Experiment -2.1

Student Name: Sushant Jha **UID: 22BDO10052**
Branch: CSE (Devops) **Section/Group: 22BCD-1/A**
Semester: 4th **Date of Performance: 06/02/24**
Subject Name: Git and Git-Hub **Subject Code: 22CSH-293**

1. **Aim/Overview of the practical:**

   Editing a file and committing changes on Git-Hub.

2. **Software Used:**

   Git Bash, Git-Hub.

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

1. Start by creating or cloning a repository on your local machine and then open GIT BASH.
2. Navigate to the repository directory using the cd command.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4 (main)
$ git init
Initialized empty Git repository in C:/Users/ASUS TUF/Desktop/gitandgitHUb/git4/
.git/

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4 (main)
$ git clone https://github.com/Sushantjha1236/new.git
Cloning into 'new'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 31 (delta 9), reused 2 (delta 0), pack-reused 0
Receiving objects: 100% (31/31), 24.22 KiB | 359.00 KiB/s, done.
Resolving deltas: 100% (9/9), done.
```

3. Create a new file or open an existing file in the main branch, for example, file.py, and add some code to it.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4 (main)
$ cd git4
bash: cd: git4: No such file or directory

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4 (main)
$ cd new

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4/new (main)
$ ls
LICENSE  README.md  fileA.txt  java.java  java2

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4/new (main)
$ git checkout -b br1
Switched to a new branch 'br1'
```

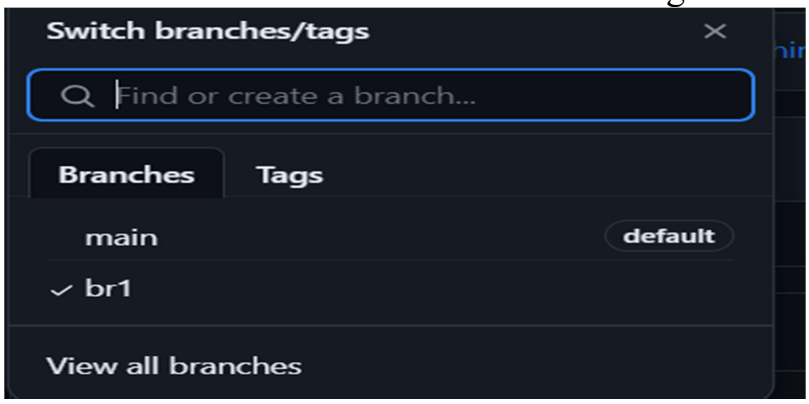
4. Add the files to the staging area using git add and then commit the changes using the git commit command.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4/new (br1)
$ vi java.java

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4/new (br1)
$ git add .

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4/new (br1)
$ git commit -m "branch commit"
[br1 5bcd6b8] branch commit
 1 file changed, 1 insertion(+), 1 deletion(-)
```

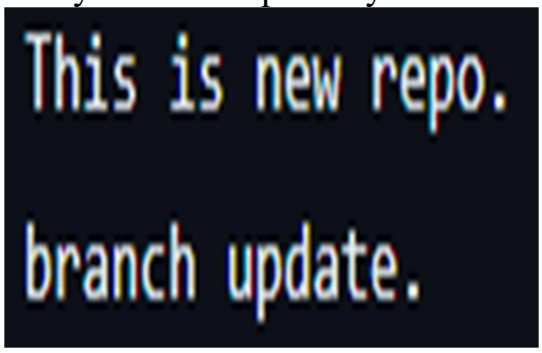
5. Push the changes to the remote repository using the command git push <remote_name> <branch_name>.

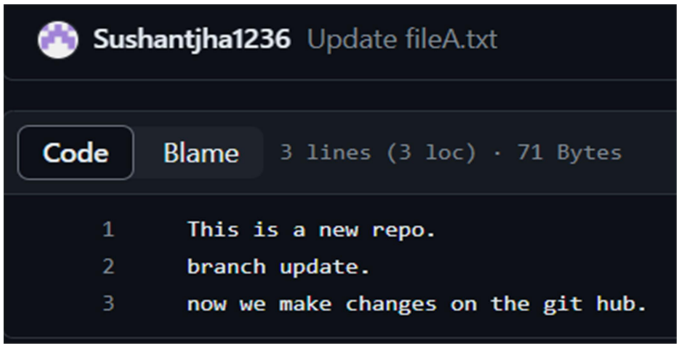
```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4/new (main)
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 325 bytes | 325.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Sushantjha1236/new.git
   358839a..5bcd6b8  main -> main
```

6. You should now be able to observe the changes in the remote repository.



7. Next, make some modifications to the file in the remote repository and pull those changes into your local repository.





8. Create a new branch and switch to it using the git checkout -b command, for instance, feature.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4/new (main)
$ git checkout -b br1
Switched to a new branch 'br1'
```

9. Edit the file using a text editor like vi and make some alterations.
10. Merge the changes made in the feature branch with the main branch and handle any conflicts that may arise manually.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4/new (br1)
$ vi java.java

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4/new (br1)
$ git add .

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4/new (br1)
$ git commit -m "branch commit"
[br1 5bcd6b8] branch commit
 1 file changed, 1 insertion(+), 1 deletion(-)
```

11. Push the main and feature branches to the remote repository.
12. You should see the new changes reflected in the remote repository.
13. Now, navigate to the repository on Git-Hub, switch to the feature branch, and modify a file.

14. Commit the changes and then switch back to the main branch. Click on the Compare & Pull request button.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4/new (main)
$ git commit -m "main commit"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
```

15. Create the pull request, address any merge conflicts that arise (if any), and then merge the pull request.
16. After the merge, you may choose to delete your feature branch.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git4/new (main)
$ git merge br1
Updating 358839a..5bcd6b8
Fast-forward
 java.java | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

17. Finally, pull the changes to your local repository using git pull.
18. You should now see the changes reflected in your local repository.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gita
$ cat fileA.txt
This is new repo.
branch update.
now we make change on git hub.
```

Learning outcomes (What I have learnt):

1. Basic Git Commands.
2. Branching and Merging.
3. Collaborative Workflow.
4. Remote Repository Interaction.
5. Conflict Resolution.

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.			