Assignment 2: Version Control with Git

Name : Souvik Roy

Id : 25SUB4508_56133

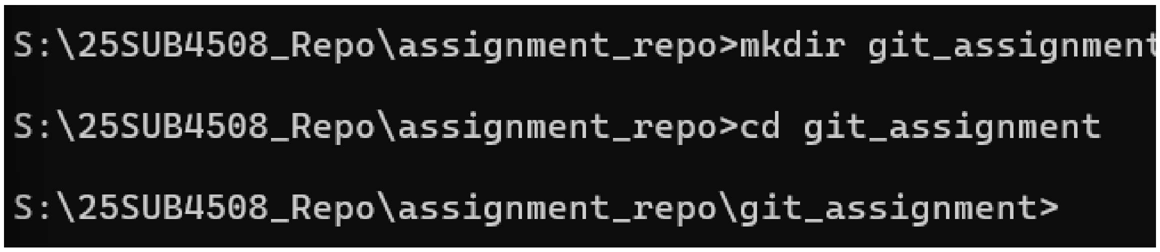
Objective:
To Demonstrate basic Git version control operations including repository initialization, committing changes, branching, merging, and resolving merge conflicts.

Section 1: Initializing Git Repository

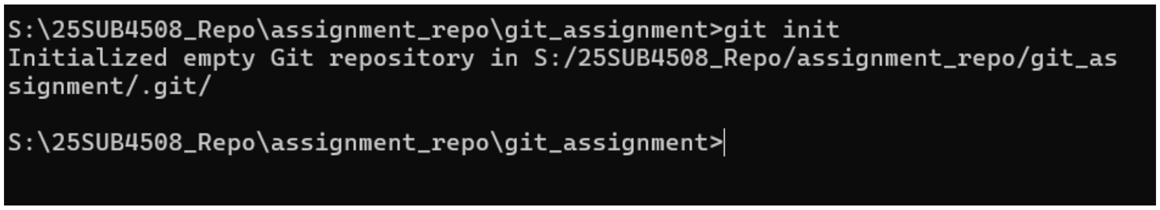
Command:
git init

Output:
Initialized empty Git repository

Screenshot:



```
S:\25SUB4508_Repo\assignment_repo>mkdir git_assignment

S:\25SUB4508_Repo\assignment_repo>cd git_assignment

S:\25SUB4508_Repo\assignment_repo\git_assignment>
```



```
S:\25SUB4508_Repo\assignment_repo\git_assignment>git init
Initialized empty Git repository in S:/25SUB4508_Repo/assignment_repo/git_as
signment/.git/

S:\25SUB4508_Repo\assignment_repo\git_assignment>
```

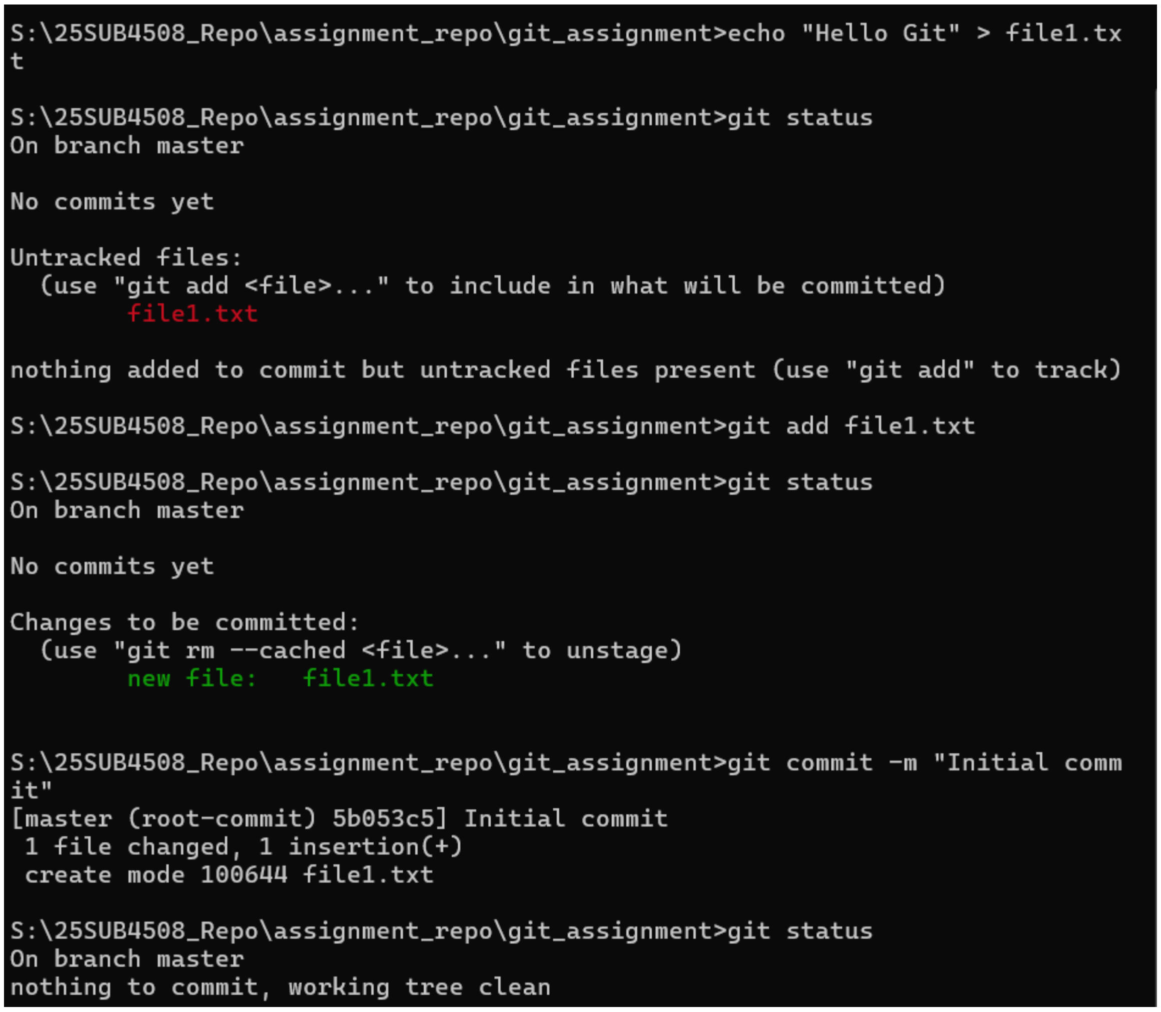
—

Section 2: Adding Files and Committing

Commands:
echo "Hello Git" > file1.txt
git add file1.txt
git commit -m "Initial commit"

Output:
1 file changed

Screenshot :



```
S:\25SUB4508_Repo\assignment_repo\git_assignment>echo "Hello Git" > file1.tx
t

S:\25SUB4508_Repo\assignment_repo\git_assignment>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1.txt

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

S:\25SUB4508_Repo\assignment_repo\git_assignment>git add file1.txt

S:\25SUB4508_Repo\assignment_repo\git_assignment>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt


S:\25SUB4508_Repo\assignment_repo\git_assignment>git commit -m "Initial comm
it"
[master (root-commit) 5b053c5] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt

S:\25SUB4508_Repo\assignment_repo\git_assignment>git status
On branch master
nothing to commit, working tree clean
```

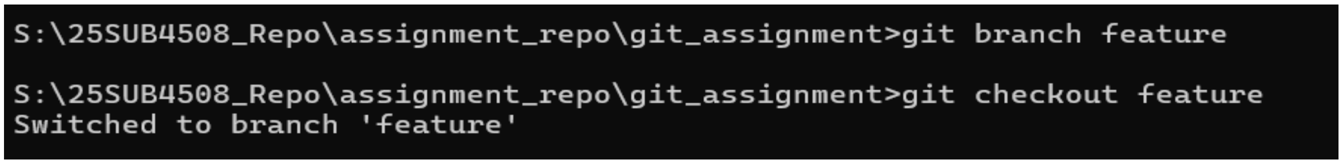
—

Section 3: Creating and Switching Branch

Commands:
git branch feature
git checkout feature

Output:
Switched to branch 'feature'

Screenshot :



```
S:\25SUB4508_Repo\assignment_repo\git_assignment>git branch feature

S:\25SUB4508_Repo\assignment_repo\git_assignment>git checkout feature
Switched to branch 'feature'
```

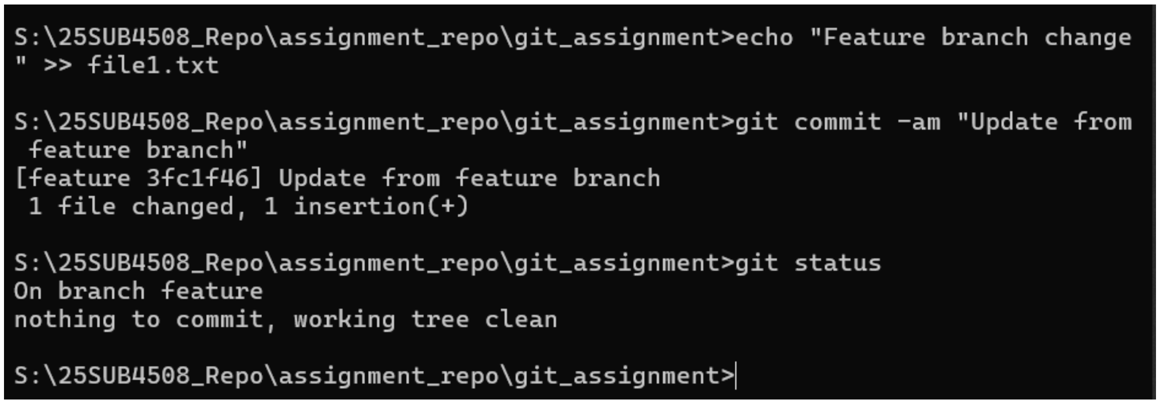
—

Section 4: Modifying File in Feature Branch

Commands:
echo "Feature change" >> file1.txt
git commit -am "Update from feature branch"

Output:
1 file changed

Screenshot :



```
S:\25SUB4508_Repo\assignment_repo\git_assignment>echo "Feature branch change
" >> file1.txt

S:\25SUB4508_Repo\assignment_repo\git_assignment>git commit -am "Update from
 feature branch"
[feature 3fc1f46] Update from feature branch
 1 file changed, 1 insertion(+)

S:\25SUB4508_Repo\assignment_repo\git_assignment>git status
On branch feature
nothing to commit, working tree clean

S:\25SUB4508_Repo\assignment_repo\git_assignment>
```

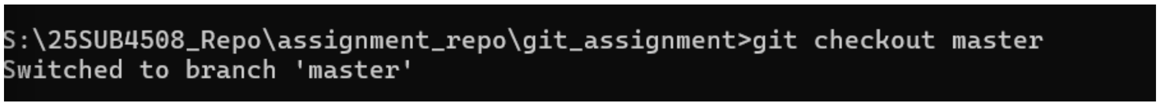
—

Section 5: Merge Conflict Simulation

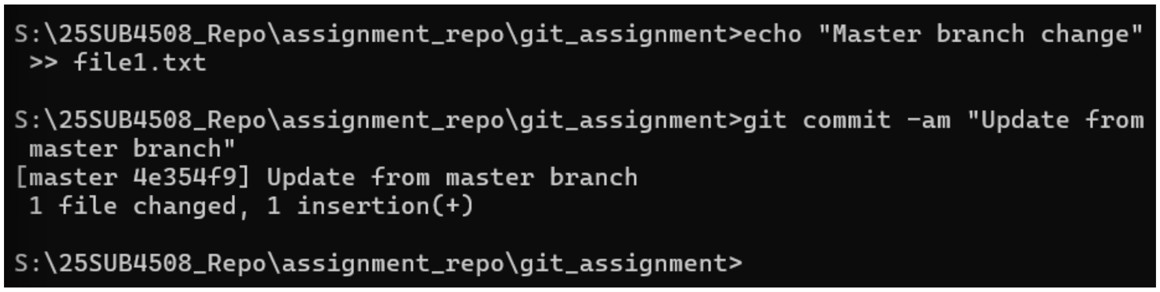
Commands:
git checkout master
echo "Master branch change" >> file1.txt
git commit -am "Update from master branch"
git merge feature

Output:
CONFLICT (content): Merge conflict in file1.txt

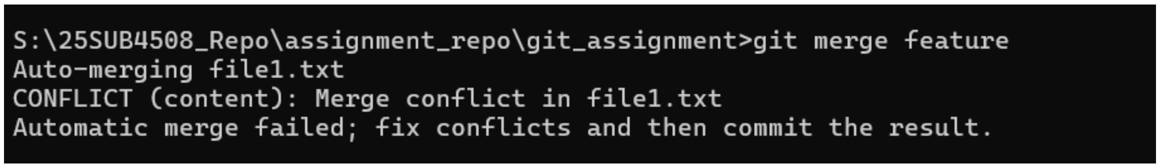
Screenshot :



```
S:\25SUB4508_Repo\assignment_repo\git_assignment>git checkout master
Switched to branch 'master'
```



```
S:\25SUB4508_Repo\assignment_repo\git_assignment>echo "Master branch change"
 >> file1.txt

S:\25SUB4508_Repo\assignment_repo\git_assignment>git commit -am "Update from
 master branch"
[master 4e354f9] Update from master branch
 1 file changed, 1 insertion(+)

S:\25SUB4508_Repo\assignment_repo\git_assignment>
```



```
S:\25SUB4508_Repo\assignment_repo\git_assignment>git merge feature
Auto-merging file1.txt
CONFLICT (content): Merge conflict in file1.txt
Automatic merge failed; fix conflicts and then commit the result.
```

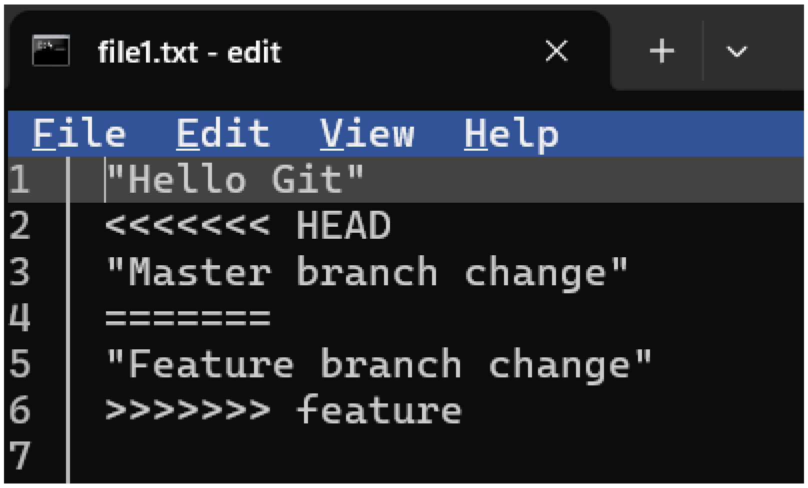
—

Section 6: Resolving Conflict

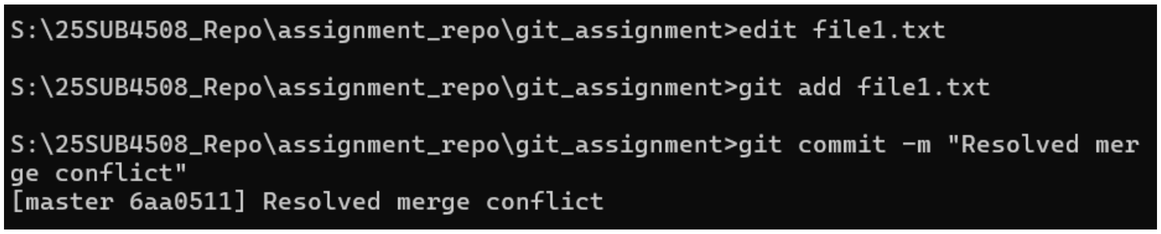
Edit file, resolve conflict, then:

git add file1.txt
git commit -m "Resolved merge conflict"

Screenshot :



```
"Hello Git"
<<<<<<< HEAD
"Master branch change"
=======
"Feature branch change"
>>>>>>> feature

```



```
S:\25SUB4508_Repo\assignment_repo\git_assignment>edit file1.txt

S:\25SUB4508_Repo\assignment_repo\git_assignment>git add file1.txt

S:\25SUB4508_Repo\assignment_repo\git_assignment>git commit -m "Resolved mer
ge conflict"
[master 6aa0511] Resolved merge conflict
```

—

Section 7: Verifying Branches and Commit History

After completing the merge and resolving the conflict, Git provides commands to verify the current branch structure and commit history. This step helps confirm that all branches are correctly merged and the repository is in a clean state.

Command 1: View Available Branches, Command 2: View Commit History in Graph Format

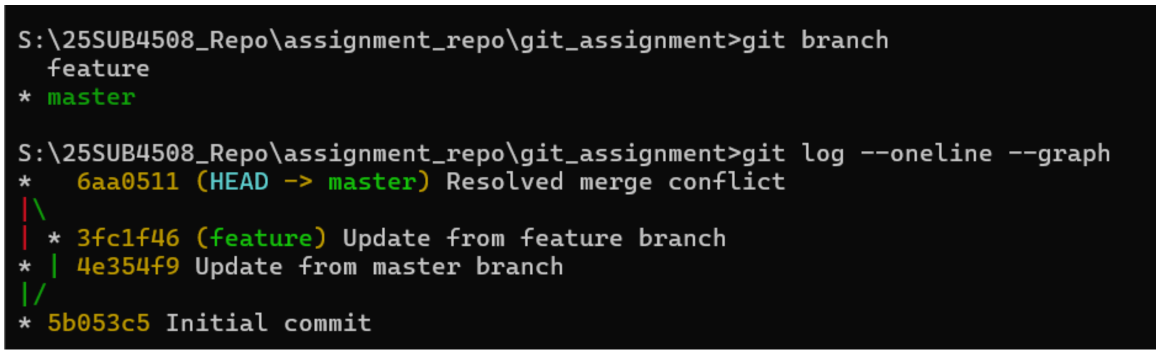
```
git branch

git log --oneline --graph
```

Output:

```
* master

  feature

*   a1b2c3d Resolved merge conflict

|\

| * d4e5f6g Update from feature branch

* | h7i8j9k Update from master branch

|/

* k1l2m3n Initial commit
```

This confirms that the user is currently on the `master` branch and the `feature` branch exists.

Screenshot :



```
S:\25SUB4508_Repo\assignment_repo\git_assignment>git branch
  feature
* master

S:\25SUB4508_Repo\assignment_repo\git_assignment>git log --oneline --graph
*   6aa0511 (HEAD -> master) Resolved merge conflict
|\
| * 3fc1f46 (feature) Update from feature branch
* | 4e354f9 Update from master branch
|/
* 5b053c5 Initial commit
```

External References:

1. GitHub Docs