PURBANCHAL UNIVERSITY

KHWOPA ENGINEERING COLLEGE

LIBALI-08, BHAKTAPUR

LAB REPORT ON .NET

LAB NO. 01

SUBMITTED BY:

Name: Jenisha Shrestha

Roll No. : 770315

Group: A

SUBMITTED TO:

Department of Computer Engineering

Submission: 2081/12/09

Theory:

1. **Git:**
   Git is a distributed version control system used for tracking the changes in the source code during software development. It allows multiple developers to collaborate efficiently by managing different version of project. Git enables branching, merging and reverting changes, making code management easier. It is widely used open-source and commercial projects. Popular platform like GitHub, GitLab, and Bitbucket provide remote repositories for Git-based collaboration.

2. **GitHub**
   GitHub is a web-based platform for version control and collaboration using Git. It allows developers to store, manage, and share code repositories efficiently. GitHub supports features like branching, pull requests, issue tracking, and CI/CD integration. It is widely used for open-source and private projects, enabling seamless teamwork. GitHub also provides cloud-based hosting, making it accessible from anywhere.

General Git and GitHub Commands:

Git Configuration

git config --global user.name "Your Name"
This command sets the global username for the Git commits.
git config --global user.email "your_email@example.com"
This command sets the global email associated with Git commits.

Initializing

git init
initializes a new Git repository in the current directory.

Staging and Commits

git add .
It stages all changes and new files for commit.
git commit -m "Your commit message"
Saves the staged changes with a descriptive message.

Branching and Merging

git branch
Lists all the branches in the repository.
git branch <branch_name>
Creates a new branch for separate development.
git checkout <branch_name> / Git switch <branch_name>
Switches to the specified branch

git merge <branch_name>
Merges changes from the specified branch into the current branch.

Pushing and Pulling

git push -u origin <branch_name>
Uploads the local changes to the remote repository.
git pull origin <branch_name>
Fetches and merge the latest changes from the remote repository.

Status and Logs

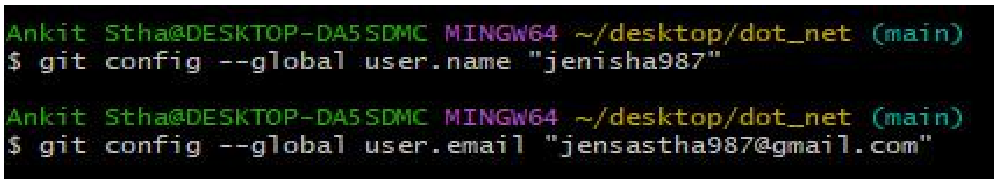
git status
Show the current state of the files in the working directory (modified, staged or untracked).
git log
Displays the commit history of the repository.

GitHub Specific

git remote add origin <repo_url>
Links the local repository to a remote repository on GitHub.

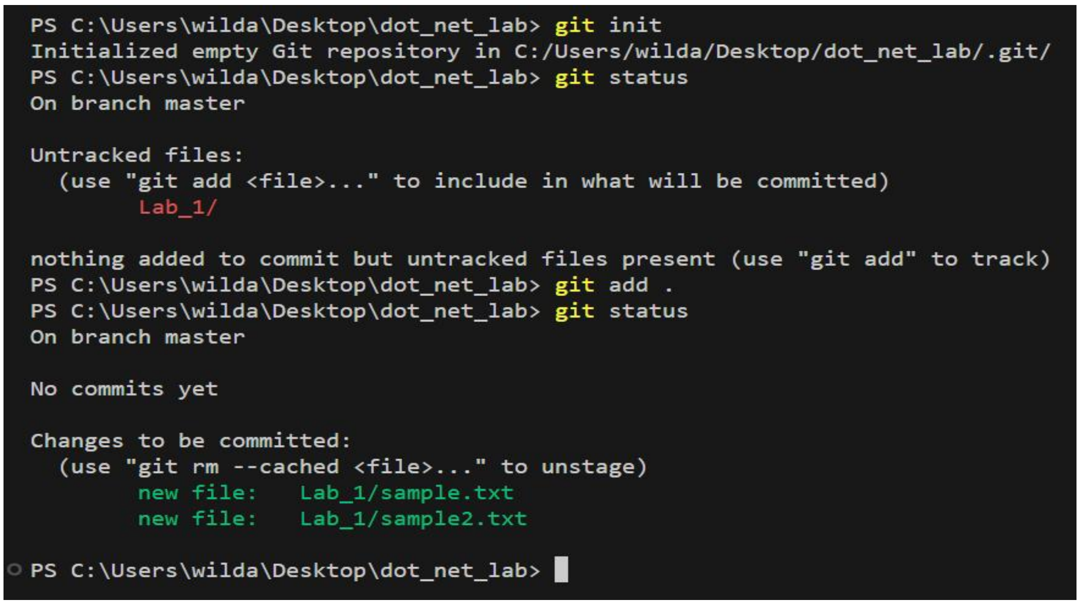
Lab Works

First set the global username and email of the GitHub.



```
Ankit Stha@DESKTOP-DA5SDMC MINGW64 ~/desktop/dot_net (main)
$ git config --global user.name "jenisha987"

Ankit Stha@DESKTOP-DA5SDMC MINGW64 ~/desktop/dot_net (main)
$ git config --global user.email "jensastha987@gmail.com"
```

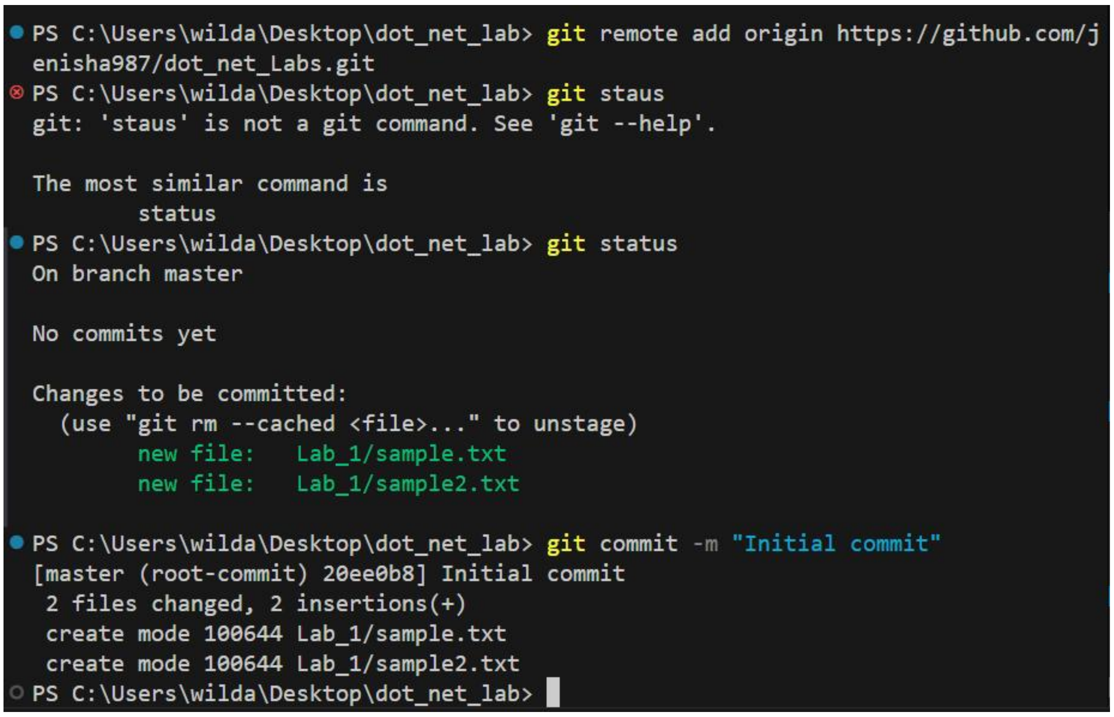
Create a folder and inside it files as per the user desire so that we can identify the changes inside the file using the version control (Git).



```
PS C:\Users\wilda\Desktop\dot_net_lab> git init
Initialized empty Git repository in C:/Users/wilda/Desktop/dot_net_lab/.git/
PS C:\Users\wilda\Desktop\dot_net_lab> git status
On branch master

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Lab_1/

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\wilda\Desktop\dot_net_lab> git add .
PS C:\Users\wilda\Desktop\dot_net_lab> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Lab_1/sample.txt
        new file:   Lab_1/sample2.txt

PS C:\Users\wilda\Desktop\dot_net_lab> 
```

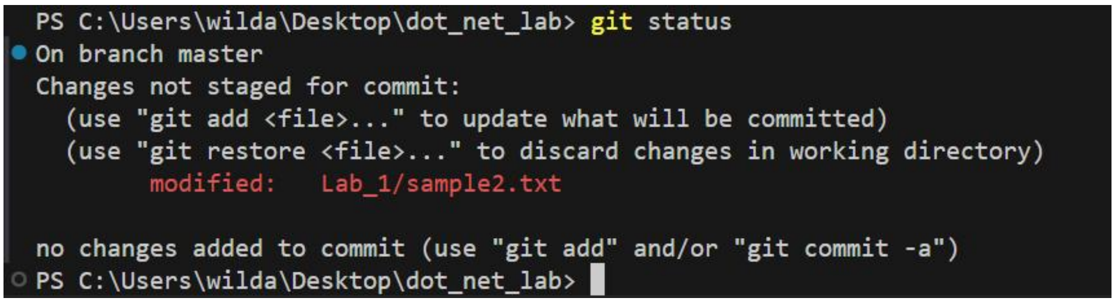
On creating the new files, initially the files are in the untracked stage so sent the untracked files to the staging stage. To do so first initialize the directory and staged the files.

Now commit the files such that the files are stored in the local repository.



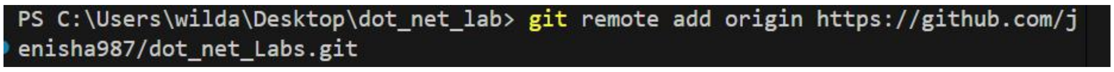
```
PS C:\Users\wilda\Desktop\dot_net_lab> git remote add origin https://github.com/j
enisha987/dot_net_Labs.git
PS C:\Users\wilda\Desktop\dot_net_lab> git staus
git: 'staus' is not a git command. See 'git --help'.

The most similar command is
        status
PS C:\Users\wilda\Desktop\dot_net_lab> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Lab_1/sample.txt
        new file:   Lab_1/sample2.txt

PS C:\Users\wilda\Desktop\dot_net_lab> git commit -m "Initial commit"
[master (root-commit) 20ee0b8] Initial commit
 2 files changed, 2 insertions(+)
 create mode 100644 Lab_1/sample.txt
 create mode 100644 Lab_1/sample2.txt
PS C:\Users\wilda\Desktop\dot_net_lab>
```

Make certain changes inside the file to see the changes in the file status.



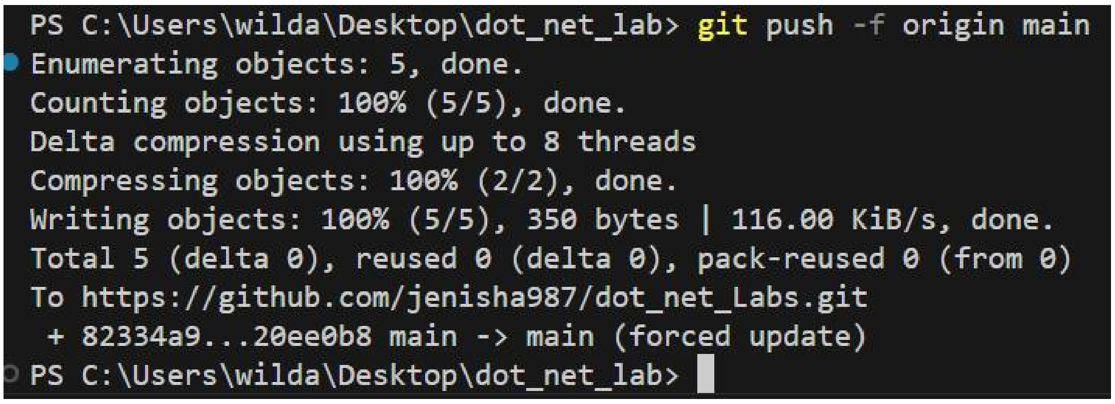
```
PS C:\Users\wilda\Desktop\dot_net_lab> git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Lab_1/sample2.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\wilda\Desktop\dot_net_lab>
```

After changing the contents in the file **"sample2.txt"** add the file and commit it.
All of these files are saved in the local repository. Now to add these files in the remote repository create the repository in the GitHub and copy the url of the repo and use the following code.

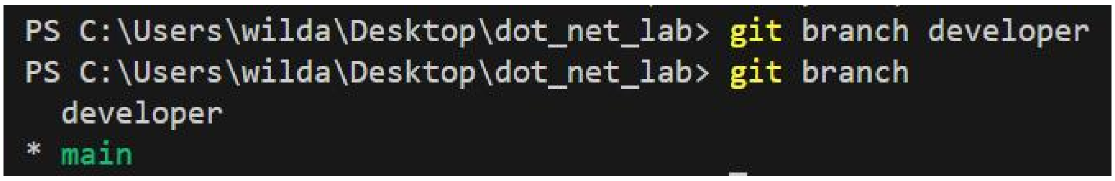


```
PS C:\Users\wilda\Desktop\dot_net_lab> git remote add origin https://github.com/j
enisha987/dot_net_Labs.git
```

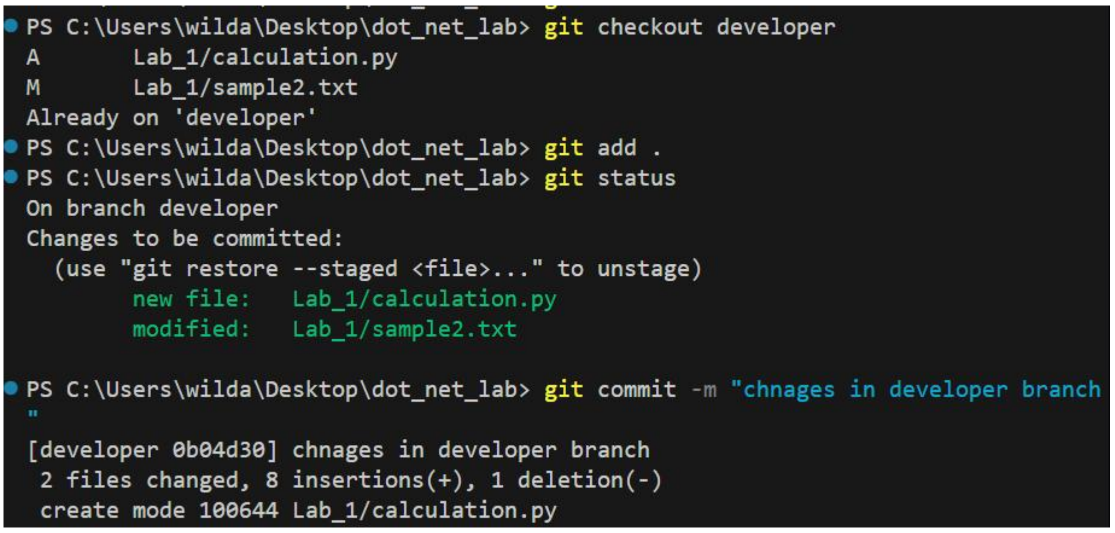
Now push the files in the repository created.


```
PS C:\Users\wilda\Desktop\dot_net_lab> git push -f origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (5/5), 350 bytes | 116.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/jenisha987/dot_net_Labs.git
 + 82334a9...20ee0b8 main -> main (forced update)
PS C:\Users\wilda\Desktop\dot_net_lab>
```

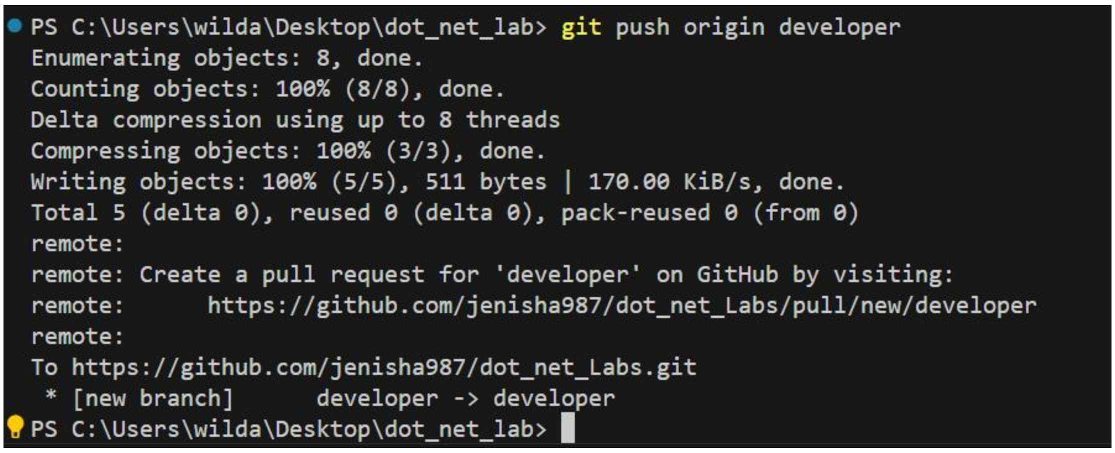
Now creating branches, allowing the work on different version of a project without affecting the main codebase.


```
PS C:\Users\wilda\Desktop\dot_net_lab> git branch developer
PS C:\Users\wilda\Desktop\dot_net_lab> git branch
  developer
* main
```

Moving on to the recently created branch to modify the contents in the file without affecting the main codebase.

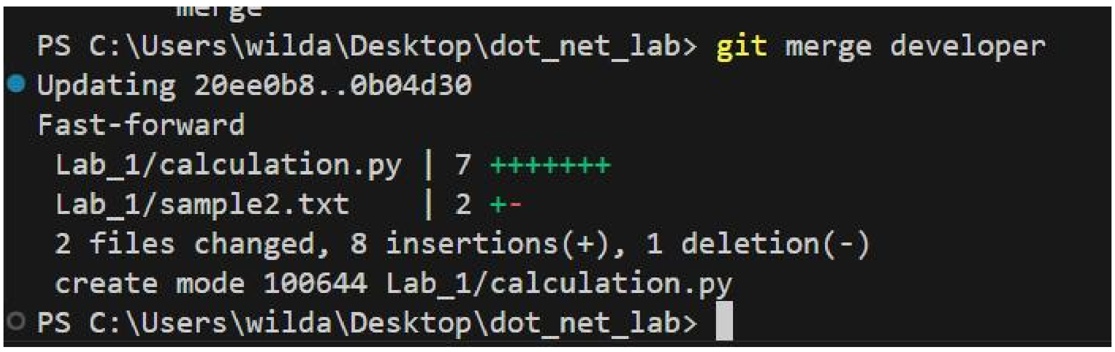

```
PS C:\Users\wilda\Desktop\dot_net_lab> git checkout developer
A       Lab_1/calculation.py
M       Lab_1/sample2.txt
Already on 'developer'
PS C:\Users\wilda\Desktop\dot_net_lab> git add .
PS C:\Users\wilda\Desktop\dot_net_lab> git status
On branch developer
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Lab_1/calculation.py
        modified:   Lab_1/sample2.txt

PS C:\Users\wilda\Desktop\dot_net_lab> git commit -m "chnages in developer branch
"
[developer 0b04d30] chnages in developer branch
 2 files changed, 8 insertions(+), 1 deletion(-)
 create mode 100644 Lab_1/calculation.py
```

To change the branch, we can use the command *"git switch main"*. To make sure the branch is visible to other users of the repository push the branch in the GitHub.



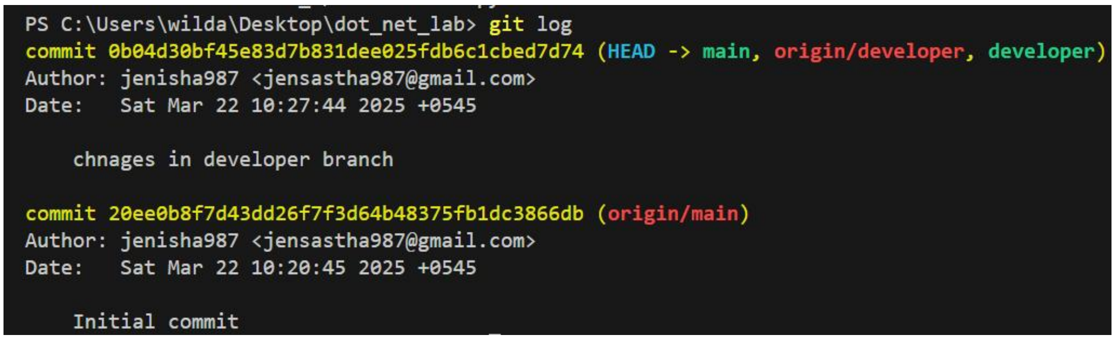
```
PS C:\Users\wilda\Desktop\dot_net_lab> git push origin developer
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 511 bytes | 170.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'developer' on GitHub by visiting:
remote:      https://github.com/jenisha987/dot_net_Labs/pull/new/developer
remote:
To https://github.com/jenisha987/dot_net_Labs.git
 * [new branch]      developer -> developer
PS C:\Users\wilda\Desktop\dot_net_lab>
```

Merging the branches such that the changes in the new branch or new features added in the new branch is added to the main code base.



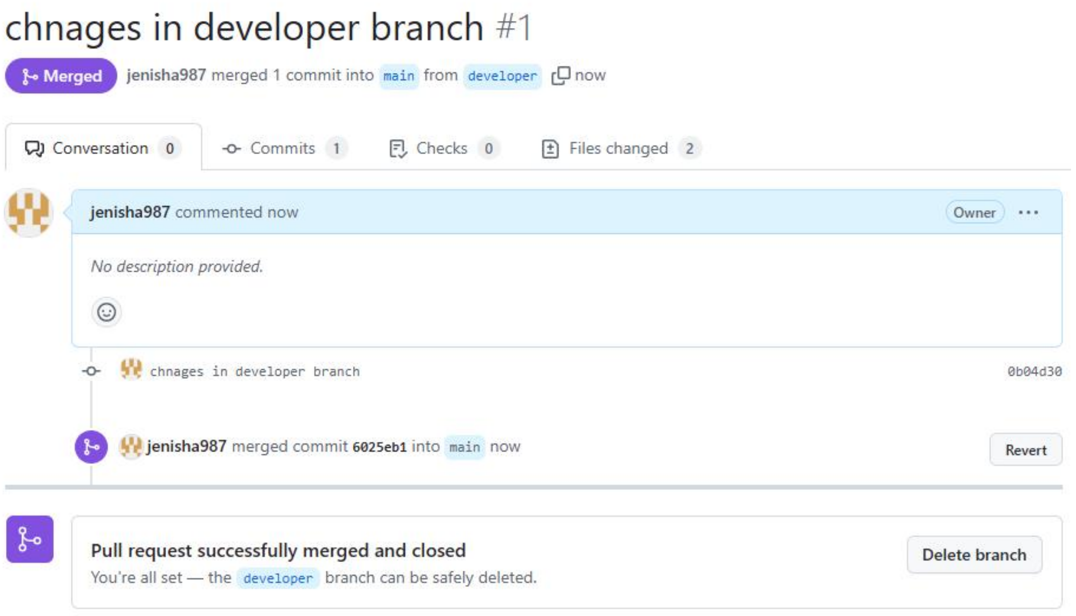
```
PS C:\Users\wilda\Desktop\dot_net_lab> git merge developer
Updating 20ee0b8..0b04d30
Fast-forward
 Lab_1/calculation.py | 7 +++++++
 Lab_1/sample2.txt    | 2 +-
 2 files changed, 8 insertions(+), 1 deletion(-)
 create mode 100644 Lab_1/calculation.py
PS C:\Users\wilda\Desktop\dot_net_lab>
```

To check the commits performed in the past



```
PS C:\Users\wilda\Desktop\dot_net_lab> git log
commit 0b04d30bf45e83d7b831dee025fdb6c1cbed7d74 (HEAD -> main, origin/developer, developer)
Author: jenisha987 <jensastha987@gmail.com>
Date:   Sat Mar 22 10:27:44 2025 +0545

    chnages in developer branch

commit 20ee0b8f7d43dd26f7f3d64b48375fb1dc3866db (origin/main)
Author: jenisha987 <jensastha987@gmail.com>
Date:   Sat Mar 22 10:20:45 2025 +0545

    Initial commit
```

Merging the branch in the GUI GitHub (Web)


chnages in developer branch #1

Merged jenisha987 merged 1 commit into main from developer now

Conversation 0 Commits 1 Checks 0 Files changed 2

jenisha987 commented now Owner

No description provided.

chnages in developer branch 0b04d30

jenisha987 merged commit 6025eb1 into main now Revert

Pull request successfully merged and closed

You're all set — the developer branch can be safely deleted. Delete branch

Conclusion:

In this lab, we learn about the basics of the Git and GitHub. We perform initialization, branching, merging, pushing and commit.