

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI:
XÂY DỰNG WEBSITE DỰ BÁO XU HƯỚNG VIỆC LÀM

Người hướng dẫn: PGS.TS. Nguyễn Tấn Khôi

Sinh viên thực hiện: Đỗ Cao Cường

Số thẻ sinh viên: 102210309

Lớp: 21TCLC_KHDL2

Đà Nẵng, 6/2025

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI:
XÂY DỰNG WEBSITE DỰ BÁO XU HƯỚNG VIỆC LÀM

Người hướng dẫn: PGS.TS. Nguyễn Tấn Khôi

Sinh viên thực hiện: Đỗ Cao Cường

Số thẻ sinh viên: 102210309

Lớp: 21TCLC_KHDL2

Đà Nẵng, 6/2025

TÓM TẮT

Tên đề tài: Xây dựng Website dự báo xu hướng việc làm

Sinh viên thực hiện: Đỗ Cao Cường

MSSV: 102210309

Lớp SH: 21TCLC_KHDL2

Đề tài "Xây dựng Website dự báo xu hướng việc làm" nhằm liệt kê, phân tích và đưa cho người dùng những công việc phù hợp với bản thân. Ngoài ra website còn tích hợp các tính năng ứng dụng AI để dự đoán xu hướng công việc, so sánh các công việc để tìm ra công việc phù hợp.

Nội dung dự án bao gồm 4 chương:

- Chương 1: Cơ sở lý thuyết
- Chương 2: Hệ thống phân tích và thiết kế
- Chương 3: Ứng dụng học máy
- Chương 4: Thiết kế Chatbot AI
- Chương 5: Kết quả thực hiện và đánh giá

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Đỗ Cao Cường

Số thẻ sinh viên: 102210309

Lớp: 21TCLC_KHDL2

Khoa: Công nghệ Thông tin

Ngành: Khoa học dữ liệu và Trí tuệ nhân tạo

1. Tên đề tài đồ án:

Xây dựng Website dự báo xu hướng việc làm

2. Đề tài thuộc diện: Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu: Không có.

4. Nội dung các phần thuyết minh và tính toán:

Nội dung các phần thuyết minh bao gồm:

- Mở đầu: Đưa ra giới thiệu chung về đề tài, mục tiêu, đối tượng và bảng kế hoạch thực hiện.

- Chương 1 - Trình bày về cơ sở lý thuyết

- Chương 2 - Trình bày về hệ thống phân tích và thiết kế

- Chương 3 - Trình bày ứng dụng học máy

- Chương 4 – Trình bày về thiết kế Chatbot AI

- Chương 5 - Trình bày về kết quả thực hiện và đánh giá

- Kết luận: Tổng kết những kết quả đạt được cũng như các hạn chế và trình bày một số hướng phát triển cho hệ thống.

5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):

Biểu đồ usecase, sơ đồ hoạt động, sơ đồ tuần tự

6. Họ tên người hướng dẫn: PGS.TS. Nguyễn Tấn Khôi

7. Doanh nghiệp hướng dẫn:

8. Ngày giao nhiệm vụ đồ án: / / 2025

9. Ngày hoàn thành đồ án: / / 2025

Đà Nẵng, ngày.....tháng.....năm 2025

Trưởng Bộ môn

Người hướng dẫn

LỜI NÓI ĐẦU

Trong suốt quá trình học tập tại trường Đại học Bách Khoa - Đại học Đà Nẵng và thời gian thực hiện đồ án này, tôi đã nhận được sự giúp đỡ và hướng dẫn tận tình từ các Thầy, Cô trong khoa Công Nghệ Thông Tin. Tôi xin bày tỏ lòng biết ơn chân thành đến tất cả các Thầy, Cô đã truyền đạt kiến thức chuyên môn và động viên tinh thần học tập, giúp tôi có nền tảng để thực hiện đồ án tốt nghiệp. Việc thực hiện đề tài này không chỉ là một cơ hội để tôi áp dụng những kiến thức đã học vào thực tế mà còn là một thách thức, đòi hỏi sự nghiên cứu, sáng tạo và nỗ lực không ngừng.

Đặc biệt, tôi xin gửi lời cảm ơn sâu sắc đến PGS.TS. Nguyễn Tấn Khôi, người đã tận tâm hướng dẫn, giúp tôi hiểu rõ mục tiêu của đồ án và đưa ra những lời khuyên quý báu trong việc lựa chọn đề tài. Thầy đã dành thời gian tổ chức các buổi gặp gỡ để thảo luận và giải đáp thắc mắc, luôn đồng hành và hỗ trợ tôi trong suốt quá trình thực hiện đồ án.

Để có thể đạt được thành quả ngày hôm nay, tôi rất biết ơn gia đình đã tạo điều kiện, khích lệ và động viên tôi tiếp tục học tập. Cảm ơn những người bạn, anh chị tôi đã luôn đồng hành và hỗ trợ tôi trong suốt quá trình học tập.

Trong quá trình học tập và thực hiện đồ án, không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được những góp ý thẳng thắn từ các Thầy, Cô để có thể hoàn thiện hơn kết quả của mình và tích lũy thêm kinh nghiệm cho công việc sau này.

Một lần nữa, tôi xin chân thành cảm ơn!

LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT

Tôi xin cam đoan Đồ án tốt nghiệp có đề tài: “Xây dựng Website dự báo xu hướng việc làm” là công trình nghiên cứu của chính cá nhân tôi dưới sự hướng dẫn trực tiếp của PGS.TS.Nguyễn Tấn Khôi. Tôi đã tự đọc, nghiên cứu, dịch tài liệu và tổng hợp các kiến thức đã làm nên báo cáo này và đảm bảo không sao chép ở bất cứ đâu. Những lý thuyết trong báo cáo đều được sử dụng tài liệu như tôi đã tham khảo ở phần tài liệu tham khảo đã có trong báo cáo.

Tôi xin chịu hoàn toàn trách nhiệm và mọi hình thức kỉ luật nếu như vi phạm liêm chính học thuật từ phía khoa và nhà trường.

Sinh viên thực hiện

Đỗ Cao Cường

MỤC LỤC

LỜI NÓI ĐẦU	i
LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT	ii
MỤC LỤC	iii
DANH SÁCH HÌNH ẢNH.....	vii
DANH SÁCH BẢNG.....	ix
DANH SÁCH CHỮ VIẾT TẮT	x
MỞ ĐẦU.....	1
1. Lý do chọn đề tài.....	1
2. Mục tiêu	1
3. Các bước thực hiện	2
4. Công cụ và môi trường phát triển.....	2
CHƯƠNG I. CƠ SỞ LÝ THUYẾT.....	3
1. Tính cấp thiết của đề tài	3
2. Tổng quan về giải pháp kỹ thuật.....	3
2.1. Lập kế hoạch và thiết kế	3
2.2. Thiết lập môi trường phát triển.....	3
2.3. FrontEnd với NextJs	3
2.4. BackEnd với FastAPI	3
2.5. Kết nối FrontEnd và BackEnd.....	4
2.6. Kiểm tra và gỡ lỗi	4
2.7. Giám sát và bảo trì	4
3. Lý thuyết và công nghệ.....	4
3.1. HTML	4
3.2. CSS	5
3.3. JavaScript.....	5
3.4. MongoDB	6
3.5. API RESTful.....	7
3.6. Tổng quan về NextJs.....	8
3.6.1. Next.js.....	8
3.6.2. Các tính năng của Next.js	8

3.6.3. Tại sao sử dụng Next.js	8
3.6.4. API Routes.....	9
3.6.5. Next.js với Redux	9
3.7. FastAPI	9
3.7.1. Đặc điểm của FastAPI	9
3.7.2. Lợi ích khi sử dụng FastAPI.....	10
3.7.3. Một số ứng dụng sử dụng FastAPI	10
3.8. Python	10
3.9. Docker.....	12
3.10. Github	13
3.11. Visual Code.....	13
3.12. Postman.....	14
3.13. Google Gemini.....	14
CHƯƠNG II. HỆ THỐNG PHÂN TÍCH VÀ THIẾT KẾ	15
1. Phân tích yêu cầu chức năng	15
1.1. Khách	15
1.2. Người sử dụng	15
1.3. Người quản trị.....	15
2. Biểu đồ Usecase	16
2.1. Biểu đồ Usecase tổng quát.....	16
2.2. Sơ đồ Usecase cho khách.....	17
2.2.1. Đăng ký.....	17
2.2.2. Tìm kiếm việc làm.....	18
2.3. Sơ đồ Usecase cho người dùng.....	19
2.3.1. Đăng nhập.....	19
2.3.2. AI Dự đoán xu hướng việc làm	20
2.3.3. Quản lý tài khoản.....	21
2.4. Sơ đồ Usecase cho Quản trị viên	22
2.4.1. Quản lý danh sách việc làm	22
2.4.2. Quản lý người dùng	22
2.4.3. Crawl data.....	23
3. Sơ đồ hoạt động.....	24
3.1. Chức năng “Đăng ký”	24
3.2. Chức năng “Đăng nhập”	25

3.3. Chức năng “Crawl data”	25
3.4. Chức năng dự đoán xu hướng việc làm	26
4. Sơ đồ tuần tự	27
4.1. Chức năng đăng ký người dùng	27
4.2. Chức năng đăng nhập.....	28
4.3. Chức năng thu thập dữ liệu	28
4.4. Chức năng dự đoán xu hướng việc làm	28
5. Thiết kế cơ sở dữ liệu.....	30
6. Các bảng trong cơ sở dữ liệu	30
7. Đặc tả các usecase	33
7.1. Đặc tả UC đăng ký.....	33
7.2. Đặc tả UC đăng nhập.....	33
7.3. Đặc tả UC tìm kiếm việc làm.....	34
7.4. Đặc tả UC quản lý người dùng	34
7.5. Đặc tả UC quản lý việc làm	35
7.6. Đặc tả UC Crawl data	35
7.7. Đặc tả UC dự đoán xu hướng việc làm.....	36
7.8. Đặc tả UC quản lý thông tin cá nhân	36
CHƯƠNG III. ỨNG DỤNG HỌC MÁY ĐỂ DỰ BÁO	37
1. Phát biểu vấn đề.....	37
2. Thuật toán	37
2.1. Long short-term memory (LSTM).....	37
2.2. Cấu trúc và hoạt động của LSTM.....	38
2.3. Thuật toán Transformer	39
2.4. Cấu trúc và hoạt động của Transformer.....	40
3. Áp dụng học máy	41
3.1. Thực hiện và đánh giá.....	41
3.1.1. Thu thập và xử lý dữ liệu.....	41
3.1.2. Xây dựng mô hình Transformer	43
3.2. Chia dữ liệu.....	45
3.3. Dự đoán.....	45
3.4. Kết quả.....	47
3.4.1. Thang đo MAE	47

3.4.2. Thang đo MSE	48
3.4.3. Thang đo F1 Score.....	48
3.4.4. Kết quả của mô hình	48
CHƯƠNG IV. Thiết kế Chatbot AI	50
1. Mục tiêu	50
2. Kiến trúc tổng quan.....	50
3. Pipeline hoạt động.....	51
CHƯƠNG V. KẾT QUẢ THỰC HIỆN VÀ ĐÁNH GIÁ.....	52
1. Triển khai	52
2. Mô hình hệ thống	52
3. Kết quả thực hiện.....	52
3.1. Trang đăng nhập	52
3.2. Trang đăng ký	53
3.3. Trang chủ	53
3.4. Thiết lập người dùng.....	54
3.5. Chatbot.....	54
3.6. Trang chi tiết công việc	55
3.7. Trang dự đoán xu hướng việc làm	55
3.8. Quản lý.....	56
3.9. Quản lý đơn ứng tuyển	56
3.10. Quản lý đơn ứng tuyển	57
4. Kết chương	57
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	58
1. Kết quả đạt được.....	58
2. Các vấn đề tồn tại	58
3. Hướng phát triển	58
TÀI LIỆU THAM KHẢO.....	59

DANH SÁCH HÌNH ẢNH

Hình 1: Kết hợp trang web	5
Hình 2: MongoDB	6
Hình 3: Mô hình Rest API	7
Hình 4: Framework Next.js	8
Hình 5: Framework FastAPI.....	9
Hình 6: Python	11
Hình 7: Docker	12
Hình 8: Github	13
Hình 9: Postman	14
Hình 10: Biểu đồ UC tổng quát	16
Hình 11: Sơ đồ Usecase “Đăng ký”	17
Hình 12: Sơ đồ Usecase “Tìm kiếm việc làm”	18
Hình 13: Sơ đồ Usecase “Đăng nhập”	19
Hình 14: Sơ đồ Usecase “Dự báo xu hướng”	20
Hình 15: Sơ đồ Usecase “Quản lý tài khoản”	21
Hình 16: Sơ đồ Usecase “Quả lý danh sách việc làm”	22
Hình 17: Sơ đồ Usecase “Quản lý người dùng”	22
Hình 18: Sơ đồ Usecase “Crawl data”	23
Hình 19: Sơ đồ hoạt động chức “năng đăng ký người dùng”	24
Hình 20: Sơ đồ hoạt động chức năng “đăng nhập”	25
Hình 21: Sơ đồ hoạt động chức năng “Crawl data”	25
Hình 22: Sơ đồ hoạt động chức năng “Dự đoán xu hướng việc làm”	26
Hình 23: Sơ đồ chức năng “Đăng ký”	27
Hình 24: Sơ đồ tuần tự chức năng “Đăng nhập”	28
Hình 25: Sơ đồ tuần tự chức năng “Crawl Data”	28
Hình 26: Sơ đồ tuần tự chức năng” Dự đoán xu hướng việc làm”	29
Hình 27: Cơ sở dữ liệu.....	30
Hình 28: Sơ đồ triển khai mô hình học máy	37
Hình 29: Cấu trúc của một LSTM	38
Hình 30: Mô hình Transformer.....	39
Hình 31: Mô hình thuật toán Transformer	40
Hình 32: Bộ dữ liệu chuẩn bị cho mô hình.....	42
Hình 33: Xử lý dữ liệu	42
Hình 34: Dữ liệu sau khi xử lý	42

Hình 35: Dữ liệu của "NodeJs_Đà Nẵng"	43
Hình 36: Phân tích mùa vụ và xu hướng	43
Hình 37: Giá trị trend của " Backend Developer_Đà Nẵng"	44
Hình 38: Mô hình LSTM.....	44
Hình 39: Dự đoán xu hướng	45
Hình 40: Kết quả thu được.....	46
Hình 41: Đánh giá độ chính xác trên tập train.....	46
Hình 42: Dự đoán và độ chính xác trên tập train của “Data Scientist – Đà Nẵng”	46
Hình 43: Đánh giá độ chính xác trên tập test.....	47
Hình 44: Dự đoán và độ chính xác trên tập test của “Nodejs – Hà Nội”	47
Hình 45: Công thức MAE.....	47
Hình 46: Công thức MSE	48
Hình 47: Công thức F1 Score	48
Hình 48: Kết quả thang đo của 2 mô hình	48
Hình 49: Kiểm thử với tập test	48
Hình 50: Kiến trúc RAG.....	50
Hình 51: Khai triển cấu trúc API.....	52
Hình 52: Trang đăng nhập	53
Hình 53: Trang đăng kí.....	53
Hình 54: Trang chủ.....	54
Hình 55: Quản lí đăng kí	54
Hình 56: Chatbot tư vấn cho người dùng	55
Hình 57: Trang đăng tin tuyển dụng.....	55
Hình 58: Dự đoán xu hướng việc làm cho “Backend Developer _Hà Nội”	56
Hình 59: Trang quản lý người dùng	56
Hình 60: Trang quản lý jobs	57
Hình 61: Quản lý đơn ứng tuyển	57

DANH SÁCH BẢNG

Bảng 1: Table users	30
Bảng 2: Table users skill.....	31
Bảng 3: Table job embeddings	31
Bảng 4: Table craw data	31
Bảng 5: Table major catefory	32
Bảng 6: Table mail schedule.....	32
Bảng 7: Đặc tả UC đăng ký	33
Bảng 8: Đặc tả UC đăng nhập	33
Bảng 9: Đặc tả UC tìm kiếm việc làm	34
Bảng 10: Đặc tả UC quản lý người dùng.....	34
Bảng 11: Đặc tả UC quản lý việc làm	35
Bảng 12: Đặc tả UC Crawl data	35
Bảng 13: Đặc tả UC dự đoán xu hướng việc làm	36
Bảng 14: Đặc tả UC quản lý thông tin cá nhân	36
Bảng 15: So sánh LSTM và Transformer.....	41

DANH SÁCH CHỮ VIẾT TẮT

STT	Chữ viết tắt	Diễn giải
1	UI	User Interface
2	HTML	Hypertext Markup Language
3	CSS	Cascading Style Sheets
4	URL	Uniform Resource Locator
5	API	Application Programming Interface
6	CSDL	Cơ sở dữ liệu
7	Trí tuệ nhân tạo	Trí tuệ nhân tạo
8	RNN	Mạng nơ-ron hồi quy
9	LSTM	Bộ nhớ dài hạn ngắn hạn

MỞ ĐẦU

1. Lý do chọn đề tài

Trong một thế giới phát triển nhanh chóng được thúc đẩy bởi những tiến bộ công nghệ, các vấn đề việc làm đang trở nên ngày càng phức tạp, đòi hỏi các giải pháp hiện đại. Sự kết hợp giữa trí tuệ nhân tạo và mạng lưới thông tin mở ra vô số cơ hội để cung cấp thông tin chính xác và hỗ trợ kịp thời cho người tìm việc. Điều này không chỉ làm giảm áp lực cá nhân mà còn góp phần vào sự phát triển bền vững của nền kinh tế.

Mục tiêu của dự án này là xây dựng một hệ thống toàn diện, tận dụng trí tuệ nhân tạo, cung cấp thông tin toàn diện, chi tiết, chính xác để hỗ trợ người tìm việc đa dạng. Hệ thống không chỉ phân tích và đề xuất việc làm phù hợp mà còn cung cấp thông tin về xu hướng thị trường lao động, giúp người tìm việc tiếp cận nguồn tuyển dụng nhanh chóng và hiệu quả.

Hệ thống sẽ sử dụng thuật toán trí tuệ nhân tạo để thu thập và phân tích dữ liệu trích xuất từ danh sách việc làm trên các nền tảng tuyển dụng phổ biến hiện nay như Facebook, LinkedIn, TopDev, Vieclam24h,... Công nghệ máy học sẽ được ứng dụng để tối ưu hóa quá trình dự đoán cung cầu trong tương lai cho các vị trí việc làm khác nhau.

Dự án này không chỉ mang lại lợi ích trực tiếp cho người tìm việc mà còn góp phần giảm thiểu tình trạng thất nghiệp, giúp người tìm việc đón đầu xu hướng và chuẩn bị kỹ lưỡng cho con đường sự nghiệp của mình. Hơn nữa, dự án còn mở ra cơ hội nghiên cứu và phát triển trong lĩnh vực trí tuệ nhân tạo và ứng dụng công nghệ để giải quyết các vấn đề xã hội cấp bách.

Website sẽ được xây dựng và phát triển trên nền tảng internet, cho phép người dùng truy cập hệ thống để tìm kiếm cơ hội việc làm phù hợp và cung cấp phản hồi về dự đoán nhu cầu việc làm trong tương lai.

2. Mục tiêu

Xây dựng hệ thống giúp người tìm việc dễ dàng tìm được việc làm từ các tin tuyển dụng trích xuất từ các trang web khác.

Xây dựng hệ thống cho phép người dùng xem số liệu thống kê tuyển dụng tại 3 thị trường lao động lớn hiện nay tại Việt Nam là Hà Nội, Đà Nẵng và Hồ Chí Minh, bao gồm tình hình nhu cầu tuyển dụng.

Ứng dụng AI để dự đoán nhu cầu việc làm trong tương lai.

Xây dựng Chatbot nhằm mục đích tư vấn chính xác ngành nghề phù hợp cho người dùng.

3. Các bước thực hiện

Để đạt được các mục tiêu trên, đề tài cần thực hiện các bước sau:

- Nghiên cứu về chủ đề này.
- Thu thập dữ liệu và tập hợp dữ liệu về thông tin tuyển dụng.
- Tìm hiểu về thuật toán LSTM, Transformer.
- Tiến hành đào tạo mô hình.
- Kiểm tra độ chính xác của mô hình.
- Xây dựng cơ sở dữ liệu.
- Thiết kế giao diện người dùng (Design).
- Thiết kế giao diện trang quản lý (Thiết kế).
- Phân tích và thiết kế chức năng.
- Chọn công nghệ kỹ thuật.
- Tiến hành mã hóa.
- Kiểm tra.
- Viết báo cáo và triển khai chương trình.

4. Công cụ và môi trường phát triển

Các công nghệ được sử dụng trong chủ đề này:

- Back-end: FastAPI
- Front-end: NextJS
- AI: Python, LSTM, Transformer

Phần mềm và công cụ:

- Cursor
- Google Colab
- Drawl io, Bắt đầu UML

Hệ thống quản lý cơ sở dữ liệu: MongoDB

CHƯƠNG I. CƠ SỞ LÝ THUYẾT

1. Tính cấp thiết của đề tài

Lợi thế cạnh tranh: Với số lượng trang web tuyển dụng ngày càng tăng, việc thiết lập sự hiện diện trực tuyến để luôn đi trước đối thủ cạnh tranh là điều bắt buộc. Xây dựng một trang web tổng hợp thông tin từ các trang web tuyển dụng khác nhau một cách nhanh chóng và đầy đủ giúp người tìm việc tiếp cận nhanh chóng và đơn giản, đồng thời giành được lợi thế cạnh tranh trong ngành.

Kỳ vọng của người tiêu dùng: Người tiêu dùng hiện đại mong muốn sự tiện lợi và khả năng truy cập nhanh chóng vào nhiều loại bài đăng việc làm khi tìm kiếm việc làm. Trang web tuyển dụng cung cấp cho người dùng khả năng duyệt danh sách và xem thông tin chi tiết về bài đăng việc làm một cách nhanh chóng và chính xác. Các tính năng AI cũng có thể được sử dụng để dự đoán các cơ hội việc làm trong tương lai để lập kế hoạch kỹ năng và dự đoán việc làm trong tương lai, do đó làm giảm khả năng thất nghiệp.

2. Tổng quan về giải pháp kỹ thuật

2.1. Lập kế hoạch và thiết kế

- Xác định mục tiêu và yêu cầu của trang web tổng hợp việc làm.
- Phân tích yêu cầu, vẽ sơ đồ hoạt động, trường hợp sử dụng, thiết kế cơ sở dữ liệu...đánh giá và triển khai.
- Thiết kế giao diện người dùng cho phần front-end bằng các công cụ như Figma hoặc tham khảo giao diện của trang web liên quan.

2.2. Thiết lập môi trường phát triển

- Cài đặt Node.js, python, Docker, mongoDB và npm (Trình quản lý gói Node) trên máy tính.
- Thiết lập thư mục dự án mới cho trang web.

2.3. FrontEnd với NextJs

- Tạo các thành phần NextJs cho các phần khác nhau của trang web như đầu trang, chân trang, danh sách bất động sản, chức năng tìm kiếm và biểu mẫu liên hệ.
- Sử dụng React Router để định tuyến phía máy khách nhằm điều hướng giữa các trang hoặc chế độ xem khác nhau.
- Triển khai thiết kế đáp ứng bằng CSS hoặc các khung CSS như Mantine để đảm bảo trang web trông đẹp trên nhiều thiết bị khác nhau.

2.4. BackEnd với FastAPI

- Xác định các tuyến đường và điểm cuối để xử lý các yêu cầu từ giao diện người dùng, chẳng hạn như tìm danh sách bất động sản, gửi yêu cầu hoặc xử lý xác thực người dùng.
- Tích hợp với cơ sở dữ liệu MongoDB để lưu trữ và truy xuất dữ liệu như danh sách bất động sản, tài khoản người dùng và yêu cầu.

2.5. Kết nối FrontEnd và BackEnd

- Triển khai API RESTful ở phần phụ trợ để xử lý các yêu cầu này và trả về phản hồi phù hợp.
- Bảo mật các điểm cuối API bằng cơ chế xác thực và ủy quyền để bảo vệ dữ liệu nhạy cảm và ngăn chặn truy cập trái phép.

2.6. Kiểm tra và gỡ lỗi

- Kiểm tra kỹ lưỡng trang web để xác định và khắc phục mọi lỗi hoặc sự cố.

2.7. Giám sát và bảo trì

- Thiết lập giám sát và ghi nhật ký cho trang web để theo dõi số liệu hiệu suất, lỗi và tương tác của người dùng.
- Cập nhật thường xuyên trang web với các tính năng mới, cải tiến và bản vá bảo mật khi cần thiết.

3. Lý thuyết và công nghệ

3.1. HTML

HTML (Ngôn ngữ đánh dấu siêu văn bản) là ngôn ngữ đánh dấu mà chúng ta sử dụng để cấu trúc và cung cấp ý nghĩa cho nội dung trang web của mình, ví dụ như xác định đoạn văn, tiêu đề và bảng dữ liệu hoặc nhúng hình ảnh và video vào trang.

Năm 1990, như một phần trong tầm nhìn của mình về Web, Tim Berners-Lee đã định nghĩa khái niệm siêu văn bản, mà Berners-Lee đã chính thức hóa vào năm sau thông qua một đánh dấu chủ yếu dựa trên SGML. IETF bắt đầu chính thức chỉ định HTML vào năm 1993 và sau một số bản thảo đã phát hành phiên bản 2.0 vào năm 1995. Năm 1994, Berners-Lee thành lập W3C để phát triển Web. Năm 1996, W3C tiếp quản công việc HTML và công bố khuyến nghị HTML 3.2 một năm sau đó. HTML 4.0 được phát hành vào năm 1999 và trở thành tiêu chuẩn ISO vào năm 2000.

Vào thời điểm đó, W3C gần như đã từ bỏ HTML để chuyển sang XHTML, thúc đẩy việc thành lập một nhóm độc lập có tên là WHATWG vào năm 2004. Nhờ WHATWG, công việc về HTML vẫn tiếp tục: hai tổ chức đã phát hành bản thảo đầu tiên của HTML5 vào năm 2008 và một tiêu chuẩn chính thức vào năm 2014. Thuật ngữ

"HTML5" chỉ là một từ thông dụng dùng để chỉ các công nghệ web hiện đại, là một phần của HTML Living Standard

3.2. CSS

CSS (Cascading Style Sheets) [1] là ngôn ngữ của các quy tắc kiểu mà chúng ta sử dụng để áp dụng kiểu cho nội dung HTML của mình, ví dụ như thiết lập màu nền và phông chữ, và bố trí nội dung của chúng ta thành nhiều cột.

3.3. JavaScript

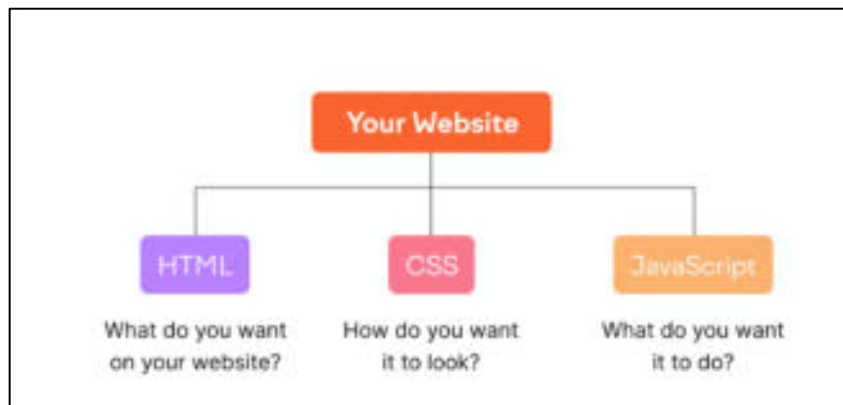
JavaScript (JS) [2] là một ngôn ngữ lập trình máy tính được sử dụng để làm cho các trang web và ứng dụng trở nên năng động và tương tác.

Tính năng này độc đáo ở chỗ nó có thể chạy trực tiếp trên trình duyệt của bạn chứ không chỉ trên máy chủ.

Cùng với ngôn ngữ đánh dấu siêu văn bản (HTML) và biểu định kiểu xếp tầng (CSS), JavaScript là một trong những ngôn ngữ lập trình được sử dụng phổ biến nhất trên internet.

Trên thực tế, tính đến tháng 3 năm 2023, 98,4% tất cả các trang web đều sử dụng JavaScript.

JavaScript, CSS và HTML hoạt động cùng nhau để tạo nên các thành phần hướng đến người dùng của hầu hết các trang web và ứng dụng trực tuyến.



Hình 1: Kết hợp trang web

JavaScript bổ sung hành vi và tính tương tác. Nhưng nó có thể làm được nhiều hơn thế nữa. Sự phổ biến ngày càng tăng của nó đã tạo điều kiện thuận lợi cho việc phát triển các thư viện và khuôn khổ, các công cụ JS chuyên dụng như V8 (cung cấp năng lượng cho Chrome) và các môi trường phía máy chủ.

- Thư viện:

- Thư viện là tập hợp các mã được viết sẵn có thể được sử dụng để dễ dàng triển khai các tác vụ JavaScript thường dùng.

- Thư viện giúp phát triển nhanh hơn vì các nhà phát triển không cần phải viết từng dòng mã.
- Khung là tập hợp các thư viện.
- Hãy coi thư viện như công cụ và khung công tác như hộp công cụ đầy đủ chức năng.
- Mặc dù thư viện có thể cung cấp mã cho một chức năng cụ thể trên trang web của bạn, nhưng một khung có thể cung cấp toàn bộ mã bạn cần để xây dựng trang web.
- Các thư viện JavaScript phổ biến bao gồm jQuery và React . Các khuôn khổ JavaScript phổ biến bao gồm Vue.js, Node.js và Angular.

JavaScript được sử dụng để làm gì: JavaScript được sử dụng để tạo các trang web động, ứng dụng web và di động, trò chơi, máy chủ web, cơ sở hạ tầng phụ trợ, v.v.

3.4. MongoDB

MongoDB là một hệ thống quản lý cơ sở dữ liệu NoSQL được phát triển bởi MongoDB Inc., lần đầu tiên ra mắt vào năm 2009. MongoDB được thiết kế để lưu trữ dữ liệu dưới dạng tài liệu BSON (một dạng nhị phân của JSON), giúp linh hoạt hơn so với các hệ thống quản lý cơ sở dữ liệu quan hệ truyền thống. MongoDB tương thích rộng rãi với nhiều công nghệ và kiến trúc hiện đại, và có thể chạy trên hầu hết các nền tảng lớn, bao gồm các hệ điều hành dựa trên Unix như Linux, Mac OS và cả Windows.

MongoDB được phát hành dưới giấy phép mã nguồn mở và hỗ trợ truy vấn dữ liệu bằng cú pháp JSON-like. Nó hoạt động nhanh, dễ mở rộng, và rất hiệu quả với các bộ dữ liệu lớn, phi cấu trúc hoặc bán cấu trúc. Mặc dù không sử dụng mô hình bảng như hệ quản trị quan hệ, nhưng mô hình tài liệu của MongoDB mang tính trực quan cao và đặc biệt phù hợp cho các ứng dụng hiện đại yêu cầu linh hoạt và khả năng mở rộng.



Hình 2: MongoDB

3.5. API RESTful

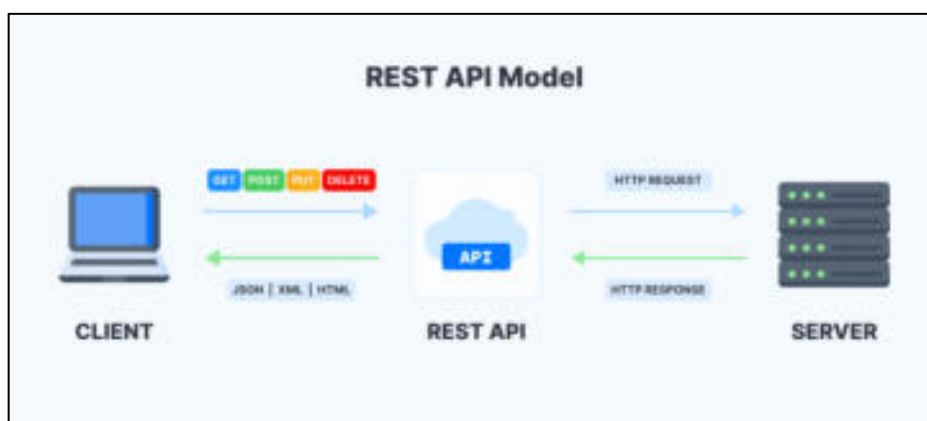
API là một tiêu chuẩn để tạo API cho các ứng dụng web (thiết kế dịch vụ Web) nhằm giúp quản lý tài nguyên dễ dàng hơn. Nó tập trung vào tài nguyên thiết bị (tệp văn bản, hình ảnh, âm thanh, video hoặc dữ liệu động), cũng như trạng thái tài nguyên được định dạng và gửi qua HTTP. API Restful hoạt động để cung cấp cho máy khách một loạt kết quả sau khi họ liên kết đến URL. Để API Restful hoạt động, người ta phải sử dụng đúng các thủ tục HTTP. Các thủ tục này chỉ định thao tác cần thực hiện với API.

REST xác định cấu trúc của API. Các nhà phát triển phải tuân thủ một số quy tắc nhất định khi xây dựng API. Ví dụ, một quy tắc nêu rằng liên kết đến URL phải trả về một số thông tin. Mỗi URL được gọi là Yêu cầu, trong khi dữ liệu được trả về được gọi là Phản hồi. Nó sử dụng các yêu cầu sau:

- GET để lấy dữ liệu
- PUT để thay đổi trạng thái của dữ liệu (chẳng hạn như đối tượng, tệp hoặc khối)
- POST để tạo dữ liệu
- XÓA các phương pháp để loại bỏ nó

REST API sử dụng phần Status-Line của thông báo phản hồi HTTP để thông báo cho khách hàng về kết quả bao quát của yêu cầu. RFC 2616 định nghĩa cú pháp Status-Line. HTTP định nghĩa các mã trạng thái chuẩn này có thể được sử dụng để truyền đạt kết quả yêu cầu của khách hàng. Các mã trạng thái được chia thành năm loại.

- 1xx: Thông tin: Truyền đạt thông tin ở cấp độ giao thức truyền tải.
- 2xx: Thành công: Chỉ ra rằng yêu cầu của khách hàng đã được chấp nhận thành công.
- 3xx: Chuyển hướng: Chỉ ra rằng máy khách phải thực hiện một số hành động bổ sung để hoàn tất yêu cầu của họ.
- 4xx: Lỗi của khách hàng: Loại mã trạng thái lỗi này chỉ ra lỗi của khách hàng.
- 5xx: Lỗi máy chủ : Máy chủ chịu trách nhiệm về các mã trạng thái lỗi này.



Hình 3: Mô hình Rest API

3.6. Tổng quan về NextJs

3.6.1. Next.js

Next.js là một framework mã nguồn mở được xây dựng trên React, giúp phát triển ứng dụng web với khả năng kết xuất phía máy chủ (SSR) và tạo trang tĩnh (SSG) dễ dàng hơn. Next.js được phát triển bởi Vercel và lần đầu ra mắt vào năm 2016. Framework này tối ưu cho hiệu suất, khả năng SEO, và trải nghiệm người dùng.



Hình 4: Framework Next.js

3.6.2. Các tính năng của Next.js

- Kết xuất phía máy chủ (SSR) và tạo trang tĩnh (SSG) giúp cải thiện tốc độ tải trang và SEO.
- Hệ thống định tuyến dựa trên thư mục, tự động tạo route từ file trong thư mục pages/.
- Hỗ trợ API routes, cho phép viết backend nhỏ ngay trong dự án.
- Tối ưu hóa hình ảnh tự động với next/image, giúp cải thiện hiệu suất.
- Hỗ trợ TypeScript, CSS Modules, Tailwind CSS,... ngay từ đầu.
- Tự động chia nhỏ code, giúp ứng dụng chỉ tải những gì cần thiết.
- Tích hợp tốt với Vercel để deploy ứng dụng nhanh chóng.

3.6.3. Tại sao sử dụng Next.js

- Kết hợp giữa client-side và server-side rendering một cách linh hoạt.
- Tối ưu SEO hơn so với SPA thông thường vì nội dung có thể được render trước trên server.
- Đơn giản hóa quá trình routing và cấu hình so với dùng React thuần.
- Dễ dàng mở rộng với các plugin và cấu hình tùy chỉnh khi cần.
- Phù hợp cho cả ứng dụng nhỏ (blog, landing page) và ứng dụng lớn (dashboard, hệ thống).

3.6.4. API Routes

Next.js cho phép tạo các endpoint phía server trong thư mục `pages/api`. Đây là giải pháp lý tưởng khi bạn muốn xử lý logic backend nhẹ mà không cần server riêng biệt.

3.6.5. Next.js với Redux

- Có thể tích hợp Redux để quản lý trạng thái trong các ứng dụng phức tạp.
- Kết hợp với middleware như `redux-thunk` hoặc `redux-saga` để xử lý logic bất đồng bộ.
- Cần cấu hình thêm để Redux hoạt động đúng trong môi trường SSR của Next.js (ví dụ dùng `next-redux-wrapper`).

3.7. FastAPI

FastAPI là một framework web hiện đại, nhanh và hiệu quả được xây dựng bằng Python. Nó được thiết kế để tạo các API RESTful hiệu suất cao và dễ bảo trì, tận dụng sức mạnh của Python 3.6+ với `type hint`.



Hình 5: Framework FastAPI

3.7.1. Đặc điểm của FastAPI

- Hiện đại và hiệu suất cao: Sử dụng `Starlette` và `Pydantic` giúp FastAPI nhanh gần như Node.js và Go.
- Tự động sinh tài liệu API: Hỗ trợ Swagger UI và ReDoc mặc định.
- Hỗ trợ kiểm tra kiểu mạnh mẽ: Dựa trên `type hint` của Python, giúp phát hiện lỗi sớm.
- Hỗ trợ `async/await`: Dễ dàng xây dựng các API bất đồng bộ.
- Dễ tích hợp: Có thể tích hợp dễ dàng với các ORM như SQLAlchemy, Tortoise, hay `asyncpg`.

3.7.2. Lợi ích khi sử dụng FastAPI

- Phát triển nhanh: Viết ít code hơn, ít lỗi hơn nhờ hệ thống kiểm tra kiểu.
- Dễ bảo trì: Code rõ ràng, dễ đọc và dễ mở rộng.
- Tự sinh Swagger docs: Không cần cấu hình thêm để có tài liệu API trực quan.
- Hiệu suất cao: Ngang tầm với Node.js hoặc Go trong các bài benchmark.
- Cộng đồng đang phát triển mạnh: FastAPI được sử dụng bởi nhiều công ty lớn như Microsoft, Netflix, Uber,...

3.7.3. Một số ứng dụng sử dụng FastAPI

- **Netflix**: Dùng FastAPI cho các microservices hỗ trợ backend.
- **Microsoft**: Dùng FastAPI trong các dịch vụ Azure.
- **Uber**: Dùng FastAPI để build các dịch vụ ML nội bộ.
- **Explosion AI**: Dùng FastAPI cho thư viện NLP `spaCy`.

FastAPI là lựa chọn lý tưởng cho các backend Python hiện đại – đặc biệt khi cần hiệu suất cao, phát triển nhanh, và tích hợp AI/ML.

3.8. Python

Ngôn ngữ Python[3] được sử dụng rộng rãi trong lĩnh vực phát triển ứng dụng, bao gồm các ví dụ sau:

Phát triển web phía máy chủ:

- Python cung cấp nhiều loại framework, mỗi framework có điểm mạnh và tính năng riêng. Ví dụ, Django là một framework cấp cao nhấn mạnh vào phát triển nhanh và bao gồm các tính năng tích hợp như ORM (Object-Relational Mapping) để tương tác với cơ sở dữ liệu, bảng điều khiển quản trị và xác thực. Mặt khác, Flask là một micro-framework nhẹ cung cấp tính linh hoạt và đơn giản, cho phép các nhà phát triển lựa chọn và tích hợp chỉ các thành phần họ cần.
- Các khung Python cung cấp giao diện để sử dụng để tương tác với cơ sở dữ liệu. Các thư viện ORM như ORM của Django và SQLAlchemy trong Flask trừu tượng hóa lớp cơ sở dữ liệu, cho phép các nhà phát triển làm việc với các đối tượng cơ sở dữ liệu bằng mã Python thay vì trực tiếp truy vấn SQL.
- Các khuôn khổ web phía máy chủ trong Python sử dụng cơ chế định tuyến để ánh xạ URL tới các hàm Python được gọi là chế độ xem. Chế độ xem tạo nội dung động hoặc hiển thị các mẫu để trả về dưới dạng phản hồi HTTP cho các yêu cầu của máy khách.
- Các khuôn khổ web Python thường bao gồm các công cụ mẫu để tạo HTML động. Các công cụ mẫu này cho phép các nhà phát triển viết các mẫu HTML với các trình

giữ chỗ cho nội dung động, sau đó được kết xuất bằng dữ liệu do mã phía máy chủ cung cấp.

- Các ứng dụng web Python có thể được triển khai bằng nhiều máy chủ web và công cụ triển khai khác nhau. Các tùy chọn triển khai bao gồm từ các máy chủ web truyền thống như Apache và Nginx đến các máy chủ ứng dụng như Gunicorn và uWSGI. Ngoài ra, các nền tảng đám mây như AWS, Google Cloud Platform và Heroku cung cấp các giải pháp lưu trữ có khả năng mở rộng cho các ứng dụng web Python.

Khoa học dữ liệu và học máy:

- Python cung cấp một bộ thư viện và công cụ phong phú cho các tác vụ khoa học dữ liệu và học máy. Một số thư viện được sử dụng rộng rãi nhất bao gồm NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, TensorFlow và PyTorch . Các thư viện này cung cấp chức năng để thao tác dữ liệu, trực quan hóa, phân tích thống kê và xây dựng các mô hình học máy.

- Các thư viện như NumPy và Pandas cung cấp các công cụ mạnh mẽ để xử lý và phân tích dữ liệu. NumPy cung cấp hỗ trợ cho các phép toán số và mảng đa chiều, trong khi Pandas cung cấp các cấu trúc dữ liệu và hàm để làm việc với dữ liệu có cấu trúc, chẳng hạn như khung dữ liệu.

- Matplotlib, Seaborn và Plotly là các thư viện Python phổ biến để trực quan hóa dữ liệu.

- Python được sử dụng rộng rãi để xây dựng các mô hình học máy do các thư viện như Scikit-learn, TensorFlow và PyTorch. Scikit-learn cung cấp nhiều thuật toán và công cụ cho các tác vụ như phân loại, hồi quy, phân cụm và giảm chiều. TensorFlow và PyTorch là các khuôn khổ học sâu cung cấp các công cụ linh hoạt và có thể mở rộng để xây dựng và đào tạo mạng nơ-ron.



Hình 6: Python

3.9. Docker

Docker là một nền tảng mở để phát triển, vận chuyển và chạy các ứng dụng. Docker cho phép bạn tách các ứng dụng của mình khỏi cơ sở hạ tầng để bạn có thể phân phối phần mềm nhanh chóng. Với Docker, bạn có thể quản lý cơ sở hạ tầng của mình theo cùng cách bạn quản lý các ứng dụng của mình. Bằng cách tận dụng các phương pháp của Docker để vận chuyển, thử nghiệm và triển khai mã, bạn có thể giảm đáng kể độ trễ giữa việc viết mã và chạy mã trong sản xuất.

Docker cung cấp khả năng đóng gói và chạy ứng dụng trong môi trường được cô lập lỏng lẻo gọi là container. Sự cô lập và bảo mật cho phép bạn chạy nhiều container cùng lúc trên một máy chủ nhất định. Container nhẹ và chứa mọi thứ cần thiết để chạy ứng dụng, do đó bạn không cần phải dựa vào những gì được cài đặt trên máy chủ. Bạn có thể chia sẻ container trong khi làm việc và đảm bảo rằng mọi người bạn chia sẻ đều có cùng một container hoạt động theo cùng một cách.

Docker hợp lý hóa vòng đời phát triển bằng cách cho phép các nhà phát triển làm việc trong môi trường chuẩn hóa bằng cách sử dụng các container cục bộ cung cấp các ứng dụng và dịch vụ của bạn. Container rất phù hợp cho quy trình làm việc tích hợp liên tục và phân phối liên tục (CI/CD).

Sử dụng các container Docker làm khối xây dựng cốt lõi có thể cho phép tạo ra các ứng dụng và nền tảng hiện đại. Docker tạo điều kiện xây dựng và chạy các kiến trúc vi dịch vụ phân tán, triển khai mã của bạn với các quy trình tích hợp liên tục và phân phối liên tục được chuẩn hóa, xây dựng các hệ thống xử lý dữ liệu có khả năng mở rộng cao, cũng như tạo ra các nền tảng được quản lý hoàn toàn và dễ quản lý cho các nhà phát triển của bạn. Sự hợp tác gần đây giữa AWS và Docker giúp triển khai các thành phần giống Docker Compose cho Amazon ECS và AWS Fargate dễ dàng hơn bao giờ hết.

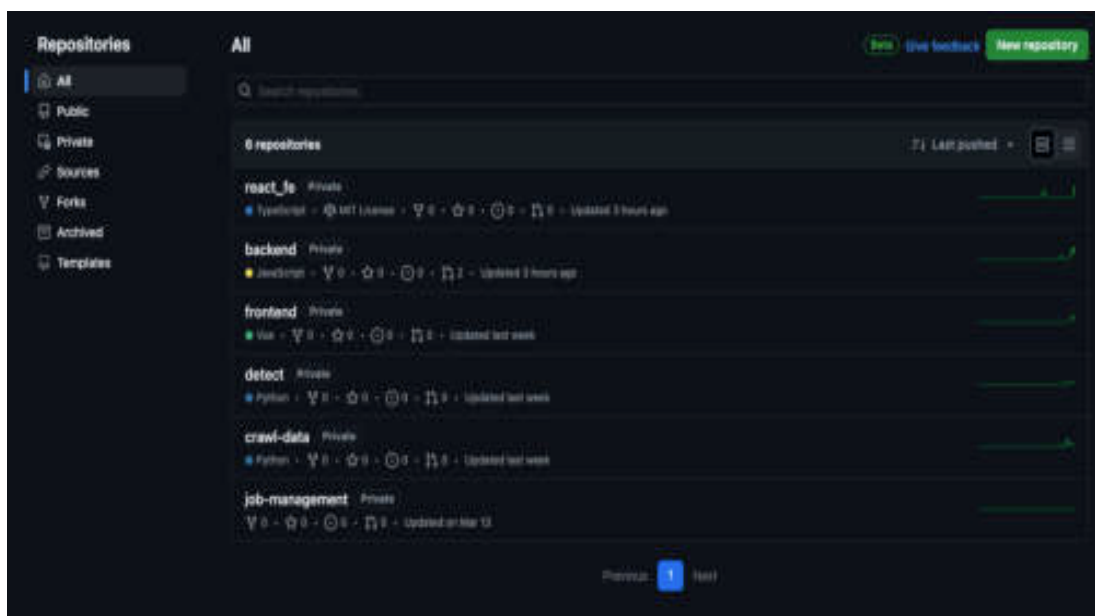
	Name	Container ID	Image	Port(s)	CPU (%)	Last start	Actions
<input type="checkbox"/>	job-managemet				2.54%	31 seconds	
<input type="checkbox"/>	app-frontend	cb05f9133b56	job-manage	3000:5173	0.19%	31 seconds	
<input type="checkbox"/>	app-api	c2aa7aa31bb2	job-manage	3001:3001	0.05%	31 seconds	
<input type="checkbox"/>	rabbitmq	8fca0d7d708f	rabbitmq3	15672:15672	0.25%	1 minute	

Hình 7: Docker

3.10. Github

GitHub là một hệ thống quản lý phiên bản dự án và mã hoạt động như một mạng xã hội dành cho các lập trình viên. Các lập trình viên có thể sao chép mã nguồn từ một kho lưu trữ và Github là một dịch vụ máy chủ lưu trữ công cộng, mỗi người có thể tạo một tài khoản trên đó để tạo kho lưu trữ riêng của họ để làm việc. [7]

Sử dụng Github để quản lý mã nguồn như hiển thị bên dưới:



Hình 8: Github

3.11. Visual Code

Trong quá trình xây dựng code chương trình cho phần front-end và back-end, công cụ biên tập được sử dụng là VSCode (hình 13), đặc điểm nổi bật là đơn giản, gọn nhẹ, dễ cài đặt và hỗ trợ nhiều hệ điều hành cũng như nhiều ngôn ngữ lập trình khác nhau.

3.12. Postman

Trong quá trình kiểm thử, thử nghiệm và thử nghiệm các API đã được lập trình trước khi áp dụng để gọi các yêu cầu từ trang web, công cụ được sử dụng là Postman.



Hình 9: Postman

3.13. Google Gemini

Trong quá trình xử lý dữ liệu thu thập được từ các trang web tuyển dụng việc làm. Phần AI được sử dụng để phân tích văn bản và trích xuất thông tin có liên quan và chính xác để lưu vào cơ sở dữ liệu được sử dụng bởi google gemini AI, một ứng dụng AI cung cấp tốc độ xử lý nhanh và độ chính xác cao.

CHƯƠNG II. HỆ THỐNG PHÂN TÍCH VÀ THIẾT KẾ

1. Phân tích yêu cầu chức năng

1.1. Khách

- Đăng ký: Khách hàng có thể đăng ký tài khoản trên website để lưu tin tuyển dụng, sử dụng công cụ AI trong hệ thống và đăng ký email để nhận thông tin về việc làm mới trong hệ thống.
- Filter Search: Tính năng này cho phép khách hàng tìm kiếm việc làm và lọc theo nhiều lĩnh vực (kỹ năng, nghề nghiệp, mức lương...).
- Danh sách việc làm: Sau khi sử dụng bộ lọc tìm kiếm, người truy cập có thể xem danh sách việc làm phù hợp. Danh sách này sẽ bao gồm thông tin cơ bản về vị trí tuyển dụng, nội dung tuyển dụng, yêu cầu tuyển dụng và thông tin liên quan.

1.2. Người sử dụng

- Đăng xuất: Cho phép người dùng kết thúc phiên làm việc của họ trên hệ thống.
- Danh sách việc làm: Sau khi sử dụng bộ lọc tìm kiếm, người truy cập có thể xem danh sách việc làm phù hợp. Danh sách này sẽ bao gồm thông tin cơ bản về vị trí tuyển dụng, nội dung tuyển dụng, yêu cầu tuyển dụng và thông tin liên quan.
- Cập nhật thông tin cá nhân: Cho phép người dùng cập nhật thông tin cá nhân và kỹ năng làm việc để hệ thống có thể gửi thông tin công việc có liên quan.
- Thay đổi mật khẩu: Cho phép người dùng thay đổi mật khẩu tài khoản của họ.
- Quản lý Resume: Cho phép người dùng thêm, sửa, xoá các CV của người dùng đó. Chọn ra CV mặc định cho những chức năng sau
- Chatbot: Cho phép người dùng tương tác với Chatbot của hệ thống, ngoài tư vấn còn có thể search được các công việc trong cơ sở dữ liệu nhằm mục đích kiếm ra công việc phù hợp rồi tự động ứng tuyển vào công việc đó

1.3. Người quản trị

Quản lý người dùng:

- Cho phép người quản trị xem thông tin chi tiết về người dùng đã đăng ký trong hệ thống.
- Cho phép người quản trị sửa đổi thông tin người dùng bao gồm thông tin cá nhân và thông tin liên lạc.
- Cung cấp khả năng xóa tài khoản người dùng nếu cần, cùng với dữ liệu liên quan.

Quản lý công việc:

- Cho phép người quản trị xem thông tin chi tiết về bài đăng được liệt kê trong hệ

thống.

- Cung cấp khả năng xóa hoặc vô hiệu hóa bài đăng khỏi hệ thống, cùng với dữ liệu liên quan.

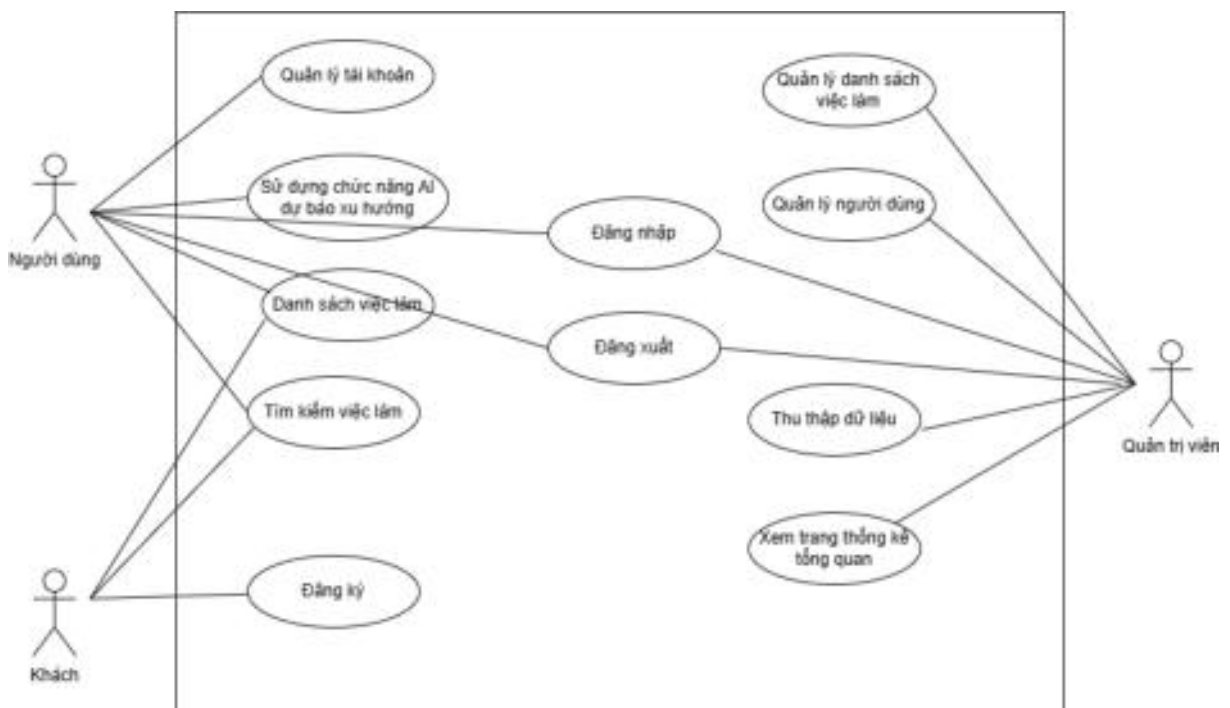
- Cho phép người quản trị xem thông tin chi tiết về bài đăng được liệt kê trong hệ thống.

Quản lý đơn ứng tuyển:

- Cho phép người quản trị xem thông tin chi tiết và trạng thái của đơn ứng tuyển từ người dùng đến công việc

2. Biểu đồ Usecase

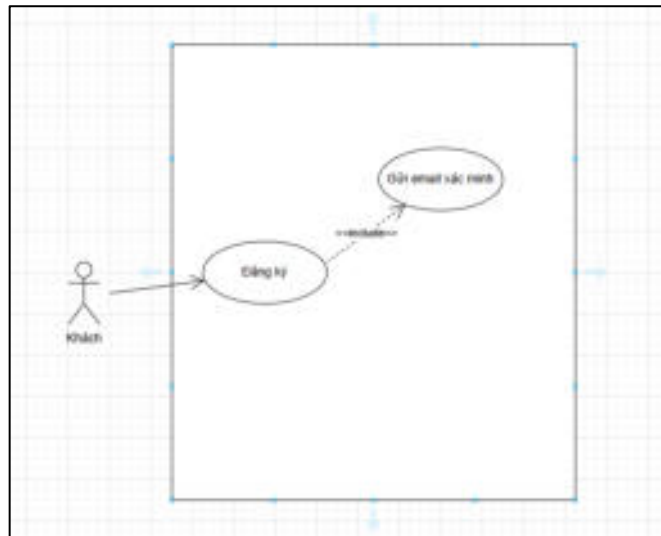
2.1. Biểu đồ Usecase tổng quát



Hình 10: Biểu đồ UC tổng quát

2.2. Sơ đồ Usecase cho khách

2.2.1. Đăng ký

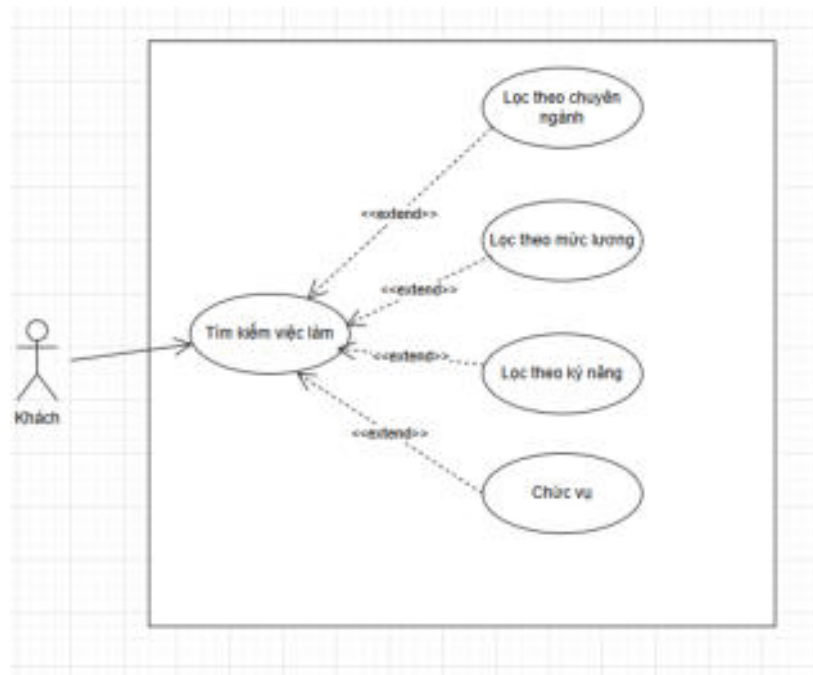


Hình 11: Sơ đồ Usecase “Đăng ký”

Mô tả sơ đồ:

- Khách tương tác với Hệ thống để "Đăng ký tài khoản".
- Hệ thống sử dụng Dịch vụ Email để "Gửi Email xác minh".
- Dịch vụ Email sẽ tương tác lại với Khách bằng cách gửi email xác minh.
- Khách tương tác với Gmail để "Xác minh Email".

2.2.2. Tìm kiếm việc làm



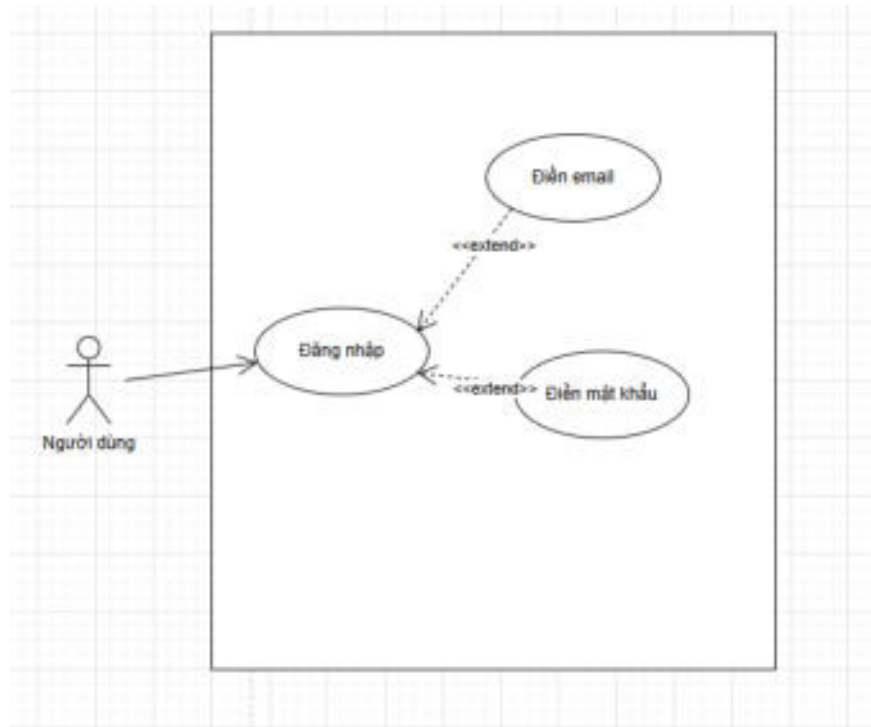
Hình 12: Sơ đồ Usecase “Tìm kiếm việc làm”

Mô tả sơ đồ:

- Nhập tiêu chí tìm kiếm vào thanh tìm kiếm.
- Gửi truy vấn tìm kiếm.
- Xem danh sách kết quả việc làm.

2.3. Sơ đồ Usecase cho người dùng

2.3.1. Đăng nhập

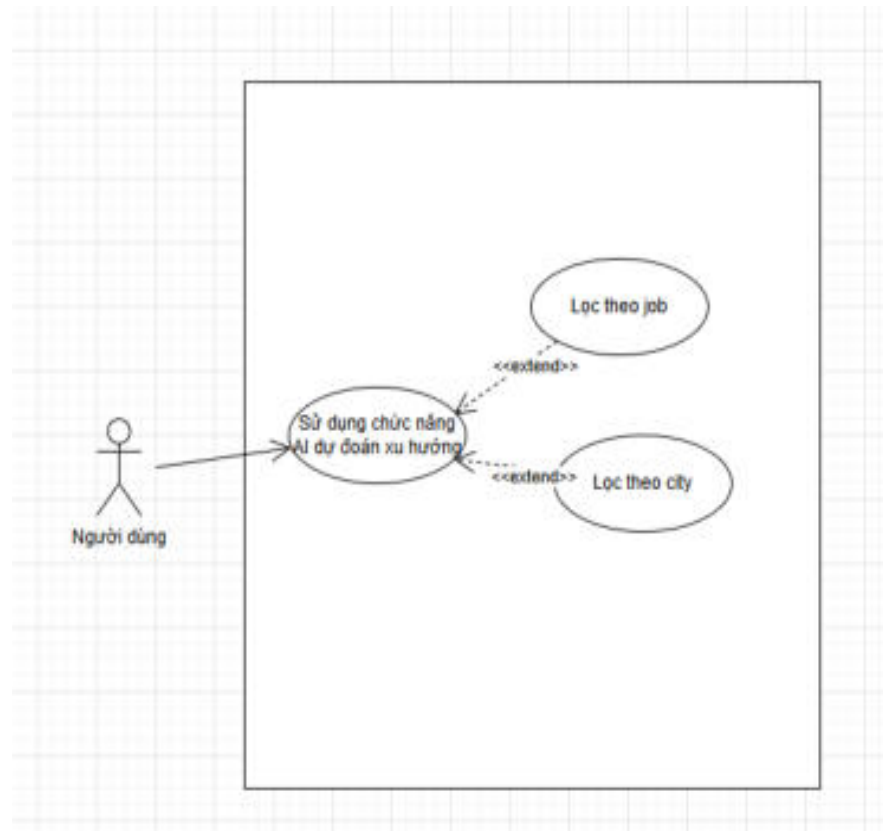


Hình 13: Sơ đồ Usecase “Đăng nhập”

Mô tả sơ đồ:

- Người dùng nhập email và mật khẩu
- Hệ thống tiến hành xác thực
- Trả lại trạng thái đăng nhập

2.3.2. AI Dự đoán xu hướng việc làm

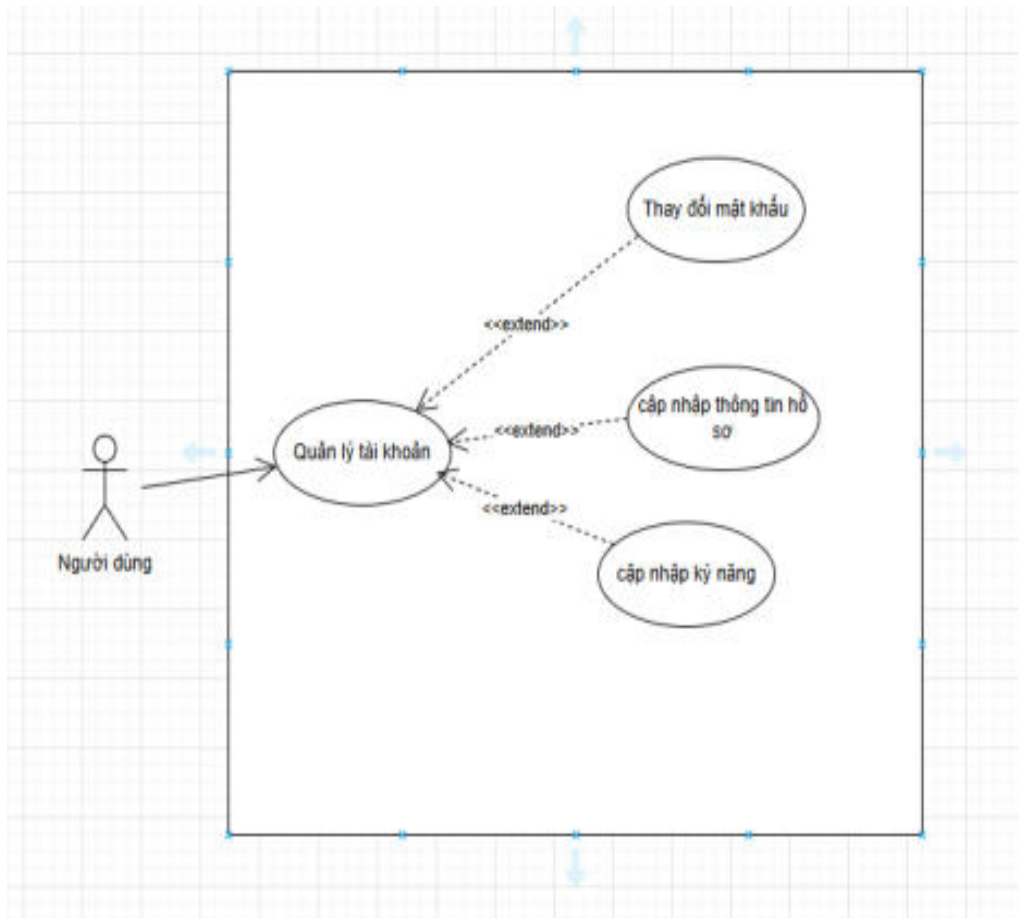


Hình 14: Sơ đồ Usecase “Dự báo xu hướng”

Mô tả sơ đồ:

- Nhập “job” và “city” cần dự đoán từ textbox.
- Nhấn dự đoán.
- Xem biểu đồ và giá trị dự đoán.

2.3.3. Quản lý tài khoản



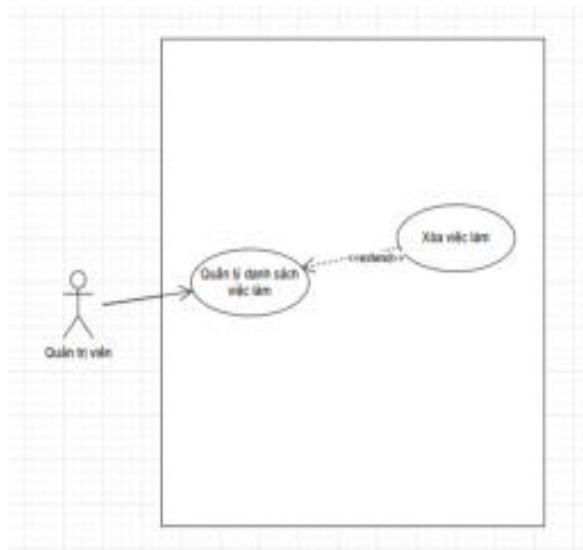
Hình 15: Sơ đồ Usecase “Quản lý tài khoản”

Mô tả sơ đồ:

- Người dùng vào setting đến quản lý tài khoản.
- Người dùng có thể tự chỉnh sửa thông tin cho bản thân.
- Lưu lại nếu đã thấy hợp lý.

2.4. Sơ đồ Usecase cho Quản trị viên

2.4.1. Quản lý danh sách việc làm

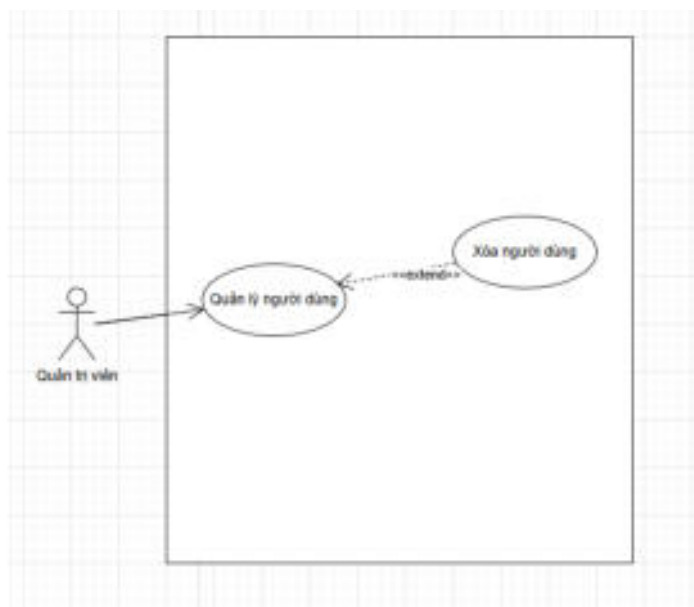


Hình 16: Sơ đồ Usecase “Quản lý danh sách việc làm”

Mô tả sơ đồ:

- Quản trị viên vào quản lý danh sách việc làm.
- Quản trị viên có thể xóa việc làm nếu muốn.
- Hệ thống sẽ hiển thị thông báo xóa thành công.

2.4.2. Quản lý người dùng

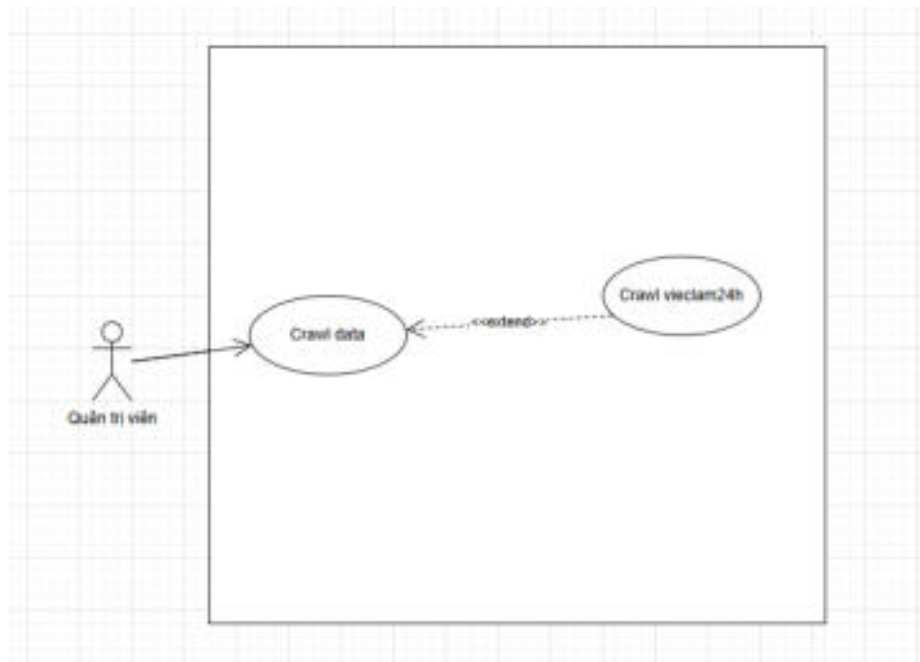


Hình 17: Sơ đồ Usecase “Quản lý người dùng”

Mô tả sơ đồ:

- Quản trị viên vào quản lý người dùng.
- Quản trị viên có thể xóa người dùng nếu muốn.
- Người dùng sẽ bị xóa khỏi danh sách người dùng.

2.4.3. Crawl data



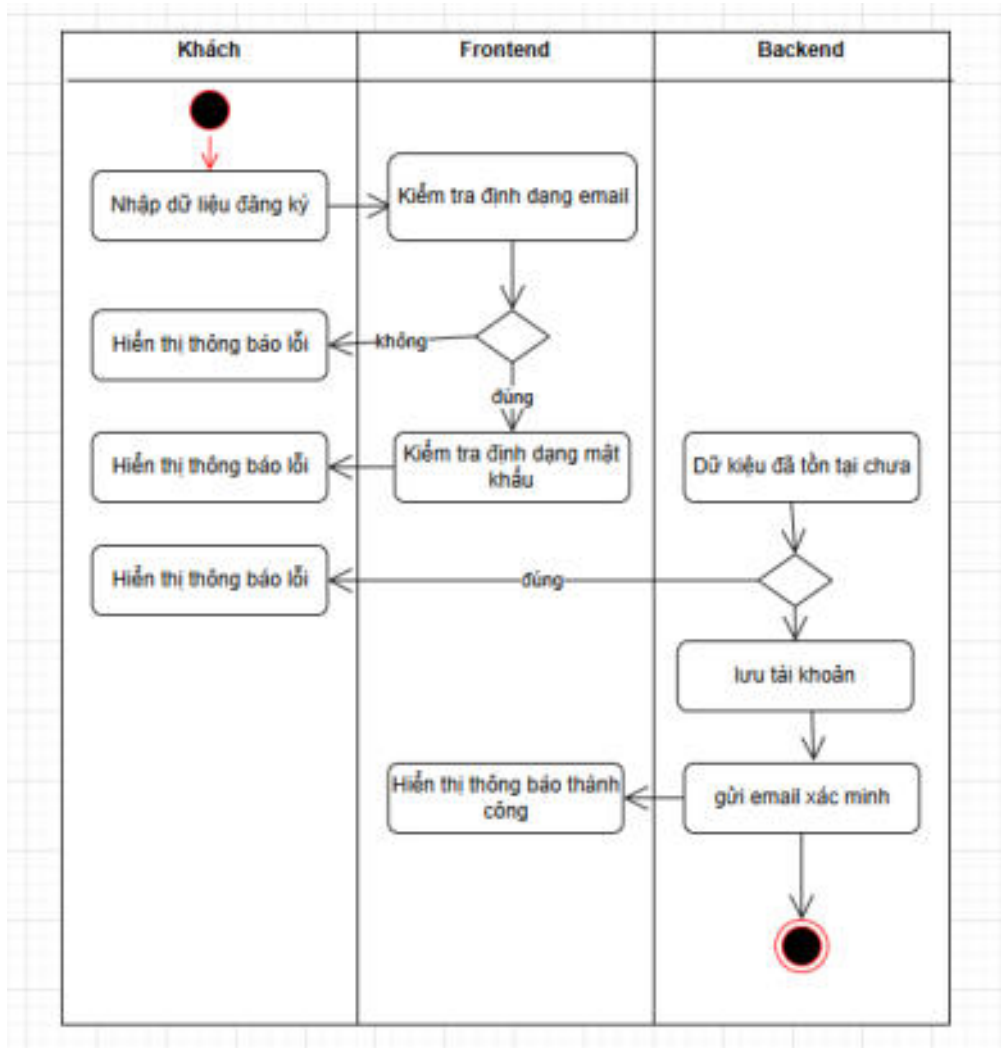
Hình 18: Sơ đồ Usecase “Crawl data”

Mô tả sơ đồ:

- Quản trị viên vào Crawl data.
- Quản trị viên có thể thu thập việc làm cho database.
- Hiện thị thông báo và lịch sử thu thập dữ liệu về database.

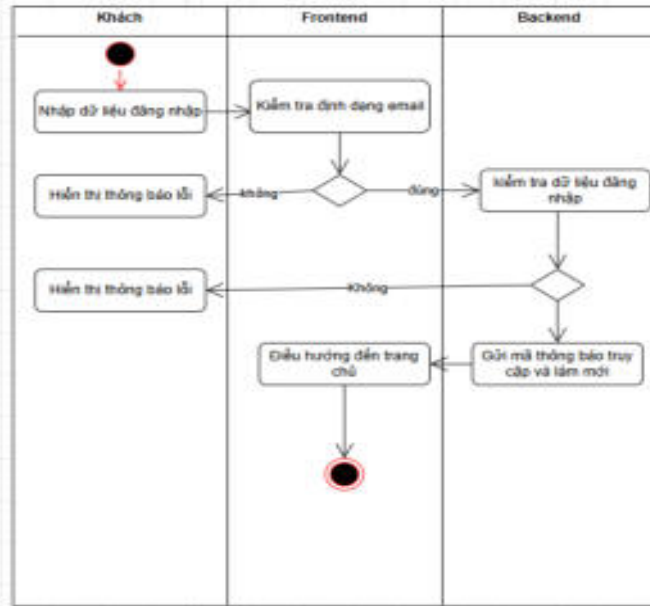
3. Sơ đồ hoạt động

3.1. Chức năng “Đăng ký”



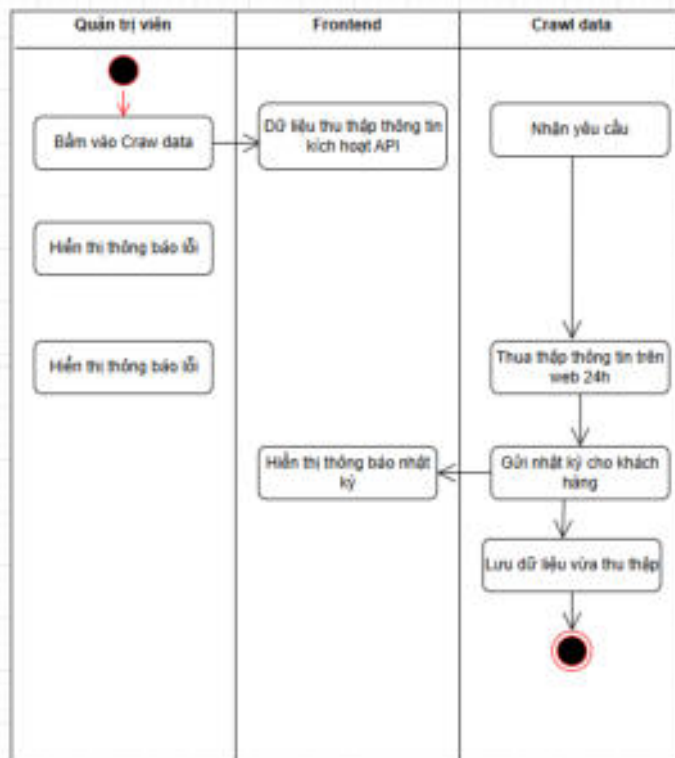
Hình 19: Sơ đồ hoạt động chức năng “Đăng ký người dùng”

3.2. Chức năng “Đăng nhập”



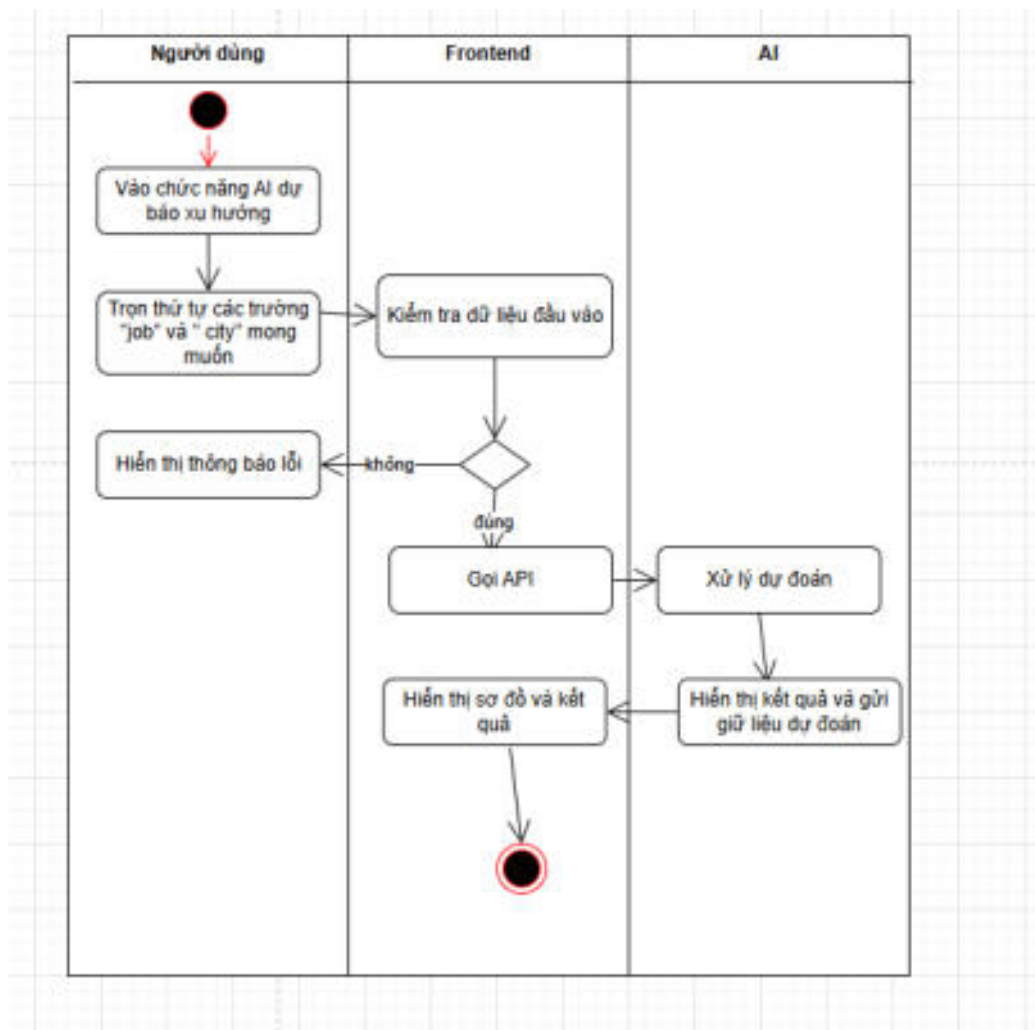
Hình 20: Sơ đồ hoạt động chức năng “đăng nhập”

3.3. Chức năng “Crawl data”



Hình 21: Sơ đồ hoạt động chức năng “Crawl data”

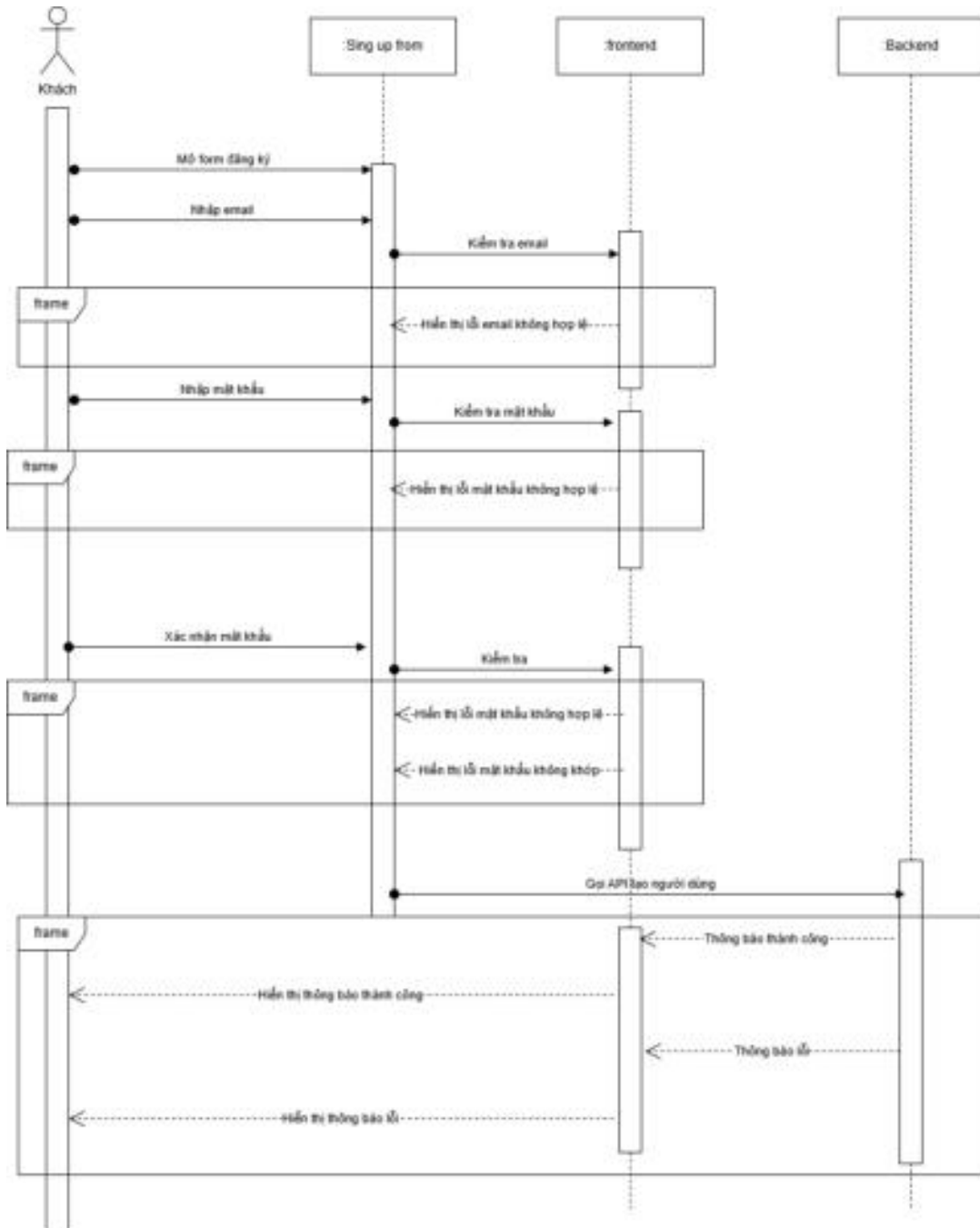
3.4. Chức năng dự đoán xu hướng việc làm



Hình 22: Sơ đồ hoạt động chức năng “Dự đoán xu hướng việc làm”

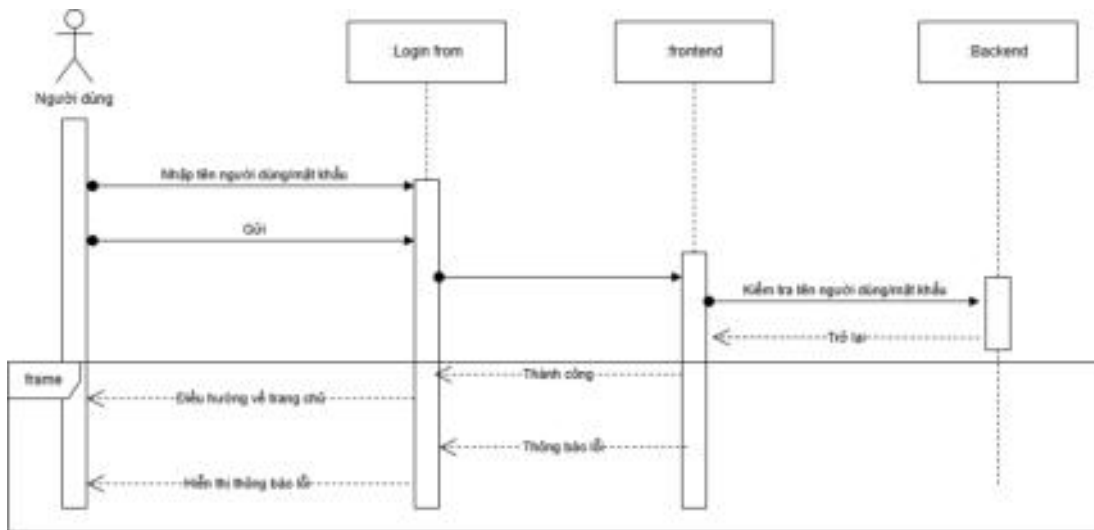
4. Sơ đồ tuần tự

4.1. Chức năng đăng ký người dùng



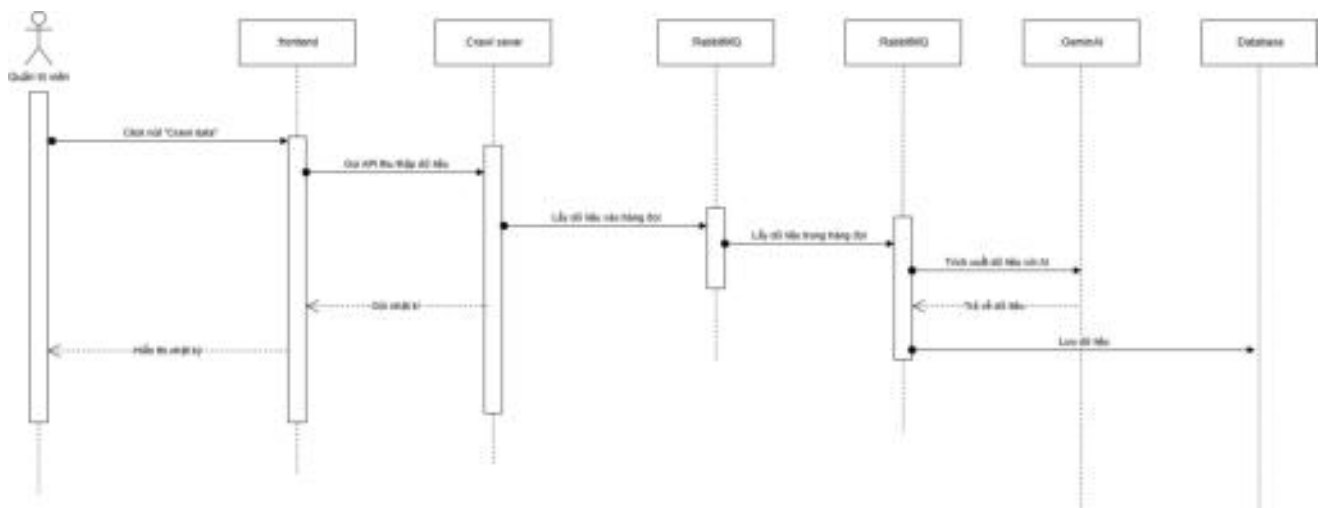
Hình 23: Sơ đồ chức năng “Đăng ký”

4.2. Chức năng đăng nhập



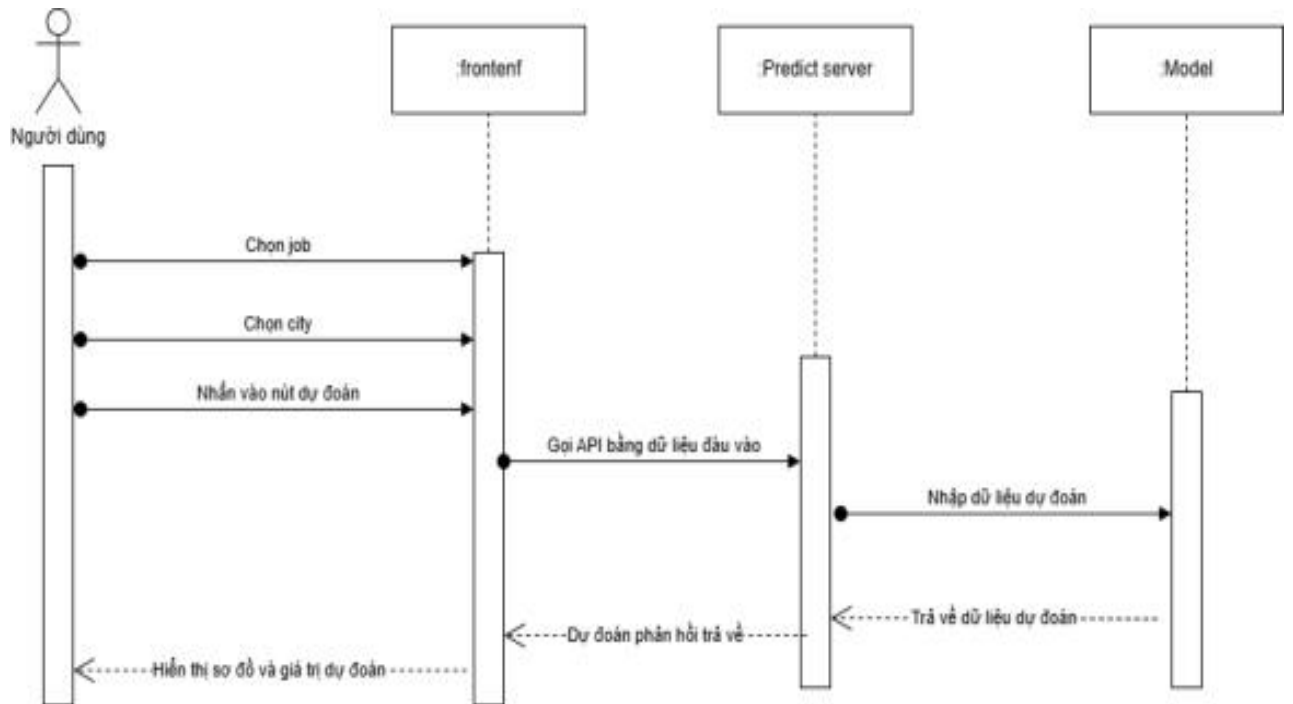
Hình 24: Sơ đồ tuần tự chức năng “Đăng nhập”

4.3. Chức năng thu thập dữ liệu



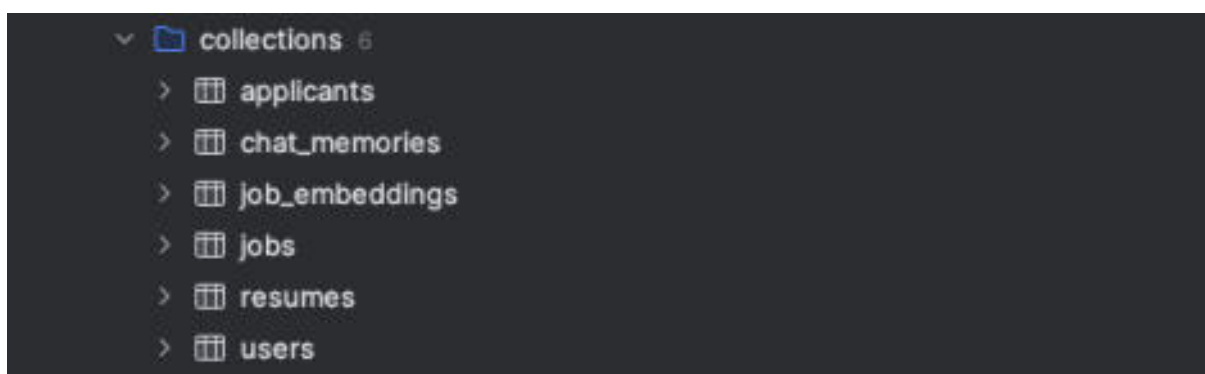
Hình 25: Sơ đồ tuần tự chức năng “Crawl Data”

4.4. Chức năng dự đoán xu hướng việc làm



Hình 26: Sơ đồ tuần tự chức năng " Dự đoán xu hướng việc làm "

5. Thiết kế cơ sở dữ liệu



Hình 27: Cơ sở dữ liệu

6. Các bảng trong cơ sở dữ liệu

Người dùng: Quản lý người dùng

Bảng 2.1. Table users

Bảng 1: Table users

Trường	Ý nghĩa	Ghi chú
id	id user	Khóa chính
email	email	
username	Tên người dùng	
password	Mật khẩu	
role	Vai trò người dùng	
default_resume	Resme mặc định của user	

Bảng 2.2. Table jobs

Bảng 2: Table users skill

Trường	Ý Nghĩa	Note
id	id user skill	Khóa chính
Company	Tên công ty	
Company Image	Hình ảnh logo công ty	
Job Name	Vị trí công việc	
Job URL	Đường link nguồn công việc	
Location	Nơi làm	
Salary	Lương	
Source	Nguồn Crawl	
benefits	Phúc lợi	
job_description	Mô tả công việc	
requirements	Yêu cầu công việc	

Bảng 2.3. Table job embeddings

Bảng 3: Table job embeddings

Trường	Ý nghĩa	Note
id	id job_embedding	Khóa chính
created_at	thời điểm tạo	
embedding	Giá trị embedding	
job_id	id của job	
text	Thông tin job	

Bảng 2.4. Table resume

Bảng 4: Table crawl data

Trường	Ý Nghĩa	Note
id	id user skill	Khóa chính
user_id	User's id	Khóa ngoại
post_id	Post's id	Khóa ngoại
status		

Bảng 2.5. Table applicant

Bảng 5: Table major category

Trường	Ý nghĩa	Note
id	Id đơn ứng tuyển	Khóa chính
apply_time	thời gian ứng tuyển	
cv_file_path	nguồn resume	
email	Email người ứng tuyển	
Job_id	Id công việc	
status	trạng thái của đơn	Chờ, thành công, từ chối
user_id	Id người ứng tuyển	

Bảng 2.6. Table chat_memories

Bảng 6: Table mail schedule

Trường	Ý nghĩa	Note
id	Id cuộc hội thoại	Khóa chính
messages	Tin nhắn	
create_at	thời gian cuộc hội thoại bắt đầu	
update_at	thời gian lần cuối cập nhật	
user_id	Id của user	

7. Đặc tả các usecase

7.1. Đặc tả UC đăng ký

Bảng 7: Đặc tả UC đăng ký

Mã ca sử dụng	UC01
Tên ca sử dụng	Đăng ký người dùng.
Mô tả	Khách có thể vào form đăng ký để đăng ký thành người dùng.
Tác nhân	Khách.
Điều kiện trước	Khách (người chưa có tài khoản).
Điều kiện sau	Khách có tài khoản mới và nhận được email xác thực.
Các bước thực hiện	<ol style="list-style-type: none"> 1.Truy cập chức năng đăng ký (trang chủ). 2. Người dùng nhập thông tin cá nhân (tên, email, số điện thoại, mật khẩu). 3.Hệ thống gửi email xác thực đến người dùng. 4.Khách xác thực tài khoản qua email.

7.2. Đặc tả UC đăng nhập

Bảng 8: Đặc tả UC đăng nhập

Mã ca sử dụng	UC02
Tên ca sử dụng	Đăng nhập người dùng.
Mô tả	Người dùng có thể đăng nhập vào hệ thống bằng tài khoản đã đăng ký.
Tác nhân	Người dùng.
Điều kiện trước	Người dùng đã có tài khoản và đã truy cập vào hệ thống.
Điều kiện sau	Người dùng đăng nhập thành công và truy cập vào trang chủ.
Các bước thực hiện	<ol style="list-style-type: none"> 1. Người dùng truy cập chức năng đăng nhập (trang chủ). 2. Người dùng nhập email và mật khẩu. 3. Hệ thống xác thực thông tin đăng nhập. 4. Người dùng được chuyển hướng tới trang chủ.

7.3. Đặc tả UC tìm kiếm việc làm

Bảng 9: Đặc tả UC tìm kiếm việc làm

Mã ca sử dụng	UC03
Tên ca sử dụng	Tìm kiếm việc làm.
Mô tả	Người dùng có thể tìm kiếm việc làm từ form tìm kiếm .
Tác nhân	Người dùng.
Điều kiện trước	Người dùng truy cập vào hệ thống.
Điều kiện sau	Hiển thị danh sách việc làm cần tìm kiếm.
Các bước thực hiện	<ol style="list-style-type: none"> 1. Người dùng truy cập vào hệ thốn. 2. Người dùng tìm kiếm theo nhu cầu việc làm từ các text box trong form tìm kiếm. 3. Người dùng xác nhận tìm kiếm. 4. Hệ thống sẽ lọc từ database và trả về kết quả công việc phù hợp.

7.4. Đặc tả UC quản lý người dùng

Bảng 10: Đặc tả UC quản lý người dùng

Mã ca sử dụng	UC04
Tên ca sử dụng	Quản lý người dùng.
Mô tả	Quản trị viên có thể xem, chỉnh sửa và xóa người dùng.
Tác nhân	Quản trị viên.
Điều kiện trước	Quản trị viên đã đăng nhập vào hệ thống.
Điều kiện sau	Thông tin người dùng được cập nhật hoặc xóa thành công.
Các bước thực hiện	<ol style="list-style-type: none"> 1. Quản trị viên truy cập chức năng quản lý người dùng. 2. Quản trị viên xem danh sách người dùng. 3. Quản trị viên có thể xóa người dùng. 4. Quản trị viên chọn người dùng cần xóa. 5. Hệ thống cập nhật thông tin người dùng.

7.5. Đặc tả UC quản lý việc làm

Bảng 11: Đặc tả UC quản lý việc làm

Mã ca sử dụng	UC05
Tên ca sử dụng	Quản lý việc làm.
Mô tả	Quản trị viên có thể xóa việc làm.
Tác nhân	Quản trị viên.
Điều kiện trước	Quản trị viên đã đăng nhập vào hệ thống.
Điều kiện sau	Thông tin các sản phẩm được cập nhật hoặc xóa thành công.
Các bước thực hiện	<ol style="list-style-type: none"> 1. Quản trị viên truy cập chức năng quản lý việc làm. 2. Quản trị viên xem danh sách các bài viết. 3. Quản trị viên thêm, chỉnh sửa hoặc xóa việc làm nếu cần thiết. 4. Hệ thống cập nhật danh sách việc làm.

7.6. Đặc tả UC Crawl data

Bảng 12: Đặc tả UC Crawl data

Mã ca sử dụng	UC06
Tên ca sử dụng	Crawl data.
Mô tả	Quản trị viên có thể thêm dữ liệu việc làm vào database.
Tác nhân	Admin
Điều kiện trước	Quản trị viên đã đăng nhập vào hệ thống.
Điều kiện sau	Database được cập nhật thêm các việc làm mới.
Các bước thực hiện	<ol style="list-style-type: none"> 1. Quản trị viên truy cập vào “Crawl data”. 2. Quản trị viên nhấn vào “Crawl data”. 3. Hệ thống tự động thu thập dữ liệu mới từ trang vieclam24h.vn 4. Database cập nhật việc làm.

7.7. Đặc tả UC dự đoán xu hướng việc làm

Bảng 13: Đặc tả UC dự đoán xu hướng việc làm

Mã ca sử dụng	UC07
Tên ca sử dụng	Dự đoán xu hướng việc làm.
Mô tả	Quản trị viên, người dùng có thể xem xu hướng công việc.
Tác nhân	Quản trị viên. Người dùng.
Điều kiện trước	Cập nhập vào hệ thống, vào chức năng dự đoán và điền đầy đủ thông tin của “job” và “city”.
Điều kiện sau	Hiển thị dự đoán nhu cầu việc làm của 12 tháng tiếp theo của job-city đã chọn.
Các bước thực hiện	<ol style="list-style-type: none"> 1. Cập nhập vào hệ thống. 2. Chọn chức năng dự đoán xu hướng công việc. 3. Điền đầy đủ các “job” và “city” cần dự đoán. 4. Hiển thị sơ đồ và giá trị dự đoán cho 12 tháng tiếp theo.

7.8. Đặc tả UC quản lý thông tin cá nhân

Bảng 14: Đặc tả UC quản lý thông tin cá nhân

Mã ca sử dụng	UC07
Tên ca sử dụng	QUản lý tài khoản.
Mô tả	Người dùng có thể quản lý tài khoản cá nhân.
Tác nhân	Người dùng.
Điều kiện trước	Người dùng đăng nhập vào hệ thống.
Điều kiện sau	Mọi thay đổi về thông tin cá nhân đã được lưu
Các bước thực hiện	<ol style="list-style-type: none"> 1. Người dùng truy cập vào quản lý tài khoản. 2. Người dùng có thể thay đổi thông tin cá nhân và có thể thay đổi mật khẩu tài khoản.

CHƯƠNG III. ỨNG DỤNG HỌC MÁY ĐỂ DỰ BÁO

1. Phát biểu vấn đề

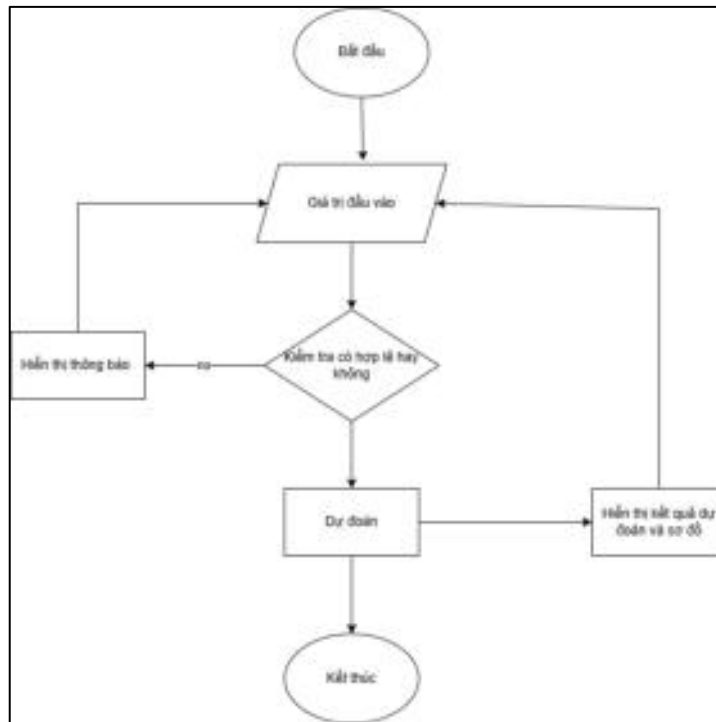
Mô tả đầu vào và đầu ra:

Đầu vào: Giá trị của 3 tham số: công việc, thành phố, nhu cầu công việc trong các tháng.

Đầu ra: Trả về kết quả dự đoán từ mô hình.

- Trường hợp 1: Mô hình nhận dạng các tham số đã gửi và sẽ trả về kết quả dự đoán và các bài đăng việc làm hiện có trong cơ sở dữ liệu.
- Trường hợp 2: Nếu các tham số không hợp lệ, sẽ báo lỗi và yêu cầu cung cấp thông tin hợp lệ để dự đoán.

Sơ đồ triển khai mô hình học máy được mô tả trong hình sau:

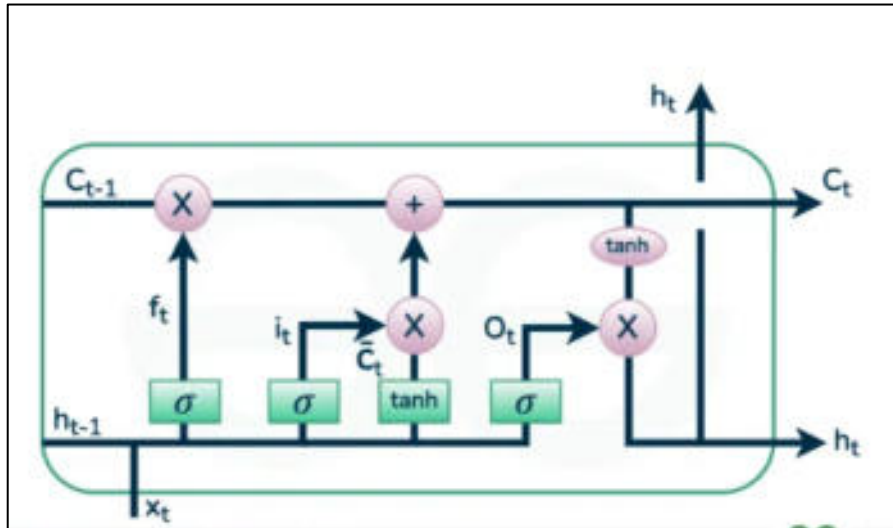


Hình 28: Sơ đồ triển khai mô hình học máy

2. Thuật toán

2.1. Long short-term memory (LSTM)

LSTM [5] được giới thiệu vào năm 1977 là một dạng mô hình Recurrent neural network (RNN) mà mạng network của nó được tổng hợp từ nhiều các đơn vị LSTM. Không giống như các mô hình RNN truyền thống, LSTM có khả năng ghi nhớ thông tin trong thời gian dài mà không có vấn đề "biến mất gradient".



Hình 29: Cấu trúc của một LSTM

Các đơn vị LSTM đều có điểm chung đó là được cấu tạo từ một cell, một input gate, một output gate và một forget gate.

Cell: sẽ có tác dụng ghi nhớ giá trị tùy ý của một khoảng thời gian bất kì và 3 công sẽ điều chỉnh giá trị của luồng thông tin vào và ra của cell.

Công: Bao gồm ba loại chính là input gate, forget gate và output gate, giúp điều chỉnh luồng thông tin:

- Input gate: Điều chỉnh thông tin mới sẽ được thêm vào trạng thái ô.
- Forget gate: Quyết định thông tin nào sẽ bị xóa khỏi trạng thái ô.
- Output gate: Xác định thông tin nào từ trạng thái tế bào sẽ được sử dụng để tạo đầu ra.

2.2. Cấu trúc và hoạt động của LSTM

Cell: là một đường truyền thông tin dài hạn, chạy qua toàn bộ chuỗi dữ liệu. Nó có thể truyền thông tin tương đối không thay đổi, trừ khi có sự can thiệp từ các công.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

Input gate: quyết định giá trị mới nào sẽ được cập nhật vào trạng thái ô. Nó sử dụng hàm sigmoid để điều chỉnh giá trị của thông tin mới và hàm tanh để tạo ra một vectơ các giá trị mới tiềm năng có thể được thêm vào trạng thái ô.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Forget gate: quyết định thông tin nào sẽ bị xóa khỏi trạng thái ô. Nó sử dụng hàm sigmoid để đánh giá tầm quan trọng của từng thông tin và quyết định giữ lại hay loại bỏ thông tin đó.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

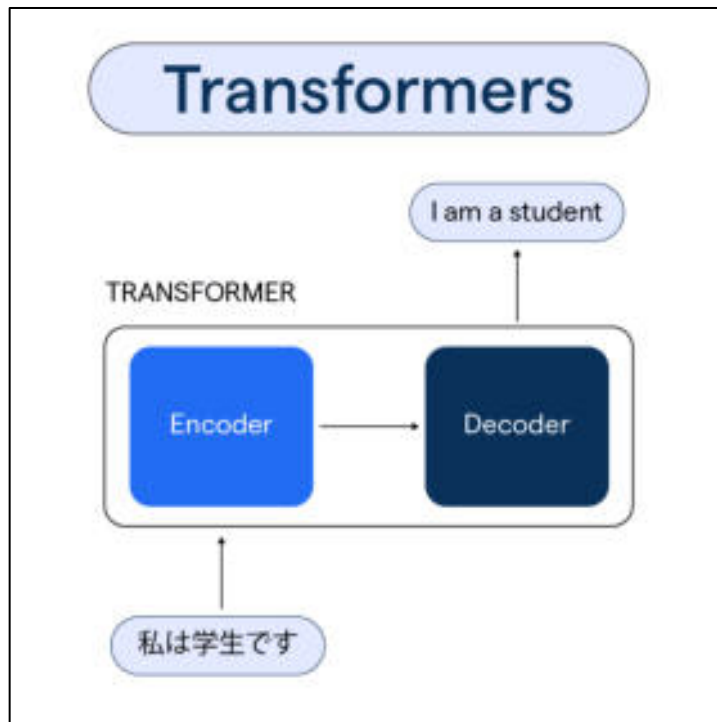
Output gate: quyết định giá trị nào từ trạng thái ô sẽ được sử dụng để tạo đầu ra cho bước hiện tại. Nó sử dụng hàm sigmoid để khớp và hàm tanh để tạo đầu ra cuối cùng.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

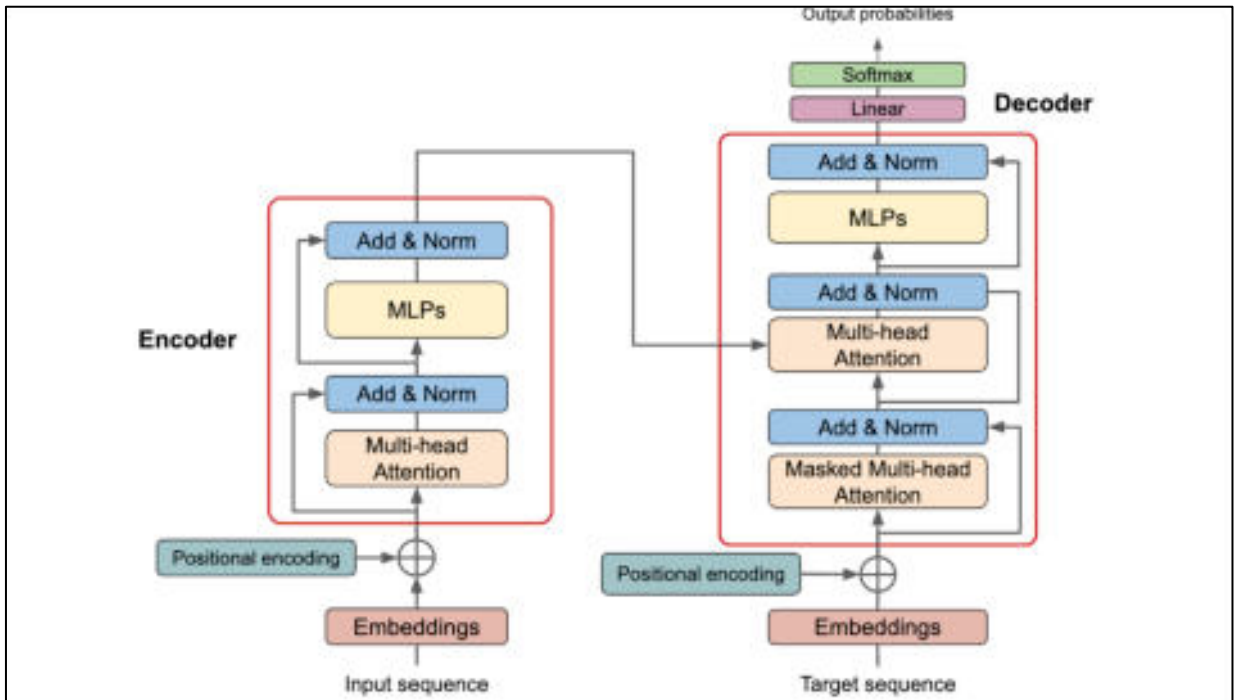
2.3. Thuật toán Transformer

Transformer là một kiến trúc mạng nơ-ron được giới thiệu trong bài báo "*Attention is All You Need*" (2017), được thiết kế đặc biệt để xử lý tuần tự dữ liệu nhưng không dùng mạng hồi tiếp (RNN) như LSTM.



Hình 30: Mô hình Transformer

2.4. Cấu trúc và hoạt động của Transformer



Hình 31: Mô hình thuật toán Transformer

Cấu trúc tổng quan của Transformer:

Transformer gồm hai phần chính:

1. **Encoder:** Nhận đầu vào và trích xuất đặc trưng.

2. **Decoder:** Dựa trên đặc trưng từ encoder và sinh ra đầu ra.

Mỗi encoder và decoder đều được xây dựng từ nhiều lớp giống nhau, gồm:

- **Encoder Layer:**

- **Multi-Head Self-Attention:** Giúp mô hình chú ý đến các vị trí khác nhau trong chuỗi đầu vào cùng lúc.

- **Feed Forward Network (FFN):** Mạng nơ-ron truyền thẳng được áp dụng cho từng vị trí riêng biệt.

- **Residual Connection + Layer Normalization:** Giúp chống trôi gradient và tăng hiệu quả huấn luyện.

- **Decoder Layer:**

- **Masked Multi-Head Attention:** Đảm bảo mô hình không "nhìn trước" tương lai trong lúc sinh đầu ra.

- **Encoder-Decoder Attention:** Cho phép decoder chú ý đến đầu ra từ encoder.

- **Feed Forward + Residual + Norm** như encoder.

Các thành phần chính:

- **Positional Encoding:** Vì không dùng RNN/CNN nên Transformer cần bổ sung thông tin thứ tự bằng các vector vị trí.
- **Attention (Scaled Dot-Product Attention):**

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

Với Q: truy vấn, K: khoá, V: giá trị.

- **Multi-Head Attention:** Thay vì thực hiện attention một lần, chia thành nhiều "đầu", mỗi đầu học một không gian khác nhau.

Bảng 15: So sánh LSTM và Transformer

Yếu tố	LSTM	Transformer
Kiểu kết nối	Tuần tự (sequential)	Song song (parallel)
Ghi nhớ dài hạn	Cell state	Attention
Tốc độ huấn luyện	Chậm (do tuần tự)	Nhanh hơn
Tối ưu độ dài chuỗi dài	Khó khăn	Hiệu quả với attention

3. Áp dụng học máy

3.1. Thực hiện và đánh giá

Transformer rất hiệu quả trong các vấn đề dự đoán chuỗi thời gian do khả năng ghi nhớ thông tin dài hạn và xử lý các mối quan hệ phức tạp trong dữ liệu. Dưới đây là các quy trình để giải quyết đề bài với việc sử dụng Transformer để dự đoán xu hướng việc làm:

3.1.1. Thu thập và xử lý dữ liệu

Thu thập dữ liệu chuỗi thời gian về tuyển dụng việc làm tại 3 thành phố lớn là Hà Nội, Đà Nẵng và Hồ Chí Minh.

	A	B	C	D	E	F	G
1	id	time	city	job	number_employees	major	level
2		1	1695908160000	Hà Nội	ReactJs		1 Công nghệ thông tin fresher
3		2	1685487066000	Hồ Chí Minh	Backend Developer		2 Công nghệ thông tin fresher
4		3	1689072982000	Hồ Chí Minh	NodeJs		2 Công nghệ thông tin senior
5		4	1714963190000	Hồ Chí Minh	DevOps Engineer		2 Công nghệ thông tin fresher
6		5	1683174809000	Hà Nội	Frontend Developer		2 Công nghệ thông tin senior
7		6	1677120936000	Hồ Chí Minh	Python Developer		3 Công nghệ thông tin junior
8		7	1695921879000	Hồ Chí Minh	DevOps Engineer		2 Công nghệ thông tin senior
9		8	1683552476000	Hà Nội	Python Developer		1 Công nghệ thông tin intern
10		9	1676637699000	Hồ Chí Minh	Backend Developer		1 Công nghệ thông tin senior
11		10	1683103577000	Hồ Chí Minh	DevOps Engineer		3 Công nghệ thông tin middle
12		11	1688908140000	Hà Nội	Data Scientist		2 Công nghệ thông tin fresher
13		12	1681753975000	Hà Nội	Data Scientist		3 Công nghệ thông tin senior
14		13	1679033506000	Hồ Chí Minh	Backend Developer		2 Công nghệ thông tin middle
15		14	1688867003000	Đà Nẵng	NodeJs		1 Công nghệ thông tin senior
16		15	1714719910000	Đà Nẵng	Java Developer		2 Công nghệ thông tin junior
17		16	1714342253000	Đà Nẵng	Backend Developer		3 Công nghệ thông tin middle
18		17	1691760786000	Đà Nẵng	Python Developer		3 Công nghệ thông tin senior
19		18	1705568071000	Đà Nẵng	Backend Developer		2 Công nghệ thông tin junior

Hình 32: Bộ dữ liệu chuẩn bị cho mô hình

Thực hiện các bước xử lý trước như: loại bỏ các cột không cần thiết ,chuẩn hóa dữ liệu và chia dữ liệu thành các khoảng thời gian thích hợp, gộp các nhóm job_city lại với nhau.

```
data = data.drop(columns=["id", "major", "level"])
data["time"] = pd.to_datetime(data["time"], unit="ms")
data["month"] = data["time"].dt.to_period("M")

# Gộp dữ liệu
# aggregated_data = data.groupby(["job", "city", "month"]).size().reset_index(name="job_postings_count")
data = data.pivot_table(index=["job", "city", "month"], aggfunc="size").reset_index(name="job_postings_count")

data["job_city"] = data["job"] + " - " + data["city"]
# Chuẩn hóa dữ liệu
# scaler = MinMaxScaler(feature_range=(0, 1))
data["job_postings_count_scaled"] = scaler.fit_transform(
    data["job_postings_count"].values.reshape(-1, 1))
}
```

Hình 33: Xử lý dữ liệu

Kết quả đạt được:

	job	city	month	job_postings_count	job_city	job_postings_count_scaled
0	Backend Developer	Hà Nội	2023-01	35	Backend Developer - Hà Nội	0.928571
1	Backend Developer	Hà Nội	2023-02	25	Backend Developer - Hà Nội	0.571429
2	Backend Developer	Hà Nội	2023-03	26	Backend Developer - Hà Nội	0.607143
3	Backend Developer	Hà Nội	2023-04	20	Backend Developer - Hà Nội	0.392857
4	Backend Developer	Hà Nội	2023-05	19	Backend Developer - Hà Nội	0.357143
...
403	ReactJs	Đà Nẵng	2024-01	17	ReactJs - Đà Nẵng	0.285714
404	ReactJs	Đà Nẵng	2024-02	24	ReactJs - Đà Nẵng	0.535714
405	ReactJs	Đà Nẵng	2024-03	17	ReactJs - Đà Nẵng	0.285714

Hình 34: Dữ liệu sau khi xử lý

	job	city	month	job_postings_count	job_postings_count_scaled
305	NodeJs	Đà Nẵng	2024-05	34	0.892857
304	NodeJs	Đà Nẵng	2024-04	27	0.642857
303	NodeJs	Đà Nẵng	2024-03	22	0.464286
302	NodeJs	Đà Nẵng	2024-02	14	0.178571
301	NodeJs	Đà Nẵng	2024-01	19	0.357143
300	NodeJs	Đà Nẵng	2023-12	29	0.714286
299	NodeJs	Đà Nẵng	2023-11	17	0.285714
298	NodeJs	Đà Nẵng	2023-10	33	0.857143
297	NodeJs	Đà Nẵng	2023-09	21	0.428571
296	NodeJs	Đà Nẵng	2023-08	24	0.535714
295	NodeJs	Đà Nẵng	2023-07	34	0.892857
294	NodeJs	Đà Nẵng	2023-06	30	0.750000
293	NodeJs	Đà Nẵng	2023-05	28	0.678571
292	NodeJs	Đà Nẵng	2023-04	30	0.750000
291	NodeJs	Đà Nẵng	2023-03	34	0.892857
290	NodeJs	Đà Nẵng	2023-02	28	0.678571

Hình 35: Dữ liệu của "NodeJs_Đà Nẵng"

3.1.2. Xây dựng mô hình Transformer

Bài toán đặt ra làm sao để giải quyết vấn đề từ tập dữ liệu lớn tôi có thể lọc ra dữ liệu của từng "job_city" và dự đoán xu hướng việc làm cho từng "job_city". Để giải quyết vấn đề này, sau khi xử lý dữ liệu mô hình sẽ duyệt qua từng "job_city" để lấy dữ liệu của từng "job_city" nhận thấy dữ liệu trong một "job_city" là nhiều cao nên có dùng thêm xác định xu hướng và mùa vụ [6] để làm mượt dữ liệu chuẩn hóa. Giá trị từ cột "job_postings_count_scaled" được lấy ra để phân tích và lấy giá trị (trend) để làm dữ liệu đầu vào cho mô hình.

```
# 2. Phân tích mùa vụ và xu hướng
seasonal_period = 6 # chu kỳ là 6 tháng
result = seasonal_decompose(job_data["job_postings_count_scaled"], model='additive', period=seasonal_period)

# Lấy phân xu hướng (trend) làm dữ liệu đầu vào
job_data["job_postings_trend"] = result.trend.fillna(method="bfill").fillna(method="ffill")

# Chọn dữ liệu xu hướng để sử dụng
data_to_use = job_data["job_postings_trend"].values.reshape(-1, 1)
```

Hình 36: Phân tích mùa vụ và xu hướng

Phân tích xu hướng và mùa vụ là một bước quan trọng trong xử lý dữ liệu chuỗi thời gian, giúp tách biệt các thành phần xu hướng (trend), mùa vụ (seasonality), và nhiễu (residual).

- Xu hướng (Trend): Giúp mô hình tập trung vào sự thay đổi dài hạn, loại bỏ ảnh hưởng ngắn hạn hoặc theo mùa vụ.
- Mùa vụ (Seasonality): Nếu không xử lý, các mẫu lặp lại theo chu kỳ có thể gây nhầm lẫn cho mô hình dự đoán.
- Nhiễu (Residual): Loại bỏ các yếu tố bất thường hoặc không liên quan, giúp cải thiện hiệu suất dự đoán.

- Thiết kế và cấu hình mô hình Transformer với các tham số phù hợp như số lớp, số đơn vị trong mỗi lớp, hàm mất mát và bộ tối ưu hóa.

	job_city	job_postings_trend
Backend Developer - Đà Nẵng		0.572917
Backend Developer - Đà Nẵng		0.572917
Backend Developer - Đà Nẵng		0.572917
Backend Developer - Đà Nẵng		0.572917
Backend Developer - Đà Nẵng		0.572917
Backend Developer - Đà Nẵng		0.572917
Backend Developer - Đà Nẵng		0.572917
Backend Developer - Đà Nẵng		0.569948
Backend Developer - Đà Nẵng		0.571429
Backend Developer - Đà Nẵng		0.556548
Backend Developer - Đà Nẵng		0.522321
Backend Developer - Đà Nẵng		0.504464
Backend Developer - Đà Nẵng		0.516369
Backend Developer - Đà Nẵng		0.535714
Backend Developer - Đà Nẵng		0.552083
Backend Developer - Đà Nẵng		0.543155
Backend Developer - Đà Nẵng		0.516369
Backend Developer - Đà Nẵng		0.497824
Backend Developer - Đà Nẵng		0.494048
Backend Developer - Đà Nẵng		0.483631

Hình 37: Giá trị trend của "Backend Developer_Đà Nẵng"

- Thiết kế và cấu hình mô hình Transformer với các tham số phù hợp như số lớp, số đơn vị trong mỗi lớp, hàm mất mát và bộ tối ưu hóa.

```

model = Sequential()

model.add(LSTM(units=250, return_sequences=True, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.3))
model.add(LSTM(units=200, return_sequences=False))
model.add(Dropout(0.3))
model.add(Dense(units=1))

model.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')
early_stopping = EarlyStopping(monitor='val_loss', patience=40, restore_best_weights=True)
history = model.fit(X_train, y_train, epochs=550, batch_size=16, validation_data=(X_val, y_val), callbacks=[early_stopping])
    
```

Hình 38: Mô hình LSTM

3.2. Chia dữ liệu

Dữ liệu được chia thành 3 phần train, test và validation.

```
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3, shuffle=False)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, shuffle=False)
```

Tập Train:

- Dùng để huấn luyện mô hình.
- Mô hình học từ dữ liệu này để điều chỉnh trọng số trong mạng Transformer, nhằm giảm thiểu hàm mất mát (loss).

Tập Validation:

- Được sử dụng để đánh giá hiệu suất của mô hình trong quá trình huấn luyện.
- Giúp mô hình điều chỉnh các siêu tham số (hyperparameters), như số lượng epoch hoặc cấu trúc của mô hình.
- Early stopping: Theo dõi hàm mất mát trên tập validation (val_loss) để quyết định dừng sớm nếu mô hình không còn cải thiện.

Tập Test:

- Được sử dụng để đánh giá hiệu suất của mô hình sau khi quá trình huấn luyện hoàn tất.
- Kết quả trên tập test cung cấp đánh giá trung thực về khả năng tổng quát hóa của mô hình đối với dữ liệu mới.
- Các chỉ số như MSE, MAE, và R^2 được tính trên tập này để báo cáo hiệu suất của mô hình.

3.3. Dự đoán

Dự đoán tháng tiếp theo dựa trên số liệu của 12 tháng trước đó:

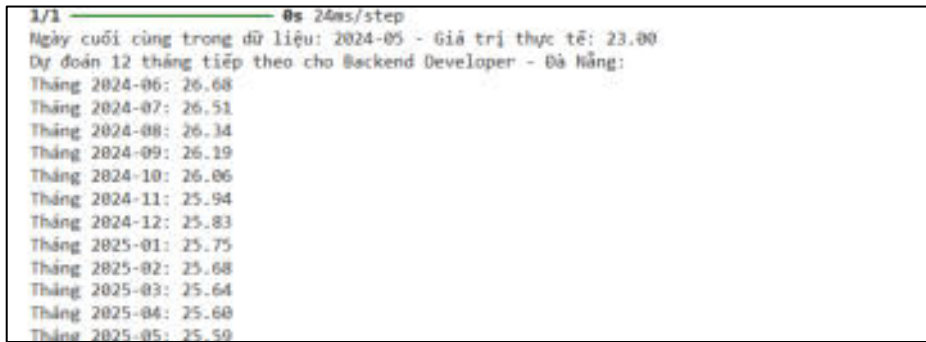
```
# 9. Dự đoán 12 tháng tiếp theo
last_sequence = X_test[-1].reshape(1, time_steps, 1) # Khởi tạo chuỗi cuối cùng
predicted_values = [] # Danh sách lưu giá trị dự đoán

for i in range(12):
    next_value = model.predict(last_sequence) # Dự đoán giá trị tiếp theo
    predicted_values.append(next_value.flatten()[0]) # Lưu giá trị dự đoán
    # Cập nhật last_sequence cho lần dự đoán tiếp theo
    last_sequence = np.append(last_sequence[:, 1:, :], next_value.reshape(1, 1, 1), axis=1)

# Chuyển giá trị dự đoán về giá trị gốc (nếu cần)
predicted_values_scaled = scaler.inverse_transform(np.array(predicted_values).reshape(-1, 1))

# Tạo danh sách ngày tháng cho 12 tháng tiếp theo
last_date = pd.to_datetime(job_data.iloc[-1]['month'])
future_dates = [last_date + pd.DateOffset(months=i+1) for i in range(12)]
```

Hình 39: Dự đoán xu hướng



Hình 40: Kết quả thu được

Đánh giá độ chính xác trên tập train và tập test bằng giá trị mse, mae và r:

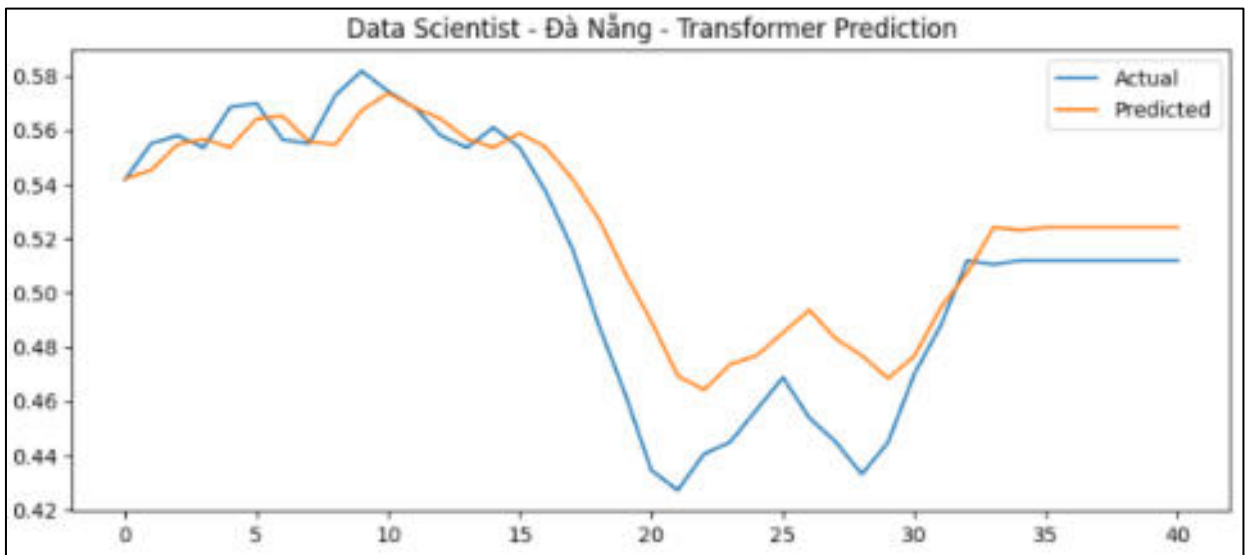
```

# 7. Đánh giá mô hình trên tập huấn luyện
train_predictions = model.predict(X_train)
train_mse = mean_squared_error(y_train, train_predictions)
train_mae = mean_absolute_error(y_train, train_predictions)
train_r2 = r2_score(y_train, train_predictions)

print(f"Training Mean Squared Error: {train_mse}")
print(f"Training Mean Absolute Error: {train_mae}")
print(f"Training R2 Score: {train_r2}")

# Vẽ đồ thị so sánh giữa giá trị thực tế và giá trị dự đoán trên tập huấn luyện
plt.figure(figsize=(12,6))
plt.plot(y_train, label='Actual (Train)')
plt.plot(train_predictions, label='Predicted (Train)')
plt.title(f'LSTM Model Predictions vs Actual (Train) - {job_city}')
plt.xlabel('Time Steps')
plt.ylabel('Job Postings Count (Scaled)')
plt.legend()
plt.show()
    
```

Hình 41: Đánh giá độ chính xác trên tập train



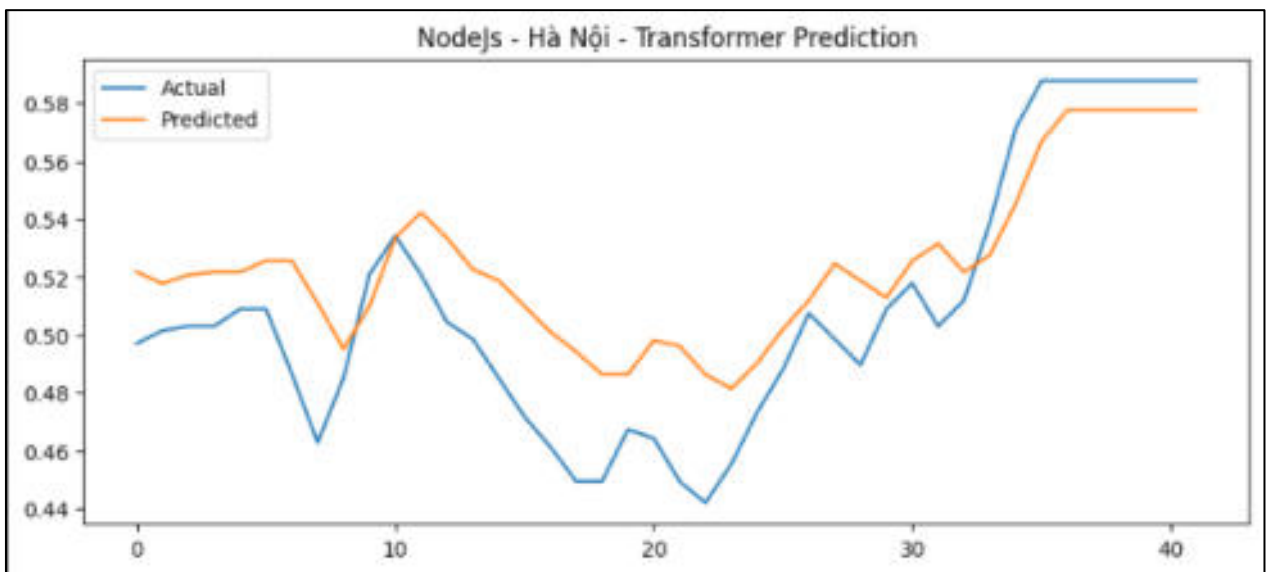
Hình 42: Dự đoán và độ chính xác trên tập train của “Data Scientist – Đà Nẵng”

```
# 8. Đánh giá mô hình trên tập test
test_predictions = model.predict(X_test)
test_mse = mean_squared_error(y_test, test_predictions)
test_mae = mean_absolute_error(y_test, test_predictions)
test_r2 = r2_score(y_test, test_predictions)

print(f"Test Mean Squared Error: {test_mse}")
print(f"Test Mean Absolute Error: {test_mae}")
print(f"Test R2 Score: {test_r2}")

# Vẽ đồ thị so sánh giữa giá trị thực tế và giá trị dự đoán trên tập test
plt.figure(figsize=(12,6))
plt.plot(y_test, label='Actual (Test)')
plt.plot(test_predictions, label='Predicted (Test)')
plt.title(f'LSTM Model Predictions vs Actual (Test) - (job_city)')
plt.xlabel('Time Steps')
plt.ylabel('Job Postings Count (Scaled)')
plt.legend()
plt.show()
```

Hình 43: Đánh giá độ chính xác trên tập test



Hình 44: Dự đoán và độ chính xác trên tập test của “Nodejs – Hà Nội”

3.4. Kết quả

3.4.1. Thang đo MAE

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Divide by the total number of data points
Actual output value
Predicted output value
Sum of
The absolute value of the residual

Hình 45: Công thức MAE

3.4.2. Thang đo MSE

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Mean
Error
Squared

Hình 46: Công thức MSE

3.4.3. Thang đo F1 Score

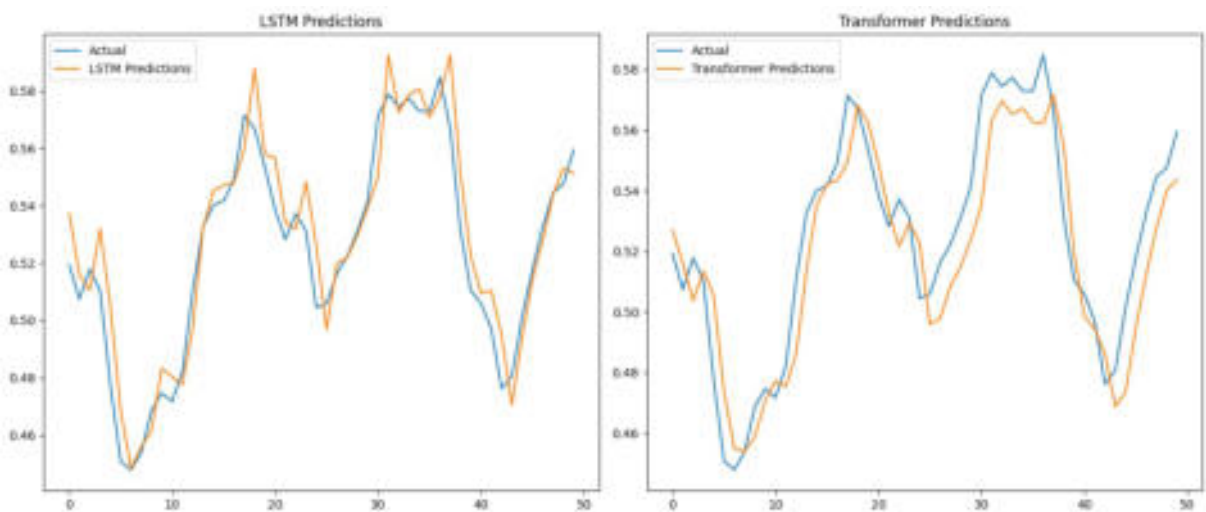
$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Hình 47: Công thức F1 Score

3.4.4. Kết quả của mô hình

Model Comparison:				
	Model	MAE	MSE	F1 Score
0	LSTM	0.009963	0.000150	0.597285
1	Transformer	0.014963	0.000344	0.492228

Hình 48: Kết quả thang đo của 2 mô hình



Hình 49: Kiểm thử với tập test

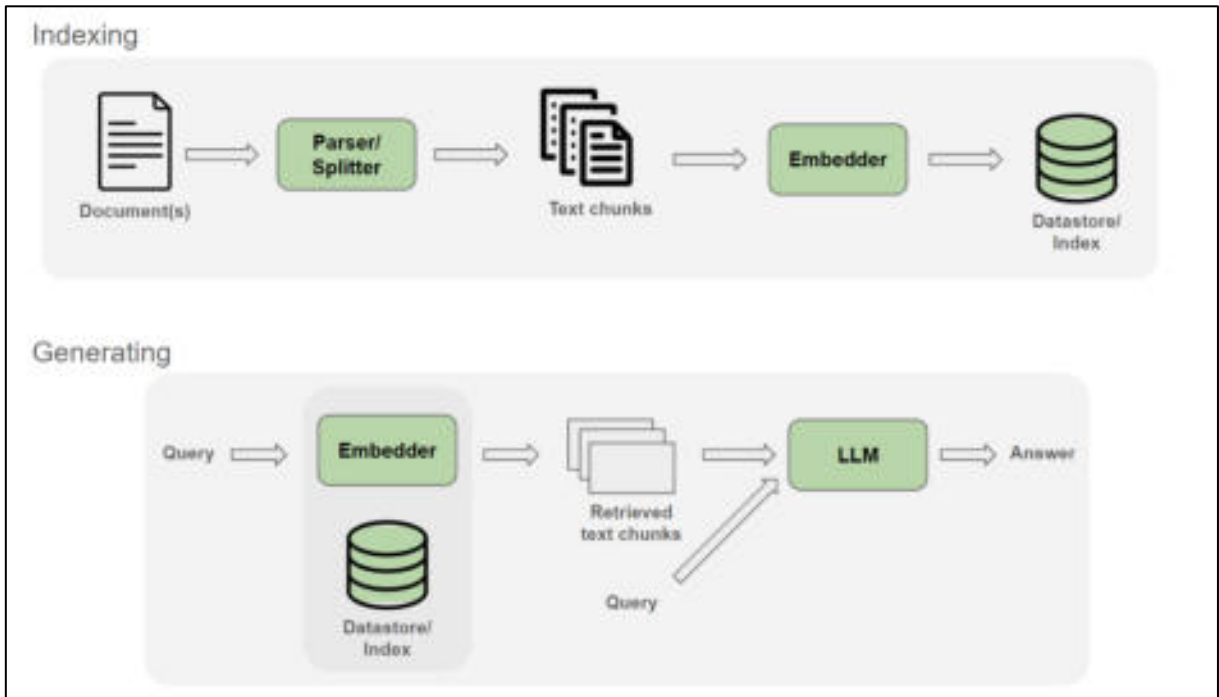
Mặc dù LSTM có sai số thấp hơn, nhưng Transformer được chọn vì thể hiện khả năng nắm bắt xu hướng dài hạn tốt hơn, phù hợp cho dự đoán xu hướng việc làm.

CHƯƠNG IV. Thiết kế Chatbot AI

1. Mục tiêu

Chatbot hỗ trợ người dùng tư vấn việc làm phù hợp dựa trên kỹ năng, sở thích, xu hướng thị trường và mô hình học máy đã triển khai. Chatbot còn có thể truy xuất thông tin từ dữ liệu tuyển dụng hiện có.

2. Kiến trúc tổng quan



Hình 50: Kiến trúc RAG

Chatbot sử dụng kiến trúc Retrieval-Augmented Generation (RAG) bao gồm:

- Retrieval (Tìm kiếm):

- Sử dụng **MongoDB Atlas Vector Search** làm VectorDB để lưu trữ và truy vấn ngữ nghĩa.
- Dữ liệu đầu vào bao gồm mô tả công việc, kỹ năng, yêu cầu — đã được embed bằng all-MiniLM-L6-v2.

- Re-ranking

- Sau khi lấy ra danh sách top-k văn bản từ VectorDB, sử dụng **reranker** để sắp xếp lại dựa trên mức độ liên quan đến câu truy vấn người dùng.

- Generation (Sinh câu trả lời):

- LLM **Gemini 2 Flash** sẽ sử dụng thông tin đã truy xuất từ bước trên để sinh ra câu trả lời tự nhiên, có ngữ cảnh.

3. Pipeline hoạt động

1. Người dùng đặt câu hỏi (ví dụ: "Tôi nên chọn ngành gì nếu tôi giỏi Python?").
2. Câu hỏi được embed và tìm top-k document trong MongoDB.
3. Reranker sắp xếp lại các document theo độ liên quan.
4. Gemini 2 Flash tạo ra câu trả lời từ nội dung đã được truy xuất + câu hỏi người dùng.
5. Trả về kết quả cho frontend.

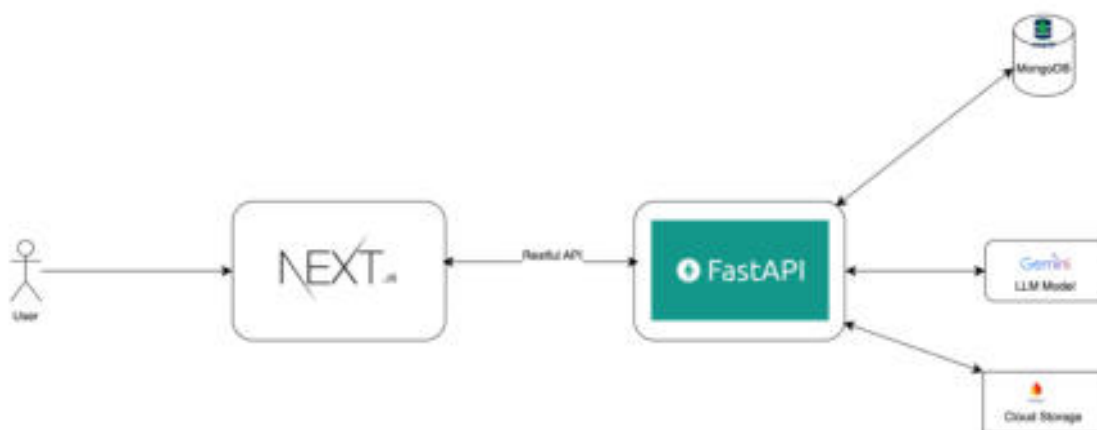
CHƯƠNG V. KẾT QUẢ THỰC HIỆN VÀ ĐÁNH GIÁ

1. Triển khai

Dưới đây là sơ đồ hệ thống sau khi đóng gói và triển khai lên Azure. Hệ thống bao gồm 5 phần chính:

- Frontend được xây dựng bằng NextJS
- Backend được xây dựng trên nền tảng ngôn ngữ Python sử dụng framework FastAPI .
- Dữ liệu Server Crawl được xây dựng bằng ngôn ngữ python, sử dụng thư viện selenium để tiến hành thu thập dữ liệu từ các trang ITViec, TopCV, TopDev
- Máy chủ người dùng được xây dựng bằng ngôn ngữ Python, sử dụng nền tảng FastAPI.
- Frontend và Server đều là các container riêng biệt được liên kết trực tiếp với nhau, giúp việc trao đổi giữa các thành phần nhanh hơn và dễ dàng hơn. Cơ sở dữ liệu được lưu trữ bằng MongoDB và kết nối trực tiếp với Backend.

2. Mô hình hệ thống



Hình 51: Khai triển cấu trúc API

3. Kết quả thực hiện

3.1. Trang đăng nhập

The screenshot shows the 'Sign in to your account' page. At the top, there is a navigation bar with 'IT Job Board' on the left, 'Find Jobs', 'AI Chat', and 'Job Trend' in the center, and 'Sign In' and 'Sign Up' buttons on the right. The main content area features a white box with the title 'Sign in to your account' and the instruction 'Enter your email and password to access the job board!'. Below this, there are input fields for 'Email' (containing 'name@example.com') and 'Password' (with a 'Forgot password?' link). A checkbox for 'Remember me for 30 days' is present. A green 'Sign In' button is at the bottom of the form. Below the button, there is a link for 'Don't have an account? Sign up' and a small disclaimer: 'By signing in, you agree to our Terms of Service and Privacy Policy'.

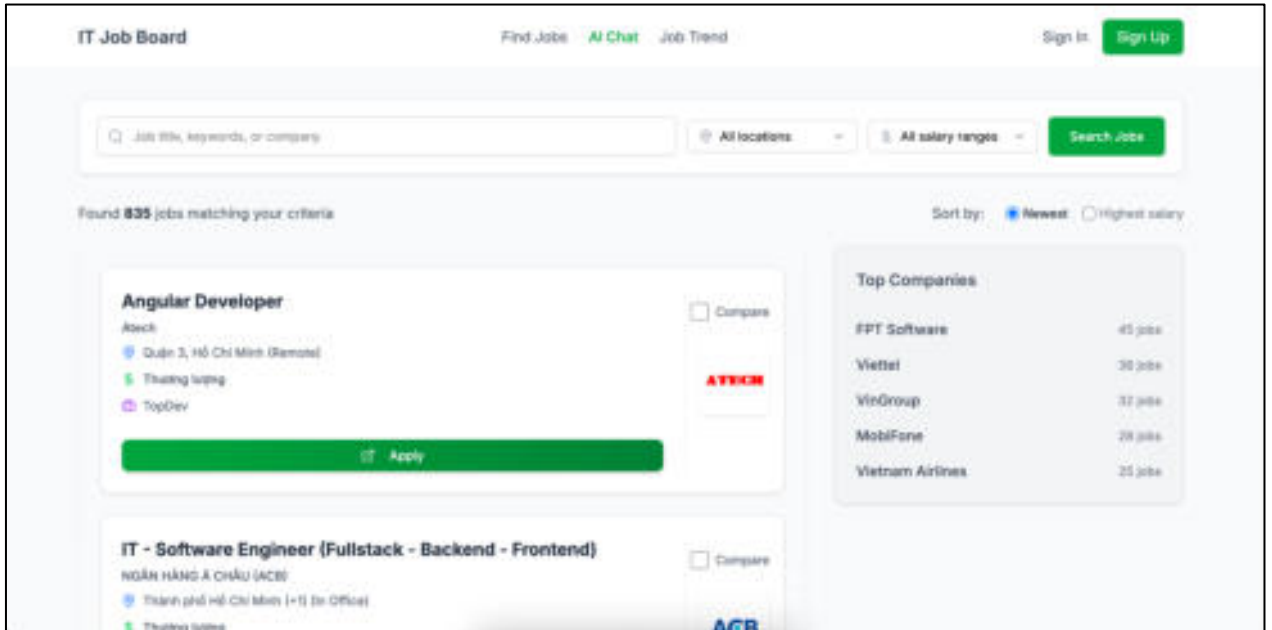
Hình 52: Trang đăng nhập

3.2. Trang đăng ký

The screenshot shows the 'Create an account' page. The navigation bar is identical to the previous page. The main content area features a white box with the title 'Create an account' and the instruction 'Enter your information to create an account!'. Below this, there are input fields for 'Username' (containing 'johnson'), 'Email' (containing 'name@example.com'), 'Password', and 'Confirm password'. A checkbox for 'I agree to the Terms of Service and Privacy Policy' is present. A green 'Create account' button is at the bottom of the form. Below the button, there is a link for 'Already have an account? Sign in'.

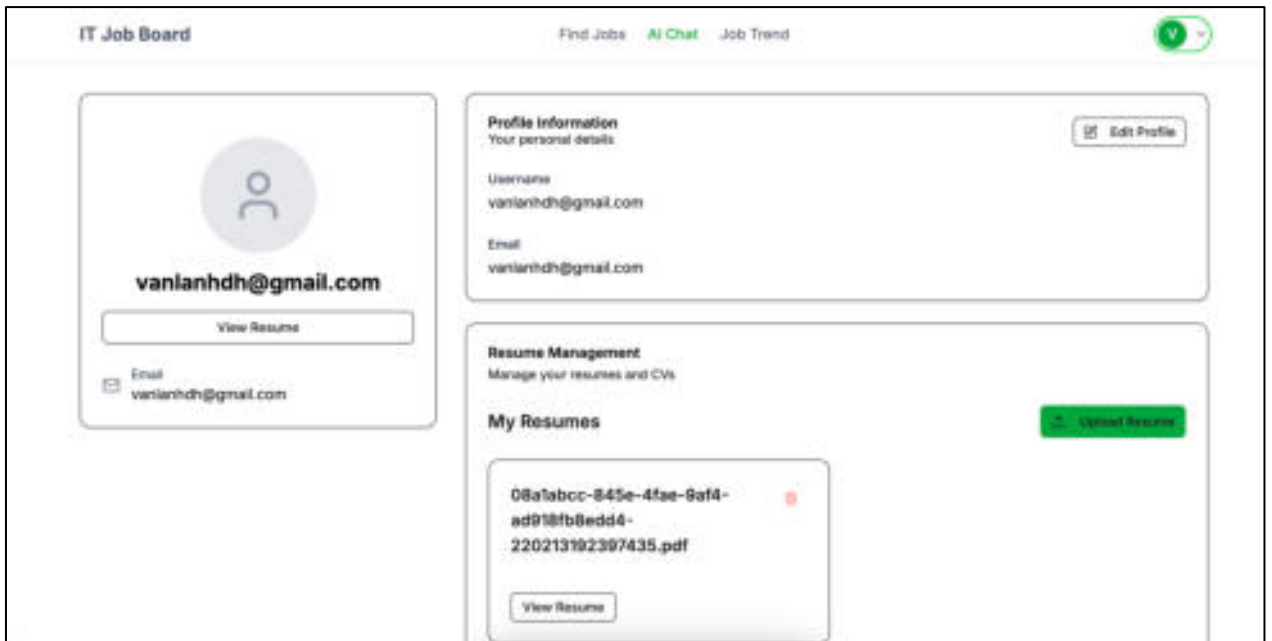
Hình 53: Trang đăng kí

3.3. Trang chủ



Hình 54: Trang chủ

3.4. Thiết lập người dùng



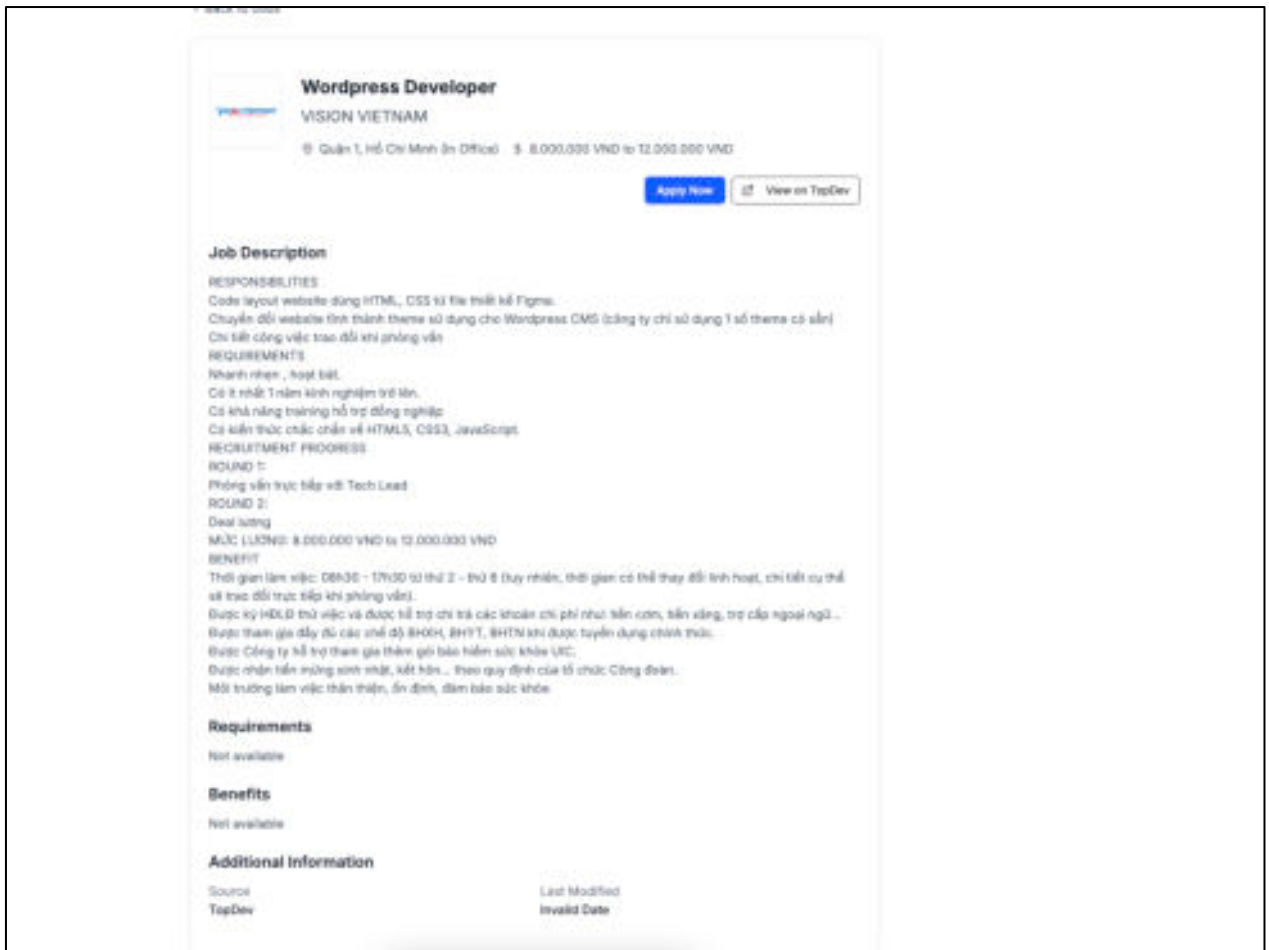
Hình 55: Quản lý đăng kí

3.5. Chatbot



Hình 56: Chatbot tư vấn cho người dùng

3.6. Trang chi tiết công việc



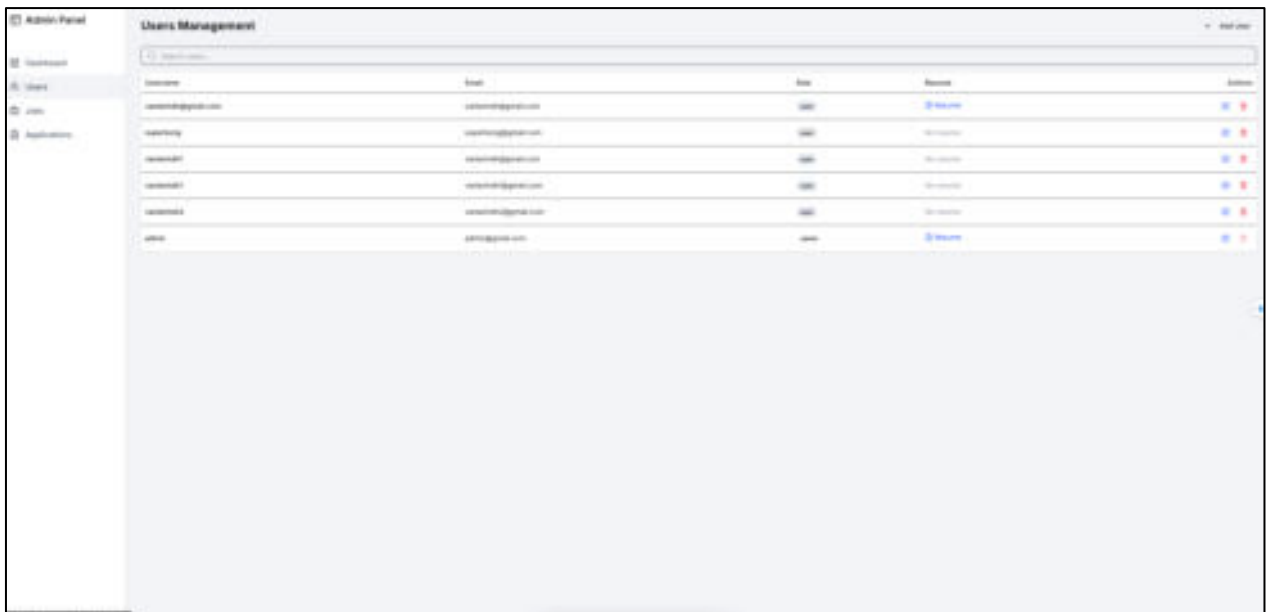
Hình 57: Trang đăng tin tuyển dụng

3.7. Trang dự đoán xu hướng việc làm



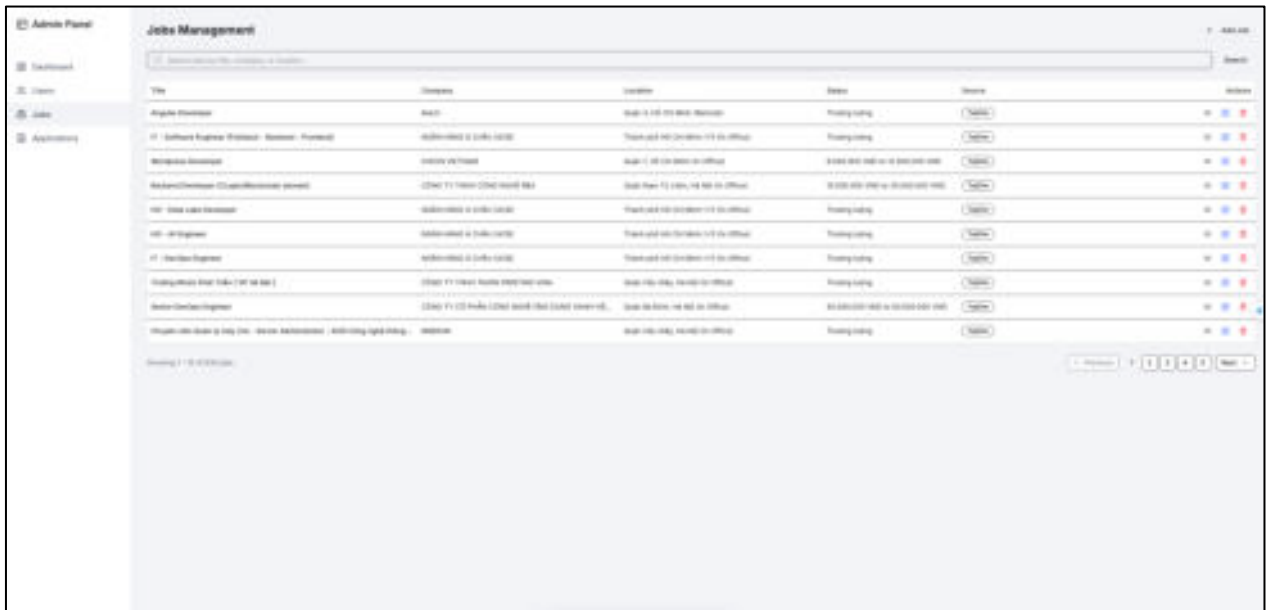
Hình 58: Dự đoán xu hướng việc làm cho “Backend Developer _Hà Nội”

3.8. Quản lý



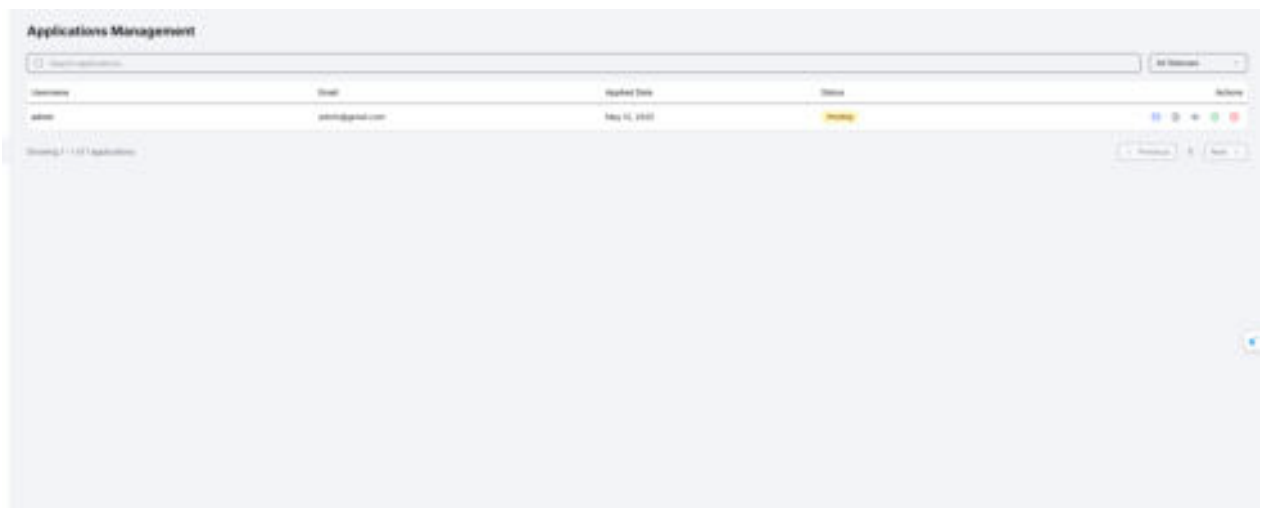
Hình 59: Trang quản lý người dùng

3.9. Quản lý đơn ứng tuyển



Hình 60: Trang quản lý jobs

3.10. Quản lý đơn ứng tuyển



Hình 61: Quản lý đơn ứng tuyển

4. Kết chương

Chương này đóng vai trò quan trọng trong việc thực hiện và đánh giá kết quả. Trong quá trình trên tôi đã đạt học được nhiều kiến thức cho bản thân .

Quá trình triển khai diễn ra một cách suôn sẻ và có kế hoạch. Giao diện thân thiện dễ tiếp cận, mô hình học máy có độ chính xác cao đáng ứng những chức năng tổng qua cho người dùng.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết quả đạt được

- Thông qua dự án này, tôi đã hiểu rõ hơn về quy trình triển khai AI và cách huấn luyện mô hình.
- Tìm hiểu thêm kiến thức về ngôn ngữ lập trình và các phân lý thuyết về những phần mới, chưa được tiếp cận trước đây.
- Trang web đã đáp ứng hầu hết các yêu cầu đã nêu, các trang được quản lý chặt chẽ, có thể chuyển đổi qua lại giữa các màn hình.
- Giao diện thân thiện, dễ sử dụng.

2. Các vấn đề tồn tại

- Hệ thống chưa được phát hành để thử nghiệm đầy đủ các chức năng.
- Dự án còn ít chức năng cần cải thiện.
- Model hiện tại chỉ dự đoán được mình nhu cầu công việc còn những giá trị khác như mức lương, tuổi tác,... thì chưa có, cần cải thiện thêm.

3. Hướng phát triển

- Tiếp tục xây dựng hệ thống ngày càng hoàn thiện hơn để có thể đưa vào thực tế.
- Xây dựng hệ thống đề xuất sản phẩm phù hợp với công việc và kỹ năng của người dùng.
- Tìm thêm nhiều tập dữ liệu dài hơn và nhiều hơn.

TÀI LIỆU THAM KHẢO

- [1] https://www.w3schools.com/html/html_css.asp
- [2] <https://www.w3schools.com/js/>
- [3] <https://www.w3schools.com/python/>
- [4] <https://aws.amazon.com/vi/docker/>
- [5] [LSTM/LSTM.ipynb at master · phamdinhhkhanh/LSTM · GitHub](#)
- [6] <https://viblo.asia/p/phan-tich-du-lieu-chuoi-thoi-gian-voi-python-3kY4g0K0JAe>
- [7] <https://learn.microsoft.com/en-us/azure/virtual-machines/overview>