

Assignment–4

Module-6: Git and GitHub

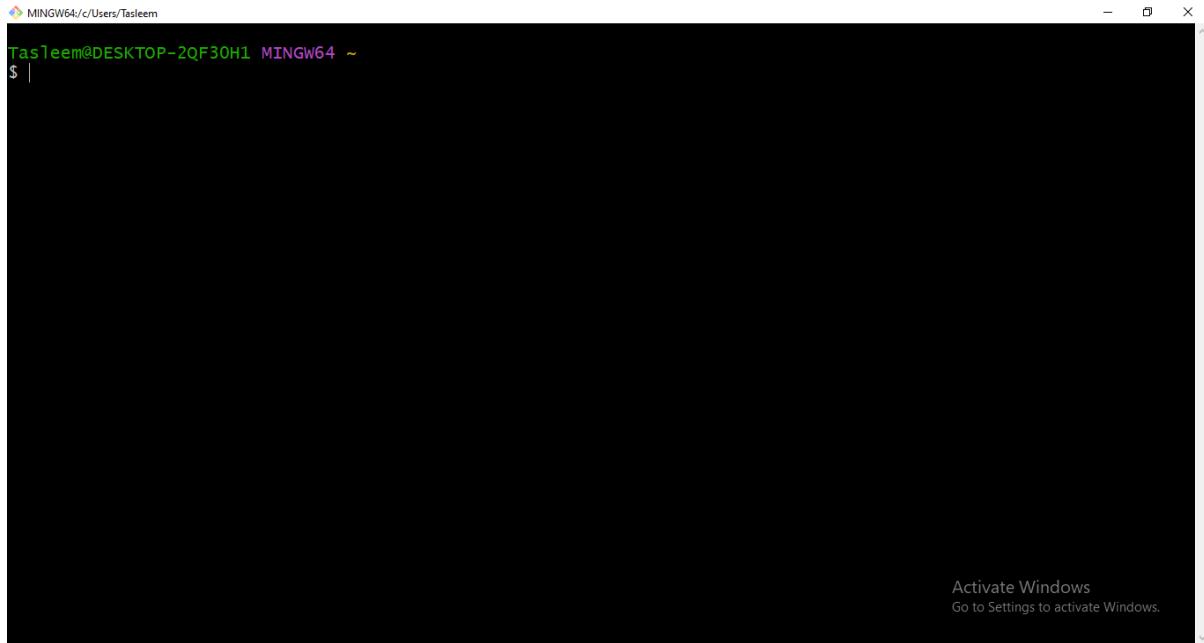
Submitted by : Shaik Junaid Adil

Date of Submission: 19-06-2024

Submitted to: Vikul

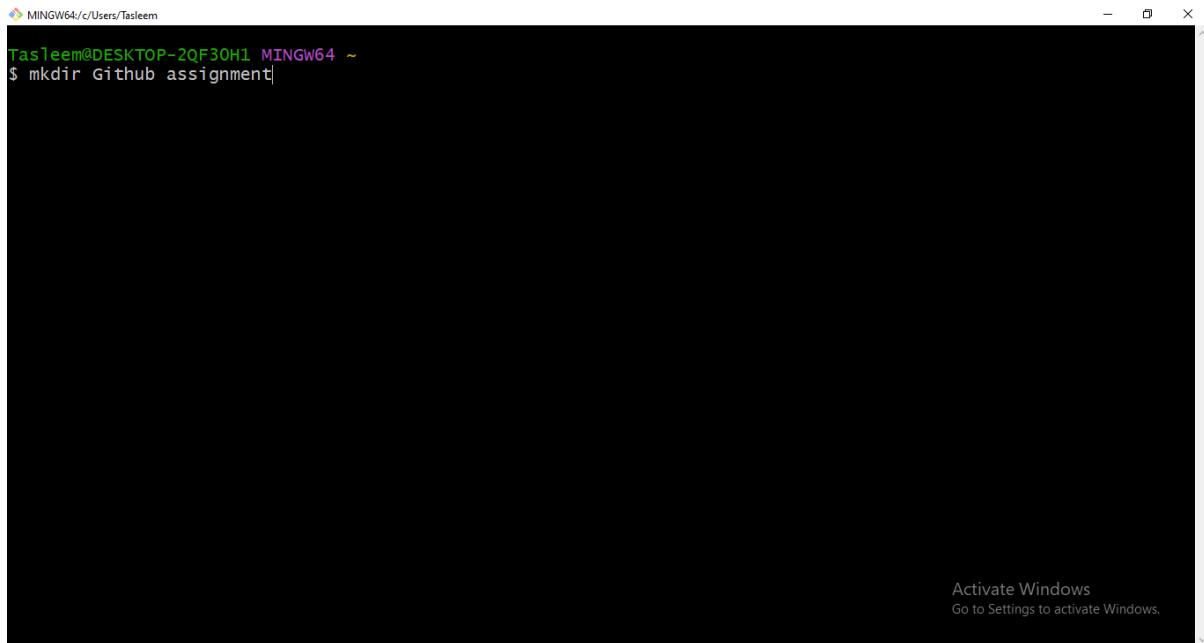
L1 - Create Local git repository and demonstrate all git reset options and revert. Compare the differences

Step-1: Open git bash Terminal



A screenshot of a terminal window titled "MINGW64/c/Users/Tasleem". The window shows the command line prompt "Tasleem@DESKTOP-2QF30H1 MINGW64 ~" followed by a "\$" sign and a vertical cursor. The rest of the window is blank black space. In the bottom right corner, there is a watermark that reads "Activate Windows Go to Settings to activate Windows."

Step-2: Create a directory “Github assignment”



A screenshot of a terminal window titled "MINGW64/c/Users/Tasleem". The window shows the command line prompt "Tasleem@DESKTOP-2QF30H1 MINGW64 ~" followed by a "\$" sign and the command "mkdir Github assignment" being typed. The rest of the window is blank black space. In the bottom right corner, there is a watermark that reads "Activate Windows Go to Settings to activate Windows."



```
MINGW64/c/Users/Tasleem
Tasleem@DESKTOP-2QF30H1 MINGW64 ~
$ mkdir Github assignment

Tasleem@DESKTOP-2QF30H1 MINGW64 ~
$ |
```

Activate Windows
Go to Settings to activate Windows.

Step-3: Go to the directory “Github”



```
MINGW64/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~ (master)
$ cd Github/

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Activate Windows
Go to Settings to activate Windows.

Step-4: Initialize Git.

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~ (master)
$ cd Github/
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git init
Initialized empty Git repository in c:/Users/Tasleem/Github/.git/
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Activate Windows
Go to Settings to activate Windows.

Step-5: Create a file “Testfile1.sh”

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~ (master)
$ cd Github/
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git init
Initialized empty Git repository in c:/Users/Tasleem/Github/.git/
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ vi Testfile1.sh
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Activate Windows
Go to Settings to activate Windows.

Step-6: Create File 2 “Testfile2.sh”

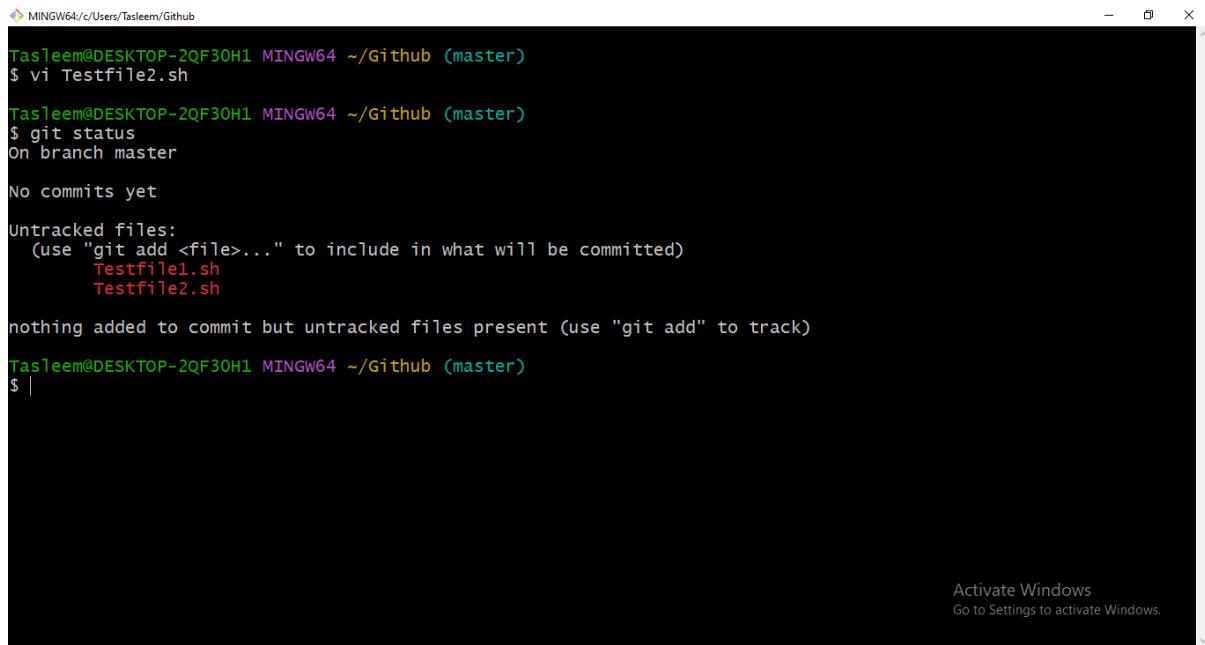


```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ vi Testfile2.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Activate Windows
Go to Settings to activate Windows.

Step-7: Run git status



```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ vi Testfile2.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Testfile1.sh
    Testfile2.sh

nothing added to commit but untracked files present (use "git add" to track)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Activate Windows
Go to Settings to activate Windows.

Step-8: Run “git add .” to add the files to staging area.

The screenshot shows a terminal window with the following command history:

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ vi Testfile2.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Testfile1.sh
    Testfile2.sh

nothing added to commit but untracked files present (use "git add" to track)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git add .
warning: in the working copy of 'Testfile1.sh', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Testfile2.sh', LF will be replaced by CRLF the next time Git touches it

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

At the bottom right of the terminal window, there is a watermark that says "Activate Windows Go to Settings to activate Windows."

Step-9: Run command ‘git commit -m “first commit”’

The screenshot shows a terminal window with the following command history:

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ vi Testfile2.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Testfile1.sh
    Testfile2.sh

nothing added to commit but untracked files present (use "git add" to track)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git add .
warning: in the working copy of 'Testfile1.sh', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Testfile2.sh', LF will be replaced by CRLF the next time Git touches it

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git commit -m "First Commit"
[master (root-commit) 14240ed] First Commit
 2 files changed, 3 insertions(+)
 create mode 100644 Testfile1.sh
 create mode 100644 Testfile2.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

At the bottom right of the terminal window, there is a watermark that says "Activate Windows Go to Settings to activate Windows."

Step-10: Edit the first file and stage it and commit.

Enter second line.

A screenshot of a terminal window titled "MINGW64:/c/Users/Tasleem/Github". The window contains the following text:

```
#!/bin/bash
echo "Hello World!"
echo "How are you!"
```

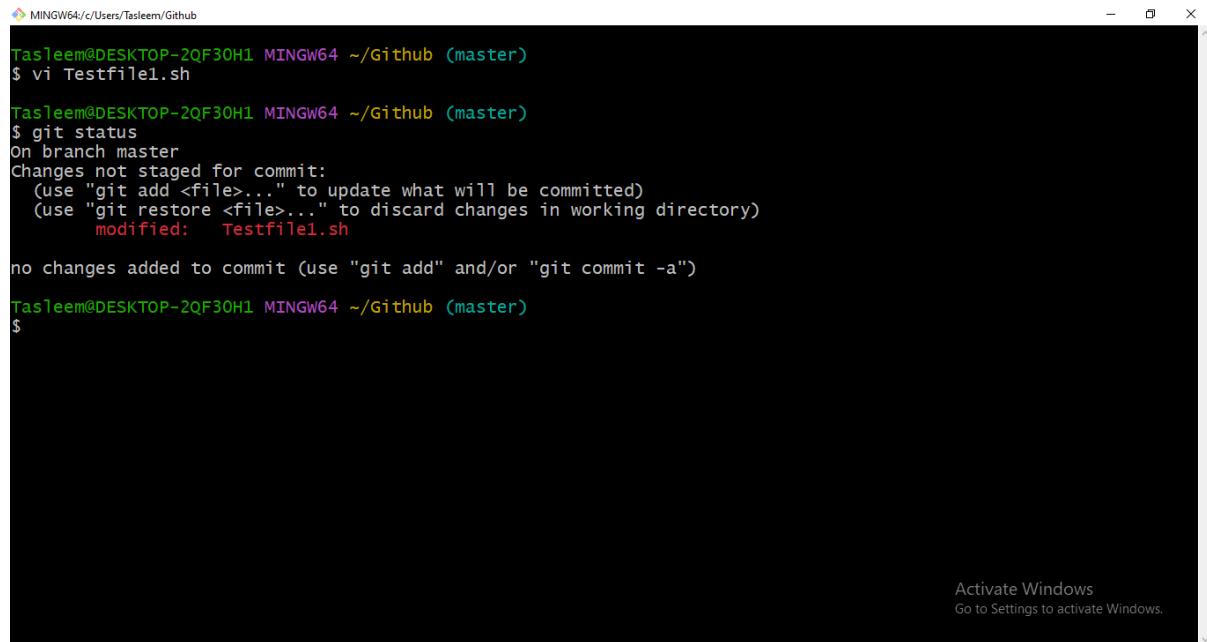
The terminal shows a cursor at the end of the second line. At the bottom of the window, there is a status bar with the text "Testfile1.sh [dos] (19:04 16/06/2024)" and "Testfile1.sh" [dos] 3L, 55B". In the bottom right corner, there is a watermark that says "Activate Windows 1,1 A11 Go to Settings to activate Windows."

A screenshot of a terminal window titled "MINGW64:/c/Users/Tasleem/Github". The window contains the following text:

```
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ vi Testfile1.sh
```

The terminal shows a cursor at the end of the command "vi Testfile1.sh". At the bottom of the window, there is a status bar with the text "Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)" and "\$". In the bottom right corner, there is a watermark that says "Activate Windows Go to Settings to activate Windows."

Step-11: Run “git status”



```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ vi Testfile1.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ 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:   Testfile1.sh

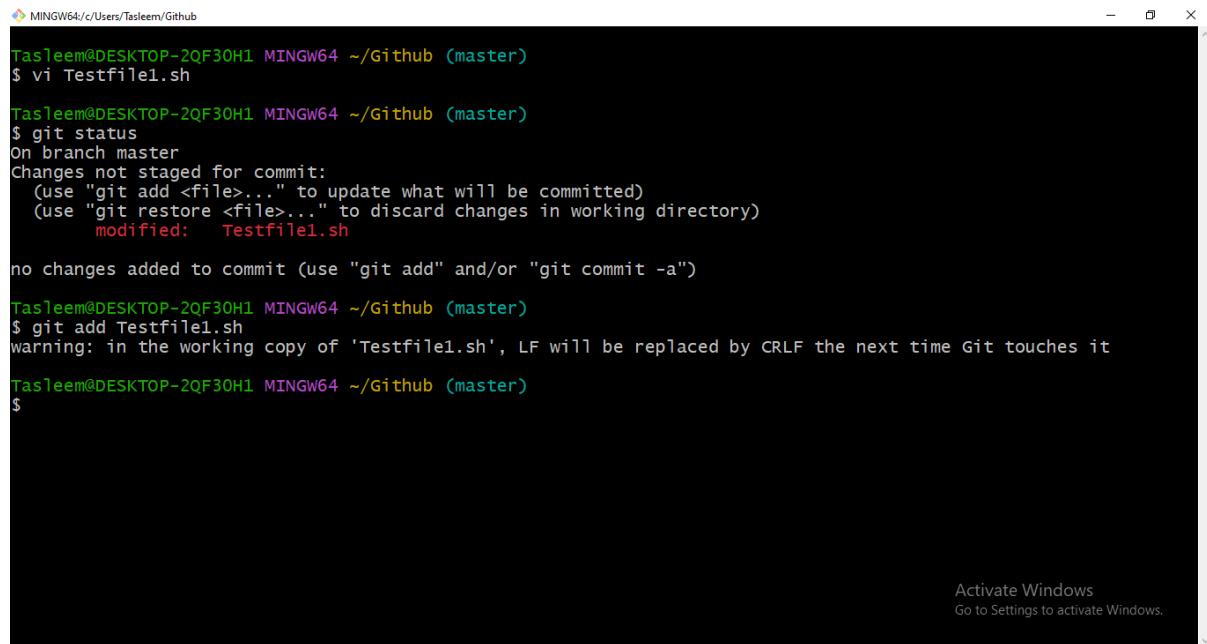
no changes added to commit (use "git add" and/or "git commit -a")

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$
```

Activate Windows
Go to Settings to activate Windows.

We can see the file is modified, so now we need to add it to staging area and commit.

File Added to the staging area



```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ vi Testfile1.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ 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:   Testfile1.sh

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

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git add Testfile1.sh
warning: in the working copy of 'Testfile1.sh', LF will be replaced by CRLF the next time Git touches it

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$
```

Activate Windows
Go to Settings to activate Windows.

Step-12: Run command “git commit -m “second commit”

The screenshot shows a terminal window with the following session:

```
MINGW64:/c/Users/Tasleem/Github
$ vi Testfile1.sh
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ 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:   Testfile1.sh

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

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git add Testfile1.sh
warning: in the working copy of 'Testfile1.sh', LF will be replaced by CRLF the next time Git touches it

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git commit -m "second commit"
[master 6817aec] second commit
 1 file changed, 1 insertion(+)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$
```

At the bottom right of the terminal window, there is a watermark that says "Activate Windows Go to Settings to activate Windows."

Step-13: Now execute command “git log --oneline” to see the commit id.

The screenshot shows a terminal window with the following session:

```
MINGW64:/c/Users/Tasleem/Github
$ vi Testfile1.sh
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ 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:   Testfile1.sh

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

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git add Testfile1.sh
warning: in the working copy of 'Testfile1.sh', LF will be replaced by CRLF the next time Git touches it

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git commit -m "second commit"
[master 6817aec] second commit
 1 file changed, 1 insertion(+)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
6817aec (HEAD -> master) second commit
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

At the bottom right of the terminal window, there is a watermark that says "Activate Windows Go to Settings to activate Windows."

Using commit id we can revert the particular commit executed.

Step-14: Run the command “git revert <commit Id>”

```
MINGW64:/c/Users/Tasleem/Github
g
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
6817aec (HEAD -> master) second commit
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git revert 6817aec
```

```
MINGW64/c/Users/Tasleem/Github
g
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
6817aec (HEAD -> master) second commit
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git revert 6817aec
[master 101ac0d] Revert "second commit"
 1 file changed, 1 deletion(-)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Open the file and check if the added line is deleted as revert command executed.

We can see the commit has been reverted

```
MINGW64:/c/Users/Tasleem/Github
g
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
6817aec (HEAD -> master) second commit
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git revert 6817aec
[master 101ac0d] Revert "second commit"
 1 file changed, 1 deletion(-)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
101ac0d (HEAD -> master) Revert "second commit"
6817aec Second commit
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Added line has been removed after revert command executed.

The "reset" command is used to undo changes to the current state of the repository in various ways. We have 3 types of methods;

1. Soft Reset
2. Hard Reset
3. Mixed Reset (Default)

“Git reset --soft” resets the HEAD to the specified commit, but keeps the changes in the staging area. It essentially undoes the commit and leaves the changes staged.

“git reset --hard” will delete uncommitted changes permanently.

“git reset” resets the HEAD to the specified commit and unstages the changes. The changes will still be present in the working directory, but not in the index. This is useful if we want to rework the commit before committing again.

Step-15: Create a file “Testfile3.sh”



A screenshot of a terminal window titled "MINGW64/c/Users/Tasleem/Github". The window shows the command \$ vi Testfile3.sh being run. The terminal is black with white text. At the bottom right, there is a watermark that says "Activate Windows Go to Settings to activate Windows."

```
MINGW64/c/Users/Tasleem/Github
TasTeem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ vi Testfile3.sh

TasTeem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Step-16: Now run command to add the file to staging area and then commit the file.

The screenshot shows a terminal window titled 'MINGW64/c/Users/Tasleem/Github'. The user runs several commands to track and commit a file:

```
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Testfile3.sh

nothing added to commit but untracked files present (use "git add" to track)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git add .
warning: in the working copy of 'Testfile3.sh', LF will be replaced by CRLF the next time Git touches it

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git commit -m "Testfile3.sh for Reset"
[master 08be5d6] Testfile3.sh for Reset
 1 file changed, 1 insertion(+)
 create mode 100644 Testfile3.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

In the bottom right corner of the terminal window, there is a watermark that says 'Activate Windows' and 'Go to Settings to activate Windows.'

Step-17: Run the command “git log --oneline” to check the commit id.

The screenshot shows a terminal window titled 'MINGW64/c/Users/Tasleem/Github'. The user runs the 'git log --oneline' command to see the commit history:

```
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Testfile3.sh

nothing added to commit but untracked files present (use "git add" to track)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git add .
warning: in the working copy of 'Testfile3.sh', LF will be replaced by CRLF the next time Git touches it

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git commit -m "Testfile3.sh for Reset"
[master 08be5d6] Testfile3.sh for Reset
 1 file changed, 1 insertion(+)
 create mode 100644 Testfile3.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
08be5d6 (HEAD -> master) Testfile3.sh for Reset
611950f Reapply "second commit"
101ac0d Revert "second commit"
6817aec second commit
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

In the bottom right corner of the terminal window, there is a watermark that says 'Activate Windows' and 'Go to Settings to activate Windows.'

Step-18: Run the command “git reset --soft <commit id>” to undo the commit and leave the changes staged.

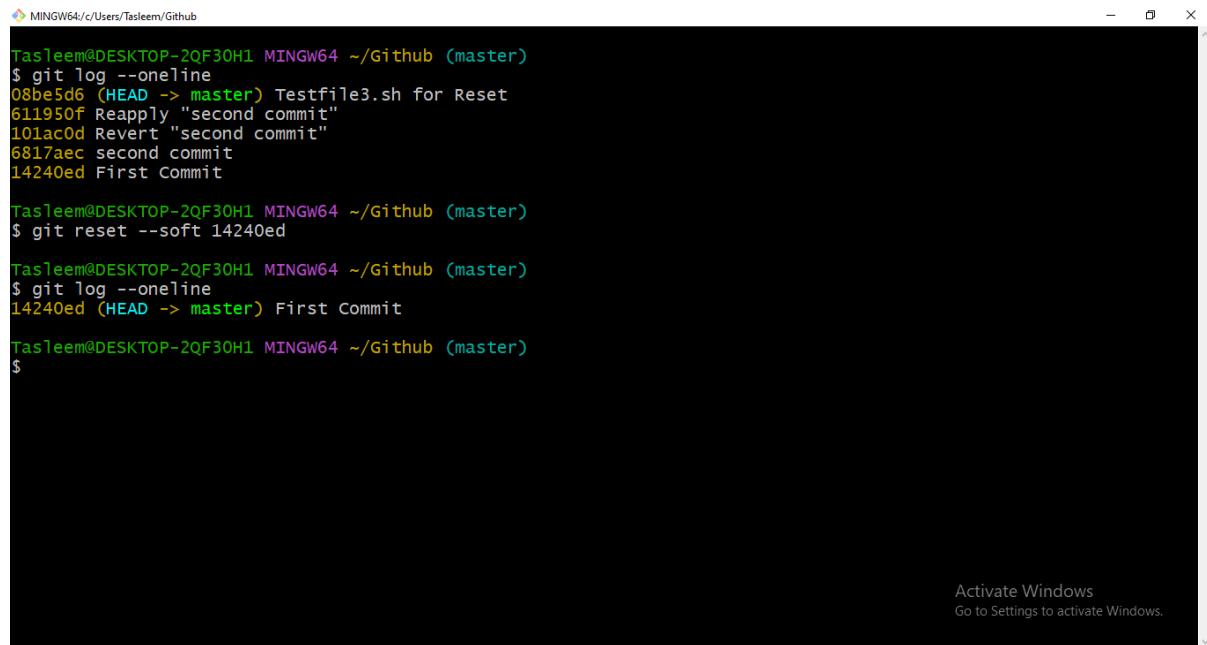


```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
08be5d6 (HEAD -> master) Testfile3.sh for Reset
611950f Reapply "second commit"
101ac0d Revert "second commit"
6817aec second commit
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git reset --soft 14240ed

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Activate Windows
Go to Settings to activate Windows.



```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
08be5d6 (HEAD -> master) Testfile3.sh for Reset
611950f Reapply "second commit"
101ac0d Revert "second commit"
6817aec second commit
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git reset --soft 14240ed

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
14240ed (HEAD -> master) First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Activate Windows
Go to Settings to activate Windows.

Step-19: Run command “git log –oneline”, Now we can see only one commit.

When we run git status, we can see the other files have been staged but uncommitted.



```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
08be5d6 (HEAD -> master) Testfile3.sh for Reset
611950f Reapply "second commit"
101ac0d Revert "second commit"
6817aec second commit
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git reset --soft 14240ed

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
14240ed (HEAD -> master) First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   Testfile1.sh
    new file:   Testfile3.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$
```

Activate Windows
Go to Settings to activate Windows.

Step-20: Now if we execute command “git reset --hard <commit id>” then the file will be deleted permanently.



```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
bd310df (HEAD -> master) reset files
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ ls
Testfile1.sh* Testfile2.sh Testfile3.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git reset --hard 14240ed

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$
```

Activate Windows
Go to Settings to activate Windows.

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
bd310df (HEAD -> master) reset files
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ ls
Testfile1.sh* Testfile2.sh Testfile3.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git reset --hard 14240ed
HEAD is now at 14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$
```

Activate Windows
Go to Settings to activate Windows.

We can see the Testfile3.sh has been deleted permanently

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
bd310df (HEAD -> master) reset files
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ ls
Testfile1.sh* Testfile2.sh Testfile3.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git reset --hard 14240ed
HEAD is now at 14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ ls
Testfile1.sh* Testfile2.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Activate Windows
Go to Settings to activate Windows.

We can also check in the staging area the file is not available



```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
bd310df (HEAD -> master) reset files
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ ls
Testfile1.sh* Testfile2.sh Testfile3.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git reset --hard 14240ed
HEAD is now at 14240ed First Commit

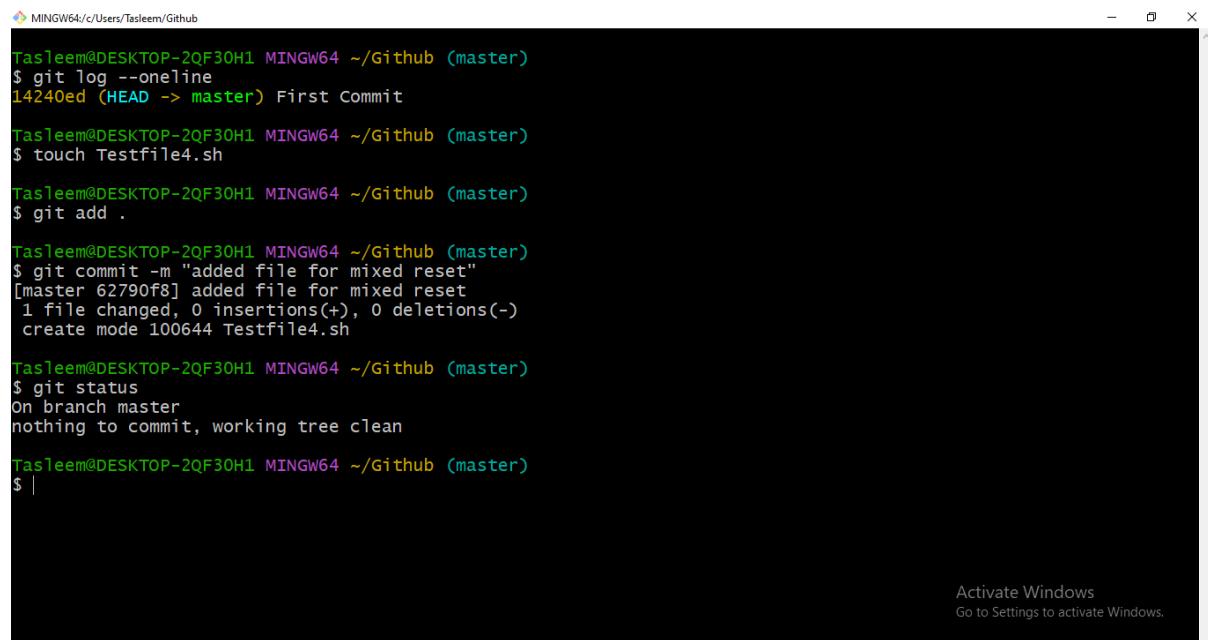
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ ls
Testfile1.sh* Testfile2.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git status
On branch master
nothing to commit, working tree clean

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$
```

Activate Windows
Go to Settings to activate Windows.

Step-21: Create a file and add the file to staging area and then commit.



```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
14240ed (HEAD -> master) First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ touch Testfile4.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git add .

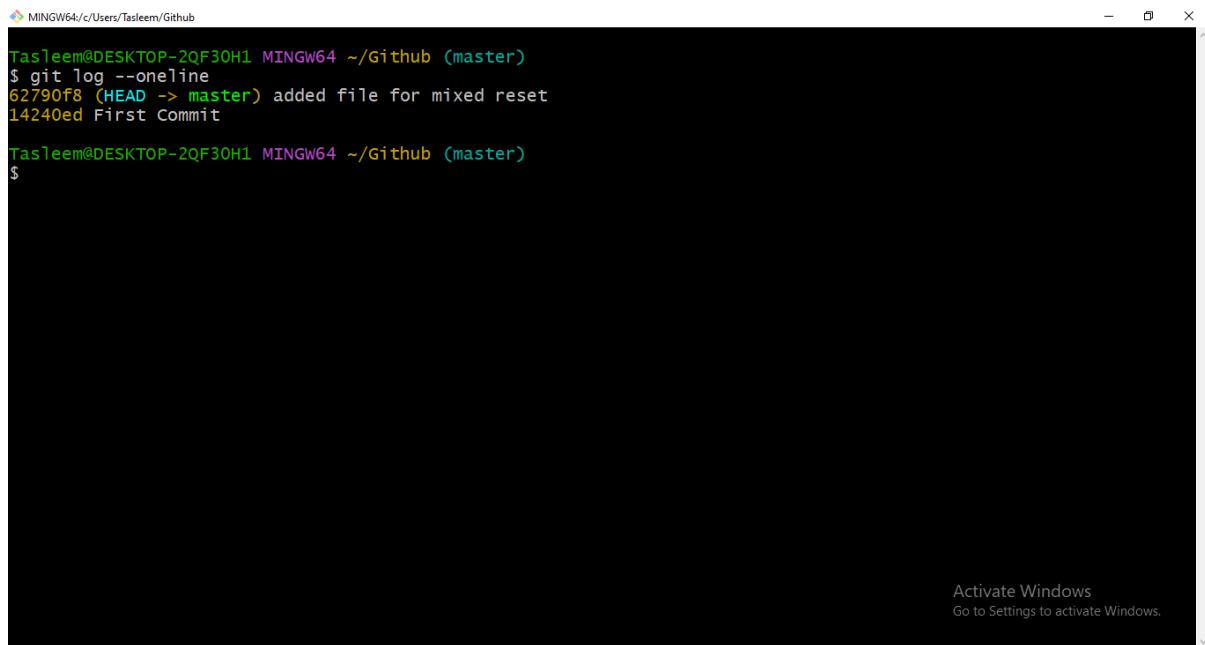
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git commit -m "added file for mixed reset"
[master 62790f8] added file for mixed reset
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Testfile4.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git status
On branch master
nothing to commit, working tree clean

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Activate Windows
Go to Settings to activate Windows.

We can see 2 commits



```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
62790f8 (HEAD -> master) added file for mixed reset
14240ed First Commit

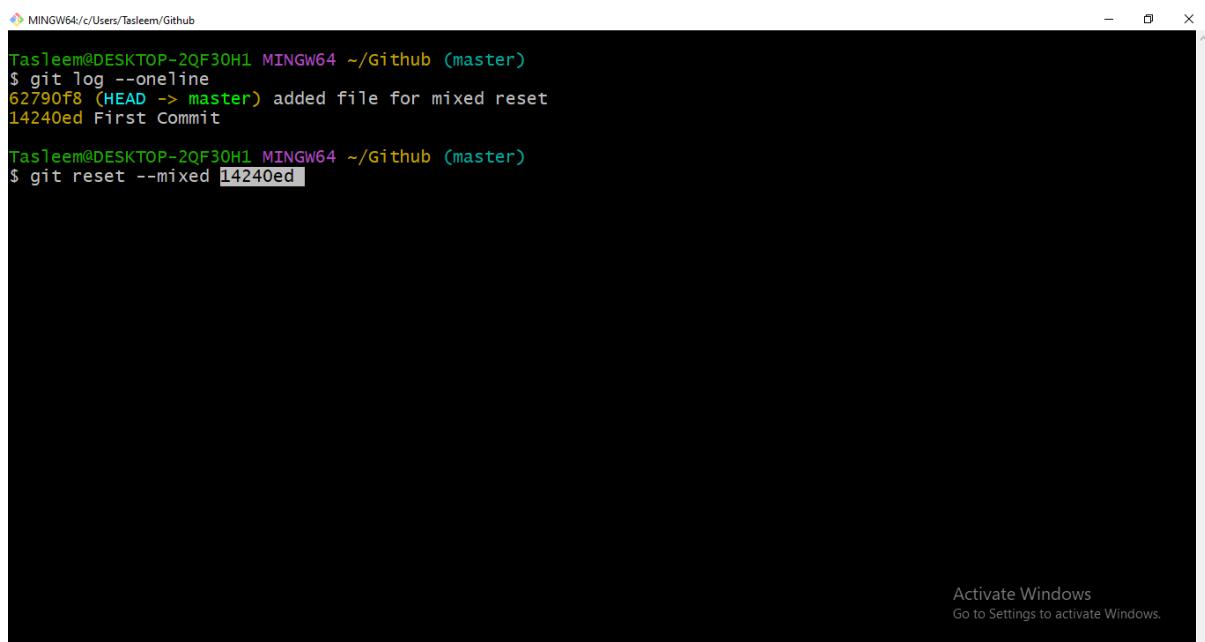
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$
```

The terminal window shows the output of the command `git log --oneline`. It displays two commits:

- The first commit is `62790f8`, which added a file for mixed reset.
- The second commit is `14240ed`, labeled "First Commit".

In the bottom right corner of the terminal window, there is a watermark that says "Activate Windows" and "Go to Settings to activate Windows."

Step-22: Run command “`git reset --mixed <commit id>`”



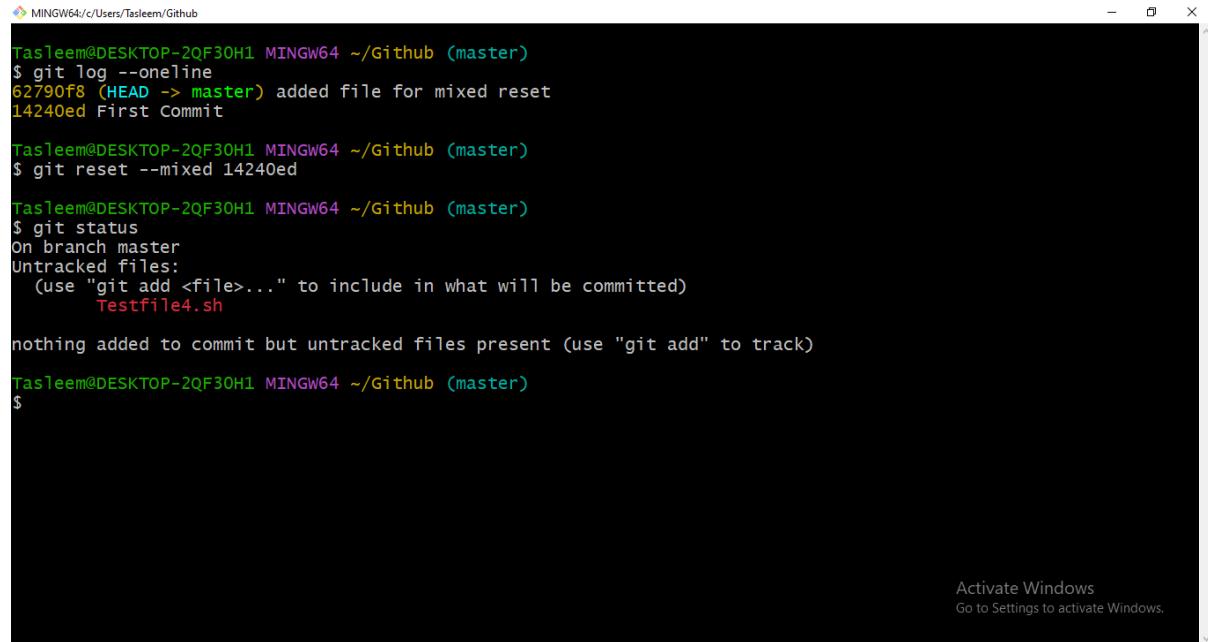
```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
62790f8 (HEAD -> master) added file for mixed reset
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git reset --mixed 14240ed
```

The terminal window shows the command `git reset --mixed 14240ed` being entered at the prompt. The commit ID `14240ed` is highlighted in red, indicating it is the target of the reset operation.

In the bottom right corner of the terminal window, there is a watermark that says "Activate Windows" and "Go to Settings to activate Windows."

We can see that the Testfile4.sh has been added back to the working directory.



```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git log --oneline
62790f8 (HEAD -> master) added file for mixed reset
14240ed First Commit

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git reset --mixed 14240ed

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Testfile4.sh

nothing added to commit but untracked files present (use "git add" to track)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$
```

Activate Windows
Go to Settings to activate Windows.

Git reset and revert are both commands used to undo changes in Git repositories, but they operate in fundamentally different ways.

Revert creates a new commit that undoes the changes made by a specific commit. Useful when we want to undo changes but keep a record of the fact that those changes were once part of the history.

Reset changes the history of commits in the repository by moving the branch pointer to a different commit. It can remove commits entirely or move the branch to a previous state. Can be used to undo commits, reset the staging area, or move the current branch to a different commit.

L2 - Create Local git repository and demonstrate git merge and Merge Conflicts with the steps to resolve merge conflicts

Step-1: Create a file and add file to staging area and commit.

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git init
Initialized empty Git repository in C:/Users/Tasleem/Github/.git/
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Test1.sh
    Test2.sh

nothing added to commit but untracked files present (use "git add" to track)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git add .
warning: in the working copy of 'Test1.sh', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'Test2.sh', LF will be replaced by CRLF the next time Git touches it

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git commit -m "Added 2 test files"
[master (root-commit) e442138] Added 2 test files
 2 files changed, 2 insertions(+)
 create mode 100644 Test1.sh
 create mode 100644 Test2.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$
```

Activate Windows
Go to Settings to activate Windows.

Step-2: Create 2 Branches branch1 and branch2

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git branch
* master

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git branch branch1

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git branch branch2

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git branch
  branch1
  branch2
* master

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ |
```

Activate Windows
Go to Settings to activate Windows.

Step-3: Go to branch1

The screenshot shows a terminal window with the following command history:

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git branch
* master

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git branch branch1

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git branch branch2

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git branch
  branch1
  branch2
* master

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (master)
$ git checkout branch1
Switched to branch 'branch1'

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch1)
$
```

In the bottom right corner of the terminal window, there is a watermark that reads "Activate Windows" and "Go to Settings to activate Windows".

Step-4: Edit the file Test1.sh to add the content

The screenshot shows a terminal window with the following command:

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch1)
$ vi Test1.sh
```

In the bottom right corner of the terminal window, there is a watermark that reads "Activate Windows" and "Go to Settings to activate Windows".

Step-5: Add the file to staging area and commit the changes

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch1)
$ vi Test1.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch1)
$ git status
On branch branch1
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:   Test1.sh

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

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch1)
$ git add .
warning: in the working copy of 'Test1.sh', LF will be replaced by CRLF the next time Git touches it

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch1)
$ git commit -m "Added File1"
[branch1 58d60a0] Added File1
 1 file changed, 1 insertion(+)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch1)
$ git status
On branch branch1
nothing to commit, working tree clean

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch1)
$ |
```

Step-6: Switch to branch2 and edit the Test1.sh file

```
MINGW64/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch1)
$ git checkout branch2
Switched to branch 'branch2'

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ ls
Test1.sh  Test2.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ |
```

Step-7: Add the file to staging area and commit the changes

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ git status
On branch branch2
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: Test1.sh

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

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ git add .

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ git commit -m "Added file1 from branch2"
[branch2 f59a67d] Added file1 from branch2
 1 file changed, 1 insertion(+)

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ git status
On branch branch2
nothing to commit, working tree clean

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$
```

Activate Windows
Go to Settings to activate Windows.

Step-8: Now Run the command “git merge <branch1> <branch2>”

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ git merge branch2 branch1
Auto-merging Test1.sh
CONFLICT (content): Merge conflict in Test1.sh
Automatic merge failed; fix conflicts and then commit the result.

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2|MERGING)
$ |
```

Activate Windows
Go to Settings to activate Windows.

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ git merge branch2 branch1
Auto-merging Test1.sh
CONFLICT (content): Merge conflict in Test1.sh
Automatic merge failed; fix conflicts and then commit the result.

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2|MERGING)
$
```

As the file name is same but the content is different in both the branches, it will show an error. So, resolving the error manually-

Step-9: Edit the file

Removed all the symbols and left the data which is required.

Step-10: Execute “git status”

```
MINGW64/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ git merge branch2 branch1
Auto-merging Test1.sh
CONFLICT (content): Merge conflict in Test1.sh
Automatic merge failed; fix conflicts and then commit the result.

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2|MERGING)
$ vi Test1.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2|MERGING)
$ git status
On branch branch2
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:  Test1.sh

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

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2|MERGING)
$
```

Step-11: Add the file to staging area and commit the changes

The screenshot shows a terminal window with the following command history:

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2|MERGING)
$ git status
On branch branch2
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  modified:  Test1.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2|MERGING)
$ git commit -m "conflicts resolved"
[branch2 355ea07] conflicts resolved

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ |
```

In the bottom right corner of the terminal window, there is a watermark that reads "Activate Windows Go to Settings to activate Windows."

#Conflicts Resolved.

The screenshot shows a terminal window with the following command history:

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2|MERGING)
$ git status
On branch branch2
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  modified:  Test1.sh

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2|MERGING)
$ git commit -m "conflicts resolved"
[branch2 355ea07] conflicts resolved

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ git status
On branch branch2
nothing to commit, working tree clean

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ |
```

In the bottom right corner of the terminal window, there is a watermark that reads "Activate Windows Go to Settings to activate Windows."

```
MINGW64:/c/Users/Tasleem/Github
Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$ git merge branch1
Already up to date.

Tasleem@DESKTOP-2QF30H1 MINGW64 ~/Github (branch2)
$
```

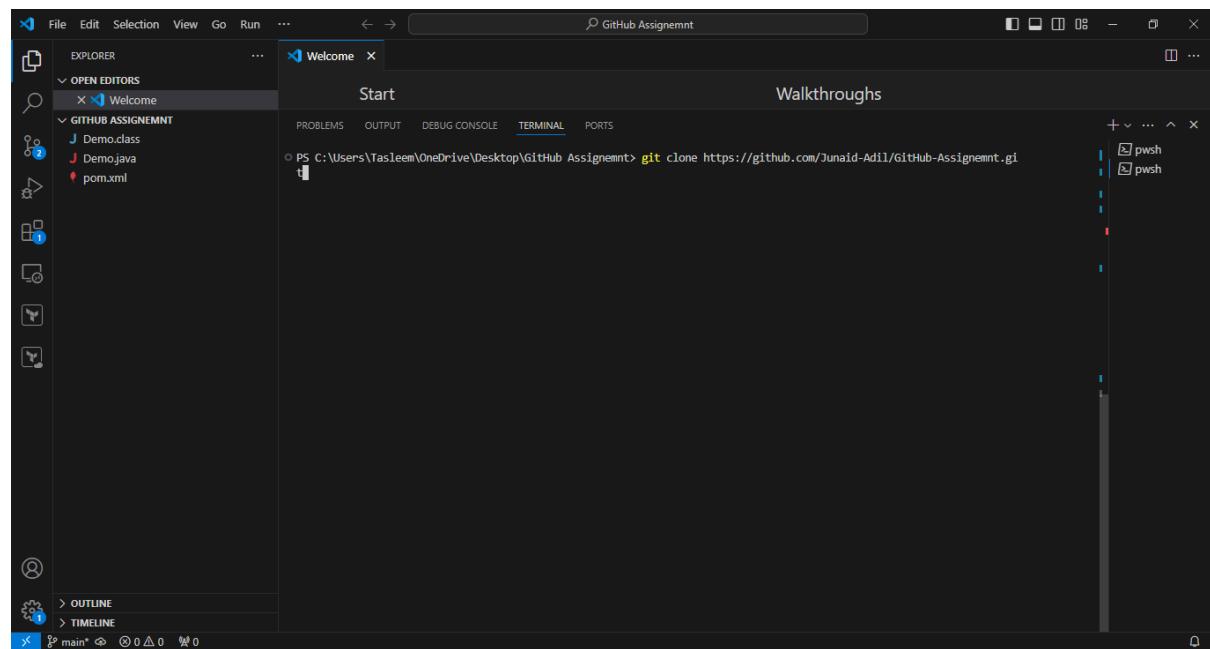
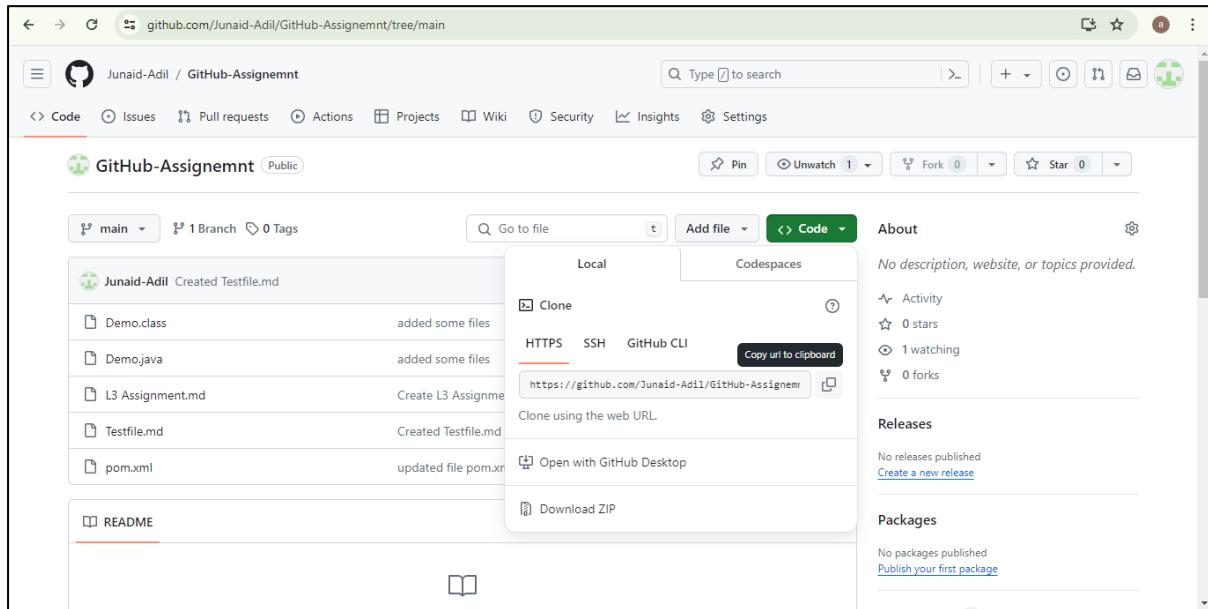
Activate Windows
Go to Settings to activate Windows.

L3 - Using Local and Remote git repositories demonstrate git pull and git fetch.

Compare the differences

Step-1: Create a Local and Github Repository, and Clone using HTTPS.

Github Repository:



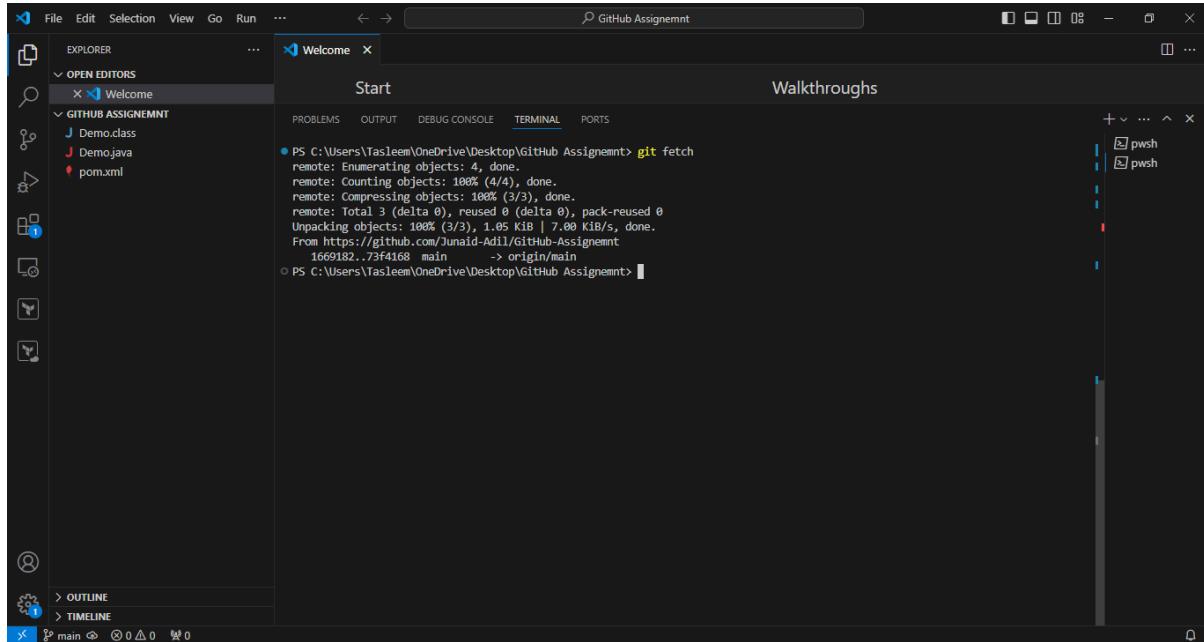
A screenshot of the Visual Studio Code (VS Code) interface. The title bar says "GitHub Assignment". The left sidebar shows an "EXPLORER" view with a folder named "GitHub Assignment" containing files "GitHub Assignment", "Demo.class", "Demo.java", and "pom.xml". The main area is titled "Welcome" with tabs for "Start" and "Walkthroughs". A "TERMINAL" tab is active, displaying the command line output of a "git clone" operation:

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git clone https://github.com/Junaid-Adil/GitHub-Assiginemnt.git
Cloning into 'GitHub-Assiginemnt'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 14 (delta 4), reused 7 (delta 2), pack-reused 0
Receiving objects: 100% (14/14), done.
Resolving deltas: 100% (4/4), done.
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment>
```

Step-2: Create a file in Github Repository, File – “L3 Assignment.md”

A screenshot of a GitHub repository page. The URL is "github.com/Junaid-Adil/GitHub-Assiginemnt/tree/main". The repository name is "GitHub-Assiginemnt". The "Code" tab is selected. In the code editor, a new file named "L3 Assignment.md" is visible. The commit message for this file is "Create L3 Assignment.md". Other files listed include "Demo.class", "Demo.java", and "pom.xml". The commit was made by "Junaid-Adil" at "73f4168 · now".

Step-3: Now run “Git fetch” command



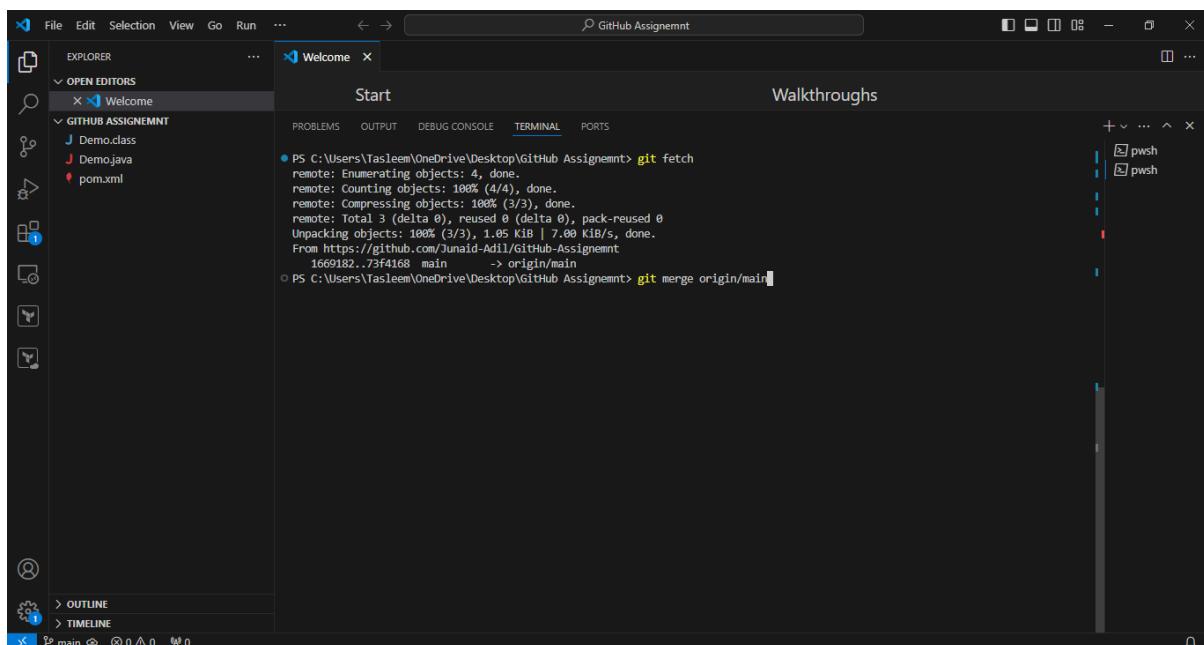
The screenshot shows the Visual Studio Code interface with the terminal tab selected. The command `git fetch` is being run in the terminal, and the output shows the process of fetching objects from the remote repository. The terminal window has a dark theme and displays the following text:

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.05 KiB | 7.00 KiB/s, done.
From https://github.com/Junaid-Adil/GitHub-Assigment
 1669182..73F4168 main      -> origin/main
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment>
```

“Git fetch” command is used to fetch updates from a remote repository but does not automatically merge them.

To get the file in local Repository we have to run command “git merge origin/main”

Step-4: Run command “git merge origin/main”



The screenshot shows the Visual Studio Code interface with the terminal tab selected. The command `git merge origin/main` is being run in the terminal, and the output shows the merge process. The terminal window has a dark theme and displays the following text:

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.05 KiB | 7.00 KiB/s, done.
From https://github.com/Junaid-Adil/GitHub-Assigment
 1669182..73F4168 main      -> origin/main
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git merge origin/main
```

Now we can see the file “L3 Assignment.md” in Local Repository.

The screenshot shows the Visual Studio Code interface with the terminal tab active. The terminal window displays the following command-line session:

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.05 KiB | 7.00 KiB/s, done.
From https://github.com/Junaid-Adil/GitHub-Assigment
  1669182..73f4168 main      -> origin/main
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git merge origin/main
Updating 1669182..73f4168
Fast-forward
 L3 Assignment.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 L3 Assignment.md
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment>
```

Step-5: Run “git log” to view the commit history of a repository

The screenshot shows the Visual Studio Code interface with the terminal tab active. The terminal window displays the following command-line session, showing the commit history:

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.05 KiB | 7.00 KiB/s, done.
From https://github.com/Junaid-Adil/GitHub-Assigment
  1669182..73f4168 main      -> origin/main
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git merge origin/main
Updating 1669182..73f4168
Fast-forward
 L3 Assignment.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 L3 Assignment.md
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git log
```

```

remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.05 KiB | 7.00 KiB/s, done.
From https://github.com/Junaaid-Adil/GitHub-Assiginemnt
 1669182..73f4168 main      -> origin/main
Updating 1669182..73f4168
Fast-forward
 L3 Assignment.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 L3 Assignment.md
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assiginemnt> git log
commit 16691827c67a68b89ae309aa326b204000a5a819
Author: junaaid <junaaid@gmail.com>
Date:   Sat Jun 15 20:07:38 2024 +0530

        updated file pom.xml

commit 44830f55083d7a68450cb7a1a4e05c6e49b5fd76
Author: junaaid <junaaid@gmail.com>
Date:   Sat Jun 15 19:49:50 2024 +0530

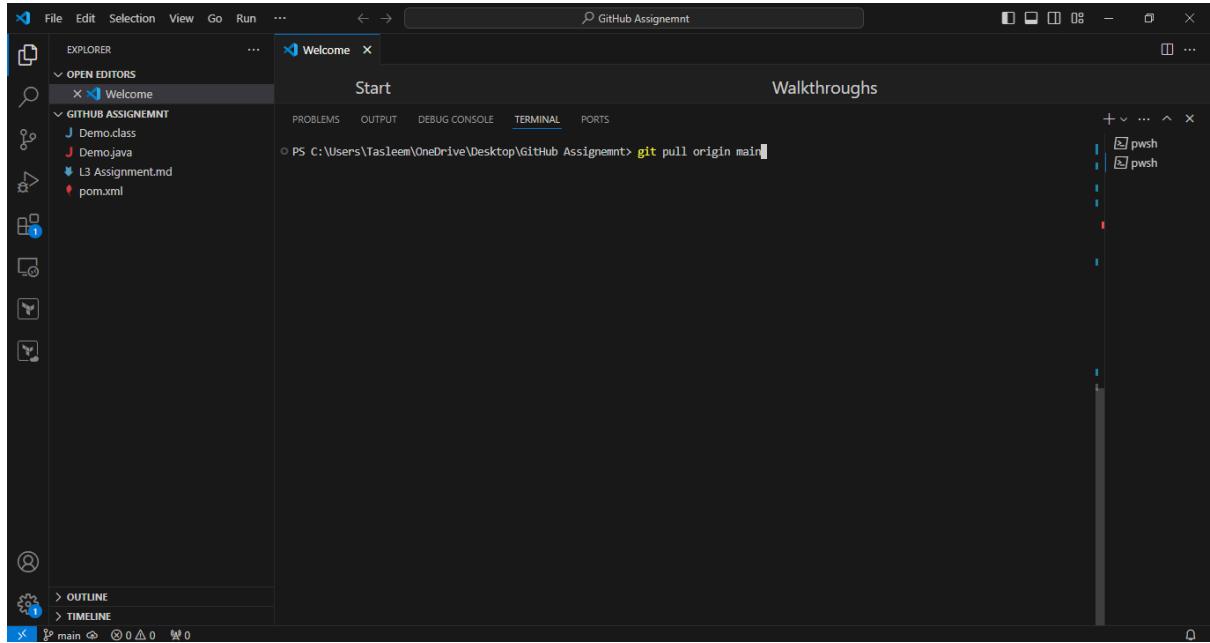
        added some files
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assiginemnt>

```

Step-6: Create a new file “Testfile.md” in Remote Repository

Name	Last commit message	Last commit date
Demo.class	added some files	20 hours ago
Demo.java	added some files	20 hours ago
L3 Assignment.md	Create L3 Assignment.md	9 minutes ago
Testfile.md	Created Testfile.md	now
pom.xml	updated file pom.xml	20 hours ago

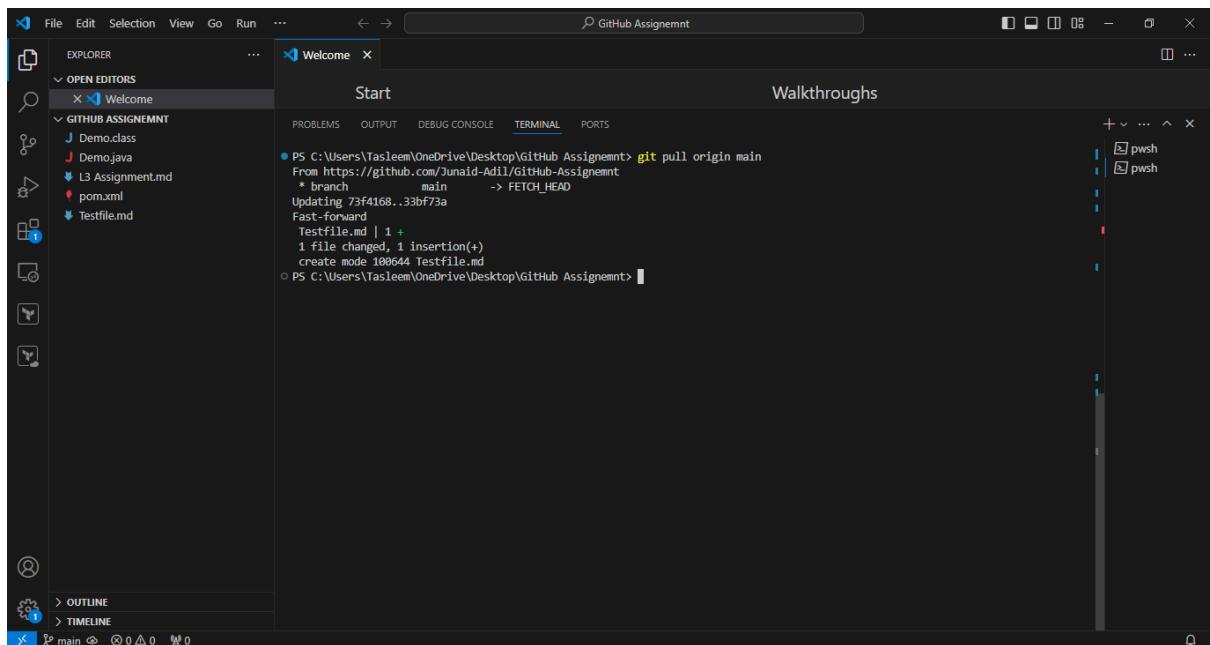
Step-7: Now run the command “git pull origin main”



The screenshot shows the VS Code interface with the terminal tab selected. The command `git pull origin main` is being run in the terminal. The output shows the repository is up-to-date, indicating no changes were pulled.

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git pull origin main
From https://github.com/Junaid-Adil/GitHub-Assigment
 * branch            main      -> FETCH_HEAD
Updating 73f4168..33bf73a
Fast-forward
 Testfile.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Testfile.md
```

We can see the file Testfile.md in Local Repository



The screenshot shows the VS Code interface with the terminal tab selected. The command `git pull origin main` is being run in the terminal. The output shows the repository has been updated from the remote branch, with a new file `Testfile.md` created.

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git pull origin main
From https://github.com/Junaid-Adil/GitHub-Assigment
 * branch            main      -> FETCH_HEAD
Updating 73f4168..33bf73a
Fast-forward
 Testfile.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Testfile.md
```

Difference between git pull and git fetch:

git pull is used to fetch and download data from a remote repository and immediately update the local repository.

It performs two operations, **git fetch** to retrieve the latest changes from the remote repository and **git merge** to integrate those changes into the current branch.

git fetch is used to retrieve new commits from a remote repository without automatically merging them into the local Repository.

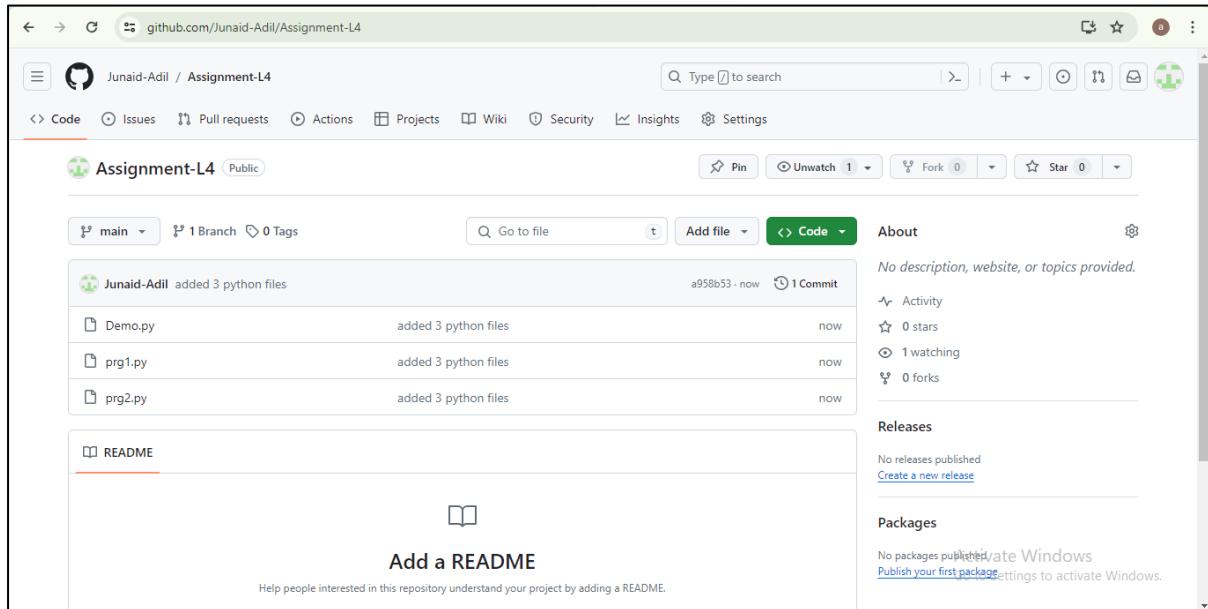
It updates the remote tracking branches (origin/main) to reflect the latest changes in the remote repository.

git fetch is for fetching and reviewing changes before integration.

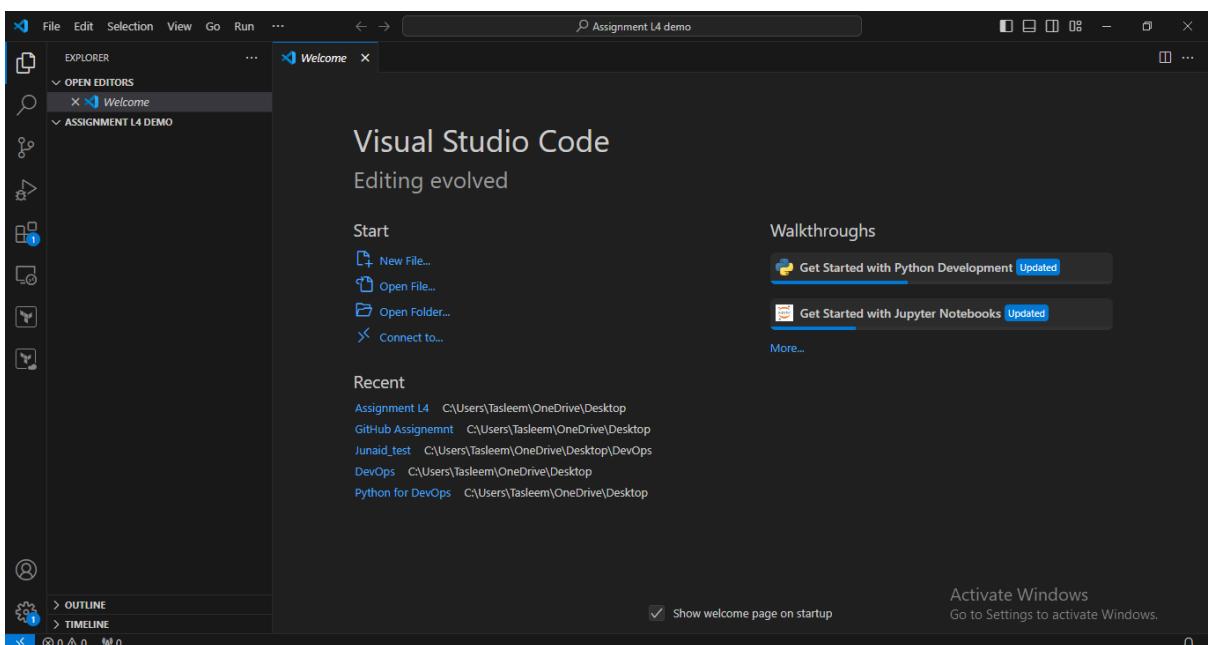
git pull is for quick updates and merges.

L4 - Clone GitHub repository using Visual Studio Code IDE

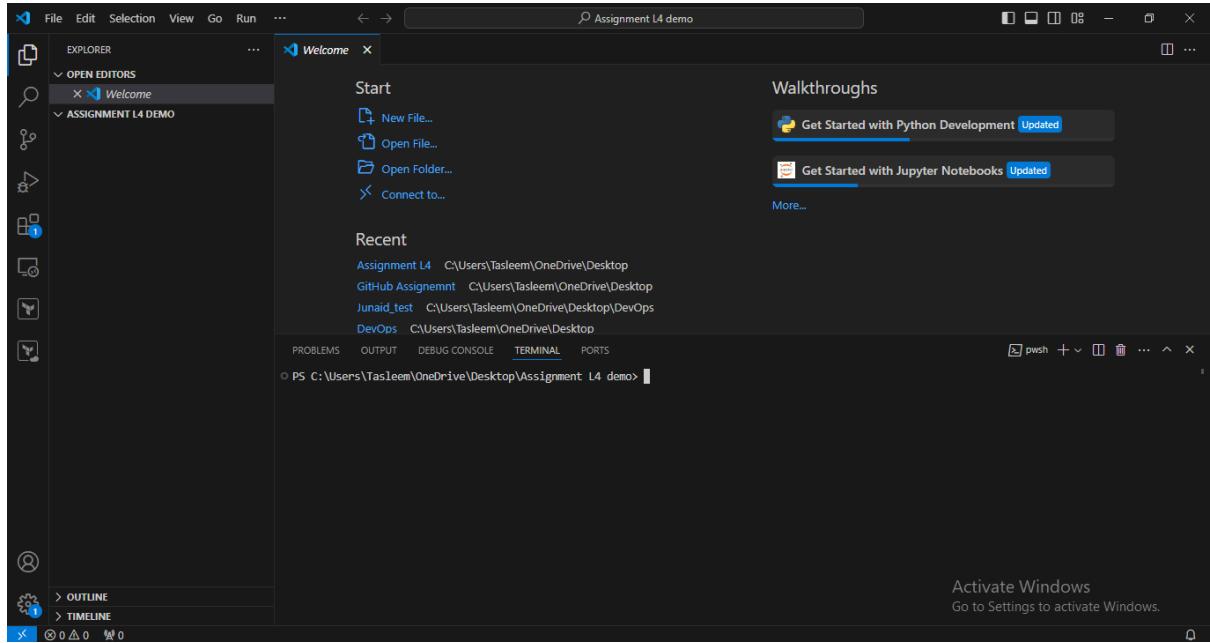
Step 1: Create a GitHub repository – “Assignment L4” with 3 python files init.



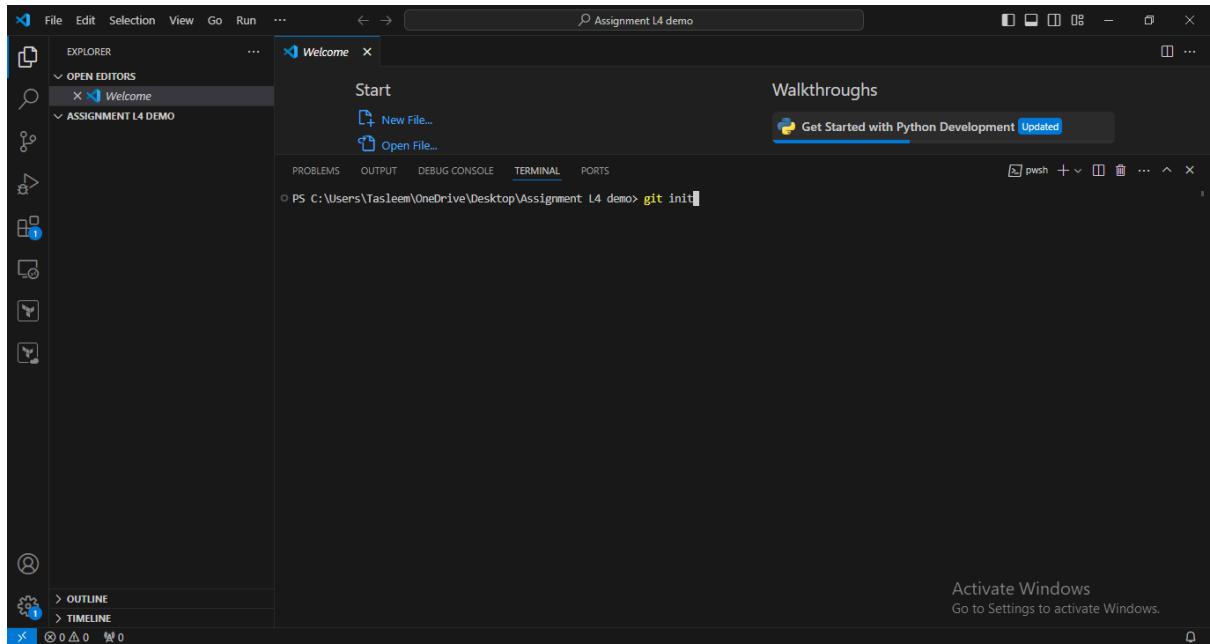
Step 2: To clone the Github repository using VSC, open Visual Studio Code and open the folder.

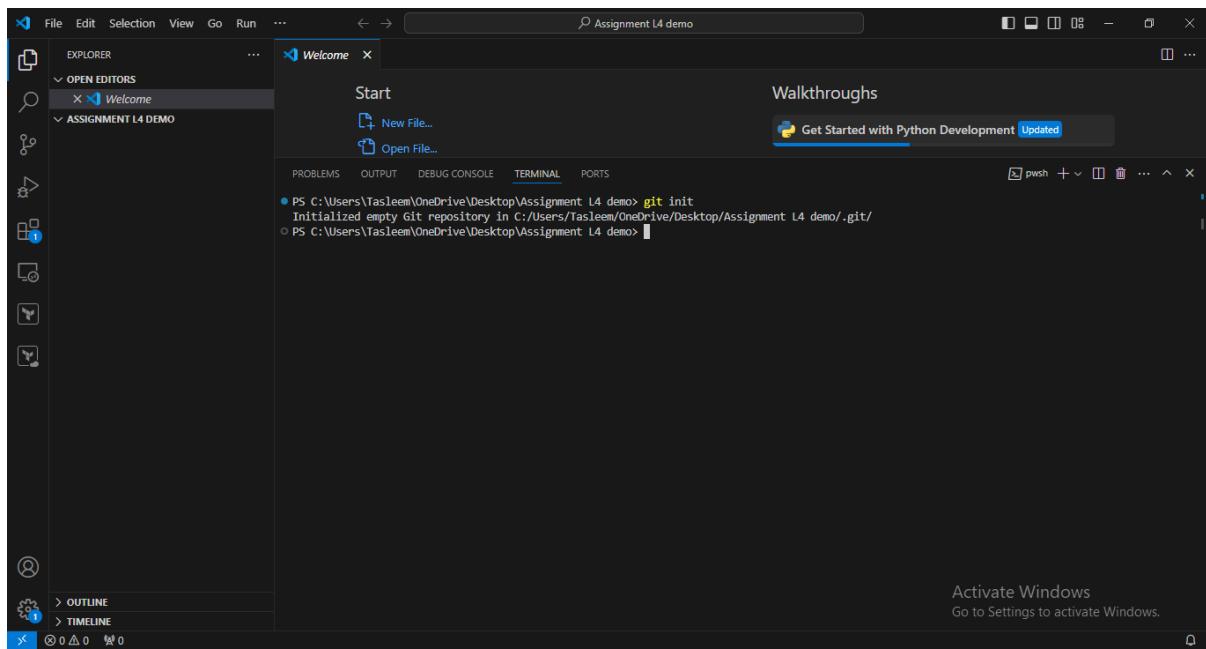


Step 3: Open the terminal

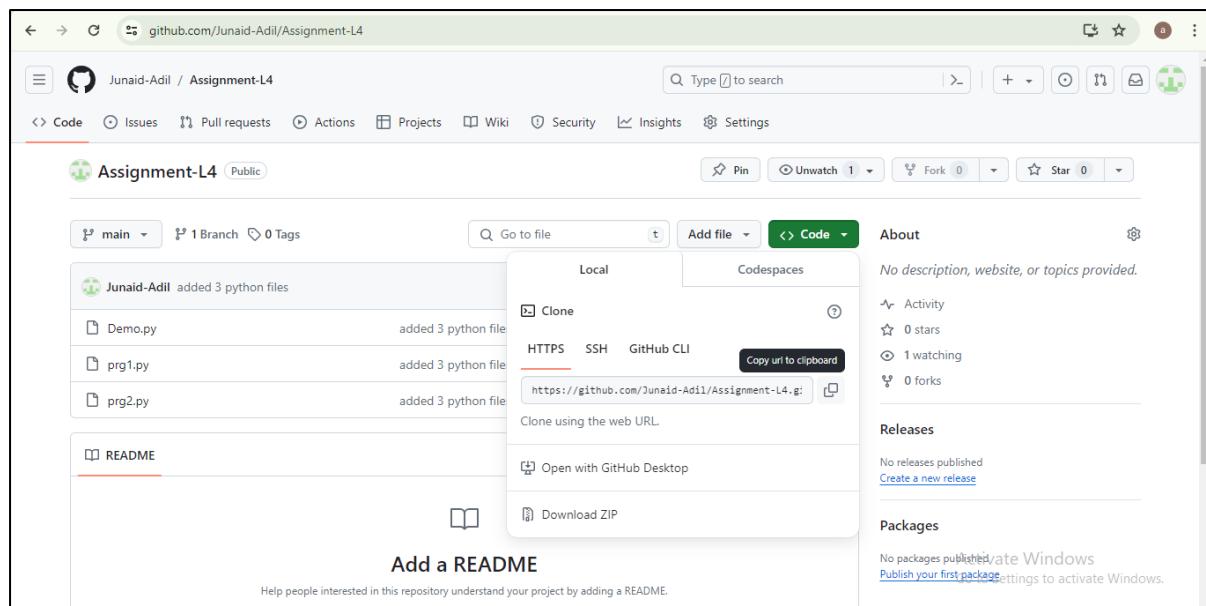


Step 4: Initialize git in it

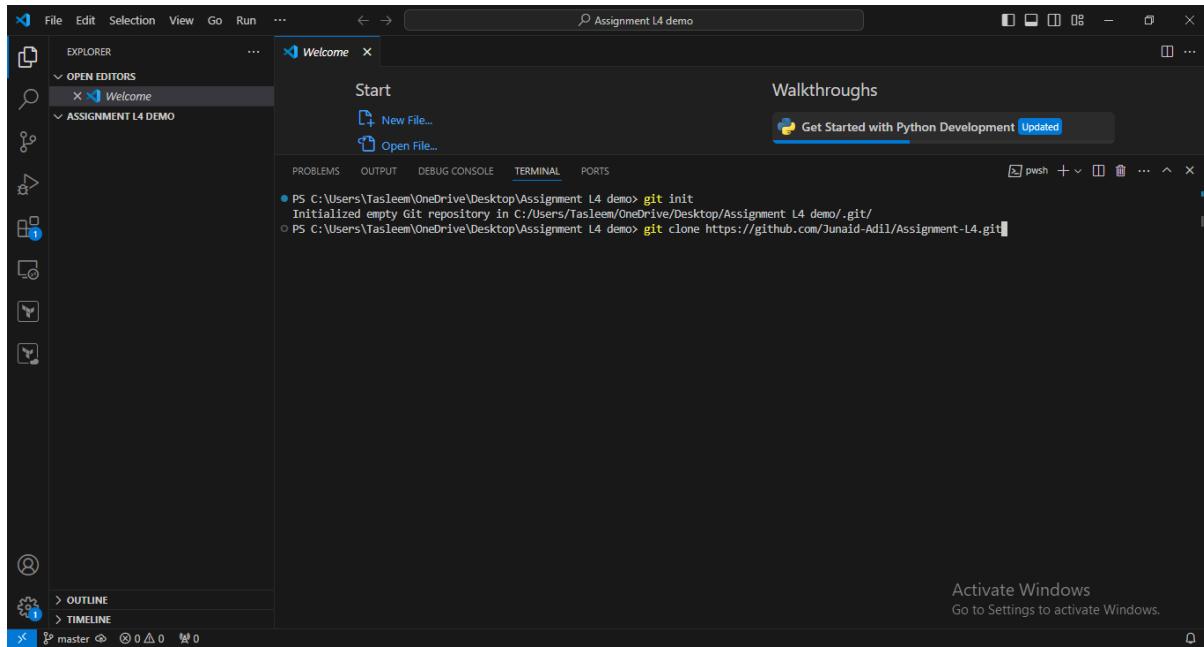




Step 5: Go to Remote Repository → Code → Copy HTTPS URL



Step 6: Go to Visual Studio Code and run the command “git clone <HTTPS URL>”

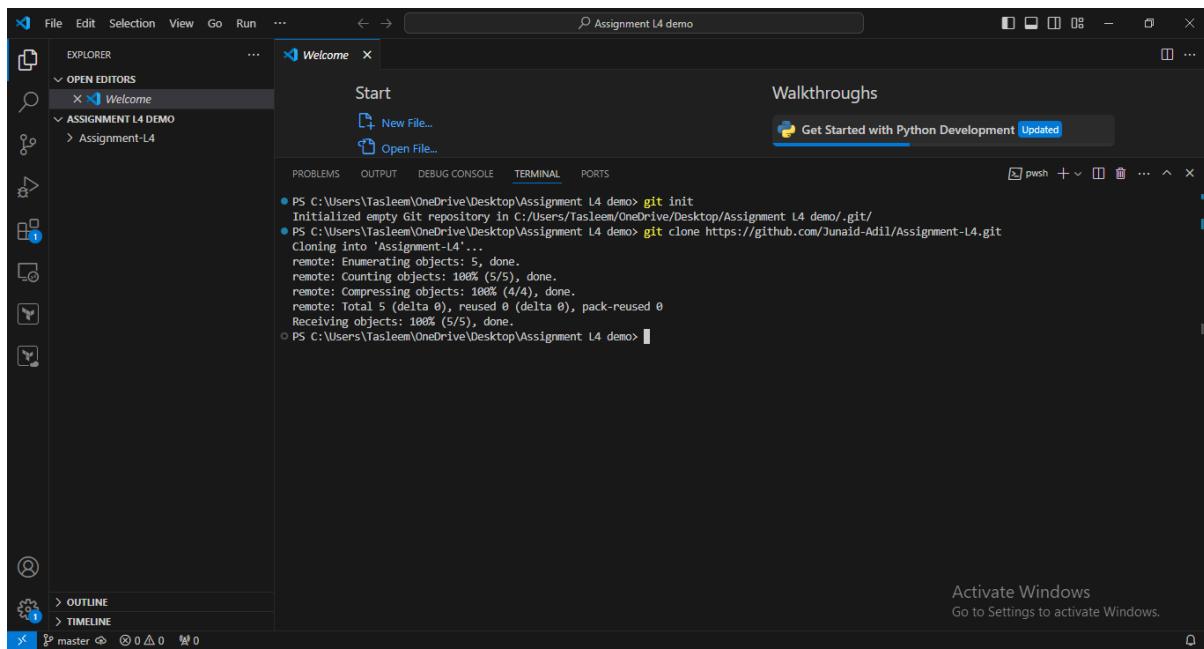


A screenshot of the Visual Studio Code interface. The title bar says "Assignment L4 demo". The left sidebar shows an "EXPLORER" view with a folder named "ASSIGNMENT L4 DEMO" expanded, containing "Welcome". The main area is the "TERMINAL" tab, which displays the following command-line session:

```
PS C:\Users\Tasleem\OneDrive\Desktop\Assignment L4 demo> git init
Initialized empty Git repository in C:/Users/Tasleem/OneDrive/Desktop/Assignment L4 demo/.git/
PS C:\Users\Tasleem\OneDrive\Desktop\Assignment L4 demo> git clone https://github.com/Junaid-Adil/Assignment-L4.git
```

The status bar at the bottom shows "master" and other git-related icons.

Successfully cloned with Local Repository

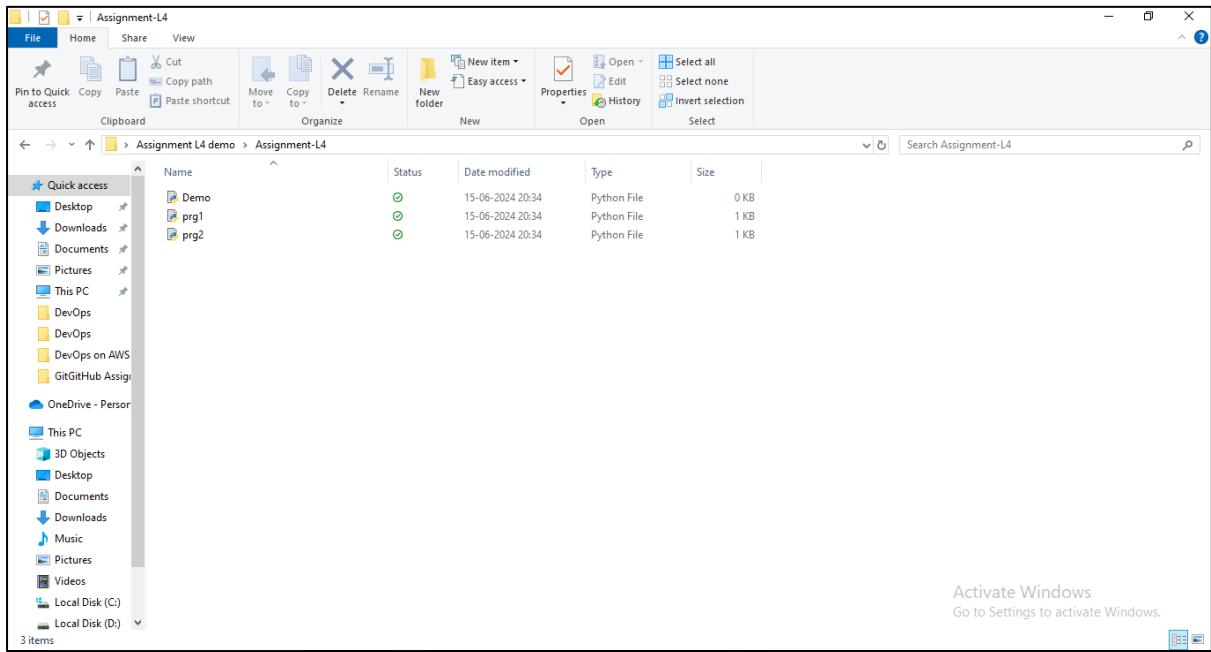


A screenshot of the Visual Studio Code interface, identical to the previous one but showing the cloning process in progress. The terminal output now includes the following additional lines:

```
Cloning into 'Assignment-L4'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

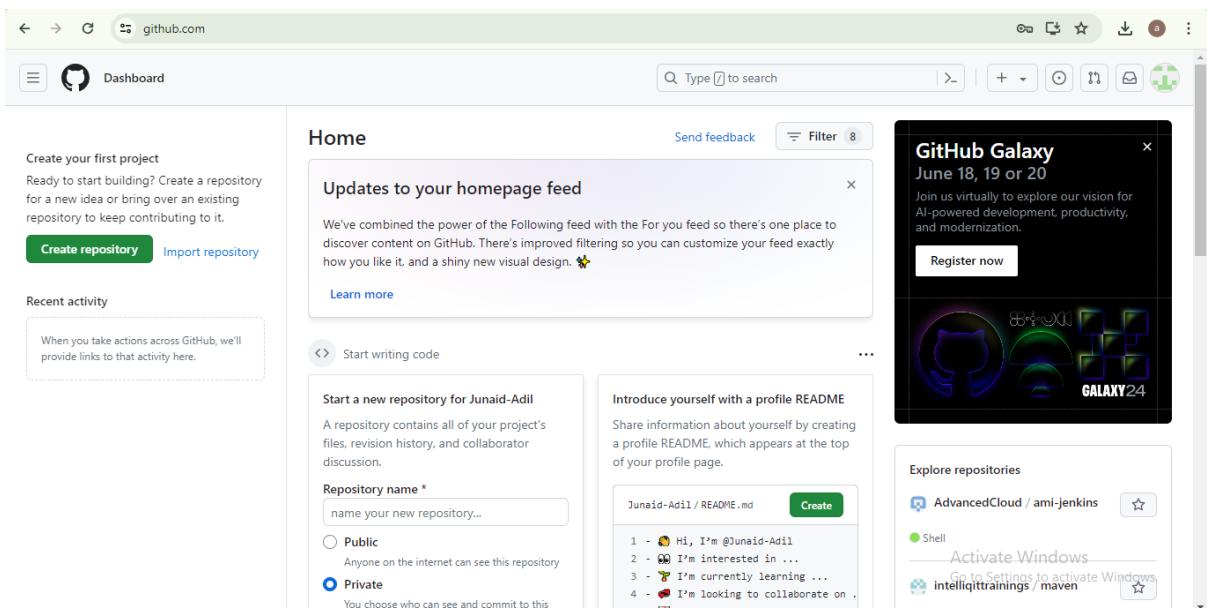
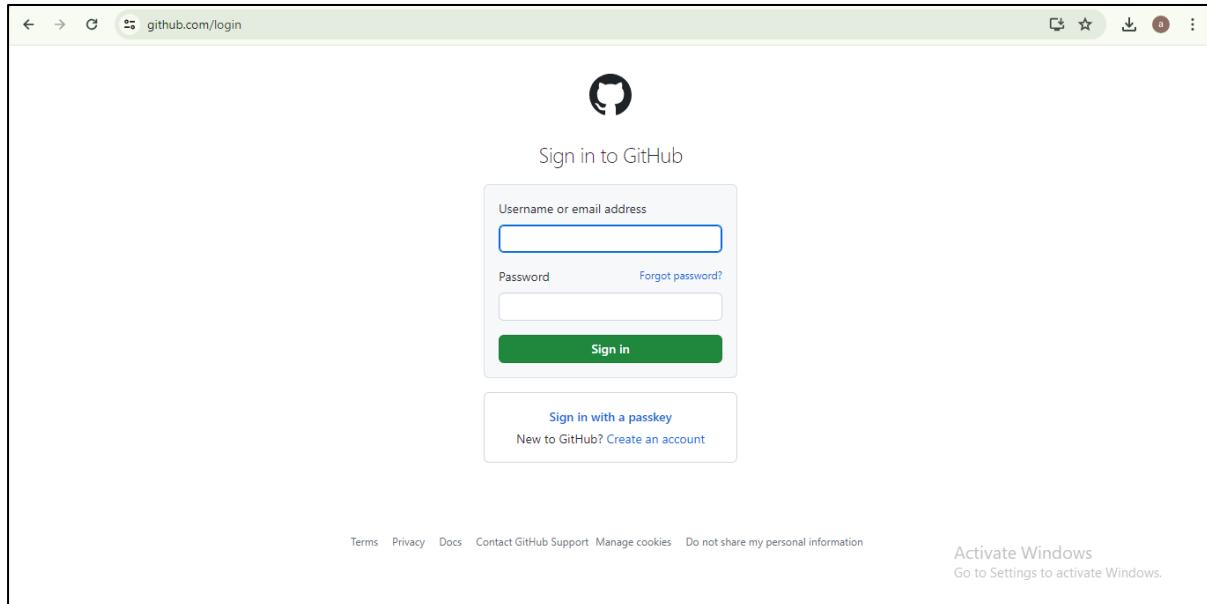
The status bar at the bottom shows "master" and other git-related icons.

Step 7: Go to local Repository which cloned, there we can see the files.

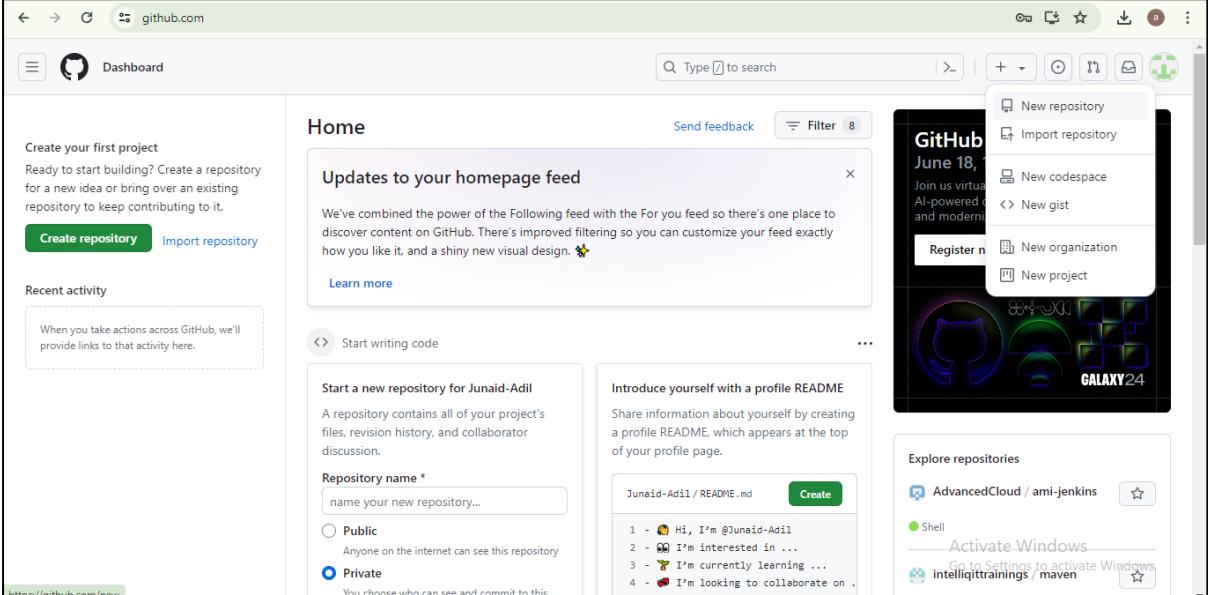


L5 - Push the incremental changes to GitHub Repository through Visual Studio Code IDE

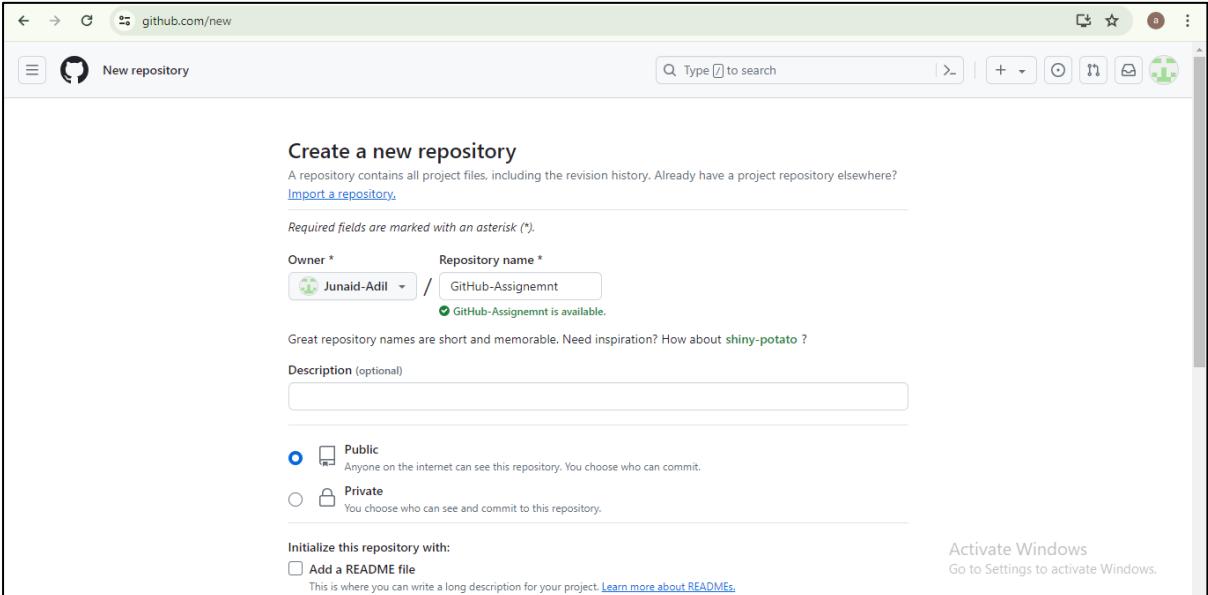
Step 1: Login to GitHub



Step 2: Create a new Repository

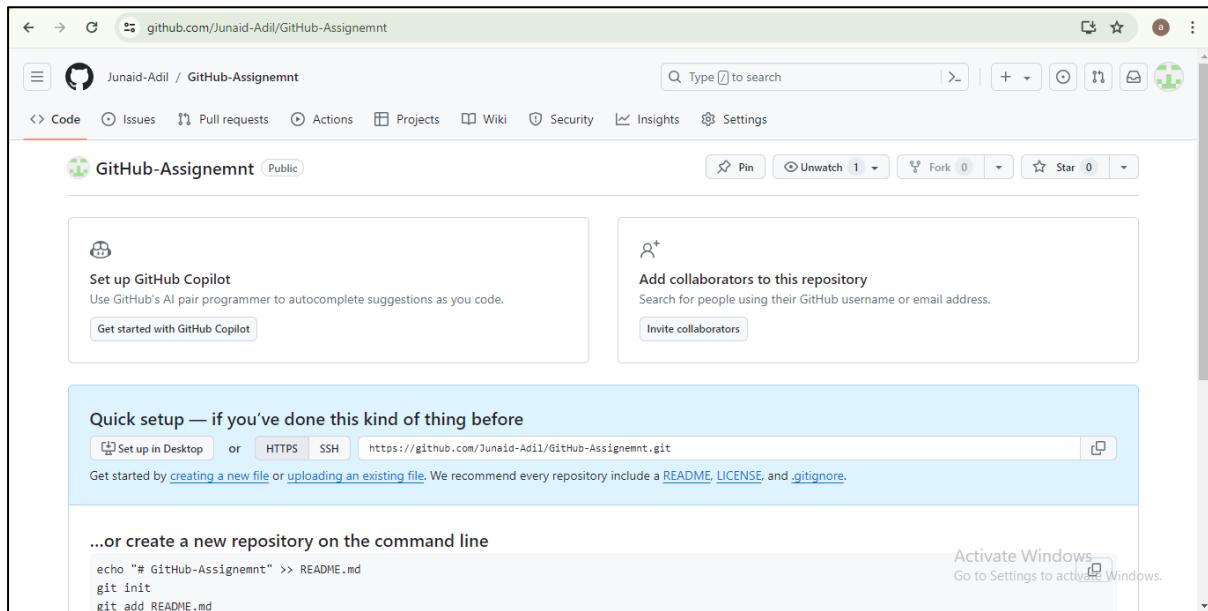


The screenshot shows the GitHub Home page. On the left, there's a sidebar with 'Create your first project' and 'Recent activity'. In the center, there's a 'Updates to your homepage feed' section and a 'Start a new repository for Junaid-Adil' card. On the right, there's a sidebar with GitHub news and a 'New repository' button highlighted in a dropdown menu.

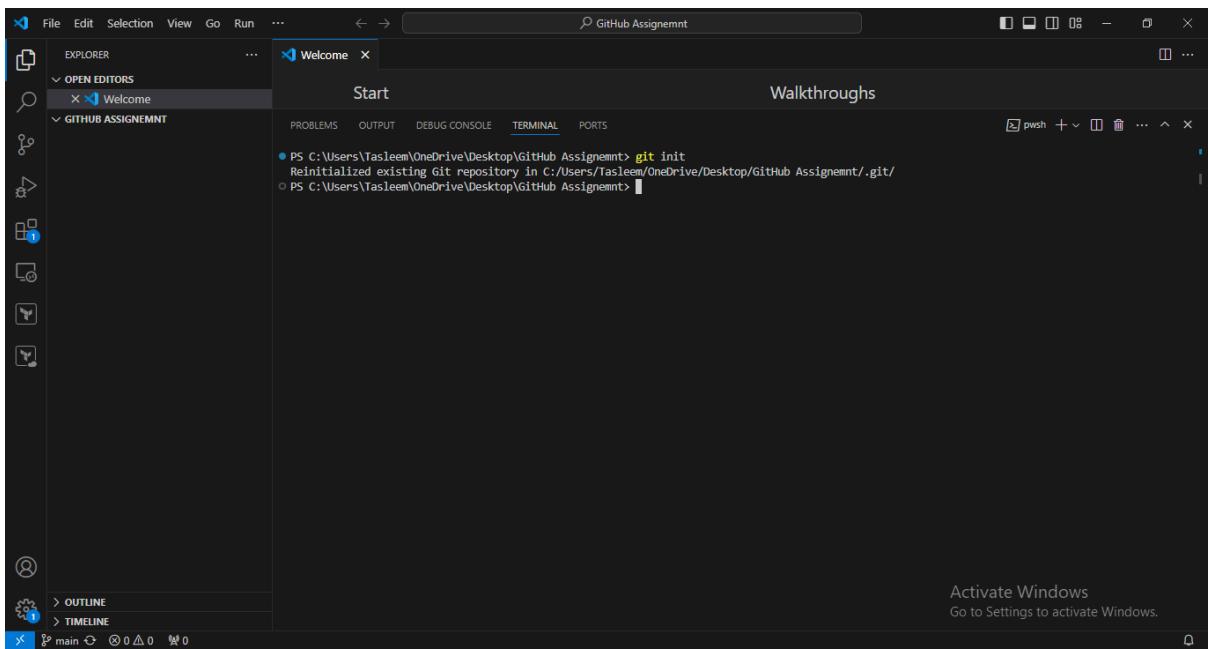


The screenshot shows the 'Create a new repository' form. It has fields for 'Owner' (set to 'Junaid-Adil'), 'Repository name' (set to 'GitHub-Assiginemnt'), and a note that it's available. There's a description field, a radio button for 'Public' (selected), and a checkbox for 'Initialize this repository with: Add a README file'. A note says 'This is where you can write a long description for your project.' On the right, there's an 'Activate Windows' message.

Created a public repository “GitHub-Assignemnt”



Step 3: Open VSC, Open a folder and then initialize Git init.



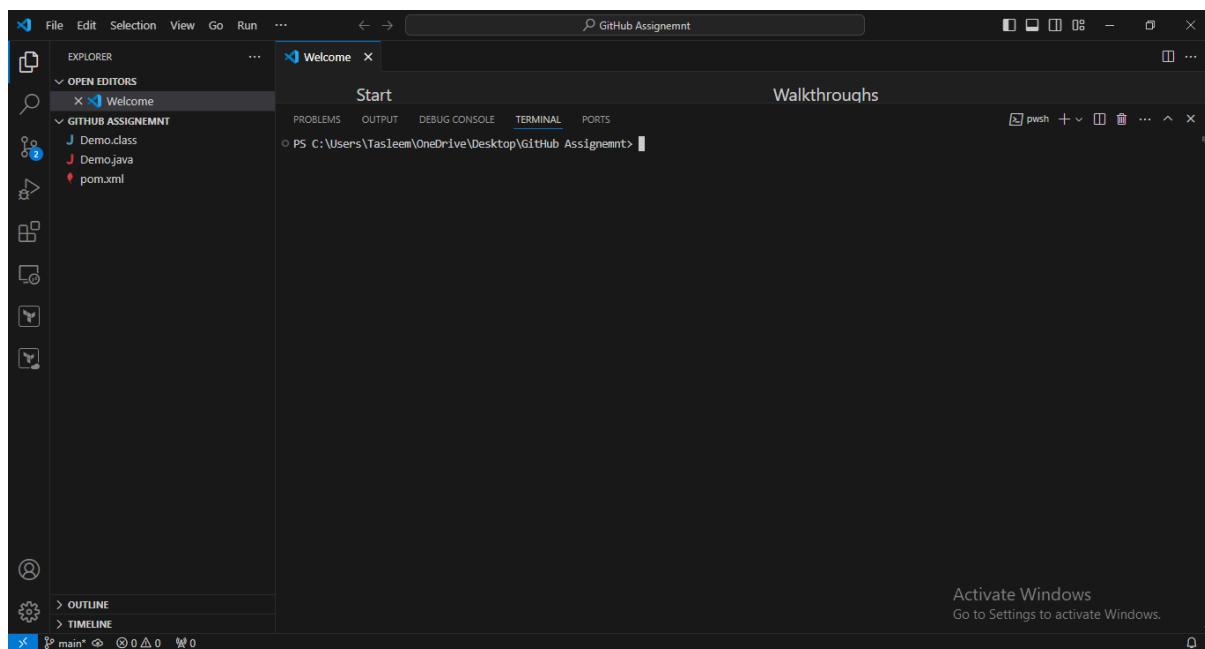
Step 4: Add some files to the local Repository and run the below mentioned commands.

- ⇒ `git branch -M main`
- ⇒ `git remote add origin <Remote Repository path>`

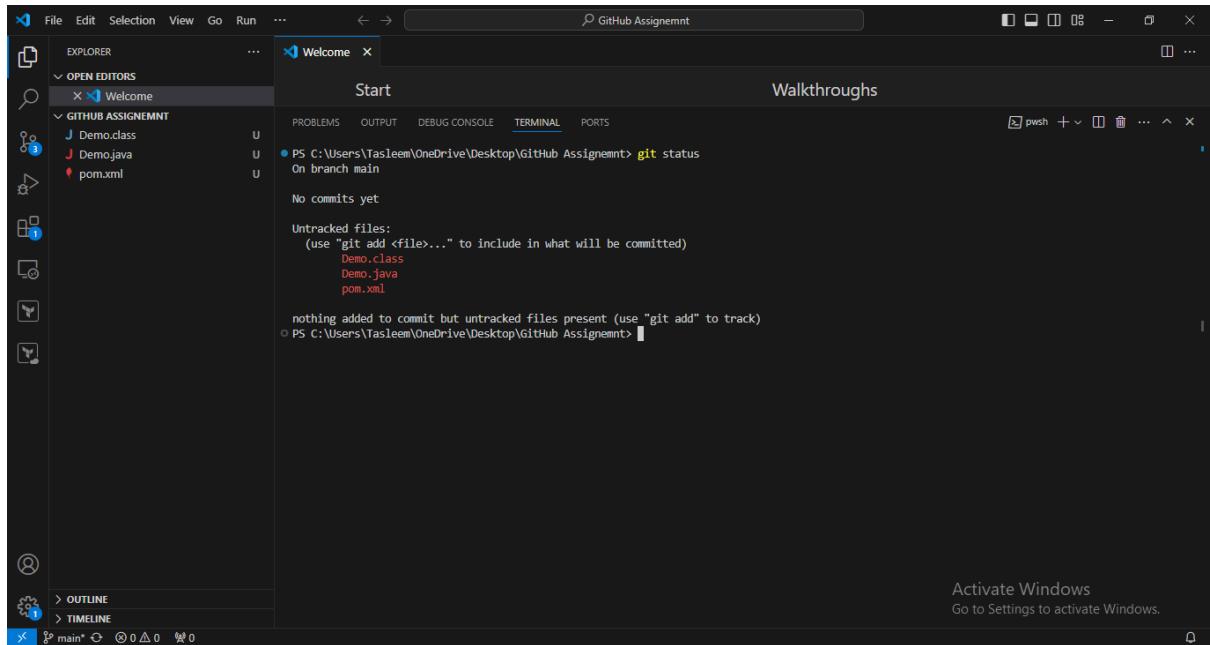
Remote Repository path: <https://github.com/Junaid-Adil/GitHub-Assiginemnt.git>

`git push -u origin <branch>` (after creating some files in Local Repository, we run this command to add those files into Remote Repository)

Added files



Step 5: Execute “git status” to check state of the local repository.



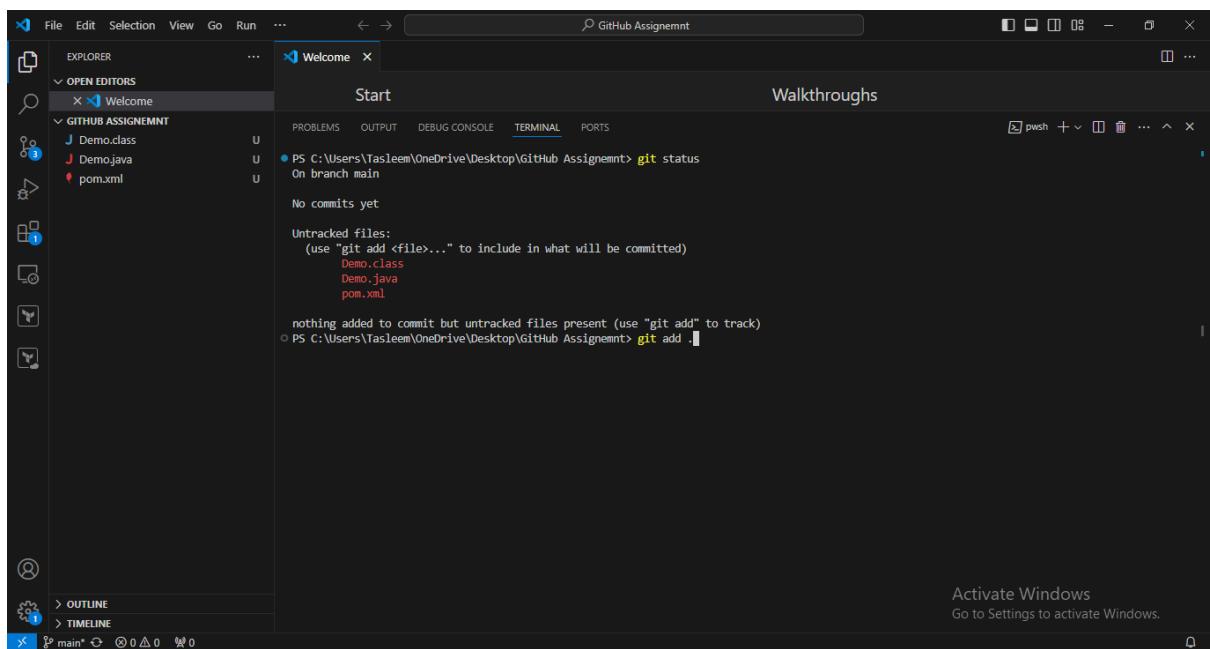
The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the command "git status" being run in a PowerShell environment (PS) on a Windows system. The output shows that there are untracked files: Demo.class, Demo.java, and pom.xml. It also indicates that there are no commits yet.

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Demo.class
    Demo.java
    pom.xml

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment>
```

Step 6: execute “git add .” to add the files to staging area.



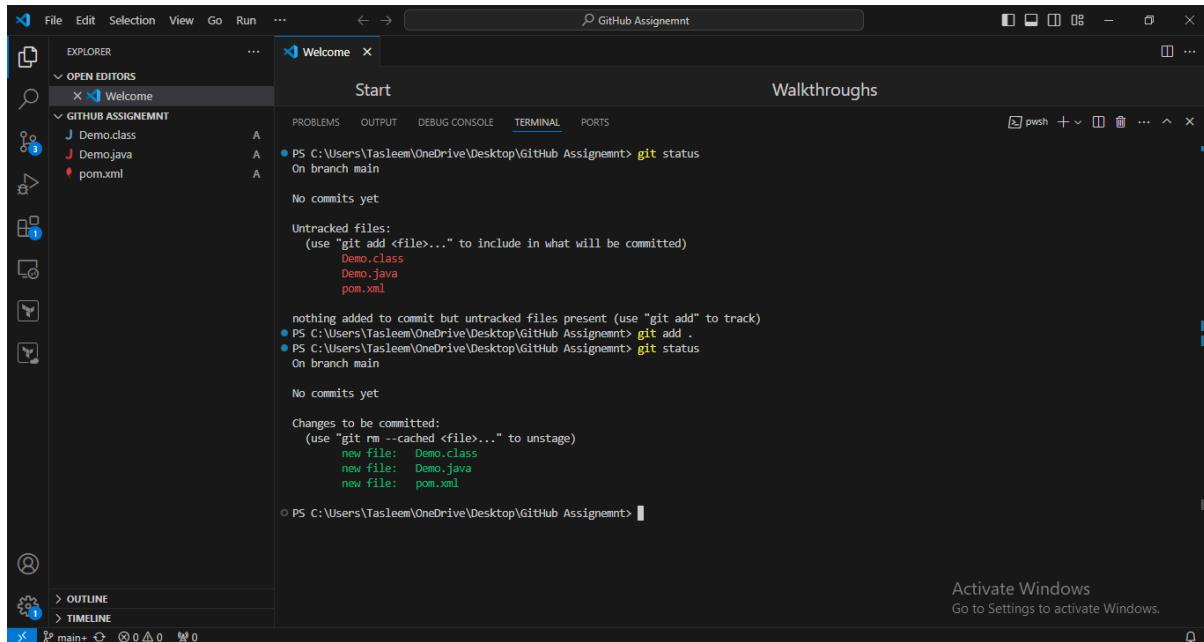
The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the command "git add ." being run in a PowerShell environment (PS) on a Windows system. The output shows that the files Demo.class, Demo.java, and pom.xml have been added to the staging area.

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Demo.class
    Demo.java
    pom.xml

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git add .
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment>
```

Step 7: execute “git commit -m <message>”



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

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
No commits yet

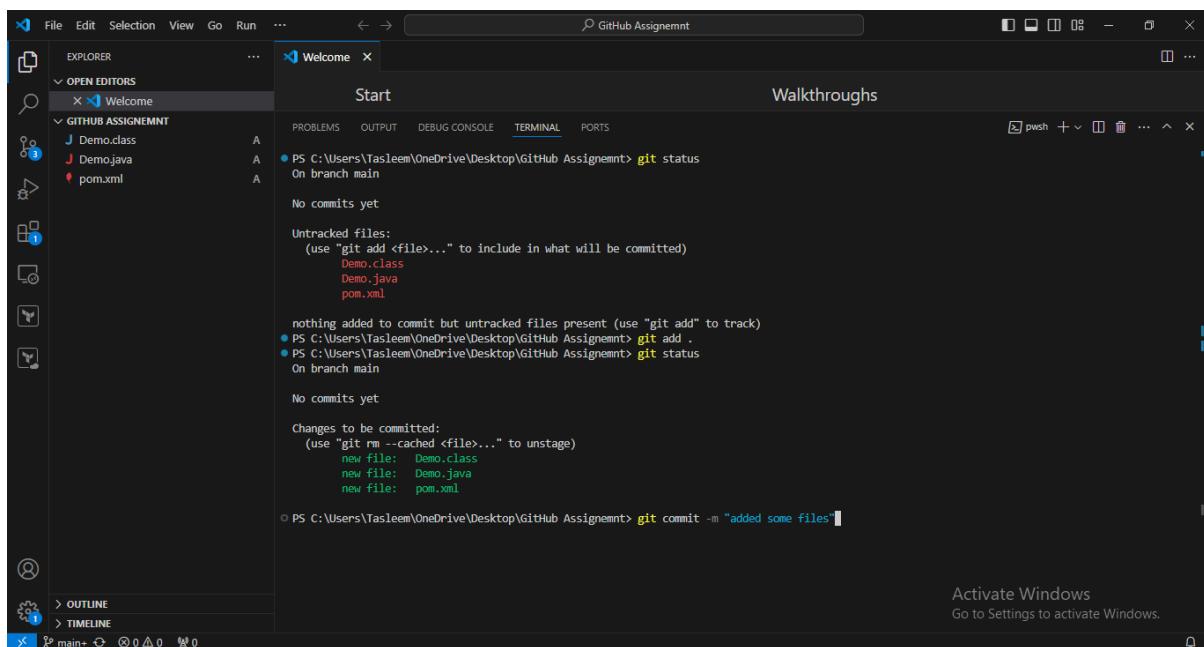
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Demo.class
  Demo.java
  pom.xml

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git add .
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   Demo.class
  new file:   Demo.java
  new file:   pom.xml

PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment>
```

At the bottom of the terminal, the command `git commit -m "added some files"` is typed and highlighted.



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

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Demo.class
  Demo.java
  pom.xml

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git add .
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   Demo.class
  new file:   Demo.java
  new file:   pom.xml

PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git commit -m "added some files"
[master (root-commit) 3a2f3d1] added some files
  3 files changed, 0 insertions(+), 0 deletions(-)
```

The terminal output shows the commit message "added some files" and the resulting commit hash "3a2f3d1".

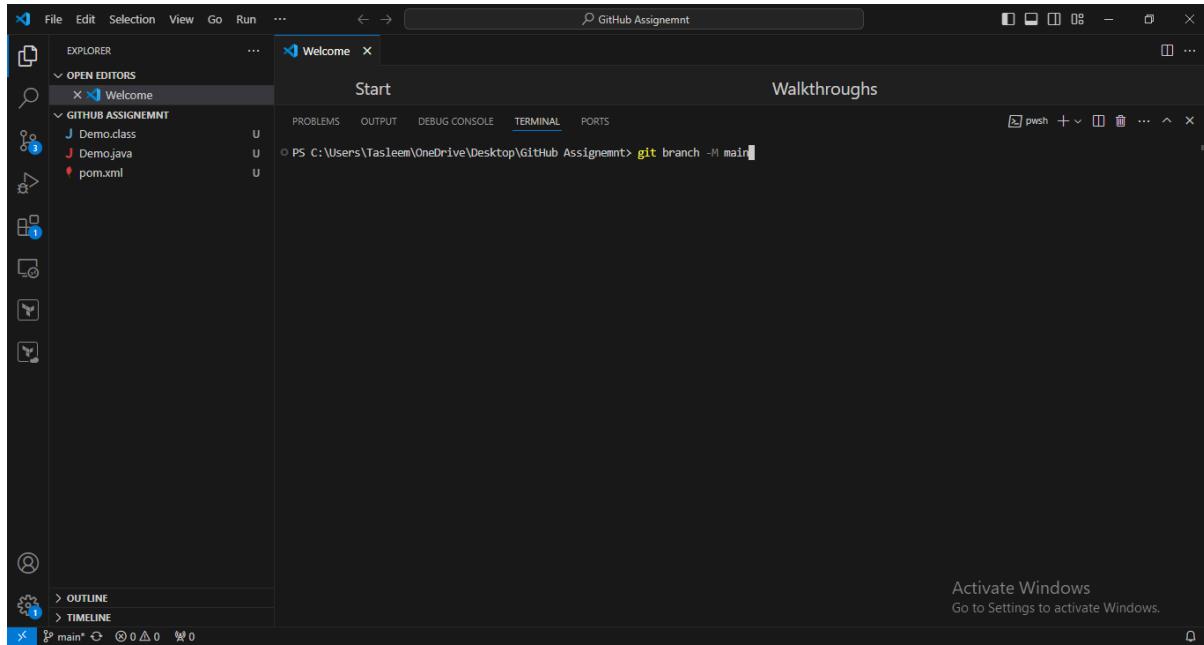
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Title Bar:** GitHub Assignment
- Explorer:** Shows an open folder named "GitHub Assignment" containing files: Demo.class, Demo.java, and pom.xml.
- Terminal:** Displays a PowerShell session (PS) with the following command history:
 - PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> **git status**
 - On branch main
 - No commits yet
 - Untracked files:
 - (use "git add <file>..." to include in what will be committed)
 - Demo.class
 - Demo.java
 - pom.xml
 - nothing added to commit but untracked files present (use "git add" to track)
 - PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> **git add .**
 - PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> **git status**
 - On branch main
 - No commits yet
 - Changes to be committed:
 - (use "git rm --cached <file>..." to unstage)
 - new file: Demo.class
 - new file: Demo.java
 - new file: pom.xml
 - PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> **git commit -m "added some files"**
 - [main (root-commit) 44830ff] added some files
 - 3 files changed, 26 insertions(+)
 - create mode 100644 Demo.class
 - create mode 100644 Demo.java
 - create mode 100644 pom.xml
- Bottom Status Bar:** Activate Windows, Go to Settings to activate Windows.

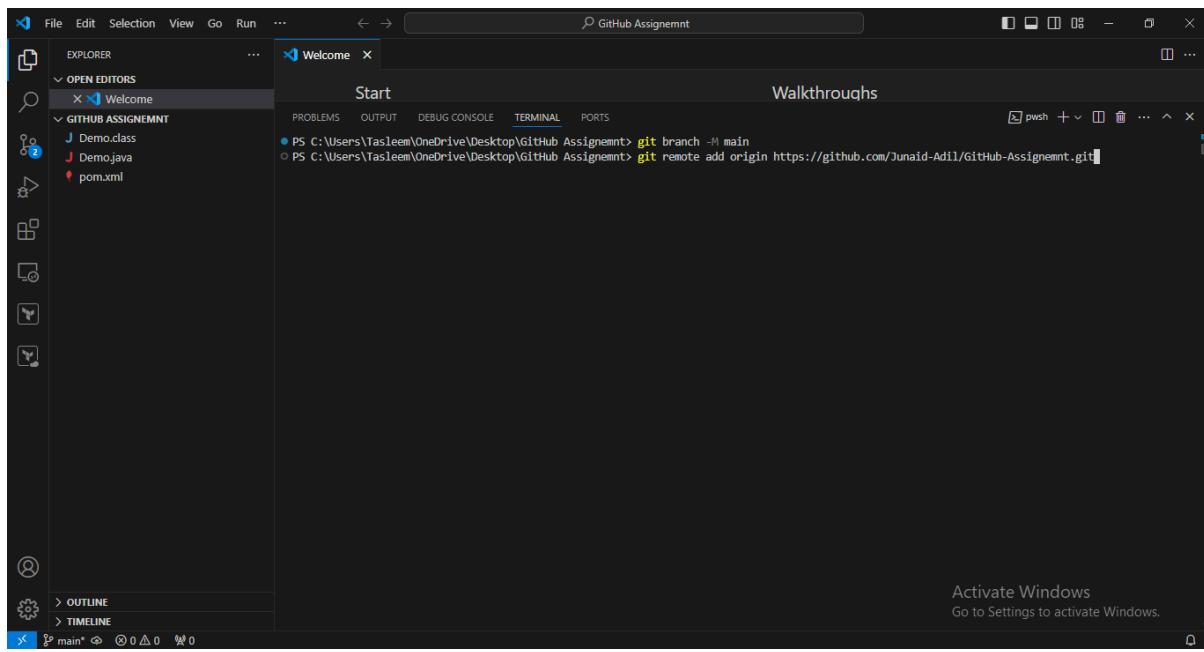
The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Title Bar:** GitHub Assignment
- Explorer:** Shows an open folder named "GitHub Assignment" containing files: Demo.class, Demo.java, and pom.xml.
- Terminal:** Displays a PowerShell session (PS) with the following command history:
 - PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> **git status**
 - On branch main
 - No commits yet
 - Untracked files:
 - (use "git add <file>..." to include in what will be committed)
 - Demo.class
 - Demo.java
 - pom.xml
 - nothing added to commit but untracked files present (use "git add" to track)
 - PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> **git add .**
 - PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> **git status**
 - On branch main
 - No commits yet
 - Changes to be committed:
 - (use "git rm --cached <file>..." to unstage)
 - new file: Demo.class
 - new file: Demo.java
 - new file: pom.xml
 - PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> **git commit -m "added some files"**
 - [main (root-commit) 44830ff] added some files
 - 3 files changed, 26 insertions(+)
 - create mode 100644 Demo.class
 - create mode 100644 Demo.java
 - create mode 100644 pom.xml
- Bottom Status Bar:** Activate Windows, Go to Settings to activate Windows.

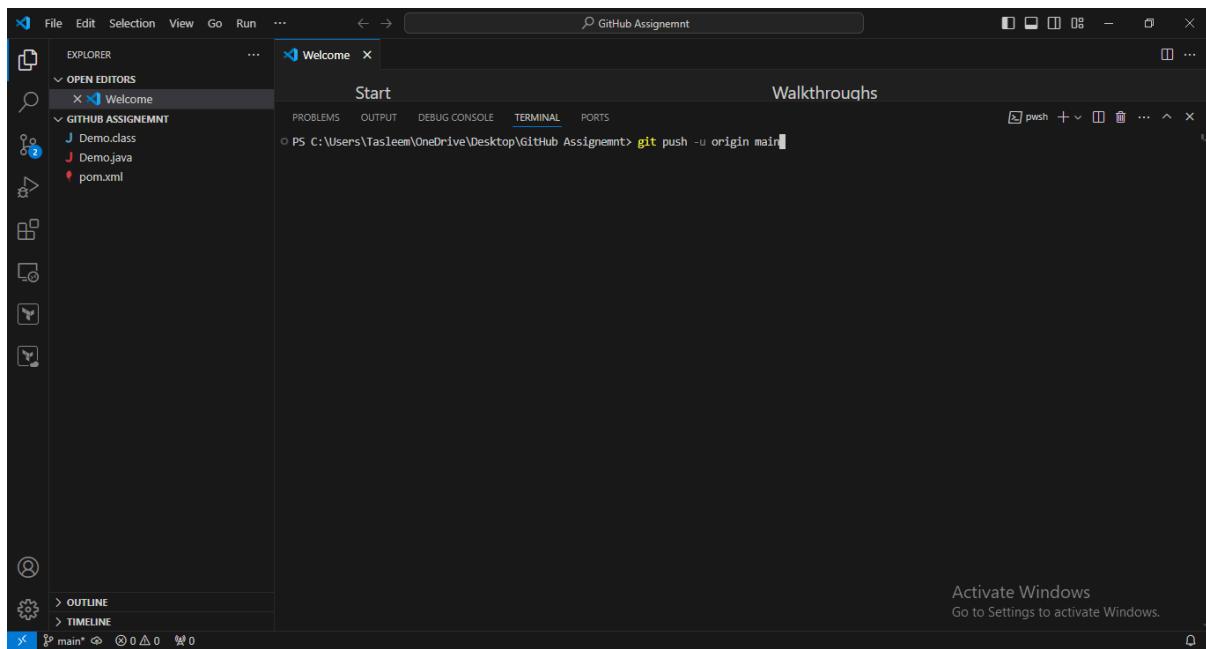
Step 8: Now execute the commands to link the Local and Remote Repository.



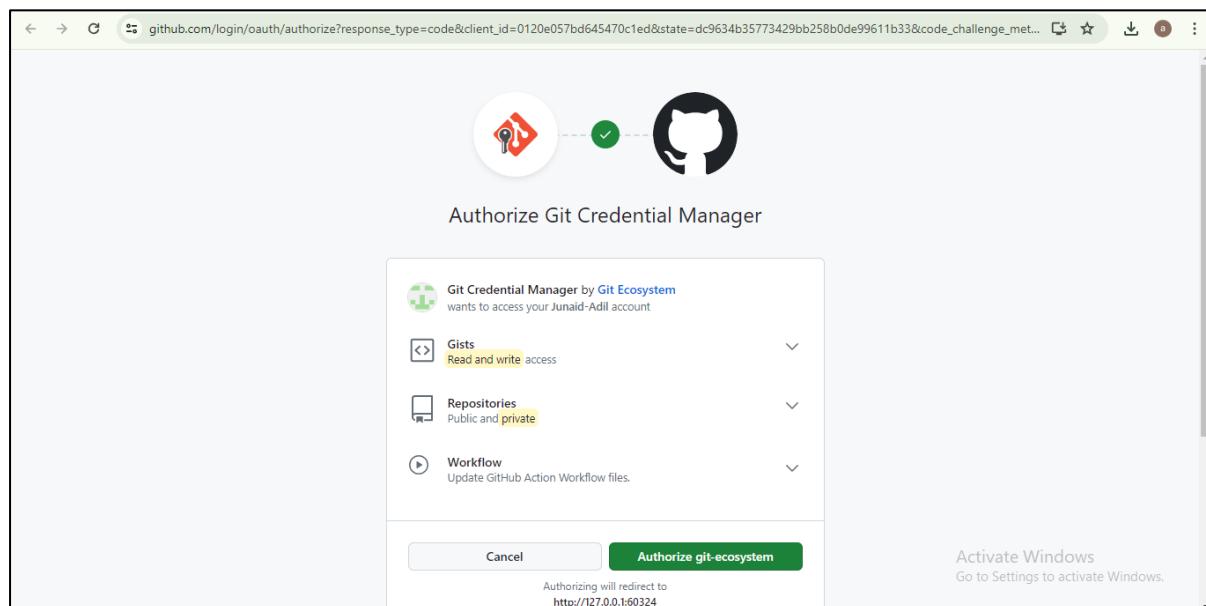
The screenshot shows the Visual Studio Code interface with a dark theme. The left sidebar has an 'EXPLORER' view showing files like 'Demo.class', 'Demo.java', and 'pom.xml'. The main area is the 'TERMINAL' tab, which displays a PowerShell session with the command 'git branch -M main' entered. The status bar at the bottom shows 'main*' and other file-related icons.

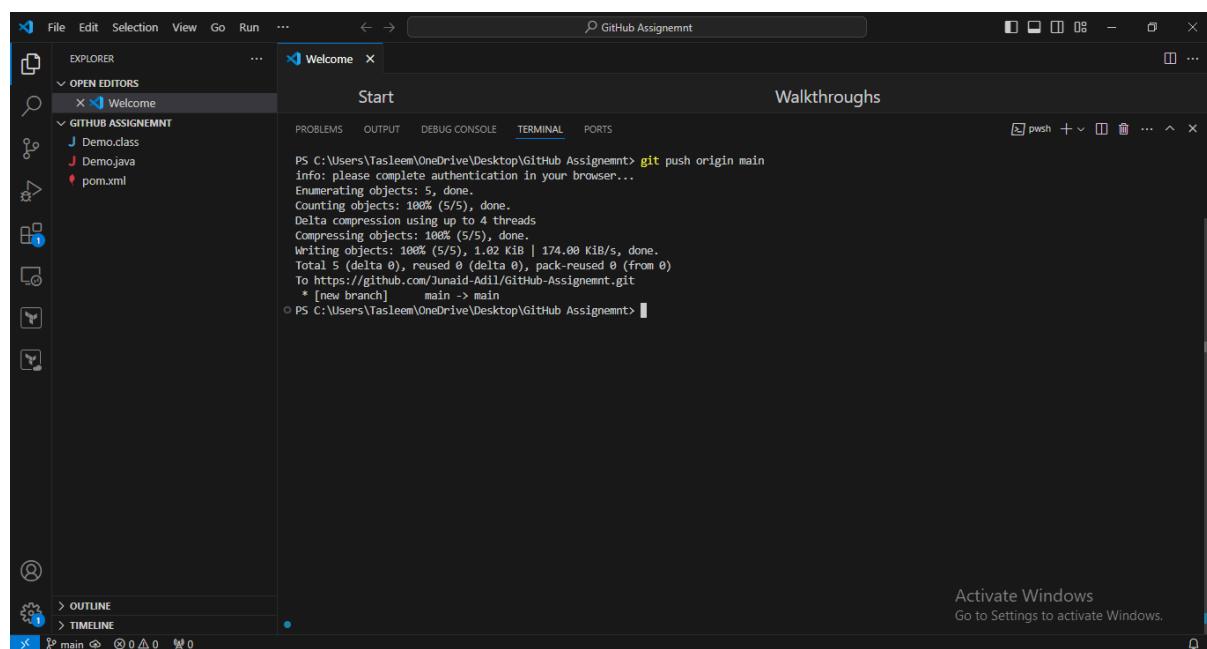
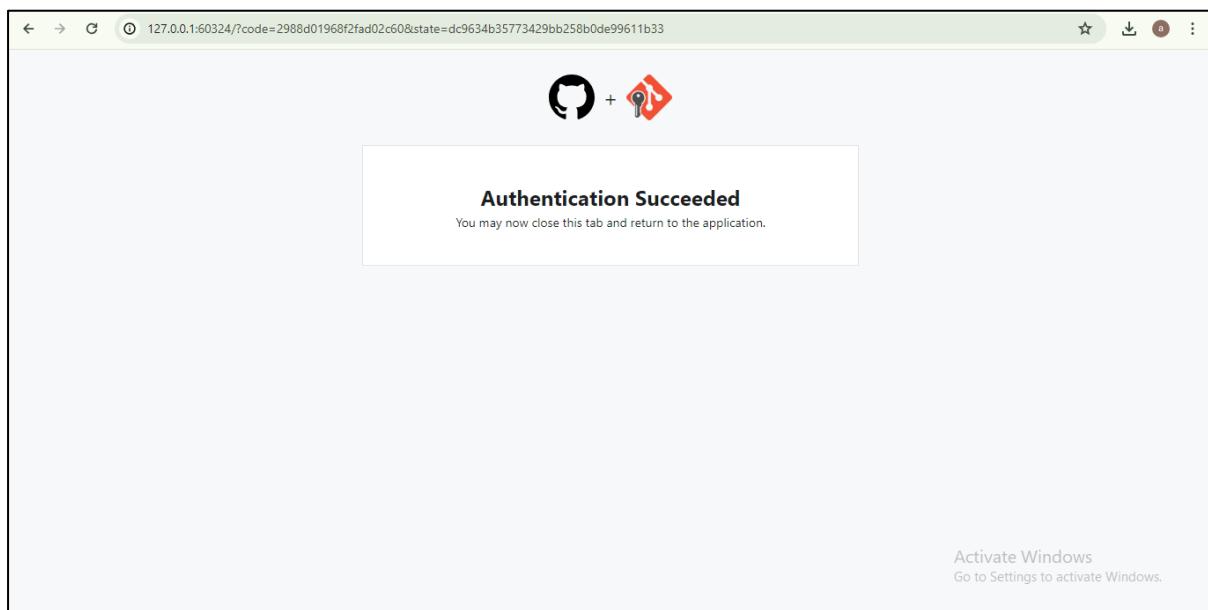


This screenshot is identical to the previous one, showing the same VS Code interface and terminal session. The difference is in the status bar at the bottom, which now shows 'main*' and other file-related icons, indicating changes have been made to the local repository.



Step 9: Page gets pop out to Authorize the permission





Step 10: Open “GitHub-Assigment” Repository in Github

This screenshot shows the GitHub profile page for the user Junaid-Adil. The top navigation bar includes links for Overview, Repositories (2), Projects, Packages, and Stars. Below the navigation is a circular profile picture and a search bar with placeholder text "Type / to search". A "Popular repositories" section displays two repositories: "Demo" (Public) and "GitHub-Assigment" (Public). The "GitHub-Assigment" repository is described as Java and was updated 1 minute ago. A "Contribution activity" chart shows contributions from June 2024, with a notable spike in April. A message at the bottom right encourages activating Windows.

This screenshot shows the GitHub repository list page for the user Junaid-Adil, specifically the "Repositories" tab. The top navigation bar is identical to the previous screenshot. The main content area lists the user's repositories: "GitHub-Assigment" (Public, Java, updated 1 minute ago) and "Demo" (Public, updated 42 minutes ago). A search bar at the top allows users to "Find a repository..." and includes filters for Type, Language, Sort, and a "New" button. A message at the bottom right encourages activating Windows.

The screenshot shows a GitHub repository page for 'GitHub-Assignemnt'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation is a header for 'GitHub-Assignemnt' with a green icon, showing it's a public repository. The main content area displays a commit history from 'junaid786-7' adding files like 'Demo.class', 'Demo.java', and 'pom.xml'. A 'Code' dropdown menu is open, showing options like 'main', 'Branch', 'Tags', 'Go to file', 'Add file', and 'Code'. To the right, there's an 'About' section with a note 'No description, website, or topics provided.', an 'Activity' section showing 0 stars, 1 watching, and 0 forks, and a 'Releases' section indicating 'No releases published' with a link to 'Create a new release'. Below the commit history, there's a 'README' section with a placeholder 'Add a README' and a note to help people understand the project.

Local Repository has been merged with Remote Repository.

Step 11: Now update the version in pom.xml file from 1.0 to 2.0

The screenshot shows a terminal window in VS Code with the title 'pom.xml'. It displays the XML content of the 'pom.xml' file, which includes details about the project's version (1.0-SNAPSHOT), group ID ('02-Maven-WebApp'), artifact ID ('Maven Webapp'), and dependencies (specifically junit). Below the code editor, the terminal shows the command 'git status' being run, with the output indicating 'nothing to commit, working tree clean'. The bottom right corner of the terminal window has a note 'Activate Windows' with a link to 'Go to Settings to activate Windows.'

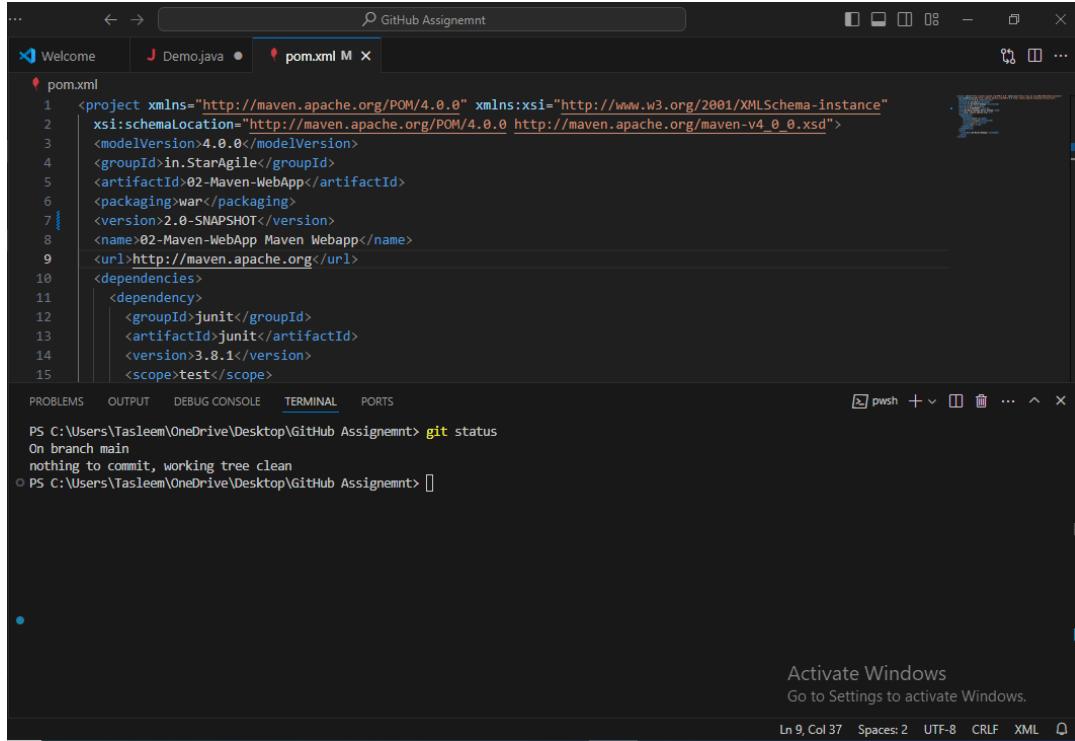
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>in.StarAgile</groupId>
  <artifactId>02-Maven-WebApp</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>02-Maven-WebApp Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assigemnt> git status
On branch main
nothing to commit, working tree clean
○ PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assigemnt>

Activate Windows
Go to Settings to activate Windows.
Ln 7, Col 13 Spaces: 2 UTRF-8 CRLF XML

Updated to 2.0



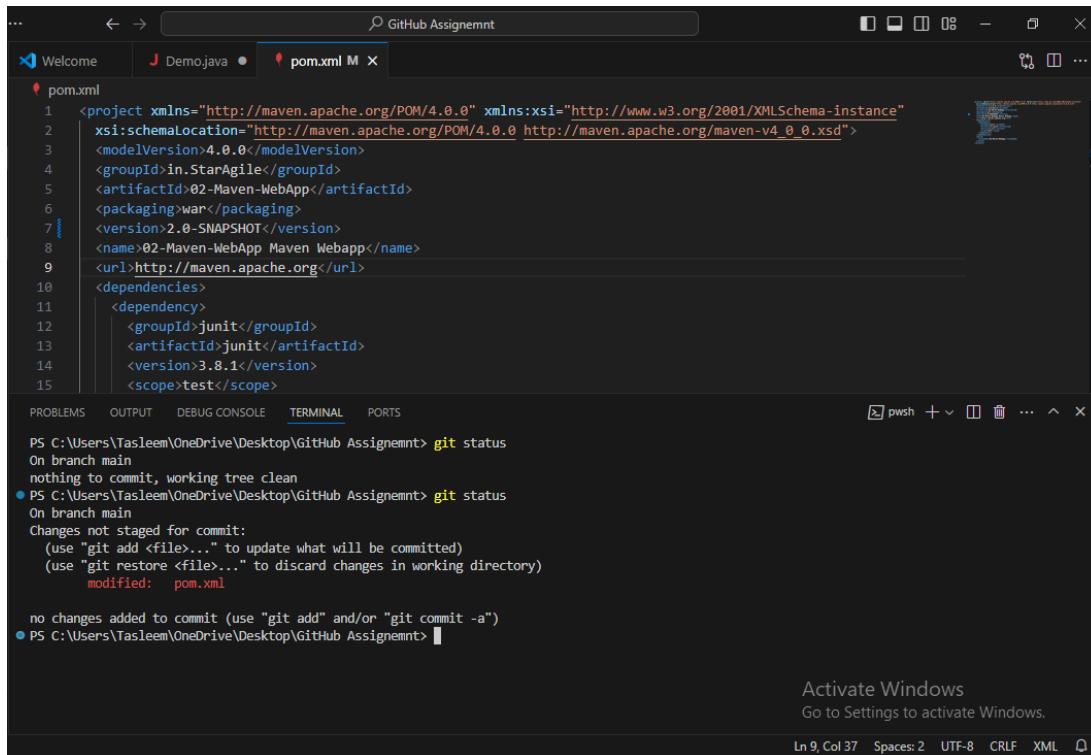
The screenshot shows the VS Code interface with the title bar "GitHub Assignemnt". The left sidebar has "Welcome" and "Demo.java" listed. The main editor tab is "pom.xml". The code in the editor is:

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>in.StarAgile</groupId>
  <artifactId>02-Maven-WebApp</artifactId>
  <packaging>war</packaging>
  <version>2.0-SNAPSHOT</version>
  <name>02-Maven-WebApp Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    
```

The terminal tab shows the command "git status" output:

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignemnt> git status
On branch main
nothing to commit, working tree clean
○ PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignemnt>
```

The status bar at the bottom right shows "Ln 9, Col 37 Spaces: 2 UTF-8 CRLF XML".



The screenshot shows the VS Code interface with the title bar "GitHub Assignemnt". The left sidebar has "Welcome" and "Demo.java" listed. The main editor tab is "pom.xml". The code in the editor is identical to the previous screenshot.

The terminal tab shows the command "git status" output, which includes a new line indicating changes were staged for commit:

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignemnt> git status
On branch main
nothing to commit, working tree clean
● PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignemnt> git status
On branch main
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:   pom.xml

no changes added to commit (use "git add" and/or "git commit -a")
● PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignemnt>
```

The status bar at the bottom right shows "Ln 9, Col 37 Spaces: 2 UTF-8 CRLF XML".

We can see as the file is updated, we need to add the file to staging area and commit the changes.

Step 12: Added to staging area

The screenshot shows the VS Code interface. The top bar has tabs for 'Welcome', 'Demojava', 'pom.xml', and 'M'. The main area displays the content of the 'pom.xml' file. Below the editor is a terminal window showing the following output:

```
nothing to commit, working tree clean
● PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
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:   pom.xml

no changes added to commit (use "git add" and/or "git commit -a")
● PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git add .
● PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   pom.xml

○ PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment>
```

The terminal also shows a message from Windows: 'Activate Windows Go to Settings to activate Windows.'

Step 13: Commit the changes

The screenshot shows the VS Code interface. The top bar has tabs for 'Welcome', 'Demojava', 'pom.xml', and 'M'. The main area displays the content of the 'pom.xml' file. Below the editor is a terminal window showing the following output:

```
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:   pom.xml

no changes added to commit (use "git add" and/or "git commit -a")
● PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git add .
● PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   pom.xml

● PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git commit -m "updated file pom.xml"
[main 1669182] updated file pom.xml
  1 file changed, 1 insertion(+), 1 deletion(-)
○ PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment>
```

The terminal also shows a message from Windows: 'Activate Windows Go to Settings to activate Windows.'

The screenshot shows the VS Code interface with the terminal tab selected. The terminal output is as follows:

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git add .
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   pom.xml

PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git commit -m "updated file pom.xml"
[main 1669182] updated file pom.xml
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
nothing to commit, working tree clean
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment>
```

Step 14: Now run the command “git push origin main” to push the updated file to remote repository

The screenshot shows the VS Code interface with the terminal tab selected. The terminal output is as follows:

```
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git add .
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   pom.xml

PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git commit -m "updated file pom.xml"
[main 1669182] updated file pom.xml
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
nothing to commit, working tree clean
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git push origin main
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files: pom.xml, Demo.java, and Demo.class.
- Editor:** The main editor window displays the content of pom.xml.
- Terminal:** The terminal window shows the output of a git commit command and a git status command.
- Status Bar:** Shows "1 unsaved" changes.

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>in.StarAgile</groupId>
  <artifactId>02-Maven-WebApp</artifactId>
  <packaging>war</packaging>
  <version>2.0-SNAPSHOT</version>
  <name>02-Maven-WebApp Maven Webapp</name>
  <url>http://maven.apache.org</url>
</project>
  
```

```

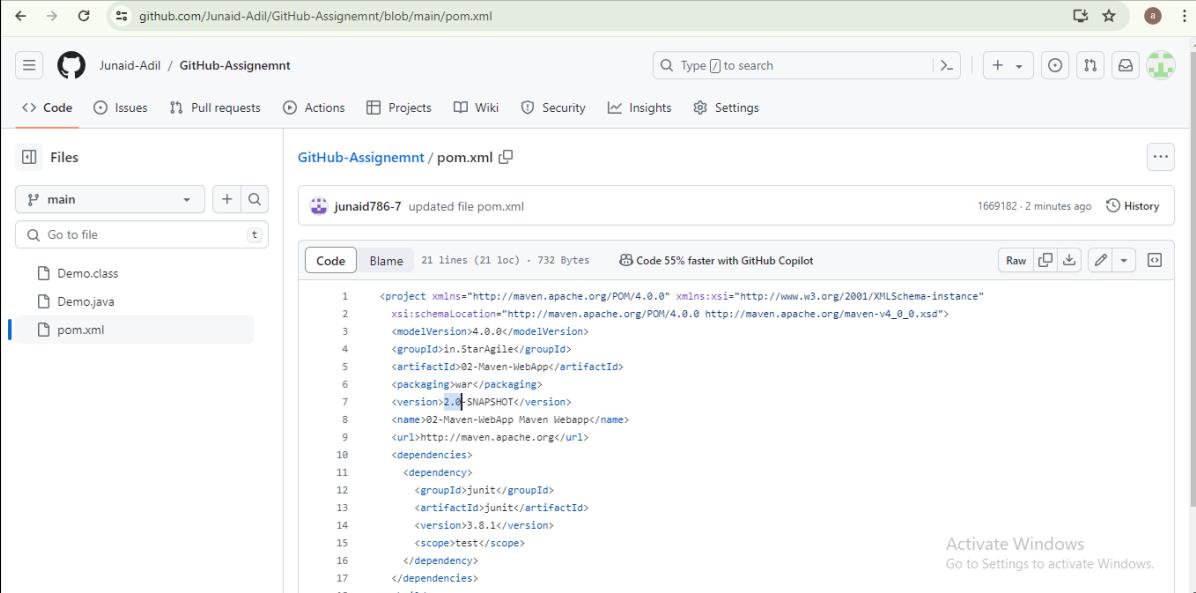
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git commit -m "updated file pom.xml"
[main 1669182] updated file pom.xml
  1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git status
On branch main
nothing to commit, working tree clean
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 284 bytes | 94.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Junaid-Adil/Github-Assigment.git
  44830F5..1669182 main -> main
PS C:\Users\Tasleem\OneDrive\Desktop\GitHub Assignment> 
  
```

When we refresh the page we can see the details/Time when the file got updated

The screenshot shows a GitHub repository page with the following details:

- Repository Name:** GitHub-Assigment
- Commit History:**
 - Junaid786-7 updated file pom.xml (1669182, 2 minutes ago)
 - Demo.class added some files (19 minutes ago)
 - Demo.java added some files (19 minutes ago)
 - pom.xml updated file pom.xml (2 minutes ago)
- Right Sidebar:**
 - About:** No description, website, or topics provided.
 - Activity:** 0 stars, 1 watching, 0 forks
 - Releases:** No releases published. Create a new release.
 - Packages:** No packages published. Publish your first package.

Step 15: Open the file ad check if the version has been updated.



The screenshot shows a GitHub repository page for 'Junaid-Adil / GitHub-Assignemnt'. The 'Code' tab is selected, displaying the file 'GitHub-Assignemnt / pom.xml'. A specific commit, 'junaid786-7', is highlighted, showing an update to the file. The commit message is 'updated file pom.xml'. The code editor shows the XML content of the pom.xml file, with line numbers 1 through 18 visible. The code includes details like the project's XML namespace, schema location, model version, group ID, artifact ID, packaging, version (2.0-SNAPSHOT), name (02-Maven-WebApp Maven Webapp), URL (http://maven.apache.org/), dependencies (JUnit dependency with version 3.8.1 and scope test), and build information. A note at the bottom right of the code editor area says 'Activate Windows Go to Settings to activate Windows.'

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>in.StarAgile</groupId>
  <artifactId>02-Maven-WebApp</artifactId>
  <packaging>war</packaging>
  <version>2.0-SNAPSHOT</version>
  <name>02-Maven-WebApp Maven Webapp</name>
  <url>http://maven.apache.org/</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
```

We can see the Incremental changes has been done through Visual Studio Code IDE.