

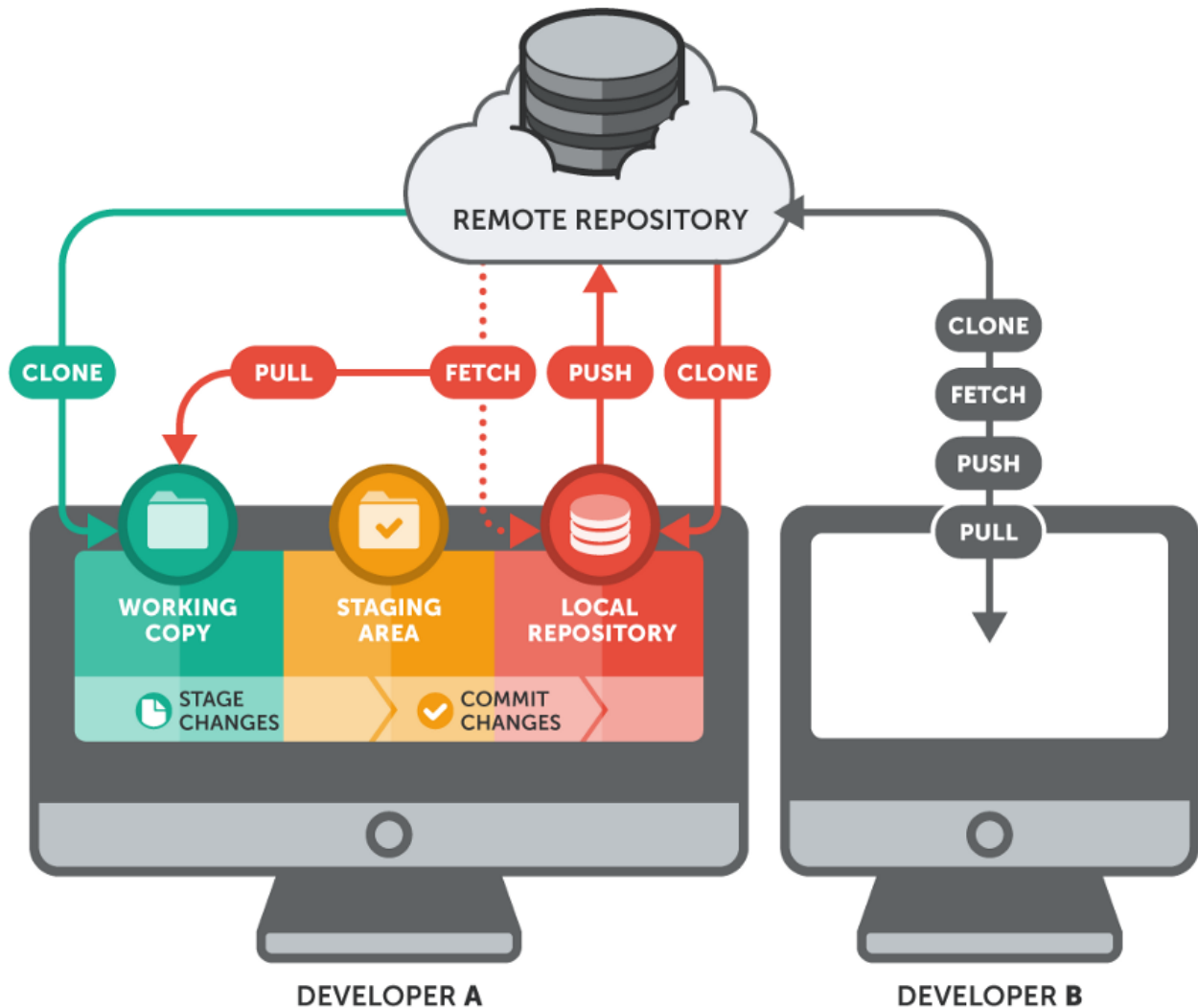
## Bài tập Lab: Quản lý source code và làm việc nhóm với Git

\*\*\*\*\*

### 1. Thông tin về GIT

- Là công cụ giúp quản lý source code được tổ chức theo dạng dữ liệu phân tán (distributed)
- Giúp đồng bộ source code của nhóm làm việc chung với nhau lên 1 server (remote)
- Hỗ trợ các thao tác kiểm tra source code của dev trong quá trình làm việc (Diff, Check modification, ShowLog, ...)

### 2. Sử dụng GIT vào dự án như thế nào?



Mô hình hoạt động của GIT cần nắm một số khái niệm:

- Repository(repo): là nơi chứa source code.
- Remote: server online.
- Local: server local trên máy.
- Working copy: nơi làm việc trực tiếp trên máy local.

Chúng ta tổ quản lý source code trên máy sẽ được lưu trữ trong local repo, khi có internet chúng ta có thể “đồng bộ” local repo lên online, tức là sync lên remote repo.

Khi trong team hoàn thành các tính năng và đã sync lên cho server online(remote repo) thông qua các thao tác (commit & push).

Thao tác cơ bản của 1 dev khi hoàn thành 1 tính năng:

- **Commit:** Lưu các thay đổi từ thư mục đang làm việc(working copy) vào server local (local repo)
- **Push:** Cập nhật các thay đổi từ server local ở máy(local repo) lên server online (remote repo)

Các thành viên trong team cần lấy về sử dụng cần làm các thao tác:

- **Fetch:** Để “đồng bộ” mọi thứ thay đổi mới nhất từ server online(remote repo)về server local(local repo)chúng ta sử dụng lệnh fetch(git fetch)
- **Pull:** Sau đó để”áp dụng” các thay đổi đó vào source code đang sử dụng. Tức là cập nhật từ server local(local repo) vào thư mục đang làm việc(working copy) (git pull)

### 3. Các thao tác sử dụng cho GIT

#### Lệnh

#### Chức năng

#### Clone

- Lấy source từ server về server local của máy (local repo).

#### Pull

- Cập nhật các thay đổi từ server local lên server online (remote repo)

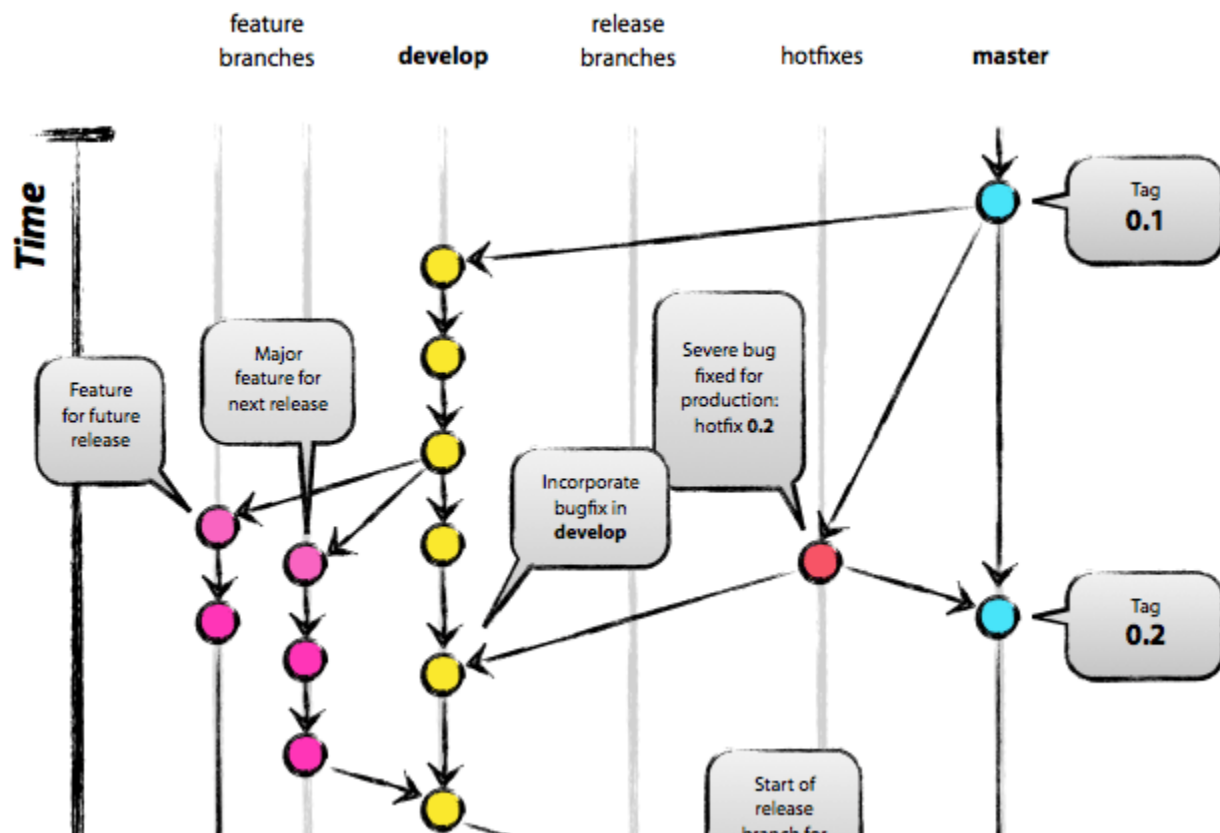
#### Checkout

- Đổi từ branch này sang branch khác.
- Cập nhật tới 1 reversion (Mã Hash)

### 4.Sử dụng GIT vào triển khai dự án như thế nào?

Để sử dụng cho quá trình triển khai dự án, chúng ta sử dụng khái niệm branch của GIT.

**Branch:** được hiểu là 1 nhánh của GIT, được dev tách ra khi cần thiết. Mục đích để đảm bảo độ ổn định của source code tại 1 thời điểm và tránh việc sử dụng các code không cần thiết vào



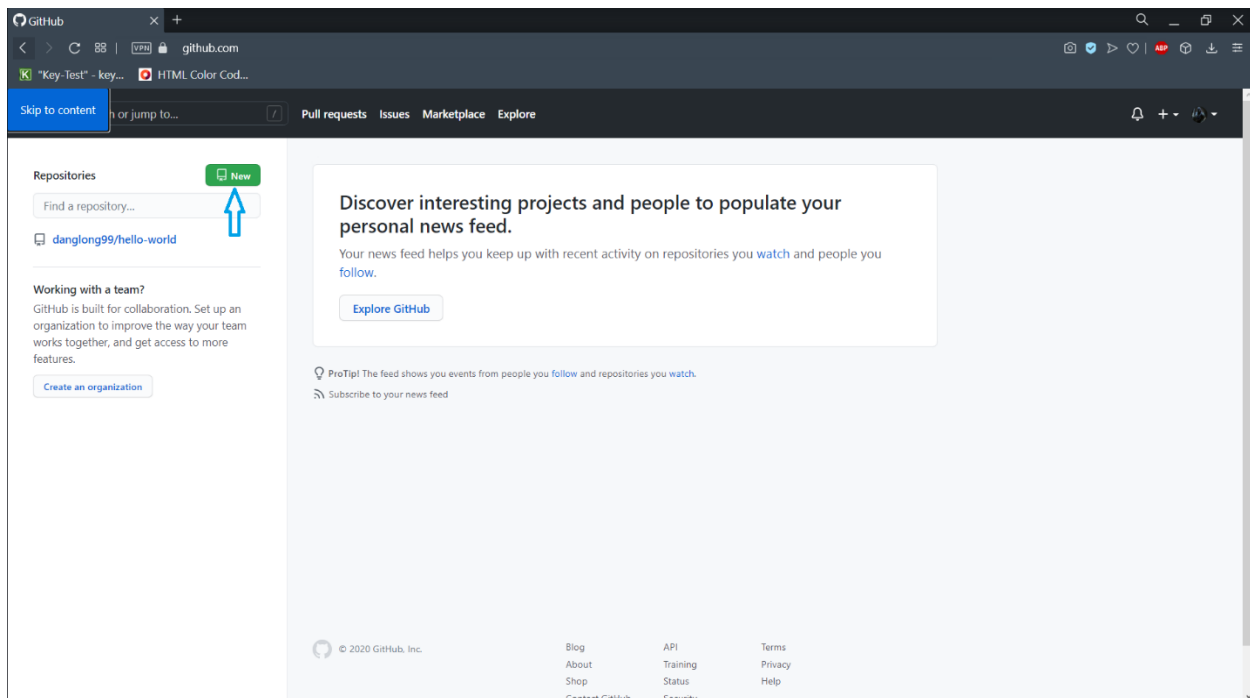
Đối với GIT, mỗi repo đều thuộc 1 nhánh nhất định (mặc định sẽ có nhánh: master remote)

### **Yêu cầu cài đặt:**

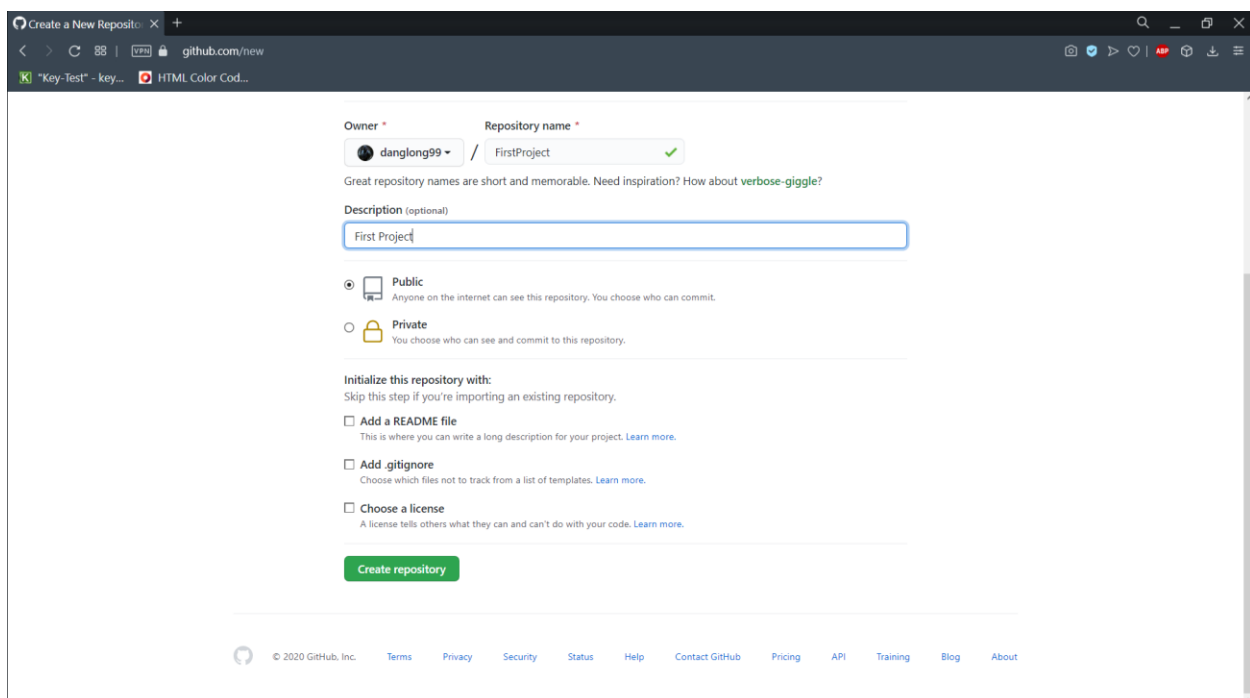
- Github Desktop
- Git Bash
- TortoiseGit

+) Bước 1: Tạo dự án với GITHUB

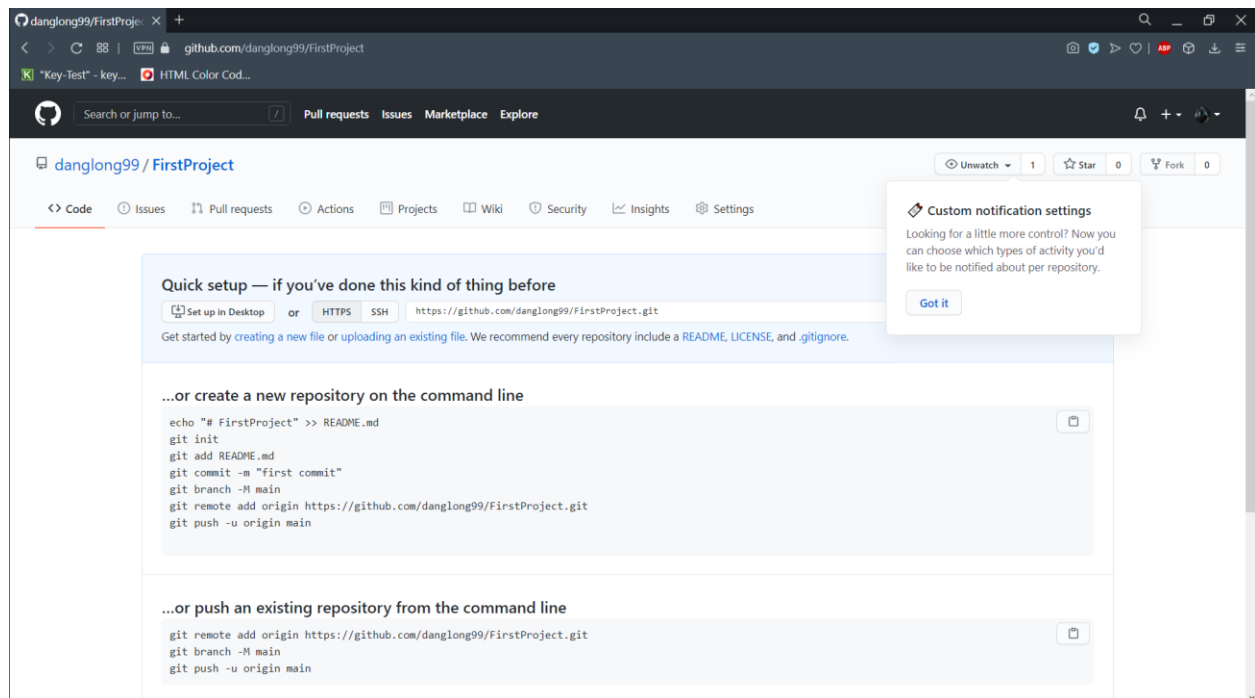
Tạo tài khoản và đăng nhập vào github: <https://github.com>



Tạo dự án và đặt tên mô tả cho dự án



Chọn Create repository, sẽ được 1 repo remote để chứa dự án



Link GIT của dự án sẽ được tạo theo tên của repo lúc đầu tạo dự án

VD: <https://github.com/danglong99/FirstProject.git>

Chúng ta có thể lấy về máy local để bắt đầu code.

GIT đã có sẵn một số câu lệnh cơ bản như hình:

```
MINGW64~/c/Users/Long
Long@DANGLONG MINGW64 ~ (master)
$ echo "#FirstProject" >> README.md

Long@DANGLONG MINGW64 ~ (master)
$ git init
Reinitialized existing Git repository in C:/Users/Long/.git/

Long@DANGLONG MINGW64 ~ (master)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

Long@DANGLONG MINGW64 ~ (master)
$ git commit -m "first commit"
[master (root-commit) d4ecaf5] first commit
1 file changed, 4 insertions(+)
create mode 100644 README.md

Long@DANGLONG MINGW64 ~ (master)
$ git branch -M main

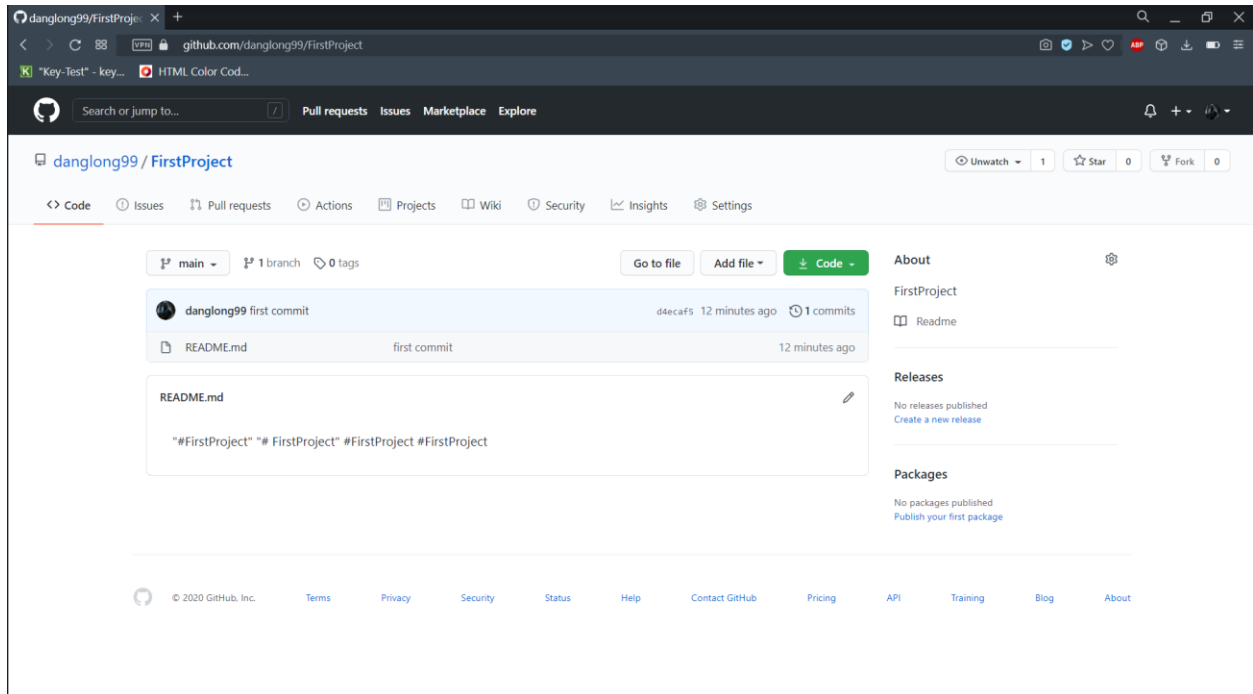
Long@DANGLONG MINGW64 ~ (main)
$ git remote add origin https://github.com/danglong99/FirstProject.git

Long@DANGLONG MINGW64 ~ (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 265 bytes | 132.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/danglong99/FirstProject.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

## Sử dụng GIT cho thư mục mới

Ví dụ, tài khoản của mình là danglong99 và mình có 1 repo là FirstProject

- Lệnh khởi tạo 1 file README.md với nội dung #FirstProject:  
echo "#FirstProject">> README.md
- Lệnh khởi tạo sử dụng GIT cho thư mục hiện tại:  
git init
- Thêm file README.md vào cho GIT quản lý(working copy):  
git add README.md
- Thêm vào server local (repo local):  
git commit -m "first commit"
- Kết nối server local vào server online:  
git remote add origin https://github.com/danglong99/FirstProject.git
- Kiểm tra lại log trên web của Github và file vừa commit:



## Sử dụng GIT cho thư mục đã có:

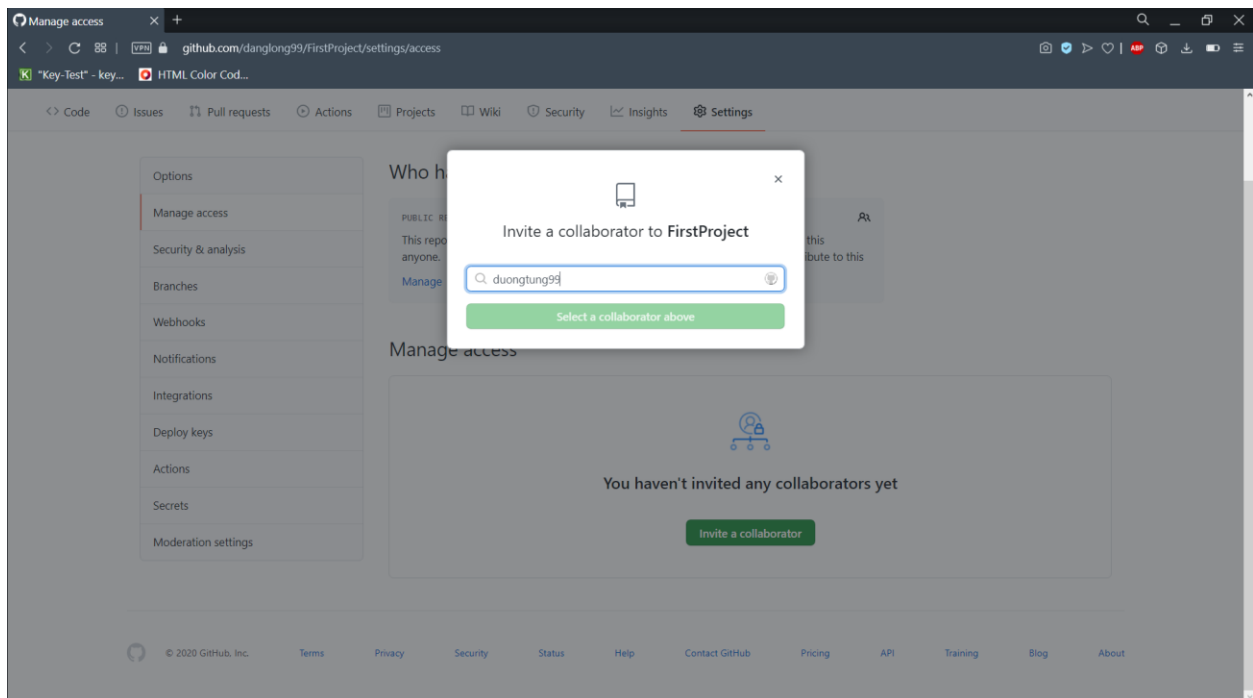
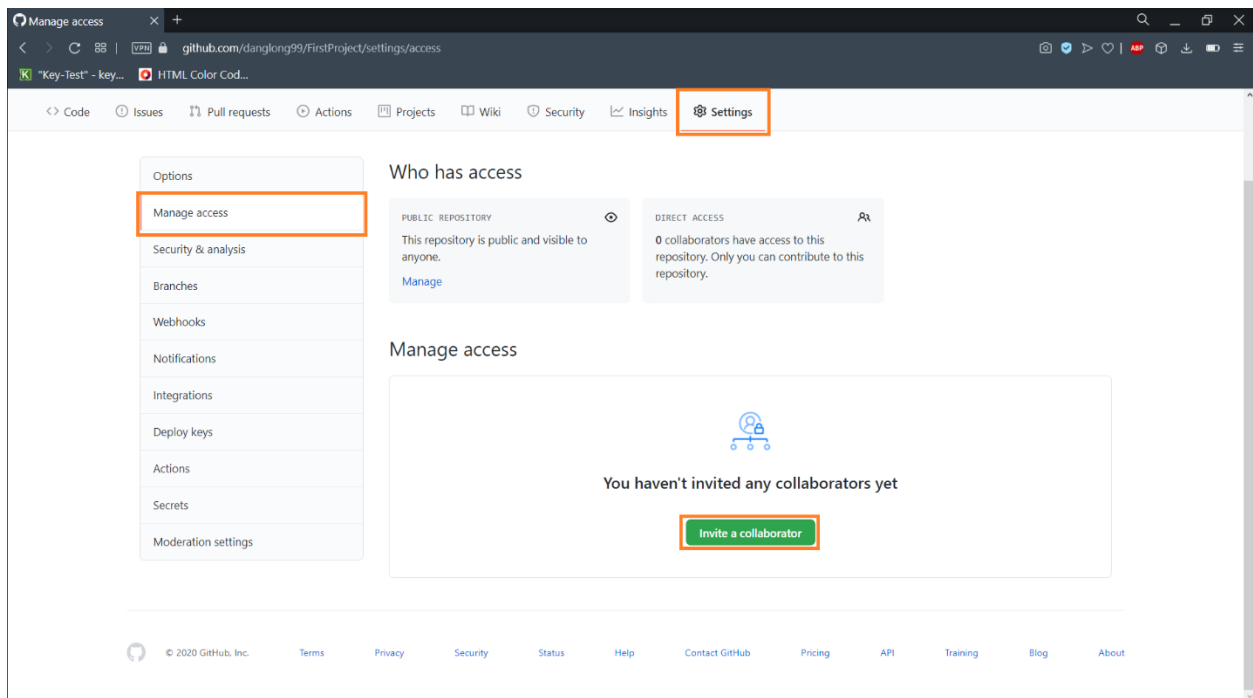
git remote add origin

<https://github.com/danglong99/FirstProject>

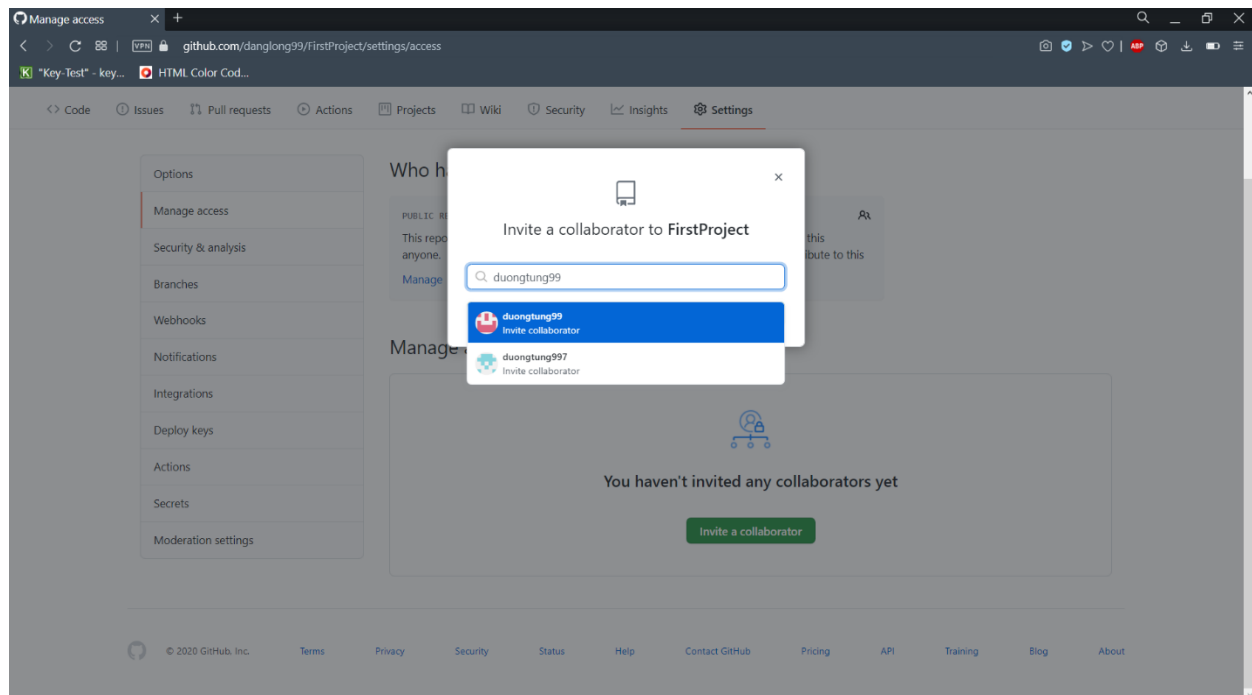
git push -u origin master

+) Bước 2: Thêm thành viên vào team để làm việc nhóm

Để thêm thành viên, vào Setting như hình và nhập vào email hoặc username Github







### +) Bước 3: Commit

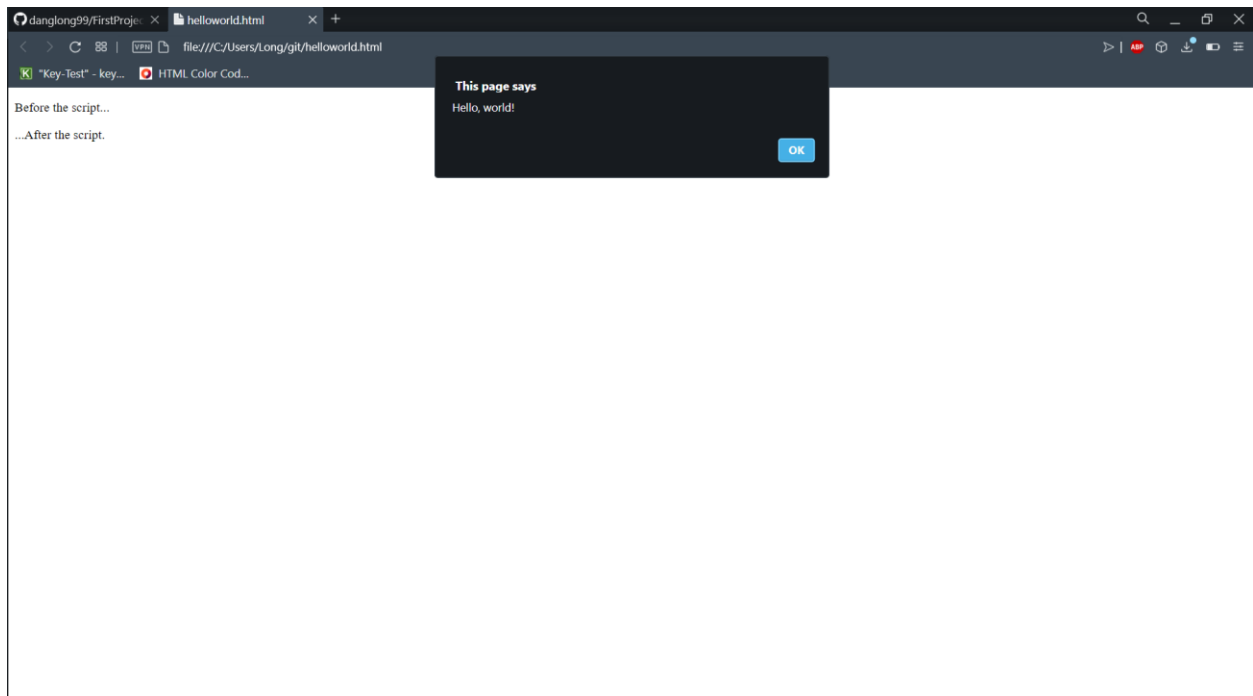
Làm việc với GIT, sau khi đã clone source về (Thư mục đã hiểu GIT)

Để đẩy code sau khi hoàn thành, chúng ta sử dụng thao tác commit

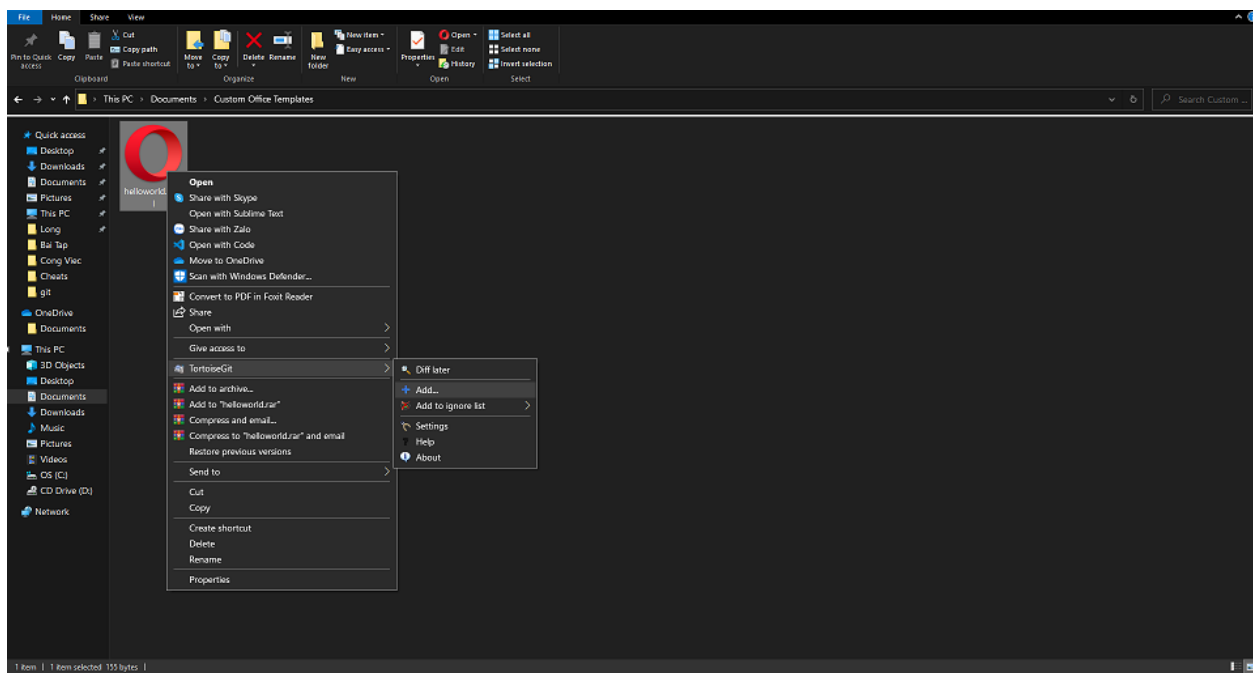
Chúng ta có ví dụ Helloworld với Javascript. Xuất ra màn hình web thông báo nội dung “Hello World”

Code:

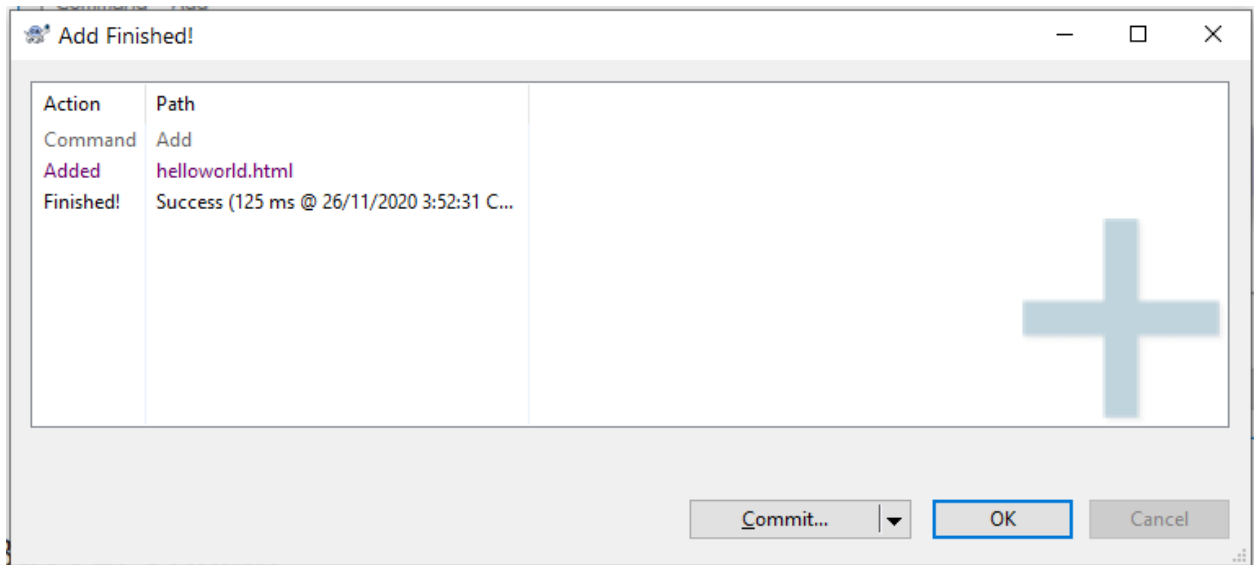
```
<!DOCTYPE HTML>
<html>
<body>
<p>Before the script...</p>
<script>
alert( 'Hello, world!' );
</script>
<p>...After the script.</p>
</body>
</html>
```



## 1) Chuột phải vào file, trở đến TortoiseGit chọn Add



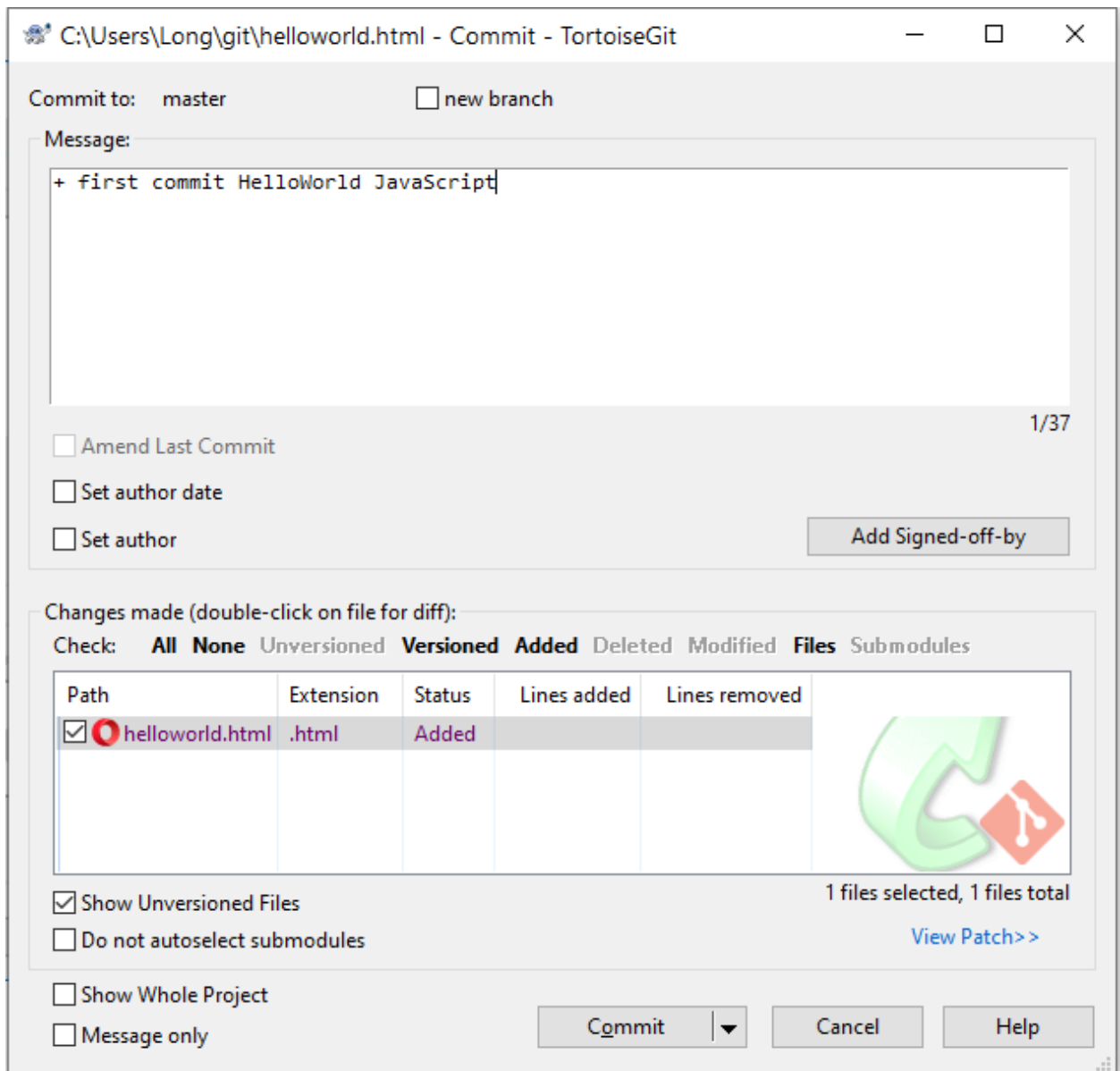
Thêm thành công file vào GIT



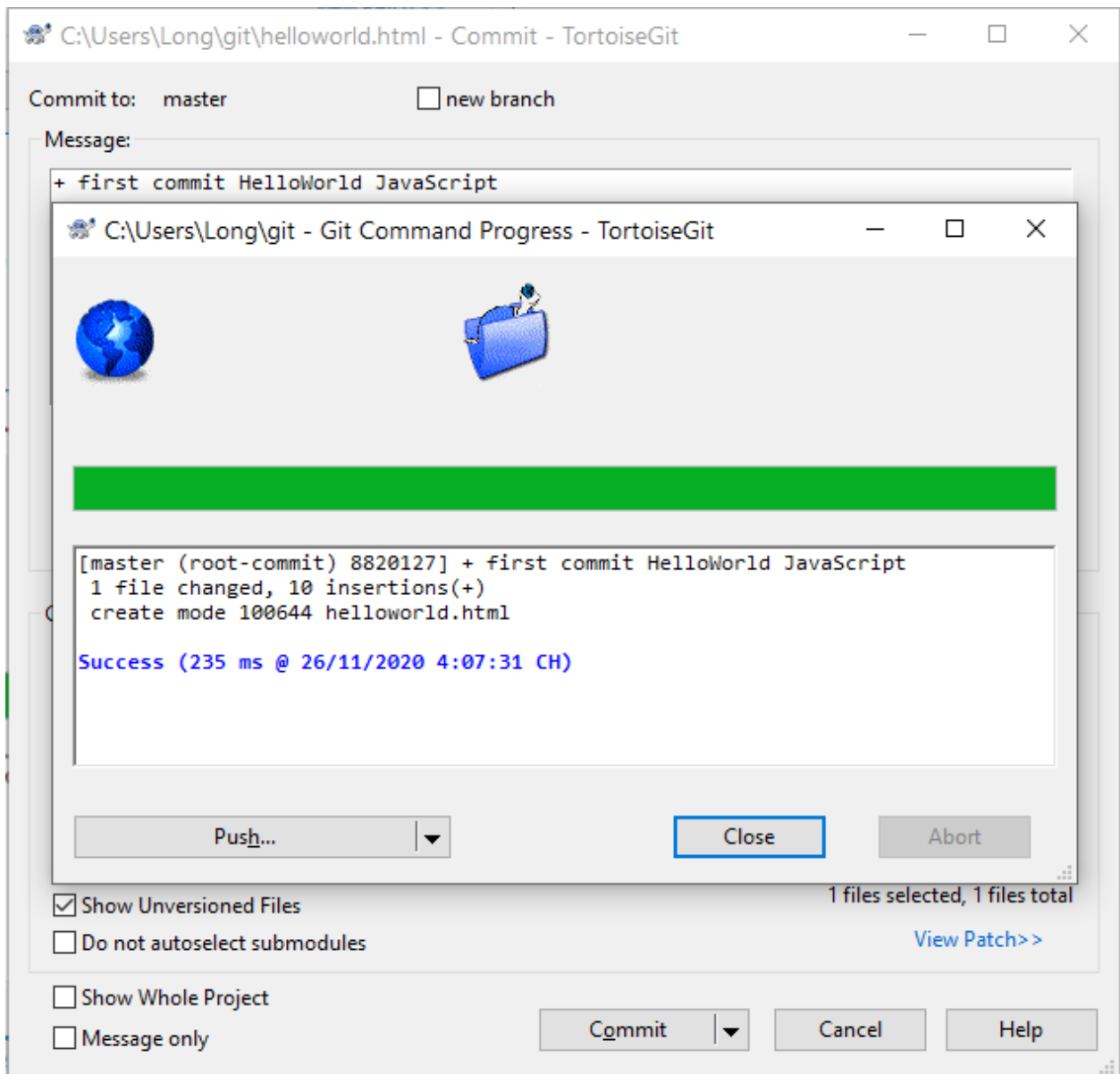
## 2) Commit

Chuột phải vào thư mục chọn Commit

## 3) Nhập log để commit



4) Chọn Push để lưu lên server online (repo remote)



Chọn branch để push (mặc định sẽ chọn master)

C:\Users\Long\git - Push - TortoiseGit

Ref

☐ Push all branches

Local: master

Remote: master

Destination

☒ Remote: origin

☐ Arbitrary URL:

Options

Force: May discard ☐ known changes ☐ unknown changes

☐ Include tags

☒ Autoload Putty Key

☐ Set upstream/track remote branch

☐ Always push to the selected remote archive for this local branch

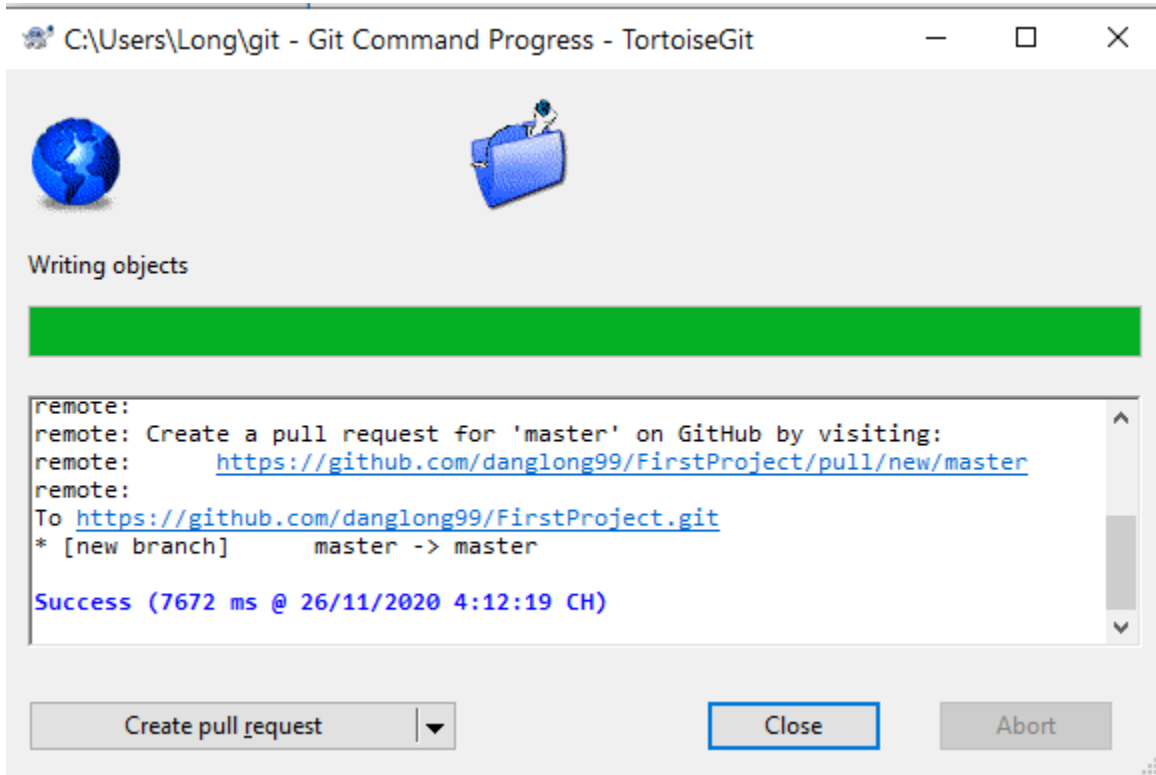
☐ Always push to the selected remote branch for this local branch

Recurse submodule: None

Push option:

OK Cancel Help

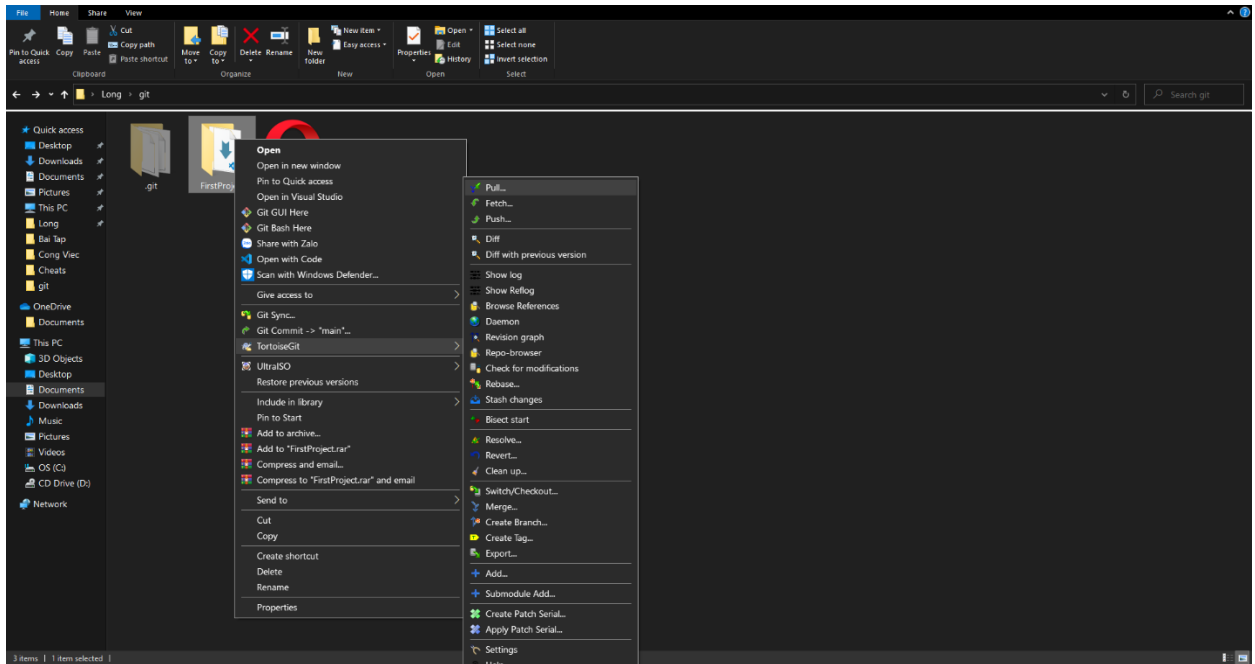
Push thành công



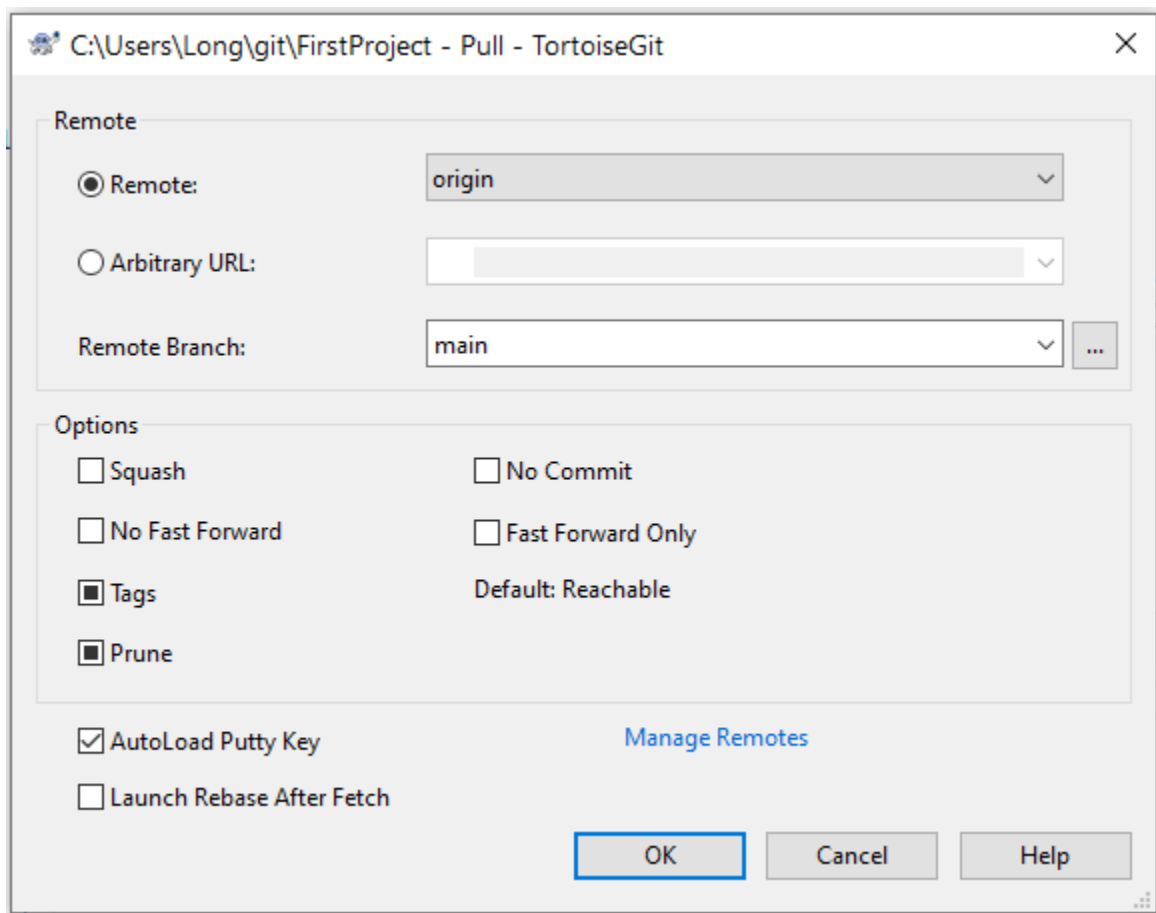
+) Bước 4: Pull

Để lấy những thay đổi mới nhất team đã commit lên server online, chúng ta sử dụng lệnh pull

Chuột phải vào thư mục chọn Pull để cập nhật thay đổi mới nhất từ server

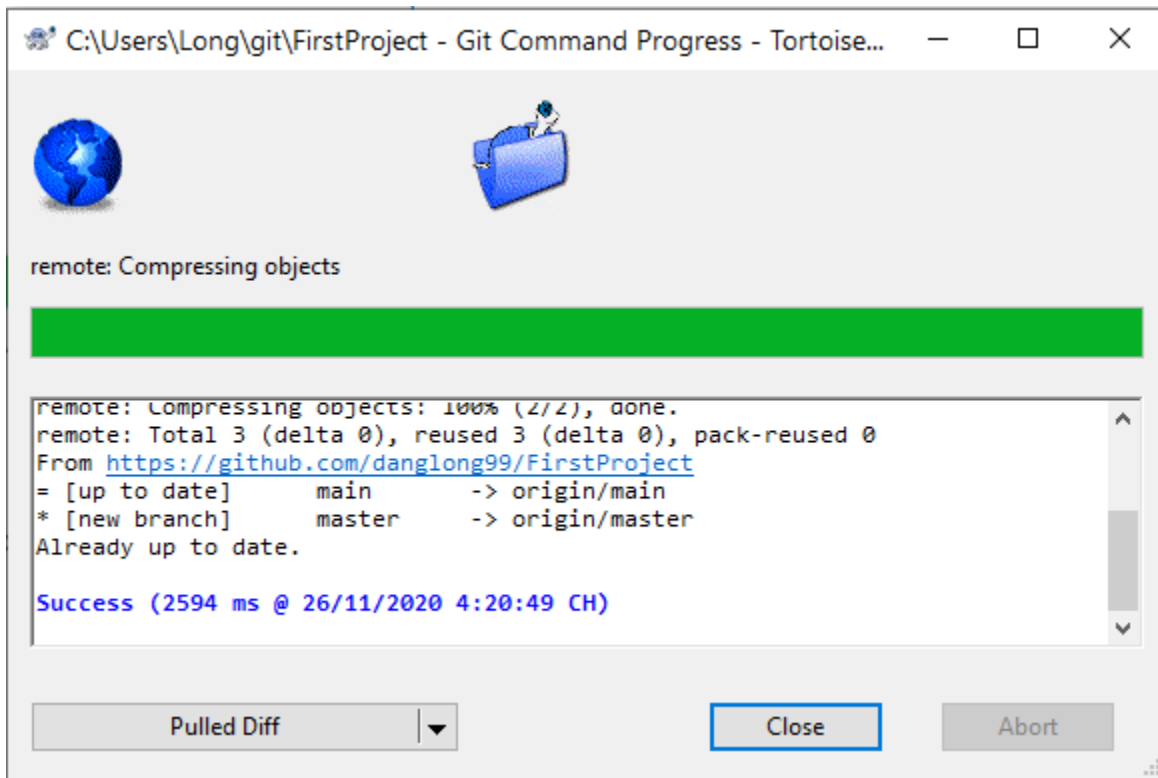


Chọn remote để lấy những thay đổi mới nhất về



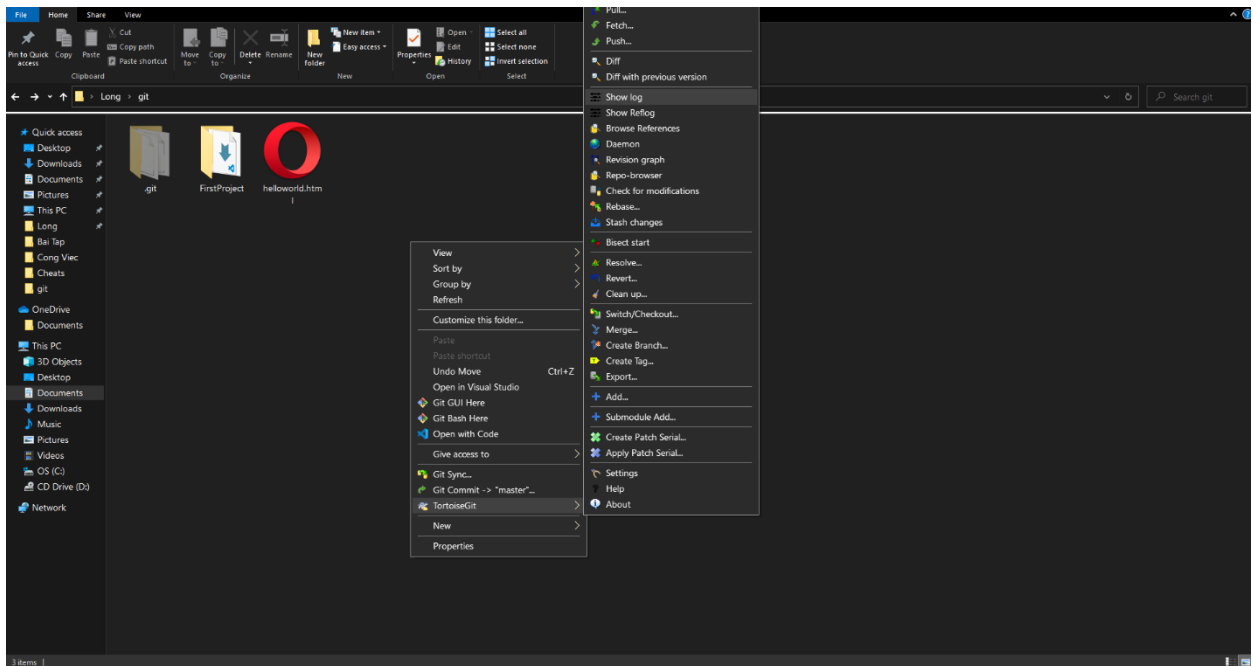


Các thay đổi được cập nhật thành công

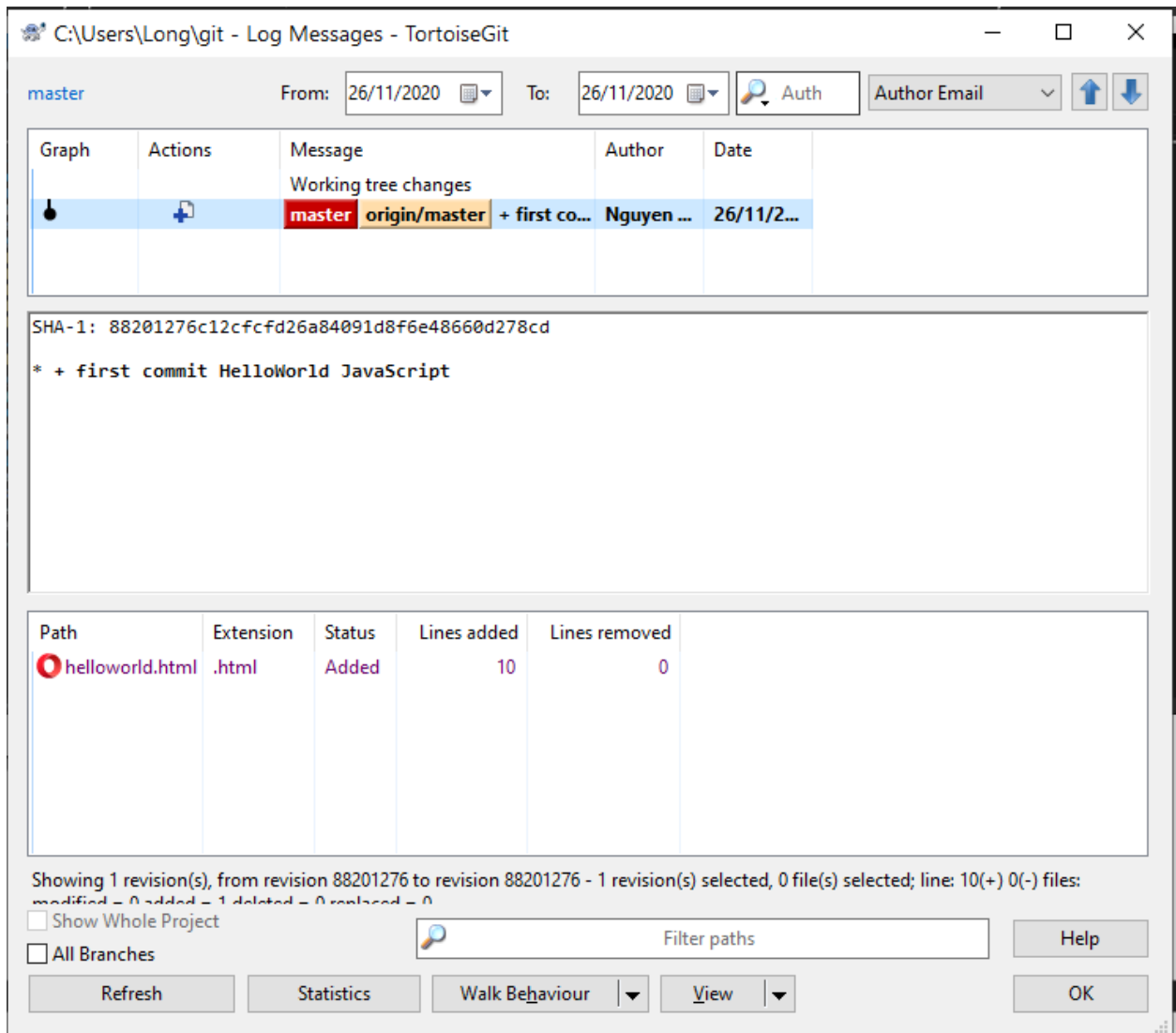


+) Bước 5: Show log

Xem các thay đổi của source



Sử dụng ShowLog để xem lịch sử thay đổi source code và log

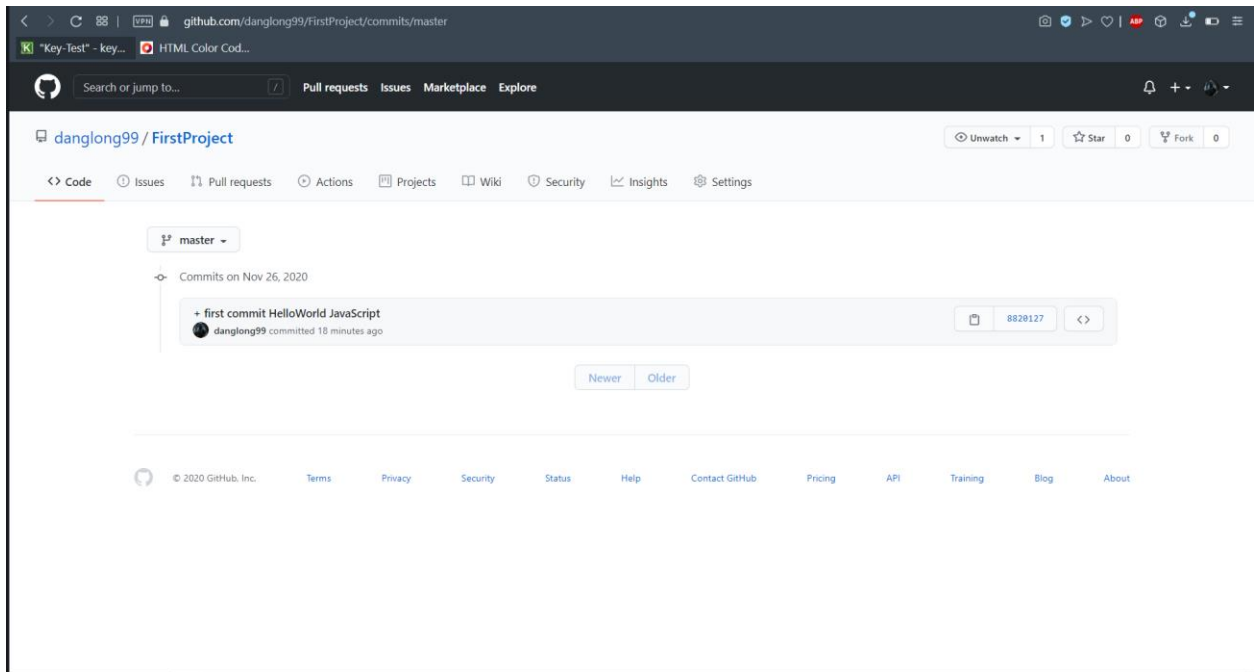


Xem log trên web:

[https://github.com/tên\\_tài\\_khoản/tên\\_repo/commits/master](https://github.com/tên_tài_khoản/tên_repo/commits/master)

Ví dụ:

<https://github.com/danglong99/FirstProject/commits/master>



Bên cạnh sử dụng GIT giúp làm việc teamwork thì sử dụng Git sẽ giúp quản lý tốt hơn source code cũng như tiến độ làm việc của dự án, tránh rủi ro xảy ra trong quá trình làm việc.