Git Assignment-1

Introduction:

This assignment aims to consolidate skills related to concepts learned in Module 1 and Module 2. In this assignment, the student is expected to do hands-on in the following areas:

- Basics of GitHub operations such as:
    - Fork
    - Clone
- Basic Git operation such as:
    - Git Commit
    - Git Status
- Basics of Git Branches

Problem Statement:

This assignment is a 5-step challenge.

- **Step-1**: Create your GitHub account if you don’t have any. You are expected to fork a public repo and then make changes to the repo & commit those changes. You have been given the link to a GitHub repository which you can fork to proceed to the next step. Although theoretically you can fork any public repo and complete the assignment by following the rest of the steps but in practical terms, it’s always recommended to use the repository that has been given to you in Step-1.
- **Step-2**: Once the fork process is complete, you need to clone the repository on your local computer. Choose a root directory wisely where you want to keep and centrally manage all your repos. This will be useful to complete the rest of the assignments and final project.
- **Step-3**: In this step, you need to be sure of the git commands such as “add”, “commit”, “status”, “log” etc. Note that unless you have configured any modern text editors as your default git editor, you should use the “vi” or “vim” (vi improved) editor for editing all your git artifacts. vi and vim editors are available by default.
- **Step:4**: This step requires you to be acquainted with basic commands for checking commit history. Few nuances have been added so that we control the display of commit history.
- **Step-5**: Git Branching, arguably the most important Git feature, needs to be understood to carry out the items in this final step. You also need to know how to effectively navigate Git branches.

Approach:

Follow the steps and complete the assignment.

Step-1**: GitHub**

a. Account creation:

i) Go to GitHub at https://github.com.

ii) Register and Create a GitHub account if you don’t have an account already.

b. Fork:

i. Go to the repository (henceforth referred to as *repo*)

https://github.com/bibroy/sample-java-project

ii. Now Fork the above repo. You may note down the data “no of times forked” before and after forking the repo. This will give you an idea about the fork count. Also, notice the change in URL (in the browser address bar) after the fork is done.

c. Once you have forked the repo, copy the URL with “clone with HTTPS” mode enabled.

d. Now open the bash terminal on your local computer.

Step-2**: Running Clone operation locally**

a. Go to a suitable directory of your choice where you keep your clone repo.

b. Run the Clone command to clone the repo.

c. Change to the directory (cd) *sample-java-project* folder created due to clone operation.

d. List out the files and folders inside the repo.

e. Also, list out the .git folder (If you are not able to find the .git folder then something has gone wrong, check the above steps and find out what went wrong).

Step-3**: Modifying, Committing changes in repo and Status check**

a. Using vi editor, edit the README.md ([1] You can also pick up any file other than README.md if you want to do so. [2] You can modify the file in any way you want; you just need to make some changes such as adding some text that makes sense.)

b. Check the repository status; confirm if everything is OK (check status using both default and short option).

c. Add the README.md file to the staging area.

d. Check the repository status; confirm if everything is OK (check status using both default and short option).

e. Commit your changes after adding a suitable commit message (let’s say “commit 1”).

f. Check the repository status; confirm if everything is OK (check status using both default and short option).

g. Edit the README.md again (you can make any change, it does not matter).

h. Now do an *express commit* after adding a suitable commit message (let’s say “commit 2”). i. Check the repository status; confirm if everything is OK (check status using both long and short options).

Step-4**: Running the commands for checking commit history**

a. Run the default command without any options (hint: default command - “git log”). Check if you can see the commits you made with commit messages, “commit 1” and “commit 2”. Press the **space bar** to scroll history or type “**q**” at “:” to quit.

b. Run the command with the “--oneline” option.

c. Run the command to display the last 5 recent commits along with the “--oneline” option

Step-5**: Playing with Git Branches**

a. Run command to display all the existing Git Branches.

b. Create a branch named “branch1”.

c. Now display all the existing Git Branches. Confirm the branch “branch 1” was created.

d. Now checkout the branch “branch1”.

e. Now display the commit history for branch “branch1”.

f. Now checkout to master branch.

g. Now display the commit history for the master branch.

h. Note if there is any difference between the commit histories of the branches - branch 1 and master.