Experiment No: 2

Performing Different operations on GIT

Aim: Understand and execute following commands

1.Create account on Github

2.creating repository locally

3.pull operation
4.add operation for one or multiple files

5.Commit operation for one or multiple files

6.Creation of Branch

7.Checkout

8.Merge and Rebase

9.push

Objectives:
1.Understand Push and pull operation
2.Understand commit concept.
3.Understand

Theory:

1) git config

Utility : To set your user name and email in the main configuration file.
How to : To check your name and email type in ***git config --global user.name*** and ***git config --global user.email***.

2) git init

Utility : To initialise a git repository for a new or existing project.
How to : ***git init*** in the root of your project directory.

3) git clone

Utility : To copy a git repository from remote source, also sets the remote to original source so that you can pull again.
How to : ***git clone <:clone git url:>***

4) git status

Utility : To check the status of files you've changed in your working directory, i.e, what all has changed since your last commit.
How to : ***git status*** in your working directory. lists out all the files that have been changed.

5) git add

Utility : adds changes to stage/index in your working directory.
How to : ***git add .***

6) git commit

Utility : commits your changes and sets it to new commit object for your remote.
How to : ***git commit -m"sweet little commit message"***

7) git push/git pull

Utility : Push or Pull your changes to remote. If you have added and committed your changes and you want to push them. Or if your remote has updated and you want those latest changes.
How to : ***git pull <:remote:> <:branch:>*** and ***git push <:remote:> <:branch:>***

8) git branch

Utility : Lists out all the branches.
How to : ***git branch*** or ***git branch -a*** to list all the remote branches as well.

9) git checkout

Utility : Switch to different branches
How to : ***git checkout <:branch:> or **_git checkout -b <:branch:>*** if you want to create and switch to a new branch.

10) git stash

Utility : Save changes that you don't want to commit immediately.
How to : ***git stash*** in your working directory. ***git stash*** apply if you want to bring your saved changes back.

11) git merge

Utility : Merge two branches you were working on.
How to : Switch to branch you want to merge everything in. git merge <***:branch_you_want_to_merge:***>

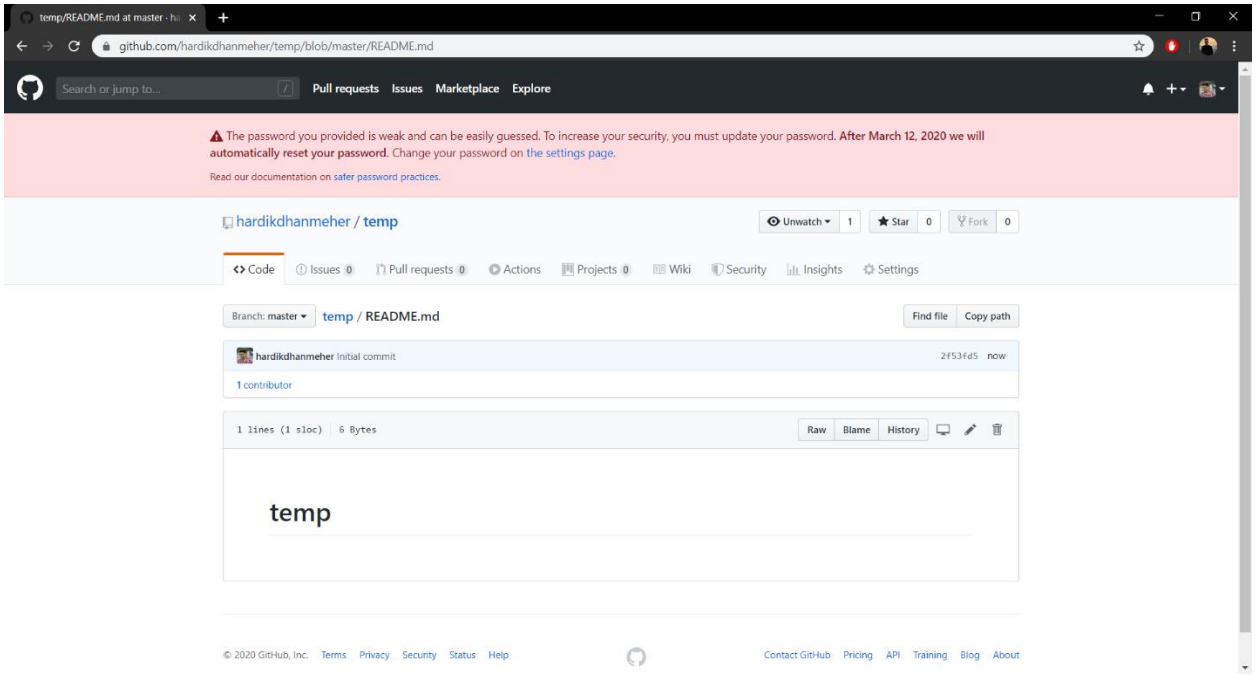
12) git reset

Utility : You know when you commit changes that are not complete, this sets your index to the latest commit that you want to work on with.
How to : ***git reset <:mode:> <:COMMIT:>***

13) git remote

Utility : To check what remote/source you have or add a new remote.
How to : ***git remote*** to check and list. And ***git remote add <:remote_url:>***

1. Create account on github




```
Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp
$ git --version
git version 2.20.1.windows.1

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp
$ git init
Initialized empty Git repository in C:/Users/Hardik/Desktop/temp/.git/


Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git remote add origin "https://github.com/hardikdhanmeher/temp.git"


Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git pull origin master
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/hardikdhanmeher/temp
 * branch            master     -> FETCH_HEAD
 * [new branch]      master     -> origin/master
```

```
Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git status
On branch master
nothing to commit, working tree clean

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git status
On branch master
Untracked files:
  (use "git add ..." to include in what will be committed)

        temp.txt

nothing added to commit but untracked files present (use "git add" to track)
Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git add temp.txt


Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD ..." to unstage)

        new file:   temp.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ ls
README.md  temp.txt
Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git commit -m "commiting first"
4[master cb9fc04] commiting first
 1 file changed, 1 insertion(+)
 create mode 100644 temp.txt
commit 2f53fd5f2674b34aa49ab30e63b64bfd7d7545b8 (origin/master)
Author: hardikdhanmeher
Date:   Tue Feb 18 14:58:12 2020 +0530

    Initial commit
```

‘Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git status
On branch master
nothing to commit, working tree clean

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git add -A

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD ..." to unstage)

        new file: temp2.txt
        new file: temp3.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git commit -a -m "commiting all files"
[master 572ca3f] commiting all files
 2 files changed, 2 insertions(+)
 create mode 100644 temp2.txt
 create mode 100644 temp3.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git status
On branch master
nothing to commit, working tree clean

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git log
commit 572ca3ffbdd6b0a14be8985c20461c078aa53c0b (HEAD -> master)
Author: altaf99
Date: Tue Feb 18 15:13:56 2020 +0530

    commiting all files

```
commit cb9fc045ec96ef5221a865d7ecf870c2f07b36b0
Author: altaf99
Date:   Tue Feb 18 15:10:51 2020 +0530

    commiting first

commit 2f53fd5f2674b34aa49ab30e63b64bfd7d7545b8 (origin/master)
Author: hardikdhanmeher
Date:   Tue Feb 18 14:58:12 2020 +0530

    Initial commit
```

Branching

```
Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git branch apple

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git checkout apple
Switched to branch 'apple'

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$
Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git branch apple

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git checkout apple
Switched to branch 'apple'

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ git add tempx.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ ls
README.md  temp.txt  temp2.txt  temp3.txt  tempx.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ git commit -m "doing changes in your branch"
[apple d9ad0e7] doing changes in your branch
```

```
 1 file changed, 1 insertion(+)
 create mode 100644 tempx.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ ls
README.md  temp.txt  temp2.txt  temp3.txt  tempx.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ git checkout master
Switched to branch 'master'

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ ls
README.md  temp.txt  temp2.txt  temp3.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git merge apple
Updating 572ca3f..d9ad0e7
Fast-forward
 tempx.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 tempx.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ ls
README.md  temp.txt  temp2.txt  temp3.txt  tempx.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git checkout apple
Switched to branch 'apple'
M       tempx.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ ls
README.md  temp.txt  temp2.txt  temp3.txt  tempx.txt
Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ git commit -a -m "commiting in tempx"
[apple ad184a5] commiting in tempx
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ cat tempx.txt
gkcbnjeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeenndkbjcb4jrbj4rb
Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ git checkout master
Switched to branch 'master'

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ cat tempx.txt
ndkbjcb4jrbj4rb
```

Rebasing

```
Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git checkout apple
Switched to branch 'apple'
Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ git add -A

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ ls
README.md  temp.txt  temp2.txt  temp3.txt  tempx.txt  text1.txt  text2.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ git commit -a -m "commitning"
[apple 2fd69f7] commitning
 2 files changed, 2 insertions(+)
 create mode 100644 text1.txt
 create mode 100644 text2.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ ls
README.md  temp.txt  temp2.txt  temp3.txt  tempx.txt  text1.txt  text2.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ git checkout master
Switched to branch 'master'


Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
```

```
$ ls
README.md  temp.txt  temp2.txt  temp3.txt  tempx.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git checkout apple
Switched to branch 'apple'

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ git rebase master
Current branch apple is up to date.

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ ls
README.md  temp.txt  temp2.txt  temp3.txt  tempx.txt  text1.txt  text2.txt

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (apple)
$ git checkout master
Switched to branch 'master'

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git rebase apple
First, rewinding head to replay your work on top of it...
Fast-forwarded master to apple.

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ ls
README.md  temp.txt  temp2.txt  temp3.txt  tempx.txt  text1.txt  text2.txt


Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Hardik/.ssh/id_rsa):
/c/Users/Hardik/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Hardik/.ssh/id_rsa.
Your public key has been saved in /c/Users/Hardik/.ssh/id_rsa.pub.
The key fingerprint is:
```

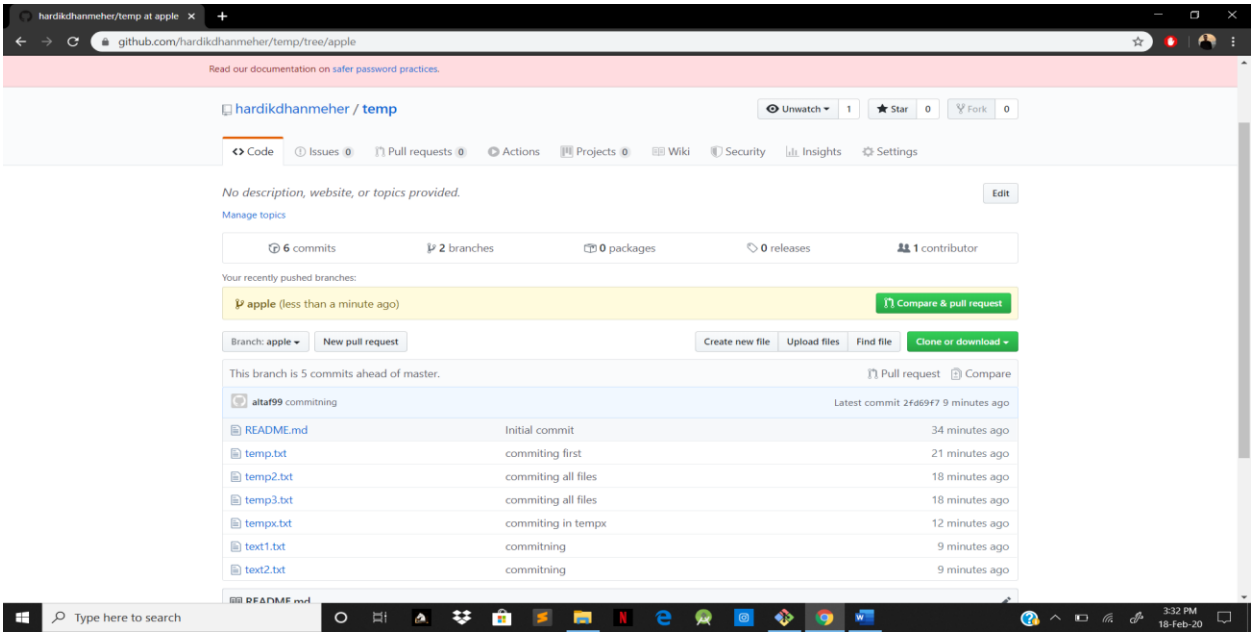
```
SHA256:pmIMcYWfaW7FWLASKzDufqjSZ23gZ5rER+zolU5uTOc
Hardik@DESKTOP-HR4KH9H
The key's randomart image is:
+---[RSA 2048]----+
|o   ..o.         |
|.o  .+ ..        |
| .o +..*         |
|.  + o* o        |
| ..  oo.S        |
|. .+.+++.        |
| + oB*Bo         |
|o o++OB E        |
|o  o+*o          |
+----[SHA256]-----+

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ ^C

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ cat /c/Users/Hardik/.ssh/id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQC2TWff2eBxIA+1n4dHTUH2cecApD
mklMEpJLTyW65N
NLkNpn9ZIUPWuwL0kmV4gKFM02QYvzhPvHkXw0SGo/DdLqxl6iilX71qj0EIBjD3
Ut08XOPt0MsszVRR
ww9ccmD+EZXK9uE9QMu1hswBHj9LsMs1GIJQwIvyg+jGxJkJKslrNtfhTvhjIoaS82I
EX0XKXYWNL7cK
zUo/8crsEJTGW5C79arncm5Gw6w90HlmC+1oGIVq75vyw/mjB1dRN0ww7JllAa+T7F
uuLVHCc/AsymoA
dDYHTi4gl61vIjshWEd+mfXcilWvChafjc122c9lZ6x4yzIsDIkhifrEBPpd
Hardik@DESKTOP-HR4K
H9H

Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ ssh -T git@github.com
Warning: Permanently added the RSA host key for IP address '13.234.210.38' to th
e list of known hosts.
Hi hardikdhanmeher! You've successfully authenticated, but GitHub does not provi
de shell access.
```

```
Hardik@DESKTOP-HR4KH9H MINGW64 ~/Desktop/temp (master)
$ git push origin apple
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 8 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (17/17), 1.24 KiB | 105.00 KiB/s, done.
Total 17 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), done.
remote:
remote: Create a pull request for 'apple' on GitHub by visiting:
remote:      https://github.com/hardikdhanmeher/temp/pull/new/apple
remote:
To https://github.com/hardikdhanmeher/temp.git
 * [new branch]      apple -> apple
```



Conclusion:

In this Experiment we had studied Different command that are used to perform operation on local Git and uploaded work on git repository