

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM**



**MA THỊ NGỌC QUỲNH – 21110918
HUỲNH THỊ TÓ TRINH – 21110932**

Đề tài:

**XÂY DỰNG WEBSITE TUYỂN DỤNG VÀ TÌM KIẾM
VIỆC LÀM SỬ DỤNG REACTJS VÀ NODEJS**

TIỂU LUẬN CHUYÊN NGÀNH KỸ SƯ CNTT

**GIẢNG VIÊN HƯỚNG DẪN
TS. LÊ VĨNH THỊNH**

KHÓA 2021 - 2025

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM



MA THỊ NGỌC QUỲNH – 21110918

HUỲNH THỊ TỔ TRINH – 21110932

Đề tài:

**XÂY DỰNG WEBSITE TUYỂN DỤNG VÀ TÌM KIẾM
VIỆC LÀM SỬ DỤNG REACTJS VÀ NODEJS**

TIỂU LUẬN CHUYÊN NGÀNH KỸ SƯ CNTT

GIẢNG VIÊN HƯỚNG DẪN

TS. LÊ VĨNH THỊNH

KHÓA 2021 – 2025

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên 1: **Ma Thị Ngọc Quỳnh**

MSSV 1: **21110918**

Họ và tên Sinh viên 2: **Huỳnh Thị Tố Trinh**

MSSV 2: **21110932**

Ngành: **Công nghệ Thông tin**

Tên đề tài: **Xây dựng website tuyển dụng và tìm kiếm việc làm sử dụng Reactjs và Nodejs**

Họ và tên Giáo viên hướng dẫn: **TS. Lê Vĩnh Thịnh**

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.....

.....

.....

2. Ưu điểm:

.....

.....

.....

3. Khuyết điểm

.....

.....

.....

4. Đề nghị cho bảo vệ hay không?

.....

5. Đánh giá loại:

6. Điểm:

Tp. Hồ Chí Minh, ngày tháng năm 2024

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên Sinh viên 1: **Ma Thị Ngọc Quỳnh**

MSSV 1: **21110918**

Họ và tên Sinh viên 2: **Huỳnh Thị Tố Trinh**

MSSV 2: **21110932**

Ngành: **Công nghệ Thông tin**

Tên đề tài: **Xây dựng website tuyển dụng và tìm kiếm việc làm sử dụng Reactjs và Nodejs**

Họ và tên Giáo viên phản biện: **TS. Huỳnh Xuân Phụng**

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.....

.....

.....

2. Ưu điểm:

.....

.....

.....

3. Khuyết điểm

.....

.....

.....

4. Đề nghị cho bảo vệ hay không?

.....

5. Đánh giá loại:

6. Điểm:

Tp. Hồ Chí Minh, ngày tháng năm 2024

Giáo viên phản biện

(Ký & ghi rõ họ tên)

LỜI CẢM ƠN

Trước hết, nhóm chúng em xin bày tỏ lòng biết ơn chân thành đến Khoa Công Nghệ Thông Tin – Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh. Chính môi trường học tập năng động, sự hỗ trợ tận tâm và nguồn tri thức phong phú từ Khoa đã trang bị cho chúng em nền tảng vững chắc để không ngừng học hỏi và sáng tạo. Những điều này đã góp phần quan trọng giúp nhóm hoàn thành đề tài “Xây dựng website tuyển dụng và tìm kiếm việc làm sử dụng React.js và Node.js”.

Đặc biệt, nhóm xin gửi lời tri ân sâu sắc đến thầy hướng dẫn, TS. Lê Vĩnh Thịnh. Thầy không chỉ đồng hành mà còn là nguồn động lực lớn lao trong suốt quá trình thực hiện đề tài. Những định hướng rõ ràng, những lời khuyên sâu sắc cùng sự tận tụy của thầy đã giúp nhóm vượt qua khó khăn, phát triển tư duy và hoàn thiện đề tài. Những bài học từ thầy sẽ mãi là hành trang quý giá đối với chúng em, không chỉ trong học tập mà còn trên hành trình sự nghiệp sau này.

Chúng em cũng xin gửi lời cảm ơn chân thành đến các thầy cô trong Khoa, những người đã tận tình truyền đạt kiến thức và chia sẻ kinh nghiệm quý báu. Đây chính là nền tảng giúp chúng em tự tin thực hiện đề tài, đồng thời chuẩn bị cho những thử thách trong tương lai.

Dù đã dành nhiều tâm huyết để hoàn thiện đề tài, chúng em nhận thức rõ rằng vẫn còn những thiếu sót và hạn chế. Vì vậy, chúng em rất mong nhận được sự góp ý từ quý thầy cô để có thể cải thiện và không ngừng hoàn thiện đề tài.

Một lần nữa, chúng em xin chân thành cảm ơn tất cả những người đã hỗ trợ, đồng hành và tạo điều kiện để nhóm hoàn thành đề tài này.

Nhóm thực hiện

Ma Thị Ngọc Quỳnh

Huỳnh Thị Tố Trinh

ĐỀ CƯƠNG TIỂU LUẬN CHUYÊN NGÀNH CÔNG NGHỆ PHẦN MỀM

Họ và tên Sinh viên 1: Ma Thị Ngọc Quỳnh

MSSV 1: 21110918

Họ và tên Sinh viên 2: Huỳnh Thị Tố Trinh

MSSV 2: 21110932

Thời gian làm tiểu luận: 19/08/2024 – 08/12/2024

Chuyên ngành: Công nghệ phần mềm

Tên đề tài: *Xây dựng website tuyển dụng và tìm kiếm việc làm sử dụng Reactjs và Nodejs*

Họ và tên Giáo viên hướng dẫn: TS. Lê Vĩnh Thịnh

Nhiệm vụ của báo cáo:

1. Lý thuyết:

- Nghiên cứu và tìm hiểu các công nghệ chính bao gồm MERN Stack, Json Web Token (JWT), Socket, và Cloudinary.

2. Thực hành:

- Xây dựng các API cho hệ thống bằng Restful API, Node.js, và Express.js.
- Thiết kế và phát triển giao diện người dùng web bằng React.js.
- Sử dụng cơ sở dữ liệu MongoDB để lưu trữ và quản lý dữ liệu.
- Tích hợp Json Web Token (JWT) để xác thực và cấp quyền, đảm bảo tính bảo mật và hiệu quả khi sử dụng hệ thống API.
- Triển khai Socket để thiết lập kết nối thời gian thực giữa các người dùng.
- Ứng dụng Cloudinary để quản lý và lưu trữ tệp hình ảnh, giúp tối ưu hóa hiệu suất hệ thống.

MỤC LỤC

PHẦN 1: MỞ ĐẦU	1
1. Tính cấp thiết của đề tài	1
2. Đối tượng nghiên cứu.....	2
3. Phạm vi nghiên cứu.....	3
4. Kết quả dự kiến đạt được	3
PHẦN 2: NỘI DUNG.....	4
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.....	4
1.1. KIẾN TRÚC HỆ THỐNG	4
1.2. REACTJS	4
1.2.1. ReactJS là gì?	4
1.2.2. Kiến trúc của ReactJS	5
1.2.3. Cách hoạt động của ReactJS	5
1.2.3.1. Component-Based Development (Phát triển dựa trên component):.....	5
1.2.3.2. Virtual DOM:.....	5
1.2.3.3. <i>One-Way Data Flow (Luồng dữ liệu một chiều):</i>	5
1.2.3.4. React State and Props:	5
1.2.4. Đặc điểm nổi bật của ReactJS	6
1.2.4.1. Component-Based Architecture (Kiến trúc dựa trên Component)	6
1.2.4.2. Virtual DOM (Document Object Model Ảo)	6
1.2.4.3. One-Way Data Binding (Ràng buộc dữ liệu một chiều).....	6
1.2.4.4. JSX (JavaScript XML)	6
1.2.4.5. Ecosystem & Community (Hệ sinh thái và cộng đồng).....	6
1.2.5. Ưu điểm của ReactJS	7
1.2.5.1. Hiệu suất cao.....	7
1.2.5.2. Tái sử dụng component	7
1.2.5.3. Học tập dễ dàng.....	7

1.2.5.4. Hỗ trợ từ các công ty lớn.....	7
1.2.6. Ứng dụng của ReactJS	7
1.2.6.1. Ứng dụng web một trang (Single Page Applications - SPA)	7
1.2.6.2. Ứng dụng di động với React Native	7
1.2.6.3. Thương mại điện tử và nội dung động.....	7
1.2.6.4. Dashboard và công cụ phân tích dữ liệu	7
1.3. NODEJS	8
1.3.1. Node.js là gì?.....	8
1.3.2. Đặc điểm nổi bật của Node.js.....	8
1.3.2.1. Single-threaded, Event-driven (Luồng đơn, điều khiển sự kiện):	8
1.3.2.2. Non-blocking I/O (Xử lý I/O không chặn):.....	8
1.3.2.3. NPM (Node Package Manager):.....	8
1.3.2.4. Cross-platform (Đa nền tảng):	8
1.3.3. Ưu điểm của Node.js	8
1.3.3.1. Hiệu suất cao:.....	8
1.3.3.2. Xử lý đồng thời:	8
1.3.3.3. Cộng đồng lớn:.....	8
1.3.3.4. Dễ học:.....	9
1.3.4. Ứng dụng của Node.js	9
1.4. EXPRESS	9
1.4.1. Express.js là gì?	9
1.4.2. Cách hoạt động của Express.js.....	9
1.4.3. Đặc điểm nổi bật của Express.js.....	9
1.4.3.1. Tối giản nhưng mạnh mẽ:.....	9
1.4.3.2. Hỗ trợ Routing linh hoạt:.....	10
1.4.3.3. Middleware:	10
1.4.3.4. Tích hợp dễ dàng:.....	10

1.4.3.5. Khả năng mở rộng:.....	10
1.4.4. Ưu điểm của Express.js.....	10
1.4.4.1. Nhanh và nhẹ:	10
1.4.4.2. Hỗ trợ mạnh mẽ từ cộng đồng:	10
1.4.4.3. Tích hợp tốt với Node.js:.....	10
1.4.4.4. Đa dụng:.....	10
1.4.5. Ứng dụng của Express.js.....	10
1.4.5.1. Phát triển API RESTful:	10
1.4.5.2. Ứng dụng web:.....	10
1.4.5.3. Ứng dụng thời gian thực:.....	11
1.4.5.4. Hệ thống quản lý nội dung:	11
1.5. MONGODB	11
1.5.1. MongoDB là gì?	11
1.5.2. Cách hoạt động của MongoDB	11
1.5.2.1. Tài liệu (Documents):.....	11
1.5.2.2. Bộ sưu tập (Collections):.....	11
1.5.2.3. Truy vấn và Tìm kiếm:	11
1.5.2.4. Replication và Sharding:	11
1.5.3. Đặc điểm nổi bật của MongoDB	11
1.5.3.1. Dữ liệu không có cấu trúc cố định (Schema-less):	11
1.5.3.2. Lưu trữ dữ liệu dưới dạng JSON/BSON:.....	12
1.5.3.3. Mở rộng dễ dàng (Scalability):	12
1.5.3.4. Tính nhất quán và tính sẵn sàng cao (High Availability):	12
1.5.3.5. Tìm kiếm mạnh mẽ:	12
1.5.4. Ưu điểm của MongoDB.....	12
1.5.4.1. Linh hoạt trong việc lưu trữ dữ liệu:.....	12
1.5.4.2. Khả năng mở rộng dễ dàng:.....	12

1.5.4.3. Cập nhật nhanh và dễ dàng:.....	12
1.5.4.4. Cộng đồng mạnh mẽ và hệ sinh thái lớn:	12
1.5.5. Ứng dụng của MongoDB.....	12
1.5.5.1. Ứng dụng Web và Di động:.....	12
1.5.5.2. Big Data và Phân tích Dữ liệu:	13
1.5.5.3. Dữ liệu không cấu trúc:	13
1.5.5.4. Internet of Things (IoT):.....	13
1.6. JSON WEB TOKEN.....	13
1.6.1. JSON Web Token (JWT) là gì?	13
1.6.2. Cấu trúc của JWT	13
1.6.3. Cách hoạt động của JWT	14
1.6.4. Ưu điểm của JWT.....	14
1.6.4.1. Tính bảo mật cao:.....	14
1.6.4.2. Độc lập với cơ sở dữ liệu:.....	14
1.6.4.3. Hỗ trợ phân quyền:.....	14
1.6.4.4. Phát hành và sử dụng dễ dàng:	14
1.6.5. Ứng dụng của JWT.....	14
1.6.5.1. Xác thực người dùng (Authentication):	14
1.6.5.2. Phân quyền truy cập (Authorization):	14
1.6.5.3. Tích hợp giữa các dịch vụ (Service-to-Service Authentication):	15
1.6.5.4. Single Sign-On (SSO):	15
1.7. CLOUDINARY	15
1.7.1. Cloudinary là gì?	15
1.7.2. Đặc điểm nổi bật của Cloudinary	15
1.7.2.1. Quản lý hình ảnh và video.....	15
1.7.2.2. Xử lý và chỉnh sửa media.....	15
1.7.2.3. Tối ưu hóa hiệu suất.....	15

1.7.2.4. CDN (Content Delivery Network).....	15
1.7.2.5. Tích hợp API mạnh mẽ	16
1.7.2.6. Bảo mật.....	16
1.7.2.7. Hỗ trợ video	16
1.7.3. Ưu điểm của Cloudinary	16
1.7.3.1. Tiết kiệm thời gian và công sức:.....	16
1.7.3.2. Hiệu suất cao:.....	16
1.7.3.3. Quản lý tập trung:.....	16
1.7.3.4. Tính linh hoạt cao:	16
1.7.3.5. Bảo mật và kiểm soát quyền truy cập:	16
1.7.4. Ứng dụng của Cloudinary	16
1.7.4.1. Xử lý và quản lý hình ảnh cho ứng dụng web và di động.....	16
1.7.4.2. Tối ưu hóa video	17
1.7.4.3. Quản lý các tài nguyên đa phương tiện trong các nền tảng thương mại điện tử.....	17
1.7.4.4. Các hệ thống blog và CMS (Content Management System)	17
1.7.4.5. Tạo và chia sẻ hình ảnh, video trên các mạng xã hội	17
CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG VÀ MÔ HÌNH HÓA YÊU CẦU.....	18
2.1. KHẢO SÁT HIỆN TRẠNG.....	18
2.1.1. TopCV	18
2.1.2. Vietnamworks.....	18
2.1.3. Kết luận	19
2.2. XÁC ĐỊNH YÊU CẦU.....	20
2.2.1. Yêu cầu chức năng.....	20
2.2.2. Yêu cầu phi chức năng.....	20
2.3. MÔ HÌNH HÓA YÊU CẦU	21
2.3.1. Lược đồ Use case.....	21

2.3.1.1. Use case Khách	21
2.3.1.2. Use case Ứng viên.....	21
2.3.1.3. Use case Nhà tuyển dụng.....	24
2.3.1.4. Use case Admin	27
2.3.2. Mô tả use case	29
CHƯƠNG 3: THIẾT KẾ PHẦN MỀM.....	44
3.1. MỘT SỐ MÔ HÌNH TUẦN TỰ TRONG ĐIỂM	44
3.1.1. Đăng nhập.....	44
3.1.2. Đăng ký vai trò nhà tuyển dụng	44
3.1.3. Lọc công việc	45
3.1.4. Upload file.....	46
3.1.5. Lưu hồ sơ.....	46
3.1.6. Tạo bài test	47
3.1.7. Lưu công việc	48
3.1.8. Ứng tuyển.....	49
3.2. LƯỢC ĐỒ LỚP.....	50
3.3. THIẾT KẾ GIAO DIỆN	52
CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ	63
4.1. CÁC CÔNG CỤ SỬ DỤNG	63
4.2. CÁC CÔNG NGHỆ SỬ DỤNG.....	63
4.3. CÀI ĐẶT ỨNG DỤNG	63
4.3.1. Yêu cầu môi trường	63
4.3.2. Các bước thực hiện	63
4.4. KIỂM THỬ	64
PHẦN 3: KẾT LUẬN.....	66
1. KẾT QUẢ ĐẠT ĐƯỢC:	66
2. ƯU ĐIỂM:.....	66

3. NHƯỢC ĐIỂM:	67
4. ĐỊNH HƯỚNG PHÁT TRIỂN:	67
DANH MỤC TÀI LIỆU THAM KHẢO.....	68

KẾ HOẠCH THỰC HIỆN

STT	Thời gian	Công việc	Ghi chú
1	22/8 – 5/9	Quyết định đề tài	
2	5/9 – 8/9	Khảo sát hiện trạng, thu thập thông tin, xác định và đặc tả yêu cầu	
3	8/9 – 10/9	Nghiên cứu và chuẩn bị môi trường phát triển (MERN Stack)	Cài đặt MongoDB, ReactJS, NodeJS
4	10/9 – 20/9	Thiết kế giao diện	
5	20/9 – 25/9	Thiết kế cơ sở dữ liệu và cấu trúc hệ thống	Xác định các bảng dữ liệu (users, jobs, company...)
6	25/9 – 1/10	Phát triển chức năng đăng ký và đăng nhập người dùng	Đăng ký tài khoản qua email, Google, JWT xác thực
7	1/10 – 8/10	Hiển thị thông tin công việc và công ty trên giao diện người dùng	Hiển thị danh sách các công việc tuyển dụng, thông tin chi tiết về từng công việc Hiển thị danh sách công ty, thông tin chi tiết từng công ty (các công việc đang tuyển dụng của công ty)
8	8/10 – 15/10	Xây dựng chức năng quản lý hồ sơ người dùng	CRUD cho hồ sơ cá nhân, kinh nghiệm, kỹ năng
9	15/10 – 29/10	Phát triển chức năng tìm kiếm và lọc thông tin	Tìm kiếm việc làm, công ty, lọc theo nhiều tiêu chí Tìm kiếm hồ sơ cá nhân của ứng viên, lọc theo các tiêu chí
10	29/10 – 5/11	Phát triển trang quản lý công việc cho ứng viên và nhà tuyển dụng	Lưu công việc đã ứng tuyển, theo dõi các công ty yêu thích
11	5/11 – 12/11	Phát triển trang quản lý công ty cho nhà tuyển dụng	CRUD thông tin công ty, đăng tin tuyển dụng
12	12/11 – 15/11	Xây dựng chức năng quản lý thông báo	Lưu thông báo, thông báo cho ứng viên, nhà tuyển dụng
13	15/11 – 17/11	Phát triển trang quản lý cuộc hẹn phỏng vấn	Thêm/Sửa/Xóa cuộc hẹn phỏng vấn
14	17/11 – 24/11	Xây dựng trang thống kê và báo cáo cho Admin	Báo cáo số lượng công việc, ứng viên, và các chỉ số khác
15	24/11 – 27/11	Phát triển tính năng lưu trữ file (Cloudinary)	Lưu trữ CV ứng viên, tài liệu công ty

16	27/11 – 4/12	Kiểm tra và sửa lỗi các tính năng của hệ thống	Kiểm thử toàn bộ hệ thống (UI/UX, API, bảo mật)
17	4/12 – 7/12	Xây dựng tài liệu hướng dẫn sử dụng và báo cáo kết quả	Soạn thảo tài liệu, viết báo cáo

DANH MỤC BẢNG

Bảng 1. Mô tả use case đăng ký ứng viên	29
Bảng 2. Mô tả use case đăng ký nhà tuyển dụng	30
Bảng 3. Mô tả use case đăng nhập tài khoản	30
Bảng 4. Mô tả use case xem hồ sơ cá nhân	31
Bảng 5. Mô tả use case upload file.....	31
Bảng 6. Mô tả use case xem chi tiết công việc	32
Bảng 7. Mô tả use case lọc và tìm kiếm thông tin	32
Bảng 8. Mô tả use case ứng tuyển công việc.....	33
Bảng 9. Mô tả use case chỉnh sửa hồ sơ	34
Bảng 10. Mô tả use case đổi mật khẩu	35
Bảng 11. Mô tả use case đăng tin tuyển dụng.....	36
Bảng 12. Mô tả use case chỉnh sửa tin tuyển dụng	37
Bảng 13. Mô tả use case xem danh sách ứng viên.....	38
Bảng 14. Mô tả use case xem hồ sơ ứng viên.....	39
Bảng 15. Mô tả use case gửi lời mời	40
Bảng 16. Mô tả use case tạo lịch phỏng vấn.....	41
Bảng 17. Mô tả use case quản lý bài test.....	42
Bảng 18. Mô tả use case vô hiệu hóa tài khoản người dùng	43
Bảng 19. Mô tả ý nghĩa các thành phần giao diện trang chủ.....	52
Bảng 20. Mô tả ý nghĩa các thành phần giao diện đăng nhập	53
Bảng 21. Mô tả ý nghĩa các thành phần giao diện đăng ký	53
Bảng 22. Mô tả ý nghĩa các thành phần giao diện quản lý hồ sơ cá nhân (Ứng viên)..	55
Bảng 23. Mô tả ý nghĩa các thành phần giao diện chi tiết công việc.....	56
Bảng 24. Mô tả ý nghĩa các thành phần giao diện quản lý công việc (Ứng viên).....	57
Bảng 25. Mô tả ý nghĩa các thành phần giao diện đăng ký nhà tuyển dụng	58
Bảng 26. Mô tả ý nghĩa các thành phần giao diện các thanh điều hướng (Nhà tuyển dụng)	59
Bảng 27. Mô tả ý nghĩa các thành phần giao diện quản lý công việc (Nhà tuyển dụng)	60
Bảng 28. Mô tả ý nghĩa các thành phần giao diện thống kê (Admin).....	61
Bảng 29. Các công cụ sử dụng xây dựng website.....	63

Bảng 30. Các công nghệ sử dụng xây dựng website	63
Bảng 31. Kết quả kiểm thử	64

DANH MỤC HÌNH ẢNH

Hình 1. Kiến trúc MERN Stack	4
Hình 2. Trang chủ TopCV	18
Hình 3. Trang chủ vietnamworks	19
Hình 4. Use case tổng quan của vai trò Khách	21
Hình 5. Use case quản lý công việc (Ứng viên).....	22
Hình 6. Use case quản lý hồ sơ cá nhân (Ứng viên)	22
Hình 7. Use case tổng quan của vai trò Ứng viên	23
Hình 8. Use case quản lý tệp đính kèm (Ứng viên)	24
Hình 9. Use case quản lý tài khoản (Ứng viên)	24
Hình 10. Use case quản lý công ty (Nhà tuyển dụng)	25
Hình 11. Use case quản lý bài test (Nhà tuyển dụng)	25
Hình 12. Use case quản lý tuyển dụng (Nhà tuyển dụng)	26
Hình 13. Use case tổng quan của vai trò Nhà tuyển dụng	27
Hình 14. Use case thống kê (Admin)	28
Hình 15. Use case quản lý tài khoản (Admin)	28
Hình 16. Use case tổng quan của vai trò Admin	29
Hình 17. Mô hình tuần tự chức năng đăng nhập	44
Hình 18. Mô hình tuần tự chức năng đăng ký vai trò nhà tuyển dụng	45
Hình 19. Mô hình tuần tự chức năng lọc công việc	45
Hình 20. Mô hình tuần tự chức năng upload file	46
Hình 21. Mô hình tuần tự chức năng lưu hồ sơ	46
Hình 22. Mô hình tuần tự chức năng tạo bài test	47
Hình 23. Mô hình tuần tự chức năng lưu công việc	48
Hình 24. Mô hình tuần tự chức năng ứng tuyển	49
Hình 25. Lược đồ cơ sở dữ liệu.....	51
Hình 26. Giao diện trang chủ	52
Hình 27. Giao diện đăng nhập.....	53
Hình 28. Giao diện đăng ký	53
Hình 29. Giao diện quản lý hồ sơ cá nhân (Ứng viên).....	54
Hình 30. Giao diện chi tiết công việc	56
Hình 31. Giao diện quản lý công việc (Ứng viên)	57

Hình 32. Giao diện đăng ký nhà tuyển dụng	58
Hình 33. Giao diện các thanh điều hướng (Nhà tuyển dụng)	59
Hình 34. Giao diện quản lý công việc (Nhà tuyển dụng)	60
Hình 35. Giao diện thống kê (Admin)	61

PHẦN 1: MỞ ĐẦU

1. Tính cấp thiết của đề tài

Hiện nay, với sự bùng nổ của công nghệ số và nhu cầu tìm kiếm nhân sự chất lượng, việc tối ưu hóa quy trình tuyển dụng không chỉ là mong muốn mà đã trở thành nhu cầu cấp bách của các doanh nghiệp. Tuy nhiên, thực tế cho thấy nhiều nền tảng hiện có vẫn chưa đáp ứng được các yêu cầu khắt khe trong việc sàng lọc ứng viên và đánh giá năng lực một cách khách quan. Điều này dẫn đến việc doanh nghiệp tốn nhiều thời gian và chi phí cho quy trình tuyển dụng nhưng không đảm bảo hiệu quả cao.

Đối với những người đi tìm việc làm, họ thường gặp khó khăn trong việc thể hiện đúng năng lực của mình trước nhà tuyển dụng, đặc biệt là ở những vị trí yêu cầu kỹ năng chuyên môn cao. Quy trình tuyển dụng truyền thống hoặc qua các nền tảng chưa tích hợp các công cụ đánh giá trước khi ứng tuyển thường làm giảm cơ hội tiếp cận của ứng viên tiềm năng. Ngoài ra, nhà tuyển dụng không có nhiều thông tin đáng tin cậy để đánh giá ứng viên trước vòng phỏng vấn, dẫn đến lãng phí nguồn lực trong khâu sàng lọc.

Dự án “*Xây dựng website tuyển dụng và tìm kiếm việc làm sử dụng Reactjs và Nodejs*” của chúng tôi ra đời với sứ mệnh giải quyết triệt để các vấn đề này. Bên cạnh việc cung cấp các tính năng cơ bản như tìm kiếm việc làm và kết nối giữa nhà tuyển dụng và ứng viên, nền tảng còn được tích hợp chức năng mời ứng viên tham gia các bài kiểm tra đánh giá trực tuyến trước khi ứng tuyển. Điều này không chỉ giúp nhà tuyển dụng sàng lọc được các ứng viên tiềm năng dựa trên kết quả bài kiểm tra, mà còn giúp ứng viên khẳng định năng lực của mình ngay từ những bước đầu tiên. Với các bài kiểm tra được thiết kế theo từng vị trí công việc, cả doanh nghiệp và ứng viên đều có thể tối ưu hóa quy trình đánh giá và ứng tuyển, từ đó tiết kiệm thời gian và nâng cao hiệu quả tuyển dụng.

Tính cấp thiết của dự án còn được thể hiện ở khả năng đáp ứng nhu cầu thực tế trong thị trường lao động ngày nay. Việc tích hợp tính năng đánh giá năng lực ứng viên giúp doanh nghiệp nhanh chóng tìm kiếm và lựa chọn đúng nhân tài, đồng thời tạo ra một quy trình tuyển dụng chuyên nghiệp, công bằng và minh bạch hơn. Với sự hỗ trợ của ReactJS và NodeJS, nền tảng sẽ đảm bảo tính mượt mà trong trải nghiệm người dùng và hiệu suất cao trong xử lý dữ liệu, từ đó đáp ứng được nhu cầu của cả nhà tuyển dụng lẫn ứng viên trong một thị trường lao động cạnh tranh và thay đổi không ngừng.

Dự án không chỉ là giải pháp công nghệ mà còn mang ý nghĩa chiến lược trong

việc phát triển bền vững thị trường lao động, tạo ra giá trị thực tế cho cả doanh nghiệp và người tìm việc, đóng góp tích cực vào sự chuyển đổi số trong lĩnh vực tuyển dụng và tìm kiếm việc làm.

2. Đối tượng nghiên cứu

Dự án “*Xây dựng website tuyển dụng và tìm kiếm việc làm sử dụng Reactjs và Nodejs*” tập trung nghiên cứu các yếu tố liên quan đến lĩnh vực công nghệ thông tin nhằm xây dựng một hệ thống hiệu quả, hiện đại và tối ưu. Các nội dung cụ thể được nghiên cứu bao gồm:

- Nghiên cứu Node.js và Express.js: Xây dựng hệ thống API RESTful hỗ trợ xử lý các chức năng như đăng nhập, đăng ký, quản lý thông tin công việc, ứng viên, và bài kiểm tra trực tuyến,... Mục tiêu là đảm bảo dữ liệu được truy xuất và xử lý một cách nhanh chóng và chính xác.

- Sử dụng cơ sở dữ liệu MongoDB: Tìm hiểu và áp dụng cơ sở dữ liệu phi quan hệ để lưu trữ và quản lý dữ liệu người dùng, thông tin tuyển dụng, bài kiểm tra, và các tương tác giữa ứng viên và nhà tuyển dụng. MongoDB đảm bảo khả năng mở rộng và linh hoạt trong việc lưu trữ dữ liệu đa dạng.

- Cloudinary: Tìm hiểu cách tích hợp và sử dụng Cloudinary để lưu trữ và quản lý các tệp đa phương tiện, bao gồm logo công ty và hình ảnh đại diện, giúp giảm tải lưu trữ cho hệ thống và tăng tốc độ tải trang.

- JSON Web Token (JWT): Nghiên cứu cơ chế xác thực và bảo mật dữ liệu, đảm bảo hệ thống an toàn cho cả người dùng (nhà tuyển dụng, ứng viên, admin) và dữ liệu nhạy cảm của hệ thống.

- Sử dụng React.js: Tìm hiểu và áp dụng thư viện React.js để thiết kế giao diện người dùng, tạo trải nghiệm tương tác mượt mà, dễ sử dụng, và thân thiện với người dùng cuối.

- Quy trình tuyển dụng trực tuyến: Phân tích các bước từ đăng tin tuyển dụng, mời ứng viên, đánh giá bài kiểm tra, đến chọn lọc ứng viên phù hợp để xây dựng hệ thống hỗ trợ toàn diện.

Việc nghiên cứu và áp dụng các công nghệ hiện đại sẽ giúp dự án đáp ứng được nhu cầu thực tế của người dùng, đồng thời đảm bảo hệ thống hoạt động hiệu quả, bảo mật, và thân thiện với người dùng.

3. Phạm vi nghiên cứu

Dự án “*Xây dựng website tuyển dụng và tìm kiếm việc làm sử dụng Reactjs và Nodejs*” sử dụng React.js để phát triển giao diện người dùng thân thiện và Node.js cùng Express.js để triển khai hệ thống RESTful API, đảm bảo hiệu suất và khả năng mở rộng.

Hệ thống kết hợp MongoDB để lưu trữ dữ liệu phi quan hệ, bao gồm thông tin ứng viên, tin tuyển dụng, và các bài kiểm tra. Bảo mật dữ liệu được đảm bảo thông qua JSON Web Token (JWT), giúp xác thực người dùng và bảo vệ thông tin cá nhân. Cloudinary được tích hợp để quản lý và lưu trữ hình ảnh, tài liệu liên quan đến hồ sơ ứng viên và công ty.

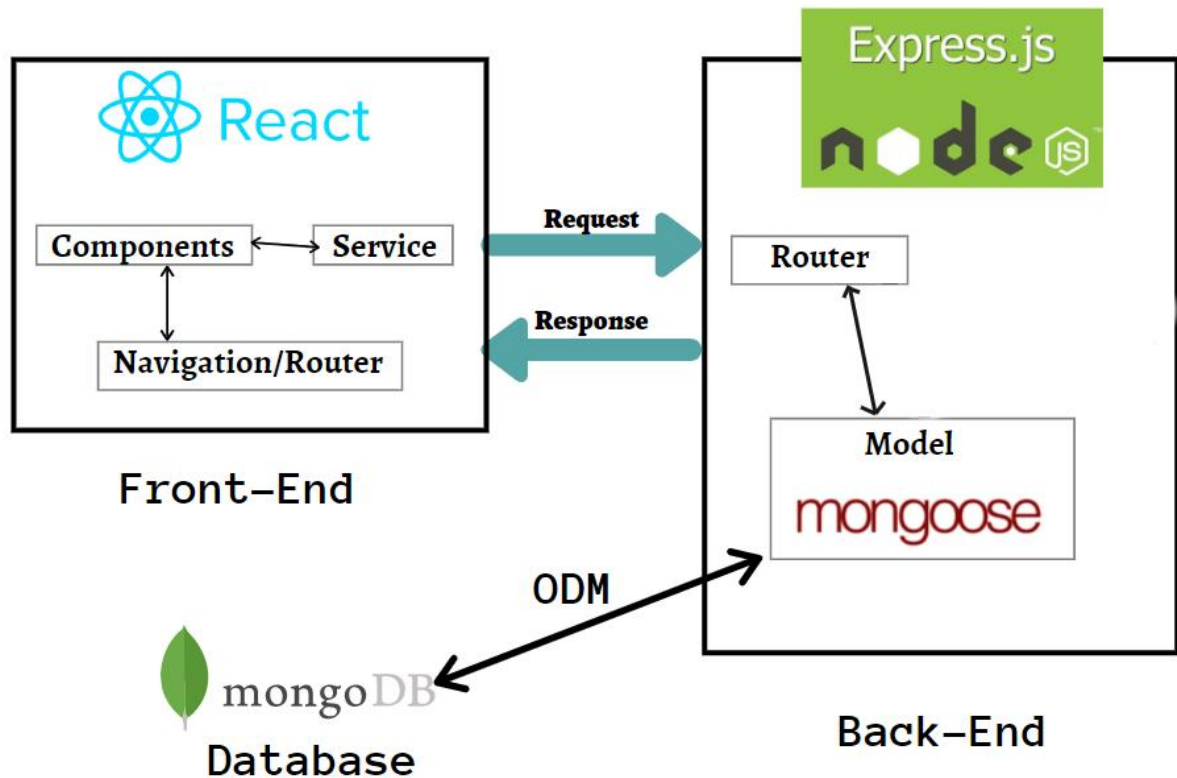
4. Kết quả dự kiến đạt được

- Xây dựng được một website cơ bản phục vụ nhu cầu tuyển dụng và tìm kiếm việc làm.
- Hoàn thành các chức năng đã được nêu ra ở “Đề cương chi tiết”.

PHẦN 2: NỘI DUNG

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1. KIẾN TRÚC HỆ THỐNG



Hình 1. Kiến trúc MERN Stack

MERN Stack là một bộ công nghệ bao gồm MongoDB, Express.js, React.js, và Node.js, được sử dụng để xây dựng các ứng dụng web full-stack. Đây là một giải pháp hoàn chỉnh, từ frontend đến backend, tất cả đều dựa trên JavaScript.

- MongoDB: Cơ sở dữ liệu NoSQL, lưu trữ dữ liệu dạng tài liệu.
- Express.js: Framework backend minimal chạy trên Node.js, giúp xây dựng API và xử lý logic server.
- React.js: Thư viện JavaScript phát triển giao diện người dùng tương tác và mượt mà.
- Node.js: Nền tảng chạy JavaScript phía server, giúp kết nối frontend và backend.

1.2. REACTJS

1.2.1. ReactJS là gì?

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook và ra mắt vào năm 2013. Mục đích chính của ReactJS là tạo ra giao diện người dùng

(UI) hiện đại, linh hoạt và hiệu quả, đặc biệt phù hợp cho các ứng dụng web có tính tương tác cao.

Trong mô hình MVC (Model-View-Controller), ReactJS chỉ tập trung vào View, tức là phần hiển thị giao diện người dùng. Điều này làm cho ReactJS trở nên gọn nhẹ, dễ tích hợp với các thư viện hoặc framework khác để đảm nhiệm các chức năng như routing, quản lý trạng thái, hoặc xử lý dữ liệu.

1.2.2. Kiến trúc của ReactJS

ReactJS có kiến trúc dựa trên các component (thành phần), cho phép phát triển giao diện web bằng cách chia nhỏ thành các khối độc lập. Những thành phần này được tổ chức thành một cấu trúc cây, bắt đầu từ component gốc và mở rộng xuống các component con.

Mỗi component hoạt động như một khối xây dựng giao diện, có thể chứa cả logic và dữ liệu, và chúng có thể tái sử dụng ở nhiều nơi trong ứng dụng.

1.2.3. Cách hoạt động của ReactJS

1.2.3.1. Component-Based Development (Phát triển dựa trên component):

ReactJS xây dựng giao diện bằng cách kết hợp nhiều component nhỏ lại với nhau. Điều này giúp dễ dàng quản lý và bảo trì các phần của ứng dụng.

1.2.3.2. Virtual DOM:

React sử dụng một bản sao ảo của DOM thật để tối ưu hóa việc cập nhật giao diện. Khi có thay đổi, React chỉ cập nhật các phần cần thiết thay vì làm mới toàn bộ trang, giúp cải thiện hiệu suất.

1.2.3.3. One-Way Data Flow (Luồng dữ liệu một chiều):

Dữ liệu trong React di chuyển theo một hướng từ component cha xuống component con, giúp dễ dàng kiểm soát và dự đoán trạng thái của ứng dụng.

1.2.3.4. React State and Props:

- State: Lưu trữ dữ liệu nội bộ trong từng component và thay đổi dựa trên tương tác của người dùng.

- Props: Truyền dữ liệu từ cha xuống con để hiển thị hoặc xử lý logic trong component con.

1.2.3.5. Tái kết xuất (Re-render):

Khi dữ liệu (state hoặc props) thay đổi, React tự động cập nhật giao diện tương ứng mà không cần tải lại toàn bộ trang.

1.2.4. Đặc điểm nổi bật của ReactJS

1.2.4.1. Component-Based Architecture (Kiến trúc dựa trên Component)

- Component là các khối xây dựng cơ bản của React.
- Mỗi component giống như một "mảnh" giao diện độc lập, có thể chứa cả logic (JavaScript) và giao diện (HTML/XML) riêng biệt. Điều này giúp bạn tái sử dụng component ở nhiều nơi khác nhau trong ứng dụng.

Ví dụ: Một ứng dụng có thể có các component như thanh điều hướng, danh sách sản phẩm, hoặc form đăng ký, và chúng có thể được sử dụng ở nhiều màn hình khác nhau.

1.2.4.2. Virtual DOM (Document Object Model Ảo)

- DOM (Document Object Model) là cấu trúc dữ liệu đại diện cho giao diện người dùng trong trình duyệt. Thao tác trực tiếp với DOM thật có thể làm giảm hiệu suất, đặc biệt với các ứng dụng lớn.

- React sử dụng Virtual DOM, là một bản sao nhẹ của DOM thật. Khi giao diện thay đổi, React sẽ so sánh DOM ảo trước và sau thay đổi (quá trình diffing) để xác định phần cần cập nhật, sau đó chỉ chỉnh sửa DOM thật tại các vị trí thay đổi, giúp cải thiện hiệu suất đáng kể.

1.2.4.3. One-Way Data Binding (Ràng buộc dữ liệu một chiều)

- Dữ liệu trong React di chuyển từ component cha xuống component con thông qua một cơ chế gọi là props.
- Điều này giúp kiểm soát dòng chảy dữ liệu dễ dàng hơn, giảm nguy cơ lỗi khi dữ liệu thay đổi. Dữ liệu một chiều cũng làm cho ứng dụng dễ dự đoán và debug.

1.2.4.4. JSX (JavaScript XML)

- JSX là một cú pháp đặc biệt cho phép bạn viết HTML trực tiếp trong JavaScript.
- Thay vì phải tách rời logic và giao diện, JSX giúp chúng được gắn kết tự nhiên hơn, khiến code dễ đọc và dễ bảo trì hơn. JSX không bắt buộc, nhưng nó là cách phổ biến nhất để làm việc với React.

1.2.4.5. Ecosystem & Community (Hệ sinh thái và cộng đồng)

- React có một hệ sinh thái phong phú với các thư viện bổ sung như:
 - + React Router để quản lý điều hướng giữa các trang.
 - + Redux hoặc Context API để quản lý trạng thái ứng dụng.
 - + Axios hoặc Fetch để gọi API và xử lý dữ liệu từ server.

- Ngoài ra, React có một cộng đồng lớn và hoạt động mạnh mẽ, cung cấp nhiều tài liệu, công cụ, và hỗ trợ nhanh chóng từ các nhà phát triển khác.

1.2.5. Ưu điểm của ReactJS

1.2.5.1. Hiệu suất cao

Nhờ Virtual DOM và cơ chế so sánh thông minh (reconciliation), React xử lý các thay đổi trên giao diện nhanh chóng, giúp cải thiện trải nghiệm người dùng, đặc biệt với các ứng dụng lớn hoặc có nhiều tương tác.

1.2.5.2. Tái sử dụng component

Các component trong React có thể tái sử dụng ở nhiều nơi, giảm thời gian phát triển và bảo trì ứng dụng. Điều này rất hữu ích trong các dự án lớn, nơi nhiều thành viên trong nhóm có thể làm việc trên các component độc lập.

1.2.5.3. Học tập dễ dàng

ReactJS chỉ tập trung vào UI, không phức tạp như một framework đầy đủ (như Angular). Lập trình viên chỉ cần nắm vững JavaScript cơ bản và JSX là có thể bắt đầu.

1.2.5.4. Hỗ trợ từ các công ty lớn

Được Facebook phát triển và bảo trì, React là một công cụ đáng tin cậy và thường xuyên được cập nhật để phù hợp với các xu hướng công nghệ mới.

1.2.6. Ứng dụng của ReactJS

1.2.6.1. Ứng dụng web một trang (Single Page Applications - SPA)

SPA là các ứng dụng mà nội dung được tải một lần, sau đó các thay đổi được cập nhật mà không cần tải lại trang.

React rất phù hợp với SPA nhờ cơ chế cập nhật nhanh và quản lý dữ liệu hiệu quả.

1.2.6.2. Ứng dụng di động với React Native

React Native, một framework dựa trên React, cho phép xây dựng các ứng dụng di động trên iOS và Android bằng cách sử dụng cùng một cơ sở code JavaScript.

1.2.6.3. Thương mại điện tử và nội dung động

Các trang web thương mại điện tử, quản lý nội dung hoặc các ứng dụng yêu cầu giao diện tương tác và cập nhật nhanh đều có thể sử dụng React để tối ưu trải nghiệm người dùng.

1.2.6.4. Dashboard và công cụ phân tích dữ liệu

React hỗ trợ tốt cho các dashboard hiển thị dữ liệu thời gian thực, nơi mà các

thành phần như biểu đồ, bảng dữ liệu cần được cập nhật liên tục.

1.3. NODEJS

1.3.1. Node.js là gì?

Node.js là một môi trường chạy JavaScript phía máy chủ (server-side), được phát triển bởi Ryan Dahl vào năm 2009. Node.js sử dụng V8 JavaScript Engine (được phát triển bởi Google), giúp JavaScript chạy nhanh và hiệu quả bên ngoài trình duyệt.

Node.js cho phép các lập trình viên sử dụng JavaScript để xây dựng các ứng dụng backend, từ máy chủ web, API, đến các hệ thống xử lý dữ liệu phức tạp.

1.3.2. Đặc điểm nổi bật của Node.js

1.3.2.1. Single-threaded, Event-driven (Luồng đơn, điều khiển sự kiện):

- Node.js hoạt động trên một luồng đơn nhưng có khả năng xử lý nhiều yêu cầu đồng thời nhờ cơ chế non-blocking I/O.

- Thay vì chờ hoàn tất một yêu cầu, Node.js xử lý các yêu cầu khác, tăng hiệu suất ứng dụng.

1.3.2.2. Non-blocking I/O (Xử lý I/O không chặn):

Nhờ cơ chế này, Node.js rất phù hợp cho các ứng dụng có nhiều yêu cầu I/O, chẳng hạn như xử lý tệp tin, cơ sở dữ liệu, hoặc gọi API.

1.3.2.3. NPM (Node Package Manager):

Node.js đi kèm với NPM, một hệ thống quản lý thư viện lớn nhất dành cho JavaScript, cho phép cài đặt và sử dụng hàng triệu thư viện dễ dàng.

1.3.2.4. Cross-platform (Đa nền tảng):

Node.js có thể chạy trên nhiều hệ điều hành như Windows, macOS, và Linux, giúp dễ dàng triển khai ứng dụng.

1.3.3. Ưu điểm của Node.js

1.3.3.1. Hiệu suất cao:

Sử dụng V8 Engine của Google, Node.js mang lại hiệu suất nhanh và mạnh mẽ khi xử lý JavaScript.

1.3.3.2. Xử lý đồng thời:

Nhờ mô hình xử lý sự kiện và non-blocking I/O, Node.js xử lý hàng nghìn yêu cầu một cách hiệu quả.

1.3.3.3. Cộng đồng lớn:

Node.js có một cộng đồng phát triển sôi động với nhiều thư viện, công cụ, và hỗ

trợ tốt từ các nhà phát triển trên toàn thế giới.

1.3.3.4. Dễ học:

Với lập trình viên JavaScript, học và sử dụng Node.js rất dễ dàng vì sử dụng cùng ngôn ngữ cho cả frontend và backend.

1.3.4. Ứng dụng của Node.js

Node.js được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

- API Backend: Phát triển API RESTful cho các ứng dụng web và di động.
- Ứng dụng thời gian thực: Chat, ứng dụng cộng tác (như Google Docs).
- Xử lý tệp tin: Hệ thống quản lý tệp hoặc xử lý dữ liệu lớn.
- Thương mại điện tử: Các nền tảng bán hàng trực tuyến.
- Streaming: Xây dựng các hệ thống phát trực tiếp (livestream) hoặc truyền dữ liệu theo thời gian thực.

1.4. EXPRESS

1.4.1. Express.js là gì?

Express.js là một framework web tối giản và mạnh mẽ được xây dựng trên nền tảng Node.js, giúp dễ dàng phát triển các ứng dụng web và API. Được ra mắt vào năm 2010, Express.js thường được gọi là "framework backend tiêu chuẩn" cho Node.js nhờ tính đơn giản, linh hoạt, và khả năng mở rộng.

1.4.2. Cách hoạt động của Express.js

Express.js hoạt động theo cơ chế nhận yêu cầu HTTP và trả về phản hồi thông qua các bước sau:

- Xử lý yêu cầu: Khi một yêu cầu HTTP được gửi tới máy chủ, Express kiểm tra tuyến đường và phương thức HTTP tương ứng (GET, POST, v.v.).
- Middleware: Middleware trong Express thực thi lần lượt để xử lý yêu cầu, như xác thực người dùng, phân tích dữ liệu yêu cầu, hoặc ghi log.
- Trả về phản hồi: Sau khi yêu cầu được xử lý, Express gửi lại phản hồi (response) cho người dùng, có thể là dữ liệu JSON, HTML, hoặc thông báo lỗi.

1.4.3. Đặc điểm nổi bật của Express.js

1.4.3.1. Tối giản nhưng mạnh mẽ:

Express cung cấp các công cụ cơ bản để xử lý các yêu cầu HTTP, routing (định tuyến), và middleware, giúp lập trình viên xây dựng ứng dụng nhanh chóng mà không bị ràng buộc bởi các quy tắc phức tạp.

1.4.3.2. Hỗ trợ Routing linh hoạt:

Express cho phép định nghĩa các tuyến đường (routes) để xử lý các yêu cầu với các phương thức HTTP như GET, POST, PUT, DELETE,... phù hợp cho phát triển API.

1.4.3.3. Middleware:

Express có hệ thống middleware mạnh mẽ, cho phép xử lý các yêu cầu hoặc phản hồi trước khi chúng đến đích cuối cùng. Middleware rất hữu ích để xác thực, xử lý lỗi, hoặc ghi log.

1.4.3.4. Tích hợp dễ dàng:

Express tích hợp tốt với các cơ sở dữ liệu như MongoDB, MySQL, và các thư viện khác để hỗ trợ xây dựng ứng dụng backend toàn diện.

1.4.3.5. Khả năng mở rộng:

Nhờ thiết kế module hóa, Express có thể mở rộng bằng cách sử dụng các thư viện hoặc middleware từ cộng đồng hoặc tự xây dựng.

1.4.4. Ưu điểm của Express.js

1.4.4.1. Nhanh và nhẹ:

Với thiết kế tối giản, Express không yêu cầu cấu hình phức tạp, giúp khởi tạo ứng dụng nhanh chóng và dễ dàng.

1.4.4.2. Hỗ trợ mạnh mẽ từ cộng đồng:

Express có một cộng đồng lớn và hệ sinh thái phong phú, với hàng nghìn plugin và tài liệu hỗ trợ từ các lập trình viên trên toàn thế giới.

1.4.4.3. Tích hợp tốt với Node.js:

Express tận dụng các tính năng của Node.js như non-blocking I/O và event-driven architecture để đảm bảo hiệu suất cao.

1.4.4.4. Đa dụng:

Express có thể được sử dụng để xây dựng các loại ứng dụng khác nhau, từ website, API RESTful, đến các ứng dụng thời gian thực.

1.4.5. Ứng dụng của Express.js

1.4.5.1. Phát triển API RESTful:

Express là lựa chọn hàng đầu để xây dựng API nhanh, nhẹ, và dễ bảo trì.

1.4.5.2. Ứng dụng web:

Dùng để tạo các trang web động hoặc tích hợp với các công cụ như React, Angular để phát triển full-stack.

1.4.5.3. Ứng dụng thời gian thực:

Kết hợp với WebSocket hoặc thư viện như Socket.io để xây dựng các ứng dụng chat, theo dõi thời gian thực.

1.4.5.4. Hệ thống quản lý nội dung:

Express hỗ trợ tốt cho việc phát triển các hệ thống quản lý nội dung (CMS) hoặc các nền tảng thương mại điện tử.

1.5. MONGODB

1.5.1. MongoDB là gì?

MongoDB là một cơ sở dữ liệu NoSQL mã nguồn mở, được thiết kế để lưu trữ và quản lý dữ liệu không theo cấu trúc bảng (như trong các hệ quản trị cơ sở dữ liệu quan hệ). Thay vì sử dụng bảng và dòng như trong SQL, MongoDB lưu trữ dữ liệu dưới dạng các dokument (tài liệu) JSON hoặc BSON (Binary JSON), dễ dàng mở rộng và linh hoạt.

1.5.2. Cách hoạt động của MongoDB

1.5.2.1. Tài liệu (Documents):

Dữ liệu trong MongoDB được lưu trữ dưới dạng tài liệu BSON, tương tự như các đối tượng JSON. Mỗi tài liệu có thể chứa các trường với các giá trị có thể là số, chuỗi, mảng, hoặc các tài liệu lồng nhau.

1.5.2.2. Bộ sưu tập (Collections):

Các tài liệu trong MongoDB được tổ chức thành collections (tương đương với các bảng trong cơ sở dữ liệu quan hệ). Mỗi collection có thể chứa nhiều tài liệu với cấu trúc khác nhau.

1.5.2.3. Truy vấn và Tìm kiếm:

MongoDB cung cấp các truy vấn mạnh mẽ cho việc tìm kiếm, chèn, cập nhật và xóa tài liệu. Các chỉ mục giúp tối ưu hóa việc truy vấn dữ liệu.

1.5.2.4. Replication và Sharding:

- Replication giúp sao chép dữ liệu từ một server chính (primary) sang các server phụ (secondary) để đảm bảo tính sẵn sàng cao.

- Sharding giúp phân mảnh dữ liệu và phân phối chúng trên các máy chủ khác nhau để tăng cường khả năng mở rộng.

1.5.3. Đặc điểm nổi bật của MongoDB

1.5.3.1. Dữ liệu không có cấu trúc cố định (Schema-less):

Dữ liệu trong MongoDB có thể có các trường khác nhau, giúp lưu trữ dữ liệu linh hoạt mà không cần phải tuân theo một sơ đồ (schema) cố định, làm cho việc thay đổi cấu trúc dữ liệu trở nên dễ dàng hơn.

1.5.3.2. Lưu trữ dữ liệu dưới dạng JSON/BSON:

Mỗi tài liệu trong MongoDB là một đối tượng JSON hoặc BSON, giúp dễ dàng đọc và ghi dữ liệu. Điều này cũng giúp tương thích tốt với các ngôn ngữ lập trình như JavaScript.

1.5.3.3. Mở rộng dễ dàng (Scalability):

MongoDB hỗ trợ sharding (phân mảnh dữ liệu), giúp phân phối dữ liệu trên nhiều máy chủ và xử lý khối lượng lớn dữ liệu mà không ảnh hưởng đến hiệu suất.

1.5.3.4. Tính nhất quán và tính sẵn sàng cao (High Availability):

MongoDB hỗ trợ replication (sao chép dữ liệu) giữa các node trong cluster, giúp đảm bảo tính sẵn sàng cao và bảo vệ dữ liệu khỏi mất mát.

1.5.3.5. Tìm kiếm mạnh mẽ:

MongoDB cung cấp khả năng tìm kiếm mạnh mẽ với các truy vấn linh hoạt và các chỉ mục hỗ trợ việc truy xuất dữ liệu nhanh chóng.

1.5.4. Ưu điểm của MongoDB

1.5.4.1. Linh hoạt trong việc lưu trữ dữ liệu:

MongoDB không yêu cầu cấu trúc bảng như các hệ quản trị cơ sở dữ liệu quan hệ, giúp lưu trữ và xử lý dữ liệu không đồng nhất dễ dàng hơn.

1.5.4.2. Khả năng mở rộng dễ dàng:

MongoDB hỗ trợ mở rộng theo chiều ngang (horizontal scaling), nghĩa là bạn có thể dễ dàng thêm các máy chủ vào cluster để tăng dung lượng lưu trữ và khả năng xử lý mà không gặp khó khăn.

1.5.4.3. Cập nhật nhanh và dễ dàng:

MongoDB cho phép thay đổi cấu trúc dữ liệu mà không cần phải thực hiện thay đổi trên cơ sở dữ liệu toàn bộ, giúp dễ dàng thay đổi hoặc nâng cấp ứng dụng.

1.5.4.4. Cộng đồng mạnh mẽ và hệ sinh thái lớn:

MongoDB có một cộng đồng phát triển mạnh mẽ với nhiều công cụ và thư viện hỗ trợ, giúp việc triển khai, bảo trì, và mở rộng trở nên dễ dàng.

1.5.5. Ứng dụng của MongoDB

1.5.5.1. Ứng dụng Web và Di động:

MongoDB rất phù hợp cho các ứng dụng web hoặc di động có yêu cầu lưu trữ dữ liệu linh hoạt, chẳng hạn như các ứng dụng mạng xã hội, thương mại điện tử, hoặc hệ thống quản lý nội dung.

1.5.5.2. Big Data và Phân tích Dữ liệu:

Với khả năng mở rộng và xử lý dữ liệu lớn, MongoDB là lựa chọn lý tưởng cho các ứng dụng yêu cầu phân tích và xử lý dữ liệu lớn.

1.5.5.3. Dữ liệu không cấu trúc:

MongoDB rất phù hợp cho các ứng dụng lưu trữ dữ liệu không có cấu trúc cố định, chẳng hạn như các hệ thống lưu trữ dữ liệu JSON hoặc log.

1.5.5.4. Internet of Things (IoT):

MongoDB có thể được sử dụng để lưu trữ và quản lý dữ liệu từ các thiết bị IoT, vì các dữ liệu thu thập từ các thiết bị này có thể có cấu trúc không đồng nhất.

1.6. JSON WEB TOKEN

1.6.1. JSON Web Token (JWT) là gì?

JSON Web Token (JWT) là một tiêu chuẩn mở (RFC 7519) dùng để truyền tải thông tin an toàn giữa các hệ thống dưới dạng một chuỗi JSON. JWT được sử dụng phổ biến trong xác thực và phân quyền trong các ứng dụng web và API.

JWT thường được dùng trong các hệ thống bảo mật để xác thực người dùng và phân quyền truy cập vào các tài nguyên. Nó bao gồm một chuỗi mã hóa (encoded string) được chia thành ba phần: header, payload, và signature.

1.6.2. Cấu trúc của JWT

JWT được chia thành ba phần chính:

- Header: Phần này chứa thông tin về loại token và thuật toán mã hóa sử dụng. Thường thì sẽ có hai trường là *typ* (loại token) và *alg* (thuật toán mã hóa). Ví dụ, thuật toán phổ biến là HMAC SHA256 hoặc RSA.

- Payload: Payload chứa các tuyên bố (claims), tức là thông tin về đối tượng (chẳng hạn như người dùng) và các quyền truy cập hoặc dữ liệu khác. Có thể là:

- + Registered Claims: Những thông tin tiêu chuẩn như **sub** (subject), **iat** (issued at), **exp** (expiration).

- + Public Claims: Những thông tin do người sử dụng định nghĩa.

- + Private Claims: Những thông tin được chia sẻ giữa các hệ thống hoặc dịch vụ riêng biệt.

- Signature: Signature được tạo ra bằng cách mã hóa phần header và payload với một khóa bí mật (hoặc khóa công khai, nếu dùng thuật toán như RSA) và thuật toán mã hóa đã chọn. Điều này đảm bảo rằng token không bị thay đổi trong quá trình truyền tải.

1.6.3. Cách hoạt động của JWT

- Client đăng nhập: Người dùng gửi thông tin đăng nhập (như tên người dùng và mật khẩu) đến server.

- Server tạo JWT: Sau khi xác thực thông tin, server tạo ra một JWT chứa thông tin người dùng và các quyền truy cập, rồi gửi lại JWT này cho client.

- Client sử dụng JWT: Client lưu JWT trong bộ nhớ (localStorage hoặc cookie) và gửi JWT này trong các yêu cầu HTTP sau đó dưới dạng Authorization header.

- Server kiểm tra JWT: Server nhận JWT, giải mã nó và kiểm tra tính hợp lệ (chẳng hạn như kiểm tra chữ ký và hạn sử dụng). Nếu hợp lệ, server sẽ xử lý yêu cầu; nếu không, server sẽ trả về lỗi xác thực.

1.6.4. Ưu điểm của JWT

1.6.4.1. Tính bảo mật cao:

JWT sử dụng các phương thức mã hóa mạnh mẽ như HMAC SHA256 hoặc RSA để bảo vệ dữ liệu khỏi việc thay đổi trong quá trình truyền tải.

1.6.4.2. Độc lập với cơ sở dữ liệu:

Một khi JWT được cấp phát, nó có thể được sử dụng mà không cần truy cập cơ sở dữ liệu mỗi lần xác thực. Điều này giúp tăng hiệu suất và giảm tải cho server.

1.6.4.3. Hỗ trợ phân quyền:

JWT chứa thông tin về các quyền của người dùng (claims), giúp phân quyền và kiểm soát truy cập tài nguyên dễ dàng.

1.6.4.4. Phát hành và sử dụng dễ dàng:

JWT là một chuỗi văn bản dễ dàng để truyền qua các giao thức như HTTP hoặc WebSocket.

1.6.5. Ứng dụng của JWT

1.6.5.1. Xác thực người dùng (Authentication):

Đây là ứng dụng phổ biến nhất của JWT. Sau khi người dùng đăng nhập thành công, server sẽ phát hành một JWT chứa thông tin xác thực. Người dùng sẽ gửi JWT này cùng với các yêu cầu tiếp theo để chứng minh họ là ai.

1.6.5.2. Phân quyền truy cập (Authorization):

JWT có thể lưu trữ thông tin về quyền truy cập của người dùng (chẳng hạn như các role hoặc quyền), giúp kiểm soát những gì người dùng có thể và không thể làm trong ứng dụng.

1.6.5.3. Tích hợp giữa các dịch vụ (Service-to-Service Authentication):

Trong các ứng dụng microservices, JWT có thể được sử dụng để xác thực và phân quyền giữa các dịch vụ, giúp giao tiếp an toàn giữa chúng mà không cần phải chia sẻ thông tin đăng nhập.

1.6.5.4. Single Sign-On (SSO):

JWT có thể được sử dụng trong các hệ thống SSO để cho phép người dùng đăng nhập một lần và truy cập vào nhiều ứng dụng mà không cần phải đăng nhập lại.

1.7. CLOUDINARY

1.7.1. Cloudinary là gì?

Cloudinary là một dịch vụ quản lý và tối ưu hóa media (hình ảnh, video) trên đám mây, giúp các nhà phát triển dễ dàng lưu trữ, xử lý và phân phối nội dung đa phương tiện (media content) một cách hiệu quả và nhanh chóng. Cloudinary cung cấp các công cụ mạnh mẽ để lưu trữ, chỉnh sửa, tối ưu hóa, và phân phối các tài nguyên này thông qua các API và các tính năng tích hợp sẵn.

1.7.2. Đặc điểm nổi bật của Cloudinary

1.7.2.1. Quản lý hình ảnh và video

Cloudinary cho phép lưu trữ và quản lý các tệp media như hình ảnh và video. Bạn có thể tải lên, lưu trữ, và truy xuất chúng dễ dàng thông qua các API hoặc giao diện web.

1.7.2.2. Xử lý và chỉnh sửa media

Cloudinary hỗ trợ các tính năng chỉnh sửa hình ảnh và video mạnh mẽ như cắt xén, xoay, thay đổi kích thước, thêm bộ lọc, chuyển đổi định dạng, thay đổi độ phân giải, và nhiều tính năng khác mà không cần phải tải xuống tệp để chỉnh sửa thủ công.

1.7.2.3. Tối ưu hóa hiệu suất

Cloudinary tự động tối ưu hóa các tệp media (hình ảnh, video) để đảm bảo chúng tải nhanh hơn và tiết kiệm băng thông. Các tệp media sẽ được tự động thay đổi kích thước và định dạng để phù hợp với thiết bị và trình duyệt của người dùng cuối.

1.7.2.4. CDN (Content Delivery Network)

Cloudinary tích hợp với các mạng phân phối nội dung (CDN) để đảm bảo media

được phân phối nhanh chóng từ các server gần người dùng nhất. Điều này giúp giảm độ trễ và tối ưu tốc độ tải trang web hoặc ứng dụng.

1.7.2.5. Tích hợp API mạnh mẽ

Cloudinary cung cấp các API dễ sử dụng cho việc tải lên, quản lý và xử lý media. Các API này có thể dễ dàng tích hợp vào các ứng dụng web và di động, cho phép nhà phát triển linh hoạt xử lý các tệp media.

1.7.2.6. Bảo mật

Cloudinary cung cấp các tính năng bảo mật như giới hạn quyền truy cập vào các tài nguyên, mã hóa URL và kiểm soát quyền sử dụng media để bảo vệ nội dung của bạn khỏi truy cập trái phép.

1.7.2.7. Hỗ trợ video

Ngoài hình ảnh, Cloudinary còn hỗ trợ video với các tính năng như cắt video, thay đổi độ phân giải, thay đổi định dạng video và xử lý video (chuyển đổi định dạng, nén).

1.7.3. Ưu điểm của Cloudinary

1.7.3.1. Tiết kiệm thời gian và công sức:

Cloudinary giảm bớt nhu cầu lưu trữ và xử lý thủ công media. Bạn có thể tự động hóa quá trình xử lý và tối ưu hóa media mà không cần phải tự xây dựng giải pháp riêng.

1.7.3.2. Hiệu suất cao:

Với CDN tích hợp, media của bạn được phân phối toàn cầu, giúp tăng tốc độ tải trang và cải thiện trải nghiệm người dùng.

1.7.3.3. Quản lý tập trung:

Cloudinary cung cấp một nền tảng tập trung để quản lý tất cả các tệp media của bạn, từ tải lên cho đến phân phối và tối ưu hóa.

1.7.3.4. Tính linh hoạt cao:

Các tính năng của Cloudinary có thể được tích hợp vào nhiều loại ứng dụng và nền tảng, từ web cho đến di động, với khả năng xử lý mạnh mẽ và tùy chỉnh.

1.7.3.5. Bảo mật và kiểm soát quyền truy cập:

Cloudinary cho phép bạn kiểm soát ai có thể truy cập và thao tác với media của bạn thông qua các cơ chế bảo mật như mã hóa URL và quyền truy cập tài nguyên.

1.7.4. Ứng dụng của Cloudinary

1.7.4.1. Xử lý và quản lý hình ảnh cho ứng dụng web và di động

Cloudinary rất phù hợp cho các ứng dụng web và di động có nhu cầu xử lý và tối ưu hóa hình ảnh (chẳng hạn như thay đổi kích thước hình ảnh tùy thuộc vào thiết bị của người dùng).

1.7.4.2. Tối ưu hóa video

Với các công cụ chỉnh sửa video mạnh mẽ, Cloudinary giúp các ứng dụng yêu cầu phát video tối ưu hiệu suất và chất lượng mà không làm giảm trải nghiệm người dùng.

1.7.4.3. Quản lý các tài nguyên đa phương tiện trong các nền tảng thương mại điện tử

Các trang web thương mại điện tử có thể sử dụng Cloudinary để quản lý và tối ưu hóa hình ảnh và video sản phẩm, giúp giảm thời gian tải trang và cải thiện trải nghiệm mua sắm.

1.7.4.4. Các hệ thống blog và CMS (Content Management System)

Cloudinary giúp các hệ thống quản lý nội dung (CMS) dễ dàng xử lý hình ảnh và video, tối ưu hóa chúng để hiển thị tốt trên các thiết bị khác nhau.

1.7.4.5. Tạo và chia sẻ hình ảnh, video trên các mạng xã hội

Cloudinary hỗ trợ xử lý và chia sẻ hình ảnh, video trên các mạng xã hội với các tính năng chỉnh sửa trực tiếp, giúp các ứng dụng chia sẻ media dễ dàng hơn.

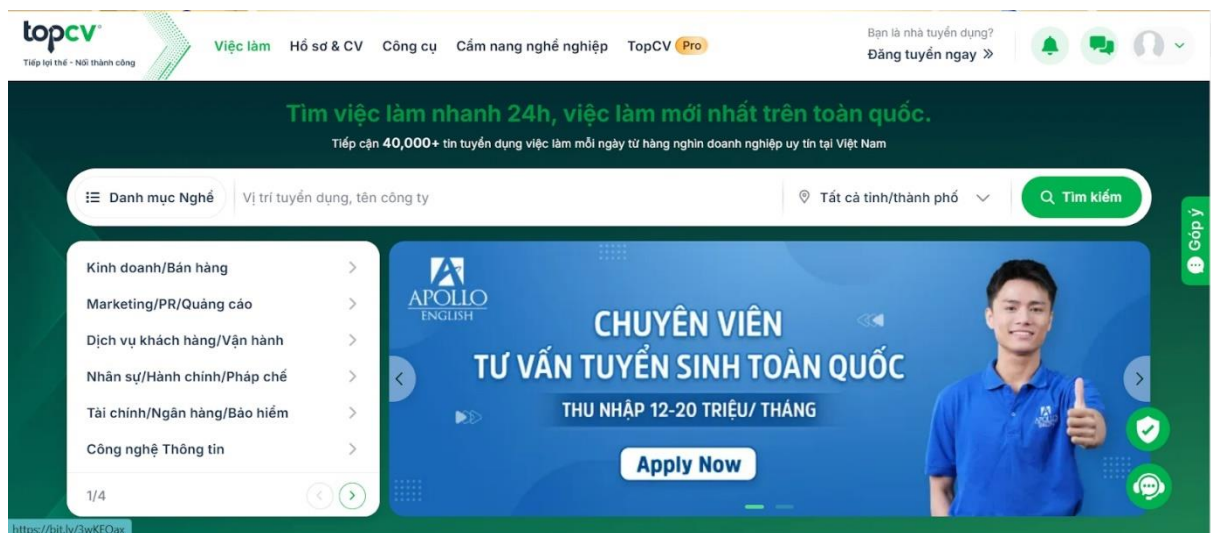
CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG VÀ MÔ HÌNH HÓA YÊU CẦU

2.1. KHẢO SÁT HIỆN TRẠNG

Để tìm hiểu sâu hơn và xác định được phạm vi của đề tài, nhóm đã tiến hành khảo sát và tìm hiểu 2 website tuyển dụng và tìm kiếm việc làm đang thịnh hành trên thị trường.

2.1.1. TopCV

Với sự phát triển nhanh chóng của công nghệ và nhu cầu kết nối nguồn nhân lực ngày càng cao, dự án của chúng tôi tập trung vào việc xây dựng một website tuyển dụng và tìm kiếm việc làm. Lấy cảm hứng từ những nền tảng thành công như TopCV, chúng tôi hướng tới việc phát triển một hệ thống thân thiện, hiệu quả, và tối ưu cho cả ứng viên và nhà tuyển dụng.



Hình 2. Trang chủ TopCV

Các chức năng chính của TopCV bao gồm:

- Hỗ trợ ứng viên tiếp cận dễ dàng hơn với các cơ hội việc làm phù hợp.
- Giúp nhà tuyển dụng tìm kiếm, quản lý, và sàng lọc ứng viên nhanh chóng, chính xác.
- Tích hợp công nghệ tiên tiến như AI và phân tích dữ liệu, đảm bảo các giải pháp tối ưu và tiết kiệm thời gian.

2.1.2. Vietnamworks

VietnamWorks được thành lập vào năm 2002 bởi Navigos Group, là một trong những nền tảng tuyển dụng hàng đầu tại Việt Nam. Với hơn 20 năm hoạt động, trang web này đã trở thành cầu nối đáng tin cậy giữa nhà tuyển dụng và người tìm việc, mang đến các giải pháp tuyển dụng hiệu quả và chuyên nghiệp.



Hình 3. Trang chủ vietnamworks

VietnamWorks cung cấp nhiều tính năng cơ bản như đăng tin tuyển dụng, tìm kiếm và quản lý ứng viên hiệu quả. Ngoài ra, nền tảng này còn hỗ trợ đào tạo kỹ năng, tư vấn tuyển dụng, và nghiên cứu thị trường lao động, giúp doanh nghiệp tối ưu hóa chiến lược tuyển dụng và ứng viên nâng cao cơ hội nghề nghiệp.

Các tính năng nổi bật nhất của VietnamWorks bao gồm bộ lọc tìm kiếm chuyên nghiệp, đăng tải tin tuyển dụng dễ dàng, trang cá nhân cho ứng viên, và hỗ trợ tư vấn nghề nghiệp. Trang web còn có chức năng tìm kiếm ứng viên thông qua các tiêu chí khác nhau và các dịch vụ hỗ trợ tuyển dụng như sàng lọc và phỏng vấn trực tuyến.

2.1.3. Kết luận

Cả TopCV và VietnamWorks đều cung cấp các tính năng cơ bản như đăng tin tuyển dụng, tìm kiếm công việc, và quản lý hồ sơ ứng viên. Tuy nhiên, VietnamWorks cung cấp các dịch vụ chuyên sâu như tư vấn tuyển dụng và nghiên cứu thị trường, trong khi TopCV tập trung mạnh vào việc tạo CV và hồ sơ ứng viên. Điều này cho thấy, để phát triển website tuyển dụng của bạn, bạn cần xác định rõ tính năng chính mà bạn muốn tập trung, ví dụ như hỗ trợ tạo CV cho ứng viên hay phát triển các dịch vụ tuyển dụng chuyên sâu cho doanh nghiệp.

TopCV đặc biệt chú trọng đến việc hỗ trợ ứng viên xây dựng hồ sơ một cách dễ dàng và nhanh chóng, trong khi VietnamWorks cung cấp nhiều công cụ cho cả nhà tuyển dụng và ứng viên, bao gồm các tính năng phân tích thị trường lao động và tư vấn nhân sự.

Định hướng phát triển và xác định phạm vi đề tài của nhóm:

- Tập trung vào việc phát triển một nền tảng tuyển dụng và tìm kiếm việc làm

đơn giản nhưng hiệu quả, với các tính năng cơ bản như đăng tuyển, tìm kiếm công việc, quản lý hồ sơ ứng viên và nhà tuyển dụng.

- Xây dựng các công cụ hỗ trợ tạo CV và hồ sơ ứng viên, giúp nâng cao trải nghiệm người dùng.

- Mở rộng dần các tính năng hỗ trợ doanh nghiệp tuyển dụng hiệu quả hơn, chẳng hạn như phân tích hồ sơ ứng viên và gợi ý công việc tự động.

- Cân nhắc việc phát triển các dịch vụ gia tăng (như tư vấn tuyển dụng, đào tạo kỹ năng, nghiên cứu thị trường) khi nền tảng đã ổn định, nhằm mang lại giá trị lâu dài cho cả người dùng và doanh nghiệp.

2.2. XÁC ĐỊNH YÊU CẦU

2.2.1. Yêu cầu chức năng

Chức năng đăng nhập

- Người dùng có thể đăng ký tài khoản thông qua email/username và password
- Người dùng có thể đăng ký/đăng nhập bằng Google.

Lưu trữ thông tin

- Lưu thông tin tài khoản của người dùng.
- Lưu thông tin hồ sơ cá nhân của các ứng viên
- Lưu thông tin kinh nghiệm, học vấn, file CV của ứng viên
- Lưu thông tin công việc ứng viên đã ứng tuyển, đã xem, đã lưu
- Lưu thông tin công ty mà ứng viên theo dõi.
- Lưu thông tin thông báo
- Lưu hồ sơ công ty
- Lưu hồ sơ công việc
- Lưu thông tin các bài test, các câu hỏi của bài test.
- Lưu thông tin cuộc hẹn phỏng vấn

Tìm kiếm/lọc thông tin

- Tìm kiếm thông tin công việc, công ty
- Lọc thông tin tìm kiếm công việc mong muốn
- Lọc hồ sơ cá nhân
- Lọc thông tin công việc dựa trên hồ sơ cá nhân

2.2.2. Yêu cầu phi chức năng

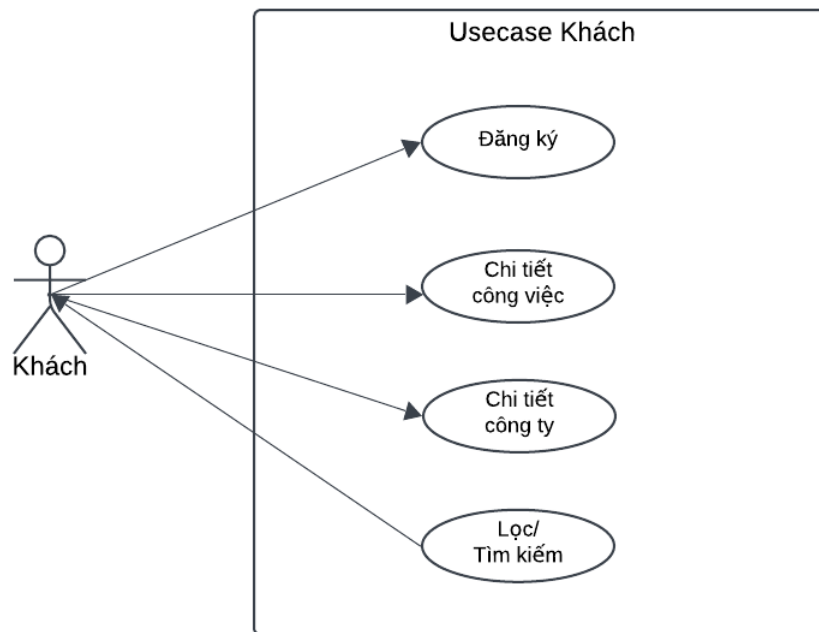
- Giao diện trực quan, thân thiện, dễ sử dụng

- Bảo mật thông tin, mã hóa dữ liệu người dùng, bảo vệ thông tin cá nhân.
- Mã nguồn phải rõ ràng, dễ bảo trì và mở rộng.
- Hệ thống có khả năng mở rộng để xử lý số lượng người dùng lớn.
- Hệ thống phải tương thích với các trình duyệt phổ biến và hỗ trợ API tích hợp.
- Tốc độ xử lý và truyền tải dữ liệu phải nhanh chóng.

2.3. MÔ HÌNH HÓA YÊU CẦU

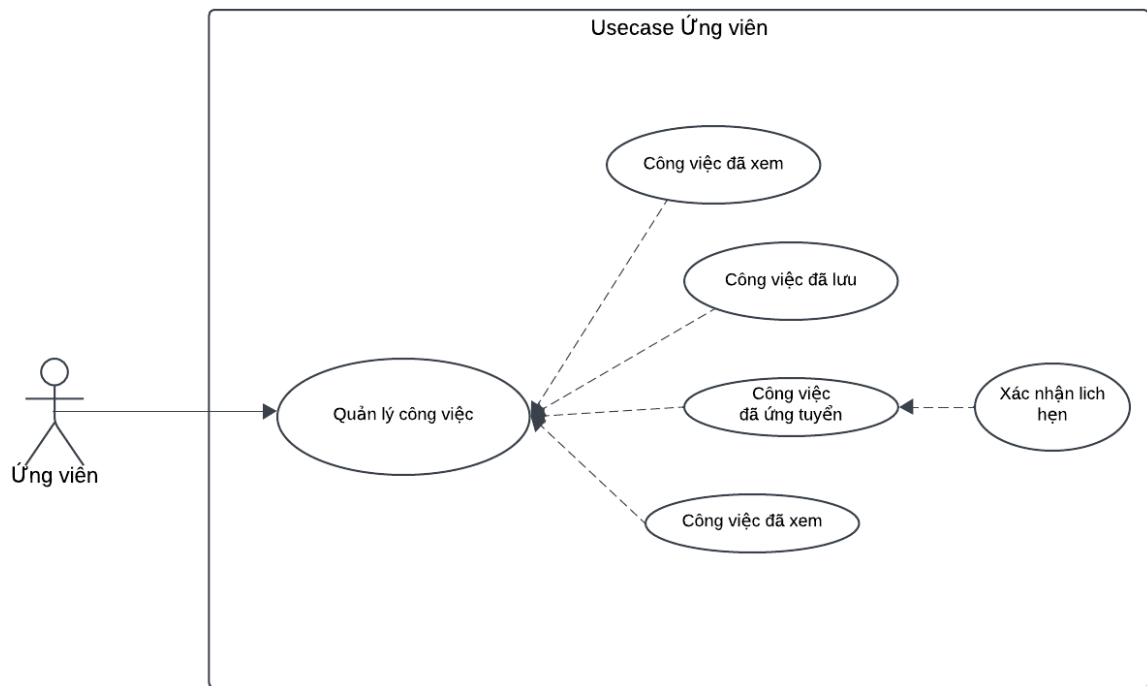
2.3.1. Lược đồ Use case

2.3.1.1. Use case Khách

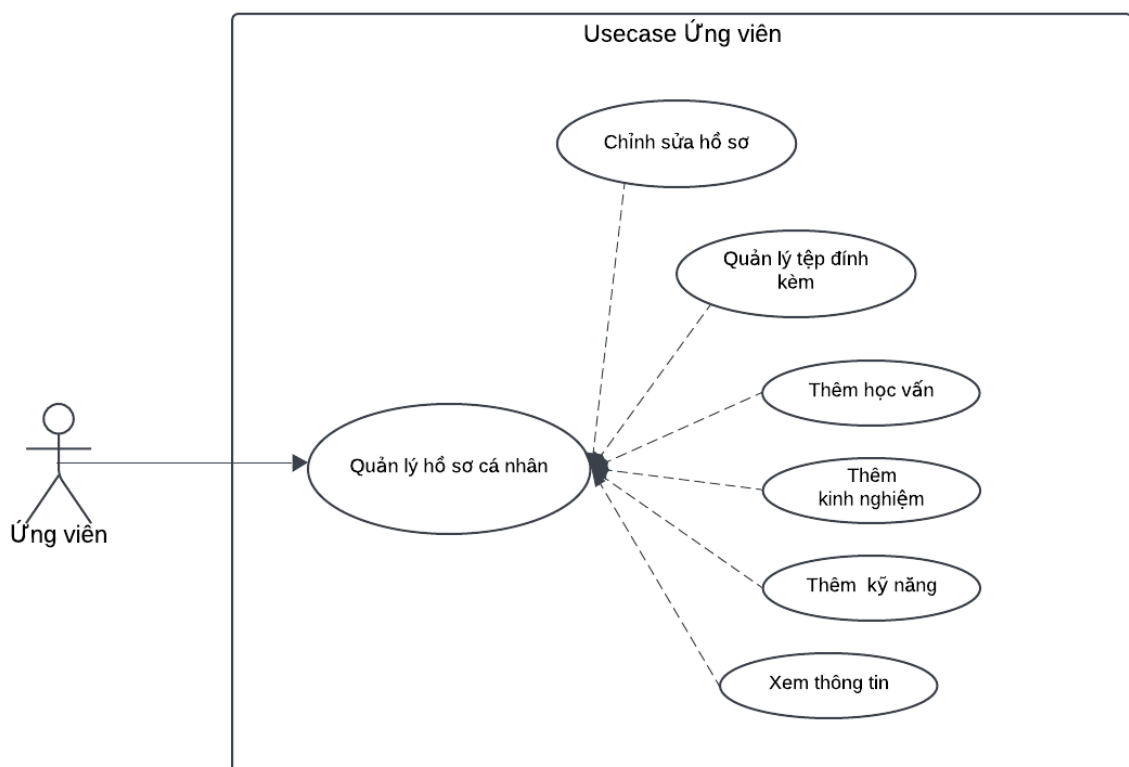


Hình 4. Use case tổng quan của vai trò Khách

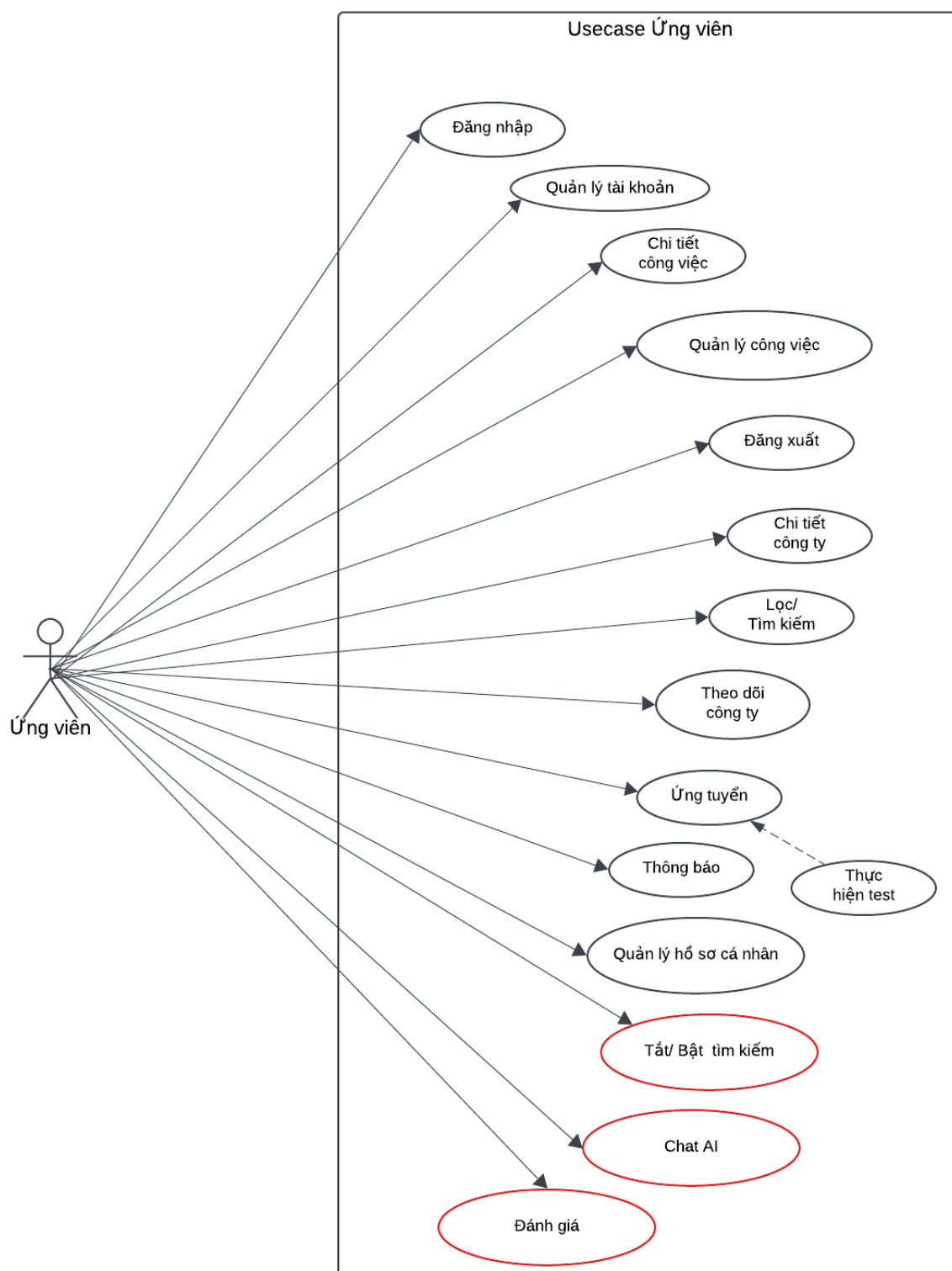
2.3.1.2. Use case Ứng viên



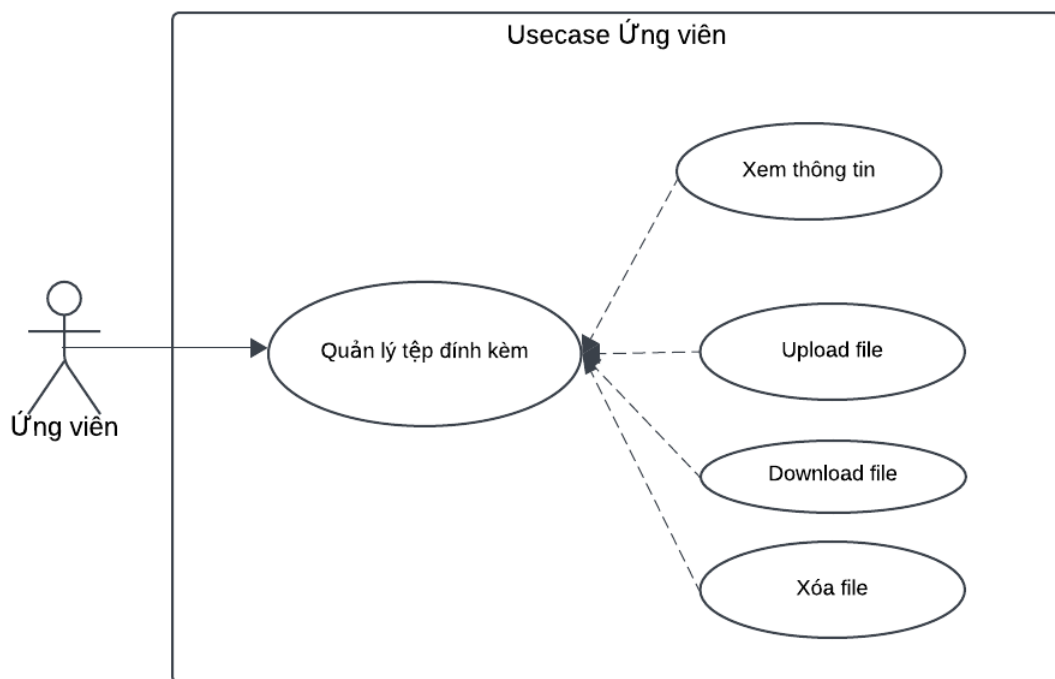
Hình 5. Use case quản lý công việc (Ứng viên)



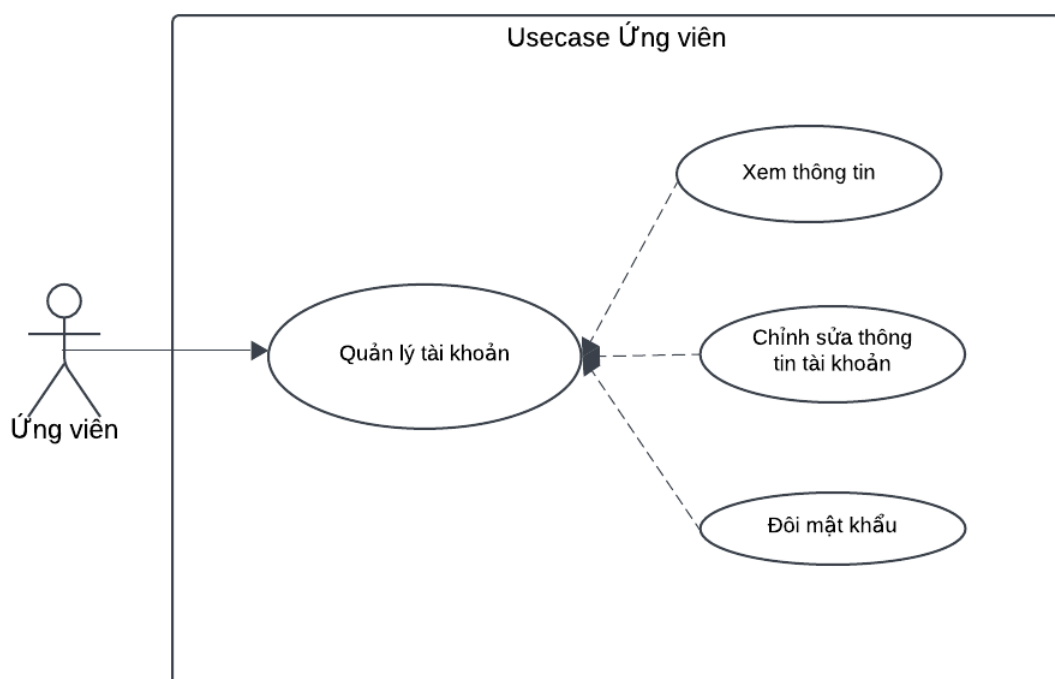
Hình 6. Use case quản lý hồ sơ cá nhân (Ứng viên)



Hình 7. Use case tổng quan của vai trò Ứng viên

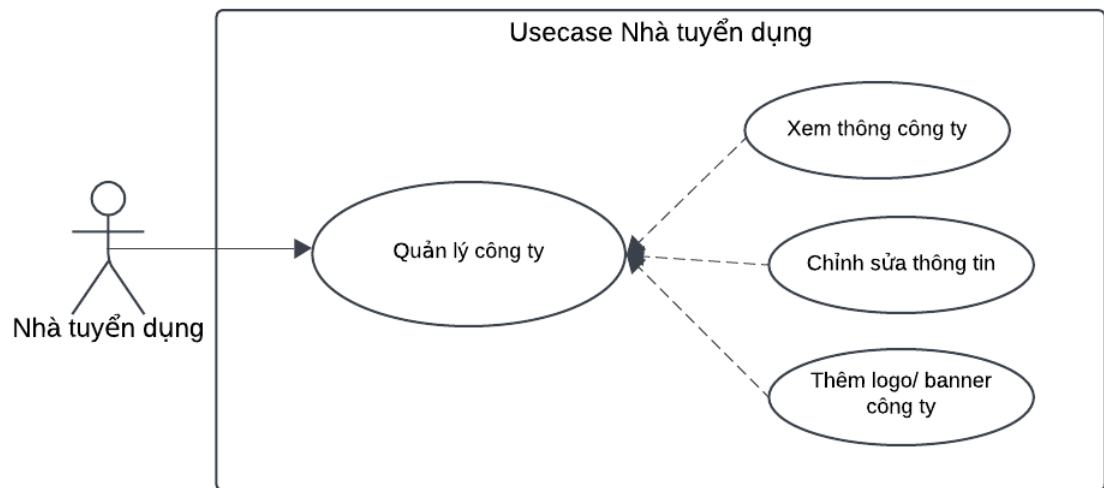


Hình 8. Use case quản lý tệp đính kèm (Ứng viên)

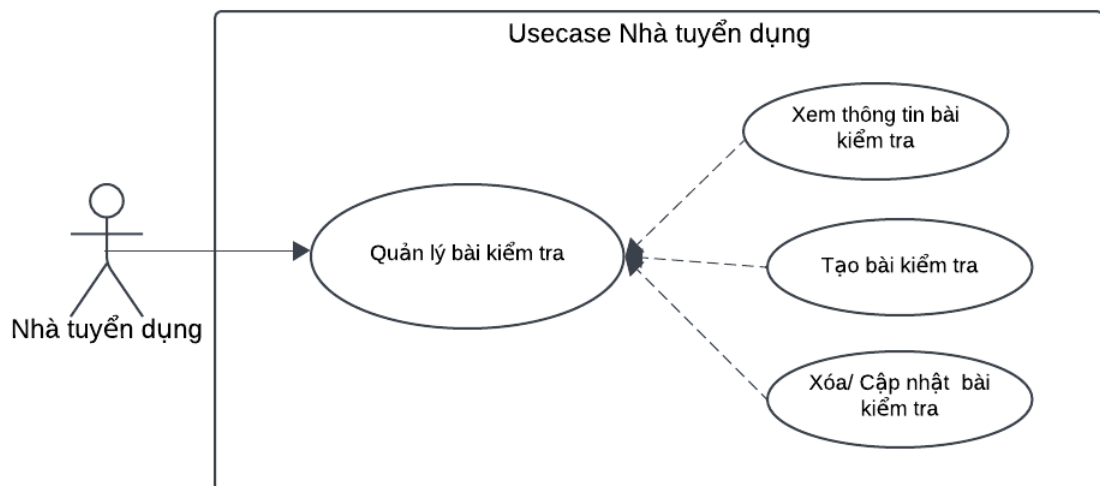


Hình 9. Use case quản lý tài khoản (Ứng viên)

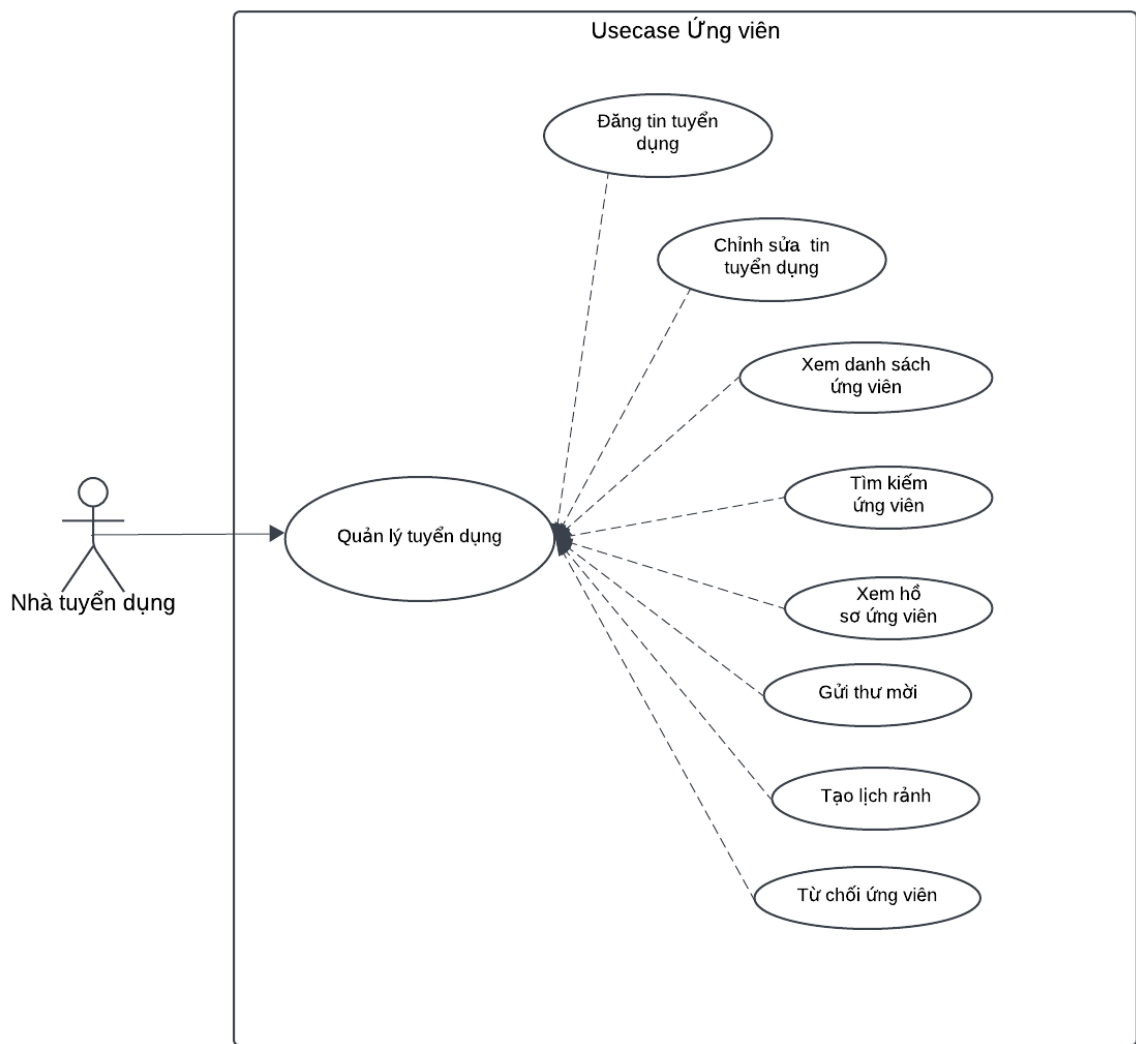
2.3.1.3. Use case Nhà tuyển dụng



Hình 10. Use case quản lý công ty (Nhà tuyển dụng)



Hình 11. Use case quản lý bài test (Nhà tuyển dụng)

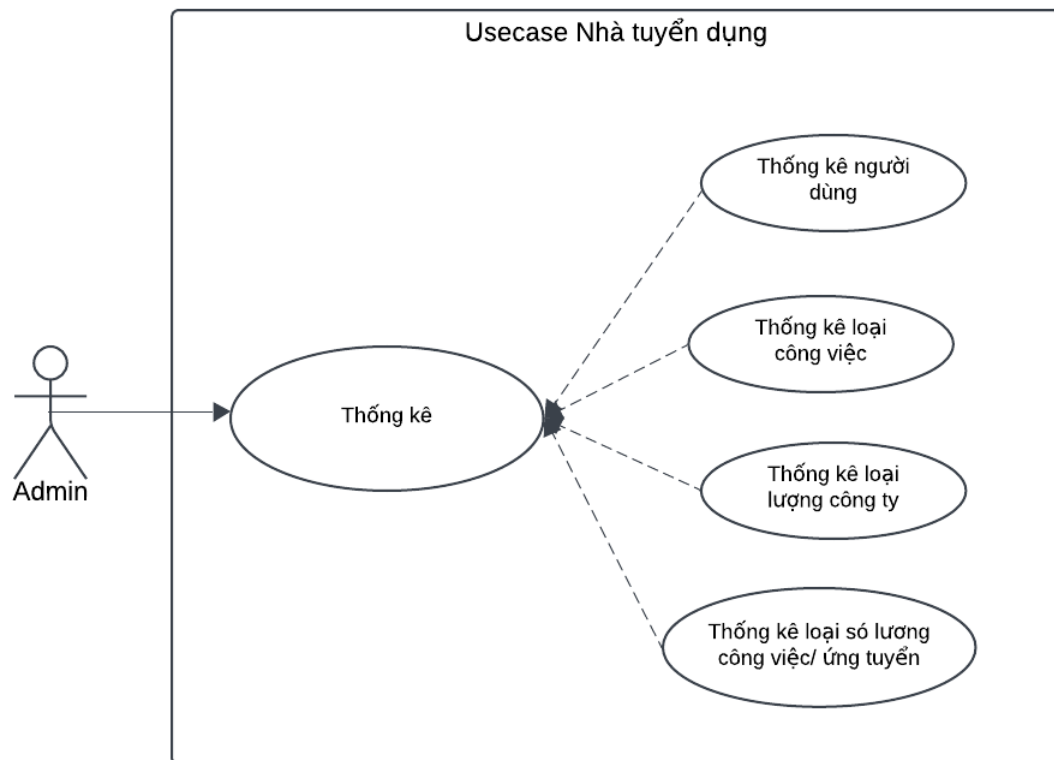


Hình 12. Use case quản lý tuyển dụng (Nhà tuyển dụng)

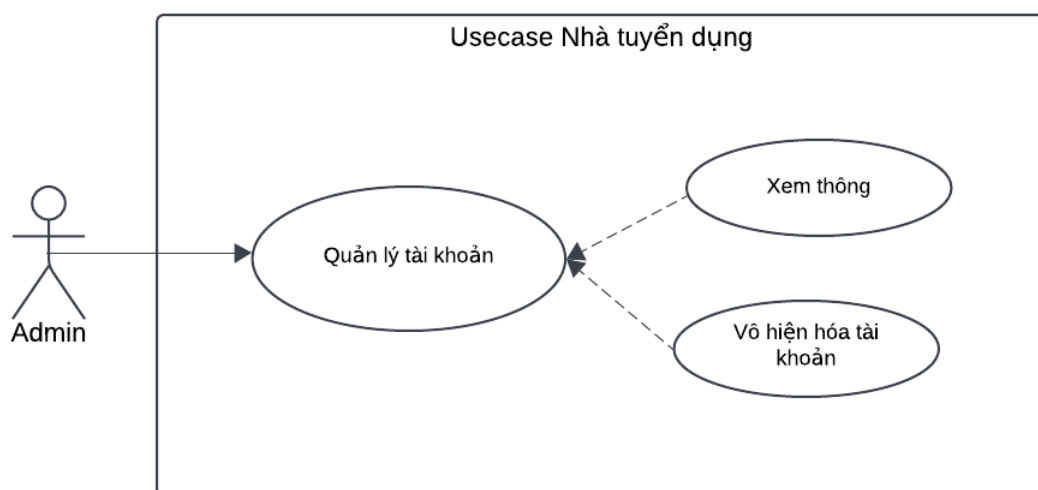


Hình 13. Use case tổng quan của vai trò Nhà tuyển dụng

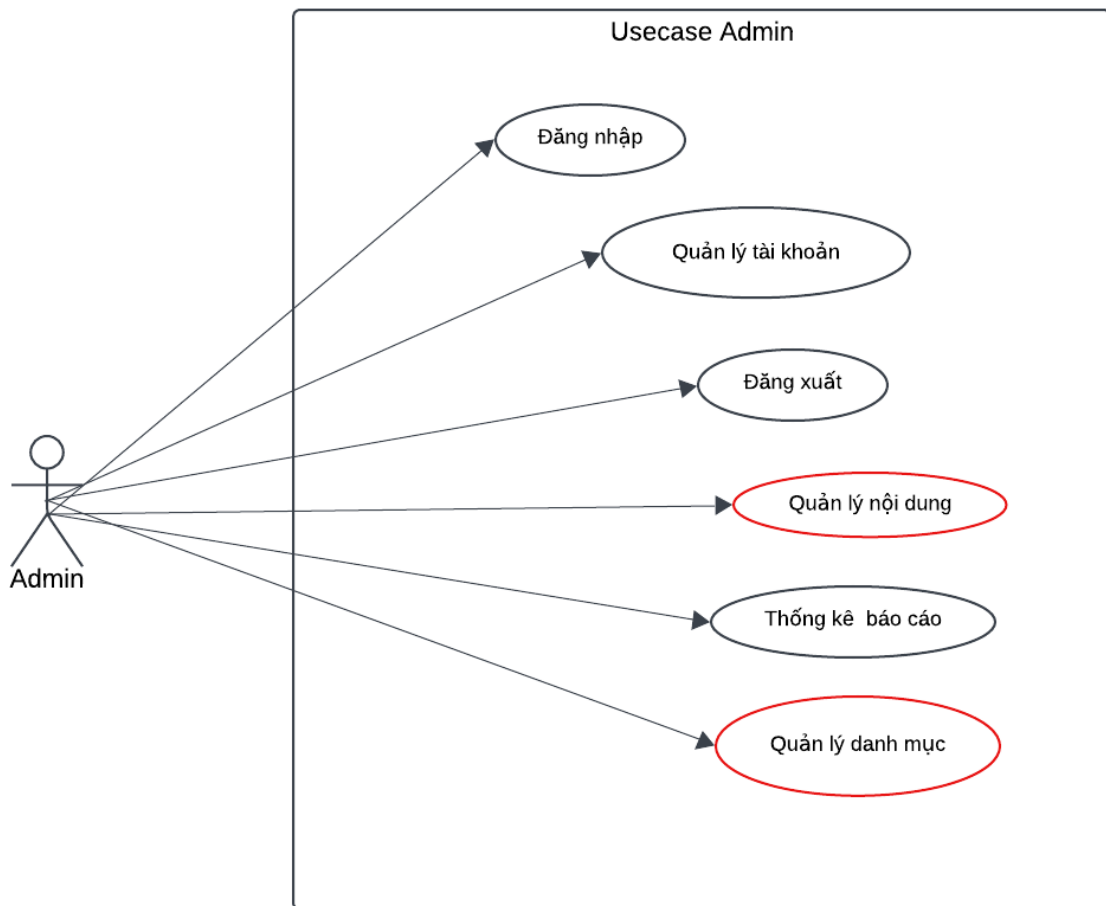
2.3.1.4. Use case Admin



Hình 14. Use case thống kê (Admin)



Hình 15. Use case quản lý tài khoản (Admin)



Hình 16. Use case tổng quan của vai trò Admin

2.3.2. Mô tả use case

Bảng 1. Mô tả use case đăng ký ứng viên

Use Case ID	UC_01
Use Case Name	Đăng ký ứng viên
Description	Ứng viên tạo tài khoản để sử dụng hệ thống tìm kiếm việc làm.
Actor	Ứng viên
Precondition	Người dùng truy cập trang web. Chưa có tài khoản trên hệ thống.
Basic Flow	Ứng viên chọn tùy chọn Sign Up Nhập thông tin: Username, Email, Password, Confirm Password Hệ thống kiểm tra tính hợp lệ của thông tin. Hệ thống lưu thông tin vào cơ sở dữ liệu. Hiển thị thông báo "Đăng ký thành công".
Alternative Flow	3.1 Nếu email/username đã tồn tại: Hiển thị thông báo "Tài khoản đã được sử dụng". 3.2 Nếu password yếu: Yêu cầu nhập lại mật khẩu.
Exception Flow	N/A

Bảng 2. Mô tả use case đăng ký nhà tuyển dụng

Use Case ID	UC_02
Use Case Name	Đăng ký nhà tuyển dụng
Description	Nhà tuyển dụng tạo tài khoản để đăng tin tuyển dụng và quản lý công ty.
Actor	Nhà tuyển dụng
Precondition	Người dùng truy cập trang web. Chưa có tài khoản trên hệ thống.
Basic Flow	Nhà tuyển dụng chọn tùy chọn "Đăng ký" Nhập thông tin: Username, Email, Password, Số điện thoại, Tên công ty, Địa chỉ, Lĩnh vực công ty. Hệ thống kiểm tra tính hợp lệ của thông tin. Hệ thống lưu thông tin vào cơ sở dữ liệu. Hiển thị thông báo "Đăng ký thành công".
Alternative Flow	3.1 Nếu email hoặc username đã tồn tại: Hiển thị thông báo "Email hoặc Username đã được sử dụng". 3.2 Nếu số điện thoại không hợp lệ: Yêu cầu nhập lại số điện thoại. 3.3 Nếu các trường thông tin bắt buộc trống: Hiển thị thông báo yêu cầu điền đầy đủ.
Exception Flow	N/A

Bảng 3. Mô tả use case đăng nhập tài khoản

Use Case ID	UC_03
Use Case Name	Đăng nhập tài khoản
Description	Người dùng (ứng viên hoặc nhà tuyển dụng) đăng nhập bằng email và mật khẩu để truy cập hệ thống.
Actor	Người dùng (Ứng viên/Nhà tuyển dụng)
Precondition	Người dùng đã có tài khoản hợp lệ. Tài khoản không bị vô hiệu hóa.
Basic Flow	Người dùng truy cập trang đăng nhập. Nhập thông tin: Email và Password. Hệ thống kiểm tra thông tin tài khoản Hiển thị giao diện phù hợp với loại tài khoản (Ứng viên hoặc Nhà tuyển dụng).
Alternative Flow	3.1 Thông tin email hoặc mật khẩu không đúng: Hiển thị thông báo "Email hoặc mật khẩu không chính xác". 3.2 Tài khoản bị vô hiệu hóa: Hiển thị thông báo "Tài khoản đã bị vô hiệu hóa, vui lòng liên hệ quản trị viên để biết thêm chi tiết".

Bảng 4. Mô tả use case xem hồ sơ cá nhân

Use Case ID	UC_04
Use Case Name	Xem hồ sơ cá nhân
Description	Ứng viên xem hồ sơ cá nhân của mình
Actor	Ứng viên
Precondition	Đăng nhập thành công dưới quyền là Ứng viên Hệ thống hoạt động ổn định.
Basic Flow	Ứng viên bấm vào Avatar trên góc phải màn hình Ứng viên chọn vào Hồ sơ cá nhân Người dùng xem thông tin cơ bản và các mục quản lý như CV, học vấn, kỹ năng.
Alternative Flow	N/A
Exception Flow	N/A

Bảng 5. Mô tả use case upload file

Use Case ID	UC_05
Use Case Name	Upload file
Description	Người dùng thêm, xóa và xem trước tệp CV trên hồ sơ cá nhân.
Actor	Ứng viên
Precondition	Giao diện hồ sơ cá nhân đã hiển thị.
Basic Flow	Người dùng bấm vào Thêm hồ sơ Giao diện hiển thị nơi chọn file Người dùng bấm chọn file mình muốn tải lên Hệ thống load CV lên giao diện Lưu CV xuống database Thông báo thêm CV thành công.
Alternative Flow	N/A
Exception Flow	N/A

Bảng 6. Mô tả use case xem chi tiết công việc

Use Case ID	UC_06
Use Case Name	Xem chi tiết công việc
Description	Người dùng (ứng viên và nhà tuyển dụng) có thể xem thông tin chi tiết của một công việc bao gồm mô tả, yêu cầu, mức lương, địa điểm và thông tin công ty.
Actor	Ứng viên, Nhà tuyển dụng
Precondition	Công việc tồn tại trong cơ sở dữ liệu.
Basic Flow	Người dùng truy cập danh sách công việc. Người dùng chọn một công việc và bấm vào để xem chi tiết. Hệ thống hiển thị trang chi tiết công việc
Alternative Flow	N/A
Exception Flow	N/A

Bảng 7. Mô tả use case lọc và tìm kiếm thông tin

Use Case ID	UC_07
Use Case Name	Lọc và tìm kiếm thông tin
Description	Người dùng có thể tìm kiếm thông tin công việc hoặc ứng viên theo từ khóa và sử dụng các tiêu chí lọc để thu hẹp kết quả.
Actor	Ứng viên, Nhà tuyển dụng, Khách
Precondition	Hệ thống đã có dữ liệu (công việc hoặc ứng viên) Người dùng truy cập trang tìm kiếm.
Basic Flow	Người dùng nhập từ khóa vào ô tìm kiếm (ví dụ: tên công việc, tên công ty). Hệ thống tìm kiếm thông tin theo từ khóa và hiển thị danh sách kết quả phù hợp. Người dùng sử dụng các tiêu chí lọc để thu hẹp kết quả (Lĩnh vực công việc, Địa điểm làm việc, Mức lương, Kinh nghiệm,...) Hệ thống áp dụng các tiêu chí lọc và cập nhật danh sách kết quả phù hợp. Người dùng bấm vào một kết quả để xem chi tiết.
Alternative Flow	N/A
Exception Flow	N/A

Bảng 8. Mô tả use case ứng tuyển công việc

Use Case ID	UC_08
Use Case Name	Ứng tuyển công việc
Description	Người dùng (ứng viên) ứng tuyển vào công việc. Nếu hồ sơ chưa đầy đủ thông tin, hệ thống yêu cầu bổ sung. Nếu công việc yêu cầu làm bài kiểm tra, ứng viên phải thực hiện và đạt điểm trên 50 mới được ứng tuyển thành công.
Actor	Ứng viên
Precondition	Người dùng đã đăng nhập và có hồ sơ cá nhân. Công việc có nút Ứng tuyển được hiển thị. Công việc có thể yêu cầu kiểm tra (nếu có).
Basic Flow	Người dùng bấm vào công việc muốn ứng tuyển để xem chi tiết. Người dùng bấm nút Ứng tuyển. Hệ thống kiểm tra thông tin hồ sơ của ứng viên Nếu công việc yêu cầu kiểm tra test, hệ thống hiển thị bài kiểm tra cho ứng viên Ứng viên làm bài kiểm tra: Nếu công việc không yêu cầu kiểm tra, hệ thống sẽ thông báo "Ứng tuyển thành công" ngay lập tức và lưu thông tin ứng tuyển vào cơ sở dữ liệu.
Alternative Flow	3.1 Nếu hồ sơ chưa đầy đủ (ví dụ: thiếu thông tin, không có CV), hệ thống hiển thị một modal yêu cầu ứng viên bổ sung thông tin. 3.2 Nếu hồ sơ đầy đủ, người dùng có thể bấm Lưu hồ sơ và tiếp tục ứng tuyển. 5.1 Nếu ứng viên đạt trên 50 điểm, hệ thống hiển thị nút Ứng tuyển và lưu thông tin vào cơ sở dữ liệu. 5.2 Nếu ứng viên không đạt, hệ thống hiển thị thông báo "Điểm không đủ, vui lòng làm lại bài kiểm tra" và yêu cầu ứng viên thực hiện lại bài kiểm tra.
Exception Flow	N/A

Bảng 9. Mô tả use case chỉnh sửa hồ sơ

Use Case ID	UC_09
Use Case Name	Chỉnh sửa hồ sơ
Description	Chức năng này mô tả quá trình người dùng chỉnh sửa thông tin trong hồ sơ cá nhân của mình trên nền tảng tuyển dụng, bao gồm thay đổi các thông tin như tên, địa chỉ, kinh nghiệm làm việc, kỹ năng, và ảnh đại diện.
Actor	Ứng viên
Precondition	Đăng nhập vào hệ thống với quyền là ứng viên.
Basic Flow	<p>Người dùng vào trang chỉnh sửa hồ sơ từ menu cá nhân.</p> <p>Hệ thống hiển thị các trường thông tin hồ sơ hiện tại của người dùng.</p> <p>Người dùng thực hiện các thay đổi mong muốn (ví dụ: cập nhật thông tin cá nhân, thêm kinh nghiệm làm việc, tải ảnh đại diện mới).</p> <p>Người dùng nhấn nút "Lưu" để lưu các thay đổi.</p> <p>Hệ thống kiểm tra tính hợp lệ của các dữ liệu đầu vào.</p> <p>Nếu tất cả dữ liệu hợp lệ, hệ thống cập nhật hồ sơ và hiển thị thông báo thành công.</p>
Alternative Flow	<p>3.1 Nếu người dùng không muốn chỉnh sửa một phần thông tin, họ có thể bỏ qua các trường đó và chỉ lưu các thay đổi còn lại.</p> <p>5.1 Nếu người dùng không nhập đầy đủ thông tin bắt buộc, hệ thống sẽ thông báo lỗi và yêu cầu người dùng nhập lại.</p>
Exception Flow	

Bảng 10. Mô tả use case đổi mật khẩu

Use Case ID	UC_10
Use Case Name	Đổi mật khẩu
Description	Use case này mô tả quy trình người dùng thay đổi mật khẩu tài khoản của họ để đảm bảo an toàn hoặc khi muốn cập nhật thông tin đăng nhập.
Actor	Ứng viên, Nhà tuyển dụng
Precondition	Người dùng phải đã đăng nhập Người dùng cần nhớ mật khẩu hiện tại để thực hiện thao tác đổi mật khẩu.
Basic Flow	<p>Người dùng vào trang "Đổi mật khẩu" trong phần cài đặt tài khoản.</p> <p>Hệ thống hiển thị các trường để nhập: mật khẩu hiện tại, mật khẩu mới, và xác nhận mật khẩu mới.</p> <p>Người dùng nhập thông tin cần thiết và nhấn nút "Lưu thay đổi".</p> <p>Hệ thống kiểm tra tính hợp lệ của các thông tin nhập vào (mật khẩu hiện tại đúng, mật khẩu mới khớp với xác nhận mật khẩu, mật khẩu mới không trùng với mật khẩu cũ).</p> <p>Nếu hợp lệ, hệ thống cập nhật mật khẩu mới cho tài khoản của người dùng và hiển thị thông báo thành công.</p>
Alternative Flow	4.1 Nếu người dùng nhập sai mật khẩu hiện tại, hệ thống sẽ thông báo lỗi và yêu cầu nhập lại
Exception Flow	

Bảng 11. Mô tả use case đăng tin tuyển dụng

Use Case ID	UC_11
Use Case Name	Đăng tin tuyển dụng
Description	Use case này mô tả quy trình nhà tuyển dụng đăng tin tuyển dụng mới
Actor	Nhà tuyển dụng
Precondition	Đã đăng nhập vào tài khoản nhà tuyển dụng.
Basic Flow	<p>Nhà tuyển dụng vào mục "Đăng tuyển việc làm mới" trên giao diện hệ thống.</p> <p>Hệ thống hiển thị biểu mẫu để nhập thông tin tuyển dụng (ví dụ: vị trí công việc, mô tả công việc, yêu cầu, mức lương, địa điểm làm việc, thời hạn ứng tuyển).</p> <p>Nhà tuyển dụng điền đầy đủ thông tin và nhấn nút "Lưu".</p> <p>Hệ thống kiểm tra tính hợp lệ của thông tin nhập vào (các trường bắt buộc đã được điền đầy đủ).</p> <p>Nếu hợp lệ, hệ thống đăng tải tin tuyển dụng và hiển thị thông báo thành công. Tin sẽ xuất hiện trên trang tìm việc của ứng viên.</p>
Alternative Flow	4.1 Nếu thông tin chưa đầy đủ hoặc chưa chính xác, hệ thống hiển thị lỗi và yêu cầu chỉnh sửa trước khi đăng.
Exception Flow	

Bảng 12. Mô tả use case chỉnh sửa tin tuyển dụng

Use Case ID	UC_12
Use Case Name	Chỉnh sửa tin tuyển dụng
Description	Use case này mô tả quy trình nhà tuyển dụng chỉnh sửa thông tin tin tuyển dụng đã đăng trước đó.
Actor	Nhà tuyển dụng
Precondition	Nhà tuyển dụng đã đăng nhập vào tài khoản của mình. Tin tuyển dụng đã được đăng tải.
Basic Flow	<p>Nhà tuyển dụng vào mục "Danh sách công việc" trên giao diện hệ thống.</p> <p>Chọn góc phải màn hình dấu 3 chấm và chọn “Chỉnh sửa”</p> <p>Hệ thống hiển thị biểu mẫu với thông tin hiện tại của tin tuyển dụng.</p> <p>Nhà tuyển dụng thực hiện các thay đổi cần thiết (ví dụ: mô tả công việc, mức lương, yêu cầu).</p> <p>Nhà tuyển dụng nhấn nút "Lưu".</p> <p>Hệ thống kiểm tra tính hợp lệ của thông tin mới.</p> <p>Nếu hợp lệ, hệ thống cập nhật tin tuyển dụng và hiển thị thông báo thành công.</p>
Alternative Flow	<p>5.1. Nếu nhà tuyển dụng không muốn tiếp tục chỉnh sửa, họ có thể nhấn "Hủy" để quay lại danh sách tin tuyển dụng mà không lưu thay đổi.</p> <p>.1. Nếu thông tin không hợp lệ (ví dụ: thiếu thông tin bắt buộc), hệ thống hiển thị thông báo lỗi và yêu cầu sửa lại.</p>
Exception Flow	

Bảng 13. Mô tả use case xem danh sách ứng viên

Use Case ID	UC_13
Use Case Name	Xem danh sách ứng viên
Description	Use case này mô tả quy trình nhà tuyển dụng xem danh sách ứng viên đã ứng tuyển vào một công việc cụ thể.
Actor	Nhà tuyển dụng
Precondition	Nhà tuyển dụng đã đăng nhập vào tài khoản của mình. Công việc phải thuộc danh sách công việc đã đăng bởi nhà tuyển dụng.
Basic Flow	Nhà tuyển dụng vào mục "Tuyển dụng" trên giao diện hệ thống. Hệ thống hiển thị danh sách các công việc mà nhà tuyển dụng đã đăng. Nhà tuyển dụng bấm vào tên của công việc mà họ muốn xem danh sách ứng viên. Hệ thống hiển thị danh sách các ứng viên đã ứng tuyển vào công việc đó.
Alternative Flow	N/A
Exception Flow	Nếu công việc không có ứng viên nào ứng tuyển, hệ thống sẽ hiển thị thông báo: "Hiện tại chưa có ứng viên nào ứng tuyển vào công việc này."

Bảng 14. Mô tả use case xem hồ sơ ứng viên

Use Case ID	UC_14
Use Case Name	Xem hồ sơ ứng viên.
Description	Use case này mô tả quy trình nhà tuyển dụng xem thông tin chi tiết hồ sơ của ứng viên bằng cách bấm vào tên ứng viên trong danh sách ứng viên ứng tuyển vào một công việc.
Actor	Nhà tuyển dụng
Precondition	Nhà tuyển dụng đã đăng nhập vào tài khoản của mình. Nhà tuyển dụng đang xem danh sách ứng viên ứng tuyển vào một công việc cụ thể.
Basic Flow	Nhà tuyển dụng truy cập danh sách ứng viên của một công việc. Hệ thống hiển thị danh sách ứng viên, bao gồm tên, email, ngày ứng tuyển, và trạng thái đơn ứng tuyển. Nhà tuyển dụng bấm vào tên ứng viên muốn xem. Hệ thống hiển thị trang chi tiết hồ sơ ứng viên, bao gồm thông tin cá nhân, kinh nghiệm, học vấn, kỹ năng, và các tài liệu đính kèm (nếu có).
Alternative Flow	N/A
Exception Flow	N/A

Bảng 15. Mô tả use case gửi lời mời

Use Case ID	UC_15
Use Case Name	Gửi lời mời
Description	Use case này mô tả quy trình nhà tuyển dụng sử dụng bộ lọc để chọn ứng viên từ danh sách, sau đó gửi lời mời đến ứng viên với thông tin công việc và lời nhắn kèm theo.
Actor	Nhà tuyển dụng
Precondition	Nhà tuyển dụng đã đăng nhập vào tài khoản của mình. Hệ thống có danh sách ứng viên phù hợp với tiêu chí tuyển dụng.
Basic Flow	Nhà tuyển dụng truy cập danh sách ứng viên trên hệ thống. Nhà tuyển dụng sử dụng bộ lọc để tìm kiếm ứng viên phù hợp. Hệ thống hiển thị danh sách các ứng viên đáp ứng tiêu chí lọc. Nhà tuyển dụng chọn một ứng viên từ danh sách và nhấn nút "Gửi lời mời". Hệ thống hiển thị biểu mẫu lời mời, bao gồm các trường: công việc muốn mời, lời nhắn, và nút "Gửi lời mời". Nhà tuyển dụng điền thông tin công việc và lời nhắn, sau đó nhấn "Gửi lời mời". Hệ thống gửi lời mời đến ứng viên và hiển thị thông báo thành công.
Alternative Flow	5.1. Nhà tuyển dụng có thể hủy bỏ quá trình gửi lời mời bằng cách nhấn "Hủy" và quay lại danh sách ứng viên. 6.1. Nếu thông tin công việc chưa được chọn, hệ thống hiển thị thông báo yêu cầu chọn công việc trước khi gửi.
Exception Flow	

Bảng 16. Mô tả use case tạo lịch phỏng vấn

Use Case ID	UC_16
Use Case Name	Tạo lịch phỏng vấn
Description	Use case này mô tả quy trình nhà tuyển dụng tạo lịch rảnh để sắp xếp buổi phỏng vấn hoặc trao đổi với ứng viên sau khi xem chi tiết hồ sơ ứng viên.
Actor	Nhà tuyển dụng
Precondition	Nhà tuyển dụng đã đăng nhập vào tài khoản Đang xem chi tiết hồ sơ của một ứng viên.
Basic Flow	Nhà tuyển dụng bấm vào nút "Tạo lịch rảnh" trên trang chi tiết ứng viên Hệ thống hiển thị giao diện chọn thời gian rảnh. Nhà tuyển dụng chọn thời gian rảnh (ngày và giờ) và nhấn nút "Thêm". Hệ thống thêm thời gian đã chọn vào danh sách lịch rảnh. Nhà tuyển dụng có thể lặp lại bước 3 để thêm nhiều thời gian rảnh khác. Sau khi hoàn tất, nhà tuyển dụng nhấn nút "Xác nhận". Hệ thống lưu lịch rảnh và hiển thị thông báo thành công.
Alternative Flow	5.1. Nếu nhà tuyển dụng thay đổi ý định họ có thể nhấn vào nút “Hủy” bên cạnh thời gian đó.
Exception Flow	

Bảng 17. Mô tả use case quản lý bài test

Use Case ID	UC_17
Use Case Name	Quản lý bài kiểm tra
Description	Use case này mô tả quy trình quản lý bài kiểm tra, bao gồm việc chỉnh sửa thông tin bài kiểm tra và thêm câu hỏi vào bài kiểm tra.
Actor	Nhà tuyển dụng
Precondition	Người dùng đã đăng nhập vào hệ thống. Danh sách các bài kiểm tra đã được tạo sẵn và hiển thị trên giao diện.
Basic Flow	<p>Người dùng truy cập mục "Quản lý bài kiểm tra" trên hệ thống.</p> <p>Hệ thống hiển thị danh sách các bài kiểm tra hiện có.</p> <p>Người dùng chọn một bài kiểm tra từ danh sách và nhấn nút "Chỉnh sửa".</p> <p>Hệ thống hiển thị giao diện chỉnh sửa thông tin bài kiểm tra và danh sách các câu hỏi hiện tại.</p> <p>Người dùng có thể chỉnh sửa thông tin bài kiểm tra (tên bài kiểm tra, mô tả, thời gian làm bài, v.v.) và nhấn "Lưu".</p> <p>Để thêm câu hỏi mới, người dùng nhấn nút "Thêm câu hỏi".</p> <p>Hệ thống hiển thị biểu mẫu để nhập nội dung câu hỏi, các lựa chọn trả lời và đáp án đúng.</p> <p>Người dùng điền đầy đủ thông tin và nhấn nút "Lưu thay đổi".</p> <p>Hệ thống lưu câu hỏi mới và cập nhật danh sách câu hỏi của bài kiểm tra.</p>
Alternative Flow	8.1. Nếu thông tin câu hỏi không đầy đủ, hệ thống hiển thị thông báo yêu cầu bổ sung trước khi thêm.
Exception Flow	N/A

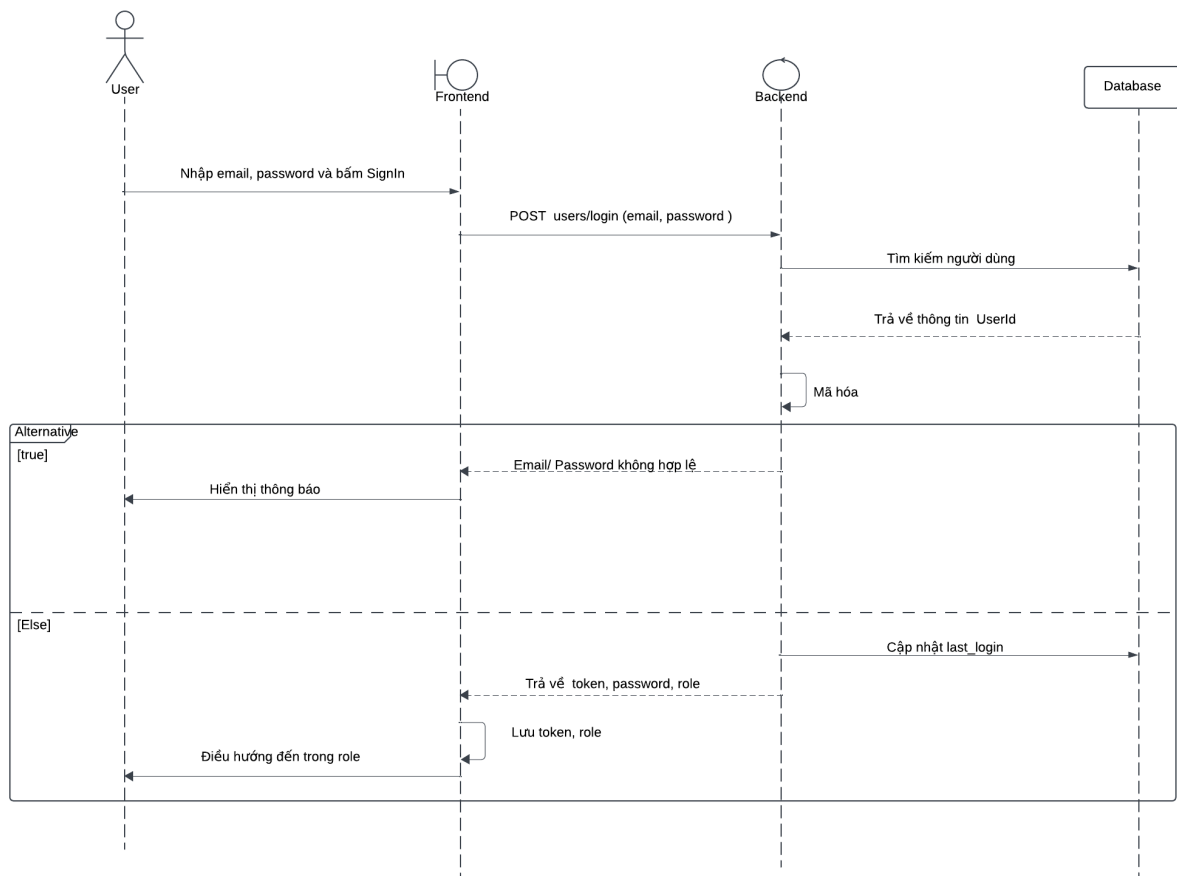
Bảng 18. Mô tả use case vô hiệu hóa tài khoản người dùng

Use Case ID	UC_18
Use Case Name	Vô hiệu hóa tài khoản người dùng
Description	Use case này mô tả quy trình Admin vô hiệu hóa tài khoản người dùng thông qua giao diện chỉnh sửa.
Actor	Admin
Precondition	Admin đã đăng nhập vào hệ thống. Danh sách tài khoản người dùng được hiển thị.
Basic Flow	Admin truy cập danh sách tài khoản người dùng trên hệ thống. Admin bấm vào nút "Chỉnh sửa" bên cạnh tài khoản muốn vô hiệu hóa. Hệ thống hiển thị biểu mẫu chỉnh sửa thông tin tài khoản. Admin tick vào checkbox "Suspend" để vô hiệu hóa tài khoản. Admin nhấn nút "Lưu". Hệ thống lưu trạng thái và hiển thị thông báo thành công.
Alternative Flow	4.1. Nếu Admin bỏ tick vào checkbox "Suspend", hệ thống sẽ kích hoạt lại tài khoản khi Admin nhấn "Lưu".
Exception Flow	

CHƯƠNG 3: THIẾT KẾ PHẦN MỀM

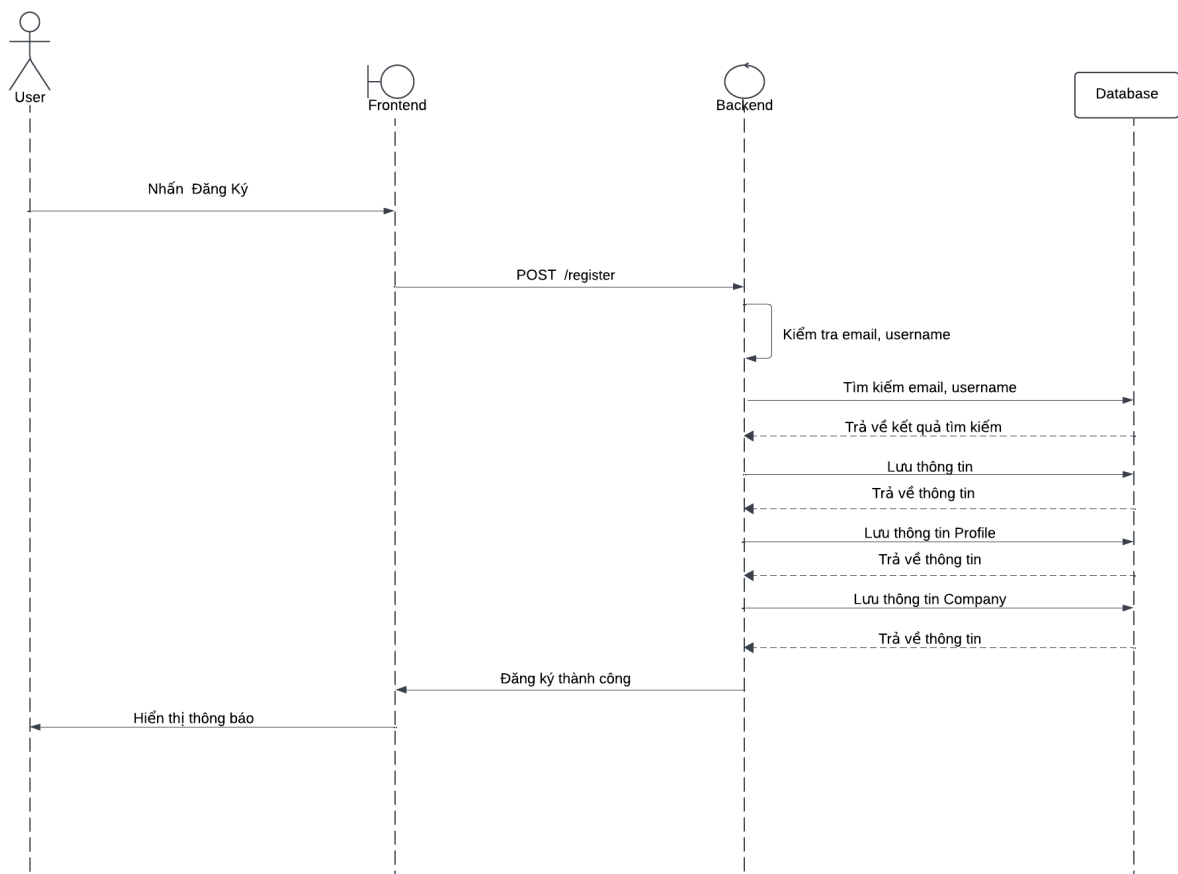
3.1. MỘT SỐ MÔ HÌNH TUẦN TỰ TRỌNG ĐIỂM

3.1.1. Đăng nhập



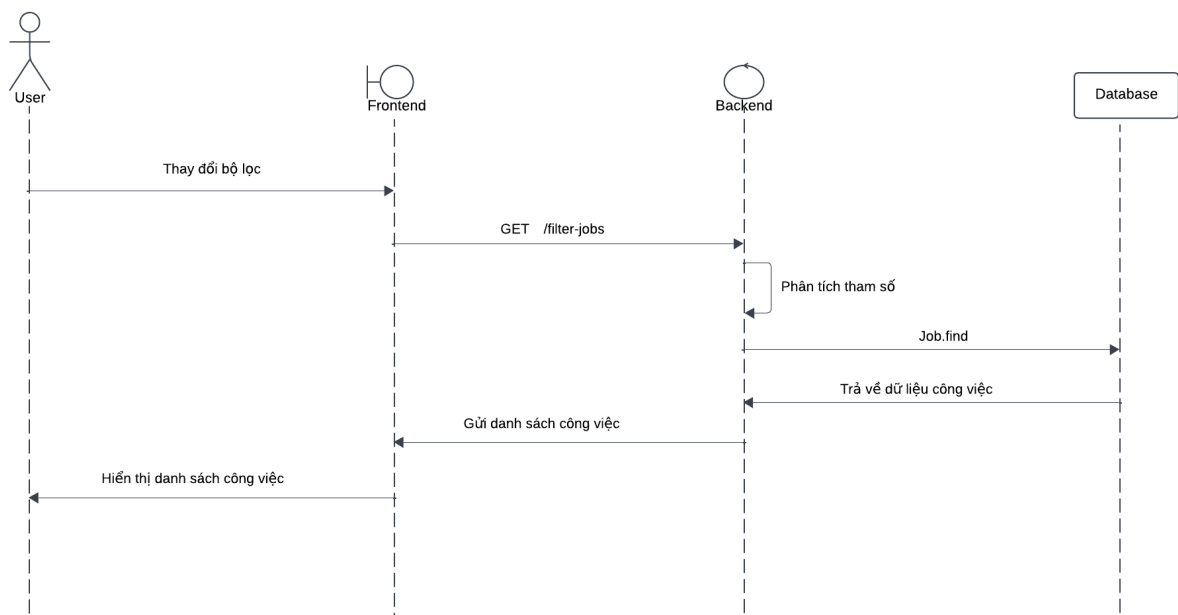
Hình 17. Mô hình tuần tự chức năng đăng nhập

3.1.2. Đăng ký vai trò nhà tuyển dụng



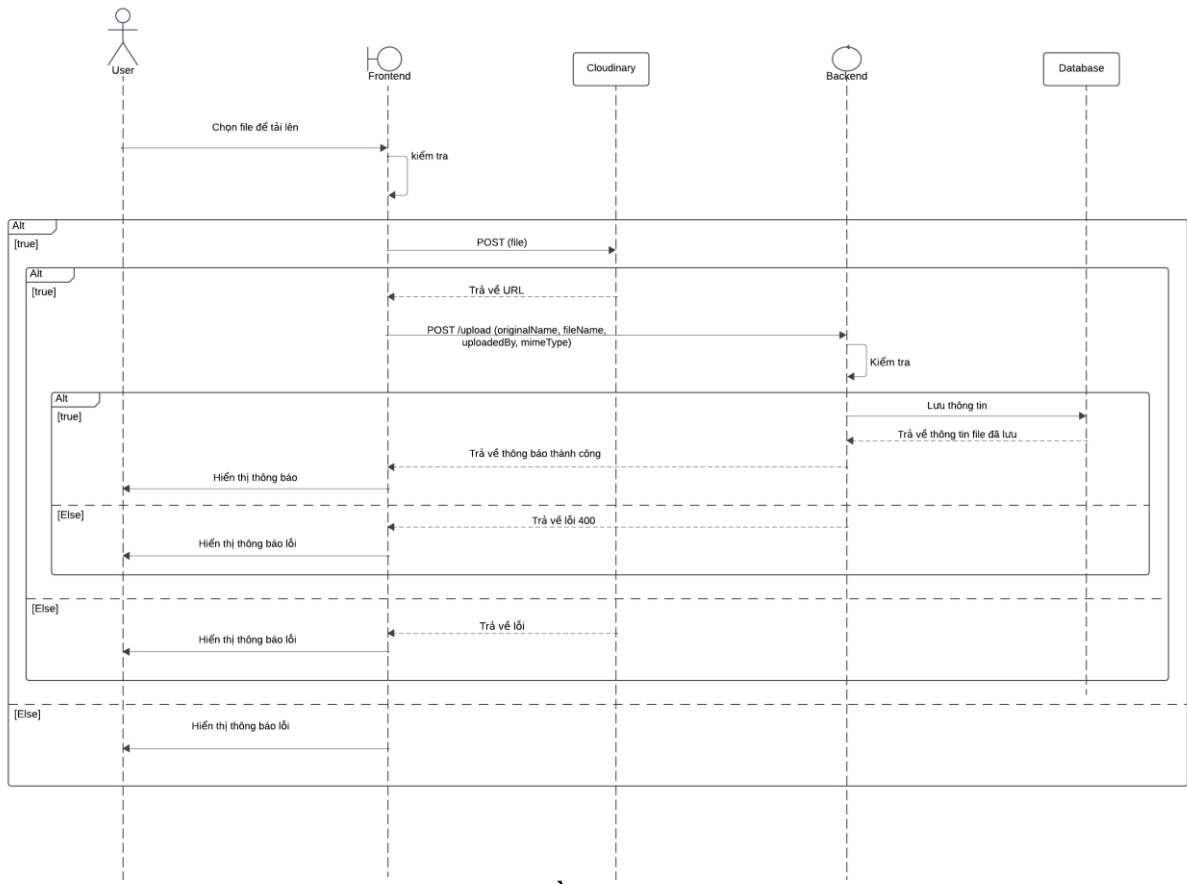
Hình 18. Mô hình tuần tự chức năng đăng ký vai trò nhà tuyển dụng

3.1.3. Lọc công việc



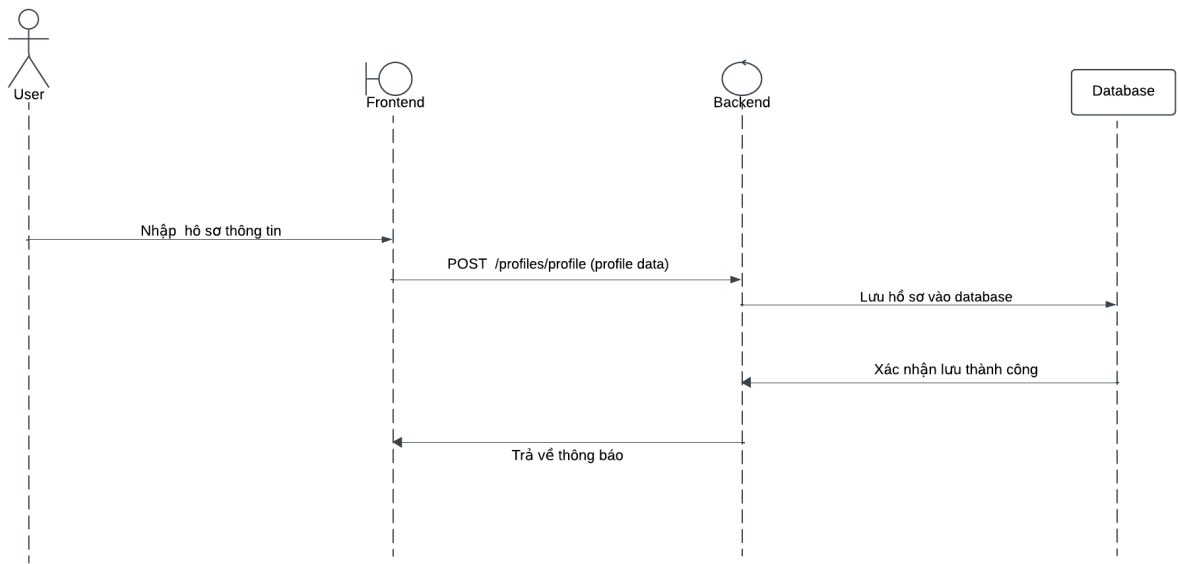
Hình 19. Mô hình tuần tự chức năng lọc công việc

3.1.4. Upload file



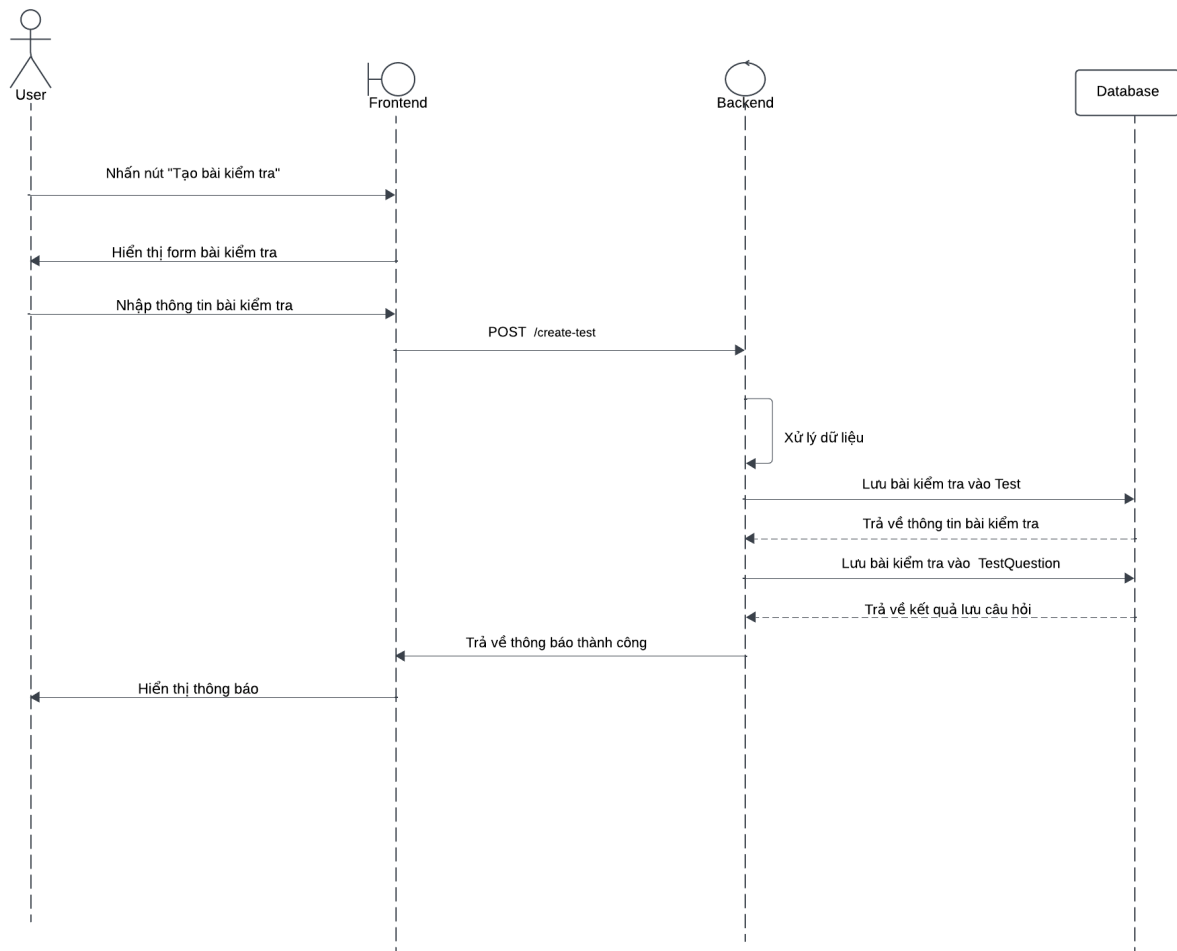
Hình 20. Mô hình tuần tự chức năng upload file

3.1.5. Lưu hồ sơ



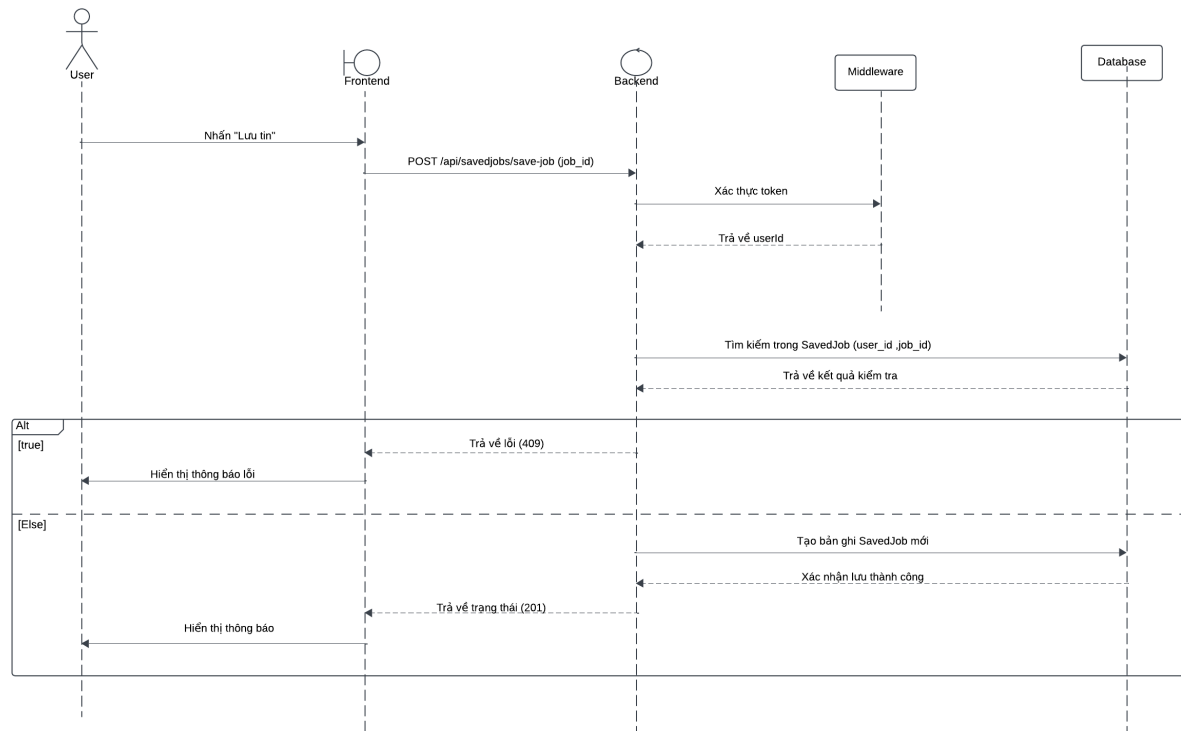
Hình 21. Mô hình tuần tự chức năng lưu hồ sơ

3.1.6. Tạo bài test



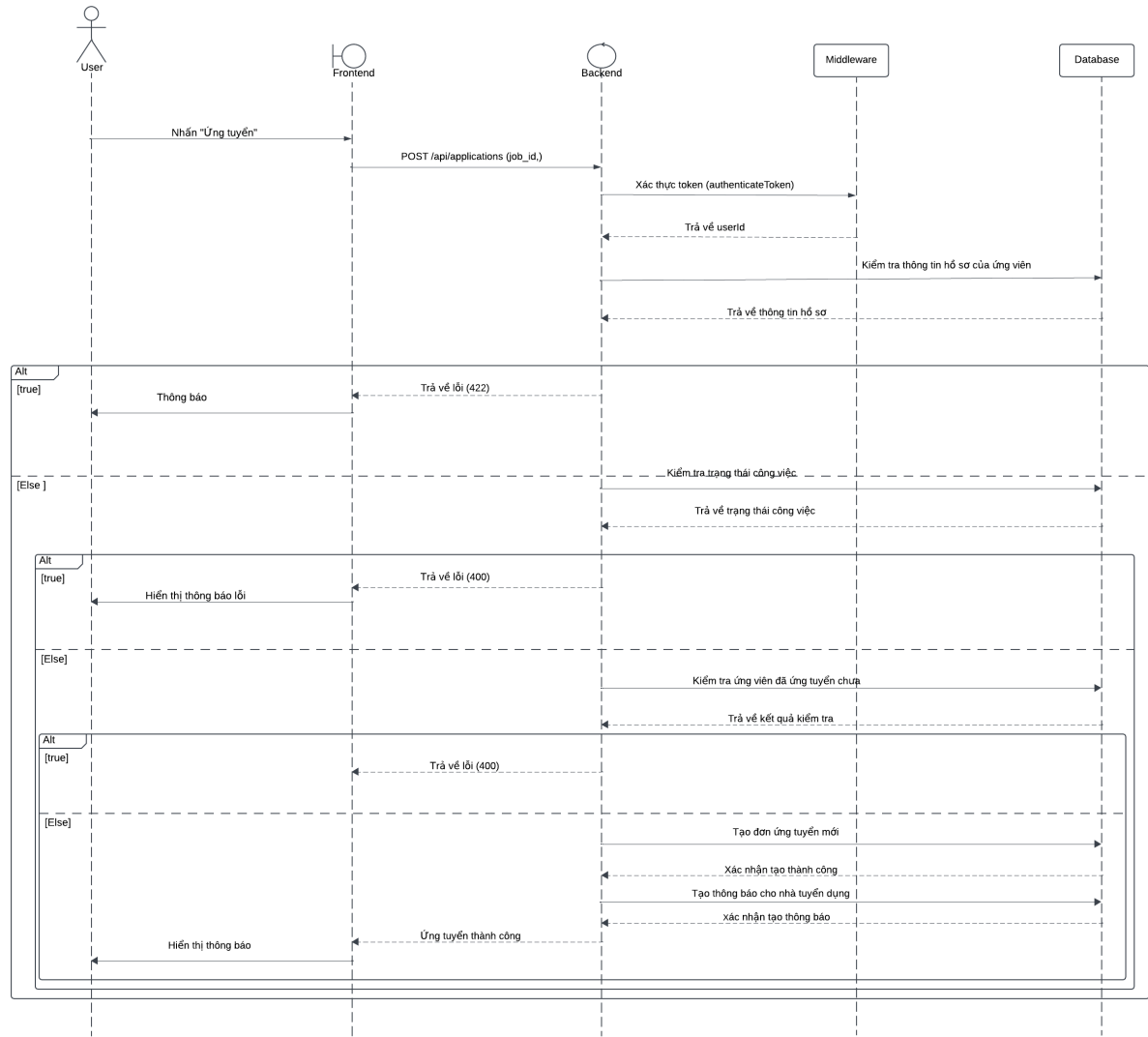
Hình 22. Mô hình tuần tự chức năng tạo bài test

3.1.7. Lưu công việc



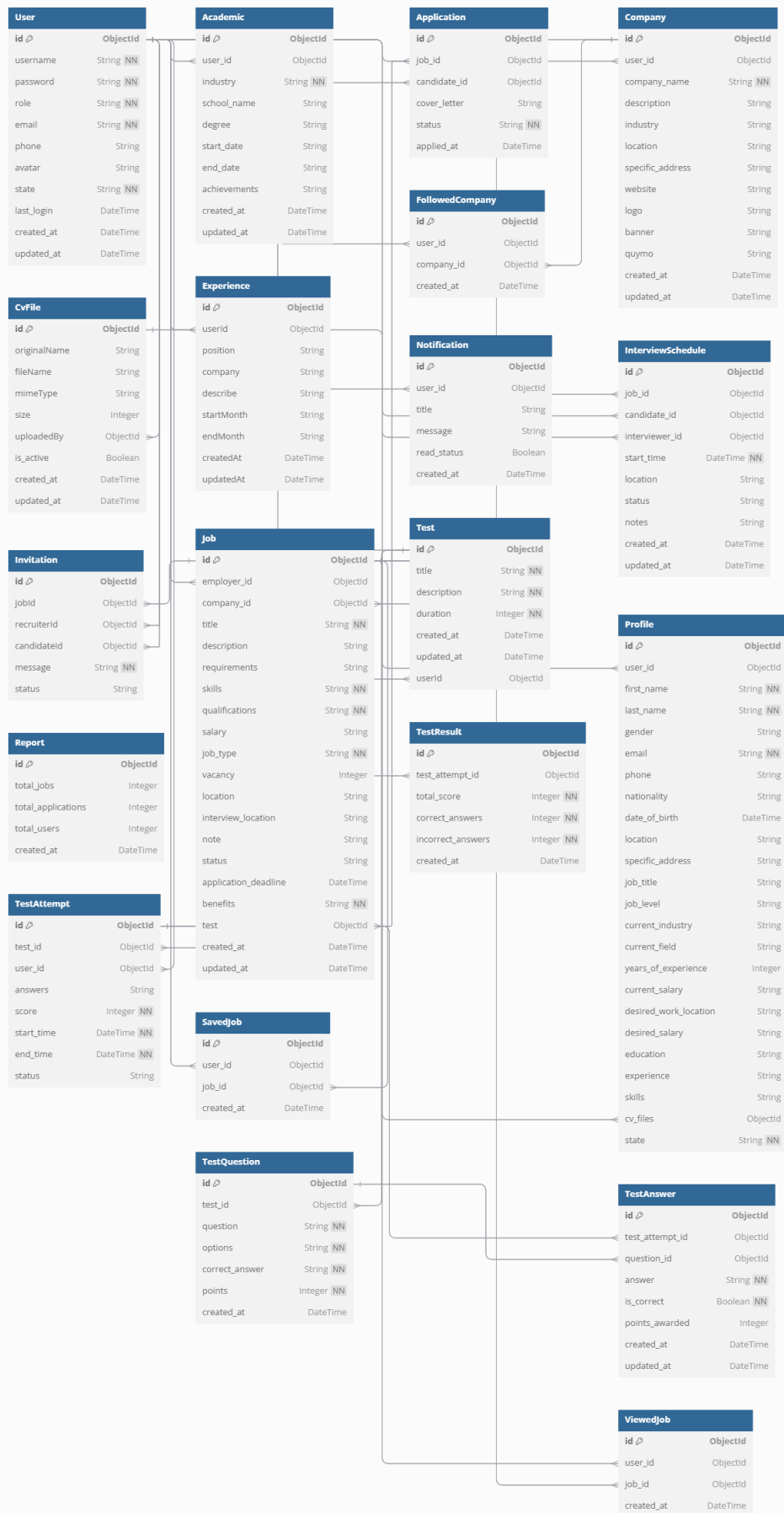
Hình 23. Mô hình tuần tự chức năng lưu công việc

3.1.8. Ứng tuyển



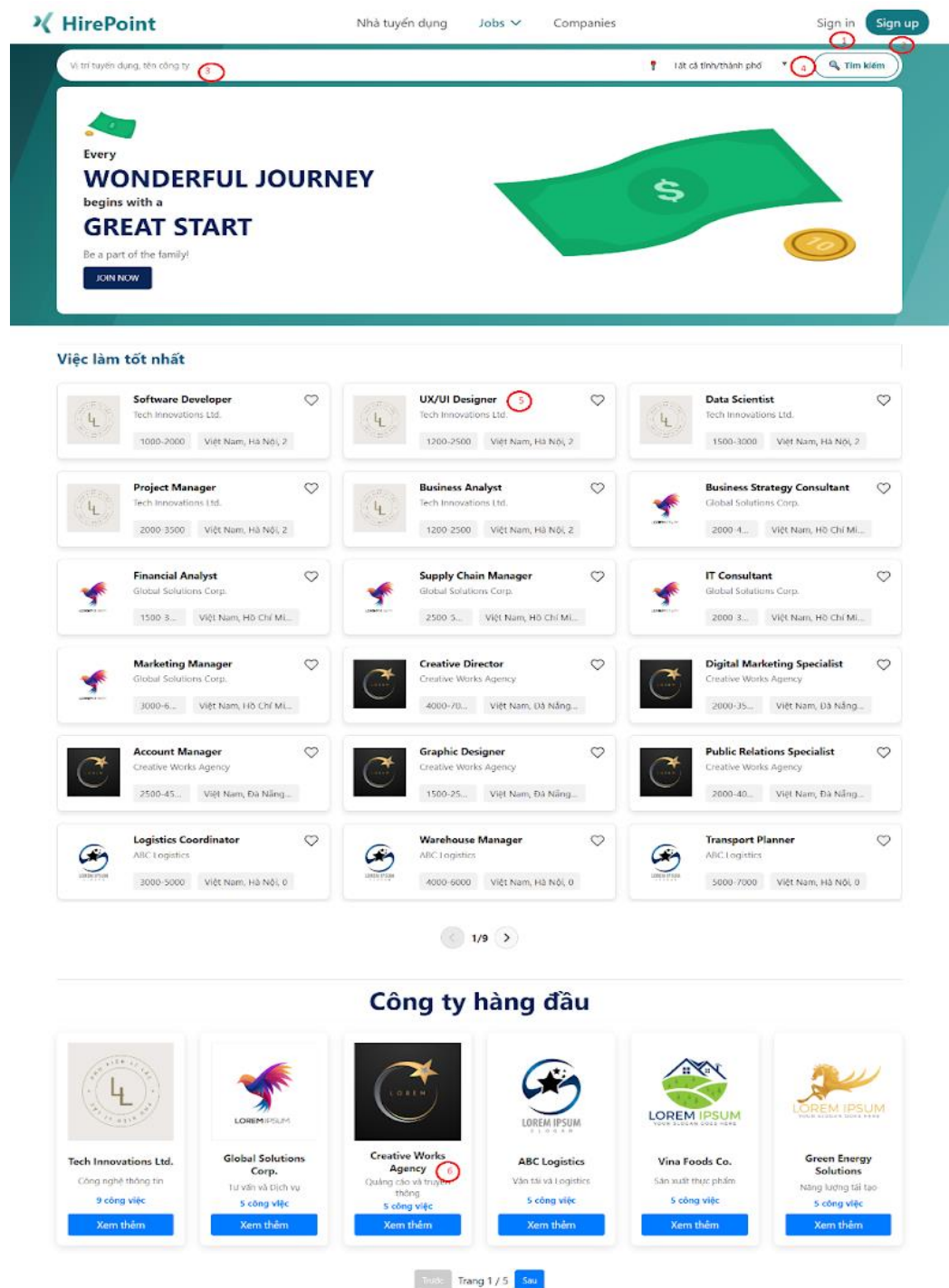
Hình 24. Mô hình tuần tự chức năng ứng tuyển

3.2. LƯỢC ĐỒ LỚP



Hình 25. Lược đồ cơ sở dữ liệu

3.3. THIẾT KẾ GIAO DIỆN

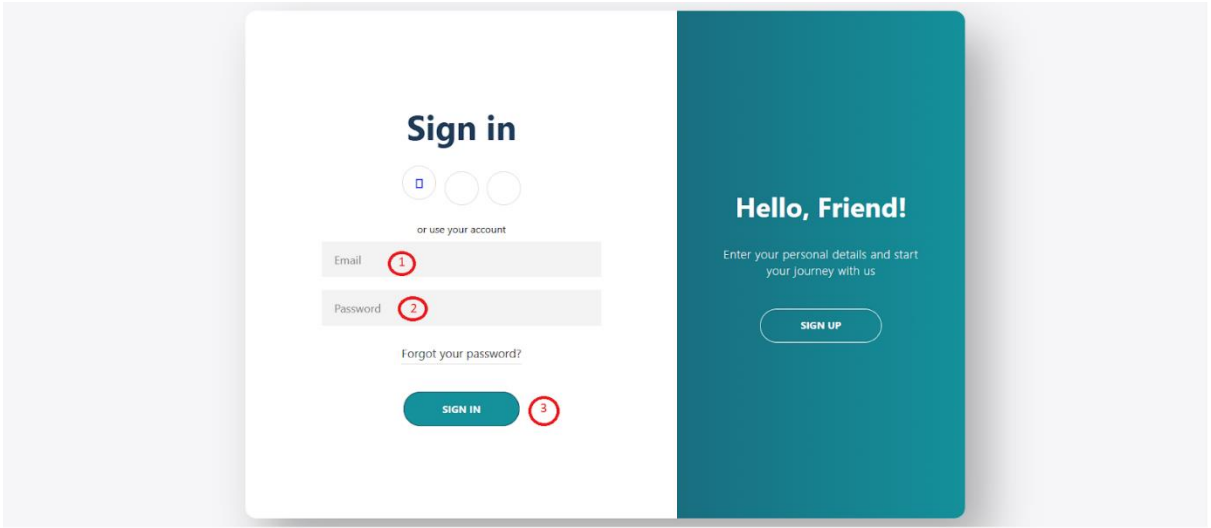


Hình 26. Giao diện trang chủ

Bảng 19. Mô tả ý nghĩa các thành phần giao diện trang chủ

STT	Loại thẻ	Ý nghĩa
1	Button	Hiện thị trang đăng nhập.
2	Button	Hiện thị trang đăng ký.
3	Input	Nơi để nhập thông tin tìm kiếm.

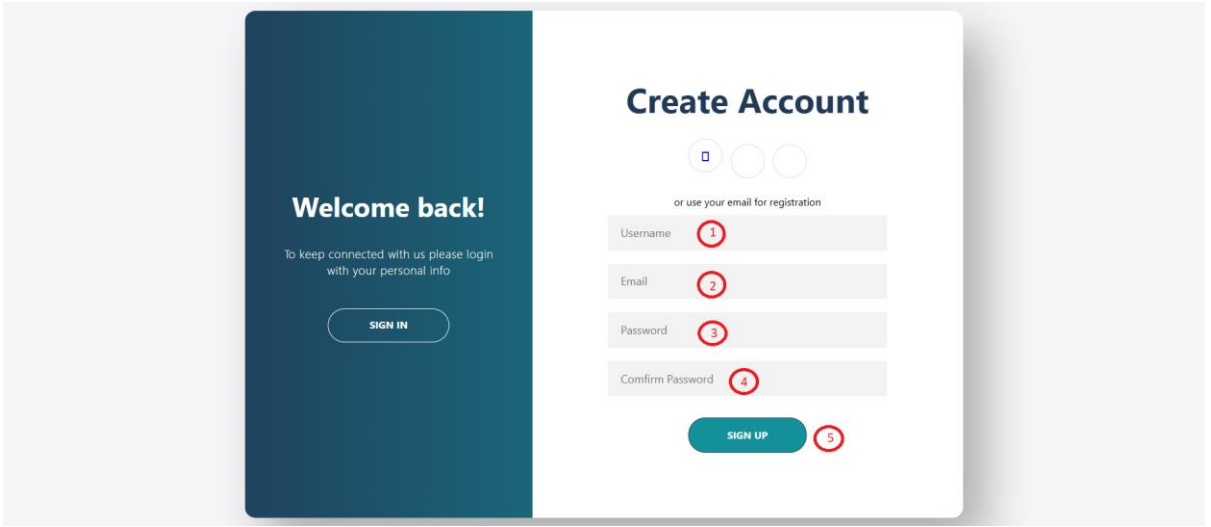
4	Button	Nhấn để thực hiện tìm kiếm.
5	Label	Tên công việc bấm để xem chi tiết.



Hình 27. Giao diện đăng nhập

Bảng 20. Mô tả ý nghĩa các thành phần giao diện đăng nhập


STT	Loại thẻ	Ý nghĩa
1	Input	Ô nhập email.
2	Input	Ô nhập password.
3	Button	Nhấn nút để đăng nhập.




Hình 28. Giao diện đăng ký


Bảng 21. Mô tả ý nghĩa các thành phần giao diện đăng ký

STT	Loại thẻ	Ý nghĩa
1	Input	Ô nhập username.
2	Input	Ô nhập email.
3	Input	Ô nhập password.
4	Input	Ô nhập lại password.
5	Button	Nhấn để đăng ký tài khoản.



Jobs ▾Companies





Nguyễn Văn An
nguyenvanan@gmail.com

Hồ Sơ Của Tôi1


Công Ty Của Tôi2

Việc Làm Của Tôi3

Thông Báo Việc Làm4

Quản Lý Tài Khoản5

Lịch hẹn phỏng vấn6



Thông tin cơ bản7

nguyenvanan@gmail.comTrưởng phòng
Bonaire, Sint Eustatius, and Saba, Bonaire, BarkaderaCử nhân Đại học Bách Khoa Hà Nội - Thạc sĩ Đại học SPKT - Thạc sĩ Đại học SPKT
0912345678

Nguyễn Văn An
Nhân viên
5 years experience

Công việc mong muốn
Nơi làm việc: 123 Đường Nguyễn Thị Minh KhaiMức lương: \$10000/Tháng

Hồ sơ đã tải lên8

CV2.pdf
Cập nhật lần cuối: 09/12/2024
Xem File9

1011

Thông tin học vấn12

Đại học Bách Khoa Hà Nội
Công nghệ thông tin
09/2016 - 06/2020
Đạt học bổng xuất sắc trong suốt 3 năm học, tham gia nghiên cứu khoa học và là thành viên tích cực của câu lạc bộ lập trình. Dẫn dắt nhóm trong các cuộc thi lập trình quốc gia và giành giải Nhì tại cuộc thi hackathon quốc gia.

Đại học SPKT
Công nghệ
12/2023 - 01/2024
Thạc sĩ loại giỏi

Đại học SPKT
Dịch vụ tư vấn
12/2023 - 01/2024
Dịch vụ

Kinh nghiệm làm việc13

Software Engineer
Tech Solutions Ltd.
Từ Tháng 01/2021 đến Tháng 12/2022
Developed backend services and improved database performance for a major client. Worked on implementing new features and bug fixes.

Junior Developer
Innovative Tech Solutions
Từ Tháng 03/2020 đến Tháng 08/2020
Assisted in front-end development of a mobile app, collaborating closely with UI/UX designers to enhance user experience.

Kỹ năng14

JavaScript

Node.js

React

GitHub

Làm nhóm

Word

Hình 29. Giao diện quản lý hồ sơ cá nhân (Ứng viên)

Bảng 22. Mô tả ý nghĩa các thành phần giao diện quản lý hồ sơ cá nhân (Ứng viên)

STT	Loại thể	Ý nghĩa
1	Label	Nhấn để hiển thị hồ sơ cá nhân
2	Label	Nhấn để hiển thị thông tin các công ty là đã theo dõi
3	Label	Nhấn để hiển thị các công việc mà đã ứng tuyển, các công việc đã lưu, các công việc đã xem hay thư mời ứng tuyển.
4	Label	Nhấn để hiển thị thông báo của người dùng.
5	Label	Nhấn để hiển thị tài khoản cá nhân.
6	Label	Nhấn để xem các lịch hẹn phỏng vấn.
7	Icon	Nhấn để hiển thị nơi để chỉnh sửa hồ sơ cá nhân.
8	Icon	Nhấn để hiển thị nơi tải file lên
9	Label	Nhấn để xem file chi tiết
10	Icon	Nhấn để tải file về
11	Icon	Nhấn để xóa file
12	Icon	Nhấn để có thể hiển thị form thêm thông tin học vấn.
13	Icon	Nhấn để hiển thị form thêm kinh nghiệm làm việc.
14	Icon	Nhấn để hiển thị form thêm kỹ năng.

Jobs Companies

Vị trí tuyển dụng, tên công ty

Tất cả tỉnh/thành phố

Tìm kiếm

UX/UI Designer

Mức lương

1200-2500 triệu

Địa điểm

Việt Nam, Hà Nội, 2

Hình thức

full_time

Hạn nộp hồ sơ

01-01-2025

Ứng tuyển ngay

Lưu tin 2

Chi tiết tin tuyển dụng

Kinh doanh kênh MT

Direct Sales

Bán lẻ - Hàng tiêu dùng - FMCG

Thực phẩm / Đồ uống

Mô tả công việc

Công ty đang tìm kiếm một chuyên viên thiết kế UX/UI để tham gia vào các dự án phát triển giao diện người dùng, đảm bảo trải nghiệm người dùng mượt mà và dễ sử dụng.

Yêu cầu ứng viên

Thiết kế UX/UI

Figma

Sketch

Adobe XD

Prototyping

Quyền lợi

Chế độ bảo hiểm

Đào tạo thường xuyên

Môi trường làm việc sáng tạo

Địa điểm làm việc

Việt Nam, Hà Nội, 2

Thời gian làm việc

full_time

Ứng tuyển ngay

Lưu tin

Hạn nộp hồ sơ: 01-01-2025

Tech Innovations Ltd.

Quy mô:

Ít hơn 10 nhân viên

Lĩnh vực:

Công nghệ thông tin

Địa điểm:

Việt Nam, Hà Nội, 2

Xem trang công ty

Thông tin chung

Cấp bậc

Nhân viên

Kinh nghiệm

Có kinh nghiệm thiết kế UX/UI, sử dụng thành thạo các công cụ thiết kế như Sketch, Figma, Adobe XD.

Số lượng tuyển

2

Hình thức làm việc

full_time

Công việc cùng công ty

Software Developer

1000-2000

Việt Nam, Hà Nội, 2

Data Scientist

1500-3000

Việt Nam, Hà Nội, 2

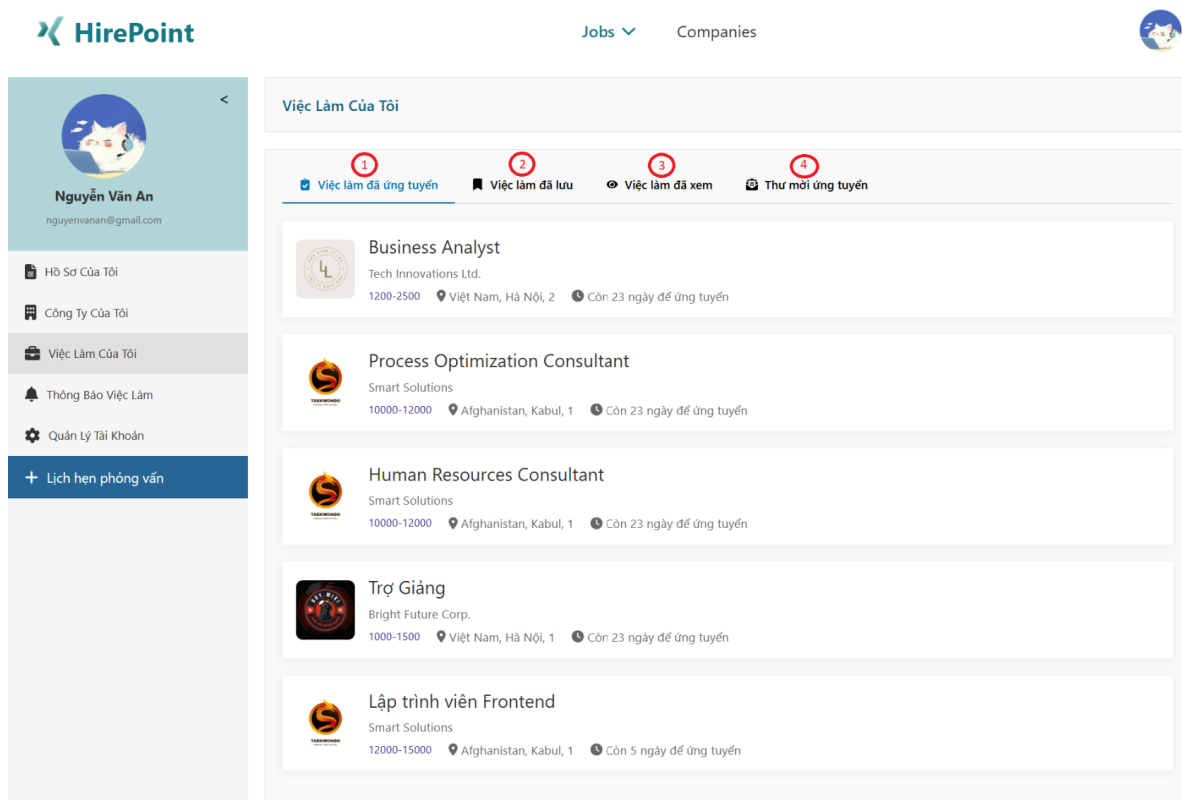
Project Manager

Hình 30. Giao diện chi tiết công việc

Bảng 23. Mô tả ý nghĩa các thành phần giao diện chi tiết công việc

STT	Loại thẻ	Ý nghĩa
1	Button	Nhấn vào để thực hiện quy trình ứng tuyển.
2	Button	Nhấn vào để lưu công việc.

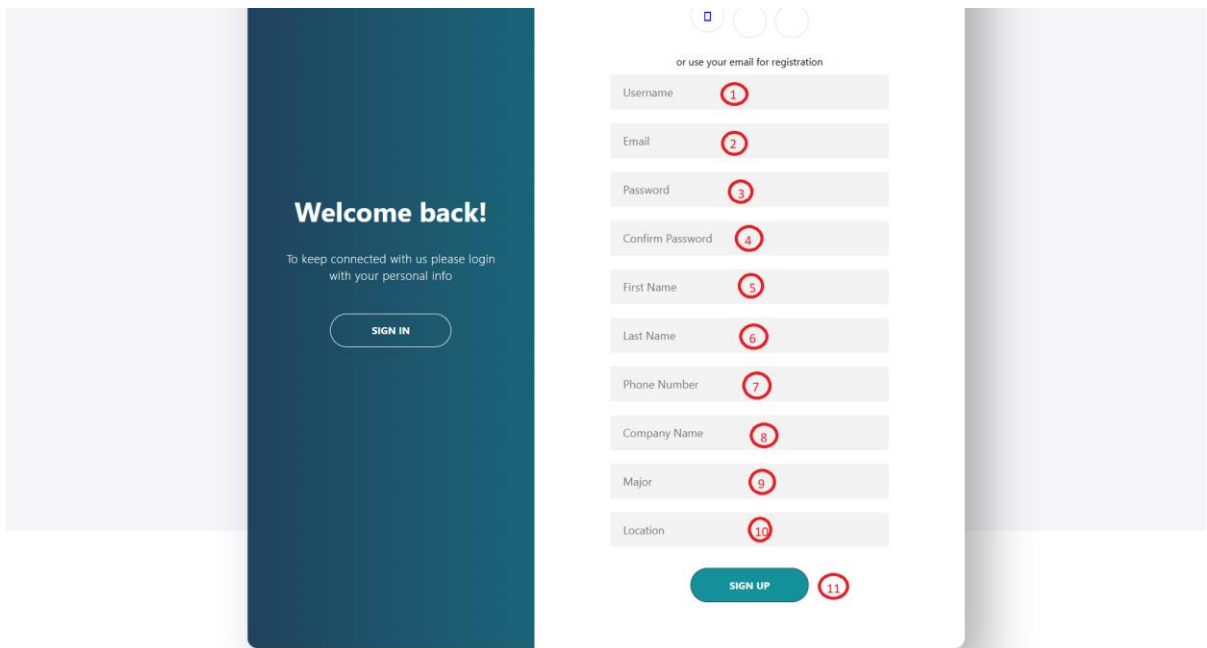
56



Hình 31. Giao diện quản lý công việc (Ứng viên)

Bảng 24. Mô tả ý nghĩa các thành phần giao diện quản lý công việc (Ứng viên)

STT	Loại thẻ	Ý nghĩa
1	Label	Nhấn vào để xem việc làm đã ứng tuyển.
2	Label	Nhấn vào để xem việc làm đã lưu.
3	Label	Nhấn vào để xem việc làm đã xem
4	Label	Nhấn vào để xem thông tin thư mời ứng tuyển



Hình 32. Giao diện đăng ký nhà tuyển dụng

Bảng 25. Mô tả ý nghĩa các thành phần giao diện đăng ký nhà tuyển dụng

STT	Loại thẻ	Ý nghĩa
1	Input	Ô để nhập username.
2	Input	Ô để nhập email.
3	Input	Ô để nhập password
4	Input	Ô để nhập lại password
5	Input	Ô để nhập First Name
6	Input	Ô để nhập Last Name
7	Input	Ô để nhập số điện thoại
8	Input	Ô để nhập tên công ty
9	Input	Ô để nhập lĩnh vực
10	Input	Ô để nhập địa chỉ
11	Button	Nhấn để tiến hành đăng ký

Hoàng Mai Lan

hoangmailan@gmail.com

Hồ Sơ Công ty

Tuyển dụng

Tìm kiếm ứng viên

Thông Báo

Quản Lý Tài Khoản

Quản Lý Bài Test

Tạo Bài Test

Thông Tin Công Ty

Thông tin chung

Theo dõi công ty

Tên công ty *

Smart Solutions

Lĩnh vực *

Dịch vụ tư vấn

Quy mô *

25-99

Địa chỉ *

Afghanistan, Kabul, 1

Địa chỉ cụ thể *

Tầng 3, Tòa nhà Tháp Giải Phóng, Đường Lý Thường Kiệt

Điện thoại *

+84 0912785634

Website *

https://smartsolutions.vn

Sơ lược về công ty

Smart Solutions là công ty chuyên cung cấp dịch vụ tư vấn cho các doanh nghiệp trong các lĩnh vực quản lý, chiến lược kinh doanh, và tối ưu hóa quy trình. Chúng tôi giúp khách hàng phát triển các giải pháp sáng tạo và hiệu quả nhằm đạt được mục tiêu kinh doanh lâu dài. Với đội ngũ chuyên gia giàu kinh nghiệm, Smart Solutions cam kết mang lại sự hỗ trợ chiến lược vững chắc, giúp các công ty tăng trưởng và phát triển bền vững.

Logo công ty

TAEKWONDO

(Tập tin với phần mở rộng .jpg, .jpeg, .png, .gif và kích thước <1MB)

Banner công ty

Explore the WORLD

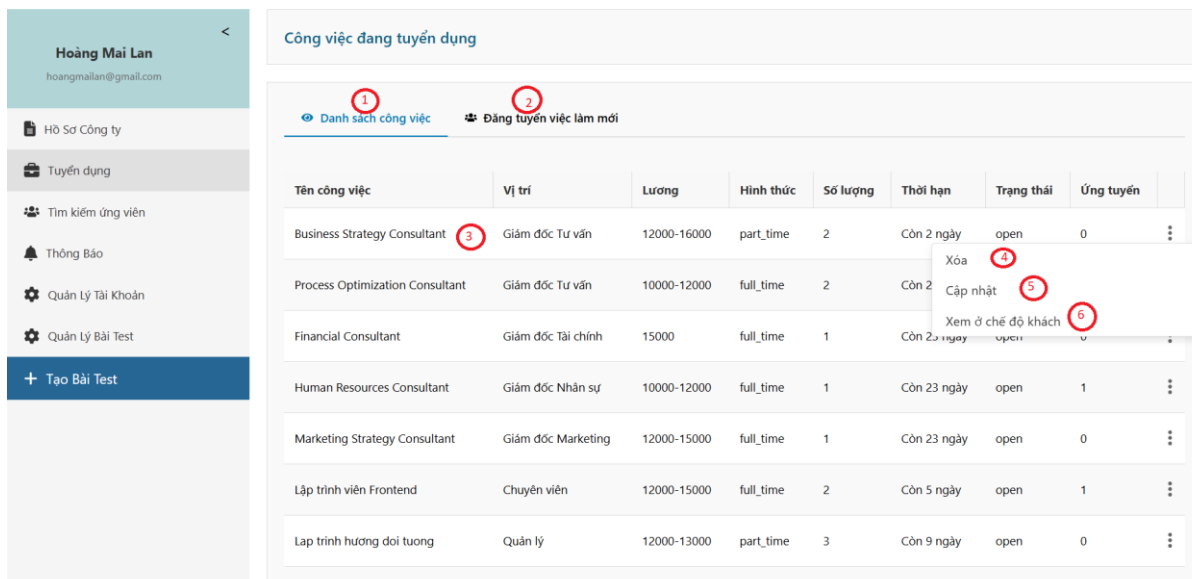
(Tập tin với phần mở rộng .jpg, .jpeg, .png, .gif và kích thước <1MB)

Lưu

Hình 33. Giao diện các thanh điều hướng (Nhà tuyển dụng)

Bảng 26. Mô tả ý nghĩa các thành phần giao diện các thanh điều hướng (Nhà tuyển dụng)

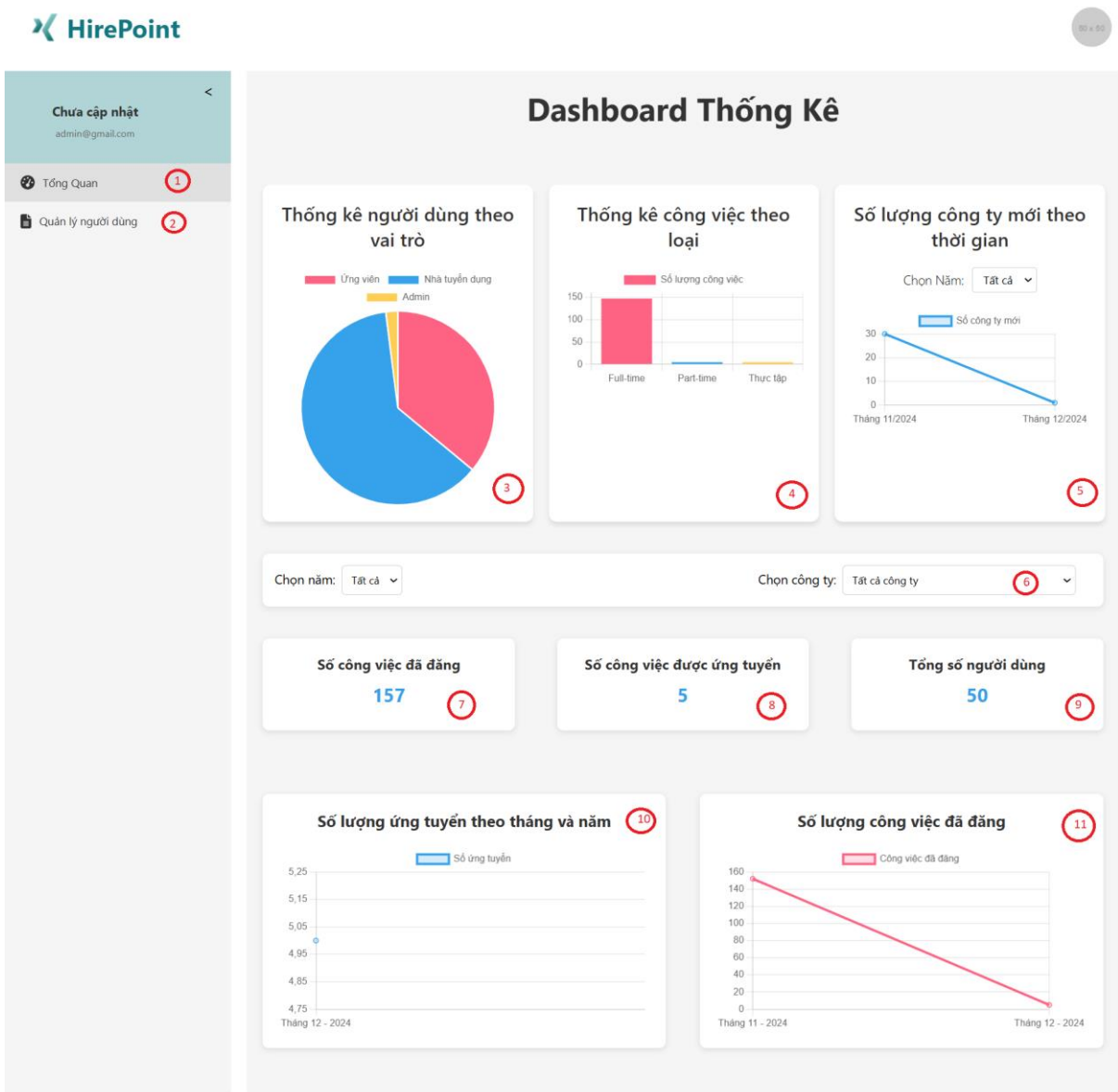
STT	Loại thẻ	Ý nghĩa
1	Label	Nhấn vào để xem hồ sơ công ty và ứng viên đang theo dõi công ty.
2	Label	Nhấn vào để xem danh sách công việc và đăng tin tuyển dụng mới.
3	Label	Nhấn vào để thực hiện lọc tìm kiếm ứng viên.
4	Label	Nhấn vào để xem thông báo người dùng.
5	Label	Nhấn vào để tới trang quản lý tài khoản
6	Label	Nhấn vào để tới trang quản lý các bài kiểm tra.
7	Label	Nhấn vào để thực hiện tạo một bài kiểm tra mới.
8	Button	Bấm để thực hiện lưu thông tin công ty sau khi thay đổi.



Hình 34. Giao diện quản lý công việc (Nhà tuyển dụng)

Bảng 27. Mô tả ý nghĩa các thành phần giao diện quản lý công việc (Nhà tuyển dụng)

STT	Loại thẻ	Ý nghĩa
1	Label	Nhấn vào để xem danh sách các công việc.
2	Label	Nhấn vào để hiển thị trang thêm công việc mới.
3	Label	Nhấn vào để xem danh sách ứng viên ứng tuyển vào công việc đó.
4	Button	Nhấn vào để thực hiện xóa công việc
5	Button	Nhấn vào để có thể cập nhật công việc
6	Button	Nhấn vào để xem chi tiết công việc ở chế độ khách.



Hình 35. Giao diện thống kê (Admin)

Bảng 28. Mô tả ý nghĩa các thành phần giao diện thống kê (Admin)

STT	Loại thẻ	Ý nghĩa
1	Label	Nhấn vào để xem thống kê tổng quan toàn bộ hệ thống.
2	Label	Nhấn vào để xem danh sách người dùng.
3	Card	Hiển thị thống kê người dùng theo vai trò.
4	Card	Hiển thị thống kê công việc theo loại.
5	Card	Hiển thị thống kê số lượng công việc mới.
6	Selectbox	Chọn công ty muốn thống kê.

7	Card	Hiển thị số lượng tin đăng tuyển.
8	Card	Hiển thị số lượng người ứng tuyển.
9	Card	Hiển thị tổng số người dùng.
10	Card	Hiển thị thống kê số lượng ứng tuyển theo năm tháng.
11	Card	Hiển thị số lượng công việc đã đăng.

CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ

4.1. CÁC CÔNG CỤ SỬ DỤNG

Bảng 29. Các công cụ sử dụng xây dựng website

Tool	Name
Environment	Windows
IDE	VSCode
Database	MongoDB
Source Control	Github

4.2. CÁC CÔNG NGHỆ SỬ DỤNG

Bảng 30. Các công nghệ sử dụng xây dựng website

Backend	Node.js, Express.js
Frontend	ReactJS, Cloudinary
Database	MongoDB

4.3. CÀI ĐẶT ỨNG DỤNG

4.3.1. Yêu cầu môi trường

Để chạy project trên máy tính cá nhân, cần chuẩn bị môi trường như sau:

- Cài đặt npm: Đảm bảo npm đã được cài đặt trên hệ thống. Bạn có thể tải Node.js (npm đi kèm theo Node.js).
- Cài đặt NodeJS: Node.js phải được cài đặt để chạy server.
- Cài đặt ReactJS: Thư viện ReactJS sẽ được cài đặt thông qua npm.
- Công cụ dòng lệnh: Windows Terminal hoặc Command Prompt.
- Visual Studio Code (VS Code): Công cụ chỉnh sửa mã nguồn.

4.3.2. Các bước thực hiện

Bước 1: Clone hoặc download project

Truy cập vào link Drive:

Hoặc truy cập link Github:

- Tải xuống toàn bộ thư mục project hoặc clone về nếu có repository.

Bước 2: Khởi chạy back-end

- Mở Command Prompt hoặc Windows Terminal.
- Khởi chạy backend bằng lệnh: ***node backend/server.js***

Bước 3: Khởi chạy front-end

- Mở Command Prompt hoặc Windows Terminal.
- Di chuyển vào thư mục chứa mã nguồn front-end bằng lệnh: ***cd frontend***
- Thực hiện chạy frontend bằng lệnh: ***npm start***

4.4. KIỂM THỬ

Bảng 31. Kết quả kiểm thử

STT	Mô tả test case	Ngày thực hiện	Kết quả	Kết quả	Chú thích
1	Kiểm thử chức năng đăng nhập	01/12/2024	Như mong đợi	Pass	Không có
2	Kiểm thử chức năng đăng ký ứng viên	01/12/2024	Như mong đợi	Pass	Không có
3	Kiểm thử chức năng đăng ký nhà tuyển	01/12/2024	Như mong đợi	Pass	Không có
4	Kiểm thử chức năng xem chi tiết công việc, công ty	01/12/2024	Như mong đợi	Pass	Không có
5	Kiểm thử chức năng ứng tuyển	01/12/2024	Như mong đợi	Pass	Không có
6	Kiểm tra chức năng lưu công việc, theo dõi công ty	01/12/2024	Như mong đợi	Pass	Không có
7	Kiểm tra chức năng chỉnh sửa hồ sơ cá nhân	01/12/2024	Như mong đợi	Pass	Không có
8	Kiểm tra chức năng thông báo	01/12/2024	Như mong đợi	Pass	Không có
9	Kiểm tra chức năng quản lý công việc của ứng viên	01/12/2024	Như mong đợi	Pass	Không có
10	Kiểm tra chức năng xác nhận lịch hẹn	01/12/2024	Như mong đợi	Pass	Không có
11	Kiểm tra chức năng quản lý công ty	01/12/2024	Như mong đợi	Pass	Không có
12	Kiểm tra chức năng quản lý công việc	01/12/2024	Như mong đợi	Pass	Không có
13	Kiểm tra chức năng tạo lịch hẹn	01/12/2024	Như mong đợi	Pass	Không có

14	Kiểm tra chức năng quản lý bài kiểm tra	01/12/2024	Như mong đợi	Pass	Không có
15	Kiểm tra chức năng vô hiệu hóa tài khoản	01/12/2024	Như mong đợi	Pass	Không có
16	Kiểm tra chức năng thống kê	01/12/2024	Như mong đợi	Pass	Không có

PHẦN 3: KẾT LUẬN

1. KẾT QUẢ ĐẠT ĐƯỢC:

Sau một thời gian nghiên cứu và phát triển dự án nhóm chúng em đã đạt được các kết quả khả quan như sau:

- Xây dựng thành công một hệ thống website hoàn chỉnh, tích hợp đầy đủ các chức năng cơ bản như quản lý, tìm kiếm, và phân loại dữ liệu.
- Áp dụng hiệu quả công nghệ MERN Stack (MongoDB, ExpressJS, ReactJS, NodeJS), đảm bảo tính linh hoạt, bảo mật cao và hiệu suất tốt. Việc sử dụng JWT để xác thực người dùng và Cloudinary để quản lý file đã góp phần nâng cao chất lượng hệ thống.
- Học được cách áp dụng công nghệ MERN Stack một cách toàn diện, từ xây dựng giao diện bằng ReactJS đến phát triển API với Node.js và ExpressJS, đồng thời nắm vững kiến thức về Restful API, JSON Web Token (JWT) để bảo mật hệ thống, và sử dụng Cloudinary để quản lý file hiệu quả.
- Nhóm đã tích lũy được nhiều kinh nghiệm quý giá trong việc lập kế hoạch, quản lý dự án và xử lý các tình huống phát sinh. Việc lựa chọn công nghệ phù hợp và phân công nhiệm vụ hợp lý đã giúp tăng cường hiệu quả làm việc nhóm. Các khó khăn trong quá trình phát triển, như xử lý lỗi và tối ưu hiệu năng, đã trở thành bài học quan trọng, giúp nhóm nâng cao khả năng giải quyết vấn đề.

2. ƯU ĐIỂM:

- Ứng dụng công nghệ hiện đại: Hệ thống được phát triển trên nền tảng MERN Stack (MongoDB, Express, ReactJS, Node.js), một công nghệ tiên tiến, linh hoạt và dễ mở rộng.
- Tính bảo mật cao: Việc sử dụng JSON Web Token (JWT) giúp đảm bảo tính an toàn, bảo mật trong quá trình xác thực và quản lý người dùng.
- Khả năng lưu trữ linh hoạt: Tích hợp Cloudinary cho phép lưu trữ và quản lý tệp tin một cách hiệu quả, giảm tải hệ thống cục bộ.
- Tính thân thiện với người dùng: Giao diện website được phát triển bằng ReactJS, đảm bảo trải nghiệm mượt mà và dễ sử dụng.
- Cấu trúc API chuẩn Restful: API tuân thủ tiêu chuẩn Restful, dễ dàng mở rộng và duy trì.

3. NHƯỢC ĐIỂM:

- Hệ thống hiện tại có thể gặp khó khăn khi xử lý lượng dữ liệu lớn hoặc người dùng đồng thời.
- Hệ thống chưa hỗ trợ mạnh mẽ cho các tác vụ quản lý nội dung phức tạp của Admin.

4. ĐỊNH HƯỚNG PHÁT TRIỂN:

Để nâng cao hiệu quả và mở rộng tính năng, hệ thống cần:

- Xây dựng thêm các chức năng quản lý (đánh giá công việc, bật/tắt tìm kiếm ứng viên, liên kết nhà tuyển dụng).
- Áp dụng AI để đề xuất tự động tạo hồ sơ và chấm điểm ứng viên dựa trên các yếu tố như kỹ năng, kinh nghiệm và sự phù hợp với công việc, đồng thời sử dụng AI để gợi ý việc làm thông minh, phân tích hành vi người dùng và cải thiện đề xuất theo thời gian; bên cạnh đó, một chatbot AI sẽ được phát triển để hỗ trợ người dùng giải đáp thắc mắc và hỗ trợ trong quá trình đăng ký, tìm kiếm việc làm và ứng tuyển.
- Tối ưu hiệu suất hệ thống bằng cách cải thiện cấu trúc cơ sở dữ liệu, sử dụng cache hoặc các công nghệ tăng tốc khác.
- Hoàn thiện khả năng quản trị nội dung của Admin để dễ dàng quản lý và vận hành hệ thống.

DANH MỤC TÀI LIỆU THAM KHẢO

1. Ethan Brown, *Web Development with Node and ExpressJS*, 2nd Edition, <https://www.oreilly.com/library/view/web-development-with/9781492053507/>, truy cập lần đầu lúc 20/09/2024.