Experiment -1.2

Student Name: Anand Tiwari
UID: 22BDO10022
Branch: AIT-CSE(DevOps)
Section/Group: 22BCD-1/A
Semester: 4th
Date of Performance: 24/01/2024
Subject Name: Git and Hub
Subject Code: 22CSH-293

1. **Aim/Overview of the practical:** Creating branches with GitHub.

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

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

ON GITHUB →

1. Login to your github account and open or create a repository.
2. Click on the 'main' tab under the repository name as shown in image 1.

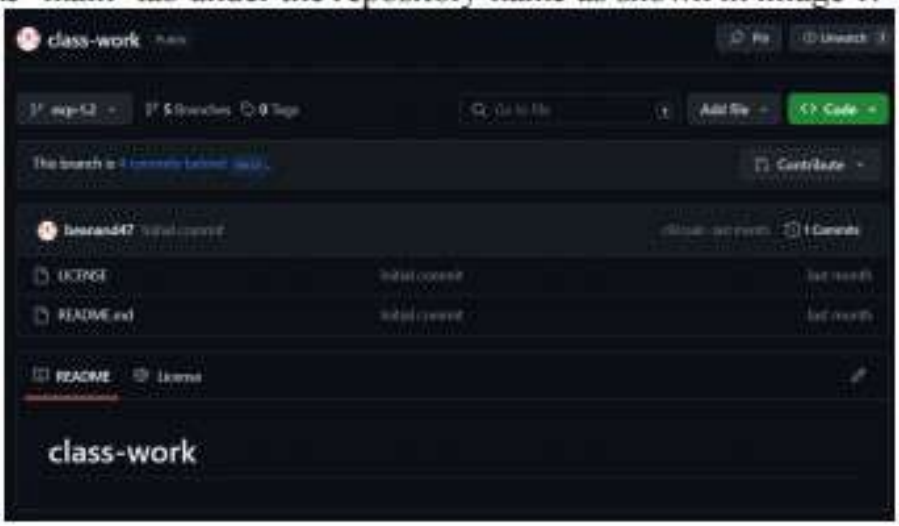

Image 1

3. Click on 'View all branches' and click on 'New branch' as shown in image 2.

Image 2

4. Type the name of the new branch and click on 'Create new branch' option as shown in image 3.

Image 3

5. In the main branch, I have created a java file with the name 'Exp1.2.java' and have the following code (image 4).

written

Image 4

6. Move to the new branch by clicking on the 'main' tab and selecting the new branch as shown in image 5 .

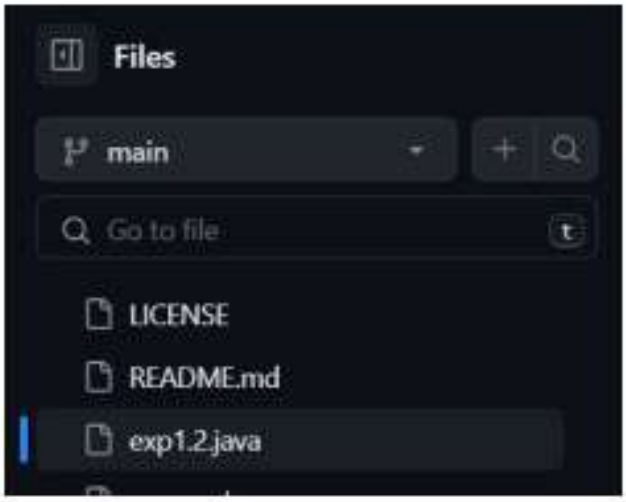

Image 5

7. Make changes in the java file in the new branch and commit the changes (image 6).

Image 6

8. Move to the main branch and click on the 'Compare & Pull request' option that you will be seeing on the top.

9. Add the title of the pull request and click on 'Create pull request' (image 8).

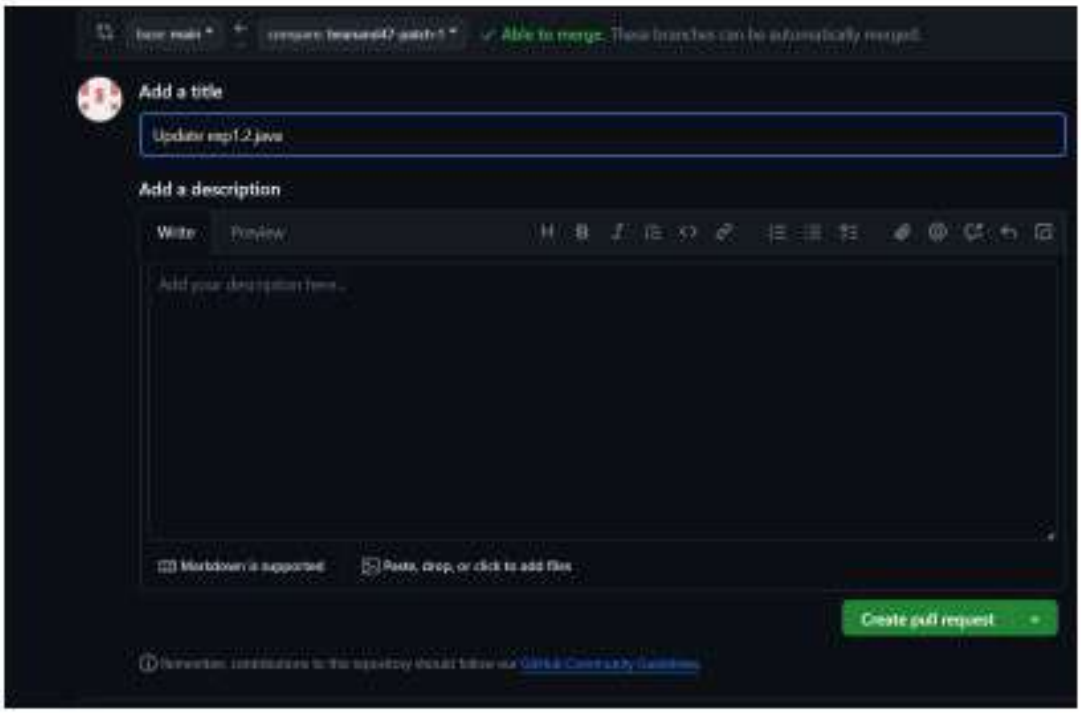

Image 8

10. Resolve the merge conflicts if necessary and click on the 'Merge pull request' option (image 9

Image 9

11. Your branch has been merged and you can now delete your branch if you want(image 10).

Image 10

ON GIT BASH →

1. Create a repository on your local machine and open 'Git bash' in the folder.
2. Initialize the repository using the ***git init*** command.
3. Create a new file in the repo using the ***vi*** command.
4. Add the file to the staged area using the ***git add <file_name>*** command.
5. Commit the changes and give a commit message using the ***git commit -m <commit_message>*** command.
6. Image 12 shows the steps from 2 to 5

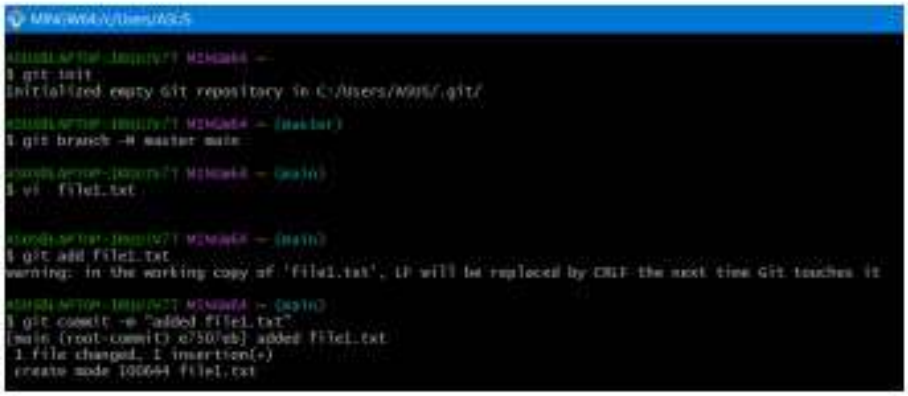

Image 12

7. Create a new branch using the ***git checkout -b <new_branch_name>*** command.
8. Open the ***file1.txt*** in the new branch and made changes in it.
9. Add the file to the staged area using ***git add*** command and commit the changes.
10. Image 13 shows the steps from 7 to 9 and Image 14 shows the text in ***file1.txt*** in the new branch

```
yashd@Tempestation MINGW64 /f/Git Practicals/Exp2 (main)
$ git checkout -b testBranch
Switched to a new branch 'testBranch'

yashd@Tempestation MINGW64 /f/Git Practicals/Exp2 (testBranch)
$ vi file1.txt

yashd@Tempestation MINGW64 /f/Git Practicals/Exp2 (testBranch)
$ git add file1.txt
warning: in the working copy of 'file1.txt', LF will be replac
ed by CRLF the next time Git touches it

yashd@Tempestation MINGW64 /f/Git Practicals/Exp2 (testBranch)
$ git commit -m "Made changes in file1.txt in testBranch"
[testBranch bc24969] Made changes in file1.txt in testBranch
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Image 14

```
MINGW64:/c/Users/ASUS
my name is anand tiwari
~
~
~
```

Image 15

11. Move to the main branch using the command ***git checkout main*** and merge the new branch with the main branch using command ***git merge <new_branch_name>*** as shown in image 16. In case of any merge conflicts, it will be either resolved automatically or you will have to resolve it manually.

```
yash@Tempestation MINGW64 /f/Git Practicals/Exp2 (testBranch)
$ git checkout main
Switched to branch 'main'

yash@Tempestation MINGW64 /f/Git Practicals/Exp2 (main)
$ git merge testBranch
Updating 2add134..bc24969
Fast-forward
 file1.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Image 16

12. Now, the changes made in the new branch have been merged with the main branch as shown in image 17.

Image 17

4. Result/Output/Writing Summary:

In this experiment, we have created a new branch, made some changes in the files in that new branch and then merged the changes with the main branch by resolving merge conflicts by using both GitHub and Git Bash.

Learning outcomes (What I have learnt):

1. Learnt how to create a branch.
2. Learnt how to create a pull request.
3. Learnt how to use vi editor on the git command line.
4. Learnt to merge two branches.
5. 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.			