CSE702040, Git/GitHub

Prepared by: Trangmx

Version Control

Quản lý phiên bản là việc quản lý các thay đổi của các dự án (mã nguồn) theo thời gian. Đây là yếu tố cần thiết trong các dự án làm việc nhóm. Khi có các thay đổi được thực hiện trên một dự án, các thay đổi này cần được theo dõi. Tập hợp của dự án và các thay đổi của nó theo thời gian được gọi là một kho lưu trữ (repo).

Repository (Repo)

Một kho lưu trữ bao gồm bản ghi theo thứ tự thời gian của tất cả các thay đổi đã thực hiện trên một dự án. Các kho lưu trữ cần được lưu trữ trên một máy chủ để chia sẻ giữa các thành viên trong nhóm. Các dịch vụ như GitHub được sử dụng để lưu trữ các kho lưu trữ. Ngoài ra, một tổ chức có thể thiết lập máy chủ riêng để quản lý các kho lưu trữ của mình.

Git

Git là một công cụ miễn phí và mã nguồn mở (tập hợp các lệnh) được sử dụng để quản lý phiên bản. Đây là hệ thống quản lý phiên bản phổ biến nhất trong ngành công nghiệp. Git cho phép quản lý dễ dàng các dự án cá nhân và nhóm. Nó cho phép nhiều người cùng làm việc trên một dự án mà không lo lắng về việc các thay đổi của họ bị ghi đè (bị mất).

Thực hành

Yêu cầu

Tạo tài khoản Github, và cài đặt Tocken truy cập cá nhân cho tài khoản.

Trả lời câu hỏi vào file `git_answers.docx`. Mỗi câu hỏi là 5 điểm.

1. Tạo thư mục project. Mở terminal và chạy các lệnh:

```
mkdir git-lab
cd git-lab
```

2. Tìm phiên bản của Git được cài đặt trên hệ thống.

```
git --version
```

Ghi lại câu trả lời vào file trả lời là `Answer 1`.

3. Cài đặt định danh của bạn (tên và email). Chạy các lệnh:

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

Xác nhận lại thay đổi bằng lệnh:

```
git config --list
```

Ghi lại câu trả lời vào file trả lời là **`Answer 2`**.

4. Git bao gồm hệ thông trợ giúp rất tốt, bạn có thể nhờ trợ giúp bằng lệnh sau:

```
git <command> --help
```

Thay `<command>` bằng bất kỳ lệnh git nào. Thử với lệnh sau:

```
git add --help
```

Chuyện gì xảy ra khi bạn chạy: `git --help`? Ghi câu trả lời ra **`Answer 3`**.

Tạo và bảo trì kho lưu trữ

---

6. Di chuyển vào thư mục `git-lab` đã tạo.

```
cd git-lab
```

Chúng ta sử dụng thư mục `git-lab` là kho lưu trữ nội bộ. Chạy lệnh sau:

```
ls -a       (Mac/Linux/WSL)
```

`-a` option allows you to see hidden files and directories (files and directories names that start with a period). In addition to the two files you created in step 2, you should see two hidden entries **`.`** and **`..`**. These are relative names for the current directory and the parent directory.

Tạo Repo cho thư mục `git-lab` bằng lệnh sau:

```
git init
```

Lệnh này sẽ tạo một thư mục con ẩn có tên là `.git`. Đây là thư mục nơi tất cả các thay đổi sẽ được theo dõi cho kho lưu trữ này. Không nên chỉnh sửa hoặc xóa thư mục này, nếu không bạn sẽ mất toàn bộ thông tin theo dõi của mình.

Workflow

Git có 03 khu vực: `Thư mục làm việc`, vùng đệm `Staging`, và `Kho lưu trữ để commit`.

- **Thư mục làm việc (cây)**

  Bất kỳ thay đổi nào bạn thực hiện đối với tệp của mình (bao gồm việc thêm tệp mới) sẽ nằm trong khu vực này. Các tệp mới tạo sẽ được coi là `untracked (chưa được theo dõi)`.

- **Staging area (index)**

  Các tệp sẽ được `tracked (theo dõi)` và các thay đổi sẽ sẵn sàng để được commit.

- **Repository Commit area (History)**

  Once files are committed, they become part of the repository history. They are tracked by the git system and a snapshot of your repository is created. Each snapshot is identified by a *hash* string. Khi các tệp được commit, chúng trở thành một phần của lịch sử kho lưu trữ. Chúng được hệ thống git theo dõi và bản sao chụp nhanh của kho lưu trữ của bạn được tạo ra. Mỗi bản sao nhanh được xác định bởi một chuỗi *hash*.

Tạo tệp `README.md`, đưa các thông tin sau vào tệp đó:

```
**full name**
Họ tên của bạ

**GitHub user name**
Tài khoả Github của bạn
```

Một lệnh thông thường được dùng để kiểm tra trạng thái của Repo là `git status`.

Chạy lệnh `git status` trong thư mục `git-lab` và ghi lại câu trả lời vào **`Answer 4`**.

Bây giờ bạn có một tệp chưa được theo dõi trong khu vực làm việc của mình. Hãy theo dõi tệp bằng cách đưa nó vào khu vực staging. Chạy lệnh:

```
git add README.md
```

Kiểm tra trạng thái của dự án của bạn và ghi lại kết quả của lệnh như là câu trả lời của bạn vào **`Answer 5`**.

Bạn có nhận thấy màu sắc khác nhau của tên tệp trong hai lệnh trạng thái không? Bây giờ tạo file thứ hai đặt tên là `answers.md`, nội dung là câu trả lời của bạn từ câu 1 đến câu 4, thêm file `answers.md`

vào khu vực staging.

Kiểm tra trạng thái và ghi lại kết quả vào câu trả lời **`Answer 6`**.

Các tệp của bạn hiện đã sẵn sàng để được commit. Chạy lệnh sau:

```
git commit -m "Initial commit"
```

Thông điệp trong lệnh commit rất quan trọng. Nó mô tả những gì đã xảy ra kể từ lần commit trước. Điều này đặc biệt hữu ích khi làm việc với các đồng đội trên cùng một dự án.

Kiểm tra trạng lại và đưa kết quả vào câu trả lời **`Answer 7`**.

Quy trình làm việc trên phác thảo các bước mà bạn thường thực hiện để tạo một commit (bản sao nhanh) của kho lưu trữ của mình tại bất kỳ thời điểm nào.

7. Xem lịch sử của kho lưu trữ (lịch sửa các commits). chạy lệnh `log`.

   ```
   git log
   ```

   Kết quả đầu ra sẽ bao gồm thông tin về các commits của bạn, bao gồm:

   - Hash value (`0c6112a...`).
   - `HEAD->master`: a pointer to a repository called `master`. `master` is your main repository `branch` name. More on branching later.
   - Author, email, date, and comment made while committing.

   Lưu lại kết quả chạy lệnh `git log` vào câu trả lời **`Answer 8`**.

Làm việc với kho lưu trữ từ xa trên GitHub

---

8. Mở trình duyệt và vào trang (`https://github.com`), đăng nhập vào Github với tài khoản của bạn.

   - Tạo kho lưu trữ *private* repository (click the + symbol on the top right).
   - Đặt tên kho lưu trữ của bạn là `git-lab`
   - Không lựa chọn gì khác nữa.
   - Click vào nốt `Create repository`.
   - Bạn đã tạo ra kho lưu trữ trông tên là `git-lab`. Bạn sẽ nhìn thấy hướng dẫn về `push an existing repository from the command line`. Có 3 lệnh cần chạy để đưa kho local của bạn lên Github.
   - Hãy chạy các lệnh sau (thay thế <user-name> với tài khoản Github của bạn bở (<>)):

     ```
     ```

```
git remote add origin https://github.com/<user-name>/git-lab.git
git branch -M main
git push -u origin main
```

or

```
git remote add origin git@github.com:<user-name>/git-lab.git
git branch -M main
git push -u origin main
```

> These commands will also give your branch the name `main`

Xem kết quả, chắc chắn ràng không có lỗi gì xảy ra. Kho lưu trữ của bạn đã được đưa lên Github. Làm mời trình duyệt, bạn sẽ thấy hai tệp `README.md` and `answers.md` trên Github.

Kiểm tra trạng thái của kho lưu trữ của bạn. Ghi kết quả ra câu trả lời **`Answer 9`**.

9. Cập nhật file `README.md` ở local. Thêm thêm tin về email của bạn vào file này. Trên terminal, thêm `README.md`, `commit` thay đổi, và `push` thay đổi lên GitHub.

```
git add README.md
git commit -m "Enter a message here"
git push
```

> Có thể cần phải điền Token của bạn ở đây.

CÒN TIẾP
