Họ và Tên: Nguyễn Xuân Hưng

MSSV:1511060509

Lớp:15DTH03

Họ và Tên: Nguyễn truong thanh nhã

MSSV:1411060908

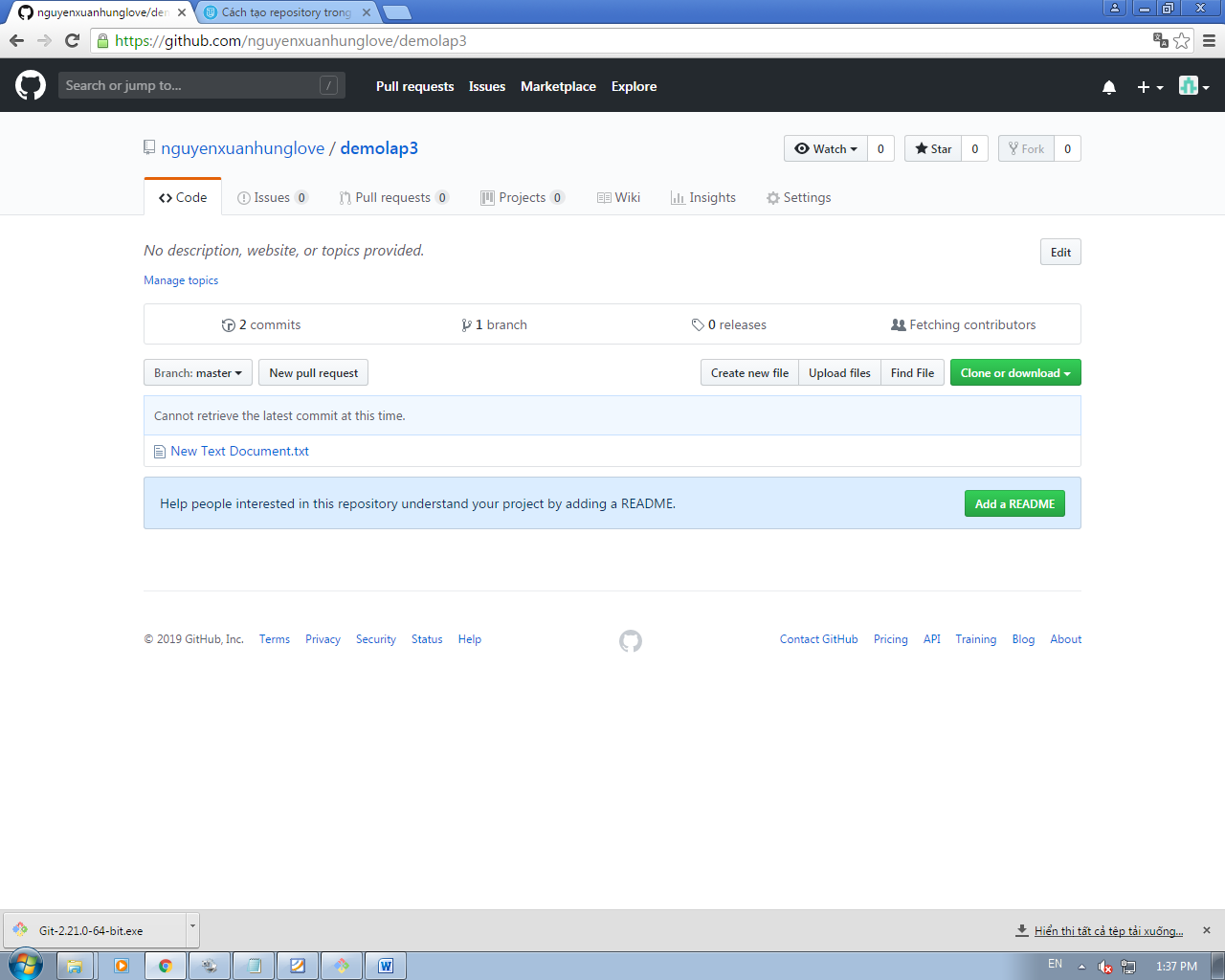
Lớp:15DTH04

TH Công Cụ Và Phát Triển môi trường

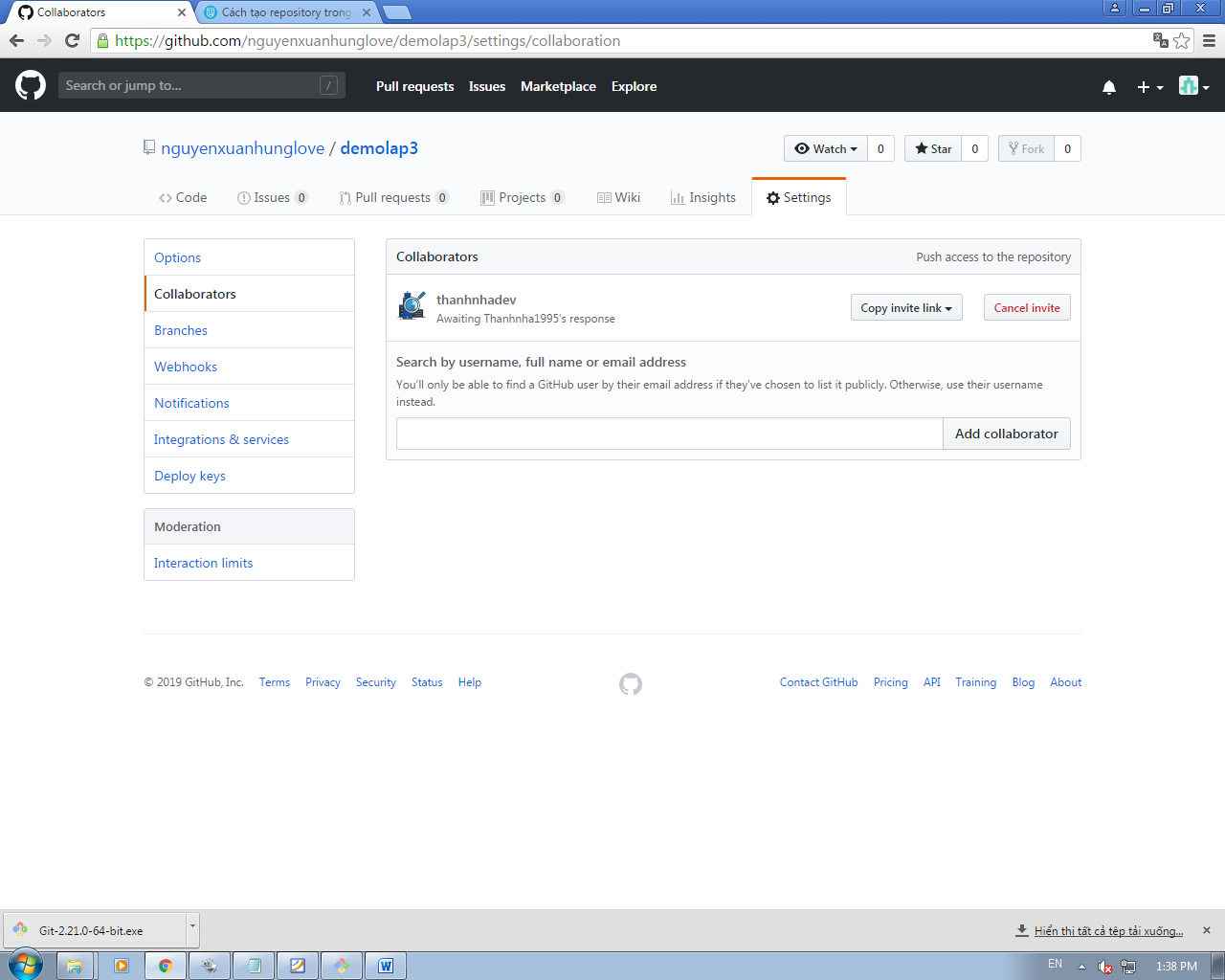
1. PHẦN 1 (2Đ)

- SỬ DỤNG GITLAB HOẶC GITHUB

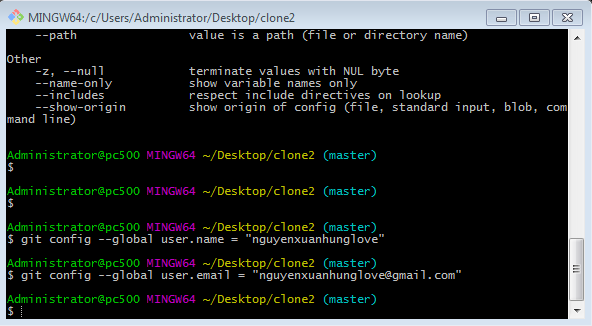
+ TẠO TÀI KHOẢN, TẠO REPO



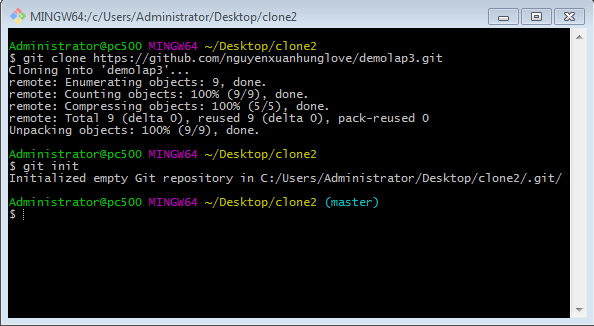
+ THÊM THÀNH VIÊN, THÊM QUYỀN



ĐỊNH DANH TÀI KHOẢN TẠI MÁY TÍNH ĐANG SỬ DUNG (GIT CONFIG)



+ INIT DỰ ÁN, CLONE DỰ ÁN

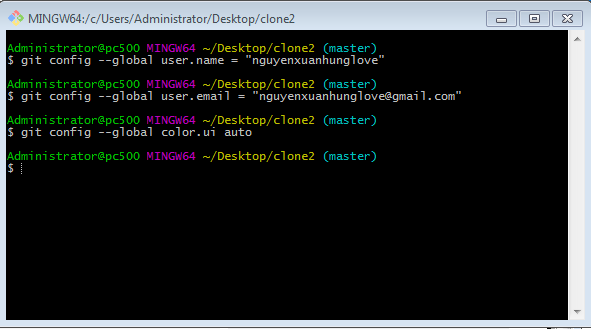


2. PHẦN 2 (5Đ)

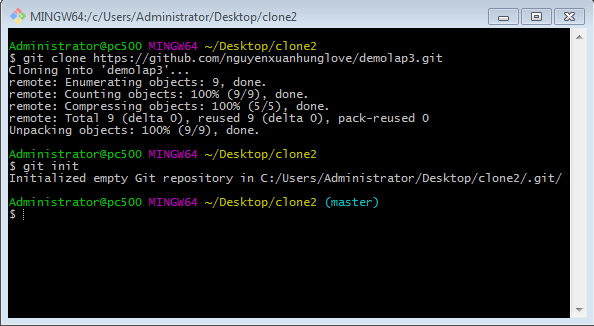
- THỰC HIỆN CÁC NỘI DUNG CÂU LỆNH CỦA GIT TRONG FILE GIT-CHEAT-SHEET, XEM THÊM HƯỚNG DẪN TRONG FILE LAB3

+ THỰC HIỆN ĐẦY ĐỦ HẾT CÁC CÂU LỆNH

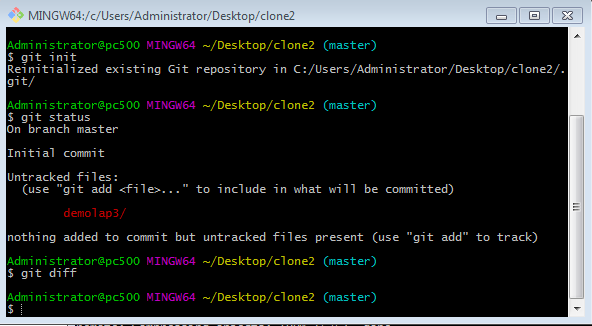
git config:



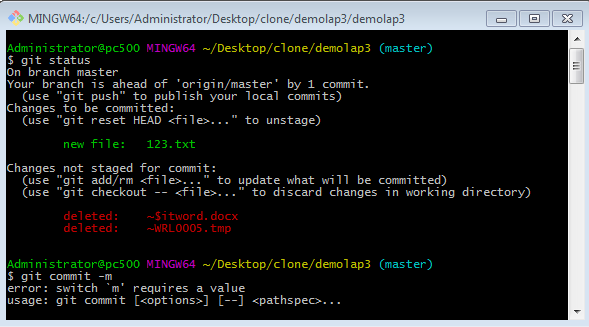
Git Init,git clone



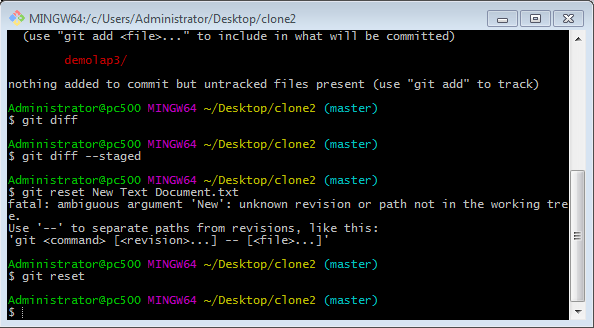
Git status,git diff



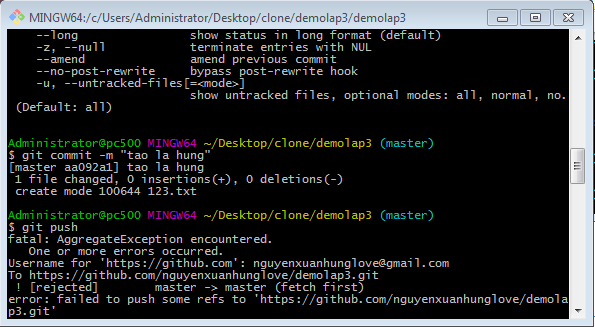
Git add



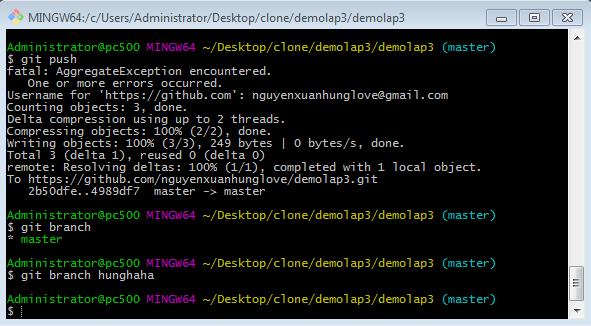
Git diff –staged,git reset.



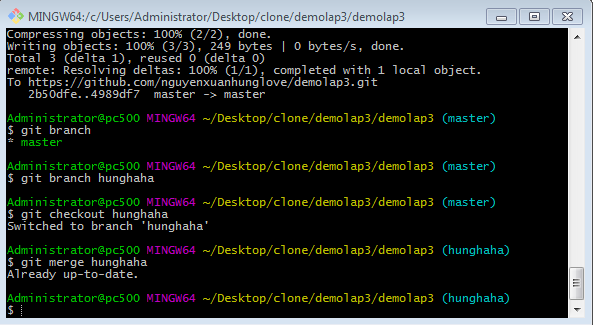
Git commit



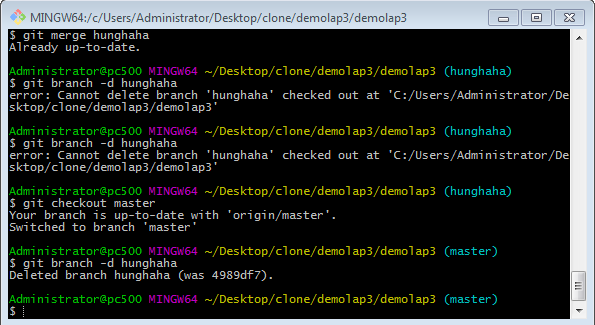
Git branch



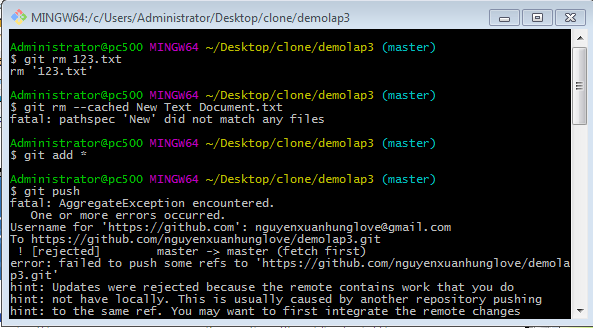
Git checkout,merge



git branch -d [branch-name]



Git rm



- TẠO XUNG ĐỘT MÃ NGUỒN VÀ CÀI ĐẶT CÔNG CỤ P4MERGE ĐỂ HỖ TRỢ GIẢI QUYẾT XUNG ĐỘT

PHẦN 4 (2Đ)

- DỊCH SANG TIẾNG VIỆT "CÓ Ý NGHĨA, MỚI ĐC ĐIỂM" CÁC CÂU LỆNH TRONG FILE GIT-CHEAT-SHEET VÀ LƯU VÀO FILE WORD ĐỂ LÀM MINH CHỨNG CHẤM ĐIỂM

* **git status:** Liệt kê tất cả dữ liệu mới hoặc sử đổi dữ liệu đã được định sẳn
* **$ git diff:** Hiển thị các thư mục chưa được định sẵn.
* **$ git add [file]:** Hiển dữ liệu đã chuẩn bị được thêm vào.
* **$ git diff –staged:** Hiển thị các thay đổi của thư mục trươc và sau phiên bản.
* **$ git reset [file]:** Loại bỏ tập tin và giữ nguyên nội dung của tập tin đó.
* **$ git commit -m "[descriptive message]":** Tin nhắn mô tả dữ liệu chuẩn bị xuất lên.
* **$ git config --global user.name "[name]":** Nhập tên người dung.
* **$ git config --global user.email "[email address]":** Nhập email người dung.
* **$ git config --global color.ui auto:** Tô màu dòng lênh quan trọng.
* **$ git branch:** Tạo nhánh
* **$ git branch [branch-name]:** Tạo nhánh mới.
* **$ git checkout [branch-name]:** Đổi nhánh và cập nhật dữ liệu đã làm
* **$ git merge [branch]:** Kết hợp lịch sử nhánh chi tiết cụ thể vào nhánh hiện tại.
* **$ git branch -d [branch-name]:** Xóa nhánh
* **$ git init [project-name]:** Tạo một kho dữ liệu mới và đặt tên.
* **$ git clone [url]:** Tại xuống toàn bộ dự án và hiển thị lịch sử tải xuống
* **$ git rm [file]:** xóa dữ liệu khỏi thư mục làm việc.
* **$ git rm --cached [file]:** Loại bỏ dữ liệu khỏi phiên bản nhưng vẫn giữ nguyện dữ liệu chung của nhóm.
* **$ git mv [file-original] [file-renamed]:** Thay đổi tên và chuẩn bị tải lên.
* **$ git log:** liệt kê lịch sử cho nhánh hiện tại
* **$ git log --follow [file]:** liệt kê lịch sử cho 1 file đã đổi tên.
* **$ git diff [first-branch]...[second-branch]:** Hiện thị nội dung 2 nhánh.
* **$ git show [commit]:** Xuất dữ liệu lớn.
* **$ git ls-files --other --ignored --exclude-standard:** Liệt kê tất cả file không liên quan trong dự án.
* **$ git reset [commit]:** Hủy bỏ tất cả câu lệnh mô tả
* **$ git reset --hard [commit]:** Hủy bỏ tất cả lịch sử và thay đổi cụ thể.
* **$ git stash:** Lưu vào bộ nhớ tạm
* **$ git stash pop:** Phục hồi file gần nhất.
* **$ git stash list:** Liệt kê tất cả các thay đổi
* **$ git stash drop:** Loại bỏ các thay đổi gần nhất.
* **$ git fetch [bookmark]:** Tải xuống tất cả lịch sử trong kho lưu trữ
* **$ git merge [bookmark]/[branch]:** Kết hợp nhánh đã đánh dấu và nhánh hiện tại
* **$ git push [alias] [branch]:** Tải tất cả nhánh lên Github
* **$ git pull:** Hiện thị tất cả các lịch sử đã thay đổi.