**TRƯỜNG CAO ĐẲNG KINH TẾ THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-------**🙠🟓🙢**-----**

**BÁO CÁO BÀI TẬP LỚN**

**MÔN PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ**

**ĐỒ ÁN: AWESOME ICONS**

**NHÓM: 07**

**Sinh viên thực hiện:**

**BÙI QUỐC BẢO – 23661086**

**HUỲNH TẤN TRIỂN – 23662003**

**NGUYỄN TUẤN KIỆT – 23662007**

**HỒ QUỲNH ANH – 23662040**

**CHIÊM TẤN HƯNG – 23662006**

*Thành phố Hồ Chí Minh,ngày tháng 12 năm 2024*

**TRƯỜNG CAO ĐẲNG KINH TẾ THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-------**🙠🟓🙢**-----**

**Đề tài:** AWESOME ICONS

**Nhóm: 07**

**Mã môn học: 010100110901 Khóa: K19**

**Sinh viên thực hiện:**

**Mssv: 23661086 Họ và tên: BÙI QUỐC BẢO**

**Mssv: 23662003 Họ và tên: HUỲNH TẤN TRIỂN**

**Mssv: 23662007 Họ và tên: NGUYỄN TUẤN KIỆT**

**Mssv: 23662040 Họ và tên: HỒ QUỲNH ANH**

**Mssv: 23662006 Họ và tên: CHIÊM TẤN HƯNG**

**Giảng viên hướng dẫn: THẠC SĨ PHẠM ĐẰNG PHƯƠNG**

***Thành phố Hồ Chí Minh,ngày tháng 12 năm 2024***

# BẢNG ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN CÔNG VIỆC NHÓM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT DS** | **MSSV** | **Họ và tên** | **Công việc đảm nhận** | **Cá nhân tự đánh giá kết quả** | **Nhóm đánh giá kết quả** | **GV đánh giá** |
| 1 |  | Nguyễn Nhật Nguyên | - Mô tả lý thuyết phản ứng phân hạch - Tìm dữ liệu liên quan về trang web - Kiểm dò nội dung chương 3 | Hoàn thành tốt, đúng hạn | - Vài hình ảnh còn mờ, không rõ mục đích - Nhiệt tình tương tác với nhóm |  |
| 2 |  |  |  |  |  |  |

# LỜI NÓI ĐẦU

Trong thời đại công nghệ 4.0, việc phát triển phần mềm mã nguồn mở đã trở thành một xu hướng mạnh mẽ, được cộng đồng công nghệ trên toàn thế giới đón nhận và phát triển. Phần mềm mã nguồn mở không chỉ mở ra cơ hội lớn cho sự sáng tạo, đổi mới mà còn góp phần thúc đẩy sự hợp tác, chia sẻ kiến thức và tài nguyên giữa các nhà phát triển.

Đồ án của nhóm em nhằm mục đích nghiên cứu và ứng dụng các công cụ để phát triển phần mềm mã nguồn mở cũng như các nguyên tắc, kiến thức cần thiết để phát triển một phần mềm. Qua đó, nhóm em có thể hiểu rõ hơn về quy trình phát triển, lợi ích, cũng như những thách thức khi tham gia vào cộng đồng mã nguồn mở.

Nhóm em hy vọng rằng đồ án này sẽ mang lại cái nhìn tổng quan và sâu sắc hơn về phần mềm mã nguồn mở, đồng thời khuyến khích tinh thần hợp tác và chia sẻ trong việc phát triển phần mềm. Môn học phát triển phần mềm mã nguồn mở không chỉ là cơ hội để áp dụng những kiến thức đã học vào thực tiễn, mà còn là bước đệm để nhóm em tiếp tục khám phá và góp phần vào cộng đồng mã nguồn mở trong tương lai.

# LỜI CẢM ƠN

Nhóm em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến giảng viên hướng dẫn (Thạc sĩ Phạm Đằng Phương) của nhóm em, người đã dạy nhóm em môn học bổ ích này cũng như hỗ trợ nhóm em trong quá trình hoàn thành đồ án. Sự chỉ bảo nhiệt tình và kinh nghiệm quý báu của thầy là nguồn động lực lớn lao giúp nhóm em hoàn thành tốt công việc của mình.

Nhóm em cũng xin cảm ơn trường Cao Đẳng Kinh tế TP Hồ Chí Minh, nơi đã tạo điều kiện thuận lợi nhất để nhóm em học tập và rèn luyện trong suốt thời gian qua. Môi trường học tập năng động và hiện đại của nhà trường đã giúp em tích lũy kiến thức và kỹ năng cần thiết để phát triển bản thân.

Nhóm em xin chân thành cảm ơn tất cả mọi người đã hỗ trợ cho đồ án này.

# DANH MỤC CÁC TỪ VIẾT TẮT TIẾNG ANH

|  |  |
| --- | --- |
| **Từ viết tắt** | **Ý Nghĩa** |
| OSS | Open-source software |
| GPL | GNU General Public License |
| CC | Creative Commons (CC) License |
| VCS | Version Control System |
| Repo | Repository |
|  |  |
|  |  |
|  |  |
|  |  |

# MỤC LỤC

[BẢNG ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN CÔNG VIỆC NHÓM i](#_Toc184513551)

[LỜI NÓI ĐẦU ii](#_Toc184513552)

[LỜI CẢM ƠN iii](#_Toc184513553)

[DANH MỤC CÁC TỪ VIẾT TẮT TIẾNG ANH iv](#_Toc184513554)

[MỤC LỤC 1](#_Toc184513555)

[Chương 1: Tổng quan về phần mềm mã nguồn mở 1](#_Toc184513556)

[1.1. Phần mềm mã nguồn mở (OSS) là gì? 1](#_Toc184513557)

[1.2. Đặc điểm của phần mềm mã nguồn mở 1](#_Toc184513558)

[1.2.1. Tự do truy cập mã nguồn: 1](#_Toc184513559)

[1.2.2. Tự do sửa đổi, tùy chỉnh: 1](#_Toc184513560)

[1.2.3. Hiệu quả về mặt chi phí: 1](#_Toc184513561)

[1.2.4. Tính độc lập của nhà cung cấp: 1](#_Toc184513562)

[1.3. Những lợi ích và thách thức khi sử dụng phần mềm mã nguồn mở 2](#_Toc184513563)

[1.3.1. Lợi ích: 2](#_Toc184513564)

[1.3.2. Thách thức: 2](#_Toc184513565)

[1.4. Giấy phép mã nguồn mở: 2](#_Toc184513566)

[1.4.1. Giấy phép mã nguồn mở là gì? 2](#_Toc184513567)

[1.4.2. Các loại giấy phép mã nguồn mở phổ biến: 2](#_Toc184513568)

[1.4.3. Một số so sánh cơ bản giữa phần mềm nguồn mở và nguồn đóng: 3](#_Toc184513569)

[Chương 2: MÔ TẢ CHI TIẾT DỰ ÁN: 3](#_Toc184513570)

[2.1. Ý tưởng đồ án: 3](#_Toc184513571)

[2.2. Mô tả đồ án: 4](#_Toc184513572)

[2.3. Giấy phép sử dụng trong đồ án: 4](#_Toc184513573)

[2.4. Phần mềm sử dụng: Inkscape 4](#_Toc184513574)

[2.4.1. Lý do lựa chọn: 4](#_Toc184513575)

[2.4.2. Hạn chế: 5](#_Toc184513576)

[2.4.3. Đóng góp cho inkscape: 5](#_Toc184513577)

[Chương 3: 7](#_Toc184513578)

[Chương 4: Tài liệu tham khảo 7](#_Toc184513579)

# Tổng quan về phần mềm mã nguồn mở

## Phần mềm mã nguồn mở (OSS) là gì?

* Phần mềm mã nguồn mở là phần mềm mà mã nguồn của nó được công khai cho cộng đồng, được sử dụng cho bất kì mục đích nào( kể cả thương mại) theo điều khoản của giấy phép mà phần mềm đăng ký. Người sử dụng có thể tải mã nguồn về/thay đổi, chỉnh sửa và xuất/chia sẻ/đóng góp lại phiên bản của mã nguồn hoặc dự án của họ trở lại cộng đồng.

## Đặc điểm của phần mềm mã nguồn mở

### Tự do truy cập mã nguồn:

* OSS cho phép người sử dụng kiểm tra nội dung mã nguồn, tìm hiểu được luồng hoạt động của mã nguồn, tăng tính minh bạch, xác minh tính bảo mật của phần mềm, khuyến khích sự hợp tác, đóng góp, sửa lỗi cho phần mềm.

### Tự do sửa đổi, tùy chỉnh:

* OSS giúp tăng tính linh hoạt nhờ vào việc điều chỉnh mã nguồn sao cho phù hợp với nhu cầu của chúng ta hoặc để có thể tích hợp với các hệ thống khác, điều mà không phải lúc nào cũng thực hiện được với phần mềm độc quyền.

### Hiệu quả về mặt chi phí:

* OSS thường có sẵn miễn phí hoặc chi phí thấp hơn đáng kể so với các phần mềm bản quyền do không phải trả phí bản quyền, thích hợp cho các công ty vừa và nhỏ, các công ty startup,..

### Tính độc lập của nhà cung cấp:

* OSS giúp người dùng không ràng buộc vào nhà cung cấp, không phụ thuộc một nhà cung cấp duy nhất để cập nhật, sửa lỗi, nâng cấp,.. mà phần mềm mã nguồn mở thường được phát triển bởi cộng đồng các lập trình viên và người dùng toàn cầu.

## Những lợi ích và thách thức khi sử dụng phần mềm mã nguồn mở

### Lợi ích:

* Về kinh tế: miễn phí hoặc giá thành thấp so với phần mềm nguồn đóng.
* Về bảo mật: mã nguồn của phần mềm được đọc, kiểm tra bởi nhiều người, tăng khả năng phát hiện lỗi, mã độc,.. và thường sẽ có các bản vá lỗi nhanh hơn so với phần mềm nguồn đóng.
* Về cộng đồng: phần mềm nguồn mở phát triển từ sự đóng góp của cộng đồng trên thế giới, những ý tưởng, tư duy sẽ đa dạng hơn đồng thời phát triển kỹ năng viết mã, tư duy phát triển dự án của lập trình viên, thúc đẩy sự cộng tác giữa các lập trình viên trên toàn cầu với nhau.

### Thách thức:

* Có thể thiếu sự “hỗ trợ chuyên nghiệp” như phần mềm nguồn đóng do sự hỗ trợ đến từ cộng đồng nên có thể không ổn định, dẫn đến chậm trễ trong việc tìm giải pháp. Người dùng có thể phải tự sửa lỗi, gây khó khăn cho người dùng không có kỹ thuật, kiến thức cao. Trường hợp tệ nhất là phần mềm không còn được phát triển tiếp nữa.
* Độ phức tạp trong sử dụng: 1 số phần mềm nguồn mở yêu cầu phải có kiến thức kỹ thuật cao để cài đặt, sử dụng. Bên cạnh đó, việc thiếu tài liệu hoặc tài liệu không dễ hiểu cũng là 1 vấn đề cần phải đối mặt.
* Tính tương thích: do được phát triển từ cộng đồng, nên một số phần mềm nguồn mở có thể không tuân theo các tiêu chuẩn chung nào đó dẫn đến phải tùy chỉnh để có thể tương thích với các phần mềm, hệ thống khác.
* Về giấy phép: việc sử dụng phần mềm mã nguồn mở yêu cầu người dùng, tổ chức phải có kiến thức, sự hiểu biết về các giấy phép mã nguồn mở, tránh vi phạm các điều khoản pháp lý của giấy phép. Nhất là khi phát triển phần mềm mã nguồn mở thành phần mềm thương mại.

## Giấy phép mã nguồn mở:

### Giấy phép mã nguồn mở là gì?

* Là một loại giấy phép cho phần mềm máy tính và các sản phẩm khác nhằm cho phép mã nguồn, bản thiết kế hoặc thiết kế được sử dụng, sửa đổi và/hoặc chia sẻ theo các điều khoản và điều kiện được xác định.

### Các loại giấy phép mã nguồn mở phổ biến:

* GNU General Public License( GPL): giấy phép này yêu cầu bất kỳ phần mềm được phát triển từ phần mềm sử dụng giấy phép GPL đều phải được cấp phép theo cùng giấy phép. Đồng thời, mã nguồn phải được công khai hoặc có cách để truy cập. GPL là một giấy phép copyleft mạnh mẽ.
* MIT License: giấy phép này cho phép người dùng tự do sử dụng, sao chép, sửa đổi,và phân phối phần mềm mà không yêu cầu cùng loại giấy phép. Có nghĩa là phần mềm được phát triển lại có thể được cấp phép theo nguồn đóng, và không cần công khai mã nguồn, ví dụ như mục đích thương mại. MIT là một giấy phép không copyleft.
* Apache License 2.0: giấy phép này cho phép người dùng sử dụng phần mềm cho bất kỳ mục đích nào, phân phối, sửa đổi và phân phối các phiên bản đã sửa đổi của phần mềm theo các điều khoản của giấy phép, bao gồm các điều khoản bảo vệ người dùng khỏi các vụ kiện tụng bằng sáng chế. Apache License là giấy phép không copyleft, có thể tương thích(kết hợp) với GPL v3 miễn là phần mềm phái sinh/phiên bản đã sửa đổi của phần mềm được cấp phép theo GPLv3.
* BSD License: giấy phép này cho phép phân phối lại ở cả các định dạng nhị phân và mã nguồn, mà không bắt ép bất kỳ điều gì trong mọi trường hợp. Nó chỉ yêu cầu công nhận các tác giả. Giấy phép BSD (không giống như một số giấy phép khác, ví dụ như [GPL](https://en.wikipedia.org/wiki/GPL) ) không yêu cầu phải phân phối mã nguồn. BSD là một giấy phép không copyleft.
* Creative Commons Licenses: là một số giấy phép bản quyền các “ tác phẩm ” ( không phải phần mềm) cho phép phân phối miễn phí các tác phẩm bản quyền, được sử dụng khi tác giả muốn cho người khác quyền chia sẻ, sử dụng, và xây dựng dựa trên tác phẩm mà tác giả đã tạo ra. Có nhiều loại giấy phép CC và 5 phiên bản 1.0(2002), 2.0(2004),2.5(2005),3.0( 2007), 4.0(2013- toàn diện nhất). mỗi giấy phép CC có 4 quyền cơ bản: ghi công(BY- toàn quyền, chỉ cần ghi công tác giả), chia sẻ tương tự(SA – toàn quyền, nhưng cần sử dụng giấy phép tương tự cho tác phẩm phái sinh/phối lại), Phi thương mại(NC- không được thương mại hóa các tác phẩm phái sinh/phối lại), không có tác phẩm phái sinh(ND- sử dụng nguyên văn của tác phẩm, từ phiên bản 4.0 tác phẩm phái sinh được sử dụng nhưng không được chia sẻ.

### Một số so sánh cơ bản giữa phần mềm nguồn mở và nguồn đóng:

|  |  |  |
| --- | --- | --- |
|  | Phần mềm nguồn mở | Phần mềm nguồn đóng |
| Mã nguồn | Công khai, mọi người có thể truy cập, sửa đổi,.. | Không công khai, chỉ nhà phát triển mới có thể truy cập |
| Chi phí | Miễn phí hoặc phí thấp | Thường cao hơn OSS do có phí bản quyền |
| Sở hữu | Không có 1 tổ chức/cá nhân cụ thể | Thường thuộc sở hữu công ty, cá nhân nào đó |
| Hỗ trợ | Từ cộng đồng | Từ nhà cung cấp |
| Giấy phép | Các giấy phép mở như GPL, BSD,Apache,.. | Giấy phép riêng( thường mang tính thương mại) |
| Tính bảo mật | Tùy thuộc khả năng phát hiện, đóng góp của cộng đồng | Phụ thuộc vào bản vá của nhà cung cấp nếu có |

# MÔ TẢ CHI TIẾT DỰ ÁN:

## Ý tưởng đồ án:

* Tạo ra một bộ icon kèm theo các tệp hình ảnh SVG của các icon đó. Các icon này có thể được sử dụng rộng rãi trong thiết kế web, in ấn và nhiều lĩnh vực khác. Mục tiêu chính của đồ án là cung cấp nguồn tài nguyên miễn phí, dễ truy cập và dễ sử dụng cho cộng đồng thiết kế và phát triển.

## Mô tả đồ án:

* Đồ án sẽ xây dựng và phát hành một bộ sưu tập các icon đa dạng, phong phú đi kèm với các tệp hình ảnh SVG. Các icon này sẽ được thiết kế tỉ mỉ, phù hợp với nhiều mục đích sử dụng khác nhau như thiết kế giao diện người dùng (UI), đồ họa in ấn, và các dự án sáng tạo khác. Bộ sưu tập sẽ bao gồm nhiều chủ đề và phong cách khác nhau, đáp ứng nhu cầu đa dạng của người sử dụng, nhất là các bạn sinh viên với mục đích học tập.

## Giấy phép sử dụng trong đồ án:

Dự án sẽ được phát hành dưới hai loại giấy phép: GNU General Public License (GPL) và Creative Commons (CC) License: CC BY

* **GNU General Public License (GPL)**: Cho phép người dùng tự do sử dụng, sao chép, phân phối và sửa đổi phần mềm, với điều kiện mọi phiên bản sửa đổi cũng phải được phát hành dưới cùng giấy phép. GPL sẽ cho phép mọi người sử dụng mã nguồn của đồ án một cách tự do, nhất là các bạn sinh viên với mục đích học tập. Đồng thời, hạn chế sự “ không chia sẻ” đối với người dùng sử dụng mã nguồn đồ án này phát triển thành mã nguồn “có thể thương mại”, trái với mục đích ban đầu của đồ án cả về hình ảnh SVG lẫn mã nguồn.
* **Creative Commons License: CC BY**: Cho phép sử dụng, phân phối, và chỉnh sửa tác phẩm với điều kiện phải ghi công tác giả ban đầu. Giấy phép này cho phép người dùng tự do sử dụng các hình ảnh theo mục đích của họ. Điều này khuyến khích sự sáng tạo và sử dụng rộng rãi trong cộng đồng.
* Cả hai giấy phép đều có mục đích là thúc đẩy sự sáng tạo, đóng góp từ cộng đồng, và ngược lại, hỗ trợ miễn phí cho người dùng khi sử dụng.

## Phần mềm sử dụng: Inkscape

### Lý do lựa chọn:

* Inkscape là một phần mềm đồ họa mã nguồn mở, miễn phí và nhẹ hơn so với các phần mềm thiết kế khác. Giao diện đơn giản, thân thiện với người dùng mới bắt đầu, dễ học và dễ tiếp cận. Hơn nữa, Inkscape có khả năng chạy trên nhiều nền tảng, từ Windows, macOS đến Linux, làm cho nó trở thành lựa chọn linh hoạt và tiện lợi.
* Inkscape sử dụng giấy phép GPL version 2 nên người dùng có thể tự do sử dụng, phân phối hoặc sửa đổi mã nguồn Inkscape cho bất kỳ mục đích nào, thương mại hoặc phi thương mại, mà không có bất kỳ hạn chế nào, chỉ có điều người sửa đổi mã nguồn phải cung cấp quyền truy cập mã nguồn của phiên bản đã sửa đổi.

### Hạn chế:

* Mặc dù Inkscape là công cụ mạnh mẽ cho việc thiết kế đồ họa, nhưng nó vẫn thiếu một số tính năng chuyên nghiệp mà các phần mềm Adobe cung cấp. Điều này có thể hạn chế một số khía cạnh của quá trình thiết kế chuyên sâu.

### Đóng góp cho inkscape:

#### **Không code:**

* Có thể đóng góp bằng cách: kiểm tra, báo cáo lỗi, dịch phần mềm( cho người dùng ở nhiều quốc gia), viết/dịch trang wiki, hoặc đơn giản là giới thiệu inkscape để mọi người biết đến nhiều hơn, tạo nguồn thu cho nhóm tác giả.

#### **Code và phát triển thêm:**

* Mã nguồn: <https://gitlab.com/inkscape/inkscape>
* Hướng dẫn dùng Git/GitLab để lấy mã nguồn: <https://inkscape.org/develop/getting-started/>
* Những quy định về cách viết mã khi muốn đóng góp: <https://inkscape.org/develop/coding-style/>
* Debugging: <https://inkscape.org/develop/debugging/>
* Viết Extension: <https://inkscape.org/develop/extensions/>

# Các công cụ phát triển phần mềm dùng trong đồ án này

## Visual Studio Code (VS Code):

* VS Code là một trình soạn thảo mã nguồn mở, miễn phí và mạnh mẽ, được phát triển bởi Microsoft. VS Code nổi bật bởi sự linh hoạt và khả năng tùy chỉnh cao, phù hợp cho nhiều ngôn ngữ lập trình và nền tảng khác nhau.
* VS Code sử dụng giấy phép MIT.
* Mã nguồn và cách đóng góp cho VS Code: <https://github.com/Microsoft/vscode>

## Git và Github:

### Khái niệm version control system ( VCS ):

* VCS là một công cụ quan trọng trong phát triển phần mềm, cho phép các nhà phát triển theo dõi và quản lý các thay đổi trong mã nguồn của dự án theo thời gian.
* Các loại VCS:
  + Local VCS: lưu trữ các phiên bản của file trên máy tính người dùng, không hỗ trợ làm việc nhóm. Ví dụ : Revision Control System ( RCS)
  + Centralized VCS: một máy chủ sẽ lưu trữ tất cả phiên bản của dự án, người dùng cần kết nối đến máy chủ để lấy/cập nhật mã nguồn, nếu máy chủ bị sập thì không ai có thể làm việc hay lưu file được, thậm chí nếu ổ đĩa cứng chứa CSDL không được backup, toàn bộ lịch sử dự án sẽ bị mất hết. Ví dụ: Subversion (SVN)
  + Distributed VCS: mỗi người dùng sẽ có một bản sao đầy đủ của toàn bộ repository trên máy của mình, hỗ trợ tốt cho làm việc nhóm và làm việc offline. Ví dụ : Git
* Những lợi ích của VCS:
  + Theo dõi thay đổi: Giúp theo dõi tất cả các thay đổi được thực hiện trên mã nguồn, từ việc ai đã thay đổi cái gì và khi nào.
  + Phục hồi dữ liệu: Cho phép quay lại các phiên bản trước đó nếu có lỗi xảy ra.
  + Cộng tác hiệu quả: Hỗ trợ làm việc nhóm, cho phép nhiều người cùng làm việc trên một dự án mà không lo xung đột.
  + Quản lý nhánh: Hỗ trợ việc tạo và quản lý các nhánh (branches) để thử nghiệm tính năng mới mà không ảnh hưởng đến mã nguồn chính.

### Git:

* Là một DVCS được phát triển bởi Linux Torvald vào năm 2005, ban đầu dành cho việc phát triển nhân Linux. Git có thể khả năng chạy trên nhiều hệ điều hành khác nhau như Linux, Windows, Mac, …
* Những lợi ích khi sử dụng Git: Cho phép người dùng tạo các nhánh mới, hợp nhất (merge) các nhánh, và xử lý xung đột mã nguồn một cách hiệu quả. Với Git, người dùng có thể làm việc offline và sau đó đồng bộ các thay đổi với repository trung tâm khi có kết nối internet

### Github:

* Là dịch vụ lưu trữ dựa trên web cho các dự án sử dụng hệ thống quản lý phiên bản Git, giúp dễ dàng chia sẻ và cộng tác với các thành viên khác.. Github cung cấp nhiều tính năng hỗ trợ cộng tác và quản lý dự án như Pull Request, Issues, và GitHub Actions.

### Cài đặt Git, tạo tài khoản Github:

* Cài đặt Git: truy cập link <https://git-scm.com/> tải phiên bản Git phù hợp hệ điều hành máy tính đang sử dụng, sau đó chạy file cài đặt và kiểm tra Git đã cài đặt thành công chưa bằng cmd hoặc Git bash ( click chuột phải -> chọn Git bash here):

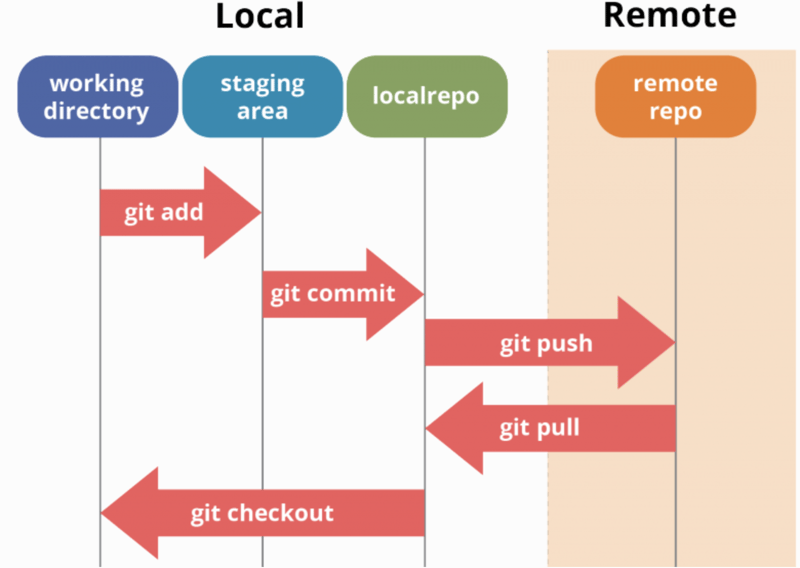
A computer screen shot of a computer program

Description automatically generated

* Tạo tài khoản Github: truy cập link https://github.com/ , chọn Sign Up và tiến hành tạo tài khoản như các trang web khác

### Một số khái niệm cơ bản trong Git:

* Repository (repo): một kho chứa chính cho toàn bộ mã nguồn, tập tin, thư mục, phiên bản, lịch sử thay đổi,.. có nội dung cần cho triển khai, phát triển và quản lý dự án trong SVC. Có 2 loại repo:
  + Local repo: nằm ở máy tính cá nhân hoặc server phát triển ở local, mọi thao tác cá nhân sẽ diễn ra ở repo này.
  + Remote repo (mặc định gọi Origin) : nằm ở “xa”, thường là một máy chủ nào đó, thao tác thông qua local repo. Khi muốn chia sẻ công việc đang làm ở local repo, ta sẽ tải lên remote repo để công khai nội dung công việc đó, để người dùng khác có thể nắm bắt được công việc ta đang làm.
* Working tree: là thư mục trên máy tính cá nhân nơi chứa các tệp và thư mục dự án. Đây là nơi thực hiện các thao tác như thêm, chỉnh sửa, hoặc xóa các tệp trước khi “đưa lên” local repo.
* Staging Area (Index) : vùng trung gian giữa working tree và repository. Nơi sẽ thực hiện các chỉnh sửa, quyết định cuối cùng trước khi “commit” dự án vào repo ( ở đây là local repo).
* Commit: quá trình “tải” các file, dự án,.. từ Staging Area lên Repository. Mỗi commit là một snapshot( bản chụp) dự án tại thời điểm hiện tại (thời điểm commit), có thể theo dõi lịch sử phát triển, khôi phục lại các phiên bản trước,..
* Branch (nhánh): một phiên bản độc lập chứa các tệp, dự án,.. như repository chính tại thời điểm tạo branch. Ở branch, ta có thể chỉnh sửa code, thêm tính năng cho dự án, fix những bug được phát hiện,… Sau khi kiểm tra cho kết quả đáp ứng được công việc, ta sẽ “trộn(merge)” branch này với repository, SVC sẽ phát hiện những thay đổi ở branch khác với repository được merge, sau đó thực hiện áp dụng những thay đổi này ở repository được merge đó.
* Checkout: quá trình chuyển đổi từ nhánh này sang nhánh khác để làm việc.
* Merge: quá trình trộn các commit từ các branch vào repository chính.
* Push: quá trình đẩy code từ local repo lên remote repo.
* Pull: quá trình lấy code về local repo từ remote repo.
* Một số workflow cơ bản của Git:



A diagram of a computer

Description automatically generated

### Một số câu lệnh phổ biến:

|  |  |
| --- | --- |
| git --version | Kiểm tra phiên bản |
| git config --global user.name "CTH" | Đặt username là <CTH > |
| git config --global user.email "cthung2301@gmail.com" | Đặt useremail là <cthung2301@gmail.com > |
| git init | Khởi tạo local repo |
| git status | Kiểm tra trạng thái các file |
| git add . | Add các file vào staging area |
| git commit -m “<thông điệp>” | Commit các file vào local repo |
| git log | Kiểm tra lịch sử commit |
| git help --all | Trợ giúp cách dùng các câu lệnh |
| git branch <tên branch> | Tạo một branch mới để thực hiện coding |
| git branch | Kiểm tra các branch hiện có |
| Git checkout [<-b>] <tên branch> | Di chuyển qua một branch, thêm -b nếu muốn vừa di chuyển vừa tạo branch |
|  |  |
|  |  |
|  |  |
|  |  |

### Demo Git/Github cho đồ án:

#### **Git:**

* Chọn thư mục chung chứa toàn bộ dự án -> chuột phải -> Git bash here.
* Kiểm tra phiên bản Git:

A screenshot of a computer

Description automatically generated

* Thiết lập username và user email, để khi commit sẽ có thông tin người thực hiện thay đổi này:

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

* Khởi tạo local repo( di chuyển đến thư mục chính chứa toàn bộ dự án), lúc này Git sẽ tạo một thư mục ẩn .git giúp theo dõi mọi thay đổi trong thư mục chính):

A screen shot of a computer

Description automatically generated

* Kiểm tra trạng thái các file trong thư mục: untracked file( màu đỏ, chưa được đưa vào staging area), tracked file( màu xanh lá, đã được đưa vào staging area), modified file( màu vàng, file đã được thay đổi nhưng chưa được đưa vào staging are)

A screen shot of a computer

Description automatically generated

* Add(thêm) file vào staging area: git add . hoặc git add -A hoặc git add –all

A computer screen with yellow text

Description automatically generated

A screenshot of a computer screen

Description automatically generated

* Commit các file ở staging area vào local repo:

A screen shot of a computer screen

Description automatically generated

A screen shot of a computer screen

Description automatically generated

* Kiểm tra lịch sử commit:

A computer screen with text and numbers

Description automatically generated

* Tạo nhánh để thực hiện các thay đổi trong code tránh ảnh hưởng code chính, ở đây nhóm em sẽ tạo một nhánh add-brand-icon để thêm các brand icons vào trang web:

A screenshot of a computer

Description automatically generated

* Kiểm tra các branch hiện có, dấu \* biểu thị ta đang ở branch đó( ở đây là master):

A screenshot of a computer

Description automatically generated

* Di chuyển qua branch khác:

A screenshot of a computer screen

Description automatically generated

* Thực hiện các thay đổi ở mã nguồn, kiểm tra, nếu ổn định, ta sẽ thực hiện add file, commit file vào local repo của “branch”, sau đó di chuyển về master( branch chính ở local):

A screenshot of a computer screen

Description automatically generated

* Do mã nguồn đã được test ổn định, ta thực hiện quá trình merge code từ branch phụ sang master branch:

# Docker

# Tài liệu tham khảo

Trích dẫn các tài liệu mà sinh viên đã tham khảo trong quá trình làm bài ( bao gồm giáo trình, tài liệu online, tài liệu khác… theo mẫu sau)

**Tài liệu tham khảo:**

1. Tài liệu NodeJS [online] Có sẵn tại: https://nodejs.org/en/docs/ [Truy cập ngày 08 tháng 9 năm 2021].
2. Hướng dẫn cấu hình CSS với npm [online] Có sẵn tại: <https://docs.npmjs.com/cli/v7/configuring-npm/package-json> [Truy cập ngày 08 tháng 9 năm 2021].
3. Bootstrap Tutorial [online] Có sẵn tại: <https://www.w3schools.com/bootstrap4/default.asp> [Truy cập ngày 08 tháng 9 năm 2021].
4. Tailwind CSS Documentation [online] Có sẵn tại: <https://tailwindcss.com/docs> [Truy cập ngày 08 tháng 9 năm 2021].
5. Vuetify Documentation [online] Có sẵn tại: https://vuetifyjs.com/en/getting-started/installation/ [Truy cập ngày 08 tháng 9 năm 2021].
6. Vue.js Documentation [online] Có sẵn tại: https://vuejs.org/v2/guide/ [Truy cập ngày 08 tháng 9 năm 2021].
7. Pinia Documentation [online] Có sẵn tại: https://pinia.vuejs.org/ [Truy cập ngày 08 tháng 9 năm 2021].
8. Vue Router Documentation [online] Có sẵn tại: https://router.vuejs.org/ [Truy cập ngày 08 tháng 9 năm 2021].
9. Axios Documentation [online] Có sẵn tại: https://axios-http.com/docs/intro [Truy cập ngày 08 tháng 9 năm 2021].