

Assignment day 5

Assignment 1: Initialize a new Git repository in a directory of your choice. Add a simple text file to the repository and make the first commit.

Step 1: Initialize a New Git Repository

1. Open Terminal (or Command Prompt)
2. Navigate to your desired directory
3. Initialize the Git repository

Git init

This command creates a new subdirectory name `.git` that contains all of your necessary repository files. This also makes the current directory the working directory.

Step 2: Add a Simple Text File

1. Create a new text file: Rohit.txt
This command creates a new file named Rohit.txt and writes "This is a Rohit text file." into it. Alternatively, you can use a text editor to create and edit the file.
2. Check the status: `git status`
This command shows the status of the repository and lists Rohit.txt as an untracked file.

Step 3: Stage and Commit the File

1. Stage the file:
`git add . Rohit.txt` This command stages Rohit.txt, preparing it for the next commit.
2. Commit the file:
`git commit -m "new commit"`
This command commits the staged file to the repository with the message "Add simple text file as the first commit".

Assignment 2: Branch Creation and Switching

Create a new branch named 'feature' and switch to it. Make changes in the 'feature' branch and commit them.

Step 1: Create and Switch to the 'feature' Branch

1. Create a new branch named 'feature':
`git branch feature`
2. Switch to the 'feature' branch:
`git checkout feature`
Alternatively, you can create and switch to the new branch in one command: `git checkout -b feature`

Step 2: Make Changes in the 'feature' Branch

1. Make some changes: Edit the Rohit.txt to add some new content. You can do this using a text editor or from the command line.
Using the command line, you can add new content like this: `>> Rohit.txt`
2. Check the status:

git status

This command will show that Rohit.txt has been modified.

Step 3: Stage and Commit the Changes

1. Stage the modified file: `Git add Rohit.txt`
2. Commit the changes: `Git commit -m "update Rohit.txt.in feature"`

Assignment 3: Feature Branches and Hotfixes

Create a 'hotfix' branch to fix an issue in the main code. Merge the 'hotfix' branch into 'main' ensuring that the issue is resolved.

Step 1: Create and Switch to the 'hotfix' Branch

1. Ensure you are on the 'main' branch:
`Git branch feature git checkout main`
2. Create a new branch named 'hotfix' and switch to it:
`Git checkout feature git checkout -b hotfix`

Step 2: Make and Commit the Hotfix

1. Make the necessary changes to fix the issue:
Edit the simplefile.txt or any other file to apply the hotfix. For example:>> Rohit.txt
2. Check the status:
`git checkout -b feature`
3. Stage the modified file:
`it add Rohit.txt`
4. Commit the changes:
`Git commit -m`

Step 3: Merge the 'hotfix' Branch into 'main'

1. Switch back to the 'main' branch:
`git checkout main`
2. Merge the 'hotfix' branch into 'main':
`git merge hotfix`

Step 4: Verify the Merge

1. Check the status and log to ensure the merge was successful:
`Git status`
`Git log --online git status`
Ensure that the commit from the 'hotfix' branch is present in the 'main' branch.