

1. Set your username and email in git config

```
[vallurusuraj@Surajs-MacBook-Pro ~ % git config --list
credential.helper=osxkeychain
user.name=Valluru Suraj
user.email=valluru.suraj@practo.com
user.username=ValluruSuraj
vallurusuraj@Surajs-MacBook-Pro ~ %
```

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

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git switch -c feature-branch
Switched to a new branch 'feature-branch'
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

3. List all branches in the repository.

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git branch
* feature-branch
  main
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

4. Delete the branch "feature-branch"

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git branch -d feature-branch
Deleted branch feature-branch (was 94bb61d).
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

5. How do you undo the last commit

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git add .
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git commit -m "Added Changes"
[main 3a57cc1] Added Changes
 1 file changed, 1 insertion(+)
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git reset HEAD^
Unstaged changes after reset:
M   README.md
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

6. Create a new branch names "conflict-branch"

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git switch -c conflict-branch
Switched to a new branch 'conflict-branch'
```

7. Create a another branch named "feature1"

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git switch -c feature1
Switched to a new branch 'feature1'
```

8. Make some changes in to feature1 branch

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git commit -m "Made changes in feature1 Branch"
[feature1 4db5741] Made changes in feature1 Branch
 1 file changed, 1 insertion(+)
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git
```

9. Merge "feature1" branch into main branch

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git merge feature1
Updating 94bb61d..4db5741
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git checkout conflict-branch  
Switched to branch 'conflict-branch'  
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git add .  
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git commit -m "Made changes in c  
onflict branch"  
[conflict-branch a052adf] Made changes in conflict branch  
1 file changed, 1 insertion(+)  
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git merge main  
Auto-merging README.md  
CONFLICT (content): Merge conflict in README.md  
Automatic merge failed; fix conflicts and then commit the result.  
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

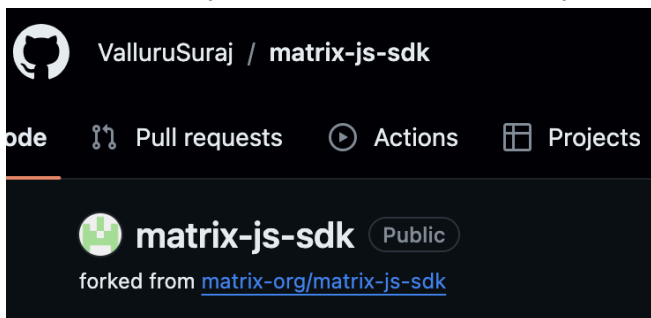
12. Resolve merge conflicts

```
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes  
<<<<<<< HEAD (Current Change)  
Changes in Conflict Branch  
=====
```

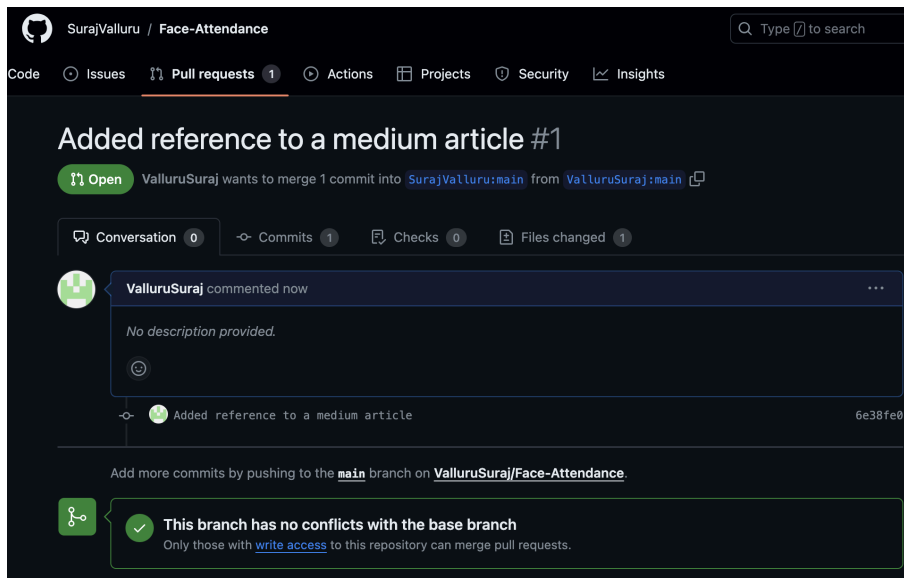
13. Add a remote named "origin" pointing to a GitHub repository.

```
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git remote add origin https://gi  
thub.com/ValluruSuraj/GitAssignment.git  
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git branch -M main
```

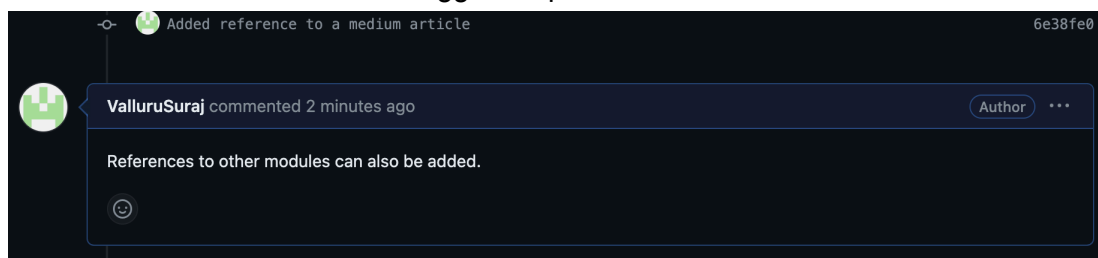
14. Fork a repository on GitHub and clone it to your local machine.



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



16. Comment on a PR and suggest improvements



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

```
[vallurusuraj@Surajs-MacBook-Pro Face-Attendance % git config --global alias.gitlol "log --oneline"]
```

```
[vallurusuraj@Surajs-MacBook-Pro Face-Attendance % git gitlol]
6e38fe0 (HEAD -> main, origin/main, origin/HEAD) Added reference to a medium article
2cb7691 Added option to voice over
4b16bcc Speaking Ability and Multi Face Detection
027e1b0 Added dates entry into report
c63d64a Fixes and Added Comments
e50b1c5 Error Fixed
5d357c9 Errors Handling
745e05e Removed use of Numpy
4f9519c Added Unknown Face Detection
a1e183e Added bxtTxt function
323851e Created Main File
70a2cad Fixes
21d316a Added prerequisites
b43b598 First Commit
vallurusuraj@Surajs-MacBook-Pro Face-Attendance %
```

18. Create a pre-commit hook

```
bin | < > bin
bin > No Selection

1 # Check if this is the initial commit
2 if git rev-parse --verify HEAD >/dev/null 2>&1
3 then
4     echo "pre-commit: About to create a new commit..."
5     against=HEAD
6 else
7     echo "pre-commit: About to create the first commit..."
8     against=4b825dc642cb6eb9a060e54bf8d69288fbee4904
9 fi
10
11 # Use git diff-index to check for whitespace errors
12 echo "pre-commit: Testing for whitespace errors..."
13 if ! git diff-index --check --cached $against
14 then
15     echo "pre-commit: Aborting commit due to whitespace errors"
16     exit 1
17 else
18     echo "pre-commit: No whitespace errors :)"
19     exit 0
20 fi
```

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?

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git stash
Saved working directory and index state WIP on main: 81b9aef Q.43 Answer
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git stash pop
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .DS_Store
    new file:   bin.sh

Dropped refs/stash@{0} (edc404a10cb3274ee4f9235a3a6d066634294fdf)
```


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

```
[vallurusuraj@Surajs-MacBook-Pro Face-Attendance % ls
Attendance.csv  Multi.jpg      README.md      known          main.py
[vallurusuraj@Surajs-MacBook-Pro Face-Attendance % rm README.md
[vallurusuraj@Surajs-MacBook-Pro Face-Attendance % ls
Attendance.csv  Multi.jpg      known          main.py
[vallurusuraj@Surajs-MacBook-Pro Face-Attendance % git checkout -- README.md
[vallurusuraj@Surajs-MacBook-Pro Face-Attendance % ls
Attendance.csv  Multi.jpg      README.md      known          main.py
```

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?

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git add .
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git commit --amend
[main 59c4d7b] Made changes in feature1 Branch and used git commit --amend to ad
d to latest commit
Date: Sat Jan 20 23:03:43 2024 +0530
1 file changed, 3 insertions(+)
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git push
```

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

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git reset --hard HEAD
HEAD is now at d8f376f Amending commit message
[vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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?

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git log
commit d8f376f4330f070223b227910dad2be26b80d5c5 (HEAD -> main)
Author: Valluru Suraj <valluru.suraj@practo.com>
Date: Sat Jan 20 23:03:43 2024 +0530

    Amending commit message

commit 94bb61d78fc2cb3cc982bd3f3a7e40107486182f
Author: Valluru Suraj <valluru.suraj@practo.com>
Date: Sat Jan 20 22:48:04 2024 +0530

    First Commit
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git show 94bb61d78fc2cb3cc982bd3f3a7e40107486182f
commit 94bb61d78fc2cb3cc982bd3f3a7e40107486182f
Author: Valluru Suraj <valluru.suraj@practo.com>
Date: Sat Jan 20 22:48:04 2024 +0530

    First Commit

diff --git a/README.md b/README.md
new file mode 100644
index 0000000..e69de29
[vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git commit --amend -m "Amending  
commit message"  
  
[main d8f376f] Amending commit message  
Date: Sat Jan 20 23:03:43 2024 +0530  
1 file changed, 3 insertions(+)  
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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?

```
usuraj@Surajs-MacBook-Pro GitAssignment % git cherry-pick 81b9aefd65232cd36bd94c35e5040c46fda21f8c  
: your local changes would be overwritten by cherry-pick.
```

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?

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git rebase -i HEAD~n  
  
fatal: invalid upstream 'HEAD~n'  
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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

```
Changes to be committed:  
(use "git restore --staged <file>..." to unstage)  
new file:   .DS_Store  
new file:   .gitignore  
deleted:    43Link.txt  
modified:   Git Assignment.pdf  
new file:   bin.sh  
  
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git reset .DS_Store  
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git status  
On branch main  
Your branch is up to date with 'origin/main'.  
  
Changes to be committed:  
(use "git restore --staged <file>..." to unstage)  
new file:   .gitignore  
deleted:    43Link.txt  
modified:   Git Assignment.pdf  
new file:   bin.sh  
  
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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?

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git touch .gitignore
git: 'touch' is not a git command. See 'git --help'.
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % touch .gitignore
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % vim .gitignore
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % cat .gitignore
*.yml
config/*
```

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

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git diff --name-only HEAD^
README.md
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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?

```
README.md
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git fetch origin
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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

```
4db5741 (origin/feature1, feature1) HEAD@{11}: commit: Made changes in feature1
Branch
94bb61d HEAD@{12}: checkout: moving from main to feature1
94bb61d HEAD@{13}: checkout: moving from feature to main
94bb61d HEAD@{14}: checkout: moving from conflict-branch to feature
94bb61d HEAD@{15}: checkout: moving from main to conflict-branch
94bb61d HEAD@{16}: reset: moving to HEAD^
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git branch recovered-branch 94bb61d
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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

```
[vallurusuraj@Surajs-MacBook-Pro GitAssignment % git clean -fdX
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```


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

```
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git cherry-pick 94bb61d78fc2cb3cc982bd3f3a7e40107486182f
error: your local changes would be overwritten by cherry-pick.
hint: commit your changes or stash them to proceed.
fatal: cherry-pick failed
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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

```
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git branch new-branch 94bb61d78fc2cb3cc982bd3f3a7e40107486182f
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git checkout main
M      README.md
Already on 'main'
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" if you want to integrate the remote branch with yours)
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git cherry-pick new-branch
```

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

```
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git cherry-pick d8f376f4330f0702123b227910dad2be26b80d5c5^ .. 94bb61d78fc2cb3cc982bd3f3a7e40107486182f
error: empty commit set passed
fatal: cherry-pick failed
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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

```
vallurusuraj@Surajs-MacBook-Pro Downloads % git clone -b feature1 https://github.com/ValluruSuraj/GitAssignment.git
Cloning into 'GitAssignment'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 15 (delta 2), reused 14 (delta 1), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (2/2), done.
vallurusuraj@Surajs-MacBook-Pro Downloads %
```

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?

```
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git push origin feature1
Everything up-to-date
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```


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

```
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git checkout -b new-branch
Switched to a new branch 'new-branch'
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git push origin new-branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'new-branch' on GitHub by visiting:
remote:   https://github.com/ValluruSuraj/GitAssignment/pull/new/new-branch
remote:
To https://github.com/ValluruSuraj/GitAssignment.git
 * [new branch]      new-branch -> new-branch
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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

```
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git log
commit d8f376f4330f070223b227910dad2be26b80d5c5 (HEAD -> main)
Author: Valluru Suraj <valluru.suraj@practo.com>
Date:   Sat Jan 20 23:03:43 2024 +0530

    Amending commit message

commit 94bb61d78fc2cb3cc982bd3f3a7e40107486182f (recovered-branch, new-branch)
Author: Valluru Suraj <valluru.suraj@practo.com>
Date:   Sat Jan 20 22:48:04 2024 +0530

    First Commit
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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?

```
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git reset HEAD^
Unstaged changes after reset:
M       README.md
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git push -f
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ValluruSuraj/GitAssignment.git
+ 59c4d7b...94bb61d main -> main (forced update)
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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

```
vallurusuraj@Surajs-MacBook-Pro GitAssignment % git push origin --delete conflict-branch
To https://github.com/ValluruSuraj/GitAssignment.git
- [deleted]          conflict-branch
vallurusuraj@Surajs-MacBook-Pro GitAssignment %
```

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

I've executed all the instructions as mentioned and committed the changes too. Few commands like deleting remote branch and resetting last commit to remove sensitive data erased my commits.

Link: <https://github.com/ValluruSuraj/GitAssignment>

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

Link: <https://github.com/formulahendry/vscode-code-runner/pull/853>