

Placement Empowerment Program

Cloud Computing and DevOps Centre

Set Up Git Branching: Create a new branch in your Git repository for testing .

Add a new feature and merge it

Name: Sandhana Jefrina J

Department: CSE

Introduction:

In this Proof of Concept (POC), Git is utilized for version control to streamline the development process. It enables developers to work on new features in separate branches, keeping them independent from the main branch. Once the features are complete, they can be merged back, promoting structured and collaborative development

Step-by-Step Overview

Step 1:

Create a folder and name it (poc-5).



Step 2:

Set the path to the folder created in first step (Git_Branching).

```
C:\Users\Jeffrina>cd C:\Users\Jeffrina\Desktop\POC 5  
C:\Users\Jeffrina\Desktop\POC 5>
```

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\Jeffrina\Desktop\POC 5>  
C:\Users\Jeffrina\Desktop\POC 5>git init  
Initialized empty Git repository in C:/Users/Jeffrina/Desktop/POC 5/.git/
```

Step 4:

Create a simple file to start the repository:

```
C:\Users\Jeffrina\Desktop\POC 5>echo "Intial file content" > first-file.txt
```

Step 5:

Add the File to Git

```
C:\Users\Jeffrina\Desktop\POC 5>git add .
```

Tell Git to track this file:

Step 6:

Save this change in Git with a commit message.

```
C:\Users\Jeffrina\Desktop\POC 5>git commit -m "Initial commit"
[master (root-commit) ca297c6] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 first-file.txt
```

Step 7:

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

```
C:\Users\Jeffrina\Desktop\POC 5>
C:\Users\Jeffrina\Desktop\POC 5>git checkout -b testing-feature switched to a new branch 'testing-feature'
fatal: Cannot update paths and switch to branch 'testing-feature' at the same time.
```

Step 8:

Let's add a new file for our feature:

```
C:\Users\Jeffrina\Desktop\POC 5>echo "Initial file content" > first-file.txt
```

Step 9:

Now, stage the changes:

```
C:\Users\Jeffrina\Desktop\POC 5>git add .
```

Step 10:

Commit the changes:

```
C:\Users\Jeffrina\Desktop\POC 5>git add .  
C:\Users\Jeffrina\Desktop\POC 5>git commit -m "Add new feature file"  
[master 946a8b5] Add new feature file  
1 file changed, 1 insertion(+), 1 deletion(-)
```

Step 11:

Switch to the master Branch

```
C:\Users\Jeffrina\Desktop\POC 5>git checkout master  
Already on 'master'
```

Step 12:

Merge Changes from testing-feature to master

```
C:\Users\Jeffrina\Desktop\POC 5>git merge testing-feature
```

Step 13:

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

```
C:\Users\Jeffrina\Desktop\POC 5>git branch -d testing-feature
```

Step 14:

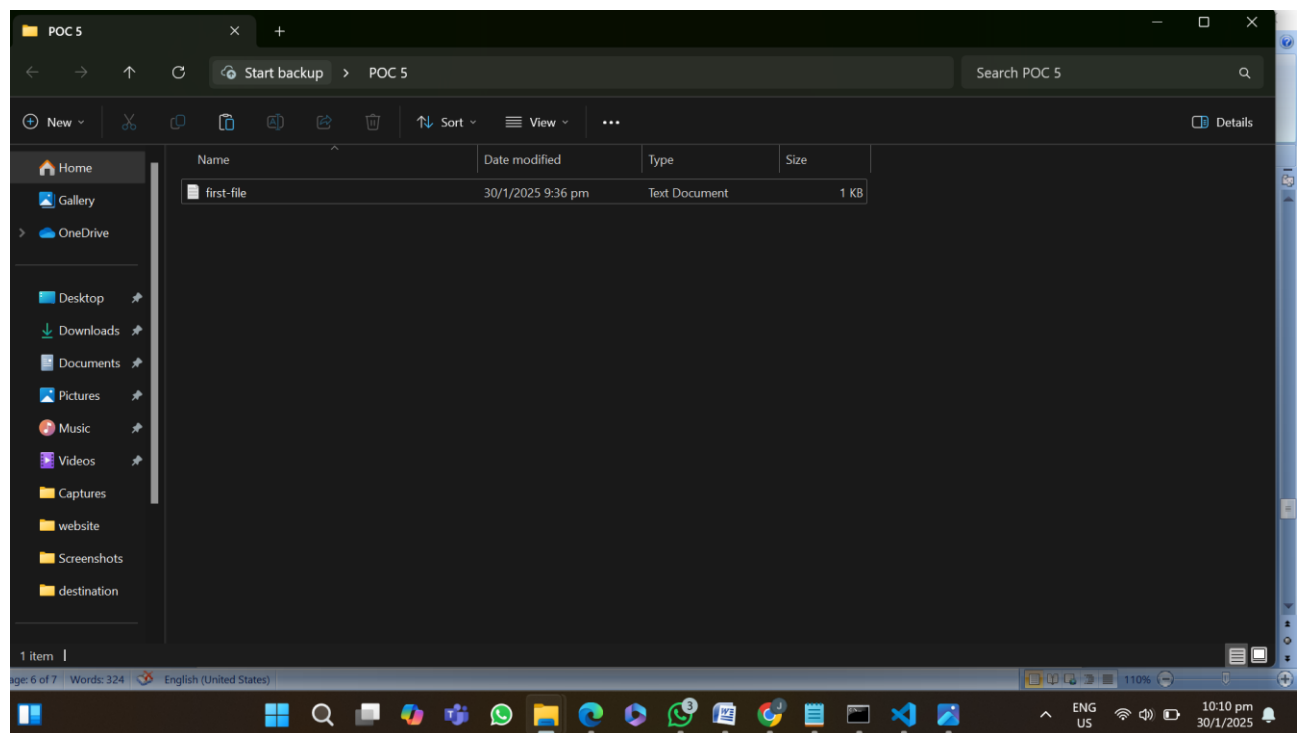
Now, check the files in the folder:

```
C:\Users\Jeffrina\Desktop\POC 5>dir
Volume in drive C is Windows-SSD
Volume Serial Number is B051-D686

Directory of C:\Users\Jeffrina\Desktop\POC 5

30/01/2025  09:32 pm    <DIR>          .
30/01/2025  08:46 pm    <DIR>          ..
30/01/2025  09:36 pm                25 first-file.txt
               1 File(s)                25 bytes
               2 Dir(s)  357,320,388,608 bytes free

C:\Users\Jeffrina\Desktop\POC 5>
```



Outcome

By completing this PoC 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`.