|  |  |  |
| --- | --- | --- |
| **TABLE OF CONTENTS** | | |
| **SERIAL NO** | **DESCRIPTION** | **PAGE NO** |
| 01 | ACKNOWLEDGEMENT |  |
| 02 | INTRODUCTION |  |
| 03 | OBJECTIVES OF THE PROJECT |  |
| 04 | FLOW OF EXECUTION |  |
| 05 | SOURCE CODE |  |
| 06 | OUTPUT |  |
| 07 | HARDWARE AND SOFTWARE REQUIREMENTS |  |
| 08 | BIBLIOGRAPHY |  |

**ACKNOWLEDGEMENT**

**I would like to express a deep sense of thanks and gratitude to my project guide Mr. Raman Kumar for guiding me immensely through the course of the project. He always evinced keen interest in my work. His constructive advice and constant motivation have been responsible for the successful completion of this project.**

**I express my deep sense of gratitude to the luminary The Principal, Colonel Arun Datta who has been continuously motivating and extending their helping hand to us.**

**I also thanks to my parents for their motivation and support. I must thanks to my classmates for their timely help and support for compilation of this project.**

**Last but not the least, I would like to thank all those who had helped directly or indirectly towards the completion of this project.**

**INTRODUCTION**

**LIBRARY MANAGEMENT SYSTEM** is a simple project designed in Python Programming Language with MySQL. This project uses the all MySQL commands DDL (CREATE DATABASE, TABLE) and DML (INSERT, UPDATE, DELETE, and SELECT) through Python. The Python and MySQL Connectivity are done by using mysql.connector package. The all basic operations like Insert, Update, View and Delete are done in this project.

**PROJECT TITLE- “LIBRARY MANAGEMENT SYSTEM”**

DBMS: MySQL

Host: localhost

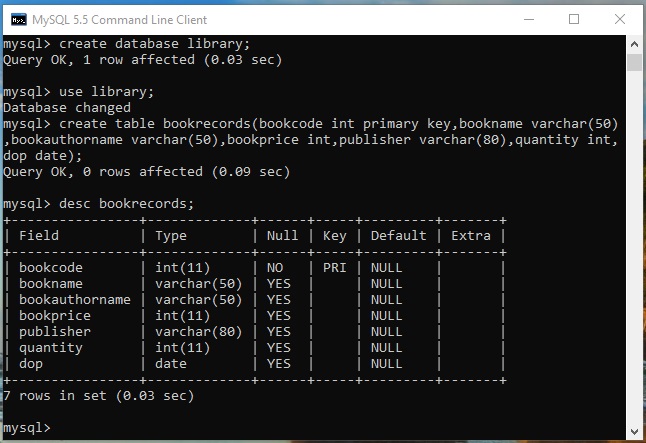
User: root

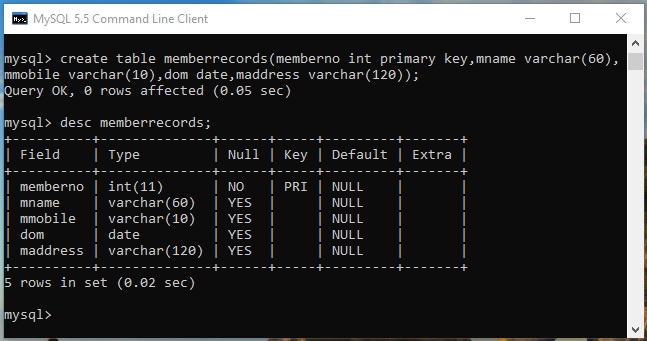
Password: root

DataBase: library

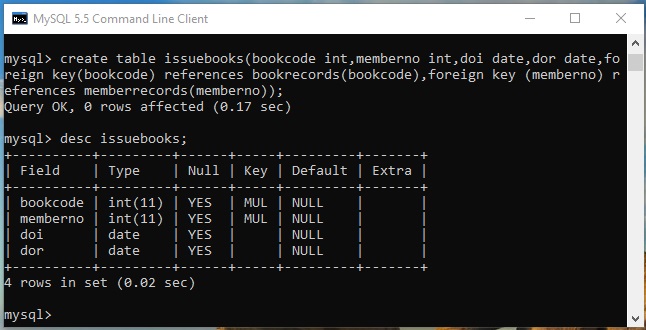
Table Structure: As per the screenshot given below:

**Table: bookrecords**



**Table: memberrecords**

**Table: issuebooks**



**OBJECTIVES OF THE PROJECT**

The objective of this project is to let the students apply the programming knowledge into a real- world situation/problem and exposed the students how programming skills helps in developing a good software.

1. Write programs utilizing modern software tools.
2. Apply object oriented programming principles effectively when developing small to medium sized projects.
3. Write effective procedural code to solve small to medium sized problems.
4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.
5. Students will demonstrate ability to conduct a research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.

**FLOW OF EXECUTION**

**SOURCE CODE**

**#Python Module:Library Management**

import menulib

import book

import member

import issue

while True:

print("\t\t\*\*\*\*\*\*Library Management\*\*\*\*\*\*\n")

print("==========================================")

print("1. Book Management")

print("2. Members Management")

print("3. Issue/Return Book")

print("4. Database Setup")

print("5. Exit")

print("==========================================")

choice=int(input("Enter choice between 1 to 4 -------> :"))

if choice==1:

menulib.MenuBook()

elif choice==2:

menulib.MenuMember()

elif choice==3:

menulib.MenuIssueReturn()

elif choice==4:

menulib.DataBase()

elif choice==5:

break

else:

print("Wrong choice.......Enter your choice again")

x=input("enter any key to continue")

**#Python Module: Menulib**

import book

import member

import issue

import MyDatabase

def MenuBook():

while True:

print("\t\t\*\*\*\*\*Book Record Management\*\*\*\*\*\n")

print("=====================================")

print("1. Add Book Records")

print("2. Display Book Records")

print("3. Search Book Records")

print("4. Delete Book Records")

print("5. Update Book Records")

print("6. Return to Main Menu")

print("======================================")

choice=int(input("Enter choice between 1 to 5-------->: "))

if choice==1:

book.AddRecords()

elif choice==2:

book.DisplayRecords()

elif choice==3:

book.SearchRecords()

elif choice==4:

book.DeleteRecords()

elif choice==5:

book.UpdateRecords()

elif choice==6:

return

else:

print("Wrong choice.......Enter your choice again")

x=input("Enter any key to continue")

def MenuMember():

while True:

print("\t\t\*\*\*\*\*Member Record Management\*\*\*\*\*\n")

print("=====================================")

print("1. Add Member Records")

print("2. Display Member Records")

print("3. Search Member Records")

print("4. Delete Member Records")

print("5. Update Member Records")

print("6. Return to Main Menu")

print("======================================")

choice=int(input("Enter choice between 1 to 5-------->: "))

if choice==1:

member.AddMember()

elif choice==2:

member.DisplayMember()

elif choice==3:

member.SearchMember()

elif choice==4:

member.DeleteMember()

elif choice==5:

member.UpdateMember()

elif choice==6:

return

else:

print("Wrong choice.......Enter your choice again")

x=input("Enter any key to continue")

def MenuIssueReturn():

while True:

print("\t\t\*\*\*\*\*Member Record Management\*\*\*\*\*\n")

print("=====================================")

print("1. Issue Book")

print("2. Display Issued Book Records")

print("3. Return Issued Book")

print("4. Return to Main Menu")

print("======================================")

choice=int(input("Enter choice between 1 to 4-------->: "))

if choice==1:

issue.IssueBook()

elif choice==2:

issue.DisplayIssuedBook()

elif choice==3:

issue.ReturnBook()

elif choice==4:

return

else:

print("Wrong choice.......Enter your choice again")

x=input("Enter any key to continue")

def DataBase():

while True:

print("\t\t\*\*\*\*\*Database Management\*\*\*\*\*\n")

print("=====================================")

print("1. Database Creation")

print("2. Creation of Relations")

print("3. List of Relations")

print("4. Return to Main Menu")

print("======================================")

choice=int(input("Enter choice between 1 to 4-------->: "))

if choice==1:

MyDatabase.CreateDatabase()

elif choice==2:

MyDatabase.CreateRelations()

elif choice==3:

MyDatabase.ShowRelations()

elif choice==4:

return

else:

print("Wrong choice.......Enter your choice again")

x=input("Enter any key to continue")

**#Python Module: Book**

from datetime import date,datetime,timedelta

import mysql.connector

def AddRecords():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

bcode=input("Enter Book Code: ")

bname=input("Enter Book Name: ")

bauthname=input("Enter Book Author's Name: ")

bprice=int(input("Enter Book Price: "))

publ=input("Enter Publisher of Book: ")

qty=int(input("Enter Quantity Purchased: "))

print("Enter Date of Purchase (Date/Month and year separately:)")

DD=int(input("Enter Date: "))

MM=int(input("Enter Month: "))

YY=int(input("Enter Year: "))

sql="insert into bookrecords values (%s,%s,%s,%s,%s,%s,%s)"

val=(bcode,bname,bauthname,bprice,publ,qty,date(YY,MM,DD))

mycursor.execute(sql,val)

mydb.commit()

mycursor.close()

mydb.close()

print("Records Inserted Successfully..........")

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

def DeleteRecords():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

bcode=input("Enter Book Code of Book to be Deleted from the Library: ")

sql="delete from bookrecords where bookcode=%s"

val=(bcode,)

mycursor.execute(sql,val)

mydb.commit()

mycursor.close()

mydb.close()

print("Records Deleted Successfully..........")

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

def SearchRecords():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

bcode=input("Enter Book Code to be Searched from the Library: ")

sql="select \* from bookrecords where bookcode=%s"

val=(bcode,)

mycursor.execute(sql,val)

rcount=0

for (bcode,bname,bauthname,bprice,publ,qty,dop) in mycursor:

rcount+=1

print("==============================================")

print("Book Code: ",bcode)

print("Book Name: ",bname)

print("Author of Book: ",bauthname)

print("Price of Book: ",bprice)

print("Publisher: ",publ)

print("Total Quantity in Hand: ",qty)

print("Purchased on: ",dop)

print("===============================================")

if rcount%2==0:

print(rcount,"Record(s) found")

mydb.commit()

mycursor.close()

mydb.close()

print("Records Searched Successfully..........")

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

def DisplayRecords():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

sql="select \* from bookrecords"

mycursor.execute(sql)

for (bcode,bname,bauthname,bprice,publ,qty,dop) in mycursor:

print("==============================================")

print("Book Code: ",bcode)

print("Book Name: ",bname)

print("Author of Book: ",bauthname)

print("Price of Book: ",bprice)

print("Publisher: ",publ)

print("Total Quantity in Hand: ",qty)

print("Purchased on: ",dop)

print("===============================================")

mydb.commit()

mycursor.close()

mydb.close()

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

def UpdateRecords():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

bcode=input("Enter Book Code of Book to be Updated from the Library: ")

sql="select \* from bookrecords where bookcode=%s"

val=(bcode,)

print("Enter New Record............")

bname=input("Enter Book Name: ")

bauthname=input("Enter Book Author's Name: ")

bprice=int(input("Enter Book Price: "))

publ=input("Enter Publisher of Book: ")

qty=int(input("Enter Quantity Purchased: "))

print("Enter Date of Purchase (Date/Month and year separately:)")

DD=int(input("Enter Date: "))

MM=int(input("Enter Month: "))

YY=int(input("Enter Year: "))

sql2="update bookrecords set bookname=%s, bookauthorname=%s, bookprice=%s, publisher=%s,quantity=%s,dop=%s where bookcode=%s"

val2=(bname,bauthname,bprice,publ,qty,date(YY,MM,DD),bcode)

mycursor.execute(sql2,val2)

mydb.commit()

mycursor.close()

mydb.close()

print("Records Updated Successfully..........")

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

**#Python Module: Member**

from datetime import date,datetime,timedelta

import mysql.connector

def AddMember():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

mcode=input("Enter Member Code: ")

mname=input("Enter Member Name: ")

mmob=input("Enter Member's Mobile Number: ")

print("Enter Date of Membership (Date/Month and year separately:)")

DD=int(input("Enter Date: "))

MM=int(input("Enter Month: "))

YY=int(input("Enter Year: "))

madd=input("Enter Member's Address: ")

sql="insert into memberrecords values(%s,%s,%s,%s,%s)"

val=(mcode,mname,mmob,date(YY,MM,DD),madd)

mycursor.execute(sql,val)

mydb.commit()

mycursor.close()

mydb.close()

print("Records Inserted Successfully..........")

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

def DeleteMember():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

mcode=input("Enter Member Code to be Deleted from the Library: ")

sql="delete from memberrecords where memberno=%s"

val=(mcode,)

mycursor.execute(sql,val)

mydb.commit()

mycursor.close()

mydb.close()

print("Records Deleted Successfully..........")

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

def SearchMember():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

mcode=input("Enter Member Code to be Searched from the Library: ")

sql="select \* from memberrecords where memberno=%s"

val=(mcode,)

mycursor.execute(sql,val)

rcount=0

for (mcode,mname,mmob,dom,madd) in mycursor:

rcount+=1

print("==============================================")

print("Member Code: ",mcode)

print("Member Name: ",mname)

print("Mobile Number of Member: ",mmob)

print("Date of membership: ",dom)

print("Address of Member: ",madd)

print("===============================================")

if rcount%2==0:

print(rcount,"Record(s) found")

mydb.commit()

mycursor.close()

mydb.close()

print("Records Searched Successfully..........")

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

def DisplayMember():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

sql="select \* from memberrecords"

mycursor.execute(sql)

for (mcode,mname,mmob,dom,madd) in mycursor:

print("==============================================")

print("Member Code: ",mcode)

print("Member Name: ",mname)

print("Mobile Number of Member: ",mmob)

print("Date of membership: ",dom)

print("Address of Member: ",madd)

print("===============================================")

mydb.commit()

mycursor.close()

mydb.close()

print("Records Displayed Successfully..........")

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

def UpdateMember():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

mcode=input("Enter Member Code to be Updated from the Library: ")

sql="select \* from memberrecords where memberno=%s"

val=(mcode,)

print("Enter New Record")

mname=input("Enter Member Name: ")

mmob=input("Enter Mobile Number of Member: ")

print("Enter Date of Membership (Date/Month and year separately:)")

DD=int(input("Enter Date: "))

MM=int(input("Enter Month: "))

YY=int(input("Enter Year: "))

madd=input("Enter Address of Member: ")

sql2="update memberrecords set mname=%s, mmobile=%s, dom=%s, maddress = %s where memberno=%s"

val2=(mname,mmob,date(YY,MM,DD),madd,mcode)

mycursor.execute(sql2,val2)

mydb.commit()

mycursor.close()

mydb.close()

print("Records Updated Successfully..........")

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

**#Python Module: Issue**

from datetime import date,datetime,timedelta

import mysql.connector

def IssueBook():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

bcode=input("Enter Book Code to issue: ")

mcode=input("Enter Member Code: ")

print("Enter Date of Issue (Date/Month and year separately:)")

DD=int(input("Enter Date: "))

MM=int(input("Enter Month: "))

YY=int(input("Enter Year: "))

sql="insert into issuebooks(bookcode,memberno,doi) values (%s,%s,%s)"

val=(bcode,mcode,date(YY,MM,DD))

mycursor.execute(sql,val)

mydb.commit()

mycursor.close()

mydb.close()

print("Records Inserted Successfully..........")

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

def ReturnBook():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

bcode=input("Enter Book Code of Book to be Returned to the Library: ")

mcode=input("Enter Member Code of Member who is returning Book: ")

dor=date.today()

sql2="update issuebooks set dor=%s where bookcode=%s and memberno=%s"

val2=(dor,bcode,mcode)

mycursor.execute(sql2,val2)

mydb.commit()

mycursor.close()

mydb.close()

print("Records Deleted Successfully..........")

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

def DisplayIssuedBook():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="library")

mycursor=mydb.cursor()

sql="select B.bookcode,B.bookname,M.memberno,M.mname,I.doi,I.dor from bookrecords B, issuebooks I, memberrecords M where B.bookcode=I.bookcode and I.memberno=M.memberno"

mycursor.execute(sql)

for (bcode,bname,mcode,mname,doi,dor) in mycursor:

print("==============================================")

print("Book Code: ",bcode)

print("Book Name: ",bname)

print("Member Code: ",mcode)

print("Member Name: ",mname)

print("Date of Issue: ",doi)

print("Date of Return: ",dor)

print("===============================================")

mydb.commit()

mycursor.close()

mydb.close()

print("I Have done It.......................")

except Exception as ex:

print(“Something went wrong”,ex)

mydb.close()

**#Python Module: MyDatabase**

#from datetime import date,datetime,timedelta

import mysql.connector

def CreateDatabase():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root")

mycursor=mydb.cursor()

print("Creating Library Database")

sql="create database if not exists LIBRARY"

mycursor.execute(sql)

print("LIBRARY Database Created Successfully....")

except Exception as ex:

print(“Something went wrong”,ex)

def CreateRelations():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="Library")

mycursor=mydb.cursor()

print("Creating bookrecords Relation")

sql="create table if not exists bookrecords(bookcode int primary key, bookname varchar(50) not null, bookauthorname varchar(50) not null, bookprice int,publisher varchar(80) not null, quantity int,dop date)"

mycursor.execute(sql)

print("bookrecords Relation Created Successfully....")

print("Creating memberrecords Relation")

sql="create table if not exists memberrecords(memberno int primary key, mname varchar(60) not null, mmobile varchar(10) not null, dom date, maddress varchar(120))"

mycursor.execute(sql)

print("memberrecords Relation Created Successfully....")

print("Creating issuebooks Relation")

sql="create table if not exists issuebooks(bookcode int, memberno int, doi date, dor date, foreign key (bookcode) references bookrecords(bookcode), foreign key(memberno) references memberrecords(memberno))"

mycursor.execute(sql)

print("issuebooks Relation Created Successfully....")

except Exception as ex:

print(“Something went wrong”,ex)

def ShowRelations():

try:

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database="LIBRARY")

mycursor=mydb.cursor()

print("Displaying List of Relations")

sql="show tables"

mycursor.execute(sql)

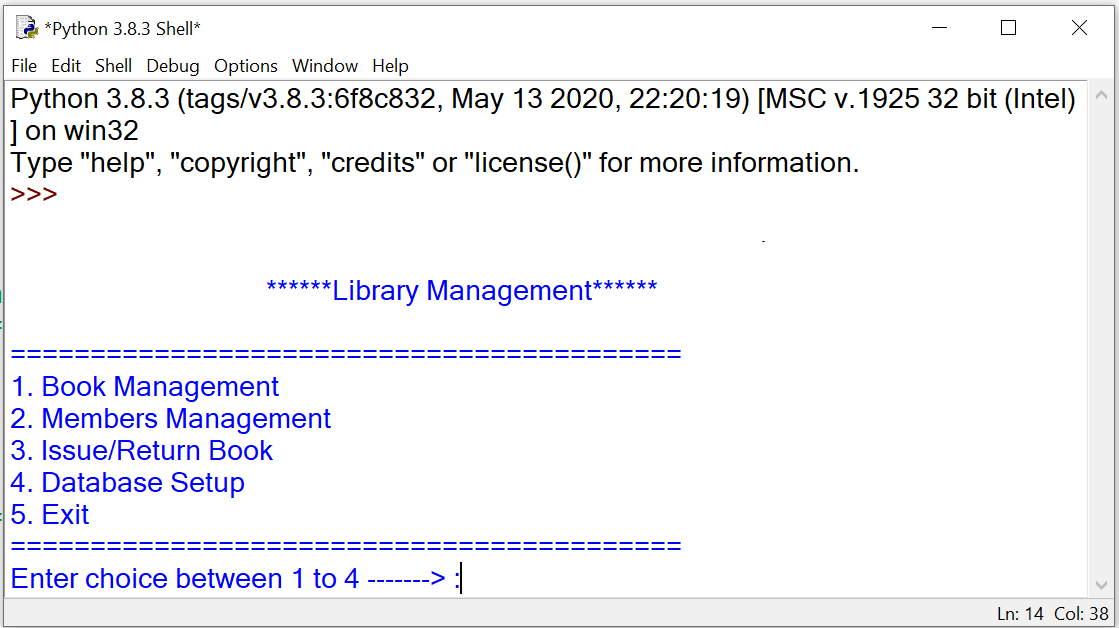
for i in mycursor:

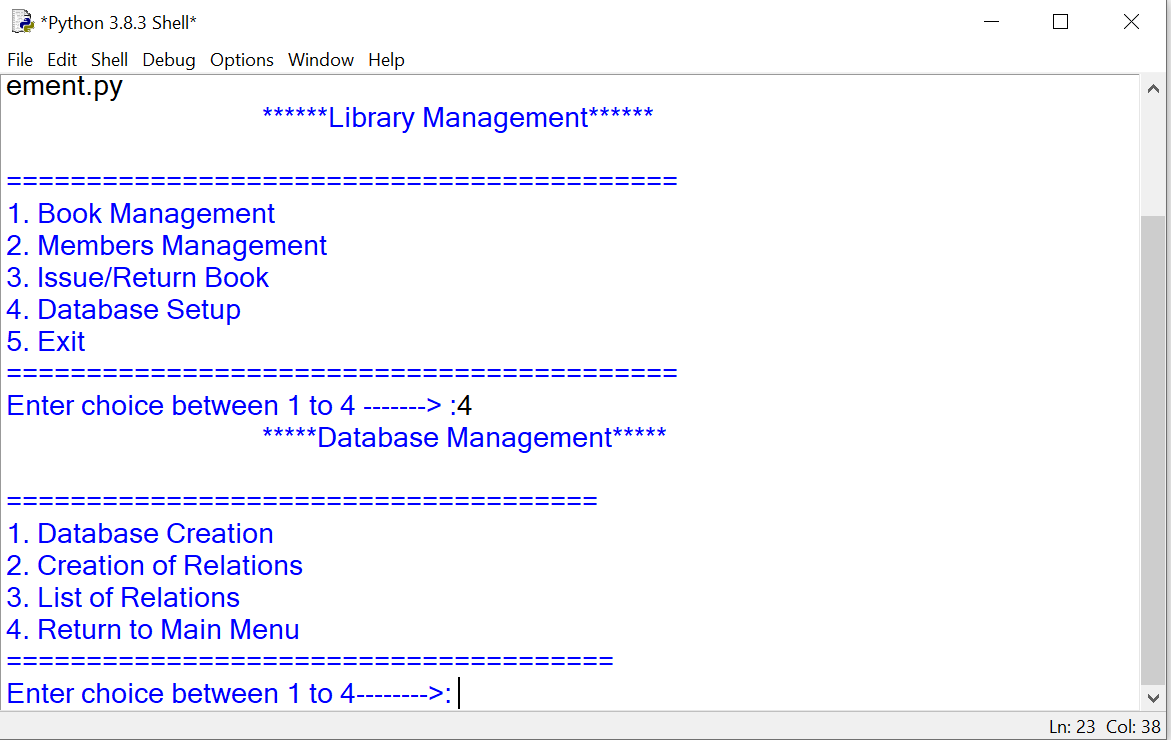
print(i)

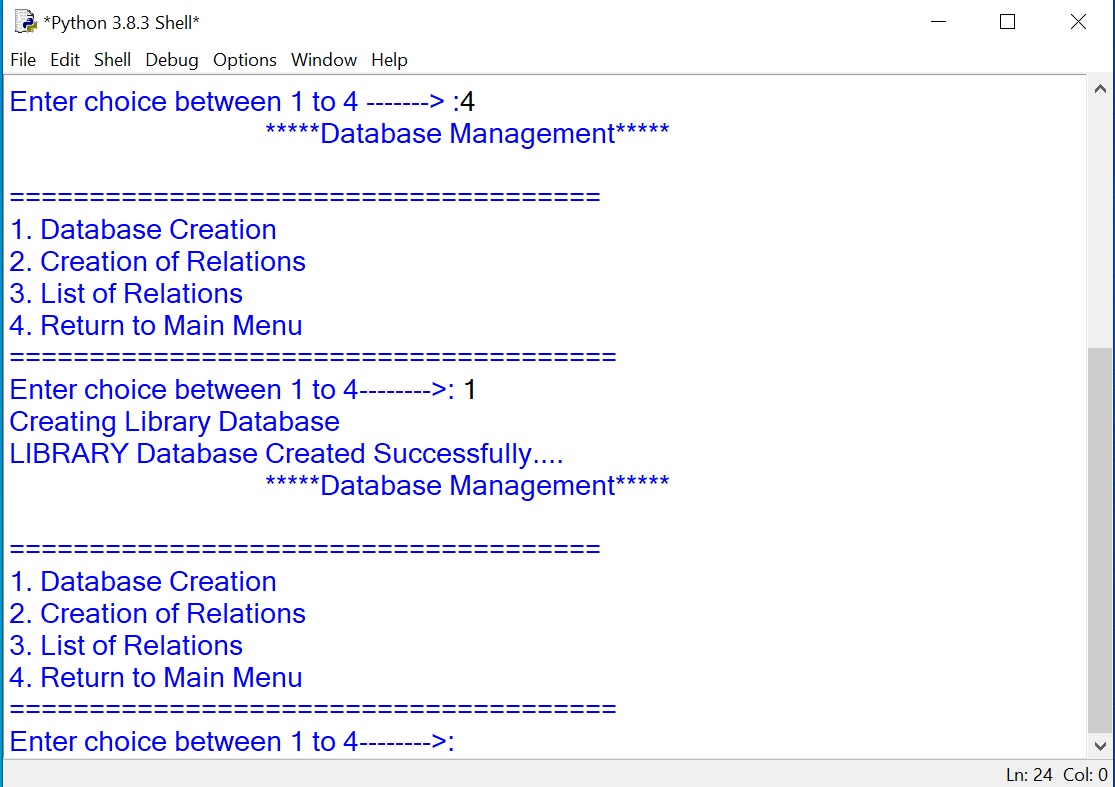
except Exception as ex:

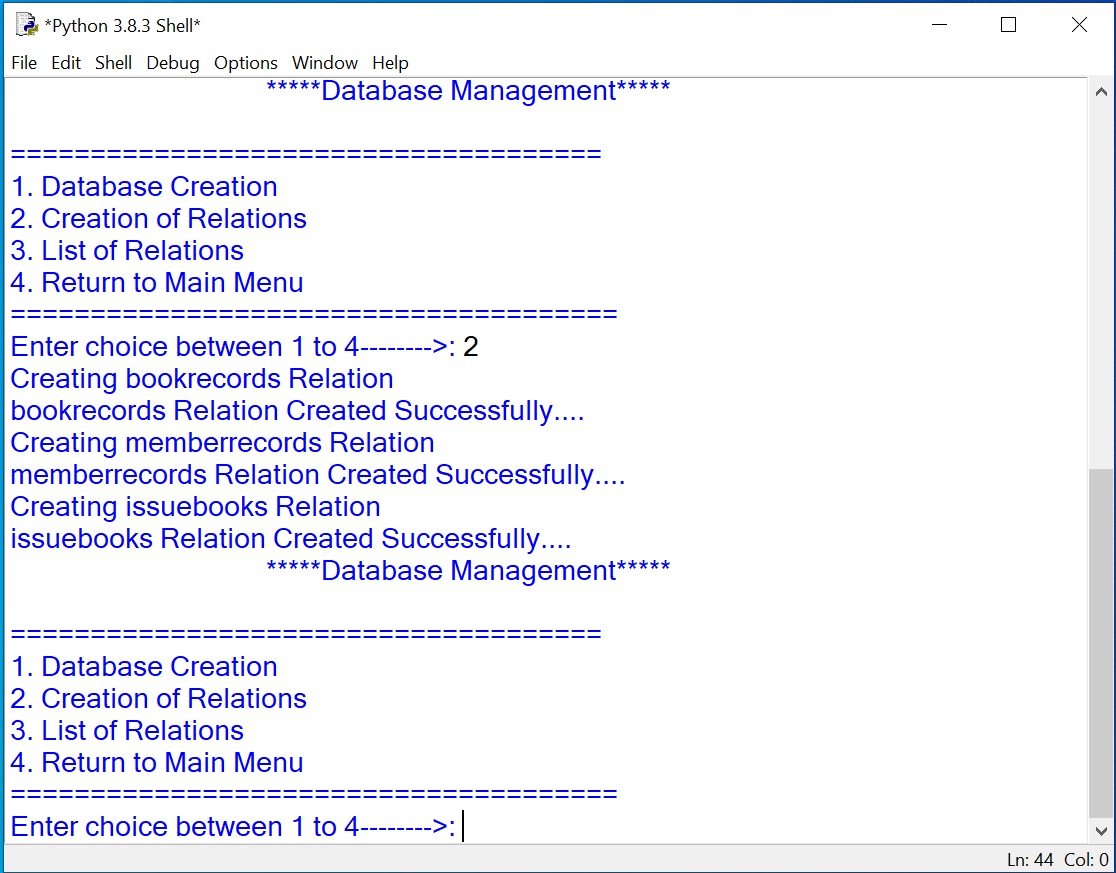
print(“Something went wrong”,ex)

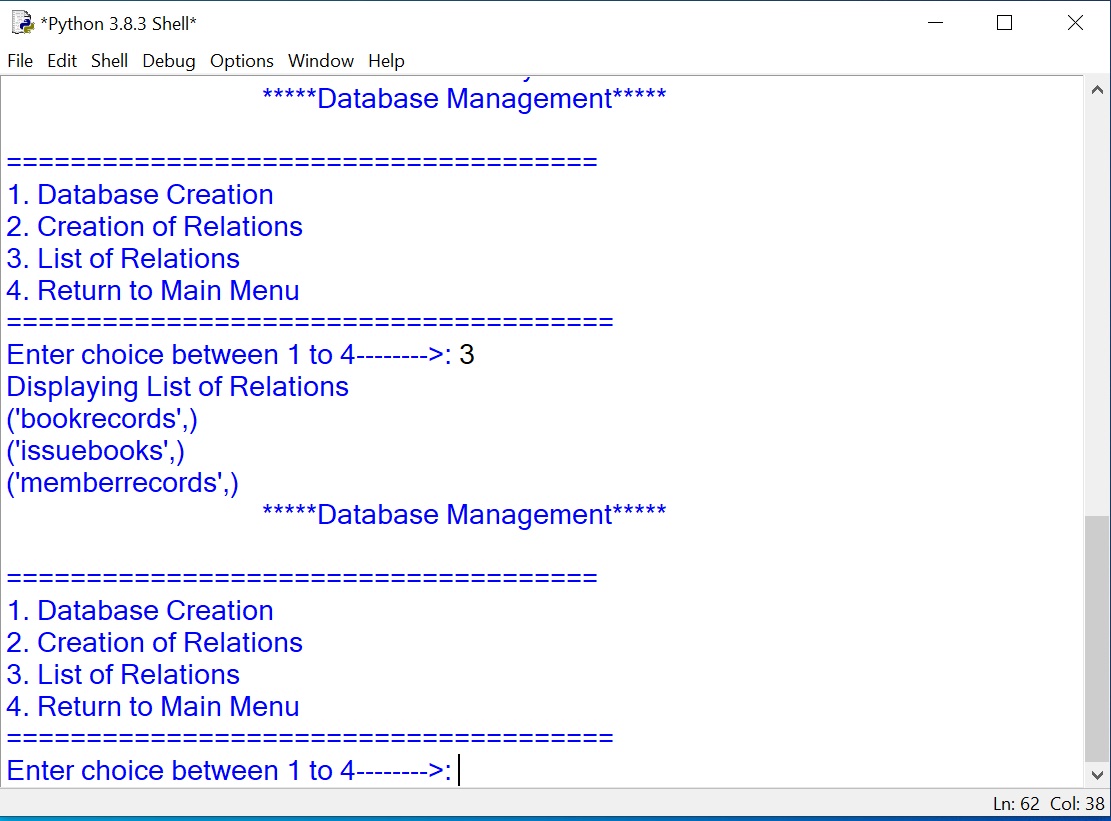
**OUTPUT**

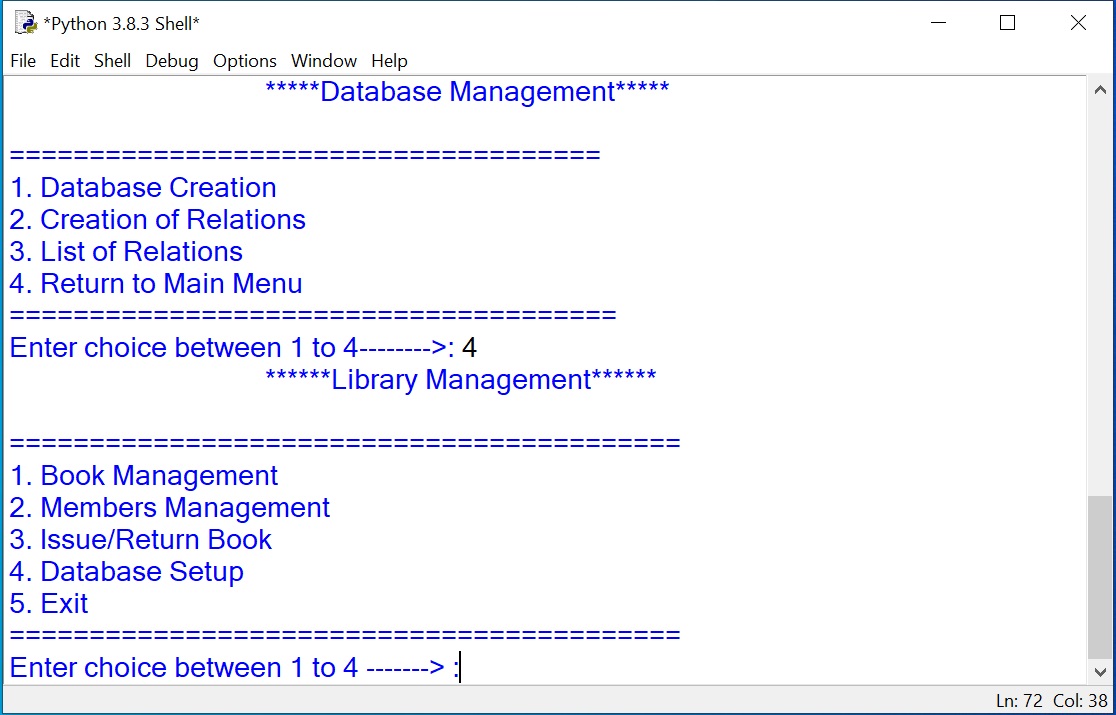


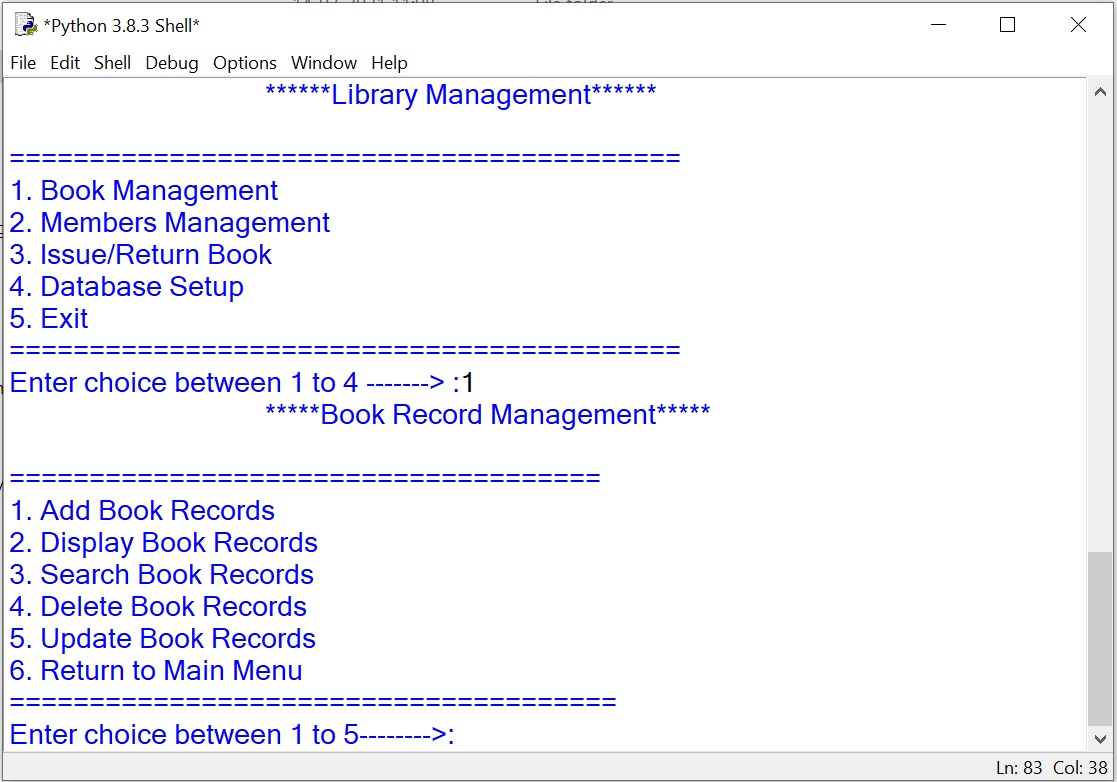






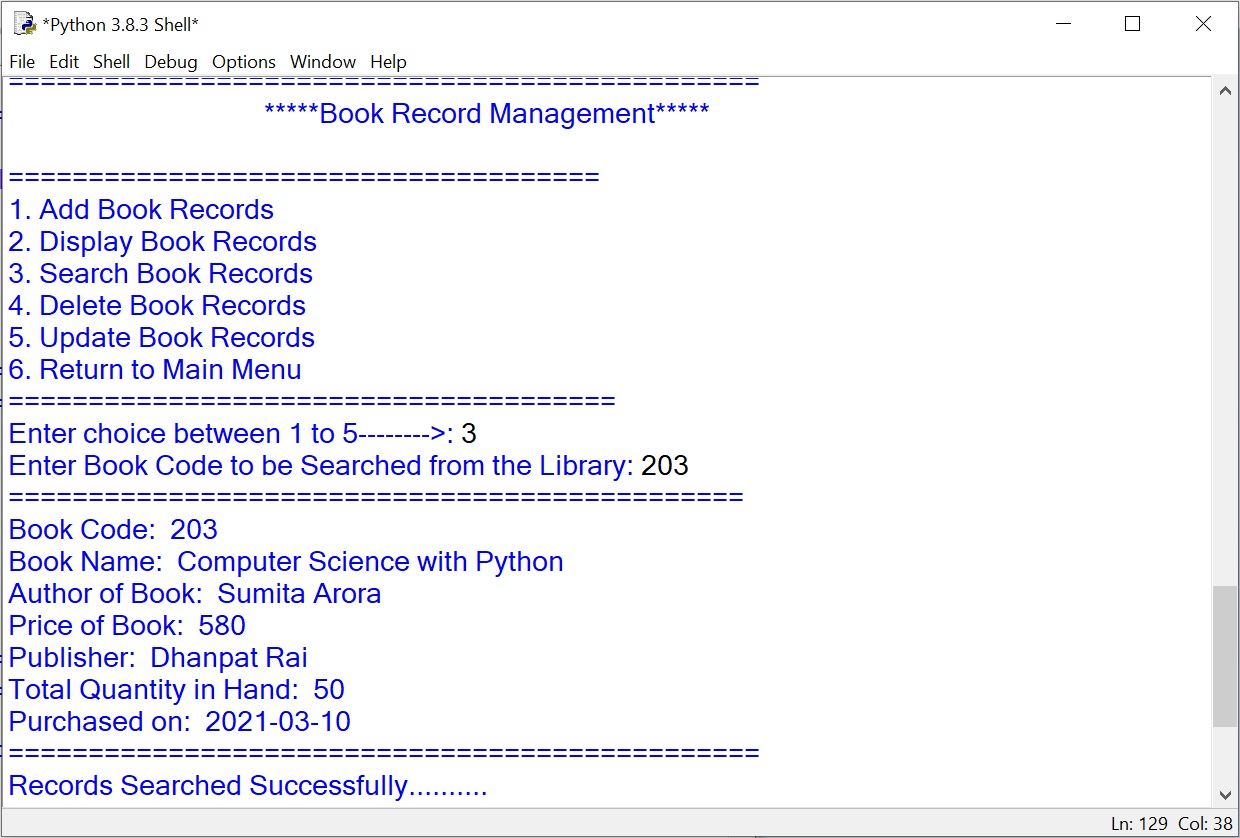


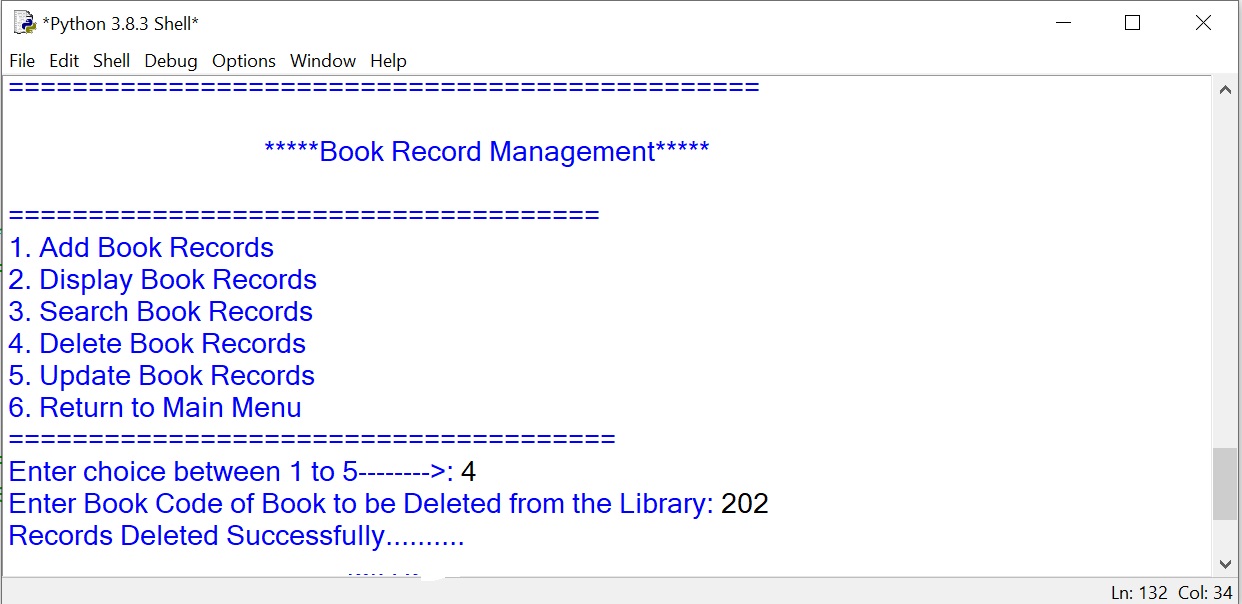


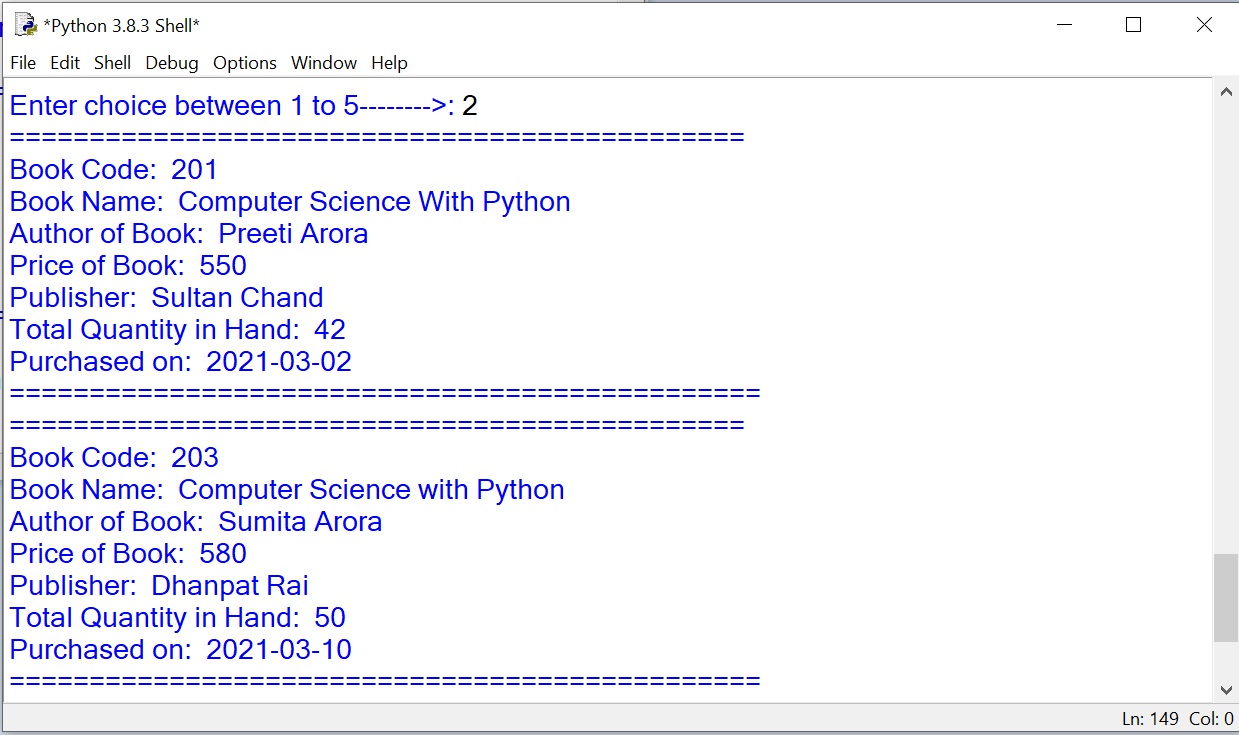


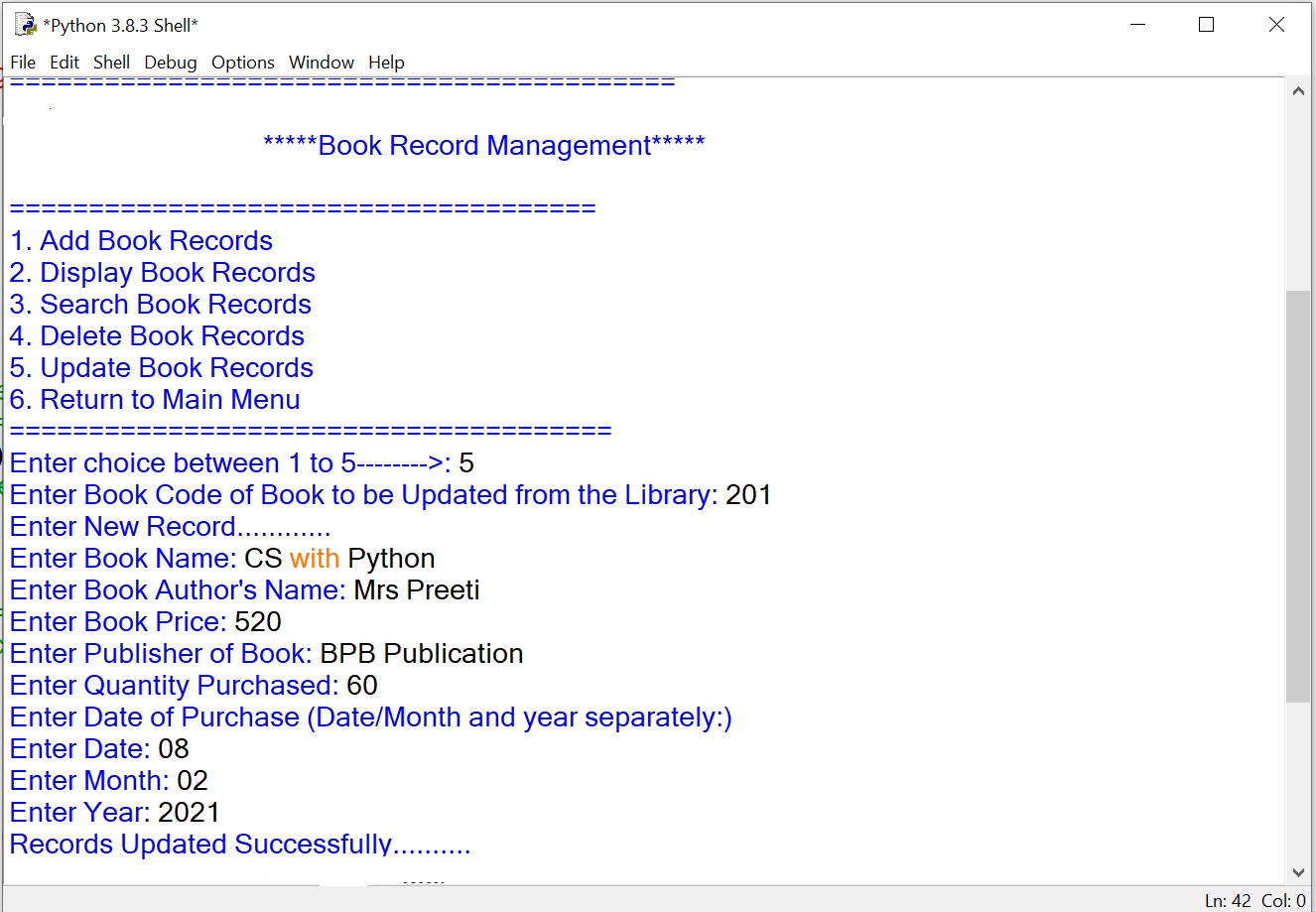


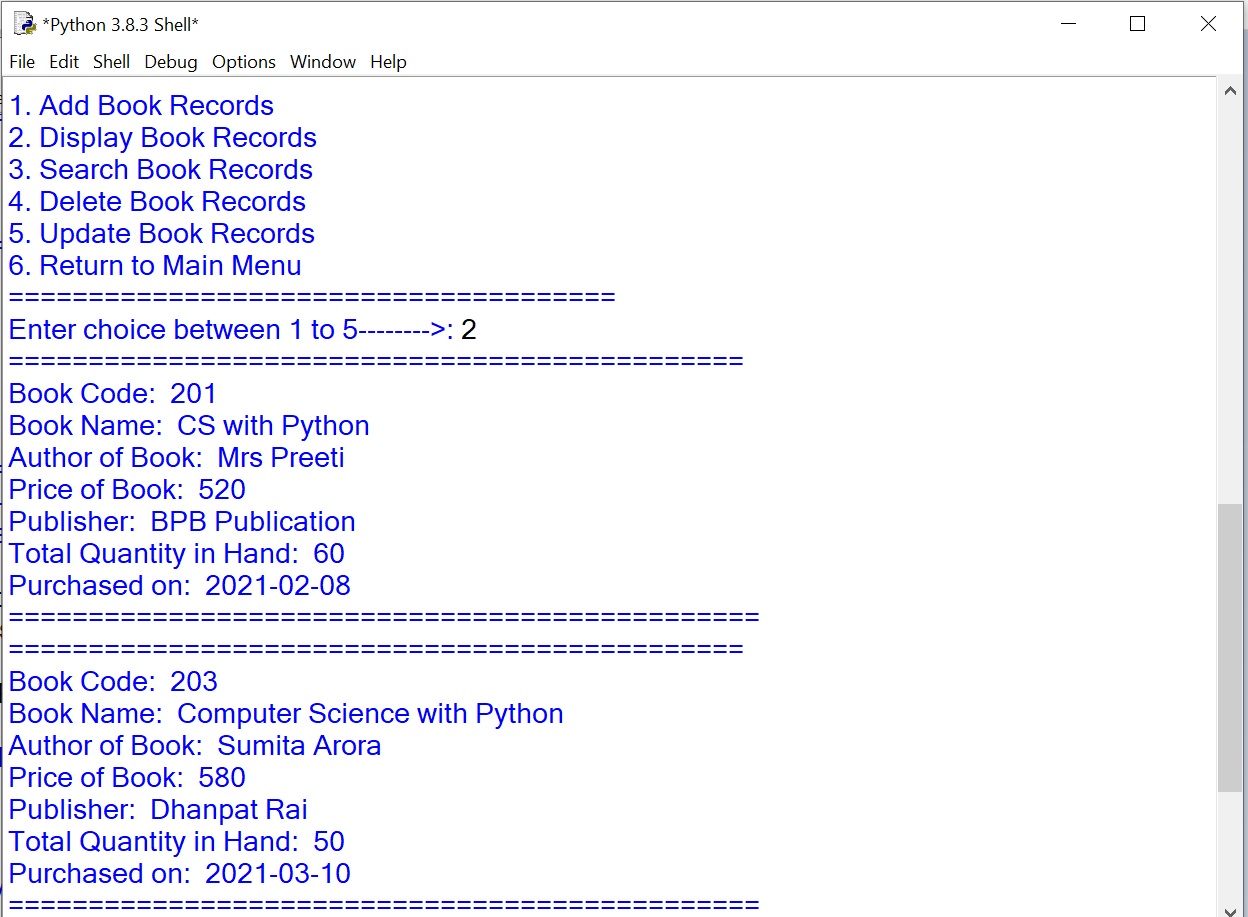




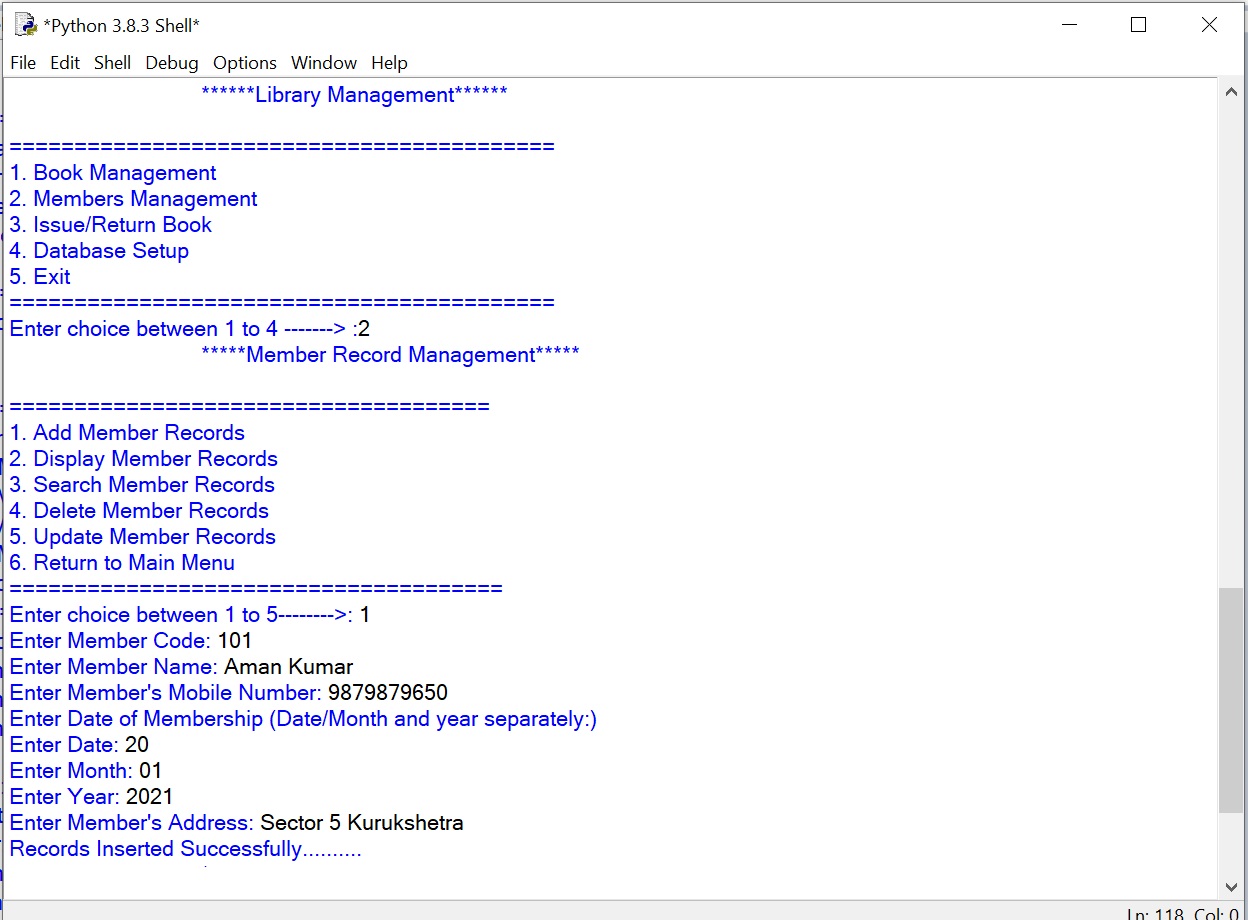


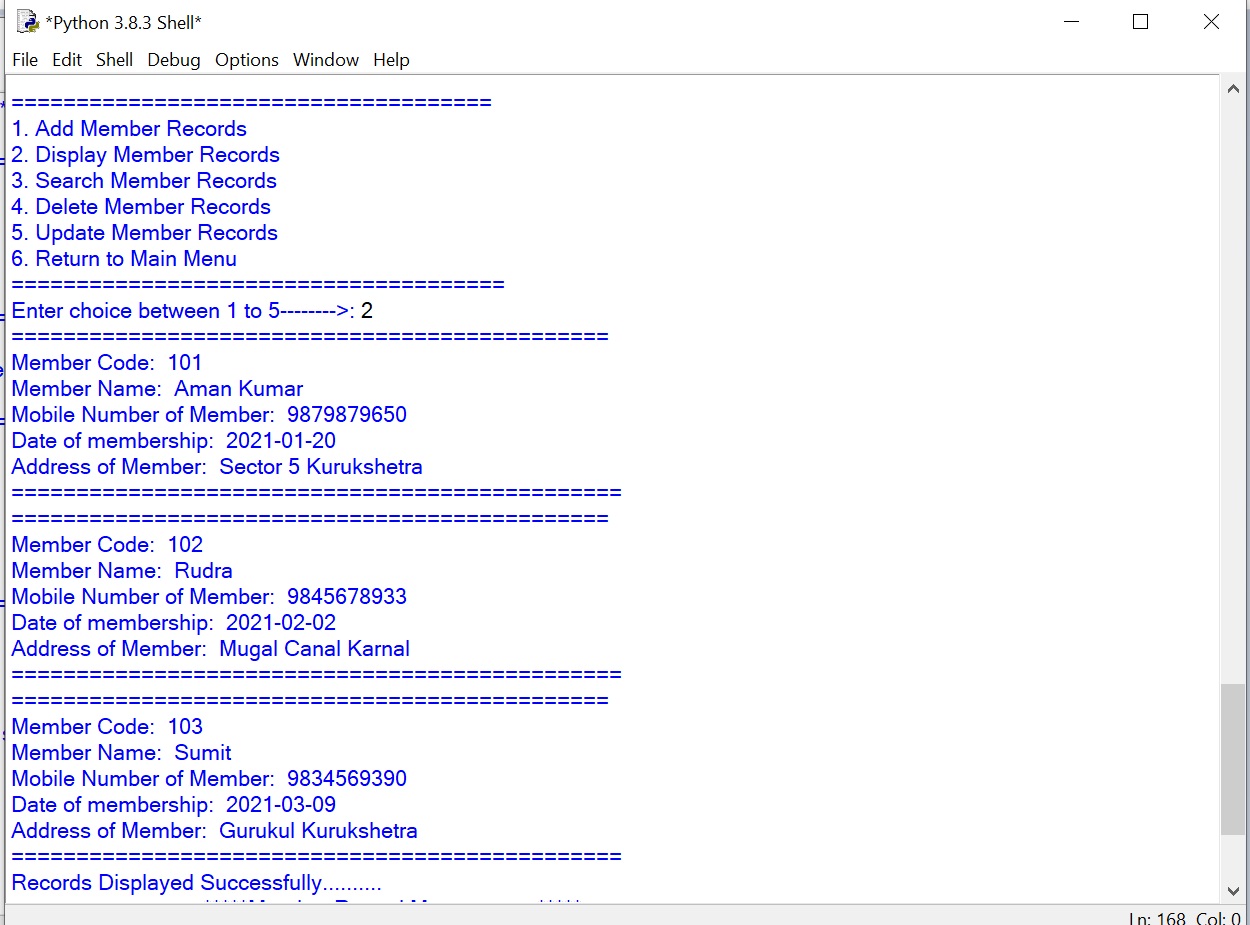




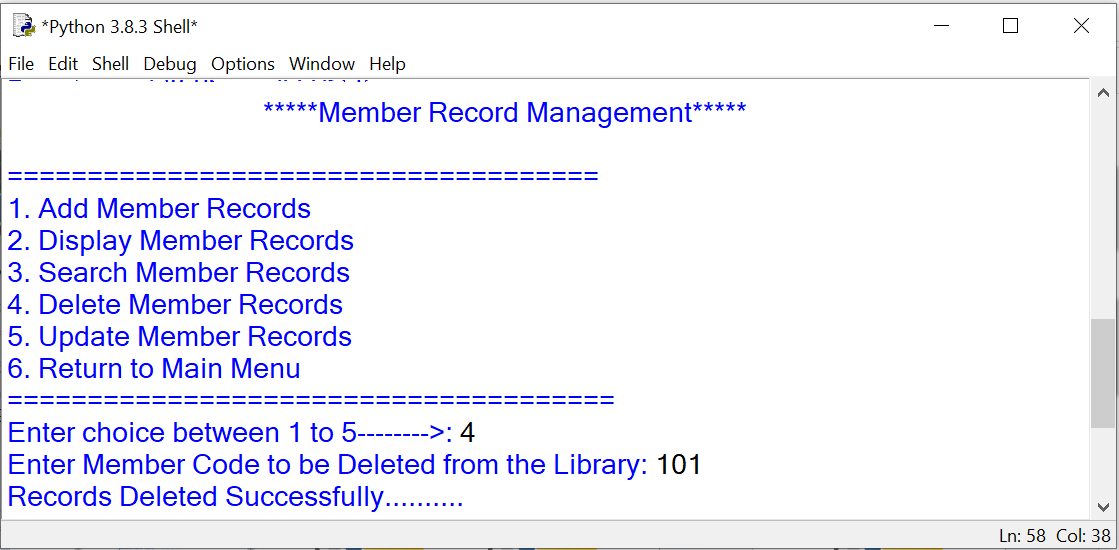


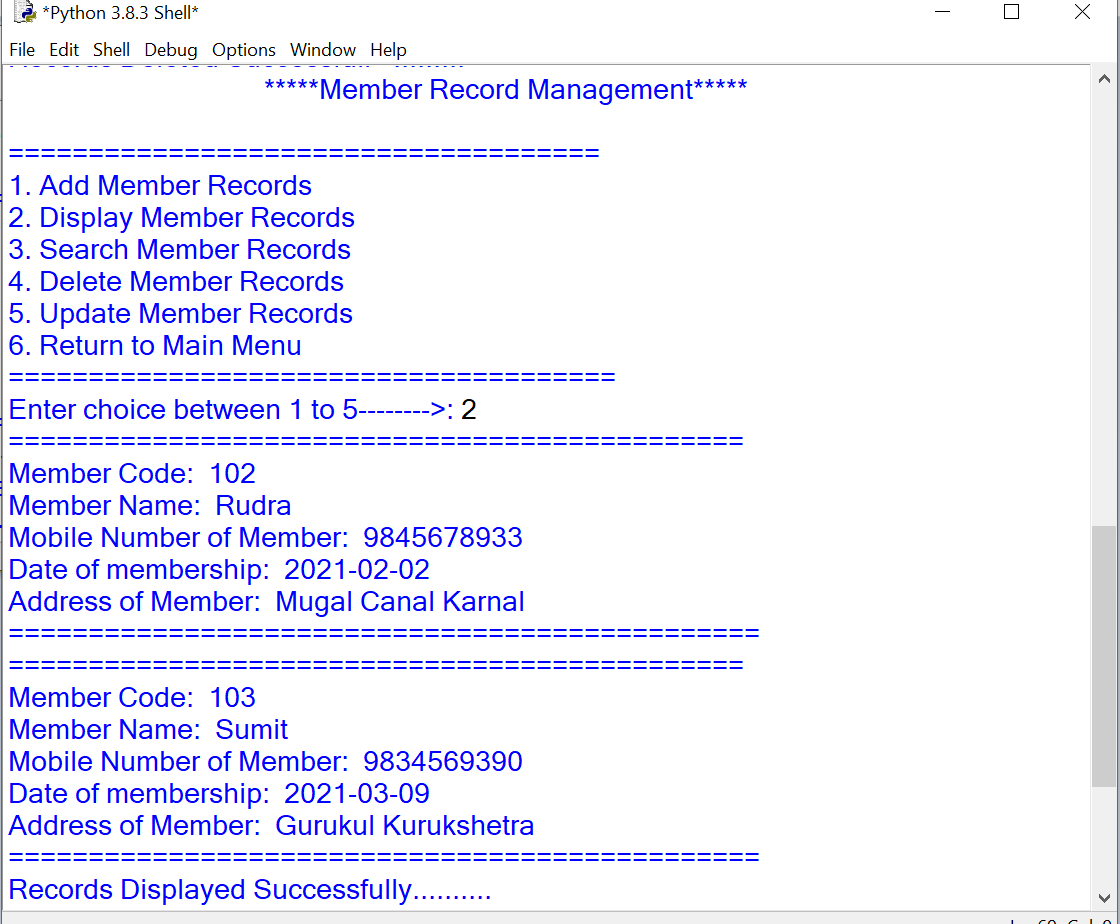


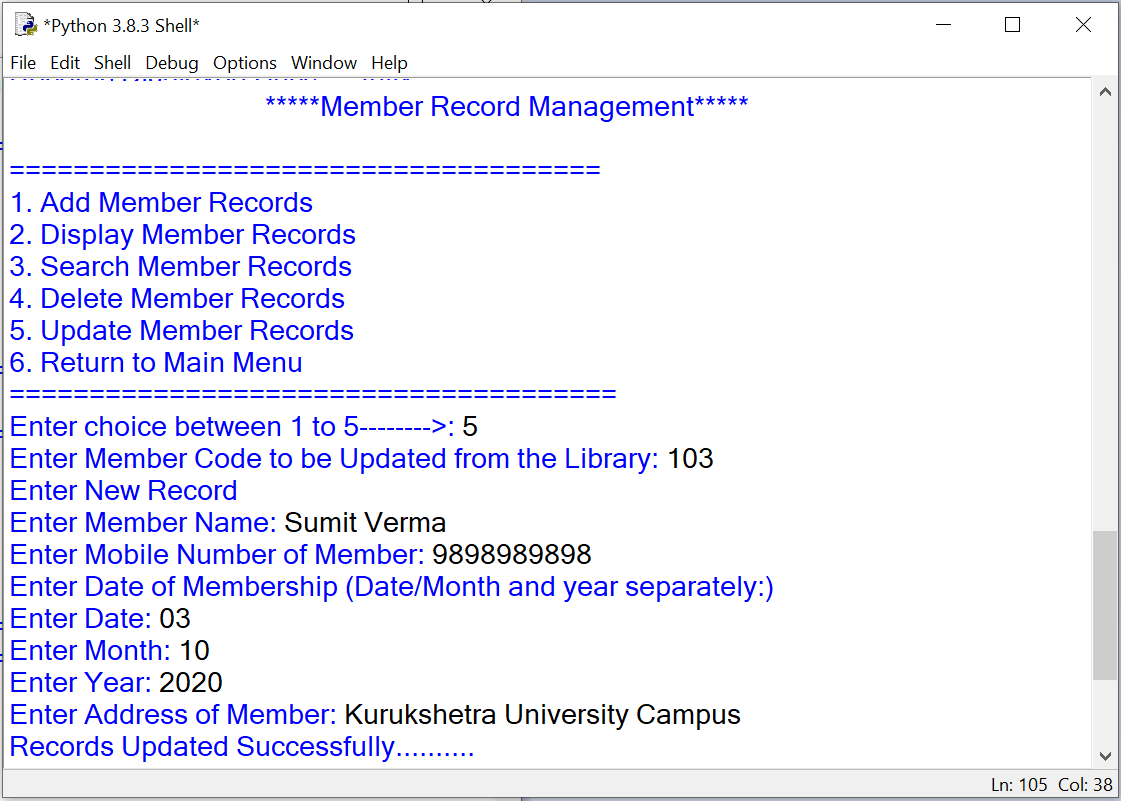


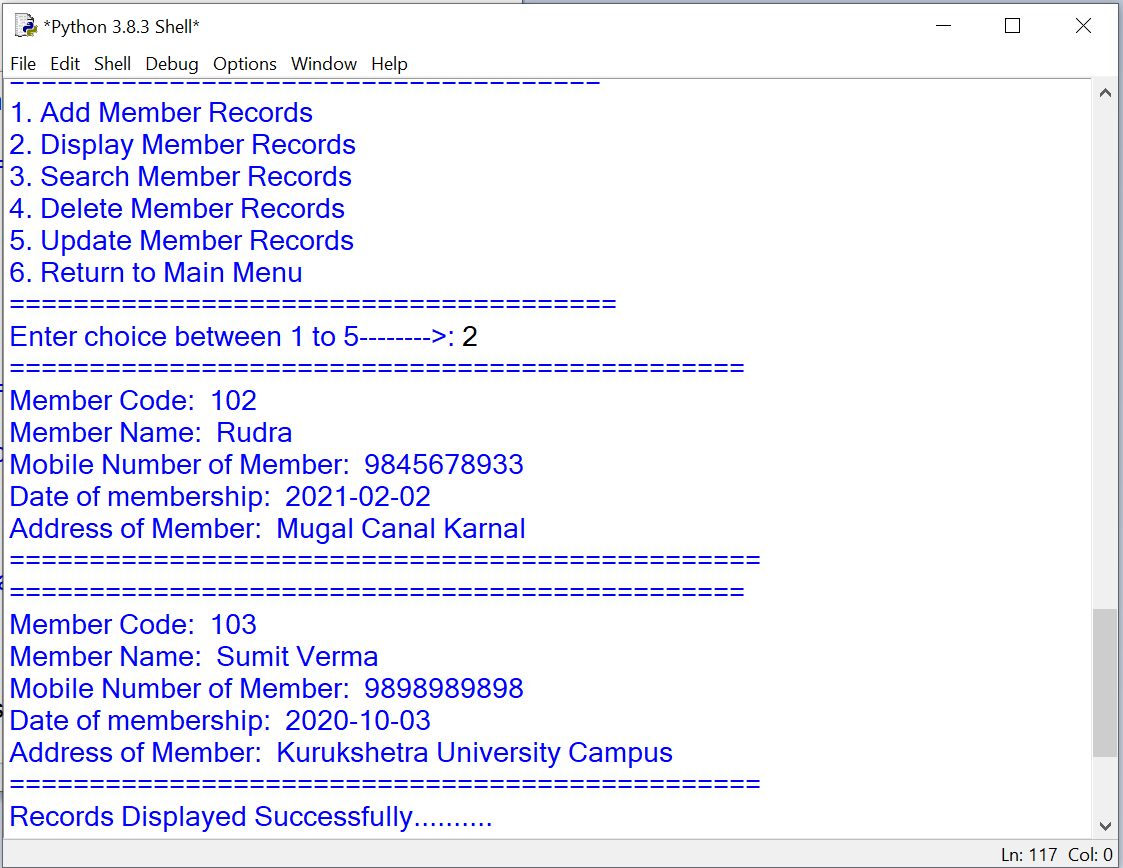


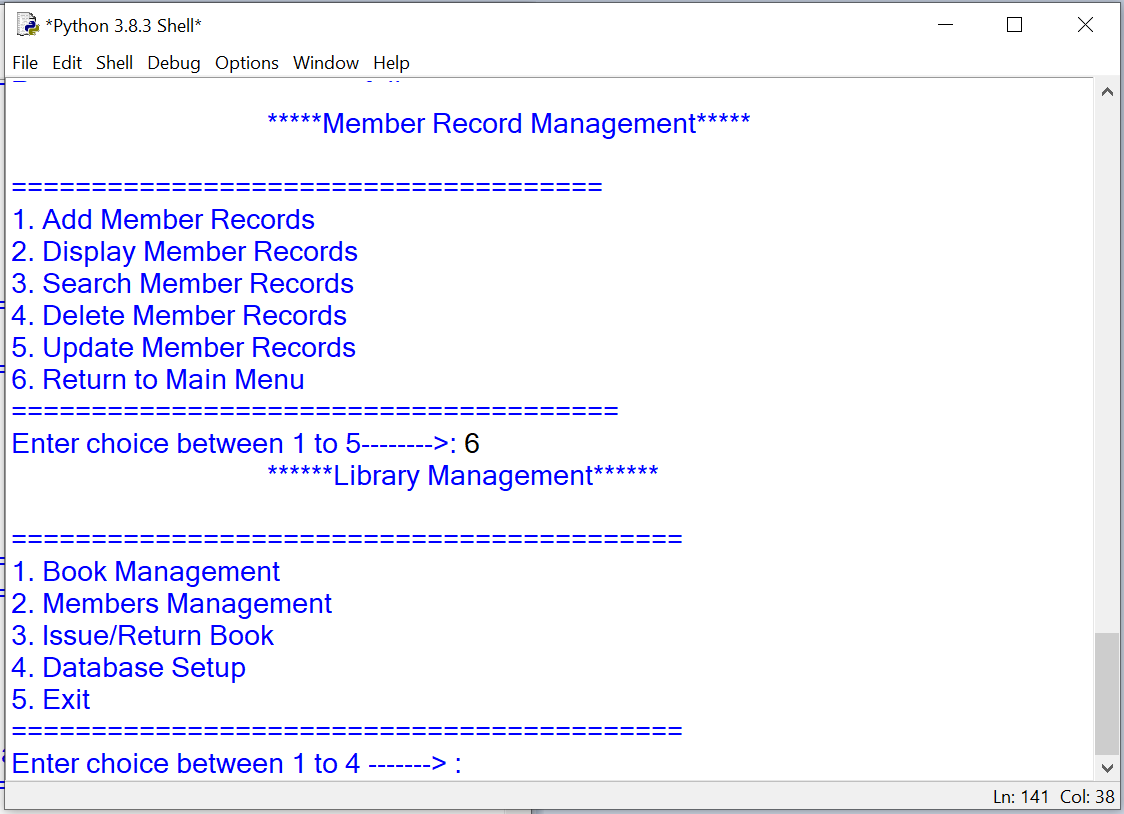




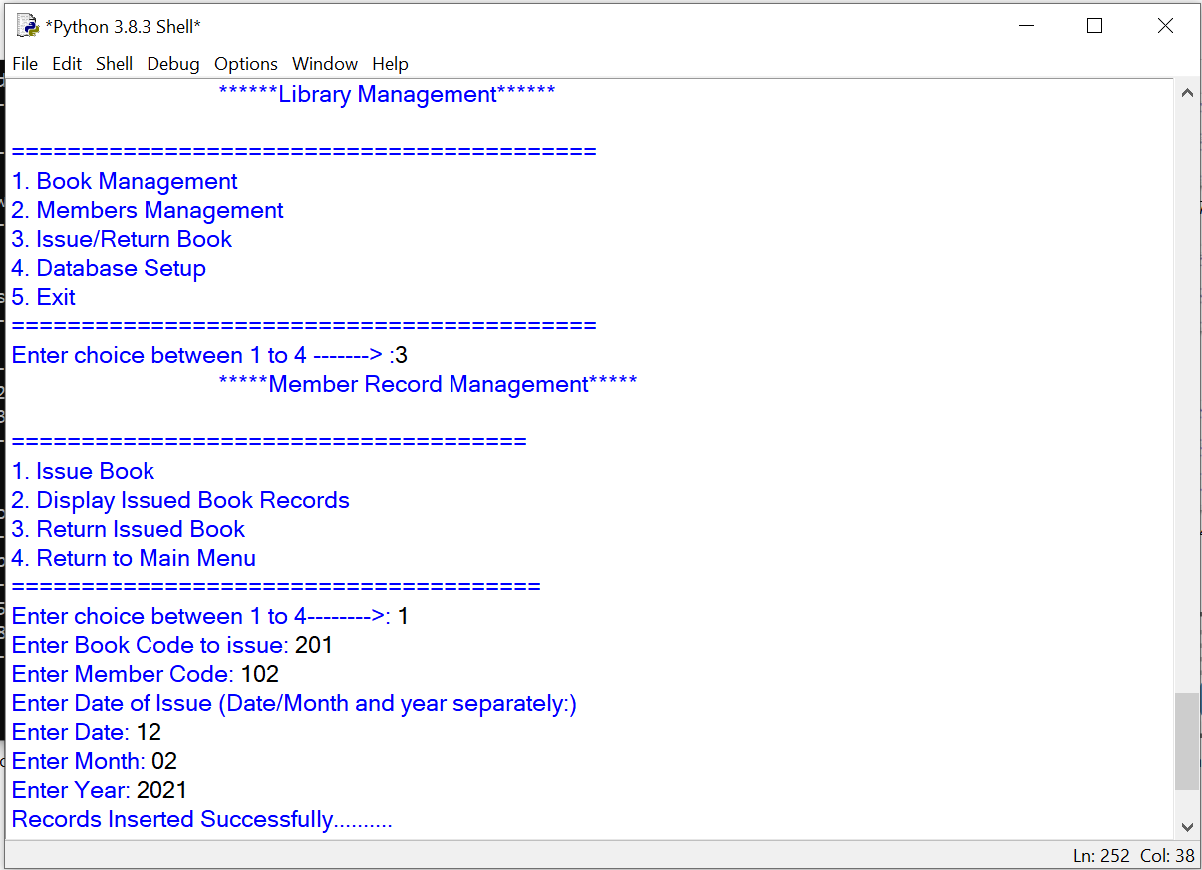


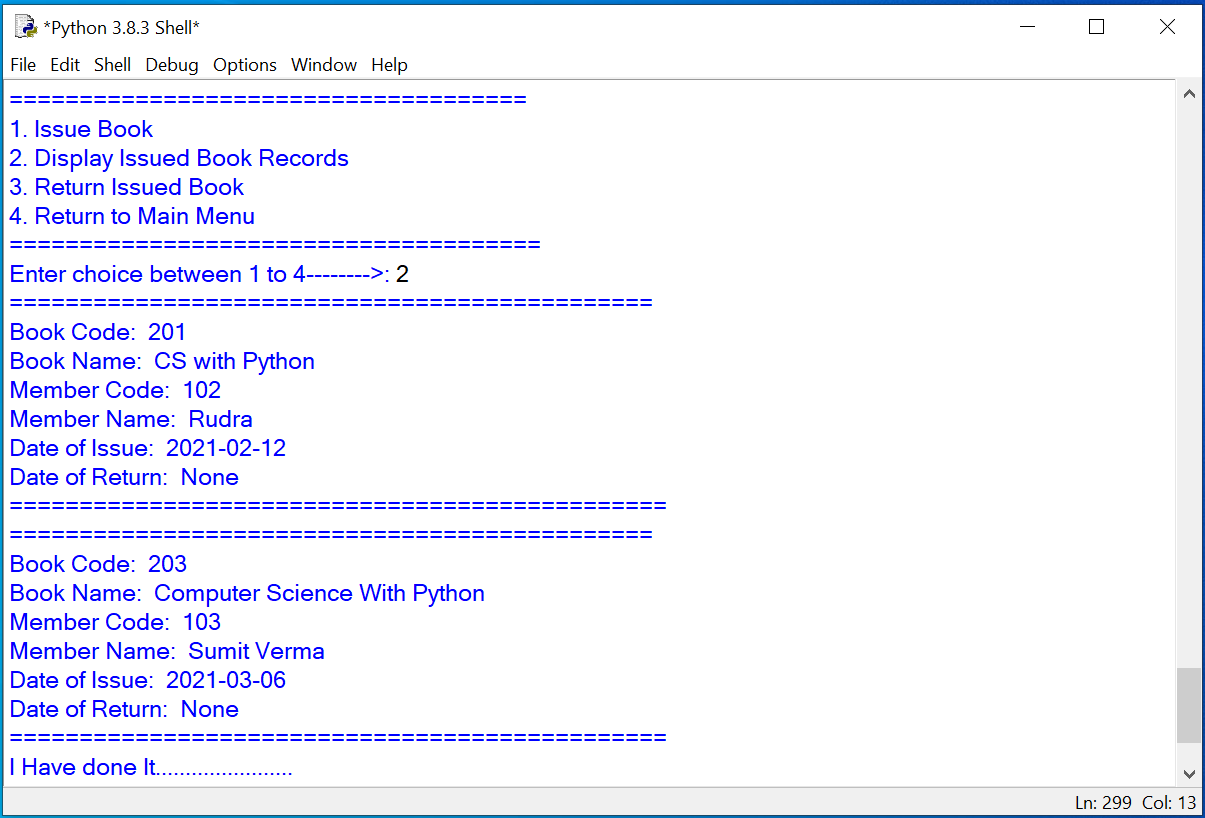


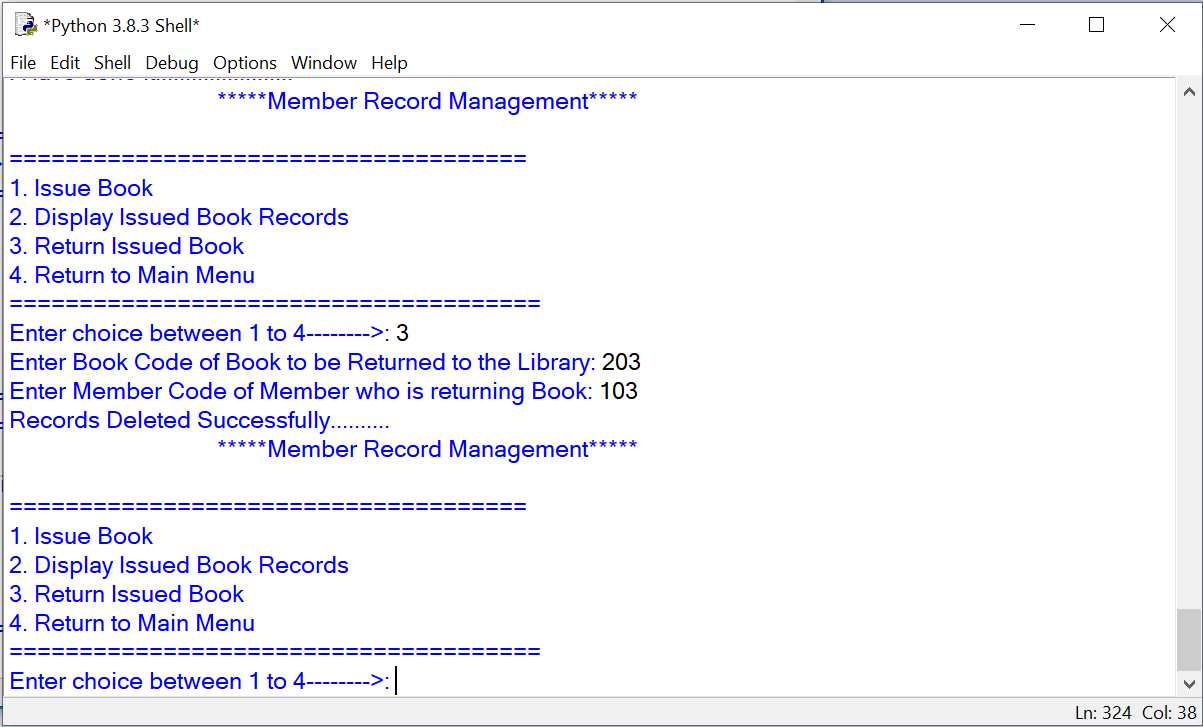




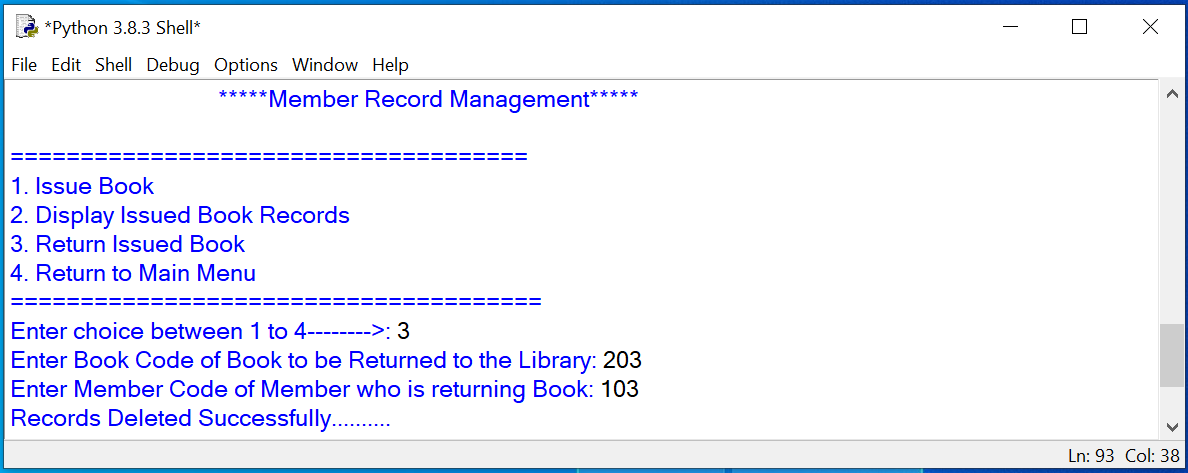


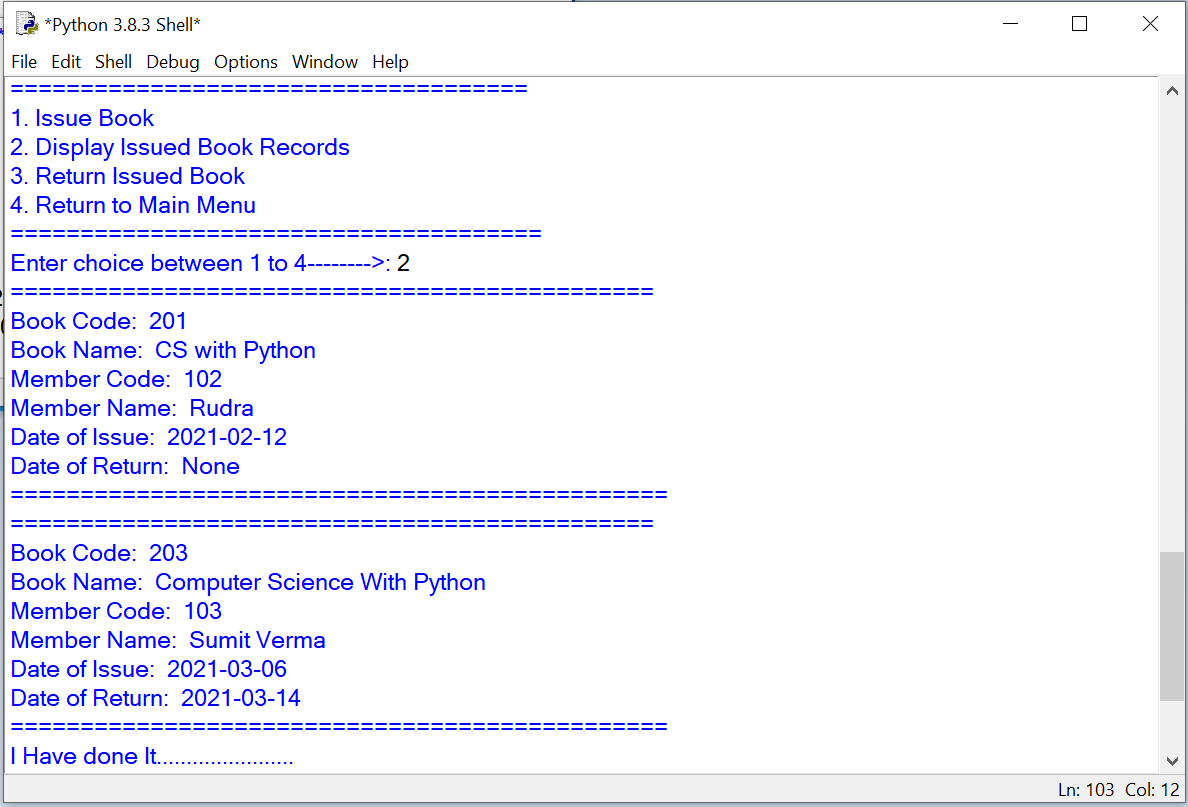


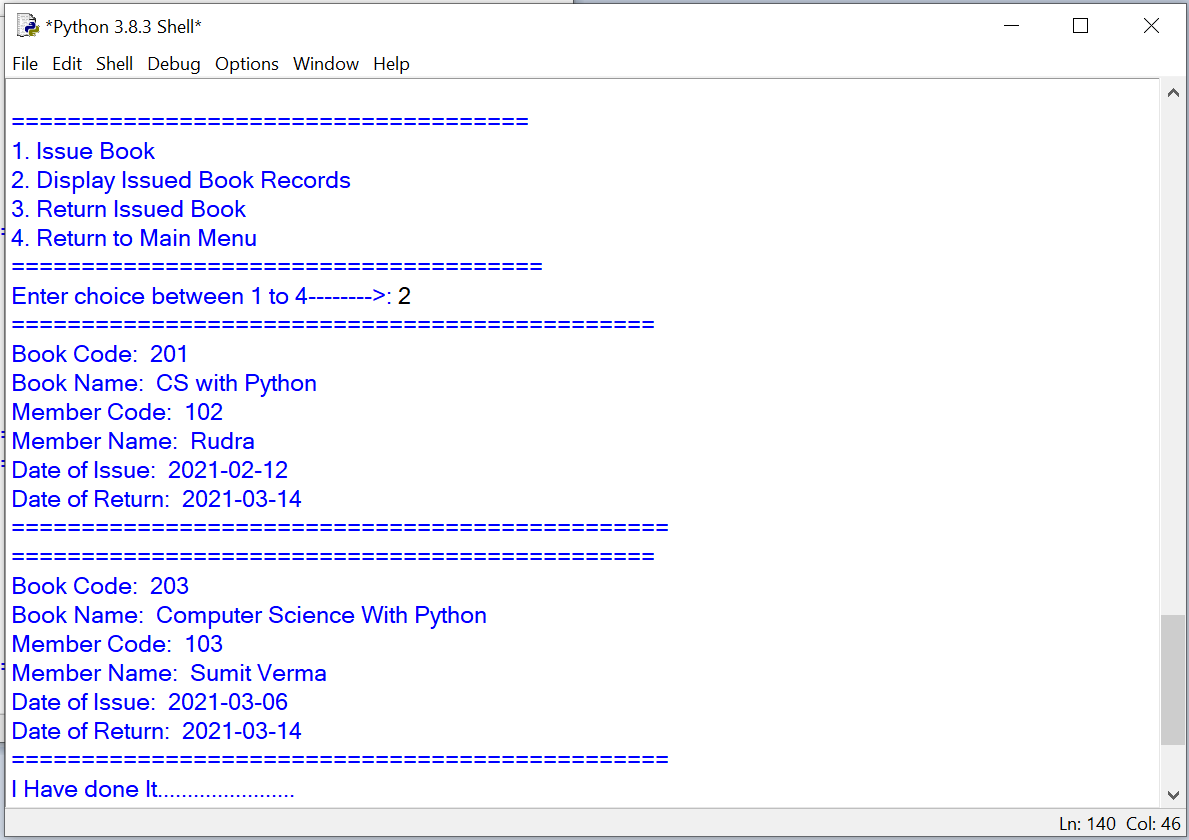


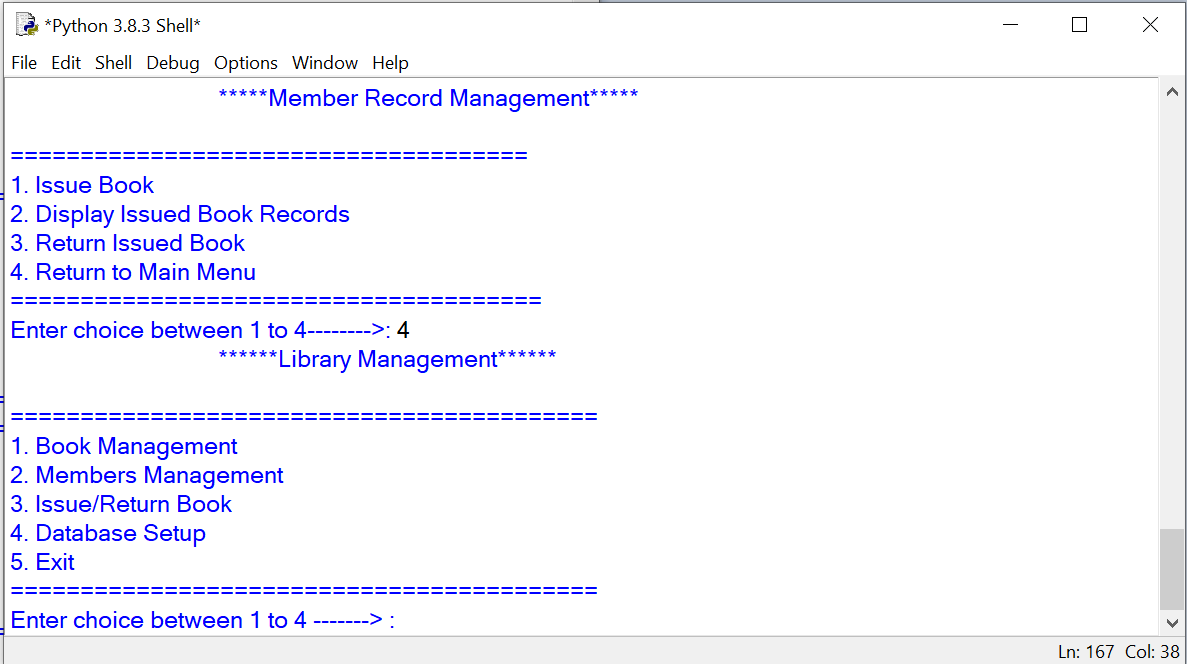












**HARDWARE AND SOFTWARE REQUIREMENTS**

I. OPERATING SYSTEM : MICROSOFT WINDOWS 10

II. PROCESSOR : DUALCORE (ANY)

III. RAM : 4 GB

IV. HARD DISK / SSD : 500 GB / 120

V. PEN DRIVE : (If Backup Required)

VI. MONITOR 14.1 or 15 -17 inch

VI. KEY BOARD AND MOUSE

VIII. PRINTER : (If Print Required – [Hard copy])

**SOFTWARE REQUIREMENTS:**

1. WINDOWS 10 OPERATING SYSTEM
2. SETUP OF PYTHON
3. MYSQL DATABASE

**BIBLIOGRAPHY**

1. ***Computer science With Python - By: Sumita Arora***
2. ***Computer science With Python - By: Preeti Arora***
3. ***Website:*** [**https://www.w3schools.com**](https://www.w3schools.com)