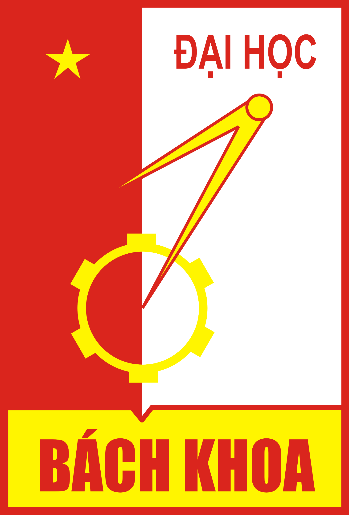
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN - TRUYỀN THÔNG**

-----🙞🙜🕮🙞🙜-----



**BÁO CÁO PROJECT III**

**MODULE I: PHÂN LOẠI/PHÂN CỤM VĂN BẢN, NER DỰA TRÊN HỌC MÁY**

**Giảng viên hướng dẫn :TS. Nguyễn Kim Hiếu**

**Sinh viên thực hiện :Lê Đức Tùng 2017 3458**

**Trần Đức Long 20183952**

**Mã lớp :709167**

**Hà Nội, 26 tháng 10 năm 2021**

**MỤC LỤC**

[**LỜI CẢM ƠN** 1](#_Toc81747326)

[**LỜI NÓI ĐẦU** 2](#_Toc81747327)

[ **GIỚI THIỆU BÀI TOÁN** 2](#_Toc81747328)

[ Mô tả bài toán thực tế: 2](#_Toc81747329)

[ Lý do chọn đề tài: 2](#_Toc81747330)

[ Các kịch bản sử dụng: 2](#_Toc81747331)

[ Đặc điểm quan trọng: 2](#_Toc81747332)

[ Yêu cầu cần đạt được: 2](#_Toc81747333)

[ Các tác nhân sử dụng: 2](#_Toc81747334)

[ **CÔNG NGHỆ** 2](#_Toc81747335)

[**1.** **GIỚI THIỆU LÝ THUYẾT, CÔNG NGHỆ SỬ DỤNG** 4](#_Toc81747336)

[**1.1.** **CƠ SỞ LÝ THUYẾT** 4](#_Toc81747337)

[1.1.1. Khái quát về Website hiện nay 4](#_Toc81747338)

[1.1.2. Khái niệm về Website quản lý 4](#_Toc81747339)

[1.1.3. Các yêu cầu với Website quản lý 4](#_Toc81747340)

[1.1.4. Các kiến thức để tạo ra một trang Web 4](#_Toc81747341)

[**1.2.** **KHÁI QUÁT CÁC CÔNG NGHỆ SỬ DỤNG** 5](#_Toc81747342)

[1.2.1. HTML, CSS, JAVASCRIPT 5](#_Toc81747343)

[**HTML** 5](#_Toc81747344)

[**CSS** 5](#_Toc81747345)

[**JAVASCRIPT** 6](#_Toc81747346)

[1.2.2. VUEJS 6](#_Toc81747347)

[1.2.2.1. VUEJS là gì ? 6](#_Toc81747348)

[1.2.2.2. Các cách cài đặt VUEJS 7](#_Toc81747349)

[1.2.2.3. Các kiến thức, khái niệm cần nắm vững của VUEJS 8](#_Toc81747350)

[1.2.3. MYSQL 9](#_Toc81747351)

[1.2.4. ASP.NET, API WEB ASP.NET 9](#_Toc81747352)

[**ASP.NET** 9](#_Toc81747353)

[**API Web ASP.NET** 10](#_Toc81747354)

[1.2.5. Các công nghệ hỗ trợ khác 11](#_Toc81747355)

[Session 11](#_Toc81747356)

[Cookie 11](#_Toc81747357)

[JWT 11](#_Toc81747358)

[**2.** **PHÂN TÍCH, THIẾT KẾ HỆ THỐNG** 12](#_Toc81747359)

[**2.1.** **Phân tích chức năng** 12](#_Toc81747360)

[2.1.1. Sơ đồ ca sử dụng 13](#_Toc81747361)

[2.1.1.1. Sơ đồ usecase tổng quan 13](#_Toc81747362)

[2.1.1.2. Sơ đồ phân rã User 13](#_Toc81747363)

[2.1.1.3. Sơ đồ phân rã Manager 14](#_Toc81747364)

[2.1.1.4. Sơ đồ phân rã Doctor 15](#_Toc81747365)

[2.1.1.5. Sơ đồ phân rã Nurse 15](#_Toc81747366)

[**2.2.** **Quy trình nghiệp vụ** 16](#_Toc81747367)

[2.2.1. Sơ đồ hoạt động 16](#_Toc81747368)

[2.2.1.1. Sơ đồ hoạt động thêm Vaccine 16](#_Toc81747369)

[2.2.1.2. Sơ đồ hoạt động thêm người dùng 16](#_Toc81747370)

[2.2.1.3. Sơ đồ hoạt động xóa Người dùng 17](#_Toc81747371)

[2.2.1.4. Sơ đồ hoạt động thay đổi thông tin 18](#_Toc81747372)

[2.2.1.5. Sơ đồ hoạt động thêm lịch tiêm 18](#_Toc81747373)

[**2.3.** **Thiết kế hệ thống** 19](#_Toc81747374)

[2.3.1. Mô hình hóa cấu trúc 19](#_Toc81747375)

[2.3.1.1. Ca sử dụng “Thêm người dùng” 19](#_Toc81747376)

[2.3.1.2. Ca sử dụng “Xóa người dùng” 20](#_Toc81747377)

[2.3.1.3. Ca sử dụng “Tìm kiếm người dùng” 20](#_Toc81747378)

[2.3.1.4. Ca sử dung “Thêm Vaccine” 20](#_Toc81747379)

[2.3.1.5. Ca sử dụng “Thêm lịch tiêm” 21](#_Toc81747380)

[2.3.1.6. Ca sử dụng “Xác nhận lịch tiêm” 21](#_Toc81747381)

[2.3.1.7. Ca sử dụng “Xóa lịch tiêm vì quá hẹn hoặc không đảm bảo” 22](#_Toc81747382)

[2.3.1.8. Ca sử dụng “Kiểm tra thông tin tiêm chủng cá nhân” 22](#_Toc81747383)

[**2.4.** **Sơ đồ lớp, thiết kế Database** 23](#_Toc81747384)

[2.4.1. Kiến trúc tổng thể Back-end 23](#_Toc81747385)

[2.4.2. Đặc tả bảng dữ liệu: 24](#_Toc81747386)

[**2.4.2.1.** **Mô tả CSDL** 24](#_Toc81747387)

[**2.4.2.2.** **Lược đồ cơ sở dữ liệu** 25](#_Toc81747388)

[2.4.3. Sơ đồ quan hệ bảng dữ liệu: 25](#_Toc81747389)

[**3.** **CHƯƠNG III: HƯỚNG DẪN CÀI ĐẶT MÔI TRƯỜNG, KIỂM THỬ** 25](#_Toc81747390)

[**3.1.** **Hướng dẫn cài đặt môi trường** 25](#_Toc81747391)

[**3.2.** **Kiểm thử** 26](#_Toc81747392)

[3.2.1. Kiểm thử chức năng 26](#_Toc81747393)

[3.2.1.1. Test link 26](#_Toc81747394)

[3.2.1.2. Test form 26](#_Toc81747395)

[3.2.1.3. Test Cookies 27](#_Toc81747396)

[3.2.2. Kiểm thử tính khả dụng 27](#_Toc81747397)

[3.2.3. Kiểm thử Database 27](#_Toc81747398)

[3.2.4. Kiểm thử khả năng tương thích 28](#_Toc81747399)

[3.2.5. Kiểm thử bảo mật 28](#_Toc81747400)

[**KẾT LUẬN** 29](#_Toc81747401)

[Các công việc đã thực hiện 29](#_Toc81747402)

[Kết quả đã hoàn thành 29](#_Toc81747403)

[Các khó khăn trong quá trình làm bài 29](#_Toc81747404)

[Khả năng phát triển, triển vọng của đồ án 29](#_Toc81747405)

**DANH MỤC HÌNH ẢNH**

[Ảnh 1.2.5. 1 12](#_Toc81747406)

[Ảnh 2.1.1. 1 13](#_Toc81747407)

[Ảnh 2.1.1. 2 14](#_Toc81747408)

[Ảnh 2.1.1. 3 15](#_Toc81747409)

[Ảnh 2.1.1. 4 15](#_Toc81747410)

[Ảnh 2.1.1. 5 15](#_Toc81747411)

[Ảnh 2.2.1 1 16](#_Toc81747412)

[Ảnh 2.2.1 2 17](#_Toc81747413)

[Ảnh 2.2.1 3 17](#_Toc81747414)

[Ảnh 2.2.1 4 18](#_Toc81747415)

[Ảnh 2.3.1. 1 19](#_Toc81747416)

[Ảnh 2.3.1. 2 20](#_Toc81747417)

[Ảnh 2.3.1. 3 20](#_Toc81747418)

[Ảnh 2.3.1. 4 21](#_Toc81747419)

[Ảnh 2.3.1. 5 21](#_Toc81747420)

[Ảnh 2.3.1. 6 22](#_Toc81747421)

[Ảnh 2.3.1. 7 22](#_Toc81747422)

[Ảnh 2.3.1. 8 23](#_Toc81747423)

[Ảnh 2.3.1. 9 24](#_Toc81747424)

[Ảnh 2.3.3. 1 25](#_Toc81747436)

[Ảnh 3.2.1.2. 1 27](#_Toc81747456)

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đối với Thầy Bùi Trọng Tùng đã tạo điều kiện cho em nhận được nhiều sự giúp đỡ, đóng góp ý kiến và nhiệt tình hướng dẫn em hoàn thành môn học Project II theo đúng tiến độ và có kết quả tốt.

Trong quá trình làm đồ án, cũng như là trong quá trình làm bài báo cáo, khó tránh khỏi sai sót, rất mong Thầy bỏ qua. Đồng thời do trình độ cũng như kinh nghiệm của em còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp của Thầy để em học thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn các môn khác trong những kỳ học sắp tới và xa hơn là cho công việc sau này.

**LỜI NÓI ĐẦU**

* **GIỚI THIỆU BÀI TOÁN**
  + **Mô tả bài toán thực tế:**
    - **Lý do chọn đề tài:**

Trong tình hình dịch bệnh diễn biến phức tạp như hiện nay, nhu câu thiết yếu đặt ra yêu cầu cần có một website có khả năng giúp cho chính quyền và người dùng có thể dễ dàng quản lý, cập nhật thông tin và kiểm tra thông tin về vaccine Covid-19.

* + - **Các kịch bản sử dụng:**
* Các cơ quan chính quyền, người quản lý có thể truy cập vào website kiểm tra thông tin tiêm chủng của một khu vực mà mình quản lý, kiểm tra thông tin vaccine, thông tin những người đã được tiêm, chưa tiêm, tình trạng sức khỏe sau tiêm, …
* Người dùng có thể vào để xem thông tin, chỉnh sửa thông tin cá nhân của mình cho đúng, kiểm tra lịch tiêm lần tiếp theo (nếu có), kiểm tra thông tin về vaccine, …
* Người dùng liên hệ với người quản lý để sửa, cập nhật thông tin.
  + - **Đặc điểm quan trọng:**

Cung cấp đầy đủ thông tin cho người dùng, người được quản lý và giúp dễ dàng xử lý hơn cho người quản lý.

* + - **Yêu cầu cần đạt được:**
* Thiết kế thành công một website quản lý thông tin cơ bản với các chức năng : Thêm, xóa, sửa, đăng nhập,…
* Làm quen với công cụ, công nghệ mới để hoàn thành công việc.
* Làm việc nghiêm túc, hiệu quả, đảm bảo tiến trình công việc với giảng viên.
* Hiểu, nắm chắc các kiến thức cơ bản về cách làm và hoạt động của một website.
  + - **Các tác nhân sử dụng:**
* Người quản lý (quyền Admin)
* Người dùng (User)
* Bác sĩ (Doctor)
* Y tá (Nurse)
* **CÔNG NGHỆ**

Giới thiệu tổng quát công nghệ sử dụng trong Project bao gồm :

* Front End : Sử dụng HTML, CSS, JS và các thư viện hỗ trợ ngày tháng như DayJs và sử dụng Framwork của Javascript là VUEJS.
* Back End: Sử dụng các kiến thức cơ bản về OOP để áp dụng lập trình với ngôn ngữ C#, sử dụng .NET Core, ASP.NET và các kiến thức về RESTFUL API.
* Database : Sử dụng hệ quản trị cơ sở dữ liệu MySQL trên nền tảng PHPMyAdmin (Quản lý SQL trên giao diện Web dùng ứng dụng XAMPP)
* Công nghệ khác : Sử dụng các kiến thức liên quan đến Session, Cookie để xử lý, lưu trữ dữ liệu và JWT ( Mã hóa thông tin theo thuật toán Hs256) dùng để xác thực.

1. **GIỚI THIỆU LÝ THUYẾT, CÔNG NGHỆ SỬ DỤNG**
   1. **CƠ SỞ LÝ THUYẾT**
      1. **Khái quát về Website hiện nay**

Trong thời đại công nghệ phát triển 4.0 như hiện nay, khoa học kĩ thuật ngày càng phát triển kéo theo nhu cầu của con người càng lên cao. Đặc biệt với sự ra đời của Internet đã giúp cuộc sống của con người có thêm rất nhiều tiện ích, theo khảo sát của VECTA( Cục thương mại điện tử và công nghệ thông tin ) Việt Nam có đến 68,17 triệu người sử dụng Internet một con số rất lớn so với dân số Việt Nam hiện nay.

Đặc biệt trong thời kì dịch bệnh như hiện nay, nhu cầu của con người càng nâng cao nhu cầu sử dụng Internet càng nhiều và từ đó khái niệm Website ngày càng phổ biến với mỗi người.

* + 1. **Khái niệm về Website quản lý**

Website càng ngày càng phổ biến do tính đa dụng và tiện ích nó mang lại cho người dùng, và một trong các loại trang web phổ biến nhất hiện nay chính là website quản lý.

Website quản lý là một loại trang web cho phép người dùng sử dụng, thay đổi, quản lý các thông tin mà mình lưu trữ, nhân được từ phía người dùng thông qua giao diện web. Người dùng có thể truy cập trang web với đúng vai trò của mình thực thi các hành động phù hợp để dễ dàng hoàn thành các yêu cầu hay đạt được các nhu cầu của bản thân. Website quản lý giúp người có quyền hành dễ quản lý, giúp người được quản lý dễ dàng sử dụng và thực hiện các hành động của mình.

* + 1. **Các yêu cầu với Website quản lý**
* Cơ sở hạ tầng bảo mật thông tin.
* Tạo mối quan hệ bằng sự tin cậy.
* Bảo mật và an toàn.
* Bảo vệ quyền lợi người dùng và sử dụng đúng chức trách người quản lý.
  + 1. **Các kiến thức để tạo ra một trang Web**
* Có kiến thức cơ bản về website
* Hiểu biết về phân tích, thiết kế
* Hiểu biết về thiết kế giao diện, API, các thành phần liên quan
* Biết sử dụng các hệ quản trị cơ sở dữ liệu
  1. **KHÁI QUÁT CÁC CÔNG NGHỆ SỬ DỤNG**
     1. **HTML, CSS, JAVASCRIPT**

**HTML**

HTML (HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản) là một [ngôn ngữ đánh dấu](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_%C4%91%C3%A1nh_d%E1%BA%A5u) được thiết kế ra để tạo nên các [trang web](https://vi.wikipedia.org/wiki/Website) với các mẩu thông tin được trình bày trên [World Wide Web](https://vi.wikipedia.org/wiki/World_Wide_Web). Cùng với [CSS](https://vi.wikipedia.org/wiki/CSS) và [JavaScript](https://vi.wikipedia.org/wiki/JavaScript), HTML tạo ra bộ ba nền tảng kỹ thuật cho các website.

[HTML](https://en.wikipedia.org/wiki/HTML) là cốt lõi của mọi trang web. Bất kể sự phức tạp của một trang web hoặc số lượng công nghệ liên quan. Đó là một kỹ năng thiết yếu cho bất kỳ chuyên gia web. Đó là điểm khởi đầu cho bất cứ ai học cách tạo nội dung cho web. Và thật may mắn cho những bạn mới bắt đầu là HTML rất dễ học.

Mỗi trang web được tạo thành từ một loạt các thẻ HTML này biểu thị từng loại nội dung trên trang. Mỗi loại nội dung trên trang được “bọc”, tức là được bao quanh bởi các thẻ HTML.

Sử dụng HTML, bạn có thể thêm tiêu đề, định dạng đoạn văn, ngắt dòng điều khiển. Tạo danh sách, nhấn mạnh văn bản, tạo ký tự đặc biệt, chèn hình ảnh, tạo liên kết. Hoặc xây dựng bảng, điều khiển một số kiểu dáng và nhiều hơn thế nữa.

HTML không phải là ngôn ngữ lập trình, đồng nghĩa với việc nó không thể tạo ra các chức năng “động” được. Nó chỉ giống như Microsoft Word, dùng để bố cục và định dạng trang web.

Tổng quan, HTML là ngôn ngữ markup, nó rất trực tiếp dễ hiểu, dễ học, và tất cả mọi người mới đều có thể bắt đầu học nó để xây dựng website. Một trang web được viết ra bằng HTML thì nó rất đơn giản. Chỉ có nội dung và bố cục. Nghe có vẻ tẻ nhạt đúng không? Và để cho trang web thêm sinh động và phù hợp với người dùng thì chúng ta cần sử dụng nhiều hơn việc chỉ sử dụng HTML. Khi đó chúng ta cần dùng đến CSS và JavaScript.

**CSS**

[CSS](https://vi.wikipedia.org/wiki/CSS) là viết tắt của Cascading Style Sheets. Ngôn ngữ lập trình này chỉ ra cách các yếu tố HTML của trang web thực sự sẽ xuất hiện trên giao diện của trang. Nếu HTML cung cấp các công cụ thô cần thiết để cấu trúc nội dung trên một trang web thì CSS sẽ giúp định hình kiểu nội dung này để trang web xuất hiện trước người dùng theo một cách đẹp hơn. Bạn có thể hiểu là nếu HTML là tường gạch thô thì CSS là sơn để trang trí cho tường gạch đó. Các ngôn ngữ này được giữ riêng biệt để đảm bảo các trang web được xây dựng chính xác trước khi chúng được định dạng lại.

Trong khi HTML là cấu trúc cơ bản của trang web của bạn. CSS mang lại cho trang web của bạn phong cách mà bạn muốn.  Những màu sắc đặc trưng, font chữ phù hợp, và hình ảnh nền của website? Tất cả là nhờ CSS. CSS gần như tạo nên bộ mặt của một website. Và nếu trang web của bạn ưa nhìn thì nó sẽ hấp dẫn và lôi cuốn được người dùng phải không.

Để có một trang web đẹp bạn không chỉ dựa vào CSS. Mà các ngôn ngữ phải được thực hiện dựa trên sự sáng tạo và các bản thiết kế bạn tạo ra trước đó. Ngoài ra, bạn cần phải kết hợp nó với nhiều hơn các ngôn ngữ khác nữa.

**JAVASCRIPT**

JavaScript là ngôn ngữ phức tạp hơn HTML hay CSS. Và nó không được phát hành ở dạng beta cho đến năm 1995. Ngày nay, JavaScript được hỗ trợ bởi tất cả các trình duyệt web hiện đại. Và được sử dụng trên hầu hết mọi trang web trên web để có chức năng mạnh mẽ và phức tạp hơn.

JavaScript là ngôn ngữ lập trình dựa trên logic. Nó có thể được sử dụng để sửa đổi nội dung trang web. Và khiến nó hoạt động theo nhiều cách khác nhau để đáp ứng với hành động của người dùng. Các cách sử dụng phổ biến cho JavaScript bao gồm các hộp xác nhận, kêu gọi hành động và thêm các danh tính mới vào thông tin hiện có.

Nói tóm lại, JavaScript là ngôn ngữ lập trình cho phép các nhà phát triển web thiết kế các trang web tương tác. Hầu hết các hành vi động mà bạn sẽ thấy trên một trang web là nhờ JavaScript. Nó giúp tăng cường các hành vi và kiểm soát mặc định của trình duyệt. Ngôn ngữ lập trình này có thể làm rất nhiều thứ trên trang web của bạn. Nó làm cho mọi thứ trở nên tiện nghi hơn bao giờ hết. Tuy nhiên để học được ngôn ngữ này bạn cần phải rất kiên trì. Bởi đây là ngôn ngữ khá phức tạp và khó để học.

* + 1. **VUEJS**
       1. **VUEJS là gì ?**

VUEJS được gọi tắt là Vue (được phát âm là /vjuː/, nó giống như từ view trong tiếng Anh). Vue.js là một framework rất linh động được dùng phổ biến để xây dựng nên các giao diện người dùng. Hoàn toàn khác với các framework nguyên khối thì Vue thường sở hữu thiết kế từ đầu theo những hướng cho phép cũng như khuyến khích làm việc để phát triển dễ dàng hơn các ứng dụng theo từng bước một.

Một  khi đã phát triển lớp giao diện (view layer) thì người dùng chỉ cần sử dụng loại thư viện lõi của Vue. Ngoài ra, nếu như bạn kết hợp với các kỹ thuật thiên hướng hiện đại thì Vue cũng có thể đáp ứng được dễ dàng mọi nhu cầu xây dựng ứng dụng của một trang với độ phức tạp cao hơn.

* + - 1. **Các cách cài đặt VUEJS**

**Cách 1:** Sử dụng CDN(Content Delivery Network)

Đây được đánh giá là một trong những phương pháp nhanh gọn nhất cho việc cài đặt Vue. Bởi bạn chỉ cần 1 đường link rồi click vào đến phiên bản mới và ổn định nhất của Vuejs là có thể cài đặt. Các đường link này hiện đã có mặt rất nhiều trên các trang mạng hãy lựa chọn địa chỉ uy tín rồi mới cài đặt.

**Cách 2:** NPM

Nếu như muốn xây dựng một ứng dụng lớn với Vue thì lựa chọn NPM là vô cùng hợp lý. NPM thường có thể hoạt động rất tốt với những module bundler như Webpack hoặc Browsertify để từ đó cung cấp được những công cụ có thể hỗ trợ để viết Single File Component. Có thể sử dụng câu lệnh “npm install vue”  để cài đặt Vue.

**Cách 3:** CLI

Hiện nay, Vue.js đang cung cấp một CLI có khả năng khởi tạo nhanh chóng các ứng dụng cho một trang. Bên trong CLI đã có chuyển bị sẵn những cài đặt phong phú nhằm đảm bảo cho quy trình front-end được hiện đại nhất. Và chỉ mất vài phút thì bạn đã có thể bắt đầu cho việc xây dựng ứng dụng với những tính năng như: hot-reload, tinh chỉnh code khi save và một số bản build đã sẵn sàng để thực hiện deploy lên production.

Bạn chỉ cần thực hiện như sau:

+ Tiến hành cài đặt vue-cli: npm install –global vue-cli

+ Tạo mới một dự án với template webpack: vue init webpack my-project

+  Tiếp tục cài đặt thêm một số thư viện liên quan rồi tiến hành chạy thử.

* + - 1. **Các kiến thức, khái niệm cần nắm vững của VUEJS**

**ViewModel ( MVVM patterns)**

Hầu hết, một Vue instance đều có thể sẽ được tạo ra bằng việc phát hành new Vue với các cách tùy chọn có thể chứa những yếu tố sau: template,các callbacks, data, methods và event. Nói một cách đơn giản hơn là bạn có thể xem Vue như một trung gian nằm giữa data và view.

**Vue Instance**

Trước khi xây dựng một ứng dụng VueJs đầu tiên thì mỗi ứng dụng Vue sở hữu phiên bản gốc được gọi là Vue và Vue Instance hoạt động tuân theo Model-view.

**Data binding trong VUEJS**

Thông thường sẽ có hai cách bind data để view trong Vue là one-way data binding và two-way data binding. 2 cách này sẽ được nhắc rõ hơn ngay sau đây:

+ Về One-way data binding: Nó liên kết trực tiếp với data trực tiếp từ code javascript của với DOM. Hầu hết những props bên trong sẽ không có khả năng làm thay đổi được cấp cha.

+ Về Two-way data binding: Thông thường, bind data từ code [Javascript](https://itnavi.com.vn/blog/ngon-ngu-lap-trinh-javascript-la-gi-su-khac-nhau-giua-java-va-javascript/" \t "_blank) của bạn để có thể view và từ đó view lại code để có thể thay đổi data ở một bên dẫn làm thay đổi global. For two-way data binding, Vue cung cấp một v-model cho mục đích này. Hãy để trở lại form feedback của chúng ta và thực hiện một số thay đổi.

– Đối với Directives : Directives là một trong những tính năng mạnh mẽ tồn tại bên trong Vue và chúng là những expressions binding với v-attribute. Bên cạnh đó thì các directive thường được cung cấp dưới dạng mặc định (v-model và v-show). Từ đó, Vue cho phép bạn có thể đăng ký các directive tùy biến dành riêng cho mình.

Bạn cần lưu ý với phiên bản Vue 2.0 thì cách tốt nhất để có thể sử dụng lại và trừu tượng code thì nên dùng component. Tuy nhiên, bạn cần phải có một số truy xuất cấp thấp đến từ những phần tử của trang web. Với những trường hợp này thì việc  directive tùy biến sẽ trở nên hữu ích hơn rất nhiều.

Ngoài ra, Vue Instance, Data binding và Directives còn được đánh giá với ba nguyên tắc cơ bản của VueJs. Để có thể hiểu rõ hơn về các thành phần, đặc tính cũng như định nghĩa khác liên quan đến Vue thì bạn nên tham khảo nhiều hơn các tài liệu trên mạng internet hoặc các anh chị đi trước.

Chắc chắn rằng, hiện nay Vue.js đã không còn quá đỗi xa lạ đối với nhiều lập trình viên. Nó được xem là một Progressive framework sở hữu kích thước nhỏ nhẹ và chỉ cần kết hợp sử dụng Virtual DOM và chỉ làm render lại cho những thành phần có sự thay đổi khác. Nó còn sở hữu cú pháp template rõ ràng cũng như đơn giản nn người sử dụng có thể dễ dàng tiếp cận.

Một điểm nữa, là hệ thống component thường cho phép bạn tái sử dụng ở những template khác nhau. Chính vì vậy, VueJS là một trong những lựa chọn hoàn hảo cho nền công nghệ mới mà ai cũng nên áp dụng.

* + 1. **MYSQL**

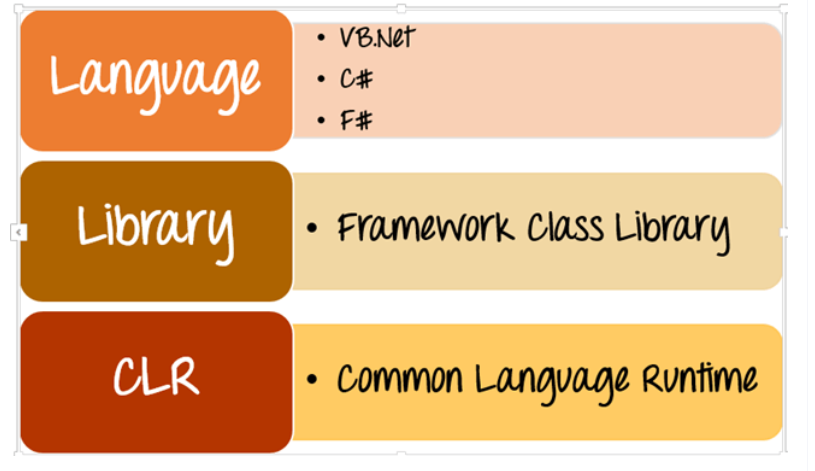
MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (Relational Database Management System, viết tắt là RDBMS) hoạt động theo mô hình client-server. [RDBMS](https://en.wikipedia.org/wiki/Relational_database_management_system) là một phần mềm hay dịch vụ dùng để tạo và quản lý các cơ sở dữ liệu (Database) theo hình thức quản lý các mối liên hệ giữa chúng.

MySQL là một trong số các phần mềm RDBMS. RDBMS và MySQL thường được cho là một vì độ phổ biến quá lớn của MySQL. [Các ứng dụng web lớn nhất](https://stackshare.io/mysql) như Facebook, Twitter, YouTube, Google, và Yahoo! đều dùng MySQL cho mục đích lưu trữ dữ liệu. Kể cả khi ban đầu nó chỉ được dùng rất hạn chế nhưng giờ nó đã tương thích với nhiều hạ tầng máy tính quan trọng như Linux, macOS, Microsoft Windows, và Ubuntu.

* + 1. **ASP.NET, API WEB ASP.NET**

**ASP.NET**

ASP.NET là một nền tảng dành cho phát triển web, được Microsoft phát hành và cung cấp lần đầu tiện vào năm 2002. Nền tảng được sử dụng để tạo ra các ứng dụng web-based. ASP là viết tắt của Active Server Pages, và .NET là viết tắt của Network Enabled Technologies. ASP.NET là một framework được sử dụng để phát triển các ứng dụng web-based với cấu trúc như sau :



Ảnh ‑1.2.4.1

**Language/Ngôn ngữ:** Có rất nhiều ngôn ngữ lập trình khác nhau tồn tại trong .net framework. Các ngôn ngữ này là VB.net và C #, có thể được sử dụng để phát triển các ứng dụng web.

**Library/Thư viện:** .NET Framework gồm có một bộ các lớp library chuẩn. Library phổ biến nhất được sử dụng cho các ứng dụng web trong .net là Web library. Web library bao gồm tất cả các thành phần cần thiết sử dụng trong phát triển các ứng dụng web-based.

**Common Language Runtime/Thời gian chạy ngôn ngữ lập trình thông thường hay CLR:** Common Language Infrastructure - Cơ sở hạ tầng ngôn ngữ lập trình phổ thông hay CLI là một nền tảng dùng để chạy các chương trình .Net. Trong đó, CLR sẽ thực hiện các tác vụ chính bao gồm xử lý các trường hợp cá biệt và thu gom rác.

**API Web ASP.NET**

Thuật ngữ API là viết tắt của Application Programming Interface – Giao diện lập trình ứng dụng. ASP.NET Web API là một framework (khung làm việc), được cung cấp bởi Microsoft, giúp dễ dàng xây dựng API Web, tức các dịch vụ dựa trên giao thức HTTP. ASP.NET Web API là một nền tảng lý tưởng để xây dựng các dịch vụ Restful trên đỉnh .NET Framework. Các dịch vụ API Web này có thể được sử dụng bởi nhiều client khác nhau, chẳng hạn như:  
Trình duyệt, ứng dụng di động, …

* + 1. **Các công nghệ hỗ trợ khác**

**Session**

**Session** là một khái niệm phổ biến được dùng trong lập trình các website có kết nối với cơ sở dữ liệu database. Đặc biệt các chức năng như đăng nhập, đăng xuất người dùng sẽ khó có thể thực hiện được nếu không sử dụng **session**.

**Session** đơn giản là 1 cách để chúng ta lưu lại dữ liệu của người dùng sử dụng website. Giá trị của **session** được lưu trong một tập tin trên máy chủ. Ví dụ khi bạn đăng nhập vào một trang web và đăng nhập với tài khoản đã đăng ký trước đó. Máy chủ sau khi xác thực được thông tin bạn cung cấp là đúng nó sẽ sinh ra một tập tin (hay chính là **session** của trình duyệt của bạn) chứa dữ liệu cần lưu trữ của người dùng.

**Cookie**

**Cookie** là một đoạn văn bản ghi thông tin được tạo ra và lưu trên trình duyệt của máy người dùng. **Cookie** thường được tạo ra khi người dùng truy cập một website, **cookie** sẽ ghi nhớ những thông tin như tên đăng nhập, mật khẩu, các tuỳ chọn do người dùng lựa chọn đi kèm. Các thông tin này được lưu trong máy tính để nhận biết người dùng khi truy cập vào một trang web.

Khi người dùng truy cập đến một trang web có sử dụng **cookie**, web server của trang đó sẽ tự động gửi **cookie** đến máy tính của người dùng. Những **cookie** này tự động được tổ chức trong hệ thống máy tính. Khi truy cập đến các trang web sử dụng được **cookie** đã lưu, những **cookie** này tự động gửi thông tin của người dùng về cho chủ của nó (người tạo ra **cookie**). Tuy nhiên những thông tin do **cookie** ghi nhận không được tiết lộ rộng rãi, chỉ có website chứa **cookie** mới có thể xem được những thông tin này. **Cookie** được xem là một thành phần không thể thiếu được với những website có khối lượng dữ liệu lớn, có số lượng người dùng đông, và có những chức năng đi kèm với thành viên đăng ký. Phần lớn các website này là các website thương mại điện tử.

**JWT**

[**JWT**](https://topdev.vn/blog/?s=JSON+Web+Token) là một phương tiện đại diện cho các yêu cầu chuyển giao giữa hai bên Client – Server , các thông tin trong chuỗi **JWT** được định dạng bằng [JSON](https://topdev.vn/blog/json-la-gi/) . Trong đó chuỗi Token phải có 3 phần là header , phần payload và phần signature được ngăn bằng dấu “.”

JWT được dùng để xác thực người dùng và cấp quyền phù hợp cho người dùng bằng cách thêm vào mã JWT.

Cấu trúc JWT:



Ảnh 1.2.5. 1

1. **PHÂN TÍCH, THIẾT KẾ HỆ THỐNG**
   1. **Phân tích chức năng**
      1. **Sơ đồ ca sử dụng**
         1. *Sơ đồ usecase tổng quan*

*Diagram

Description automatically generated*

Ảnh 2.1.1. 1

* + - 1. *Sơ đồ phân rã User*

*Diagram

Description automatically generated*

Ảnh 2.1.1. 2

* + - 1. *Sơ đồ phân rã Manager*

*Diagram

Description automatically generated*

Ảnh 2.1.1. 3

* + - 1. *Sơ đồ phân rã Doctor*

Diagram

Description automatically generated

Ảnh 2.1.1. 4

* + - 1. *Sơ đồ phân rã Nurse*

*Diagram

Description automatically generated*

Ảnh 2.1.1. 5

* 1. **Quy trình nghiệp vụ**
     1. **Sơ đồ hoạt động**
        1. *Sơ đồ hoạt động thêm Vaccine*

*A picture containing diagram

Description automatically generated*

Ảnh 2.2.1 1

* + - 1. *Sơ đồ hoạt động thêm người dùng*

*A picture containing table

Description automatically generated*

Ảnh 2.2.1 2

* + - 1. *Sơ đồ hoạt động xóa Người dùng*

*A picture containing table

Description automatically generated*

Ảnh 2.2.1 3

* + - 1. *Sơ đồ hoạt động thay đổi thông tin*

*Diagram

Description automatically generated*

Ảnh 2.2.1 4

* + - 1. *Sơ đồ hoạt động thêm lịch tiêm*

*A picture containing calendar

Description automatically generated*

Ảnh 2.2.1 5

* + - 1. *Sơ đồ hoạt động xóa lịch tiêm*

*Diagram

Description automatically generated*

Ảnh 2.2.1 6

* 1. **Thiết kế hệ thống**
     1. **Mô hình hóa cấu trúc**
        1. **Ca sử dụng “Thêm người dùng”**

**Diagram

Description automatically generated**Ảnh 2.3.1. 1

* + - 1. **Ca sử dụng “Xóa người dùng”**

**Diagram

Description automatically generated**

Ảnh 2.3.1. 2

* + - 1. **Ca sử dụng “Tìm kiếm người dùng”**

**Diagram

Description automatically generated**

Ảnh 2.3.1. 3

* + - 1. **Ca sử dung “Thêm Vaccine”**

**Diagram

Description automatically generated**

Ảnh 2.3.1. 4

* + - 1. **Ca sử dụng “Thêm lịch tiêm”**

**Diagram

Description automatically generated**

Ảnh 2.3.1. 5

* + - 1. **Ca sử dụng “Xác nhận lịch tiêm”**

**Diagram

Description automatically generated**

Ảnh 2.3.1. 6

* + - 1. **Ca sử dụng “Xóa lịch tiêm vì quá hẹn hoặc không đảm bảo”**

**Diagram

Description automatically generated**

Ảnh 2.3.1. 7

* + - 1. **Ca sử dụng “Kiểm tra thông tin tiêm chủng cá nhân”**

**Diagram

Description automatically generated**

Ảnh 2.3.1. 8

* 1. **Sơ đồ lớp, thiết kế Database**
     1. **Kiến trúc tổng thể Back-end**

Sử dụng kiến thức đa lớp (N- layer), ta có thể tách và nhóm các class thành 3 nhóm chính :

* Giao diện người dùng (User Interface)
* Xử lý nghiệp vụ (Business Logic)
* Truy xuất dữ liệu (Data Access)

Mô hình cấu trúc :

Graphical user interface, application

Description automatically generated

Ảnh 2.3.1. 9

* + 1. **Đặc tả bảng dữ liệu:**
       1. **Mô tả CSDL**

Một chương trình dùng để quản lý thông tin tiêm chủng của một nhóm, bộ phân người dùng. Sau đây là mô tả theo các nghiệp vụ của hệ thống:

Hệ thống được chia làm 4 vai trò : Người dùng (User), Bác sĩ (Doctor), Quản trị viên (Admin), Y tá (Nurse). Các vai trò này được quy định và khởi tạo ở bảng Authority trong cơ sở dữ liệu với trường NAME [varchar(50)] làm khóa chính.

Mỗi đối tượng được phân biệt bằng một mã riêng để phân biệt với nhau, kể cả vaccine hay lịch tiêm cũng được phân biệt bằng cách như vậy.

Mỗi người muốn sử dụng hệ thống sẽ phải đăng nhập bằng tài khoản và mật khẩu được cấp sẵn chứ không thể đăng ký.

Quản trị viên có thể quản lý thông tin người dùng, vaccine và xem lịch tiêm, hẹn tiêm của các thành viên quản lý về : Tên, ngày sinh, họ tên, giới tính, địa chỉ, số điện thoại và các thông tin liên quan đến quá trình tiêm.

Người dùng có thể xem thông tin của riêng mình, cập nhật sửa đổi cho đúng và người dùng có thể kiểm tra lịch tiêm của mình, cập nhật tình trạng sức khỏe của bản thân.

Bác sĩ có thể cập nhật thông tin sức khỏe, thêm sửa xóa lịch tiêm cho người dùng và kiểm tra thông tin vaccine.

Y tá có thể xác nhận các thông tin tiêm chủng, quản lý lịch tiêm, lịch hẹn.

* + - 1. **Lược đồ cơ sở dữ liệu**

Authority (Name)

User (UserId, FullName, Gender, Address, PhoneNumber, DateOfBirth, Account, Password, Authority\_Name)

Vaccine (VaccineId, Vaccine\_Name, Vaccine\_Amount, Vaccine\_Origin, Date\_Of\_Manufacture)

Vaccination\_Information (Vaccination\_InformationId, UserId,

Vaccination\_Date, Vaccine\_Name, Health\_Condition)

Vaccination\_Appointment (Vaccination\_ AppointmentId, UserId,

Vaccination\_ Appointment \_Date, Vaccine\_Name, Health\_Condition)

* + 1. **Sơ đồ quan hệ bảng dữ liệu:**

**Graphical user interface, application

Description automatically generated**

Ảnh 2.3.3. 1

1. **CHƯƠNG III: HƯỚNG DẪN CÀI ĐẶT MÔI TRƯỜNG, KIỂM THỬ**
   1. **­Hướng dẫn cài đặt môi trường**

**Bước 1 :** Chọn thư mục chứa source code, trước đó đã cài đặt GIT cho máy click chuột phải chọn Git Bash Here và nhập lệnh : git clone

[https://github.com/chukun09/VaccinationManager](%20https://github.com/chukun09/VaccinationManager)

**Bước 2:** Cài đặt Visual Studio Code và Visual Studio 2019 để chạy code và XAMPP để sử dụng Database.

**Bước 3:** Khởi động XAMPP bật MYSQL và APACHE tiếp theo mở code API bằng Visual Studio Code 2019 và Visual Studio Code thì mở code Front-end

**Bước 4:** chạy lệnh npm install trên Visual Studio Code để cài đặt các thư viện cần thiết đã được thiết lập, bên phía backend dùng tổ hợp phím ctrl F5 để chạy backend.

**Bước 5:** Sau khi chạy được backend, và front-end đã cài đặt xong. Ta chạy lệnh npm run serve để chạy chương trình.

**Bước 6:** Sử dụng chương trình web và kiểm tra lại nếu có lỗi xảy ra. Nếu không được có thể liên hệ tác giả qua email: [longtd183952@sis.hust.edu.vn](mailto:longtd183952@sis.hust.edu.vn) hoặc Facebook : Trần Đức Long.

* 1. **Kiểm thử**
     1. **Kiểm thử chức năng**
        1. **Test link**

Kiểm tra tất cả các link trong web, đảm bảo các link hoạt động chính xác không có link nào bị chết bị hỏng.

* + - 1. **Test form**

Các trường của form hoạt động đúng, có validate dữ liệu, các thao tác trên các trường ổn định, form có giao diện thân thiện, dễ nhìn.

Hình ảnh mình họa:

Graphical user interface, application

Description automatically generated

Ảnh 3.2.1.2. 1

* + - 1. **Test Cookies**

Cookie sẽ xóa khi xóa bộ nhớ cache hoặc hết thời gian đúng như cài đặt.

Sau khi xóa cookie, sẽ yêu cầu thông tin đăng nhập khi truy cập lần sau.

( Với điều kiện người dùng chọn Remember Me)

* + 1. **Kiểm thử tính khả dụng**
* Tính khả dụng của trang web được định nghĩa là trang web dễ sử dụng, có hướng dẫn sử dụng rõ ràng, rành mạch, mỗi trang đều có menu chính và menu này phải nhất quán
* Kiểm tra Navigation: Menu, button, textbox, link đến các trang khác nhau trên trang web dễ nhìn thấy và nhất quán trên tất cả các trang web.
* Kiểm tra nội dung: nội dung dễ đọc không có lỗi chính tả hoặc ngữ pháp, thân thiện với người dùng. Hình ảnh được sắp xếp gọn gàng, hợp lý.
  + 1. **Kiểm thử Database**
* Không xảy ra lỗi khi truy vấn Database, nếu có lỗi khi gọi API truy vấn database sẽ có respone dạng chuẩn Restful.
* Đảm bảo an toàn dữ liệu, tính toàn vẹn được duy trì khi tạo, xóa, sửa bản ghi.
* Đảm bảo tính chính xác của dữ liệu.
  + 1. **Kiểm thử khả năng tương thích**
* Tương tích với hầu hết các trình duyệt ( IE, Chorme, Firefox, ..)
* Chưa thể kiểm thử trên các hệ điều hành, các thiết bị khác.
  + 1. **Kiểm thử bảo mật**
* Gõ trực tiếp URL vào thanh địa chỉ của trình duyệt mà không qua đăng nhập. Các trang nội bộ không thể truy cập vào bằng cách gõ trực tiếp mà sẽ trả về trang login.
* Sau khi đăng nhập và mở các trang nội bộ, thay đổi url trực tiếp bằng cách đổi tham số ID của trang tới trang thuộc quyền người dùng đã đăng nhập khác. Truy cập phải bị từ chối bởi người dùng này không thể xem trang thống kê của người dùng khác.
* Không thể thao tác các quyền bị hạn chế.
* Sessions sẽ tự động bị hủy sau khi người dùng không hoạt động trong một thời gian
* Nhập các giá trị đầu vào không hợp lệ trong các trường Username, Password thì hệ thống phải báo lỗi.

**KẾT LUẬN**

**Các công việc đã thực hiện**

* Phân tích bài toán, thiết kế cơ sở dữ liệu (Tuần 1)
* Phân tích các yêu cầu, chức năng, vẽ biểu đồ Usecase và biểu đồ hoạt động (Tuần 2)
* Tìm hiểu các khái niệm về Authentication, JWT, Session, Cookie và bắt tay vào code phần back-end, phân chia cấu trúc theo cấu trúc phân tầng (N-Layer) (Tuần 3)
* Tìm hiểu các kiến thức liên quan đến VUEJS, bắt tay vào code giao diện (Tuần 4)
* Hoàn thiện giao diện và các chức năng của bài tập (Tuần 5)
* Sửa lỗi, hoàn thiện phần còn lại của code và làm báo cáo (Tuần 6)

**Kết quả đã hoàn thành**

* Hoàn thành trang web quản lý thông tin tiêm chủng với gần như đầy đủ các chức năng, vai trò và các yêu cầu đề bài đề ra.
* Có thêm hiểu biết về các kiến thức liên quan đến bài tập.
* Trau dồi kĩ năng, kiến thức để hiểu hơn về mảng lập trình web.

**Các khó khăn trong quá trình làm bài**

* Do thời gian ngắn của kỳ hè và ảnh hưởng của dịch bệnh nên bài tập đồ án của em có thể còn nhiều thiếu sót.
* Do kiến thức của bản thân em còn nhiều hạn chế nên làm bài còn nhiều vấn đề giải quyết chưa triệt để hoặc không làm được, một số kiến thức khác phải tìm hiểu nên mất nhiều thời gian.
* Do lần đầu làm một mình một website với những công nghệ mới chưa từng dùng bao giờ nên gặp nhiều khó khăn trong quá trình làm bài.

**Khả năng phát triển, triển vọng của đồ án**

* Hoàn toàn có thể dùng kết quả của đồ án này để quản lý thông tin tiêm chủng, có thể dùng để áp dụng vào thực tế.
* Có thể mở rộng thêm nhiều tính năng, cần thiết nhất là tính năng thông báo, có thể nhắc nhở người dùng khi sắp đến lịch tiêm, lịch tiêm bị hủy hoặc các thông tin liên quan.
* Thêm tính năng người dùng sẽ có thể đăng ký lịch tiêm theo nguyện vọng, bác sĩ sẽ là người duyệt lịch tiêm xem có phù hợp hay không và hẹn lịch khám sức khỏe cụ thể.

**Kết luận :** Sau thời gian kỳ hè dưới sự hướng dẫn, định hướng của thầy Bùi Trọng Tùng, em đã hoàn thành Project II của mình với kết quả đúng như kì vọng. Website hiện giờ còn nhiều thiếu sót và em sẽ cố gắng thực hiện và hoàn thiện trong tương lai để hy vọng có thể deploy website này để đưa vào thực tế có thể sử dụng. Em xin chân thành cảm ơn !