

## **GIT**

### **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.**

git init:

=====

mkdir assignments

cd assignments

git init

vi myfile.txt

ls

git add .

git add myfile.txt

git commit -m "text is added to myfile.txt" =

git remote add origin <https://github.com/Akkidevoju/wiprotraining>

git push origin master

**myfile.txt**

**hi welcome to git**

### **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.

cd path/to/your/directory

git branch feature

git checkout feature

# or you can use git checkout -b feature to combine branch creation and switching

echo "This is an additional line in the feature branch" >> example.txt

git add example.txt

git commit -m "Added a new line in the feature branch"

After running these commands, you will have created and switched to a new branch named 'feature', made changes to a file, and committed those changes.

git commit

### 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.

sh

Copy code

```
cd path/to/your/directory
```

Ensure you are on the 'main' branch before creating the 'hotfix' branch. Switch to 'main' if you are not already on it:

sh

Copy code

```
git checkout main
```

Create a new branch named 'hotfix' and switch to it:

sh

Copy code

```
git checkout -b hotfix
```

Make changes in the 'hotfix' branch to fix the issue. For example, let's assume there's a typo in example.txt that you need to correct. You can use a text editor to fix it or use the command line to make a simple change. Here, we append a line indicating a fix:

sh

Copy code

```
echo "This line corrects a typo in the hotfix branch" >> example.txt
```

Add the changes to the staging area:

sh

Copy code

```
git add example.txt
```

Commit the changes:

sh

Copy code

```
git commit -m "Fixed typo in example.txt"
```

Switch back to the 'main' branch:

```
sh
```

Copy code

```
git checkout main
```

Merge the 'hotfix' branch into the 'main' branch:

```
sh
```

Copy code

```
git merge hotfix
```

(Optional) Delete the 'hotfix' branch after merging to keep your branch list clean:

```
sh
```

Copy code

```
git branch -d hotfix
```

Here is the complete set of commands for reference:

```
sh
```

Copy code

```
cd path/to/your/directory
```

```
git checkout main
```

```
git checkout -b hotfix
```

```
echo "This line corrects a typo in the hotfix branch" >> example.txt
```

```
git add example.txt
```

```
git commit -m "Fixed typo in example.txt"
```

```
git checkout main
```

```
git merge hotfix
```

```
git branch -d hotfix
```

After running these commands, you will have created a 'hotfix' branch, made and committed changes to fix an issue, merged the 'hotfix' branch back into the 'main' branch, and optionally deleted the 'hotfix' branch. This ensures that the issue is resolved in the main codebase.