 

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

Set Up a Local Git Repository

Name: Monika K Department: ADS



**Objective**

Initialize a Git repository locally and version control your static website.

**Introduction**

Version control is essential for managing changes in software development. Git, a widely used version control system, helps track updates, collaborate efficiently, and revert changes when needed. In this Proof of Concept (PoC), we will initialize a local Git repository for a static website, allowing structured project management and easy collaboration.

**Overview**

1. **Installing Git**: Ensure Git is installed and properly configured.
2. **Creating a Local Repository**: Initialize a Git repository in the static website’s root folder.
3. **Staging and Committing Files**: Add files to Git tracking and commit changes.
4. **Reviewing Repository State**: Use Git commands to check the repository status.

**Objectives**

By the end of this PoC, you will:

1. Understand the basics of version control.
2. Set up a Git repository locally.
3. Track file changes efficiently.
4. Maintain a structured workflow for your static website.
5. Prepare for future collaboration with Git.

**Importance of Setting Up a Local Git Repository**

* **Track Changes**: Maintain a clear history of modifications.
* **Rollback**: Easily revert to previous versions when needed.
* **Collaboration**: Prepare for seamless teamwork.

**Step-by-Step Guide**

**Step 1: Install Git**

1. Search for "Git" online and download it from the official site.
2. Select the **Windows** option and complete the installation wizard.

**Step 2: Create a Project Folder**

1. On your desktop, create a folder named website.
2. Inside the website folder, create a simple index.html file with basic HTML content.

**Step 3: Open Command Prompt**

1. Navigate to the website folder using the command:

cd path\to\website

**Step 4: Initialize Git**

1. Run the following command to initialize Git:

git init

This command creates a .git folder, indicating that Git is tracking the project.

**Step 5: Track Files in Git**

1. To add all files to tracking, use:

git add .

**Step 6: Configure Git User Details**

1. Set up your name and email globally:
2. git config --global user.name "Your Name"

git config --global user.email "your.email@example.com"

**Step 7: Commit Your Changes**

1. Save the changes with a commit message:

git commit -m "Initial commit of my static website"

**Step 8: Create a Remote Repository on GitHub**

1. Log in to GitHub and click the **New** button.
2. Name your repository (e.g., my-website).
3. Click **Create repository** (keep other settings as default).

**Step 9: Link Local Repository to GitHub**

1. Add the remote URL to Git:

git remote add origin https://github.com/yourusername/my-website.git

*(Replace yourusername with your GitHub username and my-website with your repository name.)*

**Step 10: Rename Default Branch to Main**

1. Run the following command:

git branch -M main

**Step 11: Push Code to GitHub**

1. Upload files to GitHub:

git push -u origin main

**Step 12: Verify on GitHub**

1. Open your GitHub repository.
2. Confirm that your index.html file appears in the repository.