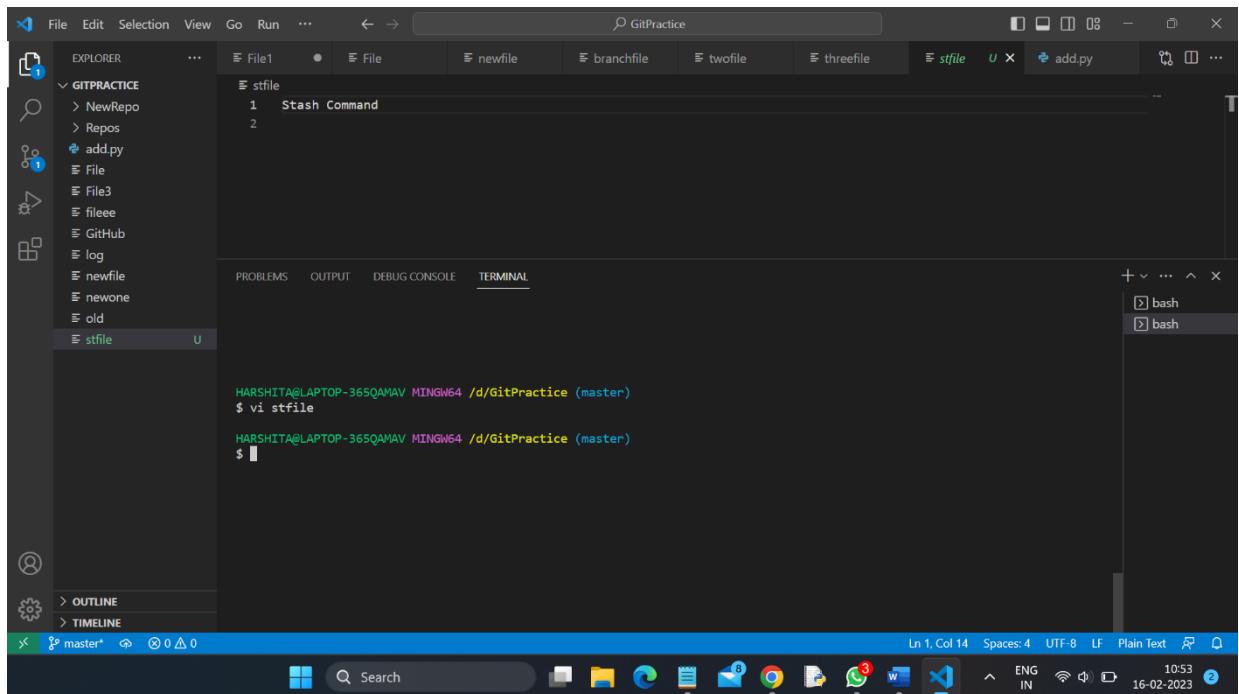


ASSIGNMENT

Git stash: Stash is used to store the changes safely in a hidden place like the stash stack.

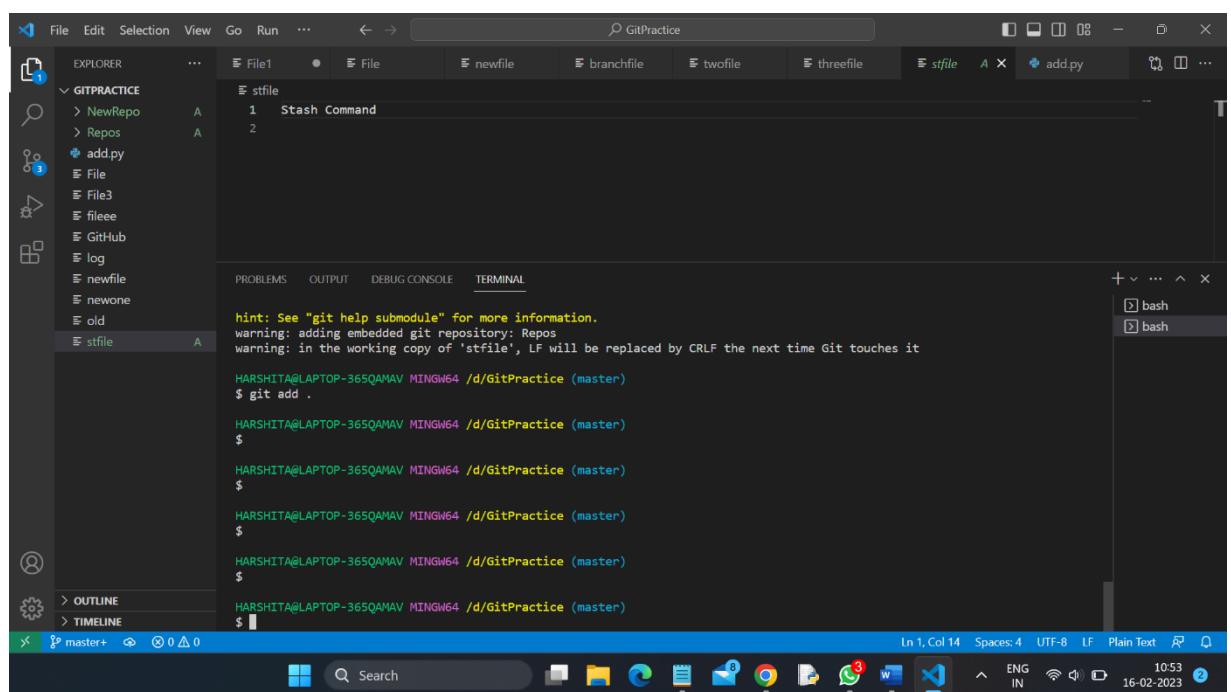
For **example**, if you are working on a file in a branch which is merged with another branch and you want to change to that another branch immediately and you do not want to commit the changes because you are not so sure about them, so you can just store them in a stack using stash and can modify the changes whenever you want to just by using the stash command.

Firstly, I created a file called stfile



The screenshot shows the VS Code interface with the following details:

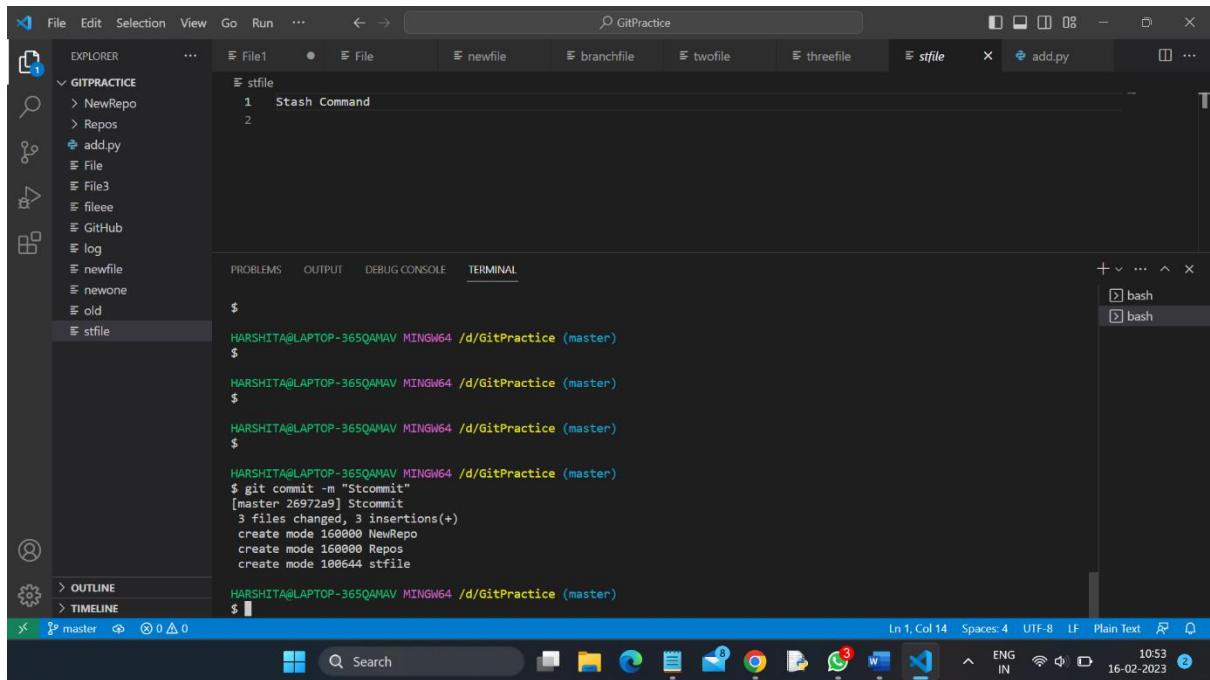
- Explorer View:** Shows a folder named "GITPRACTICE" containing "NewRepo", "Repos", "add.py", "File", "File3", "fileee", "GitHub", "log", "newfile", "newone", "old", and "stfile".
- Terminal:** Displays the command \$ vi stfile followed by the content of the file: "1 Stash Command".
- Status Bar:** Shows the path "d/GitPractice (master)", the terminal tab is active, and the status "Ln 1, Col 14 Spaces: 4 UTF-8 LF Plain Text".



The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows a folder named "GITPRACTICE" containing "NewRepo", "Repos", "add.py", "File", "File3", "fileee", "GitHub", "log", "newfile", "newone", "old", and "stfile".
- Terminal:** Displays the command \$ git add . followed by several warning messages:
 - hint: See "git help submodule" for more information.
 - warning: adding embedded git repository: Repos
 - warning: in the working copy of 'stfile', LF will be replaced by CRLF the next time Git touches it
- Status Bar:** Shows the path "d/GitPractice (master)", the terminal tab is active, and the status "Ln 1, Col 14 Spaces: 4 UTF-8 LF Plain Text".

I have then added and committed the File



```
File1 ● File newfile branchfile twofile threefile stfile add.py ...
```

```
stfile
1 Stash Command
2
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
```

```
$
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$
```

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$
```

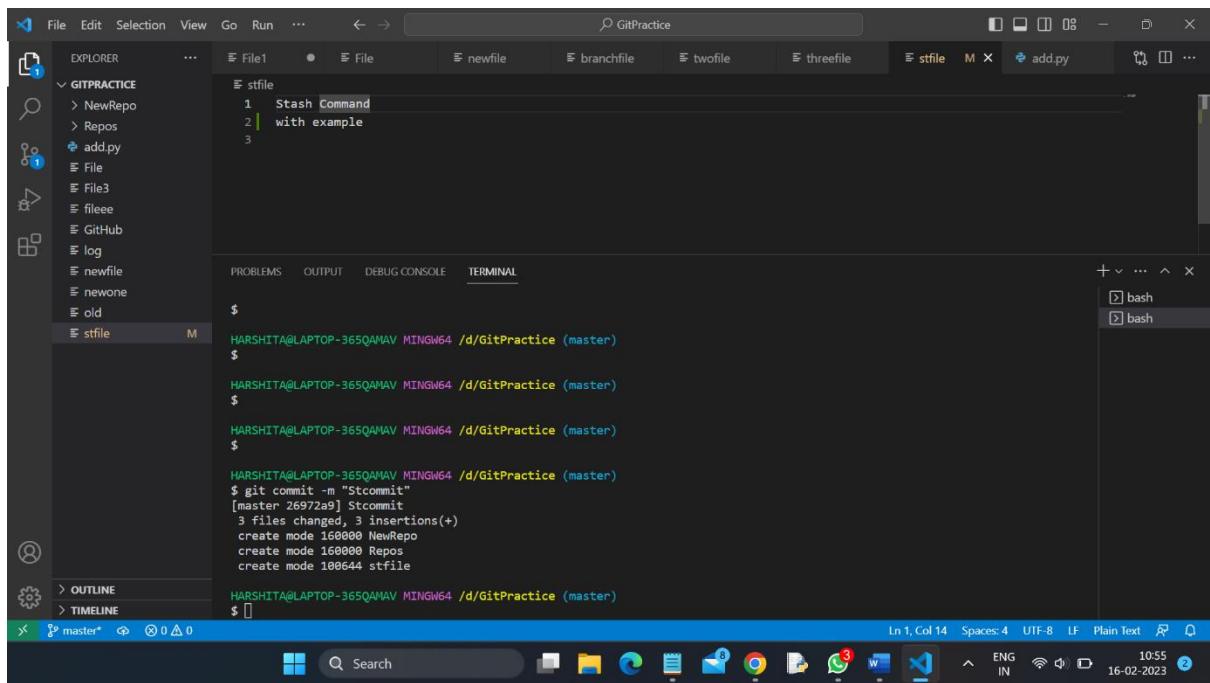
```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$
```

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ git commit -m "Stcommit"
[master 26972a9] Stcommit
 3 files changed, 3 insertions(+)
 create mode 160000 NewRepo
 create mode 160000 Repos
 create mode 100644 stfile
```

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$
```

```
Ln 1, Col 14 Spaces: 4 UTF-8 LF Plain Text ⌂ 1053 16-02-2023
```

Now, I made some changes in the File



```
File1 ● File newfile branchfile twofile threefile stfile add.py ...
```

```
stfile
1 Stash Command
2 | with example
3
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
```

```
$
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$
```

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$
```

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$
```

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ git commit -m "Stcommit"
[master 26972a9] Stcommit
 3 files changed, 3 insertions(+)
 create mode 160000 NewRepo
 create mode 160000 Repos
 create mode 100644 stfile
```

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$
```

```
Ln 1, Col 14 Spaces: 4 UTF-8 LF Plain Text ⌂ 1055 16-02-2023
```

Now, I stashed the File. The changes made are deleted

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following command and its execution:

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ git stash
warning: in the working copy of 'stfile', LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state WIP on master: 26972a9 Stcommit

HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$
```

The terminal shows that the changes made to the 'stfile' were successfully stashed, and the working directory was saved in a 'WIP' state.

To see the stash stack list, I have done git stash list

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following command and its execution:

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ git commit -m "Stcommit"
[master 26972a9] Stcommit
 3 files changed, 3 insertions(+)
 create mode 160000 NewRepo
 create mode 160000 Repos
 create mode 100644 stfile

HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ git stash
warning: in the working copy of 'stfile', LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state WIP on master: 26972a9 Stcommit

HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ git stash list
stash@{0}: WIP on master: 26972a9 Stcommit

HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$
```

The terminal shows the 'stash list' command output, which displays a single stash entry at index 0, indicating a 'WIP' state on the master branch.

Git stash apply got back the deleted changes

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following command history:

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ git stash
warning: in the working copy of 'stfile', LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state WIP on master: 26972a9 Stcommit

HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ git stash list
stash@{0}: WIP on master: 26972a9 Stcommit

HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ 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:  stfile

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

HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$
```

The status bar at the bottom indicates the current file is 'stfile'.

Git stash show, shows the number of changes made

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following command history:

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ git stash list
stash@{0}: WIP on master: 26972a9 Stcommit

HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ 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:  stfile

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

HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ git stash show
stfile | 1
 1 file changed, 1 insertion(+)

HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$
```

The status bar at the bottom indicates the current file is 'stfile'.

Use of Git Fetch and Git Merge Command:

Git Fetch is used to bring the Remote changes to the Local Repository

For example, we have created a file in the Repository locally and push that file into the Repository. We made some changes in that file Remotely and committed there. So to get that changes locally we Fetch those Changes by using the Fetch Command. To observe the Changes we will merge the branches.

Firstly, I created a file and added and committed the File

```
File1  ●  File  newfile  branchfile  twofile  threefile  stfile  add.py
gitpractice
GITPRACTICE
NewRepo
> NewRepo
file1
file2
Newfile
> Repos
add.py
File
File3
fileee
GitHub
log
newfile
newone
old

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
From Repos
* branch      HEAD      -> FETCH_HEAD
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ cd ~/NewRepo
HARSHITA@LAPTOP-365QAMAV MINGW64 ~/NewRepo (main)
$ vi ftfile
HARSHITA@LAPTOP-365QAMAV MINGW64 ~/NewRepo (main)
$ git add .
warning: in the working copy of 'ftfile', LF will be replaced by CRLF the next time Git touches it
HARSHITA@LAPTOP-365QAMAV MINGW64 ~/NewRepo (main)
$ git commit -m "ftcommit"
[main fc04403] ftcommit
 1 file changed, 1 insertion(+)
 create mode 100644 ftfile
HARSHITA@LAPTOP-365QAMAV MINGW64 ~/NewRepo (main)
$
```

Then I have pushed the File into the Repository

```
File1  ●  File  newfile  branchfile  twofile  threefile  stfile  add.py
gitpractice
GITPRACTICE
NewRepo
> NewRepo
file1
file2
Newfile
> Repos
add.py
File
File3
fileee
GitHub
log
newfile
newone
old

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
HARSHITA@LAPTOP-365QAMAV MINGW64 ~/NewRepo (main)
$ git commit -m "ftcommit"
[main fc04403] ftcommit
 1 file changed, 1 insertion(+)
 create mode 100644 ftfile
HARSHITA@LAPTOP-365QAMAV MINGW64 ~/NewRepo (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 277 bytes | 277.00 Kib/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Harshitamattaparti/NewRepo.git
 b4da064..fc04403  main -> main
HARSHITA@LAPTOP-365QAMAV MINGW64 ~/NewRepo (main)
$
```

The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is for a GitHub repository named 'Harshitamattaparti / NewRepo'. The repository page displays a list of commits:

Author	Commit Message	Date
Harshitamattaparti ftcommit	new clone	last week
Harshitamattaparti	First Commit	2 weeks ago
Harshitamattaparti	commit	last week
Harshitamattaparti	new clone	last week
Harshitamattaparti	hai	last week
Harshitamattaparti	ftfile	1 minute ago

On the right side of the repository page, there are sections for 'About' (no description), 'Releases' (no releases published), and 'Packages' (no packages published). Below the repository page, the Windows taskbar is visible, showing various pinned icons like Mail, Photos, and File Explorer.

The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is for a GitHub file page, specifically 'NewRepo/blob/main/ftfile'. The page displays the content of the file:

```
1 lines (1 sloc) | 17 Bytes
```

1 Fetching Command

Below the file content, there are links for 'Raw', 'Blame', and file operations. At the bottom of the page, there is a 'Give feedback' button. The page footer includes standard GitHub links like Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

The I edited the File Remotely and committed it

A screenshot of a web browser window showing a GitHub repository page. The URL is github.com/Harshitamattaparti/NewRepo/blob/main/ftfile. The page displays a single file named 'ftfile' with the following content:

```
1 Fetching Command
2 Edited Remotely
```

The GitHub interface shows the file has 2 lines (2 sloc) and 33 Bytes. It also indicates 1 contributor and provides options to Raw, Blame, Edit, or Delete the file.

A screenshot of the Microsoft Visual Studio Code (VS Code) interface. The Explorer sidebar shows a folder named 'GITPRACTICE' containing a 'NewRepo' folder. The 'TERMINAL' tab is active, displaying a command-line session:

```
HARSHITA@LAPTOP-365Q4NAV MINGW64 ~/NewRepo (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 277 bytes | 277.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Harshitamattaparti/NewRepo.git
 b4da064..fc04403 main -> main

HARSHITA@LAPTOP-365Q4NAV MINGW64 ~/NewRepo (main)
$ git log
commit fc04403fda5a8f4d95eef9833158ee878283242 (HEAD -> main, origin/main)
Author: 20A91A05A4@aec.edu.in <20A91A05A4@aec.edu.in>
Date: Thu Feb 16 11:32:05 2023 +0530

    ftcommit
commit b4da064fac9d039696f5b29d08cada72238cea98
```

The status bar at the bottom of the VS Code window shows the date as 16-02-2023 and the time as 11:34.

Now, I fetched the changes to the Local repository

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following command and its execution:

```
commit 44fe5961a3b1e5e8d4e918e4b131a43142c2ecc0
Author: unknown <20A01A05A4@aec.edu.in>
Date: Thu Feb 9 12:03:33 2023 +0530

    new clone

commit 151286c9ec462d937106893d4535781f4d91514a (Harshita)
Author: unknown <20A01A05A4@aec.edu.in>

HARSHITA@LAPTOP-365QAMAV MINGW64 ~/NewRepo (main)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 667 bytes | 41.00 KiB/s, done.
From https://github.com/Marshitamattaparti/NewRepo
  fc04403..d3958bb main      -> origin/main
```

Now, using the git merge, I have merged the branches

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following command and its execution:

```
commit 44fe5961a3b1e5e8d4e918e4b131a43142c2ecc0
Author: unknown <20A01A05A4@aec.edu.in>
Date: Thu Feb 9 12:03:33 2023 +0530

    new clone

commit 151286c9ec462d937106893d4535781f4d91514a (Harshita)
Author: unknown <20A01A05A4@aec.edu.in>

HARSHITA@LAPTOP-365QAMAV MINGW64 ~/NewRepo (main)
$ git merge origin/main
Updating fc04403..d3958bb
Fast-forward
  ffile | 1 +
  1 file changed, 1 insertion(+)

HARSHITA@LAPTOP-365QAMAV MINGW64 ~/NewRepo (main)
$
```

The screenshot shows the VS Code interface with the title bar "GitPractice". The Explorer sidebar on the left shows a repository named "NewRepo" under "GITPRACTICE". The terminal tab at the bottom displays the output of a git merge command:

```
$ git merge origin/main
Updating fc04403..d39588b
Fast-forward
  fftfile | 1 +
  1 file changed, 1 insertion(+)

HARSHITA@LAPTOP-365QA\AV MINGW64 ~/NewRepo (main)
$ git log
commit d39588b21e87d4434c8ea9a36f6c5a556acf52bd (HEAD -> main, origin/main)
Author: HarshitaMattaParti <84396867+Harshitamattaparti@users.noreply.github.com>
Date: Thu Feb 16 11:33:44 2023 +0530

    Update fftfile

commit fc04403fda5a8f4d99ee98331508ee878203242
Author: 20A91A05A4@aec.edu.in <20A91A05A4@aec.edu.in>
Date: Thu Feb 16 11:32:05 2023 +0530

    ffcmit

commit b4da064fac9d039696f5b29d08cada72238cea98
```

The changes made Remotely are now reflected locally

The screenshot shows the VS Code interface with the title bar "GitPractice". The Explorer sidebar on the left shows a repository named "NewRepo" under "GITPRACTICE". The terminal tab at the bottom displays the output of a git fetch command:

```
Fetching Command
Edited Remotely
~
~
```

The status bar at the bottom right shows "1,16 A11".

Difference between Git Fetch and Git Pull:

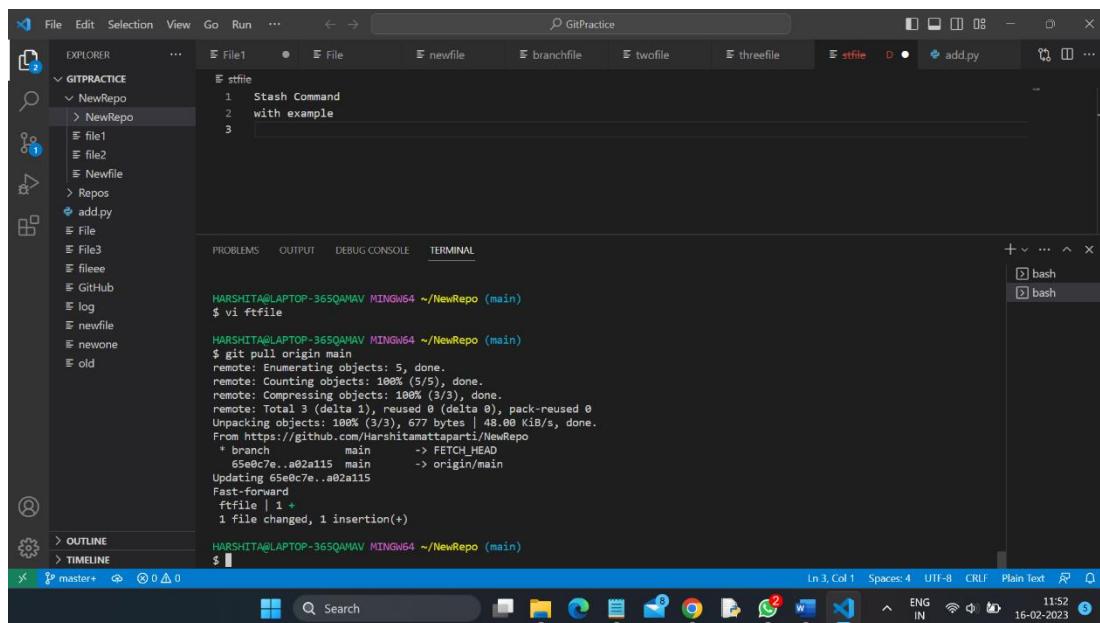
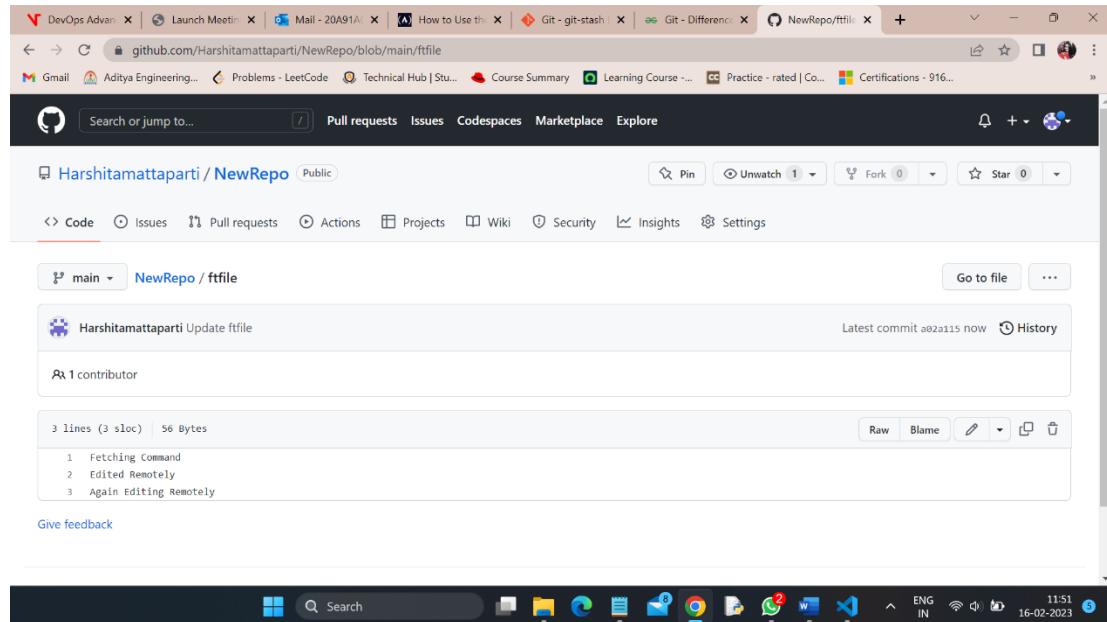
Git Fetch is the command that tells the local repository that there are changes available in the remote repository without bringing the changes into the local repository.

Git Pull on the other hand brings the copy of the remote directory changes into the local repository.

As we have seen the Git Fetch above, now let's see what is the difference between Git Fetch and Git Pull

In Git Fetch Command, first we have to fetch the changes and then merge them. But Git pull is used to Fetch and merge the changes at the same time. Instead of Giving Fetch and Merge Commands separately, we can use the pull Command.

Again I edited the file remotely



The screenshot shows a Windows 10 desktop with the Visual Studio Code application open. The title bar reads "GitPractice". The left sidebar displays a file tree under the "GITPRACTICE" repository, including files like "stfile", "file1", "file2", "Newfile", "add.py", "File", "File3", "fileee", "GitHub", "log", "newfile", "newone", and "old". The main editor area shows the content of "stfile". The terminal tab at the bottom is active, displaying the following output:

```
1 file changed, 1 insertion(+)

HARSHITA@LAPTOP-365Q0AMV MINGW64 ~/NewRepo (main)
$ git log
commit a02a115ddac502715907e145b59b53fb0ca777f (HEAD -> main, origin/main)
Author: Harshitamattaparti <84396867+Harshitamattaparti@users.noreply.github.com>
Date:   Thu Feb 16 11:51:20 2023 +0530

        Update ftfile

commit 65e0c7ee4181702421c3691869d13d705fcfa63d
Author: Harshitamattaparti <84396867+Harshitamattaparti@users.noreply.github.com>
Date:   Thu Feb 16 11:49:24 2023 +0530

        Update ftfile

commit a3ee228ac7d5fedf75ebccc470085822f2f55337
Author: 20A91A05A4@aec.edu.in <20A91A05A4@aec.edu.in>
Date:   Thu Feb 16 11:44:48 2023 +0530

ftcomm
```

The changes made remotely are now reflected locally

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.

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ vi myfile

HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ cat myfile
Name    Rollno   Gender
Kumari  532     Female
Ramya   564     Female
Sunny   567     Male
Michael 584     Male
Jonas   539     Male
Martha  574     Female
```

Same like cat command, we can give this command for printing the whole File.

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ awk '{print}' myfile
Name    Rollno   Gender
Kumari  532     Female
Ramya   564     Female
Sunny   567     Male
Michael 584     Male
Jonas   539     Male
Martha  574     Female
```

The below command gives the records having the ‘female’ string.

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ awk '/Female/ {print}' myfile
Kumari  532     Female
Ramya   564     Female
Martha  574     Female
```

The below command prints the column 1 and column 3 (\$ is used for printing the columns)

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ awk '{print $1,$3}' myfile
Name Gender
Kumari Female
Ramya Female
Sunny Male
Michael Male
Jonas Male
Martha Female
```

NR command counts the number of input records and provides the Numbering for the Records

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ awk '{print NR,$0}' myfile
1 Name Rollno Gender
2 Kumari 532 Female
3 Ramya 564 Female
4 Sunny 567 Male
5 Michael 584 Male
6 Jonas 539 Male
7 Martha 574 Female
```

NF command counts the number of fields in an input Record, Here NF is considered as the last field

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ awk '{print $1,$NF}' myfile
Name Gender
Kumari Female
Ramya Female
Sunny Male
Michael Male
Jonas Male
Martha Female
```

Bash Script for printing prime numbers between 1 to 20:

```
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
~
~
prime2.sh [unix] (21:25 16/02/2023)
-- INSERT --
```

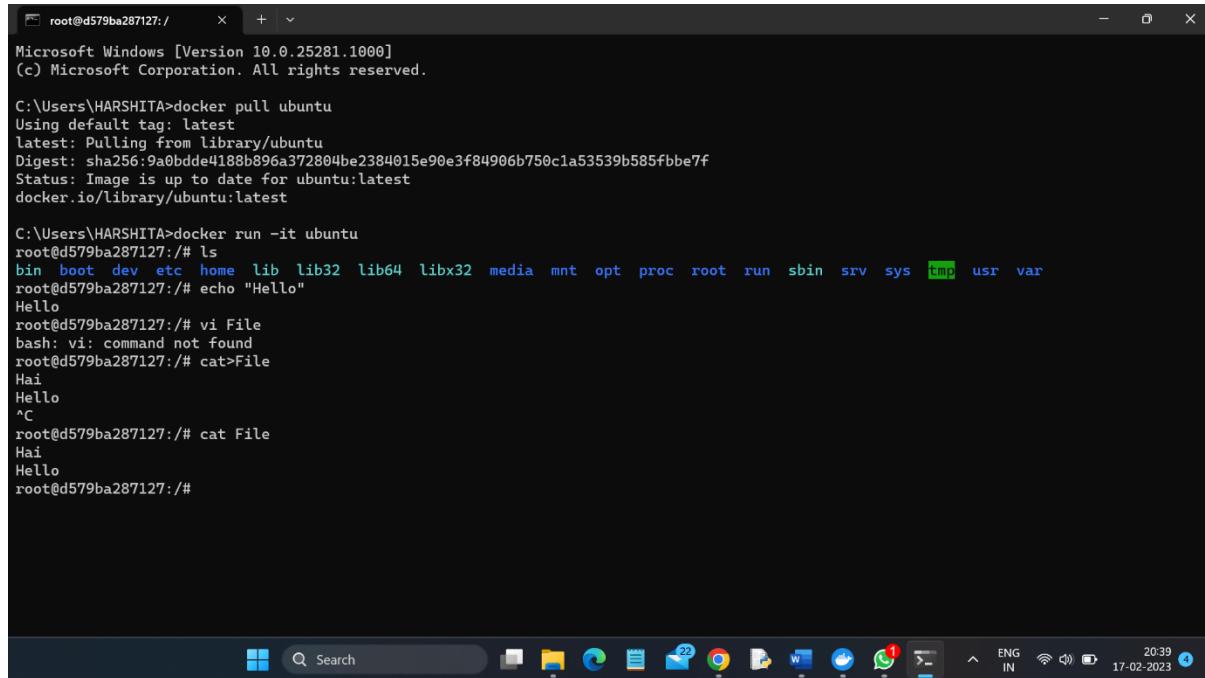
OUTPUT:

```
HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ vi prime2.sh

HARSHITA@LAPTOP-365QAMAV MINGW64 /d/GitPractice (master)
$ bash prime2.sh
prime2.sh: line 13: [: -eq: unary operator expected
3
5
7
11
13
17
19
```

Setting up a container and run a ubuntu OS:

Here, we are setting up a container by using pull and then we are running the ubuntu by using the run command.



The screenshot shows a Windows terminal window with a black background and white text. At the top, it displays the Windows logo, the title bar 'root@d579ba287127:/', and standard window controls (minimize, maximize, close). Below the title bar, the terminal output is shown in several lines:

```
Microsoft Windows [Version 10.0.25281.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HARSHITA>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
Digest: sha256:9a0bdd4188b896a372804be2384015e90e3f84906b750c1a53539b585fbbe7f
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest

C:\Users\HARSHITA>docker run -it ubuntu
root@d579ba287127:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@d579ba287127:/# echo "Hello"
Hello
root@d579ba287127:/# vi File
bash: vi: command not found
root@d579ba287127:/# cat>File
Hai
Hello
^C
root@d579ba287127:/# cat File
Hai
Hello
root@d579ba287127:/#
```

At the bottom of the terminal window, there is a taskbar with various icons for common Windows applications like File Explorer, Edge, and File History. To the right of the taskbar, system status indicators show the date (17-02-2023), time (20:39), battery level, signal strength, and language settings (ENG IN).