



## Experiment -2.1

**Student Name:** Ayush Pandey

**Branch:** CSE-DevOps

**Semester:** 4th

**Subject Name:** Git and Hub

**UID:** 22BDO10038

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

**Date of Performance:** 07/01/2024

**Subject Code:** 22CSH-293

**1. Aim/Overview of the practical:** Editing a file and committing changes on GitHub.

**2. Apparatus/Software Used:** Git Bash, GitHub.

**3. Steps for experiment/practical:**

- First open GitHub, create a repository ‘exp2.1\_editfile’ and clone a repository on git bash.

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \* ayush2442 / Repository name \* exp2.1\_editfile

exp2.1\_editfile is available.

Great repository names are short and memorable. Need inspiration? How about [psychic-broccoli](#) ?

Description (optional)

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

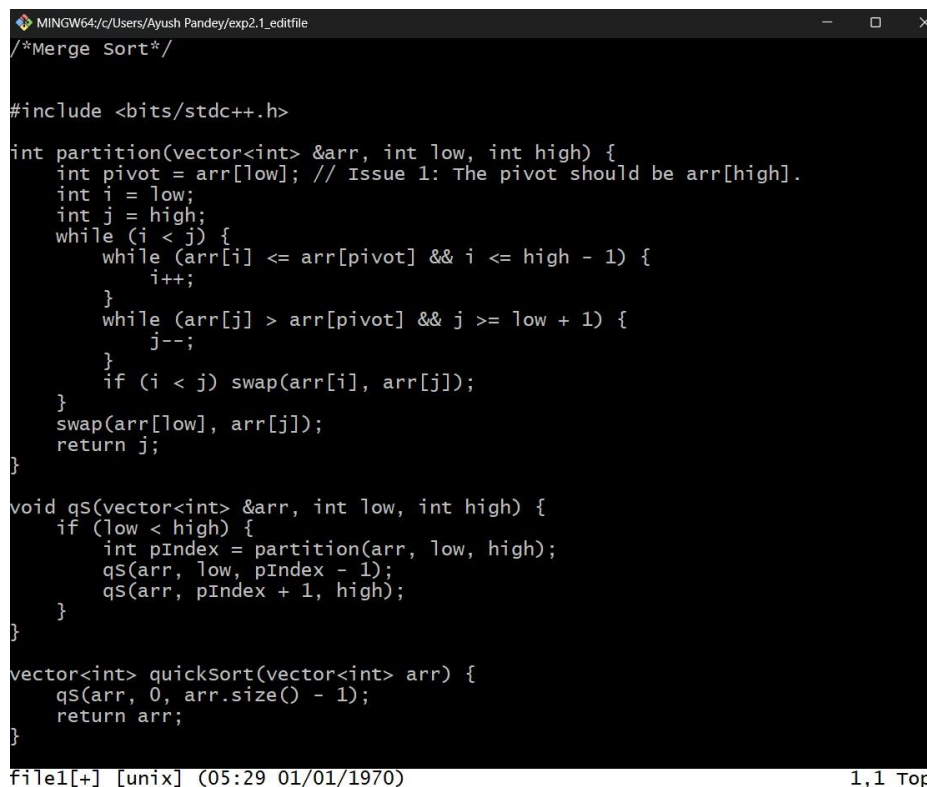
```
Ayush Pandey@Ayush MINGW64 ~ (master)
$ git clone https://github.com/ayush2442/exp2.1_editfile.git
Cloning into 'exp2.1_editfile'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack
-reused 0
Receiving objects: 100% (3/3), done.
```

- Then move to the directory using the **cd** command.

```
Ayush Pandey@Ayush MINGW64 ~ (master)
$ cd exp2.1_editfile

Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (main)
$ vi file1
```

- Now create a file named 'file1', and add some text to the files using vi editor.



```
MINGW64: c:/Users/Ayush Pandey/exp2.1_editfile
/*Merge Sort*/

#include <bits/stdc++.h>

int partition(vector<int> &arr, int low, int high) {
    int pivot = arr[low]; // Issue 1: The pivot should be arr[high].
    int i = low;
    int j = high;
    while (i < j) {
        while (arr[i] <= arr[pivot] && i <= high - 1) {
            i++;
        }
        while (arr[j] > arr[pivot] && j >= low + 1) {
            j--;
        }
        if (i < j) swap(arr[i], arr[j]);
    }
    swap(arr[low], arr[j]);
    return j;
}

void qs(vector<int> &arr, int low, int high) {
    if (low < high) {
        int pIndex = partition(arr, low, high);
        qs(arr, low, pIndex - 1);
        qs(arr, pIndex + 1, high);
    }
}

vector<int> quicksort(vector<int> arr) {
    qs(arr, 0, arr.size() - 1);
    return arr;
}

file1[+] [unix] (05:29 01/01/1970) 1,1 Top
```

- Add it to the staging area by git add command and commit it using git commit -m “message”.

```
Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (main)
$ git add file1

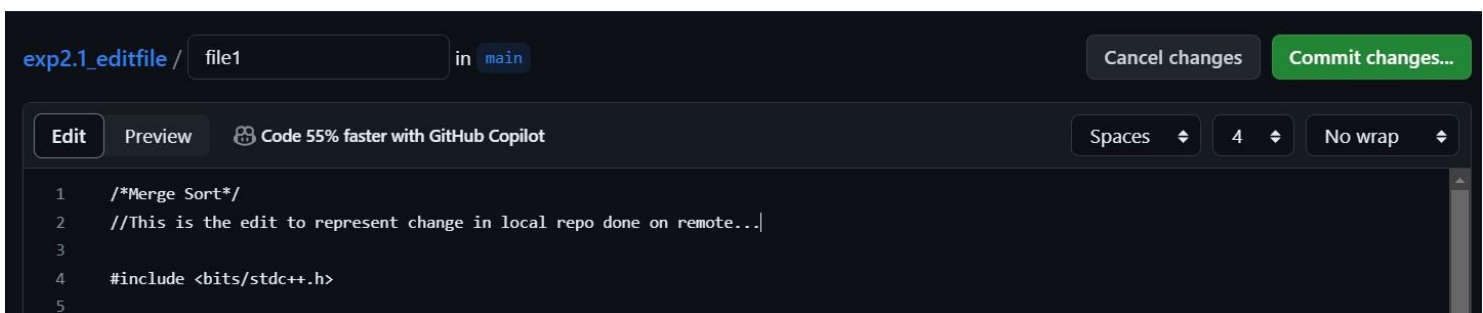
Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (main)
$ git commit -m "edited"
[main 7bd3ecc] edited
1 file changed, 47 insertions(+)
create mode 100644 file1
```

- Now push the changes to remote repository using command git push origin main.

```
Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 728 bytes | 728.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ayush2442/exp2.1_editfile.git
59a0581..7bd3ecc main -> main
```

- You can see the changes in the remote repository.

- Now make some changes in the remote repository and pull those changes in the local repository.



- Now, pull your changes using git pull origin main.

```
Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (main)
$ git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1.01 KiB | 172.00 KiB/s, done.
From https://github.com/ayush2442/exp2.1_editfile
* branch                main                -> FETCH_HEAD
   7bd3ecc..8c9dc56      main                -> origin/main
Updating 7bd3ecc..8c9dc56
Fast-forward
 file1 | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

- Now you can see the changes done into the 'file1' by command vi file1.
- Then create a new branch using the command 'git checkout -b branch name'.
- Open the 'file1' file on the vi editor and make some changes.

```
Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (main)
$ git checkout -b branch1
Switched to a new branch 'branch1'

Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (branch1)
$ vi file1

Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (branch1)
$ git add file1

Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (branch1)
$ git commit -m "file1 edited on branch1"
[branch1 486bf14] file1 edited on branch1
1 file changed, 2 insertions(+)
```

- Now merge the changes made in branch1 with the main branch.
- Then push the main and branch1 to the remote repository.



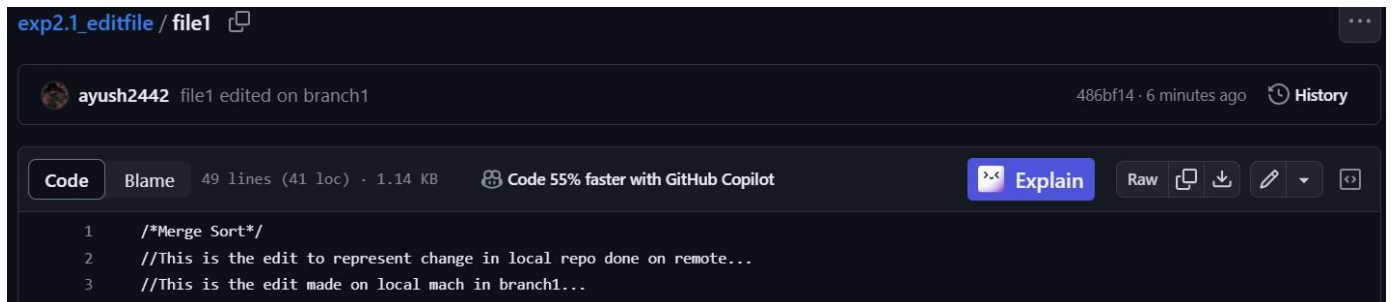
```
Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (branch1)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (main)
$ git merge branch1
Updating 8c9dc56..486bf14
Fast-forward
 file1 | 2 ++
 1 file changed, 2 insertions(+)

Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 367 bytes | 367.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ayush2442/exp2.1_editfile.git
 8c9dc56..486bf14  main -> main

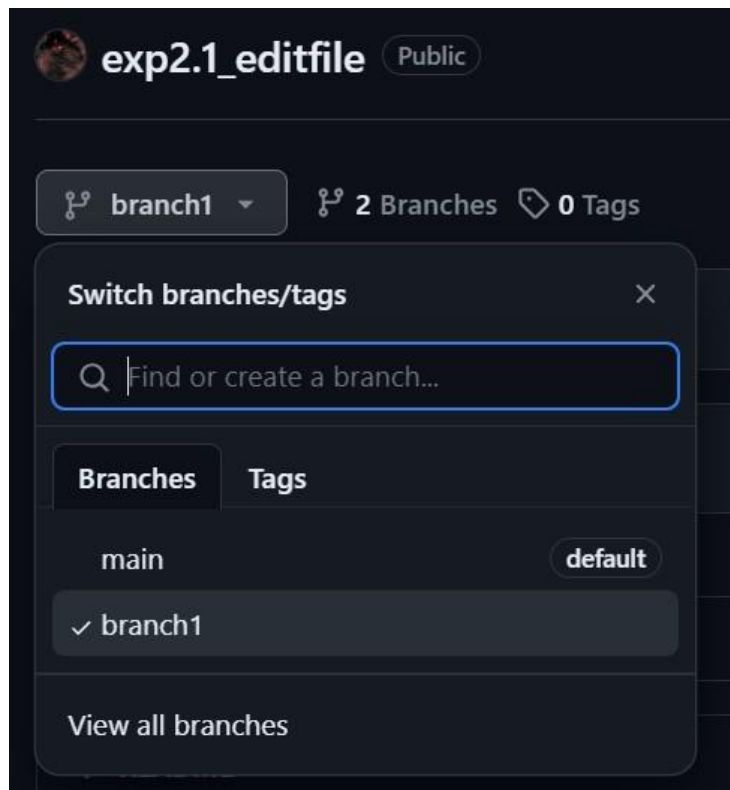
Ayush Pandey@Ayush MINGW64 ~/exp2.1_editfile (main)
$ git push origin branch1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting
remote:   https://github.com/ayush2442/exp2.1_editfile/pull/new
remote:   /branch1
remote:
To https://github.com/ayush2442/exp2.1_editfile.git
 * [new branch]      branch1 -> branch1
```

- Now you can see the changes in the remote repository.

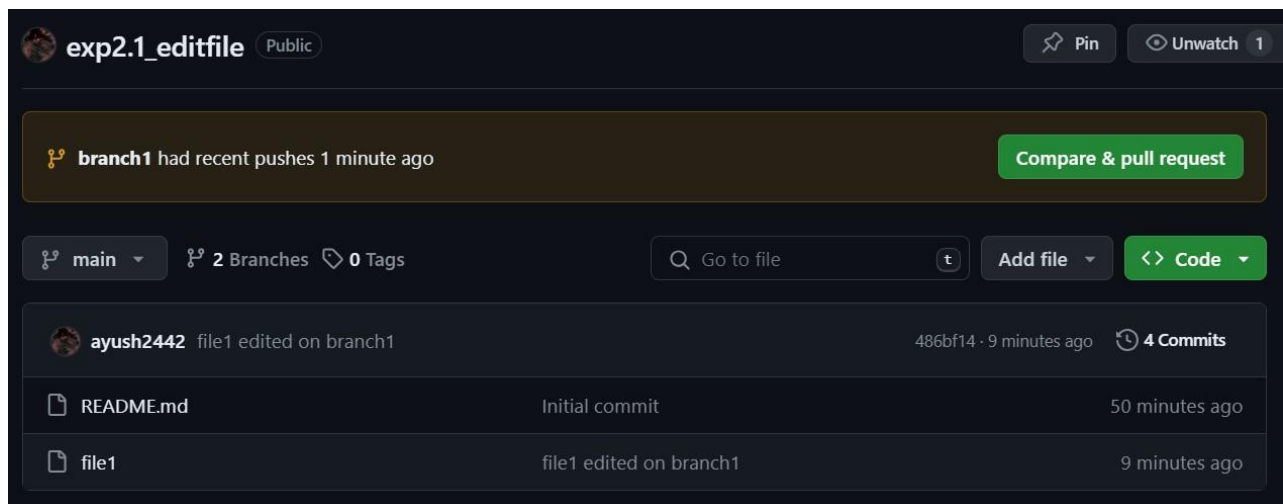


```
1 /*Merge Sort*/
2 //This is the edit to represent change in local repo done on remote...
3 //This is the edit made on local mach in branch1...
```

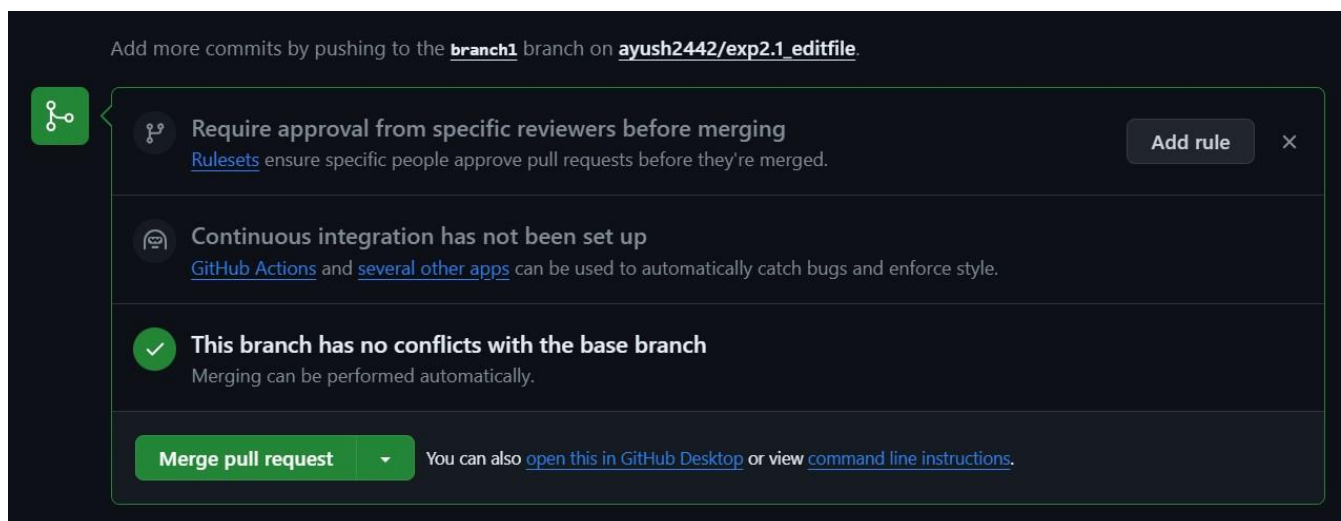
- Now go to GitHub and in the repository, go to 'branch1' and make some changes in the file.



- Now commit the changes and go to main branch.  
Compare & Pull request.



- Now create the pull request and after that click merge pull request.  
Do confirm merge.





#### 4. Result/Output/Writing Summary:

In this experiment, I have Created a Repository, created a file, editing the file and committed changes on Git(local) and GitHub(remote) machines.

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

1. I have learnt about branches and how to create them.
2. I have learnt about how to push and pull the changes.
3. I have learnt about how to merge the branches.
4. I have learnt about some new commands.
5. I have learnt how to resolve merge conflicts.

**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.			