Placement Empowerment Program

Cloud Computing and DevOps Centre

Set Up Git Branching

Create a new branch in your Git repository for testing. Add a new feature and merge it.

Name: Bhavyaa.V Department: IT

Introduction and Overview

This PowerShell session demonstrates a basic Git workflow, including cloning a repository, creating and switching branches, adding a new feature, and merging changes. The process begins with cloning an empty repository from GitHub, followed by the creation of main and testing branches. A new file, feature.txt, is added and committed in the testing branch before merging it into main. The final step verifies the successful integration of changes. This workflow is essential for structured development, version control, and efficient collaboration in software projects.

Objective

The goal of this project is to:

- **Implement a Git Workflow** – Demonstrate creating, switching, and merging branches to manage code efficiently.

- **Ensure Smooth Feature Integration** – Add a new feature (`feature.txt`) in a separate branch and merge it into `main` without conflicts.

Importance

- **Branching for Organized Development** – Creating separate branches (`main` and `testing`) allows for structured development and testing without affecting the main codebase.

- **Version Control & Change Tracking** – Staging and committing changes ensure that every modification is tracked, making it easier to review and revert if necessary.

- **Safe Integration with Merging** – Merging the `testing` branch into `main` integrates new features safely while maintaining code stability.

Step-by-Step Overview

Step1: Clone the Repository

```
git clone "https://github.com/Bhavyaa-cyber/po"
```

- This command clones the repository from the given GitHub URL into the local machine.
- The repository is empty, as indicated by the warning message.

Step 2: **Navigate into the Cloned Repository**

```
cd po
```

- Changes the current directory to the newly cloned repository `po`.

Step 3: Check Existing Branches

```
it branch
```

- Lists all available branches in the repository.
- Since this is a new repository, there might be no existing branches.

Step 4: Create and Switch to a New Branch Named `main`

```
CopyEdit
git checkout -b main
```

- Creates a new branch named `main` and switches to it.

Step 5: Create and Switch to a New Branch Named `testing`

```
git checkout -b testing
```

- Creates a new branch named `testing` and switches to it.

Step 6: Create a New File and Add Content

```
echo "feature" > feature.txt
```

- Creates a new file named `feature.txt` and writes the word "feature" into it.

Step 7: Add the File to the Staging Area

```
git add feature.txt
```

- Adds the newly created file `feature.txt` to the staging area, preparing it for commit.

Step 8: Commit the Changes

```
git commit -m "add new feature:feature.txt"
```

- Commits the staged file with the message "`add new feature:feature.txt`".

Step 9: Switch to the `main` Branch

```
git checkout -b main
```

- Switches back to the `main` branch (it appears the `main` branch was created again, which may be redundant).

Step 10: Merge `testing` Branch into `main`

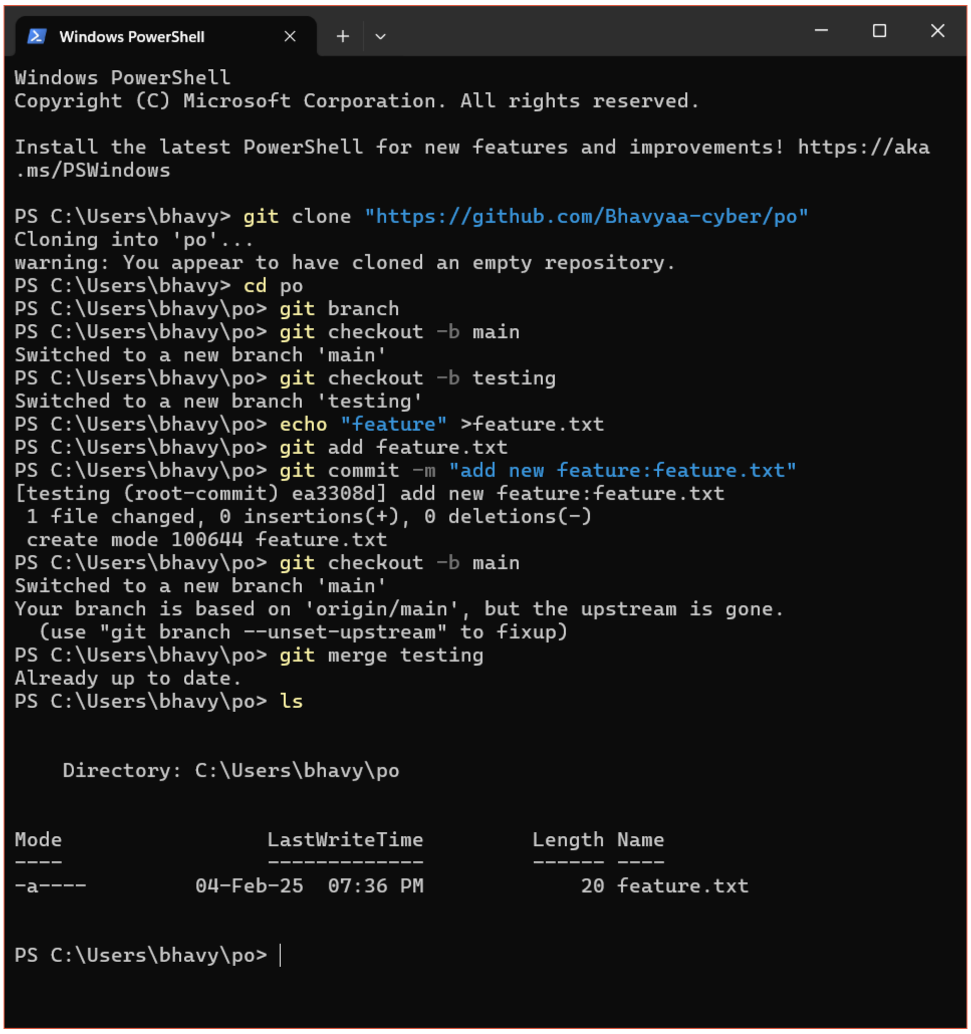
```
git merge testing
```

- Merges the changes from the `testing` branch into the `main` branch.

Step 11: Verify the Merged File

```
ls
```

- Lists the files in the current directory.
- The file `feature.txt` is present, confirming the merge was successful.



```
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\bhavy> git clone "https://github.com/Bhavyaa-cyber/po"
Cloning into 'po'...
warning: You appear to have cloned an empty repository.
PS C:\Users\bhavy> cd po
PS C:\Users\bhavy\po> git branch
PS C:\Users\bhavy\po> git checkout -b main
Switched to a new branch 'main'
PS C:\Users\bhavy\po> git checkout -b testing
Switched to a new branch 'testing'
PS C:\Users\bhavy\po> echo "feature" >feature.txt
PS C:\Users\bhavy\po> git add feature.txt
PS C:\Users\bhavy\po> git commit -m "add new feature:feature.txt"
[testing (root-commit) ea3308d] add new feature:feature.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 feature.txt
PS C:\Users\bhavy\po> git checkout -b main
Switched to a new branch 'main'
Your branch is based on 'origin/main', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)
PS C:\Users\bhavy\po> git merge testing
Already up to date.
PS C:\Users\bhavy\po> ls


    Directory: C:\Users\bhavy\po


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         04-Feb-25   07:36 PM             20 feature.txt


PS C:\Users\bhavy\po> |
```

Expected Outcome

By completing this POC, you will:

• **Successful Branch Creation & Switching** – The repository will have two branches: `main` and `testing`, with smooth transitions between them.
• **Feature Addition & Merge** – The file `feature.txt` will be created in the `testing` branch, committed, and successfully merged into `main`.

• **File Confirmation** – After merging, `feature.txt` will be visible in the `main` branch when listing directory contents.