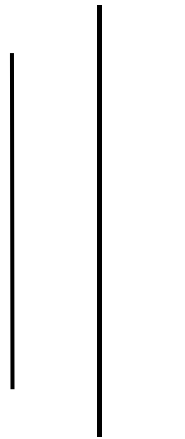


PURBANCHAL UNIVERSITY



**KHWOPA ENGINEERING COLLEGE
LIBALI-08, BHAKTAPUR**



LAB REPORT ON .NET
LAB NO. 01

SUBMITTED BY:

Name: Kritima Shrestha
Roll No. : 770318

SUBMITTED TO:

Department of
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

```
acer@DESKTOP-J3S11Q8 MINGW64 ~
$ mkdir DotNet

acer@DESKTOP-J3S11Q8 MINGW64 ~
$ cd DotNet

acer@DESKTOP-J3S11Q8 MINGW64 ~/DotNet
$ git init
Initialized empty Git repository in C:/Users/acer/DotNet/.git/

acer@DESKTOP-J3S11Q8 MINGW64 ~/DotNet (master)
$ code .
```

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.

```

PS C:\Users\acer\DotNet> git config --global user.name "Kritima Shrestha"
PS C:\Users\acer\DotNet> git config --global user.email "kritimashrestha14@gmail.com"
PS C:\Users\acer\DotNet> git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

```

We set the global username and email of the GitHub.

```

PS C:\Users\acer\DotNet> echo "this is the text file.">text.txt
PS C:\Users\acer\DotNet> echo "hello">try.txt
PS C:\Users\acer\DotNet> git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)

```

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.

```

PS C:\Users\acer\DotNet> git add .
PS C:\Users\acer\DotNet> git commit -m "Initial commit"
[master (root-commit) e2a8703] Initial commit
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 text.txt
 create mode 100644 try.txt
PS C:\Users\acer\DotNet> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\acer\DotNet> git log
commit e2a870367ca878582cf77d9c1e073b487240ecb1 (HEAD -> master)
Author: Kritima Shrestha <kritimashrestha14@gmail.com>
Date:   Fri Mar 21 12:37:34 2025 +0545

Initial commit

```

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

```
PS C:\Users\acer\DotNet> 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:   text.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

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.

```
PS C:\Users\acer\DotNet> git remote add origin https://github.com/kritima-shrestha/Dotnet.git
```

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.

```
PS C:\Users\acer\DotNet> git push -u origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 668 bytes | 668.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/kritima-shrestha/Dotnet.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

After that we push the files to the created repository.

```
PS C:\Users\acer\DotNet> git branch
* master
PS C:\Users\acer\DotNet> git branch developer
PS C:\Users\acer\DotNet> git branch
    developer
* master
```

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

```
PS C:\Users\acer\DotNet> git switch developer
Switched to branch 'developer'
PS C:\Users\acer\DotNet> git status
On branch developer
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        calculation.py

nothing added to commit but untracked files present (use "git add" to track)
```


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.

```
PS C:\Users\acer\DotNet> git commit -m "change to new branch"
[developer 1da365e] change to new branch
1 file changed, 10 insertions(+)
create mode 100644 calculation.py
PS C:\Users\acer\DotNet> git push -u origin developer
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 486 bytes | 486.00 KiB/s, done.
Total 3 (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/kritima-shrestha/Dotnet/pull/new/developer
remote:
To https://github.com/kritima-shrestha/Dotnet.git
 * [new branch]      developer -> developer
branch 'developer' set up to track 'origin/developer'.
```



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.

The screenshot shows the GitHub interface for a repository named 'Dotnet' (Private). At the top, there's a yellow banner indicating that the 'developer' branch had recent pushes 5 seconds ago, with a 'Compare & pull request' button. Below this, the repository navigation bar shows 'master' as the selected branch, '1 Branch', and '0 Tags'. There are search and file management options. The commit history table shows two commits by 'kritima-shrestha':


File	Commit Message	Time
text.txt	second commit	20 minutes ago
try.txt	Initial commit	27 minutes ago

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


change to new branch #1


 kritima-shrestha wants to merge 1 commit into `master` from `developer` 


Conversation 0 Commits 1 Checks 0 Files changed 1


 kritima-shrestha commented now


calculation



 change to new branch 1da365e



 **No conflicts with base branch**
Merging can be performed automatically.

Merge pull request  You can also merge this with the command line. [View command line instructions.](#)

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

```
PS C:\Users\acer\DotNet> git log
commit e83965faaaeba6dea83680f1dd7d612dc3619ee9 (HEAD -> test, origin/test)
Author: Kritima Shrestha <kritimashrestha14@gmail.com>
Date: Fri Mar 21 13:18:01 2025 +0545

    folder of multiply

commit 1da365ec1c7add2f180d63e2b79bba26ae6748d5 (origin/developer, developer)
Author: Kritima Shrestha <kritimashrestha14@gmail.com>
Date: Fri Mar 21 13:01:18 2025 +0545

    change to new branch

commit a2bc10f0c70bc6983d96ccacf6032b98c07f8a21 (origin/master, master)
Author: Kritima Shrestha <kritimashrestha14@gmail.com>
commit e83965faaaeba6dea83680f1dd7d612dc3619ee9 (HEAD -> test, origin/test)
Author: Kritima Shrestha <kritimashrestha14@gmail.com>
Date: Fri Mar 21 13:18:01 2025 +0545

    folder of multiply

commit 1da365ec1c7add2f180d63e2b79bba26ae6748d5 (origin/developer, developer)
Author: Kritima Shrestha <kritimashrestha14@gmail.com>
Date: Fri Mar 21 13:01:18 2025 +0545

    change to new branch


commit a2bc10f0c70bc6983d96ccacf6032b98c07f8a21 (origin/master, master)
Author: Kritima Shrestha <kritimashrestha14@gmail.com>
Date: Fri Mar 21 12:44:27 2025 +0545

    second commit

commit e2a870367ca878582cf77d9c1e073b487240ecb1
Author: Kritima Shrestha <kritimashrestha14@gmail.com>
Date: Fri Mar 21 12:37:34 2025 +0545


    Initial commit
```





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

 Dotnet Private

Unwatch 1

master 3 Branches 0 Tags Add file Code

 kritima-shrestha Merge pull request #2 from kritima-shrestha/test 9a3de62 · 4 minutes ago 6 Commits

 newtest	folder of multiply	11 minutes ago
 calculation.py	change to new branch	28 minutes ago
 text.txt	second commit	44 minutes ago
 try.txt	Initial commit	51 minutes ago

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, committing, branching and merging commands.