**System Implementations**

Recommended System Requirements

Processors: Intel® Core™ i3 processor 4300M at 2.60 GHz.

Disk space: 4 to 8 GB.

Operating systems: Windows® 10, MACOS, and UBUNTU.

Python Versions: 3.X.X or Higher.

Minimum System Requirements

Processors: Intel Atom® processor or Intel® Core™ i3 processor.

Disk space: 1 GB.

Operating systems: Windows 7 or later, MACOS, and UBUNTU.

Python Versions: 2.7.X, 3.9.X.

**ACKNOWLEDGEMENT**TTT

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

We would like to express my deep and sincere gratitude to my subject teacher, **Mr. Amit Udiwal**, for giving me the opportunity to do research and providing invaluable guidance throughout this research. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the research and to present the research works as clearly as and honour to work and study under his guidance. We are very much thankful to our **Sr. Jasmin** for giving valuable time and moral support to develop this software. We would like to take opportunity to extend my sincere thanks and gratitude to our parents for being a source of inspiration and providing time and freedom to develop this software project. We also feel indebted to my friends for the valuable suggestions during the project work.

Gaytri Yadav

[Roll No.

Class XII

**CERTIFICATE**

This is to certify that the project on ‘Book Store Management System’ is a work done by Gaytri Yadav fulfilment of CBSE’S AISSCE EXAMINATION 2022-23 and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form any other examination and does not form any other course undergone by the candidate.

Name: Gaytri Yadav [Roll No.

………………….

Signature of Teacher / Guide

Name: Mr. Amit Udiwal

Designation:

………………. ….………………

**REFERENCE**

The order to work on this project on ‘Book Store Management System’ the following books & literature are referred by me during the various phrases of department of the project.

• http://www.python.org/.

• http://www.itsourcecode.org/.

• http://www.wikipedia.org/.

• Informatics Practices for Class XII

- By Sumita Arora

• Together with informatics practices.

Other than the above mentioned books, the suggestions and supervision of my teacher and my class experience also helped me to develop this software project.

**Introduction**

This project Book Store Management system includes some facilities for the retail- bookshop to maintain records of the books and also search, display, modification, delete etc the books available. This software searches the books data which is store in the record The software used for small shops for maintaining their records related to books selling, printing the reports and cost savings Book distributors transaction handling is one of the complex process and it required computerized system to maintain overall transactions in an easier manner. Due to heavy demand of books in these competitive worlds, data increase so muck Books shop required up to date information about the customer who

**Objective and**

**Scope of The Project**

The Book Store Management System is a simple project developed using python. This project is an interesting project. The user can add the number of book details and you can see the details stored in the list form. The user can delete the list items if he/she wants to remove it. The project file contains a python script. This is a simple GUI-base project which is very easy to understand and use.

The Book Store project is simply in python. Taking about the features of this system, the user can make the list of books with their authors, year, and keep them as records. You just have to type the book information in the text fields and click on the add button to add the information on the record.

**Book Store Management System**

import pymysql as cntr , datetime as \_\_dt , matplotlib.pyplot as plt

from random import shuffle

from tempfile import mktemp

from os import system , startfile

\_\_db = cntr.connect(host = 'localhost' , user = 'root' , passwd = 'manager' , database = 'book\_shop')

\_\_cur = \_\_db.cursor()

\_\_db.autocommit(True)

#Function to check is it leap year

is\_leapyear = lambda year : year % 4 == 0

#Function to get last date of month

def last\_month(month , year):

if month in (1,3,5,7,8,10,12) : return 31

elif month == 2 and is\_leapyear(year) : return 29

elif month == 2 : return 28

else : return 30

clrscreen = lambda : system("cls")

def view\_stock() :

\_\_cur.execute("select Book\_No , Book\_Name , Available\_Stock from stock")

data = \_\_cur.fetchall()

print("Book Number\tBook Name\tStock")

for row in data : print(row[0] , '\t\t' , row[1] , '\t' , row[2])

def add\_stock() :

print('Add Stock'.center(89 , '='))

bno = unique\_book\_no()

if bno :

print("Book Number : " , bno)

else : bno = int(input("Enter book number : "))

bname = input("Enter the Book\'s Name : ")

auth = input("Enter the Author of the Book : ")

publ = input("Enter the Publisher of the Book : ")

cost = eval(input("Enter the Cost per Book : "))

stock = int(input("Enter the Quantity purchased : "))

\_\_cur.execute("insert into stock values ({} , '{}' , '{}' , '{}' , {} , {} , {} , '{}')".format(bno , bname , auth , publ , cost , stock , 0, \_\_dt.date.today()))

print("Inserted Sucessfully !!!")

def add\_user() :

user = input("Enter the user name : ")

passwd = input("Enter a Password : ")

passwd2 = input("Enter Password to confirm : ")

if passwd == passwd2 :

\_\_cur.execute("insert into users values('{}' , '{}')".format(user , passwd))

print("Created Successfully!!!")

elif passwd != passwd2 : print("You've entered different passwords")

def sell\_book() :

print('Purchase')

cname = input("Enter the Customer Name : ")

phno = int(input("Enter the phone number : "))

bno = int(input("Enter book number : "))

bname = input("Enter the name of the book : ")

cost = eval(input("Enter the cost of the book : "))

\_\_cur.execute("insert into purchased values({} , '{}')".format(bno , \_\_dt.date.today()))

\_\_cur.execute("update stock set qty\_purchased = qty\_purchased + 1 where Book\_No = {}".format(bno))

\_\_cur.execute("update stock set Available\_Stock = Available\_Stock - 1 where Book\_No = {}".format(bno))

print("Bought Successfully")

q = '''Book Shop\nName : {}\nPhone No : {}\nBook Number : {}\nBook Name : {}\nCost : {}\nDate Of Purchase : {}'''.format(cname , phno , bno , bname , cost , \_\_dt.date.today())

filename = mktemp('.txt')

open(filename , 'w').write(q)

startfile(filename , 'print')

\_\_cur.execute('select Book\_Name , Book\_No , Author from stock where Available\_Stock = 0')

if \_\_cur.rowcount == 1 :

print("STOCK OF ")

print("Book Name : " , \_\_cur.fetchall()[0][0])

print("Book Number : " , \_\_cur.fetchall()[0][1])

print("Author : " , \_\_cur.fetchall()[0][2])

print("EXHAUSTED")

\_\_cur.execute('delete from stock where Available\_Stock = 0')

def unique\_book\_no () :

\_\_cur.execute("select max(Book\_No) from stock")

data = \_\_cur.fetchall()

if bool(data[0][0]) :

L1 = [x for x in range((data[0][0] + 1) , (data[0][0] + 10000))]

shuffle(L1)

return L1.pop(0)

else : return False

def view\_sales () :

print('Overall Sales This Month')

\_\_cur.execute("select distinct(s.Book\_Name) , s.qty\_purchased from stock s , purchased p where s.Book\_No = p.Book\_No and p.purchased\_on between '{year}-{month}-01' and '{year}-{month}-{date}'".format(year = \_\_dt.date.today().year , month = \_\_dt.date.today().month , date = last\_month(\_\_dt.date.today().month , \_\_dt.date.today().year)))

data = \_\_cur.fetchall()

L1 , L2 = [] , []

for row in data :

L1.append(row[0])

L2.append(row[1])

plt.bar(L1 , L2)

plt.xlabel('Books')

plt.ylabel('Sales')

plt.title('Sales')

plt.show()

def login():

user = input("Enter the username : ")

pwd = input("Enter the password : ")

\_\_cur.execute("Select \* from users where (username = '{}' and password = '{}')".format(user , pwd))

if \_\_cur.rowcount : return True

def update\_stock() :

bno = int(input("Enter the book number : "))

\_\_cur.execute("select Book\_Name , Available\_Stock from stock where Book\_No = {}".format(bno))

data = \_\_cur.fetchall()

print("Book Name : " , data[0][0])

print("Available Stock : " , data[0][1])

stock = int(input("Enter the new stock purchased : "))

\_\_cur.execute("update stock set Available\_Stock = Available\_Stock + {}".format(stock))

print("Updated Successfully")

====================================================================

# PYTHON MODULE : **Tables\_in\_mysql**

import pymysql as cntr

db = cntr.connect(host = 'localhost' , user = 'root' , passwd = 'manager')

db.autocommit(True)

cur = db.cursor()

cur.execute("create database if not exists book\_shop")

cur.execute("use book\_shop")

cur.execute("create table stock\

(Book\_No bigint primary key,\

Book\_Name varchar(255),\

Author varchar(255),\

Publisher varchar(255),\

Cost\_per\_Book float,\

Available\_Stock bigint,\

qty\_purchased bigint,\

purchased\_on date)")

cur.execute("create table users(username varchar(255) , password varchar(255) , check (username <> 'ADMIN'))")

cur.execute("create table purchased (Book\_no bigint , purchased\_on date , foreign key(Book\_no) references stock(Book\_No))")

cur.execute("create unique index Book\_Index on stock(Book\_No)")

cur.execute("insert into users values('admin' , 'admin@123')")

print("Database and Tables created successfully")

c = input("Press any key to continue---->")

cur.close()

db.close()

====================================================================

# PYTHON MODULE : **main**

import Book

c = 'y'

while c.lower() == 'y' :

print("Book Shop Management".center(89 , '='))

print('1. Register')

print('2. Login')

print('3. Exit')

choice4 = int(input("Enter the serial number of your choice : "))

if choice4 == 1 :

Book.clrscreen()

Book.add\_user()

elif choice4 == 2 :

Book.clrscreen()

if Book.login() :

Book.clrscreen

C = 'y'

while C.lower() == 'y' :

Book.clrscreen()

print("Book Shop Management".center(89 , '='))

print("1. Book Stock")

print("2. Book Selling")

print("3. Exit")

choice = int(input("Enter the serial number of your choice : "))

if choice == 1 :

Book.clrscreen()

print("Book Book".center(89 , '='))

print("1. Add a new Stock")

print("2. View all Stock")

print("3. Update an existing Stock")

print("4. Exit")

choice2 = int(input("Enter the choice : "))

if choice2 == 1 :

Book.clrscreen()

Book.add\_stock()

elif choice2 == 2 :

Book.clrscreen()

Book.view\_stock()

elif choice2 == 3 :

Book.clrscreen()

Book.update\_stock()

elif choice2 == 4 :

print("Good Bye")

break

else : print("INVALID CHOICE")

elif choice == 2 :

Book.clrscreen()

print('Book Selling'.center(89 , '='))

print('1. Sell a book')

print('2. View Sales this month')

print("3. Exit")

choice3 = int(input("Enter your choice : "))

if choice3 == 1 :

Book.clrscreen()

Book.sell\_book()

elif choice3 == 2 :

Book.clrscreen()

Book.view\_sales()

elif choice3 == 3 :

print("Good Bye")

break

else : print("INVALID CHOICE")

elif choice == 3 :

print("Good Bye")

break

else : print("INVALID CHOICE")

C = input("Do you want to continue (y/[n]) : ")

else : print("Good Bye")

else :

print("Either your username or password is incorrect")

elif choice4 == 3 :

print("Good Bye")

break

else : print("INVALID CHOICE")

c = input("Do you want to return to main menu (y/[n]) : ")

else : print("Good Bye")