1. Create folder🡪 xyz.
2. Open git bash 🡪 working directory(xyz).
3. git init 🡪 (initialize git in this directory)
4. create first.txt and second.txt🡪 (remember these files are in untracked mode).
5. git status 🡪 Check status of untracked files.
6. git add first.txt second.txt OR git add. OR git add \*.txt 🡪 Add these two files in staging area.
7. git commit -m “implemented first feature” 🡪 Commit your files to VCS with commit message
8. git log OR git log remote/branch 🡪Check logs to see history

\*HEAD shows last commit refers your current branch\*

\*Commit Hash helps in identifying which user made specific commit\*

\* git reset will remove file from staging area and can remove changes in file only if they are not committed\*

9) git reset🡪 removes files from staging area if you can also remove changes you made in file you can use 🡪 git reset --hard. (doesn’t matter whether file is staged or not)

10) when you want to ignore some files existing in your directory but not want to be a part of VCS🡪

Create .gitignore in your working directory either directly or by using cmd touch .gitignore

Add \*.txt in. gitignore file if you want to ignore text files you can also add specific file in this ignore file and when you check git status it will ignore the files you added in .gitignore.

**Creating Branches on local server**

11) git branch 🡪 show list of branches

12) git branch -v 🡪 show list of branches with some details.

13) git branch feature 🡪 create new branch of name feature

14) git checkout feature 🡪 switch to new branch feature

15) git merge feature 🡪 merge feature branch into current active branch.

16) git log feature..master OR git log master..feature 🡪 show commit differences in two branches

**Pushing local Branches on Remote server**

17) git branch development 🡪 creates new branch of name “development” in master branch.

18) git checkout development 🡪 switch to development branch.

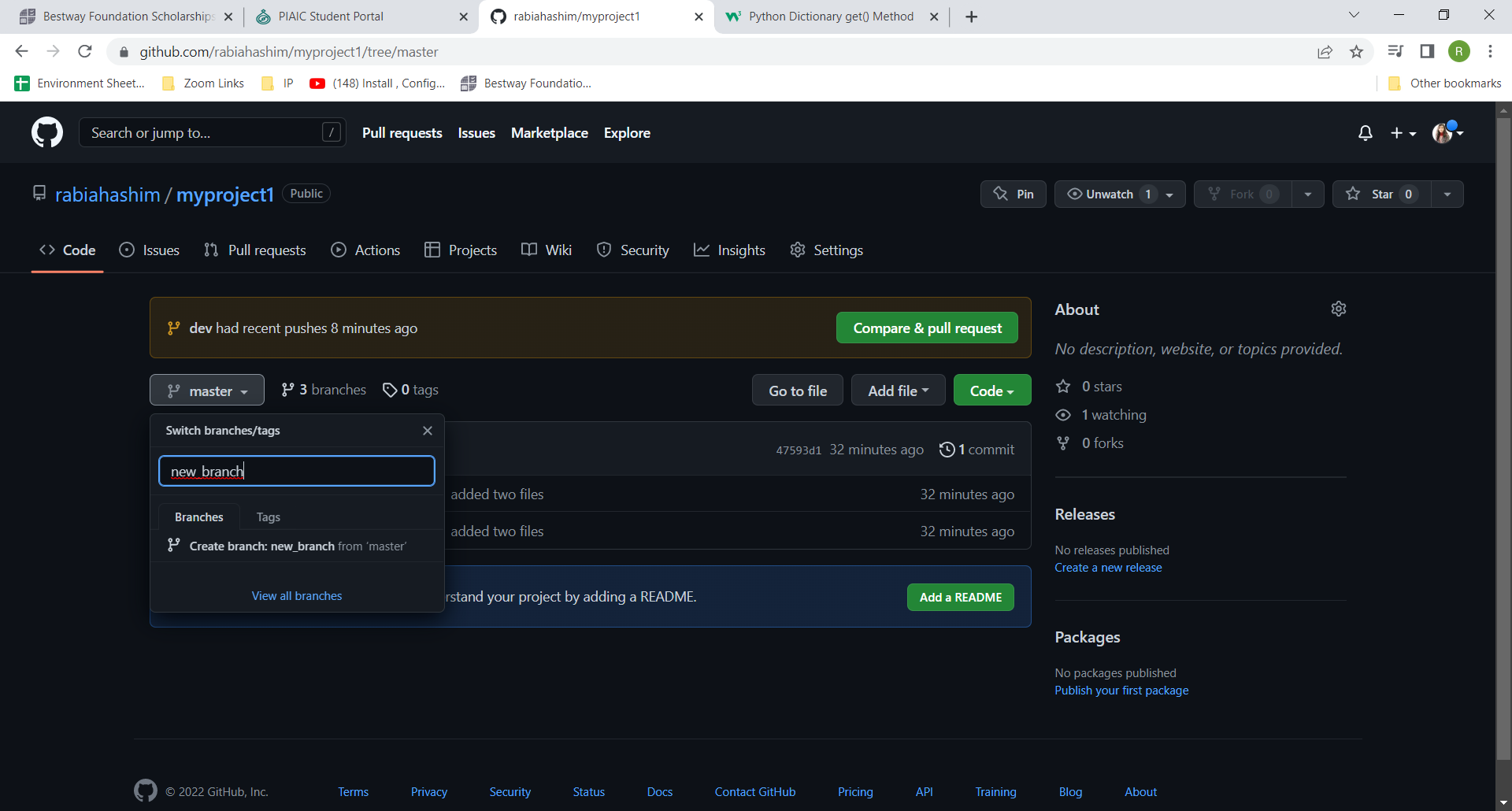
19) git push -u origin development 🡪 push changes on development branch

20) git checkout master and git merge development🡪 Switch to master so that merge dev into main branch but this is on local server.

21) git push origin development🡪 to show merge effect on remote server.

**Pushing Remote Branches on local server**

22) Create branch on remote



23) git pull 🡪 to see the new branch on your local server .

**Stash**

17) git stash 🡪 save local changes stash clipboard (those changes which are not completed for commit)

18) git stash save <name> 🡪 save local changes stash clipboard with specific name for your reference later.

19) git stash list 🡪 show list of stashes.

20) git stash pop 🡪 Apply latest stash and remove it from clipboard.

21) git stash apply stash@ {0} (check stash name from git stash list) 🡪 Apply latest stash and maintain stash list history in record.

**Remote Repositories**

Create account on any remote location like GitHub etc.

Create a repository over there where you want to push your local changes.

22) git push 🡪 push changes to remote repository (best practice is to use git push origin master)

23) git fetch 🡪 fetch changes from remote repository

24) git merge 🡪 merge changes that was fetch by git fetch command.

25) git pull 🡪 fetch and merge changes from remote repo.

26) git remote -v 🡪 show remote URLs.

27) git remote add **demoproject** https:/…/**demoproject**.git 🡪 add remote repo in local repository

28) git remote show (origin/**demoproject**) 🡪 show details of origin.

29) git push -u **demoproject** master