**GIT VÀ GITHUB**

1. **Khái niệm chung**
   1. **Git là gì ?**

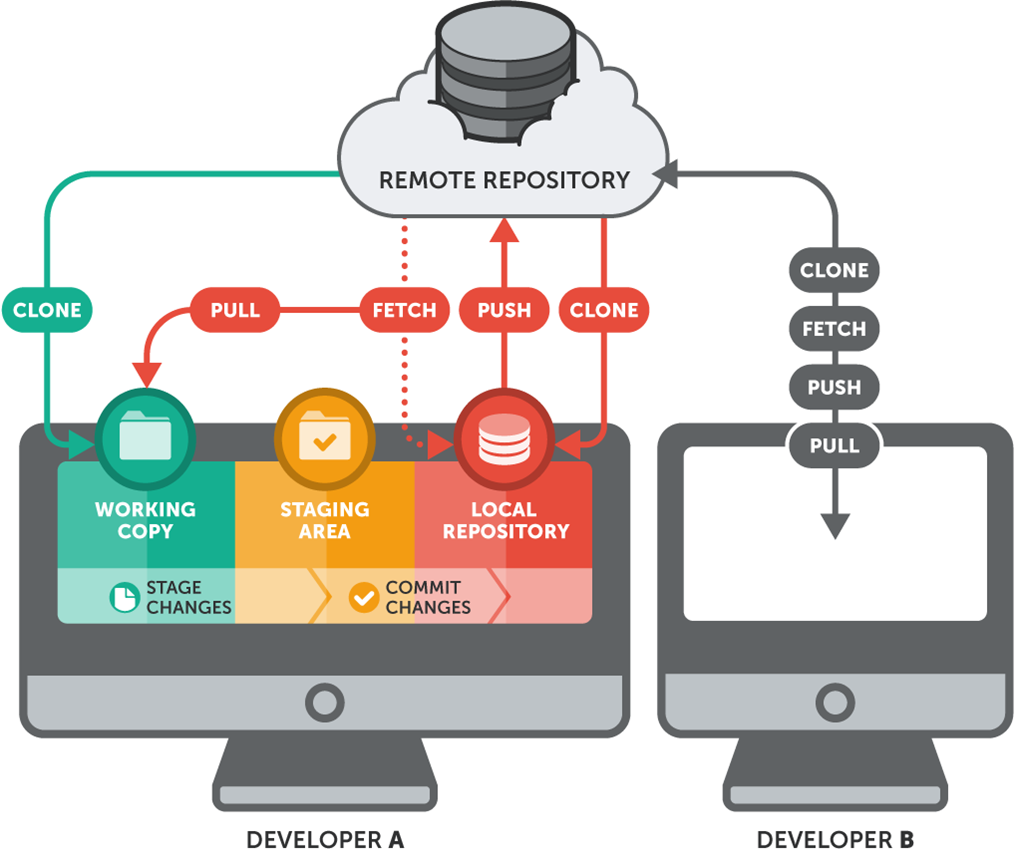
Git là một hệ thống quản lý phiên bản phân tán giúp theo dõi và quản lý các tha đổi của mã nguồn trong quá trình phát triển dự án .Các công dụng chính của Git bao gồm:

* **Quản lý phiên bản**: Lưu trữ lịch sử thay đổi mã nguồn và giúp quay lại phiên bản trước nếu cần.
* **Phân tán**: Mỗi người dùng có một bản sao đầy đủ của kho lưu trữ, làm việc offline và đồng bộ sau.
* **GitHub**: Nền tảng phổ biến sử dụng Git để lưu trữ và chia sẻ dự án.
  1. **Github là gì ?**

GitHub là một nền tảng trực tuyến giúp lưu trữ và quản lý mã nguồn dựa trên Git, cho phép các lập trình viên chia sẻ, hợp tác và phát triển phần mềm cùng nhau (nhóm nhiều người). GitHub cung cấp không gian lưu trữ các kho mã nguồn (repositories) và các công cụ hỗ trợ như quản lý vấn đề (issues), yêu cầu kéo (pull requests).

**Các tính năng chính của github**

* **Lưu trữ mã nguồn**: GitHub cho phép bạn lưu trữ các dự án mã nguồn Git trên đám mây, giúp dễ dàng truy cập và chia sẻ.
* **Hợp tác**: Nhiều lập trình viên có thể làm việc cùng nhau trên một dự án, đồng bộ hóa và hợp nhất các thay đổi.
* **Pull requests**: Công cụ giúp xem xét, thảo luận và chấp nhận thay đổi trước khi đưa vào mã nguồn chính.
* **Issues và Projects**: Quản lý lỗi, yêu cầu tính năng và theo dõi tiến độ phát triển dự án.
* **GitHub Actions**: Tích hợp tự động hóa quy trình phát triển, như kiểm thử, build, và triển khai ứng dụng.



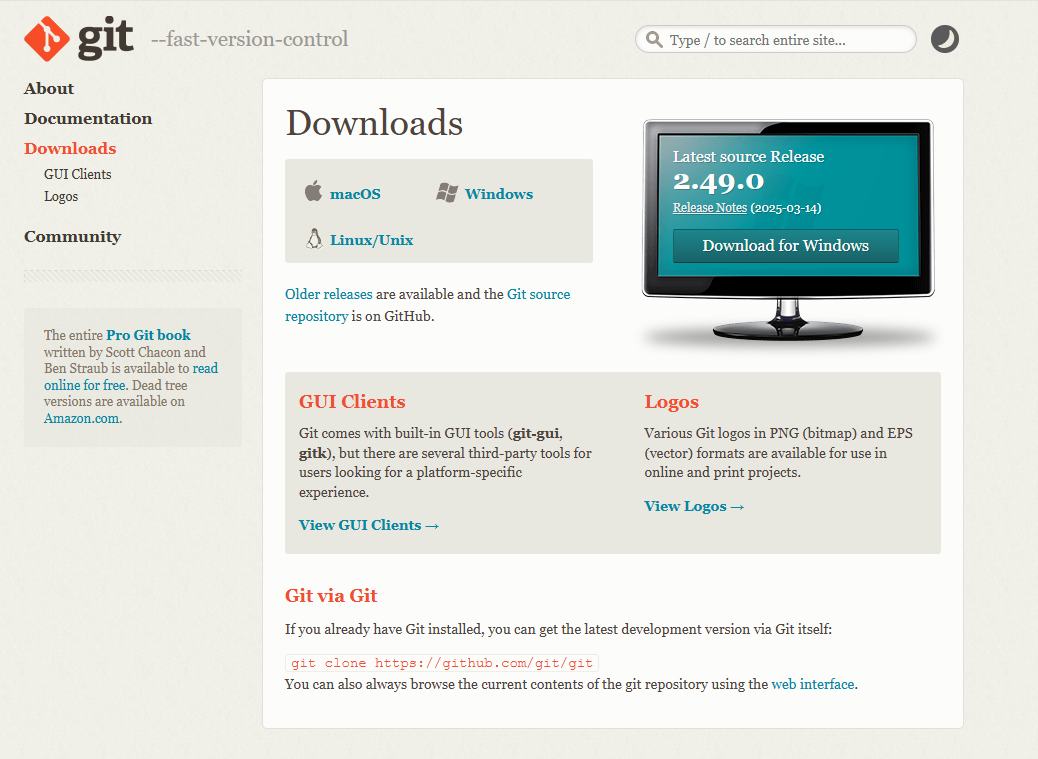
* 1. **Điểm giống và khác nhau giữa Git và Github**

Bảng so sánh Git và GitHub

| **Tiêu chí** | **Git** | **GitHub** |
| --- | --- | --- |
| **Loại công cụ** | Hệ thống quản lý phiên bản phân tán (DVCS - Distributed Version Control System). | Dịch vụ lưu trữ và quản lý kho (repository) Git trực tuyến. |
| **Chức năng chính** | Quản lý mã nguồn, theo dõi lịch sử thay đổi, hỗ trợ làm việc nhóm. | Lưu trữ, chia sẻ mã nguồn, hỗ trợ cộng tác, tích hợp CI/CD, theo dõi issue, pull request. |
| **Làm việc online hay offline?** | Hoạt động hoàn toàn offline. | Cần kết nối Internet để truy cập và quản lý repository. |
| **Cách cài đặt** | Cần cài đặt trên máy tính (CLI hoặc GUI). | Không cần cài đặt, truy cập qua trình duyệt web hoặc dùng GitHub Desktop. |
| **Phạm vi sử dụng** | Cục bộ trên máy tính của người dùng. | Dùng để lưu trữ repository Git trực tuyến, giúp chia sẻ với người khác. |
| **Tính năng chính** | - Theo dõi lịch sử thay đổi mã nguồn. - Hỗ trợ làm việc theo nhóm bằng cách merge code. - Cho phép tạo nhánh (branch) và quản lý version hiệu quả. | - Lưu trữ repository Git trên cloud. - Hỗ trợ issue tracking, pull request, code review. - Tích hợp với nhiều công cụ CI/CD. |
| **Chi phí sử dụng** | Miễn phí, mã nguồn mở. | Có cả gói miễn phí và trả phí (dành cho doanh nghiệp). |
| **Mối quan hệ** | Git là công nghệ nền tảng mà GitHub sử dụng để quản lý mã nguồn. | GitHub là dịch vụ dựa trên Git, giúp lập trình viên lưu trữ và quản lý Git repository trực tuyến. |

**2. Làm việc với Git**  
**2.1 Cách cài đặt và kiểm thử Git**

Git có thể được cài đặt trên nhiều hệ điều hành khác nhau. Dưới đây là cách cài đặt trên Windows, macOS và Linux.  
  
**Cài đặt trên Windows:**  
1. Truy cập <https://git-scm.com/downloads> và tải về phiên bản phù hợp.  
2. Chạy file cài đặt và làm theo hướng dẫn.



**3**. Kiểm tra cài đặt bằng lệnh:  
 git --version

**Cài đặt trên macOS:**  
1. Mở Terminal và chạy lệnh:  
 brew install git  
2. Kiểm tra cài đặt:  
 git --version  
  
**Cài đặt trên Linux (Ubuntu/Debian):**  
 sudo apt update  
 sudo apt install git  
 git --version  
  
Sau khi cài đặt, bạn cần cấu hình Git bằng các lệnh sau:  
 git config --global user.name "Nhut Thien"  
 git config --global user.email "your-email@example.com"  
 git config --global --list # Kiểm tra cấu hình Git

**2.2 Các lệnh cơ bản trong Git**  
Dưới đây là các lệnh Git cơ bản được chia theo từng nhóm chức năng.  
  
***1. Cấu hình Git***  
Cấu hình thông tin người dùng:  
 git config --global user.name "Nhut Thien"  
 git config --global user.email "your-email@example.com"  
 git config --global --list # Kiểm tra cấu hình Git hiện tại

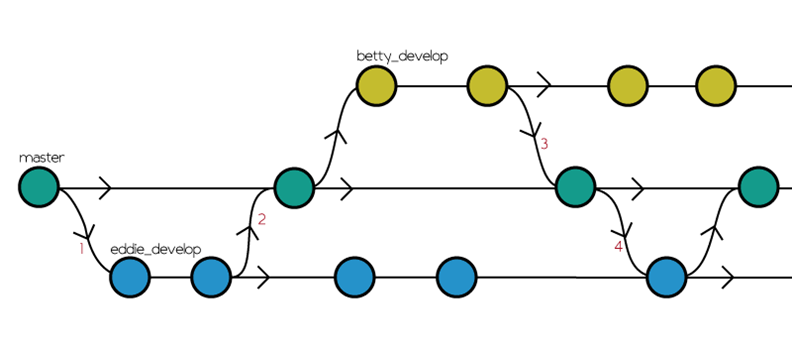
***2. Khởi tạo hoặc sao chép Repository*** git init # Khởi tạo một repository Git mới  
 git clone <URL> # Clone một repository có sẵn từ GitHub

***3. Kiểm tra trạng thái và lịch sử***  
 git status # Kiểm tra trạng thái repository (file nào thay đổi, chưa commit)  
 git log # Xem lịch sử commit  
 git log --oneline # Xem lịch sử commit dạng ngắn gọn

git log -3 # xem chỉ 3 lần commit gần nhất

***4. Làm việc với file***  
 git add <tên\_file> # Thêm một file vào staging  
 git add . # Thêm tất cả các file đã chỉnh sửa  
 git commit -m "Mô tả commit" # Lưu thay đổi vào Git  
 git rm <tên\_file> # Xóa file khỏi Git và thư mục  
 git rm --cached <tên\_file> # Xóa file khỏi Git nhưng giữ lại file trên máy

***5. Làm việc với nhánh (Branch)***  
 git branch <tên\_nhánh> # Tạo nhánh mới  
 git checkout <tên\_nhánh> # Chuyển sang nhánh khác  
 git checkout -b <tên\_nhánh> # Tạo và chuyển sang nhánh mới  
 git merge <tên\_nhánh> # Gộp nhánh vào main



***6. Làm việc với GitHub*** git remote add origin <URL> # Kết nối repository với GitHub  
 git push -u origin main # Đẩy code lên GitHub  
 git pull origin main # Lấy code mới nhất về máy

***7. Hoàn tác thay đổi*** git checkout -- <tên\_file> # Khôi phục file về trạng thái gần nhất  
 git reset --soft HEAD~1 # Hủy commit gần nhất nhưng giữ lại thay đổi  
 git reset --hard HEAD~1 # Hủy commit và xóa luôn thay đổi

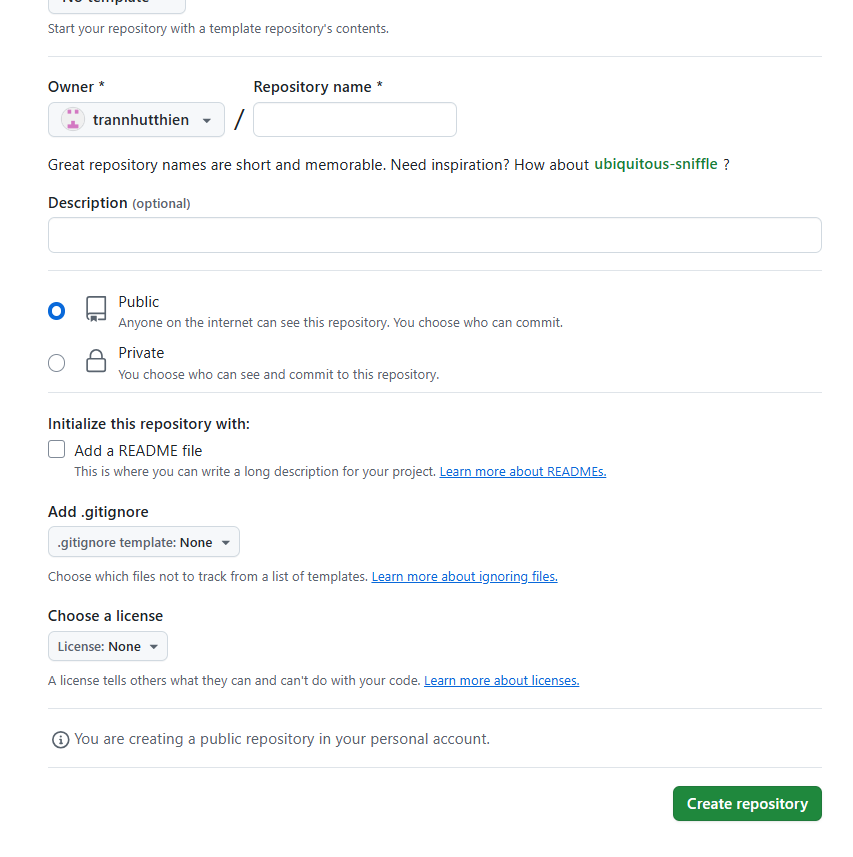
***8. Các lệnh bổ sung hữu ích***  
 git stash # Lưu tạm thời thay đổi chưa commit  
 git stash pop # Khôi phục lại thay đổi đã stash  
 git diff # Xem sự khác biệt giữa phiên bản cũ và mới  
 git reflog # Xem lịch sử thay đổi kể cả commit đã reset

*Tóm lại, Git giúp bạn theo dõi, quản lý và làm việc nhóm hiệu quả. GitHub hỗ trợ lưu trữ, chia sẻ và làm việc nhóm trên môi trường trực tuyến.*

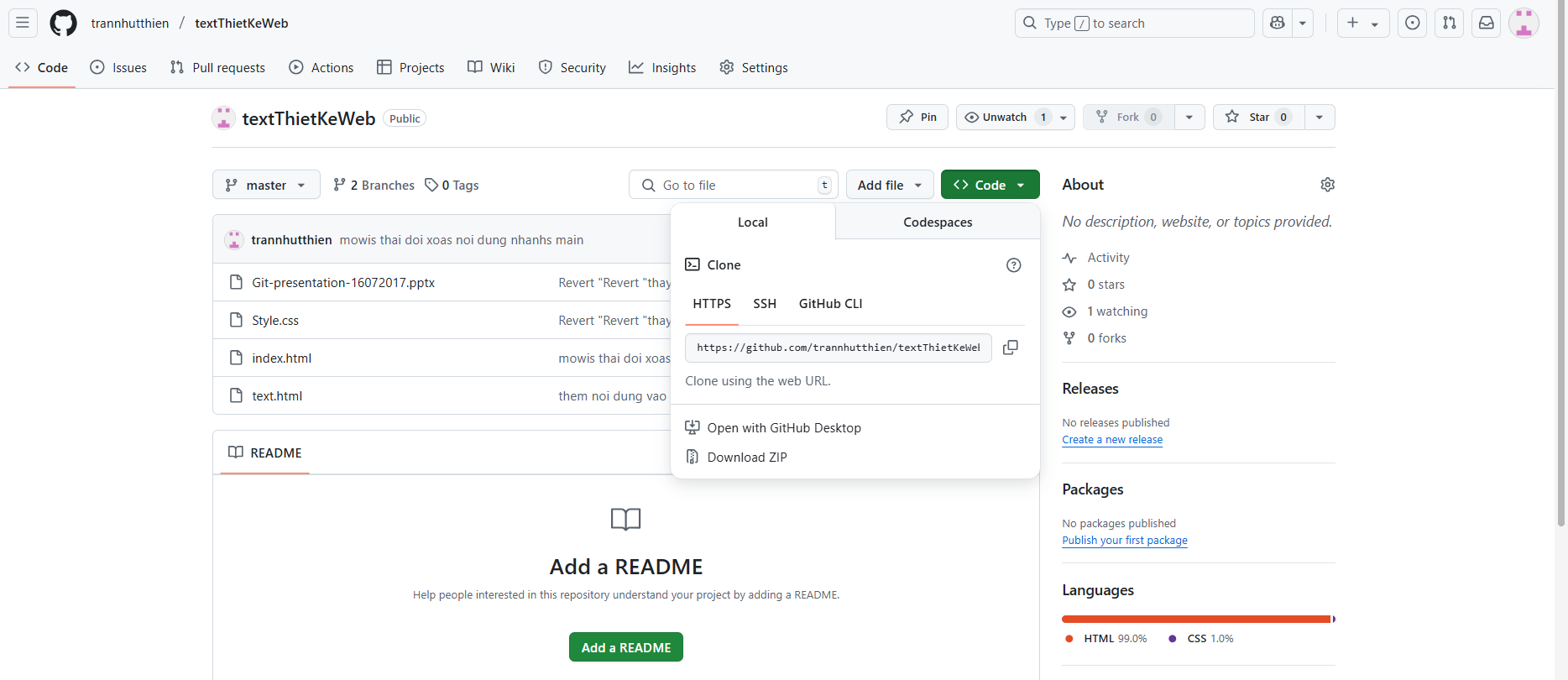
**3. Làm việc với GitHub**  
 **3.1 Tạo dự án trên GitHub**

Để tạo một repository (dự án) trên GitHub, thực hiện các bước sau:

**1**. Truy cập <https://github.com/> và đăng nhập vào tài khoản.  
**2**. Nhấn vào biểu tượng "+" ở góc phải và chọn **New repository**.  
**3**. Nhập tên repository, chọn chế độ **Public** hoặc **Private**.  
**4**. Chọn Initialize this repository with a README nếu muốn có file README.  
**5**. Nhấn **Create repository** để hoàn tất.

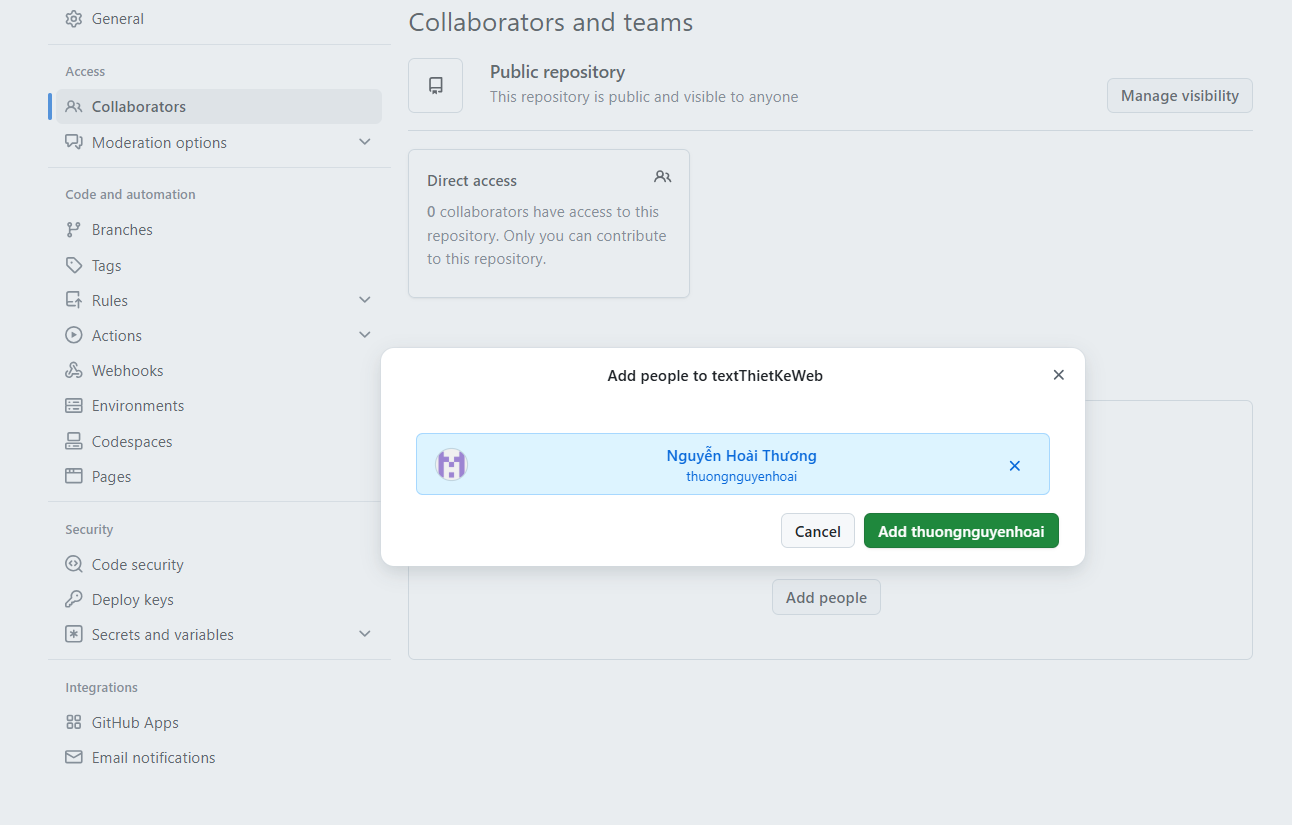


**6. Sao chép link repository và sử dụng lệnh:**   
 git remote add origin <URL>  
 git push -u origin main  
   
để kết nối repository cục bộ với GitHub.



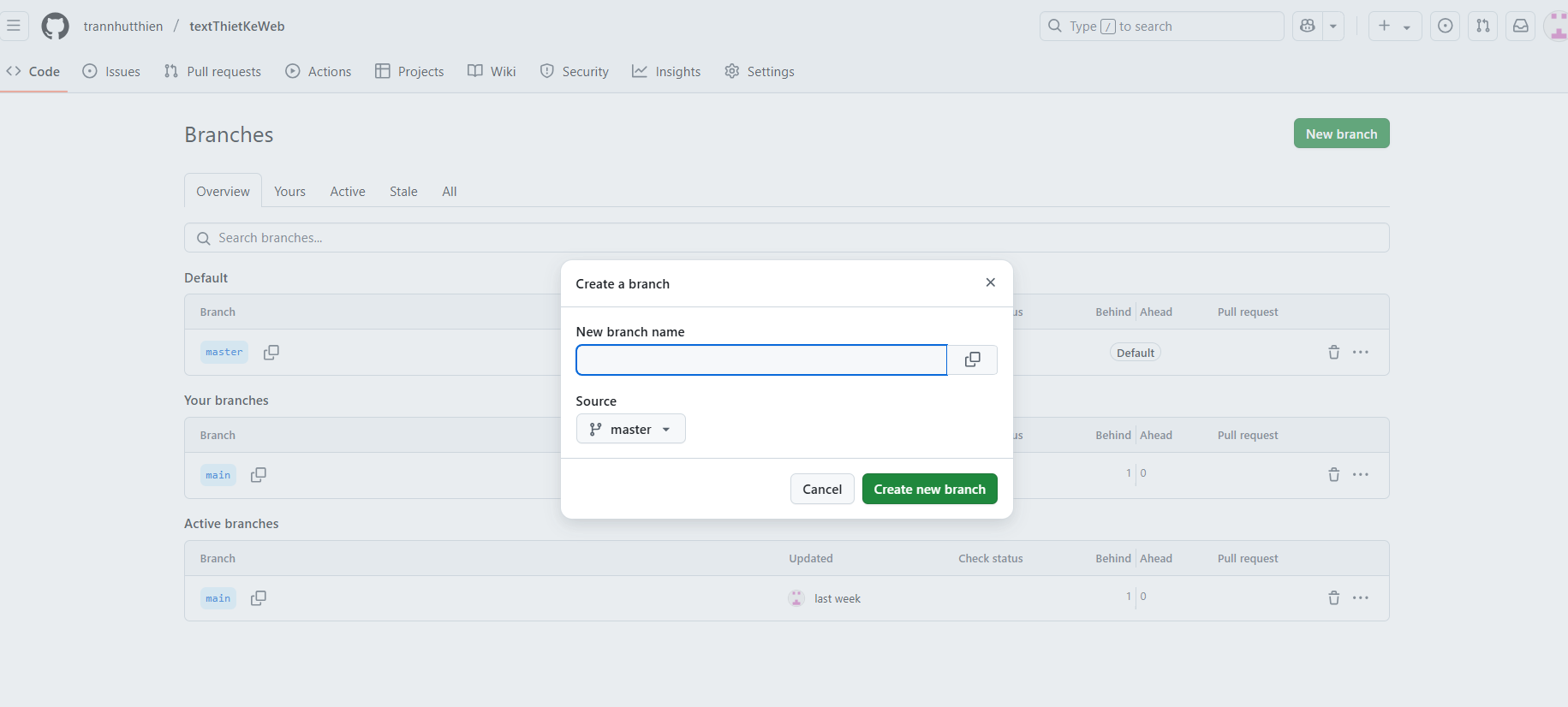
**3.2 Thêm cộng tác viên**

Để thêm cộng tác viên vào repository trên GitHub:  
  
**1**. Mở repository trên GitHub.  
**2**. Vào **Settings** → **Collaborators**.  
**3**. Nhấn **Add people** , nhập tên tài khoản GitHub của người cần thêm.  
**4**. Chọn quyền truy cập (Read, Write, Admin) và nhấn **Add collaborator** .  
**5**. Người được mời sẽ nhận email và cần chấp nhận lời mời.



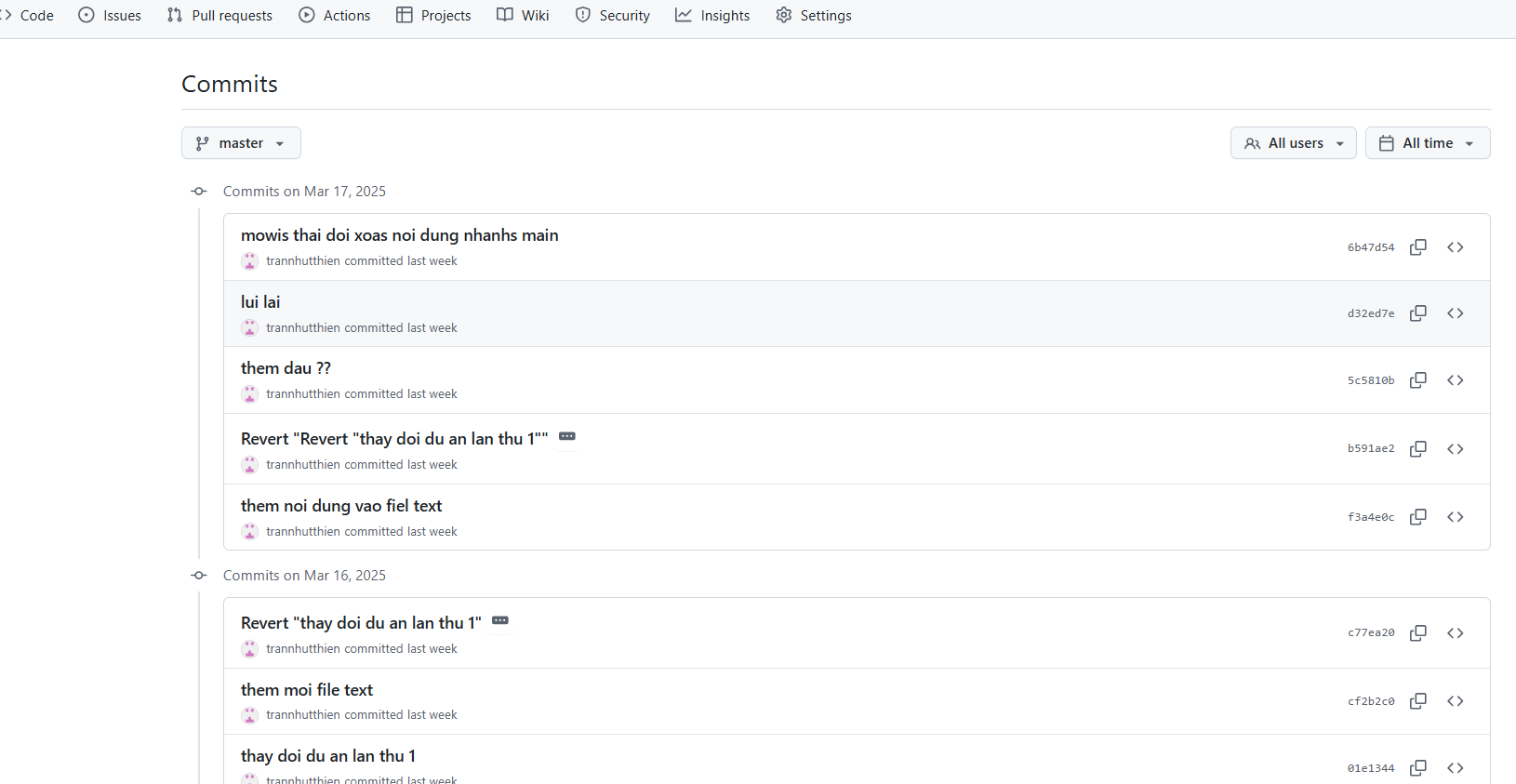
**3.3 Tạo nhánh và quản lý nhánh**

GitHub cho phép tạo và quản lý nhánh để phát triển tính năng riêng biệt:  
Tạo nhánh trực tiếp trên web Github:



**3.4 Xem thay đổi code và quản lý commit**

Để theo dõi thay đổi và quản lý commit trên GitHub:  
  
**Xem sự khác biệt giữa hai commit**.  
 git diff <commit1> <commit2>  
   
  
**Xem lịch sử commit trên GitHub:**  
 1. Mở repository trên GitHub.  
 2. Vào tab **Commits** để xem danh sách commit.



**Tạo Pull Request (PR) để merge code**  
 1. Chuyển đến tab **Pull Requests**.  
 2. Nhấn **New pull request**.  
 3. Chọn nhánh muốn merge, nhập mô tả.  
 4. Nhấn **Create pull request**

