

Home  
Page



Signup



Login



Signout



Add  
+

Update



# Library Management System

Lend



User



Issue



Return



Search



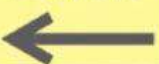
Rating



Admin



Back



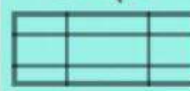
Continue



Python



SQL



Connect



By:- Anushka Kundu  
Podar International School



# Table Of Contents

**1. Certificate**

**2. Acknowledgement**

**3. Introduction**

**4. What is a library management system?**  
Why do we need it?

**5. Source Code**

**6. Outputs**

**7. Bibliography**

**8. Conclusion**



# **CERTIFICATE**

This is to certify that Anushka Kundu,  
a student of class XII, has successfully completed the  
research project on the topic “ Library Management  
System”, under the guidance of  
Sandhya PK .

References taken in making this project have been  
declared at the end of the report.

**Principal**

**Teacher In-charge**

**External Examiner**



# **ACKNOWLEDGEMENT**

I hereby acknowledge my deep sense of gratitude and indebtedness to the following personalities whose immense help, genius guidance, encouragement, necessary suggestions, initiations, enthusiasm and inspiration made this work a master art and a joint enterprise.

\_\_\_\_\_ :- (Principal)

\_\_\_\_\_ :- (PGT \_\_\_\_\_ )





# Introduction

## **Python**

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python was created in the late 1980s, and first released in 1991, by Guido van Rossum as a successor to the ABC programming language. Python's name is derived from the British comedy group Monty Python, whom Python creator Guido van Rossum enjoyed while developing the language.

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the TIOBE Programming Community Index where, as of February 2020, it is the third most popular language (behind Java, and C). It was selected Programming Language of the Year in 2007, 2010, and 2018.

## **MySQL**

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database

in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

### **Python and MySQL connection**

There are times while using Python, when you need to manipulate data stored in databases through applications designed by you. The MySQL connector allows you to do exactly that.

Before you start working with Python mysql connector, you need to install it on your computer, with the help of pip and Conda installation of Python. After you have installed Python MySQL connector, you can write Python scripts using MySQL.connector library, that can connect to MySQL databases from within Python

# What is a library management system?

## Why do we need it?

The library management system is a software that replaces manual work in a library.

The software helps to add and update book status in real time, unlike library records that are maintained manually. Users can easily lend and return books, LMS facilitates easy searching of books by users and access information about them. Users can also search for other users.

Signup, login and logout features for users, prevent access of account by third parties.

Admin privileges allow users to add, update and delete books. They can also view other user's lending history and check for unreturned books.

The electronic management via the software is essential to track information like issue date, due date, who has borrowed any material, etc. The system is developed and designed with an aim to facilitate efficient management to the schools to manage a modern library with accurate data management.

The modern day libraries has dynamic functionality, and in the modern world there is no place for error.

A Library Management System has innumerable benefits:

- (i) Reduces human efforts and time, preserves energy.
- (ii) Maintains data integrity.
- (iii) Keeps systematic records of books and users in real time.
- (iv) Multiple users can access information at same time, anytime, anywhere.

(v) Reduce library's operating cost.

(vi) modifying the source code can always be incorporated, to keep on increasing the functionality in the same structure without data losses.

Our Library Management System, provides all the above mentioned functionalities. Let's see.

## Source Code

```
import mysql.connector as sqltor

mydb = sqltor.connect(host="localhost", user="root", password="root")
cursor = mydb.cursor()
if mydb.is_connected():
    print("You are successfully connected to the database!\n")
    cursor.execute("create database if not exists lib_mang")
    cursor.execute("use lib_mang")
    cursor.execute("create table if not exists users(roll_no varchar(20) not
    null, primary key (roll_no), username "
        "varchar(20) not null ,password varchar(20) not null)")

def after_query():
    print("Here are all the books in the library:")

    cursor.execute("select * from books")
    books_output = cursor.fetchall()
    for row in books_output:
        print(row)

def after_login(user):
    cursor.execute(
        "create table if not exists books(book_name varchar(30) not null,
        primary key(book_name), no_of_books int, authors_name
        varchar(30))")
    permit = input("login with admin permission to access special
    admin features.
```

or continue as user

Do you have admin permission? (y/n): """)

while user is not None:

print("""

Press 0 to Log out

Press 1 to lend/issue book

Press 2 to return book

Press 3 to display your lending history

Press 4 to search books

Press 5 to search user

Press 6 to rate book

Press 7 to display ratings of book """)

if permit.lower() == "y":

print("""

Special Admin Features:

Press 8 to add book

Press 9 to update book

press 10 to delete book

press 11 to see lending history of some or all people

Press 12 to display users who have not returned books """)

ch2 = int(input("0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10  
/ 11 / 12 : "))

print("\n")

if ch2 > 12:

print("Error, enter value less than 12.")

print("\n")

continue

elif permit.lower() == "n":

ch2 = int(input("0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 : "))

if ch2 > 7:

print("Error, enter value less than 7.")

```

        print("\n")
        continue

    else:
        print("Invalid input, enter y or n only.")
        break

# LEND/ISSUE A BOOK

if ch2 == 1:
    print("""If you want to go back press 1
If you want to continue press 2""")
    b = int(input("Enter choice: "))
    if b == 1:
        continue

    elif b == 2:
        cursor.execute(
            "create table if not exists transact_lib(roll_no varchar(20)
not null, foreign key(roll_no) references users(roll_no), book_name
varchar(30) not null, foreign key(book_name) references
books(book_name), authors_name varchar(30), lend_date date,
exp_return_date date,act_return_date date)")
        mydb.commit()
        print("\n")
        print("Proceeding to lend/issue book")

        roll_no = input("Enter roll no: ")
        book_name = str(input("Enter name of book: "))
        authors_name = str(input("Enter name of author of book: "))
        lend_date = str(input("Enter date book was issued/lent
(yyyy-mm-dd): "))

```

```

        print("""Enter expected return date as after 21 days from
lend date
If book not returned then enter actual return date same as lend date
""")
        exp_return_date = str(input("Enter (expected) date of return
(yyyy-mm-dd): "))
        act_return_date = str(input("Enter (actual) date of return
(yyyy-mm-dd): "))

        cursor.execute("select * from books where book_name=" +
book_name + "")
        for (x, y, z) in cursor:
            if y != 0:
                cursor.execute(
                    "Insert into transact_lib values(" + roll_no + "," +
book_name + "," + authors_name + "," + lend_date + "," +
exp_return_date + "," + act_return_date + ")")
                cursor.execute("update books set
no_of_books=no_of_books-1 where book_name=" + book_name +
")")

                mydb.commit()
                print("\n")
                print("-----BOOK HAS LENT/ISSUED-----")
                print("\n")

            else:
                print("\n")
                print("There are no books named", book_name,
"available right now")
                print("\n")
        after_query()

```



```
else:
    print("ERROR: enter 1 or 2 only! ")
    continue
```

```
# RETURNING A BOOK
```

```
elif ch2 == 2:
    print("RETURNING A BOOK")
    print("if you wanna go back press 1")
    print(" ")
    print("if you wanna continue press 2")
    print(" ")
    a = int(input("enter your choice:"))
    if a == 2:
        roll_no = input("Enter roll no: ")
        book_name = str(input("Enter book name:"))
        print("If book is returned on same day as lend date, enter
date of returning as next day. ")
        date_of_return = str(input("Enter date of
returning(yyyy-mm-dd):"))
        cursor.execute(
            "update transact_lib set act_return_date=" +
date_of_return + " where book_name=" + book_name + "and
roll_no=" + roll_no + "")
        mydb.commit()
        print("\n")
        print("-----BOOK HAS BEEN RETURNED-----")
        print("\n")
    continue
```

```
#DISPLAY LENDING HISTORY
```

```
elif ch2 == 3:
    print("Proceeding to display lending history")
    print("\n")
    roll_no = input("Enter roll no: ")
    cursor.execute("select * from transact_lib where roll_no=" +
roll_no + "")
    print(roll_no, "S LENDING HISTORY: ")
    for i in cursor:
        print("\n")
        print(i)
    continue
```

#### #SEARCH BOOKS

```
elif ch2 == 4:
    print("Proceeding to search books")
    print("\n")
    print("""If you want to go back press 1
If you want to continue press 2""")
    b = int(input("Enter choice: "))
    if b == 1:
        continue

    elif b == 2:
        d = int(input("""Press 1 to display all books
Press 2 to display information about a particular book: """))

        if d == 1:
            cursor.execute("select * from books")
            print("\n")
            print("ALL BOOKS IN LIBRARY: ")
            for i in cursor:
```

```

        print("\n")
        print(i)

    elif d == 2:
        bookname = input("enter the name of the book, you need
information about: ")
        cursor.execute("select * from books where book_name
like '" + f"%{bookname}%" + "'")
        for i in cursor:
            print("\n")
            print("Information found: ")
            print("\n")
            print(i)
    else:
        print("ERROR: enter 1 or 2 only ")
        continue

# SEARCH PERSON

elif ch2 == 5:
    print("proceeding to search users")
    print("\n")
    print("""If you want to go back press 1
If you want to continue press 2""")
    b = int(input("Enter choice: "))
    if b == 1:
        continue

    elif b == 2:
        search_user = input("Enter the Name or Roll Number of
the user you want to search: ")
        if search_user.isdigit():

```

```

        cursor.execute("select roll_no, username from users
where roll_no ='" + search_user + "'")
        print('Users Found:')
        for i in cursor:
            print("\n")
            print(i)
    else:
        cursor.execute("select roll_no, username from users
where username like '" + f"%{search_user}%" + "'")
        print('Users Found:')
        print("\n")
        for i in cursor:
            print(i)
    else:
        print("ERROR: enter 1 or 2 only ")
        continue

```

#RATE BOOK

```

elif ch2 == 6:
    print("If you want to go back press 1
If you want to continue press 2 ")
    b=int(input("Enter choice: "))
    if b == 1:
        after_query()
        continue

    elif b == 2:
        cursor.execute(
            "create table if not exists rate_book(bookname
varchar(30) not null, foreign key(bookname) references
books(book_name), rating double, no_of_raters int)")
        bookname=input("Enter name of book to be rated: ")

```

```

value=float(input("Rate the book on a scale of 1 to 5: "))
if float(value)>5.0:
    print("ERROR: enter value below 5 only ")
elif float(value)<1.0:
    print("ERROR: enter value above 1 only ")
elif float(value)<=5.0 and float(value)>=1.0:
    cursor.execute("select * from rate_book where
bookname='"+ bookname + "'")
    cot=cursor.fetchone()

    if cot is not None:
        (x,y,z)=cot
        rating= value + z*y
        z=z+1
        y=rating / z
        cursor.execute(
            "update rate_book set rating={} where
bookname='{}'".format(y,bookname))
        cursor.execute(
            "update rate_book set no_of_raters={} where
bookname='{}'".format(z,bookname))
        mydb.commit()
        print("\n")
        print("RATING ADDED SUCCESSFULLY")
        continue

    elif cot is None:
        cursor.execute(
            "insert into rate_book
values('{}',{},{})".format(bookname,value,1))
        mydb.commit()
        print("\n")

```

```

        print("RATING ADDED SUCCESSFULLY")
        continue
    else:
        print("enter a number only")

    else:
        print("ERROR: enter 1 or 2 only ")
        continue

#DISPLAY RATING
elif ch2 == 7:
    print("""If you want to go back press 1
If you want to continue press 2 """)
    c=int(input("Enter choice: "))
    if c == 1:
        after_query()
        continue

    if c == 2:
        print("Do you want to display rating of particular book or all
books?")
        print("""
Press 1 to display rating of a particular book
Press 2 to display rating of all books """)
        d=int(input("Enter choice: "))

        if d == 1:
            print("Proceeding to display rating of particular book")
            bookname=str(input("Enter book name: "))
            cursor.execute(
                "select * from rate_book where
bookname='"+bookname+"'")

```

```
for i in cursor:
    print("\n")
    print("RATING OF",bookname,"IS: ")
    print(i)
    continue
```

```
elif d == 2:
    print("Proceeding to display ratings of all books")
    cursor.execute("select * from rate_book")
    print("RATING OF ALL BOOKS")
    for j in cursor:
        print("\n")
        print(j)
        continue
```

```
else:
    print("ERROR: enter 1 or 2 only ")
    continue
```

# ADD A BOOK

```
if ch2 == 8:
    print("""If you want to go back press 1
If you want to continue press 2""")
    b = int(input("Enter choice: "))
    if b == 1:
        continue
    elif b == 2:
        print("Proceeding to add books")
        print("\n")
        print("PLEASE FILL BOOK DETAILS")
```

```

        book_name = str(input("enter book name: "))
        no_of_books = input("enter number of books: ")
        authors_name = str(input("Enter the authors name: "))
        cursor.execute(
            "insert into books values('" + book_name + "','" +
no_of_books + "','" + authors_name + "')"
        )
        mydb.commit()
        print("\n")
        print("-----BOOK ADDED SUCESSFULLY-----")
        after_query()
        continue

```

#### # UPDATE BOOK DETAILS

```

elif ch2 == 9:
    print("""If you want to go back press 1
If you want to continue press 2""")
    b = int(input("Enter choice: "))
    if b == 1:
        continue
    elif b == 2:
        print("Proceeding to update book details")
        print("\n")
        print("""press 1 to update book name
Press 2 to update quantity of books
Press 3 to update author name""")
        ch3 = int(input(" 1 / 2 / 3 : "))
        present_bookname = input("Enter the BOOK NAME whose
detais are to be updated: ")
        if ch3 == 1:
            new_bookname = input("Enter new name for book: ")
            cursor.execute(

```



```
        "update books set book_name=" + new_bookname + "
where book_name=" + present_bookname + """)
        mydb.commit()
        print("\n")
        print("-----THE BOOK'S NAME HAVE BEEN
UPDATED-----")
```

```
    elif ch3 == 2:
        new_no_of_books = input("Enter new number of books
(quantity) : ")
        cursor.execute(
            "update books set no_of_books=" + new_no_of_books
+ " where book_name=" + present_bookname + """)
        mydb.commit()
        print("\n")
        print("-----THE NUMBER OF BOOKS HAVE BEEN
UPDATED-----")
```

```
    elif ch3 == 3:
        new_authors_name = input("Enter new author(s)
name(s): ")
        cursor.execute(
            "update books set authors_name=" +
new_authors_name + " where book_name=" + present_bookname +
""")
        mydb.commit()
        print("\n")
        print("-----THE AUTHOR'S NAME(S) HAVE BEEN
UPDATED-----")
```

```
    else:
        print("ERROR: please enter number between 1 and 3")
```

```
after_query()
continue
```

## #DELETE A BOOK

```
elif ch2 == 10:
    print("""If you want to go back press 1
          If you want to continue press 2""")
    b = int(input("Enter choice: "))
    if b == 1:
        continue
    elif b == 2:
        book_name = input("Enter the BOOK NAME to be deleted:
")
        cursor.execute("delete from transact_lib where
book_name='" + book_name + "'")
        cursor.execute("delete from books where book_name='" +
book_name + "'")
        print("\n")
        print("-----"+book_name.upper()+" DELETED-----")
        after_query()
    continue
```

## #DISPLAY LENDING HISTORY OF SOME/ALL PEOPLE

```
elif ch2 == 11:
    print("""If you want to go back press 1
          If you want to continue press 2""")
    b = int(input("Enter choice: "))
    if b == 1:
        after_query()
    elif b == 2:
```

```

    print("Proceeding to display lending history")
    print("\n")
    print("Do you want to display only individual user's lending
history or everyone's? ")
    c = int(input("Press 1 to display individual user's lending
history
Press 2 to display everyone's lending history: "))

    if c == 1:
        roll_no = input("Enter roll no: ")
        cursor.execute("select * from transact_lib where roll_no="
+ roll_no + ")")
        for i in cursor:
            print("\n")
            print(roll_no, "S LENDING HISTORY: ")
            print("\n")
            print(i)

    elif c == 2:
        cursor.execute("select * from transact_lib")
        print("\n")
        print("EVERYONE'S LENDING HISTORY")
        for j in cursor:
            print("\n")
            print(j)

    else:
        print("ERROR: enter 1 or 2 only! ")

else:
    print("ERROR: enter 1 or 2 only! ")
    continue

```

```
#DISPLAY USERS WHO HAVE NOT RETURNED
```

```
elif ch2 == 12:
    print("Proceeding to display list of users who have not returned
book")
    print("\n")
    cursor.execute("select * from transact_lib where
lend_date=act_return_date")
    print("USERS WHO HAVE NOT RETURNED")
    for k in cursor:
        print("\n")
        print(k)
        continue
```

```
# LOGOUT
```

```
elif ch2 == 0:
    print("Do you want to logout?")
    a = int(input("""Enter 1 to go back
Enter 2 to continue: """))

    if a == 1:
        continue
    elif a == 2:
        user = None
        print("\n")
        print("""-----YOU ARE LOGGED OUT-----""")

    else:
        print("""Invalid input
""")
```

```

while True:
    print("\n")
    print("""press 1 to create a new account (signup)
Press 2 if you are a existing user (login)
Press 3 to delete account
Press 4 to exit """)

    ch1 = int(input(" 1 / 2 / 3 / 4 : "))

    # user puts 1 or 2 to sign up or login

    # CREATE NEW ACCOUNT

    if ch1 == 1:
        print("\n")
        global roll_no
        roll_no = input("roll number: ")
        print("\n")
        global username
        username = input("USERNAME: ")
        print("Hello, please create a PASSWORD")
        password = input("PASSWORD: ")

        cursor.execute("insert into users values('" + roll_no + "','" +
username + "','" + password + "'")
        # note: "'+ +' syntax is used to add data in users table

        mydb.commit()
        print("YOUR ACCOUNT HAS BEEN CREATED")
        print("\n")

```

```
print("now login into your account to continue\n")
```

```
# LOGIN
```

```
elif ch1 == 2:
```

```
    roll_no = input("roll number: ")
```

```
    username = input("USERNAME: ")
```

```
    cursor.execute("select username from users where username=" + username + " and roll_no=" + roll_no + "")  
    pot = cursor.fetchone()
```

```
    if pot is not None:
```

```
        print("hello", username, "please insert PASSWORD")
```

```
        password = input("PASSWORD: ")
```

```
        cursor.execute("select password from users where password=" + password + "")
```

```
        login = cursor.fetchone()
```

```
        if login is not None:
```

```
            print("\n")
```

```
            print("""-----YOU ARE LOGGED IN-----
```

```
            """)
```

```
            after_login(login)
```

```
        if login is None:
```

```
            print("Invalid password for roll no ",roll_no,". Please retry.")
```

```
    if pot is None:
```

```
print("Invalid username for roll no ",roll_no,". Please retry.")
```

```
#DELETE ACCOUNT
```

```
elif ch1 == 3:
```

```
    print("Delete account?")
```

```
    a = int(input("""Enter 1 to go back  
Enter 2 to continue: """))
```

```
    if a == 1:
```

```
        cursor.execute("select * from users")
```

```
        for i in cursor:
```

```
            print(i)
```

```
    elif a == 2:
```

```
        cursor.execute("select * from users")
```

```
        for i in cursor:
```

```
            print(i)
```

```
    print("Enter deleting details: ")
```

```
    roll_no = input("Enter roll no: ")
```

```
    username = input("Enter username: ")
```

```
    password = input("Enter password: ")
```

```
    cursor.execute("""delete from users where roll_no="" + roll_no + ""  
                    and username="" + username + "" and  
password="" + password + """"")  
    mydb.commit()
```

```
    print("\n")
```

```
    print("-----ACCOUNT DELETED SUCCESSFULLY-----")
```

```
    print("\n")
```

```
cursor.execute("select * from users")
for i in cursor:
    print(i)
    continue
```

```
#EXIT
```

```
elif ch1 == 4:
    break
```

```
else:
    print("""invalid input
          """)
```



# Outputs

## Signup

```
press 1 to create a new account (signup)
Press 2 if you are a existing user (login)
Press 3 to delete account
Press 4 to exit
1 / 2 / 3 / 4 : 1
```

```
roll number: 5
```

```
USERNAME: user
Hello, please create a PASSWORD
PASSWORD: pass
YOUR ACCOUNT HAS BEEN CREATED
```

```
now login into your account to continue
```

## Login

```
press 1 to create a new account (signup)
Press 2 if you are a existing user (login)
Press 3 to delete account
Press 4 to exit
1 / 2 / 3 / 4 : 2
roll number: 5
USERNAME: user
hello user please insert PASSWORD
PASSWORD: pass
```

```
-----YOU ARE LOGGED IN-----
```



## Delete user

```
press 1 to create a new account (signup)
Press 2 if you are a existing user (login)
Press 3 to delete account
Press 4 to exit
```

```
1 / 2 / 3 / 4 : 3
Delete account?
```

```
Enter 1 to go back
Enter 2 to continue: 2
('1', 'ana', 'random')
('10', 'stuart', 'student')
('2', 'uv', 'notuvrays')
('3', 'han_solo', '4to7')
('4', 'abc', 'abc456')
('5', 'xyz', 'xyz123')
('6', 'jo', 'jolly')
('7', 'tia_cosmos', 'cosmic')
('8', 'henry', 'kowalski')
('9', 'randy', 'miranda')
Enter deleting details:
```

```
Enter roll no: 10
```

```
Enter username: stuart
```

```
Enter password: student
```

```
-----ACCOUNT DELETED SUCCESSFULLY-----
```

## Admin features

```
login with admin permission to access special admin features.
```

```
or continue as user
```

```
Do you have admin permission? (y/n): y
```

```
Press 0 to Log out
```

```
Press 1 to lend/issue book
```

```
Press 2 to return book
```

```
Press 3 to display your lending history
```

```
Press 4 to search books
```

```
Press 5 to search user
```

```
Press 6 to rate book
```

```
Press 7 to display ratings of book
```

```
Special Admin Features:
```

```
Press 8 to add book
```

```
Press 9 to update book
```

```
press 10 to delete book
```

```
press 11 to see lending history of some or all people
```

```
Press 12 to display users who have not returned books
```

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : |
```



## Add books

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 8
```

```
If you want to go back press 1
If you want to continue press 2
Enter choice: 2
Proceeding to add books
```

```
PLEASE FILL BOOK DETAILS
enter book name: book
enter number of books: 6
Enter the authors name: author
```

```
-----BOOK ADDED SUCESSFULLY-----
Here are all the books in the library:
('book', 6, 'author')
('book1', 1, 'author1')
('book2', 2, 'author2')
('book3', 3, 'author3')
```

## Update book name

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 9
```

```
If you want to go back press 1
If you want to continue press 2
Enter choice: 2
Proceeding to update book details
```

```
press 1 to update book name
Press 2 to update quantity of books
Press 3 to update author name
1 / 2 / 3 : 1
Enter the BOOK NAME whose detais are to be updated: book
Enter new name for book: new book
```

```
-----THE BOOK'S NAME HAVE BEEN UPDATED-----
Here are all the books in the library:
('book1', 1, 'author1')
('book2', 2, 'author2')
('book3', 3, 'author3')
('new book', 6, 'author')
```



## Update book quantity

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 9
```

```
If you want to go back press 1
If you want to continue press 2
Enter choice: 2
Proceeding to update book details
```

```
press 1 to update book name
Press 2 to update quantity of books
Press 3 to update author name
 1 / 2 / 3 : 2
Enter the BOOK NAME whose details are to be updated: new book
Enter new number of books (quantity) : 3
```

```
-----THE NUMBER OF BOOKS HAVE BEEN UPDATED-----
Here are all the books in the library:
('book1', 1, 'author1')
('book2', 2, 'author2')
('book3', 3, 'author3')
('new book', 3, 'author')
```

## Update book's author

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 9
```

```
If you want to go back press 1
If you want to continue press 2
Enter choice: 2
Proceeding to update book details
```

```
|
```

```
press 1 to update book name
Press 2 to update quantity of books
Press 3 to update author name
 1 / 2 / 3 : 3
Enter the BOOK NAME whose details are to be updated: new book
Enter new author(s) name(s): new author
```

```
-----THE AUTHOR'S NAME(S) HAVE BEEN UPDATED-----
Here are all the books in the library:
('book1', 1, 'author1')
('book2', 2, 'author2')
('book3', 3, 'author3')
('new book', 3, 'new author')
```



## Delete book

```
Press 0 to Log out
Press 1 to lend/issue book
Press 2 to return book
Press 3 to display your lending history
Press 4 to search books
Press 5 to search user
Press 6 to rate book
Press 7 to display ratings of book

Special Admin Features:
Press 8 to add book
Press 9 to update book
press 10 to delete book
press 11 to see lending history of some or all people
Press 12 to display users who have not returned books

0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 10
If you want to go back press 1
    If you want to continue press 2

Enter choice: 2

Enter the BOOK NAME to be deleted: A Tale of Two Cities

-----A TALE OF TWO CITIES DELETED-----
Here are all the books in the library:
('Alice in Wonderland', 2, 'Lewis Carrol')
('Gulliver's Travels', 1, 'Jonathan Swift')
```

## Lend book

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 : 1
If you want to go back press 1
If you want to continue press 2

Enter choice: 2

Proceeding to lend/issue book

Enter roll no: 1

Enter name of book: Lores From The Ancient World

Enter name of author of book: Sammie Wane

Enter date book was issued/lent (yyyy-mm-dd): 2020-12-03
Enter expected return date as after 21 days from lend date
If book not returned then enter actual return date same as lend date

Enter (expected) date of return (yyyy-mm-dd): 2020-12-24

Enter (actual) date of return (yyyy-mm-dd): 2020-12-03
```



```
-----BOOK HAS LENT/ISSUED-----
```

Here are all the books in the library:

```
('Alph-Art', 5, 'Herge')
('Lores From The Ancient World', 1, 'Sammie Wane')
('Star Wars 7 movie novel', 3, 'LucasFilmsANON')
('TGOEM', 4, 'ANON')
('The ABC Murders', 1, 'Agatha Christie')
('The Book Thief', 2, 'Markus Zusak')
```

## Return book

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 2
```

RETURNING A BOOK

if you wanna go back press 1

if you wanna continue press 2

enter your choice:2

Enter roll no: 30

Enter book name:A Tale of Two Cities

If book is returned on same day as lend date, enter date of returning as next day.

Enter date of returning(yyyy-mm-dd):2020-11-11

```
-----BOOK HAS BEEN RETURNED-----
```

## Display users who have not returned books

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 12
```

Proceeding to display list of users who have not returned book

USERS WHO HAVE NOT RETURNED

```
('8', 'Alph-Art', 'Herge', datetime.date(2020, 11, 29), datetime.date(2020, 12, 20), datetime.date(2020, 11, 29))
```

```
('9', 'The ABC Murders', 'Agatha Christie', datetime.date(2020, 11, 29), datetime.date(2020, 12, 20),
datetime.date(2020, 11, 29))
```

```
('5', 'The Glass Bird Girl', 'Esme Kerr', datetime.date(2020, 11, 29), datetime.date(2020, 12, 20), datetime.date(2020,
11, 29))
```

```
('5', 'The Shining', 'Stephen King', datetime.date(2020, 11, 29), datetime.date(2020, 12, 20), datetime.date(2020, 11,
29))
```

```
('1', 'The Kudryavka Order', 'Hyouka', datetime.date(2020, 11, 29), datetime.date(2020, 12, 20), datetime.date(2020,
11, 29))
```



# Display lending history

## Of all users

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 11
```

```
If you want to go back press 1  
If you want to continue press 2
```

```
Enter choice: 2  
Proceeding to display lending history
```

```
Do you want to display only individual user's lending history or everyone's?
```

```
Press 1 to display individual user's lending history  
Press 2 to display everyone's lending history: 2
```

### EVERYONE'S LENDING HISTORY

```
('5', 'The ABC Murders', 'Agatha Christie', datetime.date(2020, 11, 19), datetime.date(2020, 12, 10),  
datetime.date(2020, 11, 19))
```

```
('6', 'TGOEM', 'ANON', datetime.date(2020, 11, 19), datetime.date(2020, 12, 10), datetime.date(2020, 11, 19))
```

```
('5', 'The Book Thief', 'Markus Zusak', datetime.date(2020, 11, 26), datetime.date(2020, 12, 17), datetime.date(2020,  
12, 1))
```

```
('8', 'Alph-Art', 'Herge', datetime.date(2020, 11, 29), datetime.date(2020, 12, 20), datetime.date(2020, 11, 29))
```

```
('9', 'The ABC Murders', 'Agatha Christie', datetime.date(2020, 11, 29), datetime.date(2020, 12, 20),  
datetime.date(2020, 11, 29))
```

```
('5', 'The Glass Bird Girl', 'Esme Kerr', datetime.date(2020, 11, 29), datetime.date(2020, 12, 20), datetime.date(2020,  
11, 29))
```

## Of individual user

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 11
```

```
If you want to go back press 1  
If you want to continue press 2
```

```
Enter choice: 2  
Proceeding to display lending history
```

```
Do you want to display only individual user's lending history or everyone's?
```

```
Press 1 to display individual user's lending history  
Press 2 to display everyone's lending history: 1
```

```
Enter roll no: 6
```

### 6 'S LENDING HISTORY:

```
('6', 'TGOEM', 'ANON', datetime.date(2020, 11, 19), datetime.date(2020, 12, 10), datetime.date(2020, 12, 4))
```



## Search all books

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 4
```

```
Proceeding to search books
```

```
If you want to go back press 1
```

```
If you want to continue press 2
```

```
Enter choice: 2
```

```
Press 1 to display all books
```

```
Press 2 to display information about a particular book: 1
```

```
ALL BOOKS IN LIBRARY:
```

```
('book1', 1, 'author1')
```

```
('book2', 2, 'author2')
```

```
('book3', 3, 'author3')
```

## Search a particular book

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 4
```

```
Proceeding to search books
```

```
If you want to go back press 1
```

```
If you want to continue press 2
```

```
Enter choice: 2
```

```
Press 1 to display all books
```

```
Press 2 to display information about a particular book: 2
```

```
enter the name of the book, you need information about: book1
```

```
Information found:
```

```
('book1', 1, 'author1')
```



## Search a user

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 5
proceeding to search users

If you want to go back press 1
If you want to continue press 2

Enter choice: 2

Enter the Name or Roll Number of the user you want to search: ni
Users Found:

('20', 'nitu')

('30', 'nitya')
```

## Rate a book

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 : 6
If you want to go back press 1
If you want to continue press 2

Enter choice: 2

Enter name of book to be rated: The Book Thief

Rate the book on a scale of 1 to 5: 5

RATING ADDED SUCCESSFULLY
```

## Search Rating

### Of individual Book

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 : 7
If you want to go back press 1
If you want to continue press 2

Enter choice: 2
Do you want to display rating of particular book or all books?

Press 1 to display rating of a particular book
Press 2 to display rating of all books

Enter choice: 1
Proceeding to display rating of particular book

Enter book name: The Book Thief

RATING OF The Book Thief IS:
('The Book Thief', 4.75, 2)
```



## Of All books

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 : 7
If you want to go back press 1
If you want to continue press 2

Enter choice: 2
Do you want to display rating of particular book or all books?

Press 1 to display rating of a particular book
Press 2 to display rating of all books

Enter choice: 2
Proceeding to display ratings of all books
RATING OF ALL BOOKS

('TGOEM', 4.5, 2)

('The Book Thief', 4.75, 2)
```

## Logout and exit

```
0 / 1 / 2 / 3 / 4 / 5 / 6 / 7 / 8 / 9 / 10 / 11 / 12 : 0
Do you want to logout?

Enter 1 to go back
Enter 2 to continue: 2

-----YOU ARE LOGGED OUT-----
press 1 to create a new account (signup)
Press 2 if you are a existing user (login)
Press 3 to delete account
Press 4 to exit

1 / 2 / 3 / 4 : 4

In [3]: |
```

# Bibliography

The following websites and book(s) were helpful during my project writing process:

1. Computer Science with Python. Class XII. By Sumita Arora
2. <https://www.geeksforgeeks.org/>
3. <https://www.w3schools.com/>
4. <https://www.cs4school.com/cbse/python-project-for-class-12>



# **Conclusion**

The Library Management System allows the user to store the book details and the person's details.

This software allows storing the details of all the data related to library.

Hence, we have completed our project without any problems. We have done various research for this project and this system is designed using Python in the front end and MySQL in the backend database

This has been developed by considering all the needs given in the project and by through interaction with the users of the system.

