# GIT Assignment

**Instructions –**

1. **All the steps below should be performed in the given order.**
2. **All the actions should be performed using the git command line tool only.**
3. **Do not use any web or GUI based functionality unless specifically mentioned.**

**Steps to perform: -**

1. Download and install the git using this URL. https://git-scm.com/downloads
2. Clone your project folder from GitLab to your system using command prompt/gitbash.
3. Create a folder named – GitAssignment in your cloned solution.
4. Take a screenshot of the **command** used to clone the solution and save it to above folder.
5. Check the **status** of your git project.
6. Add and commit the files to git project.
7. Take the screenshot of the command prompt with output of your **status command** and **add and commit commands**.
8. Push your code to remote. Check the status of your project on GitLab. Take screenshots of your GitLab portal and command prompt and keep it handy.
9. Create a new branch named ‘GitlabFiles’. Checkout to the new branch.
10. Using command prompt, display the list of the branches in your project and show the current active branch. Take a screenshot of your command and the output and save it to your solution. You should be on the newly created GitlabFiles Branch
11. Again, check the status of your project, add, commit, and push your files to remote.
12. Go to Gitlab portal, observe the list of branches on the portal and checkout the newly created branch. Your screenshots from step 8-11 should be present in the new branch but NOT in the master branch.
13. Create a pull/merge request and share the URL with your mentor via the word document.
14. Now complete the pull request from the GitLab portal. Switch to master branch on the GitLab portal an observe that all the changes from the GitLabFiles branch are now present in the master branch.
15. Go back to your command prompt and switch to master/main branch.
16. Observe that the new files present on the GitLab are not present on your system. Now get those files to your system using the git commands. Take a screenshot of your command.
17. Commit and push any changes on your project.
18. Create a folder inside your GitAssignment folder and name it ‘IgnoreMe’. Create a text file inside this new folder and name it ‘InsideIgnoreMe.txt’
19. Check the status from your command. You should see your new folder and files. Take a screenshot and add it to GitAssignment folder.
20. Create a .gitignore file so that the newly created folder does not show up in the git status. i.e., git should not be able to read/push it even though it is present in the project.
21. Commit and push your changes. Only the new screenshots and .gitignore file should be pushed.