A PROJECT REPORT

ON

LIBRARY MANAGEMENT SYSTEM

FOR AISSCE 2021 EXAMINATION

[AS A PART OF THE COMPUTER SCIENCE COURSE (083)]

SUBMITTED BY: -

NAME: _____

ROLLNO.:_____

UNDER THE GUIDANCE OF: Ms.Monika Kohli

PGT (COMP.SC)

CERTIFICATE

This is to certify that the Project / Dissertation entitled **Library Management System** is a bonafide work done by _____ of class XII Session 20-21 in partial fulfillment of CBSE's AISSCE Examination 2021 and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form a part of any other course undergone by the candidate.

…… **Signature of Student**

Name:

Roll No.:

…….. **Signature of Teacher/Guide**

Name: Ms.MonikaKohli

Design.: PGT Comp.Sc.

Class XII

CONTENTS

1. Introduction

This software project is developed to automate the functionalities of a travel agency.The purpose of the software project is to develop the Management Information System (MIS) to automate the record keeping of books, Members ,issue types and return transactions with a view to enhance the decision making of the functionaries.

A MIS mainly consists of a computerized database, a collection of inter-related tables for a particular subject or purpose, capable to produce different reports relevant to the user. An application program is tied with the database for easy access and interface to the database. Using Application program or front-end, we can store, retrieve and manage all information in proper way.

This software, being simple in design and working, does not require much of training to users, and can be used as a powerful tool for automating a LIBRARY MANAGEMENT SYSTEM.

During coding and design of the software Project, Java NetBeans IDE, a powerful front-end tool is used for getting Graphical User Interface (GUI) based integrated platform and coding simplicity. As a back-end a

powerful, open source RDBMS, My SQL is used as per requirement of the CBSE curriculum of Informatics Practices Course.

2. Objective & Scope of the Project

The objective of the software project is to develop a computerized IS

to automate the functions of a LIBRARY MANAGEMENT SYSTEM.

This software project is also aimed to enhance the current record keeping system, which will help managers to retrieve the up-to-date information at right time in right shape.

The proposed software system is expected to do the following functionality-

- ✔ To provide a user friendly, Graphical User Interface (GUI) based integrated and centralized environment for MIS activities.
- ✔ The proposed system should maintain all the records and transactions, and should generate the required reports and information when required.
- ✔ To provide graphical and user-friendly interface to interact with a centralized database based on client-server architecture.
- ✔ To identify the critical operation procedure and possibilities of simplification using modern IT tools and practices.

In its current scope, the software enables user to retrieve and update the information from centralized database designed with MySQL . This software does not require much training time of the users due to limited functionality and simplicity.

During the development of **LIBRARY MANAGEMENT SYSTEM** project, Python IDE, a powerful, open source event-driven form-based development environment is used for modular design and future expandability of the system.

Despite of the best effort of the developer, the following limitations and functional boundaries are visible, which limits the scope of this application software.

1. This software can store records and produce reports in pre-designed format in soft copy. There is no facility yet to produce customized reports. Only specified reports are covered.
2. There is no provision to calculate fine or penalty etc. for defaulter members; however it can be developed easily with the help of adding modules.
3. Some application area like accounting of vehicles and fines etc. are not implemented in the project. It facilitates manager to record and update only transaction record.

So far as future scope of the project is concerned, firstly it is open to any modular expansion i.e. other modules or functions can be designed and embedded to handle the user need in future. Any part of the software and reports can be modified independently without much effort.

3. System Implementation

3.1 The Hardware used:

While developing the system, the used hardware are:

PC with Intel Core i5-2400S processor having 4.00 GB RAM, 64-bit Operating System , SVGA and other required devices.

3.2 The Softwares used:

- Microsoft Windows® 10 Pro as Operating System.
- Python 3.7.2 as Front-end Development environment.
- MySQL as Back-end Sever with Database for Testing.
- MS-Word 2010 for documentation.

4. Theoretical Background

4.1 What is Database?

Introduction and Concepts:

A database is a collection of information related to a particular subject or purpose, such as tracking customer orders or maintaining a music collection. Using any RDBMS application software like MS SQL Server, MySQL, Oracle, Sybase etc, you can manage all your information from a single database file. Within the file, divide your data into separate storage containers called tables. You may and retrieve the data using queries.

A table is a collection of data about a specific topic, such as products or suppliers. Using a separate table for each topic means you can store that data only once, which makes your database more efficient and reduces data-entry errors. Table organises data into columns (called fields) and rows (called records).

A Primary key is one or more fields whose value or values uniquely identify each record in a table. In a relationship, a primary key is used to refer to specific record in one table from another table. A primary key is called foreign key when it is referred to from another table.

To find and retrieve just the data that meets conditions you specify, inclucreate a queryding data from multiple tables, . A query can also update or delete multiple records at the same time, and perform built-in or custom calculations on your data.

R

A ıas a
l ıany
d ː can
h ıtion
in its files to prepare summaries, estimates and other reports. The

management of data in a database system is done by means of a general-purpose software package called a Database Management System (DBMS). Some commercially available DBMS are MS SQL Server, MS ACCESS, INGRES, ORACLE, and Sybase. A database management system, therefore, is a combination of hardware and software that can be used to set up and monitor a database, and can manage the updating and retrieval of database that has been stored in it. Most of the database management systems have the following capabilities:

- Creating of a table, addition, deletion, modification of records.
- Retrieving data collectively or selectively.
- The data stored can be sorted or indexed at the user's discretion and direction.
- Various reports can be produced from the system. These may be either standardized report or that may be specifically generated according to specific user definition.
- Mathematical functions can be performed and the data stored in the database can be manipulated with these functions to perform the desired calculations.
- To maintain data integrity and database use.

The DBMS interprets and processes users' requests to retrieve information from a database. In most cases, a query request will have to penetrate several layers of software in the DBMS and operating system before the physical database can be accessed. The DBMS responds to a query by invoking the appropriate subprograms, each of which performs its special function to interpret the query, or to locate the desired data in the database and present it in the desired order.

4.2 What is My SQL ?

The management of data in a database system is done by means of a general-purpose software package called a Database Management

System (DBMS). Some commercially available RDBMS are MS SQL Server, MS ACCESS, INGRES, ORACLE, and Sybase.

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. MySQL is named after co-founder Monty Widenius's daughter, My. The name of the MySQL Dolphin (our logo) is "Sakila,".

- **MySQL is a database management system.**

  A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL is based on SQL.**

  A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of "MySQL" stands for "Structured Query Language." SQL is the most common standardized language used to access databases and is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, "SQL-92" refers to the standard released in 1992, "SQL:1999" refers to the standard released in 1999, and "SQL:2003" refers to the current version of the standard.

- **MySQL software is Open Source.**

  Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License),

- **The MySQL Database Server is very fast, reliable, and easy to use.**

  If that is what you are looking for, you should give it a try. MySQL Server also has a practical set of features developed in close cooperation with our users. You can find a performance comparison of MySQL Server with other database managers on our benchmark page. MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

- **MySQL Server works in client/server or embedded systems.**

  The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

The Main Features of MySQL

- Written in C and C++.
- Works on many different platforms.
- Uses multi-layered server design with independent modules.
- Provides transactional and nontransactional storage engines.
- Designed to make it relatively easy to add other storage engines. This is useful if you want to provide an SQL interface for an in-house database.
- Uses a very fast thread-based memory allocation system.
- Executes very fast joins using an optimized nested-loop join.
- Implements SQL functions using a highly optimized class library that should be as fast as possible. Usually there is no memory allocation at all after query initialization.

- Provides the server as a separate program for use in a client/server networked environment, and as a library that can be embedded (linked) into standalone applications. Such applications can be used in isolation or in environments where no network is available.
- Password security by encryption of all password traffic when you connect to a server.
- Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
- MySQL client programs can be written in many languages. A client library written in C is available for clients written in C or C++, or for any language that provides C bindings.
- APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available, enabling MySQL clients to be written in many languages.
- The Connector/ODBC (MyODBC) interface provides MySQL support for client programs that use ODBC (Open Database Connectivity) connections.
- The Connector/J interface provides MySQL support for Java client programs that use JDBC connections. Clients can be run on Windows or Unix. Connector/J source is available.

4.3 **What is Python ?**

Python is an open source , object oriented high level programming language developed by Guido Van Rossum in 1991 at the National Research Institute for Mathematics,Netherlands.

Features of Python:

- It is an interactive ,interpreted language.
- It is a loosely typed object –oriented language.
- It is a free open –source and portable language,
- It takes less time to develop programs.
- It is extensible / extendable and highly efficient .

- It supports GUI.
- It can be easily compatible with other languages like C , C++ etc.
- It is used for both scientific and non-scientific programming.

Installing Python:

It can be installed by using website :

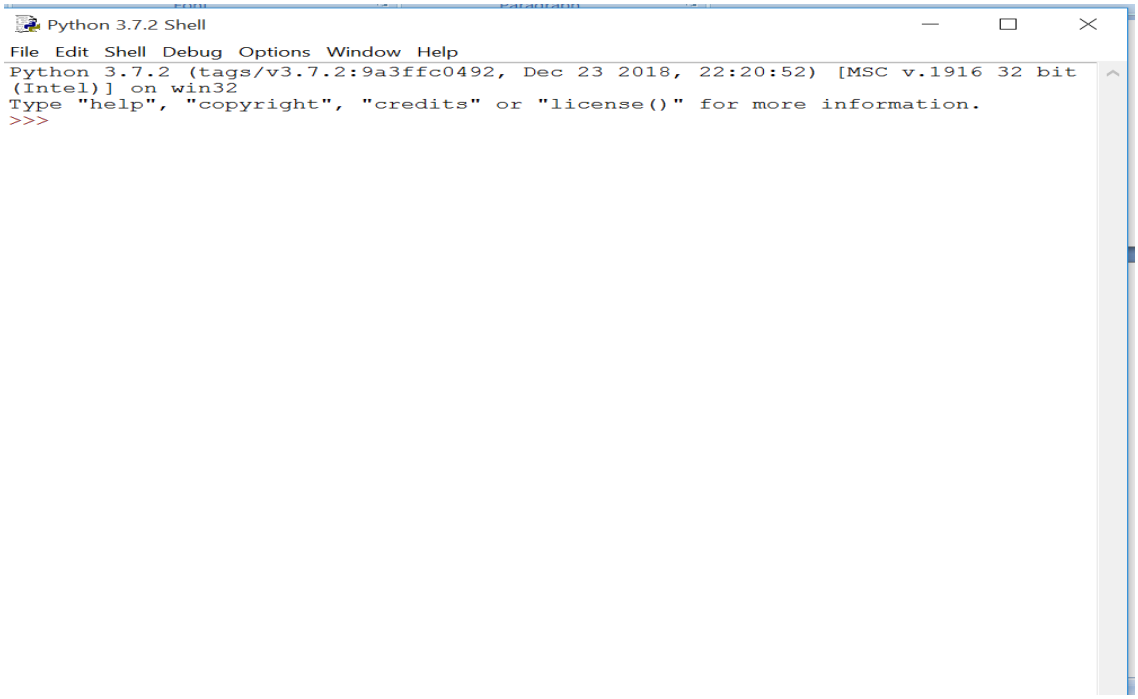
https://www.python.org/downloads/

Interacting with Python:

Python programs can be run in two ways:

- Using Command line window

- Using IDLE



5. System Design & Development

5.1 Database Design:

An important aspect of system design is the design of data storage structure. To begin with a logical model of data structure is developed first. A database is a container object which contains tables, queries, reports and data validation policies enforcement rules or contraints etc. A logical data often represented as a records are kept in different tables after reducing anomalies and redundancies. The goodness of data base design lies in the table structure and its relationship.

This software project maintains a database named lIBTARAY which contains the following tables.

Table Design:

The database of **LIBRARY MANAGEMENT SYSTEM** contains 3 tables in database Library. The tables are normalized to minimize the redundancies of data and enforcing the validation rules of the organization. Most of the tables are designed to store master records. The tables and their structure are given below.

```
mysql> create database library;
Query OK, 1 row affected (0.02 sec)
```

```
mysql> use library
Database changed
```

Table 1: BOOKRECORD

```
mysql> create table BOOKRECORD
    -> (bno varchar(5) PRIMARY KEY,
    -> BNAME VARCHAR(20) UNIQUE,
    -> AUTHOR VARCHAR(20) NOT NULL,
    -> PUBLISHER VARCHAR(50) NOT NULL,
    -> PRICE DECIMAL(6,2) NOT NULL,
    -> QTY INT NOT NULL,
    -> D_O_PUR DATE);
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> describe bookrecord;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| bno       | varchar(5)   | NO   | PRI | NULL    |       |
| BNAME     | varchar(20)  | YES  | UNI | NULL    |       |
| AUTHOR    | varchar(20)  | NO   |     | NULL    |       |
| PUBLISHER | varchar(50)  | NO   |     | NULL    |       |
| PRICE     | decimal(6,2) | NO   |     | NULL    |       |
| QTY       | int(11)      | NO   |     | NULL    |       |
| D_O_PUR   | date         | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
7 rows in set (0.16 sec)
```

```
mysql> alter table bookrecord
    -> modify bname varchar(50);
Query OK, 0 rows affected (0.19 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> insert into bookrecord values("kb001","computer science with python(11)","sumita arora","dhanpat rai",425,2,"2019/03/11");
Query OK, 1 row affected (0.02 sec)

mysql> insert into bookrecord values("kb002","computer science with python(12)","sumita arora","dhanpat rai",425,3,"2019/03/17");
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from bookrecord;
+-------+---------------------------------+-------------+-----------------+--------+-----+------------+
| bno   | bname                           | AUTHOR      | PUBLISHER       | PRICE  | QTY | D_O_PUR    |
+-------+---------------------------------+-------------+-----------------+--------+-----+------------+
| kb001 | computer science with python(11) | sumita arora | dhanpat rai     | 425.00 |   2 | 2019-03-11 |
| kb002 | computer science with python(12) | sumita arora | dhanpat rai     | 425.00 |   3 | 2019-03-17 |
| kb003 | computer sci with python(12)     | preeti arora | sultan chand rai | 325.00 |   3 | 2019-03-17 |
+-------+---------------------------------+-------------+-----------------+--------+-----+------------+
3 rows in set (0.00 sec)
```

```
mysql> show tables;
+-------------------+
| Tables_in_library |
+-------------------+
| bookrecord        |
| issue             |
| member            |
+-------------------+
3 rows in set (0.00 sec)
```

Table2: MEMBER

```
+---------+------------+------+-----+---------+-------+
| Field   | Type       | Null | Key | Default | Extra |
+---------+------------+------+-----+---------+-------+
| mno     | varchar(5) | NO   | PRI | NULL    |       |
| mname   | varchar(20)| NO   |     | NULL    |       |
| mob     | varchar(10)| YES  |     | NULL    |       |
| maddress| varchar(50)| YES  |     | NULL    |       |
| dop     | date       | YES  |     | NULL    |       |
+---------+------------+------+-----+---------+-------+
5 rows in set (0.05 sec)
```

```
mysql> insert into member values("km001","manav","9812367450","narela");
Query OK, 1 row affected (0.03 sec)

mysql> insert into member values("km002","paras","991367050","narela");
Query OK, 1 row affected (0.03 sec)

mysql> insert into member values("km003","faisal","881367500","alipur");
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from member;
+-------+--------+------------+----------+
| mno   | mname  | mob        | maddress |
+-------+--------+------------+----------+
| km001 | manav  | 9812367450 | narela   |
| km002 | paras  | 991367050  | narela   |
| km003 | faisal | 881367500  | alipur   |
+-------+--------+------------+----------+
3 rows in set (0.00 sec)
```

Table3: ISSUE

```
mysql> create table issue
    -> (bno varchar(5),
    -> mno varchar(5),
    -> d_o_issue date,
    -> d_o_ret date);
```

```
mysql> insert into issue values("KB001","KM001","2019/11/01","2019/11/07");
Query OK, 1 row affected (0.03 sec)

mysql> insert into issue values("KB002","KM002","2019/10/01","2019/10/07");
Query OK, 1 row affected (0.02 sec)

mysql> insert into issue values("KB003","KM003","2019/10/01","2019/10/07");
Query OK, 1 row affected (0.02 sec)

mysql> insert into issue values("KB004","KM004","2019/10/11","2019/10/17");
Query OK, 1 row affected (0.02 sec)
```

```
mysql> select * from issue;
+-------+-------+-----------+-----------+
| bno   | mno   | d_o_issue | d_o_ret   |
+-------+-------+-----------+-----------+
| KB001 | KM001 | 2019-11-01 | 2019-11-07 |
| KB002 | KM002 | 2019-10-01 | 2019-10-07 |
| KB003 | KM003 | 2019-10-01 | 2019-10-07 |
| KB004 | KM004 | 2019-10-11 | 2019-10-17 |
| kb001 | m0004 | 2019-12-26 | NULL       |
+-------+-------+-----------+-----------+
5 rows in set (0.00 sec)
```

TABLE JOIN

```
mysql> select * from issue , member where issue.mno=member.mno;
+-------+-------+------------+------------+-------+--------+------------+----------+
| bno   | mno   | d_o_issue  | d_o_ret    | mno   | mname  | mob        | maddress |
+-------+-------+------------+------------+-------+--------+------------+----------+
| KB001 | KM001 | 2019-11-01 | 2019-11-07 | km001 | manav  | 9812367450 | narela   |
| KB002 | KM002 | 2019-10-01 | 2019-10-07 | km002 | paras  | 991367050  | narela   |
| KB003 | KM003 | 2019-10-01 | 2019-10-07 | km003 | faisal | 881367500  | alipur   |
+-------+-------+------------+------------+-------+--------+------------+----------+
3 rows in set (0.00 sec)
```

```
mysql> select * from issue , bookrecord where issue.bno=bookrecord.bno;
+-------+-------+------------+------------+-------+-----------------------------+-----------+----------------+-------+-----+------------+
| bno   | mno   | d_o_issue  | d_o_ret    | bno   | bname                       | AUTHOR    | PUBLISHER      | PRICE | QTY | D_O_PUR    |
+-------+-------+------------+------------+-------+-----------------------------+-----------+----------------+-------+-----+------------+
| KB001 | KM001 | 2019-11-01 | 2019-11-07 | kb001 | computer science with python(11) | sumita arora | dhanpat rai    | 425.00 |   2 | 2019-03-11 |
| KB002 | KM002 | 2019-10-01 | 2019-10-07 | kb002 | computer science with python(12) | sumita arora | dhanpat rai    | 425.00 |   3 | 2019-03-17 |
| KB003 | KM003 | 2019-10-01 | 2019-10-07 | kb003 | computer sci with python(12)     | preeti arora | sultan chand rai | 325.00 |   3 | 2019-03-17 |
+-------+-------+------------+------------+-------+-----------------------------+-----------+----------------+-------+-----+------------+
3 rows in set (0.00 sec)
```


5.2 Event Coding:

#MODULE : LIBRARY MANAGEMENT

```
import MenuLib
import Book
import issue
while True:
        Book.clrscreen()
        print("\t\t\t Library Management\n")
        print("==========================================
==============")
        print("1. Book Management ")
        print("2. Members Management s ")
        print("3. Issue/Return Book ")
        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:
                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

def MenuBook():
   while True:
      Book.clrscreen()
      print("\t\t\t Book Record Management\n")

        print("==========================================
        ==============")
      print("1. Add Book Record ")
      print("2. Display Book Records ")
      print("3. Search Book Record ")
      print("4. Delete Book Record ")
```

```
        print("5. Update Book Record ")
        print("6. Return to Main Menu ")

          print("==========================================
          ===============")
        choice=int(input("Enter Choice between 1 to 5-------> : "))
        if choice==1:
           Book.insertData()
        elif choice==2:
           Book.display()
        elif choice==3:
           Book.SearchBookRec()
        elif choice==4:
           Book.deleteBook()
        elif choice==5:
           print("No such Function")
        elif choice==6:
           return
        else:
           print("Wrong Choice.....Enter Your Choice again")
           x=input("Enter any key to continue")

#-------------------------------------------------------------------------------

def MenuMember():
   while True:
      Book.clrscreen()
      print("\t\t\t Member Record Management\n")

        print("==========================================
        ==============")
      print("1. Add Member Record ")
      print("2. Display Member Records ")
      print("3. Search Member Record ")
      print("4. Delete Member Record ")
      print("5. Update Book Record ")
      print("6. Return to Main Menu ")

        print("==========================================
        ==============")
      choice=int(input("Enter Choice between 1 to 5-------> : "))
      if choice==1:
         Member.insertData()
```

```
        elif choice==2:
            Member.display()
        elif choice==3:
            Member.SearchMember()
        elif choice==4:
            Member.deleteMember()
        elif choice==5:
            print("No such Function")
        elif choice==6:
            return
        else:
            print("Wrong Choice.....Enter Your Choice again")
            x=input("Enter any key to continue")

#-------------------------------------------------------------------------------


def MenuIssueReturn():
    while True:
        Book.clrscreen()
        print("\t\t\t Issue/Return 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 5-------> : "))
        if choice==1:
            issue.issueBookData()
        elif choice==2:
            issue.ShowIssuedBooks()
        elif choice==3:
            issue.returnBook()
        elif choice==4:
            return
        else:
            print("Wrong Choice.....Enter Your Choice again")
            x=input("Enter any key to continue")
```

 # PYTHON MODULE : BOOK

```
import mysql.connector
from mysql.connector import errorcode
from datetime import date, datetime, timedelta
from mysql.connector import (connection)
import os
import platform
def clrscreen():
        if platform.system()=="Windows":
                print(os.system("cls"))
def display():
        try:
                os.system('cls')
                cnx = connection.MySQLConnection(user='root', password='h',
                host='localhost',
                database='LIbrary')
                Cursor = cnx.cursor()
                query = ("SELECT * FROM BookRecord")
Cursor.execute(query)
for (Bno,Bname,Author,price,publ,qty,d_o_purchase) in Cursor:
        print("==========================================
==============")
        print("Book Code : ",Bno)
print("Book Name : ",Bname)
print("Author of Book : ",Author)
print("Price of Book : ",price)
print("Publisher : ",publ)
print("Total Quantity in Hand : ",qty)
print("Purchased On : ",d_o_purchase)
print("==================================================
===========")
Cursor.close()
cnx.close()
print("You have done it!!!!!!")
except mysql.connector.Error as err:
if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
        print("Something is wrong with your user name or password")
elif err.errno == errorcode.ER_BAD_DB_ERROR:
        print("Database does not exist")
```

```
else:
        print(err)
else:
        cnx.close()

def insertData():
        try:
                cnx = connection.MySQLConnection(user='root', password='h',
host='127.0.0.1', database='Library')
Cursor = cnx.cursor()
bno=input("Enter Book Code : ")
bname=input("Enter Book Name : ")
Auth=input("Enter Book Author's Name : ")
price=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 seperately: ")
DD=int(input("Enter Date : "))
MM=int(input("Enter Month : "))
YY=int(input("Enter Year : "))
Qry = ("INSERT INTO BookRecord "\
"VALUES (%s, %s, %s, %s, %s, %s, %s)")
data = (bno,bname,Auth,price,publ,qty,date(YY,MM,DD))
Cursor.execute(Qry,data)
# Make sure data is committed to the database

cnx.commit()
Cursor.close()
cnx.close()
print("Record Inserted.....")
except mysql.connector.Error as err:
if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
        print("Something is wrong with your user name or password")
elif err.errno == errorcode.ER_BAD_DB_ERROR:
        print("Database does not exist")
else:
        print(err)
cnx.close()

def deleteBook():
        try:
                cnx = connection.MySQLConnection(user='root', password='h',
host='127.0.0.1', database='Library')
```

```
Cursor = cnx.cursor()
bno=input("Enter Book Code of Book to be deleted from the Library : ")
Qry =("""DELETE FROM BookRecord WHERE BNO = %s""")
del_rec=(bno,)
Cursor.execute(Qry,del_rec)

# Make sure data is committed to the database

cnx.commit()
Cursor.close()
cnx.close()
print(Cursor.rowcount,"Record(s) Deleted Successfully.....")
except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
                print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
                print("Database does not exist")
        else:
                print(err)
cnx.close()

def SearchBookRec():
        try:
                cnx = connection.MySQLConnection(user='root', password='h',
host='127.0.0.1',  database='Library)
Cursor = cnx.cursor()
bno=input("Enter Book No to be Searched from the Library : ")
query = ("SELECT * FROM BookRecord where BNo = %s ")
rec_srch=(bno,)
Cursor.execute(query,rec_srch)
Rec_count=0
for (Bno,Bname,Author,price,publ,qty,d_o_purchase) in Cursor:
        Rec_count+=1
        print("==========================================
==============")
print("Book Code : ",Bno)
print("Book Name : ",Bname)
print("Author of Book : ",Author)
print("Price of Book : ",price)
print("Publisher : ",publ)
print("Total Quantity in Hand : ",qty)
print("Purchased On : ",d_o_purchase)
```

```
print("================================================
==========")
if Rec_count%2==0:
        input("Press any key to continue")
clrscreen()
print(Rec_count, "Record(s) found")

# Make sure data is committed to the database

cnx.commit()
Cursor.close()
cnx.close()
except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
                print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
                print("Database does not exist")
        else:
                print(err)
cnx.close()

def UpdateBook():
        try:
                cnx = connection.MySQLConnection(user='root',  password='h',
host='127.0.0.1',  database='Library')
Cursor = cnx.cursor()
bno=input("Enter Book Code of Book to be Updated from the Library : ")
query = ("SELECT * FROM BookRecord where BNo = %s ")
rec_srch=(bno,)
print("Enter new data ")
bname=input("Enter Book Name : ")
Auth=input("Enter Book Author's Name : ")
price=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 seperately: ")
DD=int(input("Enter Date : "))
MM=int(input("Enter Month : "))
YY=int(input("Enter Year : "))
Qry = ("UPDATE BookRecord SET bname=%s,Author=%s,"\
"price=%s,publisher=%s,qty=%s,d_o_purchase=%s "\ "WHERE BNO=%s")
data = (bname,Auth,price,publ,qty,date(YY,MM,DD),bno)
Cursor.execute(Qry,data)
```

```
# Make sure data is committed to the database'''

cnx.commit()
Cursor.close()
cnx.close()
print(Cursor.rowcount,"Record(s) Updated Successfully.....")
except mysql.connector.Error as err:
if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
print("Something is wrong with your user name or password")
elif err.errno == errorcode.ER_BAD_DB_ERROR:
print("Database does not exist")
else:
print(err)
cnx.close()
UpdateBook()
```

#PYTHON MODULE : ISSUE

```
import mysql.connector
from mysql.connector import errorcode
from datetime import date
from mysql.connector import (connection)
import os
def clrscreen():
print('\n' *5)

def ShowIssuedBooks():
        try:
        os.system('cls')
        cnx = connection.MySQLConnection(user='root', password='h',
host='localhost', database='Library')
Cursor = cnx.cursor()
query = ("SELECT B.bno,bname,M.mno,mname,d_o_issue,d_o_ret FROM
bookRecord B,issue I"\
",member M where B.bno=I.bno and I.mno=M.mno")
Cursor.execute(query)
for (Bno,Bname,Mno,Mname,doi,dor) in Cursor:
        print("==========================================
==============")
        print("Book Code : ",Bno)
        print("Book Name : ",Bname)
        print("Member Code : ",Mno)
```

```
        print("Member Name : ",Mname)
        print("Date of issue : ",doi)
        print("Date of return : ",dor)
        print("==========================================
==============")
Cursor.close()
cnx.close()
print("You have done it!!!!!!")
except mysql.connector.Error as err:
if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
        print("Something is wrong with your user name or password")
elif err.errno == errorcode.ER_BAD_DB_ERROR:
        print("Database does not exist")
else:
        print(err)
else:
        cnx.close()

def issueBook():
        try:
                cnx = connection.MySQLConnection(user='root', password='h',
host='127.0.0.1', database='Library')
Cursor = cnx.cursor()
bno=input("Enter Book Code to issue : ")
mno=input("Enter Member Code : ")
print("Enter Date of Issue (Date/MOnth and Year seperately: ")
DD=int(input("Enter Date : "))
MM=int(input("Enter Month : "))
YY=int(input("Enter Year : "))
Qry = ("INSERT INTO issue (bno,mno,d_o_issue)"\
"VALUES (%s, %s, %s)")
data = (bno,mno,date(YY,MM,DD))
Cursor.execute(Qry,data)
cnx.commit()
Cursor.close()
cnx.close()
print("Record Inserted.....")
except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
                print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
                print("Database does not exist")
        else:
```

```
                print(err)
cnx.close()
def returnBook():
        try:
                cnx = connection.MySQLConnection(user='root', password='h',
host='127.0.0.1', database='Library')
Cursor = cnx.cursor()
bno=input("Enter Book Code of Book to be returned to the Library : ")
Mno=input("Enter Member Code of Member who is returning Book : ")
retDate=date.today()
Qry =("""Update Issue set d_o_ret= %s WHERE BNO = %s and Mno= %s """)
rec=(retDate,bno,Mno)
Cursor.execute(Qry,rec)
# Make sure data is committed to the database
cnx.commit()
Cursor.close()
cnx.close()
print(Cursor.rowcount,"Record(s) Deleted Successfully.....")
except mysql.connector.Error as err:
if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
        print("Something is wrong with your user name or password")
elif err.errno == errorcode.ER_BAD_DB_ERROR:
        print("Database does not exist")
else:
        print(err)
cnx.close()
```

#PYTHON MODULE MEMBER

```
import mysql.connector
from mysql.connector import errorcode
from datetime import date, datetime, timedelta
from mysql.connector import (connection)
import os

def clrscreen():
        print('\n' *5)

def display():
        try:
        os.system('cls')
        cnx = connection.MySQLConnection(user='root', password='h',
```

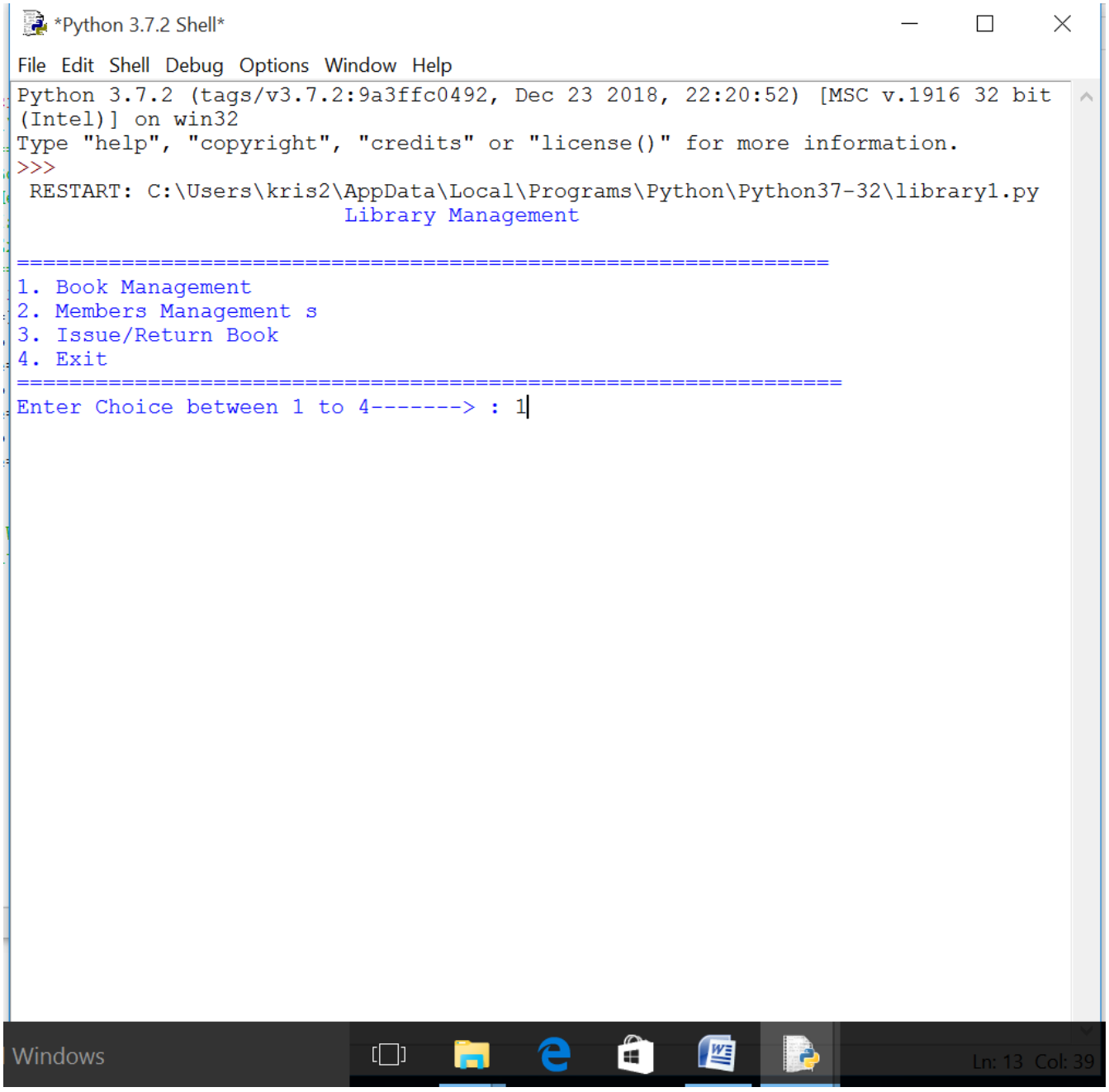
```
host='localhost', database='LIbrary')
Cursor = cnx.cursor()
query = ("SELECT * FROM Member")
Cursor.execute(query)
for (Mno,Mname,MOB,DOP,ADR) in Cursor:
print("==========================================================
==========")
print("Member Code : ",Mno)
print("Member Name : ",Mname)
print("Mobile No.of Member : ",MOB)
print("Date of Membership : ",DOP)
print("Address : ",ADR)
print("==========================================================
==========")
Cursor.close()
cnx.close()
print("You have done it!!!!!!")
except mysql.connector.Error as err:
if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
print("Something is wrong with your user name or password")
elif err.errno == errorcode.ER_BAD_DB_ERROR:
print("Database does not exist")
else:
print(err)
else:
cnx.close()

def insertMember():
        try:
cnx = connection.MySQLConnection(user='root', password='h',
host='127.0.0.1', database='Library')
Cursor = cnx.cursor()
mno=input("Enter Member Code : ")
mname=input("Enter Member Name : ")
mob=input("Enter Member Mobile No. : ")
print("Enter Date of Membership (Date/MOnth and Year seperately: ")
DD=int(input("Enter Date : "))
MM=int(input("Enter Month : "))
YY=int(input("Enter Year : "))
addr=input("Enter Member Adress : ")
Qry = ("INSERT INTO Member "\
"VALUES (%s, %s, %s, %s, %s)")
data = (mno,mname,mob,date(YY,MM,DD),addr)
```

```
Cursor.execute(Qry,data)

# Make sure data is committed to the database

cnx.commit()
Cursor.close()
cnx.close()
print("Record Inserted.....")
except mysql.connector.Error as err:
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
                print("Something is wrong with your user name or password")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
                print("Database does not exist")
        else:
print(err)
cnx.close()

def deleteMember():
        try:
                cnx = connection.MySQLConnection(user='root', password='h',
host='127.0.0.1', database='Library')
Cursor = cnx.cursor()
mno=input("Enter Member Code to be deleted from the Library : ")
Qry =("""DELETE FROM Member WHERE MNO = %s""")
del_rec=(mno,)
Cursor.execute(Qry,del_rec)
# Make sure data is committed to the database
cnx.commit()
Cursor.close()
cnx.close()
print(Cursor.rowcount,"Record(s) Deleted Successfully.....")
except mysql.connector.Error as err:
if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
print("Something is wrong with your user name or password")
elif err.errno == errorcode.ER_BAD_DB_ERROR:
print("Database does not exist")
else:
print(err)
cnx.close()


def SearchMember():
        try:
```

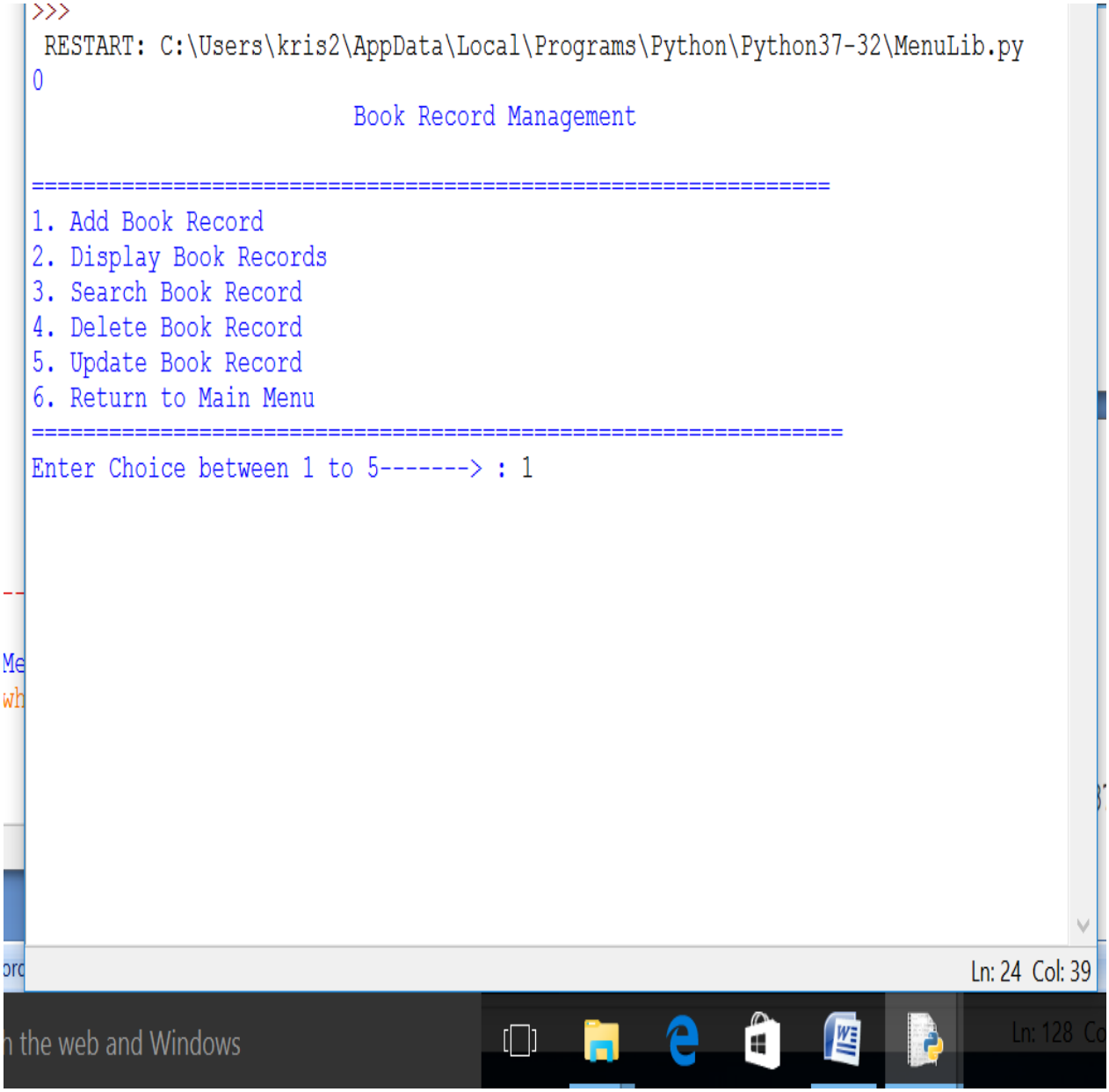
```
            cnx = connection.MySQLConnection(user='root', password='h',
host='127.0.0.1', database='Library')
Cursor = cnx.cursor()
mnm=input("Enter Book Name to be Searched from the Library : ")
query = ("SELECT * FROM Member where MName = %s ")
rec_srch=(mnm,)
Cursor.execute(query,rec_srch)
Rec_count=0
for (Mno,Mname,MOB,DOP,ADR) in Cursor:
print("==========================================================
==========")
print("Member Code : ",Mno)
print("Member Name : ",Mname)
print("Mobile No.of Member : ",MOB)
print("Date of Membership : ",DOP)
print("Address : ",ADR)
print("==========================================================
==========")
if Rec_count%2==0:
input("Press any key to continue")
clrscreen()
print(Rec_count, "Record(s) found")

# Make sure data is committed to the database

cnx.commit()
Cursor.close()
cnx.close()
except mysql.connector.Error as err:
if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
print("Something is wrong with your user name or password")
elif err.errno == errorcode.ER_BAD_DB_ERROR:
print("Database does not exist")
else:
print(err)
cnx.close()
```

6. Output

LIBRARY MANAGEMENT



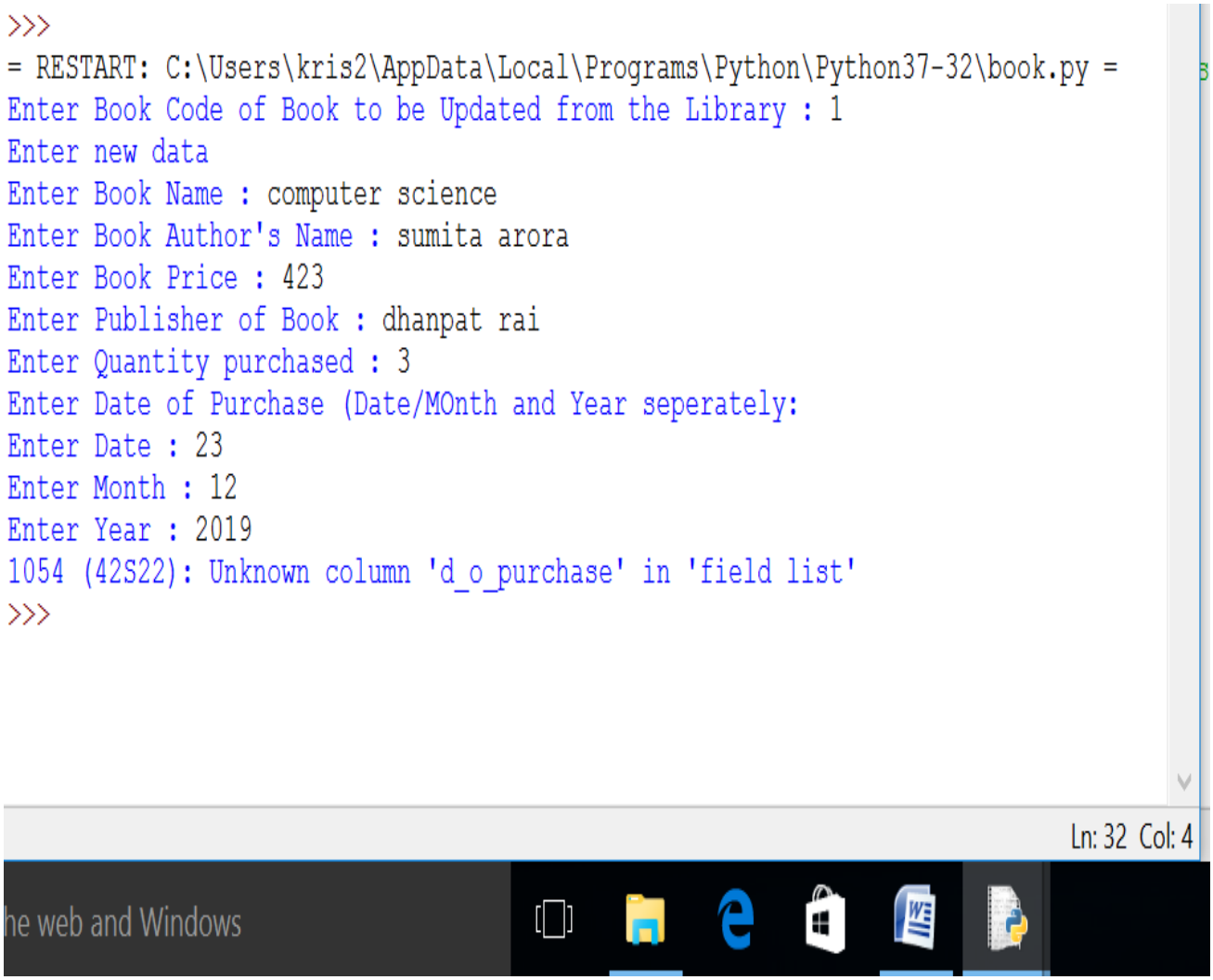
BOOK RECORD MANAGEMENT



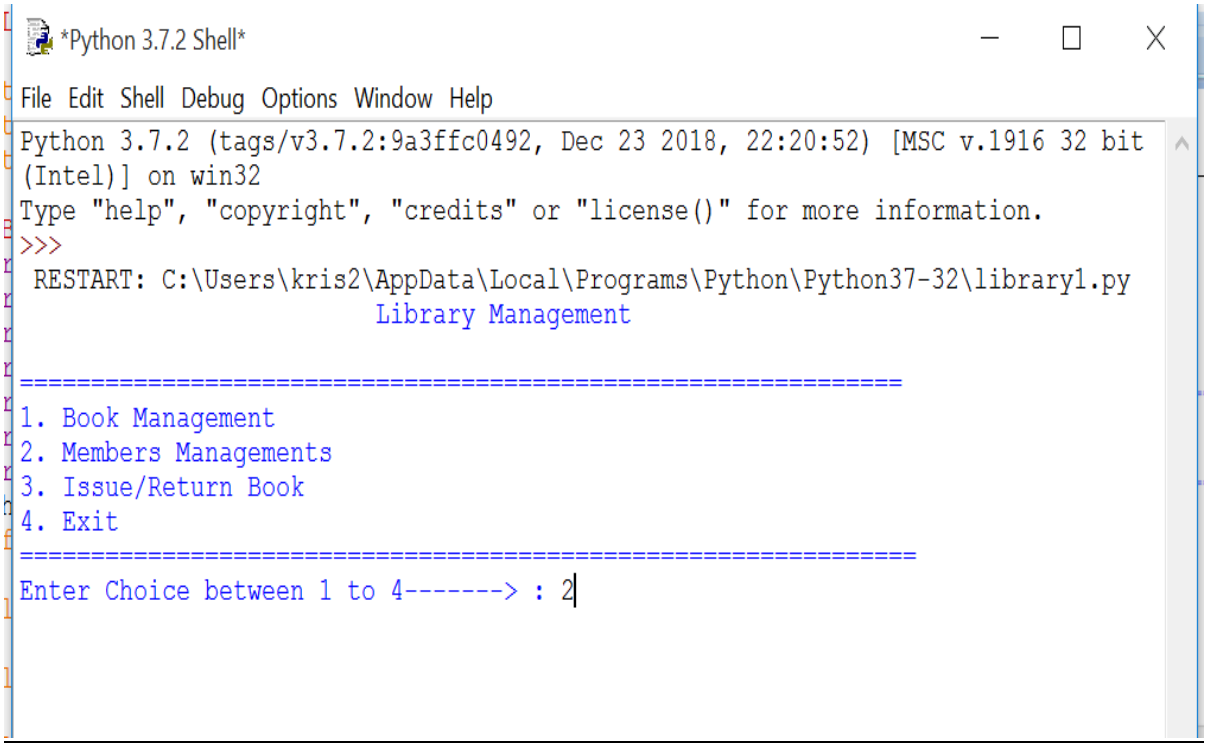
ADD A NEW BOOK

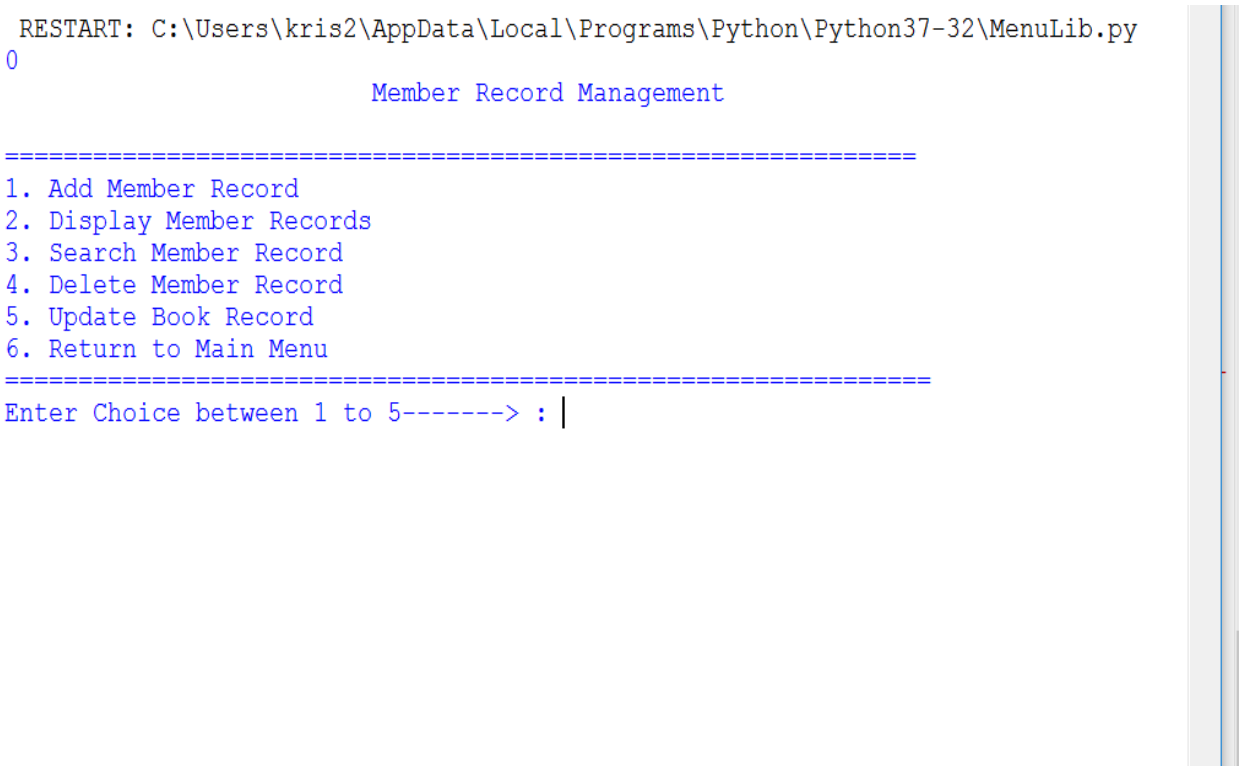
```
 RESTART: C:\Users\kris2\AppData\Local\Programs\Python\Python37-32\MenuLib.py
0
                        Book Record Management

================================================================
1. Add Book Record
2. Display Book Records
3. Search Book Record
4. Delete Book Record
5. Update Book Record
6. Return to Main Menu
================================================================
Enter Choice between 1 to 5-------> : 1
Enter Book Code : kb005
Enter Book Name : ABC OF PHYSICS
Enter Book Author's Name : LAKHMIR SINGH
Enter Book Price : 654
Enter Publisher of Book : RACHNA SAGAR
Enter Quantity purchased : 3
Enter Date of Purchase (Date/MOnth and Year seperately:
Enter Date : 12
Enter Month : 10
Enter Year : 2019
```

UPDATE AN EXISTING BOOK



MEMBER MANAGEMENT

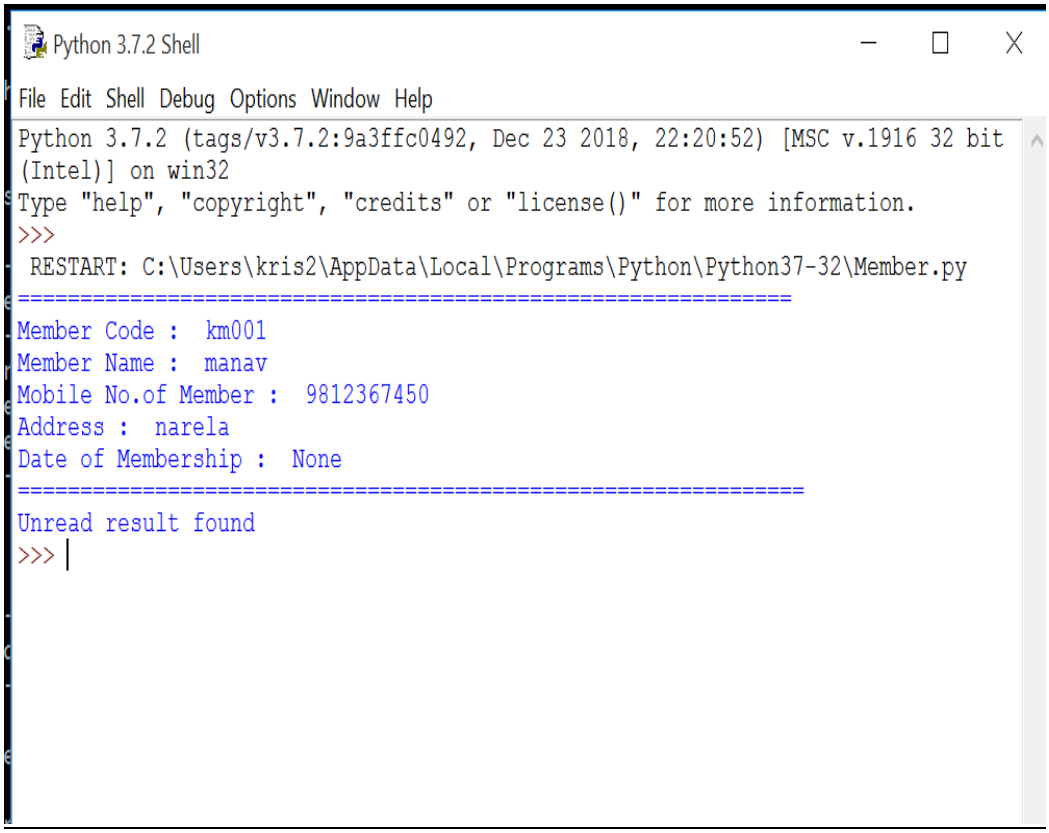




ADD A NEW MEMBER

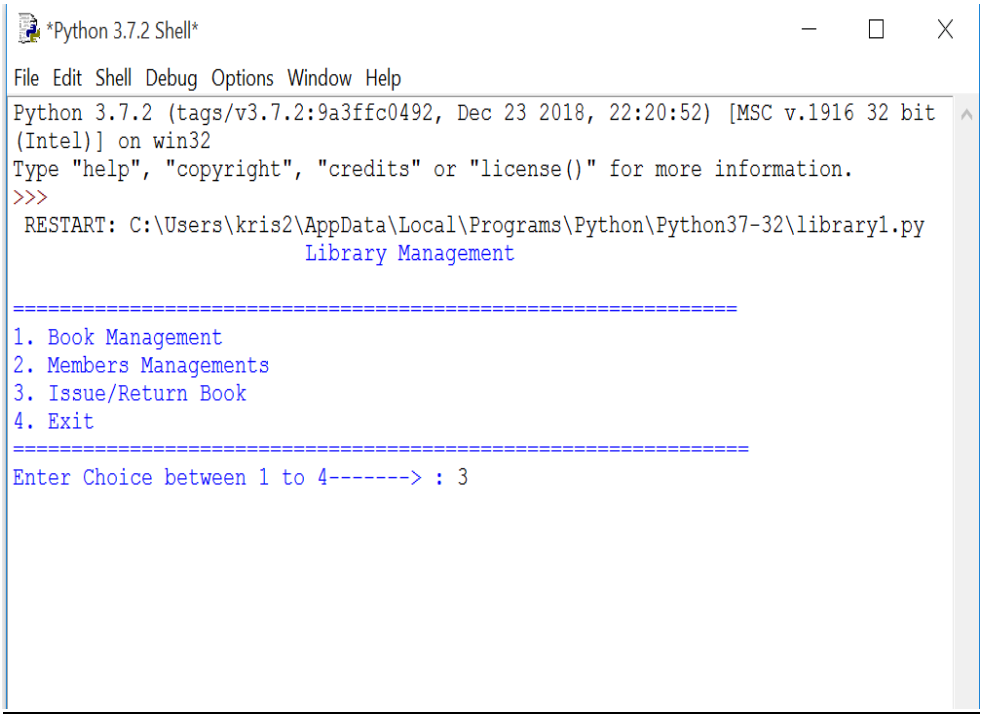
```
'''
 RESTART: C:\Users\kris2\AppData\Local\Programs\Python\Python37-32\MenuLib.py
0
                        Member Record Management

============================================================
1. Add Member Record
2. Display Member Records
3. Search Member Record
4. Delete Member Record
5. Update Book Record
6. Return to Main Menu
============================================================
Enter Choice between 1 to 5-------> : 1
Enter Member Code : m004
Enter Member Name : CHIRAG
Enter Member Mobile No. : 8976543210
Enter Date of Membership (Date/MOnth and Year seperately:
Enter Date : 15
Enter Month : 10
Enter Year : 2019
Enter Member Adress : MODEL TOWN
1136 (21S01): Column count doesn't match value count at row 1
0
```

DISPLAY MEMBER



ISSUE RECORD

LIBRARY MANAGEMENT SYSTEM



ISSUE BOOK TO A MEMBER

DISPLAY ISSUE RECORD

```
0
                    Issue/Return Record Management


=========================================================
1. Issue Book
2. Display Issued Book Records
3. Return Issued Book
4. Return to Main Menu
=========================================================
Enter Choice between 1 to 5-------> : 2
=========================================================
Book Code :  kb001
Book Name :  computer science with python(11)
Member Code :  km001
Member Name :  manav
Date of issue :  2019-11-01
Date of return :  2019-11-07
=========================================================
Unread result found
0
```

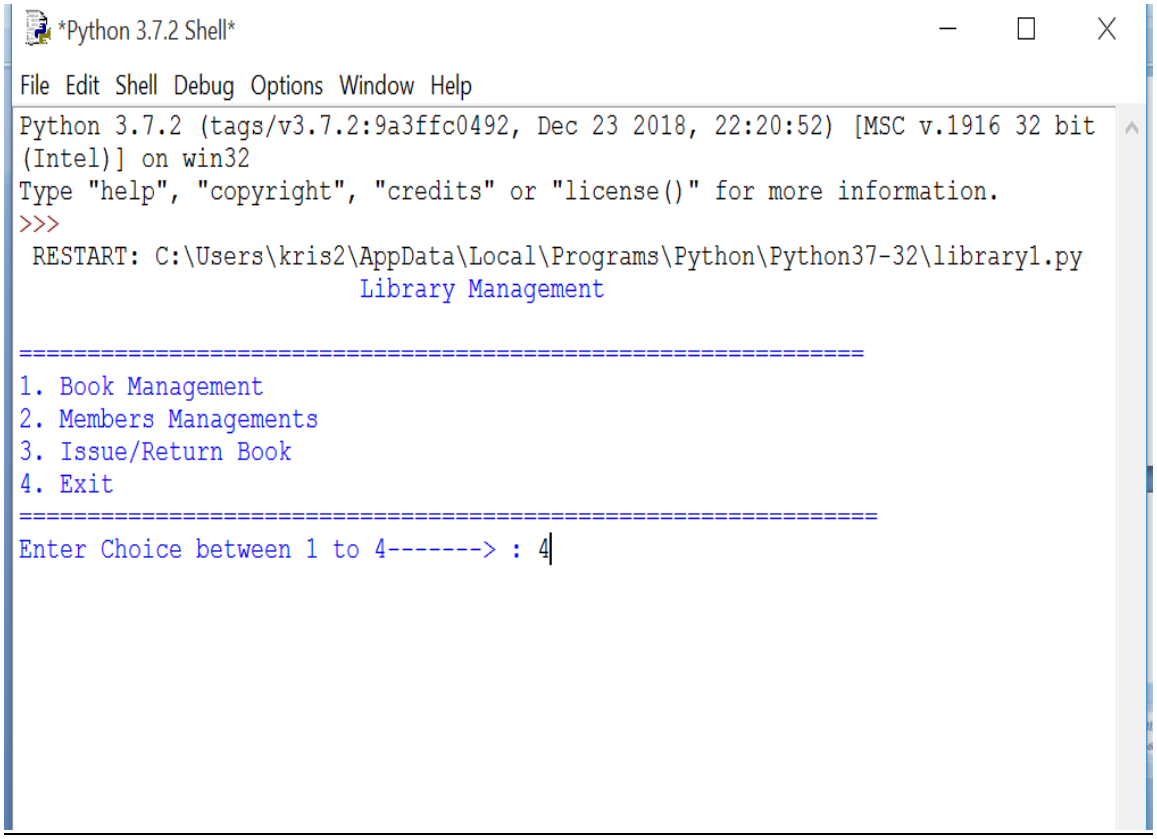
RETURN BOOK

```
                    Issue/Return Record Management


=====================================================

1. Issue Book
2. Display Issued Book Records
3. Return Issued Book
4. Return to Main Menu

=====================================================

Enter Choice between 1 to 5------> : 3
Enter Book Code of Book to be returned to the Library : KB001
Enter Member Code of Member who is returning Book : KM001
-1 Record(s) Deleted Successfully.............
0
```

RETURN TO MAIN MENU

```
0
                        Issue/Return Record Management

================================================================
1. Issue Book
2. Display Issued Book Records
3. Return Issued Book
4. Return to Main Menu
================================================================
Enter Choice between 1 to 5-------> : 4
```

EXIT FROM LIBRARY MANAGEMENT



7. User Manual

7.1 How to Install Software:

Hardware Requirement-

- Intel Pentium/Celeron or similar processor based PC at Client/Server end.
- 128 MB RAM and 4GB HDD space (for Database) is desirable.
- Standard I/O devices like Keyboard and Mouse etc.
- Printer is needed for hard-copy reports.
- Local Area Network(LAN) is required for Client-Server Installation

Software Requirement-

- Windows 2000/XP OS is desirable.
- NetBeans Ver 5.1 or higher should be installed with JDK and JVM.
- MySQL Ver 6.1 with Library Database must be present at machine.

Database Installation-

The software project is distributed with a backup copy of a Database named LIBRARY with required tables. Some dummy records are present in the tables for testing purposes, which can be deleted before inserting real data. The project is shipped with manav**.SQL** file which installs a database and tables in the computer system.

Note: The PC must have MySQL server with user (***root***) and password (h) . If root password is any other password, it can be changed by running MySQL Server Instance Configure Wizard.

Start Program MySQL MySQL Server MySQL Server Instance Config Wizard

Provide current password of root and new password as “h” , this will change the root password.

To install a MySQL database from a dump file ***(manav.sql)*** , simply follow the following steps.

Step 1: Copy the manav.sql file in **C:\Program files\Mysql\MySql server 5.1\Bin** folder.

Step 2: Open MySQL and type the following command to create the database named travelagency.

mysql> create database **library**;

Step 3: Open Command Window (Start Run cmd)

Step 4: Go to the following folder using CD command of DOS.

C:\Program files\Mysql\MySql server 5.1\Bin>

Step 5: type the following command on above prompt -

C:….\bin> mysql -u ***root*** -pmanav library <manav.sql

This will create a library database with required tables.

8.References