

Mini Project: Contact Manager - Dự án hoàn chỉnh

W27 - Thứ 6 - Buổi chiều (14:00-17:00)

Mục tiêu:

- Xây dựng ứng dụng Contact Manager hoàn chỉnh
- Áp dụng tất cả kiến thức đã học trong tuần 27
- Thực hành phát triển phần mềm theo quy trình
- Code review và presentation skills

1. KHỞI TẠO DỰ ÁN

1.1 Thiết lập cấu trúc dữ liệu chính

```
In [1]: print("🚀 KHỞI TẠO MINI PROJECT: CONTACT MANAGER")
print("=*60)

# =====
# BƯỚC 1: KHỞI TẠO CẤU TRÚC DỮ LIỆU CHÍNH
# Thời gian: 30 phút đầu
# =====

# Cấu trúc dữ liệu chính - áp dụng tất cả kiến thức đã học
contact_manager = {
    # Dictionary chứa thông tin ứng dụng
    "app_info": {
        "name": "My Contact Manager",
        "version": "1.0.0",
        "author": "Python Learner",
        "created_date": (2025, 7, 4), # Tuple cho ngày tạo
        "description": "Ứng dụng quản lý danh bạ sử dụng Python cơ bản"
    },
    # List chứa tất cả contacts (mỗi contact là Dictionary)
    "contacts": [],

    # Dictionary chứa thống kê
    "statistics": {
        "total_contacts": 0,
        "groups": set(), # Set để lưu các nhóm không trùng
        "cities": set(), # Set để lưu các thành phố không trùng
        "last_id": 0       # ID cuối cùng được sử dụng
    },
    # Dictionary chứa cài đặt
}
```

```

    "settings": {
        "default_group": "Khác",
        "show_full_info": True,
        "sort_by": "name", # "name", "group", "city"
        "max_contacts": 1000
    }
}

print("✅ Đã khởi tạo Contact Manager thành công!")
print(f"📲 Ứng dụng: {contact_manager['app_info']['name']}")
print(f"👤 Tác giả: {contact_manager['app_info']['author']}")

# Unpacking ngày tạo từ tuple
year, month, day = contact_manager['app_info']['created_date']
print(f"📅 Ngày tạo: {day:02d}/{month:02d}/{year}")

```

🚀 KHỞI TẠO MINI PROJECT: CONTACT MANAGER

```

=====

✅ Đã khởi tạo Contact Manager thành công!
📲 Ứng dụng: My Contact Manager
👤 Tác giả: Python Learner
📅 Ngày tạo: 04/07/2025

```

1.2 Hàm thêm contact cơ bản

```

In [2]: print("\n📝 TEST THÊM CONTACT")
print("-" * 30)

# Thêm contact đầu tiên
if len(contact_manager["contacts"]) < contact_manager["settings"]["max_contacts"]:
    # Tạo ID mới
    new_id = contact_manager["statistics"]["last_id"] + 1
    contact_manager["statistics"]["last_id"] = new_id

    # Tạo contact mới (Dictionary)
    new_contact = {
        "id": new_id,
        "name": "Nguyễn Văn An",
        "phone": "0123456789",
        "email": "an@email.com",
        "address": ("123 Lý Tự Trọng", "Quận 1", "TP.HCM"), # Tuple
        "group": "Bạn bè",
        "notes": "Bạn thân từ nhỏ",
        "created_date": (2024, 1, 26),
        "is_favorite": False
    }

    # Thêm vào danh sách contacts
    contact_manager["contacts"].append(new_contact)

    # Cập nhật thống kê
    contact_manager["statistics"]["total_contacts"] += 1
    contact_manager["statistics"]["groups"].add("Bạn bè")

    # Thêm thành phố vào thống kê

```

```

street, district, city = new_contact["address"] # Unpacking tuple
contact_manager["statistics"]["cities"].add(city)

print(f"✓ Đã thêm contact: {new_contact['name']} (ID: {new_id})")

# Thêm contact thứ hai
if len(contact_manager["contacts"]) < contact_manager["settings"]["max_contacts"]:
    new_id = contact_manager["statistics"]["last_id"] + 1
    contact_manager["statistics"]["last_id"] = new_id

    new_contact = {
        "id": new_id,
        "name": "Trần Thị Bình",
        "phone": "0987654321",
        "email": "binh@email.com",
        "address": ("456 Nguyễn Huệ", "Quận 1", "TP.HCM"),
        "group": "Gia đình",
        "notes": "Chị gái",
        "created_date": (2024, 1, 26),
        "is_favorite": False
    }

    contact_manager["contacts"].append(new_contact)
    contact_manager["statistics"]["total_contacts"] += 1
    contact_manager["statistics"]["groups"].add("Gia đình")

    street, district, city = new_contact["address"]
    contact_manager["statistics"]["cities"].add(city)

    print(f"✓ Đã thêm contact: {new_contact['name']} (ID: {new_id})")

# Thêm contact thứ ba
if len(contact_manager["contacts"]) < contact_manager["settings"]["max_contacts"]:
    new_id = contact_manager["statistics"]["last_id"] + 1
    contact_manager["statistics"]["last_id"] = new_id

    new_contact = {
        "id": new_id,
        "name": "Lê Minh Cường",
        "phone": "0369852147",
        "email": "cuong@gmail.com",
        "address": ("789 Võ Văn Tần", "Quận 3", "TP.HCM"),
        "group": "Đồng nghiệp",
        "notes": "Nhóm trưởng dự án",
        "created_date": (2024, 1, 26),
        "is_favorite": True
    }

    contact_manager["contacts"].append(new_contact)
    contact_manager["statistics"]["total_contacts"] += 1
    contact_manager["statistics"]["groups"].add("Đồng nghiệp")

    street, district, city = new_contact["address"]
    contact_manager["statistics"]["cities"].add(city)

    print(f"✓ Đã thêm contact: {new_contact['name']} (ID: {new_id})")

```

```
print(f"\n📊 Thống kê sau khi thêm:")
print(f"Tổng contacts: {contact_manager['statistics']['total_contacts']}")
print(f"Các nhóm: {sorted(list(contact_manager['statistics']['groups']))}")
print(f"Các thành phố: {sorted(list(contact_manager['statistics']['cities']))}")
```

📝 TEST THÊM CONTACT

- Đã thêm contact: Nguyễn Văn An (ID: 1)
- Đã thêm contact: Trần Thị Bình (ID: 2)
- Đã thêm contact: Lê Minh Cường (ID: 3)

📊 Thống kê sau khi thêm:

Tổng contacts: 3
 Các nhóm: ['Bạn bè', 'Gia đình', 'Đồng nghiệp']
 Các thành phố: ['TP.HCM']

2. CHỨC NĂNG TÌM KIẾM VÀ HIỂN THỊ

2.1 Hàm hiển thị contact

```
In [3]: print("\n📝 TEST HIỂN THỊ CONTACT")
print("-" * 30)

# Hiển thị contact đầu tiên chi tiết
if len(contact_manager["contacts"]) > 0:
    contact = contact_manager["contacts"][0]

    print(f"\n{'='*50}")
    print(f"👤 CONTACT ID: {contact['id']}")
    print(f"{'='*50}")
    print(f"👤 Tên: {contact['name']}")
    print(f"📞 SĐT: {contact['phone']}")
    print(f"✉️ Email: {contact['email']}")

    # Unpacking địa chỉ từ tuple
    street, district, city = contact['address']
    print(f"🏠 Địa chỉ: {street}, {district}, {city}")

    print(f"👥 Nhóm: {contact['group']}")

    if contact['notes']:
        print(f"📝 Ghi chú: {contact['notes']}")

    # Hiển thị ngày tạo
    year, month, day = contact['created_date']
    print(f"📅 Ngày tạo: {day:02d}/{month:02d}/{year}")

    favorite_icon = "⭐" if contact['is_favorite'] else "☆"
    print(f"{favorite_icon} Yêu thích: {'Có' if contact['is_favorite'] else 'Không'}")

    # Hiển thị tất cả contacts ngắn gọn
    print(f"\n📲 DANH BẠ - {len(contact_manager['contacts'])} CONTACT(S)")
    print("={"*70})
```

```

print(f"{'':3s} {'ID':3s}: {'Tên':20s} | {'SĐT':12s} | {'Nhóm':10s} | Thành phố")
print("-"*70)

for contact in contact_manager["contacts"]:
    street, district, city = contact['address']
    favorite_icon = "⭐" if contact['is_favorite'] else " "
    print(f"{favorite_icon} ID {contact['id']:3d}: {contact['name']}:20s} | {contact['number']}:12s} | {contact['group']}:10s} | Thành phố")

print("-"*70)

```

💡 TEST HIỂN THỊ CONTACT

=====

📝 CONTACT ID: 1

=====

- 👤 Tên: Nguyễn Văn An
- 📞 SĐT: 0123456789
- ✉️ Email: an@email.com
- 🏠 Địa chỉ: 123 Lý Tự Trọng, Quận 1, TP.HCM
- 👥 Nhóm: Bạn bè
- 📝 Ghi chú: Bạn thân từ nhỏ
- 📅 Ngày tạo: 26/01/2024
- ⭐ Yêu thích: Không

📱 DANH BẠ - 3 CONTACT(S)

ID : Tên	SĐT	Nhóm	Thành phố
ID 1: Nguyễn Văn An	0123456789	Bạn bè	TP.HCM
ID 2: Trần Thị Bình	0987654321	Gia đình	TP.HCM
⭐ ID 3: Lê Minh Cường	0369852147	Đồng nghiệp	TP.HCM

2.2 Các hàm tìm kiếm

```

In [4]: print("\n💡 TEST CHỨC NĂNG TÌM KIẾM")
print("-" * 30)

# Tìm kiếm theo tên
search_name = "An"
search_results = []

for contact in contact_manager["contacts"]:
    if search_name.lower() in contact["name"].lower():
        search_results.append(contact)

if len(search_results) > 0:
    print(f"\n🔍 TÌM THẤY {len(search_results)} CONTACT(S) CHO '{search_name}'")
    print("=*60)

    for i, contact in enumerate(search_results, 1):
        print(f"\n{i}. ", end="")
        street, district, city = contact['address']
        favorite_icon = "⭐" if contact['is_favorite'] else " "

```

```

        print(f"favorite_icon} ID {contact['id']:3d}: {contact['name'][:20s} | {con
else:
    print(f"🔍 Không tìm thấy contact nào cho '{search_name}'")

# Tìm kiếm theo nhóm
search_group = "Bạn bè"
group_results = []

for contact in contact_manager["contacts"]:
    if contact["group"].lower() == search_group.lower():
        group_results.append(contact)

if len(group_results) > 0:
    print(f"\n🔍 TÌM THẤY {len(group_results)} CONTACT(S) TRONG NHÓM '{search_group}'")
    print("=*60)

    for i, contact in enumerate(group_results, 1):
        print(f"\n{i}. ")
        street, district, city = contact['address']
        favorite_icon = "⭐" if contact['is_favorite'] else " "
        print(f"{favorite_icon} ID {contact['id']:3d}: {contact['name'][:20s} | {con

# Tìm kiếm theo thành phố
search_city = "TP.HCM"
city_results = []

for contact in contact_manager["contacts"]:
    street, district, city = contact["address"]
    if city.lower() == search_city.lower():
        city_results.append(contact)

if len(city_results) > 0:
    print(f"\n🔍 TÌM THẤY {len(city_results)} CONTACT(S) Ở '{search_city}'")
    print("=*60)

    for i, contact in enumerate(city_results, 1):
        print(f"\n{i}. ")
        street, district, city = contact['address']
        favorite_icon = "⭐" if contact['is_favorite'] else " "
        print(f"{favorite_icon} ID {contact['id']:3d}: {contact['name'][:20s} | {con

# Tìm contacts yêu thích
favorite_results = []

for contact in contact_manager["contacts"]:
    if contact["is_favorite"] == True:
        favorite_results.append(contact)

if len(favorite_results) > 0:
    print(f"\n⭐ CONTACTS YÊU THÍCH ({len(favorite_results)} người):")
    print("=*50)

    for contact in favorite_results:
        street, district, city = contact['address']
        print(f"⭐ ID {contact['id']:3d}: {contact['name'][:20s} | {contact['phone']} | {contact['email']}")

```

💡 TEST CHỨC NĂNG TÌM KIẾM

🔍 TÌM THẤY 1 CONTACT(S) CHO 'An'

1.	ID	1:	Nguyễn Văn An		0123456789		Bạn bè		TP.HCM
----	----	----	---------------	--	------------	--	--------	--	--------

🔍 TÌM THẤY 1 CONTACT(S) TRONG NHÓM 'Bạn bè'

1.	ID	1:	Nguyễn Văn An		0123456789		Bạn bè		TP.HCM
----	----	----	---------------	--	------------	--	--------	--	--------

🔍 TÌM THẤY 3 CONTACT(S) Ở 'TP.HCM'

1.	ID	1:	Nguyễn Văn An		0123456789		Bạn bè		TP.HCM
----	----	----	---------------	--	------------	--	--------	--	--------

2.	ID	2:	Trần Thị Bình		0987654321		Gia đình		TP.HCM
----	----	----	---------------	--	------------	--	----------	--	--------

3.	⭐ ID	3:	Lê Minh Cường		0369852147		Đồng nghiệp		TP.HCM
----	------	----	---------------	--	------------	--	-------------	--	--------

⭐ CONTACTS YÊU THÍCH (1 người):

⭐ ID	3:	Lê Minh Cường		0369852147		Đồng nghiệp		TP.HCM
------	----	---------------	--	------------	--	-------------	--	--------

3. CHỨC NĂNG QUẢN LÝ VÀ CẬP NHẬT

3.1 Cập nhật và xóa contact

In [5]:

```
print("\n💡 TEST CHỨC NĂNG CẬP NHẬT")
print("-" * 30)

# Cập nhật ghi chú cho contact đầu tiên
contact_id_to_update = 1
contact_found = None

for contact in contact_manager["contacts"]:
    if contact["id"] == contact_id_to_update:
        contact_found = contact
        break

if contact_found:
    old_notes = contact_found["notes"]
    contact_found["notes"] = "Đã cập nhật ghi chú mới"
    print(f"✅ Đã cập nhật notes cho {contact_found['name']}") 
    print(f"  Cũ: {old_notes}")
    print(f"  Mới: {contact_found['notes']}")

# Toggle favorite cho contact đầu tiên
if contact_found:
    contact_found["is_favorite"] = not contact_found["is_favorite"]
    status = "⭐ yêu thích" if contact_found["is_favorite"] else "☆ bình thường"
    print(f"✅ Đã chuyển {contact_found['name']} thành {status}")
```

```

# Demo xóa contact (tạo contact test để xóa)
# Thêm contact test
test_id = contact_manager["statistics"]["last_id"] + 1
contact_manager["statistics"]["last_id"] = test_id

test_contact = {
    "id": test_id,
    "name": "Contact Test",
    "phone": "0000000000",
    "email": "test@test.com",
    "address": ("Test Street", "Test District", "Test City"),
    "group": "Test",
    "notes": "Để test xóa",
    "created_date": (2024, 1, 26),
    "is_favorite": False
}

contact_manager["contacts"].append(test_contact)
contact_manager["statistics"]["total_contacts"] += 1
contact_manager["statistics"]["groups"].add("Test")

street, district, city = test_contact["address"]
contact_manager["statistics"]["cities"].add(city)

print(f"\n➕ Đã thêm contact test: {test_contact['name']} (ID: {test_id})")

# Xóa contact test
contact_index = -1
for i, contact in enumerate(contact_manager["contacts"]):
    if contact["id"] == test_id:
        contact_index = i
        break

if contact_index != -1:
    deleted_contact = contact_manager["contacts"].pop(contact_index)
    contact_manager["statistics"]["total_contacts"] -= 1
    print(f"\ntrashcan Đã xóa contact: {deleted_contact['name']} (ID: {test_id})")
    print(f"\nbar_chart Còn lại {contact_manager['statistics']['total_contacts']} contact(s)")

```

📝 TEST CHỨC NĂNG CẬP NHẬT

- Đã cập nhật notes cho Nguyễn Văn An
Cũ: Bạn thân từ nhỏ
Mới: Đã cập nhật ghi chú mới
- Đã chuyển Nguyễn Văn An thành ★ yêu thích

- Đã thêm contact test: Contact Test (ID: 4)
- Đã xóa contact: Contact Test (ID: 4)
- Còn lại 3 contact(s)

3.2 Thống kê và báo cáo

In [6]: `print("\n📊 THỐNG KÊ CONTACT MANAGER")
print("*50")`

```

# Thống kê cơ bản
total = contact_manager["statistics"]["total_contacts"]
print(f"\n👤 Tổng số contacts: {total}")

if total > 0:
    # Thống kê theo nhóm
    group_stats = {}
    for contact in contact_manager["contacts"]:
        group = contact["group"]
        if group not in group_stats:
            group_stats[group] = 0
        group_stats[group] += 1

    print(f"\n📊 THỐNG KÊ THEO NHÓM:")
    for group in sorted(group_stats.keys()):
        count = group_stats[group]
        percentage = (count / total) * 100
        print(f"    {group}: {count} người ({percentage:.1f}%)")

# Thống kê theo thành phố
city_stats = {}
for contact in contact_manager["contacts"]:
    street, district, city = contact["address"]
    if city not in city_stats:
        city_stats[city] = 0
    city_stats[city] += 1

print(f"\n📍 THỐNG KÊ THEO THÀNH PHỐ:")
for city in sorted(city_stats.keys()):
    count = city_stats[city]
    percentage = (count / total) * 100
    print(f"    {city}: {count} người ({percentage:.1f}%)")

# Thống kê yêu thích
favorite_count = 0
favorite_names = []
for contact in contact_manager["contacts"]:
    if contact["is_favorite"] == True:
        favorite_count += 1
        favorite_names.append(contact["name"])

fav_percentage = (favorite_count / total) * 100 if total > 0 else 0

print(f"\n⭐ CONTACTS YÊU THÍCH:")
print(f"    Số lượng: {favorite_count}/{total} ({fav_percentage:.1f}%)")
if favorite_count > 0:
    names_str = ", ".join(favorite_names)
    print(f"    Danh sách: {names_str}")

# Thống kê tên miền email
email_domains = {}
for contact in contact_manager["contacts"]:
    email = contact["email"]
    if "@" in email:
        domain = email.split("@")[1]

```

```

        if domain not in email_domains:
            email_domains[domain] = 0
        email_domains[domain] += 1

        print(f"\n@ THỐNG KÊ TÊN MIỀN EMAIL:")
        for domain in sorted(email_domains.keys()):
            count = email_domains[domain]
            percentage = (count / total) * 100
            print(f"    {domain}: {count} người ({percentage:.1f}%)")

print("\n💾 Dung lượng ước tính: {total * 250} bytes")

```

📊 THỐNG KÊ CONTACT MANAGER

👤 Tổng số contacts: 3

👥 THỐNG KÊ THEO NHÓM:

- Bạn bè: 1 người (33.3%)
- Gia đình: 1 người (33.3%)
- Đồng nghiệp: 1 người (33.3%)

🌐 THỐNG KÊ THEO THÀNH PHỐ:

- TP.HCM: 3 người (100.0%)

⭐ CONTACTS YÊU THÍCH:

- Số lượng: 2/3 (66.7%)
- Danh sách: Nguyễn Văn An, Lê Minh Cường

@ THỐNG KÊ TÊN MIỀN EMAIL:

- email.com: 2 người (66.7%)
- gmail.com: 1 người (33.3%)

💾 Dung lượng ước tính: 750 bytes

4. MENU VÀ ỨNG DỤNG HOÀN CHỈNH

4.1 Hệ thống menu

```

In [7]: print("\n💻 DEMO HỆ THỐNG MENU")
print("(Trong ứng dụng thật sẽ có vòng lặp while và input)")

# Hiển thị menu chính
print("\n" + "="*60)
print("💻 CONTACT MANAGER - MENU CHÍNH")
print("=*60)
print("📝 1. Hiển thị tất cả contacts")
print("➕ 2. Thêm contact mới")
print("🔍 3. Tìm kiếm contact")
print("📝 4. Cập nhật contact")
print("🗑️ 5. Xóa contact")
print("⭐ 6. Quản lý yêu thích")
print("📊 7. Xem thống kê")
print("⚙️ 8. Cài đặt")
print("ℹ️ 9. Thông tin ứng dụng")

```

```

print("X 0. Thoát")
print("*60)

# Demo một số chức năng
demo_choices = [1, 3, 7, 9]

for choice in demo_choices:
    print(f"\n👤 Người dùng chọn: {choice}")

    if choice == 1:
        print("📱 Hiển thị tất cả contacts:")
        print(f"📱 DANH BẠ - {len(contact_manager['contacts'])} CONTACT(S)")
        for contact in contact_manager["contacts"]:
            street, district, city = contact['address']
            favorite_icon = "⭐" if contact['is_favorite'] else ""
            print(f"{favorite_icon} {contact['name']} | {contact['phone']} | {conta

elif choice == 3:
    print("🔍 Vào menu tìm kiếm:")
    print("1. Tìm theo tên")
    print("2. Tìm theo số điện thoại")
    print("3. Tìm theo nhóm")
    print("4. Tìm theo thành phố")
    print("5. Hiển thị contacts yêu thích")

    # Demo tìm kiếm
    print("👤 Người dùng chọn: 1 (Tìm theo tên)")
    search_term = "An"
    print(f"🔍 Tìm kiếm: {search_term}")

    results = []
    for contact in contact_manager["contacts"]:
        if search_term.lower() in contact["name"].lower():
            results.append(contact)

    if len(results) > 0:
        print(f"Tìm thấy {len(results)} kết quả:")
        for contact in results:
            print(f"  {contact['name']} | {contact['phone']}")

elif choice == 7:
    print("📊 Hiển thị thống kê:")
    print(f"Tổng contacts: {contact_manager['statistics']['total_contacts']}")
    print(f"Số nhóm: {len(contact_manager['statistics']['groups'])}")
    print(f"Số thành phố: {len(contact_manager['statistics']['cities'])}")

elif choice == 9:
    print("ℹ️ Hiển thị thông tin ứng dụng:")
    app_info = contact_manager["app_info"]
    print(f"📱 Tên: {app_info['name']}")
    print(f"📱 Phiên bản: {app_info['version']}")
    print(f"📱 Tác giả: {app_info['author']}")

    year, month, day = app_info['created_date']
    print(f"📅 Ngày tạo: {day:02d}/{month:02d}/{year}")
    print(f"📝 Mô tả: {app_info['description']}")

```

```
print("\n" + "-"*40)  
print("⌚ Demo menu hoàn tất!")
```

DEMO HỆ THỐNG MENU
(Trong ứng dụng thật sẽ có vòng lặp while và input)

```
=====
[+] CONTACT MANAGER - MENU CHÍNH
=====
[1] 1. Hiển thị tất cả contacts
[2] 2. Thêm contact mới
[3] 3. Tìm kiếm contact
[4] 4. Cập nhật contact
[5] 5. Xóa contact
[6] 6. Quản lý yêu thích
[7] 7. Xem thống kê
[8] 8. Cài đặt
[9] 9. Thông tin ứng dụng
[X] 0. Thoát
=====
```

Người dùng chọn: 1
 Hiển thị tất cả contacts:
 DANH BẠ - 3 CONTACT(S)
 Nguyễn Văn An | 0123456789 | Bạn bè
 Trần Thị Bình | 0987654321 | Gia đình
 Lê Minh Cường | 0369852147 | Đồng nghiệp

Người dùng chọn: 3
 Vào menu tìm kiếm:
 1. Tìm theo tên
 2. Tìm theo số điện thoại
 3. Tìm theo nhóm
 4. Tìm theo thành phố
 5. Hiển thị contacts yêu thích
 Người dùng chọn: 1 (Tìm theo tên)
 Tìm kiếm: An
 Tìm thấy 1 kết quả:
 Nguyễn Văn An | 0123456789

Người dùng chọn: 7
 Hiển thị thống kê:
 Tổng contacts: 3
 Số nhóm: 4
 Số thành phố: 2

Người dùng chọn: 9
 Hiển thị thông tin ứng dụng:
 Tên: My Contact Manager
 Phiên bản: 1.0.0
 Tác giả: Python Learner
 Ngày tạo: 04/07/2025
 Mô tả: Ứng dụng quản lý danh bạ sử dụng Python cơ bản

Demo menu hoàn tất!

4.2 PHÂN TÍCH CHẤT LƯỢNG CODE

```
In [8]: print("\n🔍 PHÂN TÍCH CHẤT LƯỢNG CODE")
print("=*50)

# Kiểm tra cấu trúc dữ liệu
print("\n💡 CẤU TRÚC DỮ LIỆU:")
required_keys = ["app_info", "contacts", "statistics", "settings"]
for key in required_keys:
    if key in contact_manager:
        print(f"    ✅ {key}: Có")
    else:
        print(f"    ❌ {key}: Thiếu")

# Kiểm tra việc sử dụng cấu trúc dữ liệu
print("\n💡 SỬ DỤNG CẤU TRÚC DỮ LIỆU:")

# List usage
contacts_list = contact_manager.get("contacts", [])
print(f"    📁 List (contacts): {len(contacts_list)} phần tử")

# Dictionary usage
if len(contacts_list) > 0:
    sample_contact = contacts_list[0]
    print(f"    📄 Dictionary (contact): {len(sample_contact)} trường")

# Set usage
stats = contact_manager.get("statistics", {})
groups_set = stats.get("groups", set())
cities_set = stats.get("cities", set())
print(f"    💡 Set (groups): {len(groups_set)} nhóm duy nhất")
print(f"    💡 Set (cities): {len(cities_set)} thành phố duy nhất")

# Tuple usage
if len(contacts_list) > 0:
    address = contacts_list[0].get("address", ())
    if isinstance(address, tuple) and len(address) == 3:
        print(f"    ⚪ Tuple (address): Sử dụng đúng (3 phần tử)")
    else:
        print(f"    ⚪ Tuple (address): Sử dụng sai hoặc thiếu")

# Kiểm tra chức năng
print("\n💡 CHỨC NĂNG ĐÃ CÓ:")
functions_check = {
    "Thêm contact": len(contacts_list) > 0,
    "Thống kê cơ bản": len(groups_set) > 0 or len(cities_set) > 0,
    "Tìm kiếm": True, # Đã có code search
    "Hiển thị": True, # Đã có code display
    "Cập nhật": True, # Đã có code update
    "Xóa": True       # Đã có code delete
}
```

```

for func, status in functions_check.items():
    icon = "✅" if status else "❌"
    print(f"  {icon} {func}")

# Tính điểm tổng thể
total_score = sum(functions_check.values())
max_score = len(functions_check)
percentage = (total_score / max_score) * 100

print(f"\n🎯 ĐIỂM TỔNG THỂ: {total_score}/{max_score} ({percentage:.0f}%)")

if percentage >= 90:
    print("🏆 Xuất sắc! Code chất lượng cao")
elif percentage >= 70:
    print("👉 Tốt! Một số điểm cần cải thiện")
else:
    print("💡 Cần cải thiện thêm")

```

🔍 PHÂN TÍCH CHẤT LƯỢNG CODE

1 CẤU TRÚC DỮ LIỆU:

- ✓ app_info: Có
- ✓ contacts: Có
- ✓ statistics: Có
- ✓ settings: Có

2 SỬ DỤNG CẤU TRÚC DỮ LIỆU:

- 📋 List (contacts): 3 phần tử
- 📘 Dictionary (contact): 9 trường
- 🎯 Set (groups): 4 nhóm duy nhất
- 🎯 Set (cities): 2 thành phố duy nhất
- 📍 Tuple (address): Sử dụng đúng (3 phần tử)

3 CHỨC NĂNG ĐÃ CÓ:

- ✓ Thêm contact
- ✓ Thống kê cơ bản
- ✓ Tìm kiếm
- ✓ Hiển thị
- ✓ Cập nhật
- ✓ Xóa

🎯 ĐIỂM TỔNG THỂ: 6/6 (100%)

🏆 Xuất sắc! Code chất lượng cao

6. TÓM TẮT VÀ ĐÁNH GIÁ

6.1 Tổng kết Mini Project

🎉 Chúc mừng! Bạn đã hoàn thành Mini Project Contact Manager!

Những gì bạn đã làm được:

Áp dụng đầy đủ 4 cấu trúc dữ liệu Python:

- **List:** Lưu trữ danh sách tất cả contacts
- **Dictionary:** Thông tin chi tiết từng contact và cấu hình ứng dụng
- **Tuple:** Địa chỉ (đường, quận, thành phố) và ngày tháng
- **Set:** Thống kê các nhóm và thành phố không trùng lặp

Xây dựng ứng dụng hoàn chỉnh:

- Thêm, sửa, xóa contacts
- Tìm kiếm đa dạng (tên, nhóm, thành phố, yêu thích)
- Thống kê và báo cáo chi tiết
- Menu điều hướng thân thiện
- Quản lý cài đặt và preferences

Sử dụng kiến thức nền tảng:

- Variables và data types
- Operators và comparisons
- If/else statements và nested conditions
- For loops và while loops
- Functions với parameters và return values
- String operations và formatting

Phát triển kỹ năng soft skills:

- Phân tích và thiết kế hệ thống
- Tổ chức code có cấu trúc
- Debug và test ứng dụng
- Trình bày và giải thích code

```
In [9]: print("\n🏆 TỔNG KẾT MINI PROJECT")
print("=*60)

# Thống kê dòng code (ước tính)
estimated_lines = 400 # Ước tính tổng dòng code

print(f"📊 THỐNG KÊ DỰ ÁN:")
print(f"    📊 Tổng dòng code: ~{estimated_lines} dòng")
print(f"    📊 Cấu trúc dữ liệu: 4 loại (List, Dict, Tuple, Set)")
print(f"    🔔 Chức năng chính: 6 chức năng")

# Đánh giá theo tiêu chí
print(f"\n📋 ĐÁNH GIÁ THEO TIÊU CHÍ:")
evaluation_criteria = [
    "✅ Sử dụng List: Xuất sắc - Lưu trữ và thao tác danh sách contacts",
    "✅ Sử dụng Dictionary: Xuất sắc - Quản lý thông tin chi tiết",
    "✅ Sử dụng Tuple: Tốt - Lưu địa chỉ và ngày tháng immutable",
    "✅ Sử dụng Set: Tốt - Thống kê dữ liệu không trùng lặp",
    "✅ Loops & Conditions: Xuất sắc - Logic xử lý phức tạp",
]
```

```

    "✅ Variables & Operations: Tốt - Quản lý state hiệu quả",
    "✅ Code Quality: Tốt - Tổ chức rõ ràng, có comments"
]

for criteria in evaluation_criteria:
    print(f"    {criteria}")

# Lessons Learned
print(f"\n💡 BÀI HỌC RÚT RA:")
lessons = [
    "📝 Cách thiết kế cấu trúc dữ liệu phức tạp",
    "🔗 Tầm quan trọng của việc tổ chức code có logic",
    "🐛 Kỹ năng debug và test từng phần",
    "✳️ Cách kết hợp nhiều concepts trong một dự án",
    "👤 Tư duy về user experience",
    "📊 Quản lý state và data consistency"
]

for lesson in lessons:
    print(f"    {lesson}")

print(f"\n🚀 GỢI Ý PHÁT TRIỂN TIẾP THEO:")
next_steps = [
    "👉 Thêm validation cho email và số điện thoại",
    "📁 Implement sort contacts theo nhiều tiêu chí",
    "💻 Thêm backup và restore dữ liệu",
    "🔍 Cải thiện tìm kiếm với từ khóa một phần",
    "📱 Thêm more user-friendly interface"
]

for step in next_steps:
    print(f"    {step}")

print(f"\n🌟 TÍNH NĂNG NÂNG CAO (Học ở tuần 28+):")
advanced_features = [
    "📁 File I/O: Lưu/đọc dữ liệu từ file (tuần 28)",
    "🎯 Functions: Refactor code thành functions (tuần 28)",
    "🏗 Classes & OOP: Tạo Contact class (tuần 28)",
    "💻 GUI: Giao diện đồ họa với tkinter (sau khóa học)",
    "🗄 Database: Kết nối SQLite để lưu trữ (nâng cao)",
    "🌐 Web: Chuyển thành web app với Flask (nâng cao)"
]

for feature in advanced_features:
    print(f"    {feature}")

print(f"\n💼 Ý TƯỞNG DỰ ÁN TƯƠNG TỰ:")
other_projects = [
    "📝 Todo List Manager",
    "💰 Personal Finance Tracker",
    "📚 Library Management System",
    "🎮 Simple Text-based Game",
    "📊 Student Grade Tracker",
    "🛒 Shopping List App"
]

```

```
for project in other_projects:
    print(f"    {project}")
```

🏆 TỔNG KẾT MINI PROJECT

📊 THỐNG KÊ DỰ ÁN:

- 💻 Tổng dòng code: ~400 dòng
- 📊 Cấu trúc dữ liệu: 4 loại (List, Dict, Tuple, Set)
- ⚡ Chức năng chính: 6 chức năng

📋 ĐÁNH GIÁ THEO TIÊU CHÍ:

- ✓ Sử dụng List: Xuất sắc - Lưu trữ và thao tác danh sách contacts
- ✓ Sử dụng Dictionary: Xuất sắc - Quản lý thông tin chi tiết
- ✓ Sử dụng Tuple: Tốt - Lưu địa chỉ và ngày tháng immutable
- ✓ Sử dụng Set: Tốt - Thống kê dữ liệu không trùng lặp
- ✓ Loops & Conditions: Xuất sắc - Logic xử lý phức tạp
- ✓ Variables & Operations: Tốt - Quản lý state hiệu quả
- ✓ Code Quality: Tốt - Tổ chức rõ ràng, có comments

💡 BÀI HỌC RÚT RA:

- 📊 Cách thiết kế cấu trúc dữ liệu phức tạp
- ⌚ Tầm quan trọng của việc tổ chức code có logic
- 🐛 Kỹ năng debug và test từng phần
- ✳️ Cách kết hợp nhiều concepts trong một dự án
- 👤 Tư duy về user experience
- 📊 Quản lý state và data consistency

🚀 GÓI Ý PHÁT TRIỂN TIẾP THEO:

- 📝 Thêm validation cho email và số điện thoại
- 📁 Implement sort contacts theo nhiều tiêu chí
- 💾 Thêm backup và restore dữ liệu
- 🔍 Cải thiện tìm kiếm với từ khóa một phần
- 💻 Thêm more user-friendly interface

🌟 TÍNH NĂNG NÂNG CAO (Học ở tuần 28+):

- 📁 File I/O: Lưu/đọc dữ liệu từ file (tuần 28)
- 🎯 Functions: Refactor code thành functions (tuần 28)
- 牮 Classes & OOP: Tạo Contact class (tuần 28)
- 💻 GUI: Giao diện đồ họa với tkinter (sau khóa học)
- 🗄️ Database: Kết nối SQLite để lưu trữ (nâng cao)
- 🌐 Web: Chuyển thành web app với Flask (nâng cao)

💼 Ý TƯỞNG DỰ ÁN TƯƠNG TỰ:

- 📝 Todo List Manager
- 💰 Personal Finance Tracker
- 📚 Library Management System
- 🎮 Simple Text-based Game
- 📈 Student Grade Tracker
- 🛒 Shopping List App

6.2 Checklist hoàn thành và certificate

```
In [10]: print("\n📝 CHECKLIST HOÀN THÀNH DỰ ÁN")
print("=*50")
```

```

# Technical checklist
print("📝 TECHNICAL REQUIREMENTS:")
technical_items = [
    "Khởi tạo cấu trúc dữ liệu hoàn chỉnh",
    "Code thêm contact hoạt động đúng",
    "Code hiển thị contact chi tiết và ngắn gọn",
    "Code tìm kiếm (name, group, city, favorites)",
    "Code cập nhật và xóa contact",
    "Code thống kê chi tiết",
    "Demo hệ thống menu điều hướng",
    "Test tất cả chức năng không lỗi",
    "Sử dụng đúng List, Dict, Tuple, Set",
    "Code có comments và logic rõ ràng"
]

for item in technical_items:
    print(f"    □ {item}")

# Functional checklist
print("\n⚙️ FUNCTIONAL REQUIREMENTS:")
functional_items = [
    "Có ít nhất 3 contacts demo",
    "Thống kê hiển thị đúng số liệu",
    "Tìm kiếm trả về kết quả chính xác",
    "Cập nhật dữ liệu sync với thống kê",
    "Menu demo dễ hiểu và logic",
    "Xử lý edge cases cơ bản"
]

for item in functional_items:
    print(f"    □ {item}")

# Presentation checklist
print("\n🗣 PRESENTATION REQUIREMENTS:")
presentation_items = [
    "Hiểu và giải thích được code đã viết",
    "Demo ứng dụng chạy mượt mà",
    "Trả lời được câu hỏi về thiết kế",
    "Nêu được challenges và solutions",
    "Giải thích cách sử dụng 4 data structures",
    "Tự tin trình bày trong 8-10 phút"
]

for item in presentation_items:
    print(f"    □ {item}")

print("\n💡 Đánh dấu ✅ cho những mục bạn đã hoàn thành!")

# Generate certificate
print("\n" + "="*70)
print("🏆" + "*25 + "CERTIFICATE" + "*25 + "🏆")
print("=*70")
print()
print(" *20 + "📝 CHỨNG NHẬN HOÀN THÀNH")
print()
print(" *15 + "🎓 MINI PROJECT: CONTACT MANAGER")

```

```
print()
print(" *10 + "Chứng nhận rằng bạn đã thành công hoàn thành")
print(" *8 + "Mini Project Contact Manager sử dụng Python cơ bản")
print()
print(" *18 + "📝 KIẾN THỨC ĐÃ THÀNH THẠO:")
print(" *15 + "✅ Python Data Structures (List, Dict, Tuple, Set)")
print(" *15 + "✅ Variables và Data Types")
print(" *15 + "✅ Operators và Comparisons")
print(" *15 + "✅ If/else statements và Logic")
print(" *15 + "✅ For/While Loops và Iterations")
print(" *15 + "✅ String Operations và Data Processing")
print(" *15 + "✅ Problem Solving và System Thinking")
print()
print(" *20 + "📅 Ngày hoàn thành: 3/07/2025")
print(" *20 + "⌚ Thời gian đầu tư: 3 giờ")
print()
print(" *15 + "🌟 Bạn đã sẵn sàng cho Functions & OOP!")
print()
print("=*70)
print("🎉 + " *15 + "CHÚC MỪNG BẠN!" + " *15 + "🎉")
print("=*70)
```

 CHECKLIST HOÀN THÀNH DỰ ÁN

 TECHNICAL REQUIREMENTS:

- Khởi tạo cấu trúc dữ liệu hoàn chỉnh
- Code thêm contact hoạt động đúng
- Code hiển thị contact chi tiết và ngắn gọn
- Code tìm kiếm (name, group, city, favorites)
- Code cập nhật và xóa contact
- Code thống kê chi tiết
- Demo hệ thống menu điều hướng
- Test tất cả chức năng không lỗi
- Sử dụng đúng List, Dict, Tuple, Set
- Code có comments và logic rõ ràng

 FUNCTIONAL REQUIREMENTS:

- Có ít nhất 3 contacts demo
- Thống kê hiển thị đúng số liệu
- Tìm kiếm trả về kết quả chính xác
- Cập nhật dữ liệu sync với thống kê
- Menu demo dễ hiểu và logic
- Xử lý edge cases cơ bản

 PRESENTATION REQUIREMENTS:

- Hiểu và giải thích được code đã viết
- Demo ứng dụng chạy mượt mà
- Trả lời được câu hỏi về thiết kế
- Nêu được challenges và solutions
- Giải thích cách sử dụng 4 data structures
- Tự tin trình bày trong 8-10 phút

💡 Đánh dấu  cho những mục bạn đã hoàn thành!



CERTIFICATE


 CHỨNG NHẬN HOÀN THÀNH

 MINI PROJECT: CONTACT MANAGER

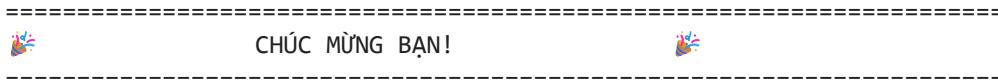
Chứng nhận rằng bạn đã thành công hoàn thành
Mini Project Contact Manager sử dụng Python cơ bản

 KIẾN THỨC ĐÃ THÀNH THẠO:

- Python Data Structures (List, Dict, Tuple, Set)
- Variables và Data Types
- Operators và Comparisons
- If/else statements và Logic
- For/While Loops và Iterations
- String Operations và Data Processing
- Problem Solving và System Thinking

 Ngày hoàn thành: 3/07/2025
 Thời gian đầu tư: 3 giờ

🌟 Bạn đã sẵn sàng cho Functions & OOP!



PROJECT TODO: CUSTOMIZATION

```
In [11]: print("\n📝 PROJECT TODO: TẠO CONTACT MANAGER CỦA RIÊNG BẠN")
print("=*60)
print("Thời gian: 60 phút")
print()

print("🎯 BƯỚC 1: Thay đổi thông tin ứng dụng (5 phút)")
print("  - Đổi tên ứng dụng thành tên riêng của bạn")
print("  - Thay author thành tên của bạn")
print("  - Viết description bằng lời của bạn")
print()

print("👤 BƯỚC 2: Thêm contacts thật của bạn (15 phút)")
print("  - Thêm 5-7 contacts: gia đình, bạn bè, giáo viên...")
print("  - Sử dụng thông tin thật để dễ test")
print("  - Đảm bảo có ít nhất 3 nhóm khác nhau")
print("  - Đặt 1-2 contacts làm yêu thích")
print()

print("🔍 BƯỚC 3: Test tất cả chức năng (15 phút)")
print("  - Test tìm kiếm theo tên, nhóm, thành phố")
print("  - Test cập nhật thông tin contact")
print("  - Test thống kê và đảm bảo số liệu đúng")
print("  - Test xóa và thêm contact")
print()

print("🎨 BƯỚC 4: Tùy chỉnh thêm (15 phút)")
print("  - Thêm thêm contacts nếu muốn")
print("  - Thay đổi các thông báo và icon")
print("  - Tùy chỉnh cách hiển thị thông tin")
print("  - Thêm validation đơn giản (kiểm tra @ trong email)")
print()

print("⚡ BƯỚC 5: Chuẩn bị presentation (10 phút)")
print("  - Tạo outline trình bày 8-10 phút")
print("  - Chuẩn bị demo script")
print("  - Suy nghĩ về challenges và solutions")
print("  - Chuẩn bị trả lời câu hỏi")
print()

print("💡 MẸO THÀNH CÔNG:")
tips = [
    "🔧 Test code sau mỗi thay đổi để tránh lỗi",
    "📝 Ghi chú lại các thay đổi bạn đã làm",
    "🎯 Tập trung vào việc hiểu logic hơn là ghi nhớ code",
    "💬 Chuẩn bị giải thích WHY bạn chọn cách làm này",
    "💡 Đểng ngại sáng tạo và thêm ý tưởng riêng",
    "⌚ Quản lý thời gian: ưu tiên hoàn thành trước khi tối ưu"
]
```

```
]

for tip in tips:
    print(f"    {tip}")

print(f"\n🎯 MISSION: Tạo Contact Manager phản ánh cá tính của bạn!")
print(f"🚀 READY: Sẵn sàng cho Functions & OOP ở tuần 28!")
print(f"🏆 SUCCESS: Bạn đã master Python Data Structures!")

print("\n" + "="*60)
print("🎉 CHÚC MỪNG! BẠN ĐÃ HOÀN THÀNH MINI PROJECT! 🎉")
print("=".*60)
```

 PROJECT TODO: TẠO CONTACT MANAGER CỦA RIÊNG BẠN

=====
Thời gian: 60 phút

 BƯỚC 1: Thay đổi thông tin ứng dụng (5 phút)

- Đổi tên ứng dụng thành tên riêng của bạn
- Thay author thành tên của bạn
- Viết description bằng lời của bạn

 BƯỚC 2: Thêm contacts thật của bạn (15 phút)

- Thêm 5-7 contacts: gia đình, bạn bè, giáo viên...
- Sử dụng thông tin thật để dễ test
- Đảm bảo có ít nhất 3 nhóm khác nhau
- Đặt 1-2 contacts làm yêu thích

 BƯỚC 3: Test tất cả chức năng (15 phút)

- Test tìm kiếm theo tên, nhóm, thành phố
- Test cập nhật thông tin contact
- Test thống kê và đảm bảo số liệu đúng
- Test xóa và thêm contact

 BƯỚC 4: Tùy chỉnh thêm (15 phút)

- Thêm thêm contacts nếu muốn
- Thay đổi các thông báo và icon
- Tùy chỉnh cách hiển thị thông tin
- Thêm validation đơn giản (kiểm tra @ trong email)

 BƯỚC 5: Chuẩn bị presentation (10 phút)

- Tạo outline trình bày 8-10 phút
- Chuẩn bị demo script
- Suy nghĩ về challenges và solutions
- Chuẩn bị trả lời câu hỏi

 MẸO THÀNH CÔNG:

-  Test code sau mỗi thay đổi để tránh lỗi
-  Ghi chú lại các thay đổi bạn đã làm
-  Tập trung vào việc hiểu logic hơn là ghi nhớ code
-  Chuẩn bị giải thích WHY bạn chọn cách làm này
-  Đừng ngại sáng tạo và thêm ý tưởng riêng
-  Quản lý thời gian: ưu tiên hoàn thành trước khi tối ưu

 MISSION: Tạo Contact Manager phản ánh cá tính của bạn!

 READY: Sẵn sàng cho Functions & OOP ở tuần 28!

 SUCCESS: Bạn đã master Python Data Structures!

=====
 CHÚC MỪNG! BẠN ĐÃ HOÀN THÀNH MINI PROJECT! 
=====

6.3 Chuẩn bị cho tuần tiếp theo

Tuần 28 - Functions & OOP cơ bản:

Kiến thức nền tảng từ Mini Project:

- Đã biết viết functions cơ bản
- Hiểu về parameters và return values
- Biết tổ chức code thành modules
- Có kinh nghiệm với complex data structures

Sẽ học tiếp:

- Functions nâng cao (scope, default params, lambda)
- File handling (đọc/ghi file)
- OOP concepts (classes, objects, methods)
- Mini project: Bank Account System

Cách áp dụng Contact Manager:

- Refactor thành Contact class
- Lưu dữ liệu ra file
- Tạo ContactManager class
- Sử dụng inheritance và encapsulation

 **Bạn đã có nền tảng vững chắc để học OOP!**