LAB PROGRAM 1:

Setting Up and Basic Commands Initialize a new Git repository in a directory.

1) Create a new file and add it to the staging area and commit the changes with an appropriate commit message.

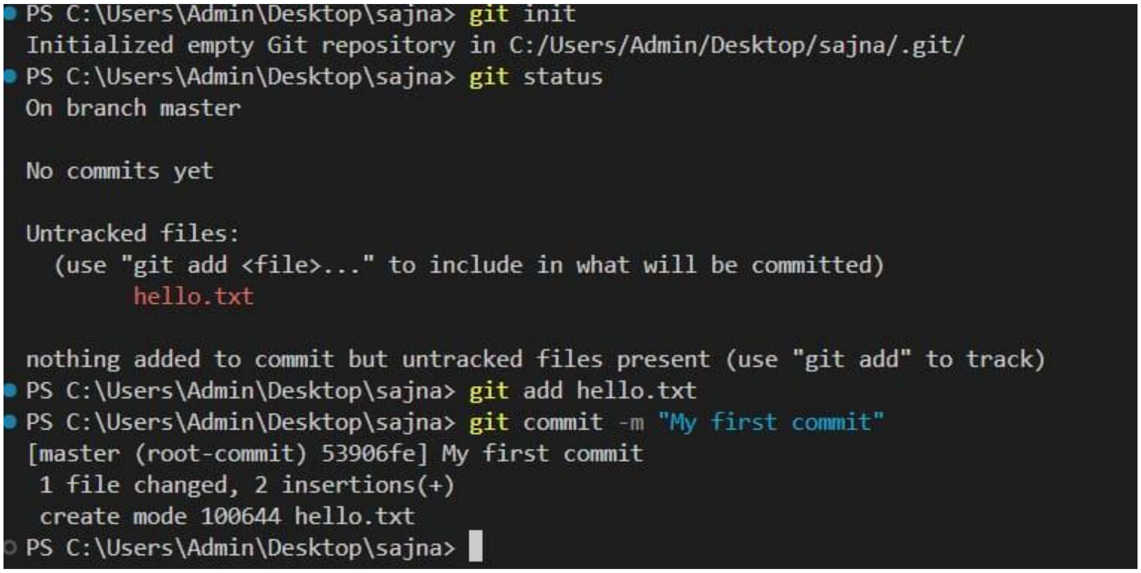
In VScode

Step1: Create a new folder(ex: git_practice)

Step2: create textfile in the newly created folder(ex:hello.txt)

Step3: in new terminal

- git
- git status
- git add hello.txt
- git commit -m “My first commit”



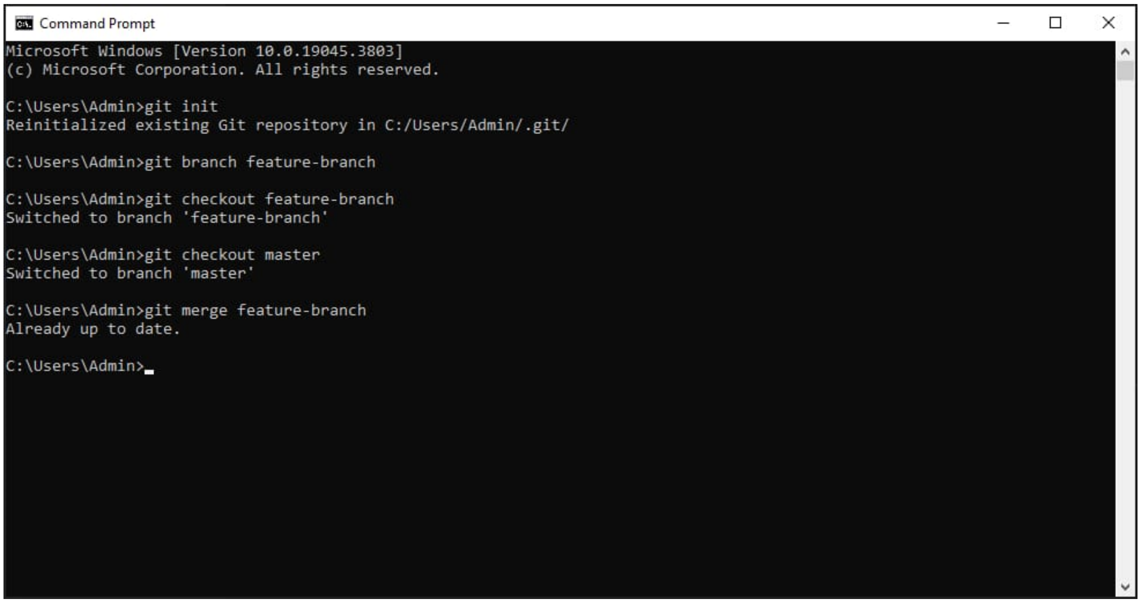
```
PS C:\Users\Admin\Desktop\sajna> git init
Initialized empty Git repository in C:/Users/Admin/Desktop/sajna/.git/
PS C:\Users\Admin\Desktop\sajna> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Admin\Desktop\sajna> git add hello.txt
PS C:\Users\Admin\Desktop\sajna> git commit -m "My first commit"
[master (root-commit) 53906fe] My first commit
 1 file changed, 2 insertions(+)
 create mode 100644 hello.txt
PS C:\Users\Admin\Desktop\sajna> 
```

LAB PROGRAM 2:

Creating and Managing Branches

2)Create a new branch named "feature-branch." Switch to the "master" branch. Merge the "feature-branch" into "master."

- git status
- git branch feature-branch (*creating new branch*)
- git checkout feature-branch (*switching to feature branch*)
- git status (*checking the current branch*)
- git checkout master (*switching to master branch*)
- git merge feature-branch (*merging branches*)



```
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>git init
Reinitialized existing Git repository in C:/Users/Admin/.git/

C:\Users\Admin>git branch feature-branch

C:\Users\Admin>git checkout feature-branch
Switched to branch 'feature-branch'

C:\Users\Admin>git checkout master
Switched to branch 'master'

C:\Users\Admin>git merge feature-branch
Already up to date.

C:\Users\Admin>
```

LAB PROGRAM 3:

Creating and Managing Branches

2) Write the commands to stash your changes, switch branches, and then apply the stashed changes.

The two commands used In this program are 1] git stash

2]git stash apply

VS code:-

New folder(ex:stash)

Two new files(ex:he.py , hello.py)

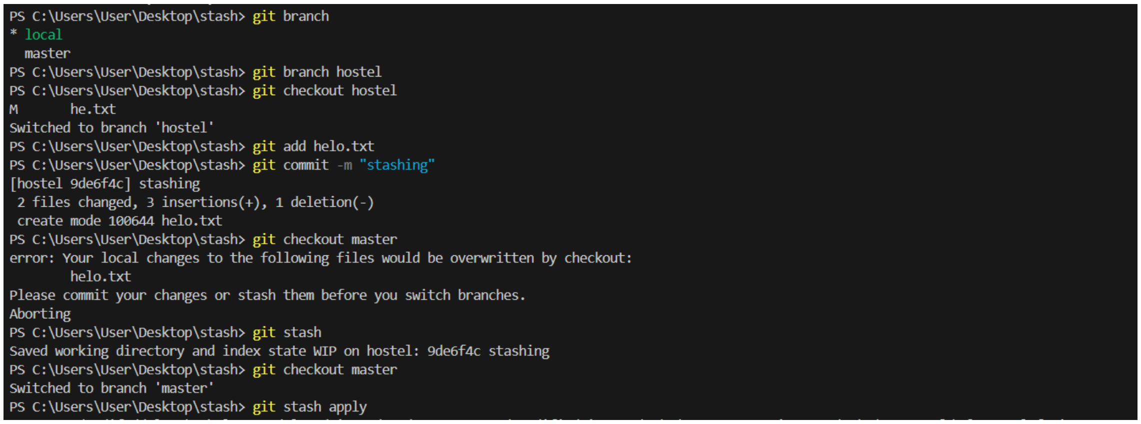
New terminal:-

- git branch *(checking branch)*
- git branch local (*creating new branch*)
- git checkout local (*switching to branch local*)
- git add hello.py
- git commit -m “stashing”
- (updating the code)
- git checkout master (*switching to master*)

(can’t switch to master)

(error is found because of changes)

- git stash
- git checkout master
- git stash apply



```
PS C:\Users\User\Desktop\stash> git branch
* local
  master
PS C:\Users\User\Desktop\stash> git branch hostel
PS C:\Users\User\Desktop\stash> git checkout hostel
M       he.txt
Switched to branch 'hostel'
PS C:\Users\User\Desktop\stash> git add helo.txt
PS C:\Users\User\Desktop\stash> git commit -m "stashing"
[hostel 9de6f4c] stashing
 2 files changed, 3 insertions(+), 1 deletion(-)
 create mode 100644 helo.txt
PS C:\Users\User\Desktop\stash> git checkout master
error: Your local changes to the following files would be overwritten by checkout:
        helo.txt
Please commit your changes or stash them before you switch branches.
Aborting
PS C:\Users\User\Desktop\stash> git stash
Saved working directory and index state WIP on hostel: 9de6f4c stashing
PS C:\Users\User\Desktop\stash> git checkout master
Switched to branch 'master'
PS C:\Users\User\Desktop\stash> git stash apply
```

LAB PROGRAM 4:

4)Collaboration and Remote Repositories Clone a remote Git repository to your local machine.

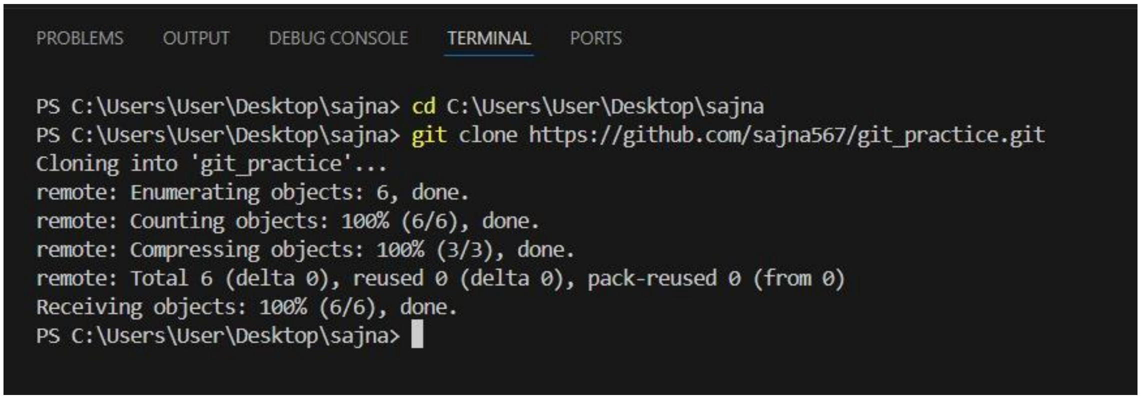
Step1: Upload a file to GitHub

Step2: To download any GitHub file to system

Step3: copy the path address from file holder of the new folder or destination

- cd (paste the file path address)
- git clone (paste URL of GitHub file)

step4: check the folder for files



```
PS C:\Users\User\Desktop\sajna> cd C:\Users\User\Desktop\sajna
PS C:\Users\User\Desktop\sajna> git clone https://github.com/sajna567/git_practice.git
Cloning into 'git_practice'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.
PS C:\Users\User\Desktop\sajna>
```

LAB PROGRAM 5:

Collaboration and Remote

5)Repositories Fetch the latest changes from a remote repository and rebase your local branch onto the updated remote branch.

Two commands

→git fetch (GitHub path URL)

→git rebase branch-name

VS code:-

Step1: new file(ex:file.txt)

- git init
- git add file1.txt
- git commit -m “c1”

(modify code or txt)

(again add and commit c2)

- git add file1.txt
- git commit -m “c2”

(create a new branch)

- git branch dev
- git checkout dev

(modify code or txt)

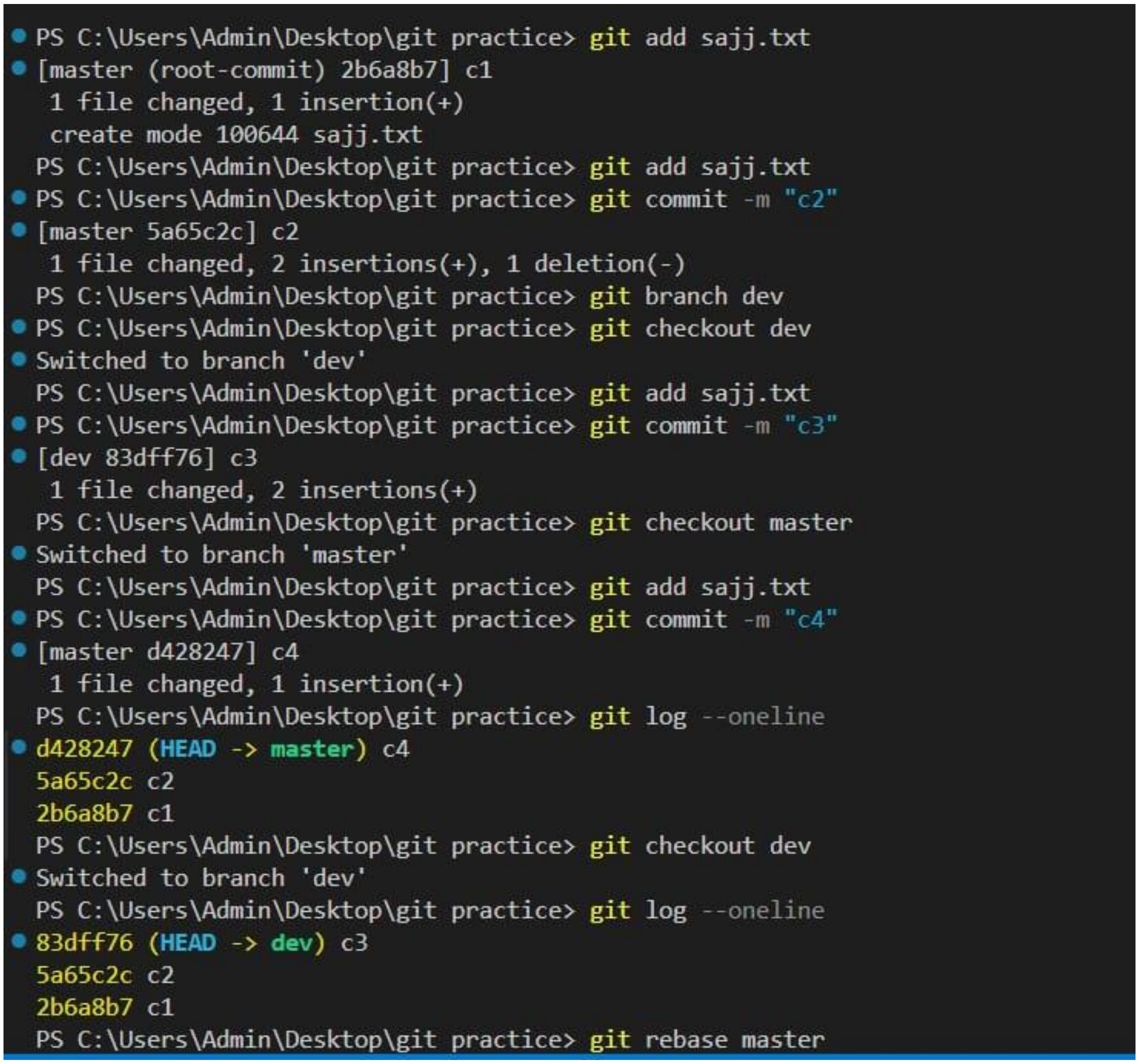
(again add and commit c3)

- git add file1.txt
- git commit -m “c3”
- git checkout master

(modify the code)

(add and commit again c4)

- git add file1.txt
- git commit -m “c4”
- git log - -oneline
- git rebase master



```
PS C:\Users\Admin\Desktop\git practice> git add sajj.txt
[master (root-commit) 2b6a8b7] c1
 1 file changed, 1 insertion(+)
 create mode 100644 sajj.txt
PS C:\Users\Admin\Desktop\git practice> git add sajj.txt
PS C:\Users\Admin\Desktop\git practice> git commit -m "c2"
[master 5a65c2c] c2
 1 file changed, 2 insertions(+), 1 deletion(-)
PS C:\Users\Admin\Desktop\git practice> git branch dev
PS C:\Users\Admin\Desktop\git practice> git checkout dev
Switched to branch 'dev'
PS C:\Users\Admin\Desktop\git practice> git add sajj.txt
PS C:\Users\Admin\Desktop\git practice> git commit -m "c3"
[dev 83dff76] c3
 1 file changed, 2 insertions(+)
PS C:\Users\Admin\Desktop\git practice> git checkout master
Switched to branch 'master'
PS C:\Users\Admin\Desktop\git practice> git add sajj.txt
PS C:\Users\Admin\Desktop\git practice> git commit -m "c4"
[master d428247] c4
 1 file changed, 1 insertion(+)
PS C:\Users\Admin\Desktop\git practice> git log --oneline
d428247 (HEAD -> master) c4
5a65c2c c2
2b6a8b7 c1
PS C:\Users\Admin\Desktop\git practice> git checkout dev
Switched to branch 'dev'
PS C:\Users\Admin\Desktop\git practice> git log --oneline
83dff76 (HEAD -> dev) c3
5a65c2c c2
2b6a8b7 c1
PS C:\Users\Admin\Desktop\git practice> git rebase master
```

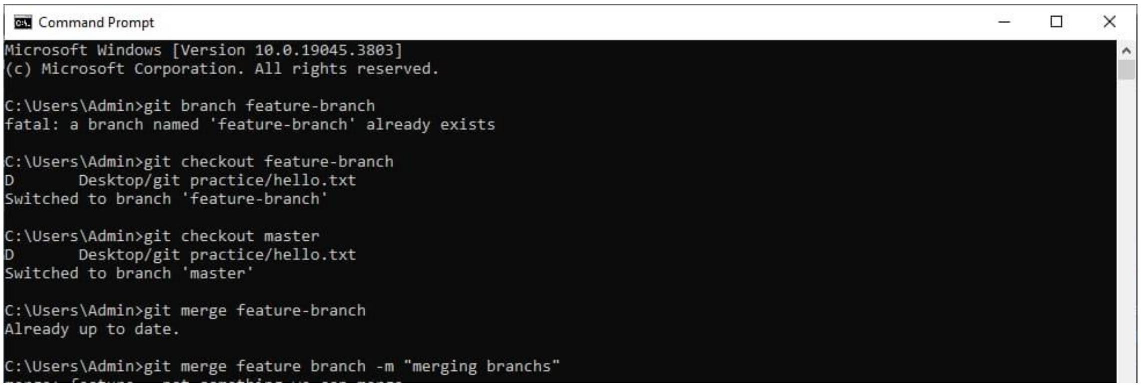
LAB PROGRAM 6:

6)Collaboration and Remote Repositories Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge.

Command →git merge feature branch

→git merge feature branch -m “   “

- ➢ git branch feature-branch
- ➢ git checkout feature branch
- ➢ git checkout master
- ➢ git merge feature-branch
- ➢ git merge feature branch -m “merging branches”



```
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>git branch feature-branch
fatal: a branch named 'feature-branch' already exists

C:\Users\Admin>git checkout feature-branch
D       Desktop/git practice/hello.txt
Switched to branch 'feature-branch'

C:\Users\Admin>git checkout master
D       Desktop/git practice/hello.txt
Switched to branch 'master'

C:\Users\Admin>git merge feature-branch
Already up to date.

C:\Users\Admin>git merge feature branch -m "merging branchs"
```

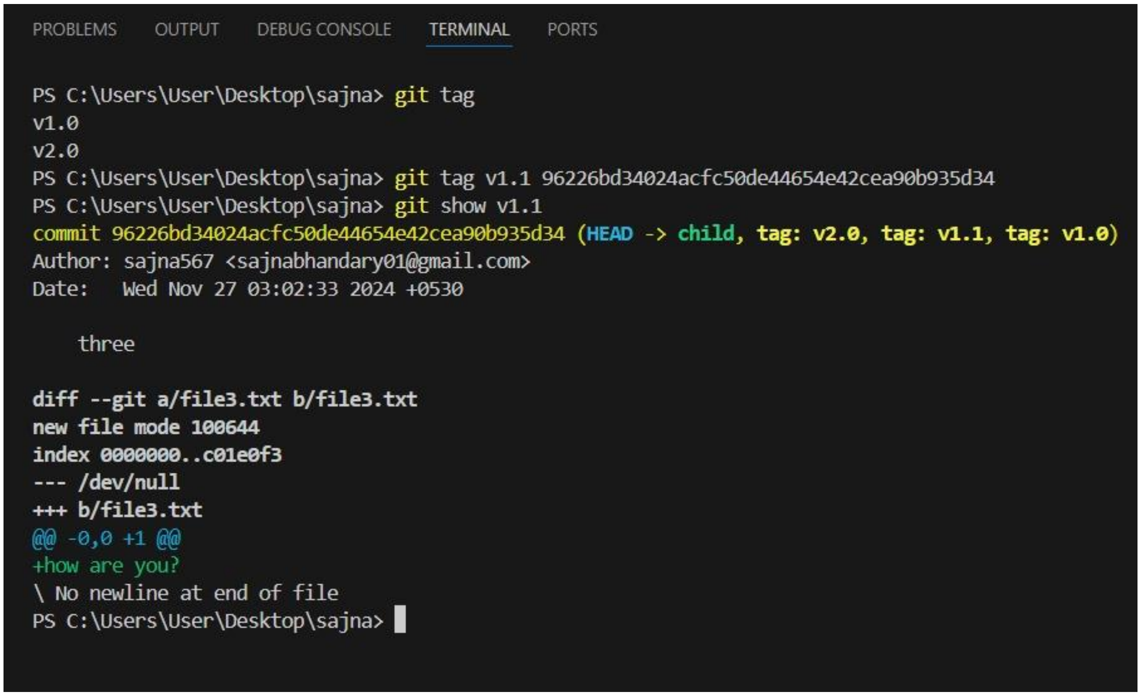
LAB PROGRAM 7:

Git Tags and Releases

7)Write the command to create a lightweight Git tag named "v1.0" for a commit in your local repository.

creating tag:

- git tag
- git tag v1.0
- git tag
- git tag v1.1
- git tag
  [creating tags for specific commit ID]
- git log
  (copy commit id)
- git tag v1.2 (paste commit id)
- git show v1.2
  (displaying commit ID of v1.2)



```
PS C:\Users\User\Desktop\sajna> git tag
v1.0
v2.0
PS C:\Users\User\Desktop\sajna> git tag v1.1 96226bd34024acfc50de44654e42cea90b935d34
PS C:\Users\User\Desktop\sajna> git show v1.1
commit 96226bd34024acfc50de44654e42cea90b935d34 (HEAD -> child, tag: v2.0, tag: v1.1, tag: v1.0)
Author: sajna567 <sajnabhandary01@gmail.com>
Date:   Wed Nov 27 03:02:33 2024 +0530

    three

diff --git a/file3.txt b/file3.txt
new file mode 100644
index 0000000..c01e0f3
--- /dev/null
+++ b/file3.txt
@@ -0,0 +1 @@
+how are you?
\ No newline at end of file
PS C:\Users\User\Desktop\sajna> 
```

LAB PROGRAM 8:

Advanced Git Operations

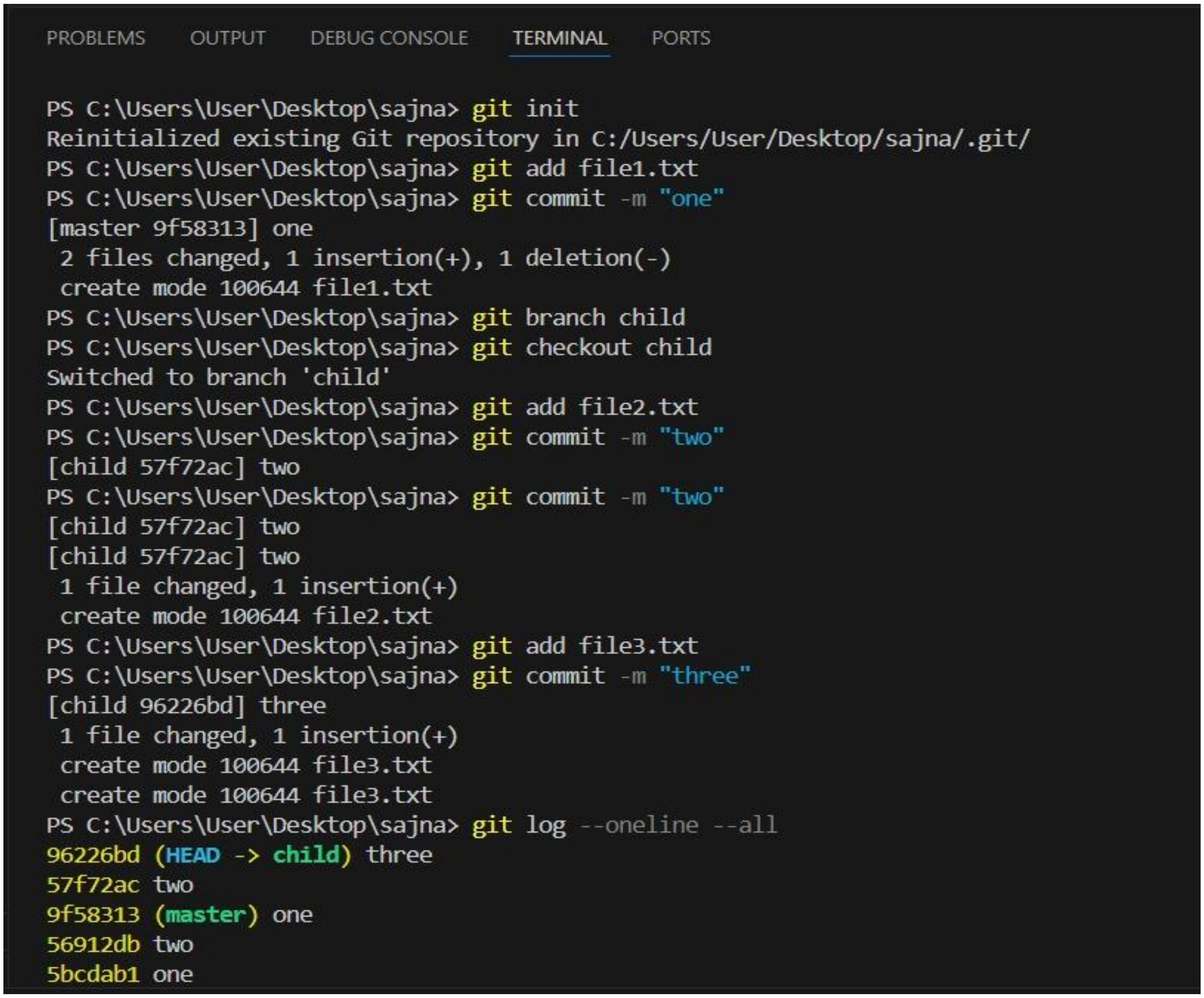
8)Write the command to cherry-pick a range of commits from "source-branch" to the current branch.

Vs code

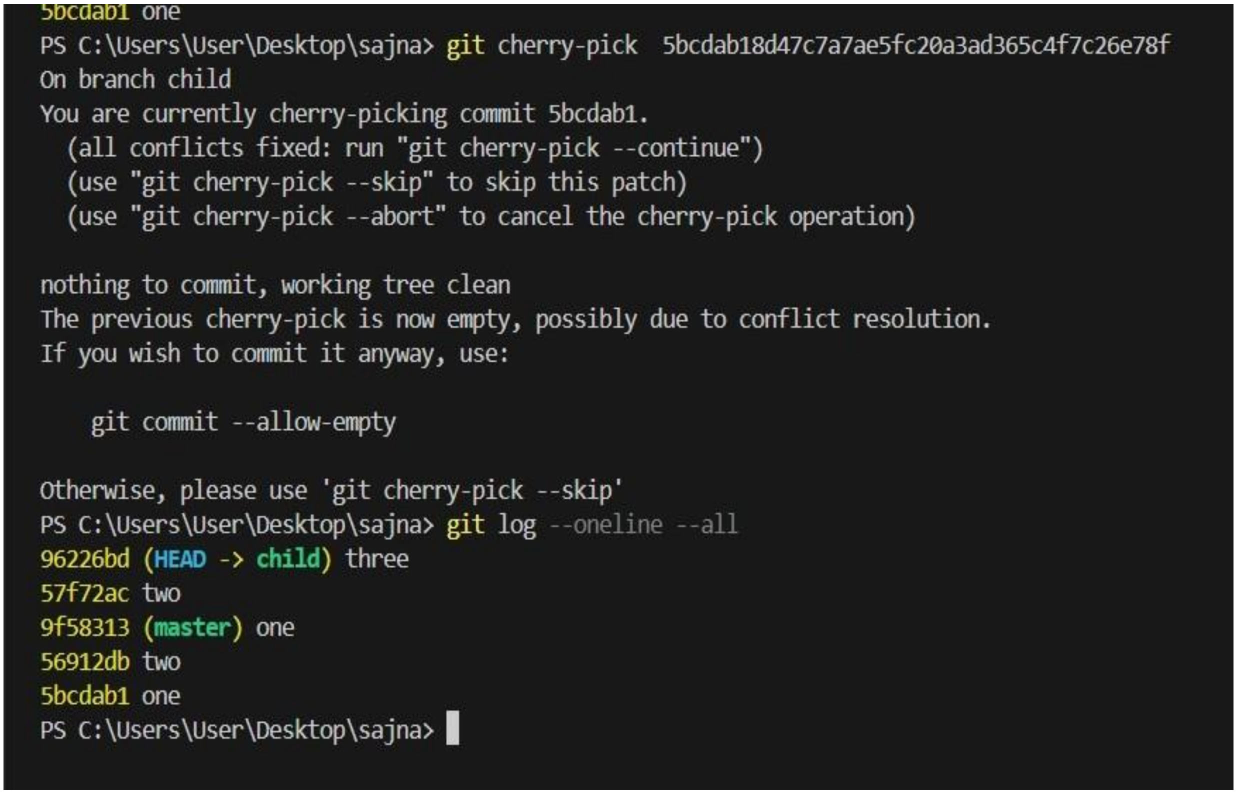
Create three file in a folder(file1.txt,file2.txt,file3.txt)

New terminal

- git init
- git add file1.txt
- git commit -m “one”
  (creating a new branch)
- git branch child
- git checkout child
  [Adding and committing file2, file3 to child branch]
- git add file2.txt
- git commit -m”two”
- git add file3.txt
- git commit -m “three”
  [checking files with branch]
- git log - -oneline - -all
- git cherry-pick (ID of any file)
- git log - -oneline - -all



```
PS C:\Users\User\Desktop\sajna> git init
Reinitialized existing Git repository in C:/Users/User/Desktop/sajna/.git/
PS C:\Users\User\Desktop\sajna> git add file1.txt
PS C:\Users\User\Desktop\sajna> git commit -m "one"
[master 9f58313] one
 2 files changed, 1 insertion(+), 1 deletion(-)
 create mode 100644 file1.txt
PS C:\Users\User\Desktop\sajna> git branch child
PS C:\Users\User\Desktop\sajna> git checkout child
Switched to branch 'child'
PS C:\Users\User\Desktop\sajna> git add file2.txt
PS C:\Users\User\Desktop\sajna> git commit -m "two"
[child 57f72ac] two
PS C:\Users\User\Desktop\sajna> git commit -m "two"
[child 57f72ac] two
[child 57f72ac] two
 1 file changed, 1 insertion(+)
 create mode 100644 file2.txt
PS C:\Users\User\Desktop\sajna> git add file3.txt
PS C:\Users\User\Desktop\sajna> git commit -m "three"
[child 96226bd] three
 1 file changed, 1 insertion(+)
 create mode 100644 file3.txt
 create mode 100644 file3.txt
PS C:\Users\User\Desktop\sajna> git log --oneline --all
96226bd (HEAD -> child) three
57f72ac two
9f58313 (master) one
56912db two
5bcdab1 one
```



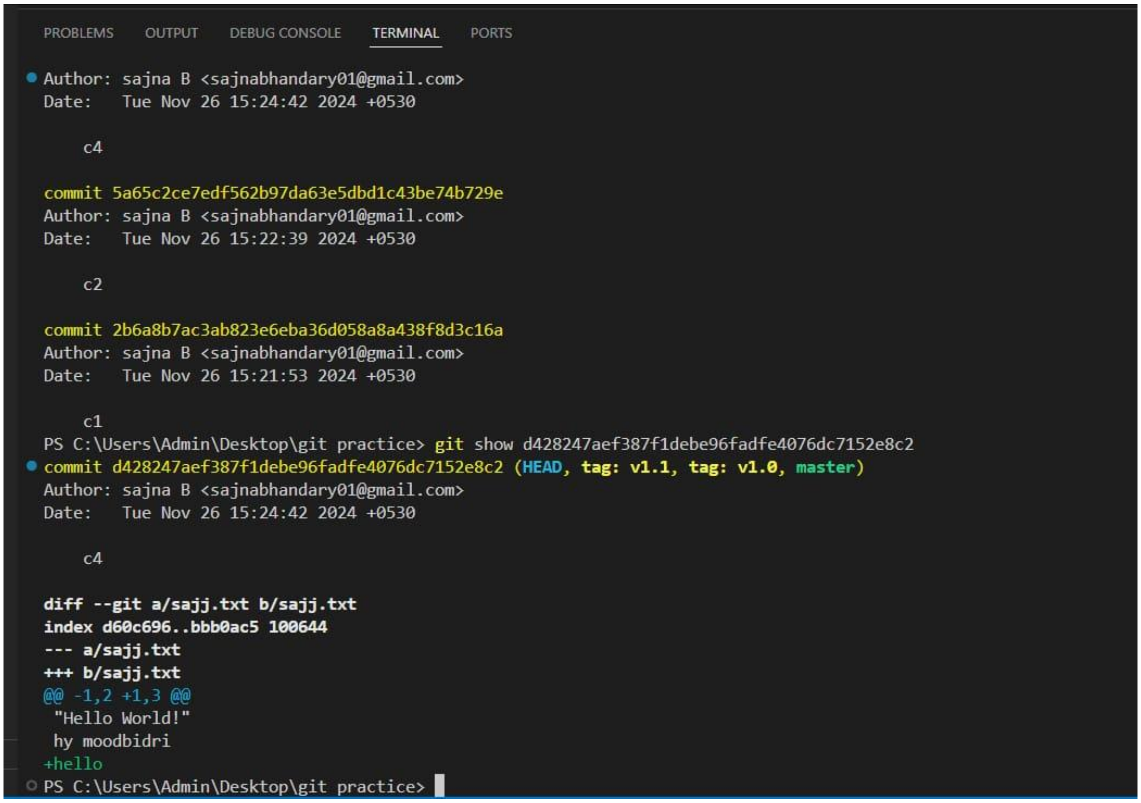
```
5bcdab1 one
PS C:\Users\User\Desktop\sajna> git cherry-pick  5bcdab18d47c7a7ae5fc20a3ad365c4f7c26e78f
On branch child
You are currently cherry-picking commit 5bcdab1.
  (all conflicts fixed: run "git cherry-pick --continue")
  (use "git cherry-pick --skip" to skip this patch)
  (use "git cherry-pick --abort" to cancel the cherry-pick operation)

nothing to commit, working tree clean
The previous cherry-pick is now empty, possibly due to conflict resolution.
If you wish to commit it anyway, use:

    git commit --allow-empty

Otherwise, please use 'git cherry-pick --skip'
PS C:\Users\User\Desktop\sajna> git log --oneline --all
96226bd (HEAD -> child) three
57f72ac two
9f58313 (master) one
56912db two
5bcdab1 one
PS C:\Users\User\Desktop\sajna> ▌
```

LAB PROGRAM 9:

Analysing and Changing Git History

9) Given a commit ID, how would you use Git to view the details of that specific commit, including the author, date, and commit message?

- ➢ git log (*to get commit history*)
  (copy commit id)
- ➢ git show (paste the id)
  (able to get details of commit ID)

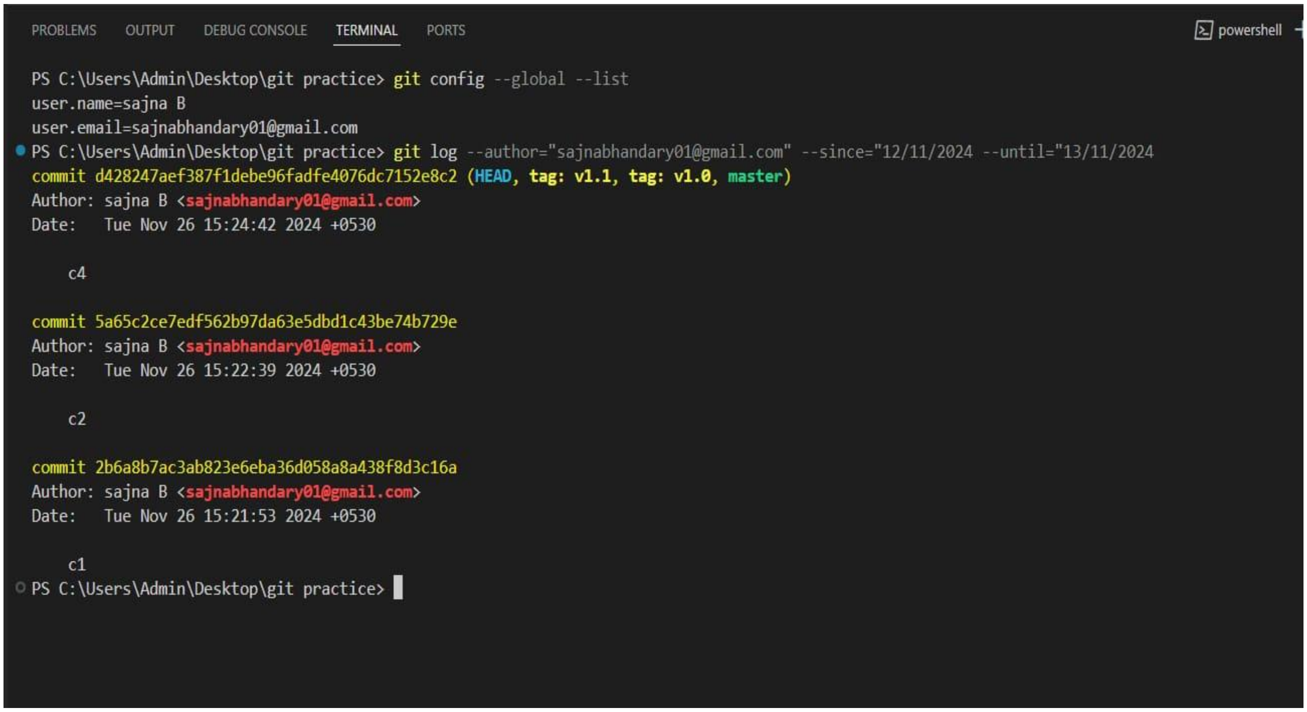


```
Author: sajna B <sajnabhandary01@gmail.com>
Date:   Tue Nov 26 15:24:42 2024 +0530

    c4

commit 5a65c2ce7edf562b97da63e5dbd1c43be74b729e
Author: sajna B <sajnabhandary01@gmail.com>
Date:   Tue Nov 26 15:22:39 2024 +0530

    c2

commit 2b6a8b7ac3ab823e6eba36d058a8a438f8d3c16a
Author: sajna B <sajnabhandary01@gmail.com>
Date:   Tue Nov 26 15:21:53 2024 +0530

    c1
PS C:\Users\Admin\Desktop\git practice> git show d428247aef387f1debe96fadfe4076dc7152e8c2
commit d428247aef387f1debe96fadfe4076dc7152e8c2 (HEAD, tag: v1.1, tag: v1.0, master)
Author: sajna B <sajnabhandary01@gmail.com>
Date:   Tue Nov 26 15:24:42 2024 +0530

    c4

diff --git a/sajj.txt b/sajj.txt
index d60c696..bbb0ac5 100644
--- a/sajj.txt
+++ b/sajj.txt
@@ -1,2 +1,3 @@
 "Hello World!"
 hy moodbidri
+hello
PS C:\Users\Admin\Desktop\git practice>
```

LAB PROGRAM 10:

10)Analysing and Changing Git History Write the command to list all commits made by the author "John Doe" between "2023-01-01" and "2023-12-31."

- git config - -global - -list (*to get user details*)
- git log - -author=Ram123@gmail.com - -since= ”(*starting date*)” - -until=”(*End date*)”



```
PS C:\Users\Admin\Desktop\git practice> git config --global --list
user.name=sajna B
user.email=sajnabhandary01@gmail.com
PS C:\Users\Admin\Desktop\git practice> git log --author="sajnabhandary01@gmail.com" --since="12/11/2024 --until="13/11/2024
commit d428247aef387f1debe96fadfe4076dc7152e8c2 (HEAD, tag: v1.1, tag: v1.0, master)
Author: sajna B <sajnabhandary01@gmail.com>
Date:   Tue Nov 26 15:24:42 2024 +0530

    c4

commit 5a65c2ce7edf562b97da63e5dbd1c43be74b729e
Author: sajna B <sajnabhandary01@gmail.com>
Date:   Tue Nov 26 15:22:39 2024 +0530

    c2

commit 2b6a8b7ac3ab823e6eba36d058a8a438f8d3c16a
Author: sajna B <sajnabhandary01@gmail.com>
Date:   Tue Nov 26 15:21:53 2024 +0530

    c1
PS C:\Users\Admin\Desktop\git practice>
```

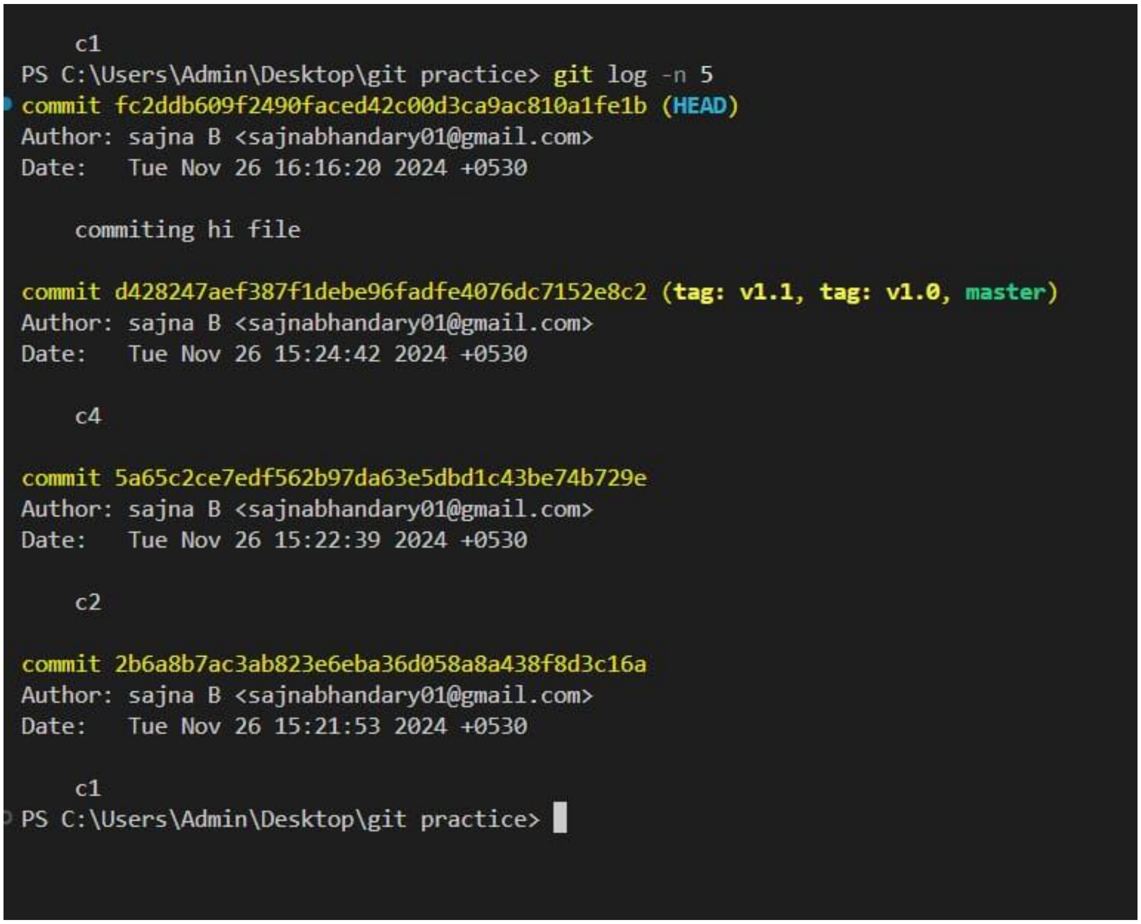
LAB PROGRAM 11:

Analysing and Changing Git History

11)Write the command to display the last five commits in the repository's history.

(To see some specific history of commit)

- git log -n 5



```
    c1
PS C:\Users\Admin\Desktop\git practice> git log -n 5
commit fc2ddb609f2490faced42c00d3ca9ac810a1fe1b (HEAD)
Author: sajna B <sajnabhandary01@gmail.com>
Date:   Tue Nov 26 16:16:20 2024 +0530

    commiting hi file

commit d428247aef387f1debe96fadfe4076dc7152e8c2 (tag: v1.1, tag: v1.0, master)
Author: sajna B <sajnabhandary01@gmail.com>
Date:   Tue Nov 26 15:24:42 2024 +0530

    c4

commit 5a65c2ce7edf562b97da63e5dbd1c43be74b729e
Author: sajna B <sajnabhandary01@gmail.com>
Date:   Tue Nov 26 15:22:39 2024 +0530

    c2

commit 2b6a8b7ac3ab823e6eba36d058a8a438f8d3c16a
Author: sajna B <sajnabhandary01@gmail.com>
Date:   Tue Nov 26 15:21:53 2024 +0530

    c1
PS C:\Users\Admin\Desktop\git practice>
```

LAB PROGRAM 12:

12) Analysing and Changing Git History Write the command to undo the changes introduced by the commit with the ID "abc123".

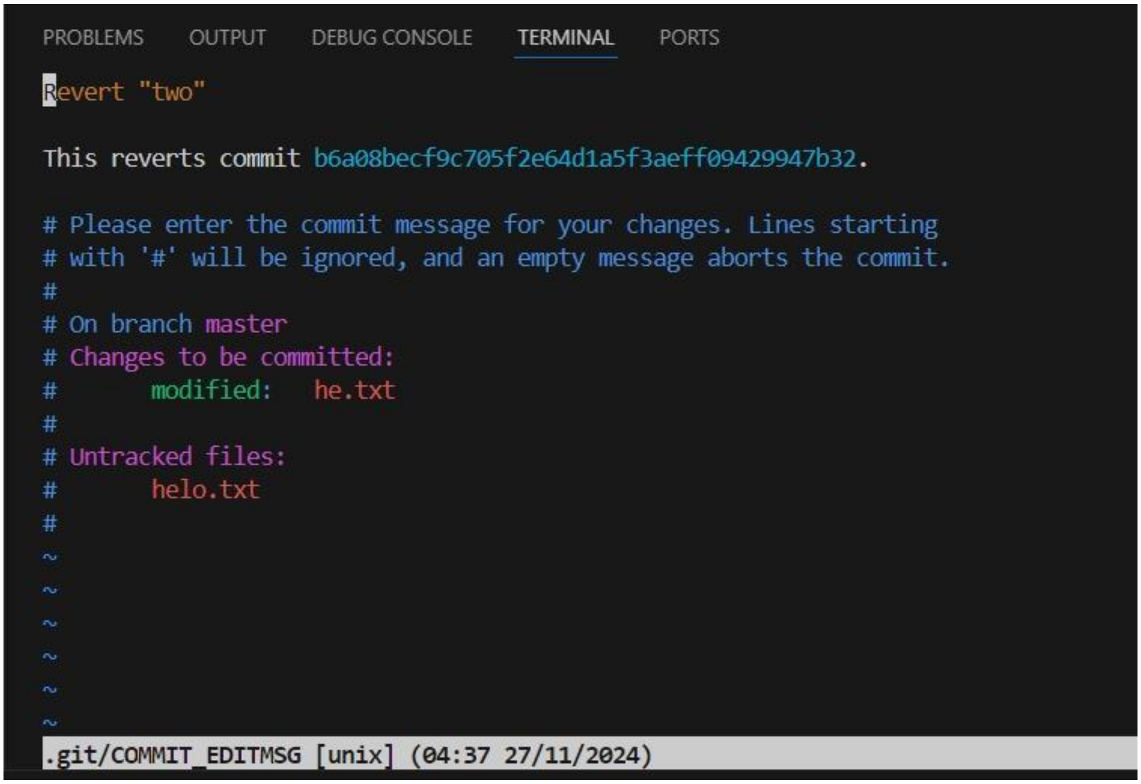
VS code

New folder = revert

New file = revert1.txt

New Terminal→

- git init
- git add revert1.txt
- git commit -m “one”
  (modify changes in the file, code)
- git add revert1.txt
- git commit -m “two”
- git log
  (copy ID of “two”)
  (clear terminal)
- git revert(paste copied ID)



```
Revert "two"

This reverts commit b6a08becf9c705f2e64d1a5f3aeff09429947b32.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Changes to be committed:
#       modified:   he.txt
#
# Untracked files:
#       helo.txt
#
~
~
~
~
~
~
.git/COMMIT_EDITMSG [unix] (04:37 27/11/2024)
```