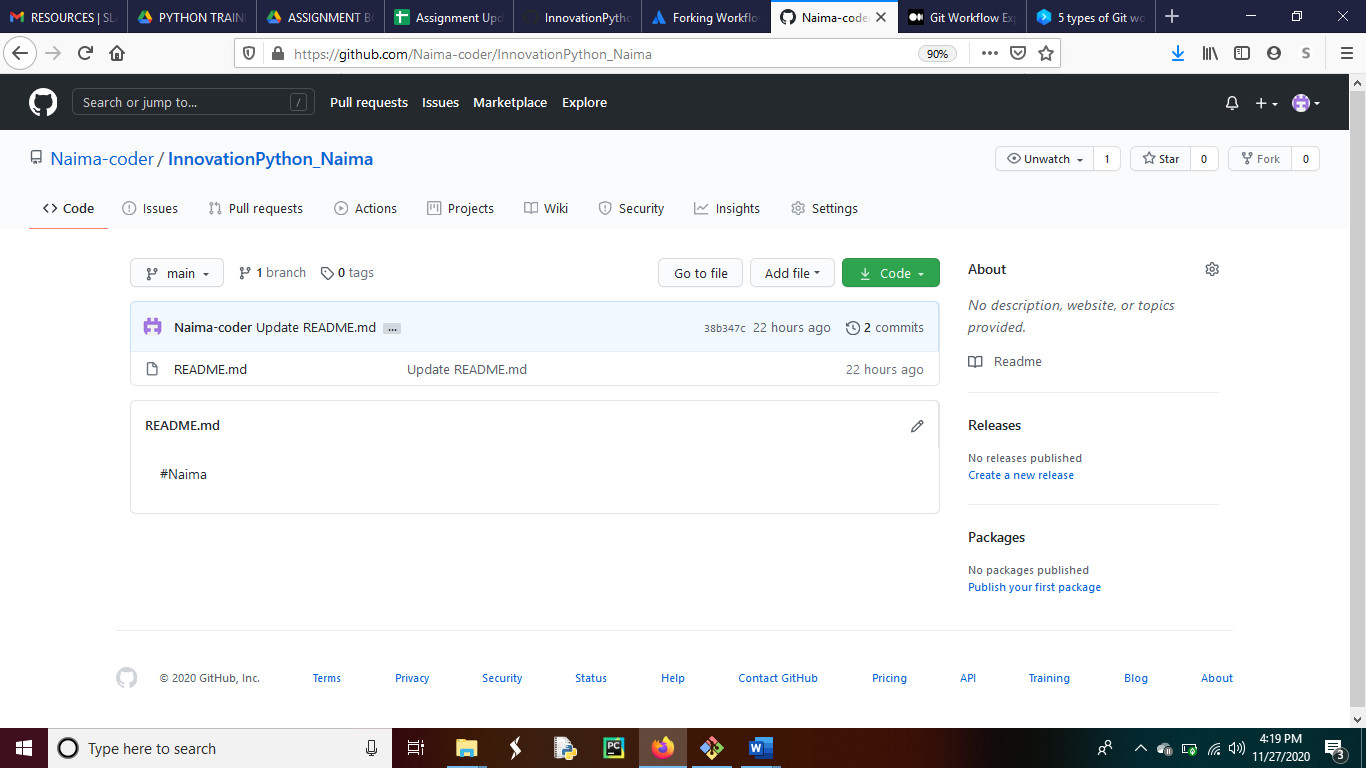
**1) Make a repository on GitHub with the name “InnovationPython\_yourname” eg:**

**“InnovationPython\_Ankush”.**



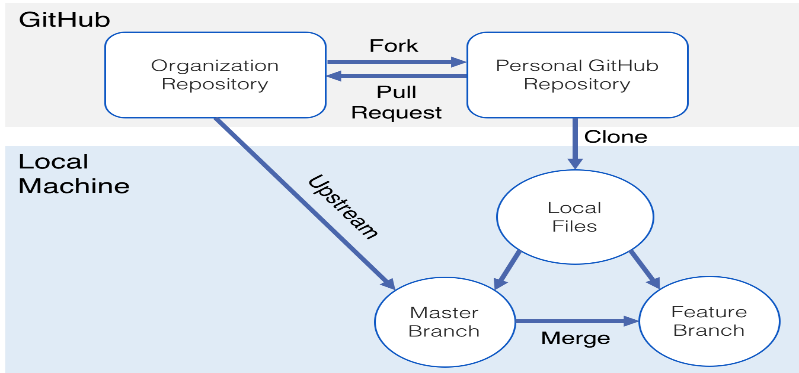
**2)Read about the difference between Git and GitHub**

Git is a distributed version control system. A version control system means something that help us to save each version of a project (any file) without over writing it. In this way we can retrieve any older version without explicitly making those changes in current version.

GitHub is a cloud-based hosting service that will manage our git repository.

**3) Read about Git Workflow**

A git workflow is the recommendation for how to use git to avoid possible issues that may happen when a group of team work on organization’s git repository.



Some simple steps or workflow that can be followed by a project team is given below.

1.Set up a GitHub organization.

2.Fork organization repo to your personal GitHub repo.

3.Clone your repo to your local machine.

4.Create a branch for your working files.

5.Set up a remote repo to GitHub organization repository.

6.Start coding in your local repo.

7.Pull organization repo and keep updated.

8.Merge master branch into feature branch.

9.Push your GitHub repo.

10.Make pull request to organization repo.

Examples of different workflows are:

1. Basic workflow.
2. Feature branch workflow.
3. Forking workflow.
4. Git workflow.

**4)How many types of version control systems are there?**

Based on its mode of operation, there are 3 type of version control systems.

1)Local version control system.

In local version control system patch sets or difference between each version is saved in a special format in version tracker which is stored in your computer hard disk. Hence if we add each relevant patch orderly then we can recreate any version that we want.

2)Centralized version control system.

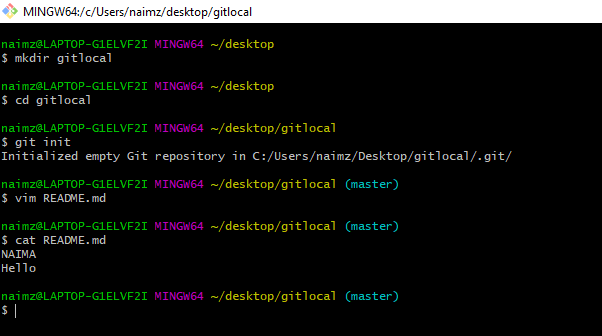
In centralized version control system file is saved in a server and is open to access for every client.

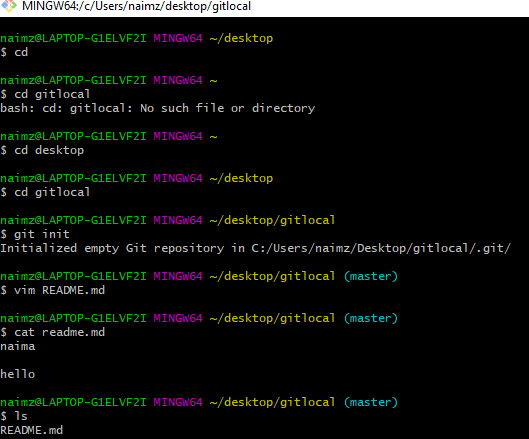
3)Distributed version control system.

A distributed version control system (DVCS) is a type of version control where the complete codebase including its full version history is mirrored on every developer's computer. Git is an example of distributed version control system.

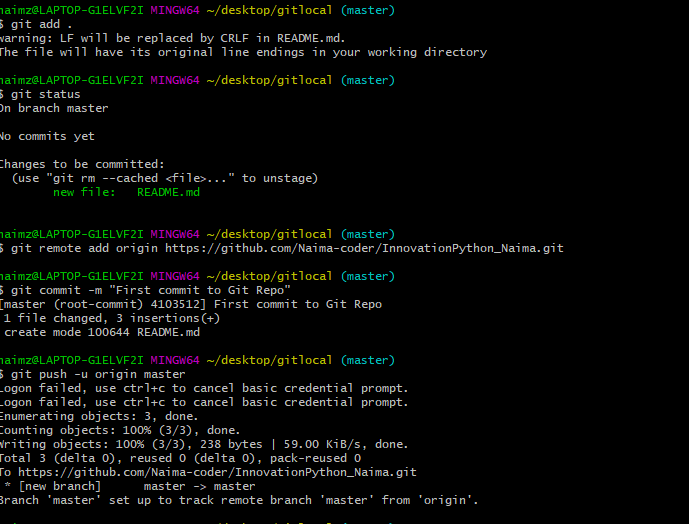
**Practical Task**

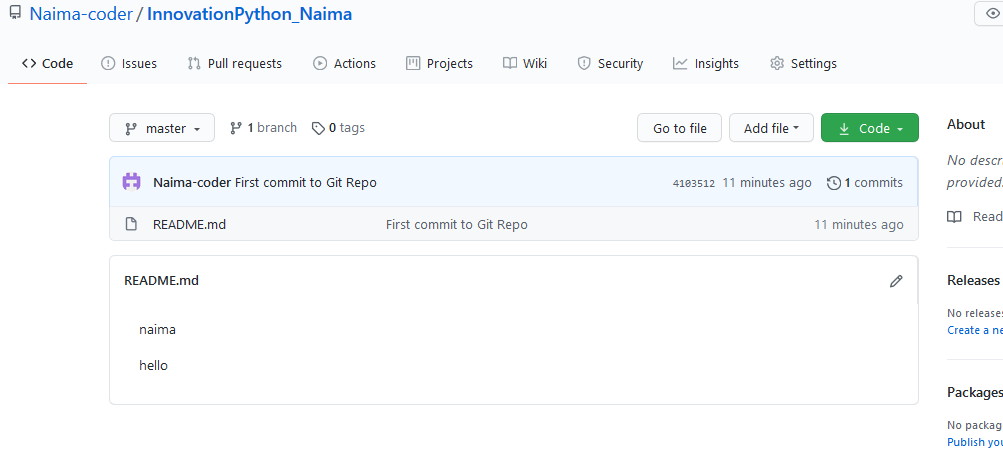
1. Initialize an empty Git repository on your local machine with the name “gitlocal” and make a README.md file in that directory which should contain your name as a heading and a hello message (https://www.makeareadme.com/).





1. Now check the status of your git directory and push all the files from that directory to your GitHub repo which you have made in the first step (“InnovationPython\_yourname”). With a massage “First Commit to Git Repo”.

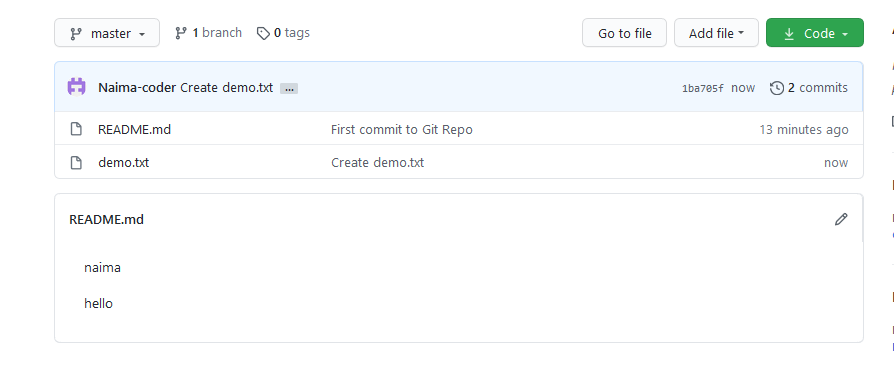




Github view after pushing README.md file

3)Now add a file to your Github repo named “demo.txt” from the github console

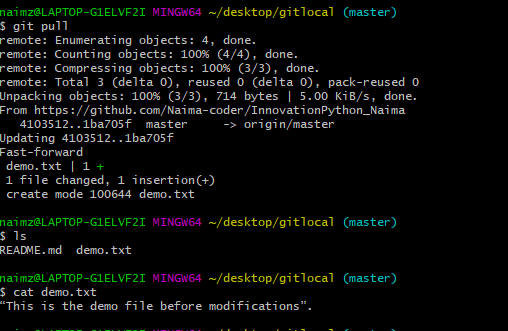
with content : “This is the demo file before modifications”.



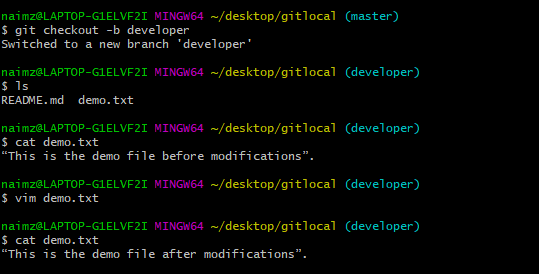
demo.txt file

4) 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 the demo file should be visible in your local machine.



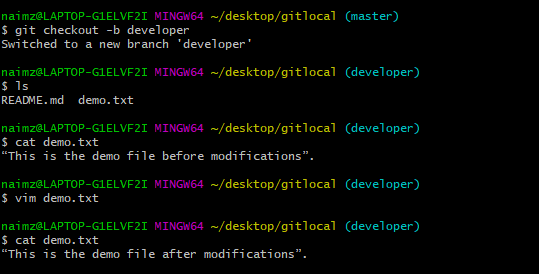
5)Now make a new branch in your local machine with the name “developer”.

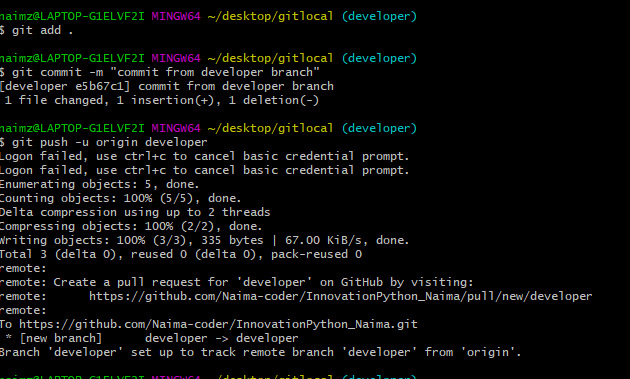


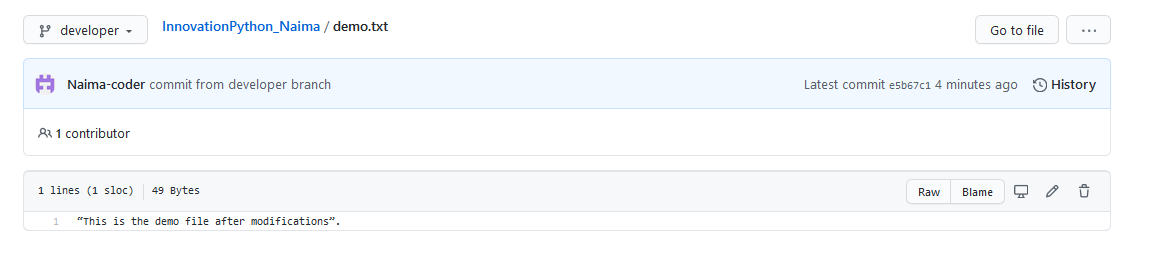
6)Edit that demo file and write some content in that eg: “This is a demo file after

modification” and push the modifications to your GitHub repo from the developer

branch with a commit message “Commit from developer branch”.

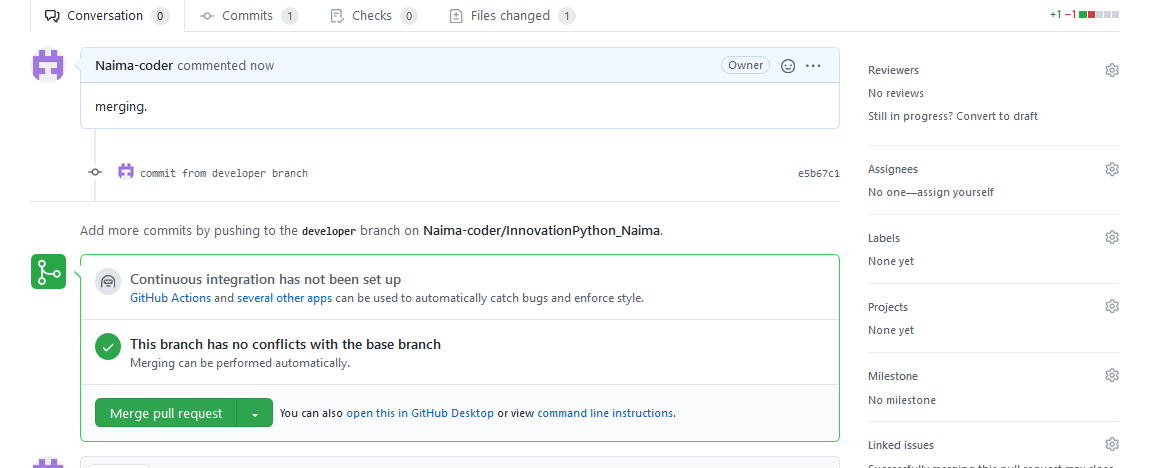


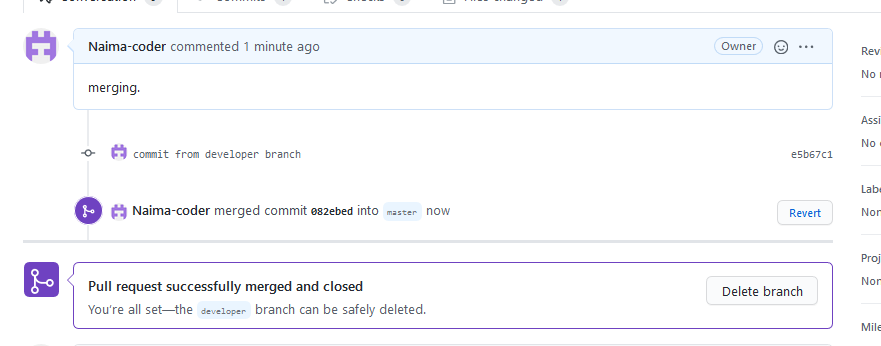




7)Go to the GitHub console and generate a merge request to master branch after

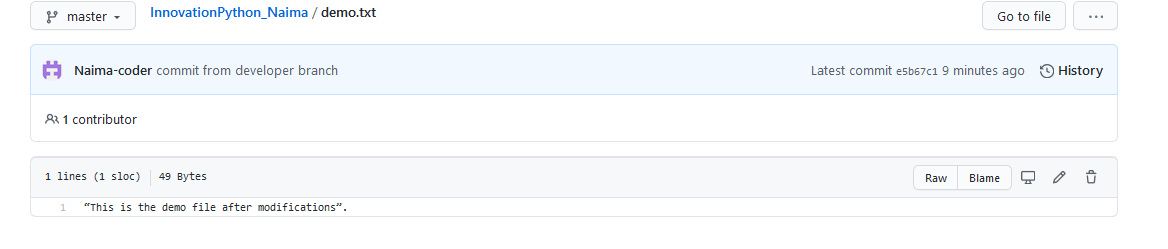
checking the modifications.

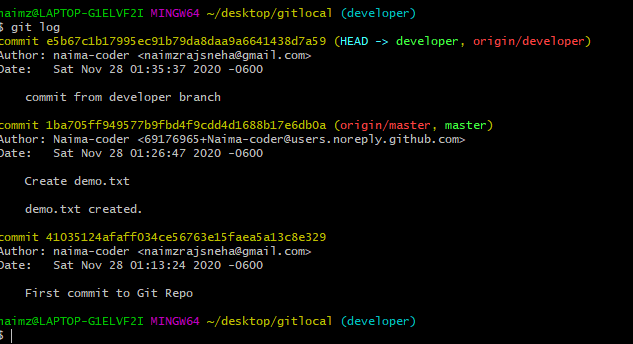


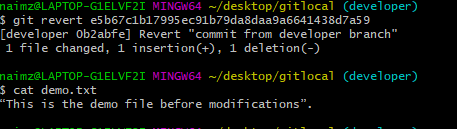


8)After merging you could see the modified content in the demo file. Now revert

back to the previous version from terminal.

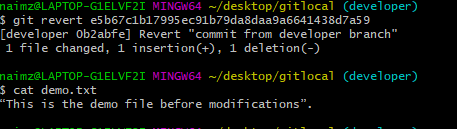






9)After switching back to the previous version your demo file should have the

content: This is a demo file before modification.



At the end delete the developer branch.

