1. Create one folder.
2. Open Git Bash and go to that folder.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~ (master)
$ cd C:\\Users\\aquis\\Desktop\\GitMerge

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$
```

3. Initialize git

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git init
Initialized empty Git repository in C:/Users/aquis/Desktop/GitMerge/.git/
```

4. Check status

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

5. Create on text file using touch command or vim editor.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ touch master1.txt

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ ls
master1.txt
```

6. Check status

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        master1.txt

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

7. Go to vim and add some content to file.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ vim  master1.txt

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ cat master1.txt
this is master1.txt under master branch

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        master1.txt

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

8. Convert the file from modified state to staged state and check the status.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git add .
warning: in the working copy of 'master1.txt', LF will be replaced by
t time Git touches it

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   master1.txt
```

9. Perform commit operation.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git commit -m "added master1.txt"
[master (root-commit) 032974d] added master1.txt
 1 file changed, 1 insertion(+)
 create mode 100644 master1.txt
```

10. Now, Create new branch say “hiray”.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git branch hiray
```

11. Switch to hiray branch by using a command → checkout

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git checkout hiray
Switched to branch 'hiray'
```

12. Check status

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (hiray)
$ git status
On branch hiray
nothing to commit, working tree clean
```

13. Create new text file as “hiray.txt” and check status.

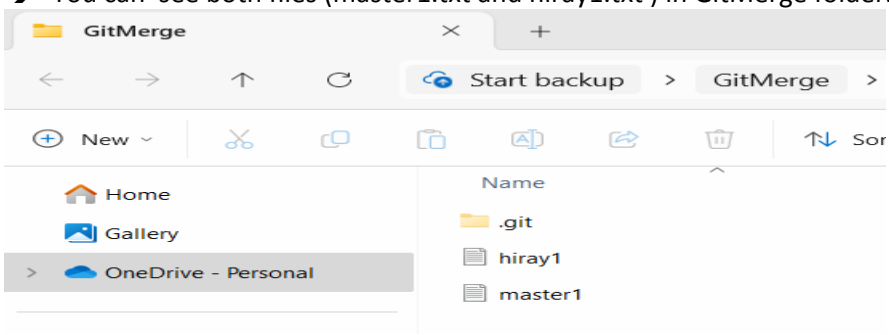
```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (hiray)
$ vim hiray1.txt

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (hiray)
$ cat hiray1.txt
this is hiray1.txt under hiray branch

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (hiray)
$ git status
On branch hiray
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hiray1.txt

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

→ Showing untracked file.

14. Add the file, perform commit operation and check the status.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (hiray)
$ git add .
warning: in the working copy of 'hiray1.txt', LF will be replac
 time Git touches it

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (hiray)
$ git commit -m "added hiray1.txt"
[hiray b80ba18] added hiray1.txt
 1 file changed, 1 insertion(+)
 create mode 100644 hiray1.txt

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (hiray)
$ git status
On branch hiray
nothing to commit, working tree clean
```

➔ You can see both files (master1.txt and hiray1.txt ) in GitMerge folder.



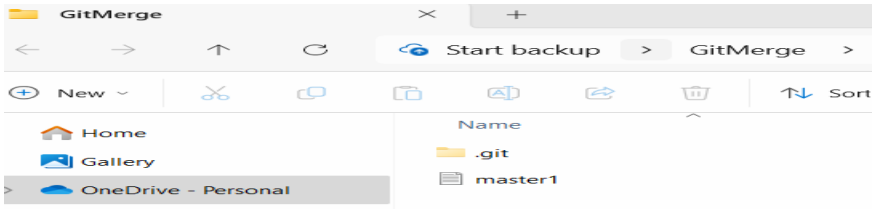

15. Check log → (it will show two commits)

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (hiray)
$ git log
commit b80ba18b49cf4cd6a0f3e28e17410d2dad3e17e9 (HEAD -> hiray)
Author: Aquila <aquishaikh@gmail.com>
Date:   Sun Mar 23 10:14:35 2025 +0530

    added hiray1.txt

commit 032974df77e5028922882884f2a68147bf6b8453 (master)
Author: Aquila <aquishaikh@gmail.com>
Date:   Sun Mar 23 10:10:47 2025 +0530

    added master1.txt
```

16. Switch to Master branch.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (hiray)
$ git checkout master
Switched to branch 'master'

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git status
On branch master
nothing to commit, working tree clean
```

17. If you switch to master branch, you won't be able to see "hiray1.txt" in the folder.




18. Let's create one more text file at master branch

➔ Perform add and commit and check log (you will be able to see two commits at master branch)

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ vim master2.txt

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ cat master2.txt
this is master2.txt under master branch

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        master2.txt

nothing added to commit but untracked files present (use "git add" to track)
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git add .
warning: in the working copy of 'master2.txt', LF will be replaced by
t time Git touches it

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git commit -m "added master2.txt"
[master a9448c3] added master2.txt
 1 file changed, 1 insertion(+)
 create mode 100644 master2.txt
```

19. Suppose at this point the changes have happened at master branch are relevant to hiray branch
20. And we want to get all changes from master branch to hiray branch.
21. **There are two options: Git merge and Git rebase**
22. Checkout to hiray branch and perform merge operation.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (master)
$ git checkout hiray
Switched to branch 'hiray'

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitMerge (hiray)
$ git status
On branch hiray
nothing to commit, working tree clean
```

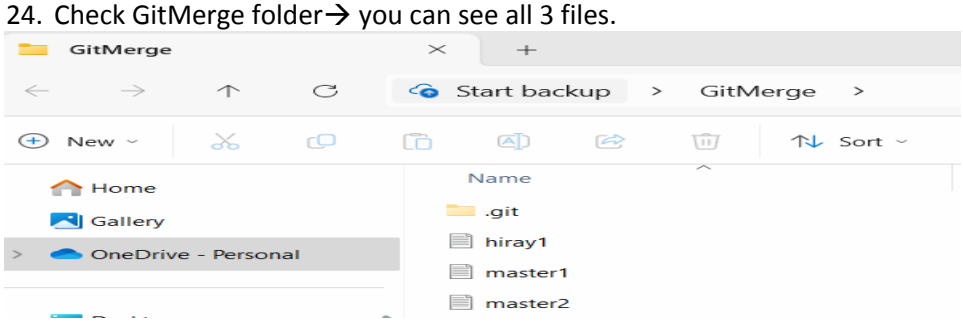
```
Merge branch 'master' into hiray
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
# performing merger at hiray
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
.git/MERGE_MSG[+] [unix] (10:22 23/03/2025)                    4,29 All
-- INSERT --
```

```
Merge made by the 'ort' strategy.
 master2.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 master2.txt
```

23. Check log

```
commit 83e9ee5632d40ffa240aec9290bd55c788cbf9ff (HEAD -> hiray)
Merge: b80ba18 a9448c3
Author: Aquila <aquishaikh@gmail.com>
Date:   Sun Mar 23 10:22:52 2025 +0530

    Merge branch 'master' into hiray

commit a9448c3cb7c88c53e1da769c6a381b699fc96ed7 (master)
Author: Aquila <aquishaikh@gmail.com>
Date:   Sun Mar 23 10:20:28 2025 +0530

    added master2.txt

commit b80ba18b49cf4cd6a0f3e28e17410d2dad3e17e9
Author: Aquila <aquishaikh@gmail.com>
Date:   Sun Mar 23 10:14:35 2025 +0530

    added hiray1.txt

commit 032974df77e5028922882884f2a68147bf6b8453
Author: Aquila <aquishaikh@gmail.com>
Date:   Sun Mar 23 10:10:47 2025 +0530

    added master1.txt
~
```

24. Check GitMerge folder→ you can see all 3 files.




Git Rebase:

1. Create on folder say “GitRbase”.
2. Open Git bash, go to that folder and initialize git.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~ (master)
$ cd C:\\Users\\aquis\\Desktop\\GitRebase

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ git init
Initialized empty Git repository in C:/Users/aquis/Desktop/GitRebase/.git/
```

3. Check status

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

4. Create m1.txt at master branch, add it and perform commit.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ touch m1.txt

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ git add .

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ git commit -m "added m1.txt"
[master (root-commit) 0316c4c] added m1.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 m1.txt
```

5. Check log (it will show one commit for m1.txt)

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ git log
commit 0316c4caee39d0de815c268379ce40fe4fa7da43 (HEAD -> master)
Author: Aquila <aquishaikh@gmail.com>
Date:   Thu Mar 20 06:38:45 2025 +0530

    added m1.txt
```

6. Create new branch hiray and switch to hiray branch.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ git branch hiray

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ git checkout hiray
Switched to branch 'hiray'

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (hiray)
$ git status
On branch hiray
nothing to commit, working tree clean
```

7. Check log (showing only one commit → m1.txt)

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (hiray)
$ git log
commit 0316c4caee39d0de815c268379ce40fe4fa7da43 (HEAD -> hiray, master)
Author: Aquila <aquishaikh@gmail.com>
Date:   Thu Mar 20 06:38:45 2025 +0530

    added m1.txt
```

8. At hiray branch, we will create new file say (h1.txt)→ add it and perform commit operation.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (hiray)
$ touch h1.txt

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (hiray)
$ git add .
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (hiray)
$ git commit -m "added h1.txt"
[hiray d21a8b9] added h1.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 h1.txt

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (hiray)
$ git status
On branch hiray
nothing to commit, working tree clean
```

9. Check status

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (hiray)
$ git status
On branch hiray
nothing to commit, working tree clean
```

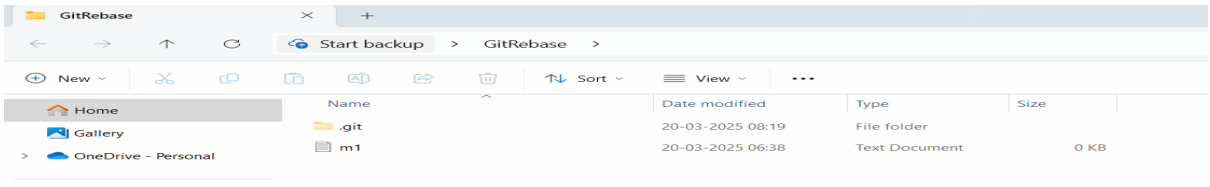
10. Check log (Showing 2 commits)

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (hiray)
$ git log
commit d21a8b940b67983831f1e14ddae9c1d60b10c8cd (HEAD -> hiray)
Author: Aquila <aquishaikh@gmail.com>
Date:   Thu Mar 20 07:44:57 2025 +0530

    added h1.txt

commit 0316c4caee39d0de815c268379ce40fe4fa7da43 (master)
Author: Aquila <aquishaikh@gmail.com>
Date:   Thu Mar 20 06:38:45 2025 +0530

    added m1.txt
```

11. Switch to Master branch

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (hiray)
$ git checkout master
Switched to branch 'master'
```

➔ As we swtich to master → we can see h1.txt is not there in the folder.




12. Now in master branch, we will create m2.txt, add it and perform commit operation.

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ touch m2.txt

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ git add .

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ git commit -m "added m2.txt"
[master c8d7cb2] added m2.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 m2.txt
```

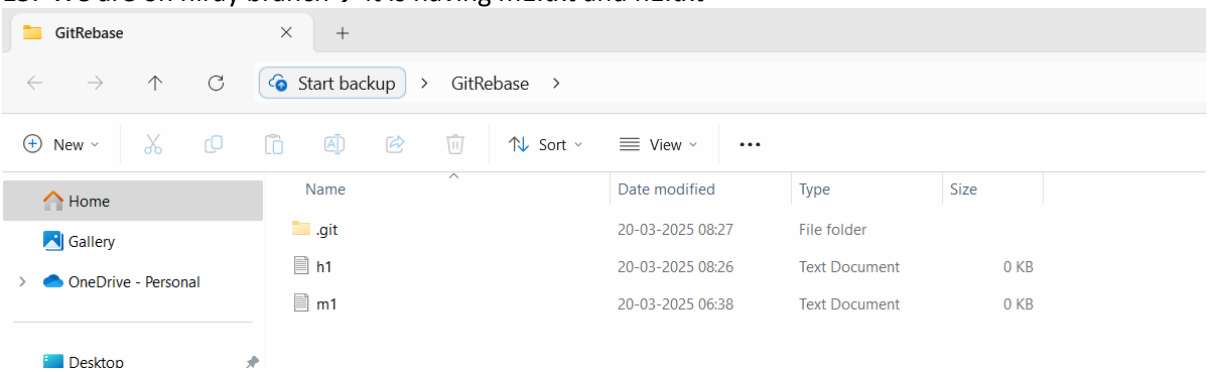
13. Check log (showing m1.txt and m2.txt)

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ git log
commit c8d7cb21fdba99a6b4da44f589dfe0c267141a42 (HEAD -> master)
Author: Aquila <aquishaikh@gmail.com>
Date:   Thu Mar 20 08:23:16 2025 +0530

    added m2.txt

commit 0316c4caee39d0de815c268379ce40fe4fa7da43
Author: Aquila <aquishaikh@gmail.com>
Date:   Thu Mar 20 06:38:45 2025 +0530

    added m1.txt
```

14. Switch to hiray branch

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (master)
$ git checkout hiray
Switched to branch 'hiray'

aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (hiray)
$ git status
On branch hiray
nothing to commit, working tree clean
```

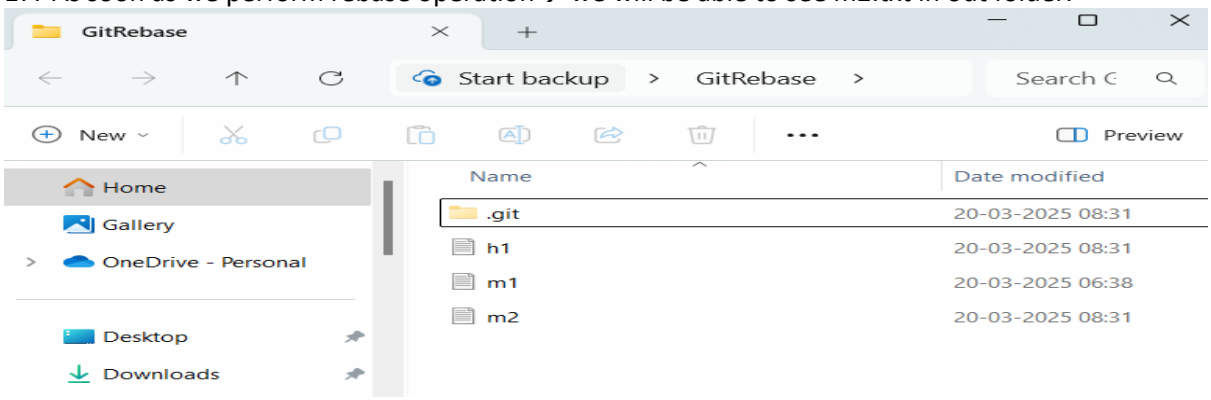
15. We are on hiray branch→ it is having m1.txt and h1.txt




16. Perform “rebase”

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (hiray)
$ git rebase master
Successfully rebased and updated refs/heads/hiray.
```

17. As soon as we perform rebase operation→ we will be able to see m2.txt in out folder.




18. Check log

```
aquis@LAPTOP-CVDR41EG MINGW64 ~/Desktop/GitRebase (hiray)
$ git log
commit c15d25ae451d4acad3a2669d7a9ceba4b40e6a8e (HEAD -> hiray)
Author: Aquila <aquishaikh@gmail.com>
Date:   Thu Mar 20 07:44:57 2025 +0530

    added h1.txt

commit c8d7cb21fdba99a6b4da44f589dfe0c267141a42 (master)
Author: Aquila <aquishaikh@gmail.com>
Date:   Thu Mar 20 08:23:16 2025 +0530

    added m2.txt

commit 0316c4caee39d0de815c268379ce40fe4fa7da43
Author: Aquila <aquishaikh@gmail.com>
Date:   Thu Mar 20 06:38:45 2025 +0530

    added m1.txt
```