# main.py

import tkinter as tk

from tkinter import messagebox, ttk

import db\_handler

import datetime

# App settings

APP\_BG = "#121212"

FG\_COLOR = "#FFFFFF"

BTN\_BG = "#1F1F1F"

BTN\_FG = "#FFFFFF"

ENTRY\_BG = "#1E1E1E"

ENTRY\_FG = "#FFFFFF"

# Initialize DB

db\_handler.create\_tables()

# Main Application Class

class HostelApp(tk.Tk):

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

super().\_\_init\_\_()

self.title("Hostel Management System")

self.geometry("800x600")

self.configure(bg=APP\_BG)

self.frames = {}

self.active\_frame = None

# Start with Login Page

self.show\_frame(LoginPage)

def show\_frame(self, page\_class):

# Destroy current frame

if self.active\_frame:

self.active\_frame.destroy()

frame = page\_class(parent=self, controller=self)

self.active\_frame = frame

frame.pack(fill="both", expand=True)

# Login Page

class LoginPage(tk.Frame):

def \_\_init\_\_(self, parent, controller):

super().\_\_init\_\_(parent, bg=APP\_BG)

self.controller = controller

tk.Label(self, text="Hostel Login", font=("Arial", 20), bg=APP\_BG, fg=FG\_COLOR).pack(pady=20)

tk.Label(self, text="Username:", bg=APP\_BG, fg=FG\_COLOR).pack(pady=5)

self.username\_entry = tk.Entry(self, bg=ENTRY\_BG, fg=ENTRY\_FG)

self.username\_entry.pack()

tk.Label(self, text="Password:", bg=APP\_BG, fg=FG\_COLOR).pack(pady=5)

self.password\_entry = tk.Entry(self, show="\*", bg=ENTRY\_BG, fg=ENTRY\_FG)

self.password\_entry.pack()

tk.Button(self, text="Login", command=self.login, bg=BTN\_BG, fg=BTN\_FG).pack(pady=20)

def login(self):

username = self.username\_entry.get()

password = self.password\_entry.get()

if username == "admin" and password == "admin123":

self.controller.show\_frame(Dashboard)

else:

messagebox.showerror("Login Failed", "Incorrect Username or Password")

# Dashboard

class Dashboard(tk.Frame):

def \_\_init\_\_(self, parent, controller):

super().\_\_init\_\_(parent, bg=APP\_BG)

self.controller = controller

tk.Label(self, text="Dashboard", font=("Arial", 24), bg=APP\_BG, fg=FG\_COLOR).pack(pady=20)

tk.Button(self, text="Manage Students", command=lambda: controller.show\_frame(ManageStudents),

bg=BTN\_BG, fg=BTN\_FG, width=20).pack(pady=10)

tk.Button(self, text="Manage Rooms", command=lambda: controller.show\_frame(ManageRooms),

bg=BTN\_BG, fg=BTN\_FG, width=20).pack(pady=10)

tk.Button(self, text="Manage Payments", command=lambda: controller.show\_frame(ManagePayments),

bg=BTN\_BG, fg=BTN\_FG, width=20).pack(pady=10)

# Manage Students Page

class ManageStudents(tk.Frame):

def \_\_init\_\_(self, parent, controller):

super().\_\_init\_\_(parent, bg=APP\_BG)

self.controller = controller

tk.Label(self, text="Manage Students", font=("Arial", 20), bg=APP\_BG, fg=FG\_COLOR).pack(pady=10)

# Entry form

form\_frame = tk.Frame(self, bg=APP\_BG)

form\_frame.pack(pady=10)

tk.Label(form\_frame, text="Name:", bg=APP\_BG, fg=FG\_COLOR).grid(row=0, column=0)

self.name\_entry = tk.Entry(form\_frame, bg=ENTRY\_BG, fg=ENTRY\_FG)

self.name\_entry.grid(row=0, column=1)

tk.Label(form\_frame, text="Age:", bg=APP\_BG, fg=FG\_COLOR).grid(row=1, column=0)

self.age\_entry = tk.Entry(form\_frame, bg=ENTRY\_BG, fg=ENTRY\_FG)

self.age\_entry.grid(row=1, column=1)

tk.Label(form\_frame, text="Room No:", bg=APP\_BG, fg=FG\_COLOR).grid(row=2, column=0)

self.room\_entry = tk.Entry(form\_frame, bg=ENTRY\_BG, fg=ENTRY\_FG)

self.room\_entry.grid(row=2, column=1)

tk.Button(self, text="Add Student", command=self.add\_student, bg=BTN\_BG, fg=BTN\_FG).pack(pady=5)

# Table

table\_frame = tk.Frame(self, bg=APP\_BG)

table\_frame.pack(pady=10, fill="both", expand=True)

self.student\_table = ttk.Treeview(table\_frame, columns=("ID", "Name", "Age", "Room"))

self.student\_table.heading("ID", text="ID")

self.student\_table.heading("Name", text="Name")

self.student\_table.heading("Age", text="Age")

self.student\_table.heading("Room", text="Room No")

self.student\_table["show"] = "headings"

vsb = ttk.Scrollbar(table\_frame, orient="vertical", command=self.student\_table.yview)

self.student\_table.configure(yscrollcommand=vsb.set)

vsb.pack(side="right", fill="y")

self.student\_table.pack(fill="both", expand=True)

tk.Button(self, text="Delete Selected Student", command=self.delete\_student, bg=BTN\_BG, fg=BTN\_FG).pack(pady=5)

tk.Button(self, text="Back", command=lambda: controller.show\_frame(Dashboard), bg=BTN\_BG, fg=BTN\_FG).pack(pady=10)

self.refresh\_table()

def add\_student(self):

name = self.name\_entry.get()

age = self.age\_entry.get()

room = self.room\_entry.get()

if name and age and room:

db\_handler.add\_student(name, age, room)

self.refresh\_table()

else:

messagebox.showerror("Error", "Fill all fields.")

def delete\_student(self):

selected = self.student\_table.selection()

if selected:

student\_id = self.student\_table.item(selected[0])['values'][0]

db\_handler.delete\_student(student\_id)

self.refresh\_table()

def refresh\_table(self):

for i in self.student\_table.get\_children():

self.student\_table.delete(i)

for row in db\_handler.view\_students():

self.student\_table.insert("", "end", values=row)

# Manage Rooms Page

class ManageRooms(tk.Frame):

def \_\_init\_\_(self, parent, controller):

super().\_\_init\_\_(parent, bg=APP\_BG)

self.controller = controller

tk.Label(self, text="Manage Rooms", font=("Arial", 20), bg=APP\_BG, fg=FG\_COLOR).pack(pady=10)

# Form

form\_frame = tk.Frame(self, bg=APP\_BG)

form\_frame.pack(pady=10)

tk.Label(form\_frame, text="Room No:", bg=APP\_BG, fg=FG\_COLOR).grid(row=0, column=0)

self.room\_no\_entry = tk.Entry(form\_frame, bg=ENTRY\_BG, fg=ENTRY\_FG)

self.room\_no\_entry.grid(row=0, column=1)

tk.Label(form\_frame, text="Capacity:", bg=APP\_BG, fg=FG\_COLOR).grid(row=1, column=0)

self.capacity\_entry = tk.Entry(form\_frame, bg=ENTRY\_BG, fg=ENTRY\_FG)

self.capacity\_entry.grid(row=1, column=1)

tk.Button(self, text="Add Room", command=self.add\_room, bg=BTN\_BG, fg=BTN\_FG).pack(pady=5)

# Table

table\_frame = tk.Frame(self, bg=APP\_BG)

table\_frame.pack(pady=10, fill="both", expand=True)

self.room\_table = ttk.Treeview(table\_frame, columns=("ID", "Room", "Capacity"))

self.room\_table.heading("ID", text="ID")

self.room\_table.heading("Room", text="Room No")

self.room\_table.heading("Capacity", text="Capacity")

self.room\_table["show"] = "headings"

vsb = ttk.Scrollbar(table\_frame, orient="vertical", command=self.room\_table.yview)

self.room\_table.configure(yscrollcommand=vsb.set)

vsb.pack(side="right", fill="y")

self.room\_table.pack(fill="both", expand=True)

tk.Button(self, text="Delete Selected Room", command=self.delete\_room, bg=BTN\_BG, fg=BTN\_FG).pack(pady=5)

tk.Button(self, text="Back", command=lambda: controller.show\_frame(Dashboard), bg=BTN\_BG, fg=BTN\_FG).pack(pady=10)

self.refresh\_table()

def add\_room(self):

room\_no = self.room\_no\_entry.get()

capacity = self.capacity\_entry.get()

if room\_no and capacity:

db\_handler.add\_room(room\_no, capacity)

self.refresh\_table()

else:

messagebox.showerror("Error", "Fill all fields.")

def delete\_room(self):

selected = self.room\_table.selection()

if selected:

room\_id = self.room\_table.item(selected[0])['values'][0]

db\_handler.delete\_room(room\_id)

self.refresh\_table()

def refresh\_table(self):

for i in self.room\_table.get\_children():

self.room\_table.delete(i)

for row in db\_handler.view\_rooms():

self.room\_table.insert("", "end", values=row)

# Manage Payments Page

class ManagePayments(tk.Frame):

def \_\_init\_\_(self, parent, controller):

super().\_\_init\_\_(parent, bg=APP\_BG)

self.controller = controller

tk.Label(self, text="Manage Payments", font=("Arial", 20), bg=APP\_BG, fg=FG\_COLOR).pack(pady=10)

# Form

form\_frame = tk.Frame(self, bg=APP\_BG)

form\_frame.pack(pady=10)

tk.Label(form\_frame, text="Student ID:", bg=APP\_BG, fg=FG\_COLOR).grid(row=0, column=0)

self.student\_id\_entry = tk.Entry(form\_frame, bg=ENTRY\_BG, fg=ENTRY\_FG)

self.student\_id\_entry.grid(row=0, column=1)

tk.Label(form\_frame, text="Amount:", bg=APP\_BG, fg=FG\_COLOR).grid(row=1, column=0)

self.amount\_entry = tk.Entry(form\_frame, bg=ENTRY\_BG, fg=ENTRY\_FG)

self.amount\_entry.grid(row=1, column=1)

tk.Button(self, text="Add Payment", command=self.add\_payment, bg=BTN\_BG, fg=BTN\_FG).pack(pady=5)

# Table

table\_frame = tk.Frame(self, bg=APP\_BG)

table\_frame.pack(pady=10, fill="both", expand=True)

self.payment\_table = ttk.Treeview(table\_frame, columns=("ID", "Student Name", "Amount", "Date"))

self.payment\_table.heading("ID", text="ID")

self.payment\_table.heading("Student Name", text="Student Name")

self.payment\_table.heading("Amount", text="Amount")

self.payment\_table.heading("Date", text="Date")

self.payment\_table["show"] = "headings"

vsb = ttk.Scrollbar(table\_frame, orient="vertical", command=self.payment\_table.yview)

self.payment\_table.configure(yscrollcommand=vsb.set)

vsb.pack(side="right", fill="y")

self.payment\_table.pack(fill="both", expand=True)

tk.Button(self, text="Back", command=lambda: controller.show\_frame(Dashboard), bg=BTN\_BG, fg=BTN\_FG).pack(pady=10)

self.refresh\_table()

def add\_payment(self):

student\_id = self.student\_id\_entry.get()

amount = self.amount\_entry.get()

today = datetime.date.today().isoformat()

if student\_id and amount:

db\_handler.add\_payment(student\_id, amount, today)

self.refresh\_table()

else:

messagebox.showerror("Error", "Fill all fields.")

def refresh\_table(self):

for i in self.payment\_table.get\_children():

self.payment\_table.delete(i)

for row in db\_handler.view\_payments():

self.payment\_table.insert("", "end", values=row)

# Run App

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

app = HostelApp()

app.mainloop()