Assignment-Day-6

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.

Solution:

Here is a step-by-step procedure to initialize a new Git repository, add a simple text file, and make the file commit:

Step 1: Open Terminal

Open your terminal application

Step 2: Navigate to the Desired Directory

Use the cd command to navigate to the directory where you want to initialize your Git repository. For example **cd/path/to/your/directory**

Step 3: Initialize the Git Repository

Initialize a new Git repository in the current directory using the git init command:

git init

This command creates a new subdirectory named .git that contains all the necessary metadata for the Git repository.

Step 4: Create a Simple Text File

Create a simple text file in the directory. You can use any text editor or the echo command. For example, using echo: **echo "Hello, Git!" > hello.txt**

This command creates a file named hello.txt with the content "Hello, Git!".

Step 5: Add the Text File to the Repository

Add the text file to the staging area using the git add command:

git add hello.txt

Step 6: Make the First Commit

Commit the file to the repository with a commit message using the git commit command

git commit -m "Initial commit: Add hello.txt

Step 7: Verify the Commit

You can verify that the commit was successful by using the git log command, which shows the commit history:

git log: You should see the initial commit with the message you provided.

Summary of Commands

cd/path/to/your/directory

git init

echo "Hello, Git!" > hello.txt

git add hello.txt

git commit -m "Initial commit: Add hello.txt"

git log

This sequence will successfully initialize a new Git repository, add a simple text file, and make 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.

Solution:

Here is the step-by-step procedure to create a new branch named feature, switch to it, make changes, and commit them:

Step 1: Ensure You Are in the Repository

Navigate to your Git repository directory if you are not already there:

cd/path/to/your/directory

Step 2: Create a New Branch Named 'feature'

Create a new branch named feature using the git branch command:

git branch feature

Step 3: Switch to the feature Branch

Switch to the newly created branch using the git checkout command:

git checkout feature

Alternatively, you can create and switch to the new branch in a single command using:

git checkout-b feature

Step 4: Make Changes in the 'feature' Branch

Edit the existing file or create a new file. For example, let's append some text to hello.txt

echo "This is a new feature." >> hello.txt

Step 5: Add the Changes to the Staging Area

Add the modified file to the staging area using the git add command: **git add hello.txt**

Step 6: Commit the Changes

Commit the changes with an appropriate commit message using the git commit command

git commit -m "Add feature to hello.txt"

Step 7: Verify the Commit and Branch

Verify that you are on the feature branch and that the commit was successful using the git branch and git log commands:

git branch

git log

You should see the feature branch highlighted and the latest commit message.

Summary of Commands

cd/path/to/your/directory

git branch feature

git checkout feature or git checkout -b feature

echo "This is a new feature" >> hello.txt

git add hello.txt

git commit -m "Add feature to hello.txt"

git branch

git log

This sequence will successfully create a new branch, switch to it, make changes, and commit those changes.

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.

Solution:

Here is a step-by-step procedure to create a hotfix branch, fix an issue, and merge the hotfix branch into main:

Step 1: Ensure You Are in the Repository

Navigate to your Git repository directory if you are not already there:

cd/path/to/your/directory

Step 2: Switch to the Main Branch

Ensure you are on the main branch before creating the hotfix branch: **git checkout main**

Step 3: Create a New Branch Named 'hotfix"

Create a new branch named hotfix using the git branch command: **git branch hotfix**

Step 4: Switch to the 'hotfix' Branch

Switch to the newly created hotfix branch using the git checkout command:

git checkout hotfix

Alternatively, you can create and switch to the new branch in a single command using

git checkout-b hotfix

Step 5: Make Changes in the 'hotfix Branch

Edit the necessary files to fix the issue. For example, let's fix an issue in hello.txt by correcting a typo: **echo "Fixing a critical issue." >> hello.txt**

Step 6: Add the Changes to the Staging Area

Add the modified file to the staging area using the git add command

git add bello.txt

Step 7: Commit the Changes

Commit the changes with an appropriate commit message using the git commit command:

git commit -m "Fix critical issue in hello.txt"

Step 8: Switch Back to the Main Branch

Switch back to the main branch to prepare for the merge:

git checkout main

Step 9: Merge the 'hotfix' Branch into 'main'

Merge the changes from the hotfix branch into the main branch using the git merge command: **git merge hotfix**

Step 10: Verify the Merge

Verify that the merge was successful and that the issue is resolved using the git log and git status commands:

git log

git status

Optional: Delete the 'hotfix' Branch

if the hotfix branch is no longer needed, you can delete it using:

git branch -d hotfix

Summary of Commands

cd/path/to/your/directory

git checkout main

git branch hotfix

git checkout hotfix or git checkout -b hotfix

echo "Fixing a critical issue." >> hello.txt

git add hello.txt

git commit -m "Fix critical issue in hello.txt"

git checkout main