**SURPRISE TEST – SET B (EVEN)**

Name : Sakshi  
UID : 22BDO10064

Section & Group : 22BCD-1/B

Subject : Git and GitHub

Problem :-   
The task, you again live in your own branch, this time we will be doing a bit of juggling with branches, to show how lightweight branches are in git. Hint: git switch will make you switch from one branch to another.

1. Use git branch to see the two branches that are relevant for this exercise

2. What branch are you on?

3. Use git branch mybranch to create a new branch called mybranch

4. Use git branch again to see the new branch created.

5. Use git switch mybranch to switch to your new branch.

6. How does the output from git status change when you switch between the master and the new branch that you have created?

7. How does the workspace change when you change between the two branches?

8. Make sure you are on your mybranch branch before you continue.

9. Create a file called file1.txt with your name.

10. Add the file and commit with this change.

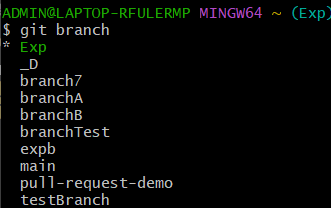
11. Use git log --oneline --graph to see your branch pointing to the new commit.

12. Switch back to the branch called master.

13. Use git log --oneline --graph and notice how the commit you made on the mybranch branch is missing on the master branch.

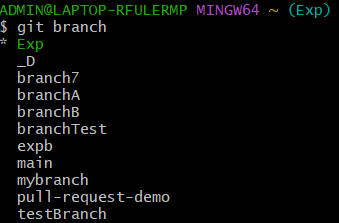
14. Make a new file called file2.txt and commit that file.  
  
  
Solution:-

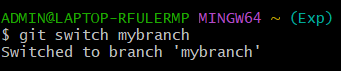
Here’s a step-to-step solution to deal with the above problem  
  
1. This command will list all the branches available in your repository   
 **$ git branch**



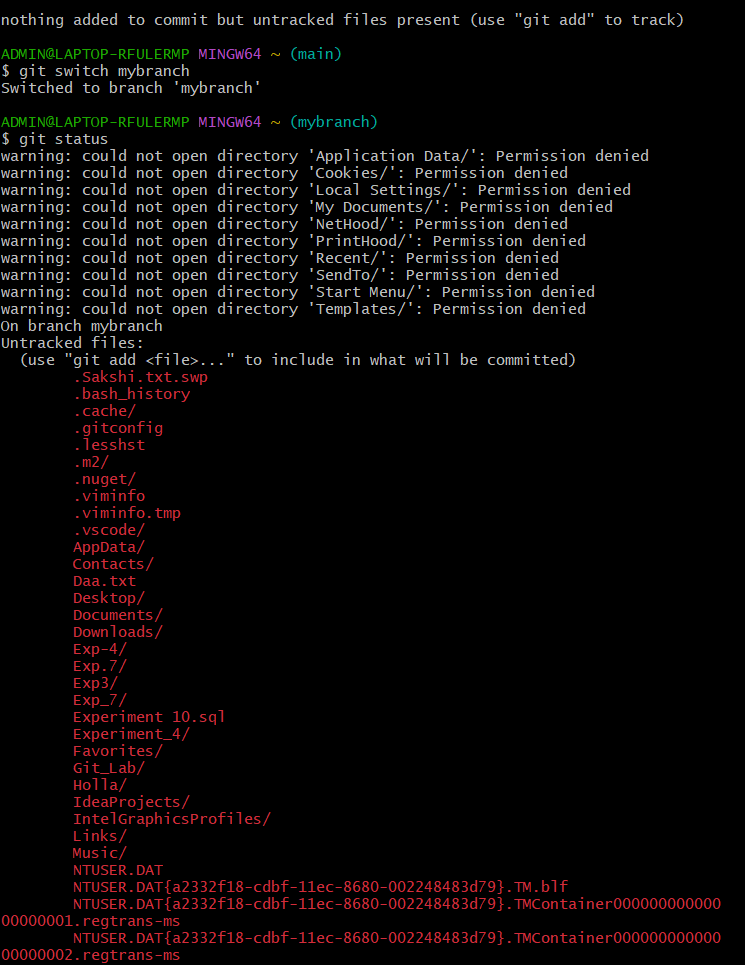
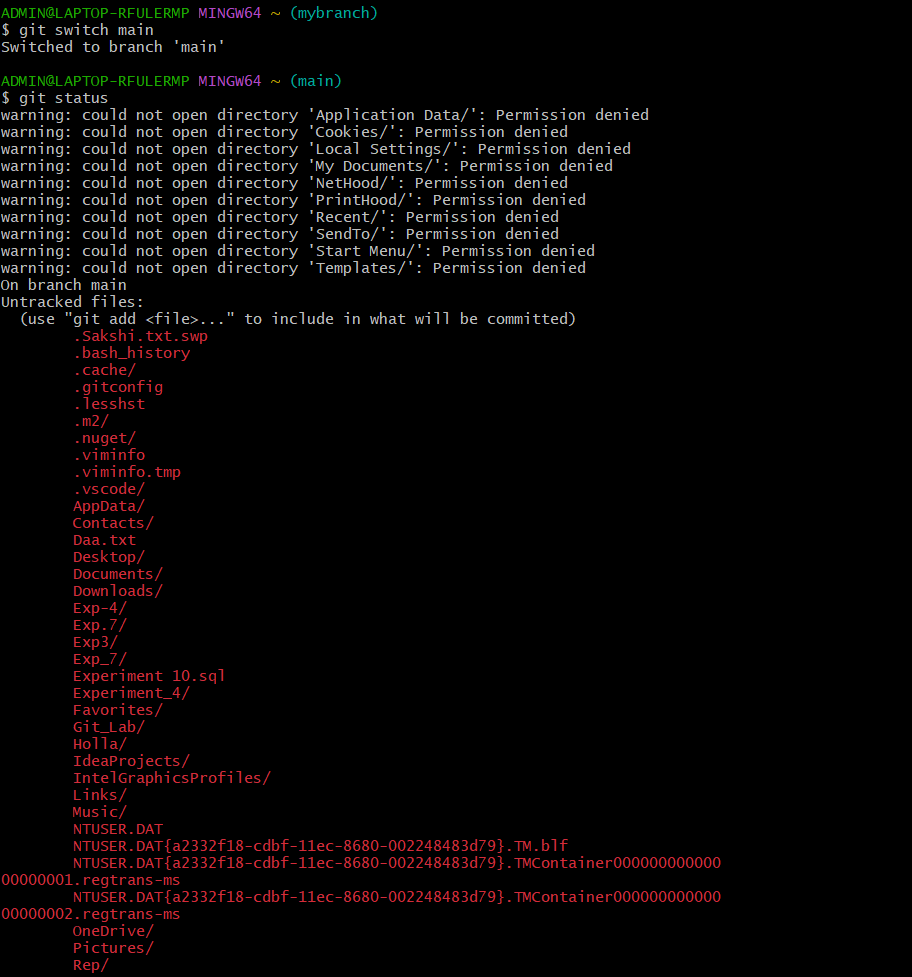
2. To check on which branch you are currently, use the command   
 **$ git branch –show-current** 

3. Now, to make a new branch named mybranch, use the command   
 **$ git branch mybranch** ****

4. Now, to see the branch again and to observe that the branch you just created is there or not, run the command   
 **$ git branch** 

5. Use the command for switching between the branches and to switch to your new branch, by using the command   
 **$ git switch mybranch  
 **

Also, you can use a different command for switching between the branches, that is   
 **$ git checkout mybranch  
 **

6. To see the status of both the branches, use the following command in both the branches: **$ git status**

7. The workspace changes when you switch between the two branches because the files and their contents may differ between branches, for which we have already used the command git status.

8. Before you continuing further, make sure that you are on mybranch. You can use the command   
 **$ git branch –show-current**to verify the current  
 

9. Create a file called “file1.txt” with your name by using any editor.

**$ touch file1.txt** -> for making an empty file



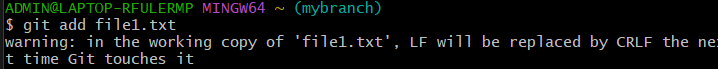
**$ cat > file1.txt   
 Sakshi** -> for entering content into the file  
 

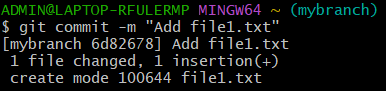
**$ cat file1.txt** -> for seeing the content of the file

****

10. Now, add the file by using the command: -

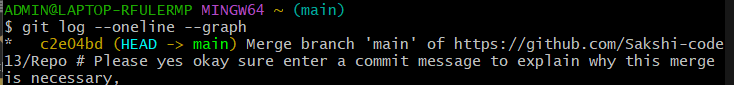
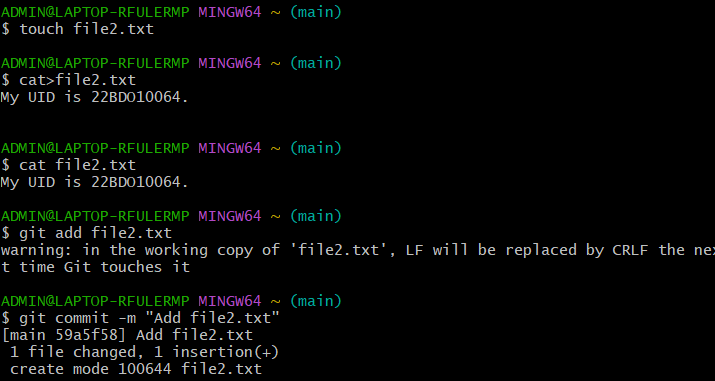
**$ git add file1.txt**

****

And, commit the changes by using the command :-  
 **$ git commit -m “Add file1.txt”  
 **11. Use the following command, to see your branch pointing to the new commit  
 **$ git log –oneline –graph  
**

12. By using the following command, switch back to the main branch  
 **$ git switch main**

****

13. Now, use the command **$ git log –oneline –graph** and notice how the commit you made on the “mybranch” branch is missing on the “main” branch  
  
  
14. Lastly, make a new file called “file2.txt” by using any editor, after that add it, and then commit it by just the way you did with file1.txt  
 **$ touch file2.txt** -> for making an empty file   
 **$ cat > file2.txt** -> for adding some content into the file  
 **$ cat file2.txt** -> for seeing the contents of the file  
 **$ git add file2.txt** -> for adding the file2.txt   
 **$ git commit -m “Add file2.txt”** -> for committing the file2.txt  
  
 Also, the reference to the solution is attached here within :  
