Helo,

This is git Manual

1. Git software is alrready in ubuntu 22.04

2. Created repo called gitproject.

3. cd gitproject

4. git init

it will create all necessary files

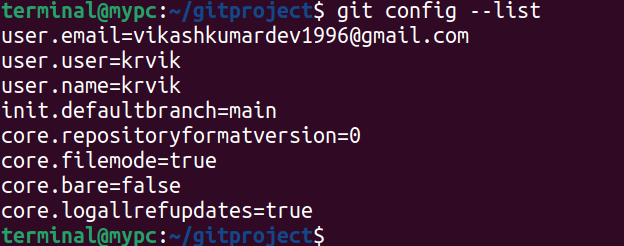
Configuring git

$git config - -global user.email “[abc@gmail.com](mailto:abc@gmail.com)”

$git config - -global user.name “username”

$git config - -global user.pass “password” ;pass is git token here.

$git config --global init.defaultBranch main

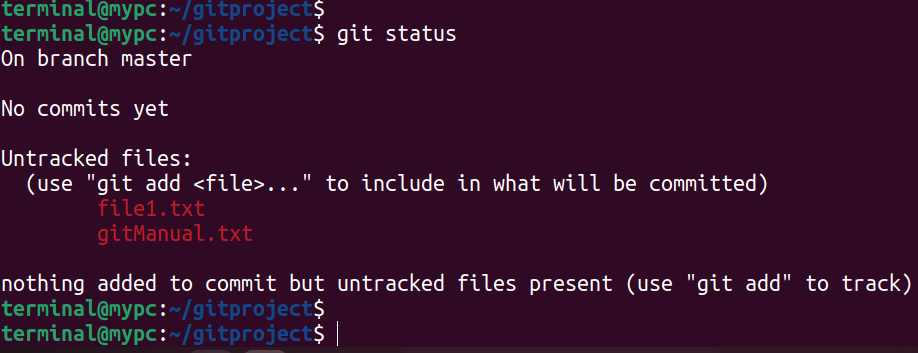


create some file

- file1.txt

then,

$git status



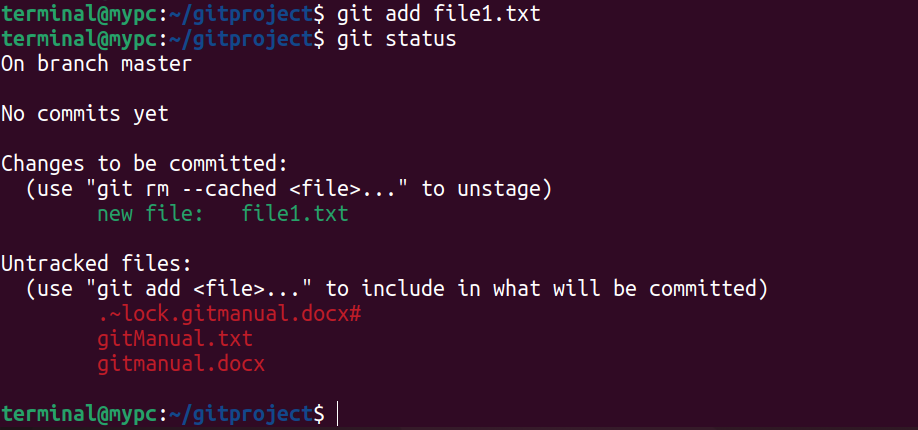
In this stage, git dont care if we make any changes bcz as you can see files are untracked.

So, we cant revert back to previous version.

So, to track the file

$git add “filename”

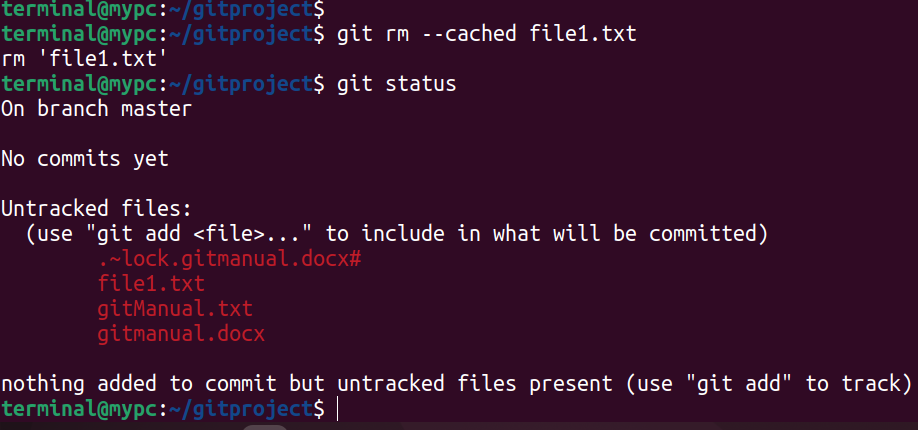
Here, you can see file1.txt is being tracked.



Suppose we dont want to track the file. To do so

$git rm - -cached file1.txt

To confirm this



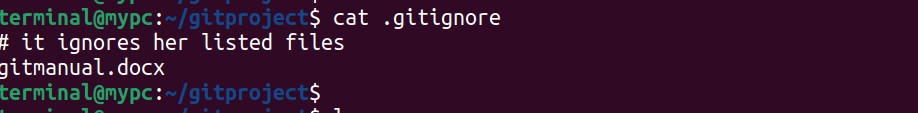
So, file is removed from tracking area of git ie cache space.

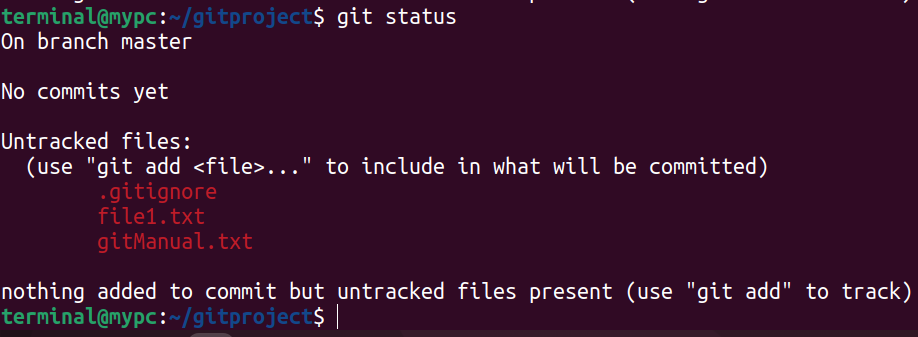
#2

if you want to ignore some files from git repo.

- create .gitignore file

- And list the filename/ extension you want to ignore.



Now, you can see gitmanual.docx file is not in the output of “**git status”**

ie git completely ignored.

If file is still open to edit something, its locked file will be listed here.

Now, we ignored the necessary file.

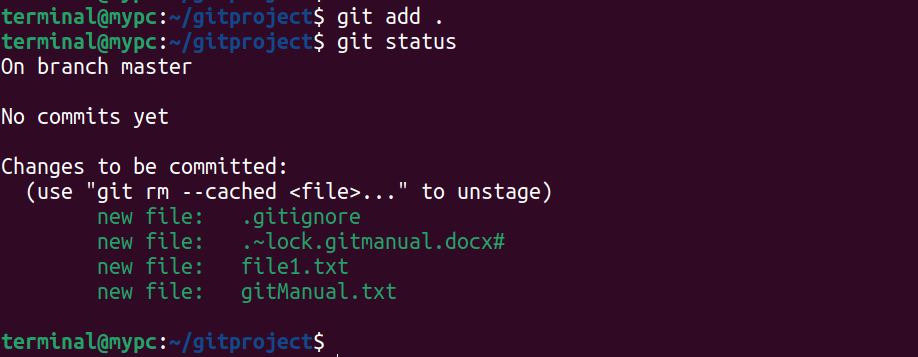
To track all files of directory except ignored

**git add -A**

**git add –all**

**git add .** Any one of these command

Lets check

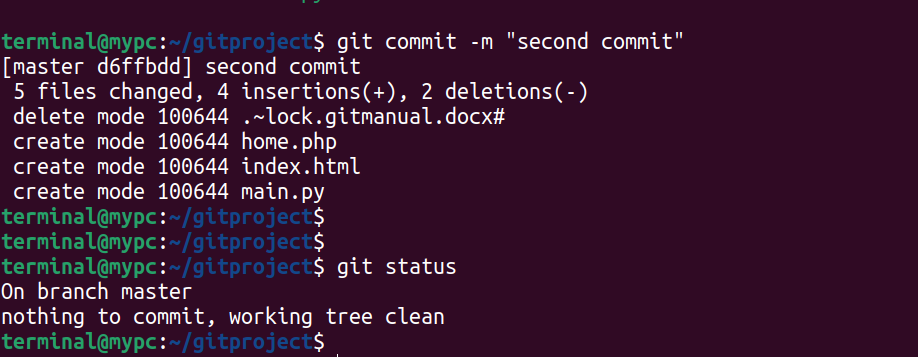
Here, you can see lock file, bcz file is gitmanual.docx file is open/editing.

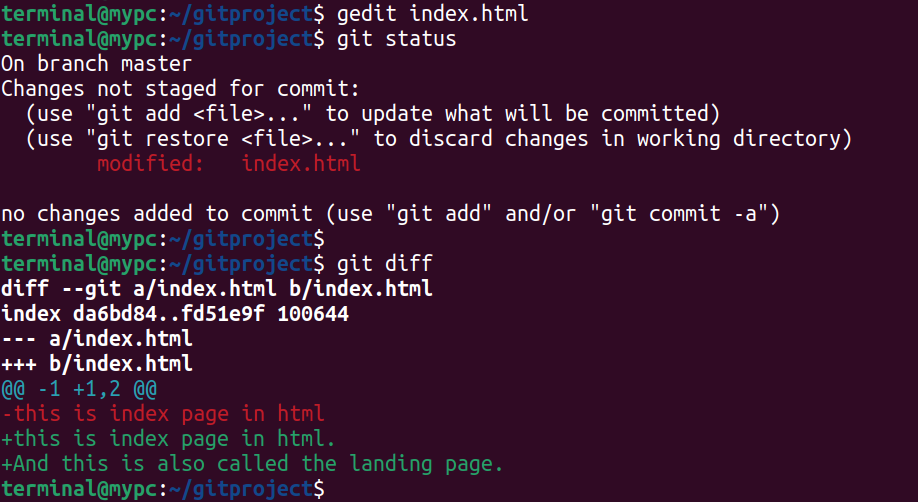
This space is also called **staging space**, means file is ready to commit.

**#3 Commit**

Commit means taking snapshot of your repo at this point of time and marking history of your repo.

So, that you can return whenever you want to.

Now you can see no file left to commit.

Lets modify the file.

You can see the dfifference

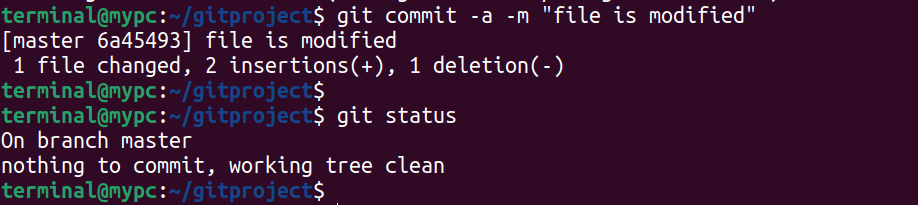
Working file ------> staging --------------> Commit

Command to remove file from staging to working area

**$git restore - -staged filename**

Command to directly commit the file by bypassing the staging area

$git commit -a -m “text message”



if you mistakenly deleted fie from file explorer,

you can get it back by git

$git status :it will show the deleted filename

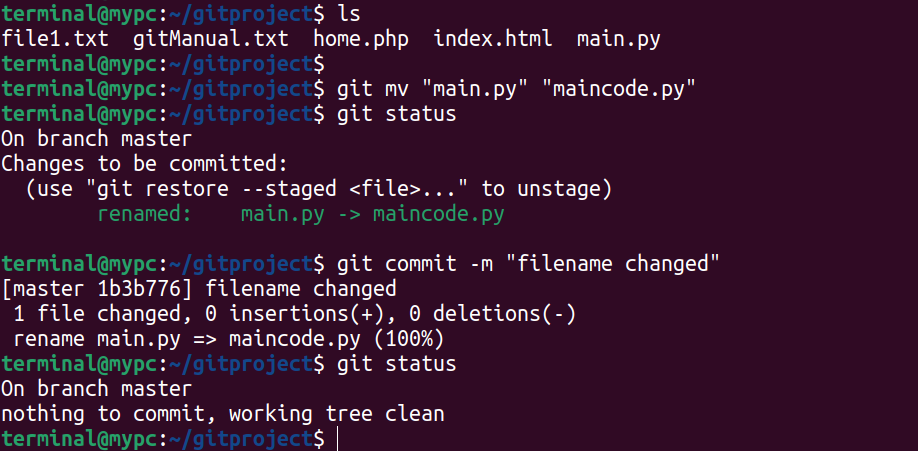
$git restore “filename” :it wil restore filename

$git mv “oldname” “newName” :renaming of file in git

And want to make this changes in hostory book

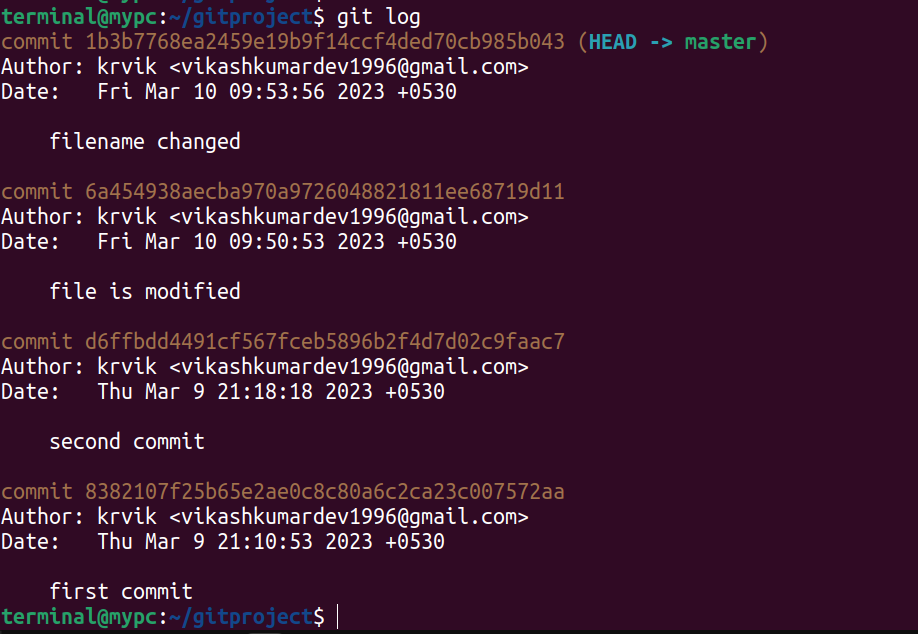
So commit it.

$git commit -m “filename changed”



Till Now, we have made so many commits, to view this entry in hoistory book

$git log



You can see every commit has different unique.

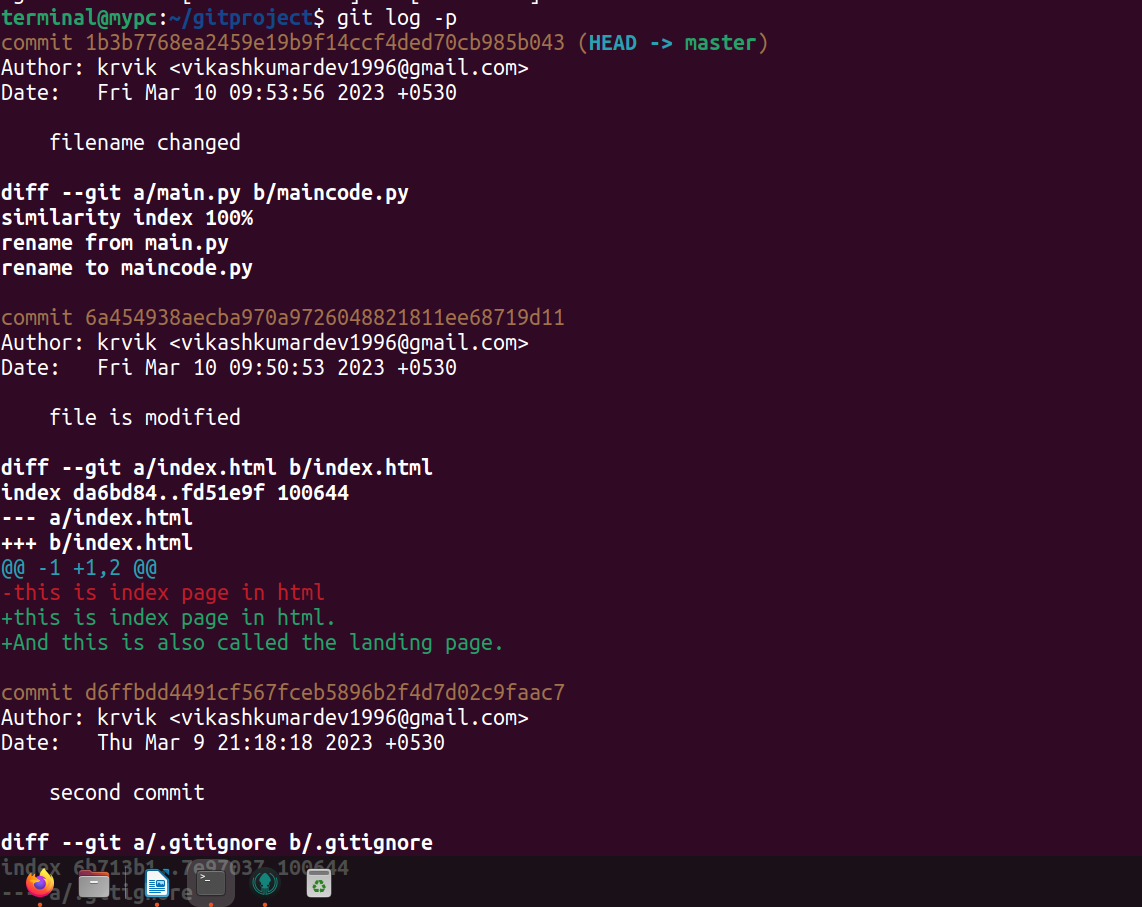
**$git log - -online** :oneline log

Want to change the commit message text.

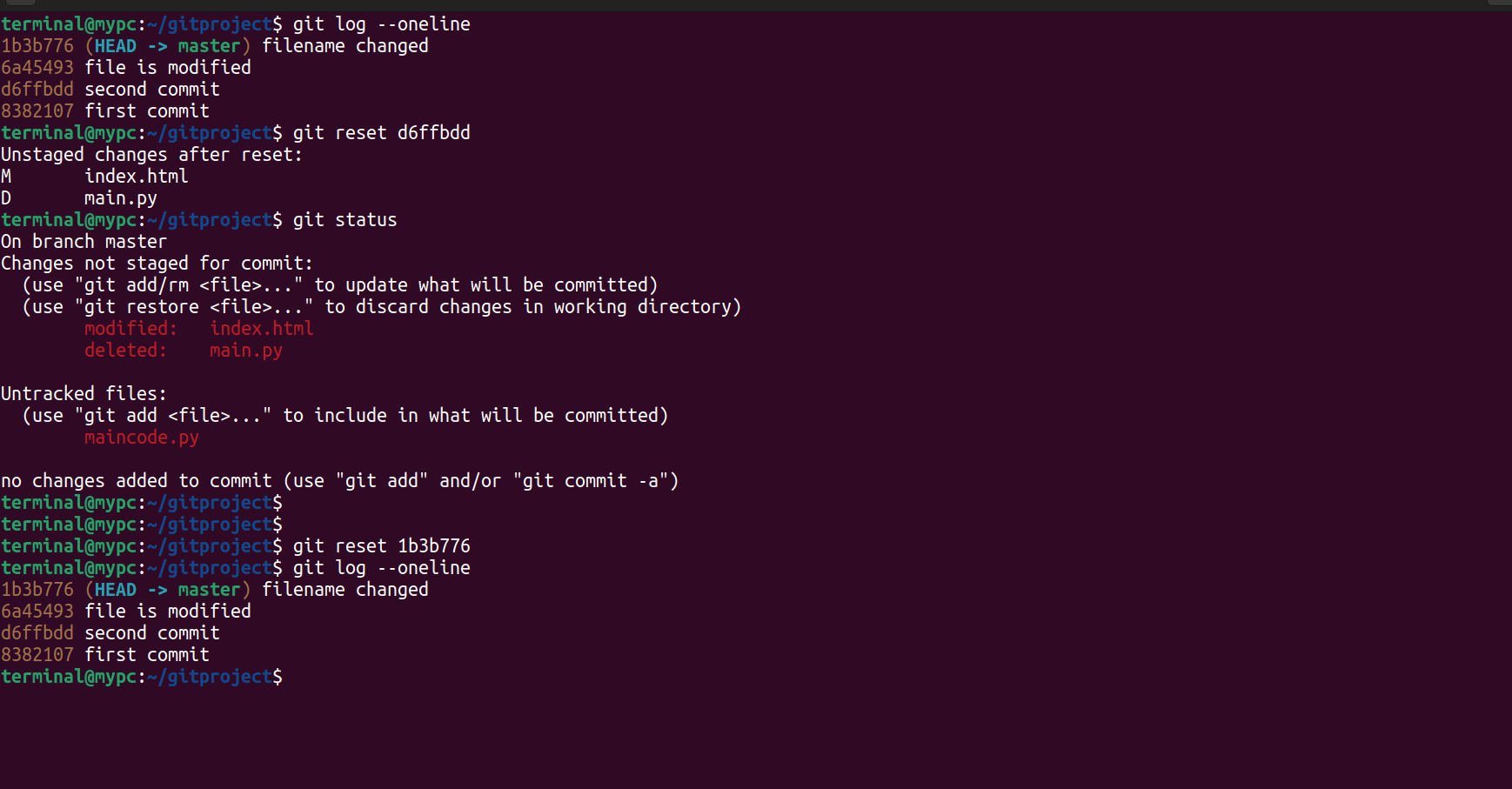
**$git commit -m “Text you want to write” - -amend**

Want o see what changes made in different commit

**$git log -p**



Restore: used to move rollback to previous changes



Rebase:

- you can also modify what appera in history book and in what order it will appear.

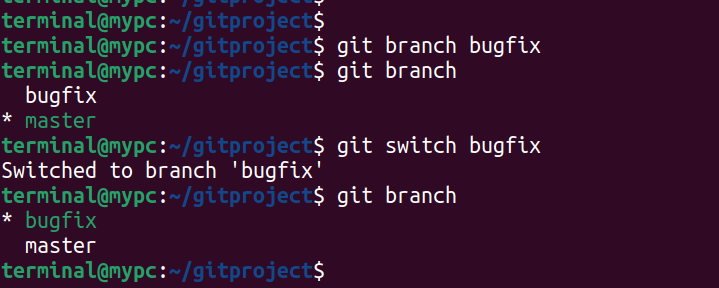
**$git rebase -i - -root**

it will open the content in text editor.

Creating branch:

Branch is the replica of main.

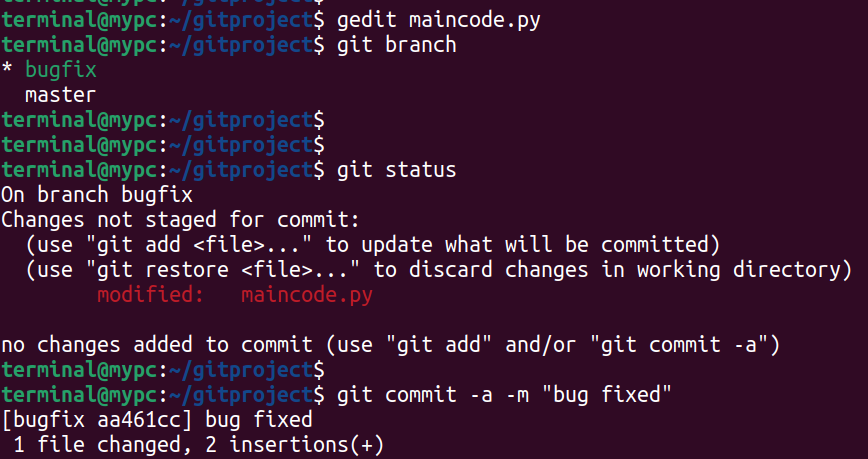
We created here bugfix

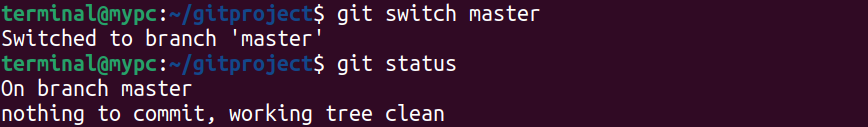


$git branch branchname

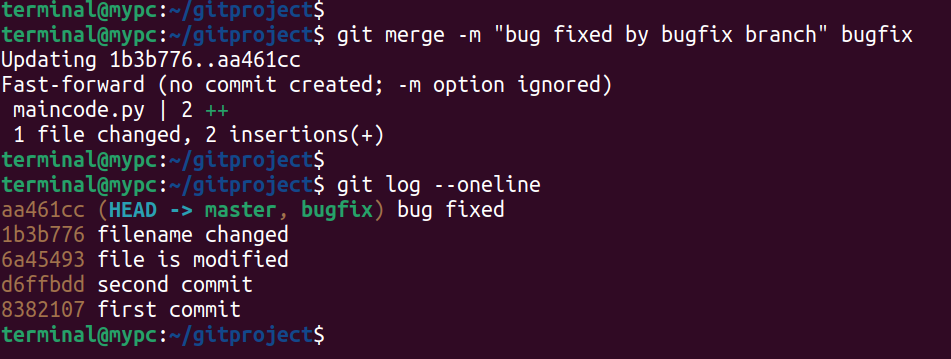
$git branch

$git switch branchname

Now, switch to master branch



Here you will find nothing is changed. Bcz its not be merge from bugfix branch



Now since bug is fixed and issue is resolved. So we dont need this branch anymore

Delete this branch

**$git branch -d branchname**

**so, you can see branch delted**

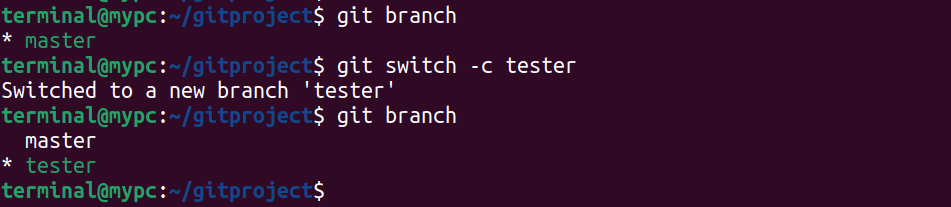
In above case, we created a branch made chages and merged it.

Here things went smooth, becasuse master branch not changed.

What if master branch also changes, in this case there is **merge conflict.**

Lets create branch tester

**$git switch -c tester** ;here branch creaed as well as switched.

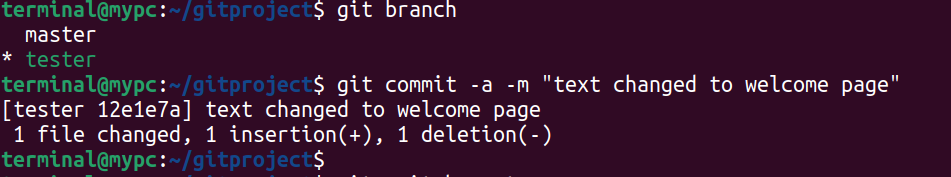
****

In index.html file:

And this is also called landing page

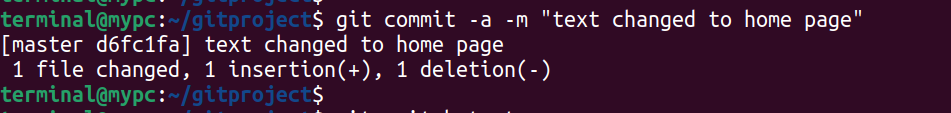
Tester branch changes it to:

And this is also called welocome page.

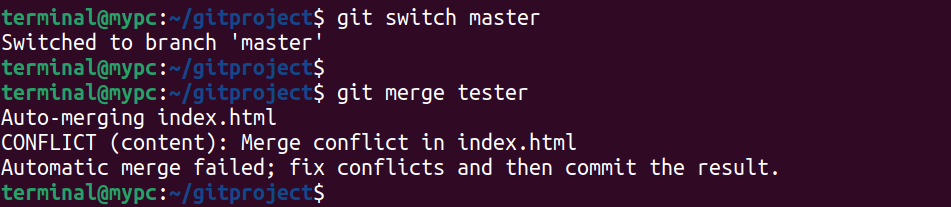


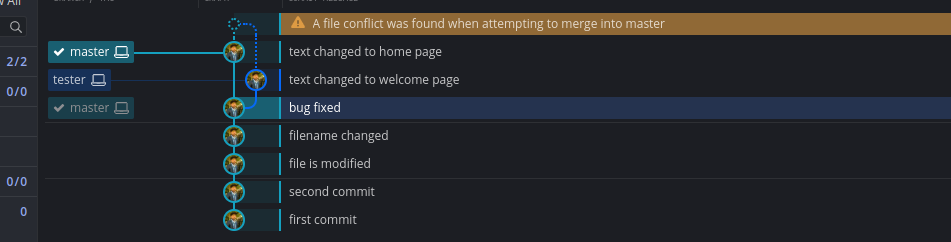
Tester branch changes it to:

And this is also called home page.

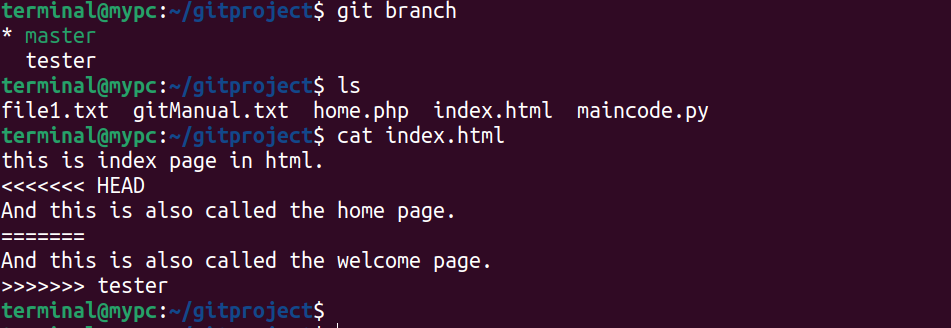


Now,



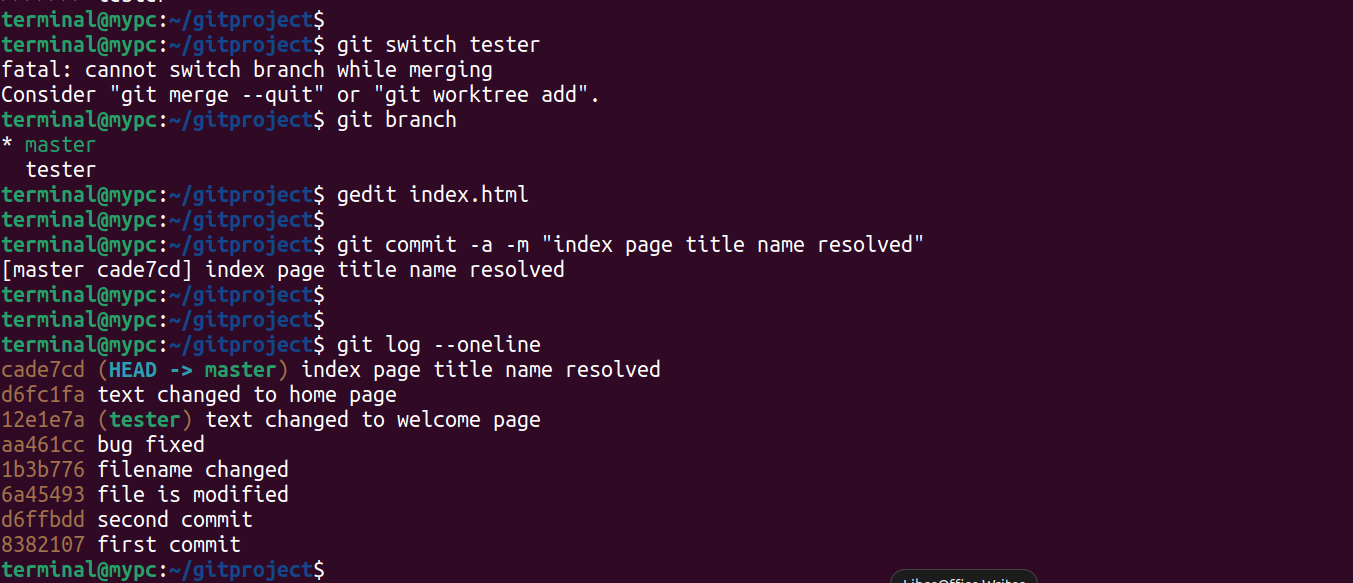


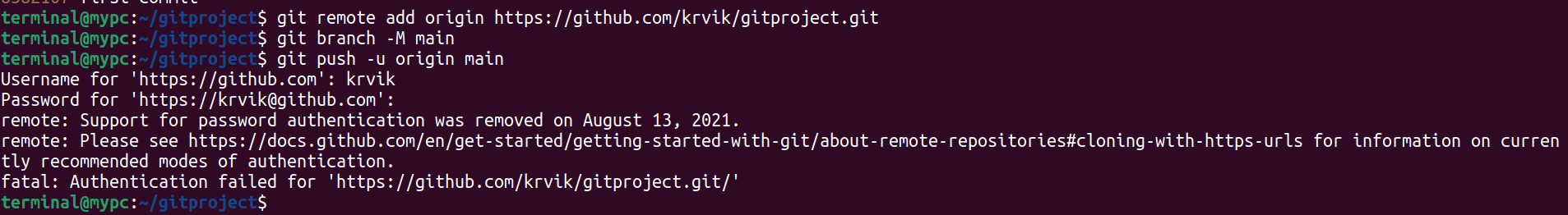
Fixing the merge conflict:



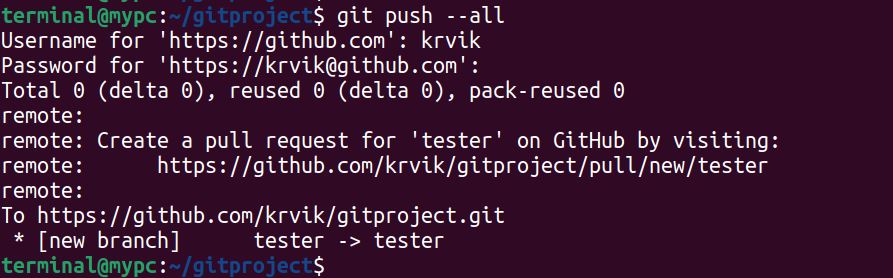
Here, you can see both the content of tester as well as of master branch

You can choose the appropriate word. And thte commit it.

Dff

To resolve create token from setting > developer settings > Personal access token > Token (classic)

to push from all branches to github

Now, you can see there are 2 separate branches in github maiin and tester

df