import tkinter as tk

from tkinter import ttk, messagebox

import db\_handler

db\_handler.connect\_db()

app = tk.Tk()

app.title("Hostel Management System")

app.geometry("850x600")

# — ROOM MANAGEMENT —

def add\_room\_ui():

try:

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

capacity = int(capacity\_entry.get())

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

messagebox.showinfo("Success", "Room added.")

refresh\_rooms()

except Exception as e:

messagebox.showerror("Error", f"Could not add room: {e}")

def refresh\_rooms():

for row in room\_table.get\_children():

room\_table.delete(row)

rooms = db\_handler.fetch\_rooms()

for room in rooms:

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

# — STUDENT MANAGEMENT —

def add\_student\_ui():

name = student\_name\_entry.get()

age = student\_age\_entry.get()

room\_no = student\_room\_entry.get()

if name and age.isdigit() and room\_no.isdigit():

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

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

refresh\_students()

else:

messagebox.showerror("Error", "Invalid input.")

def update\_student\_ui():

sid = student\_id\_entry.get()

name = student\_name\_entry.get()

age = student\_age\_entry.get()

room\_no = student\_room\_entry.get()

if sid.isdigit() and name and age.isdigit() and room\_no.isdigit():

db\_handler.update\_student(int(sid), name, int(age), int(room\_no))

refresh\_students()

messagebox.showinfo("Success", "Student updated.")

def delete\_student\_ui():

selected = student\_table.selection()

if selected:

item = student\_table.item(selected[0])

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

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

refresh\_students()

messagebox.showinfo("Deleted", "Student deleted.")

def refresh\_students():

for row in student\_table.get\_children():

student\_table.delete(row)

students = db\_handler.fetch\_students()

for student in students:

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

def search\_student\_ui():

keyword = search\_entry.get()

results = db\_handler.search\_students(keyword)

student\_table.delete(\*student\_table.get\_children())

for student in results:

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

# — ROOM FORM —

tk.Label(app, text="Room No").grid(row=0, column=0)

room\_no\_entry = tk.Entry(app)

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

tk.Label(app, text="Capacity").grid(row=1, column=0)

capacity\_entry = tk.Entry(app)

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

tk.Button(app, text="Add Room", command=add\_room\_ui).grid(row=1, column=2)

room\_table = ttk.Treeview(app, columns=("ID", "Room", "Capacity", "Occupied"), show="headings")

for col in ("ID", "Room", "Capacity", "Occupied"):

room\_table.heading(col, text=col)

room\_table.column(col, width=100)

room\_table.grid(row=2, column=0, columnspan=3, pady=10)

refresh\_rooms()

# — STUDENT FORM —

tk.Label(app, text="Student ID").grid(row=3, column=0)

student\_id\_entry = tk.Entry(app)

student\_id\_entry.grid(row=3, column=1)

tk.Label(app, text="Name").grid(row=4, column=0)

student\_name\_entry = tk.Entry(app)

student\_name\_entry.grid(row=4, column=1)

tk.Label(app, text="Age").grid(row=5, column=0)

student\_age\_entry = tk.Entry(app)

student\_age\_entry.grid(row=5, column=1)

tk.Label(app, text="Room No").grid(row=6, column=0)

student\_room\_entry = tk.Entry(app)

student\_room\_entry.grid(row=6, column=1)

tk.Button(app, text="Add Student", command=add\_student\_ui).grid(row=7, column=0)

tk.Button(app, text="Update Student", command=update\_student\_ui).grid(row=7, column=1)

tk.Button(app, text="Delete Student", command=delete\_student\_ui).grid(row=7, column=2)

# — SEARCH BAR —

tk.Label(app, text="Search").grid(row=8, column=0)

search\_entry = tk.Entry(app)

search\_entry.grid(row=8, column=1)

tk.Button(app, text="Search", command=search\_student\_ui).grid(row=8, column=2)

# — STUDENT TABLE —

student\_table = ttk.Treeview(app, columns=("ID", "Name", "Age", "Room"), show="headings")

for col in ("ID", "Name", "Age", "Room"):

student\_table.heading(col, text=col)

student\_table.column(col, width=100)

student\_table.grid(row=9, column=0, columnspan=3, pady=10)

refresh\_students()

app.mainloop()