Introduction to DevOps

Assignment -1

MASTER OF TECHNOLOGY

(SOFTWARE ENGINEERING)

Submitted By:

Pratiksha Dashrath Jadkar

2022MT93717

Date : 07 March 2023

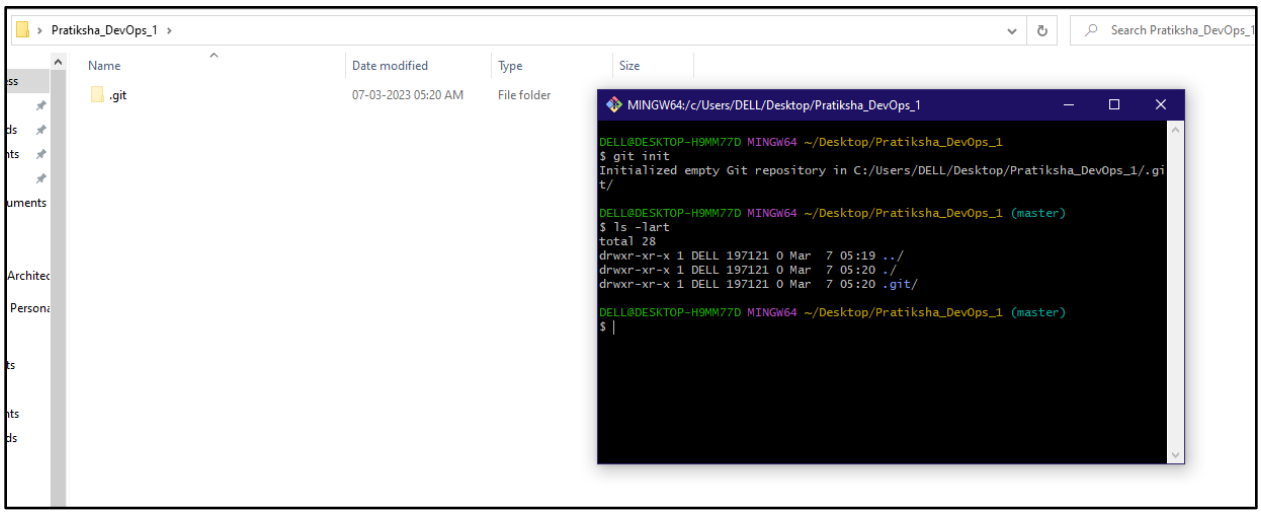
Problem Statement

ABC Organization would like to opt for the distributed version control system to upgrade their environment, where Git has been selected as the solution. You been assigned as a consultant to educate the migration process to move their Source Code from Centralized to Distributed systems. As a phase one, you would like to go ahead with a workshop to demonstrate below operation to make the ABC team comfortable.

1. Create a Repository
2. Add Two Directories and some raw code files to the repository
3. Move Code from One directory to Another Directory
4. Update one source code file and display the difference
5. Create a Branch
6. Add some raw code to the branch
7. Merge the Branch with Main line
8. PUSH to GitHub
9. Summary of advantages of moving from Centralized Source Code to Distributed Version Control.

1. Create a Repository

Command :
git init : Used to create a repository in the current directory.



Git creates a hidden directory called **.git** . This hidden **.git** directory stores all of the objects and refs that Git uses and creates as a part of your project's history . This **.git** directory separates a regular directory from a Git repository.

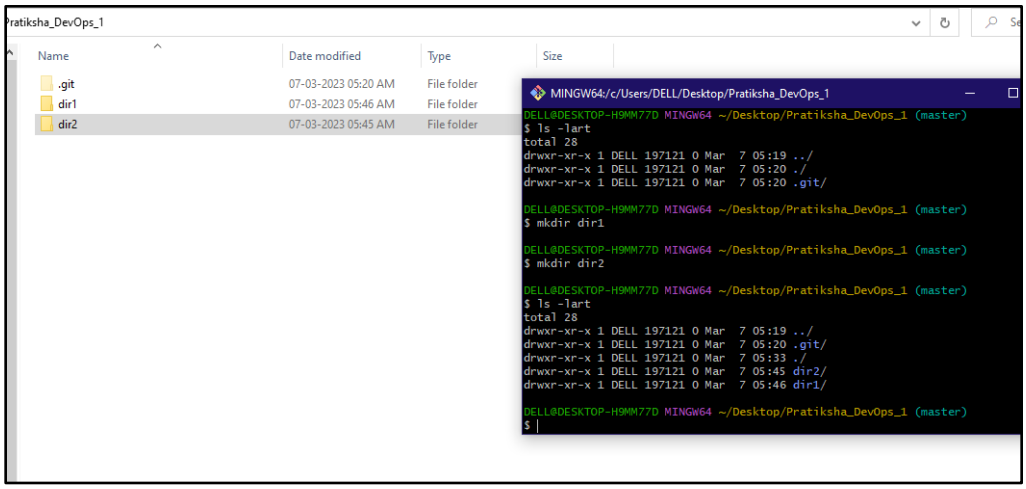
2. Add Two Directories and some raw code files to the repository

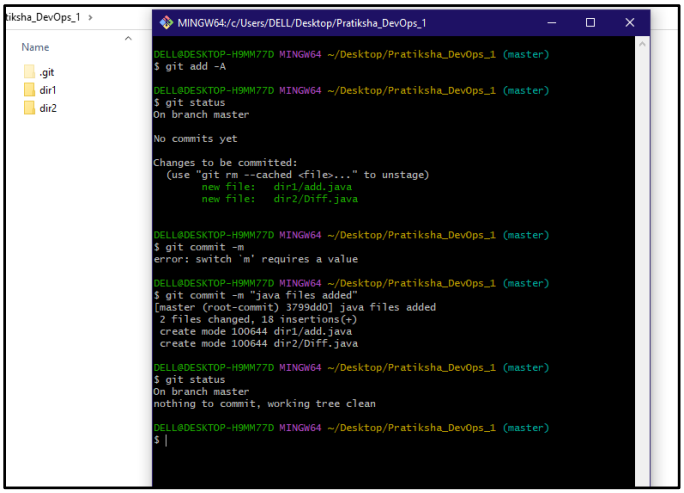
Created 2 directories **dir1** and **dir2.**

Added two java files add.java and diff.java in respective directories.

Commands :

- **git add** : used to add files to the staging area
- **git commit**:create a snapshot of the changes and save it to the git directory.
- **git status**: displays the list of changed files together with the files that are yet to be staged or committed.





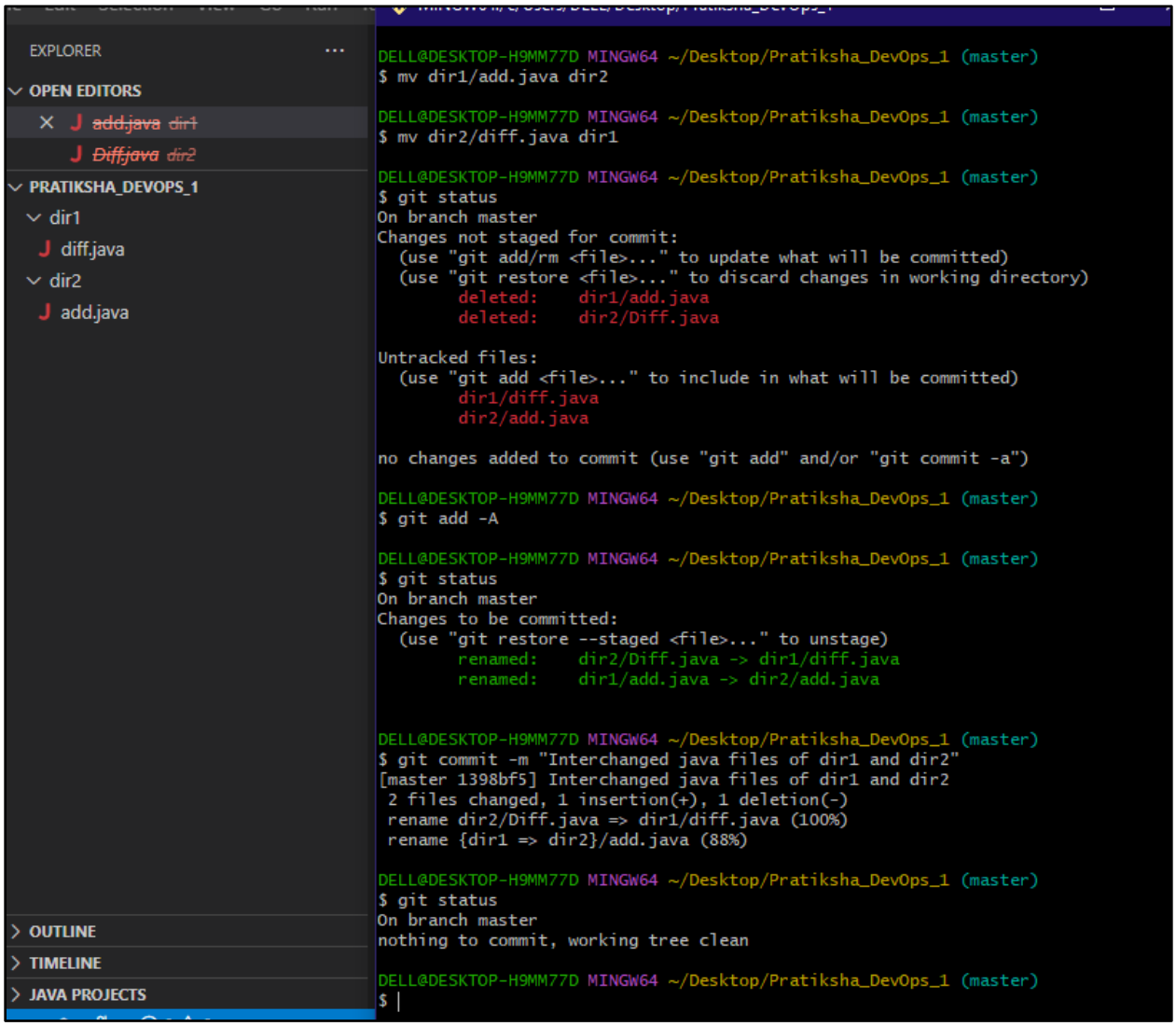
3. Move Code from One directory to Another Directory

Moved dir1 → add.java → dir2

And dir2 → diff.java → dir1

After Moving the files from one directory to another.

Run the commands **git add** to stage the files , **git commit -m "commit_message"** to commit



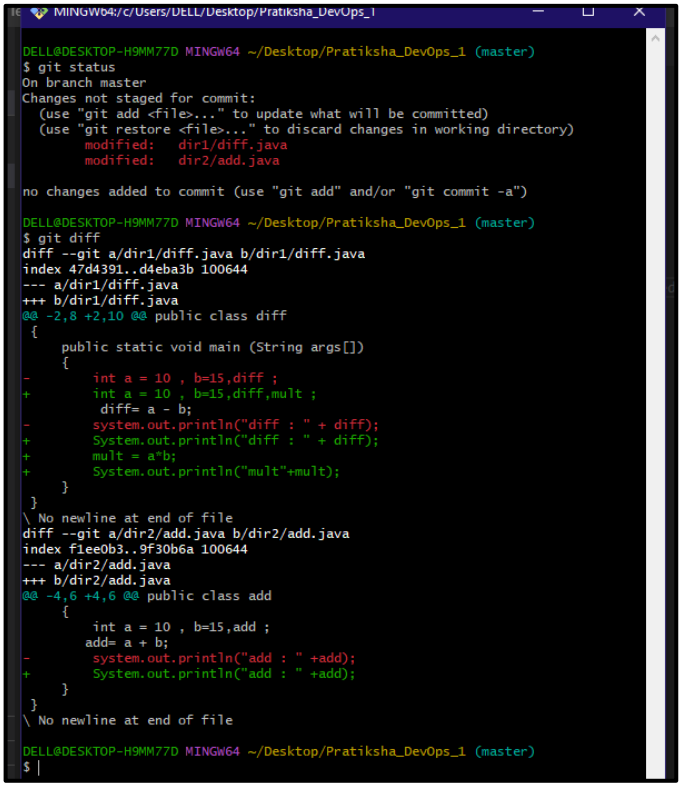
4. Update one source code file and display the difference

Command :

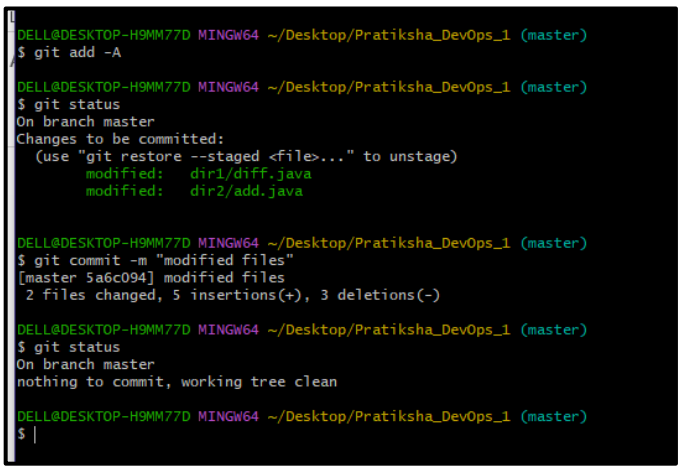
git diff : shows the changes between commits, commit and working tree, etc .

Updated both add.java and diff.java files.

Run **git status** to know which all files were modified and then **git diff** to check the difference.



Run **git add** , **git commit**



5. Create a Branch

In Git, a **Branch** is a new/separate version of the main repository.

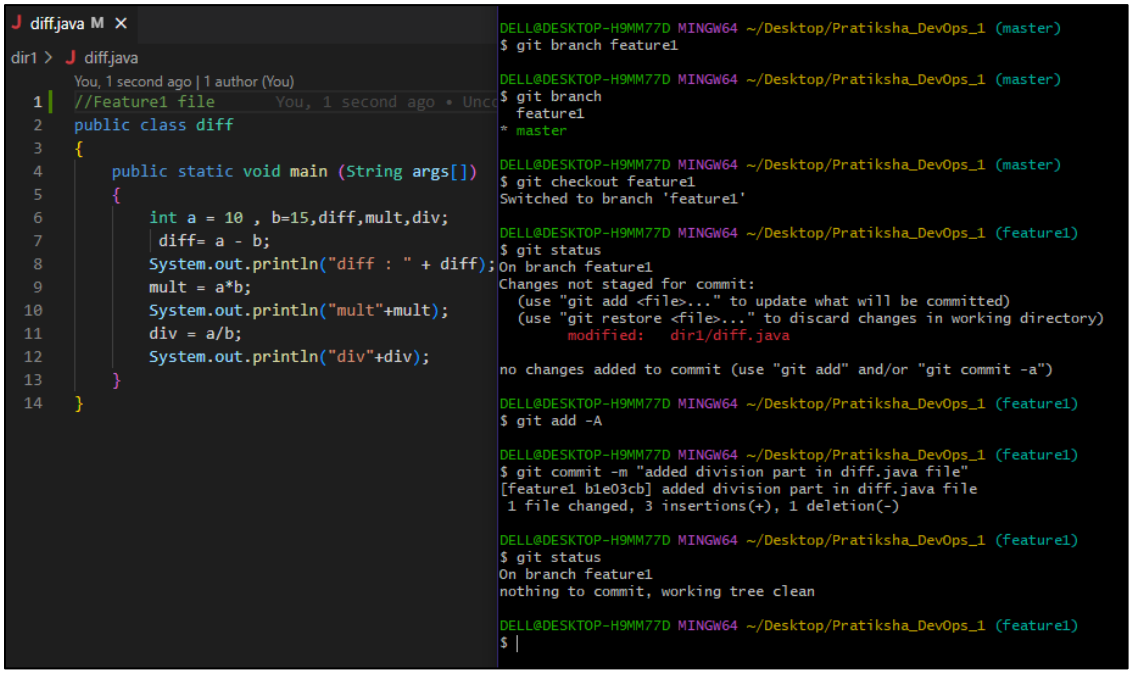
Commands :

git branch : used to create a new branch

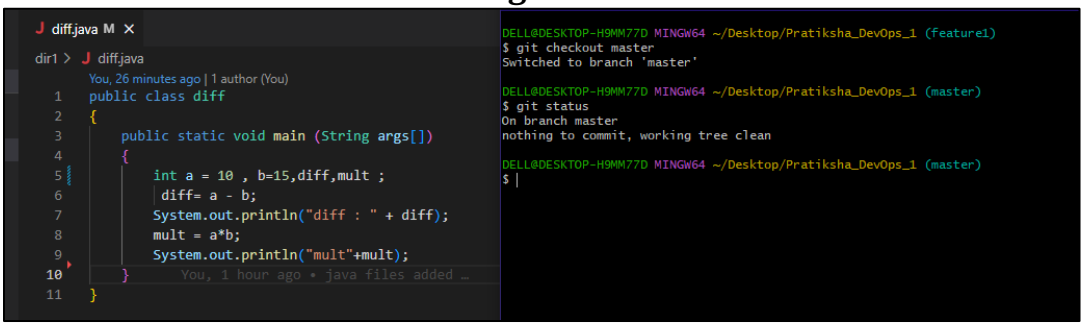
git checkout : used to switch to an existing branch.

By executing **git branch feature1** created feature1 branch

By default it's in master branch So to switch to feature1 branch and do changes run **git checkout feature1** . Once the changes are done do **git add** and **git commit**.

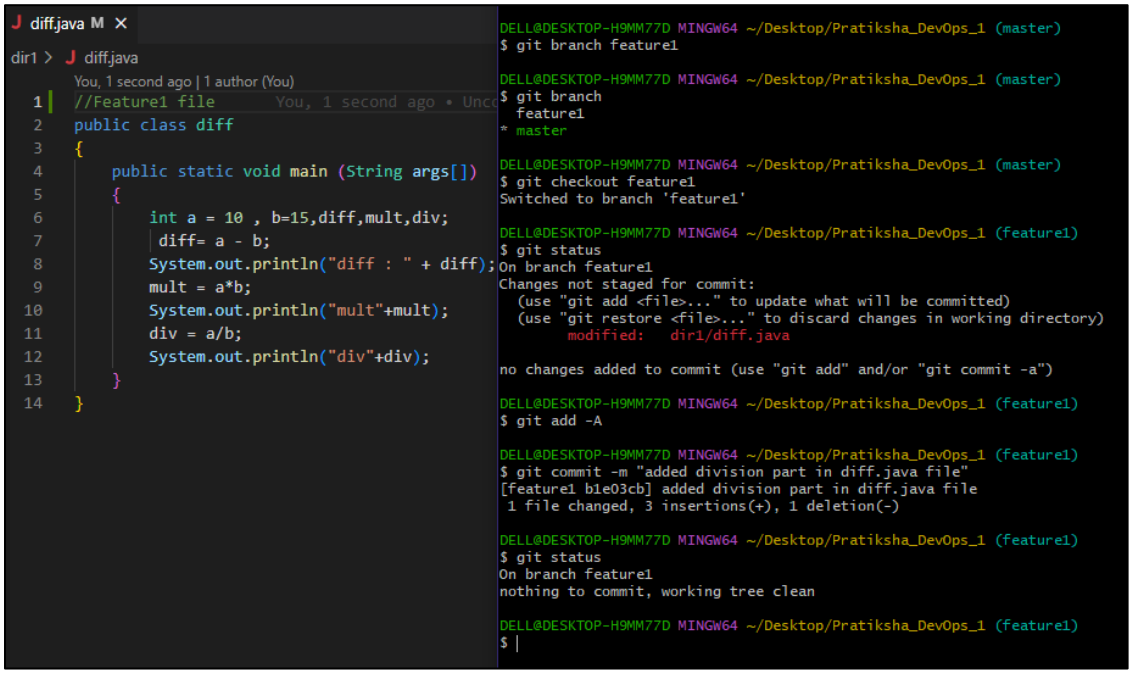


To switch back to master file run **git checkout master**



6. Add some raw code to the branch

Add Code in Feature1.java file (added division logic). Then do the **git add** and **git commit** before changing the branch to master



7. Merge the Branch with Main line

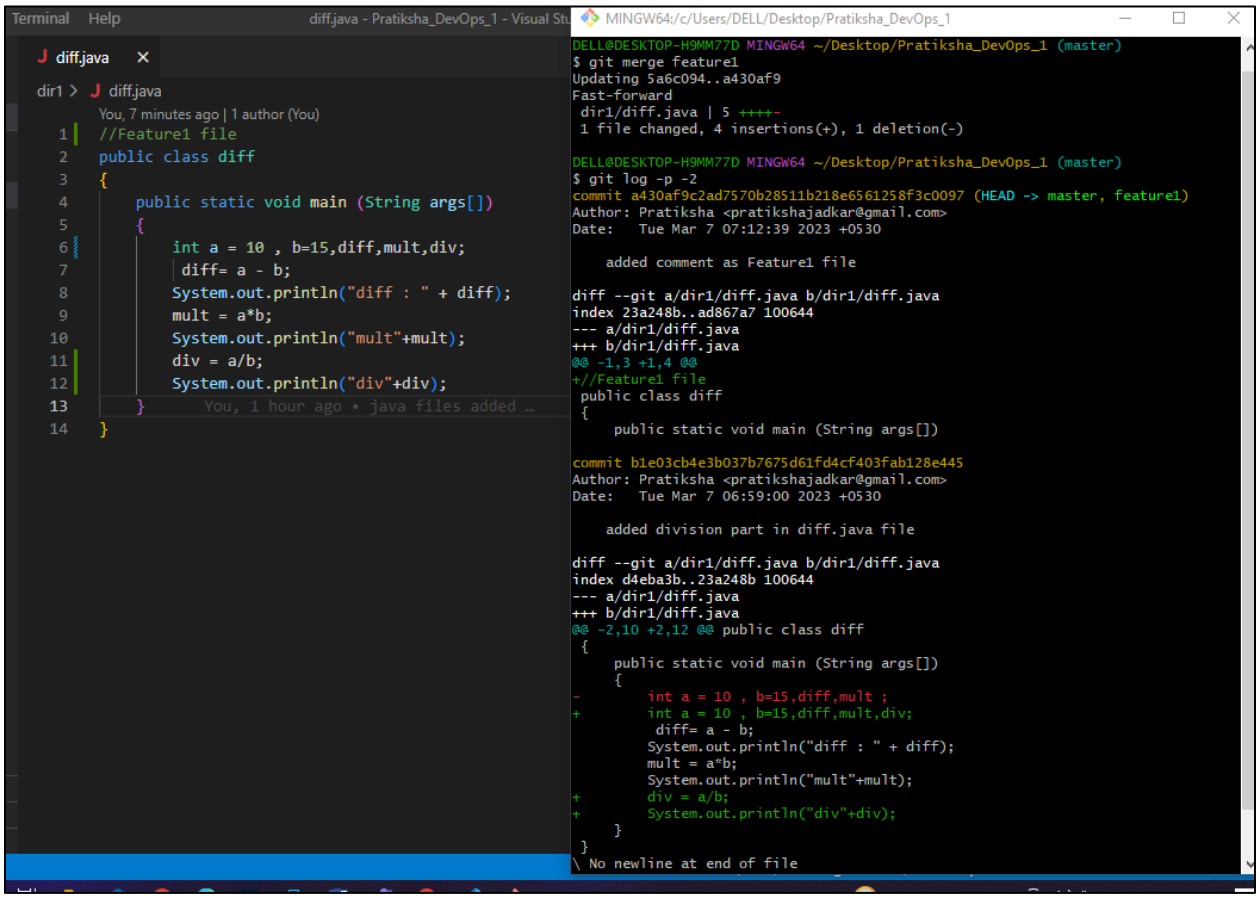
Commands :

git merge : used to merge feature file changes in master file.

git log –p -2 : Lists the commits made in that repository in reverse chronological order; that is, the most recent commits show up first. **-p** which shows the difference introduced in each commit. **-2** to show only the last two entries.

To merge feature1 file changes to master file run **git merge feature1**

Then run **git log –p -2** which will display all the feature file commits merged in master file.



8. PUSH to GitHub

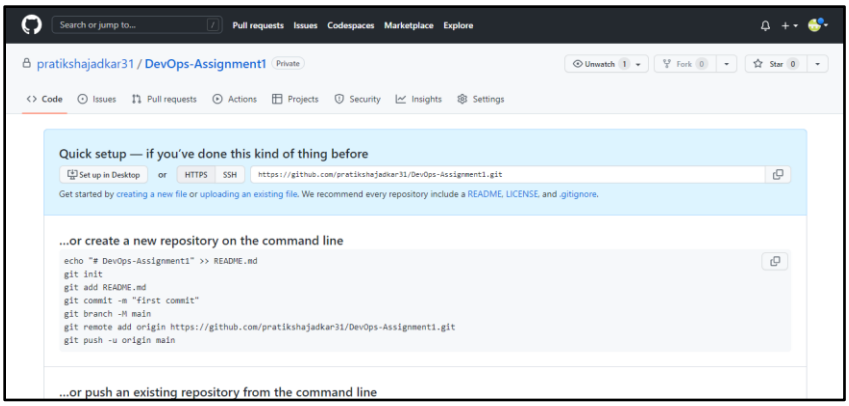
Commands:

git push: used to push local commits to a remote repository

git remote add : used to add a new remote.
It takes two arguments:

- A remote name, for example, origin
- A remote URL

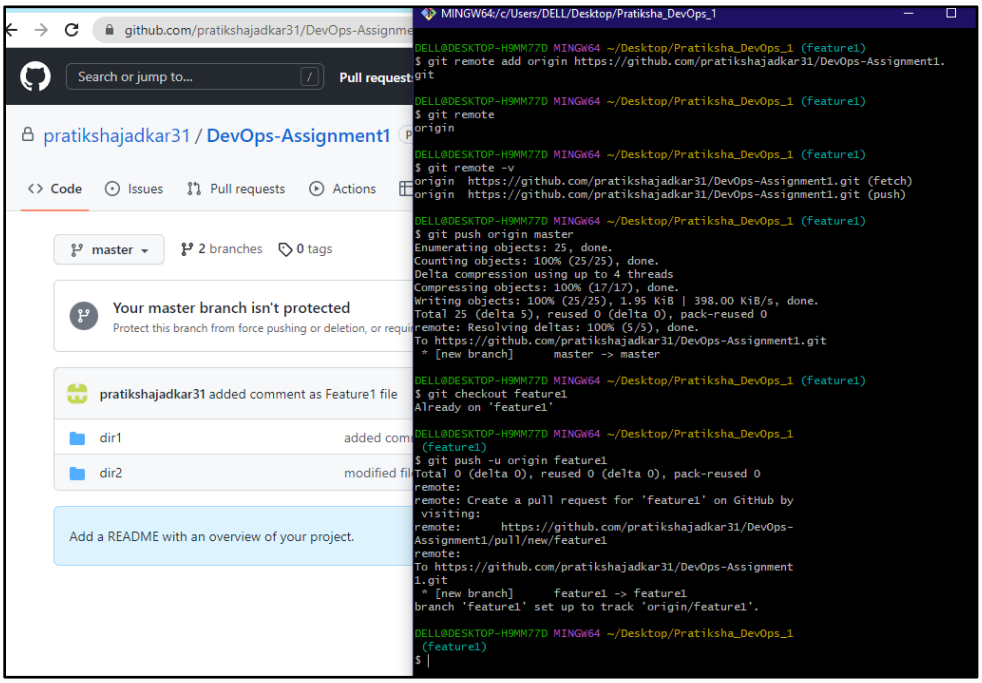
Create GitHub Account and then create New Repository



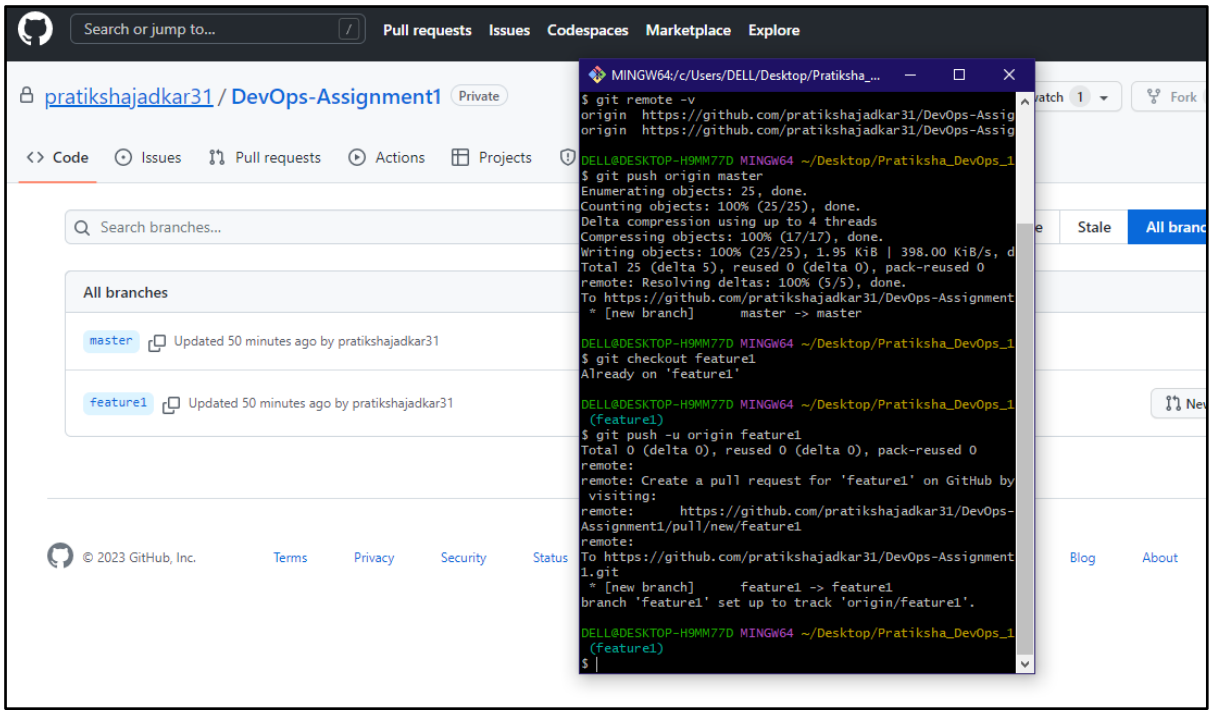
Now to push Local repository into Remote repository.

Copy the remote path and run that command

1st pushed master branch :



Push feature1 branch :



9. Advantages of moving from Centralized Source Code to Distributed Version Control.

- The full history is always available because of local commits
- No need to access a remote server (faster access)
- Each developer has a copy of the code on their system, even if the main server loses its copy, it can easily be recovered
- New feature has its own branch, which is tested and modified in isolation and later merged into the main branch once ready.
- Good for projects with off-shore developers