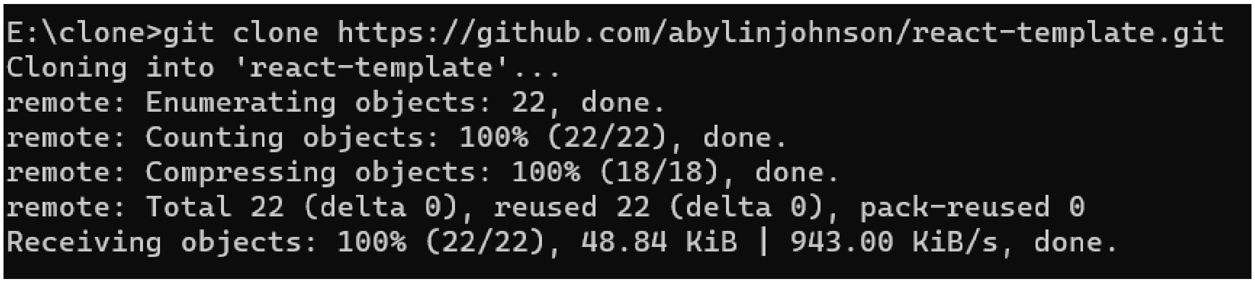
Problem Statement 8:Cloning and Pulling

• Clone a repository from a remote source using git clone.

• Make changes to the remote repository and use git pull to bring those changes into the local repository.

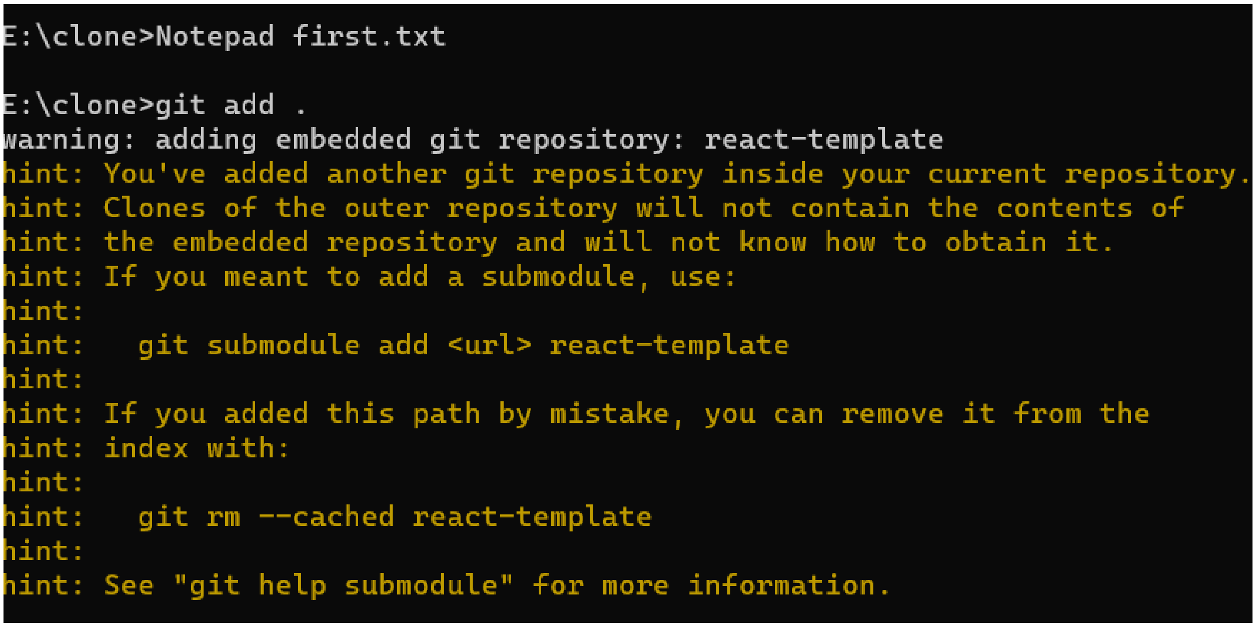
Solution :

Clone a repository from a remote source using git clone.



```
E:\clone>git clone https://github.com/abylinjohnson/react-template.git
Cloning into 'react-template'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 22 (delta 0), reused 22 (delta 0), pack-reused 0
Receiving objects: 100% (22/22), 48.84 KiB | 943.00 KiB/s, done.
```

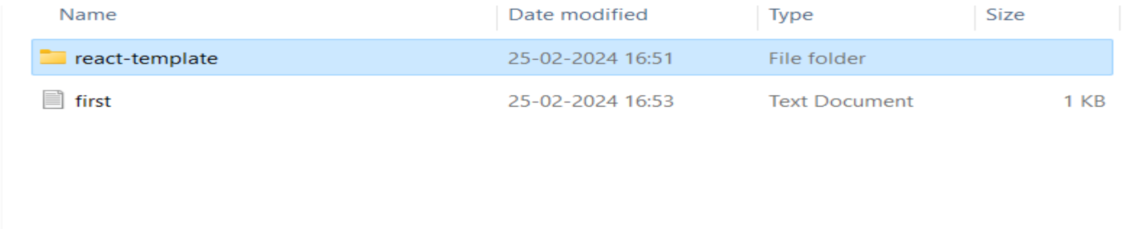
Make changes to the remote repository use git pull to bring those changes into the local repository.



```
E:\clone>Notepad first.txt

E:\clone>git add .
warning: adding embedded git repository: react-template
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> react-template
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached react-template
hint:
hint: See "git help submodule" for more information.
```

File added to local repository.



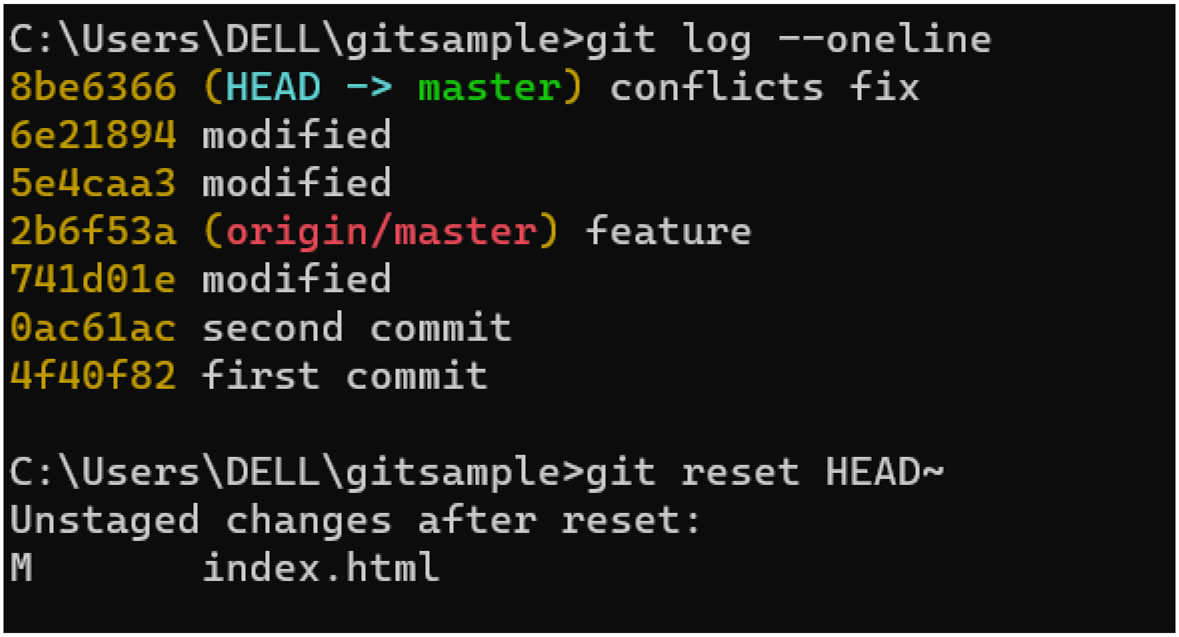
Name	Date modified	Type	Size
react-template	25-02-2024 16:51	File folder	
first	25-02-2024 16:53	Text Document	1 KB

Problem Statement 9: Undoing Changes

- Experiment with git reset and git revert to undo changes at different levels.
- Use git clean to remove untracked files.

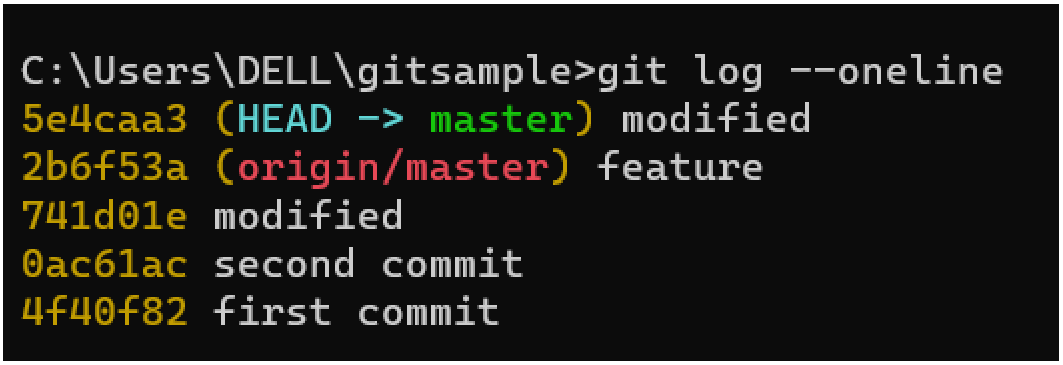
Solution :

Git reset - This command will move the branch pointer and the HEAD back one commit, effectively "undoing" the last commit.



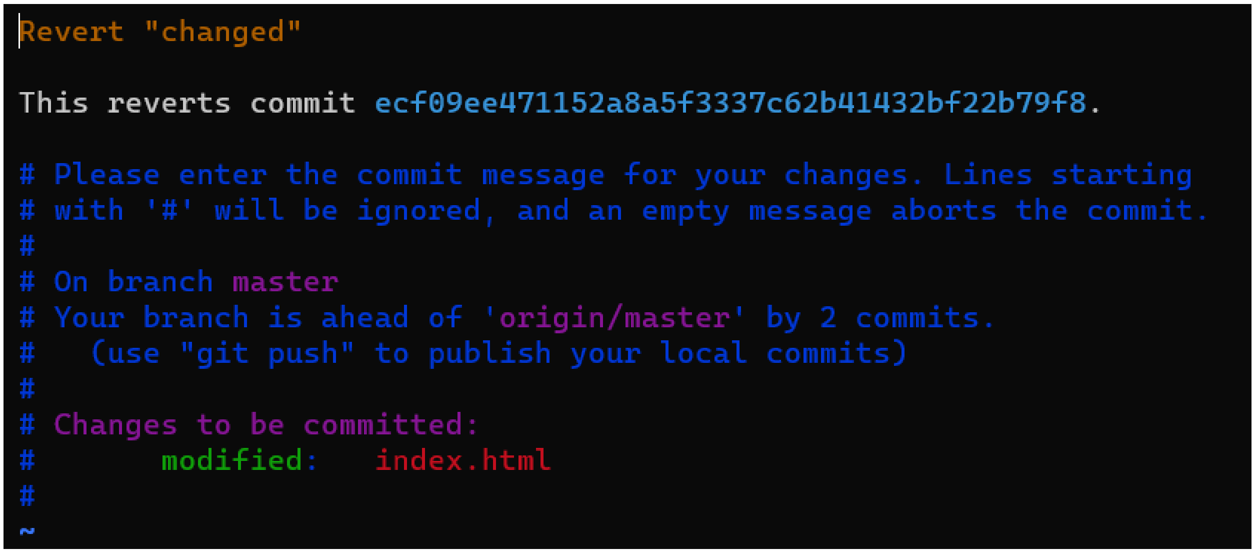
```
C:\Users\DELL\gitsample>git log --oneline
8be6366 (HEAD -> master) conflicts fix
6e21894 modified
5e4caa3 modified
2b6f53a (origin/master) feature
741d01e modified
0ac61ac second commit
4f40f82 first commit

C:\Users\DELL\gitsample>git reset HEAD~
Unstaged changes after reset:
M       index.html
```



```
C:\Users\DELL\gitsample>git log --oneline
5e4caa3 (HEAD -> master) modified
2b6f53a (origin/master) feature
741d01e modified
0ac61ac second commit
4f40f82 first commit
```

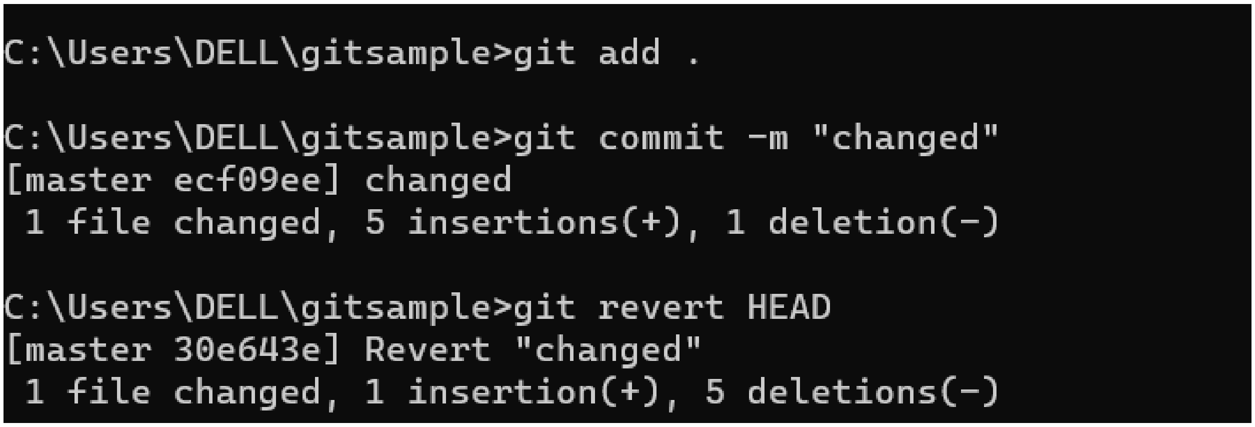
Git revert - This command will create a new commit that undoes the changes introduced in the specified commit



```
Revert "changed"

This reverts commit ecf09ee471152a8a5f3337c62b41432bf22b79f8.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Your branch is ahead of 'origin/master' by 2 commits.
#   (use "git push" to publish your local commits)
#
# Changes to be committed:
#       modified:   index.html
#
~
```



```
C:\Users\DELL\gitsample>git add .

C:\Users\DELL\gitsample>git commit -m "changed"
[master ecf09ee] changed
 1 file changed, 5 insertions(+), 1 deletion(-)

C:\Users\DELL\gitsample>git revert HEAD
[master 30e643e] Revert "changed"
 1 file changed, 1 insertion(+), 5 deletions(-)
```

Use git clean to remove untracked files:

# To dry-run and see what will be deleted without actually deleting anything
git clean -n

# To interactively choose which untracked files to delete
git clean -i

# To forcefully remove untracked files (use with caution)
git clean -f

```
# To remove untracked files and directories recursively
git clean -fd
```

- The `-n` option performs a dry-run, showing you a list of files that would be deleted without actually deleting them.
- The `-i` option runs the clean operation interactively, allowing you to choose which untracked files to delete.
- The `-f` option forces the deletion of untracked files without asking for confirmation.
- The `-d` option is used to also remove untracked directories.