

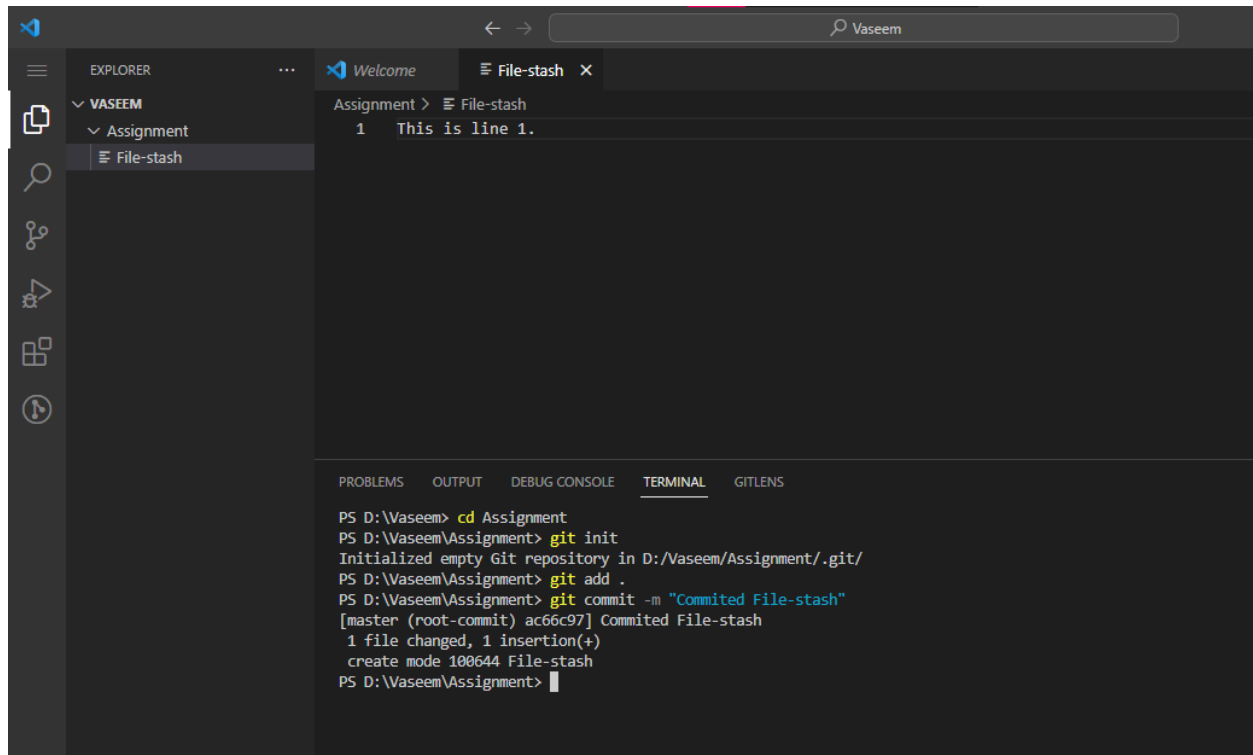
Q1. Describe the usage of the git stash command by using an example and also state the process by giving the screenshot of all the commands written in git bash.

git stash

Git stash saves the uncommitted changes locally, allowing you to make changes, switch branches, and perform other Git operations. You can then reapply the stashed changes when you need them.

First I have created a file named stashcommand and added content in it.

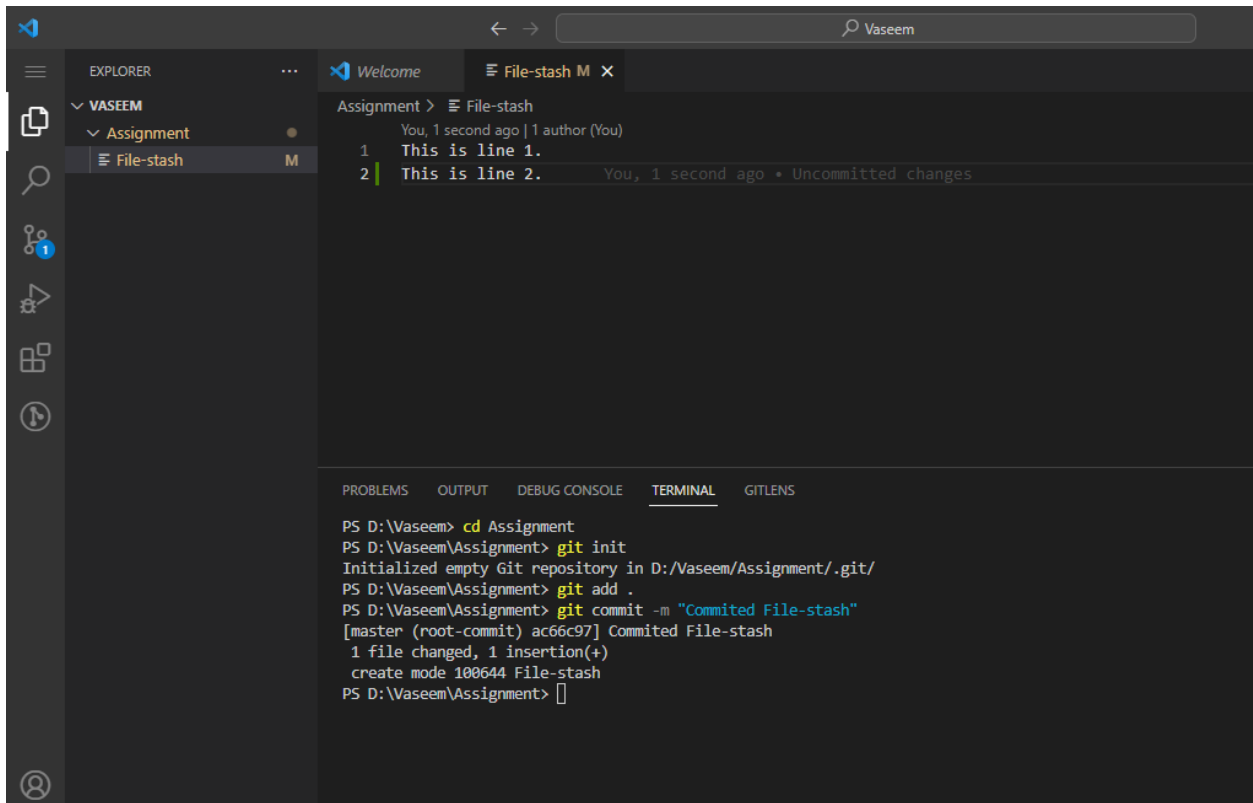
Now I have added it and committed it



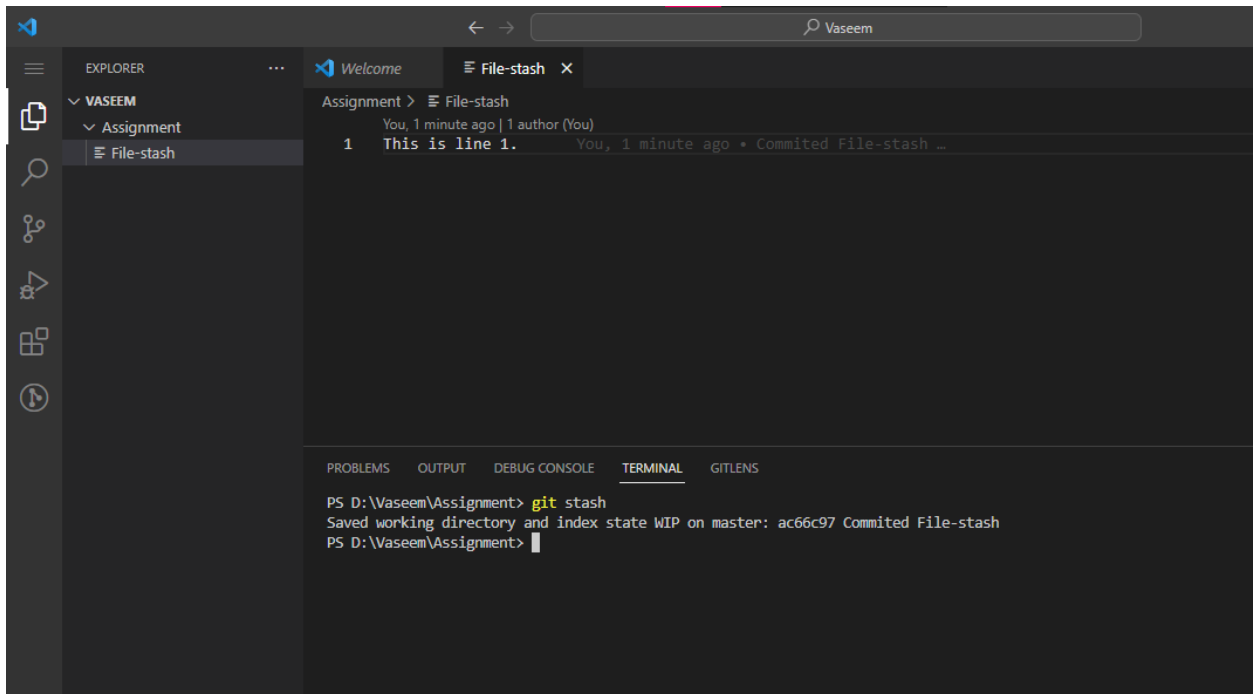
The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a project structure with 'VASEEM' as the root, containing 'Assignment' and 'File-stash'. The 'File-stash' file is selected. The main editor area shows the content of 'File-stash', which is '1 This is line 1.'. At the bottom, the Terminal pane is active, displaying the following commands and output:

```
PS D:\Vaseem> cd Assignment
PS D:\Vaseem\Assignment> git init
Initialized empty Git repository in D:\Vaseem\Assignment\.git/
PS D:\Vaseem\Assignment> git add .
PS D:\Vaseem\Assignment> git commit -m "Committed File-stash"
[master (root-commit) ac66c97] Committed File-stash
1 file changed, 1 insertion(+)
create mode 100644 File-stash
PS D:\Vaseem\Assignment> 
```

Now I have added another line in the file.

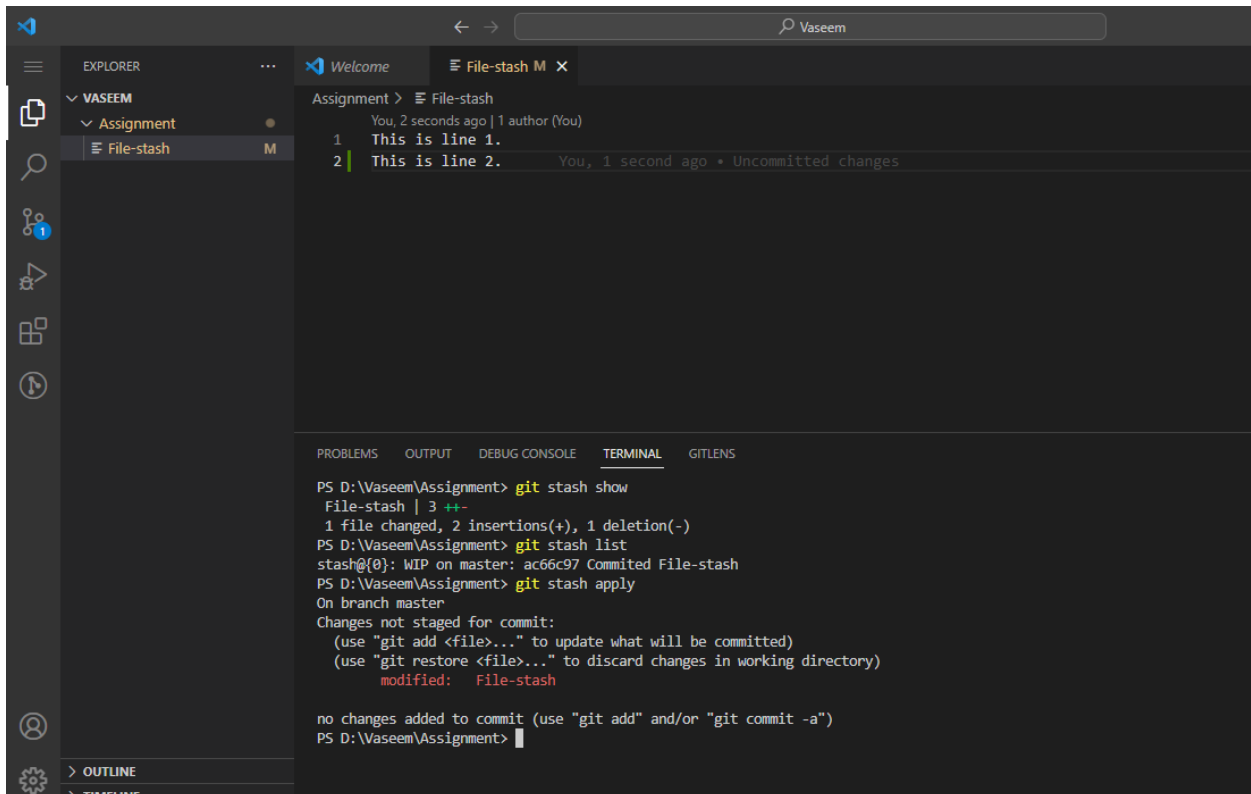


I have run the command **git stash** then the line which I have added as been stashed.



I have used the following commands related to stash

- 1) `git stash show` – It shows how many changes we have done on the files.
- 2) `git stash list` – It will show the how many changes we have stashed and their details.
- 3) `git stash apply` – It will show the change which we have stashed.



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a project named 'VASEEM' with a subfolder 'Assignment' containing a file 'File-stash'. The file is open in the editor, showing two lines of code: 'This is line 1.' and 'This is line 2.'. The terminal window at the bottom shows the following commands and output:

```
PS D:\Vaseem\Assignment> git stash show
File-stash | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)
PS D:\Vaseem\Assignment> git stash list
stash@{0}: WIP on master: ac66c97 Committed File-stash
PS D:\Vaseem\Assignment> git stash apply
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:   File-stash

no changes added to commit (use "git add" and/or "git commit -a")
PS D:\Vaseem\Assignment>
```

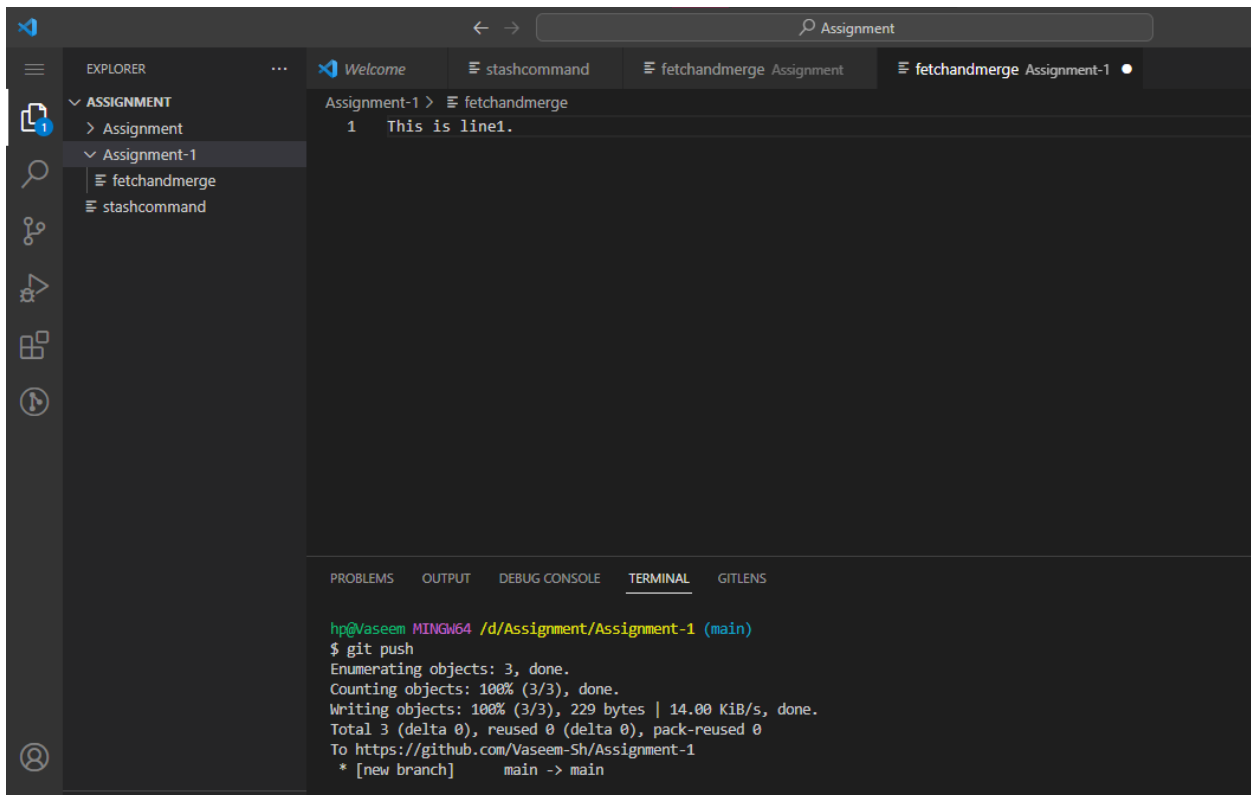
Q2. By using a sample example of your choice, use the git fetch command and also use the git merge command and describe the whole process through a screenshot with all the commands and their output in git bash.

git fetch

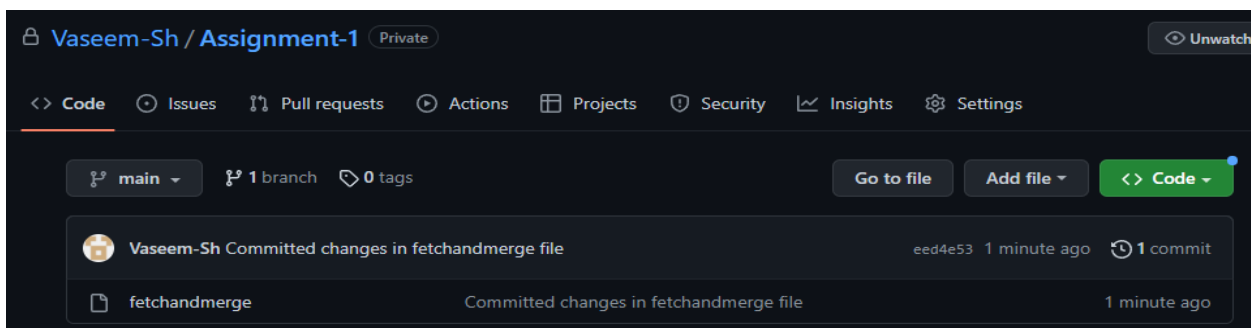
This command lets you fetch changes from the given remote repository.

It will notify you that changes are available to your remote repository but will not bring them to your local registry.

I have created a repository in the github and I have cloned it to the local machine. I have created a file (fetchandmerge) and added, committed and pushed into the repository.



```
hp@Vaseem MINGW64 /d/Assignment/Assignment-1 (main)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 229 bytes | 14.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Vaseem-Sh/Assignment-1
 * [new branch]      main -> main
```



```
Vaseem-Sh / Assignment-1 (Private) Unwatch

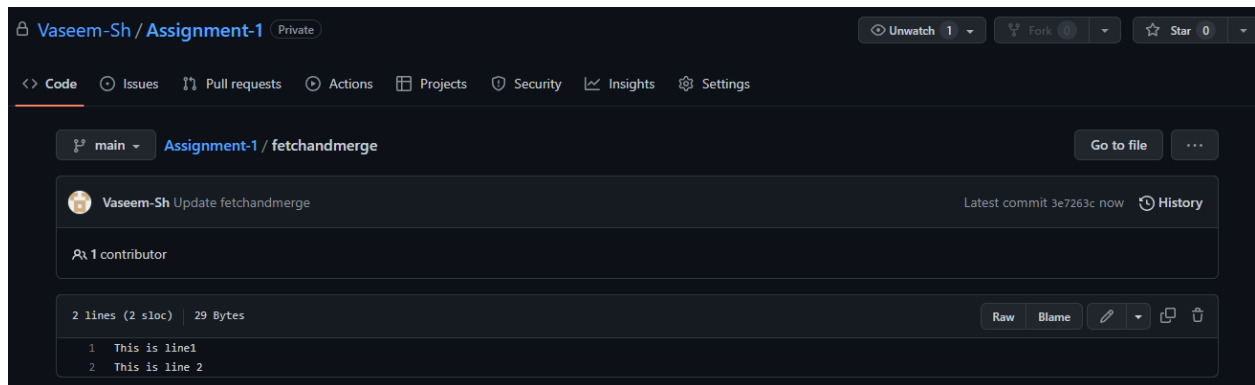
Code Issues Pull requests Actions Projects Security Insights Settings

main 1 branch 0 tags Go to file Add file <> Code

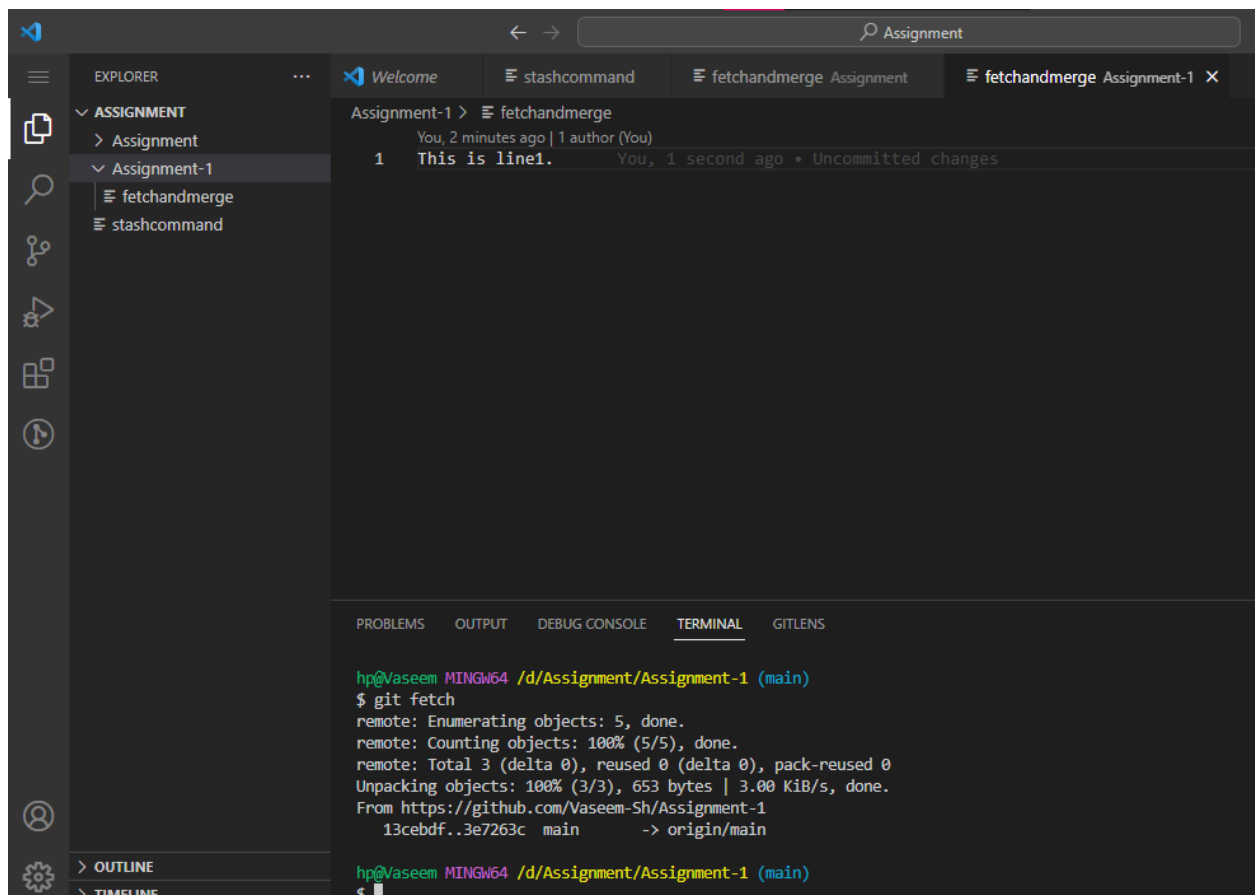
Vaseem-Sh Committed changes in fetchandmerge file eed4e53 1 minute ago 1 commit

fetchandmerge Committed changes in fetchandmerge file 1 minute ago
```

Now I have added a line into the file and committed the changes in the github website.



Now I have run the **git fetch** command then it showed the changes made in the remote repository.



Now I have executed the **git merge origin/<branch-name>** command then it showed the changes which are made in the remote repository.

```
hp@Vaseem MINGW64 /d/Assignment/Assignment-1 (main)
$ git merge origin/main
Updating 13cebd..3e7263c
Fast-forward
 fetchandmerge | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

hp@Vaseem MINGW64 /d/Assignment/Assignment-1 (main)
$
```

Q3. State the difference between git fetch and git pull by doing a practical example in your git bash and attach a screenshot of all the processes.

git fetch

This command lets you fetch changes from the given remote repository.

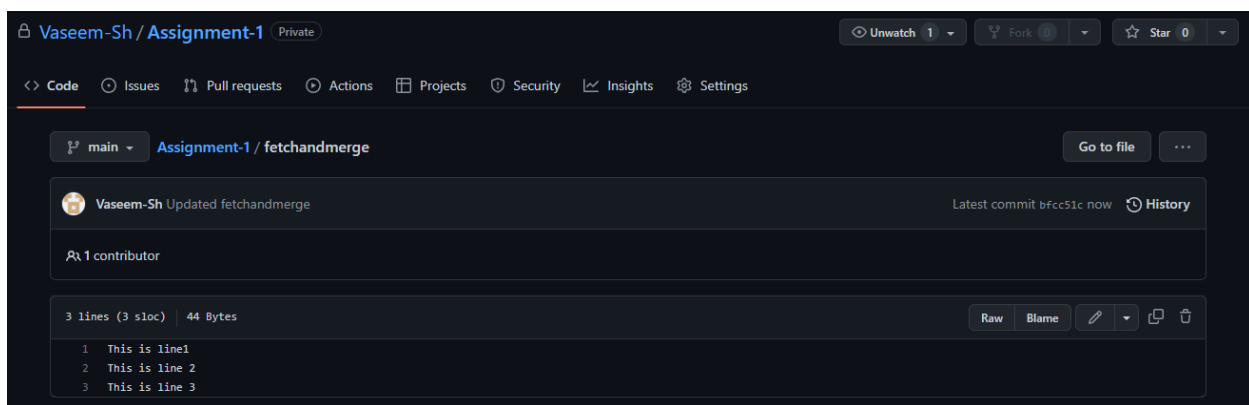
It will notify you that changes are available to your remote repository but will not bring them to your local registry.

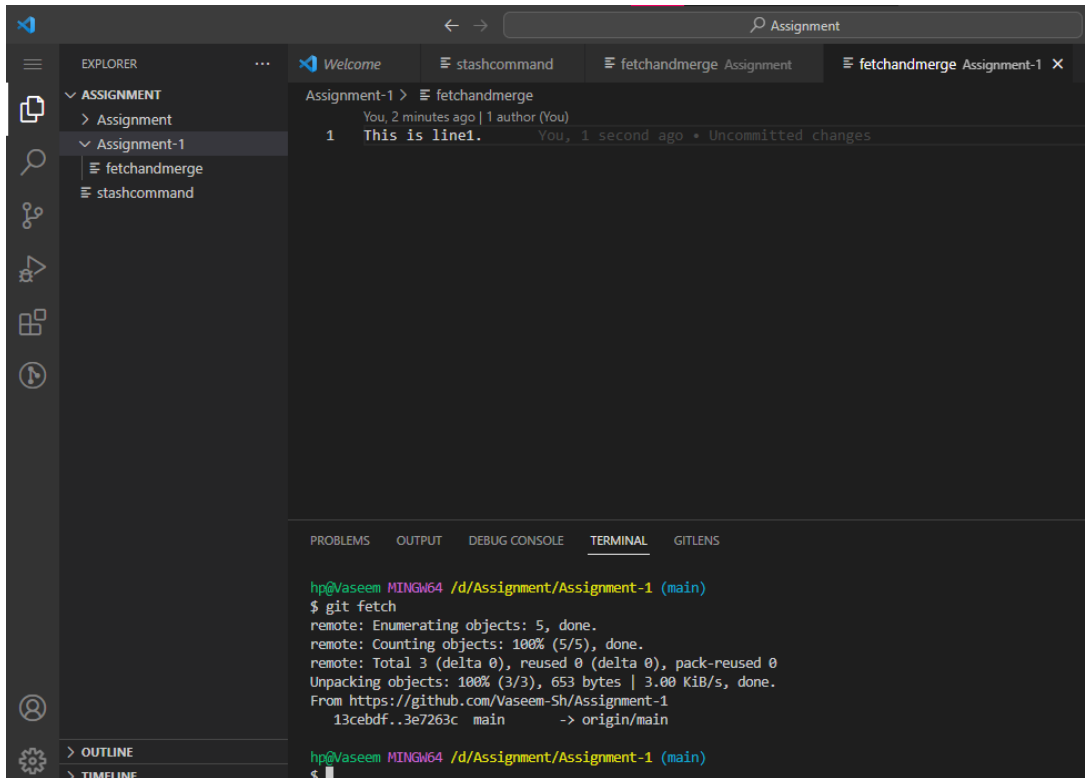
git pull

The git pull command first runs git fetch which downloads content from the specified remote repository. Then a git merge is executed to merge the remote content refs and heads into a new local merge commit.

The main difference between the git fetch and git pull commands is git fetch just shows the changes that are made in the remote repository but the git pull command pulls the changes into the local repository.

I have added a another in the file and committed it in the remote repository.

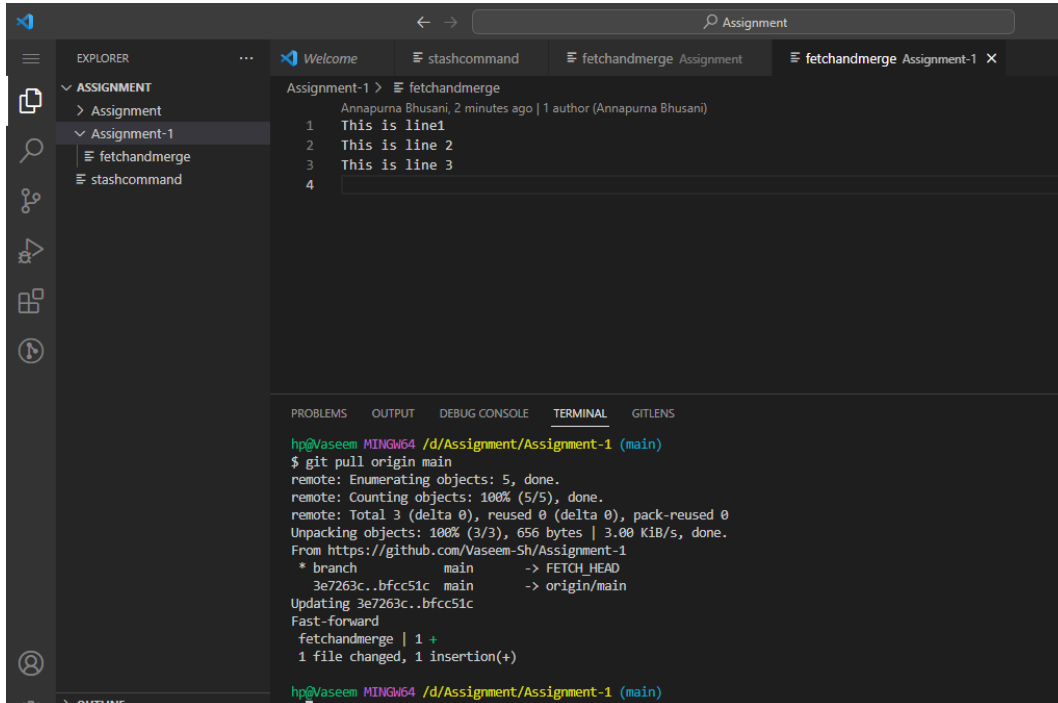




```
hp@Vaseem MINGW64 /d/Assignment/Assignment-1 (main)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 653 bytes | 3.00 KiB/s, done.
From https://github.com/Vaseem-Sh/Assignment-1
13cebdff..3e7263c  main    -> origin/main

hp@Vaseem MINGW64 /d/Assignment/Assignment-1 (main)
$
```

It just shows what changes that are made at remote repository.



```
hp@Vaseem MINGW64 /d/Assignment/Assignment-1 (main)
$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 656 bytes | 3.00 KiB/s, done.
From https://github.com/Vaseem-Sh/Assignment-1
* branch      main      -> FETCH_HEAD
3e7263c..bfcc51c  main      -> origin/main
Updating 3e7263c..bfcc51c
Fast-forward
 fetchandmerge | 1 +
1 file changed, 1 insertion(+)

hp@Vaseem MINGW64 /d/Assignment/Assignment-1 (main)
$
```

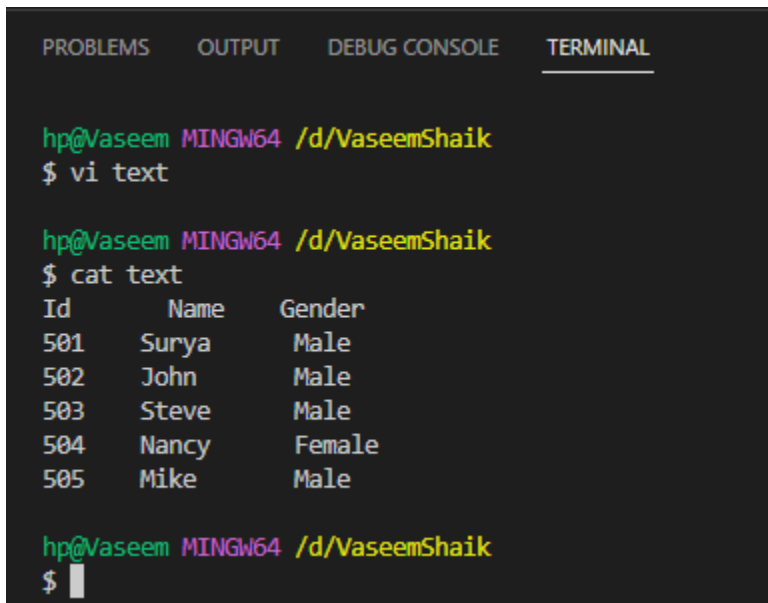
Now I have run the git pull origin main command then it both fetched and showed the line which I have added in the remote repository on my local repository

Q4. Try to find out about the awk command and use it while reading a file created by yourself. Also, make a bash script file and try to find out the prime number from the range 1 to 20.

The whole process should be carried out and by using the history command, give the screenshot of all the processes being carried out.

AWK COMMAND:

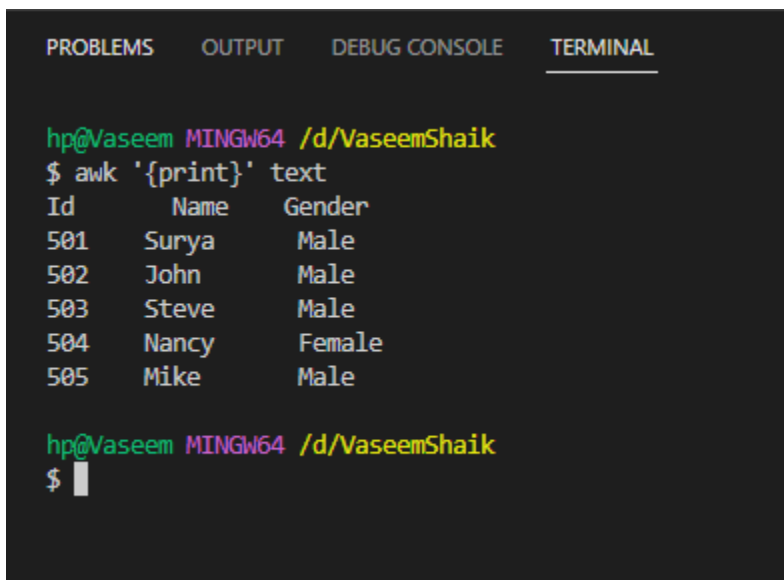
AWK is used for Pattern Scanning and Processing. It is used for Reading the Files. We can specify the patterns and fetch the data from the file. We can also count the number of input records and fields in the File.

A terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The terminal shows a user at a prompt creating a file 'text' with 'vi text'. Then, the user runs 'cat text', which displays a table with 5 rows of data: Id, Name, and Gender.

```
hp@Vaseem MINGW64 /d/VaseemShaik
$ vi text

hp@Vaseem MINGW64 /d/VaseemShaik
$ cat text
Id      Name    Gender
501     Surya   Male
502     John    Male
503     Steve   Male
504     Nancy   Female
505     Mike    Male

hp@Vaseem MINGW64 /d/VaseemShaik
$
```

A terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The terminal shows the user running 'awk '{print}' text', which outputs the same table of data as seen in the previous screenshot.

```
hp@Vaseem MINGW64 /d/VaseemShaik
$ awk '{print}' text
Id      Name    Gender
501     Surya   Male
502     John    Male
503     Steve   Male
504     Nancy   Female
505     Mike    Male

hp@Vaseem MINGW64 /d/VaseemShaik
$
```

The below command gives the data of the Male.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

hp@Vaseem MINGW64 /d/VaseemShaik
$ awk '/Male/ {print}' text
501    Surya    Male
502    John     Male
503    Steve    Male
505    Mike     Male

hp@Vaseem MINGW64 /d/VaseemShaik
$
```

Program of finding prime number between 1 to 20

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

for((i=2;i<=20;))
do
    for((j=i-1;j>=2;))
    do
        if [ `expr $i % $j` -ne 0 ] ; then
            prime=1
        else
            prime=0
            break
        fi
        j=`expr $j - 1`
    done
    if [ $prime -eq 1 ] ; then
        echo $i
    fi
    i=`expr $i + 1`
done
~
~
~
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

hp@Vaseem MINGW64 /d/VaseemShaik

\$ vi prime.sh

hp@Vaseem MINGW64 /d/VaseemShaik

\$ bash prime.sh

prime.sh: line 13: [: -eq: unary operator expected

3

5

7

11

13

17

19

hp@Vaseem MINGW64 /d/VaseemShaik

\$ █

Q5. Set up a container and run a Ubuntu operating system. For this purpose, you can make use of the docker hub and run the container in interactive mode.

All the processes pertaining to this should be provided in a screenshot for grading.

```
C:\Users\hp>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
677076032cca: Pull complete
Digest: sha256:9a0bdde4188b896a372804be2384015e90e3f84906b750c1a53539b585fbbe7f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

C:\Users\hp>docker run -it ubuntu
root@67675cb28a66:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@67675cb28a66:/# exit
exit
C:\Users\hp>
```