

INTRODUCTION TO GIT AND GITHUB

By

Mahidul Islam

ID: 2104010202300 (D)

Semester: 5th

Date of submission : 18 November, 2023

CSE306: Software Engineering and Information System Design Lab

Conclusion



Instructor:

MD. Tamim Hossain

Lecturer

Department of Computer Science and Engineering

Premier University

Tamim 18.11.23

Signature

Department of Computer Science and Engineering

Premier University

Chattogram-4000, Bangladesh

Abstract:

The research aimed to investigate the utilization of git and github, modern version control systems, within collaboration software development environments.

The experiment delved into the fundamental functionalities of git, such as branching, merging and version tracking with Github platform specific features like - pull requests, issue tracking and repository management.

The experiment provides insights into the pivotal role of git and github in modern software development practices.

The results signify the power of git and github in providing a streamlined development process, improve collaboration.

Introduction :

Git: Git is a distributed version control system designed to track changes in source code during software development. It allows multiple developers to collaborate on projects, keeping tracking of modifications, facilitating team work and enabling easy integration of changes.

Github: Github is a web based platform that utilizes git for version control and offers additional features like bug tracking, task management. It provides a graphical interface and hosting service for git repositories, allowing users to store their projects, collaborate with others and manage their codebase.

25 commands of git

1. `$ git init` \Rightarrow create new local repository
2. `$ git pull` \Rightarrow Download and commits from remote repository
3. `$ git add .` \Rightarrow stage all changes
4. `$ git push` \Rightarrow Push local commits to remote repository
5. `$ git status` \Rightarrow List all new/modified files to be committed

6. `$ git log` \Rightarrow List version history of current branch
7. `$ git branch` \Rightarrow List all local branches in current repo.
8. `$ git branch <branch-name>` \Rightarrow create a new branch
9. `$ git merge <branch>` \Rightarrow Merges the specified branch into the current branch
10. `$ git add <file>` \Rightarrow Adds a specific file to the staging area
11. `$ git tag` \Rightarrow Lists all tags in the repository
12. `$ git commit` \Rightarrow commit previously staged changes
13. `$ git commit -a` \Rightarrow commit all local changes in tracked files.
14. ~~the~~ `$ git commit --amend` \Rightarrow change the last commit
15. `$ git branch -av` \Rightarrow List all existing branches
16. `$ git remote -v` \Rightarrow List all currently configured remotes
17. `$ git remote show <remote>` \Rightarrow show information about a remote
18. publish your tags (`$ git push --tags`)
19. `$ git branch -d <branch>`
 \Rightarrow deleted a merged branch

20. `$ git diff` (show changes to unstaged files)

21. `$ git diff commit1 ID`
`commit2 ID` (show changes between two commits)

22. `$ git checkout <branch>`

⇒ switch to a branch and update the working directory.

23. `$ git checkout -b <new branch>`

⇒ create a new branch and switch to it

24. `$ git config --global user.name "[your name]"`

`$ git config --global user.email "[email-address]"`

⇒ With this command, you can set the author's name, email address

25. `$ git push -u origin`

⇒ if you have created a new branch, it also needs to be pushed to the remote repository using the following command.

Activity / Result:

Activity 1: create a git repo

1. create a directory which to set as my repository in my location:

```
$ mkdir lab1 assignment
```

2. Initialize the directory as a repository:

```
$ git init
```

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

```
$ git branch -m main
```

3. use config to add name and email:

```
$ git config --global user.name "mahid76k"
```

```
$ git config --global user.email "mdmahidulislam776@gmail.com"
```

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

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

4. create a txt script in my directory which I created
txt script:

lab_1

lab_2

5. Inside the file, Write code to print the text
"Hello World!"

```
print("Hello World!")
```

6. Inside the file, Write code to print the text
"Hello Bangladesh!"

```
print("Hello Bangladesh!")
```

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

```
$ git add lab_1.txt
```

```
$ git commit -m "lab1 file added"
```

8. To add this file into Github main branch. Which
is linked to Github account

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

mahid76K/2104010202300-D-

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

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

9. Now add 2nd txtscript into main branches
(Github)

```
$ git add lab2.txt
```

```
$ git add .
```

```
$ git commit -m "lab2 file added"
```

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


Activity 2:

create a txtscript

1. create a txtscript in my directory:

lab1.txt

2. Inside the file, write code to print the text

"Hello World!"

print ("Hello World!")

Activity 3:

change helloworld.txt and run it on the terminal.

Then add it, commit it and do git status and git diff.

a) Add file staging:

\$ git status

\$ git add lab1.txt

\$ git status

b) commit file in staging

\$ git commit -m "lab1 file added"

\$ git log

\$ git status

discussion: From this experiment we know about ^{basics of} git and github ~~basic~~. Which is very important if you want to be a software Engineer. How to create repository and create branches and merge into main branch and many commands we ~~we~~ learned from this experiment.

Reference:

1. www.geeksforgeeks.org

2. <https://build5hines.com>

3. <https://monovm.com>