import tkinter as tk  
from tkinter import messagebox  
  
  
class RailwayReservation:  
 def \_\_init\_\_(self, root):  
 self.root = root  
 self.root.title("Railway Reservation System")  
 self.root.geometry("500x600")  
  
 # Store all bookings in a list  
 self.bookings = []  
  
 # Title Label  
 tk.Label(root, text="Railway Reservation System", font=("Arial", 16, "bold")).pack(pady=10)  
  
 # From Label and Entry  
 tk.Label(root, text="From", font=("Arial", 12)).pack(pady=5)  
 self.from\_station = tk.Entry(root, width=30, font=("Arial", 12))  
 self.from\_station.pack()  
  
 # Destination Label and Entry  
 tk.Label(root, text="Destination", font=("Arial", 12)).pack(pady=5)  
 self.destination = tk.Entry(root, width=30, font=("Arial", 12))  
 self.destination.pack()  
  
 # Number of Tickets Label and Entry  
 tk.Label(root, text="Number of Tickets", font=("Arial", 12)).pack(pady=5)  
 self.num\_tickets = tk.Entry(root, width=10, font=("Arial", 12))  
 self.num\_tickets.pack()  
  
 # Class of Travel Dropdown  
 tk.Label(root, text="Class of Travel", font=("Arial", 12)).pack(pady=5)  
 self.class\_var = tk.StringVar(root)  
 self.class\_var.set("sleeper coach") # Default value  
 tk.OptionMenu(root, self.class\_var, "sleeper coach", "3rd AC", "1st AC").pack()  
  
 # Name Label and Entry  
 tk.Label(root, text="Name", font=("Arial", 12)).pack(pady=5)  
 self.name = tk.Entry(root, width=30, font=("Arial", 12))  
 self.name.pack()  
  
 # Contact Number Label and Entry  
 tk.Label(root, text="Contact Number", font=("Arial", 12)).pack(pady=5)  
 self.contact = tk.Entry(root, width=30, font=("Arial", 12))  
 self.contact.pack()  
  
 # Book Ticket Button  
 tk.Button(root, text="Book Ticket", font=("Arial", 12, "bold"), command=self.book\_ticket).pack(pady=10)  
  
 # Cancel Ticket Button  
 tk.Button(root, text="Cancel Ticket", font=("Arial", 12, "bold"), command=self.cancel\_ticket).pack(pady=5)  
  
 # View Booked Tickets Button  
 tk.Button(root, text="View Booked Tickets", font=("Arial", 12, "bold"), command=self.view\_bookings).pack(pady=5)  
  
 def book\_ticket(self):  
 # Retrieve data  
 from\_station = self.from\_station.get()  
 destination = self.destination.get()  
 num\_tickets = self.num\_tickets.get()  
 travel\_class = self.class\_var.get()  
 name = self.name.get()  
 contact = self.contact.get()  
  
 # Basic validation  
 if not (from\_station and destination and num\_tickets and name and contact):  
 messagebox.showwarning("Incomplete Details", "Please fill in all details.")  
 return  
  
 try:  
 num\_tickets = int(num\_tickets)  
 if num\_tickets <= 0:  
 raise ValueError  
 except ValueError:  
 messagebox.showerror("Invalid Input", "Please enter a valid number of tickets.")  
 return  
  
 # Save booking details  
 booking = {  
 "name": name,  
 "from": from\_station,  
 "destination": destination,  
 "num\_tickets": num\_tickets,  
 "class": travel\_class,  
 "contact": contact  
 }  
 self.bookings.append(booking)  
  
 # Clear form fields  
 self.clear\_fields()  
 messagebox.showinfo("Booking Confirmation", "Ticket booked successfully!")  
  
 def cancel\_ticket(self):  
 # Retrieve contact number for identification  
 contact = self.contact.get()  
 if not contact:  
 messagebox.showwarning("Missing Information", "Please enter a contact number to cancel the booking.")  
 return  
  
 # Find and remove booking by contact  
 for booking in self.bookings:  
 if booking["contact"] == contact:  
 self.bookings.remove(booking)  
 messagebox.showinfo("Booking Canceled", f"Booking for {contact} has been canceled.")  
 self.clear\_fields()  
 return  
  
 messagebox.showerror("Not Found", "No booking found with the provided contact number.")  
  
 def view\_bookings(self):  
 if not self.bookings:  
 messagebox.showinfo("No Bookings", "No tickets booked yet.")  
 return  
  
 # Display all bookings  
 booking\_details = ""  
 for i, booking in enumerate(self.bookings, 1):  
 booking\_details += f"Booking {i}:\n" \  
 f"Name: {booking['name']}\n" \  
 f"From: {booking['from']}\n" \  
 f"Destination: {booking['destination']}\n" \  
 f"Tickets: {booking['num\_tickets']}\n" \  
 f"Class: {booking['class']}\n" \  
 f"Contact: {booking['contact']}\n\n"  
  
 # Show booking details in a separate window  
 top = tk.Toplevel(self.root)  
 top.title("Booked Tickets")  
 top.geometry("400x300")  
  
 tk.Label(top, text="Booked Tickets", font=("Arial", 14, "bold")).pack(pady=10)  
  
 # Add Text widget for displaying bookings  
 text = tk.Text(top, font=("Arial", 10), wrap="word", height=15, bg="light yellow")  
 text.insert(tk.END, booking\_details)  
 text.config(state="disabled") # Make the text widget read-only  
 text.pack(pady=5)  
  
 def clear\_fields(self):  
 self.from\_station.delete(0, tk.END)  
 self.destination.delete(0, tk.END)  
 self.num\_tickets.delete(0, tk.END)  
 self.name.delete(0, tk.END)  
 self.contact.delete(0, tk.END)  
 self.class\_var.set("sleeper coach")  
  
  
# Run the application  
if \_\_name\_\_ == "\_\_main\_\_":  
 root = tk.Tk()  
 app = RailwayReservation(root)  
 root.mainloop()