```python
from tkinter import *
from tkinter import messagebox
import sqlite3

# Connect to database
conn = sqlite3.connect("contacts.db")
cursor = conn.cursor()
cursor.execute("""
    CREATE TABLE IF NOT EXISTS contacts (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT,
        phone TEXT,
        email TEXT,
        address TEXT
    )
""")
conn.commit()

# Functions
def add_contact():
    name = entry_name.get()
    phone = entry_phone.get()
    email = entry_email.get()
    address = entry_address.get()

    if name and phone:
        cursor.execute("INSERT INTO contacts (name, phone, email, address) VALUES (?, ?, ?, ?)",
                       (name, phone, email, address))
        conn.commit()
        messagebox.showinfo("Success", "Contact Added")
        clear_fields()
        view_contacts()
    else:
        messagebox.showwarning("Missing Info", "Name and Phone are required")

def view_contacts():
    listbox.delete(0, END)
    cursor.execute("SELECT name, phone FROM contacts")
    for row in cursor.fetchall():
        listbox.insert(END, f"{row[0]} | {row[1]}")

def search_contact():
    query = entry_search.get()
    listbox.delete(0, END)
    cursor.execute("SELECT name, phone FROM contacts WHERE name LIKE ? OR phone LIKE ?",
                   (f'%{query}%', f'%{query}%'))
    for row in cursor.fetchall():
        listbox.insert(END, f"{row[0]} | {row[1]}")

def select_contact(event):
    try:
        selected = listbox.get(listbox.curselection())
        name = selected.split('|')[0].strip()
        cursor.execute("SELECT * FROM contacts WHERE name = ?", (name,))
        contact = cursor.fetchone()

        entry_name.delete(0, END)
        entry_phone.delete(0, END)
        entry_email.delete(0, END)
        entry_address.delete(0, END)

        entry_name.insert(0, contact[1])
        entry_phone.insert(0, contact[2])
        entry_email.insert(0, contact[3])
        entry_address.insert(0, contact[4])
    except:
        pass

def update_contact():
    name = entry_name.get()
    phone = entry_phone.get()
    email = entry_email.get()
    address = entry_address.get()

    cursor.execute("UPDATE contacts SET phone=?, email=?, address=? WHERE name=?",
                   (phone, email, address, name))
    conn.commit()
    messagebox.showinfo("Updated", "Contact Updated")
    view_contacts()

def delete_contact():
    name = entry_name.get()
    if name:
        cursor.execute("DELETE FROM contacts WHERE name=?", (name,))
        conn.commit()
        messagebox.showinfo("Deleted", "Contact Deleted")
        clear_fields()
        view_contacts()
    else:
        messagebox.showwarning("Missing Info", "Select a contact to delete")

def clear_fields():
    entry_name.delete(0, END)
    entry_phone.delete(0, END)
    entry_email.delete(0, END)
    entry_address.delete(0, END)

# GUI Window
root = Tk()
root.title("Contact Management System")
root.geometry("500x500")

# Labels and Entries
Label(root, text="Name").pack()
entry_name = Entry(root)
entry_name.pack()
```

```
Label(root, text="Phone").pack()
entry_phone = Entry(root)
entry_phone.pack()

Label(root, text="Email").pack()
entry_email = Entry(root)
entry_email.pack()

Label(root, text="Address").pack()
entry_address = Entry(root)
entry_address.pack()

# Buttons
Button(root, text="Add Contact", command=add_contact).pack(pady=2)
Button(root, text="Update Contact", command=update_contact).pack(pady=2)
Button(root, text="Delete Contact", command=delete_contact).pack(pady=2)
Button(root, text="Clear Fields", command=clear_fields).pack(pady=2)

# Search
Label(root, text="Search by Name or Phone").pack(pady=5)
entry_search = Entry(root)
entry_search.pack()
Button(root, text="Search", command=search_contact).pack(pady=2)

# Contact List
listbox = Listbox(root, width=60)
listbox.pack(pady=10)
listbox.bind('<<ListboxSelect>>', select_contact)

Button(root, text="View All Contacts", command=view_contacts).pack()

# Run
view_contacts()
root.mainloop()
```