

Trường ĐH CNTT TP.HCM Khoa: CNTT Bộ môn: CNPM Phát Triển Phần Mềm Và Ứng Dụng Thông Minh	<h1 style="text-align: center;">BÀI 3</h1> <h2 style="text-align: center;">GITHUB</h2>	
--	--	---

### A. MỤC TIÊU:

- Tạo và quản lý tài khoản GitHub.
- Quản lý được một project dùng GitHub.

### B. DỤNG CỤ - THIẾT BỊ THỰC HÀNH CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

### C. NỘI DUNG THỰC HÀNH

#### 1. Tại sao cần sử dụng GitHub?

- Làm thế nào để quản lý source code chung cho cả nhóm?
- Làm thế nào để quản lý được các phiên bản trong quá trình quản lý phần mềm?
- Khi bản mới có lỗi lấy lại phiên bản trước có nhanh chóng không?
- Việc sửa code mà không sao lưu khiến cho họ phải viết lại code từ đầu khi phần mềm bị lỗi.
- Việc tự code riêng và gửi cho nhau qua email rất mất thời gian của nhau. Giả như người dùng A có thể chủ động xem những thay đổi của người dùng B từ xa và tiến hành gộp trực tiếp những thay đổi của người dùng B vào sản phẩm.

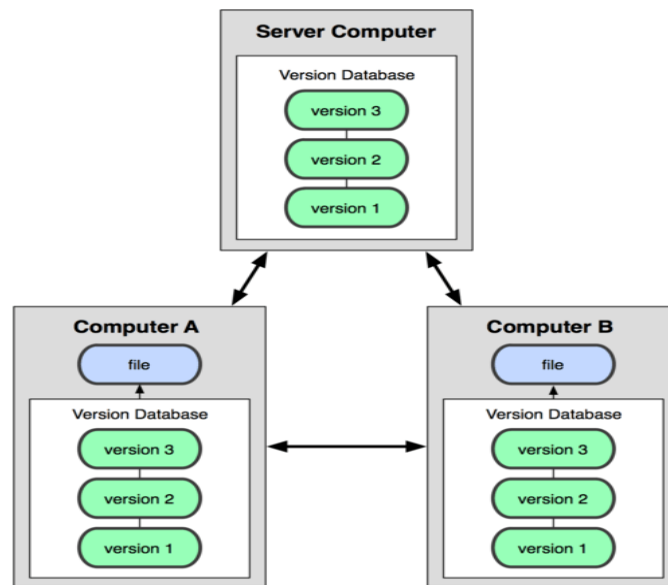
Để giải quyết vấn đề đó, chúng ta có thể sử dụng các công cụ quản lý source code như: **GitHub**, Subversion (SVN) của **Google**, Team Foundation Server (TFS) của **Microsoft**, Visual Source Safe.

#### 2. Như vậy GitHub là gì?

**GitHub** là một dịch vụ lưu trữ trên Web dành cho các dự án có sử dụng hệ thống kiểm soát **Git revision**. GitHub cung cấp chức năng social networking như feeds, followers và network graph để các Developer học hỏi kinh nghiệm làm việc

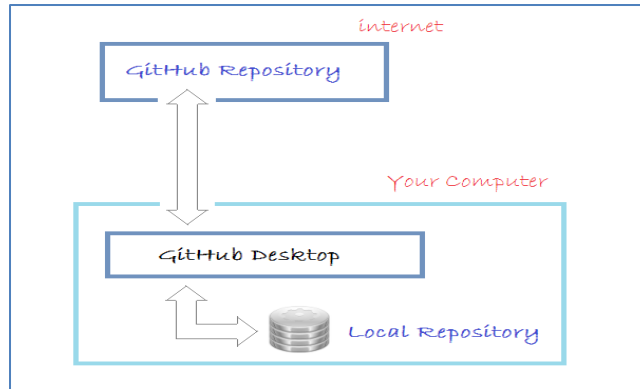
thông qua lịch sử commit. Nếu comment dùng để mô tả chức năng của đoạn code, thì commit message trên Git dùng để mô tả hành động mà Dev vừa thực hiện trên code.

Git là tên gọi của một Hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay. DVCS nghĩa là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (clone) từ một kho chứa mã nguồn (repository), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (commit) rồi đưa lên máy chủ nơi đặt kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia. Trong Git, thư mục làm việc trên máy tính gọi là Working Tree.



### 3. Hướng dẫn cấu hình và sử dụng GitHub

#### 1 Nguyên tắc hoạt động



**Để làm việc với GitHub chúng ta cần:** Đăng ký một tài khoản GitHub và tạo một Repository (GitHub Repository). Cài đặt GitHub Desktop, một công cụ trực quan quản lý Local Repository. Cấu hình để có thể đồng bộ hóa dữ liệu bằng GitHub Desktop lên Repository server.

## 2 Đăng ký tài khoản GitHub

Đăng ký miễn phí một tài khoản GitHub tại: <https://github.com>

Username  
Pick a username

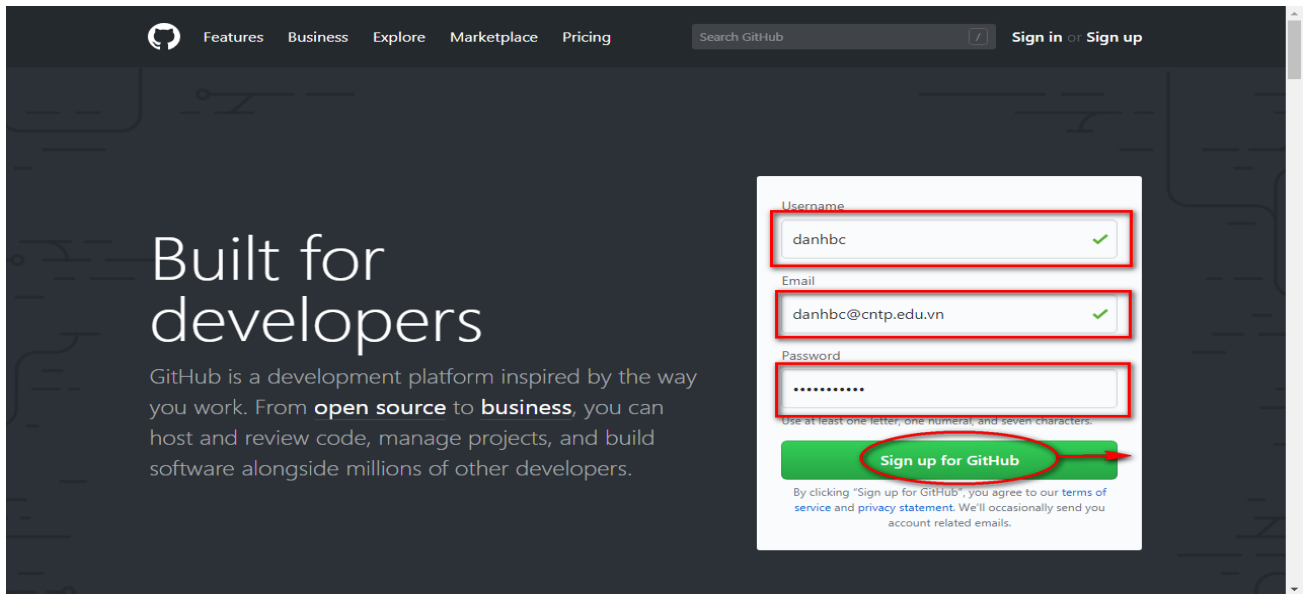
Email  
you@example.com

Password  
Create a password  
Use at least one letter, one numeral, and seven characters.

**Sign up for GitHub**

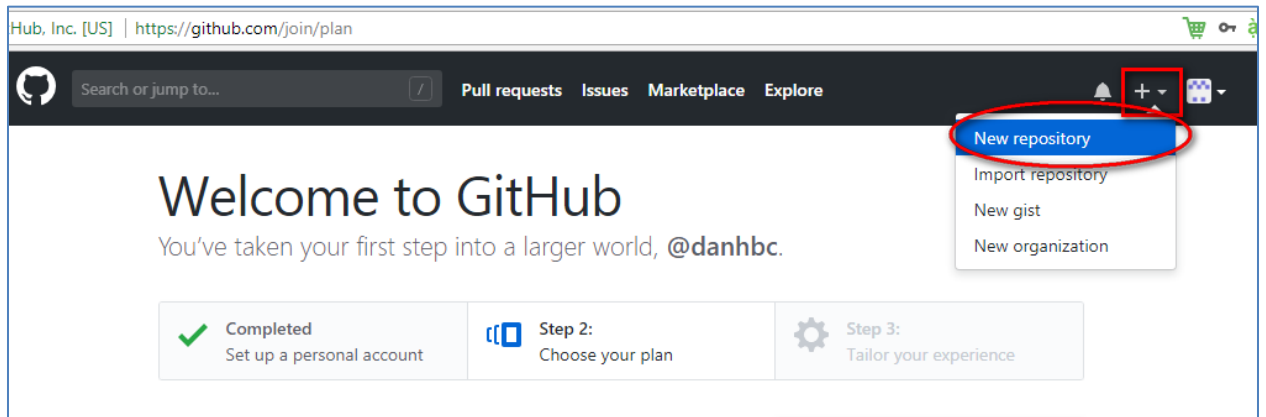
By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

Nhập đầy đủ thông tin theo yêu cầu:



### 3 Tạo GitHub Repository

B1:



B2:

## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

Repository name

 ✓

Great repository names are short and memorable. Need inspiration? How about **furry-spoon**.

Description (optional)



**Public**

Anyone can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None ▾

Add a license: None ▾



**Create repository**

**B3: Repository đã được tạo ra**

danhbc / demo\_project Watch 0 Star 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#) [Settings](#)

**Quick setup — if you've done this kind of thing before**

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) [https://github.com/danhbc/demo\\_project.git](https://github.com/danhbc/demo_project.git)

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# demo_project" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/danhbc/demo_project.git
git push -u origin master
```

**...or push an existing repository from the command line**

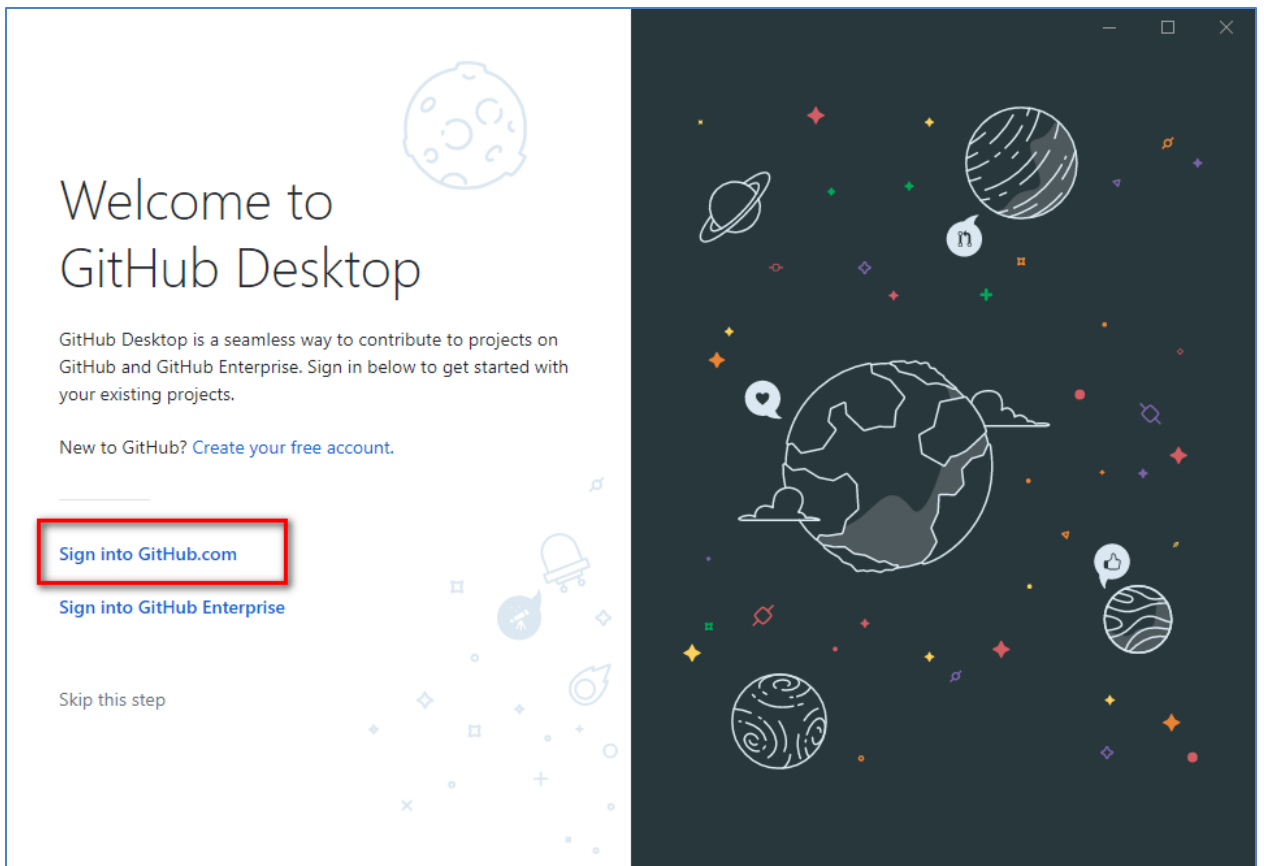
```
git remote add origin https://github.com/danhbc/demo_project.git
git push -u origin master
```

#### 4. Download & cài đặt GitHub Desktop

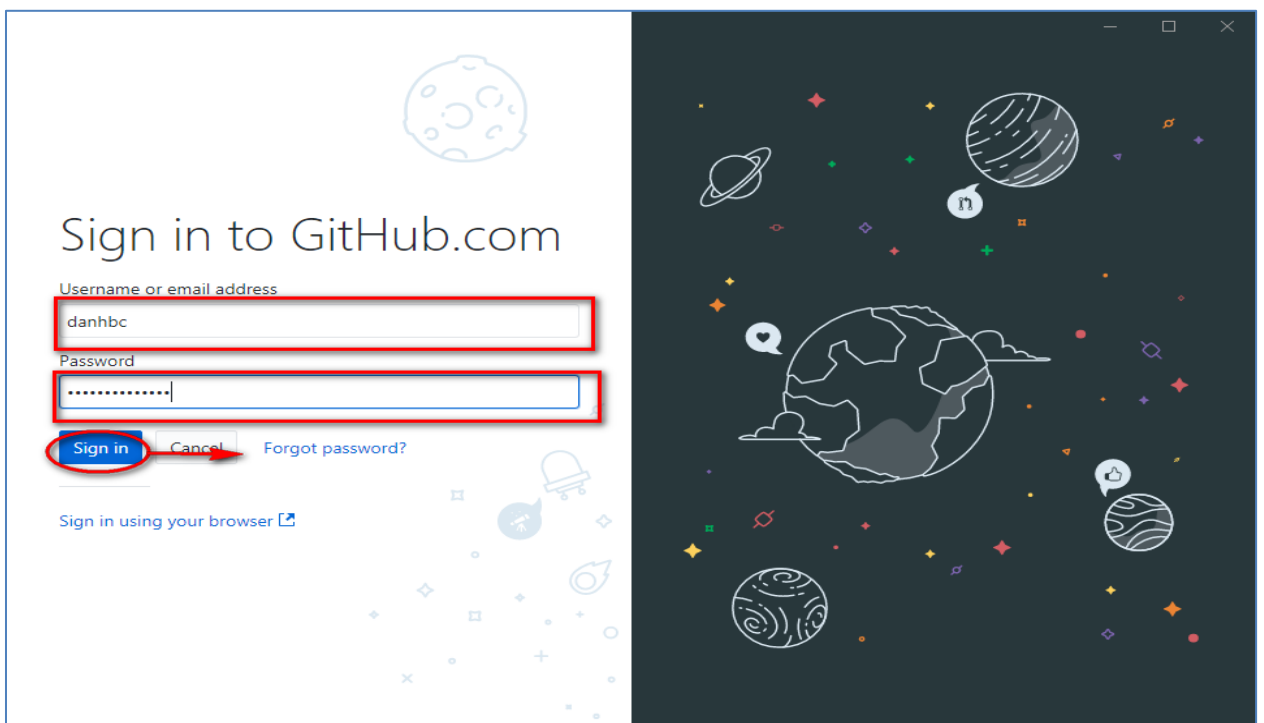
**B1:** Truy cập địa chỉ: <https://desktop.github.com/>

<https://desktop.github.com>

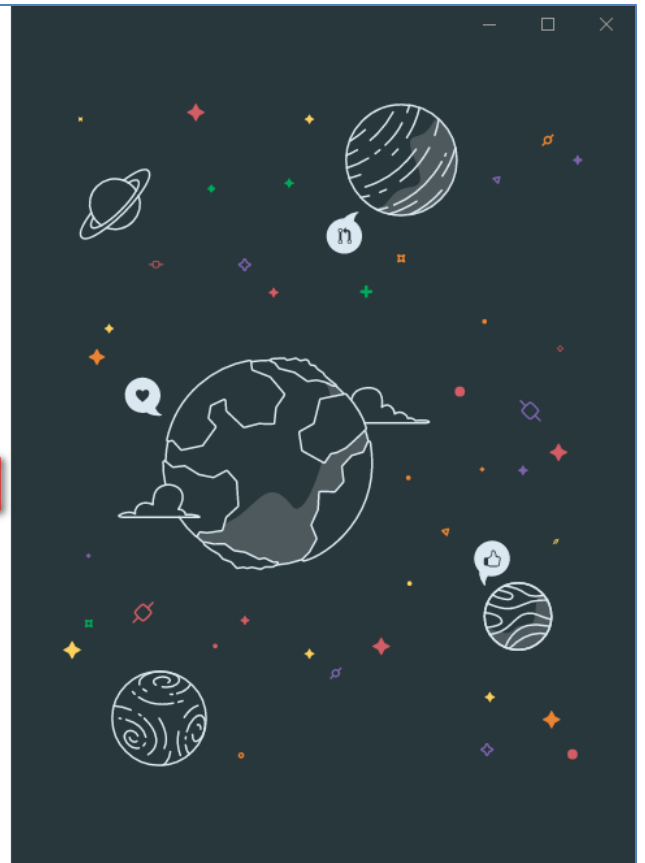
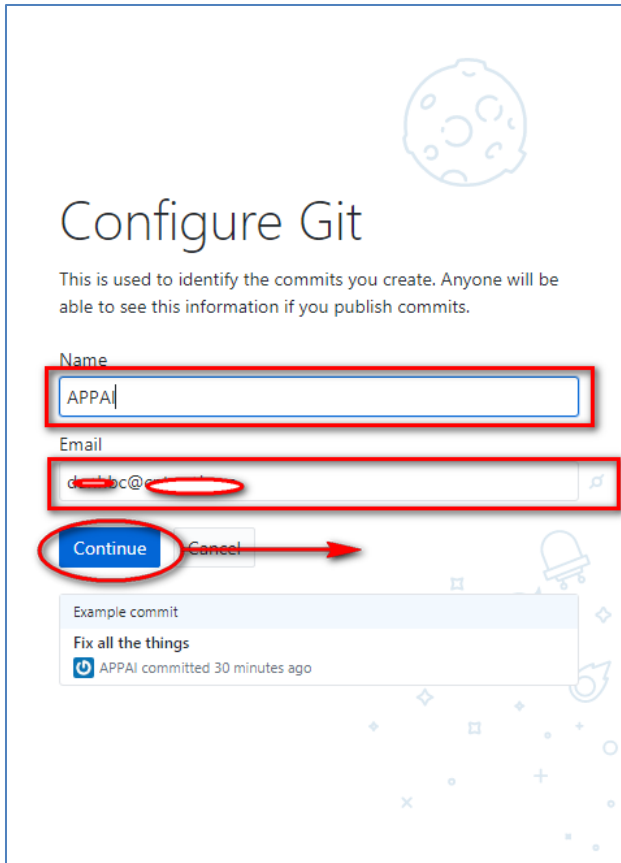
**B2:** Cài đặt GitHub Desktop vào máy tính và mở ứng dụng



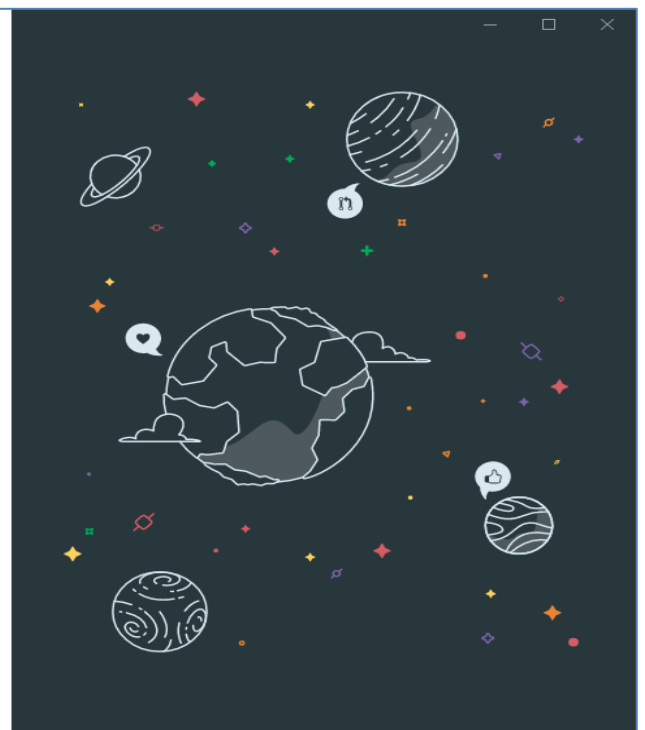
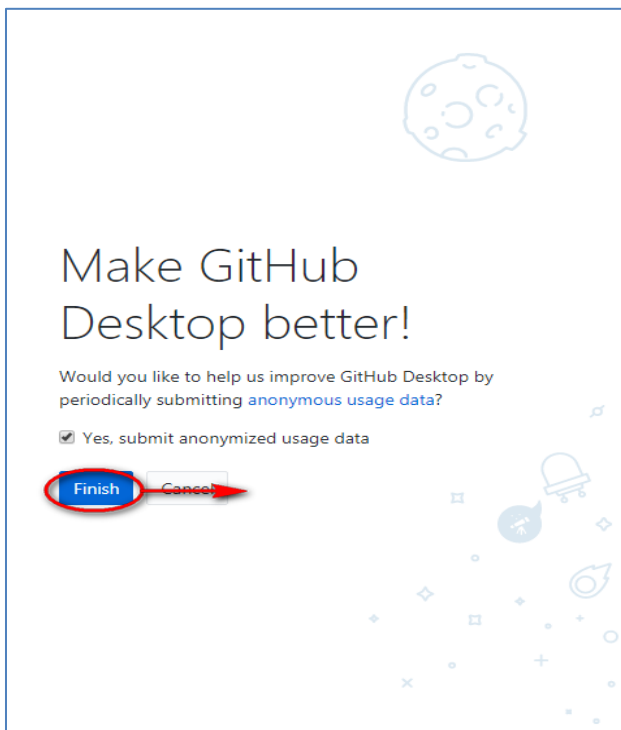
**B3:** Đăng nhập trên **GitHub Desktop** để kết nối vào tài khoản GitHub



**B4:**

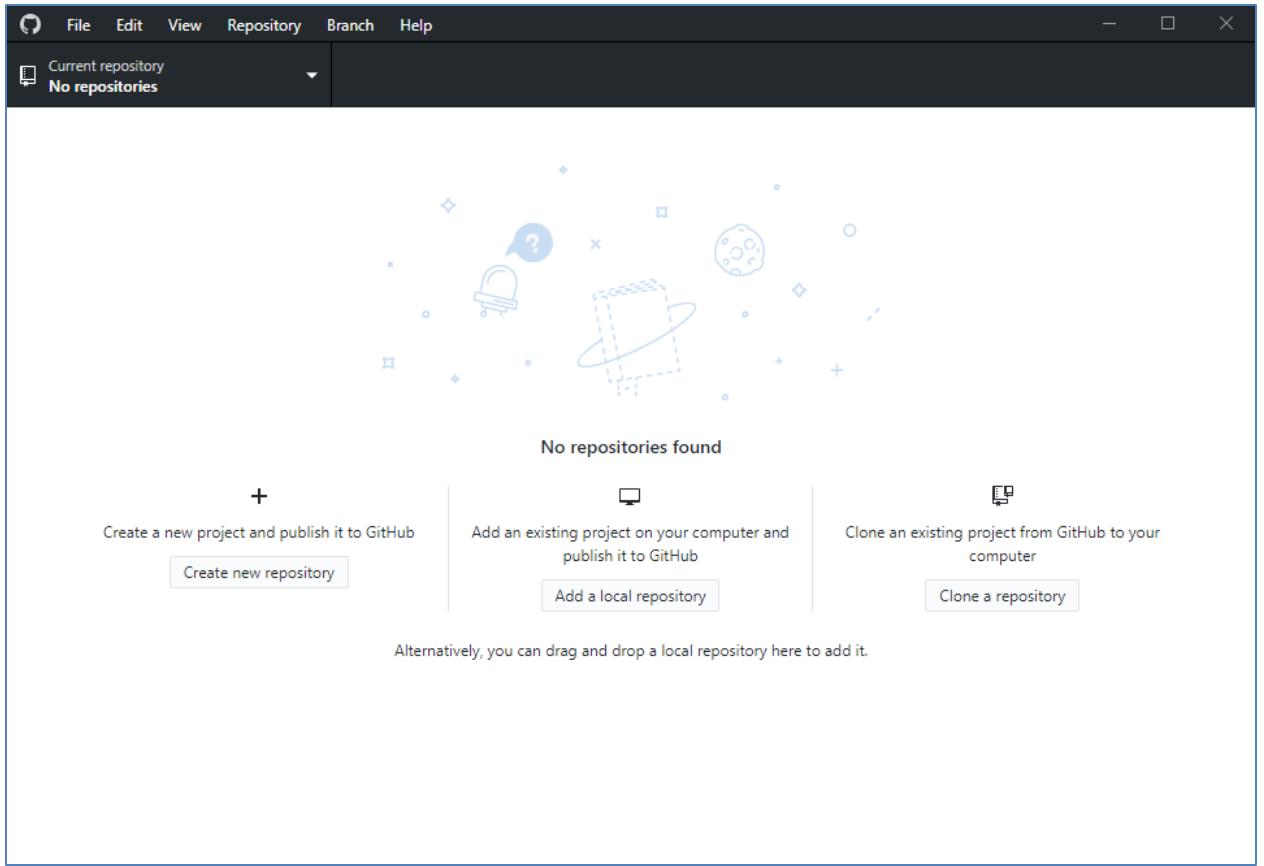


**B5:**



**B6:**

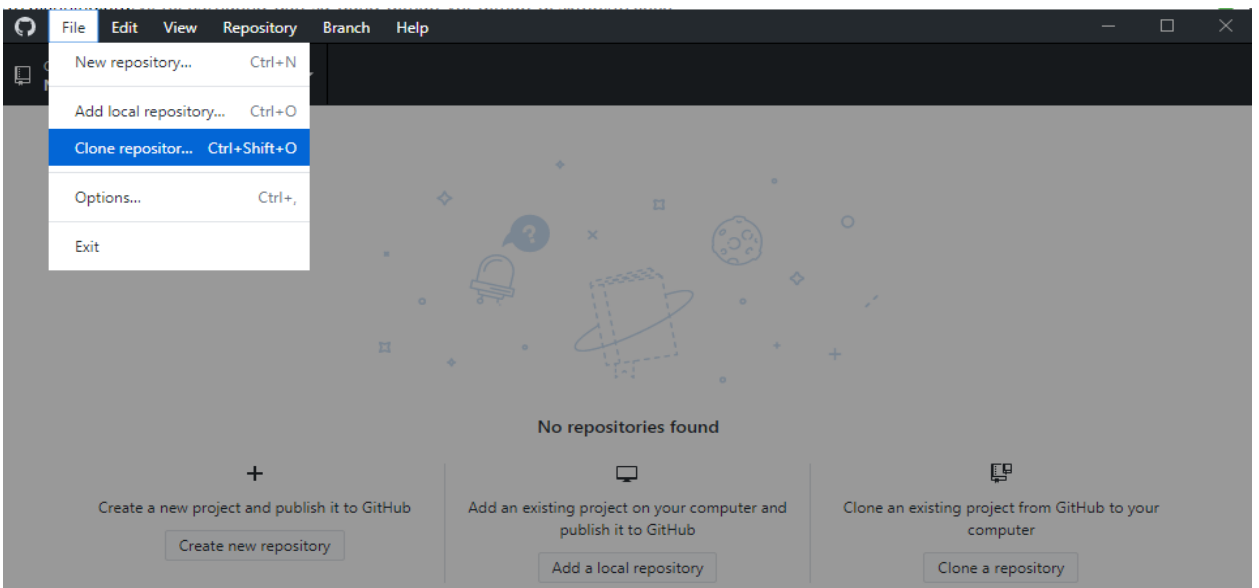




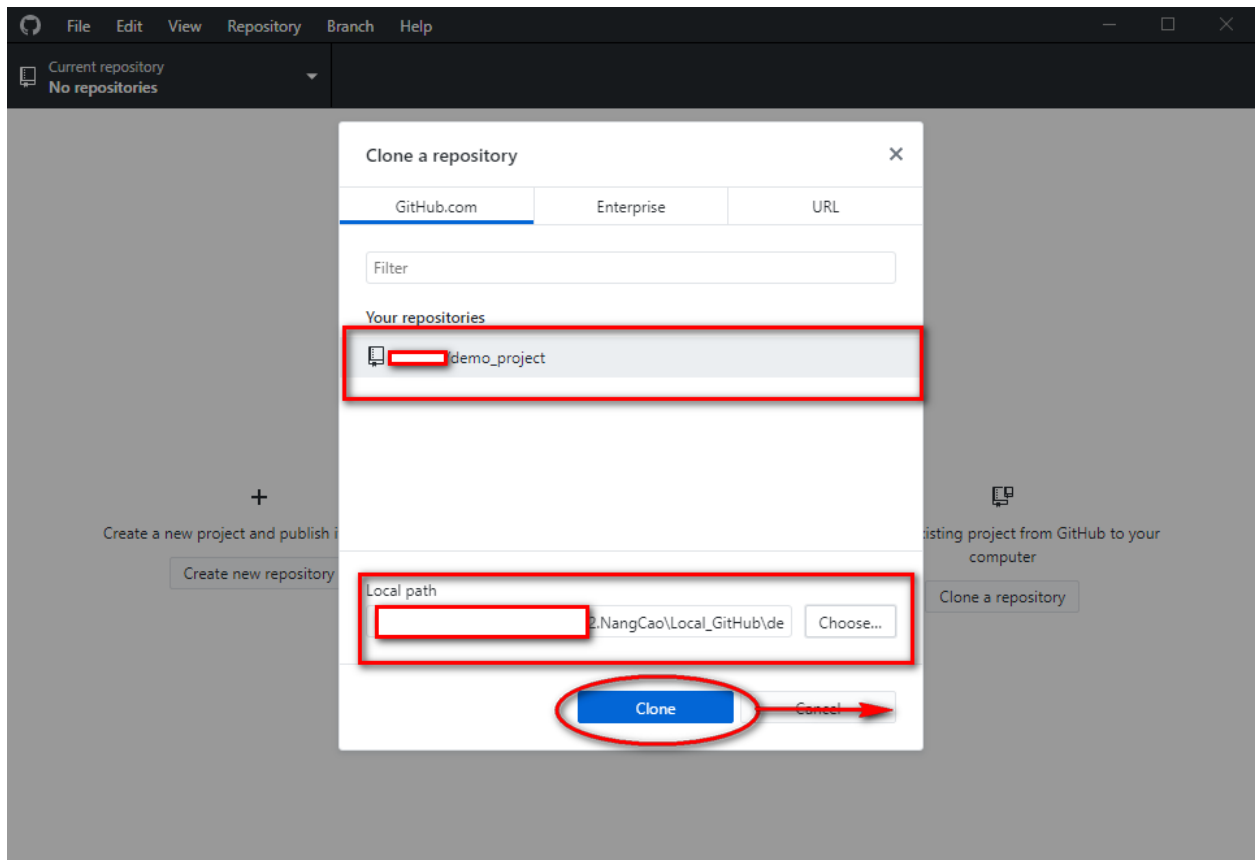
## 5. Kết nối GitHub và GitHub Desktop

Cần chọn một thư mục trống để làm vị trí chứa dữ liệu local

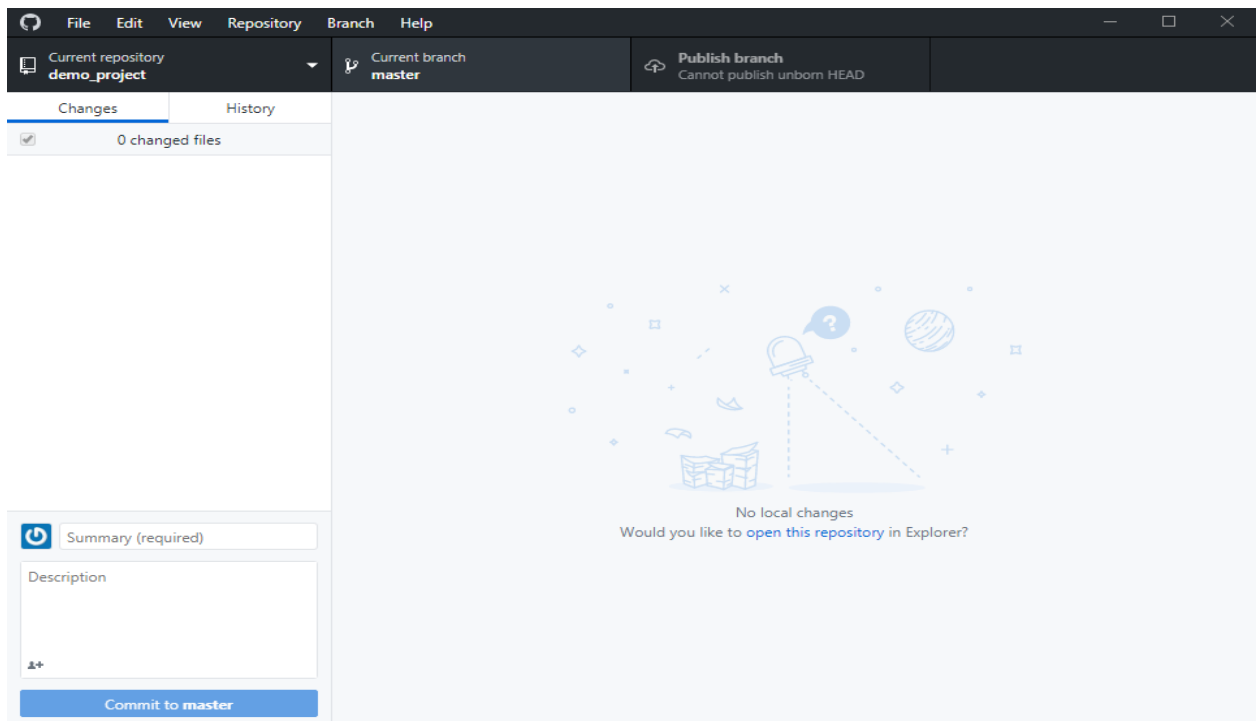
**B1:** Trên GitHub Desktop, lựa chọn một Repository bạn đã tạo trên GitHub để clone (Tạo bản sao chép) thành một bản ở máy tính local.



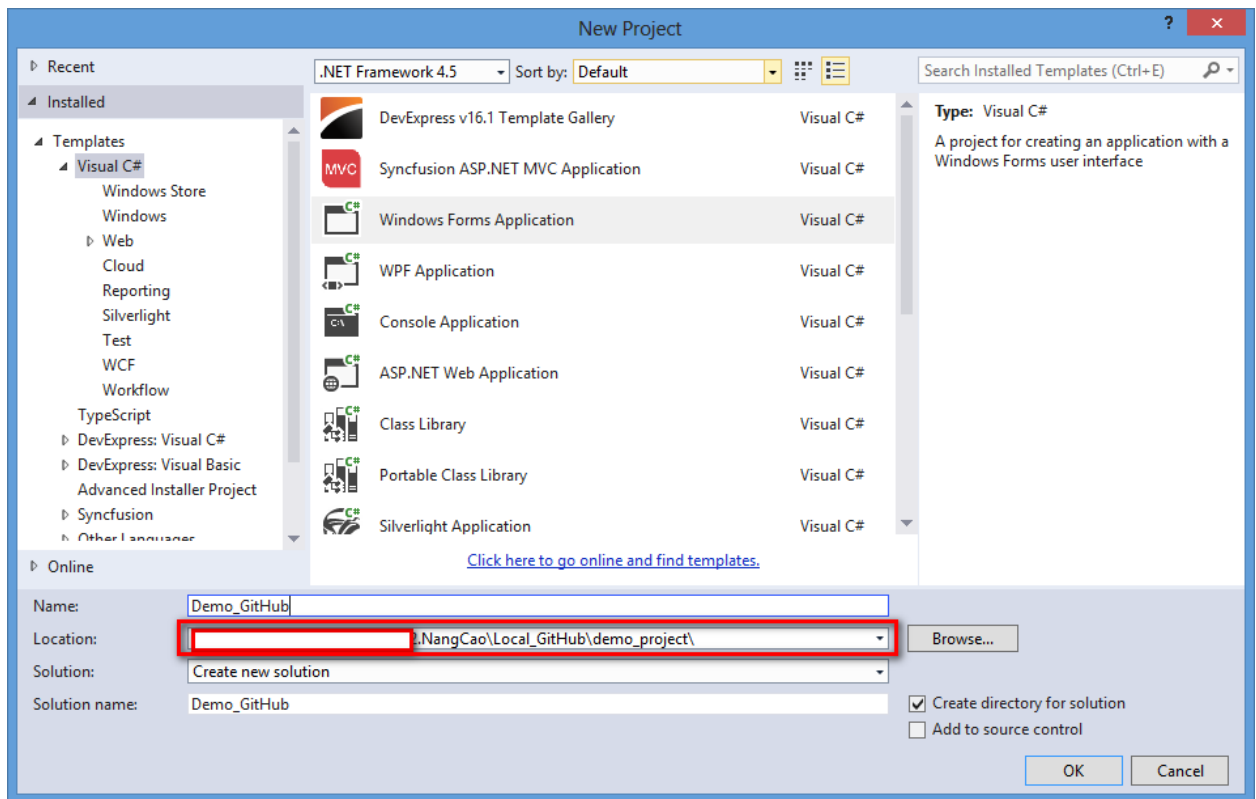
**B2:** Tiến hành chọn project đồng bộ bản online và local



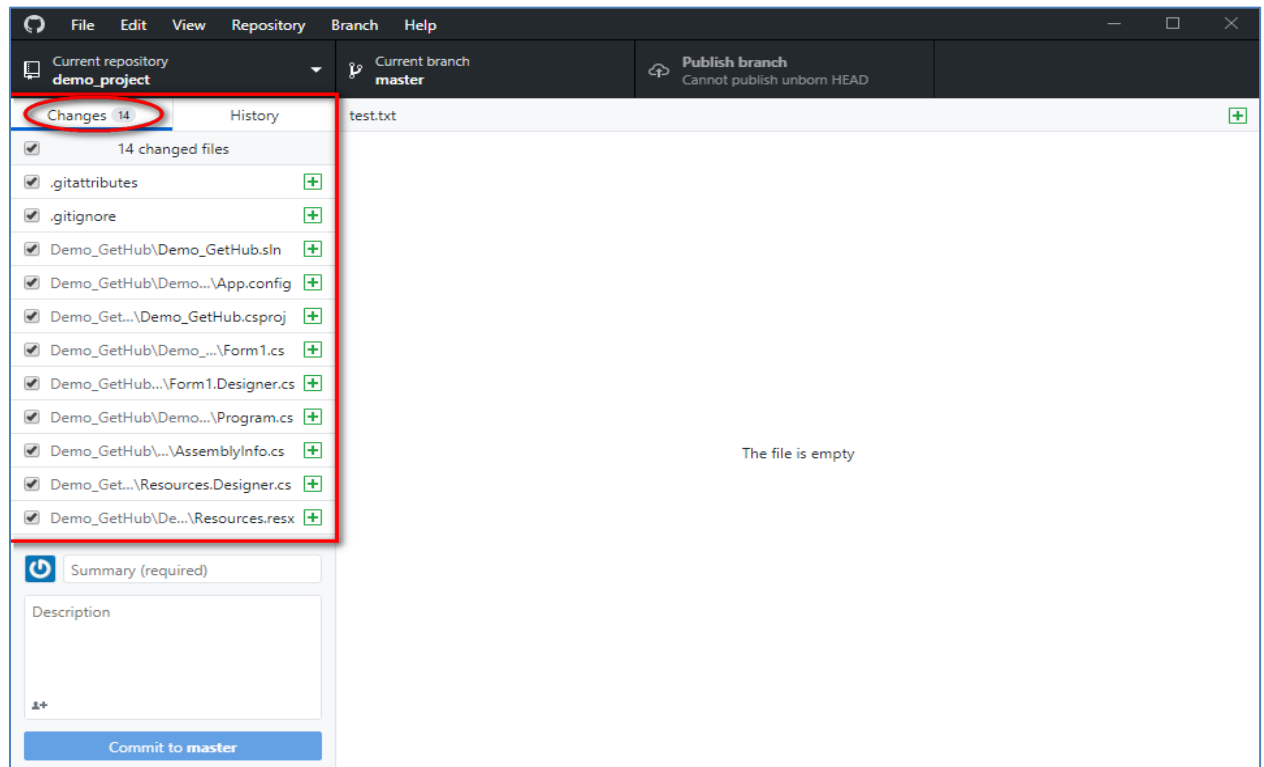
### B3: Hiện tại project đang rỗng



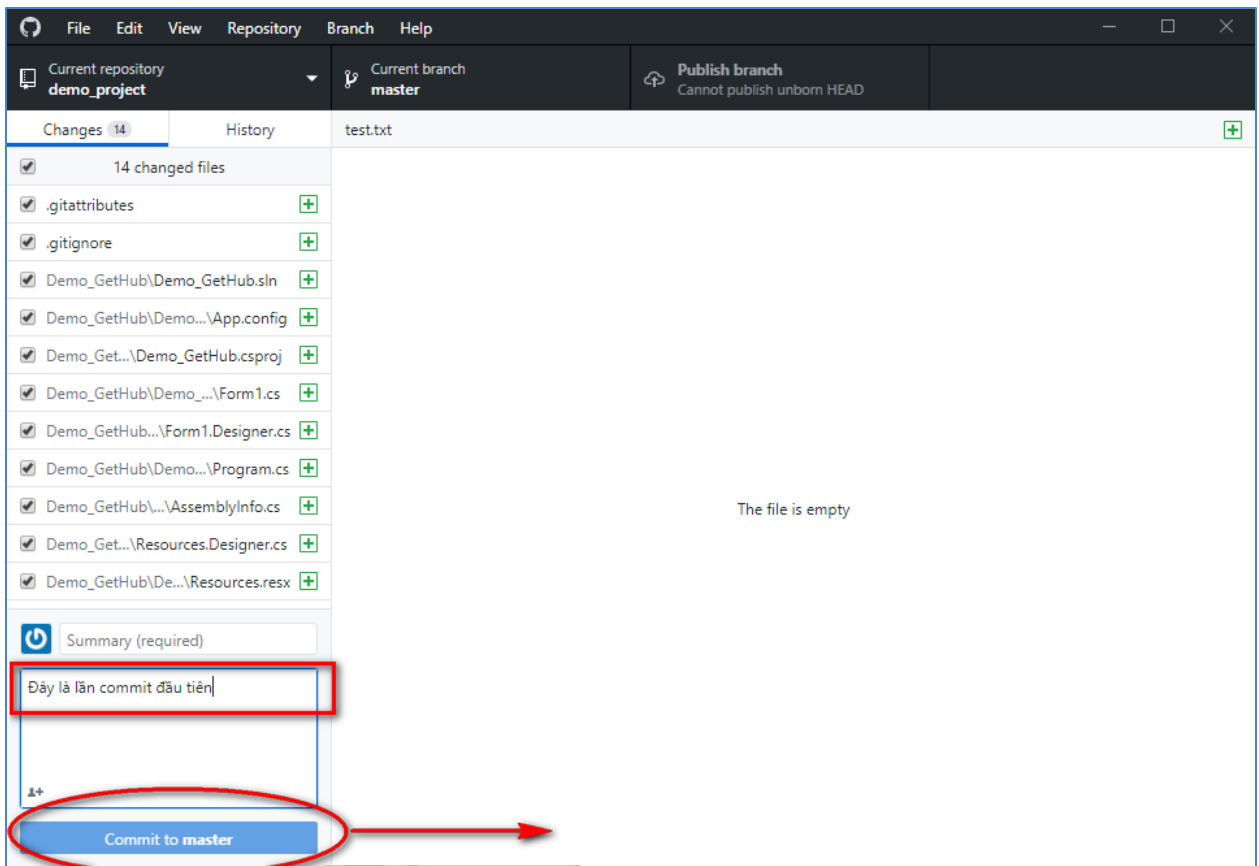
### B4: Tạo project vào thư mục Local Repository



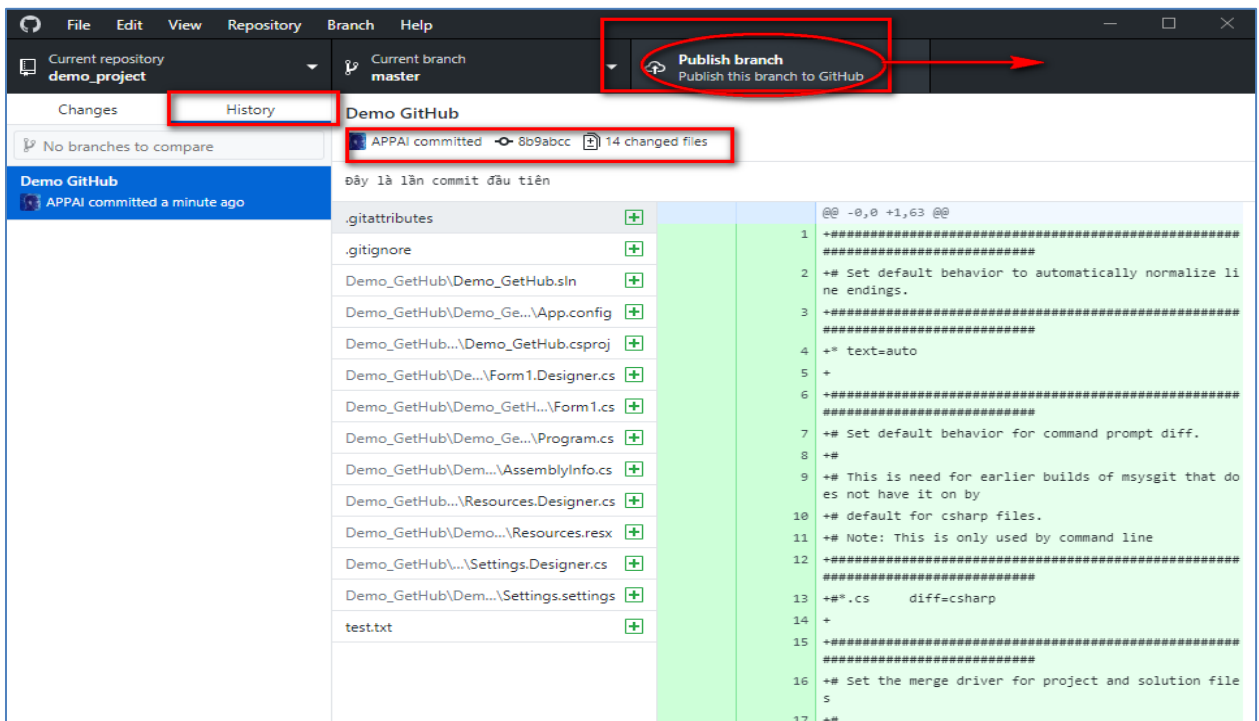
## B5: Trên Github local bạn sẽ thấy có sự thay đổi



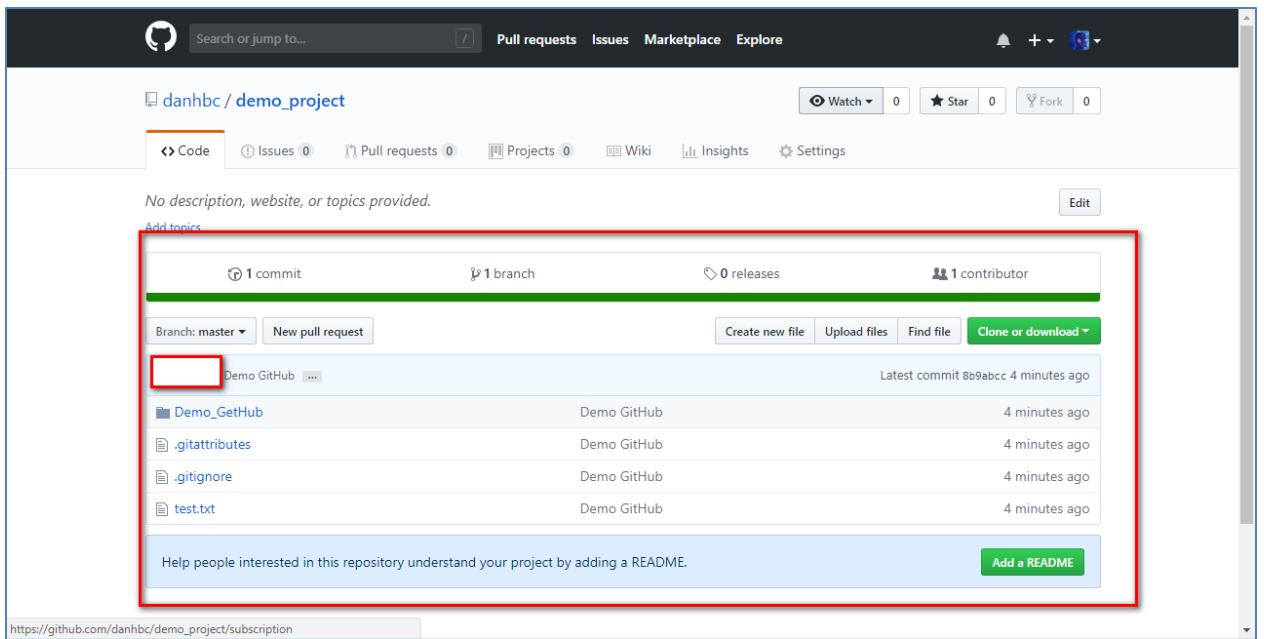
## B6: Nhập thông tin ghi chú (Comment) và nhấn Commit dữ liệu



## B7: Sau khi commit chọn đồng bộ lên server Github

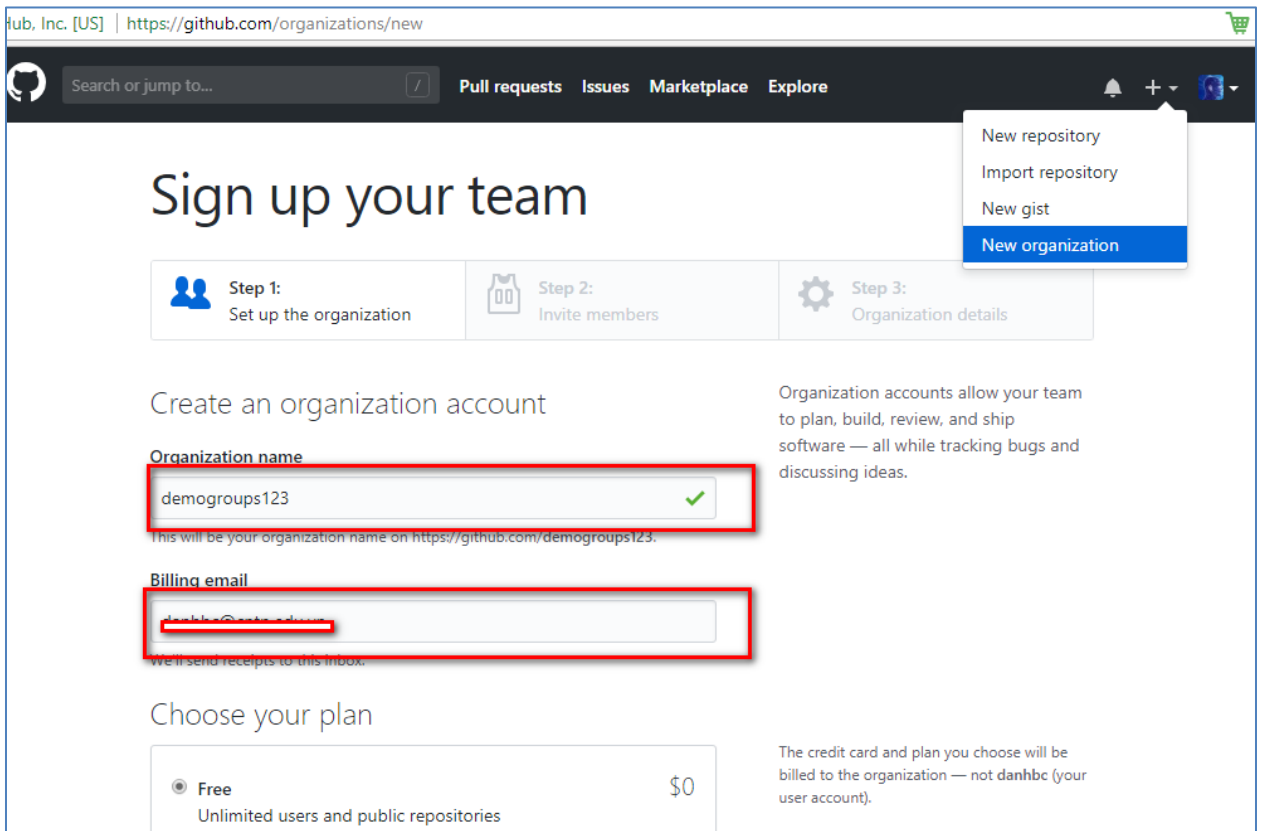


## B8: Xem lại project trên server



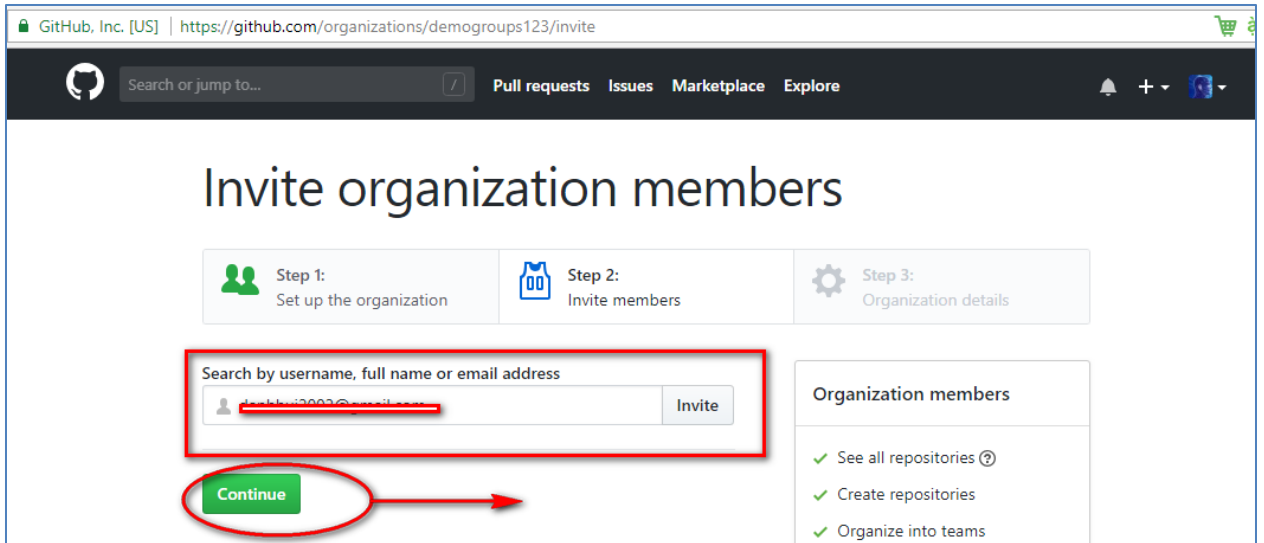
## 6. Tổ chức nhóm trong GitHub

### B1: Tạo nhóm

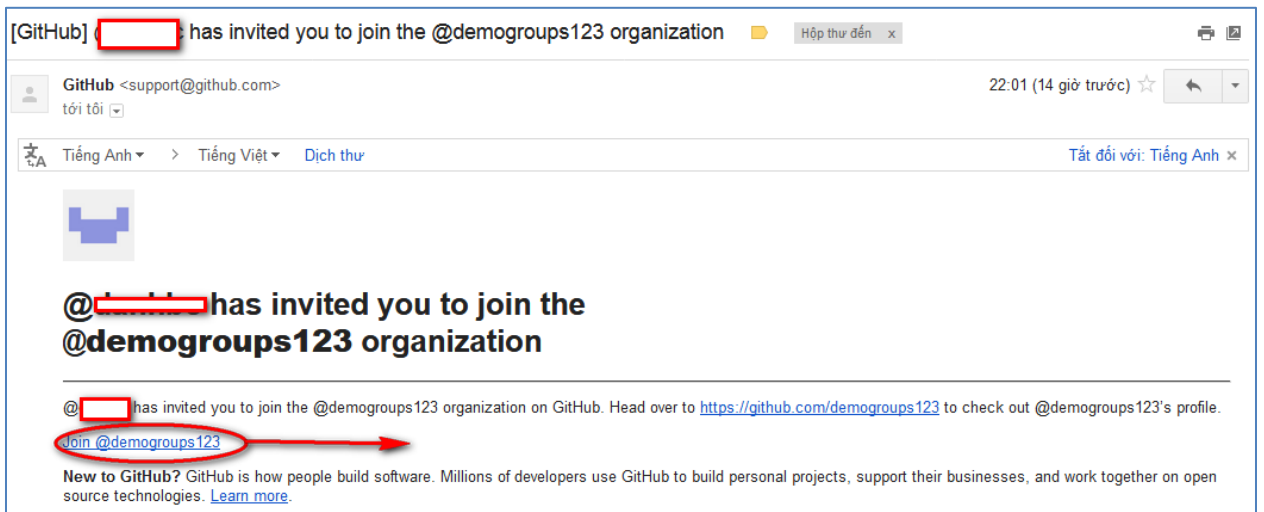


=> **Create organization**

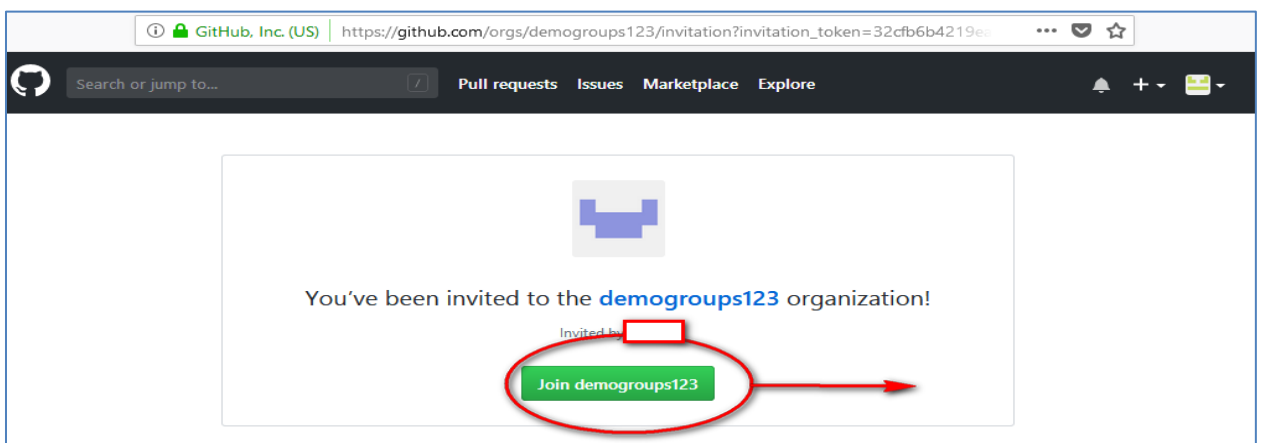
### B2: Mời thành viên tham gia project



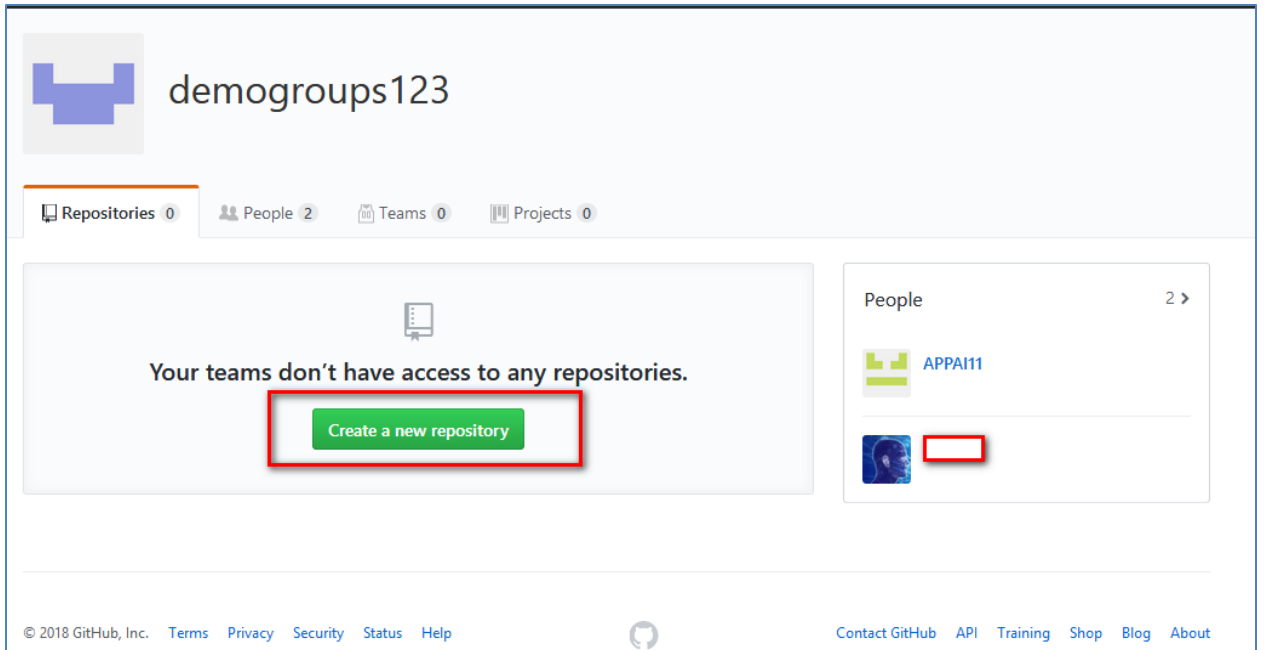
### B3: Xác nhận



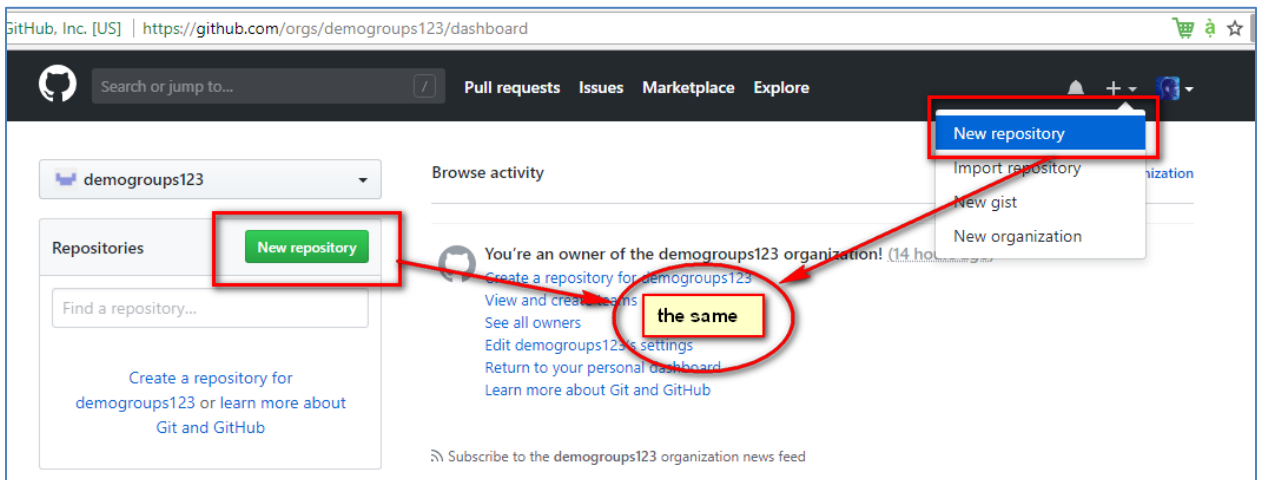
### B4: Gia nhập nhóm



### B5: Gia nhập nhóm thành công



## B6: Tạo project cho nhóm hoàn toàn giống như tạo cho cá nhân





## B7:

## Create a new repository

A repository contains all the files for your project, including the revision history.

---

**Owner** **Repository name**

 demogroups123 / Groupstest 

Great repository names are short and memorable. Need inspiration? How about **legendary-happiness**.

**Description (optional)**




---

☒ **Public**  
Anyone can see this repository. You choose who can commit.

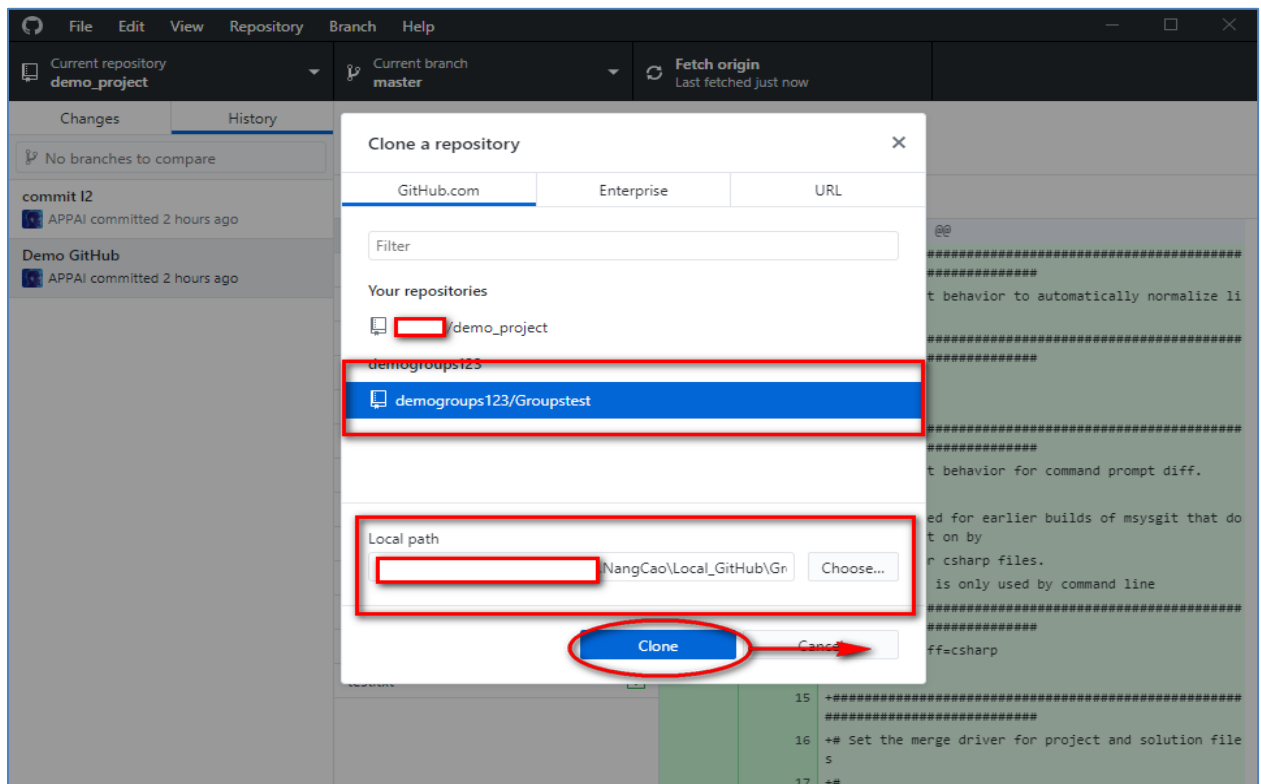
☐ **Private**  
You choose who can see and commit to this repository.

---

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

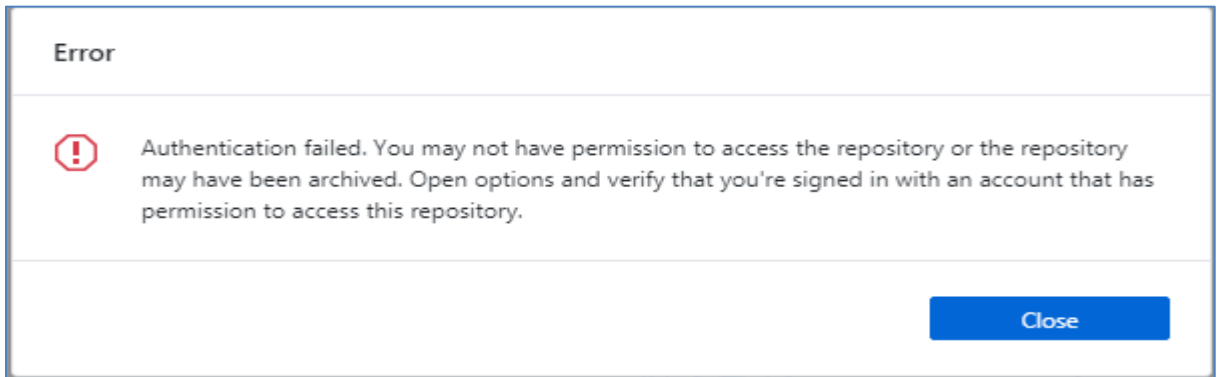
Add .gitignore: **None**  | Add a license: **None**  

**B8:**

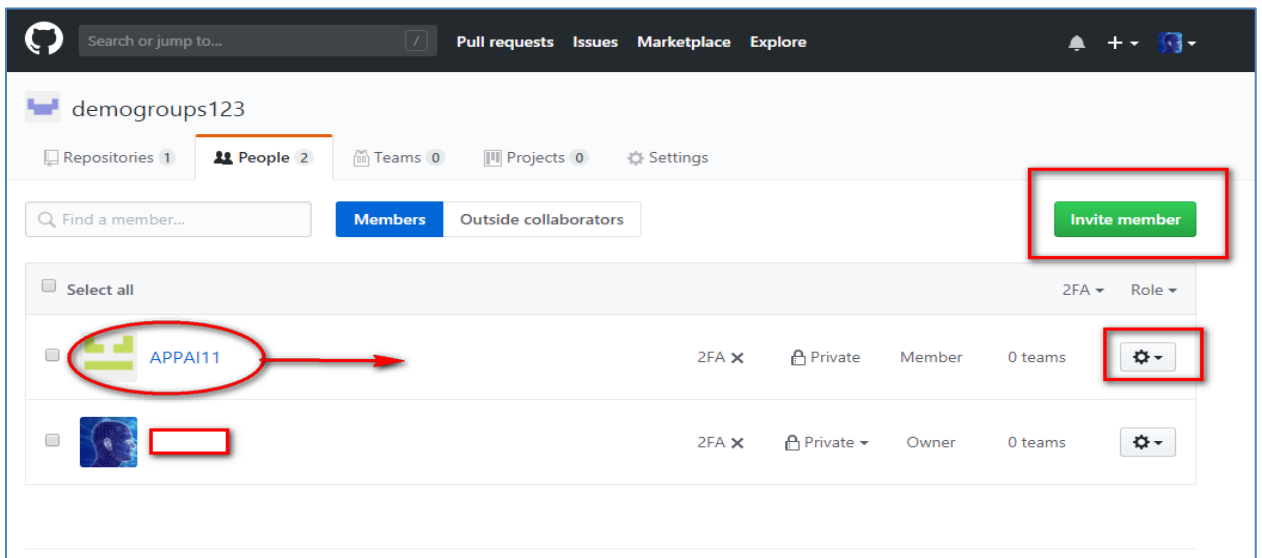




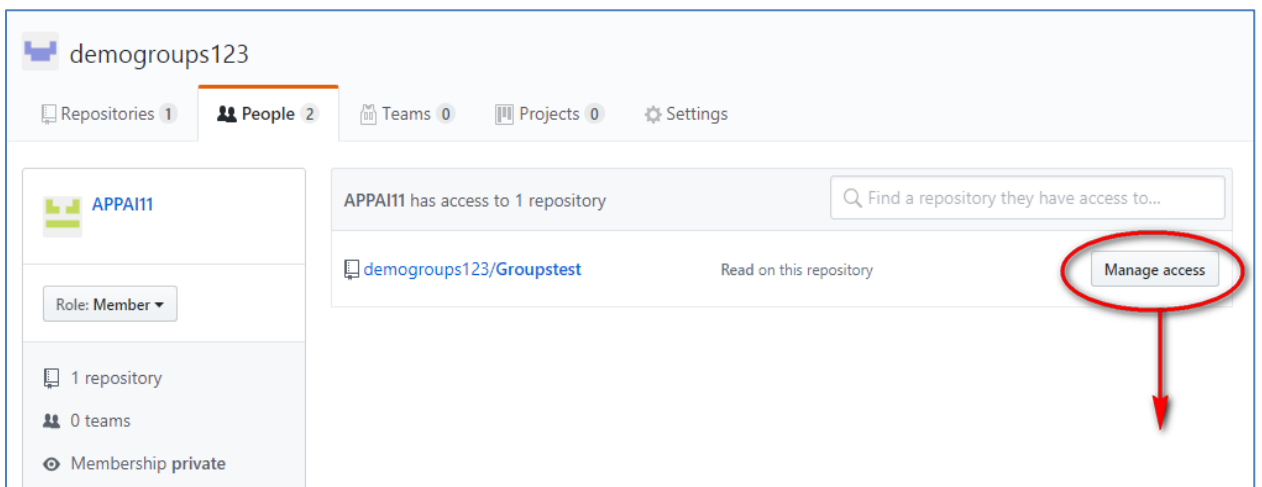
**B9:** Đăng nhập thành viên commit file mới. Có thể bị lỗi sau: Vì thành viên chưa có quyền write vào thư mục online của project



**B10:** Phân quyền cho thành viên/ quản lý thành viên



**B11:**



**B12:**

The screenshot shows the GitHub repository settings for 'demogroups123/Groupstest'. The 'People' tab is selected, showing that 'APPAI11' has 'read' access. The 'Read' permission is currently 'Active'. A red circle highlights the 'Membership privilege' option under the 'Read' section, with a red arrow pointing to the 'Write' section. The 'Write' section shows that nothing explicitly gives 'APPAI11' write access. The 'Admin' section shows that nothing explicitly gives 'APPAI11' admin access. A 'Remove access to this repository' button is visible in the top right.

**B13:**

The screenshot shows the 'Repository permissions' settings for 'demogroups123/Groupstest'. The 'Organization members' section is selected, showing that they have 'Read' access. A dropdown menu is open, showing the 'Organization member permissions' for 'Read' and 'Write'. The 'Read' permission is currently selected, and the 'Write' permission is highlighted in blue. The 'Write' permission description states: 'Members will be able to clone, pull, and push all repositories.'

**B14:**

The screenshot shows the GitHub repository settings for 'demogroups123/Groupstest'. The 'People' tab is selected, showing that 'APPAI11' has 'write' access. The 'Write' permission is currently 'Active'. A red box highlights the 'Write' section, which shows that nothing explicitly gives 'APPAI11' write access. The 'Read' section shows that nothing explicitly gives 'APPAI11' read access. The 'Admin' section shows that nothing explicitly gives 'APPAI11' admin access. A 'Remove access to this repository' button is visible in the top right.

**Kết thúc.**

7. Bài tập tự làm: Xây dựng và cấu hình đề tài môn học tạo trên GitHub theo phân cấp quản lý