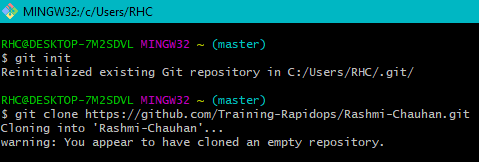
**Perform the following tasks in your repository, attach screenshots for each command and its output, along with the commands inside a doc file for this exercise which should be uploaded in your git repository.**

1. Configure your user name and email.

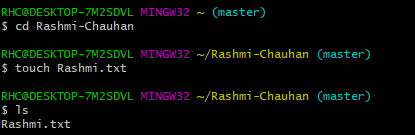
* Username: Rchauhan13
* Email: [rashmi.chauhan@rapidops.com](mailto:rashmi.chauhan@rapidops.com)

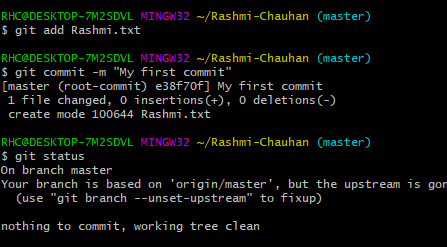
1. Clone repo of your name from GitHub to the local system.

* Git clone <https://github.com/Training-Rapidops/Rashmi-Chauhan.git>

1. Create a file inside the repo, and make your first commit "My First Commit".

* cd Rashmi-Chauhan
* touch Rashmi.txt
* add Rashmi.txt
* git commit –m “My first Commit”





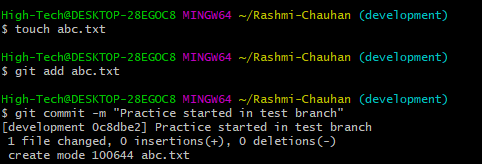
1. Create and switch to the branch 'test/development' (create from the master branch and it should be from origin).

* git branch development
* git checkout development



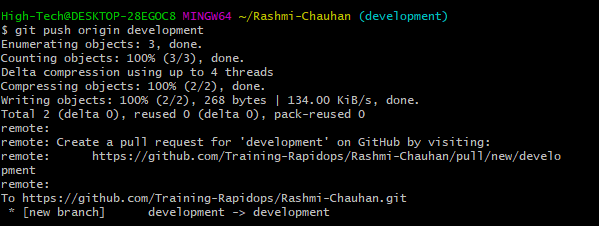
1. Add a file in this branch and commit your changes with the message "Practice started in test branch".

* touch abc.txt
* git add acb.txt
* git commit –m “Practice started in test branch”

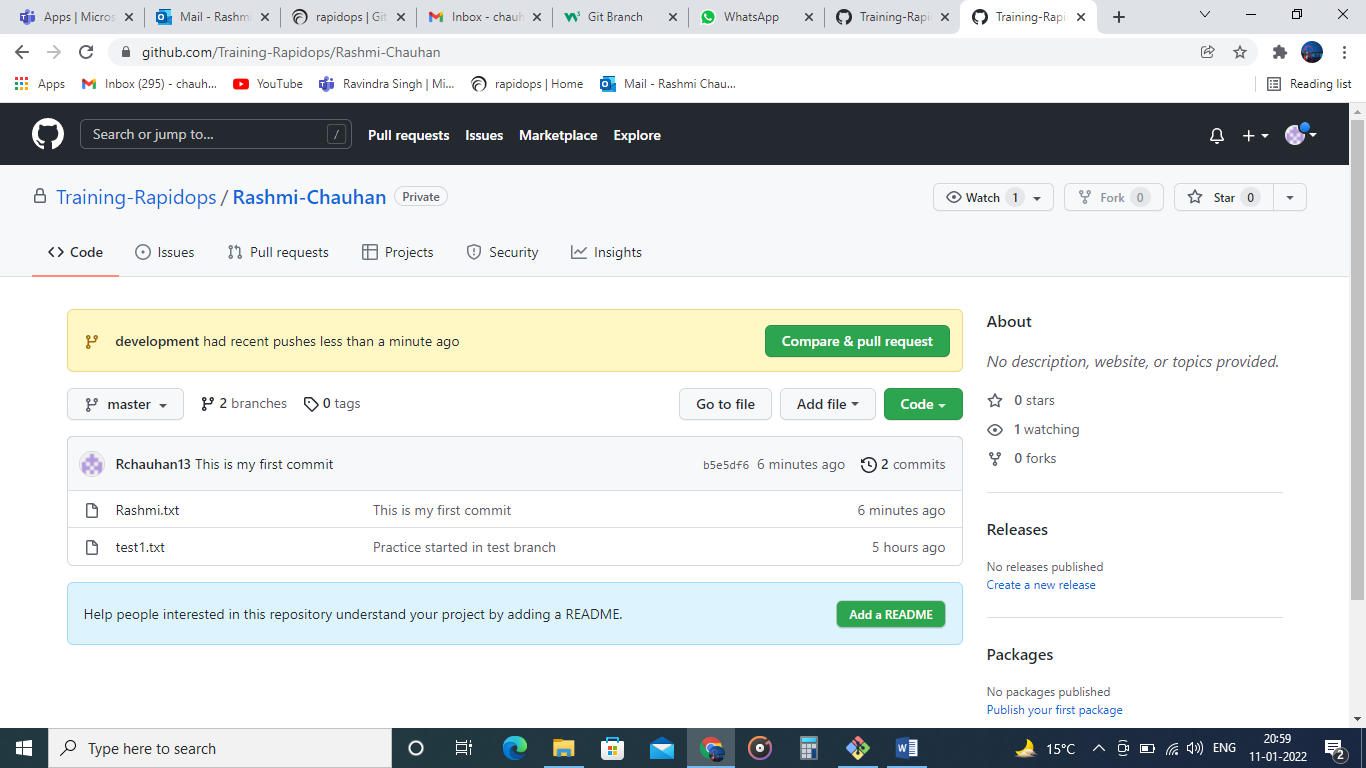


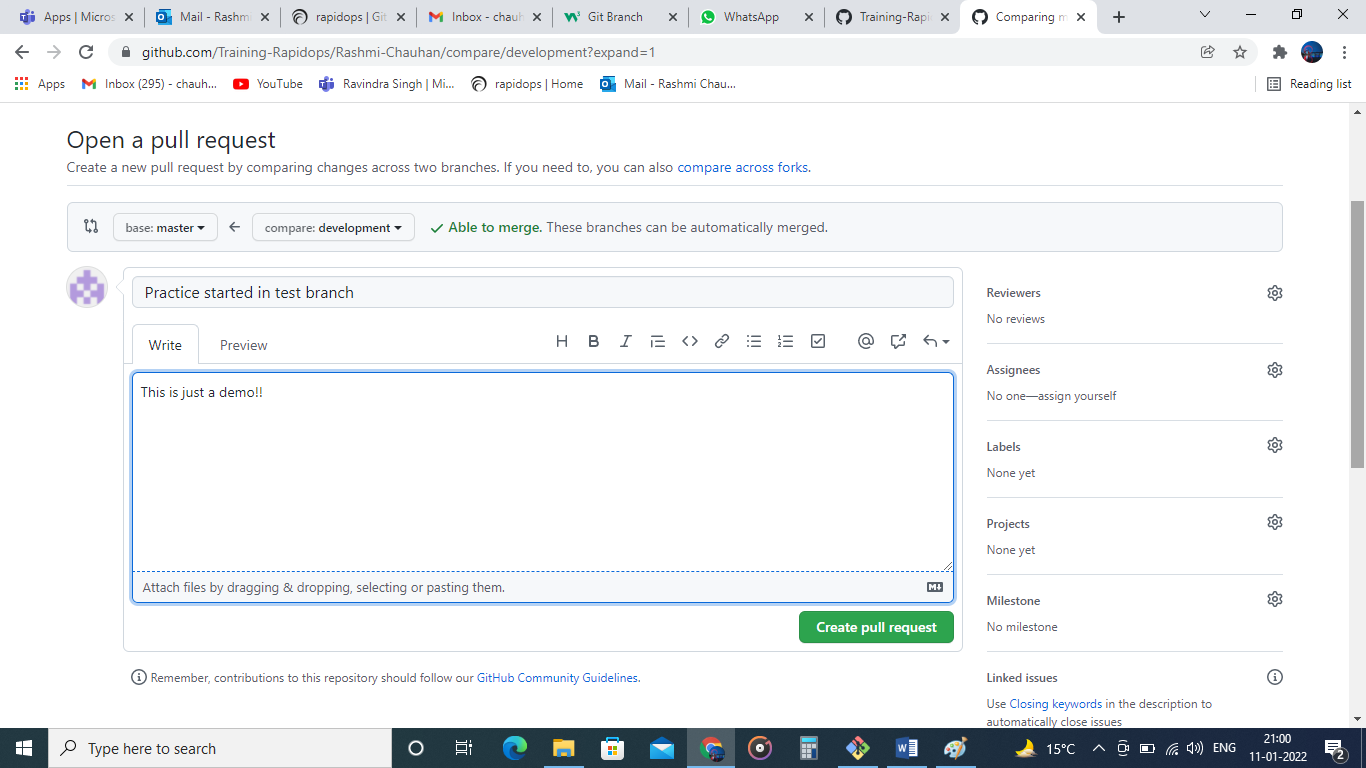
1. Now push your changes and this branch to the remote.

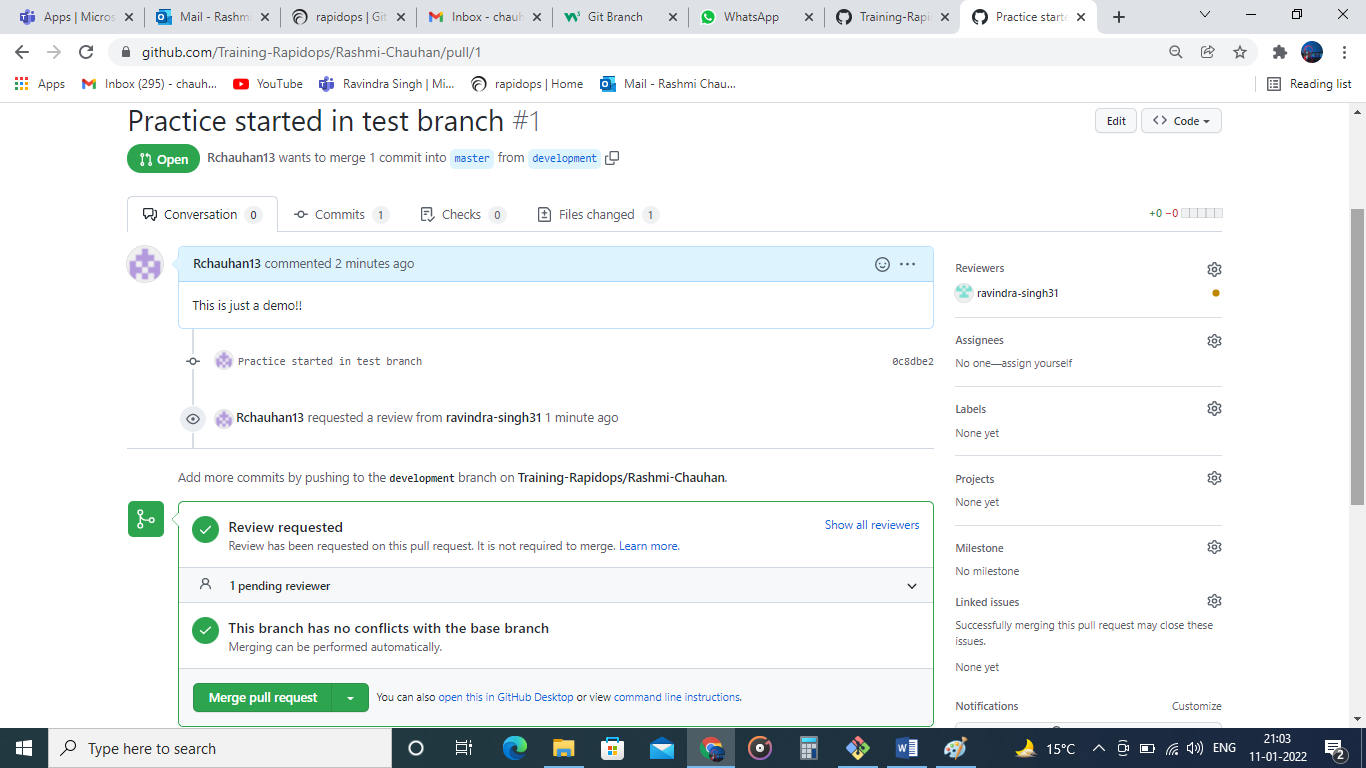
* git push origin development



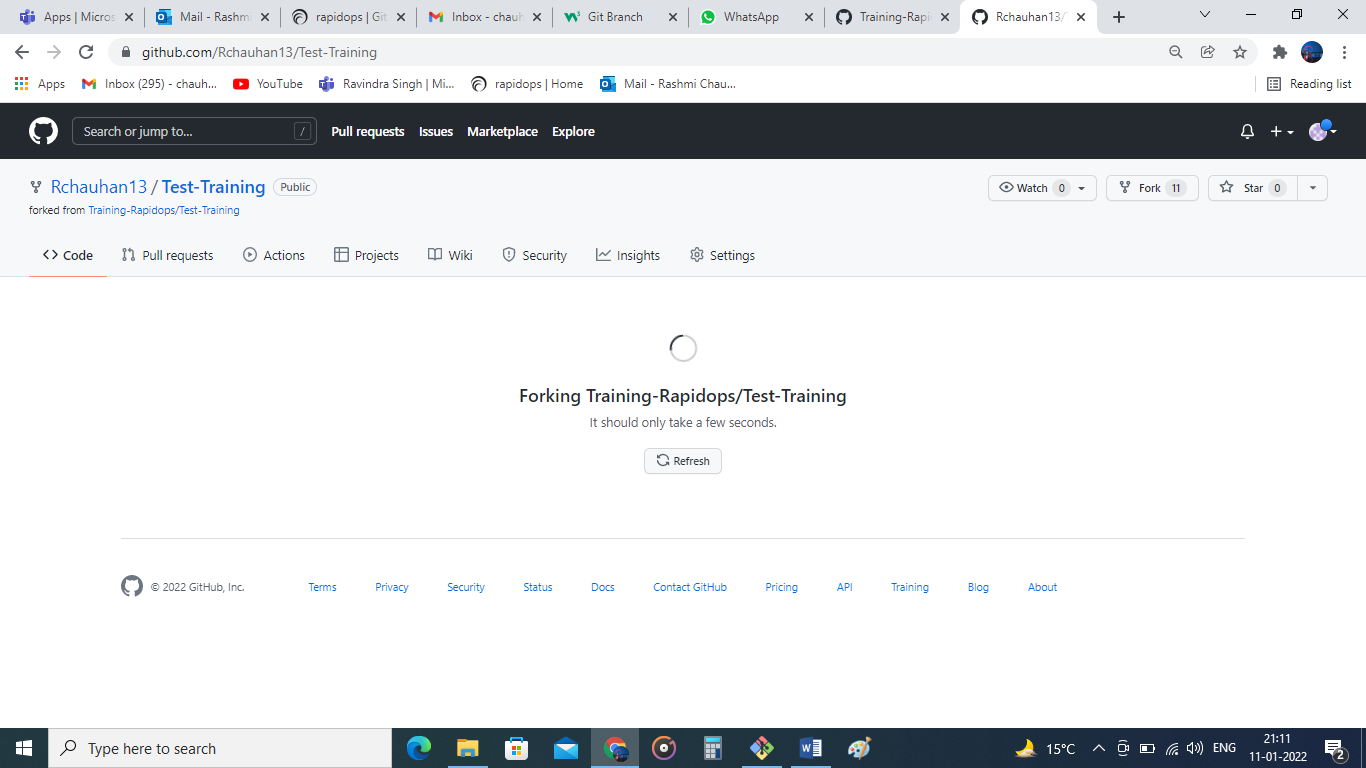
1. Go to your GitHub repository and create a pull request to merge this branch in master. Also, add Ravindra & your mentor as reviewers.

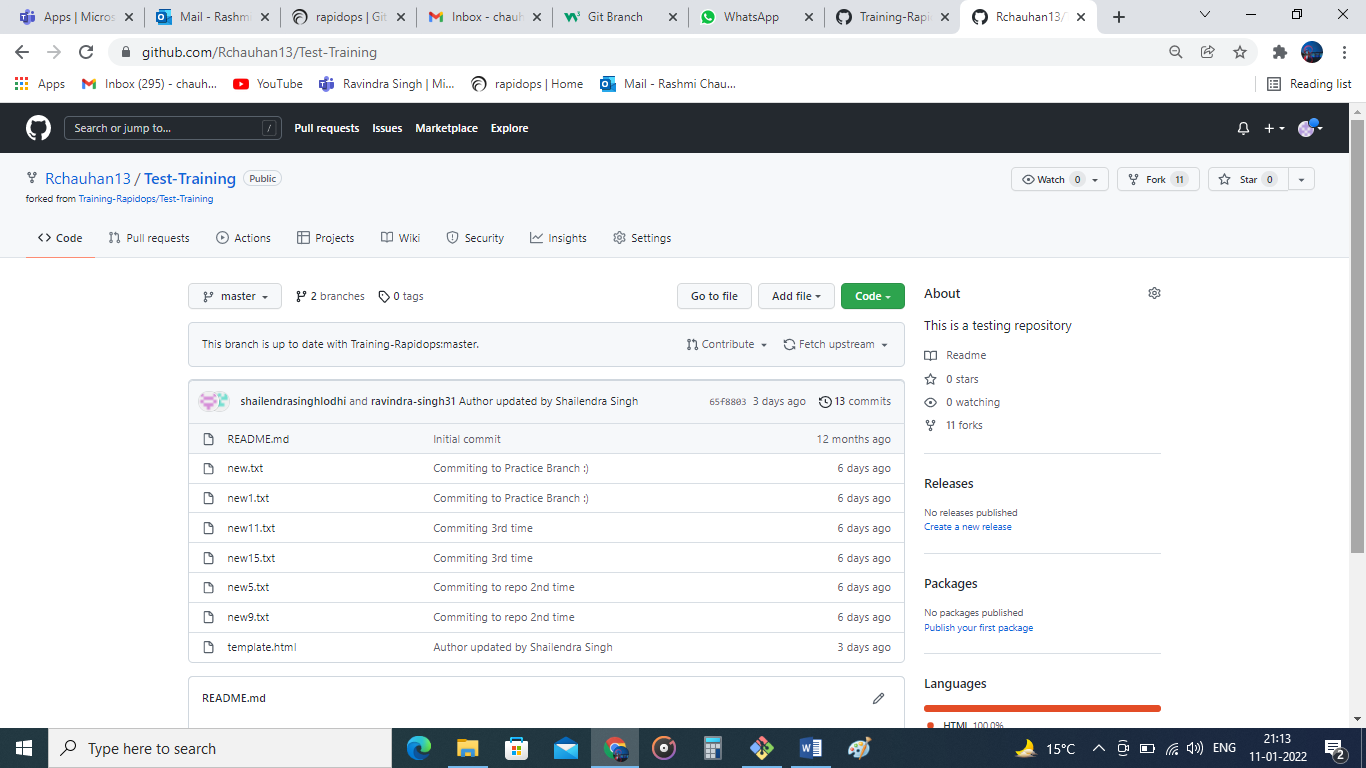






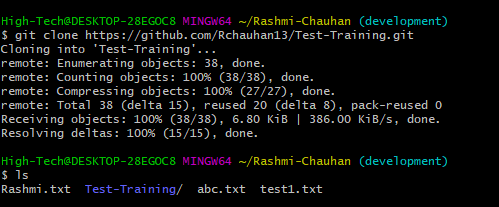
1. Fork a public repository [**Test-Training**](https://github.com/Training-Rapidops/Test-Training)

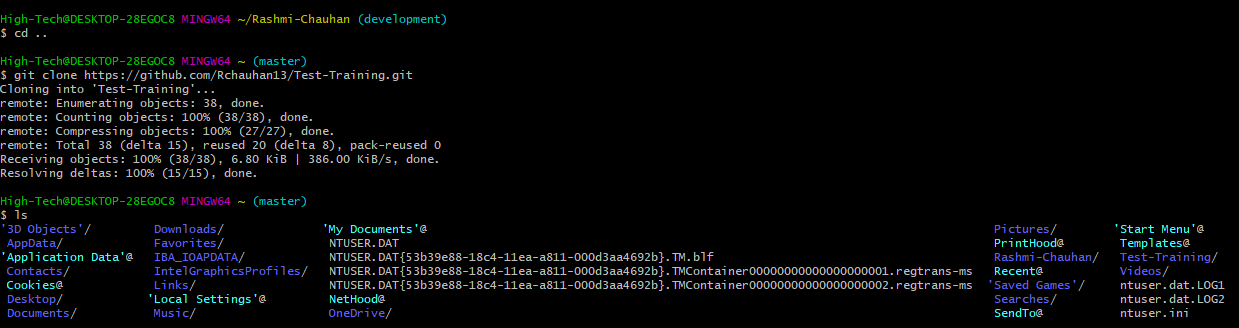


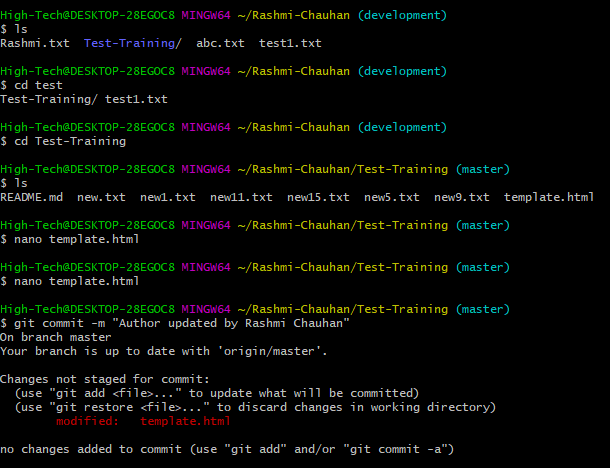


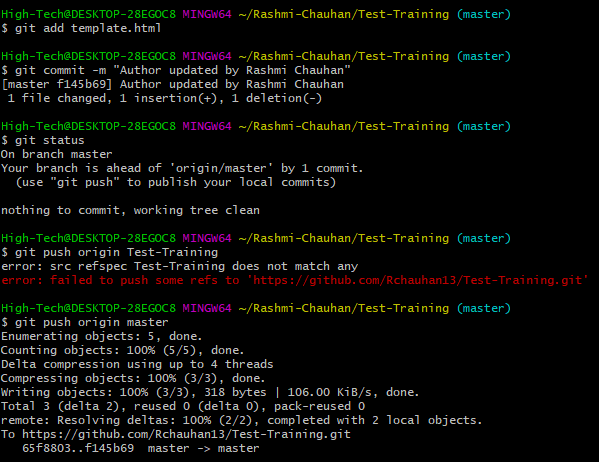
1. Make a change in the template.html file by adding 'author: {your-name}' and add commit 'Author updated by {your-name}'.

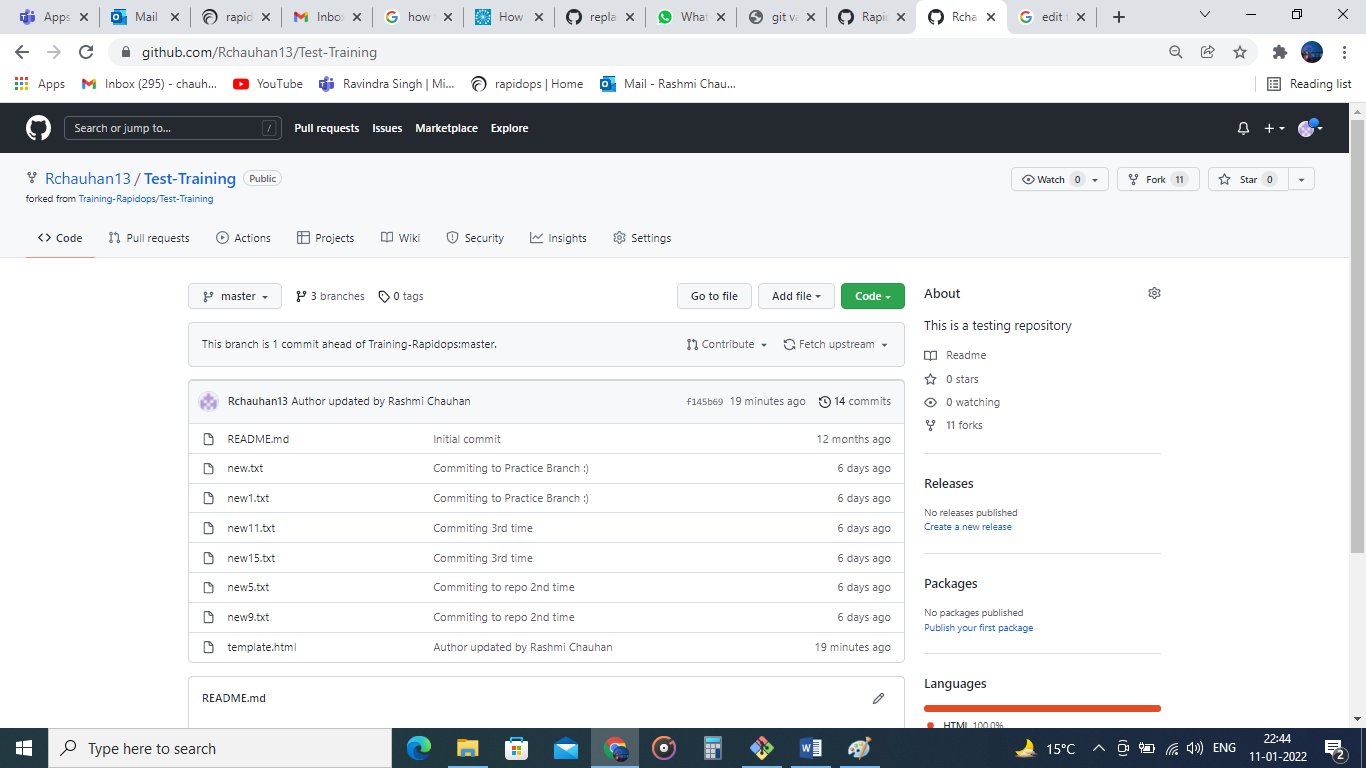
* Git clone https://github.com/Rchauhan13/Test-Training.git



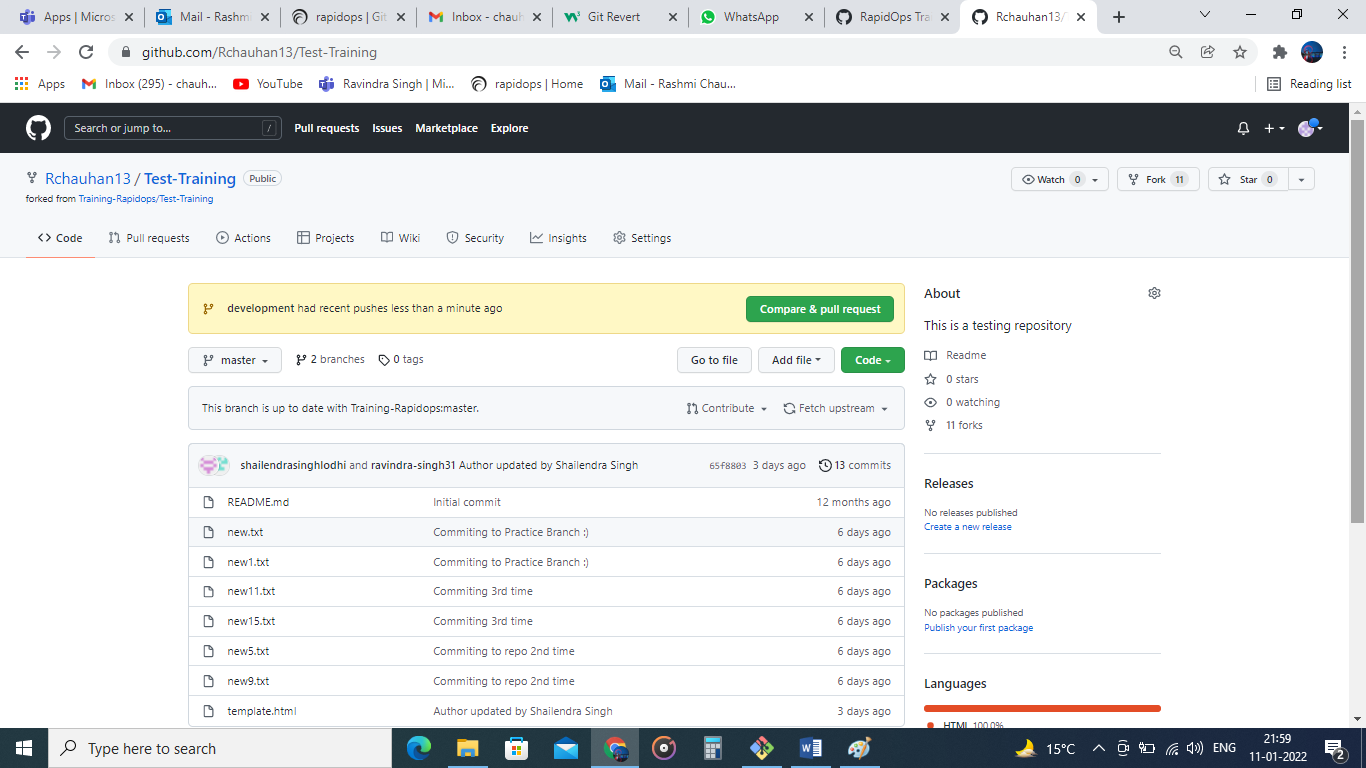


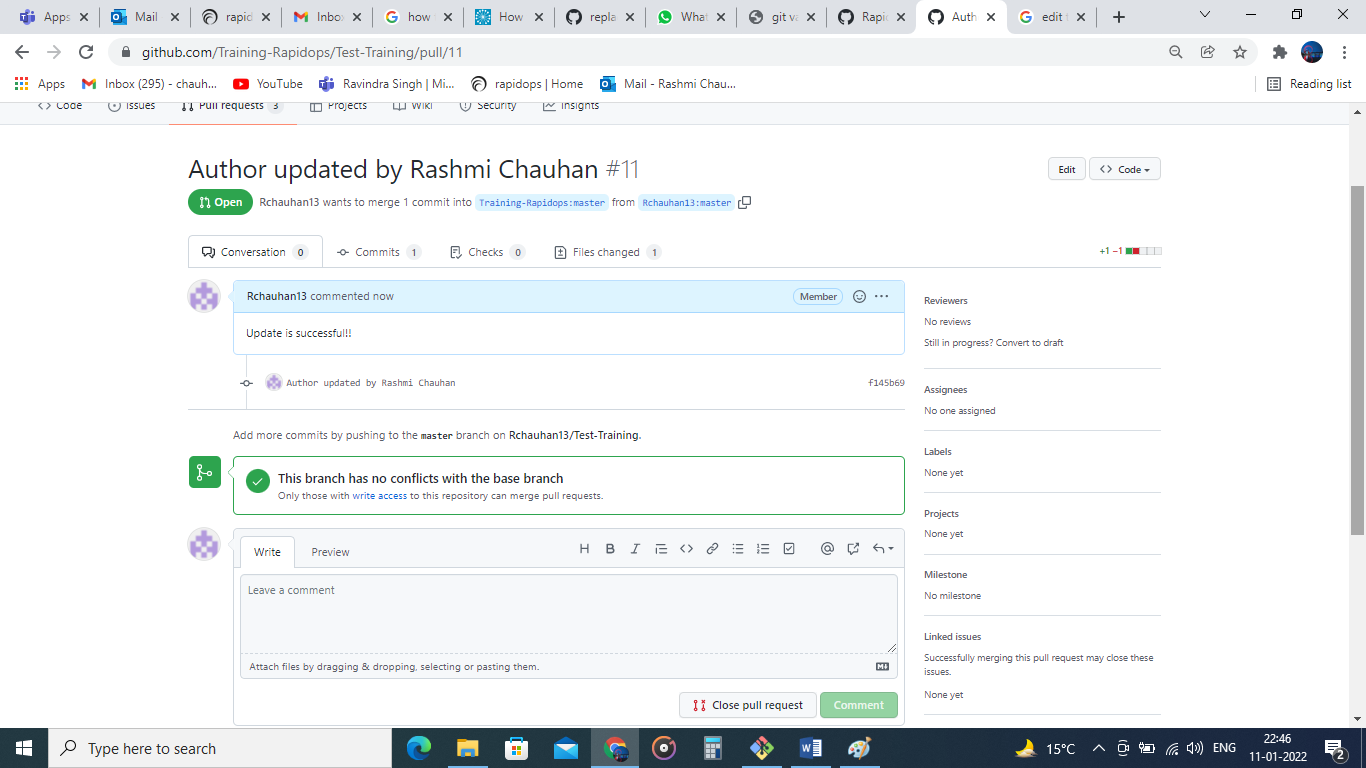






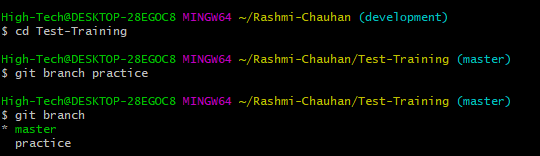
1. Give PR to merge your forked repo in the **Test-Training** repo.



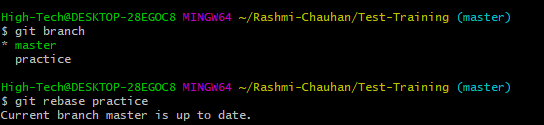
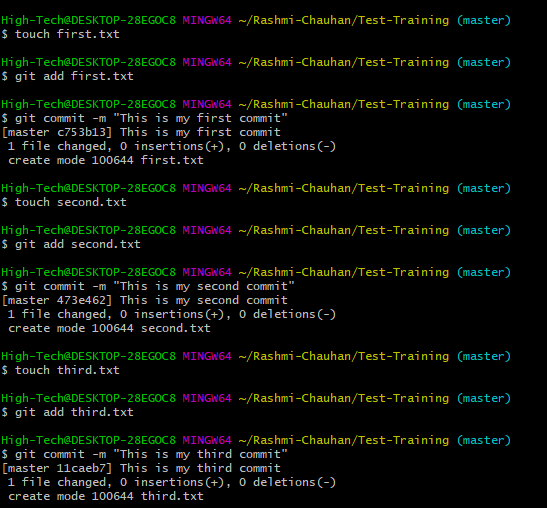


1. Come back to your repo, where you create a branch 'practice' and add 3 commits then rebase it with the master.

* Cd Test-Training
* Git branch practice

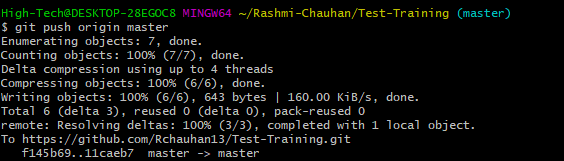


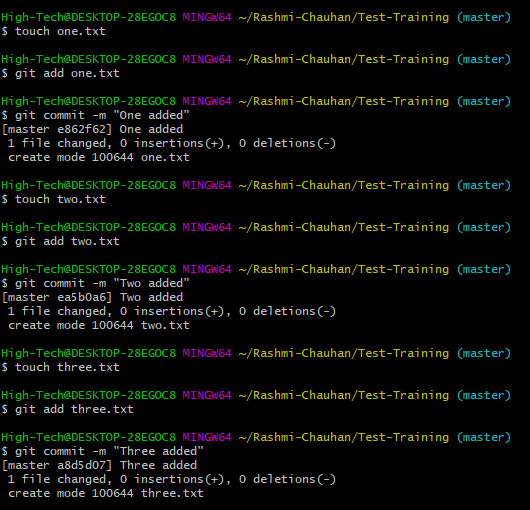
* touch first.txt
* git add first.txt
* git commit –m “This is my first commit”
* touch second.txt
* git add second.txt
* git commit –m “This is my second commit”
* touch third.txt
* git add third.txt
* git commit –m “This is my third commit”
* git rebase practice

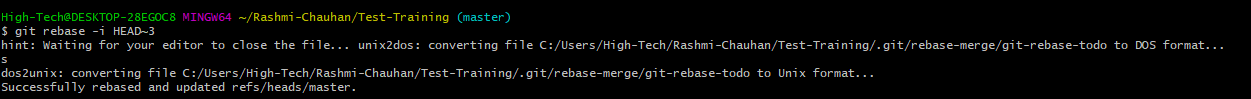


1. Push your changes to remote, then Add 3 commits again and squash them into the first commit by keeping the message "Rebase squash done".

* git push origin master
* touch one.txt
* git add one.txt
* git commit –m “one added”
* touch two.txt
* git add two.txt
* git commit –m “two added”
* touch three.txt
* git add three.txt
* git commit –m “three added”
* git rebase –i HEAD~3

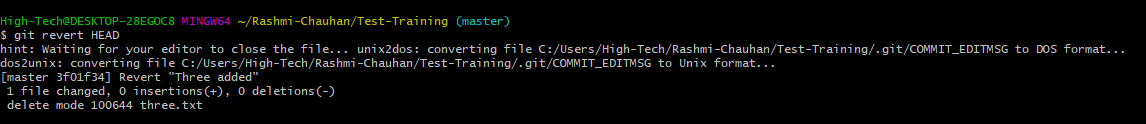






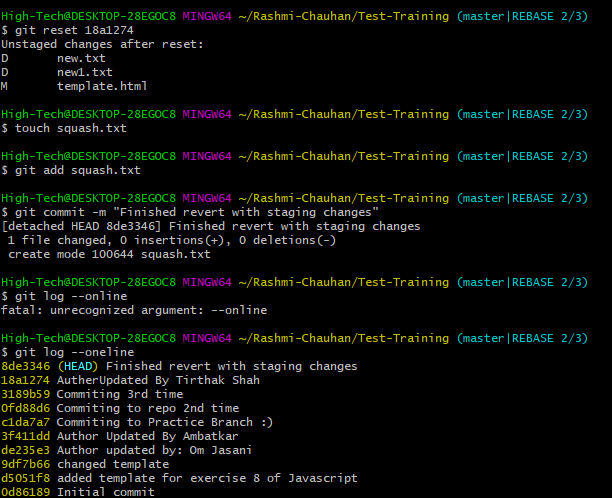
1. Now revert these changes but note that the changes must be retained in the commit history. (use default revert commit message)

* git revert HEAD



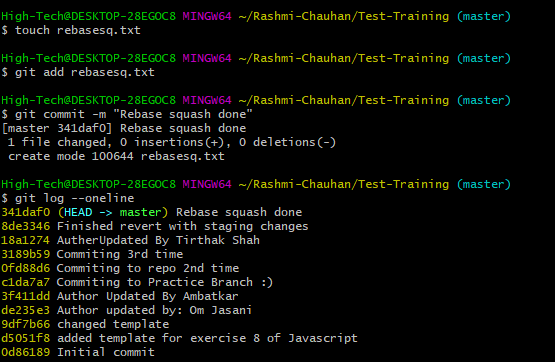
1. Reset your last commit without losing the changes and then commit with the message "Finished revert with staging changes".

* git reset 18a1274
* touch squash.txt
* git add squash.txt
* git commit –m “Finished revert with staging changes”
* git log --oneline



1. Create one commit then perform a hard reset such that you're back to the commit with the message "Rebase squash done".

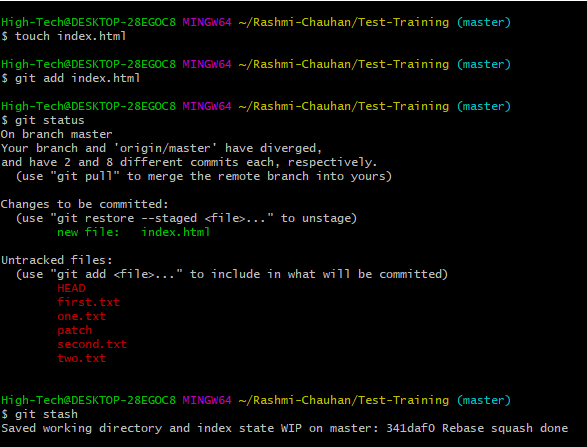
* touch rebasesq.txt
* git add rebasesq.txt
* git commit –m “Rebase squash done”
* git log - -oneline
* git reset - -hard 341daf0





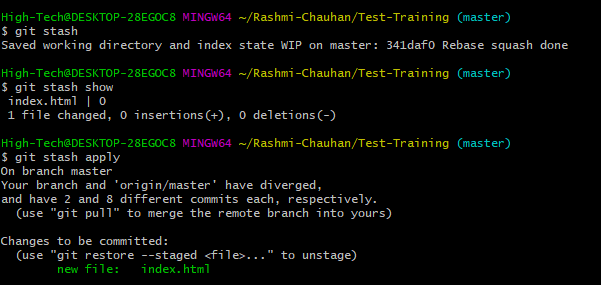
1. Create a file index.html, and add it to the staging index then stash it.

* touch index.html
* git add index.html
* git status
* git stash



1. Check the list of stash, what changes are there in the stash, then bring your changes from stash.

* git stash
* git stash show
* git stash apply



1. Commit with a message "Revert, Reset & Stash done".

* git add index.html
* git commit –m “Revert, Reset & Stash done”

