

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



ĐỒ ÁN TỐT NGHIỆP

XÂY DỰNG ỨNG DỤNG HỖ TRỢ TÌM KIẾM VIỆC LÀM TRÊN NỀN TẢNG ASP.NET CORE WEB API VÀ VUEJS

SINH VIÊN THỰC HIỆN	: PHẠM KHẮC HUY
MÃ SINH VIÊN	: 1451020107
KHOA	: CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



PHẠM KHẮC HUY

**XÂY DỰNG ỨNG DỤNG HỖ TRỢ TÌM
KIẾM VIỆC LÀM TRÊN NỀN TẢNG
ASP.NET CORE WEB API VÀ VUEJS**

**CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN
MÃ SỐ : 74.80.201**

NGƯỜI HƯỚNG DẪN : TS. LƯƠNG CAO ĐÔNG

HÀ NỘI - 2024

LỜI CAM ĐOAN

Em là Phạm Khắc Huy, sinh viên của trường Đại học Đại Nam, cam kết rằng báo cáo đồ án tốt nghiệp có đề tài là “*Xây dựng ứng dụng hỗ trợ tìm kiếm việc làm trên nền tảng ASP.Net Core API và VueJS*” được viết bởi chính em dựa trên kinh nghiệm và kiến thức thực tế trong quá trình học tập.

Em xin cam kết báo cáo đồ án này được hoàn thành dựa trên kết quả nghiên cứu, học tập của em và các kết quả này chưa được dùng cho bất cứ báo cáo (báo cáo, khóa luận tốt nghiệp) cùng cấp nào khác.

Nếu có bất kỳ phát hiện về vi phạm hoặc không trung thực, em sẽ chịu trách nhiệm pháp lý và chấp nhận các hình thức xử lý mà trường Đại học Đại Nam quy định.

Em cam đoan rằng các thông tin và kết quả được trình bày trong báo cáo là trung thực và xác thực.

Hà Nội, ngày...tháng... năm 2024

Sinh viên

(ký và ghi rõ họ và tên)

LỜI MỞ ĐẦU

Trong bối cảnh thị trường lao động ngày càng cạnh tranh và biến động không ngừng, việc tìm kiếm việc làm trở thành một thách thức lớn đối với nhiều sinh viên mới ra trường. Các kênh thông tin truyền thống và các nền tảng tìm kiếm việc làm hiện tại chưa đáp ứng được nhu cầu đa dạng và thay đổi liên tục của người lao động. Điều này khiến quá trình tìm kiếm việc làm trở nên khó khăn và mất nhiều thời gian.

Xuất phát từ thực tế này, em đã chọn đề tài "*Xây dựng ứng dụng hỗ trợ tìm kiếm việc làm trên nền tảng ASP.NET CORE WEB API và VueJS*" với mong muốn tạo ra một nền tảng trực tuyến hữu ích, giúp người dùng dễ dàng tìm kiếm các cơ hội việc làm phù hợp với kỹ năng và mong muốn của mình. Ứng dụng này sẽ cung cấp các tính năng như đăng tin tuyển dụng, tìm kiếm việc làm theo địa điểm, lĩnh vực, mức lương và các yêu cầu khác. Đồng thời, người dùng cũng có thể tạo hồ sơ cá nhân để tiện lợi trong việc quản lý và ứng tuyển vào các công việc phù hợp.

Em hy vọng rằng, với ứng dụng này, sinh viên mới ra trường sẽ có nhiều cơ hội tiếp cận với các nhà tuyển dụng, từ đó nâng cao khả năng tìm được công việc phù hợp. Ứng dụng không chỉ đem lại giá trị cho người lao động mà còn hỗ trợ các nhà tuyển dụng trong việc tìm kiếm và tuyển chọn những ứng viên tiềm năng. Qua đó, em mong rằng đề tài này sẽ góp phần nâng cao hiệu quả kết nối giữa sinh viên và nhà tuyển dụng, giúp cả hai bên đạt được mục tiêu của mình trên thị trường lao động đầy thách thức hiện nay.

LỜI CẢM ƠN

Được sự phân công của khoa Công nghệ thông tin **Trường Đại học Đại Nam**, và dưới sự hướng dẫn của thầy **TS. Lương Cao Đông**, em đã hoàn thành đề tài là “**Xây dựng ứng dụng hỗ trợ tìm kiếm việc làm trên nền tảng ASP.NET CORE WEB API và VueJS**”. Để hoàn thành đồ án này em xin chân thành cảm ơn thầy đã tận tình hướng dẫn em trong suốt quá trình làm đồ án tốt nghiệp. Với sự chỉ bảo của thầy, em đã có những định hướng trong công việc triển khai và thực hiện các yêu cầu trong quá trình làm đồ án tốt nghiệp.

Em nhận ra rằng bài viết có thể còn thiếu sót và mong nhận được sự góp ý chân thành từ các thầy cô để em có thể khắc phục.

Em xin chân thành cảm ơn!

This image shows a full page of white paper with horizontal dashed lines, typical of primary school writing paper. The lines are evenly spaced and run across the entire width of the page. There are no margins, text, or other markings present.

DANH MỤC VIẾT TẮT

STT	KÝ HIỆU CHỮ VIẾT TẮT	CHỮ VIẾT TẮT ĐẦY ĐỦ
1	AI	Artificial Intelligence
2	API	Application Programming Interface
3	CRUD	Create, Read, Update, Delete
4	CSS	Cascading Style Sheets
5	DOM	Document Object Model
6	HTML	Hypertext Markup Language
7	HTTP/HTTPS	Hypertext Transfer Protocol / Hypertext Transfer Protocol Secure
8	JSON/XML	JavaScript Object Notation / eXtensible Markup Language
9	MVC	Model-View-Controller
10	UI/UX	User Interface/User Experience

MỤC LỤC

CHƯƠNG 1 TỔNG QUAN ĐỀ TÀI VÀ KHẢO SÁT THỰC TẾ	1
1.1. Khảo sát thực tế.....	1
1.1.1. Bài toán	1
1.1.2. Thực trạng.....	1
1.2. Hướng giải quyết.....	2
1.3. Mục tiêu nghiên cứu.....	2
1.4. Đối tượng nghiên cứu	3
1.5. Cơ sở khoa học.....	3
1.6. Phương pháp nghiên cứu.....	4
1.7. Ứng dụng của kết quả nghiên cứu trong đề tài	4
CHƯƠNG 2 CÁC KIẾN THỨC NỀN TẢNG	6
2.1. VueJS	6
2.1.1. VueJS là gì?	6
2.1.2. Ưu điểm và nhược điểm.	6
a. Ưu điểm	6
b. Nhược điểm	7
2.1.3. Ứng dụng của VueJS	8
2.1.4. Thư viện Axios.	8
2.1.5. Vue router	9
2.2. Typescript.....	9
2.2.1. Interfaces.....	11
2.2.2. Classes	12
2.2.3. Modules	12
2.2.4. Ưu điểm của Typescript.....	13
a. Ưu điểm	13
b. Nhược điểm	13
2.3. Asp.Net Core Web API.....	13
2.3.1. Asp.Net Core Web API là gì ?.....	13
2.3.2. Ưu điểm và nhược điểm	13
a. Ưu điểm	13

b. Nhược điểm	14
2.2.3. Ứng dụng	14
2.4. API (Application Programming Interface)	14
2.4.1. Ứng dụng	14
2.4.2. Ứng dụng của API.	15
2.4.3. Web API hoạt động như nào ?.....	16
2.4.4. Ưu điểm và nhược điểm	16
a. Ưu điểm	16
b. Nhược điểm	17
2.4.5. Ứng dụng của Web API.....	17
2.4.6. Mô hình client/server qua HTTP	17
2.5. JWT	18
2.5.1. Ưu điểm và nhược điểm	19
CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG	21
3.1. Mô tả chức năng bài toán	21
3.1.1. Chức năng cho quản trị viên	21
a. Quản lý banner.....	21
b. Quản lý tài khoản nhà tuyển dụng.....	21
c. Quản lý nhân viên.....	21
d. Quản lý đăng tin tức	21
e. Thống kê	21
3.1.2. Chức năng cho nhà tuyển dụng	22
a. Đăng nhập/đăng ký	22
b. Đăng bài tuyển dụng.....	22
c. Xem danh sách người ứng tuyển	22
d. Xem thống kê	22
e. Quản lý thông tin cá nhân.....	22
3.1.3. Chức năng cho người tìm việc.....	22
a. Đăng nhập/đăng ký	22
b. Ứng tuyển công việc.....	23
c. Xem danh sách công việc đã ứng tuyển	23

d. Quản lý hồ sơ xin việc.....	23
e. Quản lý thông tin tài khoản	23
f. Chat liên hệ AI	23
3.2. Yêu cầu bài toán.....	24
3.2.1. Yêu cầu chức năng.....	24
3.2.2. Yêu cầu phi chức năng.....	24
3.3. Thiết kế chức năng hệ thống	24
3.3.1. Usecase tổng quan	24
3.3.2. Usecase quản trị viên	25
a. Đặc tả usecase quản lý tài khoản và phân quyền.....	25
b. Đặc tả usecase quản lý đăng tin tức	26
c. Đặc tả usecase quản lý thông tin nhà tuyển dụng.....	26
d. Đặc tả usecase xem thống kê dành cho quản trị viên.....	27
e. Đặc tả usecase quản lý banner.....	28
3.3.3. Usecase nhà tuyển dụng.....	28
a. Đặc tả usecase quản lý bài đăng tuyển dụng	29
b. Đặc tả usecase quản lý thông tin cá nhân.....	30
c. Đặc tả usecase xem danh sách hồ sơ đánh giá	30
d. Đặc tả usecase xem danh sách hồ sơ ứng tuyển.....	31
e. Đặc tả usecase xem thống kê dành cho nhà tuyển dụng	31
3.3.4. Usecase người tìm việc.....	32
a. Đặc tả usecase tạo hồ sơ/cv theo mẫu trang web	32
b. Đặc tả usecase quản lý thông tin hồ sơ xin việc.....	34
c. Đặc tả usecase quản lý thông tin cá nhân	35
d. Đặc tả usecase xem danh sách công việc yêu thích	35
e. Đặc tả usecase chat với trợ lý AI	36
f. Đặc tả usecase tìm việc và ứng tuyển công việc.....	37
3.4. Biểu đồ hoạt động	38
3.4.1. Biểu đồ hoạt động đăng nhập	38
3.4.2. Biểu đồ hoạt động đăng ký	39
3.4.3. Biểu đồ hoạt động thống kê	39

3.4.4. Biểu đồ hoạt động đăng xuất	39
3.4.5. Biểu đồ hoạt động Chat AI	40
3.5. Biểu đồ tuần tự	40
3.5.1. Biểu đồ tuần tự đăng nhập	40
3.5.2. Biểu đồ tuần tự đăng ký	41
3.5.3. Biểu đồ tuần tự đăng xuất	41
3.5.4. Biểu đồ tuần tự Thống kê	42
3.5.5. Biểu đồ tuần tự Chat AI	42
3.5.6. Biểu đồ tuần tự tạo hồ sơ/ CV	43
3.5.7. Biểu đồ tuần tự Ứng tuyển việc làm	43
3.5.8. Biểu đồ tuần tự Xem danh sách nhà tuyển dụng	44
3.5.9. Biểu đồ tuần tự Xem danh sách ứng viên ứng tuyển	44
3.5.10. Biểu đồ tuần tự Gửi thông báo qua Email	45
3.6. Thiết kế cơ sở dữ liệu	46
3.6.1. Bảng Job_Seekers	46
3.6.2. Bảng Jobs	46
3.6.3. Bảng Favourite_Job	47
3.6.4. Bảng File_Cv	47
3.6.5. Bảng Employers	47
3.6.6. Bảng Notification	48
3.6.7. Bảng Refresh_Tokens	48
3.6.8. Bảng News	49
3.6.9. Bảng Permissions	49
3.6.10. Bảng Roles	49
3.6.11. Bảng Formofworks	49
3.6.12. Bảng Professions	50
3.6.13. Bảng Levelworks	50
3.6.14. Bảng Salaries	50
3.6.15. Bảng Cities	50
3.6.16. Bảng Workexperiences	50
3.6.17. Bảng Banners	51

3.6.18. Bảng Recruitement	51
3.6.19. Bảng Users	51
3.7. Biểu đồ Database Diagram	52
CHƯƠNG 4 TRIỂN KHAI VÀ KẾT QUẢ ĐẠT ĐƯỢC	53
4.1. Triển khai bài toán	53
4.1.1. Back-end	53
4.1.2. Front-end.....	54
4.1.3. Tích hợp	54
a. Google Cloud Platform.....	54
b. Rasa	54
4.2. Kết quả đạt được	55
KẾT LUẬN	67
TÀI LIỆU THAM KHẢO	68

DANH MỤC BẢNG BIỂU

Bảng 3. 1. Usecase quản lý tài khoản và phân quyền.....	26
Bảng 3. 2. Usecase quản lý đăng tin tức.....	26
Bảng 3. 3. Usecase quản lý thông tin nhà tuyển dụng.....	27
Bảng 3. 4. Usecase xem thông kê dành cho quản trị viên	28
Bảng 3. 5. Usecase quản lý banner.....	28
Bảng 3. 6. Usecase quản lý bài đăng tuyển dụng	30
Bảng 3. 7. Usecase quản lý thông tin cá nhân	30
Bảng 3. 8. Usecase xem danh sách hồ sơ đánh giá.....	31
Bảng 3. 9. Usecase xem danh sách hồ sơ đánh giá.....	31
Bảng 3. 10. Usecase xem thông kê dành cho nhà tuyển dụng.....	32
Bảng 3. 11. Usecase Tạo hồ sơ/cv theo mẫu trang web	34
Bảng 3. 12. Usecase quản lý lý thông tin hồ sơ xin việc.....	35
Bảng 3. 13. Usecase quản lý thông tin cá nhân	35
Bảng 3. 14. Usecase danh sách công việc yêu thích.....	36
Bảng 3. 15. Usecase chat với trợ lý AI.....	37
Bảng 3. 16. Usecase tìm việc và ứng tuyển công việc	38
Bảng 3. 17. Bảng Job_Seekers	46
Bảng 3. 18. Bảng Jobs	47
Bảng 3. 19. Bảng Favourite_Job	47
Bảng 3. 20. Bảng File_Cv	47
Bảng 3. 21. Bảng Employers.....	48
Bảng 3. 22. Bảng Notification	48
Bảng 3. 23. Bảng Refresh_Tokens	49
Bảng 3. 24. Bảng News	49
Bảng 3. 25. Bảng Permissions.....	49
Bảng 3. 26. Bảng Roles	49
Bảng 3. 27. Bảng Formofworks.....	50
Bảng 3. 28. Bảng Professions.....	50
Bảng 3. 29. Bảng Levelworks	50
Bảng 3. 30. Bảng Salaries.....	50
Bảng 3. 31. Bảng Cities.....	50

Bảng 3. 32. Bảng Workexperiences	51
Bảng 3. 33. Bảng Banners	51
Bảng 3. 34. Bảng Recruitement.....	51
Bảng 3. 35. Bảng Users	52

DANH MỤC HÌNH ẢNH

Hình 2. 1. API (Application Programming Interface)	15
Hình 2. 2. Mô hình Client-Server	17
Hình 3. 1. Usecase tổng quan	24
Hình 3. 2. Usecase nhà tuyển dụng	25
Hình 3. 3. Usecase nhà tuyển dụng	29
Hình 3. 4. Usecase người tìm việc.....	32
Hình 3. 5. Biểu đồ hoạt động đăng nhập	38
Hình 3. 6. Biểu đồ hoạt động đăng ký	39
Hình 3. 7. Biểu đồ hoạt động thống kê.....	39
Hình 3. 8. Biểu đồ hoạt động đăng xuất	40
Hình 3. 9. Biểu đồ hoạt động Chat AI	40
Hình 3. 10. Biểu đồ tuần tự đăng nhập	41
Hình 3. 11. Biểu đồ hoạt động đăng ký	41
Hình 3. 12. Biểu đồ tuần tự đăng xuất.....	42
Hình 3. 13. Biểu đồ tuần tự thống kê.....	42
Hình 3. 14. Biểu đồ tuần tự Chat AI.....	43
Hình 3. 15. Biểu đồ tuần tự tạo hồ sơ/ CV	43
Hình 3. 16. Biểu đồ tuần tự Ứng tuyển việc làm.....	44
Hình 3. 17. Biểu đồ tuần tự Xem danh sách nhà tuyển dụng	44
Hình 3. 18. Biểu đồ tuần tự Xem danh sách ứng viên ứng tuyển.....	45
Hình 3. 19. Biểu đồ tuần tự Gửi thông báo qua Email	45
Hình 3. 20. Biểu đồ Database Diagram	52
Hình 4. 1. Màn hình trang chủ.....	55
Hình 4. 2. Màn hình trang việc làm.....	55
Hình 4. 3. Màn hình chi tiết công ty	56
Hình 4. 4. Màn hình trang công ty.....	56
Hình 4. 5. Màn hình chat với trợ lý AI.....	57
Hình 4. 6. Màn hình ứng tuyển việc làm	57
Hình 4. 7. Màn hình tạo hồ sơ/cv	58
Hình 4. 8. Màn hình chi tiết công việc	58
Hình 4. 9. Màn hình đăng nhập	59

Hình 4. 10. Màn hình đăng ký	59
Hình 4. 11. Màn hình thông tin tài khoản.....	60
Hình 4. 12. Màn hình tin tức	60
Hình 4. 13. Màn hình quản lý hồ sơ/cv	61
Hình 4. 14. Màn hình dashboard nhà tuyển dụng.....	61
Hình 4. 15. Màn hình đăng tin tuyển dụng.....	62
Hình 4. 16. Màn hình danh sách ứng viên.....	62
Hình 4. 17. Màn hình hồ sơ/cv phù hợp và gửi thư hẹn phỏng vấn	63
Hình 4. 18. Màn hình email sau khi gửi cho người tìm việc	63
Hình 4. 19. Màn hình thông tin tài khoản nhà tuyển dụng.....	63
Hình 4. 20. Màn hình dashboard quản trị viên	64
Hình 4. 21. Màn hình quản lý nhân viên	64
Hình 4. 22. Màn hình quản lý hình thức làm việc	64
Hình 4. 23. Màn hình quản lý kinh nghiệm làm việc	65
Hình 4. 24. Màn hình quản lý lương	65
Hình 4. 25. Màn hình quản lý thông tin ngành nghề.....	65
Hình 4. 26. Màn hình quản lý khu vực.....	66
Hình 4. 27. Màn hình quản lý thông tin nhà tuyển dụng.....	66
Hình 4. 28. Màn hình quản lý bài đăng tin tức.....	66

CHƯƠNG 1

TỔNG QUAN ĐỀ TÀI VÀ KHẢO SÁT THỰC TẾ

1.1. Khảo sát thực tế

1.1.1. Bài toán

Trong quá trình nghiên cứu và tìm hiểu về thị trường việc làm và nhu cầu của sinh viên mới ra trường, em nhận thấy rằng việc tìm kiếm việc làm thường gặp phải nhiều khó khăn đối với người lao động. Các nền tảng truyền thông truyền thống thường không đáp ứng được nhu cầu đa dạng và thường xuyên thay đổi của người lao động.

Từ những thách thức đó, em đã quyết định chọn đề tài "*Xây dựng ứng dụng hỗ trợ tìm kiếm việc làm trên nền tảng ASP.NET CORE WEB API và VueJS.*" Với mong muốn tạo ra một nền tảng trực tuyến, em hy vọng sẽ giúp người dùng dễ dàng tìm kiếm các cơ hội việc làm phù hợp với kỹ năng và mong muốn của mình.

Ứng dụng trang web sẽ cung cấp các tính năng như đăng tin tuyển dụng, tìm kiếm việc làm theo địa điểm, lĩnh vực, mức lương và các yêu cầu khác. Đồng thời, người dùng cũng có thể tạo hồ sơ cá nhân để tiện lợi trong việc quản lý và ứng tuyển vào các công việc phù hợp.

Hy vọng rằng đề tài này sẽ đem lại giá trị lớn cho cả sinh viên mới ra trường và nhà tuyển dụng, từ đó góp phần nâng cao hiệu quả trong quá trình tìm kiếm và kết nối giữa sinh viên với các nhà tuyển dụng từ đó có cơ hội việc làm trên thị trường.

1.1.2. Thực trạng

Thực trạng hiện tại là việc phát triển một ứng dụng hỗ trợ tìm kiếm việc làm trên nền tảng ASP.NET Core Web API và VueJS đang trở thành một nhu cầu cấp thiết. Điều này phản ánh nhu cầu ngày càng tăng của người dùng trong việc tìm kiếm cơ hội việc làm một cách thuận tiện và hiệu quả.

Việc xây dựng một ứng dụng như vậy sẽ giúp cung cấp cho người dùng một cách tiếp cận dễ dàng và linh hoạt đến thông tin về các cơ hội việc làm, từ đó giúp họ nắm bắt được những cơ hội phù hợp với nhu cầu và kỹ năng của mình. Ngoài ra, việc sử dụng công nghệ ASP.NET Core Web API và VueJS cũng mang lại lợi ích về hiệu suất và trải nghiệm người dùng, từ đó tạo ra một môi trường tìm kiếm việc làm mạnh mẽ và tiện lợi.

Đồng thời, việc áp dụng công nghệ hiện đại như ASP.NET Core Web API và VueJS cũng đồng nghĩa với việc tận dụng được những tính năng và khả năng mở rộng của các nền tảng này, giúp cho việc phát triển và quản lý ứng dụng trở nên linh hoạt và dễ dàng hơn.

1.2. Hướng giải quyết

Để giải quyết bài toán xây dựng ứng dụng hỗ trợ tìm kiếm việc làm trên nền tảng ASP.NET Core Web API và VueJS, có thể áp dụng các hướng giải quyết sau:

Thiết kế cơ sở dữ liệu linh hoạt: Xây dựng cơ sở dữ liệu có cấu trúc linh hoạt và phản ánh đầy đủ thông tin về các công việc và ứng viên. Sử dụng các kỹ thuật như chuẩn hóa dữ liệu để đảm bảo tính nhất quán và dễ quản lý.

Phát triển API backend ổn định: Xây dựng các API backend sử dụng ASP.NET Core Web API để cung cấp dữ liệu cho ứng dụng front-end. API này cần được thiết kế để hỗ trợ các chức năng tìm kiếm, lọc và sắp xếp việc làm một cách hiệu quả.

Phát triển giao diện người dùng hiện đại: Sử dụng VueJS để phát triển giao diện người dùng tương tác một cách mạnh mẽ và trực quan. Đảm bảo rằng giao diện người dùng thân thiện và dễ sử dụng, với khả năng tùy chỉnh và tương thích trên nhiều thiết bị.

Đảm bảo tính bảo mật: Áp dụng các biện pháp bảo mật như xác thực người dùng, mã hóa dữ liệu và bảo vệ chống lại các cuộc tấn công để đảm bảo tính an toàn cho thông tin cá nhân và tài khoản người dùng.

1.3. Mục tiêu nghiên cứu

Mục tiêu của nghiên cứu này là xây dựng và triển khai một ứng dụng web hỗ trợ tìm kiếm việc làm trên nền tảng ASP.NET Core Web API và VueJS. Cụ thể, các mục tiêu nghiên cứu bao gồm:

Phát triển API Backend ổn định: Xây dựng và triển khai các API back-end sử dụng ASP.NET Core Web API để cung cấp dữ liệu về việc làm cho ứng dụng front-end. API này cần được thiết kế để hỗ trợ các chức năng tìm kiếm, lọc và sắp xếp các công việc một cách hiệu quả.

Thiết kế giao diện người dùng thân thiện: Phát triển giao diện người dùng bằng VueJS, với mục đích tạo ra một trải nghiệm tìm kiếm việc làm thuận tiện và hấp dẫn. Giao diện cần có tính tương tác cao và phản ánh đầy đủ thông tin về các cơ hội việc làm.

Tích hợp tính năng tìm kiếm nâng cao: Tích hợp các tính năng tìm kiếm nâng cao như tìm kiếm theo từ khoá, bộ lọc theo điều kiện và sắp xếp kết quả. Đảm bảo rằng người dùng có thể dễ dàng tìm thấy các công việc phù hợp với nhu cầu và kỹ năng của mình.

Đảm bảo tính bảo mật: Áp dụng các biện pháp bảo mật để đảm bảo tính an toàn cho thông tin cá nhân và tài khoản người dùng. Bao gồm xác thực người dùng, mã hóa dữ liệu và bảo vệ chống lại các cuộc tấn công.

1.4. Đối tượng nghiên cứu

Sinh viên và người mới ra trường đang tìm kiếm việc làm: Đối với nhóm này, ứng dụng sẽ cung cấp một nền tảng để tìm kiếm và nộp đơn cho các cơ hội việc làm phù hợp với nhu cầu và kỹ năng của họ. Các tính năng như tìm kiếm nâng cao, thông tin chi tiết về công việc và giao diện người dùng thân thiện sẽ giúp họ dễ dàng tiếp cận và tương tác với các cơ hội việc làm.

Nhà tuyển dụng và doanh nghiệp: Đối với nhóm này, ứng dụng sẽ cung cấp một nền tảng để đăng tải và quản lý các cơ hội việc làm, tiếp cận và tương tác với ứng viên tiềm năng. Các tính năng như quản lý danh sách ứng viên, gửi thông báo về việc làm và thống kê doanh số sẽ giúp họ quản lý quy trình tuyển dụng một cách hiệu quả.

1.5. Cơ sở khoa học

Đề này đang yêu cầu một cái nhìn sâu hơn vào cơ sở khoa học, những nguyên lý và phương pháp được áp dụng trong quá trình phát triển ứng dụng hỗ trợ tìm kiếm việc làm trên nền tảng ASP.NET Core Web API và VueJS. Điều này bao gồm:

Nguyên lý Thiết kế Hướng dẫn Đối tượng (OOP): Sử dụng nguyên lý OOP để phân chia chức năng và logic của ứng dụng thành các đối tượng riêng biệt, từ đó tạo ra mã nguồn dễ đọc, dễ hiểu và dễ bảo trì.

Kiến thức về Cơ sở dữ liệu: Hiểu biết sâu sắc về cơ sở dữ liệu, bao gồm các khái niệm về mô hình dữ liệu, quan hệ giữa các bảng và việc tối ưu hóa câu truy vấn SQL.

Kỹ thuật Xây dựng API RESTful: Áp dụng kiến thức về việc thiết kế và triển khai API RESTful để cung cấp các dịch vụ web cho ứng dụng frontend, bao gồm quản lý tài nguyên và xác thực người dùng.

Nguyên lý Thiết kế Giao diện người dùng (UI/UX): Hiểu biết về cách thiết kế giao diện người dùng để tạo ra trải nghiệm tốt nhất cho người dùng, bao gồm việc sắp xếp các phần tử trên giao diện, lựa chọn màu sắc và kiểu chữ.

An ninh và Bảo mật Thông tin: Kiến thức về cách bảo vệ ứng dụng khỏi các mối đe dọa an ninh mạng, bao gồm xác thực và ủy quyền người dùng, mã hóa dữ liệu và bảo vệ chống lại các cuộc tấn công.

1.6. Phương pháp nghiên cứu

Phương pháp nghiên cứu sẽ bao gồm:

- Phân tích yêu cầu của người dùng và doanh nghiệp.
- Thiết kế cơ sở dữ liệu để lưu trữ thông tin về việc làm và người dùng.
- Phát triển API backend bằng Asp.Net Core để cung cấp các dịch vụ như tìm kiếm việc làm, đăng tin tuyển dụng, và tương tác với cơ sở dữ liệu.
- Xây dựng giao diện người dùng sử dụng VueJS để cung cấp trải nghiệm tốt nhất cho người dùng khi sử dụng ứng dụng.
- Kiểm thử và triển khai ứng dụng để đảm bảo tính ổn định và hiệu quả.

1.7. Ứng dụng của kết quả nghiên cứu trong đề tài

- **Nền tảng tìm kiếm việc làm:** Ứng dụng có thể được triển khai như một nền tảng trực tuyến cho phép người tìm việc dễ dàng tìm kiếm, xem và ứng tuyển vào các vị trí việc làm phù hợp với kỹ năng và mong muốn của họ.
- **Hệ thống quản lý tuyển dụng:** Các doanh nghiệp và tổ chức có thể sử dụng ứng dụng để đăng tải thông tin về các vị trí việc làm, quản lý danh sách ứng viên, lên lịch phỏng vấn và gửi thông báo về quá trình tuyển dụng.
- **Cổng thông tin việc làm của trường đại học:** Các trường đại học có thể sử dụng ứng dụng để cung cấp thông tin về cơ hội việc làm cho sinh viên và cựu sinh viên, cũng như tạo ra các dịch vụ hỗ trợ tìm kiếm việc làm sau khi tốt nghiệp.
- **Hệ thống quản lý nhân sự:** Các doanh nghiệp có thể tích hợp ứng dụng vào hệ thống quản lý nhân sự của mình để tìm kiếm và tuyển dụng nhân sự mới, cũng như quản lý thông tin về nhân viên hiện tại.

- **Nền tảng hợp tác giữa doanh nghiệp và trường đại học:** Ứng dụng có thể được sử dụng để tạo ra các cơ hội hợp tác giữa doanh nghiệp và trường đại học trong việc tuyển dụng, đào tạo và phát triển nhân sự.

CHƯƠNG 2

CÁC KIẾN THỨC NỀN TẢNG

2.1. VueJS

2.1.1. VueJS là gì?

VueJS là một framework mã nguồn mở của JavaScript được sử dụng để phát triển các giao diện web tương tác. Một trong những framework nổi tiếng được sử dụng để đơn giản hóa việc phát triển web. VueJS tập trung vào view layer, có thể dễ dàng tích hợp vào các dự án lớn để phát triển front-end mà không gặp bất kỳ sự cố nào.

Việc cài đặt VueJS rất dễ dàng, bất kỳ nhà phát triển nào cũng có thể hiểu và xây dựng giao diện web tương tác trong thời gian ngắn. VueJS được tạo ra bởi Evan You, một cựu nhân viên và lập trình viên của Google. Phiên bản đầu tiên được phát hành vào tháng 2 năm 2014. Gần đây, framework này đã đạt 64.828 sao trên github, khiến VueJS trở nên rất phổ biến [1].

2.1.2. Ưu điểm và nhược điểm.

Giống như mọi loại công nghệ, VueJS gây ra nhiều tranh cãi trong cộng đồng. Vậy ưu và nhược điểm của VueJS là gì? Trước tiên, hãy cùng glints tìm hiểu ưu điểm của framework này.

a. Ưu điểm

Kích thước nhỏ: Tập zip được tải xuống của framework này chỉ nặng 18 KB. Điều này khiến nó không chỉ cài đặt nhanh mà còn tác động tích cực đến SEO và UX của bạn.

Kết xuất và hiệu suất DOM ảo: Mô hình đối tượng tài liệu (DOM) là thứ bạn có thể gặp phải khi kết xuất các trang web. DOM đại diện cho một trang HTML với các kiểu, thành phần và nội dung dưới dạng cấu trúc cây của các đối tượng (nút). Các đối tượng cây DOM lưu trữ dưới dạng cây và được tạo bởi trình duyệt khi tải trang.

Hệ thống phản ứng và các tùy chọn ràng buộc dữ liệu: Liên kết dữ liệu là kết nối giữa mô hình dữ liệu (nguồn dữ liệu) và mẫu DOM hoặc HTML của chế độ xem. Liên kết dữ liệu một chiều cho phép thông tin truyền theo một hướng, từ mô hình sang chế độ xem hoặc ngược lại. Trong trường hợp đầu tiên, các thay đổi đối với nguồn sẽ tự động cập nhật

DOM, nhưng nó không hoạt động ngược lại vì DOM có quyền truy cập chỉ đọc vào mô hình.

b. Nhược điểm

Rào cản ngôn ngữ: Việc áp dụng VueJS của các doanh nghiệp như Xiaomi và Alibaba đã giúp phổ biến framework và tạo ra nhu cầu trên thị trường lao động. Với việc ngày càng được phổ biến ở Trung Quốc, một phần quan trọng trong nội dung và các cuộc thảo luận đều là tiếng Trung.

Bức tường lửa gắt gao của Trung Quốc khiến mọi thứ ở quốc gia này hơi khác một chút vì rất nhiều tài nguyên phổ biến hiện không có sẵn ở đó. Điều đó làm cho việc học và sử dụng React hoặc Angular đối với các nhà phát triển địa phương trở nên khó khăn hơn. Chính vì vậy, đối với các nhà phát triển ở Trung Quốc, VueJS là một biến thể thích hợp hơn.

Vì vậy, khi tìm kiếm nội dung về VueJS, chúng ta sẽ phải đối mặt với các cuộc thảo luận diễn đàn, mô tả plugin và hướng dẫn bằng tiếng Trung.

Thiếu hỗ trợ cho các dự án quy mô lớn: Quy mô nhóm phát triển và cộng đồng của VueJS vẫn không thể so sánh với Angular hay React. Framework này cũng không được hỗ trợ tài chính từ các doanh nghiệp lớn. Để được áp dụng trong các dự án quy mô lớn, công nghệ phải ổn định và được hỗ trợ mạnh mẽ để các vấn đề có thể được giải quyết nhanh chóng. Mặc dù VueJS không gặp nhiều vấn đề và thậm chí còn có nhu cầu đến từ các doanh nghiệp như IBM và Adobe, nhưng vẫn được sử dụng trong các dự án tương đối nhỏ.

Nguy cơ đến từ việc quá linh hoạt: Tính linh hoạt là một đặc tính gây tranh cãi của một dự án lớn. Cung cấp cho nhóm phát triển của bạn quá nhiều tùy chọn có thể dẫn đến các cách tiếp cận lập trình khác nhau trong một nhóm. Và kết quả là, đã trở thành một công cụ vô hiệu hóa cuối cùng thay vì một phần mềm hoạt động.

Nguồn tài nguyên giới hạn: Mặc dù hệ sinh thái khá rộng và có tất cả các công cụ cần thiết để bắt đầu phát triển với VueJS, nhưng framework này không lớn bằng React hay Angular. Nói chính xác hơn, chỉ cần so sánh số lượng plugin có sẵn cho React và VueJS, sự khác biệt là ở hàng trăm đơn vị. Các plugin hiện có có thể được sử dụng với các framework khác cũng thường không được hỗ trợ.

2.1.3. Ứng dụng của VueJS

Xử lý các nguyên mẫu: Trước hết, VueJS được thiết kế để tạo nguyên mẫu. Với liên kết dữ liệu phù hợp, thật tuyệt vời khi framework này còn có thể xử lý nhiều hoạt ảnh, yếu tố tương tác và đồ họa. Tìm hiểu giao diện người dùng của bạn, cài đặt Vue CLI và bạn có thể sử dụng các nguyên mẫu có thể nháp được.

Giữ sự tập trung vào giao diện người dùng: Tập trung vững chắc vào UI, vì nó chỉ yêu cầu HTML, CSS và JS để hoạt động mà không cần quá nhiều thứ dành riêng cho Vue. Ví dụ: IBM đã sử dụng VueJS làm khung giao diện người dùng cho Đám mây lai của mình, vì đường cong học tập nhẹ nhàng và sự phụ thuộc vào HTML, CSS và JS.

Nhu cầu hội nhập: Nếu bạn có một ứng dụng và bạn muốn thêm một số tính tương tác vào ứng dụng đó, VueJS có thể trợ giúp điều này. Vì được dựa trên JavaScript nên có thể dễ dàng tích hợp vào bất kỳ dự án. Hơn nữa, tương thích với nhiều công nghệ back-end và framework như Laravel, Express, Rails và Django. Ngay cả khi xem xét nhược điểm của VueJS, vẫn có thể được sử dụng trong các dự án lớn. Trong số những người chơi toàn cầu sử dụng VueJS để xây dựng trang web của họ có Grammarly, Upwork, Gitlab, Trivago, Nintendo và thậm chí cả Google. Danh sách này còn lâu mới hoàn thành và rõ ràng là sẽ được tiếp tục.

2.1.4. Thư viện Axios.

Axios là một thư viện HTTP client phổ biến và mạnh mẽ được sử dụng trong VueJS để thực hiện các yêu cầu HTTP từ phía client tới server. Với Axios, ta có thể dễ dàng gửi và nhận các yêu cầu HTTP như GET, POST, PUT và DELETE và xử lý các phản hồi từ server.

Với cú pháp đơn giản và dễ sử dụng, Axios giúp giảm bớt công sức và thời gian khi làm việc với yêu cầu HTTP. Ta có thể xác định các cấu hình mặc định cho Axios như URL gốc, header, timeout và xử lý lỗi. Ngoài ra, Axios hỗ trợ gửi các yêu cầu với các dữ liệu như JSON, FormData và cả các yêu cầu tùy chỉnh khác.

Thư viện Axios cũng hỗ trợ các tính năng quan trọng như hủy yêu cầu, theo dõi tiến trình yêu cầu và xử lý lỗi một cách dễ dàng. Cung cấp các phương thức cho phép bạn kiểm tra và xử lý các phản hồi từ server, như xem thông tin header, status code và dữ liệu trả về.

Với việc tích hợp tốt vào VueJS, Axios cho phép ta thực hiện các yêu cầu HTTP từ các thành phần Vue một cách dễ dàng và linh hoạt. Ta có thể sử dụng Axios để tải dữ liệu từ server khi ứng dụng được tạo, gửi dữ liệu biểu mẫu lên server hoặc tương tác với các API bên ngoài.

Với tính năng và khả năng linh hoạt mạnh mẽ, Axios là một thư viện không thể thiếu khi phát triển ứng dụng VueJS. Giúp đơn giản hóa việc làm việc với yêu cầu HTTP và cung cấp một cách tiếp cận dễ dàng và đáng tin cậy cho việc giao tiếp với server.

2.1.5. Vue router

Vue router là router chính thức cho VueJS. Các tính năng bao gồm:

- Nested route (định tuyến lồng nhau)/view mapping
- Cấu hình router dựa trên thành phần, mô-đun
- Định tuyến tham số, truy vấn, ký tự đại diện
- Xem các hiệu ứng chuyển tiếp được cung cấp bởi hệ thống chuyển tiếp của VueJS
- Kiểm soát tính năng Fine-grained navigation
- Liên kết với các active CSS class tự động
- Chế độ HTML5 history mode hay hash mode, với tính năng auto-fallback trong IE9
- Kiểm soát chính xác vị trí cuộn mỗi trang

2.2. Typescript

TypeScript là một ngôn ngữ lập trình được phát triển và duy trì bởi Microsoft. Một siêu tập hợp cú pháp nghiêm ngặt của JavaScript và thêm tính năng kiểu tĩnh tùy chọn vào ngôn ngữ. TypeScript được thiết kế để phát triển các ứng dụng lớn và biên dịch sang JavaScript. Vì TypeScript là một siêu tập hợp của JavaScript, các chương trình JavaScript hiện có cũng là các chương trình TypeScript hợp lệ về mặt cú pháp, tuy nhiên, chúng có thể bị lỗi type-check.

TypeScript có thể được sử dụng để phát triển các ứng dụng JavaScript cho cả thực thi phía máy khách và phía máy chủ (như với Node.js hoặc Deno). Có nhiều tùy chọn có sẵn để biên dịch. Có thể sử dụng Trình biên dịch TypeScript mặc định, hoặc có thể sử dụng trình biên dịch Babel để chuyển đổi TypeScript sang JavaScript.

TypeScript hỗ trợ các tệp định nghĩa có thể chứa thông tin kiểu của các thư viện JavaScript hiện có, giống như tệp tiêu đề C++ có thể mô tả cấu trúc của tệp đối tượng hiện

có. Điều này cho phép các chương trình khác sử dụng các giá trị được xác định trong tệp như thể chúng là các thực thể TypeScript được nhập tĩnh. Có các tệp tiêu đề của bên thứ ba cho các thư viện phổ biến như jQuery, MongoDB và D3.js. Tiêu đề TypeScript cho các mô-đun cơ bản của Node.js cũng có sẵn, cho phép phát triển các chương trình Node.js trong TypeScript.

Bản thân trình biên dịch TypeScript được viết bằng TypeScript và được biên dịch sang JavaScript. Được cấp phép theo giấy phép Apache 2.0, bao gồm như một ngôn ngữ lập trình hạng nhất trong Microsoft Visual Studio 2013 update 2 trở lên, bên cạnh C # và các ngôn ngữ Microsoft khác. Một phần mở rộng chính thức cho phép Visual Studio 2012 cũng hỗ trợ TypeScript. Anders Hejlsberg, kiến trúc sư chính của C # và người tạo ra Delphi và Turbo Pascal, đã tham gia vào quá trình phát triển TypeScript.

TypeScript được công bố lần đầu tiên vào tháng 10 năm 2012 (ở phiên bản 0.8), sau hai năm phát triển nội bộ tại Microsoft. Ngay sau khi thông báo, Miguel de Icaza đã ca ngợi bản thân ngôn ngữ này, nhưng chỉ trích việc thiếu hỗ trợ IDE trưởng thành ngoài Microsoft Visual Studio, vốn chưa có trên Linux và OSX vào thời điểm đó. Ngày nay, có hỗ trợ trong các IDE khác, đặc biệt là trong Eclipse, thông qua một trình cắm thêm do Palantir Technologies đóng góp. Các trình soạn thảo văn bản khác nhau, bao gồm Emacs, Vim, Webstorm, Atom và Visual Studio Code của riêng Microsoft cũng hỗ trợ TypeScript. TypeScript 0.9, được phát hành vào năm 2013, đã thêm hỗ trợ cho generic.

TypeScript 1.0 được phát hành tại hội nghị nhà phát triển Build của Microsoft vào năm 2014. Visual Studio 2013 Update 2 cung cấp hỗ trợ tích hợp cho TypeScript. Vào tháng 7 năm 2014, nhóm phát triển đã công bố một trình biên dịch TypeScript mới, đạt hiệu suất gấp 5 lần. Đồng thời, mã nguồn, ban đầu được lưu trữ trên CodePlex, đã được chuyển sang GitHub.

Vào ngày 22 tháng 9 năm 2016, TypeScript 2.0 đã được phát hành; được giới thiệu một số tính năng, bao gồm khả năng cho các lập trình viên tùy ý ngăn các biến được gán giá trị null, đôi khi được gọi là sai lầm hàng tỷ đô la.

TypeScript 3.0 được phát hành vào ngày 30 tháng 7 năm 2018, mang đến nhiều bổ sung ngôn ngữ như tuple (bộ giá trị) trong rest parameter (tham số phần còn lại) và spread expression (biểu thức trải rộng), rest parameter với các kiểu tuple, generic rest parameter...

TypeScript 4.0 được phát hành vào ngày 20 tháng 8 năm 2020. Mặc dù 4.0 không giới thiệu bất kỳ thay đổi đột phá nào, nhưng đã bổ sung các tính năng ngôn ngữ như hệ thống JSX tùy chỉnh và các kiểu Variadic Tuple (bộ giá trị đa dạng).

TypeScript bắt nguồn từ những thiếu sót của JavaScript đối với việc phát triển các ứng dụng quy mô lớn ở cả Microsoft và các khách hàng bên ngoài của công ty này. Những thách thức đối với việc xử lý mã JavaScript phức tạp đã dẫn đến nhu cầu về công cụ tùy chỉnh để dễ dàng phát triển các thành phần trong ngôn ngữ này.

Các nhà phát triển TypeScript đã tìm kiếm một giải pháp không phá vỡ tính tương thích với tiêu chuẩn và khả năng hỗ trợ đa nền tảng. Biết rằng đề xuất tiêu chuẩn ECMAScript hiện tại hứa hẹn hỗ trợ cho lập trình dựa trên lớp trong tương lai, TypeScript đã được xây dựng dựa trên đề xuất đó. Điều này dẫn đến một trình biên dịch JavaScript với một tập hợp các phần mở rộng ngôn ngữ cú pháp, một siêu tập hợp dựa trên đề xuất ECMAScript, chuyển đổi các phần mở rộng đó thành JavaScript thông thường. Theo nghĩa này, tính năng lớp của TypeScript là bản xem trước những gì mong đợi của ECMAScript 2015. Một khía cạnh duy nhất không có trong đề xuất, nhưng được thêm vào TypeScript, là tính năng kiểu tĩnh tùy chọn, cho phép phân tích ngôn ngữ tĩnh, tạo điều kiện hỗ trợ công cụ và IDE [5].

2.2.1. Interfaces

Trong TypeScript, sử dụng interface là cách khai báo trước cấu trúc cần phải có cho các class, các function, các biến... TypeScript không biên dịch interface sang mã JavaScript. Typescript chỉ dùng interface cho mục đích hỗ trợ kiểm tra kiểu và cấu trúc của các biến, function, class ... sử dụng theo interface được chính xác.

Mỗi interface được định nghĩa với từ khóa interface. Khi định nghĩa, bạn khai báo các property và method. Method thì có thể dùng kiểu function thông thường hoặc arrow function đều được. Trong TypeScript, interface có thể dùng để định nghĩa 1 kiểu dữ liệu (type) mới và cũng có thể implement trong 1 class. Sử dụng interface trong TypeScript khá đa dạng.

Bạn có thể dùng interface như là 1 kiểu hàm. Có thể sử dụng interface để tạo kiểu array. Tức dùng interface để khai báo kiểu của các giá trị trong mảng và kiểu cho index của mảng. Trong interface có thể chỉ định một thuộc tính nào đó là tùy chọn, bằng cách

dùng dấu ? Khi đó các đối tượng theo interface có thể định nghĩa thuộc tính đó hoặc không. Trong interface có thể đánh dấu 1 thuộc tính nào đó là read only. Nghĩa là chỉ có thể đọc giá trị của thuộc tính mà không thể gán giá trị cho nó. Mỗi interface có thể extend 1 hoặc nhiều interface khác.

Interface trong TypeScript cũng có thể implement cho class. Class nào được implement với interface thì phải khai báo đủ các hàm và thuộc tính trong interface. Ngoài ra có thể định nghĩa thêm các thuộc tính khác tùy ý. Sử dụng interface trong typescript rất linh động phải không. Đây là công cụ mạnh giúp bạn code javascript ngon lành, nhất là trong các dự án nghiệp vụ phức tạp.

2.2.2. Classes

Trong các dự án, ứng dụng lớn, lập trình hướng đối tượng rất hay được các Dev sử dụng nhất là trong các ngôn ngữ như Java hoặc C#.

TypeScript cung cấp hệ thống class khá tương đồng với các ngôn ngữ này ví dụ như chức năng kế thừa, abstract classes, interface implementations, setter/getters, ...

Từ phiên bản ECMAScript 2015 trở đi, classes được xem là một tính năng có sẵn trong JS và có thể không cần sử dụng TypeScript. Mặc dù 2 phiên bản này có nhiều chức năng tương tự nhau, nhưng chúng vẫn có điểm khác biệt, đó là TypeScript nghiêm ngặt hơn.

2.2.3. Modules

Kể từ ES6, JavaScript bắt đầu hỗ trợ các modules như một phần của ngôn ngữ. TypeScript chia sẻ các khái niệm module giống với JavaScript. Một module trong TypeScript có chứa cả khai báo và code. Một module thực thi trong phạm vi riêng, không phải trong phạm vi toàn cục. Điều đó có nghĩa là khi bạn khai báo các biến, hàm, lớp, giao diện, v.v., trong một module, chúng không hiển thị bên ngoài module, trừ khi bạn xuất chúng một cách rõ ràng bằng cách sử dụng câu lệnh export. Trong module này, chúng ta đặt từ khóa export trước từ khóa interface để các modules có thể sử dụng nó. Nếu bạn không sử dụng từ khóa export, thì interface Validator sẽ là kiểu private trong Validator.ts, do đó, bạn không thể sử dụng từ các modules khác.

Có thể đưa các thành phần được export trong module vào file khác để dùng. Có thể import 1 thành phần cụ thể nào đó trong module khác hoặc import toàn bộ những gì được

module khác export cũng được. Trong 1 module, có thể export tiếp tục 1 vài thành phần đã import hoặc export toàn bộ từ module đã import, khi export có thể đặt tên mới cho 1 thành phần. Trong JavaScript ES6, từ khoá module được sử dụng để định nghĩa một module, điều này cho phép bạn định nghĩa nhiều module trong một file. TypeScript sử dụng từ khoá namespace với cú pháp và mục đích tương tự

2.2.4. Ưu điểm của Typescript

a. Ưu điểm

- TypeScript thật sự thuận tiện và hoàn toàn miễn phí
- Thao tác nhanh chóng và đơn giản hơn
- Tái cấu trúc
- Giảm tỷ lệ mắc lỗi trong hệ thống
- Hạn chế thử nghiệm Boilerplate
- Hợp nhất mã đơn giản
- Hỗ trợ tối ưu hóa quy trình làm việc

b. Nhược điểm

- Bắt buộc dùng biên dịch
- Bước thiết lập công kênh
- Chỉ là phần ngôn ngữ mở rộng hỗ trợ

2.3. Asp.Net Core Web API

2.3.1. Asp.Net Core Web API là gì ?

ASP.NET Web API là một khung dùng để xây dựng và lập trình các dịch vụ web HTTP. Có dạng là một RESTful API hiện đại, hội tụ đủ các điều kiện của REST cũng như các tiêu chuẩn tương tự, được tối ưu cho các dịch vụ trực tuyến cũng như ứng dụng web hiện nay [2].

2.3.2. Ưu điểm và nhược điểm

a. Ưu điểm

- Hỗ trợ RESTful: ASP.NET Web API được thiết kế để xây dựng các dịch vụ RESTful một cách dễ dàng.

- Dựa trên ASP.NET: ASP.NET Web API được xây dựng trên nền tảng ASP.NET và hỗ trợ đường dẫn yêu cầu / phản hồi ASP.NET.
- Ánh xạ động từ HTTP: ASP.NET Web API ánh xạ các động từ HTTP thành tên phương thức.
- Hỗ trợ định dạng dữ liệu đa dạng: ASP.NET Web API hỗ trợ nhiều định dạng dữ liệu khác nhau như JSON, XML, CSV, v.v.
- Có thể mở rộng: ASP.NET Web API có thể được mở rộng bằng cách sử dụng các thư viện và framework khác nhau.
- Hiệu suất cao: ASP.NET Web API được tối ưu hóa cho hiệu suất cao.
- Mã nguồn mở: ASP.NET Web API là mã nguồn mở và miễn phí sử dụng.
- Cộng đồng lớn: ASP.NET Web API có một cộng đồng lớn và hỗ trợ tốt.

b. Nhược điểm

- Học tập: ASP.NET Web API có thể khó học đối với người mới bắt đầu.
- Khả năng bảo mật: ASP.NET Web API có thể gặp một số vấn đề về bảo mật nếu không được cấu hình đúng cách.
- Khả năng tương thích: ASP.NET Web API có thể không tương thích với một số trình duyệt cũ.

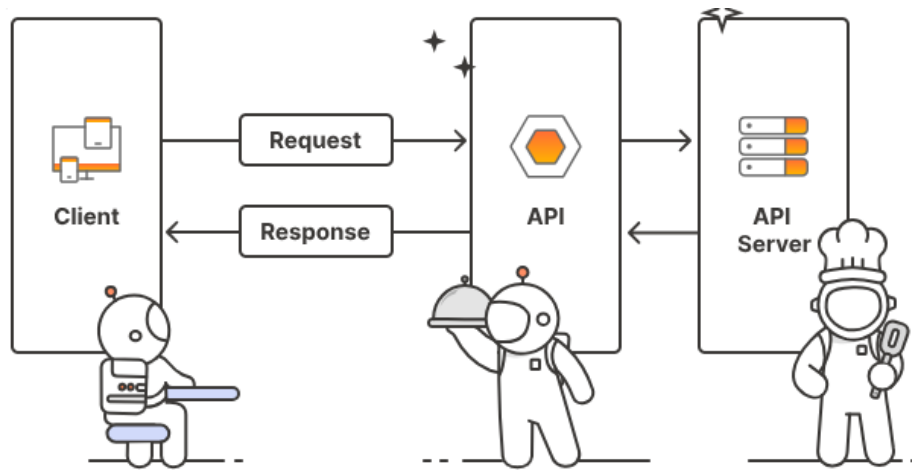
2.2.3. Ứng dụng

- Xây dựng các dịch vụ web RESTful
- Tạo API cho các ứng dụng di động
- Xây dựng các dịch vụ backend cho các ứng dụng web
- Tích hợp với các hệ thống khác

2.4. API (Application Programming Interface)

2.4.1. Ứng dụng

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng [3].



Hình 2. 1. API (Application Programming Interface)

2.4.2. Ứng dụng của API.

Web API: là hệ thống API được sử dụng trong các hệ thống website. Hầu hết các website đều ứng dụng đến Web API cho phép bạn kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu. Ví dụ: Bạn thiết kế chức năng login thông Google, Facebook, Twitter, Github... Điều này có nghĩa là bạn đang gọi đến API của. Hoặc như các ứng dụng di động đều lấy dữ liệu thông qua API.

API trên hệ điều hành: Windows hay Linux có rất nhiều API, họ cung cấp các tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức kết nối. Sẽ giúp lập trình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành.

API của thư viện phần mềm hay framework: API mô tả và quy định các hành động mong muốn mà các thư viện cung cấp. Một API có thể có nhiều cách triển khai khác nhau và nó cũng giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng thư viện được viết bằng ngôn ngữ khác. Ví dụ bạn có thể dùng Php để yêu cầu một thư viện tạo file PDF được viết bằng C++.

Web API hỗ trợ restful đầy đủ các phương thức: Get/Post/Put/Delete dữ liệu, giúp bạn xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Và cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content format.

Tự động hóa sản phẩm: Với web API, chúng ta sẽ tự động hóa quản lý công việc, cập nhật luồng công việc, giúp tăng năng suất và tạo hiệu quả công việc cao hơn.

Khả năng tích hợp linh động: API cho phép lấy nội dung từ bất kỳ website hoặc ứng dụng nào một cách dễ dàng nếu được cho phép, tăng trải nghiệm người dùng. API hoạt động như một chiếc cổng, cho phép các công ty chia sẻ thông tin được chọn nhưng vẫn tránh được những yêu cầu không mong muốn.

Cập nhật thông tin thời gian thực: API có chức năng thay đổi và cập nhật thay đổi theo thời gian thực. Với công nghệ này, dữ liệu sẽ được truyền đi tốt hơn, thông tin chính xác hơn, dịch vụ cung cấp linh hoạt hơn.

Có tiêu chuẩn chung để sử dụng: Bất kỳ người dùng, công ty nào sử dụng cũng có thể điều chỉnh nội dung, dịch vụ mà họ sử dụng. Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

2.4.3. Web API hoạt động như nào ?

Đầu tiên là xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS. Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả. Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.

Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động, dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu...

2.4.4. Ưu điểm và nhược điểm

a. Ưu điểm

- Web API được sử dụng hầu hết trên các ứng dụng desktop, ứng dụng mobile và ứng dụng website.
- Linh hoạt với các định dạng dữ liệu khi trả về client: JSON, XML hay định dạng khác.
- Nhanh chóng xây dựng HTTP service: URI, request/response headers, caching, versioning, content formats và có thể host trong ứng dụng hoặc trên IIS.
- Mã nguồn mở, hỗ trợ chức năng RESTful đầy đủ, sử dụng bởi bất kì client nào hỗ trợ XML, JSON.

- Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.
- Giao tiếp hai chiều được xác nhận trong các giao dịch, đảm bảo độ tin cậy cao.

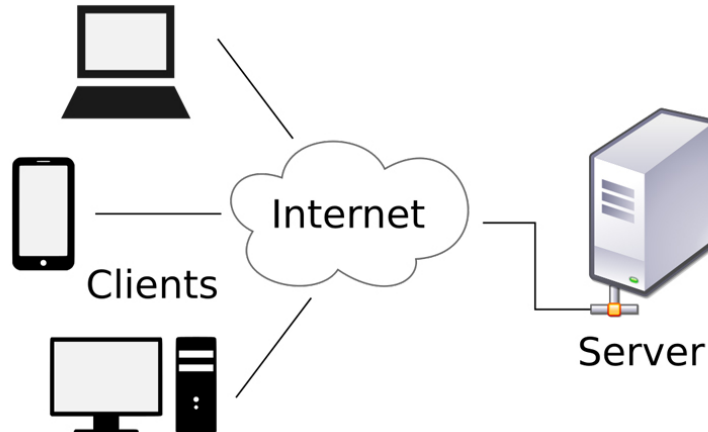
b. Nhược điểm

- Web API chưa hoàn toàn phải là RESTful service, mới chỉ hỗ trợ mặc định GET, POST
- Để sử dụng hiệu quả cần có kiến thức chuyên sâu, có kinh nghiệm backend tốt
- Tốn thời gian và chi phí cho việc phát triển, nâng cấp và vận hành

2.4.5. Ứng dụng của Web API

- Xây dựng các dịch vụ web RESTful
- Tạo API cho các ứng dụng di động
- Xây dựng các dịch vụ backend cho các ứng dụng web
- Tích hợp với các hệ thống khác

2.4.6. Mô hình client/server qua HTTP



Hình 2. 2. Mô hình Client-Server

Client/server

Phần giao diện người dùng được gọi là client. Client rất đa dạng, bao gồm ứng dụng đơn trang SPA(Single-Page-Application, chạy trình duyệt), ứng dụng desktop truyền thống, ứng dụng mobile. Thậm chí một ứng dụng web khác cũng có thể trở thành client.

Ứng dụng đơn trang(Single-Page-Application, SPA) trong những năm qua trở nên rất phổ biến. Hàng loạt framework do hãng lớn phát triển dành cho phát triển SPA client như React, Angular, VueJS.

Trong SPA, thành phần client chạy trên trình duyệt và được xây dựng hoàn toàn bằng JavaScript. Thành phần server gửi cho trình duyệt code Javascript một lần lúc trình duyệt truy xuất ứng dụng. Sau đó SPA và server chỉ còn trao đổi dữ liệu qua HTTP ở dạng JSON hoặc XML.

Ứng dụng desktop hiện nay, nhất là các ứng dụng business, cũng theo xu hướng sử dụng một server chung để lưu trữ và xử lý dữ liệu. Loại ứng dụng desktop triển khai theo mô hình này tiện lợi trong mạng cục bộ và thậm chí có thể hoạt động qua Internet.

Ứng dụng mobile cũng vô cùng phổ biến trong nhiều năm qua. Do đặc thù của thiết bị di động, việc xử lý và lưu trữ dữ của các loại ứng dụng này thường không thực hiện ngay trên thiết bị. Thay vào đó, chúng được thực hiện trên một server. Ứng dụng mobile cùng truy xuất server theo cách tương tự như ứng dụng SPA qua HTTP.

Thành phần server xây dựng bằng Asp.net core được gọi là web api. Web api chịu trách nhiệm cung cấp dữ liệu, chủ yếu ở dạng JSON và XML, cho client. Web api cung cấp một loạt URL mà client có thể sử dụng để truy xuất hoặc thay đổi dữ liệu trên server qua HTTP.

Một đặc điểm đặc biệt là cùng một Web Api có thể đồng thời sử dụng SPA, mobile, desktop và các ứng dụng khác. Điều này dẫn đến Web Api trở thành vô cùng quan trọng trong các hệ thống ứng dụng hiện đại hiện nay.

Web Api có thể triển khai song song với một ứng dụng web truyền thống hoặc có thể hoạt động hoàn toàn độc lập phục vụ các ứng dụng client khác nhau.

2.5. JWT

JWT (JSON Web Token) là một phương thức tiêu chuẩn để truyền thông tin giữa các bên dưới dạng một chuỗi JSON nhỏ gọn và tự chứa dữ liệu. Nó được sử dụng để xác thực và ủy quyền truy cập giữa các ứng dụng và hệ thống khác nhau.

JWT bao gồm ba phần chính: Header, Payload và Signature [4].

1. Header: Chứa các thông tin về loại token và thuật toán mã hóa được sử dụng để tạo chữ ký (signature) cho token. Header được mã hóa dưới dạng Base64URL.
2. Payload: Chứa các thông tin (claim) về người dùng hoặc các dữ liệu tùy ý. Payload cũng được mã hóa dưới dạng Base64URL.

3. Signature: Chữ ký (signature) được tạo bằng cách kết hợp Header, Payload và một secret key sử dụng thuật toán mã hóa đã được chỉ định trong Header. Chữ ký đảm bảo tính toàn vẹn của token và đảm bảo rằng sẽ không bị sửa đổi trong quá trình truyền tải.

Quá trình hoạt động của JWT như sau:

1. Người dùng gửi yêu cầu xác thực đến máy chủ.
2. Máy chủ xác thực xác minh thông tin đăng nhập của người dùng.
3. Máy chủ tạo JWT và gửi lại cho người dùng.
4. Người dùng gửi JWT trong mỗi yêu cầu đến máy chủ để xác thực.
5. Máy chủ xác minh tính toàn vẹn và xác thực của JWT bằng cách giải mã và kiểm tra chữ ký.
6. Nếu JWT hợp lệ, máy chủ xác thực yêu cầu và phản hồi cho người dùng.

JWT có nhiều ứng dụng rộng rãi trong việc xác thực và ủy quyền truy cập trong các ứng dụng web và dịch vụ API. Được sử dụng để truyền thông tin về người dùng, quyền truy cập, và các dữ liệu khác giữa các hệ thống một cách an toàn. JWT là một công nghệ linh hoạt và tiện dụng, giúp giảm tải cho máy chủ và hỗ trợ việc xác thực phân tán.

2.5.1. Ưu điểm và nhược điểm

Ưu điểm của JWT:

- Đơn giản và di động: JWT sử dụng định dạng JSON và có kích thước nhỏ, dễ dàng truyền tải qua mạng và hỗ trợ trên nhiều ngôn ngữ lập trình.
- Tự chứa thông tin: JWT chứa thông tin về người dùng và các claims tùy ý, giúp tránh việc phải lưu trữ trạng thái phiên (session state) trên máy chủ.
- Xác thực đáng tin cậy: JWT sử dụng chữ ký (signature) để xác thực tính toàn vẹn của token, đảm bảo rằng sẽ không bị sửa đổi trong quá trình truyền tải.
- Phân phối và mở rộng: JWT cho phép xác thực phân tán, nghĩa là mỗi hệ thống hoặc ứng dụng có thể xác thực JWT mà không cần truy cập vào cơ sở dữ liệu chung.
- Thời gian hết hạn (expiration): JWT có thể được cấu hình với thời gian hết hạn, giúp hạn chế thời gian sử dụng của token và tăng tính bảo mật.

Nhược điểm của JWT:

- Không thay đổi được thông tin: Một khi JWT đã được ký, nội dung của nó không thể thay đổi trước khi hết hạn. Điều này có thể gây rủi ro nếu thông tin trong JWT cần thay đổi ngay lập tức.
- Kích thước tăng dần: Khi chứa nhiều thông tin trong Payload, kích thước của JWT có thể tăng lên, gây tốn thêm băng thông khi truyền tải qua mạng.
- Lưu trữ trạng thái phía máy chủ: Mặc dù JWT tự chứa thông tin, máy chủ vẫn có thể cần lưu trữ một số thông tin phụ trợ để xác thực và kiểm tra tính hợp lệ của JWT.
- Không thuận tiện cho việc hủy bỏ: JWT không có cơ chế tích hợp để hủy bỏ token trước khi hết hạn. Điều này đòi hỏi các biện pháp phụ trợ để quản lý danh sách token hợp lệ và vô hiệu hóa token đã cấp.
- Tăng rủi ro nếu không được quản lý cẩn thận: Nếu JWT không được quản lý và bảo mật cẩn thận, có thể dẫn đến các vấn đề bảo mật như rò rỉ thông tin, tấn công giả mạo và truy cập trái phép.

Tuy nhiên, JWT vẫn là một công nghệ phổ biến và hữu ích trong việc xác thực và ủy quyền truy cập trong các ứng dụng web và dịch vụ API, với sự kết hợp các biện pháp bảo mật và quản lý cẩn thận.

CHƯƠNG 3. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1. Mô tả chức năng bài toán

Xây dựng ứng dụng hỗ trợ và tìm kiếm việc làm tập trung vào việc tạo ra một nền tảng hiệu quả cho người tìm việc nói chung và nhà tuyển dụng nói riêng. Dưới đây là mô tả chi tiết chức năng như sau:

3.1.1. Chức năng cho quản trị viên

- a. Quản lý banner
 - Mô tả: Quản lý banner hình ảnh cho ứng dụng.
 - Chức năng: Thêm, sửa, xóa.
- b. Quản lý tài khoản nhà tuyển dụng.
 - Mô tả: Duyệt tài khoản nhà tuyển dụng và cập nhật tài khoản đó nếu là công ty top đầu.
 - Chức năng: Thêm, Cập nhật, Xóa.
- c. Quản lý nhân viên
 - Mô tả: Quản lý và phân quyền chức năng cho các module của ứng dụng.
 - Chức năng: Thêm, sửa, xóa, phân quyền.
- d. Quản lý đăng tin tức
 - Mô tả: quản lý tin tức việc làm.
 - Chức năng: Thêm, sửa, xóa.
- e. Thống kê
 - Mô tả: Hệ thống sẽ cung cấp các chức năng như thống kê và báo cáo.
 - Chức năng: Xem thống kê tổng số nhà tuyển dụng, số việc đã có, nhân viên và mẫu cv.

3.1.2. Chức năng cho nhà tuyển dụng

- a. Đăng nhập/đăng ký
 - Mô tả: Nhà tuyển dụng sẽ đăng nhập và đăng ký tài khoản với vai trò là nhà tuyển dụng. Nếu chưa có tài khoản ứng dụng sẽ hiển thị thông báo cho nhà tuyển dụng biết và đăng ký tài khoản mới.
 - Chức năng: Đăng nhập, đăng ký tài khoản.
- b. Đăng bài tuyển dụng
 - Mô tả: nhà tuyển dụng sẽ đăng tải bài viết tuyển dụng theo yêu cầu đã đề ra và đăng tải lên ứng dụng để thu hút người ứng tuyển phù hợp
 - Chức năng: Thêm, sửa, xóa
- c. Xem danh sách người ứng tuyển
 - Mô tả: Nhà tuyển dụng sẽ xem danh sách và chọn ra một hoặc nhiều người phù hợp với công việc hiện tại. Nhà tuyển dụng sẽ đánh giá dựa trên tiêu chí "**Phù hợp với công việc**" hoặc "**Không phù hợp với công việc**". Sau khi đánh giá xong ứng dụng sẽ thông báo cho người tìm việc để họ có thể nắm bắt được thông tin.
 - Chức năng: Đánh giá mức độ phù hợp công việc.
- d. Xem thống kê
 - Mô tả: Nhà tuyển dụng sẽ xem thống kê việc tuyển dụng tháng này qua: Tổng số bài đăng, số người ứng tuyển...
 - Chức năng: Xem và thống kê công việc.
- e. Quản lý thông tin cá nhân
 - Mô tả: nhà tuyển dụng có thể quản lý thông tin cá nhân của mình, bao gồm cập nhật thông tin, đổi mật khẩu tài khoản của mình.
 - Chức năng: Thêm, sửa, xóa

3.1.3. Chức năng cho người tìm việc

- a. Đăng nhập/đăng ký
 - Mô tả: Nhà tuyển dụng sẽ đăng nhập và đăng ký tài khoản với vai trò là người tìm việc. Nếu chưa có tài khoản ứng dụng sẽ hiển thị thông báo cho và đăng ký tài khoản mới.

- Chức năng: Đăng nhập, đăng ký tài khoản.

b. Ứng tuyển công việc

- Mô tả: Người tìm việc sẽ tìm kiếm và chọn lọc công việc phù hợp với trình độ bản thân rồi tiến hành ứng tuyển công việc đó.
- Chức năng: Ứng tuyển việc làm.

c. Xem danh sách công việc đã ứng tuyển

- Mô tả: Có thể xem lại danh sách các công việc đã ứng tuyển trước đó. Từ đó rút ra kinh nghiệm thiếu sót của bản thân và rút ra kinh nghiệm để tìm công việc mới.
- Chức năng: Xem danh sách việc làm.

d. Quản lý hồ sơ xin việc

- Mô tả: Có thể tạo mới hồ sơ xin việc cho bản thân dựa và mẫu đã cho trước của hệ thống, cập nhập và xóa thông tin theo mong muốn.
- Chức năng: Tạo mới, chỉnh sửa và xóa thông tin.

e. Quản lý thông tin tài khoản

- Mô tả: Cập nhập thông tin tài khoản cá nhân để cho nhà tuyển dụng có thể xem thông tin.
- Chức năng: Cập nhập thông tin bao gồm đổi mật khẩu.

f. Chat liên hệ AI

- Mô tả: Tích hợp tính năng chat để người dùng có thể tương tác trực tiếp với trợ lý AI để nhận hỗ trợ, giải đáp thắc mắc, và thực hiện các tác vụ cần thiết.
- Chức năng:
 - Cung cấp kênh liên lạc trực tiếp với trợ lý AI thông qua chat.
 - Hỗ trợ người dùng trong việc tìm kiếm thông tin, giải quyết vấn đề, và thực hiện các nhiệm vụ.
 - Đáp ứng tự động và nhanh chóng các yêu cầu từ người dùng một cách hiệu quả.
 - Tính năng chat liên hệ với trợ lý AI giúp tạo ra một kênh giao tiếp thuận tiện và linh hoạt giữa người dùng và hệ thống, giúp họ có được sự hỗ trợ và giải đáp mọi thắc mắc một cách nhanh chóng và dễ dàng.

3.2. Yêu cầu bài toán

3.2.1. Yêu cầu chức năng

Bài toán yêu cầu xây dựng một trang web tìm kiếm và hỗ trợ việc làm cho sinh viên và người mới ra trường, sử dụng công nghệ ASP.NET Core API và VueJS. Trang web cần cung cấp các tính năng sau:

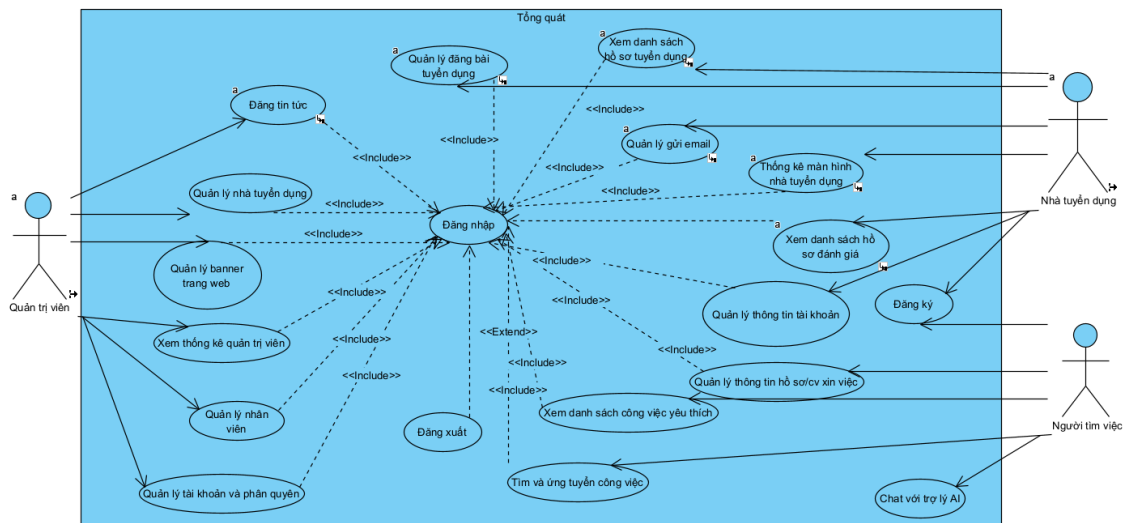
- Đăng tin tuyển dụng và quản lý tin đăng cho các doanh nghiệp.
- Tìm kiếm và lọc việc làm theo địa điểm, lĩnh vực, mức lương và các yêu cầu khác của người dùng.
- Tạo và quản lý hồ sơ cá nhân cho người dùng, bao gồm lịch sử ứng tuyển và thông tin cá nhân và CV ứng tuyển công việc.

3.2.2. Yêu cầu phi chức năng

- Hiệu suất: Trang web cần có hiệu suất cao và thời gian tải nhanh để đảm bảo trải nghiệm người dùng tốt.
- Bảo mật: Trang web cần có các biện pháp bảo mật đủ mạnh để bảo vệ thông tin cá nhân của người dùng và thông tin tuyển dụng của doanh nghiệp.
- Responsive: Giao diện của trang web cần phải responsive, tức là có thể hiển thị đúng trên các thiết bị khác nhau như máy tính, điện thoại di động và máy tính bảng.

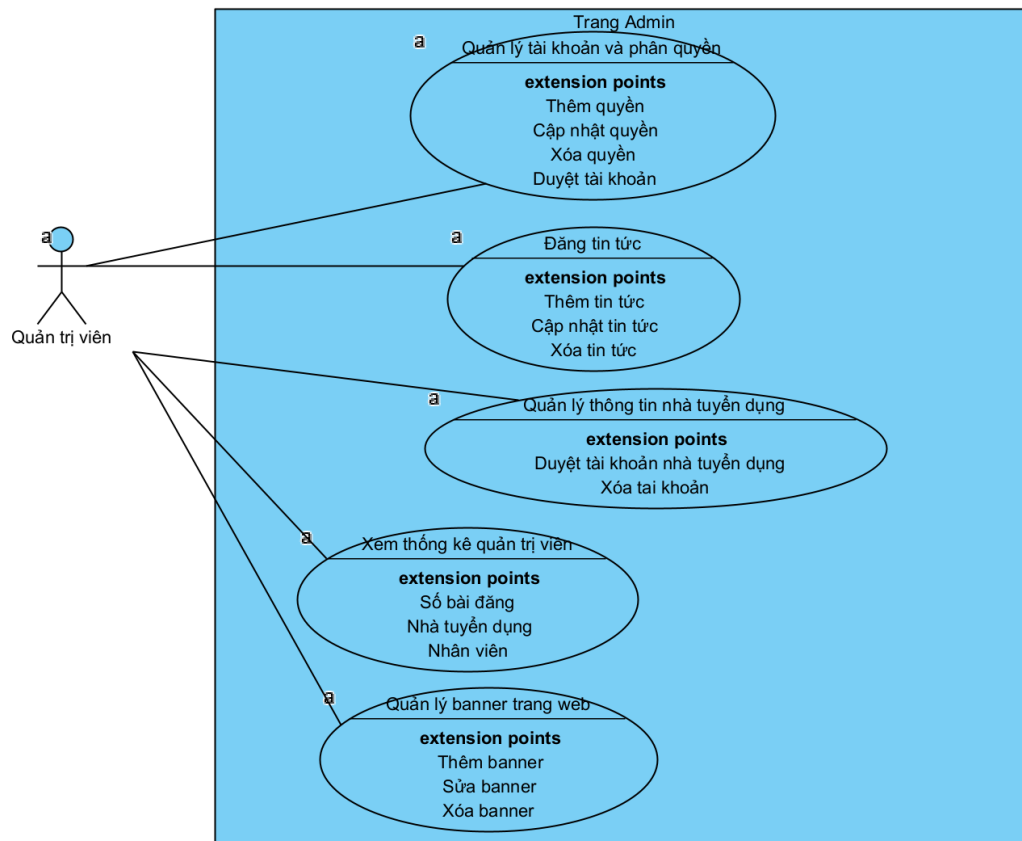
3.3. Thiết kế chức năng hệ thống

3.3.1. Usecase tổng quan



Hình 3. 1. Usecase tổng quan

3.3.2. Usecase quản trị viên



Hình 3. 2. Usecase nhà tuyển dụng

a. Đặc tả usecase quản lý tài khoản và phân quyền

Usecase	Quản lý Tài Khoản và Phân Quyền
Tác nhân	Quản trị viên
Mô tả ngắn gọn	Quản lý tài khoản và phân quyền là quy trình quản lý và kiểm soát quyền truy cập của người dùng vào hệ thống. Nó bao gồm việc tạo, chỉnh sửa, xóa tài khoản và cấp phân quyền cho từng người dùng trong hệ thống.
Điều kiện	<ol style="list-style-type: none"> 1. Quản trị viên có quyền truy cập vào hệ thống quản lý tài khoản và phân quyền. 2. Hệ thống quản lý tài khoản và phân quyền phải được triển khai và hoạt động.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Hệ thống quản lý tài khoản và phân quyền phải được triển khai và hoạt động

	2. Người quản trị hệ thống hoặc người được ủy quyền có quyền truy cập và quản lý tài khoản và phân quyền.
Ngoại lệ	Không có
Điều kiện đầu ra	<ol style="list-style-type: none"> 1. Tài khoản và phân quyền được quản lý một cách hiệu quả và an toàn trong hệ thống. 2. Người dùng được cấp phân quyền tương ứng để thực hiện các hoạt động liên quan đến công việc của mình trong hệ thống.

Bảng 3. 1. Usecase quản lý tài khoản và phân quyền

b. Đặc tả usecase quản lý đăng tin tức

Usecase	Đăng tin tức
Tác nhân	Quản trị viên
Mô tả ngắn gọn	Usecase này mô tả quá trình đăng tin tức mới vào hệ thống thông tin hoặc trang web tin tức của một tổ chức hoặc trang web.
Điều kiện	<ol style="list-style-type: none"> 1. Người dùng có quyền truy cập và đăng tin tức trên hệ thống hoặc trang. 2. Hệ thống phải hoạt động và chấp nhận các yêu cầu đăng tin tức mới.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Người dùng đăng nhập vào hệ thống hoặc trang web tin tức. 2. Chọn tùy chọn "Đăng Tin Tức". 3. Nhập tiêu đề, nội dung, hình ảnh và thông tin liên hệ liên quan đến tin tức. 4. Chọn danh mục hoặc chủ đề cho tin tức. 5. Lưu hoặc đăng tin tức mới.
Ngoại lệ	Không có
Điều kiện đầu ra	Tin tức mới được đăng thành công trên hệ thống hoặc trang web, hiển thị cho người đọc và cập nhật trong danh sách tin tức mới nhất.

Bảng 3. 2. Usecase quản lý đăng tin tức

c. Đặc tả usecase quản lý thông tin nhà tuyển dụng

Usecase	Quản lý thông tin nhà tuyển dụng
---------	----------------------------------

Tác nhân	Quản trị viên
Mô tả ngắn gọn	Usecase này mô tả quá trình quản lý thông tin về các nhà tuyển dụng trong hệ thống, bao gồm việc thêm mới, chỉnh sửa, xóa, và xem thông tin của nhà tuyển dụng.
Điều kiện	<ol style="list-style-type: none"> 1. Quản trị viên hệ thống và nhân viên quản lý thông tin nhà tuyển dụng phải có quyền truy cập vào hệ thống. 2. Hệ thống phải hoạt động và chấp nhận các yêu cầu quản lý thông tin nhà tuyển dụng.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Quản trị viên hoặc nhân viên quản lý đăng nhập vào hệ thống. 2. Chọn tùy chọn "Quản lý Thông Tin Nhà Tuyển Dụng". 3. Thực hiện các hành động sau: <ul style="list-style-type: none"> - Thêm mới thông tin về nhà tuyển dụng bằng cách nhập các chi tiết cần thiết. - Chỉnh sửa thông tin của nhà tuyển dụng bằng cách cập nhật thông tin đã có. - Xóa thông tin của nhà tuyển dụng nếu cần thiết. - Xem thông tin chi tiết về nhà tuyển dụng hiện có trong hệ thống.
Ngoại lệ	Không có
Điều kiện đầu ra	Thông tin về các nhà tuyển dụng được quản lý một cách hiệu quả và đầy đủ trong hệ thống.

Bảng 3. 3. Usecase quản lý thông tin nhà tuyển dụng

d. Đặc tả usecase xem thống kê dành cho quản trị viên

Usecase	Xem thống kê dành cho quản trị viên
Tác nhân	Quản trị viên
Mô tả ngắn gọn	Quy trình xem thống kê về số lượng bài đăng, số lượng nhà tuyển dụng và nhân viên.
Điều kiện	Quản trị viên đã đăng nhập vào ứng dụng.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Quản trị viên phải có tài khoản và quyền truy cập vào hệ thống tuyển dụng.

	2. Hệ thống tuyển dụng phải hoạt động ổn định và sẵn sàng sử dụng. 3. Phải có ít nhất một bài đăng tuyển dụng trên hệ thống.
Ngoại lệ	Không có
Điều kiện đầu ra	Quản trị viên hệ thống nhận được thông tin thống kê và báo cáo về hoạt động tuyển dụng của nhà tuyển dụng, giúp họ hiểu rõ hơn về hiệu suất và xu hướng tuyển dụng của mình.

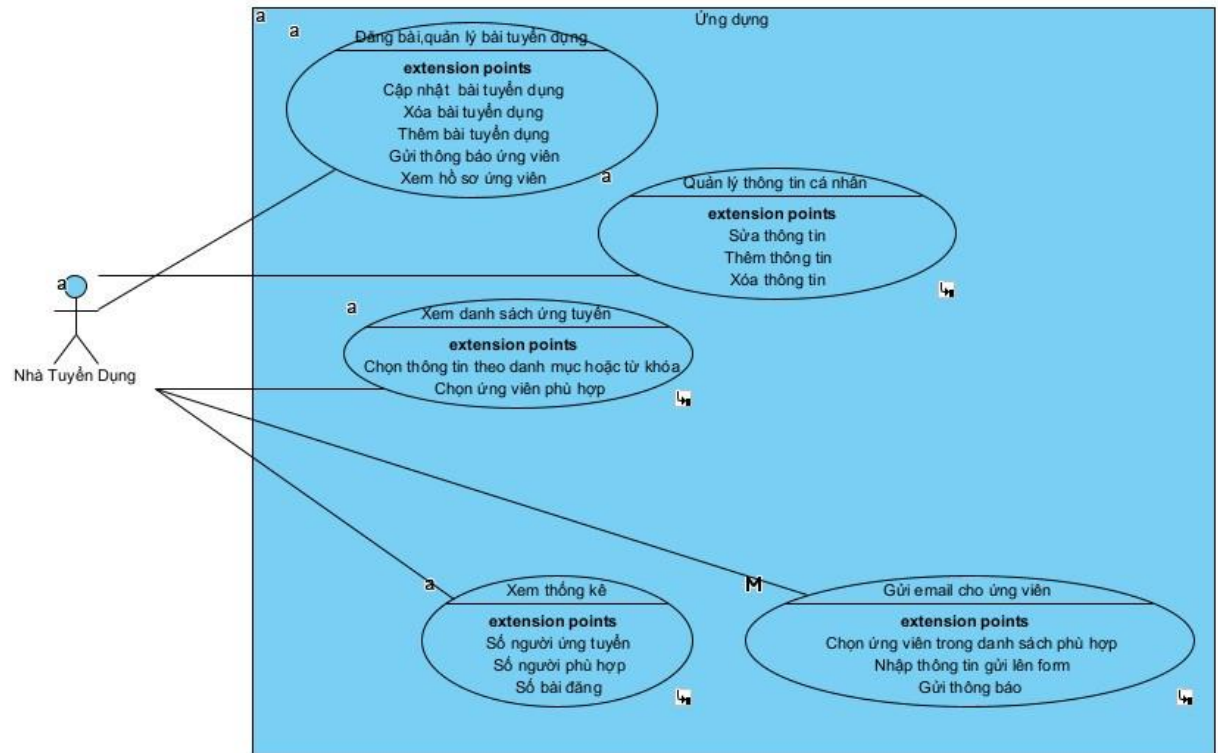
Bảng 3. 4. Usecase xem thống kê dành cho quản trị viên

e. Đặc tả usecase quản lý banner

Usecase	Quản lý Banner
Tác nhân	Quản trị viên
Mô tả ngắn gọn	Usecase này mô tả quá trình quản lý banner trên trang web hoặc ứng dụng, bao gồm thêm mới, chỉnh sửa, xóa và hiển thị banner
Điều kiện	Quản trị viên đã đăng nhập vào ứng dụng.
Chuỗi sự kiện	1. Quản trị viên hoặc người quản lý nội dung đăng nhập vào hệ thống quản lý. 2. Chọn tùy chọn "Quản lý Banner". 3. Thực hiện các hành động sau: <ul style="list-style-type: none"> - Thêm mới banner bằng cách tải lên hình ảnh, chọn vị trí, thời gian hiển thị, v.v. - Chỉnh sửa banner hiện có bằng cách cập nhật thông tin. - Xóa bỏ banner không cần thiết. - Xem danh sách các banner hiện có và trạng thái của chúng. - Lưu hoặc cập nhật thay đổi sau khi hoàn thành các hành động trên.
Ngoại lệ	Không có
Điều kiện đầu ra	Các banner được quản lý một cách hiệu quả và hiển thị đúng cách trên trang web hoặc ứng dụng.

Bảng 3. 5. Usecase quản lý banner

3.3.3. Usecase nhà tuyển dụng



Hình 3. 3. Usecase nhà tuyển dụng

a. Đặc tả usecase quản lý bài đăng tuyển dụng

Usecase	Quản lý bài đăng tuyển dụng
Tác nhân	Nhà tuyển dụng
Mô tả ngắn gọn	Quy trình quản lý bài đăng tuyển dụng bao gồm việc tạo, chỉnh sửa, duyệt và đăng các bài tuyển dụng trên trang web của công ty hoặc các nền tảng tuyển dụng khác.
Điều kiện	Nhà tuyển dụng đã đăng nhập vào ứng dụng.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Người dùng phải có tài khoản và quyền truy cập vào hệ thống quản lý tuyển dụng 2. Người dùng phải có quyền tạo, chỉnh sửa, và duyệt bài đăng tuyển dụng. 3. Hệ thống phải hoạt động ổn định và có kết nối internet.
Ngoại lệ	Nếu nhà trọ đã được lưu trữ trong danh sách nhà trọ, nhà tuyển dụng không thể lưu trữ lại.

Điều kiện đầu ra	1. Bài đăng tuyển dụng được tạo mới, chỉnh sửa, duyệt và đăng tải thành công lên các nền tảng tuyển dụng, giúp công ty thu hút được các ứng viên phù hợp.
------------------	---

Bảng 3. 6. Usecase quản lý bài đăng tuyển dụng

b. Đặc tả usecase quản lý thông tin cá nhân

Usecase	Quản lý thông tin cá nhân
Tác nhân	Nhà tuyển dụng
Mô tả ngắn gọn	Nhà tuyển dụng quản lý thông tin cá nhân của mình trong ứng dụng.
Điều kiện	Nhà tuyển dụng đã đăng nhập vào ứng dụng.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Nhà tuyển dụng truy cập vào phần quản lý thông tin cá nhân trên ứng dụng. 2. Nhà tuyển dụng cập nhật thông tin cá nhân như tên, địa chỉ, thông tin liên lạc, hình ảnh, mô tả về nhà trọ, v.v. 3. Nhà tuyển dụng lưu các thay đổi hoặc chỉnh sửa đã thực hiện.
Ngoại lệ	Không có
Điều kiện đầu ra	Hệ thống cho phép tạo và xuất các báo cáo dựa trên các đánh giá và trạng thái của ứng viên, và các báo cáo này phản ánh chính xác thông tin được lưu trữ.

Bảng 3. 7. Usecase quản lý thông tin cá nhân

c. Đặc tả usecase xem danh sách hồ sơ đánh giá

Usecase	Xem danh sách hồ sơ đánh giá
Tác nhân	Nhà tuyển dụng
Mô tả ngắn gọn	Quy trình xem và đánh giá danh sách hồ sơ ứng viên nộp vào các bài đăng tuyển dụng trên hệ thống tuyển dụng của công ty.
Điều kiện	Nhà tuyển dụng đã đăng nhập vào ứng dụng.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Người quản lý tuyển dụng phải có tài khoản và quyền truy cập vào hệ thống tuyển dụng. 2. Hệ thống tuyển dụng phải hoạt động ổn định và sẵn sàng sử dụng. 3. Có các hồ sơ ứng viên đã nộp vào các bài đăng tuyển dụng.

Ngoại lệ	Không có
Điều kiện đầu ra	Hệ thống cho phép tạo và xuất các báo cáo dựa trên các đánh giá và trạng thái của ứng viên, và các báo cáo này phản ánh chính xác thông tin được lưu trữ.

Bảng 3. 8. Usecase xem danh sách hồ sơ đánh giá

d. Đặc tả usecase xem danh sách hồ sơ ứng tuyển

Usecase	Xem danh sách hồ sơ đánh giá
Tác nhân	Nhà tuyển dụng
Mô tả ngắn gọn	Quy trình xem và đánh giá danh sách hồ sơ ứng viên nộp vào các bài đăng tuyển dụng trên hệ thống tuyển dụng của công ty.
Điều kiện	Nhà tuyển dụng đã đăng nhập vào ứng dụng.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Người quản lý tuyển dụng phải có tài khoản và quyền truy cập vào hệ thống tuyển dụng. 2. Hệ thống tuyển dụng phải hoạt động ổn định và sẵn sàng sử dụng. 3. Có các hồ sơ ứng viên đã nộp vào các bài đăng tuyển dụng.
Ngoại lệ	Không có
Điều kiện đầu ra	Các thông tin đánh giá và nhận xét về ứng viên phải được lưu trữ an toàn trong hệ thống và có khả năng tạo báo cáo và phân tích từ thông tin đã đánh giá.

Bảng 3. 9. Usecase xem danh sách hồ sơ đánh giá

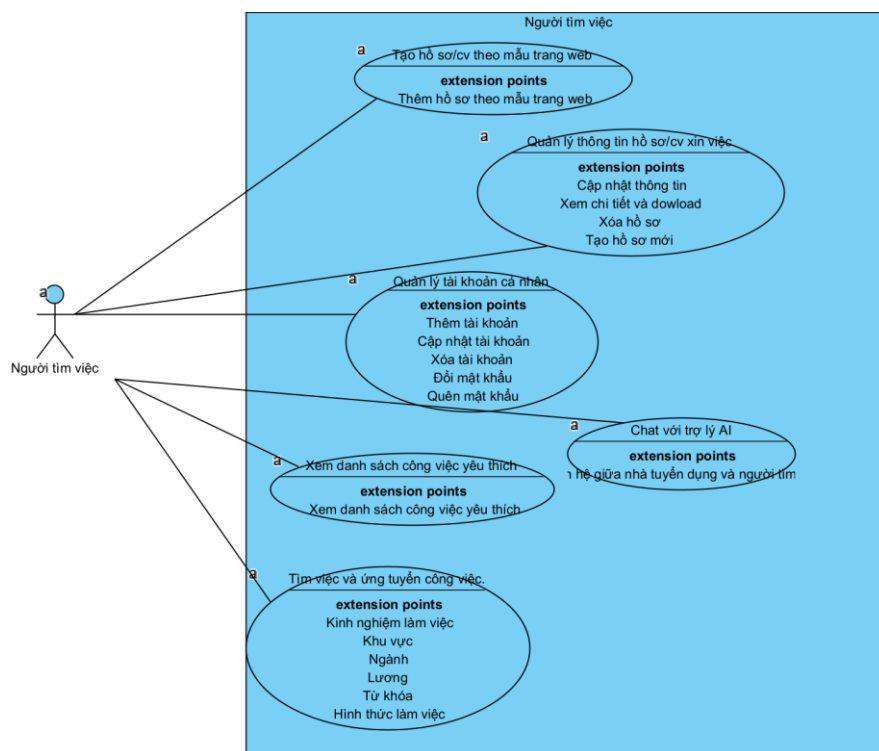
e. Đặc tả usecase xem thống kê dành cho nhà tuyển dụng

Usecase	Xem thống kê dành cho nhà tuyển dụng
Tác nhân	Nhà tuyển dụng
Mô tả ngắn gọn	Quy trình xem thống kê về số lượng bài đăng, số lượng ứng viên đã ứng tuyển và số lượng hồ sơ phù hợp với các bài đăng tuyển dụng.
Điều kiện	Nhà tuyển dụng đã đăng nhập vào ứng dụng.
Chuỗi sự kiện	<ol style="list-style-type: none"> 4. Nhà tuyển dụng phải có tài khoản và quyền truy cập vào hệ thống tuyển dụng.

	<p>5. Hệ thống tuyển dụng phải hoạt động ổn định và sẵn sàng sử dụng.</p> <p>6. Phải có ít nhất một bài đăng tuyển dụng trên hệ thống.</p>
Ngoại lệ	Không có
Điều kiện đầu ra	Các thông tin đánh giá và nhận xét về ứng viên phải được lưu trữ an toàn trong hệ thống và có khả năng tạo báo cáo và phân tích từ thông tin đã đánh giá.

Bảng 3. 10. Usecase xem thống kê dành cho nhà tuyển dụng

3.3.4. Usecase người tìm việc



Hình 3. 4. Usecase người tìm việc

a. Đặc tả usecase tạo hồ sơ/cv theo mẫu trang web

Usecase	Tạo hồ sơ/cv theo mẫu trang web
Tác nhân	Người tìm việc
Mô tả ngắn gọn	Quy trình tạo và hoàn thiện hồ sơ/CV trực tuyến theo mẫu có sẵn trên trang web tuyển dụng.
Điều kiện	1. Người tìm việc phải có tài khoản và quyền truy cập vào hệ thống tuyển dụng.

	2. Trang web tuyển dụng phải hoạt động ổn định và có các mẫu hồ sơ/CV có sẵn.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Đăng nhập hệ thống: <ul style="list-style-type: none"> ○ Ứng viên đăng nhập vào trang web tuyển dụng bằng tài khoản cá nhân. 2. Truy cập vào trang tạo hồ sơ/CV: <ul style="list-style-type: none"> ○ Chọn tùy chọn "Tạo hồ sơ/CV" từ giao diện người dùng. 3. Chọn mẫu hồ sơ/CV: <ul style="list-style-type: none"> ○ Lựa chọn mẫu hồ sơ/CV phù hợp từ các mẫu có sẵn trên trang web. 4. Nhập thông tin cá nhân: <ul style="list-style-type: none"> ○ Nhập thông tin cá nhân như tên, địa chỉ, số điện thoại, email. 5. Nhập thông tin học vấn: <ul style="list-style-type: none"> ○ Nhập thông tin về trình độ học vấn, bao gồm các trường đã học, chuyên ngành, và thời gian học. 6. Nhập thông tin kinh nghiệm làm việc: <ul style="list-style-type: none"> ○ Nhập thông tin về kinh nghiệm làm việc, bao gồm các công việc đã làm, vị trí, nhiệm vụ, và thời gian làm việc. 7. Nhập kỹ năng và chứng chỉ: <ul style="list-style-type: none"> ○ Nhập các kỹ năng chuyên môn, chứng chỉ và giải thưởng (nếu có). 8. Xem lại và chỉnh sửa: <ul style="list-style-type: none"> ○ Xem lại toàn bộ hồ sơ/CV, chỉnh sửa các thông tin nếu cần thiết. 9. Lưu và xuất bản: <ul style="list-style-type: none"> ○ Lưu lại hồ sơ/CV và xuất bản trên trang web tuyển dụng hoặc tải xuống.
Ngoại lệ	Không có

Điều kiện đầu ra	<ol style="list-style-type: none"> 1. Hồ sơ/CV được tạo hoàn chỉnh với đầy đủ thông tin cá nhân, học vấn, kinh nghiệm làm việc, kỹ năng và chứng chỉ. 2. Hồ sơ/CV được lưu trữ an toàn trên hệ thống và có thể truy cập lại để chỉnh sửa hoặc tải xuống. 3. Hồ sơ/CV sẵn sàng để xuất bản trên trang web tuyển dụng hoặc tải xuống dưới dạng tệp.
------------------	--

Bảng 3. 11. Usecase Tạo hồ sơ/cv theo mẫu trang web

b. Đặc tả usecase quản lý thông tin hồ sơ xin việc

Usecase	Quản lý thông tin hồ sơ xin việc
Tác nhân	Người tìm việc
Mô tả ngắn gọn	Quy trình quản lý thông tin hồ sơ xin việc bao gồm các thao tác cập nhật, xem chi tiết, xóa và tạo hồ sơ trên trang web tuyển dụng của công ty.
Điều kiện	<ol style="list-style-type: none"> 1. Ứng viên phải có tài khoản và quyền truy cập vào hệ thống tuyển dụng. 2. Hệ thống tuyển dụng phải hoạt động ổn định và sẵn sàng sử dụng.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Đăng nhập: Ứng viên đăng nhập vào hệ thống. 2. Tạo hồ sơ: Chọn "Tạo hồ sơ", nhập thông tin, lưu và xuất bản. 3. Xem chi tiết: Truy cập quản lý hồ sơ, chọn hồ sơ để xem. 4. Cập nhật: Chọn hồ sơ, thay đổi thông tin, lưu lại. 5. Xóa: Chọn hồ sơ, xác nhận xóa.
Ngoại lệ	Không có
Điều kiện đầu ra	<ol style="list-style-type: none"> 1. Hồ sơ xin việc được tạo, cập nhật, xem chi tiết và xóa theo yêu cầu của ứng viên. 2. Thông tin hồ sơ được lưu trữ an toàn và chính xác trong hệ thống. 3. Các thao tác quản lý hồ sơ (tạo, cập nhật, xem chi tiết, xóa) được thực hiện thành công và không gặp lỗi.

Bảng 3. 12. Usecase quản lý lý thông tin hồ sơ xin việc

c. Đặc tả usecase quản lý thông tin cá nhân

Usecase	Quản lý thông tin cá nhân
Tác nhân	Nhà tuyển dụng
Mô tả ngắn gọn	Nhà tuyển dụng quản lý thông tin cá nhân của mình trong ứng dụng.
Điều kiện	Nhà tuyển dụng đã đăng nhập vào ứng dụng.
Chuỗi sự kiện	<ol style="list-style-type: none"> 4. Nhà tuyển dụng truy cập vào phần quản lý thông tin cá nhân trên ứng dụng. 5. Nhà tuyển dụng cập nhật thông tin cá nhân như tên, địa chỉ, thông tin liên lạc, hình ảnh, mô tả về nhà trọ, v.v. 6. Nhà tuyển dụng lưu các thay đổi hoặc chỉnh sửa đã thực hiện.
Ngoại lệ	Không có
Điều kiện đầu ra	Hệ thống cho phép tạo và xuất các báo cáo dựa trên các đánh giá và trạng thái của ứng viên, và các báo cáo này phản ánh chính xác thông tin được lưu trữ.

Bảng 3. 13. Usecase quản lý thông tin cá nhân

d. Đặc tả usecase xem danh sách công việc yêu thích

Usecase	Xem danh sách công việc yêu thích
Tác nhân	Người tìm việc
Mô tả ngắn gọn	Quy trình xem danh sách các công việc đã được đánh dấu là yêu thích trên trang web tuyển dụng..
Điều kiện	Người tìm việc đã đăng nhập vào ứng dụng.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Ứng viên phải có tài khoản và quyền truy cập vào hệ thống tuyển dụng. 2. Hệ thống tuyển dụng phải hoạt động ổn định và sẵn sàng sử dụng. 3. Ứng viên đã đánh dấu ít nhất một công việc là yêu thích.
Ngoại lệ	Không có
Điều kiện đầu ra	<ol style="list-style-type: none"> 1. Danh sách công việc yêu thích được hiển thị đầy đủ và chính xác.

	2. Ứng viên có thể xem chi tiết từng công việc từ danh sách này.
--	--

Bảng 3. 14. Usecase danh sách công việc yêu thích

e. Đặc tả usecase chat với trợ lý AI

Usecase	Chat với trợ lý AI
Tác nhân	Người tìm việc
Mô tả ngắn gọn	Quy trình tương tác và trò chuyện với trợ lý AI trên hệ thống.
Điều kiện	<ol style="list-style-type: none"> 1. Người dùng phải có tài khoản và quyền truy cập vào hệ thống (nếu yêu cầu). 2. Hệ thống trợ lý AI phải hoạt động ổn định và sẵn sàng sử dụng.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Đăng nhập hệ thống (nếu cần): 2. Người dùng đăng nhập vào hệ thống bằng tài khoản cá nhân. 3. Truy cập giao diện chat với trợ lý AI: 4. Chọn tùy chọn "Chat với trợ lý AI" từ giao diện người dùng. 5. Bắt đầu cuộc trò chuyện: 6. Người dùng nhập tin nhắn hoặc câu hỏi vào ô chat. 7. Nhận phản hồi từ trợ lý AI: 8. Trợ lý AI phân tích và phản hồi tin nhắn hoặc câu hỏi của người dùng. 9. Tiếp tục trò chuyện: 10. Người dùng có thể tiếp tục nhập tin nhắn hoặc câu hỏi tiếp theo và nhận phản hồi từ trợ lý AI.
Ngoại lệ	Không có
Điều kiện đầu ra	<ol style="list-style-type: none"> 1. Người dùng nhận được phản hồi chính xác và hữu ích từ trợ lý AI. 2. Cuộc trò chuyện được thực hiện suôn sẻ mà không gặp lỗi kỹ thuật. 3. Người dùng có thể kết thúc cuộc trò chuyện bất cứ lúc nào và lưu lại lịch sử trò chuyện nếu cần.

Bảng 3. 15. Usecase chat với trợ lý AI

f. Đặc tả usecase tìm việc và ứng tuyển công việc

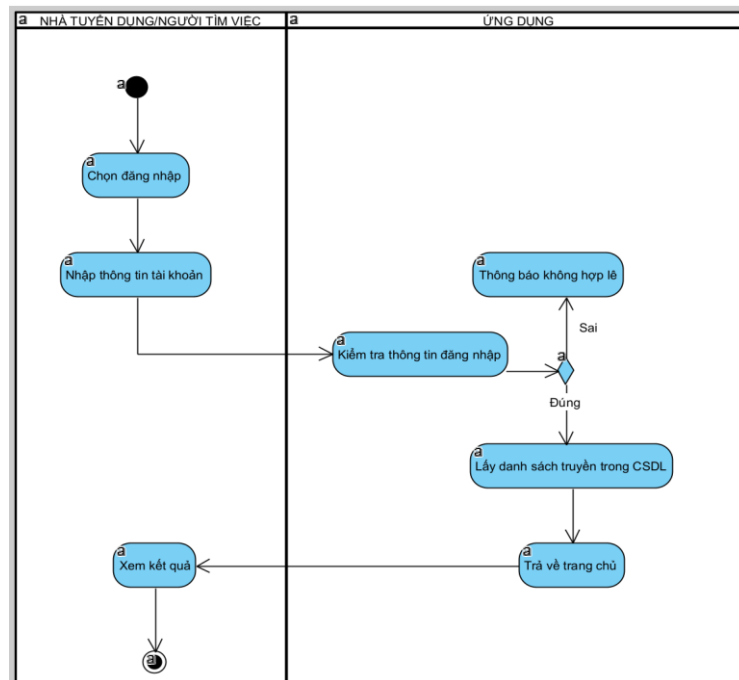
Usecase	Chat với trợ lý AI
Tác nhân	Người tìm việc
Mô tả ngắn gọn	Quy trình tìm kiếm và ứng tuyển vào các công việc trên trang web tuyển dụng.
Điều kiện	<ol style="list-style-type: none"> 1. Ứng viên phải có tài khoản và quyền truy cập vào hệ thống tuyển dụng. 2. Hệ thống tuyển dụng phải hoạt động ổn định và sẵn sàng sử dụng.
Chuỗi sự kiện	<ol style="list-style-type: none"> 1. Đăng nhập hệ thống: 2. Ứng viên đăng nhập vào hệ thống tuyển dụng bằng tài khoản cá nhân. 3. Tìm kiếm công việc: 4. Nhập các tiêu chí tìm kiếm (ví dụ: ngành nghề, địa điểm, mức lương) vào ô tìm kiếm. 5. Hệ thống hiển thị danh sách các công việc phù hợp. 6. Xem chi tiết công việc: 7. Chọn một công việc từ danh sách để xem thông tin chi tiết (mô tả công việc, yêu cầu, quyền lợi). 8. Ứng tuyển công việc: 9. Chọn tùy chọn "Ứng tuyển" trên trang chi tiết công việc. 10. Đính kèm hồ sơ/CV và các tài liệu cần thiết khác (thư xin việc, chứng chỉ). 11. Gửi đơn ứng tuyển. 12. Xác nhận ứng tuyển: 13. Hệ thống gửi thông báo xác nhận đơn ứng tuyển đã được gửi thành công.
Ngoại lệ	Không có
Điều kiện đầu ra	<ol style="list-style-type: none"> 1. Ứng viên tìm thấy và xem chi tiết các công việc phù hợp.

	<p>2. Đơn ứng tuyển được gửi thành công và ứng viên nhận được thông báo xác nhận.</p> <p>3. Thông tin ứng tuyển được lưu trữ an toàn trên hệ thống.</p>
--	---

Bảng 3. 16. Usecase tìm việc và ứng tuyển công việc

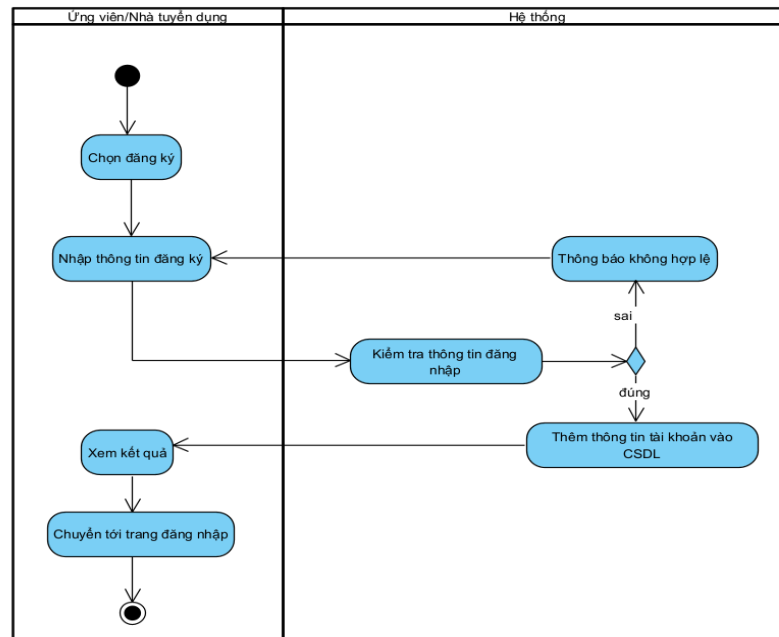
3.4. Biểu đồ hoạt động

3.4.1. Biểu đồ hoạt động đăng nhập



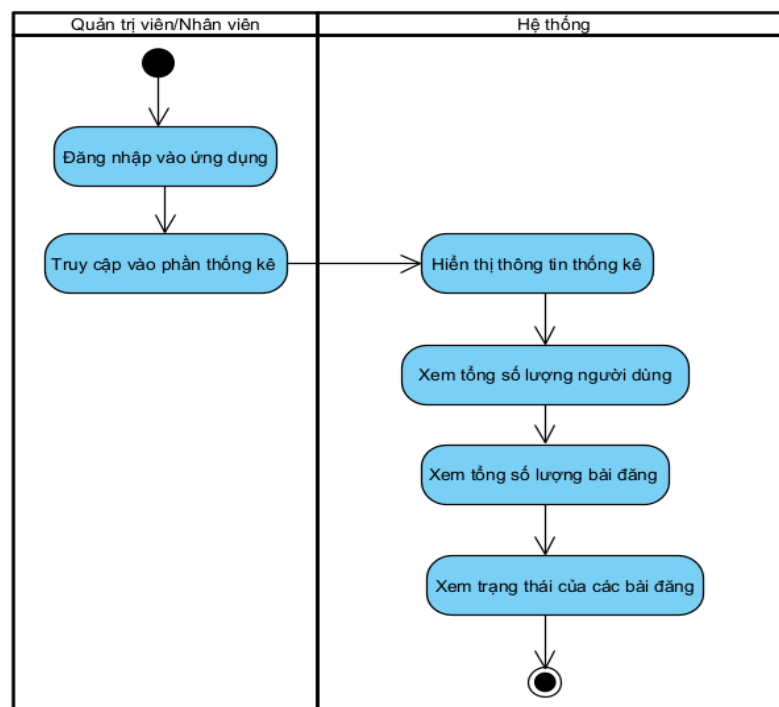
Hình 3. 5. Biểu đồ hoạt động đăng nhập

3.4.2. Biểu đồ hoạt động đăng ký



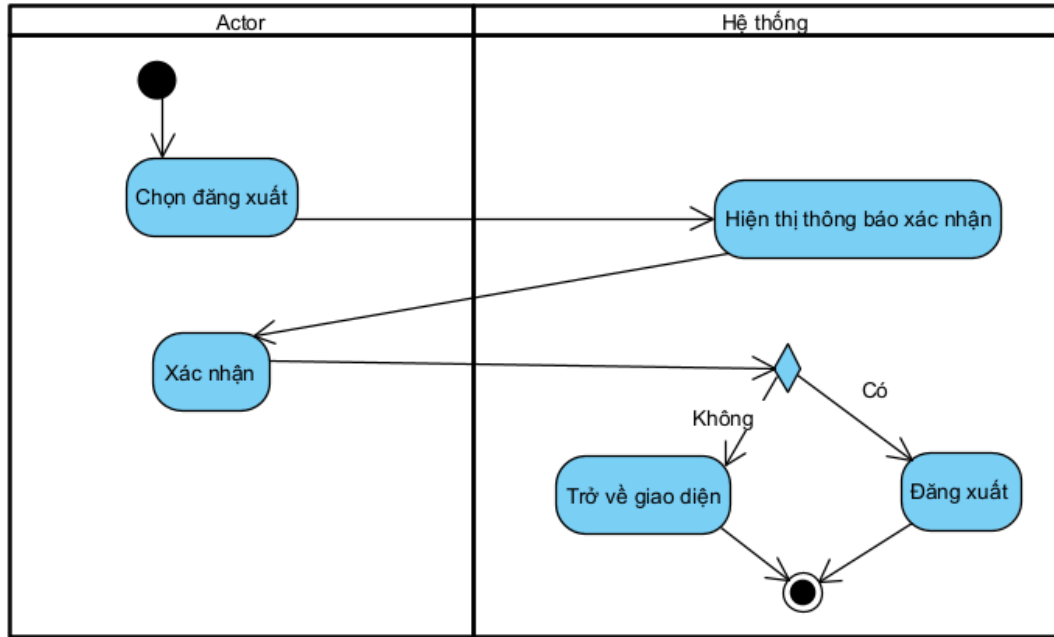
Hình 3. 6. Biểu đồ hoạt động đăng ký

3.4.3. Biểu đồ hoạt động thống kê



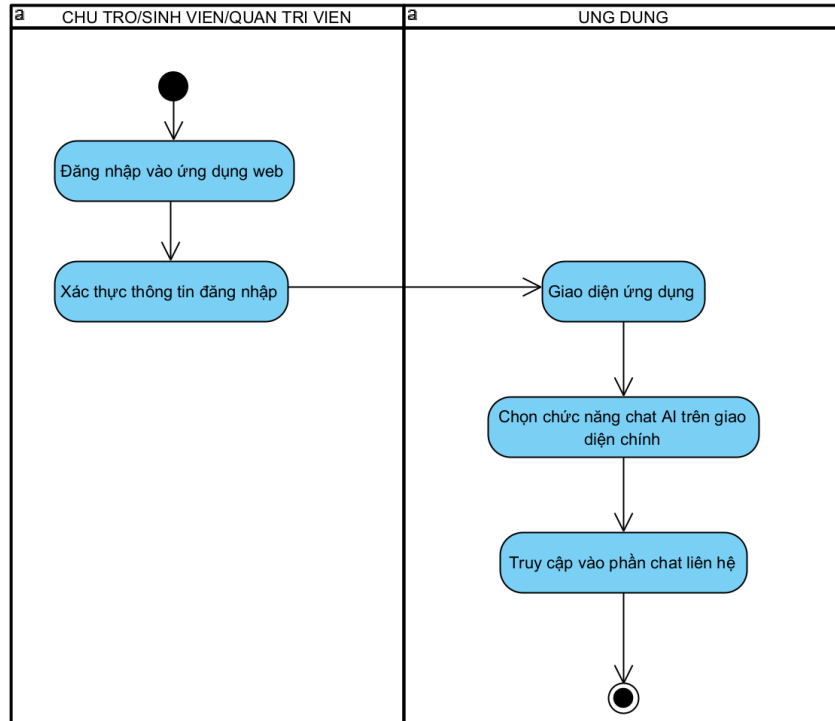
Hình 3. 7. Biểu đồ hoạt động thống kê

3.4.4. Biểu đồ hoạt động đăng xuất



Hình 3. 8. Biểu đồ hoạt động đăng xuất

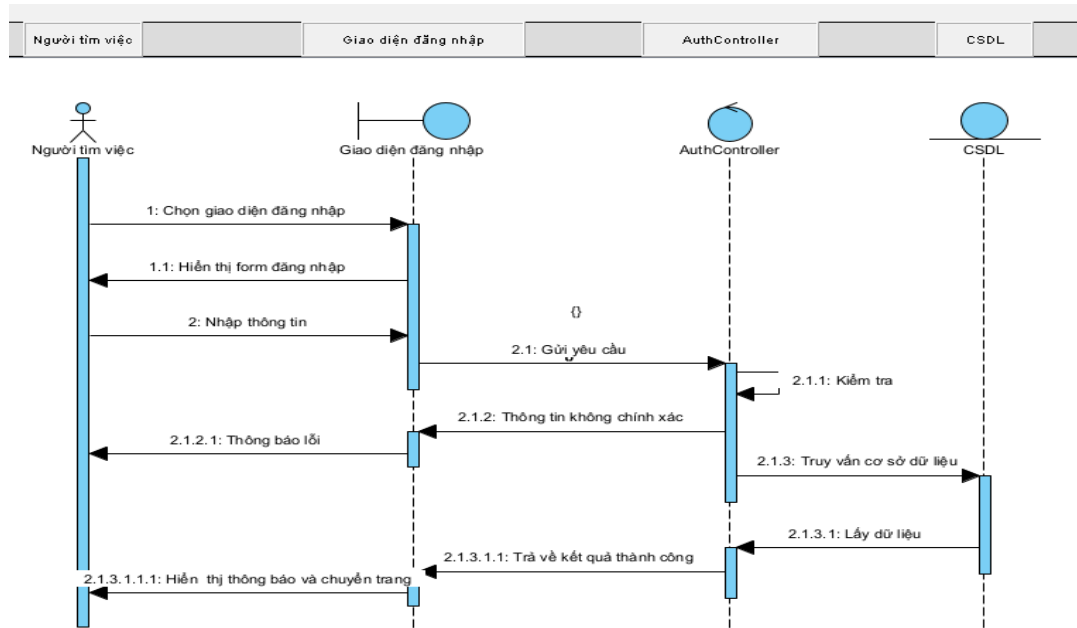
3.4.5. Biểu đồ hoạt động Chat AI



Hình 3. 9. Biểu đồ hoạt động Chat AI

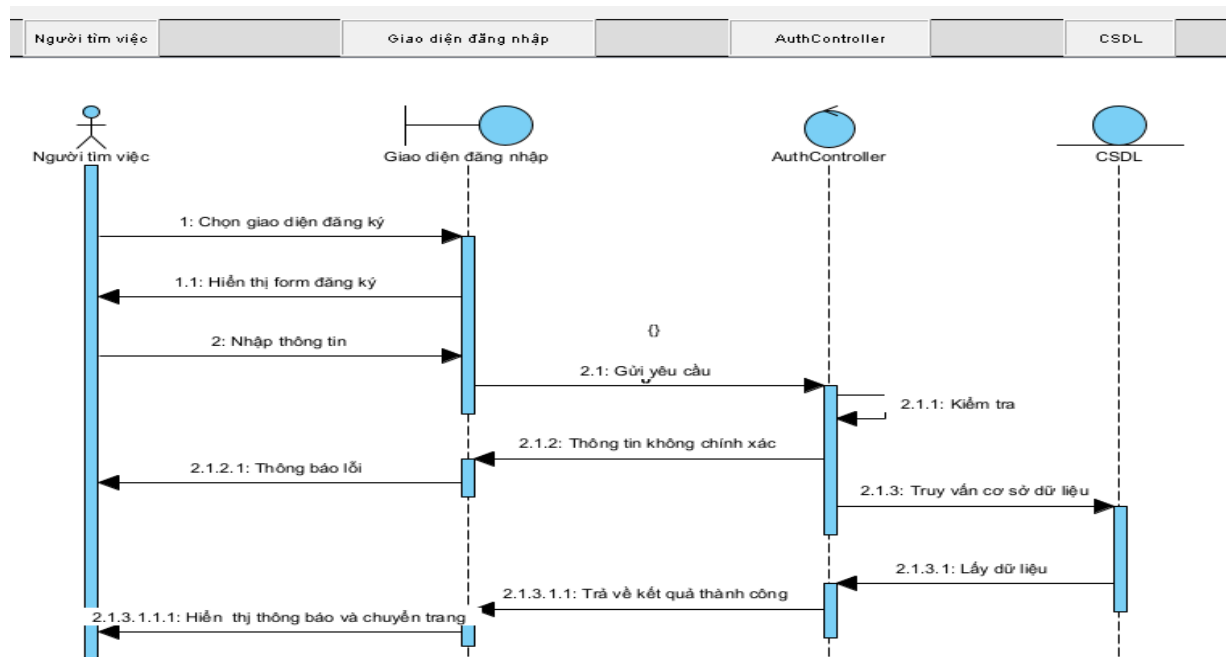
3.5. Biểu đồ tuần tự

3.5.1. Biểu đồ tuần tự đăng nhập



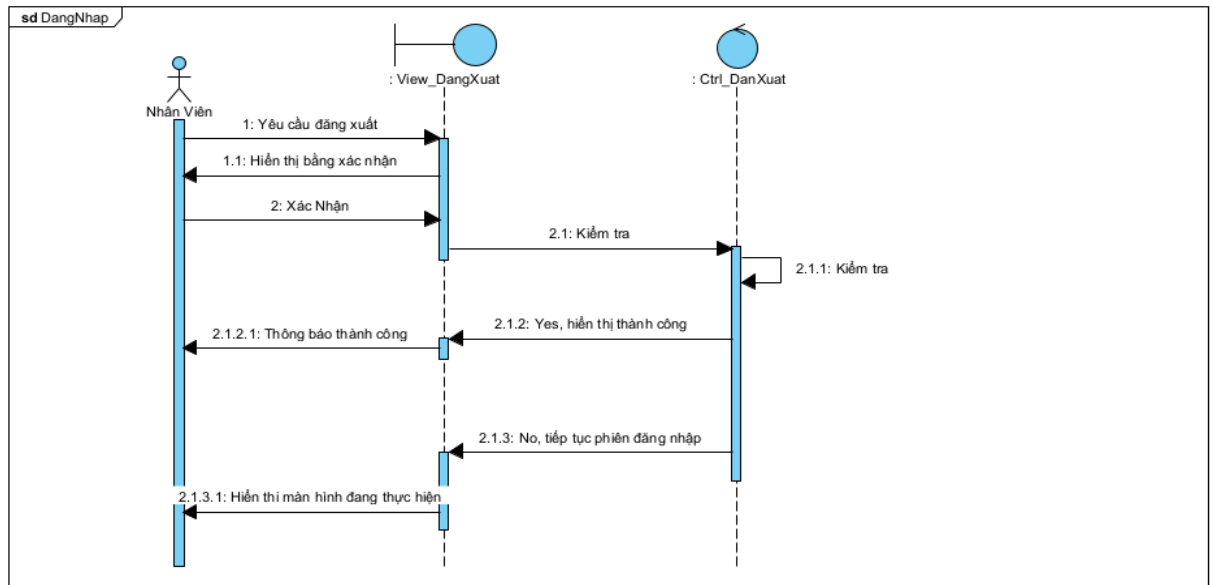
Hình 3. 10. Biểu đồ tuần tự đăng nhập

3.5.2. Biểu đồ tuần tự đăng ký



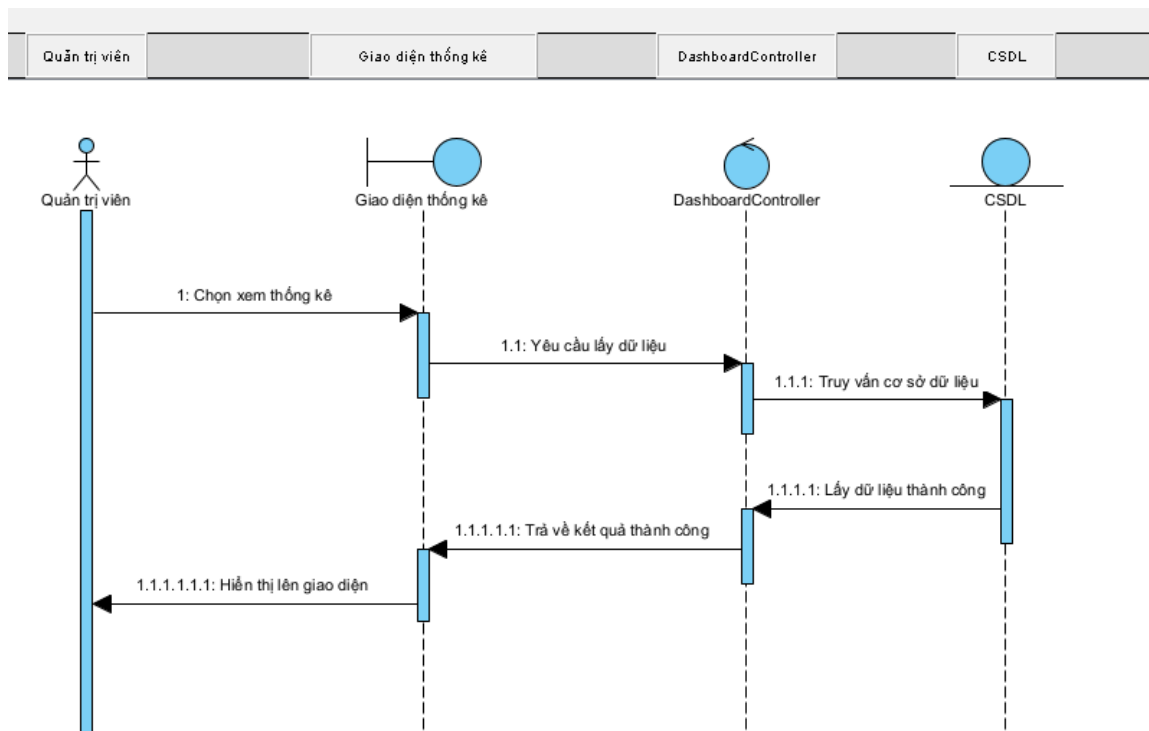
Hình 3. 11. Biểu đồ hoạt động đăng ký

3.5.3. Biểu đồ tuần tự đăng xuất



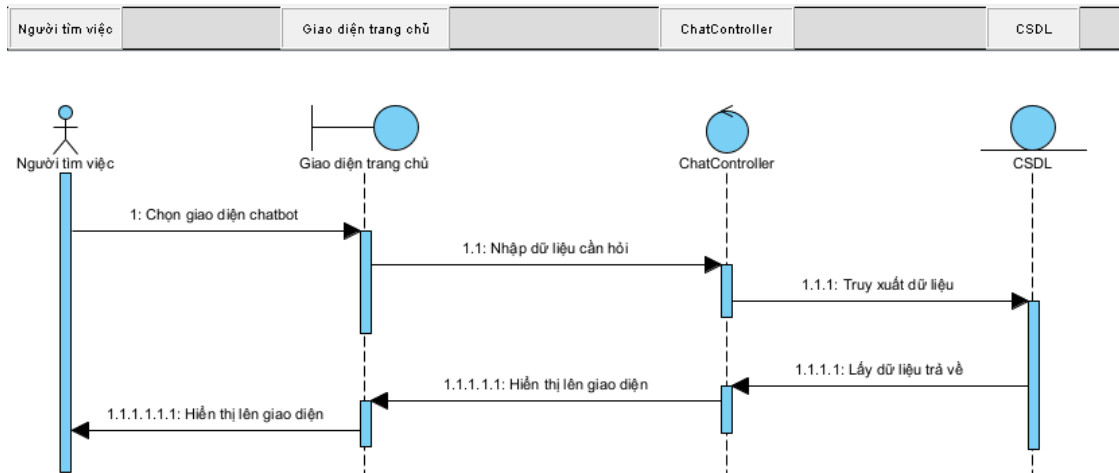
Hình 3. 12. Biểu đồ tuần tự đăng xuất

3.5.4. Biểu đồ tuần tự Thống kê



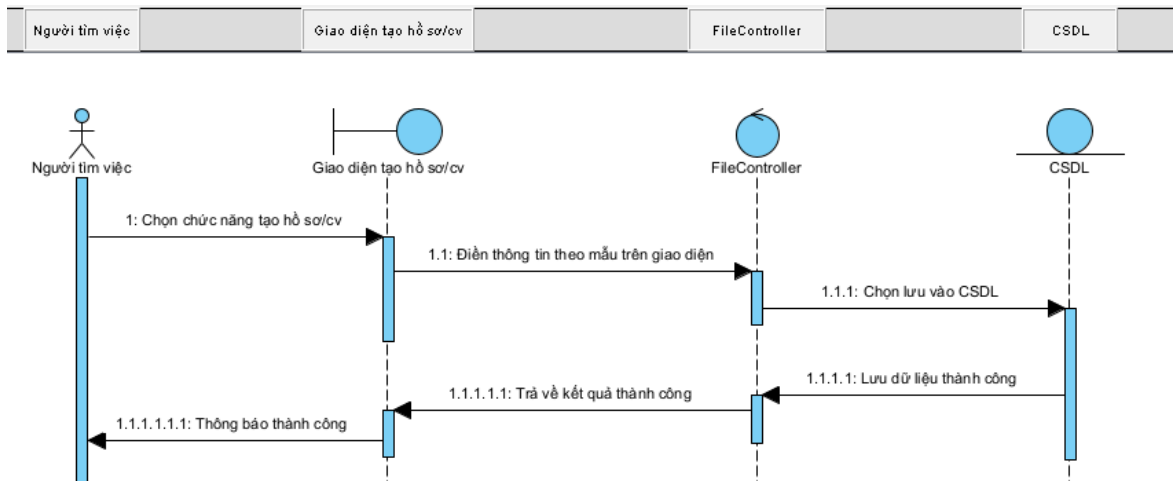
Hình 3. 13. Biểu đồ tuần tự thống kê

3.5.5. Biểu đồ tuần tự Chat AI



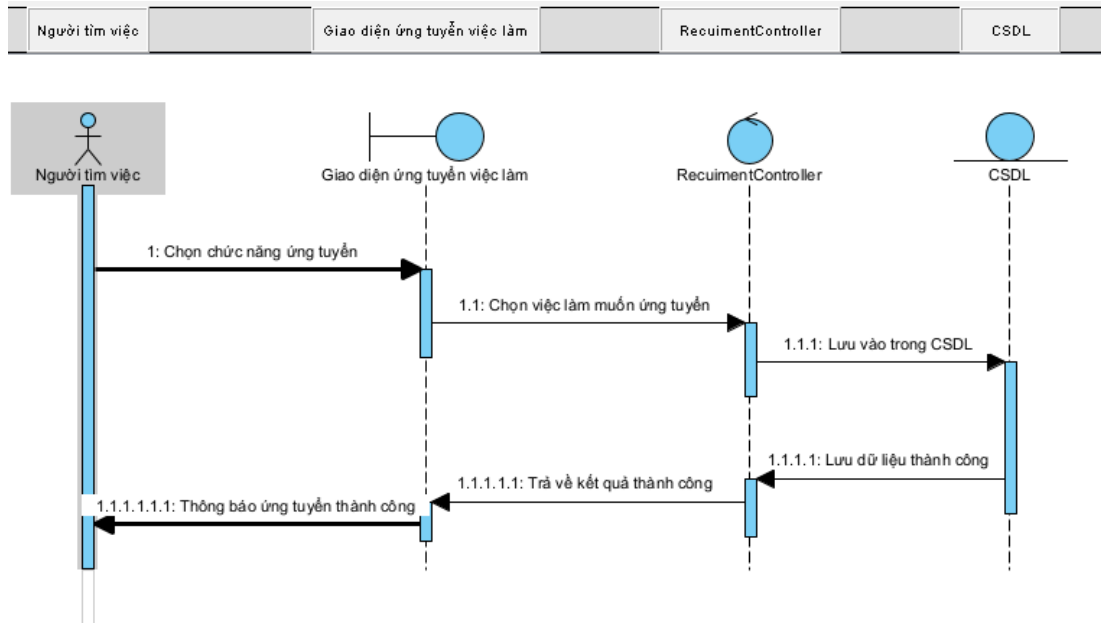
Hình 3. 14. Biểu đồ tuần tự Chat AI

3.5.6. Biểu đồ tuần tự tạo hồ sơ/ CV



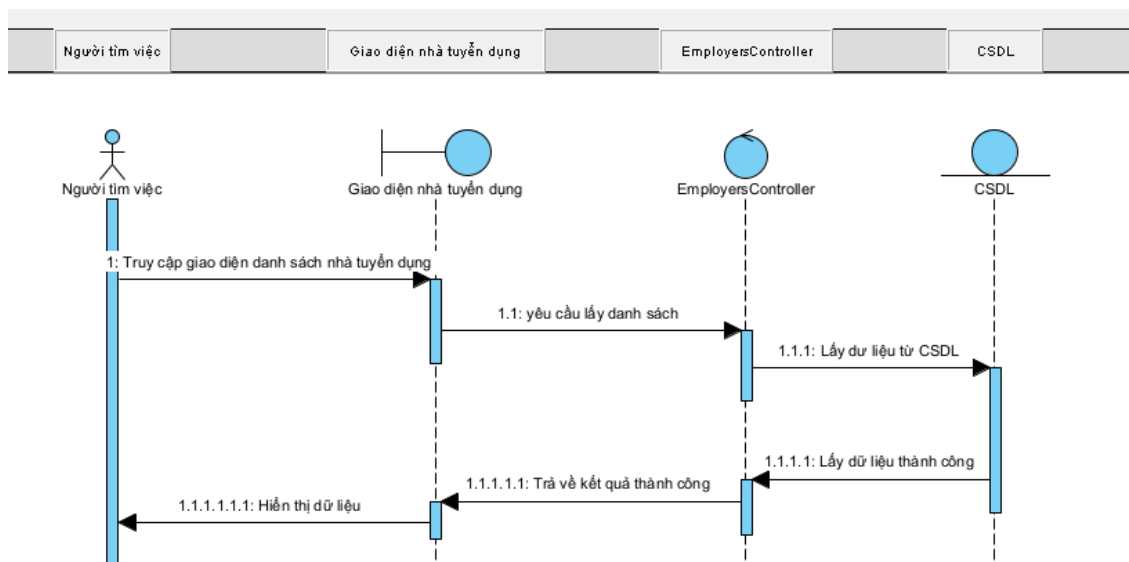
Hình 3. 15. Biểu đồ tuần tự tạo hồ sơ/ CV

3.5.7. Biểu đồ tuần tự Ứng tuyển việc làm



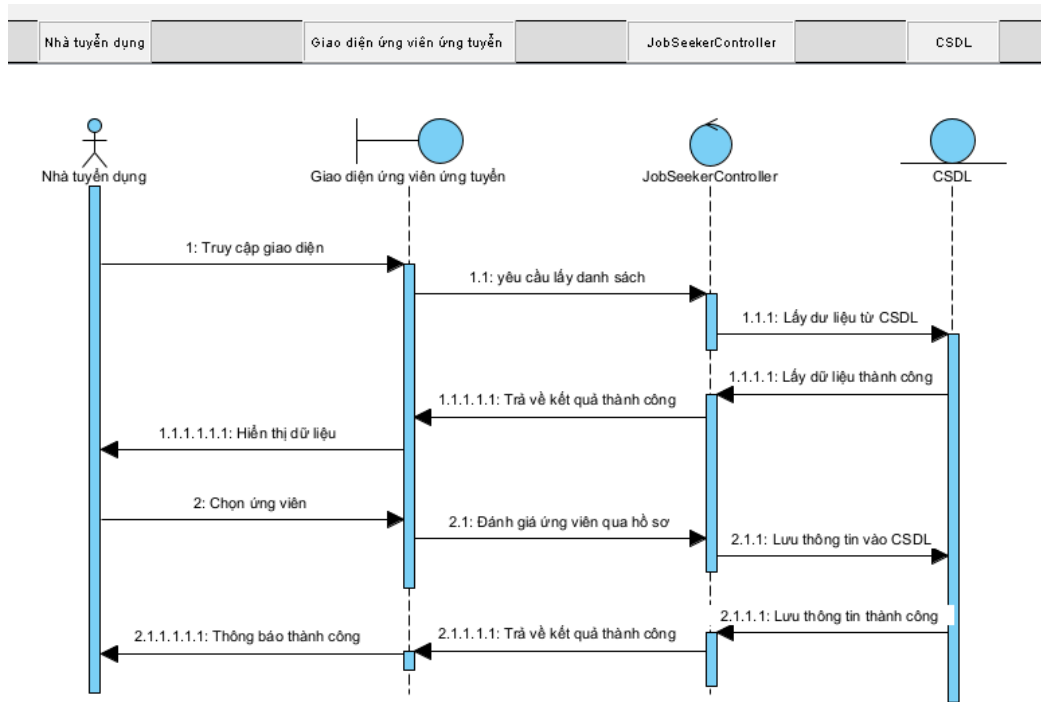
Hình 3. 16. Biểu đồ tuần tự Ứng tuyển việc làm

3.5.8. Biểu đồ tuần tự Xem danh sách nhà tuyển dụng



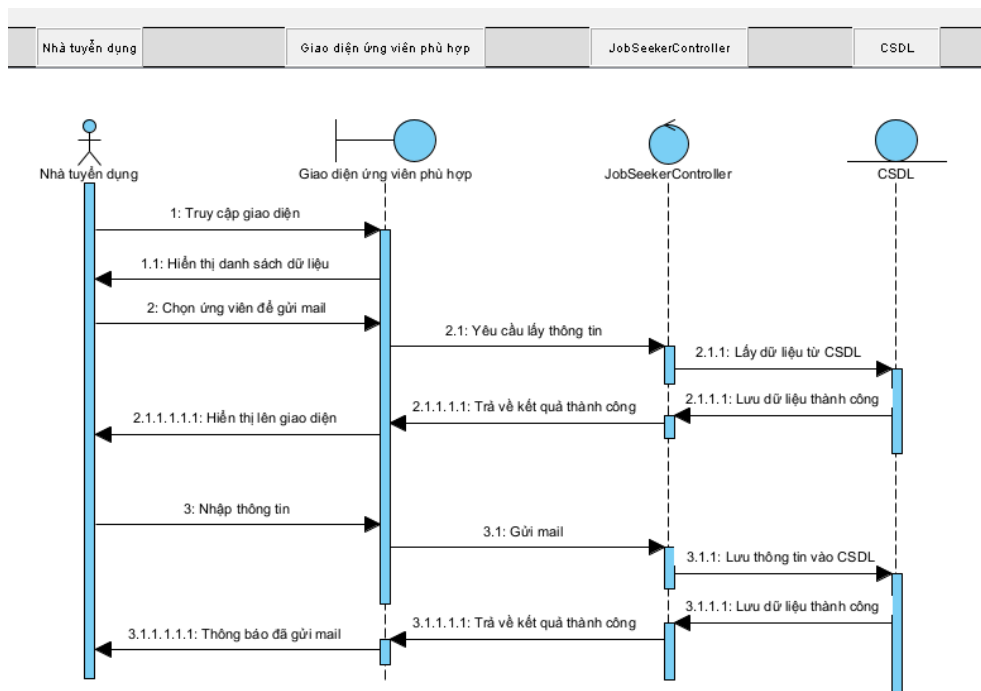
Hình 3. 17. Biểu đồ tuần tự Xem danh sách nhà tuyển dụng

3.5.9. Biểu đồ tuần tự Xem danh sách ứng viên ứng tuyển



Hình 3. 18. Biểu đồ tuần tự Xem danh sách ứng viên ứng tuyển

3.9.10. Biểu đồ tuần tự Gửi thông báo qua Email



Hình 3. 19. Biểu đồ tuần tự Gửi thông báo qua Email

3.6. Thiết kế cơ sở dữ liệu

3.6.1. Bảng Job_Seekers

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
Job_SeekerId	Long		Khóa chính	Mã người xin việc
FullName	Nvarchar	Max	Not null	Họ tên người xin việc
Birthday	Date		Null	Ngày sinh
PhoneNumber	Long		Null	Số điện thoại
Address	Nvarchar	255	Null	Địa chỉ
Avatar	Nvarchar	Max	Null	Ảnh đại diện
Email	Nvarchar	Max	Not null	Email
Password	Nvarchar	Max	Not null	Mật khẩu
Gender	Bit		Null	Giới tính
Academic_Level	Nvarchar	255	Null	Trình độ học vấn
Role	Nvarchar	255	Not null	Quyền đăng nhập

Bảng 3. 17. Bảng Job_Seekers

3.6.2. Bảng Jobs

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
JobId	Long		Khóa chính	Mã công việc
JobName	Nvarchar	Max	Not null	Tên công việc
JobDescription	Nvarchar	Max	Not null	Chi tiết công việc
RequestJob	Nvarchar	Max	Not null	Yêu cầu công việc
BenefitsJob	Nvarchar	Max	Not null	Quyền lợi khi đi làm
AddressJob	Nvarchar	Max	Not null	Địa chỉ nơi làm việc
WorkingTime	Nvarchar	Max	Not null	Thời gian làm việc
ExpirationDate	Date		Not null	Ngày hết hạn ứng tuyển
WorkexperienceId	Long		Khóa ngoại	Mã kinh nghiệm làm việc
FormofworkId	Long		Khóa ngoại	Mã hình thức làm việc
CityId	Long		Khóa ngoại	Mã khu vực, thành phố

SalaryId	Long		Khóa ngoại	Mã lương công việc
EmployersId	Long		Khóa ngoại	Mã nhà tuyển dụng
ProfessionId	Long		Khóa ngoại	Mã ngành nghề công việc
LevelworkId	Long		Khóa ngoại	Mã kinh nghiệm làm việc

Bảng 3. 18. Bảng Jobs

3.6.3. Bảng Favourite_Job

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
Favoufite_Job_Id	Long		Khóa chính	Mã công việc yêu thích
JobId	Long		Not null	Mã công việc
IsFavoufite_Job	Bit		Not nul	Yêu thích công việc
Job_SeekerId	Long		Khóa ngoại	Mã người tìm việc

Bảng 3. 19. Bảng Favourite_Job

3.6.4. Bảng File_Cv

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
FileCvId	Long		Khóa chính	Mã hồ sơ/cv
FileCVName	Nvarchar	Max	Not null	Tên hồ sơ/cv
HostPath	Nvarchar	Max	Not null	Đường dẫn máy chủ
FileCvPath	Nvarchar	255	Not null	Đường dẫn lưu hồ sơ/cv
Job_SeekerId	Long		Khóa ngoại	Mã người tìm việc

Bảng 3. 20. Bảng File_Cv

3.6.5. Bảng Employers

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
EmployersId	Long		Khóa chính	Mã nhà tuyển dụng
CompanyName	Nvarchar	255	Not null	Tên công ty

CompanyAddress	Nvarchar	Max	Null	Địa chỉ công ty
ContactEmail	Nvarchar	255	Null	Email liên hệ
ContactPhoneNumber	Nvarchar	255	Null	Số điện thoại liên hệ
CompanyWebsite	Nvarchar	Max	Null	Website công ty
CompanyDescription	Nvarchar	Max	Null	Mô tả về công ty
CompanyLogo	Nvarchar	Max	Null	Logo của công ty
Email	Nvarchar	Max	Not null	Email đăng nhập
Password	Nvarchar	Max	Not null	Mật khẩu đăng nhập
IsActive	Bit		Null	Trạng thái hoạt động
IsStatus	Bit		Null	Trạng thái toàn khoản
Role	Nvarchar	255	Not null	Quyền đăng nhập

Bảng 3. 21. Bảng Employers

3.6.6. Bảng Notification

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
NotificationId	Long		Khóa chính	Mã thông báo
Job_SeekerId	Long		Khóa ngoại	Mã người tìm việc
EmployersId	Long		Khóa ngoại	Mã nhà tuyển dụng
Message	Nvarchar	Max	Null	Tin nhắn thông báo
IsRead	Bit		Null	Trạng thái thông báo
Notification_CreatedAt	Date		Null	Ngày tạo thông báo
JobId	Long		Khóa ngoại	Mã công việc

Bảng 3. 22. Bảng Notification

3.6.7. Bảng Refresh_Tokens

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
RefreshToken_Id	Long		Khóa chính	Mã refresh token

Refresh_TokenExpires	Nvarchar	Max	Not null	Hết hạn refresh token
RefreshToken	Nvarchar	Max	Not null	Mã token
RefreshTokenExpiration	Nvarchar	Max	Not null	Thời gian hết hạn
Account_Id	Long	Max	Khóa ngoại	Mã tài khoản

Bảng 3. 23. Bảng Refresh_Tokens

3.6.8. Bảng News

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
NewsId	Long		Khóa chính	Mã tin tức
Title	Nvarchar	Max	Null	Tiêu đề tin tức
Content	Nvarchar	Max	Not null	Nội dung tin tức
Image	Nvarchar	Max	Null	Hình ảnh tin tức
News_Create	Date		Not null	Ngày đăng

Bảng 3. 24. Bảng News

3.6.9. Bảng Permissions

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
PermissionId	Long		Khóa chính	Mã quyền
PermissionName	Nvarchar	Max	Not null	Tên quyền

Bảng 3. 25. Bảng Permissions

3.6.10. Bảng Roles

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
RoleId	Long		Khóa chính	Mã chức vụ
RoleName	Nvarchar	255	Null	Tên chức vụ
RoleDescription	Nvarchar	255	Null	Mô tả chức vụ
PermissionId	Long		Khóa ngoại	Mã quyền

Bảng 3. 26. Bảng Roles

3.6.11. Bảng Formofworks

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
------------	--------------	------------	-----------	-------

FormofworkId	Long		Khóa chính	Mã hình thức làm việc
FormofworkName	Nvarchar	Max	Not null	Tên hình thức làm việc

Bảng 3. 27. Bảng Formofworks

3.6.12. Bảng Professions

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
ProfessionId	Long		Khóa chính	Mã ngành nghề
ProfessionName	Nvarchar	Max	Not null	Tên ngành nghề

Bảng 3. 28. Bảng Professions

3.6.13. Bảng Levelworks

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
LevelworkId	Long		Khóa chính	Mã hình thức làm việc
LevelworkName	Nvarchar	Max	Not null	Tên hình thức làm việc
ProfessionId	Long		Khóa ngoại	Mã ngành nghề

Bảng 3. 29. Bảng Levelworks

3.6.14. Bảng Salaries

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
SalaryId	Long		Khóa chính	Mã lương
SalaryPrice	Nvarchar	Max	Not null	Tiền lương

Bảng 3. 30. Bảng Salaries

3.6.15. Bảng Cities

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
CityId	Long		Khóa chính	Mã khu vực
CityName	Nvarchar	Max	Not null	Tên khu vực

Bảng 3. 31. Bảng Cities

3.6.16. Bảng Workexperiences

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
WorkexperienceId	Long		Khóa chính	Mã kinh nghiệm làm việc
WorkexperienceName	Nvarchar	Max	Not null	Tên kinh nghiệm làm việc

Bảng 3. 32. Bảng Workexperiences

3.6.17. Bảng Banners

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
BannerId	Long		Khóa chính	Khóa chính của banner
BannerPath	Nvarchar	Max	Not null	Đường link ảnh

Bảng 3. 33. Bảng Banners

3.6.18. Bảng Recruitement

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
RecruitmentId	Long		Khóa chính	Mã tuyển dụng
Job_SeekerId	Long		Not null	Mã người xin việc
JobId	Long		Not null	Mã công việc
EmployersId	Long		Not null	Mã nhà tuyển dụng
RecruitmentDateTime	Date		Not null	Thời gian ứng tuyển
Content	Nvarchar	Max	Not null	Nội dung ứng tuyển
IsStatus	Bit		Null	Trạng thái ứng tuyển
IsFeedback	Bit		Null	Nhận xét ứng tuyển

Bảng 3. 34. Bảng Recruitement

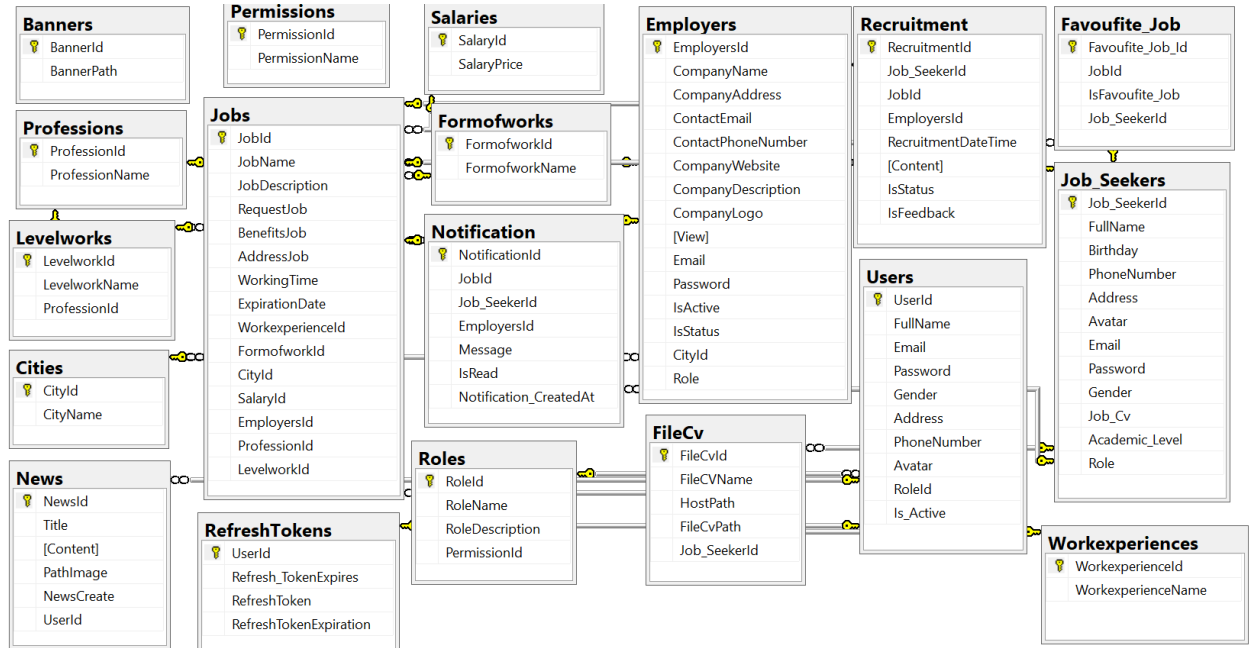
3.6.19. Bảng Users

Thuộc tính	Kiểu dữ liệu	Kích thước	Ràng buộc	Mô tả
------------	--------------	------------	-----------	-------

UserId	Long		Khóa chính	Mã quản trị viên
FullName	Nvarchar	255	Null	Tên quản trị
Email	Nvarchar	255	Not null	Email đăng nhập
Password	Nvarchar	255	Not null	Mật khẩu đăng nhập
Gender	Bit		Null	Giới tính
Address	Nvarchar	255	Null	Địa chỉ
PhoneNumber	Nvarchar	255	Null	Số điện thoại
Avatar	Nvarchar	Max	Null	Ảnh đại diện
RoleId	Long		Not null	Mã chức vụ

Bảng 3. 35. Bảng Users

3.7. Biểu đồ Database Diagram



Hình 3. 20. Biểu đồ Database Diagram

CHƯƠNG 4

TRIỂN KHAI VÀ KẾT QUẢ ĐẠT ĐƯỢC

4.1. Triển khai bài toán

4.1.1. Back-end

Khởi tạo một solution để quản lý cấu trúc dự án:

- *job_search_be.Domain*: Đây là nơi chứa các entity và các AutoMapper. Các entity sẽ đại diện cho các bảng trong cơ sở dữ liệu và các AutoMapper sẽ hỗ trợ việc ánh xạ giữa các model khác nhau. Trong dự án này, thư mục Domain sẽ không yêu cầu cài đặt bất kỳ package đặc biệt nào.
- *job_search_be.Application*: Thư mục này sẽ chứa các dịch vụ ứng dụng và các logic nghiệp vụ. Đây là nơi chúng ta xử lý các yêu cầu từ người dùng, thực hiện các quy trình nghiệp vụ và quản lý các tác vụ nghiệp vụ phức tạp.
- *job_search_be.Infrastructure*: Đây là nơi chứa các logic hạ tầng như truy cập dữ liệu và tích hợp bên ngoài. Để triển khai chức năng truy cập dữ liệu, chúng ta sẽ sử dụng Entity Framework Core và các package liên quan. Thư mục này sẽ chịu trách nhiệm giao tiếp với cơ sở dữ liệu và các dịch vụ bên ngoài, đảm bảo tính ổn định và hiệu quả của ứng dụng.
- *job_search_be.API*: Đây là dự án API, nơi triển khai các package liên quan đến ASP.NET Core, bao gồm cả Entity Framework Core. API sẽ là điểm tiếp xúc giữa người dùng và hệ thống, xử lý các yêu cầu HTTP và trả về các phản hồi tương ứng.

Xây dựng các trường dữ liệu cho ứng dụng, chúng ta sẽ sử dụng Entity Framework để định nghĩa các entity và cấu hình các mối quan hệ giữa chúng. Lưu trữ các entity trong SQL Server bằng code first. Xác thực thông tin người dùng bằng JWT, cần thêm middleware JWT vào pipeline của ứng dụng ASP.NET Core để xác thực JWT cho mỗi yêu cầu đến các API. Middleware sẽ giải mã và xác thực JWT và cung cấp thông tin xác thực của người dùng cho các controller và action tương ứng, quản lý thời gian hết hạn của JWT để đảm bảo tính bảo mật.

Sử dụng Postman để tạo và gửi các yêu cầu HTTP đến các API của mình. Tính năng quản lý môi trường của Postman cho phép dễ dàng thay đổi các giá trị như URL và thông tin xác

thực giữa môi trường phát triển, kiểm thử và sản xuất mà không cần phải thay đổi mã nguồn.

Công cụ sử dụng : Visual studio 2022, SQL Server 2022, Postman, Github

4.1.2. Front-end

Khởi tạo dự án bằng lệnh: `npm create vite@latest`. Tạo các thư mục để quản lý dự án:

- Common là nơi lưu chứa các module hoặc tập tin chứa các hàm tiện ích (utility functions) được sử dụng trong ứng dụng, như các hàm xử lý chuỗi, hàm tính toán hoặc hàm xử lý thời gian.
- Axios là nơi lưu trữ và tương tác với các API từ phía server.
- Router nơi quản lý đường dẫn, phụ trách phân quyền và middleware.
- Components giúp tổ chức các thành phần giao diện của ứng dụng một cách logic có cấu trúc và tái sử dụng nhiều lần.
- Layouts là nơi quản lý các giao diện chung của ứng dụng.

Công cụ sử dụng: Visual studio code, Github.

4.1.3. Tích hợp

a. Google Cloud Platform

Cấu hình OAuth 2.0: Sử dụng Cloud Identity Platform, bạn có thể cấu hình OAuth 2.0 để cho phép người dùng đăng nhập vào ứng dụng bằng tài khoản Google của họ. Bạn sẽ cần tạo một OAuth 2.0 client ID và client secret, và cung cấp các thông tin cần thiết như URI chuyển tiếp và phạm vi truy cập.

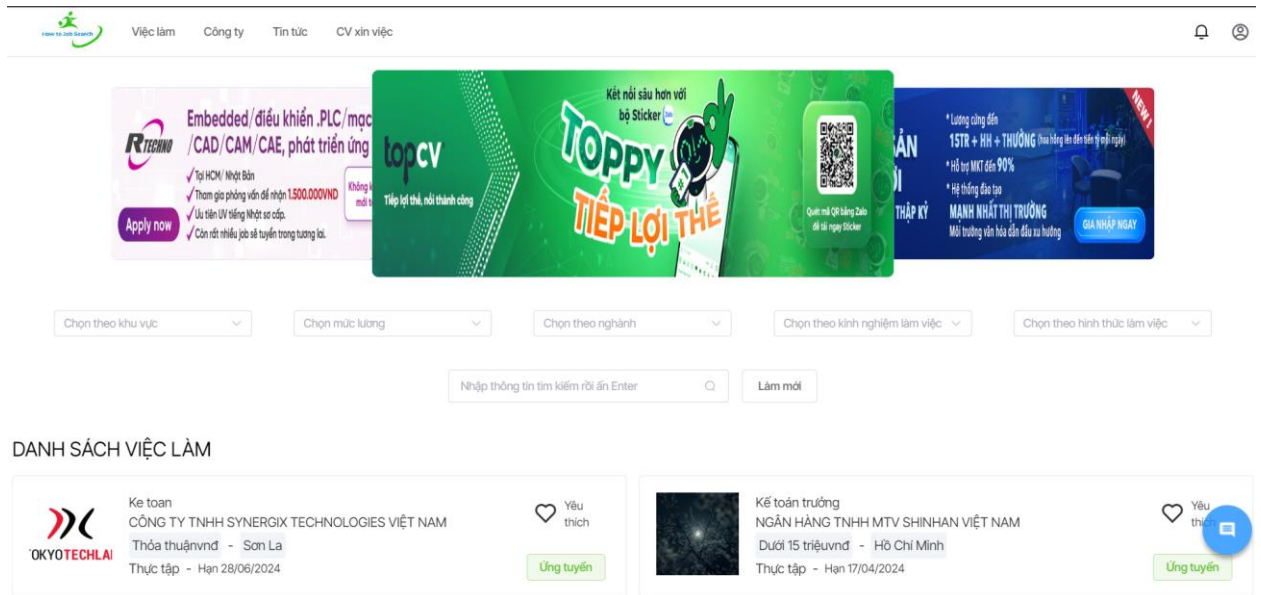
b. Rasa

Khởi tạo dự án rasa bằng lệnh: **rasa init**

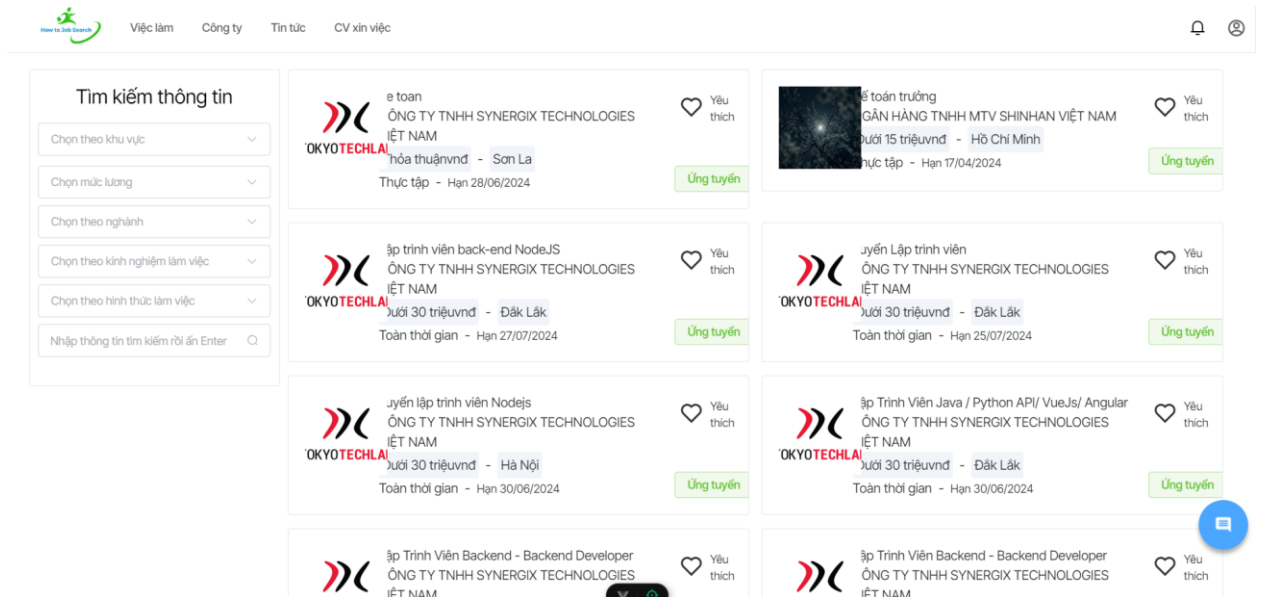
Xây dựng và huấn luyện mô hình Rasa bằng cách sử dụng lệnh: **rasa train**

Khi bạn đã huấn luyện xong mô hình, bạn có thể triển khai Agent Rasa bằng cách sử dụng lệnh sau: **rasa run -m models --enable-api --cors "*" .** Lệnh này sẽ khởi chạy một máy chủ web cục bộ và cho phép truy cập vào API của Agent Rasa từ các ứng dụng bên ngoài. Tích hợp Agent Rasa vào ứng dụng của mình bằng cách gửi các yêu cầu HTTP tới API của Agent. Ví dụ, bạn có thể sử dụng thư viện hoặc bất kỳ ngôn ngữ lập trình nào khác để gửi các yêu cầu đến API của Agent Rasa và nhận các phản hồi từ nó.

4.2. Kết quả đạt được



Hình 4. 1. Màn hình trang chủ



Hình 4. 2. Màn hình trang việc làm

The screenshot displays the website of Công ty TNHH Synergix Technologies Việt nam. The top navigation bar includes links for 'Việc làm' (Jobs), 'Công ty' (Company), 'Tin tức' (News), and 'CV xin việc' (Resume). The main content area is titled 'Thông tin tuyển dụng' (Job Information) and lists four job openings:

- Kế toán** (Accountant): CÔNG TY TNHH SYNERGIX TECHNOLOGIES VIỆT NAM. Thỏa thuận vnd - Sơn La. Thực tập - Hạn 28/06/2024.
- Lập trình viên back-end NodeJS** (Back-end NodeJS Developer): CÔNG TY TNHH SYNERGIX TECHNOLOGIES VIỆT NAM. Dưới 30 triệuvnd - Đắk Lắk. Toàn thời gian - Hạn 27/07/2024.
- Tuyển Lập trình viên** (Recruiting Developer): CÔNG TY TNHH SYNERGIX TECHNOLOGIES VIỆT NAM. Dưới 30 triệuvnd - Đắk Lắk. Toàn thời gian - Hạn 25/07/2024.
- Tuyển lập trình viên Nodejs** (Recruiting NodeJS Developer): CÔNG TY TNHH SYNERGIX TECHNOLOGIES VIỆT NAM. Dưới 30 triệuvnd - Hà Nội.

Each job listing includes a 'Yêu thích' (Like) button and an 'Ứng tuyển' (Apply) button. On the left, there is a 'Giới thiệu về công ty' (About the company) section and a 'Thông tin liên hệ' (Contact information) section with the address: Tầng 25, tháp C, tòa Central Point, 219 Trung Kính, Yên Hòa, Cầu Giấy, Hà Nội and phone number 0325472224.

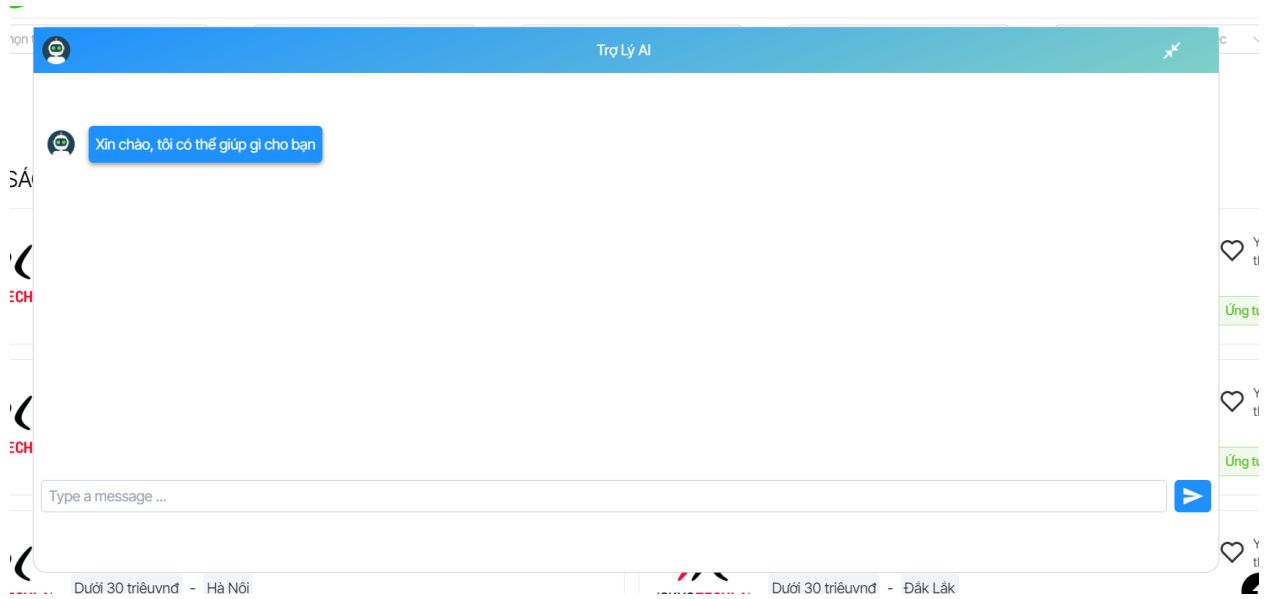
Hình 4. 3. Màn hình chi tiết công ty

The screenshot displays the website of Công ty TNHH Synergix Technologies Việt nam, focusing on the search and company listings section. The top navigation bar is identical to the previous screenshot. The main content area features a search bar with the placeholder text 'Tìm kiếm thông tin' (Search information) and a 'Nhập từ khóa rồi ấn Enter' (Enter keyword and press Enter) prompt. Below the search bar is a 'Chọn khu vực' (Select region) dropdown menu and a 'Làm mới' (Refresh) button. To the right, there are two company listings:

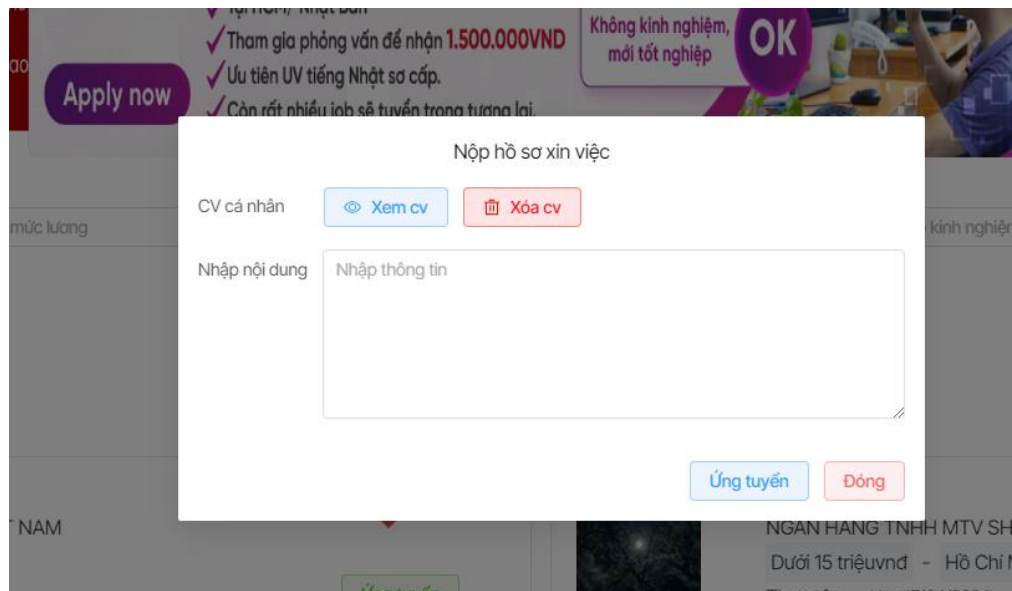
- Ngân Hàng TNHH MTV Shinhan Việt Nam** (Shinhan Bank Vietnam): Office Tower - Meco Complex - Ngõ 102 Trường Chi... 0325472224. Sơn La. @Xem chi tiết.
- Công ty TNHH Synergix Technologies Việt nam**: Tầng 25, tháp C, tòa Central Point, 219 Trung Kính, Y... 0325472224. Hà Nội. @Xem chi tiết.

Each company listing includes a profile picture, a 'Like' button, and a 'Xem chi tiết' (View details) button. A purple chat bubble icon is visible in the bottom right corner.

Hình 4. 4. Màn hình trang công ty



Hình 4. 5. Màn hình chat với trợ lý AI



Hình 4. 6. Màn hình ứng tuyển việc làm



Tải ảnh lên

Điền vị trí công việc...

Họ và tên Điền họ tên...

Giới tính Điền giới tính...

SĐT Điền số điện thoại...

Email Điền email...

Địa chỉ Điền địa chỉ...

Học vấn

Tên trường...

Nghành học...

Năm tốt nghiệp...


Mục tiêu nghề nghiệp

Điền nội dung...


Kinh nghiệm làm việc

Điền nội dung...

Hình 4. 7. Màn hình tạo hồ sơ/cv



[Việc làm](#)
[Công ty](#)
[Tin tức](#)
[CV xin việc](#)

Lập trình viên back-end NodeJS




Mức lương

Dưới 30 triệu



Khu vực

Đắk Lắk




Kinh nghiệm làm việc

Dưới 5 năm kinh nghiệm

Ứng tuyển công việc

Hạn nộp hồ sơ: 27/07/2024

 Chi tiết tuyển dụng


Mô tả công việc

Thời gian làm việc: Từ thứ 2 đến thứ 6, nghỉ Thứ 7 & Chủ Nhật. Đặc biệt bạn có thể tự chủ động giờ bắt đầu làm việc (đến trước 9h sáng), đến sớm về sớm. Miền đảm bảo đủ 8 tiếng/ngày và tiền độ dự án. Kết hợp linh hoạt giữa làm tại văn phòng và remote (tùy theo nguyện vọng của nhân viên) Mức lương cạnh tranh (tùy theo kỹ năng và kinh nghiệm) Review tăng lương 2 lần/năm Thường Tết, thưởng dự án Thường các ngày lễ: Tết Nguyên Đán, quà sinh nhật, Trung thu, 30/4, 2/9, 8/3, 20/10... Trợ cấp ăn trưa, cung cấp cung cấp trà, cà phê, trợ cấp đi lại, trợ cấp ngoại ngữ v.v.. Thanh toán OT theo quy định của luật lao động Có cơ hội đi onsite tại Tokyo hoặc Okinawa ngắn và dài hạn Bảo hiểm full lương Ngày phép: 12 ngày/năm và tăng dần theo thời gian làm việc Du lịch công ty hàng năm Khám sức khỏe định kỳ 01 lần/năm tại BV Quốc tế Hoạt động giải trí và Team Building phong phú (Cầu lạc bộ bóng đá, chạy bộ, đạp xe, PES...) Địa điểm làm việc: Tầng 10 Tòa nhà CMC, 11 Phố Duy Tân, Dịch Vọng Hậu, Cầu Giấy, Hà Nội.

Yêu cầu công việc

Thời gian làm việc: Từ thứ 2 đến thứ 6, nghỉ Thứ 7 & Chủ Nhật. Đặc biệt bạn có thể tự chủ động giờ bắt đầu làm việc (đến trước 9h sáng),

Việc làm liên quan




Tuyển Lập trình viên

CÔNG TY TNHH SYNERGIX TECHNOLOGIES VIỆT NAM

Dưới 30 triệuvnd - Đắk Lắk

Ứng tuyển




Lập Trình Viên Java / Python API/ Vue.js/ Angular

CÔNG TY TNHH SYNERGIX TECHNOLOGIES VIỆT NAM

Dưới 30 triệuvnd - Đắk Lắk

Ứng tuyển



Lập Trình Viên Backend - Backend Developer

CÔNG TY TNHH SYNERGIX TECHNOLOGIES VIỆT NAM

Dưới 30 triệuvnd - Đắk Lắk

Hình 4. 8. Màn hình chi tiết công việc



Đăng nhập

Nhập email

Nhập mật khẩu

[Quên mật khẩu?](#)

Đăng nhập

 Google

 Facebook

Nếu bạn chưa có tài khoản? [Vui lòng đăng ký](#)

Hình 4. 9. Màn hình đăng nhập



Đăng ký

Nhập họ và tên

Nhập email

Nhập mật khẩu

Nhập lại mật khẩu

Đăng ký

 Google

 Facebook

Nếu bạn có tài khoản? [Vui lòng đăng nhập](#)

Hình 4. 10. Màn hình đăng ký

Việc làm

Công ty

Tin tức

CV xin việc

Thay đổi ảnh đại diện

Thông tin tài khoản

Tên người dùng

Huy Phạm Khắc

Email

huyphk@thienhoang.com.vn

Ngày sinh

Chọn ngày sinh

Số điện thoại

Nhập thông tin

Giới tính

Chọn thông tin

Trình độ học vấn

Chọn thông tin

Địa chỉ hiện tại

Nhập thông tin

CV cá nhân

Xem cv

Xóa cv

Trạng thái tài khoản:

Đang hoạt động

Thay đổi thông tin

Lưu lại thông tin

Hình 4. 11. Màn hình thông tin tài khoản

Việc làm

Công ty

Tin tức

CV xin việc

Tin tức tuyển dụng

Lập Trình Viên Frontend

Công ty: Công ty ABC

Địa điểm: Hà Nội

Mô tả: Miêu tả công việc lập trình viên frontend.

Lập Trình Viên Backend

Công ty: Công ty XYZ

Địa điểm: TP.HCM

Mô tả: Miêu tả công việc lập trình viên backend.

Thiết Kế Giao Diện UI/UX

Công ty: Công ty 123

Địa điểm: Đà Nẵng

Mô tả: Miêu tả công việc thiết kế giao diện UI/UX.

Hỗ trợ và tìm kiếm việc làm

f

in

VỀ CHÚNG TÔI

Công ty

Nghề nghiệp

Blog

HỖ TRỢ

Liên hệ hỗ trợ

Tài nguyên hỗ trợ

Hỗ trợ viết CV

NỀN TẢNG

Điều khoản & Quyền riêng tư

FAQ

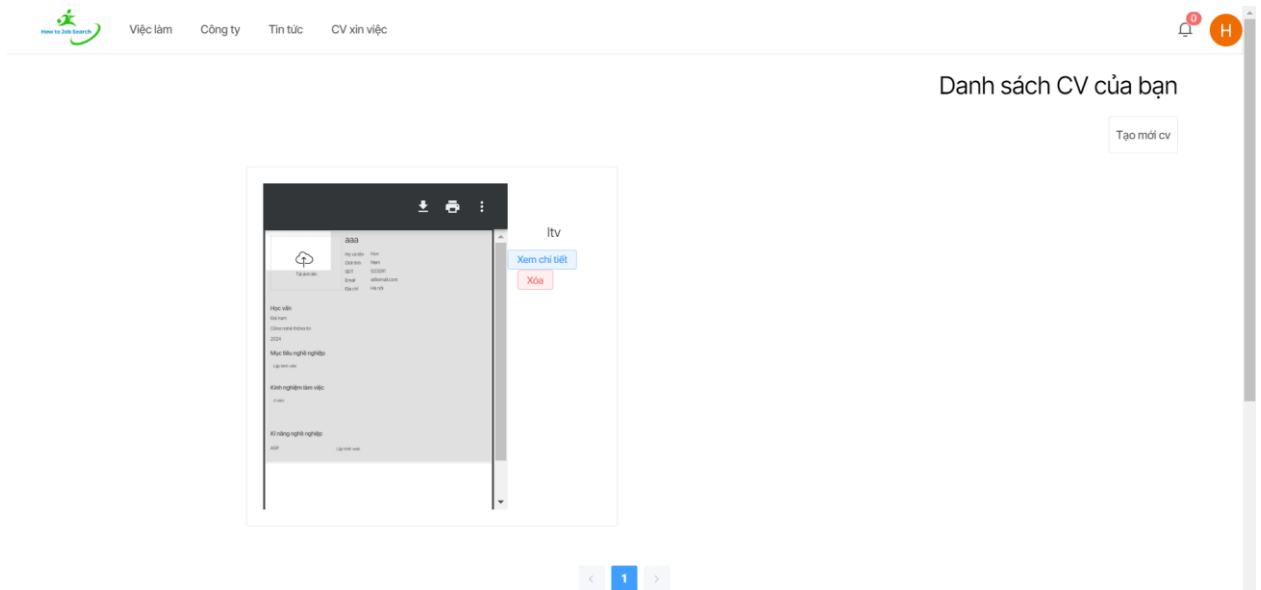
LIÊN HỆ

Gửi tin nhắn

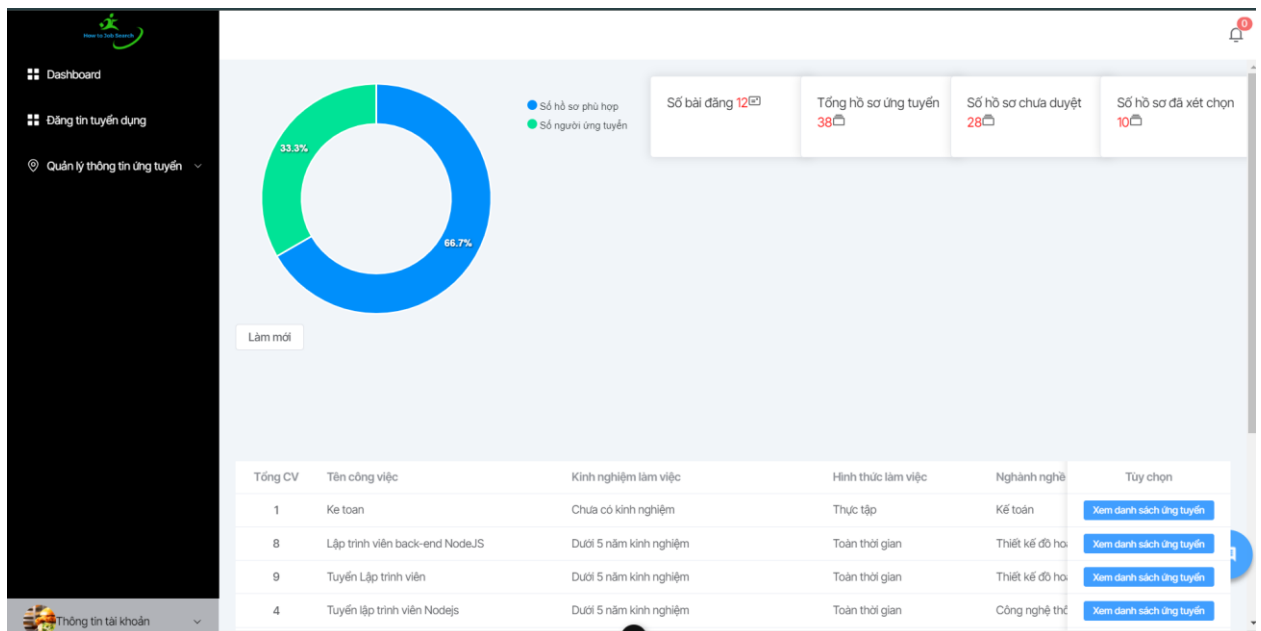
+84-325-472-224

© 2024 Phạm Khắc Huy, Công Nghệ Thông Tin - Đại Học Đại Nam

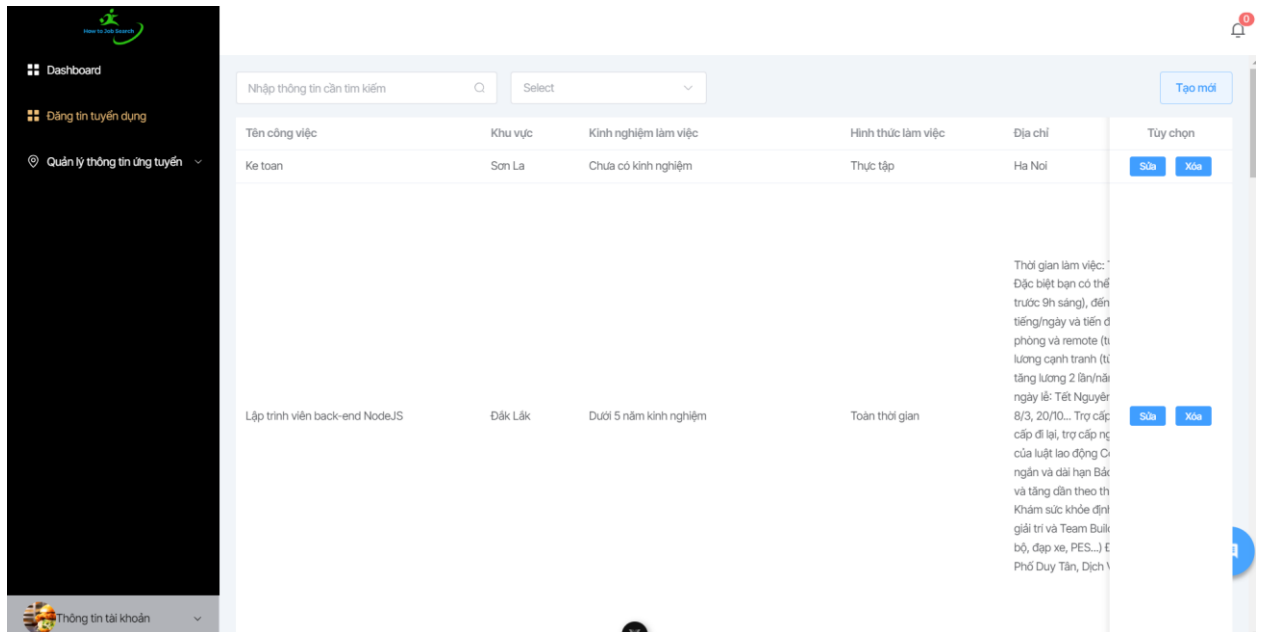
Hình 4. 12. Màn hình tin tức



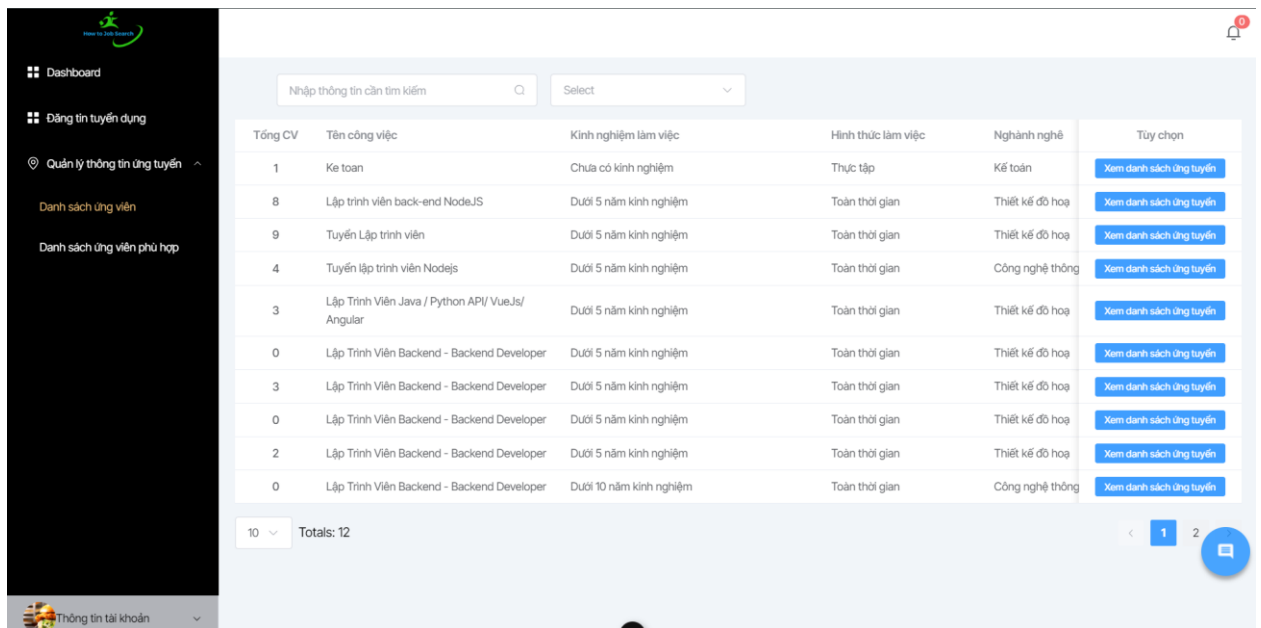
Hình 4. 13. Màn hình quản lý hồ sơ/cv



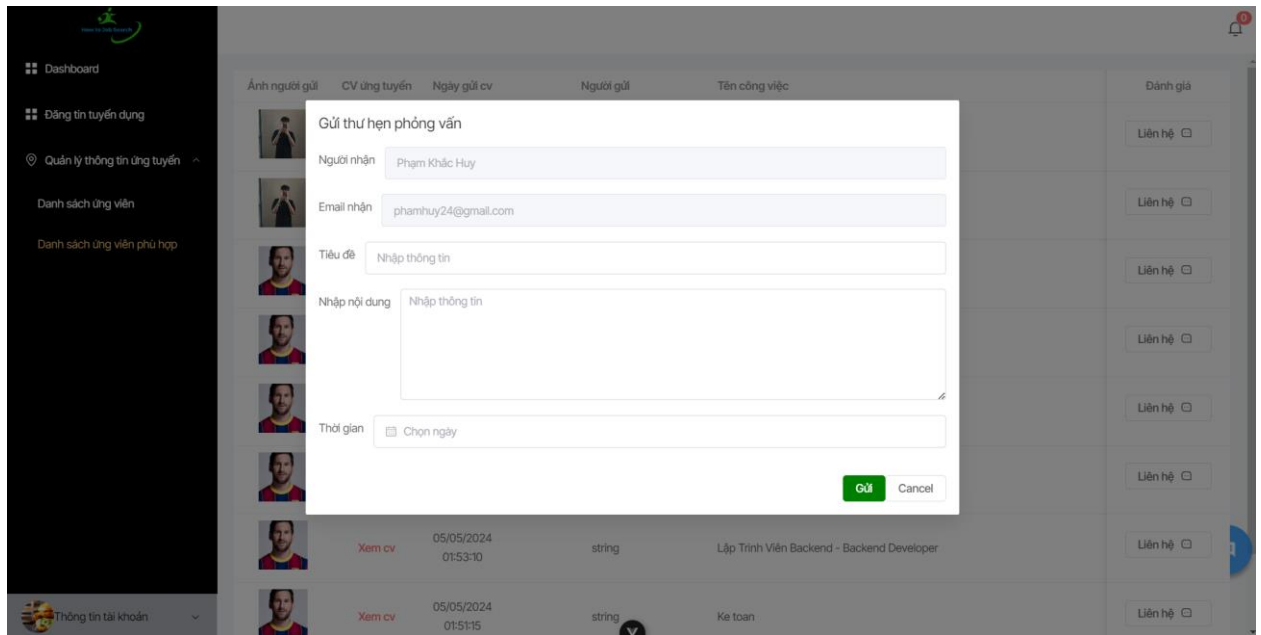
Hình 4. 14. Màn hình dashboard nhà tuyển dụng



Hình 4.15. Màn hình đăng tin tuyển dụng



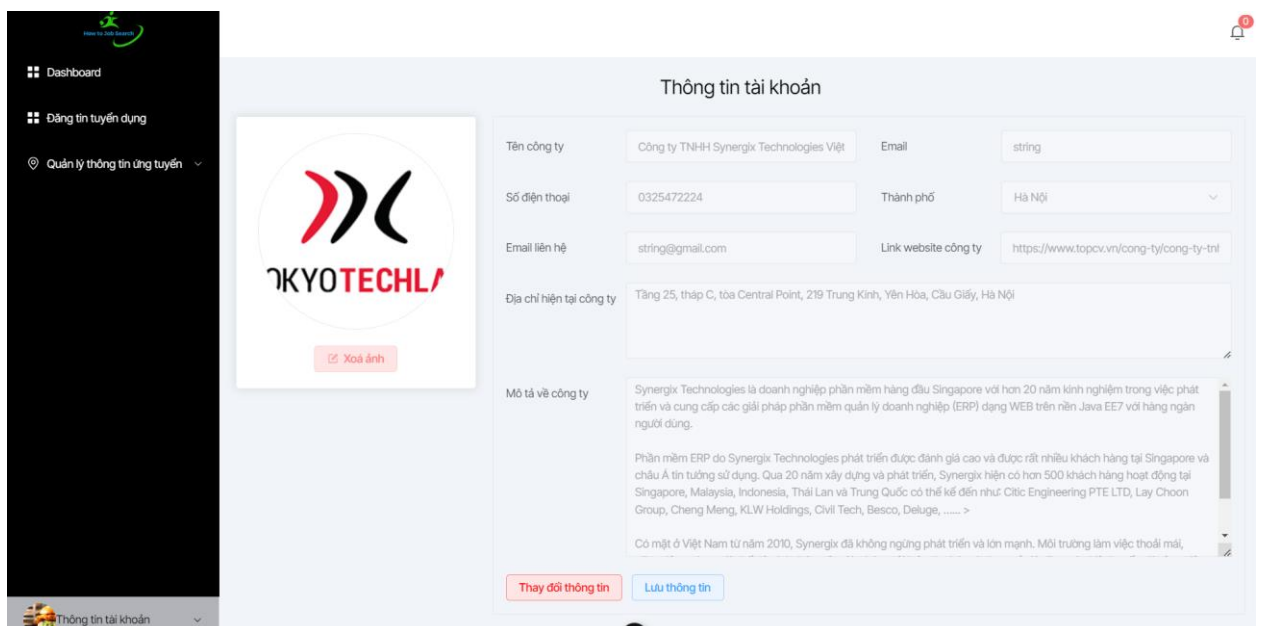
Hình 4.16. Màn hình danh sách ứng viên



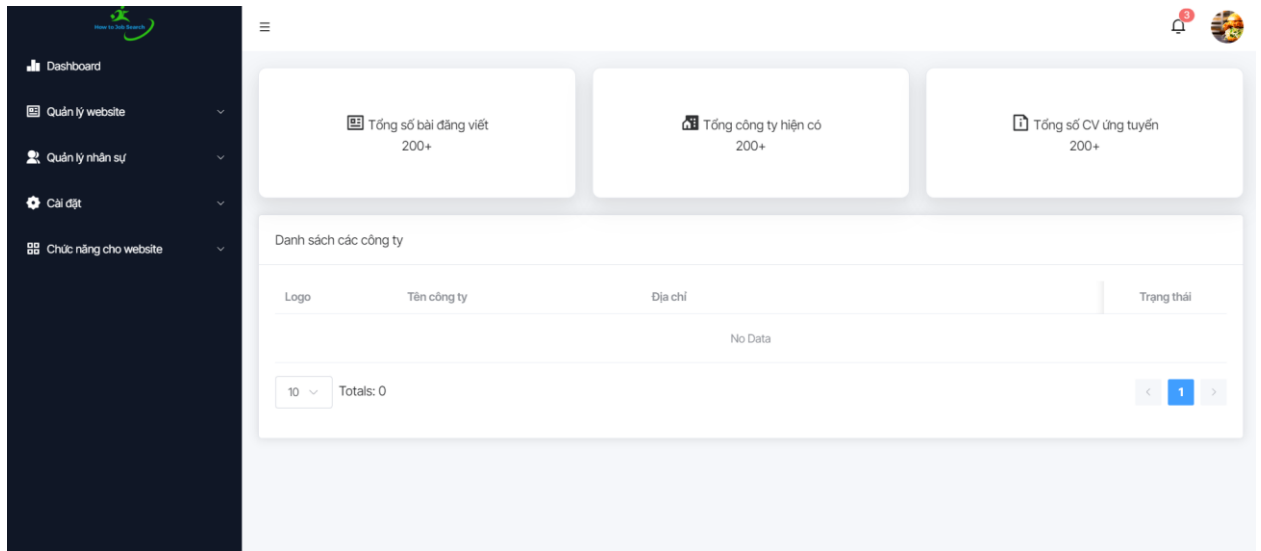
Hình 4. 17. Màn hình hồ sơ/cv phù hợp và gửi thư hẹn phỏng vấn



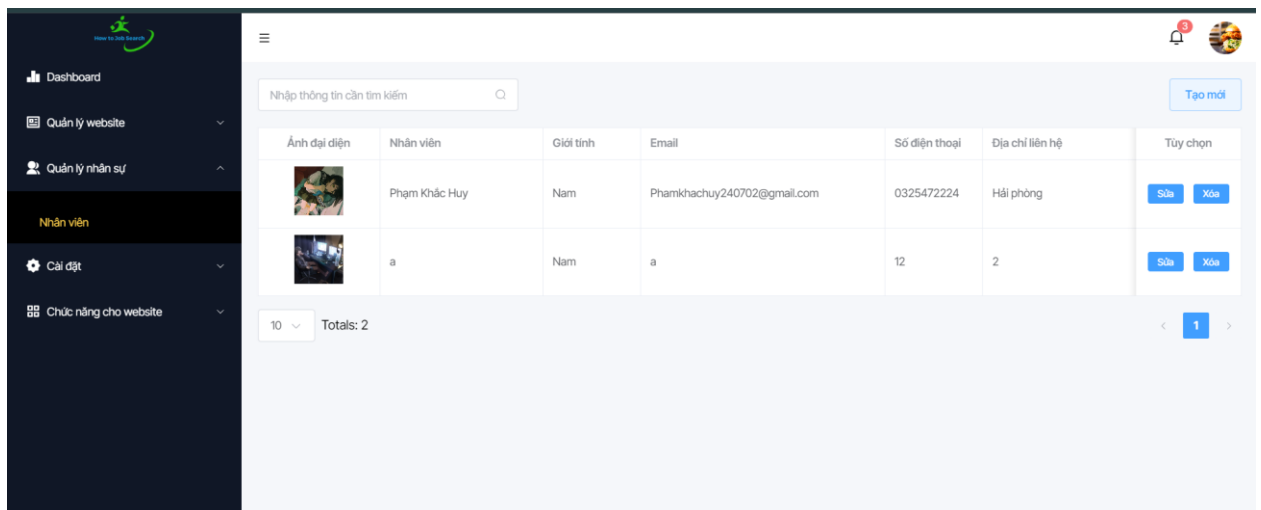
Hình 4. 18. Màn hình email sau khi gửi cho người tìm việc



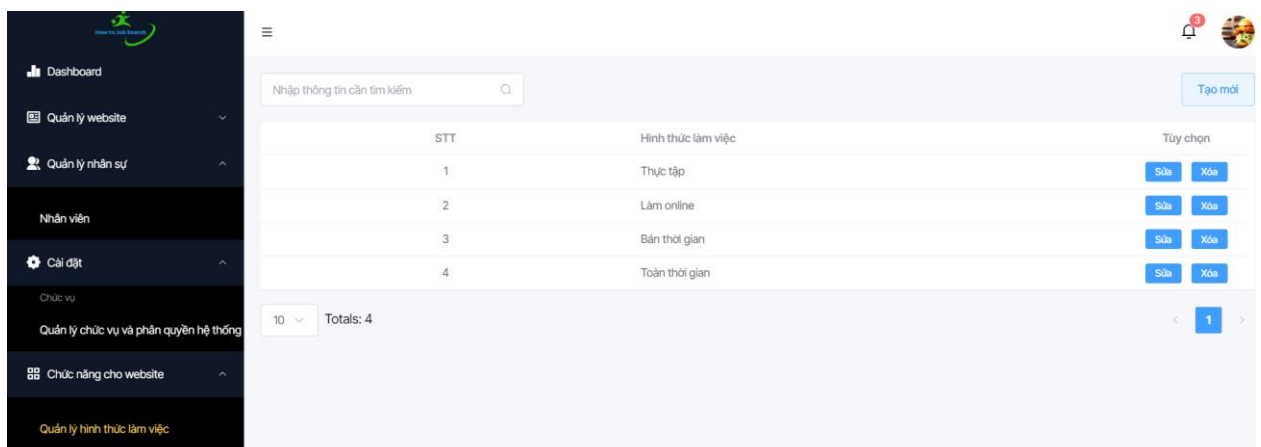
Hình 4. 19. Màn hình thông tin tài khoản nhà tuyển dụng



Hình 4. 20. Màn hình dashboard quản trị viên



Hình 4. 21. Màn hình quản lý nhân viên



Hình 4. 22. Màn hình quản lý hình thức làm việc

STT	Kinh nghiệm làm việc	Tùy chọn
1	Dưới 10 năm kinh nghiệm	Sửa Xóa
2	Trên 10 năm kinh nghiệm	Sửa Xóa
3	Dưới 5 năm kinh nghiệm	Sửa Xóa
4	Trên 5 năm kinh nghiệm	Sửa Xóa
5	3-5 năm kinh nghiệm	Sửa Xóa
6	Chưa có kinh nghiệm	Sửa Xóa
7	1-2 năm kinh nghiệm	Sửa Xóa

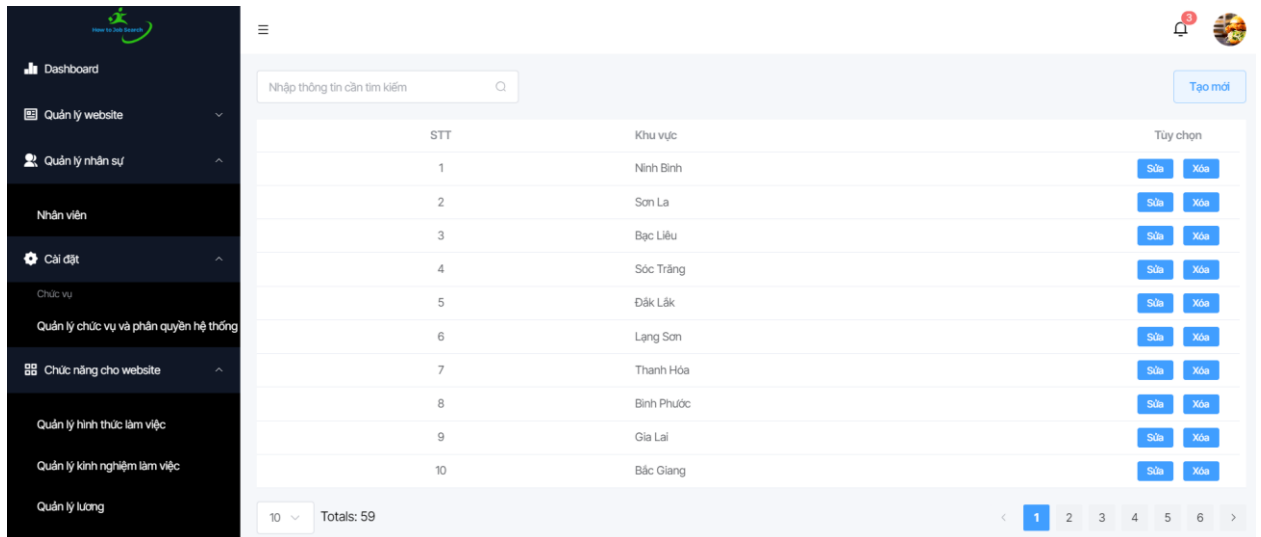
Hình 4. 23. Màn hình quản lý kinh nghiệm làm việc

STT	Mức lương	Tùy chọn
1	Thỏa thuận	Sửa Xóa
2	Dưới 15 triệu	Sửa Xóa
3	Dưới 20 triệu	Sửa Xóa
4	Dưới 30 triệu	Sửa Xóa
5	Trên 30 triệu	Sửa Xóa
6	Trên 20 triệu	Sửa Xóa
7	Trên 15 triệu	Sửa Xóa
8	Trên 10 triệu	Sửa Xóa
9	Dưới 5 triệu	Sửa Xóa
10	4.000.000 - 5.000.000	Sửa Xóa

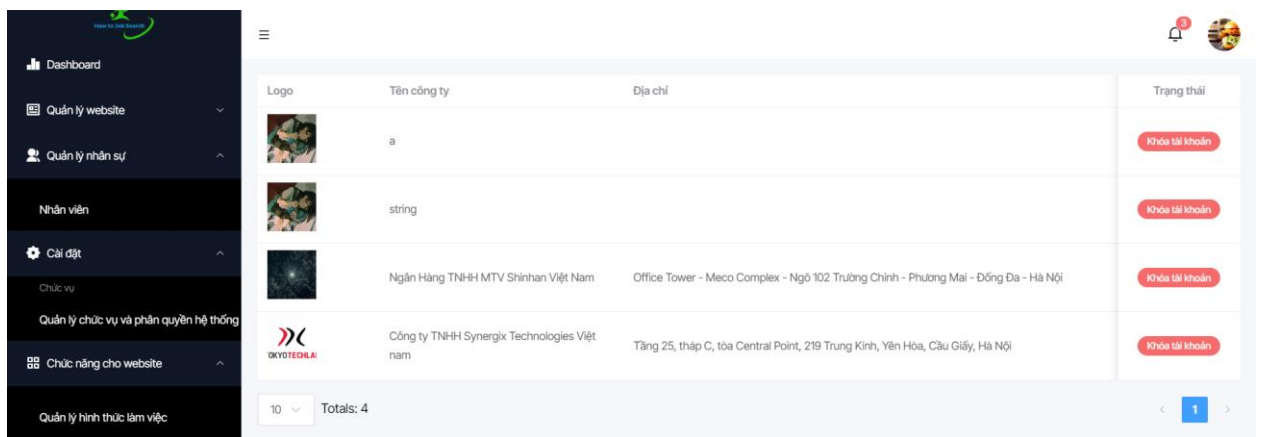
Hình 4. 24. Màn hình quản lý lương

STT	Ngành nghề	Tùy chọn
1	Ngân hàng	Chức năng
2	Kế toán	Chức năng
3	Công nghệ thông tin	Chức năng
4	Thiết kế đồ họa	Chức năng

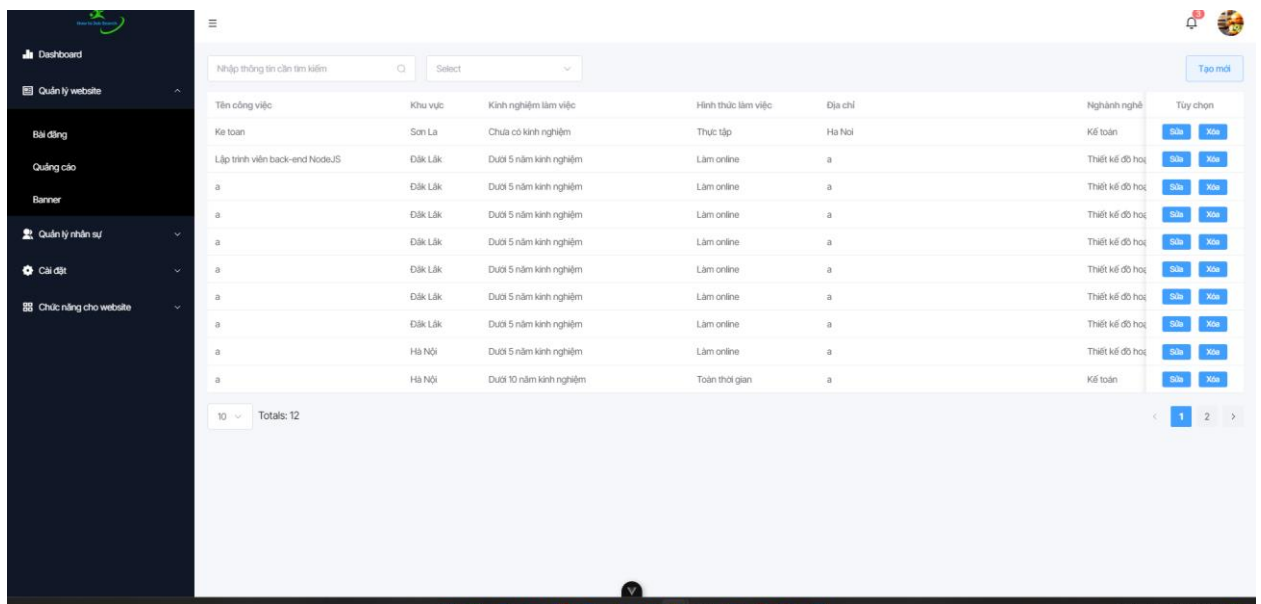
Hình 4. 25. Màn hình quản lý thông tin ngành nghề



Hình 4. 26. Màn hình quản lý khu vực



Hình 4. 27. Màn hình quản lý thông tin nhà tuyển dụng



Hình 4. 28. Màn hình quản lý bài đăng tin tức

KẾT LUẬN

Kết quả đạt được của đồ án là sự hoàn thành một loạt các mục tiêu đã được đề ra một cách xuất sắc. Hệ thống tìm kiếm việc làm đã chứng minh sức mạnh của mình trong việc giúp cả người tìm việc và nhà tuyển dụng tìm ra sự kết hợp lý tưởng giữa nhu cầu và nguồn lực nhân sự. Đầu tiên, với người tìm việc, giao diện thân thiện và tính năng tìm kiếm tiên tiến đã giúp họ dễ dàng tìm kiếm các vị trí phù hợp với kỹ năng và mục tiêu nghề nghiệp của mình. Qua việc tạo hồ sơ trực tuyến và ứng tuyển một cách thuận tiện, họ có thể nhanh chóng tiếp cận với các cơ hội mới và nâng cao khả năng thành công trong quá trình tìm việc.

Một phần quan trọng khác của thành công là sự tiện lợi và hiệu quả mà hệ thống mang lại cho nhà tuyển dụng. Họ không chỉ có thể dễ dàng đăng các vị trí làm việc lên hệ thống mà còn có thể quản lý các mốc thời gian tuyển dụng một cách linh hoạt và hiệu quả. Hệ thống cũng hỗ trợ trong việc gửi mail tự ứng viên, giúp tiết kiệm thời gian và nỗ lực cho quá trình tuyển dụng. Sự tương tác mạnh mẽ giữa người tìm việc và nhà tuyển dụng đã được thúc đẩy thông qua giao diện trực quan và tính năng thông báo tự động, tạo ra một môi trường làm việc tích cực và hiệu quả.

Trong hướng phát triển, có thể tăng cường khả năng kết nối bằng cách bổ sung các tính năng mới. Tích hợp thêm chức năng tính và thu phí đối với các tài khoản của nhà tuyển dụng. Gợi ý việc làm hàng ngày qua email cũng sẽ giúp người dùng cập nhật thông tin nhanh chóng và hiệu quả. Hơn nữa, việc thiết lập một cơ chế liên hệ trực tiếp giữa nhà tuyển dụng và ứng viên thông qua trang web sẽ tạo ra một môi trường tương tác động và sâu sắc hơn. Điều này sẽ nâng cao trải nghiệm người dùng và tăng cường sự hài lòng từ cả hai phía, góp phần vào sự phát triển bền vững của hệ thống tìm kiếm việc làm này.

TÀI LIỆU THAM KHẢO

Danh mục các Website tham khảo:

- [1]. <https://vuejs.org/>
- [2]. <https://kb.pavietnam.vn/kien-thuc-co-ban-ve-asp-net-core.html>
- [3]. <https://viblo.asia/p/tim-hieu-kien-thuc-co-ban-ve-api-maGK7A4Mlj2>
- [4]. <https://viblo.asia/p/tim-hieu-ve-json-web-token-jwt-7rVRqp73v4bP>
- [5]. <https://www.typescriptlang.org/>