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: Jenith Melkeena R M Department: CSE

Introduction:

In this Proof of Concept (POC), 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. This ensures organized 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\jenit>cd C:\Users\jenit\OneDrive\Desktop\poc-5

C:\Users\jenit\OneDrive\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\jenit\OneDrive\Desktop\poc-5>git init
Initialized empty Git repository in C:/Users/jenit/OneDrive/Desktop/poc-5/.git/
```

Step 4:

```
C:\Users\jenit\OneDrive\Desktop\poc-5>echo "Initial file conent" > first-file.txt

C:\Users\jenit\OneDrive\Desktop\poc-5>
```

Create a simple file to start the repository:

Step 5:

Add the File to Git

```
C:\Users\jenit\OneDrive\Desktop\poc-5>git add .

C:\Users\jenit\OneDrive\Desktop\poc-5>
```

Tell Git to track this file:

Step 6:

Save this change in Git with a commit message.

```
C:\Users\jenit\OneDrive\Desktop\poc-5>git commit -m "Initial commit"
[master (root-commit) baa86d8] Initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 first-file.txt


C:\Users\jenit\OneDrive\Desktop\poc-5>
```

Step 7:

```
C:\Users\jenit\OneDrive\Desktop\poc-5>git checkout -b testing-feature
Switched to a new branch 'testing-feature'
```

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

Step 8:

Let’s add a new file for our feature:

```
C:\Users\jenit\OneDrive\Desktop\poc-5>echo "Initial file content" >first-file.txt

C:\Users\jenit\OneDrive\Desktop\poc-5>
```

Step 9:

Now, stage the changes:

```
C:\Users\jenit\OneDrive\Desktop\poc-5>git add .

C:\Users\jenit\OneDrive\Desktop\poc-5>
```

Step 10:

Commit the changes:

```
C:\Users\jenit\OneDrive\Desktop\poc-5>git commit -m "Add new feature file"
On branch testing-feature
nothing to commit, working tree clean
```

Step 11:

Switch to the master Branch

```
C:\Users\jenit\OneDrive\Desktop\poc-5>git checkout master
Switched to branch 'master'
```

Step 12:

Merge Changes from testing-feature to master

```
C:\Users\jenit\OneDrive\Desktop\poc-5>git merge testing-feature
Already up to date.
```

Step 13:

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

```
C:\Users\jenit\OneDrive\Desktop\poc-5>git branch -d testing-feature
Deleted branch testing-feature (was 8c4bfdc).
```

Step 14:

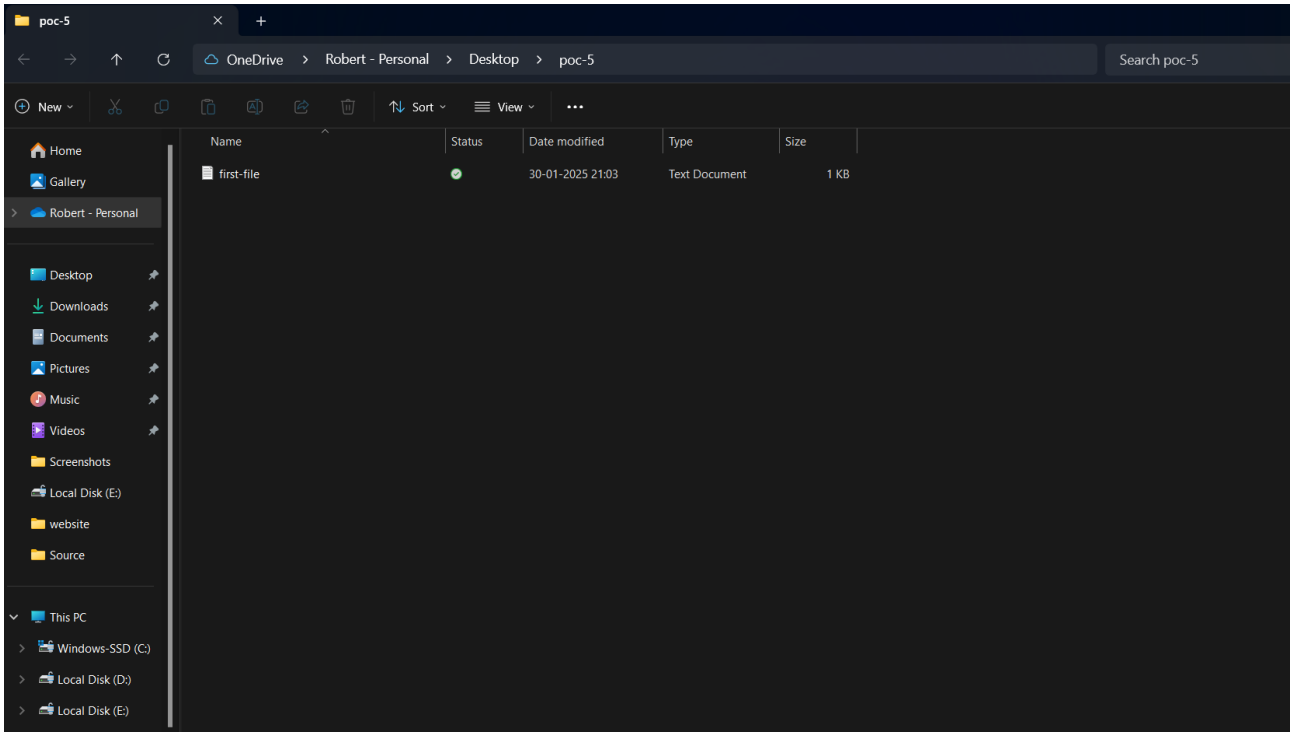
Now, check the files in the folder:

```
C:\Users\jenit\OneDrive\Desktop\poc-5>dir
 Volume in drive C is Windows-SSD
 Volume Serial Number is D6FB-B424

 Directory of C:\Users\jenit\OneDrive\Desktop\poc-5

30-01-2025  21:03    <DIR>          .
30-01-2025  19:56    <DIR>          ..
30-01-2025  21:03                25 first-file.txt
               1 File(s)             25 bytes
               2 Dir(s)  19,739,471,872 bytes free

C:\Users\jenit\OneDrive\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.