

SET UP GIT BRANCHING

AIM:

Create a new **branch** in your Git repository for testing. Add a new feature and **merge** it.

INTRODUCTION:

Git is a distributed version control system (VCS) used for tracking changes in code, collaborating on projects, and managing different versions of files efficiently. It is widely used in software development to help teams work together on the same codebase without conflicts.

Branching and Merging: Enables the creation of separate branches for new features or fixes, which can later be merged into the main project.

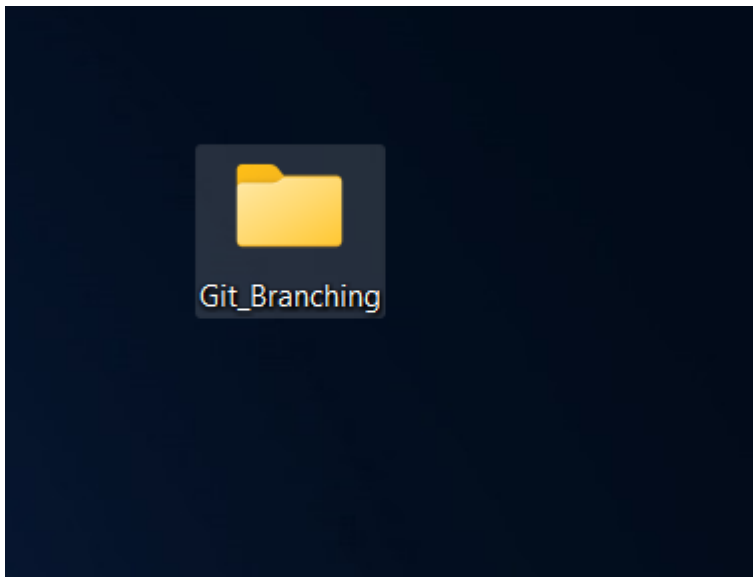
OBJECTIVES:

- To initialize and set up a Git repository.
- To create and manage feature branches.
- To demonstrate adding, committing and merging code.
- To showcase how to delete branches after their purpose is served.
- To learn how to resolve merge conflicts if any arise during the process.

STEP BY STEP OVERVIEW:

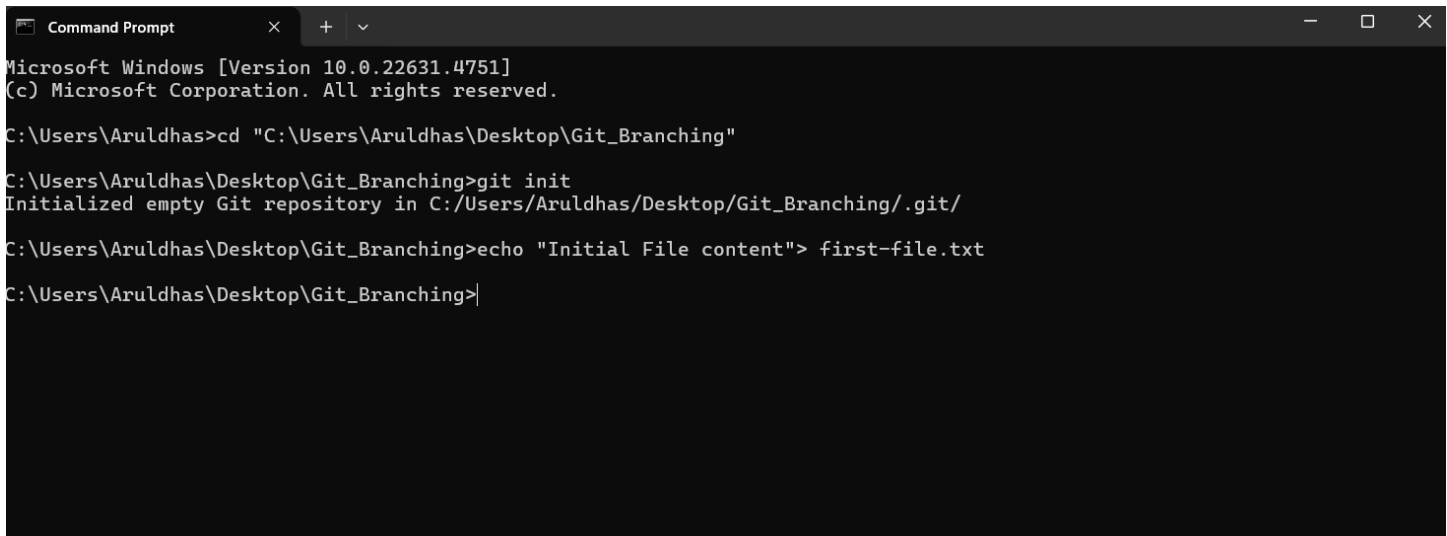
Step 1: CREATE A FOLDER

- Create a folder named 'Git_Branching' on your desktop.



Step 2: SETTING UP A PATH TO THE FOLDER

- Initialize Git by typing 'git init'.



```
Command Prompt
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

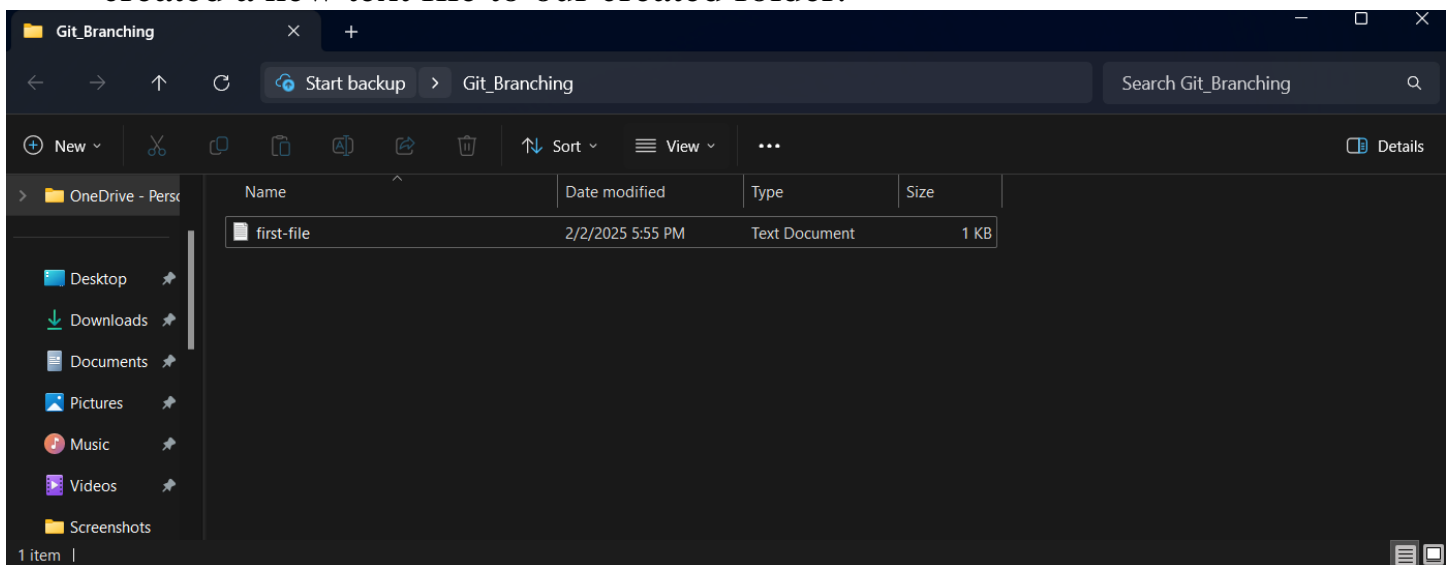
C:\Users\Arulldhas>cd "C:\Users\Arulldhas\Desktop\Git_Branching"

C:\Users\Arulldhas\Desktop\Git_Branching>git init
Initialized empty Git repository in C:/Users/Arulldhas/Desktop/Git_Branching/.git/

C:\Users\Arulldhas\Desktop\Git_Branching>echo "Initial File content"> first-file.txt

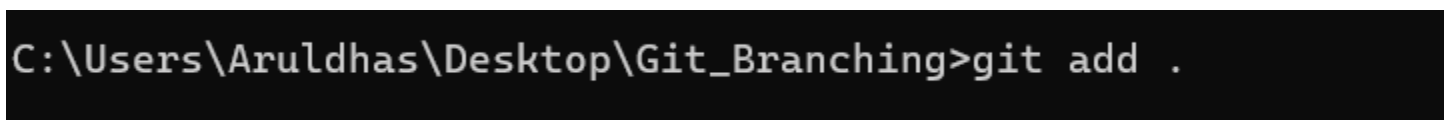
C:\Users\Arulldhas\Desktop\Git_Branching>|
```

- We will notice that the command 'echo "Initial File content"> first- file.txt' has created a new text file to our created folder.



Step 3: ADD FILE TO GIT

- Give the command 'git add.' To track the created file.



```
C:\Users\Arulldhas\Desktop\Git_Branching>git add .
```

- Save this change in Git with commit message.

```
C:\Users\Arulldhas\Desktop\Git_Branching>git commit -m "Initial Commit"
[master (root-commit) 99d617c] Initial Commit
1 file changed, 1 insertion(+)
create mode 100644 first-file.txt
```

Step 4: CREATE A BRANCH

- Create and switch to a new branch called 'testing-feature'

```
C:\Users\Arulldhas\Desktop\Git_Branching>git checkout -b testing-feature
Switched to a new branch 'testing-feature'
```

- Now add a new file from our feature.

```
C:\Users\Arulldhas\Desktop\Git_Branching>echo "update feature content"> first-file.txt
```

- Now save the changes.

```
C:\Users\Arulldhas\Desktop\Git_Branching>git add .
```

- Commit the changes.

```
C:\Users\Arulldhas\Desktop\Git_Branching>git commit -m "Update first-file.txt with new feature"
[testing-feature dd2b642] Update first-file.txt with new feature
1 file changed, 1 insertion(+), 1 deletion(-)
```

Step 6: SWITCH & MERGE TO MASTER BRANCH

- Give the command 'git checkout master' to switch it to the 'Master' branch.

```
C:\Users\Arulldhas\Desktop\Git_Branching>git checkout master
Switched to branch 'master'
```

- Now merge the branch we have just created to the master branch.

```
C:\Users\Arulldhas\Desktop\Git_Branching>git merge testing-feature
Updating 99d617c..dd2b642
Fast-forward
 first-file.txt | 2 + -
1 file changed, 1 insertion(+), 1 deletion(-)
```

Step 7: DELETE THE BRANCH

- Once the merge is done, you can delete the test-feature branch.
- For this, use 'git branch -d testing-feature' command.

```
C:\Users\Arulldhas\Desktop\Git_Branching>git branch -d testing-feature
Deleted branch testing-feature (was dd2b642).
```

Step 8: CHECK THE FINAL STATUS

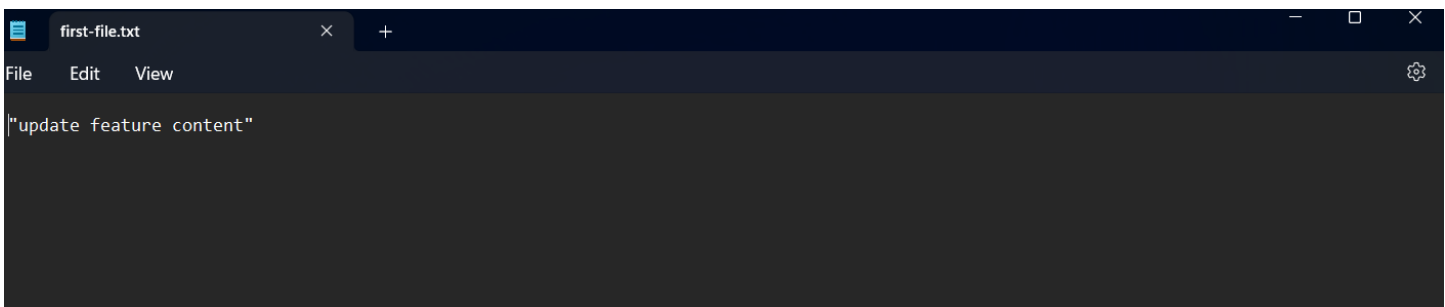
- Now check the files in the folder.

```
C:\Users\Arulldhas\Desktop\Git_Branching>dir
Volume in drive C is Windows-SSD
Volume Serial Number is 0CE8-95DB

Directory of C:\Users\Arulldhas\Desktop\Git_Branching

02/02/2025  06:15 PM    <DIR>          .
02/02/2025  05:45 PM    <DIR>          ..
02/02/2025  06:15 PM                26 first-file.txt
               1 File(s)                26 bytes
               2 Dir(s)  391,958,904,832 bytes free

C:\Users\Arulldhas\Desktop\Git_Branching>
```



CONCLUSION:

In this POC, we have gained knowledge on:

- Initializing a 'Git' repository in your local project folder.
- Create and manage multiple branches for development and experiment.
- Track and commit the changes made to files in various branches.
- Merge the feature branch to the main branch.
- Gain practical knowledge about the 'Git' commands.