|  |  |
| --- | --- |
|  | BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HỒ CHÍ MINH** |

**ĐỒ ÁN TỐT NGHIỆP**

**WEBSITE DIỄN ĐÀN DAISY FORUM**

Ngành: **CÔNG NGHỆ THÔNG TIN**

Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn : | (1) Thầy Trần Đăng Khoa |
|  | (2) Th.s Hà Huy Cường |
| Sinh viên thực hiện : | Nguyễn Phú Đức |
| MSSV: 1911066051 | Lớp: 19DTHE4 |

*TP. Hồ Chí Minh, 2023*

# LỜI CẢM ƠN

Lời cảm ơn đến các thầy cô trong khoa đã giúp đỡ, quan tâm và theo dõi quá trình học tập và nghiên cứu của em trong suốt thời gian qua. Em rất biết ơn sự quan tâm và động viên từ các thầy cô đã giúp em vượt qua những thử thách trong quá trình học tập và nghiên cứu.

Đặc biệt, em muốn gửi lời cảm ơn chân thành đến 2 thầy giảng viên hướng dẫn của em trong quá trình thực hiện đồ án tốt nghiệp. Cả hai thầy đã dành thời gian quý báu để hướng dẫn em trong từng bước của đồ án, tận tình giải đáp các thắc mắc và đưa ra những gợi ý quý giá về phương pháp nghiên cụ và giải quyết vấn đề.

Nhờ sự hỗ trợ và tư vấn của 2 thầy giảng viên hướng dẫn, em đã có được những kiến thức, kỹ năng và kinh nghiệm quan trọng để hoàn thành đồ án tốt nghiệp. Em cảm thấy may mắn khi được học tập và làm việc dưới sự hướng dẫn của 2 thầy giảng viên tận tâm và giàu kinh nghiệm như vậy.

Em xin gửi lời cảm ơn sâu sắc đến 2 thầy giảng viên hướng dẫn của em, vì đã đóng góp không nhỏ vào sự thành công của đồ án tốt nghiệp của em. Em sẽ luôn mang trong mình những giá trị mà 2 thầy giảng viên đã truyền đạt, và sẽ cố gắng áp dụng kiến thức và kinh nghiệm đó để phát triển sự nghiệptrong tương lai.

# PHỤ LỤC

[LỜI CẢM ƠN 2](#_Toc135674901)

[PHỤ LỤC 3](#_Toc135674902)

[CHƯƠNG 1: GIỚI THIỆU 3](#_Toc135674903)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 3](#_Toc135674904)

[CHƯƠNG 3: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ 15](#_Toc135674905)

[CHƯƠNG 4: THỰC HIỆN VÀ KIỂM THỬ 15](#_Toc135674906)

[CHƯƠNG 5: KẾT QUẢ VÀ ĐÁNH GIÁ 15](#_Toc135674907)

[CHƯƠNG 6: TÍNH MỚI VÀ ĐÓNG GÓP CỦA ĐỒ ÁN 15](#_Toc135674908)

[CHƯƠNG 7: HƯỚNG PHÁT TRIỂN 15](#_Toc135674909)

[CHƯƠNG 8: KẾT LUẬN 15](#_Toc135674910)

[CHƯƠNG 9: TÀI LIỆU THAM KHẢO 15](#_Toc135674911)

# CHƯƠNG 1: GIỚI THIỆU

Phần này trình bày lý do và mục đích của đồ án, vấn đề được giải quyết, phạm vi và đối tượng nghiên cứu.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

1. **MÔ HÌNH PHÁT TRIỂN**
   1. **Alige**

Agile là một phương pháp phát triển phần mềm linh hoạt, là một hướng tiếp cận cụ thể cho việc quản lý dự án phần mềm. Nó gồm một quá trình làm việc tương tác và tích hợp để có thể đưa sản phẩm đến tay người dùng càng nhanh càng tốt.

Những phương pháp phát triển phần mềm theo cách truyền thống ngày càng bộc lộ nhiều nhược điểm và tỷ lệ các dự án thất bại cao trong thời kỳ bùng phát của ngành công nghệ. Nhận ra vấn đề đó, một số cá nhân và công ty riêng lẻ đã đưa ra các phương pháp phát triển phần mềm hiện đại hơn và khác nhau để thích ứng với tình hình mới.

Những phương thức phát triển phần mềm này giúp phần nào giải quyết được một số vấn đề nhưng lại phát sinh vấn đề khác về sự cộng tác, kỹ thuật, công cụ, hướng phát triển, chia sẻ ….

Vào năm 2001, bản tuyên ngôn Agile đã được thống nhất và ra đời bởi một nhóm người có uy tính trong phát triển phần mềm:

* **Individuals and interactions over processes and tools:** Cá nhân và sự tương tác hơn là quy trình và công cụ
* **Working software over comprehensive documentation:** Phần mềm chạy tốt hơn là tài liệu đầy đủ
* **Customer collaboration over contract negotiation:** Cộng tác với khách hàng hơn là đàm phán hợp đồng
* **Responding to change over following a plan:** Phản hồi với sự thay đổi hơn là bám theo kế hoạch
  1. **Scrum**

**Scrum** là một “bộ khung làm việc” cơ bản để tiếp cận những công việc phức tạp. Dựa trên bộ khung này, nhóm làm việc có thể áp dụng những quy trình, kỹ thuật khác nhau cho công việc của mình… Nó là một thành viên của họ Alige.

Nó giúp loại bỏ những công đoạn phức tạp và chỉ tập trung vào những công đoạn cần thiết đáp ứng được nhu cầu của khác hàng đưa ra. Ba yếu tố nòng cốt tạo thành một mô hình quản lý tiến trình thực nghiệm gồm: **sự minh bạch** (transparency), **thanh tra**(inspection) và **thích nghi** (adaptation).

**Ba giá trị cốt lõi của Scrum**

* ***Minh bạch***

Muốn áp dụng thành công Scrum, các thông tin liên quan đến quá trình phải mình bạch và thông suốt. Các thông tin có thể là tầm nhìn của sản phẩm, yêu cầu của khách hàng, tiến độ công việc, các rào cản khác…

Từ đó mọi thành viên ở vai trò khác nhau có đầy đủ thông tin cần có để tiến hành quyết định trong việc nâng cao hiệu quả công việc.

* ***Thanh tra***

Phải thường xuyên thanh tra các hoạt động trong Scrum và tiến độ đến đích để phát hiện các bất thường không theo ý muốn. Tần suất thanh tra không nên quá dày để khỏi ảnh hưởng đến công việc. Công tác thanh tra khi được thực hiện bởi người có kĩ năng tại các điểm quan trọng của công việc sẽ giúp cải tiến liên tục trong Scrum.

* ***Thích nghi***

Scrum mang lợi thế là tính linh hoạt rất cao, nhờ đó mang lại tính thích nghi cao. Dựa vào thông tin liên tục và minh bạch từ quá trình thanh tra và làm việc, Scrum có thể cho lại các thay đổi tích cực, nhờ đó mang lại thành công cho dự án.

Lợi ích mà Scrum mang lại

Tính minh bạch, kiểm tra, và thích nghi là 3 nền tảng cơ bản của Scrum. Và dưới đây là những lý do tại sao nên dùng Scrum:

1. Cải thiện chất lượng phần mềm, dễ học và dễ sử dụng.
2. Rút ngắn thời gian phát hành phần mềm, cho phép khách hàng sử dụng sản phẩm sớm hơn.
3. Nâng cao tinh thần đồng đội, tối ưu hóa hiệu quả và nỗ lực của đội phát triển.
4. Gia tăng tỷ suất hoàn vốn đầu tư (ROI)
5. Tăng mức độ hài lòng của khách hàng
6. Kiểm soát dự án tốt, cải tiến liên tục
7. Giảm thiểu rủi ro khi xây dựng sản phẩm
8. **DESIGN PATTERN**

**Mô hình MVC (Model-View-Controller)**

Mô hình MVC là một trong những mô hình phát triển phổ biến nhất hiện nay, đặc biệt trong lĩnh vực phát triển ứng dụng web. Nó giúp tách biệt logic xử lý dữ liệu, logic nghiệp vụ và giao diện người dùng, giúp dễ dàng bảo trì và mở rộng ứng dụng. Do đó, việc sử dụng mô hình MVC trong dự án của bạn là một lựa chọn tốt và được khuyến khích.

**Mô hình đang sử dụng cho phần Backend**

Mô hình đang sử dụng cho phần Backend là mô hình được cải tiến từ mô hình MVC nhằm tạo ra sự linh hoạt cho quá trình phát triển của dự án với 5 thành phần chính:

* **Controller:** chứa các lớp điều khiển, xử lý tương tác giữa người dùng và ứng dụng.
* **Data:** chứa các lớp xử lý dữ liệu, kết nối và truy xuất dữ liệu từ các nguồn khác nhau.
* **Model:** chứa các lớp mô hình, định nghĩa cấu trúc dữ liệu và cách thao tác với dữ liệu.
* **Service:** chứa các lớp cung cấp các dịch vụ hỗ trợ cho các lớp khác của ứng dụng.
* **View:** chứa các giao diện người dùng, hiển thị dữ liệu và tương tác với người dùng.

1. **CÔNG NGHỆ SỬ DỤNG**
   1. **Backend API**

* ***.NET 7.0***



**.NET Core là gì?**

.NET Core là một nền tảng phát triển đa mục đích, mã nguồn mở được duy trì bởi Microsoft và cộng đồng .NET trên GitHub. Đó là nền tảng chéo (hỗ trợ Windows, macOS và Linux) và có thể được sử dụng để xây dựng các ứng dụng thiết bị, đám mây.

**Đặc điểm của .NET Core**

* *Đa nền tảng:* Chạy trên các hệ điều hành Windows, macOS và Linux.
* Nhất quán trên các kiến ​​trúc: có thể chạy mã nguồn của bạn với cùng một hành vi trên nhiều kiến ​​trúc hệ thống, bao gồm x64, x86 và ARM.
* *Các công cụ dòng lệnh:* Bao gồm các công cụ dòng lệnh dễ sử dụng, có thể được sử dụng để phát triển cục bộ và trong các tình huống tích hợp liên tục.
* *Triển khai linh hoạt:* có thể cài đặt song song (cài đặt toàn người dùng hoặc toàn hệ thống). Có thể được sử dụng với các container Docker
* *Tương thích:* .NET Core tương thích với .NET Framework, Xamarin và Mono, thông qua .NET Standard.
* *Nguồn mở:* Nền tảng .NET Core là nguồn mở, sử dụng giấy phép MIT và Apache 2. .NET Core là một dự án .NET Foundation.
* *Được hỗ trợ bởi Microsoft:* .NET Core được Microsoft hỗ trợ, theo Hỗ trợ .NET Core.

**Thành phần của .NET Core**:

* *.NET Core runtime:* cung cấp một hệ thống kiểu, tải lắp ráp, trình thu gom rác, interop gốc và các dịch vụ cơ bản khác. Các thư viện khung .NET Core cung cấp các kiểu dữ liệu nguyên thủy, các kiểu thành phần ứng dụng và các tiện ích cơ bản.
* *ASP.NET Core runtime:* cung cấp khung để xây dựng các ứng dụng kết nối internet , điện toán đám mây hiện đại, chẳng hạn như ứng dụng web, ứng dụng IoT và phụ trợ di động.
* *.NET Core SDK và trình biên dịch ngôn ngữ (Roslyn và F #):* cho phép trải nghiệm nhà phát triển .NET Core.
* *Dotnet command:* được sử dụng để khởi chạy các ứng dụng .NET Core và các lệnh CLI. Nó chọn thời gian chạy và lưu trữ thời gian chạy, cung cấp chính sách tải lắp ráp và khởi chạy các ứng dụng và công cụ.

**Lịch sử các phiên bản và kế hoạch phát triển .NET Core:**

|  |  |  |
| --- | --- | --- |
| **Version** | **Ngày bắt đầu** | **Ngày kết thúc** |
| .NET 7 | 8 thg 11, 2022 |  |
| .NET 6.0 (LTS) | 9 thg 11, 2021 | 12 thg 11, 2024 |
| .NET 5.0 | 10 thg 11, 2020 | 10 thg 5, 2022 |
| .NET Core 3.1 | 3 thg 12, 2019 | 13 thg 12, 2022 |
| .NET Core 3.0 | 23 thg 9, 2019 | 3 thg 3, 2020 |
| .NET Core 2.2 | 4 thg 12, 2018 | 23 thg 12, 2019 |
| .NET Core 2.1 | 30 thg 5, 2018 | 21 thg 8, 2021 |
| .NET Core 2.0 | 14 thg 8, 2017 | 1 thg 10, 2018 |
| .NET Core 1.1 (LTS) | 16 thg 11, 2016 | 27 thg 6, 2019 |
| .NET Core 1.0 | 27 thg 6 , 2016 | 27 thg 6, 2019 |

* ***Identity Server 4***



**ASP.NET Core Identity là gì?**

ASP.NET Core Identity là một thư viện quản lý việc chứng thực và cấp quyền được dùng cho mọi loại project từ MVC, WebForms cho đến WebAPI. Nó đáp ứng các nhu cầu chính như sau:

* Dễ dàng tùy chỉnh profile của user
* Lưu trữ thông tin user trong cơ sở dữ liệu sử dụng Entity Framework Code First.
* Hỗ trợ unit test
* Giới hạn quyền truy cập theo quyền
* Cung cấp cơ chế làm việc với claim

**Tại sao lại cần IdentityServer4?**

ASP.NET Identity có thể nhận một token bảo mật khi đăng nhập từ bên thứ 3 như Facebook, Google, Microsoft và Twitter. Nhưng nếu bạn muốn tự sinh ra mã token bảo mật cho các ứng dụng nội bộ của bạn giống như họ thì bạn cần phải có một thư viện của bên thứ 3 giống như IdentityServer4, OpenIddict.

Nếu làm một nghiên cứu nho nhỏ thì các bạn có thể thấy là IdentityServer4 rất phổ biến và hay được dùng để làm việc này.

**IdentityServer4 là gì?**

IdentityServer4 là một framework hỗ trợ OpenID Connect và OAuth 2.0 trên ASP.NET Core. Nó cũng là một gói thư viện trên Nuget được dùng trong ASP.NET Core như một middleware cho phép đăng nhập/đăng xuất, cấp token, chứng thực và các giáo thức chuẩn khác.

Để có cái nhìn tổng quan về kiến trúc của nó, các bạn có thể tham khảo thuật ngữ ở trang chủ của IdentityServer4:

* *User:* là con người, là bạn hoặc tôi đang sử dụng một client.
* *Client:* là một phần mềm ứng dụng kiểu như trình duyệt web, mobile app hay bất cứ cái gì đang cần gọi một API resource.
* *Resources:* là các API bạn cần bảo vệ bởi IdentityServer4
* *Access Token:* nó là token được sử dụng để truy cập vào API Resource
* *Refresh Token:* mỗi một token đều có một thời gian hết hạn. Refresh token là việc lấy lại token mới mà không cần tương tác của người dùng. Client nên cho phép làm điều này bằng cách setup AllowOfflineAccess giá trị là true ở phần cài đặt client trong IdentityServer4.
* *Grant Type:* nó là loại tương tác giữa client và IdentityServer. Dựa trên client của bạn, có thể chọn loại grant type phù hợp.
* ***SQL Server***

******

**SQL Server là gì?**

SQL Server là gì? SQL Server hay Microsoft SQL Server là phần mềm ứng dụng cho hệ thống quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) được phát triển bởi Microsoft vào năm 1988. Nó được sử dụng để tạo, duy trì, quản lý và triển khai hệ thống RDBMS.

Phần mềm SQL Server được sử dụng khá rộng rãi vì nó được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn lên đến Tera – Byte cùng lúc phục vụ cho hàng ngàn user. Bên cạnh đó, ứng dụng này cung cấp đa dạng kiểu lập trình SQL từ ANSI SQL (SQL truyền thống) đến SQL và cả T-SQL (Transaction-SQL) được sử dụng cho cơ sở dữ liệu quan hệ nâng cao.

**Các phiên bản SQL Server**

* ***SQL Server 2012*** cung cấp thêm những tính năng mới như chỉ mục cột lưu trữ, có thể được sử dụng để lưu trữ dữ liệu theo định dạng cột cho các ứng dụng. Mặt khác, phiên bản này có tính khả dụng cao và được trang bị công nghệ khắc phục sau thảm họa.
* ***SQL Server 2014*** đã thêm OLTP trong bộ nhớ, từ đó cho phép người dùng chạy các ứng dụng xử lý giao dịch trực tuyến. Một tính năng mới khác trong SQL 2014 đó là phần mở rộng vùng đệm bằng cách tích hợp bộ nhớ đệm và ổ đĩa – đây là cách thiết kế để tăng thông lượng I/O thông qua việc giảm dữ liệu từ các đĩa cứng thông thường.
* ***SQL Server 2017*** hỗ trợ chạy trên Linux, điều này làm SQL Server chuyển từ nền tảng cơ sở dữ liệu sang một hệ điều hành mã nguồn mở thường thấy trong các doanh nghiệp. Thêm vào đó, ở phiên bản này còn hỗ trợ ngôn ngữ lập trình Python, một ngôn ngữ mã nguồn mở được sử dụng rộng rãi trong các ứng dụng phân tích.
* ***SQL Server 2019*** cho phép người dùng kết hợp các vùng chứa SQL Server, HDFS và Spark với nhau bằng cách sử dụng tính năng Big Data Cluster mới. Thêm vào đó, một tính năng mới khác là khả năng phục hồi dữ liệu được tăng tốc nhanh hơn.
  1. **Fontend**
     1. **Admin**
* ***Angular 15***



**Angular là gì?**

Angular là một framework JavaScript và được viết bằng TypeScript. Google tạo ra framework này với mục đích viết nên giao diện web (Front-end) đúng chuẩn “ít nỗ lực”. Không chỉ đem lại những lợi ích như một framework, mà Angular còn giữ nguyên cấu trúc như ngôn ngữ lập trình tiêu chuẩn. Điều đó giúp các nhà phát triển vừa dễ mở rộng dự án cũng như việc bảo trì.

**Ưu điểm của Angular**

* *Custom components:* Angular cho phép người dùng xây dựng các component (thành phần) của riêng họ, có thể đóng gói chức năng cùng với logic cho các component để tạo thành các phần có thể tái sử dụng. Các component này cũng hoạt động tốt với các component khác của web.
* *Data binding:* Data Binding là kĩ thuật dùng để tạo gắn kết giữa phần giao diện (UI) và dữ liệu thông qua phần business logic ( tạm dịch – logic xử lý việc trao đổi thông tin dữ liệu với giao diện người dùng). Nhờ Data Binding, UI có thể tự động cập nhật lại để hiển thị các thay đổi trong dữ liệu.
* *Dependency injection:* Angular cho phép người dùng viết các câu lệnh và đưa chúng vào bất cứ nơi nào họ cần. Điều này cải thiện khả năng kiểm tra và khả năng tái sử dụng của các câu lệnh giống nhau.
* *Testing:* Kiểm tra là công cụ hạng nhất của Angular vì ngay từ bản Angular 1 thì tính năng này đã được áp dụng. Bạn sẽ có thể kiểm tra mọi phần trong ứng dụng của mình điều này là điểm cộng nếu so với JavaScripts.
* *Comprehensive ( Tính toàn diện ):* Angular là framework được Google chống lưng nên đây là framework có “ bảo hiểm” và cung cấp các giải pháp tiện ích toàn diện cho giao tiếp máy chủ.
* *Browser Compatibility:* Angular là đa nền tảng và tương thích với nhiều trình duyệt. Một ứng dụng Angular thường có thể chạy trên tất cả các trình duyệt (Ví dụ: Chrome, Firefox) và hệ điều hành, chẳng hạn như Windows, macOS và Linux.

**Nhược điểm của Angular**

* *Steep learning curve ( Đường cong học tập):* Các thành phần cơ bản của Angular mà tất cả người dùng nên biết bao gồm chỉ thị (directives), mô-đun, trình trang trí (decorators), thành phần (components), dịch vụ (services), dependency injection, pipes, và mẫu (templates)...Các chủ đề nâng cao hơn bao gồm biên dịch AoT và Rx.js… Đối với người mới bắt đầu, Angular 4 có thể khó học vì nó là framework hoàn chỉnh.
* *Limited SEO options:* Angular cung cấp các tùy chọn SEO hạn chế và khả năng tiếp cận kém với trình thu thập thông tin của công cụ tìm kiếm.
* *Verbose/ Complex ( Rườm rà ):* Một vấn đề phổ biến trong cộng đồng Angular là sự rườm rà của framework này. Nó cũng khá phức tạp so với các công cụ front-end khác.
  + 1. **Portal**
* ***Razor View***



**View Engine là gì?**

View Engine có trách nhiệm xuất ra HTML cuối cùng khi Controller gọi Action Method. Controller action method trả về các kiểu khác nhau của response. Nó gọi là Action Result. ViewResult là một ActionResult nó trả ra HTML Response. ViewResult được cung cấp bởi View Engine. Nó sẽ sản sinh ra HTML.

**Razor View Engine là gì?**

Razor View Engine là View Engine mặc định của ASP.NET Core. Nó lấy mã Razor trong file View và chuyển sang HTML response.

**Razor Markup**

Controller trong MVC gọi View bằng cách gán dữ liệu để tạo giao diện. View phải có khả năng xử lý dữ liệu và tạo response. Điều này được xử lý bằng cách dùng Razor, nó cho chúng ta sử dụng C# code trong file HTML. Razor View Engine xử lý các lệnh này và tạo ra HTML.

Các file chứa Razor có đuôi .cshtml. Cú pháp Razor thường ngắn hơn và đơn giản hơn cũng dễ học như C# hoặc VB. Visual Studio IntelliSense cũng hỗ trợ cú pháp Razor.

* 1. **Git**

***Github***



**Github là gì?**

Githublà một dịch vụ nổi tiếng cung cấp kho lưu trữ mã nguồn [Git](https://topdev.vn/blog/git-la-gi/) cho các dự án phần mềm. Github có đầy đủ những tính năng của Git, ngoài ra nó còn bổ sung những tính năng về social để các developer tương tác với nhau.

**Vài thông tin về GIT:**

* Là công cụ giúp quản lý source code tổ chức theo dạng dữ liệu phân tán.
* Giúp đồng bộ source code của team lên 1 server.
* Hỗ trợ các thao tác kiểm tra source code trong quá trình làm việc (diff, check modifications, show history, merge source, …)

GitHub có 2 phiên bản: miễn phí và trả phí. Với phiên bản có phí thường được các doanh nghiệp sử dụng để tăng khả năng quản lý team cũng như phân quyền bảo mật dự án.  
Còn lại thì phần lớn chúng ta đều sử dụng Github với tài khoản miễn phí để lưu trữ source code.

Github cung cấp các tính năng social networking như feeds, followers, và network graph để các developer học hỏi kinh nghiệm của nhau thông qua lịch sử commit.

Nếu một comment để mô tả và giải thích một đoạn code. Thì với Github, commit message chính là phần mô tả hành động mà bạn thực hiện trên source code.

Github trở thành một yếu tố có sức ảnh hưởng lớn trong cộng động nguồn mở. Cùng với Linkedin, Github được coi là một sự thay thế cho CV của bạn. Các nhà tuyển dụng cũng rất hay tham khảo Github profile để hiểu về năng lực coding của ứng viên.

**Tính năng của Github**

GitHub được coi là một mạng xã hội dành cho lập trình viên lớn nhất và dễ dùng nhất với các tính năng cốt lõi như:

* Wiki, issue, thống kê, đổi tên project, project được đặt vào namespace là user.
* Watch project: theo dõi hoạt động của project của người khác. Xem quá trình người ta phát triển phầm mềm thế nào, project phát triển ra sao.
* Follow user: theo dõi hoạt động của người khác.

Có 2 cách tiếp cận GitHub: Tạo project của riêng mình Contribute cho project có sẵn: fork project có sẵn của người khác, sửa đổi, sau đó đề nghị họ cập nhật sửa đổi của mình (tạo pull request).

**Một vài khái niệm của Git bạn cần nắm:**

* **git**: là prefix của các lệnh được sử dụng dưới CLI
* **branch**: được hiểu như là nhánh, thể hiện sự phân chia các version khi 2 version đó có sự sai khác nhất định và 2 version đều có sự khác nhau.
* **commit**: là một điểm trên cây công việc (Work Tree ) hay gọi là cây phát triển công việc
* **clone**: được gọi là nhân bản, hay thực hiện nhân bản. Sử dụng để clone các project, repository trên các hệ thống chạy trên cơ sở là git, ví dụ như: bitbucket, github, gitlab, cor(1 sản phẩm mã nguồn mở cho phép người dùng tự tạo git server cho riêng mình trên vps, server),… Việc clone này sẽ sao chép repository tại commit mình mong muốn, dùng để tiếp tục phát triển. Thao tác này sẽ tải toàn bộ mã nguồn, dữ liệu về máy tính của bạn.
* **folk**: Folk là thao tác thực hiện sao chép repository của chủ sở hữu khác về git account của mình. sử dụng và đối xử như 1 repository do mình tạo ra.
* **repository**: Kho quản lý dữ liệu, là nơi lưu trữ các dữ liệu, mã nguồn của project.
* **tag**: sử dụng để đánh dấu một commit khi bạn có quá nhiều commit tới mức không thể kiểm soát được.
* **remote**: sử dụng để điều khiển các nhánh từ một repository trên git server, đối xử với các nhánh trên remote tương tự như đối xử với các nhánh trên local
* **diff:** So sánh sự sai khác giữa phiên bản hiện tại với phiên bản muốn so sánh, nó sẽ thể hiện các sự khác nhau
* **.gitignore:** file mặc định của git sử dụng để loại bỏ (ignore) các thư mục, file mà mình không muốn push lên git server

# CHƯƠNG 3: PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ

Phần này trình bày quá trình phân tích yêu cầu và thiết kế hệ thống, bao gồm các yêu cầu, sơ đồ kiến trúc, thiết kế cơ sở dữ liệu, hướng tiếp cận và phương pháp giải quyết vấn đề.

# CHƯƠNG 4: THỰC HIỆN VÀ KIỂM THỬ

Phần này trình bày quá trình thực hiện và kiểm thử hệ thống, bao gồm các công cụ, kỹ thuật và phương pháp sử dụng để triển khai hệ thống.

# CHƯƠNG 5: KẾT QUẢ VÀ ĐÁNH GIÁ

Phần này trình bày kết quả đạt được từ đồ án và đánh giá hiệu quả của hệ thống.

# CHƯƠNG 6: TÍNH MỚI VÀ ĐÓNG GÓP CỦA ĐỒ ÁN

Phần này trình bày tính mới và đóng góp của đồ án đối với lĩnh vực nghiên cứu.

# CHƯƠNG 7: HƯỚNG PHÁT TRIỂN

Phần này trình bày tính mới và đóng góp của đồ án đối với lĩnh vực nghiên cứu.

# CHƯƠNG 8: KẾT LUẬN

Phần này trình bày những kết luận chính của đồ án, bao gồm đánh giá tổng thể và những hạn chế của đồ án.

# CHƯƠNG 9: TÀI LIỆU THAM KHẢO

Phần này liệt kê các tài liệu tham khảo được sử dụng trong quá trình nghiên cứu và viết báo cáo.

<https://topdev.vn/blog/agile-la-gi-scrum-la-gi/>

<https://codegym.vn/blog/2020/06/29/net-core-la-gi-tong-quan-ve-net-core/>

<http://docs.identityserver.io/en/latest/intro/terminology.html>

<https://fsoft-academy.edu.vn/tin-tuc-su-kien/angular-la-gi-uu-va-nhuoc-diem/>

<https://tedu.com.vn/lap-trinh-aspnet-core/razor-view-engine-trong-aspnet-core-mvc-240.html>