

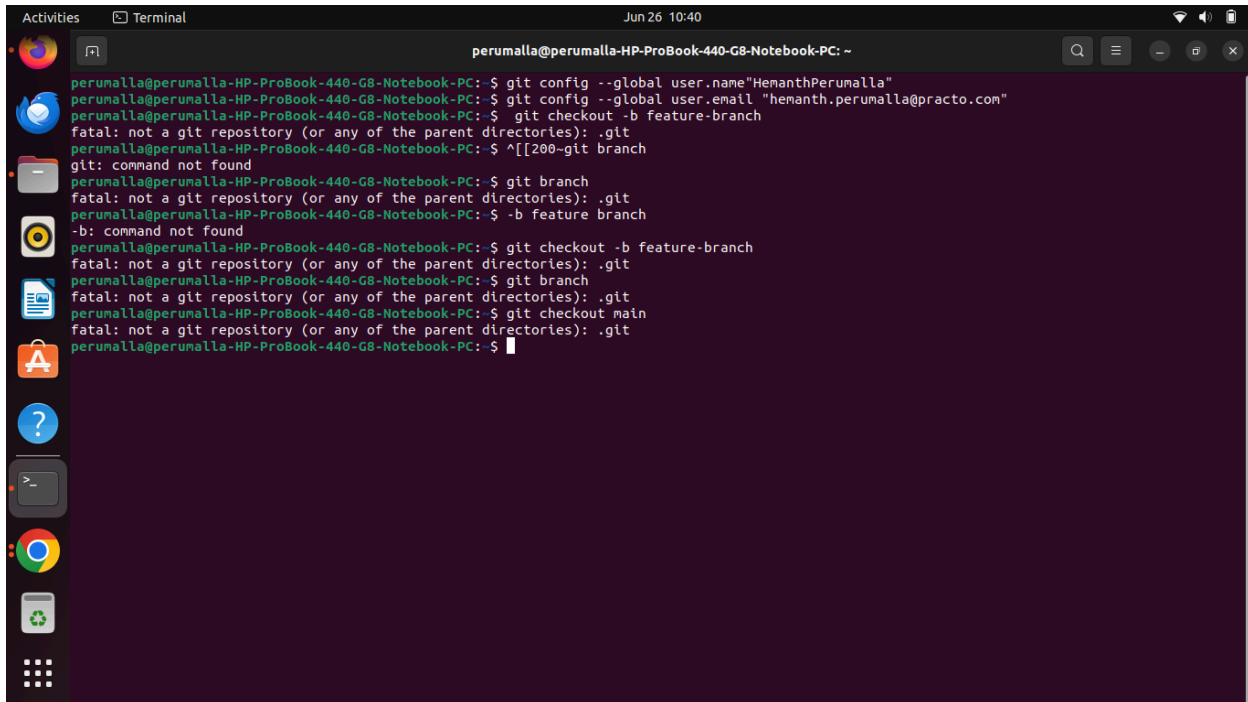
1. Set your username and email in git config

```
git config --global user.name "HemanthPerumalla"  
git config --global user.email "hemanth.perumalla@practo.com"
```

2. Create a new branch named "feature-branch" and switch to it.

```
git checkout -b feature-branch
```

I am facing issues with my branching so i am submitting all the git commands.



```
perumalla@perumalla-HP-ProBook-440-G8-Notebook-PC: ~  
perumalla@perumalla-HP-ProBook-440-G8-Notebook-PC: $ git config --global user.name "HemanthPerumalla"  
perumalla@perumalla-HP-ProBook-440-G8-Notebook-PC: $ git config --global user.email "hemanth.perumalla@practo.com"  
perumalla@perumalla-HP-ProBook-440-G8-Notebook-PC: $ git checkout -b feature-branch  
fatal: not a git repository (or any of the parent directories): .git  
perumalla@perumalla-HP-ProBook-440-G8-Notebook-PC: $ ^[[200~git branch  
git: command not found  
perumalla@perumalla-HP-ProBook-440-G8-Notebook-PC: $ git branch  
fatal: not a git repository (or any of the parent directories): .git  
perumalla@perumalla-HP-ProBook-440-G8-Notebook-PC: $ -b feature branch  
-b: command not found  
perumalla@perumalla-HP-ProBook-440-G8-Notebook-PC: $ git checkout -b feature-branch  
fatal: not a git repository (or any of the parent directories): .git  
perumalla@perumalla-HP-ProBook-440-G8-Notebook-PC: $ git branch  
fatal: not a git repository (or any of the parent directories): .git  
perumalla@perumalla-HP-ProBook-440-G8-Notebook-PC: $ git checkout main  
fatal: not a git repository (or any of the parent directories): .git  
perumalla@perumalla-HP-ProBook-440-G8-Notebook-PC: $
```

3. List all branches in the repository.

```
git branch
```

4. Delete the branch "feature-branch"

```
git branch -d feature-branch
```

5. How do you undo the last commit

By using revert and reset

The **revert** command will create a commit that reverts the changes of the commit being Targeted.

The reset command also works similar to revert but it changes the commit history.

```
git revert HEAD
```

```
git reset --hard HEAD~1
```

6. Create a new branch names “conflict-branch”

```
git checkout -b conflict-branch
```

7. Create a another branch named “feature1”

```
git checkout -b feature1
```

8. Make some changes in to feature1 branch

9. Merge “feature1” branch into main branch

```
git checkout main  
git merge feature1
```

10. Make changes in “conflict-branch”, in the same file and line that you had made changes in feature1

11. Merge master into conflict-branch [Attach screenshot of terminal & file]

```
git checkout conflict-branch  
git merge main
```

12. Resolve merge conflicts

Open the conflicted files and resolve the conflicts by editing the markers

<<<<<<, =====, and >>>>>>.

```
git add .  
git commit
```

13. Add a remote named "origin" pointing to a GitHub repository.

```
git remote add origin https://github.com/HemanthPerumalla/PractoTest.git
```

14. Fork a repository on GitHub and clone it to your local machine.

```
git clone https://github.com/HemanthPerumalla/forked-repository.git  
cd forked-repository
```

15. Create a new branch on your fork, make changes, and open a pull request to the original repository.

```
git checkout -b new-feature
```

```
git add .  
git commit -m "Added new feature"  
git push origin new-feature
```

16. Comment on a PR and suggest improvements

17. Create a Git alias for the command `git log --oneline` named `gitlol`.

```
git config --global alias.gitlol "log --oneline"
```

18. Create a pre-commit hook

19. You have made local changes in your branch, but you need to switch to another branch urgently without committing. How would you handle this situation?

```
git stash  
git checkout another-branch  
git checkout original-branch  
git stash pop
```

20. You accidentally deleted a file in your local repository. How do you restore it using Git?

```
git checkout -- PractoTest/temp.txt
```

21. You have committed changes to your branch but forgot to include a file. How do you add the file to the last commit without creating a new commit?

```
git add temp.txt  
git commit --amend --no-edit
```

22. You want to discard all changes in your working directory and revert to the last commit. What Git command would you use?

```
git reset --hard HEAD
```

23. You need to view a specific commit's changes. What Git command can be used to show the changes introduced by a particular commit?

```
git show <commit-hash>
```

24. You want to change a commit message, after you have already committed, how do you do so?

```
git commit --amend -m "New committed message"
```

25. Your colleague has made changes in their branch, and you want to incorporate those changes into your branch without merging. How do you achieve this?

```
git checkout intern-branch  
git fetch origin colleague-branch  
git cherry-pick commit-hash
```

26. You've made several commits on a branch, but you want to club them into a single commit before pushing to the remote repository. How would you do that?

```
git rebase -i HEAD~n
```

27. You accidentally staged a file that you don't want to commit. How do you unstage it?

```
git reset HEAD PractoTest/temp.txt
```

28. You don't want to commit files that have .yml in the end, and also files inside folder config. How do you do that?

```
*.yml  
config/
```

29. You want to see a list of all the files changed in the last commit. What Git command would you use?

```
git diff --name-only HEAD~1 HEAD
```

30. You realize that your local branch is outdated, and you want to fetch the latest changes from the remote repository. How do you do this without merging?

```
git fetch origin
```

31. You accidentally deleted a branch. How do you recover it?

```
git reflog  
git checkout -b deleted-branch-name commit-hash
```

32. You want to remove untracked files and directories from your working directory. What Git command would you use?

```
git clean -fd
```

33. You have a commit from a feature branch that you want to apply to the main branch without merging the entire feature branch.

```
git checkout main  
git cherry-pick commit-hash
```

34. You mistakenly committed a change to the wrong branch and need to apply that commit to the correct branch.

```
git checkout correct-branch  
git cherry-pick commit-hash
```

35. There is a series of commits on a feature branch, but you only want to cherry-pick a specific range of commits.

```
git cherry-pick commit-hashA^..commit-hashB
```

36. You want to clone a GitHub repository onto your local machine, but you only need a specific branch. How can you achieve this?

```
git clone --branch branch-name --single-branch  
https://github.com/HemanthPerumalla/repository.git
```

37. You've made changes to your local repository and want to push them to your fork on GitHub. What Git commands would you use?

```
git add .  
git commit -m "Pushing to the github"  
git push origin branch-name
```

38. You want to create a new branch both locally and on GitHub to work on a new feature. What commands would you use?

```
git checkout -b new-feature  
git push origin new-feature
```

39. You want to see the commit history of a GitHub repository. How can you do this using Git commands?

```
git log
```

40. You've accidentally committed sensitive information and want to remove the commit from both your local and remote repositories on GitHub. What commands would you use?

```
git log  
git revert bad_commit_hash
```

41. You want to delete a remote branch on GitHub. What Git command would you use?

```
git push origin --delete Temp
```

42. Create a git repository for all your assignments and upload them in it. Ask your peers to code review it, and you need to code review your peers assignments

43. Create a pull request on any open source library on github, attach the pull request link to the readme file of this project's repository