

Case Study Git

- Local Operations :-

1] Create a new local git repository on to your machine and configure it for user.name and user.email attribute.

```
ASUS@LAPTOP-D9036EAH MINGW64 ~ (master)
$ git init
Reinitialized existing Git repository in C:/Users/ASUS/.git/

ASUS@LAPTOP-D9036EAH MINGW64 ~ (master)
$ git config --global user.name Bishal

ASUS@LAPTOP-D9036EAH MINGW64 ~ (master)
$ git config --global user.email vishalh016@gmail.com

ASUS@LAPTOP-D9036EAH MINGW64 ~ (master)
$ git config --global user.name
Bishal

ASUS@LAPTOP-D9036EAH MINGW64 ~ (master)
$ git config --global user.email
vishalh016@gmail.com
```

2] Create a files in directory (File1, IgnoreFile and FileToDelete provided), say directory contains – File1.txt, IgnoreFile.txt and FileToDelete.txt, now start tracking only the File1.txt and FileToDelete.txt, and ignore the IgnoreFile.txt by using .gitignore.

```
C:\Users\ASUS\Documents>cd Git06

C:\Users\ASUS\Documents\Git06>touch File1.txt

C:\Users\ASUS\Documents\Git06>touch IgnoreFile1.txt FileToDelete.txt

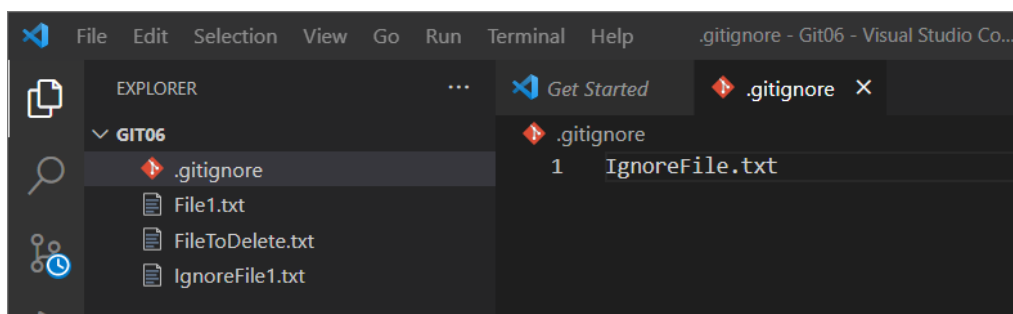
C:\Users\ASUS\Documents\Git06>git init
Initialized empty Git repository in C:/Users/ASUS/Documents/Git06/.git/

C:\Users\ASUS\Documents\Git06>git add File1.txt

C:\Users\ASUS\Documents\Git06>git add FileToDelete.txt

C:\Users\ASUS\Documents\Git06>touch .gitignore
```

Now, we've to place/write the "IgnoreFile.txt" inside the .gitignore file



3] Commit these changes to repository.

Use commit message as “This is First Commit, Tracking File1 and FileToDelete”

Note: - only File1.txt and FileToDelete.txt should be tracked and not the IgnoreFile.txt

```
C:\Users\ASUS\Documents\Git06>git status
On branch master
nothing to commit, working tree clean
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   File1.txt
        new file:   FileToDelete.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        IgnoreFile1.txt

C:\Users\ASUS\Documents\Git06>git commit -m "This is First Commit, Tracking File1 and FileToDelete"
[master (root-commit) 8355919] This is First Commit, Tracking File1 and FileToDelete
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 File1.txt
create mode 100644 FileToDelete.txt
```

4] Make 2 more commits by making following changes in the File1.txt

Add changes to file as:-

Change1 – commit with message as “Change1 in File1”

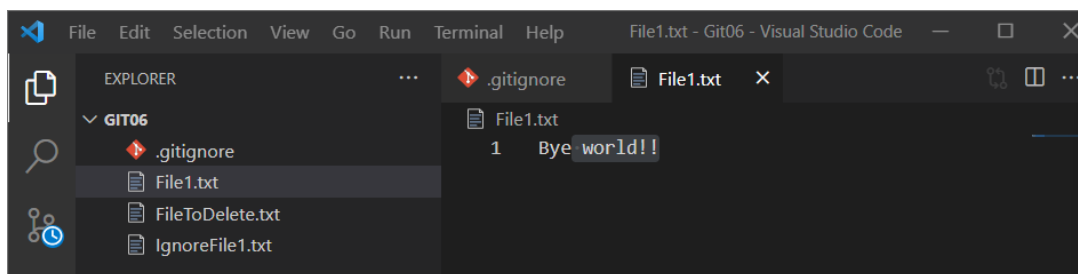
Change2 – commit with message as “Change2 in File1”

And if you check the log it should display in following manner:-Change2 in File1

Change1 in File1

This is First Commit, Tracking File1 and FileToDelete

Previously, I've created File1.txt and made my very first commit. Then I've added text (as shown below) to the file and commit the changed file.



```

C:\Users\ASUS\Documents\Git06>git commit -m "This is First Commit, Tracking File1 and FileToDelete"
[master (root-commit) 8355919] This is First Commit, Tracking File1 and FileToDelete
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 File1.txt
create mode 100644 FileToDelete.txt

C:\Users\ASUS\Documents\Git06>git add File1.txt

C:\Users\ASUS\Documents\Git06>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   File1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        IgnoreFile1.txt

C:\Users\ASUS\Documents\Git06>git commit -m "Change1 in File1"
[master 78c2e3d] Change1 in File1
1 file changed, 1 insertion(+)

C:\Users\ASUS\Documents\Git06>git commit -m "Change1 in File1"
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        IgnoreFile1.txt

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

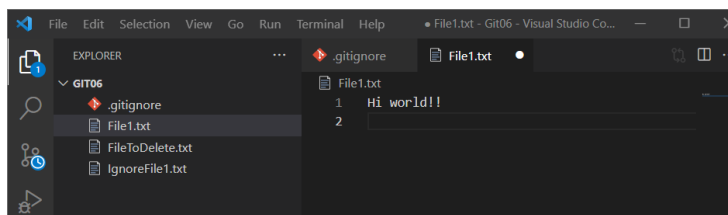
C:\Users\ASUS\Documents\Git06>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        IgnoreFile1.txt

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

C:\Users\ASUS\Documents\Git06>

```

Then I've changed the content of the File1.txt again and commit the newly changed files.



```

C:\Users\ASUS\Documents\Git06>git add File1.txt

C:\Users\ASUS\Documents\Git06>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   File1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        IgnoreFile1.txt

C:\Users\ASUS\Documents\Git06>git commit -m "Change2 in File1"
[master 7fd4e29] Change2 in File1
1 file changed, 1 insertion(+), 1 deletion(-)

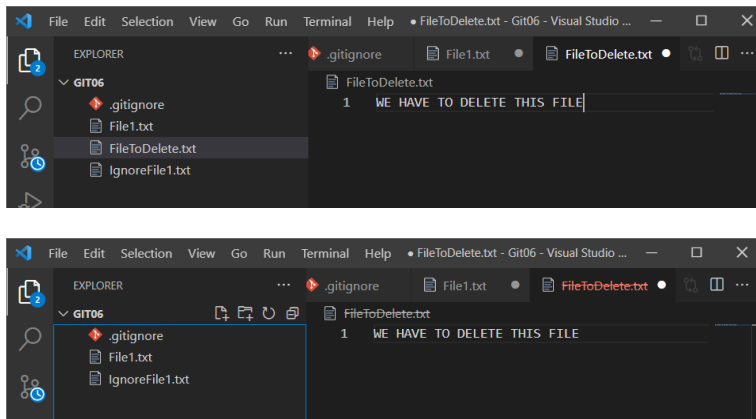
```

Commit log shows as follows:

```
C:\Users\ASUS\Documents\Git06>git log --oneline
7fd4e29 (HEAD -> master) Change2 in File1
78c2e3d Change1 in File1
8355919 This is First Commit, Tracking File1 and FileToDelete
```

5] Delete the FileToDelete.txt file from directory and commit with message as “FileToDelete is deleted”

We have deleted the FileToDelete.txt manually



Git command as follows:

```
C:\Users\ASUS\Documents\Git06>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:    FileToDelete.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .gitignore
    IgnoreFile1.txt

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

C:\Users\ASUS\Documents\Git06>git commit -am "FileToDelete is deleted"
[master 09d6170] FileToDelete is deleted
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 FileToDelete.txt
```

6] Display logs:-

Show all the commit logs.

Show last three logs in one line

Show only commits that occur between from date and to date

Show only those commit message that contains “File1” as in commits.

- All commit logs:

```
C:\Users\ASUS\Documents\Git06>git log
commit 09d6170ca10748adab08c71b8f60355eec4bb692 (HEAD -> master)
Author: Bishal <vishalh016@gmail.com>
Date: Tue Feb 1 13:28:42 2022 +0530

    FileToDelete is deleted

commit 7fd4e29081fd343721b68e814ec3b96e3b90e733
Author: Bishal <vishalh016@gmail.com>
Date: Tue Feb 1 11:16:28 2022 +0530

    Change2 in File1

commit 78c2e3ddd19c4eeaea019ce3388568c92cb5878d
Author: Bishal <vishalh016@gmail.com>
Date: Tue Feb 1 11:12:05 2022 +0530

    Change1 in File1

commit 8355919ea316482097240096e77bb1f5cef3ca66
Author: Bishal <vishalh016@gmail.com>
Date: Tue Feb 1 10:58:43 2022 +0530

    This is First Commit, Tracking File1 and FileToDelete

C:\Users\ASUS\Documents\Git06>git log --oneline
09d6170 (HEAD -> master) FileToDelete is deleted
7fd4e29 Change2 in File1
78c2e3d Change1 in File1
8355919 This is First Commit, Tracking File1 and FileToDelete
```

- Last 3 commit log:

```
C:\Users\ASUS\Documents\Git06>git log --oneline -3
09d6170 (HEAD -> master) FileToDelete is deleted
7fd4e29 Change2 in File1
78c2e3d Change1 in File1
```

- Commits between 30/01/2022 to 02/02/2022:

```
C:\Users\ASUS\Documents\Git06>git log --after="2022-01-30" --before="2022-02-2" --oneline
09d6170 (HEAD -> master) FileToDelete is deleted
7fd4e29 Change2 in File1
78c2e3d Change1 in File1
8355919 This is First Commit, Tracking File1 and FileToDelete
```

- All Commits containing "File1" as commit:

```
C:\Users\ASUS\Documents\Git06>git log --grep="File1"
commit 7fd4e29081fd343721b68e814ec3b96e3b90e733
Author: Bishal <vishalh016@gmail.com>
Date: Tue Feb 1 11:16:28 2022 +0530

    Change2 in File1

commit 78c2e3ddd19c4eeaea019ce3388568c92cb5878d
Author: Bishal <vishalh016@gmail.com>
Date: Tue Feb 1 11:12:05 2022 +0530

    Change1 in File1

commit 8355919ea316482097240096e77bb1f5cef3ca66
Author: Bishal <vishalh016@gmail.com>
Date: Tue Feb 1 10:58:43 2022 +0530

    This is First Commit, Tracking File1 and FileToDelete
```

7] Permanently undo a “FileToDelete is deleted” commit snapshot.

```
C:\Users\ASUS\Documents\Git06>git log --oneline
09d6170 (HEAD -> master) FileToDelete is deleted
7fd4e29 Change2 in File1
78c2e3d Change1 in File1
8355919 This is First Commit, Tracking File1 and FileToDelete

C:\Users\ASUS\Documents\Git06> git reset --hard 09d6170
HEAD is now at 09d6170 FileToDelete is deleted
```

8] Rename File1.txt to TestFile.txt

```
C:\Users\ASUS\Documents\Git06>git mv File1.txt TestFile.txt
fatal: bad source, source=File1.txt, destination=TestFile.txt
```

- Branching and Rebasing

1] Create a branch 'bug', checkout to bug branch to solve the bug in file1.

Bug is – Put a “ ”(space) between numerical and letters

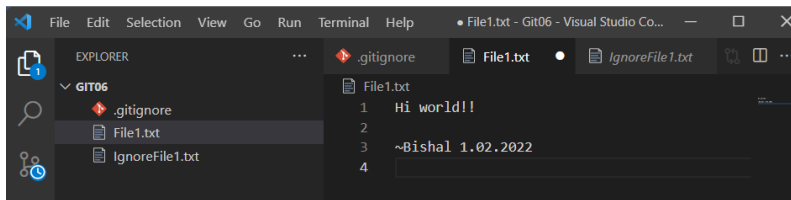
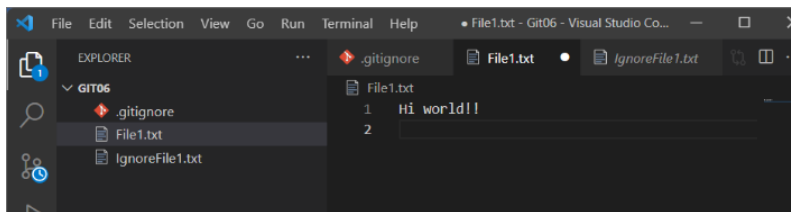
e.g.- Change1 --> Change 1

```
C:\Users\ASUS\Documents\Git06>git branch -a
* master

C:\Users\ASUS\Documents\Git06>git checkout -b bug
Switched to a new branch 'bug'

C:\Users\ASUS\Documents\Git06>git branch -a
* bug
  master
```

I've changed the contents of file1.txt (adding a new line and a blank space between numerical and letters) as shown below.



2] Commit this change with message as “Bug Resolved – Space added between numerical and letters”

```
C:\Users\ASUS\Documents\Git06>git branch -a
* bug
  master

C:\Users\ASUS\Documents\Git06>git add File1.txt

C:\Users\ASUS\Documents\Git06>git status
On branch bug
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   File1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        IgnoreFile1.txt

C:\Users\ASUS\Documents\Git06>git commit -m "Bug Resolved - Space added between numerical and letters"
[bug b654fad] Bug Resolved - Space added between numerical and letters
1 file changed, 3 insertions(+), 1 deletion(-)
```

3] Merge the changes to master assuming bug is resolved.

```
C:\Users\ASUS\Documents\Git06>git checkout master
Switched to branch 'master'

C:\Users\ASUS\Documents\Git06>git merge bug
Updating 09d6170..b654fad
Fast-forward
 File1.txt | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

4] Delete branch 'bug'

```
C:\Users\ASUS\Documents\Git06>git branch -a
bug
* master

C:\Users\ASUS\Documents\Git06>git branch -d bug
Deleted branch bug (was b654fad).
```

5] Recover the deleted branch 'bug' and rename it to 'bug123'

```
C:\Users\ASUS\Documents\Git07>git branch -a
* master

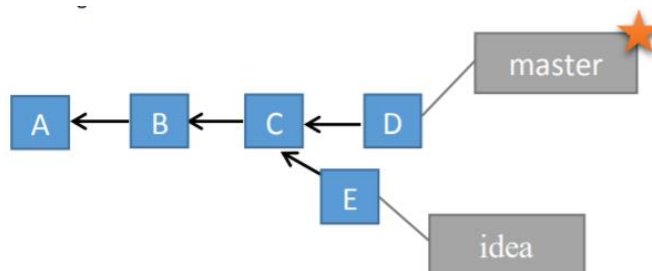
C:\Users\ASUS\Documents\Git07>git checkout -b bug ce8e2bd
Switched to a new branch 'bug'

C:\Users\ASUS\Documents\Git07>git branch -m bug bug123

C:\Users\ASUS\Documents\Git07>git status
On branch bug123
nothing to commit, working tree clean

C:\Users\ASUS\Documents\Git07>git branch -a
* bug123
  master
```


- 6] Consider the following scenario and achieve the same to Rebase:-
Create new branch 'idea' and make a new commit on 'idea' branch
Now checkout to master and make a new commit on master
Refer the below image:-



```
C:\Users\ASUS\Documents\Git07>git branch -a
* bug123
  master

C:\Users\ASUS\Documents\Git07>git checkout -b idea
Switched to a new branch 'idea'

C:\Users\ASUS\Documents\Git07>git commit -m "Commit: Idea"
On branch idea
nothing to commit, working tree clean

C:\Users\ASUS\Documents\Git07>git checkout master
Switched to branch 'master'

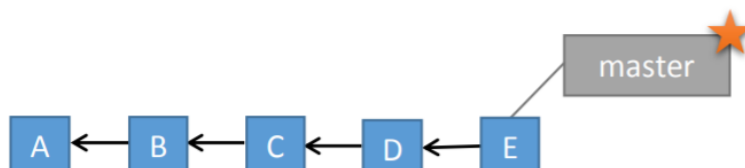
C:\Users\ASUS\Documents\Git07>git checkout master
Already on 'master'

C:\Users\ASUS\Documents\Git07>git branch -a
  bug123
  idea
* master

C:\Users\ASUS\Documents\Git07>git commit -am "New commit || to idea"
error: switch `m' requires a value

C:\Users\ASUS\Documents\Git07>git commit -am "New commit || to idea"
On branch master
nothing to commit, working tree clean
```

Now use Rebase to get the linear story line as follows:-



```
C:\Users\ASUS\Documents\Git07>git rebase master
Current branch idea is up to date.
```

8] Now say bug is resolved and you are good to release this stable version, tag this release as 'v1.0' with message as "Stable version 1.0 released".

```
C:\Users\ASUS\Documents\Git07> git tag 'v1.0' -m "Stable version 1.0 released."
C:\Users\ASUS\Documents\Git07>git push origin1 master
Everything up-to-date
```

9] Consider the following scenario to work with Stashing:-

Checkout to 'bug123' branch first and add text as "Change 3" in File1.txt, stage this change and don't commit the same so that you can get a dirty state of working directory.

Now assume that you have a priority task to be made on master branch, checkout to master and store this dirty state (on 'bug123') to stash so that you get back to this state later.

```
C:\Users\ASUS\Documents\Git07>git add .
C:\Users\ASUS\Documents\Git07>git stash
Saved working directory and index state WIP on master: 3eac6de Change4 in RemoteFile
```

```
C:\Users\ASUS\Documents\Git07>git checkout master
```

10] Display the list of all stash.

```
C:\Users\ASUS\Documents\Git07>git stash list
stash@{0}: WIP on master: 3eac6de Change4 in RemoteFile
stash@{1}: WIP on idea: c12c4bc Change2 in remote
```

11] Checkout to 'bug123' branch, reapply the stash which is stored recently and finally commit this change with the message as "Change 3 in File1"

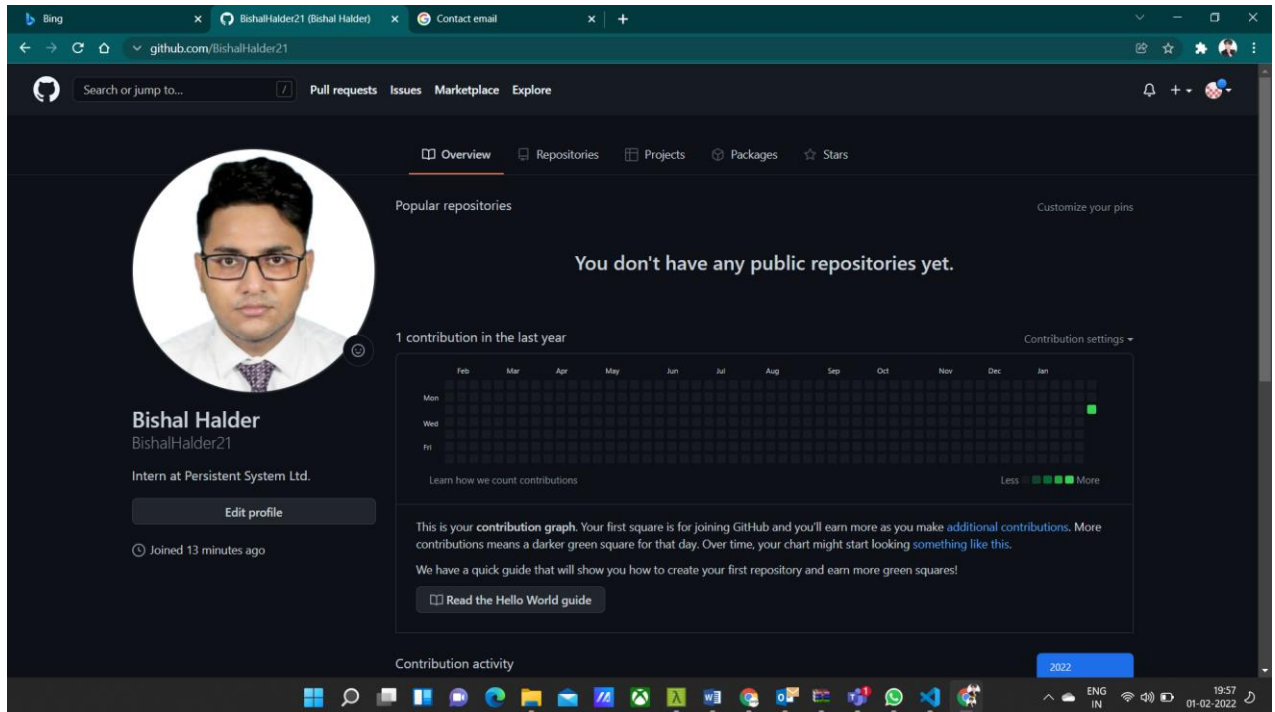
```
C:\Users\ASUS\Documents\Git07>git stash apply 0
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   File1.txt

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

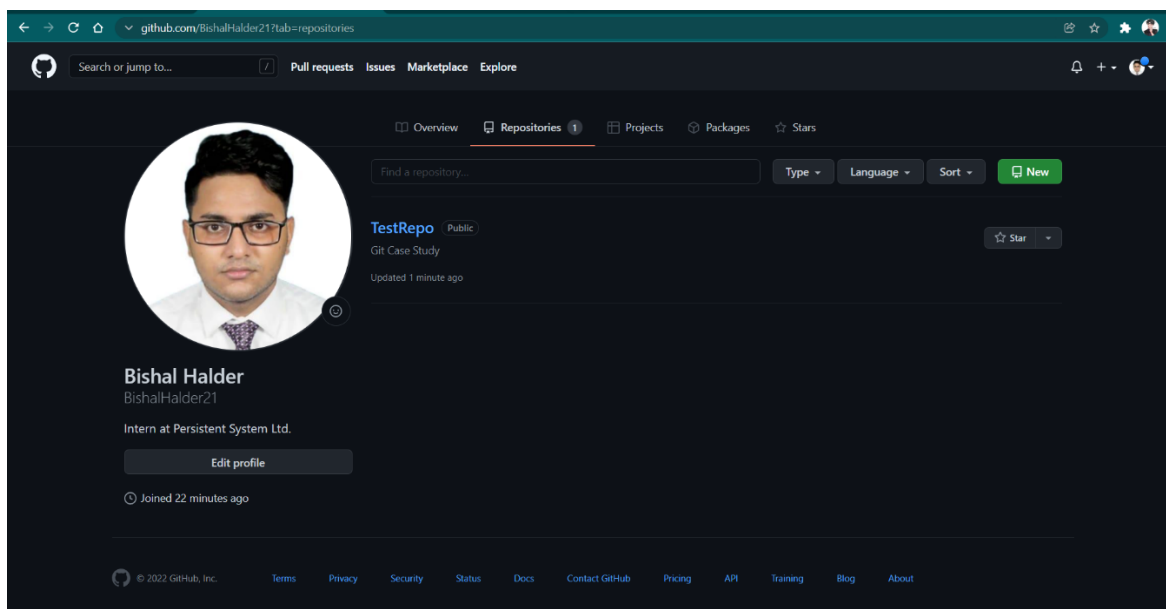
C:\Users\ASUS\Documents\Git07>git commit -am "Change3 in File1"
[master fc61b2b] Change3 in File1
1 file changed, 4 insertions(+), 2 deletions(-)
```

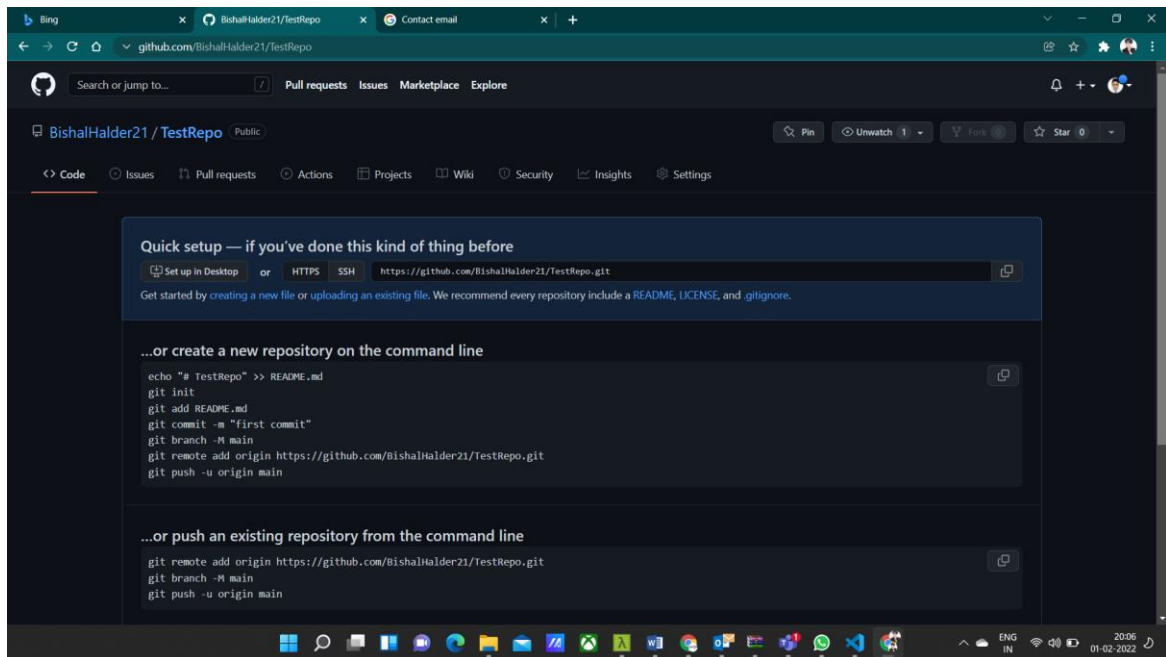
- **Git Remote:-**

1] Create (Signup) GitHub account by visiting <https://github.com/>



2] Create new repository as “TestRepo”.





3] Add remote on to your local repository and name it as 'Origin'.

```
C:\Users\ASUS\Documents\Git07>git remote add origin https://github.com/vishalh016/TestRepo.git
error: remote origin already exists.

C:\Users\ASUS\Documents\Git07>git remote add origin1 https://github.com/vishalh016/TestRepo.git
repository not found

C:\Users\ASUS\Documents\Git07>git push origin1 master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (6/6), 458 bytes | 458.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/vishalh016/TestRepo.git
* [new branch]      master -> master
```

4] Display all remote names with their details.

```
C:\Users\ASUS\Documents\Git07>git remote -v
origin  https://github.com/BishalHalder21/TestRepo.git (fetch)
origin  https://github.com/BishalHalder21/TestRepo.git (push)
origin1 https://github.com/vishalh016/TestRepo.git (fetch)
origin1 https://github.com/vishalh016/TestRepo.git (push)
```

5] Rename remote branch 'Origin' to 'RemoteBranch'.

```
C:\Users\ASUS\Documents\Git07>git remote rename origin RemoteBranch
```

6] Take a copy (Clone) of remote repository to your local machine.

```
C:\Users\ASUS\Documents\Git07>git clone https://github.com/vishalh016/TestRepo.git
Cloning into 'TestRepo'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
```

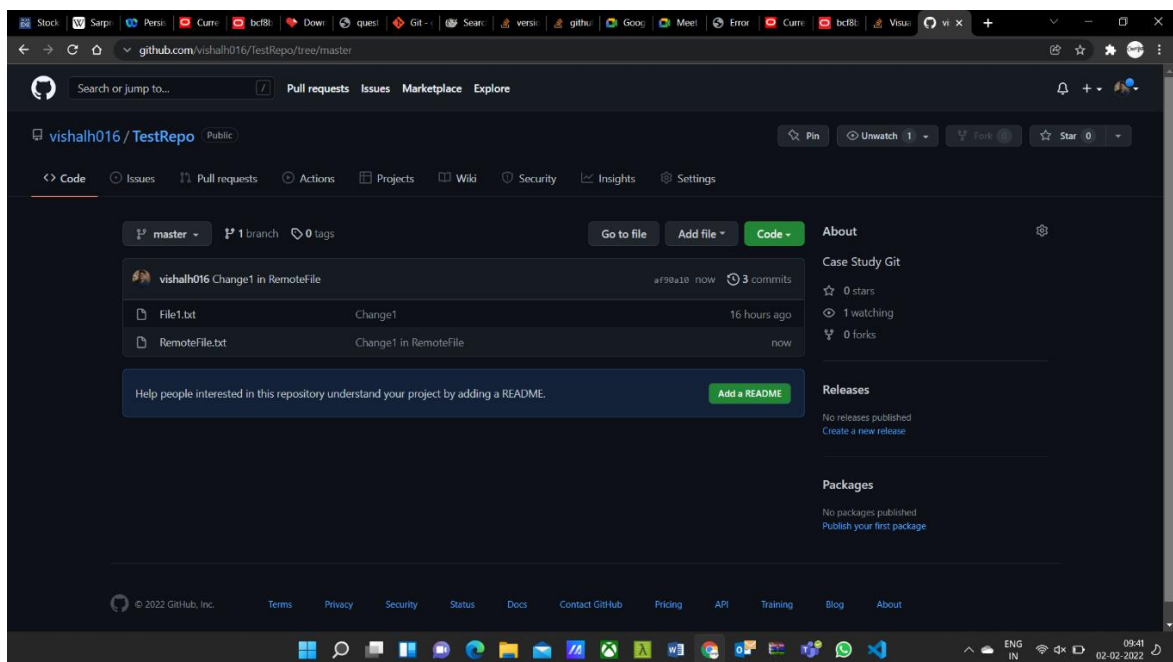
7] Now create a new file as 'RemoteFile.txt' on *remote* repository, make 1 commit by making following change in the RemoteFile.txt

Add changes to file as:-

Change1 – commit with message as “Change1 in RemoteFile”

And if you check the log on *remote* it should display in following manner:-
Change1 in RemoteFile

Note: - If you check log on local there will be no commits which are made on remote.



Log on local shows no new commits as this is made on remote Repo

```
C:\Users\ASUS\Documents\Git07>git log --oneline
ce8e2bd (HEAD -> idea, origin1/master, master, bug123) Change1
99ef0ab first commit
```

8]Fetch remote 'Origin' to synch up with remote.

```
C:\Users\ASUS\Documents\Git07>git fetch origin master
fatal: 'origin' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

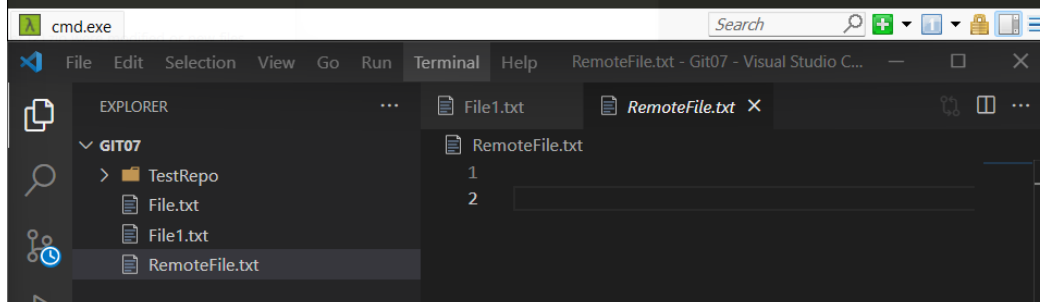
C:\Users\ASUS\Documents\Git07>git pull origin1 master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 665 bytes | 73.00 KiB/s, done.
From https://github.com/vishalh016/TestRepo
* branch                master      -> FETCH_HEAD
   ce8e2bd..af90a10      master      -> origin1/master
Updating ce8e2bd..af90a10
Fast-forward
 RemoteFile.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 RemoteFile.txt
```

9] Finally merge the remote changes to local master.

```
C:\Users\ASUS\Documents\Git07>git branch -a
bug123
* idea
master
remotes/origin1/master

C:\Users\ASUS\Documents\Git07>git branch merge remotes/origin1/master
Branch 'merge' set up to track remote branch 'master' from 'origin1'.

C:\Users\ASUS\Documents\Git07>
```

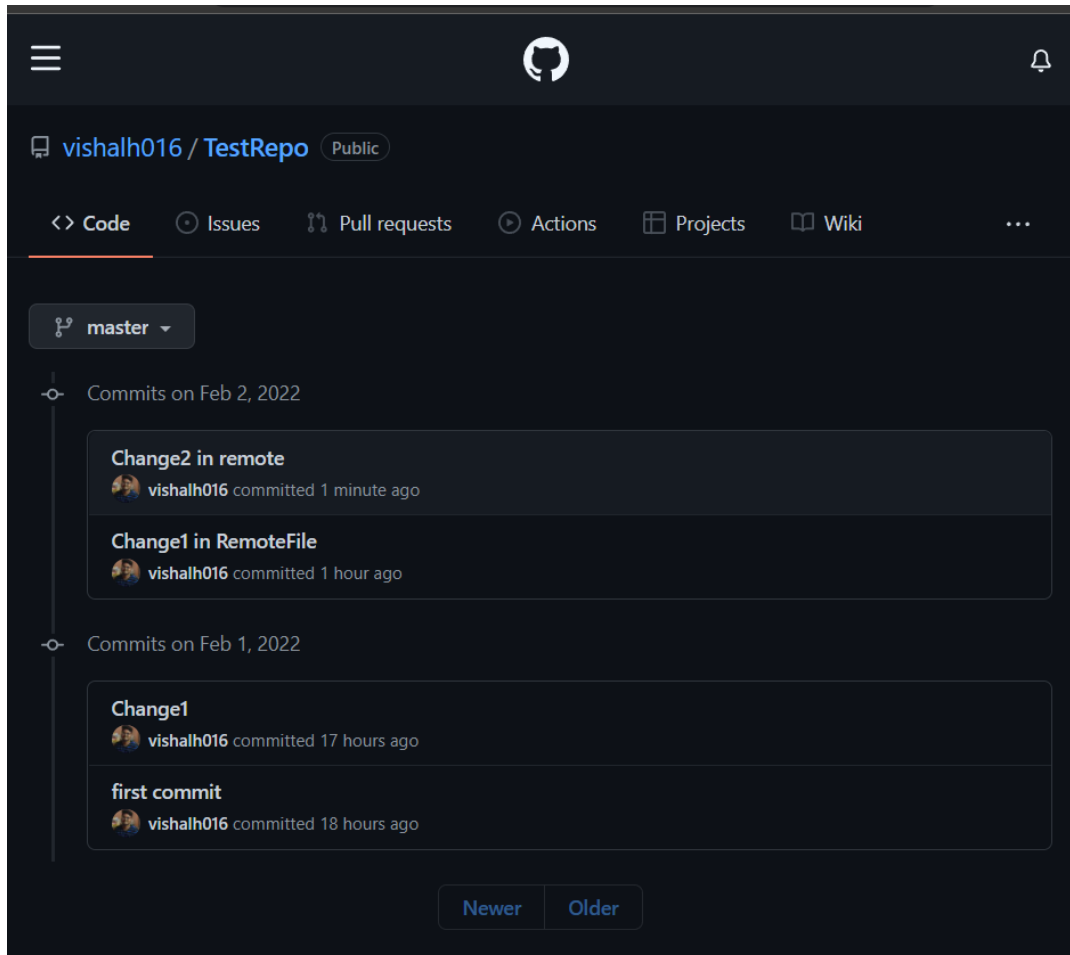


10] Now, make 1 more commit on *remote (GitHub)* by making following change in the RemoteFile.txt Add changes to file as:-

Change2 – commit with message as “Change2 in RemoteFile”

And if you check the log on *remote* it should display in following manner:-Change2 in RemoteFile

Change1 in RemoteFile



11] Use Pull to take these recent changes to local.

```
C:\Users\ASUS\Documents\Git07>git pull origin1 master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 674 bytes | 74.00 KiB/s, done.
From https://github.com/vishalh016/TestRepo
 * branch                master      -> FETCH_HEAD
    af90a10..c12c4bc      master      -> origin1/master
Updating af90a10..c12c4bc
Fast-forward
 RemoteFile.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```


12] Now, make 2 new commits on */local* by making following change in the RemoteFile.txt Add changes to file as:-

Change3 – commit with message as “Change3 in RemoteFile”

Change4 – commit with message as “Change4 in RemoteFile”

And if you check the log on */local* it should display in following manner:-

Change4 in RemoteFile

Change3 in RemoteFile

Change2 in RemoteFile

Change1 in RemoteFile

```
C:\Users\ASUS\Documents\Git07>git checkout master
Switched to branch 'master'

C:\Users\ASUS\Documents\Git07>git pull origin1 master
From https://github.com/vishalh016/TestRepo
* branch          master       -> FETCH_HEAD
Auto-merging RemoteFile.txt
CONFLICT (add/add): Merge conflict in RemoteFile.txt
Automatic merge failed; fix conflicts and then commit the result.

C:\Users\ASUS\Documents\Git07>git add .

C:\Users\ASUS\Documents\Git07>git commit -m "Change3 in RemoteFile"
[master d71ee89] Change3 in RemoteFile

C:\Users\ASUS\Documents\Git07>git push origin1 master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 437 bytes | 437.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/vishalh016/TestRepo.git
c12c4bc..d71ee89  master -> master

C:\Users\ASUS\Documents\Git07>git add .

C:\Users\ASUS\Documents\Git07>git pull origin1 master
From https://github.com/vishalh016/TestRepo
* branch          master       -> FETCH_HEAD
Already up to date.

C:\Users\ASUS\Documents\Git07>git add .

C:\Users\ASUS\Documents\Git07>git commit -m "Change4 in RemoteFile"
[master 3eac6de] Change4 in RemoteFile
1 file changed, 1 insertion(+)

C:\Users\ASUS\Documents\Git07>git push origin1 master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 337 bytes | 337.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/vishalh016/TestRepo.git
d71ee89..3eac6de  master -> master

C:\Users\ASUS\Documents\Git07>git pull origin1 master
From https://github.com/vishalh016/TestRepo
* branch          master       -> FETCH_HEAD
Already up to date.

C:\Users\ASUS\Documents\Git07>git log --oneline
3eac6de (HEAD -> master, origin1/master) Change4 in RemoteFile
d71ee89 Change3 in RemoteFile
c12c4bc (idea) Change2 in remote
af90a10 (merge) Change1 in RemoteFile
```

13] Push these recent changes to remote.

```
C:\Users\ASUS\Documents\Git07>git pull origin1 master
From https://github.com/vishalh016/TestRepo
* branch          master      -> FETCH_HEAD
Already up to date.

C:\Users\ASUS\Documents\Git07>git add .

C:\Users\ASUS\Documents\Git07>git commit -m "Change4 in RemoteFile"
[master 3eac6de] Change4 in RemoteFile
1 file changed, 1 insertion(+)

C:\Users\ASUS\Documents\Git07>git push origin1 master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 337 bytes | 337.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/vishalh016/TestRepo.git
d71ee89..3eac6de master -> master
```