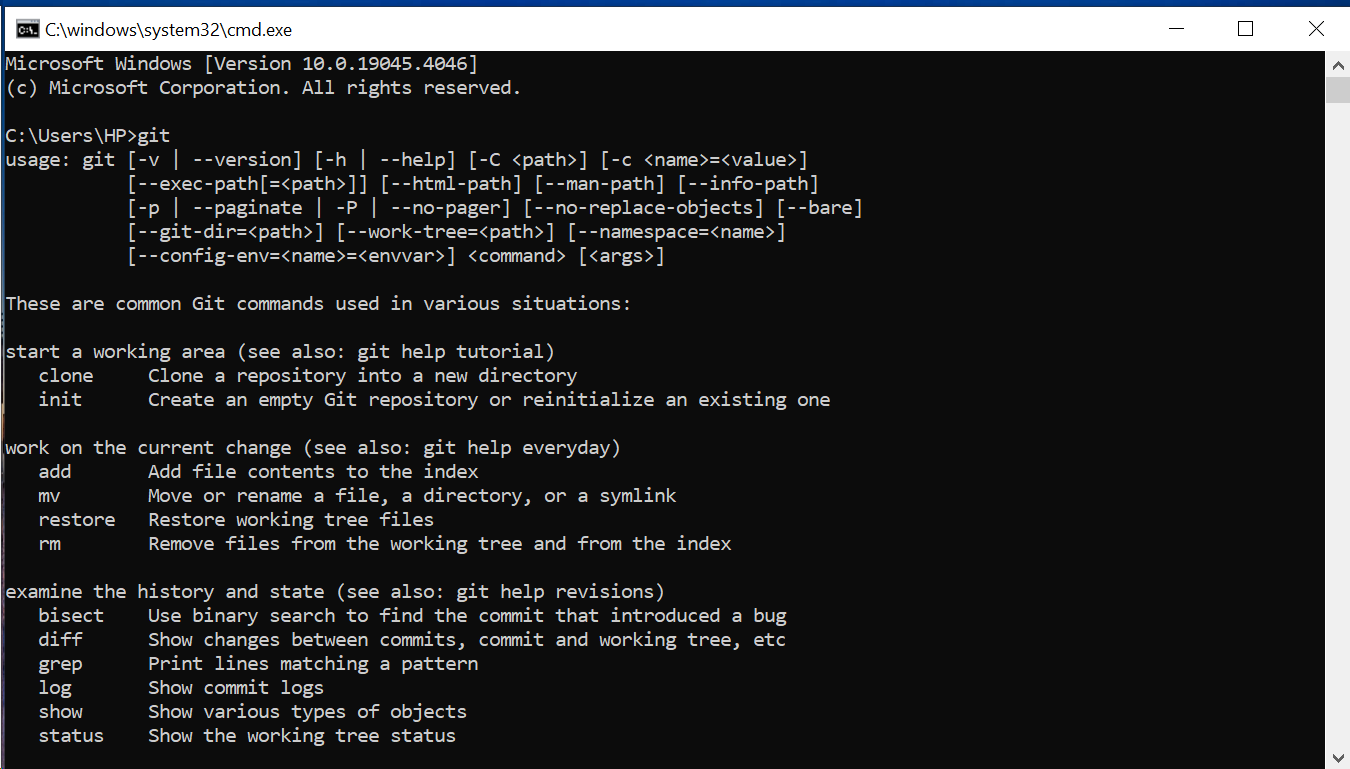
**Bài thực hành 05:**

**LÀM QUEN VỚI GITHUB**

**Câu 1:**

- Cài đặt github trên máy:



**Câu 2:**

**\*GIT:**

- Git là một hệ thống kiểm soát phiên bản phân tán code nguồn mở. Nó được thiết kế để xử lý các dự án từ nhỏ đến lớn với tốc độ và hiệu quả cao. Nó được phát triển để điều phối công việc giữa các developer. Kiểm soát phiên bản cho phép chúng ta theo dõi và làm việc cùng với các thành viên trong nhóm của chúng ta tại cùng một không gian làm việc.

+ Là công cụ giúp quản lý source code tổ chức theo dạng dữ liệu phân tán.

+ Giúp đồng bộ source code của team lên 1 server.

+ Hỗ trợ các thao tác kiểm tra source code trong quá trình làm việc (diff, check modifications, show history, merge source, …).

- Các lợi ích của git:

+ Tiết kiệm thời gian(save time):

Git là công nghệ nhanh như chớp. Mỗi lệnh chỉ mất vài giây để thực thi nên chúng ta có thể tiết kiệm rất nhiều thời gian so với đăng nhập vào tài khoản GitHub và tìm hiểu các tính năng của nó.

+ Làm việc ngoại tuyến(offline working):

Một trong những lợi ích quan trọng nhất của Git là nó hỗ trợ làm việc ngoại tuyến . Nếu chúng ta đang gặp phải sự cố kết nối internet, nó sẽ không ảnh hưởng đến công việc của chúng ta. Trong Git, chúng ta có thể làm hầu hết mọi thứ cục bộ. So sánh, các CVS khác như SVN bị hạn chế và thích kết nối với kho lưu trữ trung tâm hơn.

+ Hoàn lại sai lầm(undo mistakes):

Một lợi ích bổ sung của Git là chúng ta có thể Hoàn tác lỗi. Đôi khi hoàn tác có thể là một lựa chọn cứu tinh cho chúng ta. Git cung cấp tùy chọn hoàn tác cho hầu hết mọi thứ.

+ Theo dõi các thay đổi(Track changes):

Git tạo điều kiện với một số tính năng thú vị như Diff, Log và Status , cho phép chúng ta theo dõi các thay đổi để chúng ta có thể kiểm tra trạng thái, so sánh các file hoặc chi nhánh của mình.

(TLTK: [Tự học Git | Giới thiệu chi tiết về Git và tại sao lại sử dụng nó? » Cafedev.vn](https://cafedev.vn/tu-hoc-git-gioi-thieu-chi-tiet-ve-git-va-tai-sao-lai-su-dung-no/" \l ":~:text=M%E1%BB%99t s%E1%BB%91 l%E1%BB%A3i %C3%ADch %C4%91%C3%A1ng k%E1%BB%83 c%E1%BB%A7a vi%E1%BB%87c,Theo d%C3%B5i c%C3%A1c thay %C4%91%E1%BB%95i (Track changes) ))

**\*GITHUB:**

- GitHub là một dịch vụ nổi tiếng cung cấp kho lưu trữ mã nguồn Git cho các dự án phần mềm. GitHub cung cấp đầy đủ các tính năng của Git và còn bổ sung thêm các tính năng về mặt xã hội để các lập trình viên, nhà phát triển (developer - Dev) tương tác với nhau.

- Các lợi ích của github:

+ Dễ dàng quản lý source code.

+ Dễ dàng theo dõi sự thay đổi qua các version.

+ Xây dựng thương hiệu cá nhân, chứng tỏ năng lực của bản thân.

+ Cải thiện kỹ năng code và theo dõi bug.

+ GitHub là một kho tài nguyên tuyệt vời, vô hạn.

+ Mở rộng mối quan hệ, gặp gỡ các nhà phát triển trên toàn thế giới, dễ dàng chia sẻ kinh nghiệm của bản thân.

(TLTK: [GitHub là gì? GitHub mang lại những lợi ích gì? - QuanTriMang.com](https://quantrimang.com/cong-nghe/github-la-gi-193520" \l ":~:text=V%E1%BB%9Bi c%C3%A1c l%E1%BA%ADp tr%C3%ACnh vi%C3%AAn v%C3%A0 nh%C3%A0 ph%C3%A1t,nguy%C3%AAn tuy%E1%BB%87t v%E1%BB%9Di, v%C3%B4 h%E1%BA%A1n C%C3%A1c m%E1%BB%A5c kh%C3%A1c))

**Câu 3:**

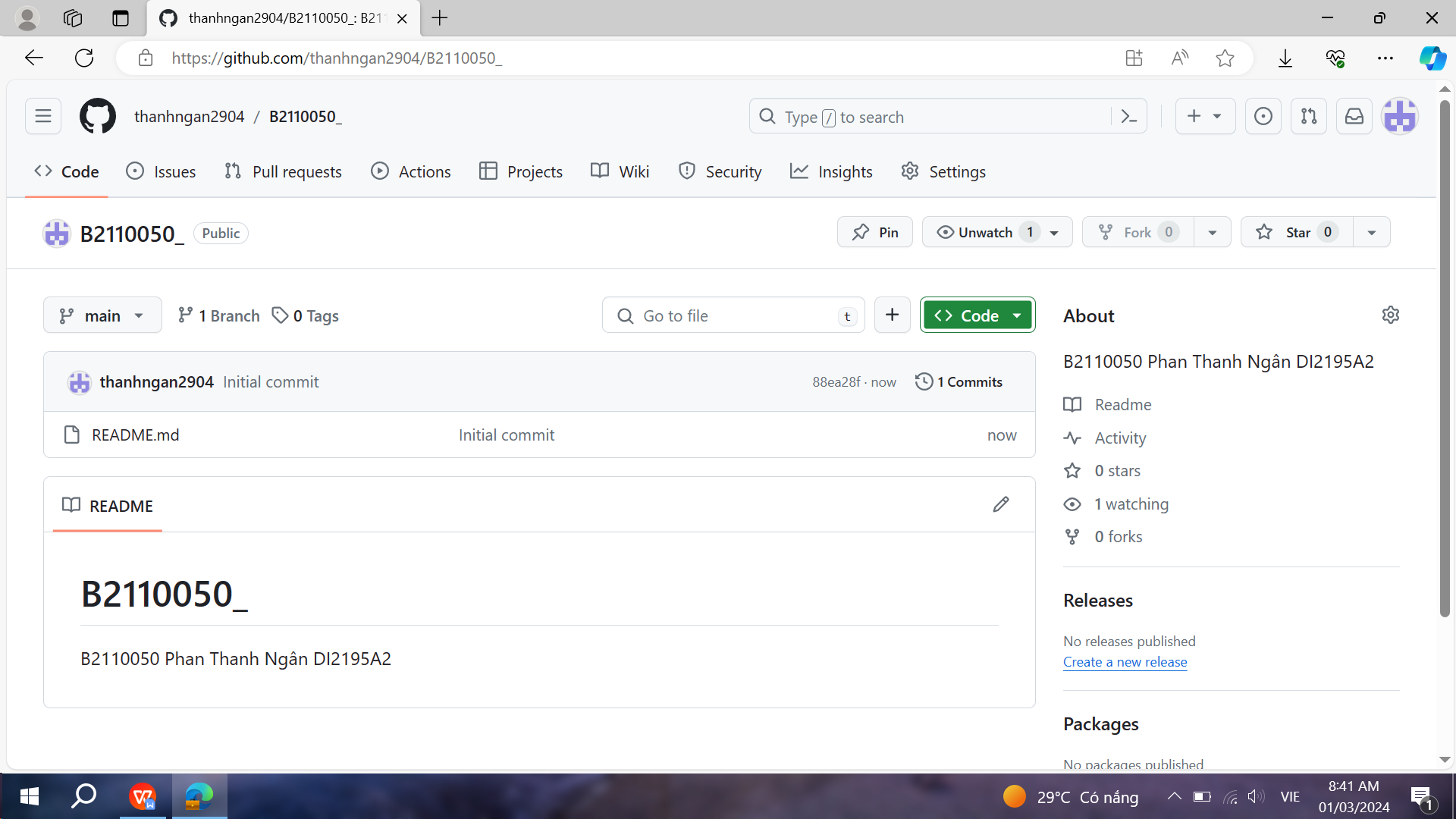
- Repository được hiểu là một kho lưu trữ nơi chứa các files của dự án. Các file đó có thể là code, hình ảnh, âm thanh hoặc mọi thứ liên quan đến dự án. Bạn có thể tổ chức kho lưu trữ của mình dưới nhiều hình thức khác nhau, hai loại kho lưu trữ trong Github là Local Repository và Remote Repository.

- Dung lượng tối đa cho mỗi repository là 100 GB. Nếu repository vượt quá giới hạn này, cần nâng cấp lên một tài khoản trả phí hoặc sử dụng các dịch vụ lưu trữ khác.

- Về mặt kỹ thuật, Git không có kích thước tệp tối đa, nhưng nó bắt đầu bị lỗi khi bắt đầu đạt đến kích thước tệp nhất định. Github đặt mức tối đa này là 100MB cho mỗi kho lưu trữ.

(TLKT: [Cách lưu trữ các tệp lớn trong Git - (tremplin-numerique.org)](https://www.tremplin-numerique.org/vi/c%C3%A1ch-l%C6%B0u-tr%E1%BB%AF-c%C3%A1c-t%E1%BB%87p-l%E1%BB%9Bn-trong-git" \l ":~:text=V%E1%BB%81 m%E1%BA%B7t k%E1%BB%B9 thu%E1%BA%ADt, Git kh%C3%B4ng c%C3%B3 k%C3%ADch,c%C3%A1ch Git l%C6%B0u tr%E1%BB%AF d%E1%BB%AF li%E1%BB%87u n%E1%BB%99i b%E1%BB%99.))

- Tạo 1 repository - giới thiệu thông tin bản thân:



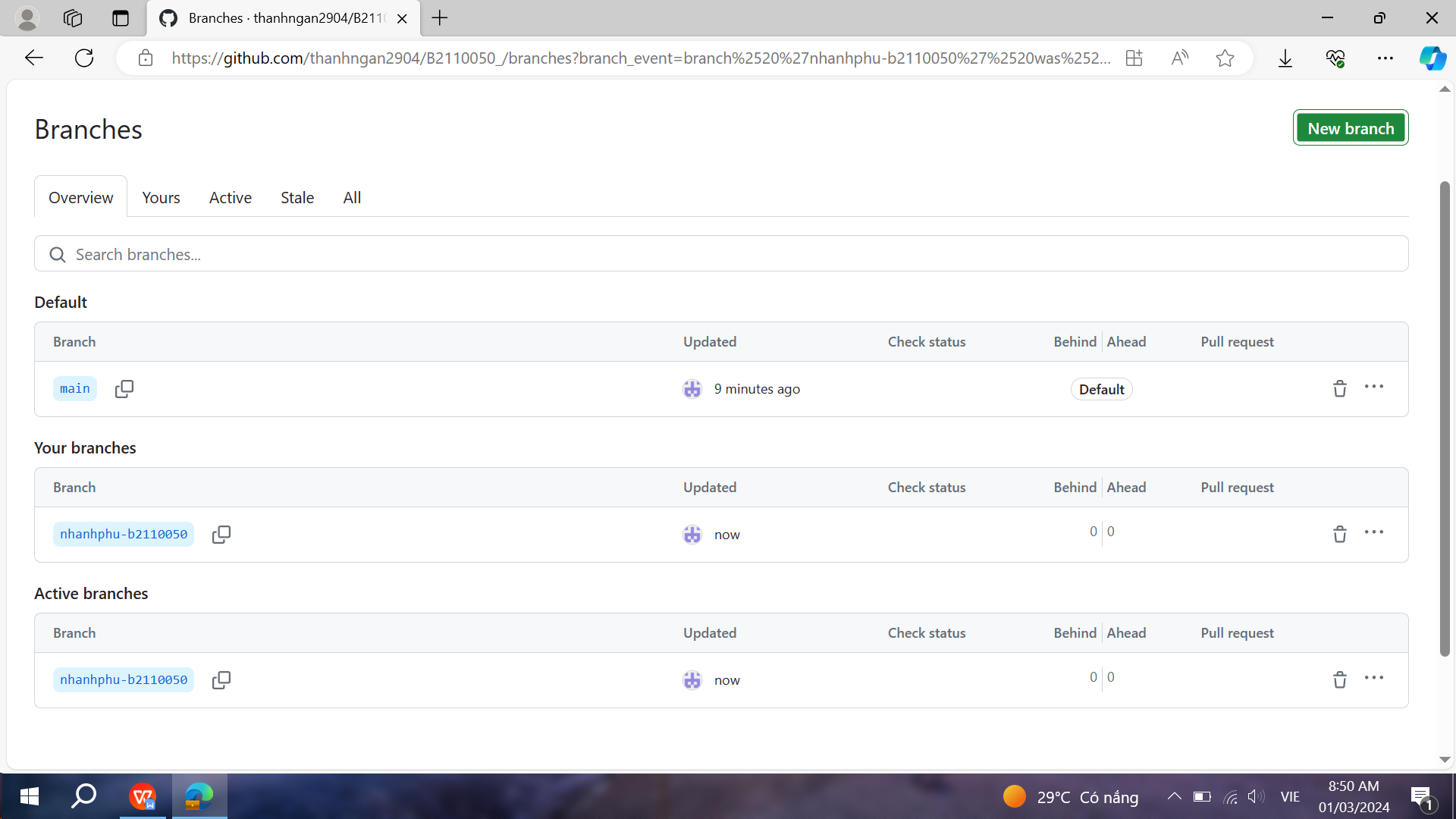
- Đường link vào bài tập: https://github.com/thanhngan2904/B2110050\_.git

**Câu 4:**

- Branch là nhánh của repository, hiểu đơn giản nó như một khu vực làm việc riêng (workspace). Khi tạo ra repository (kho chứa) ở git sẽ có 1 branch chính (master/main). Từ branch chính này có thể phân chia thành nhiều branch nhỏ. Branch đã phân nhánh này không ảnh hưởng đến branch chính vậy nên ta có thể tiến hành nhiều thay đổi trên cùng 1 repository. Đồng thời ta có thể merge (kết hợp) các branch với nhau.

(TLTK: [[Phần 2]Tổng quan về Git và những câu lệnh cơ bản (viblo.asia)](https://viblo.asia/p/phan-2tong-quan-ve-git-va-nhung-cau-lenh-co-ban-L4x5xqrmKBM))

- Tạo 1 branch mới:



**Câu 5:**

- Commit: lệnh này được hiểu như là việc ghi lại các ảnh chụp nhanh "snapshot" của kho lưu trữ tại một thời điểm nhất định. Ảnh chụp nhanh đã được commit sẽ không bao giờ thay đổi trừ khi chúng ta có sự thay đổi trong kho lưu trữ một cách rõ ràng.

- Push: Lệnh này chuyển commit từ kho lưu trữ cục bộ của bạn sang kho lưu trữ từ xa. Nó là đối nghịch của lệnh pull.

- Pull: lệnh git pull tìm nạp các thay đổi từ kho lưu trữ từ xa sang kho lưu trữ cục bộ. Nó hợp nhất các thay đổi ở nguồn trong kho lưu trữ cục bộ của bạn, đây là một nhiệm vụ phổ biến trong việc cộng tác dựa trên Git.

(TLTK: [Bài 4: Hướng dẫn sử dụng Git - Các lệnh cơ bản cần phải biết (aptechbmt.edu.vn)](https://aptechbmt.edu.vn/hoc-lap-trinh/git-github/bai-4-huong-dan-su-dung-git-cac-lenh-co-ban-can-phai-biet-6.html))

Câu 6:

Câu 7:

Câu 8:

Câu 9:

Câu 10: