



Delhi Public School

Udaipur

PROJECT (SYNOPSIS+CODE)

LIBRARY MANAGEMENT

SESSION: 2021-22

SUBJECT: COMPUTER SCIENCE

CLASS: XII-B

SUBMITTED TO:

VINAY K. TIWARI

CREATED BY:

SHAILJA SHAKTAWAT

CERTIFICATE

This is to certify that this synopsis of the project entitled Library Management Project based on Python and MySQL connectivity has been prepared by SHAILJA SHAKTAWAT of class XII from DPS, Udaipur.

She has put her sincere efforts into the collection of data and performance of experiments for the project successfully under my supervision

**SANJAY NARVARIA
PRINCIPAL**

**VINAY KRISHAN TIWARI
PGT(CS)**

ACKNOWLEDGEMENT

I feel immense pleasure and deep feeling gratitude towards Mr Vinay K Tiwari of Delhi Public School, Udaipur for his skilful guidance, constructive and valuable suggestions with encouraging co-operation for my project, which not merely helped but enable me to give effort towards the synopsis of the project. I am also thankful to Mr Sanjay Narvaria, Principal Delhi Public School, Udaipur for his encouragement and valuable suggestions given time to time. I also extend my thanks to all colleagues and friends for their valuable assistance and kind co-operation during the course of this investigation.

SHAILJA SHAKTAWAT

XII-B

DPS UDAIPUR



INDEX

SYNOPSIS

PROTOTYPE

MYSQL DATABASE

CODE

OUTPUT

SCOPE & IMPROVEMENT

NOTES

BIBLIOGRAPHY

SYNOPSIS

The present synopsis is for a library management system. The present software functions in a way that it allows to manage and run an E-library.

LOGIN & REGISTRATION

- This would be linked to the username and password database containing a few entries.
- And obviously registration is available too.

ADMIN RIGHTS

- Admin login is unique
- Book entries.
- Managing member data

FOR MEMBERS

- Explore the books.
- Issue/Return
- Pretty straightforwardly check the status of a book availability,etc.

FOR BOOKS

- Saved in a separate section, available in various genres.
- All paperback
- Pretty convenient.

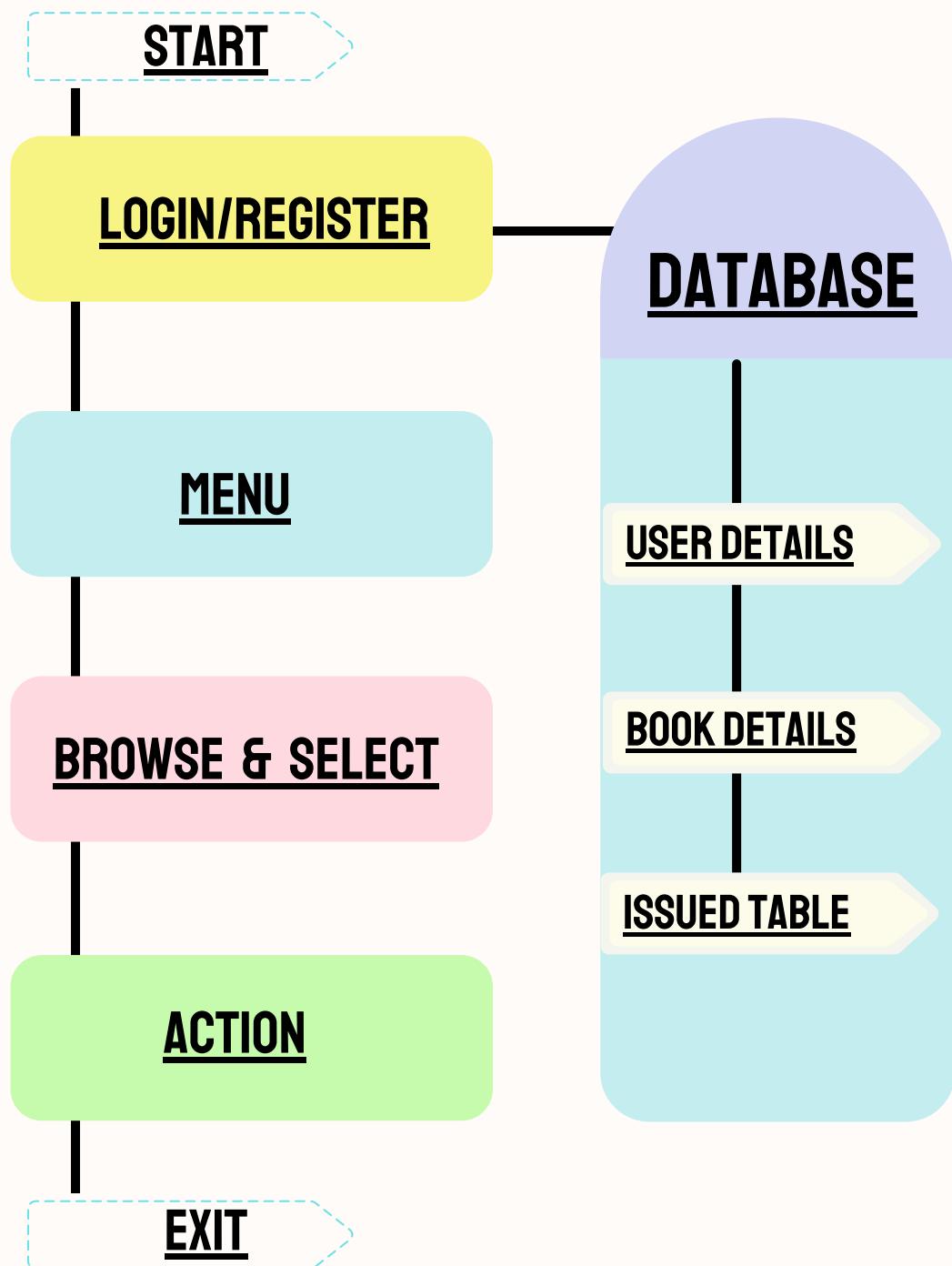
THE ENDS

- Front end: Python
- Back end: MySQL
- Both ends connect together for proper functionality of the application.

BYE BYE!

- Just logging off after finishing one's business.
- In fact, there is exit option available at every major menu

PROTOTYPE



MYSQL DATABASE

Database: library

Tables:

- book_details
- member_details
- issued

acc_no	book_name	author_name	genre	availability	language
1000	Harry Potter and the Philosophers Stone	J.K. Rowling	Fantasy	Available	English
1001	Harry Potter and the Chamber of Secrets	J.K. Rowling	Fantasy	Available	English
1002	Harry Potter and the Prisoner of Azkaban	J.K. Rowling	Fantasy	Available	English
1003	Harry Potter and the Goblet of Fire	J.K. Rowling	Fantasy	Available	English
1004	Harry Potter and the Order of the Phoenix	J.K. Rowling	Fantasy	Available	English
1005	Harry Potter and the Half-Blood Prince	J.K. Rowling	Fantasy	Available	English
1006	Harry Potter and the Deathly Hallows	J.K. Rowling	Fantasy	Available	English
1007	The Tales of Beedle the Bard	J.K. Rowling	Fantasy	Available	English
1008	Quidditch Through the Ages	J.K. Rowling	Fantasy	Available	English
1009	The Ickabog	J.K. Rowling	Fantasy	Available	English
2000	Chronicles of Narnia : The Magicians Nephew	C.S. Lewis	Fantasy	Available	English
2001	Chronicles of Narnia : The Lion, The Witch & The Wardrobe	C.S. Lewis	Fantasy	Available	English
2002	Chronicles of Narnia : The Horse and His Boy	C.S. Lewis	Fantasy	Available	English
2003	Chronicles of Narnia : Prince Caspian	C.S. Lewis	Fantasy	Available	English
2004	Chronicles of Narnia : The Voyage of the Dawn Treader	C.S. Lewis	Fantasy	Available	English
2005	Chronicles of Narnia : The Silver Chair	C.S. Lewis	Fantasy	Available	English
2006	Chronicles of Narnia : The Last Battle	C.S. Lewis	Fantasy	Available	English
2007	The Secret	Rhonda Byrne	Self-Help	Available	English
2008	Sommnath	Acharya Chatursen	Historical Fiction	Available	Hindi
2009	The Immortals of Meluha	Amish Tripathi	Historical Fiction	Available	English
3000	The Secret of the Nagas	Amish Tripathi	Historical Fiction	Available	English
3001	The Oath of the Vayuputras	Amish Tripathi	Historical Fiction	Available	English
3002	Turbulence: A True Story of Survival	Annette Herfkens	Autobiography	Available	English
3003	The Power of Habit	Charles Duhigg	Self-Help	Available	English
3004	Type Talk	Otto Kroeger & Janet M. Thuesen	Psychology	Available	English
3005	Linda Goodmans Sun signs	Linda Goodman	Astrology	Available	English
3006	Murder on the Orient Express	Agatha Christie	Detective fiction	Available	English
3007	The A.B.C. Murders	Agatha Christie	Detective fiction	Available	English
3008	And Then There Were None	Agatha Christie	Detective fiction	Available	English
3009	The Diary of a Young Girl	Anne Frank	Autobiography	Available	English
4000	Indian Folk Tales	Ruskin Bond	Children	Available	English
4001	The Canterville Ghost	Oscar Wilde	Humour	Available	English
4002	Charlie and the Chocolate Factory	Roald Dahl	Children	Available	English
4003	The Picture Of Dorian Gray	Oscar Wilde	Philosophy	Available	English
4004	Stories by Munshi Prem Chand	Munshi Prem Chand	Hindi Literature	Available	Hindi
4005	Stories by Rabindranath Tagore	Rabindranath Tagore	Hindi Literature	Available	Hindi
4006	Stories by Manu Bhandari	Manu Bhandari	Hindi Literature	Available	Hindi
4007	Stories by Mahadevi Verma	Mahadevi Verma	Hindi Literature	Available	Hindi
4008	Twilight	Stephanie Meyer	Young Adult Fantasy	Available	English
4009	New Moon	Stephanie Meyer	Young Adult Fantasy	Available	English
5000	Eclipse	Stephanie Meyer	Young Adult Fantasy	Available	English

Table:book_details

member_id	first_name	last_name	address_	dob	phone_no	pass_key	email_id
11	Mark	Lee	1,Inception Street,Neo City	1999-04-02	1234567	onyourmark	mork@prettygrass.com
22	Mafalda	Hopkirk	2.The Seed, Neo City	1945-07-20	2345678	muffinhop	icerystal@mom.com
33	Johnny	Suh	3,Sputnick Road, Neo City	1990-09-09	3456789	jcc	notmyproblem@jfe.com
44	Katsuki	Bakugo	4,Godrik Hollow, Ncity	2000-05-01	4567890	lordexplosionmurder	starrypuma@nha.com
55	Rowena	Ravenclaw	5,Snorkacks Hill, Laputa	1999-08-18	1023456	phoenix	dareyou@trickme.com
66	Yangyang	Liu	6,Elysian Place, Laputa	2000-11-03	2034567	bella	myson11@lowlow.com
77	Luna	Lovegood	7,Mystic Falls, Ncity	2004-12-25	3045678	heliospaths	turbulent@paradise.com
88	Shoto	Todoroki	8,The Waterfall, Vercity	2000-06-22	4056789	coldsoba	fireice@robertfrost.com
99	Izuku	Midoriya	9,U.A. School	2000-02-22	7890123	allmightyjr	green@strawberry.com
100	Gojo	Satauro	17,Ikigai Lake,Laputa	1999-09-19	2819891820	baka	sensei@thecoolest.com
111	Itadori	Yuji	9,Jujutsu High,Tokyo	2004-08-25	3981930930	enfp	flabbergasted@me.com
127	Lune	Fleur	Turnip Turn, Lake Yamanoko	2003-06-24	1271271270	Czennie	felicity@feliznavidad.com

Table:member_details

CODE

```
import mysql.connector as mysql
from mysql.connector import Error
import pandas as pd

mycon = mysql.connect(
    host='localhost',
    user='root',
    password='password',
    database='library')

if mycon.is_connected():
    print('MySQL connection successfully established.')

cur = mycon.cursor()

#CREATING TABLES IN THE DATABASE

create_member_table = """
CREATE TABLE member_details (
    member_id INT PRIMARY KEY,
    first_name VARCHAR(40) NOT NULL,
    last_name VARCHAR(40) NOT NULL,
    address_ VARCHAR(40) NOT NULL,
    dob DATE,
    phone_no VARCHAR(20),
    pass_key VARCHAR(40)
);
"""

cur.execute(create_member_table)

create_book_table = """
CREATE TABLE book_details (
    acc_no INT PRIMARY KEY,
    book_name VARCHAR(40) NOT NULL,
    author_name VARCHAR(40) NOT NULL,
    genre VARCHAR(40),
    type VARCHAR(40),
    availability VARCHAR(40),
    language VARCHAR(40)
);
"""

cur.execute(create_member_table)
```

```

#Functions in sequence:
##
1) exit_func
2) check_id
3) register
4) issue_book
5) return_book
6) book_exc
7) member_exc
8) menu3-For Admin
9) menu2-issue,return,exit
10) login
11) startmenu1-login,register,exit
##
#To exit the running application anytime.

def exit_func():
    print('Thank you for paying a visit to us! Have a fulfilling day! :)')
    wait = input('\n\n\n Press any key to exit...')

#To check if the inputed id already exists.

def check_id():
    correct=True
    while correct:
        m_id = int(input('Enter a chosen id number:'))

        cur.execute('SELECT member_id FROM member_details WHERE member_id = %s;%(m_id)')
        x=cur.fetchall()
        if x==[]:
            correct=False
            break
        y=x.pop(0)
        z= y[0]

        if z == m_id:
            print('The ID already exists, try again')
            pass

#Registration
def register():
    print('So you wanna become a member of our low profile hi-tech library? \nSure. \nHere you go.')
    check_id()
    mem_id=input('Confirm your id: ')
    f_name = input('Enter first name: ')
    l_name = input('Enter last name: ')
    add = input('Enter address: ')
    dob = input('Date of Birth (YYYY-MM-DD): ')
    phone = input('Contact number: ')
    password=input('Set password: ')
    email=input('Enter your email id:')

    cur.execute('INSERT INTO
    member_details(member_id,first_name,last_name,address_,dob,phone_no,pass_key,email_id) VALUES(?,?,?,?,?,?,?,?)',
    (mem_id,f_name,l_name,add,dob,phone,password,email))
    mycon.commit()

    print('Welcome to the club',f_name,'!')
    wait = input('\n\n\n Press any key to continue....')

```

```

#To issue books by members

def issue_book():
    cans="y"
    while (cans=="y"):
        print('You can choose any genre from the following: ')
        genre_list=pd.Series(["Fantasy",'Self-Help','Historical Fiction', 'Autobiography','Psychology','Astrology','Detective Fiction','Children','Humour','Philosophy','Hindi Literature','Young Adult Fantasy'])
        print(genre_list)
        genre=input('Select a genre name\IN CASE SENSITIVE: ')
        cur.execute('SELECT*FROM book_details WHERE genre = "%s";%(genre)')

        o=cur.fetchall()

        p=pd.DataFrame(o)
        print(p)

        book_choice = int(input('Enter accession no. of the book you would like to issue: '))

        cur.execute('SELECT * FROM book_details WHERE availability = "Available" AND acc_no = %s;%(book_choice)')
        u = cur.fetchall()
        #type conversion
        l=u.pop(0)
        b_name=l[1]
        a_name=l[2]
        m=pd.Series(l)

        print('Book details-----')
        print(m)

        cur.execute('SELECT phone_no FROM member_details WHERE member_id = %s;%(mid)')
        so=cur.fetchall()
        #type conversion
        y=so.pop(0)
        z= y[0]
        pno=int(z)

        cur.execute('UPDATE book_details SET availability = "%s" WHERE acc_no = %s.format("Unavailable",book_choice)')
        mycon.commit()

        cur.execute('INSERT INTO issued(m_id,book_acc_no,book_name,author,f_name,phone_no,status,Date_issued) VALUES(%s,%s,%s,%s,%s,%s,%s,CURDATE()).format(mid,book_choice,b_name,a_name,k,pno)')
        mycon.commit()

        print('Your book has been successfully issued. Kindly note that your due date is 1 month from the date of issuance.')
        print('Thank you and have a good day!')
        proceed=input('Enter any key to proceed... ')
        cans=input('Do you want to issue any more books?y/n ')

```

```

#To return books by members

def return_book():
    cans="y"
    while (cans=="y"):
        b_no=int(input("Enter accession no. of the book you would like to return: "))
        cur.execute("UPDATE book_details SET availability = \"{}\" WHERE acc_no = {}".format("Available",b_no))#updating in book table
        mycon.commit()
        cur.execute("UPDATE issued SET status = \"RETURNED\" WHERE book_acc_no = {}".format(b_no))#updating in issued table
        mycon.commit()

        cur.execute("UPDATE issued SET Date_returned = CURDATE() WHERE book_acc_no = {}".format(b_no))#updating in issued table
        mycon.commit()
        print('Your book has been successfully returned.')
        print('Thank you and have a good day!')
        proceed=input('Enter any key to proceed...')
        cans=input("Do you want to return any more books?y/n ")

```

""

The next set of functions are specifically the types which can view,insert or delete both book and member records, as this is considered as sensitive information.

The following falls under the category of ADMIN RIGHTS.

Kindly note that Admin id:127 and passkey-Czennie

""

```

#ADMIN RIGHTS

def book_exc():
    print('Editing/viewing library books-----')
    cans="y"
    while (cans=="y"):

        chooce=input('What would you like to do? \n V-To view all library book records \n A-To add library book records \n D-To delete library book records \n E-Exit \t >>>')

        if chooce=='V':
            cur.execute('SELECT*FROM book_details')
            det=cur.fetchall()
            chnge=pd.DataFrame(det)
            print(chnge)

        elif chooce=='A':
            cur.execute('SELECT acc_no FROM book_details WHERE acc_no=(SELECT MAX(acc_no) FROM book_details)')
            chk1=cur.fetchall()
            chk2=chk1.pop(0)#type conversion
            chk3= chk2[0]
            last_num=int(chk3)

            print('Kindly note that the last book entry accession no. is:',last_num)
            print('Format as follows eg. 2003,3005,etc.')

        b_no=int(input('Book Accession Number: '))
        b_name=input('Book name: ')
        a_name=input('Author name: ')

```

```

print('Choose any genre from the following: if it doesn't exist...add it:')
genre_list=pd.Series(['Fantasy','Self-Help','Historical
Fiction','Autobiography','Psychology','Astrology','Detective Fiction','Children','Humour','Philosophy','Hindi
Literature','Young Adult Fantasy'])
print(genre_list)

genre=input('Book Genre: ')
ava=input('Available/Unavailable: ')
lang=input('Book language: ')
cur.execute('INSERT INTO book_details(acc_no,book_name,author_name,genre,availability,language)
VALUES(?,?,?,?,?,?)'.format(b_no,b_name,a_name,genre,ava,lang))
mycon.commit()

print('Book successfully added to the Official Neo Library Book Records.')

elif choice=='D':
    print('Take a look at the book list first.')
    cur.execute('SELECT acc_no,book_name FROM book_details')
    info=cur.fetchall()
    info1=pd.DataFrame(info)
    print(info1)

    book_no=int(input('Book Accession number of the book to be DELETED: '))
    cur.execute('SELECT book_name FROM book_details WHERE acc_no=%s'%(book_no))
    chk1=cur.fetchall()
    chk2=chk1.pop(0)#type conversion
    chk3= chk2[0]
    chk4=str(chk3)
    print('Are you sure you wanna delete---',chk4)

    conf=input('Enter C to confirm:')
    if conf=='C':
        cur.execute('DELETE FROM book_details WHERE acc_no = %s'%(book_no))
        mycon.commit()

    print('Book successfully deleted from the Neo Library Records.')

elif choice=='E':
    exit_func()

else:
    print('Invalid input! Try again.')
    pass

cans=input("Do you want to continue or not? y/n ")

```

```

def member_exc():
    print('Editing/viewing Library Member Records-----')
    cans="y"
    while (cans=="y"):
        chooce=input('What would you like to do? \n V-To view all library member records \n A-To add a new member record \n D-To delete a member record \n E-Exit \t >>>')

        if chooce=='V':
            cur.execute('SELECT*FROM member_details')
            det=cur.fetchall()
            chnge=pd.DataFrame(det)
            print(chnge)

        elif chooce=='A':
            cur.execute('SELECT member_id FROM member_details WHERE member_id=(SELECT MAX(member_id) FROM member_details)')
            chk1=cur.fetchall()
            chk2=chk1.pop(0)#type conversion
            chk3= chk2[0]
            last_num=int(chk3)

            print('Choose a unique id greater than ',last_num)
            check_id()

            mem_id=input('Confirm your id(in digits):')
            f_name = input('Enter first name: ')
            l_name = input('Enter last name: ')
            add = input('Enter address: ')
            dob = input('Date of Birth (YYYY-MM-DD): ')
            phone = input('Contact number: ')
            password=input('Set password: ')
            email=input('Enter your email id:')

            cur.execute('INSERT INTO member_details(member_id,first_name,last_name,address_,dob,phone_no,pass_key,email_id) VALUES(?,?,?,?,?,?,?,?)',mem_id,f_name,l_name,add,dob,phone,password,email))
            mycon.commit()
            print('New member successfully added to the Official Neo Library Member Records.')

        elif chooce=='D':
            print('Take a look at the existing member records first.')
            cur.execute('SELECT member_id,first_name,last_name FROM member_details')
            info=cur.fetchall()
            info1=pd.DataFrame(info)
            print(info1)

            mem_no=int(input('Member id number of the chosen record to be DELETED: '))
            cur.execute('SELECT first_name FROM member_details WHERE member_id=%s',(mem_no))
            chk1=cur.fetchall()
            chk2=chk1.pop(0)#type conversion
            chk3= chk2[0]
            chk4=str(chk3)
            print('Are you sure you wanna delete the records of---',chk4)

            conf=input('Enter C to confirm:')
            if conf=='C':
                cur.execute('DELETE FROM member_details WHERE member_id = %s',(mem_no))
                mycon.commit()
                print('Member records of the chosen member successfully deleted from the Neo Library Records.')

```

```

elif chooce=='E':
    exit_func()

else:
    print('Invalid input! Try again.')
    pass

cans=input("Do you want to continue or not?y/n ")

#Admin Menu-Modify books and library member records
def menu3():
    print('Welcome Admin! Good to see you!')
    choose=input('What would you like to do? \n B-To edit/view all library books \n M-To edit/view library member details \n E-Exit \t >>>')
    if choose=='B':
        book_excl()
    elif choose=='M':
        member_excl()
    elif choose=='E':
        exit_func()
    else:
        print('Invalid input. Try again.')
        pass

#DISPLAY MENU 2 - Issue, Return, Exit
def menu2():

print('\nMENU\n')
choice=input('\tSo what would you like to do today? \t I-Issue a book \t R-Return a book \t E-Exit \t >>>')

if choice=="I":
    issue_book()

elif choice=="R":
    return_book()

elif choice == 'E':
    exit_func()

```

```

#Login function has been added here for a reason, as in case of admin logins---startmenu1 to login to menu3.
def login():
    correct=False
    while not correct:
        global mid
        mid=int(input('Login ID: '))
        pwd=input('Passkey: ')

        cur.execute('SELECT member_id FROM member_details WHERE pass_key=%s AND member_id = %s;%' 
        (pwd,mid))
        a=cur.fetchall()
        c=a.pop(0)#type conversion
        d= c[0]
        u=cur.execute('SELECT first_name FROM member_details WHERE pass_key=%s AND member_id = %s;%' 
        (pwd,mid))
        n=cur.fetchall()
        l=n.pop(0)#type conversion
        r=l[0]
        global k
        k=str(r)

    #ADMIN Login
    if mid==127 and pwd=='Czennie':
        menu3()

    elif d==mid:
        correct=True
        print('Welcome back',k,'!')
        menu2()
        break

    else:
        print('Try again!')
        pass

#DISPLAY MENU1 - Login, Register, Exit
def startmenu1():
    print(" Welcome to the Neo Culture Library ")
    entry = input("1. Login - L \n2. Register - R \n3. Exit - E \n\n ^_^\n --- ")

    if entry == 'L':
        login()

    elif entry == 'R':
        register()
        menu2()

    elif entry == 'E':
        exit_func()

    startmenu1()

```

OUTPUT

LOGIN

```
MySQL connection successfully established.  
Welcome to the Neo Culture Library
```

- 1. Login - L
- 2. Register - R
- 3. Exit - E

```
^ ^ --- L  
Login ID: 11  
Passkey: onyourmark  
Welcome back Mark !
```

REGISTER

```
Welcome to the Neo Culture Library  
1. Login - L  
2. Register - R  
3. Exit - E  
  
^ ^ --- R  
So you wanna become a member of our low profile hi-tech library?  
Sure.  
Here you go.  
Enter a chosen id number:99  
The ID already exists, try again  
Enter a chosen id number:91  
Confirm your id:91  
Enter first name: Ryugji  
Enter last name: Ken  
Enter address: 56,xyza,Shibuya  
Date of Birth (YYYY-MM-DD): 1999-12-25  
Contact number: 3309678193  
Set password: thatteatree  
Enter your email id:draken@vc.com  
Welcome to the club Ryugji !
```

RETURN

MENU

```
So what would you like to do today? I-Issue a book  
R-Return a book E-Exit >>>R  
Enter accession no. of the book you would like to return: 4000  
Your book has been successfully returned.  
Thank you and have a good day!  
Enter any key to proceed...  
Do you want to return any more books?y/n |
```

ISSUE

MENU

```

So what would you like to do today?           I-Issue a book
R-Return a book      E-Exit      >>>I
You can choose any genre from the following:
0          Fantasy
1          Self-Help
2  Historical Fiction
3          Autobiography
4          Psychology
5          Astrology
6  Detective Fiction
7          Children
8          Humour
9          Philosophy
10         Hindi Literature
11         Young Adult Fantasy
dtype: object
Select a genre name\IN CASE SENSITIVE:Detective Fiction
      0          1 ...        4          5
0  3006  Murder on the Orient Express  ... Available English
1  3007  The A.B.C. Murders  ... Available English
2  3008  And Then There Were None  ... Available English

[3 rows x 6 columns]
Enter accession no. of the book you would like to issue: 3007
Book details-----
0          3007
1  The A.B.C. Murders
2  Agatha Christie
3  Detective fiction
4          Available
5          English
dtype: object
Your book has been successfully issued. Kindly note that your due date is
1 month from the date of issuance.
Thank you and have a good day!
Enter any key to proceed...|
```

ADMIN THINGS:

VIEWING

```

Welcome Admin! Good to see you!
What would you like to do?
B-To edit/view all library books
M-To edit/view library member details
E-Exit      >>>M
Editing/viewing Library Member Records-----
What would you like to do?
V-To view all library member records
A-To add a new member record
D-To delete a member record
E-Exit      >>>V
      0          1          2          3          4          \
0  11    Mark     Lee  1,Inception Street,Neo City  1999-04-02
1  22  Mafalda   Hopkirk  2,The Seed, Neo City  1945-07-20
2  33    Johnny    Suh  3,Sputnick Road, Neo City  1990-09-09
3  44    Katsuki   Bakugo  4,Godric Hollow, Ncity  2000-05-01
4  55    Rowena  Ravenclaw  5,Snorkacks Hill, Laputa  1999-08-18
5  66  Yangyang    Liu  6,Elysian Place, Laputa  2000-11-03
6  77      Luna  Lovegood  7,Mystic Falls, Ncity  2004-12-25
7  88      Shoto Todoroki  8,The Waterfall, Vercity  2000-06-22
8  91    Ryugji     Ken  56,xyzza,Shibuya  1999-12-25
9  99      Izuku Midoriya  9,U.A. School  2000-02-22
10 100      Gojo   Satauro  17,Ikigai Lake,Laputa  1999-09-19
11 111    Itadori    Yuji  9,Jujutsu High,Tokyo  2004-08-25
12 127      Lune   Fleur  Turnip Turn, Lake Yamanoko  2003-06-24
```

ADMIN THINGS: ADDING A NEW BOOK

```
Login ID: 127
Passkey: Czennie
Welcome Admin! Good to see you!
What would you like to do?
    B-To edit/view all library books
    M-To edit/view library member details
    E-Exit      >>>B
Editing/viewing library books-----
What would you like to do?
    V-To view all library book records
    A-To add library book records
    D-To delete library book records
    E-Exit      >>>A
Kindly note that the last book entry accession no. is: 5001
Format as follows eg. 2003,3005,etc.
Book Accession Number: 5002
Book name: The Song of Achilles
Author name: Madeline Miller
Choose any genre from the following; if it doesn't exist...add it:
0
1          Fantasy
2          Self-Help
3          Historical Fiction
4          Autobiography
5          Psychology
6          Astrology
7          Detective Fiction
8          Children
9          Humour
10         Philosophy
11         Hindi Literature
12         Young Adult Fantasy
dtype: object
Book Genre(): Historical Fiction
Available/Unavailable: Available
Book language: English
Book successfully added to the Official Neo Library Book Records.
Do you want to continue or not?y/n |
```

ADMIN THINGS: DELETING MEMBER DETAILS

```
Login ID: 127
Passkey: Czennie
Welcome Admin! Good to see you!
What would you like to do?
B-To edit/view all library books
M-To edit/view library member details
E-Exit      >>>M
Editing/viewing Library Member Records-----
What would you like to do?
V-To view all library member records
A-To add a new member record
D-To delete a member record
E-Exit      >>>D
Take a look at the existing member records first.
     0      1      2
0   11    Mark     Lee
1   22  Mafalda  Hopkirk
2   33    Johnny    Suh
3   44   Katsuki   Bakugo
4   55   Rowena   Ravenclaw
5   66  Yangyang   Liu
6   77      Luna  Lovegood
7   88     Shoto  Todoroki
8   91   Ryugji     Ken
9   99    Izuku  Midoriya
10 100   Gojo   Satauro
11 111  Itadori   Yuji
12 127    Lune   Fleur
Member id number of the chosen record to be DELETED: 99
Are you sure you wanna delete the records of--- Izuku
Enter C to confirm:C
Member records of the chosen member successfully deleted from the Neo Library Records.
Do you want to continue or not?y/n |
```

SCOPE AND IMPROVEMNET

In the process of making this project, despite having many ideas for further improvements due to certain limitations, couldn't execute them.

Thereby, the ideas that can be implemented for further enhancements:

- Converting the entire program into a stand-alone GUI using Tkinter/kivy libraries.
- Adding a contact module for communication and notification exchange.
- Implementing foreign key constraints in the program code, for better functionality.

NOTES

- Admin login is uniques and well-identified by the code.
- For Admin Id-127, Passkey-Czennie
- The 'issued' table is not fixed and is something like a rotational/everchanging table since it stores and updates information with the issuing and returning phase.
- Issued table was made alternatively.
- Version and language: Python 3.8.6 and MySQL 5.7.

BIBLIOGRAPHY

- FreeCodeCamp
- CS NCERT Grade 12
- Tutorialspoint.com
- W3schools.com
- Stack Overflow
- GeeksforGeeks
- realpython.com
- Computer Science by Sumita Arora

WITH BOOKS, YOU
CAN GO PLACES!

