

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**BÁO** **CÁO**

**Kỹ năng nghề nghiệp - SS004.P14**

**Nhóm 11**

**Github, Git và các ứng dụng**

|  |  |  |
| --- | --- | --- |
| 1. | Phan Hàn Thái | 23521419 |
| 2. | Trần Thảo Vy | 23521838 |
| 3. | Nguyễn Ngọc Tâm | 23521391 |
| 4. | Nguyễn Duy Tân | 23521402 |
| 5. | Đỗ Phú Sơn | 23521351 |

Sinh viên thực hiện:

|  |  |  |
| --- | --- | --- |
| 1. | Nguyễn Ngọc Tâm | 23521391 |
| 2. | Trần Thảo Vy | 23521838 |
| 3. | Phan Hàn Thái | 23521419 |
| 4. | Nguyễn Duy Tân | 23521402 |
| 5. | Đỗ Phú Sơn | 23521351 |

Sinh viên thực hiện:

**MỤC LỤC**

[I. GIỚI THIỆU CHUNG VỀ GITHUB VÀ GIT 2](#_Toc177571649)

[A. Github là gì? 2](#_Toc177571650)

[B. Git là gì? 2](#_Toc177571651)

[II. CÁC LỆNH VÀ CHỨC NĂNG CƠ BẢN 2](#_Toc177571652)

[A. Chức năng và lệnh Github trong một dự án 2](#_Toc177571653)

[1. Tab `< > Code` 2](#_Toc177571654)

[2. Tab `Issues` 4](#_Toc177571655)

[3. Tab `Pull Requests` 5](#_Toc177571656)

[4. Tab `Actions` 7](#_Toc177571657)

[5. Tab `Projects` 7](#_Toc177571658)

[6. Tab ‘Wiki’ 8](#_Toc177571659)

[7. Tab ‘Security’ 8](#_Toc177571660)

[8. Tab `Insights` 8](#_Toc177571661)

[9. Tab ‘Settings’ 9](#_Toc177571662)

[B. Chức năng và lệnh Git 10](#_Toc177571663)

[1. Các thuật ngữ 10](#_Toc177571664)

[2. Các câu lệnh cơ bản: 11](#_Toc177571665)

[III. CÁC ỨNG DỤNG CỦA GITHUB VÀ GIT 17](#_Toc177571666)

[A. Xây dựng và quản lý dự án lập trình 17](#_Toc177571667)

[B. Một số ứng dụng khác 17](#_Toc177571668)

**DANH SÁCH HÌNH ẢNH**

[Hình 1: Tab 'Code' 3](#_Toc177573037)

[Hình 2: Tab 'Commit' 4](#_Toc177573038)

[Hình 3: Tab 'Branches' 4](#_Toc177573039)

[Hình 4: 'Download' 5](#_Toc177573040)

[Hình 5: Tạo issue mới 5](#_Toc177573041)

[Hình 6: Một issue 6](#_Toc177573042)

[Hình 7: Nhãn dán 7](#_Toc177573043)

[Hình 8: Tab 'Pull Request' 8](#_Toc177573044)

[Hình 9: Phần comment 8](#_Toc177573045)

[Hình 10: Merge pull request 9](#_Toc177573046)

[Hình 11: Workflow 9](#_Toc177573047)

[Hình 12: Tạo project 10](#_Toc177573048)

[Hình 13: Tab 'Insights' 11](#_Toc177573049)

[Hình 14: git -v / git –version 13](#_Toc177573050)

[Hình 15: git -h / git --help 14](#_Toc177573051)

[Hình 16: git config 14](#_Toc177573052)

[Hình 17: git init 14](#_Toc177573053)

[Hình 18: git clone 15](#_Toc177573054)

[Hình 19: git status 15](#_Toc177573055)

[Hình 20: git add 15](#_Toc177573056)

[Hình 21: git commit 16](#_Toc177573057)

[Hình 22: git log 16](#_Toc177573058)

[Hình 23: git pull 17](#_Toc177573059)

[Hình 24: git push 17](#_Toc177573060)

[Hình 25: git branch 17](#_Toc177573061)

[Hình 26: git checkout 18](#_Toc177573062)

[Hình 27: git stash 18](#_Toc177573063)

[Hình 28: git merge 18](#_Toc177573064)

[Hình 29: git reset 19](#_Toc177573065)

# I. GIỚI THIỆU CHUNG VỀ GITHUB VÀ GIT

## Github là gì?

* Github là một nền tảng hỗ trợ lập trình viên phát triển, lưu trữ, quản lý và chia sẻ các dự án code mã nguồn mở. Từ năm 2018, Github là công ty con của Microsoft và được đặt trụ sở tại California.

## Git là gì?

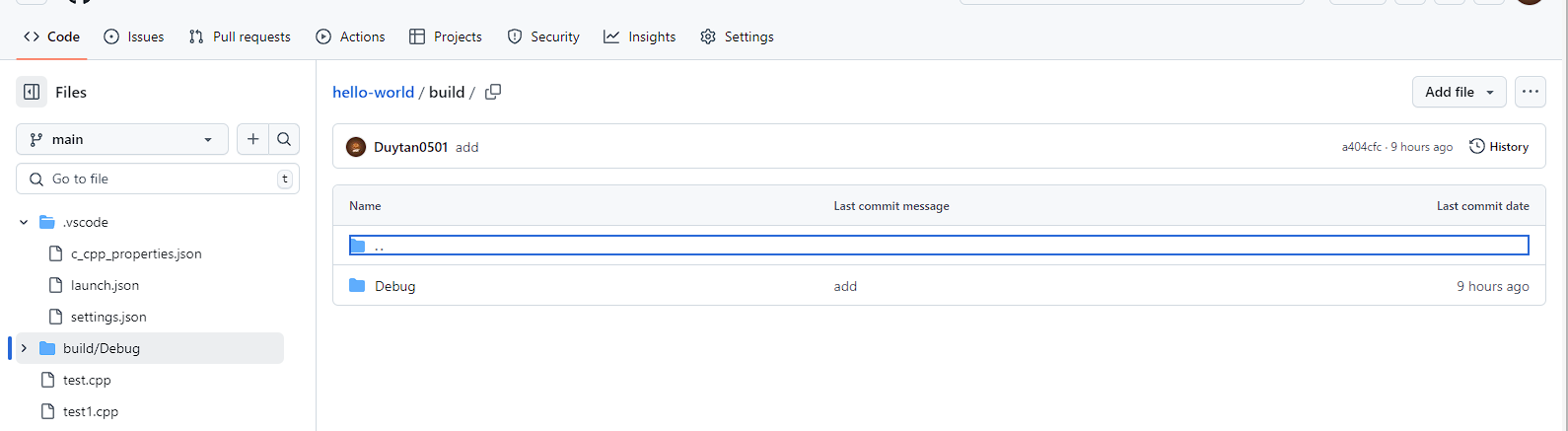
* Git là một hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS), hệ thống này cho phép nhiều lập trình viên quản lý chỉnh sửa trên cùng một codebase, lưu lại log để khôi phục và source code được lưu trữ đầy đủ cả ở máy chủ trung tâm và máy tính của lập trình viên. Git được tích hợp vào Github như một cách để quản lý các dự án trên Github.

# CÁC LỆNH VÀ CHỨC NĂNG CƠ BẢN

## Chức năng và lệnh Github trong một dự án

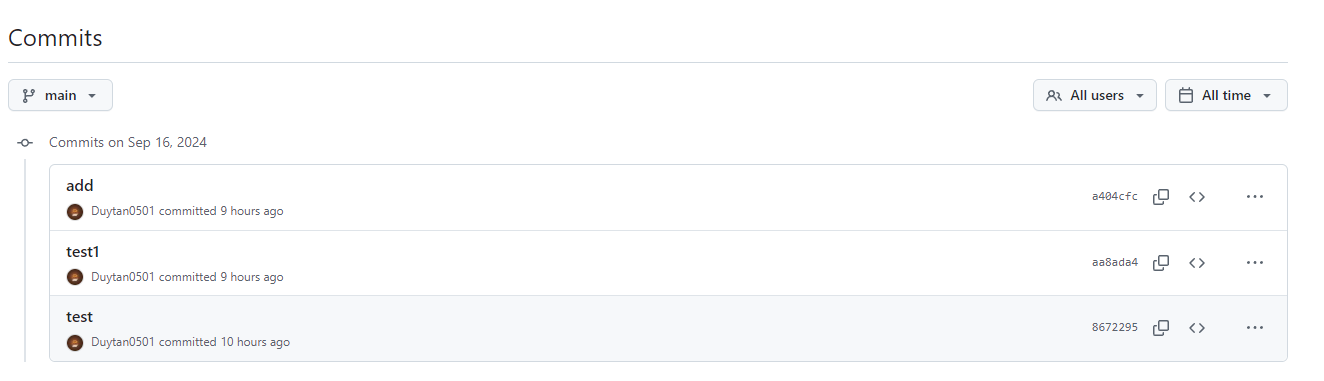
### Tab `< > Code`:

* + **Chức năng:** Đây là tab chính hiển thị toàn bộ source code của project, cho phép người dùng xem và tải xuống project.
  + **Các thao tác:**
    - ***Duyệt source code:*** Người dùng có thể xem các thư mục, tập tin trong dự án. Những tệp này có thể được xem chi tiết ngay trên GitHub mà không cần tải về.



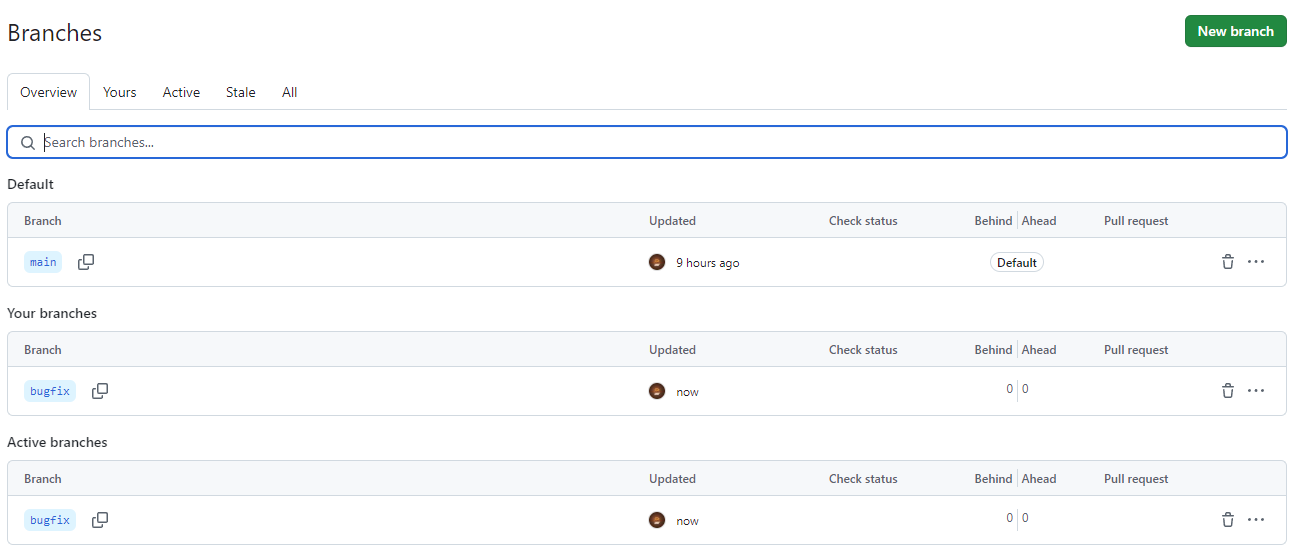
Hình : Tab 'Code'

* + - ***Xem commit:*** Cho phép người dùng xem lịch sử commit, nghĩa là toàn bộ những thay đổi đã được thực hiện trong dự án, bao gồm thông tin về người thực hiện, thời gian và mô tả của mỗi lần commit.



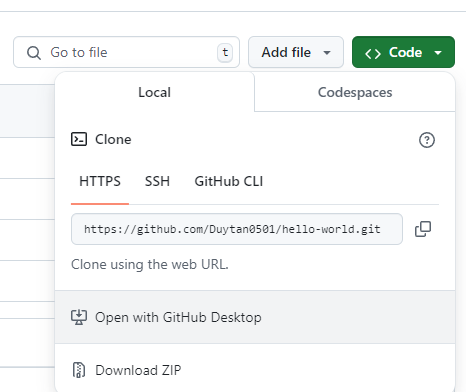
Hình : Tab 'Commit'

* + - ***Branch:*** Người dùng có thể chuyển đổi giữa các nhánh khác nhau của dự án. Mỗi nhánh là một phiên bản khác của source code, thường dùng để phát triển các tính năng riêng biệt hoặc tiến hành phiên bản thử nghiệm.



Hình : Tab 'Branches'

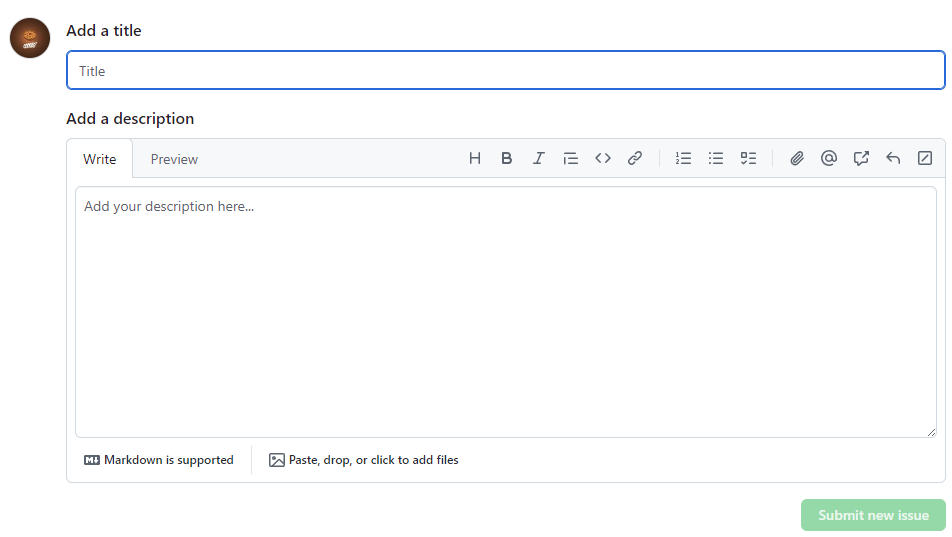
* + - ***Download:*** Bằng cách nhấn nút "Code", người dùng có thể tải toàn bộ source code của dự án về máy dưới dạng file `.zip` hoặc clone (sao chép) nó về qua GitHub CLI hoặc Git.



Hình : 'Download'

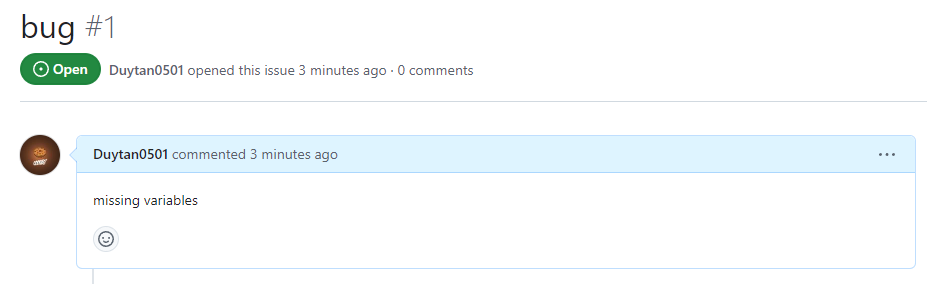
### Tab `Issues`:

* **Chức năng:** Quản lý các vấn đề (issue) phát sinh trong dự án như bug, một yêu cầu tính năng mới, hoặc thảo luận về vấn đề.
* **Các thao tác:**
  + - * ***Tạo issue mới:*** Tạo một issue mới để báo cáo lỗi hoặc yêu cầu cải tiến, đồng thời phân công cho các thành viên khác.



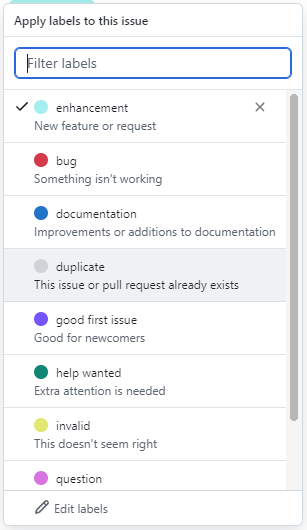
Hình : Tạo issue mới

* + - ***Theo dõi issue:*** Các issue hiện hữu được liệt kê với trạng thái như “Open” (Đang mở) hoặc “Closed” (Kết thúc). Thành viên có thể thêm bình luận để thảo luận về cách giải quyết.



Hình : Một issue

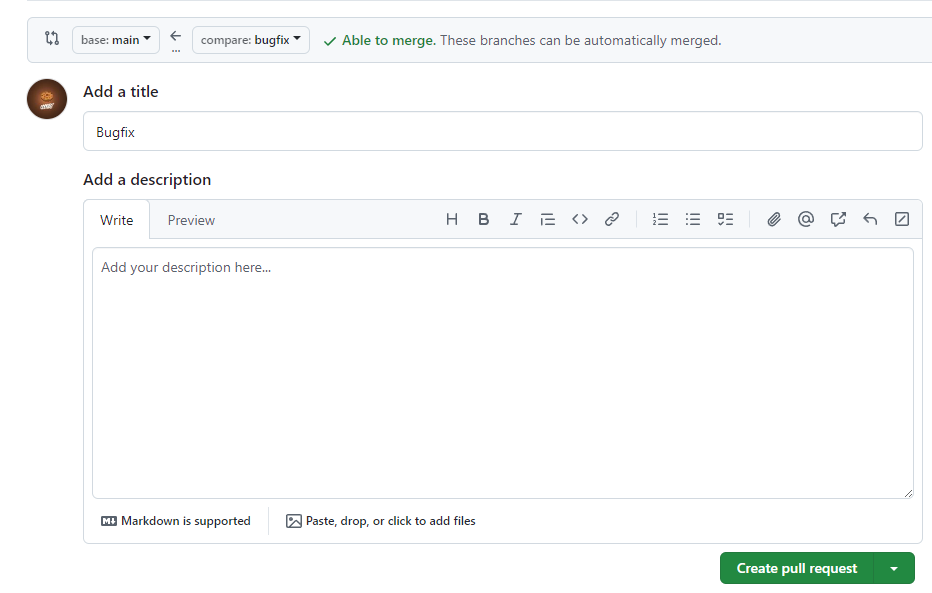
* + - ***Gán nhãn (Labels):*** Việc gán các nhãn như “bug” hay “enhancement” (cải tiến) sẽ giúp phân loại issue dễ dàng hơn.



Hình : Nhãn dán

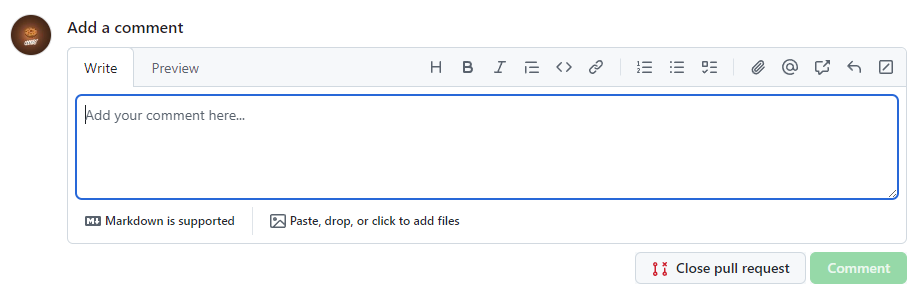
### Tab `Pull Requests`:

* **Chức năng:** Pull Requests (PR) là nơi mà các thành viên gửi yêu cầu hợp nhất source code từ nhánh của họ vào nhánh chính (thường là `main` hoặc `master`).
* **Các thao tác:**
  + ***Tạo PR mới:*** Tạo một PR khi muốn hợp nhất các thay đổi từ nhánh phụ vào nhánh chính.



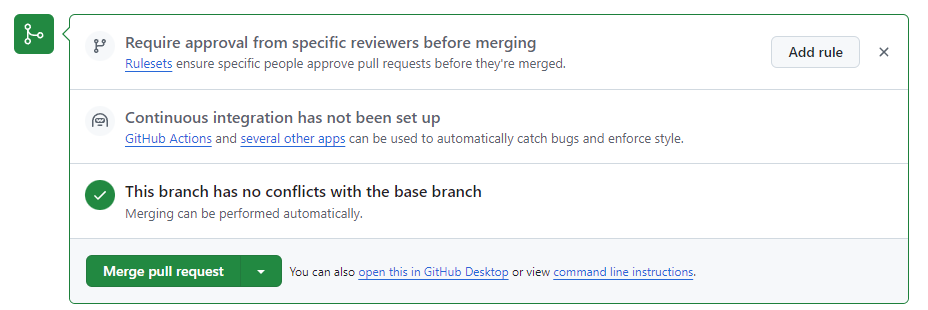
Hình : Tab 'Pull Request'

* + ***Thảo luận về PR:*** Mỗi PR đều có một mục bình luận để người dùng khác đánh giá source code trước khi nó được hợp nhất. Người dùng có thể thêm nhận xét, yêu cầu thay đổi hoặc xác nhận PR.



Hình : Phần comment

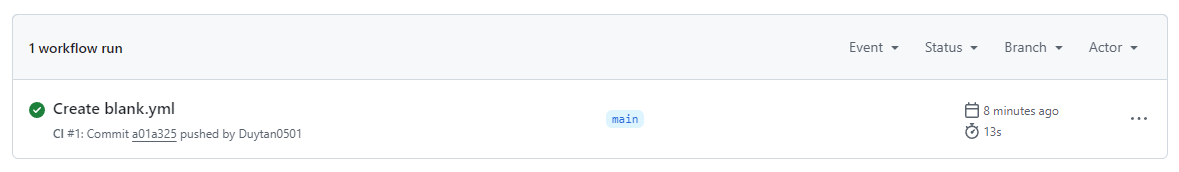
* + ***Merge PR:*** Sau khi PR đã được kiểm tra, chủ dự án hoặc người quản lý có thể hợp nhất nó vào nhánh chính.



Hình : Merge pull request

### Tab `Actions`:

* **Chức năng:** GitHub Actions cung cấp các công cụ tự động hóa như tự động chạy kiểm thử (test), triển khai các lệnh sau mỗi lần commit.
* **Các thao tác:**
* ***Xem trạng thái workflows:*** Tab này hiển thị trạng thái của các hành động tự động như “build”, “test”, hoặc “deploy”. Mỗi workflow sẽ có nhật ký (log) chi tiết cho người dùng theo dõi.

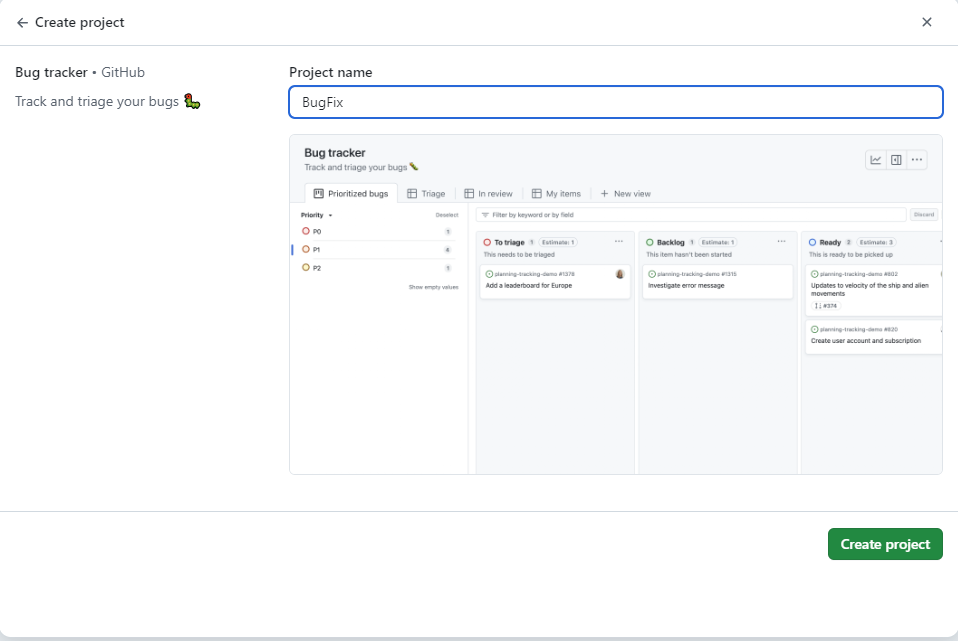


Hình : Workflow

* ***Tạo workflows mới:*** Thiết lập các workflow mới để tự động kiểm tra, xây dựng dự án hoặc triển khai nó lên máy chủ.

### Tab `Projects`:

* **Chức năng:** Nơi tổ chức các công việc liên quan đến dự án theo phương pháp Kanban board. Nó giống như một bảng công việc hỗ trợ quản lý, lập kế hoạch và theo dõi tiến độ của các task.
* **Các thao tác:**
  + ***Tạo Project:*** Tạo bảng dự án để sắp xếp các task, phân công chúng cho các thành viên và theo dõi tiến độ.



Hình : Tạo project

* + ***Theo dõi tiến trình:*** Có thể liên kết mỗi task với một issue hoặc pull request cụ thể, giúp quản lý dễ dàng hơn.

### Tab ‘Wiki’:

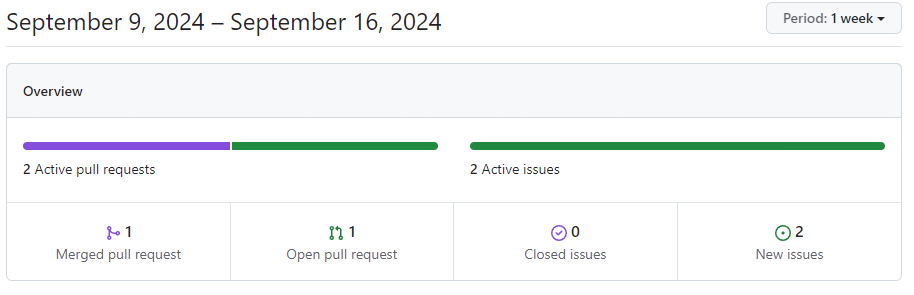
* **Chức năng:** Wiki là nơi lưu trữ và chia sẻ tài liệu chính của dự án. Chủ dự án hoặc thành viên nhóm có thể tạo nhiều trang wiki để giải thích cách sử dụng source code, hướng dẫn cài đặt hoặc tài liệu khác.
* **Các thao tác:**
  + ***Tạo và chỉnh sửa trang wiki:*** Thêm hoặc chỉnh sửa nội dung các trang wiki để cung cấp thông tin chi tiết cho người dùng.
  + ***Duyệt wiki:*** Tài liệu trong wiki được sắp xếp dưới dạng các trang để dễ dàng tham khảo.

### Tab ‘Security’:

* **Chức năng:** Cung cấp các công cụ quản lý các vấn đề bảo mật của dự án.
* **Các thao tác:**
  + ***Quản lý bảo mật:*** GitHub cung cấp các cảnh báo bảo mật, cho phép người dùng phát hiện lỗ hổng trong source code hoặc các thư viện phụ thuộc.

### Tab `Insights`:

* **Chức năng:** Cung cấp thống kê về tiến độ phát triển của dự án.
* **Các thao tác:**
* ***Xem số liệu:*** Người dùng có thể theo dõi các biểu đồ về số lượng commit, issues, số pull request, đóng góp của các thành viên cũng như sự phát triển của dự án theo thời gian.



Hình : Tab 'Insights'

### Tab ‘Settings’:

* **Chức năng:** Kiểm soát mọi khía cạnh của repository, từ bảo mật, thiết lập chung, quản lý nhánh, tự động hóa cho đến việc điều hành cộng tác viên.
* **Các thao tác:**
  + ***General (Cài đặt chung):***

*Repository name:* Người dùng có thể đổi tên của repository bằng cách chọn "Rename".

*Template repository:* Biến repository thành một template mẫu để tạo các dự án mới có thể sử dụng lại cùng cấu trúc thư mục và tệp tin.

*Danger Zone:* Nơi thực hiện các thay đổi quan trọng, cần phải cẩn thận vì có thể gây mất dữ liệu, thay đổi quyền sở hữu, hoặc xóa repository.

* + ***Collaborators:*** Quản lý quyền truy cập của các thành viên vào dự án. Chủ sở hữu có thể mời những người khác và chỉ định quyền cụ thể cho họ (chẳng hạn như chỉ xem, hoặc commit).
  + ***Branches:*** Thiết lập quy tắc bảo vệ cho từng nhánh, nhằm tránh tình trạng nhánh chính bị ghi đè hoặc thay đổi không hợp lệ.
  + ***Tags:*** Quản lý các tag được gắn cho các commit quan trọng trong lịch sử thay đổi của dự án, giúp đánh dấu các phiên bản hoặc điểm quan trọng trong source code.
  + ***Rules:*** Đặt các quy tắc nhất định cho repository, ví dụ như quy tắc bảo vệ nhánh, yêu cầu kiểm duyệt pull request trước khi merge.
  + ***Actions:*** Nơi thiết lập các workflow automation trong dự án, chẳng hạn như chạy kiểm tra source code tự động mỗi khi có commit mới hoặc xây dựng và triển khai mã.
  + ***Webhooks:*** Cài đặt các hook nhằm tự động thông báo hoặc kích hoạt các hành động bên ngoài khi có sự kiện xảy ra trong repository (ví dụ như khi có pull request, issue mới).
  + ***Codespaces:*** Cho phép thiết lập môi trường phát triển (IDE) ngay trong GitHub, giúp người dùng dễ dàng phát triển dự án mà không cần cài đặt môi trường cục bộ.
  + ***Pages:*** Tạo và quản lý GitHub cho dự án, thường được sử dụng để tạo tài liệu hoặc trang demo của dự án.
  + ***Security:***

*Code security and analysis*: Tính năng bảo mật, giúp phát hiện các lỗi bảo mật hoặc cảnh báo về các thư viện có lỗ hổng.

*Deploy keys*: Thêm các khóa bảo mật để cấp quyền truy cập chỉ đọc hoặc ghi cho repository từ ứng dụng bên ngoài.

## Chức năng và lệnh Git

### Các thuật ngữ

**Branch:**

* Các Branch (nhánh) là các phiên bản tách ra từ project chính, cho phép bạn theo dõi các thay đổi thử nghiệm bạn thực hiện đối với kho lưu trữ và có thể trở lại các phiên bản cũ hơn.

**Commit:**

* Một commit được xem như một checkpoint trong lịch sử dự án. Trong Git bạn có thể khôi phục lại tập tin trong lịch sử commit để chia cho một branch khác.

**Fork:**

* Một fork là một bản sao của một kho lưu trữ (repository). Các lập trình viên thường tận dụng lợi ích của fork để thử nghiệm các thay đổi mà không ảnh hưởng đến dự án chính.

**Head:**

* Các commit ở đầu của một branch được gọi là head. Nó đại diện cho commit mới nhất của repository mà bạn hiện đang làm việc.

**Index:**

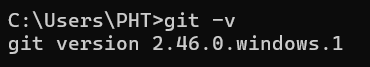
* Khu vực lưu trữ hành động thêm xóa thay đổi file trước khi commit. Nó như là khu vực tổ chức (stagging area) cho Git.

### Các câu lệnh cơ bản:

Người dùng có thể sử dụng qua Git GUI (để quản lý repository) và Git CMD. Trong Git CMD, trước các câu lệnh luôn luôn có từ khóa ‘git’.

**git -v / git –version**

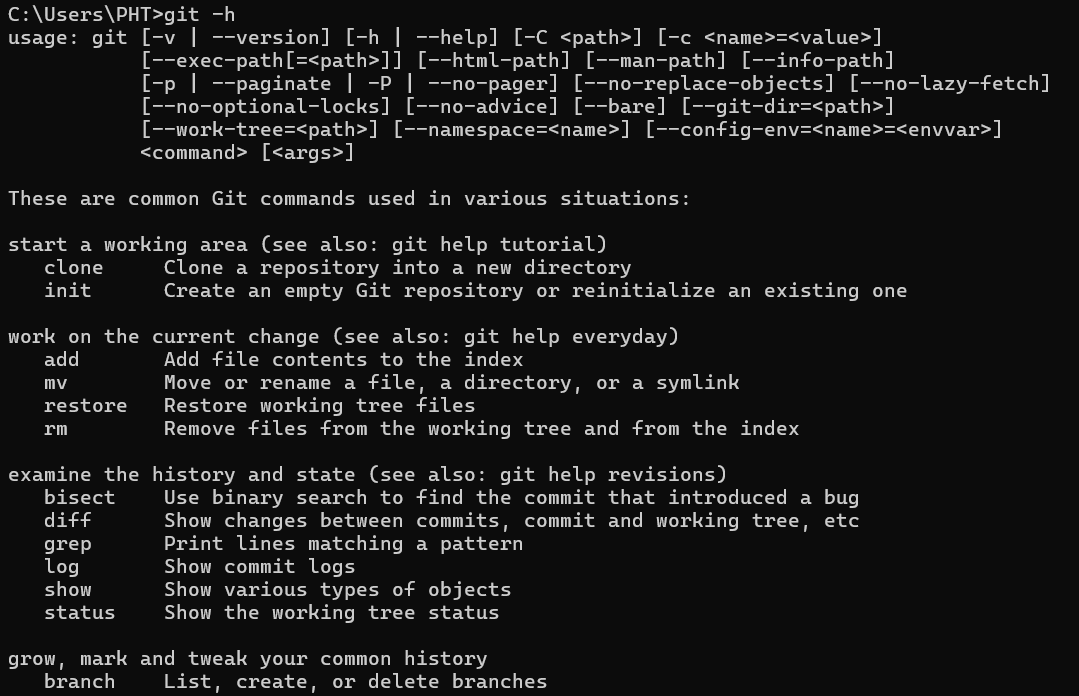
* kiểm tra phiên bản git.



Hình : git -v / git –version

**git -h / git --help**:

* hỗ trợ các câu lệnh

****

Hình : git -h / git --help

**git config**

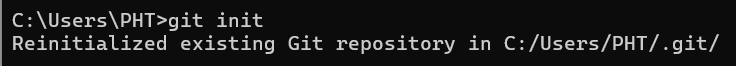
* Chủ yếu dùng để set user.name và user.mail
* git config user.email [mail] | git config user.name [Tên]



Hình : git config

**git init [Tên]**

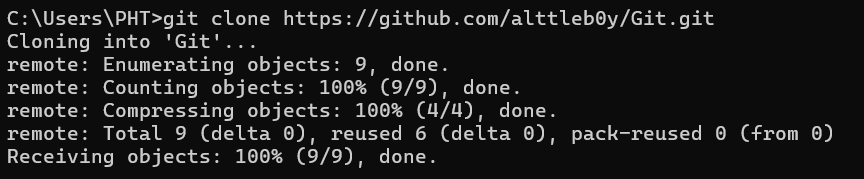
* Khởi tạo 1 git repository 1 project mới hoặc đã có. Nếu không đặt tên thì mặc định sẽ là “.git”.



Hình : git init

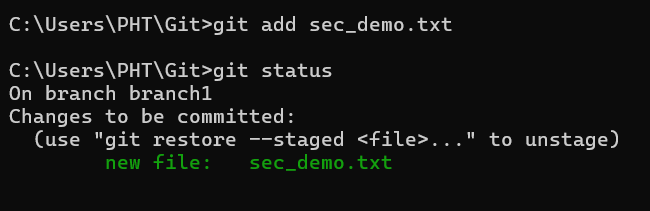
**git clone [URL repository]**

* Tạo bản sao 1 git repository từ remote source.



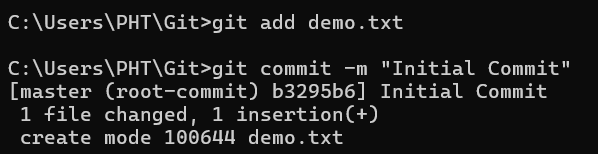
Hình : git clone

**git status**

* Để kiểm tra trạng thái của những file bạn đã thay đổi trong thư mục làm việc kể từ lần commit cuối cùng.

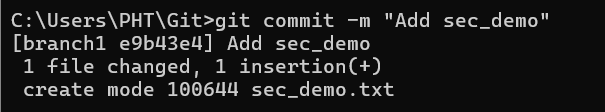
Hình : git status

**git add [Tên file]**

* Thêm file vào thư mục làm việc.

Hình : git add

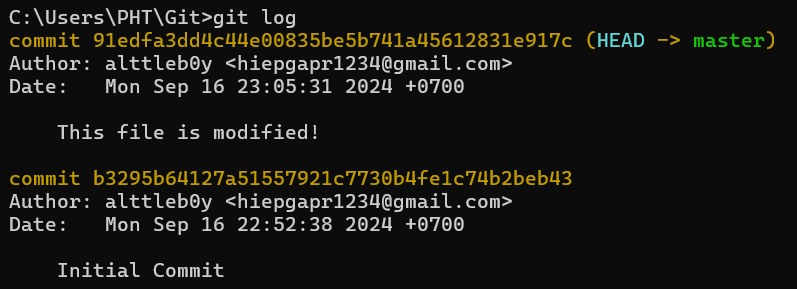
**git commit – m “[Note những thay đổi kể từ lần commit trước]”**

* Lưu lại một snapshot của các thay đổi nằm trong Staging Area. Mỗi lần commit nó sẽ được lưu lại lịch sử chỉnh sửa của code kèm theo tên và địa chỉ email của người commit. 

Hình : git commit

**git log**

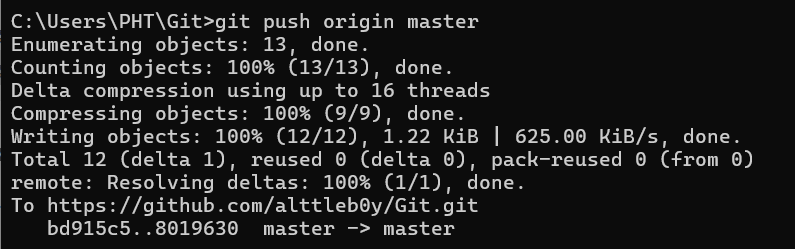
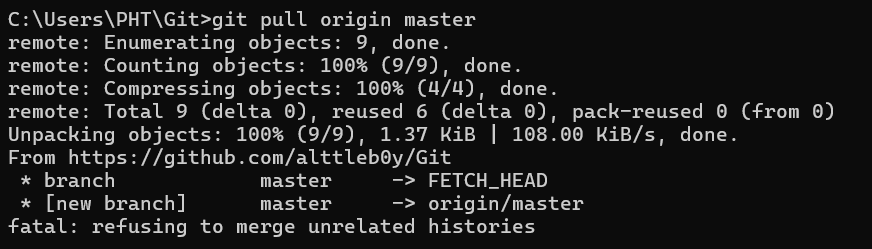
* Xem lại lịch sử commit



Hình : git log

**git push/git pull**

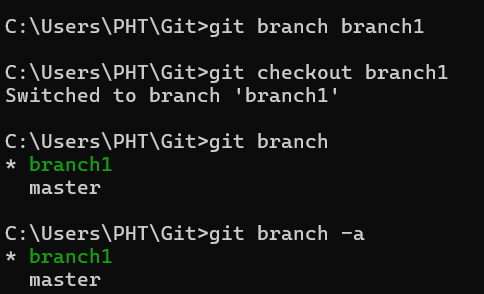
* Tải về hoặc đẩy các commit lên remote.



Hình : git pull

Hình : git push

**git branch**

* Liệt kê tất cả các local branch (nhánh) hoặc có thể dùng git branch -a để liệt kê tất cả các branch.

Hình : git branch

* “git branch [Tên branch]”: tạo branch mới.

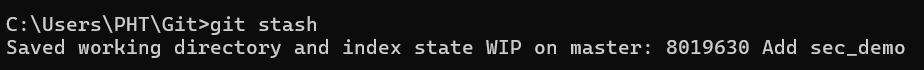
**git checkout [Tên branch]**

* Chuyển sang branch khác

Hình : git checkout

**git stash**

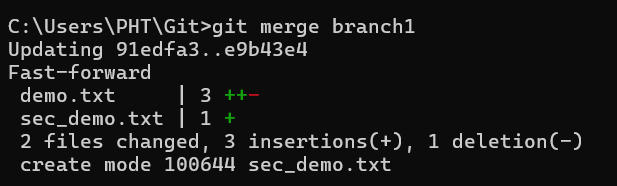
* Lưu thay đổi mà bạn không muốn commit ngay lập tức.



Hình : git stash

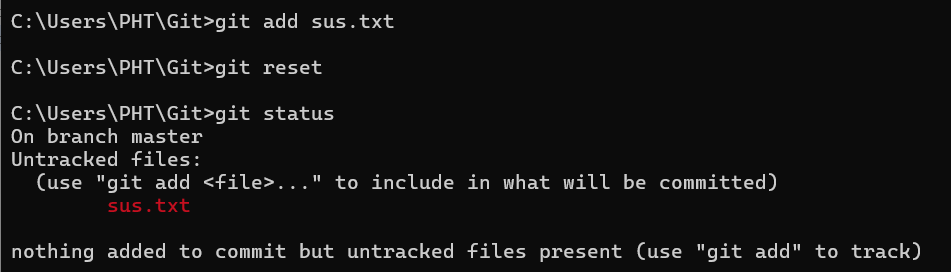
**git merge [Branch muốn merge]**

* Kết hợp 2 branch lại với nhau. Trước khi merge phải checkout ra branch muốn merge.



Hình : git merge

**git reset**

* Loại bỏ file đã add để lần commit tiếp theo không commit file được reset.

Hình : git reset

# CÁC ỨNG DỤNG CỦA GITHUB VÀ GIT

## Xây dựng và quản lý dự án lập trình

* Github và Git được xây dựng để phục vụ cho việc lập trình với nhiều tính năng nổi bật. Với các ‘Tab’ trong dự án phân chia rõ nhiệm vụ và hỗ trợ tương tác thời gian thực.

## Một số ứng dụng khác

* Dùng để theo dõi một chương trình ta nghi là chương trình gây hại bằng cách cho chạy git trong cả ổ đĩa lưu phần mềm đó. Nếu là một chương trình vô hại thì ta sẽ chỉ thấy trong git là phần mềm chỉ tải một vài thứ được tải về hay một số folder mới được tạo. Nhưng nếu ta thấy một số thứ phần mềm đó không nên chạm vào như mở một file không liên quan, thay đổi những thứ mà nó không nên thì từ đó ta xác định được đó là chương trình gây hại.
* Lưu trữ thông tin đăng nhập, tệp save game, tranh vẽ,… Bất kỳ thứ gì mà ta không thể “undo” để trả về trạng thái trước đó.
* Dùng như một database để lưu trữ và theo dõi những dữ liệu mới, thay đổi của chúng, ngày tháng của chúng,.. tương tự như với code.
* Dùng Github như một giải pháp kinh phí thấp để tạo web blog.
* Dùng để so sánh những thay đổi trong cách lập trình qua thời gian.
* Dùng làm nơi làm việc chung cho các dự án bất kỳ không liên quan đến lập trình.