

try to create master branch

in master branch create a file biggest3.c

in master branch create a file fact.c and give tag name

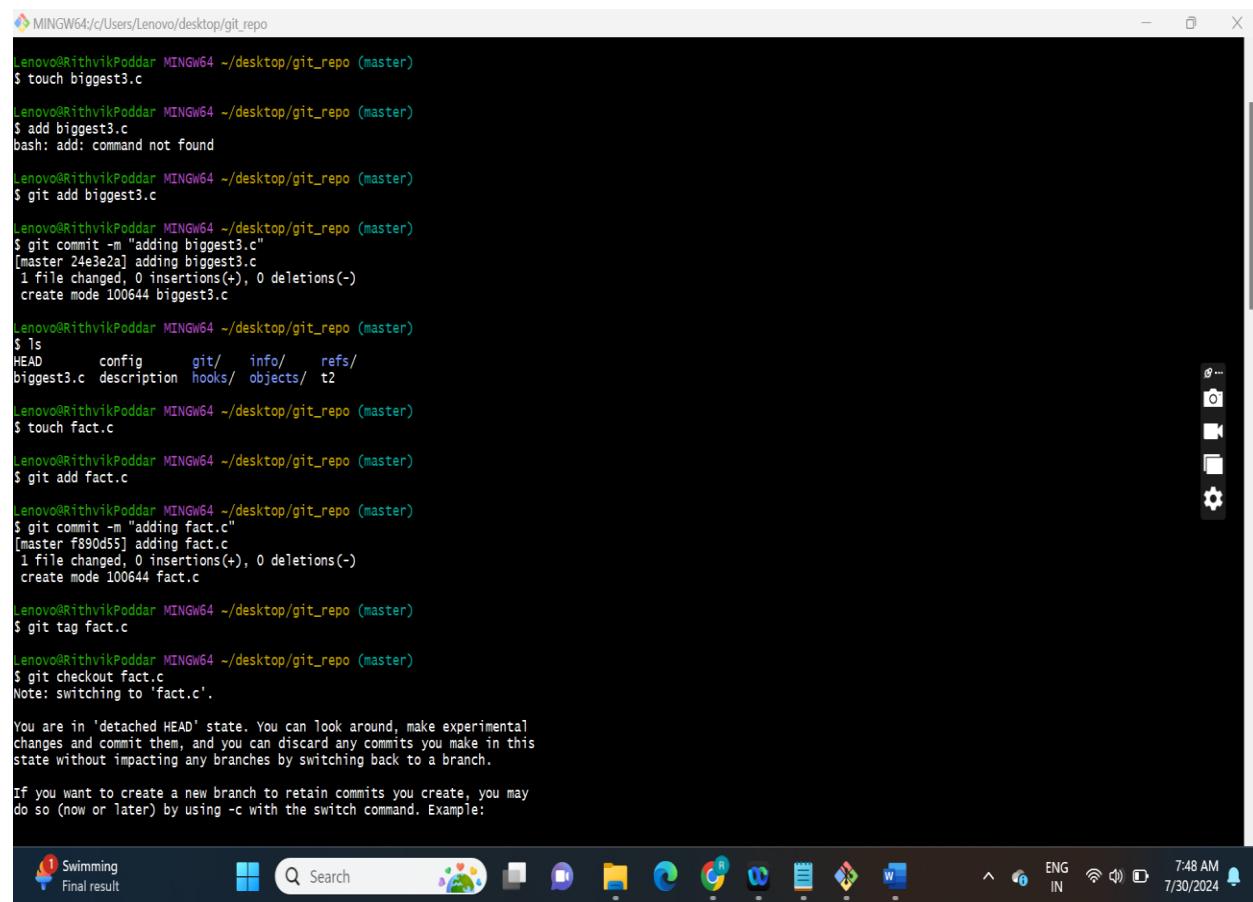
under master you create one more brach --> branch 1

in branch 1 create a file with code of reverse.c

again create a branch 2 in master branch

in branch 2 create a file fibbonacci.called

merge branch 1 and branch 2 in master branch



```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ touch biggest3.c

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ add biggest3.c
bash: add: command not found

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git add biggest3.c

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git commit -m "adding biggest3.c"
[master 24e3e2a] adding biggest3.c
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 biggest3.c

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ ls
HEAD config git/ info/ refs/
biggest3.c description hooks/ objects/ t2

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ touch fact.c

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git add fact.c

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git commit -m "adding fact.c"
[master f890d55] adding fact.c
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fact.c

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git tag fact.c

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git checkout fact.c
Note: switching to 'fact.c'.

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:

 1 Swimming
 Final result
```

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git checkout fact.c
Note: switching to 'fact.c'.
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 f890d55 adding fact.c

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((fact.c))
$ git branch
* (HEAD detached at fact.c)
  branch1
  branch2
  master

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((fact.c))
$ ls
HEAD  config  fact.c  hooks/  objects/  t2
biggest3.c  description  git/  info/  refs/
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((fact.c))
$ git checkout -b branch1
fatal: a branch named 'branch1' already exists

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((fact.c))
$ git checkout master
Switched to branch 'master'

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git checkout branch1
Switched to branch 'branch1'

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ touch reverse.c


```

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ touch reverse.c

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git add reverse.c
bash: git: command not found

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git add reverse.c

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git commit -m "adding reverse.c"
[branch1 2483d89] adding reverse.c
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 reverse.c

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ ls
HEAD  description  hooks/  objects/  reverse.c  t2
config  git/  info/  refs/  t1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git checkout master
Switched to branch 'master'

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git checkout -b branch2
fatal: a branch named 'branch2' already exists

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git checkout -b branch2
fatal: a branch named 'branch2' already exists

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git branch
  branch1
  branch2
* master

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git checkout branch2
Switched to branch 'branch2'

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)


```

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git branch
  branch1
  branch2
* master

lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git checkout branch2
Switched to branch 'branch2'

lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
$ touch fibonacci
$ touch fact.c

lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
$ git add fibonacci
$ git commit -m "adding fibonacci"
[branch2 3c6f758] adding fibonacci
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 fibonacci

lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
$ git checkout master
Switched to branch 'master'

lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git merge branch1
Merge made by the 'ort' strategy.
 reverse.c | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 reverse.c

lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git merge branch2
Merge made by the 'ort' strategy.
 fibonacci | 0
 t4 | 0
 t5 | 0
 y1 | 0
 4 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 fibonacci
 create mode 100644 t4
 create mode 100644 t5
 create mode 100644 y1

lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ |
```

```
MINGW64:/c/Users/lenovo/Desktop/git_repo
lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git log
commit 5916af4f29281149f7ee3e47fbfcdf6f42c32ff (HEAD -> master)
Merge: fe661a9 3c6f758
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:44:46 2024 +0530

    Merge branch 'branch2'

commit fe661a99513070f41cccd046af4755ff6acabb6f69
Merge: f890d55 2483d89
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:44:14 2024 +0530

    Merge branch 'branch1'

commit 3c6f7586825ea51481d46a2109638092e3c4115 (branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:43:00 2024 +0530

    adding fibonacci

commit 2483d895a04c2d78ed7a3179b8d4b40c571662 (branch1)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:36:44 2024 +0530

    adding reverse.c

commit f890d55605c5c89f6d2bca09d97bb9f24c11032d (tag: fact.c)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:29:58 2024 +0530

    adding fact.c

commit 24e3ea0d7ad557526160b5518e5fd79cad400bf
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:28:32 2024 +0530

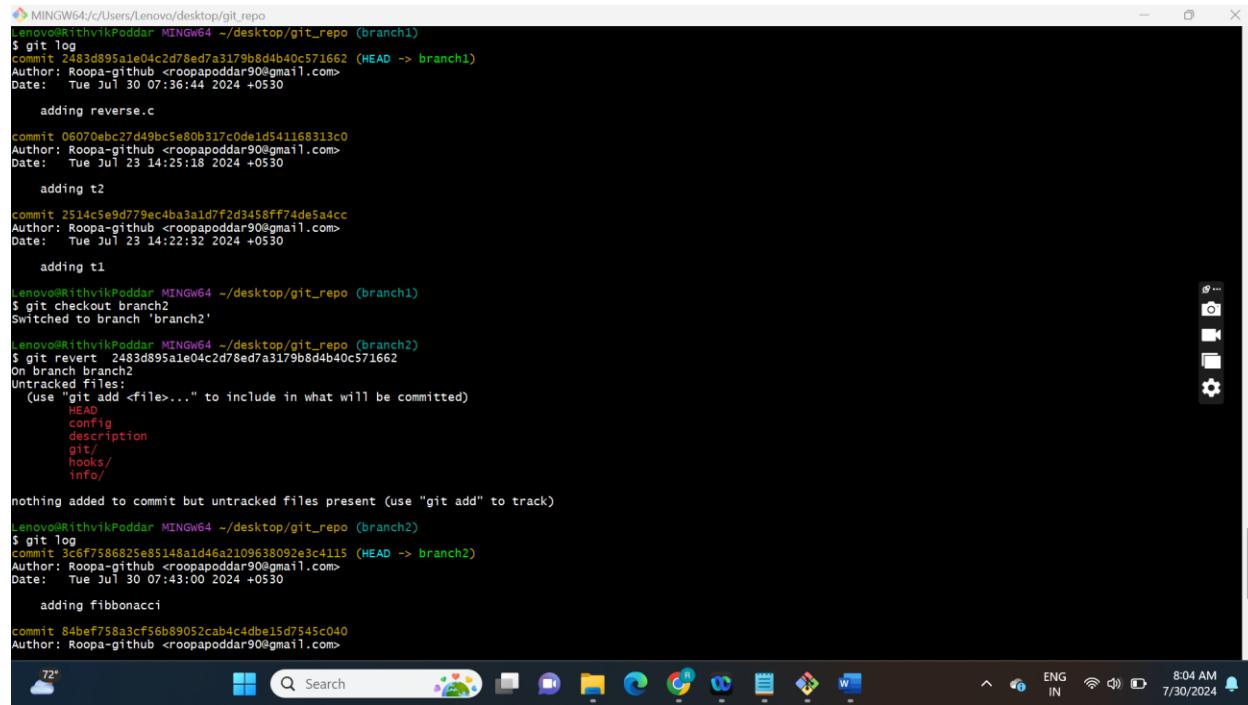
    adding biggest3.c

commit 5e2481a2b34812b83e564ccb3d295469b8f96fa
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Fri Jul 26 15:09:01 2024 +0530

    Revert "adding t1"

This reverts commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc.
```

git revert:



```
Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch1)
$ git log
commit 2483d895a1e04c2d78ed7a3179b8d4b40c571662 (HEAD -> branch1)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:36:44 2024 +0530

    adding reverse.c

commit 06070ebc27d49bc5e80b317c0de1d541168313c0
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:25:18 2024 +0530

    adding t2

commit 2514c5e9d779ec4ba3a1df2d3458ff74de5a4cc
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:22:32 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch1)
$ git checkout branch2
Switched to branch 'branch2'

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch2)
$ git revert 2483d895a1e04c2d78ed7a3179b8d4b40c571662
On branch branch2
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    HEAD
    config
    description
    git/
    hooks/
    info/

nothing added to commit but untracked files present (use "git add" to track)

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch2)
$ git log
commit 3c6f7586825e85148a1d46a2109638092e3c4115 (HEAD -> branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:43:00 2024 +0530

    adding fibonacci

commit 84bef758a3cf56b89052cab4c4dbe15d7545c040
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:43:00 2024 +0530

    adding fibonacci

commit 84bef758a3cf56b89052cab4c4dbe15d7545c040
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Fri Jul 26 12:59:40 2024 +0530

    adding t5

commit e07cfea673298df1bd32a7f7f765f206158f693e
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Fri Jul 26 12:59:09 2024 +0530

    adding t4

commit b8c9245b07d30aaafc10b03296b2dda474cae8dff
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:28:00 2024 +0530

    adding y1

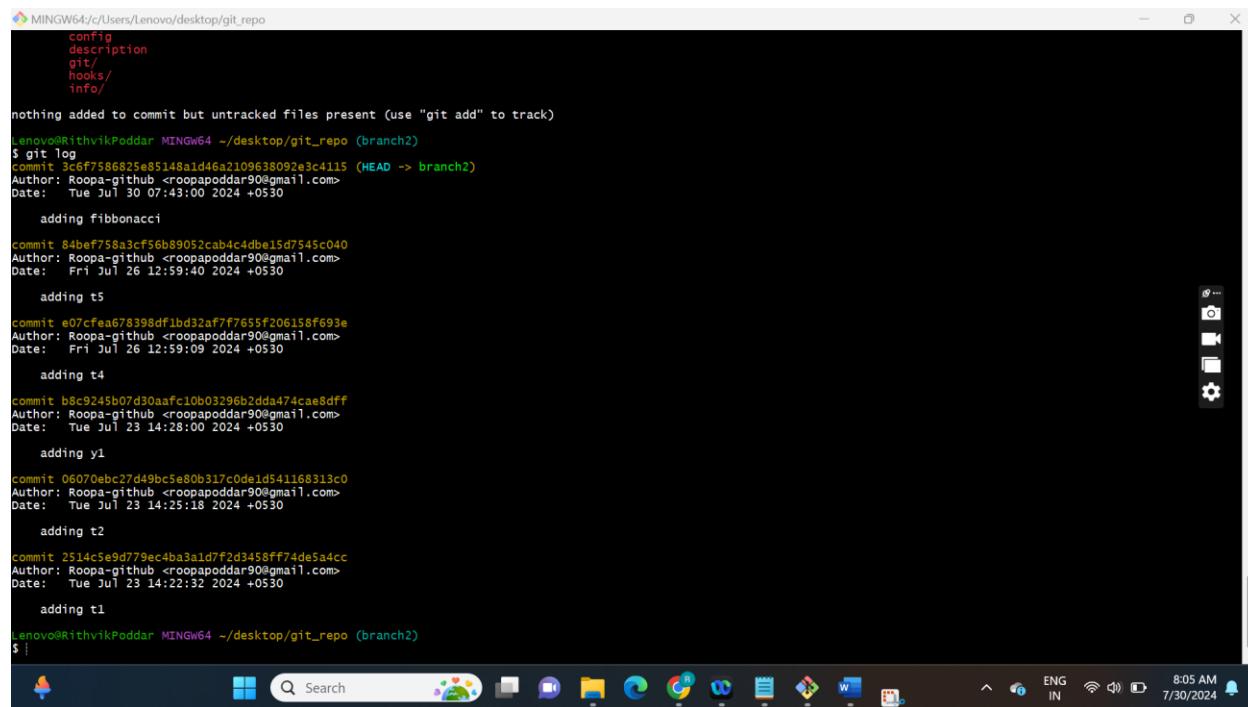
commit 06070ebc27d49bc5e80b317c0de1d541168313c0
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:25:18 2024 +0530

    adding t2

commit 2514c5e9d779ec4ba3a1df2d3458ff74de5a4cc
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:22:32 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch2)
$
```



```
Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo
config
description
git/
hooks/
info/

nothing added to commit but untracked files present (use "git add" to track)

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch2)
$ git log
commit 3c6f7586825e85148a1d46a2109638092e3c4115 (HEAD -> branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:43:00 2024 +0530

    adding fibonacci

commit 84bef758a3cf56b89052cab4c4dbe15d7545c040
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Fri Jul 26 12:59:40 2024 +0530

    adding fibonacci

commit 84bef758a3cf56b89052cab4c4dbe15d7545c040
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Fri Jul 26 12:59:09 2024 +0530

    adding t5

commit e07cfea673298df1bd32a7f7f765f206158f693e
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Fri Jul 26 12:59:09 2024 +0530

    adding t4

commit b8c9245b07d30aaafc10b03296b2dda474cae8dff
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:28:00 2024 +0530

    adding y1

commit 06070ebc27d49bc5e80b317c0de1d541168313c0
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:25:18 2024 +0530

    adding t2

commit 2514c5e9d779ec4ba3a1df2d3458ff74de5a4cc
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:22:32 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch2)
$
```

```
MINGW64:/c/Users/Lenovo/desktop/git_repo
config
description
git/
hooks/
info/
nothing added to commit but untracked files present (use "git add" to track)

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch2)
$ git log
commit 3c6f7586825e85148a1d46a2109638092e3c4115 (HEAD -> branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:43:00 2024 +0530

    adding fibonacci

commit 84be0758a3cF5b89052cab4c4dbe15d7545c040
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Fri Jul 26 12:59:40 2024 +0530

    adding t5

commit e07cfea678398df1bd32af7f7655f206158f693e
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Fri Jul 26 12:59:09 2024 +0530

    adding t4

commit b8c9245b07d30aaafc10b03296b2ddaa474cae8dff
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:28:00 2024 +0530

    adding y1

commit 06070ebc27d49bc5e80b317c0de1d541168313c0
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:25:18 2024 +0530

    adding t2

commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:22:32 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch2)
$ git reset --mixed HEAD~2
```

Git reset:

```
git reset --soft HEAD~2
```

```
MINGW64:/c/Users/Lenovo/desktop/git_repo
Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch2)
$ git reset --mixed HEAD~2

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch2)
$ git log
commit e07cfea678398df1bd32af7f7655f206158f693e (HEAD -> branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Fri Jul 26 12:59:09 2024 +0530

    adding t5

commit b8c9245b07d30aaafc10b03296b2ddaa474cae8dff
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:28:00 2024 +0530

    adding y1

commit 06070ebc27d49bc5e80b317c0de1d541168313c0
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:25:18 2024 +0530

    adding t2

commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:22:32 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch2)
$ git status
On branch branch2
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    HEAD
    config
    description
    fibonacci
    git/
    hooks/
    info/
    t5

nothing added to commit but untracked files present (use "git add" to track)

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch2)
$ !
```

git reset --soft HEAD~1

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
hooks/
  info/
  t5

nothing added to commit but untracked files present (use "git add" to track)
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
$ 

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
$ git reset --soft HEAD~2
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
$ git status
On branch branch2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  t4
    new file:  y1

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    HEAD
    config
    description
    fibonacci
    git/
    hooks/
    info/
    t5

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
$ git log
commit 06070ebc27d49bc5e80b317c0de1d541168313c0 (HEAD -> branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 23 14:25:18 2024 +0530

    adding t2

commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 23 14:22:32 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
$ |
```



git reset --hard HEAD~1

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 t2
create mode 100644 t4
create mode 100644 y1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
$ git log
commit 0a01b74c406b9c507acc6dd97e38814016f2069e (HEAD -> branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 30 08:12:47 2024 +0530

    adding all files

commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 23 14:22:32 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
$ git reset --hard HEAD-1
HEAD is now at 2514c5e adding t1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
$ git log
commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc (HEAD -> branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 23 14:22:32 2024 +0530

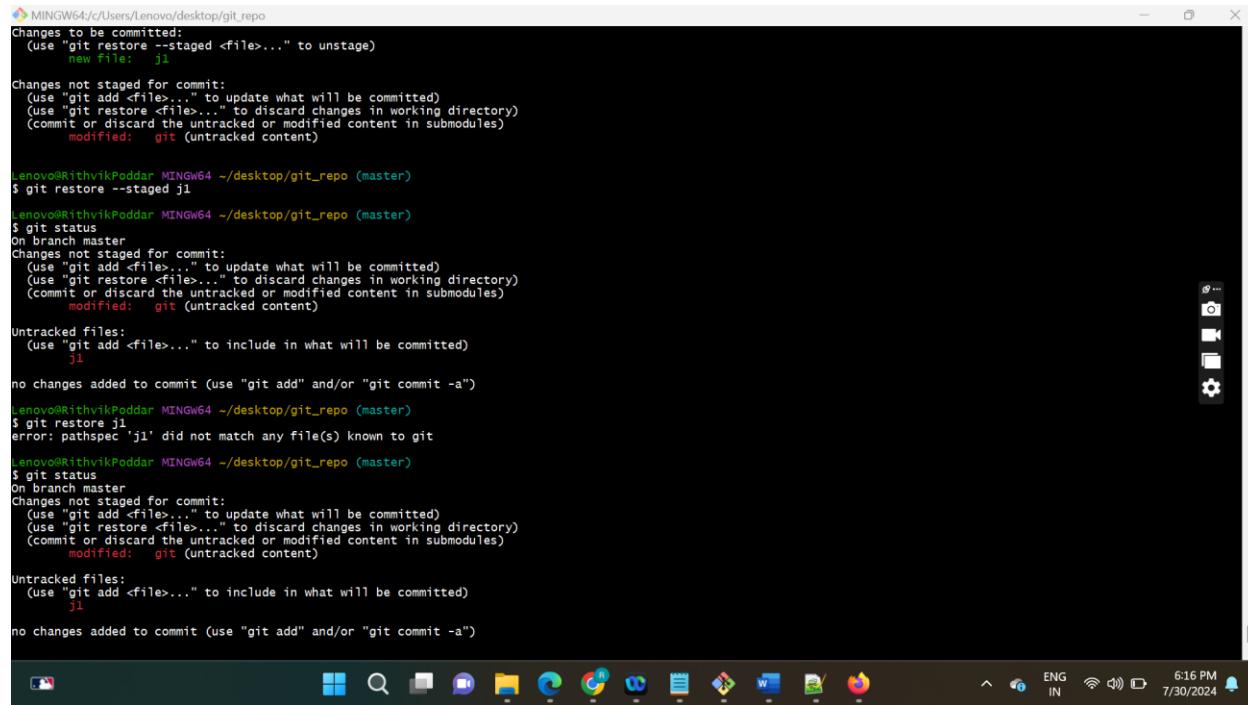
    adding t1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
$ git status
On branch branch2
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    HEAD
    config
    description
    fibonacci
    git/
    hooks/
    info/
    t5

nothing added to commit but untracked files present (use "git add" to track)
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch2)
```



git restore:-



```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   j1

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    (commit or discard the untracked or modified content in submodules)
      modified:   git (untracked content)

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git restore --staged j1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    (commit or discard the untracked or modified content in submodules)
      modified:   git (untracked content)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    j1

no changes added to commit (use "git add" and/or "git commit -a")

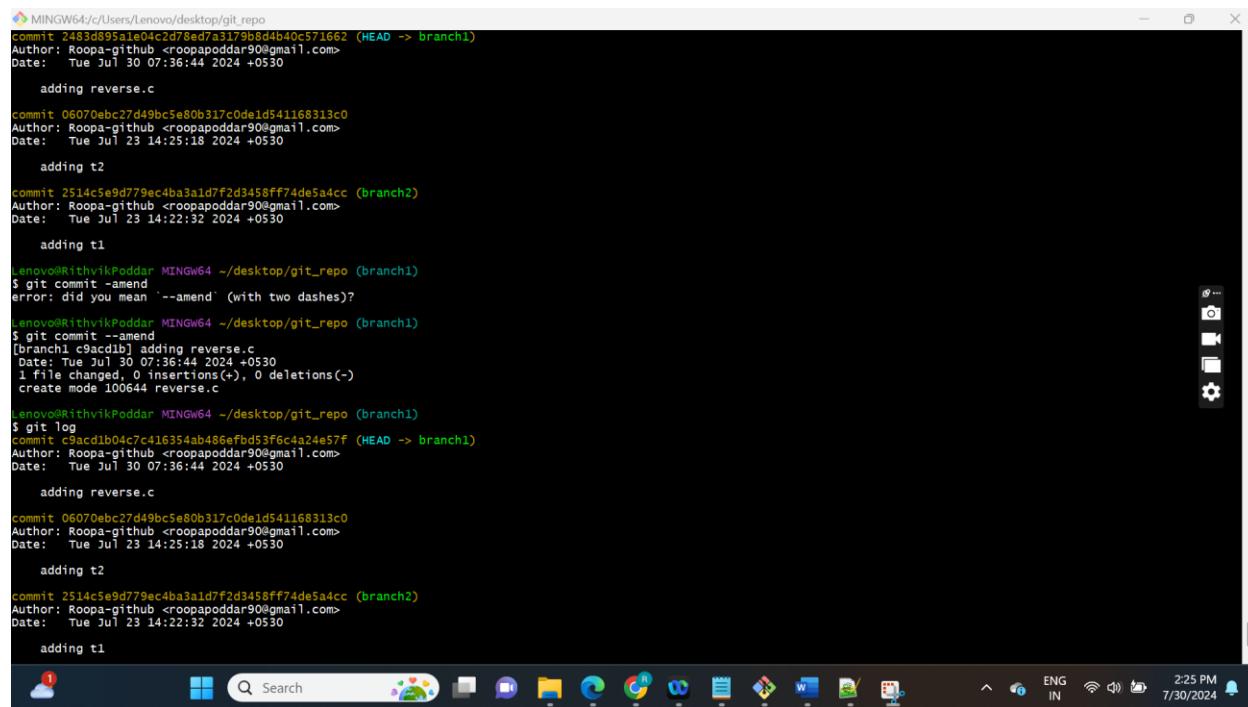
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git restore j1
error: pathspec 'j1' did not match any file(s) known to git

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    (commit or discard the untracked or modified content in submodules)
      modified:   git (untracked content)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    j1

no changes added to commit (use "git add" and/or "git commit -a")
```

Git amend:--



```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
commit 2483d895a1e04c2d78ed7a3179b8d4b40c571662 (HEAD -> branch1)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 07:36:44 2024 +0530

  adding reverse.c

commit 06070ebc27d49bc5e80b317c0de1d541168313c0
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 23 14:25:18 2024 +0530

  adding t2

commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc (branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 23 14:22:32 2024 +0530

  adding t1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git commit --amend
error: did you mean '--amend' (with two dashes)?

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git commit --amend
[branch1 c9acd1b] adding reverse.c
Date: Tue Jul 30 07:36:44 2024 +0530
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 reverse.c

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git log
commit c9acd1b04c7c416354ab486efbd53f6c4a24e57f (HEAD -> branch1)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 07:36:44 2024 +0530

  adding reverse.c

commit 06070ebc27d49bc5e80b317c0de1d541168313c0
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 23 14:25:18 2024 +0530

  adding t2

commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc (branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 23 14:22:32 2024 +0530

  adding t1
```

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
adding reverse.c

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Tue Jul 30 07:36:44 2024 +0530
#
# On branch branch1
# Changes to be committed:
#   new file:   reverse.c
#
# Untracked files:
#   HEAD
#   config
#   description
#   git/
#   hooks/
#   info/
#
# .git/COMMIT_EDITMSG [unix] (14:22 30/07/2024)
# ~/Desktop/git_repo/.git/COMMIT_EDITMSG [unix] 19L, 349B
1,1 All


```

Git ignore:-

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
.css
.txt
.java

# .gitignore [unix] (14:48 30/07/2024)
# 3,6 All


```

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo

commit 2514c5e9d779ec4ba3a1df72d3458ff74de5a4cc (branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:22:32 2024 +0530

adding t1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git status
On branch branch1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
      HEAD
      config
      description
      git/
      hooks/
      info/
nothing added to commit but untracked files present (use "git add" to track)

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git add .gitignore
warning: in the working copy of '.gitignore', LF will be replaced by CRLF the next time Git touches it

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git commit -m "adding .gitignore"
[branch1 eb89f24] adding .gitignore
 1 file changed, 3 insertions(+)
 create mode 100644 .gitignore

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ cat .gitignore
*.css
*.txt
*.java

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git log --grep "ignore"
commit eb89f24e06015ee23143db6ae49940f50e573976 (HEAD -> branch1)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 15:57:01 2024 +0530

  adding .gitignore

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ 
```

Git diff:

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git diff branch1 branch2
diff --git a/.gitignore b/.gitignore
deleted file mode 100644
index d297cca..0000000
--- a/.gitignore
+++ /dev/null
@@ -1,3 +0,0 @@
-*.css
-*.txt
-*.java
diff --git a/reverse.c b/reverse.c
deleted file mode 100644
index e69de29..0000000
diff --git a/t2/b/t2
deleted file mode 100644
index e69de29..0000000

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git master branch2
git: 'master' is not a git command. See 'git --help'.

The most similar command is
  restore

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git diff HEAD
fatal: ambiguous argument 'HEAD': both revision and filename
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git diff --staged

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git diff

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git log
commit eb89f24e06015ee23143db6ae49940f50e573976 (HEAD -> branch1)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 15:57:01 2024 +0530

  adding .gitignore

commit c9acd1b04c7c416354ab486efbd53f6c4a24e57f
Author: Roopa-github <roopapoddar90@gmail.com>


```

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
Date: Tue Jul 30 15:57:01 2024 +0530
    adding .gitignore

commit c9acd1b04c7c416354ab486efbd53f6c4a24e57f
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:36:44 2024 +0530

    adding reverse.c

commit 06070ebc27d49bc5e80b317c0de1d541168313c0
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:25:18 2024 +0530

    adding t2

commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc (branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 23 14:22:32 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git diff eb9f24e06015ee23143db6ae49940f50e573976 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc
diff --git a/.gitignore b/.gitignore
deleted file mode 100644
index d297cca..0000000
-- a/.gitignore
+++ /dev/null
@@ -1 +0,0 @@
-  
CSS
-- txt
-- java
diff --git a/reverse.c b/reverse.c
deleted file mode 100644
index e69de29..0000000
diff --git a/t2 b/t2
deleted file mode 100644
index e69de29..0000000
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git diff c9acd1b04c7c416354ab486efbd53f6c4a24e57f 06070ebc27d49bc5e80b317c0de1d541168313c0
diff --git a/reverse.c b/reverse.c
deleted file mode 100644
Index e69de29..0000000
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ |
```

Git bisect:

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1|BISECTING)
$ git bisect good <good_commit>
bash: syntax error near unexpected token `newline'

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1|BISECTING)
$ git bisect good
eb9f24e06015ee23143db6ae49940f50e573976 was both good and bad

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1|BISECTING)
$ git bisect bad
eb9f24e06015ee23143db6ae49940f50e573976 was both good and bad

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1|BISECTING)
$ git bisect reset
Already on 'branch1'

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git bisect good commit
You need to start by "git bisect start"

Do you want me to do it for you [Y/n]? y
error: Bad rev input: commit
status: waiting for both good and bad commits

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1|BISECTING)
$ git bisect reset
Already on 'branch1'

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git bisect good
You need to start by "git bisect start"

Do you want me to do it for you [Y/n]? y
status: waiting for both good and bad commits
status: waiting for bad commit, 1 good commit known

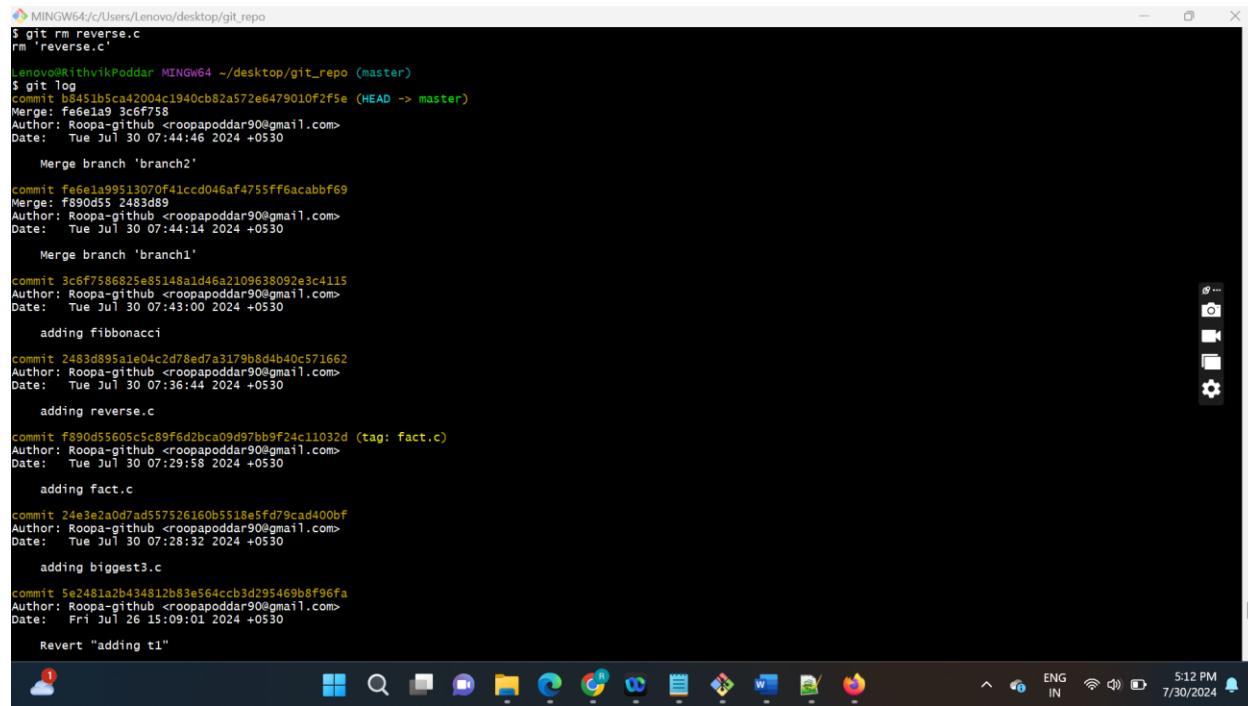
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1|BISECTING)
$ git bisect start
Already on 'branch1'
status: waiting for both good and bad commits

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1|BISECTING)
$ git bisect bad
status: waiting for good commit(s), bad commit known

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1|BISECTING)
$ |
```

Git instaweb

Git drop: git reset, git checkout, git revert, git clean, git branch -d, and git rm.



```
$ git rm reverse.c
$ git log
commit b8451b5ca42004c1940cb82a572e6479010f2f5e (HEAD -> master)
Merge: fe661a9 3c6f758
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 30 07:44:46 2024 +0530

Merge branch 'branch2'

commit fe661a99513070f41cccd046af4755ff6acabff69
Merge: f890d55 2483d89
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 30 07:44:14 2024 +0530

Merge branch 'branch1'

commit 3c6f7586625e85148a1d46a2109638092e3c4115
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 30 07:43:00 2024 +0530

    adding fibonacci

commit 2483d895a1e04c2d78ed7a3179b8d4b40c571662
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 30 07:36:44 2024 +0530

    adding reverse.c

commit f890d55605c5c89f6d2bca09d97bb9f24c11032d (tag: fact.c)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 30 07:29:58 2024 +0530

    adding fact.c

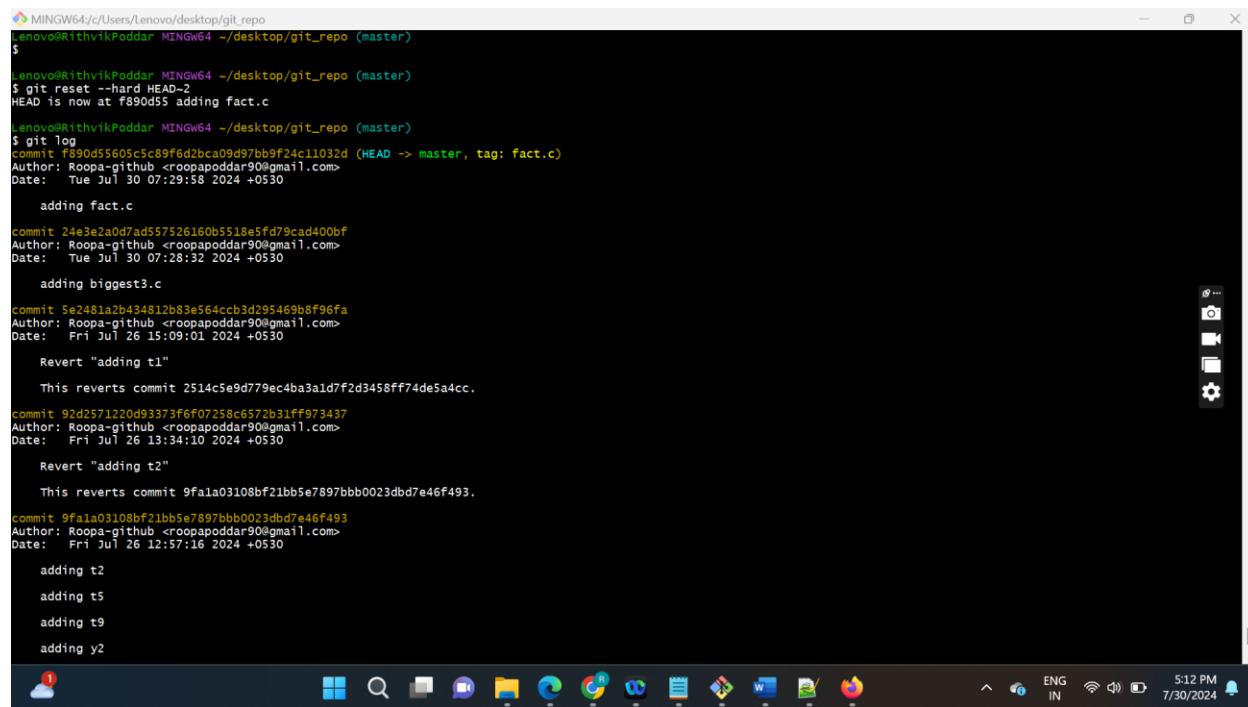
commit 24e3e2a0d7ad57526160b5518e5fd79cad400bf
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 30 07:28:32 2024 +0530

    adding biggest3.c

commit 5e2481a2b434812b83e564ccb3d295469b8f96fa
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Fri Jul 26 15:09:01 2024 +0530

    Revert "adding t1"


```



```
$ git log
commit f890d55605c5c89f6d2bca09d97bb9f24c11032d (HEAD -> master, tag: fact.c)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 30 07:29:58 2024 +0530

    adding fact.c

commit 24e3e2a0d7ad57526160b5518e5fd79cad400bf
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Tue Jul 30 07:28:32 2024 +0530

    adding biggest3.c

commit 5e2481a2b434812b83e564ccb3d295469b8f96fa
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Fri Jul 26 15:09:01 2024 +0530

    Revert "adding t1"

This reverts commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc.

commit 92d2571220d93373f6f0725c6572b31ff973437
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Fri Jul 26 13:34:10 2024 +0530

    Revert "adding t2"

This reverts commit 9fa1a03108bf21bb5e7897bbb0023dbd7e46f493.

commit 9fa1a03108bf21bb5e7897bbb0023dbd7e46f493
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Fri Jul 26 12:57:16 2024 +0530

    adding t2

    adding t5

    adding t9

    adding y2


```

Amazon EMR [Amazon Elastic MapReduce]:----

Amazon EMR (Elastic MapReduce) is a cloud-based big data platform provided by Amazon Web Services (AWS) that allows you to process and analyze large data sets using open-source tools such as Apache Hadoop, Apache Spark, Apache HBase, Apache Flink, Apache Hudi, and Presto. EMR simplifies running big data frameworks and allows you to focus on analyzing data without worrying about the underlying infrastructure.

Key Features of Amazon EMR

1. **Scalability:** Automatically scales the number of nodes up or down based on your processing needs.
2. **Cost-Effectiveness:** Pay only for the resources you use. You can also use spot instances to reduce costs.
3. **Integration:** Seamlessly integrates with other AWS services like S3, DynamoDB, RDS, and Redshift.
4. **Flexibility:** Supports multiple big data frameworks and can run custom applications.
5. **Managed Service:** AWS handles the setup, configuration, and management of the infrastructure.

Use Cases

1. **Data Processing:** Perform large-scale data processing tasks such as ETL (Extract, Transform, Load) jobs.
2. **Machine Learning:** Run machine learning algorithms on large data sets using Spark MLlib or other frameworks.
3. **Real-time Data Analysis:** Use frameworks like Apache Flink to analyze streaming data in real-time.
4. **Data Warehousing:** Use Presto for interactive querying of large datasets stored in S3.
5. **Log Analysis:** Analyze log files to gain insights and monitor system performance.

Types of AMI :

AMI is divided into two categories:

1. EBS - backed Instances
2. Instance Store - backed Instances.

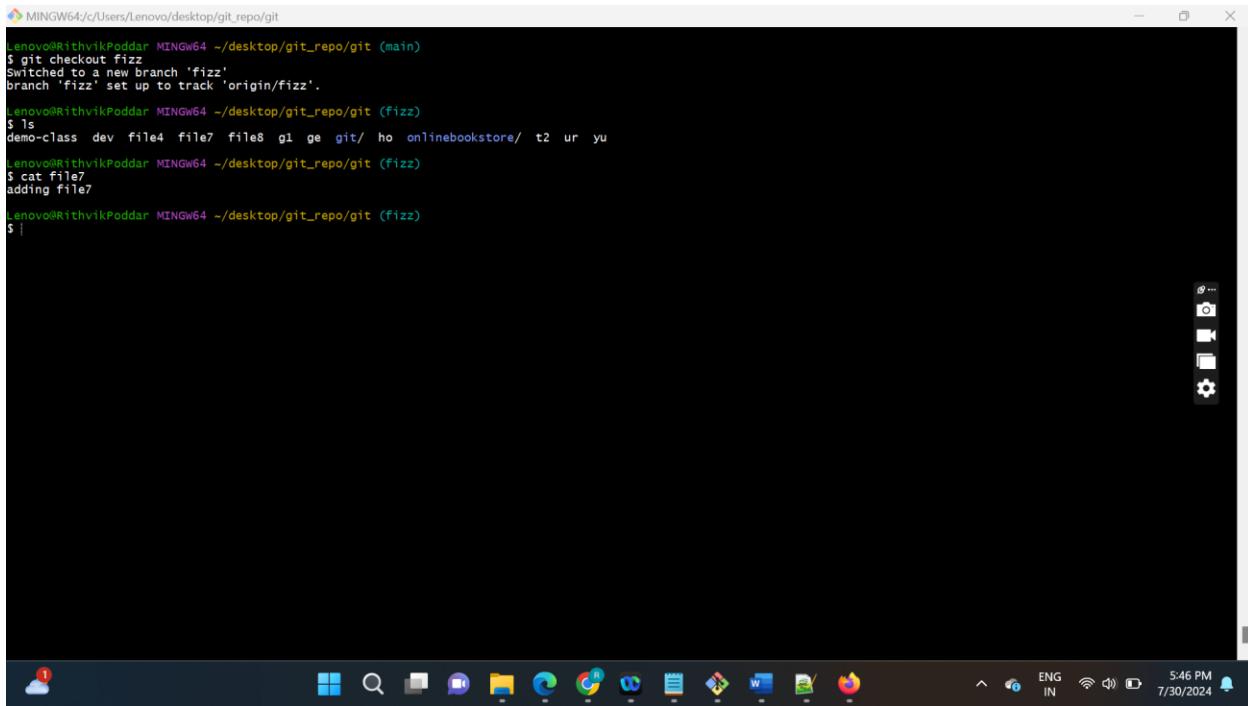
raise a pull request and merge to the master branch from git hub

This screenshot shows the GitHub repository page for 'Roopa-github/git'. The repository name is 'git' and it is private. The 'fizz' branch is selected. A tooltip is open over the 'Code' button, showing options like 'Create new file' and 'Upload files'. The repository has 8 branches and 0 tags. The last commit was a merge pull request from 'Roopa-github/giga' with 37 commits, dated 731f29f · 4 days ago. The commit message is 'Merge pull request #5 from Roopa-github/giga'. Below the commit list, there are links for 'demo-class', 'dev', 'file4', 'file8', 'g1', and 'ge'. On the right side, there are sections for 'About', 'Activity', 'Releases', and 'Packages'. The status bar at the bottom shows the date as 7/30/2024 and the time as 5:22 PM.

This screenshot shows the 'Commit changes' dialog box on GitHub. The repository is 'Roopa-github/git' and the branch is 'fizz'. The commit message is 'Create file7'. There is an optional extended description field with placeholder text 'Add an optional extended description...'. Two radio button options are available: 'Commit directly to the fizz branch' (selected) and 'Create a new branch for this commit and start a pull request'. At the bottom are 'Cancel' and 'Commit changes' buttons. The status bar at the bottom shows the date as 7/30/2024 and the time as 5:23 PM.

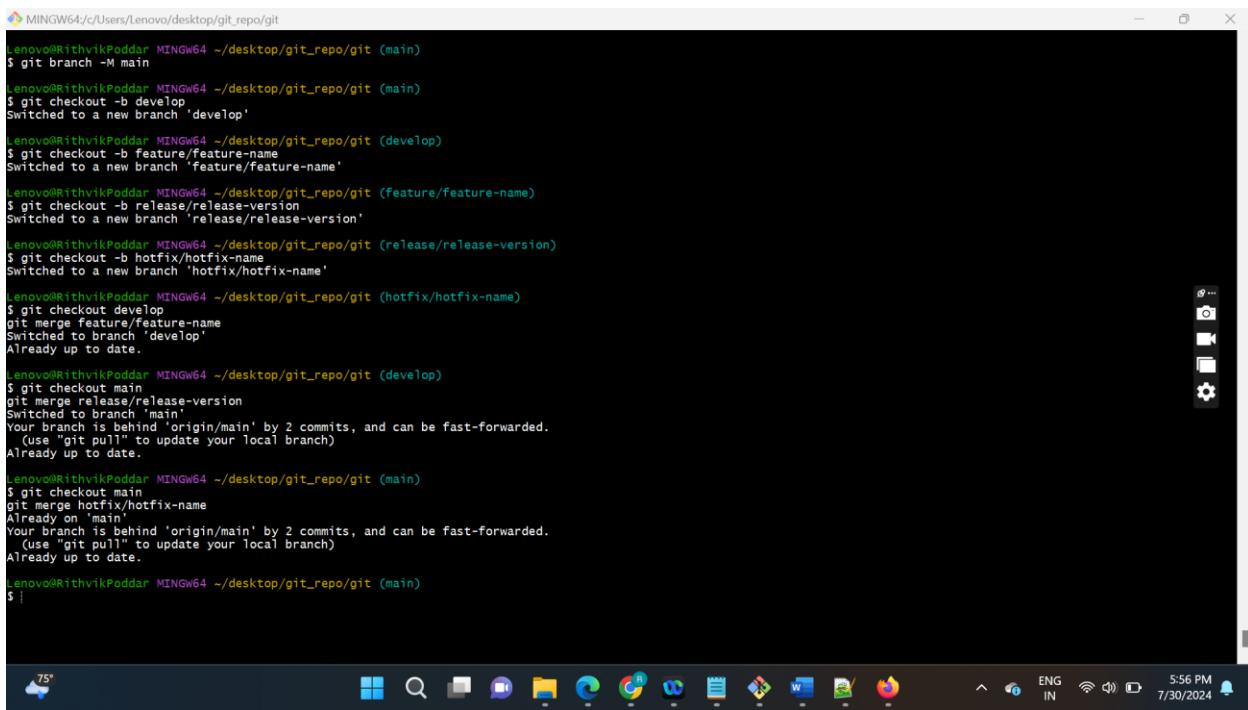
The screenshot shows the GitHub 'Pull requests' page for the repository 'Roopa-github/git'. A yellow banner at the top indicates that 'fizz' has recent pushes less than a minute ago. The main search bar shows 'is:pr is:open'. There is one open pull request from 'gir' (#3), which was opened 5 days ago by Roopa-github. The pull request details are visible below the search bar.

The screenshot shows the GitHub 'Merge pull request' dialog for pull request #7. It displays the message 'Roopa-github wants to merge 1 commit into main from fizz'. Below this, there is a text input field for adding more commits to the 'fizz' branch. On the right side, there are sections for Labels, Projects, Milestone, and Development, all currently set to 'None yet'. At the bottom, there are 'Confirm merge' and 'Cancel' buttons. Below the dialog, there is a 'Add a comment' section with a rich text editor and a note that Markdown is supported. The GitHub interface is shown on a Windows desktop with a taskbar at the bottom.



```
MINGW64:/c/Users/Lenovo/Desktop/git_repo/git
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ git checkout fizz
Switched to a new branch 'fizz'
branch 'fizz' set up to track 'origin/fizz'.
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (fizz)
$ ls
demo-class dev file4 file7 file8 gl ge git/ ho onlinebookstore/ t2 ur yu
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (fizz)
$ cat file7
adding file7
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (fizz)
$ |
```

create a branch strategy in the terminal.



```
MINGW64:/c/Users/Lenovo/Desktop/git_repo/git
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ git branch -M main
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ git checkout -b develop
Switched to a new branch 'develop'
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (develop)
$ git checkout -b feature/feature-name
Switched to a new branch 'feature/feature-name'
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (feature/feature-name)
$ git checkout -b release/release-version
Switched to a new branch 'release/release-version'
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (release/release-version)
$ git checkout -b hotfix/hotfix-name
Switched to a new branch 'hotfix/hotfix-name'
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (hotfix/hotfix-name)
$ git checkout develop
git merge feature/feature-name
Switched to branch 'develop'
Already up to date.

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (develop)
$ git checkout main
git merge release/release-version
Switched to branch 'main'.
Your branch is behind 'origin/main' by 2 commits, and can be fast-forwarded.
 (use "git pull" to update your local branch)
Already up to date.

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ git checkout main
git merge hotfix/hotfix-name
Already on 'main'
Your branch is behind 'origin/main' by 2 commits, and can be fast-forwarded.
 (use "git pull" to update your local branch)
Already up to date.

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ |
```

1) What is Git branching strategy for organization? How it works and Types of branches and its importance.

A Git branching strategy is a set of guidelines for managing branches in a Git repository. Here's

a common strategy used by organizations:

Main Branches:

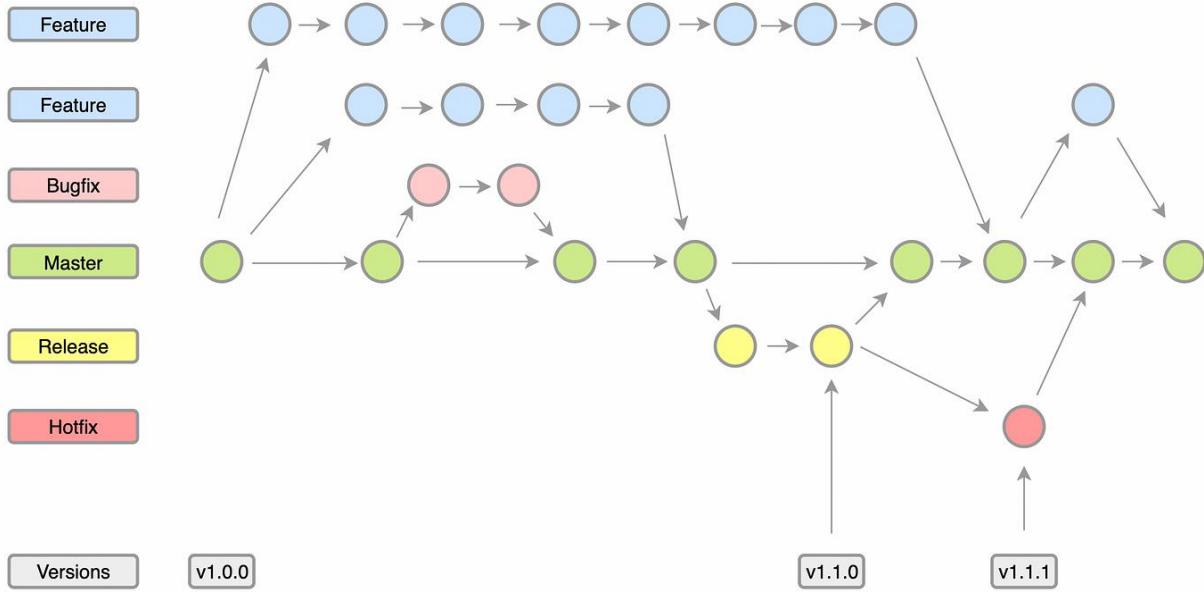
1. Master: The production-ready branch, always stable and deployable.
2. Develop: The main development branch, where new features and fixes are integrated.

Supporting Branches:

1. Feature: Branches for new features, named feature/<feature-name>.
2. Fix: Branches for bug fixes, named fix/<fix-name>.
3. Release: Branches for preparing new releases, named release/<version>.
4. Hotfix: Branches for urgent bug fixes, named hotfix/<fix-name>.

Workflow:

1. Developers create feature or fix branches from develop.
2. They work on their branches, committing changes.
3. Once ready, they merge their branches into develop.
4. Develop is merged into master when ready for production.
5. Release branches are created from master for final testing.
6. Hotfix branches are created from master for urgent fixes



Types of Branches and Its importance:

1. Short-lived branches: feature, fix, and hotfix branches, merged quickly.
2. Long-lived branches: master, develop, and release branches, persisting longer.

Importance:

1. Organization: Clear branch naming and structure.
2. Collaboration: Multiple developers work on separate branches.
3. Stability: master branch always stable, develop branch for integration.
4. Flexibility: Easy to manage multiple features and fixes.
5. Scalability: Supports large teams and complex projects.

2) Difference between a branch and environment.

In Git and software development, a branch and an environment are two distinct concepts:

Branch:

A branch in Git is a separate line of development in a repository. It's a way to diverge from

the main codebase (usually the "master" branch) and work on new features, fixes, or

experiments without affecting the main codebase. Branches are used for:

- Developing new features
- Fixing bugs
- Experimenting with new ideas
- Merging changes from other branches

Environment:

An environment, in the context of software development, refers to a specific setup or configuration for running and testing an application. Environments are used for:

- Development (local machine)
- Testing (staging or QA)
- Production (live deployment)
- Staging (pre-production testing)

Key differences:

- A branch is a version control concept, while an environment is a deployment and testing concept.
- Branches are used for managing code changes, while environments are used for managing application deployments.
- Multiple branches can exist in a single environment, and multiple environments can be used for a single branch.

To illustrate the difference:

- You might have a "feature/new-login-system" branch in your repository, which you work on in your local development environment.
- Once you're ready to test the new login system, you might deploy it to a staging environment for testing.
- After testing, you might merge the branch into the "master" branch and deploy it to the production environment.

- 1.git init : command is used to initialize the respective directory as git repository
- 2.how to check a directory is git repository or not ??
if .git folder is present in the current directory.then it is a git repository
- 3.git status :it will show whether files are in workspace or staging are on in git repo
- 4.git commit -m "message" :this command commits the staged changes to the local repository.

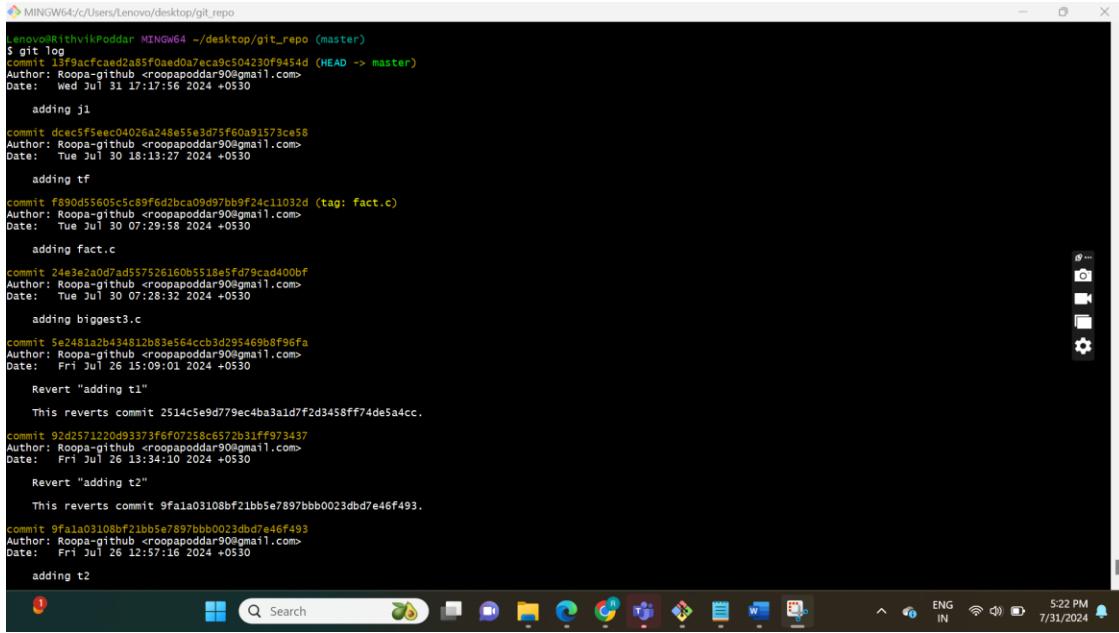
```

MINGW64/c/Users/Lenovo/Desktop/git_repo
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git init
Reinitialized existing git repository in c:/Users/Lenovo/Desktop/git_repo/.git/
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ ls
HEAD      description  hooks/  objects/   t2
biggest3.c fact.c     info/    refs/    tf
config     git/        j1       simple-java-maven-app/
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    (commit or discard the untracked or modified content in submodules)
           modified:   git (untracked content)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
d)         j1
          simple-java-maven-app/
no changes added to commit (use "git add" and/or "git commit -a")
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git add ji
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git commit -m "adding j1"
[master 13f9acfc] adding j1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 j1
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    (commit or discard the untracked or modified content in submodules)
           modified:   git (untracked content)

```

git log : gives history of file or repo.



The screenshot shows a terminal window titled 'MINGW64/c/Users/Lenovo/Desktop/git_repo' running on a Windows 10 desktop. The terminal displays a series of git commits from a repository. The commits are as follows:

- commit d5eac5f5ec04076a248e55e3d75f60a91573ce58
Author: Roopa-github <roopapoddar90@gmail.com>
Date: wed Jul 31 17:56 2024 +0530
adding tf
- commit f890d55605c5c89f6d2bca09d97bb9f24c11032d (tag: fact.c)
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 18:37 2024 +0530
adding fact.c
- commit 2443e2a0d7ad57526160b5518e5fd79cad400bf
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Tue Jul 30 07:28:32 2024 +0530
adding biggest3.c
- commit 5e2481a2b434812b83e564ccb3d295469b8f96fa
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Fri Jul 26 15:09:01 2024 +0530
Revert "adding t1"
This reverts commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc.
- commit 92d2571220d9337f6f07258c6572b31f#f973437
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Fri Jul 26 13:34:10 2024 +0530
Revert "adding t2"
This reverts commit 9fala03108bf21bb5e7897bbb0023dbd7e46f493.
- commit 9f1a03108bf21bb5e7897bbb0023dbd7e46f493
Author: Roopa-github <roopapoddar90@gmail.com>
Date: Fri Jul 26 12:57:16 2024 +0530
adding t2

The terminal window is located at the bottom of the screen, above the taskbar. The taskbar shows various pinned icons and the system tray indicates the date and time as 7/31/2024 at 5:22 PM.

git branch --> to check number of branches. or list all branches

* --> indicates the current branch or indicates we are on that branch

git branch <branchname> --> to create a branch

git checkout <branch name> --> to switch to another branch

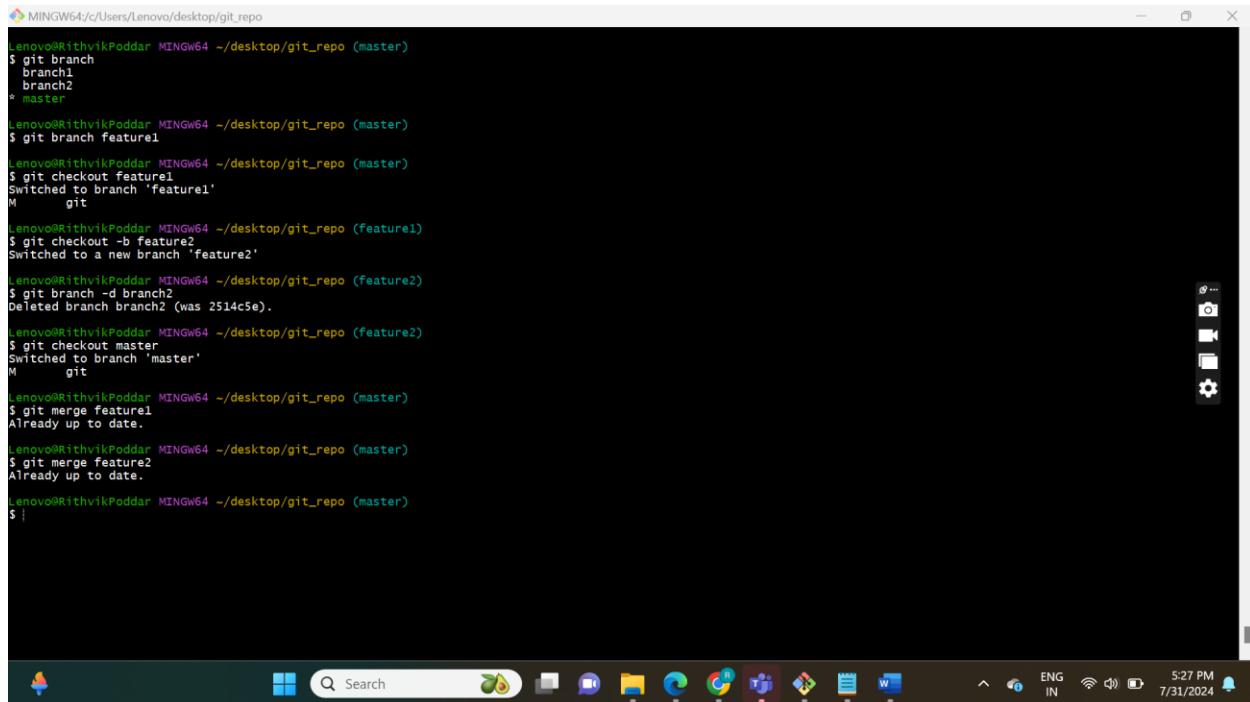
git branch -d branchname -> to delete the branch

git checkout -b branchname -> to create new branch ==> Merging :

git merge <branch name> --> to merge specified branch to checked out branch

merging is only 1 way -- from source to destination.

only 1 master per repository



```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git branch
* master
  branch1
  branch2

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git branch feature1
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git checkout feature1
Switched to branch 'feature1'
M       git

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (feature1)
$ git checkout -b feature2
Switched to a new branch 'feature2'

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (feature2)
$ git branch -d branch2
Deleted branch branch2 (was 2514c5e).

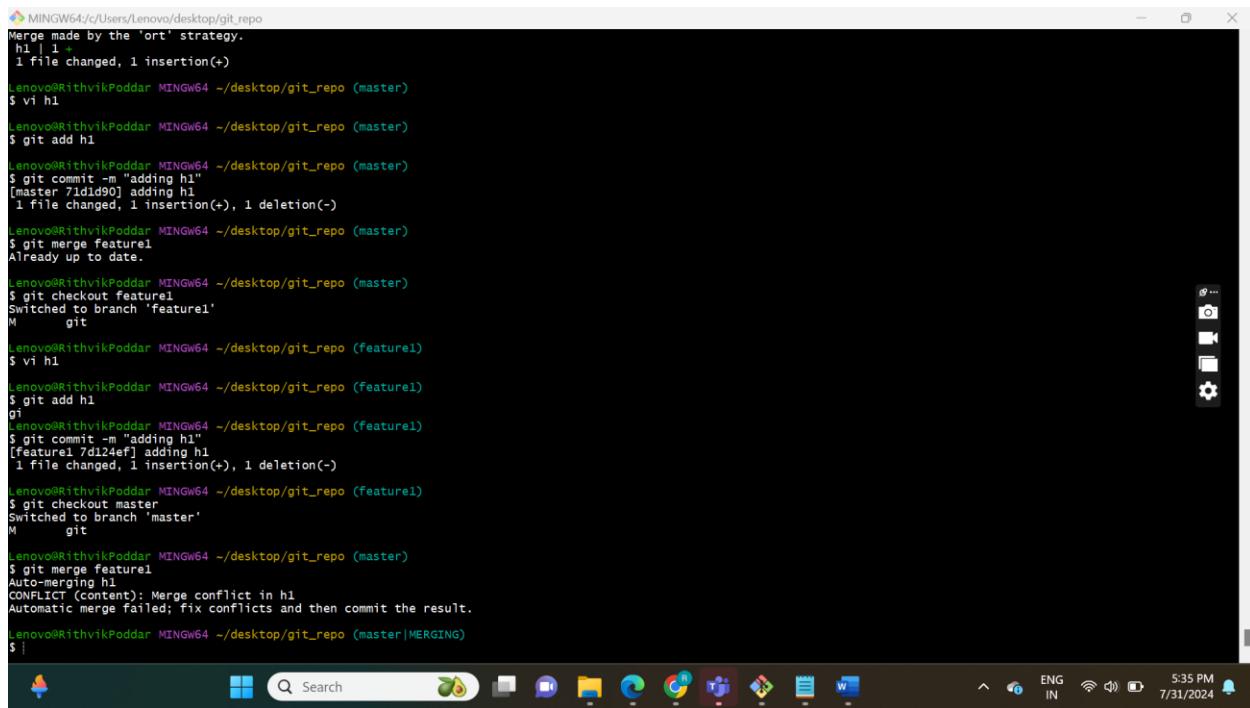
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (feature2)
$ git checkout master
Switched to branch 'master'
M       git

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git merge feature1
Already up to date.

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git merge feature2
Already up to date.

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ |
```

Merge Conflict : occurs when same piece of code is been worked on different branches, and when we try to merge, the conflict occurs we can contact particular developer to modify the changes



```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
Merge made by the 'ort' strategy.
 h1 | 1 +
 1 file changed, 1 insertion(+)

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ vi h1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git add h1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git commit -m "adding h1"
[master 71d1d90] adding h1
 1 file changed, 1 insertion(+), 1 deletion(-)

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git merge feature1
Already up to date.

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git checkout feature1
Switched to branch 'feature1'
M       git

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (feature1)
$ vi h1

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (feature1)
$ git add h1
gi
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (feature1)
$ git commit -m "adding h1"
[feature1 7d124ef] adding h1
 1 file changed, 1 insertion(+), 1 deletion(-)

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (feature1)
$ git checkout master
Switched to branch 'master'
M       git

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git merge feature1
Auto-merging h1
CONFLICT (content): Merge conflict in h1
Automatic merge failed; fix conflicts and then commit the result.

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)|MERGING|
$ |
```

git cherry_pick commit_id --> to merge specific commit on branch(currently checkout branch)

```
MINGW64:/c/Users/lenovo/desktop/git_repo
Switched to branch 'branch1'

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch1)
$ git log
commit eb89f24e06015ee23143db6ae49940f50e573976 (HEAD -> branch1)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 15:57:01 2024 +0530

    adding .gitignore

commit c9accd1b04c7c416354ab486efbd53f6c4a24e57f
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 07:36:44 2024 +0530

    adding reverse.c

commit 06070ebc27d49bc5e80b317c0de1d541168313c0
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 23 14:25:18 2024 +0530

    adding t2

commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 23 14:22:32 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch1)
$ git cherry-pick 24a3e2a0d7ad557526160b5518e5fd79cad400bf
[branch1 5ff6a96] adding biggest3.c
Date: Tue Jul 30 07:28:32 2024 +0530
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 biggest3.c

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch1)
$ git log
commit 5ff6a969e647a6476c80105a3bf764c0bf041990 (HEAD -> branch1)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 07:28:32 2024 +0530

    adding biggest3.c

commit eb89f24e06015ee23143db6ae49940f50e573976
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 15:57:01 2024 +0530

    adding .gitignore

    adding reverse.c

commit 06070ebc27d49bc5e80b317c0de1d541168313c0
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 23 14:25:18 2024 +0530

    adding t2

commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 23 14:22:32 2024 +0530

    adding t1
```

```
MINGW64:/c/Users/lenovo/desktop/git_repo
commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 23 14:22:32 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch1)
$ git cherry-pick 24a3e2a0d7ad557526160b5518e5fd79cad400bf
[branch1 5ff6a96] adding biggest3.c
Date: Tue Jul 30 07:28:32 2024 +0530
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 biggest3.c

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch1)
$ git log
commit 5ff6a969e647a6476c80105a3bf764c0bf041990 (HEAD -> branch1)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 07:28:32 2024 +0530

    adding biggest3.c

commit eb89f24e06015ee23143db6ae49940f50e573976
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 15:57:01 2024 +0530

    adding .gitignore

commit c9accd1b04c7c416354ab486efbd53f6c4a24e57f
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 07:36:44 2024 +0530

    adding reverse.c

commit 06070ebc27d49bc5e80b317c0de1d541168313c0
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 23 14:25:18 2024 +0530

    adding t2

commit 2514c5e9d779ec4ba3a1d7f2d3458ff74de5a4cc
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 23 14:22:32 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (branch1)
$
```

git tag tagname --> to create a tag name

git checkout tagname --> switch to tag

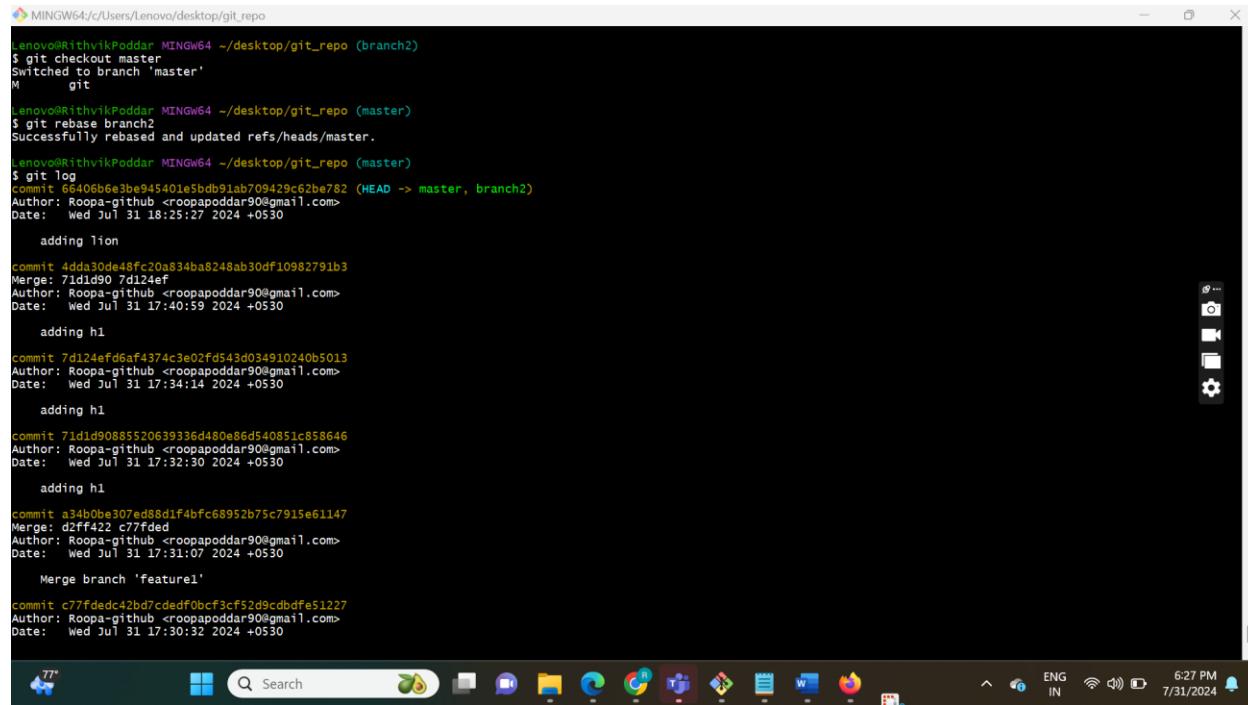
git tag --> to list all tags

git tag -d tagname -> to delete the tag

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
create mode 100644 y2
create mode 100644 y3
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git tag tagname
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (branch1)
$ git checkout tagname
Note: switching to 'tagname'.
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 e18e0e5 adding y1 y2 y3
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((tagname))
$ touch j1 j2 j3 j4
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((tagname))
$ git add j1 j2 j3 j4
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((tagname))
$ git commit -m "adding j1 j2 j3 j4"
[detached HEAD 29e519f] adding j1 j2 j3 j4
4 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 j1
create mode 100644 j2
create mode 100644 j3
create mode 100644 j4
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((29e519f...))
$ cat tagname
cat: tagname: No such file or directory
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((29e519f...))
$ |
```

```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
Or undo this operation with:
  git switch -
Turn off this advice by setting config variable advice.detachedHead to false
HEAD is now at e18e0e5 adding y1 y2 y3
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((tagname))
$ touch j1 j2 j3 j4
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((tagname))
$ git add j1 j2 j3 j4
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((tagname))
$ git commit -m "adding j1 j2 j3 j4"
[detached HEAD 29e519f] adding j1 j2 j3 j4
4 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 j1
create mode 100644 j2
create mode 100644 j3
create mode 100644 j4
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((29e519f...))
$ cat tagname
cat: tagname: No such file or directory
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((29e519f...))
$ git tag -d tagname
Deleted tag 'tagname' (was e18e0e5)
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo ((29e519f...))
$ git checkout master
Warning: you are leaving 1 commit behind, not connected to
any of your branches:
  29e519f adding j1 j2 j3 j4
If you want to keep it by creating a new branch, this may be a good time
to do so with:
  git branch <new-branch-name> 29e519f
Switched to branch 'master'
M       git
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ |
```

Rebase : similar to merge, rewrite the commit history-> git rebase branch name



```
Lenovo|RithvikPoddar MINGW64 ~/desktop/git_repo (branch2)
$ git checkout master
Switched to branch 'master'
M   git

Lenovo|RithvikPoddar MINGW64 ~/desktop/git_repo (master)
$ git rebase branch2
Successfully rebased and updated refs/heads/master.

Lenovo|RithvikPoddar MINGW64 ~/desktop/git_repo (master)
$ git log
commit 66406b6e3be945401e5bdb91ab708429c62be782 (HEAD -> master, branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 18:25:27 2024 +0530

    adding lion

commit 4dd430de48fc20a834ba8248ab30df10982791b3
Merge: 71d1d90 7d124ef
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 17:40:59 2024 +0530

    adding h1

commit 7d124efd6af4374c3e02fd543d034910240b5013
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 17:34:14 2024 +0530

    adding h1

commit 71d1d90885520639336d4808e6d540851c858646
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 17:32:30 2024 +0530

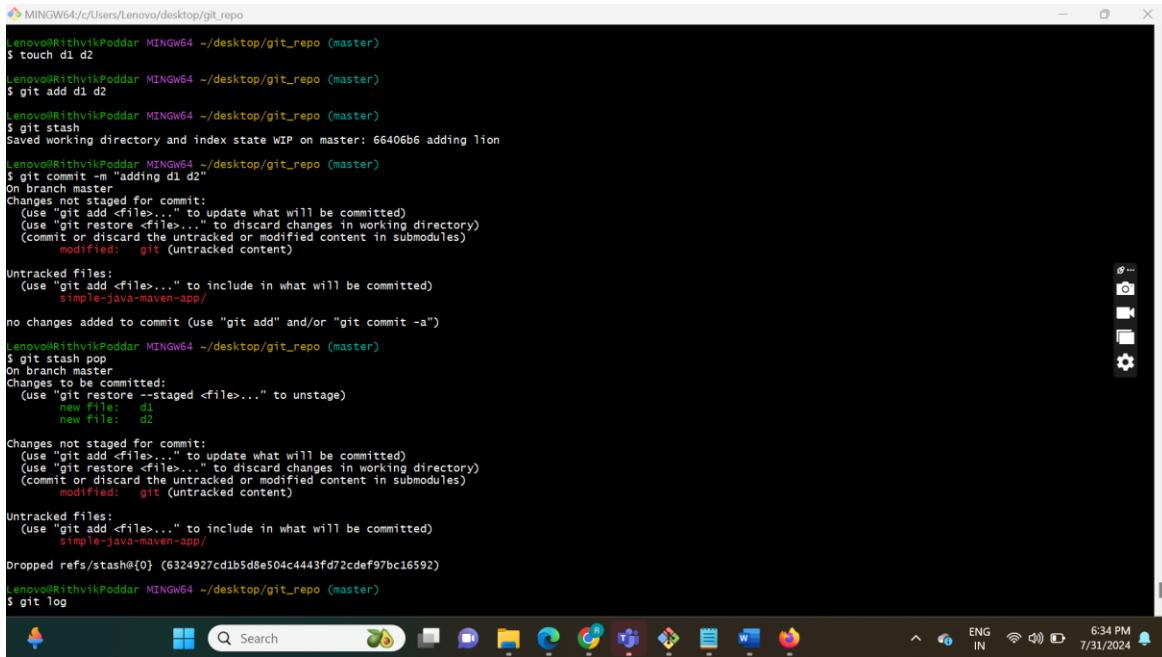
    adding h1

commit a34b0be307ed88d1f4bfc68952b75c7915e61147
Merge: d2ff422 c77fded
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 17:31:07 2024 +0530

    Merge branch 'feature1'

commit c77fdedc42bd7cdedf0bcf3cf52d9cdbdfe51227
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 17:30:32 2024 +0530
```

Stash : if i am working on one branch and i get critical work/bug to be fixed on another branch,i dont want to commit changes in the current brach,so i will do git stash where the files will be stored in area, and switch to other branch i will fix the issue and come back to previous branch to continue and work-> I need to do git stash pop,to get back files from temporary area git stash git stash pop



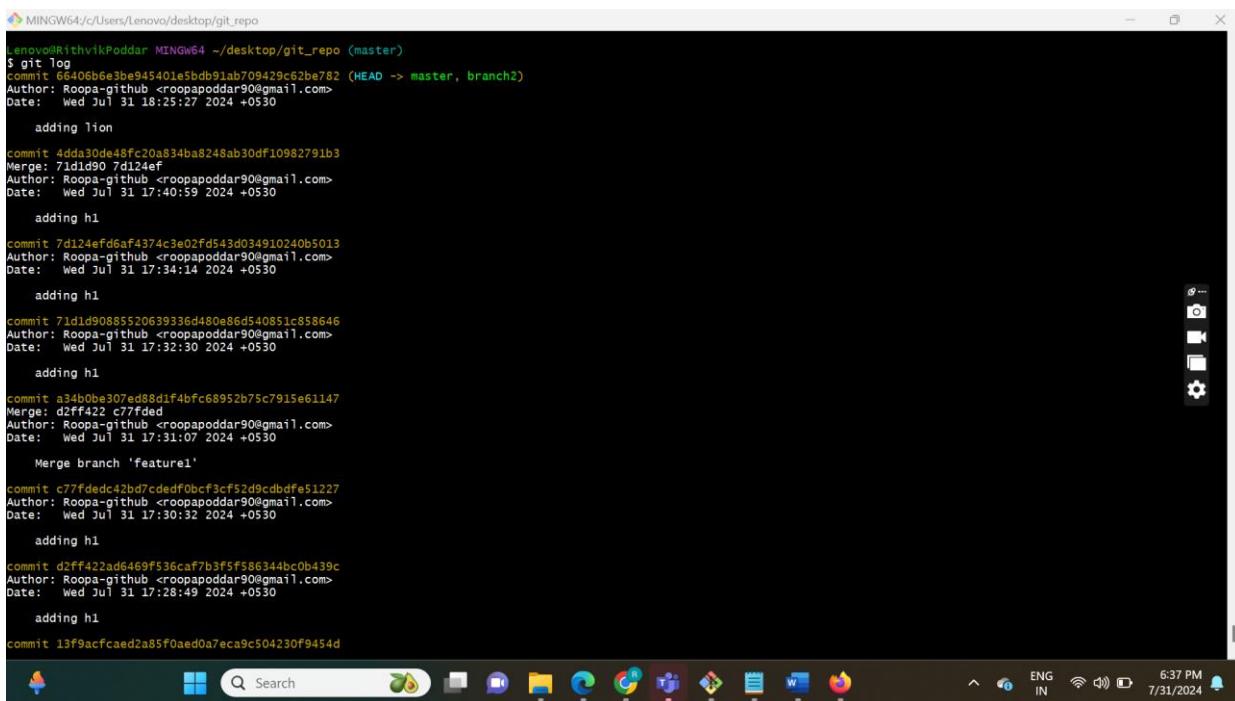
```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
Lenovo|RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ touch d1 d2
Lenovo|RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git add d1 d2
Lenovo|RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git stash
Saved working directory and index state WIP on master: 66406b6 adding lion
Lenovo|RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git commit -m "adding d1 d2"
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    (commit or discard the untracked or modified content in submodules)
      modified:   git (untracked content)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    simple-java-maven-app/
no changes added to commit (use "git add" and/or "git commit -a")

Lenovo|RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git stash pop
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   d1
    new file:   d2

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    (commit or discard the untracked or modified content in submodules)
      modified:   git (untracked content)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    simple-java-maven-app/
Dropped refs/stash@{0} (6324927cd1b5d8e504c4443fd72cdef97bc16592)
Lenovo|RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git log
```



```
MINGW64:/c/Users/Lenovo/Desktop/git_repo
Lenovo|RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git log
commit 66406b6e3be945401e5bd91ab709429c62be782 (HEAD -> master, branch2)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 18:25:27 2024 +0530

  adding lion

commit 4dd40de48fc20a834ba8248ab30df10982791b3
Merge: 71d1d90 7d124ef
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 17:40:59 2024 +0530

  adding h1

commit 7d124ef64f4374c3a02f4543d034910240b5013
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 17:34:14 2024 +0530

  adding h1

commit 71d1d90885520693336d480e86d540851c858646
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 17:32:30 2024 +0530

  adding h1

commit a34b0be307ed83d1f4bfc68952b75c7915e61147
Merge: d2ff422 c77fded
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 17:31:07 2024 +0530

  Merge branch 'feature1'

commit c77fdedc42bd7cdedf0bcf3cf52d9cdbdfe51227
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 17:30:32 2024 +0530

  adding h1

commit d2ff422ad6469f536caf7b3f5f586344bc0b439c
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 17:28:49 2024 +0530

  adding h1

commit 13f9acfcaed2a85f0aed0a7eca9c504230f9454d
```

Squash : it is a technique to condense large number of commits to make into small number of meaningful commits so that we can make git histroy clear --> git rebase -n HEAD~n

```
MINGW64 /c/Users/Lenovo/Desktop/git_repo

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (master)
$ git log
commit 406023e8cf15b904caccefef8ad1075b64f4e0067 (HEAD -> master)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 19:02:20 2024 +0530

    adding t3

commit 475b3f2951258540857c449e22a43fc2d2d900f0
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 19:01:48 2024 +0530

    adding t2

commit 0613cd7f0a4f4599268509541972aba1728db84
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 19:01:01 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (master)
$ git rebase -i HEAD~2
[detached HEAD 5c6c351] adding t2
Date: Wed Jul 31 19:01:48 2024 +0530
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 t2
create mode 100644 t3
Successfully rebased and updated refs/heads/master.

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (master)
$ git log
commit 5c6c3515a9069702e02be09a862ec7d2a13140c (HEAD -> master)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 19:01:48 2024 +0530

    adding t2

    adding t3

commit 0613cd7f0a4f4599268509541972aba1728db84
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 19:01:01 2024 +0530

    adding t1

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo (master)
$ |
```

The screenshot shows a GitHub interface with multiple tabs at the top: maven repos, simple java, Roopa-github, sample java, Branches · Roopa-github, GitHub Dashboards, and Roopa-github. The main navigation bar includes Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. A search bar at the top right says "Type / to search". A modal window titled "Create a branch" is open, prompting for a "New branch name" (branch5) and a "Source" (main). A note says "Your main branch isn't protected" and "Protect this branch". Buttons for "Cancel" and "Create new branch" are visible. The background shows a list of branches: main, fizz, and a partially visible branch starting with 'fizz'. The status bar at the bottom indicates ENG IN, 7:09 PM, and 7/31/2024.

The screenshot shows a GitHub interface with the same tab arrangement as the previous one. The main navigation bar is identical. A modal window titled "Commit changes" is open, containing fields for "Commit message" (Create tree) and "Extended description" (Add an optional extended description...). It includes radio buttons for "Commit directly to the branches branch" (selected) and "Create a new branch for this commit and start a pull request". Buttons for "Cancel" and "Commit changes..." are present. The background shows a list of branches: git / tree in branch5, with the "tree" branch selected. The status bar at the bottom indicates ENG IN, 7:09 PM, and 7/31/2024.

git fetch --> bring changes from central repo to separate branch (under FETCH_HEAD) without merging.

```
MINGW64:/c/Users/Lenovo/desktop/git_repo/git
and the repository exists.

LenovoRithvikPoddar MINGW64 ~/desktop/git_repo/git (main)
$ git fetch origin main
From https://github.com/Roopa-github/git
 * branch          main       -> FETCH_HEAD

LenovoRithvikPoddar MINGW64 ~/desktop/git_repo/git (main)
$ git checkout FETCH_HEAD
Note: switching to 'FETCH_HEAD'.

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 efb3497 Merge pull request #7 from Roopa-github/fizz

LenovoRithvikPoddar MINGW64 ~/desktop/git_repo/git ((efb3497...))
$ git log
commit efb3497fd53f388db7068bb8e901cd918b627b92 (HEAD, origin/main, origin/HEAD, main)
Merge: 731f29f 9a583f2
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 17:24:43 2024 +0530

Merge pull request #7 from Roopa-github/fizz

  Create file7

commit 9a583f246f252983e0ebe3e53f21d860a9fc6593 (origin/fizz)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 17:23:27 2024 +0530

  Create file7

commit 731f29f8332e4bdd58fc7b69fd15fc2909892f8
Merge: c77c84c 1053ee7
Author: Roopa-github <roopapoddar90@gmail.com>

  Create file7

  Create file7

commit 9a583f246f252983e0ebe3e53f21d860a9fc6593 (origin/fizz)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 17:24:43 2024 +0530

  Create file7

commit 731f29f8332e4bdd58fc7b69fd15fc2909892f8
Merge: c77c84c 1053ee7
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Fri Jul 26 16:53:02 2024 +0530

  Merge pull request #5 from Roopa-github/giga

  Create yu

commit 1053ee7cc0b387691fa5ed31f2b9c72c326d68eb (origin/giga)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Fri Jul 26 16:50:56 2024 +0530

  Create yu

commit c77c84cb8925fa728a6c757aa945d6f97f9da3bc
Merge: 91e2bf7 5303c0e
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Fri Jul 26 16:28:31 2024 +0530

  Merge pull request #4 from Roopa-github/branch8

  Create t2

commit 5303c0ebaa6acc8866573da31278a8495e5b7a (origin/branch8)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Fri Jul 26 16:26:10 2024 +0530

  Create t2

commit 91e2bf7a86a7547e3413d0e2169fc24157c23bd6
Author: Roopa-github <roopapoddar90@gmail.com>
::
```

```
MINGW64:/c/Users/Lenovo/desktop/git_repo/git
commit efb3497fd53f388db7068bb8e901cd918b627b92 (HEAD, origin/main, origin/HEAD, main)
Merge: 731f29f 9a583f2
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 17:24:43 2024 +0530

Merge pull request #7 from Roopa-github/fizz

  Create file7

commit 9a583f246f252983e0ebe3e53f21d860a9fc6593 (origin/fizz)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 17:23:27 2024 +0530

  Create file7

  Create file7

commit 9a583f246f252983e0ebe3e53f21d860a9fc6593 (origin/fizz)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Tue Jul 30 17:24:43 2024 +0530

  Create file7

commit 731f29f8332e4bdd58fc7b69fd15fc2909892f8
Merge: c77c84c 1053ee7
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Fri Jul 26 16:53:02 2024 +0530

  Merge pull request #5 from Roopa-github/giga

  Create yu

commit 1053ee7cc0b387691fa5ed31f2b9c72c326d68eb (origin/giga)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Fri Jul 26 16:50:56 2024 +0530

  Create yu

commit c77c84cb8925fa728a6c757aa945d6f97f9da3bc
Merge: 91e2bf7 5303c0e
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Fri Jul 26 16:28:31 2024 +0530

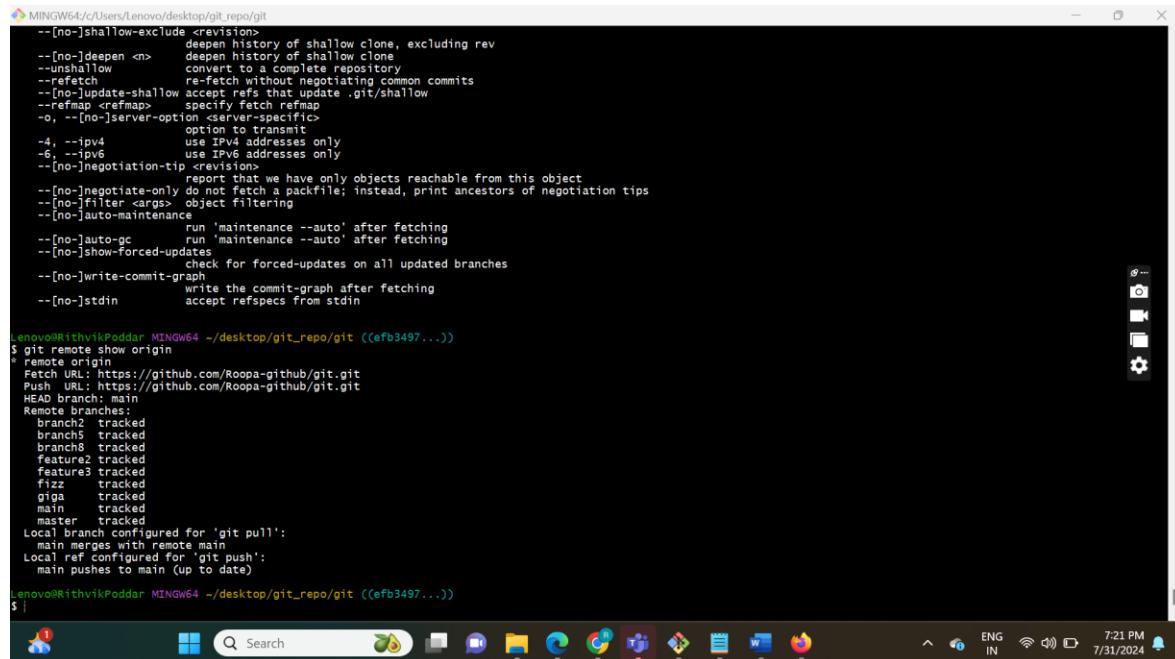
  Merge pull request #4 from Roopa-github/branch8

  Create t2

commit 5303c0ebaa6acc8866573da31278a8495e5b7a (origin/branch8)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:  Fri Jul 26 16:26:10 2024 +0530

  Create t2

commit 91e2bf7a86a7547e3413d0e2169fc24157c23bd6
Author: Roopa-github <roopapoddar90@gmail.com>
::
```



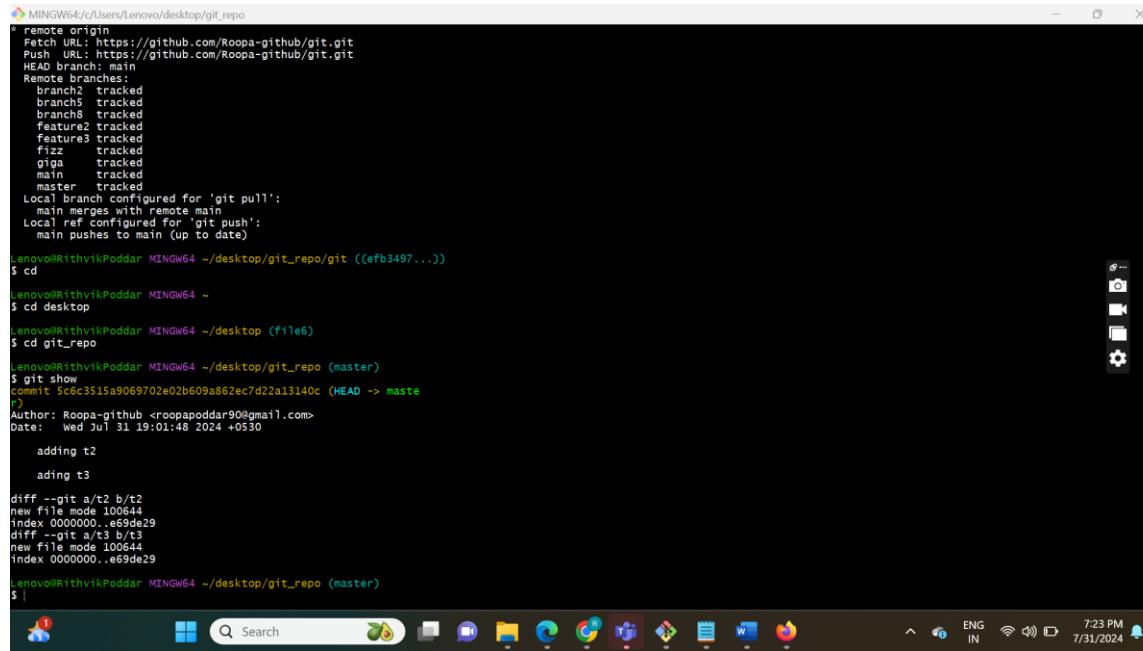
```

MINGW64:/c/Users/Lenovo/Desktop/git_repo/git
--[no-]shallow-exclude <revision>
          deepen history of shallow clone, excluding rev
--[no-]deepen <n>
          deepen history of shallow clone
--unshallow
          convert to a complete repository
--refetch
          re-fetch objects incorporating common commits
--[no-]update-shallow
          accept refs that update .git/shallow
--refmap <refmap>
          specify fetch refmap
-o, --[no-]server-option <server-specific>
          option to transmit
-4, --ipv4
          use IPv4 addresses only
-6, --ipv6
          use IPv6 addresses only
--[no-]negotiation-tip <version>
          report what we only have objects reachable from this object
--[no-]negotiate-only do not fetch a packfile; instead, print ancestors of negotiation tips
--[no-]filter <args>
          object filtering
--[no-]auto-maintenance
          run 'maintenance --auto' after fetching
--[no-]auto-gc
          run 'maintenance --auto' after fetching
--[no-]show-forced-updates
          check for forced-updates on all updated branches
--[no-]write-commit-graph
          write the commit-graph after fetching
--[no-]stdin
          accept refspecs from stdin

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git ((efb3497...))
$ git remote show origin
remote: origin
Fetch URL: https://github.com/Roopa-github/git.git
Push URL: https://github.com/Roopa-github/git.git
HEAD branch: main
Remote branches:
  branch2 tracked
  branch3 tracked
  branch4 tracked
  feature2 tracked
  feature3 tracked
  fizz tracked
  giga tracked
  main tracked
  master tracked
Local branch configured for 'git pull':
  main merges with remote main
Local ref configured for 'git push':
  main pushes to main (up to date)
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git ((efb3497...))
$ |

```

Git Show : git-show is a command line utility that is used to view expanded details on Git objects such as blobs, trees, tags, and commits. git-show has specific behavior per object type. ... Blobs show the direct content of the blob. Commits show a commit log message and a diff output of the changes in the commit-> Git show is similar to git log but it also shows which line and what has been modified

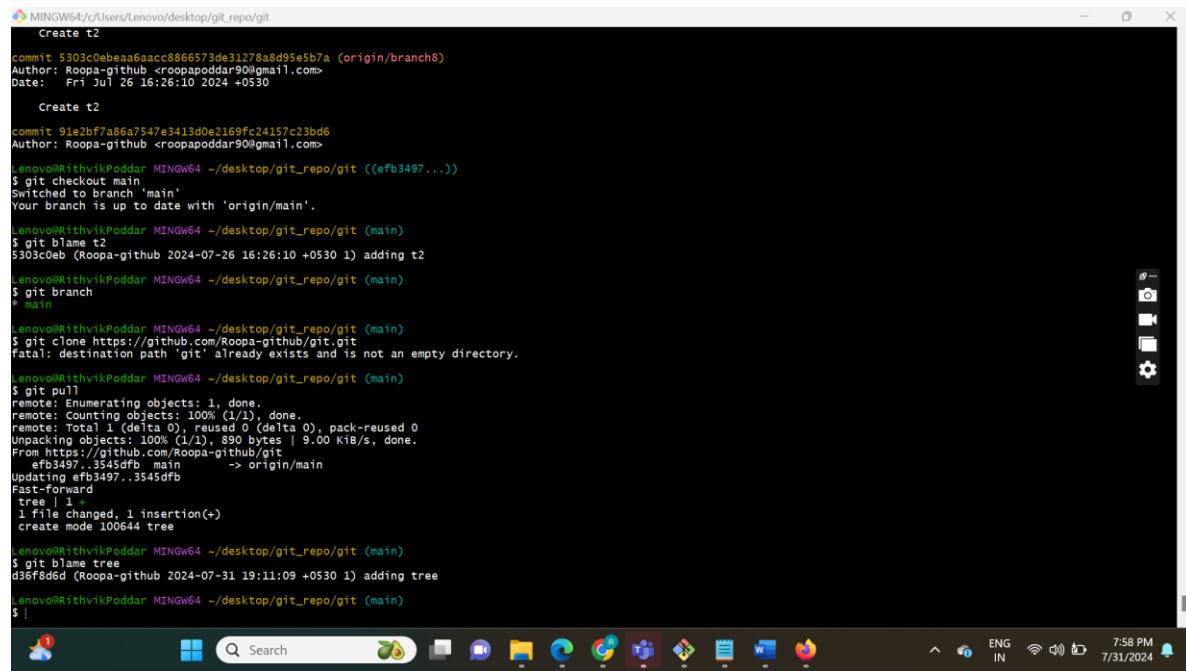


```

MINGW64:/c/Users/Lenovo/Desktop/git_repo
$ remote origin
* remote origin
  Fetch URL: https://github.com/Roopa-github/git.git
  Push URL: https://github.com/Roopa-github/git.git
  HEAD branch: main
  Remote branches:
    branch2 tracked
    branch3 tracked
    branch4 tracked
    feature2 tracked
    feature3 tracked
    fizz tracked
    giga tracked
    main tracked
    master tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (up to date)
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git ((efb3497...))
$ cd
Lenovo@RithvikPoddar MINGW64 ~
$ cd desktop
Lenovo@RithvikPoddar MINGW64 ~/Desktop (file6)
$ cd git_repo
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ git show
commit 5c6c3515a9069702e02b609a862ec7d2a13140c (HEAD -> master)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Wed Jul 31 19:01:48 2024 +0530
        adding t2
        adding t3
diff --git a/t2 b/t2
new file mode 100644
index 000000..e69de29
diff --git a/t3 b/t3
new file mode 100644
index 000000..e69de29
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo (master)
$ |

```

Git blame : it will show who has modified the code(each line of the codes)



```
MINGW64:/c/Users/Lenovo/Desktop/git_repo/git
Create t2
commit 5303cbea6aacc8866573de31278a8d95e5b7a (origin/branch8)
Author: Roopa-github <roopapoddar90@gmail.com>
Date:   Fri Jul 26 16:26:10 2024 +0530

Create t2
commit 91e2bf7a86a7547e3413d0e2169fc24157c23bd6
Author: Roopa-github <roopapoddar90@gmail.com>

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git ((efb3497...))
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ git blame t2
5303c0eb (Roopa-github 2024-07-26 16:26:10 +0530 1) adding t2

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ git branch
* main

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ git clone https://github.com/Roopa-github/git.git
fatal: destination path 'git' already exists and is not an empty directory.

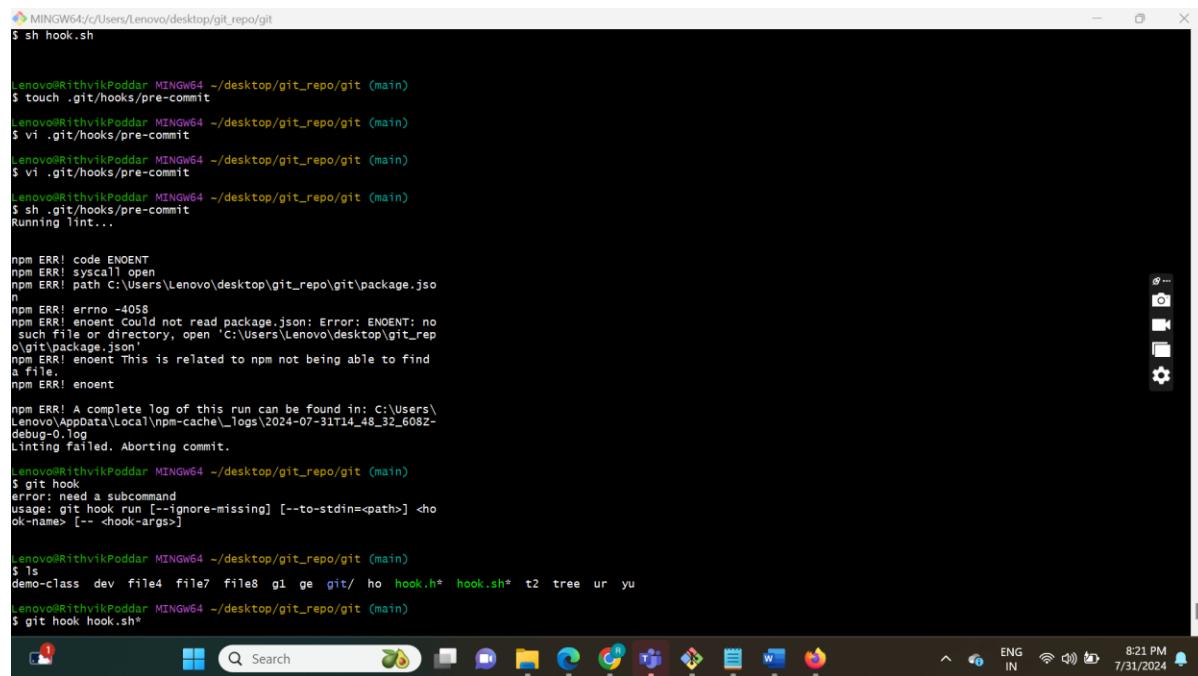
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ git pull
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Compressing objects: 100% (1/1), done (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), 890 bytes | 9.00 KiB/s, done.
From https://github.com/Roopa-github/git
  efb3497..3545dfb main      -> origin/main
Updating efb3497..3545dfb
Fast-forward
 tree | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 tree

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ git blame tree
d36f8d6d (Roopa-github 2024-07-31 19:11:09 +0530 1) adding tree

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$
```

Git Hook :Git hook are scripts that run automatically every time a particluar event occur in a git repository

Git hooks are scripts that Git executes before or after certain events, like committing or pushing. These hooks can be used to enforce policies, automate workflows, and perform various checks or actions. Here's a guide on how to set up and use Git hooks:



```
MINGW64:/c/Users/Lenovo/Desktop/git_repo/git
$ sh hook.sh

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ touch .git/hooks/pre-commit
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ vi .git/hooks/pre-commit
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ vi .git/hooks/pre-commit
Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ sh .git/hooks/pre-commit
Running lint...

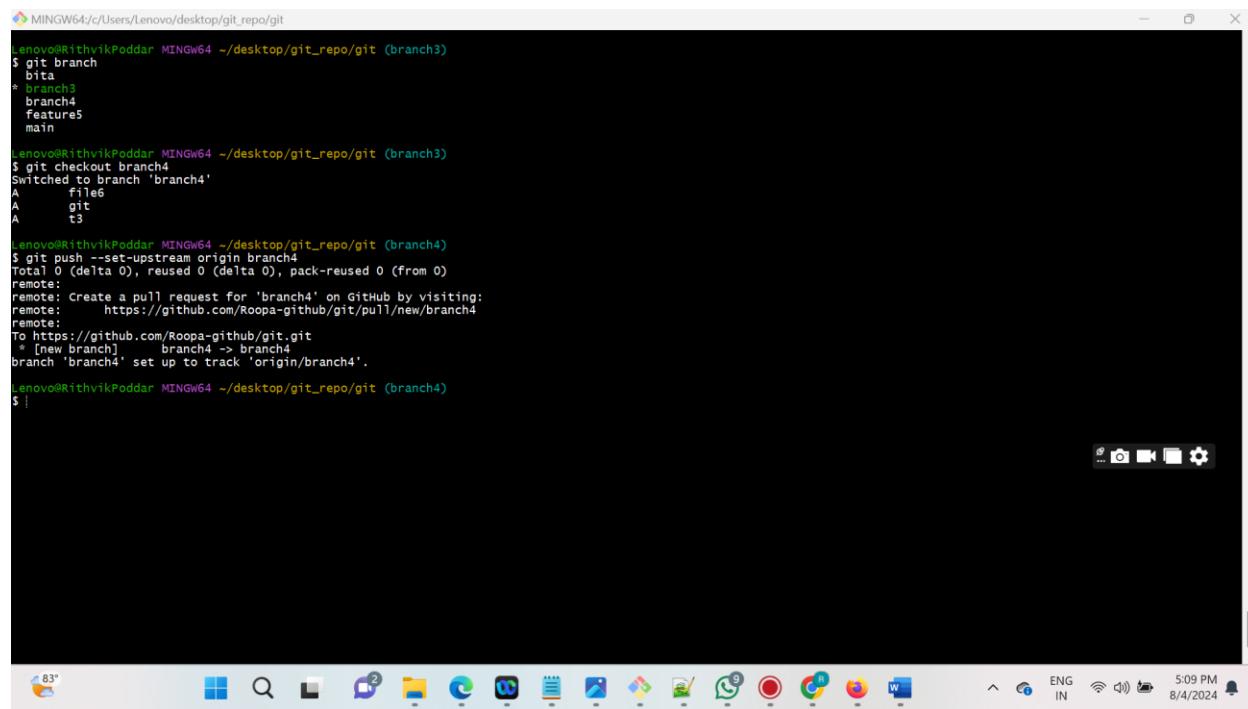
npm ERR! code ENOENT
npm ERR! syscall open
npm ERR! path C:\Users\Lenovo\Desktop\git_repo\git\package.json
...
npm ERR! errno -4058
npm ERR! enoent could not read package.json: Error: ENOENT: no such file or directory, open 'C:\Users\Lenovo\Desktop\git_repo\git\package.json'
npm ERR! enoent This is related to npm not being able to find a file.
npm ERR! enoent
npm ERR! enoent A complete log of this run can be found in: C:\Users\Lenovo\AppData\Local\npm-cache\_logs\2024-07-31T14_48_32_608Z-debug-0.log
Linting failed. Aborting commit.

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ git hook
error: need a subcommand
usage: git hook run [--ignore-missing] [--to-stdin=<path>] <hook-name> [-- <hook-args>]

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ ls
demo-class dev file4 file7 file8 g1 ge git/ ho hook.h* hook.sh* t2 tree ur yu

Lenovo@RithvikPoddar MINGW64 ~/Desktop/git_repo/git (main)
$ git hook hook.sh*
```

Git push:

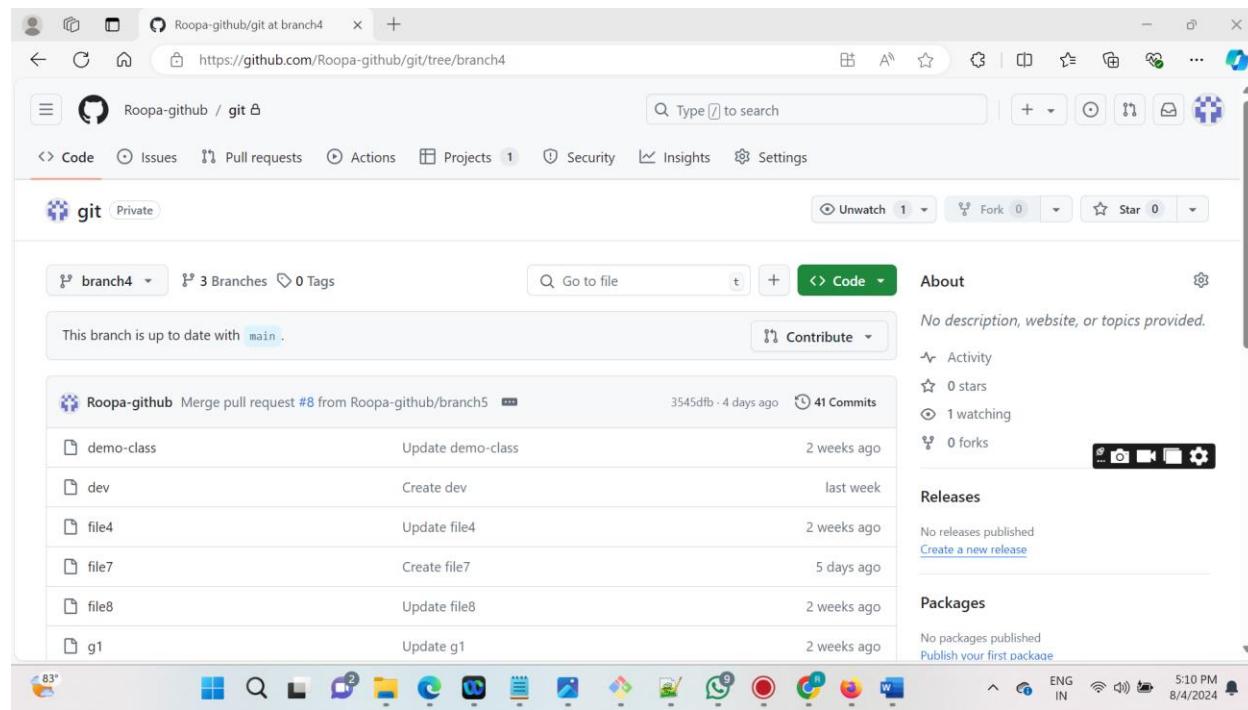


```
Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo/git (branch3)
$ git branch
  bits
* Branch3
  branch4
  feature5
  main

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo/git (branch4)
$ git checkout branch4
Switched to branch 'branch4'
A       file6
A       git
A       t3

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo/git (branch4)
$ git push --set-upstream origin branch4
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'branch4' on GitHub by visiting:
remote:   https://github.com/Roopa-github/git/pull/new/branch4
remote:
To https://github.com/Roopa-github/git.git
 * [new branch]  branch4 -> branch4
branch 'branch4' set up to track 'origin/branch4'.

Lenovo@RithvikPoddar MINGW64 ~/desktop/git_repo/git (branch4)
$
```



The screenshot shows a GitHub repository page for 'Roopa-github/git at branch4'. The repository has 3 branches and 0 tags. It contains 41 commits from the 'Roopa-github' user. Recent commits include 'Update demo-class', 'Create dev', 'Update file4', 'Create file7', 'Update file8', and 'Update g1'. The repository has 0 forks and 0 stars. There are sections for Releases (no releases published) and Packages (no packages published). The GitHub interface includes a search bar, navigation links like Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings, and a sidebar with repository statistics.

Branches · Roopa-github/git

Overview Yours Active Stale All

Search branches...

Default

Branch	Updated	Check status	Behind Ahead	Pull request
main	4 days ago	Default	1 0	#3

Your branches

Branch	Updated	Check status	Behind Ahead	Pull request
branch4	3 minutes ago	0 0		
branch3	2 hours ago	0 0		

Active branches

Branch	Updated	Check status	Behind Ahead	Pull request
branch4	3 minutes ago	0 0		
branch3	2 hours ago	0 0		