NEW ERA PUBLIC SCHOOL

Mayapuri, New Delhi

COMPUTER SCIENCE

PROJECT

2020-2021

TOPIC:

BOOKSTORE MANAGEMENT SYSTEM

Student name: Abhinav Sharma
Class: XII G
Roll no. 2

TABLE OF CONTENTS

Introduction to Python

Python is a high-level, interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. The initial version was published at the alt.sources newsgroup in 1991, and version 1.0 was released in 1994. The latest version of python is currently python 3.0.

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourage program modularity and code reuse. Python supports a wide variety of different programming paradigms, including procedural programming, object oriented programming and functional programming. It's flexible enough to adapt to a lot of different needs.

Introduction to the Project

Book store management software is a sales and issue handling software which deals with the data concerning about the books in the store and their details such as price, quantity, authors and etc. the software is meant to be used by store clerks and managers to keep track of their stock and to help the customers get and find what the need. The software has several useful functions.

This software also deals with issuing books as well as buying them. The software automatically reduces the quantity of books as a book is issued or bought and gains quantity when it has returned. Only registered workers can use the system and the system can also register you but will need the password of the admin for security purposes and can also remove registered users by the same protocol. The program is neatly ordered at the backend by the functions and also has a clean and fully functional front end. The software is easy to comprehend.

Acknowledgement

I thank my Computer Science teacher Mrs. Gurjeet Kaur for her guidance and support in this terrible situation due to COVID-19, I am also thankful to our principal Mrs. Vandana Chawla. I would also thank my parents for encouraging me during the course of the project. Finally, I would like to thank CBSE for giving me this opportunity to undertake this project.

Certificate

This is to certify that Abhinav Sharma of Class XII, New Era Public School, Mayapuri New Delhi has successfully completed his project in Computer Science practical for the AISSCE as prescribed by the CBSE in the year of 2020-2021.

Roll No:

Sign. of Internal

Sign. of External

System Requirements

Hardware Requirement:

- Printer (optional to print records etc)
- Compact Drive
- Processor: Pentium III or above
- Ram: 350 MB(minimum)
- Hard-Disk: 20 GB(minimum)

Software Requirement:

- Windows 7 or higher
- MySql server 5.5 or higher(as backend)
- Python Idle 3.6 or any compatible python interpreter
- Microsoft Word 2010 or higher for documentation.

Backend Code

```
import mysql.connector as sql
from Animations import Animations as animate


class bookdb:
    conec = sql.connect(host="localhost",user="root",password="",database="bookstore")
    if conec.is_connected():
        print("connection certified")


    def showtable(self):
        conec =
         sql.connect(host="localhost",user="root",password="",database="bookstore")
        cur=conec.cursor()
        print("showing data in the system")
        animate.dots(self,5)
        cur.execute("select * from books")
        show=cur.fetchall()


        print ("+" ,"-"*120,"+")
        print ("| Book ID          ","| Book Name                              ","| Author                  ","|
         Quantity     ","| Price         |")
        for row in show:
            bookID = "{:<16}".format(row[0])
            bookName = "{:<38}".format(row[1])
            bookAuthor = "{:<25}".format(row[2])
            bookPrice = "{:<12}".format(row[3])
```

```
        bookQuantity = row[4]
        bookID = str(bookID)
        bookPrice = str(bookPrice)
        bookQuantity = str(bookQuantity)
        print ("|" ,"-"*17, "+","-"*39,"+","-"*26,"+","-"*13,"+","-"*13,"|")
        print ("|",bookID,' '*(15-len(bookID)) , "|",bookName,' '*(37-len(bookName)) ,
      "|",bookAuthor,' '*(24-len(bookAuthor)) , "|",bookPrice,' '*(11-len(bookPrice)) ,
      '|',bookQuantity, ' '*(12-len(bookQuantity)),"|")
    print ("+" ,"-"*120,"+")
    go = input("PRESS ENTER TO GO BACK")

def addtable(self):
    conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
    cur=conec.cursor()
    reply = "y"
    while reply == "y":
        ID_input = int(input("Enter ID: "))
        title_input = input("Enter Title name: ")
        author_input = input("Enter Author: ")
        price_input = int(input("Enter Price: "))
        quantity_input = int(input("Enter Quantity: "))
        rec = [ID_input,title_input,author_input,price_input,quantity_input]
        query = "INSERT INTO books VALUES (%s,%s,%s,%s,%s)"
        cur.execute(query,rec)
        conec.commit()
        print ("")
        print("     **New data added**")
        print ("")
        print("Add more? (y/n) ")
        reply = (input())
```

```
def searchbook(self):
    conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
    cur=conec.cursor()
    reply = "y"
    while reply == "y":
        cond = input("Enter the BookID you need to find: ")
        cur.execute("select * from books where BookID = '%s'" %(cond))
        show=cur.fetchall()

        print ("+" ,"-"*120,"+")
        print ("| Book ID          ","| Book Name                              ","| Author
     ","| Quantity     ","| Price         |")
        for row in show:
            bookID = "{:<16}".format(row[0])
            bookName = "{:<38}".format(row[1])
            bookAuthor = "{:<25}".format(row[2])
            bookPrice = "{:<12}".format(row[3])
            bookQuantity = row[4]
            bookID = str(bookID)
            bookPrice = str(bookPrice)
            bookQuantity = str(bookQuantity)
            print ("|" ,"-"*17, "+","-"*39,"+","-"*26,"+","-"*13,"+","-"*13,"|")
            print ("|",bookID,' '*(15-len(bookID)) , "|",bookName,' '*(37-len(bookName)) ,
      "|",bookAuthor,' '*(24-len(bookAuthor)) , "|",bookPrice,' '*(11-len(bookPrice)) ,
      '|',bookQuantity, ' '*(12-len(bookQuantity)),"|")
            print ("+" ,"-"*120,"+")
        print("Search more? (y/n) ")
        reply = (input())

def search_author(self):
```

```
    conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
    cur=conec.cursor()
    reply = "y"
    while reply == "y":
        cond = input("Enter the Author you need to find: ")
        cur.execute("select * from books where Author = '%s'" %(cond))
        show=cur.fetchall()

        print ("+" ,"-"*120,"+")
        print ("| Book ID          ","| Book Name                              ","| Author
     ","| Quantity     ","| Price         |")
        for row in show:
            bookID = "{:<16}".format(row[0])
            bookName = "{:<38}".format(row[1])
            bookAuthor = "{:<25}".format(row[2])
            bookPrice = "{:<12}".format(row[3])
            bookQuantity = row[4]
            bookID = str(bookID)
            bookPrice = str(bookPrice)
            bookQuantity = str(bookQuantity)
            print ("|" ,"-"*17, "+","-"*39,"+","-"*26,"+","-"*13,"+","-"*13,"|")
            print ("|",bookID,' '*(15-len(bookID)) , "|",bookName,' '*(37-len(bookName)) ,
      "|",bookAuthor,' '*(24-len(bookAuthor)) , "|",bookPrice,' '*(11-len(bookPrice)) ,
      '|',bookQuantity, ' '*(12-len(bookQuantity)),"|")
        print ("+" ,"-"*120,"+")
        print("search more? (y/n) ")
        reply = (input())

def search_price(self):
    conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
```

```
    cur=conec.cursor()
    reply = "y"
    while reply == "y":
        cond = input("Enter the mininum price: ")
        cond1 = input("Enter the maximum price: ")
        inp = [cond,cond1]
        query ="select * from books where Price between %s and %s"
        cur.execute(query,inp)
        show=cur.fetchall()
        print ("+" ,"-"*120,"+")
        print ("| Book ID          ","| Book Name                              ","| Author
     ","| Quantity     ","| Price         |")
        for row in show:
            bookID = "{:<16}".format(row[0])
            bookName = "{:<38}".format(row[1])
            bookAuthor = "{:<25}".format(row[2])
            bookPrice = "{:<12}".format(row[3])
            bookQuantity = row[4]
            bookID = str(bookID)
            bookPrice = str(bookPrice)
            bookQuantity = str(bookQuantity)
            print ("|" ,"-"*17, "+","-"*39,"+","-"*26,"+","-"*13,"+","-"*13,"|")
            print ("|",bookID,' '*(15-len(bookID)) , "|",bookName,' '*(37-len(bookName)) ,
      "|",bookAuthor,' '*(24-len(bookAuthor)) , "|",bookPrice,' '*(11-len(bookPrice)) ,
      '|',bookQuantity, ' '*(12-len(bookQuantity)),"|")
        print ("+" ,"-"*120,"+")
        print("search more? (y/n) ")
        reply = (input())

def deletebook(self):
    conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
```

```
    cur=conec.cursor()
    reply = "y"
    while reply == "y":
        cond = int(input("Enter the BookID you need to delete: "))
        cur.execute("DELETE FROM books WHERE BookID = '%s'" %(cond))
        conec.commit()
        print("")
        print("     **Author deleted successfully**")
        print("")
        print("Delete more records? (y/n) ")
        reply = (input())

def deleteauthor(self):
    conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
    cur=conec.cursor()
    reply = "y"
    while reply == "y":
        cond = (input("Enter the Author whose books you want to delete: "))
        cur.execute("DELETE FROM books WHERE Author = '%s'" %(cond))
        conec.commit()
        print("")
        print("     **Author deleted successfully**")
        print("")
        print("Delete more records? (y/n) ")
        reply = (input())

def makebill(self):
    conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
    cur=conec.cursor()
    bill = 0
```

```
    reply = "y"
    while reply == "y":
        cond = int(input("Enter the BookID : "))
        quan = int(input("Enter the number of books to be bought: "))
        cur.execute("select Quantity from books where BookID = '%s'" %(cond))
        change = cur.fetchone()
        for j in change:
            change2 = int(j)
        change2 = change2-quan
        print (change2)
        cond1 = [change2,cond]
        query = "update books set Quantity = '%s' where BookID = '%s'"
        cur.execute(query,cond1)
        conec.commit()
        cur.execute("select Price from books where BookID = '%s'" %(cond))
        show = cur.fetchone()
        for i in show:
            pri = int(i)
        amount = pri*quan
        bill += amount
        print ("Amount is ",amount)
        print("add more items to the bill? (y/n) ")
        reply = (input())
    print ("The Total bill is ",bill)
    go = input("PRESS ENTER TO GO BACK")

def login(self):
    user = input("Enter Username: ")
    password = input("Enter Password: ")
    conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
    cur=conec.cursor()
```

```
    cur.execute("select Password from admin where Username = '%s'" %(user))
    show = cur.fetchone()
    if show is not None:
        for i in show:
            if i == password:
                return True

def register(self):
    reply = "y"
    while reply == "y":
        print("Create ID:")
        userName=input()
        print("Create Password")
        userPass = input()
        conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
        cur=conec.cursor()
        inp = [userName,userPass]
        query = "insert into admin values(%s,%s)"
        cur.execute(query,inp)
        conec.commit()
        animate.dots(self,4)
        print ("     **You have been added to the system**")
        print("Add more Users? (y/n) ")
        reply = (input())

def deleteuser(self):
    access = input("Enter Admin password: ")
    print("")
    if access == "1111" :
        reply = "y"
        while reply == "y":
```

```
            cond = (input("Enter the Username : "))
            print("")
            conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
            cur = conec.cursor()
            cur.execute("DELETE FROM admin WHERE Username = '%s'" %(cond))
            conec.commit()
            print ("     ***USER REMOVED*** ")
            print("")
            print("Delete more Users? (y/n) ")
            conec.commit()
            reply = (input())
    else:
        print("")
        print("**Not Authorised for such actions**")

def updatePri(self):
    conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
    cur=conec.cursor()
    reply = "y"
    while reply == "y":
        cond = int(input("Enter the BookID you want to updated: "))
        cond1 = int(input("Set Price to: "))
        condition = [cond1,cond]
        query = "update books set Price = '%s' where BookID = '%s'"
        cur.execute(query,condition)
        conec.commit()
        print("")
        print("     **Price updated successfully**")
        print("")
        print("Update more Prices? (y/n) ")
```

```
        reply = (input())

def updateQuan(self):
        conec =
      sql.connect(host="localhost",user="root",password="",database="bookstore")
        cur=conec.cursor()
        reply = "y"
        while reply == "y":
            cond = int(input("Enter the Book ID of book you want to update: "))
            cond1 = int(input("Set Quantity to: "))
            condition = [cond1,cond]
            query = "update books set Quantity = '%s' where BookID = '%s'"
            cur.execute(query,condition)
            conec.commit()
            print("")
            print("     **Quantity updated successfully**")
            print("")
            print("Update more Quantities? (y/n) ")
            reply = (input())

def updateID(self):
        conec =
      sql.connect(host="localhost",user="root",password="",database="bookstore")
        cur=conec.cursor()
        reply = "y"
        count = 0
        result = False
        while reply == "y":
            cur.execute("select BookID from books")
            show = cur.fetchall()
            print ("Already entered IDs:")
            for i in show:
```

```
            i = str(i)
            print (i)
        cond = int(input("Enter the Book ID you want to update: "))
        cond1 = int(input("Change ID to: "))
        condition = [cond1,cond]
        query = "update books set BookID = %s where BookID = %s"
        cur.execute(query,condition)
        conec.commit()
        print("")
        print("     **ID updated successfully**")
        print("")
        print("Update more IDs? (y/n) ")
        reply = (input())

def showissue(self):
    conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
    cur=conec.cursor()
    print("showing data in the system")
    animate.dots(self,5)
    cur.execute("select * from issue")
    show=cur.fetchall()

    print ("+" ,"-"*101,"+")
    print ("| Member ID    ","| Name                        ","| Date of Issue   ","| Date of
     Return  ","| Book ID      |")
    for row in show:
        memberID = row[0]
        Name = row[1]
        doi = row[2]
        dor = row[3]
        bookID = row[4]
```

```
        bookID = str(bookID)
        memberID = str(memberID)
        print ("|" ,"-"*13, "+","-"*32,"+","-"*16,"+","-"*16,"+","-"*12,"|")
        print ("|",memberID,' '*(12-len(memberID)) , "|",Name,' '*(31-len(Name)) ,
      "|",doi,' '*5, "|",dor,' '*5, '|',bookID, ' '*(11-len(bookID)),"|")
    print ("+","-"*101,"+")
    go = input("PRESS ENTER TO GO BACK ")

def deleteissue(self):
    conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
    cur=conec.cursor()
    reply = "y"
    while reply == "y":
        cond = int(input("Enter the Memeber ID of issued book: "))
        cur.execute("DELETE FROM issue WHERE MemberID = '%s'" %(cond))
        conec.commit()
        print("")
        print("     ** BOOK ISSUE RECORD DELETED**")
        cur.execute("select Quantity from books B , issue I where B.BookID=I.BookID
       and MemberID = '%s'" %(cond))
        change = cur.fetchone()
        for j in change:
            change2 = int(j)
        change2 = change2+1
        cond1 = [change2,cond]
        print (change2)
        query = "update books set Quantity = '%s' where BookID = '%s'"
        cur.execute(query,cond1)
        conec.commit()
        print("")
        print("Delete more records? (y/n) ")
```

```
        reply = (input())

def issuebook(self):
    conec =
     sql.connect(host="localhost",user="root",password="",database="bookstore")
    cur=conec.cursor()
    reply = "y"
    while reply == "y":
        ID_input = int(input("Enter Member ID: "))
        title_input = input("Enter member's Name: ")
        author_input = input("Enter Date of issue (YYYY-MM-DD) : ")
        price_input =  input("Enter Date of return (YYYY-MM-DD) : ")
        quantity_input = int(input("Enter Book ID: "))
        rec = [ID_input,title_input,author_input,price_input,quantity_input]
        query = "INSERT INTO issue VALUES (%s,%s,%s,%s,%s)"
        cur.execute(query,rec)
        conec.commit()
        cur.execute("select Quantity from books where BookID = '%s'"
     %(quantity_input))
        change = cur.fetchone()
        for j in change:
            change2 = int(j)
        change2 = change2-1
        cond1 = [change2,quantity_input]
        query = "update books set Quantity = '%s' where BookID = '%s'"
        cur.execute(query,cond1)
        conec.commit()
        print ("")
        print("     **BOOK ISSUED**")
        print ("")
        print("Issue more? (y/n) ")
        reply = (input())
```

Frontend code

```
import mysql.connector as sql
from Animations import Animations
import time
import sys
import os
import getpass
from mysql_bookmanagement import bookdb

class Main:
    def __init__(self):
        self.clear()
        self.animate = Animations()
        self.animate.dots(4)
        print("Getting you online")
        self.animate.dots(3)
        self.DB = bookdb()

    def clear(self):
        os.system('cls' if os.name=='nt' else 'clear')



if __name__ == "__main__":
    obj=Main()
    print("Starting the Software")
    obj.animate.dots(4)
    conec = sql.connect(host="localhost",user="root",password="",database="bookstore")
    if conec.is_connected():
        print("connection certified")
```

```
else:
    print("Connection Failed Try later.....")
obj.clear()
print("")
print("Enter your choice")
print("1.Login")
print("2.Register")
print("3.Remove User")
choice = input()
obj.clear()
if(choice=="1"):
    log = obj.DB.login()
    if log == True:
        print("")
        print ("      **User confirmed**")
        print("")
        print("Logging you in please wait")
        obj.animate.dots(7)
        selectInput = 0
        while selectInput != "12":
            print("\n-----------------WELCOME TO THE BOOK
     MANAGEMT----------------- \n")
            print("--------------------------MENU------------------------------ \n")
            print("    1. Search by Book ID")
            print("    2. Search by Author")
            print("    3. Search by Price range")
            print("    4. Show Book List")
            print("    5. Add item")
            print("    6. Delete by Book")
            print("    7. Delete by Author")
            print("    8. Make bill")
            print("    9. Update Book ID")
```

```
print("   10. Update Price")
print("   11. Update Quantity")
print("                    Issued Books")
print("       a. Show issued book list")
print("       b. Issue books")
print("       c. Delete issue records")
print("   12. Exit")
print("")
print("-----------------------------------------------------------------")
print("")
selectInput = input("Enter your choice:\n")
if(selectInput=="1"):
    obj.DB.searchbook()
elif(selectInput=="2"):
    obj.DB.search_author()
elif(selectInput=="3"):
    obj.DB.search_price()
elif(selectInput=="4"):
    obj.DB.showtable()
elif(selectInput=="5"):
    obj.DB.addtable()
elif(selectInput=="6"):
    obj.DB.deletebook()
elif(selectInput=="7"):
    obj.DB.deleteauthor()
elif(selectInput=="8"):
    obj.DB.makebill()
elif(selectInput=="9"):
    obj.DB.updateID()
elif(selectInput=="10"):
    obj.DB.updatePri()
elif(selectInput=="11"):
```

```
            obj.DB.updateQuan()
        elif(selectInput=="a"):
            obj.DB.showissue()
        elif(selectInput=="b"):
            obj.DB.issuebook()
        elif(selectInput=="c"):
            obj.DB.deleteissue()
        elif(selectInput=="12"):
            obj.animate.dots(4)
            print("logging you out")
            obj.animate.dots(3)
        else:
            print("Choose between 1 to 12 or a,b,c")
            break
    else:
        print("")
        print("      **Access Denied**")

elif(choice=="2"):
    access = input("Enter the Admin Password to create a new Account: ")
    if access == "1111":
        obj.DB.register()
        print("")
        print ("restart to continue")
    else:
        print("")
        print("**Not Authorised for such actions**")

elif(choice=="3"):
    obj.DB.deleteuser()
```

Animation Code:

```
import time

class Animations:
        def dots(self, x):
                for i in range(1,x):
                        print(".")
                        time.sleep(0.3)
```

Motive

- To keep records of bookstores digital and organized.

- To increase security of the records

- To make running a bookstore easier.

- To present computing skills I have picked up in the last 2 years.

Output Screenshots

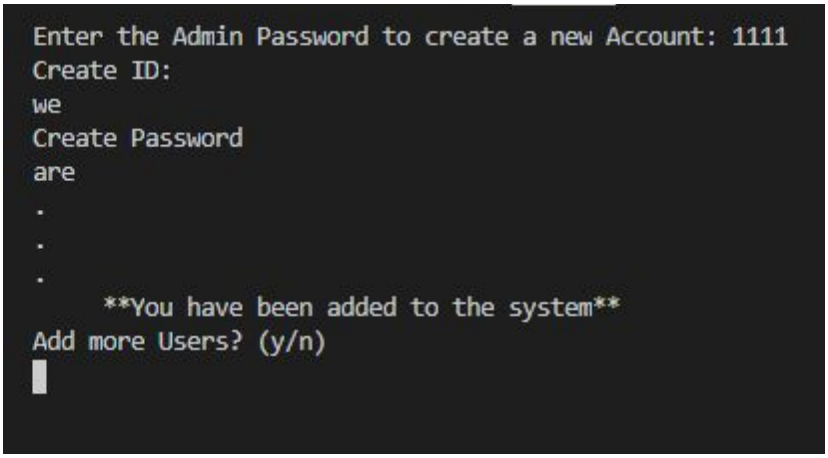
MySql tables before:

```
Database changed
mysql> select * from books;
+---------+------------------------+----------------+----------+--------+
| BookID  | Book_Name              | Author         | Quantity | Price  |
+---------+------------------------+----------------+----------+--------+
|       1 | The Promised Neverland | Kaiu Shirai    |       38 |    300 |
|     123 | we are                 | abhinav        |      900 | 123456 |
|  195588 | Beastars               | Paru Itagaki   |     4992 |     40 |
| 1955886 | Attack on Titan        | Hajime Isayama |      573 |     33 |
+---------+------------------------+----------------+----------+--------+
4 rows in set (0.10 sec)

mysql> select * from issue;
+----------+-----------------+------------+------------+---------+
| MemberID | Name            | DOI        | DOR        | BookID  |
+----------+-----------------+------------+------------+---------+
|        2 | Abhinav Sharma  | 2020-01-11 | 2020-09-11 | 1955883 |
|        5 | Shinra Kuzakabe | 2020-09-23 | 2021-01-23 |  195588 |
+----------+-----------------+------------+------------+---------+
2 rows in set (0.08 sec)

mysql> select * from admin;
+----------+----------+
| Username | Password |
+----------+----------+
| 1        | 1        |
| a        | a        |
| abhinav  | 1111     |
| shruti   | 5678     |
+----------+----------+
```

Main project outputs:



```
------------------WELCOME TO THE BOOK MANAGEMT------------------

-----------------------------MENU---------------------------------

    1. Search by Book ID
    2. Search by Author
    3. Search by Price range
    4. Show Book List
    5. Add item
    6. Delete by Book
    7. Delete by Author
    8. Make bill
    9. Update Book ID
   10. Update Price
   11. Update Quantity
                        Issued Books
       a. Show issued book list
       b. Issue books
       c. Delete issue records
   12. Exit

-------------------------------------------------------------------

Enter your choice:
```

```
Enter your choice:
1
Enter the BookID you need to find: 195588
+ ---------- + ---------------------------------------- + ------------------------ + ------------ + ------------ +
| Book ID    | Book Name                                | Author                   | Quantity     | Price        |
| ---------- + ---------------------------------------- + ------------------------ + ------------ + ------------ |
| 195588     | Beastars                                 | Paru Itagaki             | 4992         | 40           |
+ ---------- + ---------------------------------------- + ------------------------ + ------------ + ------------ +
Search more? (y/n)
n

------------------WELCOME TO THE BOOK MANAGEMT------------------
```

```
Enter your choice:
2
Enter the Author you need to find: abhinav
+ ---------- + ---------------------------------------- + ------------------------ + ------------ + ------------ +
| Book ID    | Book Name                                | Author                   | Quantity     | Price        |
| ---------- + ---------------------------------------- + ------------------------ + ------------ + ------------ |
| 123        | we are                                   | abhinav                  | 900          | 123456       |
+ ---------- + ---------------------------------------- + ------------------------ + ------------ + ------------ +
search more? (y/n)
n
```

```
Enter your choice:
3
Enter the mininum price: 100
Enter the maximum price: 1000
+ ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- +
| Book ID    | Book Name                          | Author                 | Quantity   | Price      |
| ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- |
| 1          | The Promised Neverland             | Kaiu Shirai            | 38         | 300        |
+ ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- +
search more? (y/n)
y
Enter the mininum price: 10
Enter the maximum price: 100
+ ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- +
| Book ID    | Book Name                          | Author                 | Quantity   | Price      |
| ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- |
| 195588     | Beastars                           | Paru Itagaki           | 4992       | 40         |
| ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- |
| 1955886    | Attack on Titan                    | Hajime Isayama         | 573        | 33         |
+ ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- +
search more? (y/n)
n
```

```
Enter your choice:
4
showing data in the system
.
.
.
.
+ ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- +
| Book ID    | Book Name                          | Author                 | Quantity   | Price      |
| ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- |
| 1          | The Promised Neverland             | Kaiu Shirai            | 38         | 300        |
| ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- |
| 123        | we are                             | abhinav                | 900        | 123456     |
| ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- |
| 195588     | Beastars                           | Paru Itagaki           | 4992       | 40         |
| ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- |
| 1955886    | Attack on Titan                    | Hajime Isayama         | 573        | 33         |
+ ---------- + ---------------------------------- + ---------------------- + ---------- + ---------- +
PRESS ENTER TO GO BACK
```

```
Enter your choice:
5
Enter ID: 112
Enter Title name: school
Enter Author: Vinay kumar
Enter Price: 234
Enter Quantity: 12

     **New data added**

Add more? (y/n)
```

```
Enter your choice:
6
Enter the BookID you need to delete: 123

     **Author deleted successfully**

Delete more records? (y/n)
```

```
Enter your choice:
7
Enter the Author whose books you want to delete: Kaiu Shirai

     **Author deleted successfully**

Delete more records? (y/n)
```

```
Enter your choice:
8
Enter the BookID : 112
Enter the number of books to be bought: 3
231
Amount is  36
add more items to the bill? (y/n)
y
Enter the BookID : 1955886
Enter the number of books to be bought: 34
539
Amount is  1122
add more items to the bill? (y/n)
n
The Total bill is  1158
```

```
Enter your choice:
9
Already entered IDs:
(112,)
(195588,)
(1955886,)
Enter the Book ID you want to update: 1955886
Change ID to: 23

     **ID updated successfully**

Update more IDs? (y/n)
n
```

```
Enter your choice:
10
Enter the BookID you want to updated: 23
Set Price to: 555

     **Price updated successfully**

Update more Prices? (y/n)
```

```
Enter your choice:
11
Enter the Book ID of book you want to update: 112
Set Quantity to: 1

     **Quantity updated successfully**

Update more Quantities? (y/n)
```

```
Enter your choice:
a
showing data in the system
.
.
.
.
+ ---------- + ------------------------------ + ---------------- + ---------------- + ------------ +
| Member ID  | Name                           | Date of Issue    | Date of Return   | Book ID      |
| ---------- + ------------------------------ + ---------------- + ---------------- + ------------ |
| 2          | Abhinav Sharma                 | 2020-01-11       | 2020-09-11       | 1955883      |
| ---------- + ------------------------------ + ---------------- + ---------------- + ------------ |
| 5          | Shinra Kuzakabe                | 2020-09-23       | 2021-01-23       | 195588       |
+ ---------- + ------------------------------ + ---------------- + ---------------- + ------------ +
PRESS ENTER TO GO BACK
```

```
Enter your choice:
b
Enter Member ID: 6
Enter member's Name: Raj
Enter Date of issue (YYYY-MM-DD) : 2021-01-04
Enter Date of return (YYYY-MM-DD) : 2021-02-04
Enter Book ID: 23

     **BOOK ISSUED**

Issue more? (y/n)
```

```
Enter your choice:
c
Enter the Memeber ID of issued book: 5

      ** BOOK ISSUE RECORD DELETED**
```

```
Enter your choice:
12
.
.
.
logging you out
.
.
```

MySql tables after:

```
mysql> select * from books;
+--------+-----------------+----------------+----------+-------+
| BookID | Book_Name       | Author         | Quantity | Price |
+--------+-----------------+----------------+----------+-------+
|     23 | Attack on Titan | Hajime Isayama |      538 |   555 |
|    112 | school          | Vinay kumar    |        1 |    12 |
| 195588 | Beastars        | Paru Itagaki   |     4992 |    40 |
+--------+-----------------+----------------+----------+-------+
3 rows in set (0.00 sec)

mysql> select * from issue;
+----------+----------------+------------+------------+---------+
| MemberID | Name           | DOI        | DOR        | BookID  |
+----------+----------------+------------+------------+---------+
|        2 | Abhinav Sharma | 2020-01-11 | 2020-09-11 | 1955883 |
|        6 | Raj            | 2021-01-04 | 2021-02-04 |      23 |
+----------+----------------+------------+------------+---------+
2 rows in set (0.00 sec)

mysql> select * from admin;
+----------+----------+
| Username | Password |
+----------+----------+
| 1        | 1        |
| abhinav  | 1111     |
| shruti   | 5678     |
| we       | are      |
+----------+----------+
4 rows in set (0.00 sec)
```

BIBLIOGRAPHY

Books:

COMPUTER SCIENCE WITH PYTHON- BY PREETI ARORA

Class notes

Websites:

www.geeksforgeeks.org

https://www.w3schools.com/python/

Limitations

- Invalid input might trigger an error.

- Members can not sign up without issuing a book

- After registering or deleting, the user must restart the program.