Experiment -2.4

Student Name: Km Ayushi **UID:22BDO10055**
Branch: CSE(DevOps) **Section/Group-22BCD-1(B)**
Semester: 4th **Date of Performance:29/02/24**
Subject Name- Git and GitHub **Subject Code:22CSH-293**

1. Aim/Overview of the practical: Git Merge Conflicts and resolving Git merge conflicts

2. Software used: Git bash and GitHub

3. Steps for experiment/practical:

- Create a directory name e7 now use git init and add the origin by copying the URL of the exp7 repository.
- Create a file name file1 add some content to the file and then commit that file.

```
ayush@LAPTOP-14JQ3IMU MINGW64 ~ (main)
$ mkdir e7

ayush@LAPTOP-14JQ3IMU MINGW64 ~ (main)
$ cd e7

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ git remote add origin https://github.com/Ayushi121104/exp7.git

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ git init
Initialized empty Git repository in C:/Users/ayush/e7/.git/

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ git remote add origin https://github.com/Ayushi121104/exp7.git

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ git remote add origin https://github.com/Ayushi121104/exp7.git
error: remote origin already exists.

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ touch file1

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ vi file1

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ git add file1
warning: in the working copy of 'file1', LF will be replaced by CRLF the next time Git touches it

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ git commit -m"file1 added"
[main (root-commit) b96a9ca] file1 added
 1 file changed, 15 insertions(+)
 create mode 100644 file1
```

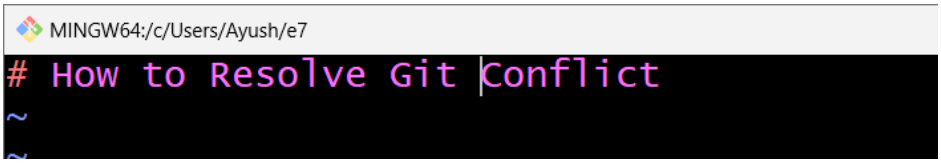
- Now create a new branch named Master Open the same file and make some changes check out to the main branch and merge it to the master branch. Now you can easily check the status with the help of a git log.

```
ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ git checkout -b master
Switched to a new branch 'master'

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (master)
$ vi file1

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (master)
$ git add file1
warning: in the working copy of 'file1', LF will be rep
laced by CRLF the next time Git touches it

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (master)
$ git commit -m"file1 updated"
[master 69edc58] file1 updated
 1 file changed, 4 insertions(+), 4 deletions(-)

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (master)
$ git checkout main
Switched to branch 'main'

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ git merge master
Updating b96a9ca..69edc58
Fast-forward
 file1 | 8 ++++----
 1 file changed, 4 insertions(+), 4 deletions(-)

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ git log
commit 69edc585303cacae049ccd6cd0870b0af90605b2 (HEAD -
> main, master)
Author: Ayushi <ayushi121104@gmail.com>
Date:   Mon Apr 8 14:19:03 2024 +0530

    file1 updated
```

- Now push the main branch to the remote repository.

Merge conflict and resolving it –

- Open a new file named readme add some content to that file and then commit it.



- Now create a new branch name read me add some content in that file and update it.

```
MINGW64:/c/Users/Ayush/e7
# How to Resolve Git merge Conflict
~
```

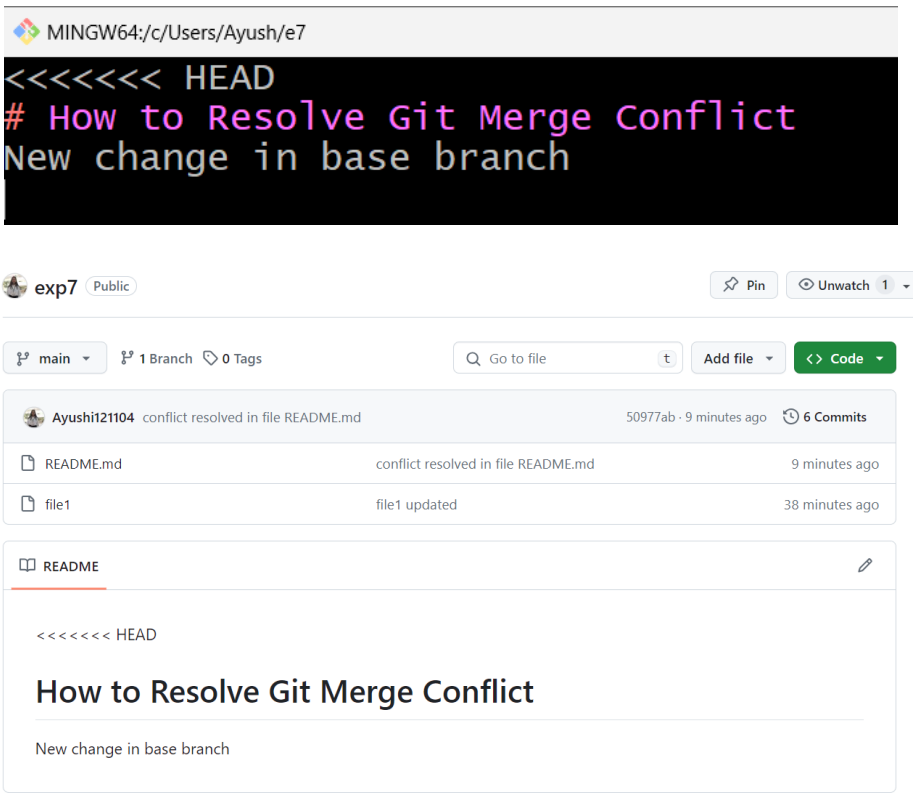
- Now check out to the main branch open the same file update some content of the file.

```
ayush@LAPTOP-14JQ3IMU MINGW64 ~ (main)
$ git push origin main
To https://github.com/Ayushi121104/merge-conflict.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/
Ayushi121104/merge-conflict.git'
hint: Updates were rejected because the remote contains
 work that you do not
hint: have locally. This is usually caused by another r
epository 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.
```

- After comparing the main branch with the readme file, it appears that conflicts are preventing the file from merging.

```
ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ git merge readme
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit t
he result.

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main|MERGING)
$ vi README.md

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main|MERGING)
$ git commit -am "conflict resolved in file README.md"
[main 50977ab] conflict resolved in file README.md

ayush@LAPTOP-14JQ3IMU MINGW64 ~/e7 (main)
$ git push origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (12/12), 1.15 KiB | 1.15 MiB/s, d
one.
Total 12 (delta 1), reused 0 (delta 0), pack-reused 0 (
from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Ayushi121104/exp7.git
   69edc58..50977ab  main -> main
```

- Open the file in the main branch and resolve the conflict Then commit the changes, That's how the conflict can be solved Then push the file to the main branch.



9. Result/Output/Writing Summary:

In this experiment, we learned how to merge two branches and resolve merge conflicts that may occur during a project.

Learning outcomes (What I have learnt):

1. Learn about merging the branch into the main.
2. Learn about Merge conflict.
3. Learn about how to solve merge conflicts.
4. Learn how to check on the vi editor when conflict occurs.
5. Learn how to find the difference between the content of the two branches.

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.			