**Assignment A1**

* Install GIT & make sure it is added into PATH.

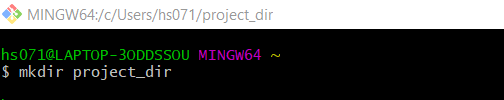
Installed Git and added git into the PATH.

1. Create a directory ‘project\_dir’ & cd to ‘project\_dir’.

**Command used to make new directory:**

mkdir project\_dir

**Ouput:**



**Command used to move to the directory:**

cd project\_dir

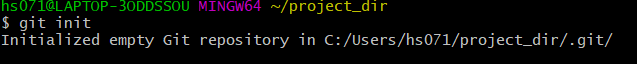
**Output:**



1. Initialize git version database. (git init).

**Before initializing git repository in our directory, we should initialize the name and email of author with the following commands:**

**Output:**



**3. Create a new file index.html.**

**Command used for creating new HTML file:**

touch index.html

**Output:**

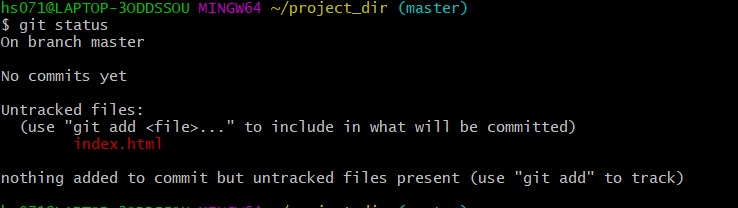


**4. Check the git status. You should find index.html as untracked file.**

**Command used for checking status of the repository:**

git status

**Output:**

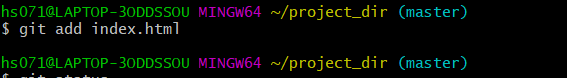


**5. Stage the index.html file.**

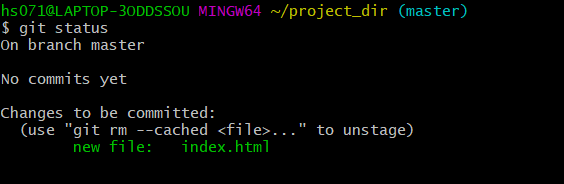
**Command used for staging the files:**

git add index.html

**Output:**



**Command: git status**

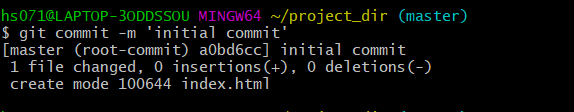


**6. Commit index.html**

**Command used for committing the staged files:**

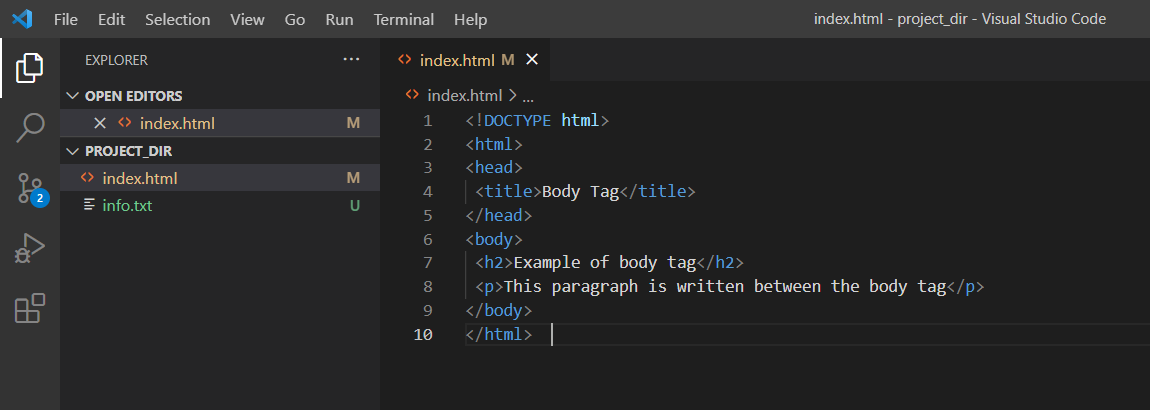
git commit -m “Initial Commit”

**Output:**



**7. Make few changes in index.html & create a new file info.txt file.**

**Changes made in index.html**



**Command used for creating info.txt:**

touch info.txt

**Output:**

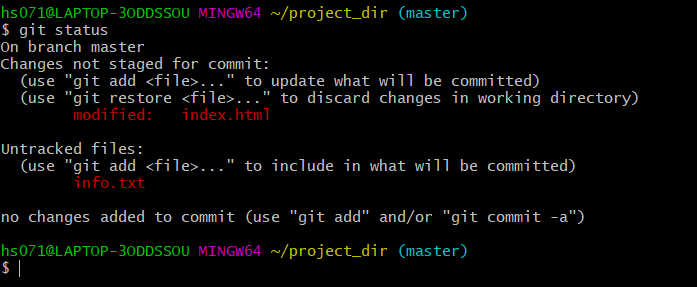


**8. Check git status. You should find index.html & info.txt as untracked files.**

**Command used for checking status of the repository:**

git status

**Output:**



**9. Configure GIT to ignore all txt files.**

* Create a .gitignore file using the following command in git bash:

**Command used for creating .gitignore:**

touch .gitignore

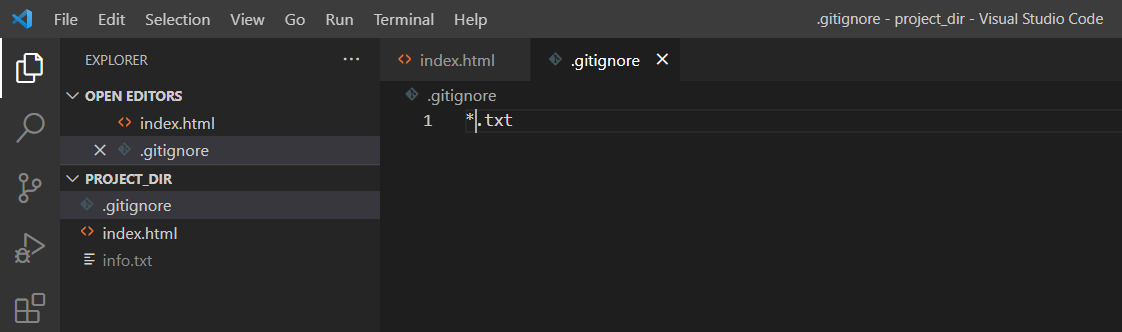
**Output:**



* **Add line in the .gitignore file created in the repository to not include all the text files for the staging**

Line to be inserted for not including all the .txt files in the .gitignore file:

**\*.txt**

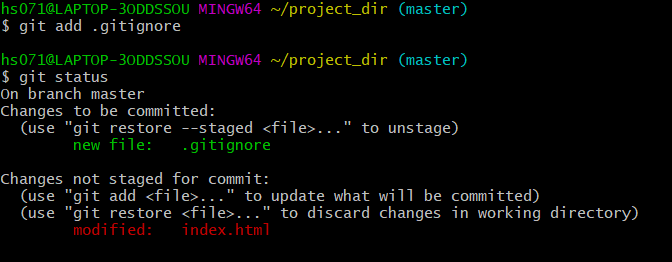


* **Add the .gitignore file to the staging area to make it a tracked file and track all the changes**

**Command used for adding to staging area:**

git add .gitignore

**Output:**

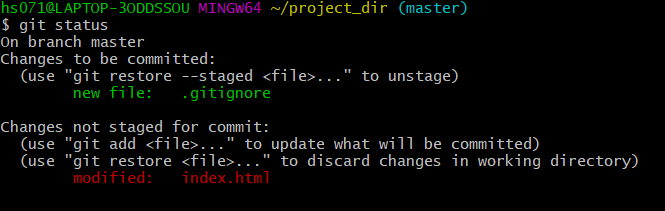


**10.** **Again check the git status. You should find only index.html as untracked file.**

**Command used for checking status of the repository:**

git status

**Output:**

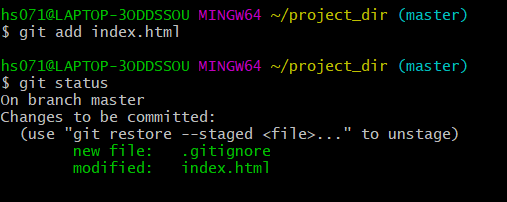


**11. Stage & commit index.html.**

**Command used for staging index.html:**

git add index.html

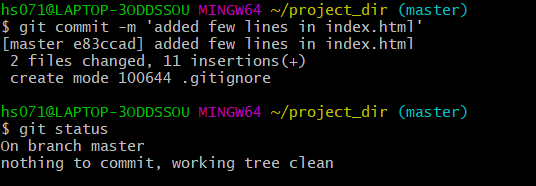
**Output:**



**Command used for commit index.html:**

git commit -m “added few lines in index.html”

**Output:**

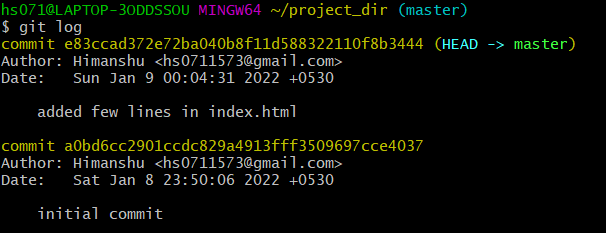


**12. Log all your comments so far.**

**Command used for logging commit:**

git log

**Output:**

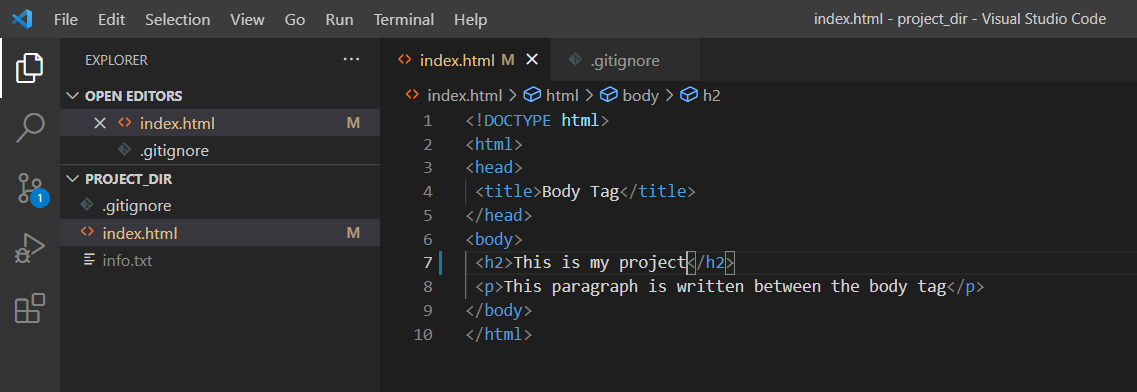


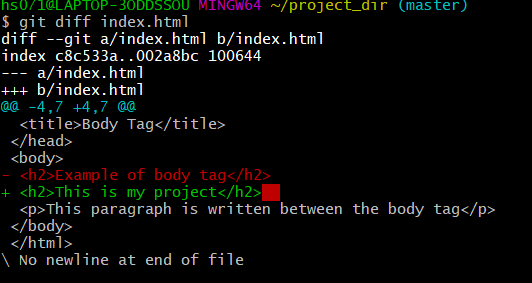
**13. Make some changes in index.html.**

**Command used for checking the differences:**

git diff index.html

**Output:**



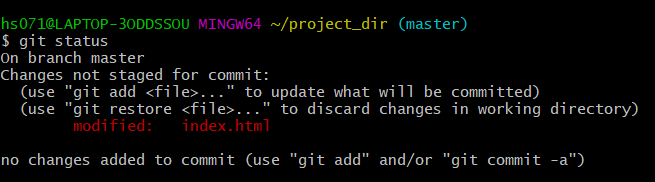


**14. Revert the change made in the previous step using git command.**

**Command used for reverting back the changes made:**

git checkout index.html

**Output:**

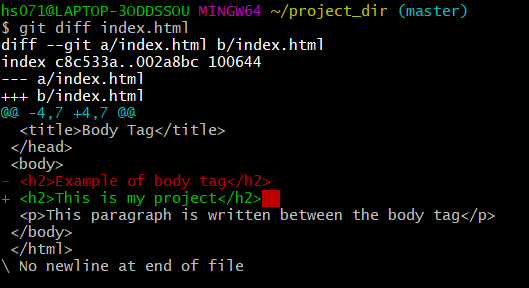


**15. Again change index.html.**

**Command used for checking the differences:**

git diff index.html

**Output:**

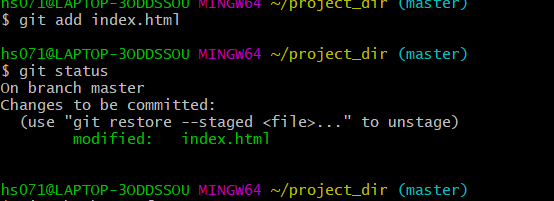


**16. Stage index.html**

**Command used for staging index.html:**

git add index.html

**Output:**

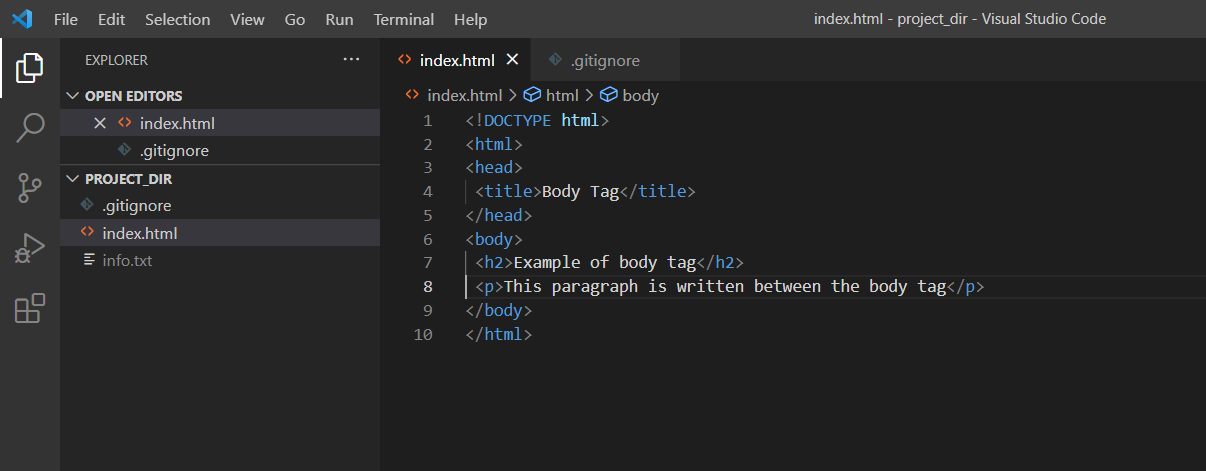


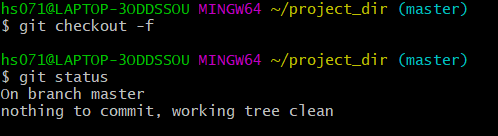
**17. Revert back the last stage.**

**Command used for reverting back to previous commit:**

git checkout -f

**Output:**



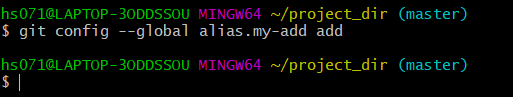


**18. Rename ‘add’ command to ‘my-add’.**

**Command used for renaming ‘add’ to ‘my-add’:**

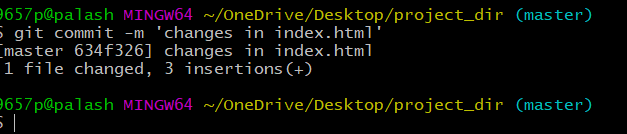
git config --global alias.my-add add

**Output:**



**19.** **Using my\_add command Stage index.html again & commit the changes**

**Output:**



**20. Revert the last commit.**

**Command used for reverting to the last commit:**

>>git revert HEAD

**Output:**



