

1.1 GitHub là gì?

1.1.1 Git là gì?

Trước tiên, phải biết rõ Git là gì, vì nó là trái tim của GitHub. Git: một hệ quản trị phiên bản được phát triển bởi Linus Torvalds (tên rất quen phải không, người tạo ra Linux đó).

1.1.2 Hub là gì?

Nếu Git là trái tim thì Hub lại là phần hồn của GitHub. Hub: nơi biến những dòng lệnh, Git, thành một mạng xã hội khổng lồ cho lập trình viên.

Bên cạnh đóng góp vào những dự án chính, GitHub cho phép người dùng tương tác theo kiểu mạng xã hội. Bạn có thể theo dõi, và xem những người bạn thích làm gì, họ đang kết nối với ai,...

1.1.3 Tóm lại, GitHub là gì?

GitHub: sự kết hợp giữa 2 từ, Git – hệ thống quản lý dự án, phiên bản code và Hub – mạng xã hội cho lập trình viên. GitHub được sử dụng chủ yếu cho dự án có nhiều người cùng hợp tác và cần giám sát toàn bộ thay đổi của dự án, cũng như khả năng khôi phục code khi cần thiết. Khi sử dụng GitHub thì ngoài công việc chính như tạo Branch, tạo Pull Request và Fork một Repository, bạn có thể theo dõi, tương tác với người khác như một mạng xã hội thông thường.

1.2 Một vài khái niệm của Git

- **Git**: là prefix của các lệnh được sử dụng dưới CLI
- **Branch**: được hiểu như là nhánh, thể hiện sự phân chia các version khi 2 version đó có sự sai khác nhất định và 2 version đều có sự khác nhau.
- **Commit**: là một điểm trên cây công việc (Work Tree) hay gọi là cây phát triển công việc
- **Clone**: được gọi là nhân bản, hay thực hiện nhân bản. Sử dụng để clone các project, repository trên các hệ thống chạy trên cơ sở là git, ví dụ như: bitbucket, github, gitlab, cor(1 sản phẩm mã nguồn mở cho phép

người dùng tự tạo git server cho riêng mình trên vps, server),... Việc clone này sẽ sao chép repository tại commit mình mong muốn, dùng để tiếp tục phát triển. Thao tác này sẽ tải toàn bộ mã nguồn, dữ liệu về máy tính của bạn.

- **Folk**: Folk là thao tác thực hiện sao chép repository của chủ sở hữu khác về git account của mình. sử dụng và đối xử như 1 repository do mình tạo ra.
- **Repository**: Kho quản lý dữ liệu, là nơi lưu trữ các dữ liệu, mã nguồn của project.
- **Tag**: sử dụng để đánh dấu một commit khi bạn có quá nhiều commit tới mức không thể kiểm soát được.
- **Remote**: sử dụng để điều khiển các nhánh từ một repository trên git server, đối xử với các nhánh trên remote tương tự như đối xử với các nhánh trên local
- **Diff**: So sánh sự sai khác giữa phiên bản hiện tại với phiên bản muốn so sánh, nó sẽ thể hiện các sự khác nhau
- **Gitignore**: file mặc định của git sử dụng để loại bỏ (ignore) các thư mục, file mà mình không muốn push lên git server
- **Github Package Registry**: cái package registry này cho phép lập trình viên duy trì distribution registries của họ, bao gồm npm, docker, maven, nuget và Ruby gems.
- Đối với hệ thống kiểm soát version phân tán như Git, nếu bạn muốn thực hiện thay đổi cho một dự án nào đó, bạn phải sao chép (clone) toàn bộ kho lưu trữ về local. Bạn thực hiện các thay đổi trên bản sao local của mình, sau đó bạn có thể “push” đồng loạt các thay đổi đó lên server trung tâm. Điều này sẽ giúp các lập trình viên tạo ra được các thay đổi mà không cần kết nối với server. Việc kết nối với server chỉ cần khi “push” có các thay đổi lên server.

1.3 Cách hoạt động trên GitHub

- Quy tắc làm việc trên GitHub. Có hai nền tảng để làm việc với GitHub là local workflow và server workflow.
- Ở local workflow, bạn có thể thay đổi source code. Sau đó xác nhận các thay đổi đó tại server workflow. Lưu ý, bản xác nhận trên server phải chứa tính năng hoàn chỉnh hoặc có thể chạy được. Việc xác nhận các đoạn code chưa hoàn chỉnh, còn dở dang sẽ ảnh hưởng các thành viên khác khi họ cùng sử dụng kho lưu trữ.
- Từ kho lưu trữ của GitHub, bạn có thể tạo bản build cho production site bằng cách gửi các source code thay đổi lên đó. Khi sử dụng kho lưu trữ của server, người dùng sẽ phải cung cấp mã chứng nhận, hệ thống sẽ so sánh SSH key ở local của bạn và SSH key trên server tương ứng với account mà đã đăng ký trước đó.

1.4 Chức năng

1.4.1 Collaborative coding

- Codespaces

Code, kiểm tra và gỡ lỗi với môi trường phát triển đám mây. Có thể truy cập từ bất kỳ thiết bị nào.

- Pull requests

Cho phép những người đóng góp dễ dàng thông báo về những thay đổi đã đưa vào kho lưu trữ - với quyền truy cập giới hạn cho những người đóng góp được chỉ định.

- Team Discussions

Bắt đầu các cuộc trò chuyện nhóm hoặc riêng tư cho những người tham gia thảo luận bằng các bản cập nhật và liên kết từ mọi nơi.

- Team reviewers

Yêu cầu một nhóm trên GitHub xem xét pull requests. Các thành viên của nhóm sẽ nhận được thông báo cho biết yêu cầu đánh giá của họ.

1.4.2Automation and CI/CD

- Action

Tự động hóa tất cả các quy trình phát triển phần mềm. Viết các nhiệm vụ và kết hợp chúng để xây dựng, kiểm tra và triển khai nhanh hơn từ GitHub.

- GitHub Pages

Tạo và xuất bản các trang web về bản thân, tổ chức của hoặc dự án trực tiếp từ kho lưu trữ GitHub.

- Webhooks

Tích hợp và tự động hóa công việc cho kho lưu trữ, tổ chức hoặc ứng dụng.

- Hosted runners

Tự động tải lên đám mây với các môi trường Linux, Windows và MacOS theo yêu cầu để chạy quy trình công việc, được lưu trữ bởi GitHub.

1.4.3Security

- Private repos

Trong kho lưu trữ GitHub riêng tư chỉ bạn và những người bạn chia sẻ chúng mới có thể truy cập được.

- Code scanning

Tìm lỗi hỏng trong mã tùy chỉnh bằng cách sử dụng phân tích tĩnh. Ngăn chặn các lỗi hỏng mới được đưa vào bằng cách quét các pull requests

- Dependabot alerts

Nhận thông báo khi có lỗi hỏng bảo mật mới ảnh hưởng đến kho lưu trữ. GitHub phát hiện và cảnh báo người dùng về các yếu tố phụ thuộc dễ bị tấn công trong các kho lưu trữ công khai và riêng tư.

1.4.4Team administration

- Organizations

Thiết lập nhóm tài khoản người dùng sở hữu kho lưu trữ. Quản lý quyền truy cập trên cơ sở từng nhóm hoặc người dùng cá nhân.

- Invitations

Dễ dàng thêm thành viên GitHub vào kho lưu trữ của bạn bằng tên người dùng hoặc địa chỉ email GitHub và yêu cầu họ xác nhận quyền truy cập

- Audit Log API

Xem xét các hành động được thực hiện bởi các thành viên trong team. Theo dõi quyền truy cập, thay đổi quyền, thay đổi người dùng và các sự kiện khác.

- Notification restriction

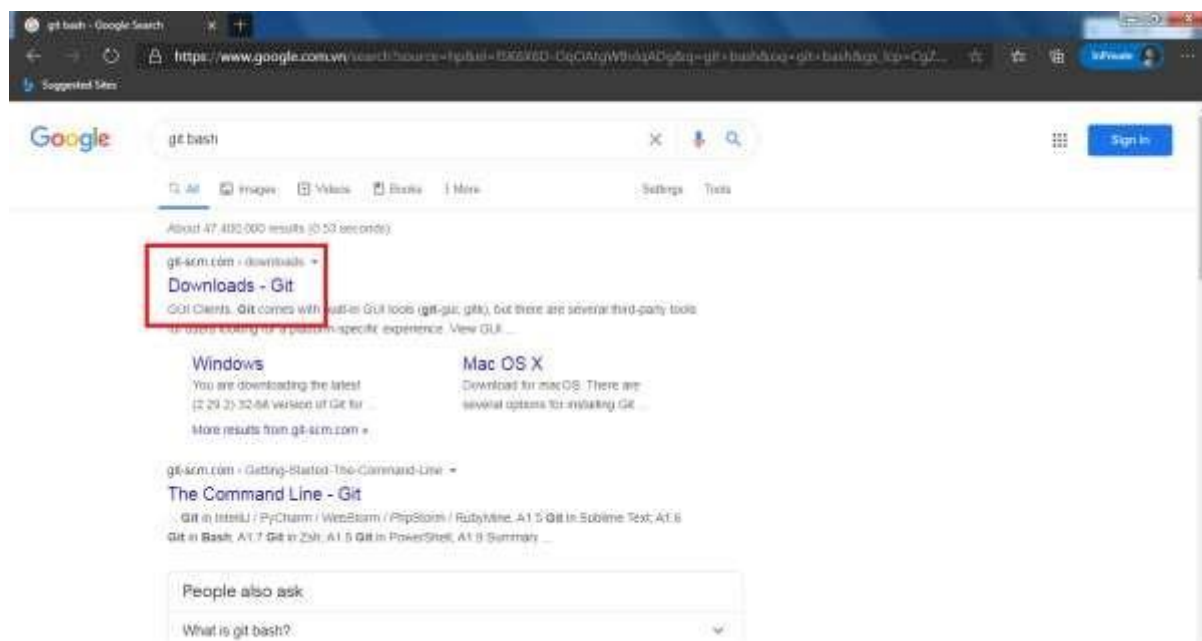
Bảo vệ thông tin về những gì nhóm của bạn đang làm việc bằng cách hạn chế thông báo qua email đối với các miền email đã được phê duyệt.

1.5 Hướng dẫn cách tải mã nguồn lên GitHub

1.5.1 GitBash

- Đầu tiên để sử dụng GitHub, ta cần có một công cụ để giao tiếp với GitHub là GitBash.

B1: Download GitBash.

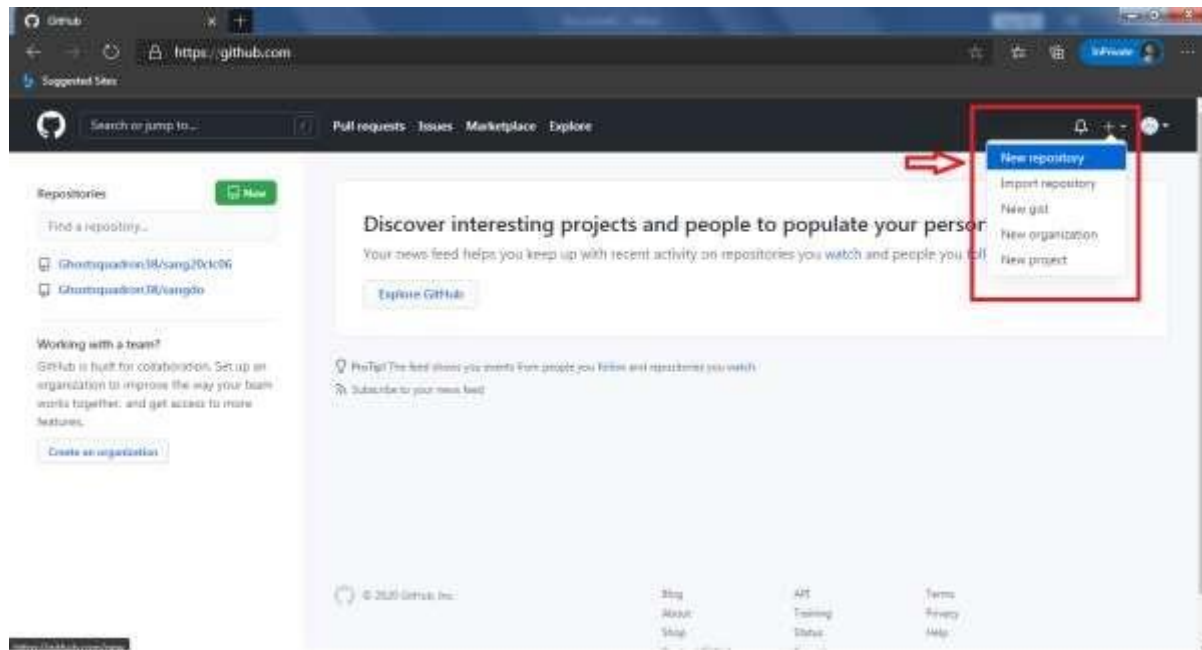


B2: Chọn phiên bản phù hợp với hệ điều hành của máy tính.

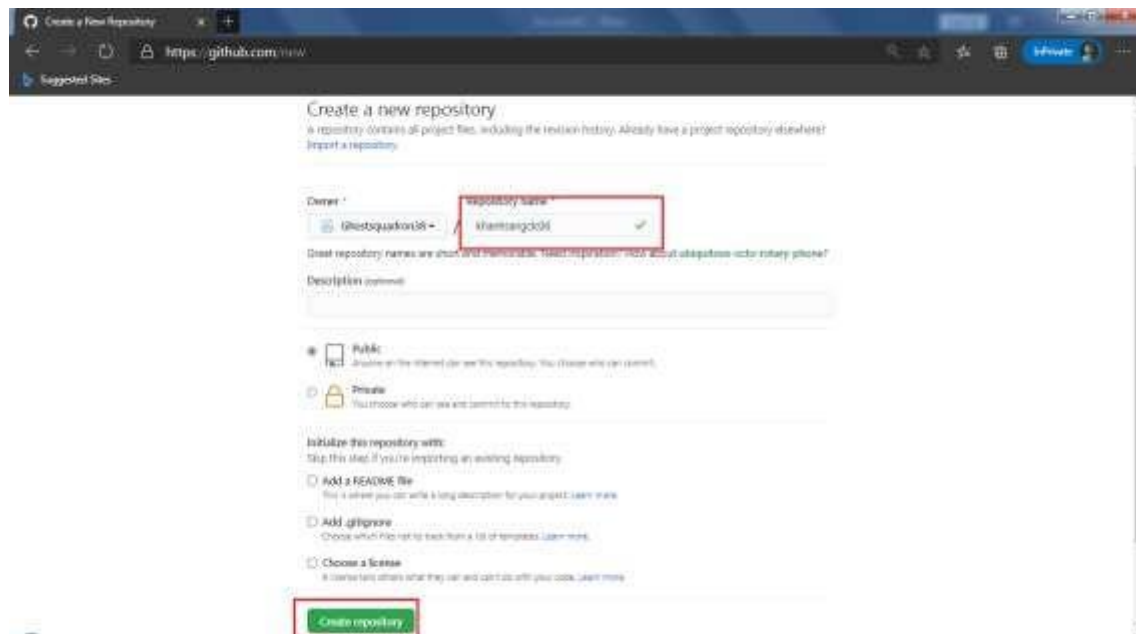


- Sau khi đã có GitBash ta vào GitHub để tạo một kho lưu trữ mới.

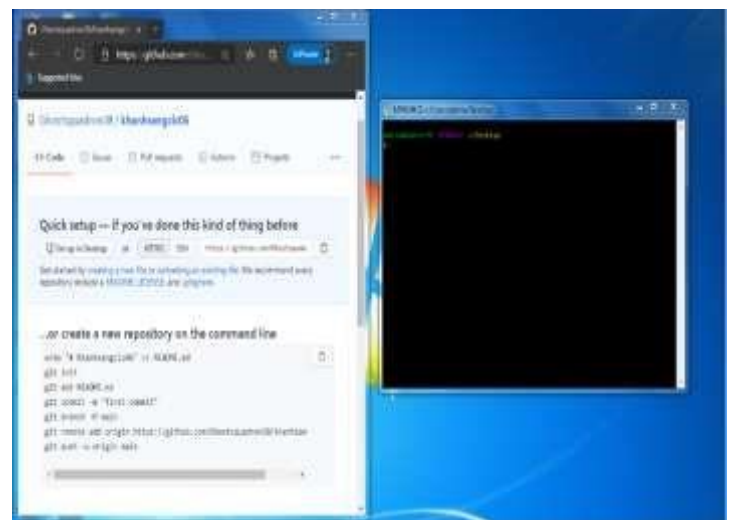
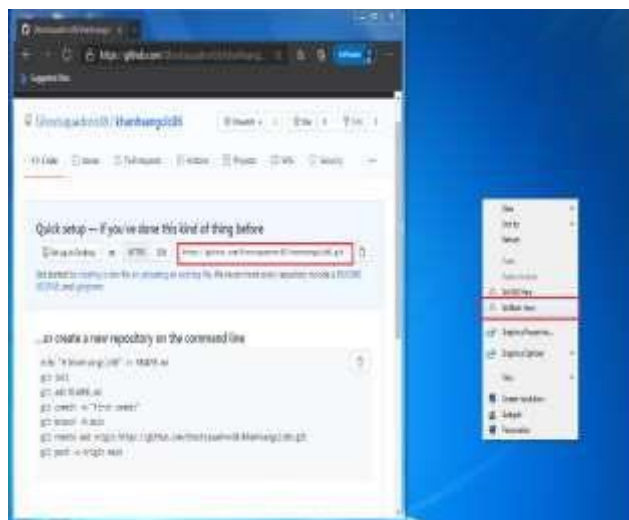
B1: Đăng nhập vào GitHub, tiếp chọn dấu “+” và chọn “New Repository”.



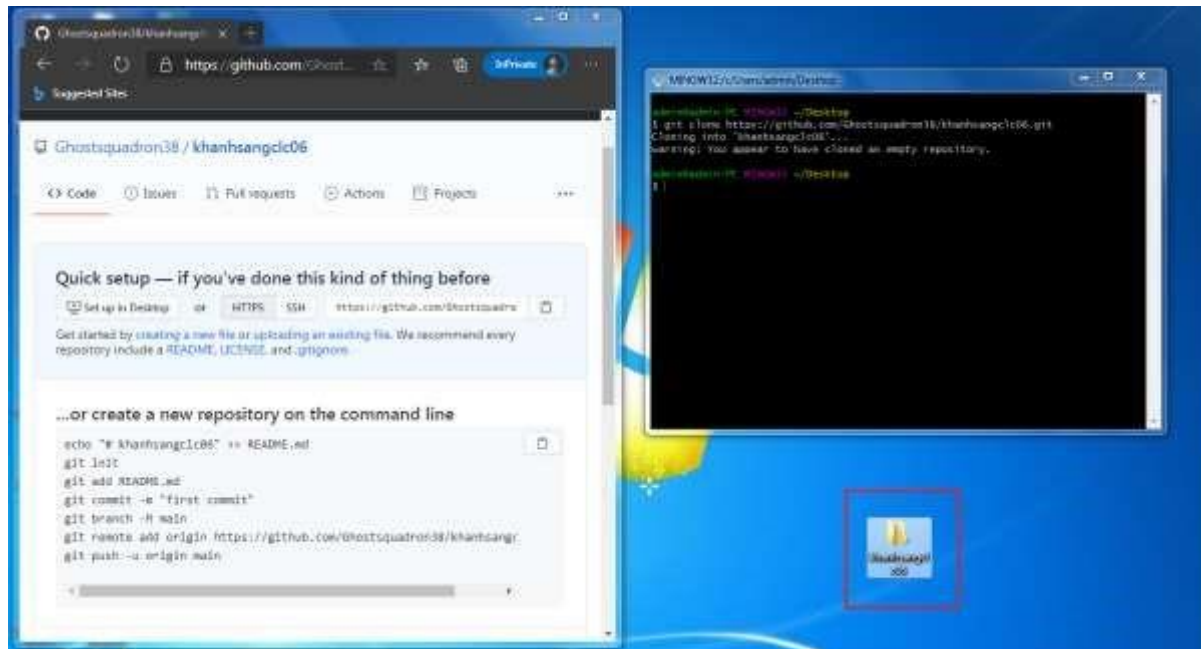
B2: Đặt tên cho kho lưu trữ (VD: khanhsangclc06) và chọn Create repository



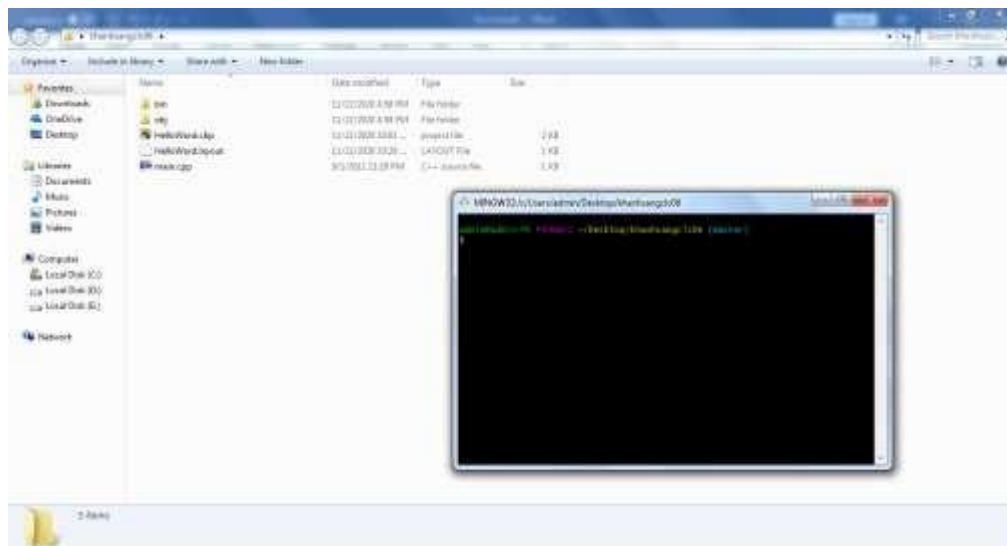
B3: Copy đoạn link ở trang GitHub. Ở ngoài màn hình Desktop, nhấp chuột phải chọn “Git Bash Here”. Sau đó sẽ xuất hiện một Command line



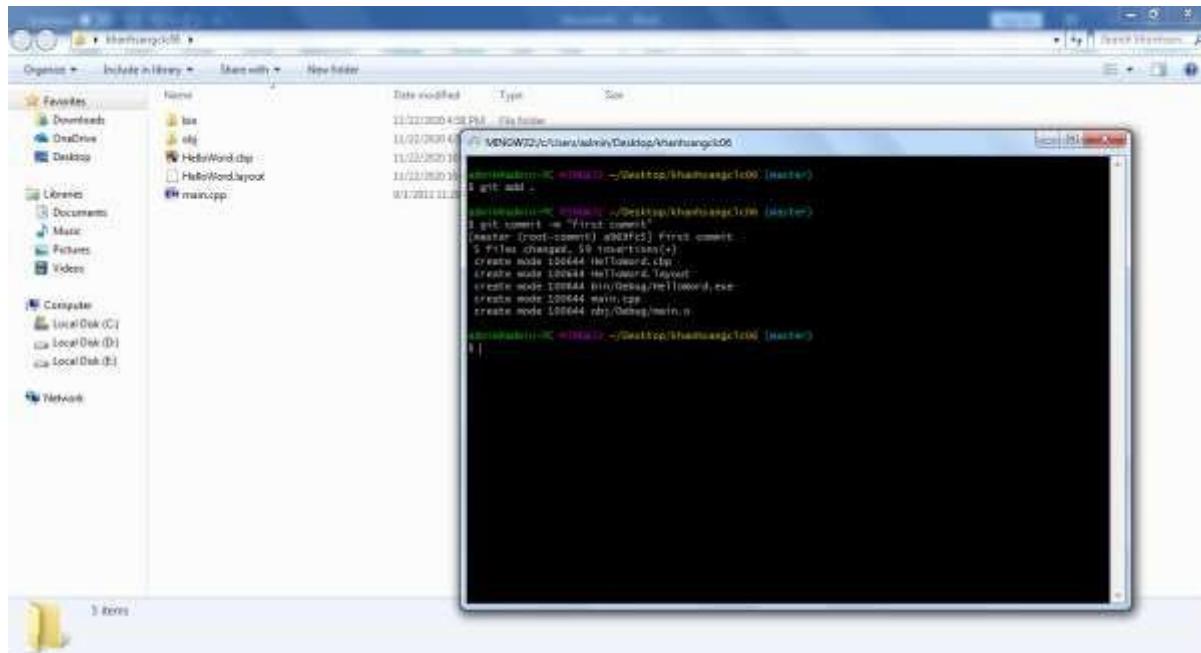
B4: Ở Command line, bấm “git clone <đường link vừa copy>”, sau đó nhấn Enter. Trên màn hình Desktop sẽ xuất hiện 1 file theo tên vừa đặt cho kho lưu trữ.



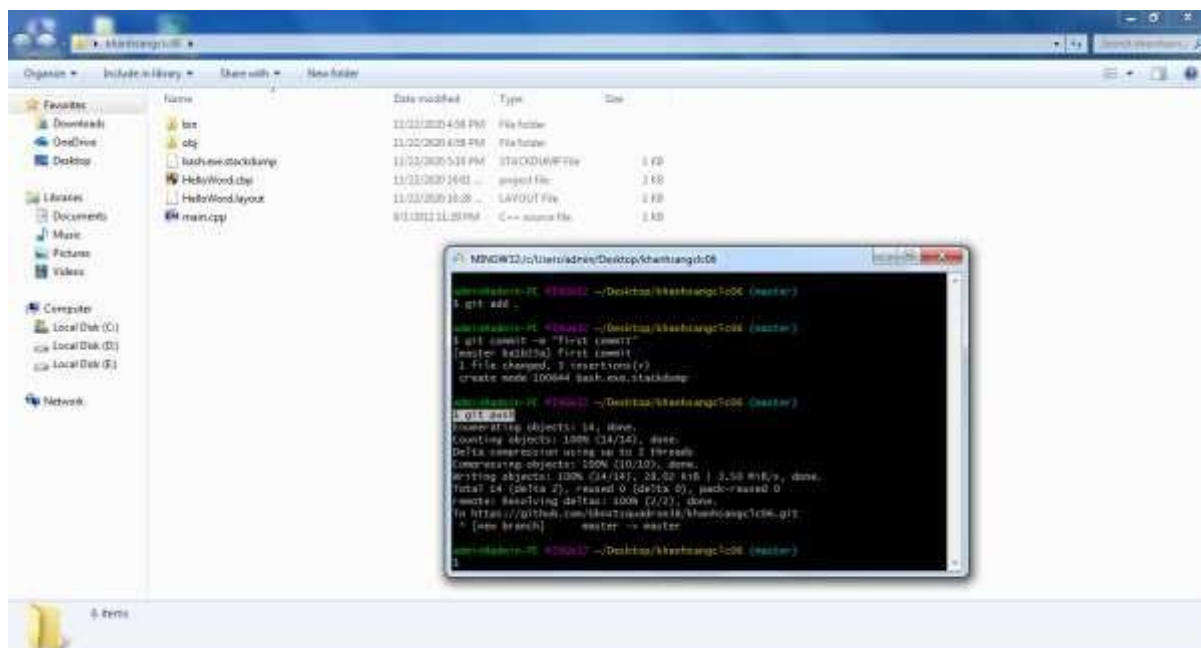
B5: Copy mã nguồn muốn tải lên GitHub vào file vừa được tạo, nhấp chuột phải và chọn “GitBash here”



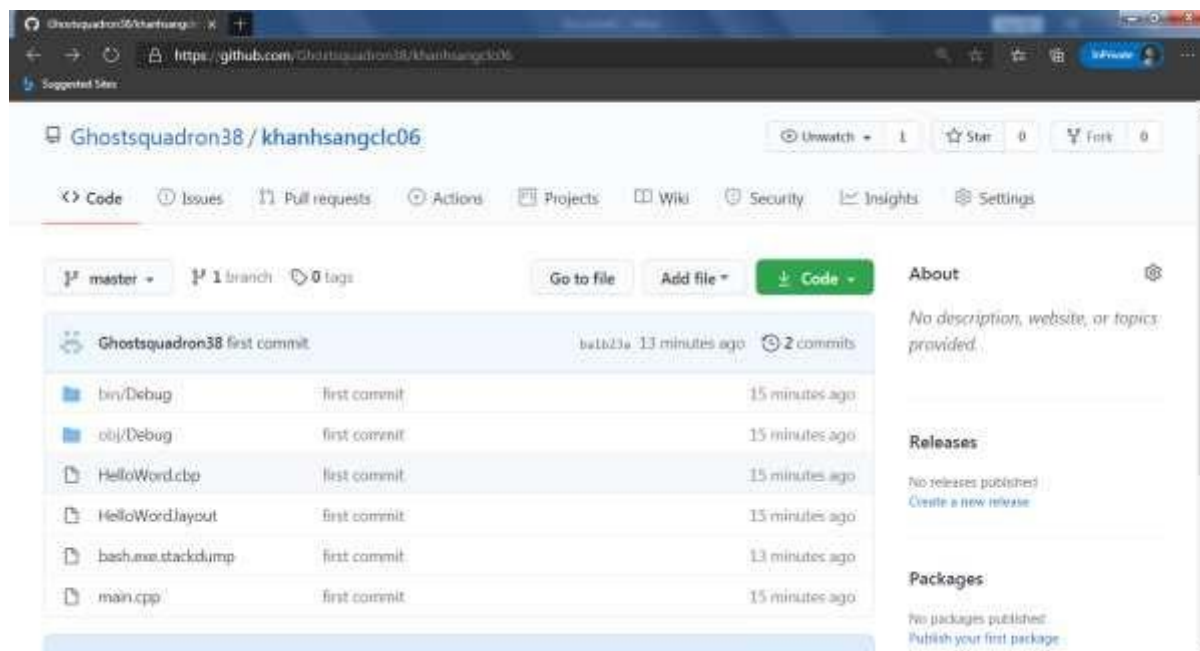
B6: Ở Command line, nhập “git add .” rồi Enter. Sau đó nhập “git commit -m "first commit"” và Enter lần nữa.



B7: Nhập “git push” để tải mã nguồn lên GitHub. Lưu ý ở bước này phải nhập lại tài khoản và mật khẩu ứng với tài khoản muốn tải lên kho lưu trữ.



Sau khi hoàn tất, mã nguồn đã hoàn toàn được up lên GitHub.



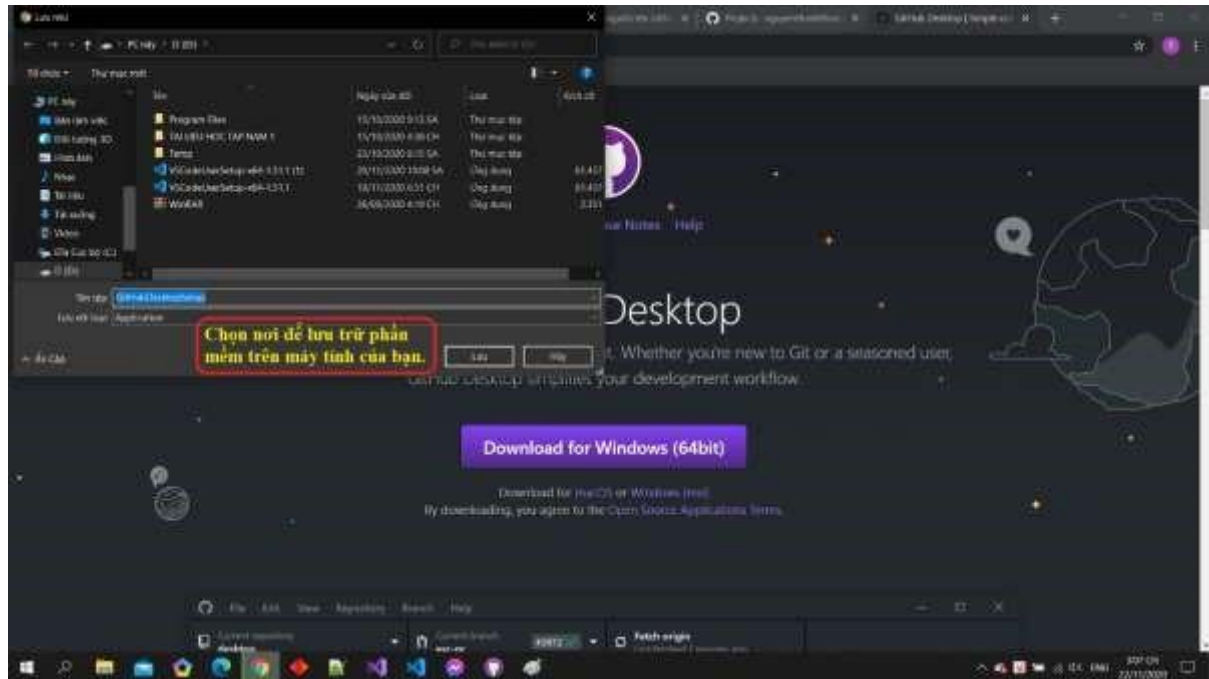
1.5.2 GitHub Desktop

B1: Cài đặt GitHub bằng cách tải xuống GitHub theo đường link sau:

<https://git-scm.com/downloads>



B2: Chọn nơi lưu trữ trên máy tính của bạn.

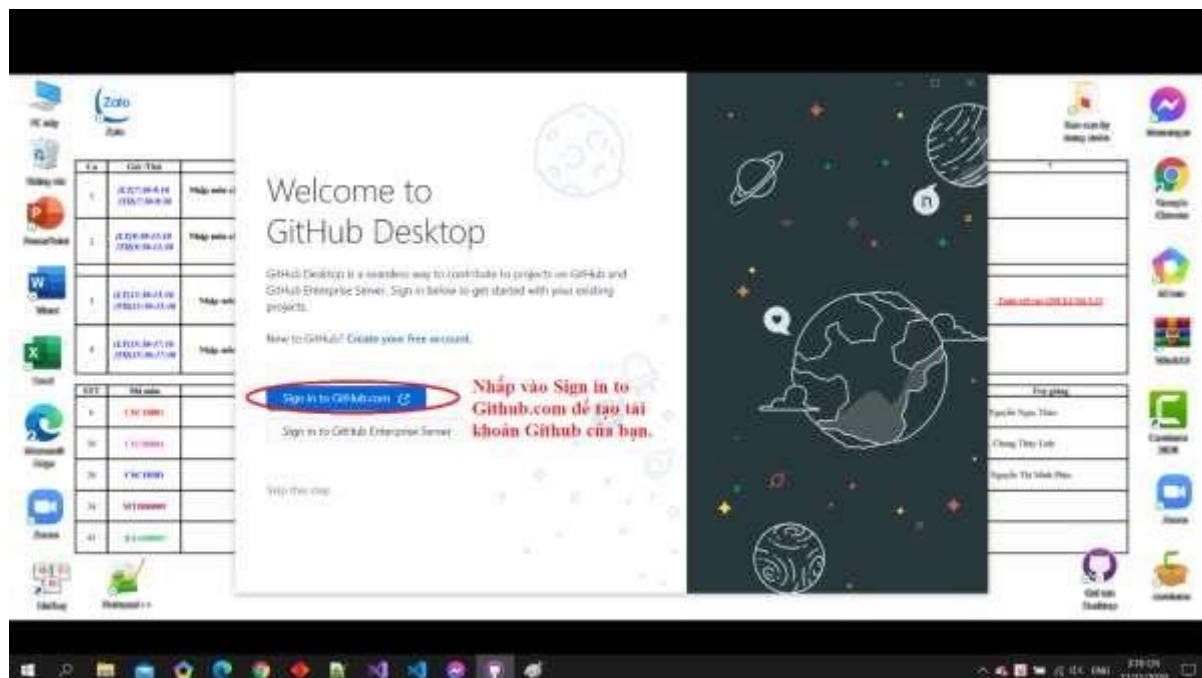


B3: Nháy chuột vào biểu

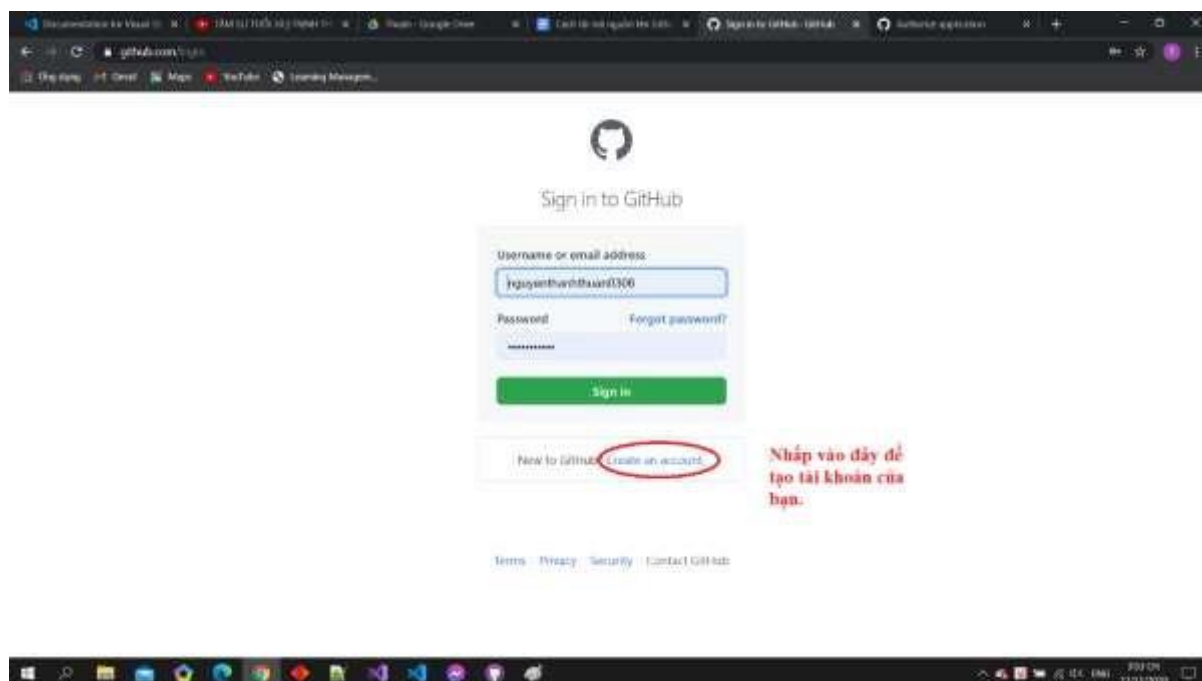


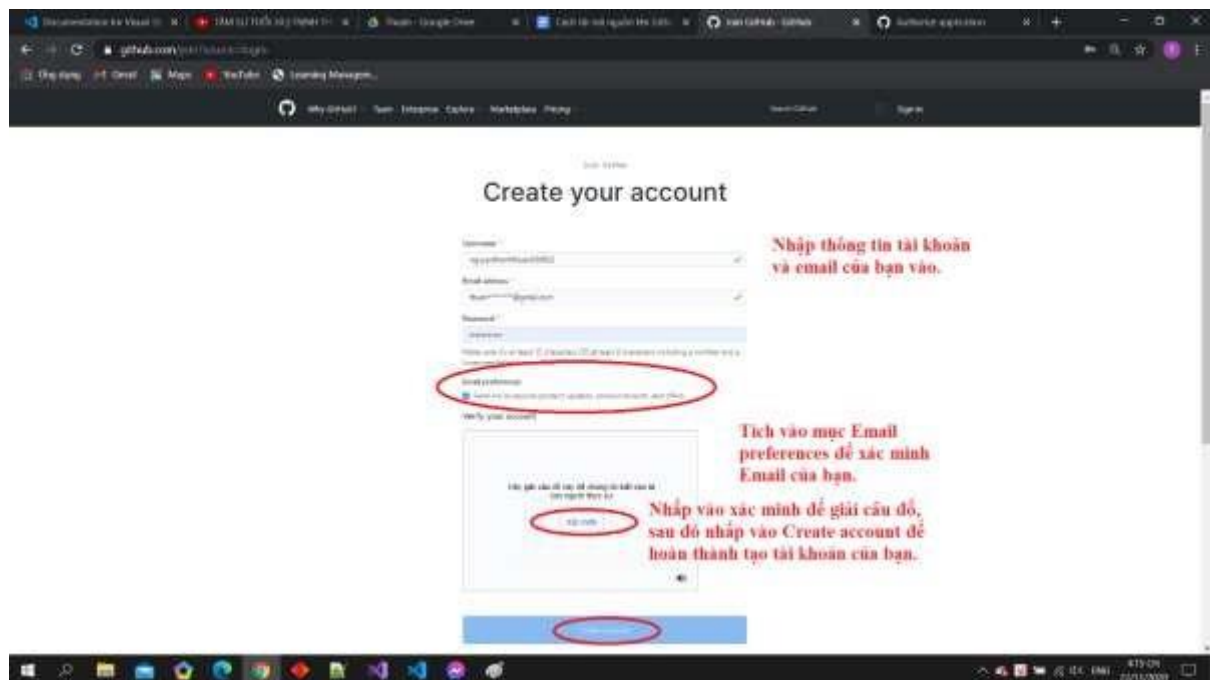
tượng trên màn hình desktop.

B4: Tạo tài khoản GitHub miễn phí

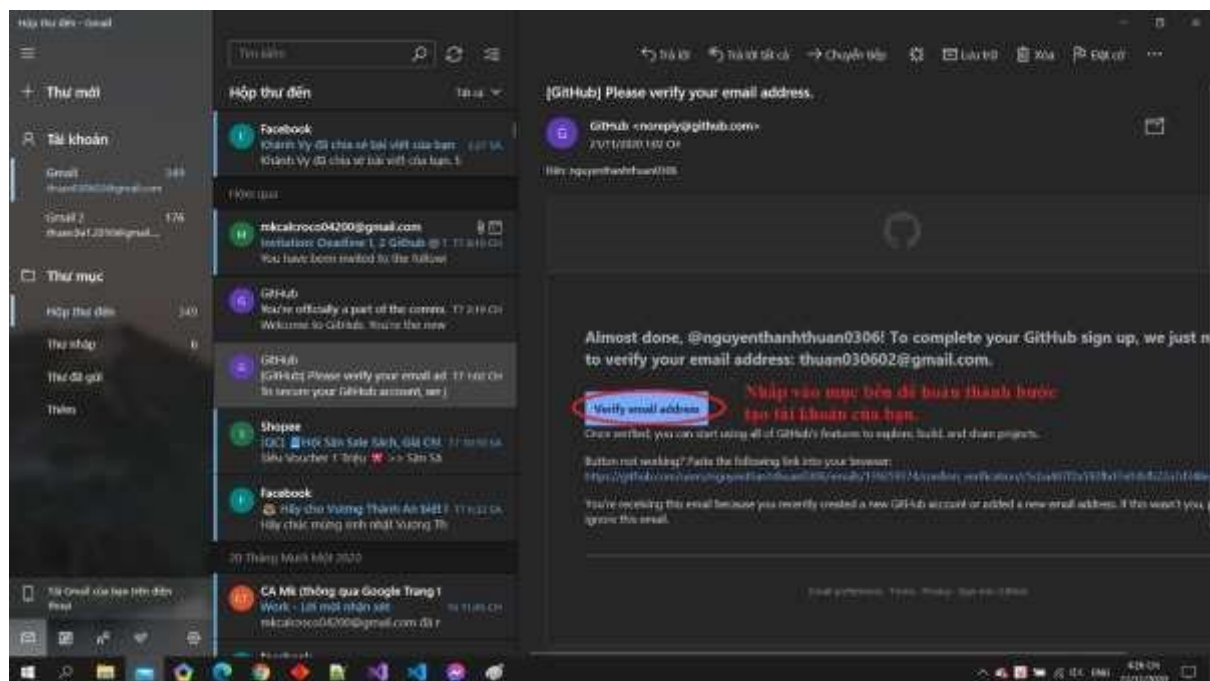


B5: Điền các thông tin cá nhân và nhấn Create account

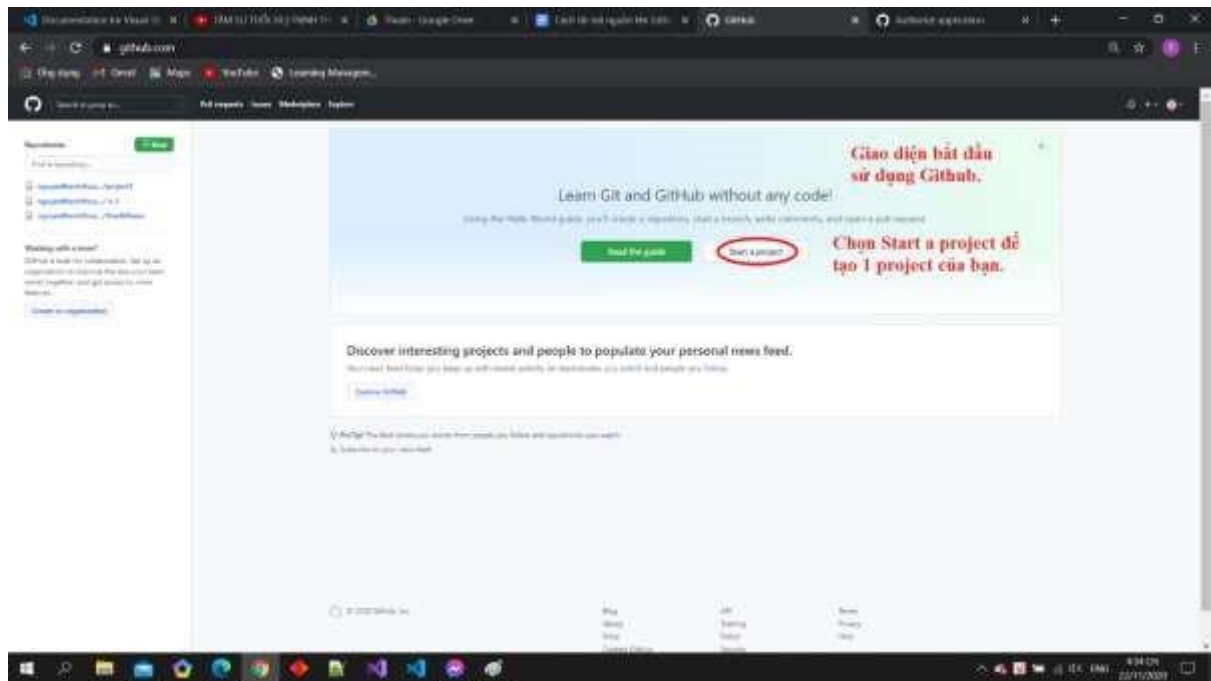




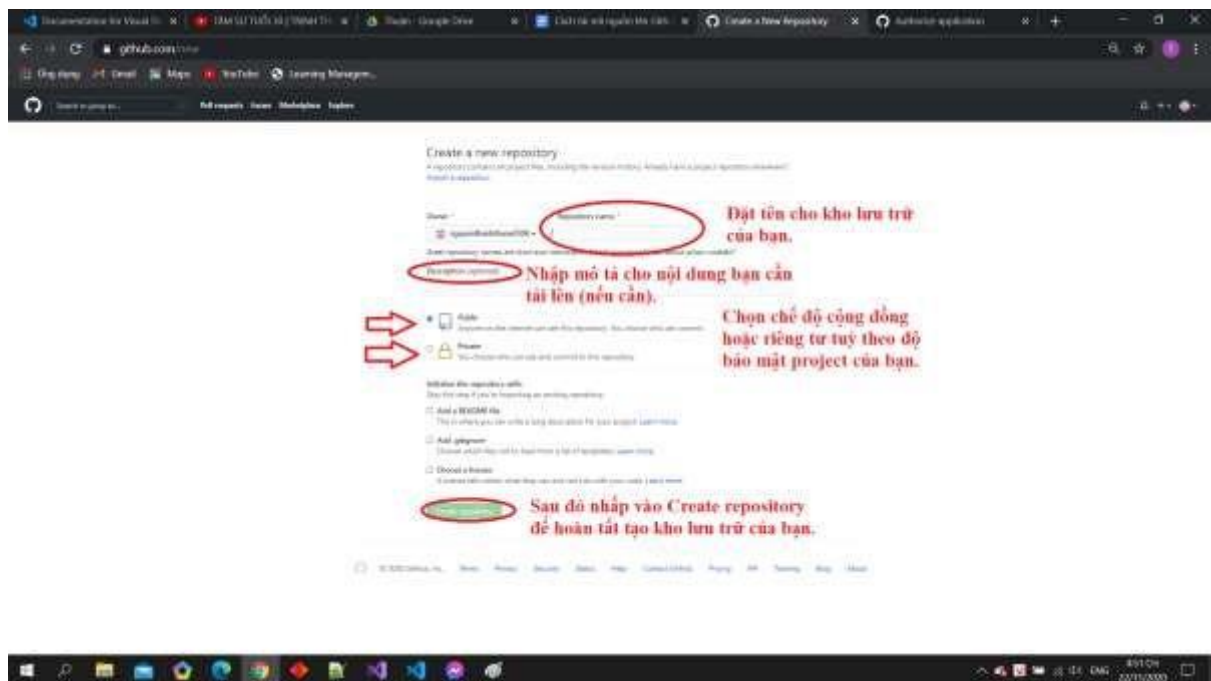
B6: Xác nhận email được gửi từ GitHub



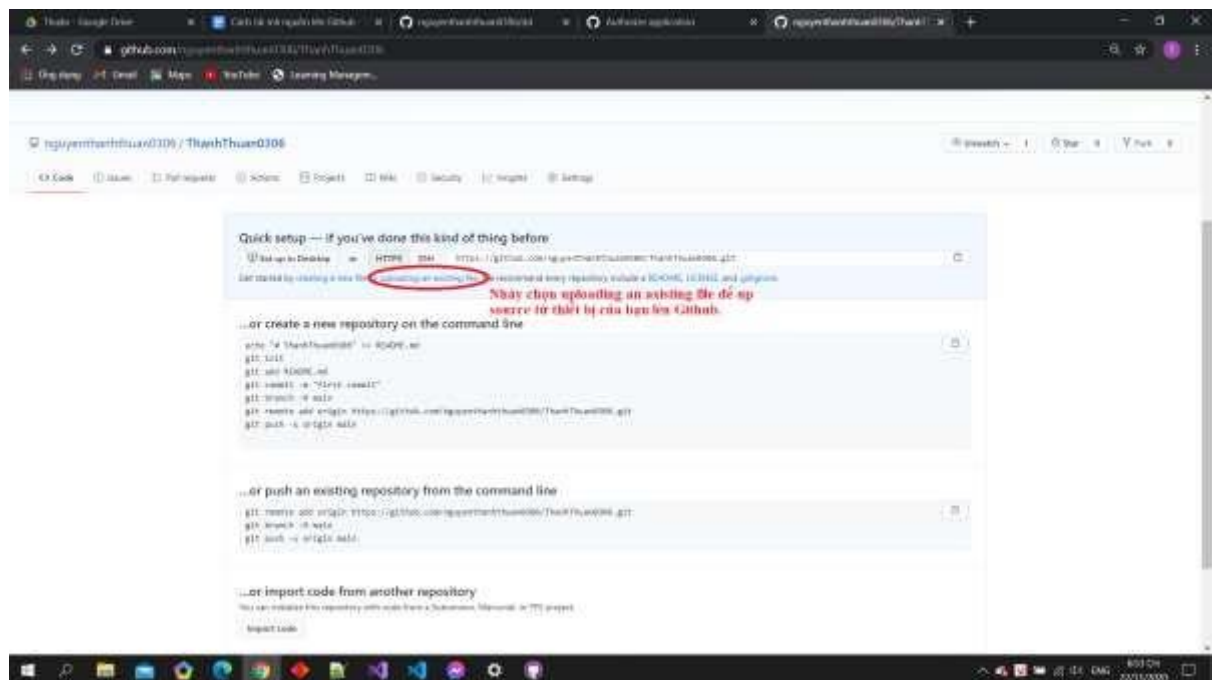
B7: Tạo ra 1 project của bạn



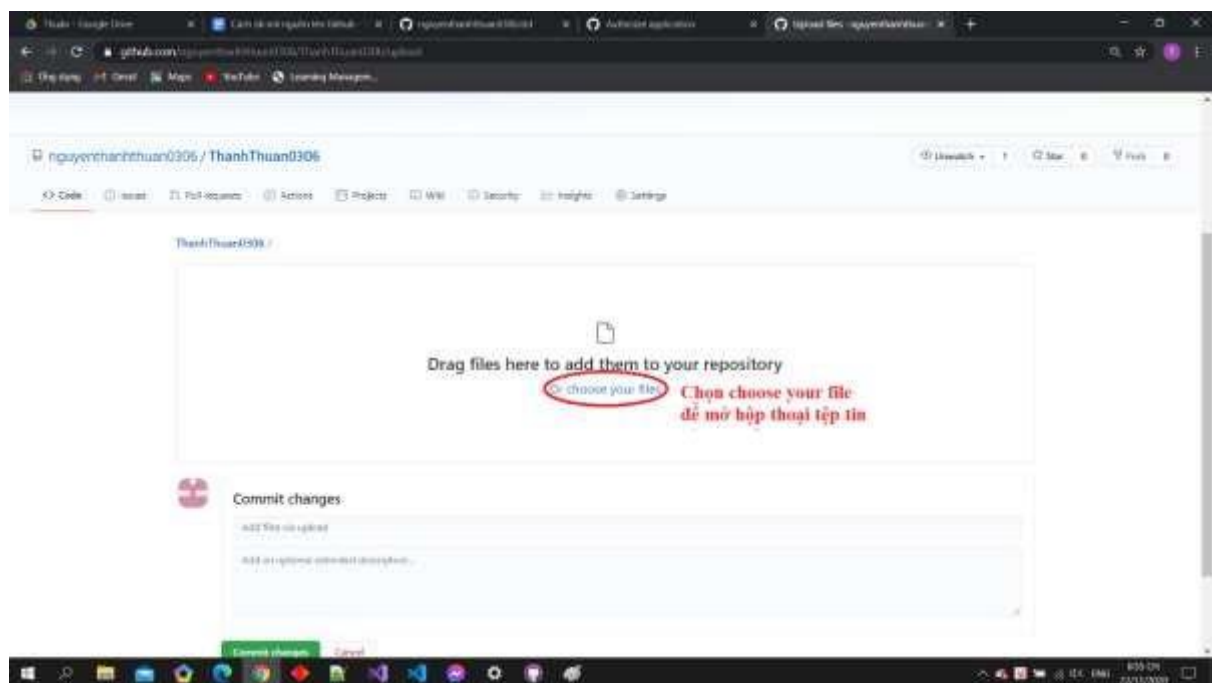
B8: Tạo kho lưu trữ mới

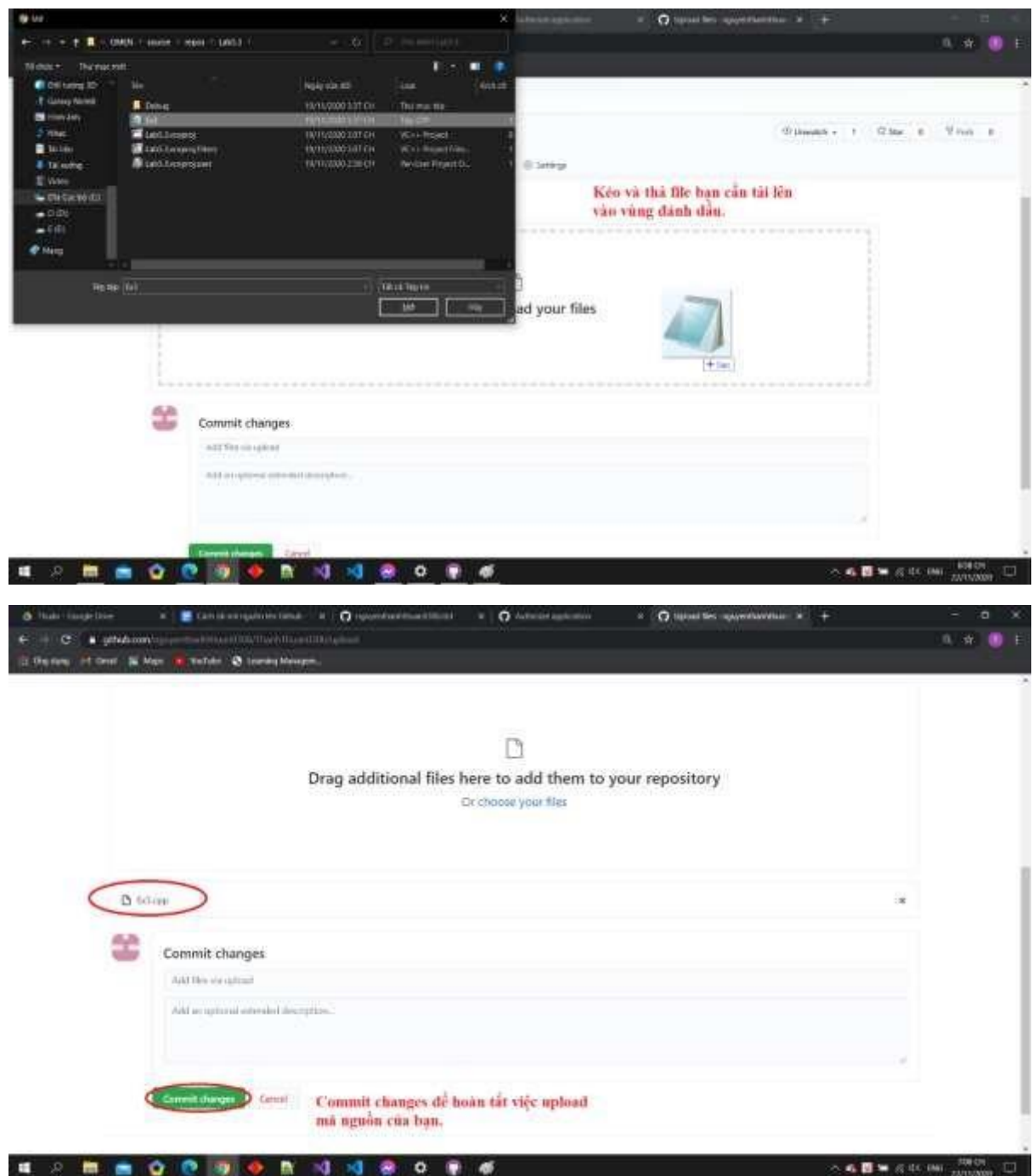


B9: Up mã nguồn từ thiết bị bạn lên GitHub

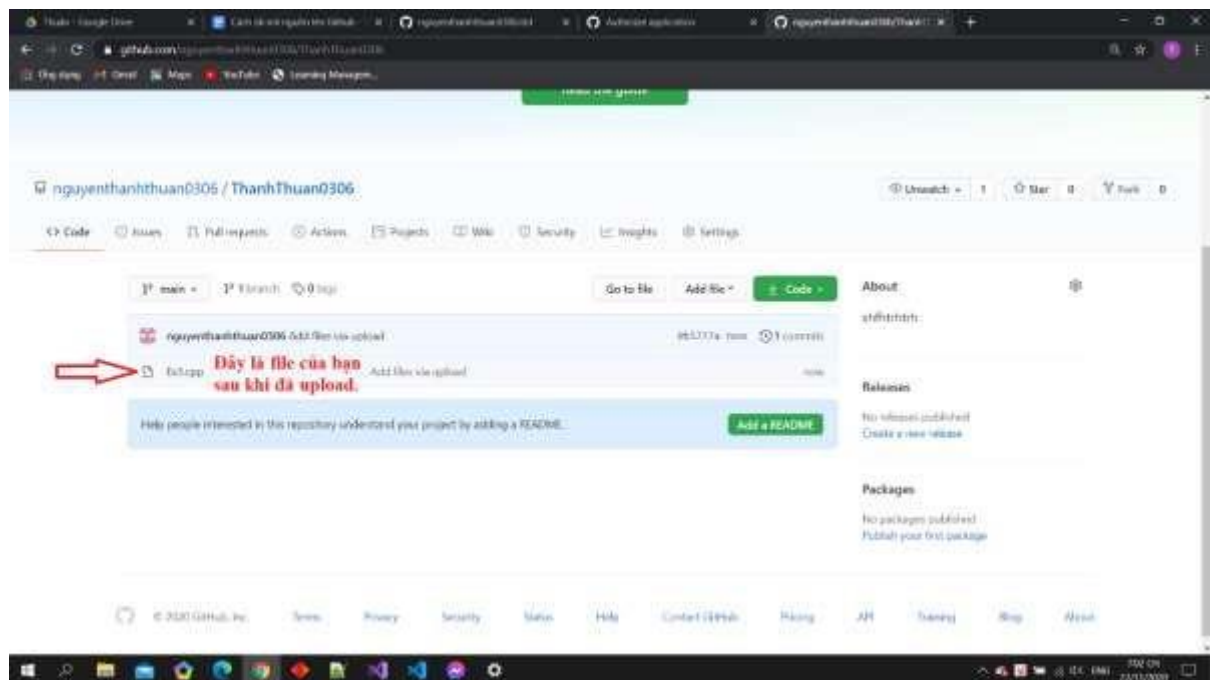


B10: Chọn file và kéo thả vào vùng đánh dấu. Sau đó xác nhận sự thay đổi





B11: Đã hoàn thành việc up mã nguồn lên GitHub

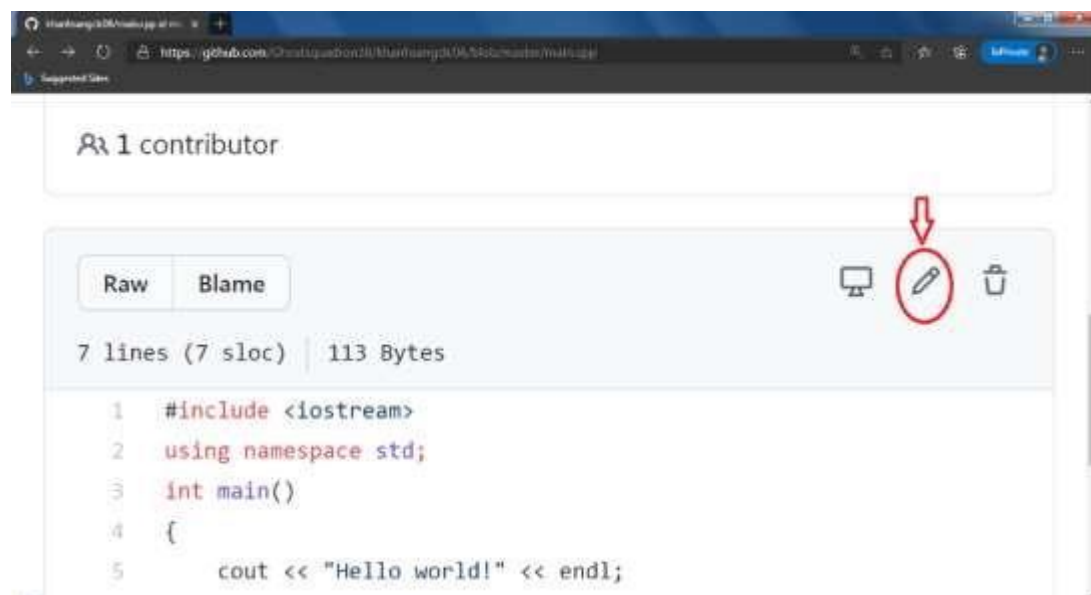


1.5.3 Các tính năng của GitHub

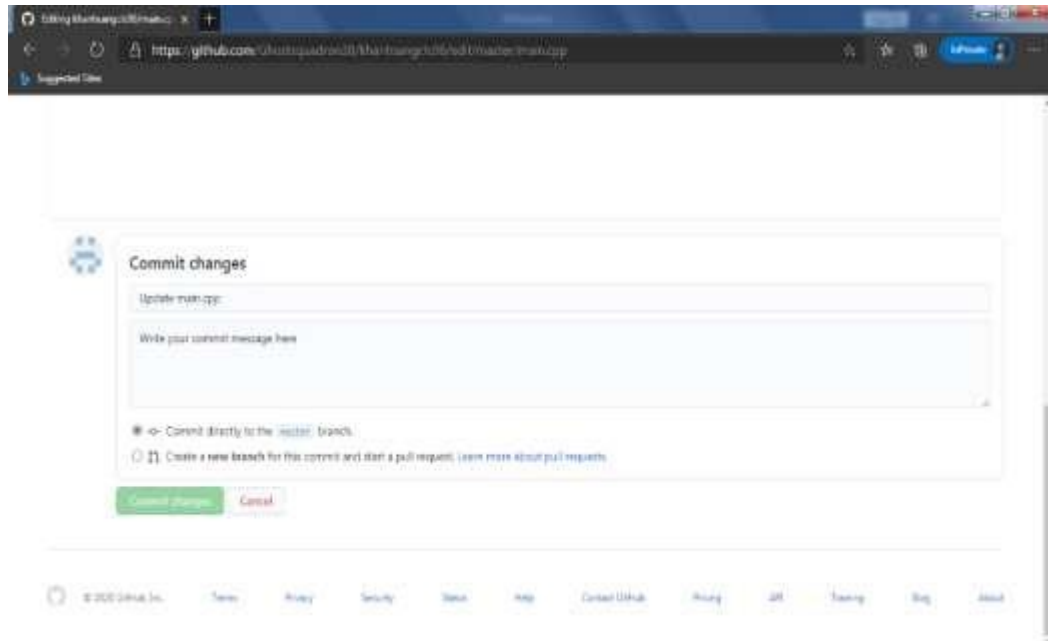
1.5.3.1 Commit message

- Giúp ghi nhận sự thay đổi theo tiến trình update của lập trình viên.

B1: Chọn biểu tượng edit ở góc phải.

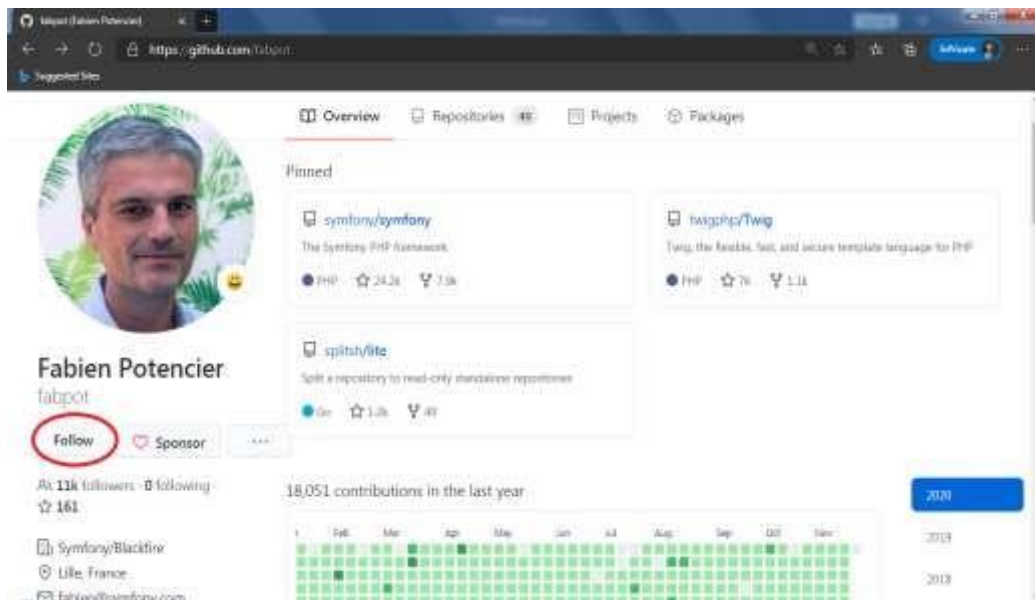


B2: Chỉnh sửa code, sau đó viết commit message để cập nhật và báo cáo về những thay đổi. Sau đó chọn Commit changes.



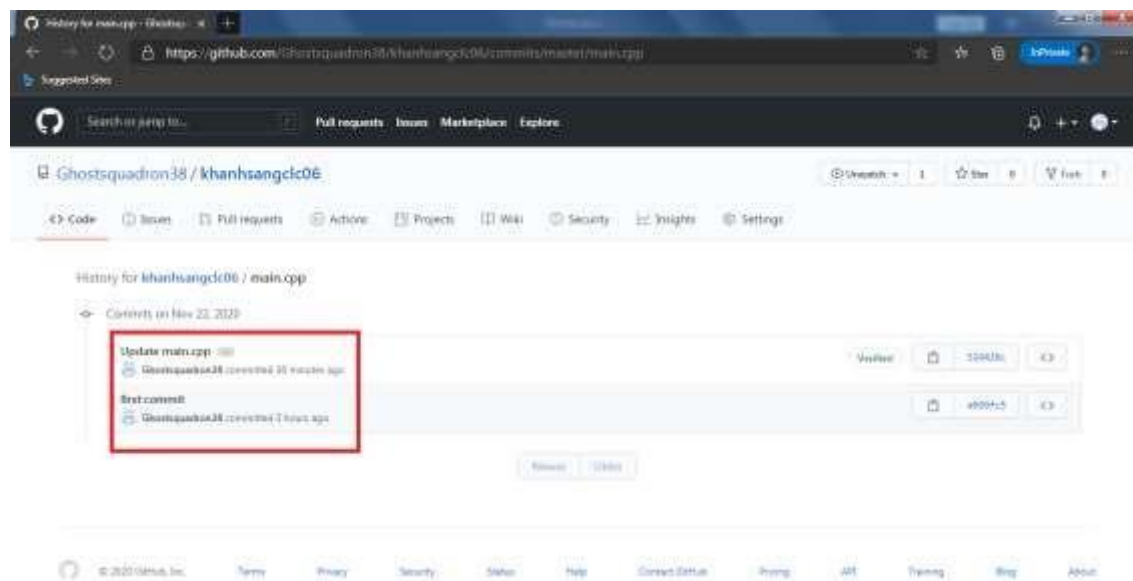
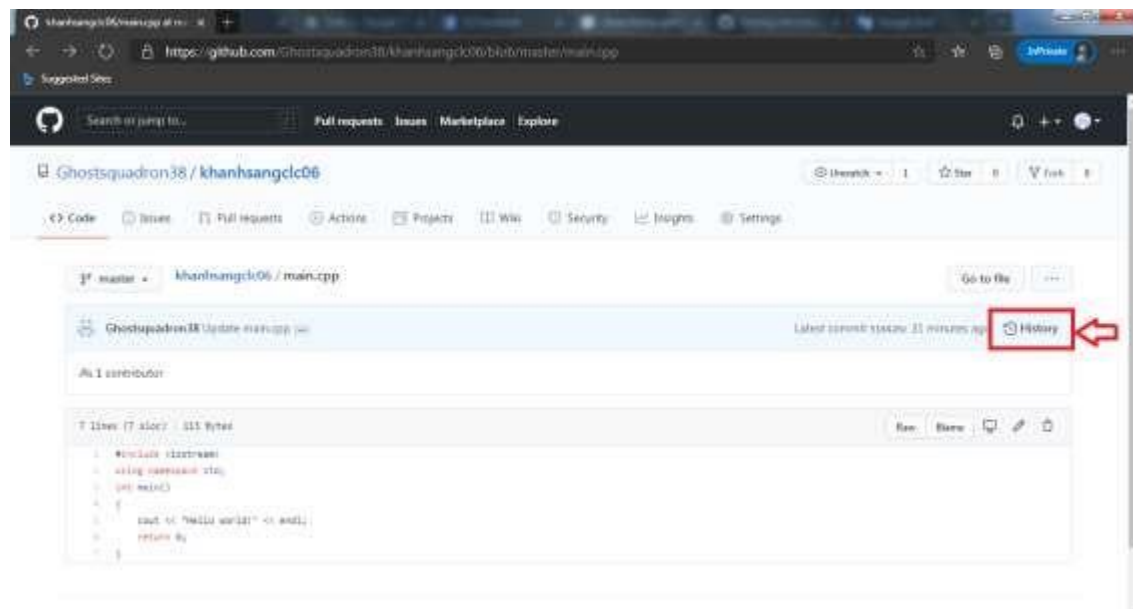
1.5.3.2 *Follow*

- Chọn ai đó mà bạn muốn theo dõi trên GitHub và nhấn nút "theo dõi".



1.5.3.3 *Show history*

- Click "history". Sau đó lịch sử chỉnh sửa sẽ được hiện lên



Nguồn:

- <https://wiki.matbao.net/github-la-gi-cach-chia-se-ma-nguon-tai-code-de-dang-tren-github/>
- github.com/features
- <http://dotnet.edu.vn/TinTucND/Huong-dan-pushday-code-len-Github1884.aspx>