# 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()

# Login Window

def login\_window():

def verify\_login():

user = username\_entry.get()

pwd = password\_entry.get()

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

login.destroy()

dashboard()

else:

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

login = tk.Tk()

login.title("Hostel Management Login")

login.geometry("400x300")

login.configure(bg=BG\_COLOR)

tk.Label(login, text="Username:", bg=BG\_COLOR).pack(pady=10)

username\_entry = tk.Entry(login, bg=ENTRY\_COLOR)

username\_entry.pack()

tk.Label(login, text="Password:", bg=BG\_COLOR).pack(pady=10)

password\_entry = tk.Entry(login, show="\*", bg=ENTRY\_COLOR)

password\_entry.pack()

tk.Button(login, text="Login", bg=BTN\_COLOR, command=verify\_login).pack(pady=20)

login.mainloop()

# Dashboard

def dashboard():

dash = tk.Tk()

dash.title("Hostel Management System")

dash.geometry("600x400")

dash.configure(bg=BG\_COLOR)

tk.Button(dash, text="Manage Students", bg=BTN\_COLOR, width=30, command=manage\_students).pack(pady=20)

tk.Button(dash, text="Manage Rooms", bg=BTN\_COLOR, width=30, command=manage\_rooms).pack(pady=20)

tk.Button(dash, text="Manage Payments", bg=BTN\_COLOR, width=30, command=manage\_payments).pack(pady=20)

dash.mainloop()

# Students Management

def manage\_students():

win = tk.Toplevel()

win.title("Manage Students")

win.geometry("800x500")

win.configure(bg=BG\_COLOR)

def refresh():

for i in tree.get\_children():

tree.delete(i)

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

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

def add\_student():

student\_id = id\_entry.get()

name = name\_entry.get()

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

course = course\_entry.get()

if not (student\_id and name and course):

messagebox.showerror("Error", "Please fill all fields!")

return

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

if success:

messagebox.showinfo("Success", msg)

refresh()

else:

messagebox.showerror("Error", msg)

def search():

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)

def delete\_student():

selected = tree.selection()

if selected:

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

db\_handler.delete\_student(student)

refresh()

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

# Entries

id\_entry = tk.Entry(win, bg=ENTRY\_COLOR)

name\_entry = tk.Entry(win, bg=ENTRY\_COLOR)

age\_entry = tk.Entry(win, bg=ENTRY\_COLOR)

course\_entry = tk.Entry(win, bg=ENTRY\_COLOR)

tk.Label(win, text="Student ID:", bg=BG\_COLOR).pack()

id\_entry.pack()

tk.Label(win, text="Name:", bg=BG\_COLOR).pack()

name\_entry.pack()

tk.Label(win, text="Age:", bg=BG\_COLOR).pack()

age\_entry.pack()

tk.Label(win, text="Course:", bg=BG\_COLOR).pack()

course\_entry.pack()

tk.Button(win, text="Add Student", bg=BTN\_COLOR, command=add\_student).pack(pady=10)

search\_entry = tk.Entry(win, bg=ENTRY\_COLOR)

search\_entry.pack()

tk.Button(win, text="Search", bg=BTN\_COLOR, command=search).pack(pady=5)

tree = ttk.Treeview(win, columns=("ID", "Name", "Age", "Course", "Room No."), show='headings')

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

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

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

tree.heading('Course', text="Course")

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

tree.pack(pady=10, fill='both', expand=True)

tk.Button(win, text="Delete Student", bg=BTN\_COLOR, command=delete\_student).pack(pady=10)

refresh()

# Manage Rooms

def manage\_rooms():

win = tk.Toplevel()

win.title("Manage Rooms")

win.geometry("500x400")

win.configure(bg=BG\_COLOR)

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

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

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

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

tree.pack(pady=20, fill='both', expand=True)

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

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

# Manage Payments

def manage\_payments():

win = tk.Toplevel()

win.title("Manage Payments")

win.geometry("600x400")

win.configure(bg=BG\_COLOR)

def refresh():

for i in tree.get\_children():

tree.delete(i)

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

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

def add\_payment():

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()

messagebox.showinfo("Success", "Payment recorded!")

sid\_entry = tk.Entry(win, bg=ENTRY\_COLOR)

amount\_entry = tk.Entry(win, bg=ENTRY\_COLOR)

tk.Label(win, text="Student ID:", bg=BG\_COLOR).pack()

sid\_entry.pack()

tk.Label(win, text="Amount:", bg=BG\_COLOR).pack()

amount\_entry.pack()

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

tree = ttk.Treeview(win, columns=("Payment ID", "Student ID", "Amount", "Date"), show='headings')

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(pady=10, fill='both', expand=True)

refresh()

# Start

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

login\_window()