Experiment no. 3

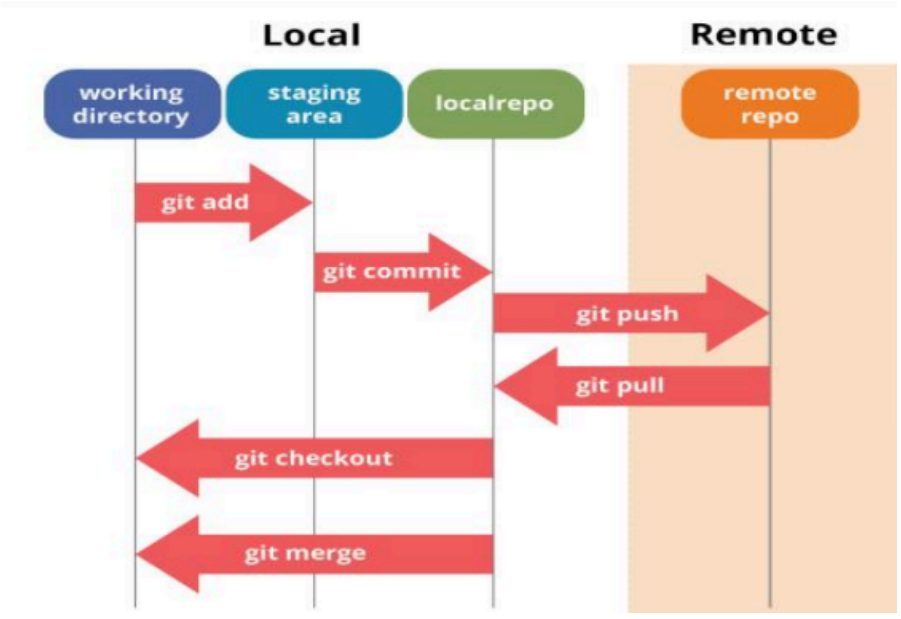
Aim: To perform various GIT operations on local and remote repositories using GIT Cheat-sheet

Theory:
Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Some of the basic operations in Git are:
1. Initialize
2. Add
3. Commit
4. Pull
5. Push

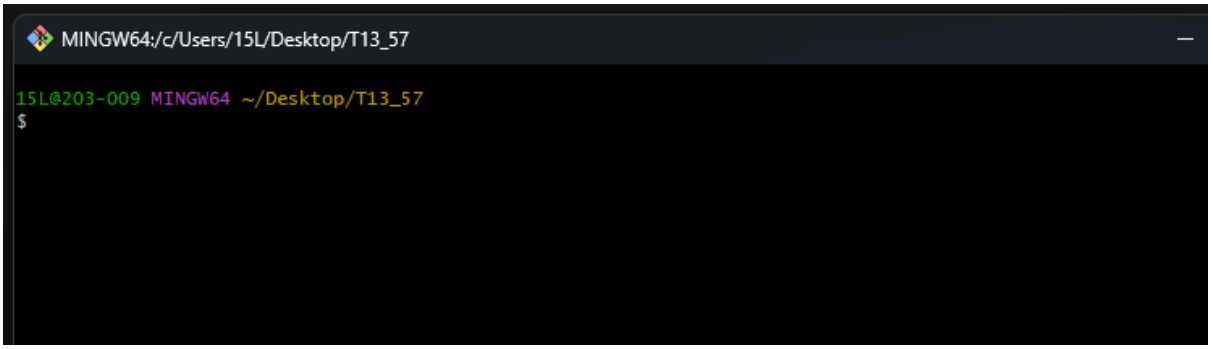
Some advanced Git operations are:
1. Branching
2. Merging
3. Rebasing

The following diagram depict the all supported operations in GIT



Implementation:

Open git bash in folder



Create a new directory

```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs
$ git config --global user.name "nidhi"

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs
$ git config --global user.email "nidhimhatre26@gmail.com"

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs
$ git config --global --list
user.name=nidhi
user.email=nidhimhatre26@gmail.com
```

```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs
$ git config --global user.name "nidhi"

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs
$ git config --global user.email "nidhimhatre26@gmail.com"

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs
$ git config --global --list
user.name=nidhi
user.email=nidhimhatre26@gmail.com

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs
$ cat ~/.gitconfig
[user]
        name = nidhi
        email = nidhimhatre26@gmail.com
```

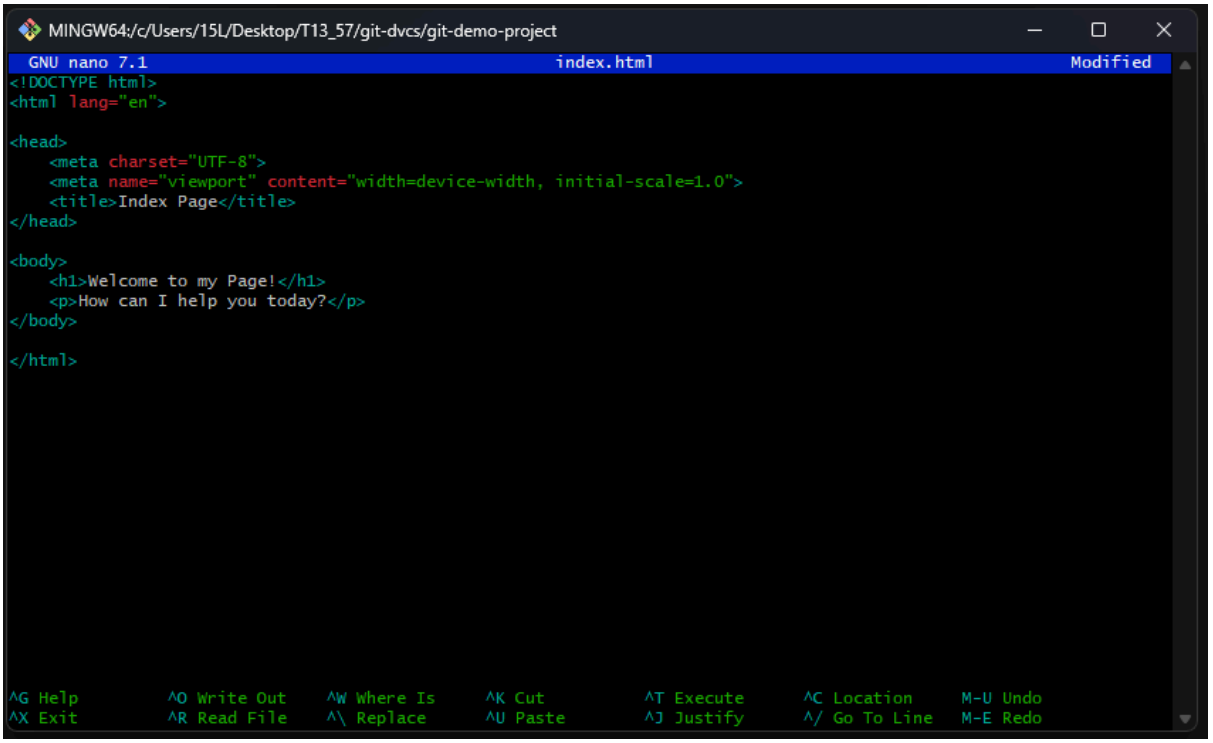
Create a demo project in the directory

```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs
$ mkdir git-demo-project

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs
$ cd git-demo-project/

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project
$ git init
Initialized empty Git repository in C:/Users/15L/Desktop/T13_57/git-dvcs/git-demo-project/.git/
```

Create a index.html file in the project and add to git

```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ ls -a
./  ../  .git/

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$  git add .

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Screenshot 2025-01-29 090523.png


15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git commit -m "First Commit"
[master (root-commit) 901c60e] First Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Screenshot 2025-01-29 090523.png

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$
```

```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git commit -m "next commit"
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html

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

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git add .

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git commit -am "Express Commit"
[master b156f7e] Express Commit
 1 file changed, 14 insertions(+)
 create mode 100644 index.html
```




```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ 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:   index.html

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

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ touch teststatus

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ 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:   index.html

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

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

Changes are Discarded by checkout
(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git checkout -- teststatus
error: pathspec 'teststatus' did not match any file(s) known to git

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git checkout -- index.html
```

```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git add index.html

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        teststatus

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

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git add teststatus

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   teststatus


15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git commit -am "Express Commit"
[master 841f404] Express Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 teststatus

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git status
On branch master
nothing to commit, working tree clean
```

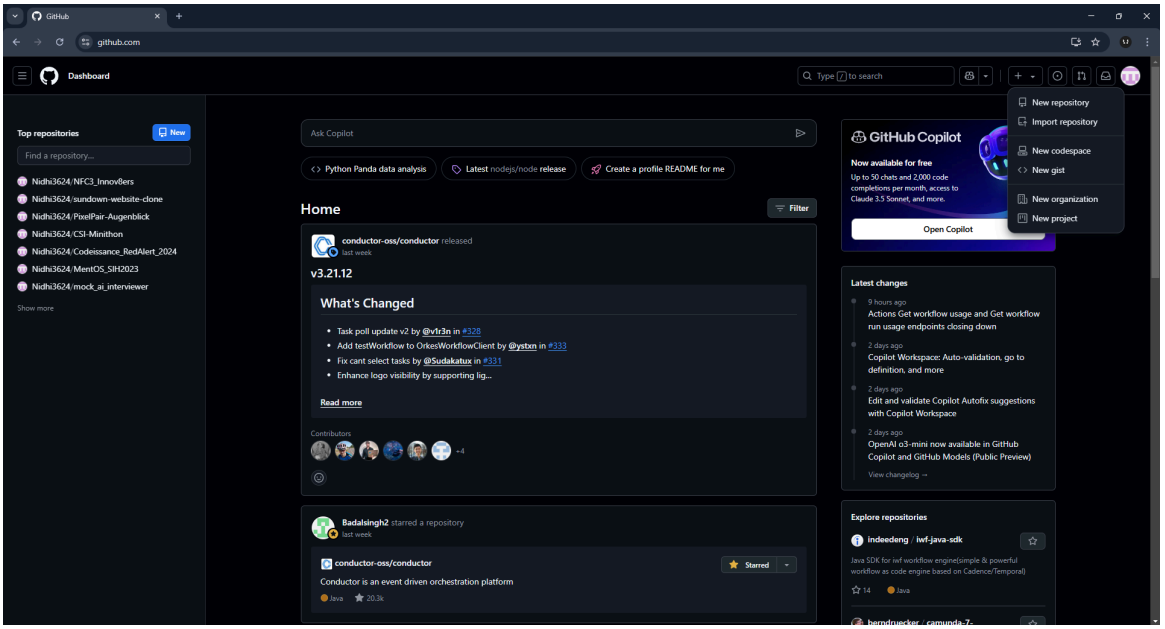
Now let us see the history of commits. The log command is used for seeing the commit history.

$ git log

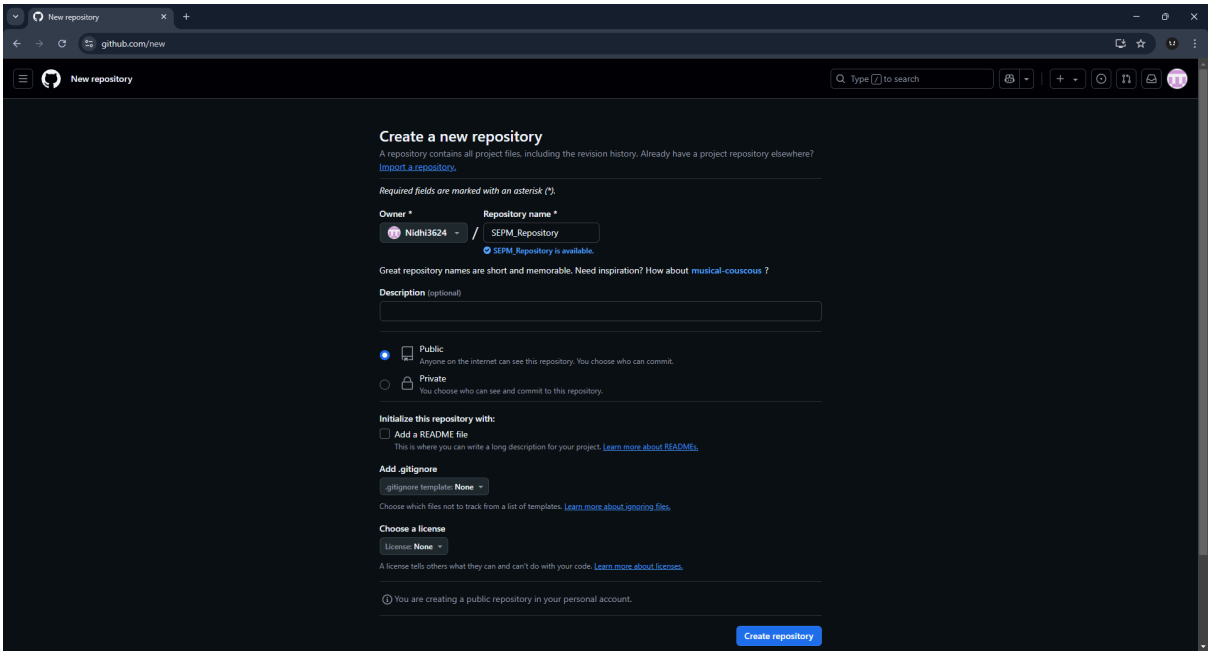
```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git log
commit 841f404a73be7b4133ef921a5c14a82ac33f905a (HEAD -> master)
Author: nidhi <nidhimhatre26@gmail.com>
Date:   Mon Feb 3 13:02:08 2025 +0530

    Express Commit

commit b156f7ead53996e672259c82937c1a04fe11acf7
Author: nidhi <nidhimhatre26@gmail.com>
Date:   Mon Feb 3 12:53:06 2025 +0530

    Express Commit

commit 901c60ed45a768fbbc650b629c5e86d14d0a1685
Author: nidhi <nidhimhatre26@gmail.com>
Date:   Mon Feb 3 12:47:16 2025 +0530

    First Commit

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git log --oneline
841f404 (HEAD -> master) Express Commit
b156f7e Express Commit
901c60e First Commit

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git log --oneline teststatus
841f404 (HEAD -> master) Express Commit

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git log --oneline -n 2
841f404 (HEAD -> master) Express Commit
b156f7e Express Commit
```

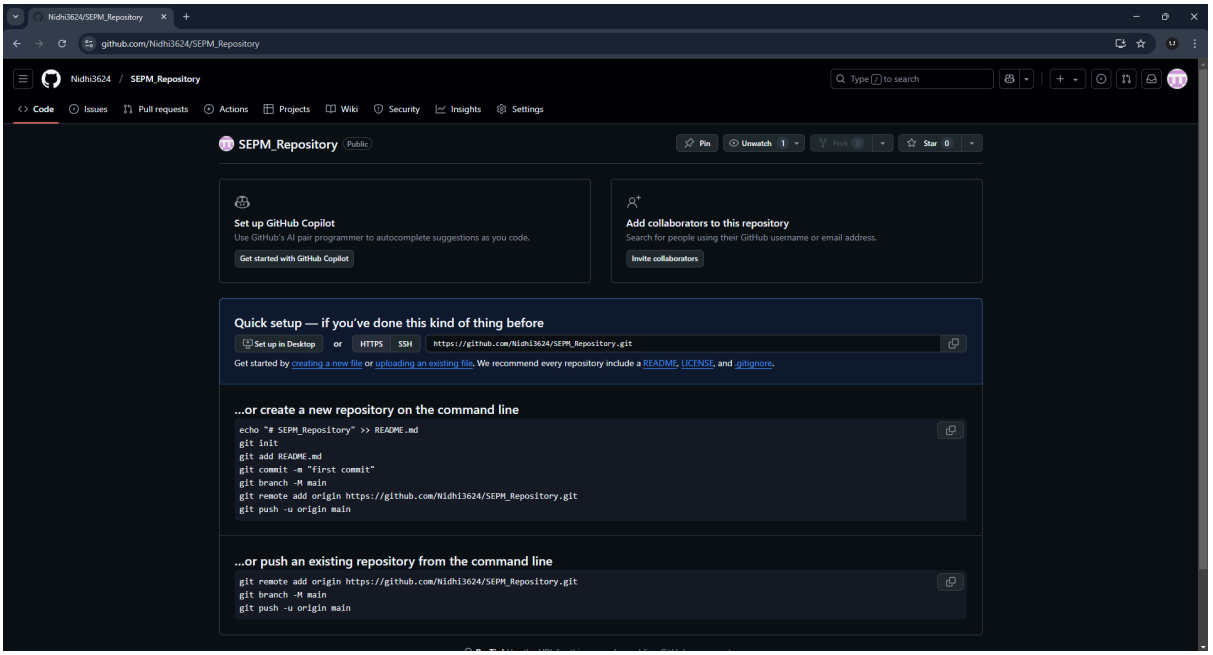
Example 2: Performing Version control in GITHUB with Pull and Push commands

Open github.com→ create an account→After login Select New repository from the menu



Now Specify a Name to repository and select public option followed by create repository

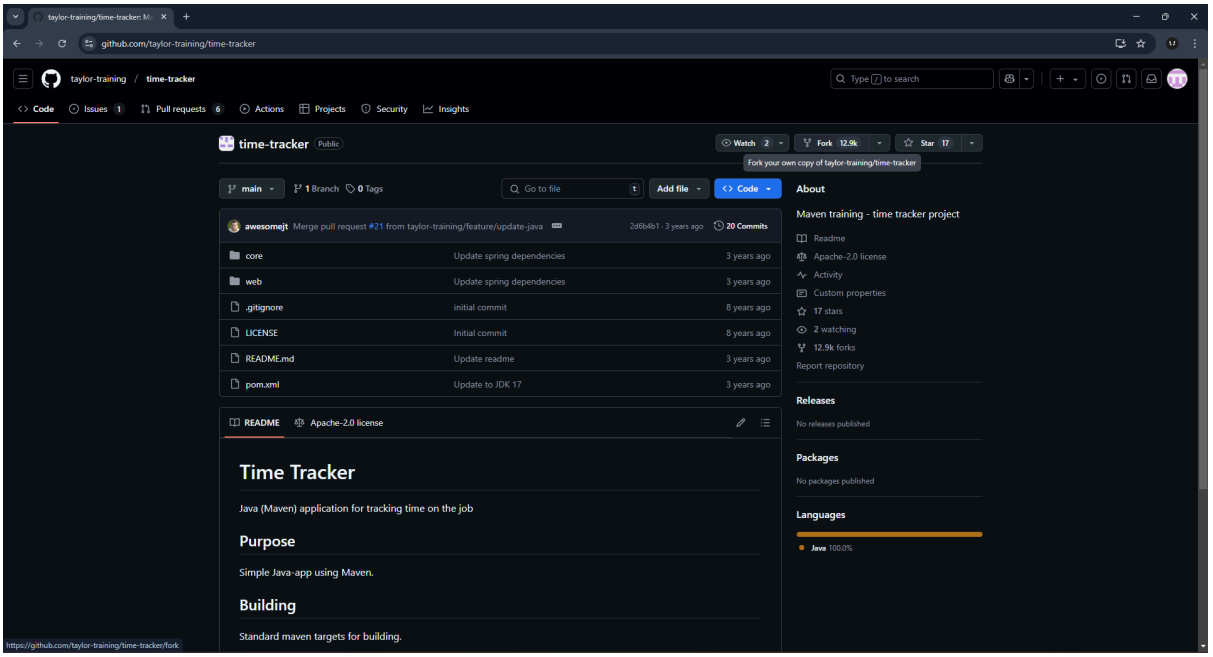




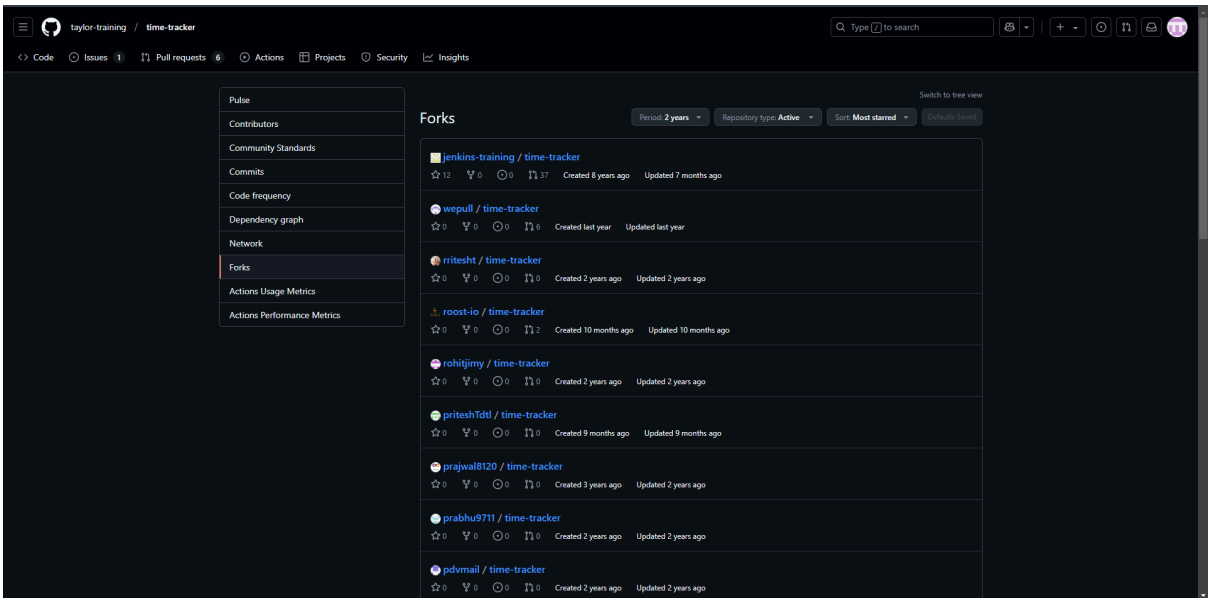
By default, we can create public repository in Github. So we can copy the entire public repository of any other users in to own account using “FORK” Operation. Now fork the repository (Sharing with other users who wants to contribute).

Login with another account→Copy and Paste URL of repository→then just click on fork to clone to others account. Suppose we want to fork public repository “timetracker”. So search for “timetracker” github repository on google and once its opened clicked on “Fork button” from the top of the github web page

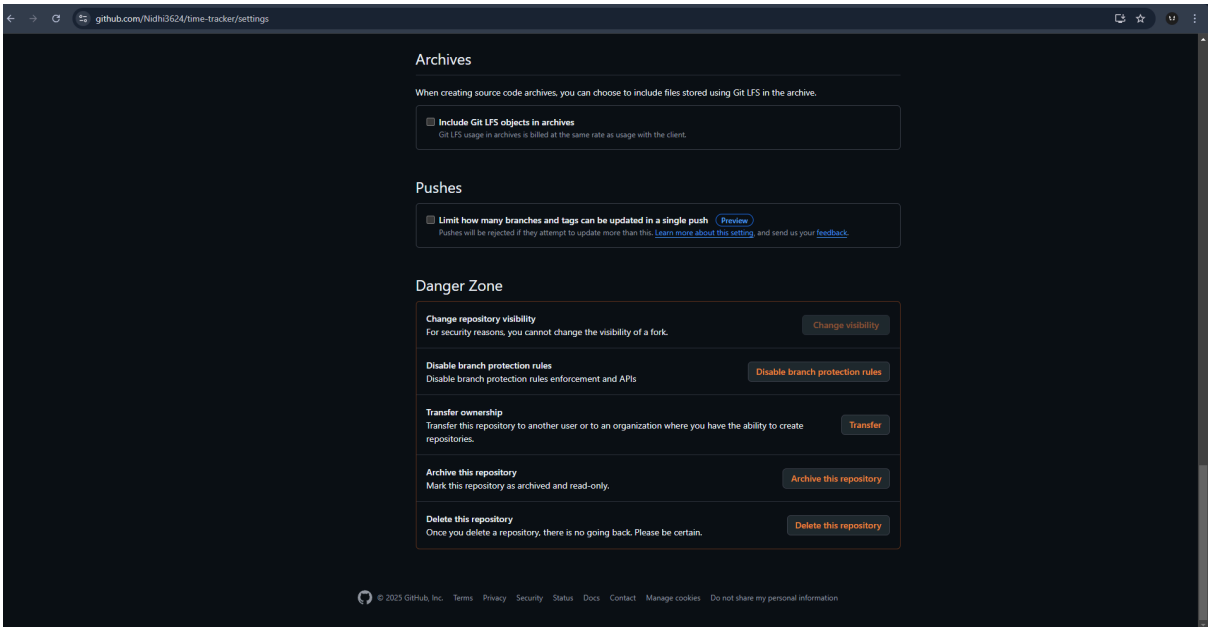
as shown below.

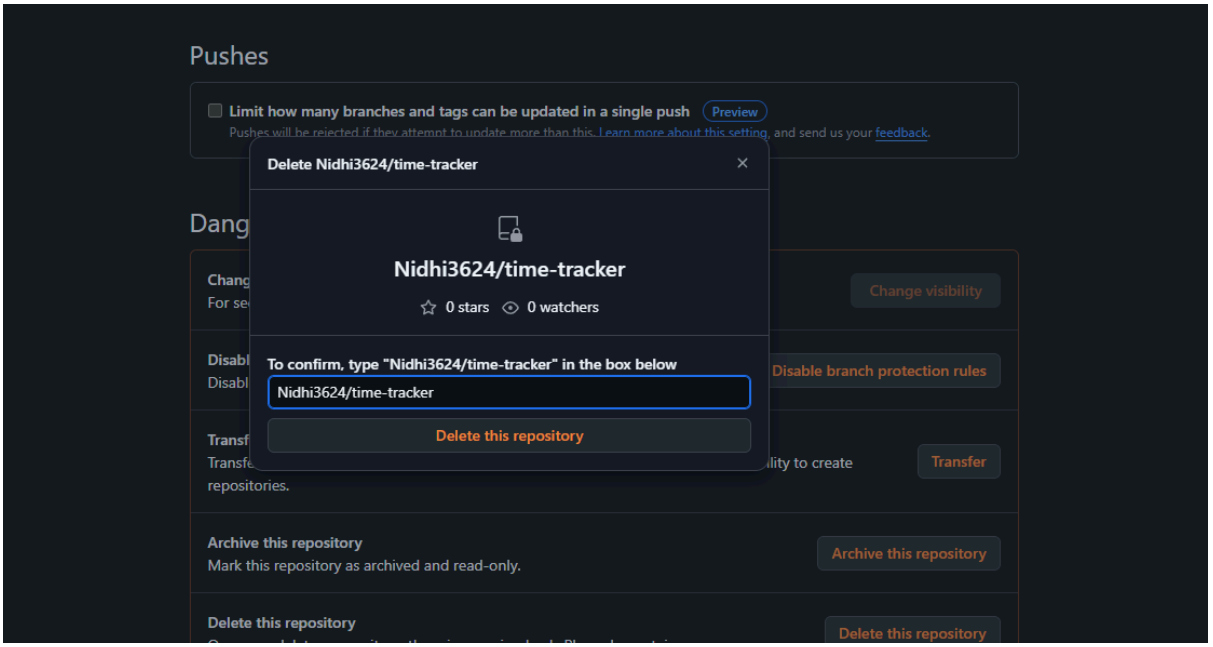


After fork it will be added in your local repository



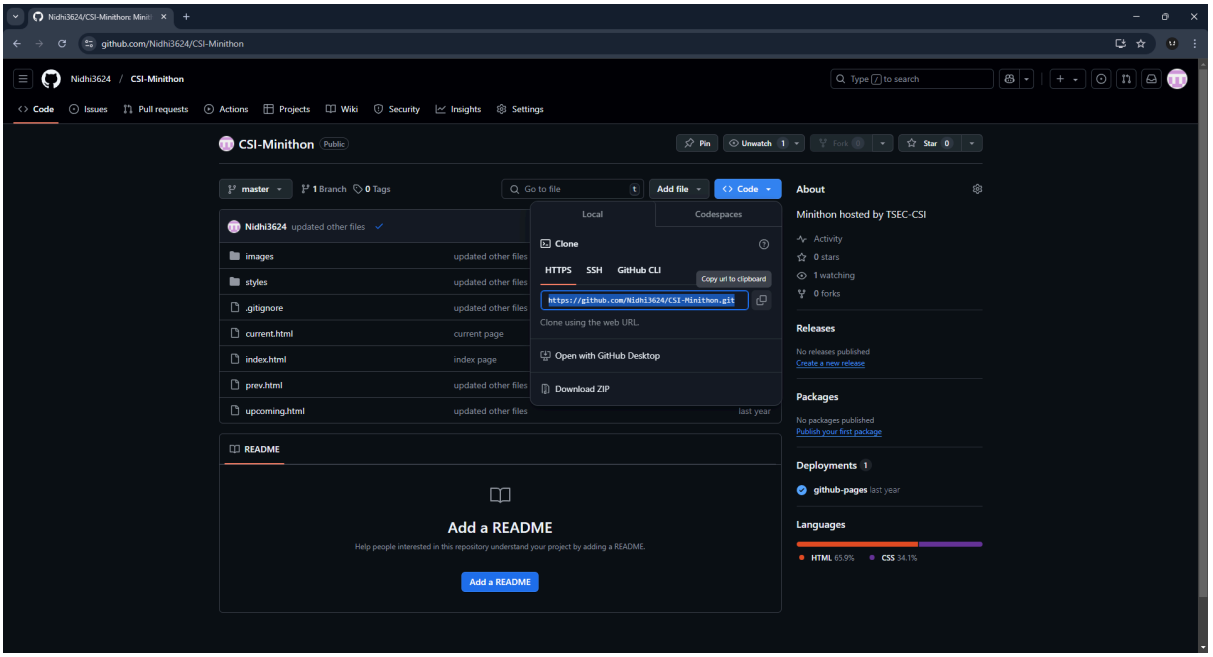
To delete the repository, open the desired repository you want to delete and go to the settings option. There you will see delete repository button to delete it.





Now, if you want to download a repository in local machine, then git clone command is used followed by path to repository. In GitHub the path of repository can be known through clone or download button and it can be

downloaded using git clone command as shown below.



Pull and Push Processes

The pull command used to fetch the repository from github to local while push is used to commit files from local repository to Github.
Push → Push changes to Web repository
Pull → Pull changes to Local repository

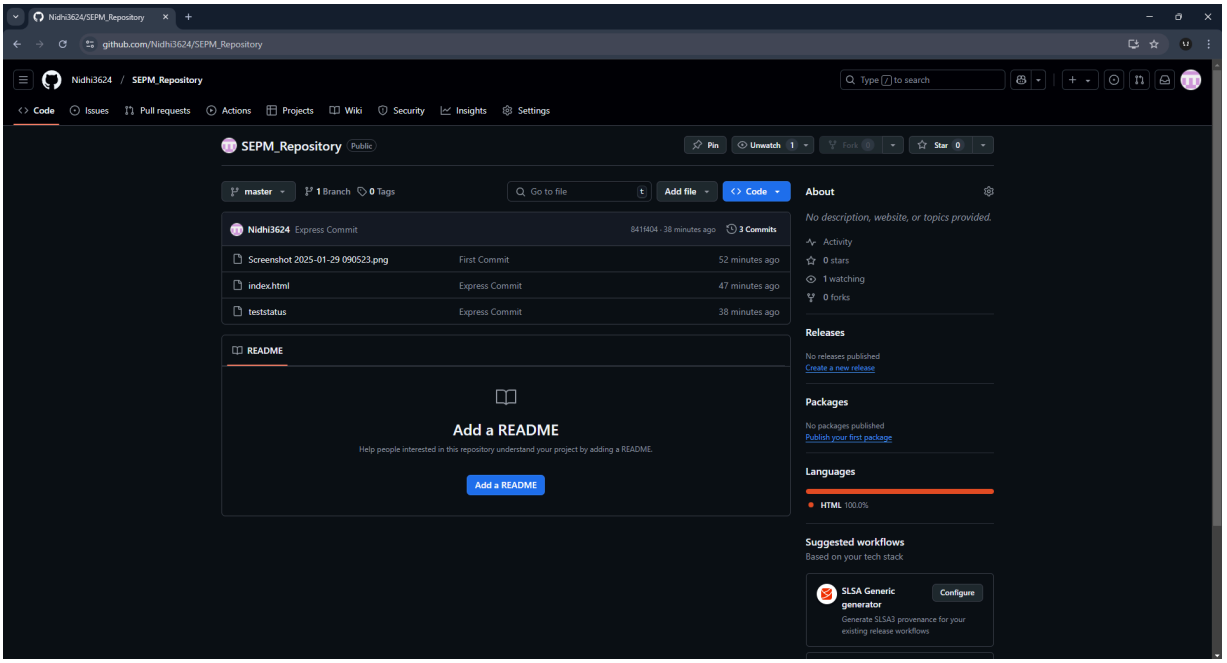
The following commands are used for pull and push repositories

A) Push command

```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git remote show origin
* remote origin
  Fetch URL: https://github.com/Nidhi3624/SEPM_Repository.git
  Push  URL: https://github.com/Nidhi3624/SEPM_Repository.git
  HEAD branch: master
  Remote branch:
    master tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)
```

```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git push -u origin master
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 20 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 73.65 KiB | 24.55 MiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/Nidhi3624/SEPM_Repository.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Now you can check the github for updated contents



B) Pull Changes

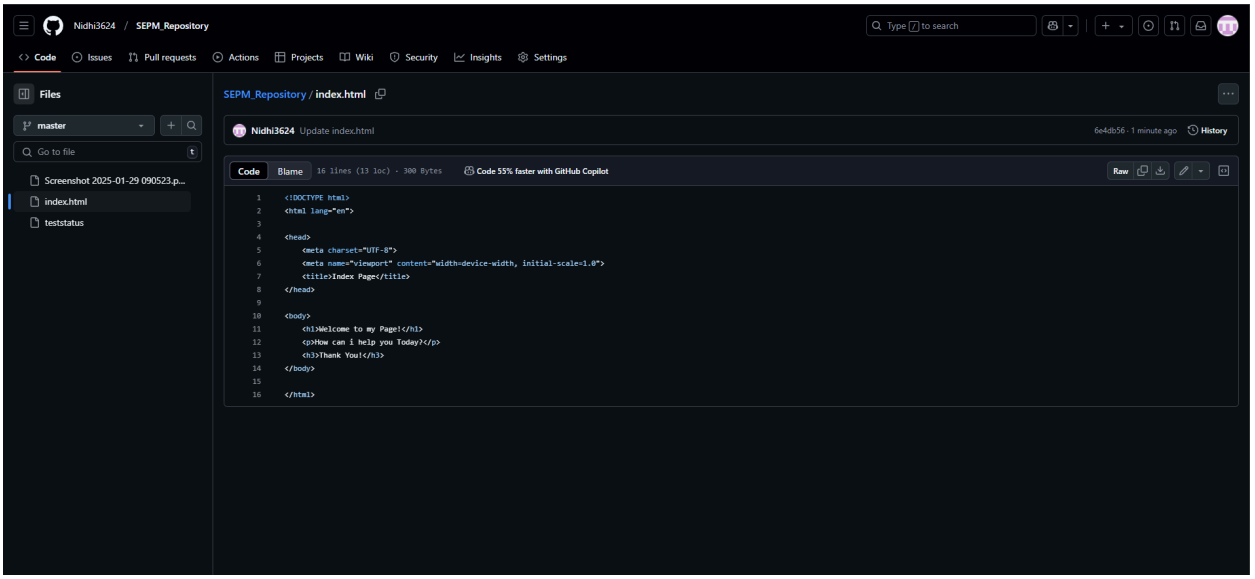
Pull command is used to download the remote updated repository into local one. The command for download is

```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1.06 KiB | 67.00 KiB/s, done.
From https://github.com/Nidhi3624/SEPM_Repository
   841f404..6e4db56  master     -> origin/master
Updating 841f404..6e4db56
Fast-forward
 index.html | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git log --oneline origin/master
6e4db56 (HEAD -> master, origin/master) Update index.html
841f404 Express Commit
b156f7e Express Commit
901c60e First Commit
```

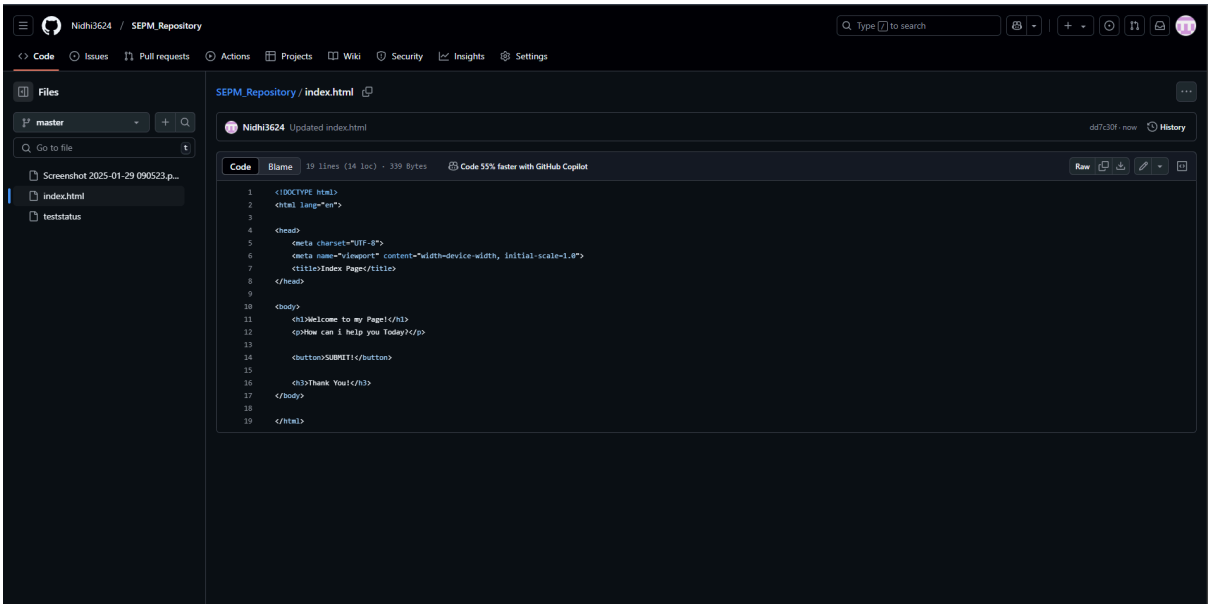
C) Fetch

Suppose you have a file in github and you have changes that

Original file:



Changed file:



Now we use fetch command to fetch the changes, which will show you both the files like original and changed in local repository

```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1.02 KiB | 61.00 KiB/s, done.
From https://github.com/Nidhi3624/SEPM_Repository
   6e4db56..dd7c30f  master     -> origin/master

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git log --oneline origin/master
dd7c30f (origin/master) Updated index.html
6e4db56 (HEAD -> master) Update index.html
841f404 Express Commit
b156f7e Express Commit
901c60e First Commit
```

Here fetch will not show you updated changes file as like push. So use the merge command to merge the changes so use the following command for merge.

$ git merge origin/master

```
15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ cat index.html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Index Page</title>
</head>

<body>
    <h1>Welcome to my Page!</h1>
    <p>How can i help you Today?</p>
    <h3>Thank You!</h3>
</body>

</html>

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ git merge
Updating 6e4db56..dd7c30f
Fast-forward
 index.html | 3 +++
 1 file changed, 3 insertions(+)

15L@203-009 MINGW64 ~/Desktop/T13_57/git-dvcs/git-demo-project (master)
$ cat index.html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Index Page</title>
</head>

<body>
    <h1>Welcome to my Page!</h1>
    <p>How can i help you Today?</p>

    <button>SUBMIT!</button>

    <h3>Thank You!</h3>
</body>

</html>
```

Conclusion: Hence we have performed various GIT operations on local and remote repositories using GIT Cheat-sheet