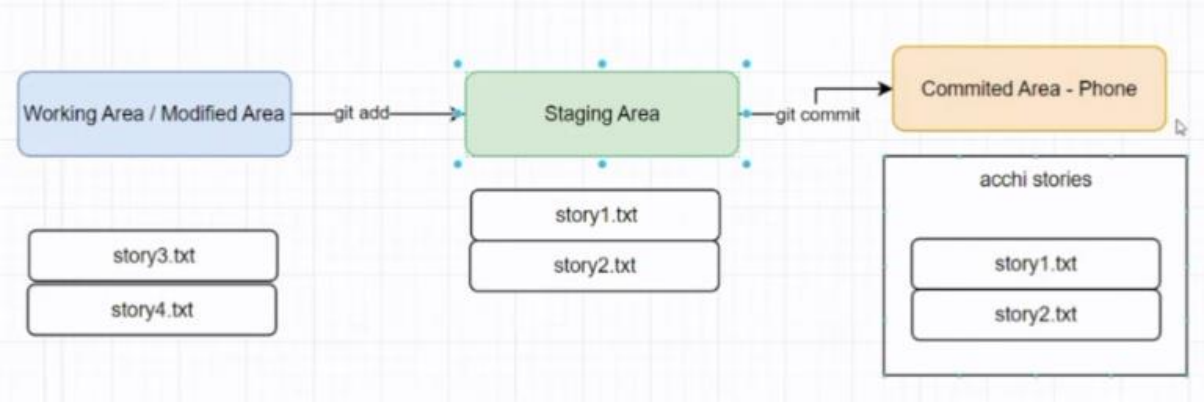
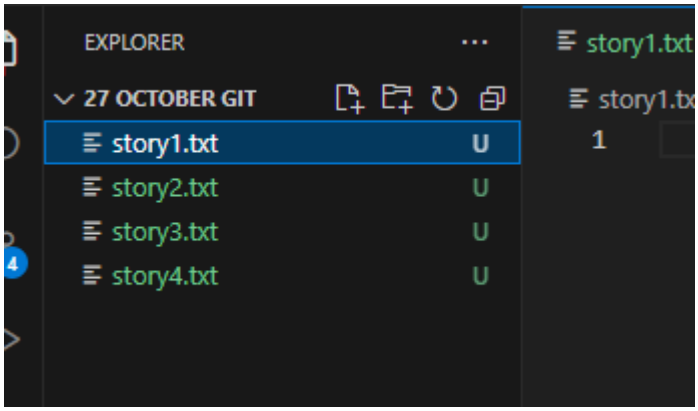
26 OCT

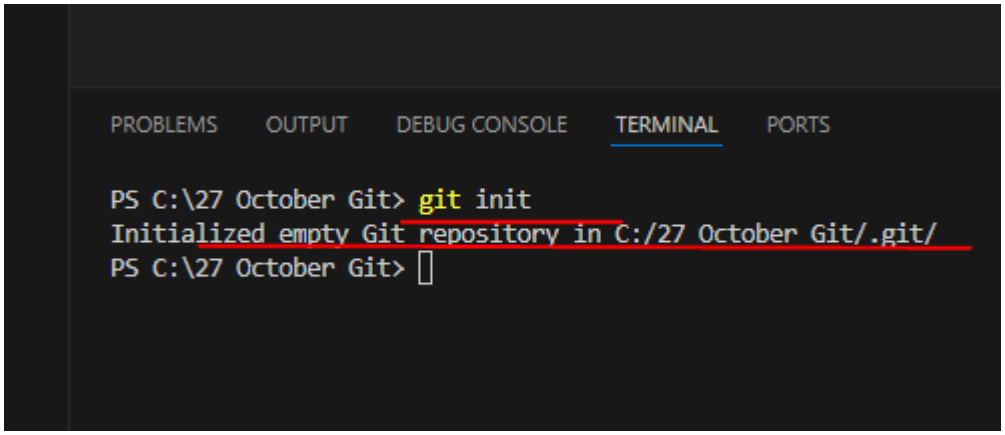
1)



2) Suppose created folder “27 October Git” and created 4 txt files story1.txt, story2.txt, story3.txt, story4.txt.



3) **git init** – creates and empty repository



4) **git add .\story1.txt** – To put file from working area to staging area

5) **git status**

```
PS C:\27 October Git> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   story1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        story2.txt
        story3.txt
        story4.txt

PS C:\27 October Git>
```

6) **git add .\story2.txt**

git status

```
PS C:\27 October Git> git add .\story2.txt
PS C:\27 October Git> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   story1.txt
        new file:   story2.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        story3.txt
        story4.txt

PS C:\27 October Git>
```

7) **git add . –** now all files get set into staging area

git status

```
PS C:\27 October Git> git add
PS C:\27 October Git> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   story1.txt
        new file:   story2.txt
        new file:   story3.txt
        new file:   story4.txt

PS C:\27 October Git>
```

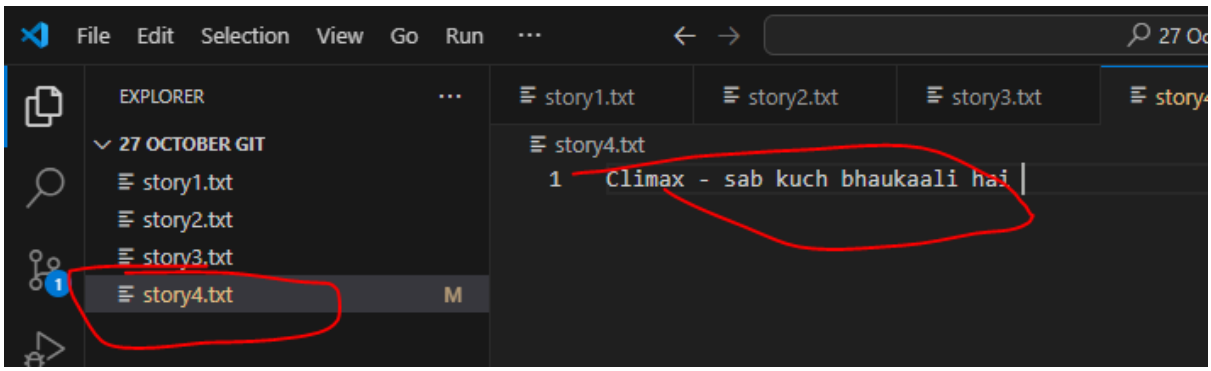
8) **git commit -m "story added" –** all files have been added into database

```
PS C:\27 October Git> git commit -m "story added"
[master (root-commit) bb2eebc] story added
 4 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 story1.txt
 create mode 100644 story2.txt
 create mode 100644 story3.txt
 create mode 100644 story4.txt
PS C:\27 October Git>
```

9) **git status** – nothing to commit

```
PS C:\27 October Git> git status
On branch master
nothing to commit, working tree clean
PS C:\27 October Git>
```

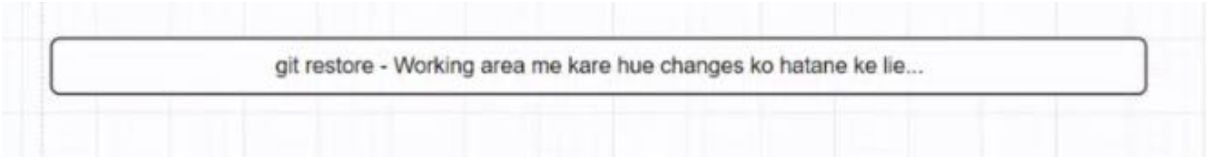
10) Suppose added something in story 4



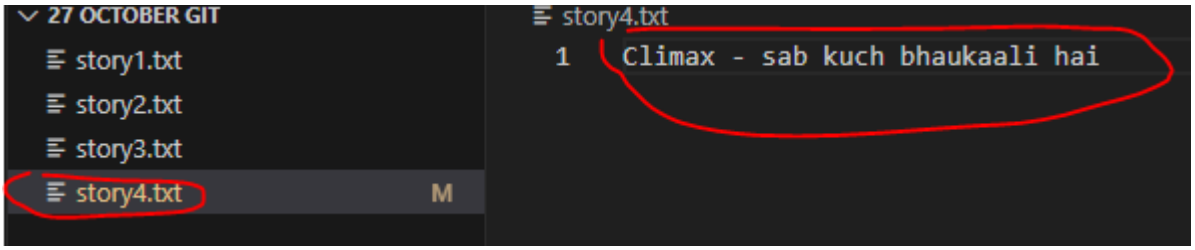
11) git status

```
PS C:\27 October Git> 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)
        modified:   story4.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\27 October Git>
```

12) **git restore** – back to previous state



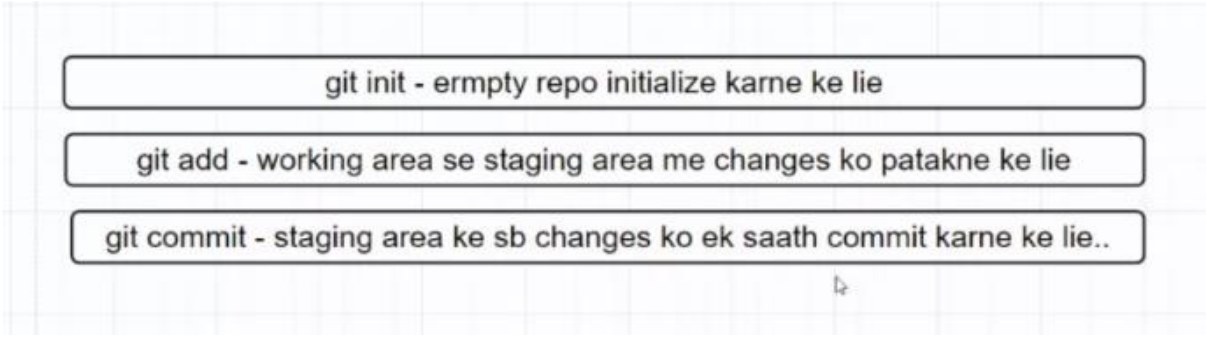
13) Now if added part reached in staging area then how can we restore. So restore file from staging area to working area.




```
PS C:\27 October Git> 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)
        modified:   story4.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\27 October Git> git add .
PS C:\27 October Git> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   story4.txt

PS C:\27 October Git>
```

git restore --staged story4.txt

```
PS C:\27 October Git> git restore --staged story4.txt
PS C:\27 October Git> 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)
        modified:   story4.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\27 October Git>
```



14) added climax in story4.txt

git add .

git commit -m “added climax”

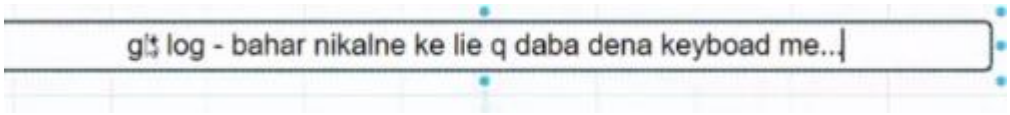
```
PS C:\27 October Git> git add .
PS C:\27 October Git> git sttatus
git: 'sttatus' is not a git command. See 'git --help'.

The most similar command is
        status
PS C:\27 October Git> git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   story4.txt

PS C:\27 October Git> git commit -m "added climax"
[master bfc24e2] added climax
 1 file changed, 1 insertion(+)
PS C:\27 October Git> git status
On branch master
nothing to commit, working tree clean
PS C:\27 October Git>
```

15) **git log** – to see how many commits are there in committed area

```
nothing to commit, working tree clean
PS C:\27 October Git> git log
commit bfc24e2c16f45aadb18f1068ee86db2d17d1fe65 (HEAD -> master)
Author: Gauravrajsingh <er.gauravraj8052625814@gmail.com>
Date:   Thu Oct 31 13:15:10 2024 +0530

    added climax

commit bb2eebc3b068c8bbfc2759f2058cade37dc28523
Author: Gauravrajsingh <er.gauravraj8052625814@gmail.com>
Date:   Wed Oct 30 17:36:28 2024 +0530

    story added
PS C:\27 October Git>
```

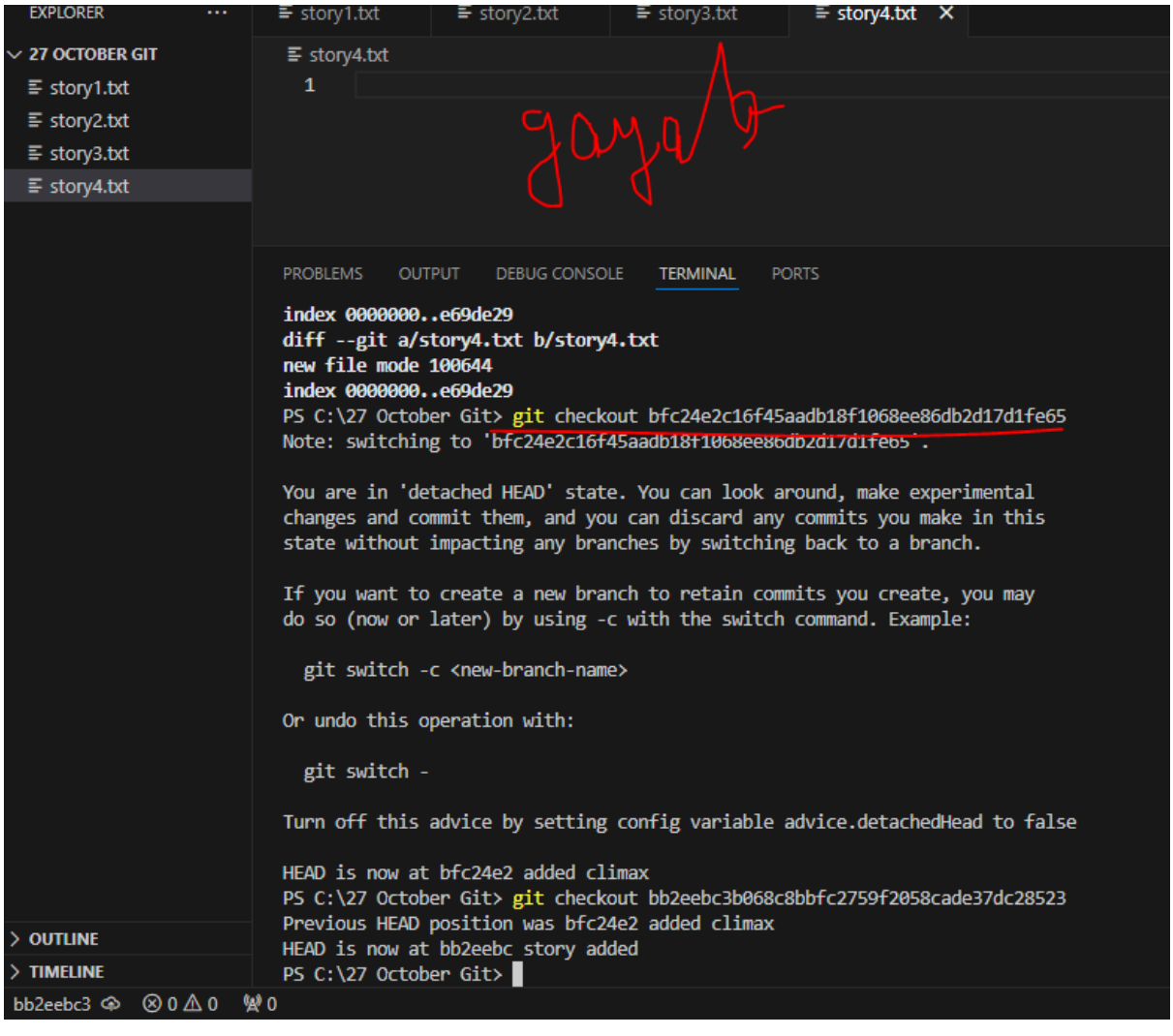
16) **git log -p** = shows previous deleted and newly added things

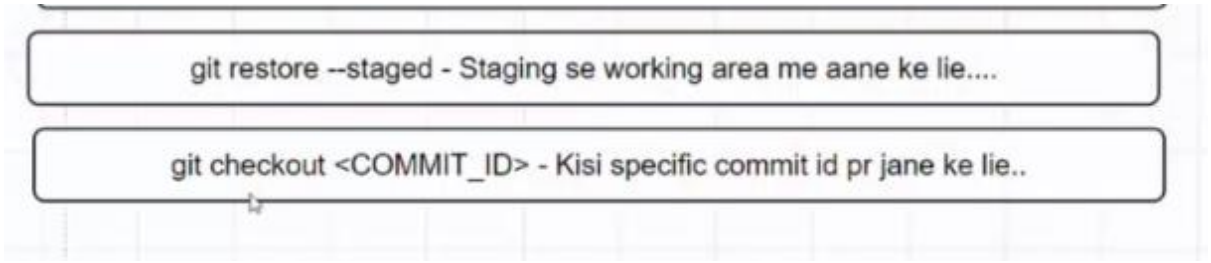
```
PS C:\27 October Git> git log -p
commit bfc24e2c16f45aadb18f1068ee86db2d17d1fe65 (HEAD -> master)
Author: Gauravrajsingh <er.gauravraj8052625814@gmail.com>
Date:   Thu Oct 31 13:15:10 2024 +0530

    added climax

diff --git a/story4.txt b/story4.txt
index e69de29..3bb748c 100644
--- a/story4.txt
+++ b/story4.txt
@@ -0,0 +1 @@
+Climax - sab kuch bhaukaali hai
\ No newline at end of file

commit bb2eebc3b068c8bbfc2759f2058cade37dc28523
Author: Gauravrajsingh <er.gauravraj8052625814@gmail.com>
Date:   Wed Oct 30 17:36:28 2024 +0530

    story added

diff --git a/story1.txt b/story1.txt
new file mode 100644
index 0000000..e69de29
diff --git a/story2.txt b/story2.txt
new file mode 100644
index 0000000..e69de29
diff --git a/story3.txt b/story3.txt
new file mode 100644
index 0000000..e69de29
diff --git a/story4.txt b/story4.txt
new file mode 100644
index 0000000..e69de29
```



17) After doing git commit we will not be able to bring it back to previous state i.e. staging or working area. But we can use

git checkout bfc24e2c16f45aadb18f1068ee86db2d17d1fe65 – upar se edit likha gayab as reached to previous state





18) REVISION

git init - empty repo initialize karne ke lie

git add - working area se staging area me changes ko patakne ke lie

git commit - staging area ke sb changes ko ek saath commit karne ke lie..

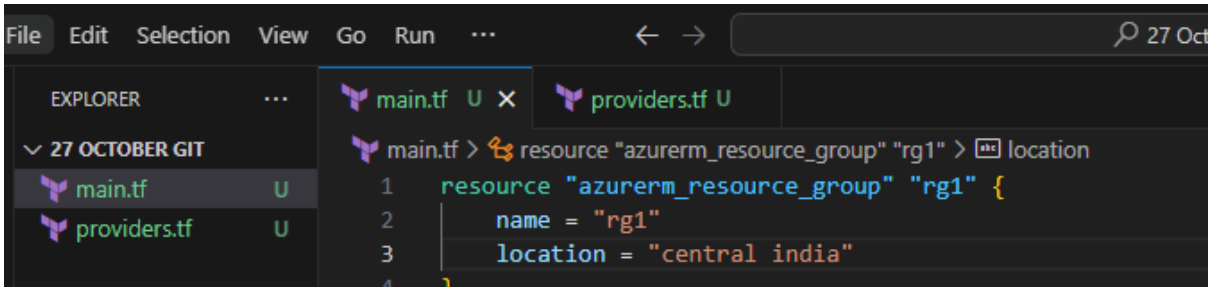
git log - bahar nikalne ke lie q daba dena keyboad me...

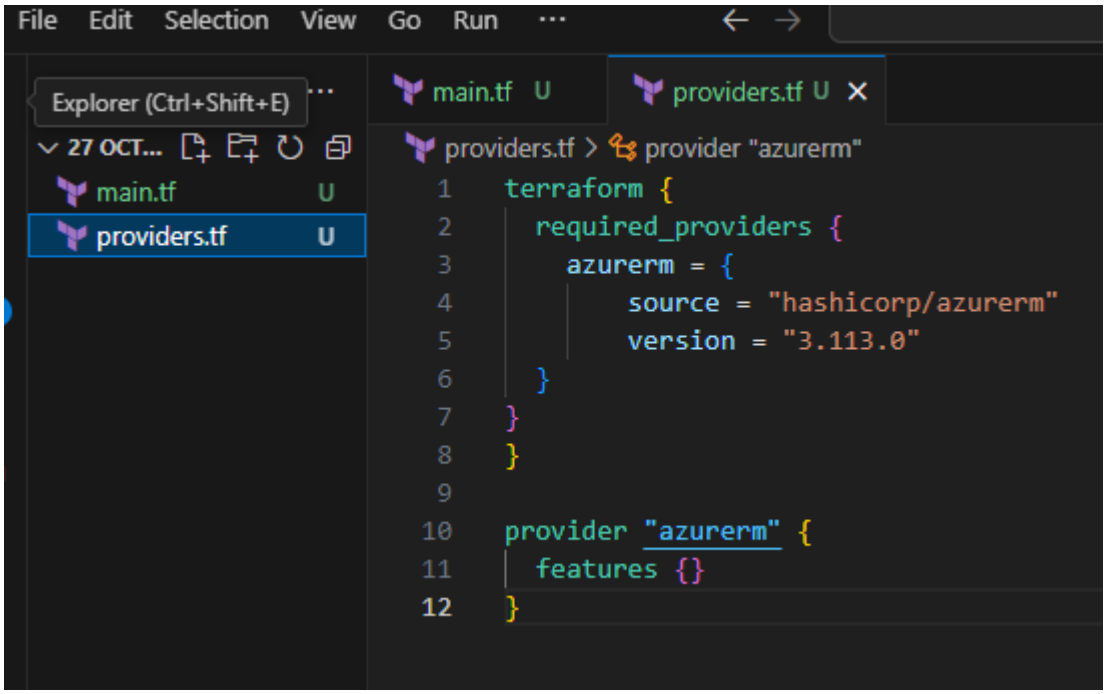
git log -p - kis commit me kya kya changes hai sb dikha dega

++

AGENDA –

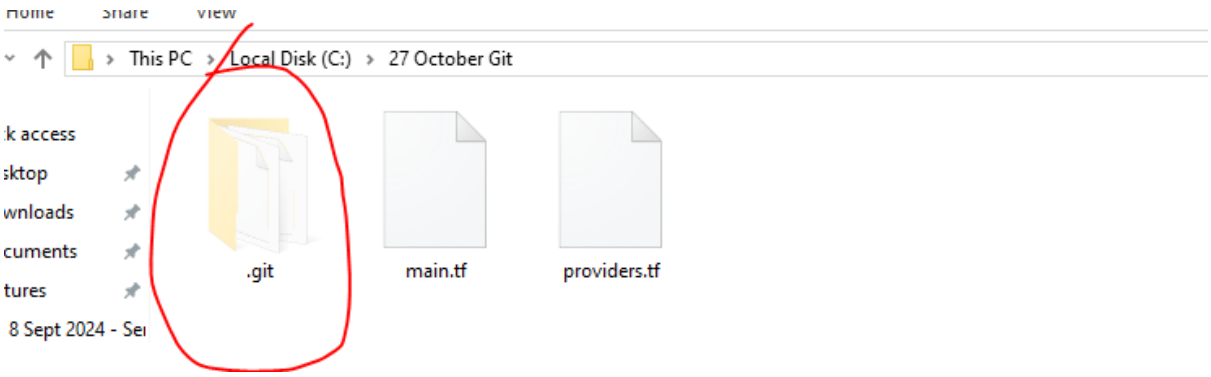
1) Create files in vs code





2) **git init** – Initialized empty Git repository and git folder is created

```
PS C:\27 October Git> git init
Initialized empty Git repository in C:/27 October Git/.git/
PS C:\27 October Git>
```



3) **git add .**

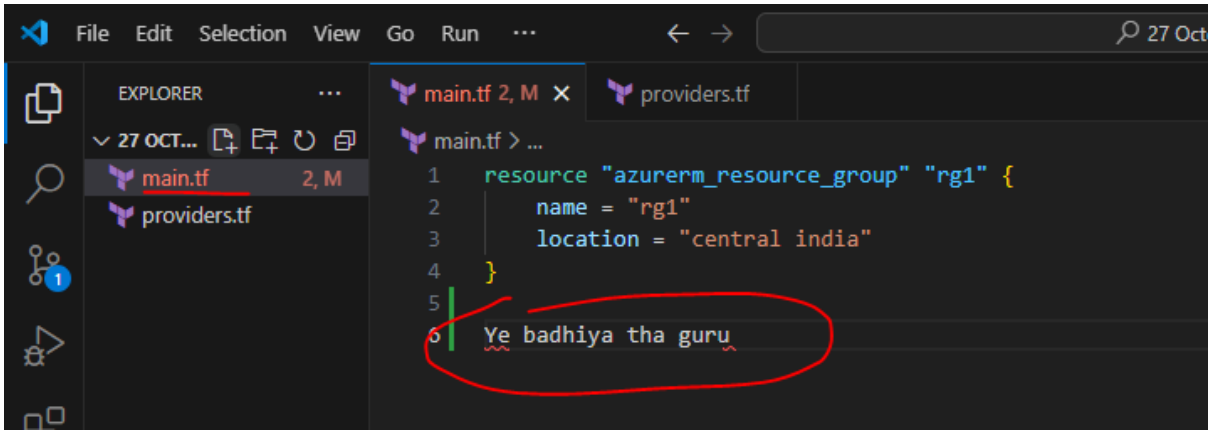
git commit -m "added rg1"

```
PS C:\27 October Git> git add .
PS C:\27 October Git> git commit -m "added rg1"
[master (root-commit) 78a9117] added rg1
 2 files changed, 16 insertions(+)
 create mode 100644 main.tf
 create mode 100644 providers.tf
PS C:\27 October Git>
```

4) git log -p

```
PS C:\27 October Git> git log -p
commit 78a9117b5bc1933d859403a219a3bcbddca2e2e6 (HEAD -> master)
Author: Gauravrajsingh <er.gauravraj8052625814@gmail.com>
Date:   Thu Oct 31 18:31:16 2024 +0530

    added rg1

diff --git a/main.tf b/main.tf
new file mode 100644
index 0000000..371c6bf
--- /dev/null
+++ b/main.tf
@@ -0,0 +1,4 @@
+resource "azurerm_resource_group" "rg1" {
+    name = "rg1"
+    location = "central india"
+}
\ No newline at end of file
diff --git a/providers.tf b/providers.tf
new file mode 100644
index 0000000..942c16a
--- /dev/null
+++ b/providers.tf
@@ -0,0 +1,12 @@
+terraform {
commit 78a9117b5bc1933d859403a219a3bcbddca2e2e6 (HEAD -> master)
Author: Gauravrajsingh <er.gauravraj8052625814@gmail.com>
Date:   Thu Oct 31 18:31:16 2024 +0530

    added rg1

diff --git a/main.tf b/main.tf
new file mode 100644
```

5) Add a line in main.tf file



6) **git add .**

git commit -m "guru added"

```
PS C:\27 October Git> git add .
PS C:\27 October Git> git commit -m "guru added"
[master 8b12ed8] guru added
 1 file changed, 3 insertions(+), 1 deletion(-)
PS C:\27 October Git>
```

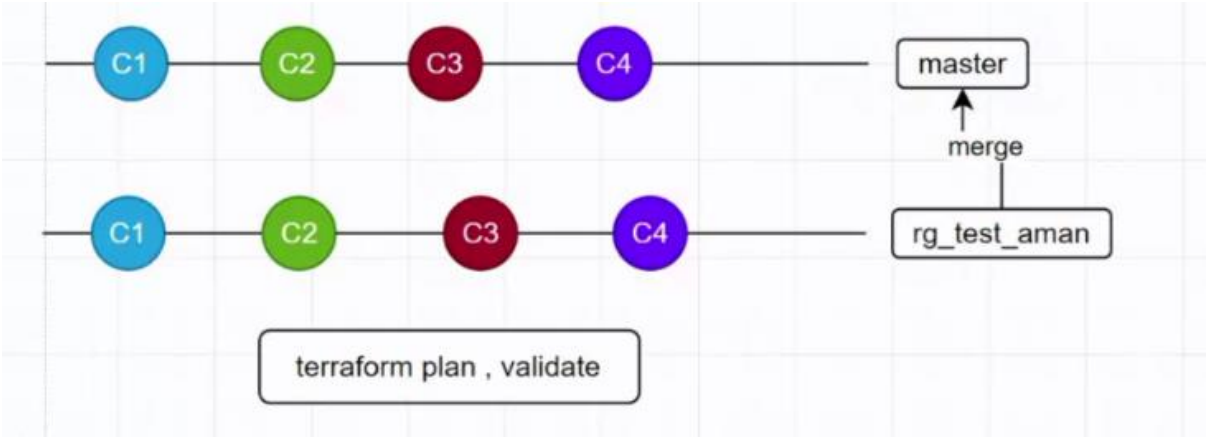
7) **git log –** we can see there are 2 commits

```
PS C:\27 October Git> git log
commit 8b12ed8d5ad37d0dd3d9a97628a91f9ebacfd03e (HEAD -> master)
Author: Gauravrajsingh <er.gauravraj8052625814@gmail.com>
Date:   Thu Oct 31 18:36:51 2024 +0530

    guru added

commit 78a9117b5bc1933d859403a219a3bcbddca2e2e6
Author: Gauravrajsingh <er.gauravraj8052625814@gmail.com>
Date:   Thu Oct 31 18:31:16 2024 +0530

    added rg1
PS C:\27 October Git>
```

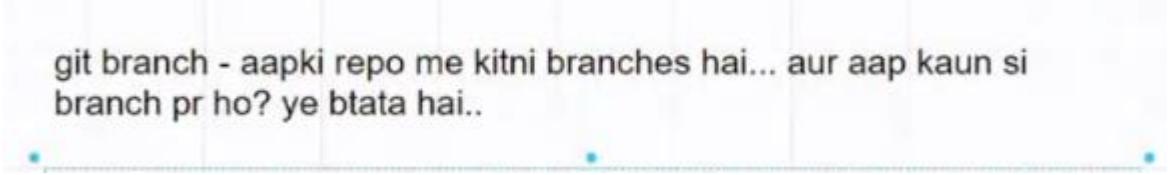
8) So both commits are inside master

++

AGENDA – Branch merge



1) **git branch**



```
PS C:\27 October Git> git branch
* master
PS C:\27 October Git>
```

2) **git branch gauravtest** – creates the branch

```
PS C:\27 October Git> git branch gauravtest
PS C:\27 October Git>
```

3) **git branch –** shows all number of created branches

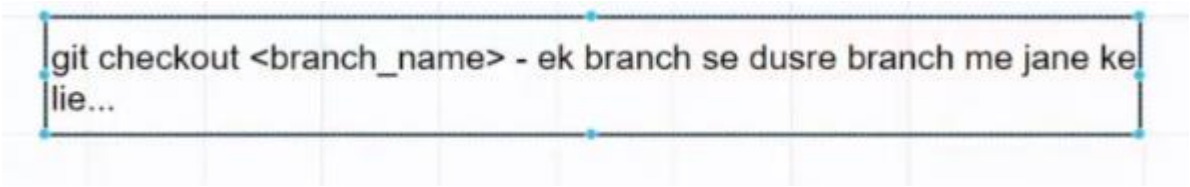
Now gauravtest branch will have same content as master branch

```
PS C:\27 October Git> git branch
  gauravtest
* master
PS C:\27 October Git>
```

4) **git branch ahsishtest** – creates the branch

```
PS C:\27 October Git> git branch ashishtest
PS C:\27 October Git> git branch
  ashishtest
  gauravtest
* master
PS C:\27 October Git>
```

4) **git checkout <branch_name> -** to move from 1 branch to another branch



5) Now, Suppose we have to switch to gauravtest branch

git checkout gauravtest

```
PS C:\27 October Git> git checkout gauravtest
Switched to branch 'gauravtest'
PS C:\27 October Git>
```

6) **git branch**

Here * green colour indicates currently in which branch we are in

```
PS C:\27 October Git> git branch
  ashishtest
* gauravtest
  master
PS C:\27 October Git>
```

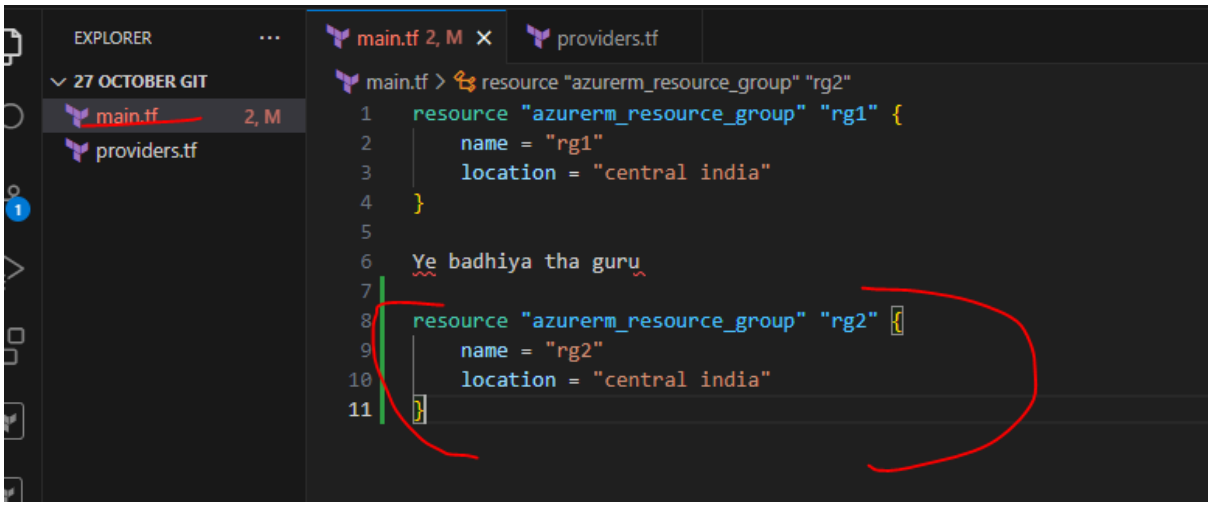
7) **git checkout -b rakhisawant** – creates as well as switched to a new branch

```
PS C:\27 October Git> git checkout -b rakhisawant
Switched to a new branch 'rakhisawant'
PS C:\27 October Git>
```

8) **git branch**

```
PS C:\27 October Git> git branch
  ashishtest
  gauravtest
  master
* rakhisawant
PS C:\27 October Git>
```

9) After switching into that branch just add rg code in main.tf file

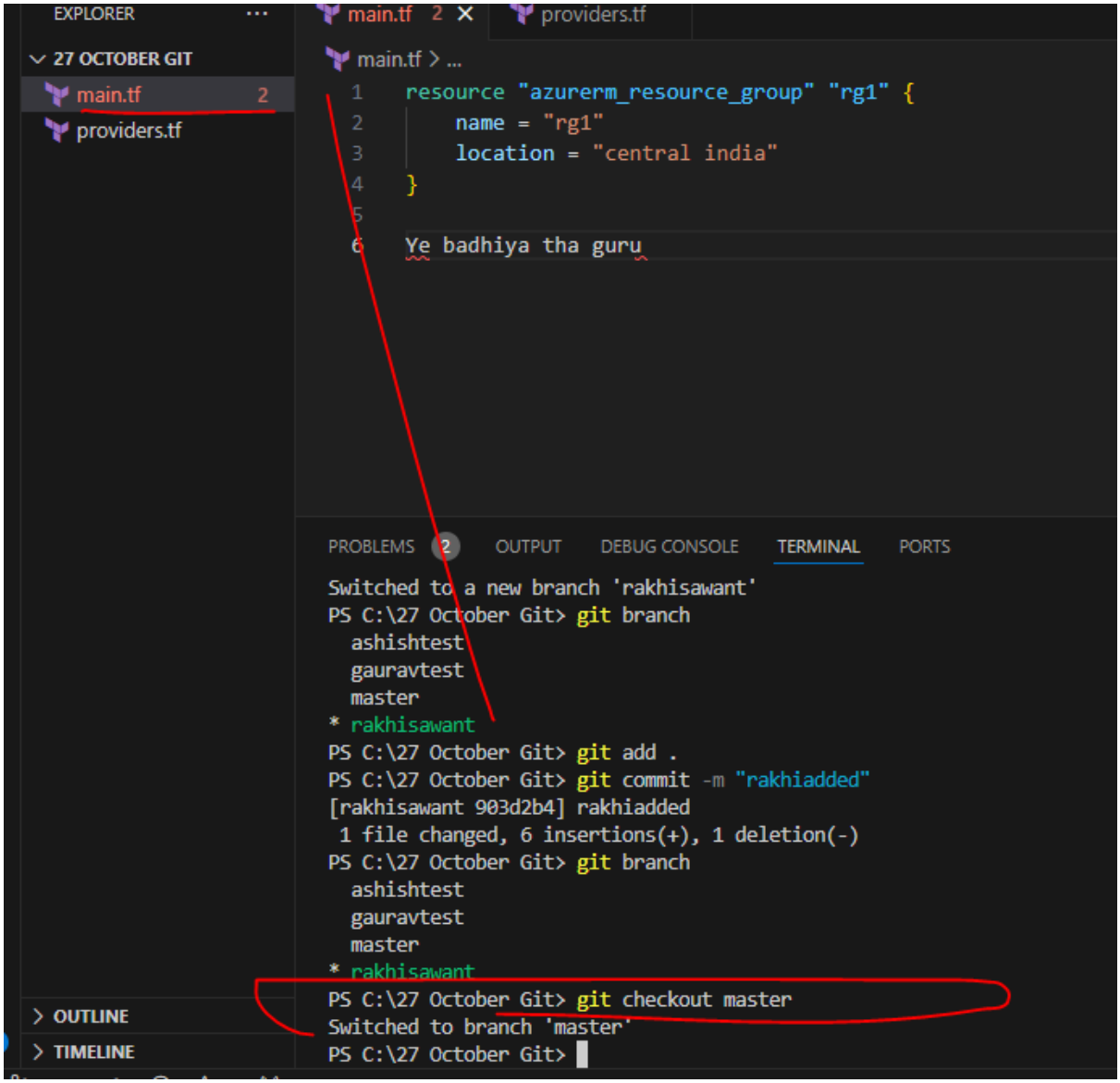


10) **git add .**

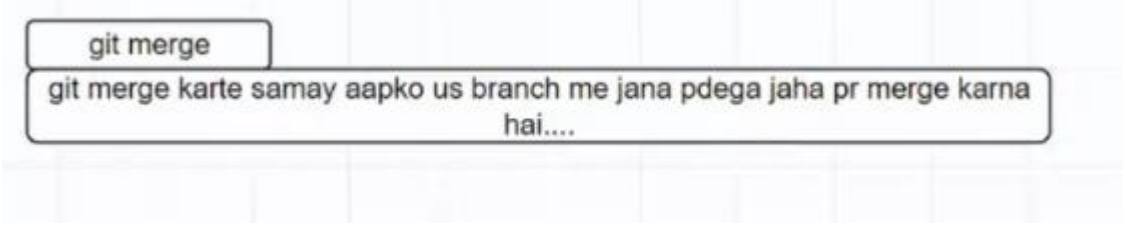
git commit -m "rakhiadded"

```
PS C:\27 October Git> git add .
PS C:\27 October Git> git commit -m "rakhiadded"
[rakhisawant 903d2b4] rakhiadded
 1 file changed, 6 insertions(+), 1 deletion(-)
PS C:\27 October Git>
```

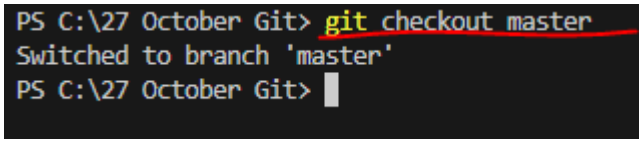
11) **git checkout master** – switched to master branch shows the rg added is not there



12) Now merging "**rakhisawant**" branch to "**master**" branch.



git checkout master



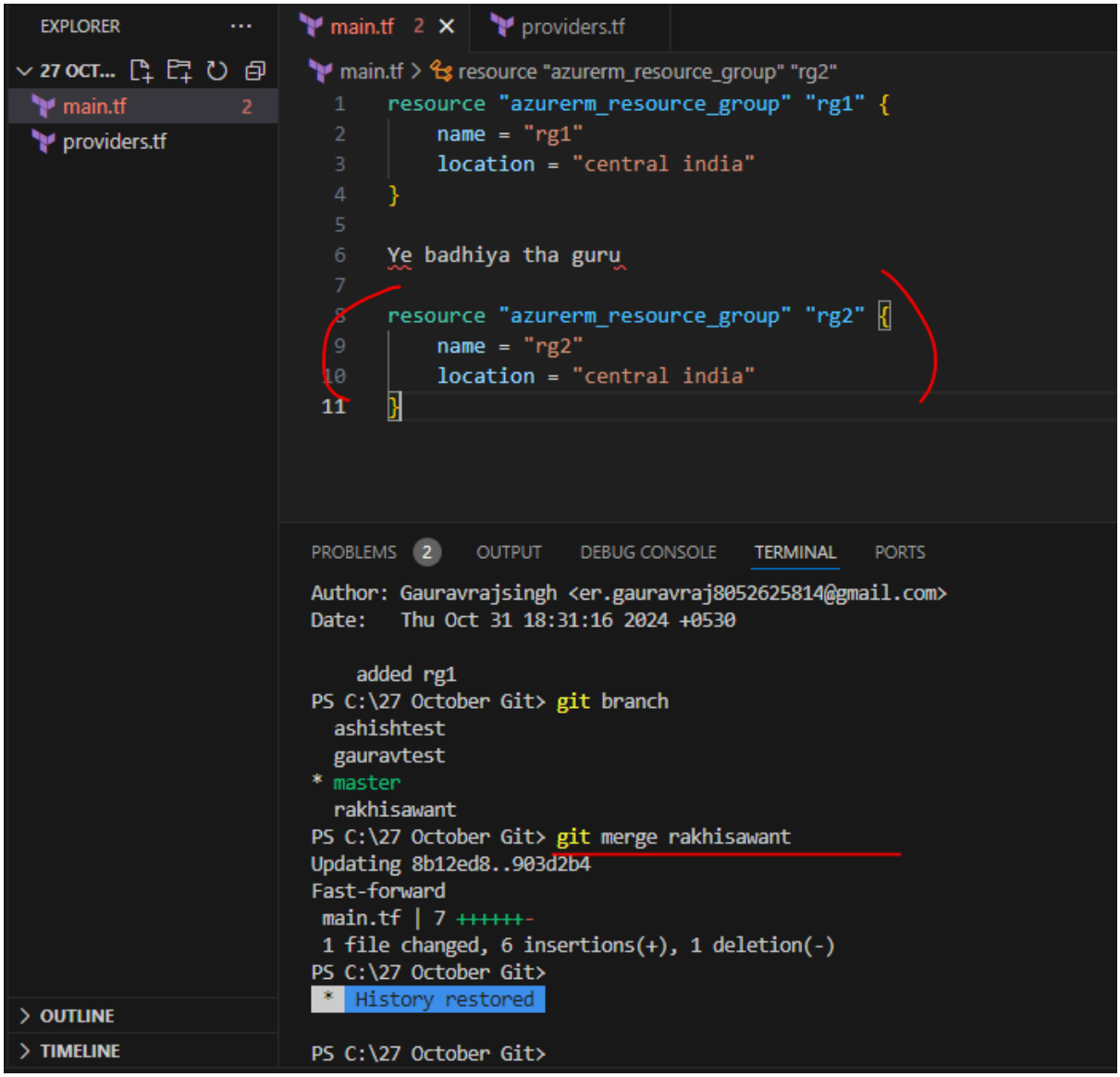
git log – shows we have 2 commits

```
PS C:\27 October Git> git log
commit 8b12ed8d5ad37d0dd3d9a97628a91f9ebacfd03e (HEAD -> master, gauravtest, ashishtest)
Author: Gauravrajsingh <er.gauravraj8052625814@gmail.com>
Date:   Thu Oct 31 18:36:51 2024 +0530

    guru added

commit 78a9117b5bc1933d859403a219a3bcbddca2e2e6
Author: Gauravrajsingh <er.gauravraj8052625814@gmail.com>
Date:   Thu Oct 31 18:31:16 2024 +0530

    added rg1
PS C:\27 October Git>
```

git merge rakhisawant

```
PS C:\27 October Git> git merge rakhisawant
Updating 8b12ed8..903d2b4
Fast-forward
 main.tf | 7 ++++++-
 1 file changed, 6 insertions(+), 1 deletion(-)
PS C:\27 October Git>
```

13) So we can see code got merged in master branch also


EXPLORER
27 OCT...
main.tf 2
providers.tf
main.tf 2
providers.tf
main.tf > resource "azurerm_resource_group" "rg2"
1 resource "azurerm_resource_group" "rg1" {
2 name = "rg1"
3 location = "central india"
4 }
5
6 Ye badhiya tha guru
7
8 resource "azurerm_resource_group" "rg2" {
9 name = "rg2"
10 location = "central india"
11 }
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Author: Gauravrajsingh <er.gauravraj8052625814@gmail.com>
Date: Thu Oct 31 18:31:16 2024 +0530
added rg1
PS C:\27 October Git> git branch
ashishtest
gauravtest
* master
rakhisawant
PS C:\27 October Git> git merge rakhisawant
Updating 8b12ed8..903d2b4
Fast-forward
main.tf | 7 ++++++-
1 file changed, 6 insertions(+), 1 deletion(-)
PS C:\27 October Git>
* History restored
PS C:\27 October Git>
OUTLINE
TIMELINE