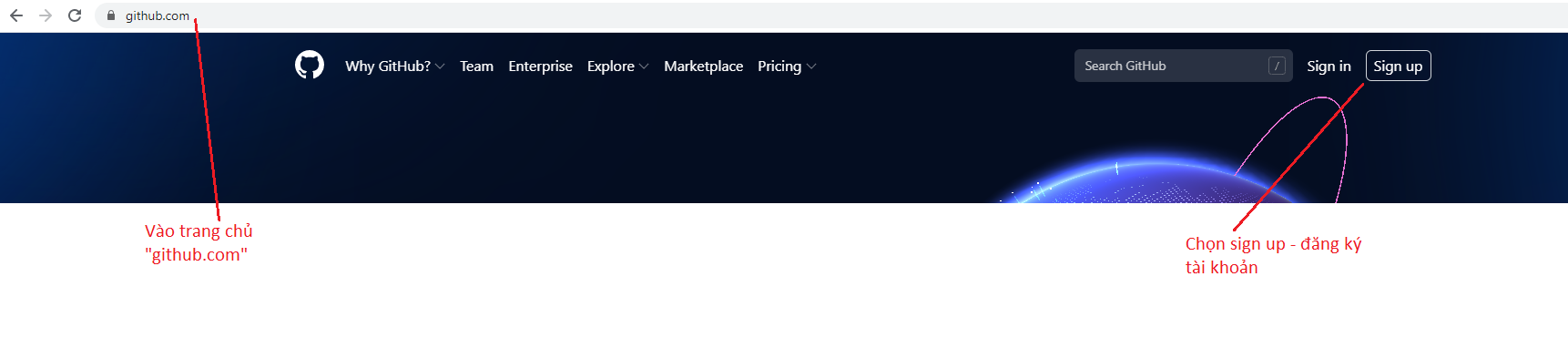
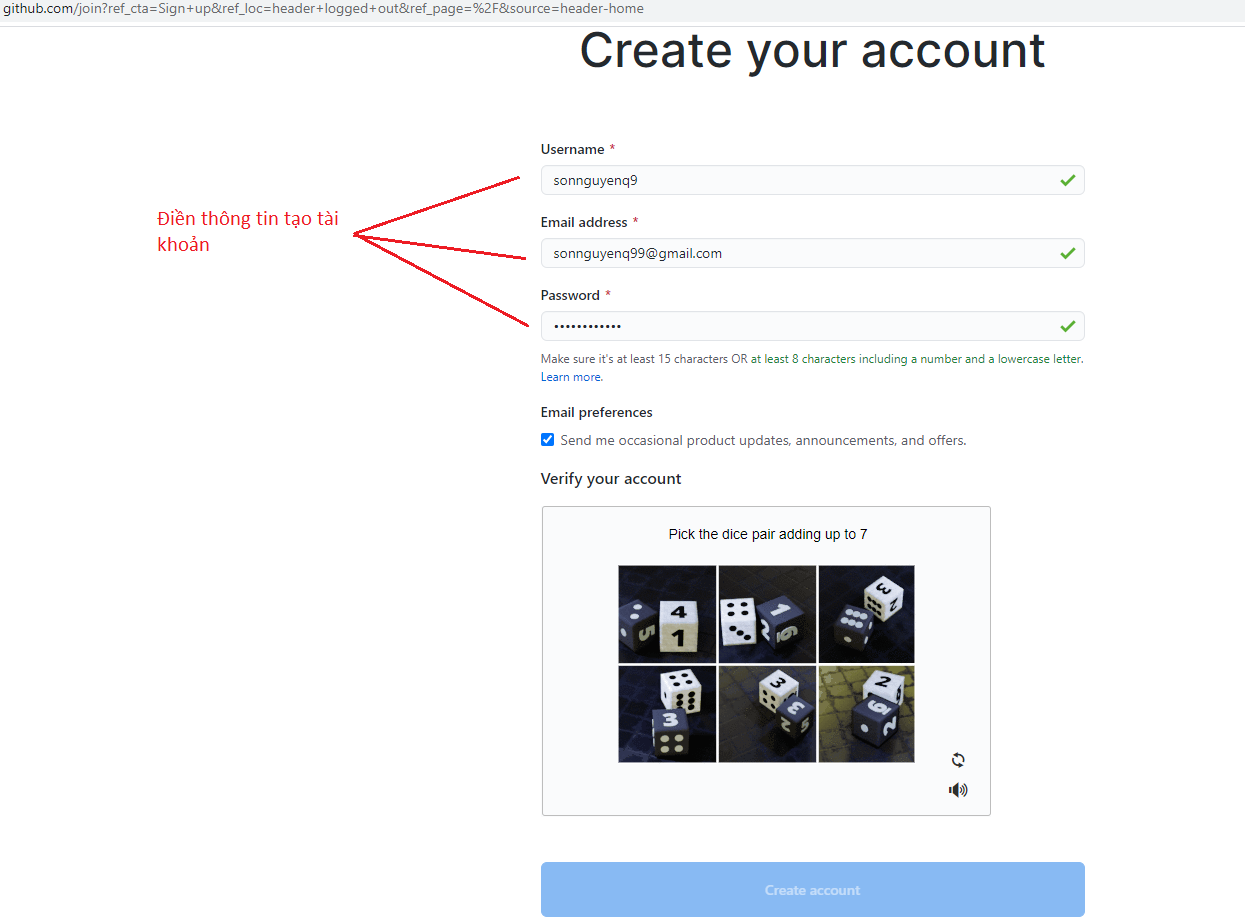
NGUYỄN ĐĂNG SƠN – 1411061644

BÁO CÁO GITHUB

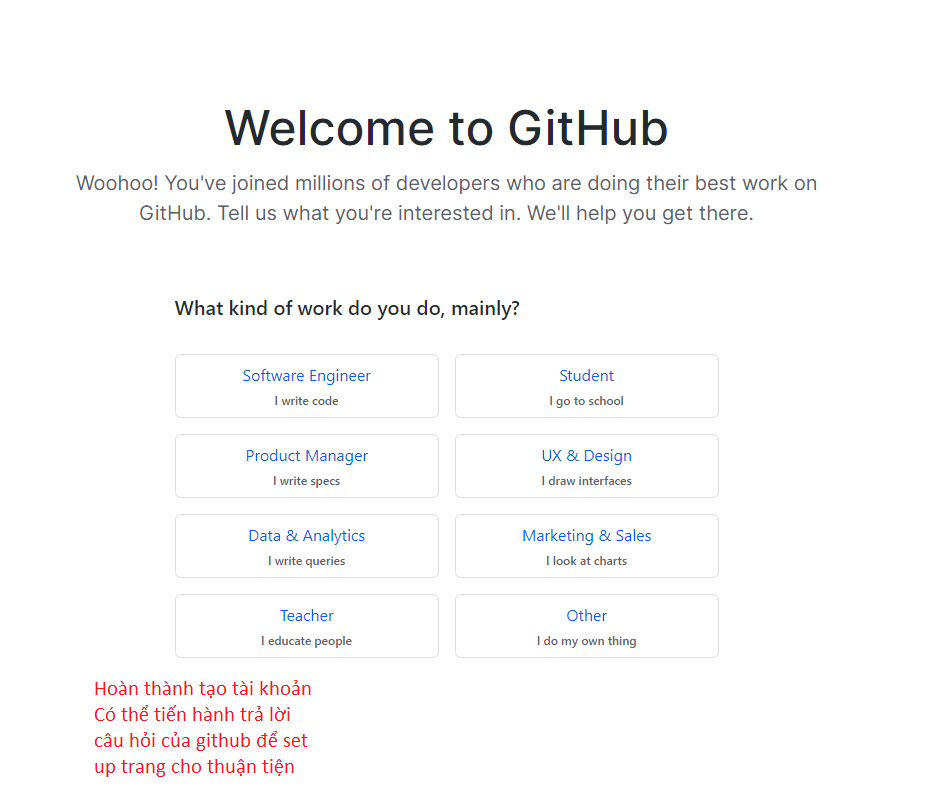
Đăng ký tài khoản:



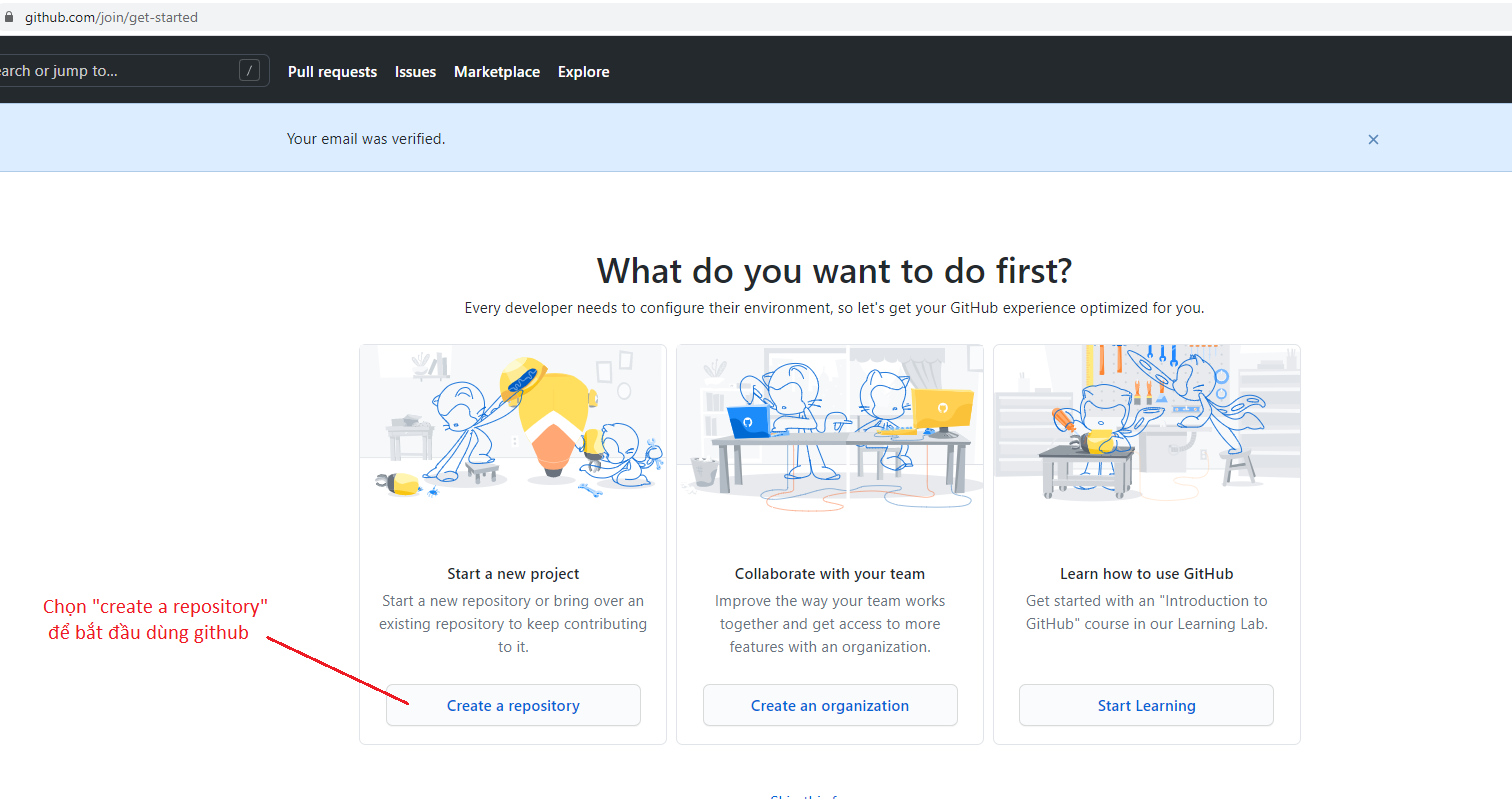
Điền thông tin tạo tài khoản



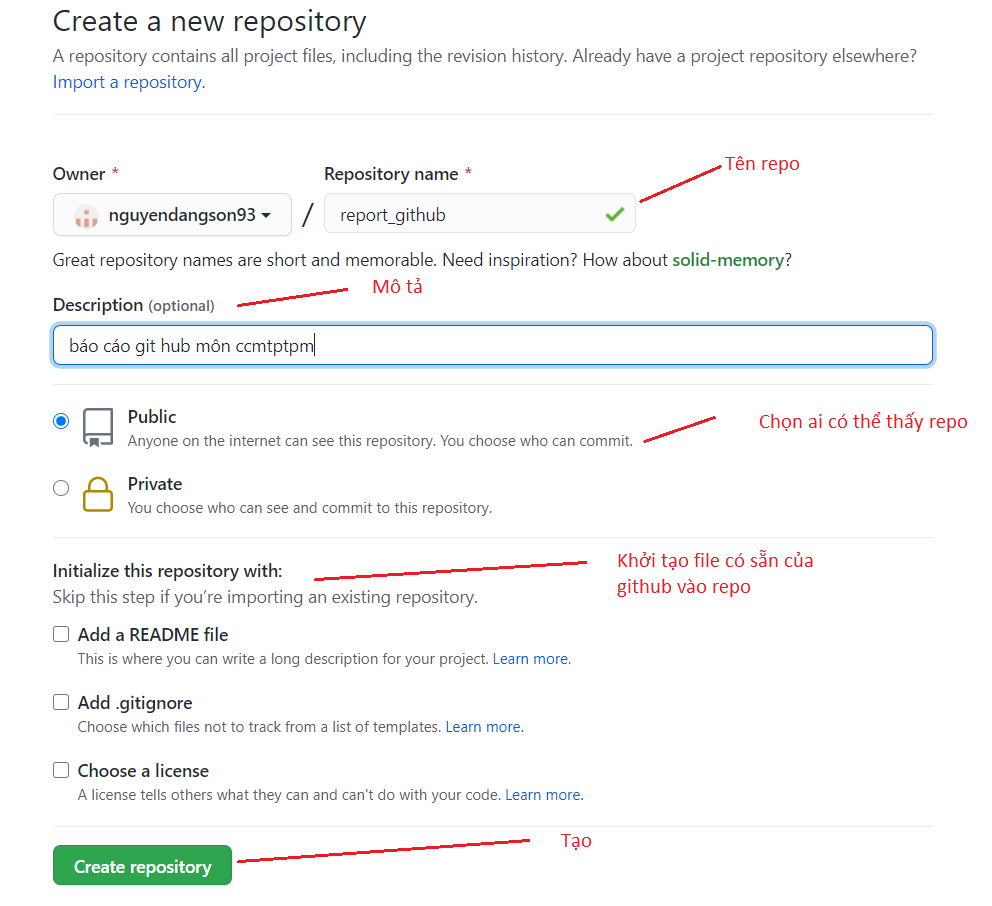
Hoàn thành đăng ký



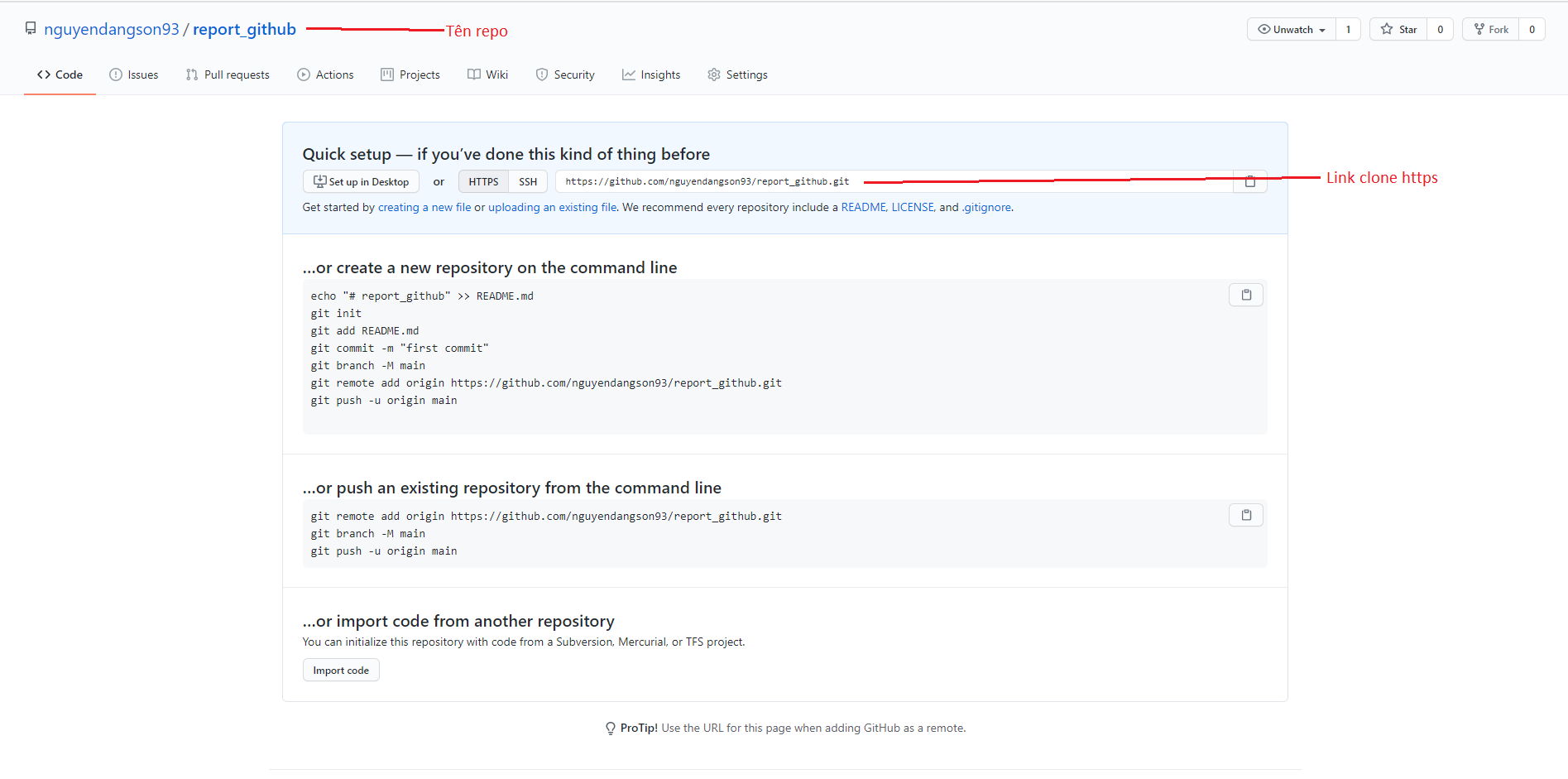
Bắt đầu dùng github bằng cách tạo 1 repository



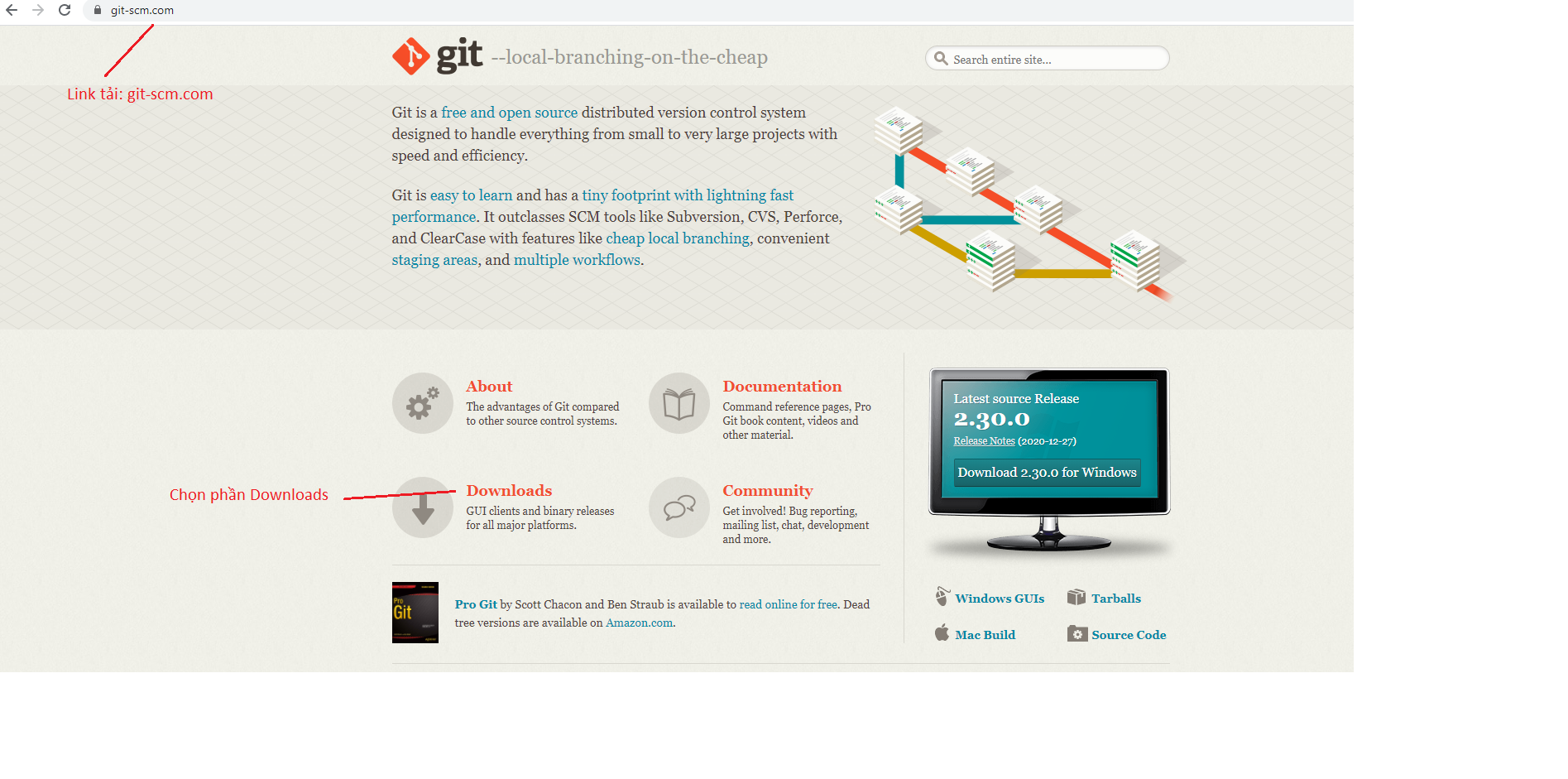
Tạo repo



Trang repo sau khi tạo

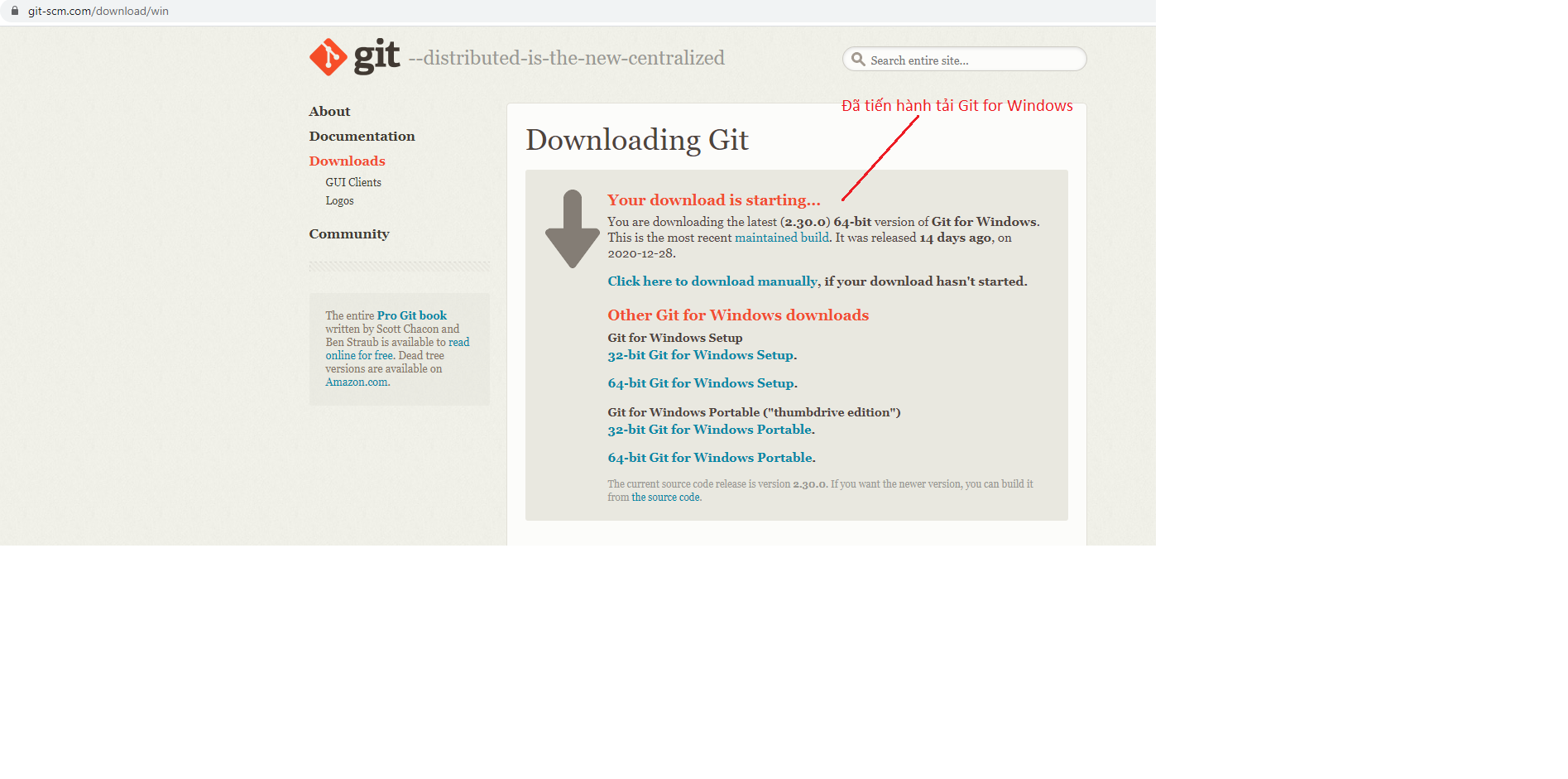


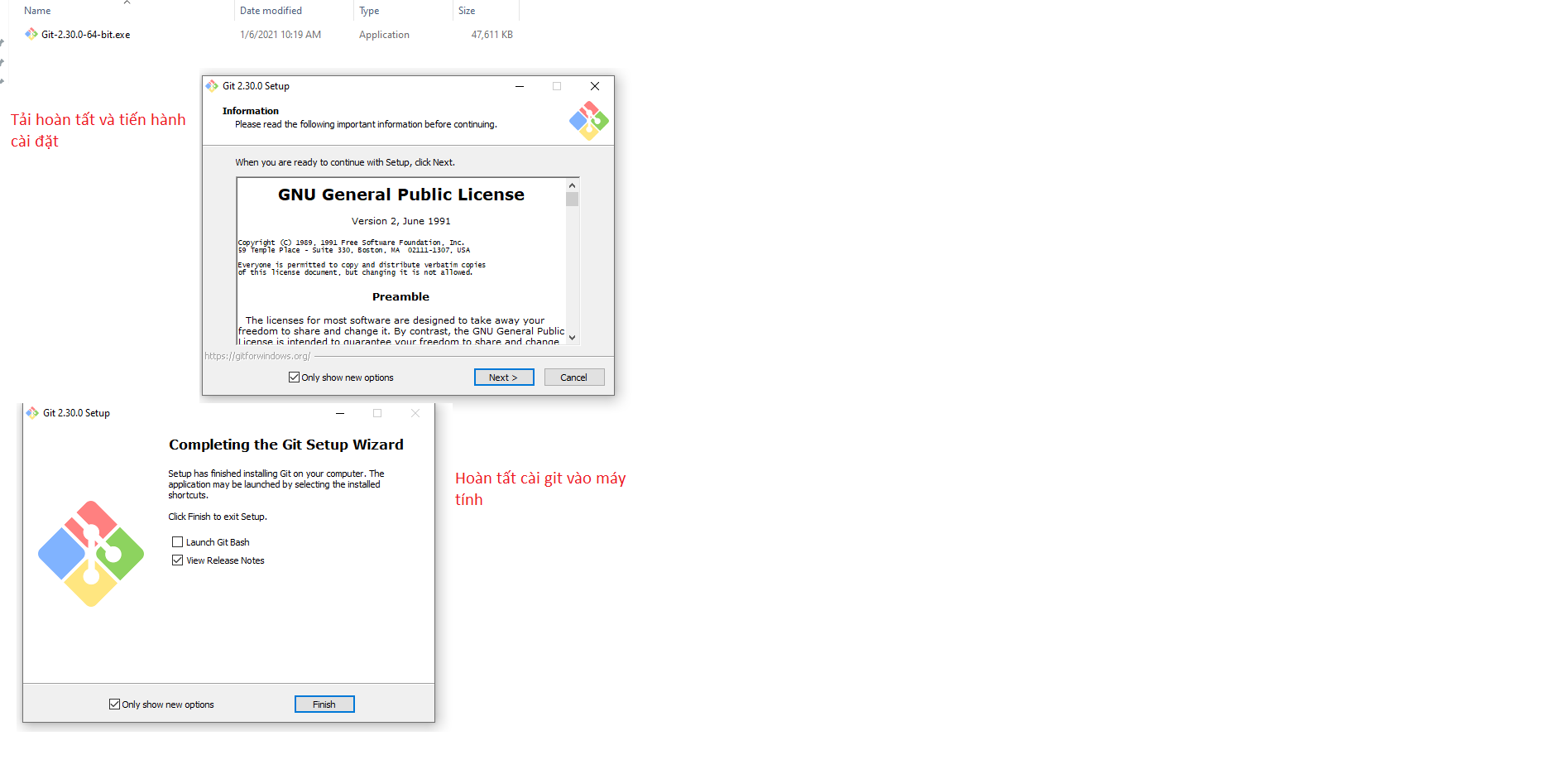
Tải git cho máy



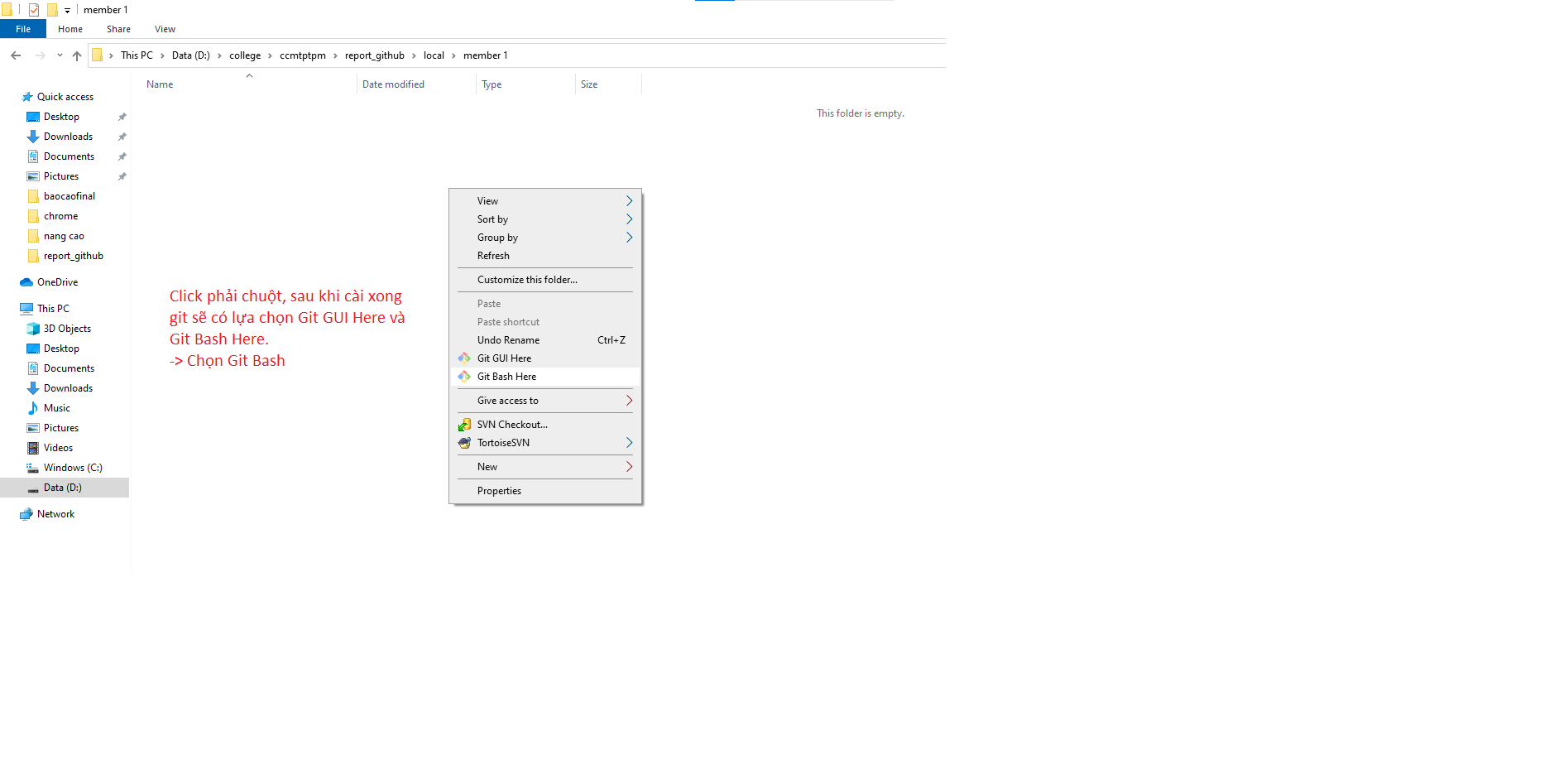
Chọn nền tảng tải và cài đặt

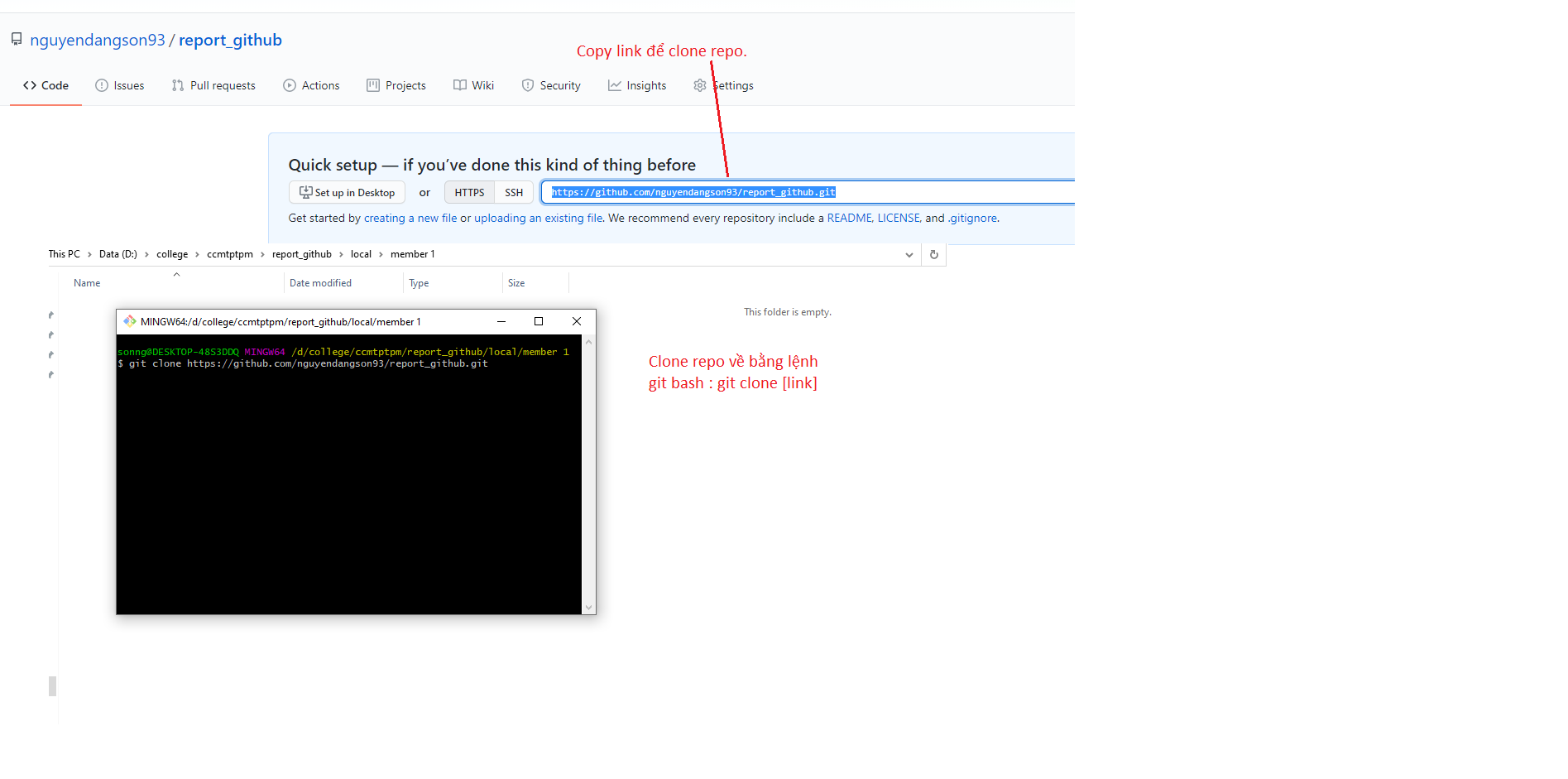




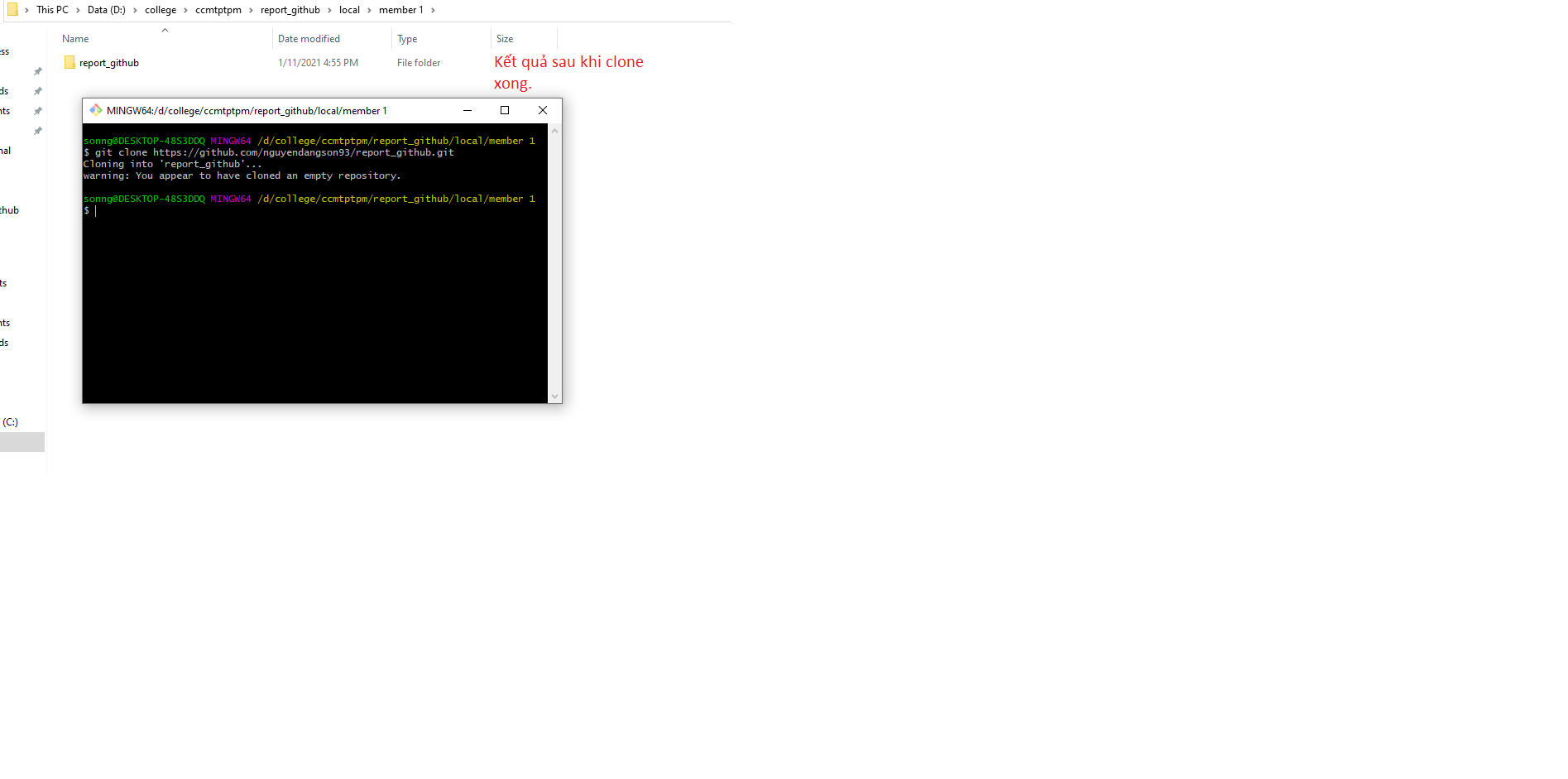


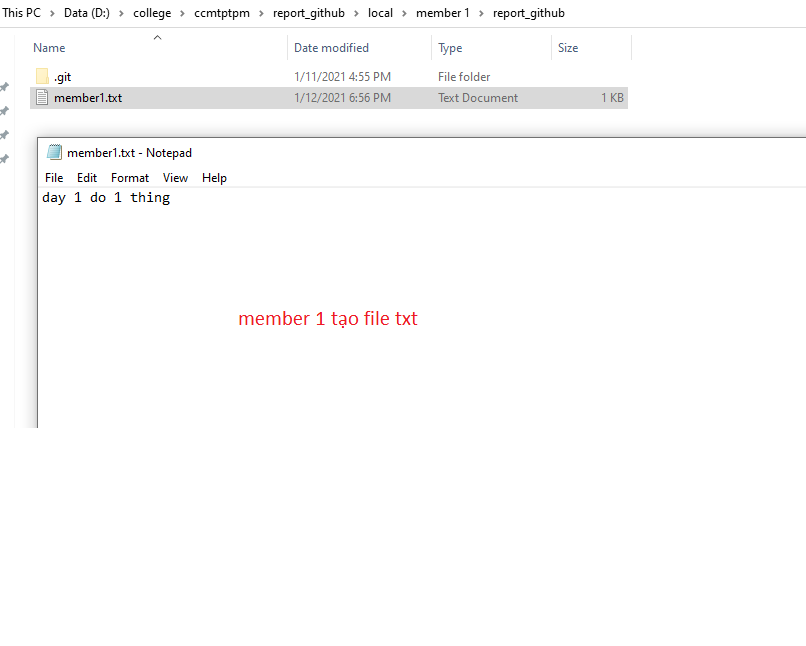
Khởi tạo folder local để clone repo



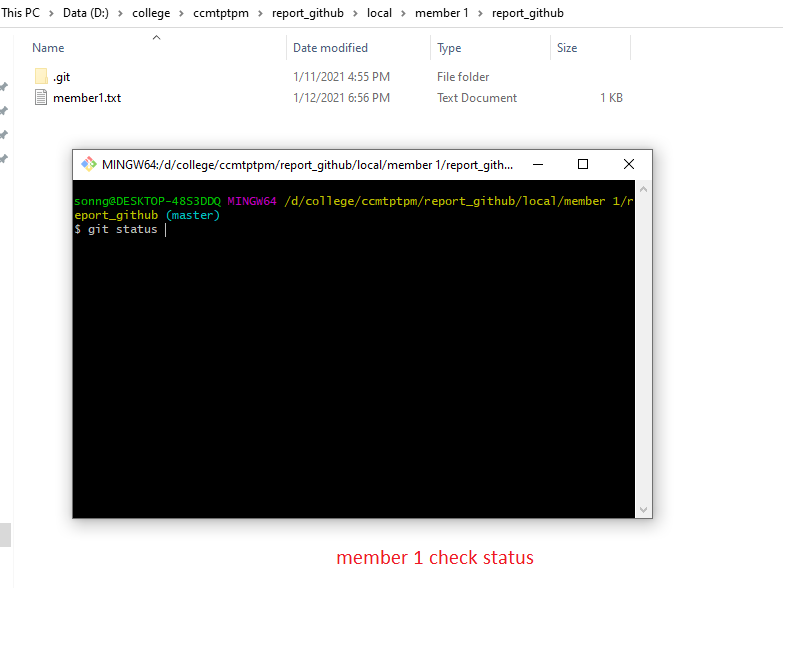


Kết quả sau khi clone xong

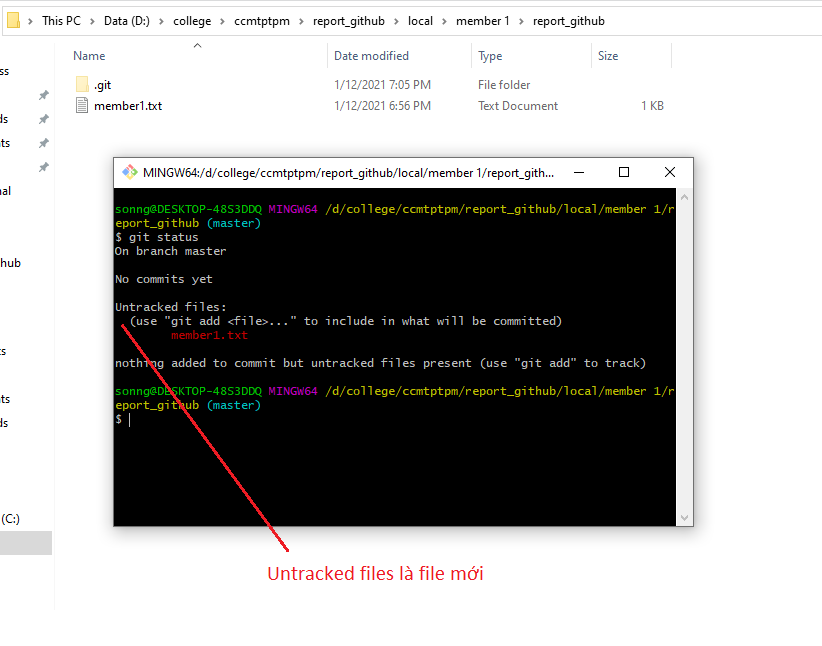
Menber 1 tạo file text



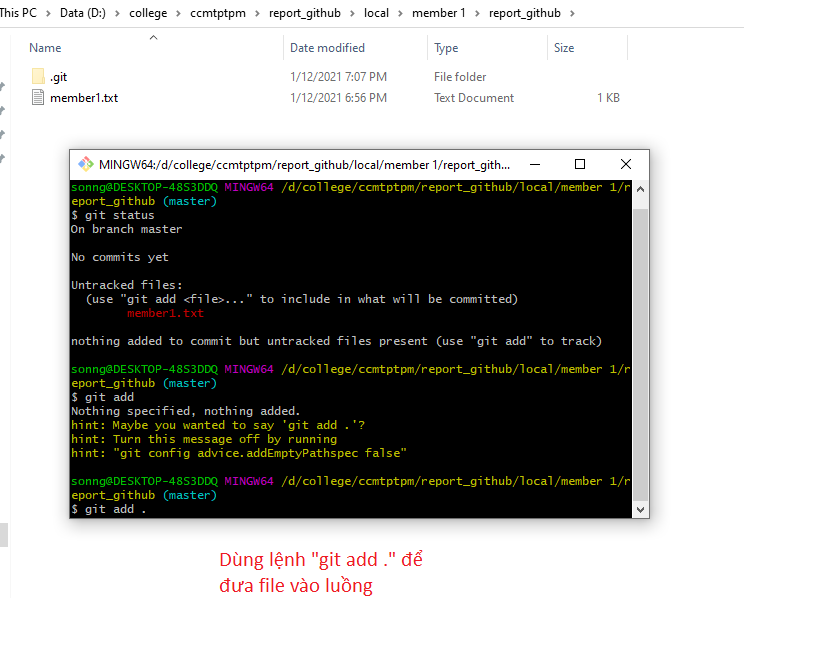
Member 1 check status ở working copy bằng lệnh “git status”



Có thay đổi trên working copy so với staging



Đưa thay đổi vào luồng làm việc trên working copy



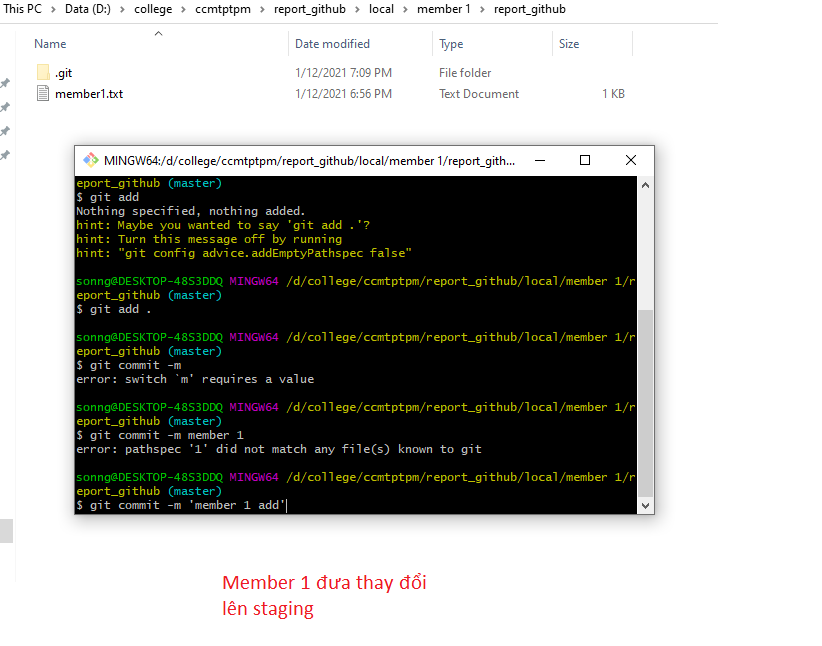
Lưu ý khi dùng lần đầu git bash sẽ không rõ người dùng nào đang muốn commit



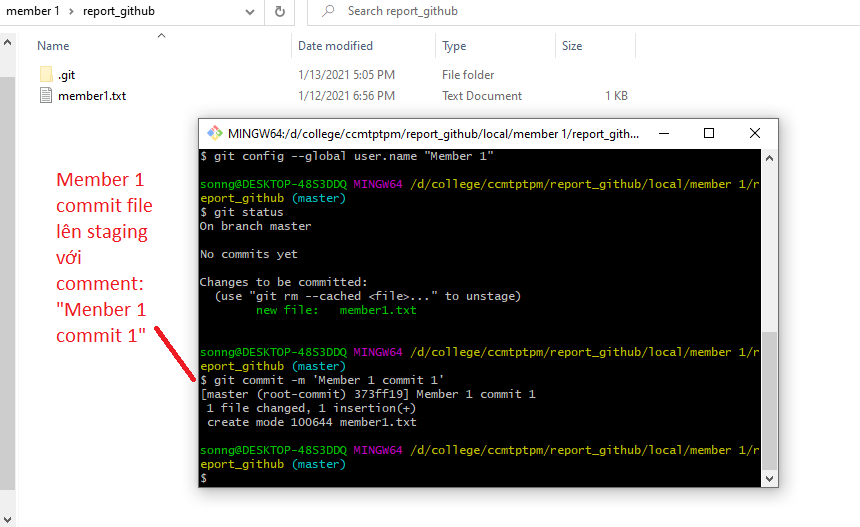
Có thể config người dùng bằng lệnh “git config –global user.email “[email người dùng]”

“git config –global user.name “[tên người dùng]”

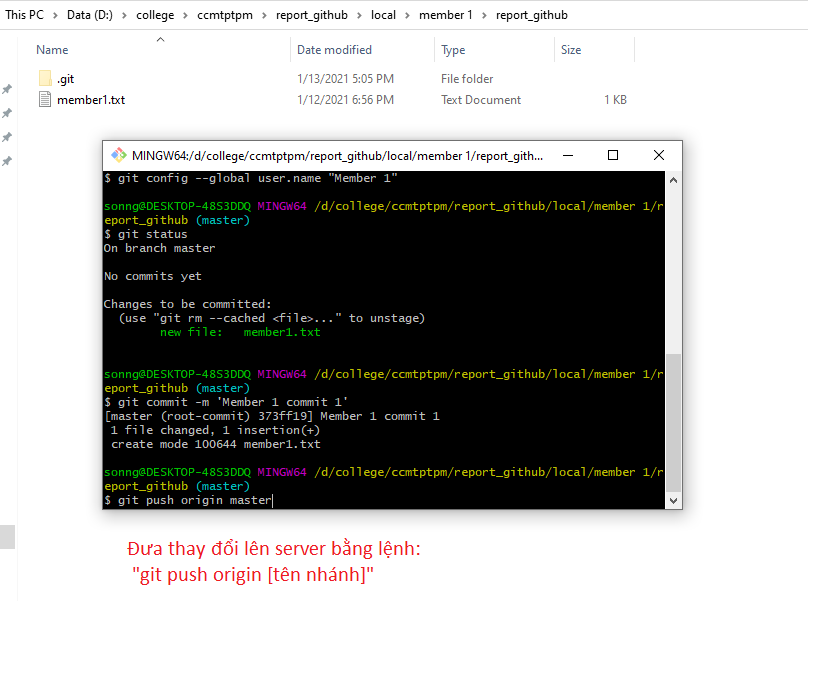
Đưa thay đổi lên stage



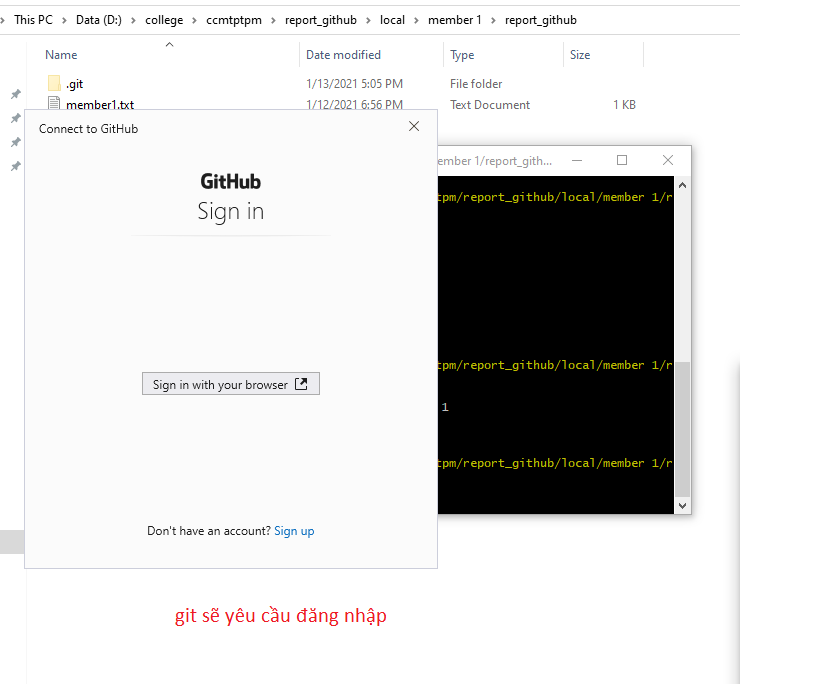
Member 1 đã commit xong



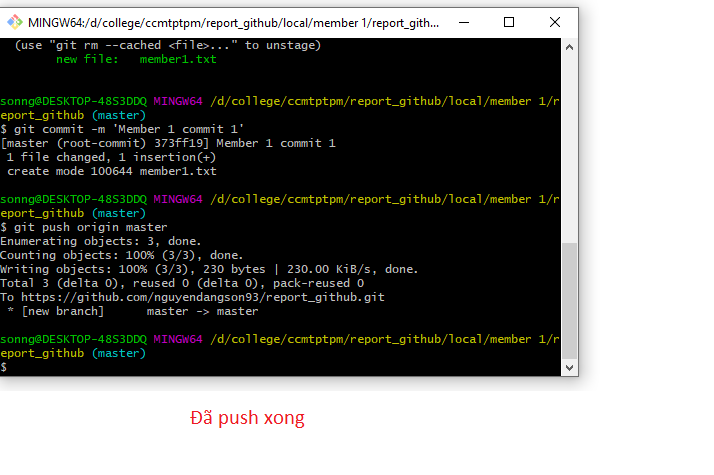
Đưa các thay đổi từ staging lên server



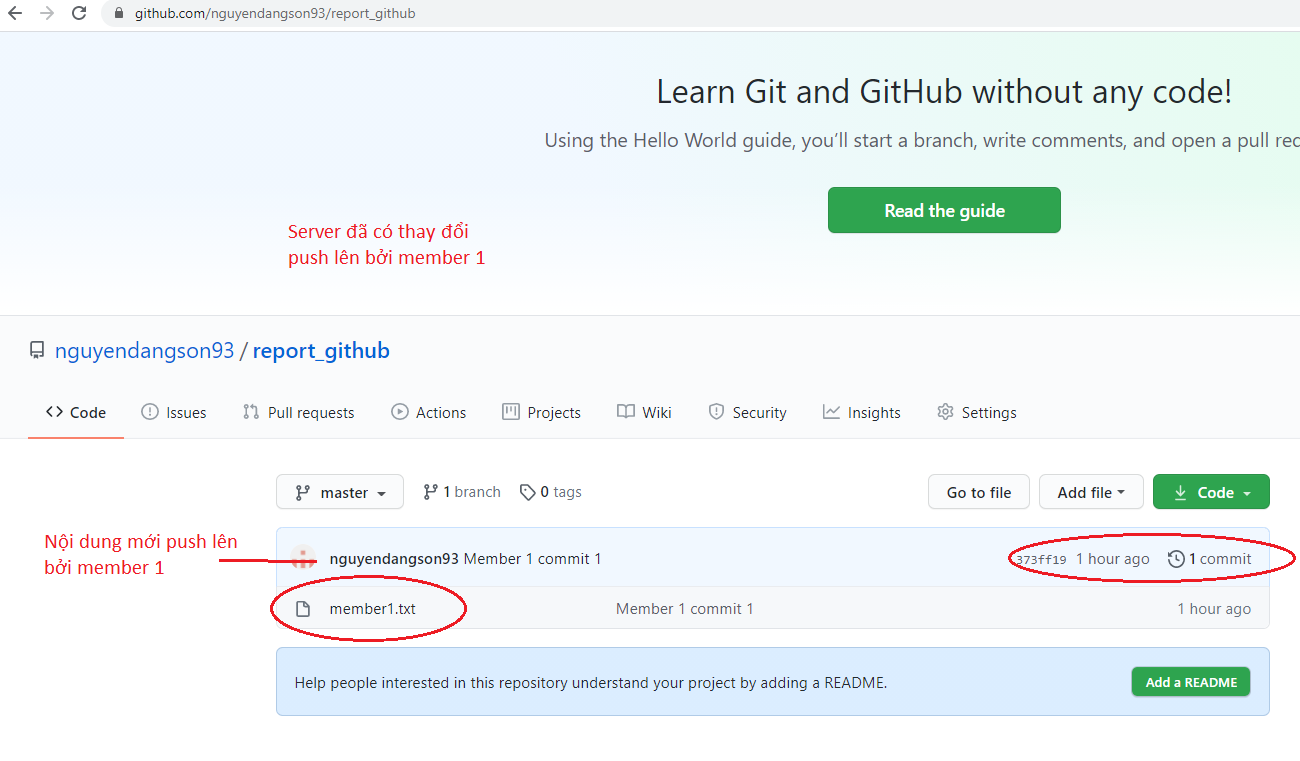
Git sẽ yêu cầu đăng nhập nếu người dùng chưa config tài khoản github trên git ở local



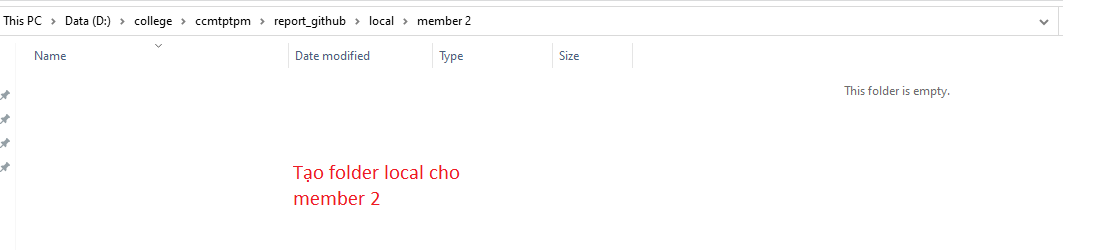
Tiến hành đăng nhập tài khoản github, sau khi hoàn thành đăng nhập thì git bash sẽ chạy lệnh push lên server



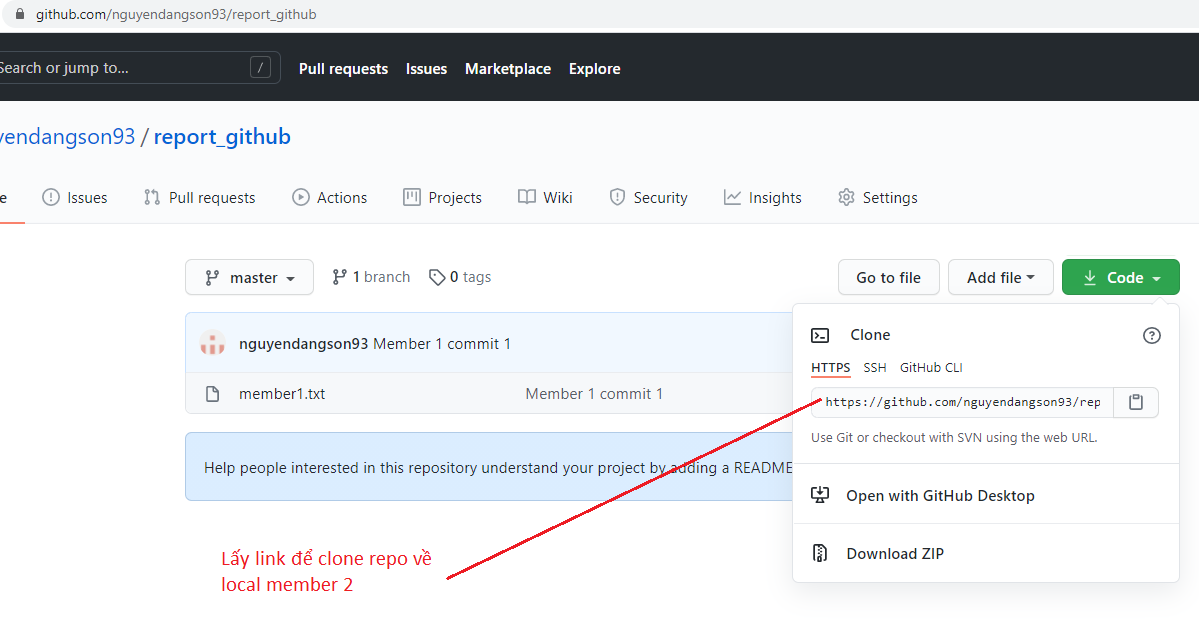
Kiểm tra thay đổi trên server

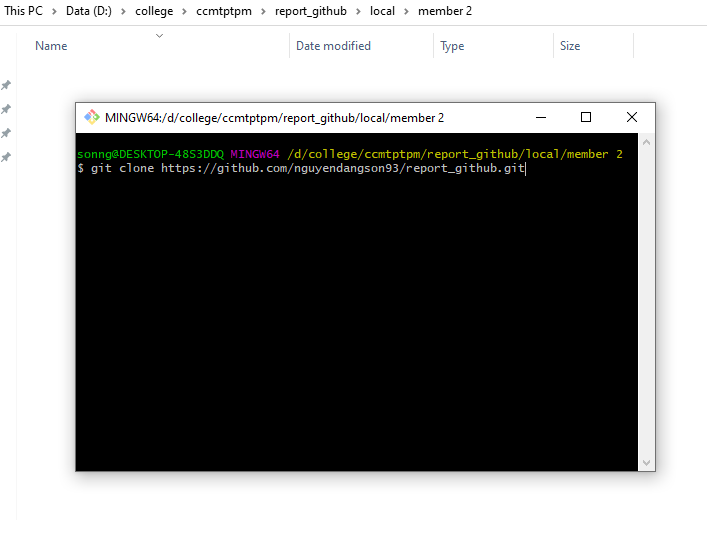


Tạo 1 folder local cho member 2

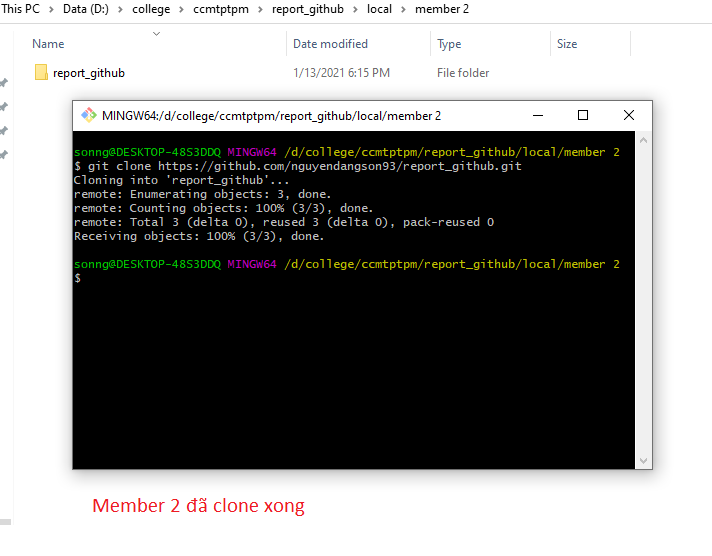


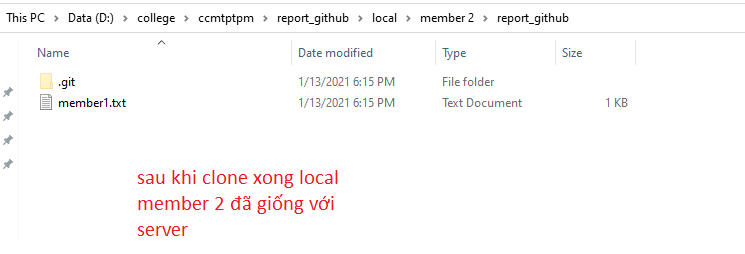
Lấy link để clone repo về local cho member 2



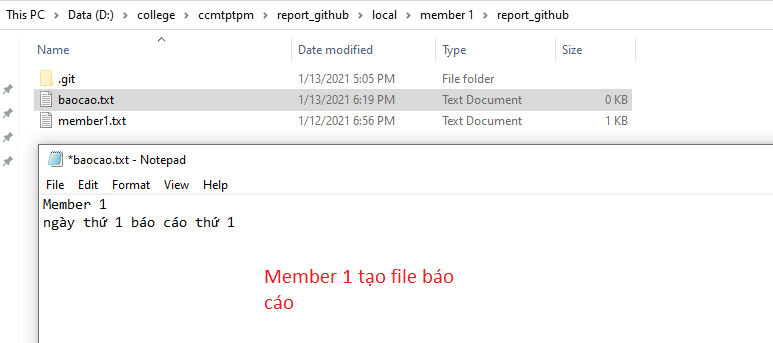


Member 2 hoàn tất clone repo về local

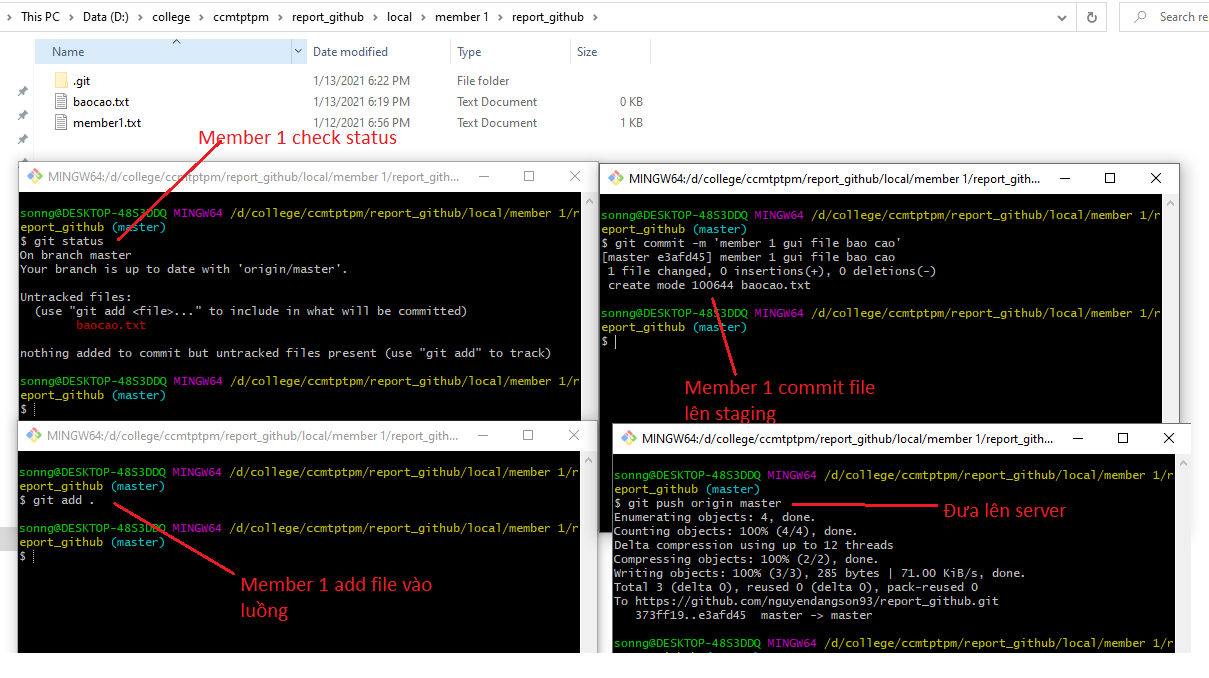




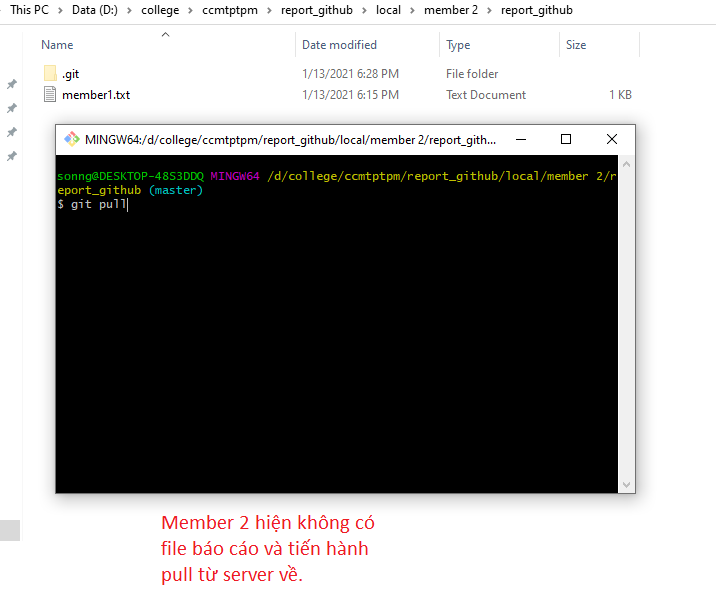
Member 1 tạo 1 file báo cáo mới



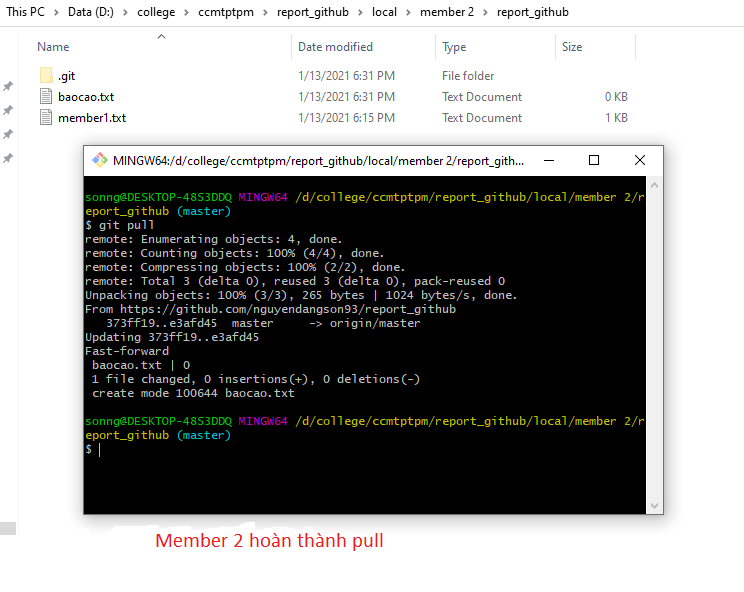
Và đưa file lên server



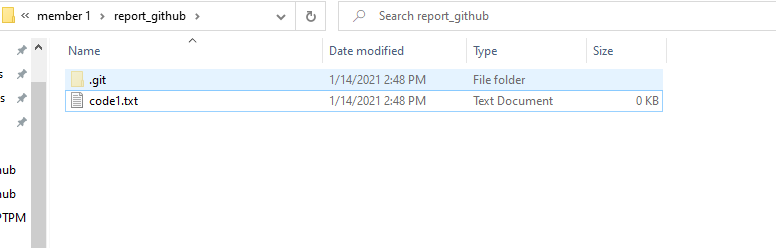
Member 2 tiến hành pull thay đổi từ server về local với lệnh git: “git pull”



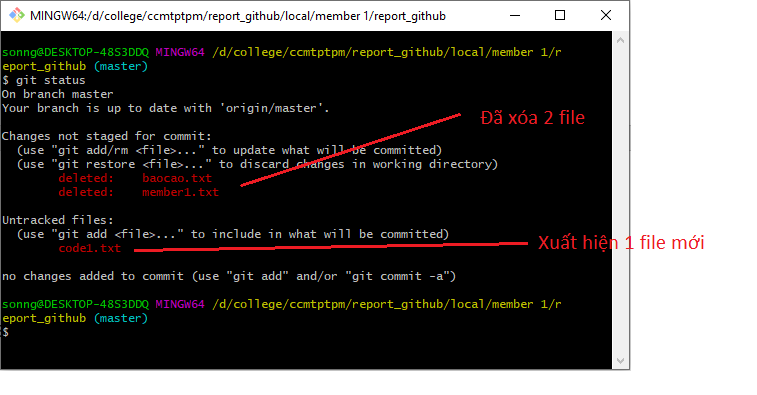
Member hoàn thành pull thay đổi từ server về và đã có file báo cáo



Kịch bản mới: member 2 xóa file baocao.txt và member1.txt, đồng thời tạo mới file code1.txt



Member 2 check status



Git thông báo working directory hiện tại trên local member 2 có những thay đổi:

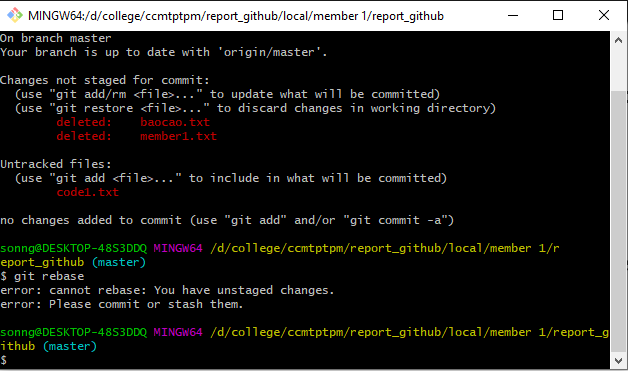
* Đã có 2 file bị xóa baocao.txt và member1.txt
* Có file mới untracked code1.txt

Bây giờ ta có 2 hướng:

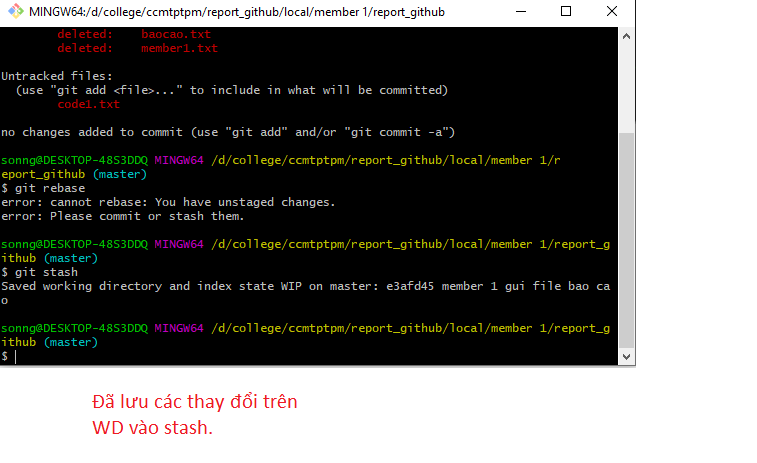
1/ Đưa WD về trạng thái giống trên server bằng lệnh “git rebase”

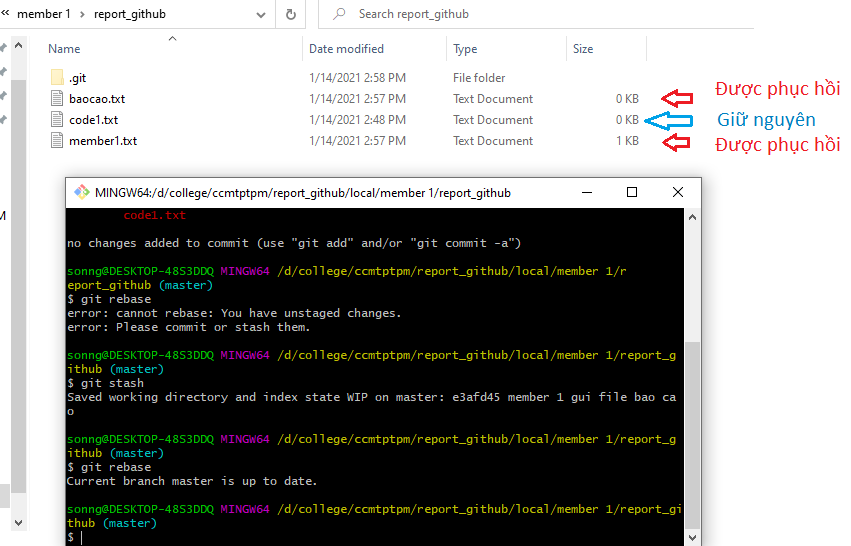
2/ Đưa WD về trạng thái trước đó của chính WD, nghĩa là chỉ thao tác trên local

Lúc này nếu dùng lệnh “git rebase” để đưa working directory(WD) về phiên bản trước thì git sẽ báo không thực hiện được, cần phải lưu các thay đổi trên WD trước khi quay lại phiên bản trước. Ta sẽ dùng lệnh “git stash” để lưu các status hiện tại trên WD.



Giờ đây các status trên WD đã được lưu lại, ta có thể tiến hành quay lại phiên bản trước với lệnh “git rebase”.

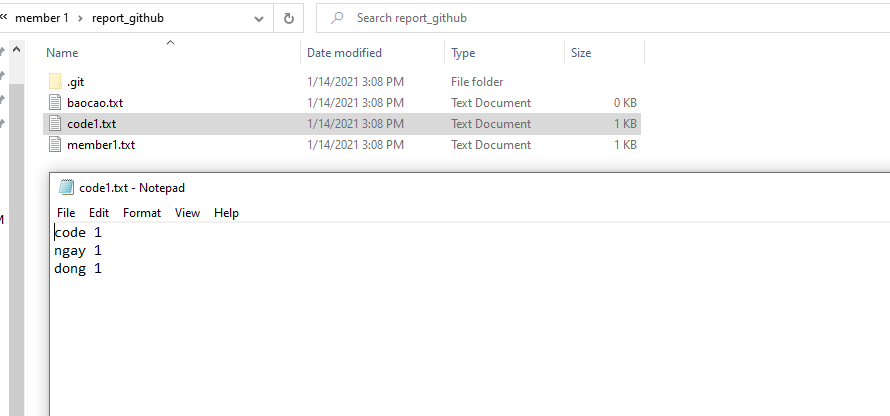




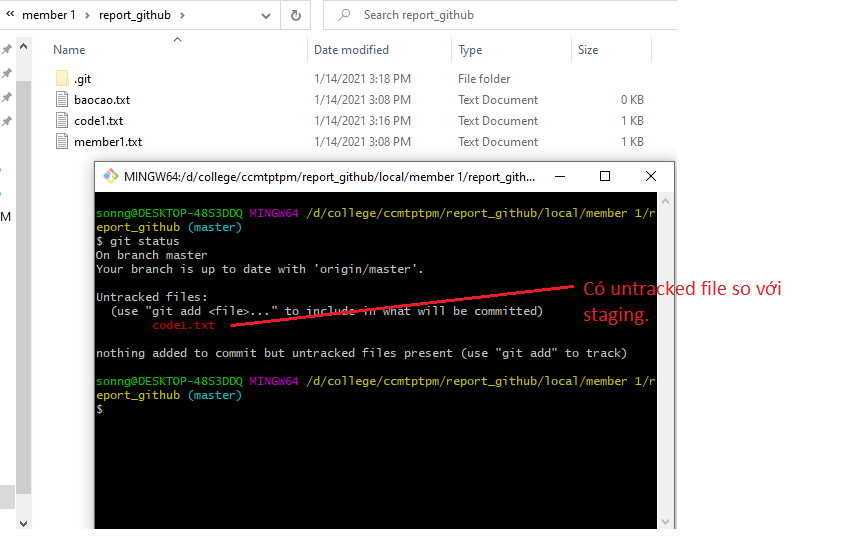
Đây và với hướng phục hồi lại các file trên server về WD với lệnh “git rebase”

Với hướng ta muốn WD quay trở lại phiên bản trước trên chính WD trước đây, ta thực hiện như sau

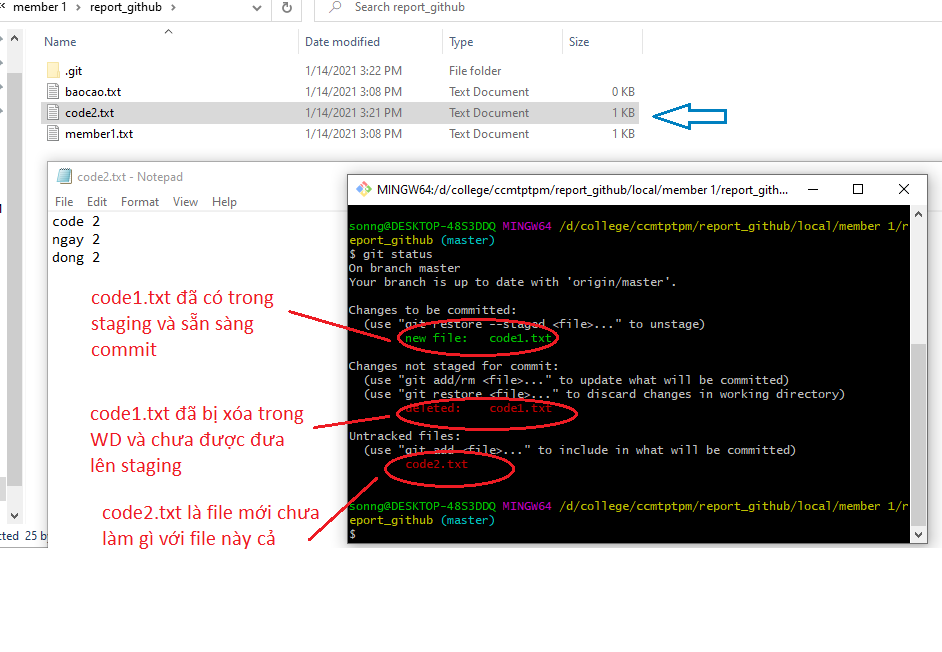
Giả sử member 1 đang làm việc với file code1.txt



Member 1 check status so với staging



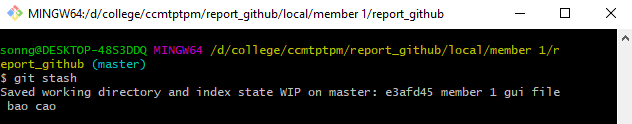
Member 1 tiến hành đưa các thay đổi này vào staging với lệnh “git add .” và xóa file code1.txt, đồng thời tạo file code2.txt và check status



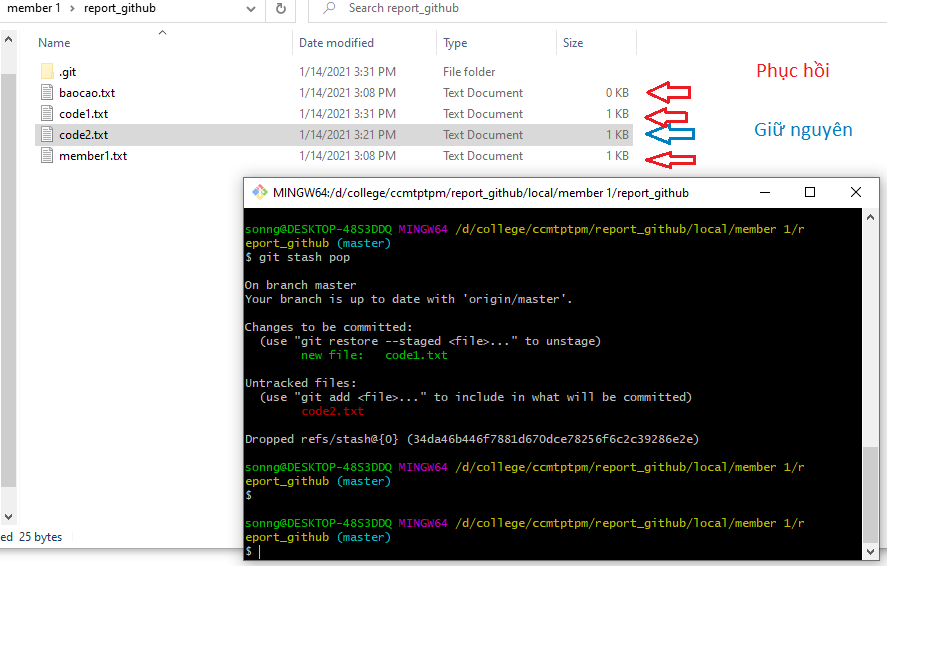
Có nghĩa là code1.txt đã ở trong staging, trên WD thì code1.txt đã có trạng thái mới là “đã bị xóa” và trên WD cũng có file mới chưa thao tác gì là code2.txt

Như vậy bây giờ member 2 muốn đưa code1.txt trở lại WD thì thực hiện như sau

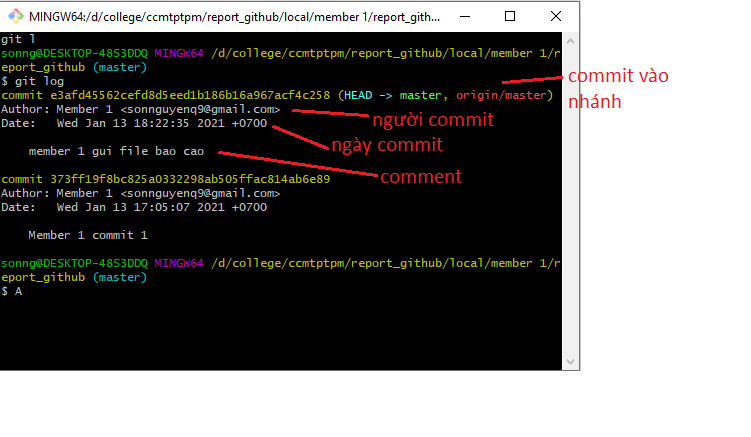
Sử dụng lệnh “git stash” để lưu các status hiện tại trên WD



Tiếp theo dùng lệnh “git stash pop” để đưa WD trở lại trạng thái trước

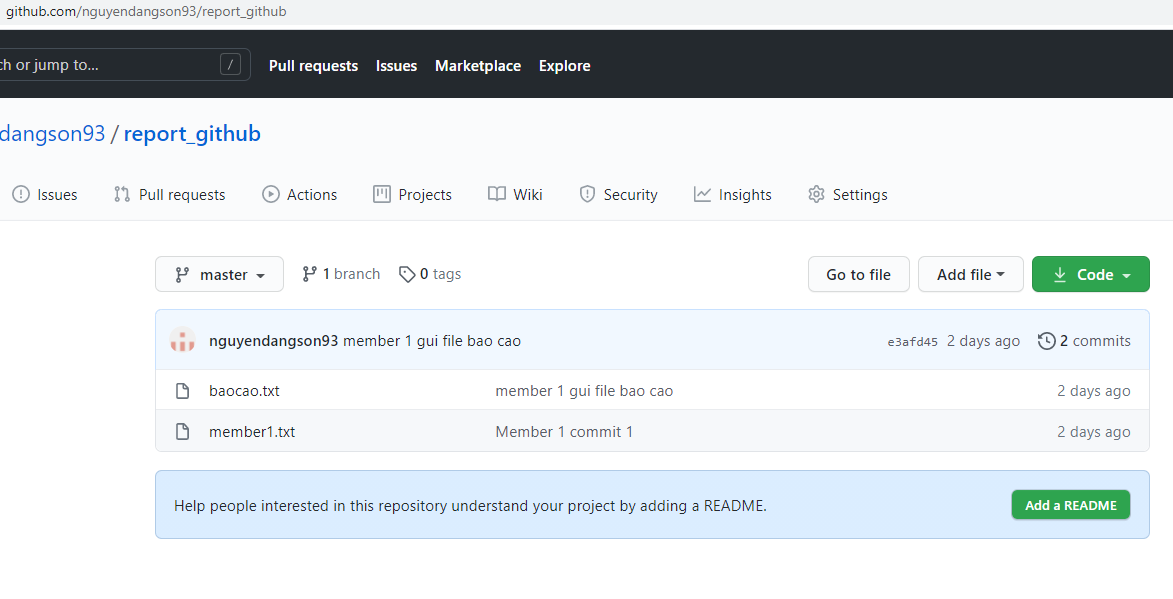


Xem lịch sử các phiên bản với lệnh “git log”



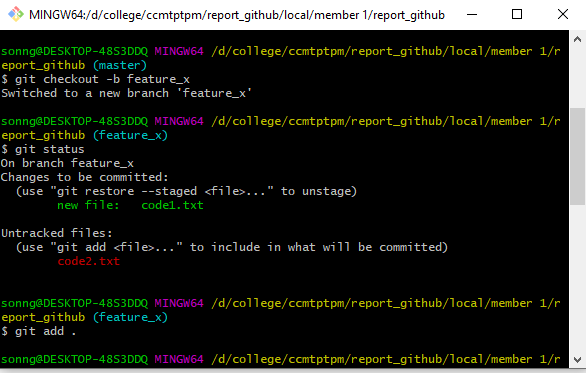
Tạo nhánh với lệnh “git checkout –b [tên nhánh]”

Các nhánh hiện tại của repo chỉ có 1 nhánh master

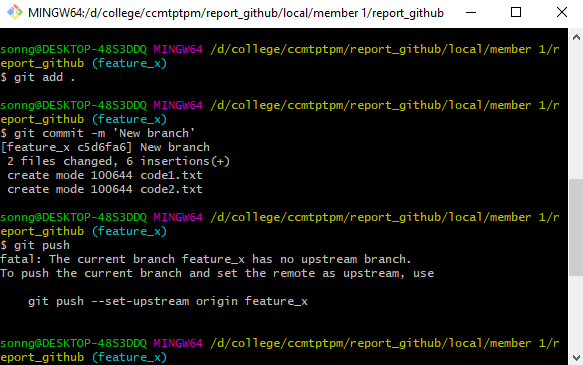


Sau khi tạo nhánh với lệnh “git checkout –b [feature\_x]”

Kiểm tra lại status, ta sẽ thấy git thông báo trên nhánh mới có thay đổi có thể commit



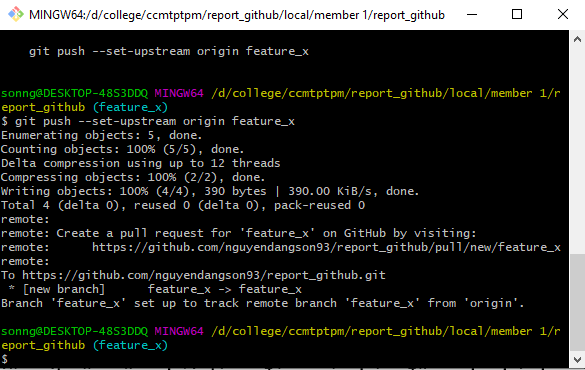
Ta thực hiện thao tác push lên server với lệnh “git add .” , “git commit” và “git push”



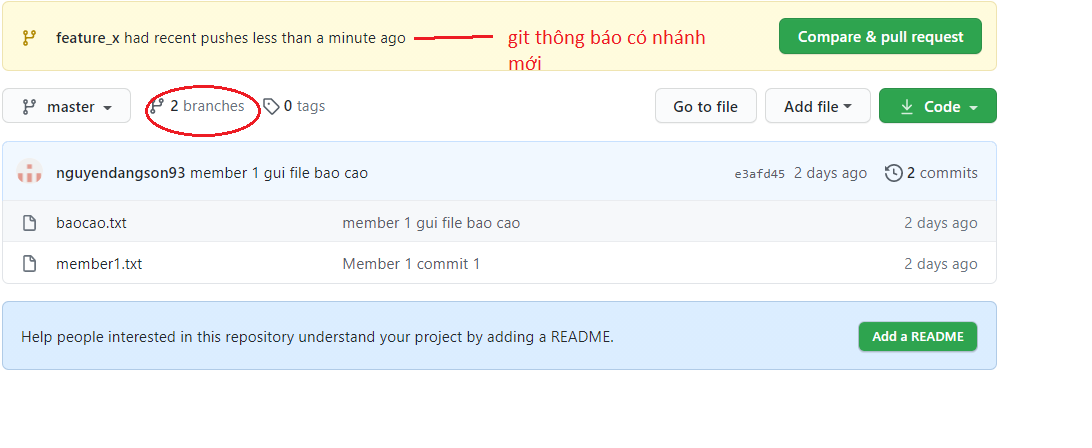
Tuy nhiên git yêu cầu phải dùng đúng cú pháp để push nhánh mới lên server là

“git push –set-upstream origin [tên nhánh mới]”

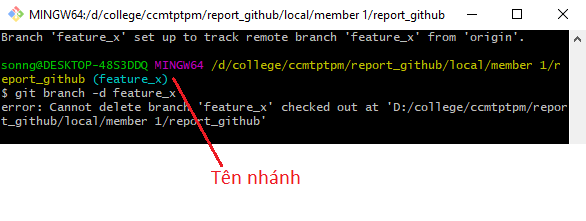
Kết quả



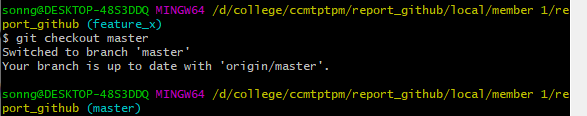
Kiểm tra repo trên server



Xóa nhánh với lệnh “git branch –d [Tên nhánh muốn xóa]”

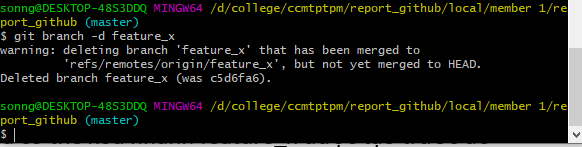


Do ta đang làm việc trên nhánh [feature\_x] nên git không cho phép xóa nhánh này, ta thực hiện chuyển nhánh làm việc với lệnh “git checkout [tên nhánh]”

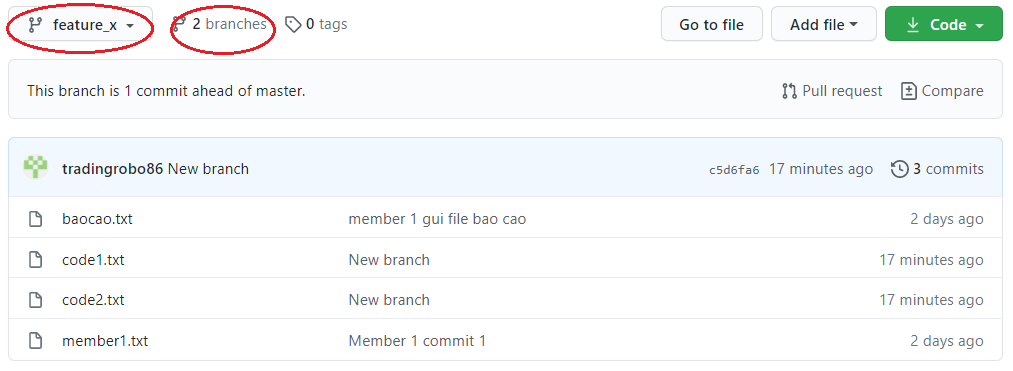


Sau khi checkout master thì nhánh làm việc hiện tại đã đổi thành (master)

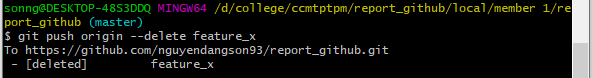
Lúc này ta đã có thể xóa nhánh feature\_x được tạo trước đó



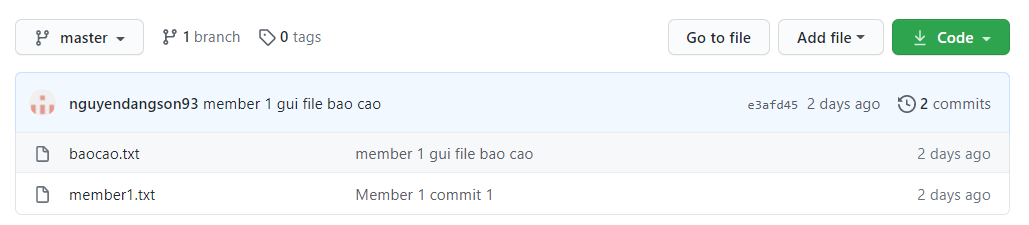
Git có cảnh báo rằng: ta chỉ mới xóa nhánh trên WD, trên repo server nhánh này vẫn còn



Ta có dùng lệnh “git push origin –delete [tên nhánh]

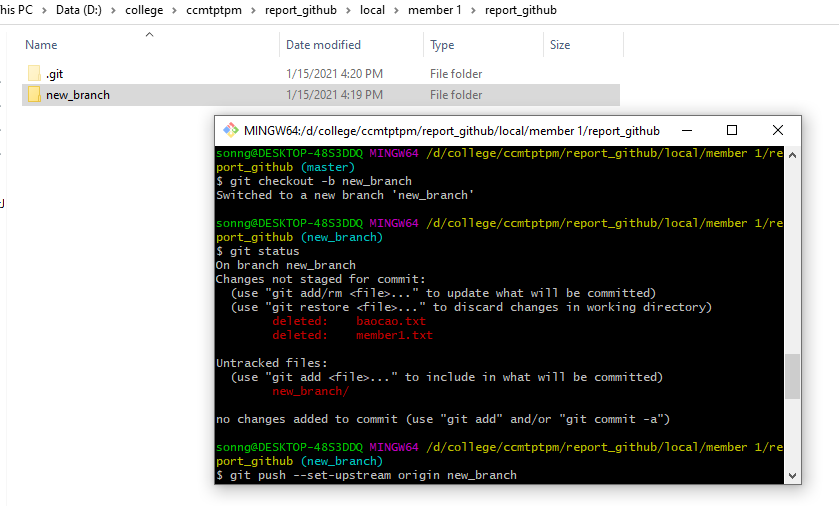


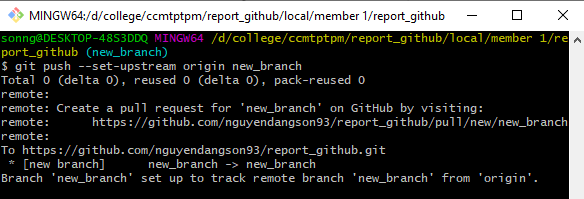
Kiểm tra trên repo

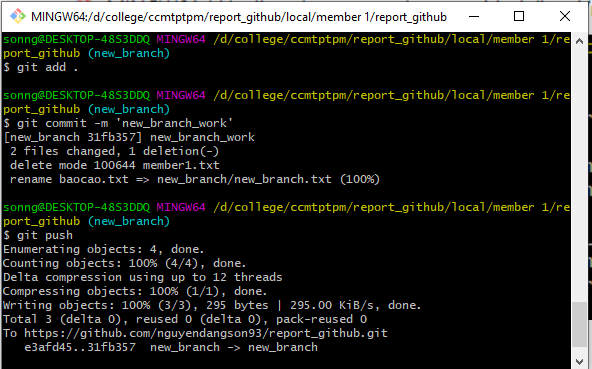


Hiện repo chỉ có 1 nhánh master. Đã hoàn thành xóa nhánh remotely.

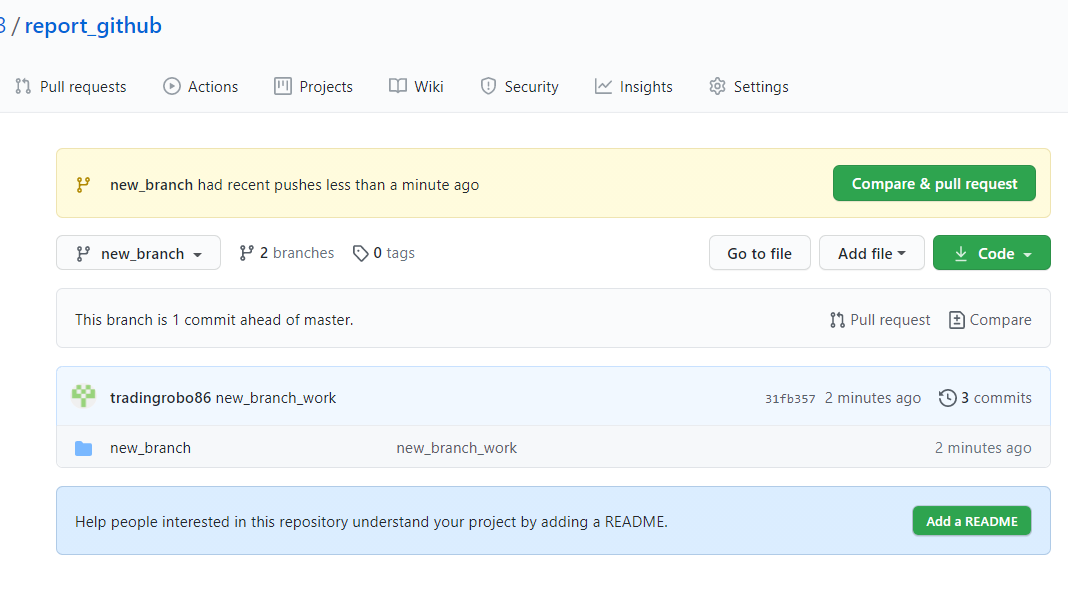
Tạo 1 nhánh mới tên new\_branch trên WD, local tạo 1 folder new\_branch, tiến hành add, commit, push lên repo server



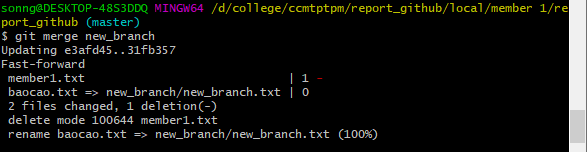




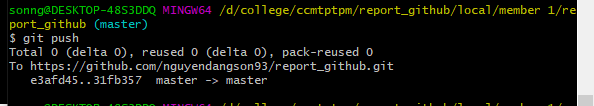
Kiểm tra repo server đã có nhánh mới new\_branch



Ta có thể nhập nhánh [đang làm việc] với [tên nhánh] bằng lệnh “git merge [tên nhánh]”



Như vậy trên WD, nhánh master đã giống với nhánh new\_branch, ta push lên server



Kết quả:

Nhánh master đã giống nhánh new\_branch

