A white rectangular frame with black border

AI-generated content may be incorrect.A white rectangular frame with black border

AI-generated content may be incorrect.

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

**KHOA ĐA PHƯƠNG TIỆN**

**A red circle with a yellow star in the middle

AI-generated content may be incorrect.**

**BÁO CÁO**

**THỰC TẬP TỐT NGHIỆP**

**ĐỀ TÀI: HỆ THỐNG ĐỐI SOÁT VÀ GIAO VIỆC**

Đơn vị thực tập : **Công ty Cổ phần YOTTATECK**

Giảng viên hướng dẫn : **Nguyễn Thị Thanh Tâm**

Lớp : **D21PTDPT**

Sinh viên: : **Hoàng Văn Hùng**

Mã sv: : **B21DCPT123**

***Hà Nội - 2025***

A close-up of a document

AI-generated content may be incorrect.

|  |  |
| --- | --- |
| HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  KHOA ĐA PHƯƠNG TIỆN | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA  VIỆT NAM  Độc lập - Tự do - Hanh phúc |
|  |  |

**ĐÁNH GIÁ KẾT QUẢ THỰC TẬP TỐT NGHIỆP**

(Thời gian thực tập: Từ ngày 16/06/2025 đến ngày 16/09/2025 )

Họ và tên sinh viên : Hoàng Văn Hùng

Mã sinh viên : B21DCPT123

Lớp : D21PTDPT

**NỘI DUNG ĐÁNH GIÁ**

1. **Chấp hành kỷ luật: (Tốt, Trung bình, hoặc Yếu)**
2. **Ý thức học tập: (Tốt, Trung bình, hoặc Yếu)**
3. **Quan hệ, giao tiếp: (Tốt, Trung bình, hoặc Yếu)**
4. **Điểm (Thang điểm 10)**

**Các ý kiến khác** (nếu có):

|  |  |
| --- | --- |
|  | Ngày tháng năm 20….  Giáo viên hướng dẫn thực tập  *(Ký và ghi rõ họ tên)* |
|  |  |

# **LỜI CẢM ƠN**

Thực tập tốt nghiệp là một yêu cầu bắt buộc đối với sinh viên Học viện Công nghệ Bưu chính Viễn thông trước khi kết thúc thời gian học tại trường. Một mặt là yêu cầu, nhưng mặt khác đây cũng là một giai đoạn hết sức ý nghĩa, giúp sinh viên tập làm quen với công việc thực tế.

Để cho chúng em có thể nắm chắc kiến thức và tiếp cận với thực tế nhà trường đã tạo điều kiện cho chúng em thực tập và làm đồ án tốt nghiệp. Trong thời gian thực tập, em đã nhận được sự giúp đỡ tận tình của các giảng viên khoa Đa phương tiện, đặc biệt là giảng viên hướng dẫn: cô Nguyễn Thị Thanh Tâm. Em xin gửi lời cảm ơn chân thành sâu sắc đến các anh chị, các bạn ở Công ty Cổ phần Yottateck đã tạo cơ hội cho chúng em được tiếp xúc thực tế với ngành nghề và trải nghiệm những khó khăn cũng như thử thách của công việc thực tế.

Đồng thời, chúng em cũng xin gửi lời cảm ơn tới giảng viên khoa Đa phương tiện, GV Nguyễn Thị Thanh Tâm đã hướng dẫn chúng em trong thời gian thực tập.

Em xin chân thành cảm ơn!

**MỤC LỤC**

[LỜI CẢM ƠN 4](#_Toc206418865)

[DANH MỤC HÌNH ẢNH 7](#_Toc206418866)

[DANH MỤC BẢNG BIỂU 8](#_Toc206418867)

[PHẦN A: ĐỀ CƯƠNG THỰC TẬP 9](#_Toc206418868)

[PHẦN B : GIỚI THIỆU ĐƠN VỊ THỰC TẬP 11](#_Toc206418869)

[I. Thông tin chung 11](#_Toc206418870)

[II. Cơ cấu tổ chức 11](#_Toc206418871)

[III. Các lĩnh vực hoạt động 12](#_Toc206418872)

[3.1. Kho dữ liệu doanh nghiệp 12](#_Toc206418873)

[3.2. BI – Báo cáo dữ liệu 12](#_Toc206418874)

[3.3. Hệ thống giám sát và phân tích dữ liệu 13](#_Toc206418875)

[3.4. Dịch vụ tư vấn dữ liệu lớn 13](#_Toc206418876)

[3.5. Ứng dụng dựa trên dữ liệu lớn theo yêu cầu 13](#_Toc206418877)

[3.6. Phân tích khách hàng 13](#_Toc206418878)

[PHẦN C : NỘI DUNG THỰC TẬP 15](#_Toc206418879)

[I. Đào tạo về Java Core 15](#_Toc206418880)

[1.1. Tổng quan Java: JDK, JVM, kiểu dữ liệu, biến, toán tử, cấu trúc chương trình 15](#_Toc206418881)

[1.2. Điều kiện & vòng lặp: if, switch, for, while, do-while 15](#_Toc206418882)

[1.3. OOP – Class/Object: Class, Object, Constructor, Method, Access modifier 15](#_Toc206418883)

[1.4. OOP – Inheritance & Abstraction: Kế thừa, abstract class, interface, override 15](#_Toc206418884)

[1.5. OOP – Polymorphism & Exception Encapsolution: Overloading, Exception (try-catch-finally, throws) 16](#_Toc206418885)

[II. Đào tạo về Java nâng cao 16](#_Toc206418886)

[2.1. Collection Framework: List, Set, Map, Iterator, HashMap vs TreeMap 16](#_Toc206418887)

[2.2. Generics & Lambda: <>, Functional Interface, Lambda expression 16](#_Toc206418888)

[2.3. Stream API: map, filter, collect, sort, group 17](#_Toc206418889)

[2.4. Multithreading: thread, runnable, syschronized, ExcutorService 17](#_Toc206418890)

[2.5. Maven + JUnit: cấu trúc Maven project, Junit test cơ bảm, dependency management 17](#_Toc206418891)

[III. Spring framework – springboot backend cơ bản 17](#_Toc206418892)

[3.1. Spring Intro & IoC: IOC, Bean, Autowiring, cấu trúc Spring Boot project 17](#_Toc206418893)

[3.2. REST API với Spring Boot: Controller, RequestMapping, @Get/Post/Put/DeleteMapping 17](#_Toc206418894)

[3.3. Spring Data JPA: Entity, Repository, CrudRepository, query method 18](#_Toc206418895)

[3.4. Kết nối DB: Cấu hình datasource, MySQL/PostgreSQL, file application.yml/application.properties 18](#_Toc206418896)

[3.5. Validation & Exception: @Valid, @NotNull, GlobalExceptionHandler 18](#_Toc206418897)

[IV. Springboot nâng cao 18](#_Toc206418898)

[4.1. Security cơ bản: @PreAuthorize, basic auth, config SecurityFilterChain 18](#_Toc206418899)

[4.2. Swagger, Logging: Swagger UI, cấu hình logback, log ra file 18](#_Toc206418900)

[4.3. Pagination, sorting: Pageable, PageRequest, Sort 18](#_Toc206418901)

[4.4. Quan hệ Entity & DTO: @OnetoMany, DTO Mapping bằng Mapstruct hoặc thủ công 19](#_Toc206418902)

[4.5. Dự án CRUD thực tế Bắt đầu project: API quản lý người dùng/sản phẩm (CRUD + validation + filter + pagination) 19](#_Toc206418903)

[V. Dự án thực tế 19](#_Toc206418904)

[VI. Dự án cá nhân 21](#_Toc206418905)

[6.1. Giới thiệu đề tài 21](#_Toc206418906)

[6.1.1. Tổng quan 21](#_Toc206418907)

[6.1.2. Bối cảnh và lý do chọn đề tài 21](#_Toc206418908)

[6.1.3. Mục tiêu dự án 21](#_Toc206418909)

[6.1.4. Các thành phần chính của hệ thống 22](#_Toc206418910)

[6.2. Phạm vi và chức năng chính 23](#_Toc206418911)

[6.2.1. Đối tượng người dùng 23](#_Toc206418912)

[6.2.2. Chức năng chính 23](#_Toc206418913)

[6.3. Thiết kế cơ sở dữ liệu 23](#_Toc206418914)

[6.4. Phân tích thiết kế hệ thống 27](#_Toc206418915)

[6.5. Sơ đồ tuần tự cho usecase 36](#_Toc206418916)

[6.6. Thiết kế API document cho bên actor Nhân viên 40](#_Toc206418917)

[6.7. Giao diện cho admin 44](#_Toc206418918)

[VII. Kết luận 47](#_Toc206418919)

[7.1. Kết quả đạt được cho công việc 47](#_Toc206418920)

[7.2. Kết quả đạt được cho cá nhân 48](#_Toc206418921)

# **DANH MỤC HÌNH ẢNH**

*Hình 1. Cơ cấu tổ chức của công ty cổ phần Yottateck*

*Hình 2. Cơ sở dữ liệu của hệ thống*

*Hình 3. Sơ đồ usecase tổng quát dự kiến cho admin*

*Hình 4. Sơ đồ usecase tổng quát dự kiến cho nhân viên*

*Hình 5. Giao diện đăng nhập*

*Hình 6. Giao diện trang chủ*

*Hình 7. Giao diện hợp đồng*

*Hình 8. Giao diện Quản lý thuế suất*

*Hình 9. Giao diện quản lý giao việc*

*Hình 10. Giao diện phê duyệt đối tác*

# **DANH MỤC BẢNG BIỂU**

*Bảng 1: Bảng công việc được giao*

*Bảng 2: Bảng mối quan hệ giữa các thực thể*

*Bảng 3. UC thêm hợp đồng*

*Bảng 4. UC sửa hợp đồng*

*Bảng 5. UC xóa hợp đồng*

*Bảng 6. UC hiển thị danh sách hợp đồng*

*Bảng 7. UC hiển thị Task*

*Bảng 8. UC thêm Task*

*Bảng 9. UC sửa Task*

*Bảng 10. UC xóa Task*

*Bảng 11. UC giao Task*

*Bảng 12. UC phê duyệt đối tác*

*Bảng 13. UC thêm thuế*

*Bảng 14. API login*

*Bảng 15. API logout*

*Bảng 16. API thêm partner*

*Bảng 17. API lấy danh sách hợp đồng cần duyệt*

*Bảng 18. API lấy danh sách công việc cần làm*

# **PHẦN A: ĐỀ CƯƠNG THỰC TẬP**

|  |  |
| --- | --- |
| HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM |
| **KHOA ĐA PHƯƠNG TIỆN** | **Độc lập – Tự do – Hạnh phúc** |
|  |  |
| **KẾ HOẠCH ĐĂNG KÝ THỰC TẬP TỐT NGHIỆP** | |

**Họ và tên sinh viên:** Hoàng Văn Hùng

**Lớp:** D21PTDPT

**Địa chỉ liên hệ:** Mỗ Lao – Hà Đông – Hà Nội

**Điện thoại:** 0329307747 **E-mail:** hunghv.b21pt123@stu.ptit.edu.vn

**Đơn vị thực tập tốt nghiệp:** Công ty cổ phần Yottateck

**Thời gian thực tập:** 16/06/2025 - 16/09/2025

**Người hướng dẫn trực tiếp:** Phạm Đức Việt **Chức vụ:** Giám đốc

**KẾ HOẠCH THỰC TẬP:**

**Nội dung công việc:**

Công việc 1: Làm quen với môi trường và các yêu cầu nơi mình thực tập.

Công việc 2: Đào tạo kiến thức về Java Core.

Công việc 3: Đào tạo kiến thức về Back-End và frameword cụ thể (Spring framework – Springboot).

Công việc 4: Tham gia vào các dự án của công ty.

Công việc 5: Hoàn thiện dự án cá nhân để báo cáo trên khoa và học viện

**Lịch trình thực hiện:**

|  |  |  |
| --- | --- | --- |
| **TT** | **Nội dung thực tập** | **Thời gian** |
| 1 | Làm quen với môi trường làm việc, tìm hiểu về công ty: cơ cấu tổ chức, chức năng, nhiệm vụ, nội quy của công ty. | 16/06 - 18/06/2025 |
| 2 | Đào tạo kiến thức về Java Core | 19/06 - 01/07/2025 |
| 3 | Đào tạo kiến thức về Back-end và framework cụ thể (Spring framework – springboot) | 02/07 -18/07/2025 |
| 4 | Tìm hiểu về sản phẩm của công ty và tham gia vào dự án | 19/07 - 16/09 |
| 5 | Hoàn thiện dự án cá nhân | 28/07 - 10/08 |

*Chú thích:*

*………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………*

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm 2025* |
| **GIÁO VIÊN HƯỚNG DẪN** | **SINH VIÊN** |

# **PHẦN B : GIỚI THIỆU ĐƠN VỊ THỰC TẬP**

1. **Thông tin chung**

Tên công ty: Công ty Cổ phần Yottateck

Địa chỉ: Tầng 31, Tòa nhà Handico, Đường Phạm Hùng, Phường Từ Liêm, TP Hà Nội, Việt Nam

Điện thoại: 0966298666

Email: [contact@yottatek.vn](mailto:contact@melisoft.vn)

Website: https://yottateck.com

Fanpage: [https://www.facebook.com/yottateck.vn](https://www.facebook.com/melisoft.vn)

Linkedin: https://www.linkedin.com/company/yottateck

1. **Cơ cấu tổ chức**

Yottateck là công ty phát triển giải pháp công nghệ phần mềm có trụ sở tại Việt Nam. Các dịch vụ của họ bao gồm phát triển web, nền tảng thương mại điện tử, phát triển ứng dụng di động và thiết kế trải nghiệm người dùng. Yottateck làm việc với các công ty thuộc mọi quy mô, từ những công ty mới thành lập nhỏ đến các doanh nghiệp lớn, để biến ý tưởng của họ thành hiện thực và giúp họ đạt được mục tiêu của mình. Với đội ngũ kỹ sư phần mềm, nhà thiết kế và nhà phát triển kỹ năng cao, Yottateck có chuyên môn và kinh nghiệm để phát triển các ứng dụng di động và dựa trên web được tùy chỉnh hoàn toàn, có thể giúp doanh nghiệp vượt lên trên đối thủ. Bằng cách tận dụng các công nghệ mới nhất, Yottateck có thể tạo ra các giải pháp sáng tạo phù hợp với nhu cầu của từng khách hàng. Cho dù doanh nghiệp của bạn cần một hệ thống cấp doanh nghiệp hay một giải pháp linh hoạt, Yottateck sẽ giúp bạn đạt được kết quả mong muốn. Ngoài việc cung cấp các dịch vụ phần mềm, Yottateck cũng cam kết hỗ trợ khách hàng của họ trong các chiến lược chuyển đổi kỹ thuật số, đưa ra lời khuyên và hướng dẫn để giúp khách hàng của họ duy trì tính cạnh tranh. Họ đã tham gia vào nhiều sáng kiến nhằm thúc đẩy hiểu biết về kỹ thuật số và áp dụng công nghệ, chẳng hạn như tổ chức hội thảo và hội nghị.

A diagram of a company

AI-generated content may be incorrect.

*Hình 1. Cơ cấu tổ chức của công ty cổ phần Yottateck*

1. **Các lĩnh vực hoạt động**
   1. **Kho dữ liệu doanh nghiệp**

Khai thác dữ liệu sâu sắc để cải thiện hiệu suất và lợi nhuận của doanh nghiệp. Yottateck giúp bạn xây dựng kho lưu trữ dữ liệu tài nguyên chính, thực hiện phân tích P&L, dự báo và dự đoán với tất cả các phụ thuộc lẫn nhau, v.v.

* 1. **BI – Báo cáo dữ liệu**

Yottateck phát triển nền tảng hỗ trợ cộng đồng khởi nghiệp tại melihub.com. Melihub, nền tảng huy động vốn, kết nối các nhà đầu tư chuyên nghiệp và những ý tưởng công nghệ tuyệt vời. Ngoài việc tư vấn, họ còn trực tiếp tham gia đầu tư vào các ý tưởng công nghệ dưới dạng nền tảng. Với chương trình đồng hành này, các startup hoàn toàn có thể yên tâm về giải pháp công nghệ. Khách hàng chỉ cần tập trung vào các hoạt động phát triển kinh doanh và đối tác để thúc đẩy doanh thu

* 1. **Hệ thống giám sát và phân tích dữ liệu**

Xử lý và phân tích các luồng dữ liệu lớn theo thời gian thực mà trước đây chưa được khai thác hoặc không thể truy cập để cho phép thông tin chi tiết mới dẫn đến quyết định tốt hơn và nhanh hơn cho doanh nghiệp.

* 1. **Dịch vụ tư vấn dữ liệu lớn**

Yottateck lắng nghe câu chuyện của bạn, kiểm tra các vấn đề cơ bản và hợp tác chặt chẽ với bạn để tìm ra giải pháp tốt nhất giúp củng cố các quyết định kinh doanh chiến lược của bạn và giúp bạn thích ứng với nhu cầu thị trường đang thay đổi.

* 1. **Ứng dụng dựa trên dữ liệu lớn theo yêu cầu**

Yottateck cung cấp các giải pháp Dữ liệu lớn đáp ứng nhu cầu của bạn để giúp bạn đưa ra quyết định tự tin, trao quyền cho nhân viên của bạn với những hiểu biết đáng tin cậy và thúc đẩy văn hóa dựa trên dữ liệu.

* 1. **Phân tích khách hàng**

Nâng cao mức độ hài lòng của khách hàng thông qua phân tích khách hàng 360 độ. Bạn sẽ được phép tối đa hóa nỗ lực tiếp thị của mình trên các nền tảng kỹ thuật số để xây dựng và duy trì mối quan hệ khách hàng bền chặt.

1. **Các quy định chung của công ty**

Cũng giống như các công ty khác, Yottateck cũng có những quy tắc, quy định chung của công ty và các nhân viên cần phải tuân thủ. Quan điểm của Yottateck là 1 nhân viên giỏi không chỉ có chuyên môn giỏi mà còn phải giao tiếp, làm việc nhóm tốt và hơn nữa là phải có trách nhiệm trong công việc. Chính vì vậy mà ngay từ những ngày đầu của quá trình thực tập, em đã được học hỏi những kỹ năng và quy định của công ty

* 1. **Vấn đề bảo mật**

Tất cả những tài liệu, source code của dự án đều phải bảo mật, không được phát tán lên mạng internet. Vì vậy các source code của công ty hay của bất kì dự án nào đều không được công khai

* 1. **Kỷ luật lao động**

Trong môi trường làm việc có rất nhiều kỉ luật mà nhân viên cần chấp hành, tại Yottateck, mỗi nhân viên sẽ phải chấp hành theo quy định của công ty về giờ giấc làm việc, xin nghỉ với lý do chính đáng, quy định vệ sinh,...

* 1. **Thái độ, trách nhiệm trong công việc**

Luôn đảm bảo công việc được giao trong ngày, vào 9h30 hàng ngày sẽ báo cáo tiến độ và công việc của ngày hôm đó. Ngoài ra, phải đưa ra dự định ngày hôm sau sẽ làm gì và dự đoán tiến độ.

Nhiệt tình trong công việc và hòa đồng trong công ty. Làm việc nghiêm túc, chơi hết mình.

# **PHẦN C : NỘI DUNG THỰC TẬP**

1. **Đào tạo về Java Core**

Mục tiêu: Tìm hiểu tổng quan về Java, cú pháp, câu điều liện, vòng lặp, OOP trong Java.

* 1. **Tổng quan Java: JDK, JVM, kiểu dữ liệu, biến, toán tử, cấu trúc chương trình**
* Tìm hiểu các khái niệm liên quan đến jdk, jvm , jre trong java.
* Viết chương trình cơ bản : nhập 2 số nguyên, in ra tổng, hiệu, tích, thương của hai số đó.
* Viết chương trình tính chu vi và diện tích hình tròn.
  1. **Điều kiện & vòng lặp: if, switch, for, while, do-while**
* Tìm hiểu về cách hoạt động của các loại vòng lặp, phân biệt giữa các loại vòng lặp
* Viết chương trình kiểm tra số chẵn/lẻ, chia hết cho 3 hay không.
* In bảng cửu chương từ 2 đến 9 bằng các loại vòng lặp
  1. **OOP – Class/Object: Class, Object, Constructor, Method, Access modifier**
* Tìm hiểu về các khái niệm Class, Object, Constructor, Method trong OOP và cách xây dựng lên chúng
* Tạo class Student (name, age, score) tạo 3 Object và in ra thông tin.
* Viết hàm tính điểm trung bình của danh sách sinh viên.
  1. **OOP – Inheritance & Abstraction: Kế thừa, abstract class, interface, override**
* Tìm hiểu khái niệm về kế thừa và trừu tượng (OOP)
* Viết class Shape (abstactt), 2 lớp con Circle và Rectangle, override phương thức getArea().
* Tạo interface Playable, lớp MusicPlayer implement và in ra "Playing music..."
  1. **OOP – Polymorphism & Exception Encapsolution: Overloading, Exception (try-catch-finally, throws)**
* Tìm hiểu về tính đa hình và đóng gói (OOP)
* Cách ném ra ngoại lệ (Exception)
* Viết chương trình chia 2 số, xử lý ngoại lệ chia cho 0
* Đọc ghi trên file: đọc file text chứa danh sách tên, in ra từng tên một

1. **Đào tạo về Java nâng cao**

Mục tiêu: Tìm hiểu collection, generics, lambda expression, stream api, thread, maven và Unit test

* 1. **Collection Framework: List, Set, Map, Iterator, HashMap vs TreeMap**
* Tìm hiểu về các class, interface trong Collection
* Phân biệt cách hoạt động, cách dung của các collection đó.
* Tạo danh sách List<String> tên người, sắp xếp và in ra theo alphabet
* Dùng Map<String, Integer> đếm số lần xuất hiện của từ trong một đoạn văn bản.
  1. **Generics & Lambda: <>, Functional Interface, Lambda expression**
* Viết Generic method in ra tất cả các phần tử của mảng
* Dùng lambda để lọc ra các số chẵn trong List<Integer>
  1. **Stream API: map, filter, collect, sort, group**
* Dùng Stream lọc danh sách Student có điểm >= 7 và sắp xếp giảm dần theo điểm
* Dùng Collections.groupingBy() nhóm sinh viên theo tuổi
  1. **Multithreading: thread, runnable, syschronized, ExcutorService**
* Tạo 2 thread song song : 1 thread in số chẵn, 1 thread in số lẻ từ 1-20
* Sử dụng ExecutorService để chạy 3 tác vụ song song
  1. **Maven + JUnit: cấu trúc Maven project, Junit test cơ bảm, dependency management**
* Tạo Maven project Hello World
* Viết Junit test kiểm tra hàm tính tổng 2 số.
* Viết hàm tính giai thừa có test case cho n = 0, 1, 5.

1. **Spring framework – springboot backend cơ bản**

Mục tiêu: Hiểu về Restful API, kết nối tới DB, thực hiện CRUD. Hoàn thiện API CRUD

* 1. **Spring Intro & IoC: IOC, Bean, Autowiring, cấu trúc Spring Boot project**
* Tìm hiểu về các khái niệm IoC, Bean
* Tạo project springboot đơn giản, test REST API có endpoint /hello trả về “hello world”
* Hiểu và giải thích về @Component, @Service, @Autowire
  1. **REST API với Spring Boot: Controller, RequestMapping, @Get/Post/Put/DeleteMapping**
* Tạo REST controller /products:
  + GET /products
  + POST /products
  + GET /products/{id}
  1. **Spring Data JPA: Entity, Repository, CrudRepository, query method**
* Tạo Entity Product (id, name, price, category)
* Tạo Repository để lưu và tìm kiếm sản phẩm
  1. **Kết nối DB: Cấu hình datasource, MySQL/PostgreSQL, file application.yml/application.properties**
* Kết nối MySQL/PostgreSQL, viết SQL tạo bảng product.
* Viết API CRUD
  1. **Validation & Exception: @Valid, @NotNull, GlobalExceptionHandler**
* Áp dụng @Valid để validate Product (name không rỗng, price > 0).
* Tạo class GlobalExceptionHandler trả về lỗi chuẩn.

1. **Springboot nâng cao**

Mục tiêu: Tìm hiểu về cơ chế security và phân quyền trong springboot, logging, paging, mapping.

* 1. **Security cơ bản: @PreAuthorize, basic auth, config SecurityFilterChain**
* Cấu hình spring security để yêu cầu đăng nhập auth basic cho mọi API
* Tạo username/password trong memory để test đăng nhập
  1. **Swagger, Logging: Swagger UI, cấu hình logback, log ra file**
* Tích hợp swagger UI hiển thị nội dung cho toàn bộ API REST
* Thêm mô tả cho mỗi endpoint
  1. **Pagination, sorting: Pageable, PageRequest, Sort**
* Viết API hỗ trợ phân trang GET /products?page=0&size=5
* Thêm sort theo tên tăng/giảm
  1. **Quan hệ Entity & DTO: @OnetoMany, DTO Mapping bằng Mapstruct hoặc thủ công**
* Ánh xạ @OneToMany giữa Category và Product
* Tạo DTO ProductDTO, viết hàm convert từ Entity → DTO
  1. **Dự án CRUD thực tế Bắt đầu project: API quản lý người dùng/sản phẩm (CRUD + validation + filter + pagination)**
* Tổng hợp: Làm project CRUD quản lý sản phẩm gồm các tính năng:
  + Thêm/sửa/xóa/tìm
  + Validate + Exception
  + Swagger + Security
  + Phân trang + tìm kiếm theo tên
  + Mapping DTO

1. Dự án thực tế

*Vì tính chất bảo mật của dự án, em xin phép trình bày ngắn gọn về tổng quan và mục đích của dự án như sau:*

**Dự án:** Quản lí và đối soát hợp đồng, giao việc.

**Khách hàng:** Đối tác Viettel

**Phát biểu bài toán:** Hiện nay, các hoạt động quản lý hợp đồng, giao việc và đối soát tại đơn vị đối tác đang thực hiện qua nhiều công cụ và quy trình phức tạp, gây khó khan khi theo dõi tiến độ, kiểm soát thông tin và tổng hợp báo cáo. Việc thiếu một hệ thống tập trung dẫn đến nguy cơ sai sót trong quản lý dữ liệu hợp đồng, biên bản, hóa đơn, thông tin đối tác,… Do đó, cần xây dựng một hệ thống quản lý và đối soát hợp đồng để dễ dàng quản lý và theo dõi.

**Mục tiêu hệ thống:**

* Cho phép người dùng (vai trò admin/trưởng phòng/giám đốc,…) khai báo hợp đồng, quản lí danh sách hợp đồng số.
* Quản lí giao việc cho nhân viên
* Quản lí các loại biên bản hóa đơn
* Quản lí đối soát
* Quản lí thông tin đối tác
* Quản lí thuế suất của hợp đồng
* Quản lí các đề nghị chuyển tiền
* Thiết lập cấu hình xuất báo cáo
* Báo cáo giao việc, tiến độ công việc.
  + Giao diện dễ thao tác và sử dụng

**Các nhiệm vụ được giao trong dự án:**

|  |  |  |
| --- | --- | --- |
| STT | Tên nhiệm vụ | Mô tả / kết quả |
| 1 | Thiết lập cấu hình dự án trong VDI | VDI – Desktop ảo cho lập trình viên.  Cài đặt dự án vào VDI để có thể thực hiện công việc lập trình. |
| 2 | Tìm hiểu về cấu trúc dự án | Dự án được xây bằng framework ZK giúp xây dựng ứng dụng web mà không cần nhiều JS  ZK Được viết bằng Java ở phía server, zk framework có base là java servlet nhưng có thêm các thư mục và file được thù cho zul và cấu hình zk |
| 3 | Thực hiện thử nghiệm và báo cáo lỗi | Dựa vào việc sử dụng ứng dụng và báo cáo lỗi ở phía tester để sửa lỗi và báo cáo lỗi lại cho bên phát triển |
| 4 | Sửa lỗi giao diện | Thực hiện sử lỗi giao diện dựa vào yêu cầu của người hướng dẫn |

*Bảng 1: Bảng công việc được giao*

1. Dự án cá nhân

*Mô tả: dự án cá nhân này được xây dựng dựa trên những kiến thức đã được học và làm việc trong quá trình học và thực tập tại công ty Yottateck. Do vấn đề bảo mật nên không công khai dự án ra ngoài, vì vậy dự án này chỉ mô tả lại một số module của hệ thống*

* 1. **Giới thiệu đề tài**
     1. **Tổng quan**
  + Tên đề tài: Ứng dụng quản lí hợp đồng và giao việc
  + Phần công việc phụ trách: xây dựng ứng dựng cho admin và nhân viên.

- Admin: Sử dụng ZK framework để xây dựng

- Nhân viên: Sử dụng Springboot

* + Triển khai dự án: Deploy trên máy ảo
    1. **Bối cảnh và lý do chọn đề tài**

Để vận dụng và hiểu hết được những kiến thức, kinh nghiệm đã được học và trau dồi trong quá trình Thực tập tại công ty, em đã quyết định xây dựng lại các module chức năng của hệ thống. Các module này có thể được ứng dụng và triển khai trong những ứng dụng hay đề tài thực tế.

* + 1. **Mục tiêu dự án**

Mục tiêu của dự án là phát triển một hệ thống đơn giản, dễ thao tác và sử dụng cho người dùng, hỗ trợ quản lí hợp đồng, giao việc, thuế suất và báo cáo một cách trực quan. Cụ thể:

* + - Người dùng có thể xem thêm sửa xóa việc khai báo hợp đồng, đồng thời giao cho nhân viên duyệt hợp đồng đó.
    - Nhân viên / trưởng phòng có thể xem được các hợp đồng mà mình cần phải duyệt
    - Người dùng tạo danh sách công việc liên quan đến các hợp đồng như: Đóng dấu, tạo hợp đồng, đối soát hợp đồng, …Sau khi tạo xong có thể giao các công việc cho nhân viên
    - Nhân viên có thể xem được các công việc mà mình cần làm.
    - Người dùng quản lý thông tin đối tác và phê duyệt thông tin đối tác
    - Quản lí thuế suất cho các loại hợp đồng và đối tác
    1. **Các thành phần chính của hệ thống**
       1. **Hệ thống admin**

Hệ thống phía admin sử dụng zk framework, nó gió xây dựng giao diện người dùng giàu tương tác. ZK là server-side rendering (SSR), cách hoạt động của zk như sau:

* + - * Client gửi yêu cầu
      * Server ZK xử lý: tạo UI bằng java dựa trên file .zul và logic Java, sau đó
      * Render ra HTML/CSS Trả về client
      * Khi có sự kiện (click, nhập dự liệu): event được xử lý trực tiếp bằng java như lập trình desktop
      1. **Hệ thống phía nhân viên**

Xây dựng hệ thống Backend bằng Java Springboot, xử lý logic nghiệp vụ, cung cấp các endpoints REST API cho phục vụ frontend.

* + - 1. **Cơ sở dữ liệu**

Hệ quản trị cơ sử dữ liệu (Local): Mysql wordbench 8.0

Hệ quản trị cơ sở dữ liệu (Remote): Container Docker

* + - 1. **Hệ thống lưu cache Redis**
      2. **Mô trường triển khai**

Môi trường triển khai local: Docker.

Môi trường triển khai remote: Máy ảo (VM) hệ điều hành Linux (Debian 11 do AWS cung cấp)

* 1. **Phạm vi và chức năng chính**
     1. **Đối tượng người dùng**

Đề tài này hướng đến 2 nhóm người sử dụng:

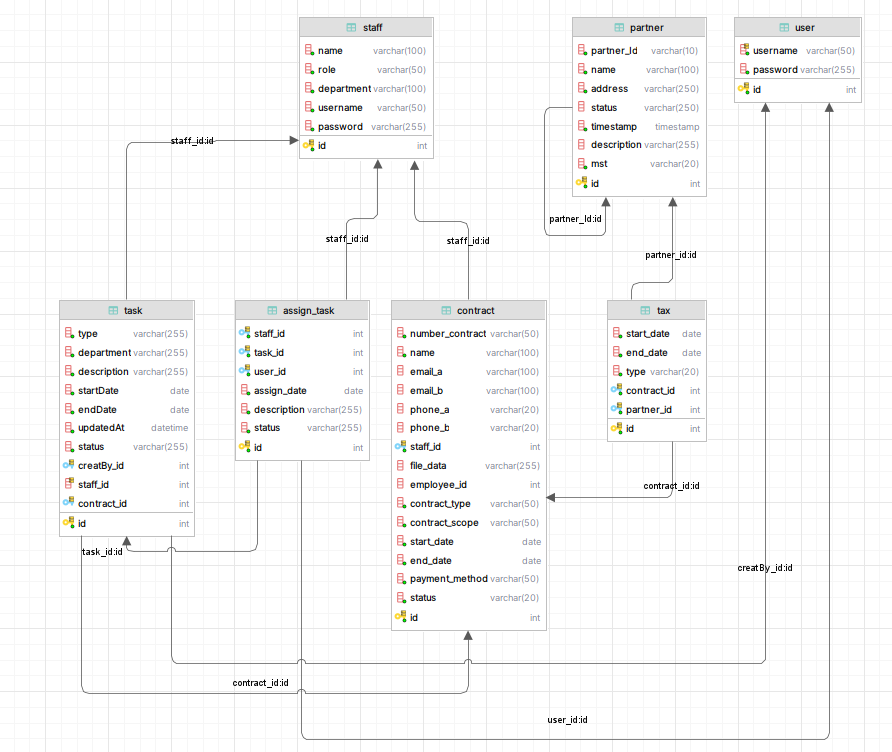
* + - * Quản trị viên (Admin): Là những người có chức cao nhất, chịu trách nhiệm quản lí các thông tin của công ty từ hợp đồng, nhân viên, thông tin đối tác, có quyền giao việc cho nhân viên cấp dưới.
      * Nhân viên cấp dưới: dưới quyền quản lí của admin thực hiện các nhiệm vụ do admin quản lí vào giao việc
    1. **Chức năng chính**

**Chức năng chính bên Admin:**

* + - Chức năng xác thực người dùng
    - Quản lí hợp đồng (xem/thêm/sửa/ xóa) thông tin hợp đồng
    - Quản lí giao việc (thêm nhiệm vụ và giao nhiệm vụ cho nhân viên)
    - Quản lí thông tin đối tác
    - Quản lí thuê suất hợp đồng và đối tác

**Chức năng chính của bên người dùng:**

* + - Xem danh sách hợp đồng cần phải phê duyệt
    - Xem danh sách nhiệm vụ cần phải làm
    - Tạo thông tin đối tác cần phải xác thực
    - Xác thực thông tin đối tác
  1. **Thiết kế cơ sở dữ liệu**



*Hình 2. Cơ sở dữ liệu của hệ thống*

* + 1. Mô tả chi tiết các thực thể

**- User:** Đại diện cho người dùng có vai trò cao nhất tương đương với admin. Có thể giao việc cho nhân viên cấp dưới. Các thuộc tính:

* id: mã định danh
* username: tên đăng nhập
* password: mật khẩu

**- Staff:** Đại diện cho nhân viên cấp dưới, bao gồm các thuộc tính:

* id: mã định danh
* name: tên của nhân viên
* username: tên đăng nhập
* password: Mật khẩu
* role: vai trò
* department: phòng ban

**- Partner:** Đại diện cho đối tác, bao gồm các thuộc tính:

* id: mã định danh
* partner\_Id: mã người dùng
* name: tên doanh nghiệp
* address: địa chỉ doanh nghiệp
* status: trạng thái phê duyệt
* timestamp: Thời gian cập nhật
* description: mô tả
* mst: mã số thuế

**- Task:** Đại diện cho các nhiệm vụ, bao gồm các thuộc tính:

* id: Mã định danh
* type: loại công việc
* department: phòng ban nhân viên
* description: mô tả cho nhiệm vụ
* startDate: ngày bắt đầu nhiệm vụ
* endDate: ngày kết thúc nhiệm vụ
* updateAt: ngày cập nhật
* status: Trạng thái nhiệm vụ
* createBy\_id: mã định danh cho người tạo
* staff\_id: mã định danh cho nhân viên
* contract\_id: mã định cho hợp đồng

**- Contract:** đại diện cho hợp đồng

* id: mã định danh cho hợp đồng
* number\_contract: mã số hợp đồng
* name: tên hợp đồng
* email\_a: email bên A
* email\_b: email bên B
* phone\_a: số điện thoại bên A
* phone\_b: số điện thoại bên B
* staff\_id: mã định danh staff
* file\_data: url của file hợp đồng (Nếu có để upload)
* contract\_type: loại hợp đồng
* contract\_scope: phạm vi hợp đồng
* start\_date: ngày bắt đầu hợp đồng
* end\_date: ngày kết thúc hợp đồng
* payment\_method: phương thức thanh toán
* status: trạng thái hợp đồng

**- Assign\_task:** đại diện cho các công việc được giao cho nhân viên

* id: mã định danh cho assign\_task
* staff\_id: Mã định danh cho staff
* task\_id: mã định danh cho task
* user\_id: mã định doanh cho user
* assign\_date: ngày giao task
* description: mô tả công việc giao
* status: trạng thái công việc được giao

**-Tax:** đại diện cho khai báo thuế

* id: Mã định danh cho tax
* start\_date: ngày bắt đầu tính thuế
* type: loại thuế (Có / không tính thuế)
* contract\_id: mã định danh cho contract

partner\_id: mã định danh cho partner

* + 1. Mối quan hệ giữa các thực thể

|  |  |
| --- | --- |
| Mối quan hệ | Mô tả chi tiết |
| Contract – Staff | Mỗi **Contract** được gán cho một **Staff** (contract.staff\_id → staff.id), quan hệ **N:1** (nhiều hợp đồng thuộc về một nhân viên). |
| Tax – Contract | Mỗi **Tax** thuộc về một **Contract** (tax.contract\_id → contract.id), quan hệ **N:1**. |
| Tax – Partner | Mỗi **Tax** liên kết với một **Partner** (tax.partner\_id → partner.id), quan hệ **N:1** |
| Task – Contract | Mỗi **Task** có thể liên kết với một **Contract** (task.contract\_id → contract.id), quan hệ **N:1**. |
| Task – User | Mỗi **Task** được tạo bởi một **User** (task.creatBy\_id → user.id), quan hệ **N:1** |
| Assign\_Task – Task | Mỗi **Assign\_Task** gán cho một **Task** (assign\_task.task\_id → task.id), quan hệ **N:1**. |
| Assign\_Task – Staff | Mỗi **Assign\_Task** gán cho một **Staff** (assign\_task.staff\_id → staff.id), quan hệ **N:1**. |
| Assign\_Task – User | Mỗi **Assign\_Task** được gán bởi một **User** (assign\_task.user\_id → user.id), quan hệ **N:1**. |

*Bảng 2: Bảng mối quan hệ giữa các thực thể*

* 1. **Phân tích thiết kế hệ thống**

**Các tác nhân tham gia:**

* + - Admin / người có quyền cao nhất
    - Nhân viên: dưới quyền admin

**Sở đồ Usecase tổng quát dự kiến:**

A diagram of a person with text

AI-generated content may be incorrect.

*Hình 3. Sơ đồ usecase tổng quát dự kiến cho admin*

A diagram of a person with text

AI-generated content may be incorrect.

*Hình 4. Sơ đồ usecase tổng quát dự kiến cho nhân viên*

**Scennario cho hệ thống:**

* *UC: Thêm hợp đồng*

|  |  |
| --- | --- |
| **Tên usecase** | Thêm hợp đồng |
| Tác nhân | Admin |
| Mô tả mục tiêu | Thêm thông tin hợp đồng vào database |
| Yêu cầu liên quan | Không |
| Tiền điều kiện | Không |
| Mô tả | 1. Người dùng nhập thông tin hợp đồng vào input tương ứng và ấn nút lưu 2. Hệ thống hiển thị màn hình xác nhận thành công 3. Màn hình hiển thị danh sách hợp đồng |
| Hậu điều kiện | Người dùng xem được thông tin danh sách hợp đồng |
| Biến thể | Không |
| Ngoại lệ | Không |

*Bảng 3. UC thêm hợp đồng*

* *UC: Sửa hợp đồng*

|  |  |
| --- | --- |
| **Tên usecase** | **Sửa hợp đồng** |
| Tác nhân | Admin |
| Mô tả mục tiêu | Sửa thông tin hợp đồng trong database |
| Yêu cầu liên quan | Không |
| Tiền điều kiện | Không |
| Mô tả | 1. Người dùng nhập thông tin hợp đồng vào input tương ứng và ấn nút lưu 2. Hệ thống hiển thị màn hình xác nhận có sửa hay không 3. Người dùng xác nhận 4. Hệ thống lưu thông tin thành công |
| Hậu điều kiện | Thông tin được cập nhật thành công |
| Biến thể | Không |
| Ngoại lệ | Không |

*Bảng 4. UC sửa hợp đồng*

* *UC: Xóa hợp đồng*

|  |  |
| --- | --- |
| **Tên usecase** | **Xóa hợp đồng** |
| Tác nhân | Admin |
| Mô tả mục tiêu | Sửa thông tin hợp đồng trong database |
| Yêu cầu liên quan | Không |
| Tiền điều kiện | Không |
| Mô tả | 1. Người dùng click vào button xóa ở danh sách hợp đồng 2. Hệ thống hiển thị thông báo xóa 3. Người dùng xác nhận xóa 4. Hệ thống xóa thành công thông tin trong DB Và hiển thị danh sách hợp đồng sau xóa. |
| Hậu điều kiện | Thông tin được xóa thành công |
| Biến thể | Không |
| Ngoại lệ | Không |
|  |  |

*Bảng 5. UC xóa hợp đồng*

* *UC: Hiển thị danh sách hợp đồng*

|  |  |
| --- | --- |
| **Tên usecase** | **Hiển thị danh sách hợp đồng** |
| Tác nhân | Admin |
| Mô tả mục tiêu | Hiển thị danh sách hợp đồng trong database |
| Yêu cầu liên quan | Không |
| Tiền điều kiện | Không |
| Mô tả | 1. Người dùng click vào menu hợp đồng 2. Hiển thị danh sách hợp đồng có trong db |
| Hậu điều kiện | Thông tin được hiển thị thành công |
| Biến thể | Không |
| Ngoại lệ | Không |

*Bảng 6. UC hiển thị danh sách hợp đồng*

* *UC: Hiển thị task*

|  |  |
| --- | --- |
| **Tên usecase** | **Hiển thị danh sách task** |
| Tác nhân | Admin |
| Mô tả mục tiêu | Hiển thị danh sách task trong database |
| Yêu cầu liên quan | Không |
| Tiền điều kiện | Không |
| Mô tả | 1. Người dùng click vào menu giao việc 2. Hiển thị danh sách task có trong db |
| Hậu điều kiện | Thông tin được hiển thị thành công |
| Biến thể | Không |
| Ngoại lệ | Không |

*Bảng 7. UC hiển thị Task*

* *UC: thêm task*

|  |  |
| --- | --- |
| **Tên usecase** | **Thêm task** |
| Tác nhân | Admin |
| Mô tả mục tiêu | Thêm task trong database |
| Yêu cầu liên quan | Không |
| Tiền điều kiện | Không |
| Mô tả | 1. Sau khi hiển thị được danh sách task, người dùng nhập thông tin task trong input và nhấn nút lưu 2. Hệ thống gửi thông báo lưu thành công |
| Hậu điều kiện | Thông tin được lưu thành công |
| Biến thể | Không |
| Ngoại lệ | Không |

*Bảng 8. UC thêm Task*

* *UC: sửa task*

|  |  |
| --- | --- |
| **Tên usecase** | **Sửa task** |
| Tác nhân | Admin |
| Mô tả mục tiêu | Sửa task trong database |
| Yêu cầu liên quan | Không |
| Tiền điều kiện | Không |
| Mô tả | 1. Người dùng chọn 1 task trong list task 2. Hệ thống tự động fill task 3. Người dùng điền thông tin cần sửa và ấn nút lưu 4. Hệ thống hiển thị lưu thành công |
| Hậu điều kiện | Thông tin được lưu thành công |
| Biến thể | Không |
| Ngoại lệ | Không |

*Bảng 9. UC sửa Task*

* *UC: xóa task*

|  |  |
| --- | --- |
| **Tên usecase** | **Sửa task** |
| Tác nhân | Admin |
| Mô tả mục tiêu | Xóa task trong database |
| Yêu cầu liên quan | Không |
| Tiền điều kiện | Không |
| Mô tả | 1. Người dùng chọn 1 task trong list task 2. Hệ thống tự động fill task 3. Người dùng ấn nút xóa 4. Hệ thống hiển thị xác nhận xóa 5. Người dùng ấn nút xác nhận |
| Hậu điều kiện | Thông tin được xóa thành công |
| Biến thể | Không |
| Ngoại lệ | Không |

*Bảng 10. UC xóa Task*

* *UC; giao task*

|  |  |
| --- | --- |
| **Tên usecase** | **Giao task** |
| Tác nhân | Admin |
| Mô tả mục tiêu | Giao task cho nhân viên và lưu DB |
| Yêu cầu liên quan | Không |
| Tiền điều kiện | Không |
| Mô tả | 1. Người dùng chọn mã công việc và nhập thông tin cần thiết 2. Hệ thống tự động fill các thông tin còn lại 3. Người dùng ấn nút lưu |
| Hậu điều kiện | Thông tin được xóa thành công |
| Biến thể | Không |
| Ngoại lệ | Không |

*Bảng 11. UC giao Task*

* *UC: phê duyệt đối tác*

|  |  |
| --- | --- |
| **Tên usecase** | **Phê duyệt đối tác** |
| Tác nhân | Admin |
| Mô tả mục tiêu | Cập nhật trạng thái phê duyệt ở DB |
| Yêu cầu liên quan | Không |
| Tiền điều kiện | Không |
| Mô tả | 1. Người dùng chọn đối tác trong danh sách đối tác và ấn phê duyệt 2. Hệ thống cập nhật phê duyệt |
| Hậu điều kiện | Thông tin được cập nhật thành công |
| Biến thể | Không |
| Ngoại lệ | Không |

*Bảng 12. UC phê duyệt đối tác*

* *UC: thêm thuế*

|  |  |
| --- | --- |
| **Tên usecase** | **Thêm thuế** |
| Tác nhân | Admin |
| Mô tả mục tiêu | Thêm thuế vào trong DB |
| Yêu cầu liên quan | Không |
| Tiền điều kiện | Không |
| Mô tả | 1. Người dùng chọn menu thuế suất, điền thông tin liên quan trong input sau đó ấn nút lưu 2. Hệ thống lưu thành công thông tin thuế |
| Hậu điều kiện | Thông tin lưu thành công |
| Biến thể | Không |
| Ngoại lệ | Không |

*Bảng 13. UC thêm thuế*

* 1. **Sơ đồ tuần tự cho usecase**

A diagram of a contract

AI-generated content may be incorrect.

A diagram of a project

AI-generated content may be incorrect.

A diagram of a software project

AI-generated content may be incorrect.

A diagram of a contract

AI-generated content may be incorrect.

A diagram of a task

AI-generated content may be incorrect.

A diagram of a task

AI-generated content may be incorrect.

A diagram of a task

AI-generated content may be incorrect.

A diagram of a task

AI-generated content may be incorrect.

A diagram of a company

AI-generated content may be incorrect.

A diagram of a tax

AI-generated content may be incorrect.

* 1. **Thiết kế API document cho bên actor Nhân viên**
     1. **API login**

|  |  |
| --- | --- |
| Name | Login |
| Description | Đăng nhập vào hệ thống |
| URL | http://47.129.227.197:8081/auth/login |
| Method | POST |
| Params Query | "username": "test2",  "password": "15122003" |
| Response | {    "code": 200,    "message": "Success",    "result": {      "accessToken": "",      "refreshToken": ""    }  } |

*Bảng 14. API login*

* + 1. **API logout**

|  |  |
| --- | --- |
| Name | Logout |
| Description | Đăng xuất khỏi hệ thống |
| URL | http://47.129.227.197:8081/auth/logout |
| Header | **'Authorization: Bearer Token’** |
| Method | POST |
| Params Query |  |
| Response | {  "code": 200,  "message": "Log-out successfully"  } |

*Bảng 15. API logout*

* + 1. **API thêm partner**

|  |  |
| --- | --- |
| Name | Partner |
| Description | Thêm partner |
| URL | 'http://47.129.227.197:8081/auth/logout' |
| Header | 'Authorization: Bearer Token’ |
| Method | POST |
| Params Query | partner\_id=ab  mst=ab  name=ab  address=ab  status=ACCEPT  description=ab |
| Response | {  "code": 200,  "message": "Created partner successfully"  } |

*Bảng 16. API thêm partner*

* + 1. **API lấy danh sách hợp đồng cần duyệt**

|  |  |
| --- | --- |
| Name | Contract |
| Description | Lấy danh sách hợp đồng cần duyệt |
| URL | http://47.129.227.197:8081/staff/contract |
| Header | 'Authorization: Bearer Token’ |
| Method | GET |
| Params Query |  |
| Response | {  "code": 200,  "message": "Get list contract to accept successfully",  "result": [  {  "id": 10,  "number\_contract": "HD0010",  "name": "Hop dong kiem thu",  "email\_a": "a@gmail.com",  "email\_b": "b@gmail.com",  "phone\_a": "0123456789",  "phone\_b": "0123456789",  "staff\_id": 2,  "file\_data": null,  "employee\_id": null,  "contract\_type": "Hợp đồng mua bán",  "contract\_scope": "Nội bộ",  "start\_date": "2025-08-13",  "end\_date": "2025-08-14",  "payment\_method": "Chuyển khoản",  "status": "PENDING"  }  ]  } |

*Bảng 17. API lấy danh sách hợp đồng cần duyệt*

* + 1. **API lấy danh sách công việc cần làm**

|  |  |
| --- | --- |
| Name | Task |
| Description | Lấy danh sách công việc cần làm |
| URL | http://47.129.227.197:8081/task |
| Header | 'Authorization: Bearer Token’ |
| Method | GET |
| Params Query |  |
| Response | {  "code": 200,  "message": "Get all assigned task successfully!",  "result": [  {  "id": 11,  "staff\_id": 2,  "task\_id": 14,  "user\_id": 1,  "assign\_date": "2025-08-13",  "description": "Description",  "status": "DONE"  }  ] |

*Bảng 18. API lấy danh sách công việc cần làm*

* 1. **Giao diện cho admin**

A screenshot of a login form

AI-generated content may be incorrect.

*Hình 5. Giao diện đăng nhập*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 6. Giao diện trang chủ*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 7. Giao diện hợp đồng*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 8. Giao diện Quản lý thuế suất*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 9. Giao diện quản lý giao việc*

A screenshot of a computer

AI-generated content may be incorrect.

*Hình 10. Giao diện phê duyệt đối tác*

1. **Kết luận**
   1. **Kết quả đạt được cho công việc**

Sau quá trình thực tập tại công ty, em đã được tham gia vào quy trình đào tạo cho thực tập sinh Java BackEnd và dự án thực tế của công ty. Tuy đóng góp chưa lớn nhưng cũng có một số đóng góp sau:

* Hoàn thiện hơn quy trình đào tạo cho thực sinh tại công ty bằng cách đưa ra các ý kiến hay nhận xét về quy trình đào tạo của người hướng dẫn trong quá trình học để có thể mang lại hiệu quả tốt nhất.
* Tham gia vào dự án và hoàn thành các yêu cầu được giao nhằm đảm bảo tiến độ công việc đúng thời hạn.
* Góp phần hỗ trợ nhóm/đội dự án trong công việc hàng ngày.

Mang lại nguồn nhân sự tiềm năng cho công ty sau kỳ thực tập giúp công ty dễ dàng mở rộng

* 1. **Kết quả đạt được cho cá nhân**

Một số kết quả đạt được cho cá nhân có thể được kể đến như sau:

* Nâng cao kiến thức về Java Core và các frameword phát triển hệ thống BackEnd như Spring framework, ZK framework.
* Được tiếp xúc với dự án thực tế, cấu trúc tổ chứuc của một dự án lớn
* Học được các xây dựng hệ thống
* Hiểu thêm về quy trình đào tạo của công ty.
* Học cách sử dụng các công cụ quản lý công việc: Jira, Confluence, Basevn,..
* Học thêm về kĩ năng DevOps

Ngoài ra, các kĩ năng mềm còn được nâng cao khi tham gia thực tập tại công ty. Khi được tiếp xúc với môi trường làm việc chuyên nghiệp góp phần nâng cao ý thức và tinh thần kỉ luật. Trong suốt 6 tuần tham gia thực tập cùng các anh chị, các bạn trong công ty, em được trải nghiệm môi trường làm việc thực sự với sự tương tác giữa các thành viên trong nhóm. Đây là công việc không hề dễ dàng, đòi hỏi mọi người cần có tinh thần trách nhiệm đối với công việc được giao và sự sáng tạo cần thiết. Sau thời gian làm việc, em đã tích luỹ được nhiều kinh nghiệm cho bản thân thông qua sự phê hướng dẫn, phê bình, khen ngợi của các anh chị trong nhóm. Kỳ thực tập tại Yottateck là một khoảng thời gian vô cùng có ý nghĩa cho quá trình học tập cũng như cho công việc sau này.