# main.py

import tkinter as tk

from tkinter import ttk, messagebox

import db\_handler

import datetime

# Colors

BG\_COLOR = "#E0F2F1" # Mint green

BTN\_COLOR = "#CE93D8" # Light purple

ENTRY\_COLOR = "#FFFFFF" # White

db\_handler.create\_tables()

class HostelApp:

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

self.root = root

self.root.title("Hostel Management System")

self.root.geometry("1000x650")

self.root.configure(bg=BG\_COLOR)

self.root.resizable(False, False)

self.current\_frame = None

self.login\_screen()

def clear\_frame(self):

if self.current\_frame:

self.current\_frame.destroy()

def login\_screen(self):

self.clear\_frame()

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

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

tk.Label(frame, text="Hostel Management System", font=("Arial", 28, "bold"), bg=BG\_COLOR, fg="black").pack(pady=40)

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

form\_frame.pack()

tk.Label(form\_frame, text="Username:", font=("Arial", 14), bg=BG\_COLOR).grid(row=0, column=0, padx=10, pady=10, sticky="e")

self.username\_entry = tk.Entry(form\_frame, font=("Arial", 14), bg=ENTRY\_COLOR)

self.username\_entry.grid(row=0, column=1, padx=10, pady=10)

tk.Label(form\_frame, text="Password:", font=("Arial", 14), bg=BG\_COLOR).grid(row=1, column=0, padx=10, pady=10, sticky="e")

self.password\_entry = tk.Entry(form\_frame, show="\*", font=("Arial", 14), bg=ENTRY\_COLOR)

self.password\_entry.grid(row=1, column=1, padx=10, pady=10)

tk.Button(frame, text="Login", font=("Arial", 14), bg=BTN\_COLOR, command=self.check\_login).pack(pady=30)

self.current\_frame = frame

def check\_login(self):

user = self.username\_entry.get()

pwd = self.password\_entry.get()

if db\_handler.authenticate(user, pwd):

self.dashboard()

else:

messagebox.showerror("Error", "Invalid Credentials!")

def dashboard(self):

self.clear\_frame()

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

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

tk.Label(frame, text="Main Menu", font=("Arial", 26, "bold"), bg=BG\_COLOR, fg="black").pack(pady=30)

tk.Button(frame, text="Manage Students", font=("Arial", 16), bg=BTN\_COLOR, width=30, height=2, command=self.manage\_students).pack(pady=15)

tk.Button(frame, text="Manage Rooms", font=("Arial", 16), bg=BTN\_COLOR, width=30, height=2, command=self.manage\_rooms).pack(pady=15)

tk.Button(frame, text="Manage Payments", font=("Arial", 16), bg=BTN\_COLOR, width=30, height=2, command=self.manage\_payments).pack(pady=15)

tk.Button(frame, text="Logout", font=("Arial", 14), bg=BTN\_COLOR, width=15, command=self.login\_screen).pack(pady=30)

self.current\_frame = frame

def manage\_students(self):

self.clear\_frame()

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

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

tk.Label(frame, text="Manage Students", font=("Arial", 24, "bold"), bg=BG\_COLOR, fg="black").pack(pady=20)

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

form\_frame.pack()

tk.Label(form\_frame, text="Student ID:", font=("Arial", 12), bg=BG\_COLOR).grid(row=0, column=0, padx=5, pady=5, sticky="e")

id\_entry = tk.Entry(form\_frame, font=("Arial", 12), bg=ENTRY\_COLOR)

id\_entry.grid(row=0, column=1, padx=5, pady=5)

tk.Label(form\_frame, text="Name:", font=("Arial", 12), bg=BG\_COLOR).grid(row=1, column=0, padx=5, pady=5, sticky="e")

name\_entry = tk.Entry(form\_frame, font=("Arial", 12), bg=ENTRY\_COLOR)

name\_entry.grid(row=1, column=1, padx=5, pady=5)

tk.Label(form\_frame, text="Age:", font=("Arial", 12), bg=BG\_COLOR).grid(row=0, column=2, padx=5, pady=5, sticky="e")

age\_entry = tk.Entry(form\_frame, font=("Arial", 12), bg=ENTRY\_COLOR)

age\_entry.grid(row=0, column=3, padx=5, pady=5)

tk.Label(form\_frame, text="Course:", font=("Arial", 12), bg=BG\_COLOR).grid(row=1, column=2, padx=5, pady=5, sticky="e")

course\_entry = tk.Entry(form\_frame, font=("Arial", 12), bg=ENTRY\_COLOR)

course\_entry.grid(row=1, column=3, padx=5, pady=5)

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

search\_frame.pack(pady=10)

tk.Label(search\_frame, text="Search:", font=("Arial", 12), bg=BG\_COLOR).pack(side="left", padx=5)

search\_entry = tk.Entry(search\_frame, font=("Arial", 12), bg=ENTRY\_COLOR, width=30)

search\_entry.pack(side="left", padx=5)

tree = ttk.Treeview(frame, columns=("ID", "Name", "Age", "Course", "Room"), show="headings")

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

tree.heading(col, text=col)

tree.column(col, width=150)

tree.pack(fill="both", expand=True, padx=20, pady=10)

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

button\_frame.pack(pady=10)

def refresh\_students():

for i in tree.get\_children():

tree.delete(i)

for student in db\_handler.get\_students():

tree.insert('', 'end', values=student)

def add\_student():

sid = id\_entry.get()

name = name\_entry.get()

try:

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

except ValueError:

messagebox.showerror("Error", "Age must be a number!")

return

course = course\_entry.get()

if not (sid and name and course):

messagebox.showerror("Error", "All fields required!")

return

success, msg = db\_handler.add\_student(sid, name, age, course)

if success:

messagebox.showinfo("Success", msg)

refresh\_students()

else:

messagebox.showerror("Error", msg)

def delete\_student():

selected = tree.selection()

if selected:

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

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

refresh\_students()

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

def search\_student():

query = search\_entry.get()

for i in tree.get\_children():

tree.delete(i)

for student in db\_handler.search\_students(query):

tree.insert('', 'end', values=student)

tk.Button(button\_frame, text="Add Student", font=("Arial", 12), bg=BTN\_COLOR, command=add\_student).pack(side="left", padx=10)

tk.Button(button\_frame, text="Search", font=("Arial", 12), bg=BTN\_COLOR, command=search\_student).pack(side="left", padx=10)

tk.Button(button\_frame, text="Delete Student", font=("Arial", 12), bg=BTN\_COLOR, command=delete\_student).pack(side="left", padx=10)

tk.Button(button\_frame, text="Back", font=("Arial", 12), bg=BTN\_COLOR, command=self.dashboard).pack(side="left", padx=10)

refresh\_students()

self.current\_frame = frame

def manage\_rooms(self):

self.clear\_frame()

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

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

tk.Label(frame, text="Manage Rooms", font=("Arial", 24, "bold"), bg=BG\_COLOR).pack(pady=20)

tree = ttk.Treeview(frame, columns=("Room", "Capacity", "Occupants"), show="headings")

for col in ("Room", "Capacity", "Occupants"):

tree.heading(col, text=col)

tree.column(col, width=300)

tree.pack(fill="both", expand=True, padx=20, pady=10)

for room in db\_handler.get\_rooms():

tree.insert('', 'end', values=room)

tk.Button(frame, text="Back", font=("Arial", 14), bg=BTN\_COLOR, command=self.dashboard).pack(pady=20)

self.current\_frame = frame

def manage\_payments(self):

self.clear\_frame()

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

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

tk.Label(frame, text="Manage Payments", font=("Arial", 24, "bold"), bg=BG\_COLOR).pack(pady=20)

sid\_entry = tk.Entry(frame, font=("Arial", 12), bg=ENTRY\_COLOR)

amount\_entry = tk.Entry(frame, font=("Arial", 12), bg=ENTRY\_COLOR)

tk.Label(frame, text="Student ID:", font=("Arial", 12), bg=BG\_COLOR).pack()

sid\_entry.pack()

tk.Label(frame, text="Amount:", font=("Arial", 12), bg=BG\_COLOR).pack()

amount\_entry.pack()

tree = ttk.Treeview(frame, columns=("PID", "SID", "Amount", "Date"), show="headings")

for col in ("PID", "SID", "Amount", "Date"):

tree.heading(col, text=col)

tree.column(col, width=200)

tree.pack(fill="both", expand=True, padx=20, pady=10)

def refresh\_payments():

for i in tree.get\_children():

tree.delete(i)

for payment in db\_handler.get\_payments():

tree.insert('', 'end', values=payment)

def add\_payment():

try:

sid = sid\_entry.get()

amount = float(amount\_entry.get())

date = datetime.date.today().strftime("%Y-%m-%d")

db\_handler.add\_payment(sid, amount, date)

refresh\_payments()

messagebox.showinfo("Success", "Payment Added")

except ValueError:

messagebox.showerror("Error", "Amount must be number!")

tk.Button(frame, text="Add Payment", font=("Arial", 12), bg=BTN\_COLOR, command=add\_payment).pack(pady=10)

tk.Button(frame, text="Back", font=("Arial", 12), bg=BTN\_COLOR, command=self.dashboard).pack(pady=10)

refresh\_payments()

self.current\_frame = frame

# Run

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

root = tk.Tk()

app = HostelApp(root)

root.mainloop()