**Bài tập thực hành 06**

Làm quen với Github

# Yêu cầu bài thực hành:

1. Bạn theo link sau: <https://www.techrepublic.com/article/how-to-install-github-desktop/>, theo hướng dẫn trong link để tiến hành cài đặt git trên máy.
2. Bạn hãy tìm hiểu và liệt kê những lợi ích của git và github.

=>

[GitHub](https://github.com/) là một dịch vụ nổi tiếng cung cấp kho lưu trữ mã nguồn [Git](https://topdev.vn/blog/git-la-gi/) cho các dự án phần mềm. **Github có đầy đủ những tính năng của Git**, ngoài ra nó còn bổ sung những tính năng về social để các developer tương tác với nhau.

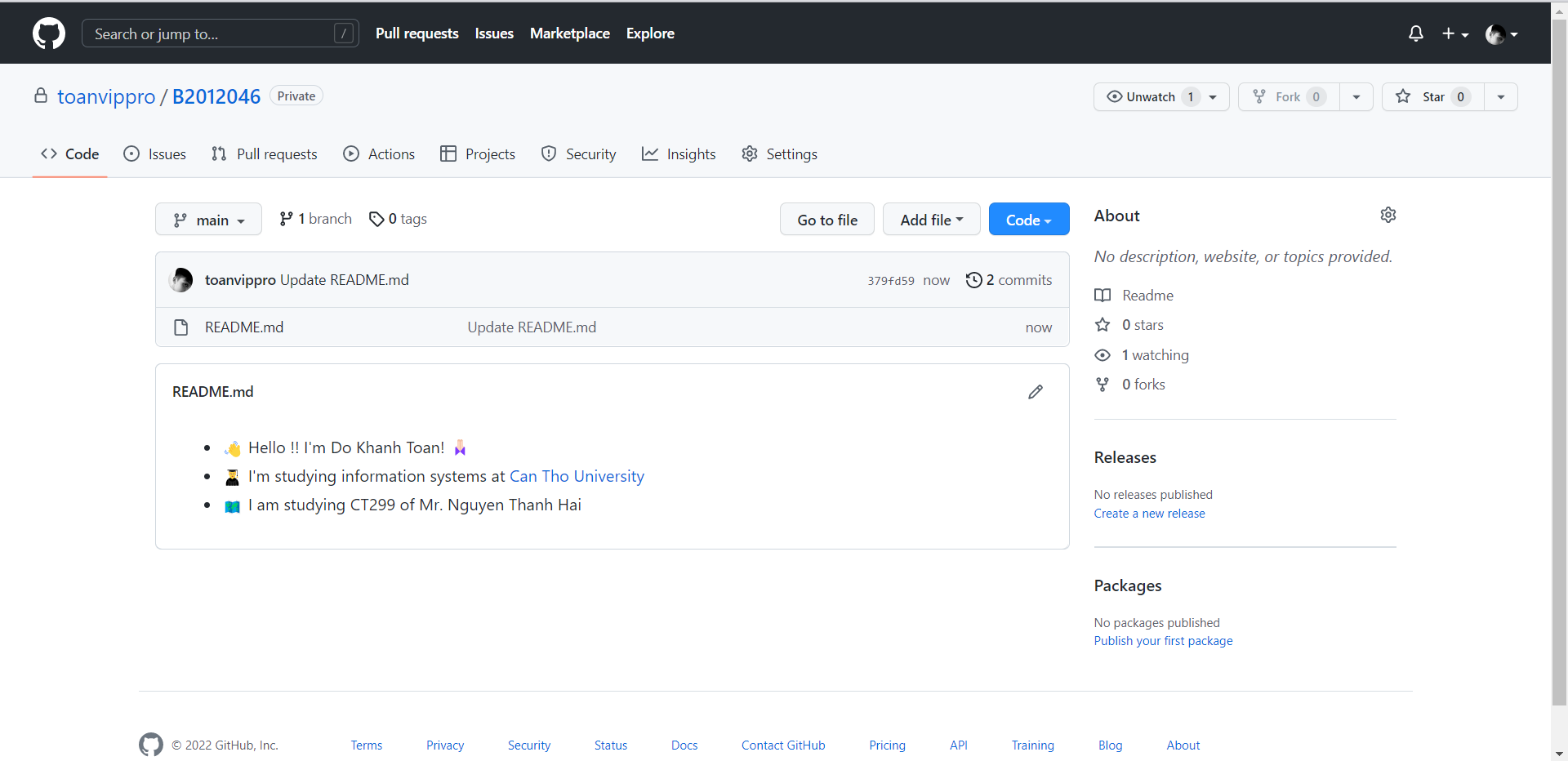
GIT Là công cụ giúp **quản lý source code** tổ chức theo **dạng dữ liệu phân tán**.

* Giúp **đồng bộ source code** của team lên 1 server.
* Hỗ trợ các thao tác **kiểm tra source code** trong quá trình làm việc (diff, check modifications, show history, merge source, …)

1. Repositories trong github là gì (Tham khảo [[1]](#footnote-0))? Có thể chứa được tối đa dung lượng bao nhiêu miễn phí? Cho biết kích thước tập tin lớn có thể tải lên repository github. Tham khảo ở[[2]](#footnote-1), bạn hãy tạo 1 repository với tên là mã số sinh viên của bạn. Trong file readme bạn hãy giới thiệu thông tin về bạn. Chụp hình trang readme đã tạo và dán đường link vào bài tập này

=>

* Repository được hiểu là một kho lưu trữ nơi chứa các files của dự án. Các file đấy có thể là code, hình ảnh, âm thanh hoặc tất cả mọi thứ liên quan đến dự án. Bạn sẽ tổ chức kho lưu trữ của mình dưới nhiều hình thức không giống nhau, hai loại kho lưu giữ trong Github là Local Repository và Remote Repository.
* Có thể chứa tối đa 2GB free github

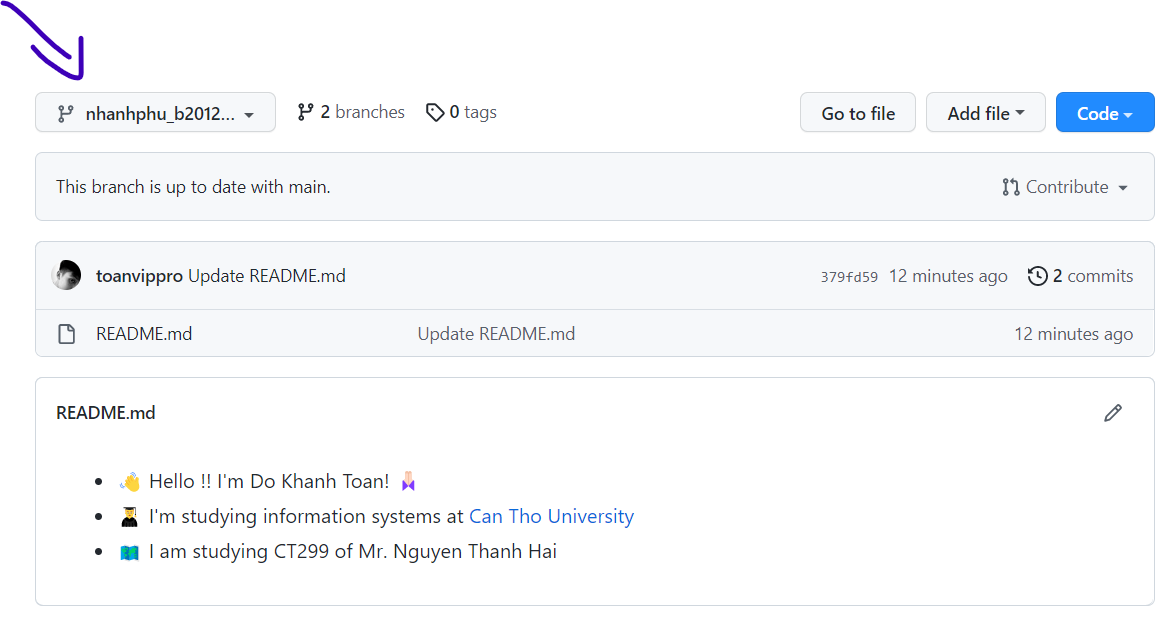


https://github.com/toanvippro/B2012046

1. Branch trong git là gì? Bạn hãy tạo 1 branch tên “nhanhphu-<mssv của bạn>”

=>

* Branch là cái dùng để phân nhánh và ghi lại luồng của lịch sử. Branch đã phân nhánh sẽ không ảnh hưởng đến branch khác nên có thể tiến hành nhiều thay đổi đồng thời trong cùng 1 repository.



1. Hãy cho biết hành động commit, push, pull trong git là gì?

COMMIT:

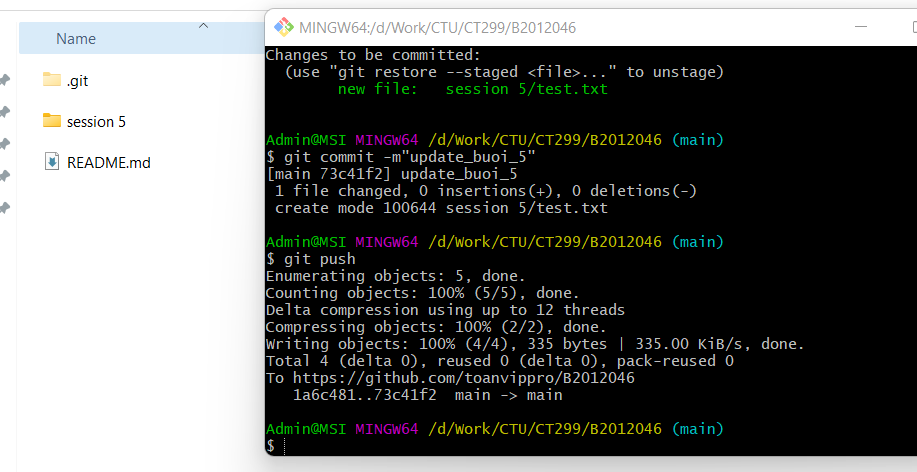
* Git Commit là một lệnh cam kết trong Git, được dùng để lưu lại những thay đổi trong repository bằng các ảnh chụp nhanh (snapshot). Các ảnh snapshot đã cam kết được coi là phiên bản “an toàn” của một dự án vì Git luôn hỏi trước khi thay đổi chúng. Trước khi bắt đầu lệnh Git Commit, lệnh Git Add sẽ được chạy để thúc đẩy các thay đổi của dự án mà sau đó nó sẽ được lưu trữ trong một cam kết (commit).
* Commit là những bức ảnh chụp nhanh của dự án. Mọi commit đều được ghi lại trong nhánh chính của repository. Hai commit khác nhau sẽ không bao giờ ghi đè vì mỗi commit có ID riêng. ID của mỗi commit là một số mật mã được tạo bởi thuật toán SHA (Secure Hash Algorithm).
* git commit -a: Lệnh commit này cung cấp tùy chọn -a để chỉ định một số cam kết. Nó được sử dụng để xác nhận các ảnh chụp nhanh của tất cả các thay đổi. Tùy chọn này chỉ áp dụng với các tệp đã được thêm vào Git. Nó sẽ không commit các tệp mới được tạo.
* git commit: Thao tác này sẽ khởi chạy một trình soạn thảo văn bản nhắc bạn về một thông báo cam kết. Sau khi bạn đã nhập thư, hãy lưu tệp và đóng trình chỉnh sửa để tạo cam kết thực tế.
* git commit -m “commit message”: Tùy chọn cho phép bạn viết thông báo cam kết trên dòng lệnh. Git Commit sẽ mở trình soạn thảo văn bản và nhắc nhập thông báo cam kết.
* git commit -am “commit message”: Kết hợp tùy chọn -a và -m để tạo lệnh commit cho tất cả các thay đổi theo giai đoạn. Với lệnh này, người dùng sẽ nhận thông báo cam kết nội tuyến.
* git commit –amend: Đây là một tùy chọn cho phép chỉnh sửa cam kết cuối cùng. Các thay đổi theo giai đoạn được thêm vào cam kết trước đó. Lệnh này mở trình soạn thảo văn bản đã cấu hình của hệ thống và thay đổi thông báo cam kết đã chỉ định trước đó.

PUSH:

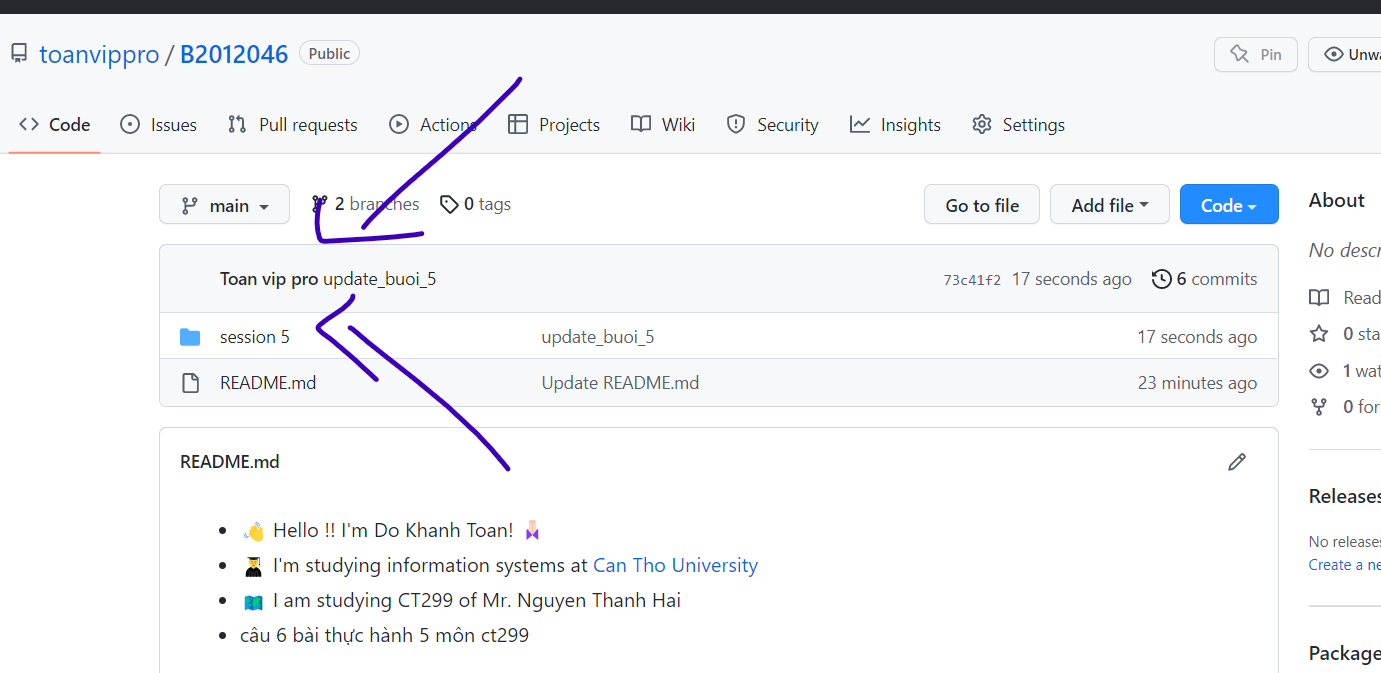
* Lệnh git push được sử dụng để đẩy các commit mới ở máy trạm (local repo) lên server (remote repo). Nguồn để đẩy lên là nhánh mà con trỏ HEAD đang trỏ tới (nhánh làm việc).
* Đích mà nó đẩy lên (ghi vào nhánh nào) có thể chỉ định trong tùy chọn của lệnh, tuy nhiên cũng không cần chỉ định nếu có thiết lập liên hệ giữa nguồn và đích trước. Lệnh git push cũng có thể xóa đi các nhánh của remote
* Lệnh git push được sử dụng để đẩy các commit mới ở máy trạm (local repo) lên server (remote repo). Nguồn để đẩy lên là nhánh mà con trỏ HEAD đang trỏ tới (nhánh làm việc).
* Đích mà nó đẩy lên (ghi vào nhánh nào) có thể chỉ định trong tùy chọn của lệnh, tuy nhiên cũng không cần chỉ định nếu có thiết lập liên hệ giữa nguồn và đích trước. Lệnh git push cũng có thể xóa đi các nhánh của remote

1. Hãy thực hiện hành động upload tập tin thực hành buổi 5 **lần trước** lên github và commit.

=>



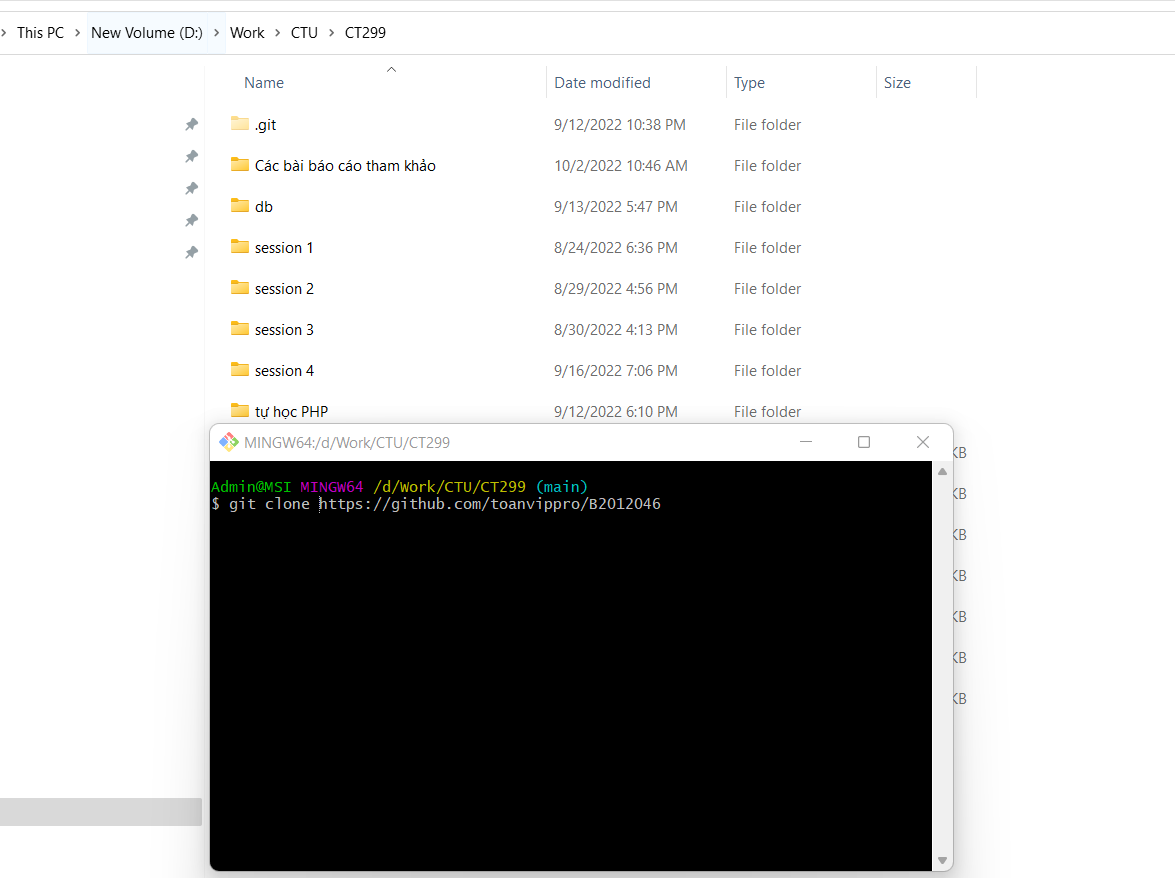
KẾT QUẢ:

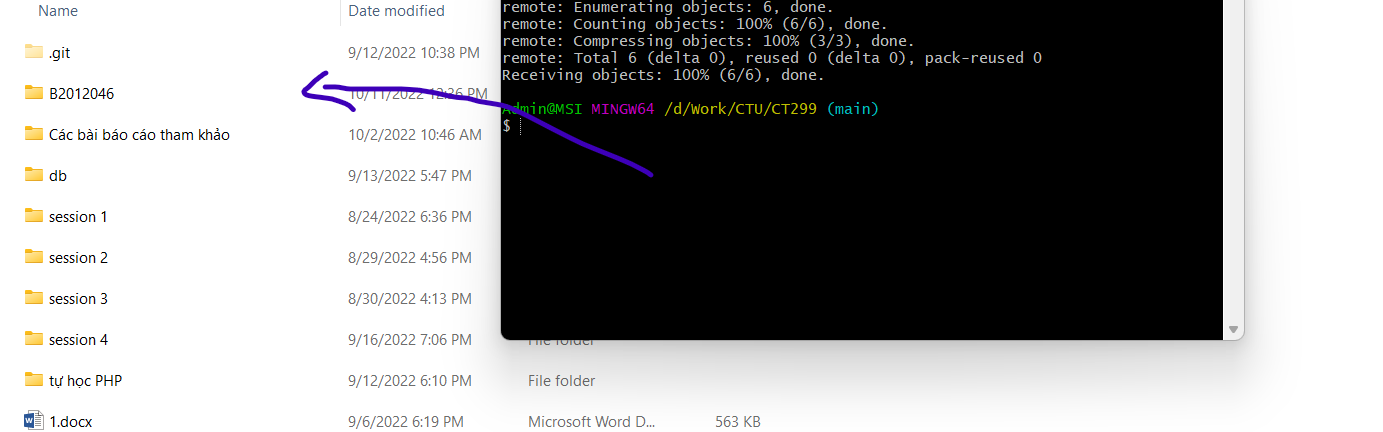


1. Thực hiện lệnh git clone … để download code từ repository bạn tạo ở câu số 3.



=>





=> Clone về thành công

1. Chép 1 tập tin thực hành buổi 4 vào thư mục vừa download xuống và thực hiện lệnh

git add .

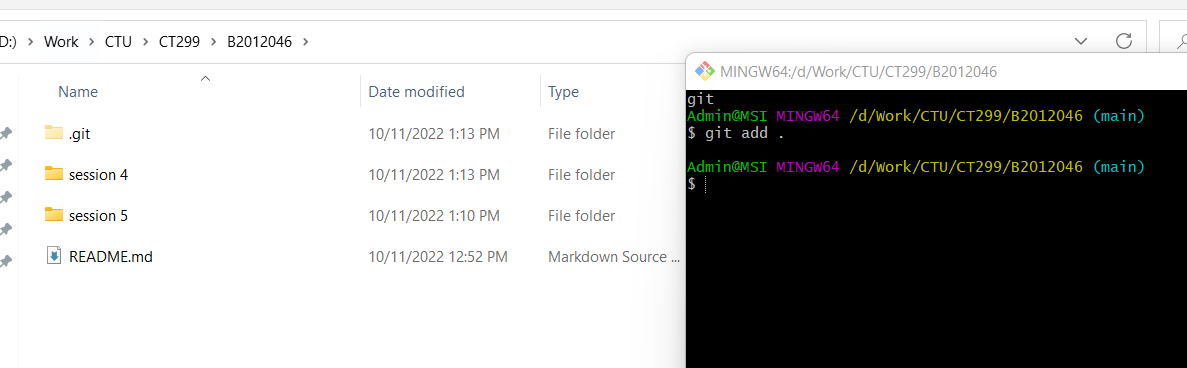
git commit -m “upload du lieu”

git push

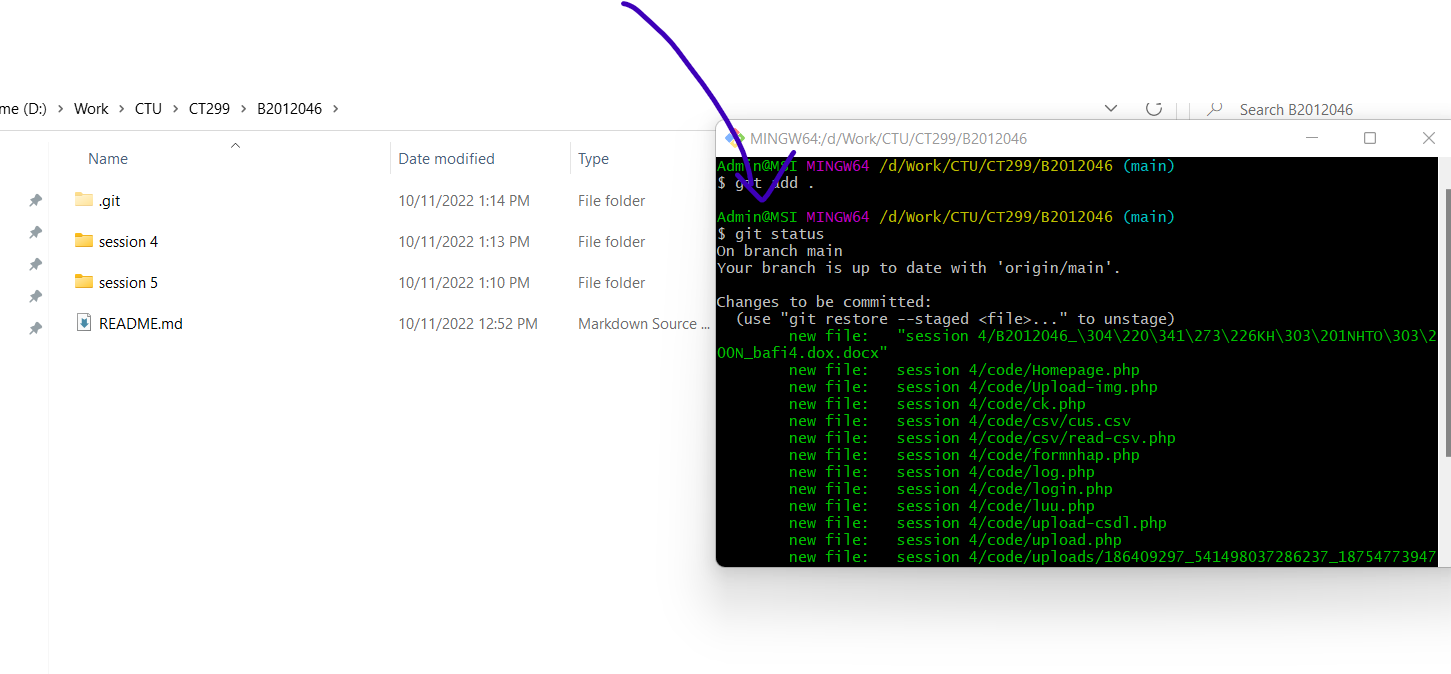
Hãy cho biết ý nghĩa 3 lệnh trên. Và quan sát kết quả trên trang github của bạn đã tạo ở câu 3 và nhận xét

=>

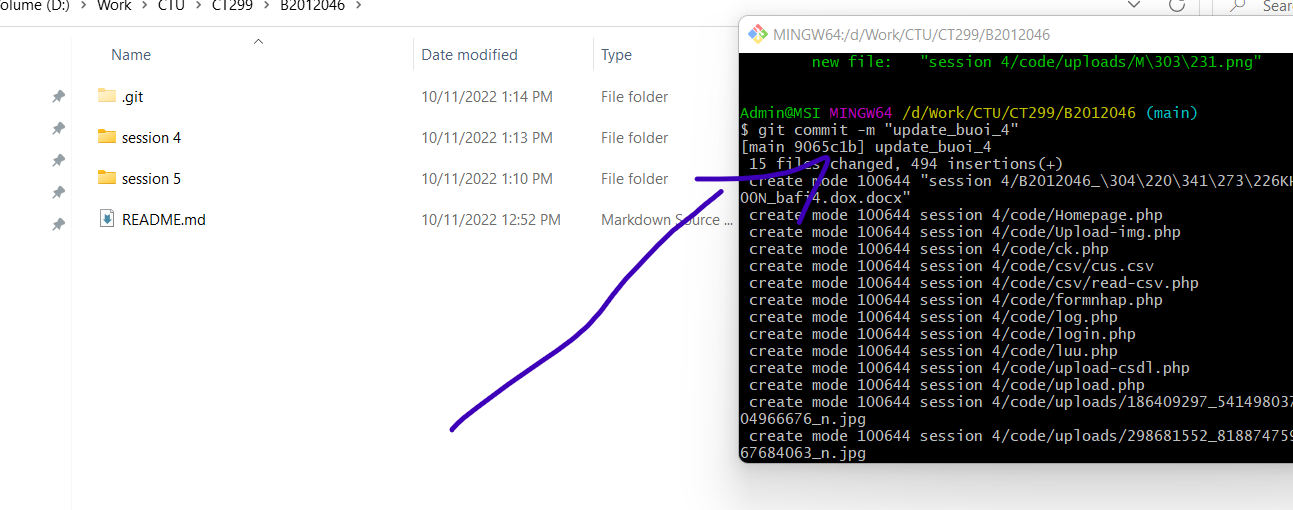
1. Git add .

.

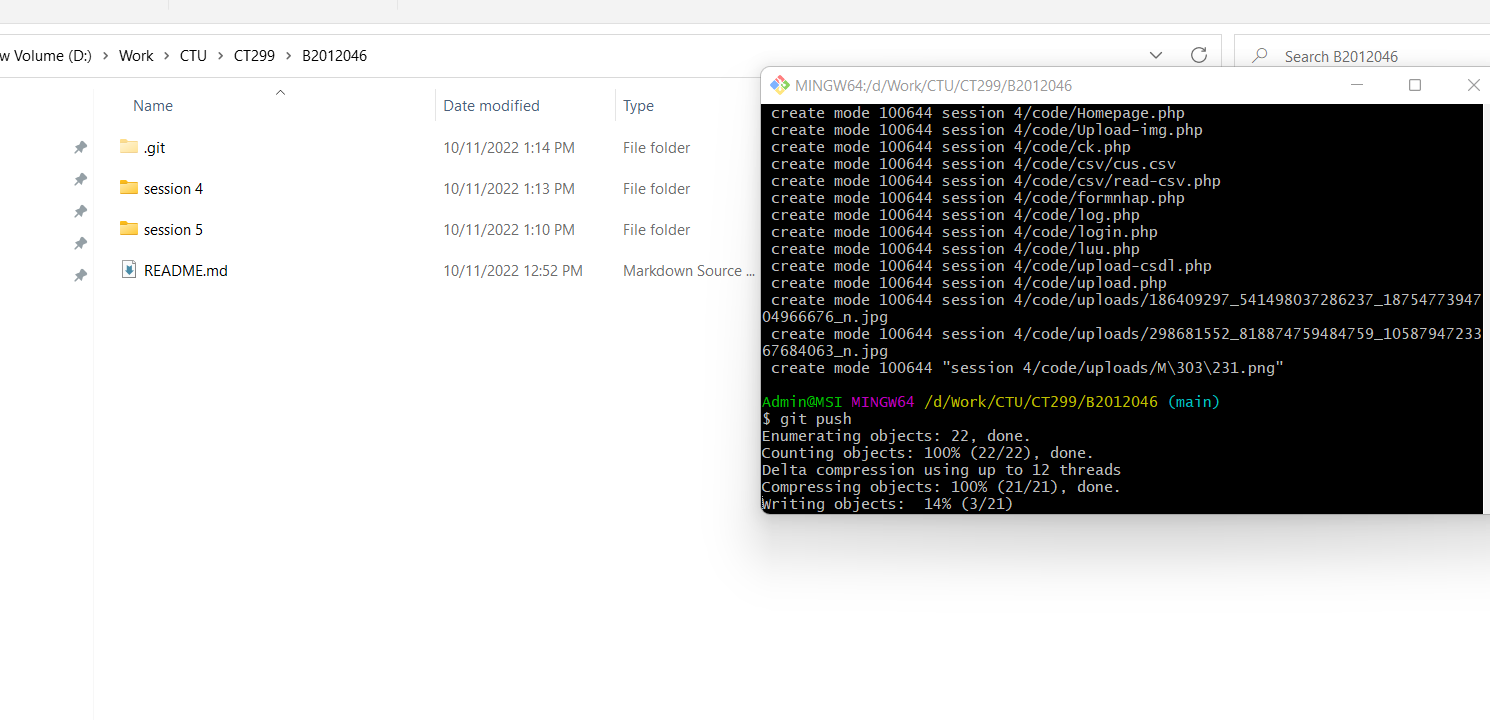
1. Git status



1. Git commit -m”update”

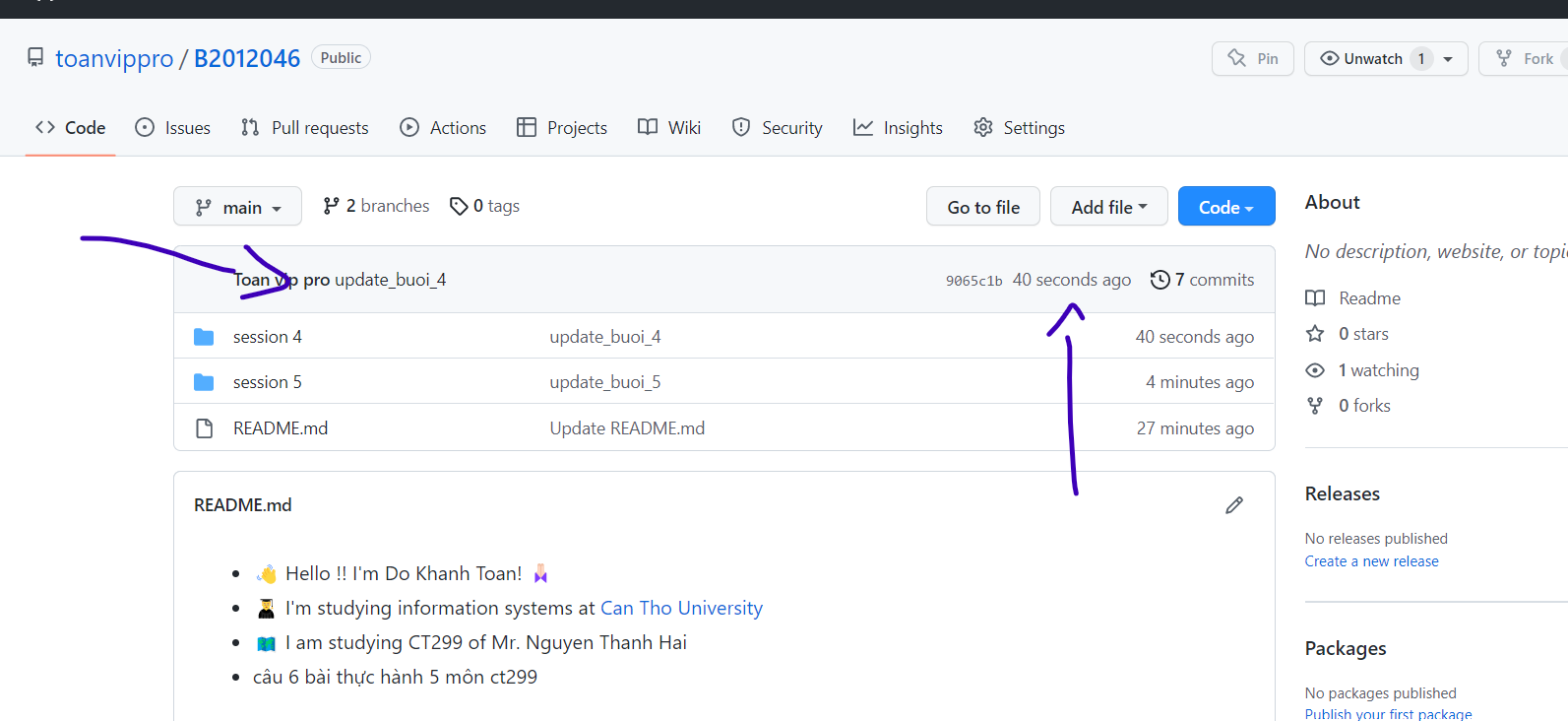


1. Git push



=> push lên github thành công

KẾT QUẢ:



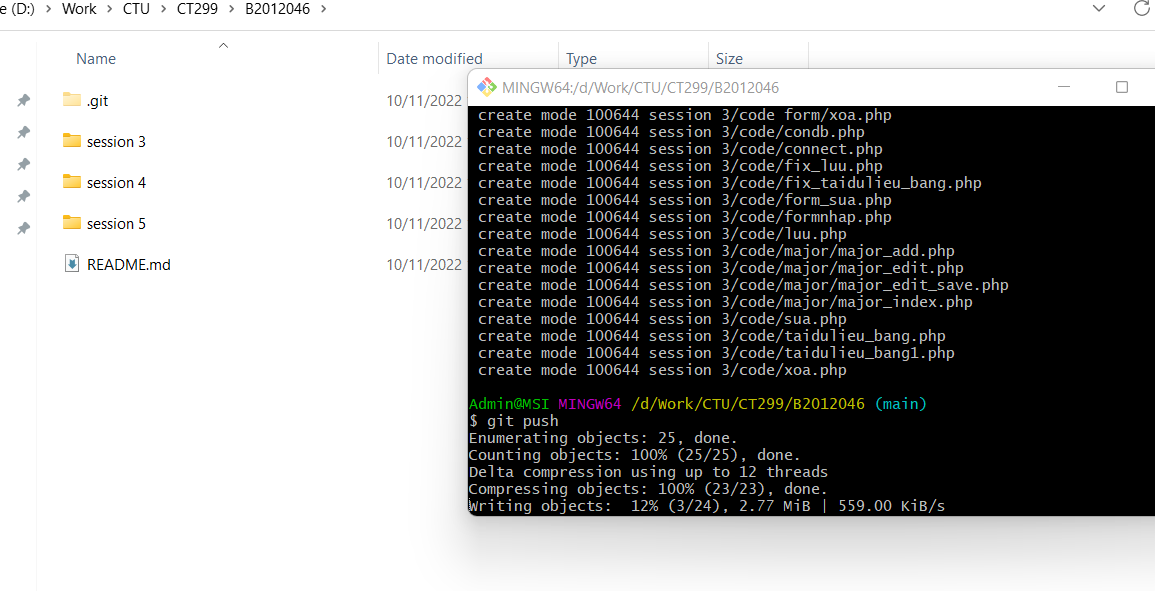
1. Upload tập tin thực hành buổi 3 lên trang github của bạn và thực hiện lệnh

git pull

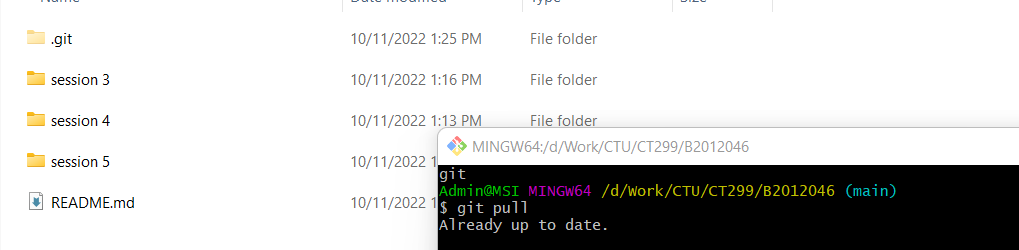
Quan sát kết quả đạt được và nhận xét.

=>

1. Push buổi 3 lên github:



1. Bắt đầu rỏ lệnh git bull

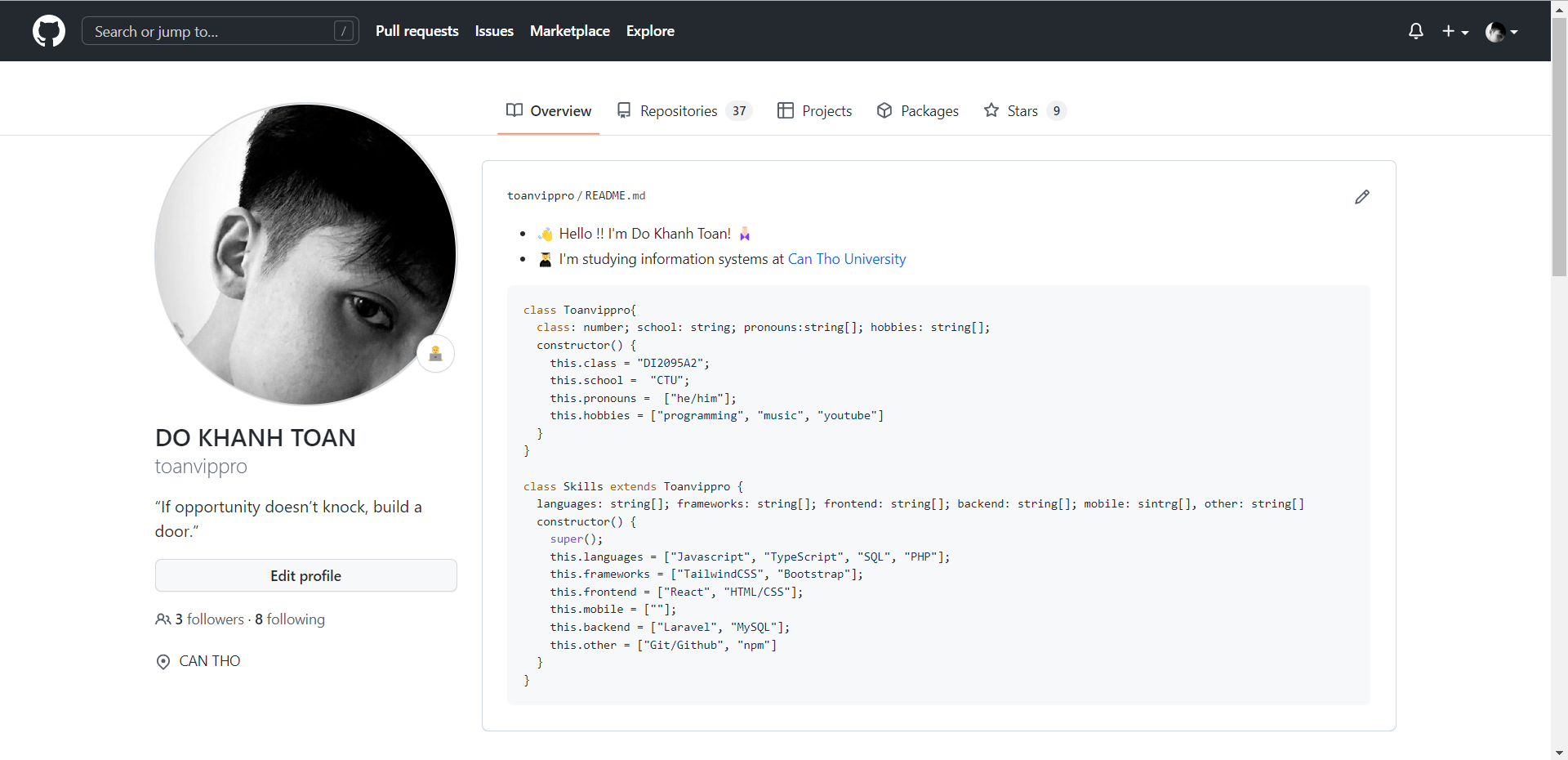


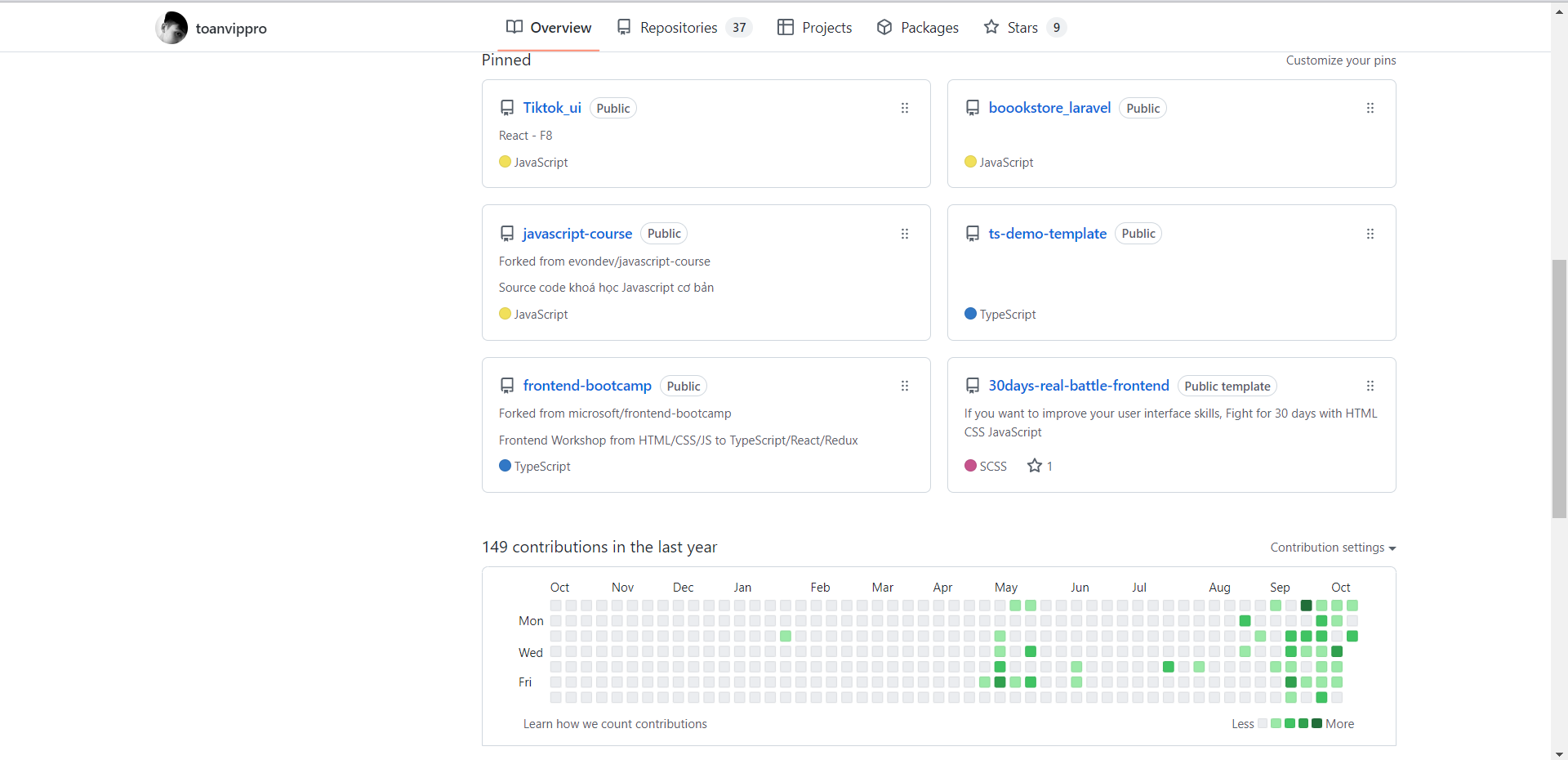
=> lệnh git pull có nghĩa là đồng bộ dữ liệu từ trên github và dữ liệu của máy, ở đây nó báo Already up to date có nghĩa là lúc bây h dữ liệu từ máy và từ github đã đồng bộ với nhau nên không có gì thay đổi

1. Bạn hãy add các contributor là những bạn sẽ làm cùng nhóm với bạn



1. Tham khảo ở: <https://www.makeareadme.com/>, hãy trang trí trang readme của bạn sao cho đẹp với các mô tả thông tin bài tập nhóm.





Chú ý:

* Các bạn nộp file word: Đặt tên file: <mssv><hoten><bai5>.docx nộp lên classroom, kèm với các file code đã tạo (**ngoại trừ file word trả lời câu hỏi, các file còn lại các bạn nén vào file .zip**).
* Mỗi câu các bạn trả lời bằng hình/trả lời dạng text tùy vào yêu cầu của câu hỏi, và ghi rõ số thứ tự câu hỏi mà bạn trả lời.
* Các bài phát hiện copy từ các bài của bạn khác sẽ không được tính điểm.
* *Chú ý: Các câu trả lời có tham khảo trên Internet phải trích dẫn link*

1. https://docs.github.com/en/repositories/creating-and-managing-repositories/about-repositories [↑](#footnote-ref-0)
2. https://docs.github.com/en/get-started/quickstart/hello-world [↑](#footnote-ref-1)