Assignment 2

Introduction

The goal of this assignment is to consolidate skills related to concepts learned in Module 3 and also utilize the concepts and practices covered in Module 1 and Module 2.

In this assignment, the student is expected to do hands-on in the following areas:

- Undoing Git operations such as:
    - Git Checkout
    - Git Revert and
    - Git Reset
- Fundamental Git & GitHub operations such as:
    - Push and
    - Pull
- Fundamentals of SSH communication between local and remote repositories (in our case GitHub).

Note: The yellow highlighted words which you will find in the steps below indicate the current git branch you are on.

Problem Statement

This assignment is a 4-step challenge.

- **Step-1**: Create a local Git repository from scratch. In assignment-1, we made things easy by providing you a ready repository which could be forked and cloned. In this assignment, we have gone one level further by requiring you to create a git repository from scratch which is what is frequently required in a real world scenario. Here we have used file names such as testfile1,testfile2 etc. However it is not mandatory to use files of such names. The name of artifacts (files and directories) can be different. Whatever be the name of your artifacts, you are expected to carry out exactly the same Git operations as mentioned in all the steps of the assignment. Even you can use a different repo name if you like so.

- **Step-2**: This step is essentially a setup process wherein you will build a commit history required for this assignment. It's simple and fun to build up a clean commit history for your repo.

- **Step-3**: Undoing commit history is a salient step towards mastering Git but sometimes ignored mistakenly. In the real world, there is not always a happy path forward. One invariably makes mistakes which needs to be undone to proceed in the right direction. Even though every undoing commands have not been covered in this assignment, enough has been provided as an exercise to get you going so that you are well prepared for the upcoming tests and final project as you progress further ahead in the course.

- **Step-4**: It's not always fun to work locally. GitHub throws in that extra spice to make version control more enjoyable while working with collaborators in a team. Here you will learn to implement the important skills pull and push which are the most fundamental operations while working with both downstream and upstream repositories. Assume that "origin" the name of the remote reference available locally. Also you are required to work with a fresh new repository that has been provided for you to fork and clone.

Approach: Follow the steps and complete the assignment

1. **Step-1: Creation of Git repo (repository) from scratch**
    a. Downloading project: Either you can use one of your existing projects which are unversioned, meaning it's not part of a git repo, or you can download a template project zipped file from http://www.initializr.com/.
    b. Let's call the project "assignment2-repo" for the sake of naming it.
    c. Change directory into *assignment2-repo*.
    d. Initialize the repo with the necessary git command.

2. **Step-2: Setting up the repository**
    a. Add all the project artifacts (files and folders) to the staging area.
    b. Commit all artifacts.
    c. Building commit history: Modify file or files to build up an initial commit history of 6 commits in total for the purpose of our assignment that might look like this if we execute the command "git log --oneline"
        i. commit-id-5 commit-message-5 (**HEAD**)
        ii. commit-id-4 commit-message-4
        iii. commit-id-3 commit-message-3
        iv. commit-id-2 commit-message-2
        v. commit-id-1 commit-message-1
        vi. commit-id-0 initial commit
    d. Let's assume we have 3 files called testfile1, testfile2, and testfile3 for the sake of this assignment.

3. **Step-3: Undoing changes in the repository:** Let’s say you either made some mistakes or discovered some bug or changed your mind related to changes you made in the repo. As a result, you want to undo your changes. So following are some undo operations which you need to carry out (note: the yellow highlighted words indicates the current git branch you are on
    a. Working with the checkout command:
        i. (master) Checkout to *commit-id-3*
        ii. (commit-id-3) Modify testfile1 in some way and commit it. Let’s say the commit message is “commit-message-7” (and commit-id-7)
        iii. (testbranch1) Persist the modification in the previous action by checking out to a new branch, let’s say *testbranch1*
        iv. (testbranch1) Switch to master
        v. (master) Let’s say, you are not happy with the current state of *testfile2* and you want to undo the change by checking out the *testfile2* to *commit-id-2.* Run the necessary command
        vi. (master) Commit the changes in the previous action. Let’s say the commit message is “commit-message-8” (and commit-id-8)
    b. Working with the revert command:
        i. (master) Let’s say, you are not happy with *commit-id-8* (current HEAD) and you want to revert back to *commit-id-7.* Run the necessary command
    c. Working with the reset command:
        i. (master) Let’s modify *testfile2* and add it to the staging area
        ii. (master) Let's decide not to commit the change in the previous action and instead reset the changes in the staging area only. Run the necessary command
        iii. (master) Let’s modify *testfile3* and add it to the staging area
        iv. (master) But as an afterthought, we decide not to commit the change in the previous action and reset the changes in both the staging area and the working directory. Run the necessary command

4. **Step-4: Synchronizing changes in the repository with the remote repo**
    a. Visit this repo - https://github.com/bibroy/git-github-simple-demo
    b. Fork this repo git-with-github-demo.
    c. Clone the repo git-with-github-demo.
    d. Edit the file 404.html (make any simple change) in your local machine.
    e. Add the changes to staging and commit the file. Let’s say the commit message is “commit-message-9” (and commit-id-9).
    f. Now push the changes to the remote repo.
    g. Now Edit the file 404.html (make any simple change) directly in the remote repo (normally one does not make changes in the remote repo directly; rather in a team scenario, some team member submits a change and other team members pull in those changes; here we are directly changing

in GitHub for the purpose demonstration of the pull command) and commit the changes; Let’s say the commit message is “commit-message-10” (and commit-id-10).

h. Now synchronize the local repo with remote repo by pulling in the changes in your local repo.

Submission

After completing the assignment, upload the text solution file. This file should contain the following:

- All git commands run for each of the steps of the assignment/project. Each command should start with “$” + single space
- Each command should be preceded by a line (or lines) of comment starting with a “#” + single space. The comment should describe the purpose of execution of the command.
- If the assignment/project involves working with GitHub, describe the steps taken in a step by step manner. The steps can be explained using comments (beginning with a “#” + single space)
- Each step mentioned in the assignment should be separated from other steps by few blank i.e new lines. You need to mention the step number, optionally with a description mentioning the purpose..

Solution

You can download the solution to the assignment from the progress report once you have uploaded the solution. You may compare your solution with the provided one.