```
from datetime import date, timedelta
from flask import Flask,request,render_template,redirect,url_for
from flaskext.mysql import MySQL
app=Flask(__name__)

def getMysqlConnection():
        mysql = MySQL()
        app.config['MYSQL_DATABASE_USER']='root'
        app.config['MYSQL_DATABASE_PASSWORD']=''
        app.config['MYSQL_DATABASE_HOST']='localhost'
        app.config['MYSQL_DATABASE_DB']='lms'
        mysql.init_app(app)
        connection=mysql.connect()
        cursor=connection.cursor()
        return {"cursor":cursor,"connection":connection}

db=getMysqlConnection()
cursor = db['cursor']
connection = db['connection']

@app.route('/')
def library():
        return render_template('index.html')
"""

@app.route('/admin.html')
def adminblog():
```

```
    return render_template('admin.html')


@app.route('/admin.html',methods=['POST'])
def authenticate():
    admin_username=request.form.get('uname')
    admin_password=request.form.get('pwd')
    cursor.execute("SELECT * FROM admin_login WHERE username='"+admin_username+"' and password='"+admin_password+"'")
    myresult=cursor.fetchone()
    if myresult is None:
        return render_template('admin.html', message="UserName or Password is incorrect")
    else:
        return render_template('adminblog.html')


@app.route('/addbooks.html')
def addbook1():
    return render_template('addbooks1.html')




@app.route('/addbooks.html',methods=['POST'])
def addbook2():
    add_bookid=request.form.get('bid')
    add_booktitle=request.form.get('title')
    add_bookauthor=request.form.get('author')
    add_bookqty=request.form.get('qty')
    cursor.execute("SELECT * FROM addbooks WHERE Title='"+add_booktitle+"' and Author='"+add_bookauthor+"'")
```

```
        myresult=cursor.fetchone()
        if myresult is None:
                cursor.execute("INSERT INTO addbooks (ISBN,Title,Author,Quantity) VALUES(%s, %s, %s, %s)",(add_bookid, add_booktitle, add_bookauthor, add_bookqty))
                connection.commit()
                return render_template('addbooks1.html',message="Added Book Successfully" , clr="Green")
        else:
                return render_template('addbooks1.html',message="The book you entered is already exist. If u want to update the book Go to Books Update",clr="Red")


@app.route('/booksearch.html')
def booksearch1():
        return render_template('booksearch.html')


@app.route('/booksearch.html',methods=['POST'])
def booksearch2():
        search_bookName=request.form.get('bname')
        cursor.execute("SELECT * FROM addbooks WHERE Title='"+search_bookName+"'")
        myresult=cursor.fetchall()
        if myresult is None:
                return render_template('booksearch.html',message="book is NOT available",output_data=myresult)
        else:
                return render_template('bksearch.html',output_data=myresult)
```

```
@app.route('/booksupdate.html')
def booksupdate1():
    return render_template('booksupdate.html')




@app.route('/bkupdate',methods=['POST'] )
def booksupdate2():
    update_bookname=request.form['bkname']
    update_author=request.form['bkauthor']
    cursor.execute("SELECT * FROM addbooks WHERE Title='"+update_bookname+"' and
Author='"+update_author+"'")
    myresult=cursor.fetchone()
    if myresult is None:
        return render_template('booksupdate.html', message='No Books
available',clr='Red')
    else:
        return render_template('bkupdate.html',output_data=myresult)


@app.route('/bk_edit',methods=['POST'])
def bkedit():
    edit_isbn=request.form.get('isbn_no')
    edit_title=request.form.get('book_name')
    edit_author=request.form.get('book_author')
    edit_qty=request.form.get('ed_qty')
    edit_id=request.form.get('idd')
```

```
        return render_template('addbooks2.html',isbn=edit_isbn, title=edit_title,
author=edit_author,edit_qty=edit_qty, edit_id=edit_id)



@app.route('/bk_update',methods=['POST'] )

def booksupdate3():

        update_isbn=request.form.get('bid')

        update_title=request.form.get('title')

        update_author=request.form.get('author')

        update_qty=request.form.get('qty')

        update_id=request.form.get('id')

        cursor.execute("UPDATE addbooks SET ISBN=%s, Title=%s, Author=%s,Quantity=%s WHERE
ID=%s", (update_isbn,update_title,update_author,update_qty,update_id))

        connection.commit()

        return render_template('booksupdate.html',message=update_id,clr='Green')



@app.route('/bk_delete',methods=['POST'] )

def booksdelete():

        edit_id=request.form.get('idd')

        cursor.execute("DELETE FROM addbooks WHERE ID=%s", (edit_id,))

        connection.commit()

        return render_template('booksupdate.html',message="deleted successfully",clr='Green')



@app.route('/student.html')

def student():

        return render_template('student.html')
```

```
@app.route('/student.html', methods=['POST'])
def student1():
    stu_login_username=request.form.get('uname')
    stu_login_password=request.form.get('pwd')
    cursor.execute("SELECT * FROM student WHERE Email_ID='"+stu_login_username+"' and Password='"+stu_login_password+"'")
    myresult=cursor.fetchone()
    if myresult is None:
        return render_template('student.html', message="UserName or Password is incorrect")
    else:
        return render_template('sturegister.html',stuprof=myresult)


@app.route('/student_register.html')
def student_register():
    return render_template('student_register.html')


@app.route('/student_register.html', methods=['POST'])
def student_details():
    stu_name=request.form.get('stu_name')
    stu_rollno=request.form.get('rno')
    stu_dept=request.form.get('dept')
    stu_year=request.form.get('optradio')
```

```
        stu_emailid=request.form.get('email')
        stu_address=request.form.get('address')
        stu_password=request.form.get('pwd')
        cursor.execute("INSERT INTO student (Name , Roll_No , Dept , Year , Email_ID , Address ,Password ) VALUES(%s, %s, %s, %s, %s, %s, %s)",(stu_name, stu_rollno, stu_dept, stu_year, stu_emailid, stu_address, stu_password))
        connection.commit()
        return render_template('student_register.html', message="Registered Successfully")



@app.route('/sturegister.html')
def student_login():
        return render_template('sturegister.html')



@app.route('/stuprofile.html/<name>')
def student_profile(name):
        cursor.execute("SELECT * FROM student WHERE Email_ID='"+name+"'")
        myresult=cursor.fetchone()
        return render_template('stuprofile.html',stuprof=myresult)

@app.route('/stubooksearch.html/<name>')
def studentbksearch(name):
        cursor.execute("SELECT * FROM student WHERE Email_ID='"+name+"'")
        myresult=cursor.fetchone()
        cursor.execute("SELECT * FROM addbooks")
        myresult1=cursor.fetchall()
        return render_template('stubooksearch.html',stuprof=myresult, booklist=myresult1)
```

```
@app.route('/stubooksearch.html/<name>',methods=['POST'])
def stubooksearch2(name):
        search_bookName=request.form.get('bname')
        cursor.execute("SELECT * FROM addbooks WHERE Title='"+search_bookName+"'")
        myresult=cursor.fetchall()
        cursor.execute("SELECT * FROM student WHERE Email_ID ='"+name+"'")
        myresultprof=cursor.fetchone()
        if myresult is None:
                return render_template('stubooksearch.html',message="book is NOT
available",stuprof=myresultprof)
        else:
                return
render_template('stubksearch.html',output_data=myresult,stuprof=myresultprof)

@app.route('/bookorder.html/<name>')
def stubookorder(name):
        cursor.execute("SELECT * FROM student WHERE Email_ID='"+name+"'")
        myresult=cursor.fetchone()
        return render_template('bookorder.html',stuprof=myresult)

@app.route('/bookorder.html/<name>',methods=['POST'] )
def booksorder(name):
        cursor.execute("SELECT * FROM student WHERE Email_ID='"+name+"'")
        myresultprof=cursor.fetchone()
        update_bookname=request.form.get('bkname')
        update_author=request.form.get('bkauthor')
```

```
        cursor.execute("SELECT * FROM addbooks WHERE Title='"+update_bookname+"' and
Author='"+update_author+"'")

        myresult=cursor.fetchone()

        up_bkqty=myresult[4]

        if myresult is None or  myresult[3]<1:

                return render_template('bookorder.html', message='No Books
available',clr='Red',stuprof=myresultprof)

        else:

                currentdate=date.today()

                returndate=date.today()+timedelta(10)

                cursor.execute("UPDATE addbooks SET Quantity=%s WHERE Title=%s and
Author=%s", ((up_bkqty-1),update_bookname,update_author))

                connection.commit()

                cursor.execute("INSERT INTO bookorder (ISBN, Title, Author, Quantity, Name,
Roll_No , Dept , Year , Email_ID , Address, Password, Taken_Date, Return_Date, ) VALUES(%s, %s, %s,
%s, %s, %s, %s, %s,%s, %s, %s, %s,
%s)",(myresult[1],myresult[2],myresult[3],myresult[4],myresultprof[1],myresultprof[2],myresultprof
[3],myresultprof[4],myresultprof[5],myresultprof[6],myresultprof[7],currentdate,returndate))

                connection.commit()

                return render_template('bookorder.html',message='Ordered
Successfully',clr='Green',stuprof=myresultprof)
```

```
@app.route('/dateinfo.html')
def dateinfo():
        return render_template('dateinfo.html')



@app.route('/dateinfo.html',methods=['POST'])
def dateinfobksearch():
        search_bookName=request.form['bname']
        search_author=request.form['bauthor']
        cursor.execute("SELECT * FROM bookorder WHERE Title='"+search_bookName+"'and
Auhtor='"+search_author+"'")
        myresult=cursor.fetchall()
        if myresult is None:
                return render_template('dateinfo.html',message="book is NOT taken")
        else:
                return render_template('dateinfoshow.html',output_data=myresult)

"""



if __name__ == '__main__':
  app.run(debug=True)
```