A PROJECT REPORT

ON

BOOK STORE MANAGEMENT SYSTEM

SUBMITTED TO

KIIT (Deemed to be University)

IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD

BACHELOR'S DEGREE IN COMPUTER SCIENCE AND ENGINEERING

By

ADITYA GUPTA - 1705479

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

APRIL 2020

CONTENTS

KIIT (Deemed to be University)

SCHOOL OF COMPUTER ENGINEERING

BHUBANESWAR, ODISHA - 751024

CERTIFICATE

This is to certify that the project entitled

"BOOKSTORE MANAGEMENT SYSTEM"

Submitted by

Aditya Gupta

Roll No. – 1705479

is a record of bonafide work carried out by me, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer Science & Engineering at KIIT (Deemed to be University), Bhubaneswar, Odisha-751024.

This work is done during the academic year 2019-2020. I hereby declare that which is being presented in the project entitled "Bookstore Management System" is an authentic record of my own work.

Date : 25.04.2020

ACKNOWLEDGEMENT

I am profoundly grateful to all my professors and lecturers for their expert guidance and encouragement throughout to see that this project rights its target since its commencement to its completion. The work is a team effort of all the professors and lecturers minus which the completion of this project was not possible. I would also like to thanks my parents and my family for their constant support and ideas throughout the project which helped me to make this project a reality. I would also like to thank my friends who helped me whenever I needed their help for the project.

Aditya Gupta

ABSTRACT

Today it is becoming very difficult to maintain records manually. Software system easily does the job of maintaining daily records as well as the transaction according to the user requirements. Only basic knowledge of computers is required for operations. The software system consists of all information of books and sold to the customer. The proposed system provides lots of facility to the user to get information of the books and it provide information in quick time in a systematic manner. The processing time on the data is very fast. It provides required data quickly to the user and also in specified manner to the user. There is lot of duplicate works, and chance of mistake when the records are changed they need to update each and every excel file. There is no option to find and print previous saved records, there is no security and anybody can access any report and sensitive data. This bookstore management system is used to overcome entire problem which they are facing currently, and making complete automation of manual system to computerized system. Here, I have tried to develop this system which provide the automation to any type of the bookstore. In this system all records are saved in the database for report generation. In present system during issuing order of more stock, the product register is required to check the availability of stock in hand, and it takes time to check records. In each process whether it is product management, maintaining customer records, payment management, report generation, user has to pay attention to a greater extent while performing the tasks. Book shop management system will help the customers query whether a book is in stock, the user can query the availability of a book either by using the book title or by using the book code. Bookstore management software for monitoring and controlling the transactions in a bookshop. Our software is easy to use for both beginners and advanced users. It features a familiar and well thought-out, attractive user interface, combined with strong searching insertion and reporting capabilities.

INTRODUCTION

Bookstore Management software project that acts as a central database containing various books in stock along with their title and cost. This project is a Python Int that acts as a central bookstore. This project is developed using python as the front end and MySQL as a back-end platform. The MySQL database stores various book related details. A user visiting the application can see a wide range of books arranged in various categories. The user may select desired book and view its price. The user may even search for specific books on the system. Once the user selects a book, he then has to fill in a form and the book is booked for the user. Book store project is an application which is implemented in Python platform. It integrates all components related to a book store. The database flexibility, convenient features, maximum increase in customer service and access to accurate information makes this ideal for all sizes of stores.

PRESENT BOOK STORE MANAGEMENT SYSTEM

At present, the Wholesale and Retail outlets are working under manual management. The client uses MS Excel, All records related to Products, Sales, Suppliers, Orders, Payment are stored in excel files. there is lot of duplicate work, and chance of mistake. When the records are changed they need to update each and every excel file. In case of Customer records, all information related to customers and the product which the customer has purchased is to be stored in the Customers excel files. If the changes in the customer profile (like Phone no., Address) occur, excel file must be updated.

To manage the whole data, the person maintaining records has to take great pain. Various excel files has to be maintained for each separate process. There is no option to find previous saved records. There is no security; anybody can access any report and sensitive data. Some of the problems being faced in manual system are as follows:

1) Fast report generation is difficult.

2) Tracing a book is tedious.

3) Information about issue of the books are not properly maintained.

4) No central database can be created as information is not present in database

PROPOSED SYSTEM

This Book Shop Management System is used to overcome the entire problem which they are facing currently, and making manual system to computerized system. It is an automated Bookshop Management System. Through our software user can add members, add books, search books, purchase book in quick time. Our proposed system has the following advantages.

1) It provides "better and efficient" services to members.

2) Reduce the workload of employee.

3) Faster retrieval of information about the desired book.

4) Provide facility for proper monitoring reduce paper work and provide data security.

5) All details will be available on a click.

6) It has more storage capacity and search facility and fast access to database.

All the manual difficulties in managing the Bookshop have been rectified by implementing computerization.

Bookstore management system should update the stock and generate the sales receipt for the book.

BUSINESS GOALS AND RULES

The main objective of this database system is to keep track of stores information and allow access to information at any time. The purpose of this system is to help the Book Store Management System in maintaining information about the book sales by tracking books, book instances information, order information, customer personal information etc. The final result will be better access by maintaining book store information.

BUSINESS RULES

- **Whenever a new order is placed, the book store database system is updated with new order details.**
- **Whenever a particular book is purchased, the book store database system updates book instance, order details, and customer details.**
- **Existing customer's information is updated periodically.**

USER REQUIREMENTS

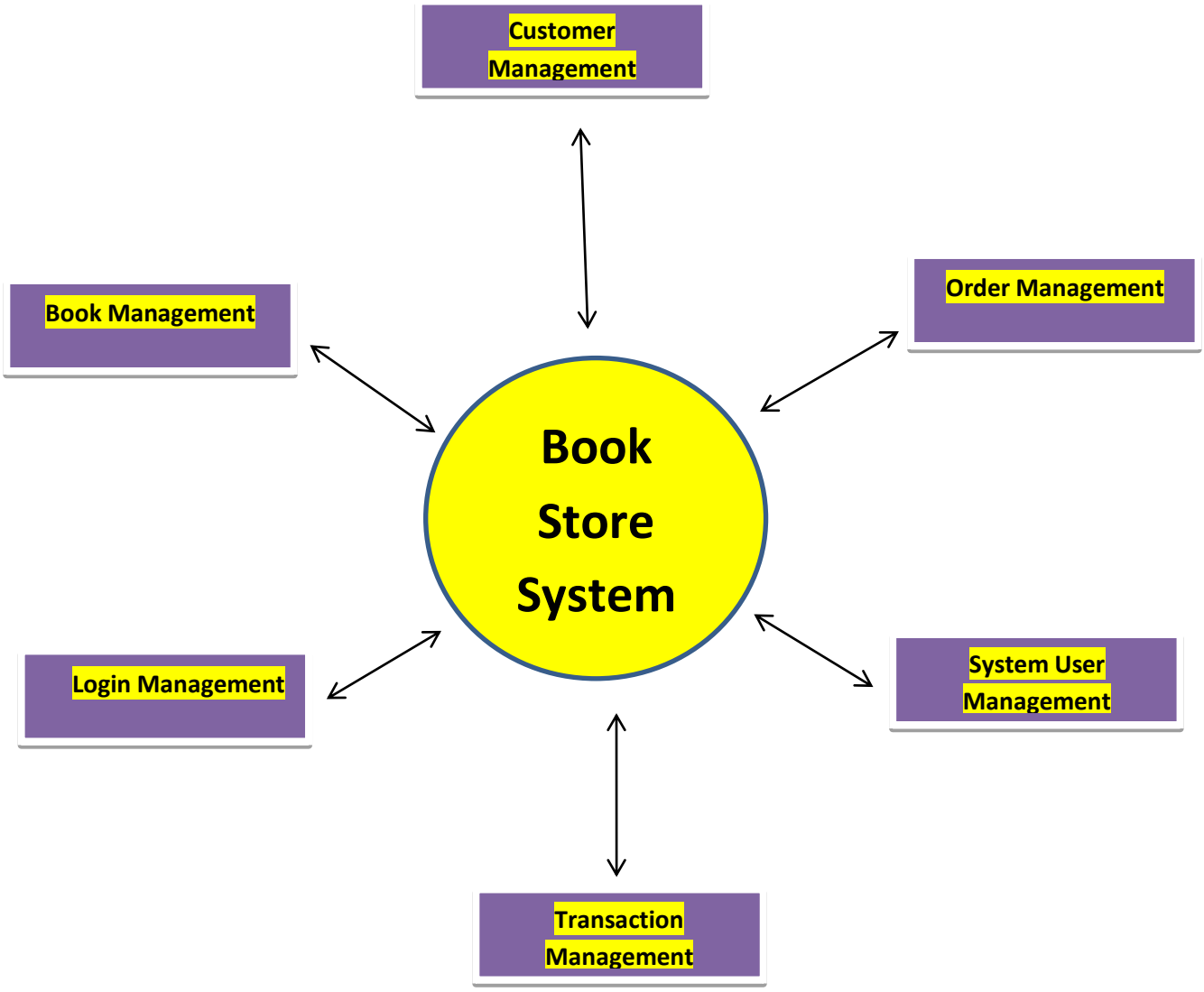
The following are the user requirements

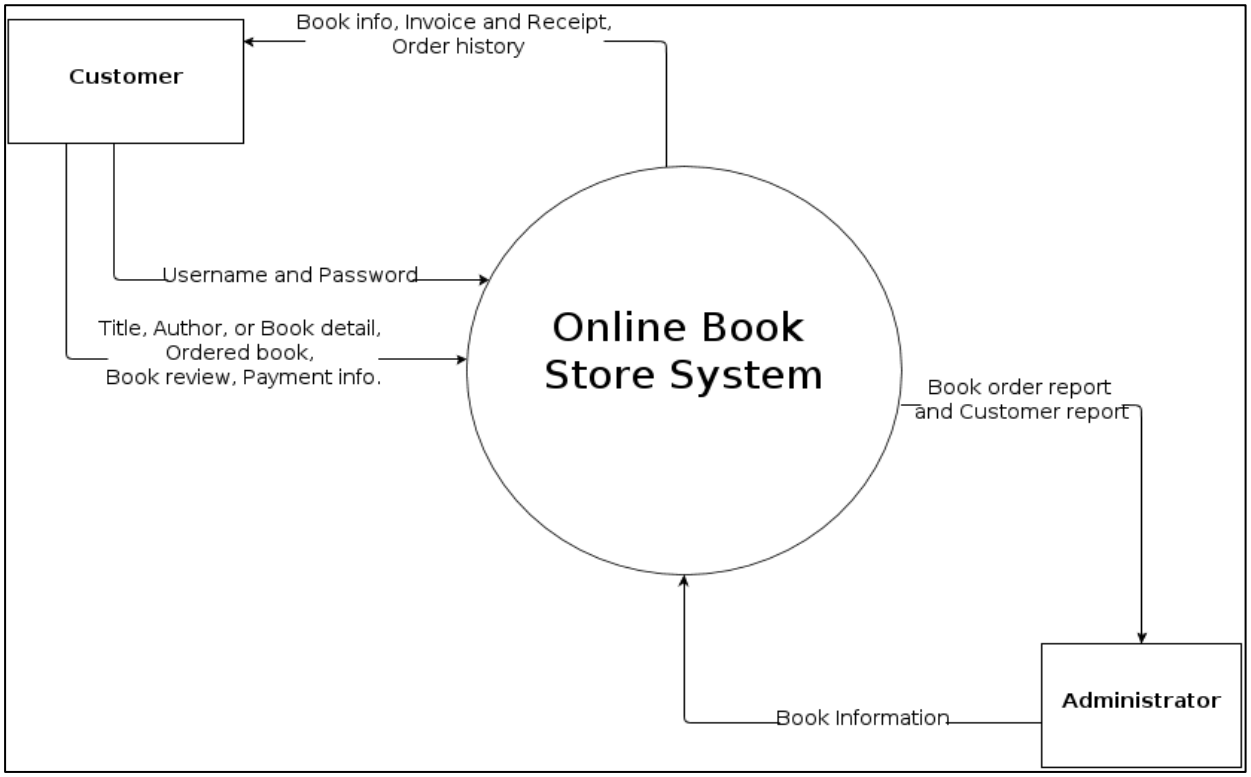
- **Database should be straightforward and simple to utilize.**
- **Access to timely, actionable data every day and every hour**
- **Queries should run speedier.**
- **Database should be backed up regularly.**
- **Database should be secured by a secret password so that unauthorized person shouldn't manipulate or access the data.**
- **Database should be sufficiently adaptable to store enough data in case of expansion.**

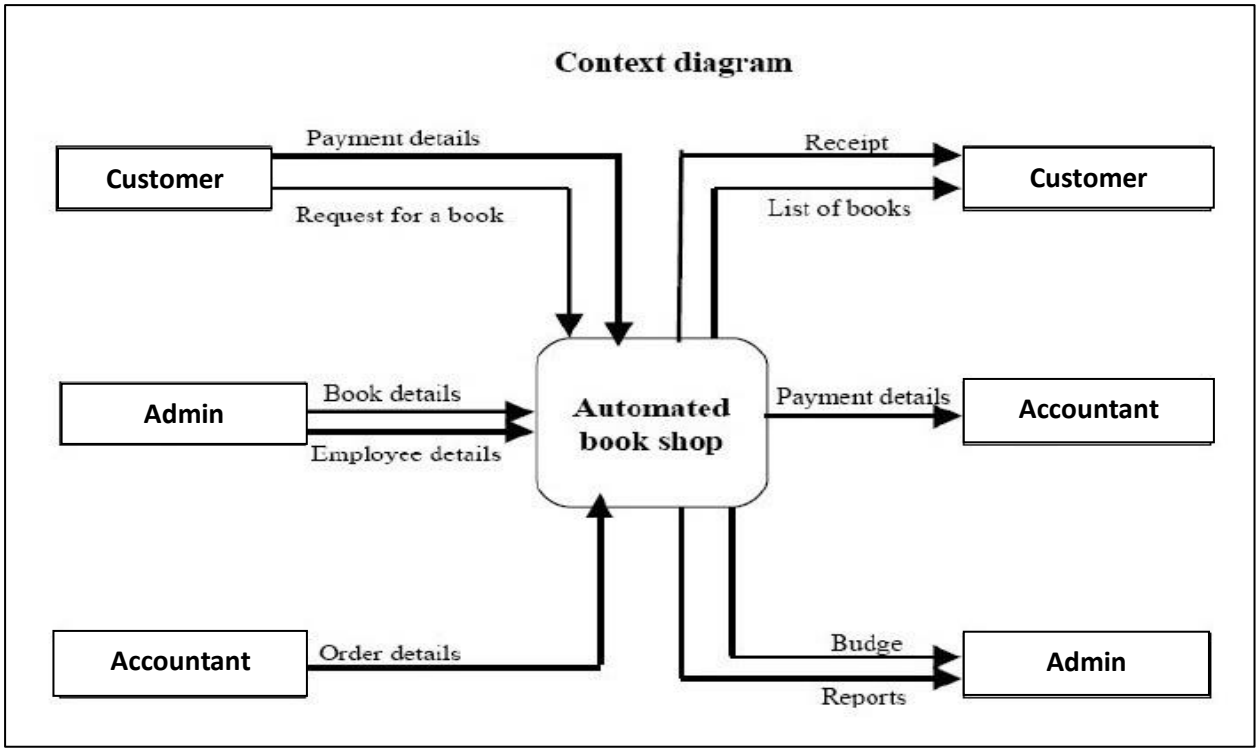
DATA REQUIREMENTS

All previous data which is maintained using excel need to be imported into database. Any new user, personal data needs to be directly entered into the database. The database needs to have 3 separate tables which hold data about Stores Books & Book Instances details, Order details & Order Details information, and Customers personal information.

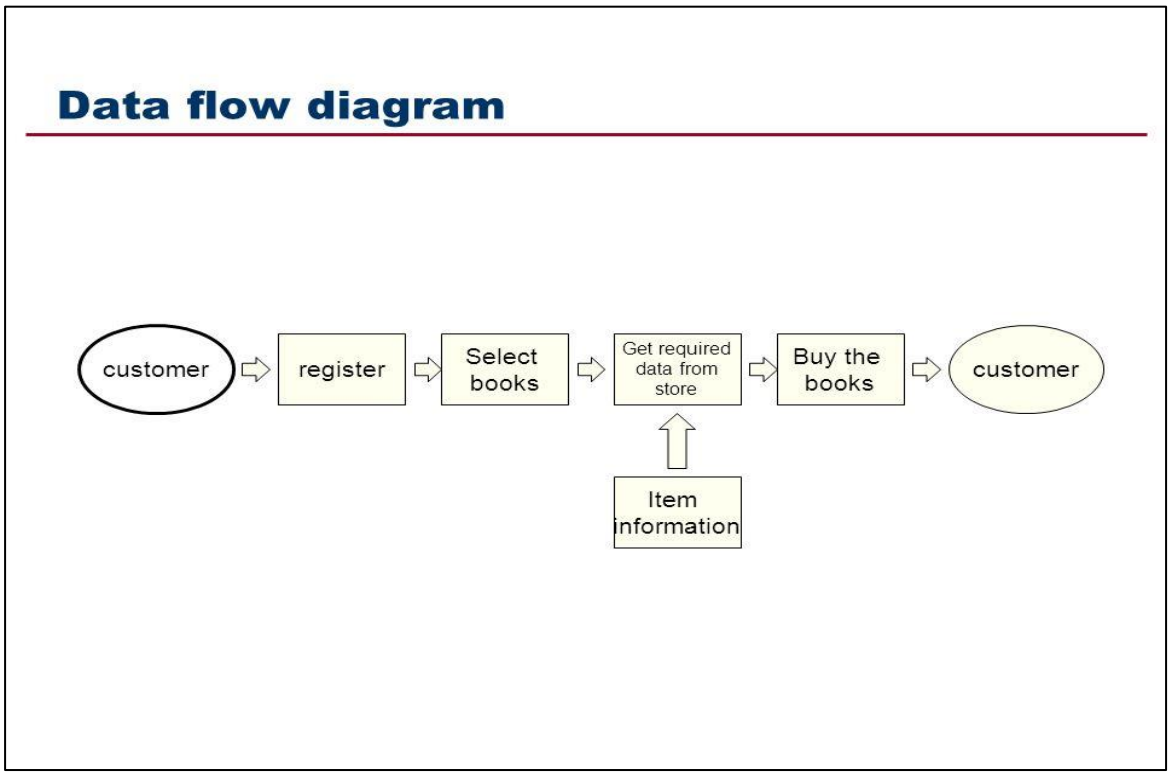
DATA FLOW DIAGRAM OF BOOKSTORE MANAGEMENT DIAGRAM




Customer
Book info, Invoice and Receipt,
Order history
Username and Password
Title, Author, or Book detail,
Ordered book,
Book review, Payment info.
Online Book
Store System
Book order report
and Customer report
Administrator
Book Information


Context diagram
Customer
Payment details
Request for a book
Receipt
List of books
Customer
Admin
Book details
Employee details
Automated
book shop
Payment details
Accountant
Accountant
Order details
Budge
Reports
Admin

ACTIVITY DIAGRAM FOR USERS



BOOKSTORE MANAGEMENT PROJECT DETAILS

FRONT END : PYTHON PROGRAMMING LANGUAGE (Python 3.7)

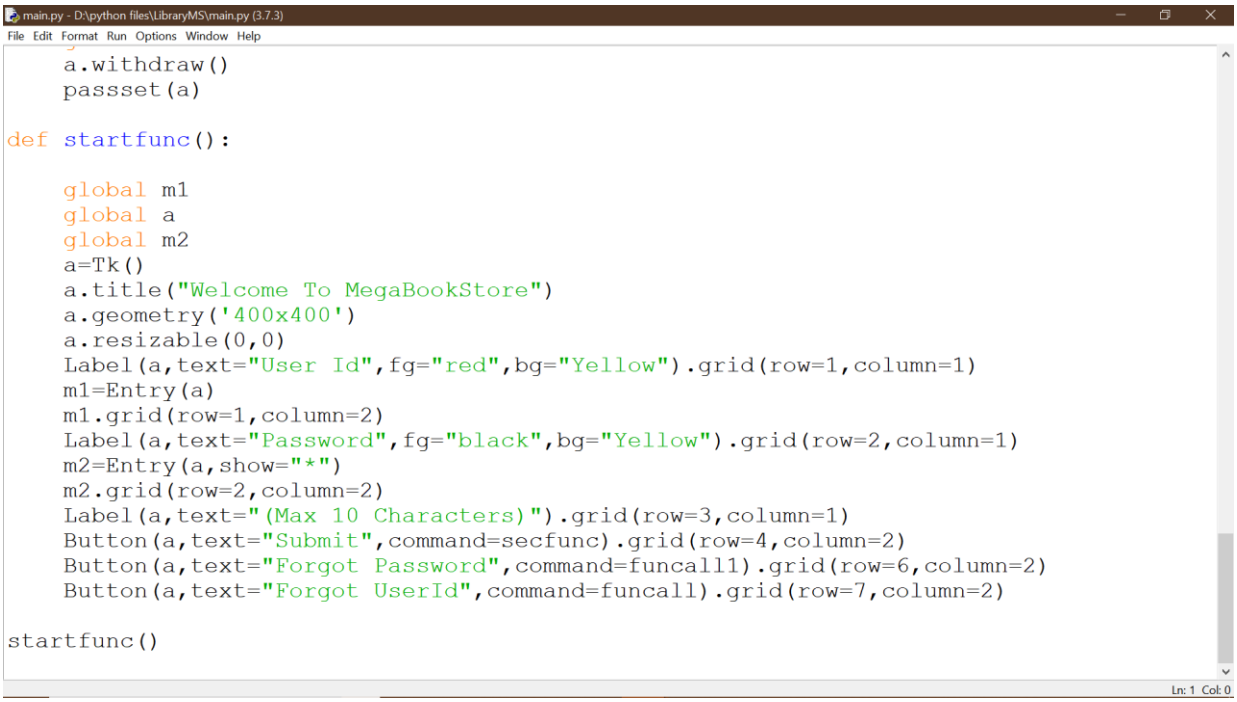
BACK END : MySQL ORACLE SERVER 2015 FOR DATABASE

MODULES OF THE BOOKSTORE MANAGEMENT PROJECT

1) MAIN PAGE MODULE :

It is a main login module where a customer or an admin or the accountant can login into the Mega Book Store application system by entering the valid user id and password.

Once successfully logged in you will be automatically directed to the respective page as per your assigned role and access to the system.



2) USER MODULE :

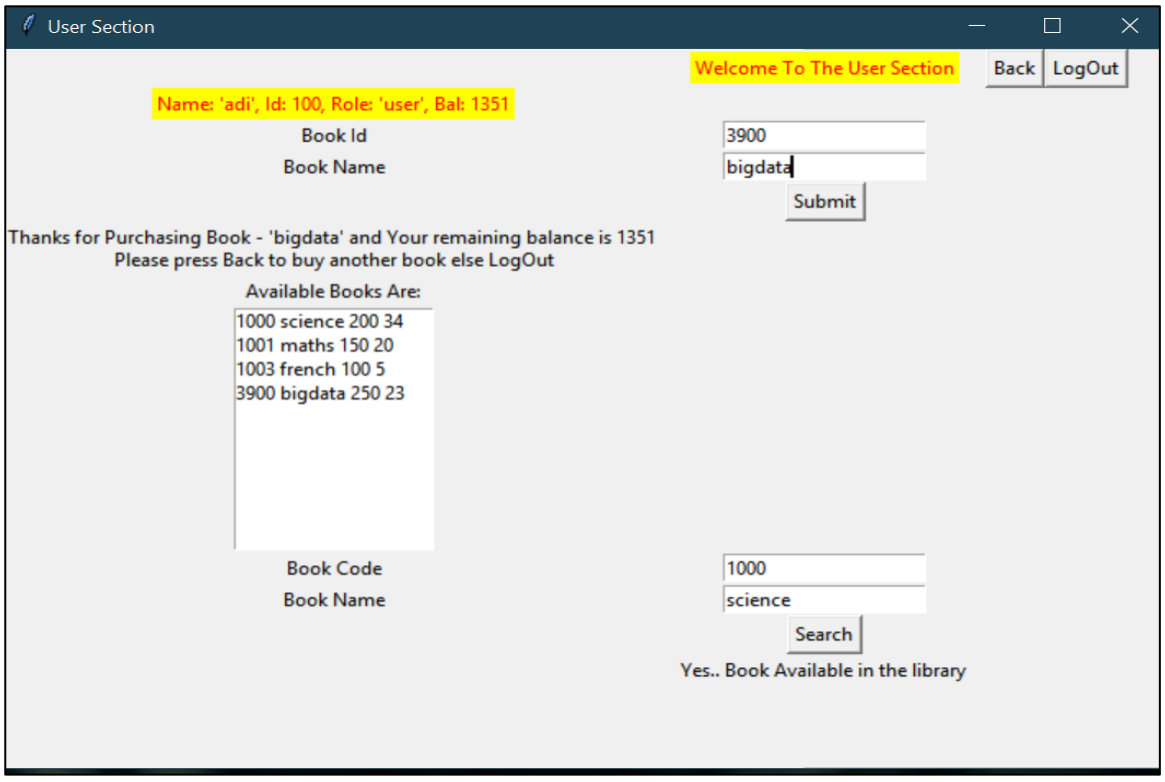
It is a module where the user can login to the system and see the list of books available in the store. The user can search the book from the search option either by entering the Book code, the Book name or by entering both code and the name. User can also buy the book by entering the book id and the book code if he has sufficient balance in his membership account. Once the transaction is successful then the book is reserved for the customer.

user.py - D:\python files\LibraryMS\user.py (3.7.3)

File Edit Format Run Options Window Help

```
import pymysql
import datetime
from mailer import *

from tkinter import *
a2=pymysql.connect("localhost","root","12345","std1")
b1=a2.cursor()

def searchfunc():
    if b1.execute("select * from book where bookname='{}'".format(sname.get())):
            res=b1.fetchall()
            for row in res:
                b_id=row[0]
                b_name=row[1]
                b_price=row[2]
                b_quan=row[3]
            Label(b,text="  Yes.. Book Available in the library   ").grid(row=24,colun

    elif b1.execute("select * from book where bookid={} and bookname='{}'".format(scoc
            res=b1.fetchall()
            for row in res:
                b_id=row[0]
                b_name=row[1]
                b_price=row[2]
                b_quan=row[3]
```

Ln: 1 Col: 0



3) ADMIN MODULE :

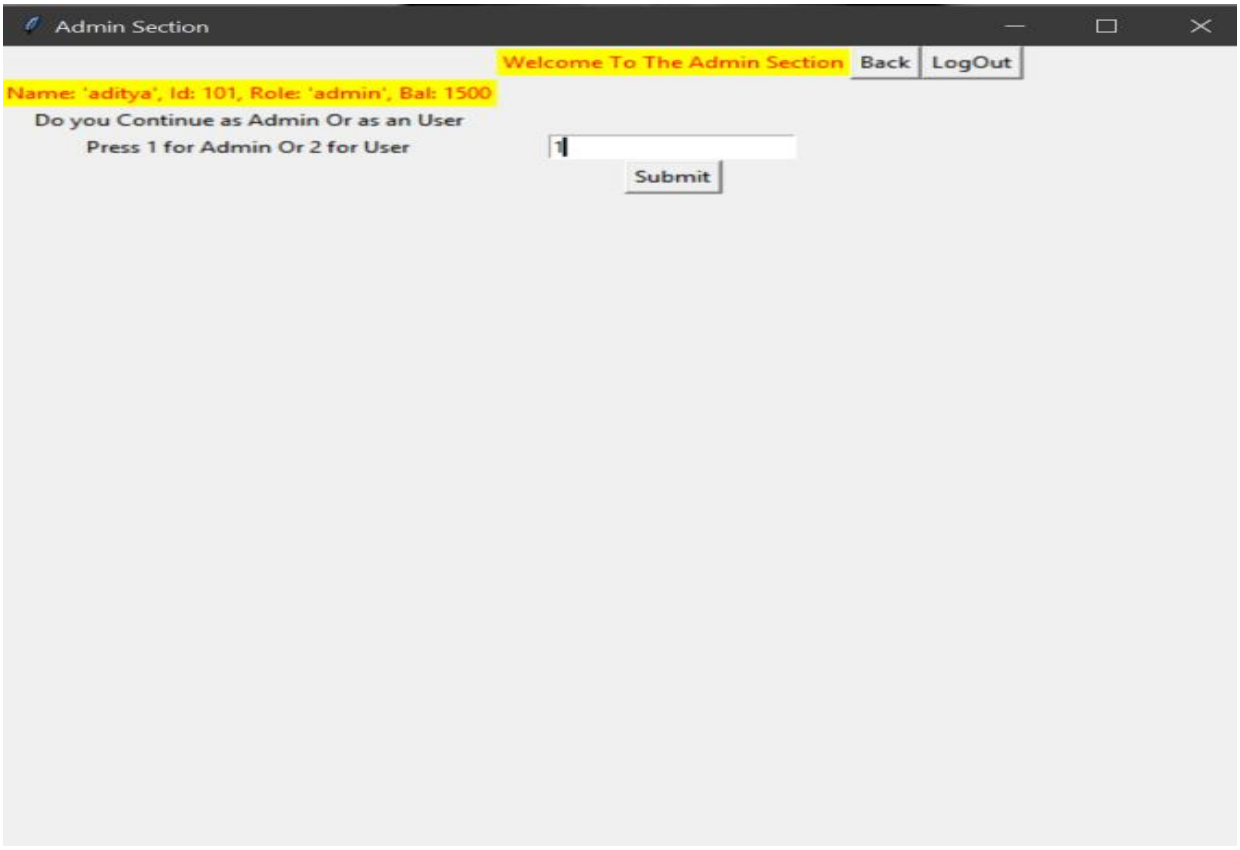
In this module once the admin logs in to the admin section then he is given an option to work as a user or as an admin. If he wants to buy a book then he can choose the user option and then admin will be directed to the user section where a book purchase can be initiated. Else in the Admin section, he/she gets the access to add new books or delete the books available in the store. Admin has the access to add new users in the system and delete users. Transaction List of the past six months can also be viewed by the admin from his account .

admin.py - D:\python files\LibraryMS\admin.py (3.7.3)

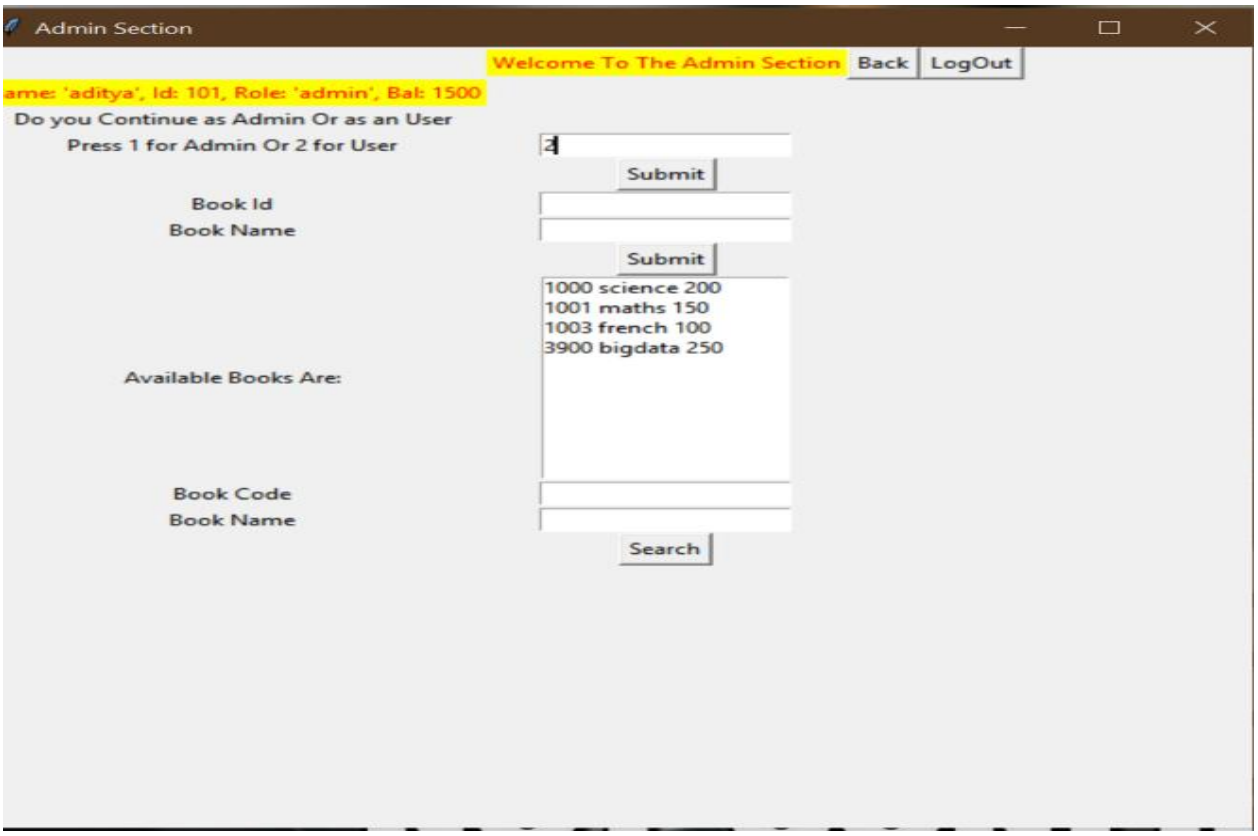
File Edit Format Run Options Window Help

```
def admin(eid,epass,a):

    global c
    global e_pass
    global ename
    global e_id
    global e_phone
    global erole
    global ebal
    c=Toplevel(a)
    c.title("Admin Section")
    c.geometry('625x625')
    c.resizable(1,1)
    Label(c,text="Welcome To The Admin Section",fg="red",bg="Yellow").grid(row=1,colum
    b2.execute("select * from emp where id={} and pass='{}'".format(eid,epass))
    res=b2.fetchall()
    for row in res:
        ename=row[0]
        e_id=row[1]
        e_pass=row[2]
        erole=row[3]
        ebal=row[4]
        e_phone=row[6]
    Label(c,text="Name: '{}', Id: {}, Role: '{}', Bal: {}".format(ename,e_id,erole,eba
    Label(c,text="Do you Continue as Admin Or as an User").grid(row=3,column=1)
```

Ln: 1 Col: 0


Admin Section
Welcome To The Admin Section
Back
LogOut
Name: 'aditya', Id: 101, Role: 'admin', Bal: 1500
Do you Continue as Admin Or as an User
Press 1 for Admin Or 2 for User
1
Submit


Admin Section
Welcome To The Admin Section
Back
LogOut
Name: 'aditya', Id: 101, Role: 'admin', Bal: 1500
Do you Continue as Admin Or as an User
Press 1 for Admin Or 2 for User
1
Submit
1 for Adding Book, 2 for Deleting Book
3 for Adding User, 4 for Deleting User
5 for Transaction List
3
Submit
Id Of User
Name Of User
Role Of User
Contact No.
Pass Of User
Email Of User
Submit



4) ACCOUNTANT MODULE :

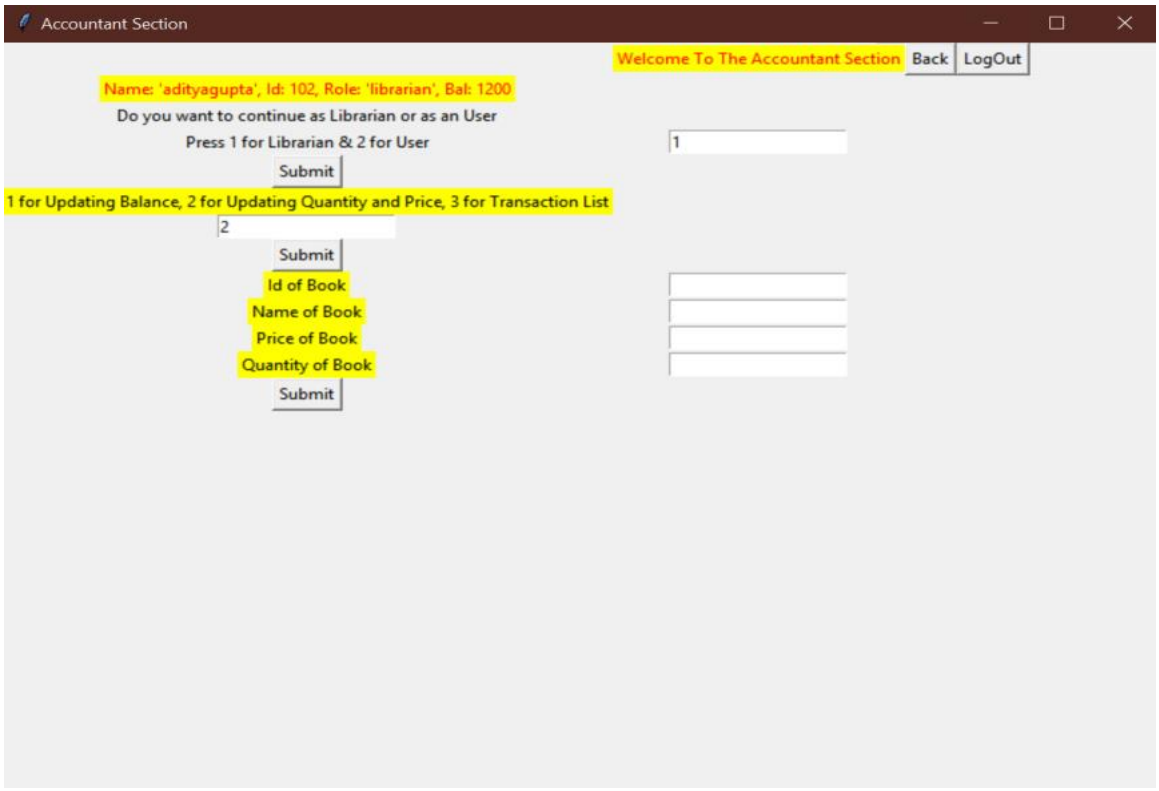
In this module once the person logs in as accountant, the person is given a choice to perform a transaction as a user or perform the duties of a accountant cum librarian where the jobs to be performed are updating the account balance for the users, updating the price and quantity of the stock in the bookstore as well as the person also has the access to see the transaction list of the past 6 months from his account.

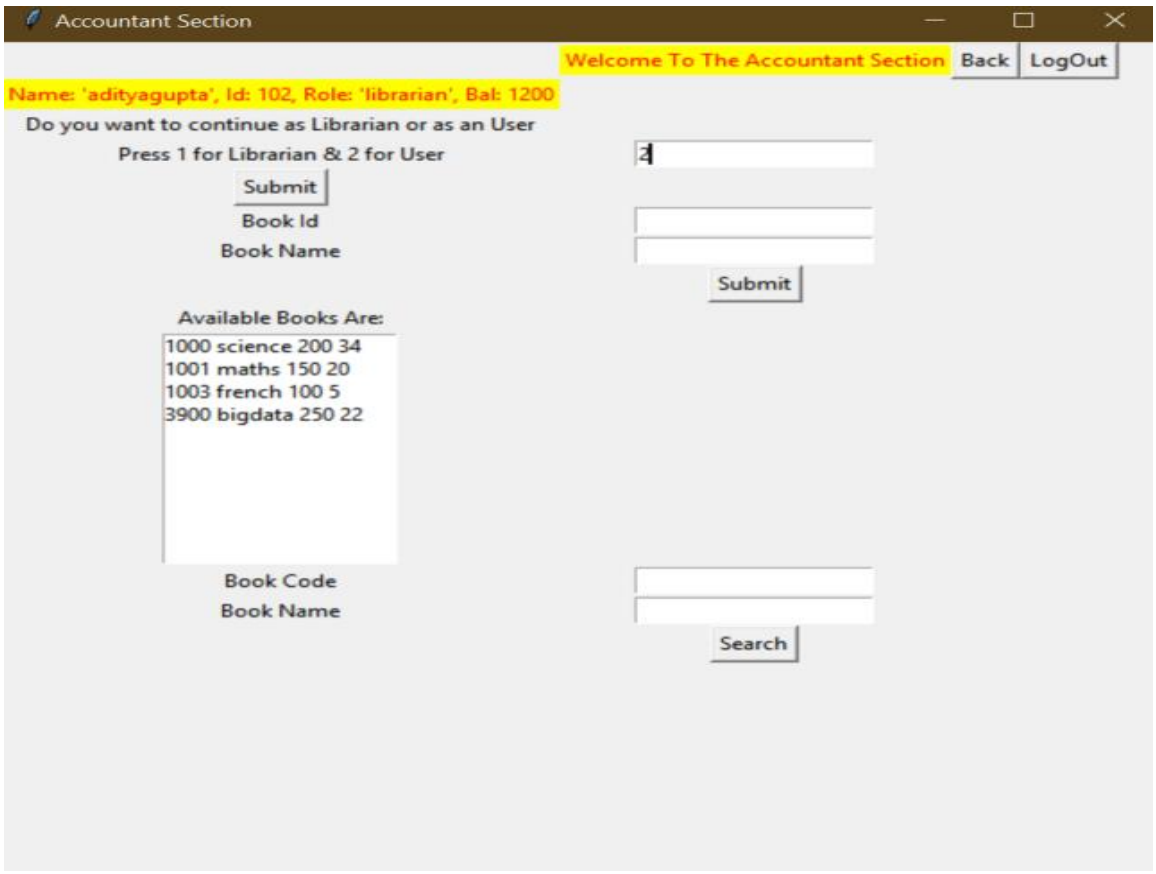
librarian.py - D:\python files\LibraryMS\librarian.py (3.7.3)

File Edit Format Run Options Window Help

```
    global d
    global m1
    global e_id
    global e_name
    global e_pass
    global e_role
    global e_phone
    global e_bal
    d=Toplevel(a)
    d.title("Accountant Section")
    d.geometry('600x600')
    d.resizable(1,1)
    Label(d,text="Welcome To The Accountant Section",fg="red",bg="Yellow").grid(row=1,
    Button(d,text="LogOut",command=exitfunc).grid(row=1,column=7)
    Button(d,text="Back",command=lib).grid(row=1,column=6)
    b3.execute("select * from emp where id={} and pass='{}'".format(eid,epass))
    res=b3.fetchall()
    for row in res:
        e_name=row[0]
        e_id=row[1]
        e_pass=row[2]
        e_role=row[3]
        e_bal=row[4]
        e_phone=row[6]
    Label(d,text="Name: '{}', Id: {}, Role: '{}', Bal: {}".format(e_name,e_id,e_role,e
```

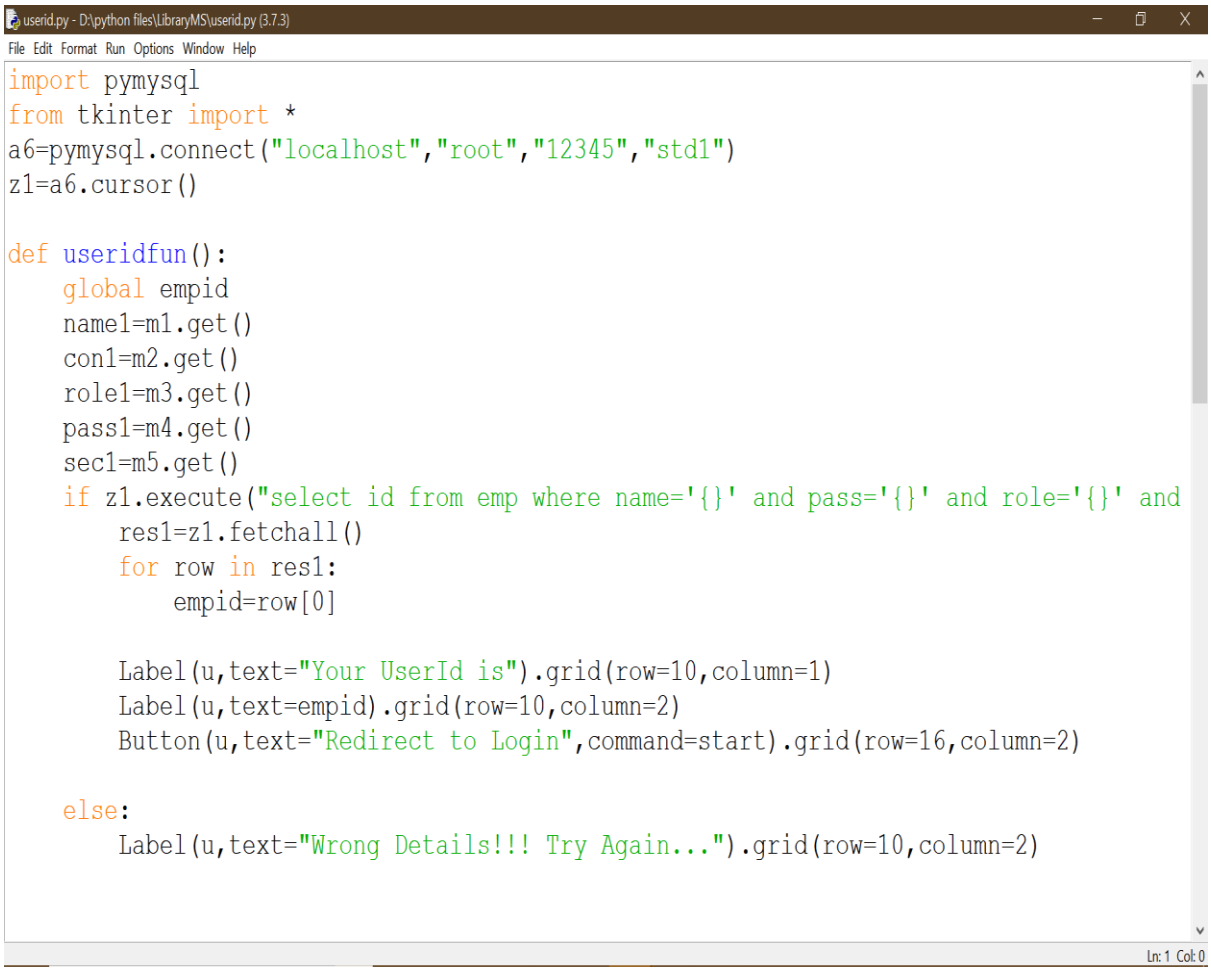
Ln: 1 Col: 0

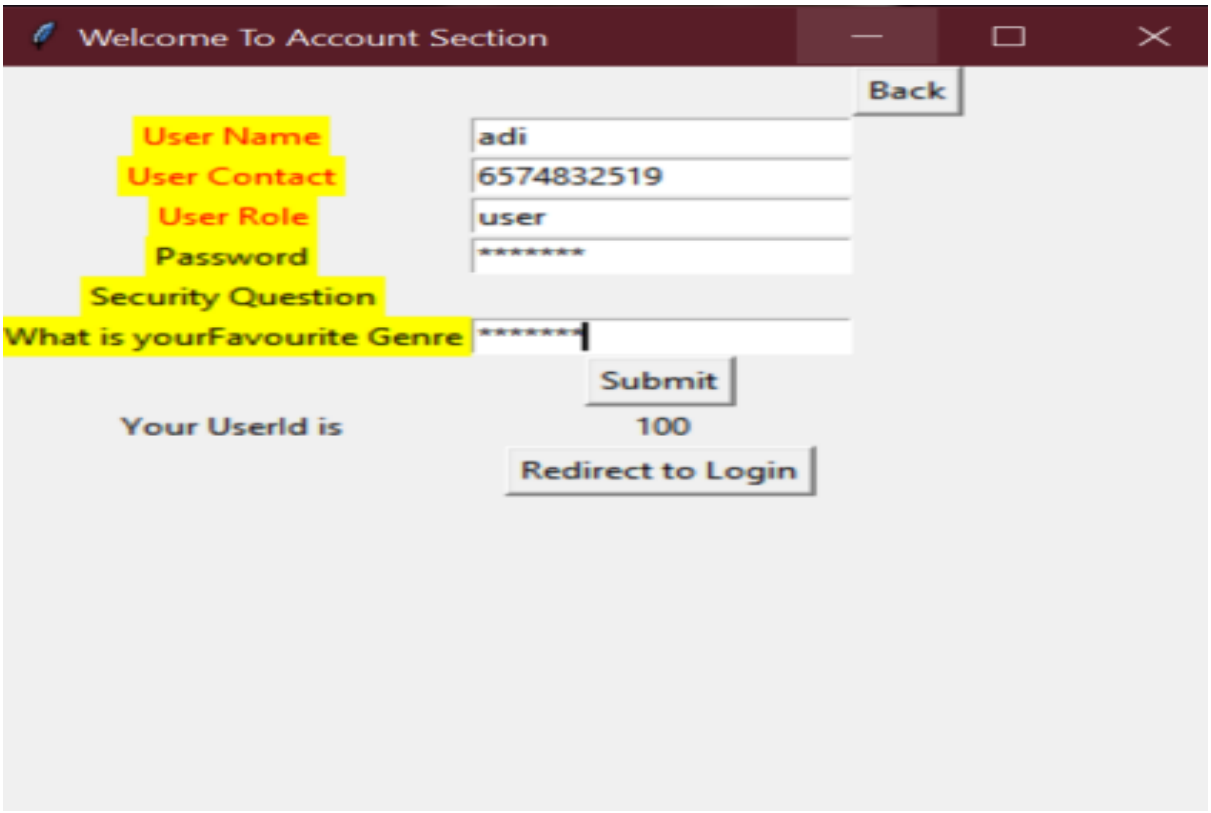

Accountant Section
Welcome To The Accountant Section
Back
LogOut
Name: 'adityagupta', Id: 102, Role: 'librarian', Bal: 1200
Do you want to continue as Librarian or as an User
Press 1 for Librarian & 2 for User
1
Submit
1 for Updating Balance, 2 for Updating Quantity and Price, 3 for Transaction List
2
Submit
Id of Book
Name of Book
Price of Book
Quantity of Book
Submit


Accountant Section
Welcome To The Accountant Section
Back
LogOut
Name: 'adityagupta', Id: 102, Role: 'librarian', Bal: 1200
Do you want to continue as Librarian or as an User
Press 1 for Librarian & 2 for User
2
Submit
Book Id
Book Name
Submit
Available Books Are:
1000 science 200 34
1001 maths 150 20
1003 french 100 5
3900 bigdata 250 22
Book Code
Book Name
Search

5) USER ID MODULE :

Through this module the user can know details of the user id if the person can't remember the user ID but remembers the password and other details. The person will be asked to enter all the details like Name, Contact, Password, Role. He also has to answer the Security Question in order to get the detail about the user id.





6) FORGOT PASSWORD MODULE :

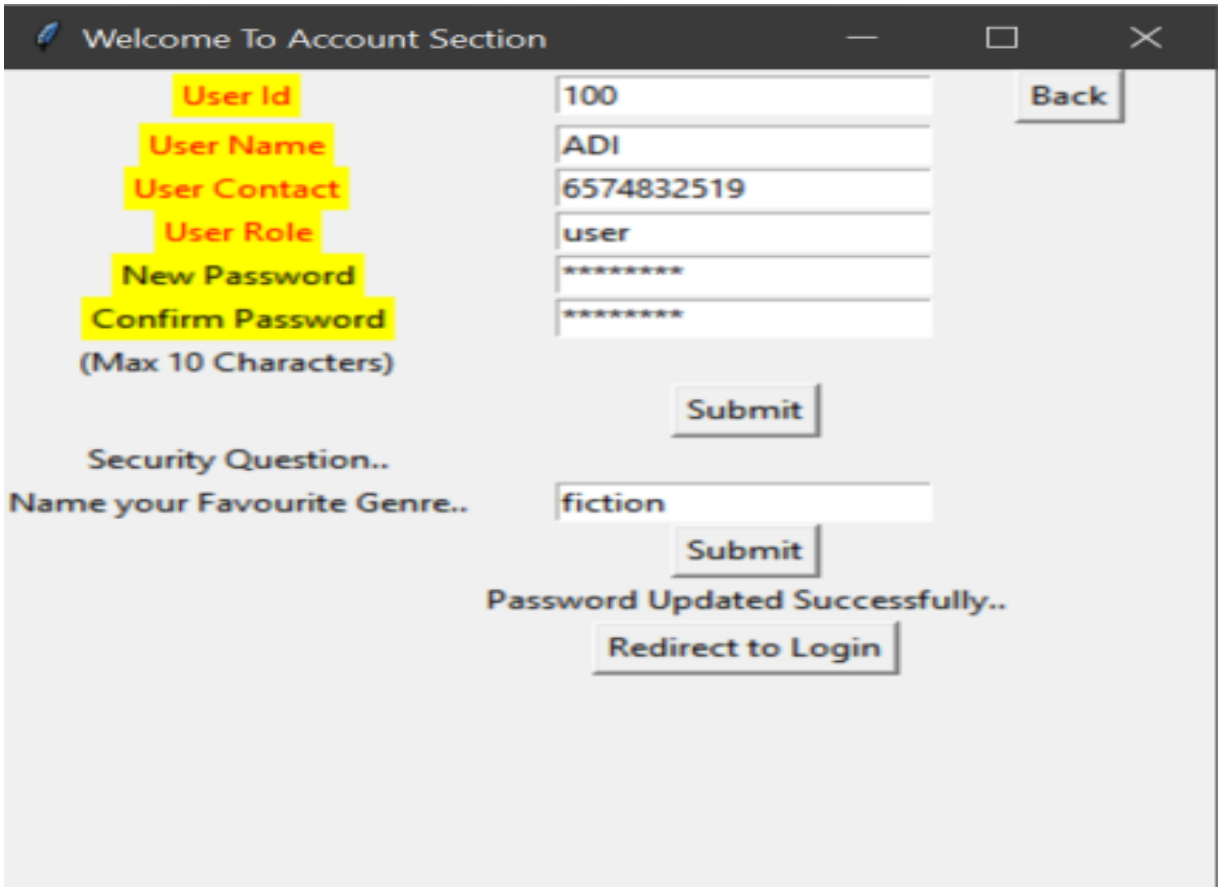
This module helps to reset a new password to the user once he enters his complete details like user id, user name, contact no. , user role, and a 10 character new pass word. Once his details are verified by the system he is asked a security question after the correct answer of which the password is changed into the database and the person can login with his new password to the system. A confirmation mail is also sent to the user on his registered mail id once the password is reset so that if the someone else has changed the password then the user can report the matter to the bookstore authority.

forgotpass.py - D:\python files\LibraryMS\forgotpass.py (3.7.3)

File Edit Format Run Options Window Help

```
def forgfun():
    global pass1
    pass1=m5.get()
    pass2=m6.get()
    global ans
    if pass1==pass2:

        Label(m,text="Security Question..").grid(row=10,column=1)
        Label(m,text="Name your Favourite Genre..").grid(row=12,column=1)
        ans=Entry(m)
        ans.grid(row=12, column=2)
        Button(m,text="Submit",command=seccheck).grid(row=13,column=2)

def seccheck():
    global x
    x=m1.get()
    z.execute("select security from emp where id={} and name='{}'".format(m1.get(),m2.
    res=z.fetchall()
    for row in res:
        esecurity=row[0]
    if esecurity==ans.get():
        z.execute("update emp set pass='{}' where id={}".format(pass1,m1.get()))
        mailfunc1(x)
        Label(m,text="Password Updated Successfully..").grid(row=14,column=2)
        Button(m,text="Redirect to Login",command=start).grid(row=16,column=2)
```

Ln: 1 Col: 0



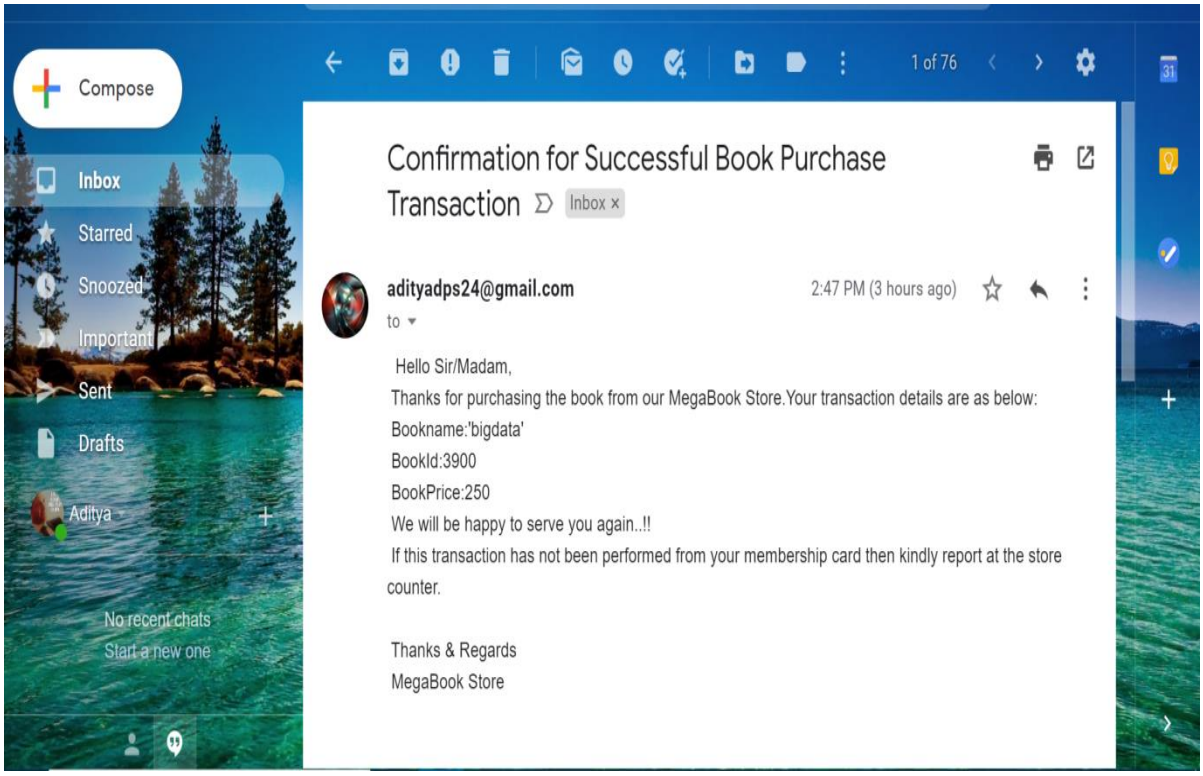
7) MAILER MODULE :

Through this module a confirmation mail will be sent to the registered email id of the customer by the book store giving details about the book purchased by him at the book store ie. The book price, book id, book name .

mailer.py - D:\python files\LibraryMS\mailer.py (3.7.3)

File Edit Format Run Options Window Help

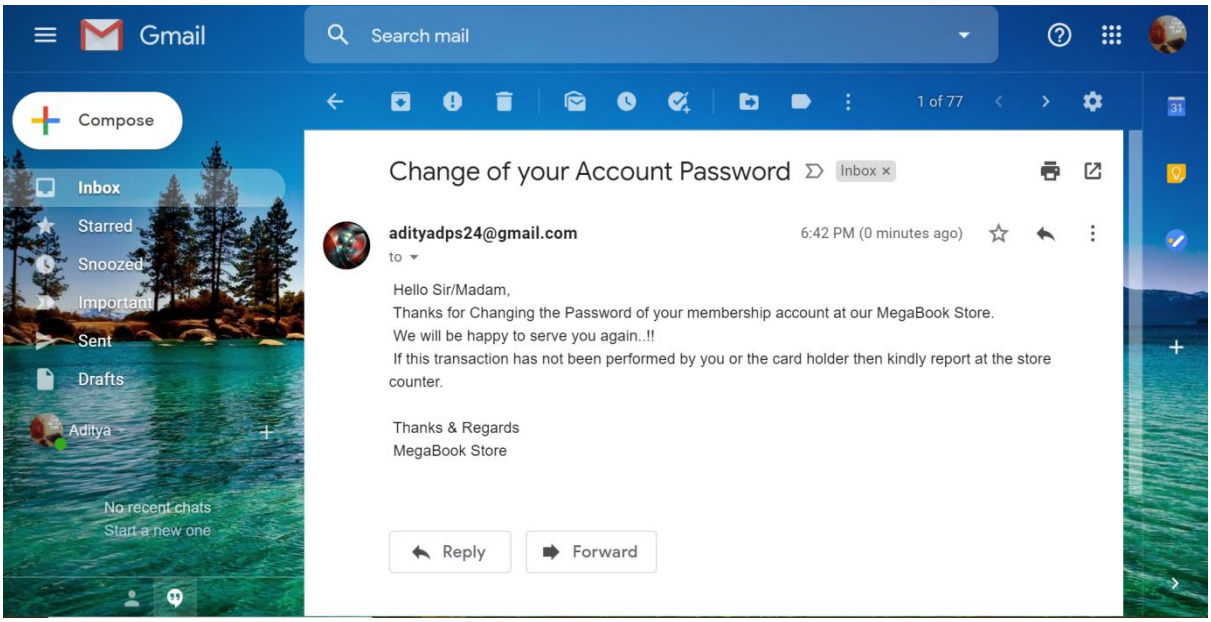
```
# Open GMAIL, Go to setting > Security > Allow Less Secure Apps

import smtplib
import pymysql
a10=pymysql.connect("localhost","root","12345","std1")
b10=a10.cursor()



def mailfunc(e_id,b_id,b_name,b_price):
   global a
   a=e_id
   if b10.execute('select emailid from emp where id={}'.format(a)):
      res=b10.fetchall()
      for row in res:
         mail=row[0]

      EMAIL="adityadps24@gmail.com"
      PASSWORD="**************"
      TO=mail
      message = "Subject : Confirmation for Successful Book Purchase Transaction
      try:

         server = smtplib.SMTP('smtp.gmail.com:587')
         #smtpObj.sendmail(sender, receivers, message)
```



8) MAILER1 MODULE :

The job of this module is to sent the confirmation email to the registered email id of the user as n when he/she changes the password of his membership account. This will help the person to report the matter to the authorities in case the transaction has not initiated by him/her.

```
mailer1.py - D:\python files\LibraryMS\mailer1.py (3.7.3)
File  Edit  Format  Run  Options  Window  Help

# Open GMAIL, Go to setting > Security > Allow Less Secure Apps

import smtplib
import pymysql
a11=pymysql.connect("localhost","root","12345","std1")
b11=a11.cursor()


def mailfunc1(x):
    global a
    a=x
    if b11.execute('select emailid from emp where id={}'.format(a)):
        res=b11.fetchall()
        for row in res:
            mail=row[0]

        EMAIL="adityadps24@gmail.com"
        PASSWORD="***************"
        TO=mail
        message = "Subject : Change of your Account Password\n\n Hello Sir/Madam,\n Thar
        try:

            server = smtplib.SMTP('smtp.gmail.com:587')
                                                                    Ln: 20  Col: 30
```



9) MAILER2 MODULE :

The job assigned to this module is to send the confirmation mail to the registered id of the user whenever the person recharges money his membership card. It will send the mail containing details of the person and details of the amount added as well as the updated current balance

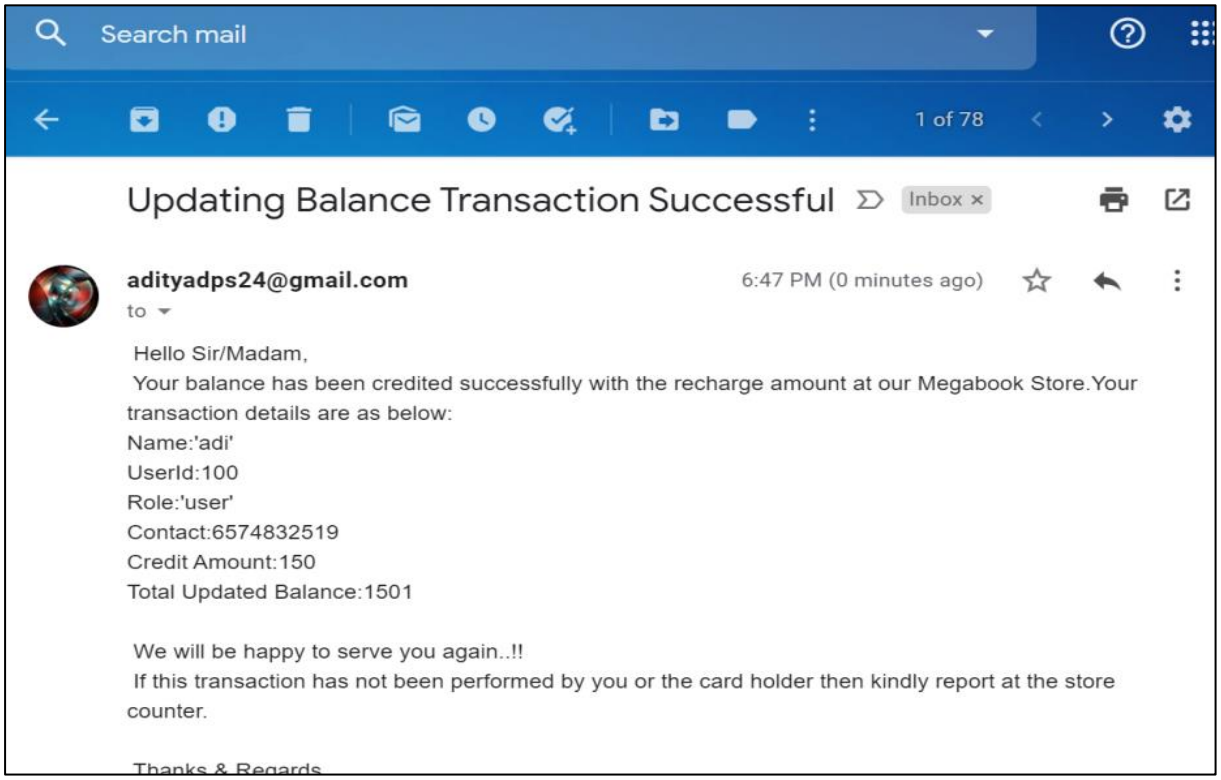
which will help the customer to keep a track of the money in the membership card.

mailer2.py - D:\python files\LibraryMS\mailer2.py (3.7.3)

File Edit Format Run Options Window Help

```
b12=a12.cursor()



def mailfunc2(addm,amt,zid):
   global a
   a=zid
   if b12.execute('select emailid from emp where id={}'.format(a)):
      res=b12.fetchall()
      for row in res:
         mail=row[0]
      b12.execute('select name,role,contact from emp where id={}'.format(a))
      res=b12.fetchall()
      for row in res:
         name=row[0]
         role=row[1]
         cont=row[2]



      EMAIL="adityadps24@gmail.com"
      PASSWORD="**************"
      TO=mail
      message = "Subject : Updating Balance Transaction Successful\n\n Hello Sir/Madam
      try:
```

Ln: 27 Col: 30



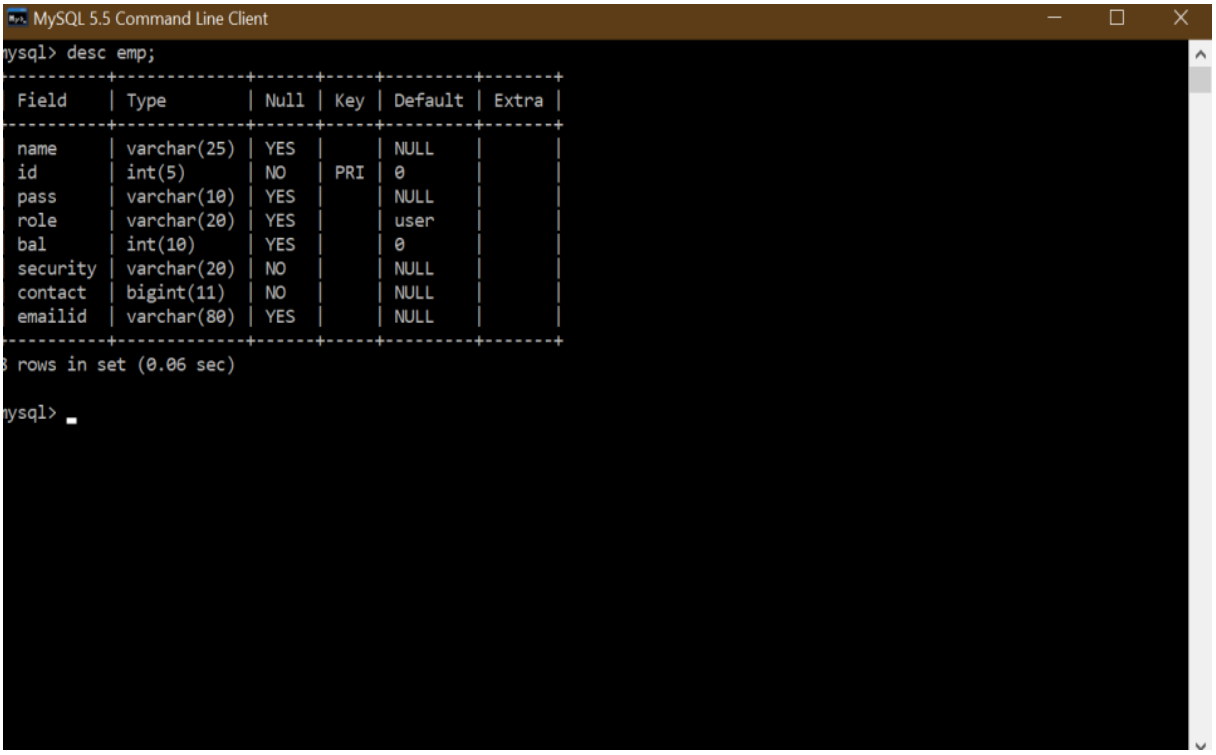
10) DATABASE MODULE :

(A) EMP MODULE :

This database module stores the details of the all the employees and user members of the Book store. Data Dictionary for the emp module is as follows :

- **Name - Stores name with the defined datatype as varchar.**
- **Id - Stores the id of the employees and the users with defined datatype as int. This is set as the primary key for the emp module.**
- **Pass - Stores the password of the member's account with defined datatype as varchar.**
- **Role - A specific role is assigned to the user according to which he is granted the access to the system. It is defined with varchar as the datatype.**
- **Bal - Stores the current updated balance amount of the members of the book store and is defined with int as the datatype.**
- **Security - Stores the answer to the security question which is used to verify the details of the customer when he changes details in his membership account. It's defined datatype is varchar.**

- **Contact - Stores the contact details of the members with defined datatype as bigint.**
- **Email - stores the email id of members which is used to sent transaction details and confirmation status to the members with the defined datatype as varchar.**

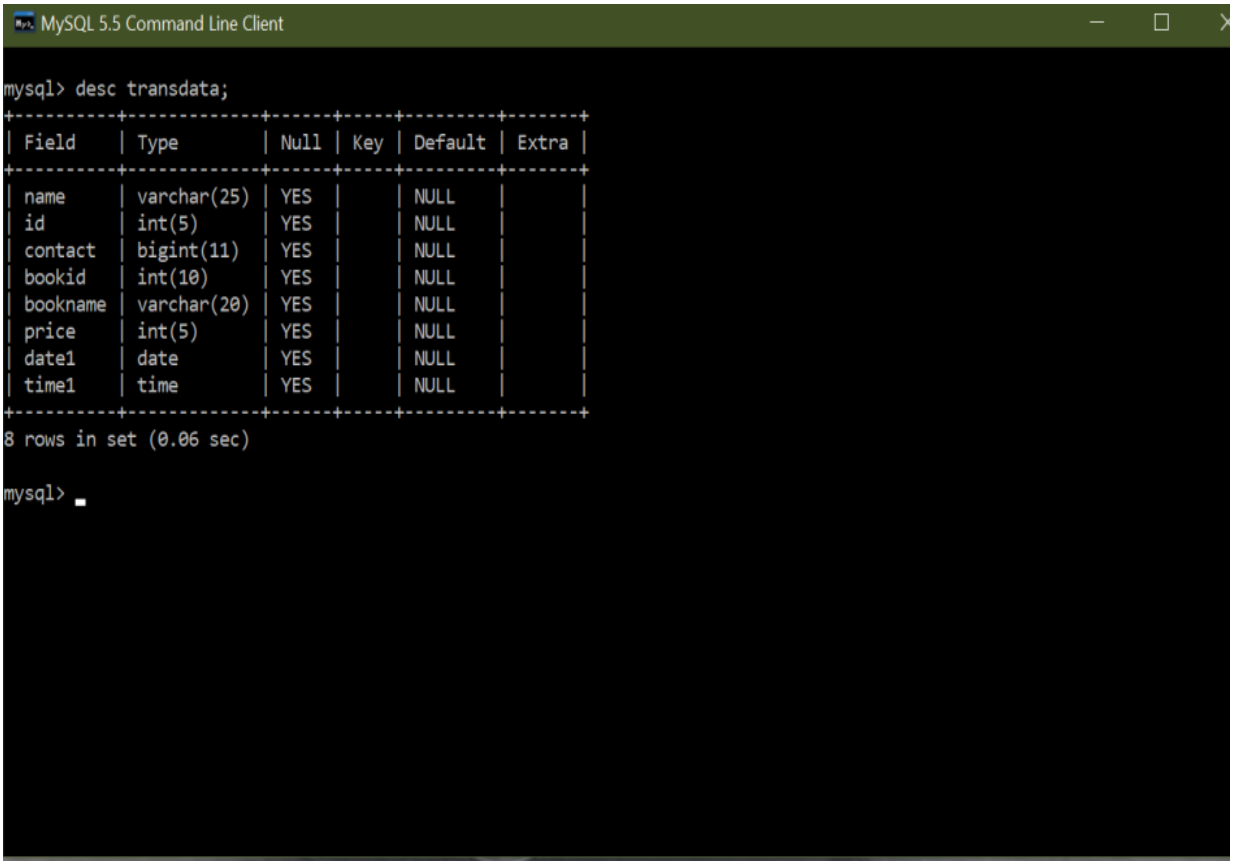


(B) TRANSDATA MODULE :

This module stores the day to day transaction details of book purchase with their transaction date and transaction time. The module stores the transaction details for a period of 6 months from the day of transaction and the database automatically removes the data after 6 months from its transaction date. So the database doesn't need to be

cleaned manually and periodically. Data dictionary for the transdata module is as follows :

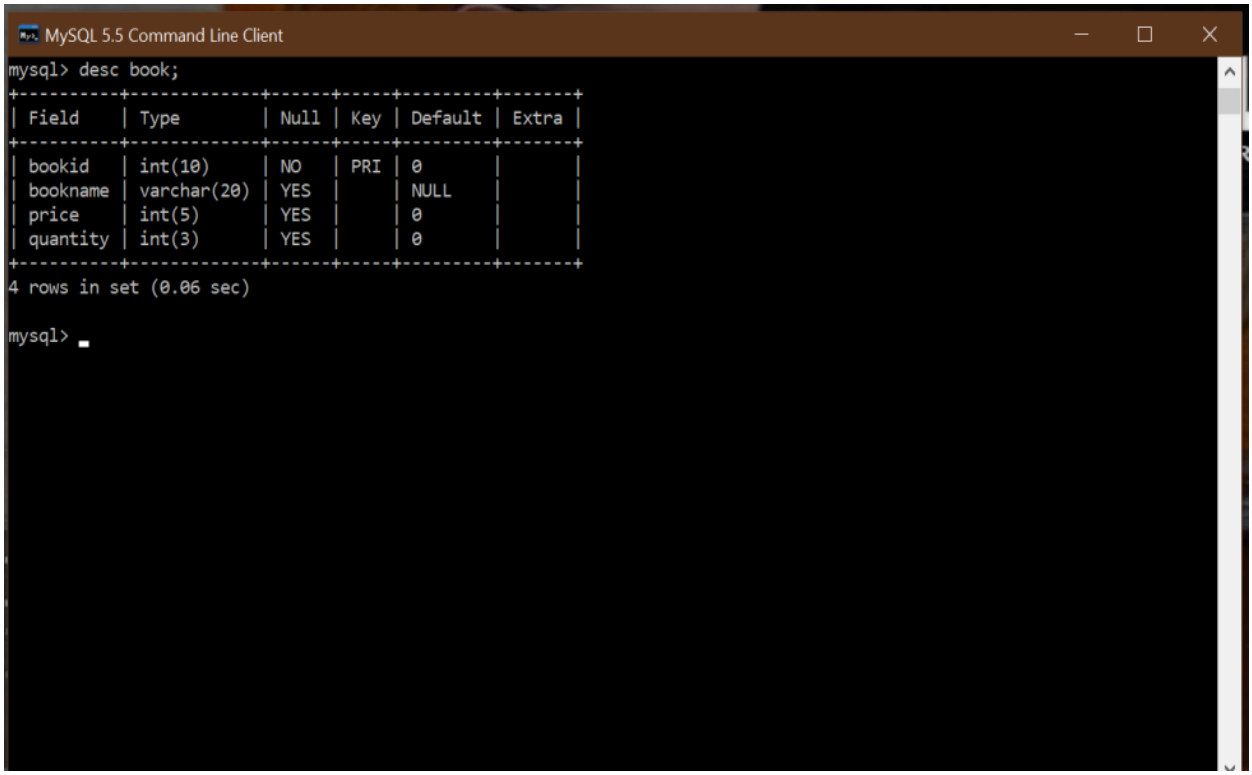
- **Name - It stores the name of the person who does the transaction of purchasing the book. It is defined with varchar as the datatype.**
- **Id - Stores the user id assigned to the person with its defined datatype as int.**
- **Contact - Stores the contact details of the person performing the transaction. It is defined with bigint as the datatype.**
- **BookId - Stores the unique book id of the book that is being purchased by the buyer. It is defined with int as the datatype.**
- **BookName - Stores the book name of the book that is being purchased by the buyer. It is defined with varchar as the datatype.**
- **Price - Stores the price of the book that is being purchased by the buyer. It is defined with int as the datatype.**
- **Date1 - It stores the date on which the transaction successfully happened at the store. It is defined with date as the datatype in the format DD:MM:YYYY .**
- **Time1 - It stores the time at which the transaction successfully happened at the store. It is defined with time as the datatype in the format HH:MM:SS.**



(C) BOOK MODULE :

This module is assigned the job to store the details of the books available in the Bookstore with its assigned book id, book name, its quantity and price. Data Dictionary for the book module is as follows :

- **BookId - Stores the assigned unique bookid with its defined datatype as int. It is assigned as the primary key for this module.**
- **BookName - Stores the book's name with it's defined datatype as varchar.**
- **Price - Stores the price of the book with its defined datatype as int.**
- **Quantity - Stores the quantity of the book available in stock at the book store with it's defined datatype as int .**



DATABASE MANAGEMENT SYSTEM (DBMS)

Keeping in mind the future and considering scalability, stability MySQL Server is used as database because of the following reasons:

- **Scalability and Flexibility**
- **Improved reliability,**
- **Better performance,**
- **Data Integrity**

DATABASE BACK-UP & RECOVERY

To minimize the risk of data loss, backup of databases on a regular basis is required. A planned backup and restore strategy help protect the databases against data loss caused by a variety of failures. The SQL server offers many options for creating backups. The SQL server backup and restore components provide an essential safeguard for protecting critical data stored in the database. With valid backups of the database, data can be recovered from failures like: Media failure, user errors, hardware failures and also from natural disasters. The logical backup technique uses IMPORT/EXPORT utilities to create the backup copy of the database. Physical backup is done by copying all the data files, redo log files and control files on a storage device such as drives or tapes. The following are the steps for backup and recovery procedure:

- **For redundancy purpose, copy of backup is transferred via FTP and stored on another system. RAID 5 (Redundant array of independent disks) is also used on all the systems to have protection from disk failures.**

CONCLUSION

Book shop management system is an attempt to overcome the present in efficient and time consuming process of locating reserving and purchasing quality reading materials available in the shop. Through automated book shop solution, it provides an easy way of searching reserving and purchasing of books. This software also reduces the work load of the Bookstore owner to manage the bookstore. By developing a database for the bookstore, we can have the complete details regarding the books available, customer & employee details, and order details. The user can get the flexibility of managing the entire information of the store by using this database model. It's worth analyzing and identifying the benefits as it would directly influence the productivity of the shop.

FUTURE SCOPE

The software development is never completed. There is always a room for modification. There could have been other approaches to implement the system. I have tried my level best to make the system as interactive as possible. Sending the confirmation and transactional details through SMS system can also be added as one more feature. This system can be launched through a website for online selling,feature of card payments can also be added. if possible implementation is not limited to books. Planning can be done to expand it with groceries or gift articles but not in near future. Any education institute can make use of it for providing information about author, content of the available books in their library. Modifications can be easily done according to requirements and when necessary. It can be used in any type of Book Shop for managing all the sales and purchased activities and managing the data records related to Book house.

REFERENCES

1. **www.academia.edu**
2. **www.freeprojectz.com**
3. **www.geeksforgeeks.com**
4. **www.wikipedia.com**
5. **www.googleimages.com**
6. **www.freeCodeCamp.org**
7. **www.tutorialspoint.com**
8. **www.realpython.com**