**Mobile Computing**

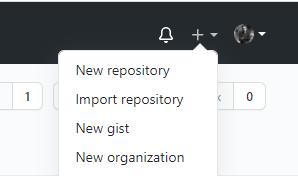


**Submitted to: Sir Haq Nawaz**

**Submitted by: Hafiz Muhammad Ali**

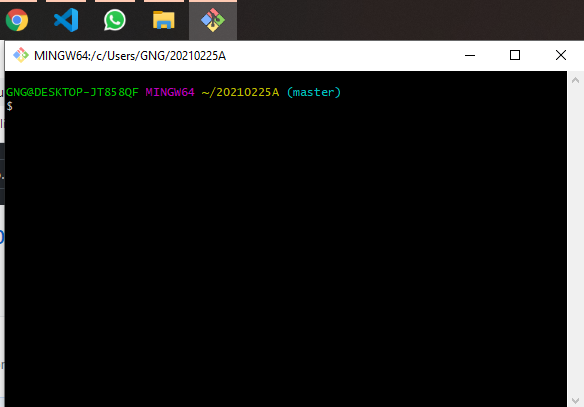
**Roll no: BSEF18A002**

After creating my account on github I have created a new repository on github with this name > “20210225A”





Downloaded git bash

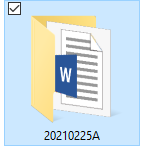


And then cloned my repository from central repository to local repository by using this command

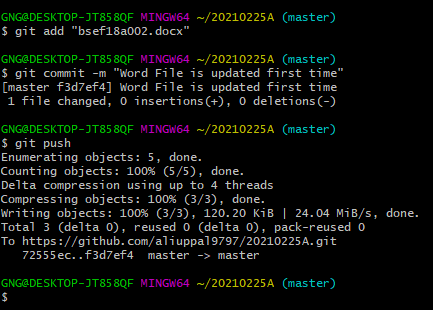
git clone [url]



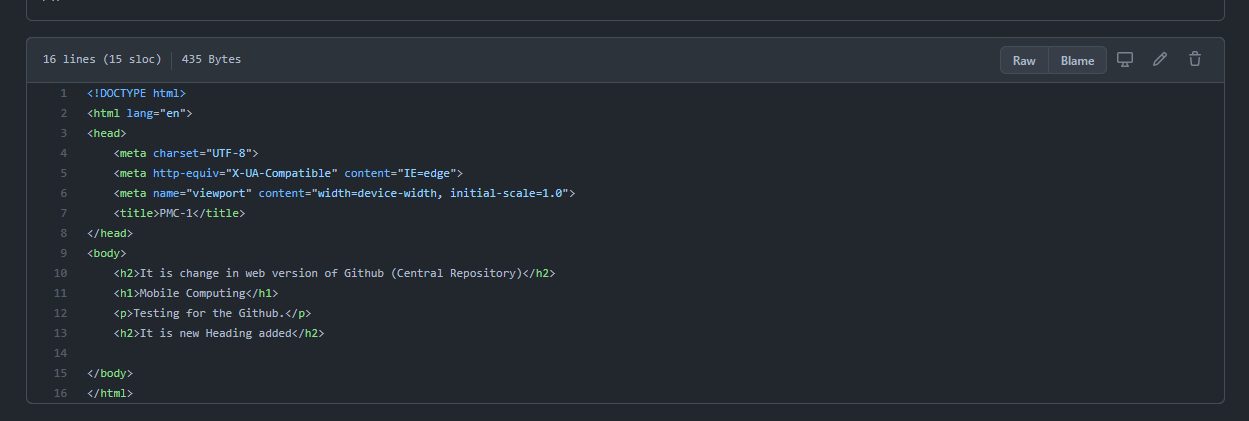
Here is our local repository which is linked with the central repository and in that repository I have created a docx file in it.



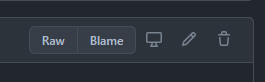
Then I moved to local repository and added some content in docx file and then updated this file on central repository.



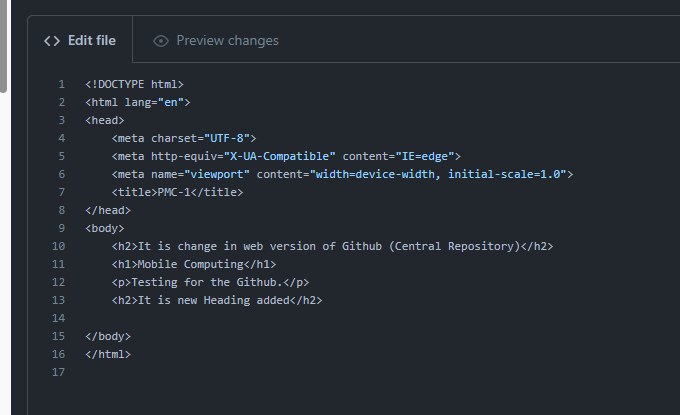
On central repository we open our file and make some changes in it.

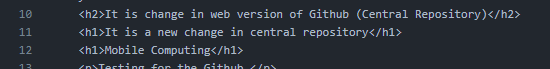


Here we get the edit option.

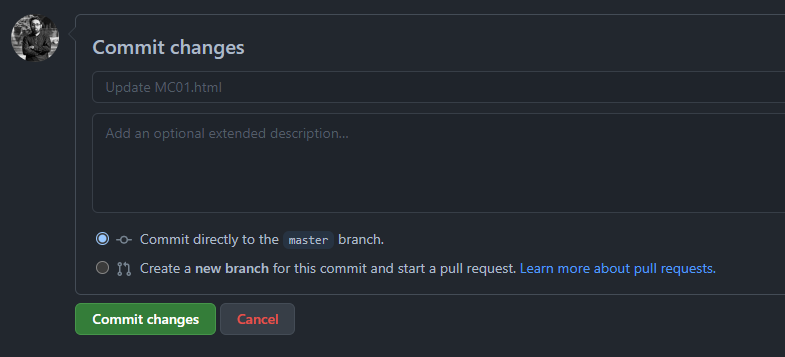


Then on central repository editor will be opened for editing.

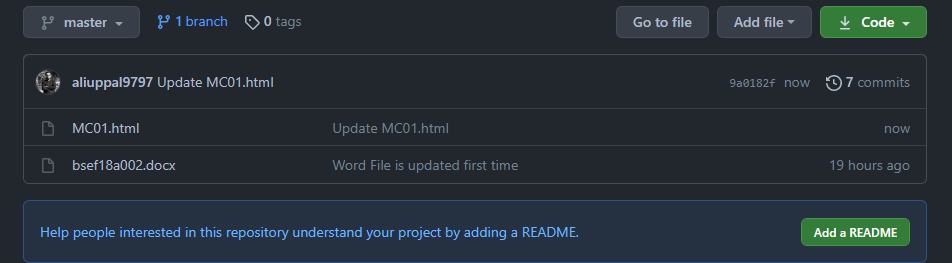


I have added line number 11 here. 

Then we can place a commit message here and then click on Commit Changes option. This will update MC01.html.

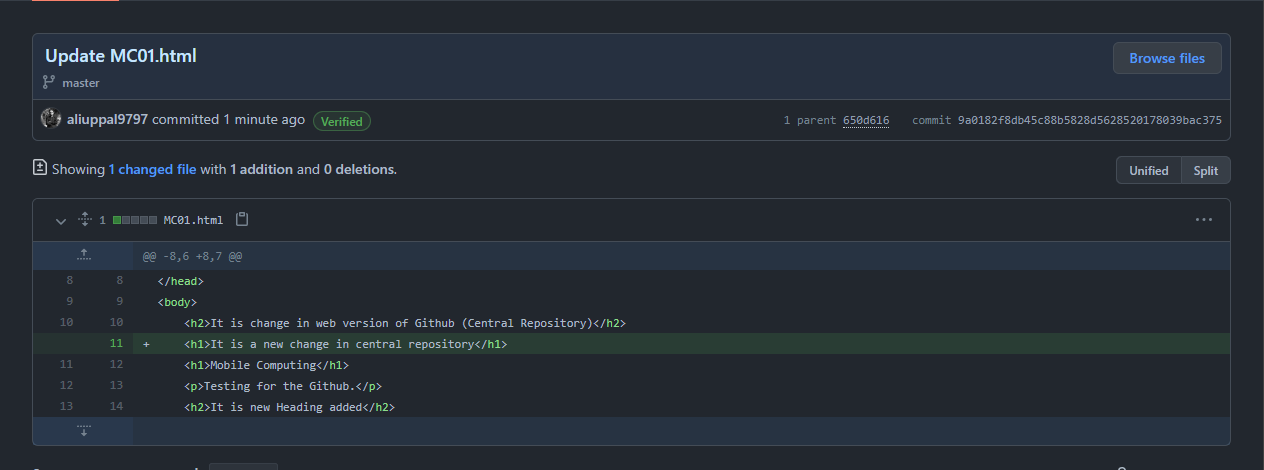


Here we can check that MC01.html is updated and a new commit is added there.

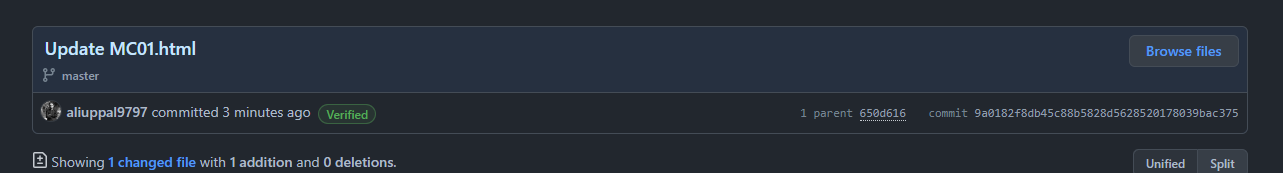


By clicking on the new commit name, we can see the changes in our central repository.

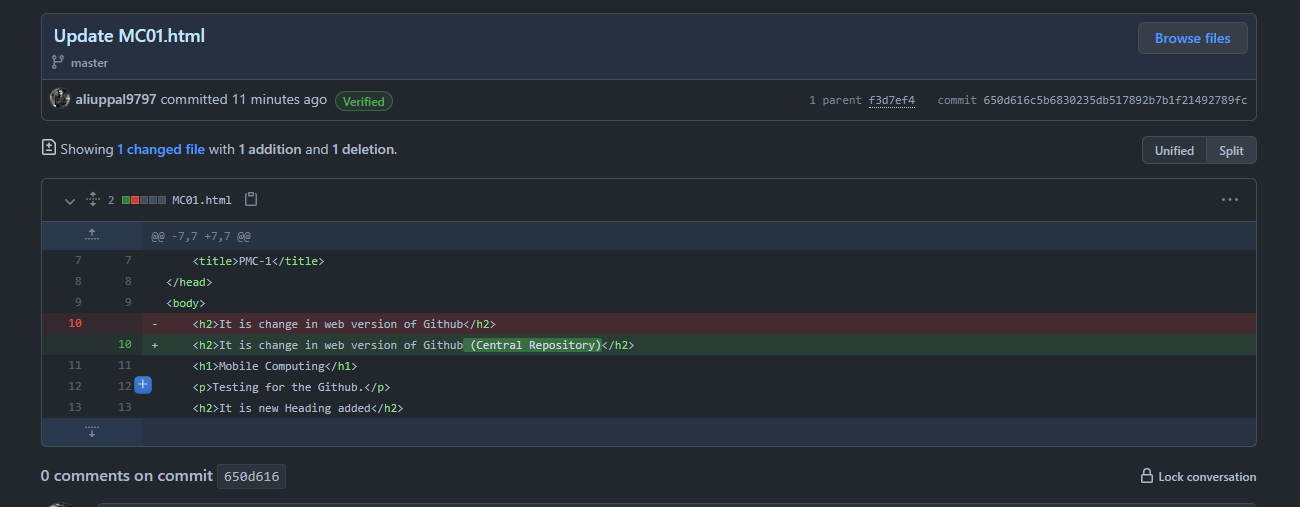
Here we can see that line number 11 is greenish which means this code is added in this repository.



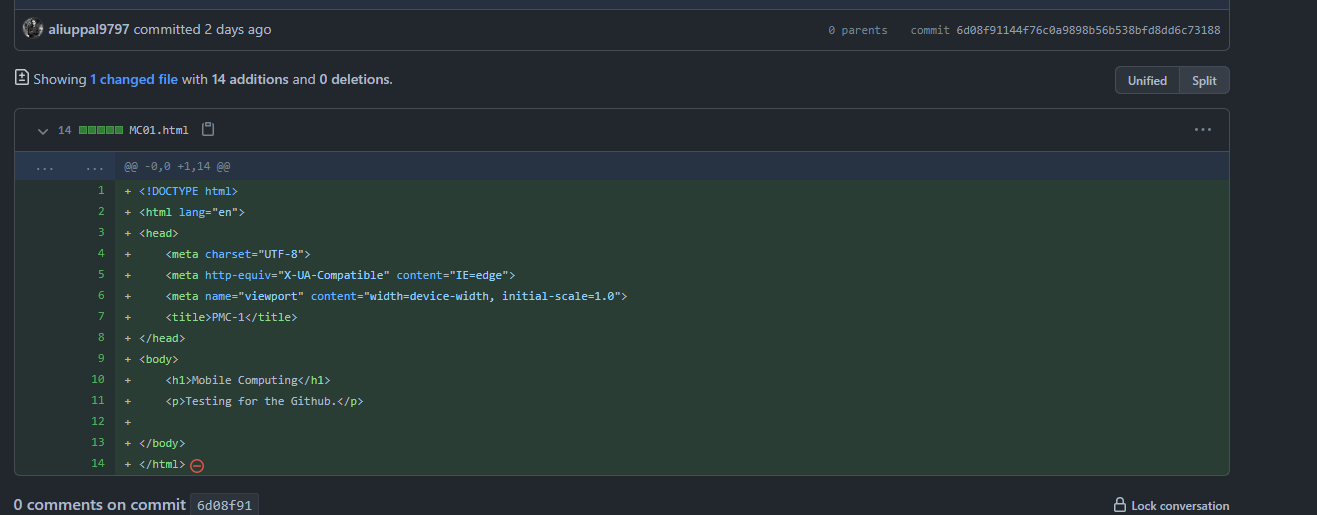
Here we can see it has one parent. We can see its parent by clicking on parent’s file number.



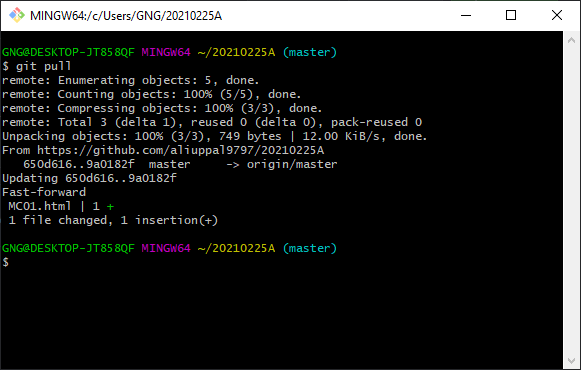
Here is the parent of latest commit. We can see it. And it also has one more parent.



By clicking on its parent’s file number we can see its parent.

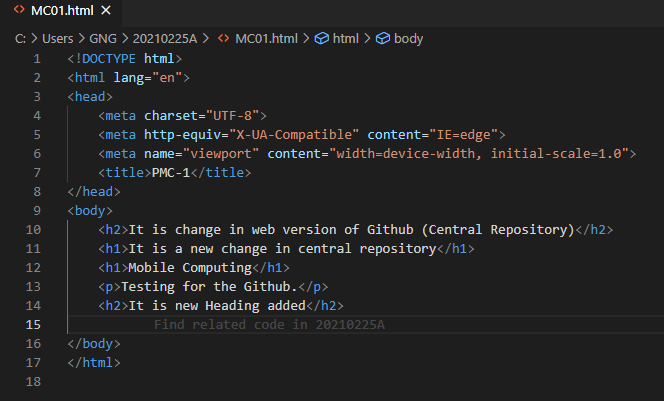


Now we come on git bash and use   
git pull

Command 

Now our local repository is synchronized with the central repository.

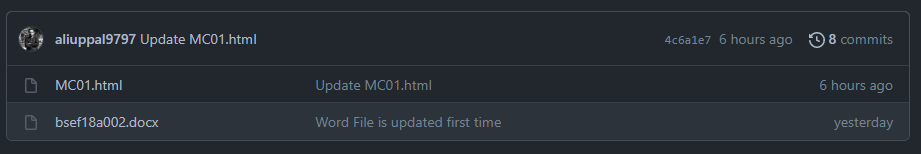
Here we can see that MC01.html on our local repository is updated.

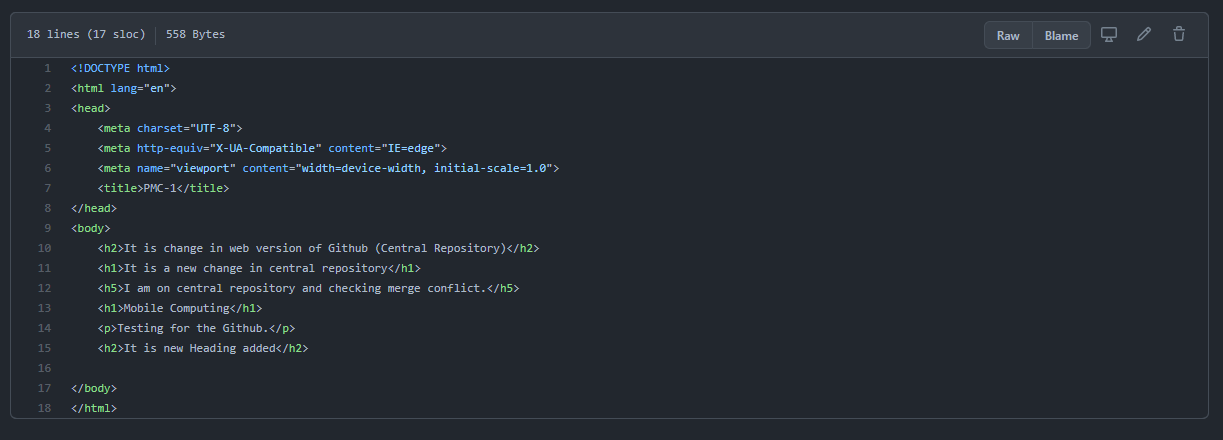


**Merge conflict**

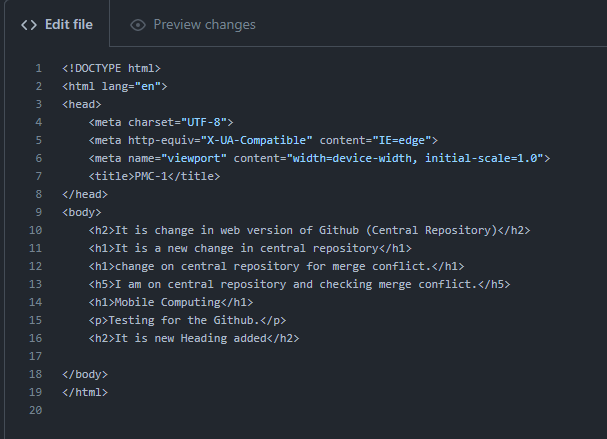
Merge conflict occurs when same line is changed on both repositories (on local and on central repository).

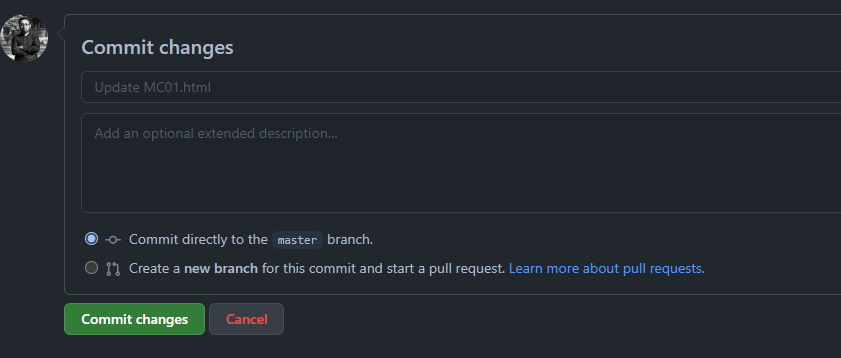
To elaborate merge conflict first of all on central repository we open MC01.html by clicking on its name.

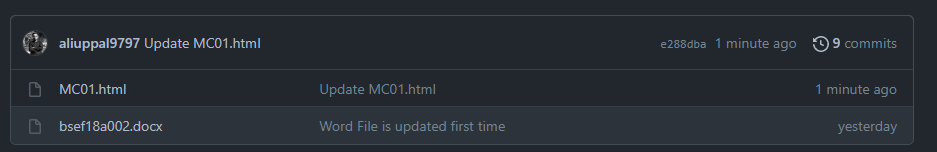


Then this window will appear. 

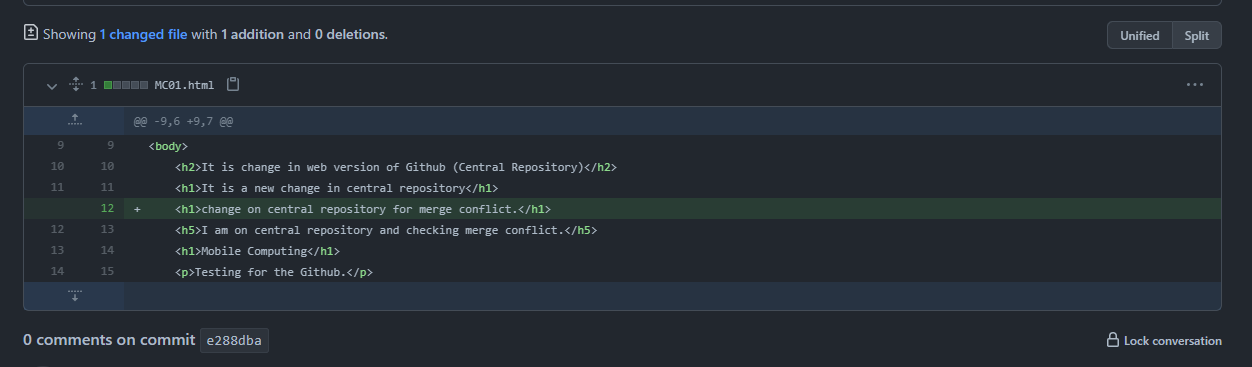
Click on edit option. (which is on upper right corner .

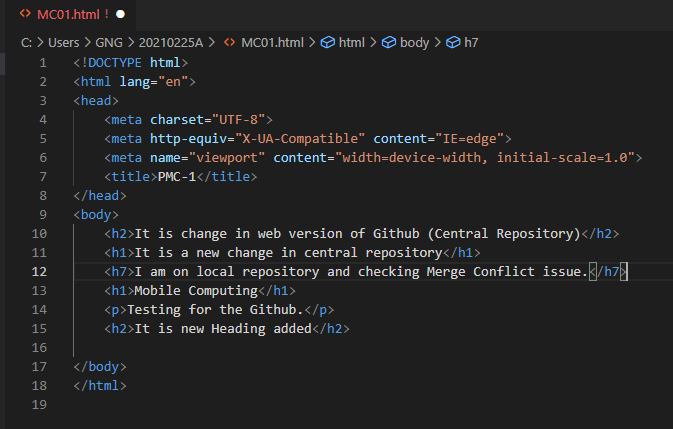
Now we added line number 12 on central repository.

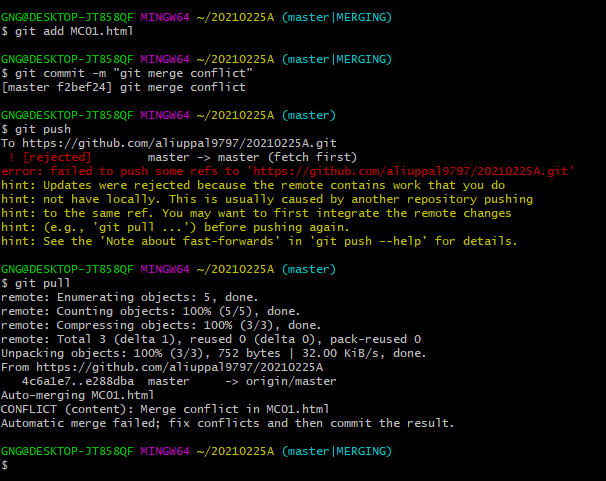
Now after writing commit message and name, click on commit changes. 

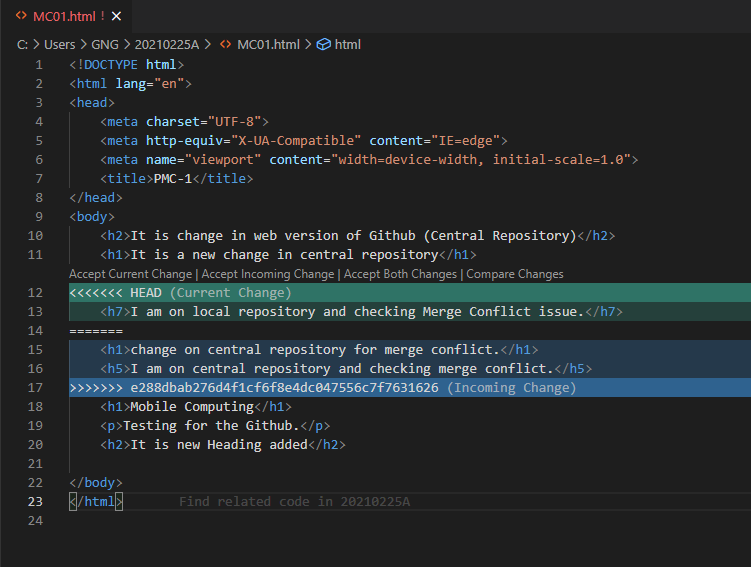
Your commit changed successfully. For verifying that changes. Click on new commit’s option (Update MC01.html) 

You can check line number 12 is changed

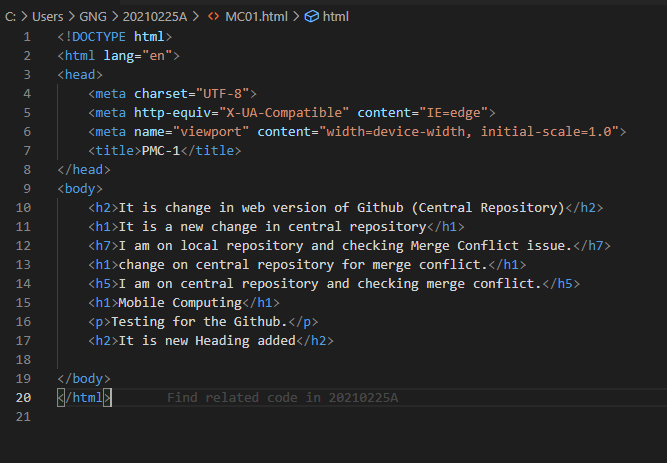


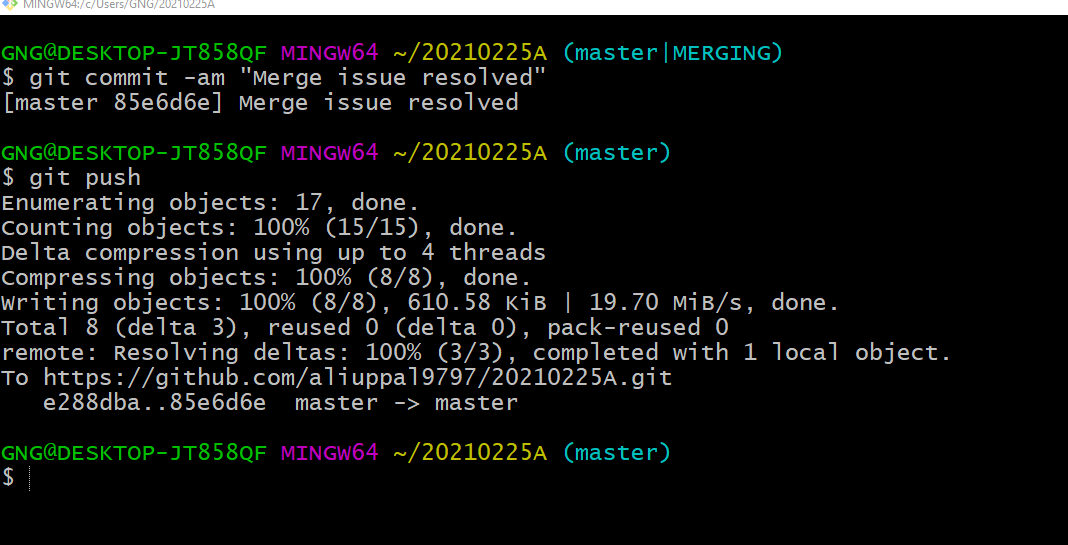
Now on local repository we update MC01.html by adding line number 12. 

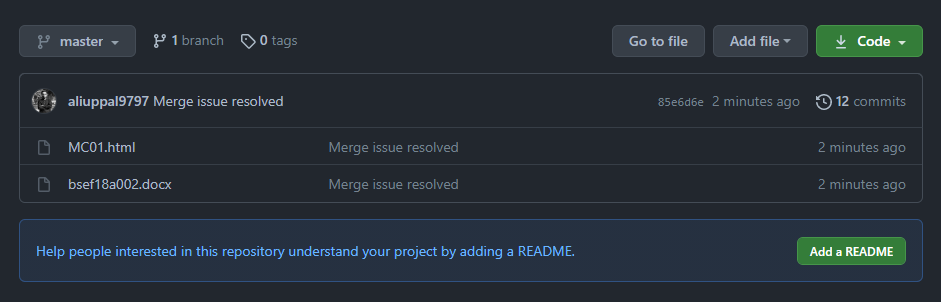
After saving this file on local repository, we open git bash and write these commands on it. And the conflict error arises.

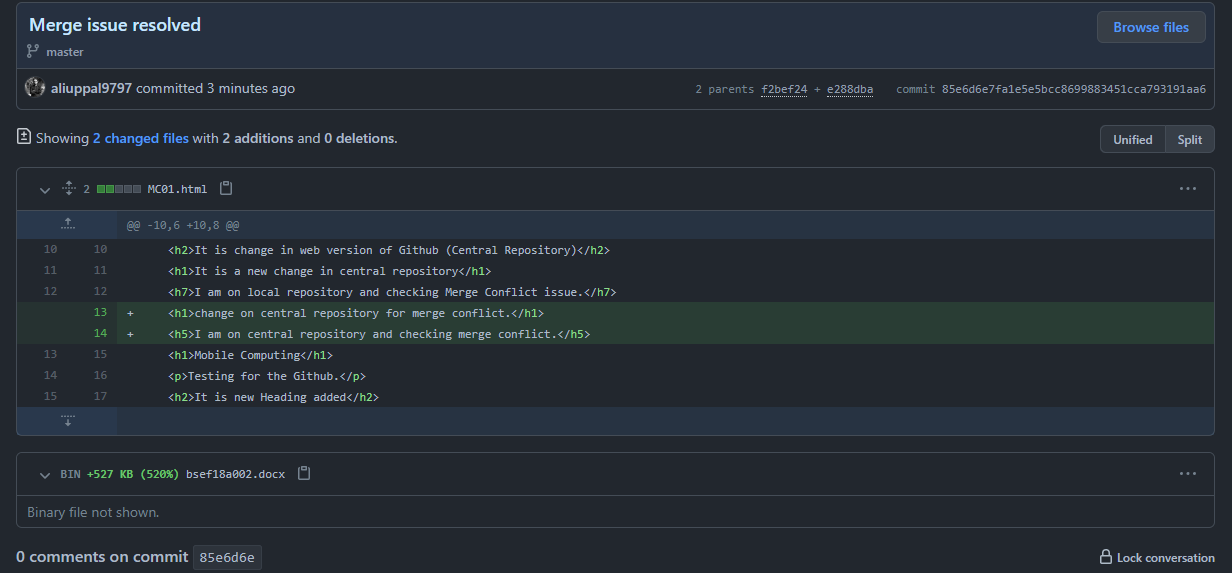
When you open MC01.html on your IDE on local repository. Then you will get these options. 

You can chose one of these options for solving this conflict. 

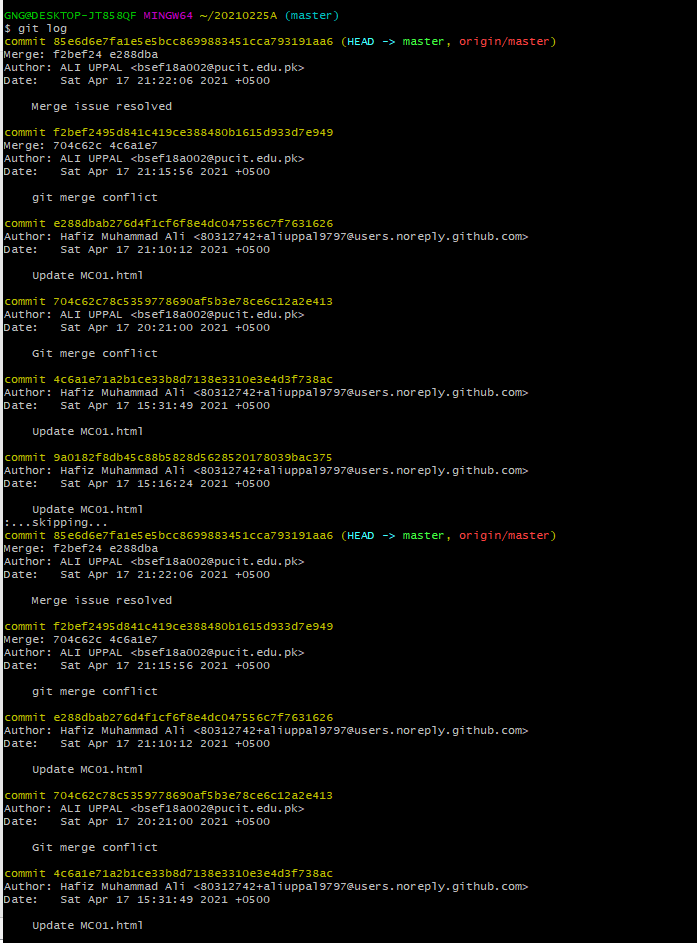
I chosed Accept Both Changes. Now this conflict is resolved and both changes are updated on my local repository. 

After solving conflict issue we open git bash and run these commands. And you can see these is no error anymore. 

For varrifying we go to our central repository. 

Click on new Commit which is Merge Issue Resolved. And you can see your file is updated. 

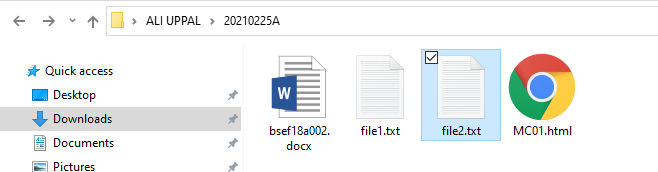
**Git log**

This command is used for checking log of our gits. 

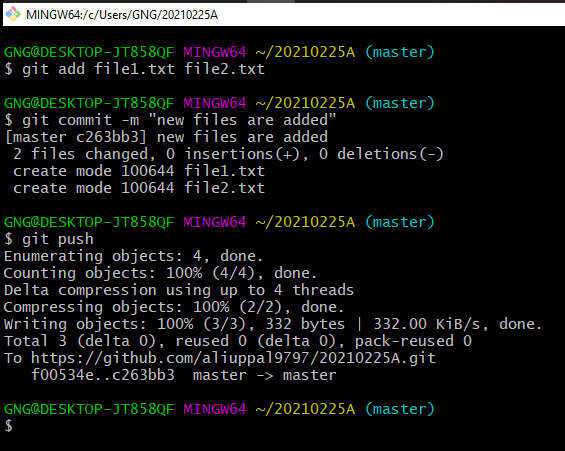
**Git rm**

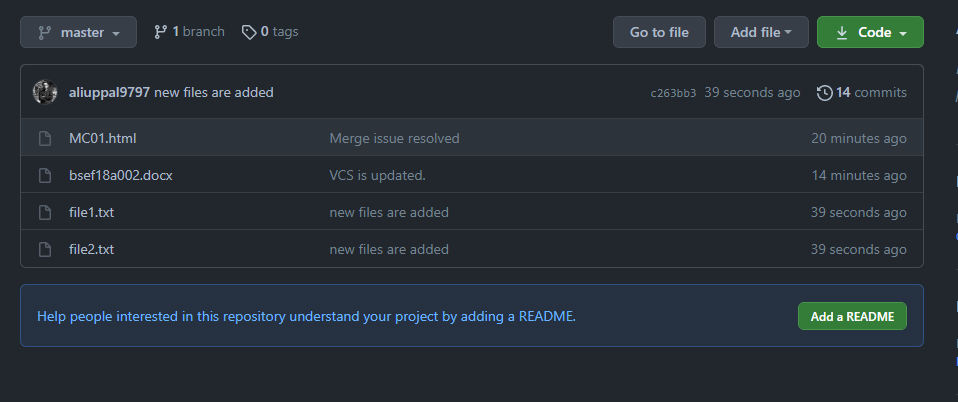
First of all we create 2 new files on our local repository.

* file1.txt
* file2.txt.



now add these two files on our central repository by using these commands on git bash.

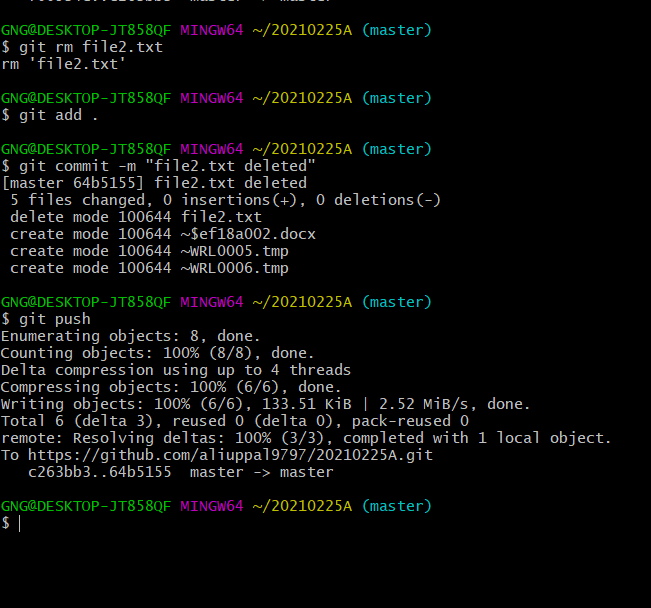


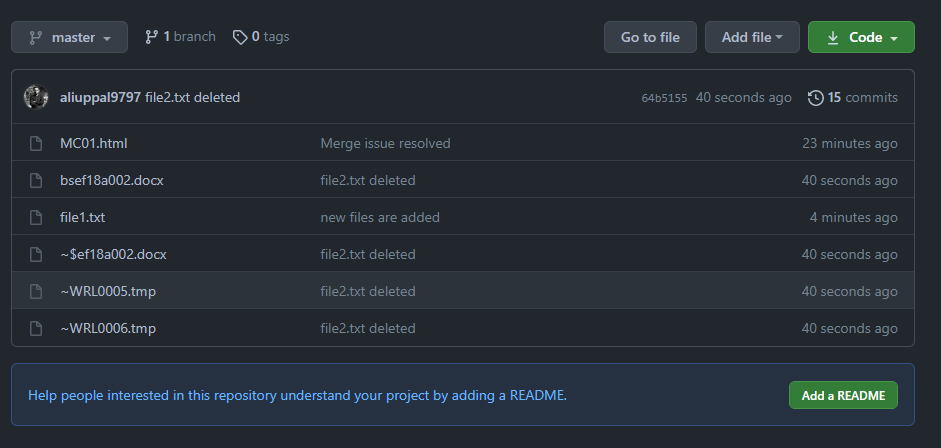
You can see 2 file1.txt and file2.txt is uploaded on our central repository. 

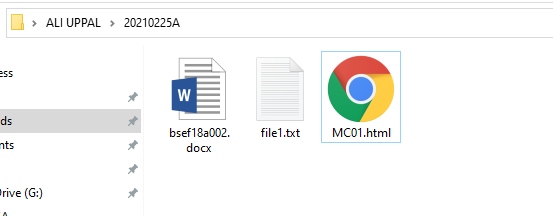
Now we remove file2.txt by using

**Git rm file name**

on git bash



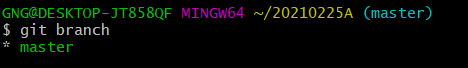
You can see file2.txt is deleted from our central repository. 

File2.txt is also deleted from central repository. 

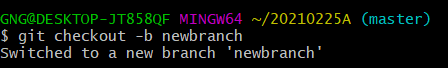
**BRANCHING**

* git branch
* git checkout
* git merge

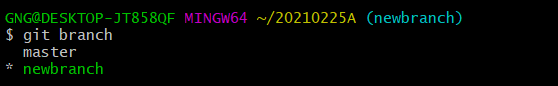
git branch

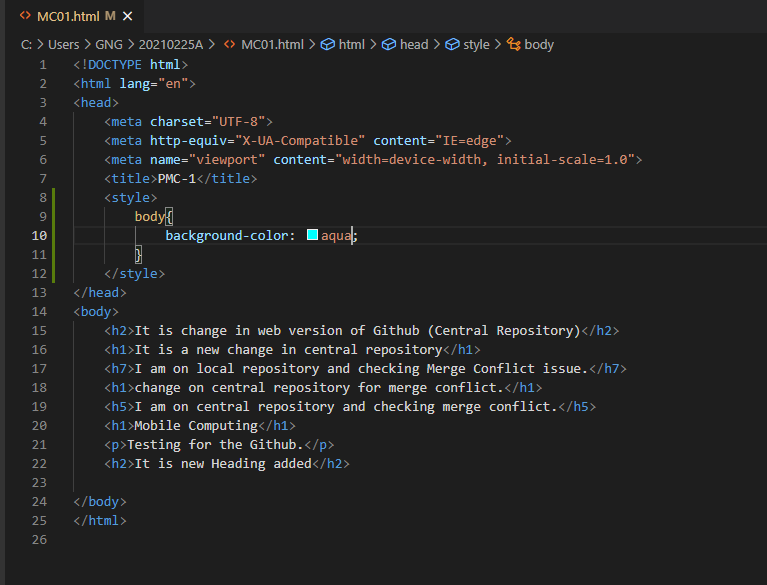
by using git branch we can check our current branch just like this. It shows master as our current branch.

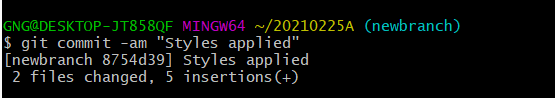
git checkout –b branch\_name

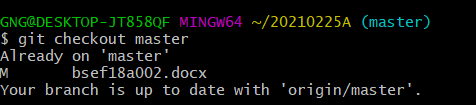
by using this git checkout –b branch\_name, we make a branch just like this. 

now you can check that your branch is changed from master to newbranch



now we updated MC01.html and added some CSS in it. 

by using git commit –am we added and committed with a single command 

now we move back to master branch 

and we can see that our MC01.html file is reloaded by the previous code. And there is no CSS applied here. 