import tkinter as tk

from tkinter import ttk, messagebox

import db\_handler

# Colors

BG\_COLOR = "#d4f4e4" # Mint green

HEADER\_COLOR = "#c6d5f7" # Light purple

TEXT\_COLOR = "#333"

def clear\_window(root):

for widget in root.winfo\_children():

widget.destroy()

def dashboard(root):

clear\_window(root)

tk.Label(root, text="Hostel Management Dashboard", font=("Helvetica", 20, "bold"), bg=HEADER\_COLOR).pack(pady=20)

btn\_frame = tk.Frame(root, bg=BG\_COLOR)

btn\_frame.pack(pady=20)

tk.Button(btn\_frame, text="Manage Students", width=20, command=lambda: manage\_students(root)).grid(row=0, column=0, padx=10, pady=10)

tk.Button(btn\_frame, text="Manage Rooms", width=20, command=lambda: manage\_rooms(root)).grid(row=0, column=1, padx=10, pady=10)

tk.Button(btn\_frame, text="Manage Payments", width=20, command=lambda: manage\_payments(root)).grid(row=0, column=2, padx=10, pady=10)

def manage\_students(root):

clear\_window(root)

tk.Label(root, text="Manage Students", font=("Helvetica", 18, "bold"), bg=HEADER\_COLOR).pack(pady=10)

frame = tk.Frame(root, bg=BG\_COLOR)

frame.pack()

id\_entry = tk.Entry(frame)

name\_entry = tk.Entry(frame)

age\_entry = tk.Entry(frame)

labels = ["ID:", "Name:", "Age:"]

entries = [id\_entry, name\_entry, age\_entry]

for idx, text in enumerate(labels):

tk.Label(frame, text=text, bg=BG\_COLOR).grid(row=idx, column=0, sticky="w", pady=2)

entries[idx].grid(row=idx, column=1, pady=2)

def add\_student():

try:

db\_handler.add\_student(id\_entry.get(), name\_entry.get(), age\_entry.get())

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

refresh\_tree()

except Exception as e:

messagebox.showerror("Error", str(e))

def delete\_selected():

selected = tree.selection()

if selected:

item = tree.item(selected)

student\_id = item['values'][0]

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

refresh\_tree()

def search\_students():

keyword = search\_entry.get()

rows = db\_handler.search\_student(keyword)

update\_tree(rows)

tk.Button(frame, text="Add Student", command=add\_student, bg=HEADER\_COLOR).grid(row=3, column=0, pady=10)

tk.Button(frame, text="Delete Selected", command=delete\_selected, bg=HEADER\_COLOR).grid(row=3, column=1, pady=10)

search\_entry = tk.Entry(root)

search\_entry.pack(pady=5)

tk.Button(root, text="Search", command=search\_students, bg=HEADER\_COLOR).pack(pady=5)

tree\_frame = tk.Frame(root)

tree\_frame.pack()

tree\_scroll = tk.Scrollbar(tree\_frame)

tree\_scroll.pack(side=tk.RIGHT, fill=tk.Y)

tree = ttk.Treeview(tree\_frame, columns=("ID", "Name", "Age", "Room"), show="headings", yscrollcommand=tree\_scroll.set)

tree.heading("ID", text="ID")

tree.heading("Name", text="Name")

tree.heading("Age", text="Age")

tree.heading("Room", text="Room")

tree.pack()

tree\_scroll.config(command=tree.yview)

def refresh\_tree():

rows = db\_handler.view\_students()

update\_tree(rows)

def update\_tree(rows):

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

for row in rows:

tree.insert("", "end", values=row)

refresh\_tree()

tk.Button(root, text="Back to Dashboard", command=lambda: dashboard(root), bg=HEADER\_COLOR).pack(pady=10)

def manage\_rooms(root):

clear\_window(root)

tk.Label(root, text="Manage Rooms", font=("Helvetica", 18, "bold"), bg=HEADER\_COLOR).pack(pady=10)

tree\_frame = tk.Frame(root)

tree\_frame.pack()

tree\_scroll = tk.Scrollbar(tree\_frame)

tree\_scroll.pack(side=tk.RIGHT, fill=tk.Y)

tree = ttk.Treeview(tree\_frame, columns=("Room No", "Capacity", "Occupants"), show="headings", yscrollcommand=tree\_scroll.set)

tree.heading("Room No", text="Room No")

tree.heading("Capacity", text="Capacity")

tree.heading("Occupants", text="Occupants")

tree.pack()

tree\_scroll.config(command=tree.yview)

def refresh\_tree():

rows = db\_handler.view\_rooms()

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

for row in rows:

tree.insert("", "end", values=row)

refresh\_tree()

tk.Button(root, text="Back to Dashboard", command=lambda: dashboard(root), bg=HEADER\_COLOR).pack(pady=10)

def manage\_payments(root):

clear\_window(root)

tk.Label(root, text="Manage Payments", font=("Helvetica", 18, "bold"), bg=HEADER\_COLOR).pack(pady=10)

tree\_frame = tk.Frame(root)

tree\_frame.pack()

tree\_scroll = tk.Scrollbar(tree\_frame)

tree\_scroll.pack(side=tk.RIGHT, fill=tk.Y)

tree = ttk.Treeview(tree\_frame, columns=("Payment ID", "Student ID", "Amount", "Date"), show="headings", yscrollcommand=tree\_scroll.set)

tree.heading("Payment ID", text="Payment ID")

tree.heading("Student ID", text="Student ID")

tree.heading("Amount", text="Amount")

tree.heading("Date", text="Date")

tree.pack()

tree\_scroll.config(command=tree.yview)

def refresh\_tree():

rows = db\_handler.view\_payments()

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

for row in rows:

tree.insert("", "end", values=row)

refresh\_tree()

tk.Button(root, text="Back to Dashboard", command=lambda: dashboard(root), bg=HEADER\_COLOR).pack(pady=10)

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

db\_handler.create\_tables()

root = tk.Tk()

root.title("Hostel Management System")

root.geometry("900x600")

root.configure(bg=BG\_COLOR)

dashboard(root)

root.mainloop()