BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO THỰC TẬP CHUYÊN NGÀNH**

**THIẾT KẾ WEBSITE CHO CÔNG TY VELOX TECH**

**Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh**

**Sinh viên thực hiện: Nguyễn Minh Hoàng**

**Mã số sinh viên: 59130802**

KHÁNH HÒA-2021

TRƯỜNG ĐẠI HỌC NHA TRANG

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**



**BÁO CÁO THỰC TẬP CHUYÊN NGÀNH**

**THIẾT KẾ WEBSITE CHO CÔNG TY VELOX TECH**

Giảng viên hướng dẫn: ThS. Đoàn Vũ Thịnh

Sinh viên thực hiện: Nguyễn Minh Hoàng

Mã số sinh viên: 59130802

Khánh Hòa, tháng 01/2021

**NHẬN XÉT CỦA CÁN BỘ HƯỚNG DẪN**

**1. NHẬN XÉT CHUNG:**

**2. ĐÁNH GIÁ:**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Nội dung đánh giá** | **Điểm tối đa** | **Điểm đánh giá** |
| 01 | Chấp hành nội quy làm việc tại cơ quan | **01** |  |
| 02 | Tham gia đầy đủ và nghiệm túc các buổi thực tập tại đơn vị thực tập | **02** |  |
| 03 | Có quan hệ và thái độ giao tiếp tốt đối với cán bộ - công nhân viên tại đơn vị thực tập | **02** |  |
| 04 | Tích cực tham gia và học tập chuyên môn trong quá trình thực tập tại đơn vị | **03** |  |
| 05 | Hoàn thành tốt các công việc được giao | **02** |  |
|  | **Tổng cộng** | |  |

|  |  |
| --- | --- |
| **XÁC NHẬN CỦA CƠ QUAN THỰC TẬP**  *(ký, ghi rõ họ tên và đóng dấu)* | **CÁN BỘ HƯỚNG DẪN**  *(ký và ghi rõ họ tên)* |

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**Điểm thực tập: ……… / 10**

|  |  |
| --- | --- |
|  | **GIẢNG VIÊN HƯỚNG DẪN**  *(ký và ghi rõ họ tên)* |

# LỜI CẢM ƠN

Để có thể hoàn thành đợt thực tập lần này, tôi xin chân thành cảm ơn đến công ty Velox Tech đã cho tôi có cơ hội được thực hiện việc thiết kế một website hoàn chỉnh, vừa được hiện thực hóa các kiến thức đã được học từ ghế Nhà trường, vừa được tiếp cận với các yếu tố kỹ thuật của công nghệ mới.

Xin gửi lời cảm ơn sâu sắc đến các thầy cô giáo của Khoa Công nghệ Thông tin đã truyền đạt cho em những kiến thức để có thể thực hiện được những yêu cầu của đợt thực tập lần này.

Xin gửi lời cảm ơn chân thành đến các anh chị trong công ty đã phát hiện ra những khiếm khuyết của sản phẩm để có thể có những phiên bản sau tốt hơn phiên bản trước.

Xin được cảm ơn đặc biệt đến ông Jon Inge Sun đã cho tôi có thời gian thực sự ý nghĩa tại đây, được học hỏi, được trải nghiệm công việc thực tế, được tận hưởng văn hóa công ty trong suốt thời gian qua.

**MỤC LỤC**

**Đề mục Trang**

[LỜI CẢM ƠN iii](#_Toc62067326)

[DANH MỤC HÌNH vi](#_Toc62067327)

[DANH MỤC BẢNG vii](#_Toc62067328)

[DANH MỤC CÁC TỪ VIẾT TẮT viii](#_Toc62067329)

[CHƯƠNG 1: TỔNG QUAN 1](#_Toc62067330)

[1.1. Tổng quan về đề tài 1](#_Toc62067331)

[1.2. Yêu cầu của đề tài 1](#_Toc62067332)

[1.3. Thời gian thực hiện 2](#_Toc62067333)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 4](#_Toc62067334)

[2.1. Quy trình xây dựng hệ thống 4](#_Toc62067335)

[2.1.1. Giai đoạn 1: Khảo sát dự án 4](#_Toc62067336)

[2.1.2. Giai đoạn 2: Phân tích hệ thống 4](#_Toc62067337)

[2.1.3. Giai đoạn 3: Thiết kế 5](#_Toc62067338)

[2.1.4. Giai đoạn 4: Thực hiện 5](#_Toc62067339)

[2.1.5. Giai đoạn 5: Kiểm thử 5](#_Toc62067340)

[2.1.6. Giai đoạn 6: Triển khai và bảo trì 5](#_Toc62067341)

[2.2. Tìm hiểu về Client - Server, Client - Side rendering 6](#_Toc62067342)

[2.2.1. Client - Server 6](#_Toc62067343)

[2.2.2. Client - Side rendering 7](#_Toc62067344)

[2.3. Mô hình Model - View - Controller (MVC) 8](#_Toc62067345)

[2.4. Giao diện lập trình ứng dụng (RESTful API) 9](#_Toc62067346)

[2.4.1. Khái niệm RESTful API 9](#_Toc62067347)

[2.4.2. Cách thức hoạt động của giao thức RESTful API 10](#_Toc62067348)

[2.5. Hệ Quản trị cơ sở dữ liệu MySQL và ngôn ngữ lập trình Node JS, React JS 11](#_Toc62067349)

[2.5.1. MySQL là gì 11](#_Toc62067350)

[2.5.2. Cách thức hoạt động của MySQL 13](#_Toc62067351)

[2.5.3. Giới thiệu về Node JS 15](#_Toc62067352)

[2.5.4. Giới thiệu NPM 18](#_Toc62067353)

[2.5.5. Giới thiệu về Express Framwork 18](#_Toc62067354)

[2.5.6. Giới thiệu React JS 19](#_Toc62067355)

[2.5.7. Giới thiệu Axios 20](#_Toc62067356)

[2.6. Phần giao diện 20](#_Toc62067357)

[2.6.1. HTML 20](#_Toc62067358)

[2.6.2. CSS 21](#_Toc62067359)

[2.6.3. Giới thiệu về Javascript 21](#_Toc62067360)

[2.6.4. Giới thiệu về ES6 21](#_Toc62067361)

[2.5.3. Các chức năng mới của ES6 21](#_Toc62067362)

[2.6.5. Boostrap 4 22](#_Toc62067363)

[2.7. Phần mềm lập trình 22](#_Toc62067364)

[2.7.1. Phần mềm Visual Sudio Code 22](#_Toc62067365)

[2.7.2. Một số tính năng của Visual Studio Code 22](#_Toc62067366)

[2.8. Hệ thống quản lý souce code 24](#_Toc62067367)

[2.8.1. Giới thiệu GIT 24](#_Toc62067368)

[CHƯƠNG III: KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG 24](#_Toc62067369)

[3.1. Khảo sát hiện trạng 24](#_Toc62067370)

[3.1.1. Hiện trạng của Velox Tech 24](#_Toc62067371)

[3.2. Xác định và mô tả yêu cầu chức năng 26](#_Toc62067372)

[3.2.1. Sơ đồ chức năng 26](#_Toc62067373)

[3.3. Các mô hình dữ liệu 28](#_Toc62067374)

[3.3.1. Mô hình logic dữ liệu 28](#_Toc62067375)

[3.3.2. Mô hình vật lý dữ liệu 34](#_Toc62067376)

[CHƯƠNG IV: CÀI ĐẶT HỆ THỐNG VÀ KẾT QUẢ ĐẠT ĐƯỢC 35](#_Toc62067377)

[4.1. Cài đặt hệ thống 35](#_Toc62067378)

[4.1.1. Cài đặt Visual Studio Code 35](#_Toc62067379)

[4.1.2. Cài đặt Boostrap 4 35](#_Toc62067380)

[4.1.2. Cài đặt Node JS 36](#_Toc62067381)

[4.1.3. Cài đặt React JS 37](#_Toc62067382)

[4.1.4. Cài đặt Axios 38](#_Toc62067383)

[4.2. Kết quả đạt được 40](#_Toc62067384)

[CHƯƠNG V: ƯU NHƯỢC ĐIỂM VÀ HƯỚNG PHÁT TRIỂN 44](#_Toc62067385)

[5.1. Ưu, nhược điểm của thực tập chuyên ngành 44](#_Toc62067386)

[5.1.1. Ưu điểm 44](#_Toc62067387)

[5.1.2. Nhược điểm 44](#_Toc62067388)

[5.2. Hướng phát triển 44](#_Toc62067389)

[KẾT LUẬN 45](#_Toc62067390)

[TÀI LIỆU THAM KHẢO 47](#_Toc62067391)

# DANH MỤC HÌNH

[Hình 2.1. Mô hình Client - Server 6](#_Toc62140786)

[Hình 2.2. Cơ chế hoạt động của Client Side Rendering 8](#_Toc62140787)

[Hình 2.3. Mô phỏng mô hình MVC 9](#_Toc62140788)

[Hình 2.4. Sơ đồ cấu trúc API 10](#_Toc62140789)

[Hình 2.5. Cấu trúc Request trong RESTfull API 11](#_Toc62140790)

[Hình 2.6. Cấu trúc Response trong API 11](#_Toc62140791)

[Hình 2.5. Giới thiệu MySQL 12](#_Toc62140792)

[Hình 2.6. Giao diện quản lý cơ sở dữ liệu của MySQL 13](#_Toc62140793)

[Hình 2.7. Cách thức hoạt động của MySQL 13](#_Toc62140794)

[Hình 2.8. Giới thiệu về Node JS 15](#_Toc62140795)

[Hình 2.9. Giới thiệu Node JS – NPM 18](#_Toc62140796)

[Hình 2.10. Giới thiệu React JS 19](#_Toc62140797)

[Hình 2.11. Giới thiệu về Boostrap 4 22](#_Toc62140798)

[Hình 2.12. Giới thiệu phần mềm lập trình 24](#_Toc62140799)

[Hình 3.1. Hình Mô hình thực thể kết hợp 28](#_Toc62140800)

[Hình 3.2. Mô hình vật lý 34](#_Toc62140801)

[Hình 4.1. Cài đặt extension cho phần mềm Visual Studio Code 35](#_Toc62140802)

[Hình 4.2. Cài đặt Node JS - NPM 36](#_Toc62140803)

[Hình 4.3. Trang chủ website khách hàng Velox Tech 40](#_Toc62140804)

[Hình 4.4. Trang thông tin liên hệ của công ty Velox Tech 41](#_Toc62140805)

[Hình 4.5. Trang thông tin liên hệ giử ý kiến người dùng 41](#_Toc62140806)

[Hình 4.6. Trang tuyển dụng công ty Velox Tech 42](#_Toc62140807)

[Hình 4.7. Trang tuyển dụng sau khi xem thêm 42](#_Toc62140808)

[Hình 4.8. Trang tuyển dụng sau khi nhấn nút ứng tuyển 43](#_Toc62140809)

[Hình 4.9. Trang giới thiệu về công ty Velox Tech 43](#_Toc62140810)

[Hình 4.10. Trang đăng nhập vào hệ thống web nội bộ 43](#_Toc62140811)

[Hình 4.11. Trang lịch ngày nghỉ của nhân viên 44](#_Toc62140812)

[Hình 4.12. Trang đổi mật khẩu người dùng 44](#_Toc62140813)

[Hình 4.13. Trang quản lý công việc của nhân viên 44](#_Toc62140814)

[Hình 4.13. Trang đăng và chỉnh sửa tuyển dụng 45](#_Toc62140815)

# DANH MỤC BẢNG

[Bảng 3.1. Bảng dữ liệu nhân viên (Employee) 29](#_Toc62116302)

[Bảng 3.2. Bảng dữ liệu tài khoản người dùng (User) 29](#_Toc62116303)

[Bảng 3.3. Bảng dữ liệu kế hoạch (Plan) 30](#_Toc62116304)

[Bảng 3.4. Bảng dữ liệu hoạt động (Activity) 30](#_Toc62116305)

[Bảng 3.5. Bảng dữ liệu dự án (Project) 31](#_Toc62116306)

[Bảng 3.6. Bảng dữ liệu lịnh trình (Todo) 31](#_Toc62116307)

[Bảng 3.7. Bảng dữ liệu cột mốc (Milestone) 32](#_Toc62116308)

[Bảng 3.8. Bảng dữ liệu vòng lặp dự án (Iteration) 32](#_Toc62116309)

[Bảng 3.9. Bảng dữ liệu tuyển dụng công việc (Job) 32](#_Toc62116310)

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Thuật ngữ** | **Ý nghĩa** |
| Client | Máy trạm, máy con |
| Server | Máy chủ |
| Client - Server | Mô hình máy khách - máy chủ |
| Client - Side rendering | Kết xuất phía máy chủ |
| CNTT | Công nghệ Thông tin |
| TNHH | Trách nhiệm hữu hạn |
| HTML | Ngôn ngữ đánh dấu siêu văn bản |
| CSS | Ngôn ngữ định kiểu theo tầng |
| SEO | Tối ưu hoá công cụ tìm kiếm |
| SPA | Hiển thị nội dung trong trình duyệt bằng cách sử dụng JavaScript |
| NPM | Trình quản lý các thư viện lập trình Javascript |
| MVC | Model-View-Controller |
| API | Giao diện lập trình ứng dụng |
| GUI | Giao diện đồ họa người dùng |
| ES6 | ECMAScript 6 |
| VCS | hệ thống kiểm soát các phiên bản phân tán mã nguồn mở |

# CHƯƠNG 1: TỔNG QUAN

Công ty TNHH Velox Tech ra đời năm 2018 tại Na Uy là một công ty hoạt động trong lĩnh vực nghiên cứu và phát triển, tư vấn về các giải pháp công nghệ đặc biệt là trong ngành thủy sản. Vì Na Uy là một trong những nước có ngành công nghiệp thủy sản lớn trên thế giới, song với việc nuôi trồng đánh bắt thủy – hải sản, nhiều công ty đã và đang đưa công nghệ vào việc nuôi, trong đó có công ty Velox Tech. Với sự phát triển nhanh chóng và mong muốn được mở rộng quy mô, thế nên Việt Nam là một trong sự lựa chọn của Velox tech để nghiên cứu và phát triển, vào năm 2020 văn phòng chi nhánh thành lập tại Việt Nam là 48C Mạc Đĩnh Chi, Phước Tiến, Thành Phố Nha Trang, Tỉnh Khánh Hòa.

Dẫu công ty Velox Tech hoạt động được hai năm kể từ năm 2018 đến nay nhưng vẫn chưa có website quản lý công việc cho nhân viên. Do đó, khi triển khai các công việc vẫn phải thực hiện thủ công bằng bảng, ghi chép hay sử dụng một số nền tảng trên di động. Đi kèm với đó là việc quảng bá hình ảnh của công ty chưa được tốt nên việc tìm kiếm khách hàng đối với công ty vẫn còn khó khăn. Vì thế trong đợt thực tập này, tôi được công ty yêu cầu thực hiện nhiệm vụ thiết kế website cho công ty bao gồm 2 chức năng: (1) website giới thiệu về công ty và các hoạt động tuyển dụng, buôn bán, tư vấn, chăm sóc khách hàng; (2) website nội bộ quản lý công việc của nhân viên.

## 1.1. Tổng quan về đề tài

Xây dựng website quản lý nội bộ và website giới thiệu công ty TNHH Velox Tech.

Mục tiêu: Giúp công ty quản lý công việc nhanh, dễ dàng trên môi trường Internet. Cụ thể, giúp khách hàng tiềm năng tìm đến công ty dễ dàng hơn. Việc đăng tin tuyển dụng của công ty cũng thuận tiện hơn. Ngoài ra, với website nội bộ, các yêu cầu phân công công việc, báo cáo tiến độ công việc cũng được thực hiện đồng bộ và kịp thời.

Tính khả thi: Với những kiến thức đã học được tại trường, kinh nghiệm thực hiện các dự án nhỏ thông qua các môn học và việc triển khai website cho một vài khách hàng trước đó đã giúp dự án này trở nên dễ dàng hơn. Hơn nữa, trong quá trình thực hiện đề tài này còn có sự trợ giúp không nhỏ của các anh chị và các bạn trong công ty Velox, nhờ đó mà cả 2 trang web (giới thiệu và nội bộ) đã được hoàn thành trước thời hạn.

## 1.2. Yêu cầu của đề tài

Tìm hiểu về ngôn ngữ lập trình Node JS, React JS, Pug và hệ quản trị cơ sở dữ liệu MySQL phù hợp với công nghệ hiện nay, có thể dễ dàng quản lý source code theo mô hình MVC cộng với sự hổ trợ rất nhiều frameword, library có sẵn từ ngôn ngữ các nhà phát triển trước đó giúp việc lập trình trở nên nhanh chống, dễ dàng, dễ nhìn, dễ sửa chữa và nâng cấp.

1. Website giới thiệu công ty TNHH Velox Tech đáp ứng yêu cầu:

* Thể hiện đầy đủ thông tin của công ty lên trang web.
* Giao diện đơn giản, màu sắc phù hợp dễ nhìn, đúng kích thước, thân thiện với thiết bị di động.
* Khách hàng gửi email để được tư vấn.
* Trang tuyển dụng việc làm của công ty.

1. Website nội bộ của công ty TNHH Velox Tech đáp ứng yêu cầu:

*Đối với nhân viên (EMPLOYEE)*

* + - Xem, thêm, xóa, sửa công việc hằng ngày.
    - Thêm, điều chỉnh tiến độ công việc.
    - Xem thông báo quan trọng trong tháng.
    - Xem thông tin cá nhân.
    - Đổi mật khẩu cá nhân.

*Đối với đơn vị tuyển dụng (HR)*

* + - Có những chức năng của nhân viên.
    - Đăng bài tuyển dụng việc làm.
    - Sửa thông tin nhân viên.

*Đối với người quản trị (ADMIN)*

* + - Có tất cả các chức năng của nhân viên.
    - Xét duyệt công việc của nhân viên.
    - Đăng bài viết, xóa sửa thông báo quan trọng trong tháng.

## 1.3. Thời gian thực hiện

|  |  |  |
| --- | --- | --- |
| **KẾ HOẠCH THỰC TẬP TẠI CÔNG TY VELOX TECH 7/12/2020-16/1/2021** | | |
| **TUẦN** | **CÔNG VIỆC** | **MỤC ĐÍCH** |
| 1 | + Gặp Jon Inge Sun  + Tham quan tất cả phòng làm việc của công ty | + Làm quen với cách làm việc của công ty.  + Tìm hiểu về công ty.  + Yêu cầu công việc phải làm. |
| 2 | +Gặp GVHD Đoàn Vũ Thịnh  +Nhận tài liệu | Chọn lựa đề tài thực tập phù hợp. |
| 3 | Làm theo GVHD và trong tài liệu | Qua trình thực tập được tốt hơn. |
| 4 | Cài đặt và chỉnh sửa Module. Thêm sản phẩm vào website. | Hoàn thiện các chức năng của Webiste. |
| 5 | Cài đặt và chỉnh sửa Module. Thêm sản phẩm vào website. | Hoàn thiện các chức năng của Webiste |
| 6 | +Làm bài word chuẩn bị cáo báo cho công ty.  +Nộp báo cáo cho Khoa và GVHD để đánh giá. | Đánh giá kết quả trong quá trình thực tập. |

**1.4. Cấu trúc của báo cáo**

Báo cáo thực tập gồm 5 chương:

**Chương 1:** Tổng quan

**Chương 2:** Cơ sở lý thuyết

2.1. Quy trình xây dựng hệ thống

2.2. Tìm hiểu về Client - Server, Client - Side rendering

2.3. Mô hình Model - View - Controller (MVC)

2.4. Giao diện lập trình ứng dụng (RESTful API)

2.5. Hệ quản trị cơ sở dữ liệu MySQL và nền tảng Node JS, React JS

2.6. Phần giao diện

**Chương 3:** Khảo sát và phân tích hệ thống

3.1. Khảo sát hiện trạng

3.2. Xác định và mô tả yêu cầu chức năng

3.3. Các mô hình dữ liệu

**Chương 4:** Cài đặt hệ thống và kết quả đạt được

4.1. Trang web dành cho khách hàng

4.2. Trang web dành cho nội bộ công ty

**Chương 5**: Kết luận

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. Quy trình xây dựng hệ thống

Quy trình phát triển mội hệ thống gồm 6 giai đoạn:

### 2.1.1. Giai đoạn 1: Khảo sát dự án

Khảo sát hiện trạng là giai đoạn đầu tiên cũng là giai đoạn quan trọng trong quá trình phát triển của một website. Nhiệm vụ chính là tìm hiểm, thu thập thông tin cần thiết để chuẩn bị việc giải quyết các vấn đề, yêu cầu đặt ra của dự án. Giai đoạn khảo sát được chia làm 2 bước:

**Bước 1:** Khảo sát công ty

Khảo sát sơ bộ: tìm hiểu các yếu tố cơ bản (tổ chức, con người) tạo tiền đề để phát triển hệ thống thông tin phù hợp với dự án và doanh nghiệp.

Khảo sát chi tiết: thu thập thông tin chi tiết của hệ thống (chức năng xử lí, thông tin nhập, xuất, ràng buộc, giao diện cơ bản, nghiệp vụ) phục vụ thiết kế.

**Bước 2:** Đặt ra các vấn đề quan trọng cần giải quyết như:

* + Thông tin cần đưa vào hệ thống như thế nào?
  + Ràng buộc giữa các đối tượng trong quan hệ cần được xây dựng ra sao?
  + Dữ liệu đầu vào và đâu ra khác nhau ở điểm nào?
  + Chức năng và quy trình xử lý của hệ thống cần đảm bảo những yếu tố nào?
  + Cần sử dụng giải pháp nào? tính khả thi cho từng giải pháp ra sao?

### 2.1.2. Giai đoạn 2: Phân tích hệ thống

Xác định các thông tin và chức năng xử lý của hệ thống. Bao gồm:

Xác định yêu cầu hệ thống thông tin: 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 mở rộng trong tương lai.

Phân tích và đặc tả mô hình phân rã chức năng và mô hình luồng dữ liệu.

Phân tích bảng dữ liệu. Đưa vào hệ thống những bảng dữ liệu (Data table) gồm các trường dữ liệu (Data field), khóa chính (Primary key), khóa ngoại (Foreign key) cũng như mối quan hệ giữa các bảng dữ liệu (Relationship) và ràng buộc dữ liệu (Constraint) dữ liệu cần thiết. Giai đoạn này sẽ đặc tả sơ bộ các bảng dữ liệu trên giấy để có cái nhìn khách quan. Qua đó, xác định các giải pháp tốt nhất cho hệ thống, đảm bảo hệ thống đúng với yêu cầu của công ty đặt ra và những khảo sát trước đó.

### 2.1.3. Giai đoạn 3: Thiết kế

Thông qua thông tin được thu thập từ quá trình khảo sát và phân tích, từ đó chuyển hóa thành website. Cụ thể:

**Bước 1:** Thiết kế tổng thể

Dựa trên cơ sở các bảng dữ liệu đã khảo sát và phân tích ở giai đoạn phân tích hệ thống trên. Từ đó thiết kế mô hình phân rã chức năng, thiết kế dữ liệu mô phỏng.

**Bước 2:** Thiết kế chi tiết

Thiết kế cơ sở dữ liệu (Database) với mô hình vật lý hoàn chỉnh ở môi trường phát triển được, kết xuất thành thư mục với định dạng SQL.

Thiết kế thủ tục, hàm thu thập, xử lý thông tin chuẩn theo đúng nghiệp vụ.

Thiết kế giao diện website đảm bảo phù hợp với yêu cầu, dễ thao tác.

Thiết kế các tính năng đảm bảo tính logic trong quá trình nhập, xuất dữ liệu.

Như vậy, thiết kế là việc áp dụng các công cụ, phương pháp và thủ tục để tạo ra mô hình hệ thống cần sử dụng. Sản phẩm cuối cùng của giai đoạn thiết kế là đặc tả hệ thống ở dạng thực thể sao cho người lập trình viên giao tiếp với phần cứng máy tính dễ dàng chuyển thành chương trình cấu trúc hệ thống.

### 2.1.4. Giai đoạn 4: Thực hiện

Xây dựng hệ thống theo các thiết kế ở giai đoạn đã xác định. Giai đoạn này gồm các công việc sau:

Lựa chọn hệ quản trị cơ sở dữ liệu và cài đặt cơ sở dữ liệu.

Lựa chọn công cụ lập trình dễ xây dựng các module chương trình của hệ thống.

Viết tài liệu hướng dẫn sử dụng, tài liệu kỹ thuật cho dự án.

### 2.1.5. Giai đoạn 5: Kiểm thử

Thiết kế các test case dựa trên các ràng buộc, dữ liệu vào - ra, các chức năng của hệ thống; Kiểm tra các module chức năng của hệ thống thông tin, chuyển các thuyết kế thành các chương trình; Thử nghiệm trên môi trường phát triển; Cuối cùng là khắc phục các lỗi của hệ thống.

### 2.1.6. Giai đoạn 6: Triển khai và bảo trì

Thuê, thiết lập hosting.

Cài đặt phần mềm nếu có.

Chuyển đổi hệ thống từ môi trường phát triển sang môi trường thực tế trên Internet.

Kiểm tra phát hiện sai sót, khuyết điểm của hệ thống.

Sửa chữa, khắc phục đến hoàn thiện.

Đào tạo và hướng dẫn sử dụng.

Cải tiến và chỉnh sửa theo yêu cầu.

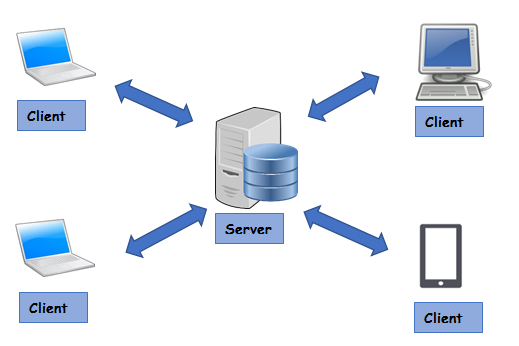
Bảo trì và chỉnh sửa hệ thống.

Nâng cấp chương trình khi có phiên bản mới.

## 2.2. Tìm hiểu về Client - Server, Client - Side rendering

### 2.2.1. Client - Server

Mô hình client - server là một mô hình phổ biến trong mạng máy tính, được ứng dụng rộng rãi và là mô hình của mọi trang web hiện có. Ý tưởng của mô hình này là máy con (máy khách) gửi một yêu cầu (request) để máy chủ (máy cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách. Máy chủ sẽ cung cấp dịch vụ cho các client khi có yêu cầu, các dịch vụ như: thao tác với cơ sở dữ liệu, in ấn, truyền nhận tập tin. Trong khi, máy khách là các máy tính như desktop, laptop, điện thoại. Trong mô hình này, các máy khách gửi yêu cầu lên phía máy chủ để thực hiện các yêu cầu như thao tác dữ liệu, in ấn, gửi mail.



Hình 2.1. Mô hình Client - Server

Mô hình Client - Server giúp tối ưu hóa việc lưu trữ và chia sẻ dữ liệu bằng cách tập trung chúng vào một máy chủ duy nhất thay vì phân tán cùng một nội dung trên nhiều máy khách khác nhau.

### 2.2.2. Client - Side rendering

Trước đây, chỉ có 1 cách duy nhất để thực hiện các yêu cầu HTML từ trình duyệt được thực hiện là sử dụng server - side rendering. Client tải các trang .html trên máy chủ bằng cách gửi yêu cầu lên máy chủ, sau đó máy chủ sẽ xử lý các yêu cầu của client và trả về trên trình duyệt của người dùng. Server - side đảm nhiệm hầu như mọi công việc lớn nhỏ. May thay, vào thời kỳ đó hầu hết các trang web chủ yếu chỉ để hiển thị hình ảnh tĩnh và văn bản, rất ít các tương tác với người dùng. Sau một thời gian dài, nhận thấy việc sử dụng server - side đảm nhiệm quá nhiều nhiệm vụ dẫn đến nhiều lúc server bị quá thời gian xử lý (downtime), ảnh hưởng nhiều đến các website khác. Hơn nữa, các website hiện nay phải thực hiện thêm nhiều chức năng tương tác với người dùng. Vì vậy, việc rendering từ phía máy chủ được giảm tối đa và phương pháp rendering các trang trực tiếp từ phía client trở nên phổ biến với việc sử dụng các thư viện JavaScript.

Cách thức hoạt động của Server - Side rendering: Bất cứ khi nào máy client truy cập vào website, trình duyệt sẽ gửi yêu cầu lên máy chủ chứa nội dung của trang web. Thông thường, mỗi yêu cầu thường chỉ mất vài mili giây, nhưng thực tế lại phụ thuộc khá nhiều vào các yếu tố sau đây:

Tốc độ đường truyền Internet của người dùng?

Vị trí máy chủ của trang website?

Có bao nhiêu người truy cập cùng một thời điểm bạn gửi yêu cầu?

Cách tối ưu trang của website đó như thế nào?

Khi yêu cầu được hoàn tất, HTML sẽ được trả về cho trình duyệt để hiển thị lên màn hình của người dùng. Và cứ như thế, khi chuyển sang một liên kết (trang) khác thì trình duyệt sẽ lại tiếp tục gửi một yêu cầu mới đến server.

**Ưu điểm**

Các công cụ tìm kiếm có thể thu thập dữ liệu một cách dễ dàng.

Tải trang ban đầu nhanh hơn.

Là sự lựa chọn tối ưu cho các website tĩnh.

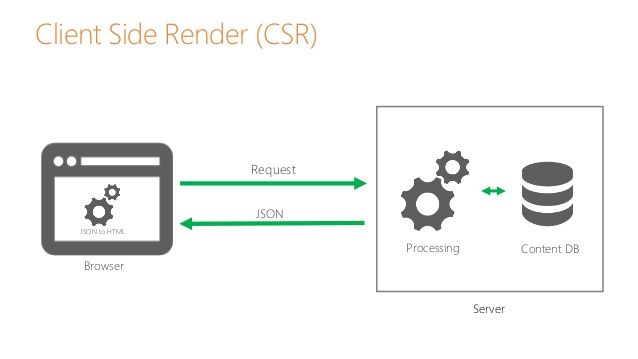
**Nhược điểm**

Thường xuyên phải gửi yêu cầu lên máy chủ .

Hiển thị trang chậm.

Tải lại toàn bộ trang.

Cách hoạt động của Client - Side rendering: Cùng với phát triển của javascript và ajax, cơ chế client - side rendering được sử dụng để xây dựng các ứng dụng dưới dạng SPA. Vì vậy, thay vì nhận được tất cả nội dung chính từ HTML, thì người dùng sẽ nhận được tài liệu HTML là khung của website kèm theo một tệp JavaScript, từ đó phần nội dung của trang web sẽ được dựng trở lại trên máy client. Ví dụ, website *instagram.com* của *Facebook* xây dựng dựa trên ReactJS của chính nó.



Hình 2.2. Cơ chế hoạt động của Client Side Rendering

**Ưu điểm**

Tương tác người dùng tốt.

Hiển thị trang nhanh sau khi tải ban đầu.

Nhiều thư viện JavaScript mạnh mẽ hỗ trợ.

**Nhược điểm**

Điểm SEO thấp nếu không thực hiện các tùy chỉnh.

Tải ban đầu cần nhiều thời gian.

Hầu hết thường phải sử dụng các thư viện bên thứ 3.

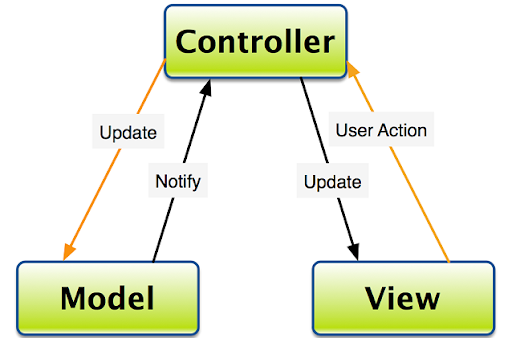
## 2.3. Mô hình Model - View - Controller (MVC)

MVC là từ viết tắt bởi 3 từ Model - View - Controller. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình mã nguồn (source code) 3 thành phần tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình.

**Model (M):** Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML. Model thể hiện rõ các thao tác với cơ sở dữ liệu như: xem, truy xuất, xử lý dữ liệu…

**View (V):** Đây là phần giao diện (theme) dành cho người sử dụng. Người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website. Thông thường, các website sử dụng MVC View như một phần của hệ thống, nơi các thành phần HTML được tạo ra. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller.

**Controller (C):** Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Từ đó, Controller đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với 2 thành phần Model và View.



Hình 2.3. Mô phỏng mô hình MVC

Trong hình trên, có thể thấy:

User tương tác với View, bằng cách click vào button, user gửi yêu cầu đi.

Controller nhận và điều hướng chúng đến đúng phương thức xử lý ở Model.

Model nhận thông tin và thực thi các yêu cầu.

Khi Model hoàn tất việc xử lý, View sẽ nhận kết quả từ Model và hiển thị lại cho người dùng.

**Ưu điểm:** Do được chia thành các thành phần độc lập việc phát triển ứng dụng nhanh chóng, đơn giản, dễ nâng cấp, bảo trì

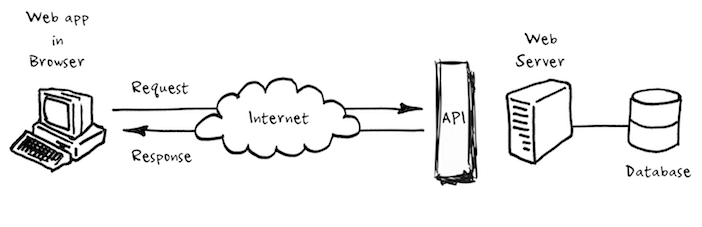
**Nhược điểm:** Đối với dự án nhỏ việc áp dụng mô hình MVC gây mất thời gian trong quá trình phát triển. Mất thời gian trung chuyển dữ liệu giữa các thành phần.

## 2.4. Giao diện lập trình ứng dụng (RESTful API)

### 2.4.1. Khái niệm RESTful API

**RESTful API** là tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web, tiện cho việc quản lý các tài nguyên do chú trọng vào tài nguyên hệ thống (tệp văn bản, hình ảnh, âm thanh, video, hoặc dữ liệu động).

**API** là viết tắt của Application Programming Interface (giao diện lập trình ứng dụng), là phần mềm trung gian cho phép 2 ứng dụng giao tiếp với nhau. Mỗi khi sử dụng các ứng dụng như Facebook, hoặc các ứng dụng gửi tin nhắn tức thời hoặc kiểm tra thời tiết đồng nghĩa với việc sử dụng API. Với API, các lập trình viên có thể tiếp cận, truy xuất dữ liệu từ máy chủ và thể hiện chúng trên ứng dụng phần mềm hoặc website của mình một cách dễ dàng hơn. Tính tới nay, API đã phát triển với nhiều loại ứng dụng và phần mềm khác nhau. Thế hệ mới nhất của web/ app API có thể ứng dụng được ở mọi hệ thống từ cơ sở dữ liệu, hệ điều hành, hệ thống nền web, thư viện hay thậm chí là phần cứng máy tính.



Hình 2.4. Sơ đồ cấu trúc API

Trong API thì có 2 định dạng chính là XML và JSON.

### 2.4.2. Cách thức hoạt động của giao thức RESTful API

**REST** hoạt động chủ yếu dựa vào giao thức HTTP:

GET (SELECT): Trả về một Resource hoặc một danh sách Resource.

POST (CREATE): Tạo mới một Resource.

PUT (UPDATE): Cập nhật thông tin cho Resource.

DELETE (DELETE): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create (Tạo), Read (Đọc), Update (Cập nhật), Delete (Xóa).

Giao thức HTTP hoạt động dựa trên Request (Yêu cầu) và Response (phản hồi). Client gửi yêu cầu tới server, server gửi lại phản hồi cho client. API được xây dựng trên chính 2 thành phần: Request và Response.

Request chuẩn bao gồm: URL, Method, Headers, Body (Hình 2.5): Trong đó, Headers chứa các thông tin cần thiết của một request nhưng người dùng không biết sự tồn tại của nó. Body: nơi chứa thông tin của client được gửi đi.

|  |
| --- |
| URL |
| Method |
| Headers |
| Body |

Hình 2.5. Cấu trúc Request trong RESTfull API

http:example.com

POST

User-Agent

Data

Request

Response: Sau khi nhận được request từ phía client, server sẽ xử lý request đó và trả về cho client những phản hồi (response). Cấu trúc của response tương đối giống với phần request nhưng Status code sẽ được thay thế cho URL và Method. Vì vậy response có cấu trúc 3 thành phần gồm: *Status code, Header, Body*.

|  |
| --- |
| Status code |
| Headers |
| Body |

Response

200 (Success!)

Content-type: text

Data

Hình 2.6. Cấu trúc Response trong API

Status code là những con số có 3 chữ số và có duy nhất một ý nghĩa. Một số status code phổ biến như: 200 (thành công), 404 (không tìm thấy trang), 503 (dịch vụ không khả dụng). Phần headers và phần Body của response tương tự như phần header và body của request.

## 2.5. Hệ Quản trị cơ sở dữ liệu MySQL và ngôn ngữ lập trình Node JS, React JS

### 2.5.1. MySQL là gì

Công ty Thụy Điển MySQL AB phát triển MySQL vào năm 1994. Công ty công nghê Mỹ Sun Microsystem sau đó giữ quyền sở hữu MySQL sau khi mua lại MySQL vào năm 2008. Năm 2010, công ty lớn Oracle mua lại Sun Microsystem và MySQL thuộc quyển sở hữu của Oracle từ đó.

MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở (Relational Database Management System viết tắt là RDBMS có nghĩa là có nghĩa là hệ quản trị cơ sỡ dữ liệu quan hệ) hoạt động theo mô hình Client – Server. RDBMS là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng. MySQL là một trong số các phần mềm RDBMS. RDBMS và MySQL thược được cho là một vì độ phổ biến quá lớn của MySQL. Các ứng dụng web lớn như Facebook, Twitter, Youtube, Google… đề dùng MySQL cho mục đích lưu trữ dữ liệu. Kể cả khi ban đầu dùng rất hạn chế nhưng giờ đã tương thích với nhiều hạ tầng máy tính quan trọng như Linux, MacOS, Microsoft Windows và cả Ubuntu.



Hình 2.5. Giới thiệu MySQL

*(Nguồn:* [*https://blog.tinohost.com/mysql-la-gi/*](https://blog.tinohost.com/mysql-la-gi/)*)*

MySQL là ngôn ngữ phổ biến nhất để thêm, truy cập và quản lý nội dung trong cơ sở dữ liệu.

MySQL là một RDBMS nhanh và dễ dàng để sử dụng. MySQL đang được sử dụng cho nhiều công việc kinh doanh từ lớn tới nhỏ. MySQL được phát triển, được công bố, được hỗ trợ bởi MySQL AB, là một công ty của Thụy Điển. MySQL trở thành khá phổ biến vì nhiều lý do:

+ MySQL là mã ngồn mở. Vì thế, hoàn toàn miễn phí khi sử dụng.

+ MySQL là một chương trình rất mạnh mẽ.

+ MySQL sử dụng một Form chuẩn của ngôn ngữ dữ liệu nổi tiếng là SQL.

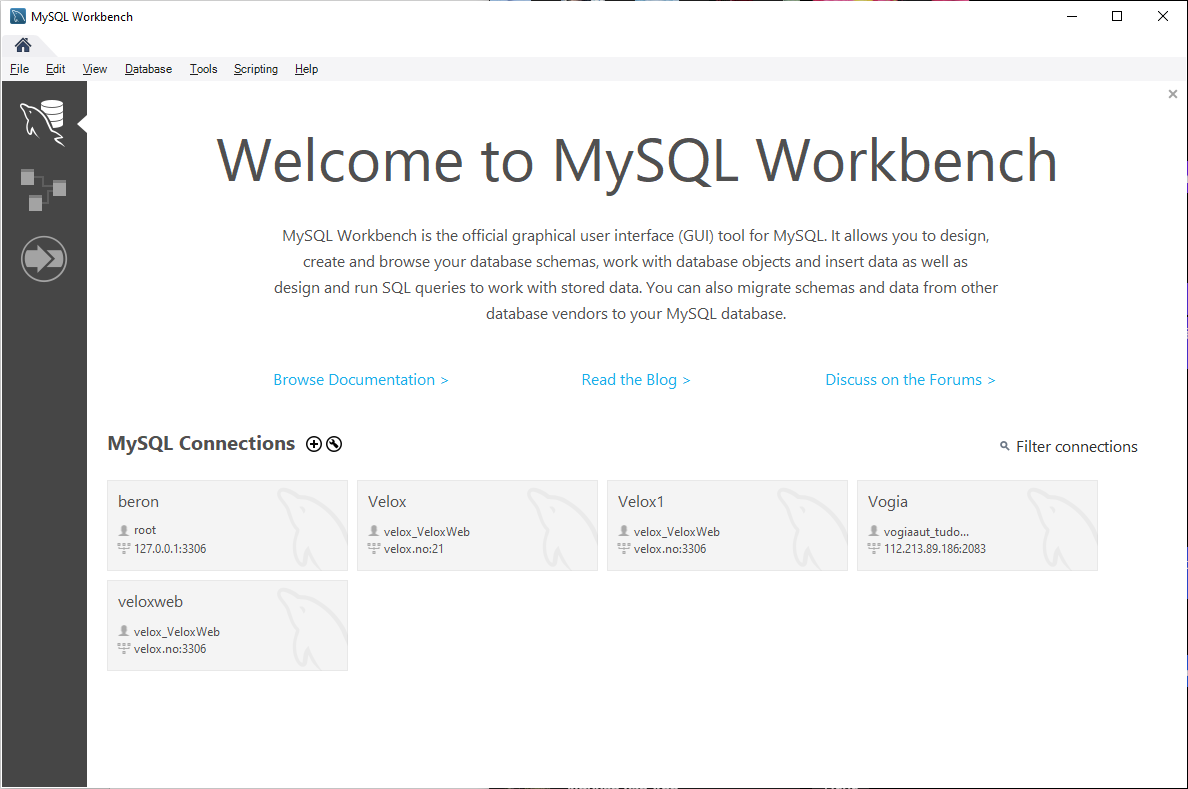
+ MySQL làm việc trên nhiều Hệ điều hành và với nhiều ngôn ngữ như PHP, PERL, C, C++, Java, …

+ MySQL làm việc nhanh và khỏe ngay cả với các tập dữ liệu lớn.

+ MySQL rất thân thiện với PHP, một ngôn ngữ rất đáng giá để tìm hiểu để phát triển Web.

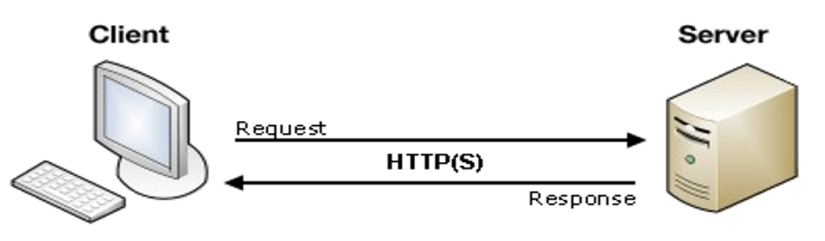
+ MySQL hỗ trợ các cơ sở dữ liệu lớn, lên tới 50 triệu hàng hoặc nhiều hơn nữa trong một bảng. Kích cỡ file mặc định được giới hạn cho một bảng là 4 GB, có thể tăng kích cỡ này (nếu hệ điều hành có thể xử lý nó) để đạt tới giới hạn lý thuyết là 8 TB.

+ MySQL là có thể điều chỉnh. Giấy phép GPL mã nguồn mở cho phép lập trình viên sửa đổi phần mềm MySQL để phù hợp với môi trường cụ thể.



Hình 2.6. Giao diện quản lý cơ sở dữ liệu của MySQL

### 2.5.2. Cách thức hoạt động của MySQL



Hình 2.7. Cách thức hoạt động của MySQL

*(Nguồn:* [*https://www.hostinger.vn/huong-dan/mysql-la-gi/*](https://www.hostinger.vn/huong-dan/mysql-la-gi/) *)*

Mô hình Client – Server là mô hình giúp các máy tính giao tiếp truyền tải dữ liệu cho nhau. Nhắc đến Client với Server thì đây là 2 từ khóa khá phổ biến trong mảng lập trình web. Client và Server về bản chất là 2 máy tính giao tiếp và truyền tải dữ liệu cho nhau.

Máy tính đóng vai trò là máy khách – Client: Với vai trò là máy khách, chúng sẽ không cung cấp tài nguyên đến các máy tính khác mà chỉ sử dụng tài nguyên được cung cấp từ máy chủ. Một client trong mô hình này có thể là một server cho mô hình khác, tùy thuộc vào nhu cầu sử dụng của người dùng.

Máy tính đóng vai trò là máy chủ – Server: Là máy tính có khả năng cung cấp tài nguyên và các dịch vụ đến các máy khách khác trong hệ thống mạng. Server đóng vai trò hỗ trợ cho các hoạt động trên máy khách client diễn ra hiệu quả hơn.

Nguyên tắc hoạt động của Mô hình Client – Server là server chấp nhận tất cả các yêu cầu hợp lệ từ mọi nơi khác nhau trên Internet, sau đó trả kết quả về máy tính đã gửi yêu cầu đó.

Máy tính được coi là máy khách khi làm nhiệm vụ gửi yêu cầu đến các máy chủ và đợi câu trả lời được gửi về. Để máy khách và máy chủ có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn nhất định, và chuẩn đó được gọi là giao thức. Một số giao thức được sử dụng phổ biến hiện nay như: HTTPS, TCP/IP, FTP,…

Nếu máy khách muốn lấy được thông tin từ máy chủ, thì phải tuân theo một giao thức mà máy chủ đó đưa ra. Nếu yêu cầu đó được chấp nhận thì máy chủ sẽ thu thập thông tin và trả về kết quả cho máy khách yêu cầu. Bởi vì Server – máy chủ luôn luôn trong trạng thái sẵn sàng để nhận request từ Client nên chỉ cần client gửi yêu cầu tín hiệu và chấp nhận yêu cầu đó thì Server sẽ trả kết quả về phía Client trong thời gian ngắn nhất.

Từ máy client, việc chọn GUI MySQL khá quan trọng. Nếu như GUI nhẹ thì việc thao tác quản lý data cũng sẽ dễ dàng và nha chóng hơn. MySQL GUI phổ biến nhất là MySQL WordBench, SequePro, DBVisualizer và Navicat DB admin tool. Một vài trong số GUI được nêu ở trên là miễn phí, một vài bản thương mại, một vài bản chỉ chạy trên MacOS, một vài ứng dụng chạy được hết trên các hệ điều hành phổ biến. Client nên chọn GUI tùy vào nhu cầu của công ty.

Các nhánh của MySQL bao gồm:

+ Drijection: một hệ thống quản lý cơ sở dữ liệu ngồn mở nhẹ được phát triển dựa trên MySQL 6.0

+ MariaDB: một sự thay thế phổ biến do cộng đồng phát triển cho MySQL sử dụng các lệnh và API MySQL.

+ Percona Server với XtraDB: một phiên bản nâng cao của MySQL được biết đến với khả năng mở rộng theo chiều ngang.

**Ưu điểm**

Giúp làm việc trên bất kì một máy tính nào có hỗ trợ giao thức truyền thông. Giao thức chuẩn này cũng giúp các nhà sản xuất tích hợp lên nhiều sản phẩm khác nhau mà không gặp phải khó khăn gì.

Có thể có nhiều server cùng làm một dịch vụ, chúng có thể nằm trên nhiều máy tính hoặc một máy tính.

Chỉ mang đặc điểm của phần mềm mà không hề liên quan đến phần cứng, ngoài yêu cầu duy nhất là server phải có cấu hình cao hơn các client.

Hỗ trợ người dùng nhiều dịch vụ đa dạng và sự tiện dụng bởi khả năng truy cập từ xa.

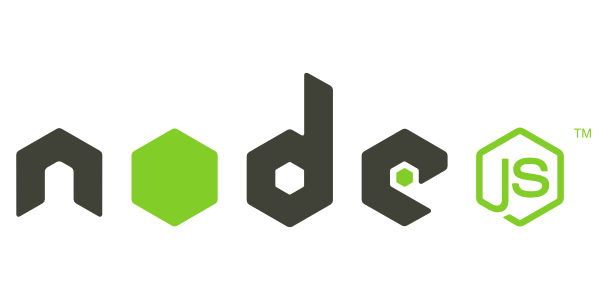
Cung cấp một nền tảng lý tưởng, cho phép cung cấp tích hợp các kỹ thuật hiện đại như mô hình thiết kế hướng đối tượng, hệ chuyên gia, hệ thông tin địa lý (GIS).

**Nhược điểm**

Vấn đề bảo mật dữ liệu thông tin đôi khi còn chưa được an toàn, do phải trao đổi dữ liệu giữa 2 máy tính khác nhau ở 2 khu vực địa lý cách xa nhau. Và đây cũng nhược điểm duy nhất của mô hình này.

Tuy nhiên vấn đề này thì có một số giao thức đã hỗ trợ bảo mật dữ liệu khi truyền tải. Giao thức được sử dụng phổ biến như HTTPS.

### 2.5.3. Giới thiệu về Node JS



Hình 2.8. Giới thiệu về Node JS

*(Nguồn:* [*https://nodejs.org/en/about/resources/*](https://nodejs.org/en/about/resources/)*)*

NodeJS là một nền tảng mã nguồn mở được viết bởi Ryan Dahl vào năm 2009 dựa trên Javascript engine V8 của Google. Về bản chất NodeJS là Javascript. Trước khi NodeJS ra đời, Javascript chủ yếu được sử dụng để nhúng vào HTML sử dụng trên các website cho phép người dùng tương tác với trình duyệt. Với sự ra đời của NodeJS, lập trình viên có thể dùng Javsacript để phát triển các ứng dụng trên máy chủ hoặc các ứng dụng về network. Node.js không phải là một ngôn ngữ lập trình mới, thay vào đó nó chỉ là một nền tảng máy chủ được viết dựa trên ngôn ngữ JavaScript. Cú pháp khi viết chương trình sử dụng Node.js giống với cú pháp ngôn ngữ Javascript.

NodeJS là một nền tảng dựa vào Chrome Javascript runtime để xây dựng các ứng dụng nhanh, có độ lớn. Node.js sử dụng các phần phát sinh các sự kiện (event-driven), mô hình non-blocking I/O để tạo ra các ứng dụng nhẹ và hiệu quả cho các ứng dụng về dữ liệu thời gian thực chạy trên các thiết bị phân tán.

NodeJS là một mã nguồn mở, đa nền tảng cho phát triển các ứng dụng phía Server và các ứng dụng liên quan đến mạng. Ứng dụng Node.js được viết bằng Javascript và có thể chạy trong môi trường Node.js trên hệ điều hành Window, Linux...

NodeJS cũng cung cấp các module Javascript đa dạng, có thể đơn giản hóa sự phát triển của các ứng dụng web sử dụng Node JS với các phần mở rộng.

**Ba thành phần chính trong Node JS**

- *Event Queue:* Đóng vai trò một cái kho để lưu trữ các Event (sự kiện), Event ở đây có thể hiểu là một xử lý nào đó trong chương trình, mỗi Event bao gồm thông tin để phân loại xem Event đó là gì và các dữ liệu đi kèm. Vì là cấu trúc hàng đợi nên Event Queue sẽ hoạt động theo nguyên tắc vào trước ra trước (First In First Out – FIFO). Điều này đảm bảo rằng thứ tự của các Event được lấy ra để xử lý cũng giống như các thứ tự mà chúng được đưa vào bên trong hàng đợi Event Queue.

- *Main Thread:* là Thread chính của ứng dụng NodeJS, nói đây là Thread chính vì nó sẽ là ngọn nguồn của chương trình cũng như sự kết thúc của chương trình. Thread chính này có nhiệm vụ là xử lý các tính toán khi nhận được các Event từ Event Queue. Đây cũng chính là Thread duy nhất của ứng dụng Node mà lập trình viên có quyền điều khiển. Đó cũng chính là lý do vì sao mà người ta vẫn nói ứng dụng Node là đơn tuyến (single-thread). Và vì là đơn tuyến nên khi lập trình các lập trình viên sẽ không phải đau đầu về các vấn đề xử lý đồng bộ (concurrency) giữa các thread như trong một số các nền tàng khác như Java.

- *Node API:* Là đơn vị chịu trách nhiệm xử lý các thao tác IO. Các thao tác IO sẽ được xử lý ở đây bằng cơ chế đa tuyến (multi-threading). Và mỗi một thao tác IO sau khi hoàn thành sẽ trả về kết quả dưới dạng một Event và Event này sẽ được đưa vào hàng đợi Event Queue.

Với ba thành phần trên, thì cách ứng dụng Node xử lý sẽ như sau:

+ Main thread sẽ chạy các câu lệnh xử lý tính toán được khai báo trong mã nguồn. Ở bất cứ chỗ nào có liên quan đến thao tác IO, Main thread sẽ tạo ta một lời gọi Non-Blocking sang cho Node API và sau đó tiếp tục thực hiện các lệnh khác mà không cần phải chờ thao tác IO kia hoàn thành

+ Node API khi nhận được yêu cầu từ Main Thread sẽ trực tiếp xử lý các thao tác IO bằng các Thread khác nhau. Một khi có thao tác IO hoàn thành, Node API sẽ gói kết quả dưới dạng một Event và đặt kết quả vào hàng đợi Event Queue

+ Event ở trong hàng đợi Event Queue sẽ lại lần lượt được xử lý bởi Main Thread. Trong Main Thread, đoạn mã dùng để xử lý các Event thường được khai báo ở dưới dạng các callback.

Quá trình trên cứ như vậy lặp đi lặp lại, tạo nên một vòng tuần hoàn của các Event trong ứng dụng. Việc lập trình sẽ được áp đặt theo phong cách hướng đến xử lý các sự kiện thay vì kiểu nghĩ xử lý tuần tự truyền thống. Và phong cách lập trình như vậy được đặt một cái tên là Reactive Programming.

**Ưu Điểm**

Node.js có tốc độ rất nhanh: đây là điều quan trọng nhất đối với một trang web đảm bảo có thể mở rộng nhanh chóng, đáp ứng được một lượng lớn người dùng khi trang web của bạn phát triển lên.

Node.js có thể xử lý hàng ngàn kết nối đồng thời. Bên cạnh các lợi ích về tốc độ thực thi và khả năng mở rộng.

**Nhược Điểm**

Giống như hầu hết các công nghệ mới, việc triển khai Node JS trên host không phải là điều dễ dàng. Nếu một web hosting xài chung, khôngđơn giản tải lên một ứng dụng Node.js và mong chờ nó hoạt động tốt. VPS và dedicated server là một sự lựa chọn tốt hơn có thể cài đặt Node JS.

Dễ dàng cài đặt Node JS chạy cục bộ trên máy tính sử dụng các hệ điều hành như Windows, Mac hoặc Linux và bắt đầu phát triển ứng dụng ngay lập tức – chỉ việc tải phiên bản Node.js tương ứng tại đây. Một điều quan trọng nên chú ý là Node JS không chỉ đơn giản là một sự thay thế cho Apache, các ứng dụng web đang tồn tại sẽ không có khả năng tương thích và làm việc hiệu quả với những ứng dụng phát triển từ đầu.

### 2.5.4. Giới thiệu NPM

NPM viết tắt của Node Package Manager là một công cụ (chương trình) quản lý các thư viện lập trình Javascript cho Node.js, công cụ này là thật sự cần thiết cho thế giới mã nguồn mở. Trong cộng đồng Javascript, các lập trình viên chia sẻ hàng trăm nghìn các đoạn code giúp cho các dự án mới tránh phải viết lại các thành phần cơ bản, các thư viện lập trình hay thậm chí cả các framework. Mỗi đoạn code này có thể phụ thuộc vào rất nhiều các mã nguồn mở khác, thật may mắn khi các công cụ quản lý thư viện ra đời, nếu không sẽ mất rất nhiều công sức trong việc quản lý các thư viện này.

NPM về cơ bản khá giống với Maven, sự khác biệt ở đây là NPM quản lý các thư viện Javascript trong khi đó Maven quản lý các thư viện Java.



Hình 2.9. Giới thiệu Node JS – NPM

*(Nguồn:* [*https://blog-xtraffic.pep.vn/huong-dan-cach-cai-dat-node-js-npm-tren-centos/*](https://blog-xtraffic.pep.vn/huong-dan-cach-cai-dat-node-js-npm-tren-centos/) *)*

Một phần mềm tương tự với NPM là Yarn được phát triển bởi Facebook với các tính năng vượt trội đang nổi lên như một đối thủ thay thế NPM. Trong một tình thế tương tự phần mềm quản lý thư viện Java là Gradle với tính năng vượt trội cũng đang dần thay thế Maven.

### 2.5.5. Giới thiệu về Express Framwork

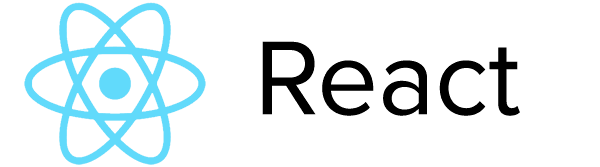
Express là một framework nhỏ và tiện ích để xây dựng các ứng dụng web, cung cấp một lượng lớn của tính năng mạnh mẽ để phát triển các ứng dụng web và mobile. Express dễ dàng để phát triển các ứng dụng nhanh dựa trên Node.js cho các ứng dụng Web. Dưới đây là các tính năng cơ bản của Express framework.

- Cho phép thiết lập các lớp trung gian để trả về các HTTP request.

- Định nghĩ bảng 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.

### 2.5.6. Giới thiệu React JS



Hình 2.10. Giới thiệu React JS

*(Nguồn:* [*http://free-php-editor.com/reactjs-la-gi/*](http://free-php-editor.com/reactjs-la-gi/)*)*

React là thư viện JavaScript phổ biến nhất để xây dựng giao diện người dùng (UI). React cho tốc độ phản hồi tuyệt vời khi user nhập liệu bằng cách sử dụng phương pháp mới để render trang web.

Năm 2013, Facebook cho ra đời một thư viện JavaScript mã nguồn mở có tên là ReactJS. Hiện nay, React JS đã được sử dụng phổ biến nhất hiện nay. ReactJS được Netflix, Airbnb, American Express, Facebook, WhatsApp, eBay, Instagram sử dụng để xây dựng website. Đa số người dùng đánh giá React JS là thư viện JavaScript phổ biến nhất để xây dựng giao diện người dùng (UI). Tốc độ của trang web được tải với tốc độ rất nhanh, khả năng mở rộng cao và đơn giản – đây chính là đặt điểm nổi bật của React JS.

Lợi ích đầu tiên mà React JS đó chính là việc tạo ra một dom ảo, đây là nơi mà các component được tồn tại trên đó. Việc tạo ra dom như vậy giúp cải thiện hiệu suất làm việc rất nhiều, khi có tính toán cần thay đổi hoặc cần cập nhật những gì lên Dom thì React JS đều tính toán trước và việc còn lại chỉ là thực hiện chúng lên Dom, làm như vậy sẽ giúp cho React JS tránh được những thao tác cần có trên một Dom mà không cần tốn thêm bất cứ chi phí nào.

Lợi ích tiếp theo mà React JS đem lại đó chính là việc viết các đoạn code JS sẽ trở nên dễ dàng hơn vì nó sử dụng một cú pháp đặc biệt đó chính là cú pháp JSX nghĩa là cú pháp này cho phép ta trộn được giữa code HTML và Javascript. Ngoài ra ta còn có thể đem đoạn code thêm vào trong hàm render mà không cần phải thực hiện việc nối chuỗi và đây được đánh giá là một trong những đặc tính thú vị của ReactJS và việc chuyển đổi các đoạn HTML thành các hàm khởi động đều được thực hiện từ bộ biến đổi chính đó chính là JSX.

Thân thiện với SEO: đây là một trong những điều đặc biệt và chỉ có mỗi React JS đây cũng chính là vấn đề lớn của các JS Frameworks vì hầu như các JS Frameworks không thân thiện với các cổ máy tìm kiếm mặc dù đã được cải thiện nhiều. Còn riêng đối với React JS thì khá tự hào vì không nằm trong nhóm không thân thiện với SEO vì dưới sự hỗ trợ của các render và trả về trình duyệt dưới dạng webpage khi mà bạn chạy React JS trên sever và các Virtual Dom. Chính vì lý do này mà React có thể đáp ứng đầy đủ được tính SEO Friendly.

### 2.5.7. Giới thiệu Axios

Axios là một HTTP client được viết dựa trên Promises được dùng để hỗ trợ cho việc xây dựng các ứng dụng API. Axios có thể được sử dụng ở cả browser hoặc Node. js. Axios có thể hoạt động trong hầu hết tất các trình duyệt web hiện đại. Người ta dùng Axios để tiến hành phương pháp **Server-side rendering** như đã giới thiệu ở phần trên.

**Các alias cho việc tạo ra các request với phương thức tương ứng**

* axios.request(config)
* axios.get(url[, config])
* axios.delete(url[, config])
* axios.head(url[, config])
* axios.options(url[, config])
* axios.post(url[, data[, config]])
* axios.put(url[, data[, config]])
* axios.patch(url[, data[, config]])

## 2.6. Phần giao diện

Phần giao diện là để người dùng có thể tương tác với trang web trên trình duyệt thông qua internet. Giao diện website bao gồm tất cả những gì xuất hiện trên website bao gồm hình ảnh, thông tin, video, các điều hướng người dùng trên website, liên kết trên web hay tất cả những gì người dùng nhìn thấy, tương tác trên website (truy cập danh mục, đặt hàng, chat online).

### 2.6.1. HTML

HTML (Hypertext Markup Language) dùng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, links, blockquotes, vâng vâng. HTML không phải là ngôn ngữ lập trình, đồng nghĩa với việc nó không thể tạo ra các chức năng “động” được.

### 2.6.2. CSS

CSS là chữ viết tắt của Cascading Style Sheets là một ngôn ngữ được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu (ví dụ như HTML). Hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,… thì CSS sẽ có thể thêm giao diện đẹp vào các phần tử HTML đó như đổi màu sắc trang, đổi màu chữ, thay đổi cấu trúc.

### 2.6.3. Giới thiệu về Javascript

JavaScript là một ngôn ngữ lập trình HTML và WEB. JavaScript tốc độ nhanh và được sử dụng phổ biến cho các trang web. JavaScript được biết đến đầu tiên với tên gọi Mocha, sau đó là LiveScript rồi JavaScript. JavaScript xuất hiện lần đầu trong Netscape 2.0 năm 1995 với tên LiveScript. Core đa năng của ngôn ngữ này đã được nhúng vào Netscape, IE, và các trình duyệt khác.

JavaScript code được thực thi khi người sử dụng đệ trình Form, và chỉ nếu tất cả đầu vào là hợp lệ, chúng sẽ được đệ trình tới Web Server.

JavaScript có thể được sử dụng để nắm bắt các sự kiện được khởi tạo từ người sử dụng như nhấp chuột, điều hướng link, và các hoạt động khác mà người dùng khởi tạo.

### 2.6.4. Giới thiệu về ES6

ES6 là chữ viết tắt của ECMAScript 6, là phiên bản mới nhất của chuẩn ECMAScript. ECMAScript do hiệp hội các nhà sản xuất máy tính Châu Âu đề xuất làm tiêu chuẩn của ngôn ngữ Javascript. hiện nay có khá nhiều trình duyệt Browser ra đời và nếu mỗi Browser lại có cách chạy Javascript khác nhau thì các trang web không thể hoạt động trên tất cả các trình duyệt đó được, vì vậy cần có một chuẩn chung để bắt buộc các browser phải phát triển dựa theo chuẩn đó.

ES6 ra đời vào năm 2015 nên cái tên ES2015 được lấy làm tên chính thức với nhiều tính năng mới học hỏi các ngôn ngữ cấp cao khác, dần theo thời gian Javascript trở thành một ngôn ngữ lập trình hướng đối tượng.

### 2.5.3. Các chức năng mới của ES6

- Arrow function: tạo hàm bằng cách sử dụng dấu mũi tên =>.

- Block Scoped: Định nghĩa biến với từ khóa let, cách định nghĩa này thì biến chỉ tồn tại trong phạm vi khối của nó (Block Scope).

- Destructuring Assignments: Bạn có thể khởi tạo các biến từ một mảng bằng một dòng code đơn giản.

- Default Parameters: gán giá trị mặc định cho các tham số.

- Template String: Tạo templaet HTML cực kì đơn giản.

- Weak, Set: các kiểu dữ liệu phức tạp mới.

### 2.6.5. Boostrap 4

Bootstrap là framework HTML, CSS, JavaScript phổ biến nhất để phát triển một trang web đáp ứng, thân thiện với thiết bị di động. Về cơ bản, nó bao gồm một tập hợp các chuỗi code (teamplate) HTML, JavaScript và CSS có thể tái sử dụng như typography, form, button, table, điều hướng... chỉ cần nhập những chuỗi này vào code để nhanh chóng tạo ra loạt tính năng cần thiết cho trang web.

Bootstrap hoàn toàn miễn phí để tải và sử dụng. Nhờ bộ sưu tập code được viết sẵn và khả năng tái sử dụng Bootstrap giúp các nhà phát triển tiết kiệm được nhiều thời gian, công sức khi code web.



Hình 2.11. Giới thiệu về Boostrap 4

*(Nguồn:* [*https://getbootstrap.com.vn/blog/getbootstrap-4-la-gi*](https://getbootstrap.com.vn/blog/getbootstrap-4-la-gi)*)*

## 2.7. Phần mềm lập trình

### 2.7.1. Phần mềm Visual Sudio Code

Visual Studio Code Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor.

Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.

### 2.7.2. Một số tính năng của Visual Studio Code

*• Hỗ trợ nhiều ngôn ngữ lập trình*

Visual Studio Code hỗ trợ nhiều ngôn ngữ lập trình như C/C++, C#, F#, Visual Basic, HTML, CSS, JavaScript, … Vì vậy, nó dễ dàng phát hiện và đưa ra thông báo nếu chương chương trình có lỗi.

*• Hỗ trợ đa nền tảng*

Các trình viết code thông thường chỉ được sử dụng hoặc cho Windows hoặc Linux hoặc Mac Systems. Nhưng Visual Studio Code có thể hoạt động tốt trên cả ba nền tảng trên.

*• Cung cấp kho tiện ích mở rộng*

Trong trường hợp lập trình viên muốn sử dụng một ngôn ngữ lập trình không nằm trong số các ngôn ngữ Visual Studio hỗ trợ, họ có thể tải xuống tiện ích mở rộng. Điều này vẫn sẽ không làm giảm hiệu năng của phần mềm, bởi vì phần mở rộng này hoạt động như một chương trình độc lập.

*• Kho lưu trữ an toàn*

Đi kèm với sự phát triển của lập trình là nhu cầu về lưu trữ an toàn. Với Visual Studio Code, người dùng có thể hoàn toàn yên tâm vì nó dễ dàng kết nối với Git hoặc bất kỳ kho lưu trữ hiện có nào.

*• Hỗ trợ web*

Visual Studio Code hỗ trợ nhiều ứng dụng web. Ngoài ra, nó cũng có một trình soạn thảo và thiết kế website.

*• Lưu trữ dữ liệu dạng phân cấp*

Phần lớn tệp lưu trữ đoạn mã đều được đặt trong các thư mục tương tự nhau. Ngoài ra, Visual Studio Code còn cung cấp các thư mục cho một số tệp đặc biệt quan trọng.

*• Hỗ trợ viết Code*

Một số đoạn code có thể thay đổi chút ít để thuận tiện cho người dùng. Visual Studio Code sẽ đề xuất cho lập trình viên các tùy chọn thay thế nếu có.

*• Hỗ trợ thiết bị đầu cuối*

Visual Studio Code có tích hợp thiết bị đầu cuối, giúp người dùng khỏi phải chuyển đổi giữa hai màn hình hoặc trở về thư mục gốc khi thực hiện các thao tác.

*• Màn hình đa nhiệm*

Người dùng Visual Studio Code có thể mở cùng lúc nhiều tệp tin và thư mục – mặc dù chúng không hề liên quan với nhau.



Hình 2.12. Giới thiệu phần mềm lập trình

*(Nguồn:* [*https://code.visualstudio.com/download*](https://code.visualstudio.com/download) *)*

## 2.8. Hệ thống quản lý souce code

### 2.8.1. Giới thiệu GIT

VCS là viết tắt của Version Control System là hệ thống kiểm soát các phiên bản phân tán mã nguồn mở. Các VCS sẽ lưu trữ tất cả các file trong toàn bộ dự án và ghi lại toàn bộ lịch sử thay đổi của file. Mỗi sự thay đổi được lưu lại sẽ được và thành một version (phiên bản).

VCS nghĩa là hệ thống giúp lập trình viên có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (clone) từ một kho chứa mã nguồn (repository), mỗi thay đổi vào mã nguồn trên local sẽ có thể ủy thác (commit) rồi đưa lên server nơi đặt kho chứa chính.

Và một máy tính khác nếu người khác có quyền truy cập cũng có thể tải lại mã nguồn từ kho chứa hoặc tải lại một tập hợp các thay đổi mới nhất trên máy tính kia.

Lập trình viên có thể xem lại danh sách các sự thay đổi của file như xem một dòng thời gian của các phiên bản. Mỗi phiên bản bao gồm: nội dung file bị thay đổi, ngày giờ sửa đổi, người thay đổi là ai, lý do thay đổi hay tên phiên bản…

# CHƯƠNG III: KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG

## 3.1. Khảo sát hiện trạng

### 3.1.1. Hiện trạng của Velox Tech

Qua quá trình khảo sát tại công ty Velox Tech hiện tại công ty đang phát triển về nghiệp vụ chính của công ty nên vẫn chưa có website chính vì vậy việc quản lý công việc của nhân viên vẫn còn phải thủ công mất rất nhiều thời gian cho việc họp của công ty, quản lý thông tin của nhân viên phải trên sổ sách nên việc tìm kiếm cũng khá khó khăn làm mất rất nhiều thời gian, mặt khác việc tìm kiếm khách hàng cũng gặp nhiều bất lợi vì thông tin của công ty trên internet còn khá hạn chế, ít khách hàng biết đến, nên việc tìm đối tác còn mất nhiều thời gian. Không những thế, công ty hiện đang chuyên về nghiên cứu và phát triển lĩnh vực công nghệ cho nên phải có một trang web thể hiện sự chuyên nghiệp của công ty nó còn giúp cho tên tuổi của công ty đến gần hơn với nhiều đối tác.

Về phía nhân viên tuyển dụng của công ty có nhiệm vụ tuyển dụng việc làm, quảng cáo công ty, hằng ngày phải tìm kiếm những trang tuyển dụng việc làm và tìm kiếm nhân viên cho công ty, mất khá nhiều thời gian cho việc tìm kiếm từng trang web như thế.

Về phía giám đốc mỗi ngày phải quan sát, đánh giá tiến độ công việc, dự án, của nhân viên làm hằng ngày điều này mất khá nhiều thời gian lãng phí tài nguyên.

Về phía nhân viên, hàng ngày phải báo cáo tiến độ công việc, dự án và báo cáo những công việc mỗi ngày sẽ làm gì trong ngày qua cuộc họp mỗi sáng việc này sẽ tốn từ 30 phút đến 45 phút cho mỗi lần họp. làm thời gian hoàn thành tiến công việc của mỗi người ngắn lại, đôi khi phải kéo dài tiến độ của công việc ra vì không thể hoàn thành đúng như dự tính.

## 3.2. Xác định và mô tả yêu cầu chức năng

### 3.2.1. Sơ đồ chức năng

**Chức năng trong website khách hàng:**

*Chức năng gửi mail:*

Mục tiêu: Cho phép khách hàng có thể gửi mail đến công ty với thông tin cá nhân kèm vấn đề, câu hỏi cần được các nhân viên bên công ty giải đáp và tư vấn. Phía bên công ty nhận mail từ phía khách hàng sau đó tiến hành tìm hiểu vấn đề của khách hàng sau đó đưa ra cách giải quyết cuối cùng trả lời cho khách hàng thông qua những thông tin khách hàng đã cũng cấp trước đó.

*Chức năng ứng tuyển việc làm:*

Mục tiêu: giúp công ty nhận được những đơn xin việc online thông qua yêu cầu tuyển dụng từ phía người ứng tuyển. từ phía người ứng tuyển sau khi chọn công việc muốn ứng tuyển sau đó gửi thông tin cá nhân kèm theo nội dung mong muốn ứng tuyển cho công ty. Phía công ty sẽ nhận được yêu cầu ứng tuyển xem xét và trả lời lại qua mail cho người ứng tuyển.

Chức năng trong website nội bộ công ty:

**- Phía nhân viên:**

*Chức năng đổi mật khẩu*

Mục tiêu: giúp nhân viên quản lý tài khoản hiệu quả và bảo mật. Ngăn chặn việc người khác cố tình đăng nhập vào tài khoản.

*Chức năng thêm, xóa sửa công việc hằng ngày*

Mục tiêu: giúp giám đốc quản lý được công việc của từng thành viên trong công ty, đánh giá được quá trình làm việc, mức độ công việc của nhân viên.

*Chức năng xem thông tin cá nhân*

Mục tiêu: xem và kiểm tra thông tin cá nhân của từng người.

*Chức năng xem thông báo*

Mục tiêu: xem những thông báo quan trọng của giám đốc về công ty, và những thông báo về ngày nghỉ của nhân viên.

*Chức năng thêm xóa sửa tiến độ công việc*

Mục tiêu: giám xác quá trình, tiến độ của công việc của mỗi người. Giúp người quản lý và nhân viên khác quan sát quá trình của mình.

*Chức năng thông báo đã xem*

Mục tiêu: giúp quan sát những thành viên trong công ty đã xem thông báo.

- phía tuyển dụng:

*Chức năng thêm xóa sửa nhân viên*

Mục tiêu: thêm, xóa, sửa nhân viên vừa vào công ty, cung cấp tài khoản cho nhân viên mới để đăng nhập vào hệ thống.

*Chức năng thêm xóa sửa thông tin tuyển dụng*

Mục tiêu: thêm, xóa, sửa thông tin về tuyển dụng giúp công ty đang thiếu nhân lực tìm kiếm những nhân viên muốn tham gia vào công ty.

**- Phía giám đốc**

*Chức năng thêm, xóa, sửa thông báo của công ty*

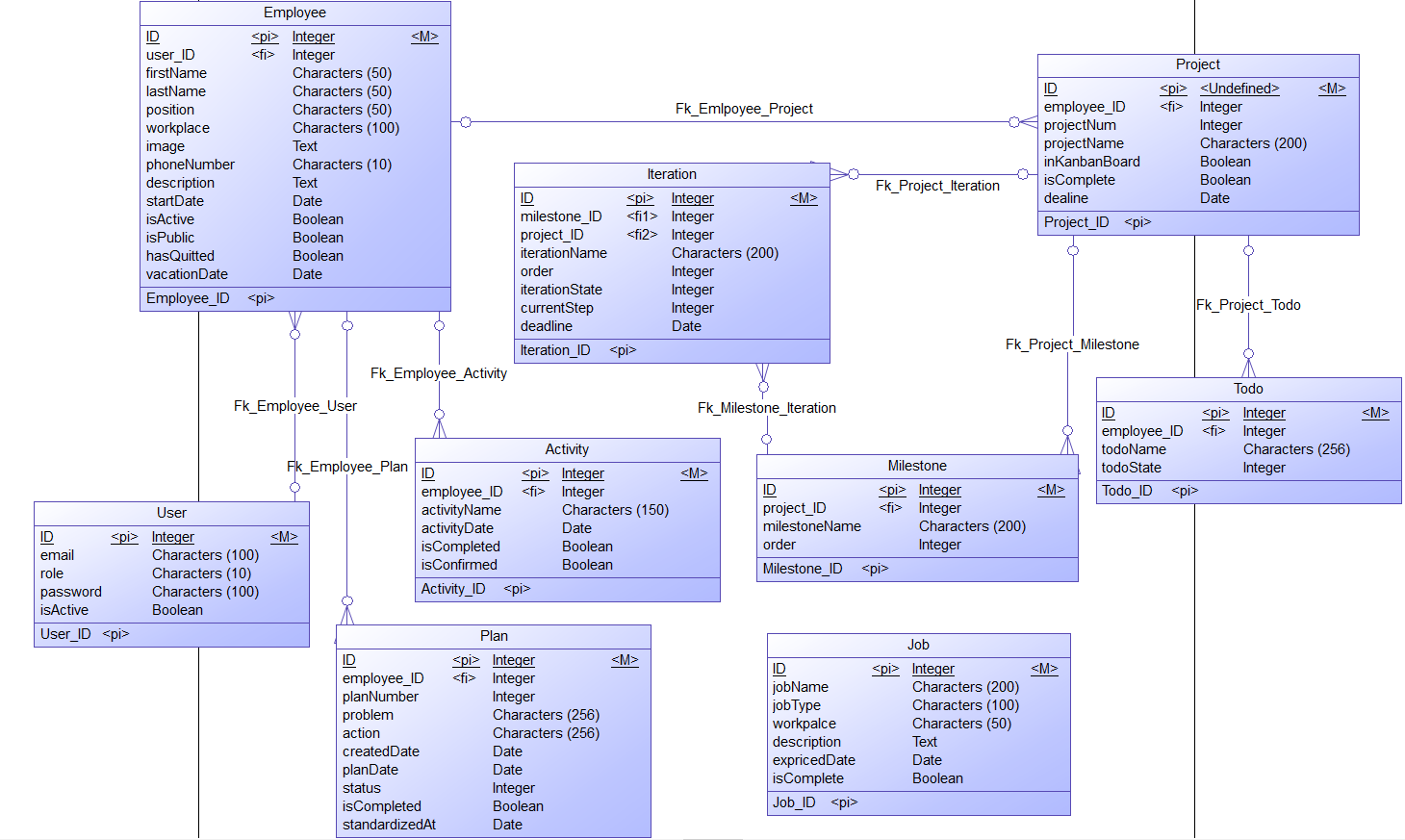
Mục tiêu: thêm và điều chỉnh thông báo giúp giám đốc thông báo đến các nhân viên trong công ty mà không cần phải tổ chức cuộc họp.

*Chức năng phê duyệt công việc của nhân viên*

Mục tiêu: thông báo cho giám đốc được những công việc của nhân viên khi thêm xem và phê duyệt công việc của nhân viên.

## 3.3. Các mô hình dữ liệu

### 3.3.1. Mô hình logic dữ liệu



Hình 3.1. Hình Mô hình thực thể kết hợp

Diễn dãi các bảng dữ liệu như sau:

Bảng 3.1. Bảng dữ liệu nhân viên (Employee)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ghi chú** | **Ràng buộc** |
| 1 | ID | Unsigned int | 1, 2, … + | Mã nhân viên | Khóa chính |
| 2 | firstName | Character | Tối đa 50 kí tự | Họ, tên lót nhân viên |  |
| 3 | lastName | Character | Tối đa 50 kí tự | Tên nhân viên |  |
| 4 | position | Character | Tối đa 100 kí tự | Vị trí |  |
| 5 | workplace | Character | Tối đa 100 kí tự | Làm việc tại khu vực |  |
| 6 | Image | Text |  | Hình ảnh |  |
| 7 | phoneNumber | Character | Tối đa 10 kí tự | Số điên thoại |  |
| 8 | description | Text |  | Giới thiệu bản thân |  |
| 9 | startDate | Date |  | Ngày bắt đầu làm |  |
| 10 | isActive | Boolean | True, False | Đang hoạt động |  |
| 11 | isPublic | Boolean | True, False | Công khai |  |
| 12 | hasQuitted | Boolean | True, False | Đã nghỉ việc |  |
| 13 | vacationDate | Date |  | Ngày nghỉ |  |
| 14 | user\_ID | Unsigned int | 1, 2, … + | Mã tài khoản | Khóa phụ |

Bảng 3.2. Bảng dữ liệu tài khoản người dùng (User)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ghi chú** | **Ràng buộc** |
| 1 | ID | unsigned int | 1, 2, … + | Khóa chính | Khóa chính |
| 2 | Email | Character | Tối đa 256 kí tự | Email |  |
| 3 | Role | Character | ['admin', 'hr', 'staff'] | Cấp bật |  |
| 4 | password | Character |  | Mật khẩu |  |
| 5 | isActive | Boolean | True, False | Đang hoạt động |  |

Bảng 3.3. Bảng dữ liệu kế hoạch (Plan)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ghi chú** | **Ràng buộc** |
| 1 | ID | Unsigned int | 1, 2, … + | Mã kế hoạch | Khóa chính |
| 2 | planNumber | Unsigned int | 1, 2, … + | Số kế hoạch |  |
| 3 | problem | Character | Tối đa 256 kí tự | Vấn đề |  |
| 4 | Action | Character | Tối đa 256 kí tự | Hoạt động |  |
| 5 | createdDate | Date |  | Ngày tạo |  |
| 6 | planDate | Date |  | Số ngày hoàn thành kế hoạch |  |
| 7 | status | Int | [0, 1, 2, 3, 4] | Trạng thái |  |
| 8 | isCompleted | Boolean | True, False | Đã hoàn thành |  |
| 9 | standardizedAt | Date |  | Đang tại tiêu chuẩn |  |
| 10 | employee\_ID | Unsigned int | 1, 2, … + | Mã nhân viên | Khóa phụ |
| 11 | creator\_ID | Unsigned int | 1, 2, … + | Mã người tạo | Khóa phụ |

Bảng 3.4. Bảng dữ liệu hoạt động (Activity)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ghi chú** | **Ràng buộc** |
| 1 | ID | Unsigned int | 1, 2, … + | Mã hoạt động | Khóa chính |
| 2 | activityName | Character | Tối đa 100 kí tự | Tên hoạt động |  |
| 3 | activityDate | Date |  | Ngày hoạt động |  |
| 4 | isCompleted | Boolean | True, False | Đã xong |  |
| 5 | isConfirmed | Boolean | True, False | Đã duyệt |  |
| 6 | employee\_ID | Unsigned int | 1, 2, … + | Mã nhân viên | Khóa phụ |

Bảng 3.5. Bảng dữ liệu dự án (Project)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ghi chú** | **Ràng buộc** |
| 1 | ID | Unsigned int | 1, 2, … + | Mã dự án | Khóa chính |
| 2 | projectNum | Character | Tối đa 100 kí tự | Số dự án |  |
| 3 | projectName | Character | Tối đa 100 kí tự | Tên dự án |  |
| 4 | inKanbanBoard | Boolean | True, False | ở trên bảng |  |
| 5 | isCompleted | Boolean | True, False | Đã hoàn thành |  |
| 6 | deadline | Date |  | Ngày cuối |  |
| 7 | creator\_ID | Unsigned int | 1, 2, … + | Mã người tạo | Khóa phụ |

Bảng 3.6. Bảng dữ liệu lịnh trình (Todo)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ghi chú** | **Ràng buộc** |
| 1 | ID | Unsigned int | 1, 2, … + | Mã lịch trình | Khóa chính |
| 2 | todoName | Character | Tối đa 100 kí tự | Tên lịch trình |  |
| 3 | todoState | Unsigned int |  | Trạng thái lịch trình |  |
| 4 | project\_ID | Unsigned int | 1, 2, … + | Mã dự án | Khóa phụ |

Bảng 3.7. Bảng dữ liệu cột mốc (Milestone)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Stt | Thuộc tính | Kiểu dữ liệu | Miền giá trị | Ghi chú | Ràng buộc |
| 1 | ID | Unsigned int | 1, 2, … + | Mã cộc mốc | Khóa chính |
| 2 | milestoneName | Character | Tối đa 100 kí tự | Tên cột mốc |  |
| 3 | Order | Int | 1, 2,…, 100 | Phần trăm |  |
| 4 | project\_ID | Unsigned int | 1, 2, … + | Mã dự án | Khóa phụ |

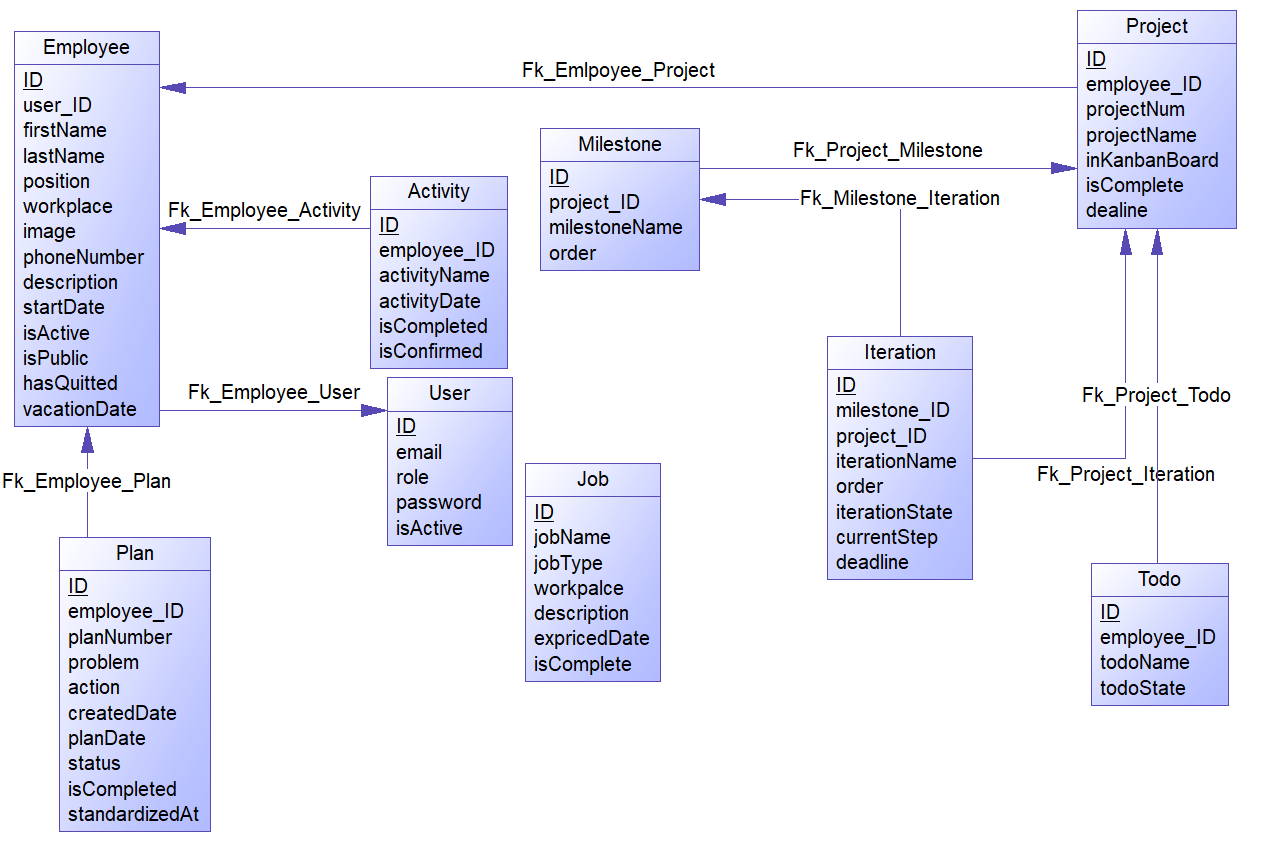
Bảng 3.8. Bảng dữ liệu vòng lặp dự án (Iteration)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ghi chú** | **Ràng buộc** |
| 1 | ID | Unsigned int | 1, 2, … + | Mã vòng lập | Khóa chính |
| 2 | iterationName | Character | Tối đa 100 kí tự | Tên vòng lập |  |
| 3 | Order | Int | … + | Thứ tự |  |
| 4 | milestone\_ID | Unsigned int | 1, 2, … + | Mã Cột mốc |  |
| 5 | iterationState | Unsigned int | [0, 1, 2] | Trạng thái vòng lặp |  |
| 6 | currentStep | Unsigned int | [1, 2, 3, 4, 5, 6] | Bước hiện tại |  |
| 7 | deadline | Date |  | Hạn cuối |  |
| 8 | project\_ID | Unsigned int | 1, 2, … + | Mã dự án | Khóa phụ |

Bảng 3.9. Bảng dữ liệu tuyển dụng công việc (Job)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Stt** | **Thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ghi chú** | **Ràng buộc** |
| 1 | ID | Unsigned int | 1, 2, … + | Mã công việc | Khóa chính |
| 2 | jobName | Character | Tối đa 100 kí tự | Tên công việc |  |
| 3 | jobType | Character | Tối đa 100 kí tự | Kiểu công việc |  |
| 4 | workplace | Character | Tối đa 100 kí tự | Nơi làm việc |  |
| 5 | description | Text |  | Nội dung |  |
| 6 | expiredDate | Date |  | Ngày hết hạn |  |
| 7 | isCompleted | Boolean | True, False | Đã hoàn thành |  |

### 3.3.2. Mô hình vật lý dữ liệu



Hình 3.2. Mô hình vật lý

# CHƯƠNG IV: CÀI ĐẶT HỆ THỐNG VÀ KẾT QUẢ ĐẠT ĐƯỢC

## 4.1. Cài đặt hệ thống

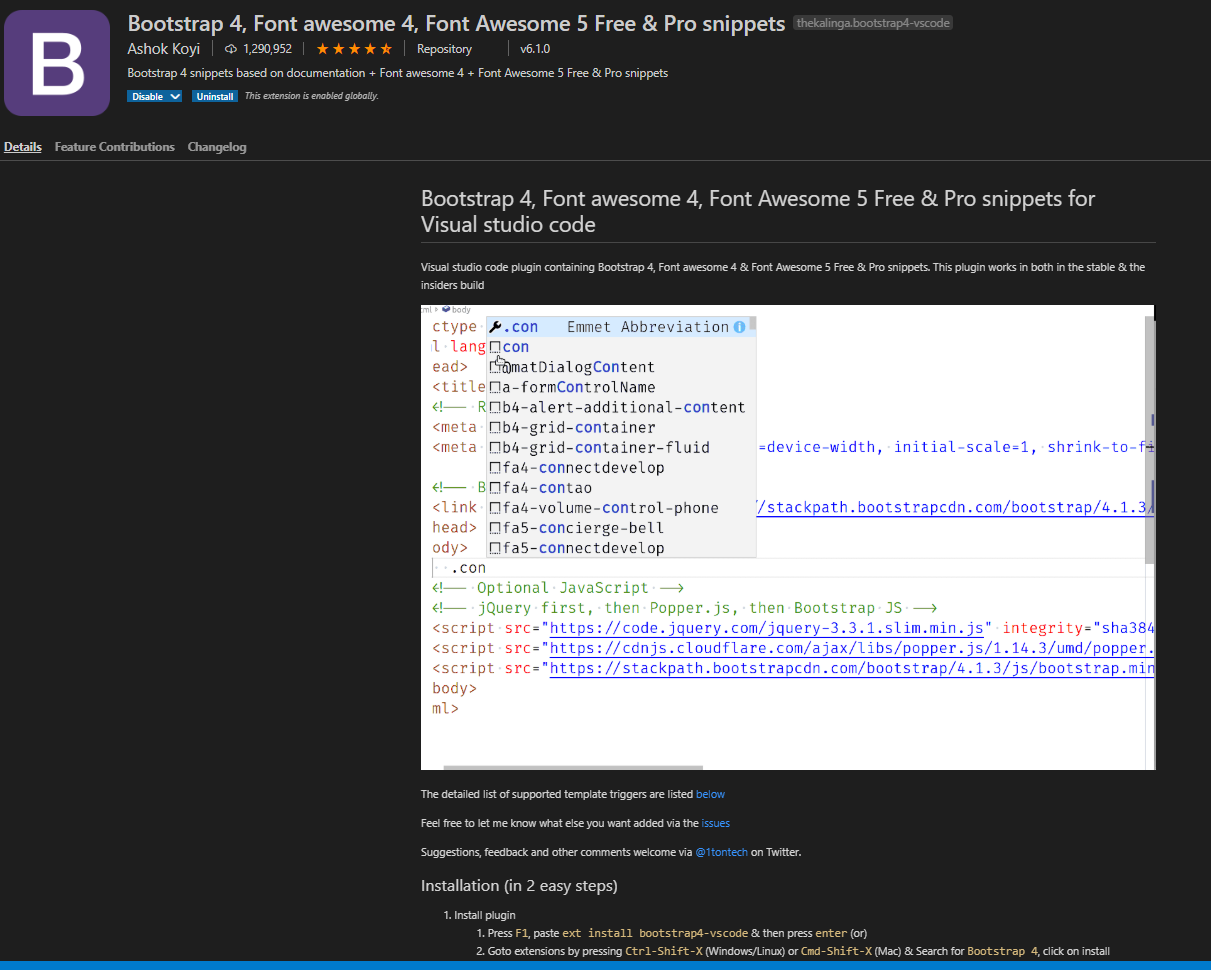
### 4.1.1. Cài đặt Visual Studio Code

Để lập trình tốt về vấn đề trên cần phải có phần mềm lập trình với nhiều công cụ hỗ trợ lập trình tốt, dễ sử dụng, dễ cài đặt, ứng dụng nghẹ, hoạt động nhanh

Để download ứng dụng Visual Studio Code truy cập vào đường dẫn: <https://code.visualstudio.com/download>

Để download phần mềm nay cần phải tìm hiểu về hệ điều hành của mình là hệ đề hành nào rồi download cho phù hợp.

Sau khi download phần mềm Visual Studio Code xong tiếp theo tải công cụ hổ trợ gợi ý lập trình nhanh, ở đây là lập trình webisite nên cần cài những extension hổ trợ về lập trình như ở hình 4.2 đã cài extenstion Boostrap 4.



Hình 4.1. Cài đặt extension cho phần mềm Visual Studio Code

Tương tự sau khi cài đặt Boostrap 4 xong, cài những extention khác như:

- JavaScript (ES6) code snippets

- ES7 React/Redux/GraphQL/React-Native snippets

- ESLint

### 4.1.2. Cài đặt Boostrap 4

Như đã nói ở trên phần giới thiệu, Boostrap 4 hổ trợ lập trình giao diên nhanh thông qua những class CSS và các funtion Javascript giúp thiếu kế giao diện đa màn hình như: Điện thoại, tablet, laptop…

Để cài đặc Boostrap 4 truy cập vào đường dẫn:

<https://getbootstrap.com/docs/4.1/getting-started/download/>

Nếu muốn website nhẹ ta nên để đường dẫn trực thiếp thay vì download Boostrap4 việc này sẽ làm cho website nặng hơn nếu như Server chứa website với cấu hình yếu thì không nên sử dụng cách download. Vì thế thay về cách ở trên ta chèn đường dẫn của Boostrap 4 vào các file HTML.

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymous">

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbWh2IMqE241rYiqJxyMiZ6OW/JmZQ5stwEULTy" crossorigin="anonymous"></script>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49" crossorigin="anonymous"></script>

### 4.1.2. Cài đặt Node JS

Một thành phần không thể không nhắc đến đó là Node JS như đã giới thiệu ở phần trên với vai trò làm server của website cũng là hầu như nhưng chức năng của website đều phải sử lý bằng Node JS. Nó đóng vai trò chủ yếu để giao tiếp giao diện với cơ sở dữ liệu. Để cài đặt Node JS truy cập vào địa chỉ nodejs.org. Sau khi đã cài đặt xong Node JS vào máy, tiếp theo cần phải cài đặt Node JS - NPM.

Đầu tiên mở phần mềm Visual Studio Code lên, tiếp theo tạo thư mục cần chứ source code của mình để dễ quản lý. Kế tiếp bật teminal của phần mềm Visual lên đi đến thư mục vừa tạo như trên như Hình 4.



Hình 4.2. Cài đặt Node JS - NPM

Sau đó, cài Node JS npm bằng lệnh:

sudo npm install npm -g

Như vậy trình quản lý thư viện của Node JS đã được cài đặt, tiếp theo ta cài thêm Express Framwork và các pagkage hổ trợ server Node JS khác với lệnh:

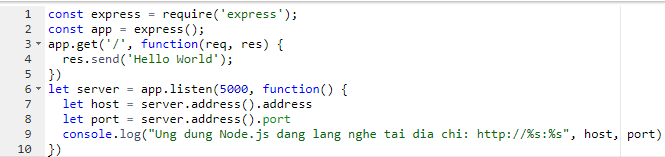
npm install express --save

npm install body-parser --save

npm install cookie-parser --save

npm install multer –save

Cuối cùng tạo một file server.js để khai báo Server của Node JS với Express.



### 4.1.3. Cài đặt React JS

Với React JS đóng vai trò chủ đạo về phía giao diện tương tác với người dùng. Nhưng để sử dụng thì cần phải cài đặt React JS cho website tương tự cài đặt Node JS

npm install -g create-react-app

create-react-app my-app

cd my-app

npm start

Sau khi đã cài xong Reat JS ta cà thêm những package hổ trợ cho React JS:

npm install --save-dev webpack webpack-dev-server html-webpack-plugin

npm install react --save

npm install react-dom --save

npm install babel-core

npm install babel-loader

npm install babel-preset-react

npm install babel-preset-es2015

Lệnh này sẽ giúp cài đặt các gói webpack, webpack-dev-server và html-webpack-plugin, các dependency vào project, lệnh --save-dev chỉ định sẽ lưu dependency của các gói này vào file package.json để chúng ta có thể mang source code đi nhiều nơi mà không cần mang theo các libs từ project. Sau khi cài đặt xong thêm start script để chạy webpack-dev-server vào trong file package.json như sau.



Vậy là đã cài đặt xong hết tất cả các Pagkage, các môi trường lập trình cho hệ thống. Cuối cùng là thiết kế giao diện và các chức năng cho website.

### 4.1.4. Cài đặt Axios

Như đã giới thiệu ở phần trên Axios dùng để hỗ trợ cho kĩ thuật Client Side Rendering trong quá trình xây dựng các ứng dụng API từ đơn giản đến phức tạp và có thể được sử dụng ở Node.js một trong các cách thường thấy để cài đặt Axios đó là thông qua trình quản lý các npm package.

npm i axios

và sau đó include nó vào trong code ở những nơi cần:

// ES2015 style import

import axios from 'axios';

// Node.js style require

const axios = require('axios');

Hoặc nếu không sử dụng một loại module bundler nào (ví dụ: webpack), thì có thể kéo thư viện này từ một CDN như theo cách truyền thống:

<script src="https://unpkg.com/axios/dist/axios.min.js"> </script>

**Tạo một request với Axios**

Giống như với hàm $.ajax của jQuery, có thể tạo bất kỳ một request HTTP nào bằng cách truyền vào các object option cho Axios:

axios({

method: 'post',

url: '/login',

data: {

user: 'test',

lastName: 'test1'

}

});

Ở đây cung cấp cho Axios biết rằng phương thức HTTP nào muốn dùng (Ví dụ: GET/POST/DELETE...) và URL nào được sử dụng để gửi request lên.

Đồng thời cũng cung cấp một số dữ liệu để có thể gửi kèm với request dưới dạng các object Javascript đơn giản gồm các cặp key/value. Mặc định, Axios sẽ serialize các object này thành JSON và gắn chúng ở phần body của request.

**Nhận về một response**

Một khi đã tạo ra một request, Axios trả về một promises mà sẽ resolve ngay cả khi kết quả trả về là một object response hay một object error.

axios.get('/login)

.then(response => console.log(response))

.catch(error => console.log(error));

**Response Object**

Khi một request được thực hiện thành công, thì hàm then() sẽ nhận được một response object với các thuộc tính như sau:

- **data**: Payload được trả về từ server. Mặc định, Axios mong chờ JSON và sẽ trở thành một Javascript Object.

- **status**: mã HTTP được trả về từ server.

- **statusText**: message của HTTP status được trả về bởi server.

- **headers**: tất cả các header được gửi về từ server.

- **config**: thiết lập ban đầu của request.

- **request**: một object XMLHttpRequest thực khi request được thực hiện ở trình duyệt.

**Error Object**

Nếu có bất kỳ vấn đề gì xảy ra với request, promise sẽ bị reject với một error object chứa những thuộc tính sau đây.

- **message**: message của error ở dạng text.

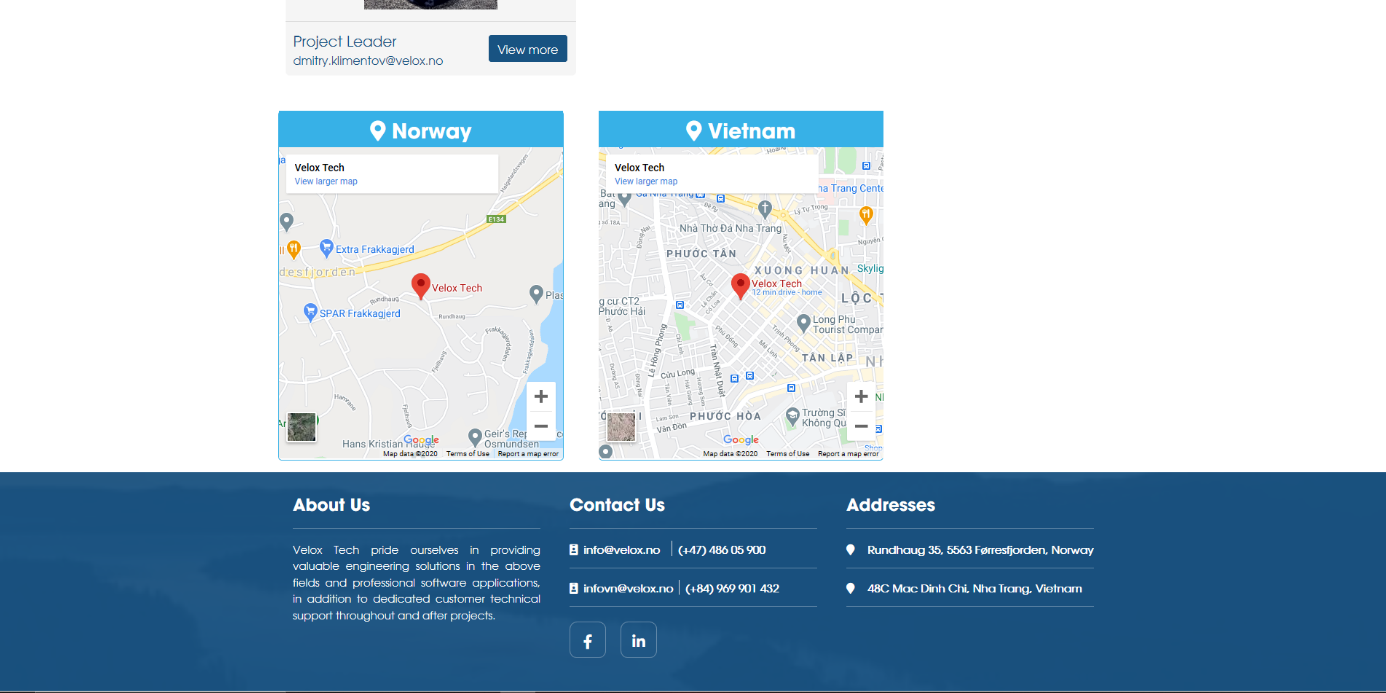
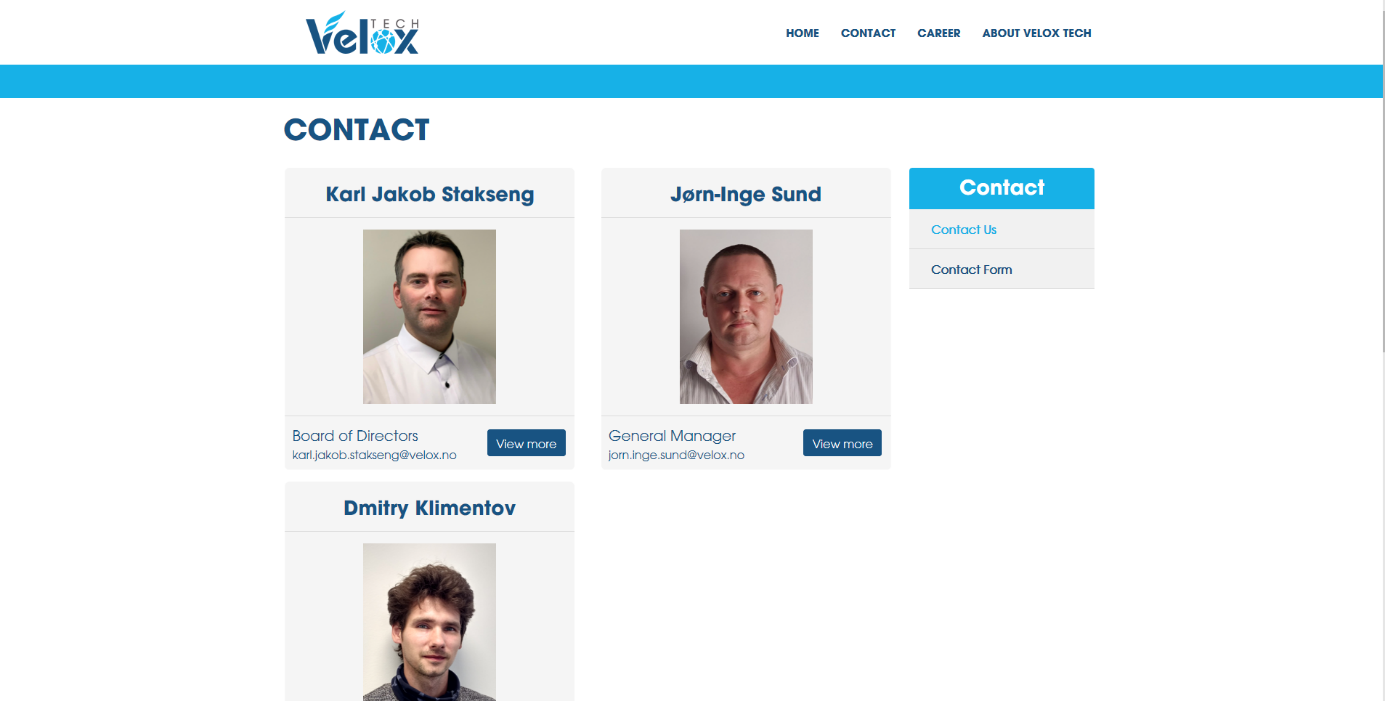
- **response**: response object được trả về.

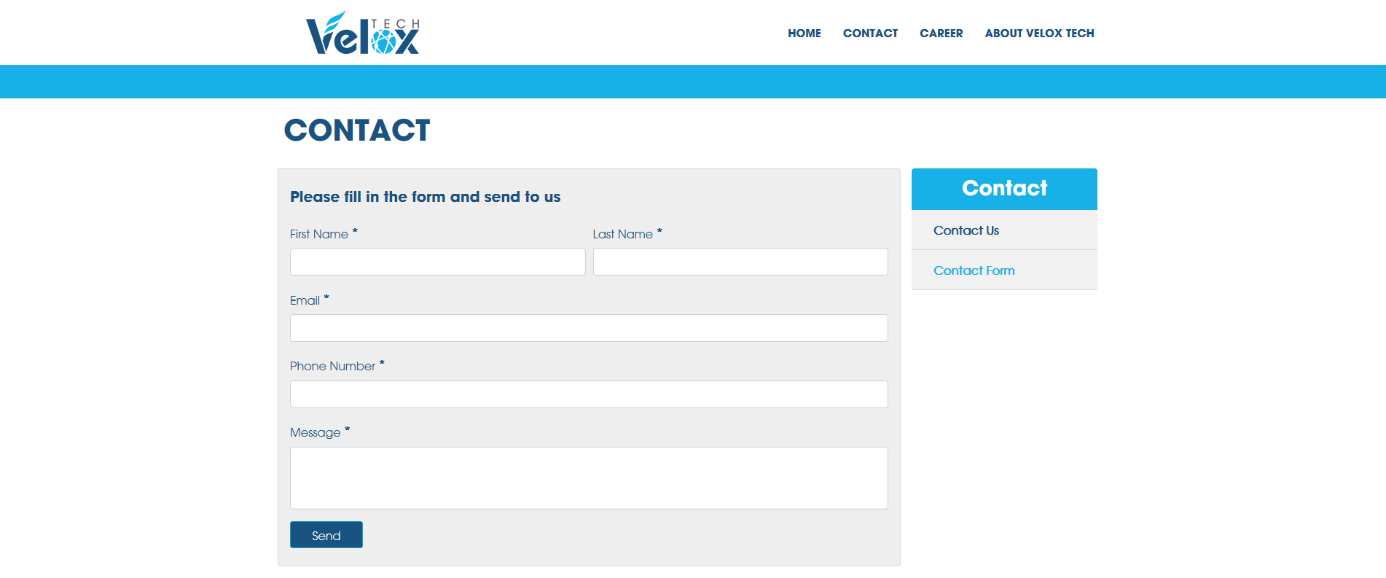
- **config**: thiết lập ban đầu của request.

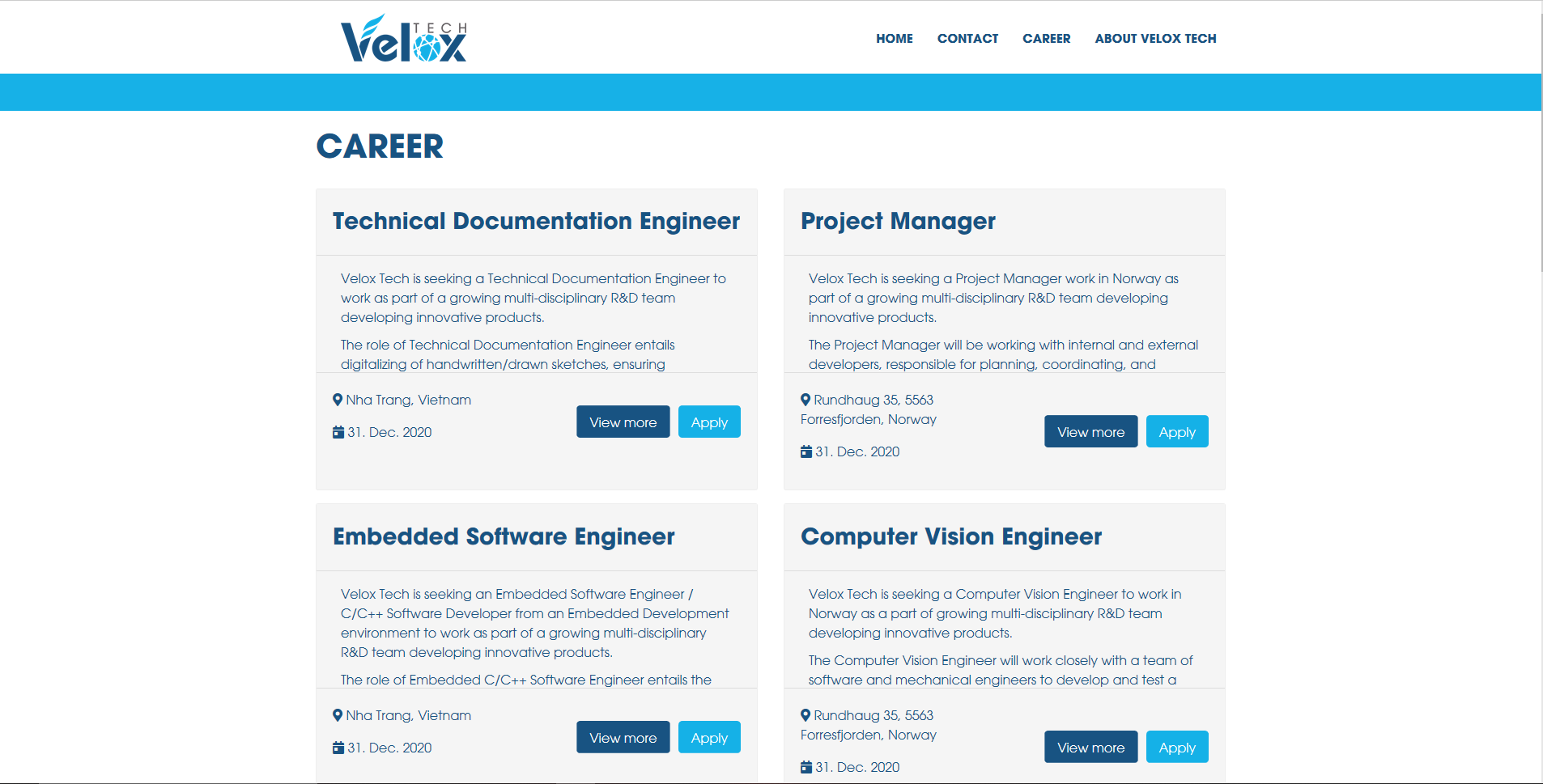
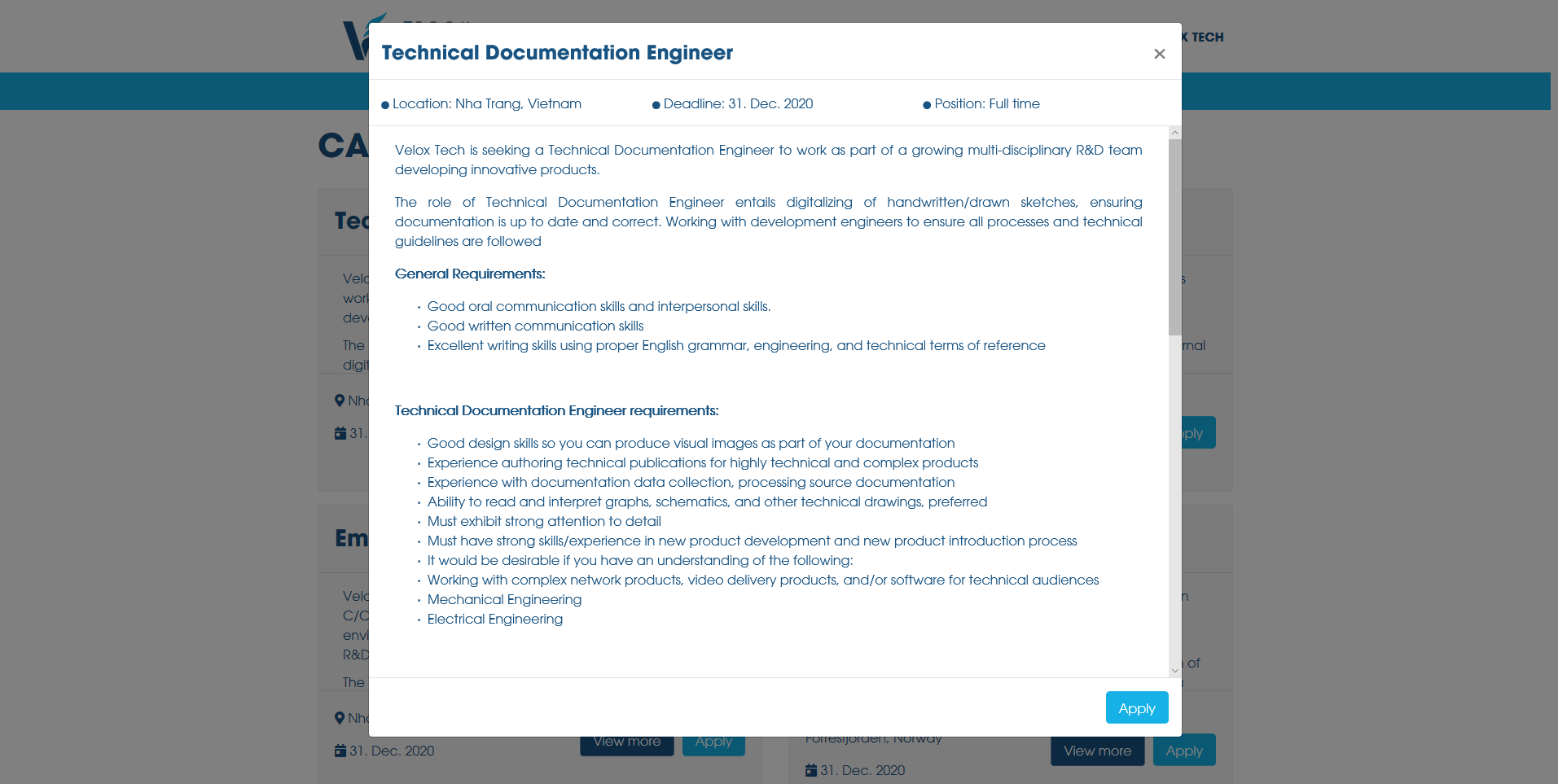
- **request**: một object XMLHttpRequest thực khi request được thực hiện ở trình duyệt.

## 4.2. Kết quả đạt được

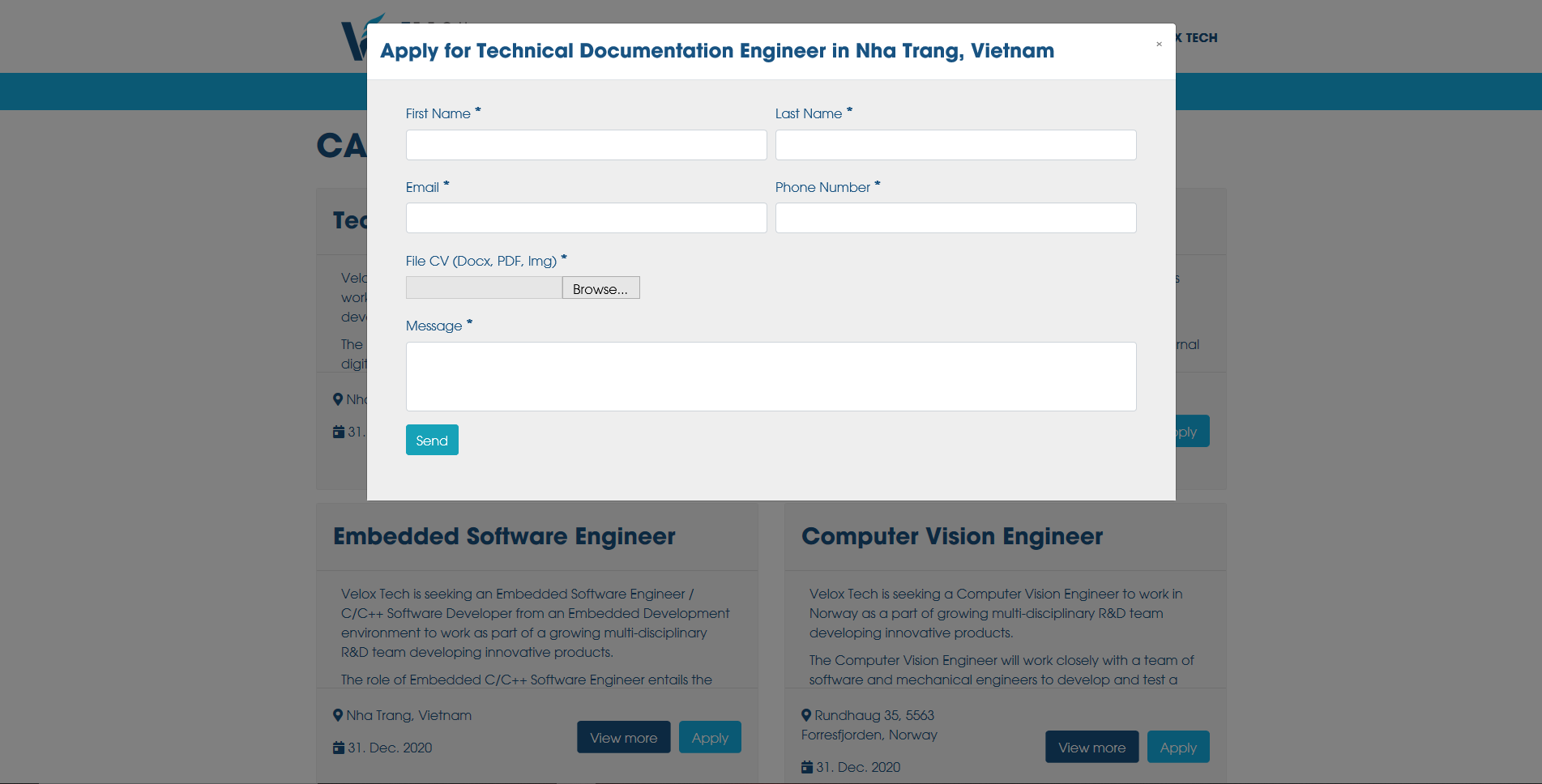
Hình 4.3. Trang chủ website khách hàng Velox Tech

*Hình 4.4. Trang thông tin liên hệ của công ty Velox Tech*

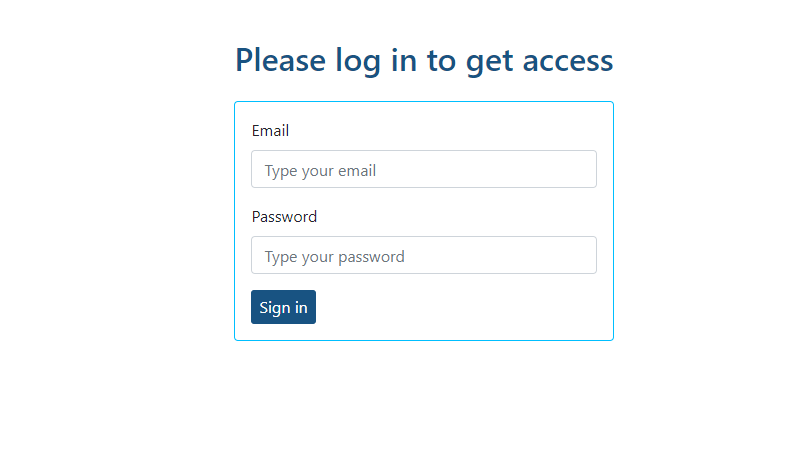
*Hình 4.5. Trang thông tin liên hệ giử ý kiến người dùng*

*Hình 4.6. Trang tuyển dụng công ty Velox Tech*

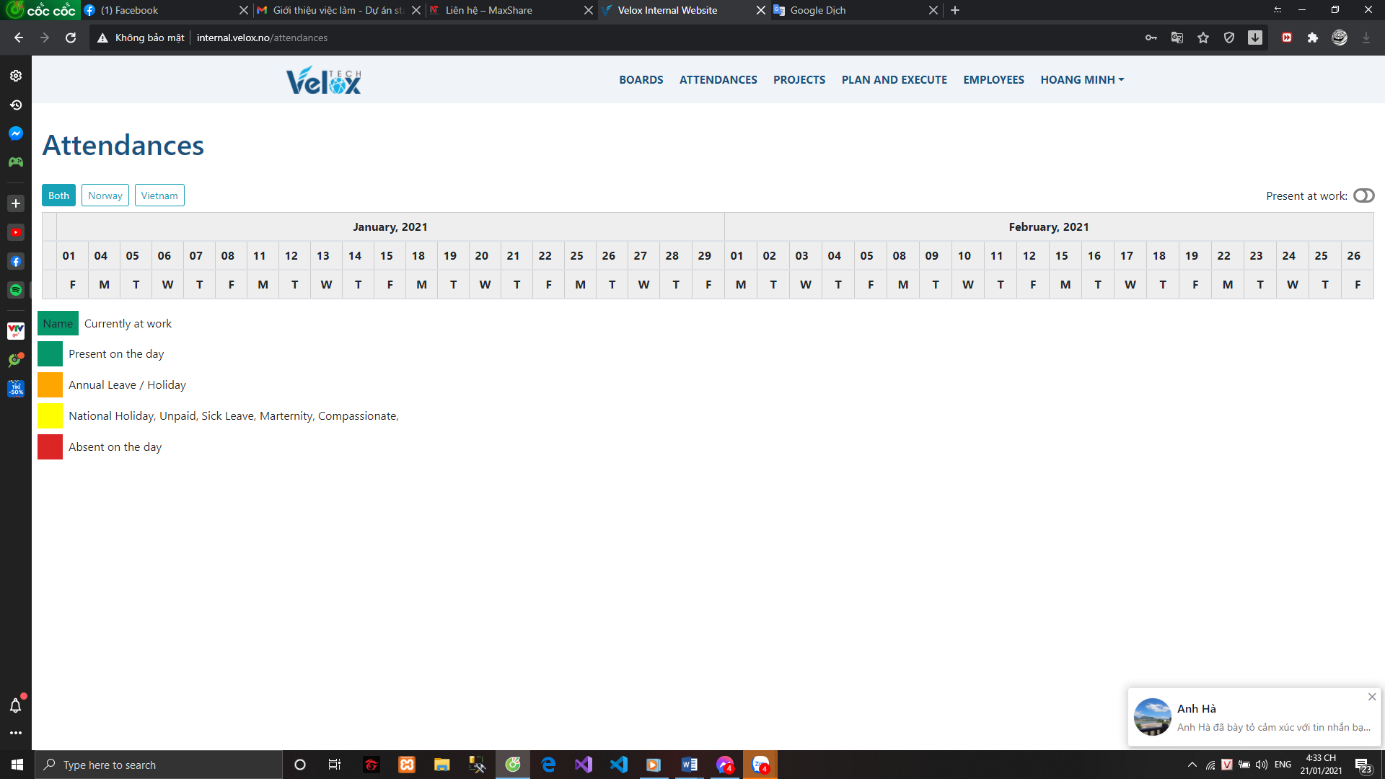
*Hình 4.7. Trang tuyển dụng sau khi xem thêm*

*Hình 4.8. Trang tuyển dụng sau khi nhấn nút ứng tuyển*

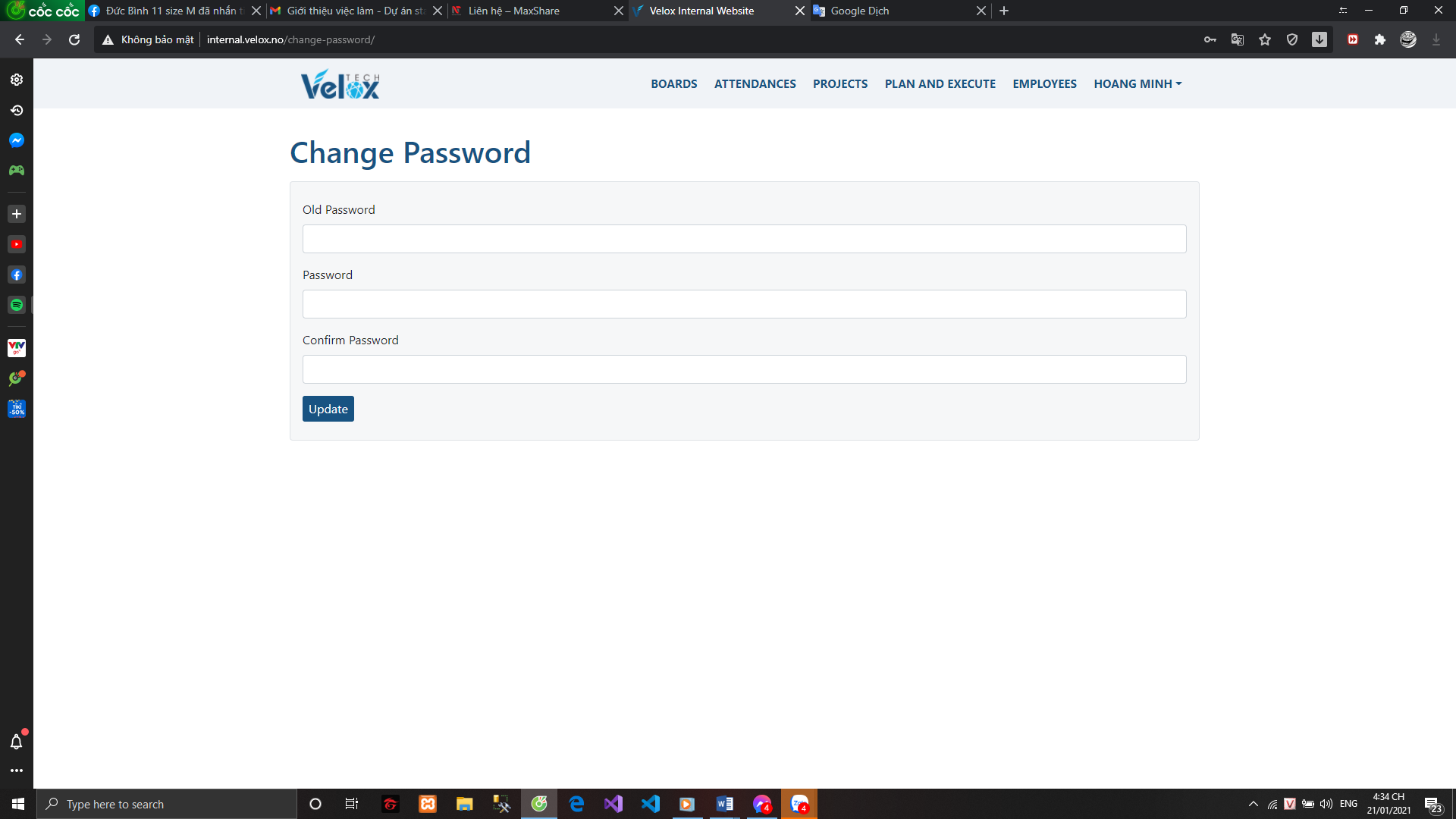
*Hình 4.9. Trang giới thiệu về công ty Velox Tech*



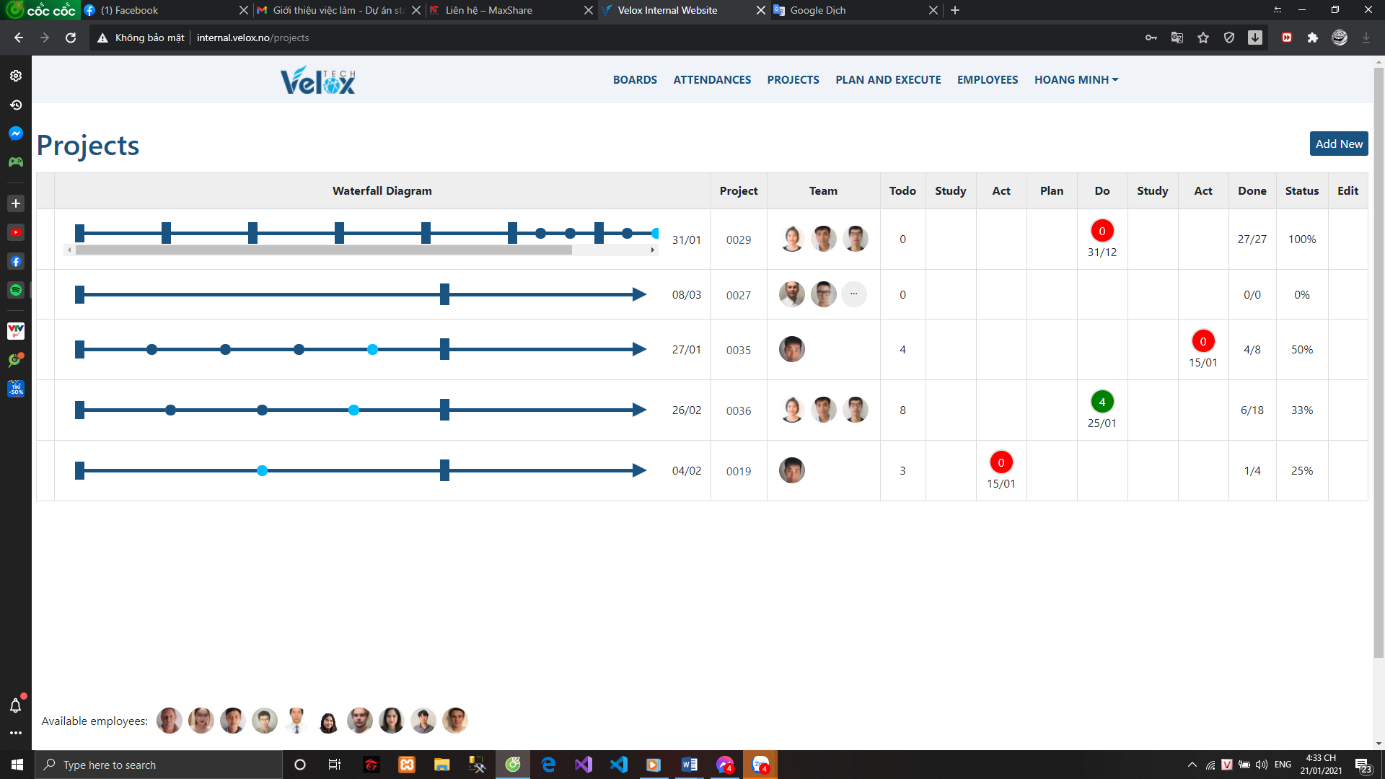
*Hình 4.10. Trang đăng nhập vào hệ thống web nội bộ*



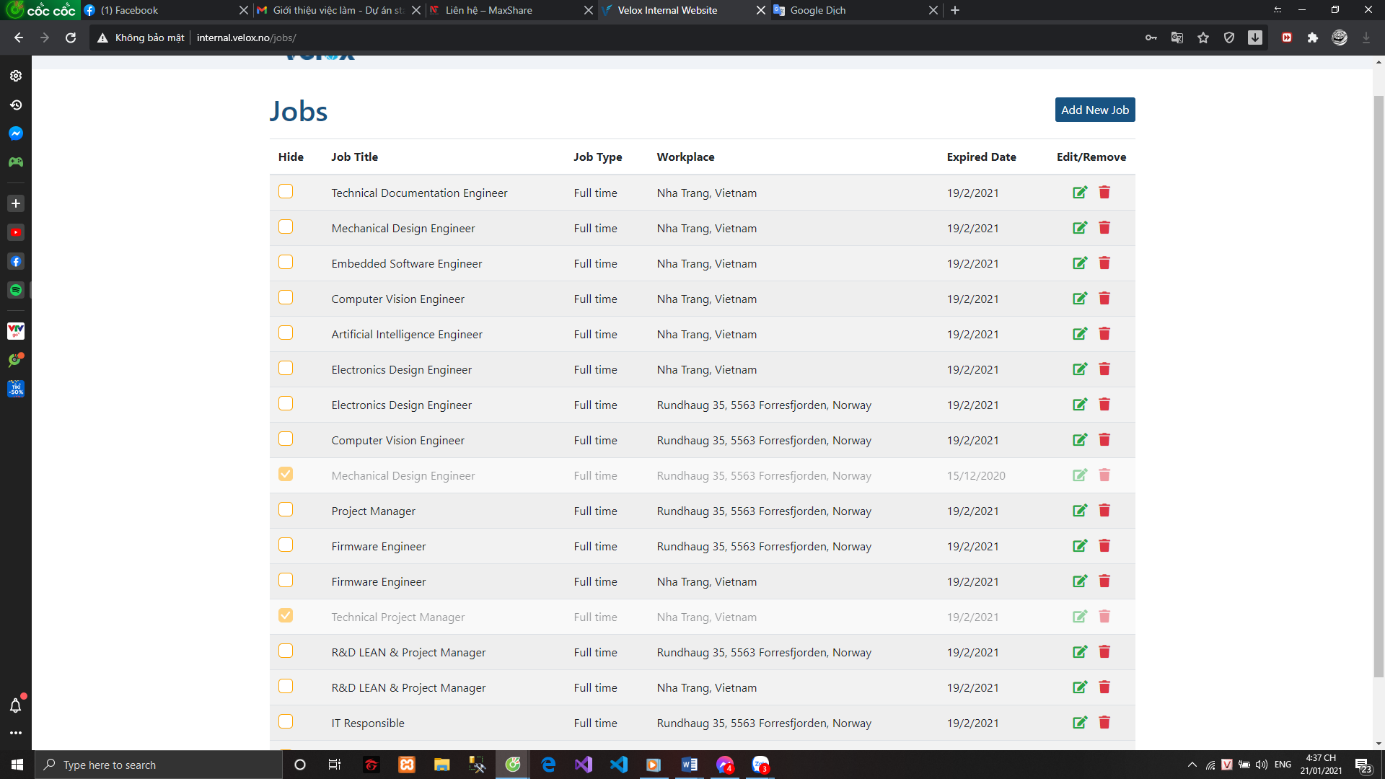
*Hình 4.11. Trang lịch ngày nghỉ của nhân viên*



*Hình 4.12. Trang đổi mật khẩu người dùng*



Hình 4.13. Trang quản lý công việc của nhân viên



Hình 4.13. Trang đăng và chỉnh sửa tuyển dụng

# CHƯƠNG V: ƯU NHƯỢC ĐIỂM VÀ HƯỚNG PHÁT TRIỂN

## 5.1. Ưu, nhược điểm của đề tài

### 5.1.1. Ưu điểm

Trong quá trình làm để tài thực tập và làm việc tại công ty Velox Tech, nhận thấy việc thiết kế website cho công ty mang lại nhiều lợi ích về phía công ty, giúp công ty có tìm được nhiều khách hàn hơn, quản lý công việc cũng trở nên nhanh và tối ưu hơn, không những thế sau khi thực tập bản thân đã tích góp được nhiều kinh nghiệm trong phát triển ứng dụng thực tế với những kiến thức như:

- Hiểu rõ hơn về Node JS, React JS, hệ quản trị cơ sở dữ liệu MySQL, HTML/CSS/Javascript, Bootstrap, thực hiện viết code trên phần mềm Visual Studio Code.

- Hiểu được quy trình tổ chức lập trình, quản lý và chia sẽ mã nguồn cho nhóm.

- Giao diện đơn giản dễ nhìn.

- Hiểu rõ được quá trình tổ chức công việc của công ty.

### 5.1.2. Nhược điểm

Do thời gian thực tập còn hạn chế nên website phải chuyển giao lại cho nhân viên của công ty phát triển tiếp tục. Những kiến thức mới còn khá mới lạ mất nhiều thời gian để làm quen

## 5.2. Hướng phát triển

Tạo ra một trang quản lý nhân sự hoàn chỉnh cho Velox Tech.

Tạo thêm các tính năng mới phù hợp với nghiệp vụ của Velox Tech.

# KẾT LUẬN

Qua quá trình thực tập nghiêm túc tại Velox Tech nhận thấy được trải nghiệm và làm việc tại môi trường thực tế, trong quá trình lập trình được học hỏi được nhiều kinh nghiệm từ dự án, nghiên cứu. Nhận được nhiều kiến thức mới, và cũng cố những kiến thức cũ áp dụng trong thực tế. Hiểu hơn về cách hoạt động của Node JS, React JS, cách tổ chức và quản lý source code. Tiếp cận được với tổ chức của công ty tham gia vào quá trình phát triển của công ty. Tuy nghiên với những kiến thức đã học trên trường vẫn chưa đủ tự tin để phát triển dự án của công ty, nhưng với sự động viên của những người anh đi trước đã giúp bản thân nổ lực và cuối cùng cũng hoàn thành website cho công ty Velox Tech, em đã học hỏi được rất nhiều từ quý công ty, từ những kinh nghiệm làm việc thực tế đến các cách ứng xử, thái độ giao tiếp trong môi trường làm việc văn phòng. Có cơ hội vận dụng được các kiến thức từ cơ bản đến nâng cao được đúc kết trong quá trình học tập tại trường Đại học Nha Trang. Môi trường làm việc ở Velox Tech giúp em phát triển các kỹ năng chuyên ngành mà mình đã học, nhìn nhận và khắc phục được các khuyết điểm còn tồn tại của bản thân. Các kết quả đạt được trong thời gian vừa qua đều là nhờ sự giúp đỡ nhiệt tình của các anh chị đi trước tại công ty Velox Tech trong quá trình làm việc, cùng với đó là sự hướng dẫn đầy tận tâm của thầy Đoàn Vũ Thịnh đã giúp em có thể hoàn thành bài báo cáo thực tập đúng kỳ hạn ở trường. Em thật sự rất biết ơn về sự hỗ trợ của quý thầy cô cùng với các anh chị.

Về dự án được phân công, tuy em chỉ được phân công một dự án nhỏ để đánh giá kỹ năng nhưng nó đã giúp em học hỏi được rất nhiều thứ. Tuy là một dự án nhỏ nhưng quy trình thực hiện và xây dựng chương trình thực tế trải qua rất nhiều công đoạn và được tiến hành một cách thật sự kỹ càng để có thể đưa ra một sản phẩm hoàn thiện cho khách hàng. Tuy một dự án nhỏ nhưng để chắc chắn nó hoàn thiện cần phải trải qua quá trình kiểm duyệt rất gắt gao, kỹ lưỡng và mất khá nhiều thời gian vì một dự án dù hoàn hiện tới đâu nhưng không thể nào tránh khỏi việc thiếu sót.

Qua quá trình thực tập em được tiếp xúc và học hỏi các công nghệ mới mà ở trường chưa được học như React JS và Node JS framework của Javascirp cùng với đó là hệ thống quản trị cơ sở dữ liệu MySQL. Biết thêm được các cách thức tổ chức code, comment ý tưởng cũng như phát triển thêm khả năng tư duy, giải quyết vấn đề trong quá trình làm việc. Được trau dồi thêm nhiều kinh nghiệm để sau này có thể thích nghi với công việc thực tế sau khi tốt nghiệp.

# TÀI LIỆU THAM KHẢO

- Trang tài liệu lập trình căn bản: <https://www.w3schools.com/>

- Trang hỏi đáp, hỗ trợ lập trình: <https://stackoverflow.com/>

- Trang hỗ trợ các thư viện lâp trình: <https://www.npmjs.com/>

- Trang hỗ trợ lập trình Node JS: <https://nodejs.org/docs/latest-v13.x/api/>