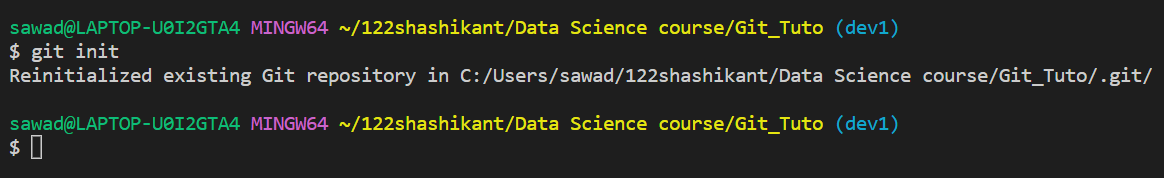
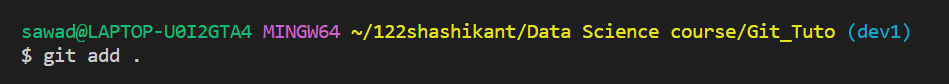
**TASK-1**

**Demonstrate minimum 15 basic Git command with explanation and screenshot**.

**1.git init:-** The git init command is used to create a new blank repository. It is used to make an existing project as a Git project.

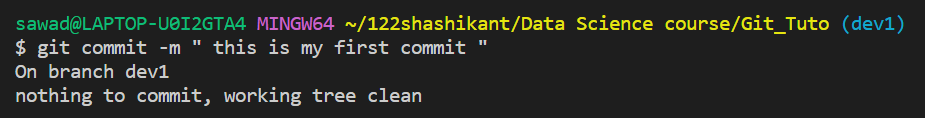


**2. git add:-** The git init command is used to create a new blank repository. It is used to make an existing project as a Git project.

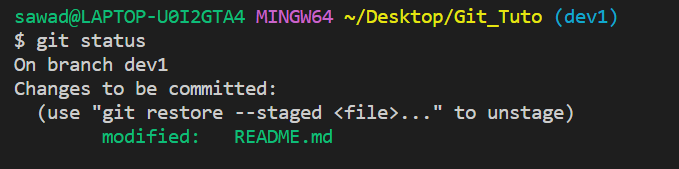


**3. git commit:-** It is used to record the changes in the repository. It is the next command after the

“ Git add. “Every commit contains the index data and the commit message.



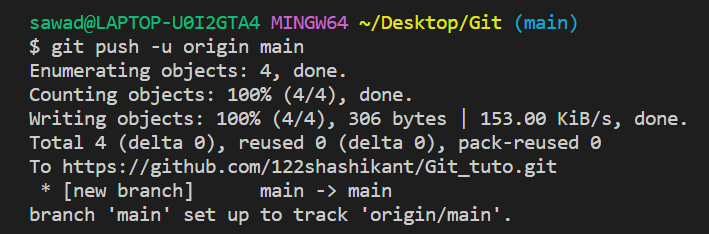
**4. git status :-** The git status command is used to display the state of the repository and staging area. It allows us to see the tracked, untracked files and changes.



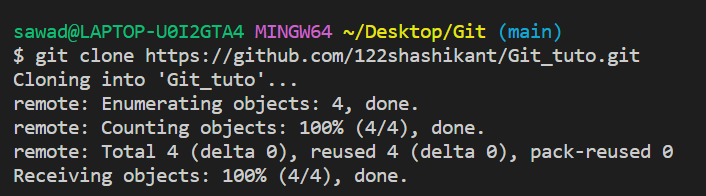
**5. git remote :-** In Git, the term remote is concerned with the remote repository. It is a shared repository that all team members use to exchange their changes.

****

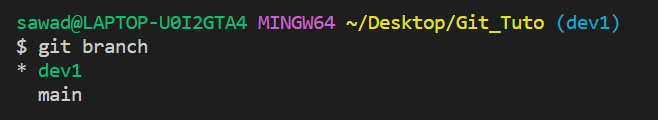
**6. git push :-** The push term refers to upload local repository content to a remote repository. Pushing is an act of transfer commits from your local repository to a remote repository.



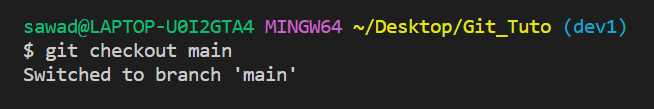
**7. git clone:-** In Git, cloning is the act of making a copy of any target repository. The target repository can be remote or local. You can clone your repository from the remote repository to create a local copy on your system. Also, you can sync between the two locations.

****

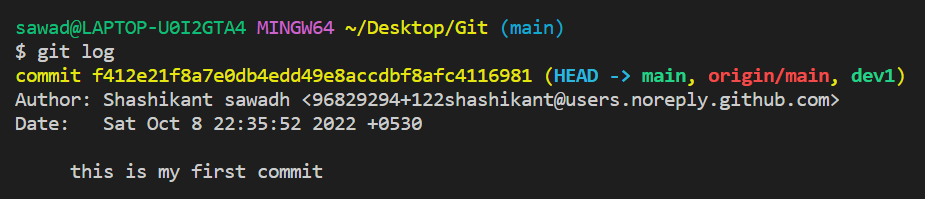
**8. git branch :-** multiple developers working on the same project or repository! To handle the workspace of multiple developers, we can use branches. To create a branch (say, the ‘name-of-the-branch’ is ‘branch1’), we use this command



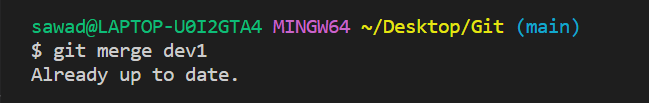
**9. git checkout :-** The **git checkout** command is used to switch between branches in a repository. Be careful with your staged files and commits when switching between branches.

****

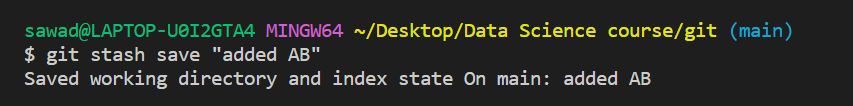
**10. git log :-** Git log command is one of the most usual commands of git. It is the most useful command for Git. Every time you need to check the history, you have to use the git log command. The basic git log command will display the most recent commits and the status of the head. It will use as:

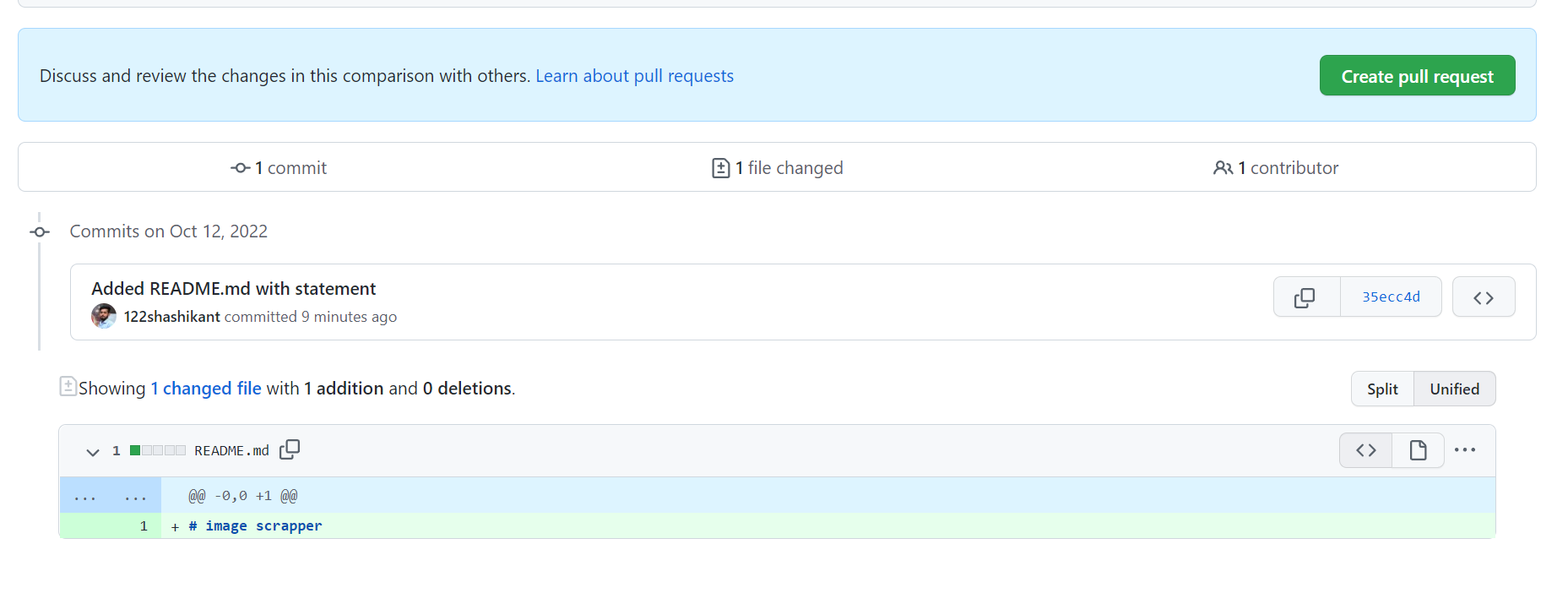
****

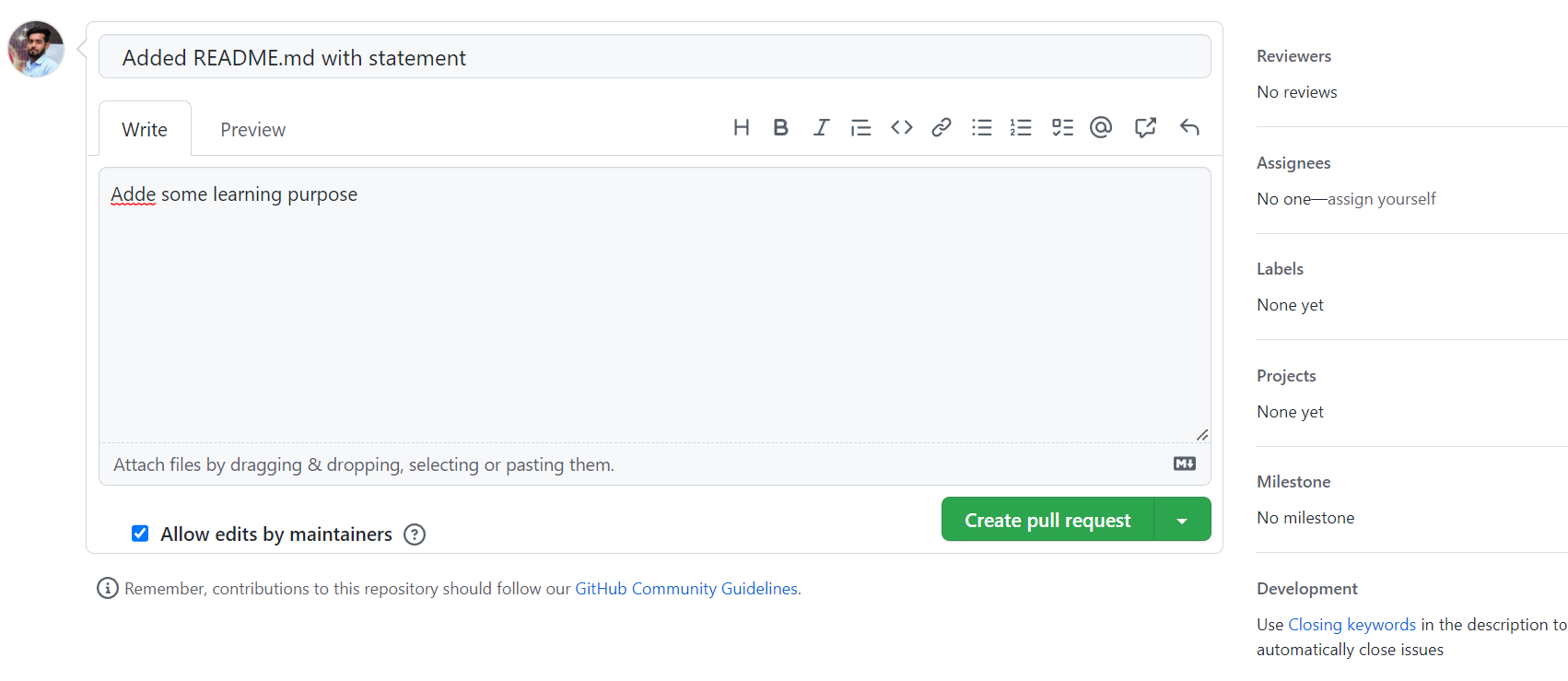
**11. git merge :-** This command will combine multiple sequences of commits into one unified history. In the most frequent use cases, git merge is used to combine two branches. The git merge command takes two commit pointers, usually the branch tips, and finds a common base commit between them. Once it finds a common base commit, it will create a commit sequence.

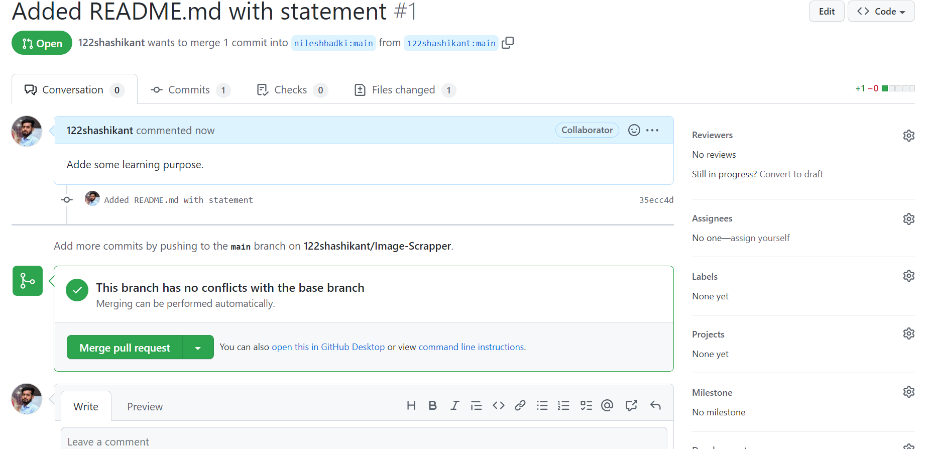


**12. git stash:-** Sometimes you want to switch the branches, but you are working on an incomplete part of your current project. You don't want to make a commit of half-done work. Git stashing allows you to do so. The **git stash command** enables you to switch branches without committing the current branch.

****

**13. git pull :-** The term pull is used to receive data from GitHub. It fetches and merges changes from the remote server to your working directory. The **git pull command** is used to pull a repository.





**14. git ignore:-** In Git, the term "ignore" is used to specify intentionally untracked files that Git should ignore. It doesn't affect the Files that already tracked by Git.Sometimes you don't want to send the files to Git service like GitHub. We can specify files in Git to ignore.

**13. git diff :-** 1> **Track the changes that have not been staged.**

**2>** **Track the changes that have staged but not committed**

**14. git rebase :-** Rebasing is a process to reapply commits on top of another base trip. It is used to apply a sequence of commits from distinct branches into a final commit. It is an alternative of git merge command. It is a linear process of merging.

**15. git fetch :-** Git "fetch" Downloads commits, objects and refs from another repository. It fetches branches and tags from one or more repositories. It holds repositories along with the objects that are necessary to complete their histories to keep updated remote-tracking branches.