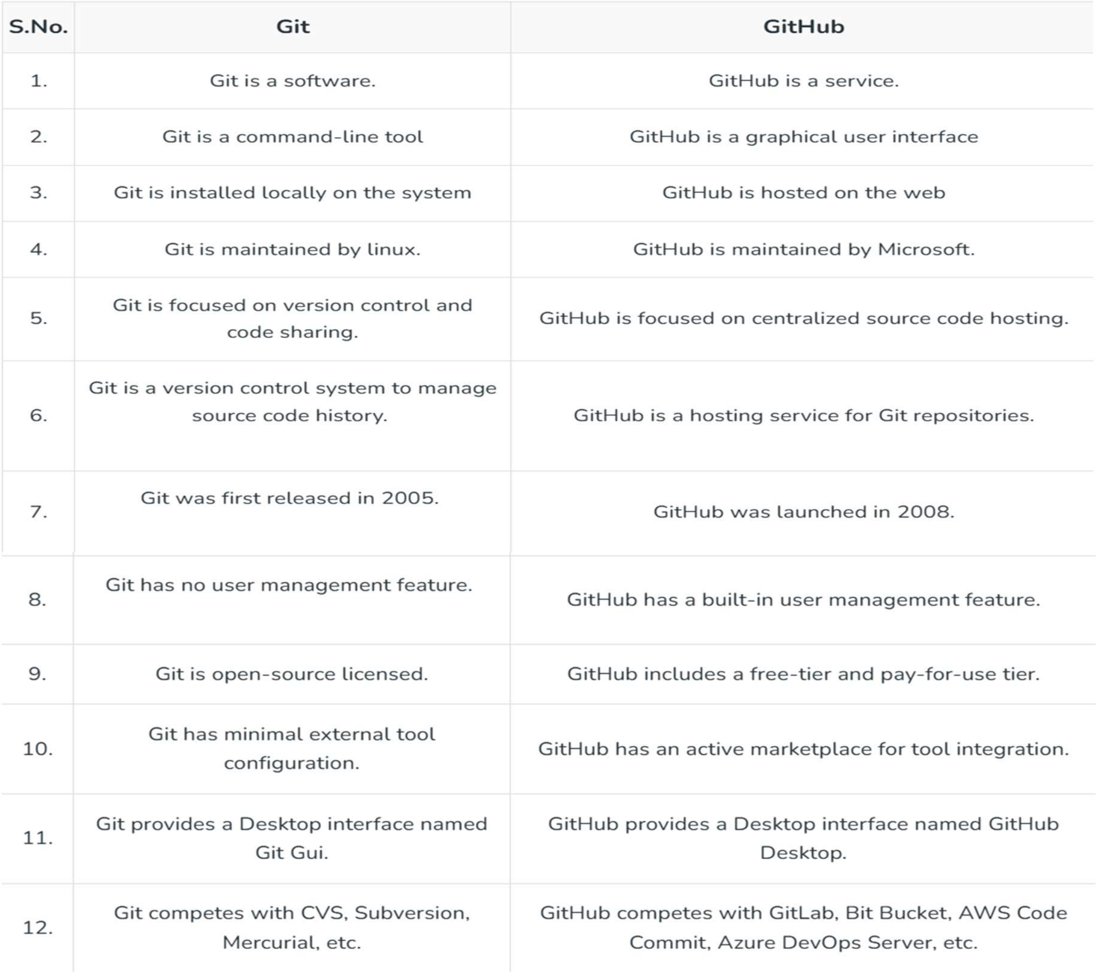
Git and GitHub

- **Git** (Version Control Tool)

  Git is a distributed version control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.

- **GitHub** (Source Code Management Tool)

  GitHub is a web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.



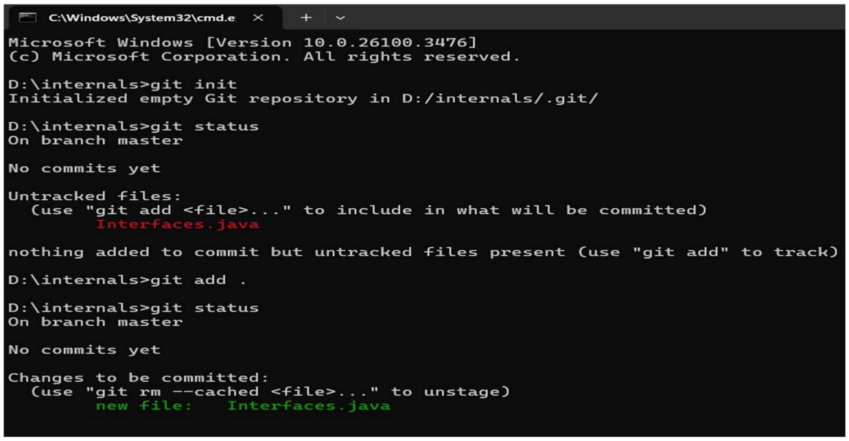
S.No.	Git	GitHub
1.	Git is a software.	GitHub is a service.
2.	Git is a command-line tool	GitHub is a graphical user interface
3.	Git is installed locally on the system	GitHub is hosted on the web
4.	Git is maintained by linux.	GitHub is maintained by Microsoft.
5.	Git is focused on version control and code sharing.	GitHub is focused on centralized source code hosting.
6.	Git is a version control system to manage source code history.	GitHub is a hosting service for Git repositories.
7.	Git was first released in 2005.	GitHub was launched in 2008.
8.	Git has no user management feature.	GitHub has a built-in user management feature.
9.	Git is open-source licensed.	GitHub includes a free-tier and pay-for-use tier.
10.	Git has minimal external tool configuration.	GitHub has an active marketplace for tool integration.
11.	Git provides a Desktop interface named Git Gui.	GitHub provides a Desktop interface named GitHub Desktop.
12.	Git competes with CVS, Subversion, Mercurial, etc.	GitHub competes with GitLab, Bit Bucket, AWS Code Commit, Azure DevOps Server, etc.

Git Commands:

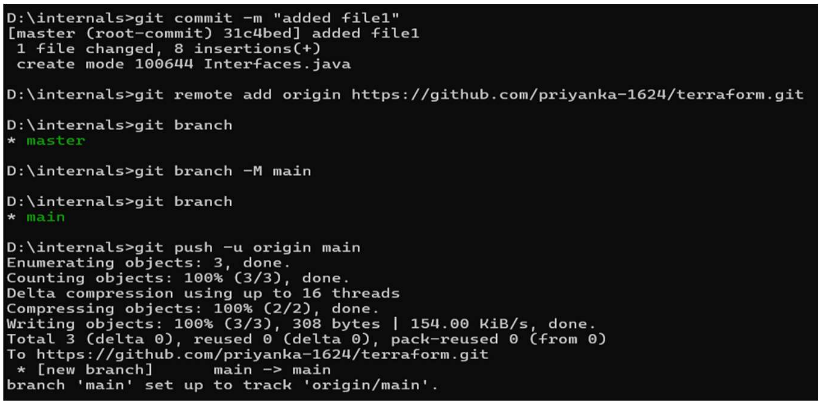
Basic commands

- **git init:** To create a local git repository for us in our store folder. This will help to manage the git commands for that particular repository.
- **git status:** To see what's changed since the last commit. It shows all the files that have been added and modified and are ready to be committed and files that are untracked.
- **git add <fille/folder>:** To add a specific file or folders to the staging area.
- **git add . :** To add all files and folders in the local repository to staging area.
- **git commit -m "message":** To commit any changes by providing a message to remember.
- **git remote add origin URL:** These commands make a bookmark which signifies that this particular remote refers to this URL. This remote will be used to pull any content from the directory and push our local content to the global server.
- **git branch -M main:** This renames the current branch (Master) to main, even if a branch with that name already exists.
- **git push -u origin main:** To push all the contents of our local repository that belong to the main branch to the server (Global repository). We can push while we are in another branch.

  NOTE: After setting a branch to upstream(-u), we can simply use git push for the next time u want to push files to the same branch in remote repository.



```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

D:\internals>git init
Initialized empty Git repository in D:/internals/.git/

D:\internals>git status
On branch master

No commits yet

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

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

D:\internals>git add .

D:\internals>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Interfaces.java
```



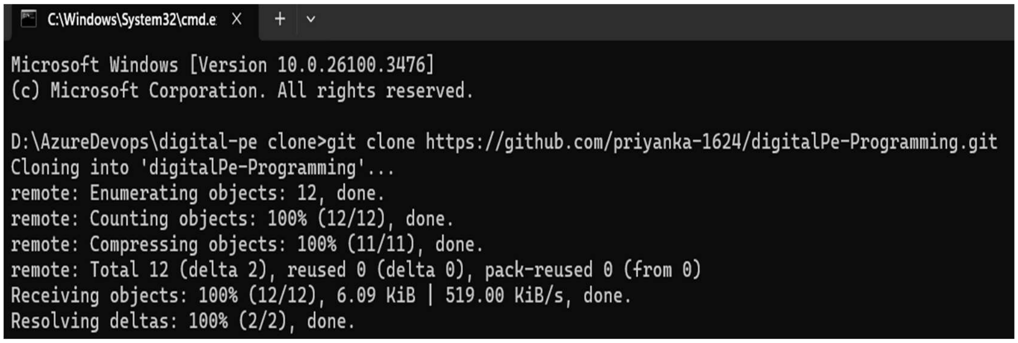
```
D:\internals>git commit -m "added file1"
[master (root-commit) 31c4bed] added file1
 1 file changed, 8 insertions(+)
 create mode 100644 Interfaces.java

D:\internals>git remote add origin https://github.com/priyanka-1624/terraform.git

D:\internals>git branch
* master

D:\internals>git branch -M main

D:\internals>git branch
* main

D:\internals>git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 308 bytes | 154.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/priyanka-1624/terraform.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Branching

- **git branch:** To see all the branches present and current branches that we are working on**.**
- **git branch <branchName>:** To create a new branch with that branch name.
- **git checkout <branchName>:** To switch to branch from another branch.
- **git log:** To check the history of commits for our reference.
- **git merge <branchName>:** To merge a branch with the current branch. That means to add content that is present in other branch but not in the current branch.

PULL and CLONE

- **git pull:** To fetch and merge latest changes from a remote repository into your local branch**. (git pull origin <branch name> or git pull)**
- **git clone URL:** To clone or make a local copy of the global repository in your system (git clone command downloads the repository and creates a remote named origin which can be checked by the command – git remote -v).



```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

D:\AzureDevops\digital-pe clone>git clone https://github.com/priyanka-1624/digitalPe-Programming.git
Cloning into 'digitalPe-Programming'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 12 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (12/12), 6.09 KiB | 519.00 KiB/s, done.
Resolving deltas: 100% (2/2), done.
```