



Placement Empowerment Program

Cloud Computing and DevOps Centre

Set Up Git Branching

Name: Shalini D

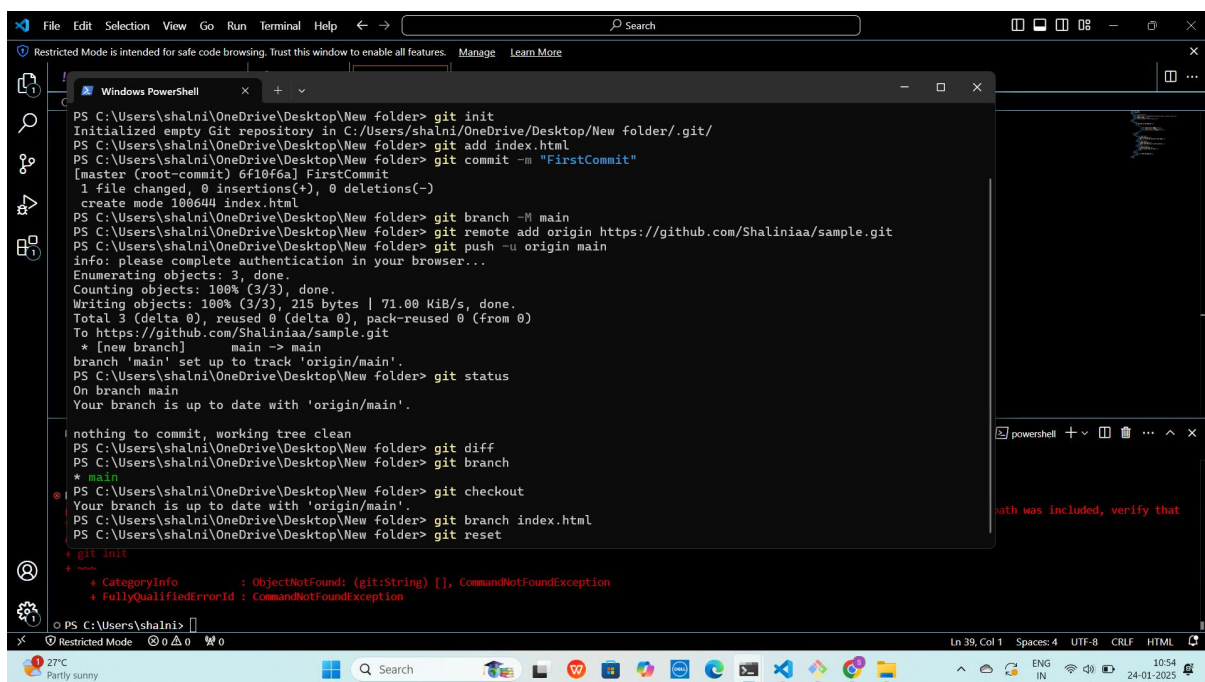
Department: IT



Introduction

Git is a powerful version control system used to track changes in code. It allows multiple developers to work on a project simultaneously without interfering with each other's work. Key concepts include

repositories (which store project files and their history), branches (independent lines of development), commits (snapshots of the repo with unique IDs and messages), merges (integrating changes from one branch to another), pulling (fetching changes from a remote repo), pushing (sending changes to a remote repo), and cloning (copying an existing remote repo to your local machine). Git is essential for collaboration, tracking changes, and managing project versions efficiently.



```
PS C:\Users\shalni\OneDrive\Desktop\New folder> git init
Initialized empty Git repository in C:/Users/shalni/OneDrive/Desktop/New folder/.git/
PS C:\Users\shalni\OneDrive\Desktop\New folder> git add index.html
PS C:\Users\shalni\OneDrive\Desktop\New folder> git commit -m "FirstCommit"
[master (root-commit) 6f10f6a] FirstCommit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
PS C:\Users\shalni\OneDrive\Desktop\New folder> git branch -M main
PS C:\Users\shalni\OneDrive\Desktop\New folder> git remote add origin https://github.com/Shaliniaa/sample.git
PS C:\Users\shalni\OneDrive\Desktop\New folder> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 215 bytes | 71.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Shaliniaa/sample.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\shalni\OneDrive\Desktop\New folder> git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Users\shalni\OneDrive\Desktop\New folder> git diff
PS C:\Users\shalni\OneDrive\Desktop\New folder> git branch
* main
PS C:\Users\shalni\OneDrive\Desktop\New folder> git checkout
Your branch is up to date with 'origin/main'.
PS C:\Users\shalni\OneDrive\Desktop\New folder> git branch index.html
PS C:\Users\shalni\OneDrive\Desktop\New folder> git reset
+ git init
+ CategoryInfo          : ObjectNotFound: (git:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

o PS C:\Users\shalni>
```

Step-by-Step Overview

Step 1: Initialize the repository.

```
PS C:\Users\shalni\OneDrive\Desktop\New folder> git init
Initialized empty Git repository in C:/Users/shalni/OneDrive/Desktop/New folder/.git/
PS C:\Users\shalni\OneDrive\Desktop\New folder> git add index.html
```

Step 2

Add the files to the repository

```
PS C:\Users\shalni\OneDrive\Desktop\New folder> git add index.html
```

Step 3

Commit the changes.

```
PS C:\Users\shalni\OneDrive\Desktop\New folder> git commit -m "FirstCommit"
[master (root-commit) 6f10f6a] FirstCommit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 index.html
```

Step 5

Create the branch

```
PS C:\Users\shalni\OneDrive\Desktop\New folder> git branch -M main
PS C:\Users\shalni\OneDrive\Desktop\New folder> git remote add origin https://github.com/Shaliniaa/sample.git
```

Step 6

Push Your Branch to Remote:



```
PS C:\Users\shalni\OneDrive\Desktop\New folder> git push -u origin main
```

Step 7

Merge the Branch: Switch back to your main branch:



```
PS C:\Users\shalni\OneDrive\Desktop\New folder> git branch  
* main
```

Expected Outcome

To set up Git branching in Windows PowerShell, navigate to your repo with `cd /path/to/your/repo`, create a new branch using `git checkout -b new-feature-branch`, add a new feature by creating or modifying a file, stage your changes with `git add new_feature.py`, commit them using `git commit -m "Add new feature: Print 'Hello, new feature!'"`, push your branch to remote with `git push origin new-feature-branch`, switch back to the main branch using `git checkout main`, merge the feature branch into the main branch with `git merge new-feature-branch`, and finally push the merged changes to remote using `git push origin main`. This will create a new branch, add a feature, merge it into the main branch, and push the changes to the remote repository.