St. JOSEPH'S GROUP OF INSTITUTIONS
OMR, CHENNAI - 119

aws

Placement Empowerment Program

Cloud Computing and DevOps Centre

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

Name: DEFIN TN Department:CSE

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY
AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

Introduction:

In this Proof of Concept (POC), Git is utilized for version control to streamline the development process. Git enables developers to create separate branches for new features, keeping them isolated from the main branch. Once the new features are completed, they can be merged back into the main branch. This approach fosters organized, collaborative development and maintains a clean, stable codebase throughout the project's lifecycle.

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).

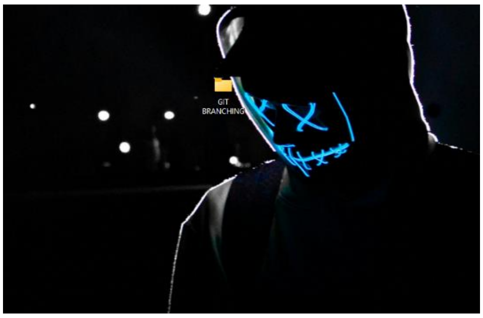
4. **Efficiency:** Merging branches allows rapid integration of new features without disrupting ongoing work.

5. **Clean Workflow:** Deleting feature branches after merging keeps the repository clean and manageable.

Step-by-Step Overview

Step 1:

Create a folder and name it (Git_Branching).



Step 2:

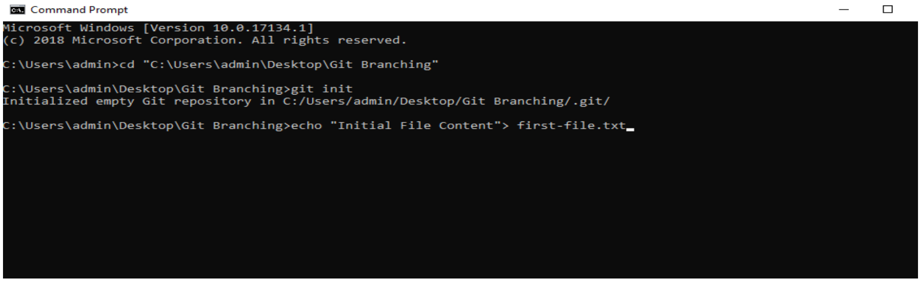
Set the path to the folder created in first step (Git_Branching).

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.

Create a simple file to start the repository:

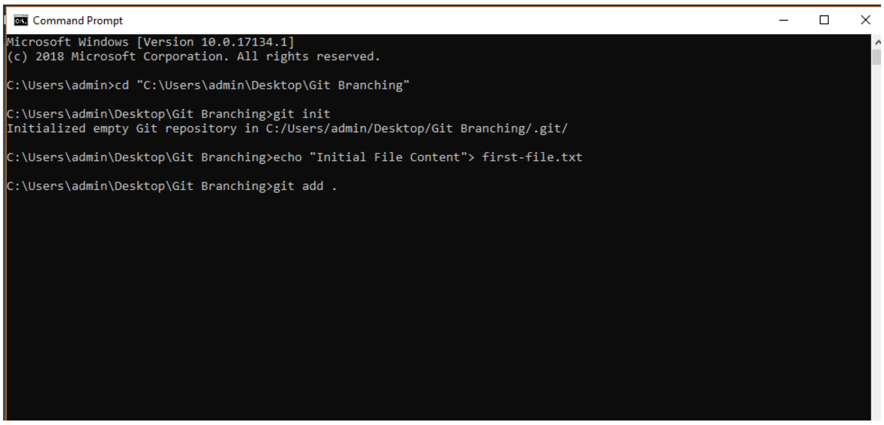


```
Microsoft Windows [Version 10.0.17134.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\admin>cd "C:\Users\admin\Desktop\Git Branching"

C:\Users\admin\Desktop\Git Branching>git init
Initialized empty Git repository in C:/Users/admin/Desktop/Git Branching/.git/

C:\Users\admin\Desktop\Git Branching>echo "Initial File Content"> first-file.txt
```

Step 4:

Add the File to Git

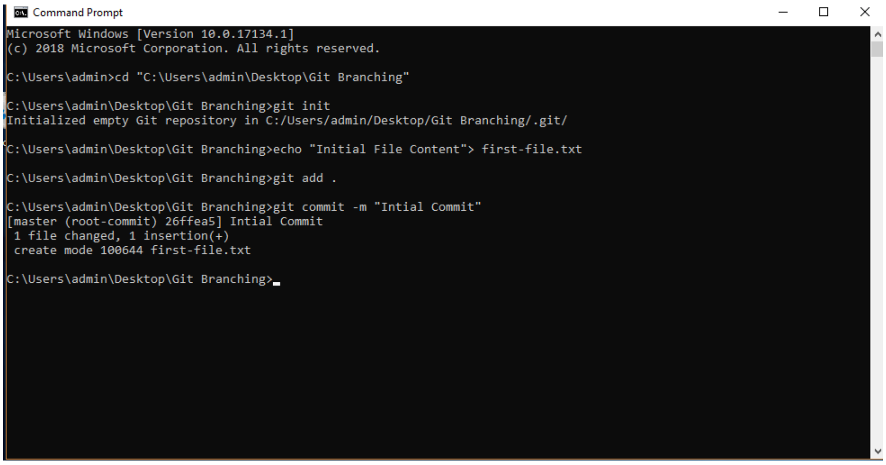
Tell Git to track this file:



```
Microsoft Windows [Version 10.0.17134.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\admin>cd "C:\Users\admin\Desktop\Git Branching"

C:\Users\admin\Desktop\Git Branching>git init
Initialized empty Git repository in C:/Users/admin/Desktop/Git Branching/.git/

C:\Users\admin\Desktop\Git Branching>echo "Initial File Content"> first-file.txt

C:\Users\admin\Desktop\Git Branching>git add .
```

Step 5:

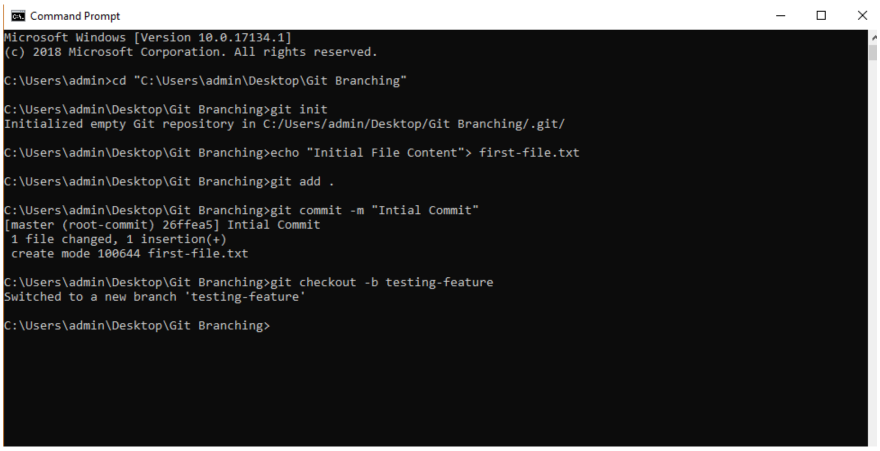
Save this change in Git with a commit message.



```
Microsoft Windows [Version 10.0.17134.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\admin>cd "C:\Users\admin\Desktop\Git Branching"

C:\Users\admin\Desktop\Git Branching>git init
Initialized empty Git repository in C:/Users/admin/Desktop/Git Branching/.git/

C:\Users\admin\Desktop\Git Branching>echo "Initial File Content"> first-file.txt

C:\Users\admin\Desktop\Git Branching>git add .

C:\Users\admin\Desktop\Git Branching>git commit -m "Intial Commit"
[master (root-commit) 26ffea5] Intial Commit
 1 file changed, 1 insertion(+)
 create mode 100644 first-file.txt

C:\Users\admin\Desktop\Git Branching>
```

Step 6:

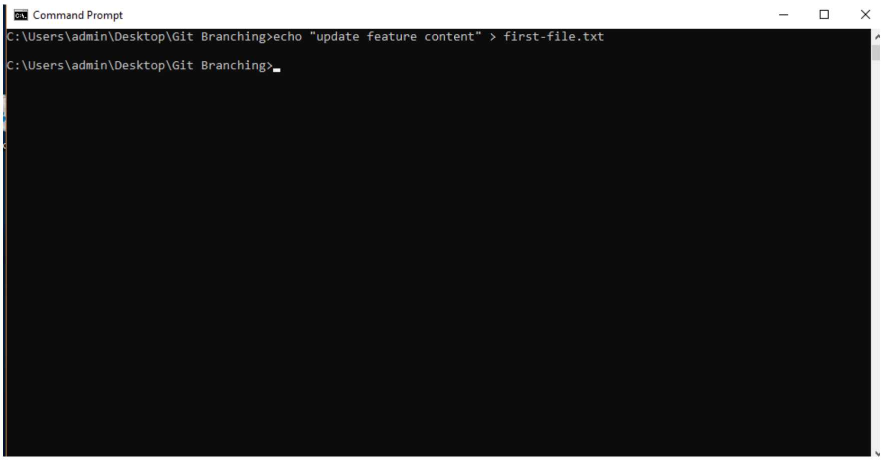
Create and switch to a new branch called testing-feature.



```
Microsoft Windows [Version 10.0.17134.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\admin>cd "C:\Users\admin\Desktop\Git Branching"

C:\Users\admin\Desktop\Git Branching>git init
Initialized empty Git repository in C:/Users/admin/Desktop/Git Branching/.git/

C:\Users\admin\Desktop\Git Branching>echo "Initial File Content"> first-file.txt

C:\Users\admin\Desktop\Git Branching>git add .

C:\Users\admin\Desktop\Git Branching>git commit -m "Intial Commit"
[master (root-commit) 26ffea5] Intial Commit
 1 file changed, 1 insertion(+)
 create mode 100644 first-file.txt

C:\Users\admin\Desktop\Git Branching>git checkout -b testing-feature
Switched to a new branch 'testing-feature'

C:\Users\admin\Desktop\Git Branching>
```

Step 7:

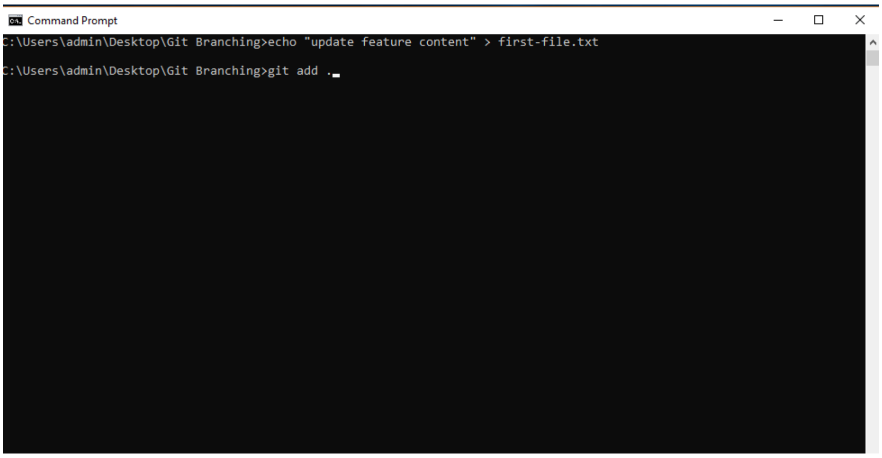
Let’s add a new file for our feature:



```
C:\Users\admin\Desktop\Git Branching>echo "update feature content" > first-file.txt

C:\Users\admin\Desktop\Git Branching>
```

Step 8:

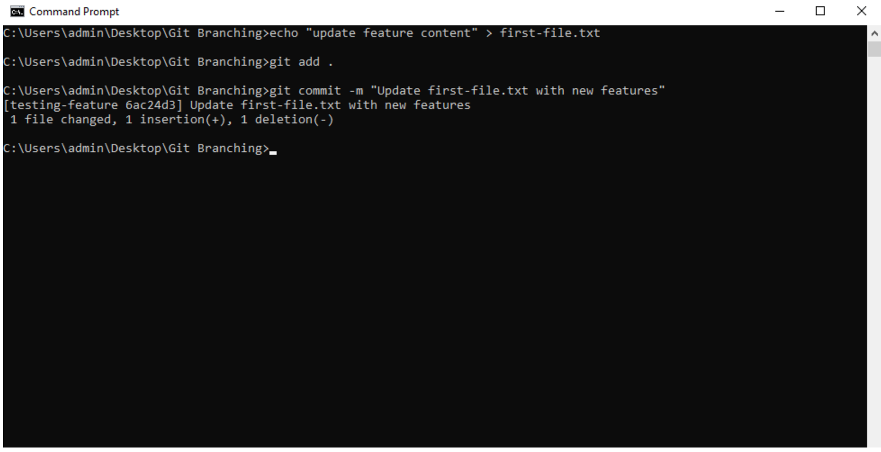
Now, stage the changes:



```
C:\Users\admin\Desktop\Git Branching>echo "update feature content" > first-file.txt

C:\Users\admin\Desktop\Git Branching>git add .
```

Step 9:

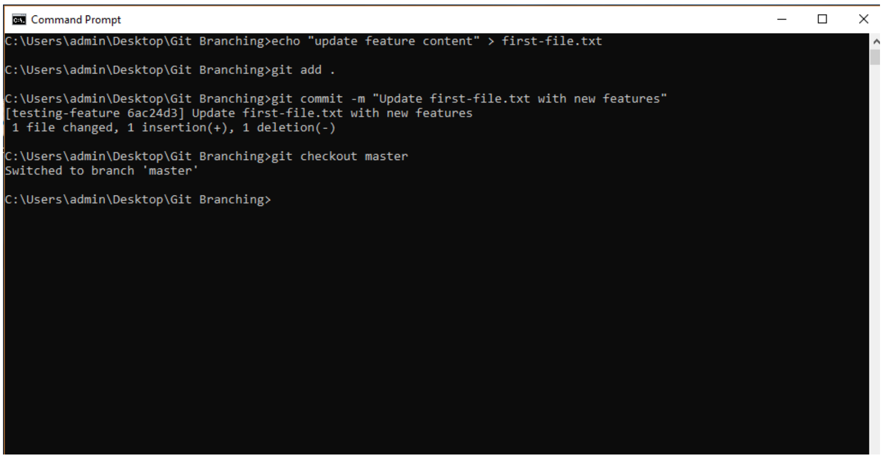
Commit the changes:



```
C:\Users\admin\Desktop\Git Branching>echo "update feature content" > first-file.txt

C:\Users\admin\Desktop\Git Branching>git add .

C:\Users\admin\Desktop\Git Branching>git commit -m "Update first-file.txt with new features"
[testing-feature 6ac24d3] Update first-file.txt with new features
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\admin\Desktop\Git Branching>
```

Step 10:

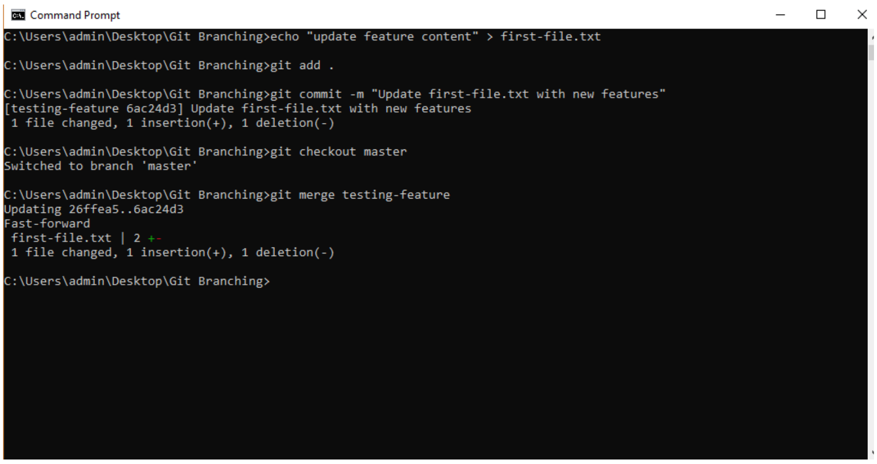
Switch to the master Branch



```
C:\Users\admin\Desktop\Git Branching>echo "update feature content" > first-file.txt

C:\Users\admin\Desktop\Git Branching>git add .

C:\Users\admin\Desktop\Git Branching>git commit -m "Update first-file.txt with new features"
[testing-feature 6ac24d3] Update first-file.txt with new features
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\admin\Desktop\Git Branching>git checkout master
Switched to branch 'master'

C:\Users\admin\Desktop\Git Branching>
```

Step 11:

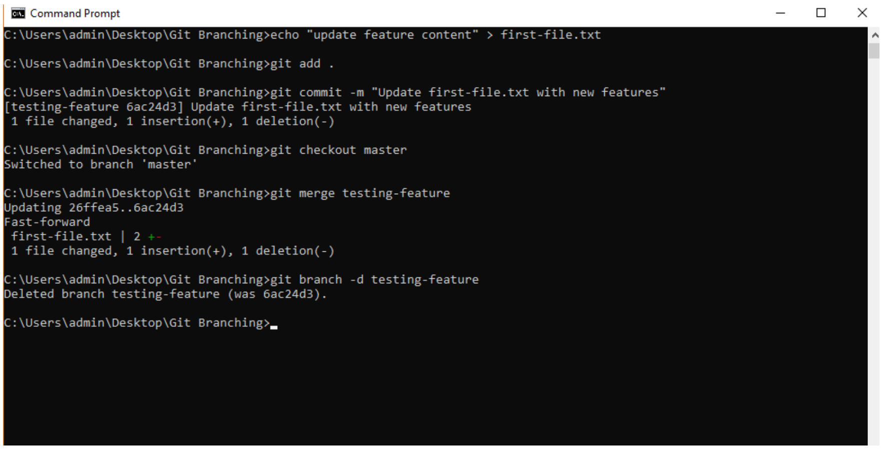
Merge Changes from testing-feature to master



```
C:\Users\admin\Desktop\Git Branching>echo "update feature content" > first-file.txt

C:\Users\admin\Desktop\Git Branching>git add .

C:\Users\admin\Desktop\Git Branching>git commit -m "Update first-file.txt with new features"
[testing-feature 6ac24d3] Update first-file.txt with new features
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\admin\Desktop\Git Branching>git checkout master
Switched to branch 'master'

C:\Users\admin\Desktop\Git Branching>git merge testing-feature
Updating 26ffea5..6ac24d3
Fast-forward
 first-file.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\admin\Desktop\Git Branching>
```

Step 12:

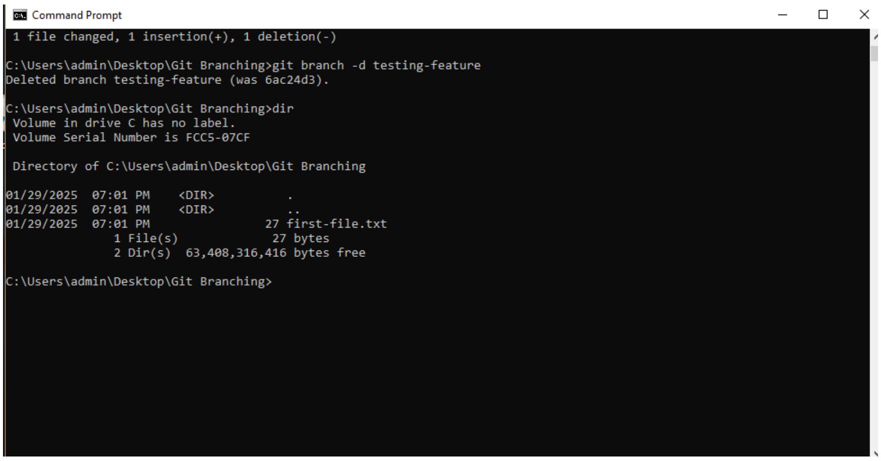
Once the merge is done, you can delete the testing-feature branch:



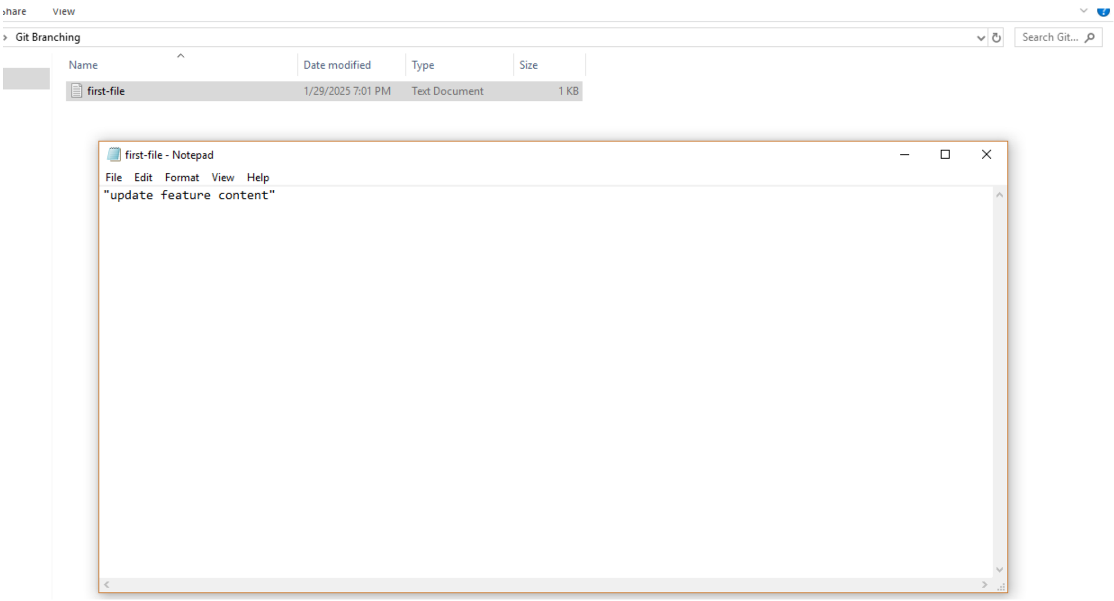
```
C:\Users\admin\Desktop\Git Branching>echo "update feature content" > first-file.txt

C:\Users\admin\Desktop\Git Branching>git add .

C:\Users\admin\Desktop\Git Branching>git commit -m "Update first-file.txt with new features"
[testing-feature 6ac24d3] Update first-file.txt with new features
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\admin\Desktop\Git Branching>git checkout master
Switched to branch 'master'

C:\Users\admin\Desktop\Git Branching>git merge testing-feature
Updating 26ffea5..6ac24d3
Fast-forward
 first-file.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\admin\Desktop\Git Branching>git branch -d testing-feature
Deleted branch testing-feature (was 6ac24d3).

C:\Users\admin\Desktop\Git Branching>
```

Step 13:

Now, check the files in the folder:



```
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\admin\Desktop\Git Branching>git branch -d testing-feature
Deleted branch testing-feature (was 6ac24d3).

C:\Users\admin\Desktop\Git Branching>dir
 Volume in drive C has no label.
 Volume Serial Number is FCC5-07CF

 Directory of C:\Users\admin\Desktop\Git Branching

01/29/2025  07:01 PM    <DIR>          .
01/29/2025  07:01 PM    <DIR>          ..
01/29/2025  07:01 PM                27 first-file.txt
               1 File(s)             27 bytes
               2 Dir(s)  63,408,316,416 bytes free

C:\Users\admin\Desktop\Git Branching>
```



Name	Date modified	Type	Size
first-file	1/29/2025 7:01 PM	Text Document	1 KB

```
"update feature content"
```

Outcome

By completing this PoC on managing branches in Git for a local repository, you will:

- Initialize a Git repository in your local project folder.
- Create and manage multiple branches for feature development and experimentation.
- Track and commit changes made to files in various branches.
- Merge feature branches back into the main branch while preserving project integrity.
- Gain practical experience with essential Git commands such as `git init`, `git add`, `git commit`, `git checkout`, and `git merge`.