

INTRODUCTION TO GIT AND GITHUB

By

Md. Mobasser Hossen

ID: 2104010202277

CSE306: Software Engineering and Information System Design Lab



Instructor:

MD. Tamim Hossain

Lecturer

Department of Computer Science and Engineering

Premier University

Signature

Department of Computer Science and Engineering

Premier University

Chattogram-4000, Bangladesh

18 November, 2023

Abstract

This lab report delves into the fundamental concepts of Git and GitHub, aiming to address the need for efficient version control & collaboration or collaborative software development. The research explores the reasons behind conducting the experiment and problems associated with traditional versioning methods. Through hands-on exploration, participation successfully implemented Git & GitHub, showcasing the tool's capabilities in managing changes handling branches and facilitating collaboration. The result signifies the power of Git & GitHub in providing a streamlined development process improved collaboration & a robust version control mechanism. The findings enhance the understanding of version control concepts & demonstrate the tool's

role in addressing collaboration challenges, paving the way for more effective and organized software development practices.

Introduction to Git and GitHub:

Git and GitHub have emerged as indispensable tools in the realm of software development, revolutionizing the way teams collaborate, track changes and manage codebases. Git, a distributed version control system, provides a robust framework for tracking modifications to source code, allowing for efficient collaboration among developers. On the other hand, GitHub, a web-based platform built around Git, serves as a centralized hub for hosting repositories, fostering collaboration through features like pull

requests, issues tracking, and seamless integration with Git workflows.

Here, is the 25 commands of a Git:

1. git init % Initializes a new Git repository.
2. git clone <repository> % Creates a copy of a remote repository on your local machine.
3. git add <file> % Stage changes for the next commit.
4. git commit -m "message" % Records changes to the repository with a descriptive message.
5. git status % Displays the status of changes as untracked, modified or staged.
6. git diff % Shows the differences between working directory, staging area & the last commit.

7. git log: List commit history, including commit messages and SHA-1 hashes.

8. git branch: Lists all branches in repository.

9. git branch <branch-name>: Switched to the specified branch. Create new branch.

10. git merge <branch-name>: Integrates changes from one branch into another.

11. git checkout <branch-name>: Switched to the specified branch.

12. git remote -v: Lists all remote repositories associated with the current repository.

13) git pull <remote> <branch> % Fetches

changes from a remote repository and merge them into the current branch.

14) git push <remote> <branch> % Pushes

local commit to a remote repository.

15) git fetch % Retrieves changes from a

remote repository without merging.

16) git reset % Unstages changes preserving

modifications in the working directory.

17) git revert <commit> % Creates a new

commit that Undoes changes made in a previous commit

(18) git rm <file>% Removes a file from both the working directory and the staging area.

(19) git tag <tag-name>% Creates lightweight tag to label specific points in history.

(20) git stash % Temporarily saves changes that are not ready to be committed.

(21) git remote add <name> <url>% Adds a new remote repository.

(22) git remote remove <name>: Removes a remote repository.

(23) git config --global user.name "Your Name":

Sets the author name to be used for all commits.

(24) git config --global user.email: Sets the author email to be used for all commits.

(25) git log --graph --online --all% Displays a concise graphical representation of the commit history.

Method and Materials:

In this lab, I'm help from PDF which I have taken the help of pdf while reporting this, which is provided by the course teacher. And the 25 commands helped by

the git cheat sheet.

Result / Activity :

Activity 1 : Create Git Repo and txt script

1. At first, Create directory which to set as my repository in my location :

```
$ mkdir Lab1-Assignment
```

* this command is should be in
~documents/Lab1-Assign-ment

2. Initialize directory as repository.

```
$ git init
```

```
$ git config --global init.defaultBranch main
```

```
$ git branch main
```

3. To use config add name and email :

```
$ git config --global user.name "aryannveer"
```

```
$ git config --global user.email "mfmobasserhossen  
aryan1998@gmail.com"
```

4. Create a txt script in my directory & which I create
2 txt script:

Home_Wok_1

Home-Work_2

5. Inside the file, add code to print text "Hello Guys"
`printf("Hello Guys")`

6. Inside the file, code to print text "Hello world"
`printf("Hello world")`

7. Add this txt script main branch and commit
this script:

`$ git add Home-Wok_1.txt`

`$ git commit -m "Home-Wok-1 file added"`

8. To add this file into Github main branch, which is linked
to Github account:

`$ git remote add origin https://github.com/`

anjanveer/2104010202977_Mobasser

*this link from github which is you create

```
$ git branch -M main
```

```
$ git push -u origin main
```

9. Now add 2nd txt script into main branches (GitHub)

```
$ git add h1 Home-Work-2.txt
```

```
$ git add .
```

```
$ git commit -m "Home-Work-2 file added"
```

```
$ git push -u origin main
```

Activity 02 Create Branches and Merge into Main branch

1. Create newBranch and create txt script into directory which is already created (Lab1_Assignment)

```
$ git checkout -b newBranch
```

```
$ git add .
```

```
$ git commit -m "New_Home-Work file added"
```

2. Push this script into main Branch & Switched into main, and merge branch into main Branch :

\$ git push -u origin main

\$ git push -u origin newBranch

\$ git checkout main

✕ switched into branch 'main'

\$ git merge newBranch

\$ git push -u origin main

\$ git pull

Here, which i created 5 more branches & it's followed to the step by step to first one.

Discussion:

In this job, I missed the first class due to which I co-opted the class with a section on Wednesday. I faced this problem during the first day of class because I was new, I gave `$ git config --global.name` instead of `$ git config --global user user.name`, then this `$ git add Lab1.tut` was having problem to add it. I fixed them later.

Conclusion:

The exploration of Git and GitHub in this report has illuminated their pivotal roles in modern software development. The robust

version control capabilities of Git, coupled with the collaborative features offered by GitHub, form a symbiotic relationship that enhances project management and fosters efficient teamwork. The hands-on experience of creating repositories, branching and merging has provided valuable insights into the power & flexibility of these tools. As we navigate the dynamic landscape of software development Git and GitHub stand as essential pillars empowering developers to iterate, collaborate and contribute seamlessly.

References%

- 25 commands from Gitcheat sheet.
- I have taken the help from PDF while reporting this which is provided by the course teacher.
- Upload PDF file into Github from Youtube channel name: Amarindaz

—o—