Exploring Useful and Miscellaneous Git Features

Objective

The objective of this lab assignment is to introduce students to various useful and miscellaneous features in Git. By the end of this assignment, students will be able to:

1. View commit history.
2. Revert to a previous commit.
3. Work with tags.
4. Stash changes.
5. Apply stashed changes.
6. View the differences between commits.

Prerequisites

1. Basic understanding of Git.
2. Git installed on your local machine.
3. A local Git repository.

Instructions

Step 1: Open Terminal or Command Prompt

Open your terminal (on macOS or Linux) or Command Prompt (on Windows).

Step 2: Navigate to Your Local Repository

Use the `cd` command to navigate to the directory of your local project. For example:

cd path/to/your/local/repository

Step 3: View Commit History

View the commit history of your repository using:

git log

Example:

$ git log

commit abcdef3... (HEAD -> master)

Author: Your Name <your.email@example.com>

Date: Thu Jan 9 10:00:00 2025 +0530

Add new feature

commit abcdef2...

Author: Your Name <your.email@example.com>

Date: Wed Jan 8 15:30:00 2025 +0530

Initial commit

Step 4: Revert to a Previous Commit

Revert to a previous commit using its commit hash. For example:

git revert abcdef2

Example:

$ git revert abcdef2

[master abcdef4] Revert "Initial commit"

2 files changed, 1 deletion(-)

delete mode 100644 file1.txt

delete mode 100644 file2.txt

Step 5: Work with Tags

Create a new tag for a commit using:

git tag -a v1.0 -m "Version 1.0"

Example:

$ git tag -a v1.0 -m "Version 1.0"

Step 6: View Tags

View all tags in the repository using:

git tag

Example:

$ git tag

v1.0

Step 7: Stash Changes

Stash your current changes to save them temporarily using:

git stash

Example:

$ git stash

Saved working directory and index state WIP on master: abcdef3 Add new feature

Step 8: Apply Stashed Changes

Apply the stashed changes back to your working directory using:

git stash apply

Example:

$ git stash apply

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

new file: feature.txt

Step 9: View Differences Between Commits

View the differences between two commits using:

git diff <commit1> <commit2>

Example:

$ git diff abcdef2 abcdef3

diff --git a/feature.txt b/feature.txt

```
new file mode 100644
index 0000000..abcdef3
--- /dev/null
+++ b/feature.txt
@@ -0,0 +1 @@
+This is a new feature.
```