PURBANCHAL UNIVERSITY

KHWOPA ENGINEERING COLLEGE

LIBALI-08, BHAKTAPUR

LAB REPORT ON .NET

LAB NO. 01

SUBMITTED BY:	**SUBMITTED TO:**
Name: Asmita Shrestha	Department of Computer Engineering
Roll No. : 770307	
Group: A	

Submission: 2081/12/09

Theory:

1. **Git:**
   Git is a distributed version control system used to track changes in source code during software development. It allows multiple developers to collaborate on a project by enabling them to work on different parts of the code simultaneously. Git tracks the history of changes, making it easy to revert to previous versions of the code, merge updates from different contributors, and manage branches for different features or bug fixes. Popular platforms like GitHub and GitLab are built on Git, allowing developers to host and share their repositories online.

2. **GitHub**
   GitHub is a web-based platform used for version control and collaboration, allowing developers to store, manage, and track changes to their code. It uses Git, a distributed version control system, to enable multiple contributors to work on a project simultaneously. GitHub also provides features like issue tracking, pull requests, and project management tools, making it popular for open-source software development and team collaboration.

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

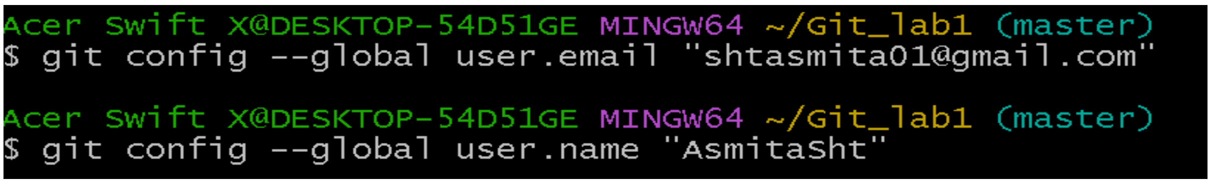
git status
Show the current state of the files in the working directory (modified, staged or untracked).
git log
Displays the commit history of the repository.

GitHub Specific

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

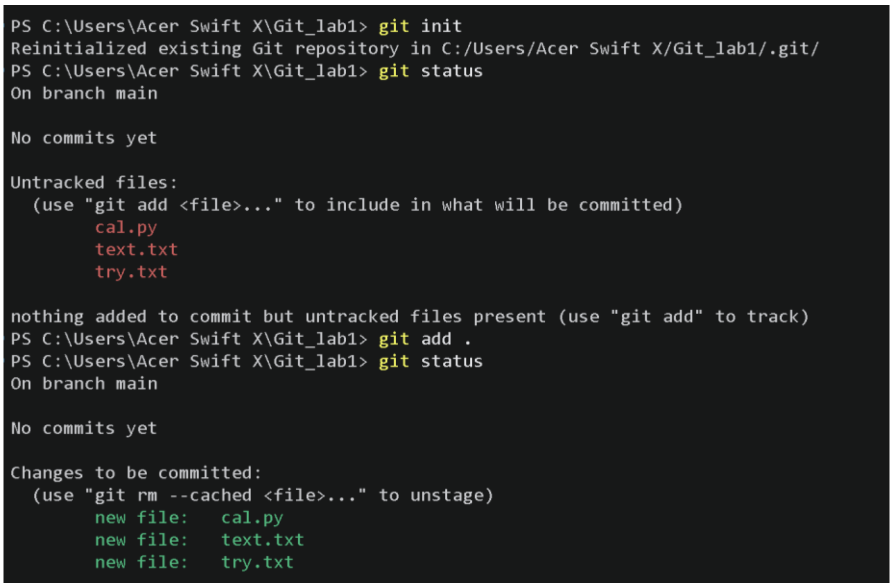
Lab Works

Set the global username and email of the GitHub.



```
Acer Swift X@DESKTOP-54D51GE MINGW64 ~/Git_lab1 (master)
$ git config --global user.email "shtasmita01@gmail.com"

Acer Swift X@DESKTOP-54D51GE MINGW64 ~/Git_lab1 (master)
$ git config --global user.name "AsmitaSht"
```

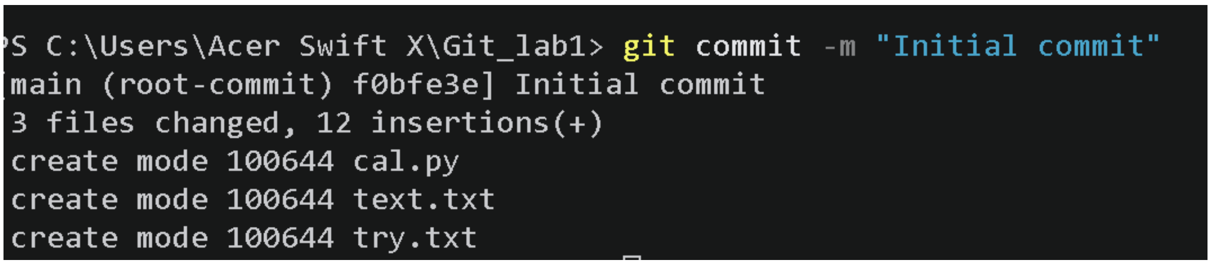
Create a folder and files as per the desire so that we can identify the changes inside the file using the version control (Git).



```
PS C:\Users\Acer Swift X\Git_lab1> git init
Reinitialized existing Git repository in C:/Users/Acer Swift X/Git_lab1/.git/
PS C:\Users\Acer Swift X\Git_lab1> git status
On branch main

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Acer Swift X\Git_lab1> git add .
PS C:\Users\Acer Swift X\Git_lab1> git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   cal.py
        new file:   text.txt
        new file:   try.txt
```

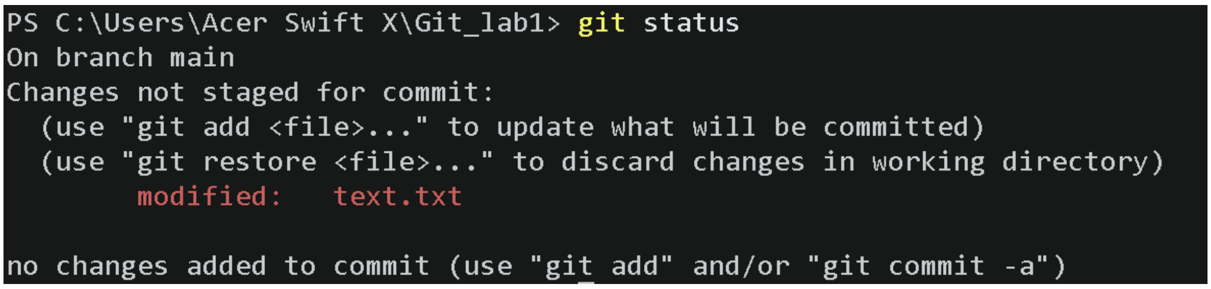
On creating the new files, initially the files are in the untracked stage so sent the untracked files to the staging stage. To do so first initialize the directory and staged the files.

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



```
PS C:\Users\Acer Swift X\Git_lab1> git commit -m "Initial commit"
[main (root-commit) f0bfe3e] Initial commit
 3 files changed, 12 insertions(+)
 create mode 100644 cal.py
 create mode 100644 text.txt
 create mode 100644 try.txt
```

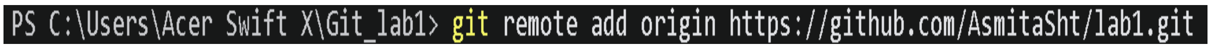
Make certain changes inside the file to see the changes in the file status.



```
PS C:\Users\Acer Swift X\Git_lab1> git status
On branch main
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")
```

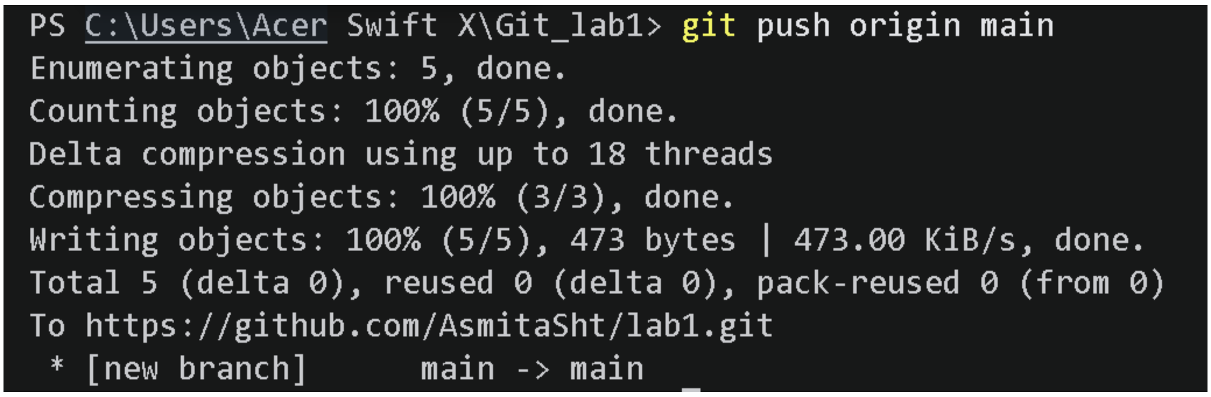
After changing the contents in the file **"text.txt"** add the file and commit it.

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.



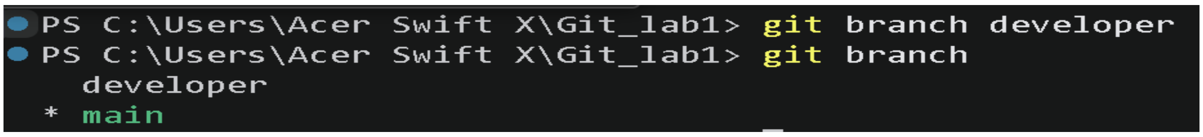
```
PS C:\Users\Acer Swift X\Git_lab1> git remote add origin https://github.com/AsmitaSht/lab1.git
```

Now push the files in the repository created.



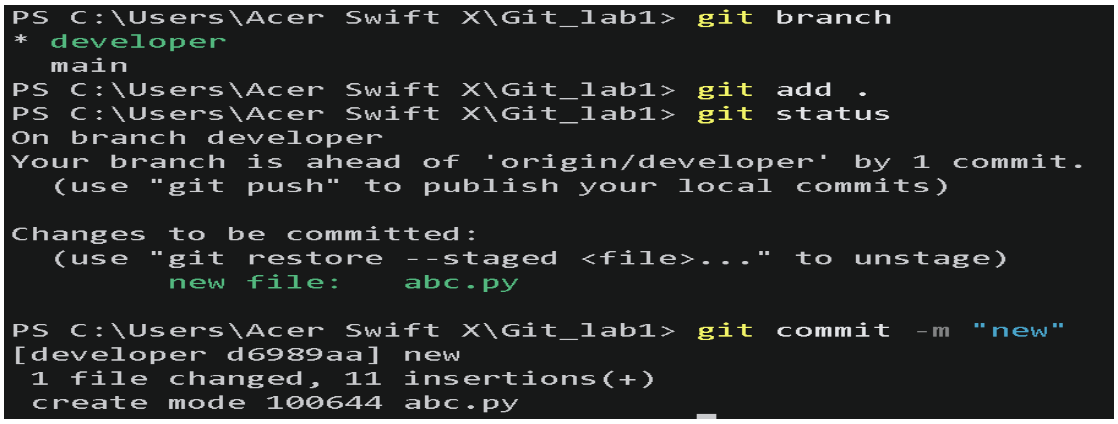
```
PS C:\Users\Acer Swift X\Git_lab1> git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 18 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 473 bytes | 473.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/AsmitaSht/lab1.git
 * [new branch]      main -> main
```

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



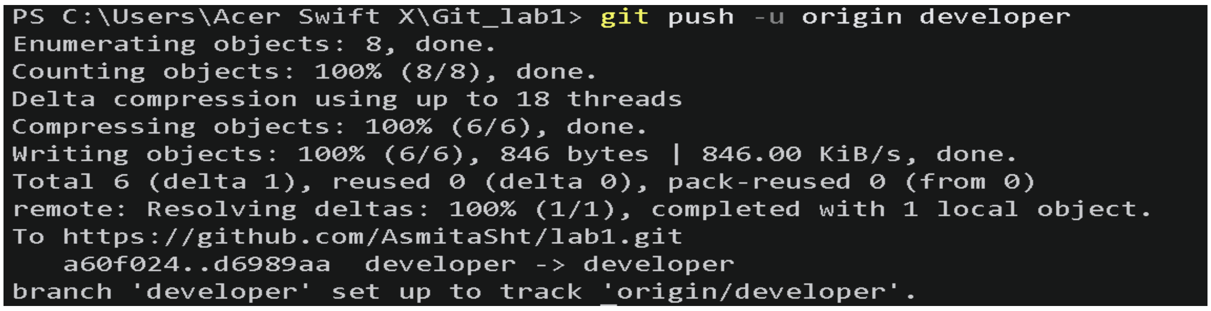
```
PS C:\Users\Acer Swift X\Git_lab1> git branch developer
PS C:\Users\Acer Swift X\Git_lab1> git branch
  developer
* main
```

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



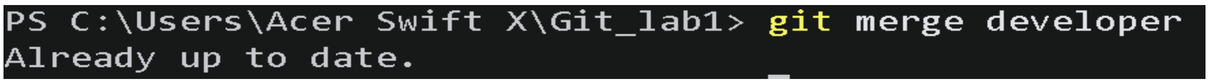
```
PS C:\Users\Acer Swift X\Git_lab1> git branch
* developer
  main
PS C:\Users\Acer Swift X\Git_lab1> git add .
PS C:\Users\Acer Swift X\Git_lab1> git status
On branch developer
Your branch is ahead of 'origin/developer' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   abc.py

PS C:\Users\Acer Swift X\Git_lab1> git commit -m "new"
[developer d6989aa] new
 1 file changed, 11 insertions(+)
 create mode 100644 abc.py
```

To change the branch, we 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.



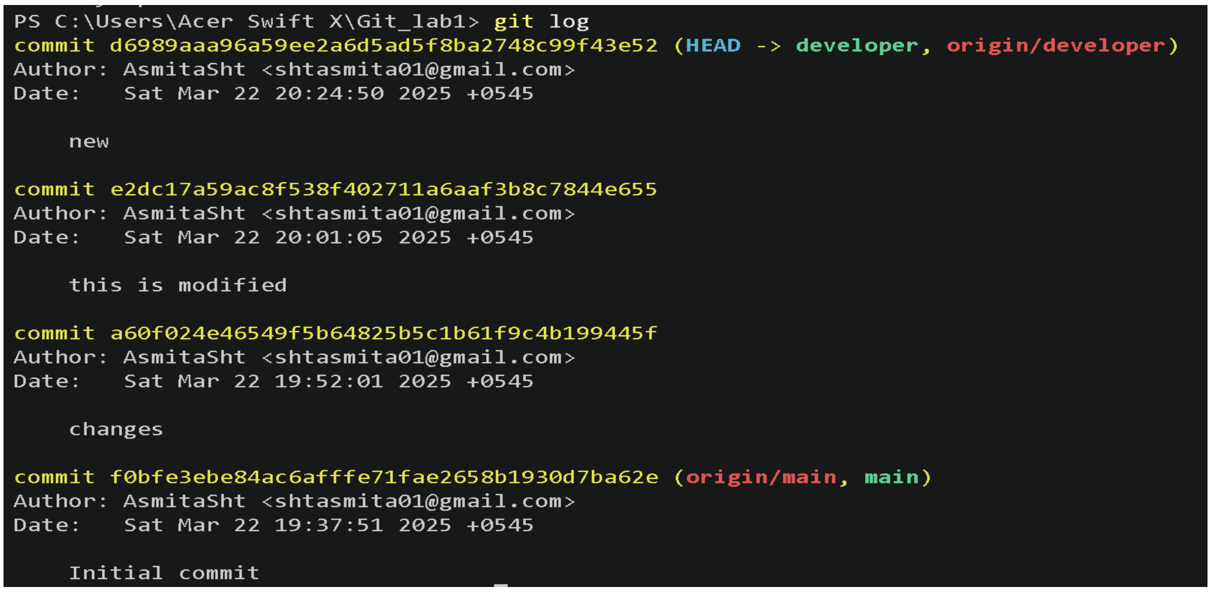
```
PS C:\Users\Acer Swift X\Git_lab1> git push -u origin developer
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 18 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 846 bytes | 846.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/AsmitaSht/lab1.git
   a60f024..d6989aa  developer -> developer
branch 'developer' set up to track 'origin/developer'.
```

Merging branches ensures that the changes or new features added in a new branch are integrated into the main codebase.



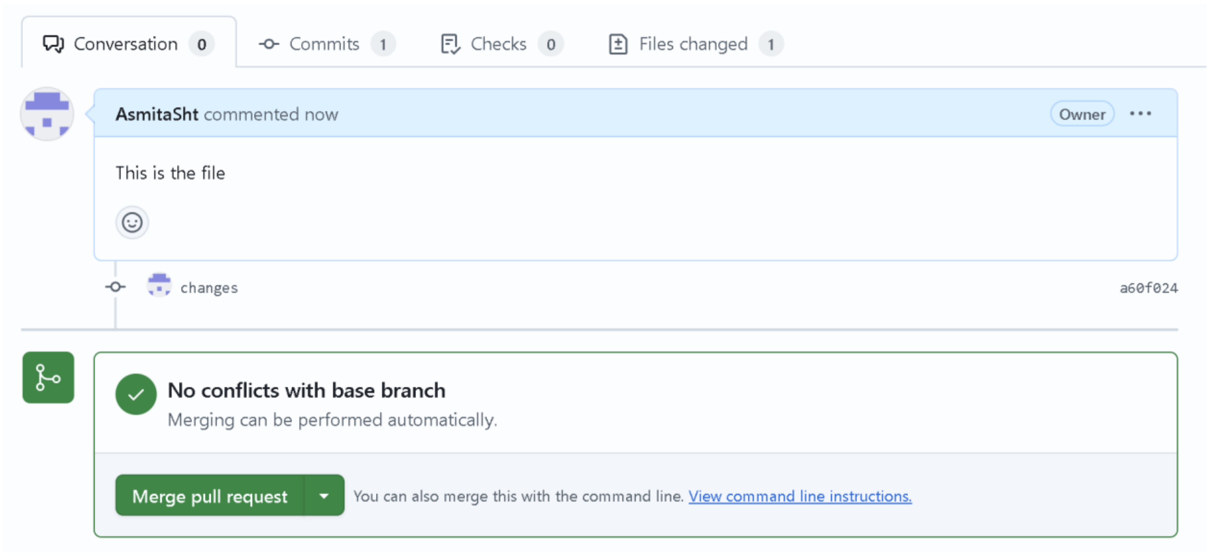
```
PS C:\Users\Acer Swift X\Git_lab1> git merge developer
Already up to date.
```

Use the ***"git log"*** command to view **past commits and their details**.



```
PS C:\Users\Acer Swift X\Git_lab1> git log
commit d6989aaa96a59ee2a6d5ad5f8ba2748c99f43e52 (HEAD -> developer, origin/developer)
Author: AsmitaSht <shtasmita01@gmail.com>
Date:   Sat Mar 22 20:24:50 2025 +0545

    new

commit e2dc17a59ac8f538f402711a6aaf3b8c7844e655
Author: AsmitaSht <shtasmita01@gmail.com>
Date:   Sat Mar 22 20:01:05 2025 +0545

    this is modified

commit a60f024e46549f5b64825b5c1b61f9c4b199445f
Author: AsmitaSht <shtasmita01@gmail.com>
Date:   Sat Mar 22 19:52:01 2025 +0545

    changes

commit f0bfe3ebe84ac6afffe71fae2658b1930d7ba62e (origin/main, main)
Author: AsmitaSht <shtasmita01@gmail.com>
Date:   Sat Mar 22 19:37:51 2025 +0545

    Initial commit
```

Merging the branch in the GitHub (Web)



AsmitaSht commented now

This is the file

changes a60f024

No conflicts with base branch

Merging can be performed automatically.

You can also merge this with the command line. View command line instructions.

Conclusion:

In this lab, we explored the fundamentals of Git and GitHub, including initialization, branching, merging, pushing, and committing. These concepts are essential for effective version control and collaborative software development.