Abstract

The Objective of this study is to comprehensively explore and elucidate the core concepts and functionalities of Git & Github, focusing on their pivotal roles in modern software development practice. The research explores the reason behind conducting the experiment and problems associated with traditional versioning methods. Through hands on exploration, participation Successfully Implemented Git & Github, showcasing the tools capabilities in managing changes handling branches and facilating collaboration. The result signify the power of Git and Github in providing a streamlined development process, improved collaboration & a robust version control Mechanism.

Introduction to Git & GitHub:

Git is a distributed version control system that enables developers to track changes in source code during software development. It offers a robust framework for managing projects, allowing multiple contributors to work on the same codebase simultaneously. GitHub is a web-based platform built around Git, providing hosting services for software development projects. It enhances collaboration and facilities project management by offering several features

Here is the 25 Commands of Git:

1. **git init**: Initialize a new Git repository
2. **git clone <repository>**: Creates a copy of a remote repository on your local machine
3. **git commit -m "message"**: Records changes to repository with a descriptive massage

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 massage and SHA 1 hasses.

8. git branch: Lists all branches in repository

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

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

11. git merge <branch-name>: Integrate changes from one branch into anthen

12. git checkout <branch-name>: switched to specified branch

13. git remote -v: Lists all remote repositoryies associated with current repository

13. git pull <remote> <branch>: Fetches changes from remote repository and merge them into current branch.

14. git push <remote> <branch>: Pushes local Commit to remote repository.

15. git fetch: Retrives 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 encodes 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 light weight 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 username "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 job, 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 Lab-1 assignment

** this command is should be documents/lab-assignment

2. Initialize directory as repository:

$ git init

$ git config --global init default branch ~~name~~ main

$ git branch main

3. To use configg add name and email:

$ git Configg --global user.name "~~Atc~~ Sifatmon

$ git Configg --global user.email "sifatnurmon @gmail.com"

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

Home_work 1

Home_work 2

5. Inside the file 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 text script main branch and commit this script:

$ git add Home_work1.txt

$ git commit -m "Home work 1 fill added"

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

$ git remote add origin.

https://github.com/Sifatmomo/210401020282_momo

=> 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 home.work 2 . txt
$ git add
$ git commit -m "Home.work 2 file added"
$ git Push -u origin main
```

Activity 2: Create Branches and Merge into Main branch

1. Create newbranch and create txt script into directory which already create "(Lab 1 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 merging 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 & its followed to the step by step ——to first one.

Discussion: In this lab, I faced lots of problem as it was my first lab. There was a lot of mistake. like i mistaked some spelling for that my result has been slightly changed. After fixing that I overcomed that.

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 ~~the~~ enhances project management and fosters efficient teamwork. The hand on experience

of creating repositories, branching and merging has provided valuable insights into the powerful flexibility of these tools. As we navigate the dynamic landscape of software development Git and Github stand as essentials pillares empowering developers to iterate collaborate and contribute seamlessly

References :

- 25 Commands from Git cheat sheet
- I have taken help from the pdf while reporting this which is Provided by the course teacher