

Name – PRASHANT GUPTA

Topic – Git

Introduction –

What is Git?

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.

(As per the documentation on official website of git)

But first of all lets understand, **what is a version control system ?**

- ➔ Version control, also known as source control, is the practice of tracking & managing changes to software code.
- ➔ While Version control systems are software tools that help software teams manage changes to source code over time.
- ➔ Also, version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier version of the code to help fix the mistake while minimizing disruption to all team members. This is done by the concepts of branching in version control systems.

Git is also a type of version control system used by the developers for managing the development process. Git is used to track changes in source code enabling multiple developers to work together on non-linear development.

Common use case

Suppose multiple developers are working on a single project where each developer is working on a specific functionality of the software.

Here git can be used to manage the changes in the source code by individual developers & update it to the common remote repository by using the concepts of the branches for each version of the software changes.

Thus, the changes made locally will not affect the original source code directly while working on it.

Also, if there is a bug or mistake in the code after updating it to the remote repository the changes made can be reverted back using the concept of branches.

Also, we can keep the track of the details of the developers who have made a change in the original project.

How it's done?

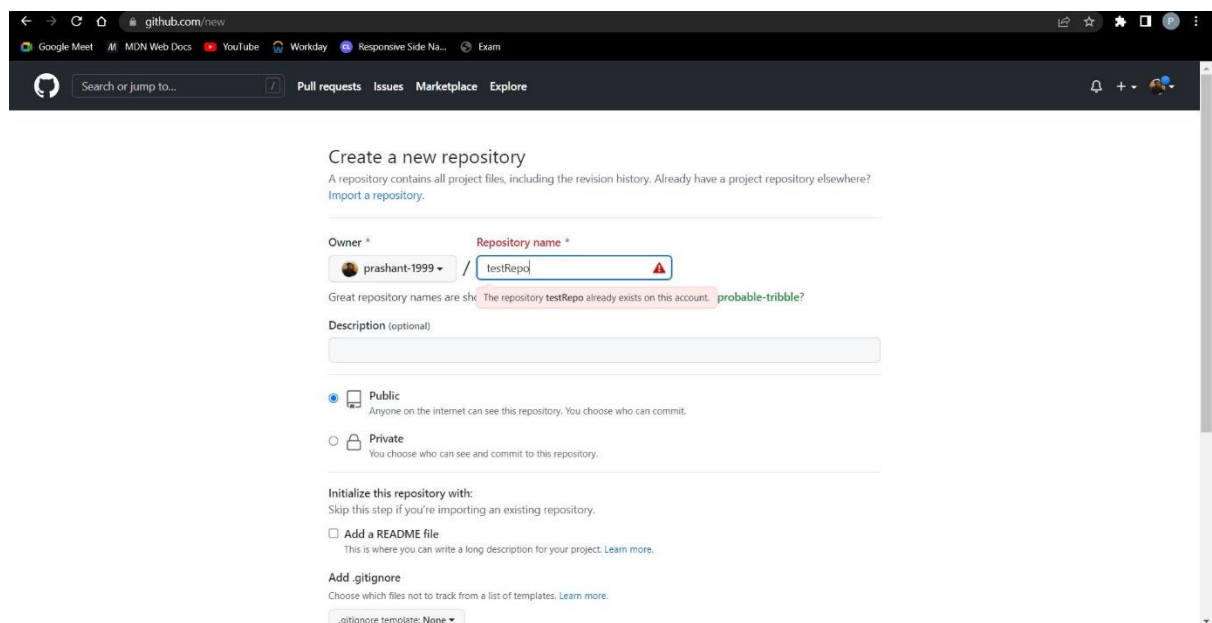
Git is used with the help of certain commands on the git bash terminal. Some of the common git bash commands are following ->

1. Git push – git push command uploads all the local commits in the project to the remote repository (in the same branch).
2. Git pull – git pull command updates the local repository with the changes in the remote repository of the project.
3. Git status – git status command shows the status of the files that whether they are in the staging area or not before commit.
4. Git init – git init command is used to initialize a local empty repository.
5. Git checkout – git checkout command is used to switch to different existing branch or to create a new branch.
6. Git add – git add command is used to add the files in the staging area and making them ready to be committed.
7. Git branch – git branch command is used to create, list out all the branches, delete & edit them.
8. Git config – This command sets the author's name and email address respectively to be used with your commits.
9. Git clone – This command is used to obtain or copy a remote repository locally.
10. Git diff – This command is used to show the differences which are not staged yet.
11. Git reset – This command is used to remove the file from staging area but preserve the content of the original file.
12. Git log – This command is used to get the version history of the current branch.
13. Git merge – This command merges the specified branch's history into the current branch.

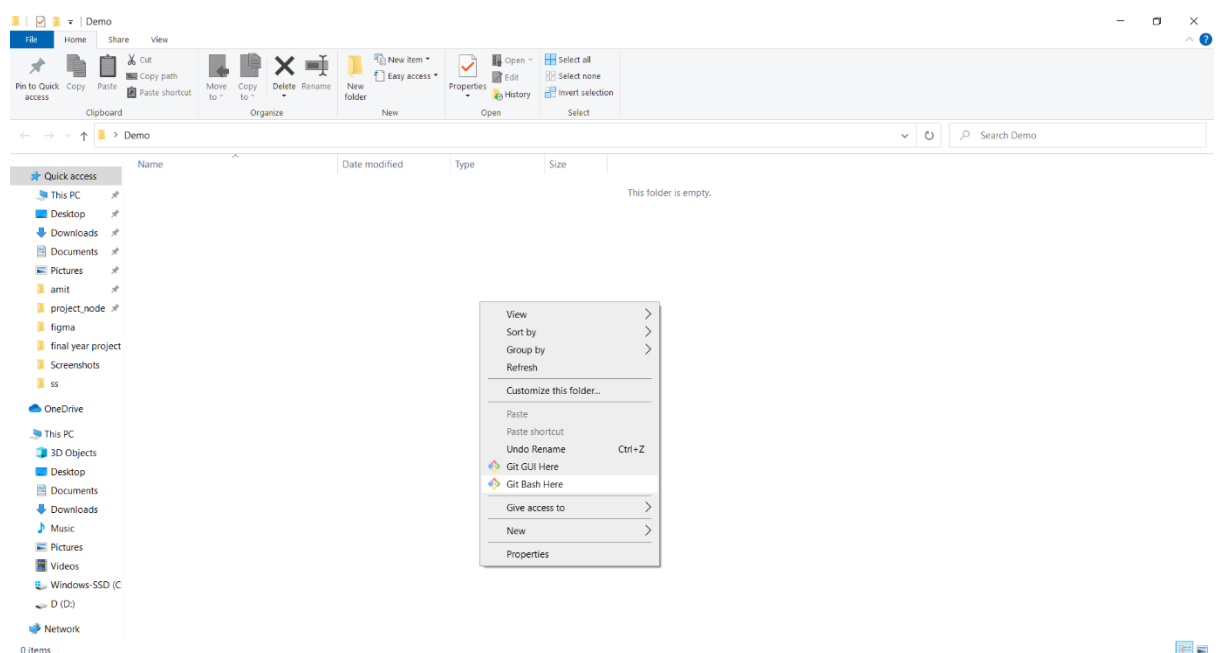
14. Git remote – This command is used to connect local repo to remote repo.
15. Git stash – This command temporarily stores all the modified tracked files.

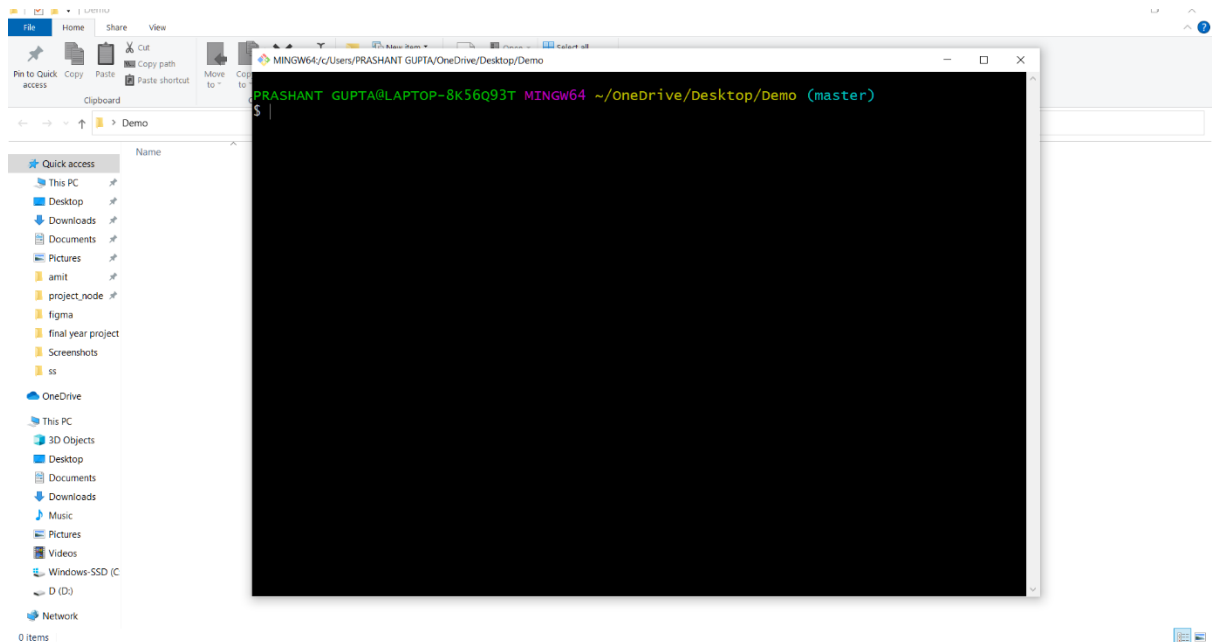
Using the **git bash terminal** for pushing the local file to remote repo ->

1. Create a github account first.
2. Download & install the git bash.
3. Create a new repository



4. Move to the folder where you want to work locally using “cd” command or just right click in the folder & select “git bash here” option.





5. Now use git commands as shown below to push the files to your remote repo

```
PRASHANT GUPTA@LAPTOP-8K56Q93T MINGW64 /d/HDFC/git/testRepo1 (master)
$ git add file.txt

PRASHANT GUPTA@LAPTOP-8K56Q93T MINGW64 /d/HDFC/git/testRepo1 (master)
$ git status
On branch master

No commits yet

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

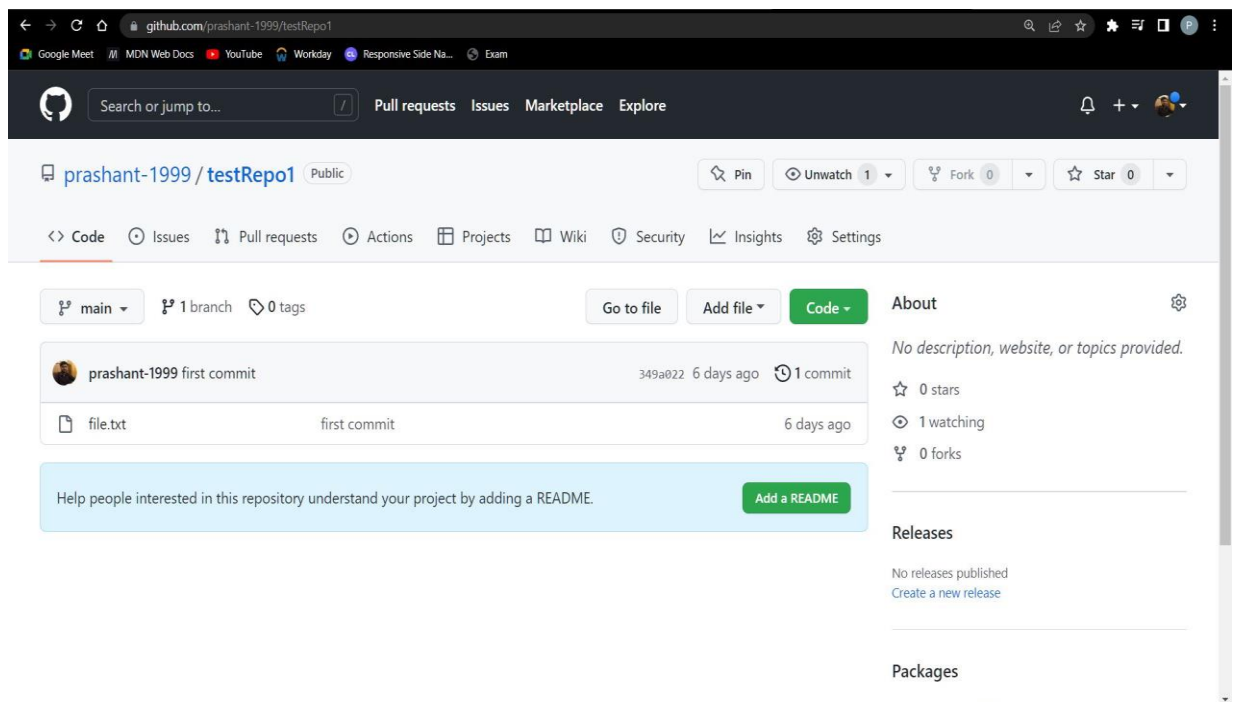
PRASHANT GUPTA@LAPTOP-8K56Q93T MINGW64 /d/HDFC/git/testRepo1 (master)
$ git commit -m "first commit"
[master (root-commit) 349a022] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.txt

PRASHANT GUPTA@LAPTOP-8K56Q93T MINGW64 /d/HDFC/git/testRepo1 (master)
$ git branch -M main

PRASHANT GUPTA@LAPTOP-8K56Q93T MINGW64 /d/HDFC/git/testRepo1 (main)
$ git remote add origin https://github.com/prashant-1999/testRepo1.git

PRASHANT GUPTA@LAPTOP-8K56Q93T MINGW64 /d/HDFC/git/testRepo1 (main)
$ git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 217 bytes | 217.00 KiB/s, done.
```

As you can below, our test file “file.txt” has been pushed to the remote repository on github



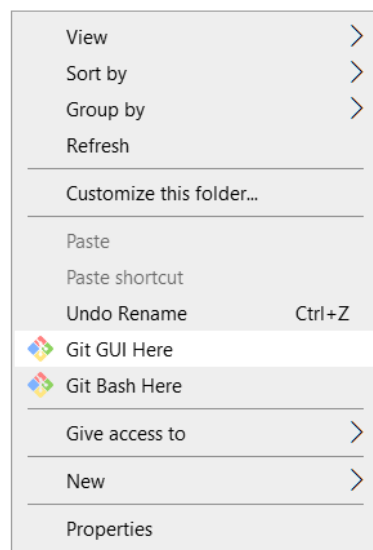
Now to copy a repository locally we can use following commands as shown –

```
MINGW64:/c:/Users/PRASHANT GUPTA/OneDrive/Desktop/Demo
PRASHANT GUPTA@LAPTOP-8K56Q93T MINGW64 ~/OneDrive/Desktop/Demo (master)
$ git pull https://github.com/prashant-1999/testRepo1.git
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 197 bytes | 16.00 KiB/s, done.
From https://github.com/prashant-1999/testRepo1
* branch      HEAD       -> FETCH_HEAD
Created autostash: 7c8786e
Applied autostash.
Successfully rebased and updated refs/heads/master.

PRASHANT GUPTA@LAPTOP-8K56Q93T MINGW64 ~/OneDrive/Desktop/Demo (master)
$ |
```

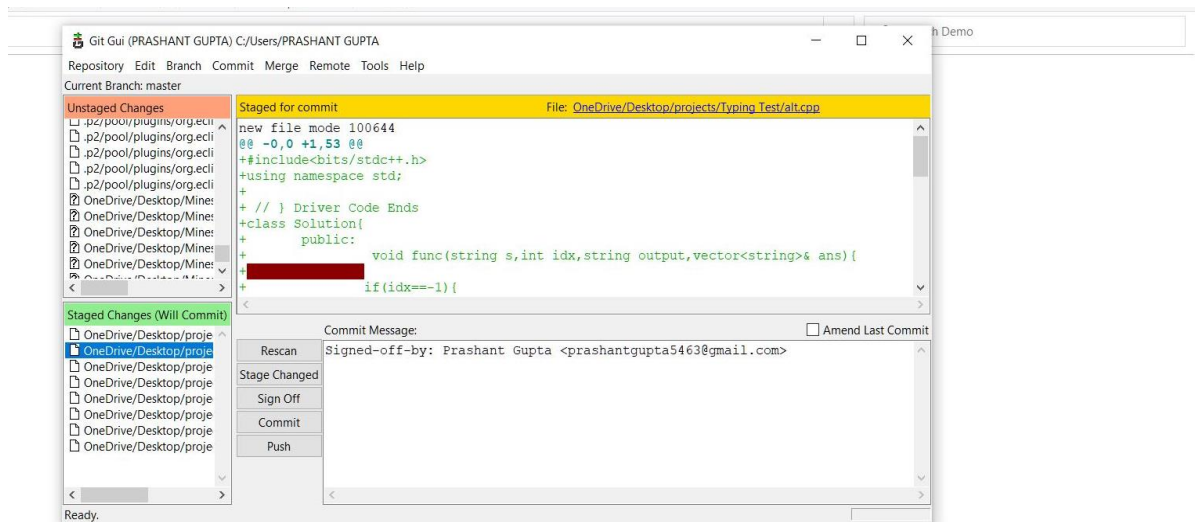
Using **Git GUI** →

1. Now after installation git & creating an account on github, lets open git gui by right clicking inside the local directory and selecting the “Git GUI here” option.



2. Now with the different menus as shown we can perform different operation like clone, push, staging, add, commit, pull, etc.

Here we don't have to use the command line rather we have the Git GUI for that.



3. We can add, commit & then push using this dialog box inside Git GUI --

