**PURBANCHAL UNIVERSITY**

****

**KHWOPA ENGINEERING COLLEGE**

**LIBALI-08, BHAKTAPUR**

LAB REPORT ON .NET

LAB NO. 01

**SUBMITTED BY: SUBMITTED TO:**

Name: Kritima Shrestha Department of

Roll No. : 770318 Computer Engineering

Submission: 2081/12/09

**LAB-1**

**Git and Github**

**Theory**

**Git**

Git is a **distributed version control system (DVCS)** that helps developers track changes in their code, collaborate with others, and manage different versions of a project efficiently. Git enables branching, merging and reverting changes, making code management easier. It helps multiple developers to work on the same project simultaneously. It is widely used open-source and commercial projects. It was created by **Linus Torvalds** in 2005 for Linux kernel development.

#### ****Git Workflow****

1. **Working Directory** – The files you are currently working on.
2. **Staging Area (Index)** – Files that are marked to be committed.
3. **Repository (Local Repo)** – The committed files stored locally.
4. **Remote Repository** – A shared repository (e.g., GitHub, GitLab, Bitbucket).

**GitHub**

GitHub is a cloud-based platform for version control and collaboration, primarily used for managing software development projects. It is built around Git, a distributed version control system created by Linus Torvalds.

**Forking & Cloning**

* **Forking** creates a personal copy of another user’s repository.
* **Cloning** downloads a repository to a local computer for offline development.

**Common 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.

**Git Initializing**

*git init*

This command initialize a new git repository in the current directory.

**Git Linking**

*git remote add origin* <repo>

This command links new repository of github with the local codes.

**Git Staging and Commits**

*git add .*

This command add files to the staging area.

*git commit -m “message”*

This command commit and save changes of stage area with a message.

**Git Status and Log**

*git status*

This command check the status of the working directory.

*git log*

This command view commit history.

**Git Branching and Merging**

*git branch*

This command list all the branches exist in the repository.

*git branch <branch\_name>*

This command creates new branch for separate development.

*git checkout <branch>*

This command switches to another branch.

*git switch <branch\_name>*

This command switches to another branch.

*git merge <branch\_name>*

This command merges a specified branch into the current branch.

**Git Push and Pull**

*git push -u origin <branch\_name>*

This command uploads commits to a remote repository

*git pull origin*

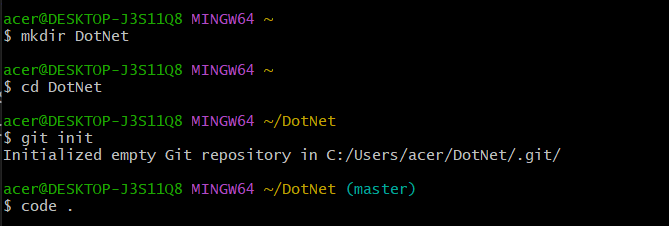
This command fetch and merge changes from a remote repository

**Git Clone**

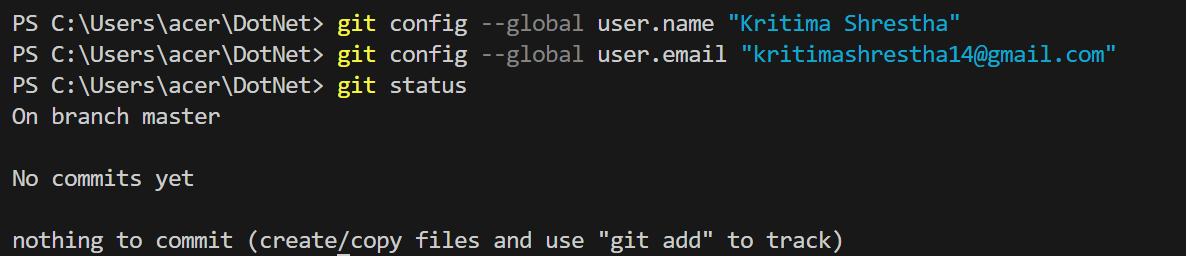
*git clone <repo\_url>*

This command copy(clone) an existing repository.

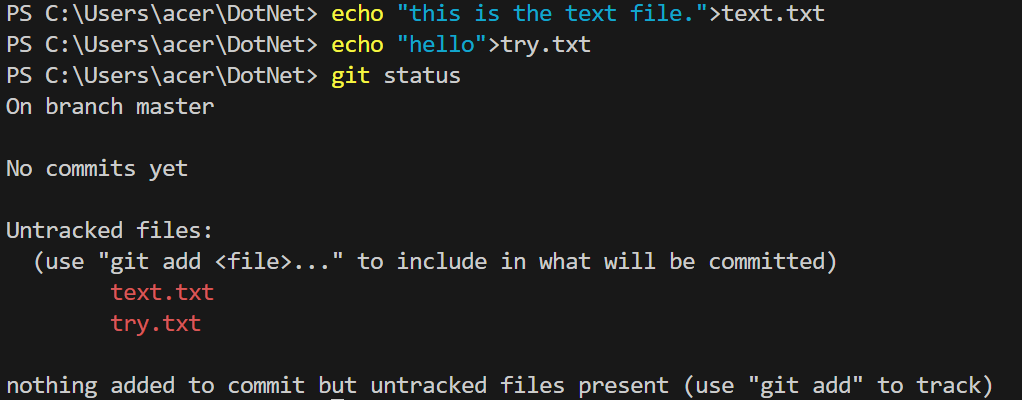
**Lab Work**



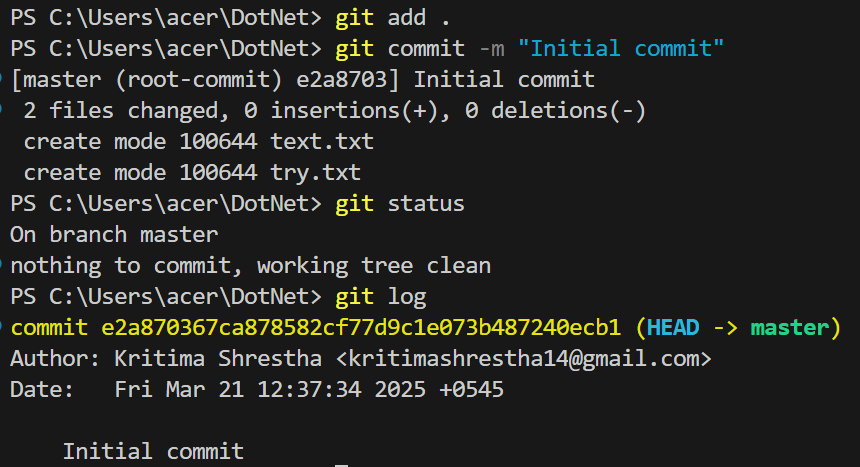
Initially, we create folder and initialize the git and as per user desire we create, change the files using the version control git with different commands.



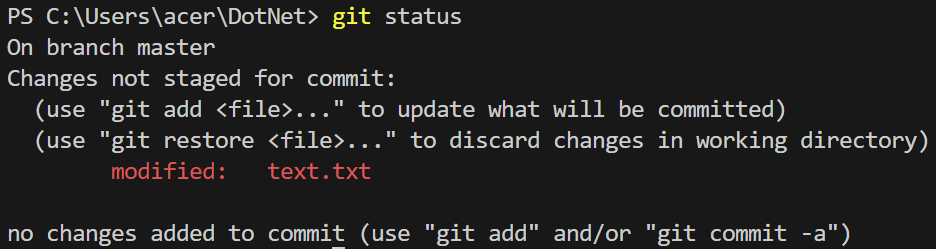
We set the global username and email of the GitHub.



We created the empty files text.txt and try.txt and insert values with echo and checked the status ,initially the files are in untracked stage and we sent the files to the staging stage.



The files are then added for staging and commit the files with the message such that the files are stored in the local repository.

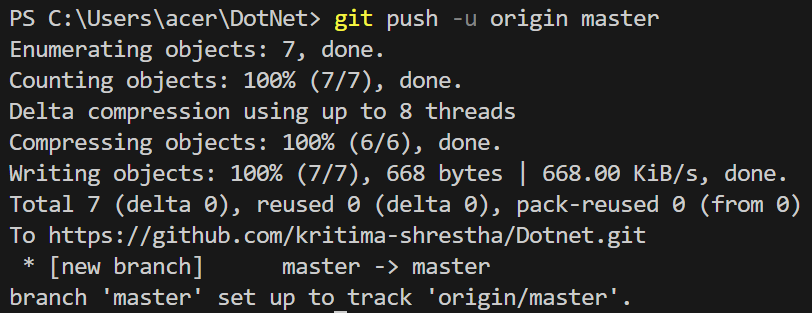


We again made certain changes in file text.txt to see certain changes in the file status.

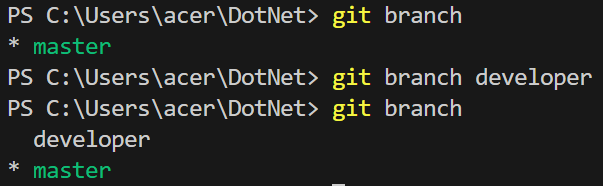
And again commit it so that all of the files are saved in the local repository.



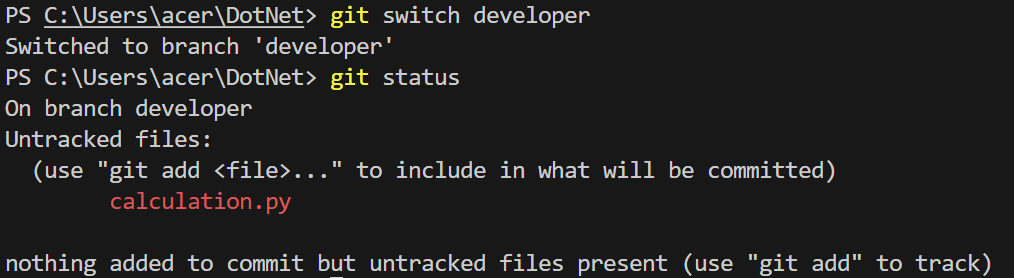
We then add the files in the remote repository by creating the repository in the GitHub and copying the url of the repo and using the above code.



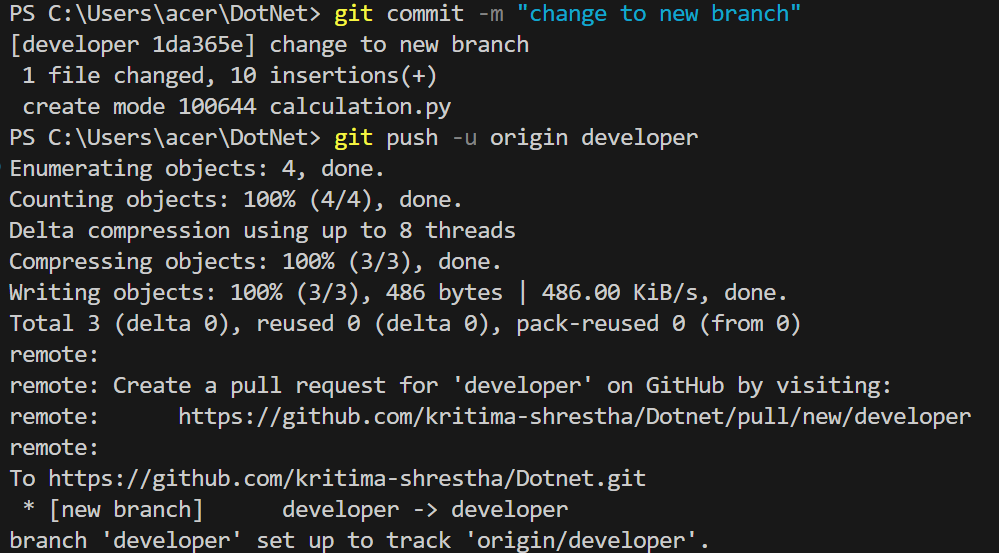
After that we push the files to the created repository.



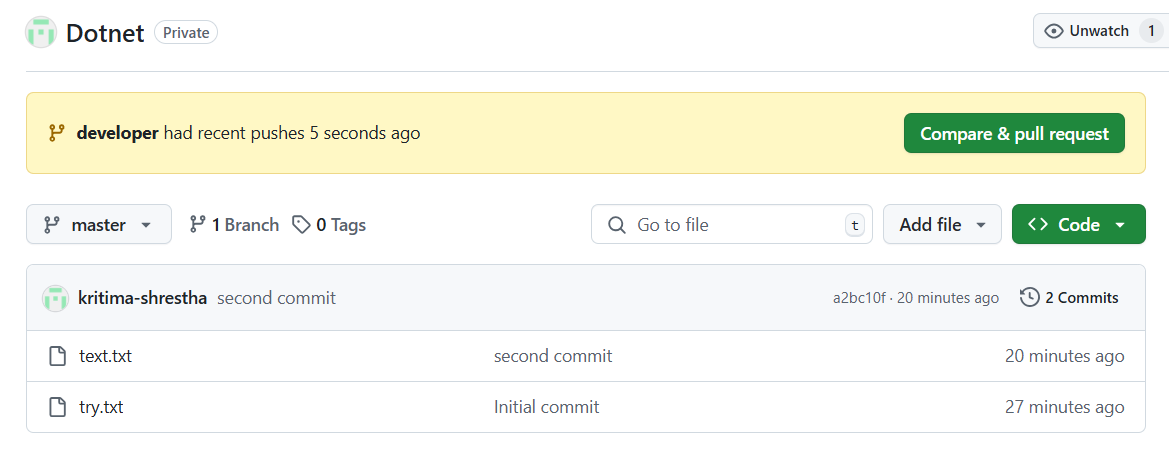
We checked the existing branch in our local repository. Then we create branches for working different version of programs without affecting the main code.



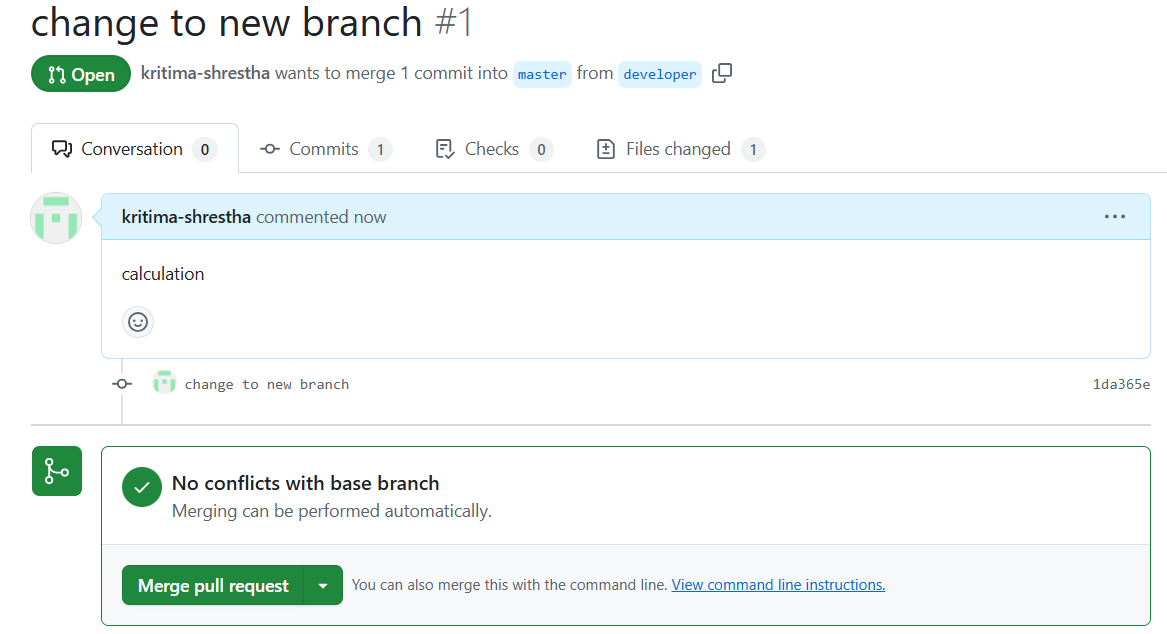
We switched to new branch “developer” branch where we modify and add different files without affecting the main code. We here added new files “calculation.py”. Initially it is in untracked stage.



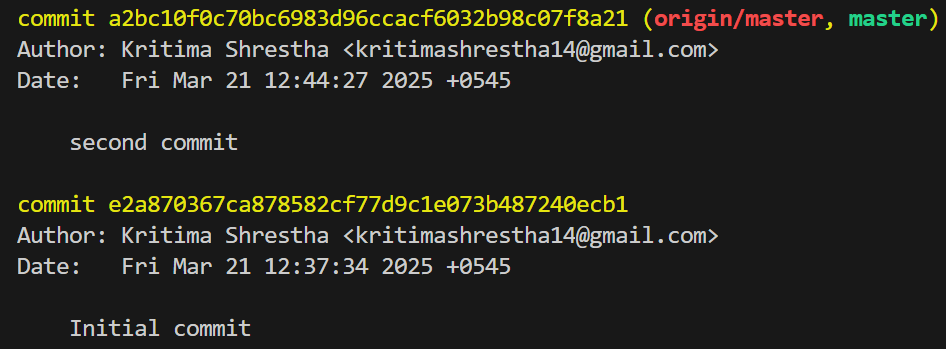
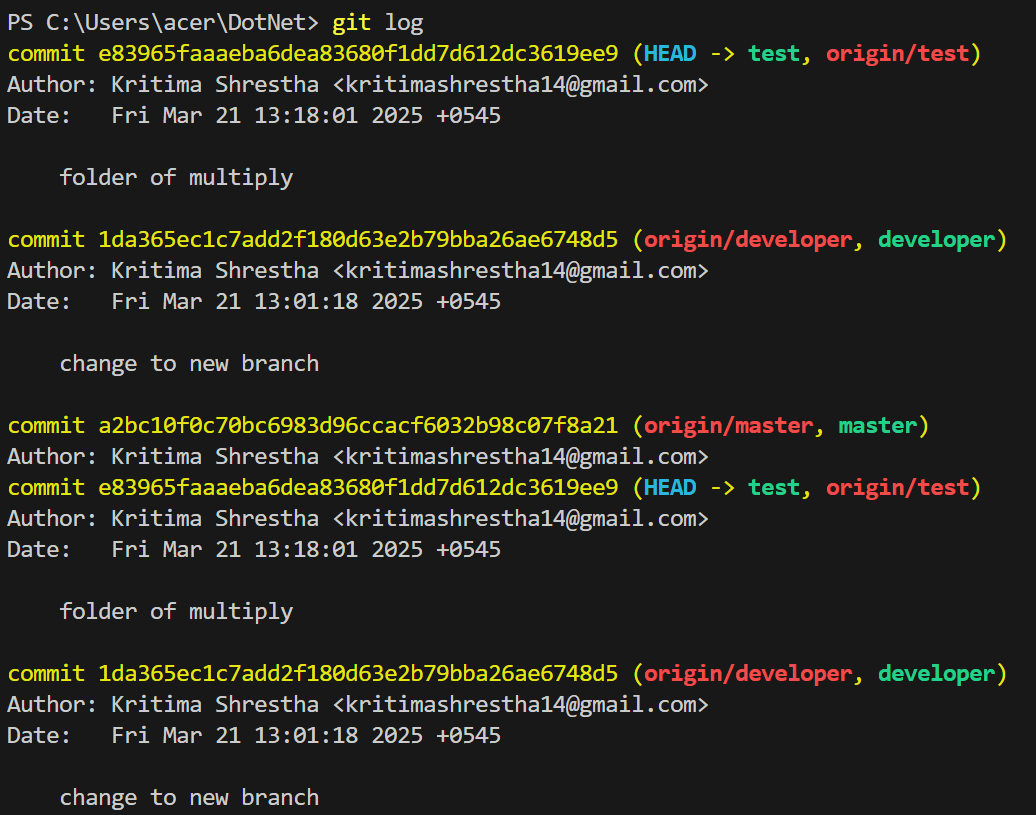
Then we commit the changes and push the branch in the GitHub to make sure the branch is visible to other users of the repository.



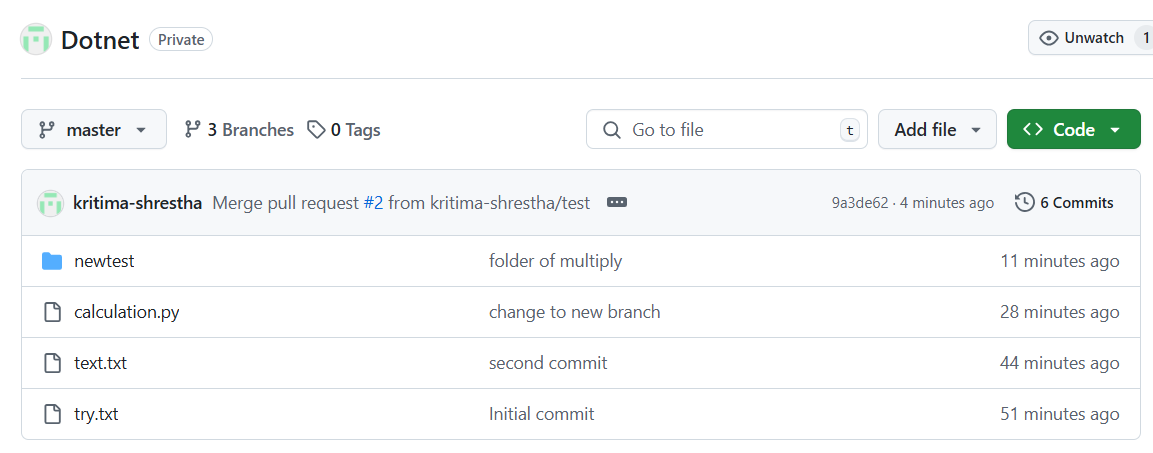
Then we create pull request for merging the latest branch pushed.



We then merge the “developer” branch with the “master” branch in the GitHub .



We use “git log” to see all the commits we performed as the history.



After completion of merge we can see all the files in master branch.

**Conclusion**

In this lab, we learned about basic commands of Git and Github. We performed initializing, linking, commiting, branching and merging commands.