```
import tkinter as tk
from tkinter import messagebox
import sqlite3

# --- Database Setup ---
def init_db():
    conn = sqlite3.connect('contacts.db')
    c = conn.cursor()
    c.execute('''
        CREATE TABLE IF NOT EXISTS contacts (
            id INTEGER PRIMARY KEY,
            name TEXT NOT NULL,
            phone TEXT NOT NULL,
            email TEXT,
            address TEXT
        )
    ''')
    conn.commit()
    conn.close()

# --- CRUD Operations ---
def add_contact():
    name = name_entry.get()
    phone = phone_entry.get()
    email = email_entry.get()
    address = address_entry.get()

    if not name or not phone:
        messagebox.showwarning("Input Error", "Name and Phone are required!")
        return

    conn = sqlite3.connect('contacts.db')
    c = conn.cursor()
    c.execute("INSERT INTO contacts (name, phone, email, address) VALUES (?, ?, ?, ?)",
              (name, phone, email, address))
    conn.commit()
    conn.close()
    clear_fields()
    show_contacts()

def show_contacts():
    contact_list.delete(0, tk.END)
    conn = sqlite3.connect('contacts.db')
    c = conn.cursor()
    c.execute("SELECT * FROM contacts")
    rows = c.fetchall()
    for row in rows:
        contact_list.insert(tk.END, f"{row[1]} | {row[2]}")
    conn.close()

def search_contact():
    keyword = search_entry.get()
    contact_list.delete(0, tk.END)
    conn = sqlite3.connect('contacts.db')
    c = conn.cursor()
    c.execute("SELECT * FROM contacts WHERE name LIKE ?", ('%' + keyword + '%',))
    rows = c.fetchall()
    for row in rows:
        contact_list.insert(tk.END, f"{row[1]} | {row[2]}")
    conn.close()

def select_contact(event):
    try:
        index = contact_list.curselection()[0]
        selected = contact_list.get(index)
        name = selected.split(" | ")[0]
        conn = sqlite3.connect('contacts.db')
        c = conn.cursor()
        c.execute("SELECT * FROM contacts WHERE name=?", (name,))
        row = c.fetchone()
        conn.close()
        if row:
            name_entry.delete(0, tk.END)
            name_entry.insert(tk.END, row[1])
            phone_entry.delete(0, tk.END)
            phone_entry.insert(tk.END, row[2])
            email_entry.delete(0, tk.END)
            email_entry.insert(tk.END, row[3])
            address_entry.delete(0, tk.END)
            address_entry.insert(tk.END, row[4])
    except IndexError:
        pass

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

    conn = sqlite3.connect('contacts.db')
    c = conn.cursor()
    c.execute("UPDATE contacts SET phone=?, email=?, address=? WHERE name=?",
              (phone, email, address, name))
    conn.commit()
    conn.close()
    clear_fields()
    show_contacts()

def delete_contact():
    name = name_entry.get()
    conn = sqlite3.connect('contacts.db')
    c = conn.cursor()
    c.execute("DELETE FROM contacts WHERE name=?", (name,))
    conn.commit()
    conn.close()
    clear_fields()
    show_contacts()

def clear_fields():
    name_entry.delete(0, tk.END)
    phone_entry.delete(0, tk.END)
    email_entry.delete(0, tk.END)
    address_entry.delete(0, tk.END)

# --- GUI Setup ---
app = tk.Tk()
app.title("Digital Contact Book")
app.geometry("500x500")

tk.Label(app, text="Name").pack()
name_entry = tk.Entry(app)
name_entry.pack()

tk.Label(app, text="Phone").pack()
phone_entry = tk.Entry(app)
phone_entry.pack()

tk.Label(app, text="Email").pack()
email_entry = tk.Entry(app)
email_entry.pack()

tk.Label(app, text="Address").pack()
address_entry = tk.Entry(app)
address_entry.pack()

tk.Button(app, text="Add Contact", command=add_contact).pack(pady=5)
tk.Button(app, text="Update Contact", command=update_contact).pack(pady=5)
tk.Button(app, text="Delete Contact", command=delete_contact).pack(pady=5)

tk.Label(app, text="Search by Name").pack()
search_entry = tk.Entry(app)
search_entry.pack()
tk.Button(app, text="Search", command=search_contact).pack(pady=5)
tk.Label(app, text="Contacts List").pack()
contact_list = tk.Listbox(app, height=10)
contact_list.pack(fill=tk.BOTH, expand=True)
contact_list.bind('<<ListboxSelect>>', select_contact)

# --- Init ---
init_db()
show_contacts()
app.mainloop()
```