**GitHub Commands for Developer:**

1. **Git Configuration:**

* git config --global user.name "Your Name" # Set your username
* git config --global user.email "your\_email@example.com" # Set your email
* git config --global core.editor "code --wait" # Set default editor (VS Code)
* git config --list # View all configuration settings

1. **Initialize a Repository:**

* git init# Initialize a new Git repository
* git clone <repo\_url># Clone an existing repository

1. **Working with Branches:**

* git branch # List branches
* git branch <branch\_name> # Create a new branch
* git checkout <branch\_name> # Switch to a branch
* git checkout -b <branch\_name> # Create and switch to a new branch
* git merge <branch\_name> # Merge a branch into the current branch
* git branch -d <branch\_name> # Delete a branch
* git branch -D <branch\_name> # Force delete a branch

1. **Staging and Committing Changes:**

* git status # Check the status of changes
* git add <file> # Stage a specific file
* git add . # Stage all changes
* git commit -m "Your commit message" # Commit staged changes
* git commit --amend -m "New message" # Amend last commit message

1. **Viewing History:**

* git log # View commit history
* git log --oneline --graph # View condensed commit history
* git log -p # View commit history with changes

1. **Working with Remote Repositories:**

* git remote add origin <repo\_url> # Link local repo to remote
* git remote -v # List remote repositories
* git push origin <branch\_name> # Push changes to a remote branch
* git pull origin <branch\_name> # Fetch and merge changes from remote
* git fetch origin # Fetch updates without merging
* git push --force # Force push changes (use cautiously)
* git push -u origin <branch\_name> # Set upstream branch for future pushes

**7. Undoing Changes**

* git reset <file> # Unstage a file
* git reset --hard HEAD # Reset working directory and staging area to last commit
* git reset --soft HEAD~1 # Undo last commit but keep changes staged
* git revert <commit\_hash> # Revert a specific commit by creating a new commit
* git checkout -- <file> # Discard local changes to a file

**8. Stashing Changes**

* git stash # Stash changes temporarily
* git stash list # View stashed changes
* git stash apply # Apply the most recent stash
* git stash drop # Remove the most recent stash
* git stash pop # Apply and remove the most recent stash

**9. Tags (Versioning)**

* git tag <tag\_name> # Create a new tag
* git tag -a <tag\_name> -m "message" # Create an annotated tag
* git tag # List tags
* git push origin <tag\_name> # Push a tag to remote
* git push origin --tags # Push all tags
* git tag -d <tag\_name> # Delete a tag
* git push origin --delete <tag\_name> # Delete a remote tag

**10. Working with Submodules**

* git submodule add <repo\_url> # Add a submodule
* git submodule update --init --recursive # Clone and initialize submodules
* git submodule foreach git pull origin main # Update submodules

**11. Advanced Git Commands**

* git rebase <branch> # Rebase current branch onto another
* git cherry-pick <commit\_hash> # Apply a specific commit from another branch
* git bisect start # Start bisecting to find a bad commit
* git reflog # View Git history including undone changes

**12. Deleting Remote & Local Branches**

* git branch -d <branch\_name> # Delete a local branch
* git branch -D <branch\_name> # Force delete a local branch
* git push origin --delete <branch> # Delete a remote branch