#Modules import

import mysql.connector as conn

from tabulate import tabulate

import datetime

#shows date and time

now=datetime.datetime.now()

print(now.strftime("%d-%m-%y %H:%M:%S"))

#connecting python to sql

mydb=conn.connect(host="localhost",user="root",passwd="passworssd",database="db")

if mydb.is\_connected():

print("Connected")

#c -> cursor

c=mydb.cursor()

#create database

c.execute("create database if not exists db")

#create tables

c.execute("create table if not exists books(BOOK\_ID varchar(20) primary key,CATEGORY VARCHAR(30),BOOK\_NAME varchar(50),AUTHOR varchar(30),STATUS varchar(30),RATING varchar(5))")

c.execute("create table if not exists books\_issued(BOOK\_ID varchar(20) primary key,BOOK\_NAME varchar(30),ISSUED\_TO varchar(30),ISSUED\_ON varchar(20),ISSUED\_TILL varchar(20),RATING varchar(5))")

#Add book details in table

def bookRegister():

bid=input("Enter the Book Id:- ")

categ=input("Enter the Category of the book:- ")

bname=input("Enter the title of the book:- ")

author=input("Enter the author of the book:- ")

status=input("Enter the status of the book(Available/Issued):- ")

rating=input("Rate the book{1-5}:- ")

t=(bid,categ,bname,author,status,rating)

y="insert into books values(%s,%s,%s,%s,%s,%s)"

c.execute(y,t)

print("Book registered sucessfully.....")

mydb.commit()

#show all books

def viewbooks():

c.execute("select \* from books")

print(tabulate(c,headers=["BOOk\_ID","CATEGORY","BOOK\_NAME","AUTHOR","STATUS","RATING"],tablefmt="fancy\_grid" ))

#delete books

def deletebook():

c.execute("select \* from books")

print(tabulate(c,headers=["BOOk\_ID","CATEGORY","BOOK\_NAME","AUTHOR","STATUS","RATING"],tablefmt="fancy\_grid" ))

bid=input("Enter the book Id:")

try:

q="delete from books where book\_id='{}'".format(bid)

c.execute(q)

print("Book deleted successfully.....")

mydb.commit()

except Exception as e:

print("Error",e)

#issue books to the user

def IssueBook():

c.execute("select \* from books")

print(tabulate(c,headers=["BOOk\_ID","CATEGORY","BOOK\_NAME","AUTHOR","STATUS","RATING"],tablefmt="fancy\_grid" ))

bid=input("Enter the book Id:- ")

issuedto=input("Enter your name:- ")

bname=input("Enter the book name:- ")

issuedon=input("Enter the issue date:- ")

issuedtill=input("Enter the return date:- ")

rate="null"

t=[bid,issuedto,bname,issuedon,issuedtill,rate]

y="insert into books\_issued values(%s,%s,%s,%s,%s,%s)"

c.execute(y,t)

mydb.commit()

try:

b="Issued"

q="update books set status='{}' where book\_id='{}'".format(b,bid)

c.execute(q)

mydb.commit()

print("Book Issued Successfully")

except Exception as e:

print("Error",e)

#return the book by the user

def ReturnBook():

c.execute("select \* from books\_issued")

print(tabulate(c,headers=["BOOk\_ID","BOOK\_NAME","ISSUED\_TO","ISSUED\_ON","ISSUED\_TILL","RATING"],tablefmt="fancy\_grid" ))

bid=input("Enter the book Id:- ")

rate=input("Rate the book{1-5}:- ")

try:

b="Available"

q0="delete from books\_issued where book\_id='{}'".format(bid)

c.execute(q0)

q="update books set status='{}' where book\_id='{}'".format(b,bid)

c.execute(q)

q1="update books set rating='{}' where book\_id='{}'".format(rate,bid)

c.execute(q1)

mydb.commit()

except Exception as e:

print("Error",e)

print("Book Returned Successfully")

#modify book details

def Modifybook():

c.execute("select \* from books")

print(tabulate(c,headers=["BOOk\_ID","CATEGORY","BOOK\_NAME","AUTHOR","STATUS","RATING"],tablefmt="fancy\_grid" ))

bid=input("Enter the Book Id:- ")

try:

s="select \* from books where BOOK\_ID="+bid

c.execute(s)

print(tabulate(c, headers=["BOOK\_ID","CATEGORY","BOOK\_NAME","AUTHOR","STATUS","RATING"], tablefmt='fancy\_grid'))

print("\n \n Type the value to modify or just press enter for no change")

x=input("Modify Category:- ")

if len(x)>0:

categ=x

s="update books set category=" + "'"+categ+"'" + "where Book\_ID="+bid

c.execute(s)

mydb.commit()

x=input("Modify Book Name:- ")

if len(x)>0:

bname=x

s="update books set BOOK\_NAME=" + "'"+bname+"'" + "where BOOK\_ID="+bid

c.execute(s)

mydb.commit()

x=input("Modify Author:- ")

if len(x)>0:

bauthor=x

s="update books set AUTHOR=" + "'"+bauthor+"'" + "where BOOK\_ID="+bid

c.execute(s)

mydb.commit()

print("Record updated successfully:")

except Exception as e:

print("Error: ",e)

#show all the categories of the books

def category():

c.execute("select category,count(\*) as 'NO. OF BOOKS' from books group by category")

print(tabulate(c,headers=["CATEGORY","NO. OF BOOKS"],tablefmt="fancy\_grid"))

x=input("Enter the category to see the books(Name should be in lower case):- ")

if x=="novel":

q="select \* from books where category='{}'".format(x)

c.execute(q)

print(tabulate(c,headers=["BOOk\_ID","CATEGORY","BOOK\_NAME","AUTHOR","STATUS","RATING"],tablefmt="fancy\_grid" ))

elif x=="biography":

q="select \* from books where category='{}'".format(x)

c.execute(q)

print(tabulate(c,headers=["BOOk\_ID","CATEGORY","BOOK\_NAME","AUTHOR","STATUS","RATING"],tablefmt="fancy\_grid" ))

elif x=="subject\_books":

q="select \* from books where category='{}'".format(x)

c.execute(q)

print(tabulate(c,headers=["BOOk\_ID","CATEGORY","BOOK\_NAME","AUTHOR","STATUS","RATING"],tablefmt="fancy\_grid" ))

elif x=="manga":

q="select \* from books where category='{}'".format(x)

c.execute(q)

print(tabulate(c,headers=["BOOk\_ID","CATEGORY","BOOK\_NAME","AUTHOR","STATUS","RATING"],tablefmt="fancy\_grid" ))

elif x=="scifi":

q="select \* from books where category='{}'".format(x)

c.execute(q)

print(tabulate(c,headers=["BOOk\_ID","CATEGORY","BOOK\_NAME","AUTHOR","STATUS","RATING"],tablefmt="fancy\_grid" ))

elif x=="encyclopedia":

q="select \* from books where category='{}'".format(x)

c.execute(q)

print(tabulate(c,headers=["BOOk\_ID","CATEGORY","BOOK\_NAME","AUTHOR","STATUS","RATING"],tablefmt="fancy\_grid" ))

#seaches the book

def search():

x=input("enter the book name:- ")

q="select \* from books where book\_name='{}'".format(x)

c.execute(q)

print(tabulate(c,headers=["BOOk\_ID","CATEGORY","BOOK\_NAME","AUTHOR","STATUS","RATING"],tablefmt="fancy\_grid" ))

#shows about the pragram

def about():

print("A library management system keeps track of the books present in the library. \

It is an important piece of software which is a must at schools and colleges\

Libraryis a collection of sources of information and similar resources, made accessible\

to adefined community for reference or borrowing.Thus the process of handling a librarymanually \

is very troublesome and clumsy.As regards to thispoint of view, the computerizedsystem for \

handling the activities of library management provides a comprehensive way tolessen physical labour,\

to reduce complexity of the manual system and soon.This projectwork aim to design and implement a\

computerized library management system.\

\nSoftwares used:\n1. Python\n2. My Sql\nDeveloped by:\nAnubhav Singh and Anuj Shrivastava\nClass XII(A)")

#Main

while(True):

welcomeMsg = '''\n =============== WELCOME TO CENTRAL LIBRARY ==================

Please choose an option:

1. Add Book Details

2. Delete Book

3. View Book List

4. Issue Book To A Student

5. Return Book

6. Modify Book Details

7. Show Category Wise Bookss

8. Search Book

9. About

10. Exit Program

\n=================================================================

'''

print(welcomeMsg)

x = input("Enter a choice: ")

if x=="1":

bookRegister()

elif x=="2":

deletebook()

elif x=="3":

viewbooks()

elif x=="4":

IssueBook()

elif x=="5":

ReturnBook()

elif x=="6":

Modifybook()

elif x=="7":

category()

elif x=="8":

search()

elif x=="9":

about()

elif x=="10":

print("Thank You For Using")

print("Developed and designed by Anuj Shrivastava & Anubhav Singh")

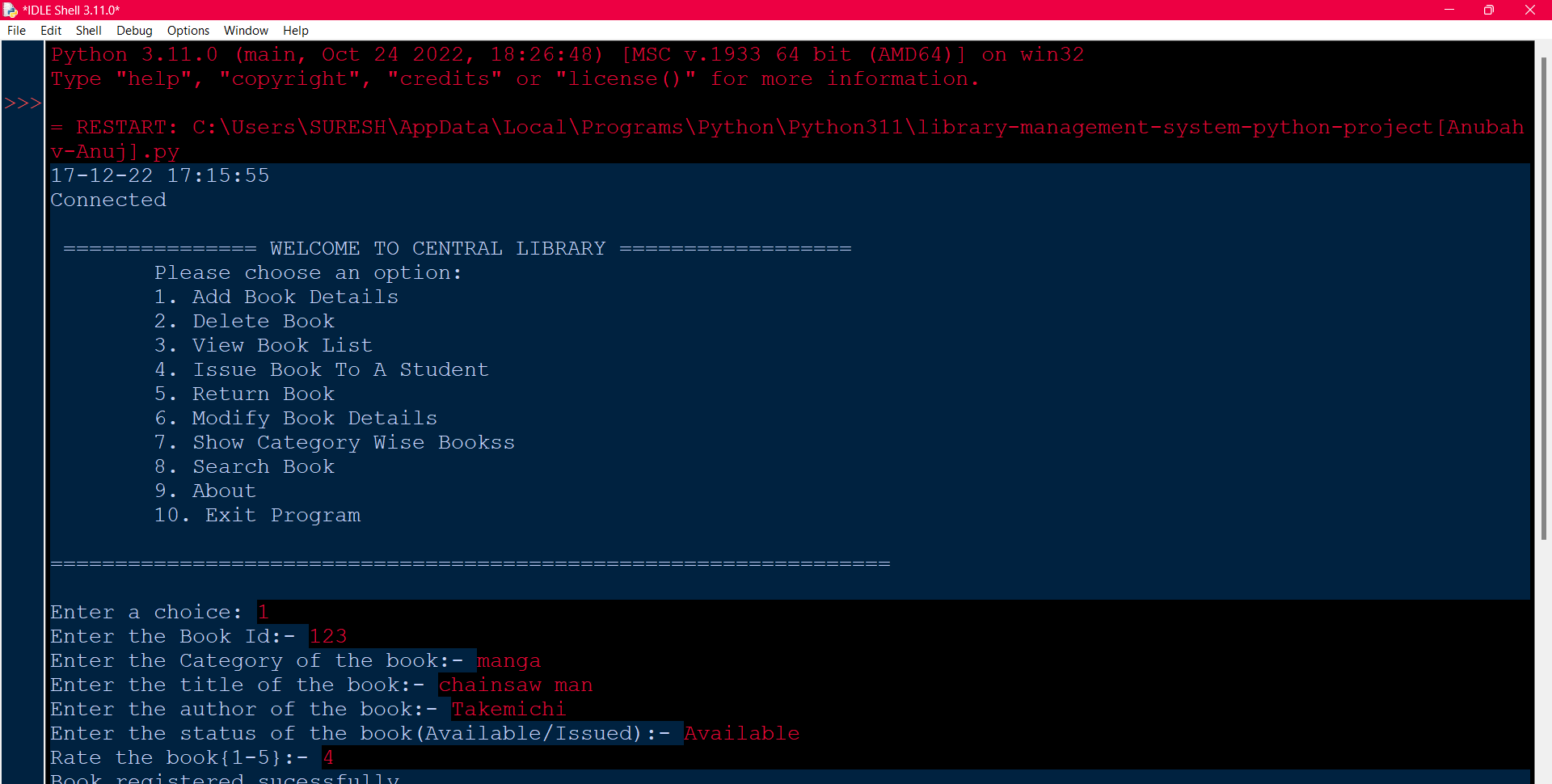
quit()

else:

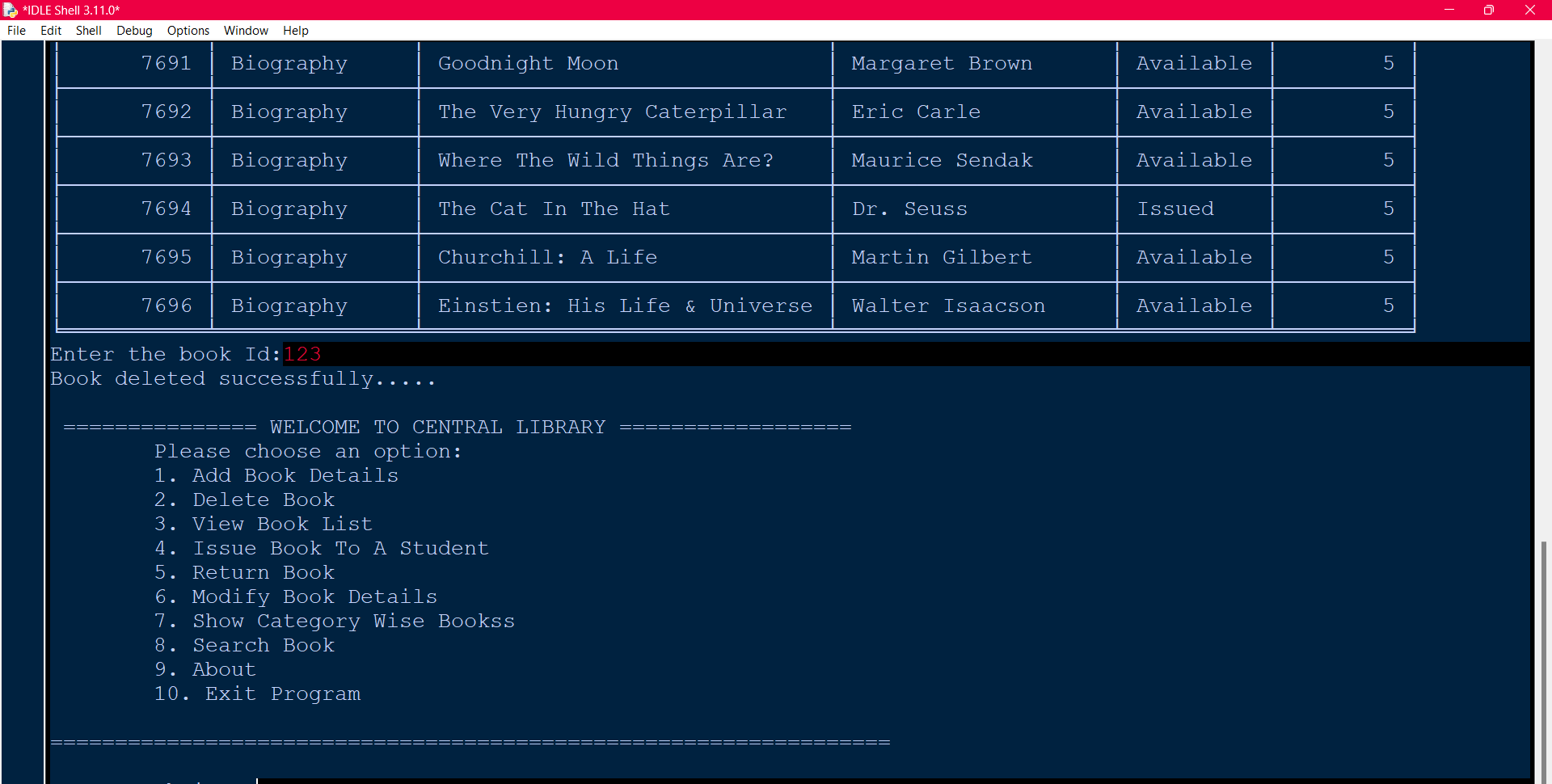
print("PLEASE INPUT A VALID CHOICE....!!")

**OUTPUTS**

**1.**

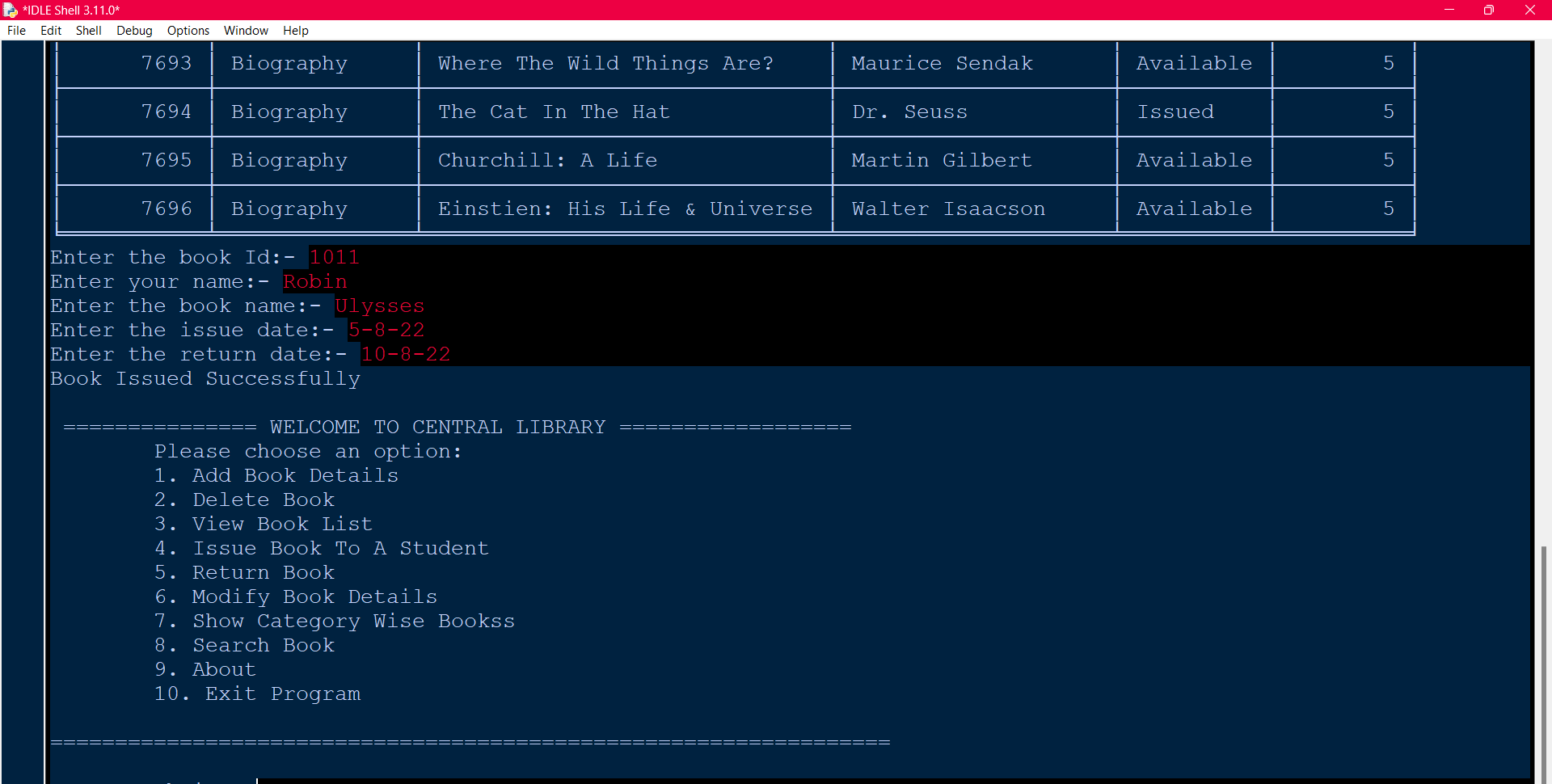
****

**2.**

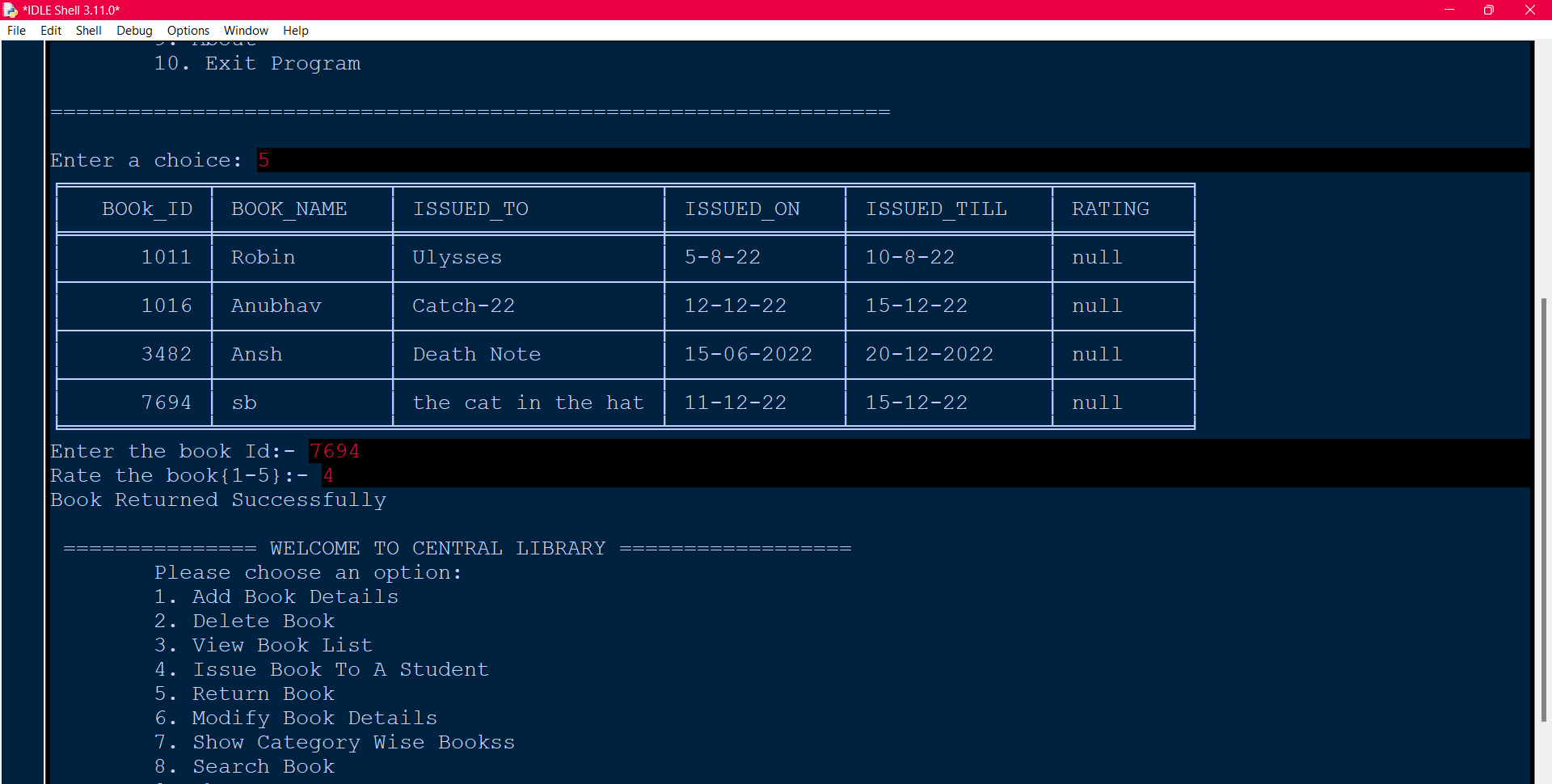
****

**3.**

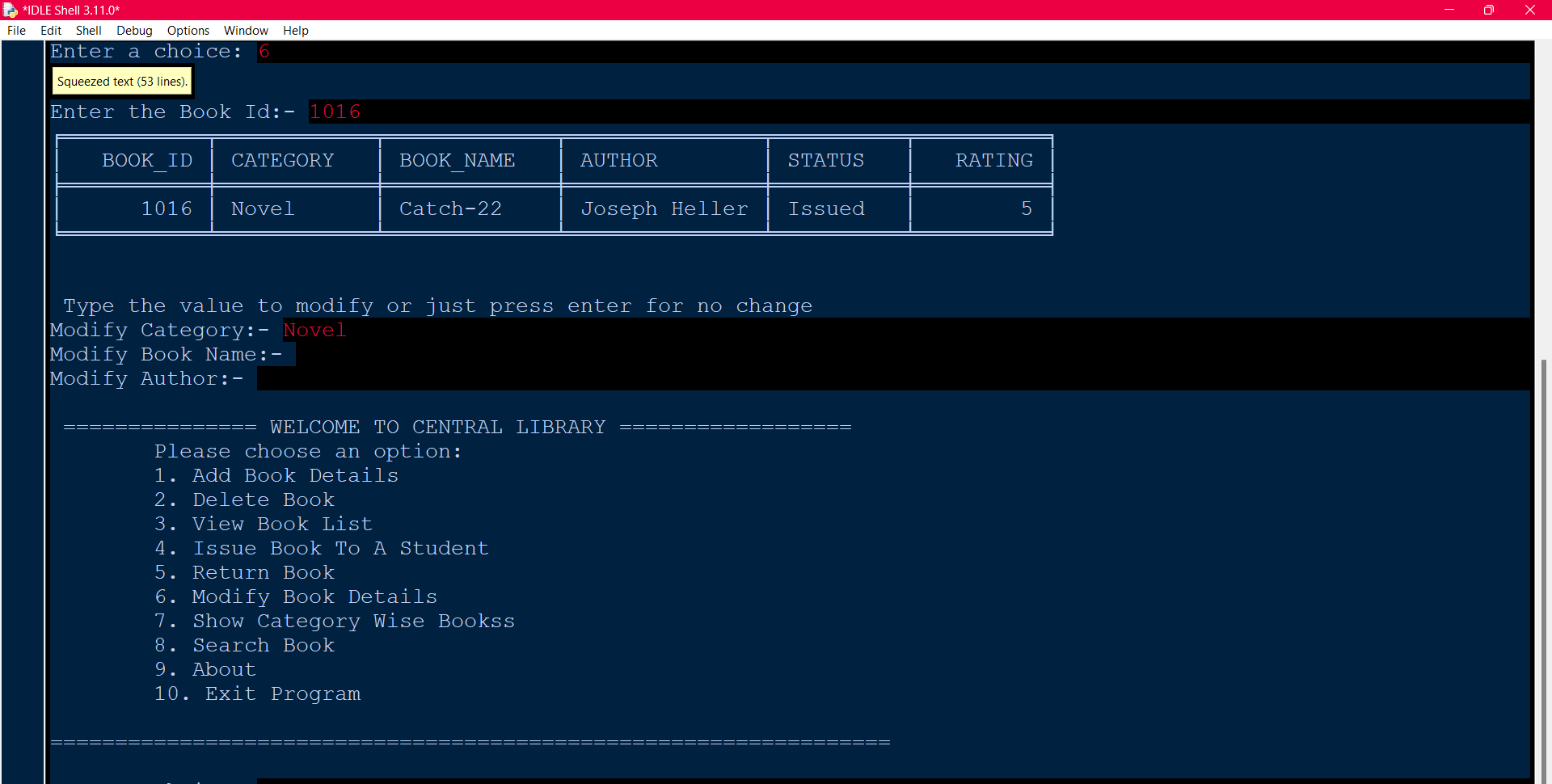
**4.**

****

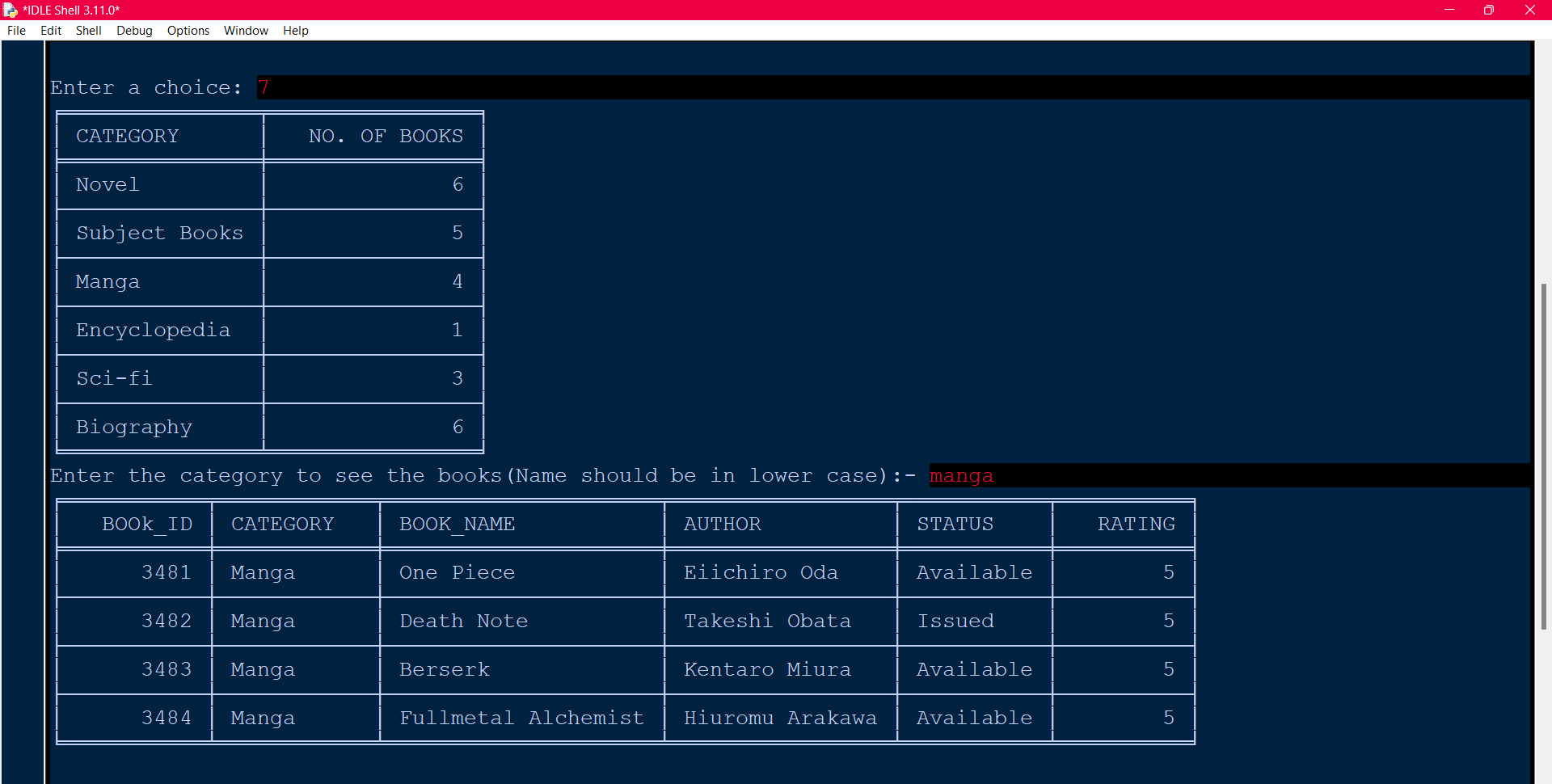
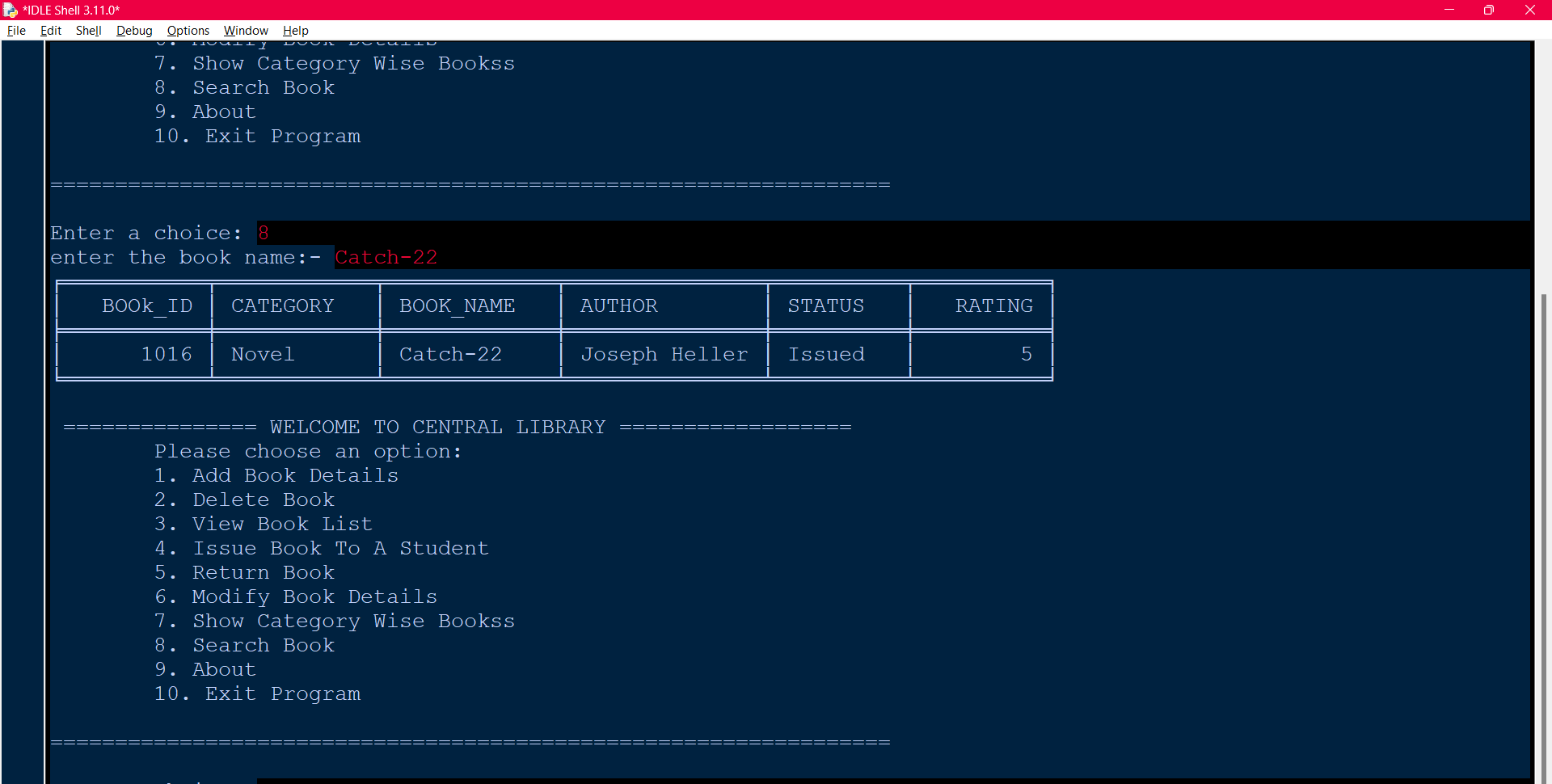
**5.**

****

**6.**

****

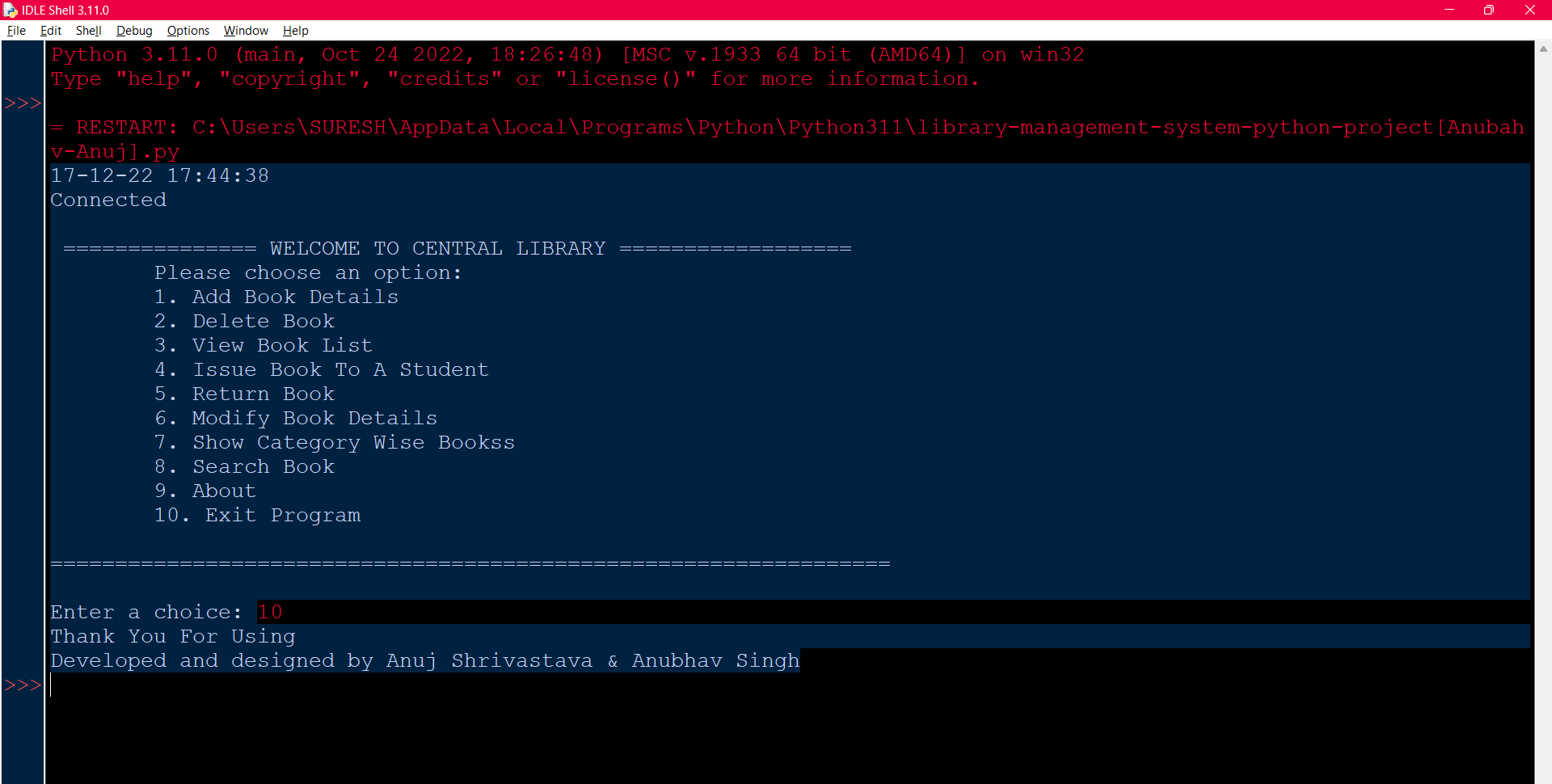
**7.**

**8.**

**9.**

****

**10.**

****