

Aim: Setting up of Git Client

Theory:

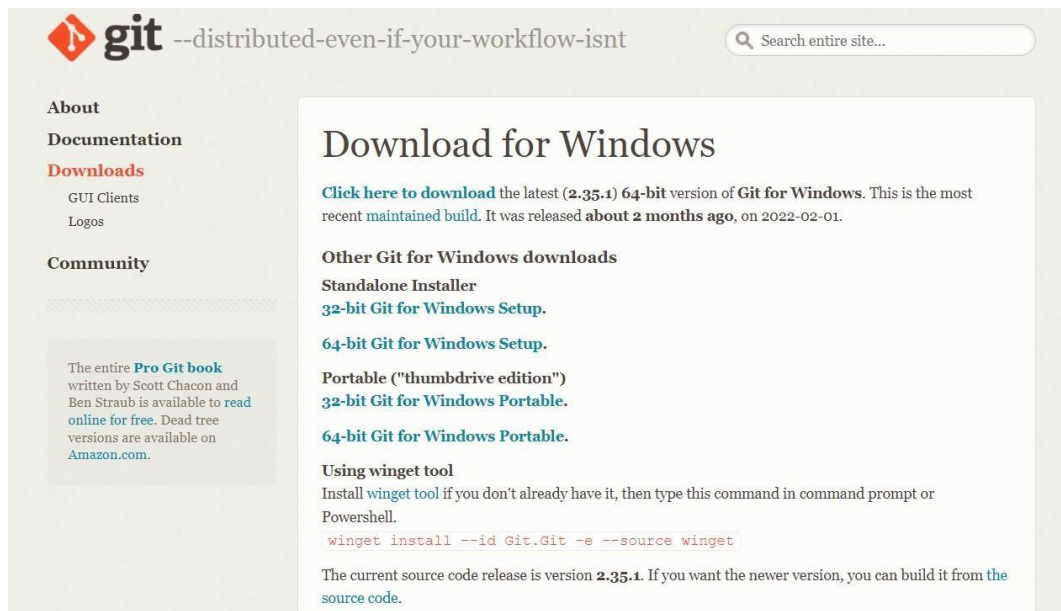
GIT → It is basically used for pushing and pulling of code. We can use git and git-hub parallelly to work with multiple members or individually. We can make, edit, recreate, copy or download any code on git hub using git.










What is GIT? → It's a Version Control System (VCS) → It is a software, or we can say a server by which we are able to track all the previous changes in the code.

Advantages of GIT →

Procedure: We can install Git on Windows, using the most official build which is available for download on the GIT's official website or by just typing (scmgit) on any search engine. We can go on <https://git-scm.com/download/win> and can select the platform and bitversion to download. And after clicking on your desired bit-version or ios it will start downloading automatically.

Snapshots of download:



	Name	Date modified	Type	Size
	 Git Bash	16-03-2022 08:51	Shortcut	2 KB
	 Git CMD	16-03-2022 08:51	Shortcut	2 KB
	 Git FAQs (Frequently Asked Questions)	16-03-2022 08:51	Internet Shortcut	1 KB
	 Git GUI	16-03-2022 08:51	Shortcut	2 KB
	 Git Release Notes	16-03-2022 08:51	Shortcut	2 KB

```
Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (master)
$ |
```

Experiment No. 02

Aim: Setting up GitHub Account

Theory:

What is GitHub -> GitHub is a website and cloud-based service (client) that helps an individual or a developers to store and manage their code. We can also track as well as control changes to our or public code.

Advantages of GitHub -> GitHub's has a user-friendly interface and is easy to use .We can connect the git-hub and git but using some commands shown below in figure 001. Without GitHub we cannot use Git because it generally requires a host and if we are working for a project, we need to share it will our team members, which can only be done by making a repository. Additionally , anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects.

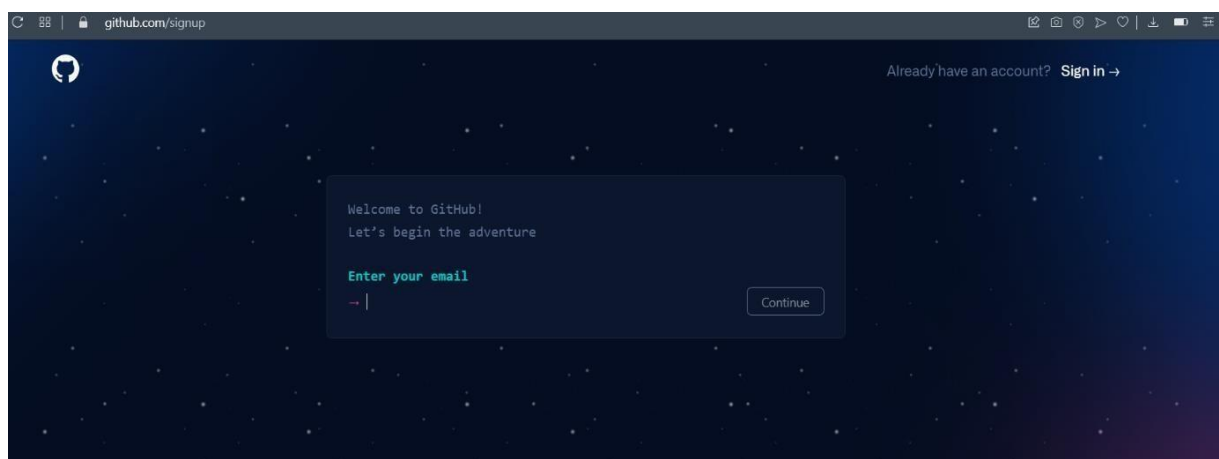
Procedure:-

Step1 :-

Google (any search engine)
Search for git-hub or (<https://github.com/signup>).


Step2 :-

Snapshots –



After visiting the link this type of interface will appear, if you already have account you can sign in and if not you can create.

Sign in into GitHub :-



Sign in to GitHub

Username or email address

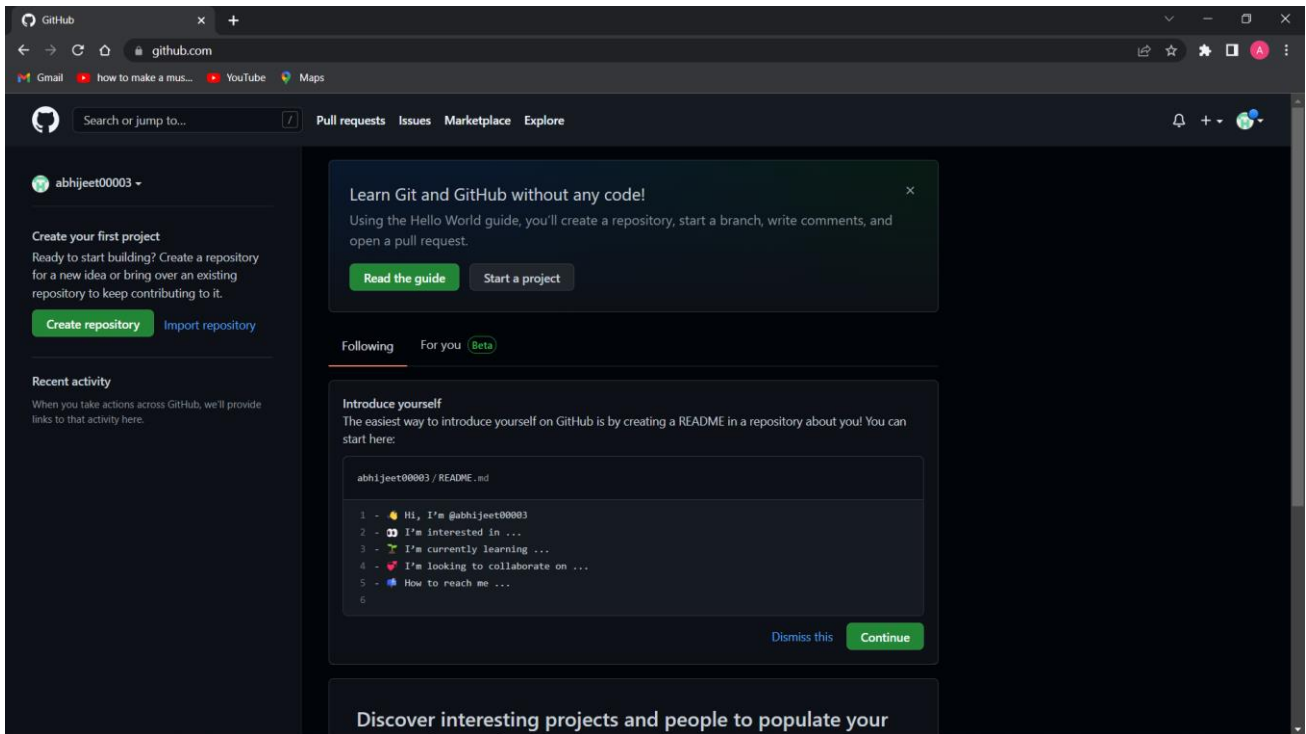
Password

[Forgot password?](#)

Sign in

New to GitHub? [Create an account.](#)

Interface of GitHub :-



To link GitHub account with Git bash –

For username:-

git config --global user.name "username in git-hub"

For user email:-

git config --global user.email "your email in git-hub"

To verify:-

git config user.name

git config user.email

Snapshot :-

```
Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (master)
$ git config --global user.name "abhijeet00003"

Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (master)
$ git config --global user.email "abhijeet0040.be21@chitkara.edu.in"
```

Theory: -

Logs -> Logs are nothing but the history which we can see in git by using the code git log. It contains all the past commits, insertions and deletions in it which we can see any time.

Why logs -> Logs helps to check that what were the changes in the code or any other file and by whom. It also contains the number of insertions and deletions including at which time it was changed.

Snapshots –

```
Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (master)
$ git log
commit 74a6aa19fb474a8e23a7028d5ea46ea9ee51af32 (HEAD -> master)
Author: abhijeet00003 <abhijeet0040.be21@chitkara.edu.in>
Date: Sun Apr 10 14:42:40 2022 +0530

    first commit| g1 repo| demo for class
```

Aim: Create and visualize branches

Create branches:-

The main branch in git is called as master branch. But we can make branches out of this main master branch. All the files present in master can be shown in branch but the file which are created in branch are not shown in master branch. We can also merge both the parent (master) and child (other branches).

Syntax: -

1. For creating a new branch. git
branch name of branch

```
Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (master)
$ git branch activity1
```

2. We can also check how many branches we have.
git branch

```
Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (master)
$ git branch
  activity1
* master
```

3. To change the present working branch.
git checkout name of branch.

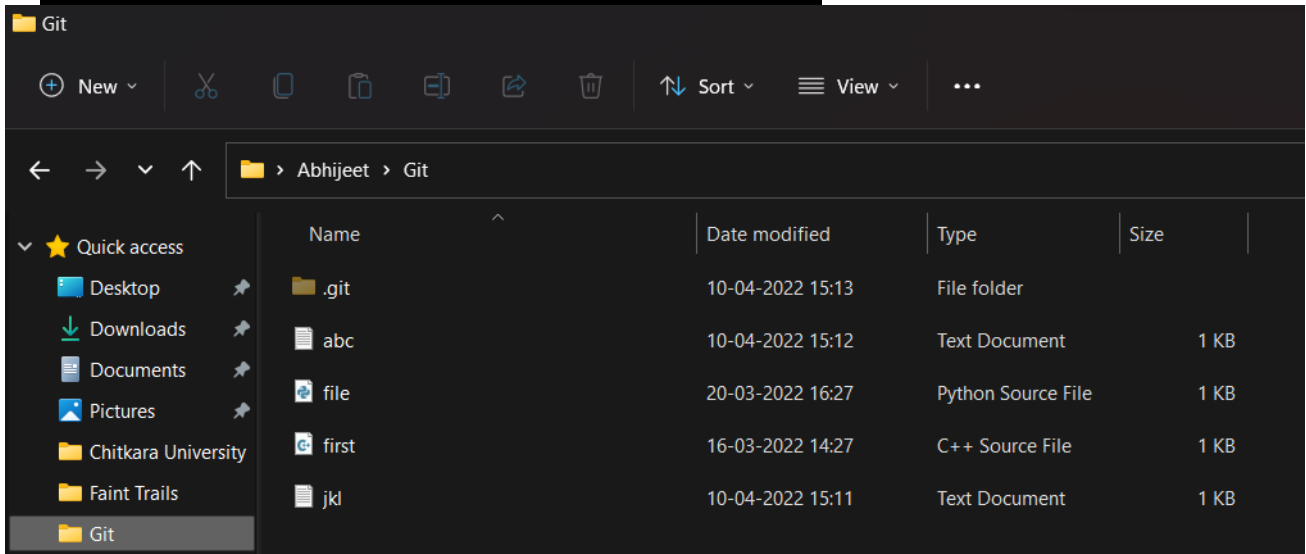
```
Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (master)
$ git checkout activity1
Switched to branch 'activity1'
```

Visualizing branches :-

To visualize I have created a new file in a new branch activity 1 instead of master branch.

```
Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (master)
$ git branch
  activity1
  activity2
  activity3
* master

Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (master)
$ git checkout activity1
Switched to branch 'activity1'
```



After this I have done the 3 step architecture which is tracking the file , send it to staging area and finally we can role back to any previously saved version of this file.


```

Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (activity1)
$ git status
On branch activity1
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        abc.txt
        jkl.txt

nothing added to commit but untracked files present (use "git add" to track)

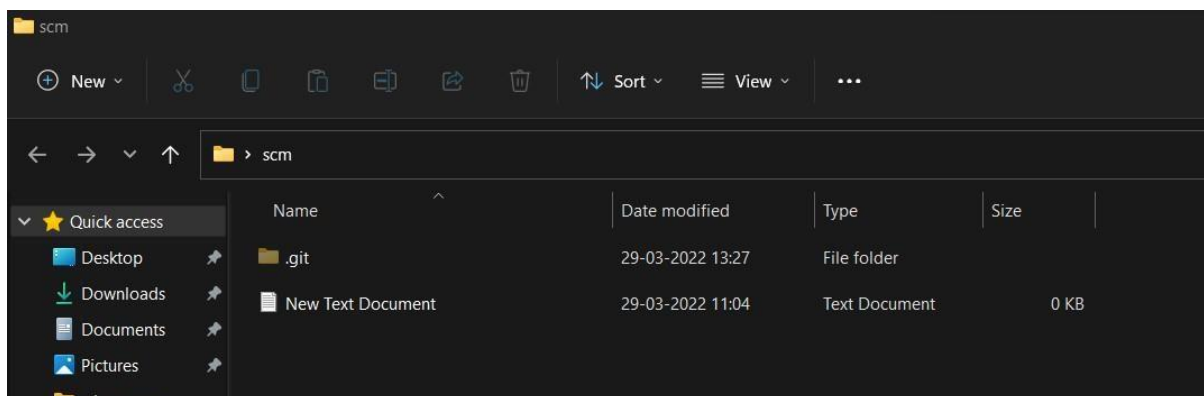
Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (activity1)
$ git add --a
warning: LF will be replaced by CRLF in abc.txt.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in jkl.txt.
The file will have its original line endings in your working directory

Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (activity1)
$ git status
On branch activity1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   abc.txt
        new file:   jkl.txt

Abhijeet@LAPTOP-CB60EGJ8 MINGW64 ~/git (activity1)
$ git commit -m "2 new files added"
[activity1 9a02870] 2 new files added
2 files changed, 4 insertions(+)
create mode 100644 abc.txt
create mode 100644 jkl.txt

```

After this we will change the branch from activity1 to master, but when I will switch to the master branch there will not be the same file in the master , it will not show the new file in the master branch.



In this way we can create and change different branches . We can also merge the branches by using git merge command.

Aim: Git lifecycle description

Theory:

Stages in GIT Life Cycle -> Files in a Git project have various stages like Creation, Modification, Refactoring, and Deletion and so on. Irrespective of whether this project is tracked by Git or not, these phases are still prevalent. However, when a project is under Git version control system, they are present in three major Git states in addition to these basic ones. Here are the three Git states:

- Working directory
- Staging area
- Git directory

Working Directory ->

Consider a project residing in your local system. This project may or may not be tracked by Git. In either case, this project directory is called your Working directory.

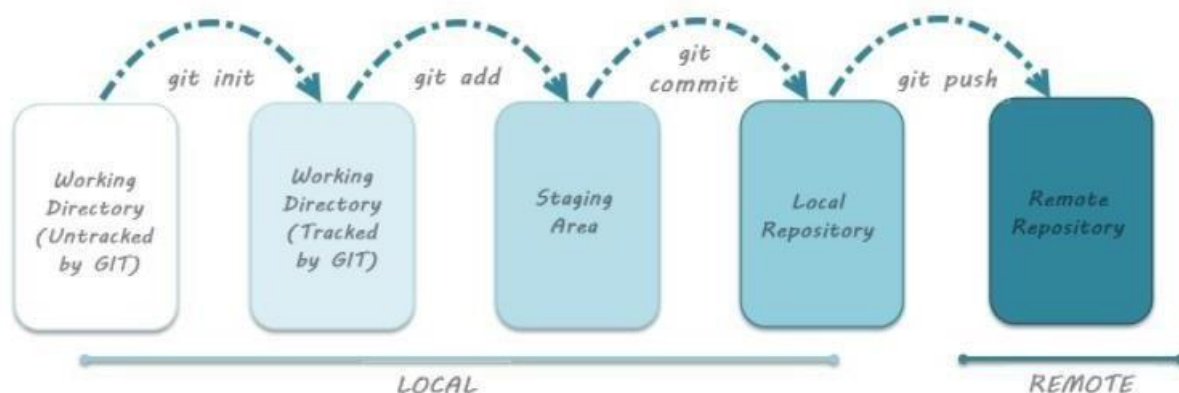
Staging Area ->

Staging area is the playground where you group, add and organize the files to be committed to Git for tracking their versions.

Git Directory ->

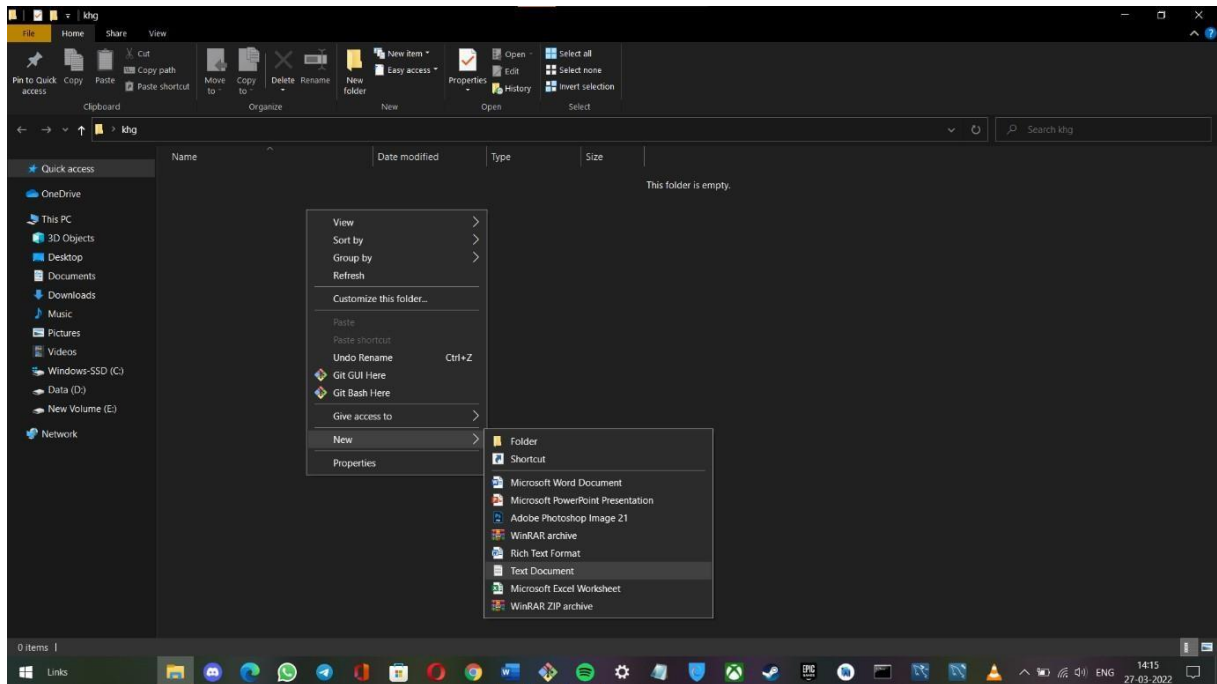
Now that the files to be committed are grouped and ready in the staging area, we can commit these files. So, we commit this group of files along with a commit message explaining what is the commit about. Apart from commit message, this step also records the author and time of the commit. Now, a snapshot of the files in the commit is recorded by Git. The information related to this commit is stored in the Git directory.

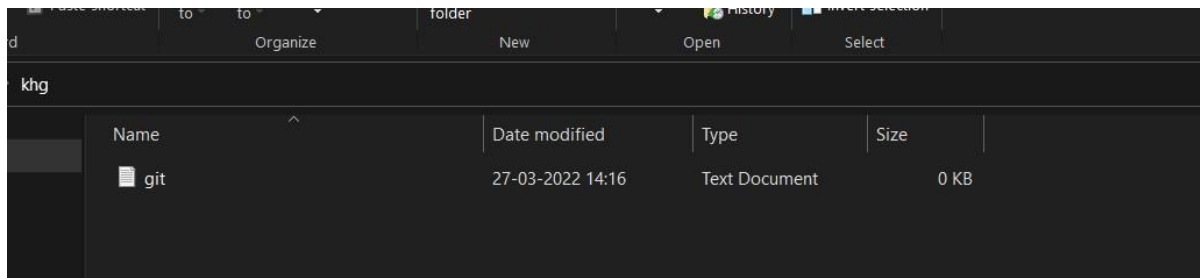
Remote Repository-> means mirror or clone of the local Git repository in GitHub. And pushing means uploading the commits from local Git repository to remote repository hosted in GitHub.



Snapshots –

```
Abhijeet@LAPTOP-CB60EGJ8 MINGW64 /c
$ git status
fatal: not a git repository (or any of the parent directories): .git
```





```
Abhijeet@LAPTOP-CB60EGJ8 MINGW64 /c
$ git status
fatal: not a git repository (or any of the parent directories): .git

Abhijeet@LAPTOP-CB60EGJ8 MINGW64 /c
$ git init
Initialized empty Git repository in C:/./git/

Abhijeet@LAPTOP-CB60EGJ8 MINGW64 /c (master)
$ git status
warning: could not open directory '$Recycle.Bin/S-1-5-18/': Permission denied
warning: could not open directory '$Recycle.Bin/S-1-5-21-2364141303-2253727387-253573133-1000/': Permission denied
warning: could not open directory 'Documents and Settings/': Permission denied
warning: could not open directory 'PerfLogs/': Permission denied
warning: could not open directory 'ProgramData/Application Data/': Permission denied
warning: could not open directory 'Recovery/': Permission denied
warning: could not open directory 'System Volume Information/': Permission denied
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  $Recycle.Bin/
  Drivers/
  DumpStack.log
  DumpStack.log.tmp
  OneDriveTemp/
  Program Files (x86)/
  Program Files/
  ProgramData/
  Riot Games/
  Users/
  windows/
  avast! sandbox/
  hiberfil.sys
  pagefile.sys
  swapfile.sys
```

```
Abhijeet@LAPTOP-CB60EGJ8 MINGW64 /c/users/abhijeet/git (activity1)
$ git add --a

Abhijeet@LAPTOP-CB60EGJ8 MINGW64 /c/users/abhijeet/git (activity1)
$ git status
On branch activity1
nothing to commit, working tree clean

Abhijeet@LAPTOP-CB60EGJ8 MINGW64 /c/users/abhijeet/git (activity1)
$ |
```

