**APARESH MUHURI**

**For Admin Login :** Id- admin, Password- admin123

**For User Login:** Id- user, Password- user123

Code-

import tkinter as tk

from tkinter import messagebox, ttk

from datetime import datetime, timedelta

users = {

"admin": {"password": "admin123", "role": "admin"},

"user": {"password": "user123", "role": "user"}

}

books = {

"The Great Gatsby": {"author": "F. Scott Fitzgerald", "available": True},

"1984": {"author": "George Orwell", "available": True}

}

class LibraryManagementSystem(tk.Tk):

def \_init\_(self):

super().\_init\_()

self.title("Library Management System")

self.geometry("400x300")

self.current\_user = None

self.create\_login\_page()

def create\_login\_page(self):

self.clear\_frame()

tk.Label(self, text="Username").pack()

self.username\_entry = tk.Entry(self)

self.username\_entry.pack()

tk.Label(self, text="Password").pack()

self.password\_entry = tk.Entry(self, show="\*")

self.password\_entry.pack()

self.role\_var = tk.StringVar(value="user")

tk.Radiobutton(self, text="Admin", variable=self.role\_var, value="admin").pack()

tk.Radiobutton(self, text="User", variable=self.role\_var, value="user").pack()

tk.Button(self, text="Login", command=self.login).pack()

def login(self):

username = self.username\_entry.get()

password = self.password\_entry.get()

role = self.role\_var.get()

if username in users and users[username]['password'] == password and users[username]['role'] == role:

self.current\_user = role

self.create\_dashboard()

else:

messagebox.showerror("Login Failed", "Invalid username or password!")

def create\_dashboard(self):

self.clear\_frame()

tk.Label(self, text=f"Welcome, {self.current\_user.capitalize()}!").pack()

if self.current\_user == "admin":

tk.Button(self, text="Maintenance", command=self.create\_maintenance\_page).pack()

tk.Button(self, text="Reports", command=self.create\_reports\_page).pack()

tk.Button(self, text="Transactions", command=self.create\_transactions\_page).pack()

tk.Button(self, text="Logout", command=self.create\_login\_page).pack()

def create\_maintenance\_page(self):

self.clear\_frame()

tk.Label(self, text="Maintenance Module (Admin Only)").pack()

tk.Button(self, text="Add Book", command=self.add\_book).pack()

tk.Button(self, text="Update Book", command=self.update\_book).pack()

tk.Button(self, text="Back", command=self.create\_dashboard).pack()

def add\_book(self):

self.clear\_frame()

tk.Label(self, text="Add New Book").pack()

self.book\_name\_var = tk.StringVar()

tk.Label(self, text="Book Name").pack()

tk.Entry(self, textvariable=self.book\_name\_var).pack()

self.author\_name\_var = tk.StringVar()

tk.Label(self, text="Author Name").pack()

tk.Entry(self, textvariable=self.author\_name\_var).pack()

tk.Button(self, text="Submit", command=self.save\_new\_book).pack()

tk.Button(self, text="Back", command=self.create\_maintenance\_page).pack()

def save\_new\_book(self):

book\_name = self.book\_name\_var.get()

author\_name = self.author\_name\_var.get()

if not book\_name or not author\_name:

messagebox.showerror("Error", "All fields are mandatory!")

else:

books[book\_name] = {"author": author\_name, "available": True}

messagebox.showinfo("Success", f"Book '{book\_name}' added successfully!")

self.create\_maintenance\_page()

def update\_book(self):

self.clear\_frame()

tk.Label(self, text="Update Book Details").pack()

self.book\_name\_var = tk.StringVar()

tk.Label(self, text="Select Book").pack()

book\_list = ttk.Combobox(self, values=list(books.keys()), textvariable=self.book\_name\_var)

book\_list.pack()

self.author\_name\_var = tk.StringVar()

tk.Label(self, text="Author Name (Non-editable)").pack()

tk.Entry(self, textvariable=self.author\_name\_var, state='disabled').pack()

tk.Button(self, text="Update", command=self.update\_book\_details).pack()

tk.Button(self, text="Back", command=self.create\_maintenance\_page).pack()

def update\_book\_details(self):

book\_name = self.book\_name\_var.get()

if book\_name:

self.author\_name\_var.set(books[book\_name]['author'])

else:

messagebox.showerror("Error", "Please select a book!")

def create\_transactions\_page(self):

self.clear\_frame()

tk.Label(self, text="Transactions Module").pack()

tk.Button(self, text="Issue Book", command=self.issue\_book).pack()

tk.Button(self, text="Return Book", command=self.return\_book).pack()

tk.Button(self, text="Back", command=self.create\_dashboard).pack()

def issue\_book(self):

self.clear\_frame()

tk.Label(self, text="Issue Book").pack()

self.issue\_book\_name\_var = tk.StringVar()

tk.Label(self, text="Book Name").pack()

book\_list = ttk.Combobox(self, values=list(books.keys()), textvariable=self.issue\_book\_name\_var)

book\_list.pack()

self.issue\_author\_name\_var = tk.StringVar()

tk.Label(self, text="Author Name (Non-editable)").pack()

tk.Entry(self, textvariable=self.issue\_author\_name\_var, state='disabled').pack()

tk.Button(self, text="Issue", command=self.process\_issue\_book).pack()

tk.Button(self, text="Back", command=self.create\_transactions\_page).pack()

def process\_issue\_book(self):

book\_name = self.issue\_book\_name\_var.get()

if not book\_name:

messagebox.showerror("Error", "Please select a book!")

else:

self.issue\_author\_name\_var.set(books[book\_name]['author'])

def return\_book(self):

self.clear\_frame()

tk.Label(self, text="Return Book").pack()

tk.Button(self, text="Back", command=self.create\_transactions\_page).pack()

def create\_reports\_page(self):

self.clear\_frame()

tk.Label(self, text="Reports Module").pack()

tk.Button(self, text="Back", command=self.create\_dashboard).pack()

def clear\_frame(self):

for widget in self.winfo\_children():

widget.destroy()

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

app = LibraryManagementSystem()

app.mainloop()