INTRODUCTION TO GIT AND GITHUB

By

Shabah Tasfiya

ID: 2104010202294

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: Git is a distributed version control system that enables multiple developers to collabrate on a project by tracking change to source code during software development. It allows user to work on different aspects of a project concurrently, merge their changes seamlessly, and maintain a detailed history of revisions. Github, on the other hand, is a web-based platform built on top of Git, providing a centralized hub for developers to host, share and collaborate on projects. It offers features like pull requests, issues tracking and project management tools, making it a powerful platform for collaborative software development.

Introduction: Git is an open-source, version control tool created in 2005 by developers working on the Linux operating system. Git is a version control system that allows developers to track changes in their code. It is a tool thats used to manage multiple versions of source code edits that are then transferred to files in a Git repository. Github is a company founded in 2008 that makes tools which integrate with git.

Github is a web-based hosting service for git repositories. It serves as a location for uploading copies of a Git repository.

Materials: Git, Github, Laptop.

Activity: 1. 'git init': Initializes a new Git repository.

2. 'git clone [url]': Creates a copy of a remote repository on your local machine.

3. 'git add [file]': Adds a file or changes to the staging area.

4. 'git commit -m "message"': Commits changes with a descriptive message.

5. 'git status': Shows the status of changes as untracked, modified or staged.

6. 'git pull': Fetches changes from a remote repository and merges them into your current branch.

7. 'git push': Pushes your local changes to a remote repository.

8. 'git branch': Lists all branches in the

repository.

9. 'git checkout [branch]': Switches to the specified branch.

10. 'git merge [branch]': Merges changes from one branch in to the current branch.

11. 'git log': Displays the commit history.

12. 'git diff': Shows the differences between the working directory and the last commit.

13. 'git remote -v'': Lists remote repositories and their urls.

14. 'git fetch': fetches changes from a remote repository without merging.

15. 'git reset [file]': Unstages a file while keeping it changes.

16. 'git revert [commit]': Reverts a commit by creating a new commit.

17. git branch -d [branch]': Deletes a branch.

18. 'git tag [tagname]': creates a light-weight tag at the current commit.

19. 'git remote add [name] [url]': Adds a new remote repository.

20. git remote remove [name]': Removes a remote repository.

21. 'git config': Sets configuration values for user information etc.

22. 'git stash': Temporally stores changes not ready to be committed.

23. 'git rm': Removes files from the working tree and index.

24. 'git grep': Prints lines matching a pattern in a repository.

25. 'git clean': Removes untracked files from the working tree.

Discussion: In my very first Lab experiment I introduced to Git and Github. I learned that Git is a distributed version control system that helps track changes in your code, allowing collaboration and managing project history efficiently. On other hand, Github is a web-based platform that utilizes Git for version control. I learned basic git command like git commit, git push etc to save change and to upload changes to a remote repository on GitHub.

Conclusion: Git and Github streamline the development process, fostering collaboration, version control, and efficient management of codebases for individuals and teams. They have become integral tools in modern software development workflows.