

## Experiment-1.3

**Student Name:** Adarsh Kumar Singh

**UID:** 22BDO10053

**Branch:** CSE(DEVOPS)

**Section/Group:** 22BCD-1/B

**Semester:** 4th

**Date of Performance:** 02 - 02 -24

**Subject Name:** Git and GitHub

**Subject Code:** 22CSH-293

**1. Aim/Overview of the practical:** To create and explore Push/Pull requests.

**2. Task to be done:** Creating a file in your Local repository and pushing it to remote repository and pulling the file from the remote repository to local repository.

**3. Software Used:** Git Bash, GitHub.

**4. Steps for Experiment: -**

- 1) Configure your credentials before working on your local repository i.e.(user name , user email).

```
adars@ASUS MINGW64 ~ (master)
$ git config --global user.name "adarshkrsingh07"

adars@ASUS MINGW64 ~ (master)
$ git config --global user.email "adarshkrdixit@gmail.com"
```

- 2) Apart from this, clone your repository in local machine. It will make a complete copy of your repository.

```
adars@ASUS MINGW64 ~/22BCD-1 (main)
$ git clone https://github.com/adarshkrsingh07/22BCD-1.git
fatal: destination path '22BCD-1' already exists and is not an empty directory.
```

- 3) Check if a repository already exists, if not, then create a repository using (mkdir repo\_name), if yes, then go to your repository i.e. (cd 22BCD-1).
- 4) Now check for the branches and for the files i.e. (git branch & ls)

```
adars@ASUS MINGW64 ~ (master)
$ cd 22BCD-1

adars@ASUS MINGW64 ~/22BCD-1 (main)
$ git branch
* main
  testbranch

adars@ASUS MINGW64 ~/22BCD-1 (main)
$ ls
22BCD-1/  Add.c  File1.txt  File2.txt
```

- 5) Now create a file in Main\_branch by using **vi** i.e. (vi exp3.txt) and some text/message inside it.

```
adars@ASUS MINGW64 ~/22BCD-1 (main)
$ vi exp3.txt

adars@ASUS MINGW64 ~/22BCD-1 (main)
$ cat exp3.txt

this is for exp3.
```

- 6) Now, Adding the file to the staging area using **git add** and then commit the changes using the **git commit** command.

```
adars@ASUS MINGW64 ~/22BCD-1 (main)
$ git add .
warning: in the working copy of 'exp3.txt', LF will be replaced by CRLF the next time Git touches it

adars@ASUS MINGW64 ~/22BCD-1 (main)
$ git commit -m "exp3 commit in main"
[main e8748f9] exp3 commit in main
3 files changed, 1 insertion(+), 3 deletions(-)
delete mode 100644 F1exp3.txt
create mode 100644 exp3.txt
delete mode 100644 hello.txt
```

- 7) Now, check for the status in your main branch if any file was not added or committed then add & commit it.

```
adars@ASUS MINGW64 ~/22BCD-1 (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 3 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

nothing to commit, working tree clean
```

8) Now, go to the new branch i.e.(test\_Branch) and write some text/message inside it.

```
adars@ASUS MINGW64 ~/22BCD-1 (main)
$ git checkout testBranch
Switched to branch 'testBranch'
```

9) Now, open the same file in the (test\_Branch) which was created in Main\_branch and do some changes in it.

```
adars@ASUS MINGW64 ~/22BCD-1 (testBranch)
$ vi exp3.txt
```

10) Now, Add the file i.e. (git add .) so that it can be track and commit it i.e. (git commit) by using git commands.

```
adars@ASUS MINGW64 ~/22BCD-1 (testBranch)
$ git add .
warning: in the working copy of 'exp3.txt', LF will be replaced by CRLF the next time Git touches it

adars@ASUS MINGW64 ~/22BCD-1 (testBranch)
$ git commit -m "committed in tesbranhc"
[testBranch ebfc1d9] committed in tesbranhc
1 file changed, 1 insertion(+)
create mode 100644 exp3.txt
```

11) Now, merge the test\_Branch in the main\_Branch while merging it will check for the conflicts if arises then resolve the conflicts and then merge it.

12) By using **git diff** command we can see the changes.

```
adars@ASUS MINGW64 ~/22BCD-1 (main)
$ git merge testBranch
Auto-merging exp3.txt
CONFLICT (add/add): Merge conflict in exp3.txt
Automatic merge failed; fix conflicts and then commit the result.

adars@ASUS MINGW64 ~/22BCD-1 (main|MERGING)
$ git diff
diff --cc exp3.txt
index 222aac6,4bf5b39..0000000
--- a/exp3.txt
+++ b/exp3.txt
@@@ -1,1 -1,1 +1,5 @@@
++<<<<<<< HEAD
+this is for exp3.
+=====
+ writing in the test branch.
++>>>>>>> testBranch
```

13) Now, open the file (**vi exp3.txt**) again resolve the conflicts manually and commit it again.

```
adars@ASUS MINGW64 ~/22BCD-1 (main|MERGING)
$ vi exp3.txt

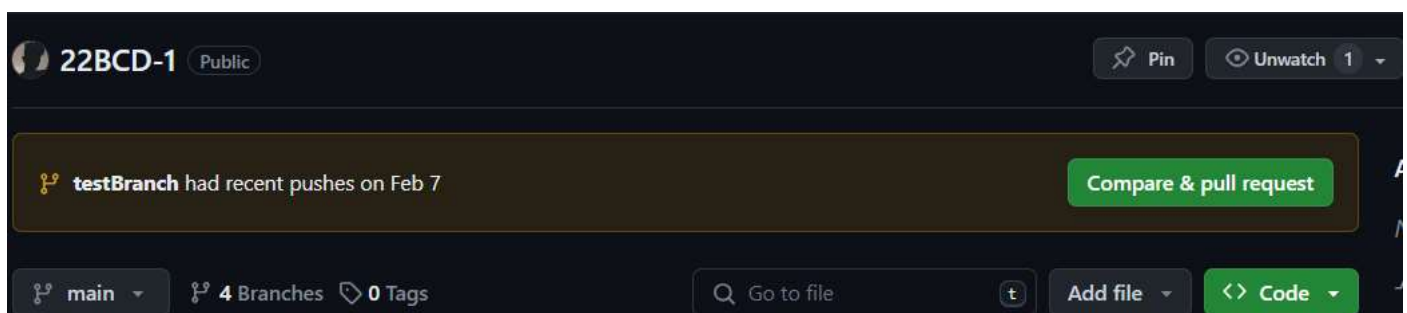
adars@ASUS MINGW64 ~/22BCD-1 (main|MERGING)
$ git add .

adars@ASUS MINGW64 ~/22BCD-1 (main|MERGING)
$ git commit -m "resolve conflicts"
[main d197a0e] resolve conflicts
```

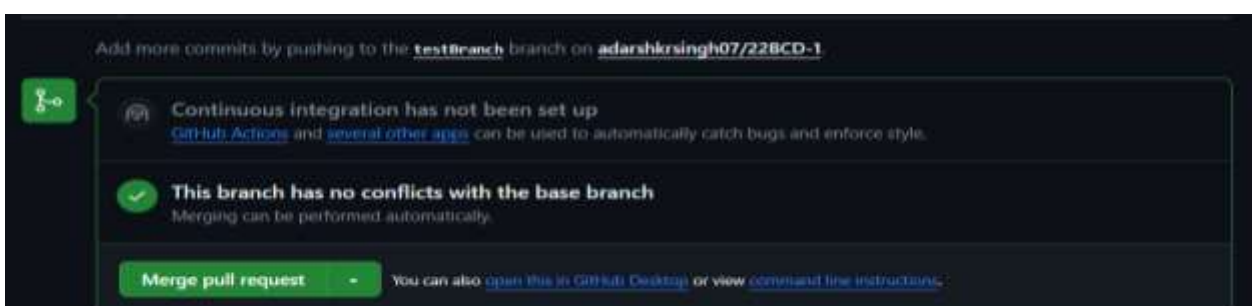
13) After resolving the conflicts, push it to the remote repository i.e. (**git push origin main**).

```
adars@ASUS MINGW64 ~/22BCD-1 (main)
$ git push origin main
Enumerating objects: 47, done.
Counting objects: 100% (43/43), done.
Delta compression using up to 8 threads
Compressing objects: 100% (27/27), done.
Writing objects: 100% (37/37), 3.36 KiB | 574.00 KiB/s, done.
Total 37 (delta 12), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (12/12), completed with 1 local object.
To https://github.com/adarshkrsingh07/22BCD-1.git
  9dcae1..d197a0e  main -> main
```

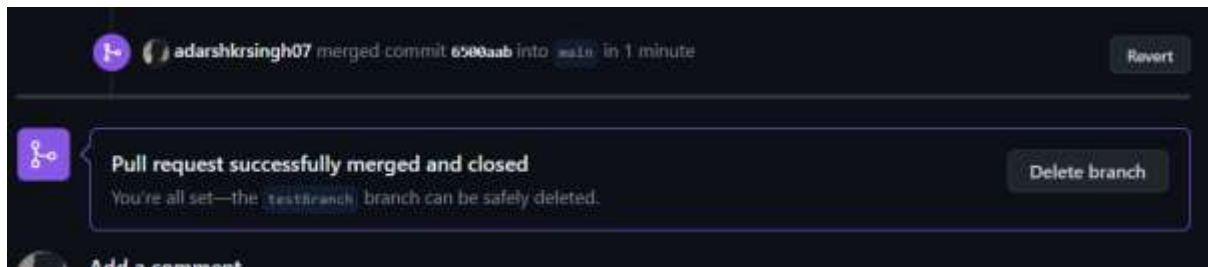
14) After pushing it to remote repository, go to your github account and create a compare and pull request and merge it from there.



15) check for the conflicts and if no conflicts arises it will automatically merge it.



16) the merging was successfully done without any conflicts.



17) Now go to your local repository and checkout to the (test\_branch) and push the files from test\_branch .

```
adars@ASUS MINGW64 ~/22BCD-1 (main)
$ git checkout testBranch
Switched to branch 'testBranch'

adars@ASUS MINGW64 ~/22BCD-1 (testBranch)
$ git push origin testBranch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/adarshkrsingh07/22BCD-1.git
 99334bf..ebfc1d9 testBranch -> testBranch
```

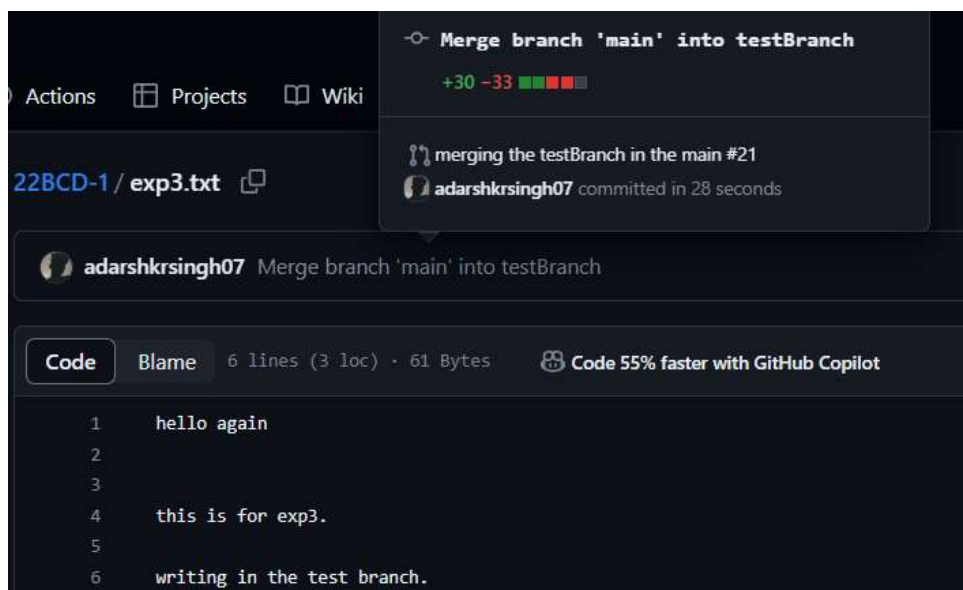
18) now, you may get the changes in your local repository using (**git pull**) and check the (**git status**) for any left file which was not committed.

```
adars@ASUS MINGW64 ~/22BCD-1 (testBranch)
$ git pull origin main
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), 2.80 KiB | 119.00 KiB/s, done.
From https://github.com/adarshkrsingh07/22BCD-1
 * branch                main          -> FETCH_HEAD
    d197a0e..6500aab      main          -> origin/main
Updating ebfc1d9..6500aab
Fast-forward
 .gitignore | 33 -----
 Add.c      | 17 ++++++++
 File1.txt  | 2 ++
 File2.txt  | 7 +++++
 exp3.txt   | 5 +++++
 5 files changed, 31 insertions(+), 33 deletions(-)
 delete mode 100644 .gitignore
 create mode 100644 Add.c

adars@ASUS MINGW64 ~/22BCD-1 (testBranch)
$ git status
On branch testBranch
nothing to commit, working tree clean
```



19) Now we can see the messages in the file which was done in the main and test\_Branch.



We have successfully created a file and merged in the Main Branch.

### Learning outcomes (What I have learnt):

1. Learnt about GitHub.
2. Learnt about Git.
3. Learnt about various git commands that can be applied on Git Bash.
4. Learnt about how to create repositories.
5. Learnt about how to pull request and push source code /files etc.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			