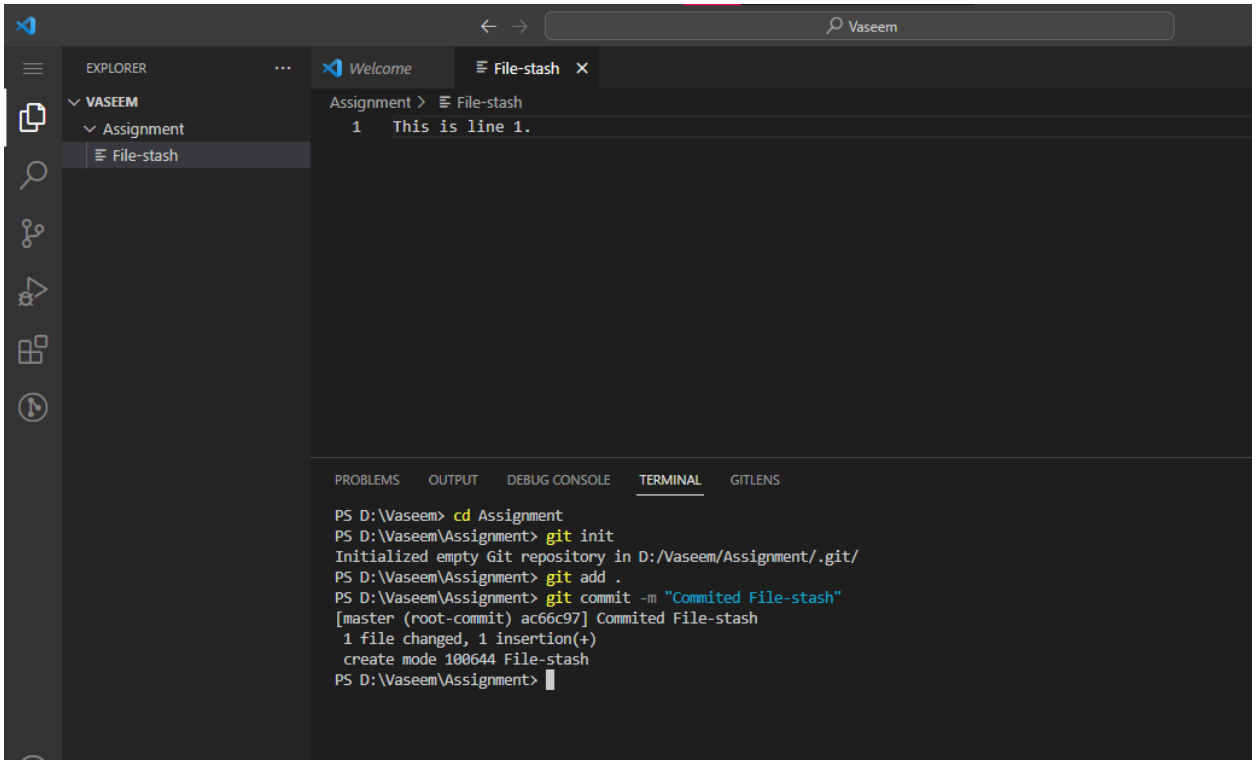
Q1. Describe the usage of the git stash command by using an example and also state the process by giving the screenshot of all the commands written in git bash.

git stash

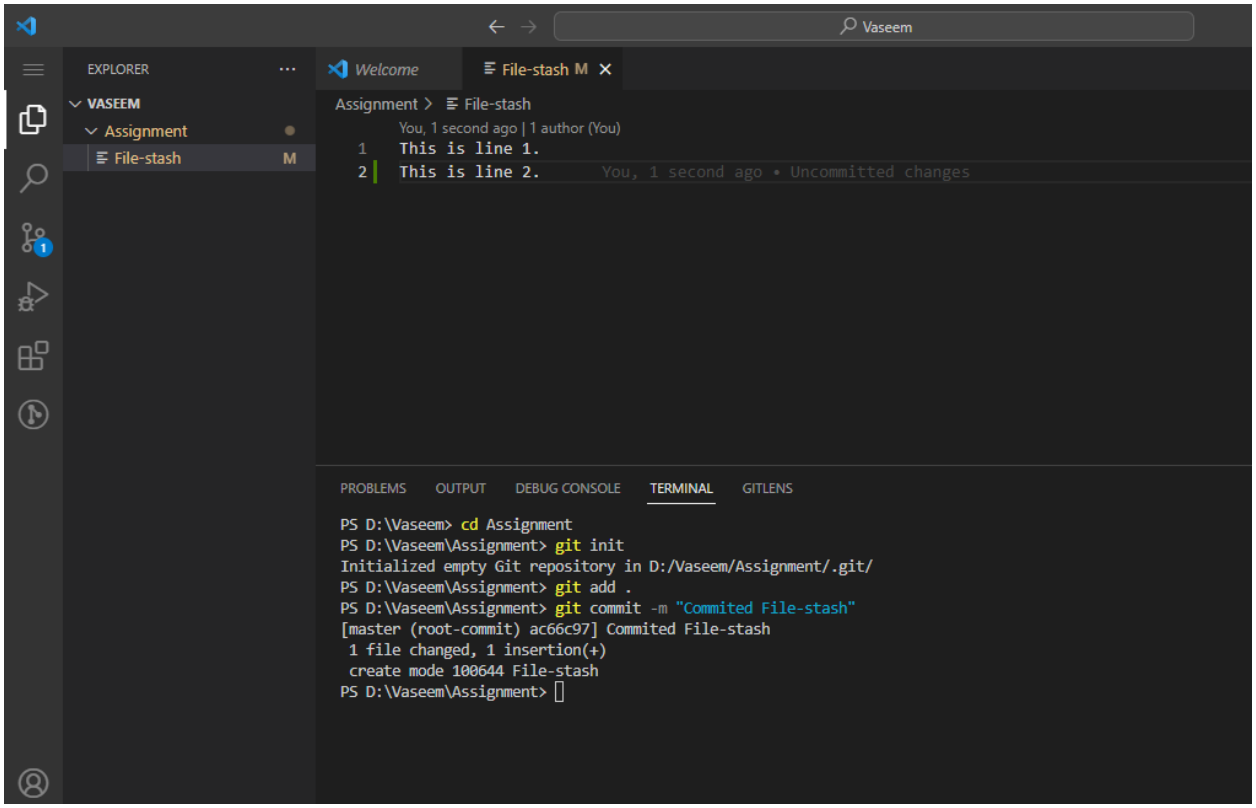
Git stash saves the uncommitted changes locally, allowing you to make changes, switch branches, and perform other Git operations. You can then reapply the stashed changes when you need them.

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

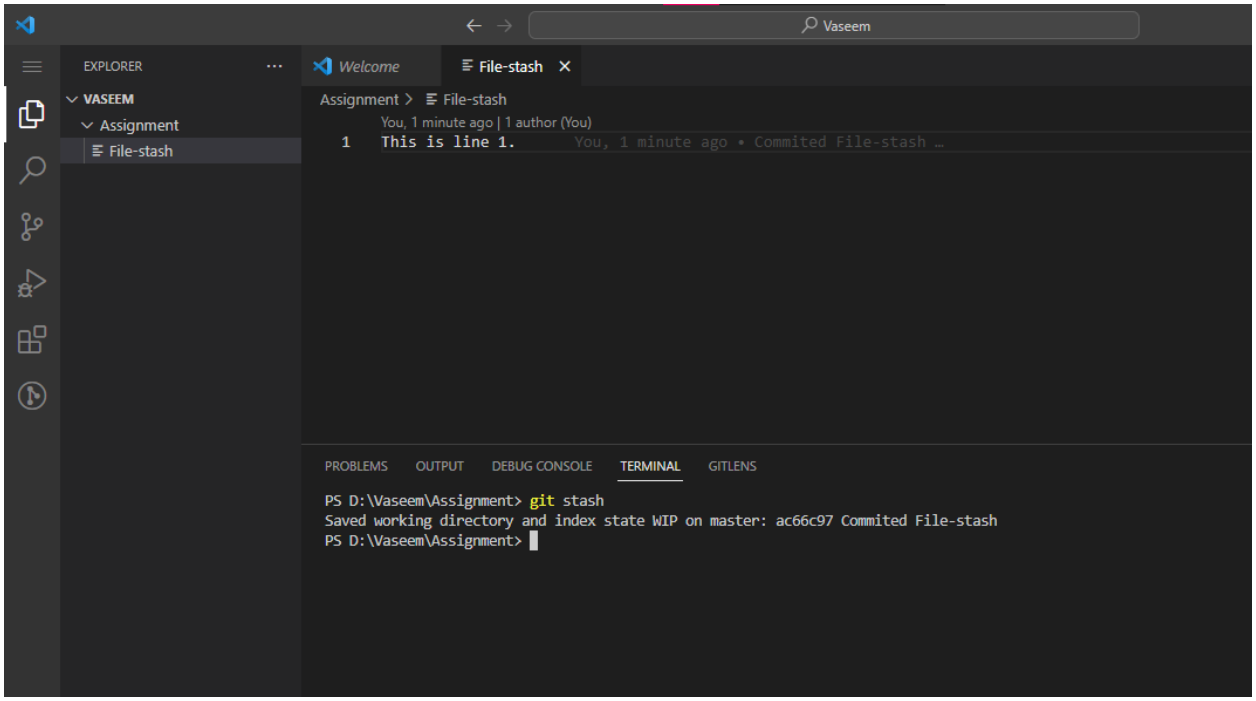
Now I have added it and committed it



Now I have added another line in the file.

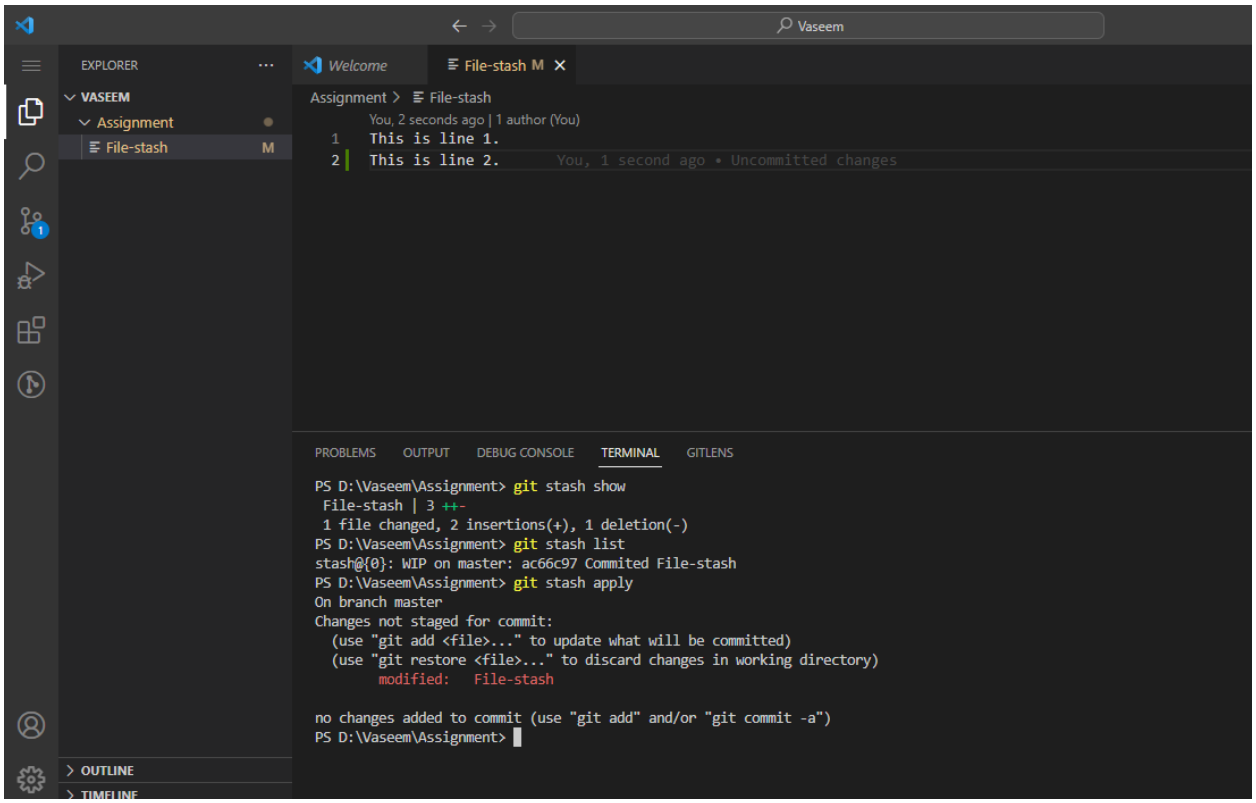


I have run the command **git stash** then the line which I have added as been stashed.



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.



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.

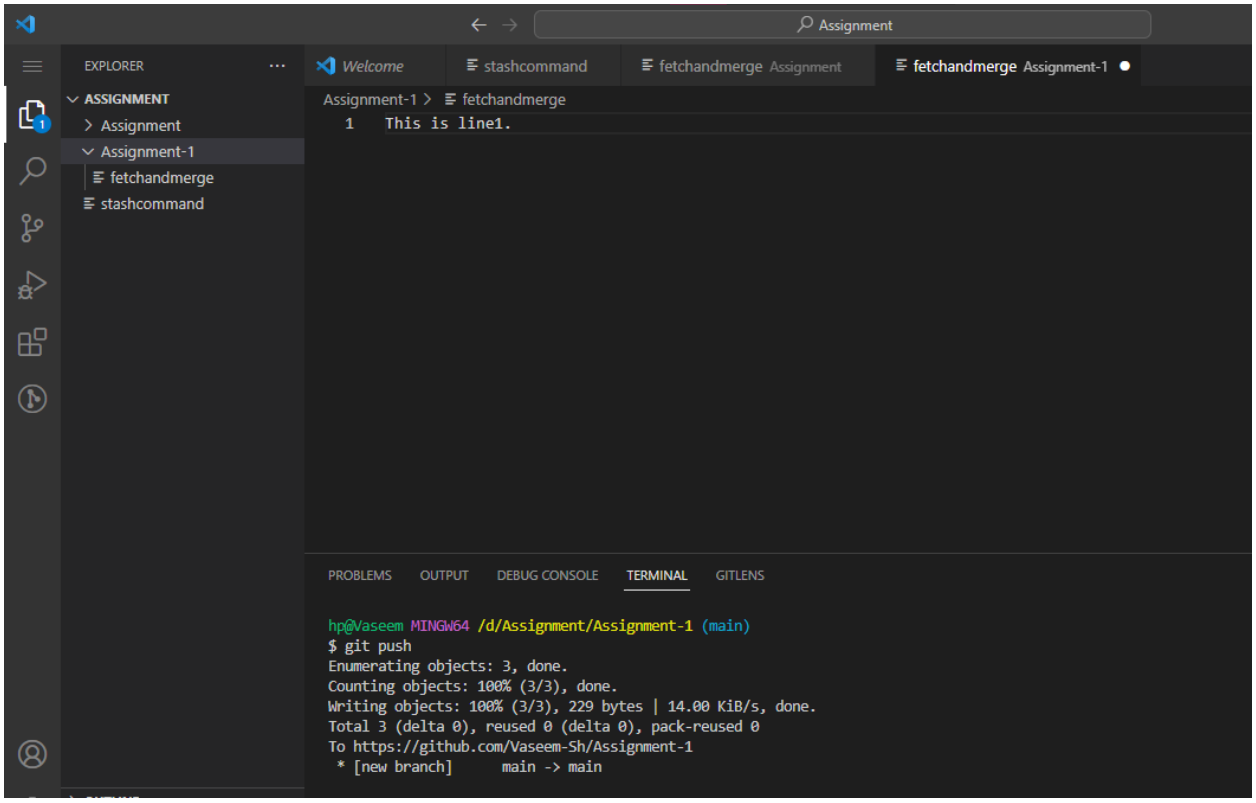
git fetch

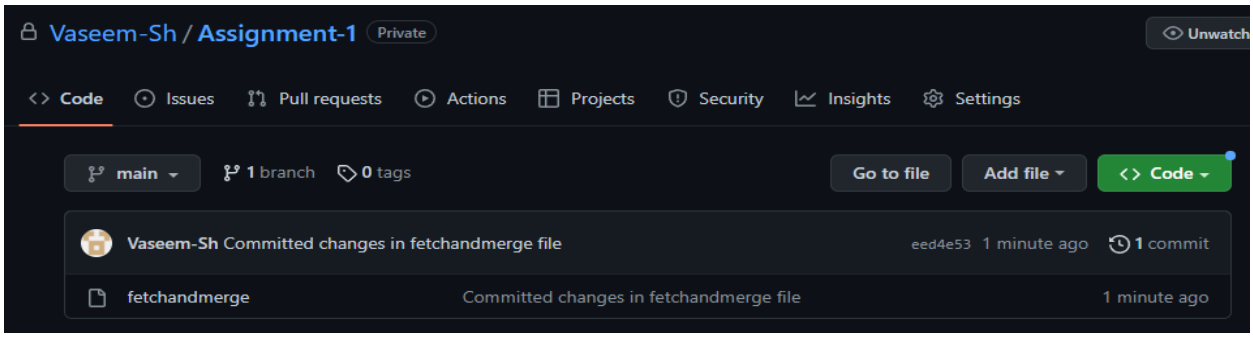
This command lets you fetch changes from the given remote repository.

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

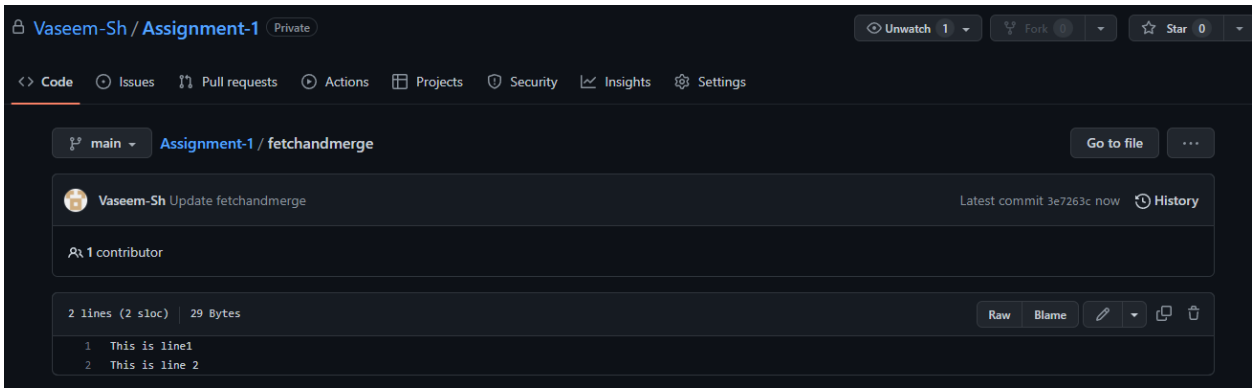
I have created a repository in the github and I have cloned it to the local machine.

I have created a file (fetchandmerge) and added, commited and pushed into the repository.

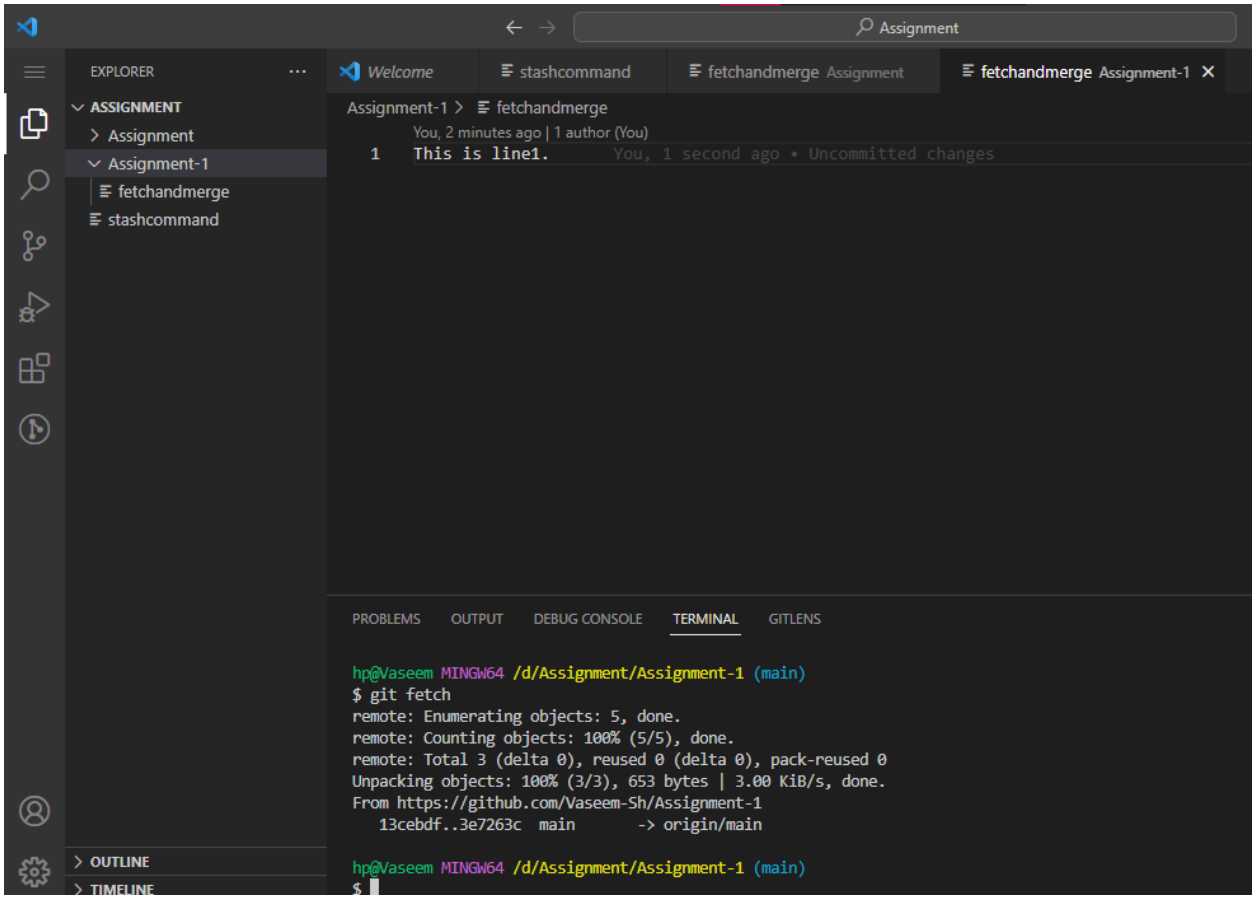




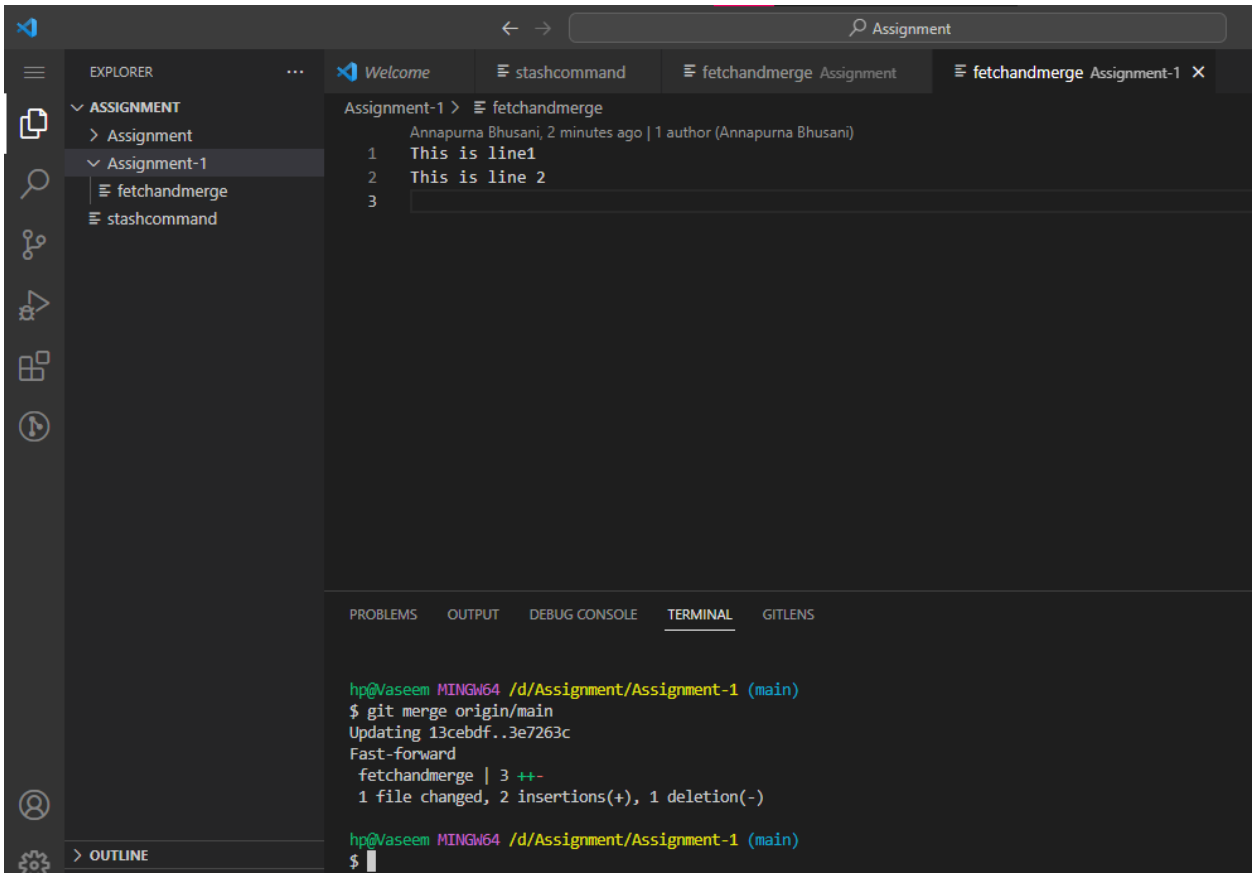
Now I have added a line into the file and commited the changes in the github website.



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.



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.

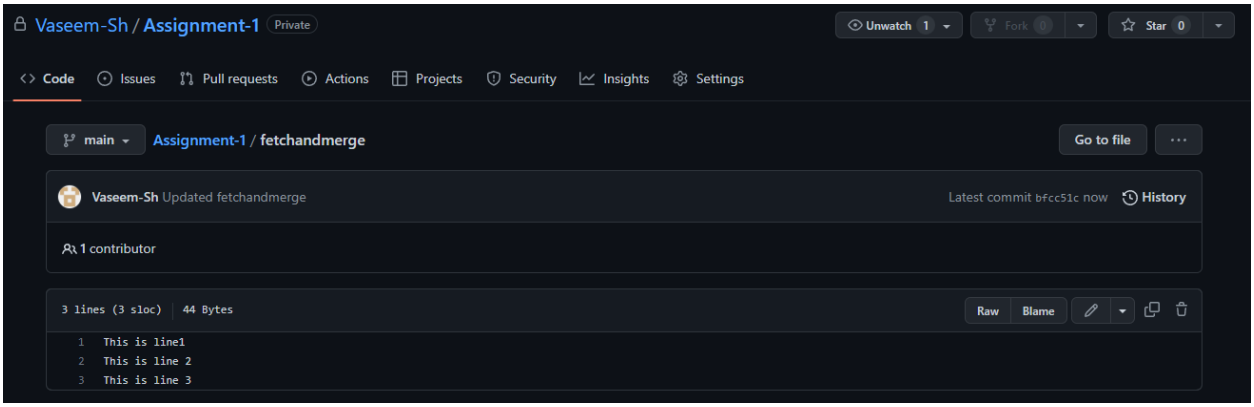
It will notify you that changes are available to your remote repository but will not bring them to your local registry.

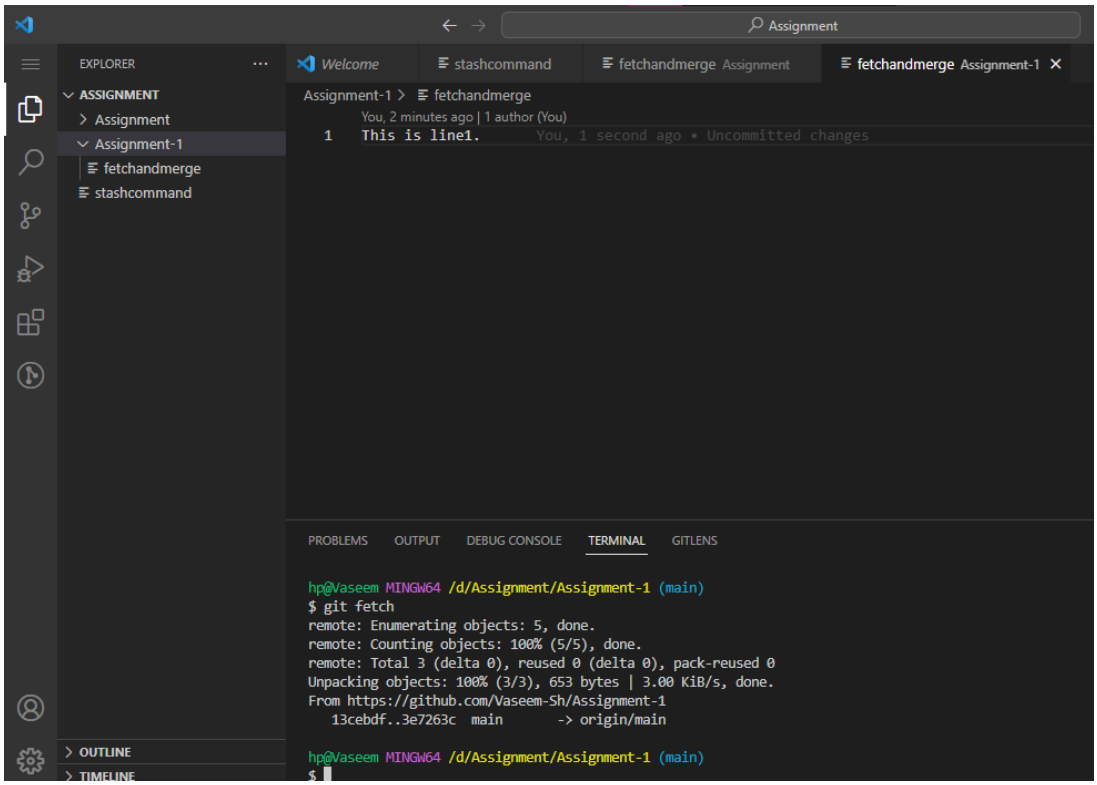
git pull

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.

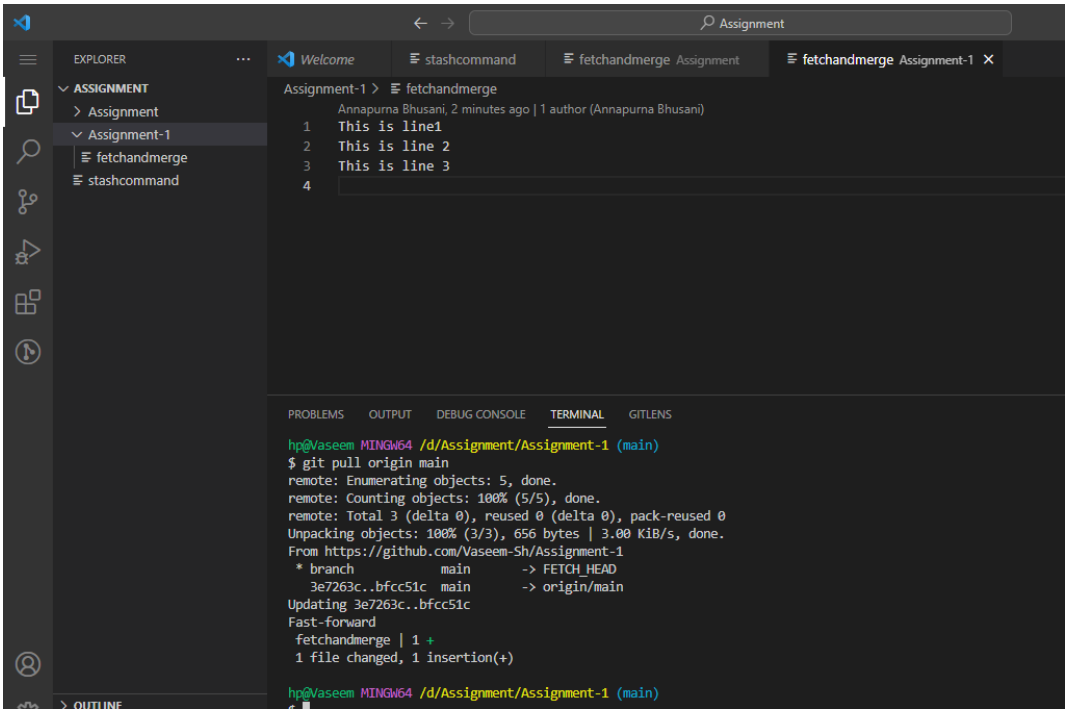
The main difference between the git fetch and git pull commands is git fetch just shows the changes that are made in the remote repository but the git pull command pulls the changes into the local repository.

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





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

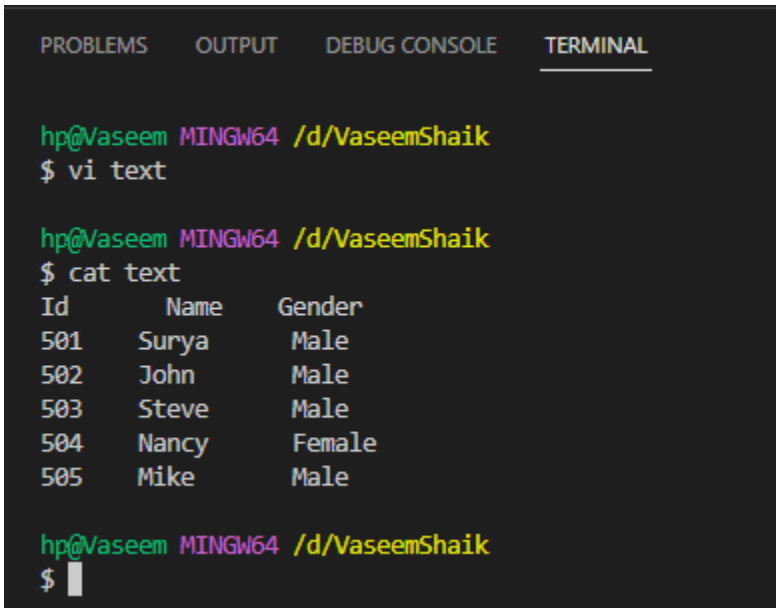


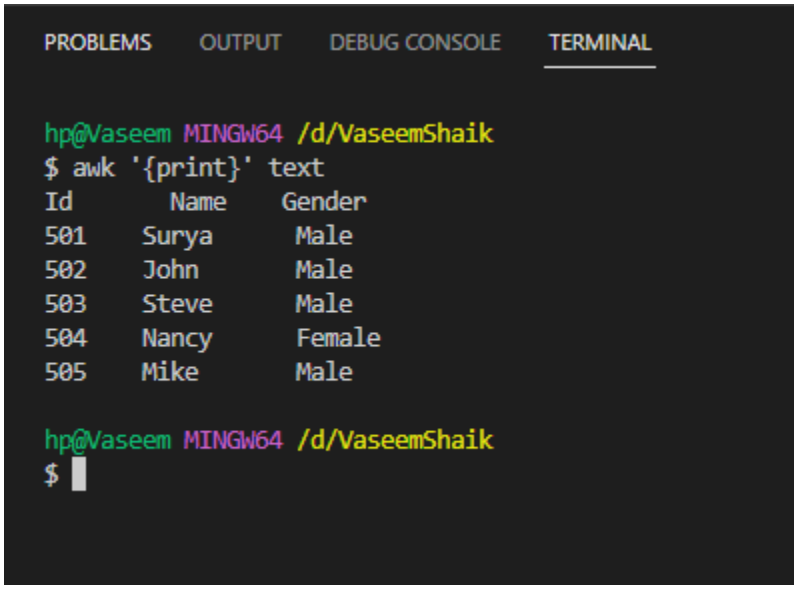
Now I have run the git pull origin main command then it both fetched and showed the line which I have added in the remote repository on my local repository

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.



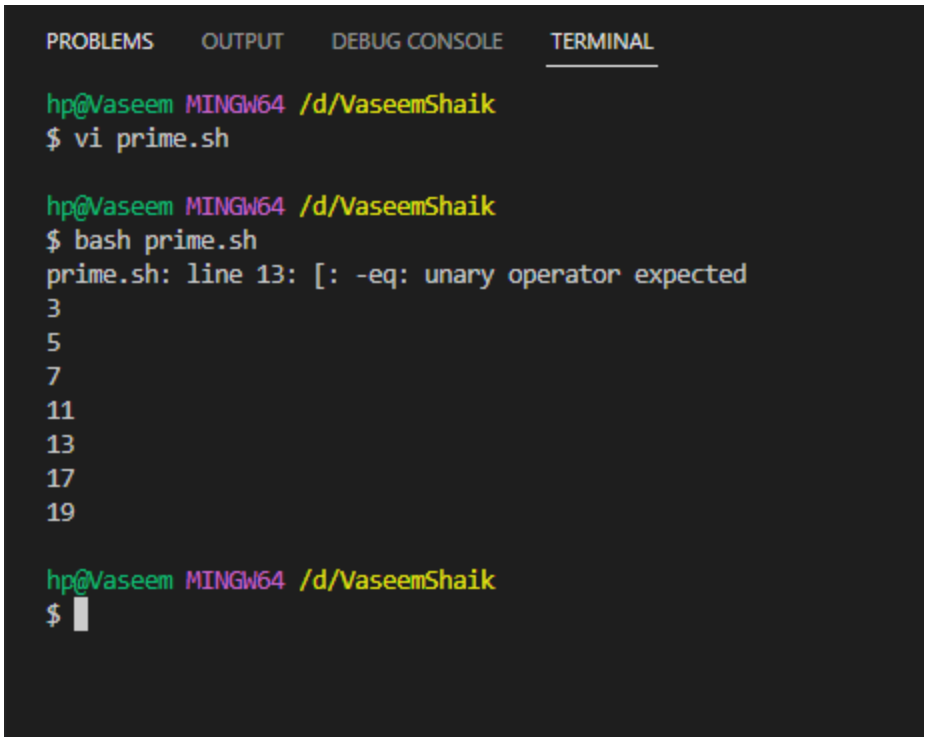


The below command gives the data of the Male.

```
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.