

Placement empowering program

Cloud Computing and DevOps Centre

Create a new branch in your Git repository for testing.

Add a new feature and merge it.

Name: *Jeffersen Godfrey*

Department: CSE

INTRODUCTION:

In this task Git is used for version control to manage the development workflow. Git allows developers to create separate branches for new features, isolate them from the main branch and merge them back after completion.

OVERVIEW:

This POC demonstrates how to:

1. Initialize a Git repository.
2. Create and switch between branches.
3. Commit changes in different branches.
4. Merge feature branches into the main branch.
5. Delete branches after completing the work.

OBJECTIVES:

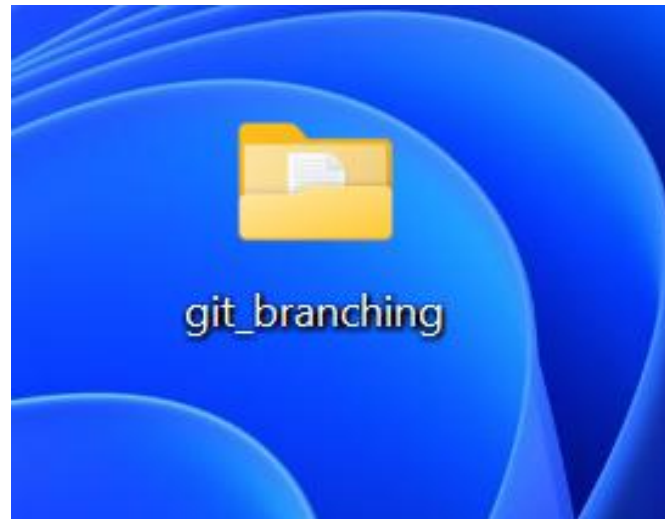
1. To initialize and set up a Git repository.
2. To create and manage feature branches (e.g., testing-feature).
3. To demonstrate adding, committing, and merging code.

4. To showcase how to delete branches after their purpose is served.
5. To learn how to resolve merge conflicts if any arise during the process.

IMPORTANCE:

- 1. Version Control:** Helps track changes, revert to previous versions, and avoid conflicts in the codebase.
- 2. Collaboration:** Different team members can work on separate features simultaneously without interfering with each other's work.
- 3. Branching:** Isolates new features or bug fixes, ensuring stability in the main branch (master or main).
- 4. Efficiency:** Merging branches allows rapid integration of new features without disrupting ongoing work.
- 5. Clean Workflow:** Deleting feature branches after merging keeps the repository clean and manageable.

Step 1: Create a new folder and give the name of your choice (Git_branching).



Step 2: Open command prompt and set path of the created folder.

```
Microsoft Windows [Version 10.0.26100.2894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jeffersen Godfrey>cd C:\Users\Jeffersen Godfrey\Desktop\git_branching
```

Step 3: Initialize Git by typing this command: **git init**

This command will create a git folder inside your folder, which tells Git to start tracking your files.

```
C:\Users\Jeffersen Godfrey\Desktop\git_branching>git init
Initialized empty Git repository in C:/Users/Jeffersen Godfrey/Desktop/git_branching/.git/
```

Step 4: Create a simple file to start the repository:

```
C:\Users\Jeffersen Godfrey\Desktop\git_branching>echo "Initial file content" > first-file.txt
```

Step 5: Give **git add .** to add the files to Git

```
C:\Users\Jeffersen Godfrey\Desktop\git_branching>git add .
```

Step 6: Save it on Git and use a commit message

```
C:\Users\Jeffersen Godfrey\Desktop\git_branching>git commit -m "initial commit"
[master (root-commit) f964493] initial commit
1 file changed, 1 insertion(+)
create mode 100644 first-file.txt
```

Step 7: Create and then switch to a new branch called testing-feature.

```
C:\Users\Jeffersen Godfrey\Desktop\git_branching>git checkout -b testing-feature
Switched to a new branch 'testing-feature'
```

Step 8: We'll add a new file for our feature:

```
C:\Users\Jeffersen Godfrey\Desktop\git_branching>echo "initial file content" > first-file.txt
```

Step 9: Now, stage the changes

```
C:\Users\Jeffersen Godfrey\Desktop\git_branching>git add .
```

Step 10: Give commit command again to commit the changes.

```
C:\Users\Jeffersen Godfrey\Desktop\git_branching>git commit -m " Add new feature file"
[testing-feature ca96e6f] Add new feature file
1 file changed, 1 insertion(+), 1 deletion(-)
```

Step 11: Then, switch to master branch.

```
C:\Users\Jeffersen Godfrey\Desktop\git_branching>git checkout master
Switched to branch 'master'
```

Step 12: Switch to master branch

```
C:\Users\Jeffersen Godfrey\Desktop\git_branching>git merge testing-feature
Updating f964493..ca96e6f
Fast-forward
 first-file.txt | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

Step 13: Once the merge is done, you can delete the testing-feature branch:

```
C:\Users\Jeffersen Godfrey\Desktop\git_branching>git branch -d testing-feature
Deleted branch testing-feature (was ca96e6f).
```

Step 14: Look for files in the folder

```

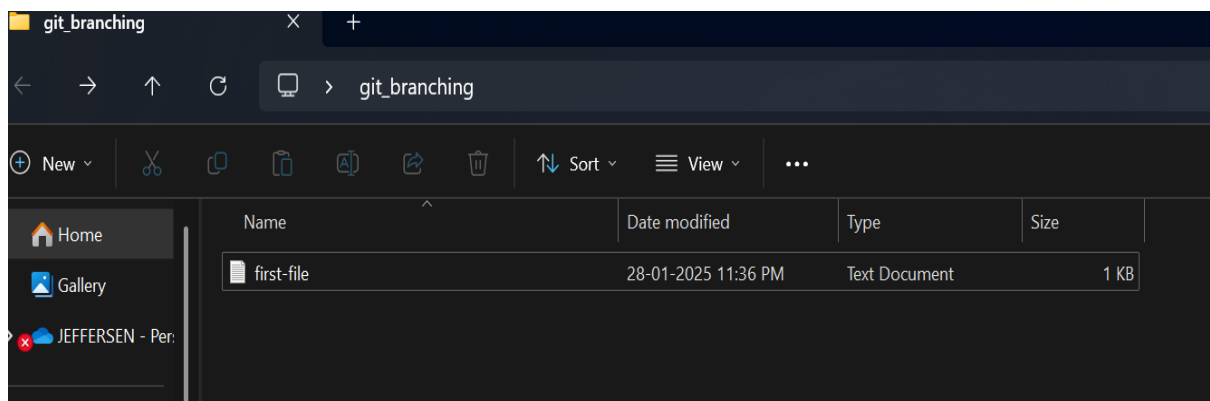
C:\Users\Jeffersen Godfrey\Desktop\git_branching>dir
Volume in drive C is OS
Volume Serial Number is 1273-2A3A

Directory of C:\Users\Jeffersen Godfrey\Desktop\git_branching

28-01-2025  11:36 PM    <DIR>          .
28-01-2025  11:22 PM    <DIR>          ..
28-01-2025  11:36 PM                25 first-file.txt
                1 File(s)                25 bytes
                2 Dir(s)  83,970,936,832 bytes free

C:\Users\Jeffersen Godfrey\Desktop\git_branching>

```



OUTCOME:

By completing this task of managing branches in Git for a local repository, you will:

1. Successfully initialize a Git repository in your local project folder.

2. Create and manage multiple branches for feature development and experimentation.
3. Track and commit changes made to files in different branches.
4. Merge feature branches back into the main branch while maintaining project integrity.
5. Gain hands-on experience with key Git commands such as `git init`, `git add`, `git commit`, `git checkout`, and `git merge`.