# Importing all necessary modules

import sqlite3

from tkinter import \*

import tkinter.ttk as ttk

import tkinter.messagebox as mb

import tkinter.simpledialog as sd

# Connecting to Database

connector = sqlite3.connect('library.db')

cursor = connector.cursor()

connector.execute(

'CREATE TABLE IF NOT EXISTS Library (BK\_NAME TEXT, BK\_ID TEXT PRIMARY KEY NOT NULL, AUTHOR\_NAME TEXT, BK\_STATUS TEXT, CARD\_ID TEXT)'

)

# Functions

def issuer\_card():

Cid = sd.askstring('Issuer Card ID', 'What is the Issuer\'s Card ID?\t\t\t')

if not Cid:

mb.showerror('Issuer ID cannot be zero!', 'Can\'t keep Issuer ID empty, it must have a value')

else:

return Cid

def display\_records():

global connector, cursor

global tree

tree.delete(\*tree.get\_children())

curr = connector.execute('SELECT \* FROM Library')

data = curr.fetchall()

for records in data:

tree.insert('', END, values=records)

def clear\_fields():

global bk\_status, bk\_id, bk\_name, author\_name, card\_id

bk\_status.set('Available')

n for i in ['bk\_id', 'bk\_name', 'author\_name', 'card\_id']:

exec(f"{i}.set('')")

bk\_id\_entry.config(state='normal')

try:

tree.selection\_remove(tree.selection()[0])

except:

pass

def clear\_and\_display():

clear\_fields()

display\_records()

def add\_record():

global connector

global bk\_name, bk\_id, author\_name, bk\_status

if bk\_status.get() == 'Issued':

card\_id.set(issuer\_card())

else:

card\_id.set('N/A')

surety = mb.askyesno('Are you sure?',

'Are you sure this is the data you want to enter?\nPlease note that Book ID cannot be changed in the future')

if surety:

try:

connector.execute(

'INSERT INTO Library (BK\_NAME, BK\_ID, AUTHOR\_NAME, BK\_STATUS, CARD\_ID) VALUES (?, ?, ?, ?, ?)',

(bk\_name.get(), bk\_id.get(), author\_name.get(), bk\_status.get(), card\_id.get()))

connector.commit()

clear\_and\_display()

mb.showinfo('Record added', 'The new record was successfully added to your database')

except sqlite3.IntegrityError:

mb.showerror('Book ID already in use!',

'The Book ID you are trying to enter is already in the database, please alter that book\'s record or check any discrepancies on your side')

def view\_record():

global bk\_name, bk\_id, bk\_status, author\_name, card\_id

global tree

if not tree.focus():

mb.showerror('Select a row!', 'To view a record, you must select it in the table. Please do so before continuing.')

return

current\_item\_selected = tree.focus()

values\_in\_selected\_item = tree.item(current\_item\_selected)

selection = values\_in\_selected\_item['values']

bk\_name.set(selection[0]) ; bk\_id.set(selection[1]) ; bk\_status.set(selection[3])

author\_name.set(selection[2])

try:

card\_id.set(selection[4])

except:

card\_id.set('')

def update\_record():

def update():

global bk\_status, bk\_name, bk\_id, author\_name, card\_id

global connector, tree

if bk\_status.get() == 'Issued':

card\_id.set(issuer\_card())

else:

card\_id.set('N/A')

cursor.execute('UPDATE Library SET BK\_NAME=?, BK\_STATUS=?, AUTHOR\_NAME=?, CARD\_ID=? WHERE BK\_ID=?',

(bk\_name.get(), bk\_status.get(), author\_name.get(), card\_id.get(), bk\_id.get()))

connector.commit()

clear\_and\_display()

edit.destroy()

bk\_id\_entry.config(state='normal')

clear.config(state='normal')

view\_record()

bk\_id\_entry.config(state='disable')

clear.config(state='disable')

edit = Button(left\_frame, text='Update Record', font=btn\_font, bg=btn\_hlb\_bg, width=20, command=update)

edit.place(x=50, y=375)

def remove\_record():

if not tree.selection():

mb.showerror('Error!', 'Please select an item from the database')

return

current\_item = tree.focus()

values = tree.item(current\_item)

selection = values["values"]

cursor.execute('DELETE FROM Library WHERE BK\_ID=?', (selection[1], ))

connector.commit()

tree.delete(current\_item)

mb.showinfo('Done', 'The record you wanted deleted was successfully deleted.')

clear\_and\_display()

def delete\_inventory():

if mb.askyesno('Are you sure?', 'Are you sure you want to delete the entire inventory?\n\nThis command cannot be reversed'):

tree.delete(\*tree.get\_children())

cursor.execute('DELETE FROM Library')

connector.commit()

else:

return

def change\_availability():

global card\_id, tree, connector

if not tree.selection():

mb.showerror('Error!', 'Please select a book from the database')

return

current\_item = tree.focus()

values = tree.item(current\_item)

BK\_id = values['values'][1]

BK\_status = values["values"][3]

if BK\_status == 'Issued':

surety = mb.askyesno('Is return confirmed?', 'Has the book been returned to you?')

if surety:

cursor.execute('UPDATE Library SET bk\_status=?, card\_id=? WHERE bk\_id=?', ('Available', 'N/A', BK\_id))

connector.commit()

else: mb.showinfo(

'Cannot be returned', 'The book status cannot be set to Available unless it has been returned')

else:

cursor.execute('UPDATE Library SET bk\_status=?, card\_id=? where bk\_id=?', ('Issued', issuer\_card(), BK\_id))

connector.commit()

clear\_and\_display()

# Variables

lf\_bg = 'LightSkyBlue' # Left Frame Background Color

rtf\_bg = 'DeepSkyBlue' # Right Top Frame Background Color

rbf\_bg = 'DodgerBlue' # Right Bottom Frame Background Color

btn\_hlb\_bg = 'SteelBlue' # Background color for Head Labels and Buttons

lbl\_font = ('Georgia', 13) # Font for all labels

entry\_font = ('Times New Roman', 12) # Font for all Entry widgets

btn\_font = ('Gill Sans MT', 13)

# Initializing the main GUI window

root = Tk()

root.title('PythonGeeks Library Management System')

root.geometry('1010x530')

root.resizable(0, 0)

Label(root, text='LIBRARY MANAGEMENT SYSTEM', font=("Noto Sans CJK TC", 15, 'bold'), bg=btn\_hlb\_bg, fg='White').pack(side=TOP, fill=X)

# StringVars

bk\_status = StringVar()

bk\_name = StringVar()

bk\_id = StringVar()

author\_name = StringVar()

card\_id = StringVar()

# Frames

left\_frame = Frame(root, bg=lf\_bg)

left\_frame.place(x=0, y=30, relwidth=0.3, relheight=0.96)

RT\_frame = Frame(root, bg=rtf\_bg)

RT\_frame.place(relx=0.3, y=30, relheight=0.2, relwidth=0.7)

RB\_frame = Frame(root)

RB\_frame.place(relx=0.3, rely=0.24, relheight=0.785, relwidth=0.7)

# Left Frame

Label(left\_frame, text='Book Name', bg=lf\_bg, font=lbl\_font).place(x=98, y=25)

Entry(left\_frame, width=25, font=entry\_font, text=bk\_name).place(x=45, y=55)

Label(left\_frame, text='Book ID', bg=lf\_bg, font=lbl\_font).place(x=110, y=105)

bk\_id\_entry = Entry(left\_frame, width=25, font=entry\_font, text=bk\_id)

bk\_id\_entry.place(x=45, y=135)

Label(left\_frame, text='Author Name', bg=lf\_bg, font=lbl\_font).place(x=90, y=185)

Entry(left\_frame, width=25, font=entry\_font, text=author\_name).place(x=45, y=215)

Label(left\_frame, text='Status of the Book', bg=lf\_bg, font=lbl\_font).place(x=75, y=265)

dd = OptionMenu(left\_frame, bk\_status, \*['Available', 'Issued'])

dd.configure(font=entry\_font, width=12)

dd.place(x=75, y=300)

submit = Button(left\_frame, text='Add new record', font=btn\_font, bg=btn\_hlb\_bg, width=20, command=add\_record)

submit.place(x=50, y=375)

clear = Button(left\_frame, text='Clear fields', font=btn\_font, bg=btn\_hlb\_bg, width=20, command=clear\_fields)

clear.place(x=50, y=435)

# Right Top Frame

Button(RT\_frame, text='Delete book record', font=btn\_font, bg=btn\_hlb\_bg, width=17, command=remove\_record).place(x=8, y=30)

Button(RT\_frame, text='Delete full inventory', font=btn\_font, bg=btn\_hlb\_bg, width=17, command=delete\_inventory).place(x=178, y=30)

Button(RT\_frame, text='Update book details', font=btn\_font, bg=btn\_hlb\_bg, width=17,

command=update\_record).place(x=348, y=30)

Button(RT\_frame, text='Change Book Availability', font=btn\_font, bg=btn\_hlb\_bg, width=19,

command=change\_availability).place(x=518, y=30)

# Right Bottom Frame

Label(RB\_frame, text='BOOK INVENTORY', bg=rbf\_bg, font=("Noto Sans CJK TC", 15, 'bold')).pack(side=TOP, fill=X)

tree = ttk.Treeview(RB\_frame, selectmode=BROWSE, columns=('Book Name', 'Book ID', 'Author', 'Status', 'Issuer Card ID'))

XScrollbar = Scrollbar(tree, orient=HORIZONTAL, command=tree.xview)

YScrollbar = Scrollbar(tree, orient=VERTICAL, command=tree.yview)

XScrollbar.pack(side=BOTTOM, fill=X)

YScrollbar.pack(side=RIGHT, fill=Y)

tree.config(xscrollcommand=XScrollbar.set, yscrollcommand=YScrollbar.set)

tree.heading('Book Name', text='Book Name', anchor=CENTER)

tree.heading('Book ID', text='Book ID', anchor=CENTER)

tree.heading('Author', text='Author', anchor=CENTER)

tree.heading('Status', text='Status of the Book', anchor=CENTER)

tree.heading('Issuer Card ID', text='Card ID of the Issuer', anchor=CENTER)

tree.column('#0', width=0, stretch=NO)

tree.column('#1', width=225, stretch=NO)

tree.column('#2', width=70, stretch=NO)

tree.column('#3', width=150, stretch=NO)

tree.column('#4', width=105, stretch=NO)

tree.column('#5', width=132, stretch=NO)

tree.place(y=30, x=0, relheight=0.9, relwidth=1)

clear\_and\_display()

# Finalizing the window

root.update()

root.mainloop()