

GIT ASSIGNMENT

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.

The screenshot shows a GitHub repository page for 'Venkatesh49 / Git_Assignment'. The repository is public and has 1 branch (main) and 0 tags. The commit history shows 4 commits. The most recent commit, 74426f0, was made 1 minute ago and is titled 'File deleted'. The commit history also shows 'First.txt' added 6 minutes ago and 'README.md' added 3 minutes ago. The README file is currently open, showing the title 'Git_Assignment'.

VENKATESH49 / Git_Assignment

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Git_Assignment Public

main 1 Branch 0 Tags

Go to file Add file Code

Venkatesh49 File deleted 74426f0 · 1 minute ago 4 Commits

First.txt	Text file added	6 minutes ago
README.md	first commit	3 minutes ago

README

Git_Assignment

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.

```
venkatesh@venky MINGW64 ~/OneDrive/Desktop/Git/Git Assignment (mai
n)
$ git branch feature

venkatesh@venky MINGW64 ~/OneDrive/Desktop/Git/Git Assignment (mai
n)
$ git switch feature
Switched to branch 'feature'

venkatesh@venky MINGW64 ~/OneDrive/Desktop/Git/Git Assignment (fea
ture)
$ echo " This is a new Feature" > Note.txt

venkatesh@venky MINGW64 ~/OneDrive/Desktop/Git/Git Assignment (fea
ture)
$ git add .
warning: in the working copy of 'Note.txt', LF will be replaced by
CRLF the next time Git touches it

venkatesh@venky MINGW64 ~/OneDrive/Desktop/Git/Git Assignment (fea
ture)
$ git commit -m "Added Note.txt file in Feature Branch"
[feature 3570c64] Added Note.txt file in Feature Branch
1 file changed, 1 insertion(+)
create mode 100644 Note.txt
```

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.

Answer:

Step 1:

Ensure you're on the 'main' branch

git checkout main

Step 2: Pull the latest changes from the remote 'main' branch

git pull origin main

Step 3: Create a new branch called 'hotfix'

```
git branch hotfix
```

Step 4: Change the main branch to hotfix branch

```
git checkout hotfix
```

Step 5: Make the necessary changes to the code to fix the issue

Step 6: Stage the changes

```
git add .
```

Step 7: Commit the changes

```
git commit -m "Fix the issue"
```

Step 8: Push the 'hotfix' branch to the remote repository

```
git push origin hotfix
```

Step 9: Checkout to the 'main' branch

```
git checkout main
```

Step 10: Merge the 'hotfix' branch into 'main'

```
git merge hotfix
```

Step 11: Push the updated 'main' branch to the remote repository

```
git push origin main
```