 

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

Set up a virtual machine in the cloud: Create a free tier AWS, ASURE, or GCP account .Launch a virtual machine and SSH into it

Name: S.KRITHIKA Department: ADS



**Introduction and Overview**

Git is a widely used version control system that allows developers to track changes in their code, collaborate efficiently, and maintain different versions of a project. One of the essential features of Git is branching, which enables developers to work on different features or bug fixes independently without affecting the main codebase. Local hosting, on the other hand, allows developers to test their applications in a controlled environment before deploying them to a live server.

This document will guide you through the process of setting up a Git branch, adding a new feature, merging it into the main branch, and understanding the importance of local hosting.

**Objectives**

* Understand the concept of Git branching and merging.
* Learn how to create a new branch and add a feature.
* Explore the significance of local hosting for testing and development.
* Follow a step-by-step approach to setting up Git and hosting locally.

**Importance of Local Hosting**

Local hosting is crucial for software development as it allows developers to:

* Test new features and bug fixes without affecting the live environment.
* Debug and optimize applications efficiently.
* Ensure a smooth deployment process by identifying and resolving potential issues beforehand.
* Improve collaboration by enabling team members to work on different parts of the project independently.

**Step-by-Step Overview**

**Step 1: Initialize a Git Repository**

If you haven’t already set up a Git repository, initialize it using the following commands:

mkdir my\_project

cd my\_project

git init

**Step 2: Create a New Branch**

To create a new branch for testing, use the following command:

git checkout -b feature-branch

This command creates and switches to a new branch named feature-branch.

**Step 3: Add a New Feature**

Make changes to the project, such as adding a new file:

echo "print('New feature added')" > feature.py

git add feature.py

git commit -m "Added a new feature"

**Step 4: Merge the Feature Branch**

Switch back to the main branch and merge the feature branch:

git checkout main

git merge feature-branch

**Step 5: Test Locally Using Local Hosting**

For web applications, you can use a local server to test changes before deployment. If using Python, start a simple HTTP server:

python -m http.server 8000

For Node.js applications:

npm start

**Step 6: Push Changes to a Remote Repository (Optional)**

If you are working with a remote repository, push the changes:

git push origin main

**Expected Outcome**

* A new branch is created and successfully merged into the main branch.
* The feature is added and tested in a local environment.
* Developers can work independently without interfering with the main codebase.
* The changes are ready for further development or deployment.