

```
In [3]: class Contact:
    def __init__(self, store_name, phone_number, email, address):
        self.store_name = store_name
        self.phone_number = phone_number
        self.email = email
        self.address = address

class ContactManager:
    def __init__(self):
        self.contacts = []

    def add_contact(self, contact):
        self.contacts.append(contact)

    def view_contacts(self):
        return self.contacts

    def search_contact(self, search_term):
        return [contact for contact in self.contacts if search_term in contact.store_name or search_term in con

    def update_contact(self, index, updated_contact):
        self.contacts[index] = updated_contact

    def delete_contact(self, index):
        del self.contacts[index]
```

```
In [5]: import tkinter as tk
from tkinter import messagebox

class ContactApp:
    def __init__(self, root, contact_manager):
        self.root = root
        self.contact_manager = contact_manager
        self.root.title("Contact Management System")

        self.store_name_var = tk.StringVar()
        self.phone_number_var = tk.StringVar()
        self.email_var = tk.StringVar()
        self.address_var = tk.StringVar()
        self.search_var = tk.StringVar()

        self.create_widgets()

    def create_widgets(self):
        tk.Label(self.root, text="Store Name").grid(row=0, column=0)
        tk.Entry(self.root, textvariable=self.store_name_var).grid(row=0, column=1)
        tk.Label(self.root, text="Phone Number").grid(row=1, column=0)
        tk.Entry(self.root, textvariable=self.phone_number_var).grid(row=1, column=1)
        tk.Label(self.root, text="Email").grid(row=2, column=0)
        tk.Entry(self.root, textvariable=self.email_var).grid(row=2, column=1)
        tk.Label(self.root, text="Address").grid(row=3, column=0)
        tk.Entry(self.root, textvariable=self.address_var).grid(row=3, column=1)

        tk.Button(self.root, text="Add Contact", command=self.add_contact).grid(row=4, column=0, columnspan=2)

        self.contact_listbox = tk.Listbox(self.root)
        self.contact_listbox.grid(row=5, column=0, columnspan=2)

        tk.Label(self.root, text="Search").grid(row=6, column=0)
        tk.Entry(self.root, textvariable=self.search_var).grid(row=6, column=1)
        tk.Button(self.root, text="Search", command=self.search_contact).grid(row=7, column=0, columnspan=2)

        tk.Button(self.root, text="Update Contact", command=self.update_contact).grid(row=8, column=0, columnspan=2)
        tk.Button(self.root, text="Delete Contact", command=self.delete_contact).grid(row=9, column=0, columnspan=2)

    def add_contact(self):
        contact = Contact(
            self.store_name_var.get(),
            self.phone_number_var.get(),
            self.email_var.get(),
            self.address_var.get()
        )
        self.contact_manager.add_contact(contact)
        self.refresh_contact_list()

    def refresh_contact_list(self):
        self.contact_listbox.delete(0, tk.END)
        for contact in self.contact_manager.view_contacts():
            self.contact_listbox.insert(tk.END, f"{contact.store_name} - {contact.phone_number}")

    def search_contact(self):
        search_term = self.search_var.get()
        results = self.contact_manager.search_contact(search_term)
        self.contact_listbox.delete(0, tk.END)
        for contact in results:
            self.contact_listbox.insert(tk.END, f"{contact.store_name} - {contact.phone_number}")
```

```
def update_contact(self):
    selected_index = self.contact_listbox.curselection()[0]
    contact = Contact(
        self.store_name_var.get(),
        self.phone_number_var.get(),
        self.email_var.get(),
        self.address_var.get()
    )
    self.contact_manager.update_contact(selected_index, contact)
    self.refresh_contact_list()

def delete_contact(self):
    selected_index = self.contact_listbox.curselection()[0]
    self.contact_manager.delete_contact(selected_index)
    self.refresh_contact_list()
```

```
root = tk.Tk()
contact_manager = ContactManager()
app = ContactApp(root, contact_manager)
root.mainloop()
```

In []:

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