")  
 print()  
 print("\t \t Students name Modified !!")  
  
 def ret\_adnum(self):  
 return self.adnum  
  
 def ret\_stdnum(self):  
 return self.stdnum  
  
 def ret\_token(self):  
 return self.token  
  
 def add\_token(self):  
 self.token = 1  
  
 def reset\_token(self):  
 self.token = 0  
  
 def get\_stdnum(self, t):  
 self.stdnum = t  
  
 def reportstud(self):  
 print(self.adnum, self.name, self.token)  
  
def writebook ():  
 ch = "Y"  
 fp = open("book1.dat", "ab")  
 while ch == "Y":  
 bk.create\_book()  
 dump(bk, fp)  
 print()  
 ch = input("\t \t Do You Want to Continue<y/n>:")  
 print()  
 print("#" \* 70)  
 print()  
 ch = ch.upper()  
  
  
def studentwrite():  
 ch = "Y"  
 fp = open("student1.dat", "ab")  
 while ch == "Y":  
 st.createstud()  
 dump(st, fp)  
 ch = input("\t \t Do you wanna continue <y/n>:")  
 ch = ch.upper()  
 print()  
  
  
def display\_alls():  
 fin = open("student1.dat", "rb")  
 if not (fin):  
 print()  
 "\t \t File is not Found..."  
 else:  
 try:  
 while True:  
 print()  
 st = load(fin)  
 st.showstud()  
 except EOFError:  
 pass  
 fin.close()  
  
  
def display\_allb():  
 fin = open("book1.dat", "rb")  
 if not(fin):  
 print()  
 print()  
 print("Book File is no found....")  
 else:  
 try:  
 while True:  
 bk = load(fin)  
 bk.display\_book()  
 except EOFError:  
 pass  
 fin.close()  
  
  
def display\_spb(no):  
 flag = 0  
 fin = open("book1.dat", "rb")  
 try:  
 while True:  
 bk = load(fin)  
 if (bk.ret\_bnum() == no):  
 bk.display\_book()  
 flag = 1  
 except EOFError:  
 pass  
 fin.close()  
 if flag == 0:  
 print()  
 print()  
 print("\t \t \t \t Book Not Present...!!")  
  
  
def display\_sps(n):  
 flag = 0  
 fin = open("student1.dat", "rb")  
 try:  
 while True:  
 st = load(fin)  
 if (st.ret\_adnum() == n):  
 st.showstud()  
 flag = 1  
 except EOFError:  
 pass  
 fin.close()  
 if flag == 0:  
 print()  
 print("\t \t \t Student is not present... ")  
  
  
def modify\_bookrecord():  
 found = 0  
 print()  
 print()  
 print("\t \t \t Modify Book")  
 print()  
 print()  
 n = input("\t \t Enter book number to modified:")  
 print()  
 fin = open("book1.dat", "rb")  
 fout = open("temp.dat", "wb")  
 try:  
 while True:  
 bk = load(fin)  
 if bk.ret\_bnum() == n:  
 print()  
 print("\t \t \t Book details")  
 bk.display\_book()  
 print()  
 print("\t \t Enter new Details")  
 print()  
 print()  
 bk.modify\_book()  
 dump(bk, fout)  
 found = 1  
 else:  
 dump(bk, fout)  
 except EOFError:  
 pass  
 if found == 0:  
 print("\t \t \t Book not present")  
 fin.close()  
 fout.close()  
 remove("book1.dat")  
 rename("temp.dat", "book1.dat")  
  
  
def modify\_student\_record():  
 found = 0  
 print()  
 print("\t \t Modify Student record")  
 print()  
 print()  
 n = input("\t \t Enter students admission number to be modified:")  
 print()  
 fin = open("student1.dat", "rb")  
 fout = open("temp.dat", "wb")  
 try:  
 while True:  
 st = load(fin)  
 if st.ret\_adnum() == n:  
 print()  
 print("\t \t \t STUDENTS DETAILS")  
 st.showstud()  
 print()  
 print("\t \t \t Enter new student details:")  
 st.modifystud()  
 dump(st, fout)  
 found = 1  
 else:  
 dump(st, fout)  
 except EOFError:  
 pass  
 if found == 0:  
 print("\t \t \t Student not present")  
 fin.close()  
 fout.close()  
 remove("student1.dat")  
 rename("temp.dat", "student1.dat")  
  
  
def del\_stud():  
 flag = 0  
 print()  
 print()  
 n = input("\t \t Enter admission to be deleted:")  
 print()  
 fin = open("student1.dat", "rb")  
 fout = open("temp.dat", "wb")  
 try:  
 while True:  
 st = load(fin)  
 if st.ret\_adnum() != n:  
 dump(st, fout)  
 else:  
 flag = 1  
 except EOFError:  
 pass  
 fin.close()  
 fout.close()  
 remove("student1.dat")  
 rename("temp.dat", "student1.dat")  
 if flag == 1:  
 print()  
 print("\t \t \t \t Record deleted..!!")  
 else:  
 print()  
 print("\t \t \t \t Sorry..!! Record does not exist..!!..")  
  
  
def del\_book():  
 flag = 0  
 print()  
 print()  
 n = input("\t \t Enter bok no to be deleted:")  
 print()  
 fin = open("book1.dat", "rb")  
 fout = open("temp.dat", "wb")  
 try:  
 while True:  
 bk = load(fin)  
 if bk.ret\_bnum() != n:  
 dump(bk, fout)  
 else:  
 flag = 1  
 except EOFError:  
 pass  
 fin.close()  
 fout.close()  
 remove("book1.dat")  
 rename("temp.dat", "book1.dat")  
 if flag == 1:  
 print("\t \t \t Record deleted")  
 else:  
 print("\t \t \t Sorry..!! Record not present..")  
  
  
def book\_issue():  
 sn = " "  
 bn = " "  
 found = 0  
 flag = 0  
 print()  
 print()  
 print("\t \t \t Book Issue \t \t \t")  
 print()  
 print()  
 sn = input("\t \t Enter the students admission number:")  
 print()  
 fin1 = open("book1.dat", "rb")  
 fin2 = open("student1.dat", "rb")  
 fout = open("temp.dat", "wb")  
 try:  
 while True:  
 st = load(fin2)  
 if (st.ret\_adnum() == sn):  
 st.showstud()  
 found = 1  
 if st.ret\_token() == 0:  
 bn = input("\t \t Enter book number:")  
 try:  
 while True:  
 bk = load(fin1)  
 if bk.ret\_bnum() == bn:  
 bk.display\_book()  
 flag = 1  
 st.add\_token()  
 st.get\_stdnum(bk.ret\_bnum())  
 dump(st, fout)  
 os.system("clear")  
 print()  
 print()  
 print("\t \t \t Book Issue sucessfully \t \t \t")  
 print()  
 print("\t Please note : Write the current date behind of your book")  
 print("\t \t and submit within 25 days")  
 print()  
 print("\t \t Fine rs 20 for each day after 25 days period..!!")  
 print()  
 except EOFError:  
 pass  
 else:  
 print("\t You have not returned the last book..")  
 except EOFError:  
 pass  
 if (flag == 0):  
 print("\t \t \t No such book present!!")  
 if (found == 0):  
 print("\t \t \t No such students exists !!")  
 fin1.close()  
 fin2.close()  
 fout.close()  
 remove("student1.dat")  
 rename("temp.dat", "student1.dat")  
  
  
def book\_deposit():  
 print()  
 print()  
 print()  
 print("\t \t \t Book Depositing..")  
 sn = " "  
 found = 0  
 flag = 0  
 day = 0  
 fine = 0  
 print()  
 print()  
 sn = input("\t \t Enter students admissions number:")  
 print()  
 fin1 = open("student1.dat", "rb")  
 fin2 = open("book1.dat", "rb")  
 fout = open("temp.dat", "wb")  
 try:  
 while True:  
 st = load(fin1)  
 if st.ret\_adnum() == sn:  
 found = 1  
 print()  
 print("\t Student token number", st.ret\_token())  
 if st.ret\_token() == 1:  
 try:  
 while True:  
 bk = load(fin2)  
 if bk.ret\_bnum() == st.ret\_stdnum():  
 bk.display\_book()  
 flag = 1  
 print()  
 days = int(input("\t Book deposited in no. of days:"))  
 if days >= 15:  
 fine = (days - 15) \* 20  
 print()  
 print("\t Fine : Rs.", fine)  
 st.reset\_token()  
 st.get\_stdnum(0)  
 st.showstud()  
 dump(st, fout)  
 print()  
 print("\t \t Book deposited !!!")  
  
 except EOFError:  
 pass  
 else:  
 print()  
 print("\t \t You have not issued the book..")  
  
 except EOFError:  
 pass  
 if (found == 0):  
 print()  
 print("\t No Sch student Exists")  
 fin1.close()  
 fin2.close()  
 fout.close()  
 remove("student1.dat")  
 rename("temp.dat", "student1.dat")  
  
  
bk = book()  
st = student()  
  
  
def adminmenu():  
 ch = "Y"  
 while ch == "Y":  
 print()  
 print()  
 print("\t \t \t Administration Menu \t \t \t")  
 print()  
 print()  
 print("\t 1. Create student record")  
 print()  
 print("\t 2. Display all students record")  
 print()  
 print("\t 3. Display specific students record")  
 print()  
 print("\t 4. Modify student record")  
 print()  
 print("\t 5. Delete student record")  
 print()  
 print("\t 6. create book")  
 print()  
 print("\t 7. Display all books")  
 print()  
 print("\t 8. Display specific book")  
 print()  
 print("\t 9. Modify book")  
 print()  
 print("\t 10. Delete book record")  
 print()  
 ch1 = int(input("\t \t Enter you choice:"))  
 print()  
 os.system("clear")  
 if ch1 == 1:  
 studentwrite()  
 elif ch1 == 2:  
 display\_alls()  
 elif ch1 == 3:  
 print()  
 print()  
 ad = input("\t \t Enter student's Admno to be dislayed:")  
 display\_sps(ad)  
 elif ch1 == 4:  
 modify\_student\_record()  
 elif ch1 == 5:  
 del\_stud()  
 elif ch1 == 6:  
 writebook()  
 elif ch1 == 7:  
 display\_allb()  
 elif ch1 == 8:  
 print()  
 print()  
 bn = input("\t \t Enter book number to be displayed:")  
 display\_spb(bn)  
 elif ch1 == 9:  
 modify\_bookrecord()  
 elif ch1 == 10:  
 del\_book()  
 print()  
 ch = input("\t \t Do you want continue with ADMINMENU<y/n>:")  
 ch = ch.upper()  
 print()  
 os.system("clear")  
 if ch == "Y":  
 continue  
 else:  
 mainmenu()  
  
  
  
  
def mainmenu():  
 ch = "Y"  
 while ch == "Y":  
 print()  
 print()  
 print("\t \t \t Main Menu \t \t \t")  
 print()  
 print("\t 1. Book issue")  
 print()  
 print("\t 2. Book Deposit")  
 print()  
 print("\t 3. Administration Menu")  
 print()  
 print("\t 4. Exit")  
 print()  
 ch1 = int(input("\t \t Enter Your Choice:"))  
 print()  
 print("\t \t Loading....")  
 time.sleep(5)  
 os.system("clear")  
 if ch1 == 1:  
 book\_issue()  
 elif ch1 == 2:  
 book\_deposit()  
 elif ch1 == 3:  
 adminmenu()  
 else:  
 exit(0)  
 print()  
 ch = input("\t \t \t Do you want to continue <y/n>:")  
 ch = ch.upper()  
 if ch == "N":  
 break  
 print()  
 print()  
 print("\t \t Loading...")  
 time.sleep(30)  
 os.system("clear")  
  
  
os.system("clear")  
mainmenu()