Karnataka Law Society's

GOGTE INSTITUTE OF TECHNOLOGY

Udyambag Belagavi -590008
Karnataka, India.

A Course Project Report on

"LIBRARY MANAGEMENT SYSTEM"

Submitted for the requirements of 4th semester B.E. in ISE for

"PYTHON PROGRAMING (18ISL46)"

Submitted by Batch 20

NAME	USN
1. Megha Magadumakar	2GI19IS024
2. Sanjana Sunadholi	2GI19IS047
3. Shweta Naik	2GI19IS050
4. Tejaswini Naganoor	2GI19IS055

Under the guidance of

Dr. Kiran K Tangod

KLS, GIT Belagavi

Academic Year 2020-2021 (Even semester)

Karnataka Law Society's
GOGTE INSTITUTE OF TECHNOLOGY
Udyambag Belagavi -590008
Karnataka, India.

Department of Information Science and Engineering

Certificate

This is to certify that the Course Project work titled **"Library Management System"** carried out by Megha Magadumakar**,** Sanjana Sunadholi, Shweta Naik, Tejaswini Naganoor bearing is 2GI19IS024, 2GI19IS047, 2GI19IS050, 2GI19IS055 submitted in partial fulfilment of the requirements for 4th semester B.E. in **INFORMATION SCIENCE AND ENGINEERING,** Visvesvaraya Technological University, Belagavi. It is certified that all corrections/ suggestions indicated have been incorporated in the report. The course project report has been approved as it satisfies the academic requirements prescribed for the said degree.

Date: 30/7/2021 Signature of Guide

Place: Belagavi Dr. Kiran K Tangod

KLS, GIT, Belagavi

Name of the Examiner Signature of the Examiners

1.______________________ 1. ______________________

2.______________________ 2. ______________________

CONTENT:

1. ABSTRACT

Library management system is a project which aims in developing a computerized system to maintain all the daily work of library .This project has many features which are generally not available in normal library management systems like facility of user login and a facility of teachers login .It also has a facility of admin login through which the admin can monitor the whole system .It also has facility of an online notice board where teachers can student can put up information about workshops or seminars being held in our colleges or nearby colleges and librarian after proper verification from the concerned institution organizing the seminar can add it to the notice board . It has also a facility where student after logging in their accounts can see list of books issued and its issue date and return date and also the students can request the librarian to add new books by filling the book request form. The librarian after logging into his account i.e., admin account can generate various reports such as student report, issue report, teacher report and book report.

Overall, this project of ours is being developed to help the students as well as staff of library to maintain the library in the best way possible and also reduce the human efforts.

2. INTRODUCTION:

The project “Library management System” is developed with using python as front-end and MySQL as back-end.

This system mainly focuses on basic operations of library like login, register, adding and deleting new member, new book, updating new information, searching books and members to issue and return book.

2.1. Project description

Library Management System is a computerized system which helps user(librarian) to manage the library daily activity in electronic format. It reduces the risk of paper work such as file lost, file damaged and time consuming. It can help user to manage the transaction or record more effectively and time-saving.

2.2. Problem statement

The problem occurred before having computerized system includes:

- File lost When computerized system is not implemented file is always lost because of human environment. Sometimes due to some human error there may be a loss of records.

- File damaged When a computerized system is not their file is always lost due to some accident like spilling of water by some member on file accidentally. Besides some natural disaster like floods or fires may also damage the files.

- Difficult to search record When there is no computerized system there is always a difficulty in searching of records if the records are large in number.

- Space consuming After the number of records become large the space for physical storage of file and records also increases if no computerized system is implemented.

- Cost consuming As there is no computerized system the to add each record paper will be needed which will increase the cost for the management of library.

3. PROJECT OBJECTIVES

- Improvement in control and performance the system is developed to cope up with the current issues and problems of library. The system can add user, validate user and is also bug free.

- Save cost After computerized system is implemented less human force will be required to maintain the library thus reducing the overall cost.

- Save time Librarian is able to search record by using few clicks of mouse and few search keywords thus saving his valuable time.

- The software keeps track of all the information about the books and their complete details.

- The system contain database where all the information will be stored safely.

3.1. Project scope

- Any education institute & government offices can make use of it for providing information about author, content of the available books etc.

- This project application will keep track of all the books and library information.

- The software will be able to handle all the necessary information.

4. SYSTEM REQUIREMENTS

4.1. Hardware Requirements

- Intel core i5 10th generation is used as a processor because it is fast than other processors an provide reliable and stable and we can run our pc for longtime. By using this processor, we can keep on developing our project without any worries.

- Ram 1 GB is used as it will provide fast reading and writing capabilities and will in turn support in processing.

4.2. Software Requirements

- Operating system- Windows 10 is used as the operating system as it is stable and supports more features and is more user friendly.
- Database “lms” is used as database as it easy to maintain and retrieve records by simple queries which are in English language which are easy to understand and easy to write.
- Development tools and Programming language- “PYTHON” is used to write the whole code.

4.3. Operation environment

PROCESSOR	INTEL CORE PROCESSOR OR BETTER PERFORMANCE
OPERATING SYSTEM	WINDOWS10
MEMORY	1GB RAM OR MORE
HARD DISK SPACE	MINIMUM 3 GB FOR DATABASE USAGE FOR FUTURE
DATABASE	MY SQL

5. METHODOLOGY

5.1. FRONT END

5.1.1. Introduction

The back end is designed using MySQL which is used to design the databases

PYTHON-

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

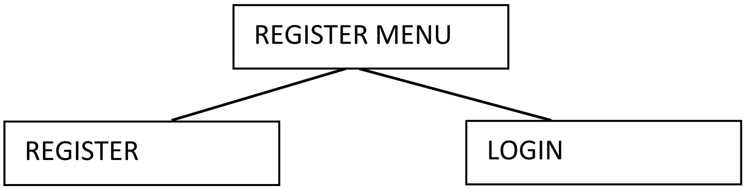
Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting. Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3. Python 2 was discontinued with version 2.7.18 in 2020.

5.1.2. Modules

For Library Management System it is divided into the following Modules:

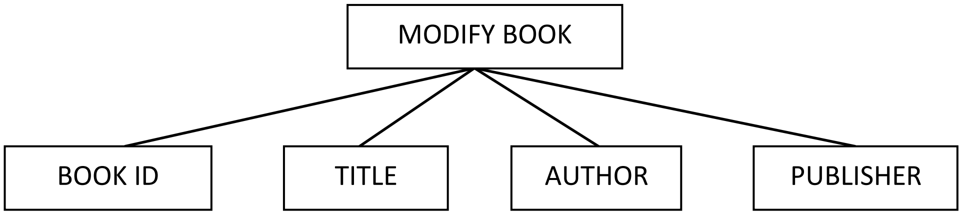
1. **Library used:**
   We have used,
   - MySql Connector : This module is used to connect mysql database to fetch/add data into database.
   - DateTime : To set date and get current date and update the details on the tables.
2. **Register :** with this feature we can add staff details Staff name, Staff ID, Date of Birth, Mobile Number, Email Address and Address of the staff.



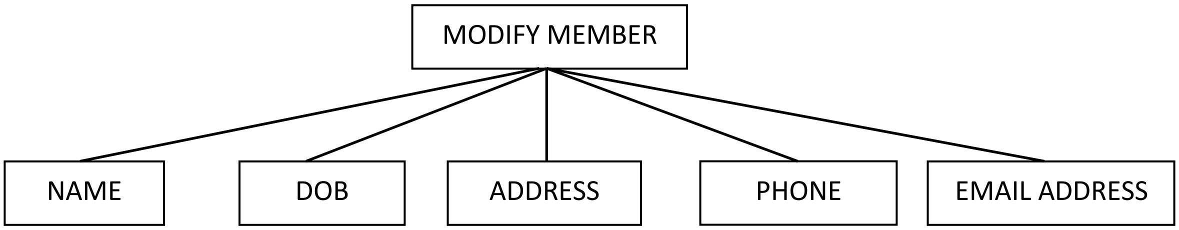
3. **Login :** with this staff/ librarian enters to the main menu, if and only if username and passwords are correct. We have used username as staff email address and password as staff ID.
4. **Add Member :** with this feature we can add member details member name, member ID, Date of Birth, Mobile Number, Email Address and Address of the Member.

5. **Add Book :** with this feature we can add books in LMS system with book details like Title, Book ID Author, Publisher, Price, and for each entry status is automatically updated to “available”.

6. **Update Book :** Update the previous entered Book details like Book ID, Title, Author, Publisher, Price.



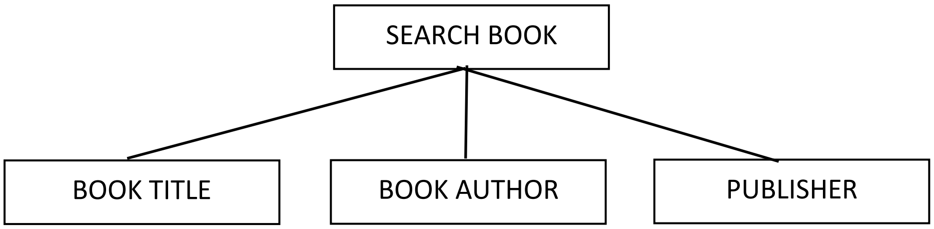
7. **Update Member :** Update the Member details like name, dob, address, phone, email address.



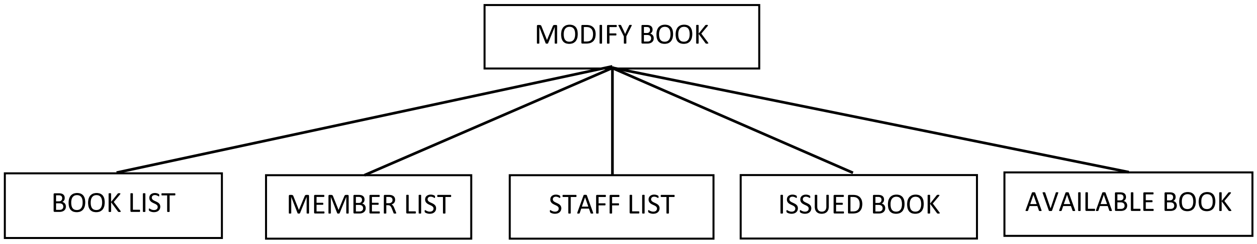
8. **Issue Book:** This will record the details on the Book issued to whom. This Feature stores the details of Book ID, Member ID , Issue Date and Return Date of book. After issue book status is automatically updated as “issued”.

9. **Return Book :** Update the Book Transactions on returned by the member. After issue book status is automatically updated as “issued”.

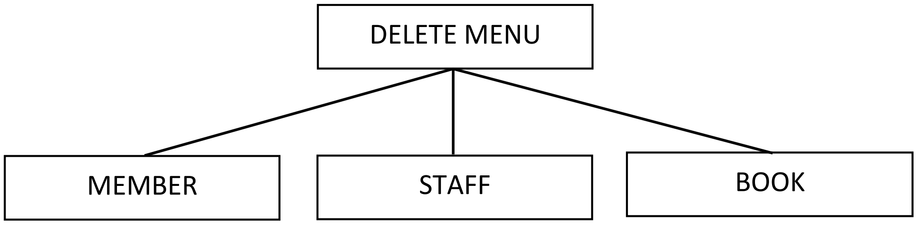
10. **Search Menu:** Search The Books which are available in the System. Based on thir title, author name and publisher name.



11. **Report Menu:** Retrieve all book list, member list, staff list, issued book list and available book list from database.



12. **Delete Menu:** Delete the information and activity of member, book and staff from the system.



5.2. BACK END

5.2.1. Introduction

The back end is designed using MySQL which is used to design the databases

MYSQL-

MySQL ("My S-Q-L", officially, but also called "My Sequel") is (as of July 2013) the world's second most widely used open-source relational database management system (RDBMS). It is named after co-founder Michael Widenius daughter, My. The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open-source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open-source projects that require a full-featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality.

Applications which use MySQL databases include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr, and YouTube

5.2.2. Database Mysql Connectivity

MySQL Database

To be able to experiment with the code examples in this tutorial, you should have MySQL installed on your computer.

You can download a free MySQL database at https://www.mysql.com/downloads/.

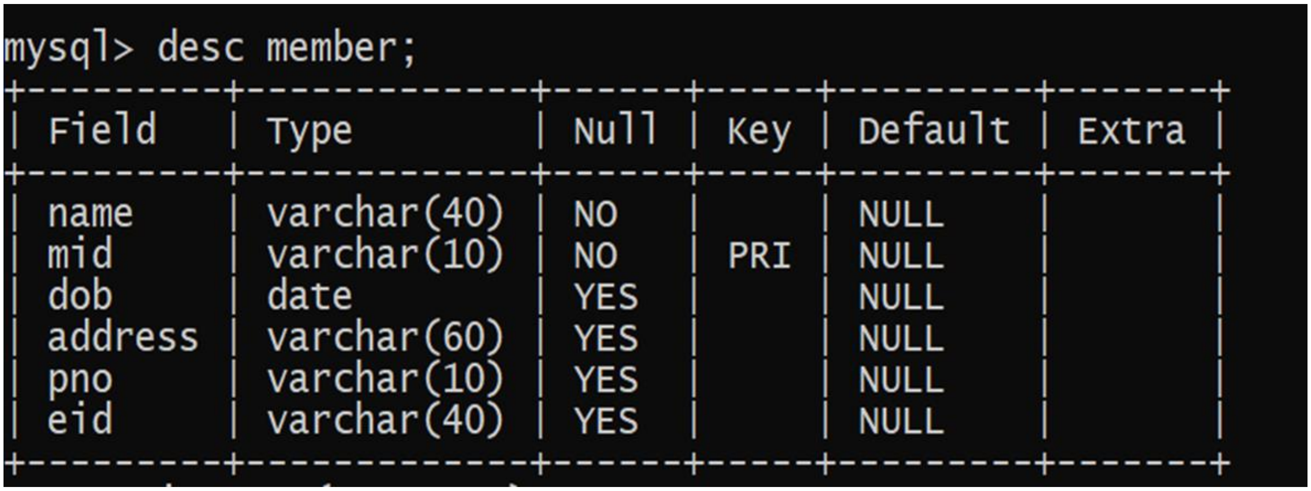
Install MySQL Driver

- Python needs a MySQL driver to access the MySQL database.
- We had used PIP to install "MySQL Connector".
- PIP is most likely already installed in your Python environment.
- Used the username and password from your MySQL database:
- Code to connect mysql and python

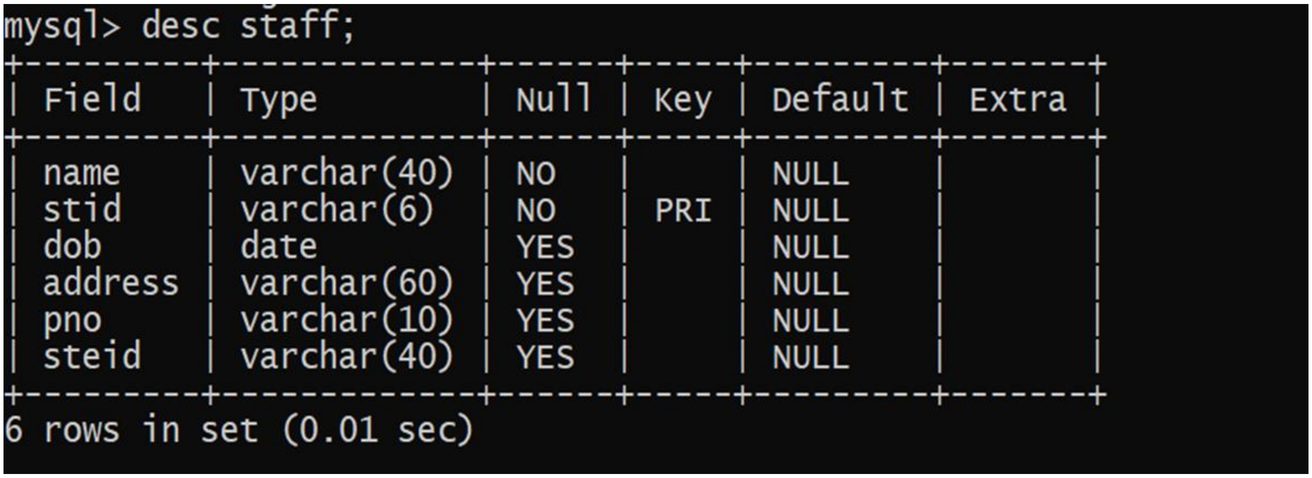
```
import mysql.connector
mydb = mysql.connector.connect(
        host="localhost",
        user="yourusername",
        password="yourpassword"
        )
print(mydb)
```

6. SYSTEM DESIGN

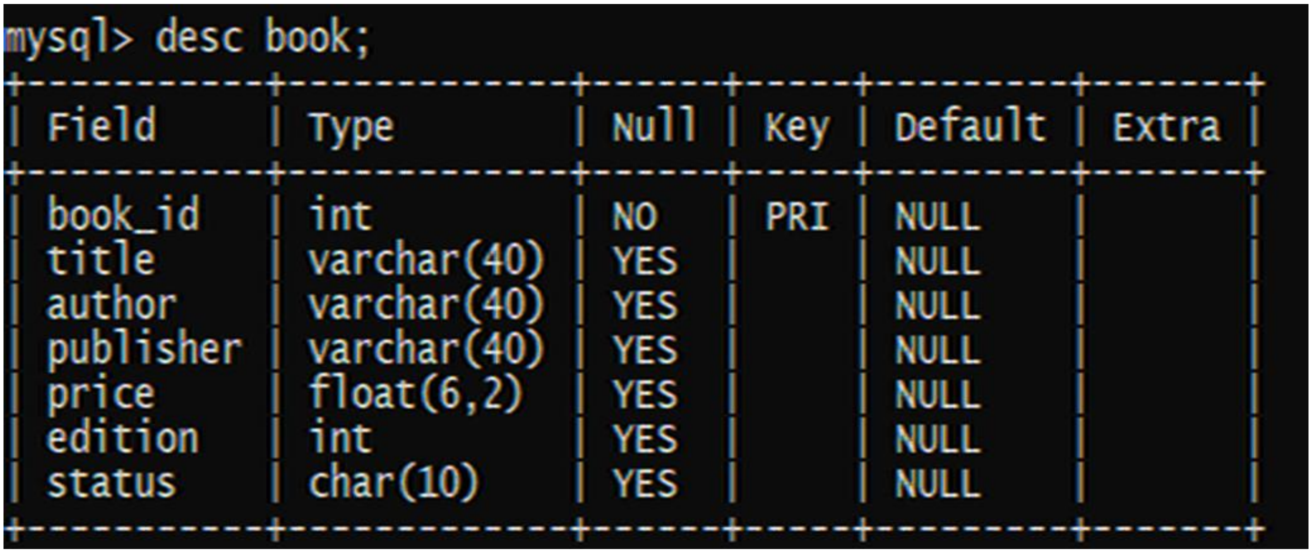
TABLES



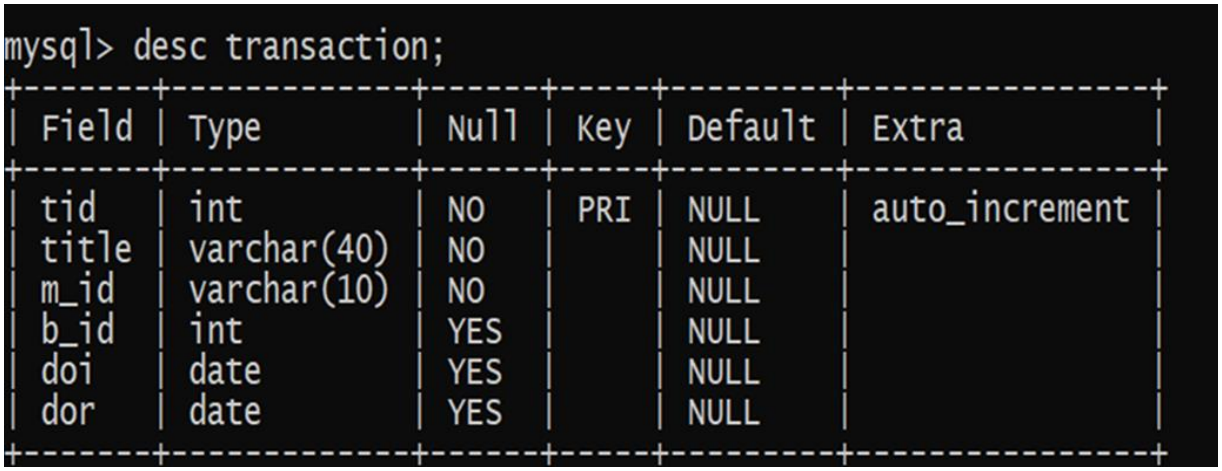
```
mysql> desc member;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| name    | varchar(40) | NO   |     | NULL    |       |
| mid     | varchar(10) | NO   | PRI | NULL    |       |
| dob     | date        | YES  |     | NULL    |       |
| address | varchar(60) | YES  |     | NULL    |       |
| pno     | varchar(10) | YES  |     | NULL    |       |
| eid     | varchar(40) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
```



```
mysql> desc staff;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| name    | varchar(40) | NO   |     | NULL    |       |
| stid    | varchar(6)  | NO   | PRI | NULL    |       |
| dob     | date        | YES  |     | NULL    |       |
| address | varchar(60) | YES  |     | NULL    |       |
| pno     | varchar(10) | YES  |     | NULL    |       |
| steid   | varchar(40) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

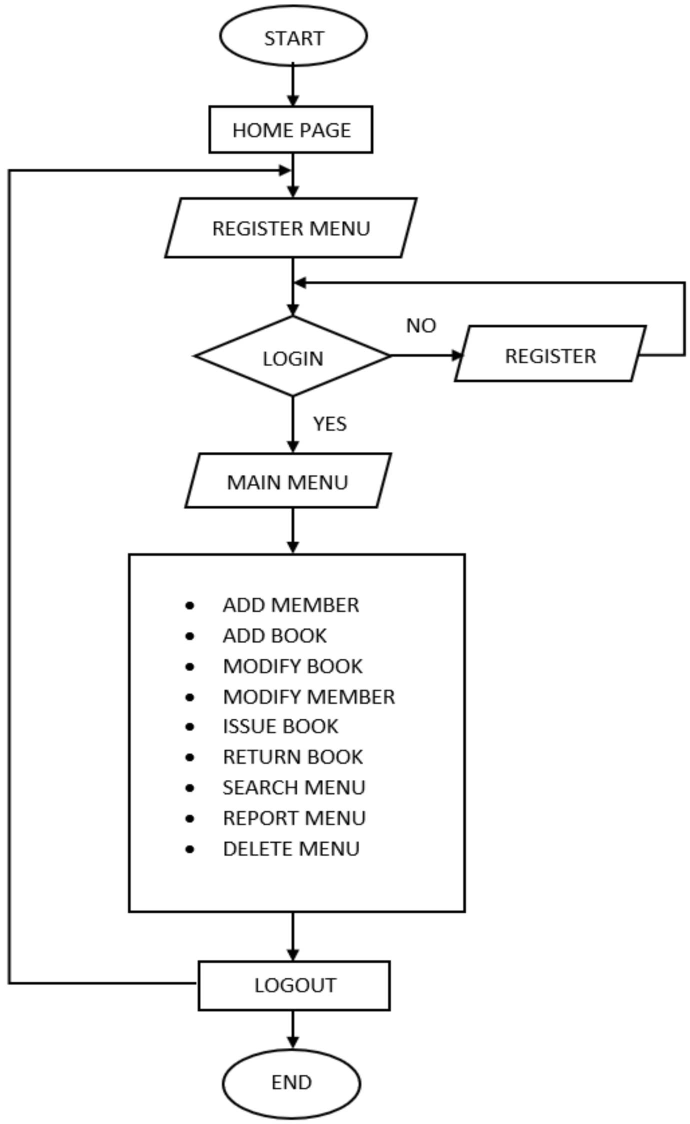


```
mysql> desc book;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| book_id   | int         | NO   | PRI | NULL    |       |
| title     | varchar(40) | YES  |     | NULL    |       |
| author    | varchar(40) | YES  |     | NULL    |       |
| publisher | varchar(40) | YES  |     | NULL    |       |
| price     | float(6,2)  | YES  |     | NULL    |       |
| edition   | int         | YES  |     | NULL    |       |
| status    | char(10)    | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

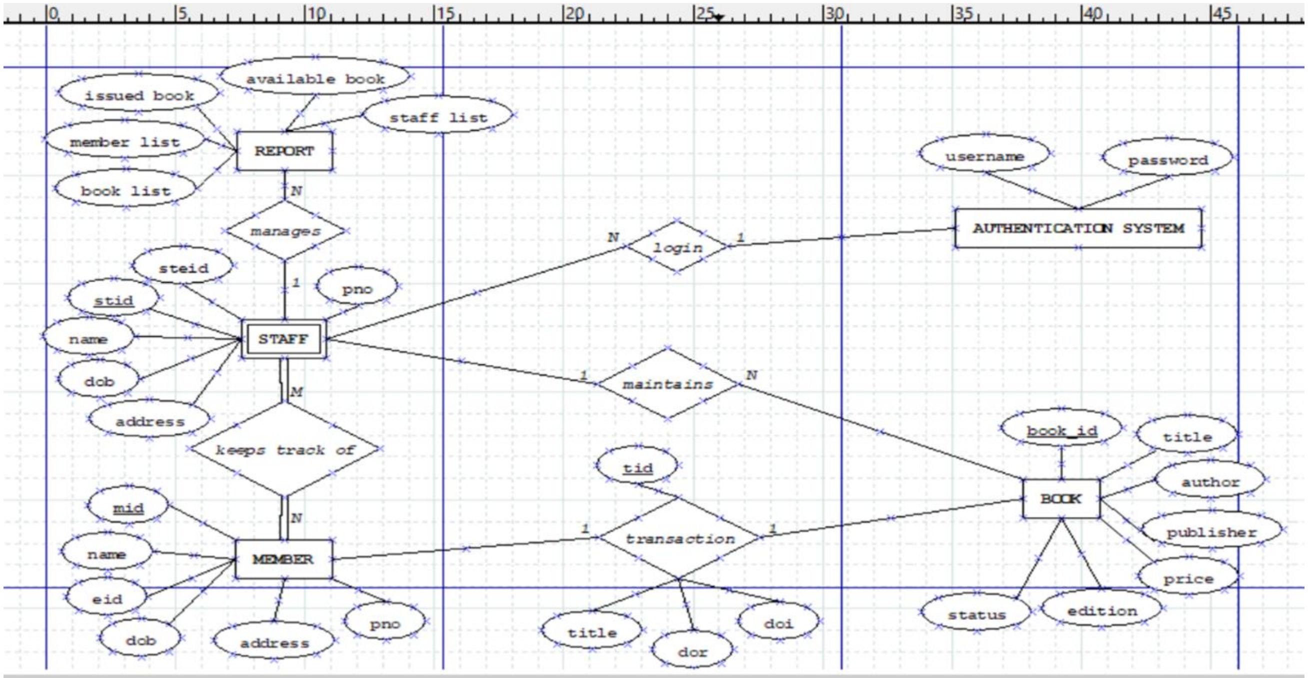


```
mysql> desc transaction;
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| tid   | int         | NO   | PRI | NULL    | auto_increment |
| title | varchar(40) | NO   |     | NULL    |                |
| m_id  | varchar(10) | NO   |     | NULL    |                |
| b_id  | int         | YES  |     | NULL    |                |
| doi   | date        | YES  |     | NULL    |                |
| dor   | date        | YES  |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+
```

6.1. SYSTEM FLOW



6.2. ER DIAGRAM



6.3. SCHEMA DIAGRAM

STAFF

stid	name	dob	address	pno	steid

MEMBER

mid	name	dob	address	pno	eid

BOOK

book_id	title	author	publisher	price	edition	status

TRANSACTION

tid	title	m_id	b_id	doi	dor

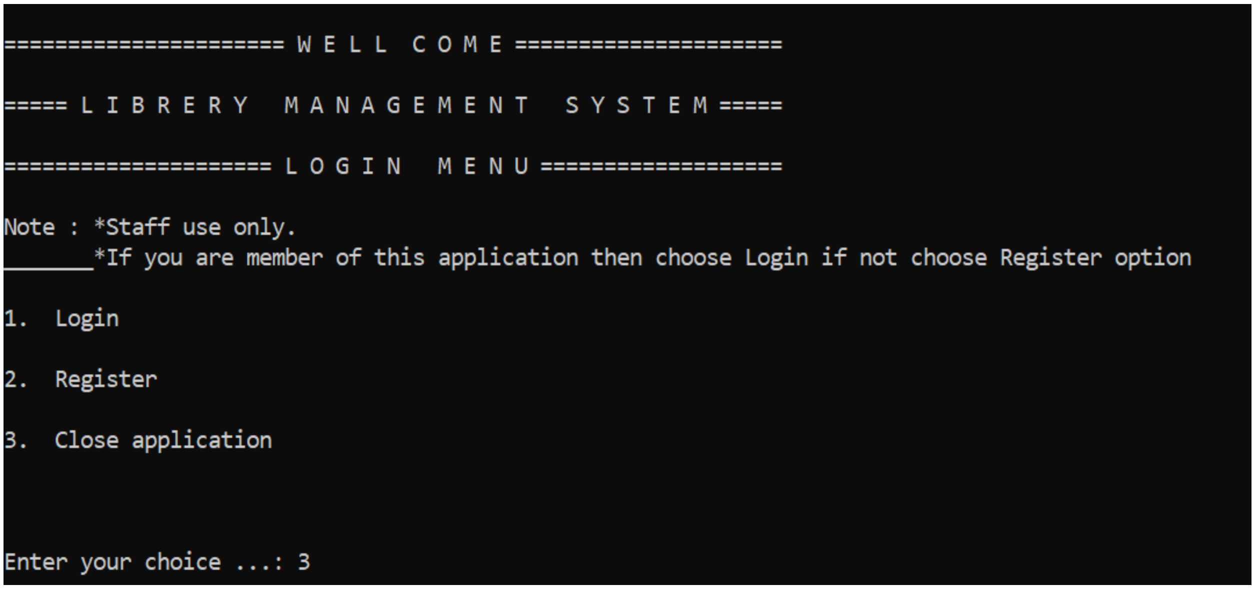
7. PSEUDOCODE

ALGORITHM

```
1. Start

2. Home page

3. Register Menu

        If (login = true)
            Main menu

        Else
            Goto step 3

4. Display menu

5. Exit menu

6. Goto step 3

7. End
```

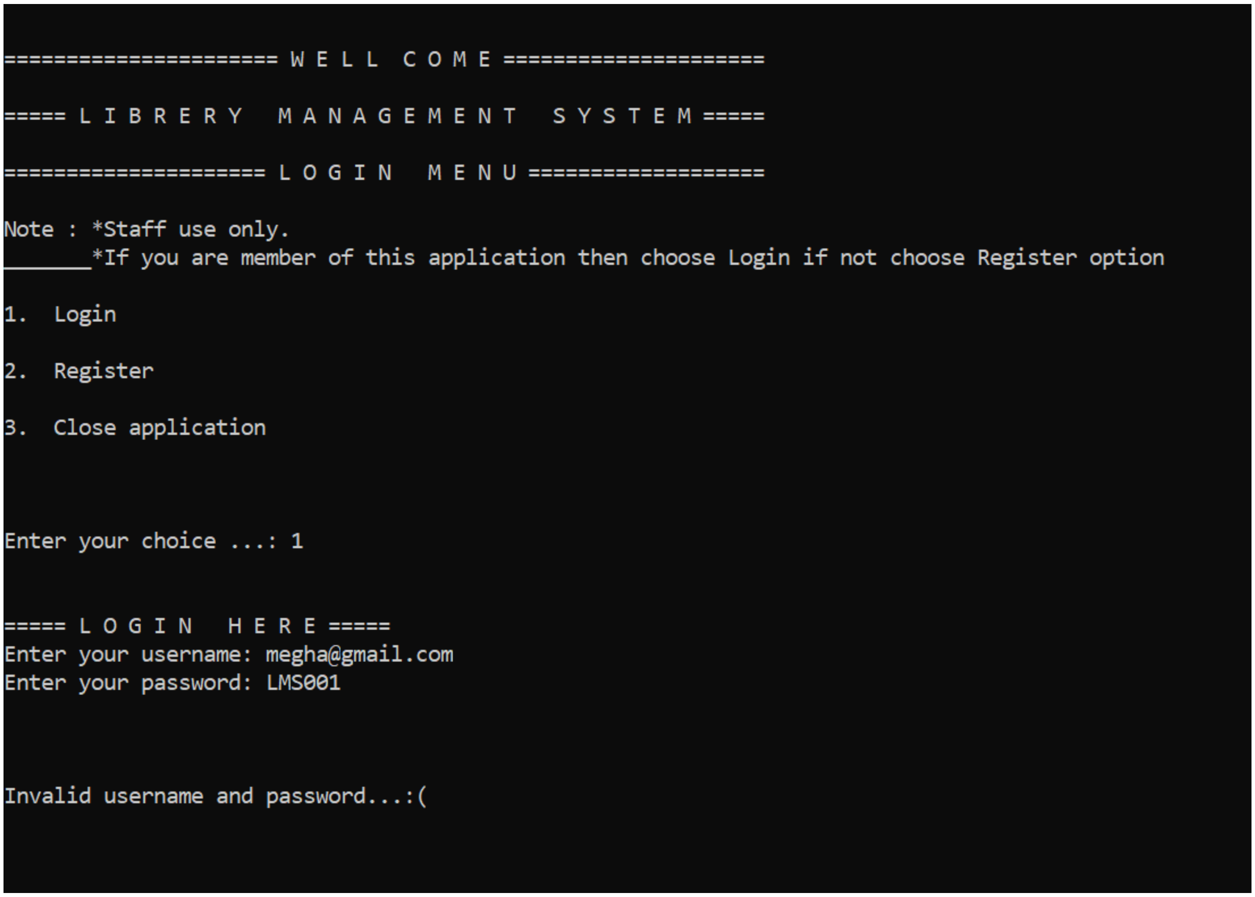
8. SCREENSHOTS

1. REGISTER MENU



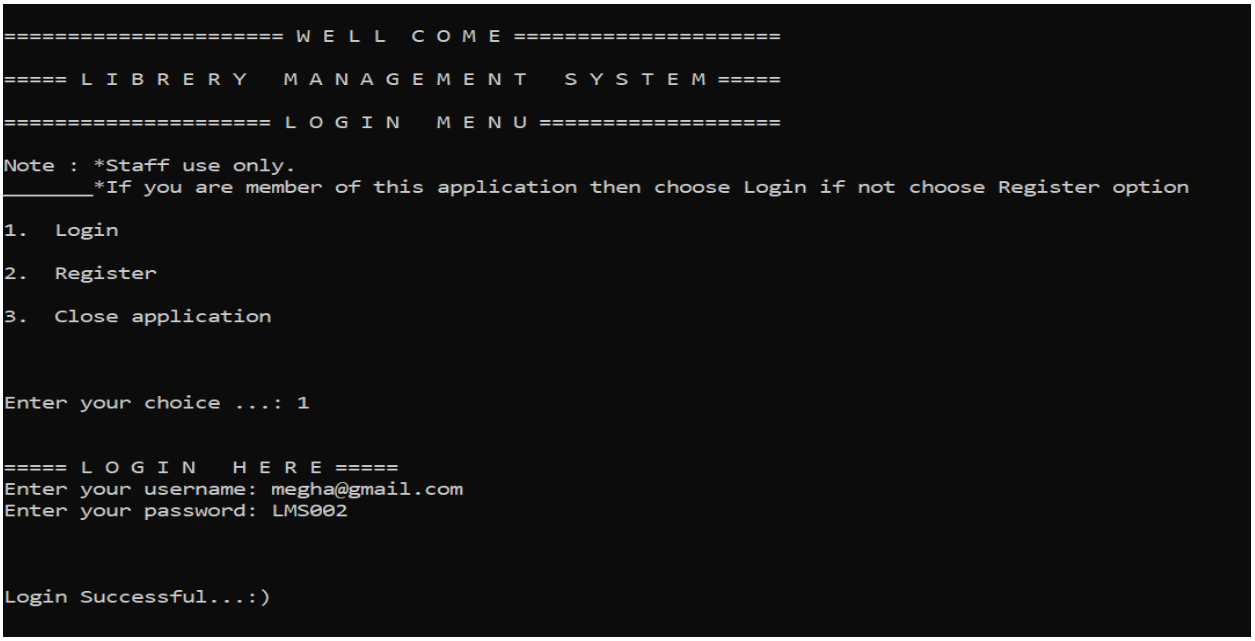
```
====================== W E L L  C O M E =====================

===== L I B R E R Y  M A N A G E M E N T  S Y S T E M =====

===================== L O G I N  M E N U ===================

Note : *Staff use only.
_______*If you are member of this application then choose Login if not choose Register option

1.  Login

2.  Register

3.  Close application


Enter your choice ...: 3
```

REGISTER UNSUCCESSFUL



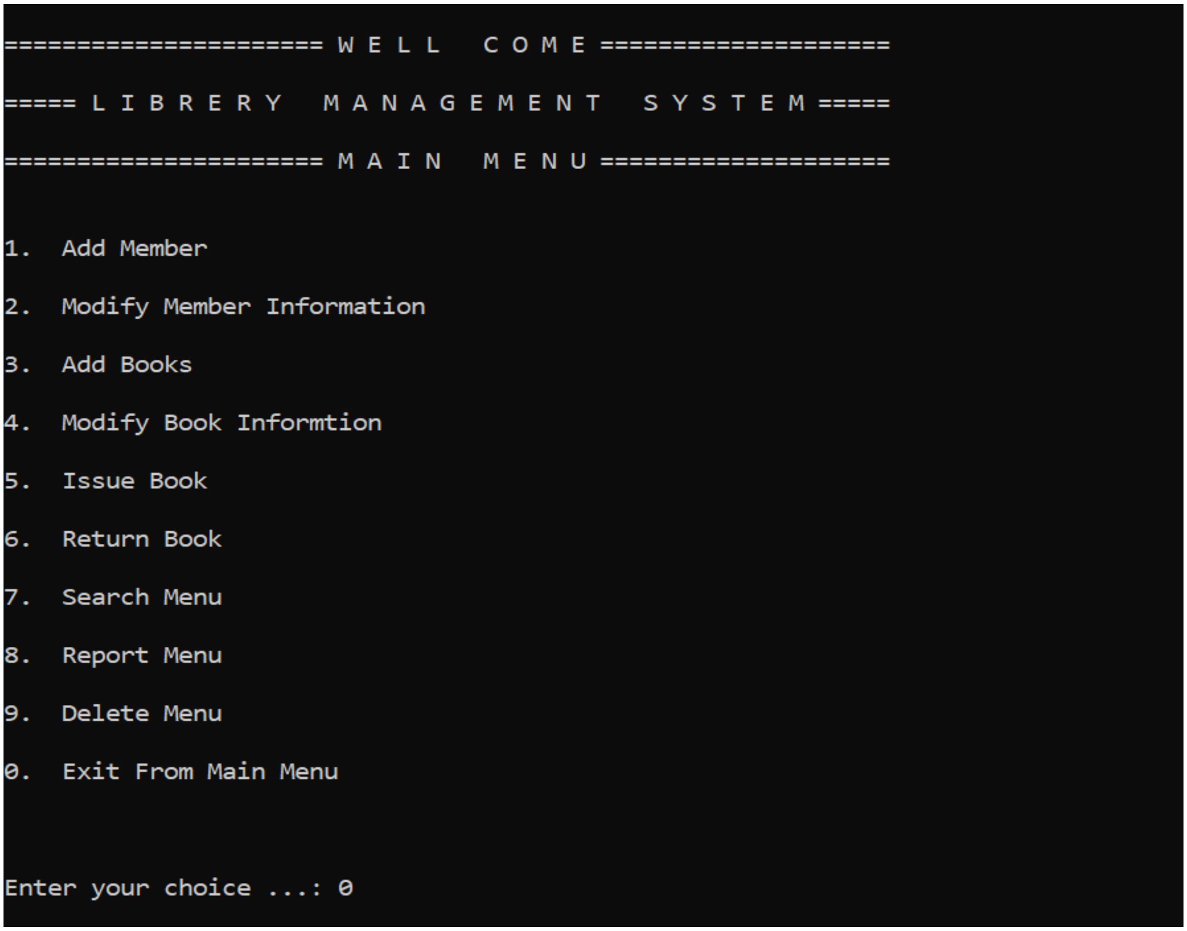
```
====================== W E L L  C O M E =====================

===== L I B R E R Y  M A N A G E M E N T  S Y S T E M =====

===================== L O G I N  M E N U ===================

Note : *Staff use only.
_______*If you are member of this application then choose Login if not choose Register option

1.  Login

2.  Register

3.  Close application


Enter your choice ...: 1


===== L O G I N  H E R E =====
Enter your username: megha@gmail.com
Enter your password: LMS001


Invalid username and password...:(
```

REGISTER SUCCESSFUL



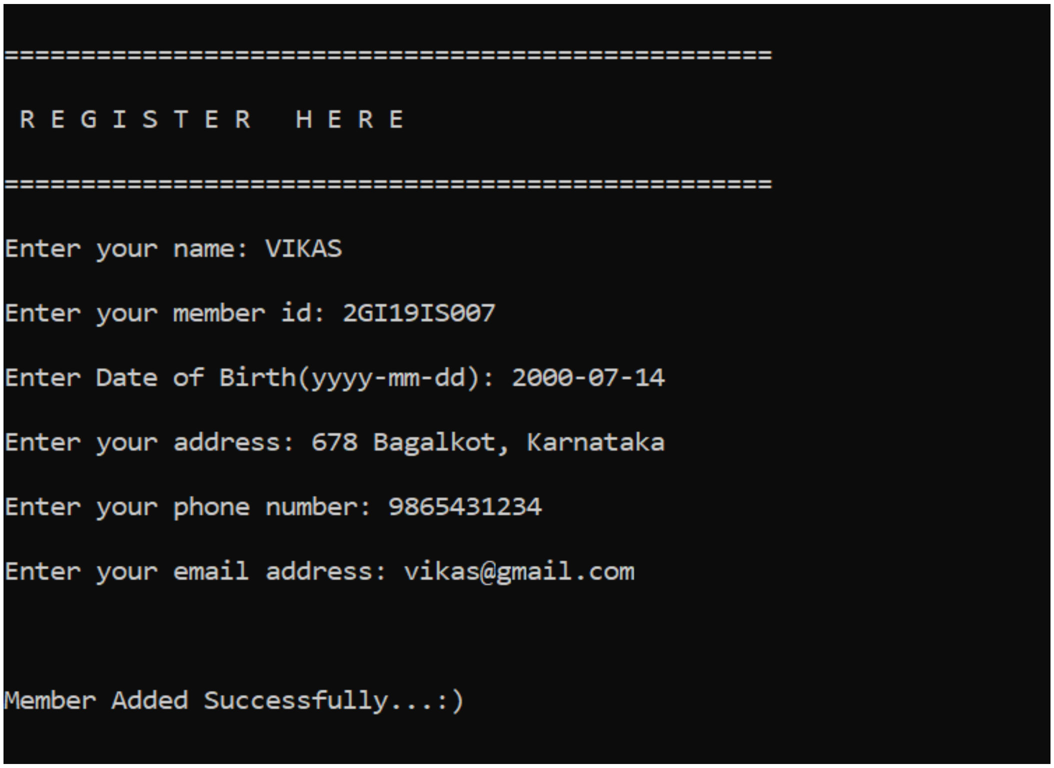
```
====================== W E L L   C O M E =====================

===== L I B R E R Y   M A N A G E M E N T   S Y S T E M =====

===================== L O G I N   M E N U ===================

Note : *Staff use only.
_______*If you are member of this application then choose Login if not choose Register option

1.  Login

2.  Register

3.  Close application


Enter your choice ...: 1


===== L O G I N   H E R E =====
Enter your username: megha@gmail.com
Enter your password: LMS002


Login Successful...:)
```

MAIN MENU



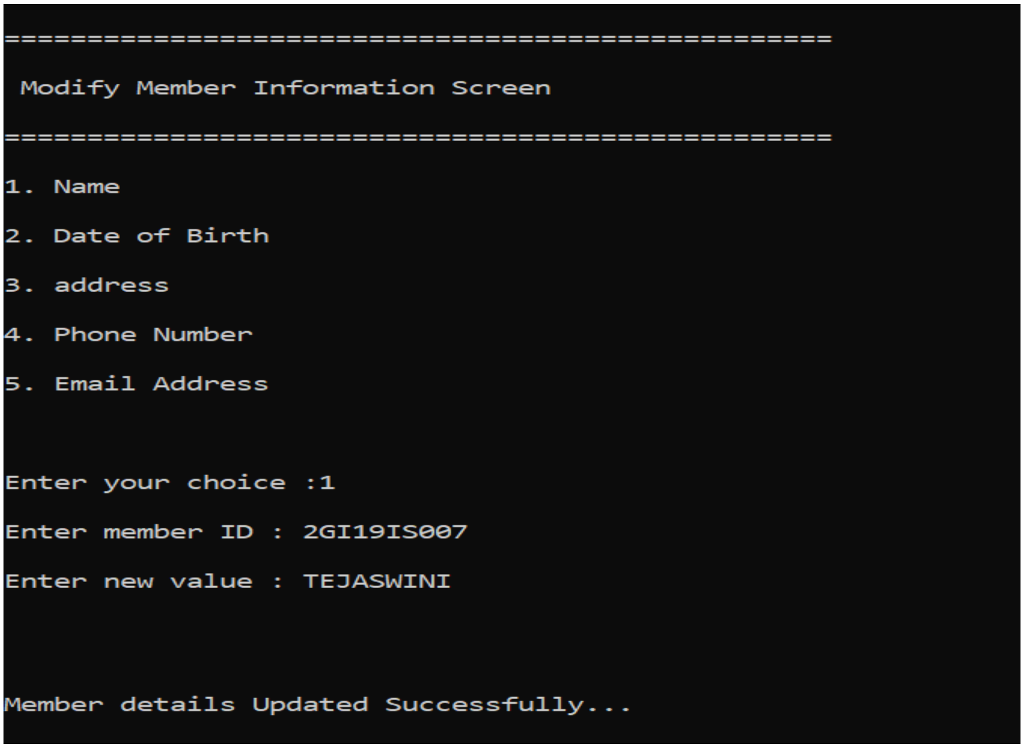
```
====================== W E L L   C O M E ====================

===== L I B R E R Y   M A N A G E M E N T   S Y S T E M =====

====================== M A I N   M E N U ====================


1.  Add Member

2.  Modify Member Information

3.  Add Books

4.  Modify Book Informtion

5.  Issue Book

6.  Return Book

7.  Search Menu

8.  Report Menu

9.  Delete Menu

0.  Exit From Main Menu


Enter your choice ...: 0
```

ADD MEMBER



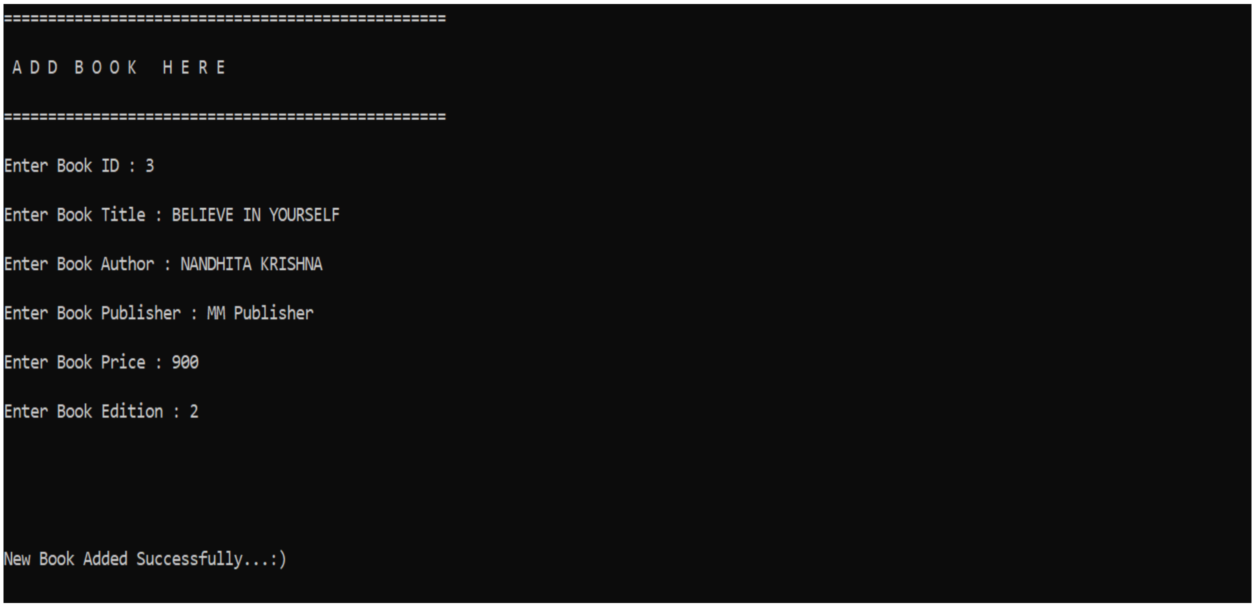
```
==================================================

 R E G I S T E R   H E R E

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

Enter your name: VIKAS

Enter your member id: 2GI19IS007

Enter Date of Birth(yyyy-mm-dd): 2000-07-14

Enter your address: 678 Bagalkot, Karnataka

Enter your phone number: 9865431234

Enter your email address: vikas@gmail.com


Member Added Successfully...:)
```

MODIFY MEMBER



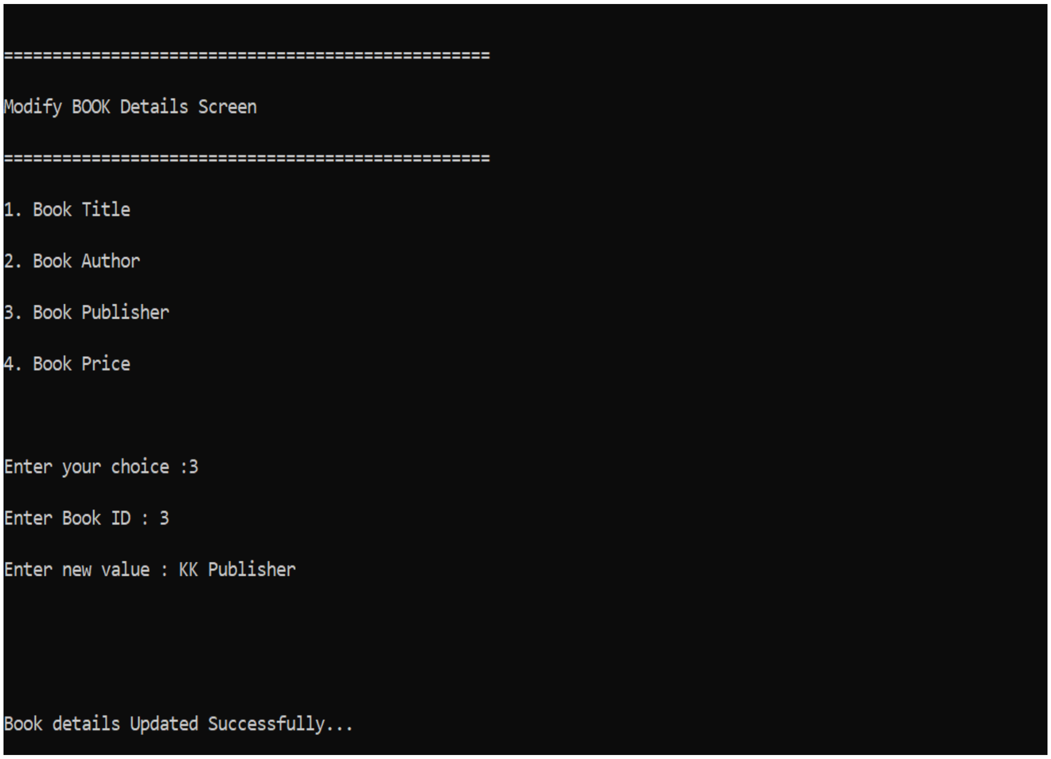
```
==================================================
 Modify Member Information Screen
==================================================
1. Name
2. Date of Birth
3. address
4. Phone Number
5. Email Address

Enter your choice :1
Enter member ID : 2GI19IS007
Enter new value : TEJASWINI


Member details Updated Successfully...
```

ADD BOOK



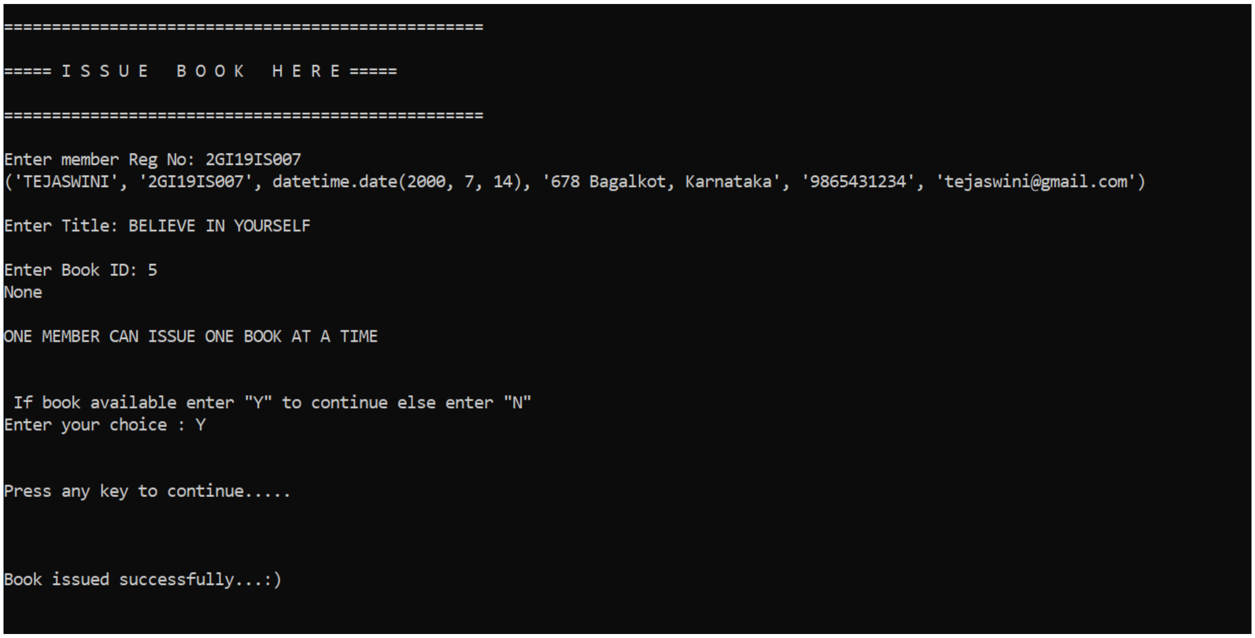
```
==================================================

 A D D  B O O K   H E R E

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

Enter Book ID : 3

Enter Book Title : BELIEVE IN YOURSELF

Enter Book Author : NANDHITA KRISHNA

Enter Book Publisher : MM Publisher

Enter Book Price : 900

Enter Book Edition : 2




New Book Added Successfully...:)
```

MODIFY BOOK INFORMATION



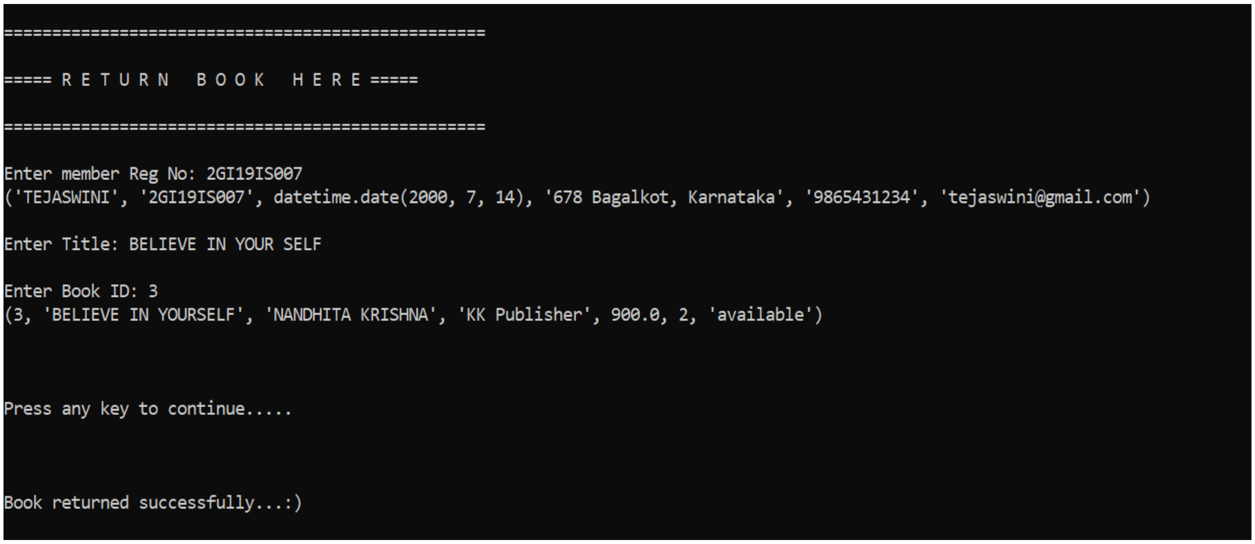
```

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

Modify BOOK Details Screen

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

1. Book Title

2. Book Author

3. Book Publisher

4. Book Price


Enter your choice :3

Enter Book ID : 3

Enter new value : KK Publisher




Book details Updated Successfully...
```

ISSUE BOOK



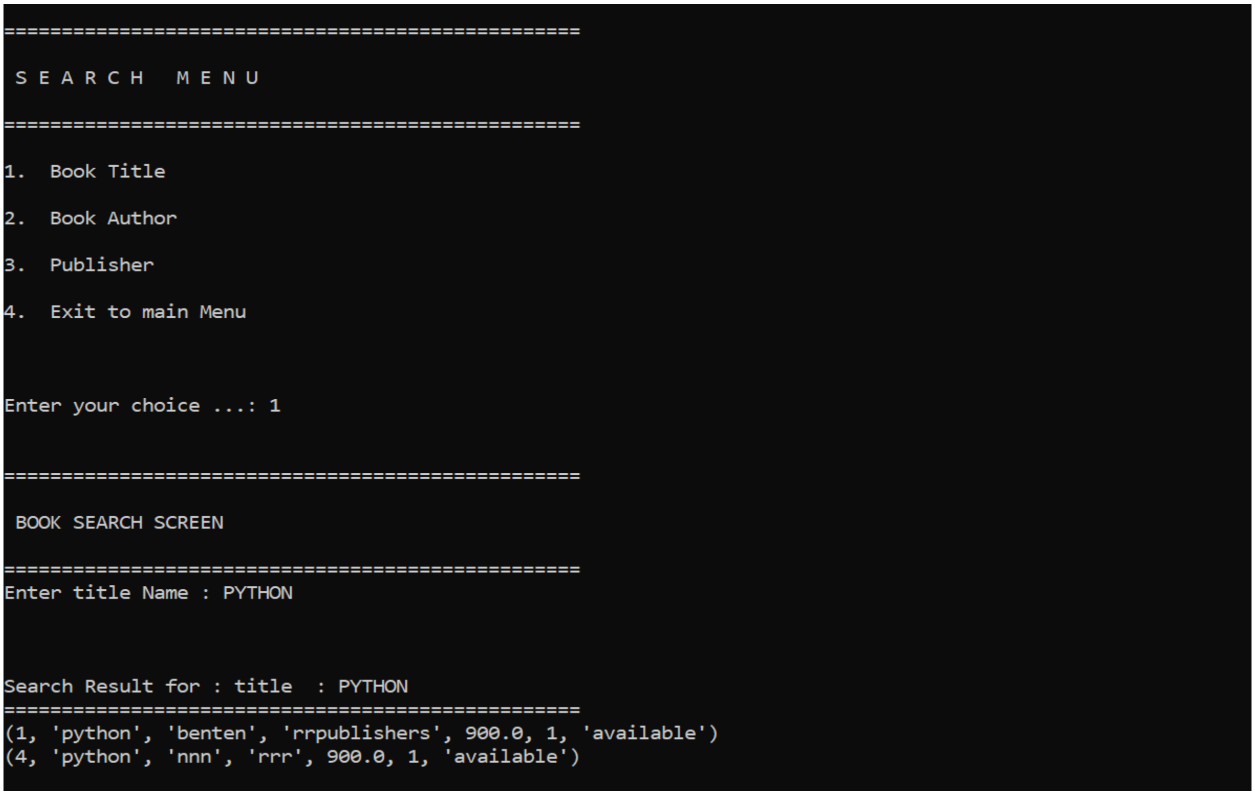
```
==================================================

===== I S S U E   B O O K   H E R E =====

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

Enter member Reg No: 2GI19IS007
('TEJASWINI', '2GI19IS007', datetime.date(2000, 7, 14), '678 Bagalkot, Karnataka', '9865431234', 'tejaswini@gmail.com')

Enter Title: BELIEVE IN YOURSELF

Enter Book ID: 5
None

ONE MEMBER CAN ISSUE ONE BOOK AT A TIME


 If book available enter "Y" to continue else enter "N"
Enter your choice : Y


Press any key to continue.....



Book issued successfully...:)
```

RETURN BOOK



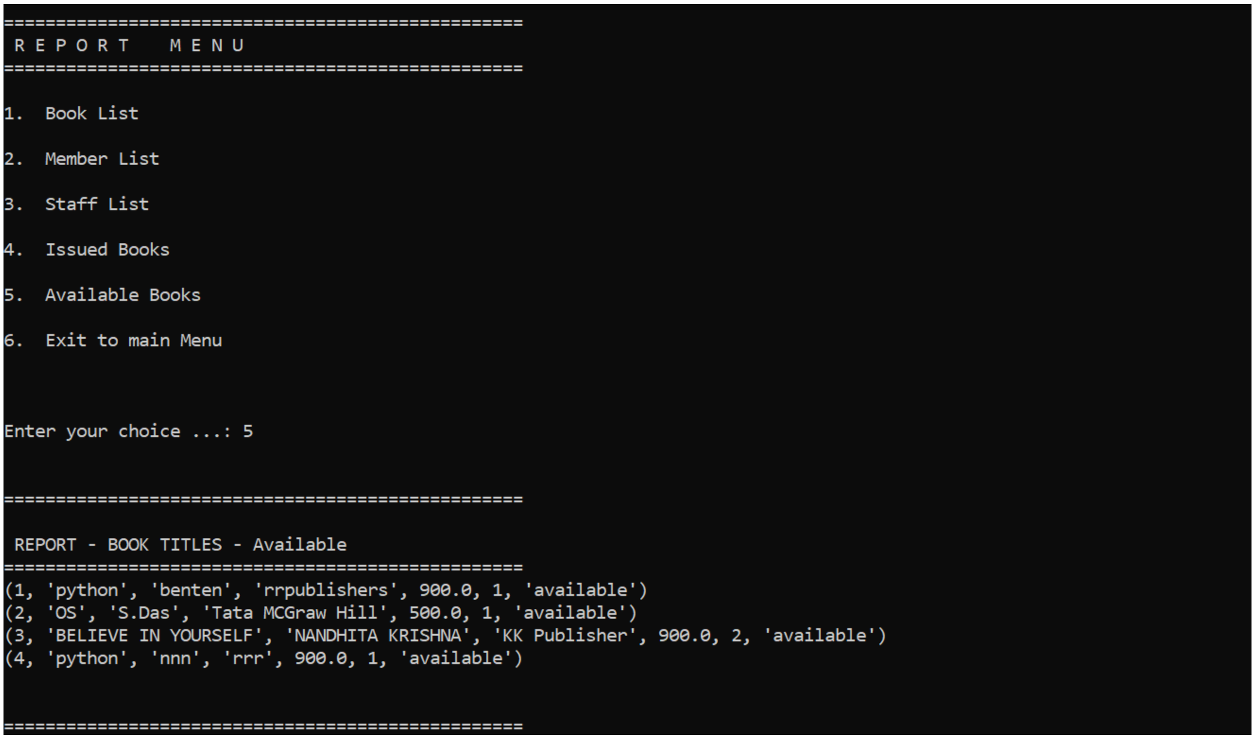
```
==================================================

===== R E T U R N   B O O K   H E R E =====

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

Enter member Reg No: 2GI19IS007
('TEJASWINI', '2GI19IS007', datetime.date(2000, 7, 14), '678 Bagalkot, Karnataka', '9865431234', 'tejaswini@gmail.com')

Enter Title: BELIEVE IN YOUR SELF

Enter Book ID: 3
(3, 'BELIEVE IN YOURSELF', 'NANDHITA KRISHNA', 'KK Publisher', 900.0, 2, 'available')



Press any key to continue.....



Book returned successfully...:)
```

SEARCH MENU



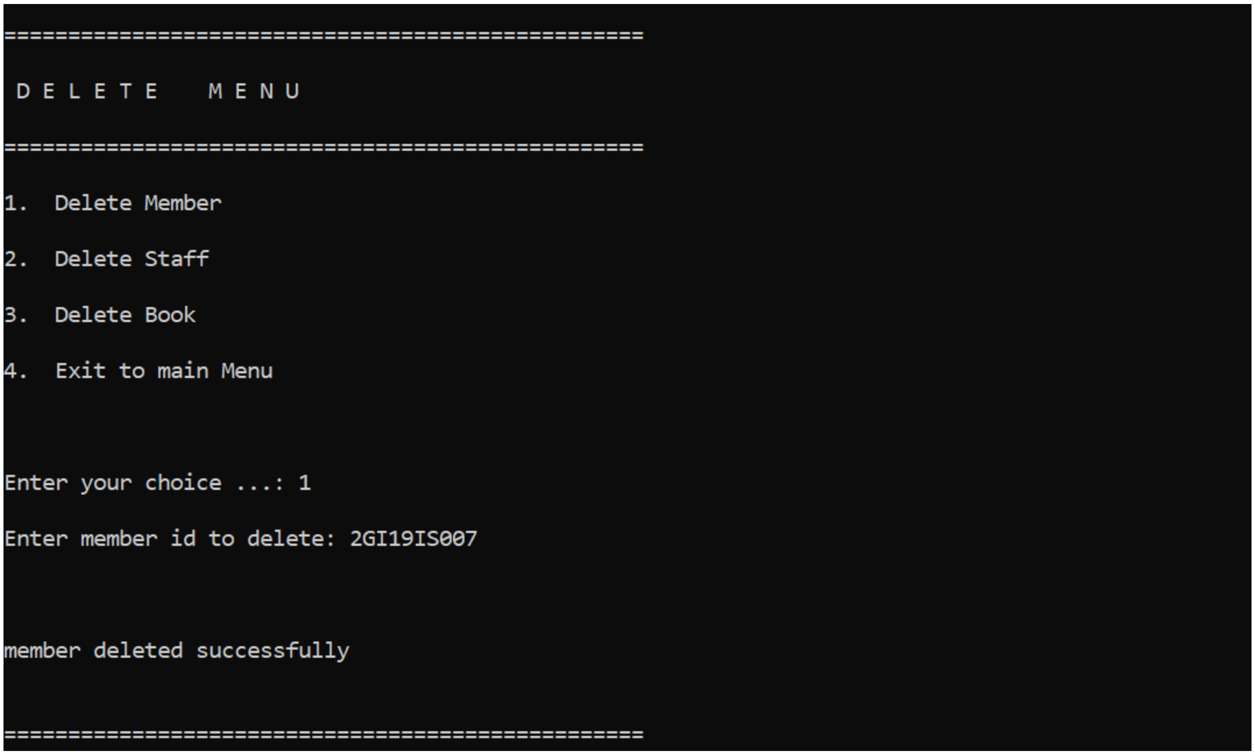
```
==================================================

 S E A R C H   M E N U

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

1.  Book Title

2.  Book Author

3.  Publisher

4.  Exit to main Menu


Enter your choice ...: 1


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

 BOOK SEARCH SCREEN

==================================================
Enter title Name : PYTHON


Search Result for : title  : PYTHON
==================================================
(1, 'python', 'benten', 'rrpublishers', 900.0, 1, 'available')
(4, 'python', 'nnn', 'rrr', 900.0, 1, 'available')
```

REPORT MENU



```
==================================================
 R E P O R T   M E N U
==================================================

1.  Book List

2.  Member List

3.  Staff List

4.  Issued Books

5.  Available Books

6.  Exit to main Menu


Enter your choice ...: 5


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

 REPORT - BOOK TITLES - Available
==================================================
(1, 'python', 'benten', 'rrpublishers', 900.0, 1, 'available')
(2, 'OS', 'S.Das', 'Tata MCGraw Hill', 500.0, 1, 'available')
(3, 'BELIEVE IN YOURSELF', 'NANDHITA KRISHNA', 'KK Publisher', 900.0, 2, 'available')
(4, 'python', 'nnn', 'rrr', 900.0, 1, 'available')


==================================================
```

DELETE MENU



```
==================================================

 D E L E T E   M E N U

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

1.  Delete Member

2.  Delete Staff

3.  Delete Book

4.  Exit to main Menu


Enter your choice ...: 1

Enter member id to delete: 2GI19IS007


member deleted successfully


==================================================
```

9. PYTHON CODE FOR IMPLIMENTATION OF LMS

```
#This LMS project store the data in database, this Project created a MySQL database with name "lms"

#MySql Connector : This module is used to connect mysql database to fetch/add data into database.

#DateTime : To set date and get current date and update the details on the tables.

import mysql.connector as a

from datetime import date

# FIRST PAGE : REGISRE MENU TO ENTER TO THE MAIN MENU

def register_menu():

    while True:

      print('\n====================== W E L L  C O M E ======================')

      print('\n===== L I B R E R Y   M A N A G E M E N T   S Y S T E M =====')

      print('\n===================== L O G I N   M E N U ===================')

      print("\nNote : *Staff use only.\n_______*If you are member of this application then choose Login if not choose Register option")

      print('\n1.  Login')

      print('\n2.  Register')

      print('\n3.  Close application')

      print('\n\n')
```

```
        choice = int(input('Enter your choice ...: '))

        if choice == 1:

            login()

        if choice == 2:

            register()

       if choice == 3:

           break

#STAFF AUTHENTICATION MENU

#TO ADD MEMBER TO TABLE- STAFF :WE CAN ADD STAFF'S BASIC DETAILS

def register():

    #MYDB : FUNCTION TO CONNECT TO THE MYSQL SERVER

    mydb=a.connect(

      host="localhost",

      user="root",

      password="Megha@2001",

      database="lms"

      )

    #MYCURSOR : COMMUNICATE WITH MYSQL DATABASE

    mycursor=mydb.cursor()

    print("\n\n===== R E G I S T E R   H E R E =====")

    print("\nNOTE : Staff use only")

    #EACH NON FORMAT ENTRY MAKE YOU TO CAME OUT OF LOOP AND EXIT FROM SYSTEM

    name=input("Enter your name: ")

    stid=input("Enter your staff id(like-LMSXXX): ")

    dob=input("Enter Date of Birth(yyyy-mm-dd): ")

    address=input("Enter your address: ")

    pno=input("Enter your phone number: ")

    steid=input("Enter your email address: ")

   # sql="create table staff(name varchar(40) not null, stid varchar(6) not null,dob date, address varchar(60), pno varchar(10), steid varchar(40),primary
key(stid));"

    sql="Insert into staff values('{}','{}','{}','{}','{}','{}');".format(name,stid,dob,address,pno,steid)

    #EXICUTE THIS MODULE

    mycursor.execute(sql)

   #CALL METHOD TO SEND COMMIT STATEMENT TO MYSQL SERVER, COMMETTING THE CURRENT TRANSACTION

    mydb.commit()

    print('\n\n')

    print("Member Added Successfully...:)\n\n")

    #CLOSE THE SQL-CONNECTION TO REUSE THE SAME CONNECTION FOR OTHER MODULES
    mydb.close()
# LOGIN : MEMBER ENTERS IF AND ONLY IF HIS/HER USERNAME AND PASSWORD MATCHES IN STAFF TABLE
# USER NAME=EMAIL ADDRESS OF MEMBER, PASSWORD=staff ID
def login():
    mydb=a.connect(
       host="localhost",
       user="root",
       password="Megha@2001",
       database="lms"
       )
    mycursor=mydb.cursor()
    print("\n\n===== L O G I N   H E R E =====")
    username=input("Enter your username: ")
    password=input("Enter your password: ")
    print('\n\n')
    # TO COME OUT OF FOR LOOP
    flag=0
    #FETCH ALL DATA FROM MEMBER TABLE TABLE AND RETURN ROWS AS DICTIONARY
    sql = "select * from staff"
    # MySQLCursorDict creates a cursor that returns rows as dictionaries
    mycursor = mydb.cursor(dictionary=True)
    mycursor.execute(sql)
    # FETCHE AND STORE DATA TO VERIFY IN ENTEIR DATASET
    records = mycursor.fetchall()
    #Fetching each row using column name
    for row in records:
        name = row["name"]
        stid = row["stid"]
        dob = row["dob"]
        address = row["address"]
        pno = row["pno"]
        steid = row["steid"]
        if username == steid and password == stid:
          print('Login Successful...:)\n')
          flag=1
          break
    mydb.close()
    if flag==1:
            main_menu()
        else:
            print("Invalid username and password...:(\n\n")
            #RETURN TO FIRST PAGE
            register_menu()
#START OF MAIN MENU
def main_menu():
    while True:
        print('\n====================== W E L L  C O M E ====================')
        print('\n===== L I B R E R Y  M A N A G E M E N T  S Y S T E M =====')
        print('\n====================== M A I N  M E N U ====================')
        print('\n\n1.  Add Member')
        print('\n2.  Modify Member Information')
        print('\n3.  Add Books')
        print('\n4.  Modify Book Informtion')
        print('\n5.  Issue Book ')
        print('\n6.  Return Book ')
        print('\n7.  Search Menu')
        print('\n8.  Report Menu')
        print('\n9.  Delete Menu')
        print('\n0.  Exit From Main Menu')
        print('\n\n')
        choice = int(input('Enter your choice ...: '))
        if choice == 1:
            add_member()
        if choice == 2:
            modify_member()
        if choice == 3:
            add_book()
        if choice == 4:
            modify_book()
        if choice == 5:
            issue_book()
        if choice == 6:
            return_book()
        if choice == 7:
            search_menu()
        if choice == 8:
            report_menu()
        if choice == 9:
            delete_menu()
        if choice == 0:
            break
def add_member():
    mydb=a.connect(
       host="localhost",
       user="root",
       password="Megha@2001",
       database="lms"
       )
    mycursor=mydb.cursor()
    print('\n')
    print('='*50)
    print("\n R E G I S T E R   H E R E \n")
    print('='*50)
    name=input("\nEnter your name: ")     #50 slots
    mid=input("\nEnter your member id: ")    #10 slots
    dob=input("\nEnter Date of Birth(yyyy-mm-dd): ")
    address=input("\nEnter your address: ")    #80 slots
    pno=input("\nEnter your phone number: ")  #10 slots
    eid=input("\nEnter your email address: ")  #40 slots
   # sql="create table member(name varchar(40) not null, mid varchar(10) not null,dob date, address varchar(60), pno varchar(10), eid varchar(40),primary key(mid));"
    sql="Insert into member values('{}','{}','{}','{}','{}','{}');".format(name,mid,dob,address,pno,eid)
    mycursor.execute(sql)
    mydb.commit()
    print('\n\n')
    print("Member Added Successfully...:)\n\n")
    mydb.close()
#TO ADD BASIC BOOK DETAILS TO TABLE
def add_book():
    mydb=a.connect(
       host="localhost",
       user="root",
       password="Megha@2001",
       database="lms"
```

```
        )
    mycursor=mydb.cursor()
    print('='*50)
    print("\n A D D  B O O K   H E R E \n")
    print('='*50)
    book_id = int(input('\nEnter Book ID : '))
    title = input('\nEnter Book Title : ')
    author = input('\nEnter Book Author : ')
    publisher = input('\nEnter Book Publisher : ')
    price = float(input('\nEnter Book Price : '))
    edition = int(input('\nEnter Book Edition : '))
    #sql="create table book(book_id int,title varchar(40),author varchar(40),publisher varchar(40),price float(6,2),edition int,status char(10),primary key(book_id));"
    sql="Insert into book values({},'{}','{}','{}',{},{},'available')".format(book_id,title,author,publisher,price,edition)
    mycursor.execute(sql)
    mydb.commit()
    print('\n\n')
    print("\n\nNew Book Added Successfully...:)\n\n")
    mydb.close()
#END OF ADD BOOK
#MADIFY MEMEBER INFORMATION
def modify_member():
    mydb=a.connect(
        host="localhost",
        user="root",
        password="Megha@2001",
        database="lms"
        )
    mycursor=mydb.cursor()
    print('\n')
    print('='*50)
    print('\n Modify Member Information Screen \n')
    print('='*50)
    print('\n1. Name')
    print('\n2. Date of Birth')
    print('\n3. address')
    print('\n4. Phone Number')
    print('\n5. Email Address')
    print('\n\n')
    choice = int(input('Enter your choice :'))
    field =''
    if choice == 1:
        field ='name'
    if choice == 2:
        field = 'dob'
    if choice ==3:
        field ='address'
    if choice == 4:
        field = 'pno'
    if choice == 5:
        field = 'eid'
    #GET NEW VALUES TO BE CHANGE FOR PARTICULAR MEMBER
    mem_id = input('\nEnter member ID : ')
    value = input('\nEnter new value : ')
    sql = 'update member set '+ field +' = "'+value+'" where mid = "'+mem_id+'";'
    mycursor.execute(sql)
    mydb.commit()
    print('\n\n')
    print('\nMember details Updated Successfully...\n')
    mydb.close()
# END OF ADD BOOK
# MODIFY BOOK DETAILS
def modify_book():
    mydb=a.connect(
       host="localhost",
       user="root",
       password="Megha@2001",
       database="lms"
       )
    mycursor=mydb.cursor()
    print('\n')
    print('='*50)
    print('\nModify BOOK Details Screen \n')
    print('='*50)
    print('\n1. Book Title')
    print('\n2. Book Author')
    print('\n3. Book Publisher')
    print('\n4. Book Price')
    print('\n\n')
    choice = int(input('Enter your choice :'))
    field = ''  ''
    if choice == 1:
        field = 'title'
    if choice == 2:
        field = 'author'
    if choice == 3:
        field = 'publisher'
    if choice == 4:
        field = 'price'
    bookid = input('\nEnter Book ID : ')
    value = input('\nEnter new value : ')
    sql = 'update book set ' + field + ' = "'+value+'" where book_id = "'+bookid+'";'
    mycursor.execute(sql)
    mydb.commit()
    print('\n\n')
    print('\n\nBook details Updated Successfully...\n')
    mydb.close()
#end of modify_book
#get all details of particular book
def book_status(b_id):
    mydb=a.connect(
       host="localhost",
       user="root",
       password="Megha@2001",
       database="lms"
       )
    mycursor=mydb.cursor()
    sql = 'select * from book where book_id ='+b_id+';'
    mycursor.execute(sql)
    #FETCH SINGLE TUPLE OF DATA OF REQUIRED BOOK ID
    result = mycursor.fetchone()
    return result
#GET ALL DETAILS OF PARTICULAR MEMBER
def mem_status(m_id):
  mydb=a.connect(
    host="localhost",
    user="root",
    password="Megha@2001",
    database="lms"
    )
  mycursor=mydb.cursor()
  sql = 'select * from member where mid ="'+m_id+'";'
  mycursor.execute(sql)
  #FETCH SINGLE TUPLE OF DATA OF REQUIRED MEMBER ID
  result = mycursor.fetchone()
  return result
#ISSUE BOOK TO MEMBER
def issue_book():
  mydb=a.connect(
    host="localhost",
    user="root",
    password="Megha@2001",
    database="lms"
    )
  mycursor=mydb.cursor()
  while True:
    print('\n')
    print('='*50)
    print("\n===== I S S U E   B O O K   H E R E =====\n ")
    print('='*50)
    m_id=input("\nEnter member Reg No: ")
    #CHECK MEMBER DETAILS EXIST IN DATABASE OR NOT
    meminfo=mem_status(m_id)
    print(meminfo)
    title=input("\nEnter Title: ")
    b_id=input("\nEnter Book ID: ")
    #CHECK BOOK DETAILS EXIST IN DATABASE OR NOT
    #CHECK AVAILABILITY OF BOOK
    bookinfo=book_status(b_id)
    print(bookinfo)
   #SET DATE WITH CURRENT DATE
    today = date.today()
```

```
       print('\nONE MEMBER CAN ISSUE ONE BOOK AT A TIME\n')
       print('\n If book available enter "Y" to continue else enter "N"')
       ch=input('Enter your choice : ')
       if ch == "N":
          break
       wait = input('\n\nPress any key to continue.....')
#sql="create table transaction(tid int NOT NULL AUTO_INCREMENT, title varchar(40) not null, m_id varchar(10) not null,b_id int, doi date,primary key(tid));"
       sql="Insert into transaction(title,m_id,b_id,doi) values('"+title+"','"+m_id+"',"+b_id+",'"+str(today)+"');"
       mycursor.execute(sql)
    #CHANGE STATUS OF BOOK
       sqlup="update book set status='issued' where book_id="+b_id+";"
       mycursor.execute(sqlup)
       mydb.commit()
       print("\n\n")
       print("Book issued successfully...:)\n\n")
       break
       mydb.close()
#SUBMIT THE BOOK RETURNED BY MEMBER
def return_book():
    mydb=a.connect(
       host="localhost",
       user="root",
       password="Megha@2001",
       database="lms"
       )
    mycursor=mydb.cursor()
    while True:
        print('\n')
        print('='*50)
        print("\n===== R E T U R N   B O O K   H E R E =====\n")
        print('='*50)
        m_id=input("\nEnter member Reg No: ")
        meminfo=mem_status(m_id)
        print(meminfo)
        title=input("\nEnter Title: ")
        b_id=input("\nEnter Book ID: ")
        bookinfo=book_status(b_id)
        print(bookinfo)
        today = date.today()

        wait = input('\n\n\nPress any key to continue.....')

        sql="update transaction set dor='"+str(today)+"' where b_id="+b_id+";"

        mycursor.execute(sql)

        #UPDATE STATUS OF BOOK

        sqlup="update book set status='available' where book_id="+b_id+";"

        mycursor.execute(sqlup)

        mydb.commit()

        print("\n\n")

        print("Book returned successfully...:)\n\n")

        break

        mydb.close()

#MENU TO SEARCH DESIRED BOOK

def search_menu():

    while True:

     print('\n')

     print('='*50)

     print('\n S E A R C H   M E N U \n')

     print('='*50)

     print("\n1.  Book Title")

     print('\n2.  Book Author')

     print('\n3.  Publisher')

     print('\n4.  Exit to main Menu')

     print('\n\n')

     choice = int(input('Enter your choice ...: '))

     field =''

       if choice == 1:

        field='title'

       if choice == 2:

        field = 'author'

       if choice == 3:

        field = 'publisher'

       if choice == 4:

        break

       search_book(field)

#RETRIVE SEARCHED BOOK DETAILS

def search_book(field):

    mydb=a.connect(
       host="localhost",
       user="root",
       password="Megha@2001",
       database="lms"
       )
    mycursor=mydb.cursor()
    print('\n')
    print('='*50)
    print('\n BOOK SEARCH SCREEN \n')
    print('='*50)
    msg ='Enter '+ field +' Name : '
    title = input(msg)
    sql ='select * from book where '+ field + ' like "%'+ title+'%";'
    mycursor.execute(sql)
    records = mycursor.fetchall()
    print('\n\n')
    print('Search Result for :',field,' :' ,title)
    print('='*50)
    for record in records:
      print(record)
    mydb.close()
#REPORT MENU TO RETRIVE BRIEF member and book details
def report_menu():
    while True:
      print('\n')
      print('='*50)
      print(' R E P O R T    M E N U ')
      print('='*50)
      print("\n1.  Book List")
      print('\n2.  Member List')
      print('\n3.  Staff List')
      print('\n4.  Issued Books')
      print('\n5.  Available Books')
      print('\n6.  Exit to main Menu')
      print('\n\n')
      choice = int(input('Enter your choice ...: '))
      if choice == 1:
        reprot_book_list()
      if choice == 2:
        report_member_list()
      if choice == 3:
        reprot_staff_list()
      if choice == 4:
        report_issued_books()
      if choice == 5:
        report_available_books()
      if choice == 6:
        break
#RETRIVE ALL BOOKS DETAILS
def reprot_book_list():
    mydb=a.connect(
       host="localhost",
       user="root",
       password="Megha@2001",
       database="lms"
       )
    mycursor=mydb.cursor()
    print('\n')
    print('='*50)
    print('\n REPORT - BOOK TITLES ')
    print('='*50)
    sql ='select * from book;'
    mycursor.execute(sql)
    records = mycursor.fetchall()
    for record in records:
       print(record)
    mydb.close()
def reprot_staff_list():
    mydb=a.connect(
       host="localhost",
       user="root",
       password="Megha@2001",
       database="lms"
       )
    mycursor=mydb.cursor()
    print('\n')
    print('='*50)
    print('\n REPORT - staff list ')
    print('='*50)
    sql ='select * from staff;'
    mycursor.execute(sql)
    records = mycursor.fetchall()
    for record in records:
        print(record)
    mydb.close()
#RETRIVE ALL MEMBER DETAILS
def report_member_list():
    mydb=a.connect(
        host="localhost",
        user="root",
        password="Megha@2001",
        database="lms"
        )
    mycursor=mydb.cursor()
    print('\n')
    print('='*50)
    print('\n REPORT - Members List ')
    print('='*50)
    sql = 'select * from member;'
    mycursor.execute(sql)
    records = mycursor.fetchall()
    for record in records:
        print(record)
    mydb.close()
#RETRIVE ALL ISSUED BOOKS
def report_issued_books():
    mydb=a.connect(
        host="localhost",
        user="root",
        password="Megha@2001",
        database="lms"
        )
    mycursor=mydb.cursor()
    print('\n')
```

```
  print('='*50)
  print('\n REPORT - BOOK TITLES - Issued')
  print('='*50)
  sql = 'select * from book where status = "issued";'
  mycursor.execute(sql)
  records = mycursor.fetchall()
  for record in records:
    print(record)
  mydb.close()
#RETRIVE ALL AVAILABLE BOOK
def report_available_books():
  mydb=a.connect(
    host="localhost",
    user="root",
    password="Megha@2001",
    database="lms"
    )
  mycursor=mydb.cursor()
  print('\n')
  print('='*50)
  print('\n REPORT - BOOK TITLES - Available')
  print('='*50)
  sql = 'select * from book where status = "available";'
  mycursor.execute(sql)
  records = mycursor.fetchall()
  for record in records:
    print(record)
  mydb.close()
def delete_menu():
  while True:
    print('\n')
    print('='*50)
    print('\n D E L E T E    M E N U \n')
    print('='*50)
    print("\n1.  Delete Member")
    print('\n2.  Delete Staff')
    print('\n3.  Delete Book')
    print('\n4.  Exit to main Menu')
```

```
        print('\n\n')
        choice = int(input('Enter your choice ...: '))
        if choice == 1:
            del_mem()
        if choice == 2:
            del_staff()
        if choice == 3:
            del_book()
        if choice == 4:
          break
#DELETE MEMBER FROM MEMBER TABLE
def del_mem():
    mydb=a.connect(
       host="localhost",
       user="root",
       password="Megha@2001",
       database="lms"
       )
    mycursor=mydb.cursor()
    mem_id=input("\nEnter member id to delete: ")
    sqlt = "delete from transaction where m_id ='"+mem_id+"';"
    mycursor.execute(sqlt)
    sql = "delete from member where mid ='"+mem_id+"';"
    mycursor.execute(sql)
    mydb.commit()
    print('\n\n')
    print("member deleted successfully")
    mydb.close()
#DELETE STAFF DETAILS FROM STAFF
def del_staff():
    mydb=a.connect(
       host="localhost",
       user="root",
       password="Megha@2001",
       database="lms"
       )
    mycursor=mydb.cursor()
    stf_id=input("\nEnter member id to delete: ")
    sql = "delete from staff where stid = '"+stf_id+"';"
    mycursor.execute(sql)
    mydb.commit()
    print('\n\n')
    print("member deleted successfully")
    mydb.close()
#DELETE BOOK FROM MEMBER TABLE
def del_book():
    mydb=a.connect(
      host="localhost",
      user="root",
      password="Megha@2001",
      database="lms"
      )
    mycursor=mydb.cursor()
    bk_id=input("\nEnter book id to delete: ")
#DELETE FROM REFERENCED TABLE
    sqlt = "delete from transaction where b_id ='"+bk_id+"';"
    mycursor.execute(sqlt)
#DELETE FROM MAIN TABLE
    sql = "delete from book where book_id ='"+bk_id+"';"
    mycursor.execute(sql)
    mydb.commit()
    print('\n\n')
    print("book deleted successfully...")
    mydb.close()
#END OF MAIN MENU AND RELATED MODULES
#PATH TO START
if __name__ == "__main__":
     register_menu()
```

10. CONCLUSION

This website provides a computerized version of library management system which will benefit the students as well as the staff of the library.

It makes entire process online where student can search books, staff can generate reports and do book transactions. It also has a facility for staff login where staff can login and can see status of books issued as well request for book or give some suggestions.

The Library management System allows the user to store the book details and the member's details. This system allows storing the details of all the data related to library. The implementation of the system will reduce data entry time and provide readily calculated reports.

11. REFERENCE

- **https://www.w3schools.com/python/python_mysql_getstarted.asp**
- **https://www.academia.edu/37726542/Library_Management_System_Mini_Project_Report_On_LIBRARY_MANAGEMENT_SYSTEM**
- **https://rrtutors.com**