**PHẠM TRUNG KIÊN**

**1821051121**

**BÁO CÁO**

**THỰC TẬP DOANH NGHIỆP**

CHUYÊN NGÀNH : KHOA HỌC MÁY TÍNH

Xây dựng trang quản trị cho thế giới di động và viết API xác thực mã OPT cho người dùng

**HÀ NỘI 5/2022**

**BÁO CÁO**

**THỰC TẬP DOANH NGHIỆP**

CHUYÊN NGÀNH : KHOA HỌC MÁY TÍNH

Xây dựng trang quản trị cho thế giới di động và viết API xác thực mã OPT cho người dùng

**Sinh viên thực tập : Phạm Trung Kiên**

**Mã sinh viên : 1821051121**

**Công ty thực tập : CÔNG TY JITS INNOVATION LABS**

**Người phụ trách : Nguyễn Thị Tuyền**

**Giảng viên hướng dẫn: Đặng Văn Nam**

**HÀ NỘI 5/202**

**Mục Lục**

[**MỞ ĐẦU** 6](#_Toc104120909)

[**CHƯƠNG 1 GIỚI THIỆU VỀ CÔNG TY VÀ VỊ TRÍ THỰC TẬP** 7](#_Toc104120910)

[**Giới thiệu công ty** 7](#_Toc104120911)

[**1.1.1 Giới thiệu chung** 7](#_Toc104120912)

[**1.1.2 Các sản phẩm chính** 7](#_Toc104120913)

[**1.1.3 Kinh nghiệm triển khai** 10](#_Toc104120914)

[**1.1.4 Khách hàng và đối tác** 10](#_Toc104120915)

[**Vị trí thực tập** 11](#_Toc104120916)

[**Đề tài thực tập** 11](#_Toc104120917)

[**Lịch làm việc** 11](#_Toc104120918)

[**Bảng kế hoạch thực tập** 11](#_Toc104120919)

[**CHƯƠNG 2 NỘI DUNG THỰC TẬP** 14](#_Toc104120920)

[**Nội dung thực tập** 14](#_Toc104120921)

[**Mô tả công việc** 14](#_Toc104120922)

[**Các công cụ đã tìm hiểu** 21](#_Toc104120923)

[**CHƯƠNG 3 ĐÁNH GIÁ KẾT QUẢ VÀ ĐỊNH HƯỚNG PHÁT TRIỂN** 34](#_Toc104120946)

[**Đánh giá kết quả thực tập** 34](#_Toc104120947)

[**Định hướng phát triển của đề tài** 34](#_Toc104120948)

[**KẾT LUẬN** 35](#_Toc104120949)

# **Danh mục các bảng**

Bảng 1-1: Kế hoạch làm việc trong 10 tuần thực tập tại đơn vị

Bảng 2-1: Mô tả chi tiết về các công việc cần thực hiện trong quá trình thực tập

Bảng 2-2:Một số công cụ đã tìm hiểu và áp dụng trong qua trình thực tập

**MỞ ĐẦU**

Trong suốt quá trình học tập và thực tập tốt nghiệp em luôn được sự quan tâm, hướng dẫn và giúp đỡ tận tình của các thầy, cô giáo trong Khoa Công nghệ thông tin cùng với sự động viên giúp đỡ của bạn bè đồng nghiệp.

Lời đầu tiên em xin được bảy tỏ lòng biết ơn sâu sắc đến Ban giám hiệu Trường Đại học Mỏ - Địa chất , khoa Công nghệ thông tin đã tận tình giúp đỡ cho em suốt thời gian thực tập .

Đặc biệt em xin bảy tỏ lòng biết ơn chân thành sâu sắc tới thầy giáo GV.Ths. Đặng Văn Nam đã trực tiếp giúp đỡ, hướng dẫn em hoàn thành quá trình thực tập này .

Nhân dịp này em xin được bày tỏ lòng biết ơn đến tập thể công ty TNHH JITS INNOVATION LABS và đặc biệt là chị Nguyễn Thị Tuyền -Trưởng phòng phát triển phần mềm của công ty đã giúp đỡ và tạo điều kiên giúp đỡ em trong suốt thời gian thực tập.

Em cũng xin bày tỏ lòng biết ơn sâu sắc đến gia đình, người thân, bạn bè và đồng nghiệp đã giúp đỡ động viên em hoàn thiện quá trình thực tập này.

Em xin trân trọng cảm ơn!

**CHƯƠNG 1****GIỚI THIỆU VỀ CÔNG TY VÀ VỊ TRÍ THỰC TẬP**

**Giới thiệu công ty**

**1.1.1 Giới thiệu chung**

JITS InnovationLab là công ty gia công và công nghệ hàng đầu cung cấp các nền tảng ngân hàng kỹ thuật số, điểm bán hàng, phát trực tuyến và giải pháp tích hợp. Đội ngũ của chúng tôi có nhiều năm kinh nghiệm trong việc triển khai và tùy chỉnh ngân hàng lõi cho khách hàng doanh nghiệp, chúng tôi đã giúp nhiều ngân hàng và tổ chức tài chính vi mô nâng cao sản phẩm của họ, số hóa đang chuyển đổi suôn sẻ với công nghệ mới nhất và phân tích kinh doanh chi tiết đáng kể.

Với đội ngũ tài năng và quy trình làm việc minh bạch, chúng tôi luôn sẵn sàng đồng hành cùng bạn trên con đường theo đuổi những ý tưởng đổi mới.

Với những nỗ lực không ngừng, JITS InnovationLab đã và đang là đối tác tin cấy của nhiều doanh nghiệp trên thế giới. Chúng tôi cam kết mang lại đến cho khách hàng những giải pháp tối ưu cả về giá thành và chất lượng .

Đại diện công ty : Ông Nguyễn Đào Thái : Tổng Giám đốc

Ông Kiểu Tố Vân : Giám đốc điều hành

Địa chỉ công ty : Số 6, đường Phạm Tuấn Tài, phường Dịch Vọng, quận Cầu Giấy, thành phố Hà Nội

Số điện thoại : 0338 355 571

Email: contact@jitsinnovationlabs.com

Website: https://jitsinnovationlabs.com

**1.1.2 Các sản phẩm chính**

Với mục tiêu đáp ứng yêu cầu bài toán của phần đông số lượng khách hàng, JITS InnovationLab đã và đang hoàn thiện các sản phẩm như một module trong hệ thống, không chỉ đáp ứng những thay đổi khách quan của chính sách, chế độ của Nhà nước mà còn hỗ trợ người dùng một cách tối đa bằng sự mềm dẻo, linh hoạt.

Các sản phẩm chính gồm:

* CLOUD POS :
* POS phát triển nội bộ của chúng tôi là một giải pháp hoàn toàn dựa trên đám mây, dành cho nhiều người thuê với các tính năng toàn diện bao gồm kiểm kê, đặt hàng, quầy, chi nhánh, đặt chỗ, khuôn khổ tự tích hợp, cấu hình sản phẩm động.
* Cloud-native, hệ sinh thái, phần mềm như một dịch vụ Saas, phù hợp với mọi doanh nghiệp với cấu hình sản phẩm động, kế toán nâng cao để theo dõi lãi lỗ, chi phí đầu tư thấp.
* Trường hợp nghiên cứu: ngân hàng hoặc ví, cung cấp máy POS miễn phí cho người bán, kiếm lợi nhuận trên MDR và ​​lòng trung thành của người bán (không muốn sử dụng thiết bị đầu cuối ngân hàng khác).
* DIGITAL WALLET
* Ví kỹ thuật số hạng nhất, thúc đẩy thành công với đám mây, mã nguồn mở và API mở.
  + Tự thanh toán, tự do khỏi CoreBanking để quản lý tài khoản, xử lý giao dịch nhanh như chớp.
  + Kiến trúc gốc đám mây với CICD hoàn toàn tự động, tự động mở rộng quy mô với cụm Kubernetes, cập nhật liên tục giúp không có thời gian chết trong quá trình di chuyển, nhẹ và hiệu suất cao, độ bền.
* CUSTOMER LOYALTY WALLET
  + Hệ thống khách hàng thân thiết nhiều người thuê, được xây dựng với công nghệ tiên tiến về phát trực tuyến, cơ sở dữ liệu Mongo, nền tảng đám mây, Kubernetes cho phép nhiều doanh nghiệp cung cấp khuyến mãi, phần thưởng cho khách hàng thân thiết cho cùng một khách hàng mà không cần phải giữ quá nhiều thẻ thành viên vật lý. Điểm có thể được trao đổi giữa các công ty để thuận tiện nhất, các chiến dịch giữ chân khách hàng có thể được thiết kế và giám sát, đồng thời được tối ưu hóa để có ROI tốt nhất. Kết hợp với thương mại điện tử, lòng trung thành có thể nhanh chóng thúc đẩy khối lượng bán hàng.
* FINERACT CORE BANKING
  + Hơn 10 năm kinh nghiệm triển khai ngân hàng lõi cho các tổ chức tài chính như ngân hàng, tài chính vi mô và ví kỹ thuật số. Các chuyên gia của chúng tôi có thể giúp phân tích yêu cầu, ước tính nỗ lực thực hiện và tư vấn để vận hành trơn tru.
* SMART VENDING MACHINE
  + Máy bán hàng tự động thông minh hỗ trợ cả tiền giấy và thanh toán kỹ thuật số (QR), có thể tùy chỉnh cho đồ uống / đồ ăn nhẹ / cà phê / mặt nạ và các sản phẩm tiêu dùng khác
  + Có thể tích hợp với ví kỹ thuật số để thanh toán, nạp tiền bằng tiền giấy, thanh toán hóa đơn bằng tiền giấy.
  + Kiếm nhiều lợi ích hơn với địa điểm kinh doanh của bạn, quảng cáo sản phẩm và dịch vụ của bạn trên màn hình cảm ứng lớn khi khách hàng mua sản phẩm. Thời gian trung bình khách hàng đứng trước máy bán hàng tự động là 2 phút.

**1.1.3 Kinh nghiệm triển khai**

Kinh nghiệm trong Hợp tác triển khai cùng với đối tác và các doanh nghiệp phần mềm khác trong nhiều dự án quan trọng:

* Công ty gia công và công nghệ hàng đầu cung cấp các nền tảng ngân hàng kỹ thuật số, điểm bán hàng, phát trực tuyến và giải pháp tích hợp
* Triển khai bộ phần mềm thanh toán điện tử, có giải pháp về lưu trữ hạ tầng đám mây.
* …

**1.1.4 Khách hàng và đối tác**

* Dịch vụ khí tượng thủy văn quốc gia
* Công ty cổ phần thanh toán quốc gia
* 2T group, GoldWell, MedComm, Mobifone, Supermind, Telemor ..
* Trường đại học mỏ địa chất
* Ngân hàng AYA

**Vị trí thực tập**

Thực tập sinh lập trình viên NodeJS. Tham gia vào dự án Momoney của công ty với vai trò lập trình viên

**Đề tài thực tập**

Xây dựng trang quản trị cho thế giới di động và viết API xác thực mã OPT cho người dùng

**Lịch làm việc**

Thời gian thực tập : Các ngày trong tuần từ 08/2/2022 đến ngày 08/04/2022

**Bảng kế hoạch thực tập**

Bảng 1-1: Kế hoạch làm việc trong 10 tuần thực tập tại đơn vị

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tuần** | **Công việc** | **Người hướng dẫn** | **Mức độ hoàn thành** | **Ghi chú** |
| 1 | -Lên đơn vị thực tập -Tìm hiểu về đơn vị thực tập  -Tìm hiểu về kiến thức chuyên môn yêu cầu: NodeJs,Express, SailsJs … | Nguyễn Thị Tuyền | Hoàn thành |  |
| 2 | -Tìm hiểu về kiến thức chuyên môn yêu cầu: NodeJs,Express, SailsJs ..  -Tìm hiểu về cơ chế xác thực trong nodejs  -Tìm hiểu khái niệm API  -Tìm hiểu về JSON  -Tìm hiểu về dự án đang tham gia, các môi trường và nghiệp vụ cần triển khai. | Nguyễn Thị Tuyền | Hoàn thành |  |
| 3 | -Tìm hiểu các công cụ hỗ trợ trong quá trình thực tập  -Tạo các dự án nhỏ bằng SailsJs và Express | Nguyễn Thị Tuyền | Hoàn thành |  |
| 4 | - Tìm hiểu về đề tài và triển khai App todo  - Xây dựng trang quản trị cho trang thế giới di động  -Lên kế hoạch các nhiệm vụ và thời gian hoàn thành các công việc do người hướng dẫn giao | Nguyễn Thị Tuyền | Hoàn thành | Học thêm bổ trợ phân tích thiết kế hệ thống |
| 5 | -Tiến hành phân tích và thiết kế cơ sở dữ liệu cho các chức năng của hệ thống  -Bắt đầu lập trình các chức năng dự trên cơ sở dữ liệu **-**Khắc phục các lỗi phát sinh sau khi hoàn thành 1 nhiệm vụ ( sửa lại lỗi phân tích)  -Báo có tiến độ với người phụ trách | Nguyễn Thị Tuyền | Hoàn thành |  |
| 6 | -Tiến hành triển khai các chức năng dự trên cơ sở dữ liệu đã thiết kế  **-**Khắc phục các lỗi phát sinh sau khi hoàn thành 1 nhiệm vụ ( sửa lại lỗi phân tích)  **-** Tìm hiểu về cơ sở dữ liệu MongoDB  -Tìm hiểu về Hàng đợi Bull và cách giải quyết  -Báo cáo tiến độ với người hướng dẫn | Nguyễn Thị Tuyền | Hoàn thành |  |
| 7 | - Tìm hiểu hiểu về Redis và thực hiện 1 số thao tác cơ bản  - Tìm hiểu về Firebase authentication và cơ chế hoạt động. | Nguyễn Thị Tuyền | Hoàn thành |  |
| 8 | - Báo có tiến độ của các công việc đang thực hiện về đề tài thế giới di động  - Cài đặt và tìm hiểu về Node -RED | Nguyễn Thị Tuyền | Hoàn thành |  |
| 9 | - Tham gia dự án Momony, tạo các biller, và hoàn thiện các công viện đc giao | Nguyễn Thị Tuyền | Hoàn thành |  |
| 10 | **-**Chuẩn bị bài nghiệp thu TTDN và xin đánh giá từ phía người hướng dẫn tại đơn vị. | Nguyễn Thị Tuyền | Hoàn thành |  |

**CHƯƠNG 2****NỘI DUNG THỰC TẬP**

**Nội dung thực tập**

**Mô tả công việc**

Mô tả chi tiết về các công việc cần thực hiện trong quá trình thực tập và kết quả của công việc đó

Bảng 2-1: Mô tả chi tiết về các công việc cần thực hiện trong quá trình thực tập

|  |  |
| --- | --- |
| Tuần | Mô tả |
| 1 | -Làm quen và điền thông tin giấy tờ cần thiết .  -Giới thiệu bản thân  -Tìm hiểu về đơn vị thực tập : Lịch sử , cơ cấu tổ chức, nhân sự , lĩnh vực hoạt động , các sản phẩm , đối tác ,…  -Tìm hiểu về kiến thức chuyên môn yêu cầu: NodeJs ,Express, SailsJs, Phân tích thiết kế hệ thống, thiết kế giao diện …  -Cài đặt các công cụ cần thiết : Node-Red, SailsJs … |
| 2 | -Tìm hiểu về kiến thức chuyên môn yêu cầu: NodeJs ,Express, SailsJs, Phân tích thiết kế hệ thống, thiết kế giao diện ..  -Tìm hiểu chức năng , cách sử dụng các công cụ hỗ trợ trong quá trình thực tập  -Tìm hiểu các cơ chế xác thực người dùng và cách hoạt động của các phương thức:   * JWT (Json Web Token) * Xác thực OPT * Firebase authentication * …   -Tìm hiểu về API  - Tìm hiểu về JSON |
| 3 | -Tìm hiểu chức năng , cách sử dụng các ông cụ hỗ trợ trong quá trình thực tập  -Tìm hiểu và xây dựng các để tải nhỏ về SailsJs và Express  + CRUD sinh viên   * Tạo bảng và kết nối đến cơ sở dữ liệu * API: list danh sách sinh viên * Thêm mới danh sách sinh viên * Sửa thông tin sinh viên * Xóa sinh viên trong danh sách   + Thêm phương thực xác thực   * Tạo bảng và kết nối đến cơ sở dữ liệu * API đăng nhập có xác thực token * Function xác thực API * API: list danh sách sinh viên * Thêm mới danh sách sinh viên * Sửa thông tin sinh viên * Xóa sinh viên trong danh sách * Api upload avatar cho sinh viên |
| 4 | -Lên kế hoạch các nhiệm vụ và thời gian hoàn thành các công việc do người hướng dẫn giao  - Tìm hiểu về đề tài và triển khai App todo  - Xây dựng trang quản trị cho trang thế giới di động  - Phân tích hệ thống theo hướng chức năng   * Xác định yêu cầu của hệ thống gồm: các chức năng chính – phụ; nghiệp vụ cần phải xử lý đảm bảo tính chính xác, tuân thủ đúng các văn bản luật và quy định hiện hành; đảm bảo tốc độ xử lý và khả năng nâng cấp trong tương lai. * Phân tích và đặc tả mô hình phân cấp chức năng tổng thể thông qua sơ đồ BFD (Business Flow Diagram), từ mô hình BFD sẽ tiếp tục được xây dựng thành mô hình luồng dữ liệu DFD (Data Flow Diagram) thông qua quá trình phân rã chức năng theo các mức 0, 1, 2 ở từng ô xử lý. |
| 5 | -Tiến hành phân tích và thiết kế cơ sở dữ liệu cho các chức năng của hệ thống   * Dựa vào phân tích và đặc tả * Thiết kế cơ sở dữ liệu theo phân tích * Mô tả đầy đủ các thông tin về cơ sở dữ liệu   -Bắt đầu lập trình các chức năng dự trên cơ sở dữ liệu  -Từ cơ sở dữ liệu bắt đầu lập trình các chức năng  **TodoApp**   * + Đăng nhập đăng ký (Xác thực token và gữi mã OTP về gmail của người dùng)   + Kiểm tra mã token và Thiết bị đăng nhập có đúng không mới cho thực hiện các công việc sau   + Viết API list các công viêc (Có xác thực token)   + APi thêm công việc (Có xác thực token)   + API sửa công viêc (Có xác thực token)   + API xóa công viêc (Có xác thực token   **Thế giới di động**   * Thiết kế giao diện trang quản trị * Thực hiện làm các chức năng * Đăng nhập đăng ký      * Quản lý danh mục      * Quản lý thương hiệu      * Quản lý sản phẩm      * Quản lý đơn hàng      * Quản lý tài khoản      * Quản lý phản hồi(đánh giá và liên hệ)        * Quản lý tin tức      * Quản lý banner     **-**Khắc phục các lỗi phát sinh sau khi hoàn thành 1 nhiệm vụ ( sửa lại lỗi phân tích)  -Báo có tiến độ với người phụ trách |
| 6 | -Bắt đầu lập trình các chức năng dự trên cơ sở dữ liệu  -Từ cơ sở dữ liệu bắt đầu lập trình các chức năng  **TodoApp**   * + Đăng nhập đăng ký (Xác thực token và gữi mã OTP về gmail của người dùng)   + Kiểm tra mã token và Thiết bị đăng nhập có đúng không mới cho thực hiện các công việc sau   + Viết API list các công viêc (Có xác thực token)   + APi thêm công việc (Có xác thực token)   + API sửa công viêc (Có xác thực token)   + API xóa công viêc (Có xác thực token   **Thế giới di động**   * Thiết kế giao diện trang quản trị * Thực hiện làm các chức năng * Đăng nhập đăng ký * Quản lý danh mục * Quản lý thương hiệu * Quản lý sản phẩm * Quản lý đơn hàng * Quản lý tài khoản * Quản lý phản hồi(đánh giá và liên hệ) * Quản lý tin tức * Quản lý banner   **-**Khắc phục các lỗi phát sinh sau khi hoàn thành 1 nhiệm vụ ( sửa lại lỗi phân tích)  **-** Tìm hiểu về cơ sở dữ liệu MongoDB  - **Index** trong mongodb :  + Tạo chỉ mục  + Single Index - vd : db.user.createIndex({ email: 1});  +Compound index - vd : db.user.createIndex({ email: 1, age:1});  +Multikey index đánh chỉ mục cho nội dung trong mảng : db.user.createIndex( {"addr.zip":1})  + Chỉ mục (Index) hỗ trợ việc phân giải các truy vấn hiệu quả hơn  - **Replica set** trong mongodb :  + Replication là tiến trình đồng bộ hóa dữ liệu từ nhiều Server. Replication cung cấp sự dư thừa và tăng dữ liệu có tính khả dụng với nhiều bản sao dữ liệu trên nhiều Database Server khác nhau.  + Replication là một tiến trình đồng bộ hóa dữ liệu.  + Dùng 2 server mongodb là A và B, khi có dữ liệu mới được lưu vào A thì nó sẽ tiến hành sao lưu, đồng bộ sang B (quá trình replica) nếu A bị chết hoặc mất dữ liệu thì ta chuyển sang kết nối tới B để lấy dữ liệu như thế sẽ đảm bảo server luôn hoạt động và dữ liệu tin cậy không bị mất mát.  -Tìm hiểu về Hàng đợi Bull và cách giải quyết  + **Bull** là một thư viện Node triển khai một hệ thống queue nhanh chóng và mạnh mẽ dựa trên Redis. Mặc dù có thể triển khai queue trực tiếp bằng lệnh Redis, Bull là một trừu tượng / trình bao bọc trên Redis. Nó cung cấp một API xử lý tất cả các chi tiết cấp thấp và làm phong phú thêm chức năng cơ bản của Redis để có thể dễ dàng xử lý các trường hợp sử dụng phức tạp hơn.   * Xử lý background. * Gửi thông tin theo hàng đợi. * Xử lý đồng bộ, concurrency.   + **Bull** là thư viện có thể giúp cả 3 vấn đề trên, và bản thân cũng có 3 thành phần chính:   * Producer: Thành phần tạo ra các công việc (job) và ném chúng vào queue (hàng đợi). * Consumer: Thành phần nhận việc từ queue. * Listener: Thành phần lắng nghe các sự kiện diễn ra trong hàng đợi (completed, failed, stalled).   Về cơ bản, message queue là cơ chế đưa các thông tin vào dãy để xử lý dần, và mặc định với Bull thì hàng đợi là dạng FIFO (dữ liệu vào trước thì ra trước).  +Tùy chọn hàng đợi  Rate Limiter(Giới hạn tỷ lệ) : giới hạn số lượng công việc trong 1 đơn vi thời gian.  limiter: {  max: 1000,  duration: 5000  }  Named jobs(Công việc đc đặt trước) : Đặt tên cho các công việc  - producer : transcoderQueue.add  -worker : transcoderQueue.process  Các loại công việc  Bị trì hoãn : const myJob = await myqueue.add({ foo: 'bar' }, { delay: 5000 });  Cuộc sống (LIFO) : onst myJob = await myqueue.add({ foo: 'bar' }, { lifo: true });  Ưu tiên : const myJob = await myqueue.add({ foo: 'bar' }, { priority: 3 });  -Báo có tiến độ với người phụ trách |
| 7 | - Tìm hiểu hiểu về Redis và thực hiện 1 số thao tác cơ bản  + **Redis** :  + Lưu trữ dưới dạng key-value  + cấu trúc dữ liệu :STRING, LIST, SET, HASH  +String - vd :set name "kien"  get name => kien  del name (xóa)  EXPIRE name 10 (set thời gian )  TTL name ( xem tg sống)  STRLEN name (xem số lượng phẩn tử)  APPEND name(thêm phàn tử)  +List - vd : RPUSH list A (thêm bên phải)  LPUSH list B (thêm bên trái)  LRANGE list 0 -1 (lấy phần tử từ đầu đến cuối)  LPOP list và RPOP list (xóa bên trái xóa bên phải)  +Hash - vd : HSET user username "kien"  HGET user username => kien  HMSET user birth 2000 sex male  HMGET user birth sex => 200 male  - Tìm hiểu về Firebase authentication và cơ chế hoạt động.  + **Firebase** là dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây – cloud. Kèm theo đó là hệ thống máy chủ cực kỳ mạnh mẽ của Google. Chức năng chính là giúp người dùng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu.  +**Firebase Authentication** là chức năng dùng để xác thực người dùng bằng Password, số điện thoại hoặc tài khoản Google, Facebook hay Twitter, v.v. |
| 8 | - Báo có tiến độ của các công việc đang thực hiện về đề tài thế giới di động  - Cài đặt và tìm hiểu về Node -RED  +Node RED là một công cụ lập trình dùng để kết nối các thiết bị phần cứng, API và các dịch vụ trực tuyến với nhau. Về cơ bản, đây là một công cụ trực quan được thiết kế cho IoT (Internet of Things), nhưng cũng có thể được sử dụng cho các ứng dụng khác nhằm liên kết nhanh các luồng (flow) dịch vụ khác nhau. |
| 9 | - Tham gia dự án Momony, tạo các biller, và hoàn thiện các công viện đc giao  - Tạo các category biller, các form biller, các biller |
| 10 | **-**Chuẩn bị bài nghiệp thu TTDN và xin đánh giá từ phía người hướng dẫn tại đơn vị. |

**Các công cụ đã tìm hiểu**

Bảng 2-2:Một số công cụ đã tìm hiểu và áp dụng trong qua trình thực tập

|  |  |
| --- | --- |
| Công cụ đã tìm hiểu | Mô tả |
| **NodeJs** | **Khái niệm** : Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript giúp chúng ta có thể xây dựng được các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp một cách nhanh chóng và dễ dàng mở rộng. NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ Window cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất. **Hoạt động :** Node js là sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng. Bởi vì, Node js có khả năng mở rộng nhanh chóng, khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao. Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống thì việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả. Chính vì lẽ đó giải pháp mà Node js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ hàng chục ngàn kết nối đồng thời.  **Cấu trúc :** **Module** -Module giống như các thư viện JavaScript sử dụng trong ứng dụng Node.js application bao gồm một bộ các chức năng. Để đưa một module vào ứng dụng Node.js thì phải sử dụng hàm require() function with the với dấu ngoặc đơn chứa tên module. **Console** -Bảng điều khiển cung cấp phương thức gỡ lỗi tương tự như bảng điều khiển cơ bản của JavaScript trên các trình duyệt internet. Nó sẽ in các thông báo ra stdout và stderr. **Cluster** -Node.js được xây dựng dựa trên ý tưởng lập trình đơn luồng. Cluster là một module cho phép đa luồng bằng cách tạo ra các quy trình con có chung cổng máy chủ và chạy đồng thời. **Global** -Biến toàn cục trong Node.js tồn tại trong tất cả các module. Những biến này bao gồm các hàm, module, string, v…v…. **Error Handling****Debugger** -Node.js có chức năng gỡ lỗi có thể được sử dụng với một client gỡ lỗi được tích hợp sẵn. Trình gỡ lỗi của Node.js không có quá nhiều tính năng nhưng nó hỗ trợ các chức năng kiểm tra code cơ bản. Trình gỡ lỗi có thể được sử dụng trong bảng lệnh bằng cách sử dụng từ khoá 'inspect' phía trước tên của file JavaScript. Ví dụ, để kiểm tra file myscript.js, **Node.js được sử dụng để :****Backend for Social Media Networking** Nhiều tên tuổi nổi tiếng như LinkedIn và Medium khác được xây dựng bởi Node.js. Có một sự kết hợp thiết yếu của các nút để tạo phần phụ trợ cho một trang mạng truyền thông xã hội. Node.js cung cấp khả năng định tuyến cực nhanh với động cơ V8 cùng với xác thực an toàn. Khả năng mở rộng là một trong những yếu tố quan trọng nhất làm phong phú thêm mạng truyền thông xã hội. **Single-page Application (SPA) Development** Cũng giống như các trang web đơn lẻ, Node.js có thể xây dựng một ứng dụng một trang, nơi giao diện tương tự như một ứng dụng dành cho máy tính để bàn. Do các tùy chọn linh hoạt của nó, Node.js rất phù hợp để xây dựng nền tảng mạng xã hội, trang web động và giải pháp gửi thư. Hơn nữa, chất lượng luồng dữ liệu không đồng bộ trên phần phụ trợ trong Node.js khiến nó trở thành lựa chọn tốt nhất để phát triển SPA. **Chatbots** Node.js cung cấp các chức năng nâng cao độc quyền cho chatbots và ứng dụng trò chuyện thời gian thực. Các tính năng chính như ứng dụng nhiều người dùng, dữ liệu chuyên sâu, lưu lượng truy cập lớn, trên tất cả các thiết bị đều rất phổ biến trong chatbots. Node.js đã đạt được điểm tốt, vì nó chạy trên các thiết bị khác nhau và bao gồm tất cả các mô hình trong ứng dụng trò chuyện và chatbot. Node.js giúp việc thực thi các thông báo đẩy và vòng lặp sự kiện phía máy chủ trở nên cực kỳ dễ dàng được sử dụng phổ biến trong IMs và ứng dụng real-time. **Data Streaming** Những gã khổng lồ như Netflix đã chào đón Node.js để phục vụ 190 quốc gia và hơn 100 triệu giờ phát trực tuyến trên 120 triệu người dùng và còn tiếp tục tăng. Node.js có một lợi thế rõ ràng về điều này là các tệp xử lý ở tốc độ siêu nhanh, đồng thời mã hóa và tải lên nhẹ nhàng. Điều này đã được nhiều cửa hàng điện tử thời trang trực tuyến và các ứng dụng có video truyền tải dữ liệu khổng lồ áp dụng. **IoT Application Development** IoT đã trở nên phổ biến mạnh mẽ trong thập kỷ gần đây và kể từ đó Node.js là một trong những giải pháp được ưa thích nhất cho các tổ chức muốn xây dựng các hệ thống giải pháp IoT công cộng và riêng tư. Lợi thế cốt lõi của Node.js là khả năng xử lý các yêu cầu đồng thời liền mạch với hàng nghìn sự kiện được phát hành bởi hàng tỷ thiết bị trên mạng của nó. Các mạng IoT. Hơn nữa, Node.js hoạt động trên các kênh và luồng có thể ghi và đọc được, đó là lý do tại sao nó là nền tảng phù hợp nhất để phát triển ứng dụng IoT. |
| **Express** | Express là một web application framework for nodejs, nó cung cấp cho chúng những rất nhiều tính năng mạnh mẽ trên nền tảng web. Express rất dễ dàng để phát triển các ứng dụng nhanh dựa trên Node.js cho các ứng dụng Web. Express hỗ trợ các phương thức HTTP và middleware tạo ra 1 API rất mạnh mẽ và sử dụng dễ dàng hơn  Khi mới tiếp cận với Express mình thực sự bị cuốn hút bởi các API của nó, từ cách sử dụng route, template, đều khá dễ tùy chọn và làm việc. Các tính năng của Express framework phải kể đến như:   * Cho phép thiết lập các lớp trung gian để trả về các HTTP request. * Định nghĩa routing có thể được sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL. * Cho phép trả về các trang HTML dựa vào các tham số truyền vào đến template  Cài đặt Express Framework  * npm install express –save * **body-parser** - Đây là một lớp trung gian node.js để xử lí JSON, dự liệu thô, text và mã hóa URL. * **cookie-parser**- Chuyển đổi header của Cookie và phân bố đến các req.cookies * **multer** - Đây là một thành phần trung gian trong node.js để xử lí phần multipart/form-data.   **Request & Response**   * [Request](http://vietjack.com/nodejs/doi_tuong_request_trong_nodejs.jsp) : Biểu diễn một HTTP request, và có các thuộc tính cho các request như các chuỗi truy vấn, tham số, HTML. * [Response](http://vietjack.com/nodejs/doi_tuong_response_trong_nodejs.jsp) : Biểu diễn HTTP response được ứng dụng Express gửi đi khi nó nhận các HTTP request.   **Định tuyến cơ bản**  Qua 2 phần giới thiệu ở trên, chắc rằng các bạn cũng đã hiểu cơ bản phần nào về Express. Trong phần này chúng ta sẽ cùng nhau tìm hiểu sâu hơn về Express.  **Làm việc với file tĩnh**  Express cung cấp lớp tiện ích trung gian **express.static** để phục vụ cho các file tĩnh như các file hình ảnh, css, js.  Cách hoạt động của nó về cơ bản bạn chỉ cần truyền tên thư mục, nơi bạn chứa cacs file tĩnh thì **express.static** sẽ xử dụng file đó một cách trực tiếp. Ví dụ, bạn muốn giữ hình ảnh, css, js trong thư mục public. Bạn có thể làm như sau:   * app.use(express.static('public'));  **Express.js hoàn toàn miễn phí** Một trong những yếu tố để tổ chức, doanh nghiệp nên sử dụng công cụ, ứng dụng nào chính là chi phí. Với một ứng dụng web với nhiều tính năng tuyệt vời như vậy mà lại không tốn một chút kinh phí nào thì không có lý do gì để không sử dụng nó. **Những tính năng của Express :**  * Phát triển máy chủ nhanh chóng: Expressjs cung cấp nhiều tính năng dưới dạng các hàm để dễ dàng sử dụng ở bất kỳ đâu trong chương trình. Điều này đã loại bỏ nhu cầu viết mã từ đó tiết kiệm được thời gian. * Phần mềm trung gian Middleware: Đây là phần mềm trung gian có quyền truy cập vào cơ sở dữ liệu, yêu cầu của khách hàng và những phần mềm trung gian khác. Phần mềm Middleware này chịu trách nhiệm chính cho việc tổ chức có hệ thống các chức năng của Express.js. * Định tuyến - Routing: Express js cung cấp cơ chế định tuyến giúp duy trì trạng thái của website với sự trợ giúp của URL. * Tạo mẫu - Templating: Các công cụ tạo khuôn mẫu được Express.js cung cấp cho phép các nhà xây dựng nội dung động trên các website bằng cách tạo dựng các mẫu HTML ở phía máy chủ. * Gỡ lỗi - Debugging: Để phát triển thành công các ứng dụng web không thể thiết đi việc gỡ lỗi. Giờ đây với Expressjs việc gỡ lỗi đã trở nên dễ dàng hơn nhờ khả năng xác định chính xác các phần ứng dụng web có lỗi. |
| SailsJs | **Khái niệm** :Sails.js được xây dựng dựa trên Express.js (Express là 1 Framework của Node.js), cộng với các chức năng nâng cao giúp giảm thiếu tối đa thời gian lập trình.  Sails.js kế thừa mô hình MVC (Model-View-Controller), cơ chế hoạt động khá giống các Framework của PHP như Laravel,…  Sails.js được tích hợp [Socket.io](http://socket.io/), cực kỳ thích hợp cho ứng dụng chat hoặc trò trơi trực tuyến. Đặc điểm **100% JavaScript**  Xây dựng trên Sails có nghĩa là ứng dụng của bạn được viết hoàn toàn bằng JavaScript, ngôn ngữ mà bạn và nhóm của bạn đã sử dụng trong trình duyệt.  **Any database**  Sails bundles là một ORM mạnh mẽ, cung cấp lớp truy cập dữ liệu đơn giản để làm việc, bất kể bạn đang sử dụng cơ sở dữ liệu nào.  **Auto-generated REST APIs**  Sails đi kèm với bản thiết kế giúp khởi động phần phụ trợ của ứng dụng mà không cần viết bất kỳ mã nào.  **Front-end agnostic**  Sails tương thích với mọi front-end: Angular, React, iOS, Android, Windows Phone, phần cứng tùy chỉnh hoặc những thứ khác hoàn toàn.  **Easy WebSocket integration**  Sails dịch các socket messages đến cho bạn, chúng tự động tương thích với mọi tuyến đường trong ứng dụng Sails của bạn.  **Professional support**  Sails cung cấp hỗ trợ thương mại để tăng tốc phát triển và đảm bảo thực hành tốt nhất trong code của bạn.  **Cấu trúc thư mục**   * **api** * **app.js** * **assets** * **config** * **Gruntfile.js** * **node\_modules** * **package.json** * **README.md** * **tasks** * **views**  Lifecycle **Load Configuration "Overrides"**  Tập hợp các giá trị config được truyền vào trong command line, trong các biến môi trường và trong config programmatic (nghĩa là các tùy chọn được truyền cho sails.load hoặc sails.lift.). Khi một ứng dụng được khởi động thông qua giao diện command-line (tức là bằng cách gõ sails lift hoặc sails console), các giá trị của bất kỳ tệp .sailsrc nào cũng sẽ được hợp nhất vào phần config overrides. Các giá trị overrides này sẽ được ưu tiên hơn bất kỳ config người dùng nào gặp phải trong bước tiếp theo.  **Load User Configuration**  Trừ khi hook userconfiguration bị vô hiệu hóa rõ ràng, Sails tiếp theo sẽ tải các tệp cấu hình trong thư mục config (và các thư mục con) bên dưới thư mục làm việc hiện tại. Xem [Concepts > Configuration](https://sailsjs.com/documentation/concepts/configuration) để biết thêm chi tiết về cấu hình người dùng. Cài đặt cấu hình từ bước 1 sẽ được hợp nhất trên đầu các giá trị này để tạo thành đối tượng sails.config.  **Load Hooks**  Tiếp theo, Sails sẽ tải các hook khác. Core hooks sẽ tải trước, tiếp theo là user hooks và installable hooks. Lưu ý rằng các hook thường bao gồm cấu hình của riêng chúng sẽ được sử dụng làm giá trị mặc định trong sails.config. Ví dụ: nếu không có cài đặt port nào được cấu hình vào thời điểm này, giá trị mặc định của http hook là 1337 sẽ được sử dụng.  **Assemble Router**  Sails chuẩn bị core Router, sau đó emit nhiều sự kiện trên đối tượng sails thông báo các hook mà họ có thể bind routes một cách an toàn.  **Expose global variables**  Sau khi tất cả các hook đã được khởi tạo, Sails hiển thị các biến toàn cục (theo mặc định: sails object, models, services, \_ và async)  **Initialize App Runtime**  Bước này không chạy khi sử dụng sails.load (). Để chạy bước Initialize, thay vào đó, hãy sử dụng sails.lift()  Chạy function bootstrap (sails.config.bootstrap) |
| **API** | **Khái niệm :** API là viết tắt của Application Programming Interface – phương thức trung gian kết nối các ứng dụng và thư viện khác nhau.  Nó cung cấp khả năng truy xuất đến một tập các hàm hay dùng, từ đó có thể trao đổi dữ liệu giữa các ứng dụng.  Thi thoảng vẫn có người lầm tưởng API là một [ngôn ngữ lập trình](https://itviec.com/blog/ngon-ngu-lap-trinh/" \t "_blank) nhưng thực ra, API chỉ là các hàm hay thủ tục thông thường. Các hàm này được viết trên nhiều ngôn ngữ lập trình khác nhau.  **Đặc điểm :**   * API sử dụng mã nguồn mở, dùng được với mọi client hỗ trợ XML, JSON. * API có khả năng đáp ứng đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content forma…. Bạn có thể sử dụng các host nằm trong phần ứng dụng hoặc trên IIS. * Mô hình web API dùng để hỗ trợ MVC như: unit test, injection, ioc container, model binder, action result, filter, routing, controller. Ngoài ra, nó cũng hỗ trợ RESTful đầy đủ các phương thức như: GET, POST, PUT, DELETE các dữ liệu. * Được đánh giá là một trong những kiểu kiến trúc hỗ trợ tốt nhất với các thiết bị có lượng băng thông bị giới hạn như smartphone, tablet…   **Ưu nhược điểm của API :** **Ưu điểm**  1. Giao tiếp hai chiều phải được xác nhận trong các giao dịch sử dụng API. Cũng chính vì vậy mà các thông tin rất đáng tin cậy. 2. API là công cụ mã nguồn mở, có thể kết nối mọi lúc nhờ vào Internet. 3. Hỗ trợ chức năng RESTful một cách đầy đủ. 4. Cấu hình đơn giản khi được so sánh với WCF (Window Communication Foundation). Cung cấp cấp trải nghiệm thân thiện với người dùng.  **Nhược điểm**  * Tốn nhiều chi phí phát triển, vận hành, chỉnh sửa. * Đòi hỏi kiến thức chuyên sâu. * Có thể gặp vấn đề bảo mật khi bị tấn công hệ thống.  **Ứng dụng của API**  * **Web API**: Là hệ thống API được sử dụng trong các hệ thống website, chẳng hạn: Google, Facebook… Hầu hết các website đều cung cấp hệ thống API cho phép bạn kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu. Đa số Web API được thiết kế theo tiêu chuẩn * **API trên hệ điều hành**: Windows hay Linux có rất nhiều API. Họ cung cấp các tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức kết nối. Nó giúp [lập trình viên](https://itviec.com/viec-lam-it/developer?utm_source=blogpost&utm_medium=center_post&utm_campaign=linktosite" \t "_blank) có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành. * **API của thư viện phần mềm (framework):** API mô tả và quy định các hành động mong muốn mà các thư viện cung cấp. Một API có thể có nhiều cách triển khai khác nhau, giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng được thư viện viết bằng ngôn ngữ khác.  **Những API miễn phí dành cho developer**  * [**Dog API**](https://dog.ceo/dog-api/): API Dog hoàn toàn miễn phí, cung cấp các endpoints cho các ảnh và dữ liệu văn bản về chó. Nó hỗ trợ CORS do đó chúng ta có thể dùng trực tiếp từ các ứng dụng front-end. * [**Petfinder**](https://www.petfinder.com/developers/v2/docs/): API này cung cấp các dữ liệu về nhận nuôi thú cưng, yêu cầu xác thực bằng OAuth để truy cập dữ liệu. API này cũng hỗ trợ CORS như Dog API. * [**shibe.online**](https://shibe.online/)**:** shibe.online cung cấp URL cho các hình ảnh về những chú chó dễ thương để làm dữ liệu. Nó dùng các chuỗi truy vấn để lấy dữ liệu từ API. * [**Anime News Network API**](https://www.animenewsnetwork.com/encyclopedia/api.php)**:** Nếu bạn thích Anime thì API này chính là dành cho bạn. Nó trả về dữ liệu ở các định dạng khác nhau như HTML hoặc XML. Do đó, nếu bạn cần thao tác và hiển thị dữ liệu thì bạn sẽ phải chuyển đổi từ XML sang JSON. * [**Jikan**](https://jikan.moe/)**:** Đây là một API dữ liệu anime khác, có 1 REST API hỗ trợ CORS bên cạnh các thư viện được tạo cho các nền tảng khác nhau như Python, Ruby và JavaScript. * [**JSONPlaceholder**](https://jsonplaceholder.typicode.com/)**:** Cung cấp một API với các yêu cầu GET, POST, PUT và DELETE có hỗ trợ CORS. Các endpoints được cung cấp với dữ liệu được tải sẵn mà chúng ta có thể dùng để hiển thị và cập nhật. * [**REQIRES**](https://reqres.in/)**:**Cũng giống như JSONPlaceholder, REQ|RES cung cấp 1 API thực sự để bạn xây dựng giao diện người dùng. Nó có các endpoints cho tất cả các thao tác HTTP được sử dụng để nhận và thao tác dữ liệu người dùng fake. Các dữ liệu này đã được chuẩn bị trước nên bạn cũng không phải tự thêm vào nữa |
| **JSON** | **Khái niệm :** JSON là tên viết tắt của cụm từ JavaScript Object Notation, nó là một kiểu định dạng dữ liệu tuân theo một quy tắc nhất định mà hầu hết mọi ngôn ngữ lập trình hiện nay đều có thể đọc được. JSON là một trong những tiêu chuẩn mở giúp trao đổi dữ liệu có trên website. Nói cách khách, JSON là định dạng giúp lưu trữ các thông tin có cấu trúc và nó chủ yếu được dùng để truyền tải dữ liệu giữa server và client.  JSON là sẽ sử dụng các cặp key-value để dữ liệu có thể sử dụng. Khi đó, nó sẽ hỗ trợ cho những cấu trúc dữ liệu dạng đối tượng và mảng. Các nhà phát triển thường xuyên sử dụng JSON để có thể làm việc với AJAX, cách định dạng này sẽ phối hợp hiệu quả với nhau để có thể nén dữ liệu tải không đồng bộ.  Có nghĩa như sau: trang web có thể được cập nhật thông tin mà bạn không cần phải làm mới trang. Đây là quá trình rất dễ thực hiện, và hiện nay thì nhiều trang web đang áp dụng AJAX, file.Json đã dần trở nên phổ biến hơn. Ngoài ra, nó còn cho phép người dùng yêu cầu dữ liệu từ nhiều domain khác bằng phương thức gọi là JSONP bằng biện pháp: áp dụng thẻ  **Ưu điểm** :  JSON đang rất được yêu thích hiện nay bởi vì nó sở hữu những tính năng nổi bật như:   * JSON là định dạng trao đổi dữ liệu của văn bản với dung lượng nhẹ. * JSON là một ngôn ngữ hoàn toàn độc lập. * JSON thường có thể tự mô tả được và rất dễ hiểu.   Hiện nay, chuỗi JSON thường sẽ được mã hóa với 2 dạng là: Mã hóa Json\_encode và Json\_decode. Và sau khi thực hiện mã hóa thì việc kiểm tra nội dung đã được mã hóa và được lưu cần đưa vào test bằng trang web chính của JSON là: json.parser.ofline.fr **Nên hiểu định dạng JSON**  Nếu như bạn còn thắc mắc hoặc không biết định dạng **JSON là gì** thì nó được hiểu như sau: Các dạng JSON hoặc định dạng JSON được thiết kế khá rõ ràng nên người dùng hoàn toàn có thể theo đó mà áp dụng với các quy tắc phù hợp nhất. Bởi vì, định dạng JSON thường chỉ là một văn bản nên nó rất dễ để chuyển đổi cũng như trao đổi dữ liệu giữa các máy chủ và các máy khác.  Nó sẽ được đem ra ứng dụng tương ứng với một định dạng dữ liệu thông dụng bởi bất kỳ một ngôn ngữ lập trình nào.Chính vì thế, nếu như bạn nhận được ít dữ liệu từ máy chủ có trong định dạng JSOn thì bạn nên sử dụng nó dựa theo bất kỳ một đối tượng nào khác có sẵn trên Javascript. |

**CHƯƠNG 3****ĐÁNH GIÁ KẾT QUẢ VÀ ĐỊNH HƯỚNG PHÁT TRIỂN**

**Đánh giá kết quả thực tập**

Tự đánh giá kết quả của quá trình thực tập:

* Về kiến thức
* Những kiến thức đã học ở trường đã được dùng trong công việc như:   
  phân tích thiết kế hệ thống,thiết kế website ….
* Về kỹ năng
* Những kiến thức mới đã được tiếp cận, tìm hiểu trong quá trình thực tập như: NodeJs , thiết kế website ,API …
* Ngoài ra em còn tiếp thu được một số kỹ năng mềm như: Kỹ năng giao tiếp, kỹ năng làm việc nhóm,….

**Định hướng phát triển của đề tài**

Đề tài phân tích thiết kế hệ thống và lập trình website là một đề tài khá phổ biến hiện nay, có rất nhiều công ty công nghệ cần nguồn lực và website quản lý, bán hàng… thật tốt để đáp ứng nhu cầu của khách hàng, vậy nên để có một phần mềm tốt , hoàn hảo thì công tác phân tích thiết kế hệ thống và lập trình website là rất quan trọng quyết định đến thành công của dự án.

Với đề tài này em có định hướng sẽ xây dựng một hệ thống quản lý đề tài tốt nghiệp của sinh viên cho khoa công nghệ thông tin thân thiện , bổ ích , dễ sử dụng để cung cấp đến thị trường .

# KẾT LUẬN

Thông qua quá trình thực tập này, cá nhân em đã thu thập được nhiều kinh nghiệm quý báu, có ý nghĩa thiết thực về nghề nghiệp, tự phát triển và rèn luyện các kỹ năng nghề nghiệp và định hướng tương lai của bản thân. Ngoài ra,quá trình thực tập tại Công ty TNHH JITS InovastionLabs em đã có cơ hội áp dụng những kiến thức đã học tại trường và được học hỏi, tiếp thu những kiến thức mới để áp dụng vào thực tế. Thực tập tại công ty giúp em có them nhiều kinh nghiệp về làm việc nhóm, kỹ năng giao tiếp và nhiều kiến thức mới .

Do còn nhiều sự thiếu sót trong kiến thức và trải nghiệm cá nhân nên báo cáo này chưa thực sự hoàn thiện, mong rằng sẽ nhận được sự góp ý từ thầy cô và các anh chị. Nhân đây, em xin chân thành cảm ơn sự giúp đỡ nhiệt tình từ Công ty TNHH JITS InovastionLabs và quý thầy cô vì đã giúp đỡ em trong quá trình thực tập của mình!