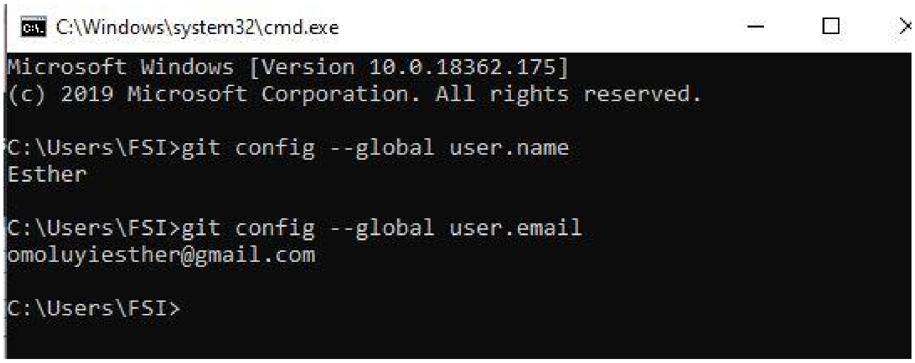
GIT-SCM-VCS-L1 ASSIGNMENT

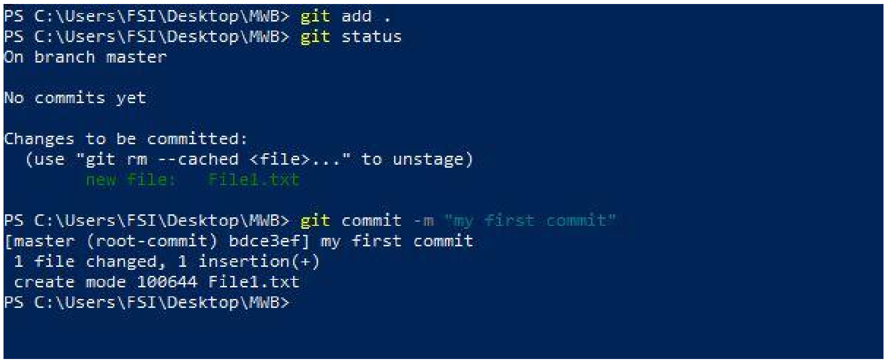
1. **Set the global configuration file with your username and email. List all the properties which you just set.**



```
Microsoft Windows [Version 10.0.18362.175]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\FSI>git config --global user.name
Esther

C:\Users\FSI>git config --global user.email
omoluyiesther@gmail.com

C:\Users\FSI>
```

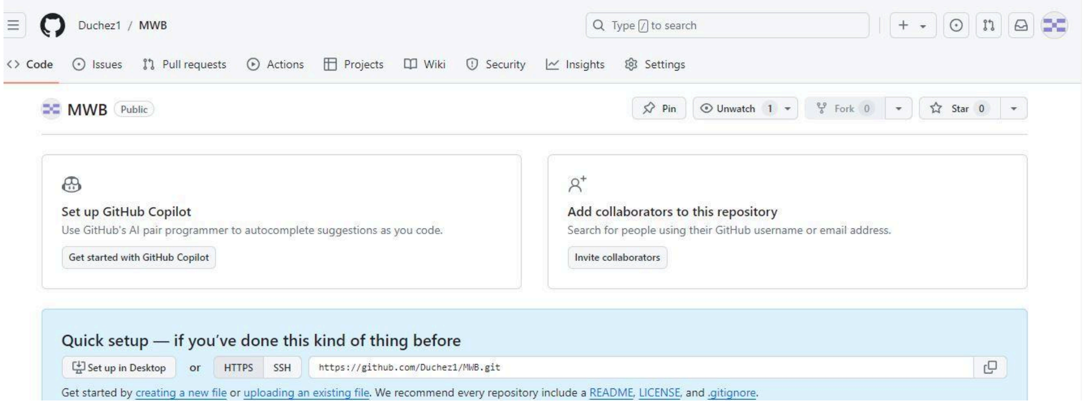
   a. git config --global --list
      i. user.name
      ii. user.email

2. **Create a fresh Git project. Add a file into the project. Commit the changes to the local repository.**



```
PS C:\Users\FSI\Desktop\MWB> git add .
PS C:\Users\FSI\Desktop\MWB> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   File1.txt

PS C:\Users\FSI\Desktop\MWB> git commit -m "my first commit"
[master (root-commit) bdce3ef] my first commit
 1 file changed, 1 insertion(+)
 create mode 100644 File1.txt
PS C:\Users\FSI\Desktop\MWB>
```

3. **Create a GitHub/Gitlab account (or use the account if already registered). Clone a project from the remote repository to your local repository**



Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Add collaborators to this repository

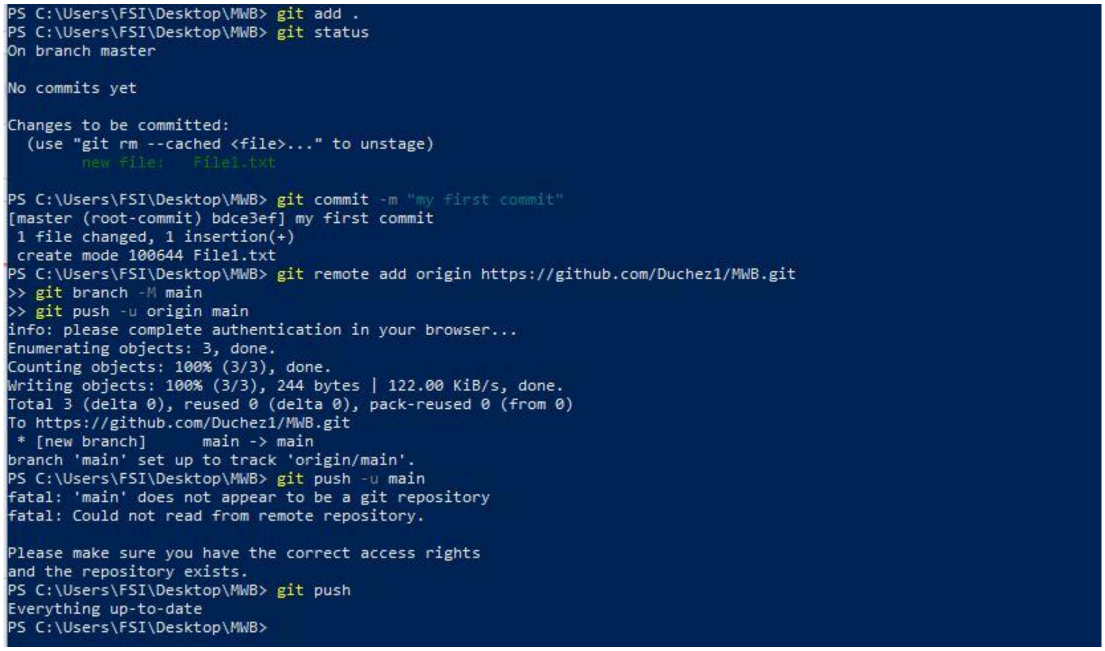
Search for people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

https://github.com/Duchez1/MWB.git

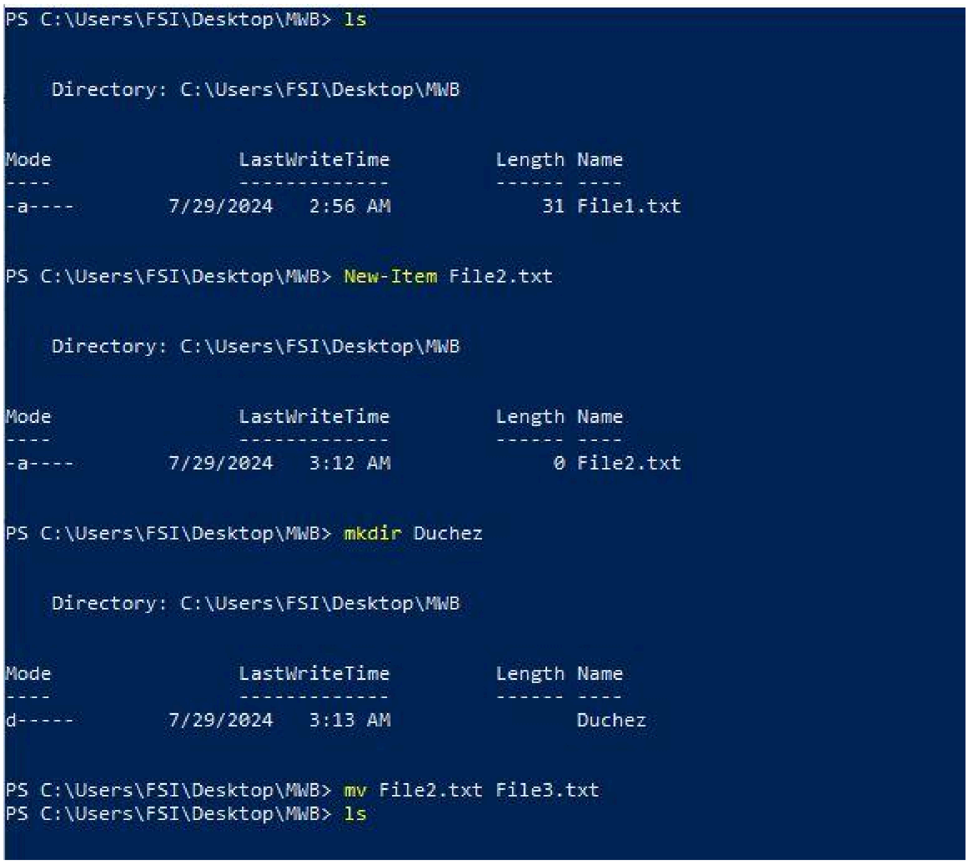
Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

4. **. Push the project created in assignment 2 to the remote repository.**

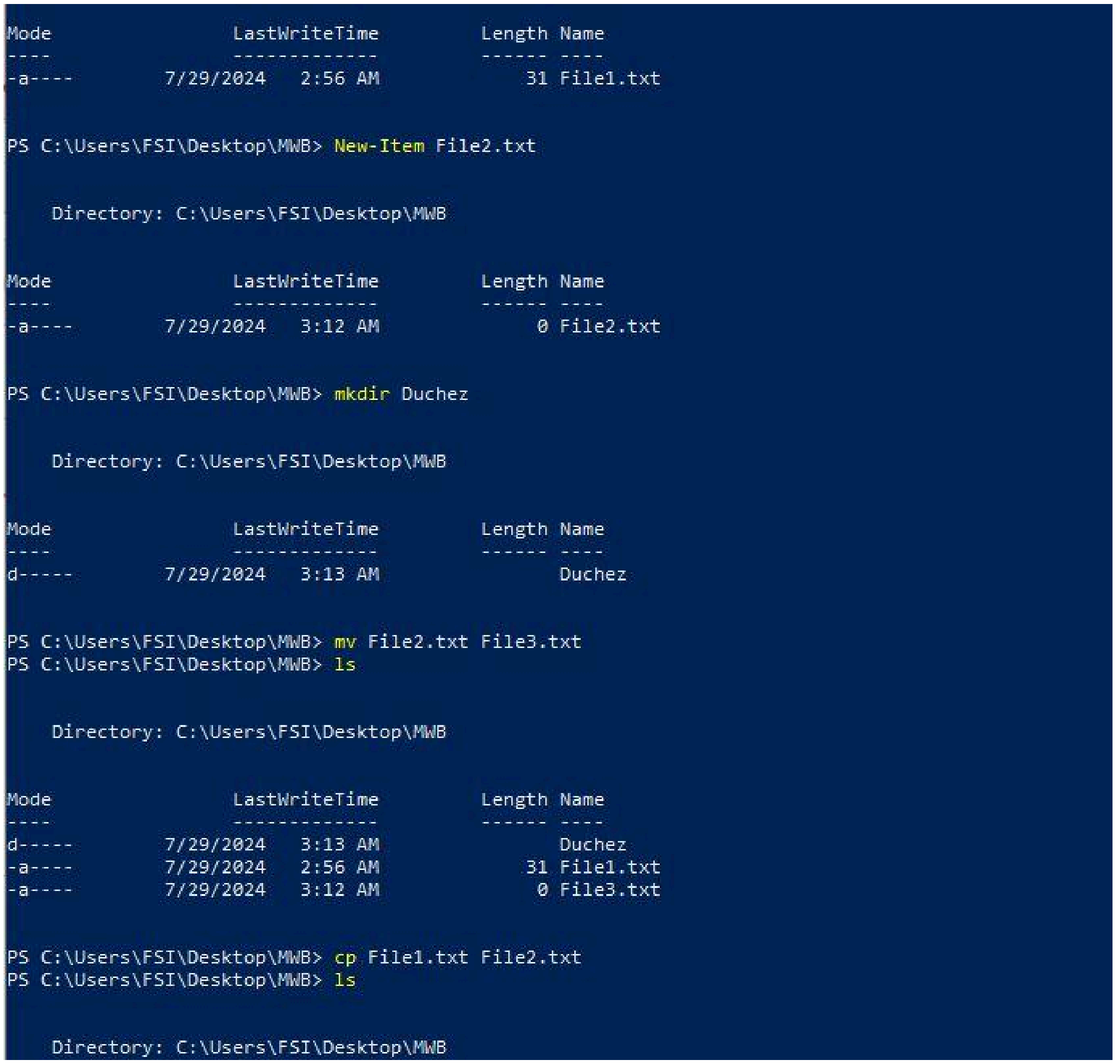


```
PS C:\Users\FSI\Desktop\MWB> git add .
PS C:\Users\FSI\Desktop\MWB> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   File1.txt

PS C:\Users\FSI\Desktop\MWB> git commit -m "my first commit"
[master (root-commit) bdce3ef] my first commit
 1 file changed, 1 insertion(+)
 create mode 100644 File1.txt
PS C:\Users\FSI\Desktop\MWB> git remote add origin https://github.com/Duchez1/MWB.git
>> git branch -M main
>> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 244 bytes | 122.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Duchez1/MWB.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\FSI\Desktop\MWB> git push -u main
fatal: 'main' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
PS C:\Users\FSI\Desktop\MWB> git push
Everything up-to-date
PS C:\Users\FSI\Desktop\MWB>
```

5. **Try out all the different ways of renaming and moving files. Understand the differences between different options.**

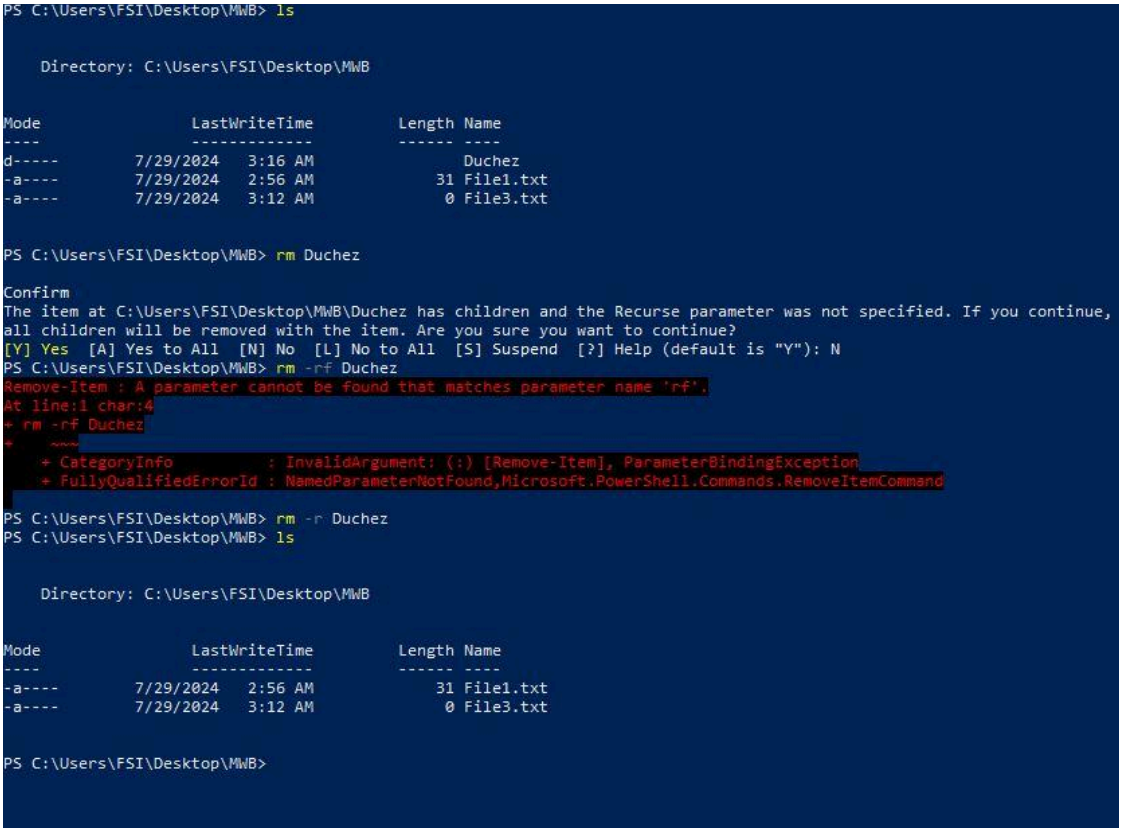


```
PS C:\Users\FSI\Desktop\MWB> ls


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        7/29/2024   2:56 AM             31 File1.txt


PS C:\Users\FSI\Desktop\MWB> New-Item File2.txt


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        7/29/2024   3:12 AM              0 File2.txt


PS C:\Users\FSI\Desktop\MWB> mkdir Duchez


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        7/29/2024   3:13 AM                Duchez


PS C:\Users\FSI\Desktop\MWB> mv File2.txt File3.txt
PS C:\Users\FSI\Desktop\MWB> ls

```



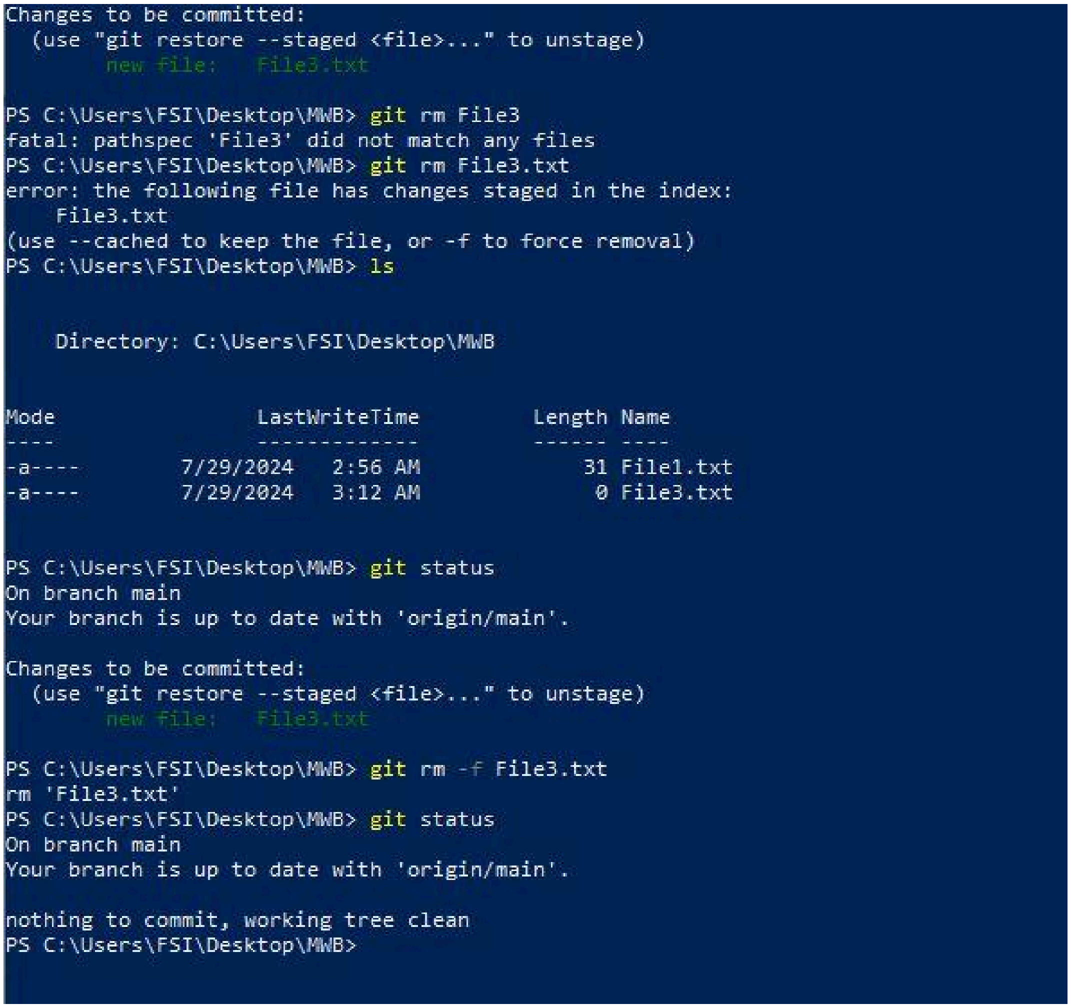
```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         7/29/2024   2:56 AM             31 File1.txt


PS C:\Users\FSI\Desktop\MWB> New-Item File2.txt


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         7/29/2024   3:12 AM              0 File2.txt


PS C:\Users\FSI\Desktop\MWB> mkdir Duchez


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         7/29/2024   3:13 AM                Duchez


PS C:\Users\FSI\Desktop\MWB> mv File2.txt File3.txt
PS C:\Users\FSI\Desktop\MWB> ls


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         7/29/2024   3:13 AM                Duchez
-a----         7/29/2024   2:56 AM             31 File1.txt
-a----         7/29/2024   3:12 AM              0 File3.txt


PS C:\Users\FSI\Desktop\MWB> cp File1.txt File2.txt
PS C:\Users\FSI\Desktop\MWB> ls


    Directory: C:\Users\FSI\Desktop\MWB
```

6. **You just created a new file in your Git project, but then you decided that the file is to be removed. How do you delete this untracked file.**

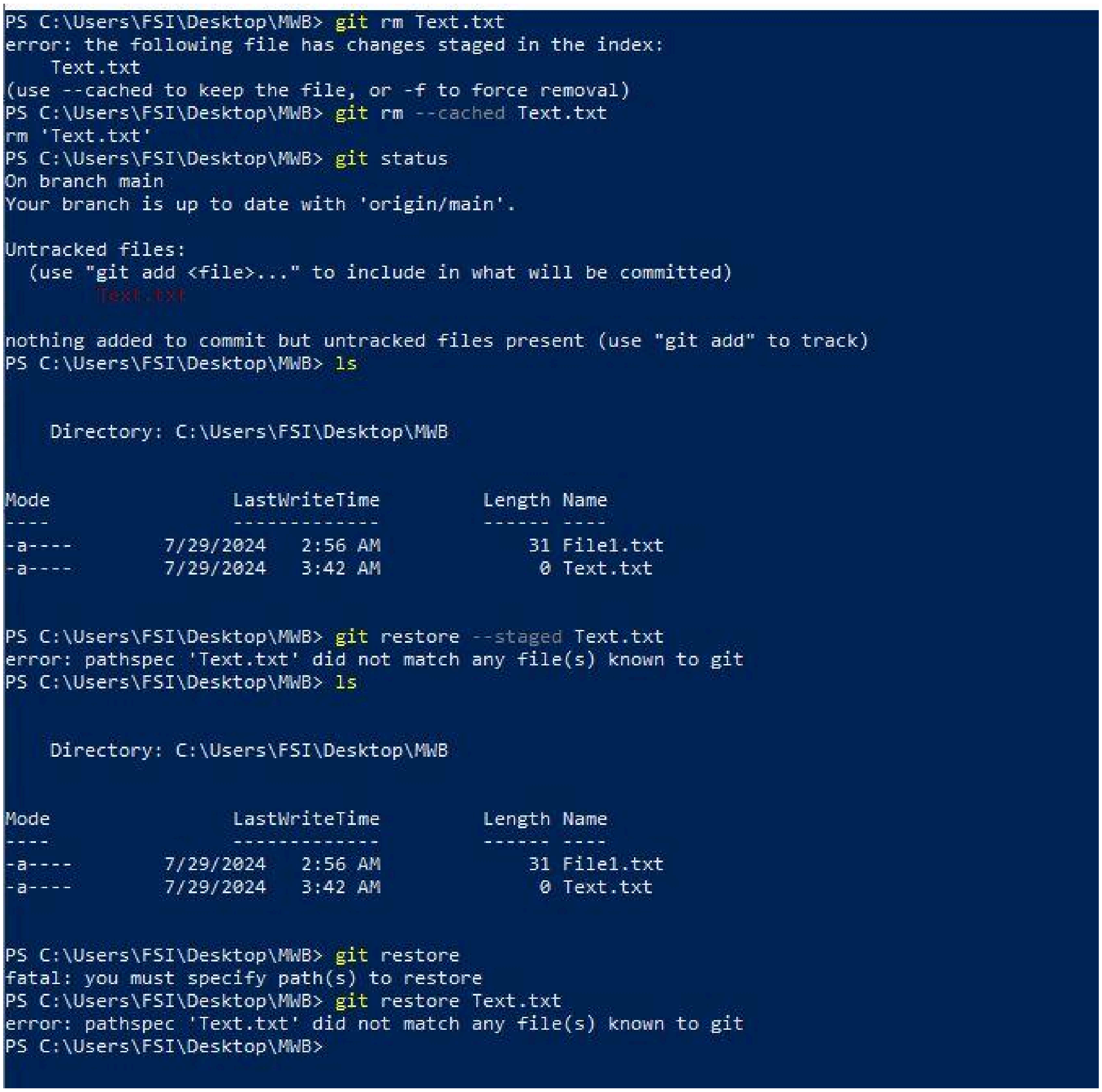


```
PS C:\Users\FSI\Desktop\MWB> ls


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         7/29/2024   3:16 AM                Duchez
-a----         7/29/2024   2:56 AM             31 File1.txt
-a----         7/29/2024   3:12 AM              0 File3.txt


PS C:\Users\FSI\Desktop\MWB> rm Duchez

Confirm
The item at C:\Users\FSI\Desktop\MWB\Duchez has children and the Recurse parameter was not specified. If you continue,
all children will be removed with the item. Are you sure you want to continue?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): N
PS C:\Users\FSI\Desktop\MWB> rm -rf Duchez
Remove-Item : A parameter cannot be found that matches parameter name 'rf'.
At line:1 char:4
+ rm -rf Duchez
+    ~~~
    + CategoryInfo          : InvalidArgument: (:) [Remove-Item], ParameterBindingException
    + FullyQualifiedErrorId : NamedParameterNotFound,Microsoft.PowerShell.Commands.RemoveItemCommand

PS C:\Users\FSI\Desktop\MWB> rm -r Duchez
PS C:\Users\FSI\Desktop\MWB> ls


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         7/29/2024   2:56 AM             31 File1.txt
-a----         7/29/2024   3:12 AM              0 File3.txt


PS C:\Users\FSI\Desktop\MWB>
```

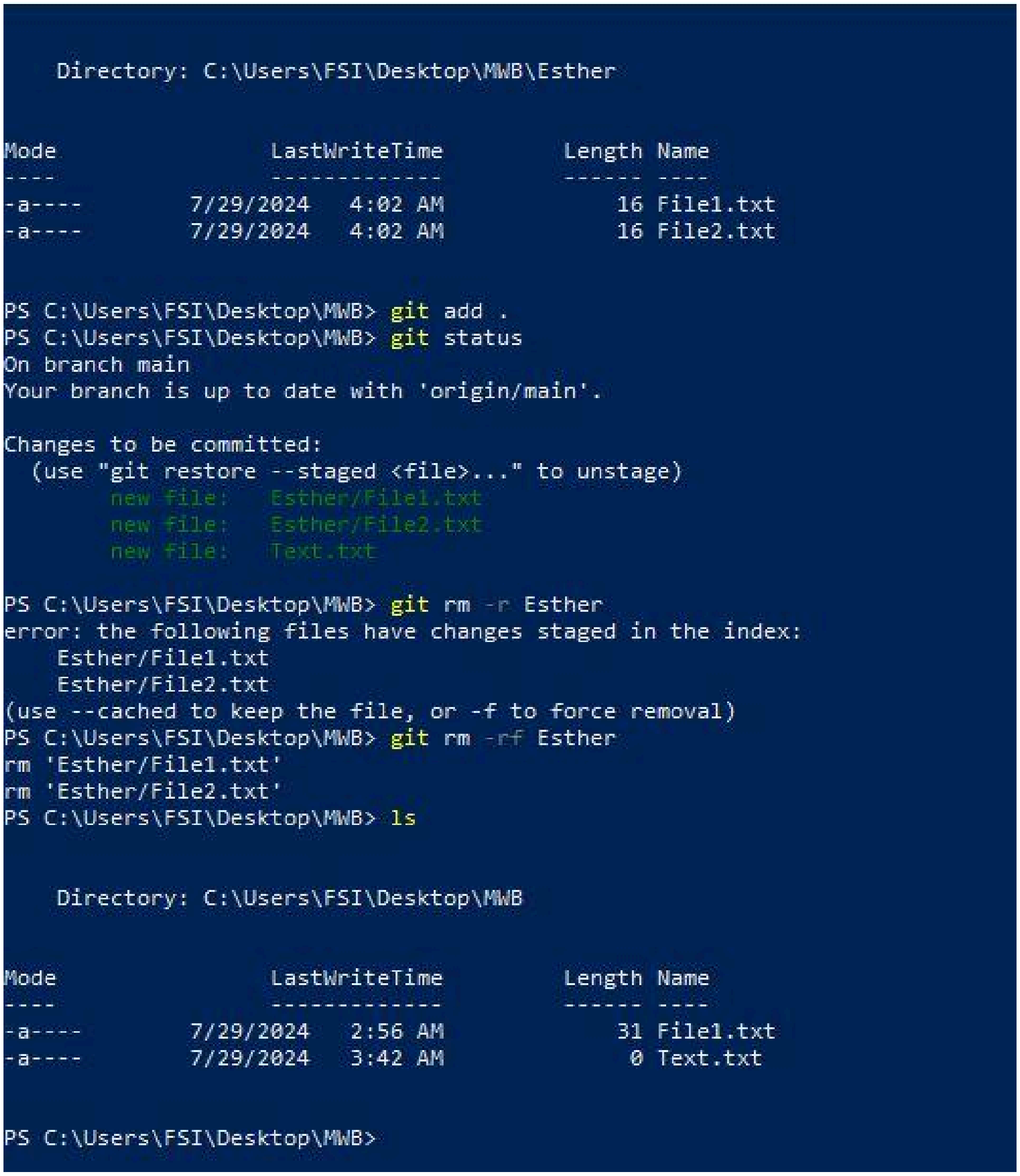
7. **Demonstrate the following: a. delete of a tracked file b. backing out staged deletion c. recursive deletion**



```
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   File3.txt

PS C:\Users\FSI\Desktop\MWB> git rm File3
fatal: pathspec 'File3' did not match any files
PS C:\Users\FSI\Desktop\MWB> git rm File3.txt
error: the following file has changes staged in the index:
    File3.txt
(use --cached to keep the file, or -f to force removal)
PS C:\Users\FSI\Desktop\MWB> ls


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         7/29/2024   2:56 AM             31 File1.txt
-a----         7/29/2024   3:12 AM              0 File3.txt


PS C:\Users\FSI\Desktop\MWB> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   File3.txt

PS C:\Users\FSI\Desktop\MWB> git rm -f File3.txt
rm 'File3.txt'
PS C:\Users\FSI\Desktop\MWB> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Users\FSI\Desktop\MWB>
```



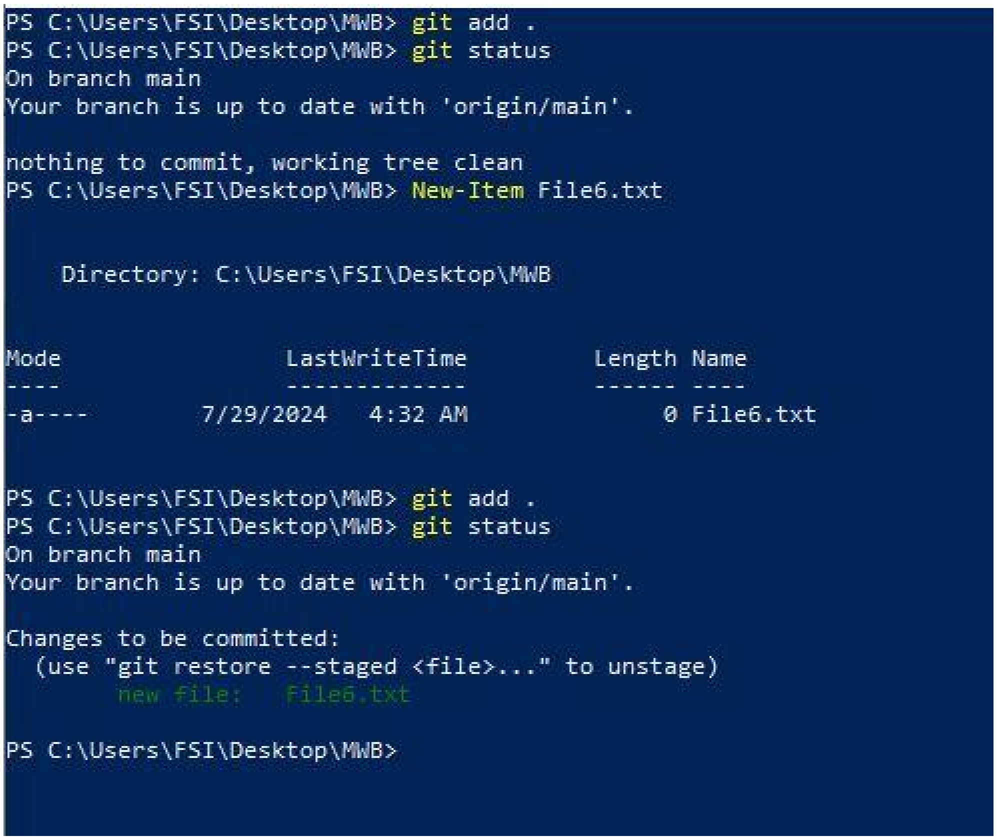
```
PS C:\Users\FSI\Desktop\MWB> git rm Text.txt
error: the following file has changes staged in the index:
    Text.txt
(use --cached to keep the file, or -f to force removal)
PS C:\Users\FSI\Desktop\MWB> git rm --cached Text.txt
rm 'Text.txt'
PS C:\Users\FSI\Desktop\MWB> git status
On branch main
Your branch is up to date with 'origin/main'.

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\FSI\Desktop\MWB> ls


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         7/29/2024   2:56 AM             31 File1.txt
-a----         7/29/2024   3:42 AM              0 Text.txt


PS C:\Users\FSI\Desktop\MWB> git restore --staged Text.txt
error: pathspec 'Text.txt' did not match any file(s) known to git
PS C:\Users\FSI\Desktop\MWB> ls


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         7/29/2024   2:56 AM             31 File1.txt
-a----         7/29/2024   3:42 AM              0 Text.txt


PS C:\Users\FSI\Desktop\MWB> git restore
fatal: you must specify path(s) to restore
PS C:\Users\FSI\Desktop\MWB> git restore Text.txt
error: pathspec 'Text.txt' did not match any file(s) known to git
PS C:\Users\FSI\Desktop\MWB>
```



```
    Directory: C:\Users\FSI\Desktop\MWB\Esther


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         7/29/2024   4:02 AM             16 File1.txt
-a----         7/29/2024   4:02 AM             16 File2.txt


PS C:\Users\FSI\Desktop\MWB> git add .
PS C:\Users\FSI\Desktop\MWB> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Esther/File1.txt
        new file:   Esther/File2.txt
        new file:   Text.txt

PS C:\Users\FSI\Desktop\MWB> git rm -r Esther
error: the following files have changes staged in the index:
    Esther/File1.txt
    Esther/File2.txt
(use --cached to keep the file, or -f to force removal)
PS C:\Users\FSI\Desktop\MWB> git rm -rf Esther
rm 'Esther/File1.txt'
rm 'Esther/File2.txt'
PS C:\Users\FSI\Desktop\MWB> ls


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         7/29/2024   2:56 AM             31 File1.txt
-a----         7/29/2024   3:42 AM              0 Text.txt


PS C:\Users\FSI\Desktop\MWB>
```

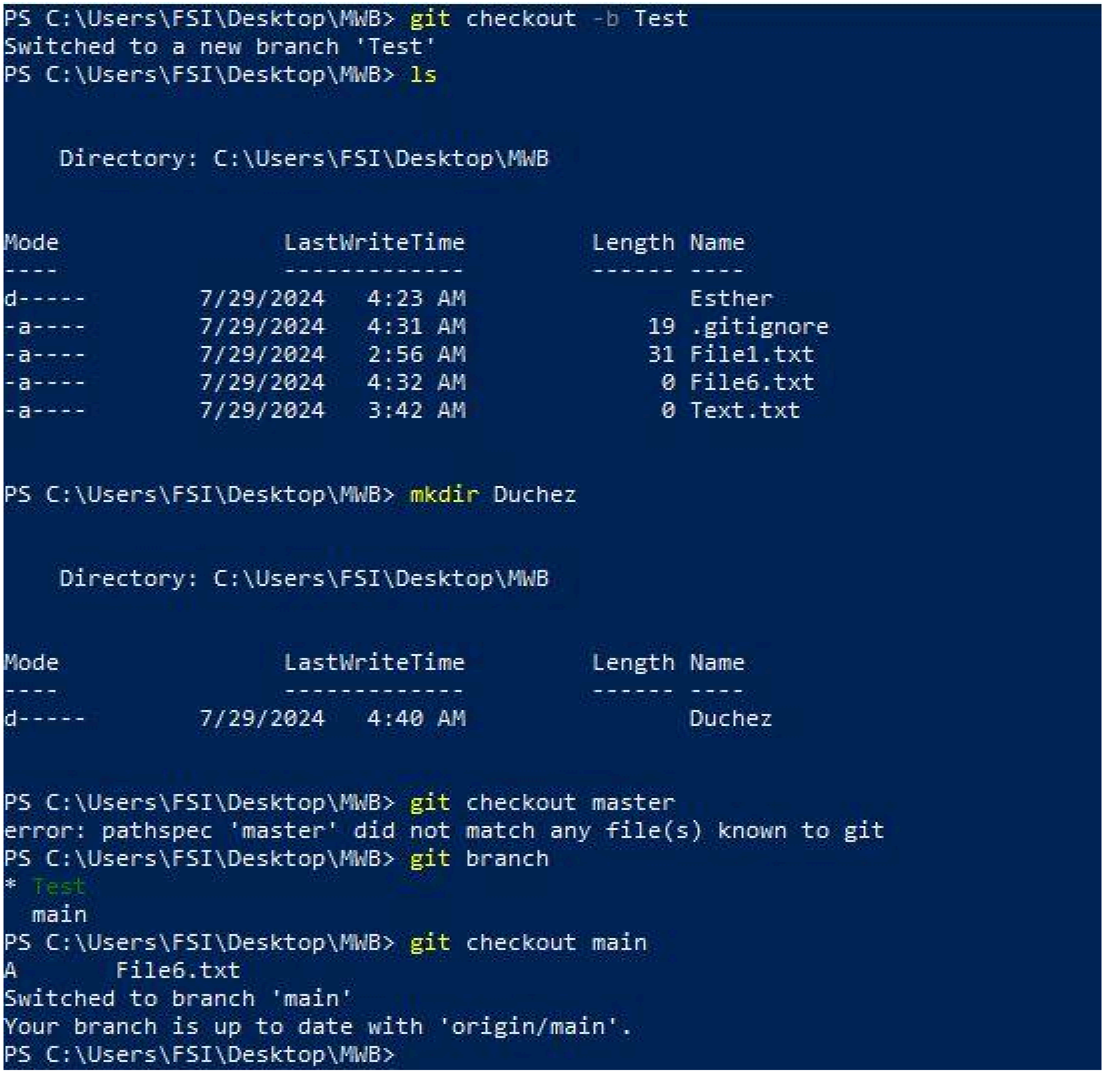
8. **You have lot of changes in your Git project but you do not want to push certain folders/files of your project. How do you manage this?**

Create a .gitignore file and add the name/ regular expression matching every file/folder you wish to ignore.

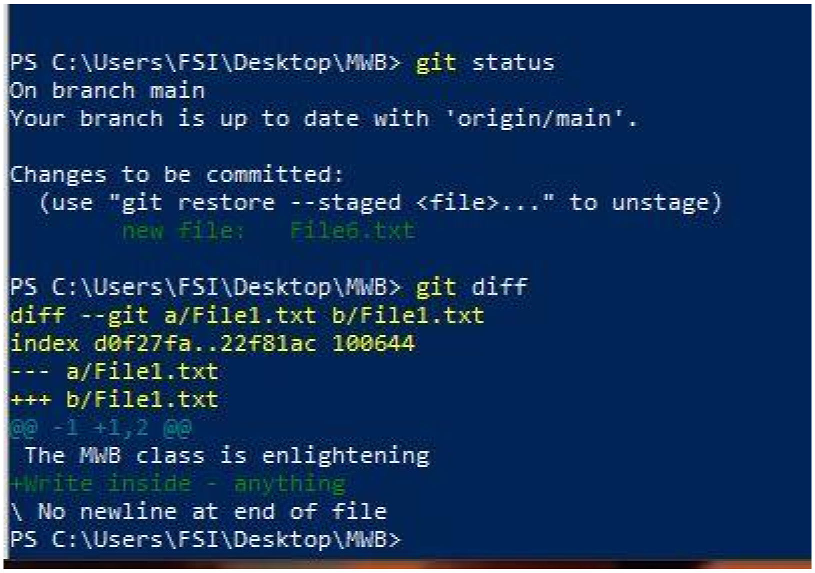


```
PS C:\Users\FSI\Desktop\MWB> git add .
PS C:\Users\FSI\Desktop\MWB> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Users\FSI\Desktop\MWB> New-Item File6.txt


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         7/29/2024   4:32 AM              0 File6.txt


PS C:\Users\FSI\Desktop\MWB> git add .
PS C:\Users\FSI\Desktop\MWB> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   File6.txt

PS C:\Users\FSI\Desktop\MWB>
```

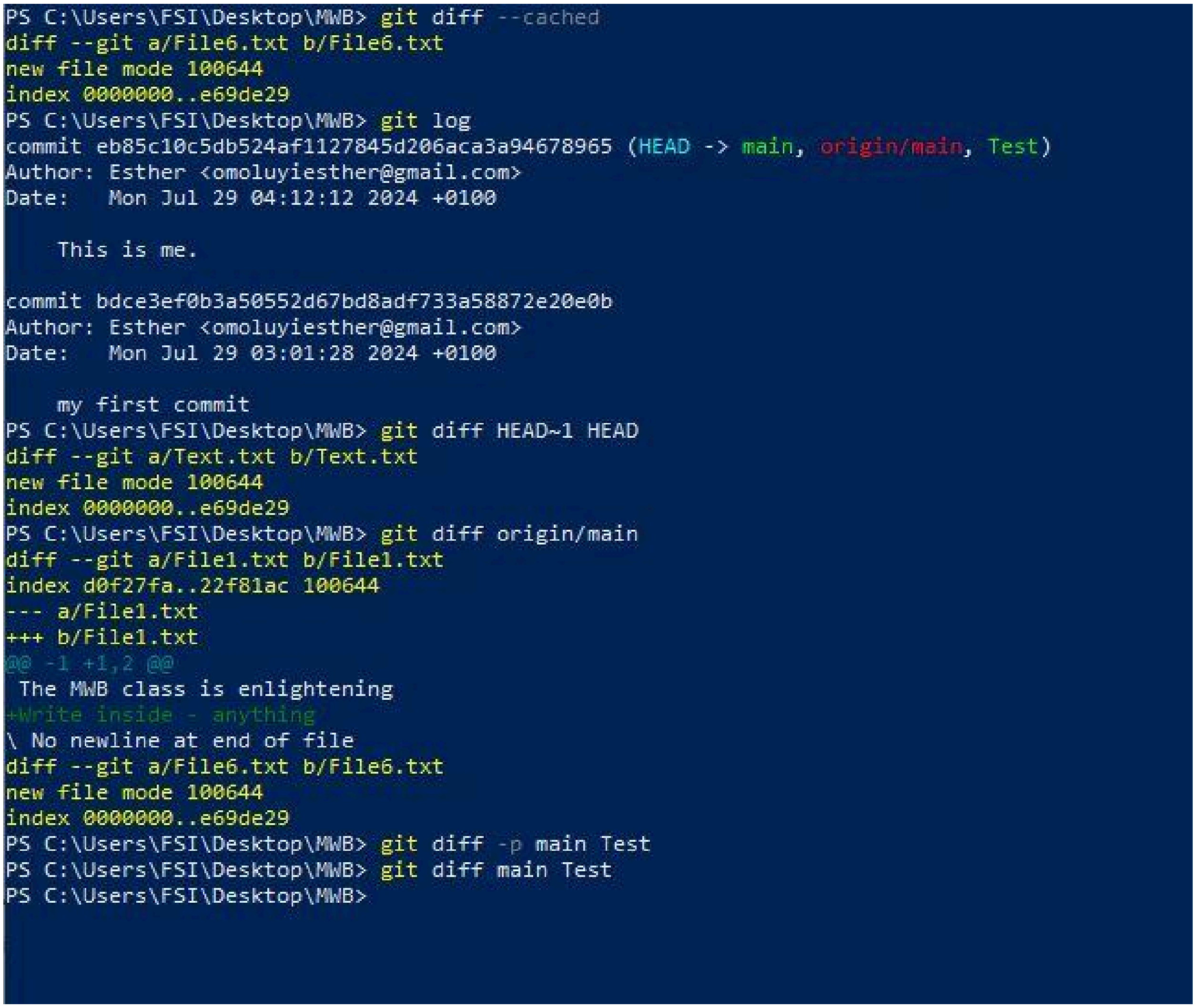
9. **Create a branch called "test". Make some changes in the master branch. Let there be some changes in the working directory and some in the staging area. Make some changes in the test branch as well. Issue the command to show the differences for**
    a. **Working directory vs Staging area**
    b. **Working directory vs Local Repository**
    c. **Staging area vs Local Repository**
    d. **Between two commits**
    e. **Between two tags**
    f. **Local vs Remote Repository**
    g. **Master branch vs test branch**



```
PS C:\Users\FSI\Desktop\MWB> git checkout -b Test
Switched to a new branch 'Test'
PS C:\Users\FSI\Desktop\MWB> ls


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         7/29/2024   4:23 AM                Esther
-a----         7/29/2024   4:31 AM             19 .gitignore
-a----         7/29/2024   2:56 AM             31 File1.txt
-a----         7/29/2024   4:32 AM              0 File6.txt
-a----         7/29/2024   3:42 AM              0 Text.txt


PS C:\Users\FSI\Desktop\MWB> mkdir Duchez


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         7/29/2024   4:40 AM                Duchez


PS C:\Users\FSI\Desktop\MWB> git checkout master
error: pathspec 'master' did not match any file(s) known to git
PS C:\Users\FSI\Desktop\MWB> git branch
* Test
  main
PS C:\Users\FSI\Desktop\MWB> git checkout main
A       File6.txt
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\FSI\Desktop\MWB>
```



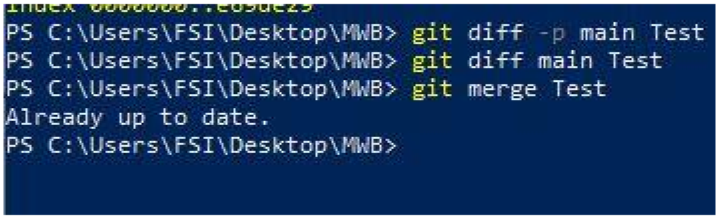
```
PS C:\Users\FSI\Desktop\MWB> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   File6.txt

PS C:\Users\FSI\Desktop\MWB> git diff
diff --git a/File1.txt b/File1.txt
index d0f27fa..22f81ac 100644
--- a/File1.txt
+++ b/File1.txt
@@ -1 +1,2 @@
 The MWB class is enlightening
+Write inside - anything
\ No newline at end of file
PS C:\Users\FSI\Desktop\MWB>
```



```
PS C:\Users\FSI\Desktop\MWB> git diff --cached
diff --git a/File6.txt b/File6.txt
new file mode 100644
index 0000000..e69de29
PS C:\Users\FSI\Desktop\MWB> git log
commit eb85c10c5db524af1127845d206aca3a94678965 (HEAD -> main, origin/main, Test)
Author: Esther <omoluyiesther@gmail.com>
Date:   Mon Jul 29 04:12:12 2024 +0100

    This is me.

commit bdce3ef0b3a50552d67bd8adf733a58872e20e0b
Author: Esther <omoluyiesther@gmail.com>
Date:   Mon Jul 29 03:01:28 2024 +0100

    my first commit
PS C:\Users\FSI\Desktop\MWB> git diff HEAD~1 HEAD
diff --git a/Text.txt b/Text.txt
new file mode 100644
index 0000000..e69de29
PS C:\Users\FSI\Desktop\MWB> git diff origin/main
diff --git a/File1.txt b/File1.txt
index d0f27fa..22f81ac 100644
--- a/File1.txt
+++ b/File1.txt
@@ -1 +1,2 @@
 The MWB class is enlightening
+Write inside - anything
\ No newline at end of file
diff --git a/File6.txt b/File6.txt
new file mode 100644
index 0000000..e69de29
PS C:\Users\FSI\Desktop\MWB> git diff -p main Test
PS C:\Users\FSI\Desktop\MWB> git diff main Test
PS C:\Users\FSI\Desktop\MWB>
```

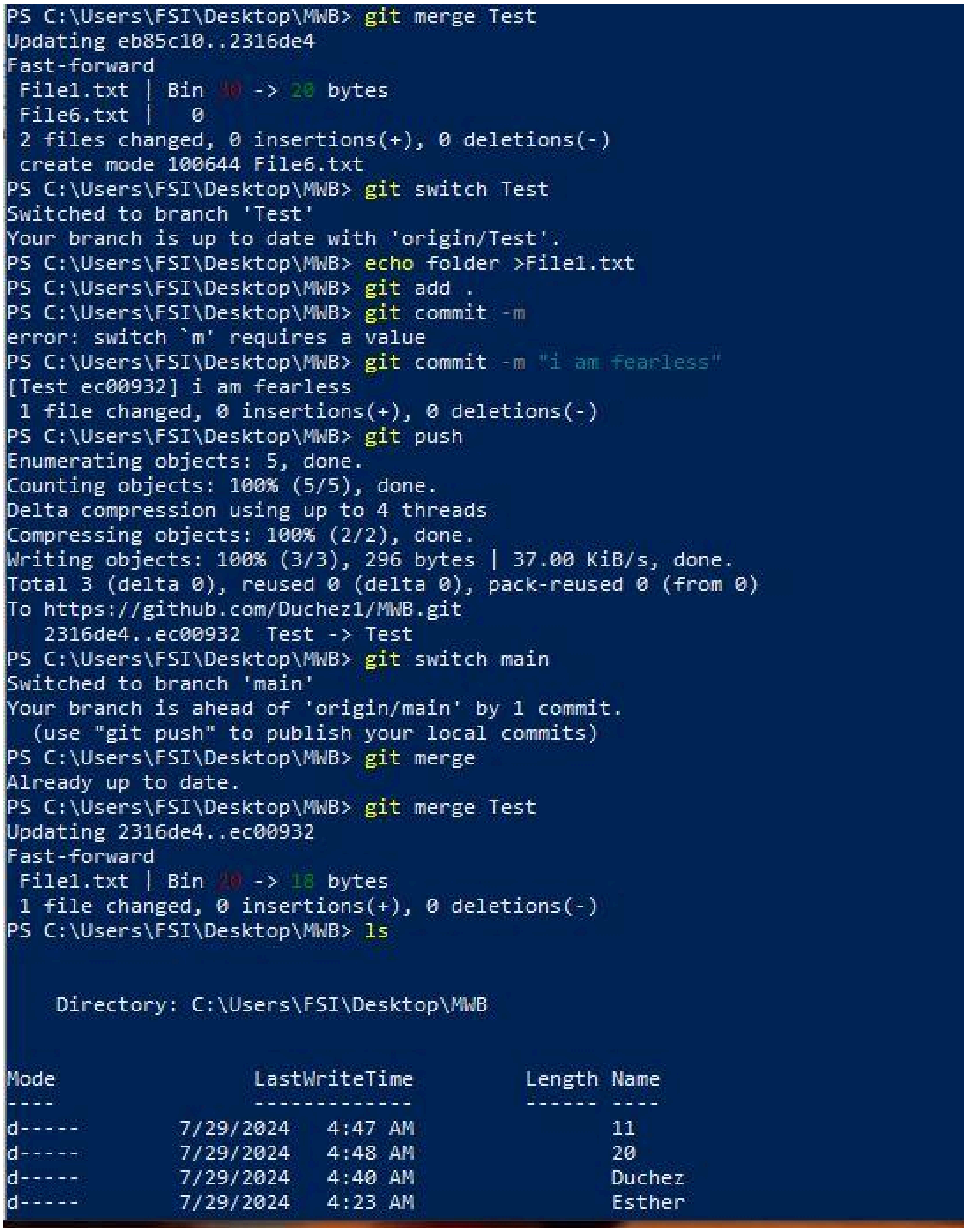
10. **Merge the changes from test branch to master branch.**
    a. **FastForward merge**
    b. **Disabling FastForward merge**
    c. **What is the difference between option (a) and option (b)**

No fast forward merge: Forces Git to create a new merge commit, even if a fast-forward is possible.



```
index 0000000..e69de29
PS C:\Users\FSI\Desktop\MWB> git diff -p main Test
PS C:\Users\FSI\Desktop\MWB> git diff main Test
PS C:\Users\FSI\Desktop\MWB> git merge Test
Already up to date.
PS C:\Users\FSI\Desktop\MWB>
```

11. **Create a merge conflict situation. Resolve the conflict and merge the changes between the branches.**



```
PS C:\Users\FSI\Desktop\MWB> git merge Test
Updating eb85c10..2316de4
Fast-forward
 File1.txt | Bin 40 -> 20 bytes
 File6.txt |   0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 File6.txt
PS C:\Users\FSI\Desktop\MWB> git switch Test
Switched to branch 'Test'
Your branch is up to date with 'origin/Test'.
PS C:\Users\FSI\Desktop\MWB> echo folder >File1.txt
PS C:\Users\FSI\Desktop\MWB> git add .
PS C:\Users\FSI\Desktop\MWB> git commit -m
error: switch `m' requires a value
PS C:\Users\FSI\Desktop\MWB> git commit -m "i am fearless"
[Test ec00932] i am fearless
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\FSI\Desktop\MWB> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 296 bytes | 37.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Duchez1/MWB.git
   2316de4..ec00932  Test -> Test
PS C:\Users\FSI\Desktop\MWB> git switch main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
PS C:\Users\FSI\Desktop\MWB> git merge
Already up to date.
PS C:\Users\FSI\Desktop\MWB> git merge Test
Updating 2316de4..ec00932
Fast-forward
 File1.txt | Bin 20 -> 18 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\FSI\Desktop\MWB> ls


    Directory: C:\Users\FSI\Desktop\MWB


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        7/29/2024   4:47 AM                11
d-----        7/29/2024   4:48 AM                20
d-----        7/29/2024   4:40 AM                Duchez
d-----        7/29/2024   4:23 AM                Esther
```

12. **What is the difference between merge and rebase, demonstrate with an example. Explain it.**

Merging is suitable for preserving the historical context of feature development (preserving the history of both branches, creating a merge commit, useful for preserving the context of branch development), while rebasing is beneficial for keeping a streamlined and linear project history (creating a linear history, rewrites commit history, useful for maintaining a clean project history without merging commits)

13. **With an example, demonstrate fetch, clone and pull. What is the use case for these operations. Are they same or different? Explain.**

Fetch is used to update your local repository with commits and updates from the remote repository without merging those changes into your current branch. It fetches updates but does not apply them.

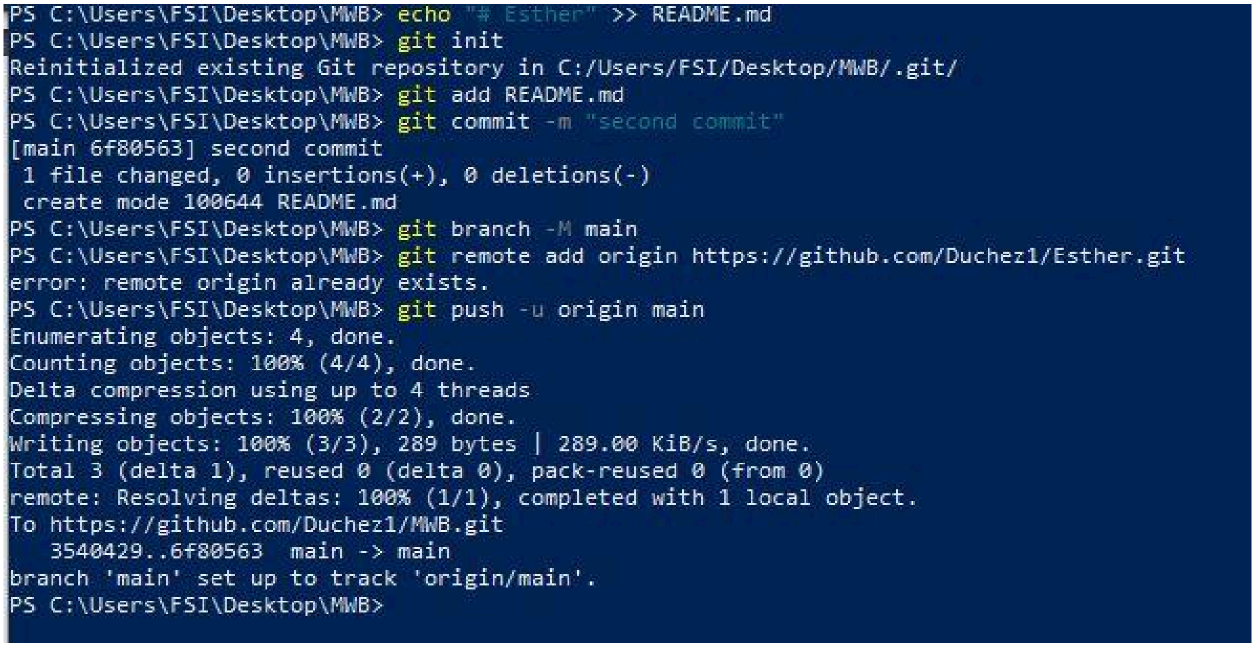
Clone is used to create a copy of a remote repository on your local machine. It is typically the first command you use when you want to work with an existing repository hosted remotely.

Pull is used to fetch updates from the remote repository and then immediately merge those updates into your current branch. It is a combination of git fetch and git merge.

	Fetch	Clone	Pull
Purpose	Download updates from the remote repository without merging them.	Create a local copy of a remote repository.	Fetch updates from the remote repository and merge them into the current branch.
Usage	Keep your local repository up to date with the remote repository without affecting your current branch.	Initial setup to start working with a repository.	Integrate changes from the remote repository into your current work.
Result	Updates are stored in the local	A new directory with the full	The working directory is

	repository, but the working directory remains unchanged.	repository history and files.	updated with changes from the remote repository and merged into the current branch.

14. Create a new repository in Github/Gitlab, with a README file. While pushing to the remote repository, if the remote branch is ahead of the local repository (new file is added in remote repository, which is not there in local repository) and pull is failing, how do you solve this problem?



```
PS C:\Users\FSI\Desktop\MWB> echo "# Esther" >> README.md
PS C:\Users\FSI\Desktop\MWB> git init
Reinitialized existing Git repository in C:/Users/FSI/Desktop/MWB/.git/
PS C:\Users\FSI\Desktop\MWB> git add README.md
PS C:\Users\FSI\Desktop\MWB> git commit -m "second commit"
[main 6f80563] second commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
PS C:\Users\FSI\Desktop\MWB> git branch -M main
PS C:\Users\FSI\Desktop\MWB> git remote add origin https://github.com/Duchez1/Esther.git
error: remote origin already exists.
PS C:\Users\FSI\Desktop\MWB> git push -u origin main
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), 289 bytes | 289.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Duchez1/MWB.git
   3540429..6f80563  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\FSI\Desktop\MWB>
```