Git Tasks

1. Make a repository on GitHub with the name **“InnovationPython_yourname”**
eg: “InnovationPython_Ankush”.
https://github.com/Pnikumbh1114/InnovationPython_Priyanka

a. Practice on following commands:

■ Git Clone

```
Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/democa2020 (master)
$ git clone https://github.com/Pnikumbh1114/InnovationPython_Priyanka.git
Cloning into 'InnovationPython_Priyanka'...
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 12 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), 2.88 KiB | 3.00 KiB/s, done.
```

■ Git Diff
■ Git Add
■ Git Status
■ Git Log
■ Git Commit
■ Git Push
■ Git Rest
■ Git Pull
■ Git Branch
■ Git Checkout

2. Read about the difference between Git and GitHub

GitHub is a online platform where we can manage the Git repositories. It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project. They are commonly used to host open source projects.

Git is a version control system that lets us manage and keep track of the source code history. More specifically it is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers/developers, but it can be used to track changes in any set of files.

3. Read about Git Workflow

4. How many types of version control systems are there?

Version control systems has 3 types viz. local version control system, centralized version control system and distributed version control system.

Local version control system maintains track of files within the local system. This approach is very common and simple. This type is also prone to errors which means the chances of accidentally writing to the wrong file is higher.

In a centralized version system, there is a single central copy of the project on a server and all the changes are committed to that central copy. In this approach, all the changes in the files are tracked under the centralized server. The centralized server includes all the information of versioned files, and list of clients that check out files from that central place. The required files can be pulled from the server but there will never be a full copy or clone of the project on the local machine.
In a distributed version system, a clone can be made of the repository on a local machine which gives the full history of the project. Distributed version control systems came into picture to overcome the drawback of centralized version control system. The clients can completely clone the repository including its full history. If any server dies, any of the client repositories can be copied on to the server which help restore the server.

5. Explain Branching concept in Git.
6. Explain Forking Workflow in Git.

PRACTICAL TASK

● Initialize an empty Git repository on your local machine with the name **"gitlocal"** and make a **README.md** file in that directory which should contain your name as a heading and a hello message (https://www.makeareadme.com/) .

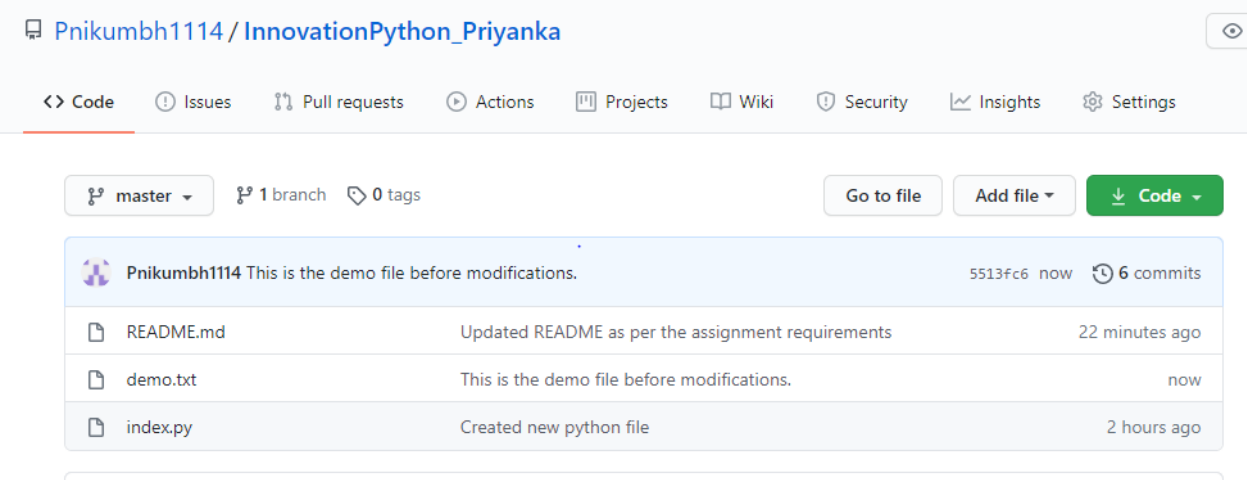
```
Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop (master)
$ git init gitlocal
Reinitialized existing Git repository in C:/Users/Nishant Nikumbh/Desktop/gitloc
al/.git/

Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop (master)
$ cd gitlocal

Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/gitlocal (master)
$ git add README.md
fatal: pathspec 'README.md' did not match any files

Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/gitlocal (master)
$ git >> README.md
```

● Now check the status of your git directory and push all the files in that directory to your GitHub repo which you have made in the first step. With a massage "First Commit to Git Repo".

```
Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/gitlocal (master)
$ git status
On branch master

No commits yet

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

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

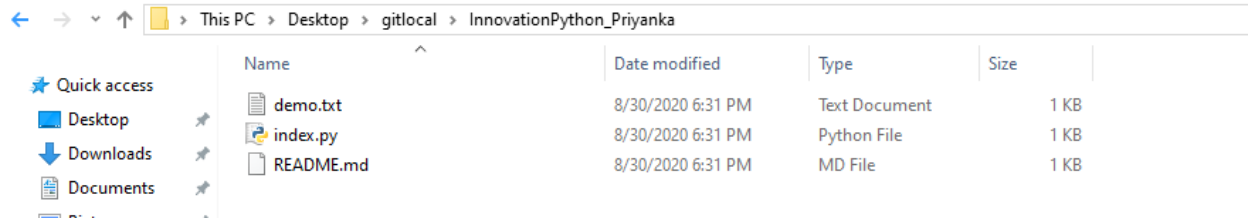
```
Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/gitlocal (master)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/gitlocal (master)
$ git status
On branch master

No commits yet

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


Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/gitlocal (master)
$ git commit README.md -m "First Commit to Git Repo"
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Nishant Nikumbh@LAPTOP-IQHOIMHM
.(none)')
```

● Now add a file to your Github repo named “demo.txt” from the github console with content : “This is the demo file before modifications”.



● Pull the changes in your git repo to your local machine git directory named **“gitlocal”** and check the status for the modifications done in that repo. This time demo should be visible in your local machine.

```
Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/gitlocal (master)
$ git pull InnovationPython_Priyanka master
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 18 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (18/18), 4.29 KiB | 3.00 KiB/s, done.
From InnovationPython_Priyanka
 * branch            master     -> FETCH_HEAD
error: Your local changes to the following files would be overwritten by merge:
        README.md
Please commit your changes or stash them before you merge.
Aborting
```



● Now make a new branch in your local machine with the name " **developer** ".

```
Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/gitlocal (master)
$ git checkout -b developer
Switched to a new branch 'developer'
```

● Edit that demo file and write some content in that eg: "This is a demo file after modification" and push the modifications to your GitHub repo from the **developer** branch with a commit message "Commit from developer branch".

```
Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/gitlocal (developer)
$ git >> demo.txt

Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/gitlocal (developer)
$ git add demo.txt
warning: LF will be replaced by CRLF in demo.txt.
The file will have its original line endings in your working directory

Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/gitlocal (developer)
$ git commit -m "Commit From Developer Branch"
[developer 0c2f0e3] Commit From Developer Branch
 1 file changed, 45 insertions(+)
 create mode 100644 demo.txt

Nishant Nikumbh@LAPTOP-IQHOIMHM MINGW64 ~/Desktop/gitlocal (developer)
$ git push InnovationPython_Priyanka developer
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.26 KiB | 429.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To InnovationPython_Priyanka
   688004e..0c2f0e3  developer -> developer
```