 

**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: S.KRITHIKA Department : ADS



**Introduction and Overview**

Git is a distributed version control system that enables developers to track changes in their code, collaborate efficiently, and manage different versions of a project. One of Git’s powerful features is branching, which allows developers to work on new features or bug fixes without affecting the main project.

Branching enables parallel development, facilitates testing, and helps maintain a clean and organized codebase. Once a new feature is developed and tested in a separate branch, it can be merged into the main branch to integrate the changes seamlessly.

**Objective**

The objective of this guide is to demonstrate how to:

1. Create a new branch in a Git repository for testing purposes.
2. Add a new feature in the newly created branch.
3. Merge the branch back into the main branch.
4. Understand the importance of local hosting in Git workflows.

By following this guide, users will gain hands-on experience with Git branching, merging, and local repository management.

**Importance of Local Hosting in Git**

Local hosting in Git refers to managing a repository on a local machine before pushing changes to a remote repository. This practice offers several benefits:

* **Offline Development:** Developers can work on their projects without needing an internet connection.
* **Faster Operations:** Local commits, branching, and merging operations are much quicker than their remote counterparts.
* **Safe Testing Environment:** Changes can be tested locally before being shared with a team, reducing errors in the main repository.
* **Version Control & Backup:** Local repositories act as a backup before syncing with remote repositories like GitHub, GitLab, or Bitbucket.

**Step-by-Step Overview**

1. **Initialize or Clone a Git Repository**
2. git init # Initialize a new repository

git clone <repository\_url> # Clone an existing repository

1. **Create a New Branch for Testing**

git checkout -b feature-branch # Create and switch to a new branch

1. **Add a New Feature**
   * Modify or create a file
   * Stage and commit changes
2. git add newfile.txt

git commit -m "Added a new feature"

1. **Switch Back to the Main Branch**

git checkout main # Switch back to the main branch

1. **Merge the Feature Branch into Main**

git merge feature-branch # Merge changes

1. **Push Changes to the Remote Repository (if applicable)**

git push origin main # Push changes to the remote repository

1. **Delete the Feature Branch (Optional)**

git branch -d feature-branch # Delete branch after merging

**Expected Outcome**

After following these steps, users will have:

* Created and worked within a new Git branch.
* Successfully added and committed a new feature.
* Merged the feature branch into the main branch.
* Understood the role and benefits of local hosting in Git workflows.