

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Phát triển ứng dụng web quản lý ghi chú cá nhân sử dụng
framework Django

GVHD: TỪ LÃNG PHIÊU
SV: PHẠM PHÚ KHÁNH - 3120410245

TP. HỒ CHÍ MINH, THÁNG 4/2024

Mục lục

1	MÔ TẢ BÀI TOÁN	2
2	CÁC CÔNG NGHỆ SỬ DỤNG	3
2.1	Nền tảng web	3
2.2	Ngôn ngữ lập trình Python	3
2.3	Framework Django	3
2.4	MariaDB thông qua XAMPP	3
2.5	LangChain	4
3	CƠ SỞ DỮ LIỆU	5
4	DEMO CHƯƠNG TRÌNH	6
4.1	Chức năng đăng nhập	6
4.1.1	Giao diện	6
4.1.2	Xử lý	6
4.2	Chức năng đăng xuất	7
4.2.1	Giao diện	7
4.2.2	Xử lý	7
4.3	Chức năng đăng ký	8
4.3.1	Giao diện	8
4.3.2	Xử lý	8
4.4	Chức năng Thêm ghi chú	9
4.4.1	Giao diện	9
4.4.2	Xử lý	9
4.5	Chức năng Xóa ghi chú	10
4.5.1	Giao diện	10
4.5.2	Xử lý	10
4.6	Chức năng Sửa thông tin ghi chú	11
4.6.1	Giao diện	11
4.6.2	Xử lý	11
4.7	Chức năng Tìm kiếm ghi chú	12
4.7.1	Giao diện	12
4.7.2	Xử lý	12
4.8	Chức năng Tổng hợp nội dung ghi chú	13
4.8.1	Giao diện	13
4.8.2	Xử lý	13
5	KẾT LUẬN	14
5.1	Tổng kết công việc đã thực hiện	14
5.2	Các mặt hạn chế	14
5.3	Hướng phát triển tiếp theo trong tương lai	14



1 MÔ TẢ BÀI TOÁN

Hiện nay, thông qua quá trình không ngừng cập nhật của kiến thức và nguồn thông tin, nhu cầu ghi nhớ và tổng hợp thông tin ngày càng được chú trọng. Việc ghi chép lại thông tin đang dần trở nên phổ biến hơn với hầu hết các đối tượng, đặc biệt đối với những đối tượng là học sinh, sinh viên, người đi làm, việc ghi chú vô cùng cần thiết để ghi nhớ và lên kế hoạch. Tùy thuộc vào ngữ cảnh sử dụng, đã có nhiều cách thức nhau trong việc ghi chú. Một số cách thức tương đối phổ biến được dùng rộng rãi hiện nay như:

- Ghi chú thông qua giấy viết thông thường
- Ghi âm lại các thông tin
- Ghi chú thông qua các ứng dụng ghi chú như Google Keep và Notion

Mục tiêu của đề án nhằm tạo ra một phần mềm quản lý ghi chú cá nhân với những tác vụ quản lý cơ bản như thêm, xóa, sửa,... Bên cạnh các chức năng cơ bản, ứng dụng còn hướng đến việc tích hợp công nghệ trí tuệ nhân tạo để nâng cao khả năng quản lý nội dung ghi chú như tổng hợp nội dung.

2 CÁC CÔNG NGHỆ SỬ DỤNG

2.1 Nền tảng web

Dự kiến phát triển ứng dụng trên nền tảng web (web-based) vì là đây một loại hình ứng dụng, dịch vụ hoặc mà người dùng truy cập và tương tác thông qua trình duyệt web trên internet. Các ứng dụng web-based không yêu cầu người dùng cài đặt phần mềm hay ứng dụng đặc biệt trên thiết bị của họ mà chỉ cần kết nối internet để truy cập và sử dụng.

2.2 Ngôn ngữ lập trình Python

Python là một lựa chọn phù hợp cho việc phát triển ứng dụng này vì có nhiều lợi ích như:

- Có cú pháp đơn giản và dễ đọc, giúp tăng tính đảm bảo và hiểu biết của đội ngũ phát triển.
- Python là một ngôn ngữ đa năng và mạnh mẽ, có thể được sử dụng cho nhiều mục đích khác nhau từ phát triển web, xử lý dữ liệu, đến trí tuệ nhân tạo và học máy.
- Có một cộng đồng lớn và tích cực, cung cấp nhiều thư viện và framework hữu ích giúp việc phát triển ứng dụng trở nên nhanh chóng và dễ dàng hơn.
- Miễn phí và mã nguồn mở, giúp tiết kiệm chi phí phát triển và tạo điều kiện thuận lợi cho sự mở rộng và phát triển của dự án trong tương lai.

2.3 Framework Django

Sử dụng Django để phát triển ứng dụng mang lại nhiều ưu điểm quan trọng.

- Django là một framework phát triển web mạnh mẽ và linh hoạt, được thiết kế để tăng tốc quá trình phát triển và cung cấp cấu trúc tổ chức chặt chẽ cho dự án. Khả năng tích hợp sẵn của Django, như hệ thống quản trị admin và xác thực người dùng, giúp giảm thời gian và công sức cần thiết để triển khai các tính năng chính của ứng dụng.
- Sự phổ biến và hỗ trợ mạnh mẽ từ cộng đồng Django cung cấp tài liệu phong phú, hướng dẫn chi tiết và các thư viện mở rộng đáng tin cậy, tạo điều kiện thuận lợi cho việc nghiên cứu và phát triển dự án.
- Django được xây dựng trên ngôn ngữ Python, với cú pháp rõ ràng và dễ đọc, giúp giảm thiểu các lỗi và tăng tính bảo trì của mã nguồn.
- Với những ưu điểm này, Django là một lựa chọn lý tưởng cho các nhà phát triển và nhà nghiên cứu muốn xây dựng các ứng dụng web phức tạp và chất lượng cao.

2.4 MariaDB thông qua XAMPP

MariaDB là một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ và phổ biến, cung cấp khả năng:

- Lưu trữ và truy xuất dữ liệu hiệu quả. Với tính năng ACID (Atomicity, Consistency, Isolation, Durability), MariaDB đảm bảo tính nhất quán và an toàn dữ liệu.
- MariaDB là một giải pháp mã nguồn mở, miễn phí và được cộng đồng lập trình viên rộng lớn hỗ trợ. Điều này mang lại sự linh hoạt và tính mở rộng cho dự án, cũng như cung cấp nhiều tài liệu và hỗ trợ từ cộng đồng.

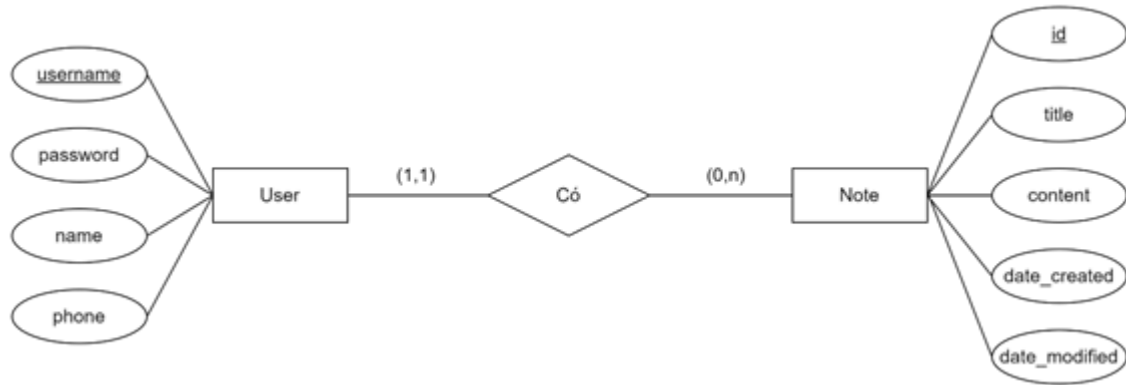


- MariaDB có khả năng xử lý tải cao và tối ưu hóa hiệu suất, cho phép ứng dụng mở rộng khi cần thiết. Nó cung cấp các công cụ quản lý dữ liệu mạnh mẽ như indexes, query optimization, và caching để tối ưu hóa hiệu suất của ứng dụng.

2.5 LangChain

LangChain là một framework phát triển ứng dụng cho phép tích hợp các mô hình ngôn ngữ. Nó giúp các ứng dụng nhận biết ngữ cảnh và lý luận dựa trên các mô hình ngôn ngữ. Các tính năng chính bao gồm khả năng tương tác với các mô hình ngôn ngữ, kết nối và truy xuất dữ liệu, tạo chuỗi mô hình, tác tử để quyết định hành động tối ưu.

3 CƠ SỞ DỮ LIỆU



Bảng User:

- username: Tên tài khoản của người dùng, đồng thời cũng là email của người dùng.
- password: Mật khẩu cho tài khoản của người dùng.
- name: Tên người dùng.
- phone: Số điện thoại người dùng.

Bảng Note:

- id: id của ghi chú, sử dụng để xác định các ghi chú.
- title: Tiêu đề ghi chú (hoặc tên ghi chú), sử dụng để đặt một tiêu đề ngắn có liên quan cho ghi chú.
- content: Nội dung ghi chú, sử dụng để ghi chép lại các thông tin cho ghi chú.
- date_created: Ngày tạo ghi chú.
- date_modified: Ngày sửa đổi ghi chú gần nhất.

4 DEMO CHƯƠNG TRÌNH

4.1 Chức năng đăng nhập

4.1.1 Giao diện

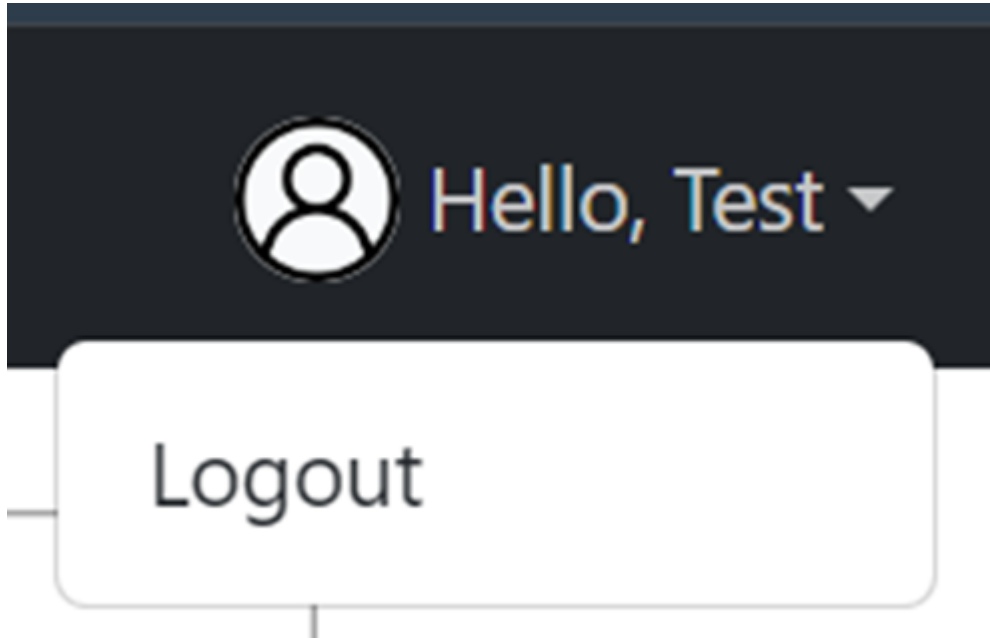
Register'." data-bbox="153 236 880 444"/>

4.1.2 Xử lý

```
1 from django.shortcuts import render, redirect
2 from django.contrib.auth import logout
3 from django.contrib import messages
4 from home.models import User
5 import hashlib
6
7 def login_user(request):
8     if request.method == "POST" and 'username' in request.POST and 'password' in
        request.POST:
9         username = request.POST['username']
10        password =
            hashlib.sha256(request.POST['password'].encode('utf-8')).hexdigest()
11        user = User.get_user_by_username(username)
12        if user is not None and user.password == password:
13            request.session['username'] = user.username
14            request.session['name'] = user.name
15            return redirect('/home')
16        else:
17            messages.error(request, 'Username or password is incorrect')
18            return render(request, 'pages/login.html', {})
19    else:
20        return render(request, 'pages/login.html', {})
```

4.2 Chức năng đăng xuất

4.2.1 Giao diện

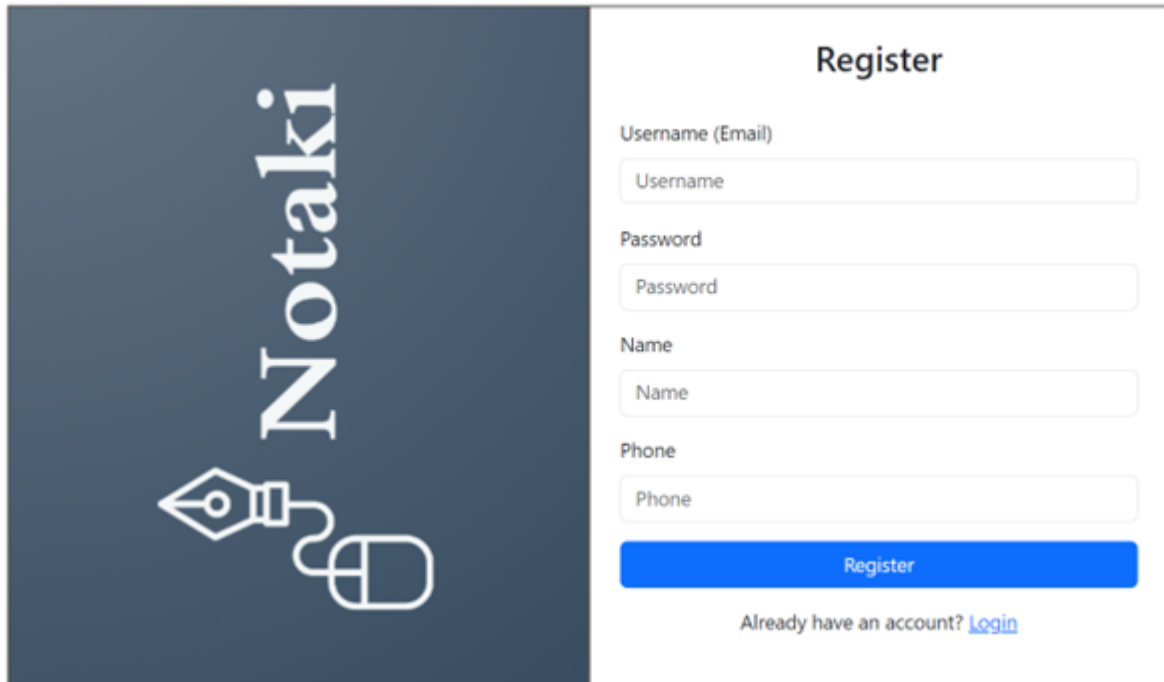


4.2.2 Xử lý

```
1 from django.shortcuts import redirect
2 from django.contrib.auth import logout
3
4 def logout_user(request):
5     logout(request)
6     return redirect('login')
```

4.3 Chức năng đăng ký

4.3.1 Giao diện



4.3.2 Xử lý

```
1 from django.shortcuts import render, redirect
2 from django.contrib.auth import logout
3 from django.contrib import messages
4 from home.models import User
5 import hashlib
6
7 def register_user(request):
8     if request.method == "POST":
9         username = request.POST['username']
10        if User.get_user_by_username(username) is not None:
11            messages.error(
12                request, 'Username already exists, please try another one')
13            return render(request, 'pages/register.html', {})
14
15        password =
16            hashlib.sha256(request.POST['password'].encode('utf-8')).hexdigest()
17        name = request.POST['name']
18        phone = request.POST['phone']
19
20        user = User(username=username, password=password,
21                    name=name, phone=phone)
22        user.save()
```



```
22
23         messages.success(request, 'User registered successfully')
24         return redirect('login')
25     else:
26         return render(request, 'pages/register.html', {})
```

4.4 Chức năng Thêm ghi chú

4.4.1 Giao diện

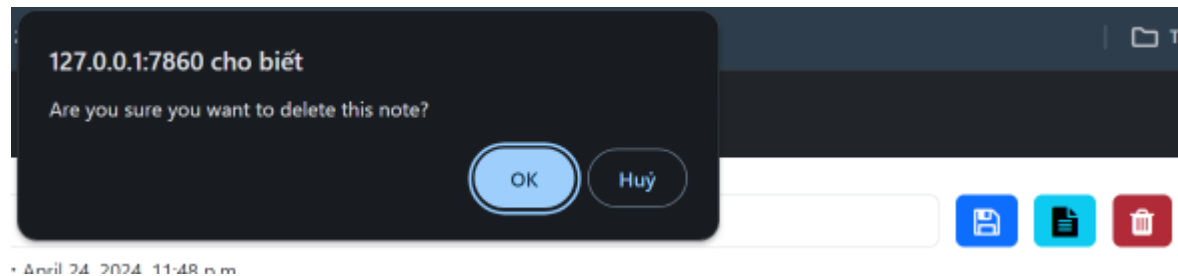
4.4.2 Xử lý

```
1     import json
2     import openaikey
3     import base64
4     from django.shortcuts import render, redirect
5     from django.views.decorators.csrf import csrf_exempt
6     from django.http import Http404
7     from home.models import Note, User
8     from django.utils import timezone
9     from django.http import JsonResponse
10    from django.contrib import messages
11
12    def add_note(request):
13        note = Note()
14        user = User.objects.get(username=request.session.get("username"))
15        note.username = user
16        note.title = ""
17        note.content = ""
18        note.date_modified = timezone.now()
19        note.date_created = timezone.now()
20        note.save()
21        note_id = note.id
22        messages.success(request, "Note added successfully")
23        return redirect(f"./edit?note_id={note_id}")
```



4.5 Chức năng Xóa ghi chú

4.5.1 Giao diện



4.5.2 Xử lý

```
1 import json
2 import openaikey
3 import base64
4 from django.shortcuts import render, redirect
5 from django.views.decorators.csrf import csrf_exempt
6 from django.http import Http404
7 from home.models import Note, User
8 from django.utils import timezone
9 from django.http import JsonResponse
10 from django.contrib import messages
11
12 def delete_note(request):
13     note_id = request.GET.get("note_id")
14     note = Note.objects.get(id=note_id)
15     note.delete()
16     messages.success(request, "Note deleted successfully")
17     return redirect("home")
```



4.6 Chức năng Sửa thông tin ghi chú

4.6.1 Giao diện

◀ Notaki

Title

Note summary Test



Created: April 19, 2024, 1:42 p.m.

Modified: April 19, 2024, 1:51 p.m.

Content

Ngày lễ 30 tháng 4, tên chính thức là Ngày Giải phóng Miền Nam, thống nhất đất nước, Ngày Chiến thắng, Ngày Thống nhất là một ngày lễ quốc gia của Việt Nam, đánh dấu sự kiện chấm dứt Chiến tranh Việt Nam khi Tổng thống Dương Văn Minh và Thủ tướng Võ Văn Mẫu của Việt Nam Cộng hòa tuyên bố đầu hàng vô điều kiện Chính phủ Cách mạng lâm thời Cộng hòa miền Nam Việt Nam vào trưa ngày 30 tháng 4 năm 1975. Ngày này là kết quả trực tiếp của Chiến dịch Mùa Xuân năm 1975 và là một mốc quan trọng trong lịch sử Việt Nam. Sự kiện 30 tháng 4 diễn ra sau khi tất cả công dân và binh lính Mỹ cùng với hàng ngàn người Việt phục vụ cho chính thể Việt Nam Cộng hòa ở miền Nam Việt Nam di tản khỏi Sài Gòn. Năm 1976, Việt Nam chính thức tuyên cử bầu Quốc hội thống nhất và Sài Gòn đổi tên thành Thành phố Hồ Chí Minh, theo tên của cố Chủ tịch Việt Nam Dân chủ Cộng hòa Hồ Chí Minh. Hằng năm, tại Việt Nam có rất nhiều hoạt động được tổ chức để kỷ niệm ngày này. Nó còn là ngày nghỉ lễ quốc gia, được ghi trong các văn bản pháp luật với tên gọi "Ngày Chiến thắng". Trong khi đó, tại một số cộng đồng người Việt hải ngoại, gồm những người đã từng phục vụ cho Hoa Kỳ và Việt Nam Cộng hòa, có tư tưởng chống chính phủ hiện tại của Việt Nam thì gọi đây là ngày Tháng Tư đen, Ngày Sài Gòn thất thủ, hoặc Ngày Quốc Hận.

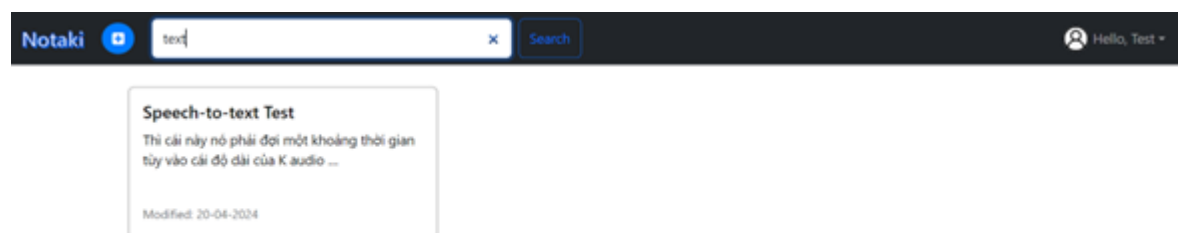
4.6.2 Xử lý

```
1 import json
2 import openaikey
3 import base64
4 from django.shortcuts import render, redirect
5 from django.views.decorators.csrf import csrf_exempt
6 from django.http import Http404
7 from home.models import Note, User
8 from django.utils import timezone
9 from django.http import JsonResponse
10 from django.contrib import messages
11
12 def edit_note(request):
13     if request.method == "POST":
14         title = request.POST.get("title")
15         content = request.POST.get("content")
16
17         note_id = request.GET.get("note_id")
18
19         if not note_id:
20             raise Http404("Note ID parameter not found in the URL")
21
22         try:
23             note = Note.objects.get(id=note_id)
24         except Note.DoesNotExist:
25             raise Http404("Note does not exist")
26
27         note.title = title
28         note.content = content
```

```
29         note.date_modified = timezone.now()
30         note.save()
31         messages.success(request, "Note saved successfully")
32         return redirect("home")
33     else:
34         # Handle GET requests (if any)
35         pass
```

4.7 Chức năng Tìm kiếm ghi chú

4.7.1 Giao diện

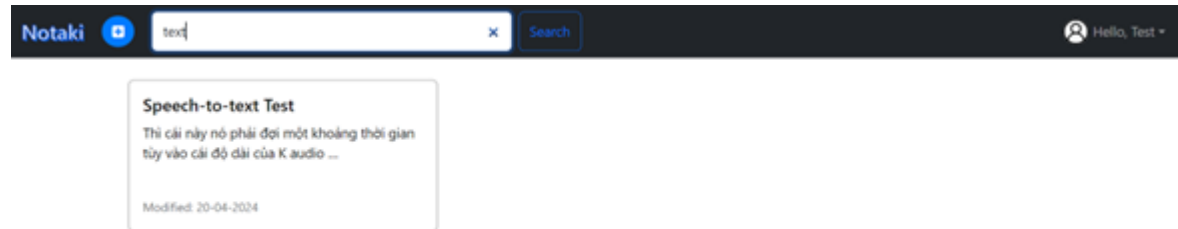


4.7.2 Xử lý

```
1     from django.shortcuts import render, redirect
2     from .models import Note
3     from django.db.models import Q
4
5     @check_login
6     def find_note(request):
7         query = request.GET.get('query', '').strip()
8         if query == '':
9             return render(request, 'pages/home.html')
10        else:
11            Data = {
12                'Notes': Note.objects.filter(
13                    Q(title__icontains=query) | Q(content__icontains=query),
14                    username=request.session['username']
15                ).order_by('-date_modified')
16            }
17        return render(request, 'pages/home.html', Data)
```

4.8 Chức năng Tổng hợp nội dung ghi chú

4.8.1 Giao diện



4.8.2 Xử lý

```
1 from langchain_openai import ChatOpenAI
2 from langchain_core.messages import HumanMessage
3 import json
4 import openaikey
5 import base64
6 from django.shortcuts import render, redirect
7 from django.views.decorators.csrf import csrf_exempt
8 from django.http import Http404
9 from home.models import Note, User
10 from django.utils import timezone
11 from django.http import JsonResponse
12 from django.contrib import messages
13 @csrf_exempt
14 def summarize_note(request):
15     chat = ChatOpenAI(openai_api_key=openaikey.OPENAI_API_KEY)
16     if request.method == "POST":
17         data = json.loads(request.body)
18         note_content = data.get("note_content")
19         note_content = (
20             "I have a note which contains the following content: \n"
21             + note_content
22             + "\n\n"
23             + "Summarize the note for me (in bullet or numbering format if
24                 necessary) in the language that I had written in the content. No
25                 yapping, Limit prose, No fluff."
26         )
27         response = chat.invoke([HumanMessage(content=note_content)]).to_json()[
28             "kwargs"
29         ]["content"]
30         print(response)
31         return JsonResponse({"summarization": response})
32     else:
33         return JsonResponse({"error": "Method not allowed"}, status=405)
```



5 KẾT LUẬN

5.1 Tổng kết công việc đã thực hiện

- Nắm bắt nghiệp vụ cũng như quy trình hoạt động của hệ thống thông qua các sơ đồ xử lý.
- Phát triển các chức năng cơ bản của một ứng dụng quản lý ghi chú cá nhân.
- Thành công tích hợp công nghệ AI đã đề ra: OpenAI LangChain.

5.2 Các mặt hạn chế

- Thời gian phát triển đồ án có phần eo hẹp dẫn đến thời gian để thực hiện các công việc hàng tuần có phần trễ hơn so với thời gian đã thống nhất trong kế hoạch.
- Phương pháp và kiến thức được sử dụng để phát triển đồ án tương đối mới, đặc biệt là framework Django.

5.3 Hướng phát triển tiếp theo trong tương lai

- Khắc phục các thành phần giao diện để giúp ứng dụng tách biệt với các ứng dụng khác có mặt trên thị trường, đồng thời nhằm tăng trải nghiệm người dùng trong quá trình sử dụng.
- Tích hợp thêm các API về AI khác như nhập liệu bằng giọng nói, dịch thuật,...