FEU Institute of Technology

MACHINE PROBLEM

7

Transaction Processing System

<Justine Guillermo>
<W22>
<Date>

Create a Python application that will serve as a simple Kiosk on a a given Library system.

SOURCE CODE:

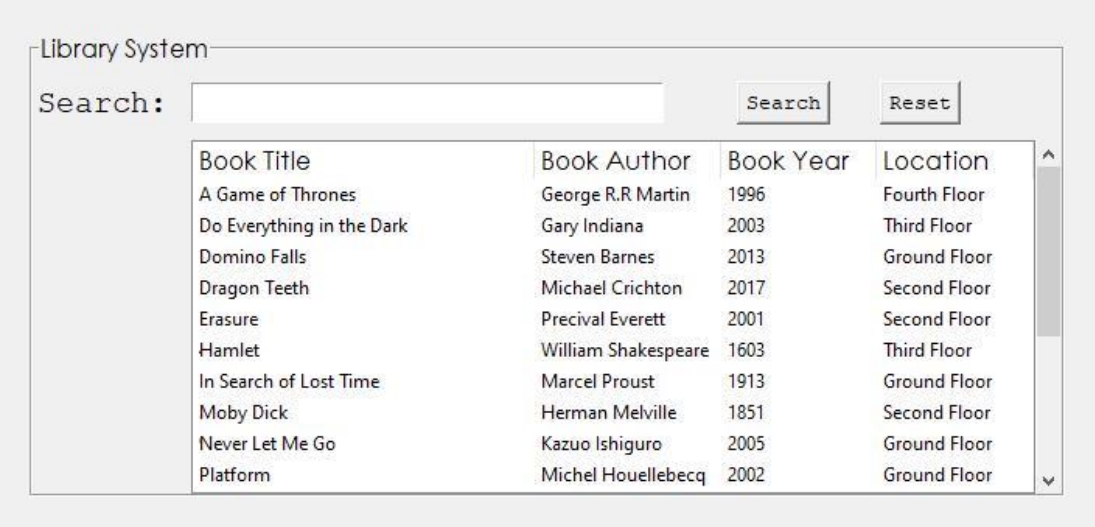
excel.py | machineproblem7.py

machineproblem7.py > Root > __init_

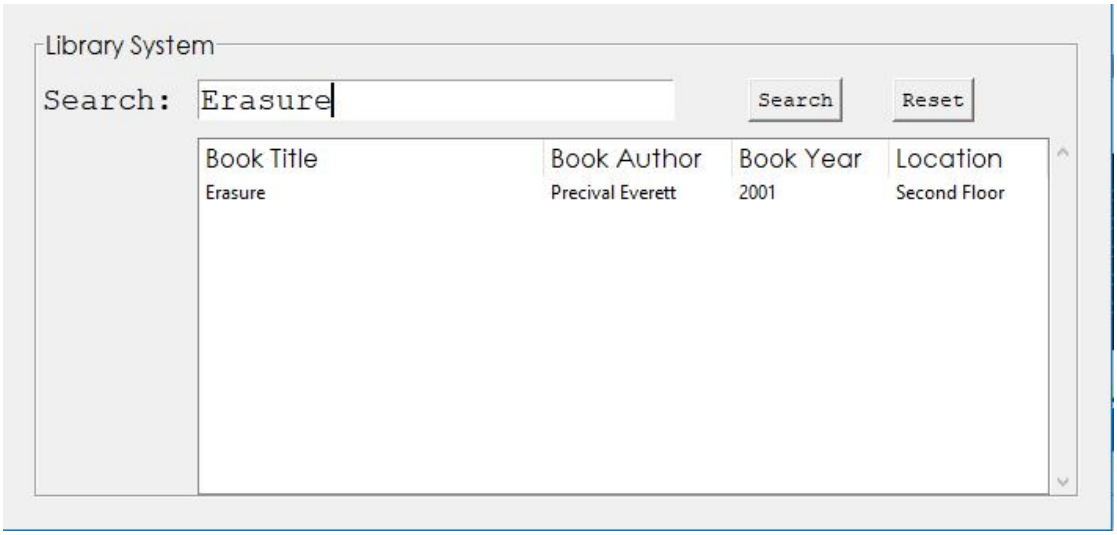
```
#MP7
import tkinter as tk
import tkinter.font as tkFont
from tkinter import *
from tkinter import ttk
import sqlite3

class Root(Tk):
    def __init__(self):
        super(Root, self).__init__()
        self.title("Library System Kiosk")
        self.fontFam = tkFont.Font(family = "Century Gothic", size = 12)
        self.minsize(600, 300)
        self.labelFrame=LabelFrame(self, text = "Library System")
        self.labelFrame.config(font = self.fontFam)
        self.labelFrame.grid(column = 0, row = 1, padx = 20, pady = 20)
        self.resizable(False, False)
        self.objects()

    def objects(self):
        def Database():
            conn = sqlite3.connect("listofbooks.db")
            cursor = conn.cursor()
            cursor.execute("CREATE TABLE IF NOT EXISTS `book` (mem_id INTEGER NOT NULL PRIMARY KEY  AUTOINCREMENT, booktitle TEXT, bookauthor TEXT, bookyear TEXT, location TEXT)")
            cursor.execute("SELECT * FROM `book`")
            if cursor.fetchone() is None:
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('In Search of Lost Time', 'Marcel Proust', '1913', 'Ground Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Ulysses', 'James Joyce', '1922', 'Second Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Moby Dick', 'Herman Melville', '1851', 'Second Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Hamlet', 'William Shakespeare', '1603', 'Third Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('The Brothers Karamazov', 'Fyodor Dostoyevsky', '1880', 'Fourth Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('The Divine Comedy', 'Dante Alighieri', '1472', 'Ground Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('The Adventures of Huckleberry Finn', 'Mark Twain', '1884', 'Second Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Pride and Prejudice', 'Jane Austen', '1813', 'Ground Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Never Let Me Go', 'Kazuo Ishiguro', '2005', 'Ground Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Erasure', 'Precival Everett', '2001', 'Second Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Platform', 'Michel Houellebecq', '2002', 'Ground Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Do Everything in the Dark', 'Gary Indiana', '2003', 'Third Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('A Game of Thrones', 'George R.R Martin', '1996', 'Fourth Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Vampire Academy', 'Richelle Mead', '2007', 'Fourth Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Sherlock Holmes and the Needles Eye', 'Len Bailey', '2013', 'Second Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Dragon Teeth', 'Michael Crichton', '2017', 'Second Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Unwholly', 'Neal Shusterman', '2012', 'Third Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('The Calling', 'Rachelle Dekker', '2016', 'Fourth Floor')")
                cursor.execute("INSERT INTO `book` (booktitle, bookauthor, bookyear, location) VALUES('Domino Falls', 'Steven Barnes', '2013', 'Ground Floor')")
                conn.commit()

            cursor.execute("SELECT * FROM `book` ORDER BY `booktitle` ASC")
            fetch = cursor.fetchall()
            for data in fetch:
                tree.insert('', 'end', values=(data))
            cursor.close()
            conn.close()

        def Search():
            if SEARCH.get() != "":
                tree.delete(*tree.get_children())
                conn = sqlite3.connect("listofbooks.db")
                cursor = conn.cursor()
                cursor.execute("SELECT * FROM book WHERE booktitle LIKE ? OR bookauthor LIKE ?", ('%'+str(SEARCH.get())+'%', '%'+str(SEARCH.get())+'%'))
                fetch = cursor.fetchall()
                for data in fetch:
                    tree.insert('', 'end', values=(data))
                cursor.close()
                conn.close()
        def Reset():
            conn = sqlite3.connect("listofbooks.db")
            cursor = conn.cursor()
            tree.delete(*tree.get_children())
            cursor.execute("SELECT * FROM `book` ORDER BY `booktitle` ASC")
            fetch = cursor.fetchall()
            for data in fetch:
                tree.insert('', 'end', values=(data))
            cursor.close()
            conn.close()

        SEARCH = StringVar()
        Label(self.labelFrame, text='Search: ', font=("Courier", 15)).grid(row=2, column=0, pady = 10, columnspan = 2)
        Entry(self.labelFrame, width=25, font=("Courier", 15), textvariable=SEARCH).grid(row=2,column=2, pady = 10, columnspan = 2)
        Button(self.labelFrame, text='Search', font=("Courier", 10), command = Search).grid(row=2,column=5, pady = 10)
        Button(self.labelFrame, text='Reset', font=("Courier", 10), command = Reset).grid(row=2,column=6, pady = 10)
        tree = ttk.Treeview(self.labelFrame, columns = header, show = "headings", selectmode="extended",)
        vsb = ttk.Scrollbar(self.labelFrame,orient = "vertical", command = tree.yview)
        tree.configure(yscrollcommand=vsb.set)
        style = ttk.Style()
        style.configure("Treeview.Heading",font = self.fontFam)
        tree.grid(column = 2, row = 3, sticky='nsew', columnspan = 6)
        vsb.grid(column=8, row=3, sticky='ns')
        tree.heading('Book Title', text="Book Title",anchor=W)
        tree.heading('Book Author', text="Book Author",anchor=W)
        tree.heading('Book Year', text="Book Year",anchor=W)
        tree.heading('Location', text="Location",anchor=W)
        tree.column('#1', minwidth=0, width=0)
        tree.column('#2', minwidth=0, width=220)
        tree.column('#3', minwidth=0, width=120)
        tree.column('#4', minwidth=0, width=100)
        tree.column('#5', minwidth=0, width=100)
        Database()
```

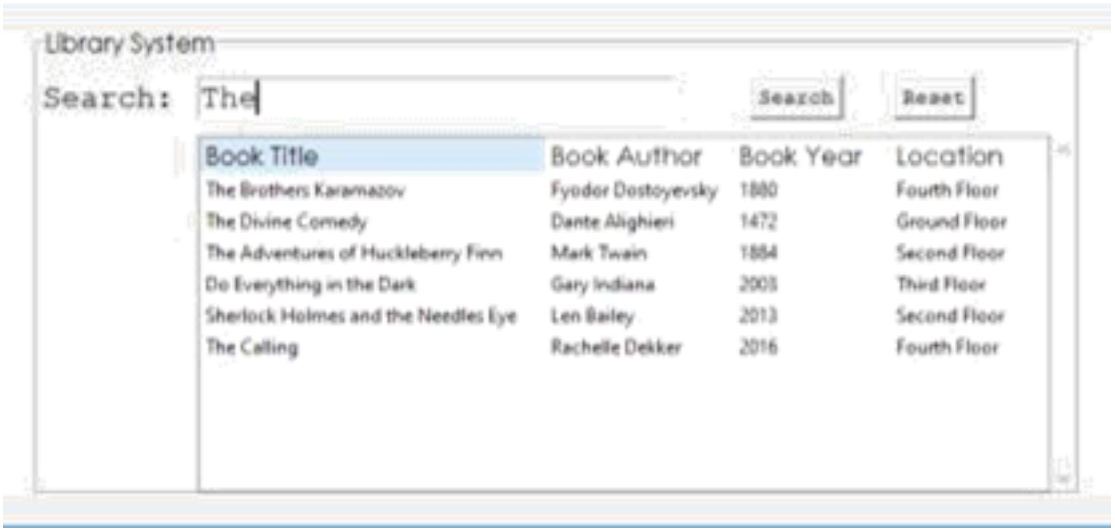
```
99
100  header = ['ID','Book Title', 'Book Author', 'Book Year', 'Location']
101
102  if __name__ == '__main__':
103      root = Root()
104      root.mainloop()
```

OUTPUT:

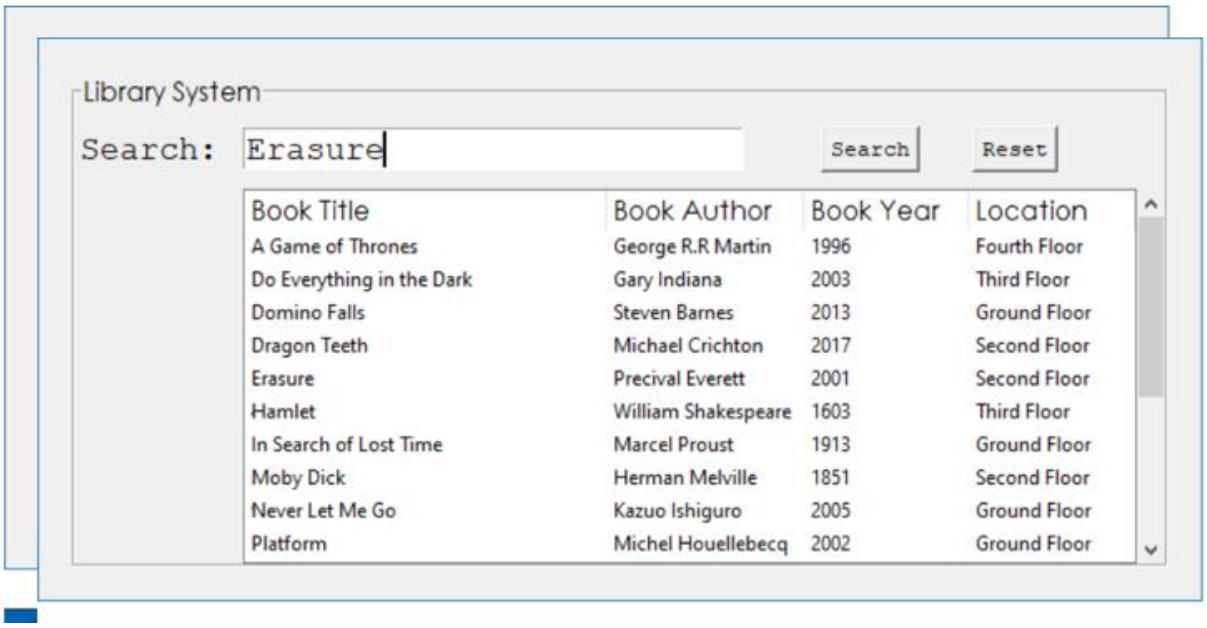


SEARCH BUTTON





RESET BUTTON




Library System
Search:
The
Search
Reset
Book Title
Book Author
Book Year
Location
A Game of Thrones
George R.R Martin
1996
Fourth Floor
Do Everything in the Dark
Gary Indiana
2003
Third Floor
Domino Falls
Steven Barnes
2013
Ground Floor
Dragon Teeth
Michael Crichton
2017
Second Floor
Erasure
Precival Everett
2001
Second Floor
Hamlet
William Shakespeare
1603
Third Floor
In Search of Lost Time
Marcel Proust
1913
Ground Floor
Moby Dick
Herman Melville
1851
Second Floor
Never Let Me Go
Kazuo Ishiguro
2005
Ground Floor
Platform
Michel Houellebecq
2002
Ground Floor