Computer Science Project File

[Library Management System]

Python and MySQL

Project Prepared By

Ajay Kisku and Rohan Kumar

XII (Science)

Session: 2020-2021

D.A.V. Public School Tenughat Dam,

Bokaro, Jharkhand

Contents | Index

Certificate

This is to certify that Ajay Kisku of class XII from Science stream, D.A.V. Public School Tenughat Dam, Bokaro, Jharkhand has successfully completed his project in Computer practical for AISSCE as prescribed by CBSE in year 2020-2021.

Date:

Registration:

Signature of Examiner

Signature of External Examiner

Acknowledgement

I thank my Computer Science teacher Mr. Mukul Kumar Singh for his guidance and support. I also thank our principal Mrs. Archana Mishra. I would also like to thank my parents and my sister for encouraging me during the course of this project. Finally, I would like to thank CBSE for giving me this opportunity to undertake this project.

-Ajay Kisku

Requirements

Windows 7 or higher with Python 3x Installed

Libraries and Modules: os, time, mysql.connector, getpass, logzero, keyboard, winsound

LMS

Library Management System

As the name this program would keep a proper record of books in library, books lending record and late fines record too. It's a python-based CLI program which uses MySQL for data management. It has been specially designed for school library. For Windows OS only.

It is a menu-based program which uses colored texts which make it easy to understand and it's very easy to use.

This program is available on GitHub - https://github.com/Raj-aj/LMS

As program will be opened it will ask for MySQL credentials as it will automatically create a well-defined database and tables inside it. Three wrong attempts for credentials and it will be closed.

Database Name - LMS

Tables Name - Books, Book_Borrowers and Students

What options you will have in the program?

For Keeping Books Record

1. Add Books
2. Update Books
3. Delete Books

For Lend and Receive Books

1. Issue Books
2. Submit Books

To Search Books

1. Show All Books
2. Search Local
3. Search Online Books

For Late Fines

1. Total Fines
2. Late Fines History

Its special features

Clear screen by clear command.

Beep Sound with incorrect commands.

Link to developers' website is there.

It has a book art, welcome art and welcome message by author's voice.

It can search books online in many websites for pdf version for free, look for the book in e-commerce book stores and also summary of the book in YouTube.

Limitation of the program is the limitation of MySQL and Python itself.

Author - *Ajay Kisku*

< CODE >

```python
import os,logzero,keyboard,time
import mysql.connector as mc
from getpass import getpass
from winsound import Beep
import winsound
import webbrowser as web

keyboard.press_and_release('alt+enter');time.sleep(1)

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

def beep(n):
    for i in range(n):
        Beep(4000,400)

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

log_in_chance=3
def connect_to_DB():
    global mydb,log_in_chance,cur
    while log_in_chance>0:
        try:
            print('\n Your mysql details are required\n\n')
            user_=input('  '        ┌──User_Name : ').strip() ;print('        ')
            pass_=getpass('        └──Password : ').strip()
            mydb=mc.connect(host='localhost',user=user_,password=pass_,)
            print('\n\n\t Connecting MySQL and Python : ')
            for i in range(51):
                time.sleep(0.07)
                print('\t',chr(9608)*i,str(i*2)+'%',end='\r')
            time.sleep(0.3)
            os.system('color 27');os.system('cls');print('\n'*10)
            print('You are most welcome'.title().center(os.get_terminal_size()[0],'
'));time.sleep(0.5)
            winsound.PlaySound("LMS.wav", winsound.SND_ASYNC)
            cur=mydb.cursor(buffered=True)
            os.system('cls')
            log_in_chance=0
        except:
            log_in_chance-=1
            if log_in_chance==0:
                os.system('cls')
                for i in range(7):
                    os.system('color 47')
                    beep(1);print('\n\n')
                    print('You are not a valid
user'.title().center(os.get_terminal_size()[0]))
                time.sleep(1);quit()
            else:
                os.system('cls & color 17')
                for i in range(2):
                    beep(1);print('\n\n')
                    print('Incorrect User_Name and Password. Try again
!'.center(os.get_terminal_size()[0],' '))

            time.sleep(1)
            os.system('cls & color 07')
            connect_to_DB()

#------------------------------------------------------------------------------
```

```
58
59 def database_structure():
60     try:
61         sql_file=open(r"db_structure(LMS).sql").read().split(';')[0:-1]
62         for query in sql_file:
63             cur.execute(query)
64         cur.execute('commit;')
65         cur.execute('use LMS;')
66     except:
67         pass
68
69 #-------------------------------------------------------------------------------
70
71 def welcome_msg():
72
73     print('''
74
75                     mmmmm                                                 mmmmm
76                     MM                                                       MM
77                     MM         `7MMF'       `7MMM.      ,MMF' .M""bgd        MM
78                     MM           MM           MMMb    dPMM  ,MI    "Y        MM
79                     MM           MM           M YM   ,M MM  `MMb.            MM
80                     MM           MM           M  Mb  M' MM    `YMMNq.        MM
81                     MM           MM      ,    M  YM.P'  MM  .     `MM        MM
82                     MM           MM     ,M    M  `YM'   MM  Mb     dM        MM
83                     MM         .JMMmmmmMMM .JML. `'  .JMML.P"Ybmmd"         MM
84                     MM                                                       MM
85                     MMmmm                                                 mmmMM
86
87     ''')
88     database_structure()
89
90 #-------------------------------------------------------------------------------
91
92 def terminal_look():
93     os.system('title [ Library Management System ] [ Aj_Indudstires ]')
94     os.system('color 07')
95     welcome_msg();time.sleep(2);os.system('cls')
96
97     print('\n'+'—'*os.get_terminal_size()[0])
98     print('Library Management System'.center(os.get_terminal_size()[0],' '))
99     print('—'*os.get_terminal_size()[0])
100
101 #-------------------------------------------------------------------------------
102
103 def shelf():
104     print('''
105  +---------------------+
106  |     Book Shelf      |
107  +---------------------+
108     ''')
109     cur.execute('select Book_ID,Book_Name,count(*)  from books group by Book_Name;')
110     count=1
111     for book in cur:
112         print(' ',str(count).rjust(3,' ')+'.',(book[1].ljust(30,'.'))[:30]+'...'+'
    ID:'+str(book[0]),'| Quantity:'+str(book[2]))
113         count+=1
114     print()
115     cur.execute('select count(*) from books')
116     for i in cur:
```

```
        print(f'  \033[0;37;41m Total Quantity \033[0;37;40m\033[0;30;47m {i[0]} \033[0;37;40m ')

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

def commands_list():

    print('''
        +--C◎MMANDS-------------------------------------+
        | »[C] : See these commands again               |
        +-----------------------------------------------+
        |   [S] : Search Books                          |
        |   [1] : Issue/Lend Books                      |
        |   [2] : Receive/Submit Books                  |
        |   [3] : Add Books     [A]                     |
        |   [4] : Update Books  [U]                     |
        |   [5] : Delete Books  [D]                     |
        |   [F] : Fine Details                          |
        |   [T] : Total Fine                            |
        |   [X] : Close the Program [ CTRL+C ]          |
        | [CLS] : Clear Screen [clear]                  |
        | [ALL] : List of all books in library          |
        +-----------------------------------------------+
        | Online_Book | Art | Devs | Kuch  | Help | Aj |
        +-----------------------------------------------+
    ''')

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

def search_books():
    print('''
    +-----------------------------+
    |   Search and Filter Books    |
    +-----------------------------+
    ''')

    def sql_search_query(column):
        print()
        cur.execute(f'select Book_name,book_id,available from books where {column} like "%{search}%"')
        slno=1
        for i in cur:
            if i[2]=='yes':
                print('   ',str(slno).rjust(2,' ')+'. '+i[0].title().ljust(30,'.'),'Book ID :',(i[1]),', Avialable' )
            else:
                print('\033[1;31;40m   ',str(slno).rjust(2,' ')+'. '+i[0].title().ljust(30,'.'),'Book ID :',(i[1]),', Not Available \033[0;37;40m')
            slno+=1

        if slno==1:
            print('\033[1;31;40mNo Book Found!\033[0;37;40m')

        print()
        print('-'*(os.get_terminal_size()[0]))

    while True:
        search=input('\nSearch Books : ').strip().lower()
        search=' '.join(search.split())

```

```
        if search=='':
            print('Searching Book(s) Canceled')
            break
        elif search.isdigit():
            sql_search_query(column='book_id')
        elif search.startswith('author'):
            search=search.split('-')[-1]
            sql_search_query(column='author')
        elif search.startswith('publisher'):
            search=search.split('-')[-1]
            sql_search_query(column='publisher')
        elif search.startswith('subject') or search.startswith('sub'):
            search=search.split('-')[-1]
            sql_search_query(column='subject')
        elif search.startswith('class'):
            search=search.split('-')[-1]
            sql_search_query(column='class')
        elif search.startswith('language') or search.startswith('lang'):
            search=search.split('-')[-1]
            sql_search_query(column='language')
        else:
            sql_search_query(column='book_name')

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

internet=False
def is_internet():
    import requests
    global internet
    try:
        re=requests.get('https://www.google.com')
        if re.status_code==200:
            internet=True
    except:
        internet=False

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

def online_books():
    is_internet()
    if internet:
        get=input('  Search on web for book : ').strip().replace(' ','+')
        if get=='':
            pass
        else:
            time.sleep(1)
            web.open(f'https://www.youtube.com/results?search_query=book+summary+
{get}')
            web.open(f'https://www.flipkart.com/search?
q=book%20'+get.replace("+","%20"))
            web.open(f'https://www.amazon.in/s?k=
{get}&i=stripbooks&ref=nb_sb_noss_1')
            web.open(f'https://www.google.com/search?tbm=bks&q={get}')
            web.open(f'https://www.google.com/search?q=filetype:pdf%20{get}')
            web.open(f'https://archive.org/details/books?and%5B%5D={get}')
            web.open( 'https://1lib.in/s/'+get.replace('+','%20'))
            web.open(f'http://books.rediff.com/#!{get}')
            web.open(f'https://www.pdfdrive.com/search?q={get}')

            time.sleep(2)
```

```
            print('  [I hope you got your book]')
    else:
        print('  No Internet Connection')

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

def add_books():
    print('''
    +--------------------+
    |   Adding Books     |
    +--------------------+
     ''')

    global Book_Name,Quantity,Class ,Subject ,Language ,Publisher , Author

    while True:
        Book_Name = input('\n\tBook Name : ').strip().title()

        if Book_Name=='':
            print('\tAdding Book(s) Canceled')
            break

        else:

            global Quantity,Class
            while True:
                Quantity  = input('\tQuantity  : ').strip()
                if Quantity=='':
                    pass
                elif Quantity.isdigit():
                    if int(Quantity)>0:
                        Quantity=int(Quantity)
                        break
                    else:
                        print('\t\t\t\t\033[1;31;40m*Quantity should be greater than
zero*\033[0;37;40m')
                        pass
                else:
                    print('\t\t\t\t\033[1;31;40m*Enter the digits
only*\033[0;37;40m')
                    pass

            while True:
                Class     = input('\tFor Class : ').strip()
                if Class=='':
                    pass
                elif Class.isdigit():
                    if int(Class)>0:
                        Class=int(Class)
                        break
                    else:
                        print('Class should be more than zero'.title())
                        pass
                else:
                    print('\t\t\t\t\033[1;31;40m*Enter the digits
only*\033[0;37;40m')
                    pass

            Subject   = input('\tSubject   : ').strip().title()
            Language  = input('\tLanguage  : ').strip().title()
```

```
            Publisher = input('\tPublisher : ').strip().title()
            Author    = input('\tAuthor    : ').strip().title()

            add_book = (f"insert into Books
(Book_Name,Class,Subject,Language,Publisher,Author,Available) values \
                        ('{Book_Name}',
{Class},'{Subject}','{Language}','{Publisher}','{Author}','yes')")

            ask=input('\n\tAdding book confirmation [y/n] : ').strip().lower()

            if ask=='y':
                for i in range(Quantity):
                    try:
                        cur.execute(add_book)
                        cur.execute('commit')
                    except:
                        print('\t\t*error due to sql datatype limit(s)*')
                if len(Book_Name)>20:
                    print('\n\t\033[32m>',str(Quantity)+f' "{Book_Name[0:20]+"..."}"
books(s) Added\033[1;37;40m','\n')
                else:
                    print('\n\t\033[32m>',str(Quantity)+f' "{Book_Name[0:20]}"
books(s) Added\033[1;37;40m','\n')

            else:
                print('\t\033[1;31;40mBook not added\033[0;37;40m')
                pass
        print('-'*(os.get_terminal_size()[0]))
        print()

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

def update_books():
    print('''
    +--------------------+
    |   Updating Books   |
    +--------------------+
    ''')
    while True:
        update=input('\nEnter Book ID : ')
        if update=='':
            print('Updating Book(s) Canceled');break
        elif update.isdigit():

            column='Book_Name Class Subject Language Publisher Author'.split()
            cur.execute(f'select
book_id,book_name,class,subject,language,publisher,author from books')
            global current_data
            current_data=[]

            book_found=False
            for detail in cur:
                if int(update)==detail[0]:
                    cur.execute(f'select
book_name,class,subject,language,publisher,author from books where Book_ID="
{update}"')
                    for i in cur:
                        current_data.extend(i)
                        book_found=True;break
                else:
```

```
                    pass
            if book_found:
                print('                                                Enter New Details')
                print('                                                ----------------')
                new_record=[]
                for i in range(len(column)):
                    if len(str(current_data[i]))<18:
                        change=input(f'{i+1}.{column[i]} '.ljust(15,'-')+f' [{current_data[i]}] : '.rjust(29,'-')).strip().title()

                        if change=='':
                            new_record.append(current_data[i])
                        else:
                            new_record.append(change)

                    else:
                        change=input(f'{i+1}.{column[i]} '.ljust(15,'-')+f' [{current_data[i][0:18]}...] : '.rjust(29,'-')).strip().title()
                        if change=='':
                            new_record.append(current_data[i])
                        else:
                            new_record.append(change)

                print()

                if current_data!=new_record:
                    if str(new_record[1]).isdigit():
                        commit=input('\nSure about the changes? [y/n] : ').lower()

                        if commit in 'y yes'.lower().split():
                            update_query=f"update books set \
                                        book_name='{new_record[0]}',class={new_record[1]},subject='{new_record[2]}',language='{new_record[3]}',\
publisher='{new_record[4]}',author='{new_record[5]}',Last_Update_On=CURRENT_TIMESTAMP \
                                        where Book_ID={int(update)}"

                            cur.execute(update_query)
                            cur.execute('commit')
                            print('Changes Saved, Sucessfully!')

                        elif commit in 'n no'.lower().split():
                            print('Changes Not Saved')
                        else:
                            print('Changes Not Saved')
                    else:
                        print('Changes Not Saved\nPlease enter only numeric value in the Class field.')

                else:
                    print('No Changes Made')

                print()

            else:
                print('There is no book with your given Book ID')

```

```
        else:
            print('\t\t\t\t\033[1;31;40m*Enter the digits only*\033[0;37;40m')
            pass

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

def delete_books():
    print('''
    +--------------------+
    |   Deleting Books   |
    +--------------------+
    ''')
    while True:
        delete=input('\n\tEnter Book ID : ')
        if delete=='':
            break
        elif delete.isdigit():
            cur.execute('select Book_ID,book_name from books')
            global found_and_deleted
            found_and_deleted=False
            for id in cur:

                if int(delete) == id[0]:
                    try:
                        cur.execute(f'delete from books where Book_ID = {id[0]}')
                        cur.execute('commit')
                        found_and_deleted=True
                    except:
                        print(id[0],'From else block')
                        print('\tMySql error!')

            if found_and_deleted:
                print('\t\033[32mDeleted : "'+id[1]+'"\033[1;37;40m')
            else:
                print('\t\t\t\033[1;31;40m*There is no such Book ID*\033[0;37;40m')
        else:
            print('\t\t\t\033[1;31;40m*Enter the digits only*\033[0;37;40m')
            pass



    print('\tDeleting Book(s) Canceled')

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

def lend_book():
    print('''
    +--------------------+
    |   Issueing Books   |
    +--------------------+
    ''')
    while True:
        global available
        available=False
        found=False
        while True:
            global id
            id  = input('\n\tBook ID : ').strip()
            if id=='':
                print('\tIssueing book(s) exit')
```

```
                break
            elif id.isdigit():
                if int(id)>0:
                    id=int(id)
                    cur.execute('select book_id from books')
                    for book_id in cur:
                        if id in book_id:
                            found=True
                            break

                    if not found:
                        print('\t\t\t\t\033[1;31;40m*No book found with given ID*\033[0;37;40m')
                        available=False
                        break
                    else:
                        cur.execute(f'select Available from books where book_id={id}')
                        for avail in cur:
                            if avail[0]=='no':
                                found=False
                                available=False
                                print('\t\t\t\t\033[1;31;40mThe book isn\'t available at the moment\033[0;37;40m')
                            else:
                                available=True
                        break

                else:
                    print('\t\t\t\t\033[1;31;40m*Invalid Input*\033[0;37;40m')
                    pass
            else:
                print('\t\t\t\t\033[1;31;40m*Enter the digits only*\033[0;37;40m')
                pass

        if id=='':break

        if found:
            while found:
                global adm_mob
                adm_mob  = input('\tAdm no. : ').strip()
                if adm_mob=='':
                    break
                elif adm_mob.isdigit():
                    if int(adm_mob)>0:
                        adm_mob=int(adm_mob)
                        if len(str(adm_mob))==10:
                            print('\n\t\t\t\tMobile no: ',adm_mob,'is being issued book "_________"\n')
                        else:
                            global adm_found
                            adm_found=False
                            cur.execute('select Adm_no from students')
                            for Adm_no in cur:
                                if adm_mob in Adm_no:
                                    adm_found=True
                                    break
                            if adm_found:
                                break
                            else:
```

```
                                print('\t\t\t\t\033[1;31;40m*No such admission number found*\033[0;37;40m')
                                pass

                    else:
                        print('\t\t\t\t\033[1;31;40m*Invalid Input*\033[0;37;40m')
                        pass
                else:
                    print('\t\t\t\t\033[1;31;40m*Enter the digits only*\033[0;37;40m')
                    pass
        else:
            pass

        if available:
            try:
                cur.execute(f'select book_name from books where book_id ={id}')
                for bname in cur:
                    book=bname[0]

                cur.execute(f'select name from students where adm_no ={adm_mob}')
                for sname in cur:
                    student=sname[0]

                cur.execute(f'select adm_no,name,email_id,phone_no from students where adm_no={adm_mob}')
                for i in cur:
                    cur.execute(f'insert into book_borrowers (adm_no,name,email_id,phone_no,Borrowed_book_id) values {i+(id,)}')
                    cur.execute('select * from book_borrowers')
                    for i in cur:
                        borrower_on=i[6]
                        borrower_no=i[0]

                cur.execute(f'update books set available="no" where book_id={id}')
                cur.execute('commit')
                cur.execute('select max(borrower_no) from book_borrowers')
                for i in cur:

                    print('\n\t\b\033[1;32;40m"\033[1;35;40m'+book+'\033[1;32;40m" book is given to "\033[1;35;40m'+student+'\033[1;32;40m" with Borrower Number :\033[1;35;40m',i[0],'\033[0;37;40m')

            except:
                pass
        else:
            pass
        print()
        print('-'*(os.get_terminal_size()[0]))

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

def submit_book():
    print('''
    +---------------------+
    |   Receving Books    |
    +---------------------+
    ''')
    while True:
        borrower_no=input('\tBorrower_No: ')
```

```
        if borrower_no=='':
            print('\tRecieving Book(s) Canceled')
            break
        elif borrower_no.isalpha():
            print('\t\t\t\t\033[1;31;40m*Enter the digits only*\033[0;37;40m')
            pass
        elif borrower_no.isdigit():
            borrower_no=int(borrower_no)
            cur.execute(f'select borrower_no,Borrowed_book_id,submit_date,fine_amount from book_borrowers where borrower_no={borrower_no}')

            global found
            found=False
            for i in cur:

                if i[0]==borrower_no and i[2]==None:
                    found=True
                    cur.execute(f'update books set available="yes" where book_id={i[1]}')
                    cur.execute(f'update book_borrowers set `submit_date`=CURRENT_TIMESTAMP  where Borrowed_book_id={i[1]}')
                    cur.execute('commit')
                    print('\t\033[32mBook Submitted Sucessfully\033[0;37;40m\n')

                    if i[3]>0:
                        print('\tLate Fine of ₹ '+str(i[3])+' on this book by the Student.')
                    else:
                        print('\t\033[1;36;40mThanks For Submitting the Book in Time. No Late Fine Charge.\033[0;37;40m')

                    print()
                    print('-'*(os.get_terminal_size()[0]),'\n')

            if not found:
                print('\t\033[1;33;40mEither Book has been Submitted or there is no such Borrower ID\033[0;37;40m\n')
        else:
            print('\t\t\t\t\033[1;31;40m*Enter the digits only*\033[0;37;40m')

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

def late_fine():

    cur.execute('update book_borrowers set total_days=datediff(CURRENT_TIMESTAMP,borrowed_date),fine_amount=total_days-5   where `submit_date`=0;')
    cur.execute('update _a.book_borrowers set fine_amount=0 where fine_amount<0')

    print('Color Code Explained :\n Green  > Book Submitted \n Red    > Book Not Submitted + Submit Date has crossed \n Yellow > Book Not Submitted + Submit Date has not crossed\n\n')

    cur.execute('select borrower_no,Borrowed_book_id,name,total_days,submit_date,fine_amount from book_borrowers')

    slno=1
    print('Slno'.rjust(8,' '),'Borrower_No'.rjust(15,' '),'Book_ID'.rjust(15,' '),'Borrower\'s Name'.rjust(19,' '),'Total Days'.rjust(24,' '),'Late Fine'.rjust(13,'
```

```
'))
    print(' ','-'*100)
    for i in cur:
        fine=str(i[5])
        if int(i[3])>5 and  i[4]==None :
            print("\033[1;31;40m"+(str(slno)+'.').rjust(8,' '),str(i[0]).rjust(14,'
'),str(i[1]).rjust(15,' '),' '*5,(str(i[2]).ljust(28,' '))[:28],str(i[3]).rjust(9,'
'),('₹ '+fine).rjust(13,' ')+"\033[0;37;40m")
        elif int(i[3])<=5 and i[4]==None:
            print("\033[1;33;40m"+(str(slno)+'.').rjust(8,' '),str(i[0]).rjust(14,'
'),str(i[1]).rjust(15,' '),' '*5,(str(i[2]).ljust(28,' '))[:28],str(i[3]).rjust(9,'
'),('₹ '+fine).rjust(13,' ')+"\033[0;37;40m")
        else:
            print("\033[1;32;40m"+(str(slno)+'.').rjust(8,' '),str(i[0]).rjust(14,'
'),str(i[1]).rjust(15,' '),' '*5,(str(i[2]).ljust(28,' '))[:28],str(i[3]).rjust(9,'
'),('₹ '+fine).rjust(13,' ')+"\033[0;37;40m")

        print(' ','-'*100)
        slno+=1

    cur.execute('commit')

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

def total_late_fine():
    cur.execute('update book_borrowers set
total_days=datediff(CURRENT_TIMESTAMP,borrowed_date),fine_amount=total_days-5   where
`submit_date`=0;')
    cur.execute('update _a.book_borrowers set fine_amount=0 where fine_amount<0')
    cur.execute('select adm_no,name,sum(fine_amount) from book_borrowers group by
adm_no')
    print(' Total Late Fine\n')
    print(' Adm','\t','Fine','\t','Name')
    print('-'*50)
    for i in cur:
        if i[2]!=0:
            print('',i[0],'\t ₹',i[2],'\t',i[1])
            print('-'*50)

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

def book_art():
    print('''
                    .-~~~~~~~~~-._       _.-~~~~~~~~~-.
                __.'              ~.   .~              `.__
              .'//                  \./                  \\`.
            .'//                     |                     \\`.
          .'// .-~"""""""~~~~-._     |     _,-~~~~"""""""~-. \\`.
        .'//.-"                 `-.  |  .-'                 "-.\\`.
      .'//______.============-..   \ | /   ..-============.______\\`.
    .'______________________________\|/______________________________`.
    ''')

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

def Ajay_Kisku_Colored():
    print()
    times=7
    center='.center(os.get_terminal_size()[0])'
    row1=f'\033[0;37;41m Ajay \033[0;37;40m\033[0;30;47m Kisku \033[0;37;40m  '*times
```

```
    row2=f'\033[0;37;40m\033[0;30;47m Kisku \033[0;37;40m '+f'\033[0;37;41m Ajay \033[0;37;40m\033[0;30;47m Kisku \033[0;37;40m  '*(times-1)+f'\b\033[0;37;41m Ajay \033[0;37;40m'

    for i in range(3):
        print(' '+row1+'\n '+row2+'\n')

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

def about_devs():
    is_internet()
    if internet:
        a='  You will be redirected to developer\'s website...'
        b=''
        for i in a:
            b+=i; time.sleep(0.1)
            print(b,end='\r')
        print(a)

        time.sleep(1)
        os.system('start http://rajaj1.000webhostapp.com/')
        time.sleep(1)
        Ajay_Kisku_Colored()
        print(' [You were on developer site]')
    else:
        Ajay_Kisku_Colored()

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

def run_command():

    print('-'*(os.get_terminal_size()[0]))
    global lms
    lms=input('\033[0;37;45m Run \033[0;30;46m [COMMAND] \033[0;37;40m > ').strip().lower()
    print('-'*(os.get_terminal_size()[0]))

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

def main():
    connect_to_DB()
    terminal_look()
    commands_list()

    while True:
        print()
        run_command()
        print()

        if lms=='':
            pass
        elif lms in 'x exit quit close'.split():
            print('[ Bye! ]          Software by -Ajay Kisku and Rohan Kumar'); time.sleep(1)
            break;exit()
        elif lms in 'cls clear'.split():
            os.system('cls')
        elif lms in 'c cmd cmds command commands'.split():
            commands_list()

```

```
        elif lms in 's find search'.split():
            search_books()
        elif lms in '1 lend issue l i'.split():
            lend_book()
        elif lms in '2 sub submit r rec receive '.split():
            submit_book()
        elif lms in '3 a add addbook'.split():
            add_books()
        elif lms in '4  u update updatebook'.split():
            update_books()
        elif lms in '5 d del delete deletebook'.split():
            delete_books()
        elif lms=='art':
            book_art()
        elif lms in 'shelf allbooks all'.split():
            shelf()
        elif lms in 'pdf web online more online_book online_books
onlinebooks'.split():
            online_books()
        elif lms in 'dev devs devloper aj ajay kisku ajaykisku'.split():
            about_devs()
        elif lms in 't total'.split():
            total_late_fine()
        elif lms in 'f fine'.split():
            late_fine()

        elif lms in 'kuch'.split():
            print('\t\t\tAre kuch nahi!')
        elif lms in 'help'.split():
            print('\t\t\t"God help those who help themselves"')
        elif lms in 'tf ff total '.split():
            total_late_fine()
        else :
            print(f'\033[1;31;40m[Invalid Command!]\033[0;37;40m')
            beep(1)

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

if __name__=='__main__':
    main()
```

</CODE >

Bibliography

Google, Stack Over Flow, GitHub, W3schools, geeksforgeeks,

Edureka, Python Official Documents, related YouTube Videos, etc.