Source Code Management

git –version :To check which version of git is present in your laptop

```
Parvesh@DESKTOP-J2IJ64N MINGW64 ~ (master)
$ git --version
git version 2.43.0.windows.1
```

git config –global user.name “type your name”

git config –global user.email “type youremail here”

```
Parvesh@DESKTOP-J2IJ64N MINGW64 ~ (master)
$ git config --global user.name "Parvesh13"

Parvesh@DESKTOP-J2IJ64N MINGW64 ~ (master)
$ git config --global user.email "parvesh1307@gmail.com"
```

Then go to any drive in your PC and create a folder in it.

-As of mine I have created a folder named “Git Folder” in E drive now I will use (cd) change directory command to change the folder location in which we will work.

```
Parvesh@DESKTOP-J2IJ64N MINGW64 ~ (master)
$ cd "E:\Git Folder"

Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder
$
```

Then we will initialize an empty Git Repository in that folder where the history of all the changes to be made by you will be stored. For initializing it we will use (git init) command.

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder
$ git init
Initialized empty Git repository in E:/Git Folder/.git/
```

You will find that there is nothing present in your folder . You can check it by using (ls) command

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ ls
```

But the file initialized has been created by the name .git and its hidden you can see it by using (ls -a) command it will show us all the hidden files.

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ ls -a
./  ../  .git/
```

We can use (ls .git) command to check what’s inside that initialized repository

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ ls -a
./  ../  .git/
```

Then we will create a file in that folder.

-For this we can generally go to the folder from library and create a file or we can use (touch names.txt) command where names.txt is your file name and I will use it you can name it as you wish.

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ touch names.txt

Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ ls
names.txt
```

Now we have created a file but when we will push this folder to git hub it will not be there as its is in unstaging area or staging area as we can see this by using command (git status)

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git status
On branch master

No commits yet

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

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

Names.txt is there in red colour because we haven't added it to staging area and for sending it to staging area we will use (git add names.txt) command

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git add names.txt
```

Now this file has been moved to staging area To commit the file as a change we can use (git commit -m "Type your message") command

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git commit -m "File Added"
[master (root-commit) 477e14d] File Added
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 names.txt
```

Now all the file present in staging area has been commited with a message (mine was File Added).

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git status
On branch master
nothing to commit, working tree clean
```

We can also check the commit history by using (git log) command

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git log
commit 477e14d0f5aa8573b6fc86f095b23bc4d07f91da (HEAD -> master)
Author: Parvesh13 <parvesh1307@gmail.com>
Date:   Sun Feb 4 16:23:08 2024 +0530

    File Added
```

We can remove file by using (rm -rf names.txt) where rm means remove ,rf means remove file and names.txt is your file name.

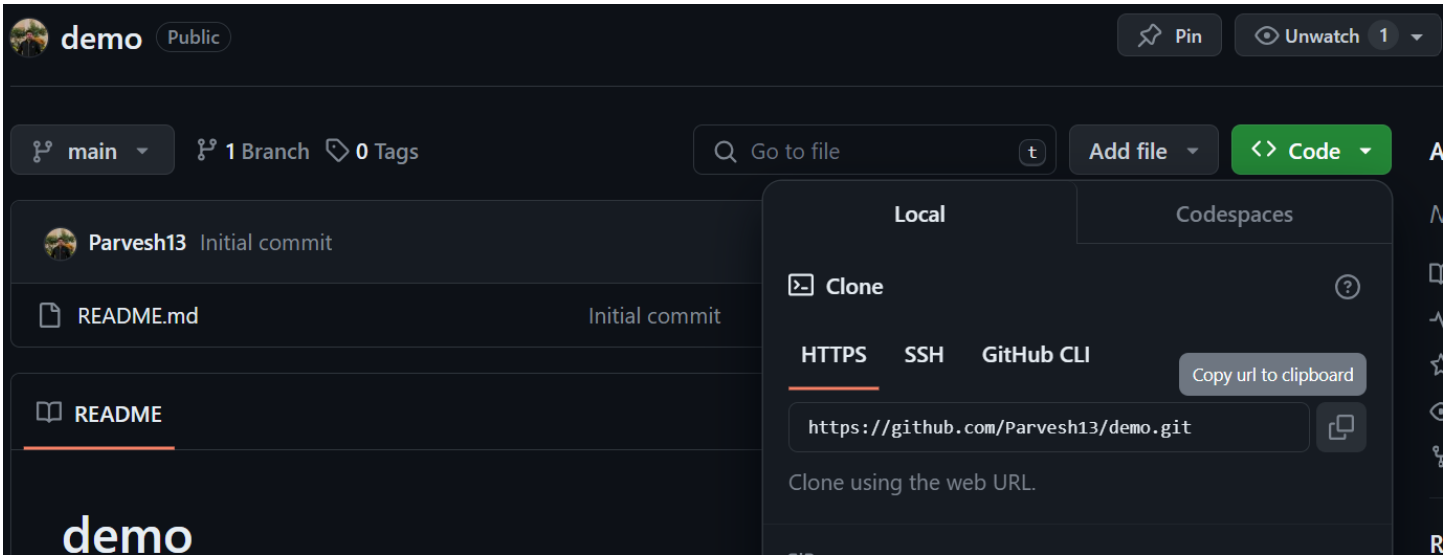
```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ rm -rf names.txt

Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    names.txt

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

But if we had done it mistakenly we can undo this by using (git restore –staged names.txt) command

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git restore --staged names.txt
```

Starting Git Hub

Go to github.com and make your account there and them make a empty repositoy with any name (I will use demo as name) and then click on Code section in your Repository and copy the url



Now we will add this remote repository with our local folder in which we have been working by using command (git remote add origin <url>) where origin is the name of url

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git remote add origin https://github.com/Parvesh13/demo.git
```

We can use (git remote -v) command to check which remote git links attached to that folder

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git remote -v
origin  https://github.com/Parvesh13/demo.git (fetch)
origin  https://github.com/Parvesh13/demo.git (push)
```

Branches

Branches are like virtual folders or pointers to all the changes .We work on different different branches in an project .There is always a main branch named Master and currently we are working on it

-To push your work on remote repository we can use (git push origin master) command where origin is name of url and master is the branch we have been working on.

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 416 bytes | 416.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/Parvesh13/demo/pull/new/master
remote:
To https://github.com/Parvesh13/demo.git
 * [new branch]      master -> master
```

We can also create our own branch by using (git branch feature) command where feature is the name we assigned to our branch (You can change it accordingly)

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git branch feature
```

Then we will use (git checkout feature) command to shift our working to that branch it means that the commits we will be doing now onwards will be in feature branch.

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git checkout feature
Switched to branch 'feature'
D       names.txt

Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (feature)
$
```

After that we can work in this branch .You can create some files and commit it like we have done before

I will be creating one file and commiting it named rollno.txt

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (feature)
$ touch rollno.txt

Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (feature)
$ git add .
```

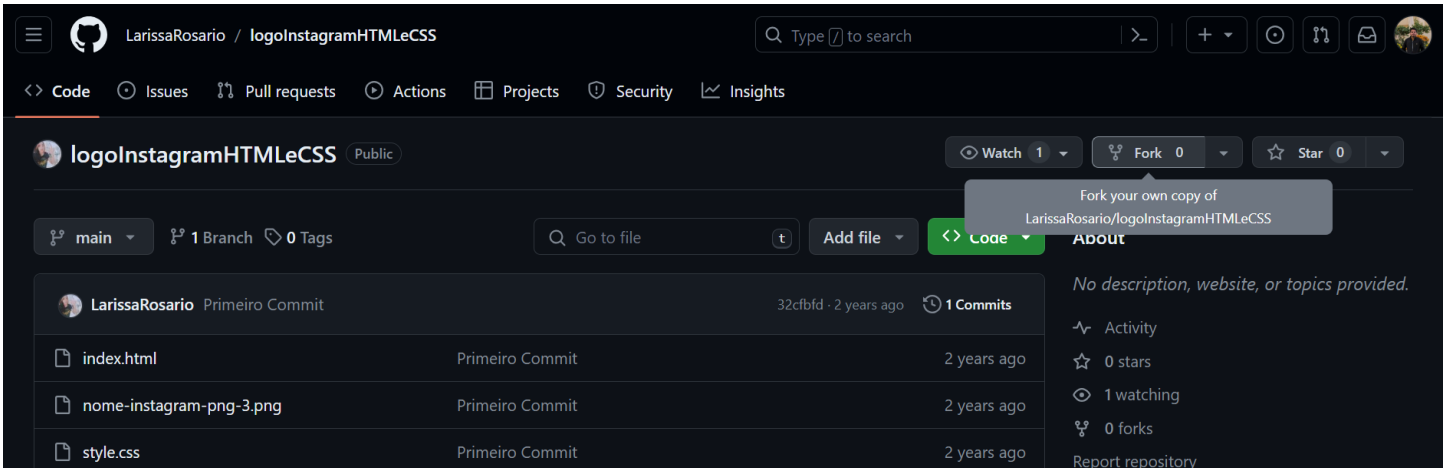
```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (feature)
$ git commit -m "File created in new branch"
[feature 39cacbf] File created in new branch
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename names.txt => rollno.txt (100%)
```

We can also merge both the branches by using (git merge feature) command

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (feature)
$ git checkout master
Switched to branch 'master'

Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git merge feature
Updating 2ddc773..39cacbf
Fast-forward
 names.txt => rollno.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 rename names.txt => rollno.txt (100%)
```

Working On Any Public Repository

For working on any remote public repository you have to firstly fork it means Go to any public repository in my case I will go to a html project repository and by using fork button I will fork it to my account as my own project



Now the project you have forked will be there in your profile as you repository and you have to clone it in your laptop by copying the url and using (git clone <url>) command

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git clone https://github.com/Parvesh13/logoInstagramHTMLeCSS.git
Cloning into 'logoInstagramHTMLeCSS'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), 59.78 KiB | 765.00 KiB/s, done.
```

We will Also add the remote repo will our folder by using (git remote origin <url>) command where origin is url name

Then we can do changes to the files we cloned or add/delete files simply as we done before and push them to the remote repository

```
Parvesh@DESKTOP-J2IJ64N MINGW64 /e/Git Folder (master)
$ git push origin master
Everything up-to-date
```

Now when you will commit changes to that repository there will be a button named create pull request by clicking on it we can request to make these changes to the main repository from where we have forked it