INTRODUCTION TO GIT AND GITHUB

By

Abdullah Al Rohan

ID: 2104010202288

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

Scanned with CamScanner

Abstract

This abstract summarizes the purpose, finding, and significance of the experiment. The research aimed to address a specific problem, leading to meaningful results that enhance understanding. By exploring and answering key questions, this abstract offers a concise overview of the entire lab report.

Introduction

Git is a distributed version control system that allows developers to track changes in their code, collaborate seamlessly, and maintain a comprehensive history of their project. GitHub, on the other hand, is a web-based platform built around Git, providing a centralized hub for version controlled repositories. It facilitates collaboration, issue tracking, and project management, making it a cornerstone for modern software development. Together, Git and GitHub empower teams to work efficiently and contribute to open-source projects on a global scale.

Scanned with CamScanner

Materials

1. Computer
2. Internet connection
3. GitHub software
4. Google.
5. Visual Studio.

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.

Scanned with CamScanner

9. git checkout [branch] — Switches to the specified branch.

10. git merge [branch] — Merges changes from one branch into 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 its 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 lightweight 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 --global username "Your Name" — Sets yo username globally.

Scanned with CamScanner

22. git config --global user.email "your@example.com" — Sets your email globally.

23. git log --oneline — Displays a simplified log with each commit on a single line.

24. git stash — Temporarily saves changes that are not ready to be committed.

25. git pull --rebase — Fetches changes and rebases your local changes on top of them.

Discussion

Analyzing Git and GitHub commands reveals collaborative strengths and potential pitfalls, particularly in branching and conflict resolution. Comparing results to similar experiments highlights Git's universal utility, though discrepancies may arise from varied implementations. Addressing errors, like command misuse, is crucial for accurate data. The study underscores the significance of stable version control but hints at gaps in knowledge, particularly in advanced commands. These findings prompt questions about workflow optimization and the exploration of advanced Git features for continual skill development.

Scanned with CamScanner

Conclusion

Git and GitHub streamline collaborative development, ensuring code integrity and efficient workflow. This underscores the importance of robust version control for successful project management. Future work involves exploring advanced Git features and optimizing strategies for enhanced collaboration and scalability.

Scanned with CamScanner