**Bài tập thực hành 05**

Làm quen với Github

**Mục tiêu cần đạt**:

Sau buổi các bạn cần nắm thao tác sử dụng github

Các bạn đọc và thực hiện viết code như hướng dẫn để làm quen, sau đó đọc trả lời các **Yêu cầu** ở dưới bài hướng dẫn để thực hiện yêu cầu bài thực hành.

**Gợi ý Tham khảo:**

* <https://docs.github.com/en/get-started/quickstart/hello-world>
* <https://www.youtube.com/watch?v=8Dd7KRpKeaE>
* [Git, GitHub, & GitHub Desktop for beginners (youtube.com)](https://www.youtube.com/playlist?list=PLJcWUrckOCKJ9cL3WtHhK_eHu5Q3aYnDs)
* https://topdev.vn/blog/github-la-gi/
* và các nguồn khác mà sinh viên tìm được
* Các lệnh trong git https://git-scm.com/docs

## Github

Là nơi các bạn có thể dùng làm nơi lưu trữ code, csdl để làm việc theo nhóm hiệu quả. Để thực hiện tạo các kho repository để lưu trữ dữ liệu bạn sẽ cần tạo 1 tài khoản github tại <http://github.com>.

Bạn hãy đọc yêu cầu bài thực hành và theo đó bạn sẽ tìm hiểu kỹ về việc dùng github. Nếu chưa có git bạn download tại đây: <https://git-scm.com/download/win> và cài đặt.

Sau đó vào cmd gõ “git để kiểm tra việc cài đặt, nếu hiện ra được thông báo này là đã cài đặt thành công:

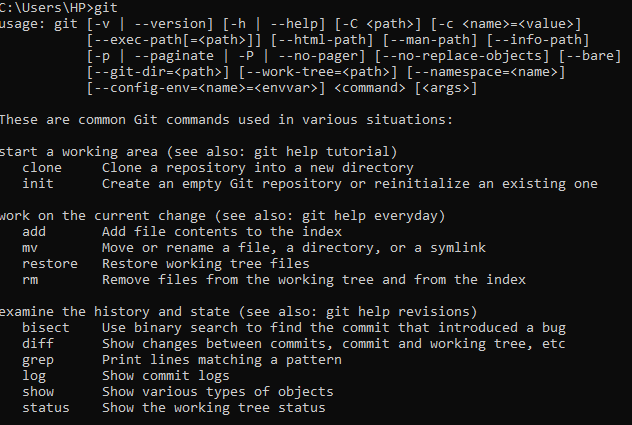


# Yêu cầu bài thực hành:

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.**

**Trả lời:**

Cài đặt git thành công



1. **Bạn hãy tìm hiểu và liệt kê những lợi ích của git và github.**

**Trả lời:**

Git và GitHub là hai công cụ quan trọng trong quá trình phát triển phần mềm, đặc biệt là khi làm việc trong các dự án phần mềm độ lớn hoặc có nhiều người cùng làm việc. Dưới đây là một số lợi ích của Git và GitHub:

- **Lợi ích của Git:**

1. Hệ thống quản lý phiên bản phân tán (DVCS):

- Mỗi người làm việc trên dự án có bản sao đầy đủ của toàn bộ repository. Điều này giúp tránh tình trạng phụ thuộc vào một máy chủ trung tâm.

- Không cần kết nối internet liên tục để làm việc.

2. Quản lý lịch sử thay đổi (Version Control):

- Lưu trữ toàn bộ lịch sử thay đổi của mã nguồn.

- Dễ dàng quay lại phiên bản trước đó, so sánh sự thay đổi, và xem ai đã thực hiện thay đổi đó.

3. Branching và Merging:

- Tạo các nhánh (branches) để phát triển tính năng mà không ảnh hưởng đến mã nguồn chính.

- Merge các nhánh lại với nhau một cách dễ dàng và an toàn.

4. Collaboration:

- Cho phép nhiều người cùng làm việc trên cùng một repository mà không gây xung đột.

- Đồng bộ hóa thay đổi giữa các thành viên đội.

5. Backup và Restore:

- Dễ dàng sao lưu toàn bộ dự án.

- Khôi phục lại trạng thái trước đó nếu có sự cố xảy ra.

- **Lợi ích của GitHub** (là một dịch vụ hosting Git):

1. Remote Repository:

- Lưu trữ mã nguồn trên server, cho phép truy cập từ bất kỳ đâu, bất kỳ thiết bị nào.

- Dễ dàng chia sẻ repository với người khác.

2. Collaboration và Code Review:

- Tạo pull request để thảo luận, kiểm tra và chấp nhận sự thay đổi từ nhiều người cùng một lúc.

- Dễ dàng quản lý công việc giữa các thành viên trong dự án.

3. Issue Tracking:

- Tạo và quản lý các vấn đề, công việc cần làm.

- Liên kết vấn đề với các commit và pull request.

4. Continuous Integration (CI) và Continuous Deployment (CD):

- Kết hợp với các dịch vụ như Travis CI, GitHub Actions để tự động kiểm tra, xây dựng và triển khai ứng dụng.

- Giúp giảm thời gian và công sức trong việc kiểm tra và triển khai.

5. Documentation và Wiki:

- Tạo wiki để lưu trữ tài liệu, hướng dẫn sử dụng.

- Thông tin dự án được tổ chức và dễ dàng tìm kiếm.

6. Community và Open Source:

- GitHub thu hút một cộng đồng lớn của các nhà phát triển, giúp dễ dàng tìm kiếm mã nguồn mở và đóng góp vào các dự án khác nhau.

-Tóm lại, Git và GitHub cung cấp một hệ sinh thái mạnh mẽ cho việc quản lý mã nguồn, hợp tác trong dự án và xây dựng ứng dụng một cách hiệu quả và linh hoạt.

1. **Repositories trong github là gì (Tham khảo [[1]](#footnote-1))? 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-2), 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**

**Trả lời:**

* **Repositories** trong github **là** thành phần cơ bản nhất của GitHub. Đó là nơi bạn có thể lưu trữ mã, tệp của mình và lịch sử sửa đổi của từng tệp. Các kho lưu trữ có thể có nhiều cộng tác viên và có thể ở chế độ công khai hoặc riêng tư.
* Tính đến thời điểm hiện tại, GitHub cung cấp không giới hạn số lượng repository bạn có thể tạo miễn phí, nhưng có giới hạn về dung lượng cho mỗi repository và tổng dung lượng của tất cả các repository. Dưới đây là giới hạn dung lượng cho các repository GitHub miễn phí:

1.Dung lượng cho mỗi repository:

. Mỗi repository Github miễn phí có giới hạn dung lượng là 1GB.

2.Tổng dung lượng của tất cả các repository:

. Tổng dung lượng của tất cả các repository trên tài khoảng Github miễn phí có giới hạn là 100GB.

Nếu repository của bạn vượt quá giới hạn 1GB, bạn sẽ nhận được thông báo và không thể thêm nhiều dữ liệu mới vào repository đó cho đến khi giải phóng không gian.

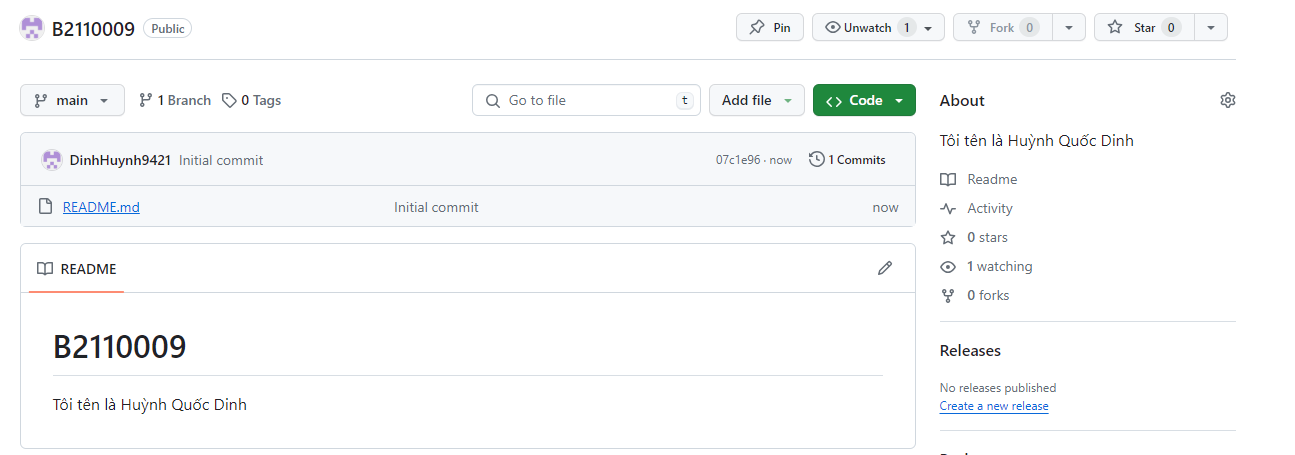
* Kích thước tập tin lớn (LFS): GitHub sử dụng Git Large File Storage (LFS) để quản lý các tập tin lớn. Kích thước tập tin mà bạn có thể tải lên LFS phụ thuộc vào loại tài khoản bạn đang sử dụng:

+ Miễn phí: 2 GB mỗi tháng cho các tập tin LFS

+ Pro: 50 GB mỗi tháng cho các tập tin LFS

+ Team và Enterprise: Không có giới hạn về kích thước tập tin LFS

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



Link: <https://github.com/DinhHuynh9421/B2110009.git>

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

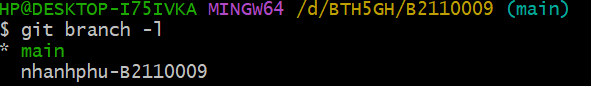
**Trả lời:**

Trong Git, một **branch** (nhánh) là một dòng phát triển độc lập của công việc trên repository. Khi bạn bắt đầu một dự án hoặc muốn thêm một tính năng mới vào dự án đã tồn tại, bạn có thể tạo ra một branch mới để phát triển tính năng đó mà không ảnh hưởng đến mã nguồn chính (thường được gọi là branch gốc hoặc default branch, thường là `main` hoặc `master`).

* **Tạo 1 branch tên “nhanhphu-<mssv của bạn>**



* Nhánh đã được tạo



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

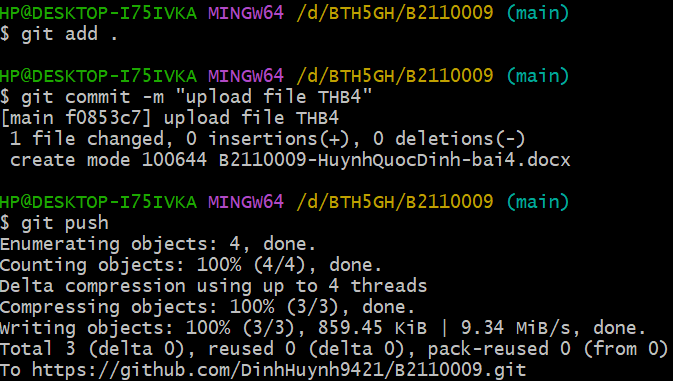
**Trả lời:**

Trong Git, các hành động commit, push, pull là những hành động quan trọng khi làm việc với repository.

* **Commit** là hành động lưu trữ các thay đổi của mã nguồn vào repository cùa bạn. Khi bạn thực hiện commit, bạn đang tạo ra một snapshot của trạng thái hiện tại của repository, bao gồm tất cả các tập tin và thư mục đã thay đổi.
* **Push** là hành động đẩy các commit từ repository local của bạn lên repository trên server (Ví dụ như Github).
* **Pull** là hành động lấy các thay đổi từ repository trên server về repository local của bạn.

1. **Hãy thực hiện hành động upload tập tin thực hành buổi 4 lần trước lên github và commit.**

**Trả lời:**



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

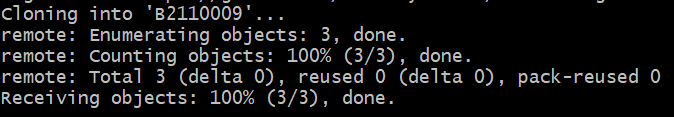
**Trả lời:**

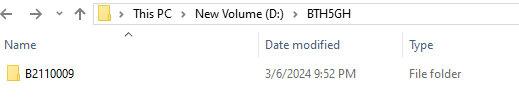


**-Thực hiện lệnh clone trên git:**



**-Kết quả:**





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.**

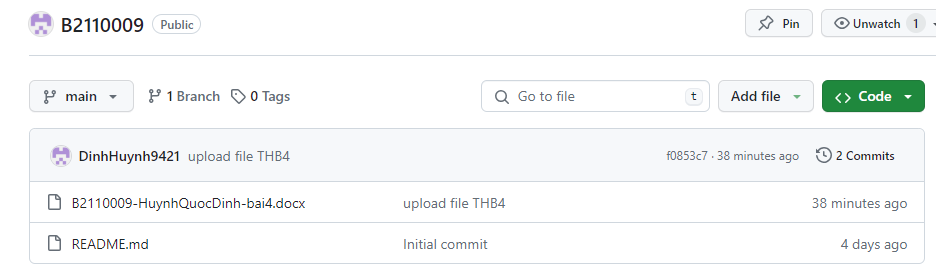
**Trả lời:**

-**git add**: Lệnh được sử dụng để di chuyển các thay đổi từ thư mục làm việc của bạn sang khu vực tổ chức (còn được gọi là chỉ mục). Khu vực tổ chức giống như khu vực lưu giữ nơi bạn có thể chuẩn bị các thay đổi của mình trước khi thực sự đưa chúng vào kho Git. Khi bạn thực hiện thay đổi đối với các tệp trong thư mục làm việc của mình, ban đầu Git không theo dõi những thay đổi này. Bạn cần thêm chúng vào khu vực tổ chức trước khi chúng có thể được commit.

-**git commit –m:** Lệnh git commit được sử dụng để lưu các thay đổi trong khu vực tổ chức vào kho Git. Mỗi commit thể hiện ảnh chụp nhanh kho lưu trữ của bạn tại thời điểm đó. Khi bạn thực hiện các thay đổi, bạn nên cung cấp thông báo cam kết giải thích những thay đổi đã được thực hiện trong lần cam kết đó. Lịch sử cam kết này giúp hiểu được sự phát triển của dự án và cho phép bạn quay trở lại trạng thái trước đó nếu cần.

**-git push:** Lệnh git push được sử dụng để gửi các thay đổi đã commit của bạn tới kho lưu trữ từ xa. Kho lưu trữ từ xa là kho lưu trữ được lưu trữ trên máy chủ, thường là trên các dịch vụ như GitHub, GitLab hoặc Bitbucket. Khi bạn đẩy các thay đổi, về cơ bản bạn đang tải các cam kết của mình lên kho lưu trữ từ xa, cung cấp chúng cho những người khác làm việc trong cùng một dự án.

**-Quan sát kết quả trên trang github của bạn đã tạo ở câu 3 và nhận xét**



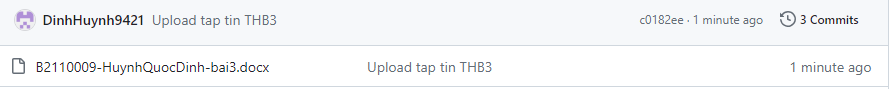
-Tập tin bài thực hành buổi 4 lúc này đã được upload lên trang đã tạo với nội dung commit là upload file THB4, và thời gian upload lên trang

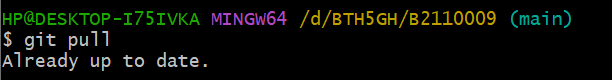
1. **Upload tập tin thực hành buổi 3 lên trang github của bạn và thực hiện lệnh**

**git pull**

**Quan sát kết quả đạt được và nhận xét.**

**Trả lời:**

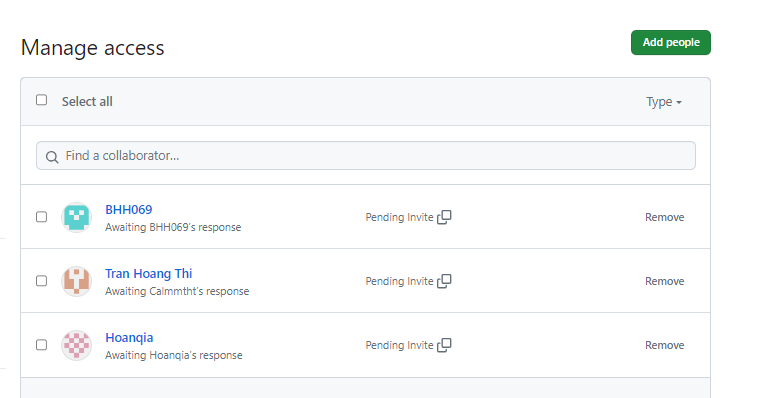




1. **Bạn hãy add các contributor là những bạn sẽ làm cùng nhóm với bạn**

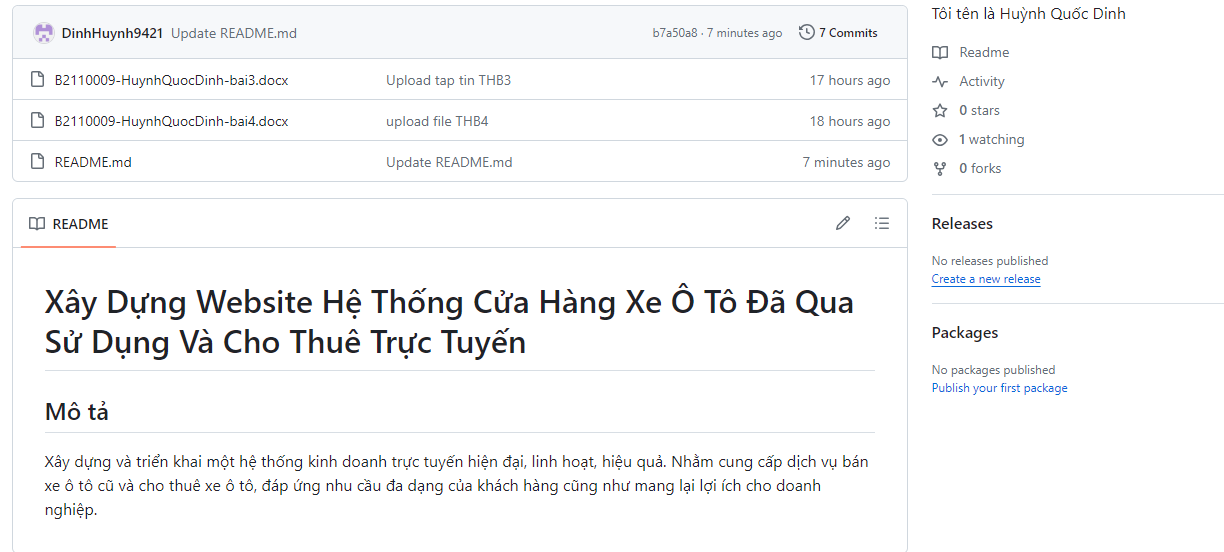


**Trả lời:**



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.**

**Trả lời:**



**Chú ý:**

* Các bạn nộp file word: Quy tắc đặt tên file: **<mssv>-<hoten>-<bai><stt\_bai thực hành>.docx** nộp lên Classroom (VD: **B123456-NguyenVanA-bai1.docx**), kèm với các file khác được yêu cầu như phần câu hỏi đã nêu. **Ngoại trừ file word trả lời câu hỏi, các file còn lại các bạn nén vào 1 file zip**. File zip đặt tên như file word.
* Mỗi câu các bạn trả lời bằng hình hoặc dạng text tùy vào yêu cầu của câu hỏi và **TRẢ LỜI THEO ĐÚNG THỨ TỰ CÂU HỎI**. Nếu câu nào không trả lời được các bạn cứ để số thứ tự câu hỏi và bỏ trống phần trả lời.
* Các câu trả lời có tham khảo trên Internet phải trích dẫn link/nguồn.
* *Vi phạm 1 trong các điều sau đây bài thực hành sẽ bị 0 điểm:*
  + Đặt tên KHÔNG ĐÚNG quy tắc được yêu cầu.
  + Bài không đủ các thành phần (word, code+data (nếu có),...) đã được yêu cầu.
  + Bài không thực hiện đúng yêu cầu “**Ngoại trừ file word trả lời câu hỏi, các file còn lại các bạn nén vào 1 file .zip”**
  + Bị phát hiện copy, sao chép từ các bạn khác
  + Phần trả lời không ghi rõ trả lời cho câu nào
  + Thứ tự câu trả lời không đúng thứ tự câu hỏi

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