



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



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: Harshana Perianayaki B

Department : IT



INTRODUCTION

Git is a powerful version control system that enables developers to track changes, collaborate efficiently, and manage code effectively. One of Git's essential features is branching, which allows multiple developers to work on different features simultaneously without affecting the main project.

OBJECTIVES

The objective of this guide is to help you:

Create a new branch in a Git repository.

Add a new feature to the branch.

Merge the branch back into the main branch.

Understand the importance of using Git branches for version control.

OVERVIEW

Git branches provide an isolated workspace for new development, allowing developers to test features, fix bugs, and implement enhancements without modifying the stable codebase. This process involves:

Creating a new branch for testing.

Switching to the new branch and making necessary changes.

Committing the changes to the branch.

Merging the branch back into the main branch after successful testing.

Deleting the branch (optional) to keep the repository clean.

IMPORTANCE OF GIT BRANCHING

- **Encourages Parallel Development** – Developers can work on multiple features simultaneously.
- **Prevents Code Conflicts** – Changes made in separate branches do not interfere with the main branch.
- **Facilitates Code Reviews & Testing** – Teams can review changes before merging them into production.
- **Enables Rollbacks** – If a feature causes issues, it can be reverted without affecting the main branch.

STEP-BY-STEP OVERVIEW

Here's how you can **set up Git branching**, create a new branch, add a feature, and merge it into the main branch.

Step 1: Initialize Git (If Not Already Initialized)

If you haven't set up a Git repository, initialize it:

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\harsh>git init
Reinitialized existing Git repository in C:/Users/harsh/.git/
```

Step 2: Create a New Branch

First, check your current branch:

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\harsh> git init
Reinitialized existing Git repository in C:/Users/harsh/.git/
PS C:\Users\harsh> git branch
PS C:\Users\harsh> |
```

Then, create a new branch for testing:

```
PS C:\Users\harsh> git branch feature-branch
```

Switch to the new branch:

```
PS C:\Users\harsh> git checkout feature-branch
```

(Or use `git switch feature-branch` in newer Git versions.)

Step 3 : Add a new feature

Modify your files or add new ones, then stage and commit the changes:

```
PS C:\Users\harsh> git add .
```

```
PS C:\Users\harsh> git commit -m "Added new feature"
warning: could not open directory 'Application Data/': Permission denied
warning: could not open directory 'Cookies/': Permission denied
warning: could not open directory 'Documents/My Music/': Permission denied
warning: could not open directory 'Documents/My Pictures/': Permission denied
warning: could not open directory 'Documents/My Videos/': Permission denied
warning: could not open directory 'Local Settings/': Permission denied
warning: could not open directory 'My Documents/': Permission denied
warning: could not open directory 'NetHood/': Permission denied
warning: could not open directory 'PrintHood/': Permission denied
warning: could not open directory 'Recent/': Permission denied
warning: could not open directory 'SendTo/': Permission denied
warning: could not open directory 'Start Menu/': Permission denied
warning: could not open directory 'Templates/': Permission denied
On branch feature-branch

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)
```

Step 4 : Merge the feature branch into main

Switch back to the main branch:

```
PS C:\Users\harsh> git checkout main
```

Merge the feature branch:

```
PS C:\Users\harsh> git merge feature-branch
```

Step 5: Clean Up (Optional)

Once merged, you can delete the feature branch:

```
PS C:\Users\harsh> git branch -d feature-branch
```

Step 6 : Push Changes (If using Remote Repository)

If you're working with a remote repository (like GitHub or GitLab), push your changes:

```
PS C:\Users\harsh> git push origin main
```

successfully created a branch, added a feature, and merged it.

OUTCOME

By the end of this process, you will:

- Understand and implement Git branching efficiently.
- Work on features in a **safe and isolated** environment.
- Merge changes into the main branch **without conflicts**.
- Improve **collaboration and workflow** in Git-based projects.

