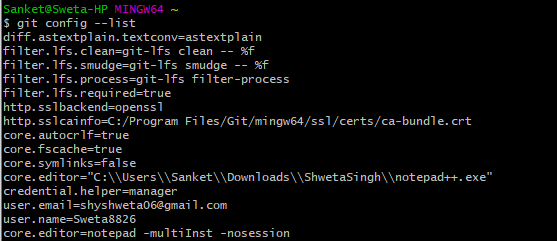
**While doing the below labs, you need to take screenshot for each step, for each assignment and create word document. Upload the word document to topgear.**

1. Set the global configuration file with your user name, email and editor as Notepad++. List all the properties which you just set.

**ANS: git config - -global user.name “sweta8826”**

**git config - -global user.email “**[**shyshweta06@gmail.com**](mailto:shyshweta06@gmail.com)**”**

**git config - -global core.editor “notepad –multiInst –nosession”**

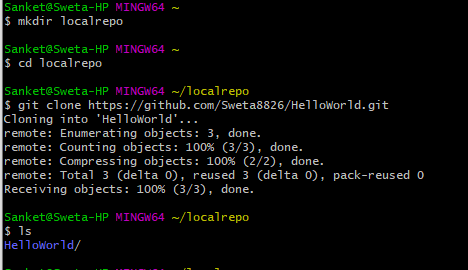


1. Make a fresh Git project

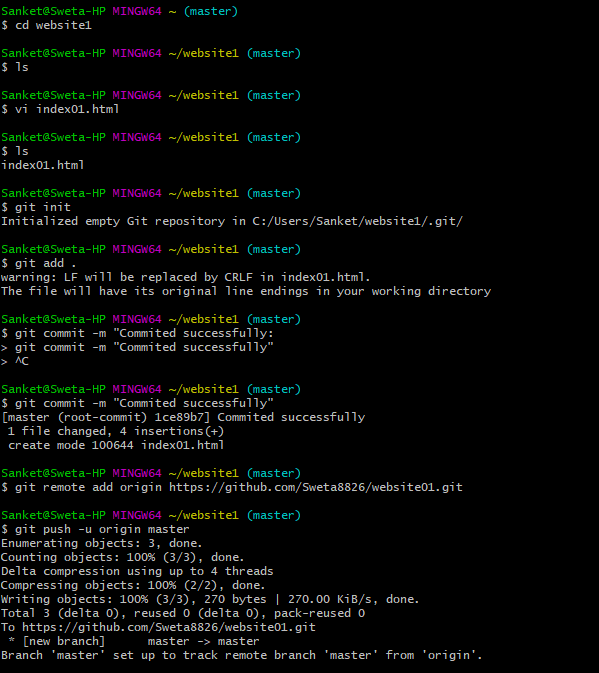
ANS: Create a directory , get inside directory ,do **git init** which will initialize git repository for the very first time and then followed by **git add .** and finally **git commit** and push it into your github.

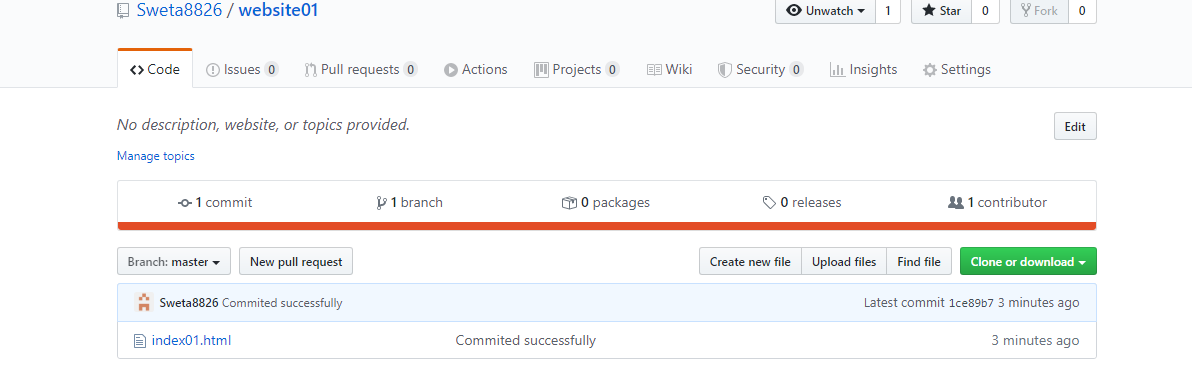
1. Create a Github account (Or use the account if already registered). Clone a project from the remote repository to your local repository.

ANS: **Create a directory and clone remote repository into that local directory.**

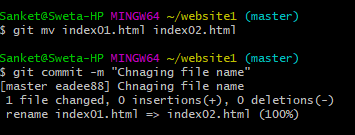


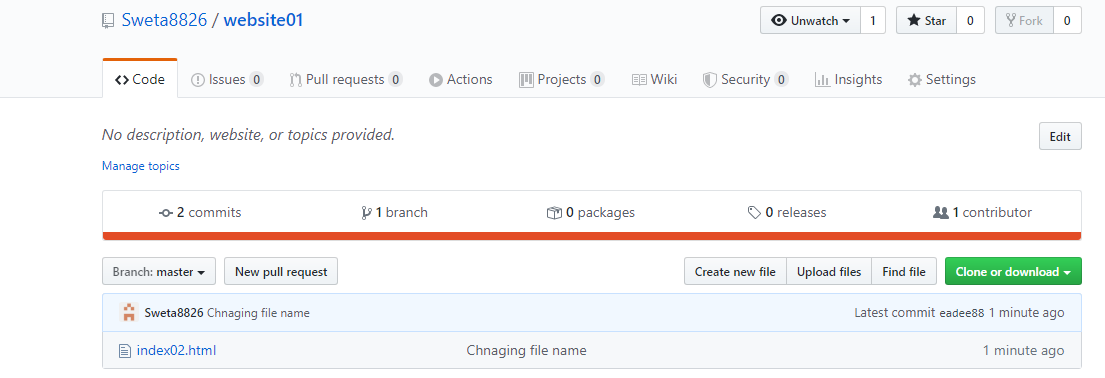
1. Push the project created in assignment 2 to the remote repository.





1. Use the different ways of renaming and moving files





1. You just created a new file, but then you decided that the file is to be removed. How do you delete this untracked file.

ANS: Un tracked file can be directly deleted using rm command or if its under gits version control ,we need to delete it using git command.

**git clean –f index002.html**

****

1. Demonstrate the following:
   1. delete of a tracked file🡪these can be achieved by 2 options:-

1. To keep the local file for you, but delete for everyone once they try to pull

**git rm - -cached <file-name> or git rm –r - -cached <folder-name>**

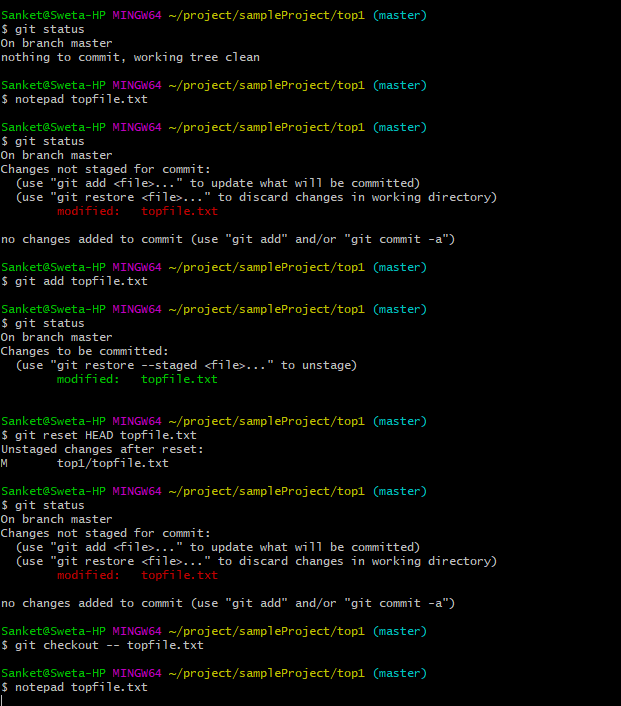
2.For optimization with large no of files, the assume-unchanged index will be reset and files overwritten if there is upstream changes to files/folder

**git update-index - -assume-unchanged <path-name>**

* 1. backing out staged deletion🡪Done some changes in file and staged it also but later we don’t want that changes,so we do

git reset HEAD file-name🡪this will only bring back to working directory but still in a staged area, so delete it from staging area also do,

**git checkout - - file-name**

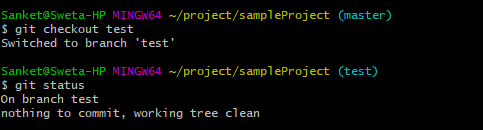
****

* 1. recursive deletion🡪to delete all file of given directory recursively we use **git rm –r <directory-name>**

1. You do not want to push certain folders/files of your project; how do you manage this?

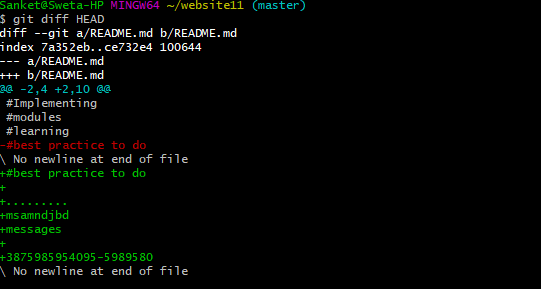
ANS: By saying them as or storing it into .gitignore files, which will directly be understandable by git that these files are not of use and need to ignored while committing any changes.

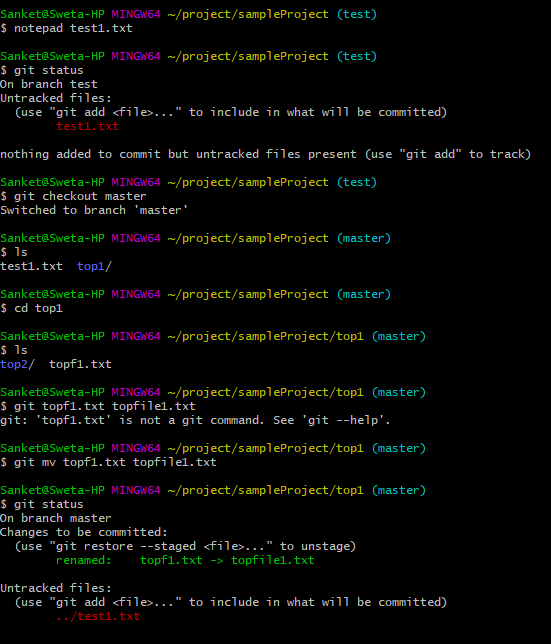
1. Create a branch called “test”. Make some changes in the master branch. Let there be some changes in the working directory and some in the staging area. Make some changes in the test branch as well. Issue the command to show the differences for



* 1. Working directory vs Staging area

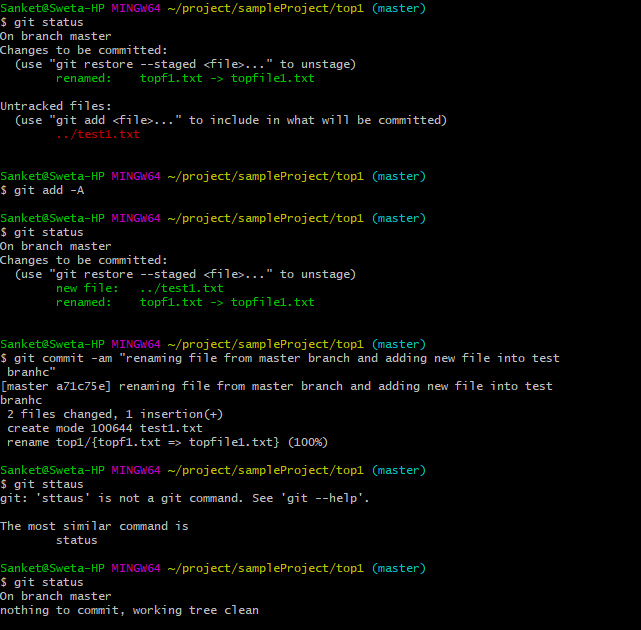
Use of **git diff HEAD** command to check the differences .





* 1. Working directory vs Local Repository

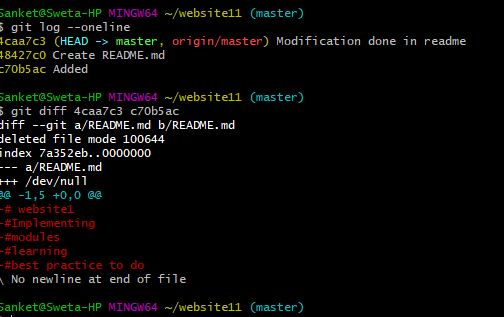
Files in working directory need to be commited(git commit) all the changes from the staging area,wrap it and put them into local repository.



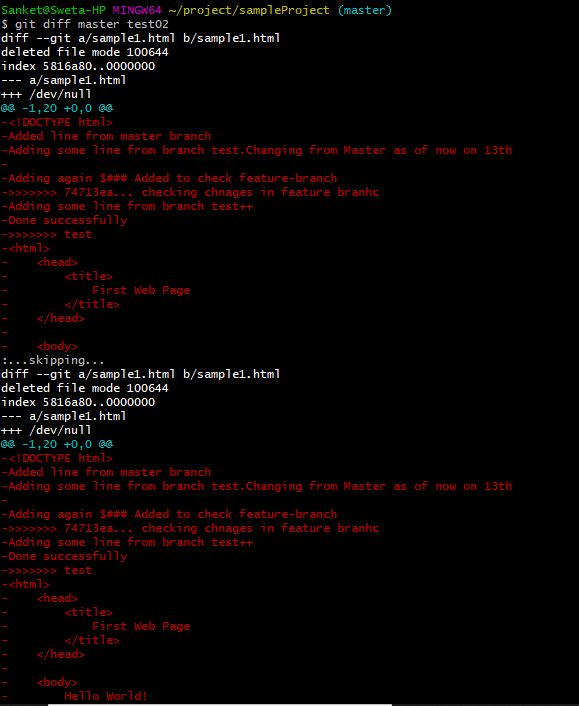
* 1. Staging area vs Local Repository

ANS: The layer between working directory and local directory comes staging area, when we add our changed file using git add it comes into staging area and then when we do git commit into push changes from staging to local reporting.

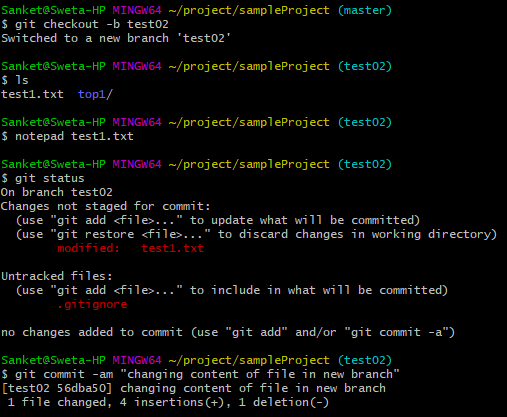
* 1. Between two commits

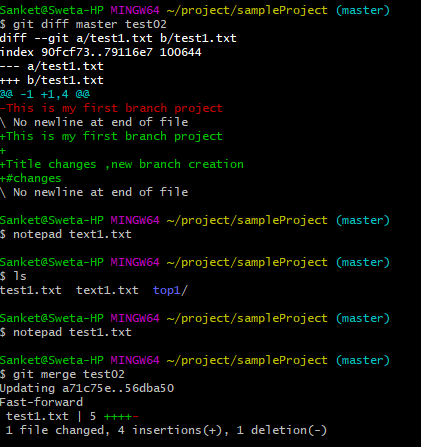


* 1. Between two tags
  2. Local vs Remote Repository
  3. Master branch vs test branch

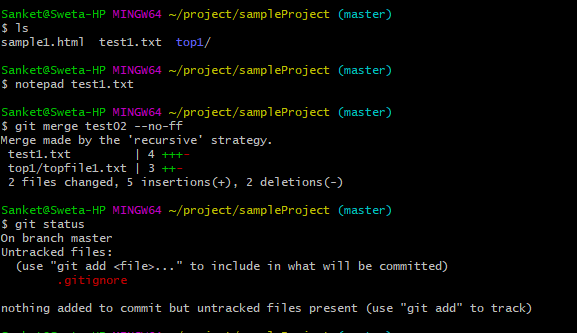


1. Merge the changes from test branch to master branch.
   1. FastForward merge



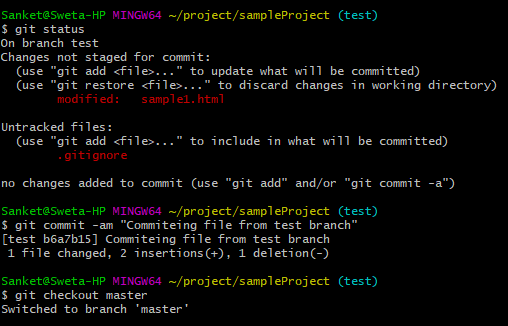


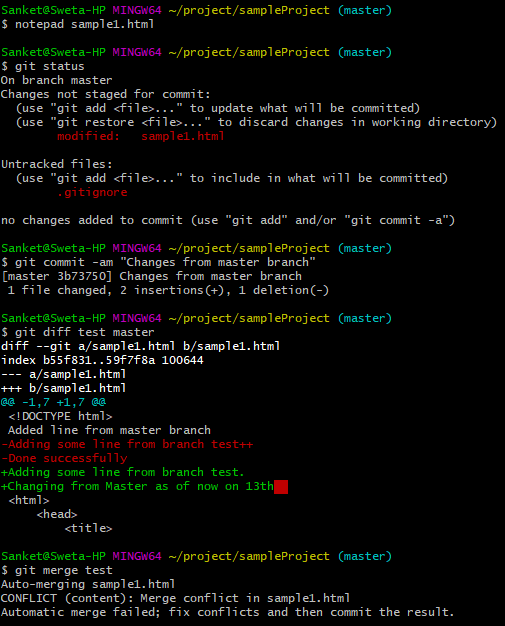
* 1. Disabling FastForward merge

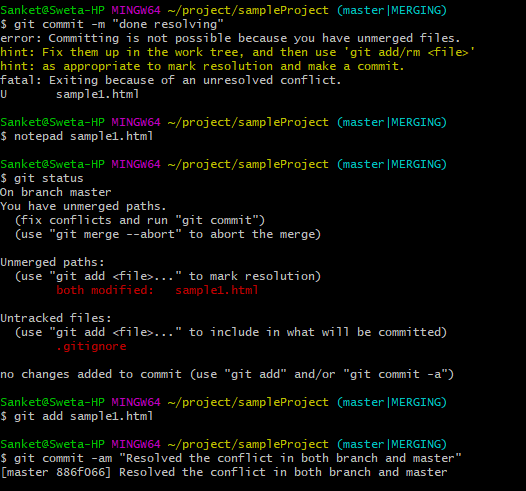


* 1. What is the difference between option a and option b🡪In fastforward merge no new commit is done on the master but when we disable fastforward merge in master ,a new commit is done on the master.Hence merge commit is created in recursive merge.

1. Create a merge conflict situation. Resolve the conflict and merge the changes between the branches.



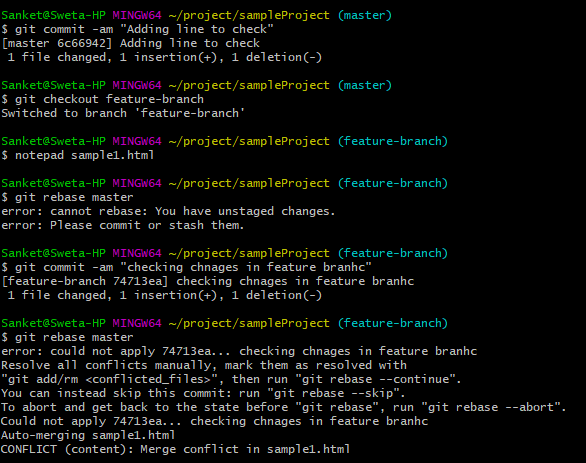


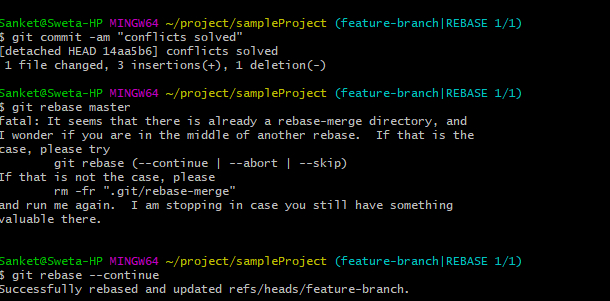


1. What is the difference between merge and rebase, demonstrate with the eg.

Merge will create a new “merge commit” in the test branch that ties together the history of both branches whereas rebase gets all unique commits from both branches and applies them one by one.

For example-If we modify some files in feature-branch and also some changes in same file from master and do git rebase ,it adds new commit line into our file and then successfully rebased.

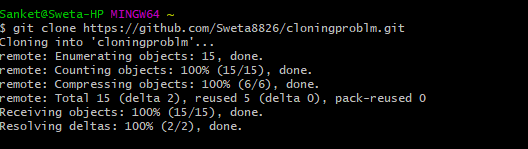


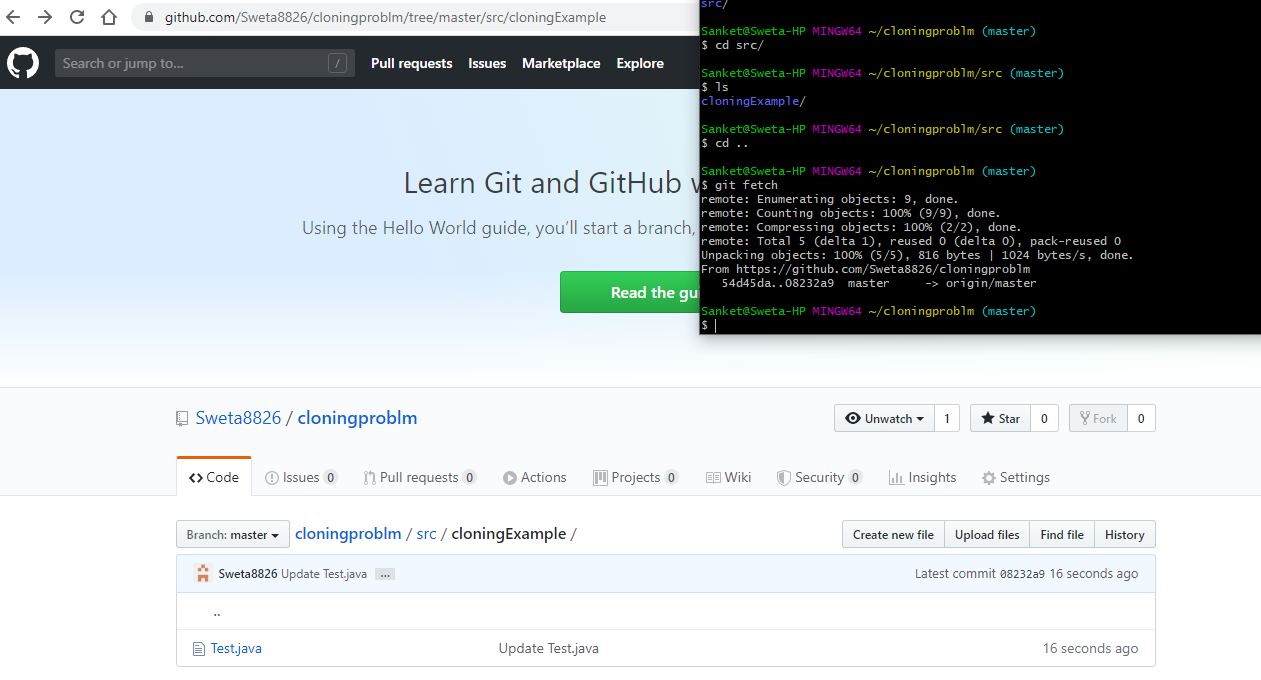


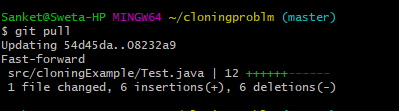
13.With an example, demonstrate fetch, clone and pull. What is the usecase for these operations. Are they same, different? Explain

ANS: Git fetch will just retrieve the metadata info from the original more like just checking is there any changes available and git pull will retrieve and bring all that changes from the remote repository .On other hand ,git clone will retrieve the exact copy of your remote/local repository into new directory.

After the clone, a plain git fetch without arguments will update all the remote-tracking branches, and a git pull without arguments will in addition merge the remote master branch into the current master branch.







1. Create a new repository in Github, with a README file. While pushing to the remote repository, if the remote branch is ahead of the local repository (new file is added in remote repository, which is not there in local repository) and pull is failing, how do you solve this problem?

