Working Description

1. Creating the Database and required Tables

The program will be creating a database named `library` and it'll contain two tables. The first named `lib` will contain all the books in the library and their respective details.

The second table named `records` will contain all the records of the library.

> For example, 'borrowing/returning a book'.

The creation of the database and tables will be done in the following functions in the `createdb.py` file.

```
def createdb():
    cur.execute('create database library')
    cur.execute('use library')
    cur.execute('create table lib (snobook int primary key not null, name
varchar(30) not null, author varchar(30) not null, price int not null, pages int
not null, available boolean not null);')
    cur.execute('create table record (sno int primary key not null, dofentry date
not null, name varchar(30) not null, class varchar(10) not null, snobook int not
null')
    db.commit()
```

2. The Library functions

The `libraryfunc.py` file contains all the library functions below. These functions work with all the books in the library database.

2.1 ADD A BOOK

Inserting books in the `lib` table in the library database. The software will take **name**, **author**, **price** and **no. of pages** as the input and will insert the book in the table with an appropriate serial number (primary key).

The fucntion is named as `insertbook()` in the `librryfunc.py` file and is defined as...

```
def insertbook():
    while True:
        try:
            name = input('Enter the name of the book: ')
            author = input( 'Enter the name of the author: ')
            price = int(input('Enter the price of the book: '))
            pages = int(input('Enter the number of pages ini the book: '))
            break
        except ValueError:
            print('Value Error! Please try again!')
```

```
(lenofdata+1,name,author,price,pages,'1'))
    try:
        db.commit()
        print('Book added successfully!')
    except mysql.connector.Error as err:
            print('ERROR! please try again...')
            print(err)
            insertbook()
```

2.2 DELETE A BOOK

Removing a book from the library incase if the book isn’t available in the library anymore. This function will just take the name of the book as the input.

The fucntion is named as `deletebook()` in the `libraryfunc.py` file and is defined as...

```
def deletebook():
    book = input('Enter the name of the book to delete: ')
    if checkbook(book) == True:
        for i in data:
            if i[1]==book:
                cur.execute('''delete from lib where name="%s"'''%(book))
                db.commit()
                print('Book deleted successfully!')
    else:
        print('Book not in database! Please try again!')
        deletebook()
```

2.3 UPDATE A BOOK

Change/update the details of a book. The function takes the name of the book as input.

The user can change any detail of the book like **name**, **author**, **price** and **genre**.

The function is named as `updatebook()` in the `libraryfunc.py` file and is defined as...

```
def updatebook():
    def inupbook(book,prop,newpropval):
        for i in data:
            if i[1]==book:
                cur.execute('''update lib set %s="%s" where name="%s"'''%
(prop,newpropval,book))
                db.commit()
    book = input('Enter the name of the book you would like to update: ')
    if checkbook(book) == True:
        print('''
        CHOOSE AN OPTION
        1) CHANGE NAME OF THE BOOK
        2) CHANGE AUTHOR OF THE BOOK
        3) CHANGE NUMBER OF PAGES
        4) CHANGE PRICE
        ''')
        optupbook = int(input('Enter your choice: '))
        if optupbook==1:
            newpropval = input('Enter new name of the book: ')
            inupbook(book,'name',newpropval)
        elif optupbook==2:
            newpropval = input('Enter new name of the author: ')
            inupbook(book,'author',newpropval)
        elif optupbook==3:
            newpropval = int(input('Enter new number of pages: '))
            inupbook(book,'pages',newpropval)
        elif optupbook==4:
            while True:
                try:
                    newpropval = int(input('Enter new price: '))
                    inupbook(book,'price',newpropval)
                except ValueError:
                    print('Please enter an integer')
                    updatebook()
        print('Book updated successfully!')
    else:
        print('Book not in database! Please try again')
        updatebook()
```

2.4. SEARCH A BOOK

Search any book with either the name of the book or name of the author, get all the details of the book. Searching for an author will get all the books written by the author.

The function is named as `getbook()` in the `libraryfunc.py` file and is defined as...

```
def getbook():
    print('''
        CHOOSE AN OPTION
        1) SEARCH BOOK BY NAME
        2) SEARCH BOOK BY AUTHOR
        3) GO BACK
        ''')
    optgb = int(input('Enter your choice :'))
    if optgb == 1:
        book = input('Enter a book name: ')
        if checkbook(book) == True:
            for i in data:
                if i[1] == book:
                    print(f'Nice {i[0]}')
        else:
            print('Book does not exist! Try again!')
            getbook()
    elif optgb == 2:
        author = input('Enter an author name: ')
        if checkauthor(author) == True:
            for i in data:
                if i[2] == author:
                    print(f'Nice {i[0]}')
        else:
            print('Author does not exist! Try again!')
            getbook()
    elif optgb == 3:
        main.libscreen()
```

2.5 CHECK BOOK AVAILABILITY

Check the availibilty of a book with the name of the book from the database.

The function is named as `checkbookavail()` in the `libraryfunc.py` file and is defined as...

```
def checkbookavail(bookno):
    for i in data:
        if i[0]==bookno:
            if i[-1]==1:
                print('Book is Available!')
            else:
                print('Book is not available!')
```

3. RECORD FUCNTIONS

The `recordfunc.py` file contains all the record functions below. These function work with all the records and the library database.

3.1 BORROW A BOOK

The function will take your **name**, **class** and the **serial number** of the book you want to borrow from the library as the input and add to the record. This function will also change the availability of the borrowed book.

The function to insert a record in the record database `insertrecord()` in `recordfunc.py` file and the fucntion to `borrowbook(book)` in the `libraryfunc.py` and are defiined as...

```
def insertrecord(name,classs,date,bookno):
    cur2.execute('''insert into record values(%d,"%s","%s","%s",%d)'''%
(lenofdata+1,date,name,classs,bookno))
    db2.commit()
```

```
def borrowbook(bookno):
    for i in data:
        if i[0]==bookno:
            cur.execute('''update lib set available="0" where sno="%s"'''%
(bookno))
            db.commit()
            print('UPDATED DATABASE SUCCESFULLY')
```

3.2 RETURN A BOOK

The function will take your **name**, **class** and the **serial number** of the book you want to return from the library as the input and add to the record. This function will also change the availability of the returned book.

The function to insert a record in the record database `insertrecord()` in `recordfunc.py` file and the fucntion to `returnbook(book)` in the `libraryfunc.py` and are defiined as...

```
def insertrecord(name,classs,date,bookno):
    cur2.execute('''insert into record values(%d,"%s","%s","%s",%d)'''%
(lenofdata+1,date,name,classs,bookno))
    db2.commit()
```

```
def returnbook(bookno):
    for i in data:
        if i[0]==bookno:
            cur.execute('''update lib set available="1" where sno="%s"'''%
(bookno))
            db.commit()
            print('UPDATED DATABASE SUCCESFULLY')
```