**GIT**

GIT stands for Global Information Tracker. It tracks changes in the source code, enabling multiple developers to work together on non-linear development.

Git is a Version Control system. A version control system (VCS) tracks every alteration to a file or set of files, enabling developers to journey back to earlier versions and collaborate seamlessly.

**TYPES OF VCS:**

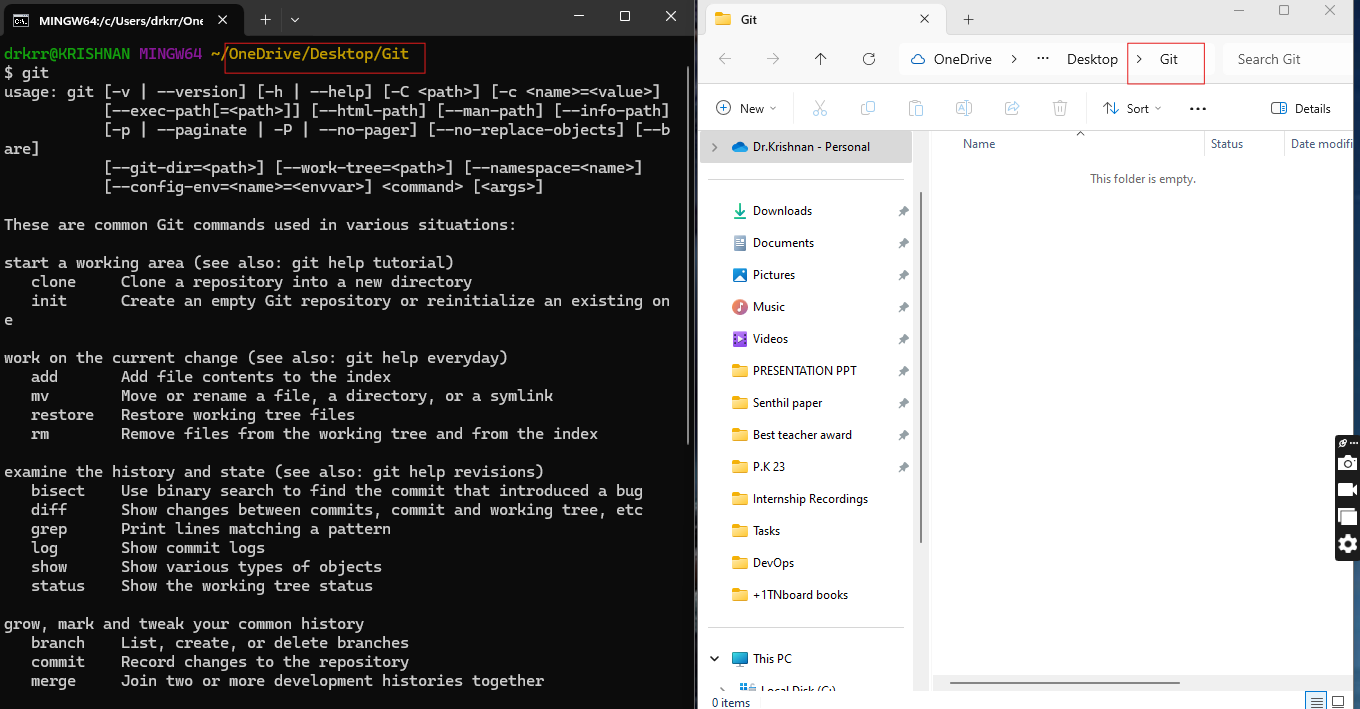
* **Centralized Version Control System**: A [centralized version control system](https://about.gitlab.com/topics/version-control/what-is-centralized-version-control-system/) (CVCS) is a type of VCS where all users are working with the same central repository. This central repository can be located on a server or on a developer's local machine. Example: CVS and SVN
* **Distributed Version Control System:** A distributed version control system (DVCS) allows users to access a repository from multiple locations. DVCSs are often used by developers who need to work on projects from multiple computers or who need to collaborate with other developers remotely. Example: Git

**GITHUB**

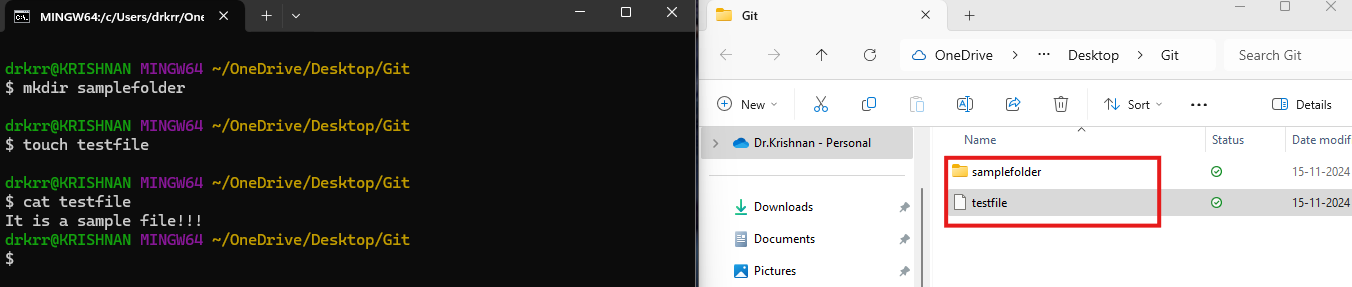
GitHub is a developer platform that allows developers to create, store, manage and share their code

**TASK: To create multiple branches with files, track the versioning of files and merge into a single file. Share git files into Github repository, with the help of CLI.**

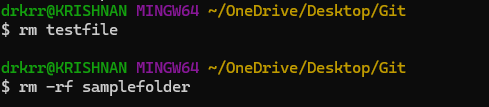
Step 1: Download and install Git in desktop.



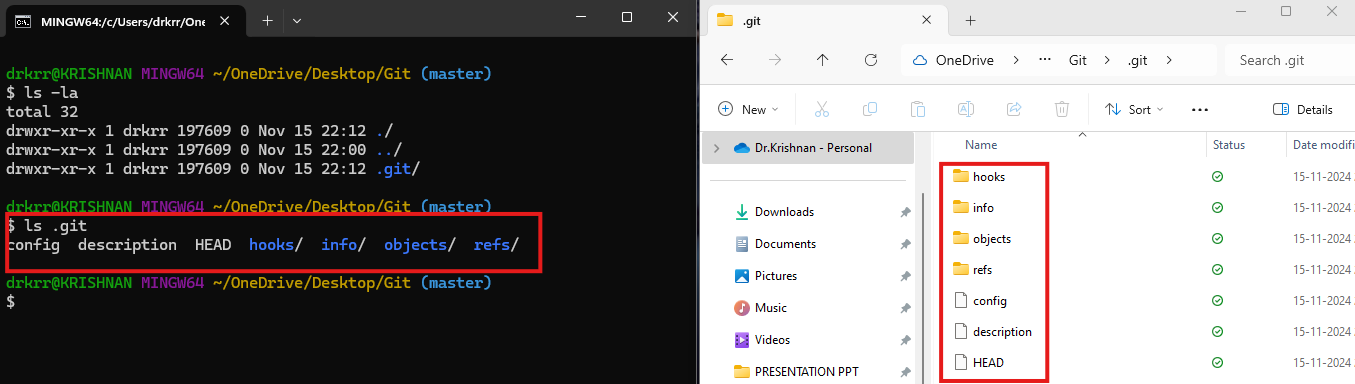
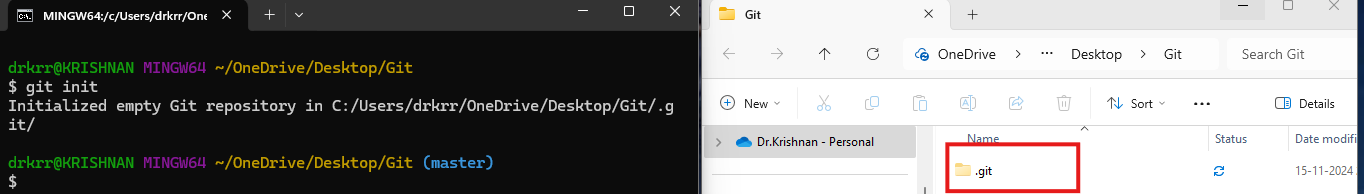
Step 2: Commands to create folder, file and to view a file content.

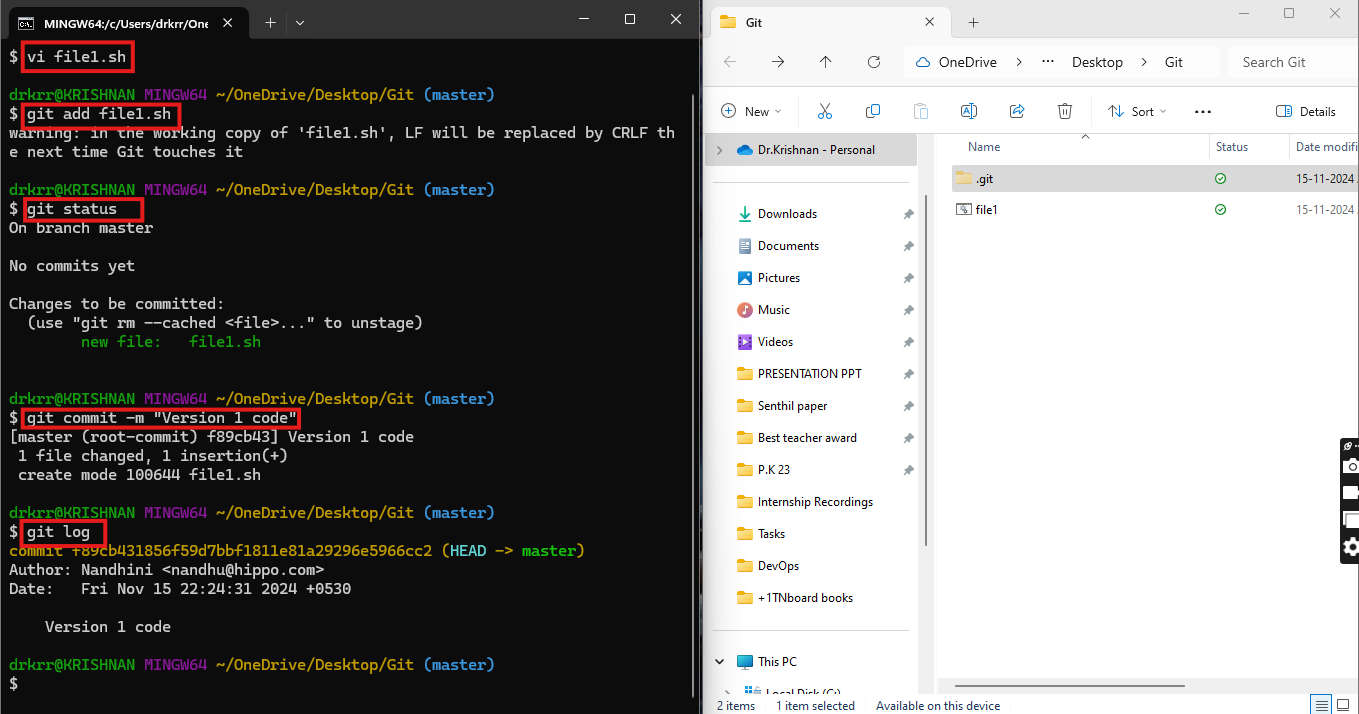


Step 3: Commands to remove file and folder.

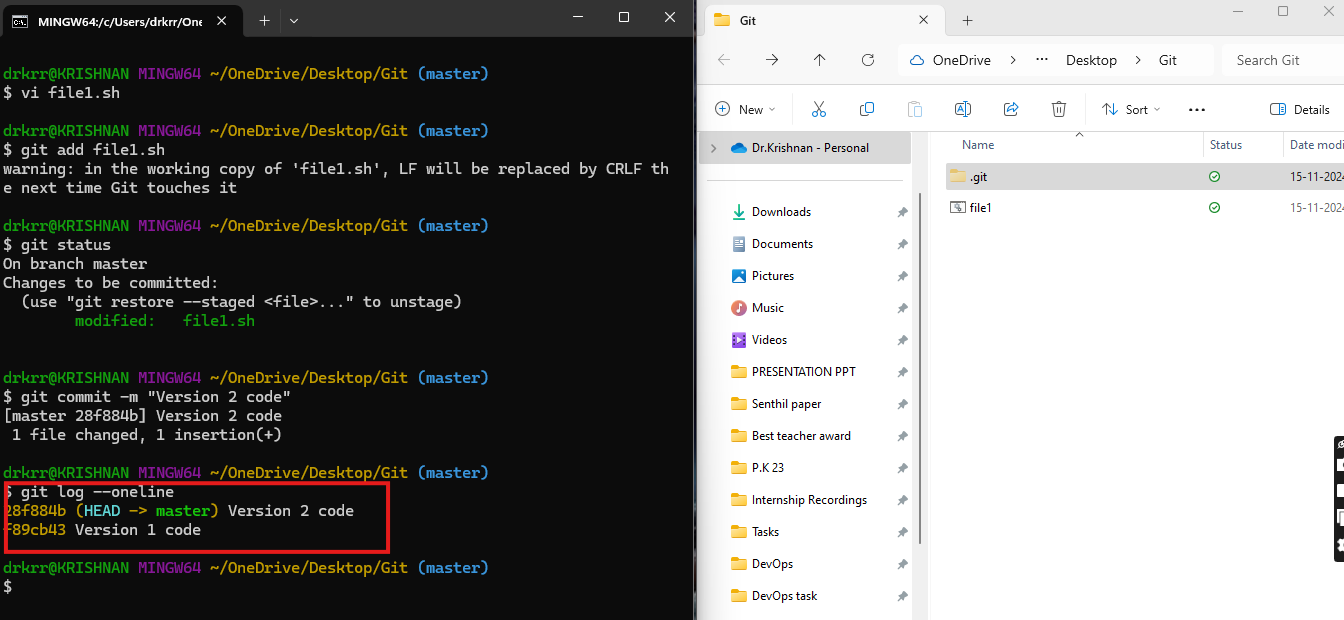


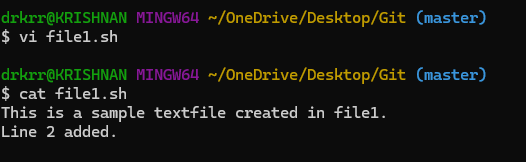
Step 4: Initiate a git in folder, view the files in Git.



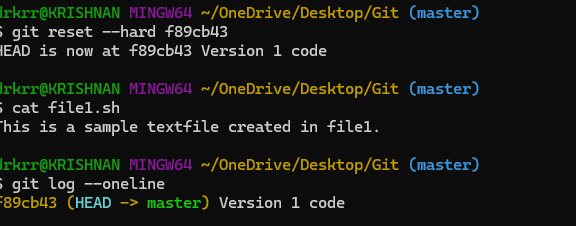
Step 5: Create file, add file for tracking, commit to store the version of file.

Step 6: Modify the file and check for versioning info using Git log command.

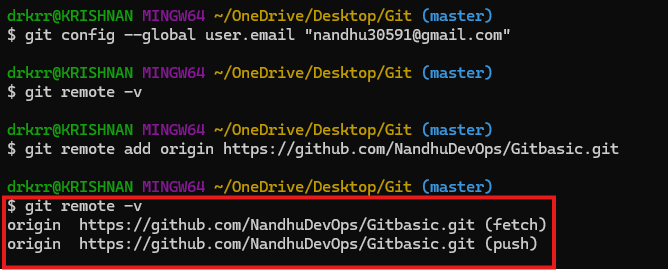
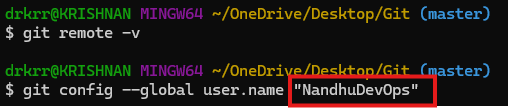




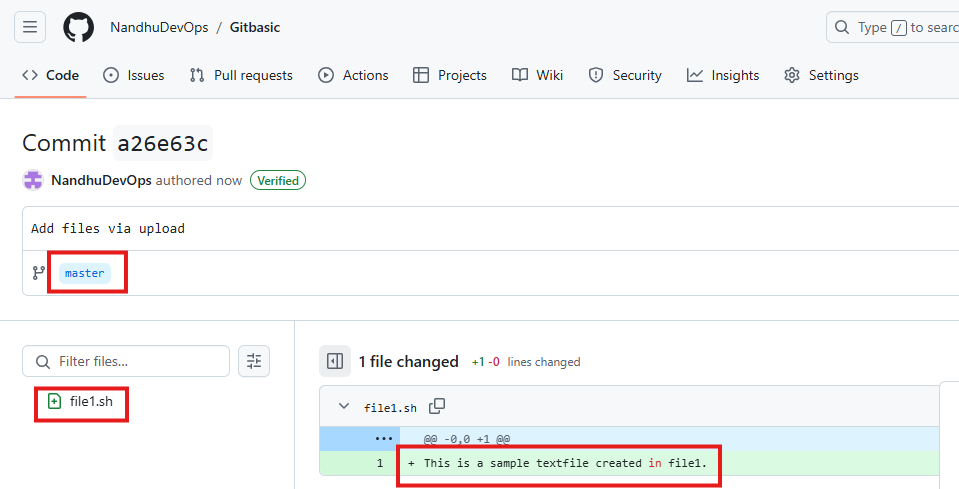
Step 7: Rollback to version 1 using git reset command



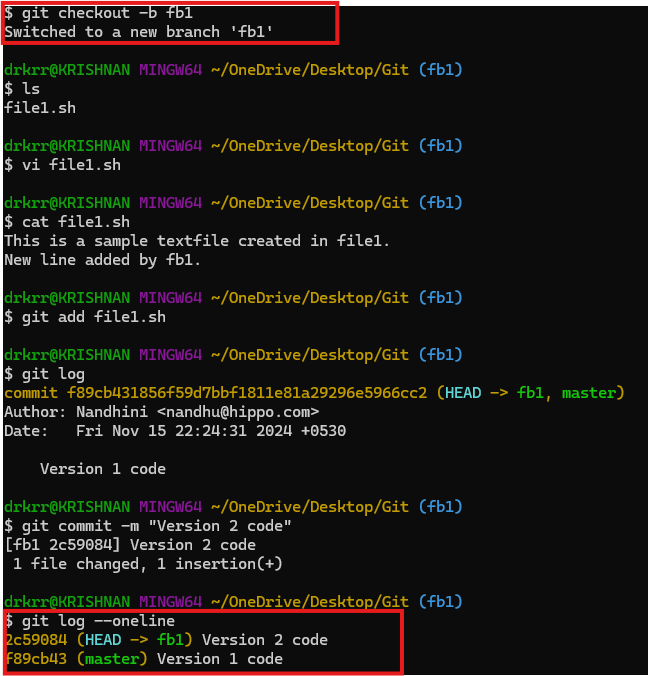
Step 8: Configure Git with Github credentials to push files into repository.



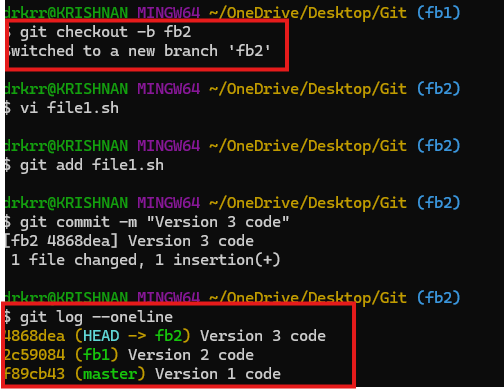
Step 9: Pushed file1 into master branch, in Gitbasic repository

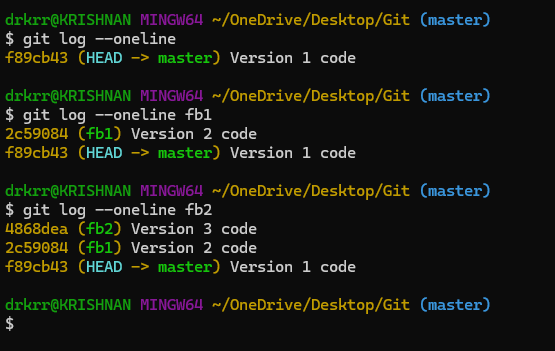


Step 10: Create another branch fb1, edit the file, track and commit it.

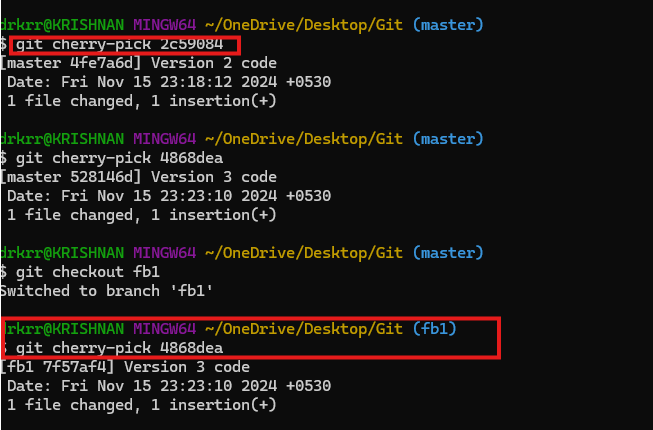


Step 11: Create 3nd branch fb2, edit the file, track and commit it.

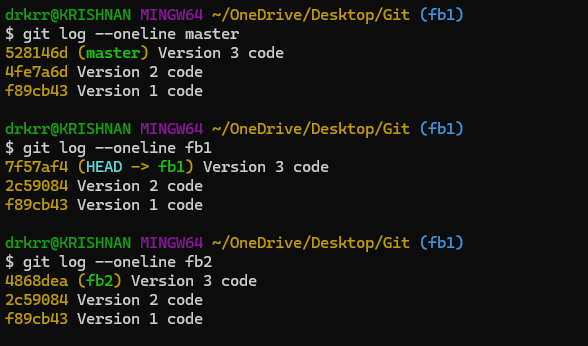


Step 11: Checked Git log for all branches, the changes made in feature branches is not reflected in master branch.

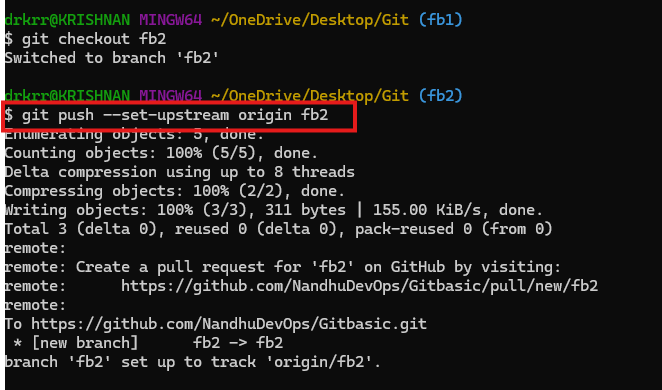
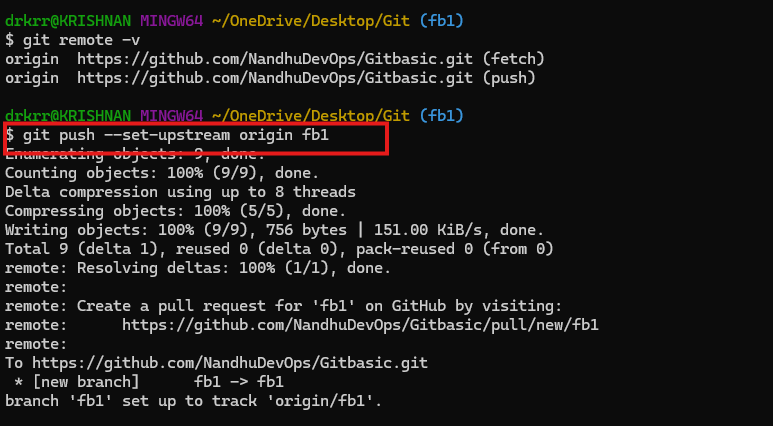
Step 12: Cherry-pick, merge and rebase are the commands used to apply commit from multiple branches into master branch. Cherry picking is the act of picking a commit from a branch and applying it to another.

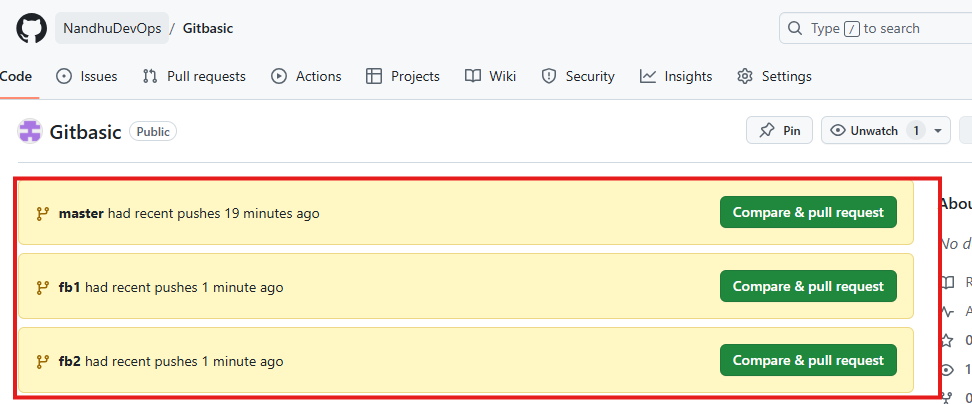


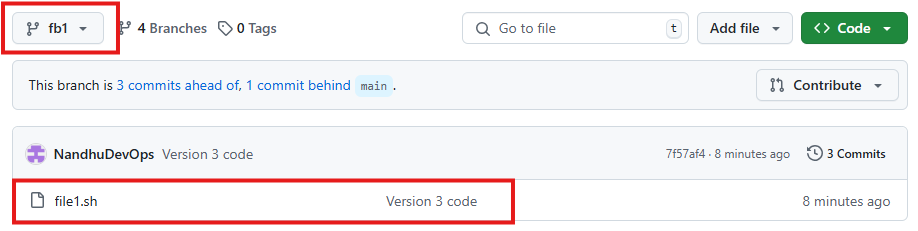
Step 14: Now all branches have same version copies.

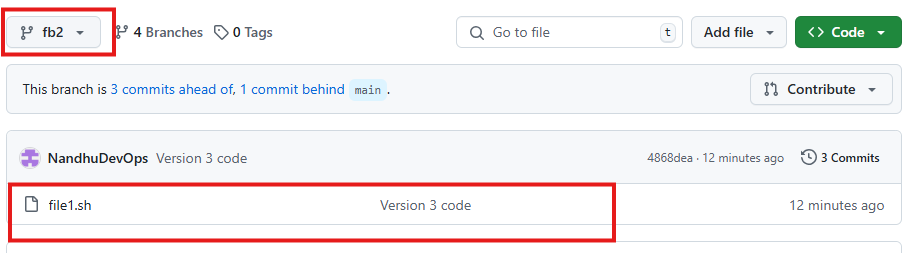


Step 15: Pushed all the branches into Github along with files.









Step 16: Deleted all the branches in repository.

