GIT&GITHUB

What is Git?

- Git is DeVops tool used for source code management.
- It is a free and open-source version control system
- It is used to handle small to very large projects efficiently.
- It helps you keep track of code changes.
- It is used to collaborate on code.

Why do we need git and github?

- To Tracking code and changes. Tracking who made changes. Coding collaboration.
- To enabling multiple developers to work together on non-linear development.

Advantages of Git:

- Good distributed model as each developer gets a local repository with a full history of commits which makes git fast compared to other VCs.
- Branching capabilities and merging are easy, good data integrity.
- They are free and open-source. We can easily download the source code and perform changes to it. They can handle larger projects efficiently.

Disadvantages of Git:

- GIT requires technical excellence and it is slower on windows. They have tedious command lines to input and don't track renames.
- It lacks window support and doesn't track empty folders.

- They do not provide access control mechanisms in case of security.

What is the version control system?

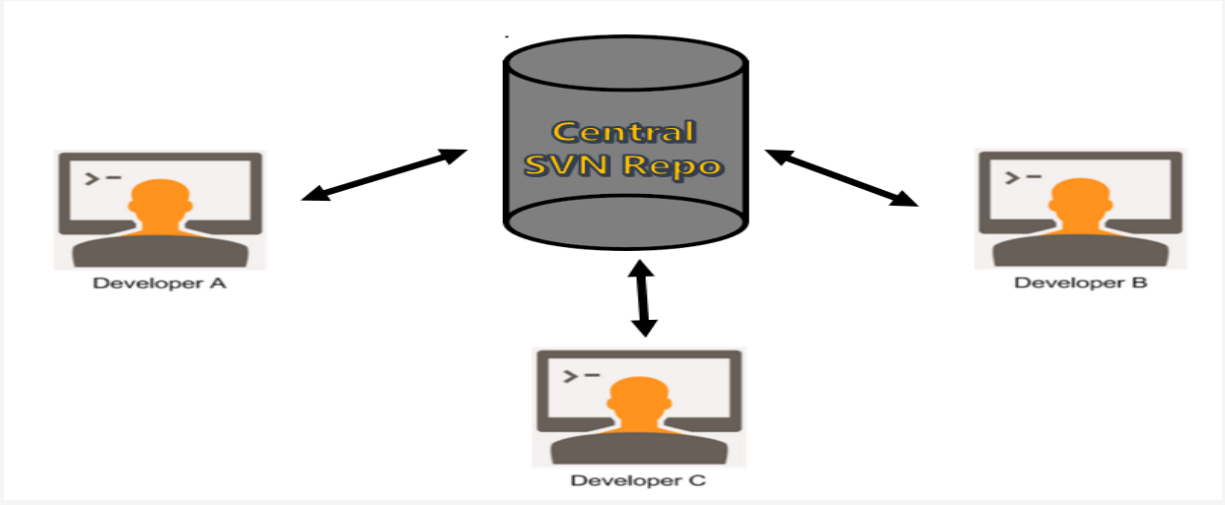
- It is the practice of tracking and managing changes to software code.
- Version control software keeps track of every modification to the code in a special kind of database.
- A version control system (VCS) tracks changes to a file or set of files over time. The most common type is a centralized VCS, which uses a server to store all the versions of a file. Developers can check out a file from the server, make changes, and check the file back in. The server then stores the new version of the file.

There are two types of version control:

Centralized and Distributed:

Centralized version control:

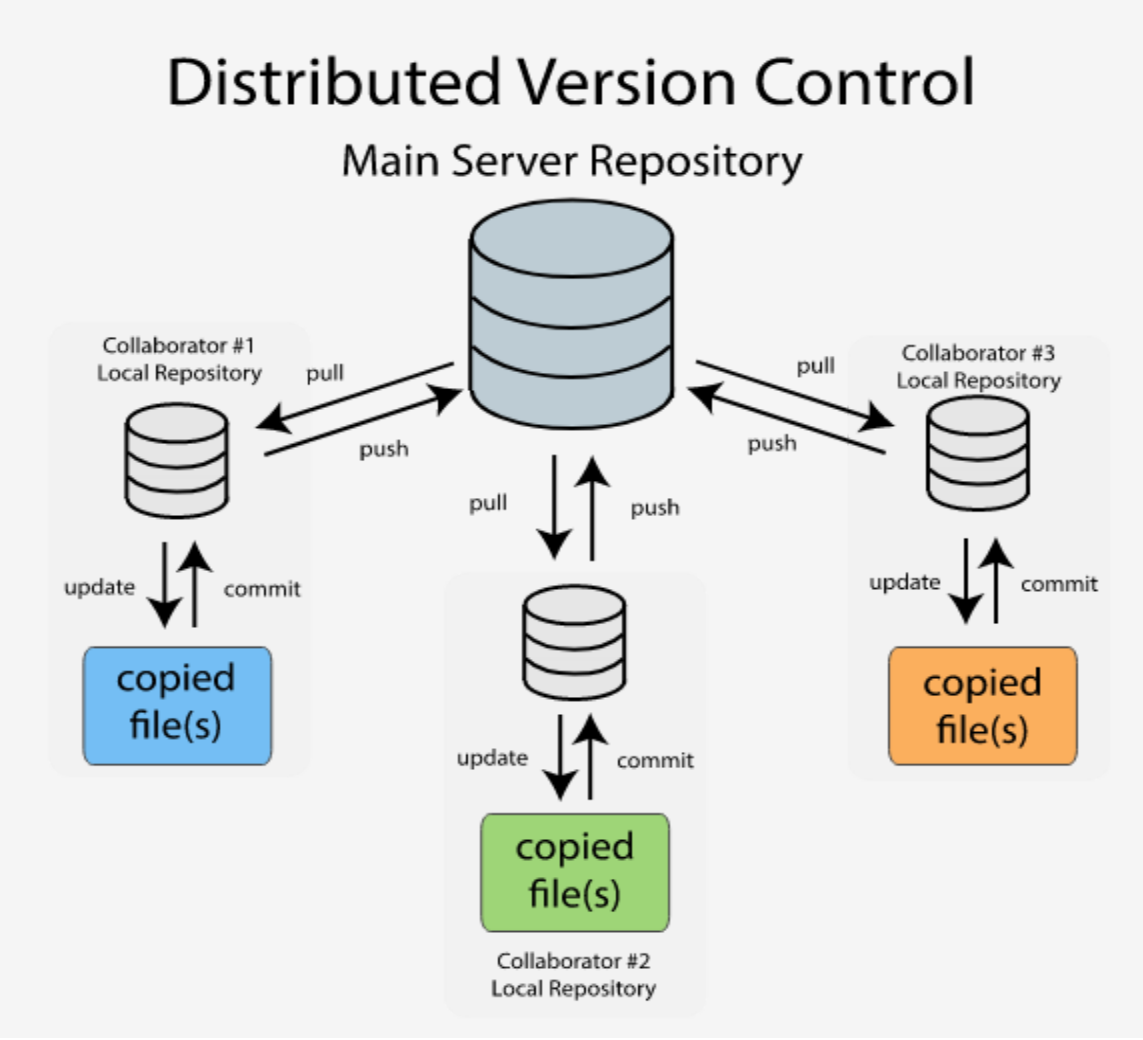
With centralized version control systems, you have a single "central" copy of your project on a server and commit your changes to this central copy. You pull the files that you need, but you never have a full copy of your project locally. Some of the most common version control systems are centralized, including Subversion(SVN) and Perforce.



Distributed version control:

With distributed version control systems (DVCS), you don't rely on a central server to store all the versions of a project's files. Instead, you clone a copy of a repository locally so that you have the full history of the project. Two common distributed version control systems are Git and Mercurial.

While you don't have to have a central repository for your files, you may want one "central" place to keep your code so that you can share and collaborate on your project with others.

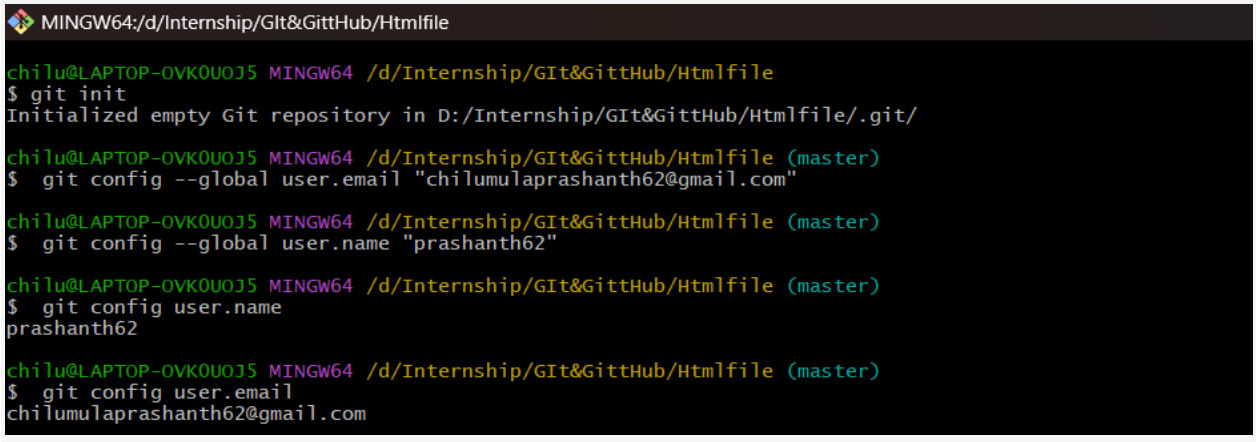


What is collaboration?

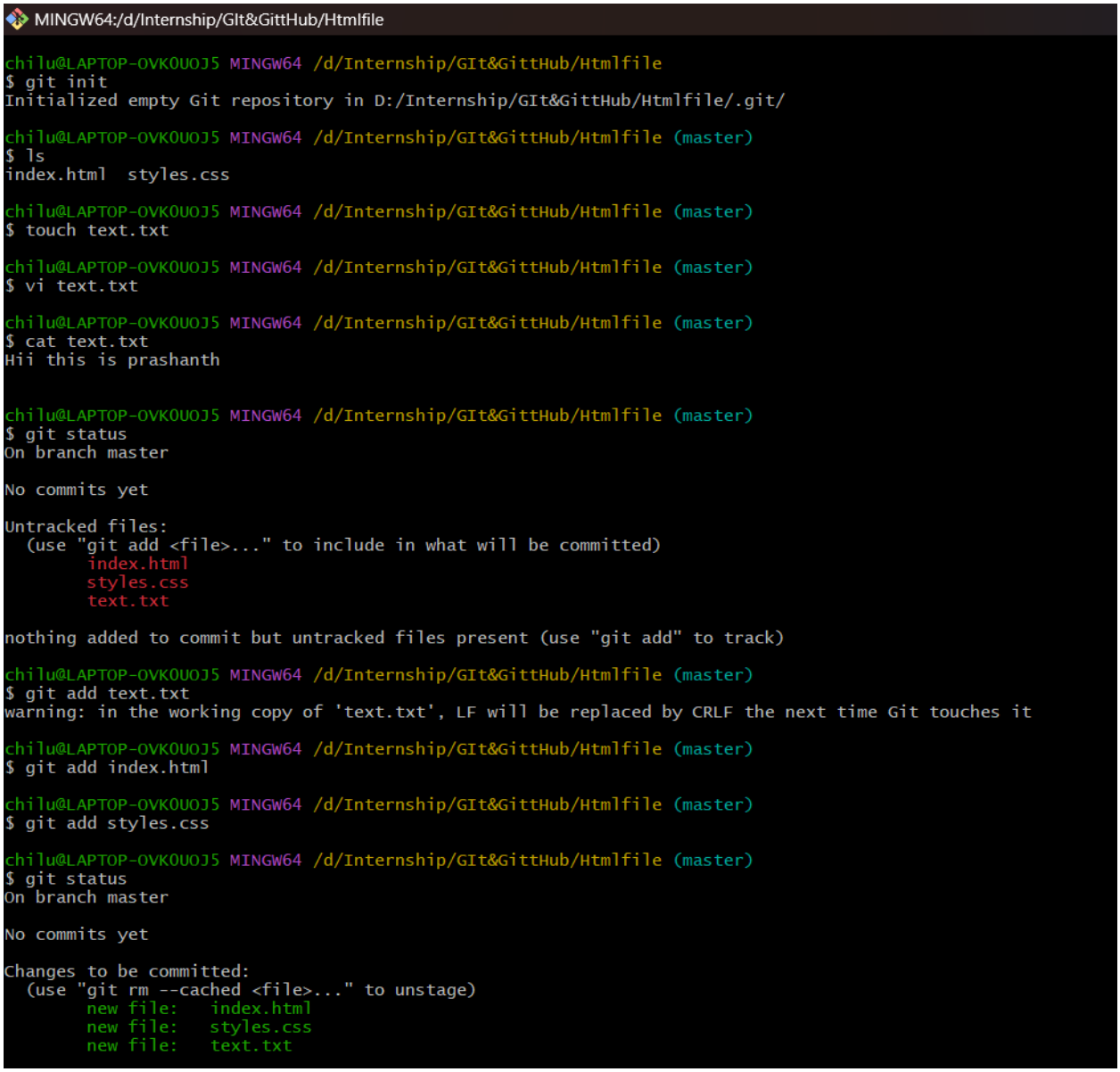
Collaboration is the way different people can work on the same project together.It is like creating a group in GitHub just like Groups in other social media.The people added to the collaborator's list can be able to push,merge,and do other kinds of similar things on the project.

Basic Git command:

- **Git init-**It creates a new Git repository & It creates a hidden directory called .git.
- git config --global user.name:it is used to set username.
- git config –global user.email: it is used to set email.



- **Ls-**listing the current directory contents and by default will not show hidden files.
- **Touch filename-** To create new files.
- **Vi filename-**To edit or modify.
- **Cat filename-**To Display the contents of one or more files without having to open the file for editing.
- **Git status-**It is used to display the state of the repository and staging area. It allows us to see the tracked, untracked files and changes.
- **git add filename:** it is used to add specific file to local repository.
- **Git add .-**It is used to add file contents to the Index (Staging Area).This command updates the current content of the working tree to the staging area.

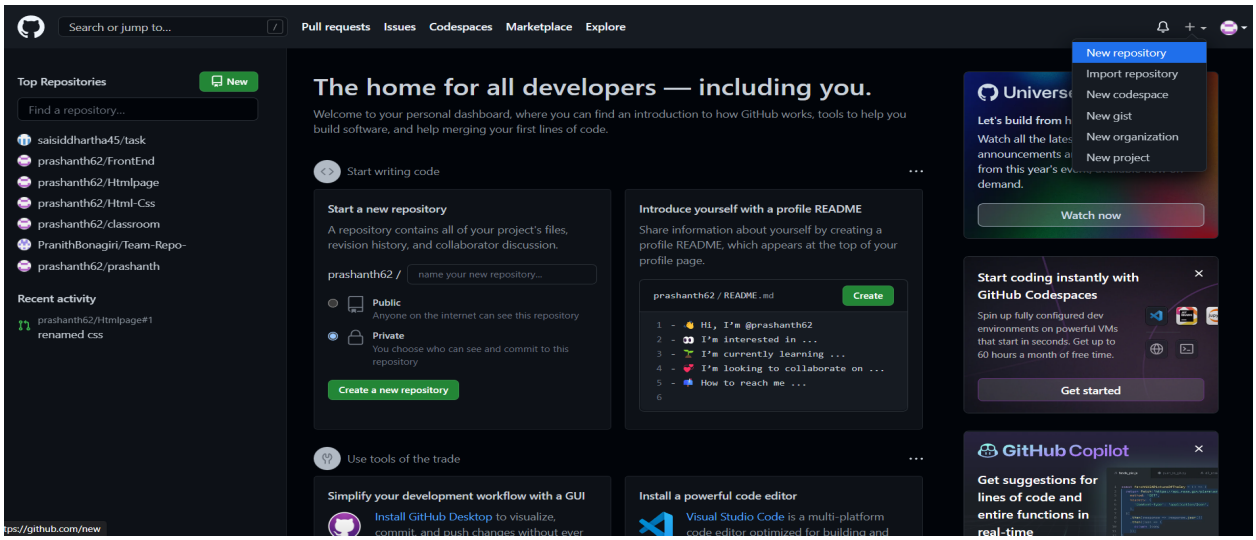


- **Git commit -m “commit message”-**It is used to record the changes in the repository. It is the next command after the git add.

What is GitHub?

- It is a code hosting platform for collaboration and version control.
- GitHub lets you(and others) work together on projects.

New Repository Creation:



- ☐ **Git Remote add origin 'url'-**The git remote command allows accessing the connection between remote and local.
- ☐ **Git push -u origin branch name-** Pushing is an act of transfer commits from your local repository to a remote repository.

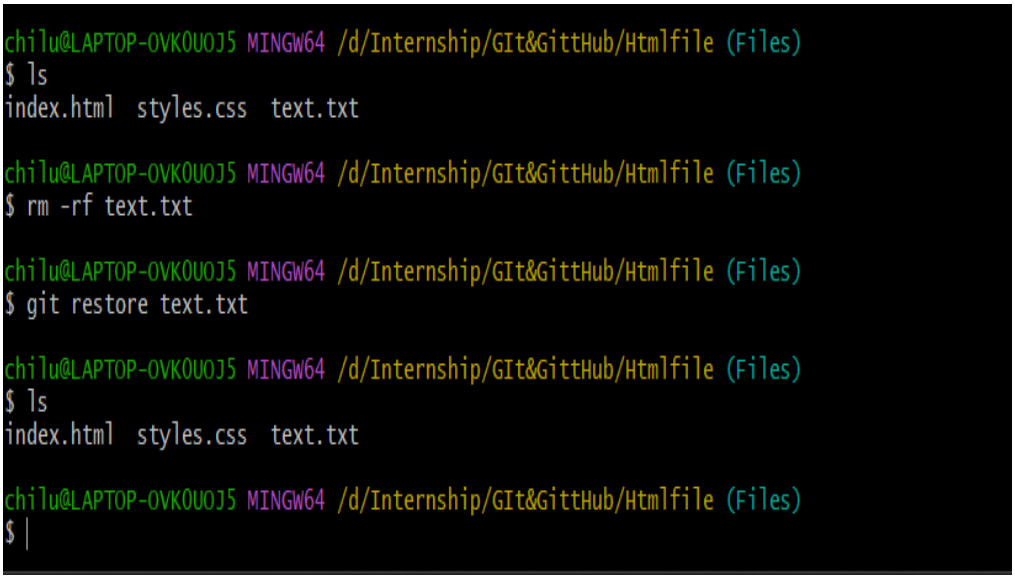
```
chilu@LAPTOP-OVKOUOJ5 MINGW64 /d/Internship/GIt&GittHub/Htmlfile (master)
$ git commit -m "Html & Css files added"
[master (root-commit) f8b3889] Html & Css files added
 3 files changed, 118 insertions(+)
 create mode 100644 index.html
 create mode 100644 styles.css
 create mode 100644 text.txt

chilu@LAPTOP-OVKOUOJ5 MINGW64 /d/Internship/GIt&GittHub/Htmlfile (master)
$ git remote add origin https://github.com/prashanth62/Html-Css.git

chilu@LAPTOP-OVKOUOJ5 MINGW64 /d/Internship/GIt&GittHub/Htmlfile (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 1.14 KiB | 389.00 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/prashanth62/Html-Css/pull/new/master
remote:
To https://github.com/prashanth62/Html-Css.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

- **Git log-**To show the commit logs
- **Git log - -oneline-**It is used to get log in oneline.
- git commit - - amend “commit”: it is used to modify commit.
- git reset “commit id”: it is used to remove commits
- git branch branchname: it is used to create a branch.
- git checkout -b branch: it is used to create and checkout into a branch

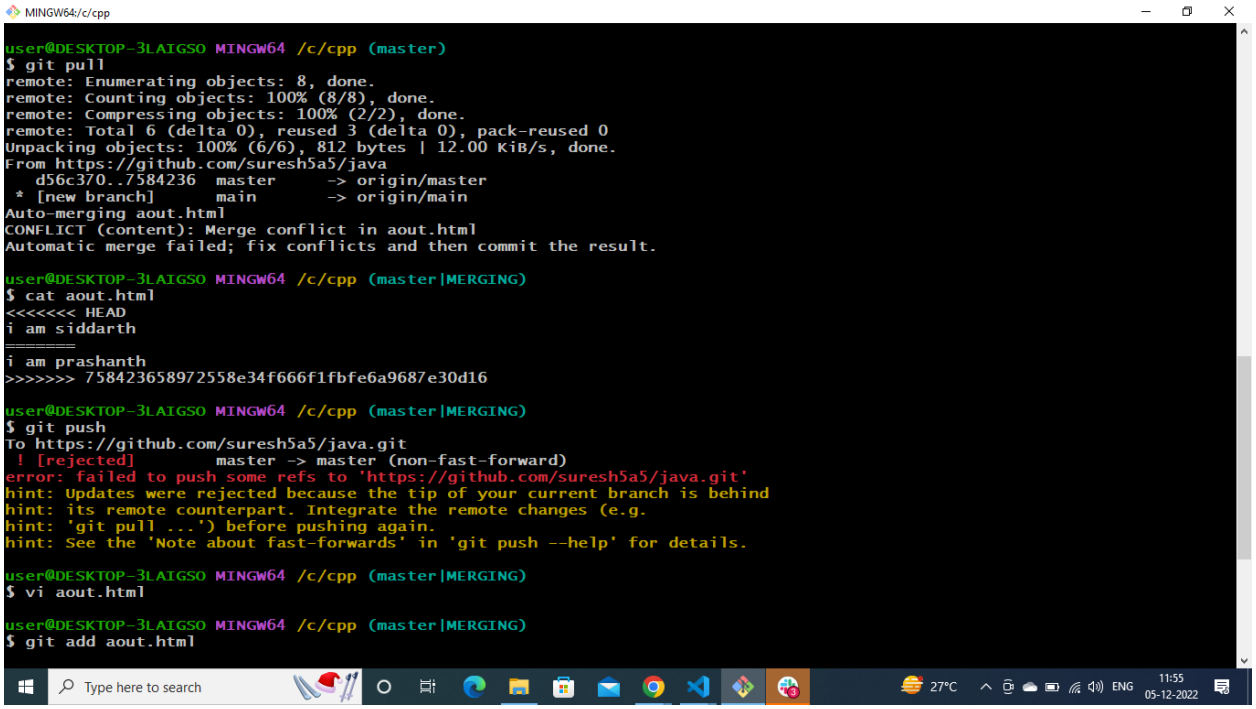
- git branch -d branch: it is used to delete a branch.
- **Git checkout branch-**switch to the given branch this will change the files in your project to the state of the branch that you are checking out.

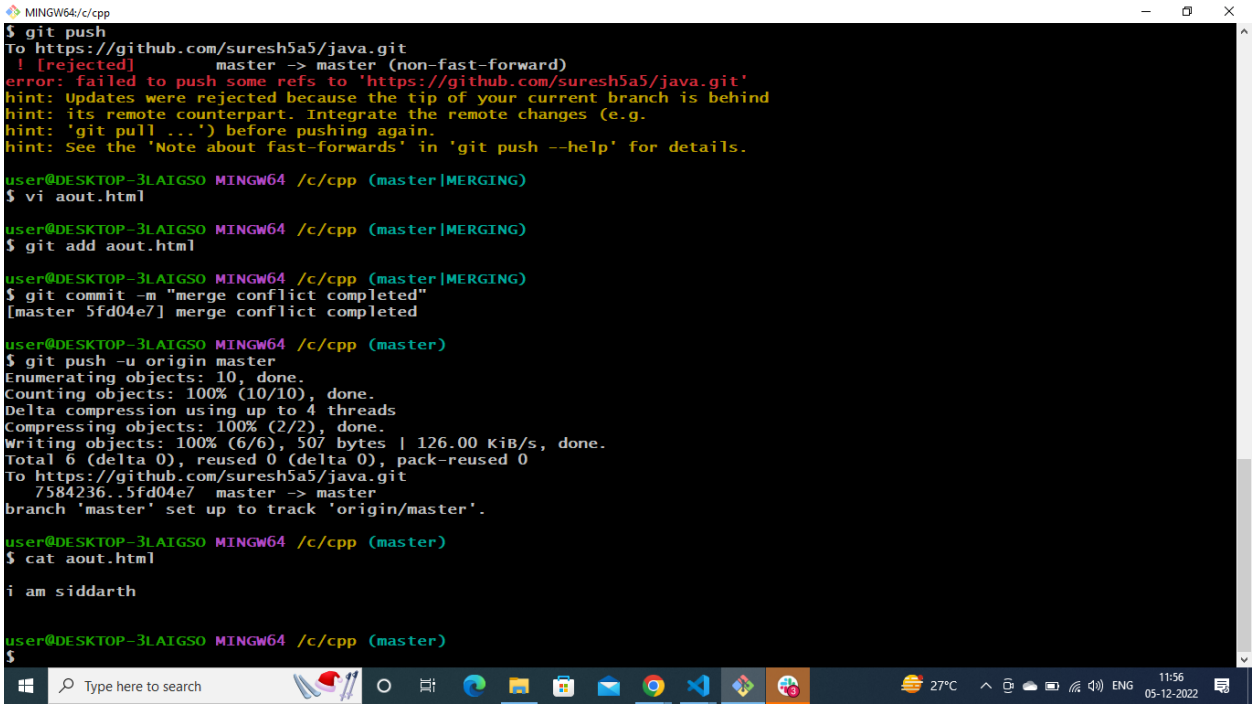
```
chilu@LAPTOP-OVKOUOJ5 MINGW64 /d/Internship/GIt&GittHub/Htmlfile (master)
$ git log
commit f8b388988b1c284f5b06b9d92fbb86809807ab93 (HEAD -> master, origin/master)
Author: prashanth62 <chilumulaprashanth62@gmail.com>
Date:   Mon Nov 14 11:21:34 2022 +0530

    Html & Css files added

chilu@LAPTOP-OVKOUOJ5 MINGW64 /d/Internship/GIt&GittHub/Htmlfile (master)
$ git branch Files

chilu@LAPTOP-OVKOUOJ5 MINGW64 /d/Internship/GIt&GittHub/Htmlfile (master)
$ git checkout Files
Switched to branch 'Files'

chilu@LAPTOP-OVKOUOJ5 MINGW64 /d/Internship/GIt&GittHub/Htmlfile (Files)
$ git commit -m "files"
On branch Files
nothing to commit, working tree clean
```

- **Git ignore-**Sometimes we don't want to send the files to Git service like GitHub. We can specify files in Git to ignore.
- **Git pull url-**pull is used to receive data from GitHub. It fetches and merges changes from the remote server to your working directory.

- **Git merge branch-**To merge the two branches.
- **Git stash-** To send a file backstage.
- **Git stash pop-**The back stage files will be back to there original place the folder.
- **Rm -rf filename-**To delete the file.(where the rm-Remove, rf-recursively and forcefully.)
- **Git restore-**It is used to restore the deleting file.
- git clone url: it is used to clone a repository into our local machines.
- 

Git Merge Conflict:




MINGW64:/c/cpp
$ git push
To https://github.com/suresh5a5/java.git
 ! [rejected]   master -> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/suresh5a5/java.git'
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
user@DESKTOP-3LAIGSO MINGW64 /c/cpp (master|MERGING)
$ vi aout.html
user@DESKTOP-3LAIGSO MINGW64 /c/cpp (master|MERGING)
$ git add aout.html
user@DESKTOP-3LAIGSO MINGW64 /c/cpp (master|MERGING)
$ git commit -m "merge conflict completed"
[master 5fd04e7] merge conflict completed
user@DESKTOP-3LAIGSO MINGW64 /c/cpp (master)
$ git push -u origin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 507 bytes | 126.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/suresh5a5/java.git
   7584236..5fd04e7 master -> master
branch 'master' set up to track 'origin/master'.
user@DESKTOP-3LAIGSO MINGW64 /c/cpp (master)
$ cat aout.html
i am siddarth
user@DESKTOP-3LAIGSO MINGW64 /c/cpp (master)
$
Type here to search
27°C
ENG
11:56
05-12-2022