INTRODUCTION TO GIT AND GITHUB

By

Refazul Hoque Priyom

ID: 2104010202267

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

22 November,2023

Abstract

This lab report is for the lab that aimed to introduce fundamental concepts and particular usage of Git, a distributed version controll system widely used in software development. The objective was to provide participants with hand-on experience in setting up a Git repository, tracking changes, and collaboration with peers. The lab covered essential Git commands and workflows, emphasizing the importance of version controll in managing project development efficiently.

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 change and manage codebase. Git a distributed version control system, provides a robust framework for tracking modification 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: creates a copy of a remote repository on your local machine.

3. git add: Stage changes for the next commit

4. g'it commit -m "message": Records changes to the repository with a descriptive manage.

5. Gait status: Display the status of changes as untracked, modified.

6. git diff: shows the differences between working directory, staging area & the last commit.

7. git log: List commit history, including commit message.

8. git branch: Lists all branches in repository.

9. git branch <branch name>: create new branch.

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

4. git checkout <branch_name>: Switched to the specified branch.

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

13. git pull <remote> <branch>: fetches change 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 modification 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 staging area.

19. git tag <tag_name>: Creates light weight tag to label specific points in history.

20. git flash: Temporarily saves changes that are not ready to be commited.

21. git remote add <name><url>: add a new remote repository.

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

23. git config --global user.name "Name": Sets the author name.

24. git config -- global user.Email : Set the author email.

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

Method and Materials:

In this lab, I have taken the help of PDF while reporting this, which provided by the course teacher. And the 25 command helped by the git cheat sheet

Result & Activity:

Activity 1: Create Git repo & text receipt

1. Initialize directory as repository:

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

2. To use config add name and email:

```
$ git config --global user.name "prajyom-ers"
$ git config --global user.email "refazulpay@gmail.com"
```

3. Create a text script in my directory which i create 2 text script.

~~Home — bash — 1~~

Priyom 1.

Priyom 2.

4. Inside the file code to print text

printf ("Hello Priyom")

~~5. Inside the file.~~

5. Add this txt script main branch and commit this script.

$ git add priyom 1. txt.

$ git commit -m "file added"

6. To add file into Github main branch, which is linked to github account.

$ git remote add origin <link>

$ git branch -M main.

$ git push -u origin main

Activity: 2 'Create Branches and Merge into main branch.

1. Create Report and create .txt script into a directory which already created.

```
$ git checkout -b Report.
$ git add .
$ git commit -m " Report added "
```

2. Push this script into ~~main~~ report & switched into main and merge branch into main Branch:

```
$ git push -u origin main.
$ git push -u origin report.
$ git checkout main.
$ git merge report.
$ git push -u origin main
$ git pull
```

Discussion:

In this lab, I face several problem while using the code it git bash. The language in case sensitive that's why withe space and the capital letter I face several problem. And in the last I face problen while pushing.

Conclusion: In conclusion, The lab experiment effectively achived its objectives and familiarizing participants with the Git and GitHub, providing a hands on experience in version control collaboration and project management. The insights gained are foundational for students entering the field of software development and the acquired skills are applicable across a wide spectrum of collaborative coding environments. This introductory lab

serves as a stepping stone for future endeavors, encouraging participants to adopt version control best practices in their coding journey.

Refferance:

- 25 command from Git cheat sheet.
- I have taken the help from PDF while reparting this which is provided by the course teacher.
- Upload PDF file into GitHub from Youtube channal name: "Amarindaz"