**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TPHCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**CHUYÊN NGÀNH – KỸ THUẬT PHẦN MỀM**



Đề tài

Ứng dụng web quản lý học viên

isc quang trung



**SINH VIÊN THỰC HIỆN:**

Huỳnh Thanh Đàng

Mssv – 1512091

**GIẢNG VIÊN HƯỚNG DẪN:**

Thầy Mai Anh Tuấn

**NGƯỜI HƯỚNG DẪN TỪ TMA SOLUTIONS:**

Anh Nguyễn Đăng Khoa

Anh Cao Quang Bình

MỤC LỤC

[Phần 1: Giới thiệu tổng quan đề tài 3](#_Toc32862966)

[1. Vấn đề 3](#_Toc32862967)

[2. Giải pháp 3](#_Toc32862968)

[3. Công cụ sử dụng 3](#_Toc32862969)

[4. Kế hoạch xây dựng 3](#_Toc32862970)

[Phần 2: Cơ sở lý thuyết 3](#_Toc32862971)

[A. ASP.NET Core 3](#_Toc32862972)

[1. Giới thiệu 3](#_Toc32862973)

[1.1 ASP.NET Core là gì ? 3](#_Toc32862974)

[1.2 Tại sao sử dụng ASP.NET Core ? 3](#_Toc32862975)

[1.3 Trích nguồn 4](#_Toc32862976)

[2. Cài đặt 4](#_Toc32862977)

[2.1 Runtime 4](#_Toc32862978)

[2.2 Editor 4](#_Toc32862979)

[2.3 Tạo ứng dụng ASP.NET Core Web API 4](#_Toc32862980)

[2.4 Diễn giải một số thành phần của project 5](#_Toc32862981)

[B. Angular 8 5](#_Toc32862982)

[1. Giới thiệu 5](#_Toc32862983)

[2. Cài đặt 12](#_Toc32862984)

[2.1 Cài đặt NodeJS 12](#_Toc32862985)

[2.2 Cài đặt Angular CLI 12](#_Toc32862986)

[2.3 Tạo dự án mẫu 12](#_Toc32862987)

[2.4 Cấu trúc chính 14](#_Toc32862988)

[2.5 Vòng đời 15](#_Toc32862989)

[2.6 Các cú pháp thường dùng 15](#_Toc32862990)

[C. Microsoft SQL Server 2014 15](#_Toc32862991)

[1. Giới thiệu 16](#_Toc32862992)

[2. Lịch sử về SQL Server 16](#_Toc32862993)

[3. Các phiên bản SQL Server 16](#_Toc32862994)

[4. Lý do sử dụng SQL Server 17](#_Toc32862995)

[5. Cài đặt 17](#_Toc32862996)

[Phần 3: Áp dụng 26](#_Toc32862997)

[1. Kiến trúc dự án 26](#_Toc32862998)

[2. Danh sách chức năng 26](#_Toc32862999)

[3. ERD 27](#_Toc32863000)

# Phần 1: Giới thiệu tổng quan đề tài

1. Vấn đề
2. Giải pháp
3. Công cụ sử dụng

|  |  |
| --- | --- |
| Tên công cụ | Mục đích sư dụng |
| Microsoft Visual Studio 2019 | Xây dựng backend với ASP.NET Core 3.1 |
| Microsoft Visual Studio Code | Xây dựng frontend với Angular 8 |
| Microsoft SQL Server 2014 Enterprise | Xây dựng database |
| Power Designer | Thiết kế diagram để xây dựng database |
| StarUML |
| Draw.io | Vẽ kiến trúc tổng quan phần mềm |
| Figma | Vẽ Mockup |

1. Kế hoạch xây dựng

Tổng thời gian thực hiện: 6 tháng (từ 7/1/2020 đến 7/7/2020)

# Phần 2: Cơ sở lý thuyết

1. ASP.NET Core
2. Giới thiệu
3. ASP.NET Core là gì ?

A screenshot of a cell phone

Description automatically generated

ASP.NET Core là một framework mã nguồn mở với hiệu suất cao, đa nền tảng, dùng cho việc xây dựng các ứng dụng hiện đại, có kết nối internet và cloud-based. Với ASP.NET Core, bạn có thể:

* + - Xây dựng ứng web, web service, ứng dụng IoT, và mobile backend.
    - Sử dụng các công cụ ưa thích trên Windows, macOS, Linux.
    - Triển khai trên cloud hoặc on-premises.
    - Chạy trên .NET Core hoặc .NET Framework

1. Tại sao sử dụng ASP.NET Core ?

Hàng triệu lập trinh viên đã và đang sử dụng ASP.NET 4.x để tạo ra ứng dụng web. ASP.NET Core là phiên bản thiết kế lại của ASP.NET 4.x, với nhiều thay đổi kiến trúc giúp framework nhẹ hơn và có tính module nhiều hơn. Do đó, các lập trình viên có thể tiếp tục xây dựng ứng dụng bằng ASP.NET Core với nền tảng hiệu suất và tính tương thích tốt hơn.

ASP.NET cung cấp các lợi ích như sau:

* + - Dùng để xây dựng giao diện Web (Web UI) cũng như các API Web.
    - Tích hợp các framework phía client hiện đại và các quy trình làm việc phát triển.
    - Hệ thống cấu hình sẵn có trên đám mây.
    - Tích hợp sẵn nhúng phụ thuộc.
    - Đường ống (pipeline) yêu cầu HTTP mang tính module, hiện suất cao và nhẹ ký.
    - Có khả năng lưu trữ (host) ở IIS, Nginx, Apache, Docker, hoặc tự host ở các tiến trình riêng.
    - Tạo mới phiên bản app side-by-side với .NET Core.
    - Tạo công cụ đơn giản hóa phát triển web hiện đại.
    - Khả năng xây dựng, chạy trên Windows, macOS, và Linux.
    - Mã nguồn mở và tập trung vào cộng đồng phát triển mã nguồn.

1. Trích nguồn

<https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-3.1>

1. Cài đặt
2. Runtime
   * + Nền tảng ứng dụng .NET Core được chạy nhờ bộ Runtime được cung cấp cài trên máy tính. Microsoft hiện cung cấp bộ Runtime này miễn phí và có thể chạy trên mọi hệ điều hành vi tính.
     + Có thể tải từ đây: <https://dotnet.microsoft.com/download/dotnet-core>
     + Phiên bản được sử dụng trong đề tài: **.NET Core 3.1**
3. Editor

Người dùng có thể sử dụng Visual Studio Code để lập trình ứng dụng Web ASP.NET Core hoặc Visual Studio 2017 bản Community (miễn phí đối với người dùng cá nhân).

1. Tạo ứng dụng ASP.NET Core Web API

Mở Visual Studio 2017, chọn New Project, sau đó ASP.NET Core Web Application như hình bên dưới

A screenshot of a cell phone

Description automatically generatedHình a-1: Tạo ứng dụng ASP.NET Core

Chọn API TemplateA screenshot of a cell phone

Description automatically generated

Hình a.2: Chọn template cho ứng dụng

1. Diễn giải một số thành phần của project

A screenshot of a cell phone

Description automatically generated

Hình a.3: Solution Explorer

* + - ***Program.cs***: Tệp chạy chính, chứa các câu lệnh khởi chạy một ứng dụng DotNet Core.
    - ***Startup.cs***: Tệp chứa các lệnh giúp thiết lập các tính năng của ứng dụng Web.
    - ***Appsettings.json***: Tệp lưu trữ các chuỗi thiết lập.
    - ***Wwwroot***: Thư mục chứa các “static files”, các tệp tin trong wwwroot sẽ được phục vụ cho các yêu cầu lưu trữ hoặc đọc.
    - ***Controllers***: Thư mục chứa các tệp cs Controller, các tệp này chứa các lệnh cấu hình cho ứng dụng lắng nghe thông qua các đường dẫn

1. Angular 8
2. Giới thiệu

A close up of a sign

Description automatically generated

(Trích: <https://tedu.com.vn/lap-trinh-angular-2-can-ban/angular-80-%E2%80%93-co-gi-moi-cach-nang-cap-ra-sao-202.html>)

Ra mắt mới đây 28 tháng 5 năm 2019 với CLI workflow improvements, Dynamic imports for lazy routes ….

* **Hỗ trợ TypeScript 3.4.x**

Angular 8 hỗ trợ bản TypeScript 3.4 hoặc cao hơn. Nếu bạn muốn dùng Angular 8 cho ứng dụng của mình thì cần upgrade TypeScript lên 3.4 hoặc cao hơn.

* **IVY Rendering Engine**

Tính năng quan trọng và được mong chờ nhất của Angular 8 là IVY render engine. IVY là trình biên dịch Angular hoạt động như một cơ chế render mới. Lợi ích của IVY là tạo ra các bundle nhỏ có thể thêm dễ dàng. IVY cơ bản là một cải tiến của Angular. Sau khi chuyển sang IVY thì ứng dụng đã có vẫn làm việc như trước đây nhưng giảm kích thước bundle. Trong Angular 8, Google giới thiệu bản preview cho IVY. Mục tiêu chính của bản này là nhận phản hồi từ các Developer liên quan tới Ivy. Nó được khuyến nghị không dùng cho môi trường production.

**A close up of text on a black background

Description automatically generated**

Hình b-1: Kích thước Bundle của ứng dụng Hello World với Angular Ivy và không có Ivy. (nguồn ngconf 2019 Keynote bởi Brad Green và Igor Minar)

Tạo ngconf 2019, Brad Green, Techincal Director của Angular Team tạo Google nói Ivy sẽ có một cải tiến đáng kể liên quan đến kích thước bundle và tương thích với tính năng Differnent Loading. Chúng ta sẽ có các lợi ích sau đây khi sử dụng Ivy:

1. Nó sẽ giúp biên dịch nhanh hơn (dự kiến phát hành ở Angular 9)
2. Nhiều cải tiến kiên quan đến check type trong template và bắt lỗi trong thời gian build và giúp người dùng giảm gặp lỗi ở giai đoạn runtime.
3. Bundle size nhỏ hơn so với hiện tại
4. Nó cũng có thể tương thích ngược với các ứng dụng đã phát triển trước đây.
5. Code được tạo bởi Angular Complier sẽ dễ dàng để đọc hiểu hơn.
6. Bạn có thể debug cả template, tính năng này chắc chắn sẽ làm rất nhiều developer thích thú.

Và nếu bạn muốn sử dụng Ivy trong ứng dụng mới của mình, bạn có thể tạo mới project với option enable-ivy

|  |
| --- |
| ng **new** ivy-project --enable-ivy |

Câu lệnh trên tạo mới Angular CLI sẽ lưu cấu hình trong tsconfig.json file

|  |
| --- |
| "angularCompilerOptions": { "enableIvy": true } |

Cấu hình trên có thể thêm bằng tay vào bất cứ ứng dụng vũ nào sau khi upgrade lên Angular 8. Một khuyến nghị là nếu bạn muốn dùng Ivy trong ứng dụng, bạn có thể chạy ứng dụng trong mode debug với mode AOT mode.

|  |
| --- |
| ng serve --aot |

* **Hỗ trợ Bazel**

Trong Angular 8, Google giới thiệu một công cụ build mới là Bazel. Thực tế nó không phải là công cụ mới hoàn toàn. Google sử dụng công cụ này nội bộ từ lâu và hiện tại họ phát hành nó như là một mã nguồn mở. Nhưng một thứ cần phải làm rõ là Bazel chưa thực sự sẵn sàng trên Angular 8. Nó được giới thiệu như là một tính năng nhỏ trong Angular 8 và được mong chờ là sẽ có bản chính thức trong Angular CLI 9. Chúng ta có thể có các lợi ích sau khi sử dụng tool này:

1. Nó sẽ giúp build nhanh hơn. Nó sẽ tốn time hơn cho lần đầu build nhưng sẽ nhanh hơn nhiều nếu như build song song.
2. Dùng công cụ này chúng ta có thể build ứng dụng như là một incremental build (build chỉ những gì thay đổi) và deploy chỉ với những gì thay đổi hơn là toàn bộ ứng dụng.
3. Chúng ta có thể ẩn file Bazel đi

Chúng ta có thể thêm Bazel vào ứng dụng sử dụng lệnh:

|  |
| --- |
| ng add @angular/bazel |

Và có thể tạo ứng dụng với tùy chọn Bazel:

|  |
| --- |
| npm install -g @angular/bazel  ng **new** my-app --colection=@angular/bazel |

* **Differential Loading cho việc tối ưu hóa hiệu năng**

Một trong những tính năng rất hay trong Angular CLI 8. Bởi vì nó giúp chúng ta chỉ ra trình duyệt nào sẽ được target và CLI sẽ build ứng dụng với các bundle JS liên quan cần thiết. Trình duyệt target mặc định trong tsconfig.json file giờ là es2015. Nghĩa là khi CLI build ứng dụng, nó sẽ build ứng dụng trong một trình duyệt cũ hơn như là IE9 sau đó chúng ta cần chỉ ra nó trong file browserlist. File này tòn tại trong thư mục gốc của ứng dụng và sử dụng

|  |
| --- |
| # For IE 9-11 support, please remove 'not' **from** the last line **of** the file and adjust **as** needed  > 0.5%  last 2 versions  Firefox ESR  not dead  not IE 9-11 |

Nếu chúng ta bỏ từ khóa not ở dòng cuối đi thì build process sẽ gen ra các file cho cả 2 version.

**A screenshot of a cell phone

Description automatically generated**

Chỉ có một nhược điểm của quá trình này là sẽ mất nhiều thời gian build hơn. Các trình duyệt khác nhau có thể sẽ cần quyết định xem bundle nào sẽ được load. Cho mục đích này, một sript reference được thêm vào index.html:

|  |
| --- |
| <script src="main-es2015.js" type="module"></script> -- For ES6 support browser or latest browser  <script src="main-es5.js" nomodule></script> -- For es5 support browser or old browser |

Hình ảnh trên chúng ta có thể thấy bundle size giảm đi đáng kể trong trường hợp ES2015 từ 9% đến 20%. Cuối cùng thì bundle size nghĩa là tối ưu hiệu năng. Vì thế Angular sẽ build thêm các file bổ sung với polyfills và chúng sẽ được thêm vào trong ứng dụng

A picture containing screenshot

Description automatically generated

* **Thay đổi Lazy Loading trong Route**

Trong Angular từ khi khởi đầu cho đến nay, cơ chế router luôn hỗ trợ khái niệm lazy loading. Đến Angular 7 nó đã được hoàn thiện:

|  |
| --- |
| {  path: 'lazy',  loadChildren: () => './admin/admin.module#AdminModule'  } |

Giá trị trước dấu # chỉ ra đường dẫn đến module file mà module đó tồn tại, còn sau dấu # chỉ ra tên của class Module. Phong cách này vẫn làm việc trong Angular 8, nhưng cách để viết lazy module được thay đổi. Chuỗi giá trị của loadChidren bị deprecated. Giờ loadChildren được khai báo như sau:

|  |
| --- |
| {  path: 'lazy',  loadChildren: () => **import**('./admin/admin.module').then(m => m.AdminModule)  } |

* **Hỗ trợ Web Worker**Như chúng ta đã biết JavaScript luôn thực thi đơn thread. Nó là điều tốt khi thực thi lượng lớn dữ liệu hoặc gọi liên tiếp nhau trong phương pháp bất đồng bộ. Nhưng theo kịch bản ứng dụng thực tế, khái niệm này không giúp gì nhiều. Đó là tại sao ngày này tất cả các trình duyệt web hỗ trợ web workder process. Cơ bản là web worker process là các script được thực thi bởi trình duyệt trong một thread tách biệt. Giao tiếp với các thread đó trong trình duyệt sẽ có thể gửi tin nhắn. Nhìn chung, web worker không liên quan đến Angular nhưng điểm chính là các script này cần được xem xét trong build process của ứng dụng. Mục tiêu chính là cung cấp một bundle cho tất cả web worker. Giờ Angular 8 đã thực hiện công việc này bởi Angular CLI.

Theo đố chúng ta có thể cấu hình Angular CLI nếu chúng ta thêm web worker cho lần đầu tiên. Trong quá trình này, CLI sẽ loại bỏ các worker.ts từ tsconfig.json và thêm các cấu hình TypeScript với tên là tsconfig.worder.json để quản lý file worker.ts. CÁc thông tin này cũng được thêm vào angular.json

|  |
| --- |
| "webWorkerTsConfig": "tsconfig.worker.json" |

* **Thu thập phân tích dữ liệu sử dụng**

Với Angular 8, Angluar CLI thu thập các dữ liệu phân tích cho Angular team có thể sắp xếp ưu tiên các tính năng và các cải tiến. Khi chúng ta update ứng dụng CLI, nó sẽ hỗ trợ tùy chọn **ng analytics**. Nếu chúng ta cho phép nó chạy toàn bộ, nó sẽ thu thập dữ liệu như các lệnh được sử dụng, các cờ được sử dụng, hệ điều hành, phiên bản Node, CPU count, Ram Size, thời gian thực thi, lỗi nếu có và Angular team có thể cải tiến và chỉnh sửa trong phiên bản tiếp.

* **Loại bỏ @angular/http**

Từ Angular 8, angular ngừng hỗ trợ @angular/http. Nó đã bị deprecated (báo ngừng hỗ trợ phiên bản sau) từ Angular 4 và cung cấp một thư viện hiệu quả và bảo mật hơn gọi là @angular/common/http. Nhưng ở Angular 7 vẫn cho phép chúng ta sử dụng @angular/http trong ứng dụng. Nhưng Angular 8 thì nó đã dừng hỗ trợ. Chúng ta cần chỉnh code lên @angular/common/http

* **Thay đổi ViewChild và ContentChild**

Trong Angular 8 có một số thay đổi lớn với việc sử dụng ViewChild và ContentChild. Nó là những thứ không mong đợi của team Angular. Hiện tại Angular 8 nó bắt buộc cung cấp một cờ Boolean static để định nghĩa instance cho ViewChild và ConetntChild. Chúng ta đã có kinh nghiệm khi 1 issue thỉnh thoảng là các instance này có sẵn trong ngOnInit, nhưng đôi khi chúng ta không tìm thấy các instance này cho đến khi ngAfterContentInit hoặc ngAfterViewInit. Nó xuất hiện chính khi các phần thử được load vào DOM sau khi dữ liệu được binding. (giống \*ngIf hoặc \*ngFor) sau đó các phần thử được insert vào DOM trong ngAfterViewInit hoặc ngAfterContentInit. Điều này gây hiểu lầm và đó là vì sao chúng ta cần chỉ ra compoinent:

|  |
| --- |
| @ViewChild('info', { static: false }) \_cityViewChild: CityComponent; |

Nếu giá trị static là true, angular sẽ cố tìm kiếm phần tử tở thời điểm khởi tạo component, ví dụ ngOnInit. Nó có thể hoàn thành khi phần tử không được sử dụng bất kỳ structure directive nào. Khi static value là false, angular sẽ tìm phần tử sau khi khởi tạo view.

* **Hỗ trợ SVG Template**

Hiện tại Angular 8 hỗ trợ template với đuôi mở rộng là SVG. Nghĩa là chúng ta có thể dùng file SVG để làm template như HTML mà không cần cấu hình gì khác cả. Nhưng câu hỏi là tại sao phải sử dụng .svg như là template thay vì sử dụng image trong HTML file. Lý do là khi sử dụng SVG template, chúng ta có thể sử dụng như là một directive và kết quả chúng ta có thể bind nó như HTML template. Với các tiếp cận này, chúng ta có thể tạo động các đồ hoạt tương tác trong ứng dụng.

|  |
| --- |
| @Component({  selector: "app-icon",  templateUrl: "./icon.component.svg",  styleUrls: ["./icon.component.css"]  })  export class DashboardComponent {...} |

* **Hỗ trợ PNPM**

Trong Angular 8, Angular CLI cũng hỗ trợ một package manager mới PNPM bao gồm NPM và Yarn. Trong câu lệnh ng add giờ cung cấp thêm 1 cái cờ nữa là –registry để thêm các packages từ bất cứ NPM private nào. Câu lệnh này đã có sẵn trên Angular CLI với ng update command.

* **Hỗ trợ Codelyzer 5.0**

Angular 8 hỗ trợ cấu hình TSLint bởi Codelyzer 5.0. Phiên bản này của Codelyzer, một vài quy tắc được đổi tên. Nhưng khi chúng ta upgrade Angular CLI, nó sẽ update file cấu hình TSLint. CLI update sẽ loại bỏ các lệnh import es6 từ file polyfills.ts. Bởi vì nó tự động được thêm vào nếu được yêu cầu bởi CLI.

* **Hỗ trợ thư viện mới New Builders/ Architect API**

Phiên bản mới của Angular CLI cho phép chúng ta sử dụng phiên bản mới của Builders được biết đến là Architect API. Angular sử dụng Builders API để thực hiện các hoạt động như server, build, test, lint và e2e. Chúng ta có thể sử dụng builders trong file angular.json

|  |
| --- |
| "projects": {  "app-name": {  "architect": {  "build": {  "builder": "@angular-devkit/build-angular:browser",  },  "serve": {  "builder": "@angular-devkit/build-angular:dev-server",  },  "test": "builder": "@angular-devkit/build-angular:karma",  },  "lint": {  "builder": "@angular-devkit/build-angular:tslint",  },  "e2e": {  "builder": "@angular-devkit/build-angular:protractor",  }  }  }  } |

* **Làm sao để nâng cấp lên Angular 8**

Đển nân cấp lên Angular 8 từ bất cứ ứng dụng có sẵn nào chạy Angular 7, bạn cần thực hiện các bước sau đây:

1. Angular 8 sử dụng TypeScript 3.4, cần cài TypeScript 3.4
2. Bạn cần sử dụng Node 12 hoặc cao hơn
3. Hãy thực thi lệnh nâng cấp Angular CLI à ng update @angular/cli @angular/core.

Cách nữa là bạn cần kiểm tra quá trình upgrade trong link: <https://update.angular.io/>

1. Cài đặt
2. Cài đặt NodeJS
   * + Với hệ điều hành Windows:

Có thể tải từ đây: <https://nodejs.org/en/> (dùng phiên bản LTS – Long Time Support, với các hỗ trợ đầy đủ nhất)

* + - Kiểm tra

Mở cmd (Windows + R) lên và kiểm tra như bên dưới

A screen shot of a computer

Description automatically generated

1. Cài đặt Angular CLI

Cài đặt Angular CLI vào máy tính Windows như sau:

* Mở cmd (Windows + R)
* Gõ: *npm install --global @angular/cli*

A screenshot of a cell phone

Description automatically generated

1. Tạo dự án mẫu

Mở giao diện dòng lệnh tại thư mục cần tạo dự án.

Sau đó, gõ lệnh: *ng new myapp*

Trong đó, *myapp* là tên của dự án người dùng muốn tạo.

A screen shot of a computer

Description automatically generated

Hình c-1: Quy trình tạo ứng dụng Angular đầu tiên

Lúc này, kiểm tra thư mục sẽ thấy các tập tin dự án đã được tạo

A screenshot of a cell phone

Description automatically generated

Hinh c-2: Các tập tin và thư mục của ứng dụng Angular

1. Cấu trúc chính

Khi khởi tạo dự án Angular, người dùng có thể thấy các thư mục sau:

* **node\_modules**: Chứa thư viện cần thiết cho ứng dụng web Angular
* **src**: Thư mục chứa nội dung dự án
* **src/app**: Thư mục chứa các thành phần của dự án, gồm css, html và ts
* **src/assets**: Thư mục chứa tập tin tài nguyên như ảnh, css, js, nhạc, video.
* **src/environment**: Thư mục chứa tập tin thiết lập môi trường cho người lập trình.

A screenshot of a cell phone

Description automatically generated

Hình 3. 10. Mô hình hoạt động của Angular

1. Vòng đời

Mỗi thành phần, trang của Angular đều có vòng đời, mỗi sự kiện của vòng đời sẽ được gọi trong Component khi lập trình viên bổ sung và ghi đè các sự kiện tương ứng đó:

* **Constructor**: Được gọi khi chạy vào script của Angular.
* **ngOnChanges**: Kích hoạt khi các biến được thay đổi giá trị.
* **ngOnInit**: Gọi khi bắt đầu ràng buộc các thuộc tính, thiết đặt các giá trị.
* **ngAfterContentInit**: Gọi khi nội dung bên ngoài đã được khởi tạo xong
* **ngAfterContentChecked**: Gọi mỗi khi nội dung bên ngoài bị thay đổi
* **ngAfterViewInit**: Được kích hoạt khi thẻ đã hoàn tất render lần đầu.
* **ngAfterViewChanges**: Chạy khi View render hoàn tất các thay đổi.
* **ngOnDestroy**: Khi thành phần này bị xóa khỏi trang hoặc rời khỏi trang.

1. Các cú pháp thường dùng

|  |  |
| --- | --- |
| Cú pháp | Ý nghĩa |
| ng new <tên project> | Tạo project mới |
| npm install | Cài đặt các thư viện cần thiết cho project |
| ng serve -o | Chạy project (tự động mở trình duyệt) |
| ng build --prod | Đóng gói ứng dụng cho bản phát hành. Các gói sẽ được tối ưu và không hiển thị lỗi khi triển khai |
| ng generate component <tên component>  *hoặc* ng g c <tên component> | Tạo component |
| ng generate service <tên service>  *hoặc* ng g s <tên service> | Tạo service |
| ng generate module <tên module>  *hoặc* ng g m <tên module> | Tạo module |

Bảng c-1: Cú pháp dòng lệnh hay sử dụng trong Angular CLI

1. Microsoft SQL Server 2014
2. Giới thiệu

A close up of a logo

Description automatically generated

(Trích nguồn: <https://www.guru99.com/sql-server-introduction.html>)

SQL Server là một hệ thống quản lý cơ sở dữ liệu quan hệ (RDMS) được phát triển bởi Microsoft. Nó chủ yếu được thiết kế và phát triển để cạnh tranh với MySQL và Oracle.

SQL Server hỗ trợ ANSI SQL – Structure Query Language – là tiêu chuẩn của ngôn ngữ SQL. Tuy nhiên, SQL Server đi kèm với việc triển khai ngôn ngữ theo một cách riêng, T-SQL (Transact-SQL)

T-SQL là ngôn ngữ thuộc sở hữu của Microsoft được biết như là Transact-SQL. Nó cung cấp các khả năng khai báo biến, xử lý ngoại lệ, thủ tục lưu trữ, …

SQL Server Management Studio (SSMS) là công cụ giao diện chính cho SQL Server, hỗ trợ cả môi trường 32bits và 64bits.

1. Lịch sử về SQL Server

Microsoft và Sybase đã phát hành bản 1.0 vào năm 1989. However, việc hợp tác kết thúc nào năm 1990. Microsoft đã duy trì việc sở hữu tên SQL Server. Từ năm 1990, các phiên bản tiếp theo của SQL Server đã được phát hành bao gồm SQL Server 2000, 2005, 2008, 2012, 2014, 2016 và 2017

1. Các phiên bản SQL Server

* SQL Server Enterprise: được sử dụng trong các doanh nghiệp lớn, cung cấp các khả năng bảo mật cao, phân tích dữ liệu, học máy,…
* SQL Server Standard: phù hợp cho các ứng dụng và kho dữ liệu Mid-Tier, bao gồm các báo cáo và phân tích cơ bản
* SQL Server WEB: được thiết kế cho các máy chủ với tùy chọn chi phí sở hữu thấp. Nó cung cấp khả năng mở rộng, khả năng chi trả và khả năng quản lý các thuộc tính WEB có quy mô từ nhỏ đến lớn.
* SQL Server Developer: gần giống như bản Enterprise dành cho môi trường phi sản xuất. Nó chủ yếu được sử dụng để build, test và demo…
* SQL Server Express: được dùng cho các ứng dụng có quy mô nhỏ, và hoàn toàn miễn phí

1. Lý do sử dụng SQL Server

* SQL là ngôn ngữ phi thủ tục, không yêu cầu cách thức truy cập cơ sở dữ liệu như thế nào. Tất cả các thông báo của SQL rất dễ dàng sử dụng và ít mắc lỗi.
* Bảo mật và khả năng mở rộng.
* Có bộ nhớ trong tích hợp, hiệu suất trung bình tăng gấp 10 lần đối với các ứng dụng mới và ứng dụng hiện có của hệ thống SQL Server.
* Dễ dàng xử lý, truy xuất tới các Cơ sở dữ liệu lớn nhỏ khác nhau.
* Phân tích nhanh hơn và chính xác từ các công cụ quen thuộc.
* Tính toàn diện và đồng nhất.

SQL cung cấp các tập lệnh phong phú cho các công việc hỏi đáp dữ liệu như:

* Chèn, xóa và cập nhật các hàng trong 1 quan hệ
* Tạo, thêm, xóa và sửa đổi các đối tượng trong của cơ sở dữ liệu.
* Điều khiển việc truy cấp tới cơ sở dữ liệu và các đối tượng của cơ sở dữ liệu để đảm bảo tính bảo mật, tính nhất quán và sự ràng buộc của cơ sở dữ liệu.

1. Cài đặt

(Trích nguồn: <https://freetuts.net/huong-dan-tai-va-cai-dat-microsoft-sql-server-2014-toan-tap-1097.html>)

**Bước 1:** Sau khi các bạn tải về phần mềm **Microsoft SQL Server 2014** chúng tôi đã cung cấp bên trên, hãy tiến hành giải nén.

A screenshot of a cell phone

Description automatically generated

**Bước 2:** Chạy tệp tin **Setup.exe** có chứa trong bộ cài mà các bạn vừa giải nén bằng quyền Admin **(Run as administrator)**.

A screenshot of a cell phone

Description automatically generated

Chờ chương trình thu thập thông tin về phần cứng và độ tương thích với hệ điều hành để xử lý cài đặt

**Bước 3:**Tại giao diện cài đặt, các bạn chọn thẻ **Installation**và chọn tiếp **New SQL Server stand-alone installation or add features to an existing installation** để tiến hành cài đặt**Microsoft SQL Server 2014**.

A screenshot of a social media post

Description automatically generated

**Bước 4:** Tại giao diện cài đặt, các bạn hãy tích dấu vào các ô kiểm như hình để đồng ý với các điều khoản của phần mềm, sau đó nhấn **Next**để tiếp tục quá trình cài đặt.

A screenshot of a cell phone

Description automatically generated

**Bước 5:**Tiếp tục tích dấu vào ô kiểm **Use Microsoft Update to check for updates (recommended)** để bật cập nhật phần mềm. sau đó chọn **Next**để tiếp tục quá trình cài đặt.

A screenshot of a cell phone

Description automatically generated

**Bước 6:**Chờ quá trình cập nhật phần mềm.

A screenshot of a cell phone

Description automatically generated

**Bước 7:** Sau khi phần mềm hoàn tất cập nhật các bạn hãy chọn các **features**cài thêm vào trong hệ thống, các bạn chọn vào nút **Select All**và nhấn **Next**để tiếp tục quá trình cài đặt.

A screenshot of a social media post

Description automatically generated

**Bước 8:**Điền tên người sử dụng vào ô **Name instance**, các bạn lưu ý **chỉ điền tên viết liền không chứa dấu câu và các ký tự đặc biệt**. Sau khi điền xong các bạn chọn vào mục **Instance ID**, phần mềm sẽ tự động điền tên mà bạn vừa nhập, sau đó chọn **Next**để tiếp tục quá trình cài đặt.

A screenshot of a cell phone

Description automatically generated

**Bước 9:** Tại phần **Server Configuation** này các bạn để trạng thái mặc định và nhấn **Next**để tiếp tục quá trình cài đặt.

A screenshot of a cell phone

Description automatically generated

**Bước 10:** Tại đây các bạn có 2 lược chọn:

**1. Windows authentication mode:** Để đăng nhập vào hệ quản trị SQL với tài khoản mặc định của máy tính bạn.

**2. Mixed Mode:** Tại chế độ này các bạn sẽ phải thiết lập mật khẩu, đây là mật khẩu cho tài khoản **sa**của **SQL Server** (tài khoản **sa**có toàn quyền quản trị hệ thống SQL Server).

Tại đây mình sẽ chọn **Mixed Mode** vì tính bảo mật cao và có thể quản trị dễ dàng các cơ sở dữ liệu. Sau khi hoàn tất thiết lập hãy nhấn **Next**để tiếp tục quá trình cài đặt.

A screenshot of a cell phone

Description automatically generated

**Bước 11:** Chọn mục **Install and configure** để cài đặt và thiết lập thủ công, sau đó chọn **Next**để quá trình cài đặt được bắt đầu.

A screenshot of a social media post

Description automatically generated

Đây là quá trình cài đặt, thời gian cài đặt khá là lâu tùy thuộc vào tốc độ xử lý nhanh hay chậm của máy tính bạn. Hãy cố gắng chờ phần mềm cài đặt hoàn tất.

A screenshot of a cell phone

Description automatically generated

**Bước 12:**Quá trình cài đặt thành công. Các bạn chọn **Close**để đóng bộ cài.

A screenshot of a social media post

Description automatically generated

Hãy thử khởi động SQL Server lên để kiểm tra nào. Các bạn nhấn vào nút Start và tìm kiếm với từ khóa **SQL Server 2014 Management Studio** để tìm phần mềm này nhé.

A screenshot of a social media post

Description automatically generated

# Phần 3: Áp dụng

1. Kiến trúc dự án

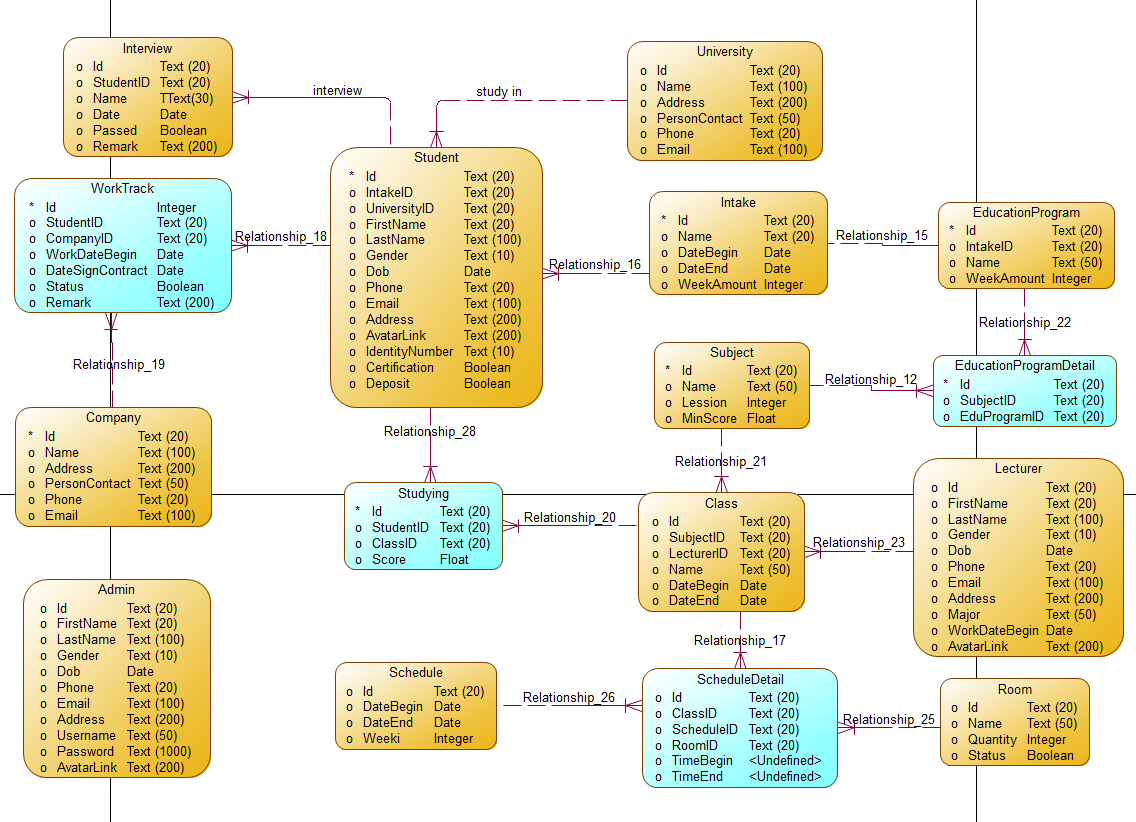
A screenshot of a map

Description automatically generated

1. Danh sách chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| Stt | Group | Admin Feature | Done (%) |
| 1 | Authentication | Login |  |
| 2 | Logout |  |
| 3 | Register |  |
| 4 | Account Management | View profile |  |
| 5 | Update profile |  |
| 6 | Update password |  |
| 7 | Notification | CRUD News |  |
| 8 | Relation | CRUD Member Companies |  |
| 9 | CRUD Linked Universities |  |
| 10 | Human | CRUD Lecturer List |  |
| 11 | CRUD Student List |  |
| 12 | Filter | Find Student |  |
| 13 | Find Lecturer |  |
| 14 | Program Structure | CRUD Intake List |  |
| 15 | CRUD Education Program |  |
| 16 | CRUD Subject List |  |
| 17 | CRUD Worktrack |  |
| 18 | Score | View Student Score List |  |
| 19 | View Transcript of each Student |  |
| 20 | Download Transcript |  |
| 21 | Update Student's Score |  |

1. Database
2. Lược đồ cơ sở dữ liệu quan hệ



1. Mô tả cơ sở dữ liệu
   * Các học viên trước khi chính thức nhập học sẽ phải trải qua phỏng vấn, có thể có nhiều đợt phỏng vấn cho học viên.
   * Mỗi học viên thì thuộc về một trường đại học.
   * Trong một khóa học thì có nhiều học viên và ở mỗi khóa học sẽ chỉ có duy nhất một chương trình học được lựa chọn bởi đa số các học viên tham gia lựa chọn trong danh sách các chương trình học.
   * Một chương trình học sẽ có nhiều môn học và ngược lại.
   * Một môn học sẽ tương ứng một lớp học được giảng dạy bởi một giảng viên tương ứng.
   * Một lớp học (hiểu nôm na gồm 1 môn học và 1 giảng viên tương ứng) sẽ được diễn ra tại một phòng học, và lớp học này sẽ có thời khóa biểu cho mỗi tuần.
   * Học viên sau khi hoàn thành khóa học sẽ được tự do lựa chọn tham gia phỏng vấn, thực tập tại các công ty khác nhau trong các công ty thành viên của QTSC.
2. Usecase

Phần 4: Tài liệu tham khảo