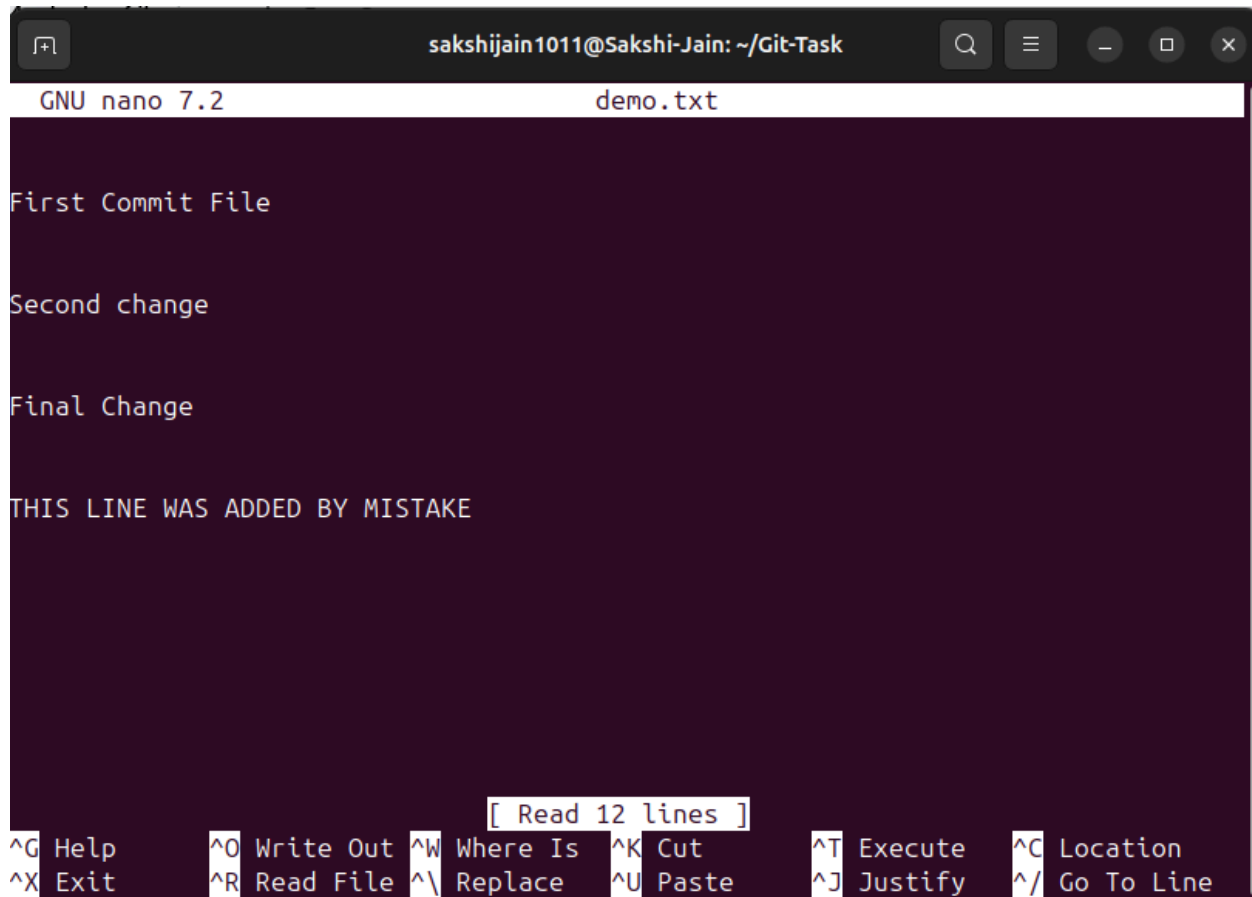


1. Make a wrong commit intentionally

Made Changes Intentionally in demo file in [feature/sakshi/squash-demo](#) branch



```
sakshijain1011@Sakshi-Jain: ~/Git-Task
GNU nano 7.2 demo.txt

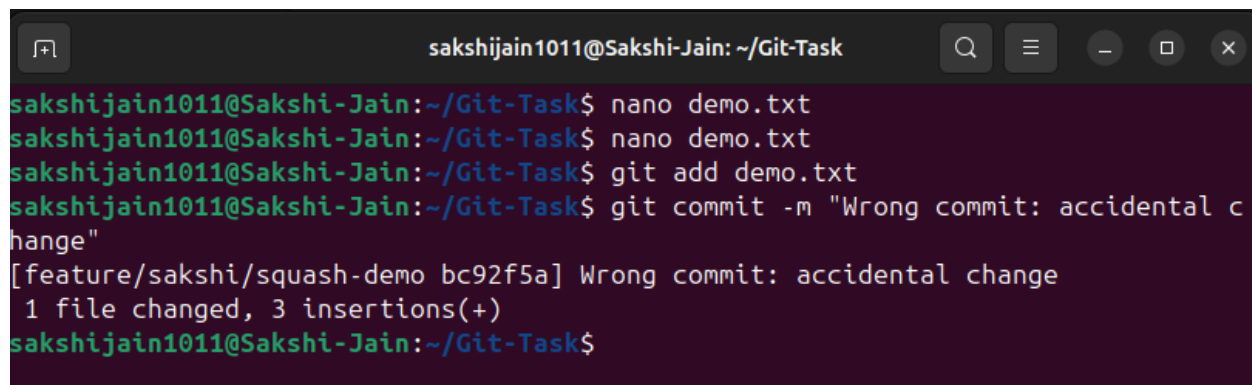
First Commit File

Second change

Final Change

THIS LINE WAS ADDED BY MISTAKE

[ Read 12 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```



```
sakshijain1011@Sakshi-Jain: ~/Git-Task
sakshijain1011@Sakshi-Jain:~/Git-Task$ nano demo.txt
sakshijain1011@Sakshi-Jain:~/Git-Task$ nano demo.txt
sakshijain1011@Sakshi-Jain:~/Git-Task$ git add demo.txt
sakshijain1011@Sakshi-Jain:~/Git-Task$ git commit -m "Wrong commit: accidental change"
[feature/sakshi/squash-demo bc92f5a] Wrong commit: accidental change
1 file changed, 3 insertions(+)
sakshijain1011@Sakshi-Jain:~/Git-Task$
```

2. Undo it using: git reset --soft OR git revert

I am using revert to undo the commit

```
sakshijain1011@Sakshi-Jain: ~/Git-Task
GNU nano 7.2 /home/sakshijain1011/Git-Task/.git/COMMIT_EDITMSG
Revert "Wrong commit: accidental change"

This reverts commit bc92f5a4f087323875a6c4d35d7ccb89456a68ec.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch feature/sakshi/squash-demo
# Changes to be committed:
#   modified:   demo.txt
#
```

```
sakshijain1011@Sakshi-Jain: ~/Git-Task
sakshijain1011@Sakshi-Jain:~/Git-Task$ git log --oneline
f6ef5a5 (HEAD -> feature/sakshi/squash-demo) Revert "Wrong commit: accidental change"
bc92f5a Wrong commit: accidental change
68b2d36 (origin/feature/sakshi/squash-demo) Final demo files
cc111db (origin/main, main) Merge pull request #1 from sakshijain-josh/feature/sakshi/basic-program
06a44db (origin/feature/sakshi/basic-program, feature/sakshi/basic-program) Add reverse string program
853fe95 Added Second COmmit
f7d6a7b Added First COmmit
sakshijain1011@Sakshi-Jain:~/Git-Task$ git push origin feature/sakshi/squash-demo
Username for 'https://github.com': sakshijain-josh
Password for 'https://sakshijain-josh@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 616 bytes | 616.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sakshijain-josh/Git-Task.git
  68b2d36..f6ef5a5 feature/sakshi/squash-demo -> feature/sakshi/squash-demo
sakshijain1011@Sakshi-Jain:~/Git-Task$
```

3. Explain : Which command you used Why

Why git revert was used

git revert creates a new commit that reverses the changes introduced by the specified commit.

This approach was chosen because:

- It preserves commit history
- It is safe for shared branches, since it does not rewrite history.
- Reviewers can clearly see:
 - the original (wrong) commit
 - and the explicit commit that undoes it

Why git reset --soft was not used

Although git reset --soft can undo a commit locally by moving HEAD while keeping changes staged, it was not appropriate here because:

- It rewrites commit history
- It can cause issues if the commit has already been pushed
- It is better suited for local-only cleanup before pushing