**WEEK 01**

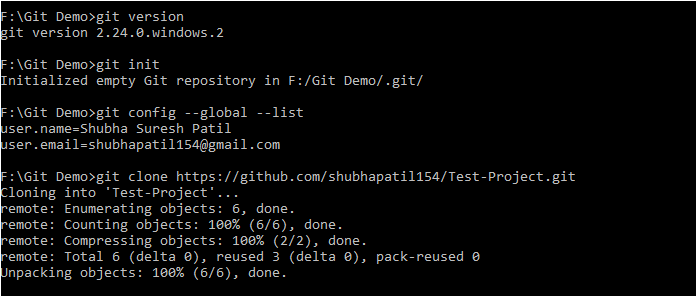
**ASSIGNMENT 01**

**TOPIC: GITHUB**

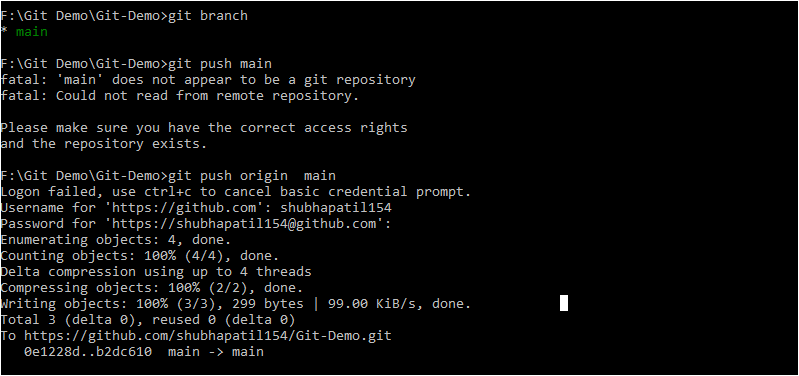
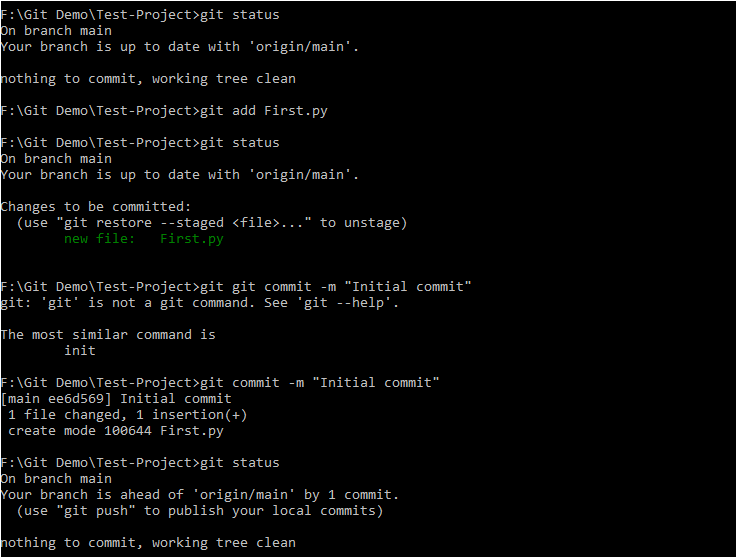
**NAME: SHUBHA SURESH PATIL**

**COLLEGE: BMSCE**

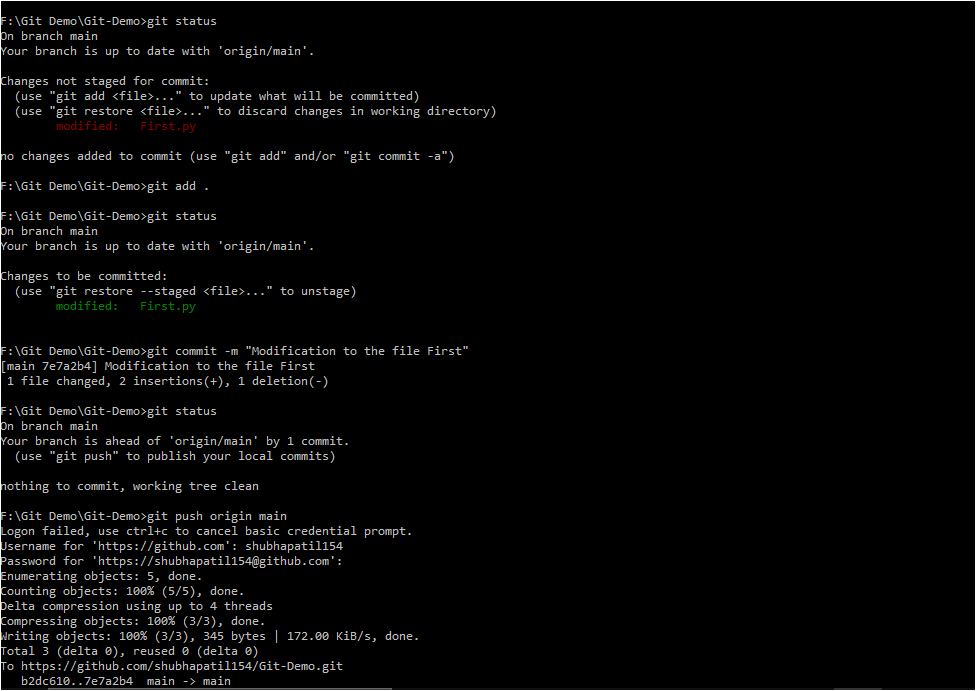
Local Repository setup

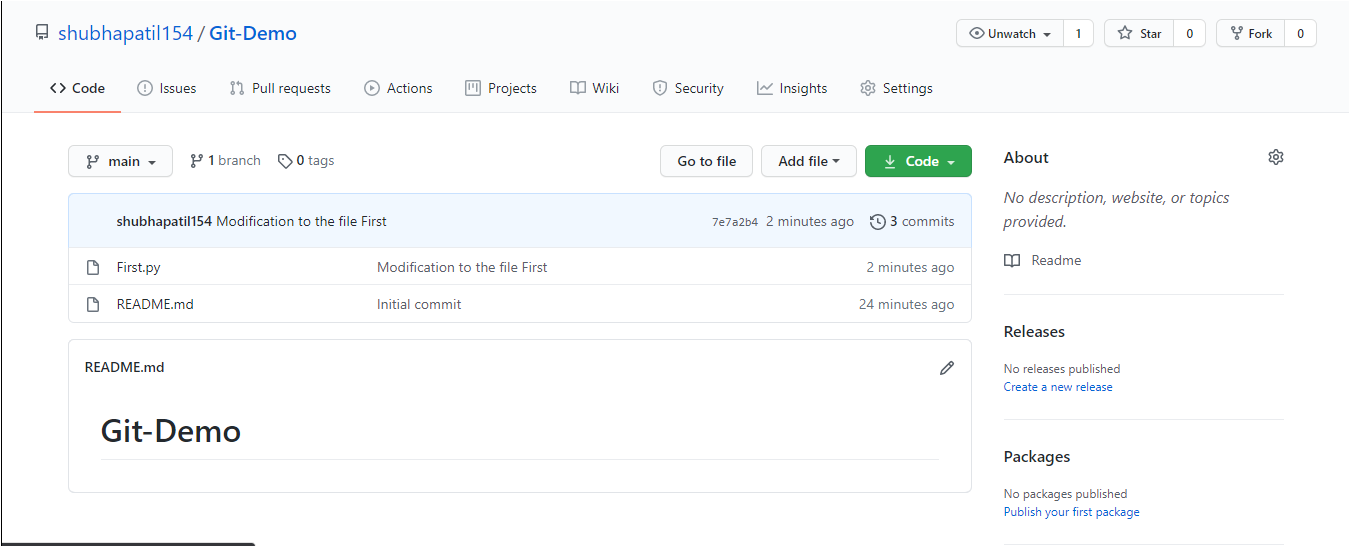


Adding File and pushing file

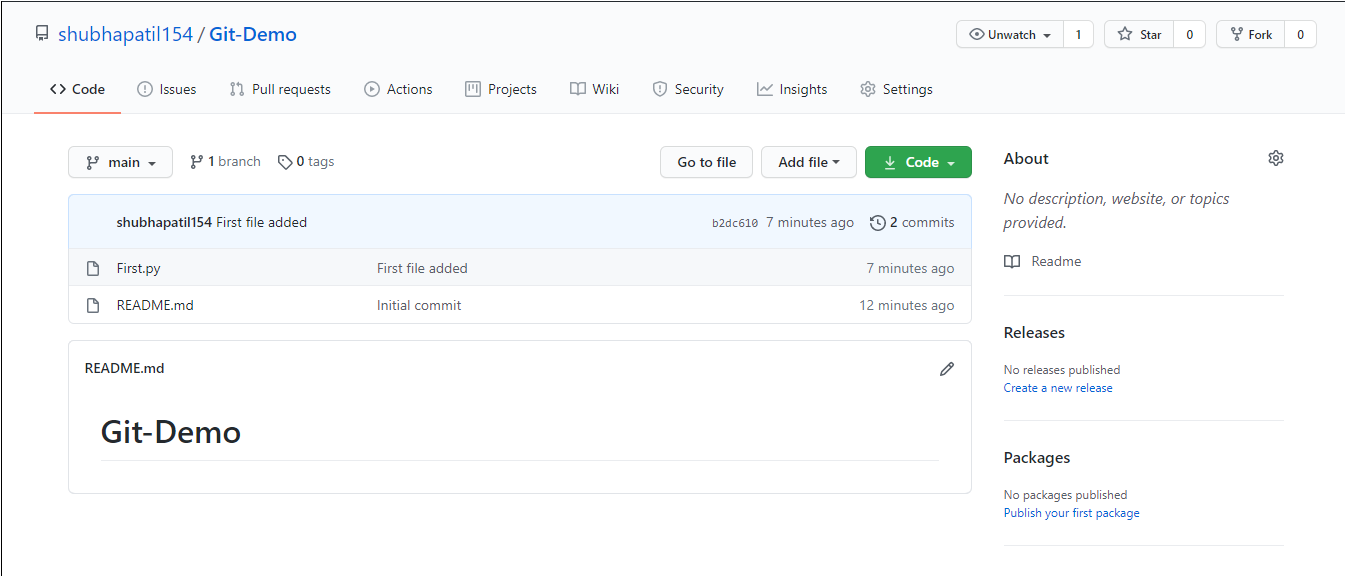


Modification to file

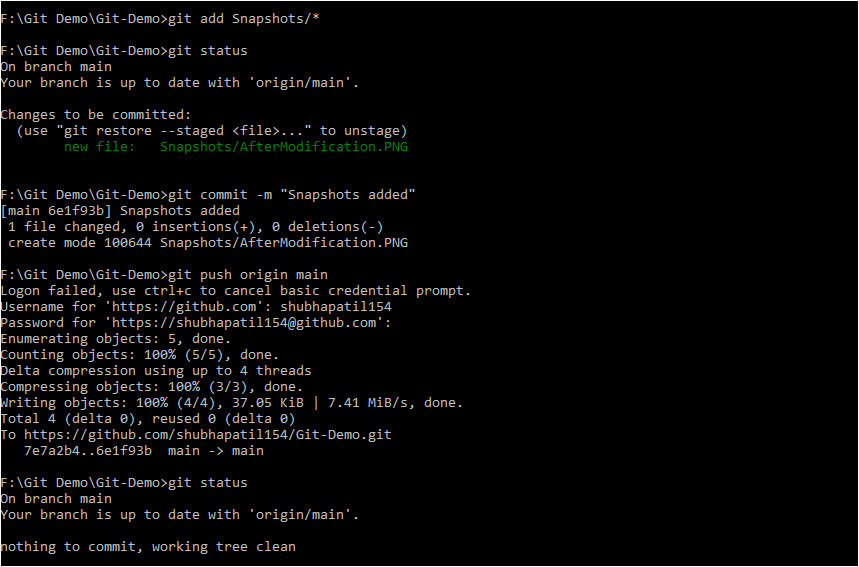


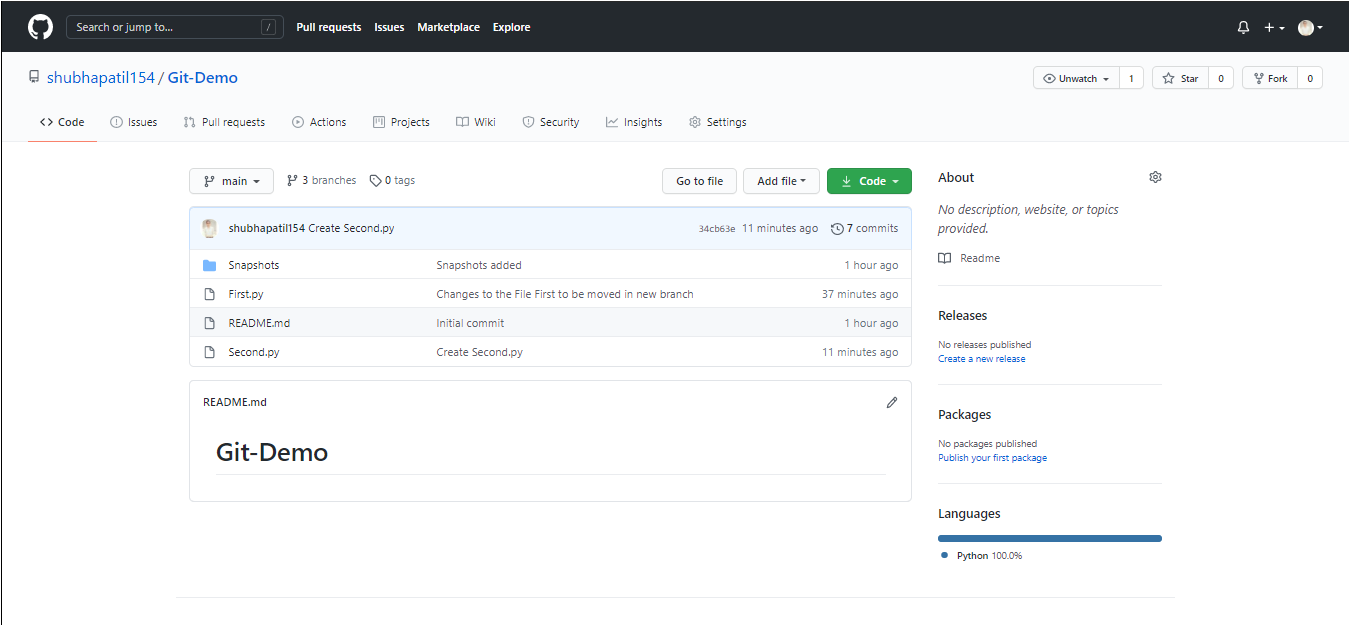


Remote Repository setup

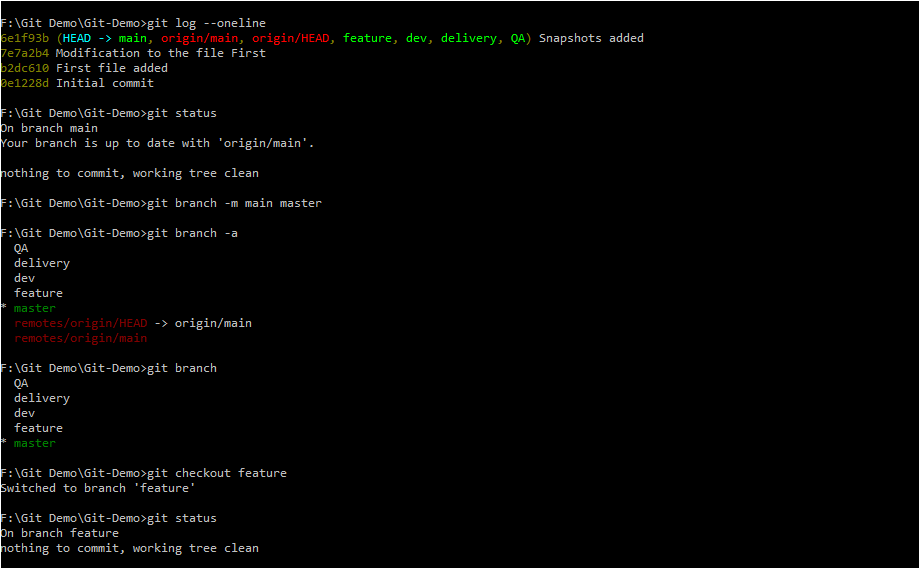
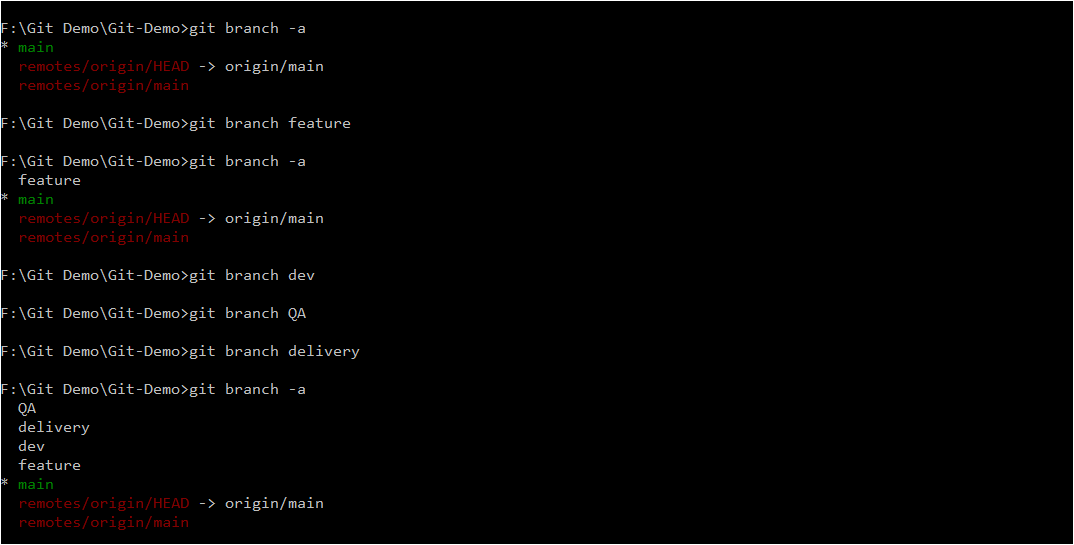
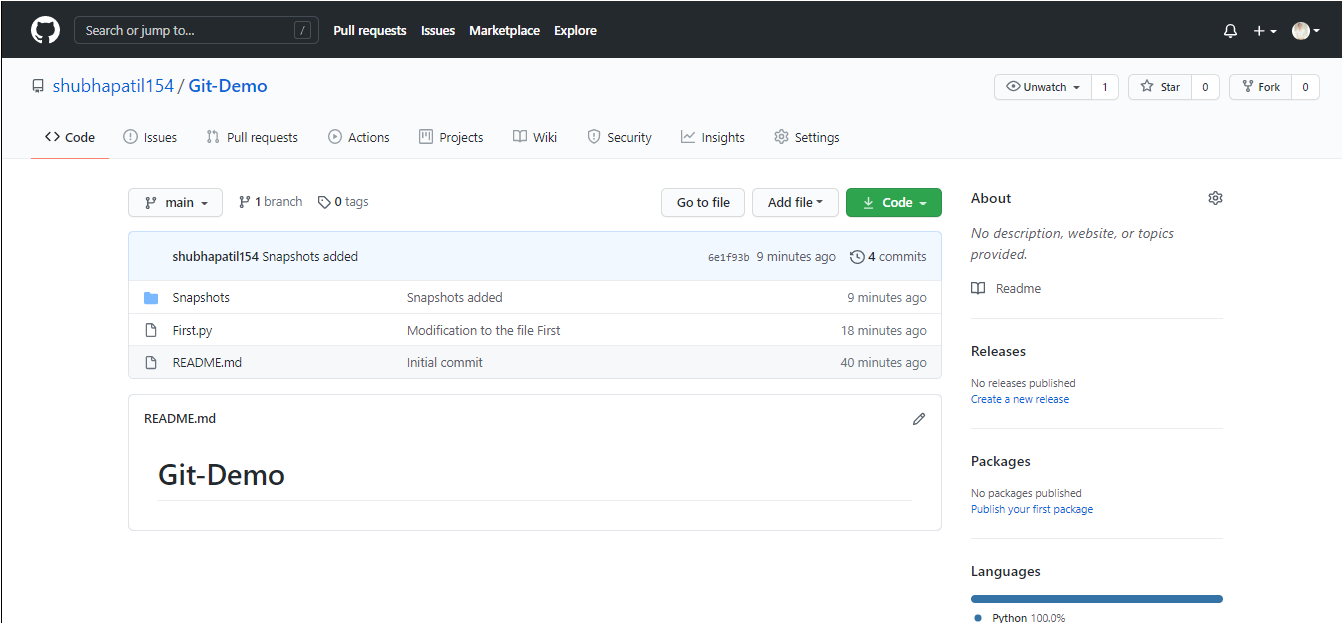
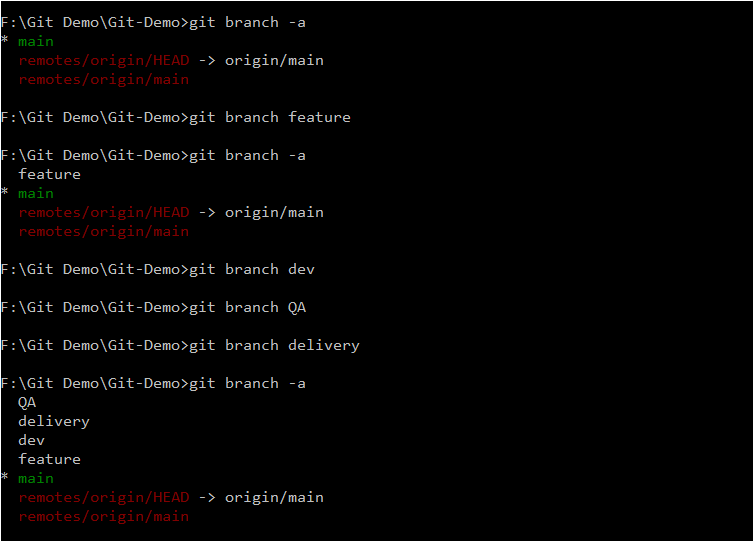


Adding folder and pushing the folder

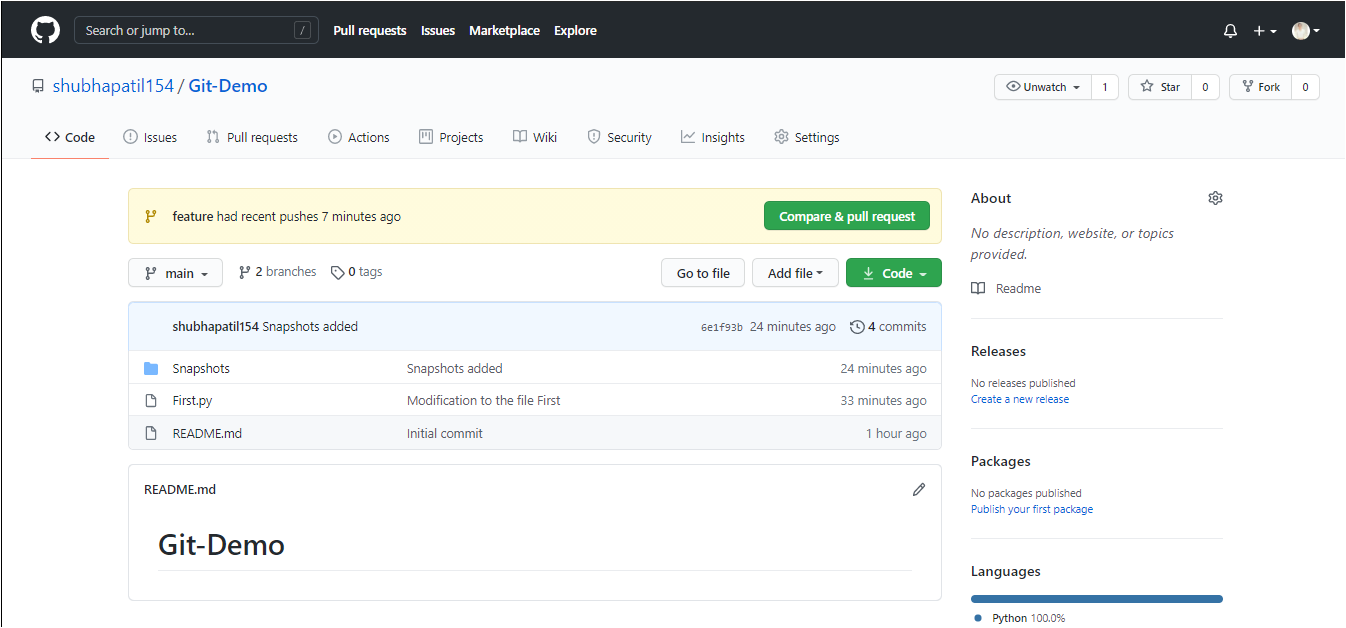
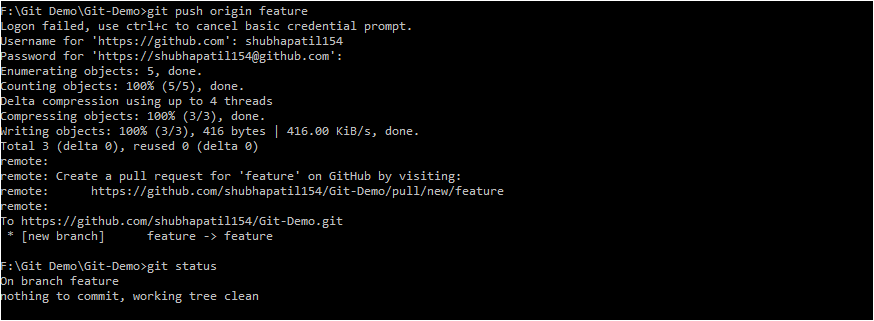
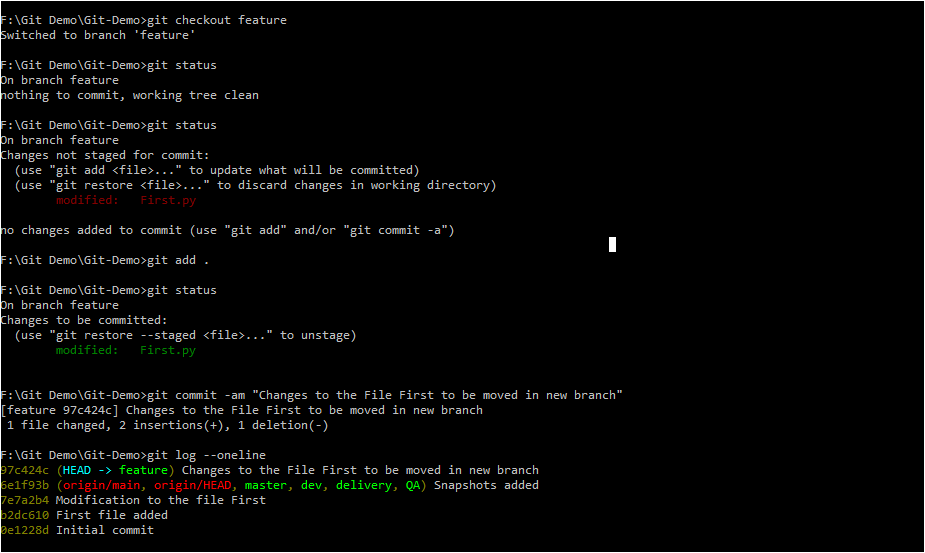




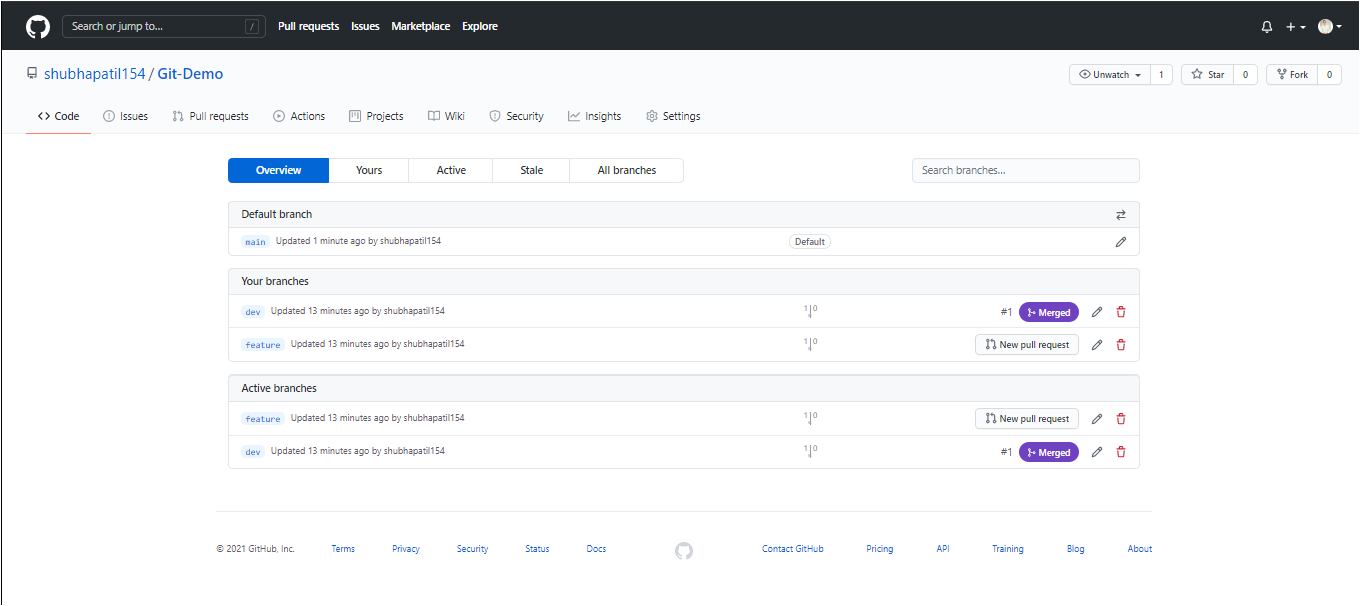
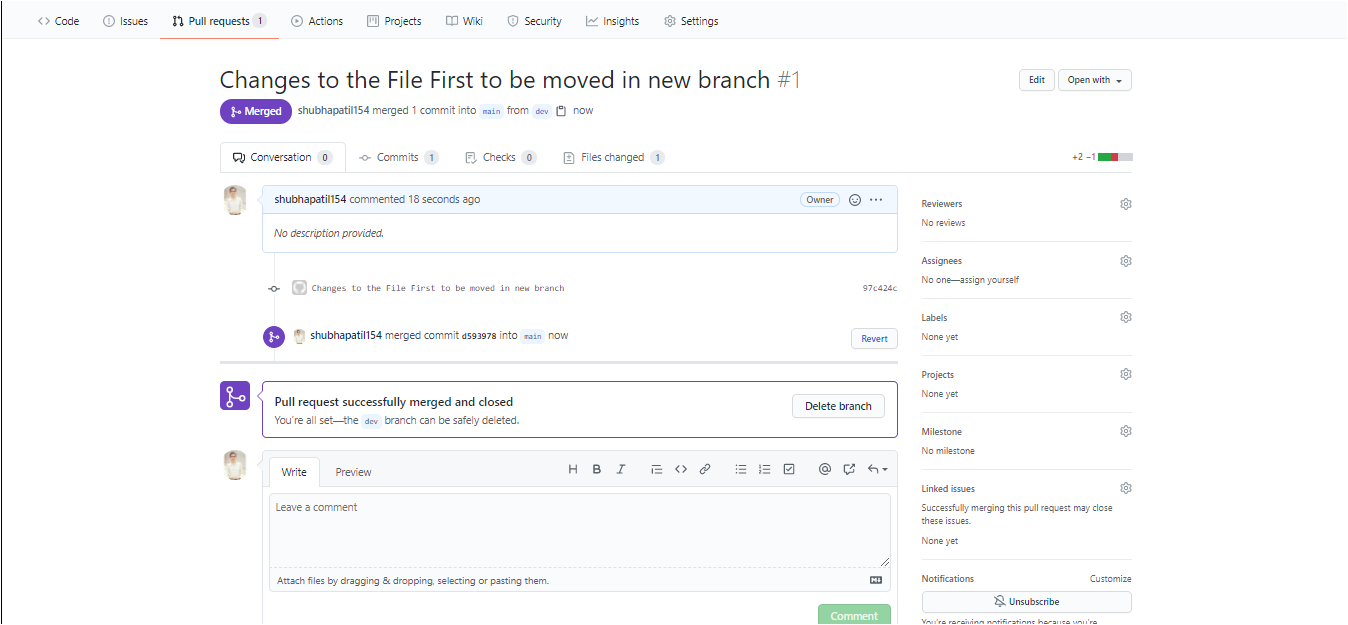
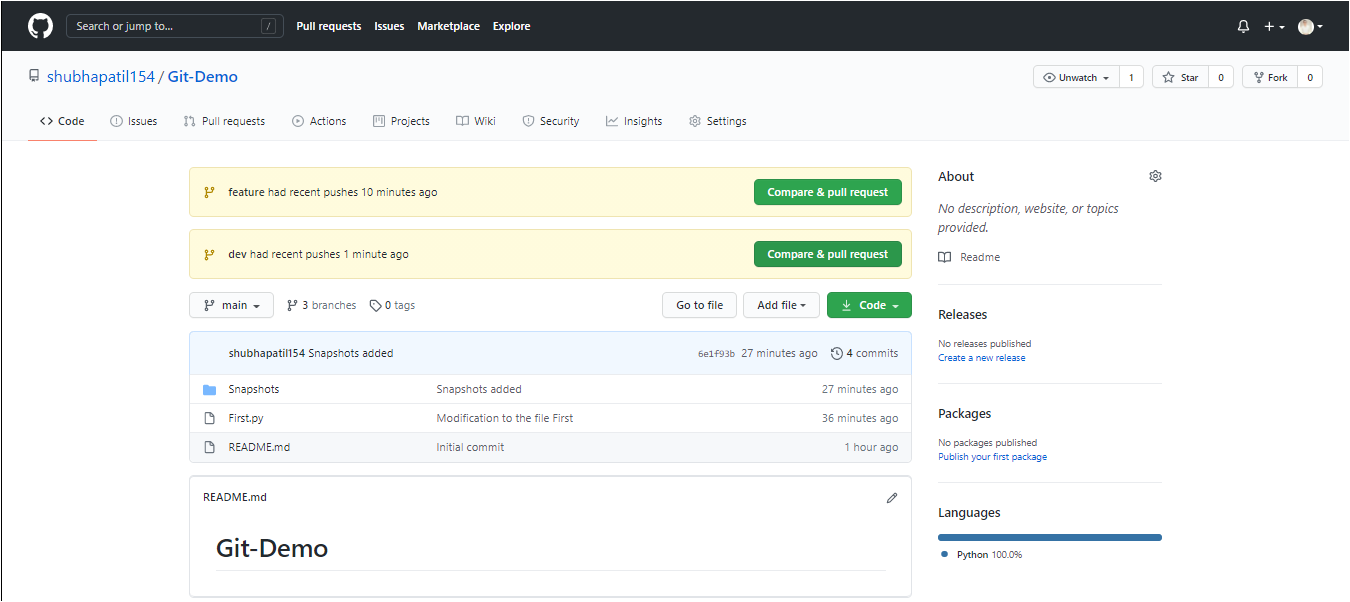
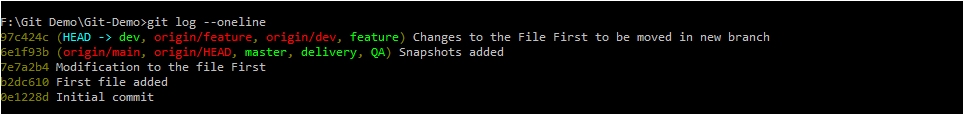
Creating branches



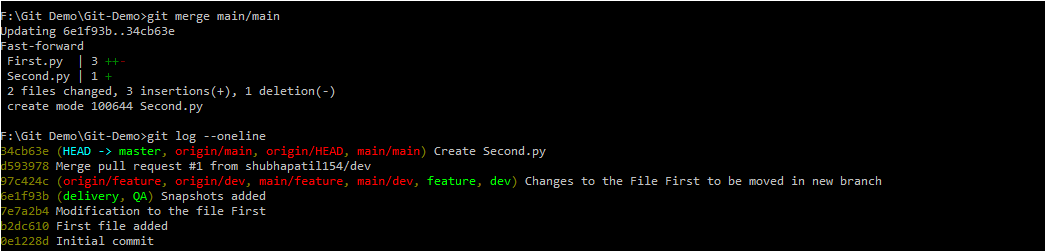
Checking out to feature branch



Promoting code from feature branch to dev branch



Fetching latest code from remote branch



Checkout vs Pull

* git pull contacts the remote repository identified by origin and looks for updates. It fetches any updates and then merges the changes into the target branch. It does not create a new branch.
* git checkout -b <branch> origin/<branch> creates a new branch based on origin/<branch>, and does not contact the remote repository. It looks at origin/<branch> as it currently exists in your local repository.
* In other words, git checkout command is used to switch (and optionally create) to a branch.
* “git pull” is a combination of two commands in one. It performs a “git fetch” which downloads changes of all remote branch (by default “origin” remote) and “git merge” which merge changes of that fetched onto your checked out branch. It often (depending on configuration settings) also sets your local checked out branch to track the origin branch as well. Also, based on your settings, it can either perform “rebase” merge or standard merge.