

GIT - GITHUB

<https://cyberlearn.vn/>

Mục Lục

01 Ver Control System

02 Git và GitHub là gì

03 Cài đặt Git

04 Khởi tạo dự án mới

05 Tham gia vào dự án có sẵn

06 Phát triển dự án

07 Branch là gì

Version Control System

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Tại sao cần quản lý mã nguồn?

➤ Các vấn đề thường gặp khi quản lý mã nguồn:

- Khi làm việc nhóm, lúc cần merge code nằm ở các máy lại với nhau ta thường dùng cách gì?
- Nếu có 2 thành viên cùng sửa trên một file, và một người vô tình xóa mất code của người kia, ta phải làm sao phục hồi lại code cũ?



➡ Cần các hệ thống giúp quản lý phiên bản mã nguồn như Git

Version Control System

- Hệ thống giúp lưu trữ source code và có thể theo dõi sự thay đổi của code.
- Hỗ trợ nhiều người dùng làm việc trên cùng 1 source
- Kiểm tra những thay đổi và biết được ai là người thay đổi
- Revert các thay đổi, đưa code trở về các phiên bản cũ, không lo bị mất code

Git và GitHub là gì?

CYBERLEARN

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Git là gì?



- Git là một trong những VCS , vốn được phát triển nhằm quản lý mã nguồn của Linux.
- Mỗi người sẽ tự làm việc trên máy tính của mình, sau khi thay đổi code, sẽ tiến hành cập nhật thay đổi lên source code chung thông qua một số quy trình
- Là một trong những VCS tốt nhất hiện tại

GitHub là gì?



- Trang web giúp lưu trữ các kho dữ liệu, có thể truy cập từ mọi nơi
- Cho phép nhiều người có thể làm việc trên cùng 1 source code

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Sơ đồ hoạt động giữa Git và GitHub

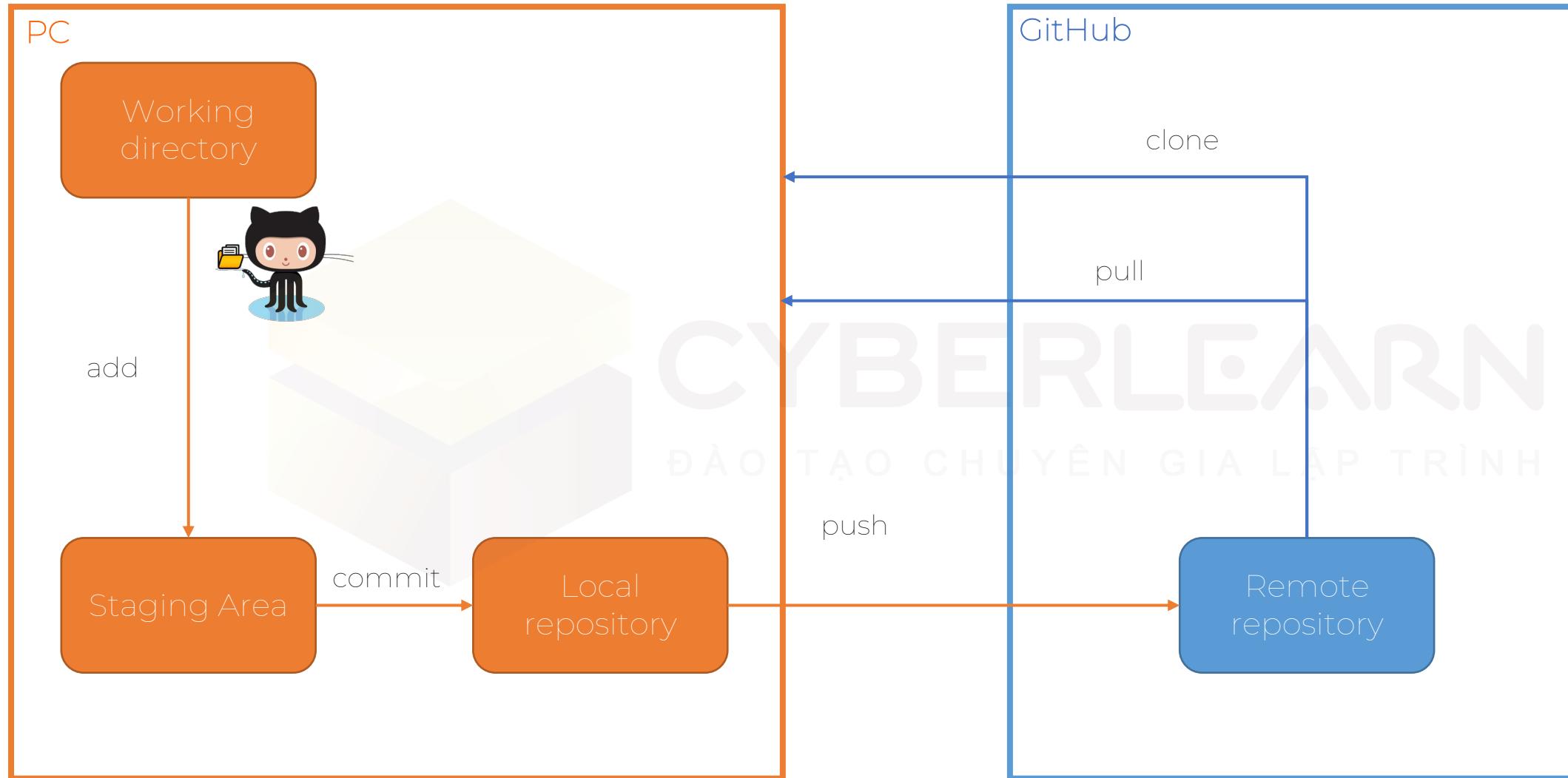
Các khái niệm cơ bản:

- Repository: kho lưu trữ source code và tất cả các thay đổi
 - Local repository : kho chứa trên máy tính chúng ta, mỗi máy sẽ là một repo
 - Remote repository : kho dữ liệu chứa trên server (kho lưu trữ online), có thể truy cập từ bất kì đâu

Sơ đồ hoạt động giữa Git và GitHub

- Staging Area: là một khu vực chứa thông tin các files sẽ được chuẩn bị cho quá trình commit
- Commit: mỗi lần thay đổi source code, tiến hành commit để tạo ra 1 phiên bản lưu trữ lại sự thay đổi, khi có lỗi thì có thể phục hồi code trở về version trước đó

Sơ đồ hoạt động giữa Git và GitHub



Thực hành Git

CYBERLEARN

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

01

Cài đặt Git

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Cài đặt Git

- Mac:

Đối với macOS cần cài đặt theo 1 trong các cách ở link bên dưới.

<https://git-scm.com/download/mac>

- Window

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

Công cụ

- Git bash here (Window)
- Terminal (Mac)
- Terminal của VS code
- Phần mềm desktop hỗ trợ git: GitHub Desktop, TortoiseGit, Sourcetree, GitKraken

Thiết lập cơ bản

Lần đầu tiên sử dụng git cho laptop, cần khai báo các thiết lập bên dưới.

- Cấu hình name và email

```
git config --global user.name "Your name"  
git config --global user.email "your@email.com"
```

Nếu muốn thiết lập lại name hoặc email mới thì chỉ cần gõ lại lệnh trên

- Kiểm tra sau khi cấu hình

```
git config --global user.name  
git config --global user.email
```



02

Khởi tạo dự án mới

CYBERLEARN

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Khởi tạo dự án mới

- Đối với người khởi tạo dự án, thực hiện các quy trình sau:
 1. Tạo remote repository (GitHub)
 2. Tạo local repository (PC)
 3. Kết nối local repository với remote repository
 4. Khai báo các files chuẩn bị đưa lên remote repository
 5. Xác nhận các files cần đưa lên remote repository
 6. Đẩy lần đầu lên remote repository

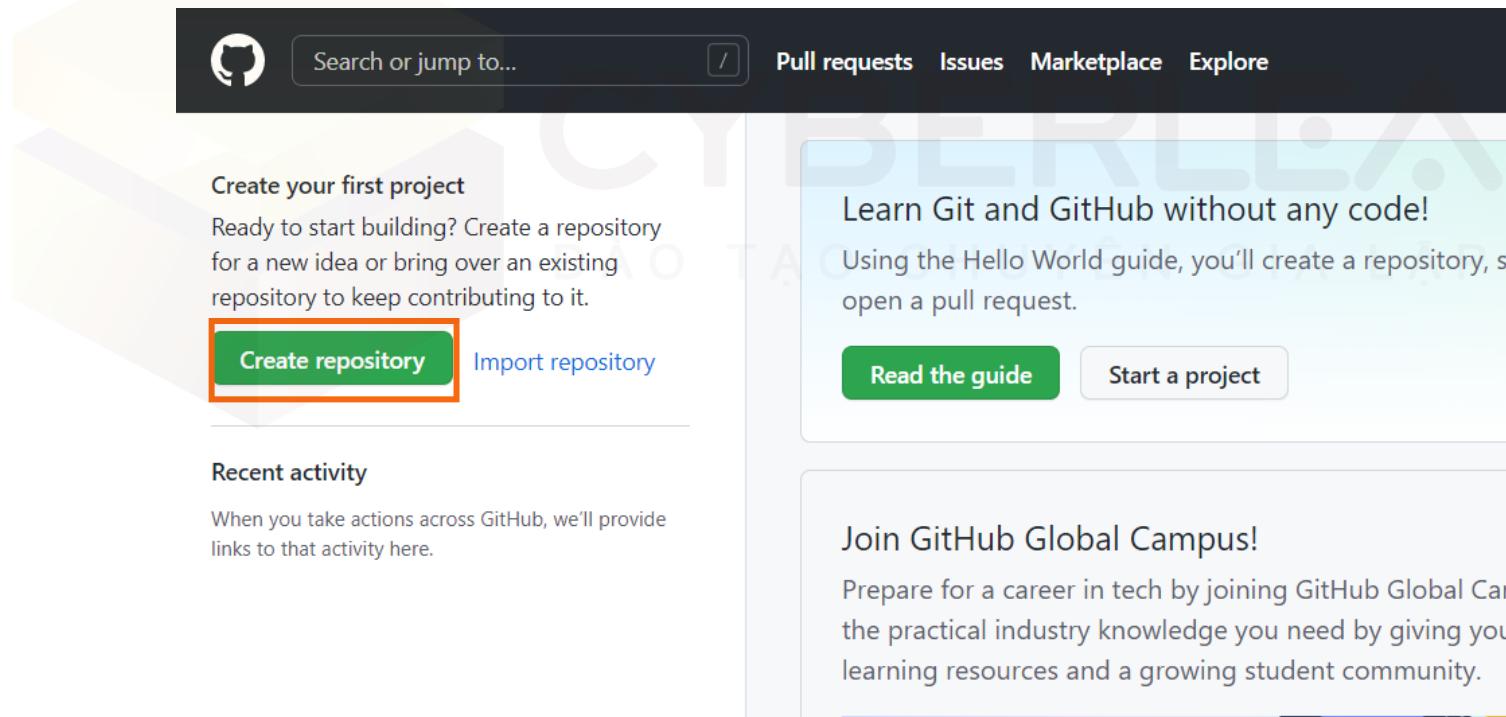


Tạo dự án trên GitHub

- Bước 1: Đăng ký 1 tài khoản trên Github

<https://github.com/signup>

- Bước 2: Tạo new repository (kho lưu trữ mới)



Tạo dự án trên GitHub

- Bước 3: Đặt tên và tạo repository (xét các cài đặt như hình bên dưới)

Owner * Repository name *

 khloe1425 / GitProject ✓

Great repository names are short and memorable. Need inspiration? How about [laughing-octo-system](#)?

Description (optional)

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

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

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more](#).

Add .gitignore
Choose which files not to track from a list of templates. [Learn more](#).

Choose a license
A license tells others what they can and can't do with your code. [Learn more](#).

Create repository

Tạo dự án ở local và đẩy lên GitHub



Khởi tạo repository local
`git init`



Khai báo các file cần thêm vào local repository
`git add [tên file]` (thêm file cụ thể) hoặc `git add .` (thêm tất cả files)



Xác nhận files cần thêm và lý do thêm files
`git commit -m "lý do"`



kết nối local repository với remote repository
`git remote add origin [đường dẫn trên github]`



Đẩy files lên remote repository
`git push -u origin master`

Sau lệnh push, git sẽ yêu cầu nhập username và pass tài khoản github
(yêu cầu này chỉ xuất hiện khi đẩy lên github lần đầu, các lần sau git sẽ tự ghi nhớ tài khoản)

03

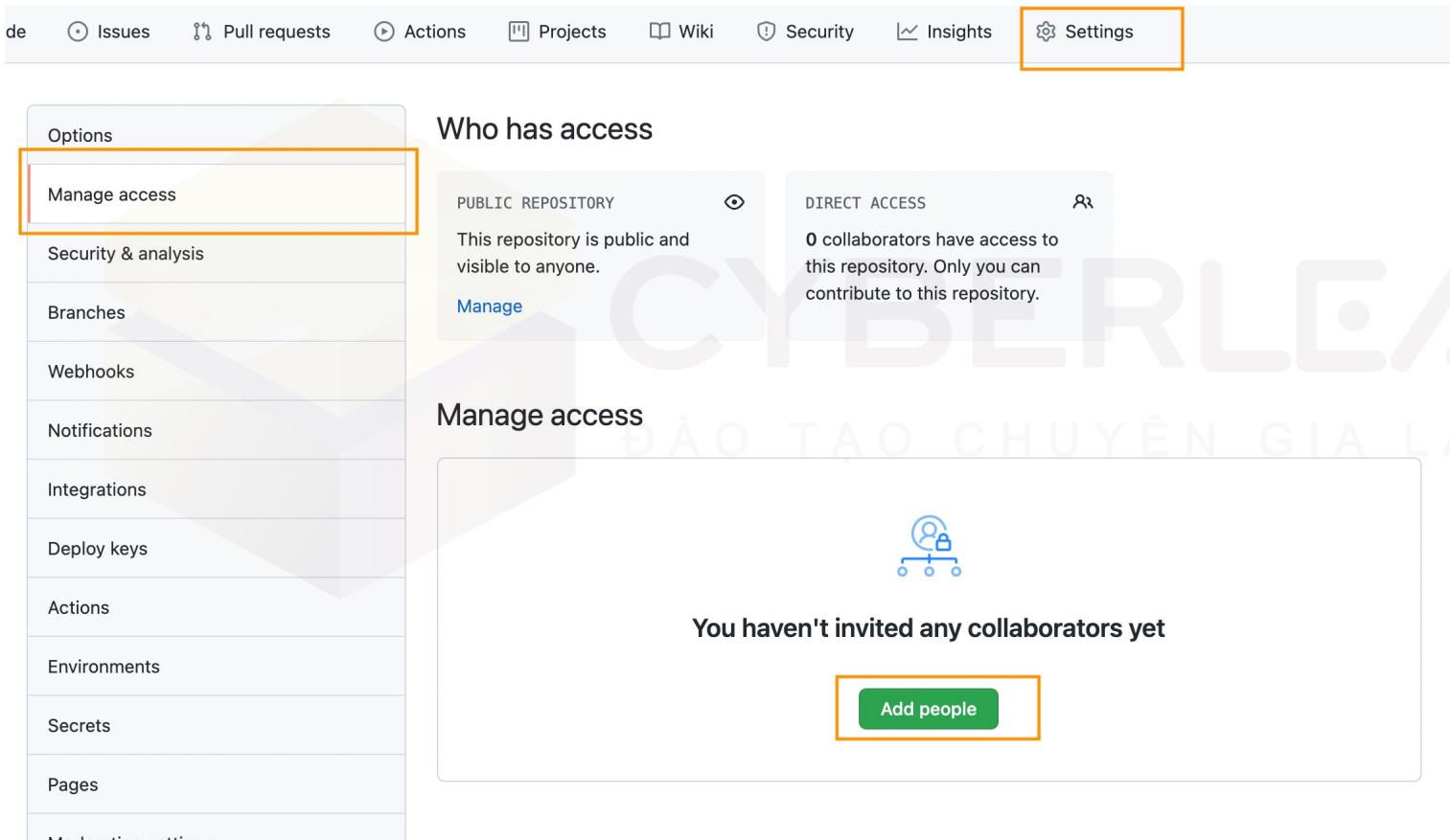
Tham gia vào dự án
có sẵn

CYBERLEARN

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

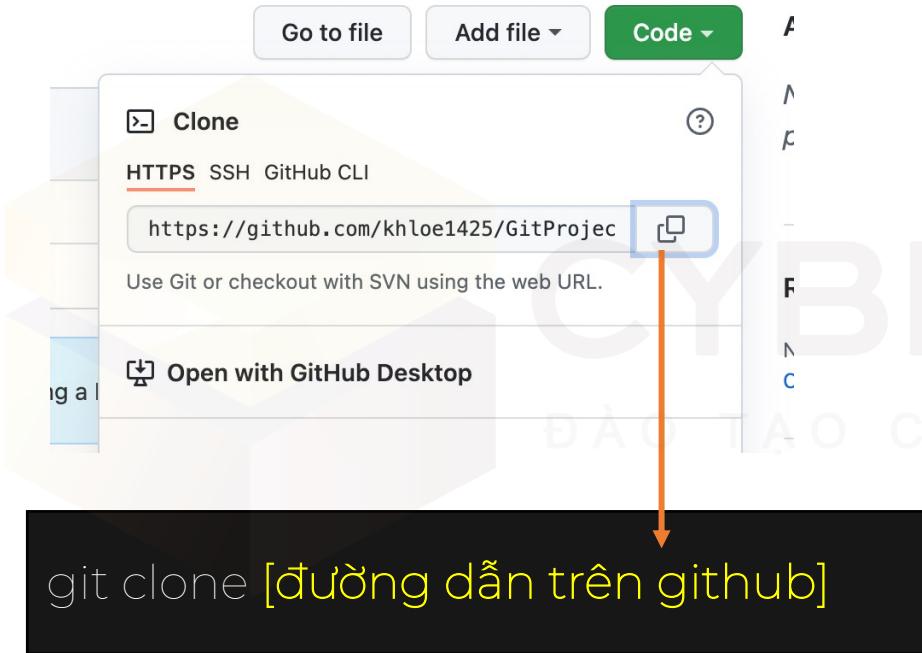
Tham gia vào dự án có sẵn

1. Người tạo dự án sẽ mời user vào dự án trên GitHub



Tham gia vào dự án có sẵn

2. Người tham gia sao chép dự án về local





04

Phát triển dự án

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Phát triển dự án

- Trước khi tiến hành chỉnh sửa phải cập nhật local repository để đảm bảo các files của local repo và remote repo giống nhau

```
git pull
```

- Sau khi chỉnh sửa, tiến hành đẩy thay đổi lên remote repository (GitHub)

```
git add [tenfile]
```

```
git commit -m "lý do thay đổi"
```

```
git push
```

So sánh clone và pull

- git clone

Lệnh này sẽ sao chép toàn bộ dữ liệu và thiết lập trên remote repository, tức là nó sẽ tự động tạo một local repository trên máy tính của bạn sau khi chạy lệnh.

Lệnh này chỉ nên sử dụng khi bạn cần tạo mới một dự án Git trên máy tính.

- git pull

Lệnh này sẽ chỉ lấy những dữ liệu có sự thay đổi từ remote repository đem về cập nhật cho local repository

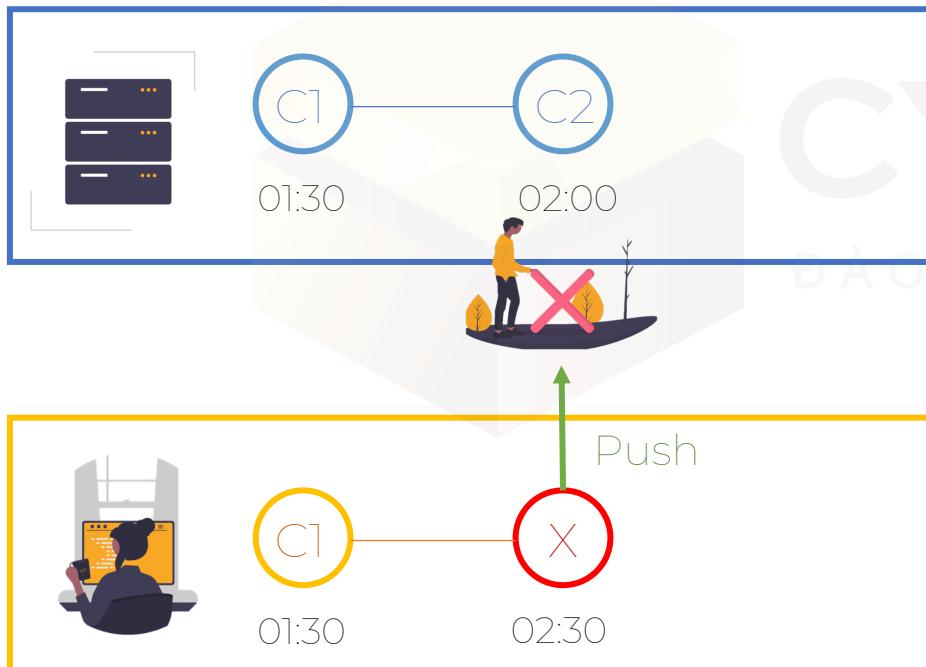
04

Branch là gì

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

➤ Lỗi xung đột (conflict) khi push

Trong khi bạn đang chỉnh sửa file ở bên dưới local repository của bạn, mà có người khác cũng chỉnh sửa cùng file dữ liệu với bạn và lỡ push lên remote repository, thì đến khi bạn push sẽ bị từ chối và xuất hiện lỗi như bên dưới:



```
MacBook-Pro-čua-THANH:GitProject thanhnguyen$ git push  
To https://github.com/khloe1425/GitProject.git  
! [rejected]          master -> master (non-fast-forward)  
error: failed to push some refs to 'https://github.com/khloe1425/GitProject.git'  
hint: Updates were rejected because the tip of your current branch is behind  
hint: its remote counterpart. Integrate the remote changes (e.g.  
hint: 'git pull ...') before pushing again.  
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

➤ Giải quyết xung đột khi push



Kéo commit mới nhất từ remote repository về cập nhật local repository
git pull



Xử lý merge file

Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
`<<<< HEAD (Current Change)`
`<title>Document123</title>`
`=====`
`<title>Document456</title>`
`>>>>> 30ec1d482e1cf4b0a7903f141f3a51c32e7ffdf5 (Incoming Change)`



Thực hiện đẩy lại file lên remote repository

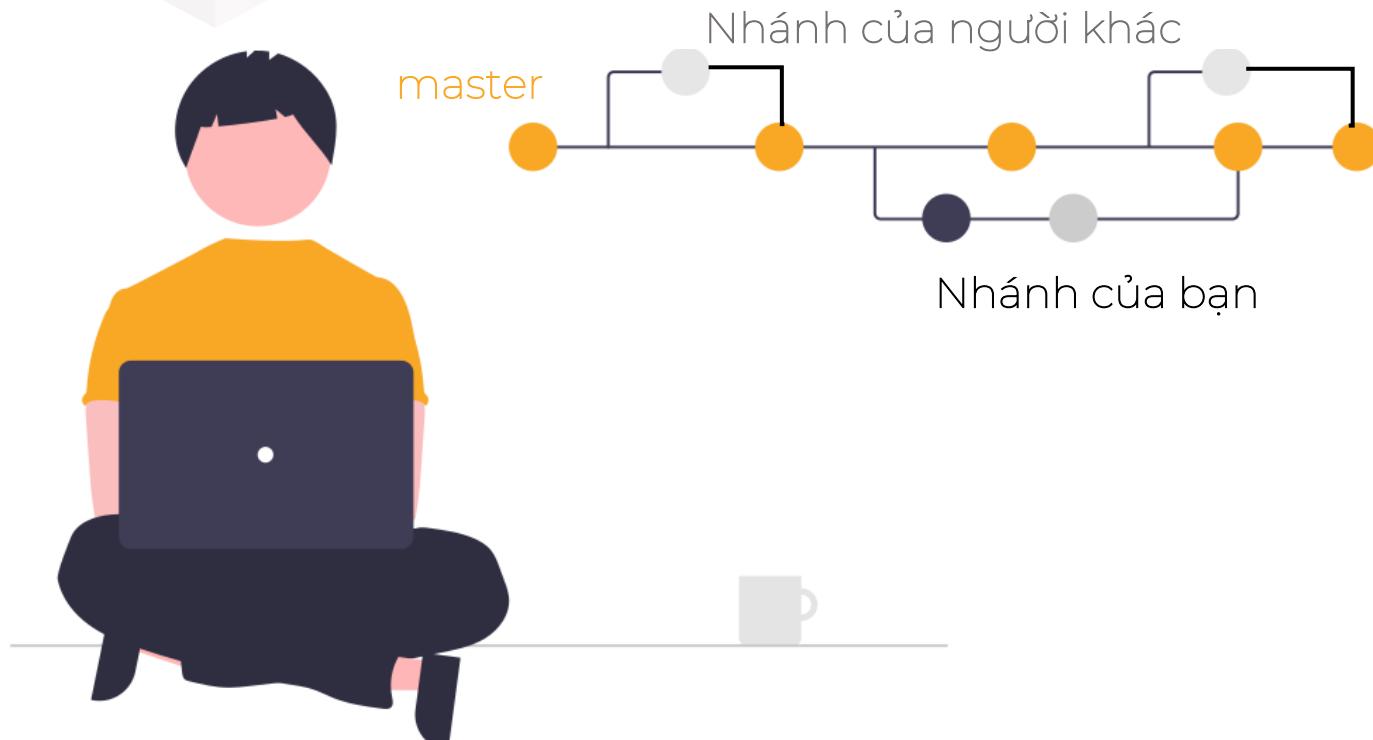
git add -A

git commit -m "lý do commit"

git push

➤ Làm việc với branch để hạn chế xung đột

- Mặc định khi làm việc với git, ta sẽ làm việc trên nhánh (branch) master
- Khi có quá nhiều người cùng làm việc trên một nhánh chính (master), dễ dẫn tới xung đột khi push hoặc đè mất code của nhánh chính
- Ta phải tiến hành chia nhánh, mỗi nhánh thực hiện một chức năng khác nhau, sau đó gộp lại vào nhánh chính, để giữ nhánh chính được toàn vẹn



➤ Các lệnh cơ bản với branch

- Tạo branch

```
git branch [tennhanh]
```

- Chuyển branch

```
git checkout [tennhanh]
```

- Tạo và chuyển branch

```
git checkout -b [tennhanh]
```

- Đẩy nhánh mới lên remote repository

```
git push  
git push --set-upstream origin [tennhanh]
```

- Xoá branch

```
git branch -D [tennhanh]
```

- Kiểm tra nhánh

```
git branch
```

➤ Các bước tạo và đẩy nhánh lên remote repository

1. Tạo branch

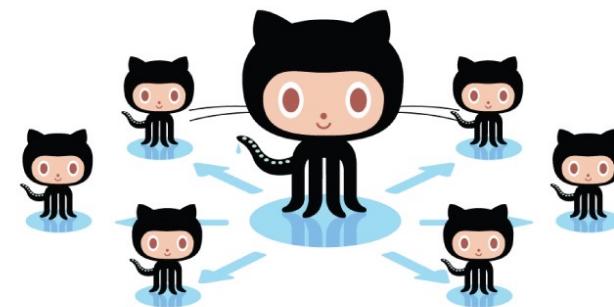
Tên branch thường đặt theo tên user, tên function, tên version của dự án

2. Chuyển sang branch mới tạo

3. Kiểm tra nhánh đang sử dụng

4. Đẩy nhánh lên remote

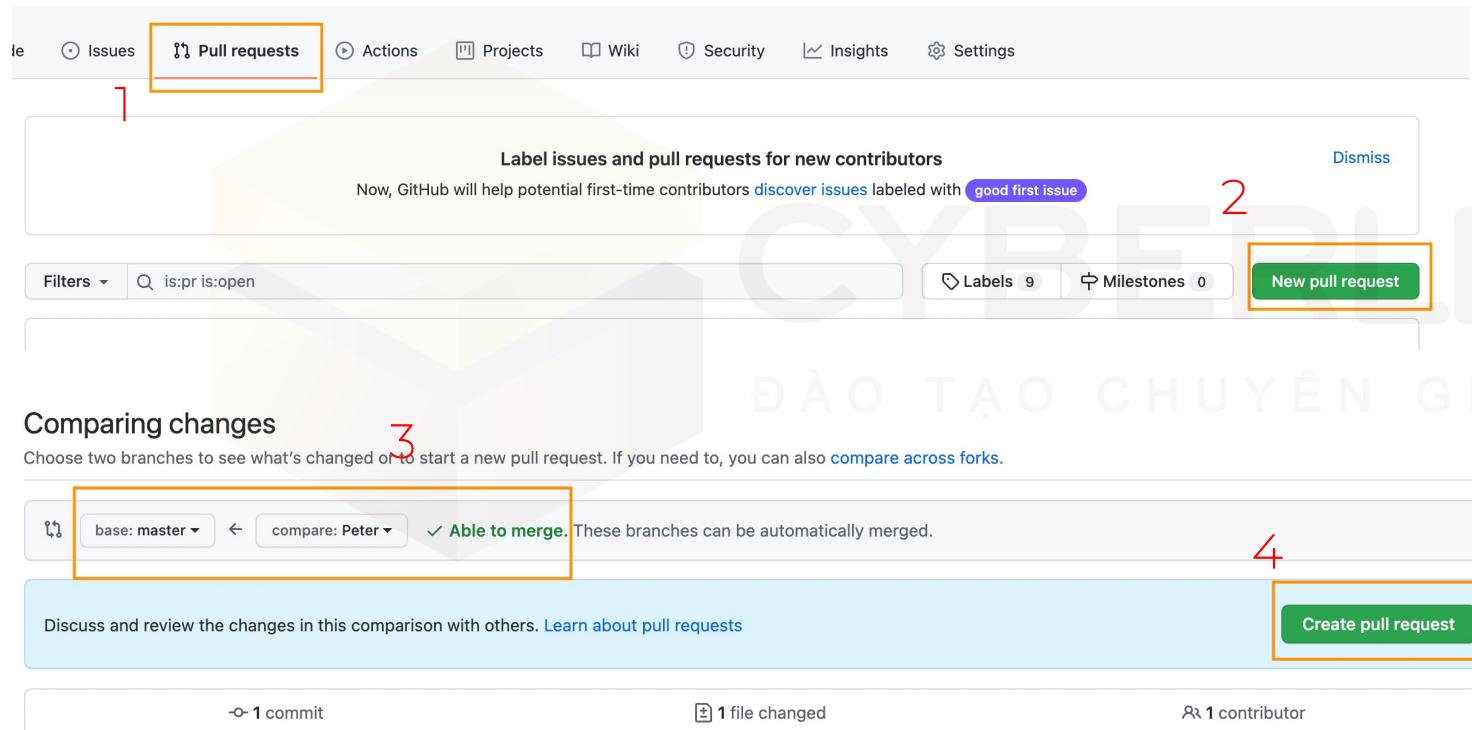
Khi tạo branch, dữ liệu từ master sẽ được sao chép sang nhánh mới ở thời điểm tạo.



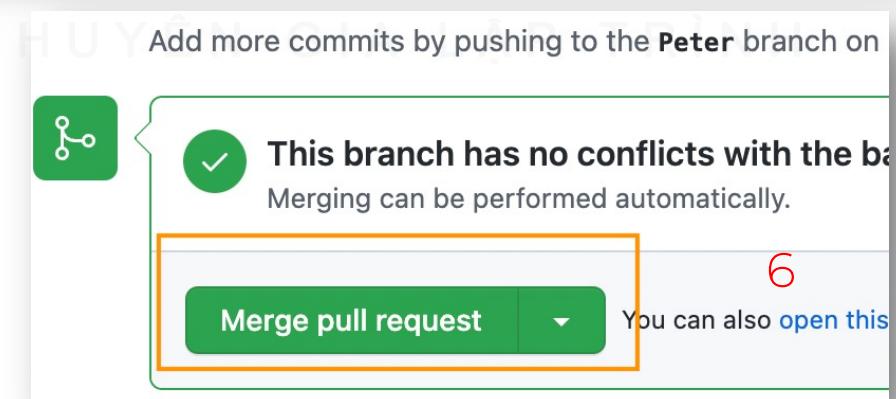
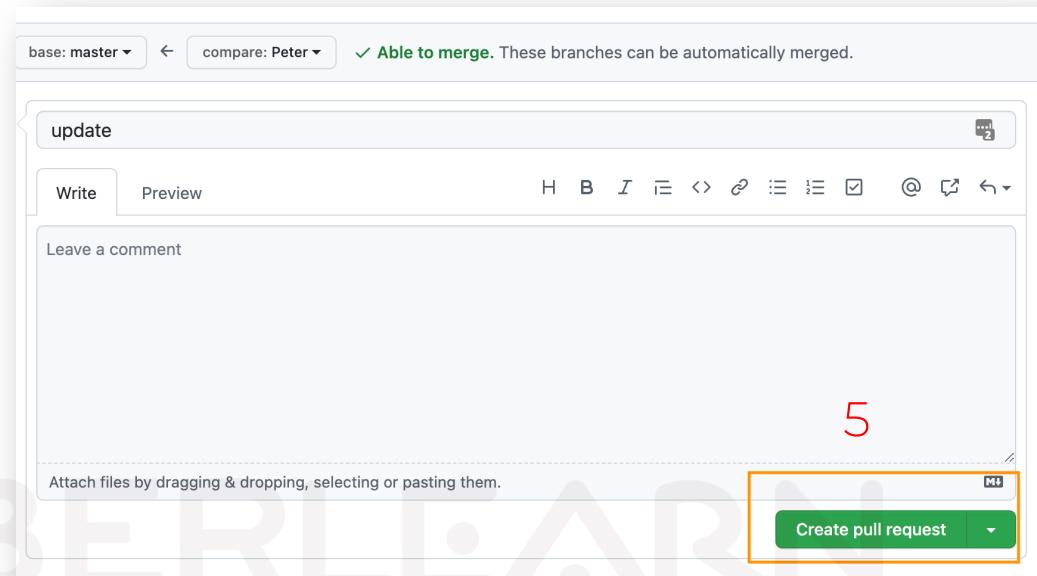
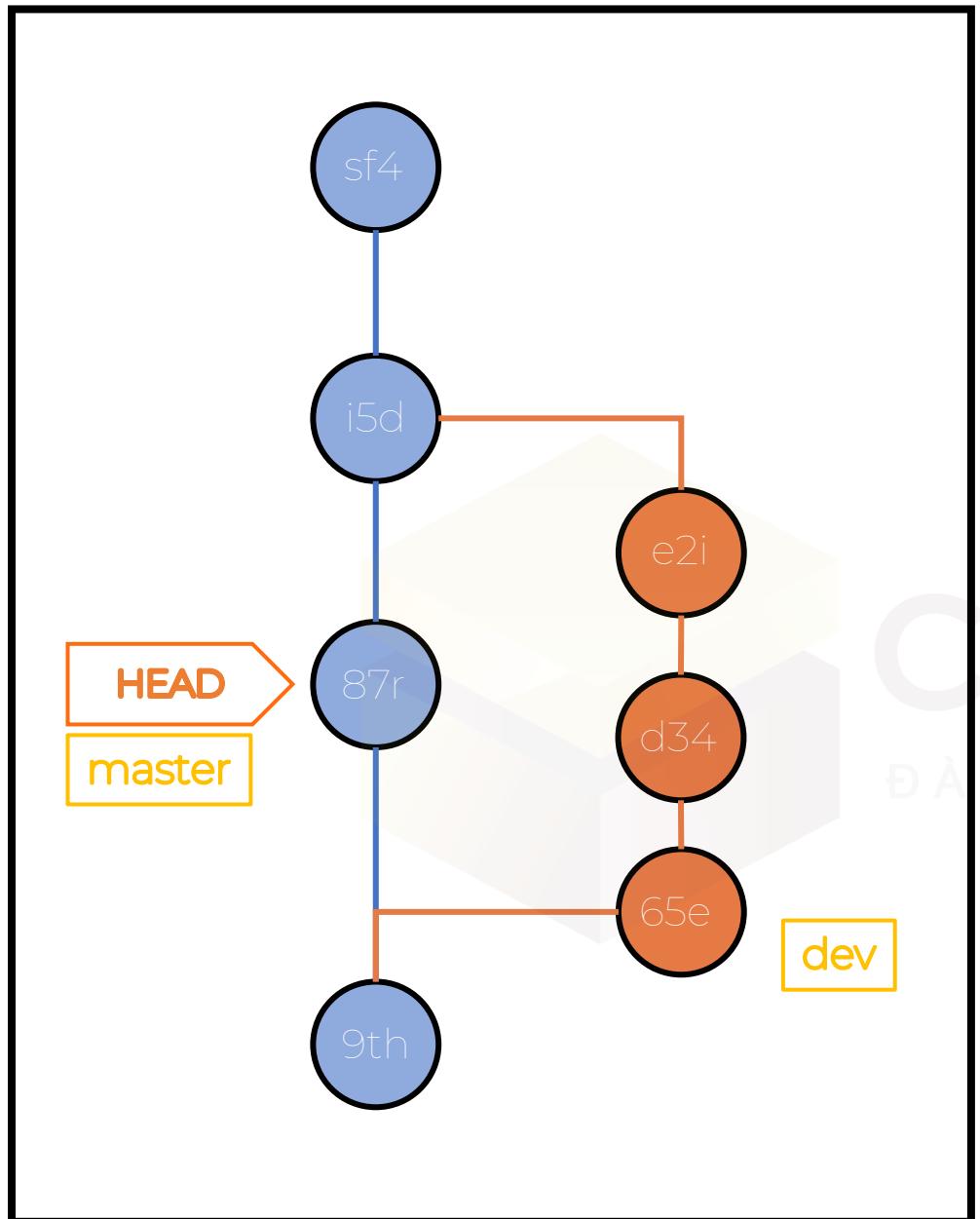
Merge nhánh

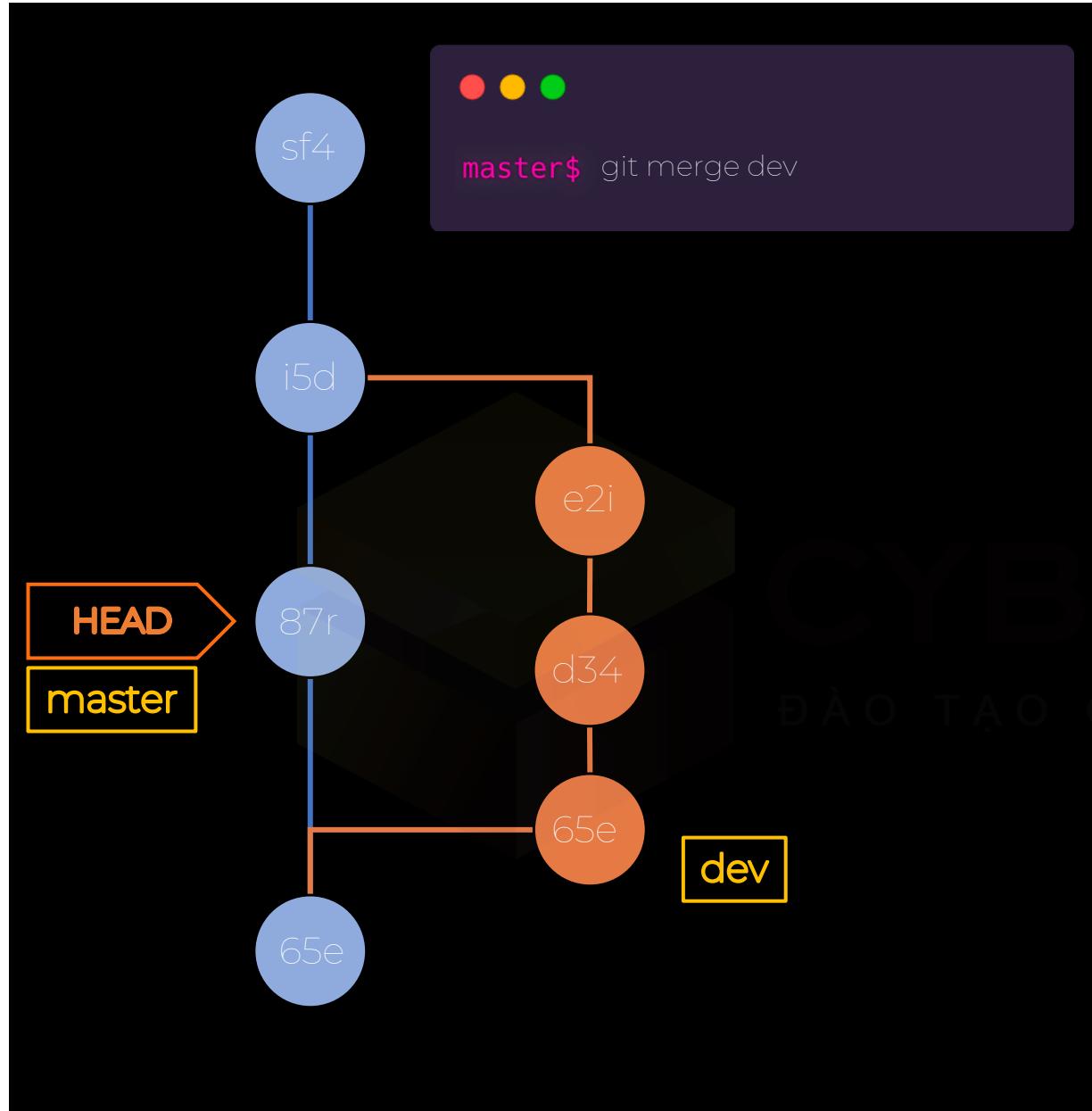
Sau khi hoàn thành thay đổi ở nhánh riêng, ta sẽ tiến hành gộp nhánh. Để merge gộp nhánh riêng vào nhánh master, ta có các cách sau

- Cách 1: Tạo pull request trên github và tiến hành merger nhánh



Merge nhánh



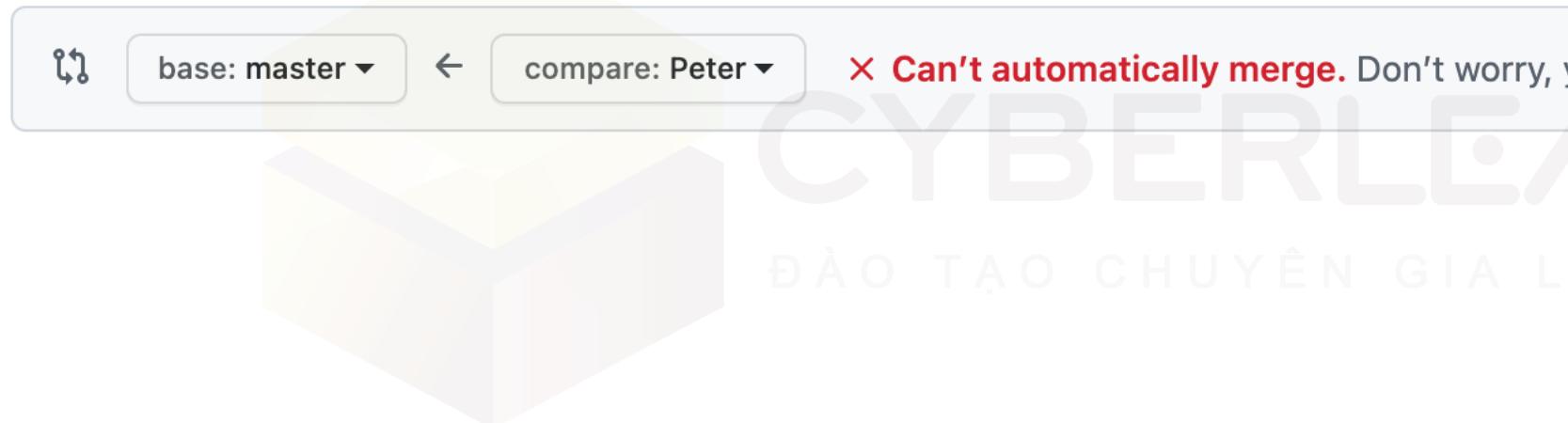


Merge nhánh

- Cách 2: Merge nhánh bằng lệnh
 - 1. Khi merge vào master phải checkout về master
 - 2. Tiến hành merge
 - 3. Commit master lên remote repository

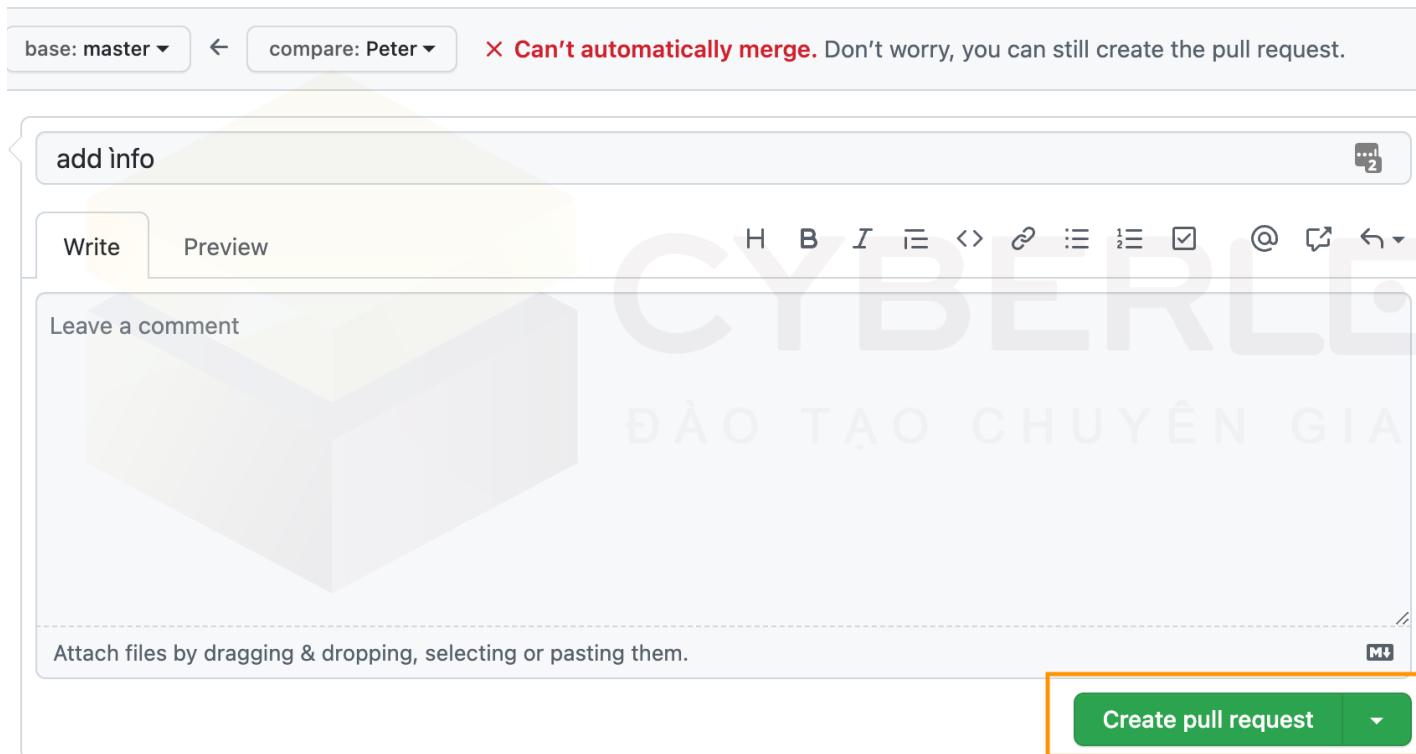
Xung đột khi merge nhánh

Xung đột xảy ra khi nhánh master bị nhánh khác merge vào và làm thay đổi, nhưng nhánh của bạn lại không cập nhật thay đổi đó. Sau khi bạn hoàn thành chức năng và cần merge vào master thì sẽ bị từ chối merge



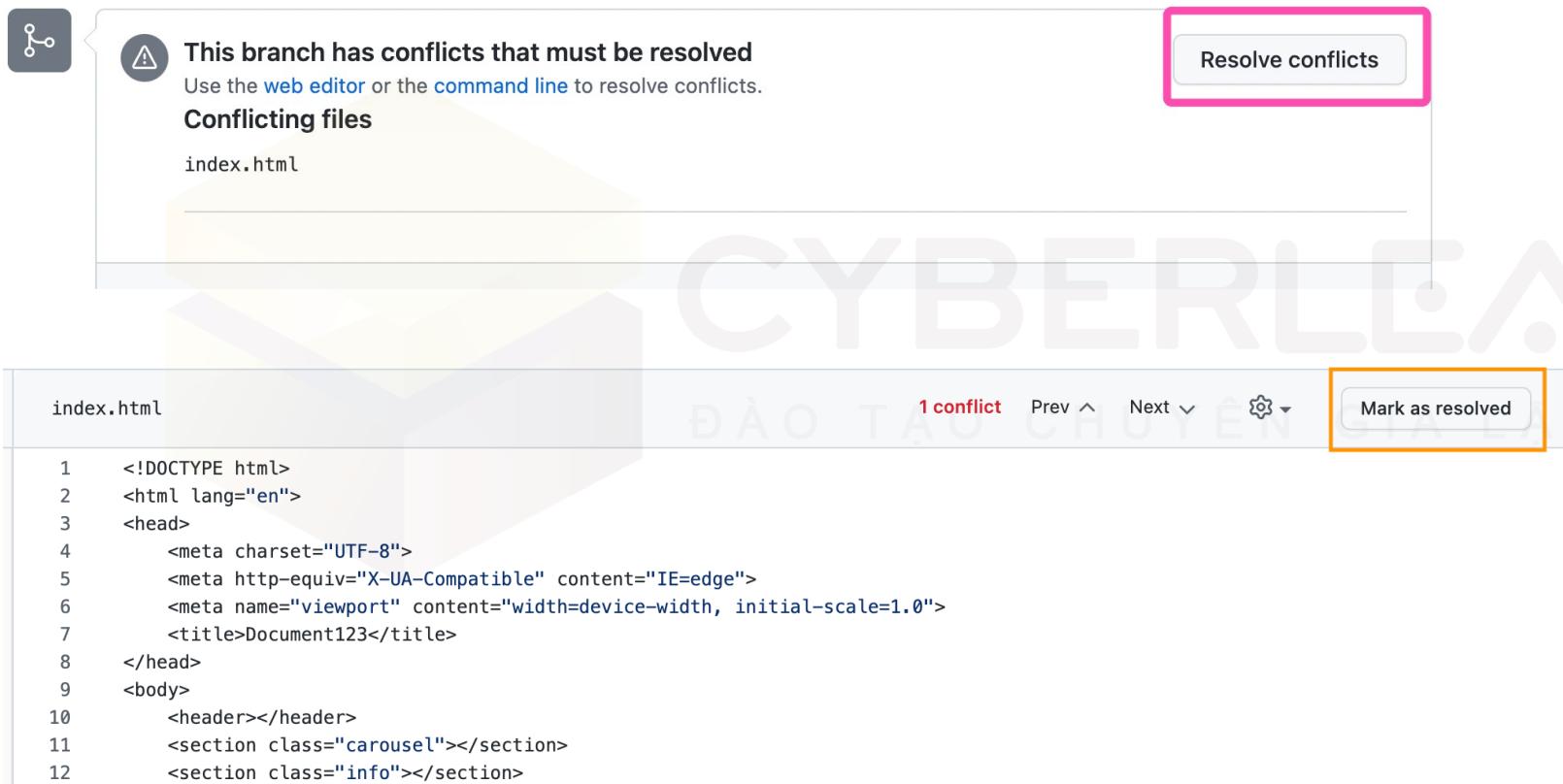
Merge nhánh

- Giải quyết xung đột khi merge bằng GitHub
- 1. Vẫn cứ tạo pull request khi thấy báo lỗi merge



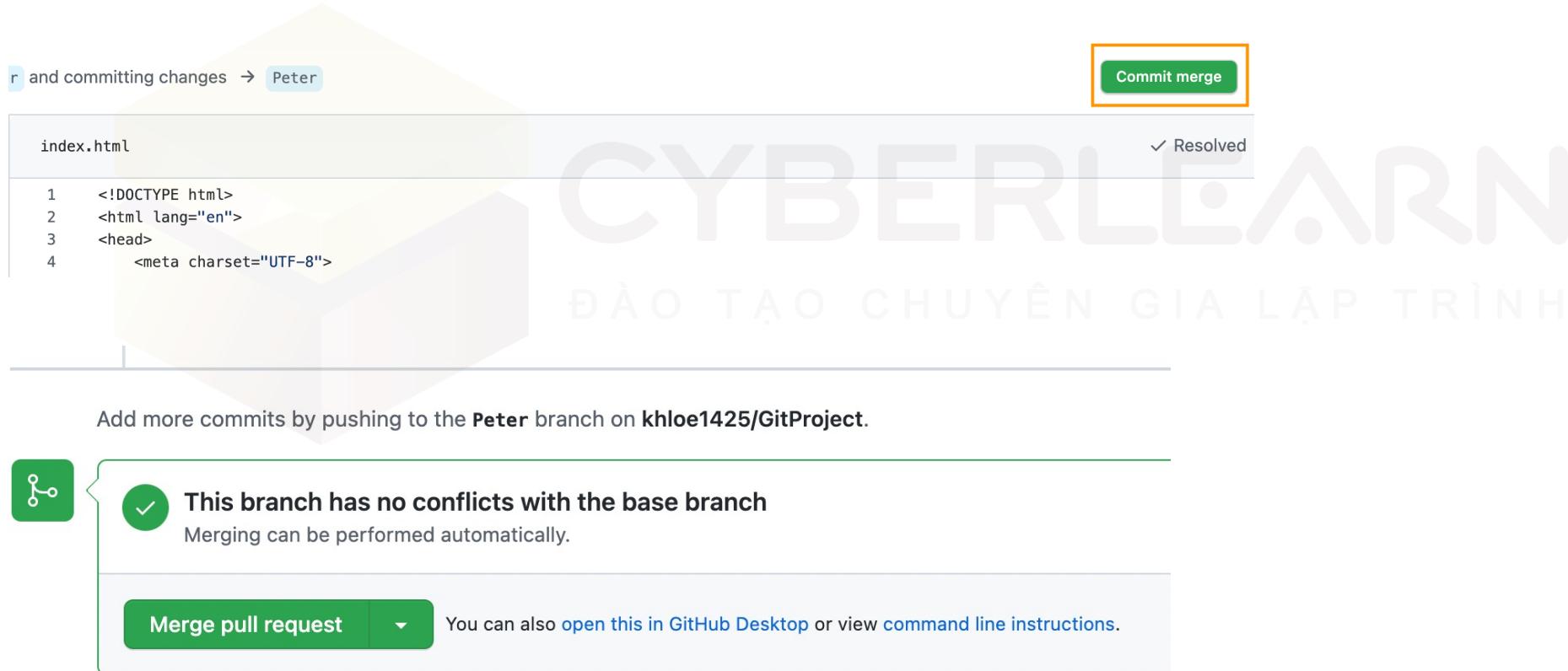
Merge nhánh

- Chọn Resolve conflict để sửa xung đột. Sau khi sửa các chỗ xung đột thì chọn Mask as resolve



Merge nhánh

3. Commit merge và tiến hành merge như bình thường
4. Sau khi merge thành công nhớ git pull cập nhật thay đổi về cho master local



Merge nhánh

Ngoài cách sửa xung đột bằng công cụ Resolved của GitHub, bạn có thể sửa xung đột bằng VS code.

1. Cập nhật code mới từ remote master về local branch của bạn
`git pull origin master`
2. So sánh và xử lý xung đột code ở VS code
3. Commit code sau khi sửa lên remote branch riêng của bạn
4. Tiến hành tạo pull request trên GitHub và merge vào master

Merge nhánh

➤ Giải quyết xung đột khi merge bằng lệnh

1. Cập nhật cho nhánh master ở local

```
git pull
```

2. Chuyển nhánh sang nhánh riêng và cập nhật code mới từ remote master về local branch của bạn

```
git pull origin master
```

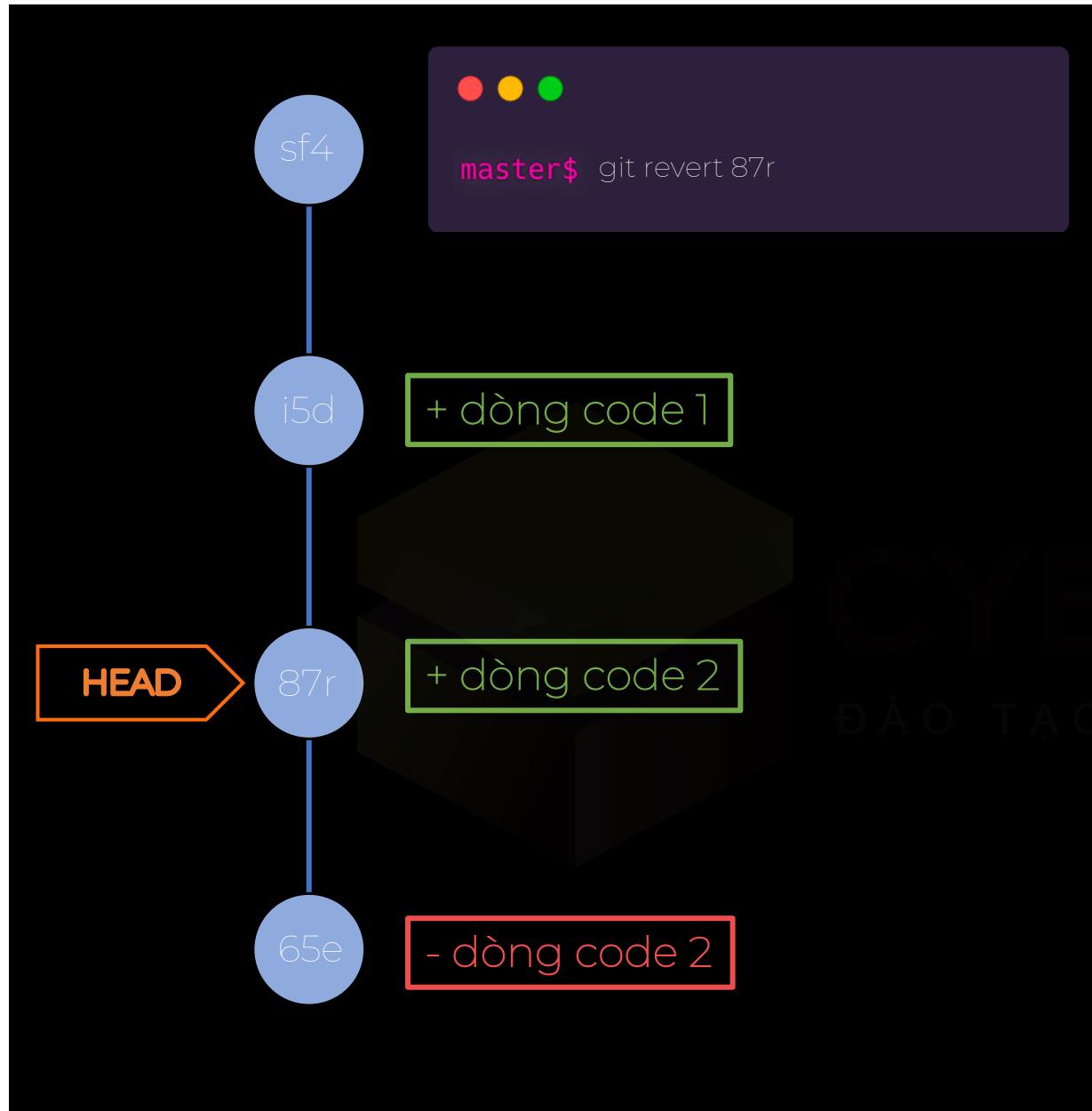
3. So sánh và xử lý xung đột code ở VS code

4. Chuyển nhánh sang master và tiến hành merge

5. Sau khi merge, đẩy thay đổi lên master ở remote

Revert

- Revert sẽ tạo một commit mới, trong đó sẽ loại bỏ thay đổi của commit cũ.



04

Deploy website

CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH



Các khái niệm về web và internet

➤ Internet

- Là mạng máy tính toàn cầu kết nối các mạng máy tính khắp nơi trên thế giới.
- Các máy tính giao tiếp với nhau trên internet thông qua tập giao thức gọi chung TCP/IP với mô hình 4 tầng.

➤ Intranet

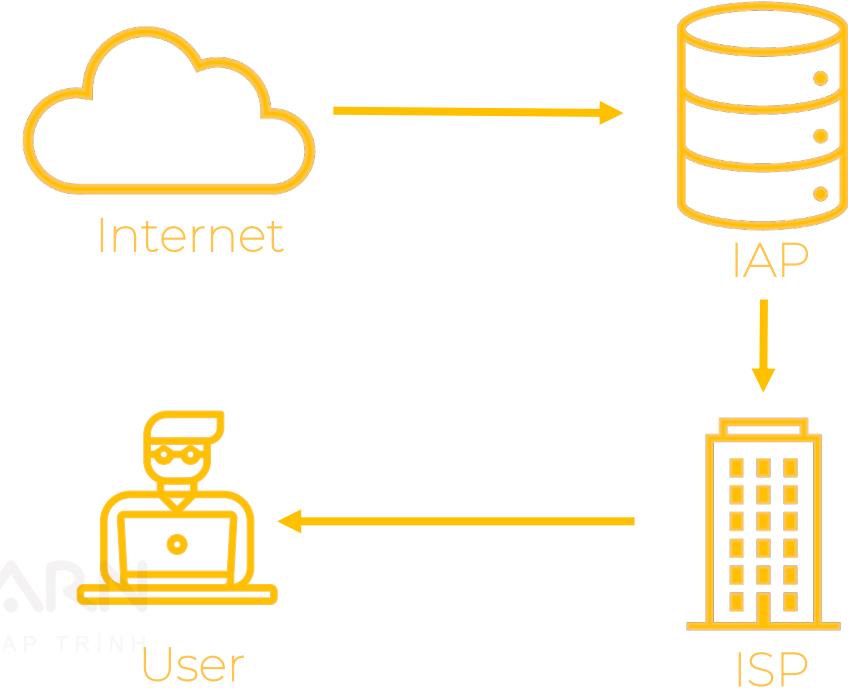
- Là mạng máy tính cục bộ có kiến trúc như internet.
- Được dùng trong các doanh nghiệp, trường học...
- Các thành viên muốn sử dụng phải có tài khoản.

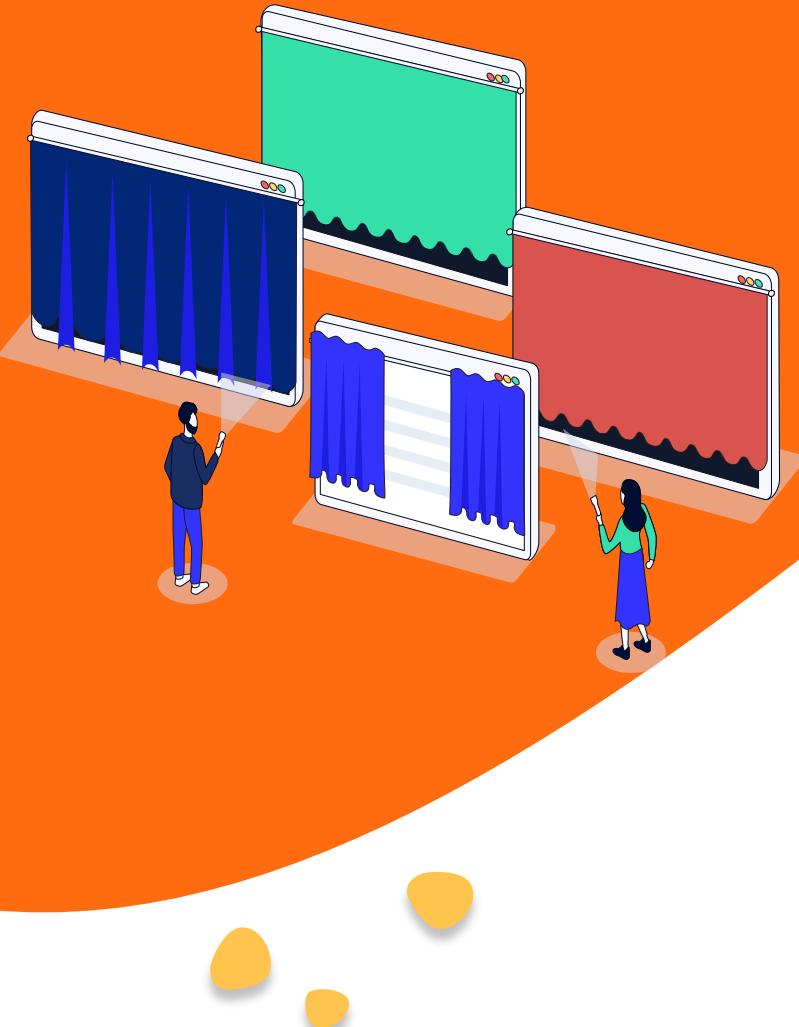
CYBERLEARN

ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Các khái niệm về web và internet

- IAP (Internet Access Provider)
Nhà cung cấp đường truyền để kết nối vào Internet
- ISP (Internet Service Provider)
Nhà cung cấp dịch vụ internet. Một số ISP ở Việt Nam hiện nay :FPT, Viettel, VDC, CMC, SPT





Các khái niệm về web và internet

➤ Webpage

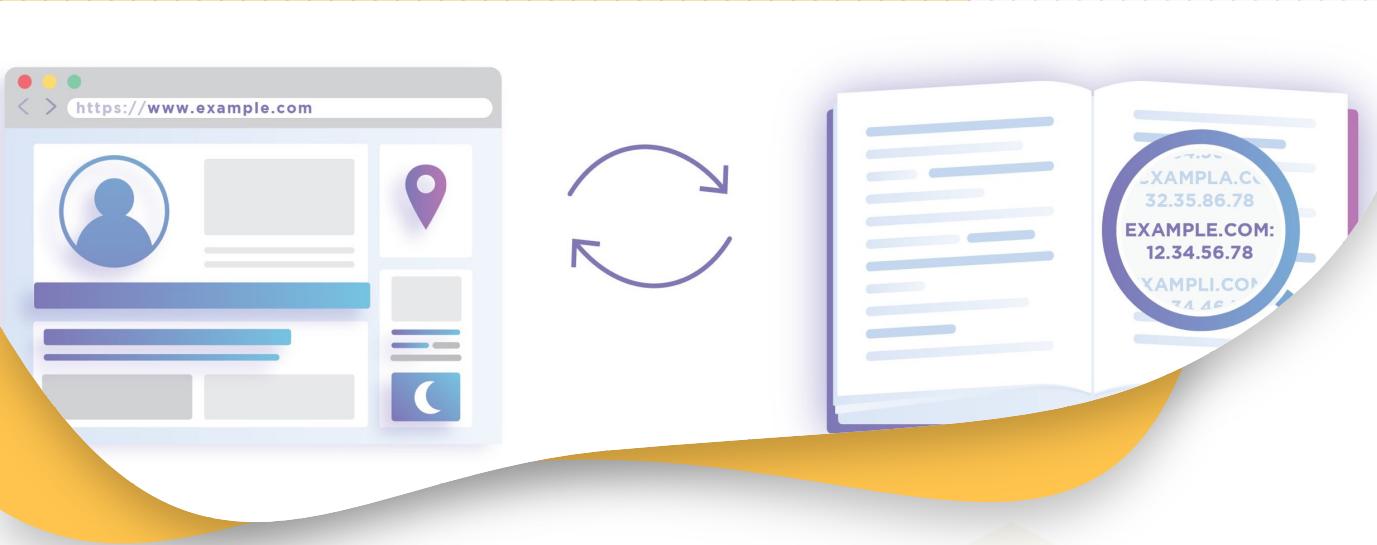
- Là một trang web hiển thị thông tin dưới dạng văn bản, âm thanh, hình ảnh.
- Các trang web có thể được liên kết với nhau.

➤ Website

- Là tập hợp những webpage có liên quan đến nhau để tạo thành một bộ web.

➤ Homepage

- Là trang chủ. Thường là trang được hiển thị đầu tiên khi ta truy cập vào website.



Các khái niệm về web và internet

➤ Địa chỉ IP

- Cung cấp nhận dạng cho một thiết bị sử dụng internet.
- Là 1 con số 32bit, thường được viết bằng dạng thập phân
- VD: 192.168.45.15..

➤ Tên miền - Domain Name

- Người dùng muốn xem 1 trang web phải truy cập vào địa chỉ IP của máy chủ chứa trang web đó.
- Thay vì sử dụng địa chỉ IP, chúng ta sử dụng tên miền thay thế để dễ nhớ.

CYBERLEARN
BÁO TAO CHUYÊN GIA LẬP TRÌNH

Các khái niệm về web và internet

➤ DNS – Domain name system

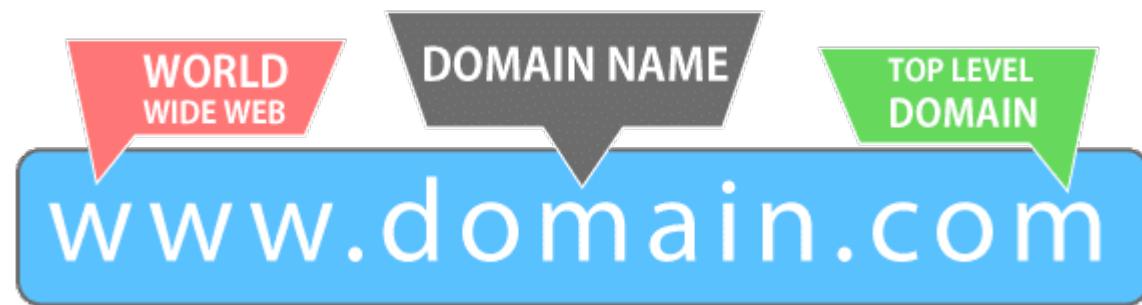
Hoạt động như một danh bạ điện thoại, lưu các địa chỉ IP dưới dạng domain name.



Các khái niệm về web và internet

➤ URL

- Mỗi tài nguyên được lưu trên internet sẽ được cấp cho một đường dẫn url
- URL là đường dẫn giúp chúng ta tìm tới được trang web



Các khái niệm về web và internet

➤ Trình duyệt web (Browser)

- Là công cụ cho phép người dùng tương tác với các văn bản, hình ảnh, đoạn phim trên trang web.
- Đọc định dạng HTML và hiển thị ra màn hình.
- Thiết lập kết nối TCP/IP gửi yêu cầu lên webserver.
- Nhận dữ liệu trả về từ webserver và hiển thị lên trình duyệt
- Một số trình duyệt phổ biến: Chrome, Firefox, IE,....

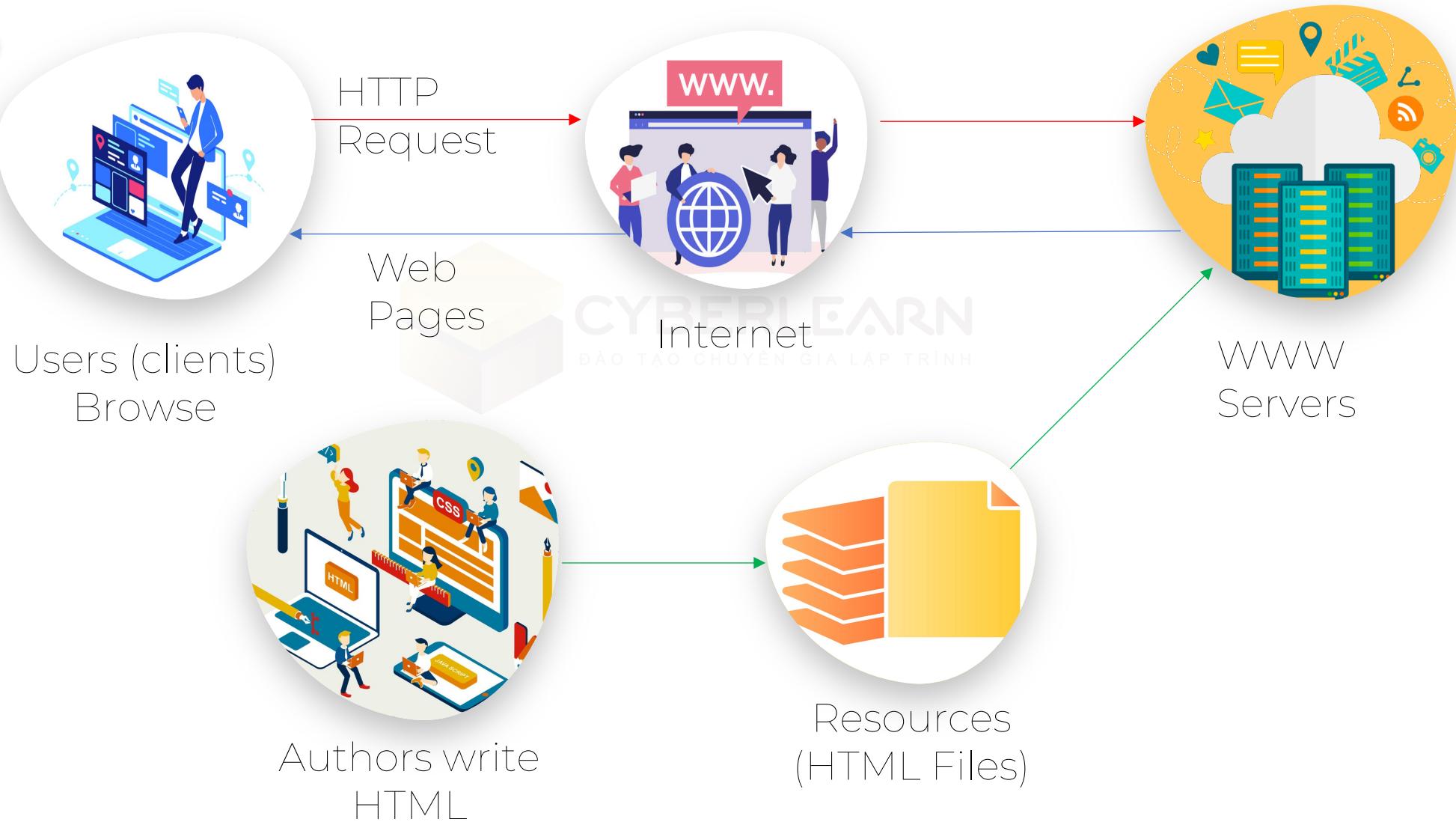




Các bước xây dựng Web

- Bước 1: Xác định yêu cầu website cần xây dựng
- Bước 2: Chọn và mua tên miền
- Bước 3: Thuê chỗ Hosting
- Bước 4: Thiết kế website
- Bước 5: Triển khai website
- Bước 6: Cập nhật thông tin, bảo trì website

Cách thức hoạt động của web



Deploy bằng github

1. Chọn tab Settings -> Pages
2. Kế đó chọn branch muốn deploy -> Save

The screenshot shows the GitHub repository settings page. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The Settings tab is active, indicated by a red underline. On the left, a sidebar lists General, Access, Collaborators, Moderation options, Code and automation (with sub-options for Branches, Tags, Actions, Webhooks, Environments, Codespaces, and Pages), and GitHub Pages. The GitHub Pages section is titled "GitHub Pages" and contains a brief description: "GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository." Below this, under "Build and deployment", there is a "Source" dropdown set to "Deploy from a branch". A "Branch" dropdown shows "main" selected, and a "Save" button is visible. At the bottom, the "Visibility" section notes that with a GitHub Enterprise account, access can be restricted by publishing privately.

ode Issues Pull requests Actions Projects Wiki Security Insights Settings

General GitHub Pages

Access

Collaborators

Moderation options Build and deployment

Code and automation

Branches Deploy from a branch

Tags

Actions

Webhooks

Environments

Codespaces

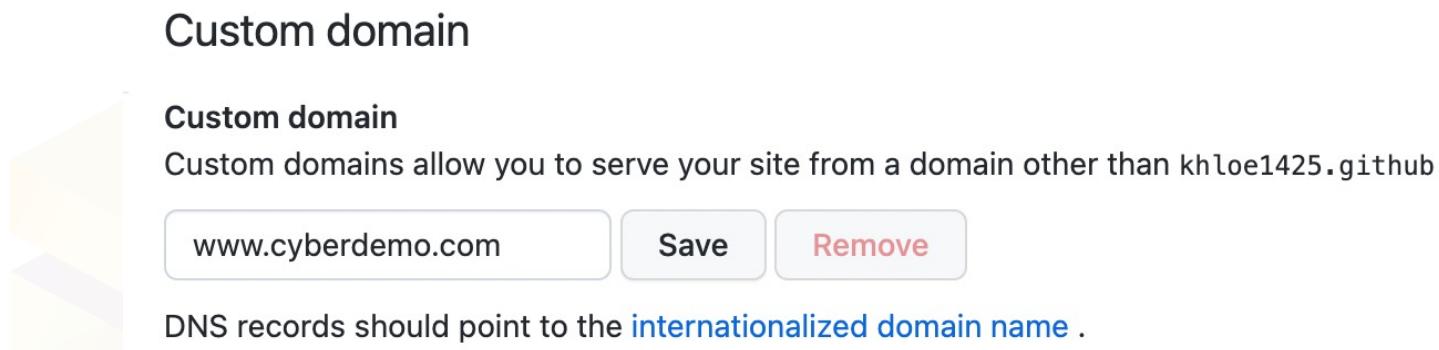
Pages

Visibility GITHUB ENTERPRISE

With a GitHub Enterprise account, you can restrict access to your GitHub Pages site by publishing it privately. A privately published site can only be accessed by people with read access to the repository the site is published from.

Deploy bằng github

- Đặt tên mới cho domain nếu bạn không muốn dùng tên mặc định. Nhấn Save sau khi điền tên demomain mới.



Kết quả sau khi deploy

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <http://www.cyberdemo.com/>

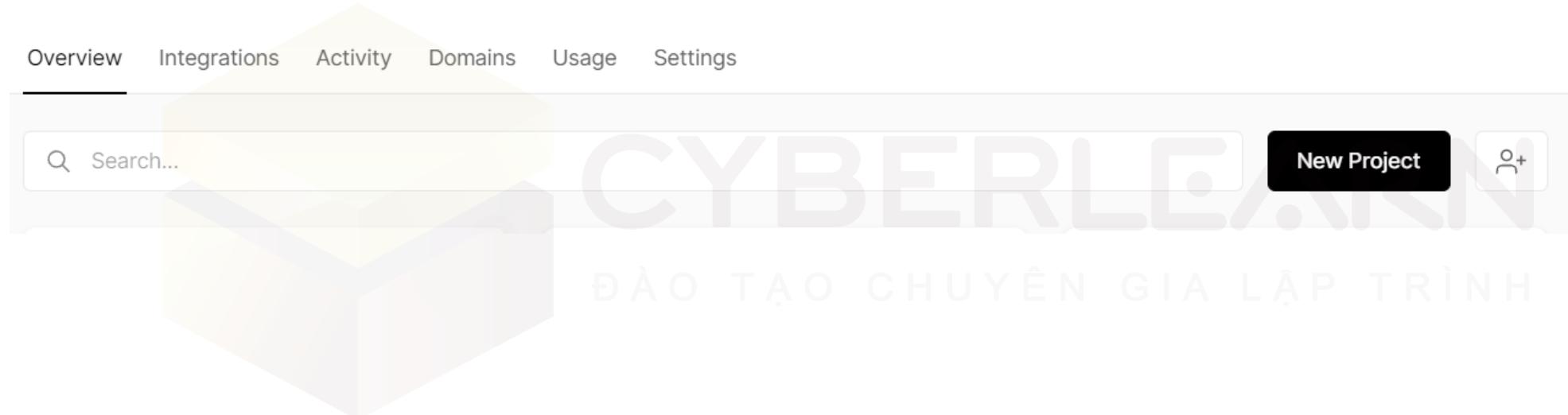
Last deployed by khloe1425 1 minute ago

[Visit site](#)

...

Deploy bằng vercel

1. Lưu project trên github
2. Tạo tài khoản Vercel: <https://vercel.com/>
3. Ở Dashboard, click New Project



Deploy bằng vercel

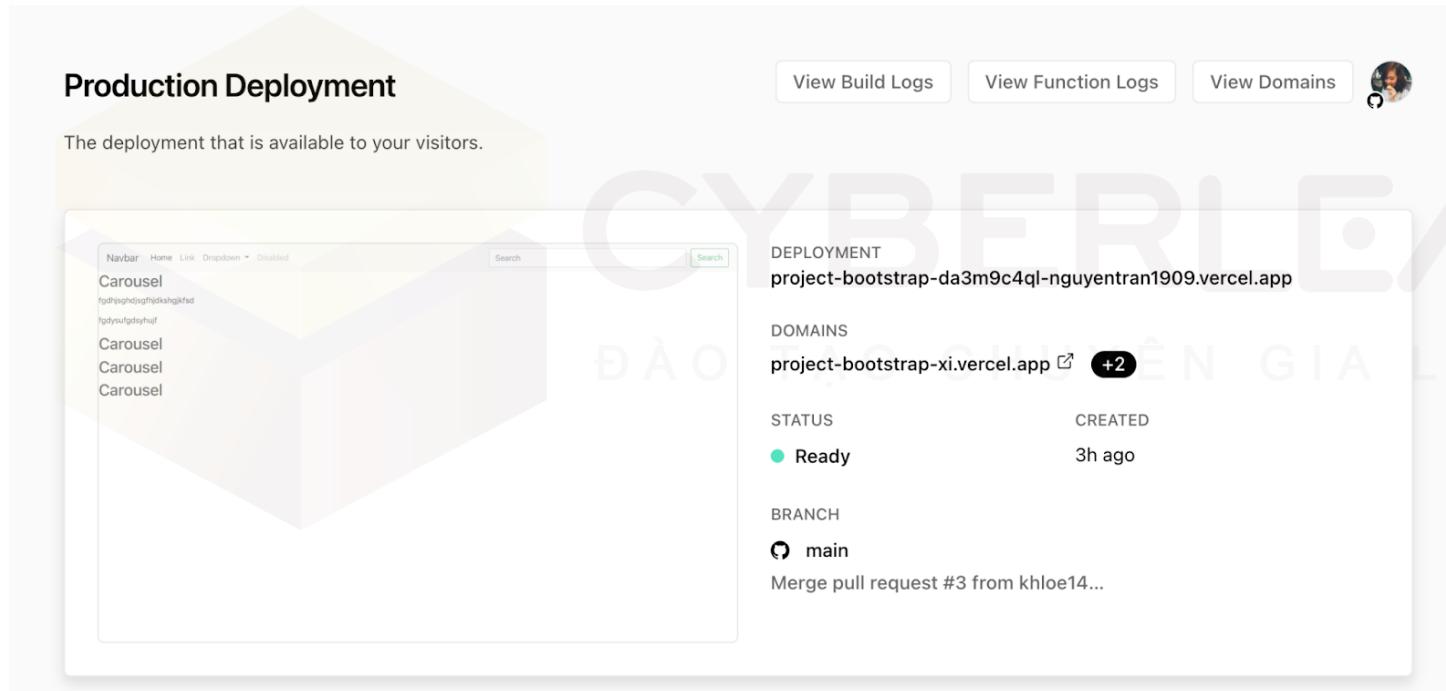
4. Thêm tài khoản github và import dự án muốn deploy
5. Kế đó nhấn button Deploy như hình bên dưới

The screenshot illustrates the Vercel deployment process:

- Import Git Repository:** On the left, a sidebar shows the user's GitHub account "khloe1425" selected. A list of repositories is shown, with the first repository, "MarketingWebsiteSne...", having its "Import" button highlighted with an orange box.
- ProjectBootstrap Configuration:** In the center, the project "ProjectBootstrap" is selected. The "Configure Project" section includes fields for "PROJECT NAME" (set to "project-bootstrap"), "FRAMEWORK PRESET" (set to "Other"), and "ROOT DIRECTORY" (set to "./"). The "Deploy" button at the bottom of this section is also highlighted with an orange box.

Deploy bằng vercel

- Sau khi deploy thành công sẽ được cung cấp các domain có sẵn như hình bên dưới





THANK YOU
CYBERLEARN
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

<https://cyberlearn.vn/>