SOURCE CODE FOR THE PROJECT

LIBRARY MANAGEMENT USING PYTHON

import tkinter as tk

from tkinter import messagebox

class LibraryManagement:

def \_\_init\_\_(self, master):

self.master = master

self.master.title("Library Management System")

self.master.geometry("400x400")

self.master.config(bg='#708090')

self.books = []

self.lend\_list = []

# Labels

self.login\_label = tk.Label(self.master, text="Library Management System", font=("Helvetica", 16), bg='#708090', fg='white')

self.login\_label.pack()

self.username\_label = tk.Label(self.master, text="Username", font=("Helvetica", 12), bg='#708090', fg='white')

self.username\_label.pack()

self.username\_entry = tk.Entry(self.master, font=("Helvetica", 12))

self.username\_entry.pack()

self.password\_label = tk.Label(self.master, text="Password", font=("Helvetica", 12), bg='#708090', fg='white')

self.password\_label.pack()

self.password\_entry = tk.Entry(self.master, font=("Helvetica", 12), show="\*")

self.password\_entry.pack()

# Login

self.login\_button = tk.Button(self.master, text="Login", command=self.login, font=("Helvetica", 12))

self.login\_button.pack()

# Register

self.register\_button = tk.Button(self.master, text="Register", command=self.register, font=("Helvetica", 12))

self.register\_button.pack()

self.username = ""

self.password = ""

self.librarians = []

def login(self):

self.username = self.username\_entry.get()

self.password = self.password\_entry.get()

for librarian in self.librarians:

if self.username == librarian[0] and self.password == librarian[1]:

self.username\_entry.delete(0, tk.END)

self.password\_entry.delete(0, tk.END)

self.login\_label.destroy()

self.username\_label.destroy()

self.username\_entry.destroy()

self.password\_label.destroy()

self.password\_entry.destroy()

self.login\_button.destroy()

self.register\_button.destroy()

self.library\_management\_screen()

return

messagebox.showerror("Error", "Invalid username or password")

def register(self):

self.username = self.username\_entry.get()

self.password = self.password\_entry.get()

self.librarians.append([self.username, self.password])

self.username\_entry.delete(0, tk.END)

self.password\_entry.delete(0, tk.END)

def library\_management\_screen(self):

self.add\_book\_label = tk.Label(self.master, text="Add Book", font=("Helvetica", 16), bg='#708090', fg='white')

self.add\_book\_label.pack()

self.add\_book\_entry = tk.Entry(self.master, font=("Helvetica", 12))

self.add\_book\_entry.pack()

self.add\_book\_button = tk.Button(self.master, text="Add Book", command=self.add\_book, font=("Helvetica", 12))

self.add\_book\_button.pack()

self.remove\_book\_label = tk.Label(self.master, text="Remove Book", font=("Helvetica", 16), bg='#708090', fg='white')

self.remove\_book\_label.pack()

self.remove\_book\_entry = tk.Entry(self.master, font=("Helvetica", 12))

self.remove\_book\_entry.pack()

self.remove\_book\_button = tk.Button(self.master, text="Remove Book", command=self.remove\_book, font=("Helvetica", 12))

self.remove\_book\_button.pack()

self.issue\_book\_label = tk.Label(self.master, text="Issue Book", font=("Helvetica", 16), bg='#708090', fg='white')

self.issue\_book\_label.pack()

self.issue\_book\_entry = tk.Entry(self.master, font=("Helvetica", 12))

self.issue\_book\_entry.pack()

self.issue\_book\_button = tk.Button(self.master, text="Issue Book", command=self.issue\_book, font=("Helvetica", 12))

self.issue\_book\_button.pack()

self.view\_books\_button = tk.Button(self.master, text="View Books", command=self.view\_books, font=("Helvetica", 12))

self.view\_books\_button.pack()

def add\_book(self):

book = self.add\_book\_entry.get()

self.books.append(book)

messagebox.showinfo("Success", "Book added successfully")

self.add\_book\_entry.delete(0, tk.END)

def remove\_book(self):

book = self.remove\_book\_entry.get()

if book in self.books:

self.books.remove(book)

messagebox.showinfo("Success", "Book removed successfully")

else:

messagebox.showerror("Error", "Book not found")

self.remove\_book\_entry.delete(0, tk.END)

def issue\_book(self):

book = self.issue\_book\_entry.get()

if book in self.books:

self.lend\_list.append(book)

self.books.remove(book)

messagebox.showinfo("Success", "Book issued successfully")

else:

messagebox.showerror("Error", "Book not found")

self.issue\_book\_entry.delete(0, tk.END)

def view\_books(self):

message = "\n".join(self.books)

messagebox.showinfo("Books", message)

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

app = LibraryManagement(root)

root.mainloop()