**BMS COLLEGE OF ENGINEERING**

**(Autonomous College under VTU)**

**Bull Temple Road, Basavanagudi, Bangalore – 560019**



**An AAT report on**

# “LIBRARY MANAGEMENT SYSTEM USING PYTHON”

Submitted in partial fulfilment of the requirements for the award of degree

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION SCIENCE AND ENGINEERING**

By

KHATEEJA BEGUM A(1BM20IS407)

VINOD J NAVALUR(1BM20IS414)

Under the guidance of

Rashmi R, Assistant Professor

**Department of Information Science and Engineering**

**Abstract**

Library Management System is a system which maintains the information about the books present in the library, their authors, the members of library to whom books are issued, library staff and all.

This is very difficult to organize manually. Maintenance of all this information manually is a very complex task. Owing to the advancement of technology, organization of an Online Library becomes much simple.

The Library Management has been designed to computerize and automate the operations performed over the information about the members, book issues and returns and all other operations.

This computerization of library helps in many instances of its maintenances. It reduces the workload of management as most of the manual work done is reduced

**Table of Contents**

|  |  |
| --- | --- |
| TOPICS | PAGE NO |
| 1 Introduction | 4 |
| 2 Problem Statement | 5 |
| 3 System Requirements | 5 |
| 4 System Design | 6 |
| 5 Implementation | 7 |
| 6 Test Results | 8 |
| 7 Conclusion | 9 |
| 8 References | 9 |

**INTRODUCTION**

Library Management System using python is an application which refers to library systems which are generally small or medium in size. It is used by librarian to manage the library using a computerized system where he/she can add new books, videos and Page sources.

Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detaile about the books a library contains.

With this computerized system there will be no loss of book record or member record which generally happens when a non-computerized system is used.

All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

* Librarians can be given their authorized login ID and password without which the system cannot be accessed.
* Students can only access limited features, i.e., public access level features which include searching a book and student registration form.
* After logging in librarians can search for a specific book, book issue or student from the home panel.
* Librarians need to make an entry for new books. To automate the process, they simply need to enter the number of issues, then the Issue ID for each book issue is generated automatically.
* Another responsibility of a librarian is to approve students in situations where approval is needed, i.e. where documents are to be verified or some manual work. Librarians have a panel to simply approve / reject students and to view all approved students.

**PROBLEM STATEMENT**

Libraries are used to store books, but require a system to navigate to a specific book or specific content within a book. The problem faced is that library users require an efficient method to find a specific book or keyword within a book given a continuously expanding library.

**SYSTEM REQUIREMENT**

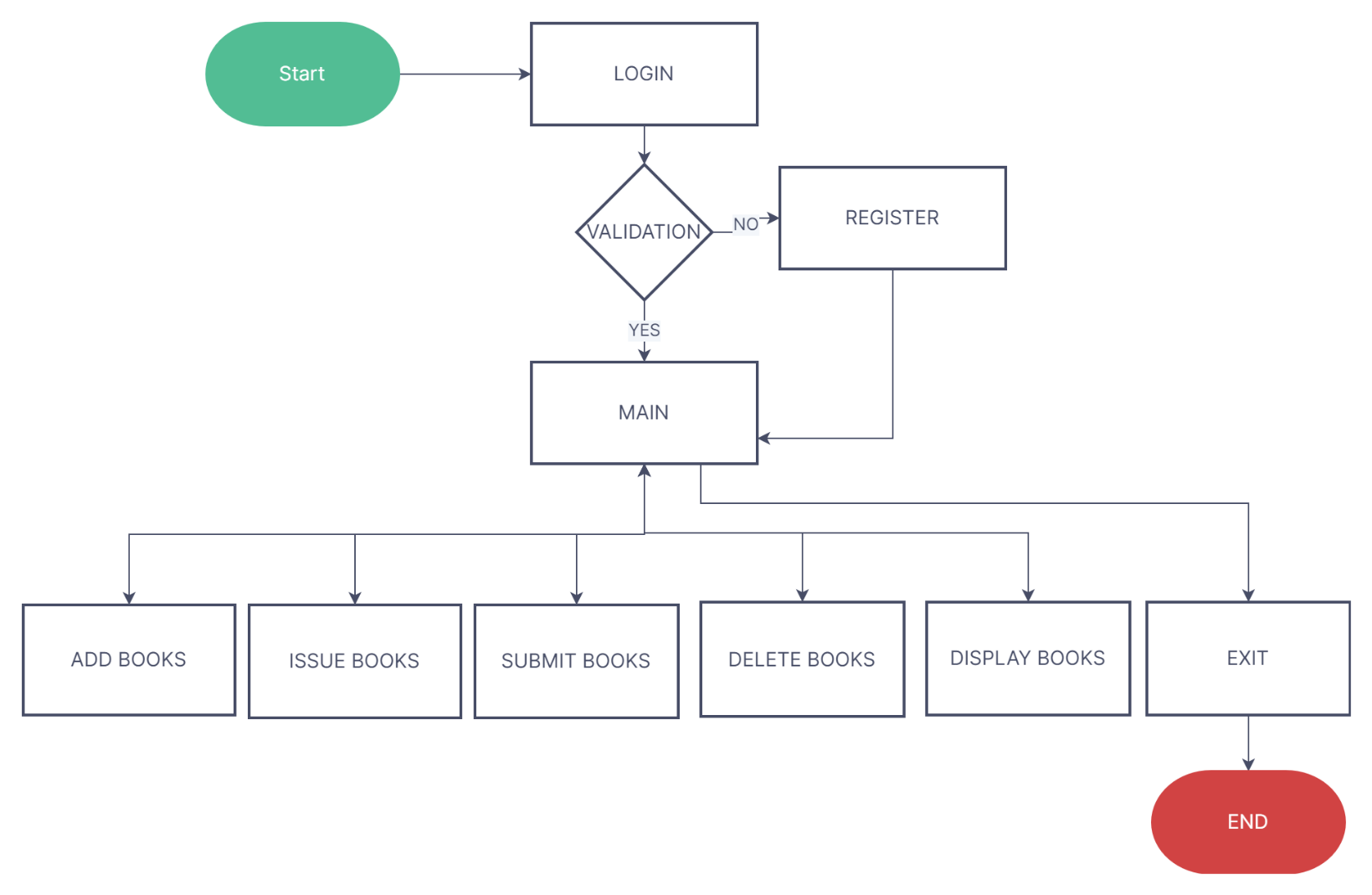
## Hardware Requirement

* i3 Processor Based Computer or higher
* Memory: 1 GB RAM
* Hard Drive: 50 GB
* Monitor

## Software Requirement

* Windows 7 or higher
* Python

**SYSTEM DESIGN / FLOW DIAGRAM**



**IMPLEMENTATION**

import sqlite3

import pandas as pd

import numpy as np

connection = sqlite3.connect("library.db")

c = connection.cursor()

#c.execute("""create table books (bname text,bcode int,total int,subject text)""")

#c.execute("""create table issue (name text,regno text,bcode int,issue date)""")

#c.execute("""create table submit (name text,regno int,bcode int,submit date)""")

#c.execute("""create table user (name text,regno int,password text)""")

def addbook():

    bn = input("enter Book name : ")

    co = int(input("enter book code :"))

    t = int(input("total books :"))

    s = input("enter subject : ")

    data = (bn,co,t,s)

    c = connection.cursor()

    #c.execute("select bcode from books where bcode = ?",(c,))

    c.execute("select \* from books where bcode = ?",(co,))

    if c.fetchone() :

        print("\n")

        print("book already exist\n")

        bookup(co,t)

    else:

        c.execute("insert into books values (?,?,?,?)",data)

        connection.commit()

        print("Insertion successful !! \n")

    main()

def issueb():

    n = input("enter student name :")

    r = input("enter reg no :")

    co = input("enter book code :")

    d = input("enter date :")

    data = (n,r,co,d)

    c = connection.cursor()

    c.execute("select \* from books where bcode = ?",(co,))

    if c.fetchone():

        c.execute("insert into issue values (?,?,?,?)",data)

        connection.commit()

        print("\n")

        print("book issues to :",n)

        bookup(co,-1)

    else:

        print("\n")

        print("book doesnt exist\n")

        main()

def submitb():

    n =  input("enter name :")

    r = input("enter reg no :")

    co = input("enter book code :")

    d = input("enter date :")

    data = (n,r,co,d)

    c = connection.cursor()

    c.execute("select \* from books where bcode = ?",(co,))

    if c.fetchone():

        c.execute("insert into submit values (?,?,?,?)",data)

        connection.commit()

        print("Book submitted sucessfully \n")

        bookup(co,1)

    else :

        print("\n")

        print("entered code is wrong\n")

        main()

def bookup(co,u):

    c = connection.cursor()

    c.execute("select total from books where bcode = ?",(co,))

    myresult=c.fetchone()

    t = myresult[0] + u

    c.execute("update books set total = ? where bcode = ?",(t,co))

    connection.commit()

    main()

def dbook():

    ac = input("enter book code :")

    c = connection.cursor()

    c.execute("select \* from books where bcode = ?",(ac,))

    if c.fetchone():

        c.execute("delete from books where bcode = ?",(ac,))

        connection.commit()

        print("\n")

        print("deleted successfully\n")

    else :

        print("\n")

        print("book doesnt exist")

    main()

def dispbook():

    c = connection.cursor()

    c.execute("select \* from books")

    myresult = c.fetchall()

    for i in myresult :

        print("book name :",i[0])

        print("book code :",i[1])

        print("total :",i[2])

    c.execute("select total from books")

    total = c.fetchall()

    total1 = np.array(total)

    total2 = np.sum(total1)

    print("total number of books present are :",total2)

    main()

def main():

    print(" LIBRARY FUNCTION MANAGMENT \n 1.ADD BOOK \n 2.ISSUE BOOK\n 3.SUBMIT BOOK\n 4.DELETE BOOK\n 5.DISPLAY BOOKS\n 6.EXPORT\n 7.LOG OUT")

    print("\n")

    choice = input("enter task to be performed\n")

    if(choice == '1'):

        addbook()

    elif(choice == '2'):

        issueb()

    elif(choice == '3'):

        submitb()

    elif(choice == '4'):

        dbook()

    elif(choice == '5'):

        dispbook()

    elif(choice == '6'):

        export()

    elif(choice == '7'):

        pre()

def export():

    c.connection.cursor()

    c.execute("select \* from books")

    myresult = c.fetchall()

    df = pd.DataFrame(myresult)

    df.to\_csv('file.csv',index = False)

    choice = input("if you want to see content of csv file press 1")

    if(choice == '1'):

        content = pd.read\_csv('file.csv', index\_col= False)

        content.columns = ['NAME','CODE','NUMBER OF BOOKS','SUBJECT']

        print(content)

        main()

    else:

        main()

def pre():

    print(" 1. Login\n 2. Register\n")

    choice = input("enter your choice")

    if(choice == '1'):

        login()

    elif(choice == '2'):

        register()

    else:

        print("wrong choice enetered\n")

        pre()

def login():

    print("enter you regno and password\n")

    reg = input("enetr your register number\n")

    pas = input("enter password\n")

    c = connection.cursor()

    c.execute("select \* from user where regno = ?",(reg,))

    myresult = c.fetchall()

    if myresult:

        for i in myresult:

            if(pas == i[2]):

                main()

            else:

                print("enter correct password\n")

    else:

        print("invalid registre number\n")

def register():

    na = input("enter user name : ")

    reg = input("enter register number :")

    pas = input("enter password :")

    data = (na,reg,pas)

    c = connection.cursor()

    #c.execute("select bcode from books where bcode = ?",(c,))

    c.execute("select \* from user where regno = ?",(reg,))

    if c.fetchone() :

        print("\n")

        print("uset already exist\n")

        login()

    else:

        c.execute("insert into user values (?,?,?)",data)

        connection.commit()

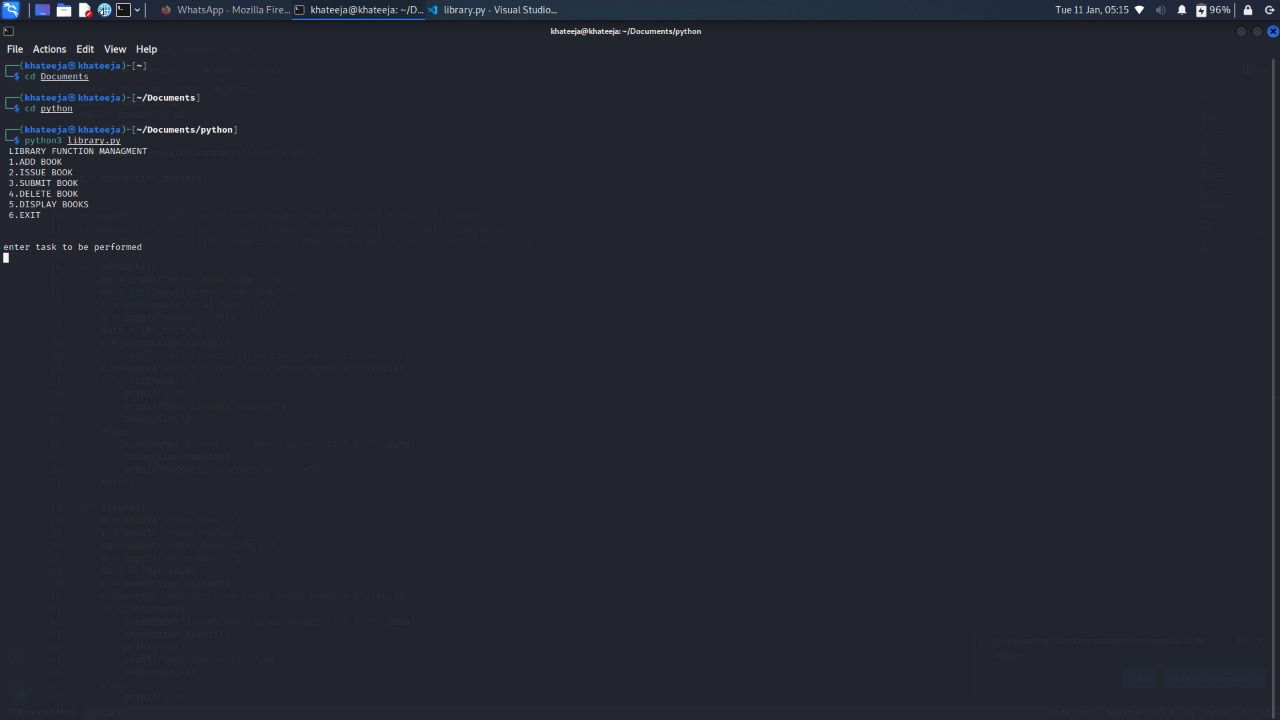
        print("Registration successful !! \n")

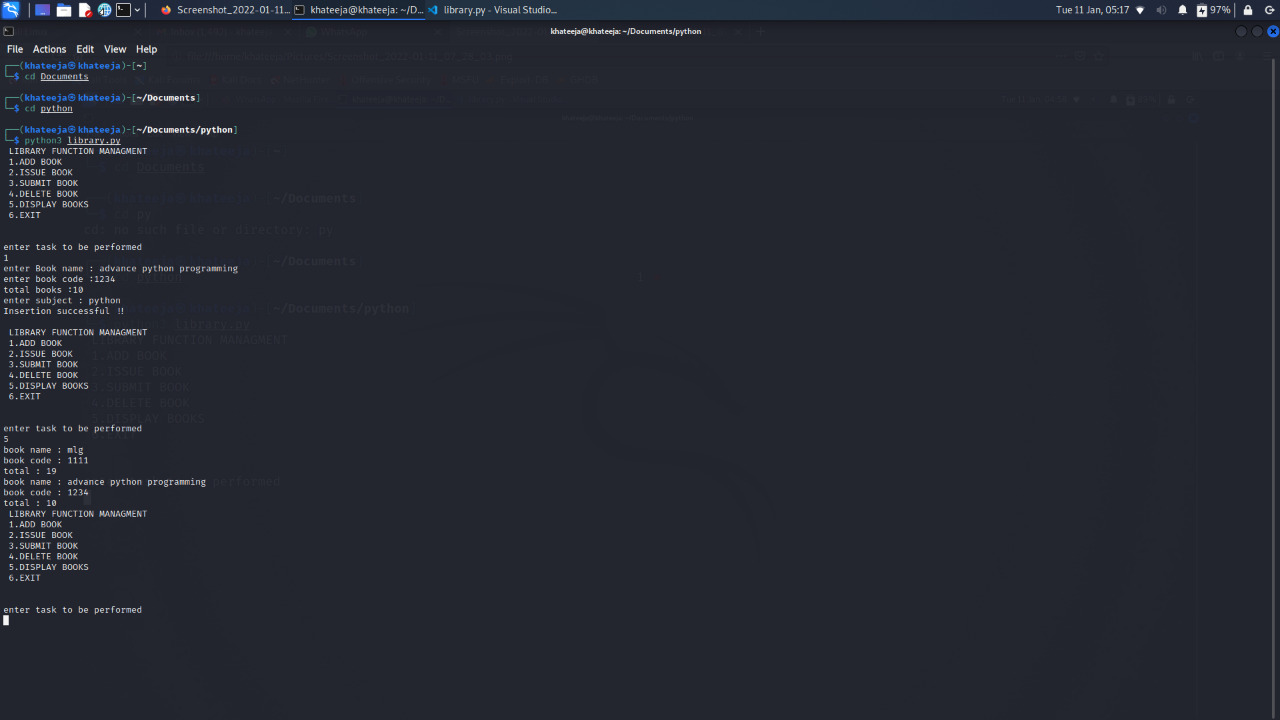
        login()

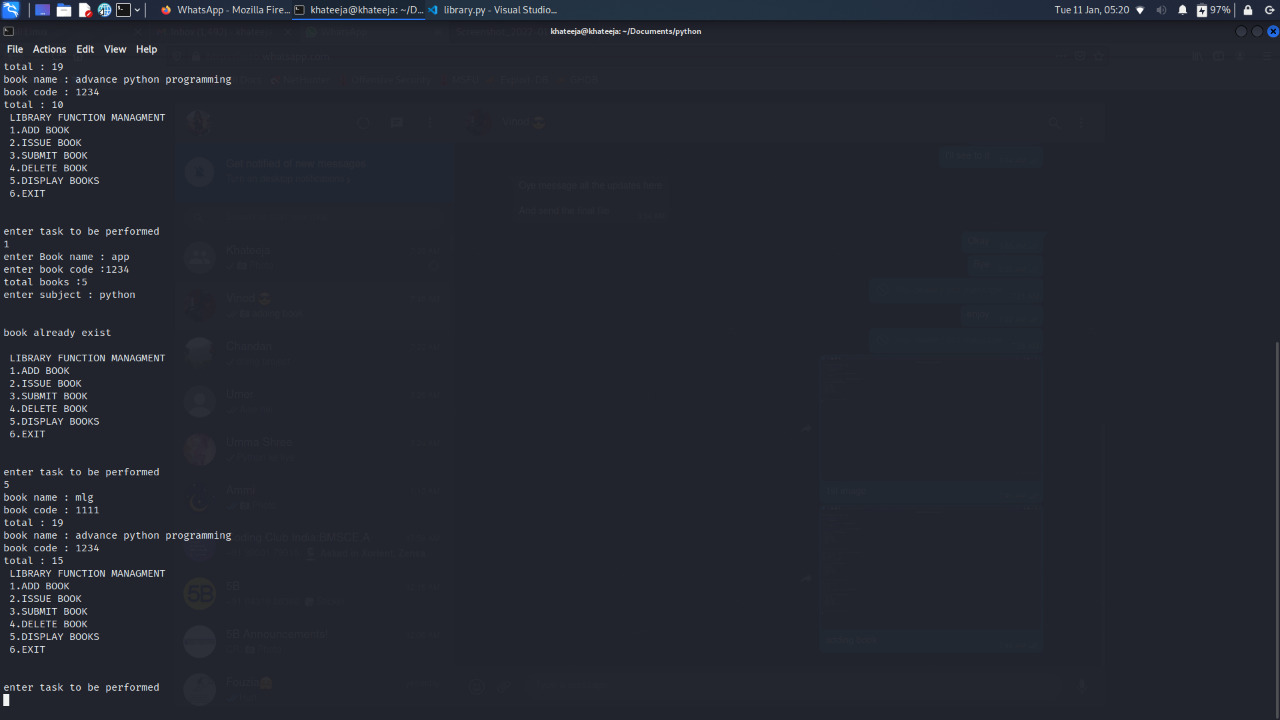
pre()

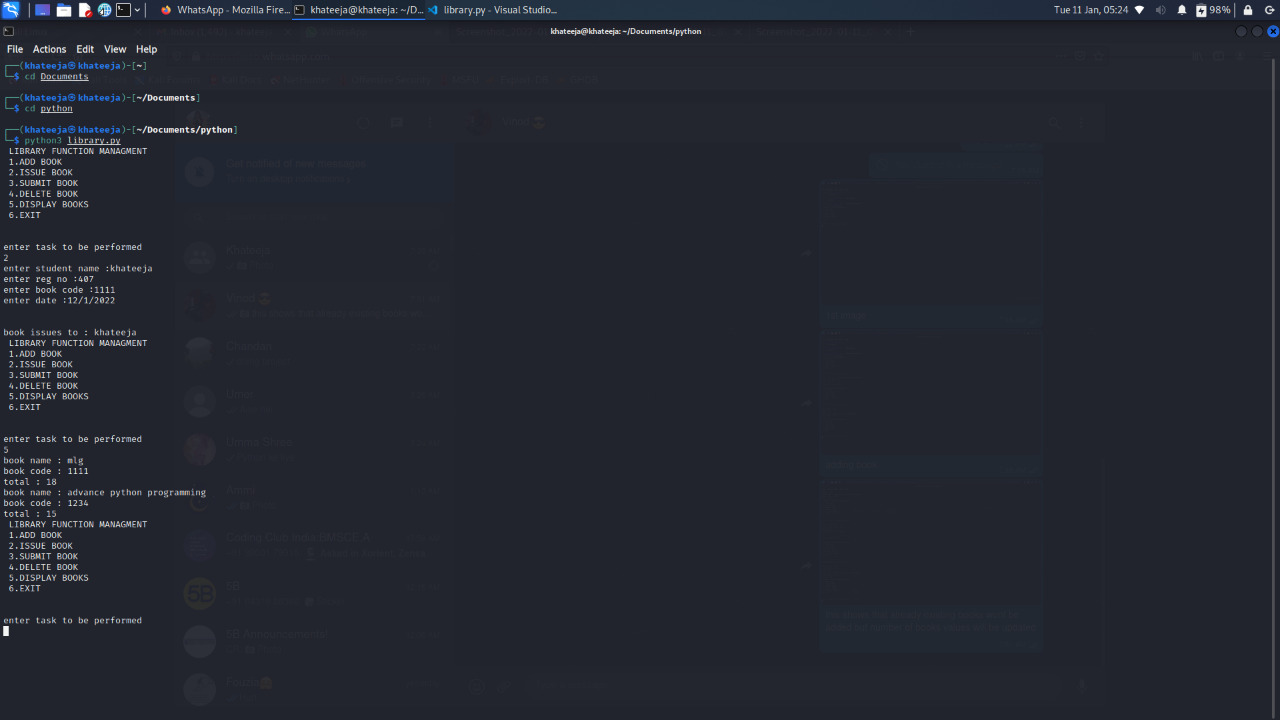
**TEST RESULTS**

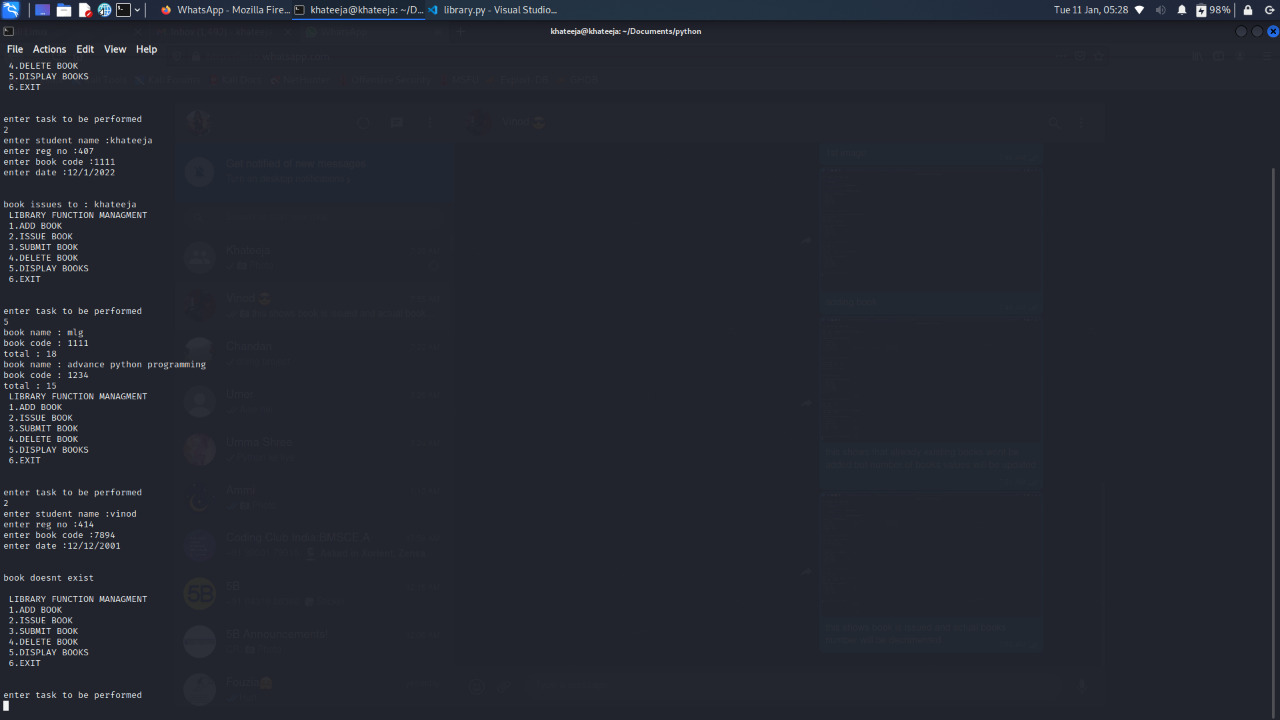
**Functionality of system :**

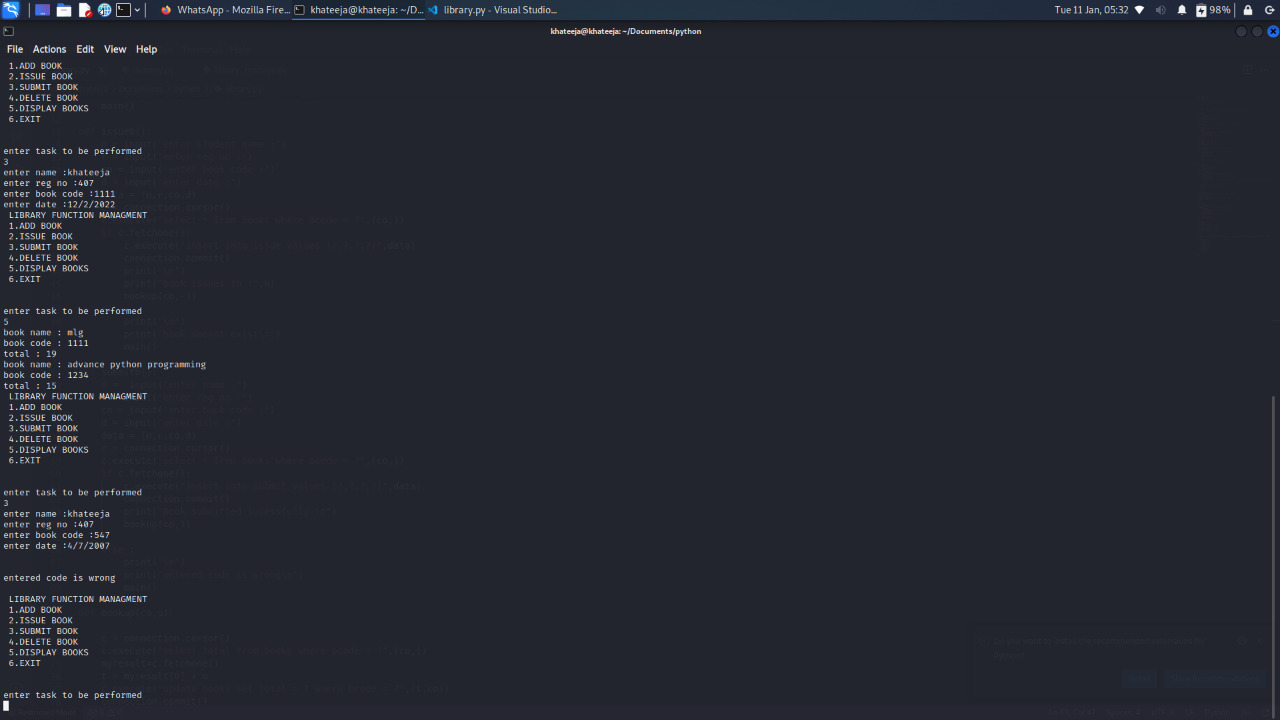


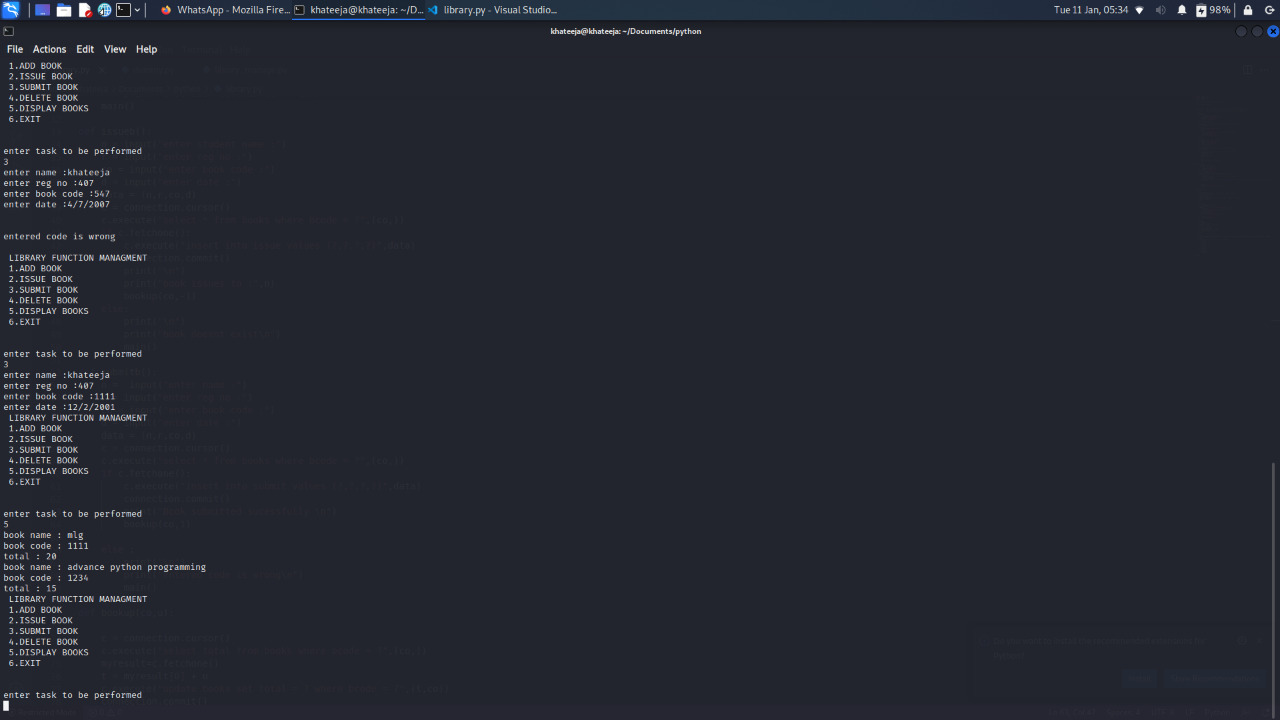


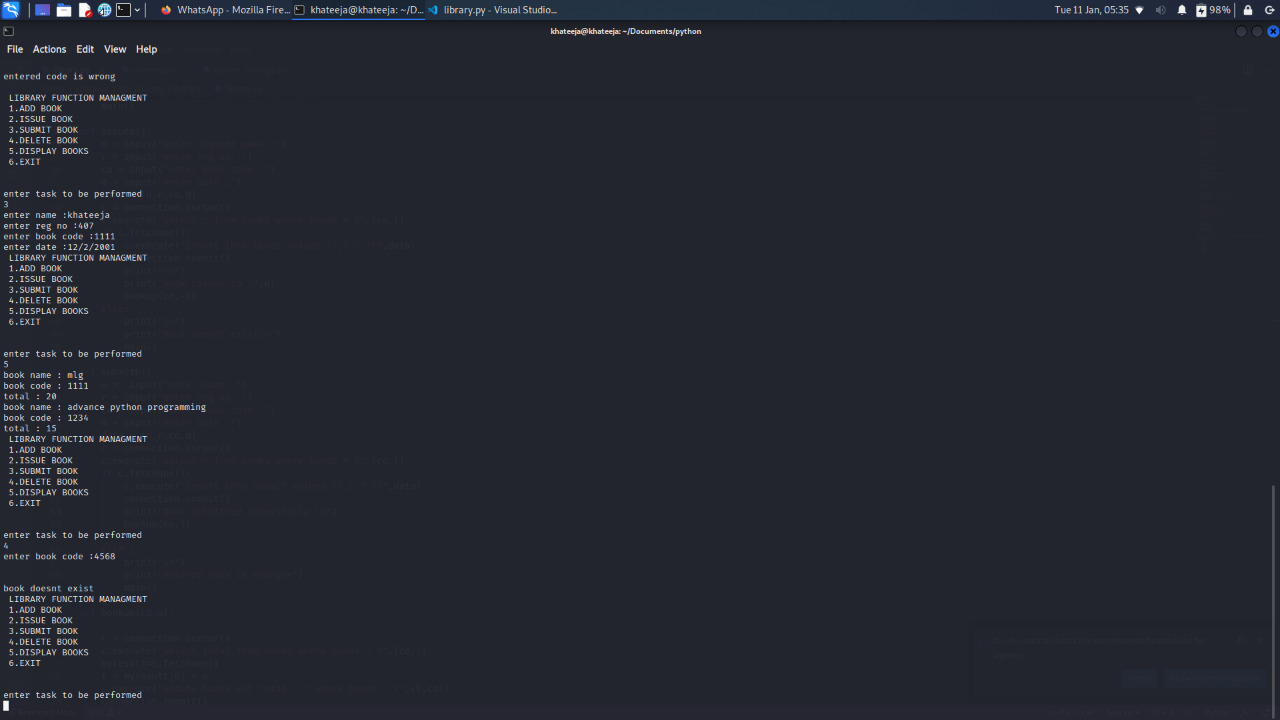


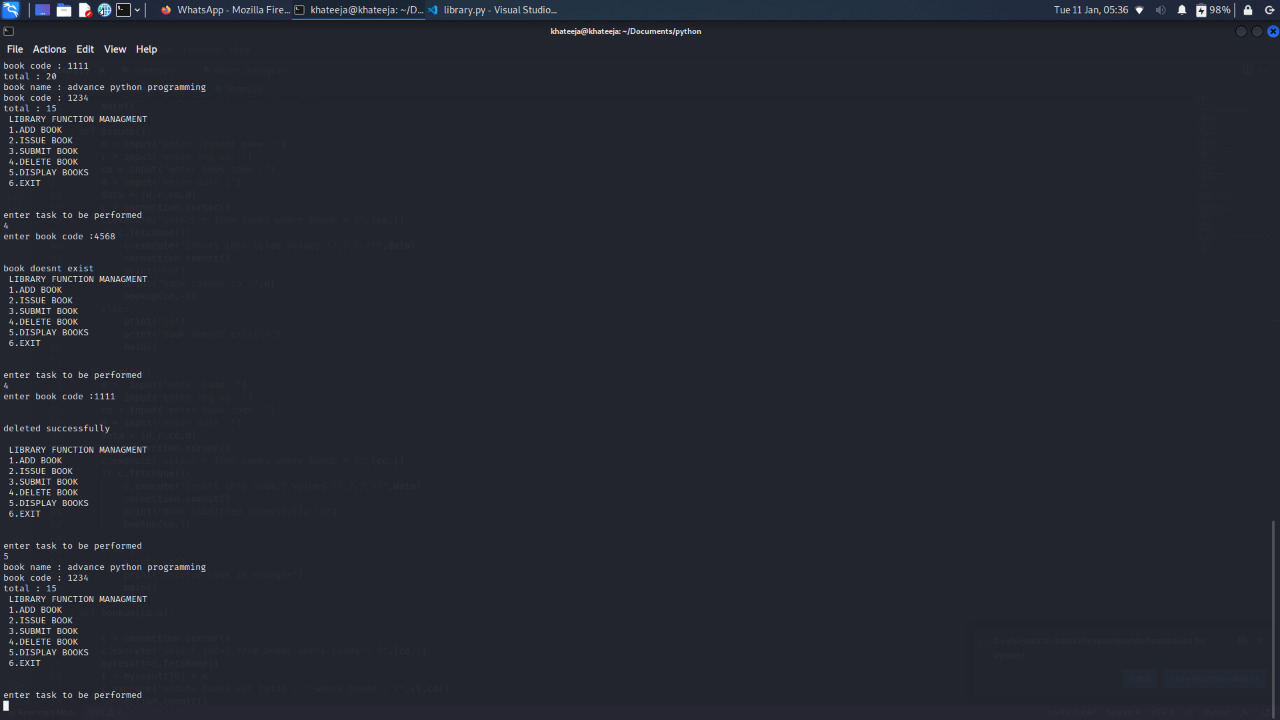












**CONCLUSION**

The “LIBRARY MANAGEMENT SYSTEM” process made computerized to reduce human errors and to increase the efficiency. The main focus of this project is to lessen human efforts.

The maintenance of the records is made efficient, as all the records are stored in the SQLITE database, through which data can be retrieved easily. The navigation control is provided in all the forms to navigate through the large number of records. The editing is also made simpler. The user has to just type in the required field and press the update button to update the desired field

The Books and Students are given a particular unique id no. So that they can be accessed correctly and without errors. Our main aim of the project is to get the correct information about a particular student and books available in the library.

**REFERENCES**

* <https://pandas.pydata.org/docs/user_guide/index.html>
* <https://numpy.org/doc/1.13/genindex.html>
* <https://docs.python.org/3/library/sqlite3.html>