

## ASSIGNMENT 2

### SOURCE REPOSITORY

#### Exercise 1

##### Main Task 1

1. Create a new directory and change into it.

```
Git CMD

C:\Users\welcome>cd Downloads

C:\Users\welcome\Downloads>mkdir GitExercise

C:\Users\welcome\Downloads>cd GitExercise
```

2. Use the init command to create a Git repository in that directory.

```
C:\Users\welcome\Downloads\GitExercise>git init
Initialized empty Git repository in C:/Users/welcome/Downloads/GitExercise/.git/
```

3. Observe that there is now a .git directory.

| Downloads > GitExercise |      |                  |             |      |
|-------------------------|------|------------------|-------------|------|
|                         | Name | Date modified    | Type        | Size |
|                         | .git | 15-07-2024 21:21 | File folder |      |

4. Create a README file.

|        |                  |                    |      |
|--------|------------------|--------------------|------|
| .git   | 15-07-2024 21:21 | File folder        |      |
| README | 15-07-2024 21:23 | Markdown Source... | 0 KB |

5. Look at the output of the status command; the README you created should appear as an untracked file.

```
C:\Users\welcome\Downloads\GitExercise>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md

nothing added to commit but untracked files present (use "git add" to track)
```

6. Use the add command to add the new file to the staging area. Again, look at the output of the status command.

```
C:\Users\welcome\Downloads\GitExercise>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    README.md

nothing added to commit but untracked files present (use "git add" to track)
```

7. Now use the commit command to commit the contents of the staging area.



```
C:\Users\welcome\Downloads\GitExercise>git add .
C:\Users\welcome\Downloads\GitExercise>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
```

8. Create a src directory and add a couple of files to it.

```
C:\Users\welcome\Downloads\GitExercise>mkdir SRC
C:\Users\welcome\Downloads\GitExercise>cd SRC
```

|   |                  |               |      |
|---|------------------|---------------|------|
|  File1 | 15-07-2024 21:26 | Text Document | 0 KB |
|  File2 | 15-07-2024 21:26 | Text Document | 0 KB |

9. Use the add command, but name the directory, not the individual files. Use the status command. See how both files have been staged. Commit them.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git add .
C:\Users\welcome\Downloads\GitExercise\SRC>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   ../README.md
        new file:   File1.txt
        new file:   File2.txt

C:\Users\welcome\Downloads\GitExercise\SRC>git commit -m "Directory Commit"
[master (root-commit) 1a6898f] Directory Commit
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
 create mode 100644 SRC/File1.txt
 create mode 100644 SRC/File2.txt
```

10. Make a change to one of the files. Use the diff command to view the details of the change.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git diff
diff --git a/SRC/File2.txt b/SRC/File2.txt
index e69de29..d19bd92 100644
--- a/SRC/File2.txt
+++ b/SRC/File2.txt
@@ -0,0 +1 @@
+DharaniChandrasekaran
\ No newline at end of file
```

11. Next, add the changed file, and notice how it moves to the staging area in the status output. Also observe that the diff command you did before using add now gives no output. Why not? What do you have to do to see a diff of the things in the staging area? (Hint: review the slides if you can't remember.)

```
C:\Users\welcome\Downloads\GitExercise\SRC>git add .
C:\Users\welcome\Downloads\GitExercise\SRC>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   File2.txt
C:\Users\welcome\Downloads\GitExercise\SRC>git diff
```

12. Now – without committing – make another change to the same file you changed in step 10. Look at the status output, and the diff output. Notice how you can have both staged and unstaged changes, even when you're talking about a single file. Observe the difference when you use the add command to stage the latest round of changes. Finally, commit them. You should now have started to get a feel for the staging area.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   File2.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   File2.txt

C:\Users\welcome\Downloads\GitExercise\SRC>git diff
diff --git a/SRC/File2.txt b/SRC/File2.txt
index d19bd92..ed46537 100644
--- a/SRC/File2.txt
+++ b/SRC/File2.txt
@@ -1,1 @@
-DharaniChandrasekaran
\ No newline at end of file
+Dharani C V
\ No newline at end of file

C:\Users\welcome\Downloads\GitExercise\SRC>git add .
C:\Users\welcome\Downloads\GitExercise\SRC>git commit -m "Change Commit"
[master 9c04eb8] Change Commit
1 file changed, 1 insertion(+)
```

13. Use the log command in order to see all of the commits you made so far.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git log
commit 9c04eb8df0dca39a957e330b737838f4937b42b1 (HEAD -> master)
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date:   Mon Jul 15 21:32:53 2024 +0530

    Change Commit

commit 1a6898ff038ba942c317877d761d1510bc18cc35
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date:   Mon Jul 15 21:29:13 2024 +0530

    Directory Commit
```

14. Use the show command to look at an individual commit. How many characters of the commit identifier can you get away with typing at a minimum?

```

C:\Users\welcome\Downloads\GitExercise\SRC>git show
commit 9c04eb8df0dca39a957e330b737838f4937b42b1 (HEAD -> master)
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date:   Mon Jul 15 21:32:53 2024 +0530

    Change Commit

diff --git a/SRC/File2.txt b/SRC/File2.txt
index e69de29..ed46537 100644
--- a/SRC/File2.txt
+++ b/SRC/File2.txt
@@ -0,0 +1 @@
+Dharani C V
\ No newline at end of file

```

15. Make a couple more commits, at least one of which should add an extra file.

```

C:\Users\welcome\Downloads\GitExercise\SRC>git add .

C:\Users\welcome\Downloads\GitExercise\SRC>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   File1.txt

C:\Users\welcome\Downloads\GitExercise\SRC>git commit -m "Final Commit"
[master 5929004] Final Commit
 1 file changed, 1 insertion(+)

C:\Users\welcome\Downloads\GitExercise\SRC>git add .

C:\Users\welcome\Downloads\GitExercise\SRC>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   File1.txt

C:\Users\welcome\Downloads\GitExercise\SRC>git commit -m "Finally Committing"
[master f573940] Finally Committing
 1 file changed, 1 insertion(+), 1 deletion(-)

```

### Stretch Task -1

1. Use the Git rm command to remove a file. Look at the status afterwards. Now commit the deletion.

```

C:\Users\welcome\Downloads\GitExercise>git mv README.md Readme.txt

C:\Users\welcome\Downloads\GitExercise>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:   README.md -> Readme.txt

C:\Users\welcome\Downloads\GitExercise>git commit -m "Readme Commit"
[master 11d4e78] Readme Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename README.md => Readme.txt (100%)

```

2. Delete another file, but this time do not use Git to do it; e.g. if you are on Linux, just use the normal (non-Git) rm command; on Windows use del.

```

C:\> Command Prompt
Microsoft Windows [Version 10.0.19045.4651]
(c) Microsoft Corporation. All rights reserved.

C:\Users\welcome>cd Downloads
C:\Users\welcome\Downloads>cd GitExercise
C:\Users\welcome\Downloads\GitExercise>cd SRC
C:\Users\welcome\Downloads\GitExercise\SRC>del File1.txt

```

3. Look at the status. Compare it to the status output you had after using the Git built-in rm command. Is anything different? After this, commit the deletion.

```

C:\Users\welcome\Downloads\GitExercise>git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    SRC/File1.txt

no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\welcome\Downloads\GitExercise>git add .
C:\Users\welcome\Downloads\GitExercise>git commit -m "Delete Commit"
[master 05fa9d6] Delete Commit
 1 file changed, 1 deletion(-)
 delete mode 100644 SRC/File1.txt

```

4. Use the Git mv command to move or rename a file; for example, rename README to README.txt. Look at the status. Commit the change.

```

C:\Users\welcome\Downloads\GitExercise\SRC>git mv File2.txt Files.c
C:\Users\welcome\Downloads\GitExercise\SRC>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        renamed:   File2.txt -> Files.c

C:\Users\welcome\Downloads\GitExercise\SRC>git add .
C:\Users\welcome\Downloads\GitExercise\SRC>git commit -m "Rename Commit"
[master 6f9e656] Rename Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename SRC/{File2.txt => Files.c} (100%)

```

5. Now do another rename, but this time using the operating system's command to do so. How does the status look? Will you get the right outcome if you were to commit at this point? (Answer: almost certainly not, so don't. 9) Work out how to get the status to show that it will not lose the file, and then commit. Did Git at any point work out that you had done a rename?

```

C:\Users\welcome\Downloads\GitExercise>ren README.txt Readme.md

```

```

C:\Users\welcome\Downloads\GitExercise>git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    Readme.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Readme.md

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\welcome\Downloads\GitExercise>git add .

C:\Users\welcome\Downloads\GitExercise>git commit -m "Rename OS Correct Commit"
[master 626aa05] Rename OS Correct Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename Readme.txt => Readme.md (100%)

```

6. Use git help log to find out how to get Git to display just the most recent 3 commits. Try it.

```

C:\Users\welcome\Downloads\GitExercise>git help log

C:\Users\welcome\Downloads\GitExercise>git show -3
commit 626aa05ffaae68c914b00764ec7f2c679943eef1 (HEAD -> master)
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date:   Mon Jul 15 21:53:37 2024 +0530

    Rename OS Correct Commit

diff --git a/Readme.txt b/Readme.md
similarity index 100%
rename from Readme.txt
rename to Readme.md

commit 6f9e656d77c61f09dbd21c030a1b713be96a4430
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date:   Mon Jul 15 21:48:36 2024 +0530

    Rename Commit

diff --git a/SRC/File2.txt b/SRC/Files.c
similarity index 100%
rename from SRC/File2.txt
rename to SRC/Files.c

commit 05fa9d6b7bddadcf5f7d15f2035f5906d6fe3ede
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date:   Mon Jul 15 21:43:38 2024 +0530

    Delete Commit

diff --git a/SRC/File1.txt b/SRC/File1.txt
deleted file mode 100644
index 63eb16a..0000000
--- a/SRC/File1.txt
+++ /dev/null
@@ -1,0,0 @@
-Vijayasri C V
\ No newline at end of file

```

7. If you don't remember, look back in the slides to see what the --stat option did on the diff command. Find out if this also works with the show command. How about the log command?



```
C:\Users\welcome\Downloads\GitExercise>git diff --stat

C:\Users\welcome\Downloads\GitExercise>git show --stat
commit 626aa05ffaae68c914b00764ec7f2c679943eef1 (HEAD -> master)
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:53:37 2024 +0530

Rename OS Correct Commit

Readme.txt => Readme.md | 0
1 file changed, 0 insertions(+), 0 deletions(-)
```

```
C:\Users\welcome\Downloads\GitExercise>git log
commit 626aa05ffaae68c914b00764ec7f2c679943eef1 (HEAD -> master)
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:53:37 2024 +0530

Rename OS Correct Commit

commit 6f9e656d77c61f09dbd21c030a1b713be96a4430
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:48:36 2024 +0530

Rename Commit

commit 05fa9d6b7bddadcf5f7d15f2035f5906d6fe3ede
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:43:38 2024 +0530

Delete Commit

commit 11d4e78c79e5f665891b2f2a05d440021a14b8ef
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:40:31 2024 +0530

Readme Commit

commit f5739406f335d48d83f30e4fb793c185d3af0dd0
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:36:55 2024 +0530

Finally Committing

commit 5929004ccf2cdfef83164b7841899120d9dc841a2
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:35:38 2024 +0530

Final Commit

commit 9c04eb8df0dca39a957e330b737838f4937b42b1
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:32:53 2024 +0530

Change Commit

commit 1a6898ff038ba942c317877d761d1510bc18cc35
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:29:13 2024 +0530

Directory Commit
```

8. Imagine you want to see a diff that summarizes all that happened between two commit identifiers. Use the diff command, specifying two commit identifiers joined by two dots (that is, something like abc123..def456). Check the output is what you expect.

```
C:\Users\welcome\Downloads\GitExercise>git diff 1a6898f..9c04eb8
diff --git a/SRC/File2.txt b/SRC/File2.txt
index e69de29..ed46537 100644
--- a/SRC/File2.txt
+++ b/SRC/File2.txt
@@ -0,0 +1 @@
+Dharani C V
\ No newline at end of file
```

## Exercise 2

### Main Task

1. Run the status command. Notice how it tells you what branch you are in.

```
C:\Users\welcome\Downloads\GitExercise>git status
On branch master
nothing to commit, working tree clean
```

2. Use the branch command to create a new branch.

```
C:\Users\welcome\Downloads\GitExercise>git branch Dharani

C:\Users\welcome\Downloads\GitExercise>git branch
  Dharani
* master
```

3. Use the checkout command to switch to it.

```
C:\Users\welcome\Downloads\GitExercise>git checkout Dharani
Switched to branch 'Dharani'
```

4. Make a couple of commits in the branch – perhaps adding a new file and/or editing existing ones.

```
C:\Users\welcome\Downloads\GitExercise>git add .

C:\Users\welcome\Downloads\GitExercise>git commit -m "New Branch Commit"
[Dharani 8fe6f9c] New Branch Commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 SRC/File1.java
 create mode 100644 SRC/File2.py

C:\Users\welcome\Downloads\GitExercise>git add .

C:\Users\welcome\Downloads\GitExercise>git commit -m "Branch Change Commit"
[Dharani 21504b5] Branch Change Commit
 1 file changed, 1 insertion(+)
```

5. Use the log command to see the latest commits. The two you just made should be at the top of the list.

```
C:\Users\welcome\Downloads\GitExercise>git log
commit 21504b5f24ece8e3ddc6d67d1770258801c1dd8c (HEAD -> Dharani)
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 22:06:41 2024 +0530

    Branch Change Commit

commit 8fe6f9cb88da46bea8db47bfd59389f3fab24a1e
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 22:05:41 2024 +0530

    New Branch Commit

commit 626aa05ffa6e68c914b00764ec7f2c679943eef1 (master)
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:53:37 2024 +0530

    Rename OS Correct Commit
```



6. Use the checkout command to switch back to the master branch. Run log again. Notice your commits don't show up now. Check the files also – they should have their original contents.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git log
commit 626aa05ffaae68c914b00764ec7f2c679943eef1 (HEAD -> master)
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:53:37 2024 +0530

    Rename OS Correct Commit

commit 6f9e656d77c61f09dbd21c030a1b713be96a4430
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:48:36 2024 +0530

    Rename Commit

commit 05fa9d6b7bddadcf5f7d15f2035f5906d6fe3ede
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:43:38 2024 +0530

    Delete Commit

commit 11d4e78c79e5f665891b2f2a05d440021a14b8ef
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:40:31 2024 +0530

    Readme Commit

commit f5739406f335d48d83f30e4fb793c185d3af0dd0
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:36:55 2024 +0530

    Finally Committing

commit 5929004ccf2cdfe83164b7841899120d9dc841a2
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:35:38 2024 +0530

    Final Commit

commit 9c04eb8df0dca39a957e330b737838f4937b42b1
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:32:53 2024 +0530
```

7. Use the checkout command to switch back to your branch. Use gitk to take a look at the commit graph; notice it's linear.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git checkout Dharani
Switched to branch 'Dharani'

C:\Users\welcome\Downloads\GitExercise\SRC>gitk
```

GitExercise: All files - gitk

File Edit View Help

● Dharani Branch Change Commit  
● master Rename OS Correct Commit  
● Dharani Rename Commit  
● Dharani Delete Commit  
● Dharani Readme Commit  
● Dharani Finally Committing  
● Dharani Final Commit  
● Dharani Directory Commit

| Commit Hash                              | Author  | Date                |
|--|---|---------------------|
| 626aa05ffaae68c914b00764ec7f2c679943eef1 | DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> | 2024-07-15 22:06:41 |
| 6f9e656d77c61f09dbd21c030a1b713be96a4430 | DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> | 2024-07-15 21:53:37 |
| 05fa9d6b7bddadcf5f7d15f2035f5906d6fe3ede | DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> | 2024-07-15 21:48:36 |
| 11d4e78c79e5f665891b2f2a05d440021a14b8ef | DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> | 2024-07-15 21:43:38 |
| f5739406f335d48d83f30e4fb793c185d3af0dd0 | DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> | 2024-07-15 21:40:31 |
| 5929004ccf2cdfe83164b7841899120d9dc841a2 | DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> | 2024-07-15 21:36:55 |
| 5929004ccf2cdfe83164b7841899120d9dc841a2 | DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> | 2024-07-15 21:35:38 |
| 9c04eb8df0dca39a957e330b737838f4937b42b1 | DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> | 2024-07-15 21:32:53 |

SHA1 ID: 21504b5f24e0e8e3ddc6d67d1770258801c1dd8c | Row 1 / 10

Find commit containing: | Exact | All fields

Search | Patch | Tree

Diff | Old version | New version | Lines of context: 3 | Ignore space change | 1

Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 22:06:41  
Committer: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 22:06:41  
Parent: 8f6f5cb88da4f6ba8db47bf959389f3fab24a1e (New Branch Commit)  
Branch: Dharani  
Follows:  
Precedes:

Branch Change Commit

----- SRC/File2.py -----  
index e69de29..eb87f35 100644  
@@ -0,0 +1 @@  
+it is Python file.  
+ No newline at end of file

8. Now checkout the master branch again. Use the merge command to merge your branch in to it. Look for information about it having been a fast-forward merge. Look at git log, and see that there is no merge commit. Take a look in gitk and see how the DAG is linear.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git checkout master
Switched to branch 'master'

C:\Users\welcome\Downloads\GitExercise\SRC>git merge Dharani
Updating 626aa05..21504b5
Fast-forward
 SRC/File1.java | 0
 SRC/File2.py   | 1 +
 2 files changed, 1 insertion(+)
 create mode 100644 SRC/File1.java
 create mode 100644 SRC/File2.py

C:\Users\welcome\Downloads\GitExercise\SRC>git log
commit 21504b5f24ece8e3ddc6d67d1770258801c1dd8c (HEAD -> master, Dharani)
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 22:06:41 2024 +0530

    Branch Change Commit

commit 8fe6f9cb88da46bea8db47bfd59389f3fab24a1e
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 22:05:41 2024 +0530

    New Branch Commit

commit 626aa05ffaae68c914b00764ec7f2c679943eef1
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:53:37 2024 +0530

    Rename OS Correct Commit

commit 6f9e656d77c61f09dbd21c030a1b713be96a4430
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:48:36 2024 +0530

    Rename Commit

commit 05fa9d6b7bddadc5f57d15f2035f5906d6fe3ede
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:43:38 2024 +0530

    Delete Commit
```

The screenshot shows the gitk graphical interface. At the top, the command prompt shows the current directory is C:\Users\welcome\Downloads\GitExercise\SRC. The main window displays a list of commits on the left, with the 'Branch Change Commit' selected. The right pane shows a detailed view of the selected commit, including the author, date, and a list of files. Below this, the 'Diff' view shows the changes in SRC/File2.py, highlighting the addition of a new line of code. The commit history is linear, showing a sequence of commits from 'Delete Commit' to 'Branch Change Commit'.

9. Switch back to your branch. Make a couple more commits.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git checkout Dharani
Switched to branch 'Dharani'

C:\Users\welcome\Downloads\GitExercise\SRC>git add .

C:\Users\welcome\Downloads\GitExercise\SRC>git commit -m "Python Commit"
[Dharani ab5f094] Python Commit
1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\welcome\Downloads\GitExercise\SRC>git add .

C:\Users\welcome\Downloads\GitExercise\SRC>git commit -m "Java Commit"
[Dharani c929770] Java Commit
1 file changed, 1 insertion(+)
```

10. Switch back to master. Make a commit there, which should edit a different file from the ones you touched in your branch – to be sure there is no conflict.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git add .

C:\Users\welcome\Downloads\GitExercise\SRC>git commit -m "C Commit"
[master eec7d68] C Commit
1 file changed, 1 insertion(+), 1 deletion(-)
```

11. Now merge your branch again. (Aside: you don't need to do anything to inform Git that you only want to merge things added since your previous merge. Due to the way Git works, that kind of issue simply does not come up, unlike in early versions of Subversion.)

```
C:\Users\welcome\Downloads\GitExercise\SRC>git merge Dharani
Merge made by the 'ort' strategy.
 SRC/File1.java | 1 +
 SRC/File2.py   | 2 +-
 2 files changed, 2 insertions(+), 1 deletion(-)
```

12. Look at git log. Notice that there is a merge commit. Also look in gitk. Notice the DAG now shows how things forked, and then were joined up again by a merge commit.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git log
commit 7468184b8c384f46a921a425ff26d137180a1fc4 (HEAD -> master)
Merge: eec7d68 c929770
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 22:19:19 2024 +0530

Merge branch 'Dharani'
```

C:\Users\welcome\Downloads\GitExercise\SRC>gitk

GitExercise: All files - gitk

File Edit View Help

master Merge branch 'Dharani' DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 22:19:19

DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 22:16:19

DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 22:15:44

DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 22:18:13

DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 22:06:41

DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 22:05:41

DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 21:53:37

DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 21:48:36

DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 21:43:38

DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 21:40:31

DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 21:36:55

SHA1 ID: 7468184b8c384f46a921a425ff26d137180a1fc4 Row 1 / 14

Find commit containing: Exact All fields

Search

Diff Old version New version Lines of context: 3 Ignore space change

Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 22:19:19

Committer: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-15 22:19:19

Parent: eec7d68c60da575d9a013fbc9d897b310b9a3b (C Commit)

Parent: c9297705b471a8519b6a94db7b6cc661b62842e (Java Commit)

Branch: master

Follows:

Precedes:

Merge branch 'Dharani'

## Stretch Task

1. Once again, checkout your branch. Make a couple of commits.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git checkout Dharani
Switched to branch 'Dharani'

C:\Users\welcome\Downloads\GitExercise\SRC>git add .

C:\Users\welcome\Downloads\GitExercise\SRC>git commit -m "CSS Commit"
[Dharani 64d741a] CSS Commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 SRC/File3.css

C:\Users\welcome\Downloads\GitExercise\SRC>git add .

C:\Users\welcome\Downloads\GitExercise\SRC>git commit -m "Python Re-Commit"
[Dharani 403fd4e] Python Re-Commit
1 file changed, 1 insertion(+), 1 deletion(-)
```

2. Return to your master branch. Make a commit there that changes the exact same line, or lines, as commits in your branch did.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git checkout master
Switched to branch 'master'

C:\Users\welcome\Downloads\GitExercise\SRC>git add .

C:\Users\welcome\Downloads\GitExercise\SRC>git commit -m "Exact Commit"
[master caaa0df] Exact Commit
2 files changed, 1 insertion(+), 1 deletion(-)
create mode 100644 SRC/File3.css
```

3. Now try to merge your branch. You should get a conflict.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git checkout master
Switched to branch 'master'

C:\Users\welcome\Downloads\GitExercise\SRC>git add .

C:\Users\welcome\Downloads\GitExercise\SRC>git commit -m "Exact Commit"
[master caaa0df] Exact Commit
2 files changed, 1 insertion(+), 1 deletion(-)
create mode 100644 SRC/File3.css
```

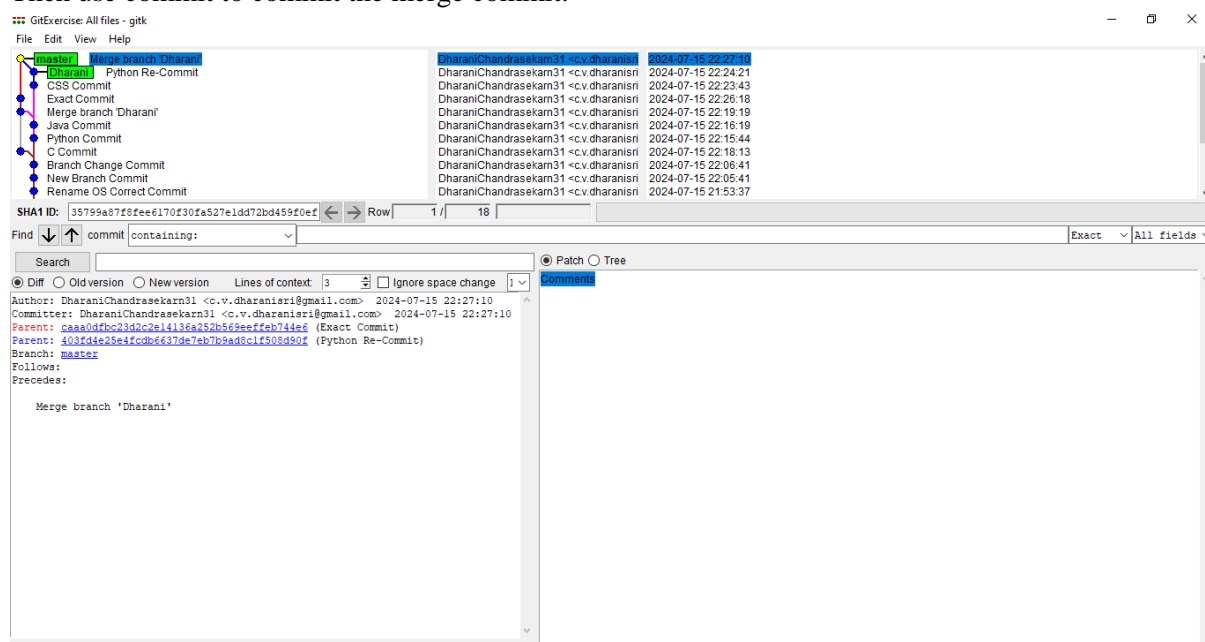
4. Open the file(s) that is in conflict. Search for the conflict marker. Edit the file to remove the conflict markers and resolve the conflict.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git merge Dharani
Merge made by the 'ort' strategy.
```

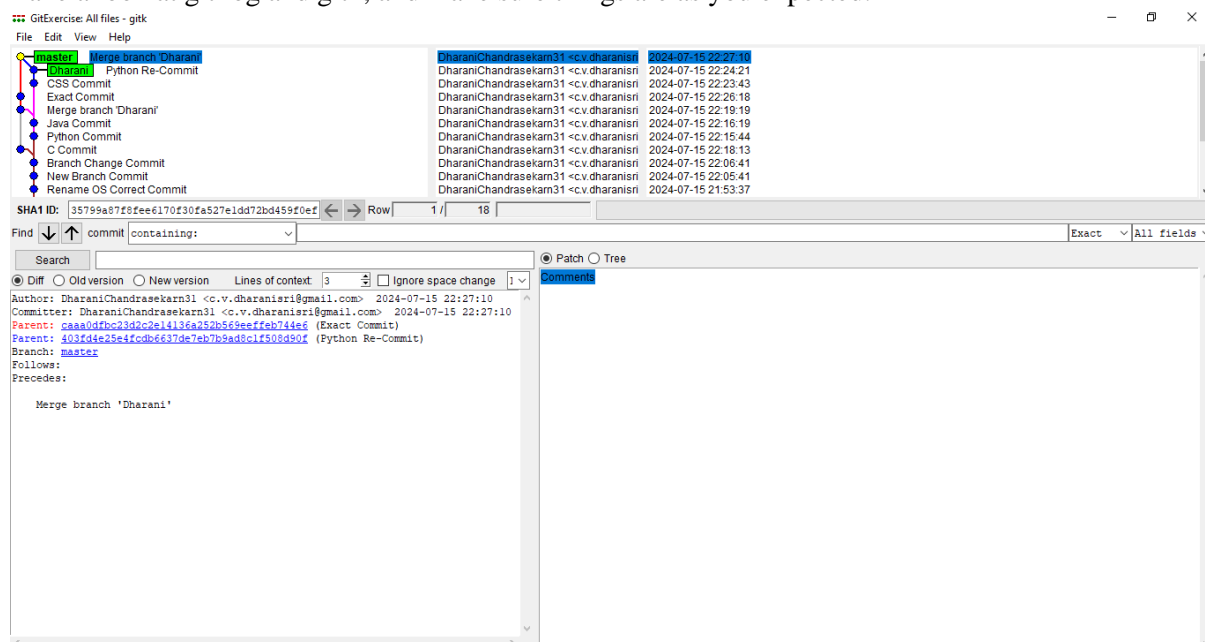
5. Now try to commit. Notice that Git will not allow you to do this when you still have potentially unresolved conflicts. Look at the output of status too.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git status
On branch master
nothing to commit, working tree clean
```

- Use the add command to add the files that you have resolved conflicts in to the staging area. Then use commit to commit the merge commit.



- Take a look at git log and gitk, and make sure things are as you expected.



- If time allows, you may wish to...

- Delete everything but your .git directory, then do a checkout command, to prove to yourself that this really will restore all of your current working copy.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git rm -r --cached --ignore-unmatch !(.git/*)
```

- Create a situation where one branch has changed a file, but the other branch has deleted it. What happens when you try to merge? How will you resolve it?

```
C:\Users\welcome\Downloads\GitExercise\SRC>git ls-tree Dharani

C:\Users\welcome\Downloads\GitExercise\SRC>git ls-tree master
100644 blob cc4a86518a0c8bbf3f00ce8dfd1d24f45ab154b5 File1.java
100644 blob 35e04991ab85fd56a40dc5dff0207226ee692420 File2.py
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391 File3.css
100644 blob efb9d86ff8d1ba856e6064504947f6482551188a Files.c
```



- Look at the help page for merge, and find out how you specify a custom message for the merge commit if it is automatically generated.

```
C:\Users\welcome\Downloads\GitExercise\SRC>git ls-tree Dharani

C:\Users\welcome\Downloads\GitExercise\SRC>git ls-tree master
100644 blob cc4a86518a0c8bbf3f00ce8dfd1d24f45ab154b5 File1.java
100644 blob 35e04991ab85fd56a40dc5dff0207226ee692420 File2.py
100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391 File3.css
100644 blob efb9d86ff8d1ba856e6064504947f6482551188a Files.c
```

- Look at the help page for merge, and find out how to prevent Git from automatically committing the merge commit it generates, but instead give you chance to inspect it and merge it yourself.

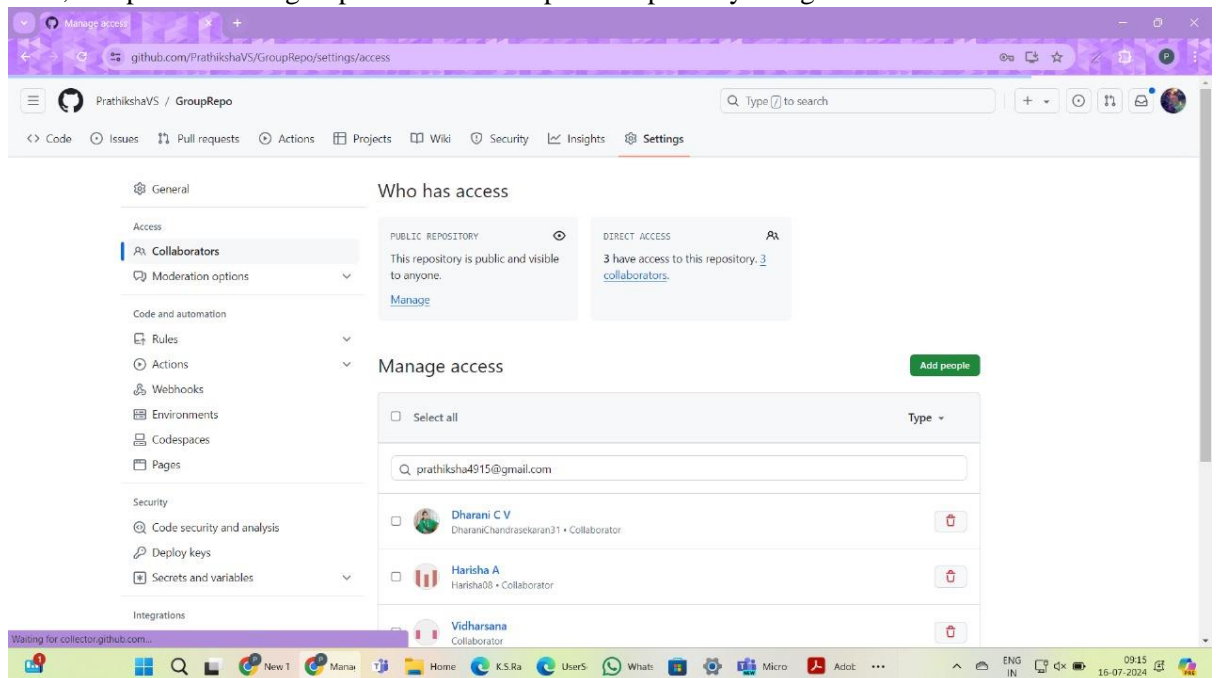
```
C:\Users\welcome\Downloads\GitExercise\SRC>git merge master
CONFLICT (modify/delete): SRC/Files.c deleted in HEAD and modified in master.
Version master of SRC/Files.c left in tree.
Automatic merge failed; fix conflicts and then commit the result.
```

### Exercise 3

For this task, you will work in a small group. Between 2 and 4 people is about right.

#### Main Task

1. First, one person in the group should create a public repository using their GitHub account.



2. This same person should then follow the instructions from GitHub to add a remote, and then push their repository. Do not forget the `-u` flag, as suggested by GitHub!

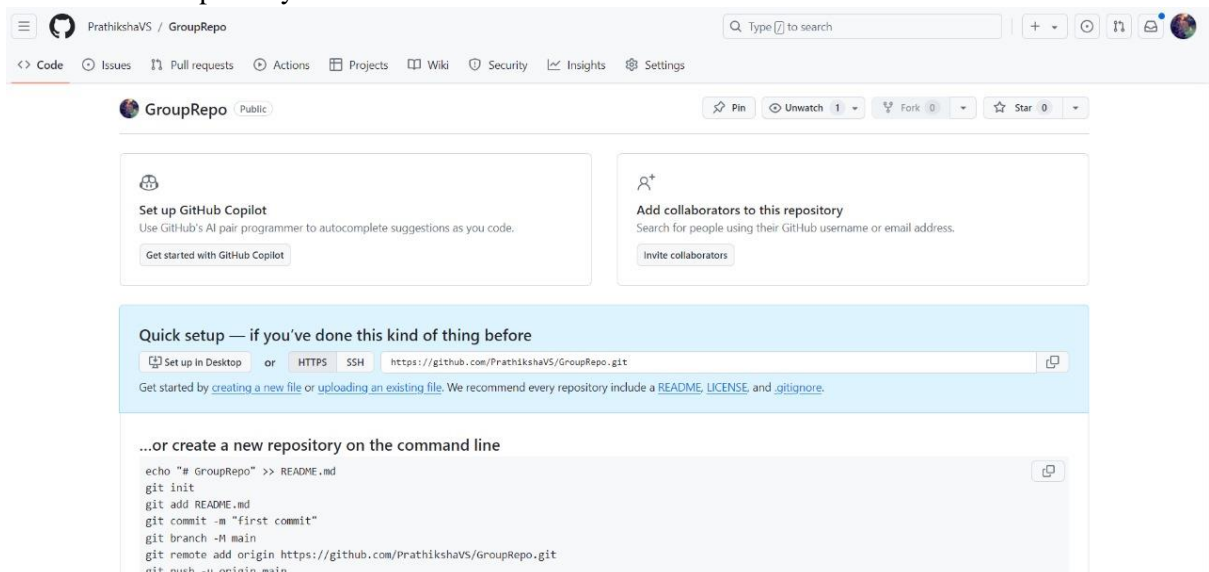
```
D:\Fullstack\GroupRepo>git remote add origin https://github.com/PrathikshaVS/GroupRepo.git

D:\Fullstack\GroupRepo>git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 218 bytes | 109.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/PrathikshaVS/GroupRepo.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

D:\Fullstack\GroupRepo>
```



3. All of the other members of the group should then be added as collaborators, so they can commit to the repository also.

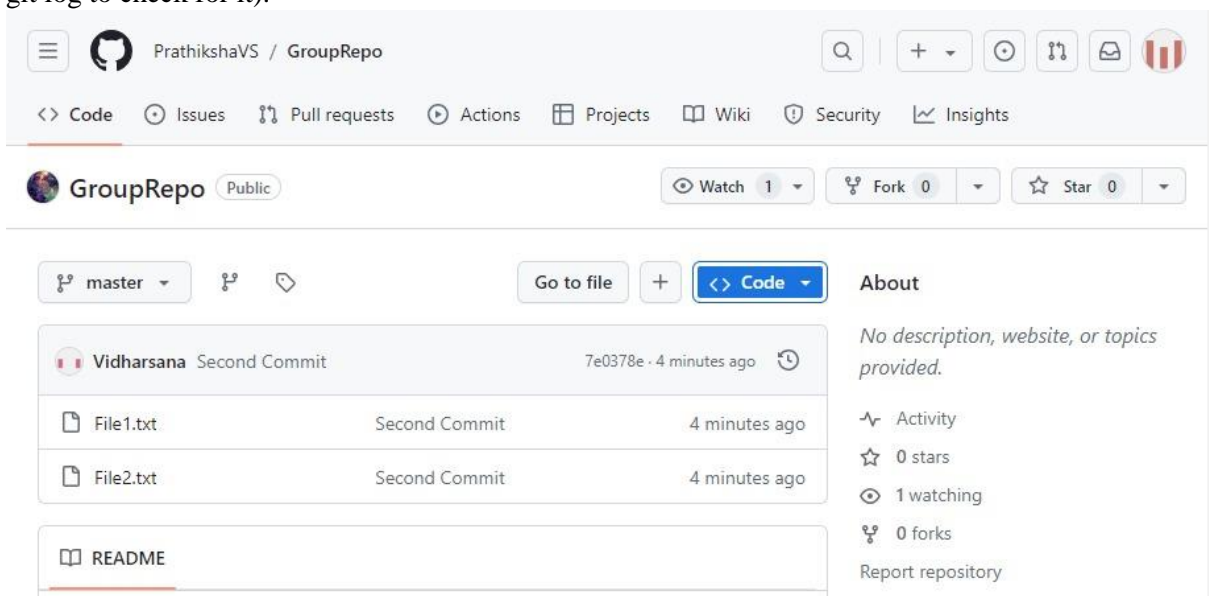


4. Next, everyone else in the group should clone the repository from GitHub. Verify that the context of the repository is what is expected.

```
C:\Users\welcome\Downloads\GitGroup>git clone https://github.com/PrathikshaVS/GroupRepo.git
Cloning into 'GroupRepo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

```
C:\Users\User\Downloads\GroupGitHub>git clone https://github.com/PrathikshaVS/GroupRepo.git
Cloning into 'GroupRepo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

5. One of the group members who just cloned should now make a local commit, then push it. Everyone should verify that when they pull, that commit is added to their local repository (use git log to check for it).



```

C:\Users\welcome\Downloads\GitGroup\GroupRepo>git pull https://github.com/PrathikshaVS/GroupRepo.git master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 271 bytes | 11.00 KiB/s, done.
From https://github.com/PrathikshaVS/GroupRepo
* branch      master      -> FETCH_HEAD
Updating beb22bf..7e0378e
Fast-forward
 File1.txt | 1 +
 File2.txt | 0
 2 files changed, 1 insertion(+)
 create mode 100644 File2.txt

C:\Users\welcome\Downloads\GitGroup\GroupRepo>git log
commit 7e0378e303c1f754445a3d53712acd9b91fab903 (HEAD -> master)
Author: Vidharsana <vidharsanasundharraj@gmail.com>
Date: Tue Jul 16 09:27:50 2024 +0530

    Second Commit

commit beb22bfccb54eaabb4e0879340afe53073db8377 (origin/master, origin/HEAD)
Author: PrathikshaVS <prathiksha4915@gmail.com>
Date: Tue Jul 16 09:17:57 2024 +0530

    First Commit

```

6. Look at each other's git log output. Notice how the SHA-1 is the same for a given commit across every copy of the repository. Why is this important?

```

C:\Users\welcome\Downloads\GitGroup\GroupRepo>git log
commit 7e0378e303c1f754445a3d53712acd9b91fab903 (HEAD -> master)
Author: Vidharsana <vidharsanasundharraj@gmail.com>
Date: Tue Jul 16 09:27:50 2024 +0530

    Second Commit

commit beb22bfccb54eaabb4e0879340afe53073db8377 (origin/master, origin/HEAD)
Author: PrathikshaVS <prathiksha4915@gmail.com>
Date: Tue Jul 16 09:17:57 2024 +0530

    First Commit

```

7. Two members of the group should now make a commit locally, and race to push it. To keep things simple, be sure to edit different files. What happens to the runner-up?

```

C:\Users\User\Downloads\GroupGitHub\GroupRepo>git add File2.txt

C:\Users\User\Downloads\GroupGitHub\GroupRepo>git commit -m "Fifth commit"
[master 8e8a80c] Fifth commit
 1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\User\Downloads\GroupGitHub\GroupRepo>git push https://github.com/PrathikshaVS/GroupRepo.git master
Enumerating objects: 13, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (9/9), 815 bytes | 815.00 KiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/PrathikshaVS/GroupRepo.git
 528815f..8e8a80c master -> master

```

```

C:\Users\welcome\Downloads\GitExercise>git log
commit 626aa05ffaee68c914b00764ec7f2c679943eef1 (HEAD -> master)
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:53:37 2024 +0530

    Rename OS Correct Commit

commit 6f9e656d77c61f09dbd21c030a1b713be96a4430
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:48:36 2024 +0530

    Rename Commit

commit 05fa9d6b7bddadcf5f7d15f2035f5906d6fe3ede
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:43:38 2024 +0530

    Delete Commit

commit 11d4e78c79e5f665891b2f2a05d440021a14b8ef
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:40:31 2024 +0530

    Readme Commit

commit f5739406f335d48d83f30e4fb793c185d3af0dd0
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:36:55 2024 +0530

    Finally Committing

commit 5929004ccf2cdfe83164b7841899120d9dc841a2
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:35:38 2024 +0530

    Final Commit

commit 9c04eb8df0dca39a957e330b737838f4937b42b1
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:32:53 2024 +0530

    Change Commit

commit 1a6898ff038ba942c317877d761d1510bc18cc35
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Mon Jul 15 21:29:13 2024 +0530

    Directory Commit

```

8. The runner-up should now pull. As a group, look at the output of the command. Additionally, look at the git log, and notice that there is a merge commit. You may also wish to view the DAG in gitk.

```

C:\Users\welcome\Downloads\GitGroup\GroupRepo>git pull https://github.com/PrathikshaVS/GroupRepo.git master
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 0), reused 9 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), 795 bytes | 11.00 KiB/s, done.
From https://github.com/PrathikshaVS/GroupRepo
* branch      master      -> FETCH_HEAD
Merge made by the 'ort' strategy.
 File2.txt | 2 ++
 1 file changed, 2 insertions(+)

```



```

C:\Users\welcome\Downloads\GitGroup\GroupRepo>git log
commit 2528aeb22ea70b9a9974c513138ef5adc3cdc4b3 (HEAD -> master)
Merge: d8af7d3 8e8a80c
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Tue Jul 16 10:18:51 2024 +0530

    Merge branch 'master' of https://github.com/PrathikshaVS/GroupRepo

commit d8af7d3c17d9311996d642633814d4e0c4d9c395
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Tue Jul 16 10:12:20 2024 +0530

    Fifth Commit

commit 8e8a80ca9991b104082279ed0cd137ace3162b18
Author: Harisha-08 <harshachitra0814@gmail.com>
Date: Tue Jul 16 10:09:33 2024 +0530

    Fifth commit

commit fe5c2117f41f8d6d4bf0080c1862cfd254515102
Merge: d090dfe 528815f
Author: Harisha-08 <harshachitra0814@gmail.com>
Date: Tue Jul 16 10:04:58 2024 +0530

    Fourth commit

commit d090dfe8f7dfc5ce1a4852ae6a41bd38f007bb4f
Author: Harisha-08 <harshachitra0814@gmail.com>
Date: Tue Jul 16 09:43:52 2024 +0530

    Fourth commit

commit 528815f4aade2395f65f10f6885f4b2a6449108
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Tue Jul 16 09:42:07 2024 +0530

    File1 Commit

commit 7e0378e303c1f754445a3d53712acd9b91fab903
Author: Vidharsana <vidharsanasundharraj@gmail.com>
Date: Tue Jul 16 09:27:50 2024 +0530

    Second Commit

commit beb22bfcbb54eaabb4e0879340afe53073db8377 (origin/master, origin/HEAD)
Author: PrathikshaVS <prathiksha4915@gmail.com>
Date: Tue Jul 16 09:17:57 2024 +0530

    First Commit
(END)

```

```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>
```

The screenshot shows the Git GUI application interface. At the top, there's a menu bar with 'File', 'Edit', 'View', and 'Help'. Below it is a commit history graph on the left, showing a sequence of commits from 'First Commit' to 'master'. The 'master' branch is highlighted in green. To the right of the graph is a table listing the commits with their SHA1 IDs, authors, and dates.

| SHA1 ID                                  | Author  | Date                |
|--|---|---------------------|
| 2528aeb22ea70b9a9974c513138ef5adc3cdc4b3 | DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> | 2024-07-16 10:18:51 |
| d8af7d3c17d9311996d642633814d4e0c4d9c395 | Harisha-08 <harshachitra0814@gmail.com>           | 2024-07-16 10:09:33 |
| 8e8a80ca9991b104082279ed0cd137ace3162b18 | Harisha-08 <harshachitra0814@gmail.com>           | 2024-07-16 10:04:58 |
| fe5c2117f41f8d6d4bf0080c1862cfd254515102 | Harisha-08 <harshachitra0814@gmail.com>           | 2024-07-16 09:43:52 |
| d090dfe8f7dfc5ce1a4852ae6a41bd38f007bb4f | DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> | 2024-07-16 09:42:07 |
| 528815f4aade2395f65f10f6885f4b2a6449108  | Vidharsana <vidharsanasundharraj@gmail.com>       | 2024-07-16 09:27:50 |
| beb22bfcbb54eaabb4e0879340afe53073db8377 | PrathikshaVS <prathiksha4915@gmail.com>           | 2024-07-16 09:17:57 |

Below the table, there's a search bar and a 'Find' button. The 'Find' button is highlighted. Below the search bar, there's a 'Diff' tab selected, showing the changes between the selected commit and its parent. The changes are listed in a table with columns for 'Author', 'Committer', 'Parent', 'Branch', and 'Follows'. The 'Diff' tab shows the changes in the file 'File1.txt'.

The 'Diff' tab shows the following changes:

- Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-16 10:18:51
- Committer: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com> 2024-07-16 10:18:51
- Parent: d8af7d3c17d9311996d642633814d4e0c4d9c395 (Fifth Commit)
- Parent: 8e8a80ca9991b104082279ed0cd137ace3162b18 (Fifth commit)
- Branch: master
- Follows:
- Precedes:

The 'Diff' tab shows the changes in the file 'File1.txt'.

9. Repeat the last two steps a couple of times, to practice.

```
D:\Fullstack\GroupRepo>git pull https://github.com/PrathikshaVS/GroupRepo.git master
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 12 (delta 0), reused 12 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), 1.03 KiB | 17.00 KiB/s, done.
From https://github.com/PrathikshaVS/GroupRepo
 * branch          master      -> FETCH_HEAD
Updating 7e0378e..8e8a80c
Fast-forward
 File1.txt | 2 +-
 File2.txt | 2 ++
 2 files changed, 3 insertions(+), 1 deletion(-)

D:\Fullstack\GroupRepo>git add .

D:\Fullstack\GroupRepo>git commit -m "File1 edit Commit"
[master 9ed30cb] File1 edit Commit
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\Fullstack\GroupRepo>git push -u origin master
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), 284 bytes | 284.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/PrathikshaVS/GroupRepo.git
 8e8a80c..9ed30cb master -> master
branch 'master' set up to track 'origin/master'.

D:\Fullstack\GroupRepo>
```

```
D:\GroupRepo>git pull https://github.com/PrathikshaVS/GroupRepo.git master
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 12 (delta 0), reused 12 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), 1.03 KiB | 80.00 KiB/s, done.
From https://github.com/PrathikshaVS/GroupRepo
 * branch          master      -> FETCH_HEAD
Updating 7e0378e..8e8a80c
error: Your local changes to the following files would be overwritten by merge:
    File2.txt
Please commit your changes or stash them before you merge.
Aborting

D:\GroupRepo>git add .

D:\GroupRepo>git commit -m "Commit1"
[master 6648a36] Commit1
 1 file changed, 1 insertion(+)
```

```
D:\GroupRepo>git push -u https://github.com/PrathikshaVS/GroupRepo.git master
To https://github.com/PrathikshaVS/GroupRepo.git
 ! [rejected]        master -> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/PrathikshaVS/GroupRepo.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. If you want to integrate the remote changes,
hint: use 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

```
D:\GroupRepo>git pull https://github.com/PrathikshaVS/GroupRepo.git master
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 15 (delta 0), reused 15 (delta 0), pack-reused 0
Unpacking objects: 100% (15/15), 1.27 KiB | 81.00 KiB/s, done.
From https://github.com/PrathikshaVS/GroupRepo
* branch          master      -> FETCH_HEAD
Auto-merging File2.txt
CONFLICT (content): Merge conflict in File2.txt
Automatic merge failed; fix conflicts and then commit the result.
```

```
D:\GroupRepo>git log
commit 6648a36d98b82857435255719ab16bda112cb41b (HEAD -> master)
Author: Vidharsana <vidharsanasundharraj@gmail.com>
Date: Tue Jul 16 10:31:23 2024 +0530
```

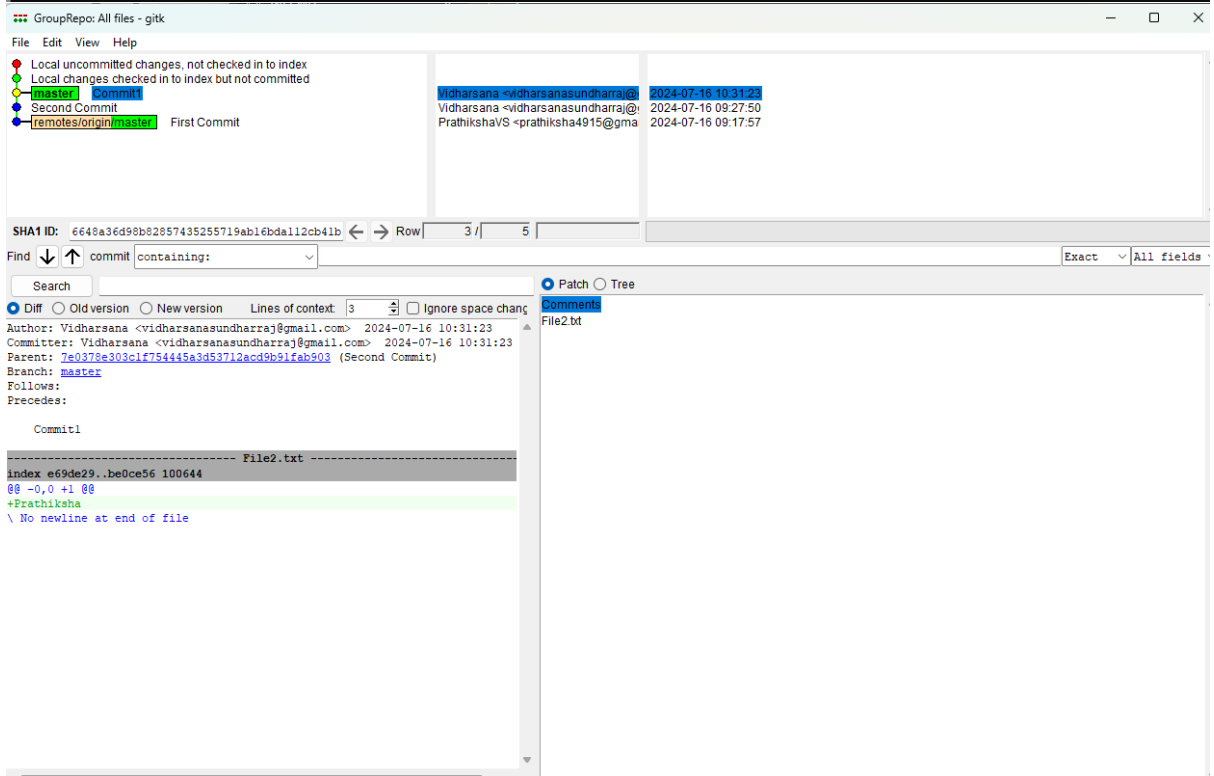
Commit1

```
commit 7e0378e303c1f754445a3d53712acd9b91fab903
Author: Vidharsana <vidharsanasundharraj@gmail.com>
Date: Tue Jul 16 09:27:50 2024 +0530
```

Second Commit

```
commit beb22bfcbb54eaabb4e0879340afe53073db8377 (origin/master, origin/HEAD)
Author: PrathikshaVS <prathiksha4915@gmail.com>
Date: Tue Jul 16 09:17:57 2024 +0530
```

First Commit



## Stretch Task

1. Now create a situation where two group members both edit the same line in the same file and commit it locally. Race to push.



```
C:\Users\User\Downloads\GroupGitHub\GroupRepo>git add File2.txt

C:\Users\User\Downloads\GroupGitHub\GroupRepo>git commit -m "stretch commit"
[master aa56306] stretch commit
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git add File2.txt

C:\Users\welcome\Downloads\GitGroup\GroupRepo>git commit -m "Stretch Commit"
[master 4159e94] Stretch Commit
1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\welcome\Downloads\GitGroup\GroupRepo>git push -u https://github.com/PrathikshaVS/GroupRepo.git master
To https://github.com/PrathikshaVS/GroupRepo.git
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/PrathikshaVS/GroupRepo.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

- When the runner-up does a pull, they should get a merge conflict.

```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git pull https://github.com/PrathikshaVS/GroupRepo.git master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 264 bytes | 8.00 KiB/s, done.
From https://github.com/PrathikshaVS/GroupRepo
* branch      master      -> FETCH_HEAD
Auto-merging File1.txt
CONFLICT (content): Merge conflict in File1.txt
Automatic merge failed; fix conflicts and then commit the result.
```

### E325: ATTENTION

```
Found a swap file by the name "~/Downloads/GroupGitHub/GroupRepo/.git/.MERGE_MSG.swp"
  owned by: User   dated: Tue Jul 16 10:03:29 2024
    file name: ~User/Downloads/GroupGitHub/GroupRepo/.git/MERGE_MSG
    modified: YES
    user name: User   host name: AIDS-LAB-15
    process ID: 1354
While opening file "/c/Users/User/Downloads/GroupGitHub/GroupRepo/.git/MERGE_MSG"
  dated: Tue Jul 16 10:47:50 2024
  NEWER than swap file!
```

- Look as a group at the file in conflict, and resolve it.

### E325: ATTENTION

```
Found a swap file by the name "~/Downloads/GroupGitHub/GroupRepo/.git/.MERGE_MSG.swp"
  owned by: User   dated: Tue Jul 16 10:03:29 2024
    file name: ~User/Downloads/GroupGitHub/GroupRepo/.git/MERGE_MSG
    modified: YES
    user name: User   host name: AIDS-LAB-15
    process ID: 1354
While opening file "/c/Users/User/Downloads/GroupGitHub/GroupRepo/.git/MERGE_MSG"
  dated: Tue Jul 16 10:47:50 2024
  NEWER than swap file!
```

- Another program may be editing the same file. If this is the case, be careful not to end up with two different instances of the same file when making changes. Quit, or continue with caution.
- An edit session for this file crashed.
  - If this is the case, use `:recover` or `"vim -r /c/Users/User/Downloads/GroupGitHub/GroupRepo/.git/MERGE_MSG"` to recover the changes (see `":help recovery"`).
  - If you did this already, delete the swap file `"/c/Users/User/Downloads/GroupGitHub/GroupRepo/.git/.MERGE_MSG.swp"` to avoid this message.

```
Swap file "~/Downloads/GroupGitHub/GroupRepo/.git/.MERGE_MSG.swp" already exists!
[O]pen Read-Only, [E]dit anyway, [R]ecover, [D]elete it, [Q]uit, [A]bort:
```

```
E211: File "~/Downloads/GroupGitHub/GroupRepo/.git/MERGE_MSG" no longer available
```

```
Swap file "~/Downloads/GroupGitHub/GroupRepo/.git/.MERGE_MSG.swp" already exists!
[O]pen Read-Only, [E]dit anyway, [R]ecover, [D]elete it, [Q]uit, [A]bort:
Swap file "~/Downloads/GroupGitHub/GroupRepo/.git/.MERGE_MSG.swp" already exists!
[O]pen Read-Only, [E]dit anyway, [R]ecover, [D]elete it, [Q]uit, [A]bort:
Swap file "~/Downloads/GroupGitHub/GroupRepo/.git/.MERGE_MSG.swp" already exists!
-- More -- SPACE/d/j: screen/page/line down, b/u/k: up, q: quit |
```

4. Use the add command to stage the fix, and then use commit to make the merge commit. Notice how this procedure is exactly the one you got used to when resolving conflicts in branches.

```
C:\Users\User\Downloads\GroupGitHub\GroupRepo>git add .

C:\Users\User\Downloads\GroupGitHub\GroupRepo>git commit -m "Merge commit"
[master ffd4f5a] Merge commit
1 file changed, 1 deletion(-)

C:\Users\User\Downloads\GroupGitHub\GroupRepo>git push -u https://github.com/PrathikshaVS/GroupRepo.git master
Enumerating objects: 15, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (11/11), 1.06 KiB | 1.06 MiB/s, done.
Total 11 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/PrathikshaVS/GroupRepo.git
9ed30cb..ffd4f5a master -> master
branch 'master' set up to track 'https://github.com/PrathikshaVS/GroupRepo.git/master'.

C:\Users\welcome\Downloads\GitGroup\GroupRepo>git pull https://github.com/PrathikshaVS/GroupRepo.git master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 264 bytes | 8.00 KiB/s, done.
From https://github.com/PrathikshaVS/GroupRepo
* branch          master       -> FETCH_HEAD
Auto-merging File1.txt
CONFLICT (content): Merge conflict in File1.txt
Automatic merge failed; fix conflicts and then commit the result.
```

## Exercise 4

### Main Task

1. Make a commit, and make a silly typo in the commit message.

```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git commit -m "Silly Commit"
git: 'commit -m' is not a git command. See 'git --help'.

The most similar command is
commit-tree
```

2. Use the --amend flag to enable you to fix the commit message.

```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git commit --amend -m "Amend Commit"
[master 55b1cca] Amend Commit
Date: Fri Jul 19 13:45:33 2024 +0530
```

3. Look at the log and notice how the mistake is magically gone.

```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git commit --amend -m "Amend Commit"
[master 55b1cca] Amend Commit
Date: Fri Jul 19 13:45:33 2024 +0530

C:\Users\welcome\Downloads\GitGroup\GroupRepo>git log
commit 55b1ccae9b48b286ce5d814fa721147107952403 (HEAD -> master)
Merge: 4159e94 9ed30cb
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Fri Jul 19 13:45:33 2024 +0530

    Amend Commit

commit 4159e948221448da225b0b7fca31f8e748fe1e77
Author: DharaniChandrasekarn31 <c.v.dharanisri@gmail.com>
Date: Tue Jul 16 10:45:00 2024 +0530

    Stretch Commit

commit 9ed30cbeedfddafb8fe6ece97b87442f9dad624
Author: PrathikshaVS <prathiksha4915@gmail.com>
Date: Tue Jul 16 10:31:24 2024 +0530
```

4. Now make a commit where you make a typo in one of the files. Once again, use --amend to magic away your problems.

```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git add .  
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git commit -m "2nd Amend"  
[master bdec6b1] 2nd Amend  
1 file changed, 1 insertion(+), 1 deletion(-)
```

5. Create a branch. Make a commit.

```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git branch Jaga  
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git checkout Jaga  
Switched to branch 'Jaga'  
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git commit -m "new branc commit"  
On branch Jaga  
nothing to commit, working tree clean
```

6. Now switch back to your master branch. Make a (non-conflicting) commit there also.

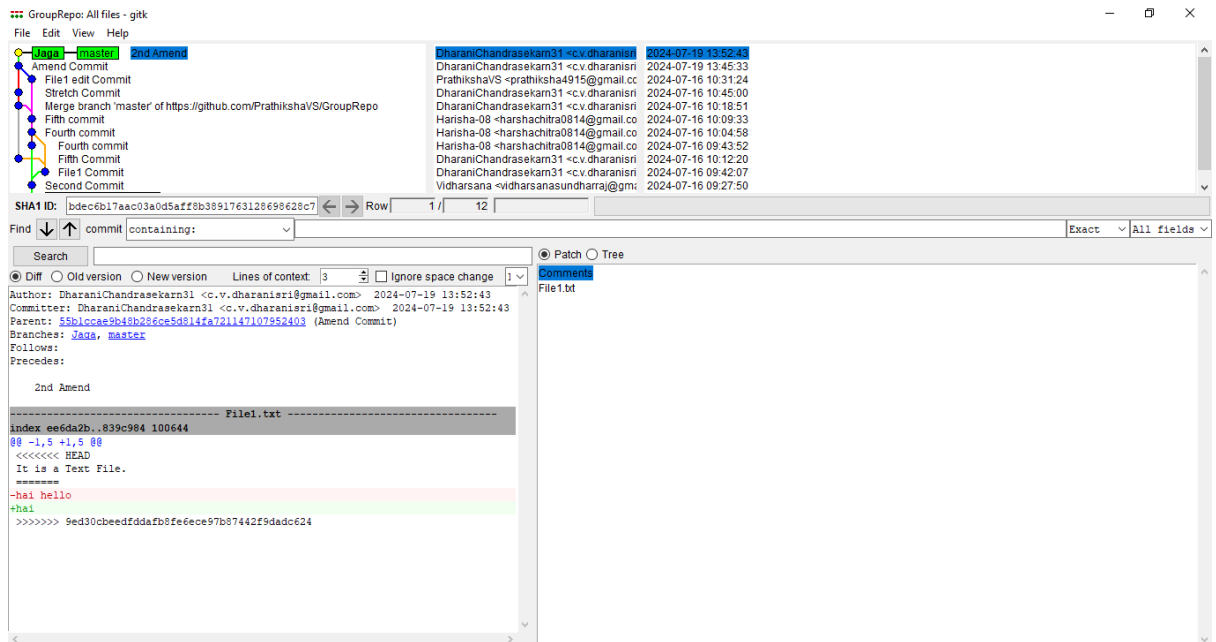
```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git checkout master  
Switched to branch 'master'  
Your branch is ahead of 'origin/master' by 11 commits.  
(use "git push" to publish your local commits)  
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git commit -m "Commit master"  
On branch master  
Your branch is ahead of 'origin/master' by 11 commits.  
(use "git push" to publish your local commits)  
nothing to commit, working tree clean
```

7. Now switch back to your branch.

```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git checkout Jaga  
Switched to branch 'Jaga'
```

8. Use the rebase command in your branch. Look at the DAG in gitk, and note that you have the commit from the master branch, but no merge commit.

```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git rebase  
There is no tracking information for the current branch.  
Please specify which branch you want to rebase against.  
See git-rebase(1) for details.  
  
git rebase '<branch>'  
  
If you wish to set tracking information for this branch you can do so with:  
  
git branch --set-upstream-to=<remote>/<branch> Jaga  
  
C:\Users\welcome\Downloads\GitGroup\GroupRepo>gitk
```



9. Make one more commit in your branch.

```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git add .

C:\Users\welcome\Downloads\GitGroup\GroupRepo>git commit -m "COMMITTED"
[Jaga 53fd304] COMMITTED
1 file changed, 1 insertion(+), 1 deletion(-)
```

10. Return to master. Merge your branch. Notice how, thanks to the rebase, this is a fastforward merge.

```
C:\Users\welcome\Downloads\GitGroup\GroupRepo>git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 11 commits.
(use "git push" to publish your local commits)

C:\Users\welcome\Downloads\GitGroup\GroupRepo>git merge Jaga
Updating bdec6b1..53fd304
Fast-forward
 File1.txt | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```