Day 5 Assignment

Digvijay Thakare

Que 1-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

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements!
https://aka.ms/PSWindows

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git init

Initialized empty Git repository in
C:/Users/Digvija/Desktop/WiproFullstack/Assignments/.git/

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git add .

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: AssignmnetDay5.txt

new file: Day 2Task1.pdf

new file: Day 2Task1.xlsx

new file: Day 4 Assignment Digvijay.docx

new file: Day 4 Assignment Digvijay.pdf

new file: Day2Task2.xlsx

new file: DigvijayThakare Assignment Day2.docx

new file: DigvijayThakare Assignment Day2.pdf

new file: day1 Assignment.docx

new file: day1 Assignment.pdf

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git commit -m "Added assignment file as Assignment day5 "

[master (root-commit) b37b710] Added assignment file as Assignment day5

10 files changed, 1 insertion(+)

create mode 100644 AssignmnetDay5.txt

create mode 100644 Day 2Task1.pdf

create mode 100644 Day 2Task1.xlsx

create mode 100644 Day 4 Assignment Digvijay.docx

create mode 100644 Day 4 Assignment Digvijay.pdf

create mode 100644 Day2Task2.xlsx

create mode 100644 DigvijayThakare Assignment Day2.docx

create mode 100644 DigvijayThakare Assignment Day2.pdf

create mode 100644 day1 Assignment.docx

create mode 100644 day1 Assignment.pdf

1. **Explanation**- **git init**: This command initializes a new Git repository in the current directory. After running this command, Git sets up all the necessary files and directories it needs to begin tracking changes in your project.

2. **git add .**: This command stages all the files in the current directory (and its subdirectories) for the next commit. Staging prepares files to be included in the next commit.
3. **git status**: This command displays the status of the repository, including any changes that have been made and the files that are staged for the next commit. It helps you see what changes you have made since the last commit.
4. **git commit -m "Added assignment file as Assignment day5"**: This command creates a new commit with the changes that are currently staged. The **-m** flag allows you to include a commit message, which describes the changes you've made in the commit. In this case, you've added a message indicating that you've added an assignment file for "Assignment day5".

Que 2 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-

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git branch

* master

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git checkout -b feature

Switched to a new branch 'feature'

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git add .

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git status

On branch feature

nothing to commit, working tree clean

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git branch

```
* feature
  master
PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git add .
PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git status
On branch feature
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Assignment5Que2text file.txt

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git commit
-m "Made changes in feaure brnch"
[feature 45ecc0e] Made changes in feaure brnch
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Assignment5Que2text file.txt
PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments>
```

Explanation-

1. **Checking Current Branch**: First, I checked which branch you were on by running **git branch**. It showed you were on the **master** branch.
2. **Creating and Switching Branch**: Then, I created a new branch called **feature** and switched to it in one go using **git checkout -b feature**. This command creates the new branch and moves you onto it.
3. **Adding Changes**: After switching to the **feature** branch, I made some changes to my files. In this case, I added a new file called **Assignment5Que2text file.txt**.
4. **Checking Status**: I checked the status of my changes using **git status**. Git told me that you had changes ready to be committed in the **feature** branch.

5. **Committing Changes**: I staged the changes with **git add .** and then committed them using **git commit -m "Made changes in feature branch"**. This saved my changes to the **feature** branch with a commit message explaining what I did.

And I created a new branch called **feature**, made changes in it, and committed those changes. Now, our **feature** branch has its own separate history from the **master** branch, and I can continue working on it without affecting the **master** branch.

Assignment 3 Feature Branches and Hotfixes
Create a 'hotfix'branch to fix an issue in the main code .Merge the 'hotfix' branch into main to ensuring the issue is resolved.

Solution-

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git checkout -b hotfix

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git branch

feature

* hotfix

master

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git add .

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git status

On branch hotfix

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: hotfixcode.txt

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git commit -m "Fixed the issue in the main code"

[hotfix 7d5ff20] Fixed the issue in the main code

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 hotfixcode.txt

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git checkout master

Switched to branch 'master'

PS C:\Users\Digvija\Desktop\WiproFullstack\Assignments> git merge hotfix

Updating b37b710..7d5ff20

Fast-forward

Assignment5Que2text file.txt | 0

hotfixcode.txt | 0

2 files changed, 0 insertions(+), 0 deletions(-)

create mode 100644 Assignment5Que2text file.txt

create mode 100644 hotfixcode.txt

Explanation-

1. **Creating Hotfix Branch**: I started by creating a new branch called **hotfix** using **git checkout -b hotfix**. This branch is for fixing an urgent issue in the main code.
2. **Checking Branches**: I checked which branches existed with **git branch**. It showed me that I was on the **hotfix** branch, and there were also **feature** and **master** branches.
3. **Making Changes**: Then, I made the necessary changes to fix the issue in the main code. I added a new file called **hotfixcode.txt** to address the problem.
4. **Checking Status**: I checked the status of my changes with **git status**. Git confirmed that there were changes ready to be committed in the **hotfix** branch.
5. **Committing Changes**: After staging the changes with **git add .**, I committed them using **git commit -m "Fixed the issue in the**

main code". This saved my changes to the **hotfix** branch with a descriptive message.

6. **Switching Back to Master**: Next, I switched back to the **master** branch using **git checkout master**. This is where I wanted to merge the hotfix changes.
7. **Merging Hotfix into Master**: Finally, I merged the changes from the **hotfix** branch into the **master** branch using **git merge hotfix**. This brought the hotfix changes into the **master** branch, ensuring that the issue was resolved in the main code.

So, I effectively managed to fix the urgent issue in the main code by creating a **hotfix** branch, making the necessary changes, and then merging those changes back into the **master** branch.