



Private International English School, Abu Dhabi

CERTIFICATE

This is to certify that Sri/Kum _____ of
Class _____ Examination Registration no _____
has satisfactorily completed the Project work in _____
during the academic session 20_____ to 20_____ as
prescribed by the C.B.S.E New Delhi.

Teacher In-charge

Principal

Date of Examination: _____

Examiner 1: _____

Examiner 2: _____

ACKNOWLEDGEMENT

In completing this investigatory project, we have been fortunate to have help, support and encouragement from many people. We would like to acknowledge them for their cooperation.

We would like to express our gratitude to our principal Mrs. Girija Baiju and our vice principal Mr. Suresh V. Balakrishnan for providing us with the platform and the opportunity to do such an innovative project.

We would like to thank our teacher Mrs. Jisha Anil Jose, our project supervisor for helping us with our queries and giving us advice on how to further improve our project. We would also like to thank Mrs. Anisha Baiju, the H.O.D of the computer science department of PRIVATE INTERNATIONAL ENGLISH SCHOOL for her candid advice all through the period of development of this project.

Our thanks also go to our lovely parents, siblings and friends for their moral and enthusiastic support. Above all, we would like to express our deepest gratitude to the Almighty God who by his grace and mercies we were able to successfully complete this project.

INDEX

SI. NO	CONTENT	PAGE NO.	DATE	SIGNATURE
1	Aim			
2	Overview of Python and SQL			
4	Requirements (Hardware and Software)			
5	Algorithm			
6	Source Code			
7	Output			
8	Conclusion			
9	Bibliography			

AIM OF THE PROJECT

This bookstore was created with two things in mind: simplicity and creativity. In the current world online bookstores are so popular with every book being available in paperback and digital format. This might lead to the obvious question---"Why a bookstore?". The answer is quite simple. It is because we had a problem when few of our books in the school library were stolen and we could not find it because they were not kept track of. This led to the urgent question of a library management system. The school started creating one and it gave us the idea to implement the same but in the form a bookstore.

This bookstore management system allows us to,

- Keep track of the different books along with stock in our INVENTORY TABLE
- Manage all purchases made by a customer and replenish/update our stock
- Generate a bill for our customer along with VAT(5% of Total Price)
- Give the customers a chance to add a book to a WISHLIST table and allow the owners to order those books in the next inventory refill.

The main purpose of this project is to manage the backend databases through MYSQL and use python to create an interactive and simple experience for our customers and fulfill their orders.

OVERVIEW OF PYTHON

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

OVERVIEW OF SQL

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

SQL is a language to operate databases; it includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an **ANSI** (American National Standards Institute) standard language, but there are many different versions of the SQL language.

REQUIREMENT ANALYSIS

Hardware Requirement

No special hardware requirements are needed to properly run the project for testing purposes. However, for optimum experience the following specification should be met:

CPU: Intel Core i3-3210 3.2 GHz / AMD A8-7600 APU 3.1 GHz or equivalent.

RAM: 4GB

Software Requirement

These software are necessary for the running of this project.:

Python

- Python 3.7.1

Pip modules

- mysql-connector-python 8.0.21

MySQL

- MySQL Shell 8.0.21
- MySQL Workbench 8.0

ALGORITHM

1. Start
2. Choose between admin and customer.
3. If you clicked on the admin button, you'll be given 2 further options which are inputting into tables and viewing tables. You can input into the Purchase table and the Inventory table. The functions called are inserttable1() and insertable2(). You can view the Purchase, Inventory and Wishlist table. The buttons here call the functions viewpurchase(), viewinventory() and viewwishlist().
4. If you clicked on the customer button you'll be able to view the Inventory table by calling viewinventory() using the button, add books to the wishlist by calling wishlist(), make a purchase by calling insertable2(), view your bill by calling bill() and making searches using search().
5. Result is shown depending on the choices made in a Tkinter built GUI with buttons for further choices.
6. Exit the program.

PROJECT SOURCE CODE

SQL SOURCE CODE:

FOR CREATING MAIN DATABASE ‘BOOKSTORE’:

```
create database bookstore;
```

USING DATABASE ‘BOOKSTORE’:

```
use bookstore;
```

CREATING TABLE INVENTORY:

```
create table inventory(sno int not null auto_increment, book_name varchar(120) not null, author varchar(120) not null, stock int, price1_aed float not null, unique(book_name), primary key(sno));
```

CREATING TABLE PURCHASE:

```
create table purchase(pid int not null auto_increment, DOP date, book_name varchar(120) not null, quantity int not null default 1, price1_aed float, total_aed float, primary key(pid), foreign key (book_name) references inventory(book_name));
```

CREATING TABLE WISHLIST:

```
create table wishlist(ISBN bigint not null, book_name varchar(120) not null, author varchar(120) not null, primary key(ISBN));
```

CREATING TABLE BILL:

```
create table bill (bid bigint not null auto_increment primary key, DOP datetime default CURRENT_TIMESTAMP, cname varchar(120) not null, quantity int default 1, totalprice int default 30, vat_5 float default 1.5);
```

INSERTING VALUES INTO INVENTORY

```
insert into inventory values (1,'Computer Science', 'Sumita Arora',5,200);
insert into inventory values(2,'Grade 12 Mathematics','RD Sharma',10,500);
insert into inventory values(3, 'Grade 12 PhysicS', 'HC Verma',15,300);
```

PYTHON SOURCE CODE

"""

Created on Wed Oct 21 15:28:53 2020

@author: sauravsoham

"""

""" BOOKSTORE MANAGEMENT SYSTEM IN PYTHON AND WITH
MYSQL DATABASE

FOR: CBSE PROJECT 2020-21

BY: SAURAV VINODH AND SOHAM VAKANI

DATE: 25TH OCTOBER 2020 """

#IMPORTING THE TWO MAIN MODULES TKINTER(for gui) and
MYSQL.CONNECTOR(for python-sql connectivity).

```
import tkinter as tk
```

```
import mysql.connector as sqlcon
```

#to create database BOOKSTORE IF IT DOESNT EXIST

```
mycon=sqlcon.connect(host='127.0.0.1',user='root',passwd='mysql')
```

```
cursor=mycon.cursor()
```

```
st="create database bookstore if not exists"
```

```
cursor.execute(st)
```

```
#to create tables INVENTORY,PURCHASE,WISHLIST,BILL

mycon=sqlcon.connect(host='127.0.0.1',user='root',passwd='mysql')
cursor=mycon.cursor()

table1="create table inventory if not exists (sno int not null
auto_increment, book_name varchar(120) not null, author
varchar(120) not null, stock int, price1_aed float not null,
unique(book_name), primary key(sno));"

table2="create table purchase if not exists (pid int not null
auto_increment, DOP date, book_name varchar(120) not null, quantity
int not null default 1, price1_aed float, total_aed float, primary
key(pid), foreign key (book_name) references
inventory(book_name));"

table3="create table wishlist if not exists (ISBN bigint not null,
book_name varchar(120) not null, author varchar(120) not null,
primary key(ISBN));"

table4="create table bill if not exists(bid bigint not null auto_increment
primary key, DOP datetime default CURRENT_TIMESTAMP, cname
varchar(120) not null,quantity int default 1, totalprice int default 30,
vat_5 float default 1.5) ;"

cursor.execute(table1)
cursor.execute(table2)
cursor.execute(table3)
cursor.execute(table4)
```

#to insert values into INVENTORY TABLE

```
mycon=mysqlcon.connect(host='127.0.0.1',user='root',passwd='mysql')
cursor=mycon.cursor()
insert1 ="insert into inventory values (1,'Computer Science', 'Sumita
Arora',5,200); "
insert2 ="insert into inventory values(2,'Grade 12 Mathematics','RD
Sharma',10,500);"
insert3 ="insert into inventory values(3, 'Grade 12 PhysicS', 'HC
Verma',15,300);"
cursor.execute(insert1)
cursor.execute(insert2)
cursor.execute(insert3)
```

#FUNCTION TO INSERT VALUES INPUTTED BY ADMIN INTO
INVENTORY TABLE

```
def inserttable1(name,author,stock,price):
    mycon=mysqlcon.connect(host='127.0.0.1',user='root',passwd='mysql',
                           database='bookstore')
    cursor=mycon.cursor()
    st="select * from inventory"
    cursor.execute(st)
    data=cursor.fetchall()
    for i in data:
```

```

if i[1]==name:

    st="update inventory set stock=stock+{} where
book_name='{}'".format(stock,name)

    print(st)

    cursor.execute(st)

    mycon.commit()

    break

else:

    st="insert into inventory(book_name,author,stock,price1_aed)
values('{}','{}',{},{})".format(name,author,stock,price)

    cursor.execute(st)

    mycon.commit()

```

#FUNCTION THAT WORKS WITH PURCHASE TABLE AND PERFORMS VARIOUS FUNCTIONS

```

def inserttable2(bookname,quantity):

mycon=mysqlcon.connect(host='127.0.0.1',user='root',passwd='mysql',
database='bookstore')

cursor=mycon.cursor()

st="insert into
purchase(book_name,quantity)values('{}',{})".format(bookname,quanti
ty)

cursor.execute(st)

```

```
mycon.commit()
```

```
#Inserting the price of 1 book into the 'Purchase' table from our  
'Inventory' table
```

```
st="update purchase set price1_aed=(select price1_aed from  
inventory where inventory.book_name='{}') where  
book_name='{}'".format(bookname,bookname)
```

```
cursor.execute(st)
```

```
mycon.commit()
```

```
#Setting the date of insertion as a default value for the date of  
purchase(DOP) column
```

```
st="update purchase set DOP=date(current_timestamp) where  
book_name='{}'".format(bookname)
```

```
cursor.execute(st)
```

```
mycon.commit()
```

```
#Setting the total column as the product of price*quantity
```

```
cursor.execute("select * from inventory")
```

```
data=cursor.fetchall()
```

```
for i in data:
```

```
    if i[1]==bookname:
```

```
    price_of_1=i[4]

    cursor.execute("select * from purchase")
    data=cursor.fetchall()
    pid=data[-1][0]
    st="update purchase set total_aed ={}*{} where
    pid={}".format(quantity,price_of_1,pid)
    cursor.execute(st)
    mycon.commit()
```

#Updating the stock in the 'Inventory' table by subtracting the quantity of books sold from the stock

```
cursor.execute("select * from inventory")
data=cursor.fetchall()

for i in data:
    if i[1]==bookname:
        stock=i[3]
        if (stock-quantity)>=0:
            st="update inventory set stock={} - {} where
            book_name='{}'".format(stock,quantity,bookname)
            cursor.execute(st)
            mycon.commit()
    else:
```

```
    print("You're claiming to have sold more books than you have  
currently stored in your inventory, There are", stock, bookname,  
"books in your inventory")
```

#Below are the functions available to the admin, this includes
inputting data into a tables and viewing tables

```
def adminb():  
    #FIRST WINDOW THAT IS DISPLAYED TO ADMIN.  
  
    window1 = tk.Tk()  
  
    greeting = tk.Label(window1, text="Choose between inputting data  
between a table and viewing a table")  
    greeting.pack(side=tk.TOP)  
  
    inputb=tk.Button(window1, text="Inputting Into  
Tables",command=insert1)#####  
    inputb.pack(side=tk.LEFT)  
  
    viewb=tk.Button(window1, text="Viewing  
Tables",command=viewadmin)  
    viewb.pack(side=tk.RIGHT)  
  
    window1.title("Bookstore Management System")  
    window1["background"]="#18C3F8"  
  
    #You can input data into PURCHASE TABLE
```

```
def insert1(): #  
    #WINDOW THAT DISPLAYS THE INPUTTING DATA TABLE  
  
    window2=tk.Tk()  
  
    greeting = tk.Label(window2, text="Choose between inputting data  
into the purchase table and the inventory table")  
    greeting.pack(side=tk.TOP)  
  
    purchaseb=tk.Button(window2, text="Purchase  
Table",command=datapurchase)#####  
    purchaseb.pack(side=tk.LEFT)  
  
    inventoryb=tk.Button(window2, text="Inventory  
Table",command=datainventory)  
    inventoryb.pack(side=tk.RIGHT)  
  
    window2.title("Bookstore Management System")  
    window2["background"]="#18C3F8"
```

#FUNCTION ALLOWING INPUTTING VALUES INTO INVENTORY

```
def datainventory():  
    def submit():  
        name=e1.get()  
        author=e2.get()  
        stock=int(e3.get())  
        priceof1=int(e4.get())
```

```
inserttable1(name,author,stock,priceof1)
```

#BASIC WINDOW FOR INPUTTING VALUES INTO INVENTORY

```
window3=tk.Tk()  
name1=tk.Label(window3, text="Name of the Book")  
name1.grid(row=0,column=0)  
author1=tk.Label(window3, text="Author")  
author1.grid(row=1,column=0)  
stock1=tk.Label(window3, text="Stock")  
stock1.grid(row=2,column=0)  
priceof1_=tk.Label(window3, text="Price of One Book")  
priceof1_.grid(row=3,column=0)  
e1=tk.Entry(master=window3)  
e1.grid(row=0,column=1)  
e2=tk.Entry(master=window3)  
e2.grid(row=1,column=1)  
e3=tk.Entry(master=window3)  
e3.grid(row=2,column=1)  
e4=tk.Entry(master=window3)  
e4.grid(row=3,column=1)  
get=tk.Button(window3, text="Submit", command=submit)  
get.grid(row=4,column=1)
```

```
window3.title("Bookstore Management System")
window3["background"]="#18C3F8"
```

#FUNCTION FOR INPUTTING VALUES INTO PURCHASE TABLE

```
def datapurchase():
    def submit():
        name=e1.get()
        qty=int(e2.get())
        inserttable2(name,qty)
```

#BASIC WINDOW FOR PURCHASE WINDOW

```
window4=tk.Tk()
name1=tk.Label(window4, text="Name of the Book")
name1.grid(row=0,column=0)
qty=tk.Label(window4, text='Quantity of Books Bought')
qty.grid(row=1,column=0)
e1=tk.Entry(master=window4)
e1.grid(row=0,column=1)
e2=tk.Entry(master=window4)
e2.grid(row=1,column=1)
get=tk.Button(window4, text="Submit", command=submit)
get.grid(row=2,column=1)
```

```
window4.title("Bookstore Management System")
window4["background"]="#18C3F8"
```

#The tables you can view as the admin are the purchase table, wishlist table and the wishlist table.

```
def viewadmin():
    #WINDOW THAT SHOWS VIEW TABLE FUNCTIONS FOR ADMIN

    window7=tk.Tk()
    purchase=tk.Button(window7, text="View the Purchase Table",command=viewpurchase)
    purchase.grid(row=0,column=0)
    inventory=tk.Button(window7, text="View the Inventory Table",command=viewinventory)
    inventory.grid(row=0, column=1)
    wishlist=tk.Button(window7, text="View the Wishlist Table", command=viewwishlist)
    wishlist.grid(row=0,column=2)
    window7.title("Bookstore Management System")
    window7["background"]="#18C3F8"
```

#The wishlist table exists so that as a admin you can check which all books are not in your inventory and then later add them.

```
def viewwishlist():

    #WINDOW THAT ALLOWS TO ADD BOOKS INTO WISHLIST OR
    VIEW WISHLIST

    window8=tk.Tk()

    mycon=sqlcon.connect(host='127.0.0.1',user='root',passwd='mysql',
    database='bookstore')

    cursor=mycon.cursor()

    cursor.execute("select * from wishlist")

    data=cursor.fetchall()

    rowcount=cursor.rowcount

    mycon.close()

    isbn=tk.Label(window8, text="ISBN")

    isbn.grid(row=0,column=0)

    name=tk.Label(window8, text="Name of the Book")

    name.grid(row=0,column=1)

    author=tk.Label(window8, text="Author")

    author.grid(row=0,column=2)

    window8.title("Bookstore Management System")

    window8["background"]="#18C3F8"

    for i in range(rowcount):

        for j in range(3):

            e=tk.Label(window8, text=str(data[i][j]))
```

```
e.grid(row=i+1,column=j)
```

#FUNCTION FOR VIEWING PURCHASE TABLE

```
def viewpurchase():
```

```
    # WINDOW THAT SHOWS BUTTON TO VIEW FROM PURCHASE  
    TABLE
```

```
window11=tk.Tk()
```

```
mycon=sqlcon.connect(host='127.0.0.1',user='root',passwd='mysql',  
database='bookstore')
```

```
cursor=mycon.cursor()
```

```
cursor.execute("SELECT * FROM PURCHASE")
```

```
data=cursor.fetchall()
```

```
rowcount=cursor.rowcount
```

```
mycon.close()
```

```
pid=tk.Label(window11, text="PID")
```

```
pid.grid(row=0, column=0)
```

```
dop=tk.Label(window11, text="DOP")
```

```
dop.grid(row=0, column=1)
```

```
name=tk.Label(window11, text="Name of the Book")
```

```
name.grid(row=0,column=2)
```

```
qty=tk.Label(window11, text="Quantity")
```

```
qty.grid(row=0,column=3)
price=tk.Label(window11, text="Price of One")
price.grid(row=0,column=4)
total=tk.Label(window11, text="Total Price")
total.grid(row=0,column=5)
window11.title("Bookstore Management System")
window11["background"]="#18C3F8"
for i in range(rowcount):
    for j in range(6):
        e=tk.Label(window11, text=str(data[i][j]))
        e.grid(row=i+1,column=j)
```

#FUNCTION FOR VIEWING INVENTORY TABLE

```
def viewinventory():
```

#WINDOW ALLOWS TO VIEW BOOKS FROM INVENTORY TABLE

```
window12=tk.Tk()
sno=tk.Label(window12,text="S.No.")
sno.grid(row=0,column=0)
bookname=tk.Label(window12, text="Name of the Book")
bookname.grid(row=0,column=1)
author=tk.Label(window12, text="Author")
```

```
author.grid(row=0,column=2)
stock=tk.Label(window12, text="Stock")
stock.grid(row=0,column=3)
price=tk.Label(window12, text="Price of One")
price.grid(row=0,column=4)

mycon=mysqlcon.connect(host='127.0.0.1',user='root',passwd='mysql',
database='bookstore')
cursor=mycon.cursor()
cursor.execute("SELECT * FROM INVENTORY")
rows=cursor.fetchall()
rowcount=cursor.rowcount
mycon.close()
window12.title("Bookstore Management System")
window12["background"]="#18C3F8"
for i in range(rowcount):
    for j in range(5):
        e=tk.Label(window12, text=str(rows[i][j]))
        e.grid(row=i+1,column=j)
```

#Below are the options available as the customer

#LABELS AND BUTTONS AVAILABLE TO CUSTOMER TO USE

```
def customerb():
```

```
global L  
  
#FIRST WINDOW THAT DISPLAYS WHEN CUSTOMER ENTERS  
BOOKSTORE
```

```
window5=tk.Tk()  
  
view=tk.Button(window5, text="Viewing  
Inventory",command=viewinventory)  
  
view.grid(row=0,column=0)  
  
wish=tk.Button(window5, text="Adding Books to  
Wishlist",command=wishlist)  
  
wish.grid(row=0,column=1)  
  
purchase1=tk.Button(window5, text="Make a  
Purchase",command=purchase)  
  
purchase1.grid(row=0,column=2)  
  
bill1=tk.Button(window5, text="View your Bill",command=bill)  
  
bill1.grid(row=0,column=3)  
  
search1=tk.Button(window5, text="Search",command=search)  
  
search1.grid(row=0,column=4)  
  
L=[]  
  
window5.title("Bookstore Management System")  
window5["background"]="#18C3F8"
```

```
#INSERT BOOKS INTO THE WISHLIST TABLE
```

```
def wishlist():
```

```
    def submit():
```

```
isbn=int(e3.get())
name=e1.get()
author=e2.get()

mycon=sqlcon.connect(host='127.0.0.1',user='root',passwd='mysql',
database='bookstore')

cursor=mycon.cursor()

st="insert into wishlist(ISBN, book_name,author)
values({},'{}','{}')".format(isbn,name,author)

cursor.execute(st)

mycon.commit()
```

#WINDOW THAT ALLOWS TO ADD BOOKS TO WISHLIST

```
window6=tk.Tk()
title=tk.Label(window6, text="Add the book to your Wishlist")
title.grid(row=0,column=1)
name1=tk.Label(window6, text="Name of the Book")
name1.grid(row=1,column=0)
author=tk.Label(window6, text="Author")
author.grid(row=2,column=0)
isbn=tk.Label(window6, text="ISBN Number")
isbn.grid(row=3,column=0)
e1=tk.Entry(master=window6)
```

```
e1.grid(row=1,column=1)
e2=tk.Entry(master=window6)
e2.grid(row=2,column=1)
e3=tk.Entry(master=window6)
e3.grid(row=3,column=1)
get=tk.Button(window6, text="Submit", command=submit)
get.grid(row=4,column=1)
window6.title("Bookstore Management System")
window6["background"]="#18C3F8"
```

FUNCTION ALLOWING CUSTOMER TO MAKE A PURCHASE BY ENTERING BOOKNAME AND QUANTITY

```
def purchase():
    def submit():
        global L
        name=e1.get()
        qty=int(e2.get())
        inserttable2(name,qty)
```

```
mycon=mysqlcon.connect(host='127.0.0.1',user='root',passwd='mysql',
database='bookstore')
cursor=mycon.cursor()
```

```
cursor.execute("select * from inventory")
purrows=cursor.fetchall()
for i in purrows:
    if i[1]==name:
        price_of_1=i[4]
window10=tk.Tk()
label=tk.Label(window10, text="The total price of your purchase
is {}".format(qty*price_of_1))
label.grid(row=1,column=1)
L.append((name,qty,price_of_1,qty*price_of_1))
window10.title("Bookstore Management System")
window10["background"]="#18C3F8"
# WINDOW THAT ALLOWS CUSTOMER TO PURCHASE BOOK
```

```
window9=tk.Tk()
name=tk.Label(window9, text="Name of the Book")
name.grid(row=0,column=0)
qty=tk.Label(window9, text="Quantity")
qty.grid(row=1,column=0)
e1=tk.Entry(master=window9)
e1.grid(row=0,column=1)
e2=tk.Entry(master=window9)
e2.grid(row=1,column=1)
get=tk.Button(window9, text="Submit", command=submit)
```

```
get.grid(row=2,column=1)
window9.title("Bookstore Management System")
window9["background"]="#18C3F8"

#FUNCTION RESPONSIBLE FOR DISPLAYING CUSTOMERS BILL
AFTER PURCHASE IS MADE

def bill():
    #WINDOW THAT SHOW THE TOTAL BILL

    window13=tk.Tk()
    global bill_id
    global L
    label3=tk.Label(window20, text="Bill ID: {}".format(bill_id))
    label3.pack()
    bill_id+=1
    total=0
    for i in L:
        total+=i[3]
    for j in L:
        label=tk.Label(window13, text="{} {} {} ".format(j[0],j[1], j[3]))
        label.pack()
```

```
label2=tk.Label(window13, text="Your total comes out to  
{}".format(total))  
label2.pack()  
window13.title("Bookstore Management System")  
window13["background"]="#18C3F8"
```

#FUNCTION THAT SEARCHES FOR BOOKS BASED ON
SNO/NAME/AUTHOR

```
def search():  
    def submit():  
        name=e1.get()  
        sno=e2.get()  
        author=e3.get()  
  
        mycon=mysqlcon.connect(host='127.0.0.1',user='root',passwd='mysql',  
database='bookstore')  
        cursor=mycon.cursor()  
        if sno and not name and not author:  
            cursor.execute("SELECT * FROM inventory WHERE  
sno={}".format(sno))  
            data=cursor.fetchall()  
            row=cursor.rowcount  
  
        elif name and not sno and not author:
```

```
        cursor.execute("SELECT * FROM inventory WHERE  
book_name='{}'".format(name))  
  
        data=cursor.fetchall()  
  
        row=cursor.rowcount  
  
  
    elif author and not sno and not name:  
  
        cursor.execute("SELECT * FROM inventory WHERE  
author='{}'".format(author))  
  
        data=cursor.fetchall()  
  
        row=cursor.rowcount  
  
  
    else:  
  
        cursor.execute("SELECT * FROM inventory WHERE sno='{}'  
AND book_name='{}' AND author='{}'".format(sno,name,author))  
  
        data=cursor.fetchall()  
  
        row=cursor.rowcount
```

#WINDOW THAT ALLOWES TO SEARCH FOR BOOKS BASED
ON CERTAIN CRITERIA

```
window15=tk.Tk()  
  
sno=tk.Label(window15,text="S.No.")  
sno.grid(row=0,column=0)  
  
bookname=tk.Label(window15, text="Name of the Book")  
bookname.grid(row=0,column=1)
```

```
author=tk.Label(window15, text="Author")
author.grid(row=0,column=2)
stock=tk.Label(window15, text="Stock")
stock.grid(row=0,column=3)
price=tk.Label(window15, text="Price of One")
price.grid(row=0,column=4)
window15.title("Bookstore Management System")
window15["background"]="#18C3F8"
for i in range(row):
    for j in range(5):
        e=tk.Label(window15, text=str(data[i][j]))
        e.grid(row=i+1,column=j)
```

```
#WINDOW THAT ALLOWS TO INPUT BOOKS
window14=tk.Tk()
name=tk.Label(window14, text="Name of the Book")
name.grid(row=0,column=0)
sno=tk.Label(window14, text="S.No.")
sno.grid(row=1,column=0)
author=tk.Label(window14, text="Author")
author.grid(row=2,column=0)
e1=tk.Entry(master=window14)
e1.grid(row=0,column=1)
```

```
e2=tk.Entry(master=window14)
e2.grid(row=1,column=1)
e3=tk.Entry(master=window14)
e3.grid(row=2,column=1)
get=tk.Button(window14,
text="Submit",bg="red",command=submit)
get.grid(row=3,column=1)
window14.title("Bookstore Management System")
window14["background"]="#18C3F8"
```

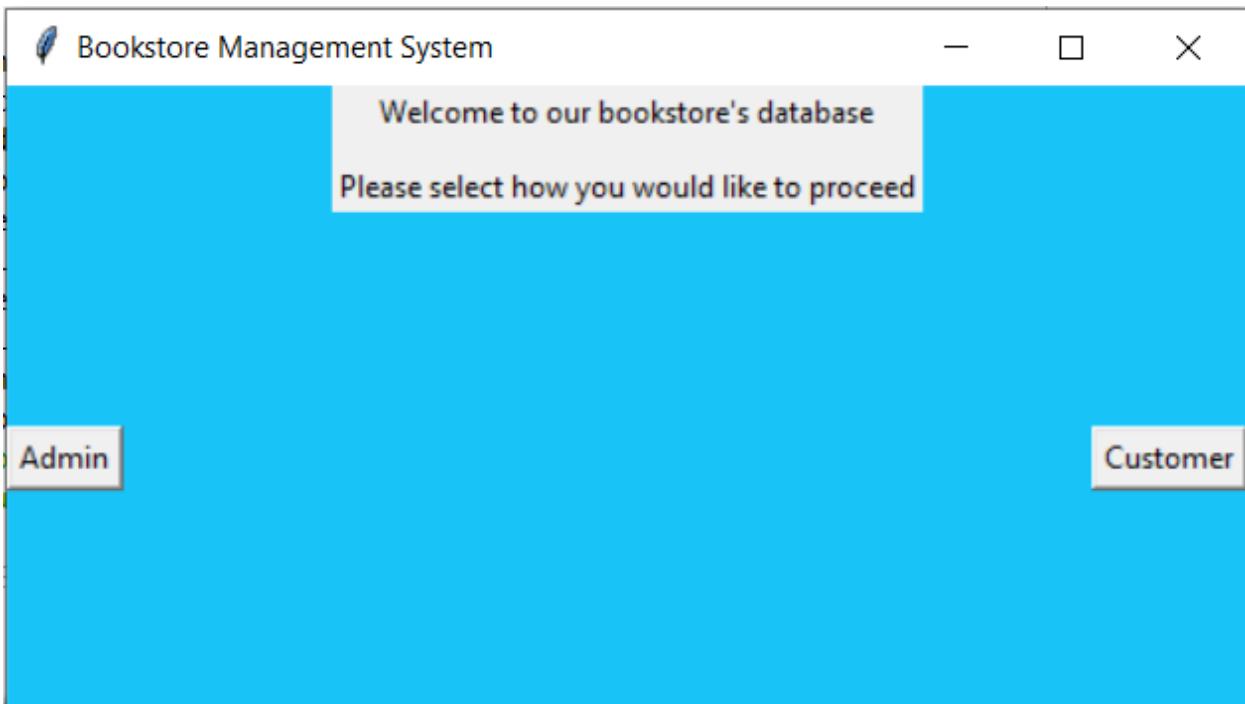
#FIRST WINDOW THAT DISPLAYS WHEN THE PROGRAM IS RUN

```
window = tk.Tk()
window.geometry("500x250")
greeting = tk.Label(text="Welcome to our bookstore's
database\n\nPlease select how you would like to proceed")
greeting.pack(side=tk.TOP)
adminb1=tk.Button(text="Admin",command=adminb)#####
adminb1.pack(side=tk.LEFT)
```

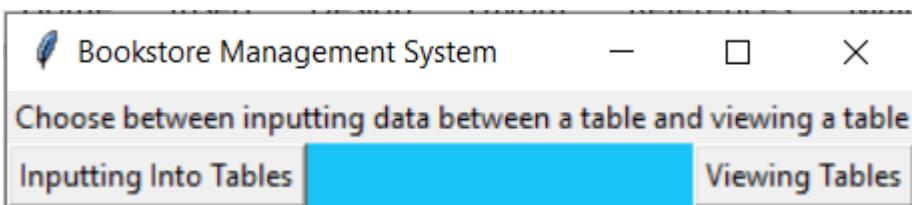
```
customerb1=tk.Button(text="Customer",command=customerb)
customerb1.pack(side=tk.RIGHT)
window.title("Bookstore Management System")
window["background"]="#18C3F8"
window.mainloop()
L=[]
bill_id= 1
```

OUTPUT

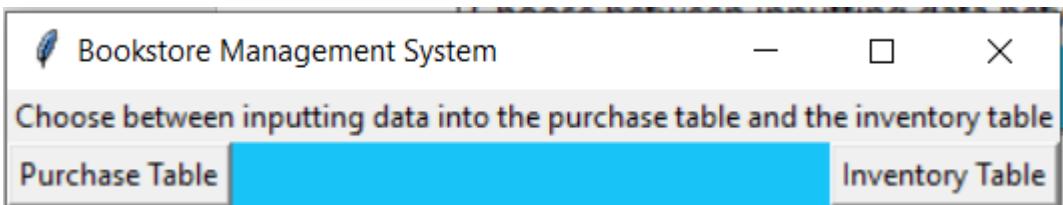
MAIN WINDOW:



IF YOU SELECT ADMIN:



IF YOU SELECT 'INPUTTING INTO TABLES'.



IF YOU SELECT ‘PURCHASE TABLE’ YOU’LL BE PRESENTED WITH TWO ENTRY BOXES TO INPUT THE NAME OF THE BOOK AND THE QUANTITY OF THE BOOK BEING BOUGHT.

Name of the Book	Computer Science
Quantity of Books Bought	2
Submit	

HIT ‘SUBMIT’.

IF YOU WANT TO ADD BOOKS TO THE INVENTORY GO BACK TO THE WINDOW SHOWN BELOW AND CHOOSE ‘INVENTORY TABLE’.

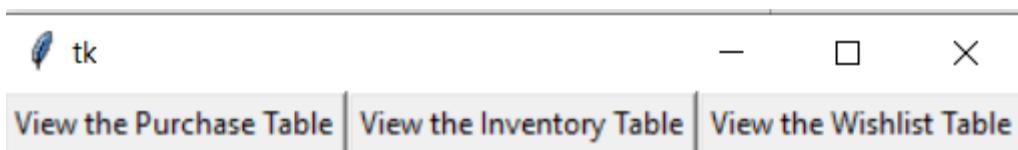
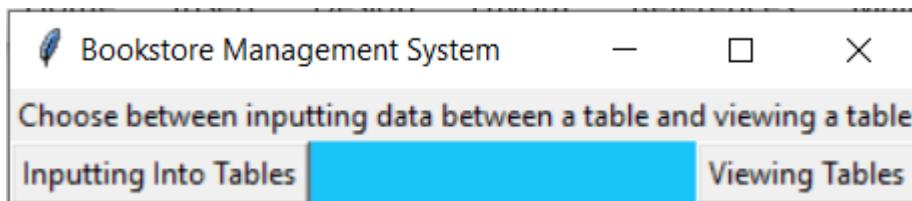
Choose between inputting data into the purchase table and the inventory table

Purchase Table Inventory Table

Name of the Book	Harry Potter
Author	J.K Rowling
Stock	20
Price of One Book	150
Submit	

AFTER HITTING ‘SUBMIT’. LET US NOW VIEW THE ADDITIONS WE HAVE MADE INTO THE PURCHASE TABLE AND THE INVENTORY TABLE.

GO BACK TO THE WINDOW SHOWN BELOW AND SELECT ‘VIEWING TABLES’



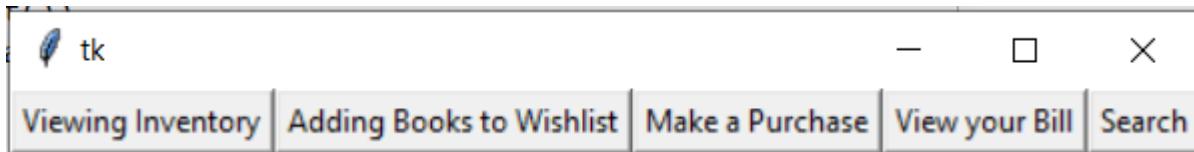
LET US FIRST VIEW THE PURCHASE TABLE

PID	DOP	Name of the Book	Quantity	Price of One	Total Price
49	2020-10-25	Computer Science	2	200.0	400.0

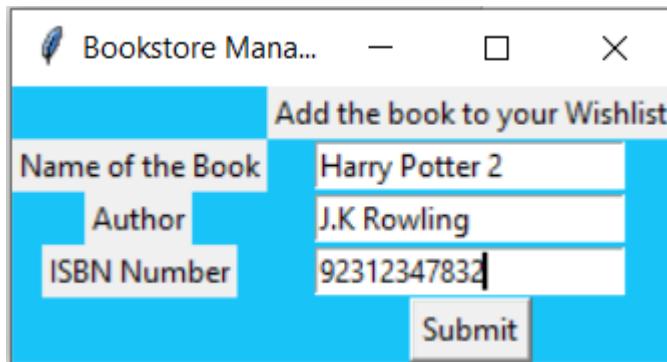
LET US NOW VIEW THE INVENTORY TABLE

S.No.	Name of the Book	Author	Stock	Price of One
1	Computer Science	Sumita Arora	3	200.0
2	Grade 12 Mathematics	RD Sharma	10	500.0
3	Grade 12 Physics	HC Verma	15	300.0
20	Harry Potter	J.K Rowling	20	150.0

WE SHALL COME BACK TO THE WISH LIST TABLE LATER. LET US INSTEAD SEE THE OPTIONS AVAILABLE WHEN YOU RUN THE PROGRAM AS A CUSTOMER.



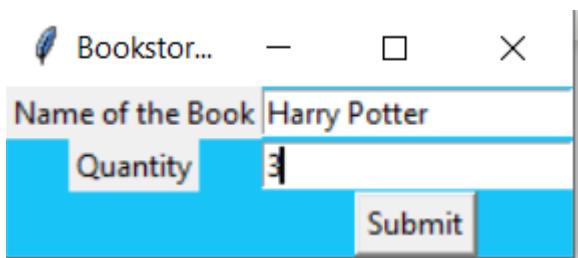
WE CAN VIEW THE INVENTORY TO SEE WHICH ALL BOOKS ARE AVAILABLE FOR THE CUSTOMER TO BUY. IF A CERTAIN BOOK IS NOT AVAILABLE, WE CAN ADD IT TO THE WISH LIST. THIS HELPS THE ADMIN CHECK WHAT ALL BOOKS THEY NEED IN THEIR STOCKS.



LET US NOW VIEW THIS ENTRY AS THE ADMIN. BY CLICKING ON THE 'VIEW THE WISHLIST TABLE'

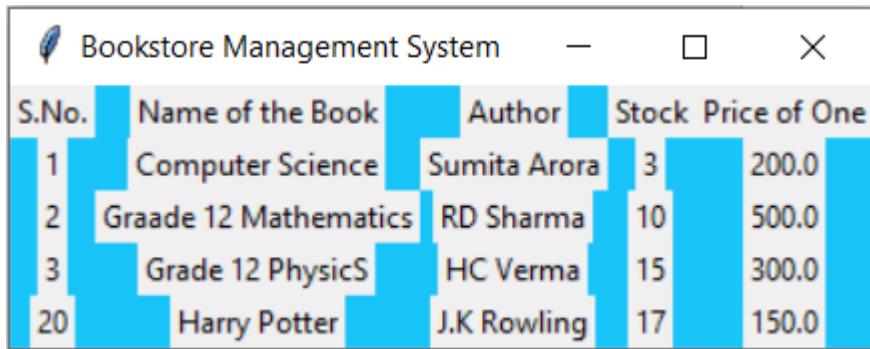
ISBN	Name of the Book	Author
992923932	wack	kichu
1232412414	flufluf	afaszfa
92312347832	Harry Potter 2	J.K Rowling
93121234143	Harry Potter 5	J.K. Rowling

NOW LET'S GO BACK TO MAKING PURCHASES BY HITTING 'MAKE A PURCHASE' BUTTON.



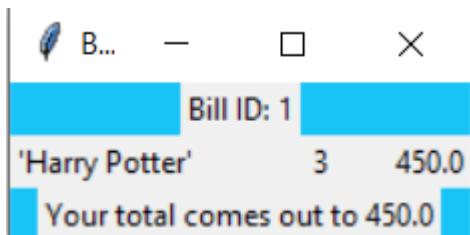
A screenshot of a web-based purchase form titled "Bookstor...". It has fields for "Name of the Book" (Harry Potter), "Quantity" (3), and a "Submit" button. The background is light blue.

ONCE YOU HIT SUBMIT THIS QUANTITY IS SUBTRACTED FROM OUR INVENTORY. LET US CHECK THE INVENTORY AGAIN TO SEE THIS.



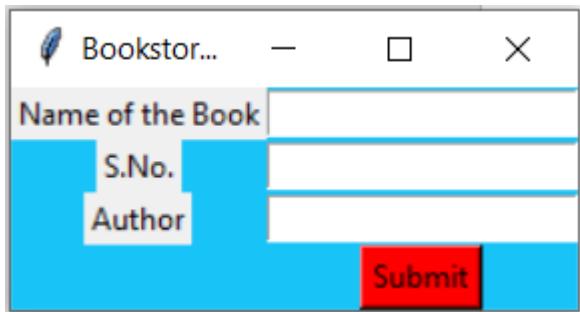
S.No.	Name of the Book	Author	Stock	Price of One
1	Computer Science	Sumita Arora	3	200.0
2	Grade 12 Mathematics	RD Sharma	10	500.0
3	Grade 12 Physics	HC Verma	15	300.0
20	Harry Potter	J.K Rowling	17	150.0

LET US SEE OUR BILL BY CLICKING ON 'VIEW YOUR BILL'



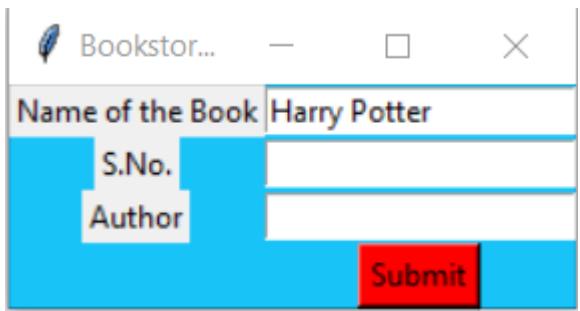
A screenshot of a bill summary titled "B...". It shows "Bill ID: 1" and a single item: "'Harry Potter' 3 450.0". Below this, it states "Your total comes out to 450.0". The background is light blue.

LET US NOW SEARCH FOR ITEMS IN THE INVENTORY BY HITTING THE ‘SEARCH BUTTON’

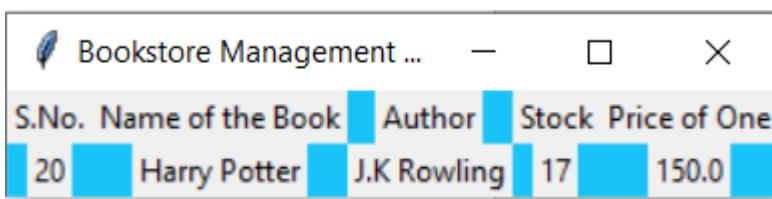


A screenshot of a search interface titled "Bookstor...". It features three input fields: "Name of the Book", "S.No.", and "Author", each with a blue background and white text. A red "Submit" button is located at the bottom right.

IF WE ENTER EITHER THE NAME OF THE BOOK, THE SERIAL NUMBER, THE AUTHOR OR ALL THREE, WE ARE ABLE TO SEE ITS DATA FROM THE INVENTORY TABLE.



A screenshot of the search interface with the "Name of the Book" field populated with "Harry Potter". The other fields ("S.No." and "Author") are empty. The "Submit" button is red.



S.No.	Name of the Book	Author	Stock	Price of One
20	Harry Potter	J.K Rowling	17	150.0

CONCLUSION

The program “BOOKSTORE MANAGEMENT SYSTEM” has been successfully brought to completion. With this program the administrator can add or remove books from INVENTORY TABLE,check the purchases from PURCHASE TABLE,check the books that customers want added,find the total sale and other features.

The customer can view all the books from the INVENTORY TABLE, make purchases by entering book name and quantity, can add books to WISHLIST TABLE, and view bill.

We wish to incorporate even better GUI and make the interface more user-friendly. We wish to find a better system wherein the program itself can recommend books based on machine learning and also allow users to rate books.

BIBLIOGRAPHY

- O'Brien, A. J. (2011). Management Information Systems. New York: NY publishers.
- Riggins, F. (1999). A framework for identifying web based electronic commerce opportunity. Journal of organisational computing and electronic commerce, 45-46.
- https://www.academia.edu/35851704/COMPUTER_SCIENCE_INVESTIGATORY_PROJECT
- <https://cbseportal.com/Project/CBSE-Class-XII-Computer-Science-Project-File-on-Book-Shop-Cpp>
- COMPUTER SCIENCE WITH PYTHON—CLASS 11—SUMITA ARORA
- Muzumdar, P. (2012). Online bookstore- A new trend in textbook sales management for services marketing. Journal of management and marketing research, 5.
- COMPUTER SCIENCE WITH PYTHON—CLASS 12—SUMITA ARORA