**Git command practice**

**Reset:**

 The git reset command is used to reset the changes. The **git reset** command has three core forms of invocation.

* **Soft**
* **Mixed**
* **Hard**

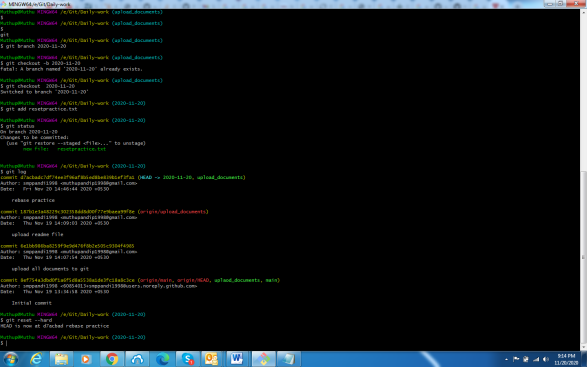
**Git hard:**

* It will move the HEAD pointer.
* It will update the staging Area with the content that the HEAD is pointing.
* It will update the working directory to match the Staging Area.
* It is the most direct, unsafe, and frequently used option

Step 1) Add new file using **git add** command

Step 2) check the status of the repository using **git status**

Step 3) check the status of the Head and previous commits using **git log**



**Git reset mixed:**

It is default option for git reset.In this method only delete in staging area and does not delete on local storage

Step 1) Create new file then add staging area using **gir add** command

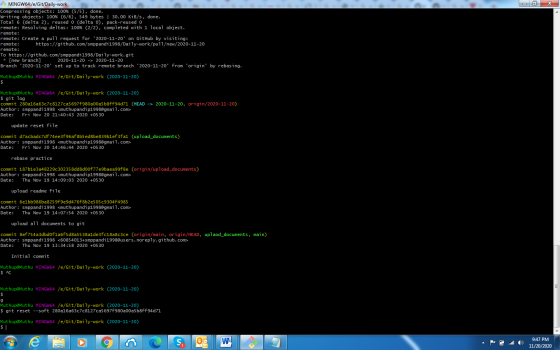
Step 2)Check the git status using **git status** command

Step 3) Enter reset using **git reset** command



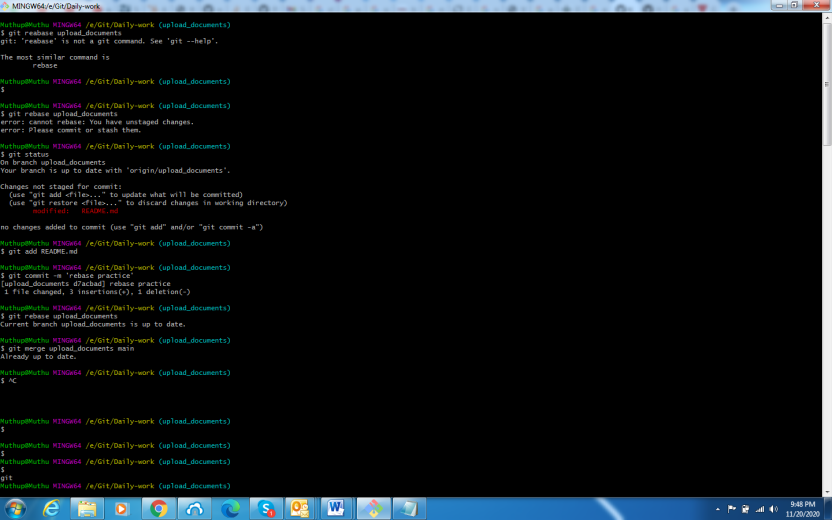
**Git reset soft:**

The soft option does not touch the index file or working tree at all, but it resets the Head as all options do.



**Rebase:**

It is used to integerate change one branch to another branch.**Git rebase** compress the all changes in to single patch and then it integerates the patch on to the target branch.In the process eliminate the history.



**Merge vs rebase:**

**Merge:**

* Simple and familiar
* It have complete history
* Maintains the comtext of the branch
* If you work with small team that time use merge option

**Rebase:**

* It is used to integerate change one branch to another branch.**Git rebase** compress the all changes in to single patch and then it integerates the patch on to the target branch.In the process eliminate the history.
* It is used to work with large team

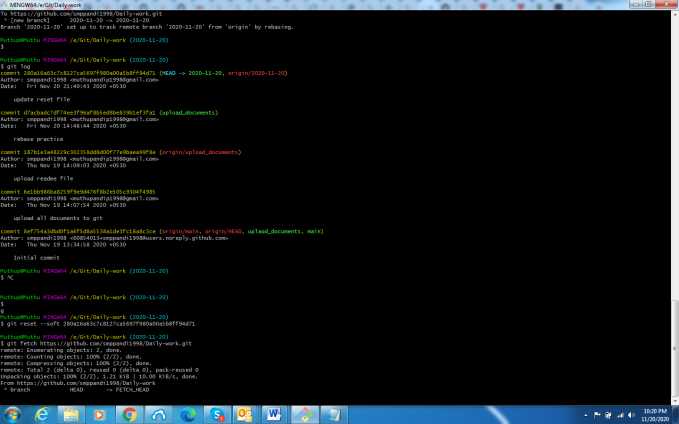
**Fetch:**

Git "fetch" Downloads commits, objects and refs from another repository. It fetches branches and tags from one or more repositories.

**Scenario1** :To fetch the remote repository

**Syntax:**

$ git fetch< repository Url>



### Scenario 2: To fetch a specific branch

### Syntax: $ git fetch <branch URL><branch name>

### 

### Git fetch vs pull:

### Git fetch:

### Fetch downloads only new data from a remote repository.

### Fetch is used to get a new view of all the things that happened in a remote repository.

### Fetch never manipulates or spoils data.

### It protects your code from merge conflict.

### It is better to use git fetch command with git merge command on a pulled repository.

### Git pull:

### Pull is used to update your current HEAD branch with the latest changes from the remote server.

### Pull downloads new data and directly integrates it into your current working copy files.

### Pull downloads the data and integrates it with the current working file.

### Git tags:

### Tags are used to mark a commit stage as relevant. We can tag a commit for future reference

### Step 1) First checkout your branch git checkout <branch name> command

### Step 2)Create tag using git tag <tagname> command

### Step 3) List of all tags type git tag command

Step 4)  display the details of a particular tag **$ git tag show <tagname>**

### 

### git config:

### This command is used to login purpose and specify configuration settings.

### Git stores all global configurations in .gitconfig file, which is located in your home directory

### if you omit --global option, then your configurations are specific for the current Git repository.

### Setting username:

$ git config --global user.name "smppandi1998"

### Setting email id:

$ git config --global user.email[muthupandip1998@gmail.com](mailto:muthupandip1998@gmail.com)

### Color highlighting:

The following commands enable color highlighting for Git in the console.

* $ git config --global color.ui true
* $ git config --global color.status auto
* $ git config --global color.branch auto

### Listing Git settings:

### This command is used to verify your git settings.

### 

### 

### Git init:

### The git init command is used to create a new blank repository. It is used to make an existing project as a Git project.

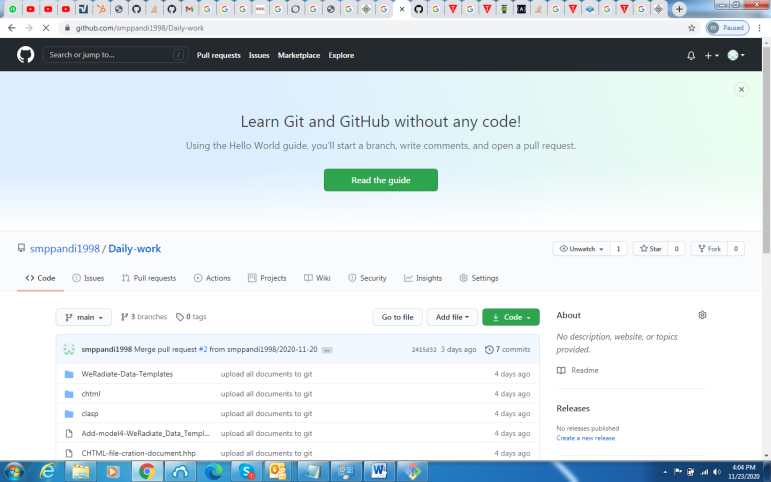
### 

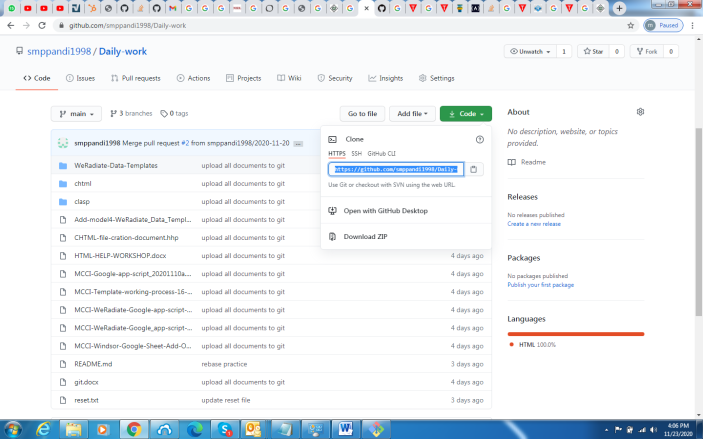
**Git clone:**

This command is used to make a local copy of a remote repository.

**Step 1**: Open GitHub and navigate to the main page of the repository.

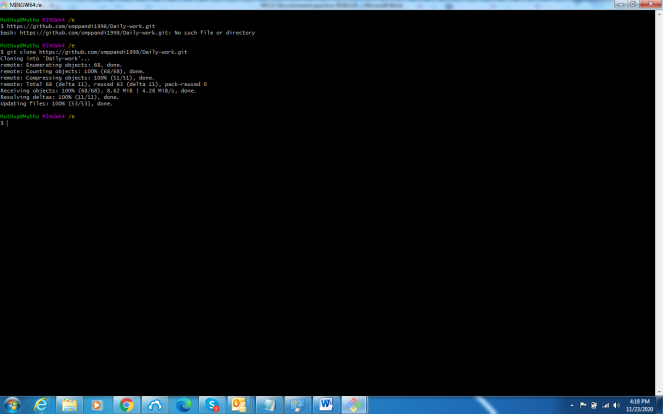
**Step2**: Under the repository name, click on **Clone or download**.



**Step 3:**Select the **Clone with HTTPs section** and **copy the clone URL** for the repository. For the empty repository, you can copy the repository page URL from your browser and skip to next step. 

Step 4:Use the git clone command with repository URL to make a copy of the remote repository. See the below command:

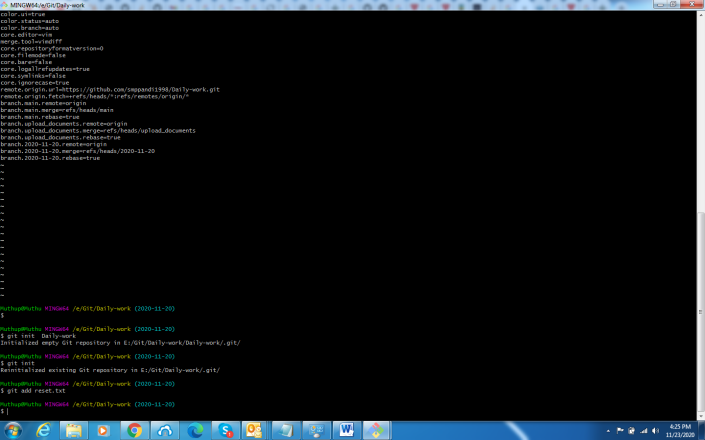
<https://github.com/smppandi1998/Daily-work.git>



**Git add:**

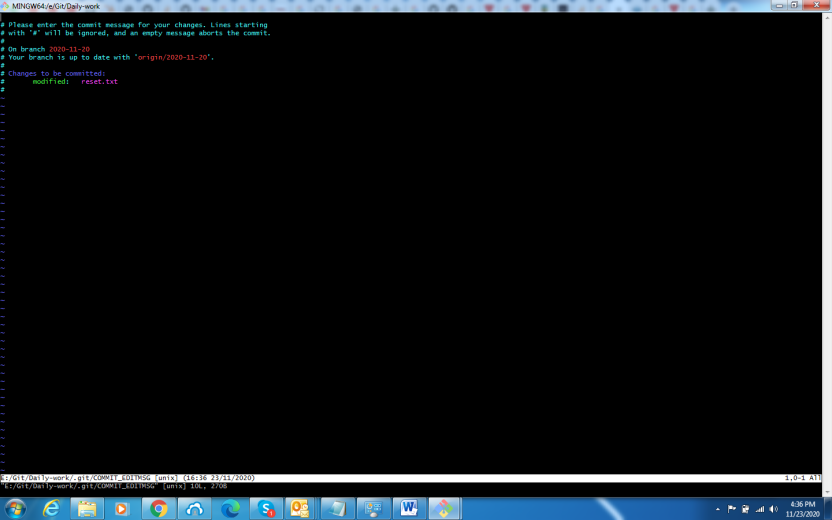
This command is used to add file contents to the Staging Area

$ git add <File name>



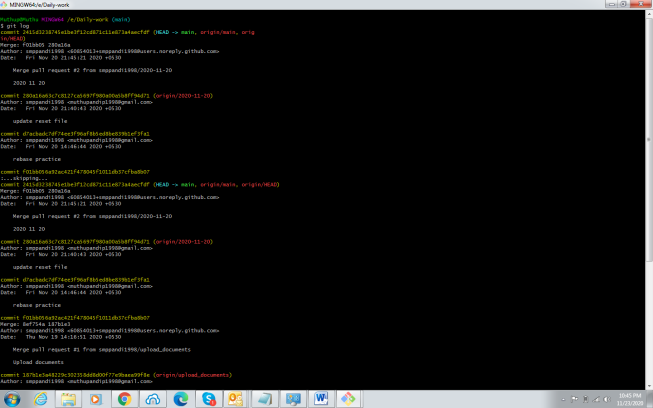
**Git commit:**

It is used to record the changes in the repository. It is the next command after the [git add](https://www.javatpoint.com/git-add).

**Git log:**

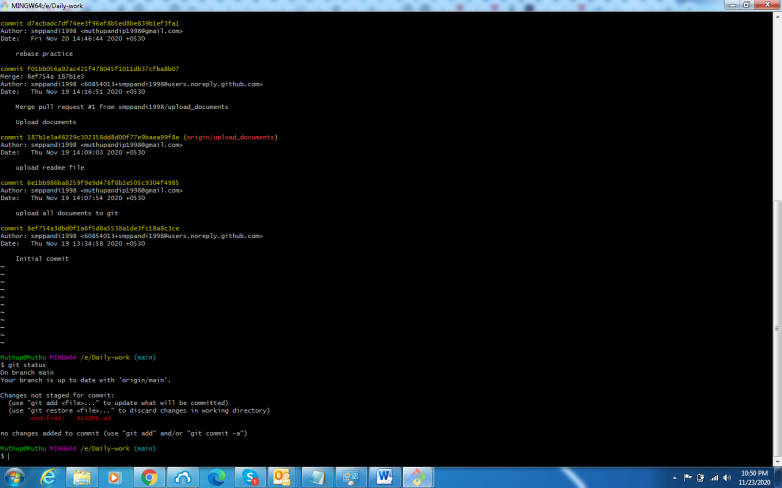
This command is used to list the recent commands

$ git log



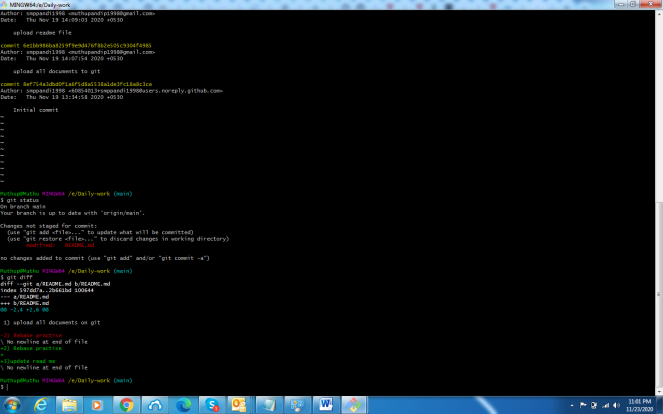
**Git status:**

This command will display the untracked file from the repository



**Git diff:**

This command is used to show changes commit and branches.



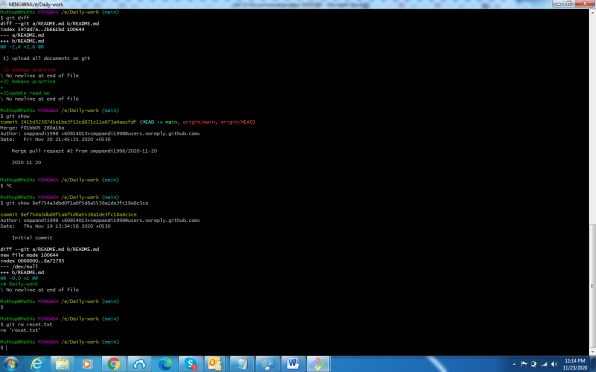
**Git show:**

This command is used to show the various type of object.

### 

### Git rm:

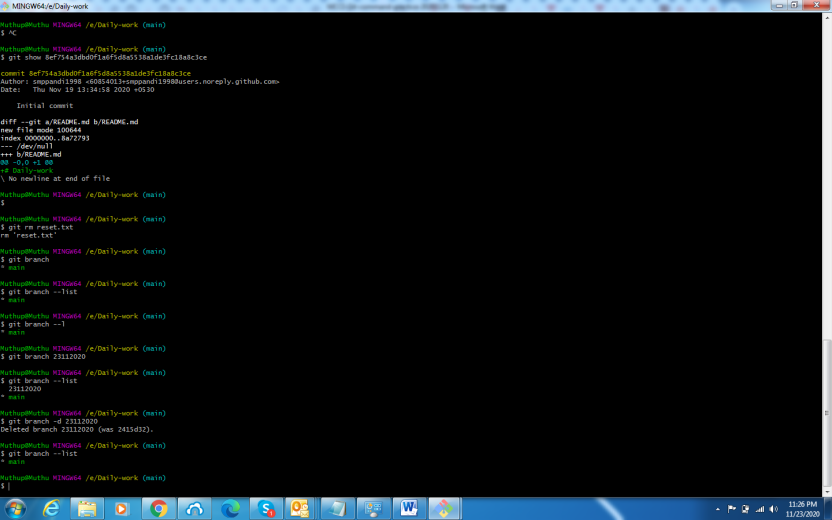
This command is used to remove file from repository.



**Git branch:**

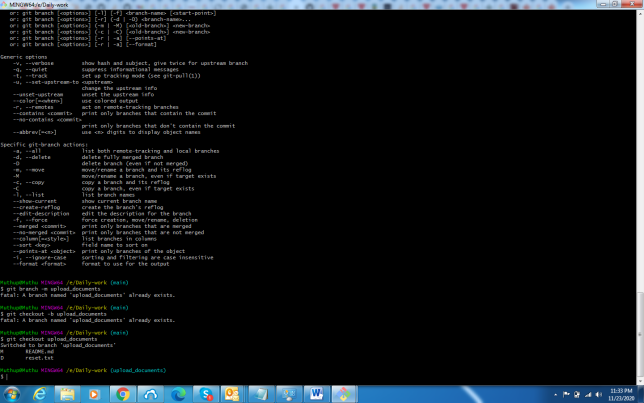
This command is used to create branch,list out branch or delete branch.

* $git branch –list out branches from repository
* $git branch –d is delete the branches
* $git branch <branch name> is create new branch



**Git checkout:**

This command is used to switch in to branch.**$git branch <branchname>**



**Git remote:**

## Check your Remote:

To check the configuration of the remote server, run the **git remote** command.

$ git remote

**Git remote -v:**

**This command is used to** show the URLs that Git has stored as a short name.

$ git remote -v

## Git Remote Add:

This command is used to add remote server.

