**BÀI THU HOẠCH**

Khóa tập huấn

“Cài đặt và sử dụng hệ thống lưu trữ mã nguồn Github cho sinh viên khối ngành IT”

Ngày 22/3/2024 và 27/3/2024

**ĐỀ BÀI:**

**Câu 1:** Liệt kê các lợi ích của Git và Github?

**Lợi ích của Git:**

1. **Hệ thống quản lý phiên bản phân tán:** Git cho phép các nhà phát triển làm việc song song trên cùng một dự án mà không cần kết nối trực tiếp với máy chủ trung tâm. Mỗi cá nhân có phiên bản địa phương của toàn bộ lịch sử thay đổi, giúp tránh được sự cản trở khi làm việc offline và tăng tính linh hoạt.
2. **Tính nhẹ nhàng và tốc độ:** Git được thiết kế để nhanh chóng và hiệu quả với các dự án lớn. Các hoạt động như commit, branch và merge đều được thực hiện một cách nhanh chóng, giúp tăng năng suất cho nhà phát triển.
3. **Branching và merging linh hoạt:** Git cho phép bạn tạo ra các nhánh (branch) để phát triển tính năng mới mà không làm ảnh hưởng đến phiên bản chính của dự án. Sau đó, các thay đổi từ các nhánh có thể được hợp nhất (merge) lại với nhau một cách dễ dàng.
4. **Hệ thống ghi lại lịch sử linh hoạt:** Git ghi lại toàn bộ lịch sử thay đổi của dự án, cho phép bạn xem xét và quay lại bất kỳ phiên bản nào từ trước đó. Điều này giúp trong việc xác định nguyên nhân của các lỗi và quản lý các phiên bản phần mềm.

**Lợi ích của GitHub:**

1. **Lưu trữ mã nguồn trực tuyến:** GitHub cung cấp một nền tảng lưu trữ mã nguồn trực tuyến, cho phép bạn lưu trữ và quản lý mã nguồn của dự án trên Internet. Điều này giúp các nhà phát triển làm việc cùng nhau từ xa một cách dễ dàng.
2. **Quản lý dự án:** GitHub cung cấp các công cụ quản lý dự án như issue tracking, project boards và wiki, giúp tổ chức công việc, tương tác và phản hồi giữa các thành viên trong dự án.
3. **Hợp nhất mã (Pull Request):** GitHub cho phép bạn tạo ra pull request, là yêu cầu để hợp nhất các thay đổi từ một nhánh (branch) vào nhánh chính của dự án. Điều này tạo ra một quy trình kiểm tra và phê duyệt cho các thay đổi, giúp cải thiện chất lượng mã nguồn.
4. **Hệ sinh thái phong phú:** GitHub có một cộng đồng lớn và sôi động, nơi bạn có thể tìm thấy nhiều dự án mã nguồn mở, thư viện, công cụ và tài nguyên hữu ích khác.

**Câu 2:** Branch trong Git là gì?

Branch là một tham chiếu (reference) đến một trạng thái cụ thể của dự án. Mỗi branch thường đại diện cho một dòng phát triển riêng biệt của mã nguồn, cho phép các nhà phát triển làm việc độc lập trên các tính năng, sửa lỗi hoặc thay đổi mà không làm ảnh hưởng đến các branch khác.

**Câu 3:** Giải thích các câu lệnh Commit, Push, Pull trong Git và cho hình minh họa.

1. **Commit:**
   * **Giải thích:** Câu lệnh commit được sử dụng để lưu trữ một bản ghi (snapshot) của tất cả các thay đổi trong mã nguồn hiện tại của bạn vào kho lưu trữ Git. Mỗi commit đi kèm với một thông điệp mô tả các thay đổi đã thực hiện, giúp dễ dàng theo dõi lịch sử và xác định nội dung của mỗi bản ghi.
2. **Push:**
   * **Giải thích:** Sau khi bạn đã thực hiện commit các thay đổi vào kho lưu trữ cục bộ, câu lệnh push được sử dụng để đẩy các commit này lên kho lưu trữ từ xa. **origin** thường là tên của kho lưu trữ từ xa mặc định, và **branch\_name** là tên của nhánh bạn muốn đẩy các commit lên.
3. **Pull:**
   * **Giải thích:** Khi các thành viên trong nhóm làm việc trên cùng một dự án và có các thay đổi mới trên kho lưu trữ từ xa, câu lệnh pull được sử dụng để kéo các thay đổi này về kho lưu trữ cục bộ của bạn. Điều này giúp đồng bộ hóa phiên bản cục bộ với phiên bản từ xa và đảm bảo bạn đang làm việc với phiên bản mới nhất của dự án.

Top of Form

**Câu 4:** Tạo một dự án mới trên Github và đưa mã nguồn (bất kỳ) lên dự án này.

**Lưu ý:**

* *Tên dự án đặt theo cấu trúc là:* ***Tên-MSSV****.*
* *Mô tả lại ý nghĩa của các câu lệnh đã sử dụng.*
* *Cung cấp link của Repository vừa tạo.*

**Câu 5:** Dùng lệnh clone để tải về mã nguồn ở Câu 4. Sinh viên cần cung cấp hình minh họa cách lấy link và cách sử dụng câu lệnh clone.