FSD LAB1

FREYA MAKWANA

1032211790

PE-51

Aim: Version control with Git.

Objectives:

- To introduce the concepts and software behind version control, using the example of Git.
- To understand the use of 'version control' in the context of a coding project.
- To learn Git version control with Clone, commit to, and push, pull from a git repository.

Theory:

1. What is branching in git?

Ans. In Git, branching is a fundamental concept that allows you to create separate lines of development within a repository. Each branch represents an independent line of work, and you can use branches to work on different features, bug fixes, or experiments without affecting the main or "master" branch until you're ready to merge your changes.

Branching is a powerful feature in Git that promotes collaborate on, experiments on, and the isolate on of different features or bug fixes. It allows teams to work on multiple aspects of a project simultaneously while maintaining a stable main branch for produce on.

2. How to create and merge branches in Git? Write the commands used.

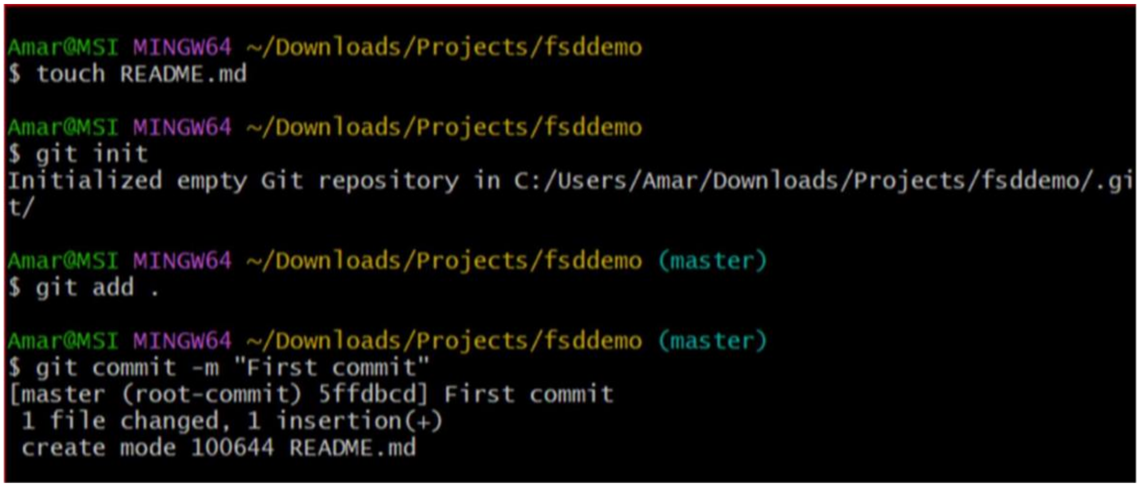
Ans. Create a new branch: To create a new branch, you can use the git branch command followed by the name you want to give to your branch. For example, if you want to create a branch called "my-feature," you would run: git branch my-feature

Merge the feature branch: Use the git merge command to merge your feature branch into the main branch: git merge my-feature

This will attempt to merge your changes from "my-feature" into the main branch. If there are no conflicts, Git will perform an automatic merge. If there are conflicts, you'll need to resolve them manually and then commit the resolved changes.

GITHUB REPO URL: https://github.com/FreyaMak/FSD-Submissions

Screenshots:

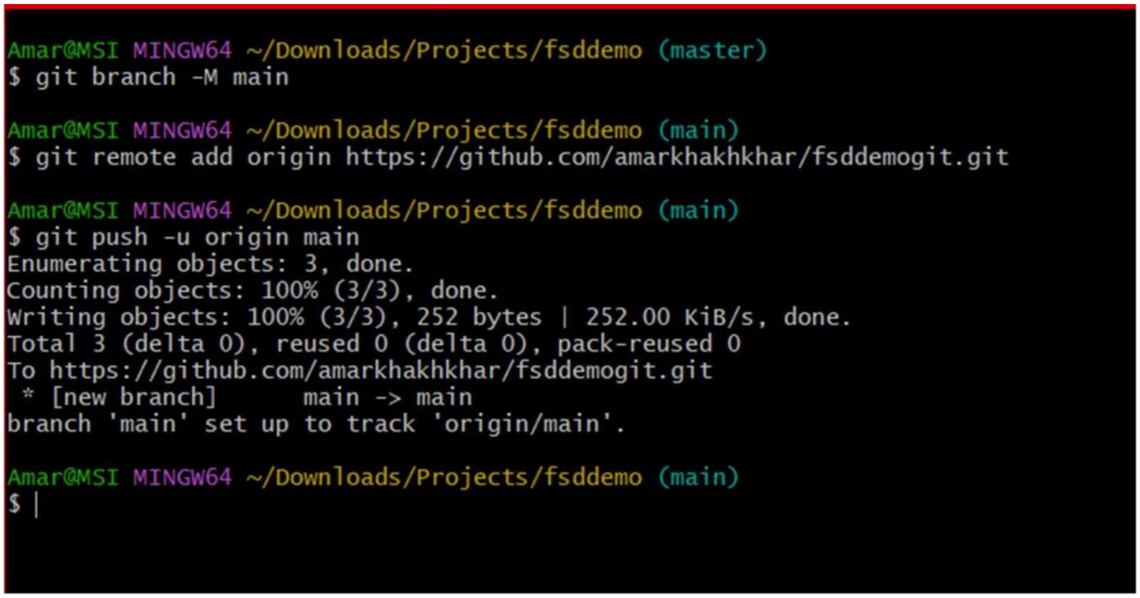


```
Amar@MSI MINGW64 ~/Downloads/Projects/fsddemo
$ touch README.md

Amar@MSI MINGW64 ~/Downloads/Projects/fsddemo
$ git init
Initialized empty Git repository in C:/Users/Amar/Downloads/Projects/fsddemo/.gi
t/

Amar@MSI MINGW64 ~/Downloads/Projects/fsddemo (master)
$ git add .

Amar@MSI MINGW64 ~/Downloads/Projects/fsddemo (master)
$ git commit -m "First commit"
[master (root-commit) 5ffdbcd] First commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

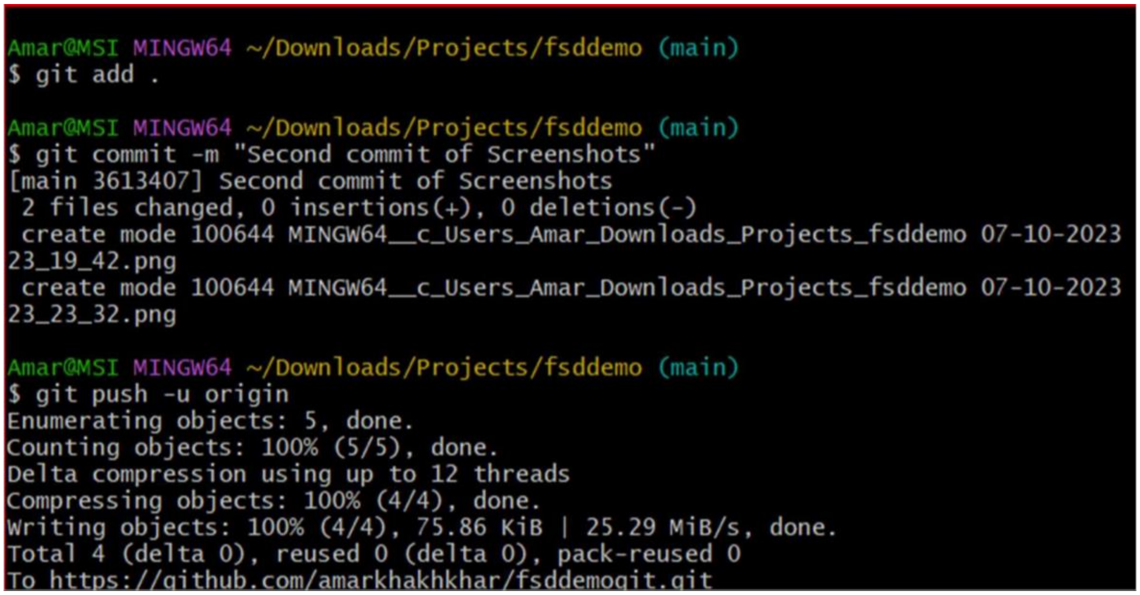


```
Amar@MSI MINGW64 ~/Downloads/Projects/fsddemo (master)
$ git branch -M main

Amar@MSI MINGW64 ~/Downloads/Projects/fsddemo (main)
$ git remote add origin https://github.com/amarkhakhkhar/fsddemogit.git

Amar@MSI MINGW64 ~/Downloads/Projects/fsddemo (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 252 bytes | 252.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/amarkhakhkhar/fsddemogit.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Amar@MSI MINGW64 ~/Downloads/Projects/fsddemo (main)
$ |
```



```
Amar@MSI MINGW64 ~/Downloads/Projects/fsddemo (main)
$ git add .

Amar@MSI MINGW64 ~/Downloads/Projects/fsddemo (main)
$ git commit -m "Second commit of Screenshots"
[main 3613407] Second commit of Screenshots
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 MINGW64__c_Users_Amar_Downloads_Projects_fsddemo 07-10-2023
23_19_42.png
 create mode 100644 MINGW64__c_Users_Amar_Downloads_Projects_fsddemo 07-10-2023
23_23_32.png

Amar@MSI MINGW64 ~/Downloads/Projects/fsddemo (main)
$ git push -u origin
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 75.86 KiB | 25.29 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/amarkhakhkhar/fsddemogit.git
```