Mini-Project on LIBRARY MANAGEMENT SYSTEM

TITLE: LIBRARY MANAGEMENT SYSTEM

OBJECTIVE:

The main **objective** of the **Python Project** on **Library Management System** is to manage the details of Member, Books, Student, Id, Issues, Name, Date Borrowed, Date Due. It manages all the information about Member, Librarian, Name, Date Borrowed, Date Due,Id.

INTRODUCTION:

The project titled Library Management System is
Library Management software for monitoring and controlling the transactions in a library .The project "Library Management System" is developed in php, which mainly focuses on basic operations in a library like adding new books, and updating new information, searching books and members and return books.

This project of "LIBRARY MANAGEMENT" of gives us the complete information about the library. We can enter the record of new books and retrieve the details of books available in the library. We can issue the books to the students and maintain their records and can also check how many books are issued and stock available in the library. In this project we can maintain the late fine of students who returns the issued books after the due date.

Throughout the project the focus has been on presenting information and comments in an easy and intelligible manner. The project is very useful for those who want to know about Library Management System.

SOFTWARE:

FRONT END   :   PYTHON IDLE 3.6
OS   :   WINDOWS 10

CODE:

```python
from tkinter import*

from tkinter import ttk

import random

from datetime import datetime

import tkinter.messagebox

class Library:

    def __init__(self,root):

        self.root = root

        self.root.title("Library management systems")

        self.root.geometry("1350x750+0+0")

        self.root.configure(background='powder blue')

        MType=StringVar()

        BookID=StringVar()

        BookTitle=StringVar()

        Name=StringVar()

        DateBorrowed=StringVar()

        ID=StringVar()

        DateDue=StringVar()

        def iDelete():

            iReset()

            self.txtDisplayR.delete("1.0",END)

            self.txtFrameDetail.delete("1.0",END)

        def iReset():

            MType.set("")

            BookID.set("")
```

```python
        BookTitle.set("")

        Name.set("")

        DateBorrowed.set("")

        ID.set("")

        DateDue.set("")

        self.txtDisplayR.delete("1.0",END)

    def iExit():

        iExit=tkinter.messagebox.askyesno ("Library Management System","Confirm If You Want
To Exit")

        if iExit>0:

            root.destroy()

            return

    def iDisplayData():
        self.txtFrameDetail.insert(END,"\t\t"+MType.get()+"\t\t"+Name.get()+"\t"+ID.get()+"\t\
        t"+BookTitle.get()+"\t\t"+BookID.get()+"\t\t\t"+DateBorrowed.get()+"\t\t"+
        DateDue.get()+"\n")

    def iReceipt():

        self.txtDisplayR.insert(END,'Member Type: \t\t'+ MType.get()+"\n")

        self.txtDisplayR.insert(END,'Book ID: \t\t'+BookID.get()+"\n")

        self.txtDisplayR.insert(END,'BookTitle: \t\t'+BookTitle.get()+"\n")

        self.txtDisplayR.insert(END,'Name: \t\t'+Name.get()+"\n")

        self.txtDisplayR.insert(END,'ID: \t\t'+ID.get()+"\n")

        self.txtDisplayR.insert(END,'Date Borrowed : \t\t'+DateBorrowed.get()+"\n")

        self.txtDisplayR.insert(END,'Date Due: \t\t'+DateDue.get()+"\n")

        #Frame

    MainFrame = Frame(self.root)

    MainFrame.grid()

    TitleFrame = Frame(MainFrame, width=1350, padx=20, bd=20, relief=RIDGE)

    TitleFrame.pack(side=TOP)
```

```python
        self.lblTitle=Label(TitleFrame,width=39,font=('arial',40,'bold'),text="\tLibrary
Management System\t",padx=12)

        self.lblTitle.grid()


        ButtonFrame=Frame(MainFrame, bd=20, width=1350, height=50, padx=20, relief=RIDGE)

        ButtonFrame.pack(side=BOTTOM)

        FrameDetail=Frame(MainFrame, bd=20, width=1350, height=100, padx=20, relief=RIDGE)

        FrameDetail.pack(side=BOTTOM)

        DataFrame=Frame(MainFrame, bd=20, width=1300, height=400, padx=20, relief=RIDGE)

    DataFrame.pack(side=BOTTOM)

        DataFrameLEFT = LabelFrame(DataFrame, bd=10, width=800, height=300, padx=20,
relief=RIDGE,font=('arial',12,'bold'),text="Library Membership Info:",)

        DataFrameLEFT.pack(side=LEFT)

        DataFrameRIGHT = LabelFrame(DataFrame, bd=10, width=450, height=300, padx=20,
relief=RIDGE,font=('arial',12,'bold'),text="Book Details:",)

        DataFrameRIGHT.pack(side=RIGHT)

        #Widget

        self.lblMemberType=Label(DataFrameLEFT,font=('arial',12,'bold'),text="Member
Type:",padx=2,pady=2)

        self.lblMemberType.grid(row=0,column=0,sticky=W)

        self.cboMemberType=ttk.Combobox(DataFrameLEFT,font=('arial',12,'bold'),state='read
only',textvariable=MType,width=23)

        self.cboMemberType['value']=('','Student','Lecture','Admin Staff')

        self.cboMemberType.current(0)

        self.cboMemberType.grid(row=0,column=1)

        self.lblBookID=Label(DataFrameLEFT,font=('arial',12,'bold'),text="Book
ID:",padx=2,pady=2)

        self.lblBookID.grid(row=3,column=0,sticky=W)

        self.txtBookID=Entry(DataFrameLEFT,     font=('arial',12,'bold'),textvariable    =    BookID
,width=25)
```

```python
        self.txtBookID.grid(row=3,column=1)

        self.lblBookTitle=Label(DataFrameLEFT,font=('arial',12,'bold'),text="Book
Title:",padx=2,pady=2)

        self.lblBookTitle.grid(row=3,column=2,sticky=W)

        self.txtBookTitle=Entry(DataFrameLEFT,        font=('arial',12,'bold'),        textvariable        =
BookTitle,width=25)

        self.txtBookTitle.grid(row=3,column=3)

        self.lblName=Label(DataFrameLEFT, font=('arial',12,'bold'),text="Name:",padx=2,pady=2)

        self.lblName.grid(row=1,column=0,sticky=W)

        self.txtName=Entry(DataFrameLEFT,font=('arial',12,'bold'),textvariable = Name,width=25)

        self.txtName.grid(row=1,column=1)

        self.lblIID= Label(DataFrameLEFT, font=('arial',12,'bold'),text="ID:",padx=2,pady=2)

        self.lblIID.grid(row=2,column=0,sticky=W)

        self.txtID = Entry(DataFrameLEFT, font=('arial',12,'bold'),textvariable= ID,width=25)

        self.txtID.grid(row=2,column=1)

        self.lblDateBorrowed=Label(DataFrameLEFT,
font=('arial',12,'bold'),text="DateBorrowed:",padx=2,pady=2)

        self.lblDateBorrowed.grid(row=1,column=2,sticky=W)

        self.txtDateBorrowed=Entry(DataFrameLEFT,font=('arial',12,'bold'),textvariable=
DateBorrowed ,width=25)

        self.txtDateBorrowed.grid(row=1,column=3)

        self.lblDateDue=Label(DataFrameLEFT,
font=('arial',12,'bold'),text="DateDue:",padx=2,pady=2)

        self.lblDateDue.grid(row=2,column=2,sticky=W)

        self.txtDateDue=Entry(DataFrameLEFT,font=('arial',12,'bold'),textvariable=
DateDue,width=25)

        self.txtDateDue.grid(row=2,column=3)
```

```python
#Widget
        self.txtDisplayR=Text(DataFrameRIGHT,font=('arial',12,'bold'),width=32,height=13,padx=
8,pady=20)

        self.txtDisplayR.grid(row=0,column=2)

        scrollbar = Scrollbar(DataFrameRIGHT)

        scrollbar.grid(row=0,column=3,sticky='ns')


    #ListBox

      scrollbar = Scrollbar(DataFrameRIGHT)

      scrollbar.grid(row=0,column=1,sticky='ns')

      ListOfBooks  =  ['AM-IV','Python','AOA','Operating  System','Computer  Graphics','Data
Structure']

      def SelectedBook(evt):

        value = str(booklist.get(booklist.curselection()))

        w = value

        if(w=="AM-IV"):

         BookID.set("LIB 74547")

         BookTitle.set("Kumbhojkar")

         iReceipt()

         import datetime

         d1=datetime.date.today()

         d2=datetime.timedelta(days=7)

         d3=(d1+d2)

         DateBorrowed.set(d1)

         DateDue.set(d3)


        if(w=="Python"):

         BookID.set("LIB 74548")
```
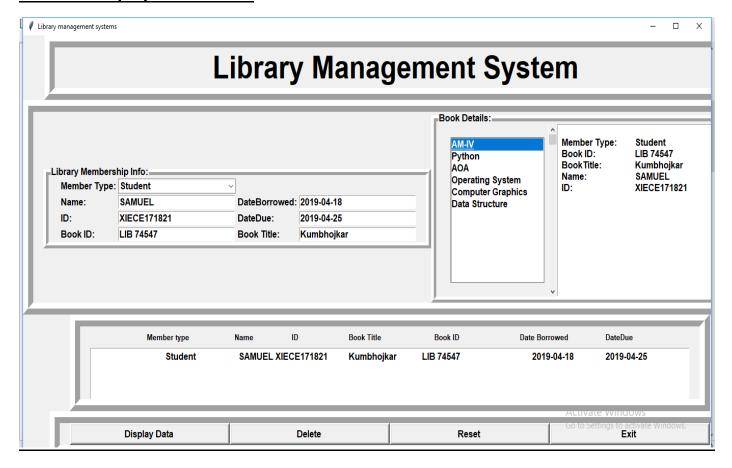
```python
        BookTitle.set("Core of python programming")

        iReceipt()

        import datetime

        d1=datetime.date.today()

        d2=datetime.timedelta(days=7)

        d3=(d1+d2)

        DateBorrowed.set(d1)

        DateDue.set(d3)

if(w=="Operating System"):

        BookID.set("LIB 74549")

        BookTitle.set("Stalling")

        iReceipt()

        import datetime

        d1=datetime.date.today()

        d2=datetime.timedelta(days=7)

        d3=(d1+d2)

        DateBorrowed.set(d1)

        DateDue.set(d3)

 if(w=="Data Structure"):

        BookID.set("LIB 74550")

        BookTitle.set("Ds structure using c")

        iReceipt()

        import datetime

        d1=datetime.date.today()

        d2=datetime.timedelta(days=7)

        d3=(d1+d2)

        DateBorrowed.set(d1)
```

```python
        DateDue.set(d3)
   if(w=="Computer Graphics"):
       BookID.set("LIB 74551")
       BookTitle.set("Techmax")
       iReceipt()
       import datetime
       d1=datetime.date.today()
       d2=datetime.timedelta(days=7)
       d3=(d1+d2)
       DateBorrowed.set(d1)
       DateDue.set(d3)
     if(w=="AOA"):
       BookID.set("LIB 74552")
       BookTitle.set("Coremann")
       iReceipt()
       import datetime
       d1=datetime.date.today()
       d2=datetime.timedelta(days=7)
       d3=(d1+d2)
       DateBorrowed.set(d1)
       DateDue.set(d3)


booklist=Listbox(DataFrameRIGHT,width=20,height=12,font=('arial',12,'bold'))

booklist.bind('<<ListboxSelect>>',SelectedBook)

booklist.grid(row=0,column=0,padx=8)

scrollbar.config(command=booklist.yview)
```
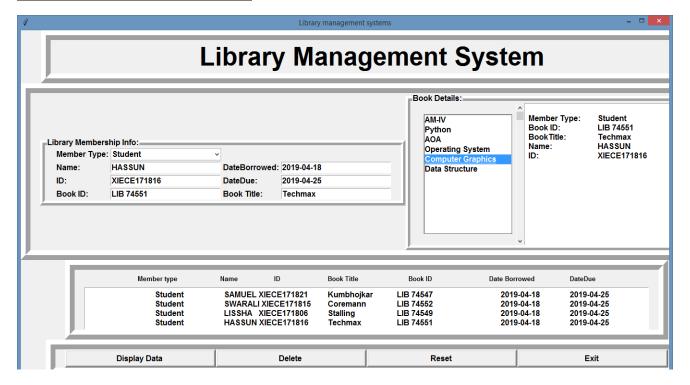
```python
        for items in ListOfBooks:

            booklist.insert(END,items)

        #====

        self.lblLabel=Label(FrameDetail,font=('arial',10,'bold'),pady=8,

        text="Member type\t\t Name \t\t ID \t\t Book Title \t\t Book ID \t\t Date Borrowed \t\t
DateDue",)

        self.lblLabel.grid(row=0,column=0)
        self.txtFrameDetail=Text(FrameDetail,font=('arial',12,'bold'),width=130,height=4,padx=2,p
        ady=4)

        self.txtFrameDetail.grid(row=1,column=0)




#Button

        self.btnDisplayData=Button(ButtonFrame,text='DisplayData
',font=('arial',12,'bold'),width=30,bd=4,command=iDisplayData)

        self.btnDisplayData.grid(row=0,column=0)

        self.btnDelete=Button(ButtonFrame,  text='Delete  ',font=('arial',12,'bold'),width=30,bd=4,
command = iDelete)

        self.btnDelete.grid(row=0,column=1)

        self.btnReset=Button(ButtonFrame,   text='Reset   ',font=('arial',12,'bold'),width=30,bd=4,
command=iReset)

        self.btnReset.grid(row=0,column=2)

        self.btnExit=Button(ButtonFrame,    text='Exit    ',font=('arial',12,'bold'),width=30,bd=4,
command=iExit)

        self.btnExit.grid(row=0,column=3)
if __name__=='__main__':

    root = Tk()

    application = Library(root)

    root.mainloop()
```
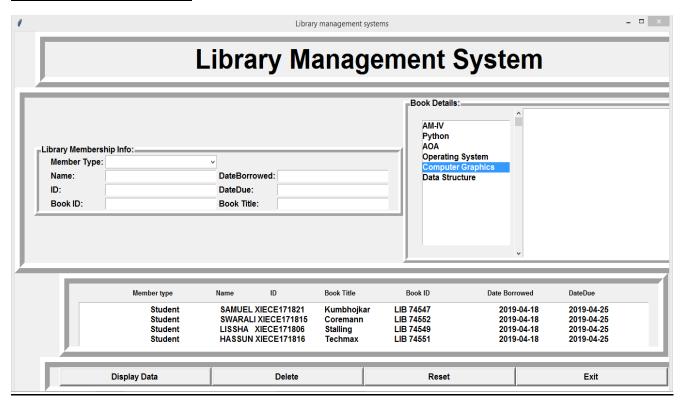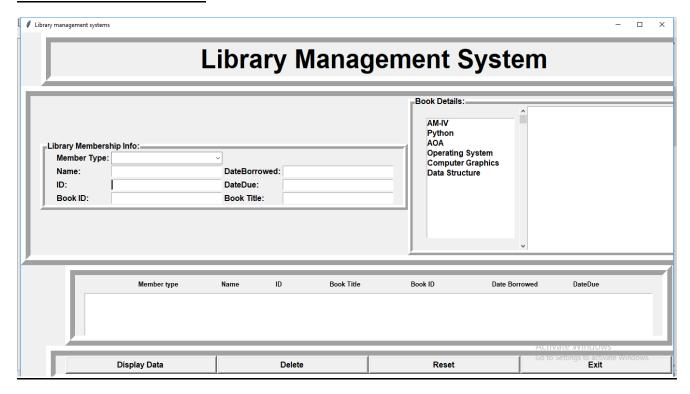
OUTPUT: -

Action of Display Data Button:

Action of Display Data Button:
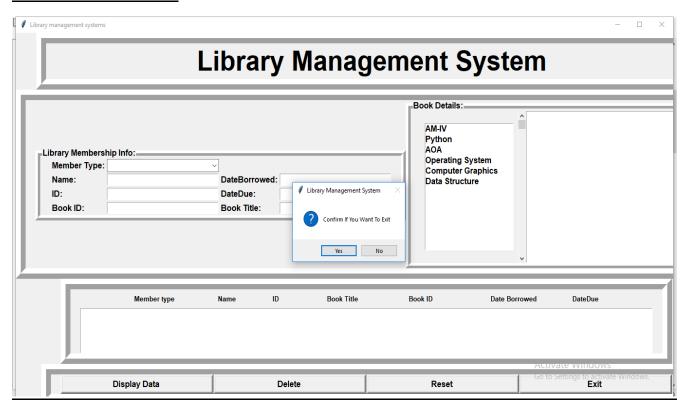

Action of Reset Button:

Action of Delete Button:

Action of Exit Button:

CONCLUSION:

The Library management System has been comuted successfully and was also tested successfully by taking "test cases". It is user friendly and has required option which can be utilized by the user to perform the desired operation. The library management system is for computerizing the working in a library. The software take care of all the requirement of an a library and is capable to provide easy and effective storage of information related to books &user.

Goals that are achieved by the project:

- Efficient management of records.
- Simplication of theoperation.
- Less processing time.
- User friendly.

REFERENCES:

1. https://geeksforgeeks.org/
2. YOUTUBE