**TASK5-CONTACT BOOK**

**PROJECT TITLE:**

**Contact Book Management System**

**DESCRIPTION:  
 A GUI-based contact management system using CustomTkinter that allows users to store, update, delete, and view contact information.  
  
FEATURES:  
1.Input fields for name, phone, email, and address.  
2.Validation for phone number format.  
3.Add, update, delete operations with data persistence in JSON.  
4.Search and filter contacts by name.  
5.All data stored in a JSON file.  
  
TECHNOLOGIES USED:  
1.Python  
2.CustomTkinter  
3.JSON, OS modules**

**TARGET USERS:  
1.Students and users looking for digital contact management.**

**2.Demonstration for beginner-level GUI and file I/O in Python.**

**APPENDIX:**

**Contact\_Book.py**

**import customtkinter as ctk**

**import tkinter.messagebox as msg**

**import json**

**import os**

**Contact\_file="contacts.json"**

**#for loading the contact**

**def load\_contacts():**

**if not os.path.exists(Contact\_file):**

**return []**

**with open(Contact\_file,'r') as f:**

**return json.load(f)**

**# for Saving the contact**

**def save\_contacts(data):**

**with open(Contact\_file,'w') as f:**

**json.dump(data,f,indent=4)**

**#Creating a class for Contact Book App**

**class contactbook(ctk.CTk):**

**def \_\_init\_\_(self):**

**super().\_\_init\_\_()**

**self.title("CONTACT BOOK")**

**self.geometry("900x600")**

**self.resizable(False,False)**

**ctk.set\_appearance\_mode("dark")**

**ctk.set\_default\_color\_theme("dark-blue")**

**self.contacts = load\_contacts()**

**self.create\_ui()**

**#Creating Functions**

**def clear\_fields(self):**

**self.name\_entry.delete(0,'end')**

**self.phone\_entry.delete(0,'end')**

**self.email\_entry.delete(0,'end')**

**self.address\_entry.delete(0,'end')**

**def add\_contact(self):**

**name=self.name\_entry.get().strip()**

**phone=self.phone\_entry.get().strip()**

**email=self.email\_entry.get().strip()**

**address=self.address\_entry.get().strip()**

**if not name or not phone:**

**msg.showwarning("Missing Info","Name and Phone are required.")**

**return**

**if not phone.isdigit() or len(phone)!=10:**

**msg.showerror("Invalid Phone","Phone Number must be 10 digits.")**

**return**

**for contact in self.contacts:**

**if contact["name"]==name or contact['phone']==phone:**

**msg.showerror("Duplicate","Contact with same name or phone exits.")**

**return**

**self.contacts.append({**

**'name':name,**

**'phone':phone,**

**'email':email,**

**"address":address**

**})**

**save\_contacts(self.contacts)**

**msg.showinfo("Success","Contacts Added.")**

**self.clear\_fields()**

**def update\_contact(self):**

**name=self.name\_entry.get().strip()**

**phone=self.phone\_entry.get().strip()**

**for contact in self.contacts:**

**if contact['name']==name:**

**contact['phone']=phone**

**contact['email']=self.email\_entry.get().strip()**

**contact['address']=self.address\_entry.get().strip()**

**save\_contacts(self.contacts)**

**msg.showinfo("Updated",f"Contact {name} updated.")**

**return**

**msg.showerror("Not Found","Contact not found.")**

**def delete\_contact(self):**

**name=self.name\_entry.get().strip()**

**for contact in self.contacts:**

**if contact['name']==name:**

**self.contacts.remove(contact)**

**save\_contacts(self.contacts)**

**msg.showinfo("Deleted",f"Contact {name} deleted.")**

**self.clear\_fields()**

**return**

**msg.showerror("Not Found","Contact not found.")**

**def show\_all\_contacts(self):**

**popup=ctk.CTkToplevel(self)**

**popup.title("All Contacts")**

**popup.geometry("800x500")**

**popup.grab\_set()**

**def filter\_Contacts(event=None):**

**search\_text=search\_entry.get().strip().lower()**

**display.delete('0.0',"end")**

**filtered = self.contacts if not search\_text else [c for c in self.contacts if search\_text in c['name'].lower()]**

**if filtered:**

**for c in filtered:**

**display.insert("end", f"Name:{c['name']}\nPhone: {c['phone']}\nEmail: {c['email']}\nAddress: {c['address']}\n\n")**

**else:**

**display.insert("end","No Contact found.")**

**search\_entry=ctk.CTkEntry(popup,placeholder\_text="Type Name to search....",width=400)**

**search\_entry.pack(pady=10)**

**search\_entry.bind("<KeyRelease>",filter\_Contacts)**

**display= ctk.CTkTextbox(popup,width=700,height=400,font=("Courier",14))**

**display.pack(padx=20,pady=10)**

**filter\_Contacts()**

**def create\_ui(self):**

**self.frame=ctk.CTkFrame(self,border\_width=2,border\_color="#5e35b1",corner\_radius=10,height=600,width=500)**

**self.frame.place(relx=0.5,rely=0.5,anchor="center")**

**ctk.CTkLabel(self.frame,text="CONTACT BOOK",text\_color="#5e35b1",font=("Helvetica",22,"bold")).grid(row=0,column=0,columnspan=2,pady=(20,10))**

**ctk.CTkLabel(self.frame,text="Name:",width=100,anchor='e').grid(row=1,column=0,padx=10,pady=10)**

**self.name\_entry=ctk.CTkEntry(self.frame,placeholder\_text="Contact Name",width=300)**

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

**ctk.CTkLabel(self.frame,text="Phone:",width=100,anchor="e").grid(row=2,column=0,padx=10,pady=10)**

**self.phone\_entry=ctk.CTkEntry(self.frame,placeholder\_text="Contact Phone",width=300)**

**self.phone\_entry.grid(row=2,column=1,padx=10,pady=10)**

**ctk.CTkLabel(self.frame,text="Email:",width=100,anchor='e').grid(row=3,column=0,padx=10,pady=10)**

**self.email\_entry =ctk.CTkEntry(self.frame,placeholder\_text="Contact Email",width=300)**

**self.email\_entry.grid(row=3,column=1,padx=10,pady=10)**

**ctk.CTkLabel(self.frame,text="Address:",width=100,anchor="e").grid(row=4,column=0,padx=10,pady=10)**

**self.address\_entry=ctk.CTkEntry(self.frame,placeholder\_text="Contact Address",width=300)**

**self.address\_entry.grid(row=4,column=1,padx=10,pady=10)**

**#Creating Buttons**

**button\_frame=ctk.CTkFrame(self.frame,fg\_color="transparent")**

**button\_frame.grid(row=5,column=0,columnspan=2,pady=20,padx=20)**

**self.frame.grid\_propagate(False)**

**ctk.CTkButton(button\_frame,text="Add Contact",fg\_color="#5e35b1",hover\_color='#009933',command= self.add\_contact,width=130).pack(side="left",padx=10)**

**ctk.CTkButton(button\_frame,text="Update Contact",fg\_color="#5e35b1",hover\_color='#0066cc',command= self.update\_contact,width=130).pack(side="left",padx=10)**

**ctk.CTkButton(button\_frame,text="Delete Contact",fg\_color="#5e35b1",hover\_color="#cc0000",command= self.delete\_contact,width=130).pack(side="left",padx=10)**

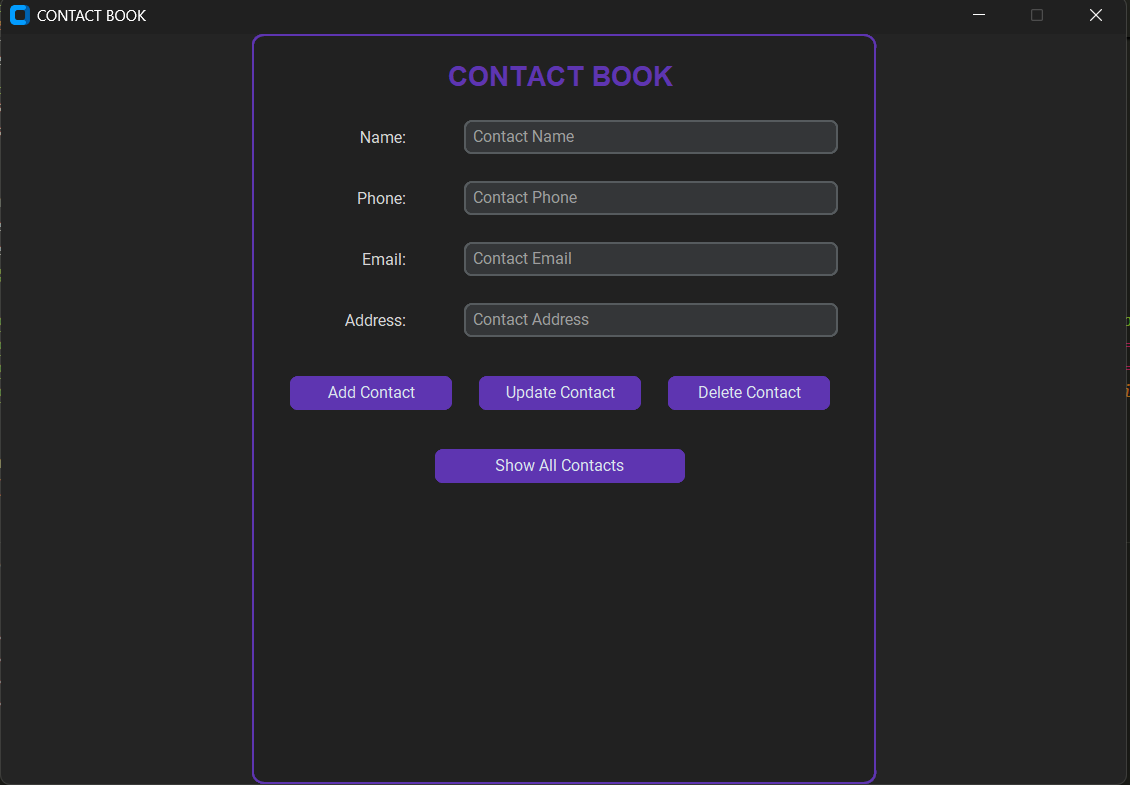
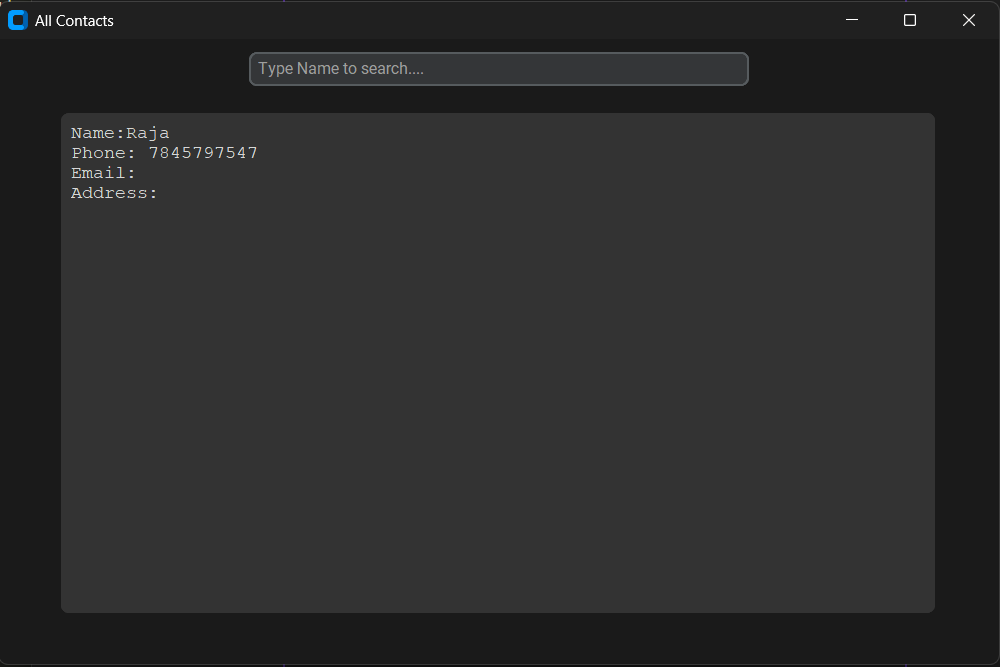
**ctk.CTkButton(self.frame,text="Show All Contacts",fg\_color="#5e35b1",hover\_color='#7e57c2',command= self.show\_all\_contacts,width=200).grid(row=6,column=0,columnspan=2,pady=10)**

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

**app=contactbook()**

**app.mainloop()**

**OUTPUTS:**

****