

Challenges of Git & Github

1. Resolve Merge Conflicts

>> Here, we have created file named “techie1” by adding content “hello people” in release branch and created same file named “techie1” by adding content “hello world”

>> by merging these 2 main & release branches I have created a conflict.

```
[sainiharikagundu@192 git-practice % vi techie1
[sainiharikagundu@192 git-practice % git add .
[sainiharikagundu@192 git-practice % git commit -m "added content in techie1"
[release 60e18fb] added content in techie1
 5 files changed, 11 insertions(+)
 create mode 100644 docker
 create mode 100644 passwords.txt
 create mode 100644 techie1
 create mode 100644 yes
 create mode 100644 yes.pub
[sainiharikagundu@192 git-practice % cat techie1
hello people
[sainiharikagundu@192 git-practice % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
[sainiharikagundu@192 git-practice % vi techie1
[sainiharikagundu@192 git-practice % git add .
[sainiharikagundu@192 git-practice % git commit -m "added content in techie1"
[main 8382efc] added content in techie1
 1 file changed, 1 insertion(+)
 create mode 100644 techie1
[sainiharikagundu@192 git-practice % git merge main
Already up to date.
[sainiharikagundu@192 git-practice % cat techie1
hello world
[sainiharikagundu@192 git-practice % git merge release
Auto-merging techie1
CONFLICT (add/add): Merge conflict in techie1
Automatic merge failed; fix conflicts and then commit the result.
sainiharikagundu@192 git-practice % ]
```

>> here we can see the conflict message

```
<<<<< HEAD
hello world
=====
hello people
>>>>> release
~
```

>> to resolve this issue we can change the the content of file

```
[sainiharikagundu@192 git-practice % vi techie1
[sainiharikagundu@192 git-practice % git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Last commands done (2 commands done):
  pick b946557 # deleted
  squash 4321084 # alphabet
Next command to do (1 remaining command):
  squash f9bb881 # changed the commit message
  (use "git rebase --edit-todo" to view and edit)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Changes to be committed:
  new file:  docker
  new file:  passwords.txt
  new file:  yes
  new file:  yes.pub

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both added:      techie1

[sainiharikagundu@192 git-practice % cat techie1
hello world and people
sainiharikagundu@192 git-practice % ]
```

>> git add .
>> git commit -m “modified the file”
>> git push -u origin main

```
[sainiharikagundu@192 git-practice % cat techie1
hello world and people
[sainiharikagundu@192 git-practice % git add .
[sainiharikagundu@192 git-practice % git commit -m
error: switch `m' requires a value
[sainiharikagundu@192 git-practice % git commit -m "modified the file"
[main 02d78c5] modified the file
[sainiharikagundu@192 git-practice % git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 10 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 1.42 KiB | 1.42 MiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To github.com:Sai-Gaya3/git-practice.git
  9b6d35c..02d78c5  main -> main
branch 'main' set up to track 'origin/main'.
[sainiharikagundu@192 git-practice % git merge
Already up to date.
sainiharikagundu@192 git-practice % ]
```

The screenshot shows a GitHub repository interface. At the top, there are navigation links: Actions, Projects, Wiki, Security, Insights, and Settings. Below this, a commit history is displayed for the repository `git-practice / techie1`. The most recent commit, shown in a box, was made by `Sai-Gaya3` and modified the file `techie1`. The commit message is `modified the file`. Below the commit box, there are tabs for `Code` and `Blame`, with `Code` being the active tab. It shows 1 line (1 loc) and 23 Bytes. The code content is `1 hello world and people`.

2. Recover Deleted Branch

- Delete a local branch and then recover it using the reflog.

>> Here, initially we have created a branch named “sai”

>> then switched to sai >> switched to main branch and deleted the new branch by using “git branch -D sai” where we will get a commit id

>>We can see by reflog we can se 7edfceb it is commit name of sai branch

>>then by using “git checkout -b sai 7edfceb” This will recover the deleted branch

```
[sainiharikagundu@192 git-practice % git branch
 * gaya3
   main
   release
[sainiharikagundu@192 git-practice % git branch sai
[sainiharikagundu@192 git-practice % git checkout sai
Switched to branch 'sai'
[sainiharikagundu@192 git-practice % ls
ansible      bharath      docker      file1      file2      hello      loke      README.md      techie
[sainiharikagundu@192 git-practice % git status
On branch sai
nothing to commit, working tree clean
[sainiharikagundu@192 git-practice % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
[sainiharikagundu@192 git-practice % ls
ansible      docker      file1      file2      loke      passwords.txt      README.md      techie1      yes      yes.pub
[sainiharikagundu@192 git-practice % git branch -d sai
error: the branch 'sai' is not fully merged
hint: If you are sure you want to delete it, run 'git branch -D sai'
hint: Disable this message with "git config set advice.forceDeleteBranch false"
[sainiharikagundu@192 git-practice % git branch -D sai
Deleted branch sai (was 7edfceb).
[sainiharikagundu@192 git-practice % git reflog
02d78c5 (HEAD -> main, origin/main, origin/HEAD) HEAD@{0}: checkout: moving from sai to main
7edfceb (gaya3) HEAD@{1}: checkout: moving from gaya3 to sai
7edfceb (gaya3) HEAD@{2}: rebase (finish): returning to refs/heads/gaya3
7edfceb (gaya3) HEAD@{3}: rebase (squash): changed the commit message
f9bb881 HEAD@{4}: rebase (start): checkout HEAD~2
d49d584 HEAD@{5}: commit: added file bharath
f9bb881 HEAD@{6}: rebase (finish): returning to refs/heads/gaya3
f9bb881 HEAD@{7}: rebase (start): checkout HEAD~5
f9bb881 HEAD@{8}: rebase (abort): returning to refs/heads/gaya3
b946557 HEAD@{9}: rebase (start): checkout HEAD~3
f9bb881 HEAD@{10}: rebase (abort): returning to refs/heads/gaya3
02d78c5 (HEAD -> main, origin/main, origin/HEAD) HEAD@{11}: commit (merge): modified the file
8382efc HEAD@{12}: commit: added content in techie1
9b6d35c HEAD@{13}: checkout: moving from release to main
60e18fb (release) HEAD@{14}: commit: added content in techie1
ec48dae (origin/release) HEAD@{15}: checkout: moving from main to release
9b6d35c HEAD@{16}: checkout: moving from b94655717af40a162aa69bd5e1a76947295b5e to main
b946557 HEAD@{17}: rebase (start): checkout HEAD~3
f9bb881 HEAD@{18}: rebase (finish): returning to refs/heads/gaya3
f9bb881 HEAD@{19}: rebase (start): checkout HEAD~3
f9bb881 HEAD@{20}: commit (amend): changed the commit message
1522442 HEAD@{21}: commit: hello file
4321084 HEAD@{22}: reset: moving to HEAD
4321084 HEAD@{23}: revert: Revert "deleted docker file"
b946557 HEAD@{24}: commit: deleted docker file
95596e2 HEAD@{25}: reset: moving to HEAD~1
d49d587 HEAD@{26}: commit: added file3
95596e2 HEAD@{27}: commit: ignore passwords file
dedc86d (origin/gaya3) HEAD@{28}: clone: from github.com:Sai-Gaya3/git-practice.git
[sainiharikagundu@192 git-practice % git checkout -b sai 7edfceb
Switched to a new branch 'sai'
[sainiharikagundu@192 git-practice % git branch
 * gaya3
   main
   release
 * sai
[sainiharikagundu@192 git-practice % ]
```

3. Undo Wrong Push

>> Here, we've created a file named "file3" added, committed and pushed to central repository

>> We copied the commit and git revert that commit after pushing

>> after git push We can see revert "wrong commit"

```
[sainiharikagundu@192 git-practice % vi file3
[sainiharikagundu@192 git-practice % cat file 3
cat: file: No such file or directory
cat: 3: No such file or directory
[sainiharikagundu@192 git-practice % cat file3
wrong commit
[sainiharikagundu@192 git-practice % git status
On branch sai
Your branch is up to date with 'origin/sai'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file3

nothing added to commit but untracked files present (use "git add" to track)
[sainiharikagundu@192 git-practice % git add .
[sainiharikagundu@192 git-practice % git push -u origin sai
branch 'sai' set up to track 'origin/sai'.
Everything up-to-date
[sainiharikagundu@192 git-practice % git commit -m "wrong commit"
[sai 9a5f7c7] wrong commit
 1 file changed, 1 insertion(+)
 create mode 100644 file3
[sainiharikagundu@192 git-practice % git push -u origin sai
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 10 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 279 bytes | 279.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Sai-Gaya3/git-practice.git
  7edfce..9a5f7c7 sai -> sai
branch 'sai' set up to track 'origin/sai'.
[sainiharikagundu@192 git-practice % git revert 9a5f7c7
[sai 1b5e454] Revert "wrong commit"
 1 file changed, 1 deletion(-)
 delete mode 100644 file3
[sainiharikagundu@192 git-practice % git revert 9a5f7c7
On branch sai
Your branch is ahead of 'origin/sai' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
[sainiharikagundu@192 git-practice % git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 10 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 256 bytes | 256.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Sai-Gaya3/git-practice.git
  9a5f7c7..1b5e454 sai -> sai
sainiharikagundu@192 git-practice % ]
```

The screenshot shows a GitHub repository page for 'git-practice'. At the top, there are navigation links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The repository name 'git-practice' is displayed with a green 'Public' badge. A message at the top says 'sai had recent pushes 21 seconds ago'. On the right, there are 'Pin' and 'Watch' buttons. Below the header, it shows '4 Branches' and '0 Tags'. A search bar and file navigation buttons ('Go to file', 'Add file', 'Code') are present. A status bar indicates the branch is '7 commits ahead of main'. The commit history lists the following commits:

Author	Commit Message	Time
Sai-Gaya3	Revert "wrong commit"	1b5e454 · 7 minutes ago
	.gitignore	ignore passwords file 16 hours ago
	README.md	Initial commit yesterday
	ansible	added file ansible 19 hours ago
	bharath	changed the commit message 13 hours ago

4. Ammend a commit

Make a commit, then add a missing file to it using `git commit -amend`.

>> here, initially we are creating a file named “file-a”, adding it and committed it.

```
[sainiharikagundu@Sais-MacBook-Air git-practice % echo "hello" > file-a
[sainiharikagundu@Sais-MacBook-Air git-practice % git add .
warning: adding embedded git repository: git-practice
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> git-practice
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached git-practice
hint:
hint: See "git help submodule" for more information.
hint: Disable this message with "git config set advice.addEmbeddedRepo false"
[sainiharikagundu@Sais-MacBook-Air git-practice % git commit -m " added file-a"
[release 36f8e0b] added file-a
2 files changed, 2 insertions(+)
create mode 100644 file-a
create mode 160000 git-practice
```

```
[sainiharikagundu@Sais-MacBook-Air git-practice % git push -u origin release
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 10 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 381 bytes | 381.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Sai-Gaya3/git-practice.git
  ec48dae..36f8e0b  release -> release
branch 'release' set up to track 'origin/release'.
sainiharikagundu@Sais-MacBook-Air git-practice %
```

>> Here, we need to create another file but instead of committing we need to use “ git commit —amend”

```
[sainiharikagundu@sais-MacBook-Air git-practice % echo "missing file" > file-b
[sainiharikagundu@sais-MacBook-Air git-practice % git add .
[sainiharikagundu@sais-MacBook-Air git-practice % git commit --amend
[release 210ef3b] added file-a
Date: Sun Dec 21 19:17:44 2025 +0530
3 files changed, 3 insertions(+)
create mode 100644 file-a
create mode 100644 file-b
create mode 160000 git-practice
sainiharikagundu@sais-MacBook-Air git-practice %
```

```
added file-a

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Sun Dec 21 19:17:44 2025 +0530
#
# On branch release
# Your branch is up to date with 'origin/release'.
#
# Changes to be committed:
#       new file:   file-a
#       new file:   file-b
#       new file:   git-practice
#
# Changes not staged for commit:
#       modified:  git-practice (untracked content)
#
~
```

>> save and exit

>> now, force push the file

```
sainiharikagundu@sais-MacBook-Air git-practice % git push --force origin release

Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 10 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 434 bytes | 434.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Sai-Gaya3/git-practice.git
 + 36f8e0b...210ef3b release -> release (forced update)
sainiharikagundu@sais-MacBook-Air git-practice %
```

	Sai-Gaya3	added file-a
git-practice		added file-a
README.md		Initial commit
file-a		added file-a
file-b		added file-a
file1		added 2 files
file2		added 2 files

```
sainiharikagundu@sais-MacBook-Air git-practice % git log -1
commit 210ef3b167fc12c54990675af4021df4441e3157 (HEAD -> release, origin/release)
Author: sai gayathri <saigayathriprasad305@gmail.com>
Date:   Sun Dec 21 19:17:44 2025 +0530

    added file-a
sainiharikagundu@sais-MacBook-Air git-practice %
```

5. Cherry-pick a Commit

Take a specific commit from one branch and apply it to another branch.

>> Here, we are currently in main branch and copied the commit id of ansible file and checked out to the release branch and pasted the commit id of ansible in release branch by using “git cherry-pick <commitid> “

```
sainiharikagundu@sais-MacBook-Air git-practice % git checkout main
M      git-practice
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
sainiharikagundu@sais-MacBook-Air git-practice % ls
ansible  docker  file-a  file1  file2  git-practice  loke  passwords.txt  README.md  techie1  yes  yes.pub
sainiharikagundu@sais-MacBook-Air git-practice % git log
commit 385fe2747663426dced87af4e135b8e8648ce9d7 (HEAD -> main)
Author: sai gayathri <saigayathriprasad305@gmail.com>
Date:   Sun Dec 21 19:13:23 2025 +0530

    added file-a

commit 02d78c5c23903535f795921f22de32f62158b425 (origin/main, origin/HEAD)
Merge: 8382efc 60e18fb
Author: sai gayathri <saigayathriprasad305@gmail.com>
Date:   Wed Dec 10 20:31:51 2025 +0530

    modified the file

commit 8382efc5b281992e63d73a5f377553af1f1d3c427e
Author: sai gayathri <saigayathriprasad305@gmail.com>
Date:   Wed Dec 10 20:28:59 2025 +0530

    added content in techie1

commit 60e18fb94de850481af13c637a0e52aeb6fbe792
Author: sai gayathri <saigayathriprasad305@gmail.com>
Date:   Wed Dec 10 20:18:04 2025 +0530

    added content in techie1

commit 9bd4d35cbc89e1bfcd8bcfaf93a255d487c85d994
Merge: c73b978 56fcf78
Author: Sai-Gaya3 <saigayathriprasad305@gmail.com>
Date:   Wed Dec 10 15:04:30 2025 +0530

    Merge pull request #1 from Sai-Gaya3/gaya3

    creating pull request

commit 56fcf7838d34f411cd0bd237373e2632abfcfbce
Author: sai gayathri <saigayathriprasad305@gmail.com>
Date:   Wed Dec 10 14:39:58 2025 +0530

    added file ansible
```

>> now we can see the ansible file in release branch as well.

```
[sainiharikagundu@sais-MacBook-Air git-practice % git checkout release
M      git-practice
Switched to branch 'release'
Your branch is up to date with 'origin/release'.
[sainiharikagundu@sais-MacBook-Air git-practice % ls
file-a      file-b      file1      file2      git-practice  loke          README.md
[sainiharikagundu@sais-MacBook-Air git-practice % git cherry-pick 56fcf7
[release 9e659e8] added file ansible
  Date: Wed Dec 10 14:39:58 2025 +0530
  1 file changed, 1 insertion(+)
  create mode 100644 ansible
[sainiharikagundu@sais-MacBook-Air git-practice % ls
ansible      file-a      file-b      file1      file2      git-practice  loke          README.md
sainiharikagundu@sais-MacBook-Air git-practice % ]
```

6. Interactive Rebase

- Reorder and squash multiple commits into a single clean commit.

>> here, we are showing multiple commits in single commit by merging
>> git rebase -i HEAD~3

```
[sainiharikagundu@sais-MacBook-Air git-practice % git rebase -i HEAD~3
warning: unable to rmdir 'git-practice': Directory not empty
[detached HEAD 8d6eec1] added file loke
  Date: Wed Dec 10 13:22:02 2025 +0530
  5 files changed, 5 insertions(+)
  create mode 100644 ansible
  create mode 100644 file-a
  create mode 100644 file-b
  create mode 160000 git-practice
  create mode 100644 loke
Successfully rebased and updated refs/heads/release.
[sainiharikagundu@sais-MacBook-Air git-practice % git log
```

```
[sainiharikagundu@sais-MacBook-Air git-practice % git log
commit 9e659e82a93d4001d3ee07c7c7e849c4a086b747 (HEAD -> release)
Author: sai gayathri <saigayathriprasad305@gmail.com>
Date:   Wed Dec 10 14:39:58 2025 +0530

  added file ansible

commit 210ef3b167fc12c54990675af4021df4441e3157 (origin/release)
Author: sai gayathri <saigayathriprasad305@gmail.com>
Date:   Sun Dec 21 19:17:44 2025 +0530

  added file-a

commit ec48daee3bb2da0c778fef7827d53d8dc6ab2ab6
Author: sai gayathri <saigayathriprasad305@gmail.com>
Date:   Wed Dec 10 13:22:02 2025 +0530

  added file loke
```

```

pick ec48dae # added file loke
squash 210ef3b # added file-a
squash 9e659e8 # added file ansible

# Rebase c73b970..9e659e8 onto c73b970 (3 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
#                               commit's log message, unless -C is used, in which case
#                               keep only this commit's message; -c is same as -C but
#                               opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [<oneline>]
#           create a merge commit using the original merge commit's
#           message (or the oneline, if no original merge commit was
#           specified); use -c <commit> to reword the commit message
# u, update-ref <ref> = track a placeholder for the <ref> to be updated
#                       to this position in the new commits. The <ref> is
#                       updated at the end of the rebase
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#

```

```

# This is a combination of 3 commits.
# This is the 1st commit message:

added file loke

# This is the commit message #2:

added file-a

# This is the commit message #3:

added file ansible

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Wed Dec 10 13:22:02 2025 +0530
#
# interactive rebase in progress; onto c73b970
# Last commands done (3 commands done):
#   squash 210ef3b # added file-a
#   squash 9e659e8 # added file ansible
# No commands remaining.
# You are currently rebasing branch 'release' on 'c73b970'.
#
# Changes to be committed:
#       new file:  ansible
#       new file:  file-a
#       new file:  file-b
#       new file:  git-practice
#       new file:  loke
#
# Changes not staged for commit:
#       modified:  git-practice (untracked content)
#
~
```

>> After squash

```
[sainiharikagundu@sais-MacBook-Air git-practice % git log
commit 8d6eec13c5493084daec79fb402b5004ce6311a9 (HEAD -> release)
Author: sai gayathri <saigayathripasad305@gmail.com>
Date:   Wed Dec 10 13:22:02 2025 +0530

    added file loke

    added file-a

    added file ansible
```

7. Tagging & Release

- Create a version tag (v1.0), push it to GitHub, then delete and restore it.

>>here, in git tag = will show you all the tags in repo, “git tag v1.0” created a tag for commit.

>>git push origin v1.0 will push the v1.0 (tag) commit

>>Git tag -d v1.0 (will delete the tag)

>>git tag -a v1.0 -m “ restored v1.0 tag “ 8e4e356

means you are creating an annotated tag named v1.0 and adding a message to it.

>>We can simple use —> git tag v1.0 8e4e356To restore the tag

```
[sainiharikagundu@sais-MacBook-Air git-practice % git log -1
commit 385fe2747663426dcfd87af4e135b8e8648ce9d7 (HEAD -> main)
Author: sai gayathri <saigayathripasad305@gmail.com>
Date:   Sun Dec 21 19:13:23 2025 +0530

    added file-a
[sainiharikagundu@sais-MacBook-Air git-practice % git tag v1.0
[sainiharikagundu@sais-MacBook-Air git-practice % git log -1
commit 385fe2747663426dcfd87af4e135b8e8648ce9d7 (HEAD -> main, tag: v1.0)
Author: sai gayathri <saigayathripasad305@gmail.com>
Date:   Sun Dec 21 19:13:23 2025 +0530

    added file-a
[sainiharikagundu@sais-MacBook-Air git-practice % git push origin v1.0
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 10 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 317 bytes | 317.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Sai-Gaya3/git-practice.git
 * [new tag]           v1.0 -> v1.0
[sainiharikagundu@sais-MacBook-Air git-practice % git push origin v1.0
Everything up-to-date
[sainiharikagundu@sais-MacBook-Air git-practice % git tag -d v1.0
Deleted tag 'v1.0' (was 385fe27)
[sainiharikagundu@sais-MacBook-Air git-practice % git tag
[sainiharikagundu@sais-MacBook-Air git-practice % git push origin --delete v1.0
To github.com:Sai-Gaya3/git-practice.git
 - [deleted]           v1.0
[sainiharikagundu@sais-MacBook-Air git-practice % git tag -a v1.0 -m "restored v1.0 tag" 385fe2
[sainiharikagundu@sais-MacBook-Air git-practice % git tag
v1.0
sainiharikagundu@sais-MacBook-Air git-practice % ]
```

9. Reset vs Revert Challenge

- Demonstrate the difference between `git reset --hard` and `git revert` in a repo.

>>here, we have created a new commit and used “git reset —hard 02d78c5” to remove present head commit

```
[sainiharikagundu@Sais-MacBook-Air git-practice % git log --oneline
385fe27 (HEAD -> main, tag: v1.0) added file-a
02d78c5 (origin/main, origin/HEAD) modified the file
8382efc added content in techie1
60e18fb added content in techie1
9b6d35c Merge pull request #1 from Sai-Gaya3/gaya3
56fcf78 added file ansible
ec48dae added file loke
c73b970 added 2 files
7c74377 Initial commit
[sainiharikagundu@Sais-MacBook-Air git-practice % git reset --hard 02d78c5
warning: unable to rmdir 'git-practice': Directory not empty
HEAD is now at 02d78c5 modified the file
[sainiharikagundu@Sais-MacBook-Air git-practice % git log --oneline
02d78c5 (HEAD -> main, origin/main, origin/HEAD) modified the file
8382efc added content in techie1
60e18fb added content in techie1
9b6d35c Merge pull request #1 from Sai-Gaya3/gaya3
56fcf78 added file ansible
ec48dae added file loke
c73b970 added 2 files
7c74377 Initial commit
. . . . .
```

```
Revert "added abc"
```

```
This reverts commit 336fe68e547baa5e0517bc6ca18c1484d1cb5c55.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
# Your branch is ahead of 'origin/main' by 1 commit.
#   (use "git push" to publish your local commits)
#
# Changes to be committed:
#       deleted:    abc
#       deleted:    git-practice
#
# Untracked files:
#       git-practice/
#
~
```

>> by using “git revert 336fe68”, we can see revert “added abc” in head

```
[sainiharikagundu@Sais-MacBook-Air git-practice % touch abc
[sainiharikagundu@Sais-MacBook-Air git-practice % git add .
warning: adding embedded git repository: git-practice
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> git-practice
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:     git rm --cached git-practice
hint:
hint: See "git help submodule" for more information.
hint: Disable this message with "git config set advice.addEmbeddedRepo false"
[sainiharikagundu@Sais-MacBook-Air git-practice % git commit -m "added abc"
[main 336fe68] added abc
 2 files changed, 1 insertion(+)
 create mode 100644 abc
 create mode 160000 git-practice
[sainiharikagundu@Sais-MacBook-Air git-practice % git log --oneline
336fe68 (HEAD -> main) added abc
02d78c5 (origin/main, origin/HEAD) modified the file
8382efc added content in techie1
60e18fb added content in techie1
9b6d35c Merge pull request #1 from Sai-Gaya3/gaya3
56fcf78 added file ansible
ec48dae added file loke
c73b970 added 2 files
7c74377 Initial commit
[sainiharikagundu@Sais-MacBook-Air git-practice % git revert 336fe68
warning: unable to rmdir 'git-practice': Directory not empty
[main 47028fa] Revert "added abc"
 2 files changed, 1 deletion(-)
 delete mode 100644 abc
 delete mode 160000 git-practice
[sainiharikagundu@Sais-MacBook-Air git-practice % git log --oneline
47028fa (HEAD -> main) Revert "added abc"
336fe68 added abc
02d78c5 (origin/main, origin/HEAD) modified the file
8382efc added content in techie1
60e18fb added content in techie1
9b6d35c Merge pull request #1 from Sai-Gaya3/gaya3
56fcf78 added file ansible
ec48dae added file loke
c73b970 added 2 files
7c74377 Initial commit
sainiharikagundu@Sais-MacBook-Air git-practice % ]
```

10. Detached HEAD Challenge

- Checkout a specific commit (detached HEAD state) and create a new branch from it.

>> first we need to check the git log —oneline

>> now, git checkout 47028fa will Checkout the old commit “Detached HEAD”

```
[sainiharikagundu@sais-MacBook-Air git-practice % git log --oneline
47028fa (HEAD -> main) Revert "added abc"
336fe68 added abc
02d78c5 (origin/main, origin/HEAD) modified the file
8382efc added content in techie1
60e18fb added content in techie1
9b6d35c Merge pull request #1 from Sai-Gaya3/gaya3
56fcf78 added file ansible
ec48dae added file loke
c73b970 added 2 files
7c74377 Initial commit
[sainiharikagundu@sais-MacBook-Air git-practice % git checkout 47028fa
Note: switching to '47028fa'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -c with the switch command. Example:

```
git switch -c <new-branch-name>
```

Or undo this operation with:

```
git switch -
```

Turn off this advice by setting config variable advice.detachedHead to false

```
HEAD is now at 47028fa Revert "added abc"
[sainiharikagundu@sais-MacBook-Air git-practice % git branch fix-abc
[sainiharikagundu@sais-MacBook-Air git-practice % git checkout fix-abc
Switched to branch 'fix-abc'
```

>> now, create a branch from the commit and git branch

>> we can see HEAD detached at 47028fa

```
[sainiharikagundu@sais-MacBook-Air git-practice % git branch
* (HEAD detached at 47028fa)
  fix-abc
  gaya3
  main
  release
  sai
sainiharikagundu@sais-MacBook-Air git-practice %
```

11. Git Hooks Challenge

- Configure a pre-commit hook to reject commits without a message format (e.g., must start with JIRA-XXX).

>> go inside hooks directory and create a file name pre-commit

```
[sainiharikagundu@sais-MacBook-Air git-practice % cd .git/hooks  
[sainiharikagundu@sais-MacBook-Air hooks % vi pre-commit  
[sainiharikagundu@sais-MacBook-Air hooks % chmod +x pre-commit
```

>> we need to paste this rule for not config any commit with out jira-xxx

```
#!/bin/bash

# Get the commit message
MESSAGE=$(git log -1 --pretty=%B)

# Check if message starts with JIRA- followed by numbers
if ! grep -qE "^JIRA-[0-9]+" <<< "$MESSAGE"; then
    echo "Commit rejected!"
    echo "Your commit message MUST start with: JIRA-XXX"
    exit 1
fi

exit 0
```

~
~

>> here, we can see the commit is getting rejected without JIRA-XXX

```
[sainiharikagundu@sais-MacBook-Air hooks % cd ../../  
[sainiharikagundu@sais-MacBook-Air git-practice % git commit -m "update file"  
Commit rejected!  
Your commit message MUST start with: JIRA-XXX  
[sainiharikagundu@sais-MacBook-Air git-practice % git commit -m "added file"  
Commit rejected!  
Your commit message MUST start with: JIRA-XXX  
sainiharikagundu@sais-MacBook-Air git-practice %
```

14. Recover Lost Commit

- Commit something, reset hard, and then recover it using git reflog.

>> Here, we have created a file, added it and committed it

```
[sainiharikagundu@sais-MacBook-Air git-practice % touch reset.txt  
[sainiharikagundu@sais-MacBook-Air git-practice % git add .
```

```
[sainiharikagundu@sais-MacBook-Air git-practice % git commit -m " added reset.txt"  
hint: The '.git/hooks/pre-commit' hook was ignored because it's not set as executable.  
hint: You can disable this warning with `git config set advice.ignoredHook false'.  
[detached HEAD b98b0b2] added reset.txt  
 2 files changed, 1 insertion(+)  
  create mode 160000 git-practice  
  create mode 100644 reset.txt  
sainiharikagundu@sais-MacBook-Air git-practice %
```

>> git reset --hard HEAD~1 will remove the head commit

```
[sainiharikagundu@sais-MacBook-Air git-practice % git log --oneline -3  
b98b0b2 (HEAD) added reset.txt  
47028fa (main, fix-abc) Revert "added abc"  
336fe68 added abc  
[sainiharikagundu@sais-MacBook-Air git-practice % git reset --hard HEAD~1  
warning: unable to rmdir 'git-practice': Directory not empty  
HEAD is now at 47028fa Revert "added abc"  
[sainiharikagundu@sais-MacBook-Air git-practice % git log --oneline  
fatal: unrecognized argument: --online  
[sainiharikagundu@sais-MacBook-Air git-practice % git log --oneline  
47028fa (HEAD, main, fix-abc) Revert "added abc"  
336fe68 added abc  
02d78c5 (origin/main, origin/HEAD) modified the file
```

>> git reflog will show us the information record

```
[sainiharikagundu@sais-MacBook-Air git-practice % git reflog -5  
47028fa (HEAD, main, fix-abc) HEAD@{0}: reset: moving to HEAD~1  
b98b0b2 HEAD@{1}: commit: added reset.txt  
47028fa (HEAD, main, fix-abc) HEAD@{2}: checkout: moving from fix-abc to 47028fa  
47028fa (HEAD, main, fix-abc) HEAD@{3}: checkout: moving from 47028fa57f9acc9607072faf86d12d9b9ecee3a9 to fix-abc  
47028fa (HEAD, main, fix-abc) HEAD@{4}: checkout: moving from main to 47028fa  
sainiharikagundu@sais-MacBook-Air git-practice %
```

>> git checkout b98b0b2, we can confirm that by seeing git reflog the commit b98b0b2 has been recovered

```
[sainiharikagundu@sais-MacBook-Air git-practice % git checkout b98b0b2  
M git-practice  
Previous HEAD position was 47028fa Revert "added abc"  
HEAD is now at b98b0b2 added reset.txt  
[sainiharikagundu@sais-MacBook-Air git-practice % git log --oneline -5  
b98b0b2 (HEAD) added reset.txt  
47028fa (main, fix-abc) Revert "added abc"  
336fe68 added abc  
02d78c5 (origin/main, origin/HEAD) modified the file  
8382efc added content in techie1  
sainiharikagundu@sais-MacBook-Air git-practice %
```

13. Fork & Pull Request Workflow

- Fork a repo, make a change, and submit a pull request to the original repo.

