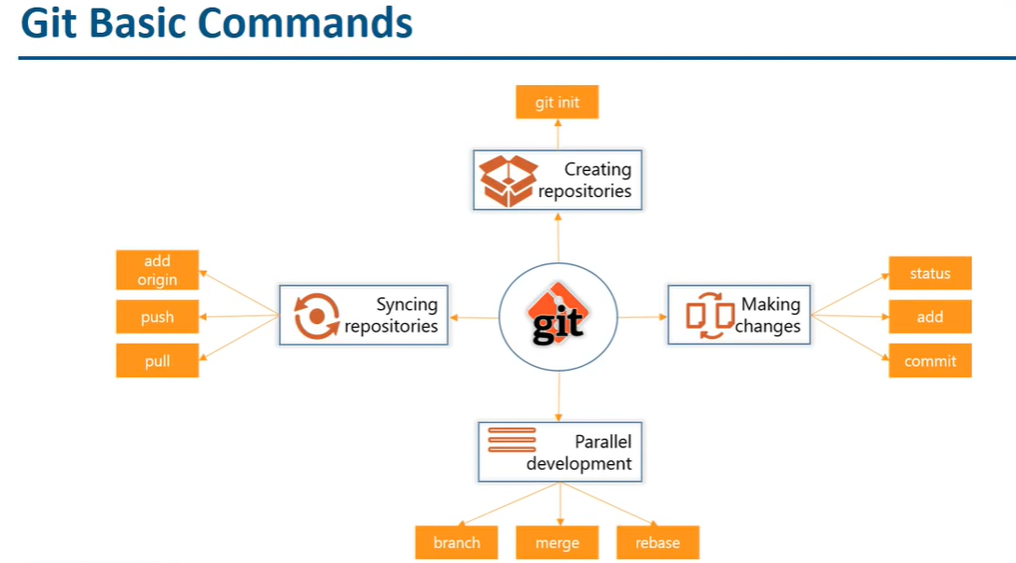
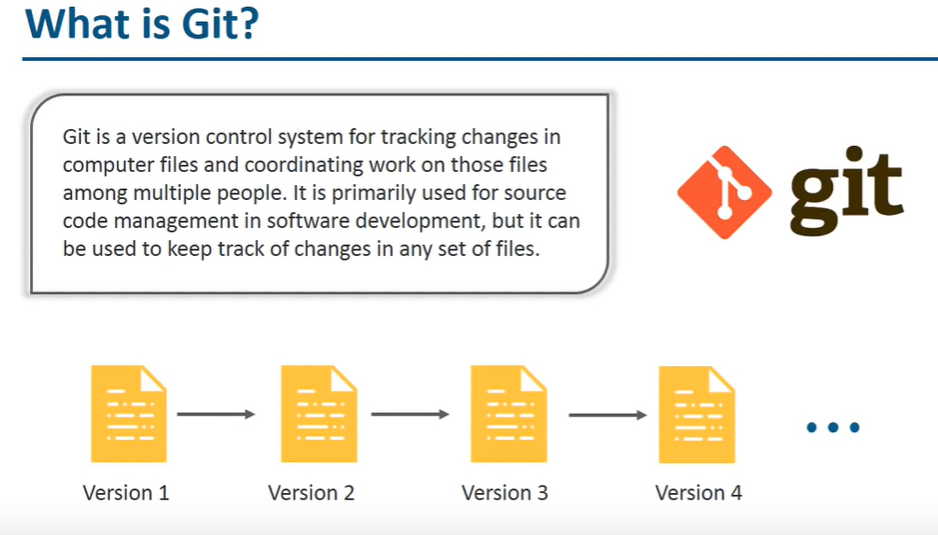
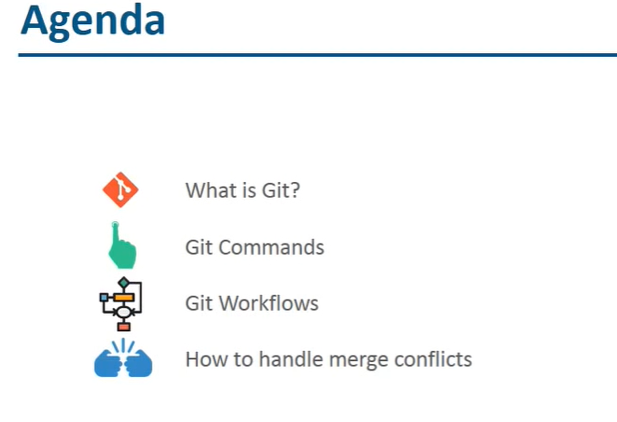
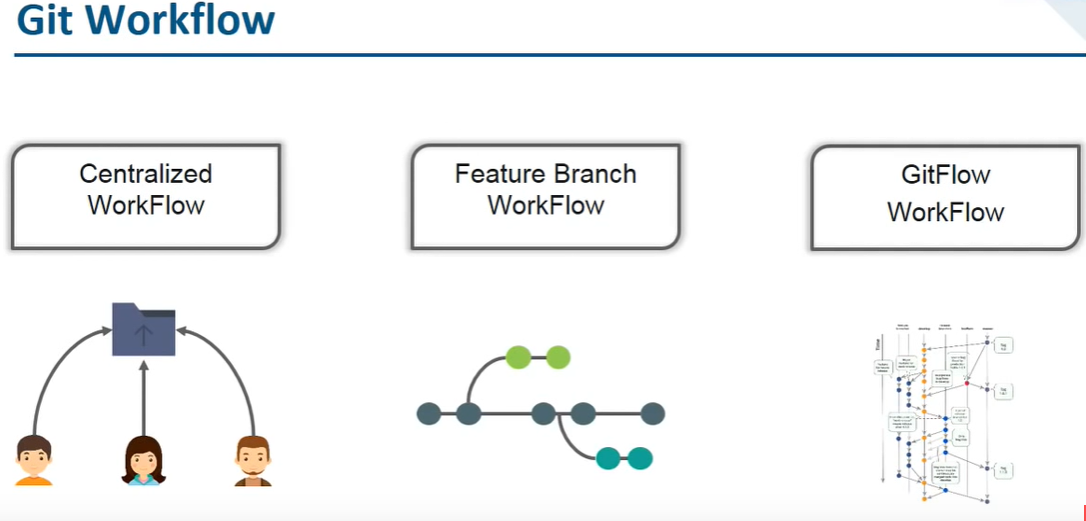
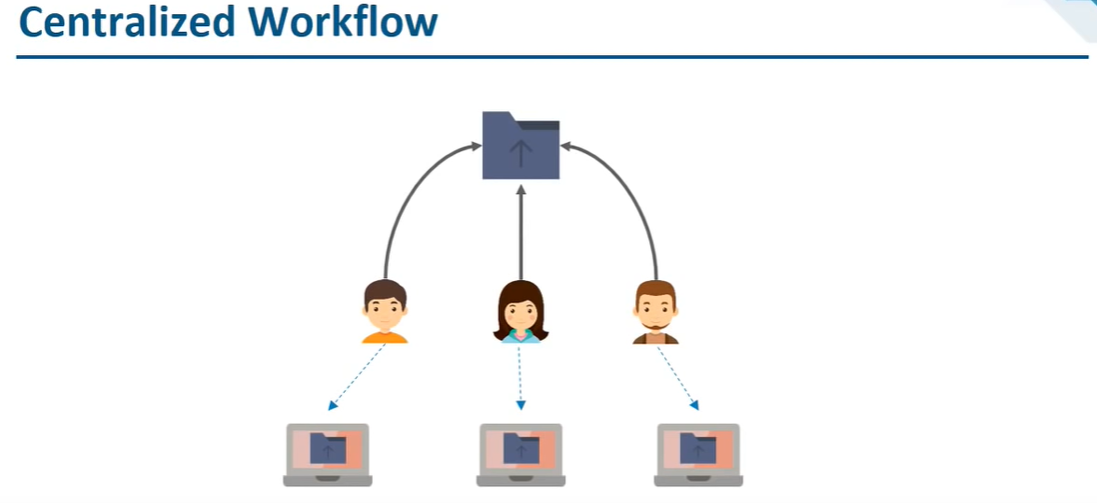
**Git Merge Conflicts Tutorial**





**Solving Merge Conflicts**

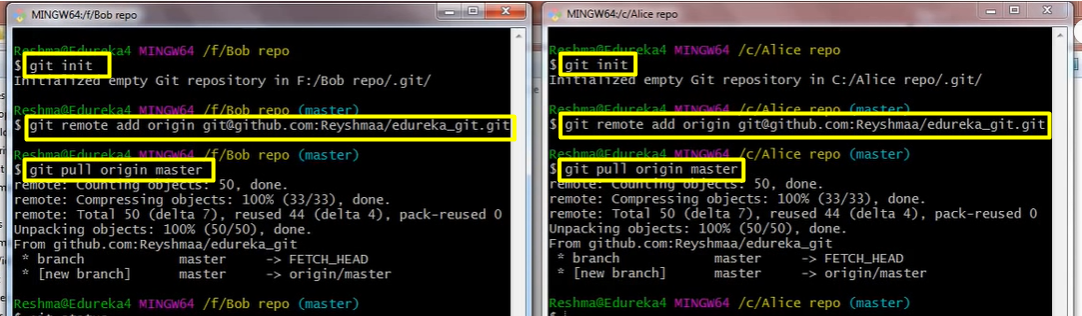


-> Centralized workflow doesn’t need any branches, only branch we have is master branch.

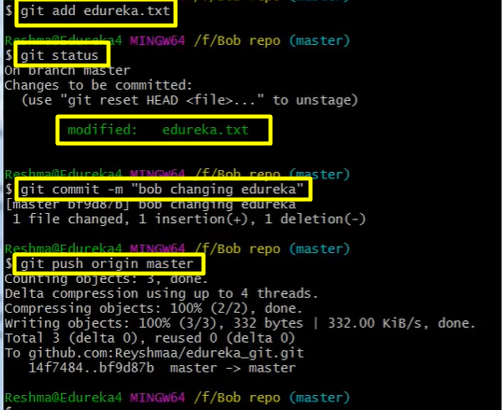
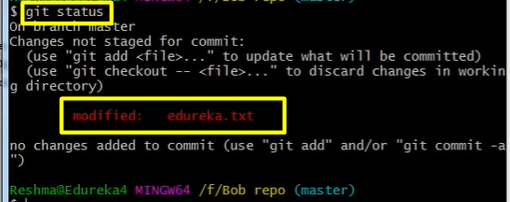
-> Create repository in github and add one text file.

-> Create two folders in local system for two users. Eg. Bob and Alice

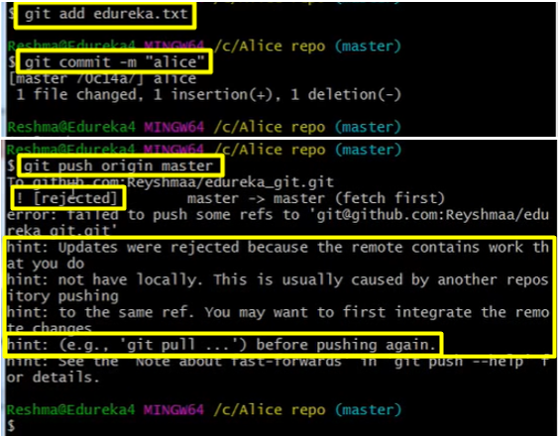
-> Pull the repository to Bob and Alice folders



-> Update the file from Bob folder and push the changes to the GitHub

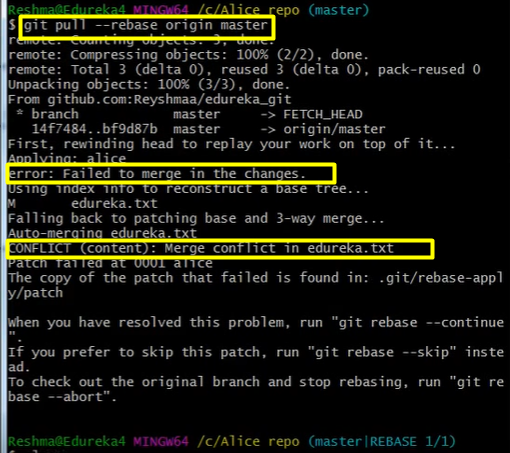


-> Update the same file in the Alice folder and try to push the changes to GitHub (remote repo), we will get the conflict

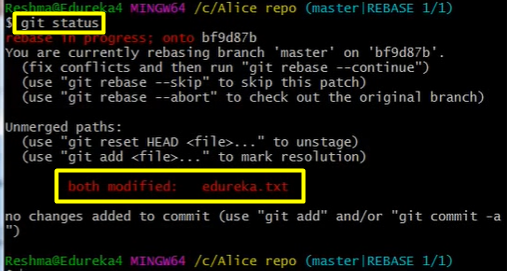


-> To resolve this conflict, first we must pull the changes in remote repo then push it again. But instead of pulling directly, we use **Rebase.**

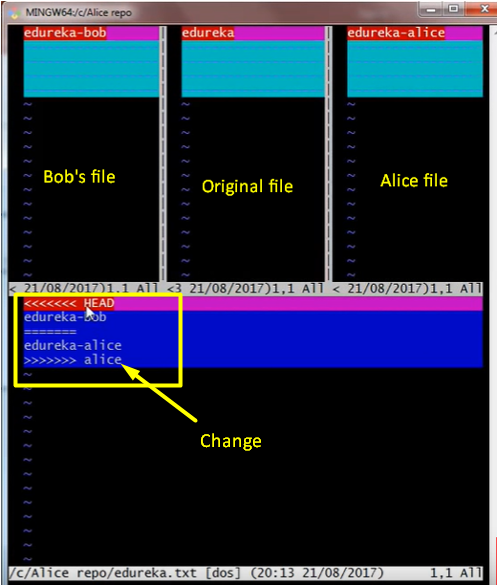
-> Rebase adds the new commits on the top of the existing workflow.



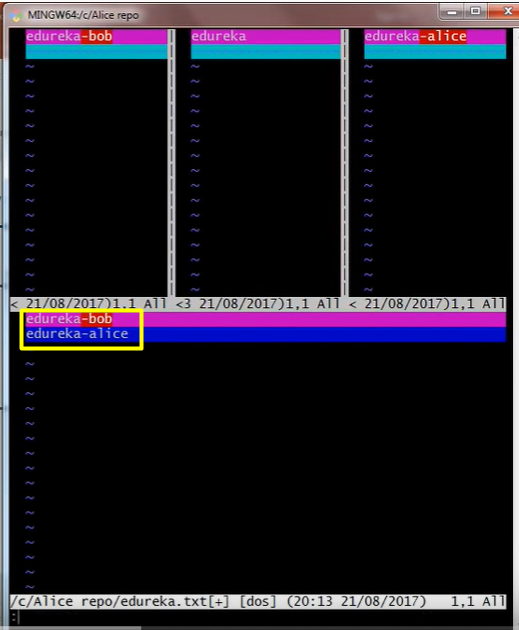
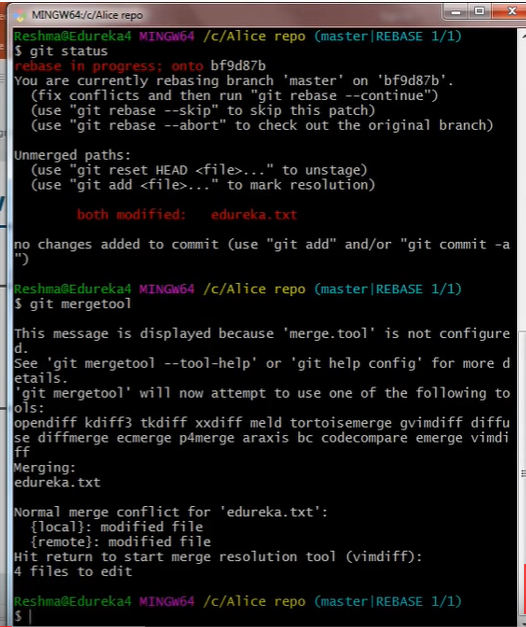
-> If you check the status where the conflict file is.

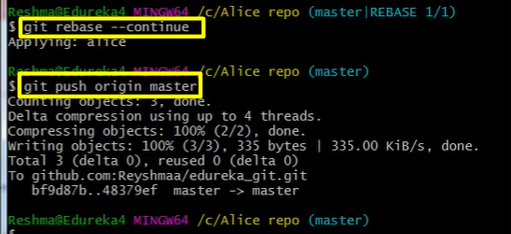


-> In order to solve this conflict, we use **git mergetool.** Type: **git mergetool** => enter



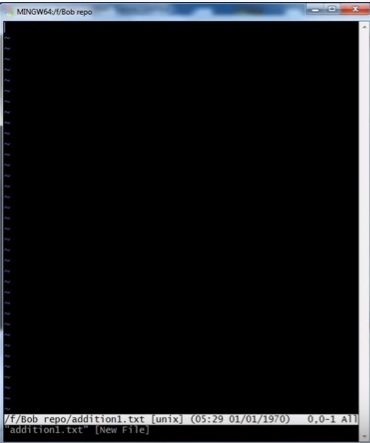
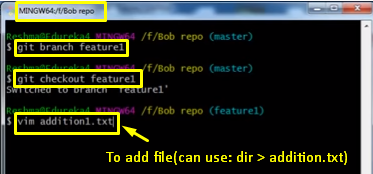
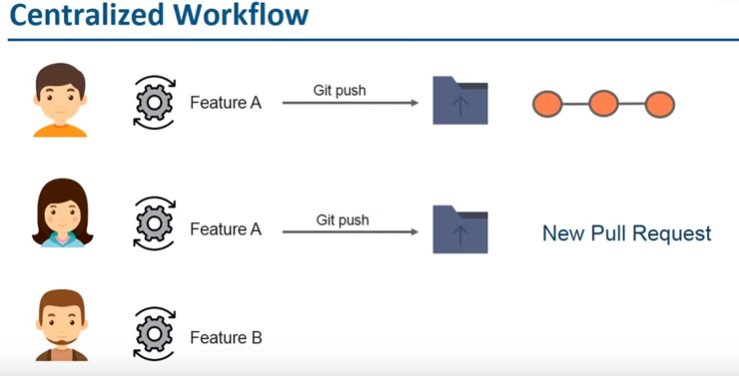
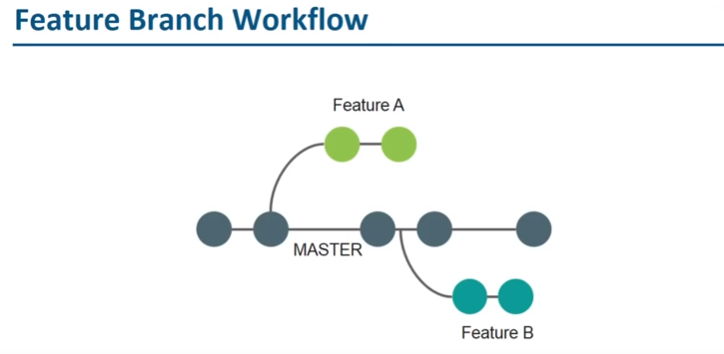
-> Modify the changes (press ‘i’ to edit and to save and exit press “:wq” or “:wq!”)

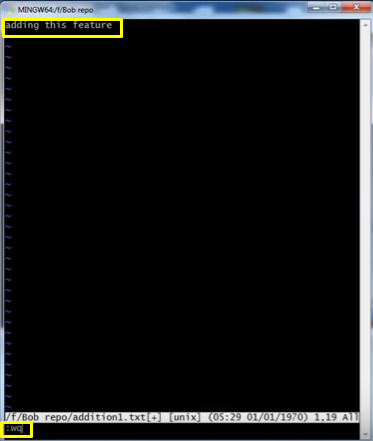


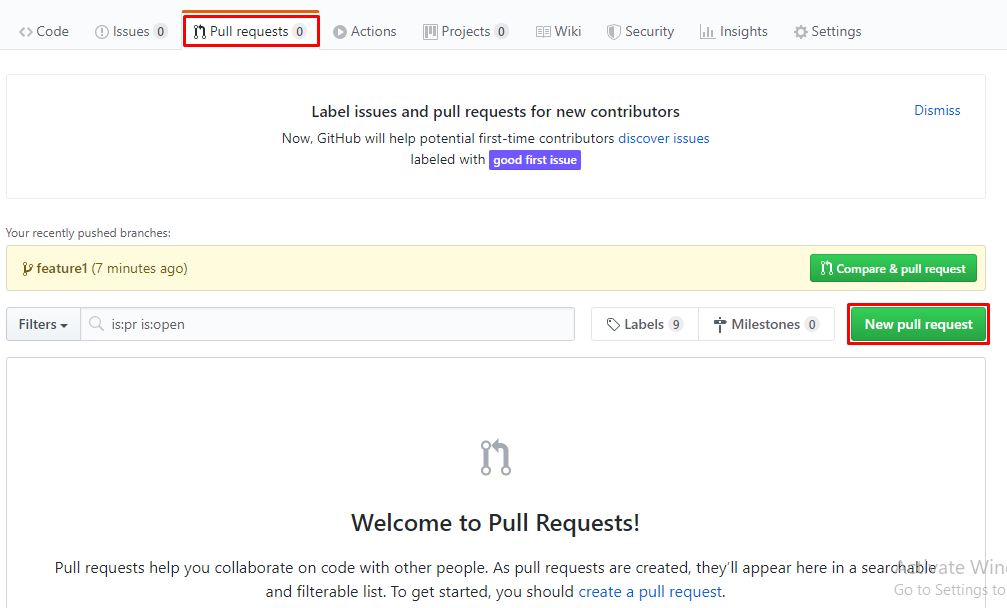
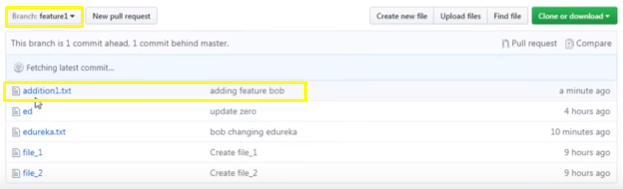
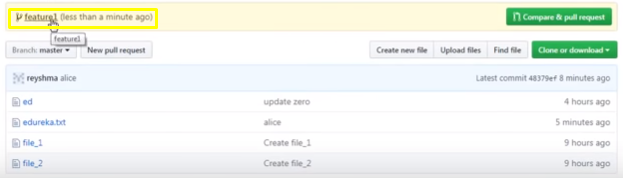
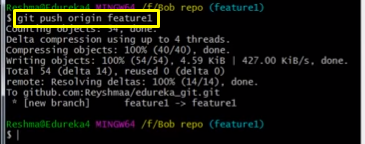
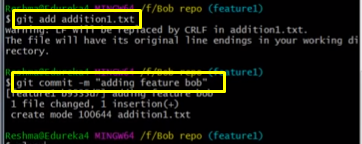
-> $ **git rebase** --continue => to continue rebase

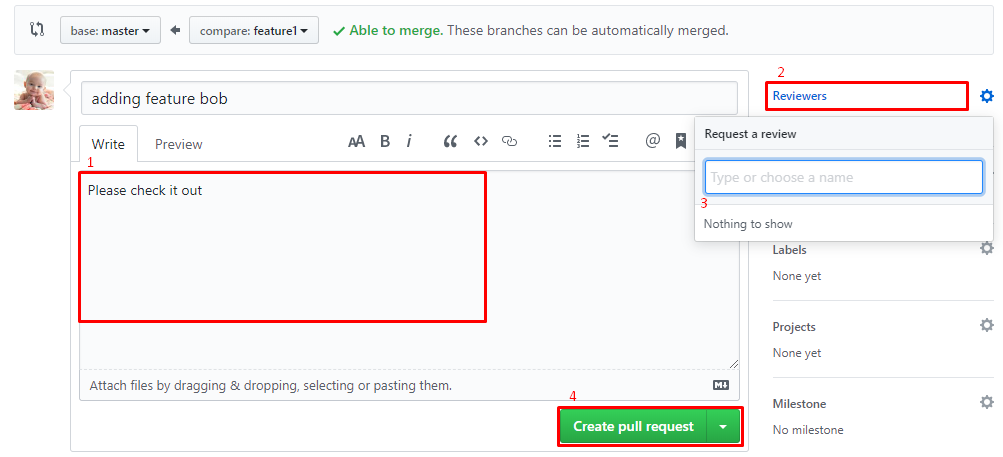
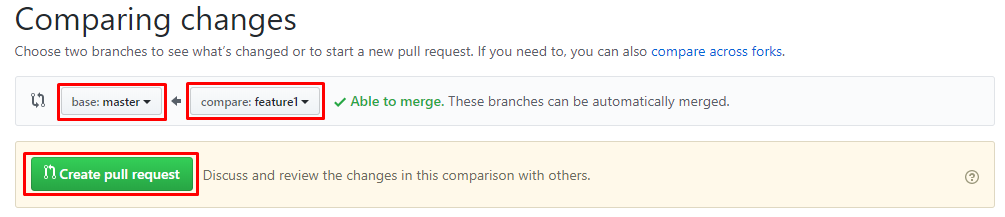
-> $ **git rebase** --abort => to exit/abort from rebase

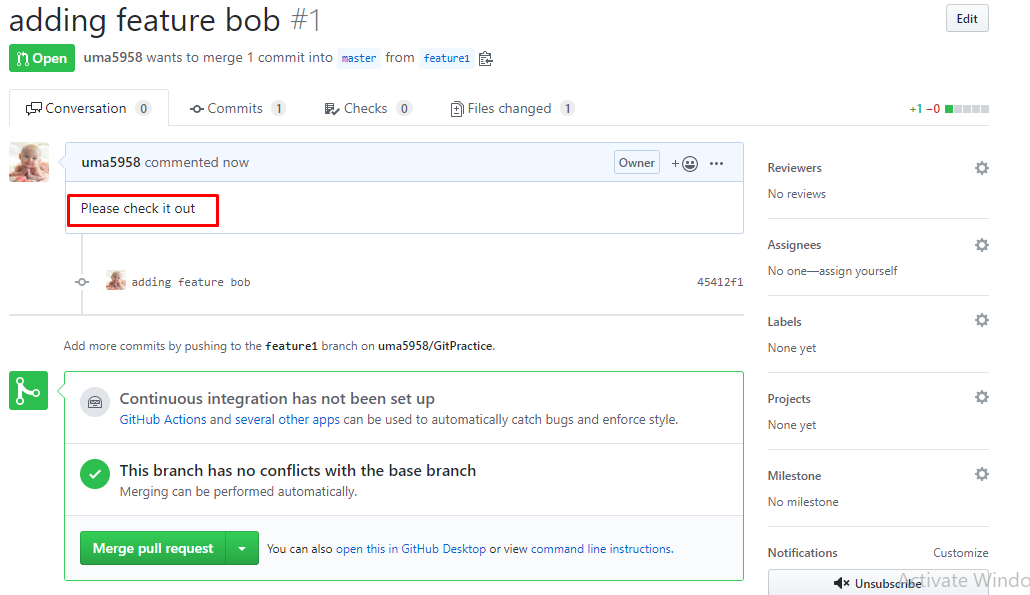


-> Press ‘**i**’ to enter text -> press ‘**Esc**’ to come out of editing -> Press “**:wq**” or “**:wq!**” to save and exit









**Git Flow**

