

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ



BÀI TẬP LỚN LẬP TRÌNH PYTHON

NGÀNH : KỸ THUẬT MÁY TÍNH

HỆ : ĐẠI HỌC CHÍNH QUY

ĐỀ TÀI: Ứng dụng quản lý danh bạ GUI.

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ



BÀI TẬP LỚN LẬP TRÌNH
PYTHON

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI: Ứng dụng quản lý danh bạ GUI.

GIÁO VIÊN HƯỚNG DẪN : Ts. Nguyễn Văn Huy

HỌ VÀ TÊN SINH VIÊN : Ma Quốc Hiếu

LỚP : K58KTP

MSSV : K225480106089

TRƯỜNG ĐHKTCN

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN TỬ

Độc lập - Tự do - Hạnh phúc

NHIỆM VỤ THIẾT KẾ BÀI TẬP LỚN LẬP TRÌNH PYTHON

Sinh viên: Ma Quốc Hiếu MSSV: K225480106089

Lớp: K58KTP

Bộ môn: Công Nghệ Thông Tin

Giáo viên hướng dẫn: TS. Nguyễn Văn Huy

1. Tên bài tập lớn

Ứng dụng quản lý danh bạ GUI

2. Các số liệu ban đầu (nếu có)

3. Nội dung các phần thuyết minh và tính toán

- Xây ứng dụng quản lý danh bạ đơn giản: thêm, sửa, xoá liên hệ với name, phone, email, lưu vào file JSON.

4. Các sản phẩm, kết quả :

- Thêm “Nguyễn”, “0123” → xuất hiện trong Table.
- Sửa số → file JSON cập nhật.

BCN KHOA

(Ký và ghi rõ họ tên)

TRƯỞNG BỘ MÔN

(Ký và ghi rõ họ tên)

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

TRƯỜNG ĐHKTCN
KHOA ĐIỆN TỬ

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

**PHIẾU GHI ĐIỂM HƯỚNG DẪN BÀI TẬP LỚN LẬP TRÌNH
PYTHON**

Sinh viên: Ma Quốc Hiếu MSSV: K225480106089

Lớp: K58KTP.

GVHD: Ts.Nguyễn Văn Huy

Đề tài: *Ứng dụng quản lý danh bạ GUI*

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....
.....
.....

Xếp loại: Điểm :

Thái Nguyên, ngày....tháng.....năm 20....

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

Lời Cam Đoan

Tôi xin cam đoan đề tài “**Ứng dụng quản lý danh bạ GUI**” là kết quả nghiên cứu và thực hiện của riêng tôi dưới sự hướng dẫn của **Ts.Nguyễn Văn Huy**. Các số liệu, kết quả nêu trong báo cáo là trung thực và chưa từng được sử dụng để bảo vệ một học phần, khóa luận hoặc đề tài nào khác.

Trong quá trình thực hiện đề tài, tôi đã tuân thủ nghiêm túc các quy định về đạo đức học thuật và không sao chép nội dung từ bất kỳ nguồn tài liệu nào mà không trích dẫn rõ ràng. Nếu phát hiện có bất kỳ hành vi sao chép, gian lận hay vi phạm bản quyền nào, tôi xin hoàn toàn chịu trách nhiệm trước nhà trường và pháp luật.

Tôi xin chân thành cảm ơn sự giúp đỡ của quý thầy cô, bạn bè và những người đã hỗ trợ tôi trong quá trình thực hiện đề tài này.

Thái Nguyên, ngày ... tháng ... năm 2025

Người cam đoan
(Ký và ghi rõ họ tên)

Mục Lục

CHƯƠNG 1. GIỚI THIỆU MÔN HỌC VÀ ĐỀ TÀI	6
1. Giới Thiệu Môn Học	6
2. Giới Thiệu Đề Tài	6
CHƯƠNG 2. NỘI DUNG	7
1. Các bước thực hiện.....	7
1.1. Nội dung thực hiện.....	7
2. Code PYTHON	11
2.1 Class Contacts trong file contacts.py quản lý danh sách, load/save JSON.....	11
CHƯƠNG 3 TỔNG KẾT	17
1. Kết luận:	17
2. Bài học rút ra:	17
Tài Liệu Tham Khảo	18

CHƯƠNG 1. GIỚI THIỆU MÔN HỌC VÀ ĐỀ TÀI

1. Giới Thiệu Môn Học

Python là một ngôn ngữ lập trình bậc cao, thông dịch, được thiết kế với cú pháp rõ ràng, dễ đọc và dễ viết. Đây là một trong những ngôn ngữ phổ biến nhất hiện nay nhờ vào khả năng linh hoạt, dễ học và ứng dụng rộng rãi trong nhiều lĩnh vực như khoa học dữ liệu, trí tuệ nhân tạo, phát triển web, và tự động hóa.

Môn học "Lập trình Python" trang bị cho sinh viên kiến thức nền tảng về cấu trúc ngôn ngữ, biến, hàm, vòng lặp, xử lý tệp, lập trình hướng đối tượng, và làm quen với một số thư viện hỗ trợ trong Python. Qua đó, sinh viên có thể xây dựng được các ứng dụng thực tiễn, phát triển kỹ năng tư duy lập trình và giải quyết vấn đề.

2. Giới Thiệu Đề Tài

Đề tài "Ứng dụng quản lý danh bạ" là một sản phẩm phần mềm nhỏ được xây dựng bằng Python nhằm giúp người dùng lưu trữ và quản lý thông tin liên lạc một cách hiệu quả. Ứng dụng cho phép người dùng thực hiện các chức năng cơ bản như:

- Thêm liên hệ mới
- Hiện thị danh sách liên hệ
- Tìm kiếm liên hệ theo tên hoặc số điện thoại
- Chỉnh sửa hoặc xóa thông tin liên hệ
- Lưu trữ và đọc dữ liệu từ tệp (hoặc cơ sở dữ liệu nếu có)

Thông qua đề tài này, sinh viên sẽ vận dụng các kiến thức đã học để thiết kế giao diện (nếu có), xử lý dữ liệu đầu vào, thao tác với tệp hoặc cơ sở dữ liệu, và tổ chức mã nguồn hợp lý. Đây là một bài tập thực hành giúp nâng cao khả năng lập trình và hiểu rõ hơn về quy trình phát triển phần mềm.

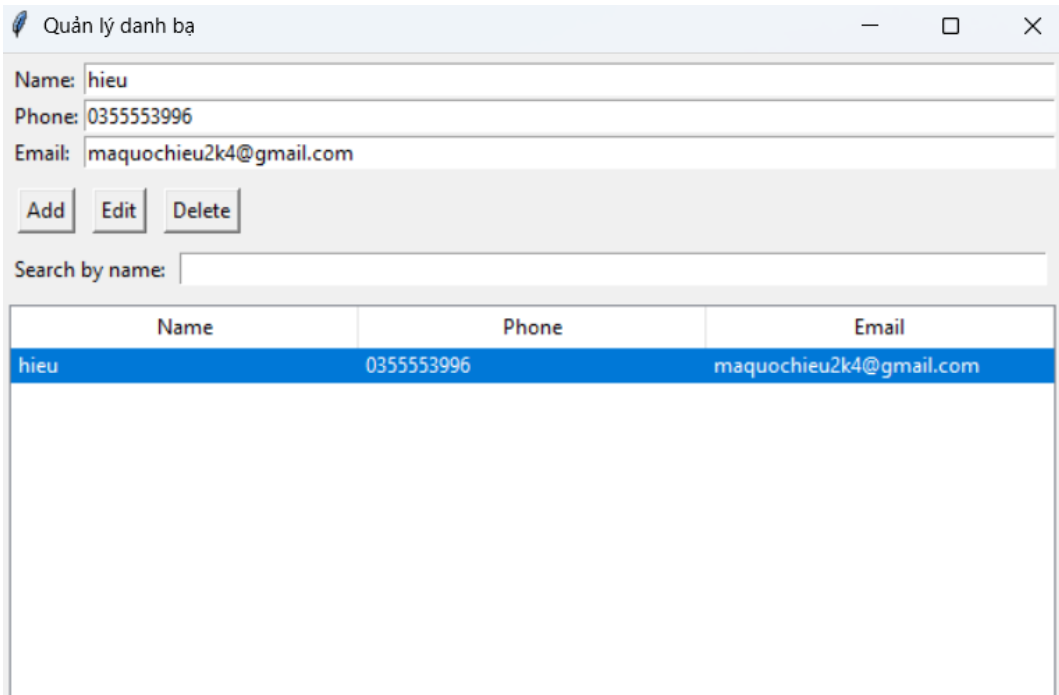
CHƯƠNG 2. NỘI DUNG

1. Các bước thực hiện.

1.1. Nội dung thực hiện

➤ **Đầu vào – đầu ra:**

- Đầu vào: Form nhập name, phone, email.
- Đầu ra: Table hiển thị danh sách, lưu file contacts.json.



Name	Phone	Email
hieu	0355553996	maquochieu2k4@gmail.com

Hình 1. Đầu vào – Đầu ra

➤ **Tính năng yêu cầu:**

- Đọc/ghi JSON với module json.
- Bắt lỗi format JSON, lưu khi thêm/sửa/xoá.
- GUI: Entry, Buttons, Treeview.
- Tìm kiếm theo tên.

- **Cách lưu thông tin**
- Nhập hết tất cả các thông tin cần lưu rồi ấn **Add**

Quản lý danh bạ

Name: hieu

Phone: 0355553996

Email: maquochieu2k4@gmail.com

Add Edit Delete

Search by name:

Name	Phone	Email
hieu	0355553996	maquochieu2k4@gmail.com
HOANG	0355553996	maquochieu2k4@gmail.com
huy	035559887	maquochieu2k4@gmail.com
quoc	038757339	abc
anh	1234	hh.com

Hình 2. Cách lưu thông tin

Hình ảnh các thông tin đã lưu

Quản lý danh bạ

Name:

Phone:

Email:

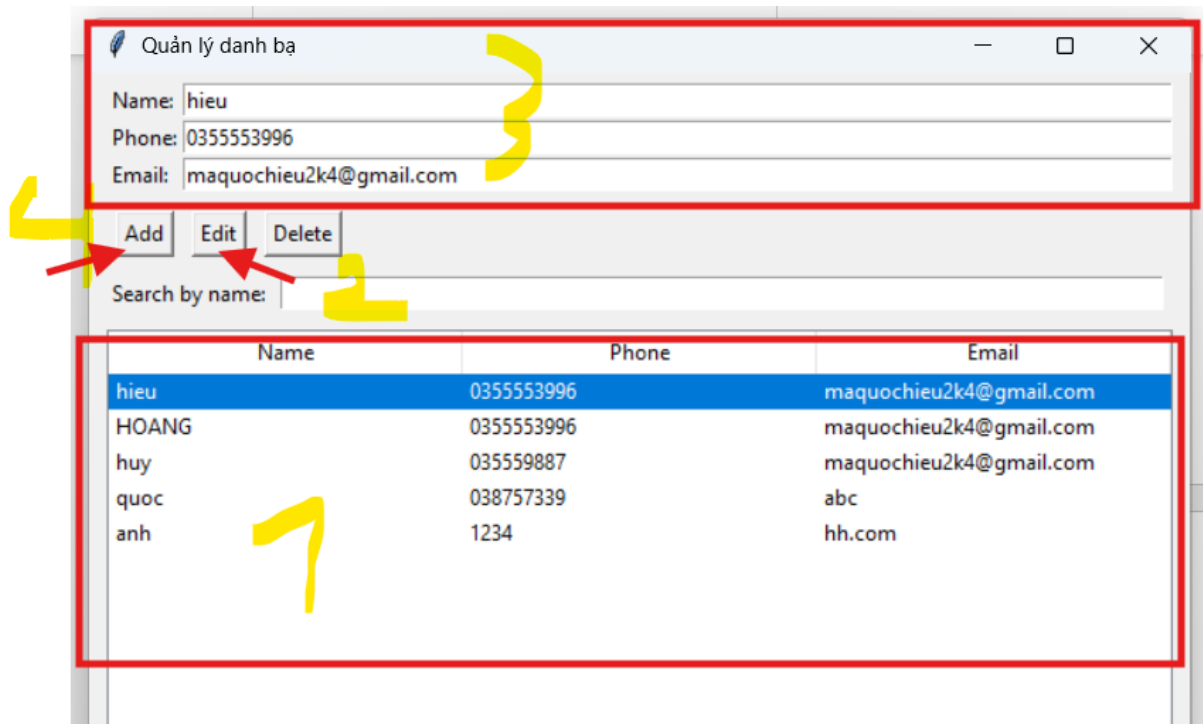
Add Edit Delete

Search by name:

Name	Phone	Email
hieu	0355553996	maquochieu2k4@gmail.com
HOANG	0355553996	maquochieu2k4@gmail.com
huy	035559887	maquochieu2k4@gmail.com
quoc	038757339	abc
anh	1234	hh.com

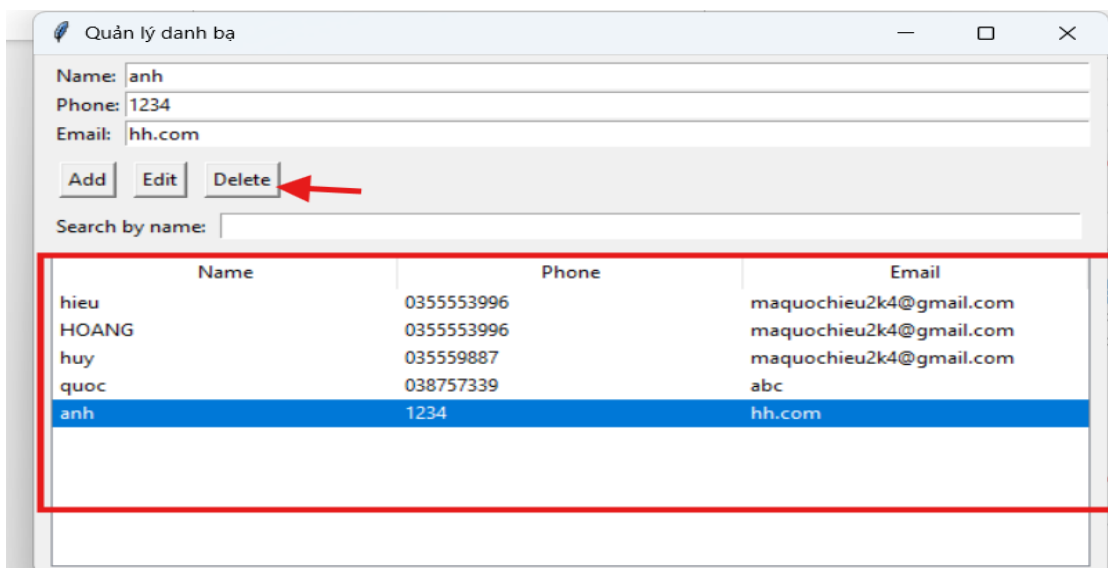
Hình 3. Danh bạ đã lưu

- Cách sửa thông tin
- Chọn thông tin cần sửa rồi bấm **Edit** , tiếp theo đó là sửa thông tin theo yêu cầu rồi **Add** lại để lưu thông tin



Hình 4. Sửa thông tin

- Cách xóa thông tin
- Chọn thông tin cần xóa rồi bấm **Delete**



Hình 5. Xóa thông tin

➤ **Kiểm tra & kết quả mẫu:**

- Thêm “Nguyen”, “0123” → xuất hiện trong Table.
- Sửa số → file JSON cập nhật.

- Thêm “Nguyen”, “0123”

Quản lý danh bạ

Name:

Phone:

Email:

Search by name:

Name	Phone	Email
hieu	0355553996	maquochieu2k4@gmail.com
HOANG	0355553996	maquochieu2k4@gmail.com
huy	035559887	maquochieu2k4@gmail.com
quoc	038757339	abc
anh	1234	hh.com
Nguyen	0123	hh.com

Hình 6. Thêm thông tin

- Sửa số → file JSON cập nhật.

Quản lý danh bạ

Name:

Phone:

Email:

Search by name:

i Sửa liên hệ thành công

Name	Phone	Email
hieu		maquochieu2k4@gmail.com
HOANG		maquochieu2k4@gmail.com
huy		maquochieu2k4@gmail.com
quoc	038757339	abc
anh	1234	hh.com
Nguyen	0123	hh.com

Hình 7. Sửa số , sửa thông tin

Quản lý danh bạ

Name:

Phone:

Email:

Add Edit Delete

Search by name:

Name	Phone	Email
hieu	0355553996	maquochieu2k4@gmail.com
HOANG	0355553996	maquochieu2k4@gmail.com
huy	035559887	maquochieu2k4@gmail.com
quoc	038757339	abc
anh	1234	hh.com
Nguyen Van Hieu	01234568	hh.commm

Hình 8. Thông tin đã được cập nhật

2. Code PYTHON

2.1 Class Contacts trong file contacts.py quản lý danh sách, load/save JSON.

- *File contacts.py*

```
import json
import os
```

```
class Contacts:
```

```
    def __init__(self, filename='contacts.json'):
        self.filename = filename
        self.contacts = []
        self.load()
```

```
    def load(self):
```

```
        if not os.path.exists(self.filename):
            self.contacts = []
            return
```

```
        try:
```

```
            with open(self.filename, 'r', encoding='utf-8') as f:
                self.contacts = json.load(f)
```

```
            if not isinstance(self.contacts, list):
                self.contacts = []
```

```
        except (json.JSONDecodeError, IOError):
            self.contacts = []
```

```

def save(self):
    try:
        with open(self.filename, 'w', encoding='utf-8') as f:
            json.dump(self.contacts, f, indent=4, ensure_ascii=False)
        return True
    except IOError:
        return False

def add(self, name, phone, email):
    self.contacts.append({'name': name, 'phone': phone, 'email': email})
    return self.save()

def update(self, index, name, phone, email):
    if 0 <= index < len(self.contacts):
        self.contacts[index] = {'name': name, 'phone': phone, 'email': email}
        return self.save()
    return False

def delete(self, index):
    if 0 <= index < len(self.contacts):
        self.contacts.pop(index)
        return self.save()
    return False

def search(self, keyword):
    keyword_lower = keyword.lower()
    return [c for c in self.contacts if keyword_lower in c['name'].lower()]

```

2.2 GUI chính trong file main.py với form nhập liệu, bảng Treeview, nút Add/Edit/Delete, tìm kiếm theo tên.

- *file main.py*

```

import tkinter as tk
from tkinter import ttk, messagebox
from contacts import Contacts

class ContactApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Quản lý danh bạ")
        self.contacts = Contacts()

```

```

self.selected_index = None

# --- Form nhập ---
frm_form = tk.Frame(root)
frm_form.pack(padx=10, pady=5, fill='x')

tk.Label(frm_form, text="Name:").grid(row=0, column=0,
sticky='w')
tk.Label(frm_form, text="Phone:").grid(row=1, column=0,
sticky='w')
tk.Label(frm_form, text="Email:").grid(row=2, column=0,
sticky='w')

self.entry_name = tk.Entry(frm_form)
self.entry_phone = tk.Entry(frm_form)
self.entry_email = tk.Entry(frm_form)

self.entry_name.grid(row=0, column=1, sticky='ew')
self.entry_phone.grid(row=1, column=1, sticky='ew')
self.entry_email.grid(row=2, column=1, sticky='ew')

frm_form.columnconfigure(1, weight=1)

# --- Buttons ---
frm_buttons = tk.Frame(root)
frm_buttons.pack(padx=10, pady=5, fill='x')

btn_add = tk.Button(frm_buttons, text="Add",
command=self.add_contact)
btn_edit = tk.Button(frm_buttons, text="Edit",
command=self.edit_contact)
btn_delete = tk.Button(frm_buttons, text="Delete",
command=self.delete_contact)

btn_add.pack(side='left', padx=5)
btn_edit.pack(side='left', padx=5)
btn_delete.pack(side='left', padx=5)

# --- Search ---
frm_search = tk.Frame(root)
frm_search.pack(padx=10, pady=5, fill='x')

```

```

tk.Label(frm_search, text="Search by name:").pack(side='left')
self.entry_search = tk.Entry(frm_search)
self.entry_search.pack(side='left', fill='x', expand=True, padx=5)
self.entry_search.bind("<KeyRelease>", self.search_contacts)

# --- Treeview ---
columns = ('name', 'phone', 'email')
self.tree = ttk.Treeview(root, columns=columns, show='headings')
self.tree.heading('name', text='Name')
self.tree.heading('phone', text='Phone')
self.tree.heading('email', text='Email')
self.tree.pack(padx=10, pady=5, fill='both', expand=True)
self.tree.bind('<<TreeviewSelect>>', self.on_tree_select)

# Load dữ liệu
self.load_contacts()

def load_contacts(self, filtered=None):
    # Xóa tất cả item hiện tại
    for item in self.tree.get_children():
        self.tree.delete(item)
    data = filtered if filtered is not None else self.contacts.contacts
    for i, c in enumerate(data):
        self.tree.insert("", 'end', iid=i, values=(c['name'], c['phone'],
c['email']))

def on_tree_select(self, event):
    selected = self.tree.selection()
    if selected:
        self.selected_index = int(selected[0])
        c = self.contacts.contacts[self.selected_index]
        self.entry_name.delete(0, tk.END)
        self.entry_name.insert(0, c['name'])
        self.entry_phone.delete(0, tk.END)
        self.entry_phone.insert(0, c['phone'])
        self.entry_email.delete(0, tk.END)
        self.entry_email.insert(0, c['email'])
    else:
        self.selected_index = None

def validate_input(self, name, phone, email):
    if not name.strip():

```

```
        messagebox.showwarning("Warning", "Name không được để  
trống!")
```

```
        return False
```

```
    # Bạn có thể thêm kiểm tra định dạng phone/email nếu muốn
```

```
    return True
```

```
def add_contact(self):
```

```
    name = self.entry_name.get()
```

```
    phone = self.entry_phone.get()
```

```
    email = self.entry_email.get()
```

```
    if not self.validate_input(name, phone, email):
```

```
        return
```

```
    if self.contacts.add(name, phone, email):
```

```
        messagebox.showinfo("Info", "Thêm liên hệ thành công")
```

```
        self.load_contacts()
```

```
        self.clear_form()
```

```
    else:
```

```
        messagebox.showerror("Error", "Lỗi khi lưu dữ liệu!")
```

```
def edit_contact(self):
```

```
    if self.selected_index is None:
```

```
        messagebox.showwarning("Warning", "Chọn liên hệ để sửa")
```

```
        return
```

```
    name = self.entry_name.get()
```

```
    phone = self.entry_phone.get()
```

```
    email = self.entry_email.get()
```

```
    if not self.validate_input(name, phone, email):
```

```
        return
```

```
    if self.contacts.update(self.selected_index, name, phone, email):
```

```
        messagebox.showinfo("Info", "Sửa liên hệ thành công")
```

```
        self.load_contacts()
```

```
        self.clear_form()
```

```
    else:
```

```
        messagebox.showerror("Error", "Lỗi khi lưu dữ liệu!")
```

```
def delete_contact(self):
```

```
    if self.selected_index is None:
```

```
        messagebox.showwarning("Warning", "Chọn liên hệ để xóa")
```

```
        return
```

```
    if messagebox.askyesno("Confirm", "Bạn chắc chắn muốn xóa?"):
```

```
        if self.contacts.delete(self.selected_index):
```

```
            messagebox.showinfo("Info", "Xóa liên hệ thành công")
```

```
            self.load_contacts()
```



```

        self.clear_form()
    else:
        messagebox.showerror("Error", "Lỗi khi lưu dữ liệu!")

def clear_form(self):
    self.entry_name.delete(0, tk.END)
    self.entry_phone.delete(0, tk.END)
    self.entry_email.delete(0, tk.END)
    self.selected_index = None
    self.tree.selection_remove(self.tree.selection())

def search_contacts(self, event):
    keyword = self.entry_search.get()
    if keyword.strip() == "":
        self.load_contacts()
    else:
        filtered = self.contacts.search(keyword)
        self.load_contacts(filtered=filtered)

if __name__ == '__main__':
    root = tk.Tk()
    app = ContactApp(root)
    root.mainloop()

```

CHƯƠNG 3 TỔNG KẾT

1. Kết luận:

Qua quá trình thực hiện đề tài "**Ứng dụng quản lý danh bạ GUI**" bằng ngôn ngữ lập trình Python, em đã vận dụng được các kiến thức lý thuyết đã học vào một bài toán thực tế. Ứng dụng đã đáp ứng được các chức năng cơ bản như thêm, sửa, xóa, tìm kiếm và lưu trữ thông tin liên hệ thông qua file `contacts.json`.

Đề tài tuy đơn giản nhưng mang lại nhiều giá trị trong việc rèn luyện kỹ năng lập trình, tư duy logic và khả năng xử lý dữ liệu trong Python. Đây là bước khởi đầu quan trọng để làm quen với các ứng dụng thực tế lớn hơn trong tương lai.

2. Bài học rút ra:

- Nắm vững kiến thức cơ bản về ngôn ngữ Python như kiểu dữ liệu, vòng lặp, điều kiện, hàm, và làm việc với file.
- Biết cách tổ chức chương trình một cách rõ ràng, dễ hiểu và dễ mở rộng.
- Rèn luyện tư duy giải quyết vấn đề thông qua việc phân tích chức năng và thiết kế luồng xử lý của ứng dụng.
- Làm quen với định dạng JSON – một định dạng phổ biến trong việc lưu trữ và truyền dữ liệu.
- Hiểu được tầm quan trọng của việc kiểm tra và xử lý lỗi trong quá trình phát triển phần mềm.

Tài Liệu Tham Khảo

1. Python Standard Library Documentation

- Nguồn: <https://docs.python.org/3/library/>
- Nội dung: Tra cứu chi tiết về các module chuẩn như json, os, tkinter.

2. Tkinter GUI Programming by Example – David Love

- Nhà xuất bản: Packt Publishing
- Nội dung: Hướng dẫn xây dựng ứng dụng GUI bằng Tkinter qua các ví dụ thực tế.

3. Python GUI Programming with Tkinter – Alan D. Moore

- Nhà xuất bản: Packt Publishing
- Nội dung: Cung cấp kiến thức nâng cao trong thiết kế giao diện người dùng với Tkinter.

4. Automate the Boring Stuff with Python – Al Sweigart

- Website: <https://automatetheboringstuff.com/>
- Chương liên quan: Làm việc với file, JSON, GUI cơ bản bằng Tkinter.

5. W3Schools Python JSON Tutorial

- Website: https://www.w3schools.com/python/python_json.asp
- Nội dung: Cách thao tác với file JSON trong Python – lưu và đọc dữ liệu.