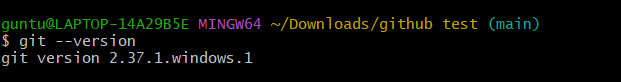
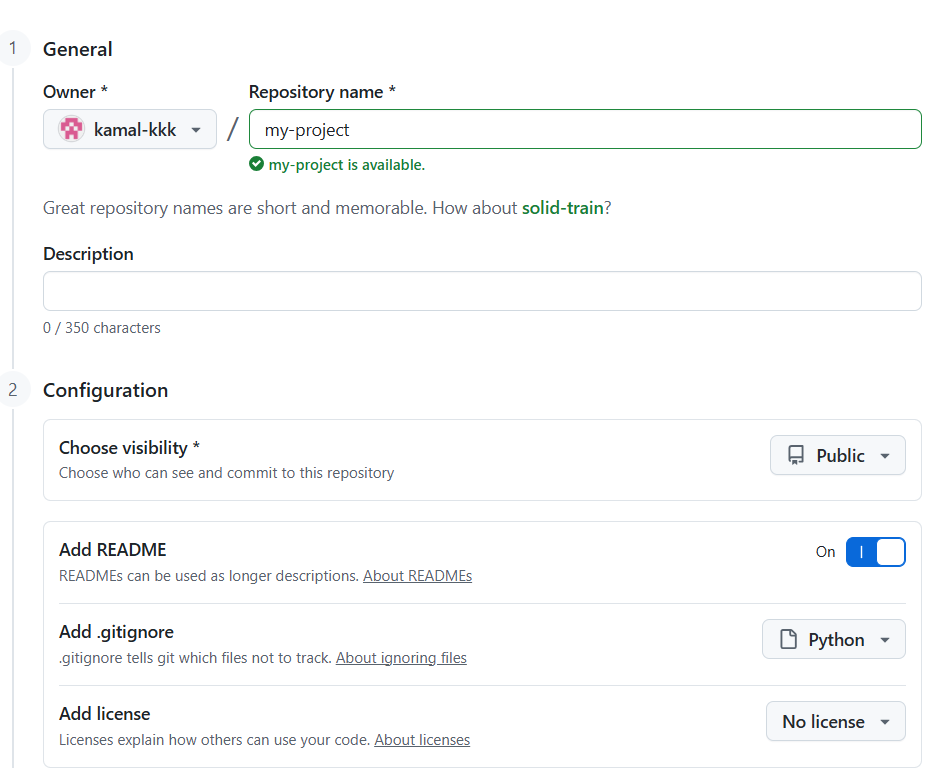
**GIT & GITHUB**

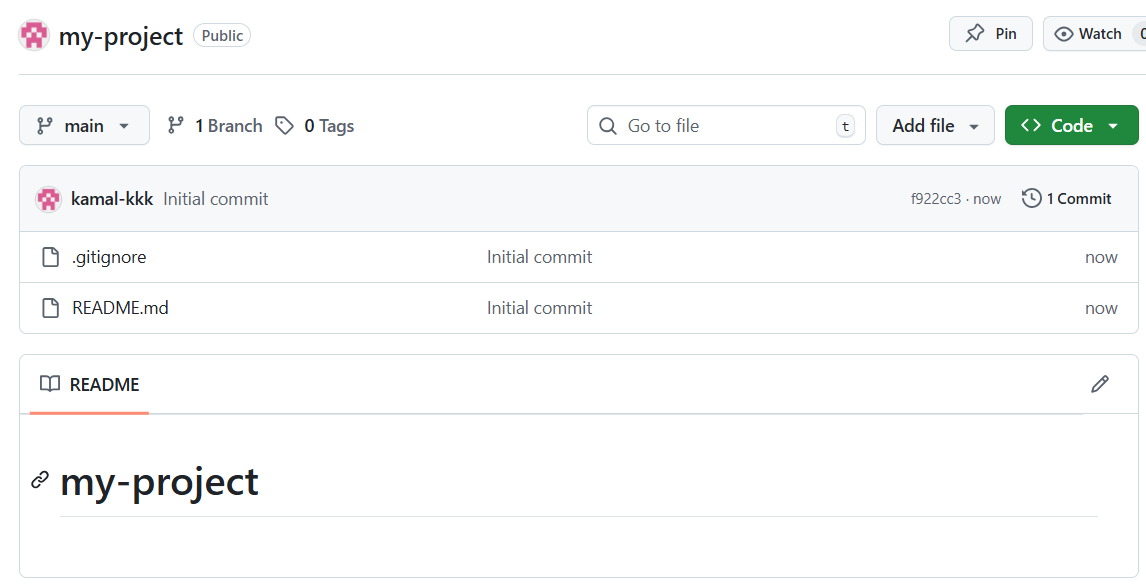
1 install git



2Create a repo in github with README.md and .ignore file.

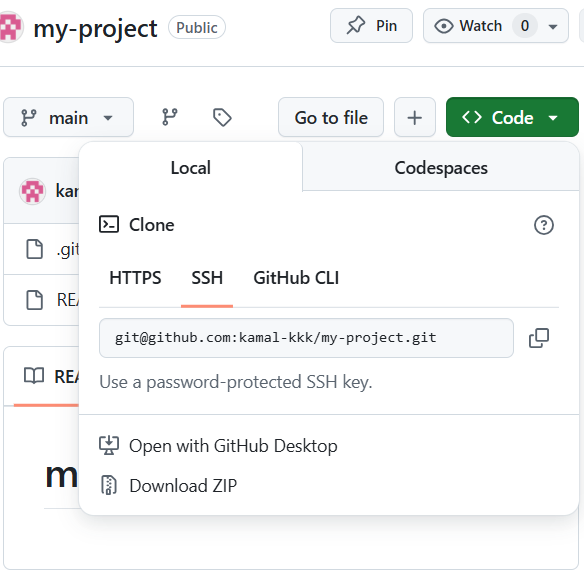
* Enter project name
* Choose visibility
* Add README
* Add. Gitignore , language python

****

****

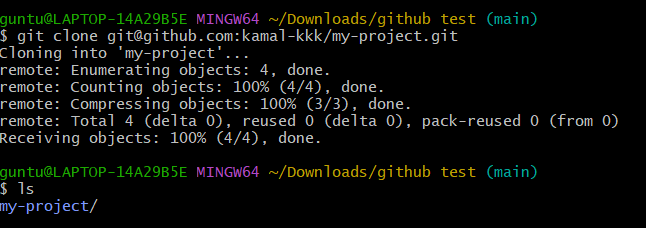
3.Clone the created repo to local**.**

Copy the url of the file



Open terminal and use command

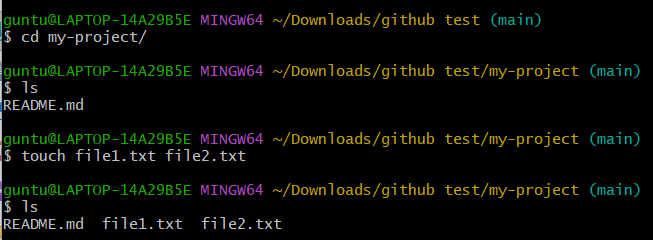
Git clone [git@github.com:kamal-kkk/my-project.git](mailto:git@github.com:kamal-kkk/my-project.git)



4. Create two files in local repo.

\* go to repo

\* enter command – touch file1 file2



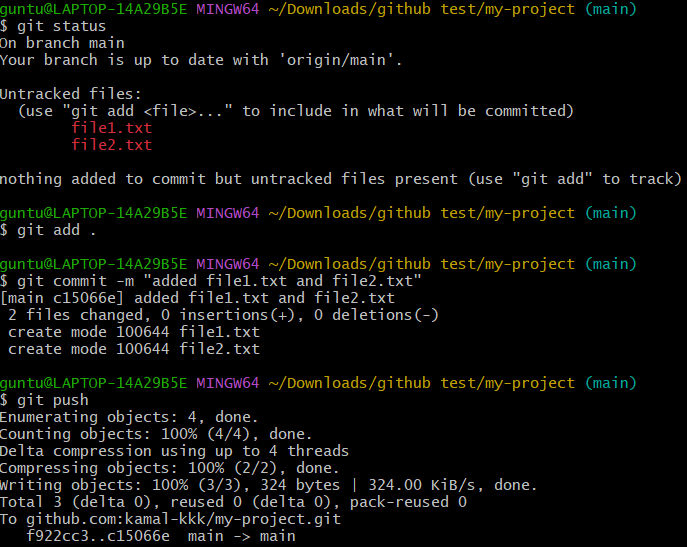
5. Commit two files and push to central Repository.

\*list the files

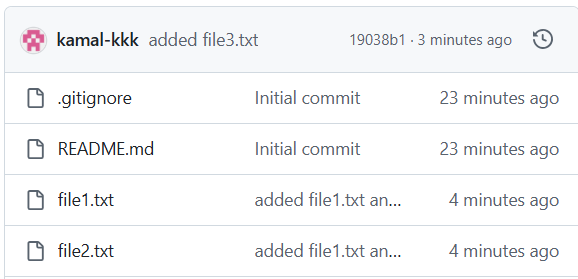
\*git add .

\*commit the files – git commit -m ‘added file1.txt,file2.txt”

\*and then push to central repo \_ git push



\*refresh github



6. Create a branch in local and create a sample file and push to central.

\*create e new branch using command git branch test

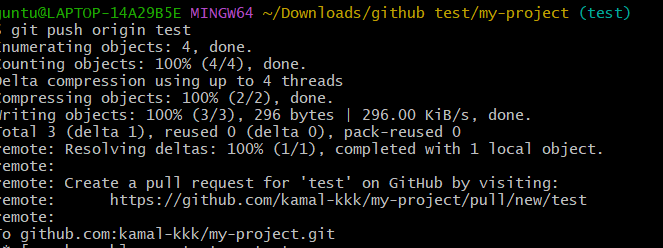
\*switch to test branch = git switch test

\*enter file = echo “testing file” > sample.txt

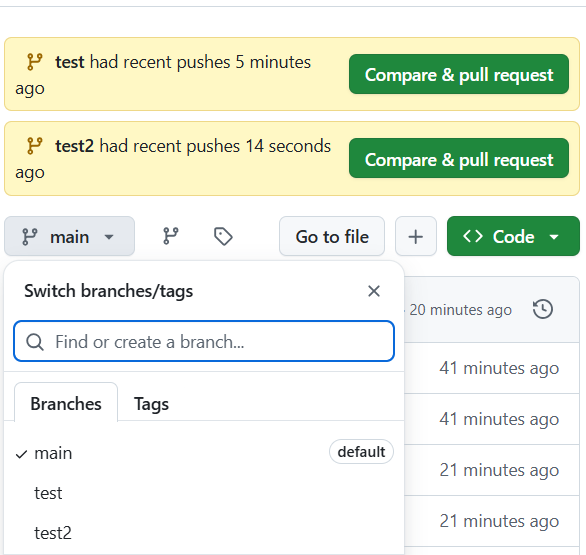
\*git add .

\* commit it = git commit -m “added sample.txt”

\*git push origin test

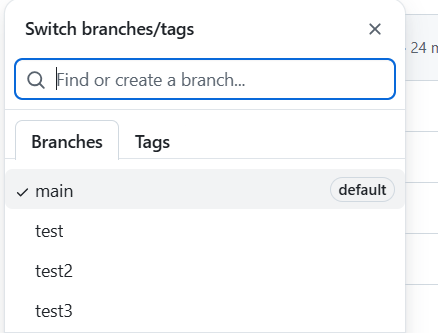


\*check github can see new branch

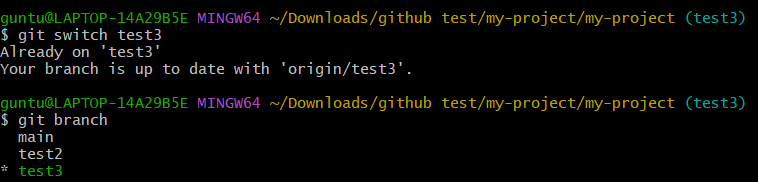


7. Create a branch in github and clone that to local.

\*In project repo click branch and enter add branch with aname test3



\*in gitbash switch to test3 branch = git switch test3



8. Merge the created branch with master in git local.

\*git branch to check running branch

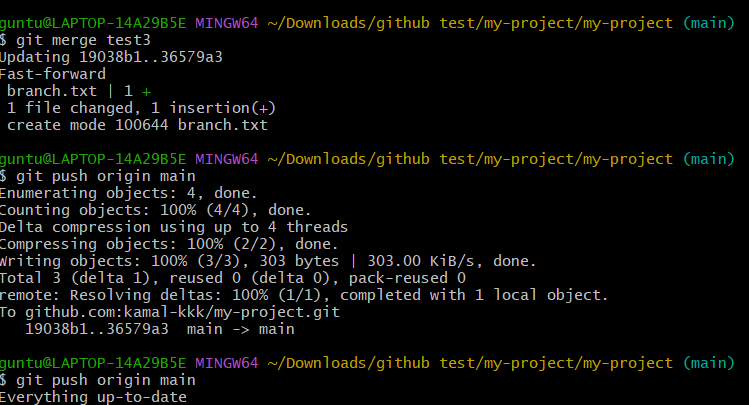
\*git switch main

\*now we are in main branch

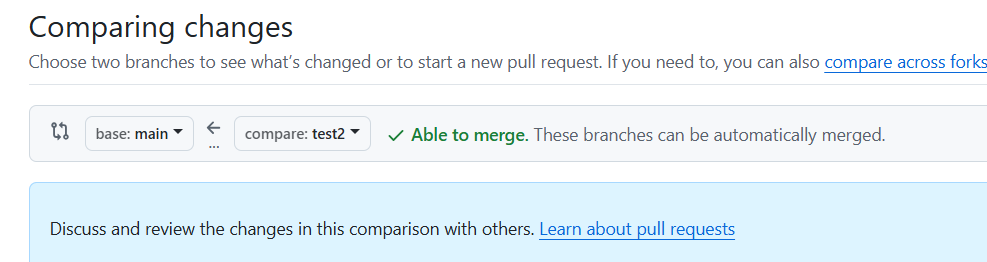
\*git merge test3

\*git push origin main

Merging done



9. Merge the created branch with master in github by sending a pull request.



10. Create a file in local and send that to branch in github.

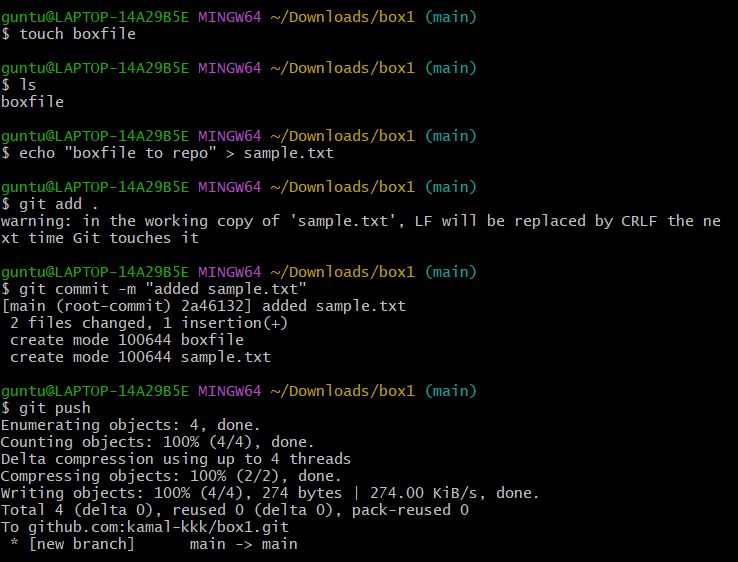
\*create a repo

\*add files as boxfile

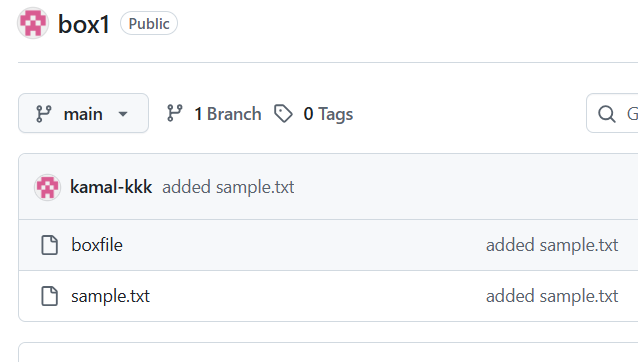
\*git add . to it

\*git commit -m “added sample.txt”

\*git push



\*refresh github

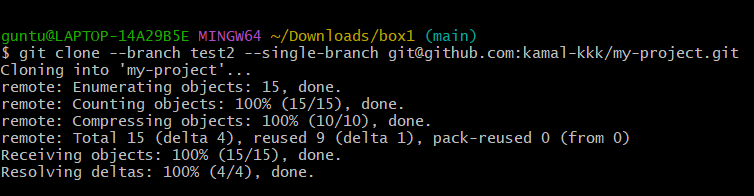


11. Clone only a branch from github to local.

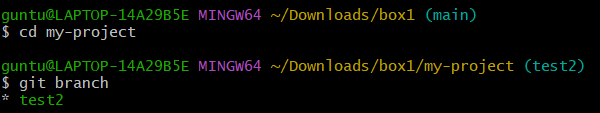
\*copy repo url

\*command= git clone –branch test2 –single-branch [git@github.com:kamal-kkk/my-project.git](mailto:git@github.com:kamal-kkk/my-project.git)

\*



\*verify = cd my-project



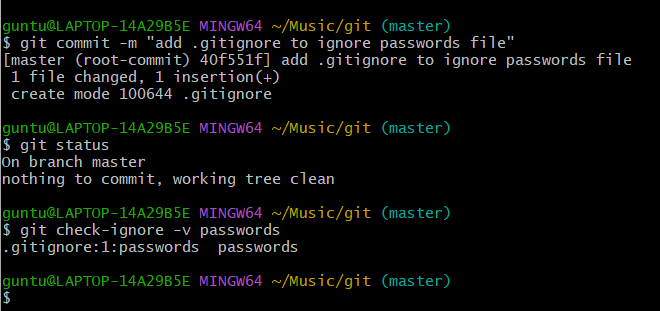
12. Create a file with all passwords and make that untrackable with git.

\*created a file passwords

\* enter passwords in the file using vi editor

\*git init

\*git commit -m “added top gitignore”

\*

\*to check = git check-ignore -v passwords

13.Make a commit and make that commit reset without savings changes.

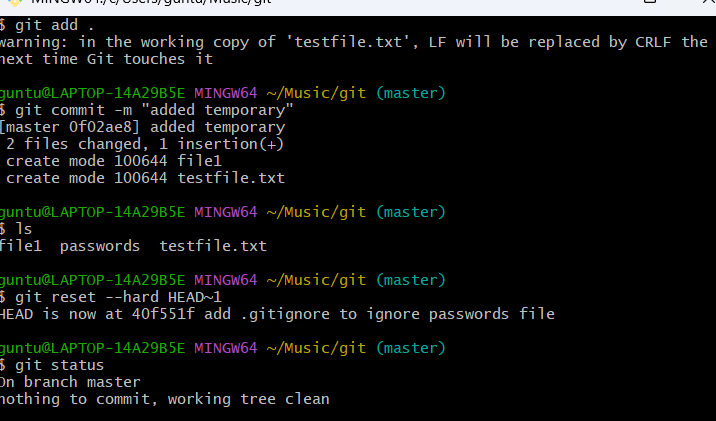
\*create a file with content

\*echo “temporary content” > sample.txt

\*git add .

\*commit it = git commit -m “added file to sample.txt”

\*git reset –hard HEAD~1

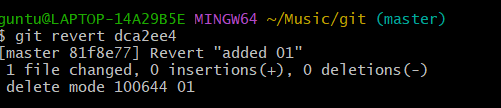


14.Revert a commited commit to the older version.

\*git log = will get all the commit we have done

\*coppy the first 5 digits of commit

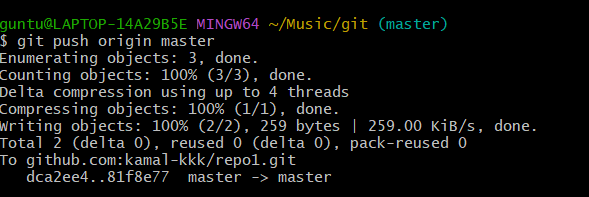
\*now, git revert {paste commit id}



\*editor will open ,just save it and close it

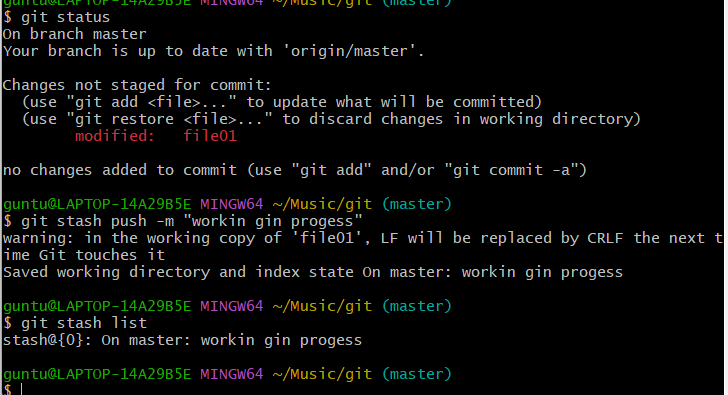
\* now git push origin master

\*file reverted

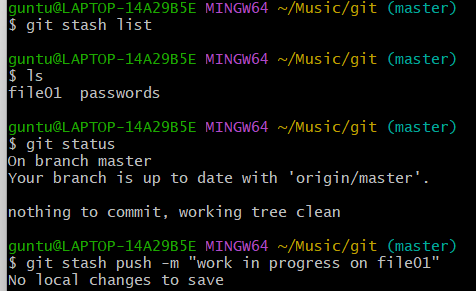


15.Push a file to stash without savings the changes and work on another file.

\*git status

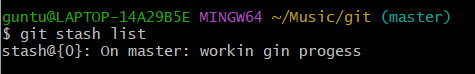


\*now, use command = git stash push -m “workingon file01”

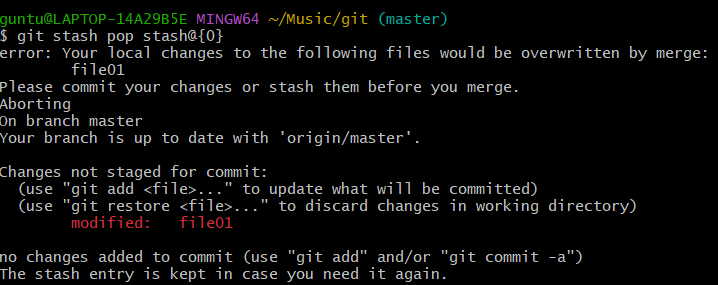


16. Undo the stash file and start working on that again.

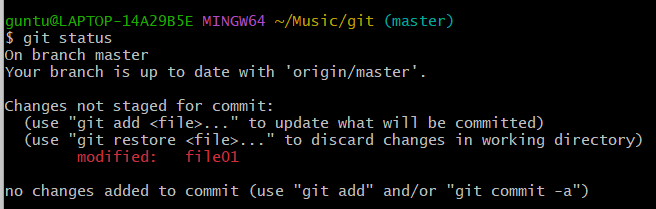
\*git stash list



\*git stash pop stash@{0}



\*git status

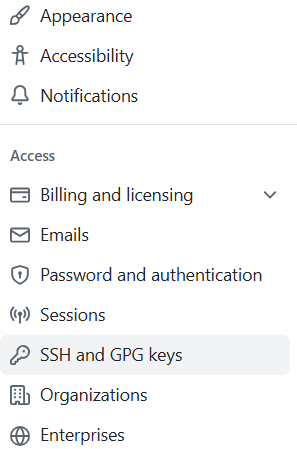


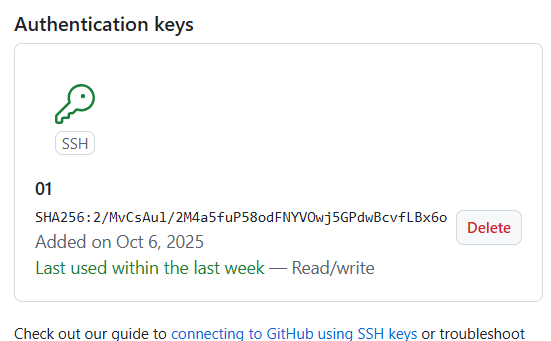
17.Generate a ssh-keygen and configure into github.

\*ssh-keygen = to generate key

\*SHA256:2/MvCsAul/2M4a5fuP58odFNYVOwj5GPdwBcvfLBx6o

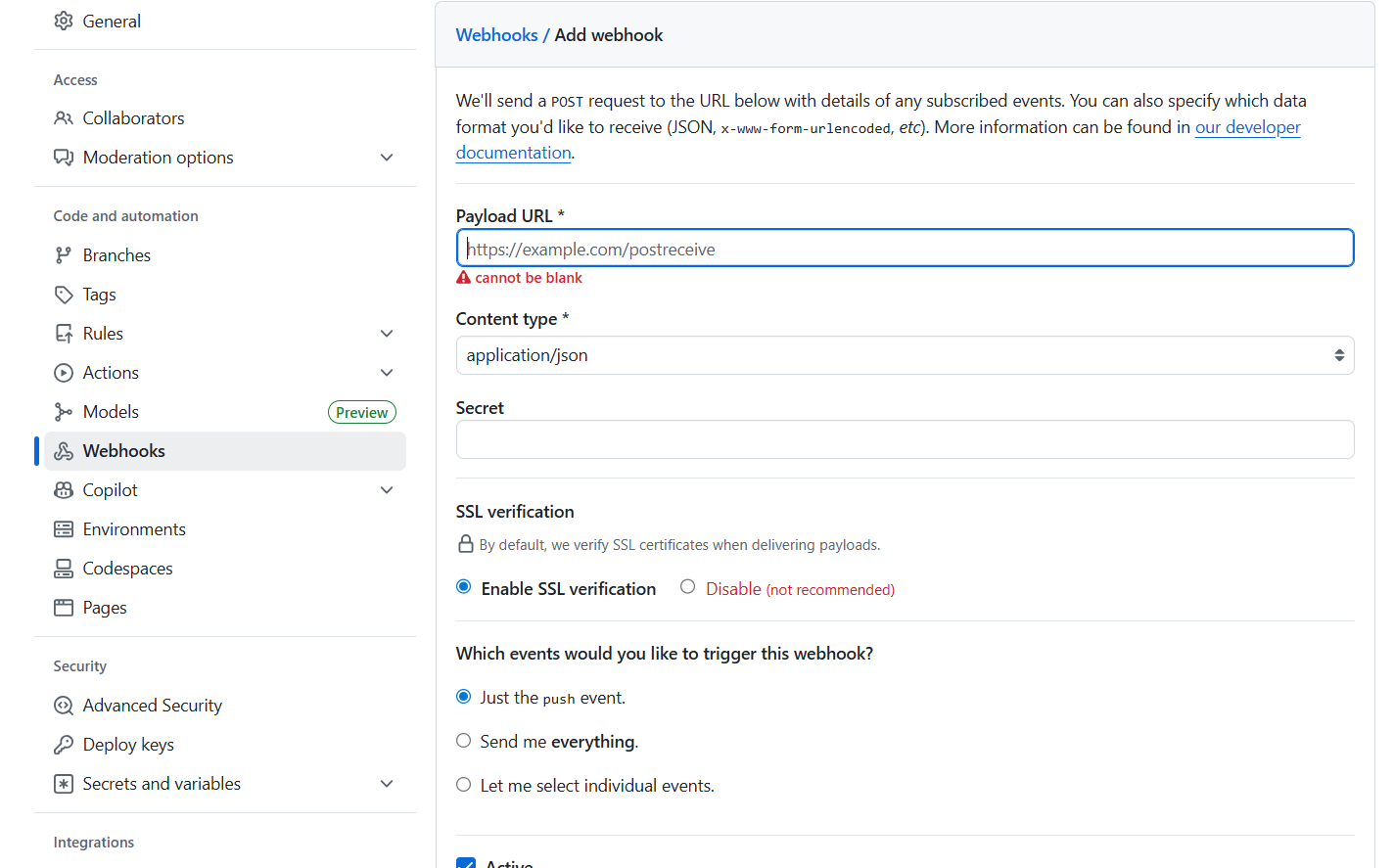
\*paste the generated key in github setting



\*

18. Configure webhooks to github.

\*playload url required



19. Basic understanding of .git file.

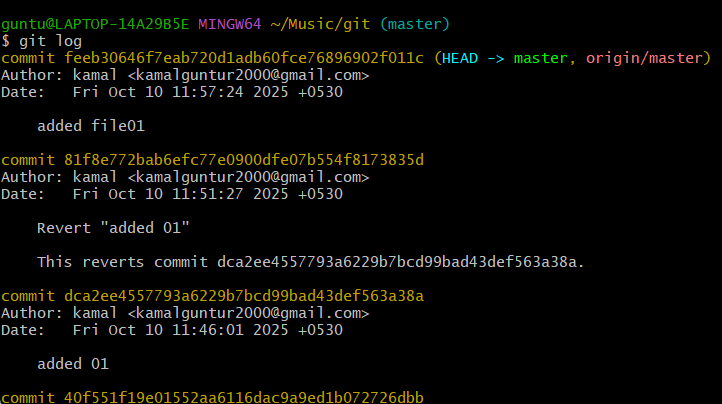
\* Every Git repository has a hidden folder called .git in its root directory.

\* It **stores info about repo**, including history, configuration, and staging info.

\*] Think of it as the **database of your Git repo**—without it, Git won’t work.

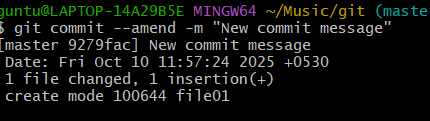
20. heck all the logs of git.

\*git log = can see all the commits



21. Rename the commit message

\*command=- git commit –amend -m



22. Merge multiple commits into single commit.

\*git commit -m “merge conflicts”

\*git push origin master

