



Experiment -1.3

To create and explore pull request.

Student Name: Sakshi

Branch: CSE(DevOps)

Semester: 4th

Subject Name: Git and GitHub

UID: 22BDO10064

Section/Group: 22BCD-1/B

Date of Performance: 02-02-24

Subject Code: 22CSH-293

1. Aim of the practical: To create and explore pull request.

2. Task to be done: In this experiment, we have to create and explore pull request in both GUI and CLI i.e., GitHub and Git Bash.

3. Steps for experiment/practical:

1. Firstly, we have to login to our GitHub account. After that we have to make one new repository on GitHub. After that, we will create the directory and then we will initialize it. By using the following command, we will initialize the git.


\$ git init

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 *

 Sakshi-code13 ▾

Repository name *

/ Exp3

✓ Exp3 is available.

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

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

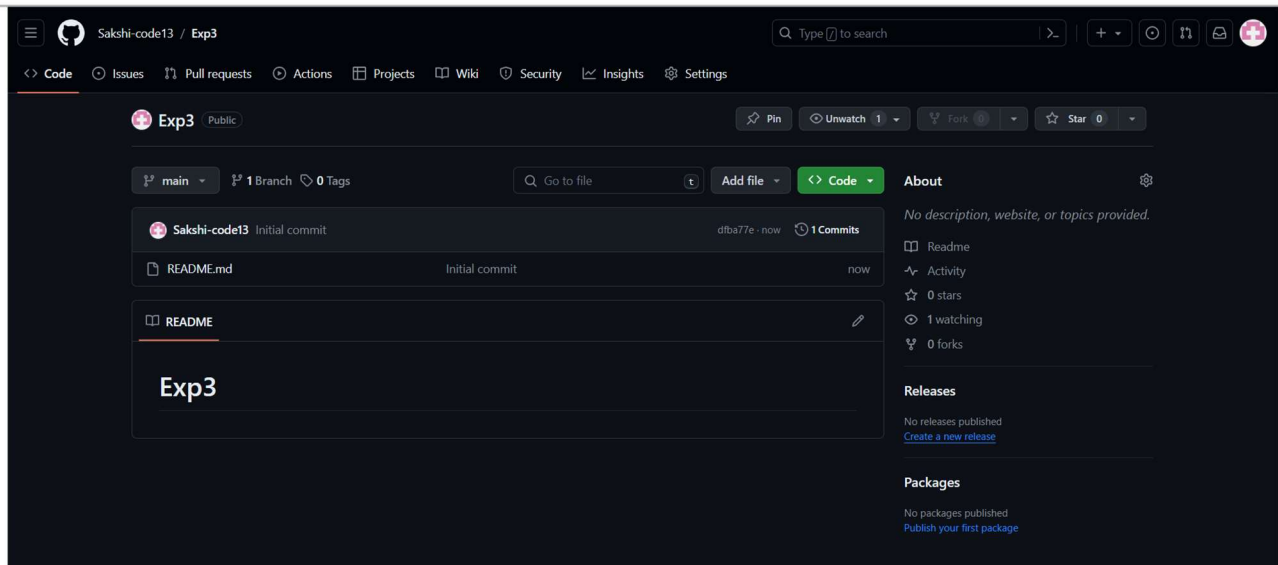
.gitignore template: None ▾

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

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)



```
MINGW64:/c/Users/ADMIN

ADMIN@LAPTOP-RFULERP MINGW64 ~ (master)
$ git init
Reinitialized existing Git repository in C:/Users/ADMIN/.git/
```

2. Then, we have to clone the repository by using the following command.

\$ git clone “url of your github repositories”

```
ADMIN@LAPTOP-RFULERP MINGW64 ~ (master)
$ git clone "https://github.com/Sakshi-code13/Exp3.git"
Cloning into 'Exp3'...
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.
```

3. Now, we will check the status by the following command: -

\$ git status

```
ADMIN@LAPTOP-RFULERP MINGW64 ~/Exp3 (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

4. We will create a file and edit it, by using the cat command: -

\$ vi File_name.txt as here we have used \$ vi Exp3.txt

```
ADMIN@LAPTOP-RFULERP MINGW64 ~/Exp3 (main)
$ vi Exp3.txt
```

5. Then, we will add the file and then commit it by using the following commands: -

\$ git add .

\$ git commit -m "Message"

```
ADMIN@LAPTOP-RFULERP MINGW64 ~/Exp3 (main)
$ git add .
warning: in the working copy of 'Exp3.txt', LF will be replaced by CRLF the next
time Git touches it
```

```
ADMIN@LAPTOP-RFULERP MINGW64 ~/Exp3 (main)
$ git commit -m"Committed Successfully"
[main c0028ce] Committed Successfully
1 file changed, 2 insertions(+)
create mode 100644 Exp3.txt
```

6. Now, we will create a new branch, by using the following command:-

\$ git checkout -b branch_name as here we have used \$ git checkout -b branch1

```
ADMIN@LAPTOP-RFULERP MINGW64 ~/Exp3 (main)
$ git checkout -b branch1
Switched to a new branch 'branch1'
```

7. In this, we have to add a file in this new branch and by using the following command we will achieve that:-

\$ vi File name.txt as here we have used \$ vi Exp3.txt

```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Exp3 (branch1)
$ vi Exp3.txt
```

8. Now, we will add and commit the file by using the following commands:-

\$ git add .

\$ git commit -m "Message you want to print"

```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Exp3 (branch1)
$ git add .
warning: in the working copy of 'Exp3.txt', LF will be replaced by CRLF the next
time Git touches it
```

```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Exp3 (branch1)
$ git commit -m "Successfully done"
[branch1 91a5c77] Successfully done
1 file changed, 1 insertion(+)
```

9. Once again we will switch the branch to the main by using the following command:-

\$ git checkout main

```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Exp3 (branch1)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
```

10. Now we will push main to the GitHub, by using the following command:-

\$ git push origin main

```
ADMIN@LAPTOP-RFULERP MINGW64 ~/Exp3 (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% (2/2), done.
Writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Sakshi-code13/Exp3.git
dfba77e..c0028ce main -> main
```

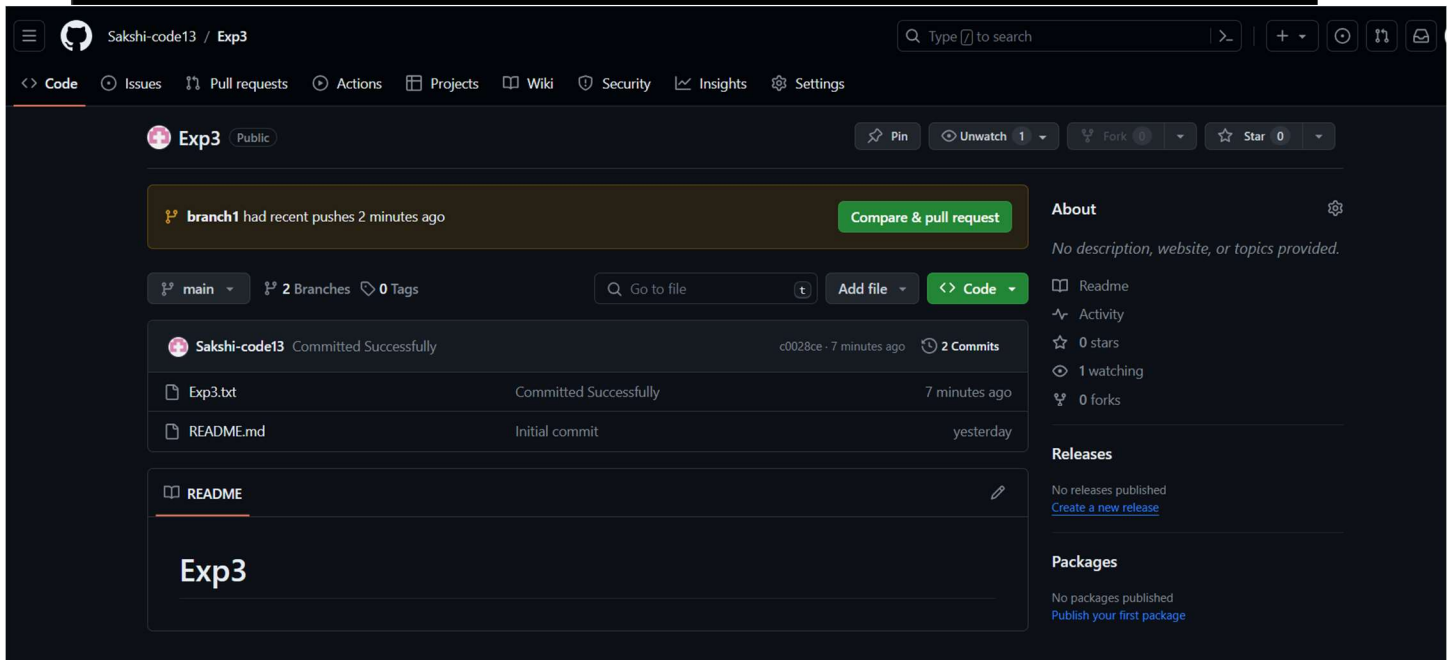
11. Then, we will checkout to the branch1 and push it too on the GitHub, by using the commands :-

\$ git checkout branch1

\$ git push origin branch1

```
ADMIN@LAPTOP-RFULERP MINGW64 ~/Exp3 (main)
$ git checkout branch1
Switched to branch 'branch1'

ADMIN@LAPTOP-RFULERP MINGW64 ~/Exp3 (branch1)
$ git push origin branch1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 311.00 KiB/s, done.
Total 3 (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/Sakshi-code13/Exp3/pull/new/branch1
remote:
To https://github.com/Sakshi-code13/Exp3.git
 * [new branch]      branch1 -> branch1
```

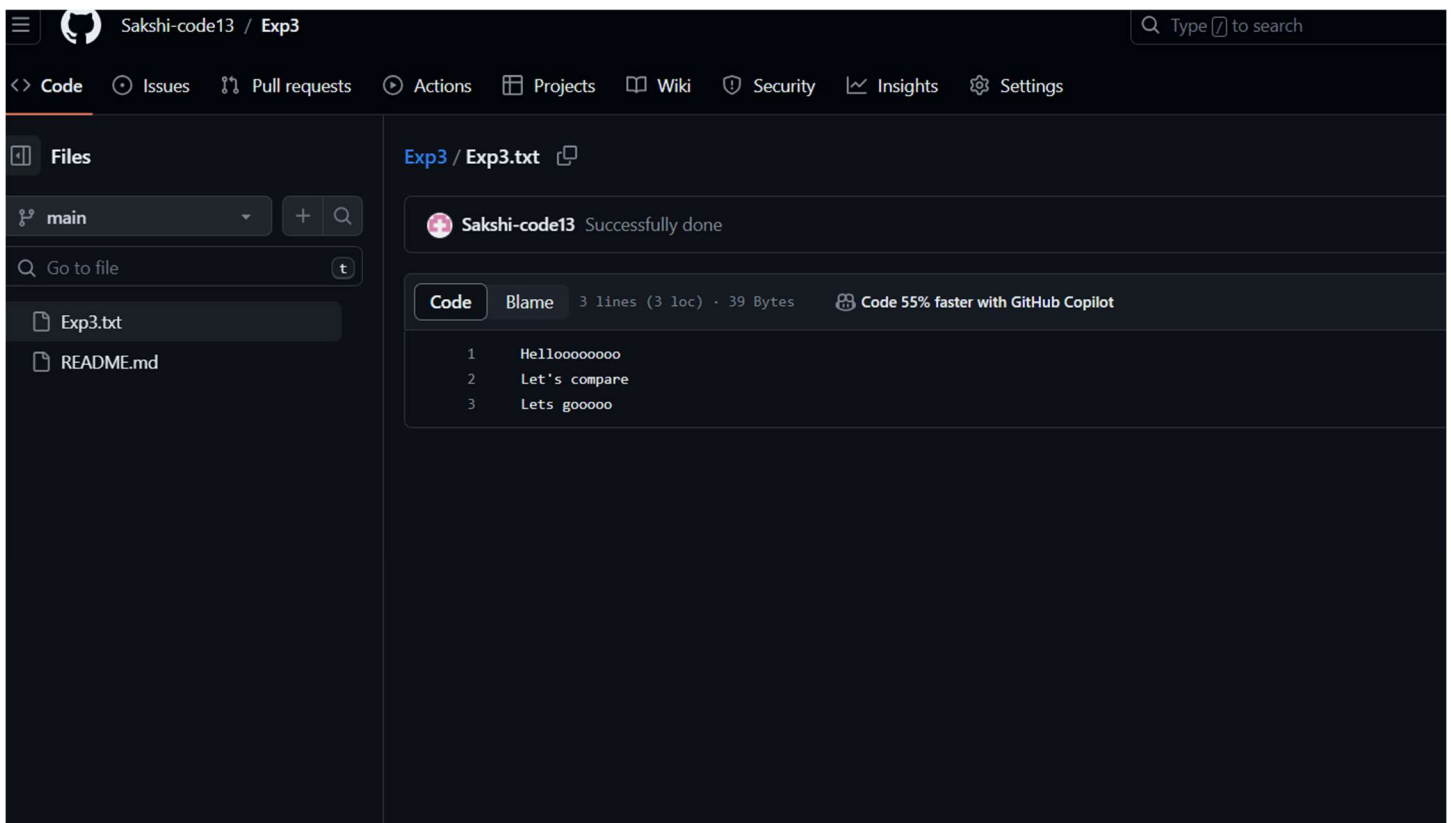


The screenshot shows a GitHub repository named 'Exp3' owned by 'Sakshi-code13'. The repository is public and has 2 branches (main and branch1) and 0 tags. A notification indicates that 'branch1' had recent pushes 2 minutes ago. The repository contains two files: 'Exp3.txt' (committed successfully 7 minutes ago) and 'README.md' (initial commit yesterday). The README file is open, showing the title 'Exp3'. The right sidebar shows the repository's statistics: 0 stars, 1 watching, and 0 forks. There are no releases or packages published.

12. Now, we can see the content of the file by using the following command, and also on GitHub:-

\$ cat Exp3.txt

```
ADMIN@LAPTOP-RFULERP MINGW64 ~/Exp3 (branch1)
$ cat Exp3.txt
Hellooooooooo
Let's compare
Lets gooooo
```



The screenshot shows the GitHub interface for the repository 'Sakshi-code13 / Exp3'. The 'Files' tab is selected, showing a list of files: 'Exp3.txt' and 'README.md'. The 'Exp3.txt' file is selected, and its content is displayed in the main area. The content of 'Exp3.txt' is:

```
1 Hellooooooooo
2 Let's compare
3 Lets gooooo
```

The interface also shows a search bar at the top right, a navigation bar with links to Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings, and a status bar at the bottom indicating 'Sakshi-code13 Successfully done'.



DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.



Commit changes

Commit message

Update Exp3.txt

Extended description

Add an optional extended description..

☒ Commit directly to the `main` branch

☐ Create a **new branch** for this commit and start a pull request
[Learn more about pull requests](#)

Cancel

Commit changes

13. Now, we will Pull and compare and after that merge it.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#). [Learn more about diff comparisons here](#).

base: main

compare: branch1

✓ Able to merge. These branches can be automatically merged.

Add a title

Successfully done

Add a description

Write

Preview

H B I

Add your description here...

Markdown is supported Paste, drop, or click to add files

Create pull request

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Use [Closing keywords](#) in the description to automatically close issues

Helpful resources

[GitHub Community Guidelines](#)

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).



DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.



Successfully done #1

[Open](#) Sakshi-code13 wants to merge 1 commit into `main` from `branch1`

Conversation 0

Commits 1

Checks 0

Files changed 1

+1 -0

Sakshi-code13 commented now

No description provided.

Successfully done

91a5c77

Add more commits by pushing to the `branch1` branch on `Sakshi-code13/Exp3`

Require approval from specific reviewers before merging

Rulesets ensure specific people approve pull requests before they're merged.

Add rule

Continuous integration has not been set up

GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Reviewers

No reviews

Still in progress? [Convert to draft](#)

Assignees

No one—[assign yourself](#)

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Sakshi-code13 / Exp3

Type to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Successfully done #1

[Merged](#) Sakshi-code13 merged 1 commit into `main` from `branch1` now

Conversation 0

Commits 1

Checks 0

Files changed 1

+1 -0

Sakshi-code13 commented now

No description provided.

Successfully done

91a5c77

Sakshi-code13 merged commit 7805f3a into `main` now

Revert

Pull request successfully merged and closed

You're all set—the `branch1` branch can be safely deleted.

Delete branch

Add a comment

egov

egov@cumail.in

14. Although after making changes to the file on the GitHub, we will use fetch command to fetch the updated details on the screen but it will show the older result.

```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Exp3 (branch1)
$ git fetch
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 1.80 KiB | 167.00 KiB/s, done.
From https://github.com/Sakshi-code13/Exp3
   c0028ce..5f675fe  main       -> origin/main

ADMIN@LAPTOP-RFULERMP MINGW64 ~/Exp3 (branch1)
$ cat Exp3.txt
Helloooooooooo
Let's compare
Lets goooooo
```

15. But, by using the command \$ git pull origin main we can see the updated result by using the command \$ cat Exp3.txt

```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Exp3 (branch1)
$ git pull origin main
From https://github.com/Sakshi-code13/Exp3
 * branch                main          -> FETCH_HEAD
Updating 91a5c77..5f675fe
Fast-forward
 Exp3.txt | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)
```

```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Exp3 (branch1)
$ cat Exp3.txt
Hello
Let's compare
Lets go...
```

16. To get the details about the differences we will use the command `$ git diff main`

```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Exp3 (branch1)
$ git diff main
diff --git a/Exp3.txt b/Exp3.txt
index 4ed5d28..3475e09 100644
--- a/Exp3.txt
+++ b/Exp3.txt
@@ -1,2 +1,3 @@
-Hellooooooooo
+Hello
  Let's compare
+Lets go...
```

17. At last we will checkout to the main branch and after that we will merge the files branch1 to the main branch and then we will get the no difference

```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Exp3 (branch1)
$ git checkout main
Switched to branch 'main'
Your branch is behind 'origin/main' by 3 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)
```

```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Exp3 (main)
$ git merge branch1
Updating c0028ce..5f675fe
Fast-forward
 Exp3.txt | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)
```

```
ADMIN@LAPTOP-RFULERMP MINGW64 ~/Exp3 (main)
$ git diff branch1
```




4. Result: In this experiment, we have created pull requests and explored with the help of Git Bash and GitHub. We initiated the creation of directory, branches, added specific files to it, added and committed it also pushed it to the GitHub, there we have compared it and subsequently merged it with the main branch.

Learning outcomes (What I have learnt):

1. Understanding Git Workflow
2. Understanding Collaborative Workflow
3. Version Control Proficiency
4. Committing changes
5. Working with staging area.

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.			