In [1]:

```
import tkinter as tk
from tkinter import messagebox

class ContactBook:
    def __init__(self, root):
        self.root = root
        self.root.title("Contact Book")

        self.contacts = []

        self.name_label = tk.Label(root, text="Name:")
        self.name_label.grid(row=0, column=0, padx=5, pady=5)
        self.name_entry = tk.Entry(root)
        self.name_entry.grid(row=0, column=1, padx=5, pady=5)

        self.phone_label = tk.Label(root, text="Phone:")
        self.phone_label.grid(row=1, column=0, padx=5, pady=5)
        self.phone_entry = tk.Entry(root)
        self.phone_entry.grid(row=1, column=1, padx=5, pady=5)

        self.email_label = tk.Label(root, text="Email:")
        self.email_label.grid(row=2, column=0, padx=5, pady=5)
        self.email_entry = tk.Entry(root)
        self.email_entry.grid(row=2, column=1, padx=5, pady=5)

        self.add_button = tk.Button(root, text="Add Contact", command=self.add_contact)
        self.add_button.grid(row=3, column=0, padx=5, pady=5)

        self.view_button = tk.Button(root, text="View Contacts", command=self.view_contacts)
        self.view_button.grid(row=3, column=1, padx=5, pady=5)

        self.search_label = tk.Label(root, text="Search:")
        self.search_label.grid(row=4, column=0, padx=5, pady=5)
        self.search_entry = tk.Entry(root)
        self.search_entry.grid(row=4, column=1, padx=5, pady=5)

        self.search_button = tk.Button(root, text="Search", command=self.search_contact)
        self.search_button.grid(row=5, column=0, padx=5, pady=5)

        self.update_button = tk.Button(root, text="Update Contact", command=self.update_contact)
        self.update_button.grid(row=5, column=1, padx=5, pady=5)

        self.delete_button = tk.Button(root, text="Delete Contact", command=self.delete_contact)
        self.delete_button.grid(row=6, column=0, padx=5, pady=5)

    def add_contact(self):
        name = self.name_entry.get()
        phone = self.phone_entry.get()
        email = self.email_entry.get()

        if name and phone and email:
            contact = {"name": name, "phone": phone, "email": email}
            self.contacts.append(contact)
            messagebox.showinfo("Success", "Contact added successfully!")
            self.clear_entries()
        else:
            messagebox.showerror("Error", "Please fill in all fields.")

    def view_contacts(self):
        if self.contacts:
            contact_list = ""
            for contact in self.contacts:
                contact_list += f"Name: {contact['name']}, Phone: {contact['phone']}, Email: {contact['email']}
            messagebox.showinfo("Contacts", contact_list)
        else:
            messagebox.showinfo("Contacts", "No contacts found.")

    def search_contact(self):
        search_term = self.search_entry.get()
        if search_term:
            found_contacts = ""
            for contact in self.contacts:
                if search_term.lower() in contact["name"].lower():
                    found_contacts += f"Name: {contact['name']}, Phone: {contact['phone']}, Email: {contact['em
            if found_contacts:
                messagebox.showinfo("Search Results", found_contacts)
            else:
                messagebox.showinfo("Search Results", "No contacts found.")
        else:
            messagebox.showerror("Error", "Please enter a search term.")

    def update_contact(self):
        name = self.name_entry.get()
        phone = self.phone_entry.get()
        email = self.email_entry.get()

        if name and phone and email:
            updated = False
            for contact in self.contacts:
                if name.lower() == contact["name"].lower():
                    contact["phone"] = phone
                    contact["email"] = email
                    updated = True
                    break
            if updated:
                messagebox.showinfo("Success", "Contact updated successfully!")
                self.clear_entries()
            else:
                messagebox.showinfo("Update Contact", "Contact not found.")
        else:
            messagebox.showerror("Error", "Please fill in all fields.")

    def delete_contact(self):
        name = self.name_entry.get()
        if name:
            deleted = False
            for contact in self.contacts:
                if name.lower() == contact["name"].lower():
                    self.contacts.remove(contact)
                    deleted = True
                    break
            if deleted:
                messagebox.showinfo("Success", "Contact deleted successfully!")
                self.clear_entries()
            else:
                messagebox.showinfo("Delete Contact", "Contact not found.")
        else:
            messagebox.showerror("Error", "Please enter a name.")

    def clear_entries(self):
        self.name_entry.delete(0, tk.END)
        self.phone_entry.delete(0, tk.END)
        self.email_entry.delete(0, tk.END)

root = tk.Tk()
contact_book = ContactBook(root)
root.mainloop()
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js