📘 Git Commands Reference for Project Maintenance

By Md. Raju Mia

## 1. Initialize a New Git Repository

Initializes a new Git repository in the current folder.

git init

## 2. Add Remote GitHub Repository

Links your local repo with your GitHub repository.

git remote add origin https://github.com/USERNAME/REPO\_NAME.git

## 3. Add Files for Tracking (Stage Changes)

Stages all changed files for the next commit.  
  
To add specific file:  
 git add filename.txt

git add .

## 4. Commit Changes with a Message

Saves your staged changes in local history.

git commit -m "Your commit message here"

## 5. Push Changes to GitHub

Pushes your commits from local 'main' branch to remote.  
  
If branch was renamed:  
 git branch -M main  
  
If pushing for the first time:  
 git push -u origin main

git push origin main

## 6. Pull Remote Changes First (if needed)

Pulls and re-applies your local commits to avoid conflicts.

git pull --rebase origin main

## 7. Check Git Status

Shows the status of your working directory, staged files, and uncommitted changes.

git status

## 8. View Commit History

Full commit details.  
  
Short commit view:  
 git log --oneline

git log

## 9. View Commit Changes (Diff)

Shows the file changes made in a specific commit.  
  
Example:  
 git show a55f864

git show <commit\_hash>

## 10. Browse Old Commit Code (Read-Only)

Moves you to that commit state (detached HEAD).  
  
Go back to latest commit in main:  
 git checkout main

git checkout <commit\_hash>

## 11. Restore or Undo

Restore a modified file.  
  
Unstage a file:  
 git restore --staged filename.txt

git restore filename.txt

## 12. Rename or Move Folder

Just rename the folder in Explorer or use these commands.

mv OldFolderName NewFolderName  
git add .  
git commit -m "Renamed folder"  
git push

## 13. View Commits on GitHub

Example: https://github.com/Md-Raju-Mia/Android\_java\_Practice\_Project/commits/main

Go to your repo → Click on <> Code tab → Click on Commits

## Extra Tip: Create .gitignore File

Ignore unnecessary files by adding patterns to .gitignore.

\*.class  
\*.log  
/build/  
.idea/